

(19) World Intellectual Property Organization
International Bureau



(10) International Publication Number
WO 2011/109476 A1

(43) International Publication Date
9 September 2011 (09.09.2011)

(51) International Patent Classification:
G06F 9/30 (2006.01) *G06F 9/38* (2006.01)

Joseph [US/US]; 5775 Morehouse Drive, San Diego, California 92121 (US).

(21) International Application Number:
PCT/US2011/026815

(74) Agent: TALPALATSKY, Sam; 5775 Morehouse Drive, San Diego, California 92121 (US).

(22) International Filing Date:
2 March 2011 (02.03.2011)

(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.

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
12/716,359 3 March 2010 (03.03.2010) US

(71) Applicant (for all designated States except US): QUALCOMM INCORPORATED [US/US]; Attn: International IP Administration, 5775 Morehouse Drive, San Diego, California 92121 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): CODRESCU, Lucian [US/US]; 5775 Morehouse Drive, San Diego, California 92121 (US). PLONDKE, Erich James [US/US]; 5775 Morehouse Drive, San Diego, California 92121 (US). INGLE, Ajay Anant [US/US]; 5775 Morehouse Drive, San Diego, California 92121 (US). VENKUMAHANTL, Suresh K. [IN/US]; 5775 Morehouse Drive, San Diego, California 92121 (US). TABONY, Charles

(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).

Declarations under Rule 4.17:

[Continued on next page]

(54) Title: SYSTEM AND METHOD OF PROCESSING HIERARCHICAL VERY LONG INSTRUCTION PACKETS

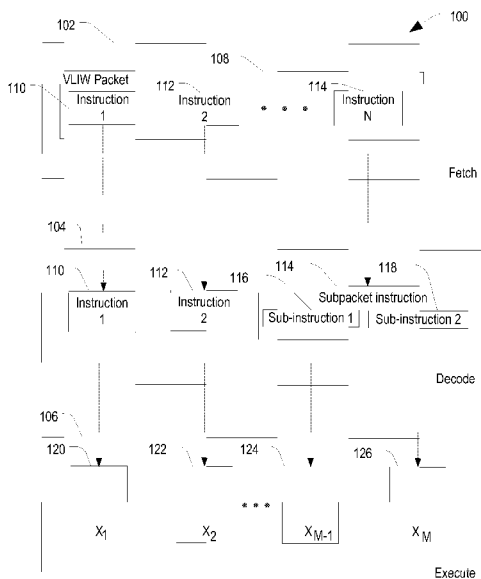


FIG. 1

(57) Abstract: A system and method of processing a hierarchical very long instruction word (VLIW) packet is disclosed. In a particular embodiment, a method of processing instructions is disclosed. The method includes receiving a hierarchical VLIW packet of instructions and decoding an instruction from the packet to determine whether the instruction is a single instruction or whether the instruction includes a subpacket that includes a plurality of sub-instructions. The method also includes, in response to determining that the instruction includes the subpacket, executing each of the sub-instructions.

WO 2011/109476 A1

- *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*
 - *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*
- Published:**
- *with international search report (Art. 21(3))*

SYSTEM AND METHOD OF PROCESSING HIERARCHICAL VERY LONG INSTRUCTION PACKETS

I. Field

[0001] The present disclosure is generally related to encoding a hierarchical very long instruction word.

II. Description of Related Art

[0002] Advances in technology have resulted in smaller and more powerful computing devices. For example, there currently exist a variety of portable personal computing devices, including wireless computing devices, such as portable wireless telephones, personal digital assistants (PDAs), and paging devices that are small, lightweight, and easily carried by users. More specifically, portable wireless telephones, such as cellular telephones and internet protocol (IP) telephones, can communicate voice and data packets over wireless networks. Further, many such wireless telephones include other types of devices that are incorporated therein. For example, a wireless telephone can also include a digital still camera, a digital video camera, a digital recorder, and an audio file player. Also, such wireless telephones can process executable instructions, including software applications, such as a web browser application, that can be used to access the Internet. As such, these wireless telephones can include significant computing capabilities.

[0003] To speed up computing and reduce program size and consequent program storage need, it would be helpful to pack instructions more efficiently. By packing the instructions more efficiently, instruction cache size can be reduced, or the same instruction cache size can hold more instructions.

III. Summary

[0004] In a particular embodiment, a method of processing instructions is disclosed. The method includes receiving a packet of instructions. The packet includes at least one instruction that may be decoded to determine whether the instruction is a single

- 2 -

instruction or whether the instruction includes a subpacket that includes a plurality of sub-instructions. The method also includes, in response to determining that the instruction includes the subpacket, executing each of the sub-instructions.

[0005] In another particular embodiment, the method of processing instruction includes receiving a packet of instructions, where the packet includes a first instruction having a first parse value and a second instruction having a second parse value. The method includes routing the first instruction to a first execution unit and identifying the second instruction as a sub-packet instruction that includes a first sub-instruction and a second sub-instruction. The method also includes routing the first sub-instruction to a second execution unit.

[0006] In another particular embodiment, a processor is disclosed. The processor includes a decoder to receive a packet of instructions. The decoder is operative to decode an instruction within the packet to determine whether the instruction is a first sub-packet instruction that includes a first sub-instruction and a second sub-instruction. The processor also includes an execution unit to execute the instruction or to execute the first sub-instruction.

[0007] One particular advantage provided by at least one of the disclosed embodiments is that instruction cache size can be reduced because instructions are packed more densely. An advantage to efficient packing of instructions is an ability to store more instructions in the same amount of cache. Another advantage provided by at least one of the disclosed embodiments is a reduction in energy needed to fetch code from memory because each packet includes a higher density of instructions, resulting in a reduction in total number of calls to retrieve instructions to execute a program.

[0008] Other aspects, advantages, and features of the present disclosure will become apparent after review of the entire application, including the following sections: Brief Description of the Drawings, Detailed Description, and the Claims.

IV. Brief Description of the Drawings

[0009] FIG. 1 is a block diagram of a particular illustrative embodiment of a very long instruction word (VLIW) packet entering a pipelined processor;

- 3 -

- [0010] FIG. 2 is a block diagram of a particular illustrative embodiment of a VLIW packet that includes a sub-packet instruction;
- [0011] FIG. 3 is a block diagram of a particular illustrative embodiment of a pipelined processor processing a VLIW packet that includes a sub-packet instruction;
- [0012] FIG. 4 is a block diagram of another particular illustrative embodiment of a pipelined processor that is processing a VLIW packet that includes a sub-packet instruction;
- [0013] FIG. 5 is a block diagram of another particular illustrative embodiment of a pipelined processor that is processing a VLIW packet that includes a sub-packet instruction;
- [0014] FIG. 6 is a block diagram of another particular illustrative embodiment of a pipelined processing unit that is processing a VLIW packet that includes a sub-packet instruction;
- [0015] FIG. 7 is a flow chart of a particular illustrative embodiment of a method of processing a VLIW packet that includes a sub-packet instruction;
- [0016] FIG. 8 is a flow chart of another particular illustrative embodiment of a method of processing a VLIW packet that includes a sub-packet instruction; and
- [0017] FIG. 9 is a block diagram of wireless device including a digital signal processor to process VLIW packets that include sub-instructions.

V. Detailed Description

- [0018] Referring to FIG. 1, a particular illustrative embodiment of a very long instruction word (VLIW) packet is disclosed. The VLIW packet 102 includes N instructions including a first instruction 110, a second instruction 112, and additional instructions up to an Nth instruction 114. At a fetch stage 102 of a pipelined processor 100, the VLIW packet 102 is initially fetched from storage (not shown) and enters a decode stage 104 of the pipelined processor 100.

- [0019] At the decode stage, 104 the instruction 114 is identified as a subpacket instruction that includes sub-instructions 116 and 118. From the decode stage 104, the plurality of N instructions in the VLIW packet 102 proceeds to an execute stage 106. The execute stage 106 includes M execution units including execution units 120, 122, 124, and 126. In a particular illustrative embodiment, M may exceed N. In another particular illustrative embodiment, M may be less than N. In another particular illustrative embodiment, M may be equal to N.
- [0020] At the execute stage 106, the first instruction 104 enters execution unit 120. The second instruction 112 enters execution unit 122. The first sub-instruction 116 enters execution unit 124, and the second sub-instruction 118 enters execution unit 126. In a particular illustrative embodiment, each of the other instructions in the VLIW packet 102 can enter a corresponding execution unit at the execute stage 106.
- [0021] In a particular illustrative embodiment, one or more of the instructions in the VLIW packet 108 are sub-packet instructions that include two or more sub-instructions. A VLIW packet that includes one or more sub-instruction packets is referred to as a hierarchical VLIW. In the illustrative embodiment shown in FIG. 1, each of the sub-instructions 116 and 118 enters a different execution unit and the sub-instructions 116 and 118 are executed concurrently. Alternatively, each of the sub-instructions 116 and 118 may be routed to a single execution unit that executes both sub-instructions 116 and 118.
- [0022] In a particular illustrative embodiment, a VLIW packet includes multiple subpacket instructions, and each subpacket instruction includes two or more sub-instructions. In a particular illustrative embodiment a subpacket instruction can be decoded to identify a sub-sub-packet of instructions (not shown), where a sub-subpacket includes a plurality of instructions that are referred to as sub-sub-instructions. In another particular illustrative embodiment, each instruction of a VLIW packet comprises a very long instruction word that includes one or more nested instructions, at least one of which is a sub-packet instruction that includes two or more sub-instructions.
- [0023] There are M execution units at the execute stage 106. When the number M is large enough to accommodate each instruction of the VLIW packet including

- 5 -

instructions 110, 112, and sub-instructions 116, and 118, M is greater than N and all of the instructions of the VLIW packet 108 can be executed concurrently. An advantage to concurrent processing of all instructions in the VLIW packet 108 is faster processing compared with sequential processing of some of the instructions included in the VLIW packet 108.

[0024] Referring to FIG. 2, a particular illustrative embodiment of a VLIW packet is disclosed and generally designated 200. The VLIW packet 200 includes instructions 202, 204, 206, and 208. Each of the instructions 202, 204, 206, and 208 typically includes a fixed number of bits. For example, each of the instructions 202, 204, 206, and 208 may contain 32 bits. Each of the instructions can include a parse field. In a particular illustrative embodiment, the parse field includes two bits. In a particular illustrative embodiment, a first parse value is stored at two adjacent internal bits of a first instruction and a second parse value is stored at two adjacent internal bits of a second instruction. For example, a parse value of 01 is stored at two adjacent internal bits of the instruction 206, and a parse value of 00 is stored at two adjacent internal bits of the instruction 208. In a particular illustrative embodiment, the parse field is located at the centermost bits of the corresponding instruction. As an example of parse fields, the instruction 202 includes a parse field 210 with a stored value of 10, the instruction 204 includes a parse field 212 with a stored value of 01, the instruction 206 includes a parse field 214 with a stored value of 01, and the instruction 208 includes a parse field 216 with a stored value of 00.

[0025] A parse field can be an indicator of whether the instruction is an end-of-packet instruction. In a particular example, a parse field of 10 indicates that the packet is not an end-of-packet and also indicates that the packet is at an end of a hardware loop. In a particular illustrative example, a parse field having a value of 01 indicates an instruction that is not an end-of-packet. In a particular illustrative example, a parse field having a value of 11 indicates an end-of-packet instruction. In a particular embodiment, an instruction with a parse field having a value of 00 also indicates an end-of-packet instruction. An instruction with a parse field having a value of 00 also indicates that the instruction is a sub-packet instruction. A sub-packet instruction is an instruction that includes two or more sub-instructions.

- [0026] For example, the instruction 208 includes the parse field 216 with a value of 00. Therefore, the instruction 208 is a sub-packet instruction and is also an end-of-packet instruction. Looking at the detailed diagram of the instruction 208 (lower portion of FIG. 2), a first sub-instruction 218 and a second sub-instruction 220 are included within the instruction 208. The first sub-instruction 218 includes data labeled A, and the second sub-instruction 220 includes data labeled B.
- [0027] In addition to sub-instructions 218 and 220, the instruction 208 includes a first instruction class field 222 and a second instruction class field 224. The instruction class fields 222 and 224 can indicate a type of instruction, such as load, store, arithmetic, etc. The contents of instruction class fields 222 and 224 can be useful at a decode stage of a pipelined processor.
- [0028] In a particular illustrative embodiment, the instruction 208 includes 32 bits, each sub-instruction 218 and 220 includes 13 bits, the parse field 216 includes 2 bits, the first instruction class field 222 includes 3 bits, and the second instruction class field 224 includes 1 bit.
- [0029] In a particular illustrative embodiment, the sub-instructions 218 and 220 are executed concurrently at an execute stage of a pipelined processor. In another particular illustrative embodiment, the sub-instructions 218 and 220 are executed serially at an execute stage of a pipelined processor.
- [0030] Referring to FIG. 3, a particular illustrative embodiment of a pipelined processor is disclosed and is generally designated 300. The pipelined processor 300 includes a plurality of stages. FIG. 3 shows three stages: a fetch stage 302, a decode stage 304, and an execute stage 306. In alternative embodiments, the pipelined processor 300 may include other stages such as a write-back stage. In a particular illustrative example, an instruction packet 308 is fetched at the fetch stage 302. The instruction packet 308 is decoded at the decode stage 304, after which a plurality of instructions included in the instruction packet 308 is sent to the execute stage 306, where each of the instructions is executed.

- [0031] In a particular illustrative example, the instruction packet 308 includes the instructions 202, 204, 206, and 208. In a particular illustrative embodiment, the instruction packet 308 is retrieved from a tightly coupled memory (not shown) associated with the processor 300. In another particular illustrative embodiment, the instruction packet is retrieved from an instruction cache (not shown) associated with the processor 300. In a particular illustrative embodiment, each of the instructions 202, 204, 206, and 208 of the instruction packet 308 has a same size (i.e., includes a same number of bits). In a particular illustrative embodiment, each of the instructions 202, 204, 206, and 208 has the size of 32 bits.
- [0032] In a particular illustrative embodiment, the instruction packet 308 is a very long instruction word (VLIW) packet. One of the instructions in the VLIW packet 308, such as the instruction 208, is a sub-packet instruction that includes a first sub-instruction 218 and a second sub-instruction 220. Each of the instructions 202, 204, 206, and 208 includes a parse field. The instruction 202 includes a parse field 210, which in the example shown in FIG. 3 has a value of 10, signifying that the instruction 202 is not an end of packet instruction and that the instruction 202 is an end of hardware loop instruction. The instructions 204 and 206 have corresponding parse fields 212 and 214, each of which has a value of 01, signifying that the corresponding instruction is not an end-of-packet instruction. The instruction 208 is a sub-packet instruction and includes a parse field 216 with a value of 00, signifying that the instruction 208 is a sub-packet instruction and also signifying that the instruction 208 is an end-of-packet instruction. In a particular illustrative embodiment, the instruction 208 includes two sub-instructions 218 and 220, each of which has a size that is smaller (i.e., includes a smaller number of bits) than the instruction 316.
- [0033] At the decode stage 304, the VLIW instruction packet 308 is decoded into the instructions 202, 204, 206, and the sub-instructions 218 and 220. Each instruction may be evaluated to determine whether the instruction is an executable instruction or a subpacket instruction that includes multiple sub-instructions. For example, the contents of the parse field 220 may be compared to a defined value that indicates a subpacket instruction. Instructions that are not indicated as subpacket instructions may be immediately routed to the execute stage. Instructions that are determined to be

subpacket instructions may be decoded into their component sub-instructions (e.g. subpacket instructions 218 and 220). In some embodiments the sub-instructions are then routed to the execution stage 306 in parallel. Sub-instructions may be executed in parallel in a single execution unit or in multiple execution units. In other embodiments the sub-instructions are routed serially. The execute stage 306 may be configured to determine whether to execute the sub-instructions serially or in parallel based on available processor resources.

[0034] Moving to the execute stage 306, in a particular illustrative example, each of the instructions leaving the decode stage enters a corresponding execution unit. The instruction 202 enters execution unit 330, the instruction 204 enters execution unit 332, the instruction 206 enters execution unit 334, the sub-instruction 218 enters execution unit 336 and the sub-instruction 220 enters execution unit 338. In a particular illustrative embodiment, all of the instructions 202, 204, 206, 218, and 220 are executed concurrently. In the illustrative embodiment of FIG. 3, the execution units 334, 336, and 338 execute data cache-related instructions such as load and store instructions, while the execution units 330 and 332 are arithmetic execution units, e.g., arithmetic logic unit (ALU). After execution, the output of each execution unit proceeds to provide results during a write-back stage (not shown).

[0035] In a particular embodiment, all of the instructions in the VLIW packet may be executed in parallel. For example, all of the instructions may be executed in parallel by the processor 300, or else a fault will occur. In another particular embodiment, a sub-instruction (e.g. 218 or 220) may be executed after the other instructions in the VLIW packet have been executed, when the processor 300 does not have sufficient resources to execute the instructions 202, 204, 206 and the sub-instructions 218 and 220 in parallel.

[0036] In a particular embodiment, a position of an instruction within the VLIW packet determines which execution unit will receive the instruction. For example, in FIG. 3 the instruction 202 is a first instruction of the VLIW packet 308 and is routed to the execution unit 330, the second instruction 204 is routed to the execution unit 332, and the third instruction 206 is routed to the execution unit 334.

- [0037] The VLIW packet may therefore indicate to the processor which instructions to process in parallel and to which execution unit each instruction is to be routed. In such embodiments, the processor may not be permitted to process instructions within a single packet over multiple cycles, or to route an instruction to an execution unit other than the execution unit indicated by the instruction's position in the VLIW packet.
- [0038] Because a total number of instructions packed in a VLIW packet that includes sub-instructions may be greater than the total number of instructions packed in a VLIW packet without sub-instructions, a total number of fetches to memory for program instructions can be reduced, with a consequent increase in computing time efficiency. Since each fetch has an energy cost, a reduction in the total number of fetches can result in reduced power consumption associated with executing a program.
- [0039] Referring to FIG. 4, a particular illustrative embodiment of a pipelined processor is disclosed and generally designated 400. The pipelined processor 400 includes a fetch stage 402, a decode stage 404, and an execute stage 406. Additional stages of the pipeline, including a write-back stage, are not shown. At a fetch-stage 402, a very long instruction word (VLIW) instruction 408 is fetched from memory such as an instruction cache or other memory. The VLIW instruction 408 includes four instructions 202, 204, 206, and 208. The instruction 202 includes a parse field 210 having a value of 10, the instruction 204 includes a parse field 212 having a value of 01, the instruction 206 includes a parse field 214 having a value of 01, and the instruction 208 includes a parse field 216 having a value of 00. The parse fields 210, 212, and 214 have values that indicate that their corresponding instructions are not end-of-packet instructions. The parse field 216 with the value of 00 indicates that the instruction 208 is a sub-packet instruction and that the instruction 208 is an end of packet instruction. The instruction 208 includes a first sub-instruction 218 and a second sub-instruction 220. The first sub-instruction 218 includes data indicated by A and the second sub-instruction 220 includes data indicated by B.
- [0040] Proceeding to the decode stage 404, the VLIW packet 408 is separated into each of the instructions 202, 204, 206, and the two sub-instructions 218 and 220. Proceeding to the execute stage 406, there are four execution units: 430, 432, 434, and 436. In a particular illustrative embodiment, the execution units 430 and 432 execute arithmetic

instructions (designated with a letter X), and the execution units 434 and 432 are data load and store execute units (designated with letter D). In a particular illustrative example, the instruction 202 is executed in the execution unit 430, the instruction 204 is executed in the execution unit 432, the instruction 206 is executed in the execution unit 434, and the first sub-instruction 218 is executed in the execution unit 436. The instructions 202, 204, 206, and the first sub-instruction 218 can be executed concurrently. After the first sub-instruction 218 is executed in the execution unit 436, the second sub-instruction 220 is executed in the execution unit 436 during a subsequent clock cycle. Thus, in the example depicted in FIG. 4, the sub-instructions 218 and 220 are executed serially in the execution unit 436. In another particular illustrative example (not shown in FIG. 4), the execution unit 436 is configured to enable concurrent execution of both sub-instructions 218 and 220. In this configuration instructions 202, 204, 206, 218, and 220 can be executed concurrently.

[0041] In yet another particular illustrative example illustrated in FIG. 5, an indicator field 502 that is not a parse field, i.e., not within 210, 212, 214, or 216, can serve as an indicator that a particular instruction is a sub-packet instruction. For example, the instruction 204 can include the indicator field 502 that when set, indicates that the instruction 204 is a sub-packet instruction that includes a first sub-instruction C 504 and a second sub-instruction D 506. Alternatively, each instruction in the VLIW packet 408 may include a sub-packet indicator field. The sub-packet instruction 204 can be decoded at the decode stage 404 and the sub-instructions 504 and 506 can then be executed. In a particular illustrative example shown in FIG. 5, a processor 500 includes an execution unit 532 configured to execute two sub-instructions concurrently, enabling sub-instructions 504 and 506 to be executed concurrently with the instructions 202, 206, and the sub-instruction 218. In another particular illustrative example (not shown in FIG. 4 or FIG. 5), each of the execution units 432 and 436 is configured to execute two sub-instructions concurrently, enabling the sub-instructions 218 and 220 and the sub-instructions 504 and 506 to be executed concurrently with the instructions 202 and 206.

[0042] Referring to FIG. 6, a particular illustrative embodiment of a pipelined processor is disclosed and generally designated 600. The pipelined processor 600 includes a

control unit 606 and execution units 608, 610, 612, and 614. The control unit 606 includes a decoder 616 and general register files 620.

[0043] In operation, a packet of instructions, such as a VLIW packet including a plurality of instructions, is retrieved and supplied to the control unit via a bus 634. The decoder 616 is operable to decode multiple instructions within a packet of instructions. The VLIW packet is unpacked into individual instructions at the decoder 616. The VLIW packet includes a plurality of instructions that includes an instruction 202, an instruction 204, and a sub-packet instruction that includes a sub-instruction 218 and a sub-instruction 220. In a particular illustrative embodiment, the sub-instructions 218 and 220 are instructions each of which is smaller in size (i.e., include fewer bits) than the sub-packet instruction in which they are packed in the VLIW packet.

[0044] After decoding the VLIW packet at the decoder 616, the decoded instructions are sent to the execute stage. Each of the instructions 202, 204, 218, and 220 enters a corresponding execution unit. As shown in FIG. 6, the instruction 202 is executed at the instruction unit 614, the instruction 204 is executed at the instruction unit 612, the sub-instruction 218 is executed at execution unit 610, and the sub-instruction 220 is executed at the instruction unit 608. Thus, in the example shown in FIG. 6, the instructions 202, 204, 218, and 220 are executed concurrently. After execution, output of each of the execution units 608, 610, 612, and 614 is written to the general register files 620. The output of the execution units 608, 610, 612, and 614 can also be sent to a storage memory via the bus 634.

[0045] Referring to FIG. 7, a particular illustrative embodiment of a method of routing instructions is shown and generally designated 700. A packet of instructions that includes a first instruction with a first parse value and second instruction with a second parse value is received, at 710. For example, in FIG. 3, the instruction 206 has a parse field 214 with a first parse value of 01 and the instruction 208 has a parse field 216 with a second parse value of 00. In a particular illustrative embodiment, the packet of instructions is a VLIW packet. For example, in FIG. 3, the packet 308 is a VLIW packet. The first instruction is routed to a first execution unit, at 720. For example, in FIG. 3, the instruction 206 is routed to the execution unit 334. Proceeding to 730, the second instruction is identified as a sub-packet instruction that includes a first sub-

- 12 -

instruction and a second sub-instruction. For example, in FIG. 3, the instruction 208 is identified as a subpacket instruction by the second parse value 00 in the parse field 216. The instruction 208 includes the first sub-instruction 218 and the second sub-instruction 220. Proceeding to 740, the first sub-instruction is routed to a second instruction unit. For example, in FIG. 3, the first sub-instruction 218 is routed to the instruction unit 336. Proceeding to 750, the second sub-instruction may be routed to a third execution unit. For example, in FIG. 3, the second sub-instruction 220 is routed to the instruction unit 338. Optionally, the first sub-instruction, the second sub-instruction, and the first instruction are executed concurrently, at 760. For example, in FIG. 3, the instruction 206 and the sub-instructions 218 and 220 are executed concurrently. The method ends at 770. In other embodiments, the second sub-instruction may be routed to the second instruction unit to be executed serially with the first sub-instruction.

[0046] Referring to FIG. 8, a particular illustrative embodiment of a method of executing instructions is disclosed and generally designated 800. A packet of instructions that includes a first instruction is received, at 810. For example, in FIG. 3, the packet 308 includes the instruction 208. The first instruction is decoded, at 820. A determination is made as to whether the first instruction is a sub-packet instruction, at 830. When the first instruction is determined to be a first sub-packet instruction that includes a plurality of sub-instructions, the method proceeds to block 840. For example, in FIG. 3, the instruction 208 is a sub-packet instruction that includes the first sub-instruction 218 and the second sub-instruction 220. The first instruction includes a parse indicator identifying the first instruction as a sub-packet instruction and indicating an end-of-packet status of the first instruction. (Any of the instructions within the packet may be identified during decoding, via the value of its parse indicator, as a sub-packet instruction or via an alternative method of identifying a sub-packet instruction.) For example, in FIG. 3, the instruction 208 includes the parse indicator 216 having a value of 00, identifying the instruction 208 as a sub-packet instruction and indicating that the instruction 208 is an end-of-packet instruction. At block 840, each of the plurality of sub-instructions is executed. For example, in FIG. 3, the first sub-instruction 218 and the second sub-instruction 220 are executed. In a particular illustrative embodiment, all of the plurality of sub-instructions are executed concurrently. For example, in FIG. 3, the first sub-instruction 218 and the second sub-

instruction 220 are executed concurrently. Returning to decision block 830, when the first instruction is not a sub-packet instruction, the first instruction is executed, at 850. For example, in FIG. 3, at the decode stage 304 it is determined that the instruction 206 is not a sub-packet instruction, and the instruction 206 proceeds from the decode stage 304 to the execute stage 306, where it is executed. The method ends at 860.

[0047] Referring to FIG. 9, a block diagram of a particular illustrative embodiment of a wireless device including a digital signal processor to process a VLIW packet that includes sub-instructions, is depicted and generally designated 900. The device 900 includes a processor, such as a digital signal processor (DSP) 910 to process a plurality of instructions in a VLIW packet 964. The VLIW packet 964 includes instructions 966, 968, 970, and 972. The VLIW packet 964 is a hierarchical instruction packet that includes a plurality of subpacket instructions. The VLIW packet 964 includes three levels of instructions in the hierarchy. Each subpacket instruction includes a plurality of sub-instructions. For example, the instruction 972 is a subpacket instruction (second level of the hierarchy) that includes sub-instructions 974 and 976. The sub-instruction 974 is a sub-subpacket instruction (third level of the hierarchy) that includes sub-sub-instructions 978 and 980, and the sub-instruction 976 is a sub-subpacket instruction (third level of the hierarchy) that includes sub-sub-instructions 982 and 984.

[0048] At a decode stage of the DSP 910, the VLIW packet 964 is decoded into the instructions 966, 968, 970, and 972. The subpacket instruction 972 is decoded into the sub-instructions 974 and 976. The sub-instruction 974 is decoded into the sub-sub-instructions 978 and 980, and the sub-instruction 976 is decoded into the sub-sub-instructions 982 and 984. In an illustrative example, the DSP 910 processes the instructions 966, 968, 970, 978, 980, 982, and 984 according to one or more of FIGs. 3-5, and according to one or more of the methods of FIGs. 6 and 7, or any combination thereof.

[0049] FIG. 9 also shows a display controller 926 that is coupled to the digital signal processor 910 and to a display 928. A coder/decoder (CODEC) 934 can also be coupled to the digital signal processor 910. A speaker 936 and a microphone 938 can be coupled to the CODEC 934.

[0050] FIG. 9 also indicates that a wireless controller 940 can be coupled to the digital signal processor 910 and to a wireless antenna 942. In a particular embodiment, the DSP 910, the display controller 926, the memory 932, the CODEC 934, and the wireless controller 940 are included in a system-in-package or system-on-chip device 922. In a particular embodiment, an input device 930 and a power supply 944 are coupled to the system-on-chip device 922. Moreover, in a particular embodiment, as illustrated in FIG. 9, the display 928, the input device 930, the speaker 936, the microphone 938, the wireless antenna 942, and the power supply 944 are external to the system-on-chip device 922. However, each of the display 928, the input device 930, the speaker 936, the microphone 938, the wireless antenna 942, and the power supply 944 can be coupled to a component of the system-on-chip device 922, such as an interface or a controller.

[0051] Those of skill would further appreciate that the various illustrative logical blocks, configurations, modules, circuits, and algorithm steps described in connection with the embodiments disclosed herein may be implemented as electronic hardware, computer software, or combinations of both. To clearly illustrate this interchangeability of hardware and software, various illustrative components, blocks, configurations, modules, circuits, and steps have been described above generally in terms of their functionality. Whether such functionality is implemented as hardware or software depends upon the particular application and design constraints imposed on the overall system. Skilled artisans may implement the described functionality in varying ways for each particular application, but such implementation decisions should not be interpreted as causing a departure from the scope of the present disclosure.

[0052] The steps of a method or algorithm described in connection with the embodiments disclosed herein may be embodied directly in hardware, in a software module executed by a processor, or in a combination of the two. A software module may reside in random access memory (RAM), flash memory, read-only memory (ROM), programmable read-only memory (PROM), erasable programmable read-only memory (EPROM), electrically erasable programmable read-only memory (EEPROM), registers, hard disk, a removable disk, a compact disc read-only memory (CD-ROM), or any other form of storage medium known in the art. An exemplary storage medium is coupled to the processor such that the processor can read information from, and write

- 15 -

information to, the storage medium. In the alternative, the storage medium may be integral to the processor. The processor and the storage medium may reside in an application-specific integrated circuit (ASIC). The ASIC may reside in a computing device or a user terminal. In the alternative, the processor and the storage medium may reside as discrete components in a computing device or user terminal.

[0053] The previous description of the disclosed embodiments is provided to enable any person skilled in the art to make or use the disclosed embodiments. Various modifications to these embodiments will be readily apparent to those skilled in the art, and the principles defined herein may be applied to other embodiments without departing from the scope of the disclosure. Thus, the present disclosure is not intended to be limited to the embodiments shown herein but is to be accorded the widest scope possible consistent with the principles and novel features as defined by the following claims.

WHAT IS CLAIMED IS:

1. A method of processing instructions, the method comprising:
receiving a packet of instructions, wherein the packet includes an instruction;
decoding the instruction to determine whether the instruction is a single
instruction or whether the instruction includes a subpacket that includes a
plurality of sub-instructions; and
in response to determining that the instruction includes the subpacket, executing
each of the sub-instructions.
2. The method of claim 1, wherein the packet of instructions includes multiple
subpackets.
3. The method of claim 1, further comprising decoding a first subpacket of the
plurality of sub-instructions to identify a first sub-subpacket of instructions.
4. The method of claim 1, wherein the decoding is performed by a decoder that
is operable to decode multiple instructions in the packet of instructions.
5. The method of claim 1, wherein the instruction includes a parse indicator that
identifies the instruction as the subpacket.
6. The method of claim 1, wherein the instruction includes an instruction class
field that is used to identify a type of instruction.
7. The method of claim 1, wherein the packet is a very long instruction word
(VLIW) packet.
8. The method of claim 1, wherein the packet includes a plurality of instructions.
9. The method of claim 1, wherein each of the plurality of sub-instructions is
executed at a first execution unit.

- 17 -

10. The method of claim 1, wherein a first of the plurality of sub-instructions is executed at a first execution unit and a second of the plurality of sub-instructions is executed at a second execution unit.

11. The method of claim 10, wherein the first of the sub-instructions and the second of the sub-instructions are executed concurrently.

12. The method of claim 1, wherein each instruction of the packet comprises a very long instruction word (VLIW) instruction.

13. A method of processing instructions, the method comprising:
receiving a packet of instructions, wherein the packet includes a first instruction having a first parse value and a second instruction having a second parse value;
routing the first instruction to a first execution unit;
identifying the second instruction as a sub-packet instruction that includes a first sub-instruction and a second sub-instruction; and
routing the first sub-instruction to a second execution unit.

14. The method of claim 13, wherein each parse value identifies the corresponding instruction to be at an end of the packet, at an end of a hardware loop, or not at an end of the packet.

15. The method of claim 13, wherein the second parse value identifies the second instruction as the sub-packet instruction.

16. The method of claim 13, wherein each of the execution units includes an arithmetic execution unit or a load/store data unit.

17. The method of claim 13, further comprising executing the first sub-instruction and the second sub-instruction concurrently.

- 18 -

18. The method of claim 13, further comprising routing the second sub-instruction to a third execution unit.

19. A processor comprising:

a decoder to receive a packet of instructions, wherein the packet includes an instruction, the decoder operative to decode the instruction to determine whether the instruction is a first sub-packet instruction that includes a first sub-instruction and a second sub-instruction; and
an execution unit to execute the instruction or to execute the first sub-instruction.

20. The processor of claim 19, further comprising a second execution unit to execute the second sub-instruction.

21. The processor of claim 19, wherein the packet further comprises a second instruction, the processor further comprising a third execution unit to execute the second instruction.

22. The processor of claim 19, wherein the packet includes a plurality of instructions, wherein each instruction of the plurality of instructions has a same size.

23. The processor of claim 22, wherein a size associated with each of the first sub-instruction and the second sub-instruction is smaller than the size associated with the instruction.

24. The processor of claim 19, wherein the instruction includes a first parse indicator having a first parse value that is associated with an end of packet status, the first parse value also indicating that the first instruction includes the first sub-instruction and the second sub-instruction.

25. The processor of claim 19, wherein a size associated with each of the first sub-instruction and the second sub-instruction is smaller than a size associated with the instruction.

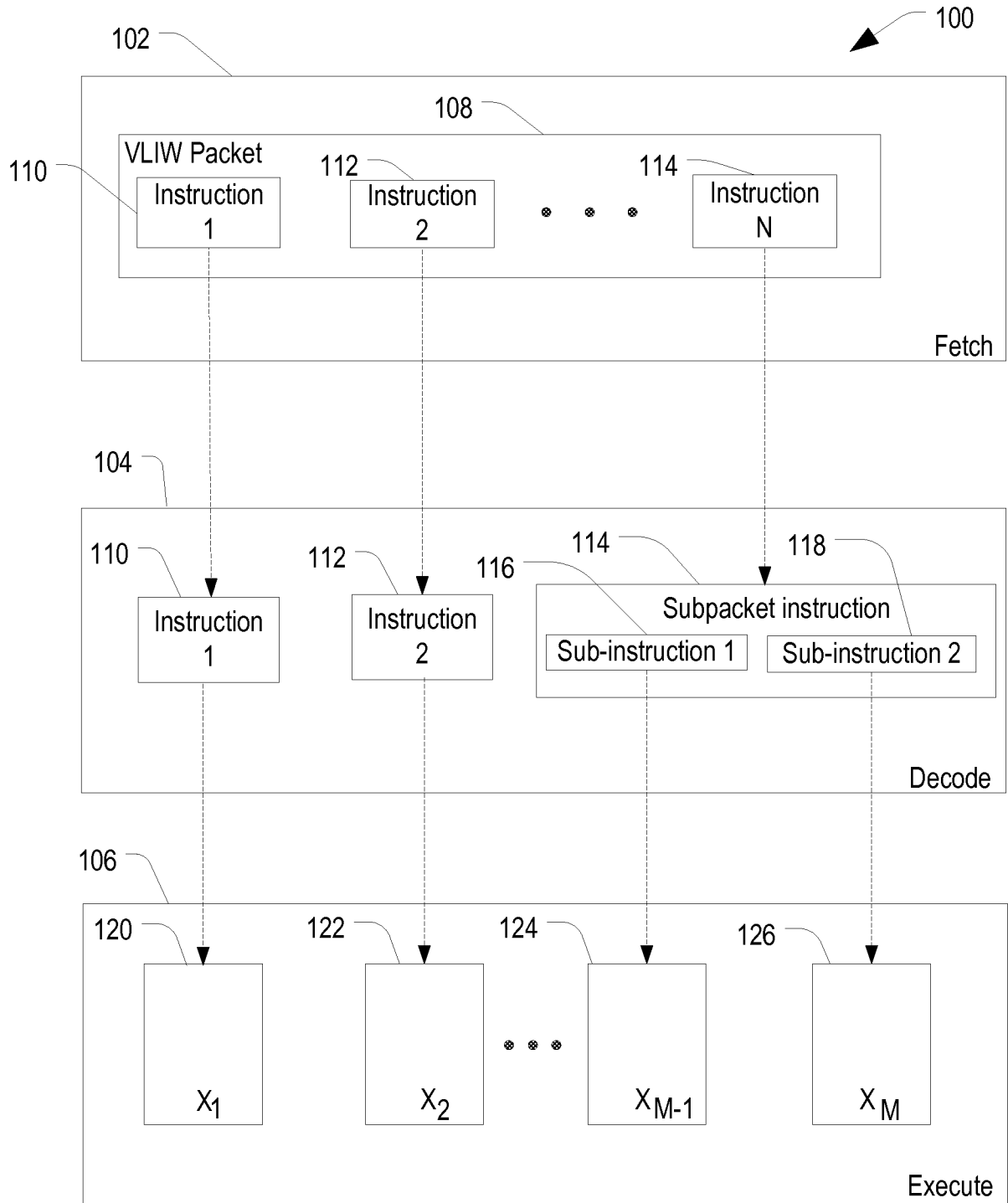


FIG. 1

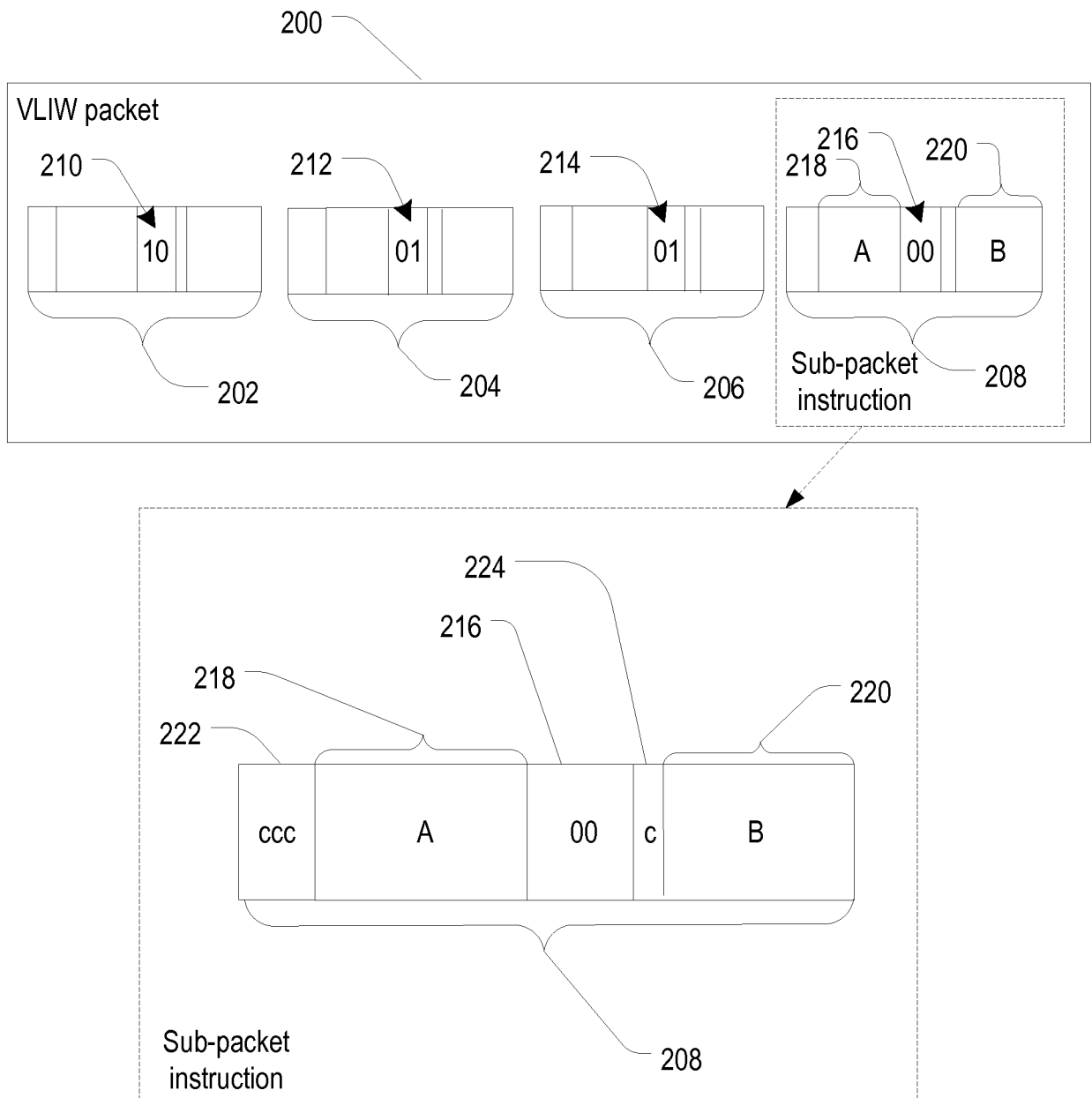


FIG. 2

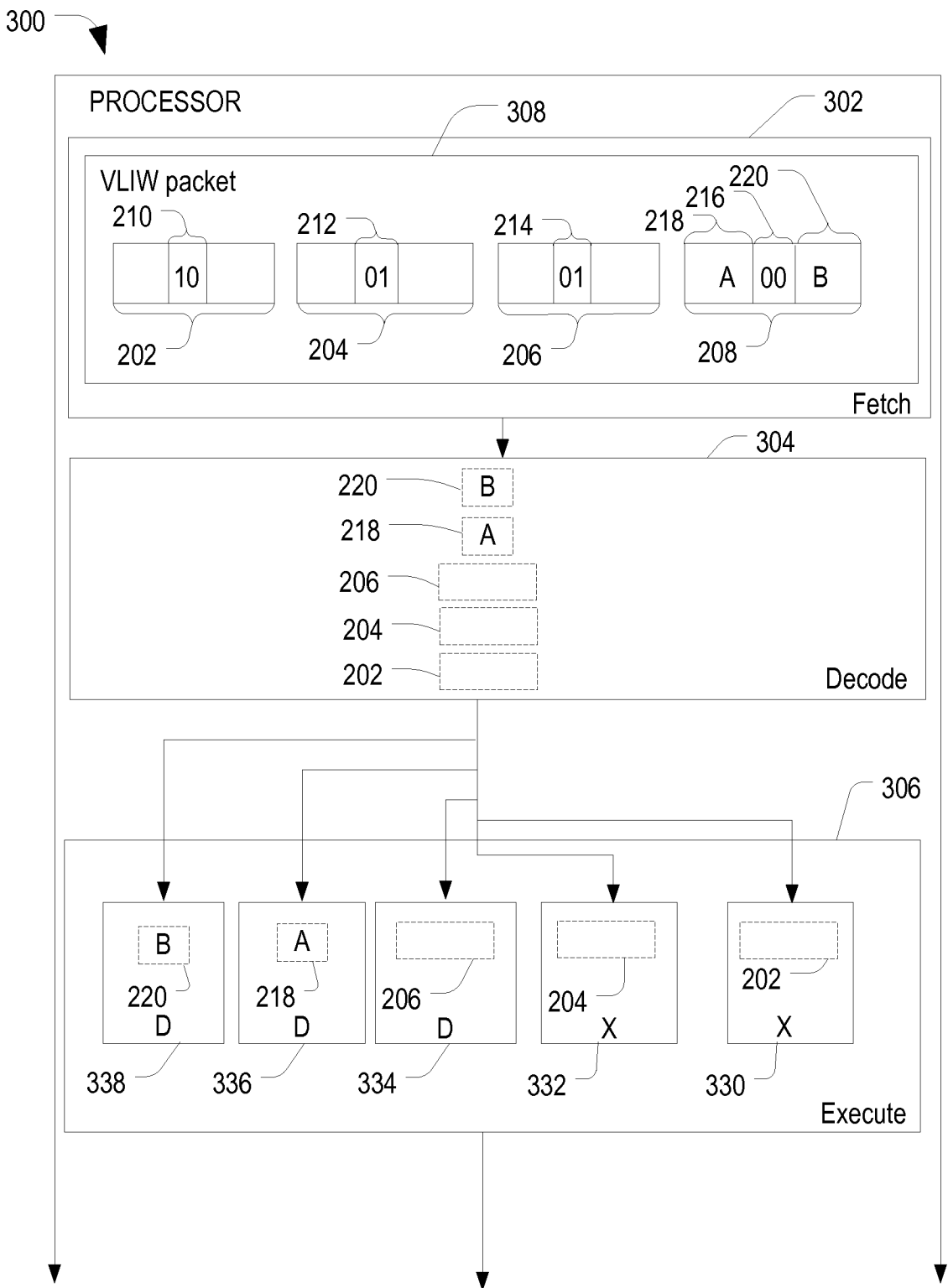


FIG. 3

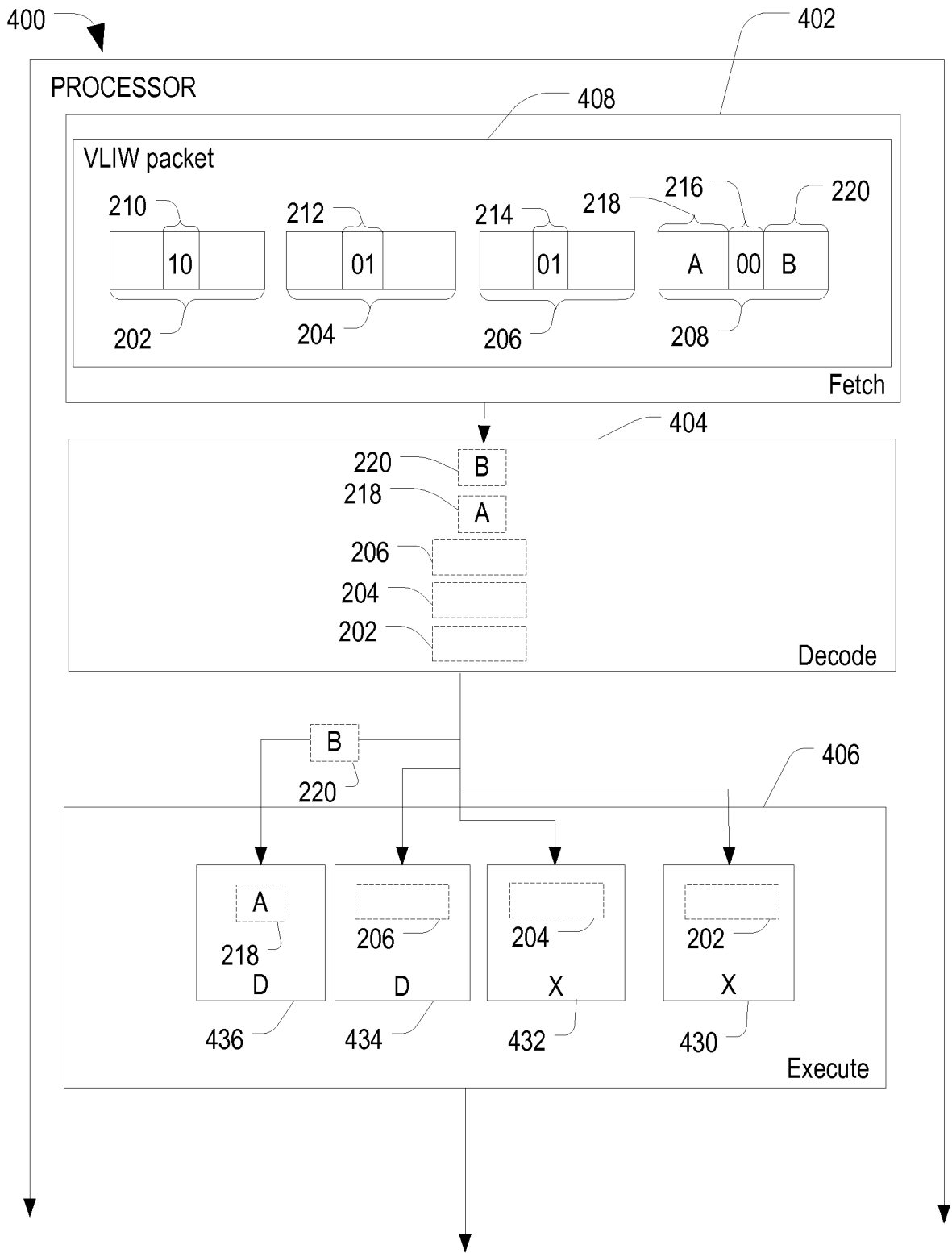


FIG. 4

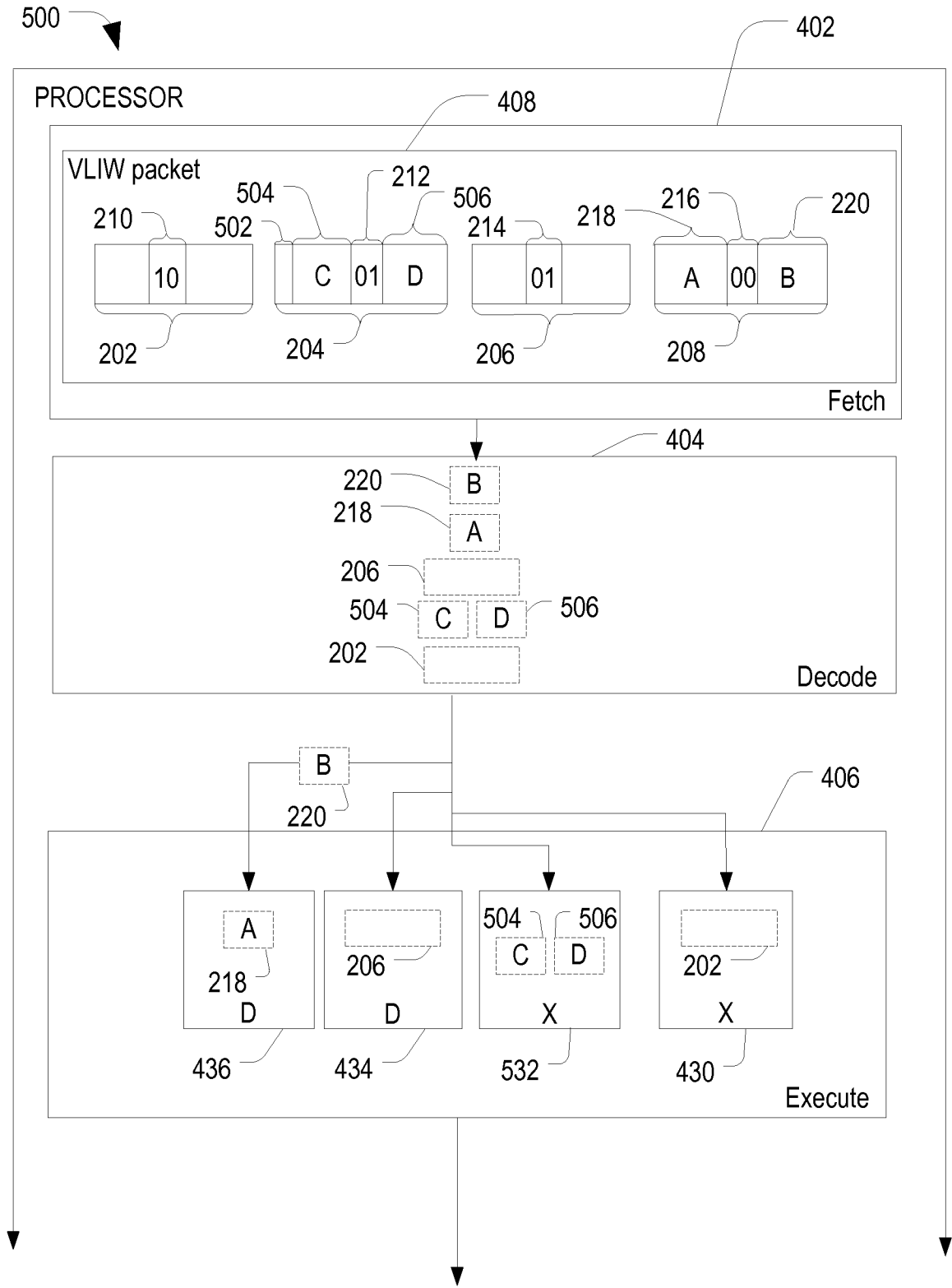


FIG. 5

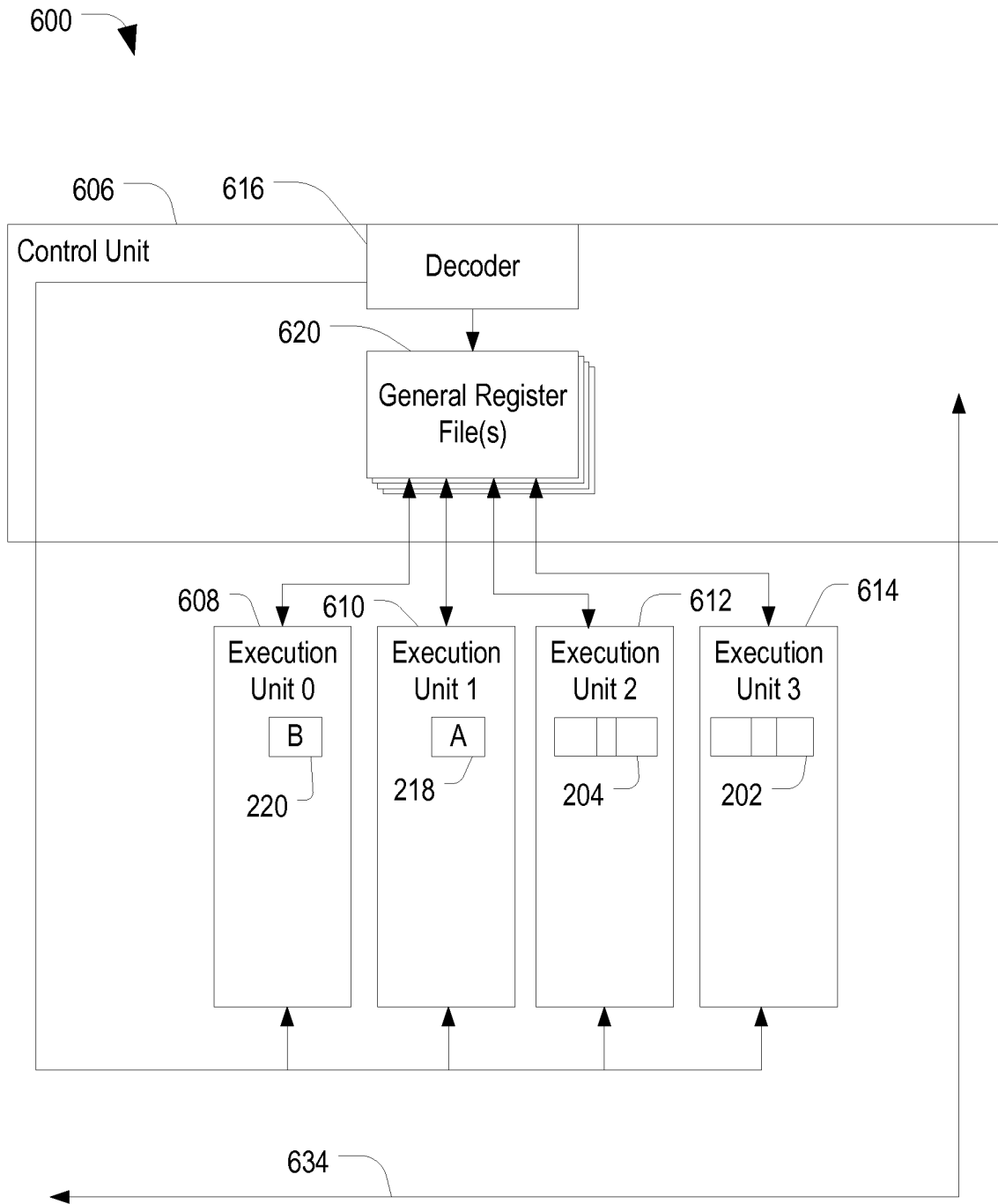


FIG. 6

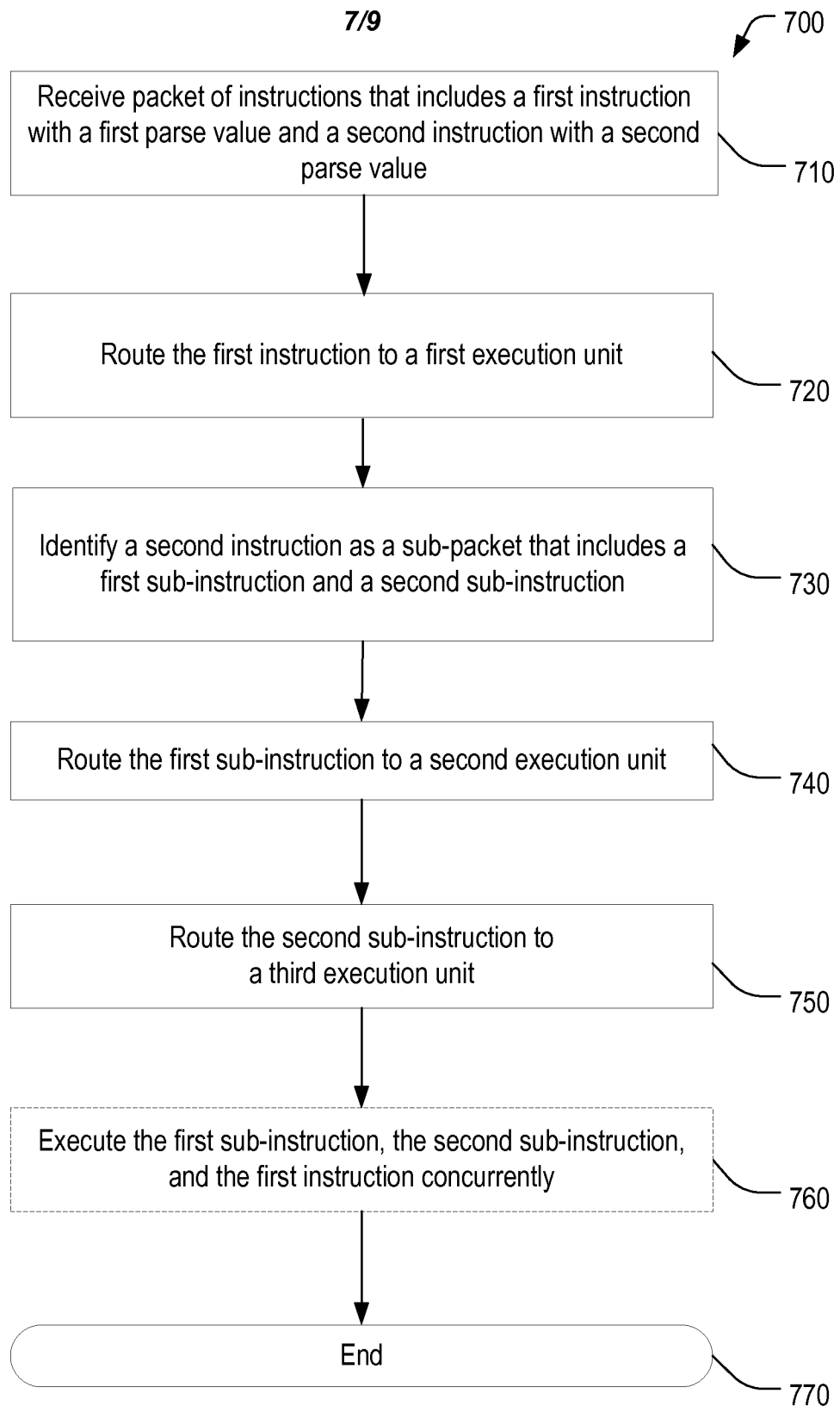


FIG. 7

8/9

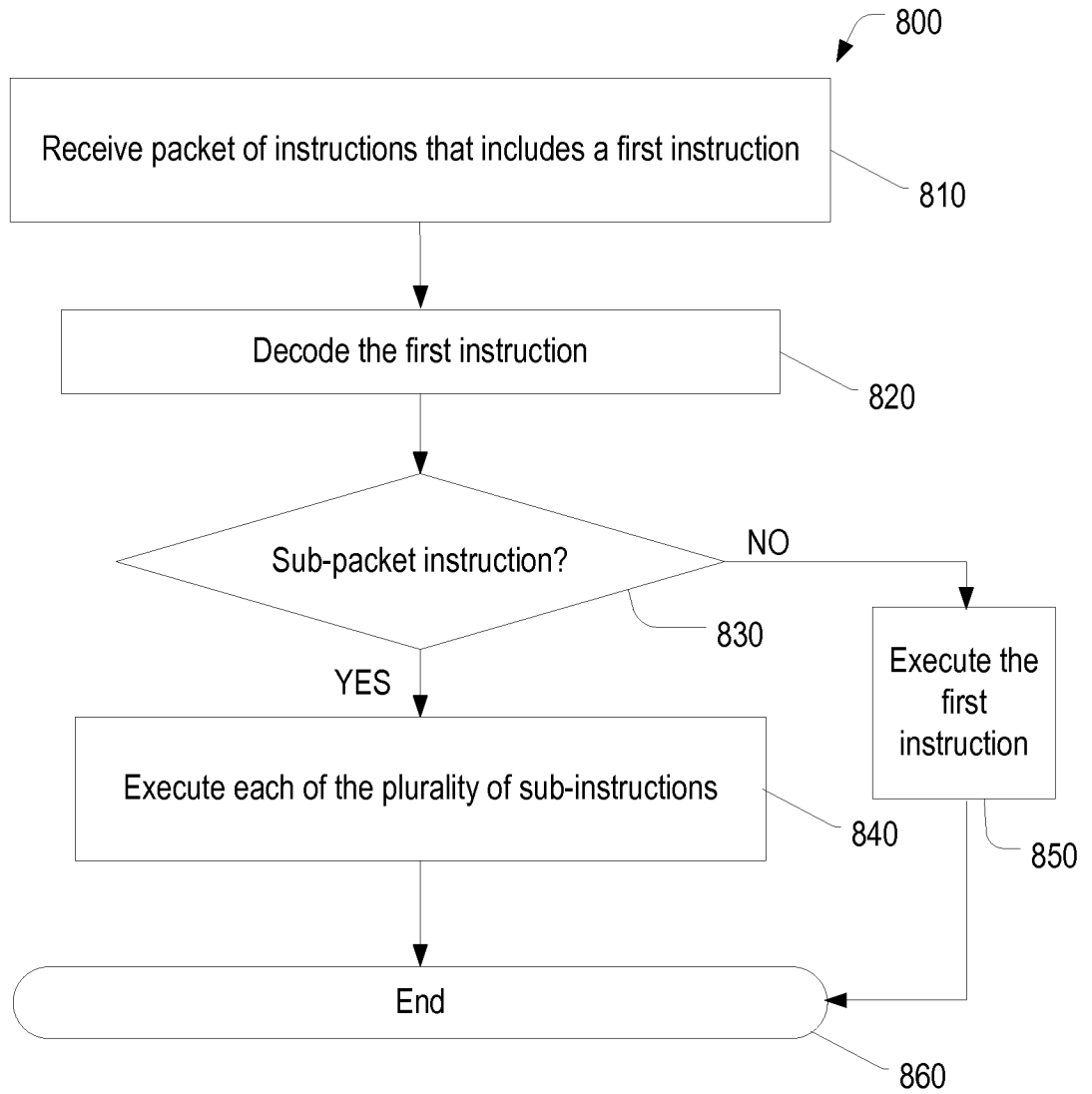


FIG. 8

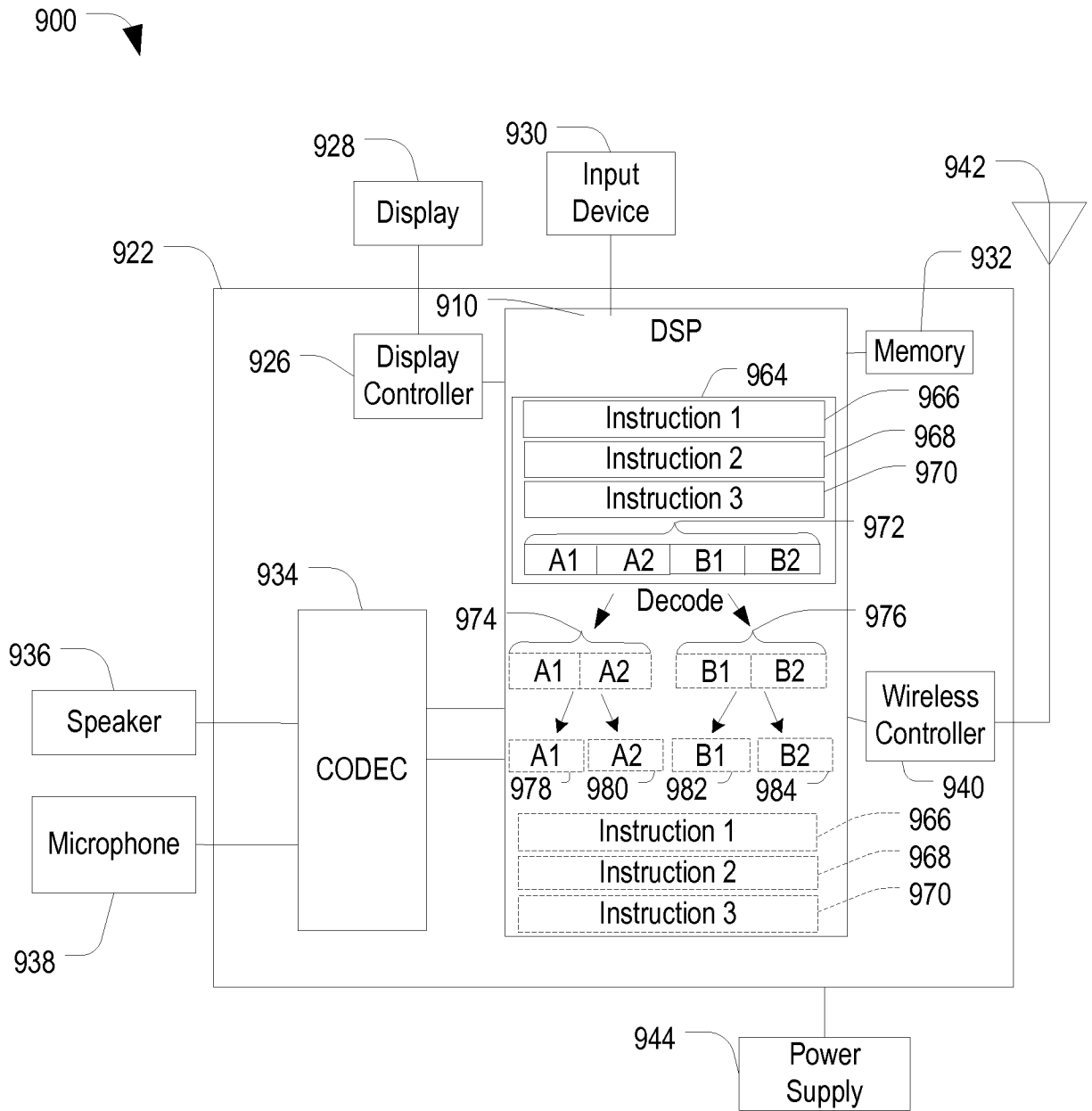


FIG. 9

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2011/026815

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F9/30 G06F9/38
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

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2006/259740 A1 (HAHN TODD T [US] ET AL) 16 November 2006 (2006-11-16) paragraph [0053] - paragraph [0062]; figures 10,11,12	1-25
X	----- WO 00/22515 A1 (CONEXANT SYSTEMS INC [US]) 20 April 2000 (2000-04-20) page 8, line 18 - page 11, line 15; figure 7	1-25
X	----- WO 2006/102664 A2 (QUALCOMM INC [US]; CODRESCU LUCIAN [US]; PLONDKE ERICH [US]; AHMED MUH) 28 September 2006 (2006-09-28)	1-13, 15-25
Y	paragraph [0028] - paragraph [0041]; figures 4,5	14
	----- -/--	

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 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 15 April 2011	Date of mailing of the international search report 26/04/2011
--	--

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

INTERNATIONAL SEARCH REPORT

International application No

PCT/US2011/026815

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	WO 2007/133893 A1 (QUALCOMM INC [US]; PLONDKE ERICH [US]; LESTER ROBERT ALLAN [US]; CODRE) 22 November 2007 (2007-11-22) the whole document -----	14

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No PCT/US2011/026815

Patent document cited in search report		Publication date		Patent family member(s)	Publication date
US 2006259740	A1	16-11-2006		NONE	

WO 0022515	A1	20-04-2000	US	6366998 B1	02-04-2002

WO 2006102664	A2	28-09-2006	CN	101176059 A	07-05-2008
			US	2006218379 A1	28-09-2006

WO 2007133893	A1	22-11-2007	CN	101438235 A	20-05-2009
			EP	2027532 A1	25-02-2009
			JP	2009536769 T	15-10-2009
			KR	20090009966 A	23-01-2009
			US	2007266229 A1	15-11-2007
