



- (51) **International Patent Classification:**
G06F 1/32 (2006.01) *G06F 9/46* (2006.01)
- (21) **International Application Number:**
PCT/US2017/045252
- (22) **International Filing Date:**
03 August 2017 (03.08.2017)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
15/252,543 31 August 2016 (31.08.2016) US
- (71) **Applicant:** INTEL CORPORATION [US/US]; 2200 Mission College Boulevard, Santa Clara, California 95054 (US).
- (72) **Inventors:** ANANTHAKRISHNAN, Avinash N.; 8545 NW Ryan St., Portland, Oregon 97229 (US). RODRIGUEZ, Jorge P.; 2566 NW Marshall St., Unit B1, Portland, Oregon 97210 (US).
- (74) **Agent:** ROZMAN, Mark J. et al.; Trop, Pruner & Hu, P.C., 1616 S. Voss Rd., Ste. 750, Houston, Texas 77057-2631 (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).

Declarations under Rule 4.17:

- as to the identity of the inventor (Rule 4.17(i))
- 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))

(54) **Title:** CONTROLLING CURRENT CONSUMPTION OF A PROCESSOR BASED AT LEAST IN PART ON PLATFORM CAPACITANCE

1800

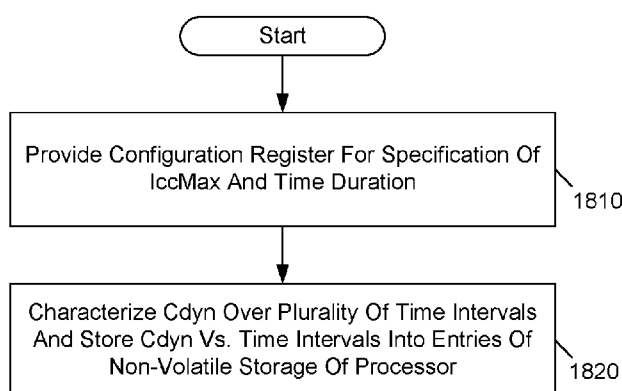


FIG. 18

(57) **Abstract:** In one embodiment, a processor includes: at least one core to execute instructions; a power controller to control power consumption of the processor; and a storage to store a plurality of entries to associate a dynamic capacitance with a time duration for which a current spike is to be exposed to a power delivery component. Other embodiments are described and claimed.

Published:

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

**CONTROLLING CURRENT CONSUMPTION OF A
PROCESSOR BASED AT LEAST IN PART ON PLATFORM CAPACITANCE**

Technical Field

[0001] Embodiments relate to power management of a system, and more particularly to power management of a multicore processor.

Background

[0002] Modern processors integrate a variety of different compute components into a single semiconductor die and can be incorporated into ever shrinking form factor computing devices. In these small form factor computing devices, processors are expected to provide the same level of performance and snappiness as traditional larger form factor computing devices. Due to form factor limitations, the size and capability of power delivery components like voltage regulators, inductors and other passive power delivery components are challenged. To ensure improved performance, a processor may desirably exit out of an idle state to an active high performance state as quickly as possible. Exiting out of an idle state oftentimes includes many different tasks, including increasing voltage to one or more components connected to the processor. However, increasing voltage at a fast rate places additional burden on power delivery components or requires increased component capability, which can adversely affect performance, cost and/or size.

[0003] Modern processors integrate multiple compute components into a single die, and a given processor design can be targeted for incorporation into a multitude of different form factor computing devices. Each individual form factor computing device comes with different power delivery capabilities and cost considerations. Extremely small form factor devices may not have the circuit board area to house a large amount of capacitors as part of a power delivery solution, and hence might have limited transient response capabilities. Large form factor devices may have additional capacitors, but adding large amounts of capacitors to the output of a voltage regulator results in larger inrush current to charge and discharge this large capacitance, which is a power tax to downstream power delivery components. In addition, more capacitance causes a cost increase.

[0004] Original equipment manufacturers (OEMs) may choose different tradeoffs, such that power delivery capabilities vary widely in different systems. However, processors offer limited flexibility in managing to specific power delivery capabilities. As such, unclaimed

performance is lost or additional cost/complexity is incurred in seeking to deliver increased levels of performance.

Brief Description of the Drawings

[0005] FIG. 1 is a block diagram of a portion of a system in accordance with an embodiment of the present invention.

[0006] FIG. 2 is a block diagram of a processor in accordance with an embodiment of the present invention.

[0007] FIG. 3 is a block diagram of a multi-domain processor in accordance with another embodiment of the present invention.

[0008] FIG. 4 is an embodiment of a processor including multiple cores.

[0009] FIG. 5 is a block diagram of a micro-architecture of a processor core in accordance with one embodiment of the present invention.

[0010] FIG. 6 is a block diagram of a micro-architecture of a processor core in accordance with another embodiment.

[0011] FIG. 7 is a block diagram of a micro-architecture of a processor core in accordance with yet another embodiment.

[0012] FIG. 8 is a block diagram of a micro-architecture of a processor core in accordance with a still further embodiment.

[0013] FIG. 9 is a block diagram of a processor in accordance with another embodiment of the present invention.

[0014] FIG. 10 is a block diagram of a representative SoC in accordance with an embodiment of the present invention.

[0015] FIG. 11 is a block diagram of another example SoC in accordance with an embodiment of the present invention.

[0016] FIG. 12 is a block diagram of an example system with which embodiments can be used.

[0017] FIG. 13 is a block diagram of another example system with which embodiments may be used.

[0018] FIG. 14 is a block diagram of a representative computer system.

[0019] FIG. 15 is a block diagram of a system in accordance with an embodiment of the present invention.

[0020] FIG. 16 is a block diagram illustrating an IP core development system used to manufacture an integrated circuit to perform operations according to an embodiment.

[0021] FIG. 17 is a graphical illustration of a dynamic capacitance of a processor over various time intervals in accordance with an embodiment of the present invention.

[0022] FIG. 18 is a flow diagram of a method in accordance with an embodiment of the present invention.

[0023] FIG. 19 is a flow diagram of a method for designing a computing platform in accordance with an embodiment.

[0024] FIG. 20 is a flow diagram of a method in accordance with yet another embodiment of the present invention.

Detailed Description

[0025] In various embodiments, a processor may be configured to maintain dynamic capacitance characterization information over a number of different time intervals. Such dynamic capacitance and corresponding time intervals may be stored in a non-volatile storage of the processor. When such processor is to be incorporated into a particular platform design, an OEM or other designer may specify a peak current delivery capability of a voltage regulator (e.g., a first stage voltage regulator) (where this peak current is referred to as I_{ccMax}) and a corresponding time duration over which current spikes up to this level are to be exposed to this voltage regulator. Such information may be provided to the processor, e.g., during initialization.

[0026] Based on this information, the processor may determine an appropriate value for dynamic capacitance using the provided time duration and the characterization information accessed from the non-volatile storage. From this information, the processor may determine

a peak frequency at which it can operate a given platform within the peak current delivery capability. In this way, a processor may enable operation at higher frequencies.

[0027] In contrast, most processors expose limited information to an OEM to configure the processor to stay within the power delivery capabilities of a particular system. For example, some processors provide a specified value for IccMax. As such, a system is designed with a first stage voltage regulator to support this specific peak current limit, or downstream power delivery components are configured to support a specific maximum power limit (Pmax). Typical processors expose limited OEM configurability via software-exposed registers to limit the maximum current or maximum power the processor is to draw. This information allows an OEM to configure the processor to stay within the power delivery capabilities provisioned by the OEM.

[0028] However, in reality IccMax (and Pmax) capability of a platform are functions of time. Depending on the bandwidth of the voltage regulator and the amount of capacitance on the platform, the peak current that a platform can handle will vary. Instantaneous current spikes that last less than a few nanoseconds (ns) are typically handled within the processor itself by an on-die power delivery network and built-in capacitances. Current spikes that last significantly longer than, e.g., 500 nanoseconds (ns) – 1 microsecond (μ s), make it out of the processor to the first stage voltage regulator. Current spikes of even longer duration (e.g., greater than 5 μ s) make it out to downstream power delivery components like a battery.

[0029] Yet typical processors only allow managing IccMax or Pmax independently of the time duration of current spikes. As such, processors manage frequency such that a current spike of shortest time duration does not make it out of the processor. If an OEM opts to use larger capacitors at the output of the first stage voltage regulator, current spikes of longer duration are filtered out by the capacitive network (as compared to if there is less output capacitance) and only the unfiltered current spike is sourced by the voltage regulator.

[0030] Although the following embodiments are described with reference to energy conservation and energy efficiency in specific integrated circuits, such as in computing platforms or processors, other embodiments are applicable to other types of integrated circuits and logic devices. Similar techniques and teachings of embodiments described herein may be applied to other types of circuits or semiconductor devices that may also

benefit from better energy efficiency and energy conservation. For example, the disclosed embodiments are not limited to any particular type of computer systems. That is, disclosed embodiments can be used in many different system types, ranging from server computers (e.g., tower, rack, blade, micro-server and so forth), communications systems, storage systems, desktop computers of any configuration, laptop, notebook, and tablet computers (including 2:1 tablets, phablets and so forth), and may be also used in other devices, such as handheld devices, systems on chip (SoCs), and embedded applications. Some examples of handheld devices include cellular phones such as smartphones, Internet protocol devices, digital cameras, personal digital assistants (PDAs), and handheld PCs. Embedded applications may typically include a microcontroller, a digital signal processor (DSP), network computers (NetPC), set-top boxes, network hubs, wide area network (WAN) switches, wearable devices, or any other system that can perform the functions and operations taught below. More so, embodiments may be implemented in mobile terminals having standard voice functionality such as mobile phones, smartphones and phablets, and/or in non-mobile terminals without a standard wireless voice function communication capability, such as many wearables, tablets, notebooks, desktops, micro-servers, servers and so forth. Moreover, the apparatuses, methods, and systems described herein are not limited to physical computing devices, but may also relate to software optimizations for energy conservation and efficiency. As will become readily apparent in the description below, the embodiments of methods, apparatuses, and systems described herein (whether in reference to hardware, firmware, software, or a combination thereof) are vital to a ‘green technology’ future, such as for power conservation and energy efficiency in products that encompass a large portion of the US economy.

[0031] Referring now to FIG. 1, shown is a block diagram of a portion of a system in accordance with an embodiment of the present invention. As shown in FIG. 1, system 100 may include various components, including a processor 110 which as shown is a multicore processor. Processor 110 may be coupled to a power supply 150 via an external voltage regulator 160, which may perform a first voltage conversion to provide a primary regulated voltage V_{reg} to processor 110.

[0032] As seen, processor 110 may be a single die processor including multiple cores 120a – 120n. In addition, each core may be associated with an integrated voltage regulator (IVR)

125a – 125n which receives the primary regulated voltage and generates an operating voltage to be provided to one or more agents of the processor associated with the IVR. Accordingly, an IVR implementation may be provided to allow for fine-grained control of voltage and thus power and performance of each individual core. As such, each core can operate at an independent voltage and frequency, enabling great flexibility and affording wide opportunities for balancing power consumption with performance. In some embodiments, the use of multiple IVRs enables the grouping of components into separate power planes, such that power is regulated and supplied by the IVR to only those components in the group. During power management, a given power plane of one IVR may be powered down or off when the processor is placed into a certain low power state, while another power plane of another IVR remains active, or fully powered. Similarly, cores 120 may include or be associated with independent clock generation circuitry such as one or more phase lock loops (PLLs) to control operating frequency of each core 120 independently.

[0033] Still referring to FIG. 1, additional components may be present within the processor including an input/output interface (IF) 132, another interface 134, and an integrated memory controller (IMC) 136. As seen, each of these components may be powered by another integrated voltage regulator 125_x. In one embodiment, interface 132 may enable operation for an Intel® Quick Path Interconnect (QPI) interconnect, which provides for point-to-point (PtP) links in a cache coherent protocol that includes multiple layers including a physical layer, a link layer and a protocol layer. In turn, interface 134 may communicate via a Peripheral Component Interconnect Express (PCIeTM) protocol.

[0034] Also shown is a power control unit (PCU) 138, which may include circuitry including hardware, software and/or firmware to perform power management operations with regard to processor 110. As seen, PCU 138 provides control information to external voltage regulator 160 via a digital interface 162 to cause the voltage regulator to generate the appropriate regulated voltage. PCU 138 also provides control information to IVRs 125 via another digital interface 163 to control the operating voltage generated (or to cause a corresponding IVR to be disabled in a low power mode). In various embodiments, PCU 138 may include a variety of power management logic units to perform hardware-based power management. For example, PCU 138 may include logic to dynamically determine an appropriate C_{dyn} value for the processor, based on a set of stored values and configuration

information received for a particular platform in which processor 100 is implemented, as described herein. Such power management may be wholly processor controlled (e.g., by various processor hardware, and which may be triggered by workload and/or power, thermal or other processor constraints) and/or the power management may be performed responsive to external sources (such as a platform or power management source or system software).

[0035] In FIG. 1, PCU 138 is illustrated as being present as a separate logic of the processor. In other cases PCU logic 138 may execute on a given one or more of cores 120. In some cases, PCU 138 may be implemented as a microcontroller (dedicated or general-purpose) or other control logic configured to execute its own dedicated power management code, sometimes referred to as P-code. In yet other embodiments, power management operations to be performed by PCU 138 may be implemented externally to a processor, such as by way of a separate power management integrated circuit (PMIC) or other component external to the processor. In yet other embodiments, power management operations to be performed by PCU 138 may be implemented within BIOS or other system software.

[0036] Embodiments may be particularly suitable for a multicore processor in which each of multiple cores can operate at an independent voltage and frequency point. As used herein the term “domain” is used to mean a collection of hardware and/or logic that operates at the same voltage and frequency point. In addition, a multicore processor can further include other non-core processing engines such as fixed function units, graphics engines, and so forth. Such processor can include independent domains other than the cores, such as one or more domains associated with a graphics engine (referred to herein as a graphics domain) and one or more domains associated with non-core circuitry, referred to herein as an uncore or a system agent. Although many implementations of a multi-domain processor can be formed on a single semiconductor die, other implementations can be realized by a multi-chip package in which different domains can be present on different semiconductor die of a single package.

[0037] While not shown for ease of illustration, understand that additional components may be present within processor 110 such as uncore logic, and other components such as internal memories, e.g., one or more levels of a cache memory hierarchy and so forth. Furthermore, while shown in the implementation of FIG. 1 with an integrated voltage regulator, embodiments are not so limited. For example, other regulated voltages may be provided to on-chip resources from external voltage regulator 160 or one or more additional external

sources of regulated voltages, and to which current spikes of predetermined durations may be safely sourced.

[0038] Note that the power management techniques described herein may be independent of and complementary to an operating system (OS)-based power management (OSPM) mechanism. According to one example OSPM technique, a processor can operate at various performance states or levels, so-called P-states, namely from P0 to PN. In general, the P1 performance state may correspond to the highest guaranteed performance state that can be requested by an OS. In addition to this P1 state, the OS can further request a higher performance state, namely a P0 state. This P0 state may thus be an opportunistic, overclocking, or turbo mode state in which, when power and/or thermal budget is available, processor hardware can configure the processor or at least portions thereof to operate at a higher than guaranteed frequency. In many implementations a processor can include multiple so-called bin frequencies above the P1 guaranteed maximum frequency, exceeding to a maximum peak frequency of the particular processor, as fused or otherwise written into the processor during manufacture. In addition, according to one OSPM mechanism, a processor can operate at various power states or levels. With regard to power states, an OSPM mechanism may specify different power consumption states, generally referred to as C-states, C0, C1 to Cn states. When a core is active, it runs at a C0 state, and when the core is idle it may be placed in a core low power state, also called a core non-zero C-state (e.g., C1-C6 states), with each C-state being at a lower power consumption level (such that C6 is a deeper low power state than C1, and so forth).

[0039] Understand that many different types of power management techniques may be used individually or in combination in different embodiments. As representative examples, a power controller may control the processor to be power managed by some form of dynamic voltage frequency scaling (DVFS) in which an operating voltage and/or operating frequency of one or more cores or other processor logic may be dynamically controlled to reduce power consumption in certain situations. In an example, DVFS may be performed using Enhanced Intel SpeedStep™ technology available from Intel Corporation, Santa Clara, CA, to provide optimal performance at a lowest power consumption level. In another example, DVFS may be performed using Intel TurboBoost™ technology to enable one or more cores or other

compute engines to operate at a higher than guaranteed operating frequency based on conditions (e.g., workload and availability).

[0040] Another power management technique that may be used in certain examples is dynamic swapping of workloads between different compute engines. For example, the processor may include asymmetric cores or other processing engines that operate at different power consumption levels, such that in a power constrained situation, one or more workloads can be dynamically switched to execute on a lower power core or other compute engine. Another exemplary power management technique is hardware duty cycling (HDC), which may cause cores and/or other compute engines to be periodically enabled and disabled according to a duty cycle, such that one or more cores may be made inactive during an inactive period of the duty cycle and made active during an active period of the duty cycle.

[0041] Power management techniques also may be used when constraints exist in an operating environment. For example, when a power and/or thermal constraint is encountered, power may be reduced by reducing operating frequency and/or voltage. Other power management techniques include throttling instruction execution rate or limiting scheduling of instructions. Still further, it is possible for instructions of a given instruction set architecture to include express or implicit direction as to power management operations. Although described with these particular examples, understand that many other power management techniques may be used in particular embodiments.

[0042] Embodiments can be implemented in processors for various markets including server processors, desktop processors, mobile processors and so forth. Referring now to FIG. 2, shown is a block diagram of a processor in accordance with an embodiment of the present invention. As shown in FIG. 2, processor 200 may be a multicore processor including a plurality of cores $210_a - 210_n$. In one embodiment, each such core may be of an independent power domain and can be configured to enter and exit active states and/or maximum performance states based on workload. One or more cores 210 may be heterogeneous to the other cores, e.g., having different micro-architectures, instruction set architectures, pipeline depths, power and performance capabilities. The various cores may be coupled via an interconnect 215 to a system agent or uncore 220 that includes various components. As seen, the uncore 220 may include a shared cache 230 which may be a last level cache. In addition, the uncore may include an integrated memory controller 240 to communicate with a system

memory (not shown in FIG. 2), e.g., via a memory bus. Uncore 220 also includes various interfaces 250 and a power control unit 255, which may include logic to perform the power management techniques described herein including a dynamic determination of dynamic capacitance for a particular platform including processor 200. This dynamic capacitance may further be used by power control unit 255 to determine a peak operating frequency at which the processor may operate, while maintaining operation at or below a configured maximum current consumption level, e.g., I_{ccMax} . To dynamically determine this dynamic capacitance, power control unit 255 may access a non-volatile storage 265, which may store a table of entries, each associating a given dynamic capacitance value with a time interval at which a current spike is exposed to an external voltage regulator.

[0043] In addition, by interfaces 250a-250n, connection can be made to various off-chip components such as peripheral devices, mass storage and so forth. While shown with this particular implementation in the embodiment of FIG. 2, the scope of the present invention is not limited in this regard.

[0044] Referring now to FIG. 3, shown is a block diagram of a multi-domain processor in accordance with another embodiment of the present invention. As shown in the embodiment of FIG. 3, processor 300 includes multiple domains. Specifically, a core domain 310 can include a plurality of cores 310a-310n, a graphics domain 320 can include one or more graphics engines, and a system agent domain 350 may further be present. In some embodiments, system agent domain 350 may execute at an independent frequency than the core domain and may remain powered on at all times to handle power control events and power management such that domains 310 and 320 can be controlled to dynamically enter into and exit high power and low power states. Each of domains 310 and 320 may operate at different voltage and/or power. Note that while only shown with three domains, understand the scope of the present invention is not limited in this regard and additional domains can be present in other embodiments. For example, multiple core domains may be present each including at least one core.

[0045] In general, each core 310 may further include low level caches in addition to various execution units and additional processing elements. In turn, the various cores may be coupled to each other and to a shared cache memory formed of a plurality of units of a last level cache (LLC) 340a - 340n. In various embodiments, LLC 340 may be shared amongst

the cores and the graphics engine, as well as various media processing circuitry. As seen, a ring interconnect 330 thus couples the cores together, and provides interconnection between the cores, graphics domain 320 and system agent circuitry 350. In one embodiment, interconnect 330 can be part of the core domain. However in other embodiments the ring interconnect can be of its own domain.

[0046] As further seen, system agent domain 350 may include display controller 352 which may provide control of and an interface to an associated display. As further seen, system agent domain 350 may include a power control unit 355 which can include logic to perform the power management techniques described herein, including the dynamic capacitance determination and use of this value for determining a peak operating frequency as described herein.

[0047] As further seen in FIG. 3, processor 300 can further include an integrated memory controller (IMC) 370 that can provide for an interface to a system memory, such as a dynamic random access memory (DRAM). Multiple interfaces 380a – 380n may be present to enable interconnection between the processor and other circuitry. For example, in one embodiment at least one direct media interface (DMI) interface may be provided as well as one or more PCIe™ interfaces. Still further, to provide for communications between other agents such as additional processors or other circuitry, one or more QPI interfaces may also be provided. Although shown at this high level in the embodiment of FIG. 3, understand the scope of the present invention is not limited in this regard.

[0048] Referring to FIG. 4, an embodiment of a processor including multiple cores is illustrated. Processor 400 includes any processor or processing device, such as a microprocessor, an embedded processor, a digital signal processor (DSP), a network processor, a handheld processor, an application processor, a co-processor, a system on a chip (SoC), or other device to execute code. Processor 400, in one embodiment, includes at least two cores—cores 401 and 402, which may include asymmetric cores or symmetric cores (the illustrated embodiment). However, processor 400 may include any number of processing elements that may be symmetric or asymmetric.

[0049] In one embodiment, a processing element refers to hardware or logic to support a software thread. Examples of hardware processing elements include: a thread unit, a thread

slot, a thread, a process unit, a context, a context unit, a logical processor, a hardware thread, a core, and/or any other element, which is capable of holding a state for a processor, such as an execution state or architectural state. In other words, a processing element, in one embodiment, refers to any hardware capable of being independently associated with code, such as a software thread, operating system, application, or other code. A physical processor typically refers to an integrated circuit, which potentially includes any number of other processing elements, such as cores or hardware threads.

[0050] A core often refers to logic located on an integrated circuit capable of maintaining an independent architectural state, wherein each independently maintained architectural state is associated with at least some dedicated execution resources. In contrast to cores, a hardware thread typically refers to any logic located on an integrated circuit capable of maintaining an independent architectural state, wherein the independently maintained architectural states share access to execution resources. As can be seen, when certain resources are shared and others are dedicated to an architectural state, the line between the nomenclature of a hardware thread and core overlaps. Yet often, a core and a hardware thread are viewed by an operating system as individual logical processors, where the operating system is able to individually schedule operations on each logical processor.

[0051] Physical processor 400, as illustrated in FIG. 4, includes two cores, cores 401 and 402. Here, cores 401 and 402 are considered symmetric cores, i.e., cores with the same configurations, functional units, and/or logic. In another embodiment, core 401 includes an out-of-order processor core, while core 402 includes an in-order processor core. However, cores 401 and 402 may be individually selected from any type of core, such as a native core, a software managed core, a core adapted to execute a native instruction set architecture (ISA), a core adapted to execute a translated ISA, a co-designed core, or other known core. Yet to further the discussion, the functional units illustrated in core 401 are described in further detail below, as the units in core 402 operate in a similar manner.

[0052] As depicted, core 401 includes two hardware threads 401a and 401b, which may also be referred to as hardware thread slots 401a and 401b. Therefore, software entities, such as an operating system, in one embodiment potentially view processor 400 as four separate processors, i.e., four logical processors or processing elements capable of executing four software threads concurrently. As alluded to above, a first thread is associated with

architecture state registers 401a, a second thread is associated with architecture state registers 401b, a third thread may be associated with architecture state registers 402a, and a fourth thread may be associated with architecture state registers 402b. Here, each of the architecture state registers (401a, 401b, 402a, and 402b) may be referred to as processing elements, thread slots, or thread units, as described above. As illustrated, architecture state registers 401a are replicated in architecture state registers 401b, so individual architecture states/contexts are capable of being stored for logical processor 401a and logical processor 401b. In core 401, other smaller resources, such as instruction pointers and renaming logic in allocator and renamer block 430 may also be replicated for threads 401a and 401b. Some resources, such as re-order buffers in reorder/retirement unit 435, branch target buffer and instruction translation lookaside buffer (BTB and I-TLB) 420, load/store buffers, and queues may be shared through partitioning. Other resources, such as general purpose internal registers, page-table base register(s), low-level data-cache and data-TLB 450, execution unit(s) 440, and portions of out-of-order unit 435 are potentially fully shared.

[0053] Processor 400 often includes other resources, which may be fully shared, shared through partitioning, or dedicated by/to processing elements. In FIG. 4, an embodiment of a purely exemplary processor with illustrative logical units/resources of a processor is illustrated. Note that a processor may include, or omit, any of these functional units, as well as include any other known functional units, logic, or firmware not depicted. As illustrated, core 401 includes a simplified, representative out-of-order (OOO) processor core. But an in-order processor may be utilized in different embodiments. The OOO core includes a branch target buffer 420 to predict branches to be executed/taken and an instruction-translation buffer (I-TLB) 420 to store address translation entries for instructions.

[0054] Core 401 further includes decode module 425 coupled to a fetch unit to decode fetched elements. Fetch logic, in one embodiment, includes individual sequencers associated with thread slots 401a, 401b, respectively. Usually core 401 is associated with a first ISA, which defines/specifies instructions executable on processor 400. Often machine code instructions that are part of the first ISA include a portion of the instruction (referred to as an opcode), which references/specifies an instruction or operation to be performed. Decode logic 425 includes circuitry that recognizes these instructions from their opcodes and passes the decoded instructions on in the pipeline for processing as defined by the first ISA. For

example, decoders 425, in one embodiment, include logic designed or adapted to recognize specific instructions, such as transactional instruction. As a result of the recognition by decoders 425, the architecture or core 401 takes specific, predefined actions to perform tasks associated with the appropriate instruction. It is important to note that any of the tasks, blocks, operations, and methods described herein may be performed in response to a single or multiple instructions; some of which may be new or old instructions.

[0055] In one example, allocator and renamer block 430 includes an allocator to reserve resources, such as register files to store instruction processing results. However, threads 401a and 401b are potentially capable of out-of-order execution, where allocator and renamer block 430 also reserves other resources, such as reorder buffers to track instruction results. Unit 430 may also include a register renamer to rename program/instruction reference registers to other registers internal to processor 400. Reorder/retirement unit 435 includes components, such as the reorder buffers mentioned above, load buffers, and store buffers, to support out-of-order execution and later in-order retirement of instructions executed out-of-order.

[0056] Scheduler and execution unit(s) block 440, in one embodiment, includes a scheduler unit to schedule instructions/operation on execution units. For example, a floating point instruction is scheduled on a port of an execution unit that has an available floating point execution unit. Register files associated with the execution units are also included to store information instruction processing results. Exemplary execution units include a floating point execution unit, an integer execution unit, a jump execution unit, a load execution unit, a store execution unit, and other known execution units.

[0057] Lower level data cache and data translation lookaside buffer (D-TLB) 450 are coupled to execution unit(s) 440. The data cache is to store recently used/operated on elements, such as data operands, which are potentially held in memory coherency states. The D-TLB is to store recent virtual/linear to physical address translations. As a specific example, a processor may include a page table structure to break physical memory into a plurality of virtual pages.

[0058] Here, cores 401 and 402 share access to higher-level or further-out cache 410, which is to cache recently fetched elements. Note that higher-level or further-out refers to

cache levels increasing or getting further away from the execution unit(s). In one embodiment, higher-level cache 410 is a last-level data cache—last cache in the memory hierarchy on processor 400—such as a second or third level data cache. However, higher level cache 410 is not so limited, as it may be associated with or includes an instruction cache. A trace cache—a type of instruction cache—instead may be coupled after decoder 425 to store recently decoded traces.

[0059] In the depicted configuration, processor 400 also includes bus interface module 405 and a power control unit 460, which may perform power management in accordance with an embodiment of the present invention. In this scenario, bus interface 405 is to communicate with devices external to processor 400, such as system memory and other components.

[0060] A memory controller 470 may interface with other devices such as one or many memories. In an example, bus interface 405 includes a ring interconnect with a memory controller for interfacing with a memory and a graphics controller for interfacing with a graphics processor. In an SoC environment, even more devices, such as a network interface, coprocessors, memory, graphics processor, and any other known computer devices/interface may be integrated on a single die or integrated circuit to provide small form factor with high functionality and low power consumption.

[0061] Referring now to FIG. 5, shown is a block diagram of a micro-architecture of a processor core in accordance with one embodiment of the present invention. As shown in FIG. 5, processor core 500 may be a multi-stage pipelined out-of-order processor. Core 500 may operate at various voltages based on a received operating voltage, which may be received from an integrated voltage regulator or external voltage regulator.

[0062] As seen in FIG. 5, core 500 includes front end units 510, which may be used to fetch instructions to be executed and prepare them for use later in the processor pipeline. For example, front end units 510 may include a fetch unit 501, an instruction cache 503, and an instruction decoder 505. In some implementations, front end units 510 may further include a trace cache, along with microcode storage as well as a micro-operation storage. Fetch unit 501 may fetch macro-instructions, e.g., from memory or instruction cache 503, and feed them to instruction decoder 505 to decode them into primitives, i.e., micro-operations for execution by the processor.

[0063] Coupled between front end units 510 and execution units 520 is an out-of-order (OOO) engine 515 that may be used to receive the micro-instructions and prepare them for execution. More specifically OOO engine 515 may include various buffers to re-order micro-instruction flow and allocate various resources needed for execution, as well as to provide renaming of logical registers onto storage locations within various register files such as register file 530 and extended register file 535. Register file 530 may include separate register files for integer and floating point operations. For purposes of configuration, control, and additional operations, a set of machine specific registers (MSRs) 538 may also be present and accessible to various logic within core 500 (and external to the core).

[0064] Various resources may be present in execution units 520, including, for example, various integer, floating point, and single instruction multiple data (SIMD) logic units, among other specialized hardware. For example, such execution units may include one or more arithmetic logic units (ALUs) 522 and one or more vector execution units 524, among other such execution units.

[0065] Results from the execution units may be provided to retirement logic, namely a reorder buffer (ROB) 540. More specifically, ROB 540 may include various arrays and logic to receive information associated with instructions that are executed. This information is then examined by ROB 540 to determine whether the instructions can be validly retired and result data committed to the architectural state of the processor, or whether one or more exceptions occurred that prevent a proper retirement of the instructions. Of course, ROB 540 may handle other operations associated with retirement.

[0066] As shown in FIG. 5, ROB 540 is coupled to a cache 550 which, in one embodiment may be a low level cache (e.g., an L1 cache) although the scope of the present invention is not limited in this regard. Also, execution units 520 can be directly coupled to cache 550. From cache 550, data communication may occur with higher level caches, system memory and so forth. While shown with this high level in the embodiment of FIG. 5, understand the scope of the present invention is not limited in this regard. For example, while the implementation of FIG. 5 is with regard to an out-of-order machine such as of an Intel® x86 instruction set architecture (ISA), the scope of the present invention is not limited in this regard. That is, other embodiments may be implemented in an in-order processor, a reduced instruction set computing (RISC) processor such as an ARM-based processor, or a processor

of another type of ISA that can emulate instructions and operations of a different ISA via an emulation engine and associated logic circuitry.

[0067] Referring now to FIG. 6, shown is a block diagram of a micro-architecture of a processor core in accordance with another embodiment. In the embodiment of FIG. 6, core 600 may be a low power core of a different micro-architecture, such as an Intel® Atom™-based processor having a relatively limited pipeline depth designed to reduce power consumption. As seen, core 600 includes an instruction cache 610 coupled to provide instructions to an instruction decoder 615. A branch predictor 605 may be coupled to instruction cache 610. Note that instruction cache 610 may further be coupled to another level of a cache memory, such as an L2 cache (not shown for ease of illustration in FIG. 6). In turn, instruction decoder 615 provides decoded instructions to an issue queue (IQ) 620 for storage and delivery to a given execution pipeline. A microcode ROM 618 is coupled to instruction decoder 615.

[0068] A floating point pipeline 630 includes a floating point (FP) register file 632 which may include a plurality of architectural registers of a given bit width such as 128, 256 or 512 bits. Pipeline 630 includes a floating point scheduler 634 to schedule instructions for execution on one of multiple execution units of the pipeline. In the embodiment shown, such execution units include an ALU 635, a shuffle unit 636, and a floating point adder 638. In turn, results generated in these execution units may be provided back to buffers and/or registers of register file 632. Of course understand while shown with these few example execution units, additional or different floating point execution units may be present in another embodiment.

[0069] An integer pipeline 640 also may be provided. In the embodiment shown, pipeline 640 includes an integer (INT) register file 642 which may include a plurality of architectural registers of a given bit width such as 128 or 256 bits. Pipeline 640 includes an integer execution (IE) scheduler 644 to schedule instructions for execution on one of multiple execution units of the pipeline. In the embodiment shown, such execution units include an ALU 645, a shifter unit 646, and a jump execution unit (JEU) 648. In turn, results generated in these execution units may be provided back to buffers and/or registers of register file 642. Of course understand while shown with these few example execution units, additional or different integer execution units may be present in another embodiment.

[0070] A memory execution (ME) scheduler 650 may schedule memory operations for execution in an address generation unit (AGU) 652, which is also coupled to a TLB 654. As seen, these structures may couple to a data cache 660, which may be a L0 and/or L1 data cache that in turn couples to additional levels of a cache memory hierarchy, including an L2 cache memory.

[0071] To provide support for out-of-order execution, an allocator/renamer 670 may be provided, in addition to a reorder buffer 680, which is configured to reorder instructions executed out of order for retirement in order. Although shown with this particular pipeline architecture in the illustration of FIG. 6, understand that many variations and alternatives are possible.

[0072] Note that in a processor having asymmetric cores, such as in accordance with the micro-architectures of FIGs. 5 and 6, workloads may be dynamically swapped between the cores for power management reasons, as these cores, although having different pipeline designs and depths, may be of the same or related ISA. Such dynamic core swapping may be performed in a manner transparent to a user application (and possibly kernel also).

[0073] Referring to FIG. 7, shown is a block diagram of a micro-architecture of a processor core in accordance with yet another embodiment. As illustrated in FIG. 7, a core 700 may include a multi-staged in-order pipeline to execute at very low power consumption levels. As one such example, processor 700 may have a micro-architecture in accordance with an ARM Cortex A53 design available from ARM Holdings, LTD., Sunnyvale, CA. In an implementation, an 8-stage pipeline may be provided that is configured to execute both 32-bit and 64-bit code. Core 700 includes a fetch unit 710 that is configured to fetch instructions and provide them to a decode unit 715, which may decode the instructions, e.g., macro-instructions of a given ISA such as an ARMv8 ISA. Note further that a queue 730 may couple to decode unit 715 to store decoded instructions. Decoded instructions are provided to an issue logic 725, where the decoded instructions may be issued to a given one of multiple execution units.

[0074] With further reference to FIG. 7, issue logic 725 may issue instructions to one of multiple execution units. In the embodiment shown, these execution units include an integer unit 735, a multiply unit 740, a floating point/vector unit 750, a dual issue unit 760, and a

load/store unit 770. The results of these different execution units may be provided to a writeback (WB) unit 780. Understand that while a single writeback unit is shown for ease of illustration, in some implementations separate writeback units may be associated with each of the execution units. Furthermore, understand that while each of the units and logic shown in FIG. 7 is represented at a high level, a particular implementation may include more or different structures. A processor designed using one or more cores having a pipeline as in FIG. 7 may be implemented in many different end products, extending from mobile devices to server systems.

[0075] Referring to FIG. 8, shown is a block diagram of a micro-architecture of a processor core in accordance with a still further embodiment. As illustrated in FIG. 8, a core 800 may include a multi-stage multi-issue out-of-order pipeline to execute at very high performance levels (which may occur at higher power consumption levels than core 700 of FIG. 7). As one such example, processor 800 may have a microarchitecture in accordance with an ARM Cortex A57 design. In an implementation, a 15 (or greater)-stage pipeline may be provided that is configured to execute both 32-bit and 64-bit code. In addition, the pipeline may provide for 3 (or greater)-wide and 3 (or greater)-issue operation. Core 800 includes a fetch unit 810 that is configured to fetch instructions and provide them to a decoder/renamer/dispatcher unit 815 coupled to a cache 820. Unit 815 may decode the instructions, e.g., macro-instructions of an ARMv8 instruction set architecture, rename register references within the instructions, and dispatch the instructions (eventually) to a selected execution unit. Decoded instructions may be stored in a queue 825. Note that while a single queue structure is shown for ease of illustration in FIG 8, understand that separate queues may be provided for each of the multiple different types of execution units.

[0076] Also shown in FIG. 8 is an issue logic 830 from which decoded instructions stored in queue 825 may be issued to a selected execution unit. Issue logic 830 also may be implemented in a particular embodiment with a separate issue logic for each of the multiple different types of execution units to which issue logic 830 couples.

[0077] Decoded instructions may be issued to a given one of multiple execution units. In the embodiment shown, these execution units include one or more integer units 835, a multiply unit 840, a floating point/vector unit 850, a branch unit 860, and a load/store unit 870. In an embodiment, floating point/vector unit 850 may be configured to handle SIMD or

vector data of 128 or 256 bits. Still further, floating point/vector execution unit 850 may perform IEEE-754 double precision floating-point operations. The results of these different execution units may be provided to a writeback unit 880. Note that in some implementations separate writeback units may be associated with each of the execution units. Furthermore, understand that while each of the units and logic shown in FIG. 8 is represented at a high level, a particular implementation may include more or different structures.

[0078] Note that in a processor having asymmetric cores, such as in accordance with the micro-architectures of FIGs. 7 and 8, workloads may be dynamically swapped for power management reasons, as these cores, although having different pipeline designs and depths, may be of the same or related ISA. Such dynamic core swapping may be performed in a manner transparent to a user application (and possibly kernel also).

[0079] A processor designed using one or more cores having pipelines as in any one or more of FIGs. 5-8 may be implemented in many different end products, extending from mobile devices to server systems. Referring now to FIG. 9, shown is a block diagram of a processor in accordance with another embodiment of the present invention. In the embodiment of FIG. 9, processor 900 may be a SoC including multiple domains, each of which may be controlled to operate at an independent operating voltage and operating frequency. As a specific illustrative example, processor 900 may be an Intel® Architecture Core™-based processor such as an i3, i5, i7 or another such processor available from Intel Corporation. However, other low power processors such as available from Advanced Micro Devices, Inc. (AMD) of Sunnyvale, CA, an ARM-based design from ARM Holdings, Ltd. or licensee thereof or a MIPS-based design from MIPS Technologies, Inc. of Sunnyvale, CA, or their licensees or adopters may instead be present in other embodiments such as an Apple A7 processor, a Qualcomm Snapdragon processor, or Texas Instruments OMAP processor. Such SoC may be used in a low power system such as a smartphone, tablet computer, phablet computer, Ultrabook™ computer or other portable computing device, which may incorporate a heterogeneous system architecture having a heterogeneous system architecture-based processor design.

[0080] In the high level view shown in FIG. 9, processor 900 includes a plurality of core units 910a-910n. Each core unit may include one or more processor cores, one or more cache memories and other circuitry. Each core unit 910 may support one or more instruction sets

(e.g., an x86 instruction set (with some extensions that have been added with newer versions); a MIPS instruction set; an ARM instruction set (with optional additional extensions such as NEON)) or other instruction set or combinations thereof. Note that some of the core units may be heterogeneous resources (e.g., of a different design). In addition, each such core may be coupled to a cache memory (not shown) which in an embodiment may be a shared level two (L2) cache memory. A non-volatile storage 930 may be used to store various program and other data. For example, this storage may be used to store at least portions of microcode, boot information such as a BIOS, other system software or so forth.

[0081] Each core unit 910 may also include an interface such as a bus interface unit to enable interconnection to additional circuitry of the processor. In an embodiment, each core unit 910 couples to a coherent fabric that may act as a primary cache coherent on-die interconnect that in turn couples to a memory controller 935. In turn, memory controller 935 controls communications with a memory such as a DRAM (not shown for ease of illustration in FIG. 9).

[0082] In addition to core units, additional processing engines are present within the processor, including at least one graphics unit 920 which may include one or more graphics processing units (GPUs) to perform graphics processing as well as to possibly execute general purpose operations on the graphics processor (so-called GPGPU operation). In addition, at least one image signal processor 925 may be present. Signal processor 925 may be configured to process incoming image data received from one or more capture devices, either internal to the SoC or off-chip.

[0083] Other accelerators also may be present. In the illustration of FIG. 9, a video coder 950 may perform coding operations including encoding and decoding for video information, e.g., providing hardware acceleration support for high definition video content. A display controller 955 further may be provided to accelerate display operations including providing support for internal and external displays of a system. In addition, a security processor 945 may be present to perform security operations such as secure boot operations, various cryptography operations and so forth.

[0084] Each of the units may have its power consumption controlled via a power manager 940, which may include control logic to perform the various power management techniques described herein.

[0085] In some embodiments, SoC 900 may further include a non-coherent fabric coupled to the coherent fabric to which various peripheral devices may couple. One or more interfaces 960a-960d enable communication with one or more off-chip devices. Such communications may be via a variety of communication protocols such as PCIe™, GPIO, USB, I²C, UART, MIPI, SDIO, DDR, SPI, HDMI, among other types of communication protocols. Although shown at this high level in the embodiment of FIG. 9, understand the scope of the present invention is not limited in this regard.

[0086] Referring now to FIG. 10, shown is a block diagram of a representative SoC. In the embodiment shown, SoC 1000 may be a multi-core SoC configured for low power operation to be optimized for incorporation into a smartphone or other low power device such as a tablet computer or other portable computing device. As an example, SoC 1000 may be implemented using asymmetric or different types of cores, such as combinations of higher power and/or low power cores, e.g., out-of-order cores and in-order cores. In different embodiments, these cores may be based on an Intel® Architecture™ core design or an ARM architecture design. In yet other embodiments, a mix of Intel and ARM cores may be implemented in a given SoC.

[0087] As seen in FIG. 10, SoC 1000 includes a first core domain 1010 having a plurality of first cores 1012a – 1012d. In an example, these cores may be low power cores such as in-order cores. In one embodiment these first cores may be implemented as ARM Cortex A53 cores. In turn, these cores couple to a cache memory 1015 of core domain 1010. In addition, SoC 1000 includes a second core domain 1020. In the illustration of FIG. 10, second core domain 1020 has a plurality of second cores 1022a – 1022d. In an example, these cores may be higher power-consuming cores than first cores 1012. In an embodiment, the second cores may be out-of-order cores, which may be implemented as ARM Cortex A57 cores. In turn, these cores couple to a cache memory 1025 of core domain 1020. Note that while the example shown in FIG. 10 includes 4 cores in each domain, understand that more or fewer cores may be present in a given domain in other examples.

[0088] With further reference to FIG. 10, a graphics domain 1030 also is provided, which may include one or more graphics processing units (GPUs) configured to independently execute graphics workloads, e.g., provided by one or more cores of core domains 1010 and 1020. As an example, GPU domain 1030 may be used to provide display support for a variety of screen sizes, in addition to providing graphics and display rendering operations.

[0089] As seen, the various domains couple to a coherent interconnect 1040, which in an embodiment may be a cache coherent interconnect fabric that in turn couples to an integrated memory controller 1050. Coherent interconnect 1040 may include a shared cache memory, such as an L3 cache, in some examples. In an embodiment, memory controller 1050 may be a direct memory controller to provide for multiple channels of communication with an off-chip memory, such as multiple channels of a DRAM (not shown for ease of illustration in FIG. 10).

[0090] In different examples, the number of the core domains may vary. For example, for a low power SoC suitable for incorporation into a mobile computing device, a limited number of core domains such as shown in FIG. 10 may be present. Still further, in such low power SoCs, core domain 1020 including higher power cores may have fewer numbers of such cores. For example, in one implementation two cores 1022 may be provided to enable operation at reduced power consumption levels. In addition, the different core domains may also be coupled to an interrupt controller to enable dynamic swapping of workloads between the different domains.

[0091] In yet other embodiments, a greater number of core domains, as well as additional optional IP logic may be present, in that an SoC can be scaled to higher performance (and power) levels for incorporation into other computing devices, such as desktops, servers, high performance computing systems, base stations forth. As one such example, 4 core domains each having a given number of out-of-order cores may be provided. Still further, in addition to optional GPU support (which as an example may take the form of a GPGPU), one or more accelerators to provide optimized hardware support for particular functions (e.g. web serving, network processing, switching or so forth) also may be provided. In addition, an input/output interface may be present to couple such accelerators to off-chip components.

[0092] Referring now to FIG. 11, shown is a block diagram of another example SoC. In the embodiment of FIG. 11, SoC 1100 may include various circuitry to enable high performance for multimedia applications, communications and other functions. As such, SoC 1100 is suitable for incorporation into a wide variety of portable and other devices, such as smartphones, tablet computers, smart TVs and so forth. In the example shown, SoC 1100 includes a central processor unit (CPU) domain 1110. In an embodiment, a plurality of individual processor cores may be present in CPU domain 1110. As one example, CPU domain 1110 may be a quad core processor having 4 multithreaded cores. Such processors may be homogeneous or heterogeneous processors, e.g., a mix of low power and high power processor cores.

[0093] In turn, a GPU domain 1120 is provided to perform advanced graphics processing in one or more GPUs to handle graphics and compute APIs. A DSP unit 1130 may provide one or more low power DSPs for handling low-power multimedia applications such as music playback, audio/video and so forth, in addition to advanced calculations that may occur during execution of multimedia instructions. In turn, a communication unit 1140 may include various components to provide connectivity via various wireless protocols, such as cellular communications (including 3G/4G LTE), wireless local area protocols such as BluetoothTM, IEEE 802.11, and so forth.

[0094] Still further, a multimedia processor 1150 may be used to perform capture and playback of high definition video and audio content, including processing of user gestures. A sensor unit 1160 may include a plurality of sensors and/or a sensor controller to interface to various off-chip sensors present in a given platform. An image signal processor 1170 may be provided with one or more separate ISPs to perform image processing with regard to captured content from one or more cameras of a platform, including still and video cameras.

[0095] A display processor 1180 may provide support for connection to a high definition display of a given pixel density, including the ability to wirelessly communicate content for playback on such display. Still further, a location unit 1190 may include a GPS receiver with support for multiple GPS constellations to provide applications highly accurate positioning information obtained using as such GPS receiver. Understand that while shown with this particular set of components in the example of FIG. 11, many variations and alternatives are possible.

[0096] Referring now to FIG. 12, shown is a block diagram of an example system with which embodiments can be used. As seen, system 1200 may be a smartphone or other wireless communicator. A baseband processor 1205 is configured to perform various signal processing with regard to communication signals to be transmitted from or received by the system. In turn, baseband processor 1205 is coupled to an application processor 1210, which may be a main CPU of the system to execute an OS and other system software, in addition to user applications such as many well-known social media and multimedia apps. Application processor 1210 may further be configured to perform a variety of other computing operations for the device.

[0097] In turn, application processor 1210 can couple to a user interface/display 1220, e.g., a touch screen display. In addition, application processor 1210 may couple to a memory system including a non-volatile memory, namely a flash memory 1230 and a system memory, namely a dynamic random access memory (DRAM) 1235. As further seen, application processor 1210 further couples to a capture device 1240 such as one or more image capture devices that can record video and/or still images.

[0098] Still referring to FIG. 12, a universal integrated circuit card (UICC) 1240 comprising a subscriber identity module and possibly a secure storage and cryptoprocessor is also coupled to application processor 1210. System 1200 may further include a security processor 1250 that may couple to application processor 1210. A plurality of sensors 1225 may couple to application processor 1210 to enable input of a variety of sensed information such as accelerometer and other environmental information. An audio output device 1295 may provide an interface to output sound, e.g., in the form of voice communications, played or streaming audio data and so forth.

[0099] As further illustrated, a near field communication (NFC) contactless interface 1260 is provided that communicates in a NFC near field via an NFC antenna 1265. While separate antennae are shown in FIG. 12, understand that in some implementations one antenna or a different set of antennae may be provided to enable various wireless functionality.

[0100] A power management integrated circuit (PMIC) 1215 couples to application processor 1210 to perform platform level power management. To this end, PMIC 1215 may issue power management requests to application processor 1210 to enter certain low power

states as desired. Furthermore, based on platform constraints, PMIC 1215 may also control the power level of other components of system 1200.

[0101] To enable communications to be transmitted and received, various circuitry may be coupled between baseband processor 1205 and an antenna 1290. Specifically, a radio frequency (RF) transceiver 1270 and a wireless local area network (WLAN) transceiver 1275 may be present. In general, RF transceiver 1270 may be used to receive and transmit wireless data and calls according to a given wireless communication protocol such as 3G or 4G wireless communication protocol such as in accordance with a code division multiple access (CDMA), global system for mobile communication (GSM), long term evolution (LTE) or other protocol. In addition a GPS sensor 1280 may be present. Other wireless communications such as receipt or transmission of radio signals, e.g., AM/FM and other signals may also be provided. In addition, via WLAN transceiver 1275, local wireless communications can also be realized.

[0102] Referring now to FIG. 13, shown is a block diagram of another example system with which embodiments may be used. In the illustration of FIG. 13, system 1300 may be mobile low-power system such as a tablet computer, 2:1 tablet, phablet or other convertible or standalone tablet system. As illustrated, a SoC 1310 is present and may be configured to operate as an application processor for the device.

[0103] A variety of devices may couple to SoC 1310. In the illustration shown, a memory subsystem includes a flash memory 1340 and a DRAM 1345 coupled to SoC 1310. In addition, a touch panel 1320 is coupled to the SoC 1310 to provide display capability and user input via touch, including provision of a virtual keyboard on a display of touch panel 1320. To provide wired network connectivity, SoC 1310 couples to an Ethernet interface 1330. A peripheral hub 1325 is coupled to SoC 1310 to enable interfacing with various peripheral devices, such as may be coupled to system 1300 by any of various ports or other connectors.

[0104] In addition to internal power management circuitry and functionality within SoC 1310, a PMIC 1380 is coupled to SoC 1310 to provide platform-based power management, e.g., based on whether the system is powered by a battery 1390 or AC power via an AC adapter 1395. For ease of illustration in FIG. 13, reference numeral 1380 further refers to a

voltage regulator, which may be an external first stage voltage regulator to provide a regulated voltage to SoC 1310. As described herein, embodiments enable appropriate control of operating frequency, such that current spikes that may reach voltage regulator 1380 are controlled to last no longer than an OEM-provided time duration. In addition to this power source-based power management, PMIC 1380 may further perform platform power management activities based on environmental and usage conditions. Still further, PMIC 1380 may communicate control and status information to SoC 1310 to cause various power management actions within SoC 1310.

[0105] Still referring to FIG. 13, to provide for wireless capabilities, a WLAN unit 1350 is coupled to SoC 1310 and in turn to an antenna 1355. In various implementations, WLAN unit 1350 may provide for communication according to one or more wireless protocols.

[0106] As further illustrated, a plurality of sensors 1360 may couple to SoC 1310. These sensors may include various accelerometer, environmental and other sensors, including user gesture sensors. Finally, an audio codec 1365 is coupled to SoC 1310 to provide an interface to an audio output device 1370. Of course understand that while shown with this particular implementation in FIG. 13, many variations and alternatives are possible.

[0107] Referring now to FIG. 14, shown is a block diagram of a representative computer system such as notebook, UltrabookTM or other small form factor system. A processor 1410, in one embodiment, includes a microprocessor, multi-core processor, multithreaded processor, an ultra low voltage processor, an embedded processor, or other known processing element. In the illustrated implementation, processor 1410 acts as a main processing unit and central hub for communication with many of the various components of the system 1400, and may include power management circuitry as described herein. As one example, processor 1410 is implemented as a SoC.

[0108] Processor 1410, in one embodiment, communicates with a system memory 1415. As an illustrative example, the system memory 1415 is implemented via multiple memory devices or modules to provide for a given amount of system memory.

[0109] To provide for persistent storage of information such as data, applications, one or more operating systems and so forth, a mass storage 1420 may also couple to processor 1410. In various embodiments, to enable a thinner and lighter system design as well as to improve

system responsiveness, this mass storage may be implemented via a SSD or the mass storage may primarily be implemented using a hard disk drive (HDD) with a smaller amount of SSD storage to act as a SSD cache to enable non-volatile storage of context state and other such information during power down events so that a fast power up can occur on re-initiation of system activities. Also shown in FIG. 14, a flash device 1422 may be coupled to processor 1410, e.g., via a serial peripheral interface (SPI). This flash device may provide for non-volatile storage of system software, including a basic input/output software (BIOS) as well as other firmware of the system.

[0110] Various input/output (I/O) devices may be present within system 1400. Specifically shown in the embodiment of FIG. 14 is a display 1424 which may be a high definition LCD or LED panel that further provides for a touch screen 1425. In one embodiment, display 1424 may be coupled to processor 1410 via a display interconnect that can be implemented as a high performance graphics interconnect. Touch screen 1425 may be coupled to processor 1410 via another interconnect, which in an embodiment can be an I²C interconnect. As further shown in FIG. 14, in addition to touch screen 1425, user input by way of touch can also occur via a touch pad 1430 which may be configured within the chassis and may also be coupled to the same I²C interconnect as touch screen 1425.

[0111] For perceptual computing and other purposes, various sensors may be present within the system and may be coupled to processor 1410 in different manners. Certain inertial and environmental sensors may couple to processor 1410 through a sensor hub 1440, e.g., via an I²C interconnect. In the embodiment shown in FIG. 14, these sensors may include an accelerometer 1441, an ambient light sensor (ALS) 1442, a compass 1443 and a gyroscope 1444. Other environmental sensors may include one or more thermal sensors 1446 which in some embodiments couple to processor 1410 via a system management bus (SMBus) bus.

[0112] Also seen in FIG. 14, various peripheral devices may couple to processor 1410 via a low pin count (LPC) interconnect. In the embodiment shown, various components can be coupled through an embedded controller 1435. Such components can include a keyboard 1436 (e.g., coupled via a PS2 interface), a fan 1437, and a thermal sensor 1439. In some embodiments, touch pad 1430 may also couple to EC 1435 via a PS2 interface. In addition, a security processor such as a trusted platform module (TPM) 1438 may also couple to processor 1410 via this LPC interconnect.

[0113] System 1400 can communicate with external devices in a variety of manners, including wirelessly. In the embodiment shown in FIG. 14, various wireless modules, each of which can correspond to a radio configured for a particular wireless communication protocol, are present. One manner for wireless communication in a short range such as a near field may be via a NFC unit 1445 which may communicate, in one embodiment with processor 1410 via an SMBus. Note that via this NFC unit 1445, devices in close proximity to each other can communicate.

[0114] As further seen in FIG. 14, additional wireless units can include other short range wireless engines including a WLAN unit 1450 and a Bluetooth™ unit 1452. Using WLAN unit 1450, Wi-Fi™ communications can be realized, while via Bluetooth™ unit 1452, short range Bluetooth™ communications can occur. These units may communicate with processor 1410 via a given link.

[0115] In addition, wireless wide area communications, e.g., according to a cellular or other wireless wide area protocol, can occur via a WWAN unit 1456 which in turn may couple to a subscriber identity module (SIM) 1457. In addition, to enable receipt and use of location information, a GPS module 1455 may also be present. Note that in the embodiment shown in FIG. 14, WWAN unit 1456 and an integrated capture device such as a camera module 1454 may communicate via a given link.

[0116] To provide for audio inputs and outputs, an audio processor can be implemented via a digital signal processor (DSP) 1460, which may couple to processor 1410 via a high definition audio (HDA) link. Similarly, DSP 1460 may communicate with an integrated coder/decoder (CODEC) and amplifier 1462 that in turn may couple to output speakers 1463 which may be implemented within the chassis. Similarly, amplifier and CODEC 1462 can be coupled to receive audio inputs from a microphone 1465 which in an embodiment can be implemented via dual array microphones (such as a digital microphone array) to provide for high quality audio inputs to enable voice-activated control of various operations within the system. Note also that audio outputs can be provided from amplifier/CODEC 1462 to a headphone jack 1464.

[0117] As further illustrated in FIG. 14, for a portable computing device that is battery operated, a battery 1402 couples to an external voltage regulator 1404, which in turn may

couple to processor 1410 via a capacitor network 1406. Depending upon the length and duration of current spikes incurred in processor 1410, such current spikes can be handled either internally to processor, by capacitor network 1406 or may sink to voltage regulator 1404 (and/or battery 1402). Using an embodiment of the present invention, with a dynamically determined dynamic capacitance for a particular system 1400, processor operation may occur at or below a determined peak operating frequency to ensure that a maximum current constraint is not violated, while at the same time ensuring that any current spikes extending beyond processor 1410 last no longer than the configured length. Although shown with these particular components in the embodiment of FIG. 14, understand the scope of the present invention is not limited in this regard.

[0118] Embodiments may be implemented in many different system types. Referring now to FIG. 15, shown is a block diagram of a system in accordance with an embodiment of the present invention. As shown in FIG. 15, multiprocessor system 1500 is a point-to-point interconnect system, and includes a first processor 1570 and a second processor 1580 coupled via a point-to-point interconnect 1550. As shown in FIG. 15, each of processors 1570 and 1580 may be multicore processors, including first and second processor cores (i.e., processor cores 1574a and 1574b and processor cores 1584a and 1584b), although potentially many more cores may be present in the processors. Each of the processors can include a PCU or other power management logic to perform processor-based power management as described herein.

[0119] Still referring to FIG. 15, first processor 1570 further includes a memory controller hub (MCH) 1572 and point-to-point (P-P) interfaces 1576 and 1578. Similarly, second processor 1580 includes a MCH 1582 and P-P interfaces 1586 and 1588. As shown in FIG. 15, MCH's 1572 and 1582 couple the processors to respective memories, namely a memory 1532 and a memory 1534, which may be portions of system memory (e.g., DRAM) locally attached to the respective processors. First processor 1570 and second processor 1580 may be coupled to a chipset 1590 via P-P interconnects 1562 and 1564, respectively. As shown in FIG. 15, chipset 1590 includes P-P interfaces 1594 and 1598.

[0120] Furthermore, chipset 1590 includes an interface 1592 to couple chipset 1590 with a high performance graphics engine 1538, by a P-P interconnect 1539. In turn, chipset 1590 may be coupled to a first bus 1516 via an interface 1596. As shown in FIG. 15, various

input/output (I/O) devices 1514 may be coupled to first bus 1516, along with a bus bridge 1518 which couples first bus 1516 to a second bus 1520. Various devices may be coupled to second bus 1520 including, for example, a keyboard/mouse 1522, communication devices 1526 and a data storage unit 1528 such as a disk drive or other mass storage device which may include code 1530, in one embodiment. Further, an audio I/O 1524 may be coupled to second bus 1520. Embodiments can be incorporated into other types of systems including mobile devices such as a smart cellular telephone, tablet computer, netbook, Ultrabook™, or so forth.

[0121] One or more aspects of at least one embodiment may be implemented by representative code stored on a machine-readable medium which represents and/or defines logic within an integrated circuit such as a processor. For example, the machine-readable medium may include instructions which represent various logic within the processor. When read by a machine, the instructions may cause the machine to fabricate the logic to perform the techniques described herein. Such representations, known as “IP cores,” are reusable units of logic for an integrated circuit that may be stored on a tangible, machine-readable medium as a hardware model that describes the structure of the integrated circuit. The hardware model may be supplied to various customers or manufacturing facilities, which load the hardware model on fabrication machines that manufacture the integrated circuit. The integrated circuit may be fabricated such that the circuit performs operations described in association with any of the embodiments described herein.

[0122] FIG. 16 is a block diagram illustrating an IP core development system 1600 that may be used to manufacture an integrated circuit to perform operations according to an embodiment. The IP core development system 1600 may be used to generate modular, reusable designs that can be incorporated into a larger design or used to construct an entire integrated circuit (e.g., an SoC integrated circuit). A design facility 1630 can generate a software simulation 1610 of an IP core design in a high level programming language (e.g., C/C++). The software simulation 1610 can be used to design, test, and verify the behavior of the IP core. A register transfer level (RTL) design can then be created or synthesized from the simulation model. The RTL design 1615 is an abstraction of the behavior of the integrated circuit that models the flow of digital signals between hardware registers, including the associated logic performed using the modeled digital signals. In addition to an

RTL design 1615, lower-level designs at the logic level or transistor level may also be created, designed, or synthesized. Thus, the particular details of the initial design and simulation may vary.

[0123] The RTL design 1615 or equivalent may be further synthesized by the design facility into a hardware model 1620, which may be in a hardware description language (HDL), or some other representation of physical design data. The HDL may be further simulated or tested to verify the IP core design. The IP core design can be stored for delivery to a third party fabrication facility 1665 using non-volatile memory 1640 (e.g., hard disk, flash memory, or any non-volatile storage medium). Alternately, the IP core design may be transmitted (e.g., via the Internet) over a wired connection 1650 or wireless connection 1660. The fabrication facility 1665 may then fabricate an integrated circuit that is based at least in part on the IP core design. The fabricated integrated circuit can be configured to perform operations in accordance with at least one embodiment described herein.

[0124] Note that peak current drawn by a processor is a sum of leakage current and dynamic current. Leakage current is a function of voltage and temperature, such that for a given frequency, voltage and temperature of operation, leakage current remains constant. Dynamic current may be determined according to:

$$I_{ccdyn} = C_{dyn} * V^2 * f \text{ [EQ. 1]}$$

where V is the operating voltage of the processor, f is the operating frequency of the processor, and C_{dyn} is the dynamic capacitance of the processor.

[0125] In an embodiment, C_{dyn} may be a measure of the number of gates in the processor that can transition from 0 to 1 or 1 to 0. The number of gates that can transition their state is a function of time. The architecture of the processor determines what percentage of the gates can transition state in a given time instant. Furthermore, the average percentage of gates that can continuously transition state over different intervals of time varies.

[0126] Referring now to FIG. 17, shown is a graphical illustration of a dynamic capacitance of a processor over various time intervals in accordance with an embodiment. As illustrated in FIG. 17, a plurality of evaluation points A-E are present, where each point has a dynamic capacitance and associated time duration. Such information may be obtained during manufacturing characterization, such as responsive to execution of test workloads during

manufacturing testing. As illustrated in FIG. 17, illustration 1700 forms a curve 1710 from these different characterization points. Such curve 1710 may be used to enable a processor to perform an interpolation or other curve fitting process to identify an appropriate dynamic capacitance for a particular processor, based on a configured time duration for which a current spike is to be exposed to the voltage regulator. Understand while the illustration in FIG. 17 includes five characterization points, particular embodiments may store more or fewer such characterization points as separate entries within a non-volatile storage of a processor.

[0127] As seen in FIG. 17, it is feasible that a large percentage of gates in a processor toggle state instantaneously, thus a worst case instantaneous C_{dyn} exists at point A. Over increasing time windows however the percentage of gates that toggles concurrently drops. This is because of inherent architectural limitations in the processor. For example, there may be data dependencies where the processor pipeline waits for a load from cache or memory to complete before peak instruction execution can be resumed. Or instructions may not operate over a worst case set of operands that cause the most gates in the node to toggle every clock cycle, among other reasons.

[0128] Also seen in FIG. 17, as peak sustainable C_{dyn} varies over different time intervals, so does peak sustainable current. Over longer intervals of time, the peak sustainable C_{dyn} is lower than over smaller (or instantaneous) intervals of time. And correspondingly, peak current draws over longer intervals of time are lower than over shorter intervals of time. Given a particular C_{dyn} value, processors manage to a programmed I_{ccMax} limit by estimating peak current at any point in time at a given frequency and then reducing frequency until a point is identified where estimated peak current is lower than a platform advertised I_{ccMax} capability.

[0129] Using an embodiment, C_{dyn} may be characterized during a manufacturing process using empirical measurements and programmed into a non-volatile memory of the processor. With reference to FIG. 17, for example, multiple (C_{dyn} , Time) tuples can be written into a non-volatile storage. In contrast, conventional processors only characterize an instantaneous peak C_{dyn} and use this single fixed value to determine operating frequency at a given I_{ccMax} constraint.

[0130] Depending on the capacitance provisioned in an OEM design, current spikes over small intervals of time are filtered out by the capacitive network and the first stage voltage regulator (VR) is exposed to unfiltered current. For a given VR capability, an OEM can allow for running at higher current levels if more capacitance is included in a platform to handle shorter duration current spikes and exposing the VR to only the unfiltered sustained, longer duration current spikes. However, this comes with a cost and area tradeoff as mentioned previously. Embodiments enable a technique to exploit this additional capacitance in the power delivery network. In other words, increased operation can be realized using an embodiment, in contrast to fixing a processor to operate at a fixed and lower operating frequency for a given I_{ccMax} constraint, irrespective of the amount of capacitance on the platform. In this way, embodiments may improve processor performance by controlling the processor to run exactly to the limits of a particular OEM design. By exposing a register interface, an OEM/platform designer can specify a peak current constraint and a time duration over which the processor is to maintain this constraint.

[0131] When a processor is integrated into a given OEM platform design, the OEM decides how much capacitance the design can support based on performance, cost and other considerations, and characterizes the time durations over which the capacitance network will filter out current spikes. Using an embodiment, the OEM may program the peak current delivery capability of the voltage regulator and the time duration over which current spikes will pass unfiltered through the capacitive network and reach the voltage regulator, e.g., by incorporation of this information into system firmware.

[0132] Given this input, the processor may be configured to identify a peak C_{dyn} that the processor can sustain over the programed time window, e.g., by an interpolation process using the stored values. Thereafter, during dynamic operation using this interpolated C_{dyn} , the processor can determine the peak frequency at which it can operate given a platform I_{ccMax} constraint. Embodiments thus allow the processor to operate at a higher frequency and deliver higher performance, exploiting the fact that some of the current transients will be filtered by the capacitive network.

[0133] Referring now to FIG. 18, shown is a flow diagram of a method in accordance with an embodiment of the present invention. Method 1800 may be performed during manufacture of a processor. As shown in FIG. 18, method 1800 begins by providing a

configuration register (block 1810). Such configuration register may be one of many different configuration registers of a processor. This configuration register (or portion) may be designated for specification of an IccMax and a corresponding time duration. More specifically, this configuration register may be exposed to an OEM and/or other platform designer to enable specifying a particular IccMax and corresponding time duration for a processor that is to be incorporated into a given a platform design for which power delivery solutions and platform capacitance is known. Table 1 below is an example structure of a configuration register to store this information. Understand while shown with this particular arrangement, other examples are possible.

Table 1

Bits	Field Name	Description
Bits[9:0]	IccMax	Peak current delivery capability of the Voltage Regulator (VR). (Units in Amperes)
Bits[19:10]	Time_Window	Time window over which current spikes are exposed to the VR. (Units in micro-seconds)

[0134] Next, control passes to block 1820 where a Cdyn is characterized. More specifically, this Cdyn may be characterized over a number of different time intervals. As an example, one or more test workloads can be executed on the processor (during manufacturing testing, e.g.,) to characterize Cdyn over different time intervals. As such, multiple Cdyn values with corresponding time intervals may be obtained. Such information can be stored into a non-volatile storage of the processor. Although the scope of the present invention is not limited in this regard, in embodiments the non-volatile storage may be fuse storage, flash memory or so forth. For example, multiple entries in such storage may be written, with each entry including a tuple of Cdyn value and a corresponding time interval. As such, rather than providing a single worst case Cdyn value for a processor, multiple values are provided. In addition, instead of providing this single worst case Cdyn value with no time information, each written Cdyn value has a time interval associated therewith. From the information

stored in these multiple entries, a curve can be established for C_{dyn} with respect to different time intervals.

[0135] At this point, a processor is fully configured during manufacturing and is available for sale to different customers such as different OEMs and/or other platform designers. Understand that with the configured information, which is provided on a per processor basis according to actual manufacturing testing performed on a specific processor, fine-grained control can be realized. In this way, improved performance may occur, such as by way of enabling an increased frequency. By using an embodiment, a processor may be able to execute all the way up to a given Icc_{max} value.

[0136] Referring now to FIG. 19, shown is a flow diagram of a method for designing a computing platform in accordance with an embodiment. As shown in FIG. 19, method 1900 may be performed, e.g., by an OEM or other platform manufacturer. More specifically, method 1900 begins at block 1910 by designing a given platform. As is well understood, platform design encompasses many steps and includes identifying a set of functionality for inclusion in the platform, along with corresponding components to effect this functionality. Note that the identification of components includes identification of one or more power delivery components, such as one or more external voltage regulators and one or more power sources, such as a particular battery source for portable devices. In addition, understand that power delivery components may also include a set of power delivery capacitors, which may include one or more capacitors adapted between a battery and/or other power source and one or more voltage regulators, as well as one or more capacitors adapted between such one or more voltage regulators and processor (and potentially other components of a system). After identification of components, a platform design is made and used to enable manufacture of the system.

[0137] With this design completed, the OEM knows the total amount of available off-chip capacitance that can be used to filter out current spikes from reaching at least certain power delivery components, including one or more external voltage regulators and one or more external power sources. With this information, method 1900 proceeds to block 1920 where a peak current capability of the platform and corresponding time duration over which this peak current capability is to be controlled can be established. In an embodiment, this peak current capability corresponds to an Icc_{Max} value, which thus identifies a maximum current spike

that is allowed to occur during operation of a processor or other SoC. In addition, the time duration identifies a length of time that this current spike may be exposed to one or more power delivery components, such as one or more external voltage regulators. This established peak current capability and corresponding time duration can be provided for storage in the given one or more configuration registers that are to be exposed to the OEM/platform designer. In one embodiment, this information can be provisioned into the configuration storage by inclusion in platform configuration information within a basic input/output system (BIOS) or other platform firmware, to enable its storage within the configuration register on system power up. Understand while shown at this high level in the embodiment of FIG. 19, many variations and alternatives are possible.

[0138] Referring now to FIG. 20, shown is a flow diagram of a method in accordance with yet another embodiment of the present invention. As shown in FIG. 20, method 2000 is a method for dynamic control of current consumption in a processor during operation, in accordance with an embodiment of the present invention. Method 2000 begins by receiving within the processor a given peak current capability and corresponding time duration of a platform and storing this information in a configuration register (block 2010). Note that this receipt and storage may occur during boot up of a platform including the processor. As an example, this storage may occur under BIOS control in which this information is written with other configuration information into one or more appropriate configuration storages. In some cases, multiple pairs of information can be provided, such that different values can be used, e.g., depending on whether a system is powered by AC power or a battery. It is also possible for updated values to be provided, e.g., in situations where platform modifications lead to changes to capacitance and/or power delivery capabilities.

[0139] Still with reference to FIG. 20, thereafter control passes to block 2020. At block 2020 one or more entries of a non-volatile storage can be accessed. More specifically, a table or other collection of corresponding C_{dyn} and time duration values may be stored in entries of the non-volatile storage, which were written into the processor during manufacturing operations, as discussed above.

[0140] Control next passes to block 2030 where a C_{dyn} value can be determined. More specifically, based on the one or more C_{dyn} /time interval values of the multiple entries, along with the given OEM-provided time duration, a particular C_{dyn} value for the processor is

established. This particular C_{dyn} value may be a selected one of the entries of the table, if an entry is present for the provided time duration. Otherwise an interpolation or other curve fitting process may be performed to generate a C_{dyn} value using the stored information. Note that this C_{dyn} value can be stored in a location accessible to, e.g., a power controller such as a PCU of the processor.

[0141] Thereafter control passes to block 2040 where, during dynamic processor operation, this C_{dyn} value can be used to ensure that the processor operates at levels at or below a configured I_{cc}Max value. In one embodiment, the determination of the peak frequency can be according to EQ. 1, where frequency is the unknown value (and I_{cc}Max and C_{dyn}, and voltage are known). By providing this particular C_{dyn} value for a given processor as manufactured (accommodating for various process considerations and/or other manufacturing tolerances of the given processor), the processor may operate at a potentially higher operating frequency (and potentially higher operating voltage), rather than use of a conventional worst case C_{dyn} value. Understand while shown at this high level in the embodiment of FIG. 20, many variations and alternatives are possible.

[0142] Thus with a manufacturing time characterization and programming of a C_{dyn}/time interval relationship into a processor, an OEM or other platform designer may be allowed to specify a peak current limit and time duration over which to control to the current limit. With this information, the processor may be configured to estimate the peak C_{dyn} it can sustain over the programmed time interval and then use this estimated C_{dyn} to dynamically determine peak frequency of operation. As such, by estimating peak current as a function of different time intervals and taking advantage of current filtering that occurs within a specific capacitive network, improved processor frequency and performance may be realized.

[0143] An interface in accordance with an embodiment may be used to configure time durations for peak current or power control. A given processor may be configured to thus operate at different operating points when changes occur to a capacitance of a power delivery network.

[0144] The following Examples pertain to further embodiments.

[0145] In one example, a processor includes: at least one core to execute instructions; a power controller coupled to the at least one core to control power consumption of the

processor; and a storage to store a plurality of entries to associate a dynamic capacitance with a time duration for which a current spike is to be exposed to a power delivery component.

[0146] In an example, the processor further comprises a configuration storage to store a peak current delivery capability of the power delivery component and a corresponding time duration, the configuration storage programmable by an OEM.

[0147] In an example, the configuration storage is to store the peak current delivery capability and the corresponding time duration after a reset of the processor and update at least one of the peak current delivery capability and the corresponding time duration responsive to a change to a system including the processor.

[0148] In an example, the power delivery component comprises an external voltage regulator and the corresponding time duration is based at least in part on an external capacitance of a system including the processor.

[0149] In an example, the power controller is to determine a sustainable peak dynamic capacitance based on the corresponding time duration and one or more of the plurality of entries of the storage.

[0150] In an example, the power controller is to determine the sustainable peak dynamic capacitance based at least in part on an interpolation between at least two of the plurality of entries of the storage.

[0151] In an example, the power controller is to determine a peak frequency for the processor based at least in part on the sustainable peak dynamic capacitance.

[0152] In an example, the power controller is to determine the peak frequency further based on a peak current delivery capability of the power delivery component.

[0153] In an example, the storage comprises a non-volatile storage to be written during manufacture of the processor.

[0154] In an example, the plurality of entries are based on manufacturing characterization of the processor.

[0155] Note that the above processor can be implemented using various means.

[0156] In an example, the processor comprises a SoC incorporated in a user equipment touch-enabled device.

[0157] In another example, a system comprises a display and a memory, and includes the processor of one or more of the above examples.

[0158] In yet another example, a method includes: obtaining a maximum current for a current spike to be incurred by a processor and a time duration for which the current spike is to be exposed to a voltage regulator; accessing a storage having a plurality of entries to associate a dynamic capacitance with a time duration for which the current spike is to be exposed to a power delivery network (including the voltage regulator); and determining a sustainable peak dynamic capacitance for the processor based at least in part on the obtained time duration and the time duration of at least two of the plurality of entries.

[0159] In an example, the method further comprises determining the sustainable peak dynamic capacitance according to an interpolation of the at least two of the plurality of entries.

[0160] In an example, the method further comprises obtaining the maximum current and the time duration from a BIOS, where the maximum current and the time duration are included in the BIOS by an OEM of a system including the processor.

[0161] In an example, the method further comprises determining a peak frequency for the processor based at least in part on the sustainable peak dynamic capacitance.

[0162] In an example, the method further comprises determining the peak frequency further based on the maximum current.

[0163] In an example, accessing the storage comprises accessing a non-volatile storage written during manufacture of the processor.

[0164] In another example, a computer readable medium including instructions is to perform the method of any of the above examples.

[0165] In another example, a computer readable medium including data is to be used by at least one machine to fabricate at least one integrated circuit to perform the method of any one of the above examples.

[0166] In another example, an apparatus comprises means for performing the method of any one of the above examples.

[0167] In yet another example, a system includes a processor having at least one core, a non-volatile storage to store a plurality of entries having characterization information to associate a dynamic capacitance with a time duration for which a current spike is to be exposed to a voltage regulator, a configuration storage to store a maximum current of the current spike and a platform time duration for which the maximum current is to be exposed to the voltage regulator, and a power controller to determine a particular dynamic capacitance for the processor based at least in part on the platform time duration and the characterization information of one or more of the plurality of entries. The system may further include one or more voltage regulators coupled to the processor and a platform capacitance coupled to the processor.

[0168] In an example, the power controller is to dynamically determine a frequency based at least in part on the particular dynamic capacitance and the maximum current.

[0169] In an example, the configuration storage is to store the maximum current and the platform time duration after a reset of the processor and update at least one of the maximum current and the platform time duration responsive to a change to at least one of the voltage regulator and the platform capacitance.

[0170] In an example, the power controller is to determine the particular dynamic capacitance based at least in part on an interpolation between the characterization information of at least two of the plurality of entries of the non-volatile storage.

[0171] In a still further example, an apparatus comprises: at least one core means for executing instructions; control means for controlling power consumption of the apparatus; and means for storing a plurality of entries to associate a dynamic capacitance with a time duration for which a current spike is to be exposed to a power delivery means.

[0172] In an example, the apparatus further comprises configuration storage means for storing a peak current delivery capability of the power delivery means and a corresponding time duration, the configuration storage means programmable by an OEM.

[0173] In an example, the configuration storage means is to store the peak current delivery capability and the corresponding time duration after a reset of the apparatus and update at least one of the peak current delivery capability and the corresponding time duration responsive to a change to a system including the apparatus.

[0174] In an example, the control means is to determine a sustainable peak dynamic capacitance based on the corresponding time duration and one or more of the plurality of entries of the means for storing.

[0175] Understand that various combinations of the above examples are possible.

[0176] Note that the terms “circuit” and “circuitry” are used interchangeably herein. As used herein, these terms and the term “logic” are used to refer to alone or in any combination, analog circuitry, digital circuitry, hard wired circuitry, programmable circuitry, processor circuitry, microcontroller circuitry, hardware logic circuitry, state machine circuitry and/or any other type of physical hardware component. Embodiments may be used in many different types of systems. For example, in one embodiment a communication device can be arranged to perform the various methods and techniques described herein. Of course, the scope of the present invention is not limited to a communication device, and instead other embodiments can be directed to other types of apparatus for processing instructions, or one or more machine readable media including instructions that in response to being executed on a computing device, cause the device to carry out one or more of the methods and techniques described herein.

[0177] Embodiments may be implemented in code and may be stored on a non-transitory storage medium having stored thereon instructions which can be used to program a system to perform the instructions. Embodiments also may be implemented in data and may be stored on a non-transitory storage medium, which if used by at least one machine, causes the at least one machine to fabricate at least one integrated circuit to perform one or more operations. Still further embodiments may be implemented in a computer readable storage medium including information that, when manufactured into a SoC or other processor, is to configure the SoC or other processor to perform one or more operations. The storage medium may include, but is not limited to, any type of disk including floppy disks, optical disks, solid state drives (SSDs), compact disk read-only memories (CD-ROMs), compact disk rewritables

(CD-RWs), and magneto-optical disks, semiconductor devices such as read-only memories (ROMs), random access memories (RAMs) such as dynamic random access memories (DRAMs), static random access memories (SRAMs), erasable programmable read-only memories (EPROMs), flash memories, electrically erasable programmable read-only memories (EEPROMs), magnetic or optical cards, or any other type of media suitable for storing electronic instructions.

[0178] While the present invention has been described with respect to a limited number of embodiments, those skilled in the art will appreciate numerous modifications and variations therefrom. It is intended that the appended claims cover all such modifications and variations as fall within the true spirit and scope of this present invention.

What is claimed is:

- 1 1. A processor comprising:
2 at least one core to execute instructions;
3 a power controller coupled to the at least one core to control power consumption of
4 the processor; and
5 a storage to store a plurality of entries to associate a dynamic capacitance with a time
6 duration for which a current spike is to be exposed to a power delivery component.
- 1 2. The processor of claim 1, further comprising a configuration storage to store a peak
2 current delivery capability of the power delivery component and a corresponding time
3 duration, the configuration storage programmable by an original equipment manufacturer
4 (OEM).
- 1 3. The processor of claim 2, wherein the configuration storage is to store the peak
2 current delivery capability and the corresponding time duration after a reset of the processor
3 and update at least one of the peak current delivery capability and the corresponding time
4 duration responsive to a change to a system including the processor.
- 1 4. The processor of claim 2, wherein the power delivery component comprises an
2 external voltage regulator and the corresponding time duration is based at least in part on an
3 external capacitance of a system including the processor.
- 1 5. The processor of claim 1, wherein the power controller is to determine a sustainable
2 peak dynamic capacitance based on the corresponding time duration and one or more of the
3 plurality of entries of the storage.
- 1 6. The processor of claim 5, wherein the power controller is to determine the sustainable
2 peak dynamic capacitance based at least in part on an interpolation between at least two of
3 the plurality of entries of the storage.

1 7. The processor of claim 5, wherein the power controller is to determine a peak
2 frequency for the processor based at least in part on the sustainable peak dynamic
3 capacitance.

1 8. The processor of claim 7, wherein the power controller is to determine the peak
2 frequency further based on a peak current delivery capability of the power delivery
3 component.

1 9. The processor of claim 1, wherein the storage comprises a non-volatile storage to be
2 written during manufacture of the processor.

1 10. The processor of claim 9, wherein the plurality of entries are based on manufacturing
2 characterization of the processor.

1 11. A method comprising:
2 obtaining a maximum current for a current spike to be incurred by a processor and a
3 time duration for which the current spike is to be exposed to a voltage regulator;
4 accessing a storage having a plurality of entries to associate a dynamic capacitance
5 with a time duration for which the current spike is to be exposed to a power delivery network,
6 the power delivery network including the voltage regulator; and
7 determining a sustainable peak dynamic capacitance for the processor based at least in
8 part on the obtained time duration and the time duration of at least two of the plurality of
9 entries.

1 12. The method of claim 11, further comprising determining the sustainable peak
2 dynamic capacitance according to an interpolation of the at least two of the plurality of
3 entries.

1 13. The method of claim 12, further comprising obtaining the maximum current and the
2 time duration from a basic input/output system, wherein the maximum current and the time
3 duration are included in the basic input/output system by an original equipment manufacturer
4 of a system including the processor.

1 14. The method of claim 11, further comprising determining a peak frequency for the
2 processor based at least in part on the sustainable peak dynamic capacitance.

1 15. The method of claim 14, further comprising determining the peak frequency further
2 based on the maximum current.

1 16. The method of claim 11, wherein accessing the storage comprises accessing a non-
2 volatile storage written during manufacture of the processor.

1 17. A computer-readable storage medium including computer-readable instructions, when
2 executed, to implement a method as claimed in any one of claims 11 to 16.

1 18. An apparatus comprising means to perform a method as claimed in any one of claims
2 11 to 16.

1 19. A system comprising:
2 a processor having at least one core, a non-volatile storage to store a plurality of
3 entries having characterization information to associate a dynamic capacitance with a time
4 duration for which a current spike is to be exposed to a voltage regulator, a configuration
5 storage to store a maximum current of the current spike and a platform time duration for
6 which the maximum current is to be exposed to the voltage regulator, and a power controller
7 to determine a particular dynamic capacitance for the processor based at least in part on the
8 platform time duration and the characterization information of one or more of the plurality of
9 entries;
10 the voltage regulator coupled to the processor; and
11 a platform capacitance coupled to the processor.

1 20. The system of claim 19, wherein the power controller is to dynamically determine a
2 frequency based at least in part on the particular dynamic capacitance and the maximum
3 current.

1 21. The system of claim 19, wherein the configuration storage is to store the maximum
2 current and the platform time duration after a reset of the processor and update at least one of
3 the maximum current and the platform time duration responsive to a change to at least one of
4 the voltage regulator and the platform capacitance.

1 22. The system of claim 19, wherein the power controller is to determine the particular
2 dynamic capacitance based at least in part on an interpolation between the characterization
3 information of at least two of the plurality of entries of the non-volatile storage.

1 23. An apparatus comprising:
2 at least one core means for executing instructions;
3 control means for controlling power consumption of the apparatus; and
4 means for storing a plurality of entries to associate a dynamic capacitance with a time
5 duration for which a current spike is to be exposed to a power delivery means.

1 24. The apparatus of claim 23, further comprising configuration storage means for storing
2 a peak current delivery capability of the power delivery means and a corresponding time
3 duration, the configuration storage means programmable by an original equipment
4 manufacturer (OEM).

1 25. The apparatus of claim 24, wherein the configuration storage means is to store the
2 peak current delivery capability and the corresponding time duration after a reset of the
3 apparatus and update at least one of the peak current delivery capability and the
4 corresponding time duration responsive to a change to a system including the apparatus.

1/18

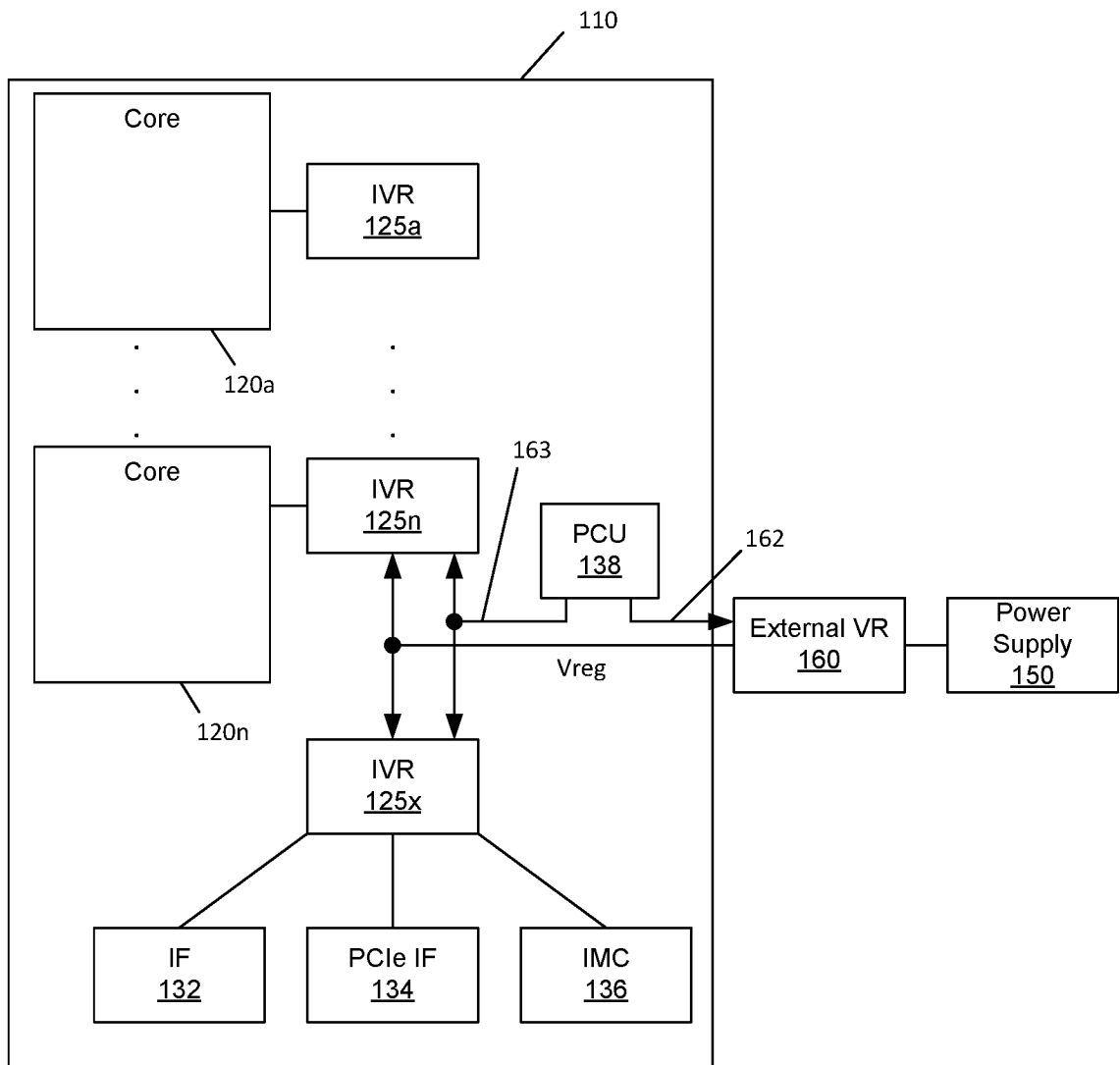
100

FIG. 1

200

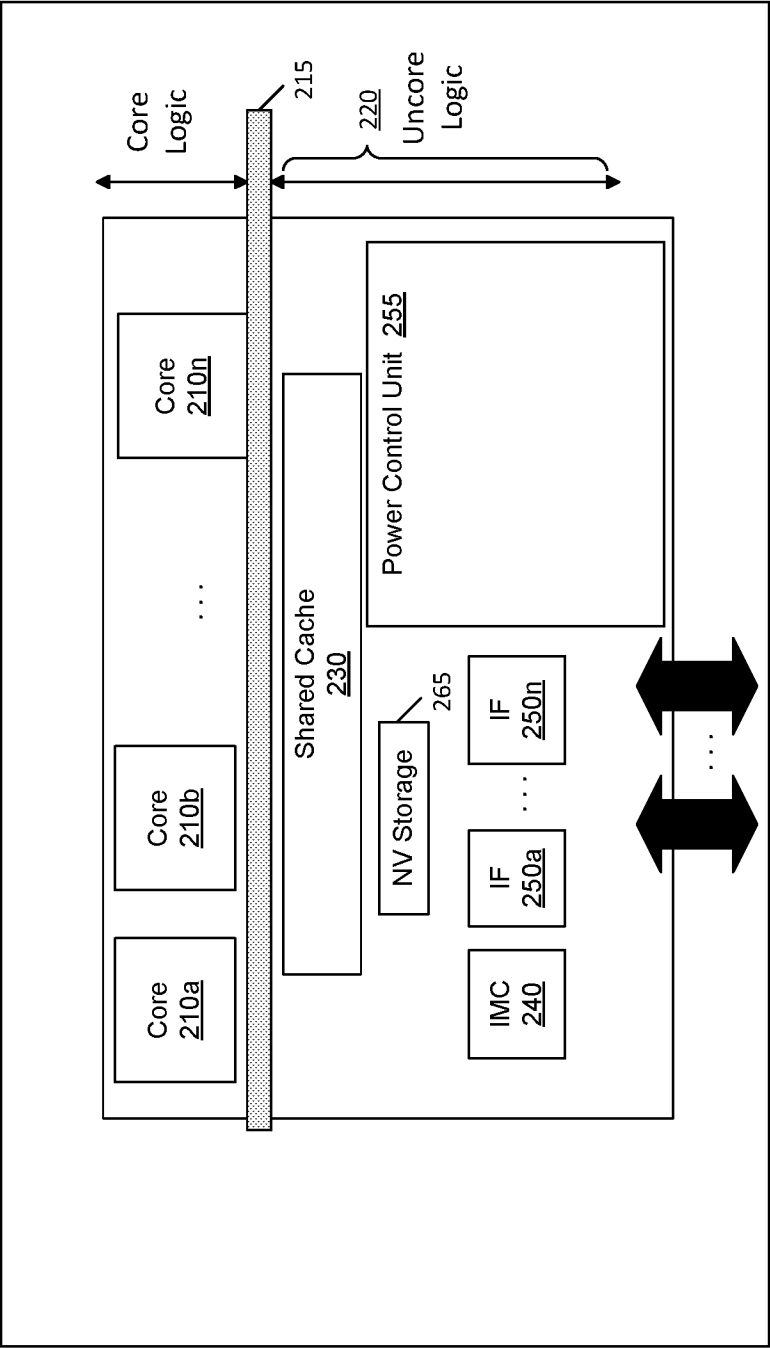


FIG. 2

300

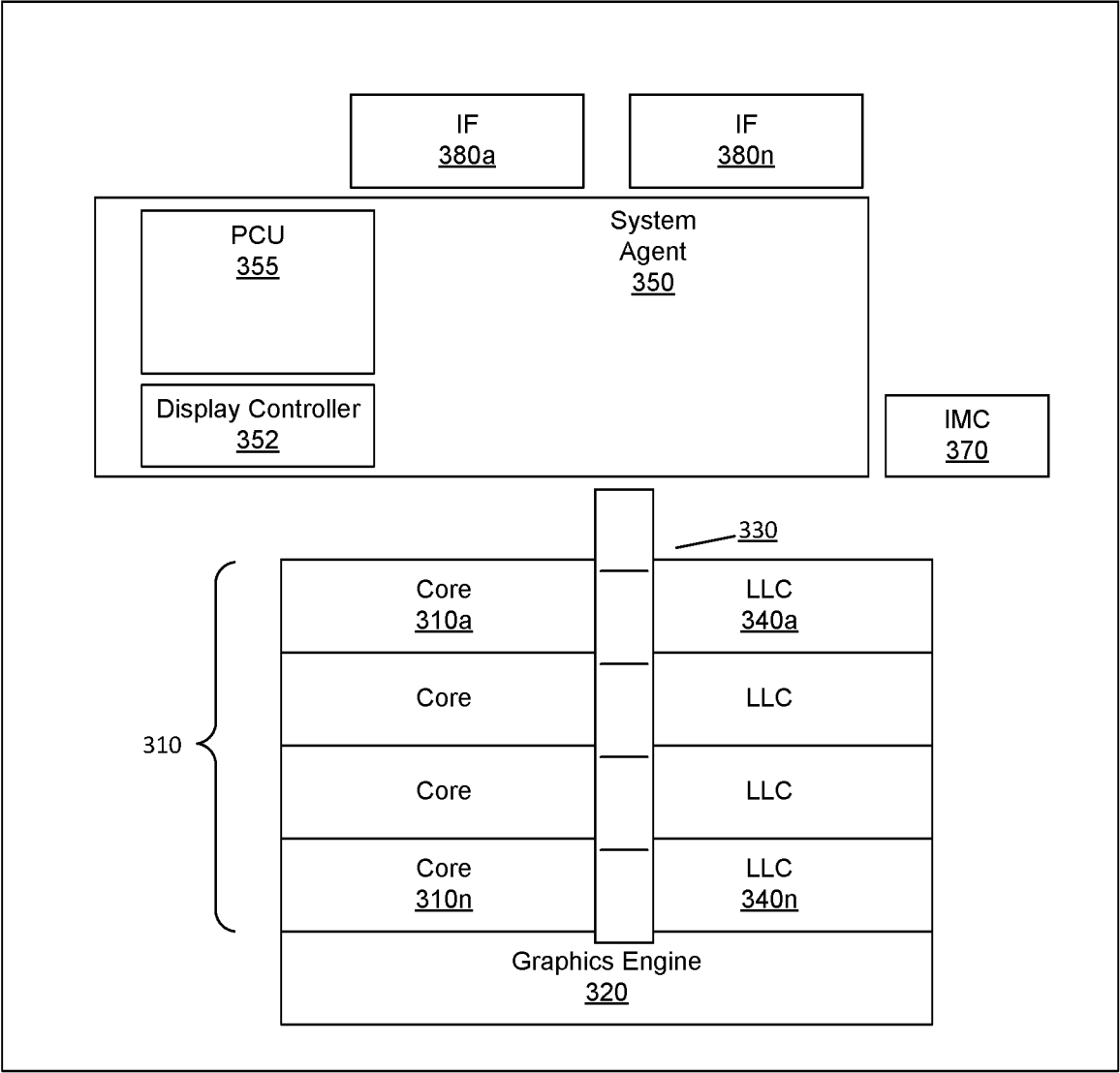


FIG. 3

4/18

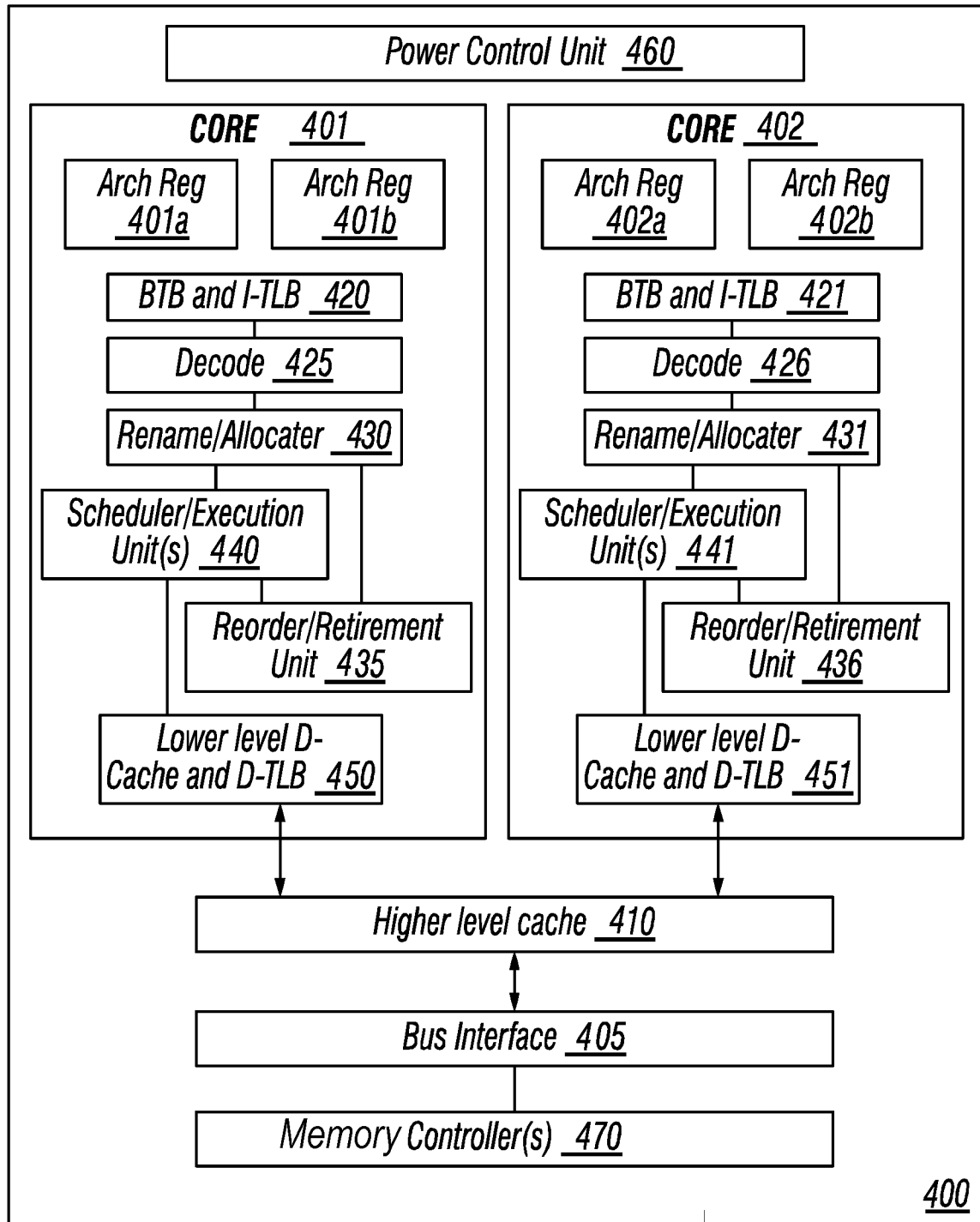


FIG. 4

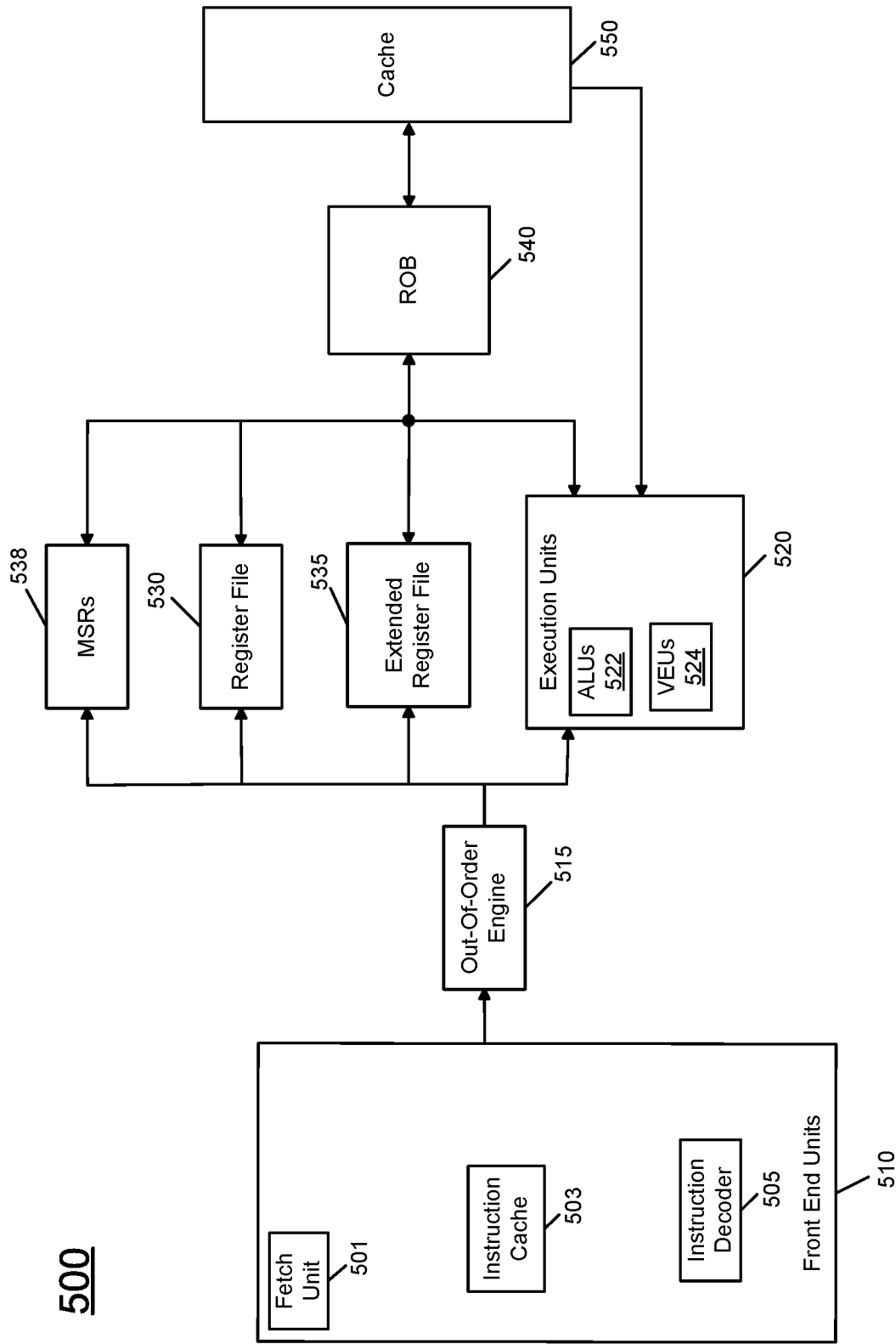


FIG. 5

6/18

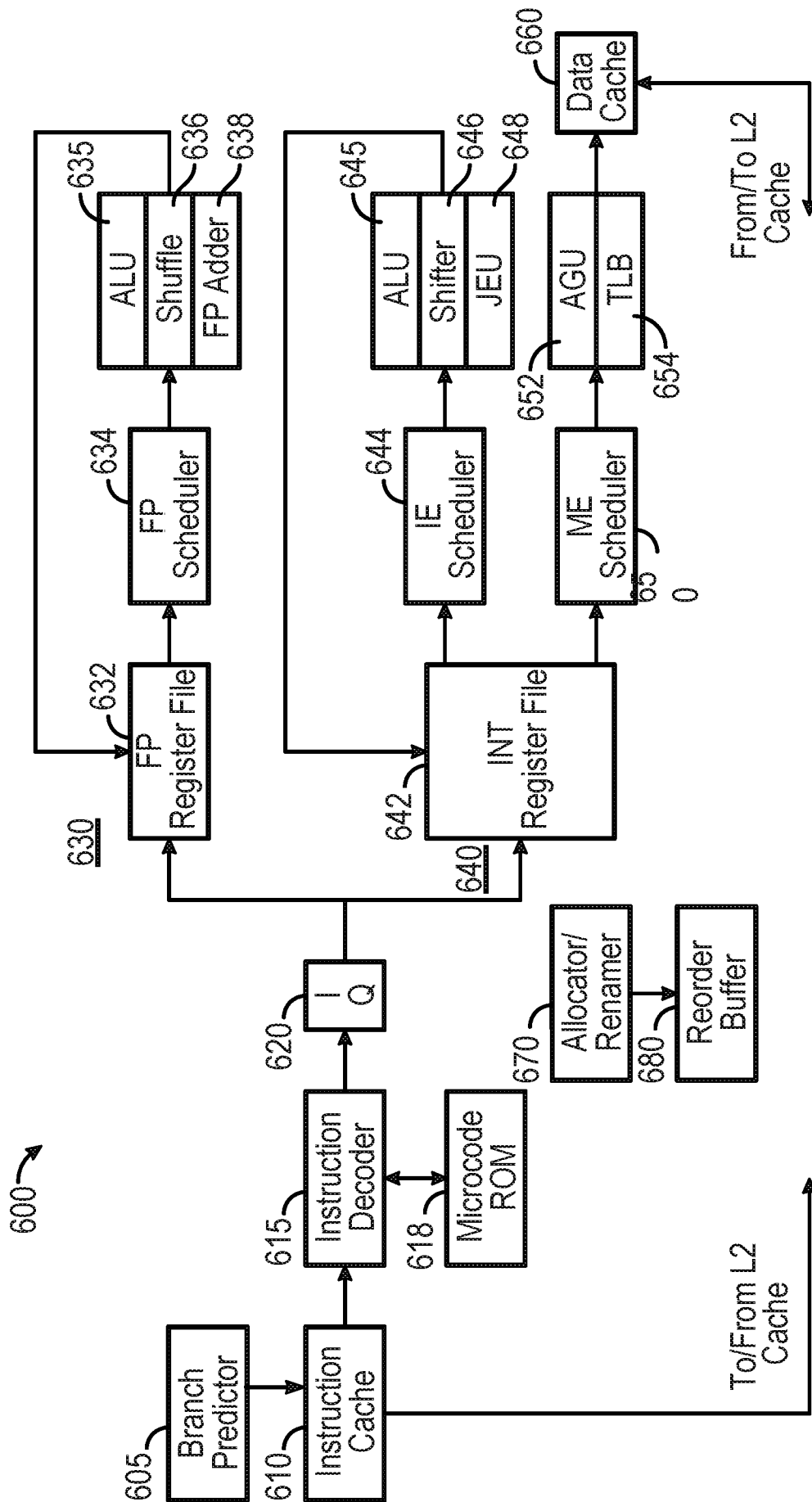
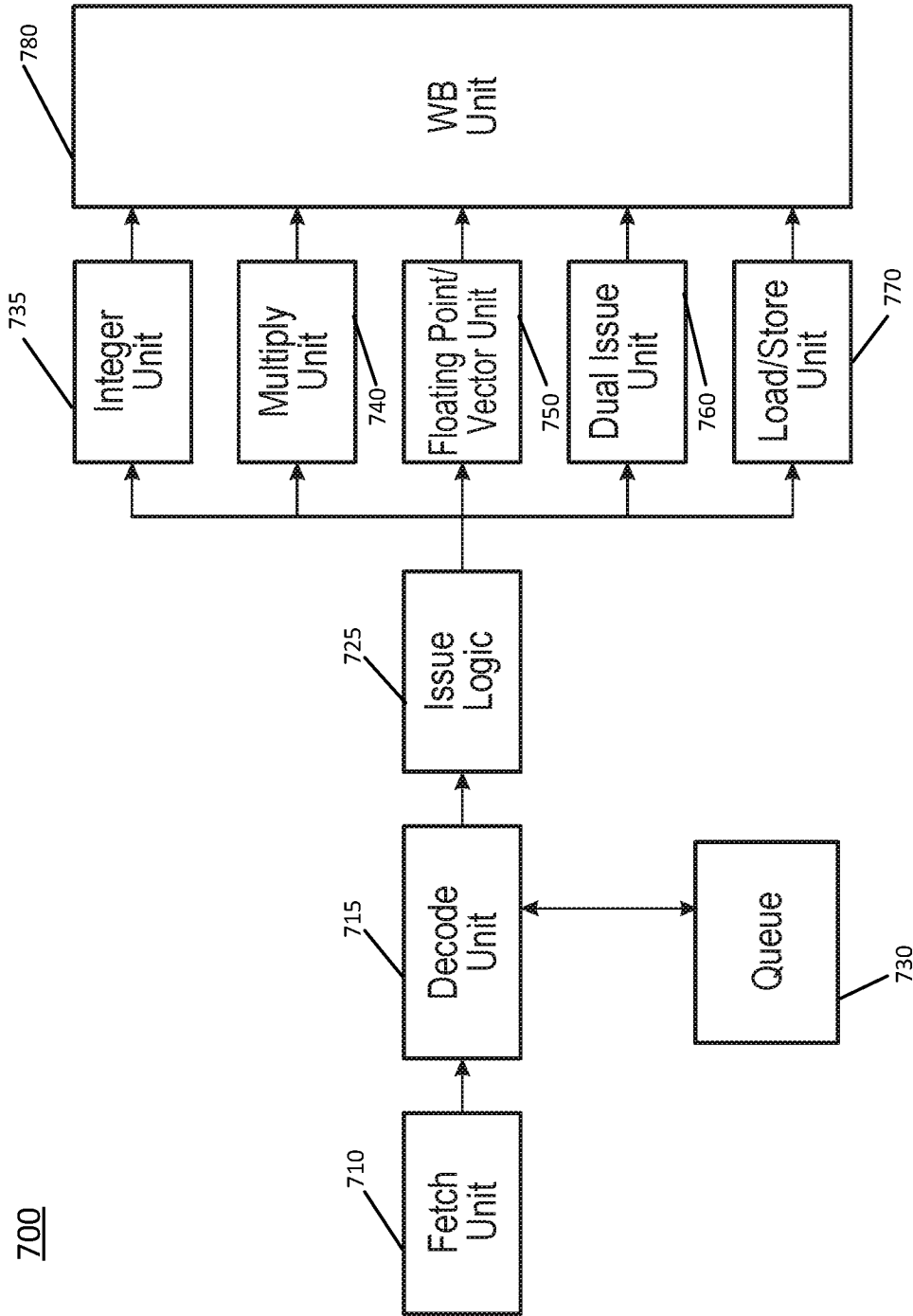


FIG. 6



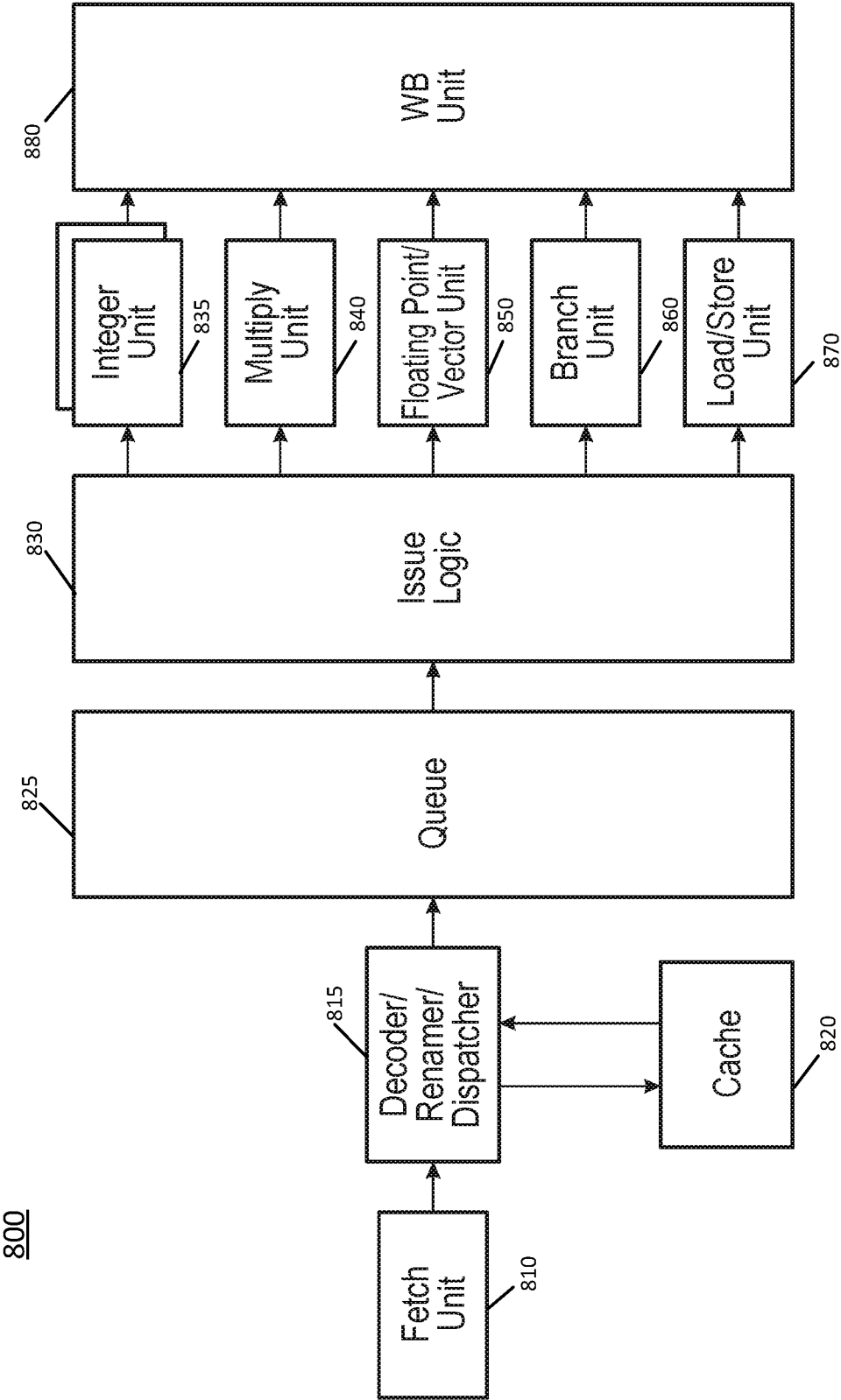


FIG. 8

900

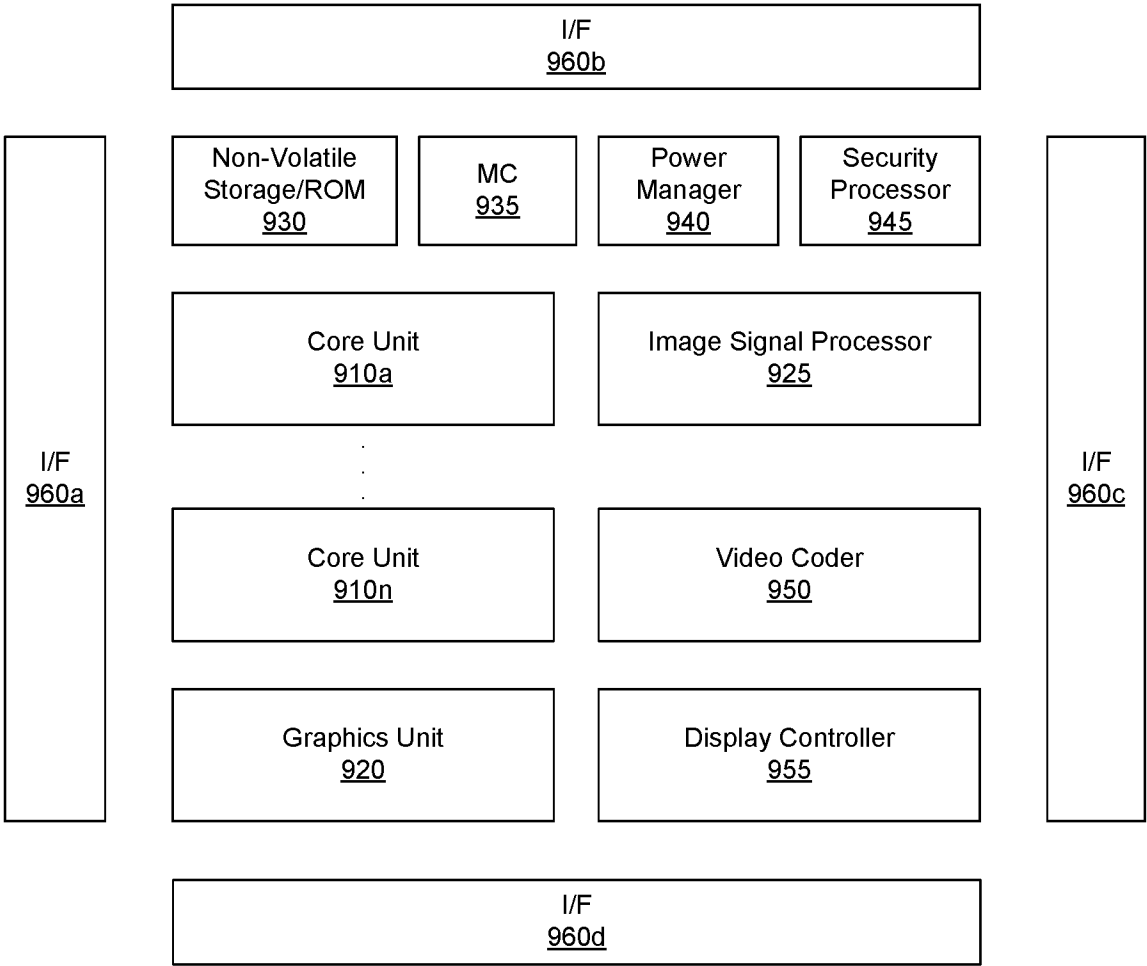


FIG. 9

1000

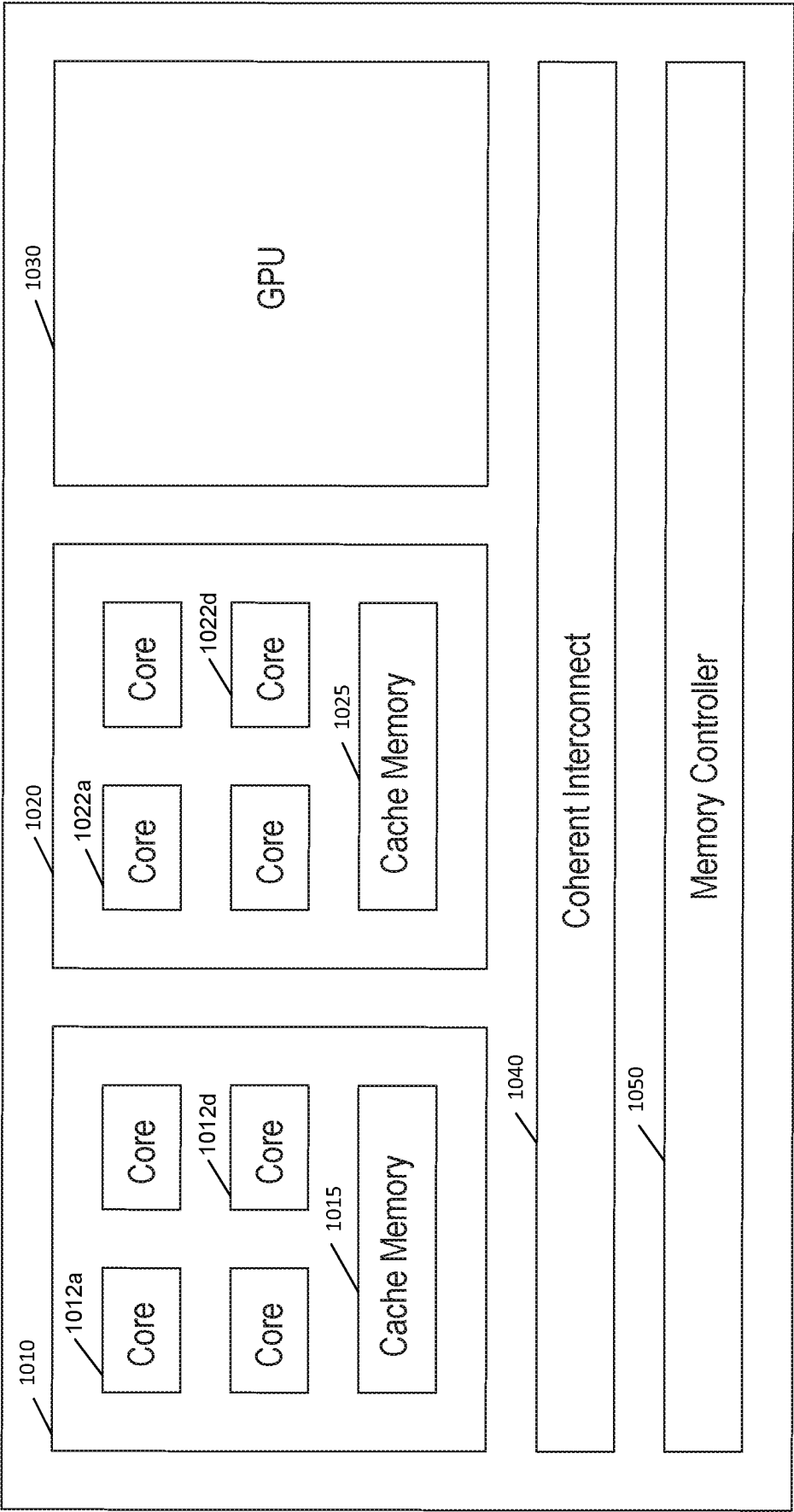


FIG. 10

11/18

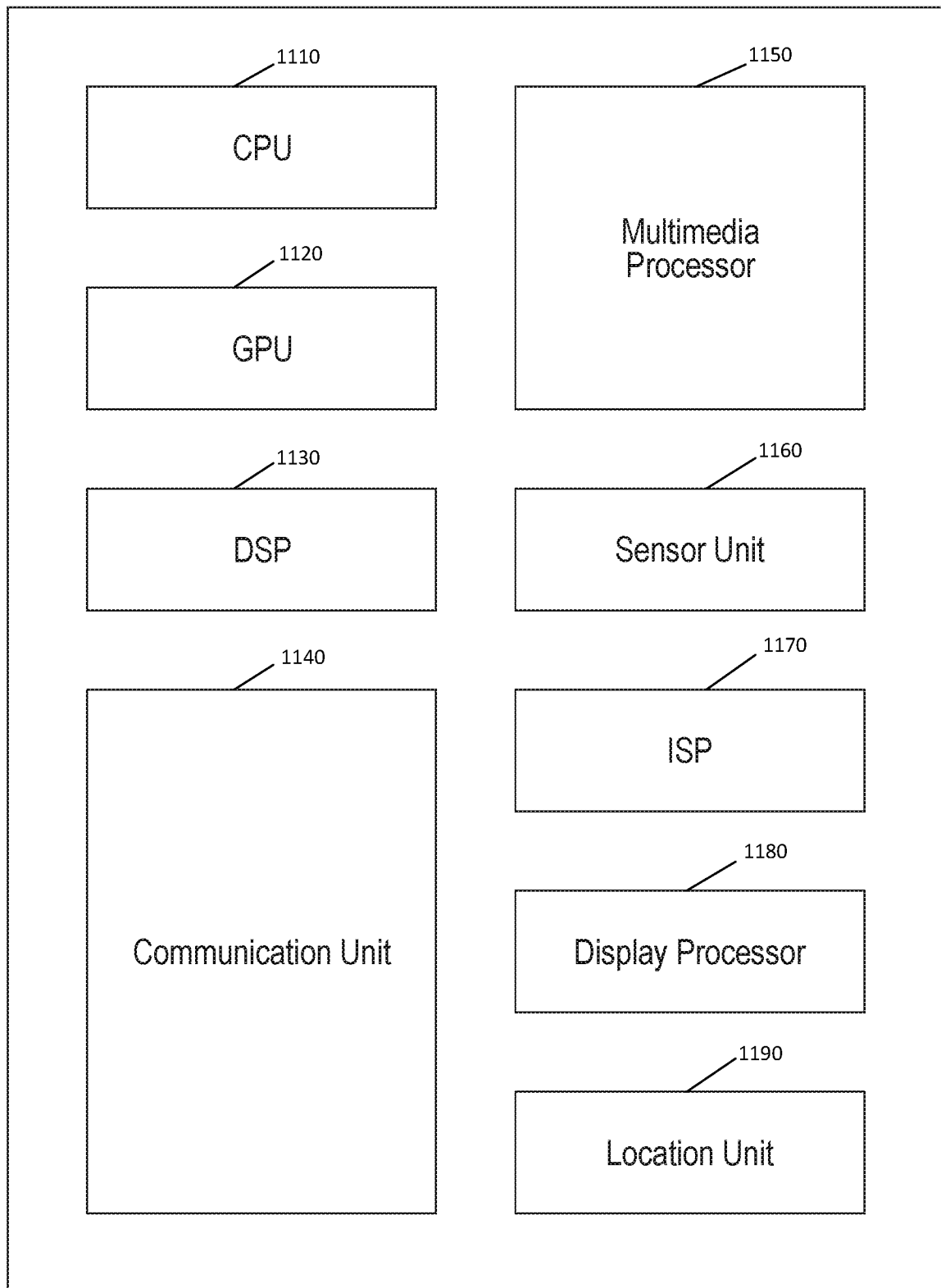
1100

FIG. 11

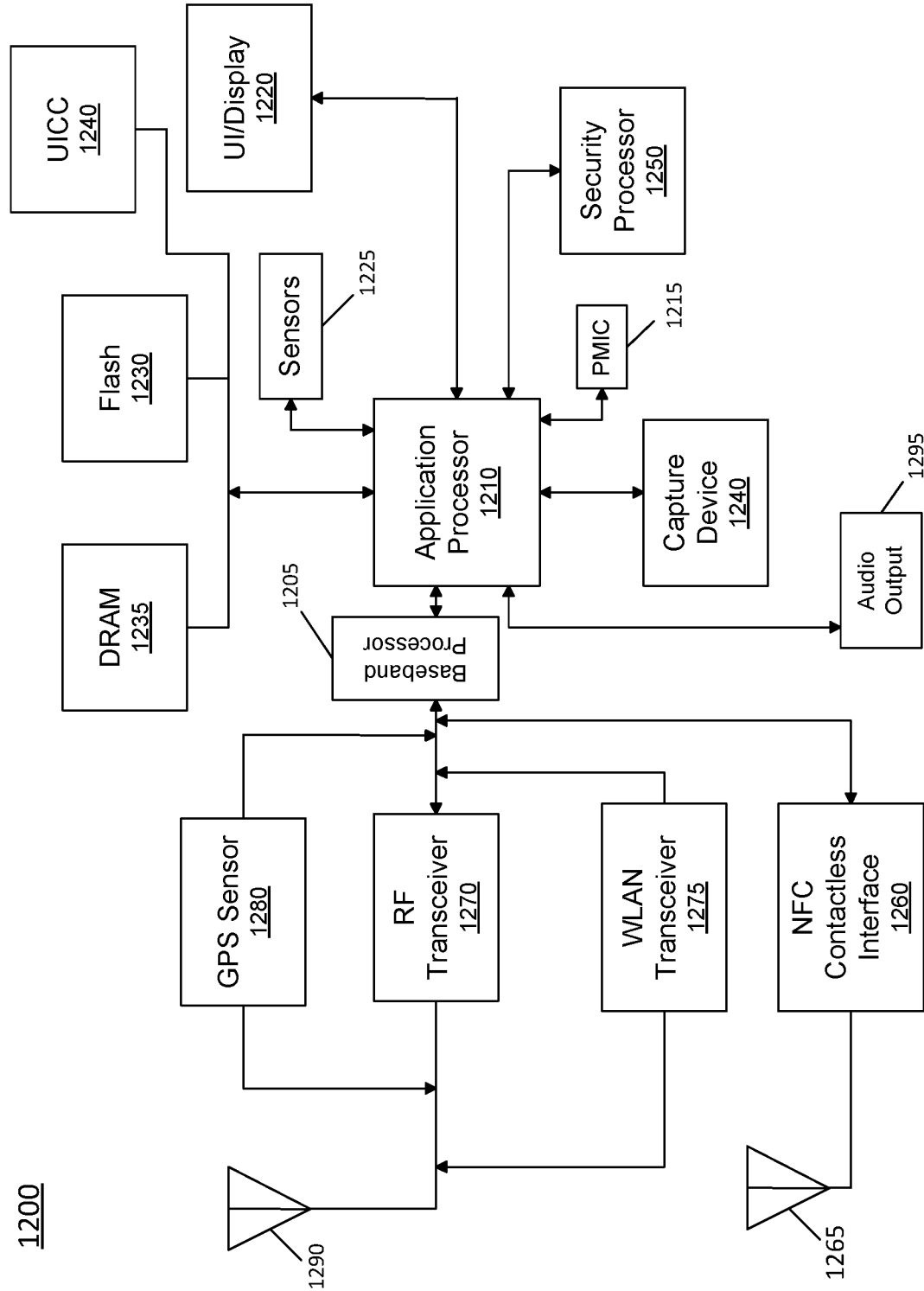


FIG. 12

1300

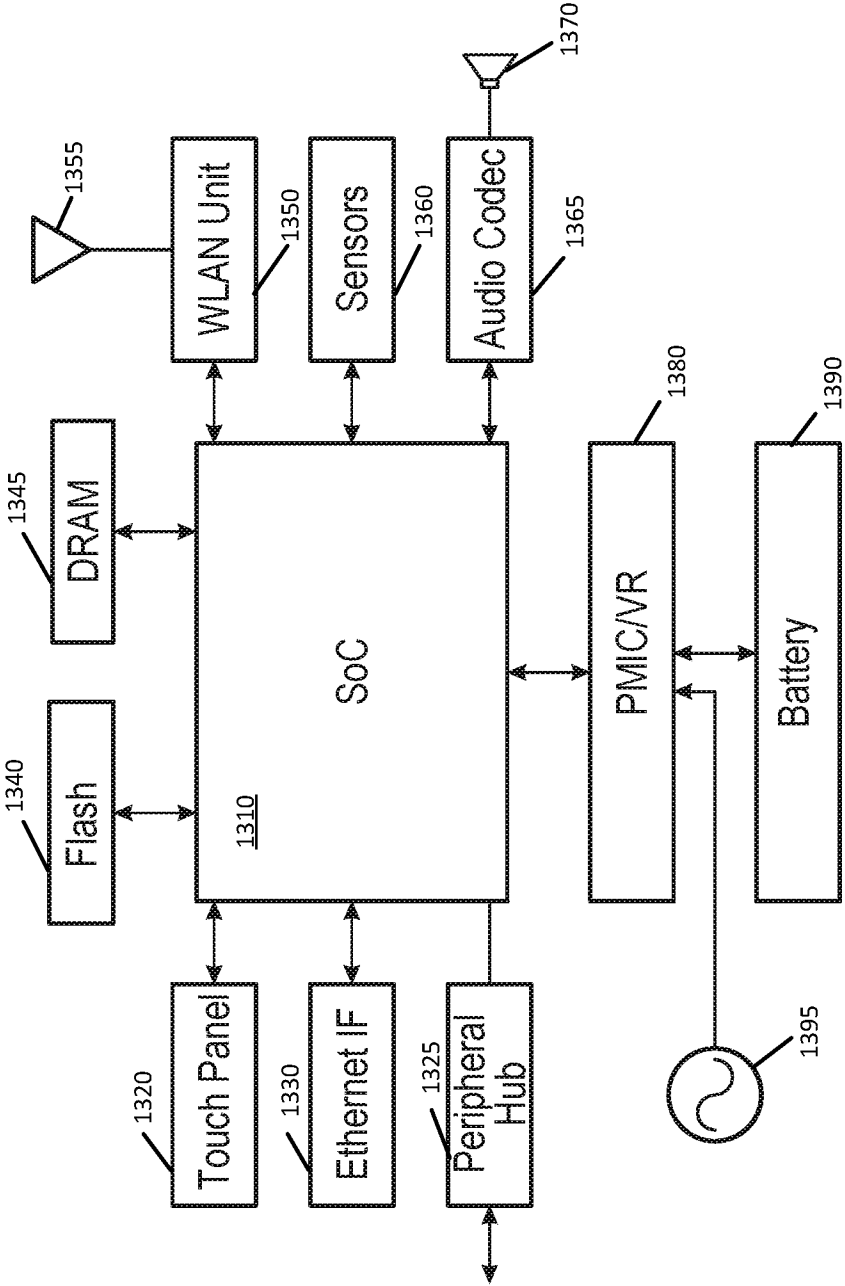


FIG. 13

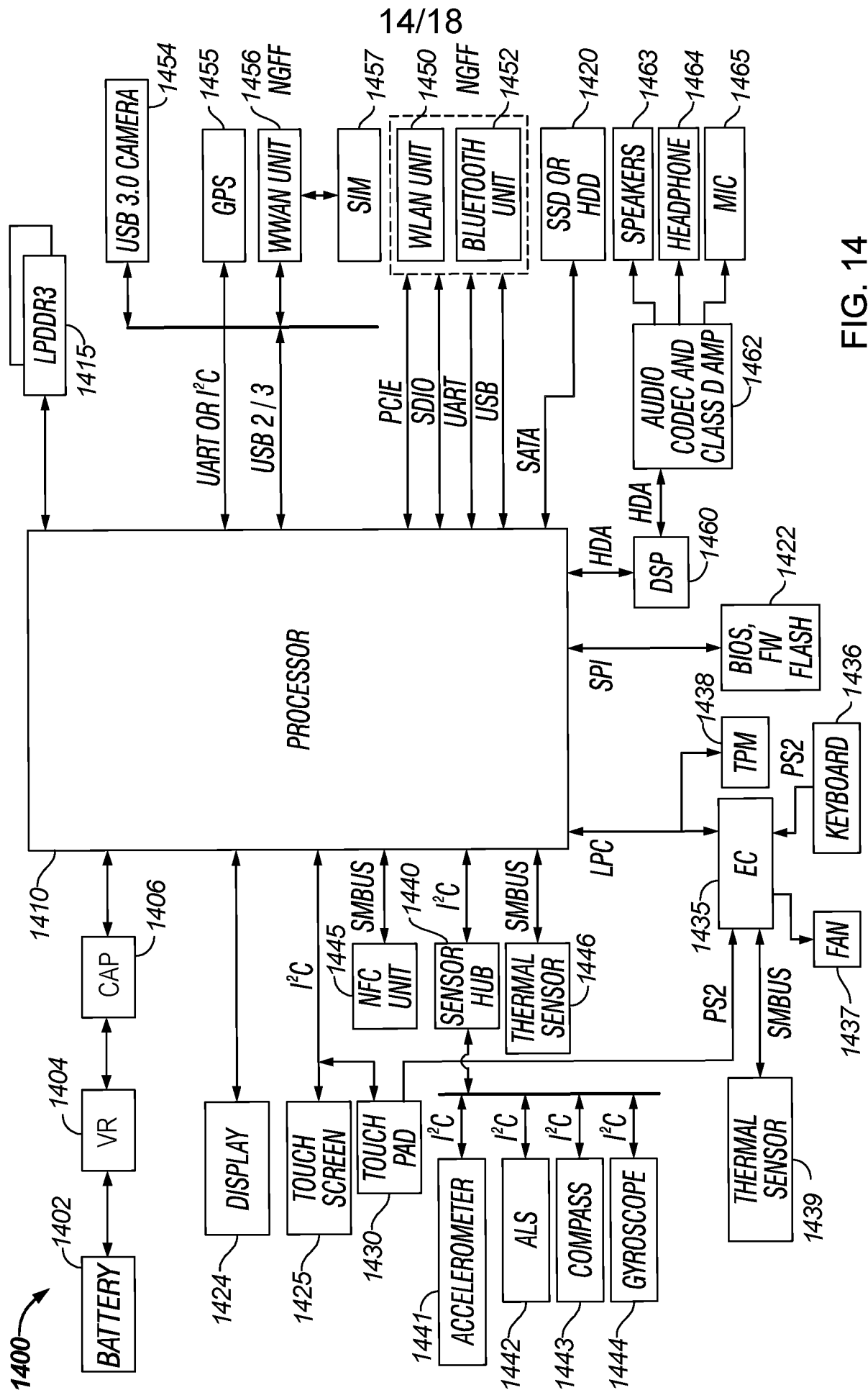


FIG. 14

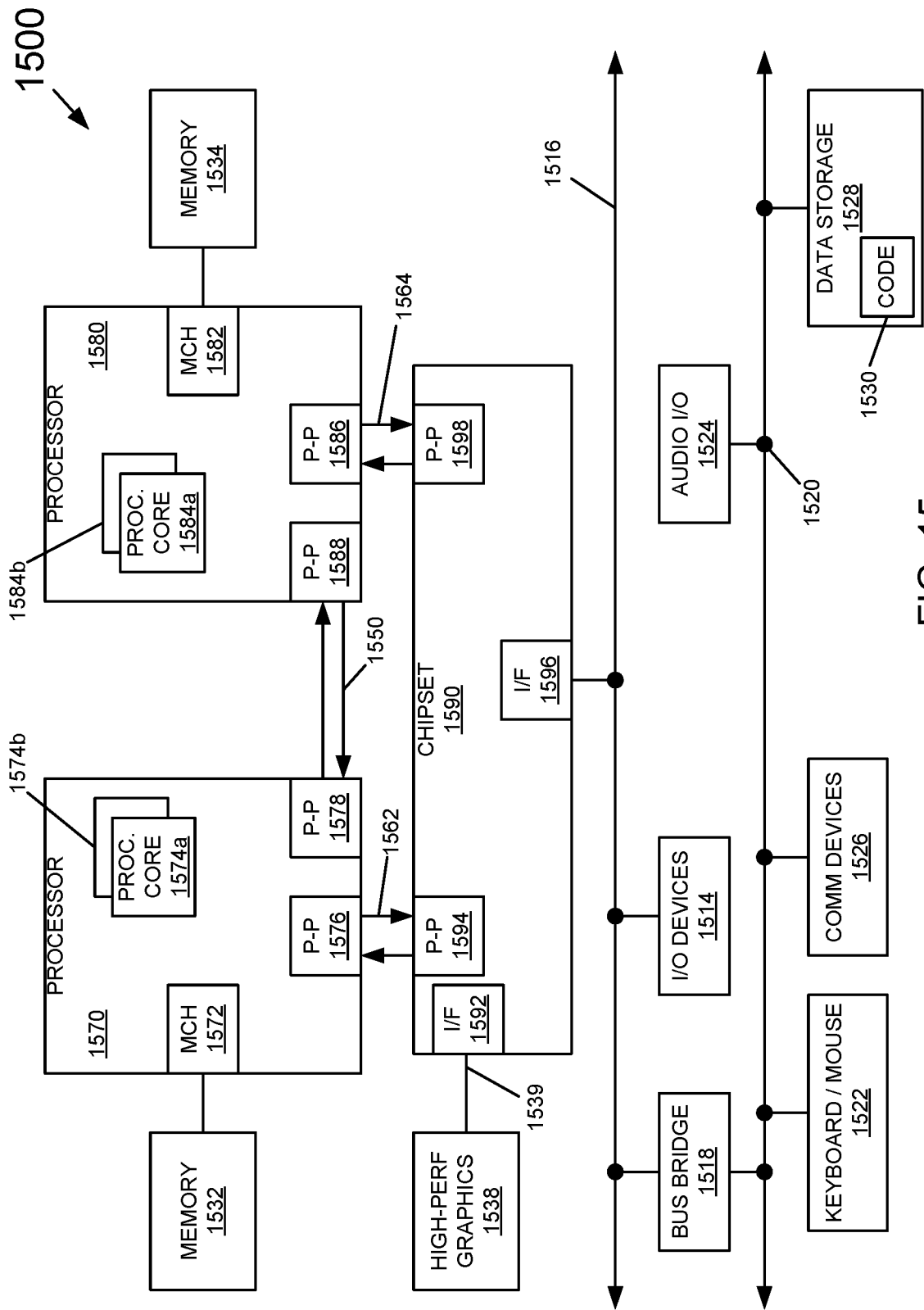


FIG. 15

1600

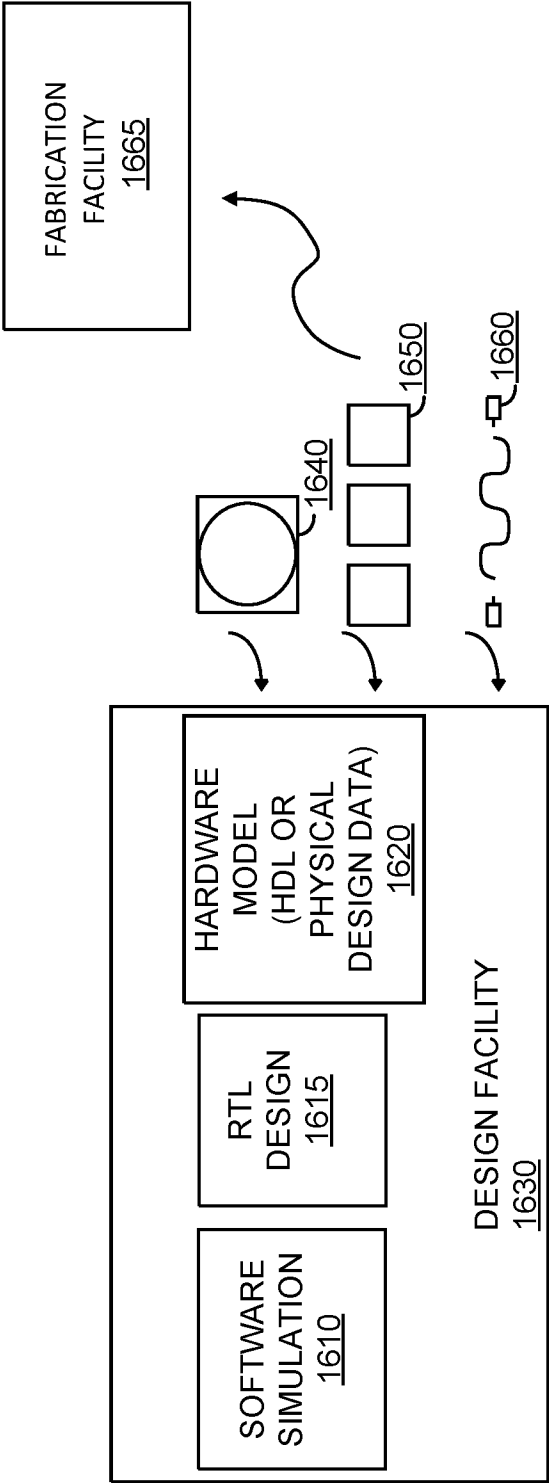


FIG. 16

17/18

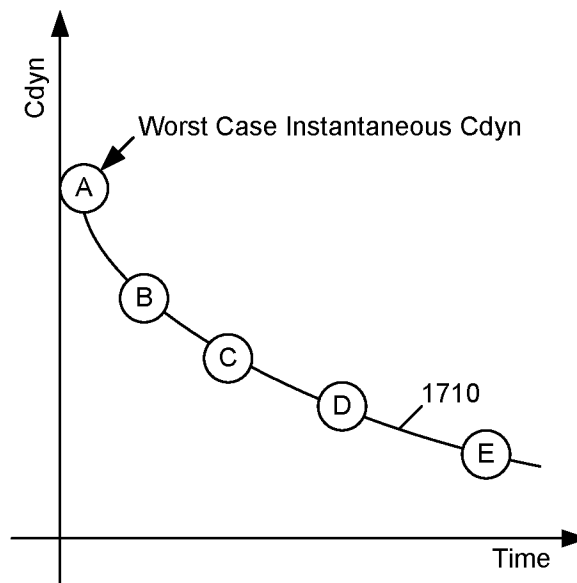
1700

FIG. 17

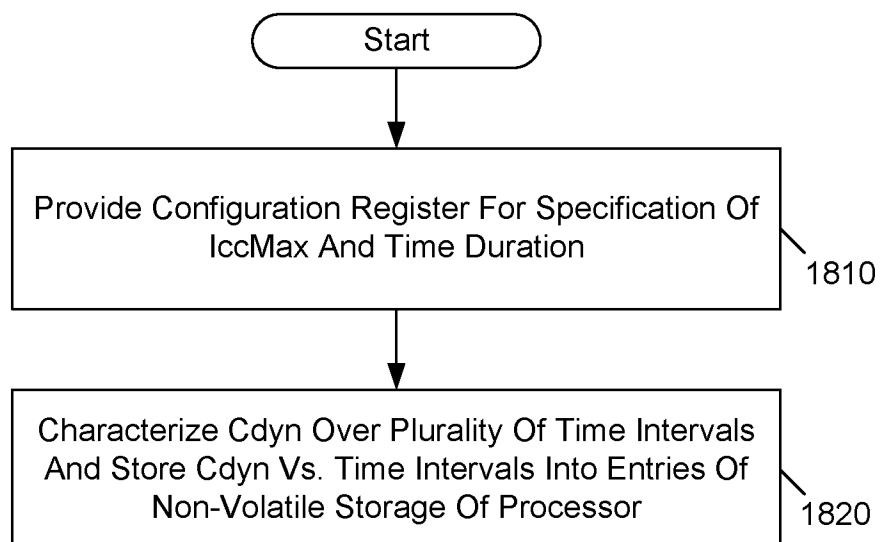
1800

FIG. 18

18/18

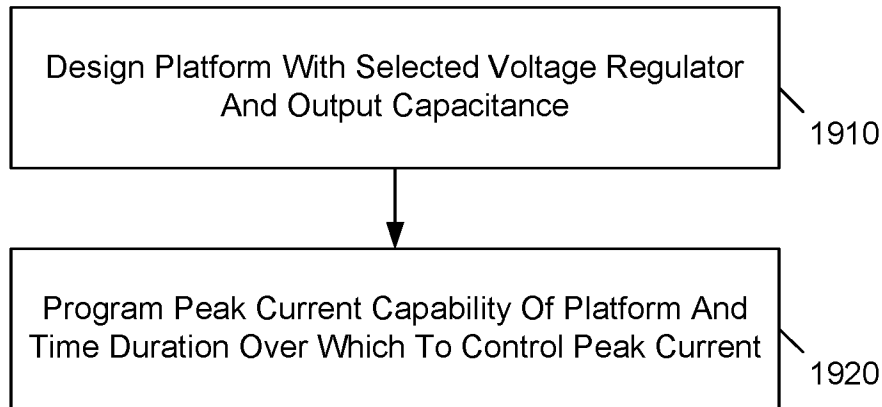
1900

FIG. 19

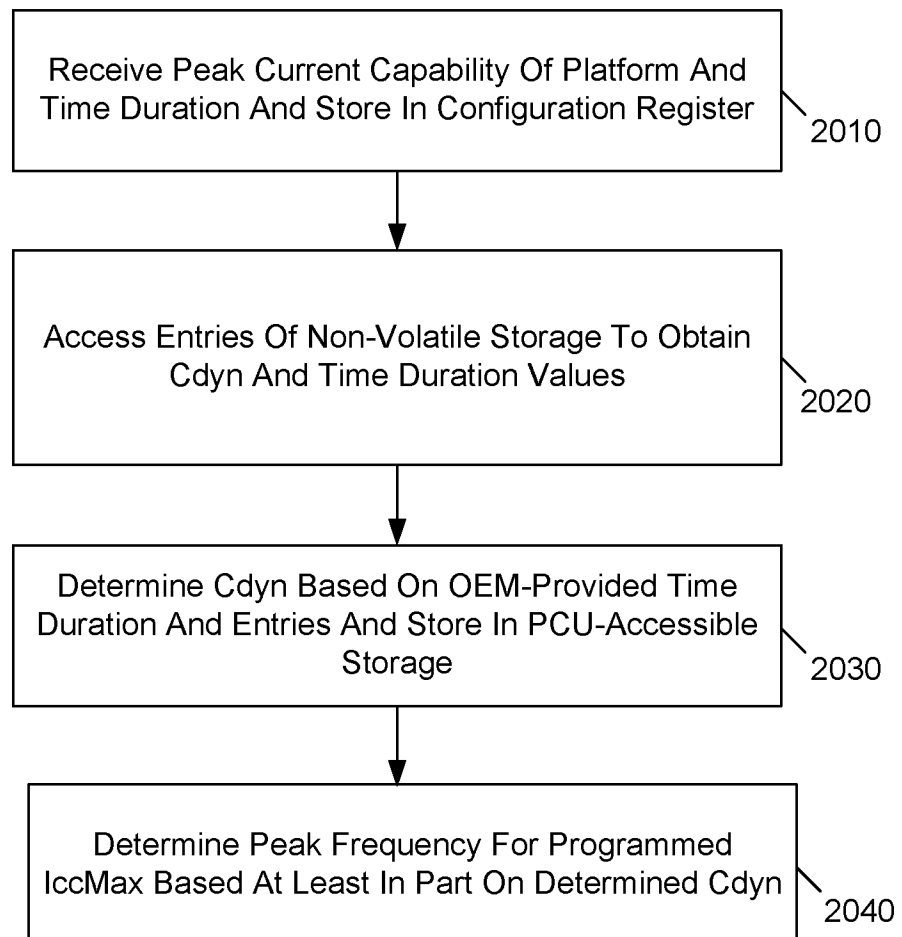
2000

FIG. 20

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2017/045252**A. CLASSIFICATION OF SUBJECT MATTER****G06F 1/32(2006.01)i, G06F 9/46(2006.01)i**

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F 1/32; G06F 1/28; G06F 17/50; G06F 1/26; G06F 1/00; G06F 9/46

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

Korean utility models and applications for utility models

Japanese utility models and applications for utility models

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

eKOMPASS(KIPO internal) & Keywords:

power consumption, processor, dynamic capacitance, time duration, current spike, voltage regulator

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2014-0359328 A1 (JAMES S. BURNS et al.) 04 December 2014 See paragraphs [0011], [0016], [0022], [0041], [0043]; claims 1, 13; and figures 3-4.	1-25
A	US 2014-0237272 A1 (ADVANCED MICRO DEVICES, INC.) 21 August 2014 See paragraph [0012]; and figure 2.	1-25
A	US 2010-0138683 A1 (EDWARD A. BURTON et al.) 03 June 2010 See paragraph [0043]; and figure 4.	1-25
A	US 2010-0011233 A1 (SAMEER HALEPETE et al.) 14 January 2010 See paragraph [0033]; and figure 2.	1-25
A	US 2006-0100840 A1 (EDWARD T. GROCHOWSKI et al.) 11 May 2006 See paragraph [0034]; and figure 3.	1-25



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

07 November 2017 (07.11.2017)

Date of mailing of the international search report

08 November 2017 (08.11.2017)

Name and mailing address of the ISA/KR

International Application Division

Korean Intellectual Property Office

189 Cheongsa-ro, Seo-gu, Daejeon, 35208, Republic of Korea

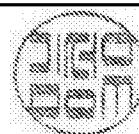


Facsimile No. +82-42-481-8578

Authorized officer

KIM, Seong Woo

Telephone No. +82-42-481-3348



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2017/045252

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2014-0359328 A1	04/12/2014	EP 2808757 A2 US 9335813 B2	03/12/2014 10/05/2016
US 2014-0237272 A1	21/08/2014	None	
US 2010-0138683 A1	03/06/2010	US 2007-0262132 A1 US 7685441 B2 US 8037326 B2	15/11/2007 23/03/2010 11/10/2011
US 2010-0011233 A1	14/01/2010	US 2002-0116650 A1 US 2013-0132749 A1 US 7100061 B2 US 7596708 B1 US 8566627 B2 US 8806247 B2 WO 01-53921 A1	22/08/2002 23/05/2013 29/08/2006 29/09/2009 22/10/2013 12/08/2014 26/07/2001
US 2006-0100840 A1	11/05/2006	AU 2002-364556 A1 AU 2002-364556 A8 CN 100489734 C CN 101539795 A CN 101539795 B CN 1739079 A DE 10297598 B4 DE 10297598 T5 GB 2400459 A GB 2400459 B GB 2412972 A JP 2006-504151 A JP 3996127 B2 KR 10-0806423 B1 KR 10-2004-0072686 A MY 136359 A US 2003-0125922 A1 US 2007-0225959 A1 US 7035785 B2 US 7236920 B2 US 7742910 B2 WO 03-058415 A2 WO 2003-058415 A3	24/07/2003 24/07/2003 20/05/2009 23/09/2009 06/02/2013 22/02/2006 10/06/2009 13/01/2005 13/10/2004 14/09/2005 12/10/2005 02/02/2006 24/10/2007 21/02/2008 18/08/2004 30/09/2008 03/07/2003 27/09/2007 25/04/2006 26/06/2007 22/06/2010 17/07/2003 30/09/2004