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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
30 August 2018 (30.08.2018)

W!P O | PCT

(10) International Publication Number
WO 2018/156635 Al

(51) International Patent Classification:
G06F 9/30 (2018.01) *G06F 9/44* (20 18.0 1)
G06F 7/38 (2006.01)

(21) International Application Number:
PCT/US20 18/0 19026

(22) International Filing Date:
2 1 February 2018 (21 .02.2018)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

15/439,540	22 February 2017 (22.02.2017)	US
17159105.0	03 March 2017 (03.03.2017)	EP

(71) Applicant: **ADVANCED MICRO DEVICES, INC** [US/US]; 2485 Augustine Drive, Santa Clara, California 95054 (US).

(72) Inventors: **MANTOR, Michael J.;** 5912 N. Dean Road, Orlando, Florida 32817 (US). **EMBERLING, Brian D.;** 4021 Villa Vis., Palo Alto, California 94306 (US). **FOWLER, Mark;** 19 Nicholas Road, Hopkinton, Massachusetts 01748 (US).

(74) Agent: **RANKIN, Rory D.;** MEYERTONS, HOOD, KIVLIN, KOWERT & GOETZEL, P.C., P.O. Box 398, Austin, Texas 78767-0398 (US).

(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, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, 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, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, 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, KM, ML, MR, NE, SN, TD, TG).

(54) Title: VARIABLE WAVEFRONT SIZE

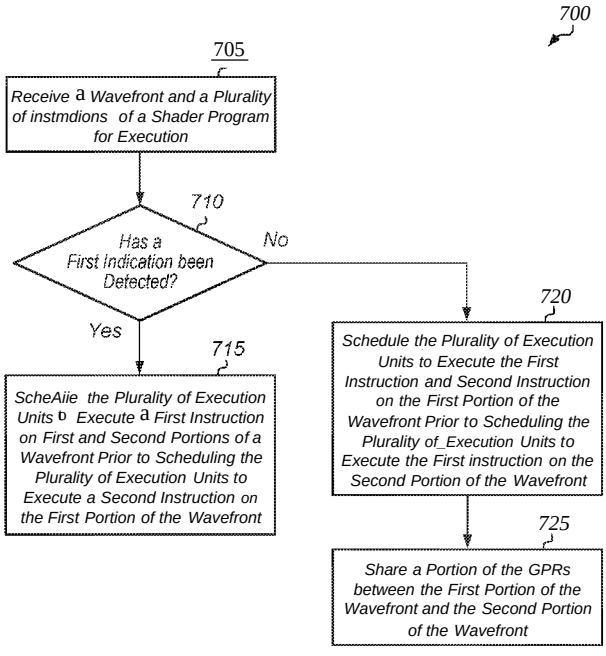

FIG. 7

(57) Abstract: Systems, apparatuses, and methods for processing variable wavefront sizes on a processor are disclosed. In one embodiment, a processor includes at least a scheduler, cache, and multiple execution units. When operating in a first mode, the processor executes the same instruction on multiple portions of a wavefront before proceeding to the next instruction of the shader program. When operating in a second mode, the processor executes a set of instructions on a first portion of a wavefront. In the second mode, when the processor finishes executing the set of instructions on the first portion of the wavefront, the processor executes the set of instructions on a second portion of the wavefront, and so on until all portions of the wavefront have been processed. The processor determines the operating mode based on one or more conditions.

WO 2018/156635 A1

[Continued on next page]

Published:
— *with international search report (Art. 21(3))*

VARIABLE WAVEFRONT SIZE

BACKGROUND

Description of the Related Art

[0001] A graphics processing unit (GPU) is a complex integrated circuit that is configured to perform graphics-processing tasks. For example, a GPU can execute graphics-processing tasks required by an end-user application, such as a video-game application. GPUs are also increasingly being used to perform other tasks which are unrelated to graphics. The GPU can be a discrete device or can be included in the same device as another processor, such as a central processing unit (CPU).

[0002] In many applications, such as graphics processing in a GPU, a sequence of work-items, which can also be referred to as threads, are processed so as to output a final result. In many modern parallel processors, for example, processors within a single instruction multiple data (SIMD) core synchronously execute a set of work-items. A plurality of identical synchronous work-items that are processed by separate processors are referred to as a wavefront or warp.

[0003] During processing, one or more SIMD cores concurrently execute multiple wavefronts. Execution of the wavefront terminates when all work-items within the wavefront complete processing. Each wavefront includes multiple work-items that are processed in parallel, using the same set of instructions. In some cases, the number of work-items in a wavefront does not match the number of execution units of the SFMD cores. In one embodiment, each execution unit of a SFMD core is an arithmetic logic unit (ALU). When the number of work-items in a wavefront does not match the number of execution units of the SIMD cores, determining how to schedule instructions for execution can be challenging.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] The advantages of the methods and mechanisms described herein may be better understood by referring to the following description in conjunction with the accompanying drawings, in which:

[0005] FIG. 1 is a block diagram of one embodiment of a computing system.

[0006] FIG. 2 is a block diagram of one embodiment of a GPU.

[0007] FIG. 3 is a block diagram of one embodiment of a set of vector general purpose registers (VGPRs).

[0008] FIG. 4 illustrates one embodiment of an example wavefront and an example instruction sequence.

[0009] FIG. 5 illustrates a diagram of one embodiment of a first mode of operation for a processor.

[0010] FIG. 6 illustrates a diagram of one embodiment of a second mode of operation for a processor.

[0011] FIG. 7 is a generalized flow diagram illustrating one embodiment of a method for scheduling instructions on a processor.

[0012] FIG. 8 is a generalized flow diagram illustrating one embodiment of a method for determining which operating mode to use in a parallel processor.

[0013] FIG. 9 is a generalized flow diagram illustrating one embodiment of a method for utilizing different operating modes for a parallel processor.

DETAILED DESCRIPTION OF EMBODIMENTS

[0014] In the following description, numerous specific details are set forth to provide a thorough understanding of the methods and mechanisms presented herein. However, one having ordinary skill in the art should recognize that the various embodiments may be practiced without these specific details. In some instances, well-known structures, components, signals, computer program instructions, and techniques have not been shown in detail to avoid obscuring the approaches described herein. It will be appreciated that for simplicity and clarity of illustration, elements shown in the figures have not necessarily been drawn to scale. For example, the dimensions of some of the elements may be exaggerated relative to other elements.

[0015] Various systems, apparatuses, methods, and computer-readable mediums for processing variable wavefront sizes on a processor are disclosed. When operating in a first mode, the processor executes the same instruction on multiple portions of a

wavefront before proceeding to the next instruction of the shader program. When operating in a second mode, the processor executes a set of instructions on a first portion of a wavefront. In the second mode, when the processor finishes executing the set of instructions on the first portion of the wavefront, the processor executes the set of instructions on a second portion of the wavefront, and so on until all portions of the wavefront have been processed. Then, the processor continues executing subsequent instructions of the shader program.

[0016] In one embodiment, an indication is declared within the code sequence, with the indication specifying which mode to utilize for a given region of the program. In another embodiment, a compiler generates the indication when generating executable code, with the indication specifying the processor operating mode. In another embodiment, the processor includes a control unit which determines the processor operating mode.

[0017] Referring now to FIG. 1, a block diagram of one embodiment of a computing system 100 is shown. In one embodiment, computing system 100 includes system on chip (SoC) 105 coupled to memory 150. SoC 105 can also be referred to as an integrated circuit (IC). In one embodiment, SoC 105 includes processing units 115A-N, input/output (I/O) interfaces 110, shared caches 120A-B, fabric 125, graphics processing unit (GPU) 130, and memory controller(s) 140. SoC 105 can also include other components not shown in FIG. 1 to avoid obscuring the figure. Processing units 115A-N are representative of any number and type of processing units. In one embodiment, processing units 115A-N are central processing unit (CPU) cores. In another embodiment, one or more of processing units 115A-N are other types of processing units (e.g., application specific integrated circuit (ASIC), field programmable gate array (FPGA), digital signal processor (DSP)). Processing units 115A-N are coupled to shared caches 120A-B and fabric 125.

[0018] In one embodiment, processing units 115A-N are configured to execute instructions of a particular instruction set architecture (ISA). Each processing unit 115A-N includes one or more execution units, cache memories, schedulers, branch prediction circuits, and so forth. In one embodiment, the processing units 115A-N are configured to execute the main control software of system 100, such as an operating system.

Generally, software executed by processing units 115A-N during use can control the other components of system 100 to realize the desired functionality of system 100. Processing units 115A-N can also execute other software, such as application programs.

[0019] GPU 130 includes at least compute units 145A-N which are representative of any number and type of compute units that are used for graphics or general-purpose processing. Compute units 145A-N can also be referred to as "shader arrays", "shader engines", "single instruction multiple data (SIMD) units", or "SIMD cores". Each compute unit 145A-N includes a plurality of execution units. GPU 130 is coupled to shared caches 120A-B and fabric 125. In one embodiment, GPU 130 is configured to execute graphics pipeline operations such as draw commands, pixel operations, geometric computations, and other operations for rendering an image to a display. In another embodiment, GPU 130 is configured to execute operations unrelated to graphics. In a further embodiment, GPU 130 is configured to execute both graphics operations and non-graphics related operations.

[0020] GPU 130 is configured to receive instructions of a shader program and wavefronts for execution. In one embodiment, GPU 130 is configured to operate in different modes. In one embodiment, the number of work-items in each wavefront is greater than the number of execution units in GPU 130.

[0021] In one embodiment, GPU 130 schedules a first instruction for execution on first and second portions of a first wavefront prior to scheduling a second instruction for execution on the first portion of the first wavefront responsive to detecting a first indication. GPU 130 follows this pattern for the other instructions of the shader program and for other wavefronts as long as the first indication is detected. It is noted that "scheduling an instruction" can also be referred to as "issuing an instruction". Depending on the embodiment, the first indication can be specified in software, or the first indication can be generated by GPU 130 based on one or more operating conditions. In one embodiment, the first indication is a command for GPU 130 to operate in a first mode.

[0022] In one embodiment, GPU 130 schedules the first instruction and the second instruction for execution on the first portion of the first wavefront prior to scheduling the first instruction for execution on the second portion of the first wavefront responsive to not detecting the first indication. GPU 130 follows this pattern for the other instructions

of the shader program and for other wavefronts as long as the first indication is not detected.

[0023] I/O interfaces 110 are coupled to fabric 125, and I/O interfaces 110 are representative of any number and type of interfaces (e.g., peripheral component interconnect (PCI) bus, PCI-Extended (PCI-X), PCIE (PCI Express) bus, gigabit Ethernet (GBE) bus, universal serial bus (USB)). Various types of peripheral devices can be coupled to I/O interfaces 110. Such peripheral devices include (but are not limited to) displays, keyboards, mice, printers, scanners, joysticks or other types of game controllers, media recording devices, external storage devices, network interface cards, and so forth.

[0024] SoC 105 is coupled to memory 150, which includes one or more memory modules. Each of the memory modules includes one or more memory devices mounted thereon. In some embodiments, memory 150 includes one or more memory devices mounted on a motherboard or other carrier upon which SoC 105 is also mounted. The RAM implemented can be static RAM (SRAM), dynamic RAM (DRAM), Resistive RAM (ReRAM), Phase Change RAM (PCRAM), or any other volatile or non-volatile RAM. The type of DRAM that is used to implement memory 150 includes (but is not limited to) double data rate (DDR) DRAM, DDR2 DRAM, DDR3 DRAM, and so forth. Although not explicitly shown in FIG. 1, SoC 105 can also include one or more cache memories that are internal to the processing units 115A-N and/or compute units 145A-N. In some embodiments, SoC 105 includes shared caches 120A-B that are utilized by processing units 115A-N and compute units 145A-N. In one embodiment, caches 120A-B are part of a cache subsystem including a cache controller.

[0025] In various embodiments, computing system 100 can be a computer, laptop, mobile device, server or any of various other types of computing systems or devices. It is noted that the number of components of computing system 100 and/or SoC 105 can vary from embodiment to embodiment. There can be more or fewer of each component/subcomponent than the number shown in FIG. 1. For example, in another embodiment, SoC 105 can include multiple memory controllers coupled to multiple memories. It is also noted that computing system 100 and/or SoC 105 can include other components not shown in FIG. 1. Additionally, in other embodiments, computing system 100 and SoC 105 can be structured in other ways than shown in FIG. 1.

[0026] Turning now to FIG. 2, a block diagram of one embodiment of a graphics processing unit (GPU) 200 is shown. In one embodiment, GPU 200 includes at least SIMDs 210A-N, branch and message unit 240, scheduler unit 245, instruction buffer 255, and cache 260. It is noted that GPU 200 can also include other logic which is not shown in FIG. 2 to avoid obscuring the figure. It is also noted that other processors (e.g., FPGAs, ASICs, DSPs) can include the circuitry shown in GPU 200.

[0027] In one embodiment, GPU 200 is configured to operate in different modes to process instructions of a shader program on wavefronts of different sizes. GPU 200 utilizes a given mode to optimize performance, power consumption, and/or other factors depending on the type of workload being processed and/or the number of work-items in each wavefront. In one embodiment, each wavefront includes a number of work-items which is greater than the number of lanes 215A-N, 220A-N, and 225A-N in SIMDs 210A-N. In this embodiment, GPU 200 processes the wavefronts differently based on the operating mode of GPU 200. In another embodiment, GPU 200 processes the wavefronts differently based on one or more detected conditions. Each lane 215A-N, 220A-N, and 225A-N of SEVIDs 210A-N can also be referred to as an "execution unit".

[0028] In one embodiment, GPU 200 receives a plurality of instructions for a wavefront with a number of work-items which is greater than the total number of lanes in SEVIDs 210A-N. In this embodiment, GPU 200 executes a first instruction on multiple portions of a wavefront before proceeding to the second instruction when GPU 200 is in a first mode. GPU 200 continues with this pattern of execution for subsequent instructions for as long as GPU 200 is in the first mode. In one embodiment, the first mode can be specified by a software-generated declaration. If GPU 200 is in a second mode, then GPU 200 executes multiple instructions on a first portion of the wavefront before proceeding to the second portion of the wavefront. When GPU 200 is in the second mode, GPU 200 shares a portion of vector general purpose registers (VGPRs) 230A-N between different portions of the wavefront. Additionally, when GPU 200 is in the second mode, if an execution mask of mask(s) 250 indicates that a given portion of the wavefront is temporarily masked out, then GPU 200 does not execute the instruction for the given portion of the wavefront.

[0029] In another embodiment, if the wavefront size is greater than the number of SIMDs, GPU 200 determines the cache miss rate of cache 260 for the program. If the cache miss rate is less than a threshold, then GPU 200 executes a first instruction on multiple portions of a wavefront before proceeding to the second instruction. GPU 200 continues with this pattern of execution for subsequent instructions for as long as the cache miss rate is determined or predicted to be less than the threshold. The threshold can be specified as a number of bytes, as a percentage of cache 260, or as any other suitable metric. If the cache miss rate is greater than or equal to the threshold, then GPU 200 executes multiple instructions on a first portion of the wavefront before executing the multiple instructions on the second portion of the wavefront. Additionally, if the cache miss rate is greater than or equal to the threshold, GPU 200 shares a portion of vector general purpose registers (VGPRs) 230A-N between different portions of the wavefront, and GPU 200 skips instructions for the given portion of the wavefront if the execution mask indicates the given portion is masked out.

[0030] It is noted that the letter "N" when displayed herein next to various structures is meant to generically indicate any number of elements for that structure (e.g., any number of SIMDs 210A-N). Additionally, different references within FIG. 2 that use the letter "N" (e.g., SIMDs 210A-N and lanes 215A-N) are not intended to indicate that equal numbers of the different elements are provided (e.g., the number of SFMDs 210A-N can differ from the number of lanes 215A-N).

[0031] Referring now to FIG. 3, a block diagram of one embodiment of a set of vector general purpose registers (VGPRs) 300 is shown. In one embodiment, VGPRs 300 are included in SIMDs 210A-N of GPU 200 (of FIG. 2). VGPRs 300 can include any number of registers, depending on the embodiment.

[0032] As shown in FIG. 3, VGPRs 300 includes VGPRs 305 for a first wavefront, VGPRs 310 for a second wavefront, and any number of other VGPRs for other numbers of wavefronts. It is assumed for the purposes of this discussion that the first and second wavefronts have 2*N number of work-items, with N being a positive integer, and with N varying from embodiment to embodiment. In one embodiment, N is equal to 32. VGPRs 305 includes a region with private VGPRs and a shared VGPR region 315. Similarly, VGPRs 310 includes a region with private VGPRs and a shared VGPR region 320.

[0033] In one embodiment, if the host GPU (e.g., GPU 200) is in a first mode, then shared VGPRs 315 and shared VGPRs 310 are not shared between different portions of the first and second wavefronts, respectively. However, when the host GPU is in a second mode, then shared VGPRs 315 and shared VGPRs 310 are shared between different portions of the first and second wavefronts, respectively. In other embodiments, the implementation of sharing or not sharing is based on the detection of a first indication rather than being based on a first mode or second mode. The first indication can be generated by software, generated based on cache miss rate, or generated based on one or more other operating conditions.

[0034] Turning now to FIG. 4, one embodiment of an example wavefront 405 and an example instruction sequence 410 are shown. Wavefront 405 is intended to illustrate one example of a wavefront in accordance with one embodiment. Wavefront 405 includes 2*N number of work-items, wherein "N" is a positive integer, and wherein "N" is the number of lanes 425A-N in vector unit 420. Vector unit 420 can also be referred to as a SIMD unit or a parallel processor. The first portion of wavefront 405 includes work-items W_0 through W_{N-1}, and the second portion of wavefront 405 includes work-items W_N through W_{2N-1}. A single portion of wavefront 405 is intended to execute on lanes 425A-N of vector unit 420 in a given instruction cycle. In other embodiments, wavefront 405 can include other numbers of portions.

[0035] In one embodiment, N is 32, and the number of work-items per wavefront is 64. In other embodiments, N can be other values. In the embodiment when N is 32, vector unit 420 also includes 32 lanes which are shown as lanes 425A-N. In other embodiments, vector unit 420 can include other numbers of lanes.

[0036] Instruction sequence 410 is illustrative of one example of an instruction sequence. As shown in FIG. 4, instruction sequence 410 includes instructions 415A-D, which are representative of any number and type of instructions of a shader program. It should be assumed for the purposes of this discussion that instruction 415A is the first instruction of instruction sequence 410, with instruction 415B the second instruction, instruction 415C the third instruction, and instruction 415D the fourth instruction. In other embodiments, an instruction sequence can have other numbers of instructions.

Wavefront 405, instruction sequence 410, and vector unit 420 are reused during the discussion that continues for FIG. 5 and FIG. 6 below.

[0037] Referring now to FIG. 5, a diagram of one embodiment of a first mode of operation for a processor is shown. The discussion of FIG. 5 is a continuation of the discussion regarding FIG. 4. In one embodiment, the size of a wavefront is twice the size of vector unit 420. In another embodiment, the size of a wavefront is an integer multiple of the size of vector unit 420. In these embodiments, a processor can implement different modes of operation for determining how to execute the work-items of the wavefront using vector unit 420.

[0038] In a first mode of operation, each instruction is executed on different subsets of the wavefront before the next instruction is executed on the different subsets. For example, instruction 415A is executed for the first half (i.e., work-items W_0 through W_{N-1}) of the wavefront during a first instruction cycle on lanes 425A-N of vector unit 420, and then instruction 415A is executed for the second half (i.e., work-items W_N through W_{2N-1}) of the first wavefront during a second instruction cycle on lanes 425A-N. For example, during the first instruction cycle, work-item W_0 can execute on lane 425A, work-item W_1 can execute on lane 425B, and so on.

[0039] Then, instruction 415B is executed for the first half of the wavefront during a third instruction cycle on lanes 425A-N, and then instruction 415B is executed for the second half of the wavefront during a fourth instruction cycle on lanes 425A-N. Next, instruction 415C is executed for the first half of the wavefront during a fifth instruction cycle on lanes 425A-N, and then instruction 415C is executed for the second half of the wavefront during a sixth instruction cycle on lanes 425A-N. Then, instruction 415D is executed for the first half of the wavefront on lanes 425A-N of vector unit 420 during a seventh instruction cycle on lanes 425A-N, and then instruction 415D is executed for the second half of the wavefront on lanes 425A-N of vector unit 420 during an eighth instruction cycle on lanes 425A-N. For the purposes of this discussion, it can be assumed that the second instruction cycle follows the first instruction cycle, the third instruction cycle follows the second instruction cycle, and so on.

[0040] Turning now to FIG. 6, a diagram of one embodiment of a second mode of operation for a processor is shown. The discussion of FIG. 6 a continuation of the

discussion regarding FIG. 5. In the second mode of operation, the entire instruction sequence 410 is executed on the same portion of the wavefront before the entire instruction sequence 410 is executed on the next portion of the wavefront.

[0041] For example, in a first instruction cycle, instruction 415A is executed for the first half (i.e., work-items W_0 through W_{N-1}) of the wavefront on lanes 425A-N of vector unit 420. Then, in a second instruction cycle, instruction 415B is executed for the first half of the wavefront on lanes 425A-N. Next, in a third instruction cycle, instruction 415C is executed for the first half of the wavefront on lanes 425A-N. Then, in a fourth instruction cycle, instruction 415D is executed for the first half of the wavefront on lanes 425A-N.

[0042] Next, instruction sequence 410 is executed on the second half of the wavefront. Accordingly, in a fifth instruction cycle, instruction 415A is executed for the second half (i.e., work-items W_N through W_{2N-1}) of the wavefront on lanes 425A-N of vector unit 420. Then, in a sixth instruction cycle, instruction 415B is executed for the second half of the wavefront on lanes 425A-N. Next, in a seventh instruction cycle, instruction 415C is executed for the second half of the wavefront on lanes 425A-N. Then, in an eighth instruction cycle, instruction 415D is executed for the second half of the wavefront on lanes 425A-N.

[0043] In another embodiment, if a wavefront had 4*N work-items, instruction sequence 410 could be executed on the first quarter of the wavefront, then instruction sequence 410 could be executed on the second quarter of the wavefront, followed by the third quarter and then the fourth quarter of the wavefront. Other wavefronts of other sizes and/or vector units with other numbers of lanes could be utilized in a similar manner for the second mode of operation.

[0044] Referring now to FIG. 7, one embodiment of a method 700 for scheduling instructions on a processor is shown. For purposes of discussion, the steps in this embodiment and those of FIG. 8-9 are shown in sequential order. However, it is noted that in various embodiments of the described methods, one or more of the elements described are performed concurrently, in a different order than shown, or are omitted entirely. Other additional elements are also performed as desired. Any of the various

systems, apparatuses, or computing devices described herein are configured to implement method 700.

[0045] A processor receives a wavefront and a plurality of instructions of a shader program for execution (block 705). In one embodiment, the processor includes at least a plurality of execution units, a scheduler, a cache, and a plurality of GPRs. In one embodiment, the processor is a GPU. In other embodiments, the processor is any of various other types of processors ((e.g., DSP, FPGA, ASIC, multi-core processor). In one embodiment, the number of work-items in the wavefront is greater than the number of execution units of the processor. For example, in one embodiment, the wavefront includes 64 work-items and the processor includes 32 execution units. In this embodiment, the number of work-items in the wavefront is equal to twice the number of execution units. In other embodiments, the wavefront can include other numbers of work-items and/or the processor can include other numbers of execution units. In some cases, the processor receives a plurality of wavefronts for execution. In these cases, method 700 can be implemented multiple times for the multiple wavefronts.

[0046] Next, the processor determines if a first indication has been detected (conditional block 710). In one embodiment, the first indication is a setting or parameter declared within a software instruction, with the setting or parameter specifying the operating mode for the processor to utilize. In another embodiment, the first indication is generated based on a cache miss rate of the wavefront. In other embodiments, other types of indications are possible and are contemplated.

[0047] If the first indication is detected (conditional block 710, "yes" leg), then the processor schedules the plurality of execution units to execute a first instruction on first and second portions of a wavefront prior to scheduling the plurality of execution units to execute a second instruction on the first portion of the wavefront (block 715). The processor can follow this same pattern of scheduling instruction for the remainder of the plurality of instructions, as long as the first indication is detected. If the first indication is not detected (conditional block 710, "no" leg), then the processor schedules the plurality of execution units to execute the first instruction and the second instruction on the first portion of the wavefront prior to scheduling the plurality of execution units to execute the first instruction on the second portion of the wavefront (block 720). The processor can

follow this same pattern of scheduling instruction for the remainder of the plurality of instructions, as long as the first indication is not detected. Also, the processor shares a portion of the GPRs between the first portion of the wavefront and the second portion of the wavefront if the first indication is not detected (block 725). After blocks 715 and 725, method 700 ends.

[0048] Turning now to FIG. 8, one embodiment of a method 800 for determining which operating mode to use in a parallel processor is shown. A control unit of a processor determines a cache miss rate of a wavefront (block 805). Depending on the embodiment, the control unit can determines the cache miss rate of a portion of the wavefront or of the entirety of the wavefront in block 805. Depending on the embodiment, the control unit is implemented using any suitable combination of hardware and/or software. In one embodiment, the control unit predicts a cache miss rate of the wavefront. In another embodiment, the control unit receives an indication generated by software, with the indication specifying the cache miss rate of the wavefront. Next, the control unit determines if the cache miss rate of the wavefront is less than a threshold (conditional block 810). Alternatively, if the control unit receives an indication generated by software, the indication can specify if the cache miss rate is less than the threshold. In one embodiment, the threshold is programmable. In another embodiment, the threshold is predetermined.

[0049] If the cache miss rate of the wavefront is less than a threshold (conditional block 810, "yes" leg), then the processor utilizes a first mode of operation when processing the wavefront (block 815). In one embodiment, the first mode of operation involves issuing each instruction on all portions of the wavefront before moving on to the next instruction in the shader program. If the cache miss rate of the wavefront is greater than or equal to the threshold (conditional block 810, "no" leg), then the processor utilizes a second mode of operation when processing the wavefront (block 820). In one embodiment, the second mode of operation involves executing a set of instructions on a first portion of the wavefront, then executing the same set of instructions on a second portion of the wavefront, and so on, until all portions of the wavefront have been processed. After blocks 815 and 820, method 800 ends.

[0050] Referring now to FIG. 9, another embodiment of a method 900 for utilizing different operating modes for a parallel processor is shown. A control unit of a processor determines the operating mode of the processor (block 905). Depending on the embodiment, the control unit is implemented using any suitable combination of hardware and/or software. The criteria the control unit utilizes to determine which operating mode to select can vary from embodiment to embodiment. One example of criteria that can be utilized is described in FIG. 8 in the discussion regarding method 800. Other examples of criteria that can be utilized for selecting the processor operating mode are possible and are contemplated.

[0051] If the control unit selects a first operating mode (conditional block 910, "first" leg), then the processor does not share registers between different subsets of the wavefront being processed by the processor (block 915). Otherwise, if the control unit selects a second operating mode (conditional block 910, "second" leg), then the control unit shares one or more registers between different subsets of the wavefront being processed by the processor (block 920). For example, in one embodiment, sharing registers involves the processor using a shared portion of a register file for a first portion of a wavefront for a first set of instructions. Then, the processor reuses the shared portion of the register file for a second portion of the wavefront. If the wavefront has more than two portions, then the processor reuses the shared portion of the register file for the additional portions of the wavefront. After block 920, method 900 ends.

[0052] It is noted that in some embodiments, a processor can have more than two operating modes. In these embodiments, conditional block 910 can be applied such that a first subset (e.g., first mode, third mode, seventh mode) of operating modes follow the "first" leg and a second subset (e.g., second mode, fourth mode, fifth mode, sixth mode) of operating modes follow the "second" leg shown in FIG. 7. Alternatively, in another embodiment, the size of the portion of the register file that is shared can vary according to different operating modes. For example, for a second mode, a first number of registers are shared, for a third mode, a second number of registers are shared, for a fourth mode, a third number of GPRs are shared, and so on.

[0053] In various embodiments, program instructions of a software application are used to implement the methods and/or mechanisms previously described. The program

instructions describe the behavior of hardware in a high-level programming language, such as C. Alternatively, a hardware design language (HDL) is used, such as Verilog. The program instructions are stored on a non-transitory computer readable storage medium. Numerous types of storage media are available. The storage medium is accessible by a computing system during use to provide the program instructions and accompanying data to the computing system for program execution. The computing system includes at least one or more memories and one or more processors configured to execute program instructions.

It should be emphasized that the above-described embodiments are only non-limiting examples of implementations. Numerous variations and modifications will become apparent to those skilled in the art once the above disclosure is fully appreciated. It is intended that the following claims be interpreted to embrace all such variations and modifications.

WHAT IS CLAIMED IS

1. A processor comprising:
    a plurality of execution units; and
    a scheduler;
    wherein the scheduler is configured to:
        schedule the plurality of execution units to execute a first instruction on first and second portions of a wavefront prior to scheduling the plurality of execution units to execute a second instruction on the first portion of the wavefront, responsive to detecting a first indication; and
        schedule the plurality of execution units to execute the first instruction and the second instruction on the first portion of a wavefront prior to scheduling the plurality of execution units to execute the first instruction on the second portion of the wavefront, responsive to not detecting the first indication.

2. The processor as recited in claim 1, wherein the first indication is a parameter declared within a software instruction.

3. The processor as recited in claim 1, wherein the processor is configured to operate in a plurality of operating modes, and wherein the first indication is a command for the processor to operate in a first mode.

4. The processor as recited in claim 3, wherein the processor further comprises a plurality of general purpose registers (GPRs), and wherein the processor is configured to share one or more GPRs between the first portion of the wavefront and the second portion of the wavefront responsive to operating in a second mode.

5. The processor as recited in claim 1, wherein the processor further comprises a cache and is configured to:
    determine a cache miss rate of the wavefront; and

generate the first indication responsive to determining the cache miss rate of the wavefront is less than a threshold.

6. The processor as recited in claim 1, wherein a number of work-items in the wavefront is greater than a number of the plurality of execution units.

7. The processor as recited in claim 1, wherein the number of work-items in the wavefront is 64, and wherein the number of the plurality of execution units is 32.

8. A method for use in a computing device, the method comprising:

scheduling a plurality of execution units to execute a first instruction on first and second portions of a wavefront prior to scheduling the plurality of execution units to execute a second instruction on the first portion of the wavefront responsive to detecting a first indication; and

scheduling the plurality of execution units to execute the first instruction and the second instruction on the first portion of a wavefront prior to scheduling the plurality of execution units to execute the first instruction on the second portion of the wavefront responsive to not detecting the first indication.

9. The method as recited in claim 8, wherein the first indication is a parameter declared within a software instruction.

10. The method as recited in claim 8, wherein the first indication is a command for a processor to operate in a first mode.

11. The method as recited in claim 10, further comprising sharing one or more general purpose registers (GPRs) between the first portion of the wavefront and the second portion of the wavefront responsive to operating in a second mode.

12. The method as recited in claim 8, further comprising:

determining a cache miss rate of the wavefront; and

generating the first indication responsive to determining the cache miss rate of the wavefront is less than a threshold.

13. The method as recited in claim 8, wherein a number of work-items in the wavefront is greater than a number of the plurality of execution units.

14. The method as recited in claim 8, wherein a number of work-items in the wavefront is 64, and wherein a number of the plurality of execution units is 32.

15. A system comprising:

a memory; and

a processor comprising:

a plurality of execution units; and

a scheduler;

wherein the scheduler is configured to:

schedule the plurality of execution units to execute a first instruction on first and second portions of a wavefront prior to scheduling the plurality of execution units to execute a second instruction on the first portion of the wavefront responsive to detecting a first indication; and

schedule the plurality of execution units to execute the first instruction and the second instruction on the first portion of a wavefront prior to scheduling the plurality of execution units to execute the first instruction on the second portion of the wavefront responsive to not detecting the first indication.

16. The system as recited in claim 15, wherein first indication is a parameter declared within a software instruction.

17. The system as recited in claim 15, wherein the processor is configured to operate in a plurality of operating modes, and wherein the first indication is a command for the processor to operate in a first mode.

18. The system as recited in claim 17, wherein the processor further comprises a plurality of general purpose registers (GPRs), and wherein the processor is configured to share one or more GPRs between the first portion of the wavefront and the second portion of the wavefront responsive to operating in a second mode.

19. The system as recited in claim 15, wherein the processor further comprises a cache, wherein the processor is configured to:

determine a cache miss rate of the wavefront; and

generate the first indication responsive to determining the cache miss rate of the wavefront is less than a threshold.

20. The system as recited in claim 16, wherein a number of work-items in the wavefront is greater than a number of the plurality of execution units.

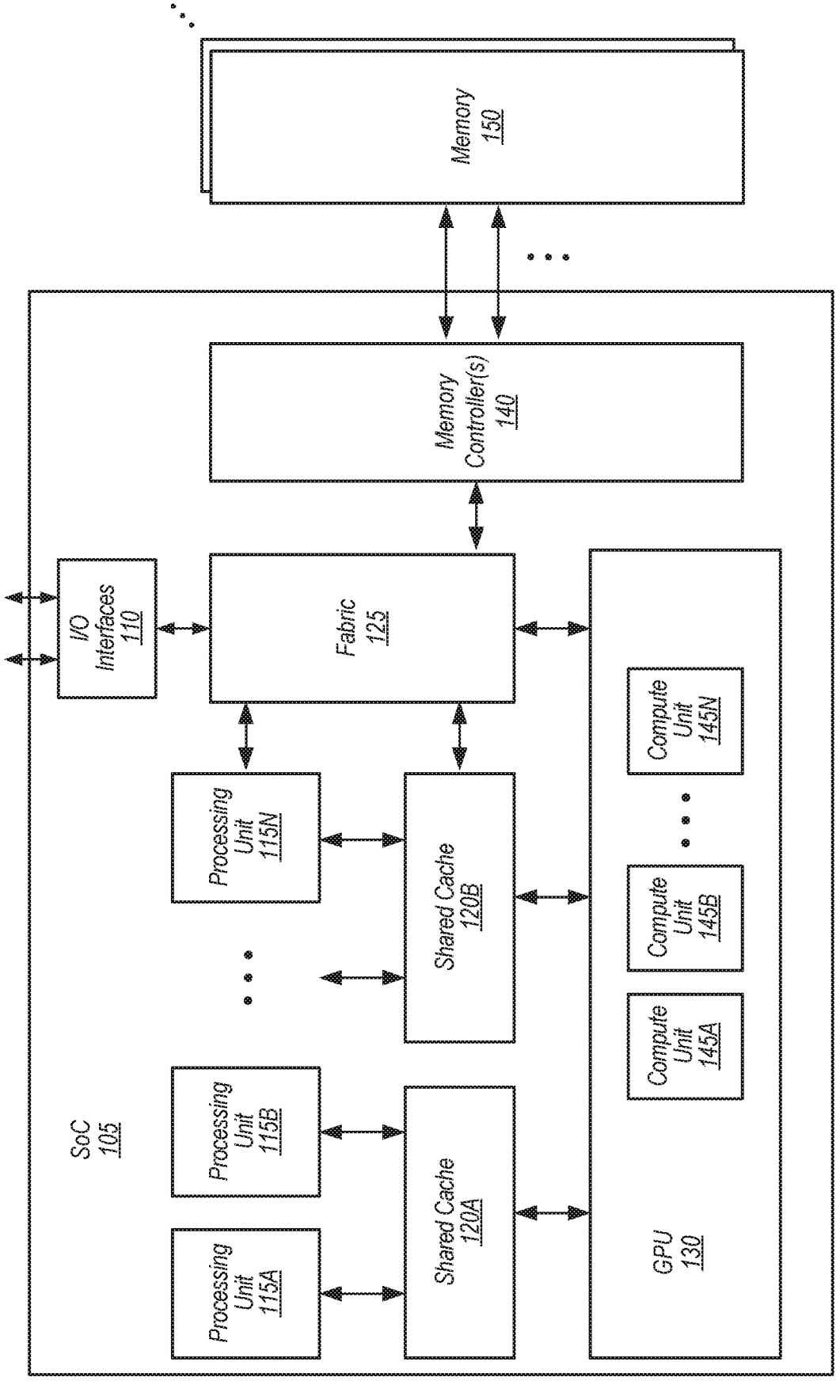
Memory 150
Memory Controller(s) 140
I/O Interfaces 110
Fabric 125
Processing Unit 115N
Shared Cache 120B
Compute Unit 145N
Compute Unit 145B
Compute Unit 145A
Processing Unit 115B
Shared Cache 120A
Processing Unit 115A
GPU 130
SoC 105
100

FIG. 1

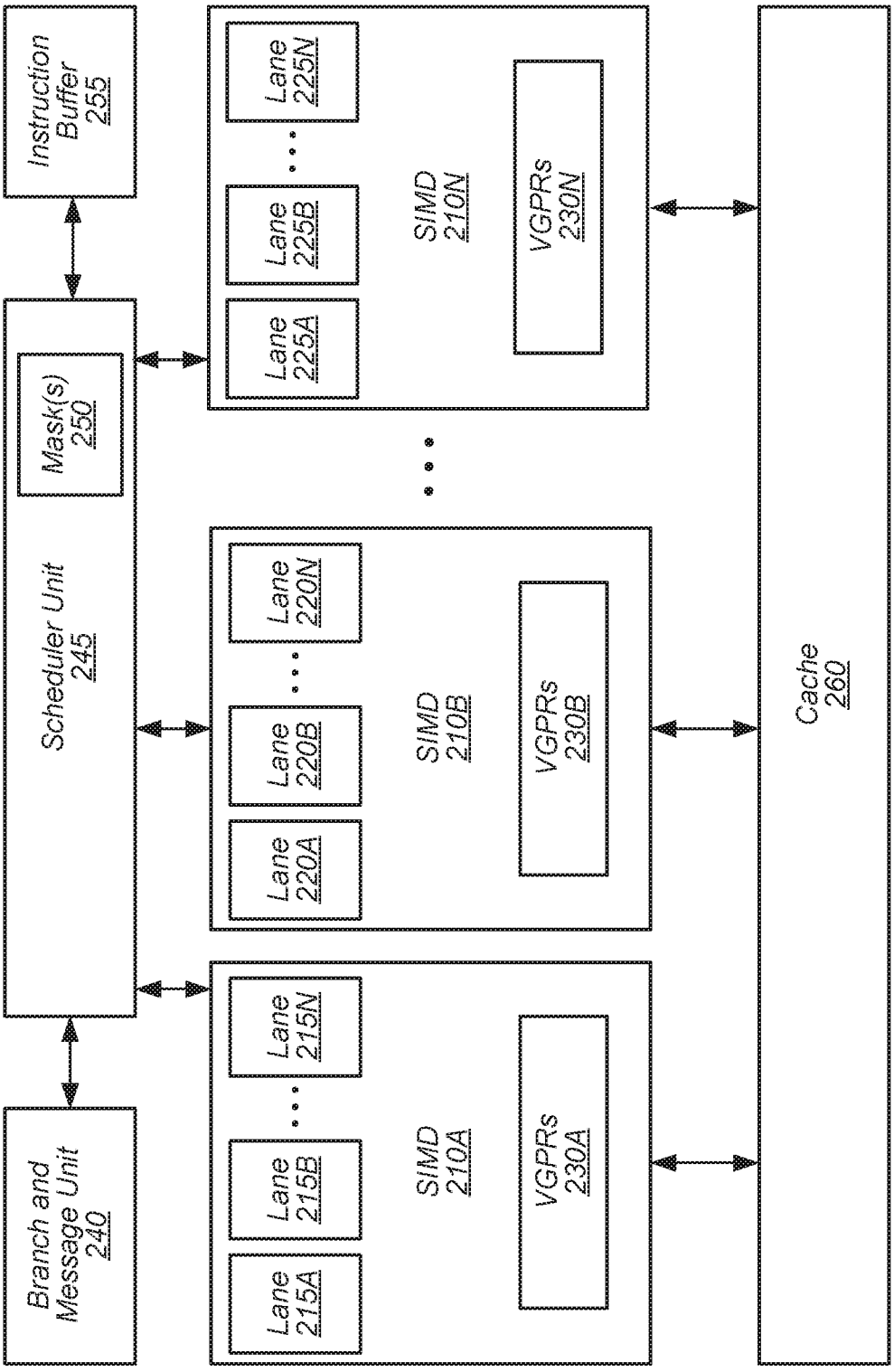
Instruction Buffer 255
Mask(s) 250
Scheduler Unit 245
Branch and Message Unit 240
Lane 215A
Lane 215B
Lane 215N
SIMD 210A
VGPRs 230A
Lane 220A
Lane 220B
Lane 220N
SIMD 210B
VGPRs 230B
Lane 225A
Lane 225B
Lane 225N
SIMD 210N
VGPRs 230N
Cache 260
200

FIG. 2

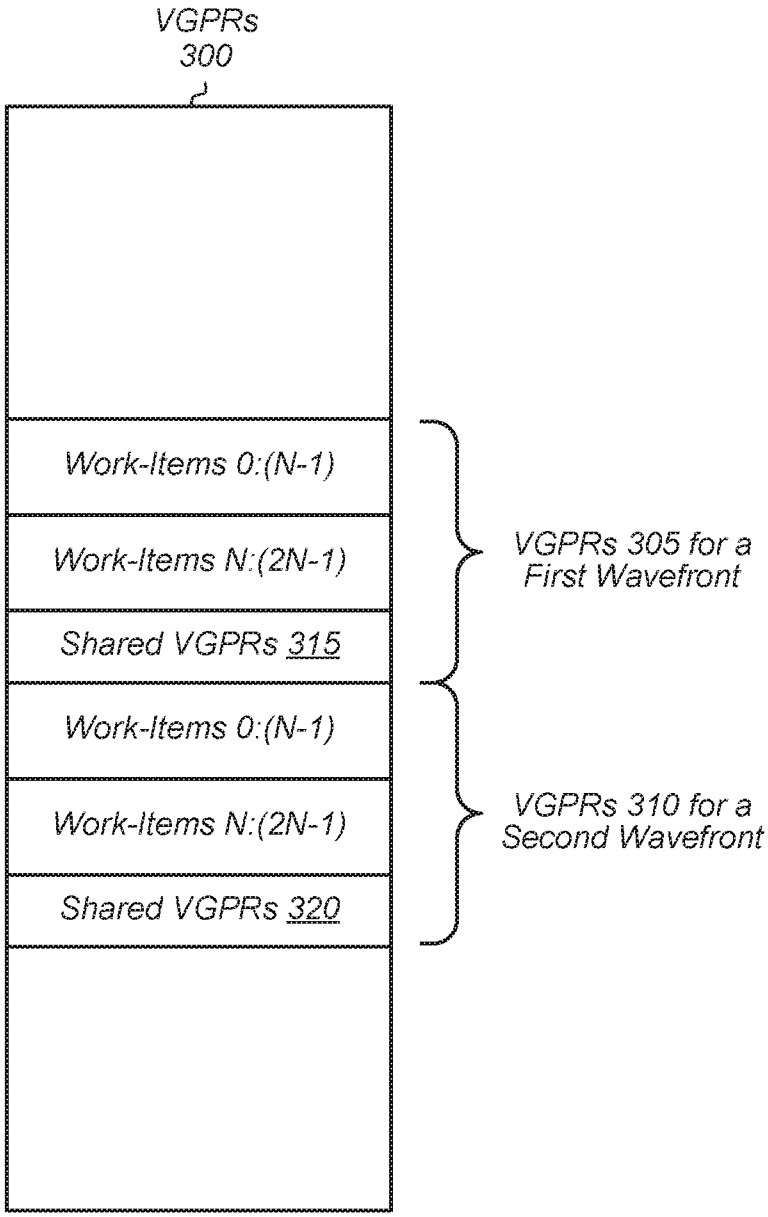
VGPRs
300
Work-Items 0:(N-1)
Work-Items N:(2N-1)
Shared VGPRs 315
Work-Items 0:(N-1)
Work-Items N:(2N-1)
Shared VGPRs 320
VGPRs 305 for a First Wavefront
VGPRs 310 for a Second Wavefront

FIG. 3

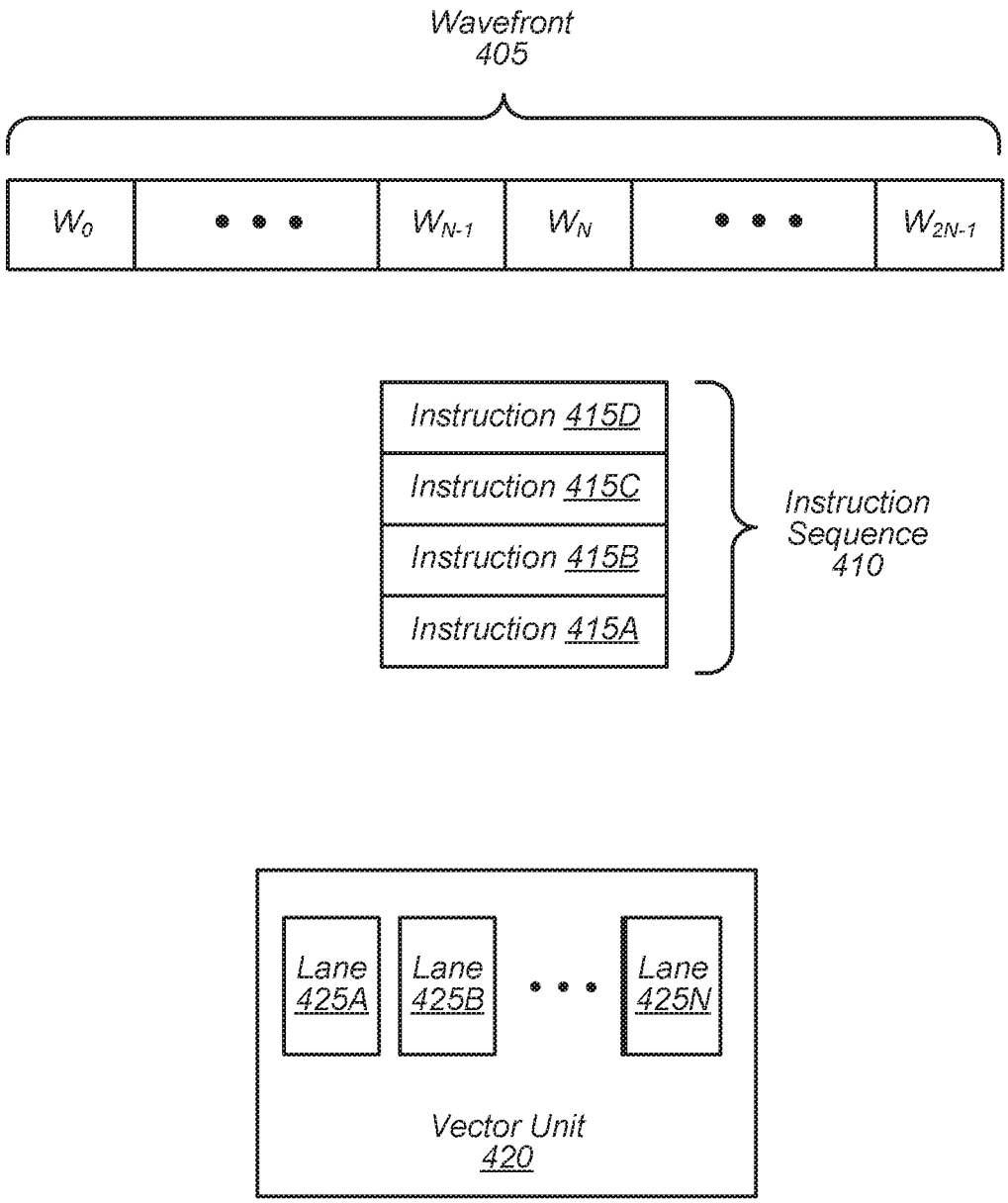
Wavefront
405
W_0
W_{N-1}
W_N
W_{2N-1}
Instruction 415D
Instruction 415C
Instruction 415B
Instruction 415A
Instruction Sequence 410
Lane 425A
Lane 425B
Lane 425N
Vector Unit 420

FIG. 4

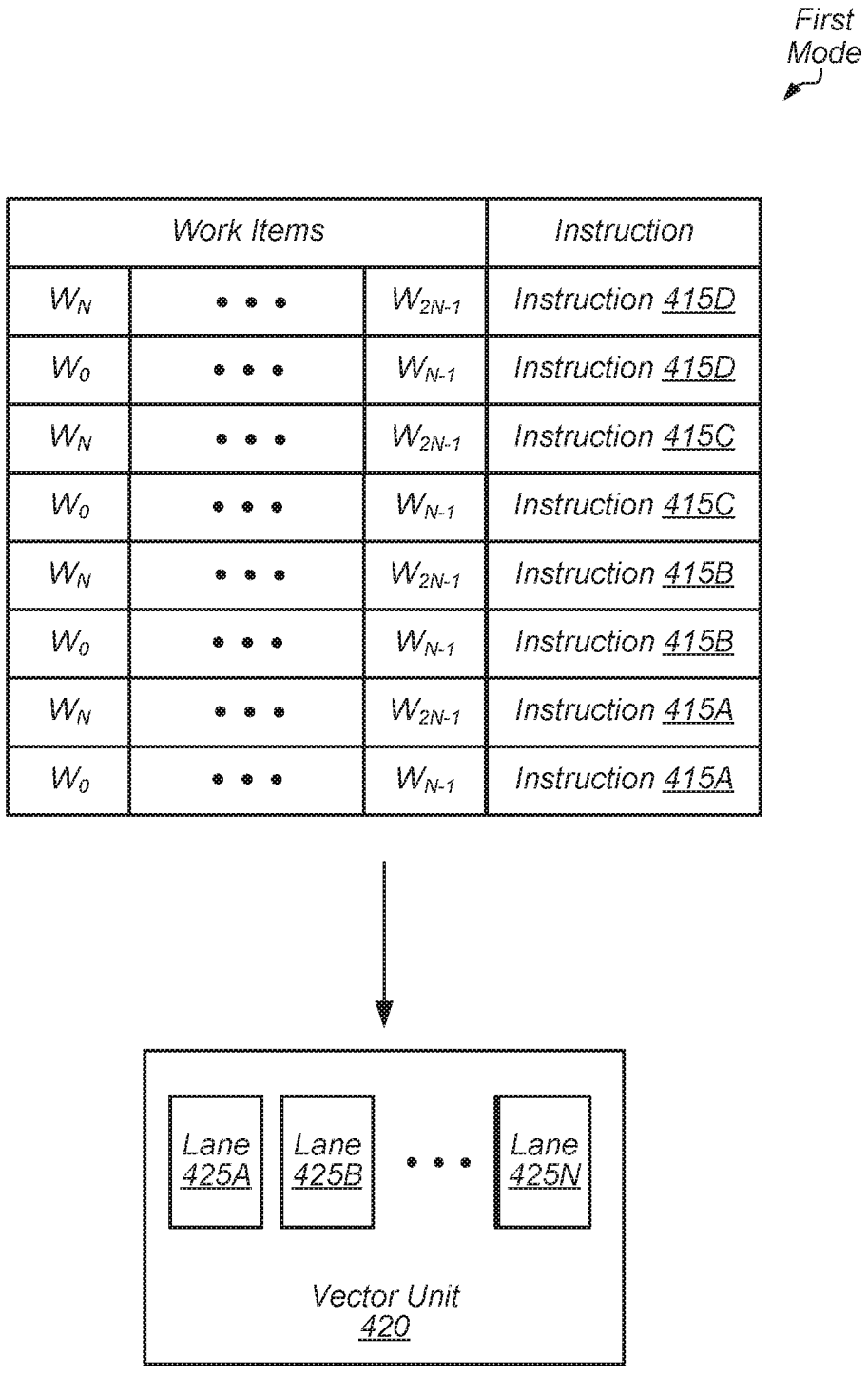
First Mode
Work Items
Instruction
W_N • • • W_{2N-1} Instruction 415D
W_0 • • • W_{N-1} Instruction 415D
W_N • • • W_{2N-1} Instruction 415C
W_0 • • • W_{N-1} Instruction 415C
W_N • • • W_{2N-1} Instruction 415B
W_0 • • • W_{N-1} Instruction 415B
W_N • • • W_{2N-1} Instruction 415A
W_0 • • • W_{N-1} Instruction 415A
Lane 425A
Lane 425B
• • •
Lane 425N
Vector Unit 420

FIG. 5

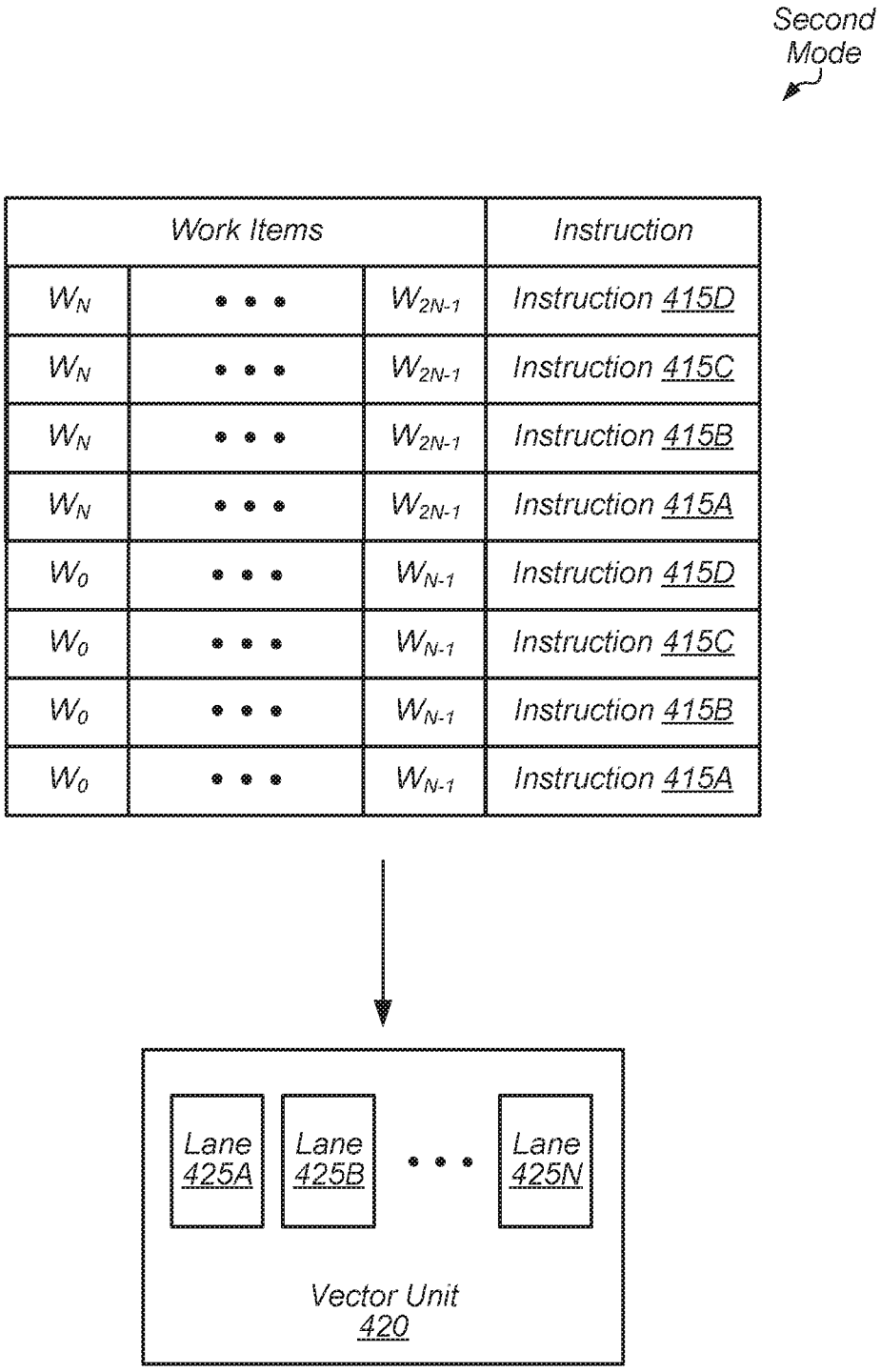
Second Mode
Work Items
Instruction
W_N • • • W_{2N-1} Instruction 415D
W_N • • • W_{2N-1} Instruction 415C
W_N • • • W_{2N-1} Instruction 415B
W_N • • • W_{2N-1} Instruction 415A
W_0 • • • W_{N-1} Instruction 415D
W_0 • • • W_{N-1} Instruction 415C
W_0 • • • W_{N-1} Instruction 415B
W_0 • • • W_{N-1} Instruction 415A
Lane 425A
Lane 425B
• • •
Lane 425N
Vector Unit 420

FIG. 6

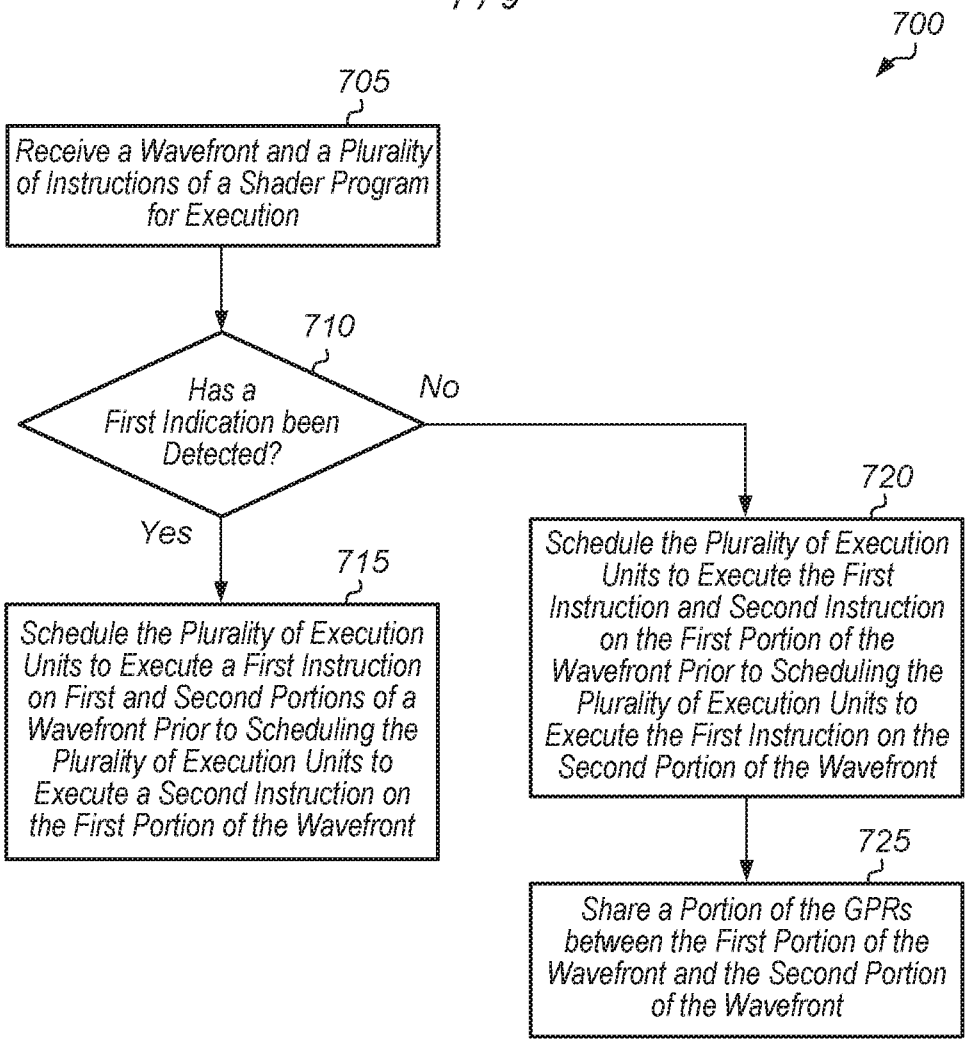
700
705
Receive a Wavefront and a Plurality of Instructions of a Shader Program for Execution
710
Has a First Indication been Detected?
No
Yes
715
Schedule the Plurality of Execution Units to Execute a First Instruction on First and Second Portions of a Wavefront Prior to Scheduling the Plurality of Execution Units to Execute a Second Instruction on the First Portion of the Wavefront
720
Schedule the Plurality of Execution Units to Execute the First Instruction and Second Instruction on the First Portion of the Wavefront Prior to Scheduling the Plurality of Execution Units to Execute the First Instruction on the Second Portion of the Wavefront
725
Share a Portion of the GPRs between the First Portion of the Wavefront and the Second Portion of the Wavefront

FIG. 7

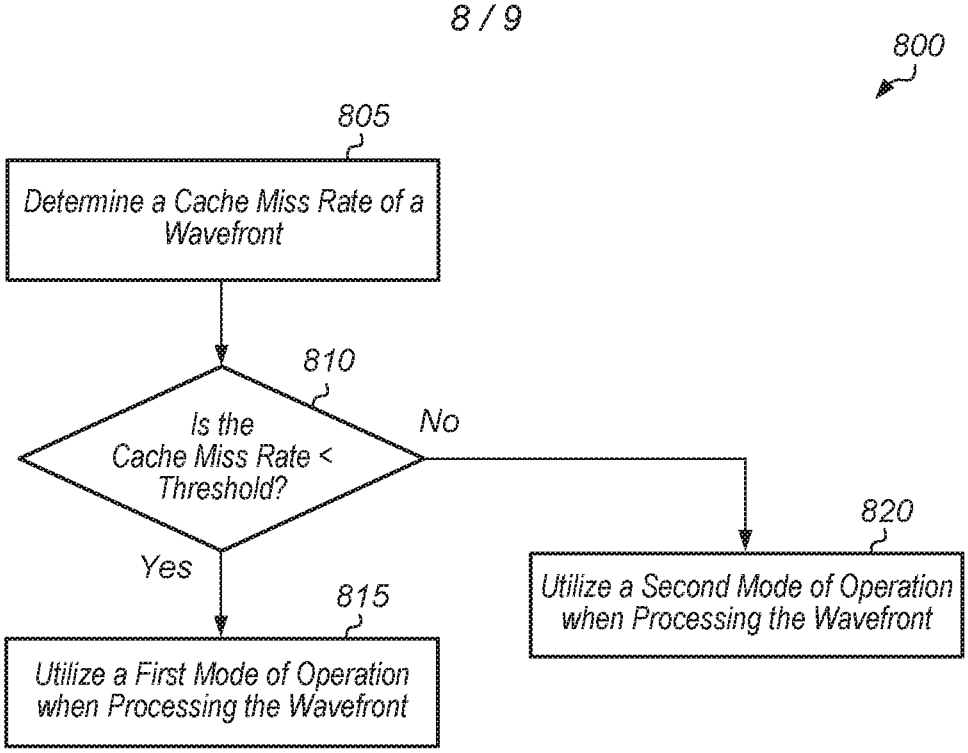
800
805
Determine a Cache Miss Rate of a Wavefront
810
Is the Cache Miss Rate < Threshold?
No
820
Utilize a Second Mode of Operation when Processing the Wavefront
Yes
815
Utilize a First Mode of Operation when Processing the Wavefront

FIG. 8

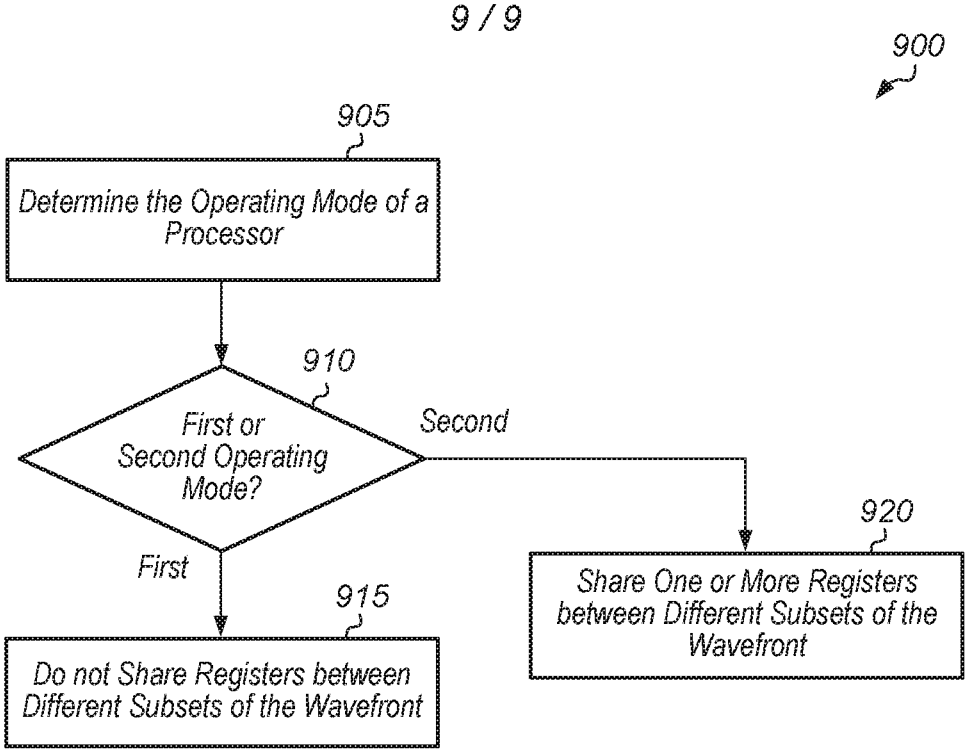
900
905
Determine the Operating Mode of a Processor
910
First or Second Operating Mode?
Second
First
915
Do not Share Registers between Different Subsets of the Wavefront
920
Share One or More Registers between Different Subsets of the Wavefront

FIG. 9

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 18/19026

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - G06F 9/30, G06F 7/38, G06F 9/44 (2018.01)
CPC - G06F 9/322, G06F 9/3851, G06F 9/3887, G06F 9/3009, G06F 9/30185, G06F 9/30072, G06F 9/30058

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)

See Search History Document

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

See Search History Document

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

See Search History Document

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2013/0117541 A1 (CHOQUETTE et al.), 09 May 2013 (09.05.2013), entire document, especially Abotraot; Para [0006], [0007] [00491-100521	1-20
Y	US 2015/0220346 A1 (Optimum Semiconductor Technologies, Inc), 06 August 2015 (06.00.2015), entire document, especially Abstract; Para [0014], [0050], [0105]-[0108]	1-20
Y	US 2014/0215187 A1 (ADVANCED MICRO DEVICES, INC), 31 July 2014 (31.07.2014), entire document, especially Abstract; Para [0040]-[0044], [0048]	6-7, 13-14, 20
A	US 7,761,697 B1 (Coon et al.), 20 July 2010 (20.07.2010), entire document	1-20
A	US 2014/0164737 A1 (COLLANGE et a.), 12 June 2014 (12.06.2014), entire document	1-20

☐ 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 another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
17 April 2018	07 MAY 2018

Name and mailing address of the ISA/US	Authorized officer:
Mail Stop PCT, Attn; ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-8300	Lee W. Young PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774

Form PCT/ISA/210 (second sheet) (January 2015)