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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
29 November 2018 (29.11.2018)

WIPO | PCT

(10) International Publication Number
WO 2018/217899 A1

(51) International Patent Classification:
G06F 12/1009 (2016.01) *G11C 14/00* (2006.01)

(21) International Application Number:
PCT/US2018/034142

(22) International Filing Date:
23 May 2018 (23.05.2018)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

201741018056	23 May 2017 (23.05.2017)	IN
62/612,382	30 December 2017 (30.12.2017)	US
15/985,859	22 May 2018 (22.05.2018)	US

(71) Applicant: **TEXAS INSTRUMENTS INCORPORATED** [US/US]; P.O. Box 655474, Mail Station 3999, Dallas, TX 75265-5474 (US).

(71) Applicant *(for JP only)*: **TEXAS INSTRUMENTS JAPAN LIMITED** [JP/JP]; 24-1, Nishi-Shinjuku 6-chome, Shinjuku-ku, Tokyo, 160-8366 (JP).

(72) Inventor: **MANDELA, Venkateswara, Rao**; 304, Chandamama Residency, 4th Cross, Abbaiah Reddy Layout, Bangalore, Karnataka 560093 (IN).

(74) Agent: **DAVIS, Michael A.** et al.; Texas Instruments Incorporated, P.O. Box 655474, Mail Station 3999, Dallas, TX 75265-5474 (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,

(54) Title: INTEGRATED CIRCUIT CAPABLE OF SWITCHING BETWEEN NON-VOLATILE MEMORY-BASED AND VOLATILE MEMORY-BASED PROCESSOR EXECUTION

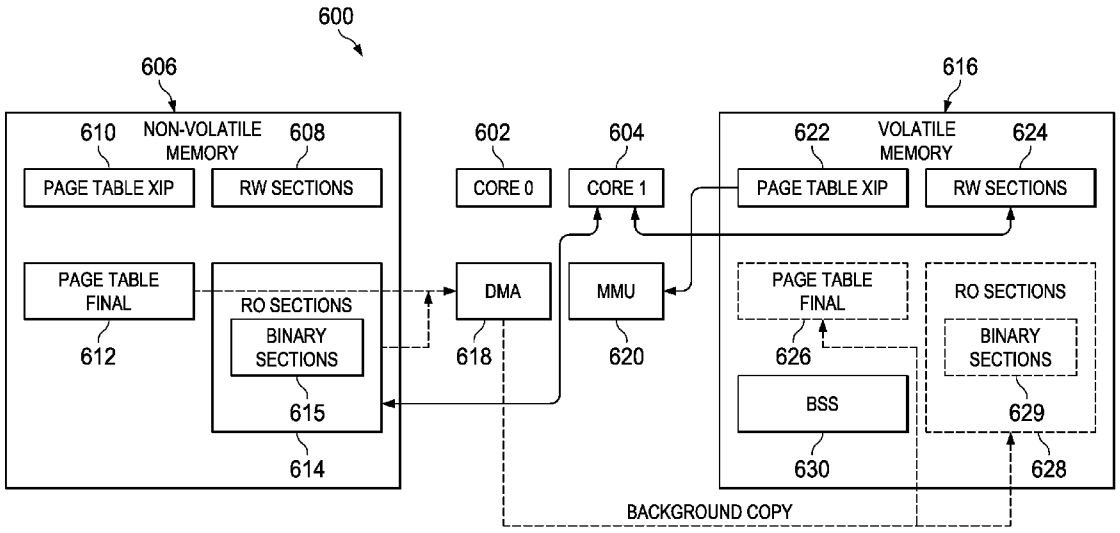


FIG. 6C

(57) Abstract: An example system (600) includes a non-volatile memory (606) including a binary section (615), a first page table (610) and a second page table (612). The system (600) also has a volatile memory (616) and a processor (604) coupled to the non-volatile memory (606) and the volatile memory (616). The processor (604) is operable to use the first page table (610) when the processor (604) is initialized. The first page table (610) includes a first pointer to the binary section (615). The processor (604) is operable to cause copying of the binary section (615) of the non-volatile memory (606) to the volatile memory (616) to create a copied binary section (629) in the volatile memory (616). Also, the processor (604) is operable to use the second page table (612) when the copying is complete. The second page table (612) includes a second pointer to the copied binary section (629).

WO 2018/217899 A1

[Continued on next page]

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 applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

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

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

INTEGRATED CIRCUIT CAPABLE OF SWITCHING BETWEEN NON-VOLATILE MEMORY-BASED AND VOLATILE MEMORY-BASED PROCESSOR EXECUTION

[0001] This relates generally to executing code in computing systems, and more particularly to execute-in-place (XIP) execution in a computing system.

BACKGROUND

[0002] In certain networks, devices connected to the network need a certain amount of time to go from a powered-off state to a communicative state. For example, vehicles often employ controller area networks (CANs). When the vehicle is off, almost all devices connected to the CAN do not have power. When the vehicle starts, these devices receive power. Vehicle manufacturers often specify that devices connected to the CAN must be able to answer a message on the CAN within a specified start-up period, which creates challenges for device designers.

SUMMARY

[0003] An example system includes a non-volatile memory including a binary section, a first page table and a second page table. The system also includes a volatile memory and a processor coupled to the non-volatile memory and the volatile memory. The processor is operable to use the first page table when the processor is initialized. The first page table includes a first pointer to the binary section. The processor is operable to cause copying of the binary section of the non-volatile memory to the volatile memory to create a copied binary section in the volatile memory. Also, the processor is operable to use the second page table when the copying is complete. The second page table includes a second pointer to the copied binary section.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] FIG. 1 is a timing diagram.

[0005] FIG. 2 is a block diagram of an example processing circuit.

[0006] FIG. 3 is a block diagram of an example processing circuit.

[0007] FIG. 4 is a block diagram of an example processing circuit.

[0008] FIGS. 5A, 5B and 5C (collectively "FIG. 5") are block diagrams illustrating an example operation of an example processor circuit.

[0009] FIGS. 6A-6D (collectively "FIG. 6") are block diagrams of an example processor circuit.

[0010] FIG. 7 is a graphical representation of the operation of the example of FIG. 6.

[0011] FIG. 8 is a flow diagram for an example method.

DETAILED DESCRIPTION OF EXAMPLE EMBODIMENTS

[0012] Corresponding numerals and symbols in the drawings generally refer to corresponding parts unless otherwise indicated. The drawings are not necessarily drawn to scale.

[0013] The term "coupled" may include connections made with intervening elements, and additional elements and various connections may exist between any elements that are "coupled."

[0014] Compiled applications (binaries) for devices on controller area networks (CANs), and the boot ROM, may be stored in non-volatile memory, for example a read-only memory (ROM), such as an electrically erasable programmable read-only memory (EEPROM), because volatile memory loses its data when it loses power. Within a 50 ms start-up period, power up and read out from the ROM usually requires about 30 ms. This leaves 20 ms for the processor to begin executing to be ready to receive messages on the CAN. The processor can run the program in two ways: directly from the ROM (non-volatile memory) or from random access memory (RAM) (volatile memory) with the program read from the ROM to the RAM. Running the program directly from the ROM may be referred to herein as execute-in-place (XIP), which allows for quick start up. However, reading from the ROM is much slower than reading from RAM. Therefore, the operational speed of XIP is slow.

[0015] Operation from RAM is much faster than operating from ROM. However, operating from RAM requires copying the binary from ROM to RAM. The copying process may take much longer than the 20 ms available start up time. Therefore, this option may not be feasible with tighter time constraints. One way to address this problem is to use a separate processor dedicated to communication with the CAN (e.g., a separate communication processor). However, using a separate processor dedicated to communication may not be cost effective in many situations.

[0016] FIG. 1 is a timing diagram 100. Period 102 shows an example 50 ms requirement for response to messages from a CAN. Period 104 shows the time required for start of the power management integrated circuit (PMIC), execution of the boot ROM and the initial reading of the boot loader. In high noise environments, such as in vehicles, a separate PMIC isolates

processing devices from noisy power sources. Therefore, processing cannot begin until 30 ms after power-up. For example, no processing occurs during period 106 and a device on the CAN must be ready to receive and respond to messages on the CAN at the end of period 108 or during the 20 ms after the end of period 106.

[0017] FIG. 2 is a block diagram of an example processing circuit 200. Processor 202 is coupled to non-volatile memory 204 and volatile memory 208. Examples of non-volatile memory 204 are read-only memory (ROM), erasable programmable read-only memory (EPROM) and flash memory. Non-volatile memory 204 stores binary section 206. Binary section 206 includes instructions for the operation of processor 202. Because non-volatile memory 204 is non-volatile, non-volatile memory 204 maintains binary section 206 even when power is off. Volatile memory 208 loses data stored in volatile memory 208 when the power is off. In this example, volatile memory 208 is double data rate (DDR) random access memory (RAM), and thus processor 202 can read from and write to volatile memory 208 very quickly.

[0018] FIG. 3 is a block diagram of an example processing circuit 300. Processor 302 is coupled to non-volatile memory 304 and volatile memory 308. Non-volatile memory 304 stores binary section 306. Binary section 306 includes instructions for the operation of processor 302. In this example, processor 302 reads the instructions directly from the binary section 306. This allows processor 302 to quickly begin operation and meet the timing requirements illustrated in FIG. 1. However, because the access time for non-volatile memory 304 is slow, the ongoing operation of processor 302 has low performance.

[0019] FIG. 4 is a block diagram of an example processing circuit 400. Processor 402 is coupled to non-volatile memory 404 and volatile memory 408. Non-volatile memory 404 stores binary section 406. Binary section 406 includes instructions for the operation of processor 402. In this example, rather than reading binary section 406 directly from non-volatile memory 404, processor 402 first makes a copy of binary section 406 in volatile memory 408. FIG. 4 shows this as copied binary section 410. Processor 402 then reads the instructions from the copied binary section 410 after the copy is complete. However, the time necessary to copy binary section 406 to volatile memory 408 is too long to enable processor 402 to meet the timing requirements of FIG. 1.

[0020] FIGS. 5A, 5B and 5C (collectively "FIG. 5") are block diagrams illustrating an example operation of processor circuit 500. In FIG. 5A, processor 502 is coupled to non-volatile

memory 504 and volatile memory 508. Non-volatile memory 504 stores binary section 506. Binary section 506 includes instructions for the operation of processor 502. In this example, processor 502 reads the instructions from the binary section 506. This allows processor 502 to quickly begin operation and meet the timing requirements illustrated in FIG. 1. As illustrated in FIG. 5B, while processor 502 is executing using binary section 506, processor 502 makes a copy of binary section 506 in volatile memory 508. FIG. 5B shows this as copied binary section 510. FIG. 5C shows that, when copied binary section 510 is complete (e.g., when the copying is complete), processor 502 executes copied binary section 510 rather than binary section 506. Therefore, processor circuit 500 meets the timing constraints of FIG. 1 while providing the improved performance of executing copied binary section 510 from volatile memory 508. FIGS. 6A-6D hereinbelow show further details of the handover of execution from binary section 506 to copied binary section 510.

[0021] FIGS. 6A-6D (collectively "FIG. 6") are block diagrams illustrating an example processor circuit 600. Processor circuit 600 includes two processing cores 602 and 604 in this example. Core 602, in an example, is an A15 core according to a design by ARM, Ltd. Core 604, in an example, is an M4 core that is also according to a design by ARM, Ltd. In this example, core 602 is the more powerful A15 core for fast start-up, while the M4 is more cost, space and power efficient for normal operation. In other examples, processor circuit 600 includes additional processing cores. In this example, core 602 is a managing core that begins processing after a reset to processor circuit 600 due to power loss, application failure, etc. Core 604 handles communication, such as with a CAN, among other functions. Core 602 and core 604 couple to memory management unit (MMU) 620 and direct memory access controller (DMA) 618. MMU 620 manages access to all memory. DMA 618 handles memory transfers without processor or MMU control, other than instructions to initialize the transfer. DMA 618 and MMU 620 couple to non-volatile memory 606 and volatile memory 616. FIG. 6A includes lines showing the coupling between core 602, core 604, DMA 618, MMU 620, non-volatile memory 606 and volatile memory 616 by, in this example, a system of buses. FIGS. 6B-6D omit these coupling lines for clarity.

[0022] Examples of non-volatile memory 606 are electrically erasable programmable read-only memory (EEPROM), flash memory (e.g., non-volatile read/write memory) or other types of read-only memory. Non-volatile memory 606 includes read-only (RO) sections 614.

For example, RO sections 614 include a communications application in compiled, binary format in a binary section 615. RO sections 614 also include constants and other invariable information necessary to execute the program. Non-volatile memory 606 also includes read/write (RW) sections 608. RW sections 608 include formatting and other structural elements for (non-constant) variables of the application. Non-volatile memory 606 also includes page table XIP 610 and page table final 612 (also referred to as XIP page table and final page table, respectively). The page tables are tables of pointers used by MMU 620 to point to the appropriate memory element at the appropriate time, as further explained hereinbelow. Briefly, core 604 addresses the binaries stored in binary section 615 using virtual addresses. MMU 620 uses page table XIP 610 to convert the virtual addresses to physical addresses in non-volatile memory 606. After RO sections 614 is copied to volatile memory 616, MMU switches to using page table final 612 (as copied to volatile memory 616, as further explained hereinbelow). Then MMU 620 uses page table final 612 to convert the virtual addresses to physical addresses in volatile memory 616.

[0023] FIG. 6B is a block diagram showing the initial steps of preparing the memory for execution of the binary stored in RO sections 614 by core 604. Core 602 begins initial execution on reset when processor circuit 600 initializes, such as due to a loss of power. Core 602 causes DMA 618 to copy RW sections 608 and page table XIP 610 to volatile memory 616 as copied RW sections 624 and copied page table XIP 622, respectively. These memory elements may be small and copy quickly (for example, in microseconds). Core 602 then triggers core 604 to begin processing. These memory elements allow core 604 to begin executing the binary stored in RO sections 614. Specifically, MMU 620 uses copied page table XIP 622 to point to memory locations to allow core 604 to begin execution in execute-in-place (XIP) mode. For example, for read-only sections, page table XIP 622 points to RO sections 614 in non-volatile memory 606. For read/write sections, page table XIP 622 points to RW sections 624 in volatile memory 616. Core 604 will also cause MMU 620 to create basic support section (BSS) 630 (FIG. 6C). BSS 630 provides basic variables for operation of core 604 apart from the specifically defined variables in RW section 624.

[0024] FIG. 6C is a block diagram showing the initial execution of the binary stored in RO sections 614 by core 604. As shown in FIG. 6C, MMU 620 uses copied page table XIP 622 to point core 604 to RO sections 614 for read-only data and to RW sections 624 for read/write data.

Also, core 602 instructs DMA 618 to initiate a copy of page table final 612 and RO sections 614 to volatile memory 616 as copied page table final 626 and copied RO sections 628 (including a copied binary section 629), respectively. In an alternate example, using instructions stored in core 604 or instructions read from RO section 614, core 604 instructs DMA 618 to initiate a copy of page table final 612 and RO sections 614 to volatile memory 616 as copied page table final 626 and copied RO sections 628 (including a copied binary section 629), respectively. DMA 618 creates these copies independently of core 604 and avoids using the resources of core 604 to make the copies. The compilation process for the compiled application stored in RO section 614 specifically defines the variables in RW sections 624. Copied page table final 626 and copied RO sections 628 have dashed outlines in FIG. 6C to indicate that copies of copied page table final 626 and copied RO sections 628 are not yet complete.

[0025] FIG. 6D is a block diagram showing that core 604 operates using volatile memory 616 after the copies of copied page table final 626 and copied RO sections 628 are complete. For example, when DMA 618 completes copying copied page table final 626 and copied RO sections 628, DMA 618 issues an interrupt to core 602. In this example, an interrupt is any indication from the DMA 618 that copying is complete without regard to any specific format. In response to the interrupt, core 602 instructs MMU 620 to use page table final 626. Copied page table final 626 points to copied RW sections 624 and copied RO sections 628 to allow execution of the binary by core 604. Because copied RO sections 628 are read-only, the transfer from RO sections 614 to copied RO sections 628 is seamless. After the transition to copied page table final 626, core 604 executes at a much faster rate made possible by using volatile memory 616, which is much faster than non-volatile memory 606. For example, flash memory (a type of non-volatile memory) has typical access times on the order of 1500 ns while synchronous double-data-rate (DDR) RAM (a type of volatile memory) has typical access times on the order of 75 ns.

[0026] In an example, processor circuit 600 is an integrated circuit as a system-on-chip (SoC). In other examples, some or all components of processor circuit may be separate components and, in some examples, are packages in hybrid modules with the other components of processor circuit 600, and in other examples, are in separate packages. Also, although FIG. 6 shows two cores, in some examples, the processor circuit can include only one core or more than two cores. Also, other types of processing modules may be used, such as microprocessors.

[0027] FIG. 7 is a graphical representation of the operation of the example of FIG. 6. The top portion of chart 700 represents XIP execution 702. When XIP execution 702 begins, start background copy 706 begins. When the background copy is complete, DMA 618 (FIG. 6) sends the copy complete notification 708. Then the intelligent and seamless switch over 710 changes switch 712 from XIP execution 702 to RAM execution 704. Switch 710 is a representation of the function where MMU 620 (FIG. 6) switches from using page table XIP 622 (FIG. 6) to final table final 626 (FIG. 6) in response copy completion notification from DMA 618 (FIG. 6), and thus core 604 operates using the RO sections 628 (FIG. 6).

[0028] FIG. 8 is a flow diagram of an example method 800. Method 800 is an example method of operating the example of FIG. 6. In this example, core 0 602 (FIG. 6) performs steps on the left of the dashed line, and core 1 604 (FIG. 6) performs steps on the right of the dashed line. Step 802 starts core 0 602 (FIG. 6). Step 804 sets up the XIP interface by copying the page table XIP 610 (FIG. 6) from non-volatile memory 606 to volatile memory 616 (FIG. 6). Step 806 sets up the MMU 620 (FIG. 6) to use copied page table XIP 622 (FIG. 6). Step 808 copies the RW sections 608 (FIG. 6) from non-volatile memory 606 to volatile memory 616 (FIG. 6). Step 810 takes core 1 604 (FIG. 6) out of reset mode. Step 812 copies the RO sections 614 (FIG. 6) in response to instructions from core 602 (FIG. 6) from non-volatile memory 606 (FIG. 6) to volatile memory 616 (FIG. 6). Step 814 copies the non-XIP page tables using DMA 618 (FIG. 6), such as page table final 612 from non-volatile memory 606 to volatile memory 616 (FIG. 6). Step 816 switches the page table used by MMU 620 (FIG 6.) to begin using copied page table final 626 (FIG. 6). In parallel to steps 810, 812, 814, and 816, step 818 begins execution of the binary by core 1 604 (FIG. 6). Step 820 operates core 1 604 (FIG. 6) in XIP manner using copied page table XIP 622 (FIG. 6). Step 822 is in response to step 816 in that when MMU 620 switches from copied page table XIP 622 to copied page table final 626 (FIG. 6), core 1 604 will run in non-XIP manner.

[0029] Modifications are possible in the described examples, and other examples are possible, within the scope of the claims.

CLAIMS

What is claimed is:

1. A system comprising:

a non-volatile memory including a binary section, a first page table and a second page table;

a volatile memory; and

a processor coupled to the non-volatile memory and the volatile memory, the processor operable to use the first page table when the processor is initialized, the first page table including a first pointer to the binary section, the processor operable to cause copying of the binary section of the non-volatile memory to the volatile memory while the processor is using the first page table to create a copied binary section in the volatile memory, and the processor operable to use the second page table when the copying is complete, the second page table including a second pointer to the copied binary section.

2. The system of claim 1 wherein the binary section is in a read-only section.

3. The system of claim 1 wherein the non-volatile memory is flash memory.

4. The system of claim 1 wherein the processor has at least one core.

5. The system of claim 1 wherein the first page table used by the processor is a copy of the first page table in the volatile memory.

6. The system of claim 1 further including a direct memory access controller wherein the processor is operable to instruct the direct memory access controller to copy the binary section from the non-volatile memory to the volatile memory and provide an indication to the processor when copying of the binary section from the non-volatile memory to the volatile memory is complete.

7. The system of claim 1 wherein the binary section includes a communications application.

8. An integrated circuit comprising:

a memory management unit;

a direct memory access controller to access a volatile memory, a non-volatile memory including a binary section, a first page table and a second page table, and the memory management unit; and

a processor to access the non-volatile memory, the volatile memory, the memory management unit and the direct memory access controller, the processor operable to cause the

memory management unit to use the first page table when the processor is initialized, the first page table including a first pointer to the binary section, the processor operable to cause the direct memory access controller to copy the binary section of the non-volatile memory to the volatile memory to generate a copied binary section in the volatile memory, and the processor operable to cause the memory management unit to use the second page table when the copying is complete, the second page table including a second pointer to the copied binary section.

9. The integrated circuit of claim 8 wherein the binary section is in a read-only section.

10. The integrated circuit of claim 8 wherein the processor has at least one core.

11. The integrated circuit of claim 8 wherein the first page table used by the processor is a copy of the first page table in the volatile memory.

12. The integrated circuit of claim 8 wherein the binary section includes a communications application.

13. A method comprising:

providing a non-volatile memory having a binary section, a first page table having a first pointer to the binary section, and a second page table;

copying the first page table to a volatile memory;

executing with a processor an application in the binary section using the first page table copied to the volatile memory;

copying the binary section and the second page table to the volatile memory while the processor executes the application using the copy of the first page table copied to the volatile memory; and

when the copying of the binary section and the second page table to the volatile memory is complete, continuing execution of the application by the processor using the second page table, the second page table having a second pointer to the copy of the binary section in the volatile memory.

14. The method of claim 13 wherein the copying of the binary section and the second page table to the volatile memory is performed by a direct memory access controller.

15. The method of claim 13 wherein: the processor accesses the non-volatile memory and the volatile memory using a memory management unit; and the memory management unit accesses the first page table and, after the copying of the binary section and the second page table

to the volatile memory is complete, accesses the copy of the second page table in the volatile memory.

16. The method of claim 13 wherein the copying of the binary section and the second page table to the volatile memory is performed by a direct memory access controller.

17. The method of claim 13 wherein the non-volatile memory is a flash memory.

18. The method of claim 13 wherein the volatile memory is a RAM.

19. The method of claim 13 wherein the binary section is part of a read-only section.

20. The method of claim 13 wherein the application is a communications application.

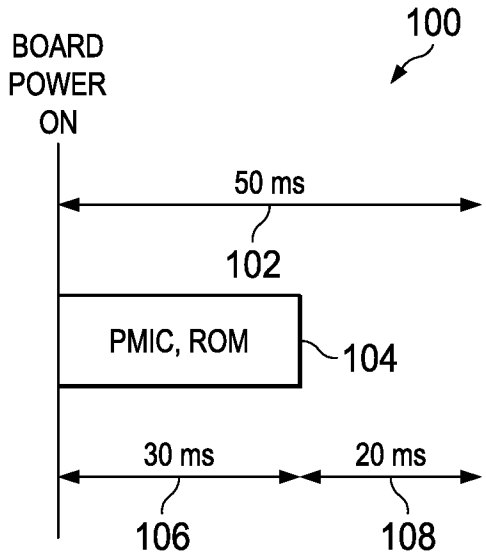


FIG. 1

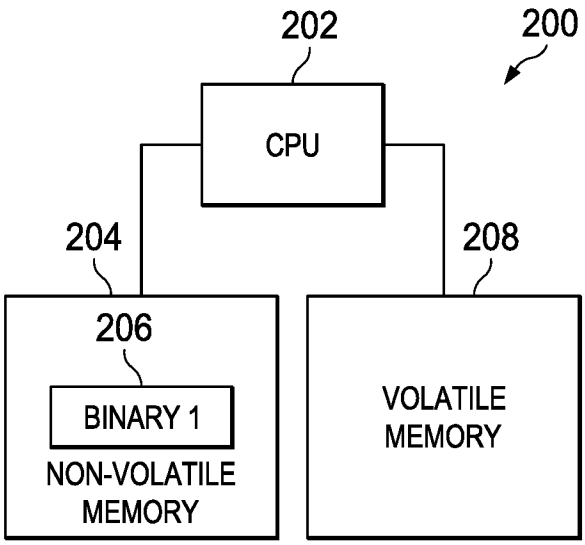


FIG. 2

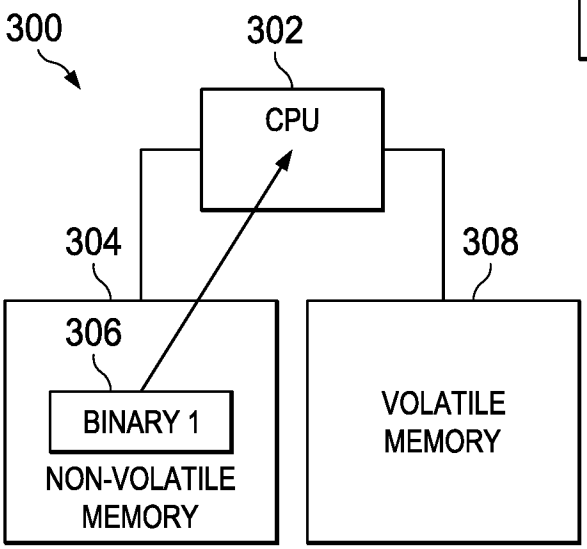


FIG. 3

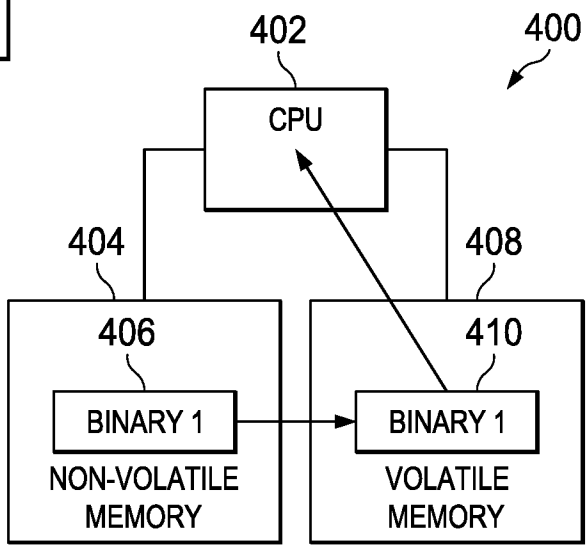


FIG. 4

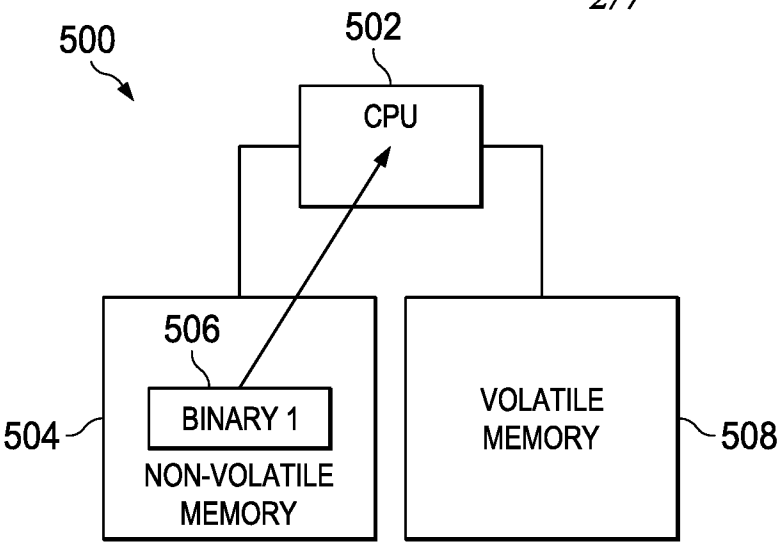


FIG. 5A

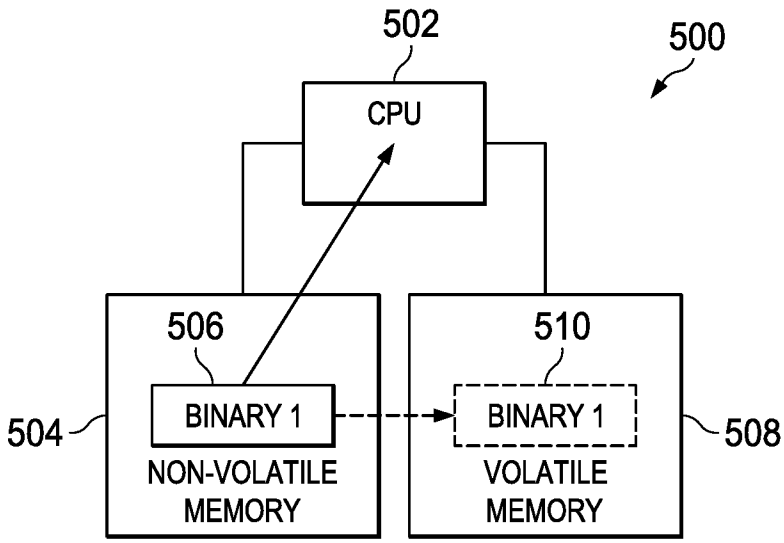


FIG. 5B

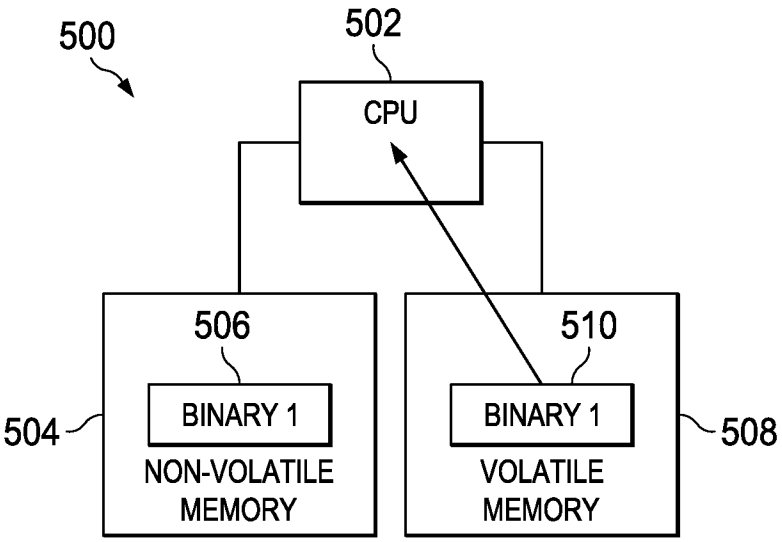


FIG. 5C

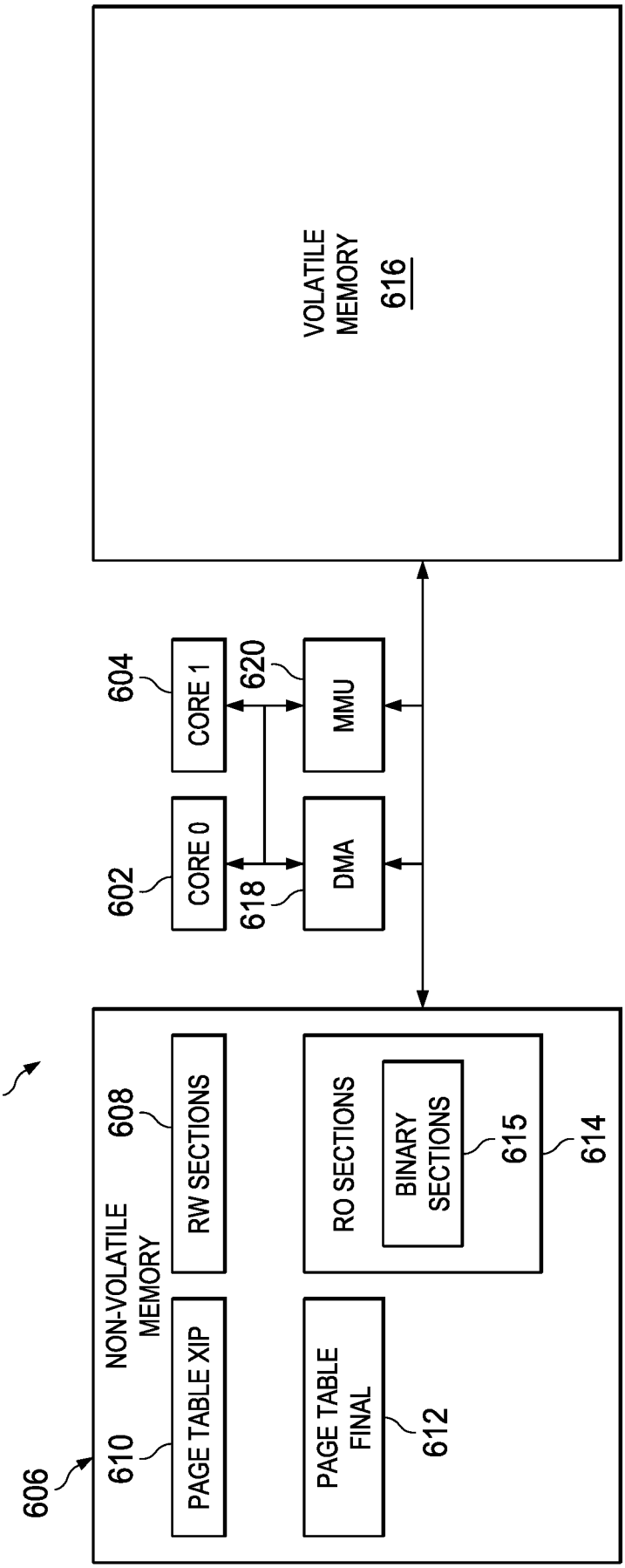

VOLATILE MEMORY
616
604
CORE 1
620
MMU
602
CORE 0
618
DMA
600
606
610
PAGE TABLE XIP
NON-VOLATILE MEMORY
608
RW SECTIONS
PAGE TABLE FINAL
612
RO SECTIONS
BINARY SECTIONS
615
614

FIG. 6A

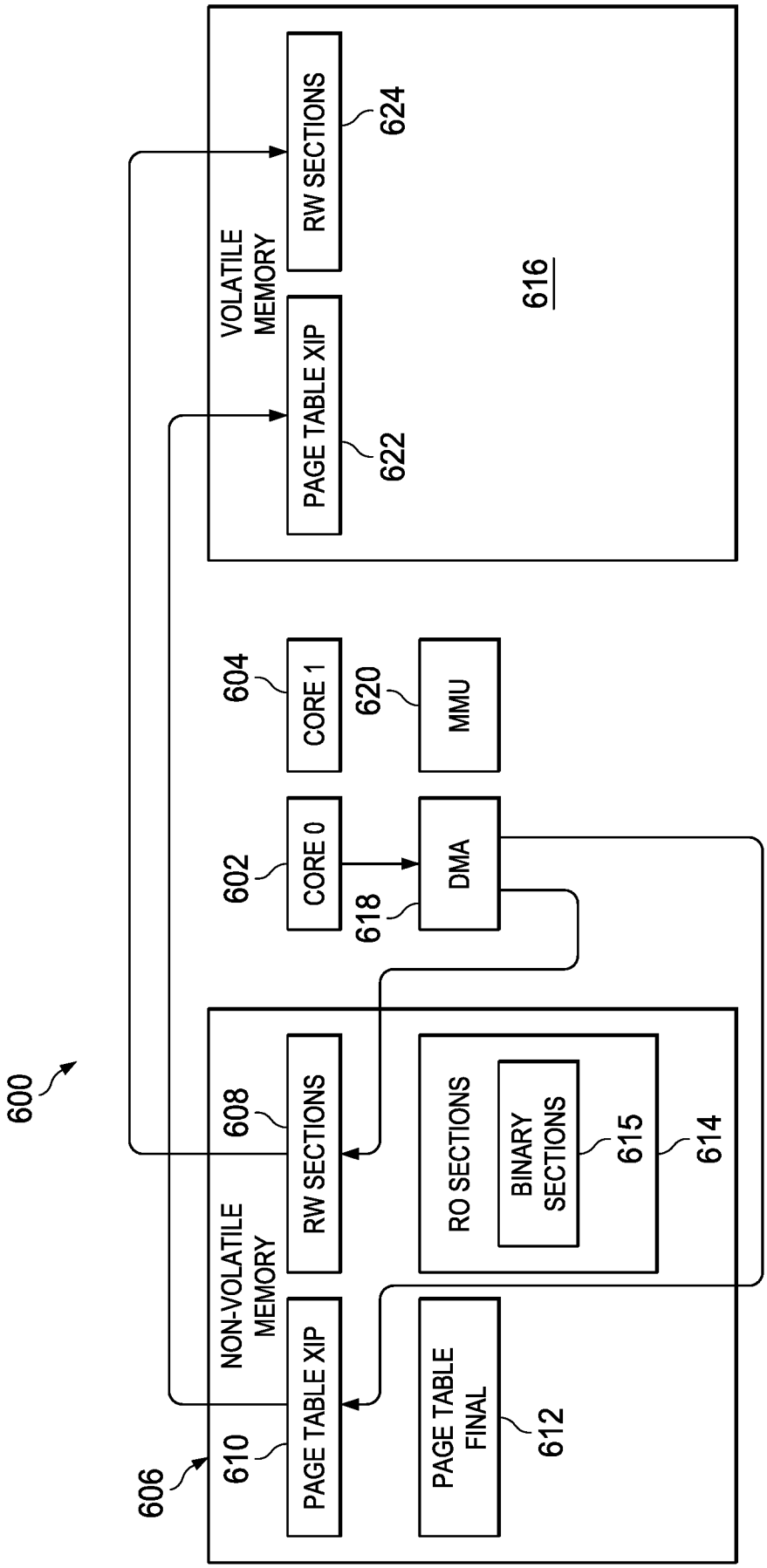

600
606
610
PAGE TABLE XIP
NON-VOLATILE MEMORY
608
RW SECTIONS
PAGE TABLE FINAL
612
RO SECTIONS
BINARY SECTIONS
615
614
602
CORE 0
618
DMA
604
CORE 1
620
MMU
VOLATILE MEMORY
PAGE TABLE XIP
622
RW SECTIONS
624
616

FIG. 6B

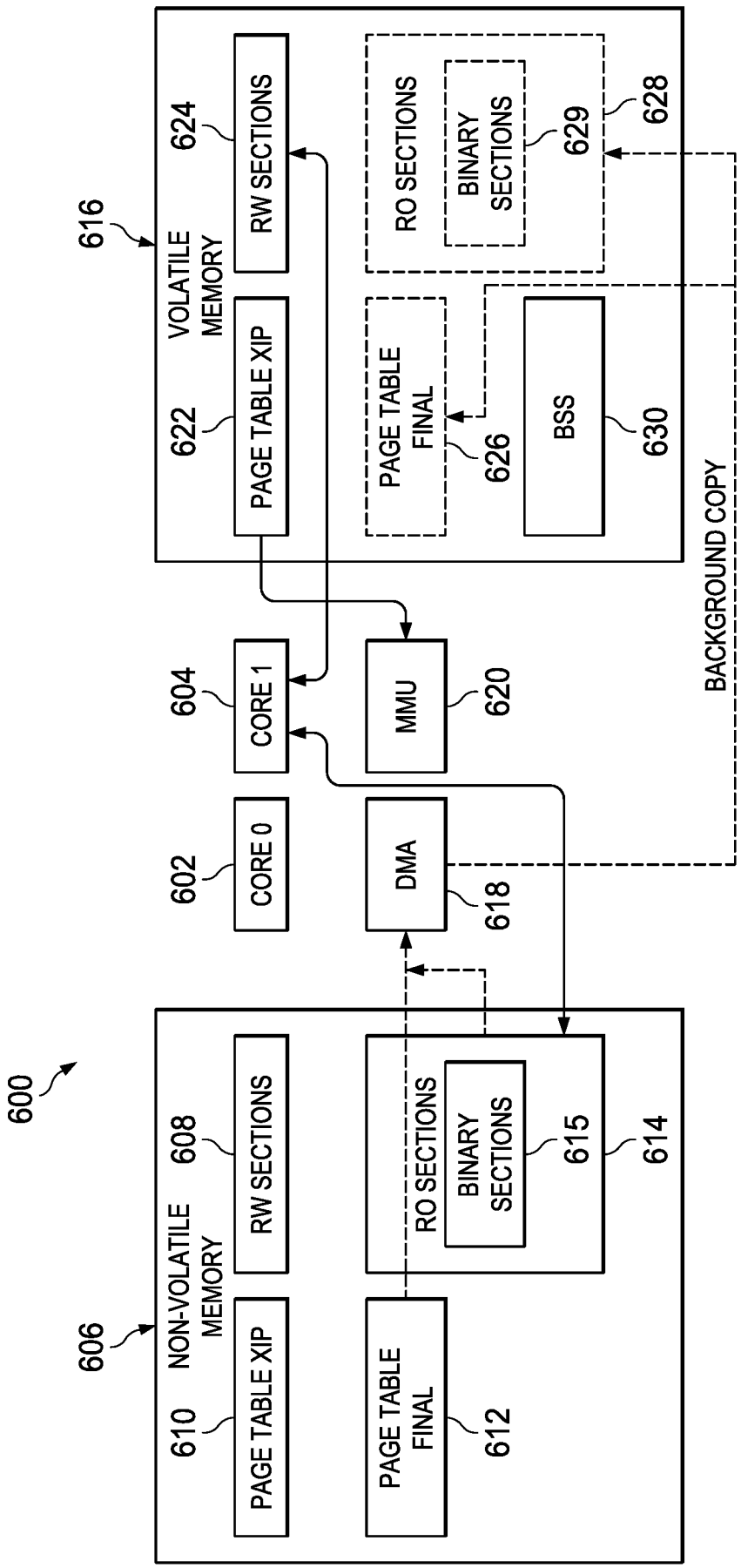

600
606
NON-VOLATILE MEMORY
610
PAGE TABLE XIP
608
RW SECTIONS
612
PAGE TABLE FINAL
614
RO SECTIONS
615
BINARY SECTIONS
602
CORE 0
604
CORE 1
618
DMA
620
MMU
616
VOLATILE MEMORY
622
PAGE TABLE XIP
624
RW SECTIONS
626
PAGE TABLE FINAL
628
RO SECTIONS
629
BINARY SECTIONS
630
BSS
BACKGROUND COPY

FIG. 6C

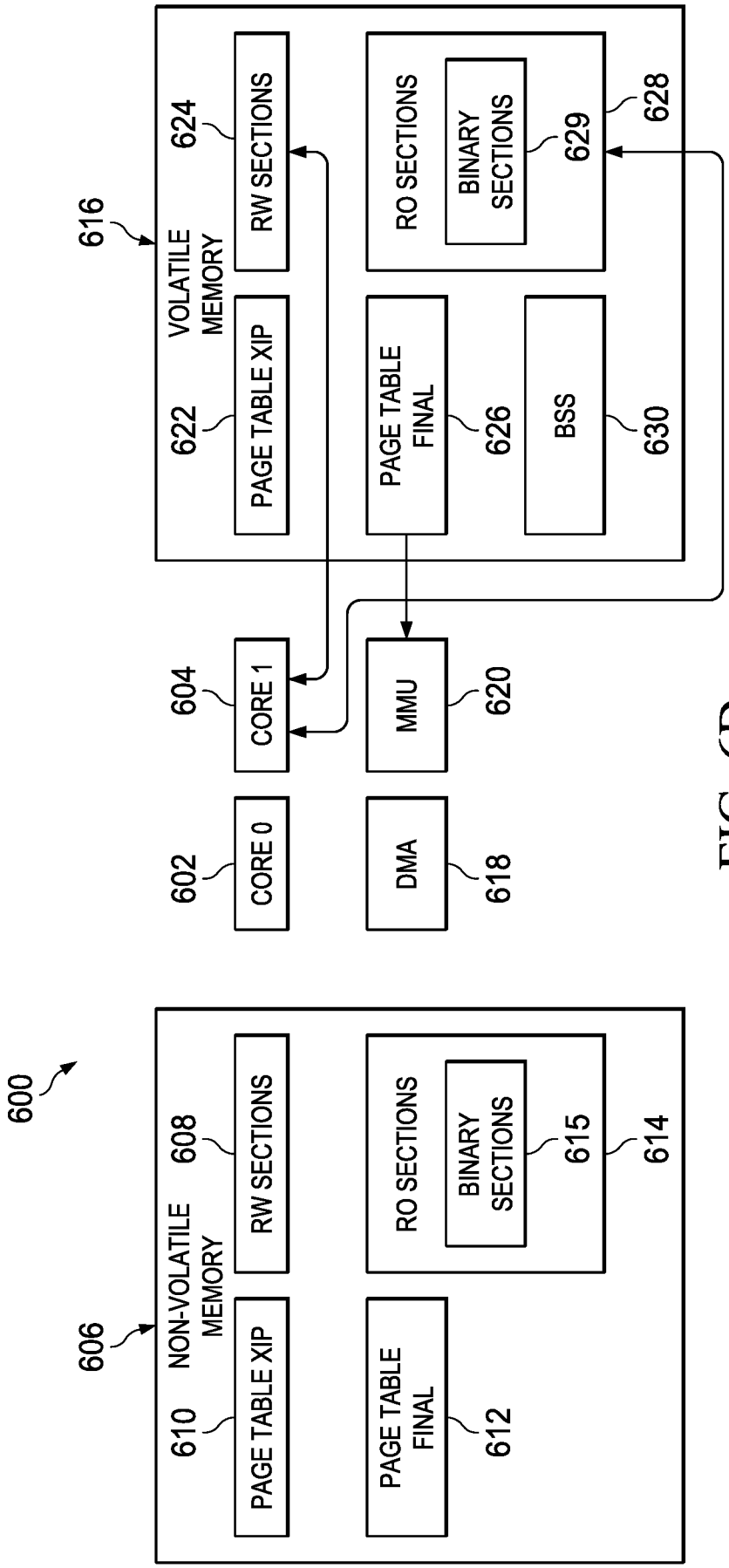

600
606
NON-VOLATILE MEMORY
608
RW SECTIONS
610
PAGE TABLE XIP
612
PAGE TABLE FINAL
614
RO SECTIONS
615
BINARY SECTIONS
602
CORE 0
604
CORE 1
618
DMA
620
MMU
616
VOLATILE MEMORY
622
PAGE TABLE XIP
624
RW SECTIONS
626
PAGE TABLE FINAL
628
RO SECTIONS
629
BINARY SECTIONS
630
BSS

FIG. 6D

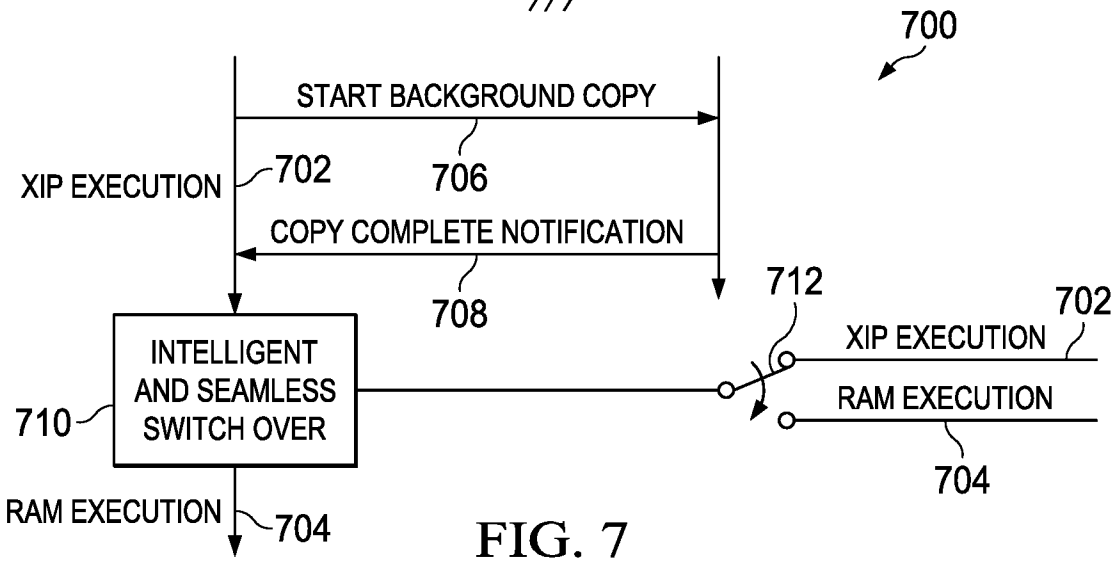

700
START BACKGROUND COPY
706
XIP EXECUTION
702
COPY COMPLETE NOTIFICATION
708
712
702
XIP EXECUTION
INTELLIGENT AND SEAMLESS SWITCH OVER
710
RAM EXECUTION
704
RAM EXECUTION
704

FIG. 7

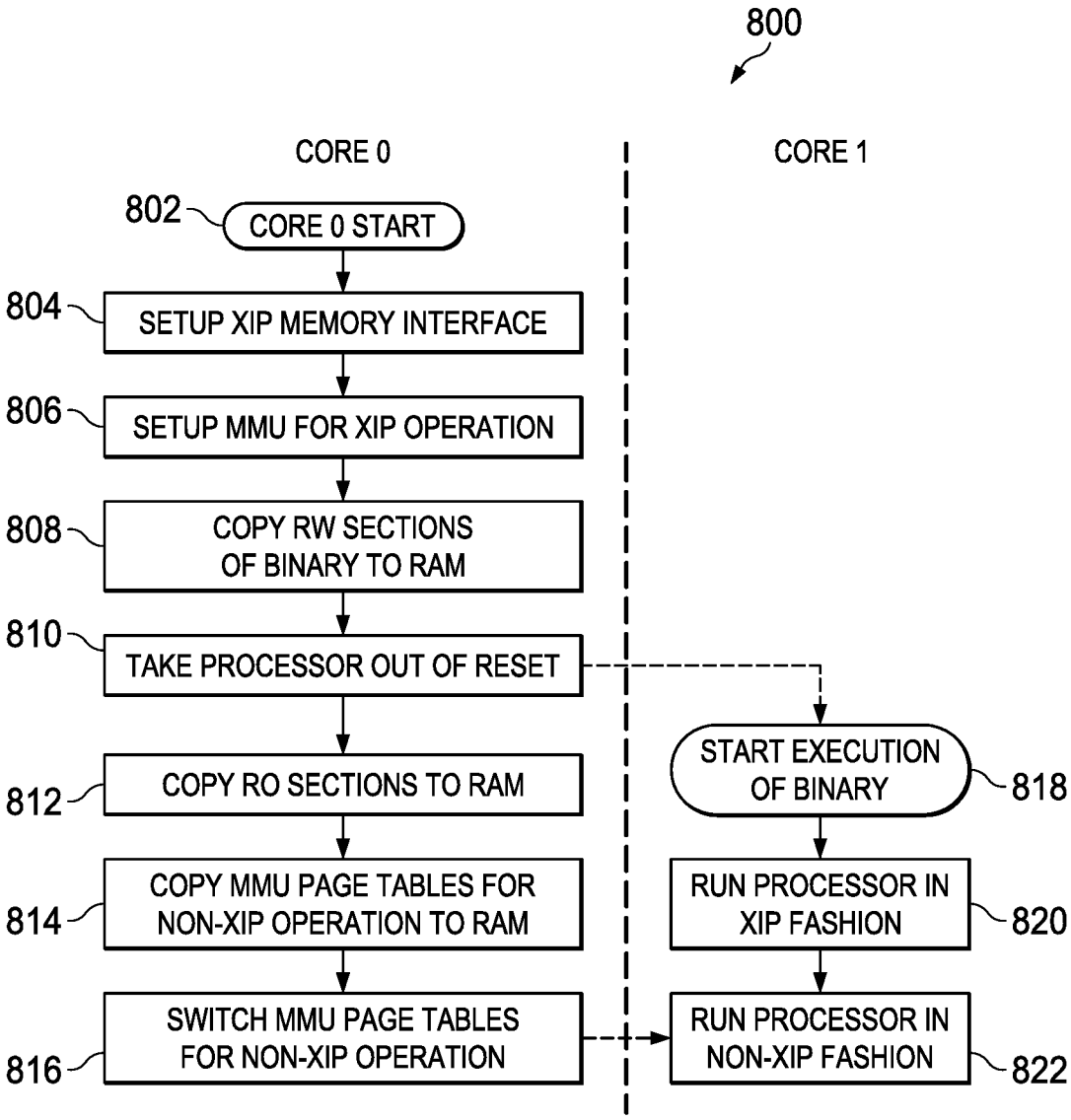

800
CORE 0
CORE 1
802
CORE 0 START
804
SETUP XIP MEMORY INTERFACE
806
SETUP MMU FOR XIP OPERATION
808
COPY RW SECTIONS OF BINARY TO RAM
810
TAKE PROCESSOR OUT OF RESET
812
COPY RO SECTIONS TO RAM
814
COPY MMU PAGE TABLES FOR NON-XIP OPERATION TO RAM
816
SWITCH MMU PAGE TABLES FOR NON-XIP OPERATION
START EXECUTION OF BINARY
818
RUN PROCESSOR IN XIP FASHION
820
RUN PROCESSOR IN NON-XIP FASHION
822

FIG. 8

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US 2018/034142

A. CLASSIFICATION OF SUBJECT MATTER

G06F 12/1009 *(2016.01)*
G11C 14/00 *(2006.01)*

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 12/00, 12/02, 12/08, 12/10, 12/1009, G11C 11/00, 14/00, 16/00, 16/02, 16/04

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

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

PatSearch (RUPTO internal), USPTO, PAJ, K-PION, Esp@cenet, Information Retrieval System of FIPS

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 6282624 B1 (SEIKO EPSON CORPORATION) 28.08.2001, col. 1, lines 31-33, col. 4, lines 56-59, col. 5, lines 1-9, 31-56, col. 12, lines 12-17, 25-29, 34-46, 49-57, col. 18, lines 28-42	1-20
A	US 7177184 B2 (SANDISK CORPORATION) 13.02.2007	1-20
A	US 2011/0153912 A1 (SERGEY ANATOLIEVICH GOROBETS et al.) 23.06.2011	1-20
A	US 2011/0022819 A1 (DANIEL JEFFREY POST et al.) 27.01.2011	1-20

☐ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* Special categories of cited documents:

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

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

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
12 October 2018 (12.10.2018)	18 October 2018 (18.10.2018)
Name and mailing address of the ISA/RU: Federal Institute of Industrial Property, Berezhkovskaya nab., 30-1, Moscow, G-59, GSP-3, Russia, 125993 Facsimile No: (8-495) 531-63-18, (8-499) 243-33-37	Authorized officer I. Kryazhev Telephone No. (495) 531-65-15