



- (51) **International Patent Classification:**
G06F 12/14 (2006.01) G06F 12/16 (2006.01)
- (21) **International Application Number:**
PCT/US2009/062799
- (22) **International Filing Date:**
30 October 2009 (30.10.2009)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
12/261,942 30 October 2008 (30.10.2008) US
- (71) **Applicant (for all designated States except US):** MICRON TECHNOLOGY, INC. [US/US]; 8000 So. Federal Way, Boise, Idaho 83716-9632 (US).
- (72) **Inventors; and**
- (75) **Inventors/Applicants (for US only):** JEDDELOH, Joe M. [US/US]; 4302 Reiland Lane, Shoreview, Minnesota 55126 (US). LABERGE, Paul A. [US/US]; 1335 Meadow Avenue, Shoreview, Minnesota 55126 (US).
- (74) **Agents:** MCCrackin, Ann M. et al.; Schwegman, Lundberg, & Woessner, P.A., P.O. Box 2938, Minneapolis, MN 55402-0938 (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, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, 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, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (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, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

- Published:**
- with international search report (Art. 21(3))
 - 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))
 - with information concerning one or more priority claims considered void (Rule 26bis.2(d))

(54) Title: MULTI-SERIAL INTERFACE STACKED-DIE MEMORY ARCHITECTURE

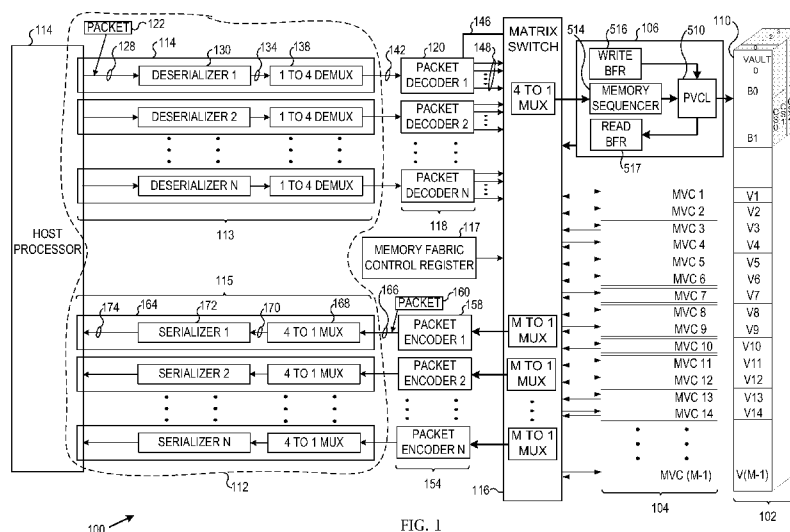


FIG. 1

(57) **Abstract:** Systems and methods disclosed herein substantially concurrently transfer a plurality of streams of commands, addresses, and/or data across a corresponding plurality of serialized communication link interfaces (SCLIs) between one or more originating devices or destination devices such as a processor and a switch. At the switch, one or more commands, addresses, or data corresponding to each stream can be transferred to a corresponding destination memory vault controller (MVC) associated with a corresponding memory vault. The destination MVC can perform write operations, read operations, and/or memory vault housekeeping operations independently from concurrent operations associated with other MVCs coupled to a corresponding plurality of memory vaults.

WO 2010/051461 A1

MULTI-SERIAL INTERFACE STACKED-DIE MEMORY ARCHITECTURE

Related Applications

5 This patent application claims priority benefit from U.S. Application No. 12/261,942 filed 30 October 2008, which is a continuation-in-part of U.S. Application Serial No. 12/166,814 filed July 2, 2008, U.S. Application Serial No. 12/166,871 filed July 2, 2008 and U.S. Application Serial No. 12/176,951 filed on July 21, 2008. These applications are incorporated herein by reference.

10

Technical Field

 Various embodiments described herein relate to apparatus, systems, and methods associated with semiconductor memories, including stacked-die memory architectures.

15

Background Information

 Microprocessor technology has evolved at a faster rate than that of semiconductor memory technology. As a result, a mis-match in performance often exists between the modern host processor and the semiconductor memory subsystem to which the processor is mated to receive instructions and data. For example, it is estimated that some high-end servers idle three out of four clocks waiting for responses to memory requests.

20

 In addition, the evolution of software application and operating system technology has increased demand for higher-density memory subsystems as the number of processor cores and threads continues to increase. However, current-technology memory subsystems often represent a compromise between performance and density. Higher bandwidths may limit the number of memory cards or modules that may be connected in a system without exceeding Joint Electron Device Engineering Council (JEDEC) electrical specifications.

25

 Extensions to JEDEC interface standards such as dynamic data rate (DDR) synchronous dynamic random access memory (SDRAM) have been

30

proposed but may be generally found lacking as to future anticipated memory bandwidths and densities. Weaknesses include lack of memory power optimization and the uniqueness of the interface between the host processor and the memory subsystem. The latter weakness may result in a need to redesign the interface as processor and/or memory technologies change.

Brief Description of the Drawings

FIG. 1 is a block diagram of a memory system according to various example embodiments of the current invention.

FIG. 2 is a cut-away conceptual view of a stacked-die 3D memory array stacked with a logic die according to various example embodiments.

FIGS. 3 and 4 are packet diagrams showing fields associated with example packets according to various example embodiments.

FIG. 5 is a block diagram of a memory vault controller and associated modules according to various example embodiments.

FIGS. 6A and 6B are flow diagrams illustrating a method according to various example embodiments.

FIGS. 7A and 7B are flow diagrams illustrating a method according to various example embodiments.

Detailed Description

FIG. 1 is a block diagram of a memory system 100 according to various example embodiments of the current invention. The memory system 100 operates to substantially concurrently transfer a plurality of outbound and/or inbound streams of commands, addresses, and/or data between one or more originating devices and/or destination devices (e.g., one or more processors) and a set of stacked-array memory “vaults.” Increased memory system density, bandwidth, parallelism, and scalability may result.

Multi-die memory array embodiments herein aggregate control logic that is normally located on each individual memory array die in previous designs. Subsections of a stacked group of dies, referred to herein as a “memory vault,” share common control logic. The memory vault architecture strategically

partitions memory control logic to increase energy efficiency while providing a finer granularity of powered-on memory banks. Embodiments herein also enable a standardized host processor to memory system interface. The standardized interface may reduce re-design cycle times as memory technology
5 evolves.

FIG. 2 is a cut-away conceptual view of a stacked-die 3D memory array 200 stacked with a logic die 202 according to various example embodiments. The memory system 100 incorporates one or more stacks of tiled memory arrays such as the stacked-die 3D memory array 200. Multiple memory arrays (e.g., the
10 memory array 203) are fabricated onto each of a plurality of dies (e.g., the die 204). The memory array dies are then stacked to form the stacked-die 3D memory array 200.

Each of the stacked dies is divided into multiple “tiles” (e.g., the tiles 205A, 205B, and 205C associated with the stacked die 204). Each tile (e.g., the
15 tile 205C) may include one or more memory arrays 203. In some embodiments, each memory array 203 may be configured as one or more independent memory banks in the memory system 100. The memory arrays 203 are not limited to any particular memory technology and may include dynamic random-access memory (DRAM), static random access memory (SRAM), flash memory, etc.

20 A stacked set of memory array tiles 208 may include a single tile from each of the stacked dies (e.g., the tiles 212B, 212C and 212D, with the base tile hidden from view in FIG. 1). Power, address, and/or data and similar common signals may traverse the stacked set of tiles 208 in the “Z” dimension 220 on
25 conductive paths (e.g., the conductive path 224) referred to herein as “through-wafer interconnects” (TWIs). It is noted that a TWI need not necessarily pass entirely through a particular wafer or die.

The stacked-die 3D memory array 200 is thus partitioned into a set of memory “vaults” (e.g., the memory vault 230). Each memory vault includes a stacked set of tiles (e.g., the set of tiles 208), one tile from each of a plurality of
30 stacked dies, together with a set of TWIs to electrically interconnect the set of tiles 208. Each tile of the vault includes one or more memory arrays (e.g., the memory array 240).

The resulting set of memory vaults 102 is shown in FIG. 1. The memory system 100 also includes a plurality of memory vault controllers (MVCs) 104 (e.g., the MVC 106). Each MVC is communicatively coupled to a corresponding memory vault (e.g., the memory vault 110 of the set 102) in a one-to-one relationship. Each MVC is thus capable of communicating with a corresponding memory vault independently from communications between other MVCs and their respective memory vaults.

The memory system 100 also includes a plurality of configurable serialized communication link interfaces (SCLIs) 112. The SCLIs 112 are divided into an outbound group of SCLIs 113 (e.g., the outbound SCLI 114) and an inbound group of SCLIs 115. Each of the plurality of SCLIs 112 is capable of concurrent operation with the other SCLIs 112. Together the SCLIs 112 communicatively couple the plurality of MVCs 104 to one or more host processor(s) 114. The memory system 100 presents a highly abstracted, multi-link, high-throughput interface to the host processor(s) 114.

The memory system 100 may also include a switch 116. In some embodiments, the switch 116 may comprise a matrix or cross-connect switch. The switch 116 is communicatively coupled to the plurality of SCLIs 112 and to the plurality of MVCs 104. The switch 116 is capable of cross-connecting each SCLI to a selected MVC. The host processor(s) 114 may thus access the plurality of memory vaults 102 across the plurality of SCLIs 112 in a substantially simultaneous fashion. This architecture can provide the processor-to-memory bandwidth needed by modern processor technologies, including multi-core technologies.

The memory system 100 may also include a memory fabric control register 117 coupled to the switch 116. The memory fabric control register 117 accepts memory fabric configuration parameters from a configuration source and configures one or more components of the memory system 100 to operate according to a selectable mode. For example, the switch 116 and each of the plurality of memory vaults 102 and the plurality of MVCs 104 may normally be configured to operate independently of each other in response to separate memory requests. Such a configuration may enhance memory system bandwidth as a result of the parallelism between the SCLIs 112 and the memory vaults 102.

Alternatively, the memory system 100 may be reconfigured via the memory fabric control register 117 to cause a subset of two or more of the plurality of memory vaults 102 and a corresponding subset of MVCs to operate synchronously in response to a single request. The latter configuration may be used to access a data word that is wider than the width of a data word associated with a single vault. This technique may decrease latency, as further described below. Other configurations may be enabled by loading a selected bit pattern into the memory fabric control register 117.

FIGS. 3 and 4 are packet diagrams showing fields associated with example packets 300 and 400, respectively, according to various example embodiments. Turning to FIG. 1 in light of FIGS. 3 and 4, the memory system 100 may also include a plurality of packet decoders 118 (e.g., the packet decoder 120) coupled to the switch 116. The host processor(s) 114 assemble an outbound packet 122 that in some embodiments may be similar in structure to the example command packet 300 or data packet 400. That is, the outbound packet 122 may contain a command field 310, an address field 320, and/or a data field 410. The host processor 114 then sends the outbound packet 122 across an outbound SCLI (e.g., the outbound SCLI 114) to the packet decoder 120 in a manner further explained below.

The outbound SCLI 114 may include a plurality of outbound differential pair serial paths (DPSPs) 128. The DPSPs 128 are communicatively coupled to the host processor(s) 114 and may collectively transport the outbound packet 122. That is, each DPSP of the plurality of outbound DPSPs 128 may transport a first data rate outbound sub-packet portion of the outbound packet 122 at a first data rate.

The outbound SCLI 114 may also include a deserializer 130 coupled to the plurality of outbound DPSPs 128. The deserializer 130 converts each first data rate outbound sub-packet portion of the outbound packet 122 to a plurality of second data rate outbound sub-packets. The plurality of second data rate outbound sub-packets is sent across a first plurality of outbound single-ended data paths (SEDPs) 134 at a second data rate. The second data rate is slower than the first data rate.

The outbound SCLI 114 may also include a demultiplexer 138 communicatively coupled to the deserializer 130. The demultiplexer 138 converts each of the plurality of second data rate outbound sub-packets to a plurality of third data rate outbound sub-packets. The plurality of third data rate
5 outbound sub-packets is sent across a second plurality of outbound SEDPs 142 to the packet decoder 120 at a third data rate. The third data rate is slower than the second data rate.

The packet decoder 120 receives the outbound packet 122 and extracts the command field 310 (e.g., of the example packet 300), the address field 320
10 (e.g., of the example packet 300), and/or the data field (e.g., of the example packet 400). In some embodiments, the packet decoder 120 decodes the address field 320 to determine a corresponding set of memory vault select signals. The packet decoder 120 presents the set of memory vault select signals to the switch 116 on an interface 146. The vault select signals cause the input data paths 148
15 to be switched to the MVC 106 corresponding to the outbound packet 122.

Turning now to a discussion of the inbound data paths, the memory system 100 may include a plurality of packet encoders 154 (e.g., the packet encoder 158) coupled to the switch 116. The packet encoder 158 may receive an inbound memory command, an inbound memory address, and/or inbound
20 memory data from one of the plurality of MVCs 104 via the switch 116. The packet encoder 158 encodes the inbound memory command, address, and/or data into an inbound packet 160 for transmission across an inbound SCLI 164 to the host processor(s) 114.

In some embodiments, the packet encoder 158 may segment the inbound packet 158 into a plurality of third data rate inbound sub-packets. The packet
25 encoder 158 may send the plurality of third data rate inbound sub-packets across a first plurality of inbound single-ended data paths (SEDPs) 166 at a third data rate. The memory system 100 may also include a multiplexer 168 communicatively coupled to the packet encoder 158. The multiplexer 168 may
30 multiplex each of a plurality of subsets of the third data rate inbound sub-packets into a second data rate inbound sub-packet. The multiplexer 168 sends the second data rate inbound sub-packets across a second plurality of inbound SEDPs 170 at a second data rate that is faster than the third data rate.

The memory system 100 may further include a serializer 172 communicatively coupled to the multiplexer 168. The serializer 172 aggregates each of a plurality of subsets of the second data rate inbound sub-packets into a first data rate inbound sub-packet. The first data rate inbound sub-packets are sent to the host processor(s) 114 across a plurality of inbound differential pair serial paths (DPSPs) 174 at a first data rate that is faster than the second data rate. Command, address, and data information is thus transferred back and forth between the host processor(s) 114 and the MVCs 104 across the SCLIs 112 via the switch 116. The MVCs 104, the SCLIs 112, and the switch 116 are fabricated on the logic die 202.

FIG. 5 is a block diagram of an MVC (e.g., the MVC 106) and associated modules according to various example embodiments. The MVC 106 may include a programmable vault control logic (PVCL) component (e.g., the PVCL 510). The PVCL 510 interfaces the MVC 106 to the corresponding memory vault (e.g., the memory vault 110). The PVCL 510 generates one or more bank control signals and/or timing signals associated with the corresponding memory vault 110.

The PVCL 510 may be configured to adapt the MVC 106 to a memory vault 110 of a selected configuration or a selected technology. Thus, for example, the memory system 100 may initially be configured using currently-available DDR2 DRAMs. The memory system 100 may subsequently be adapted to accommodate DDR3-based memory vault technology by reconfiguring the PVCL 510 to include DDR3 bank control and timing logic.

The MVC 106 may also include a memory sequencer 514 communicatively coupled to the PVCL 510. The memory sequencer 514 performs a memory technology dependent set of operations based upon the technology used to implement the associated memory vault 110. The memory sequencer 514 may, for example, perform command decode operations, memory address multiplexing operations, memory address demultiplexing operations, memory refresh operations, memory vault training operations, and/or memory vault prefetch operations associated with the corresponding memory vault 110. In some embodiments, the memory sequencer 514 may comprise a DRAM

sequencer. In some embodiments, memory refresh operations may originate in a refresh controller 515.

The memory sequencer 514 may be configured to adapt the memory system 100 to a memory vault 110 of a selected configuration or technology.

5 For example, the memory sequencer 514 may be configured to operate synchronously with other memory sequencers associated with the memory system 100. Such a configuration may be used to deliver a wide data word from multiple memory vaults to a cache line (not shown) associated with the host processor(s) 114 in response to a single cache line request.

10 The MVC 106 may also include a write buffer 516. The write buffer 516 may be coupled to the PVCL 510 to buffer data arriving at the MVC 106 from the host processor(s) 114. The MVC 106 may further include a read buffer 517. The read buffer 517 may be coupled to the PVCL 510 to buffer data arriving at the MVC 106 from the corresponding memory vault 110.

15 The MVC 106 may also include an out-of-order request queue 518. The out-of-order request queue 518 establishes an ordered sequence of read and/or write operations to the plurality of memory banks included in the memory vault 110. The ordered sequence is chosen to avoid sequential operations to any single memory bank in order to reduce bank conflicts and to decrease read-to-write turnaround time.

20 The MVC 106 may also include a memory vault repair logic (MVRL) component 524. The MVRL 524 may perform defective memory array address remapping operations using array repair logic 526. The array repair logic 526 may remap requests to redundant cells or arrays of cells located on memory vault dies (e.g., on the stacked die 204 of FIG. 2) and/or on the logic die 202 (e.g., the spare array 527). The MVRL 524 may also perform TWI repair operations associated with the corresponding memory vault 110 using TWI repair logic 528.

30 Any of the components previously described may be implemented in a number of ways, including embodiments in hardware, software, firmware, or combinations thereof. It is noted that “software” in this context refers to statutory software structures and not to mere software listings.

Thus, the memory system 100; the memory arrays 200, 203, 240, 527; the die 202, 204; the tiles 205A, 205B, 205C, 208, 212B, 212C, 212D; the “Z” dimension 220; the paths 224, 148; the memory vaults 230, 102, 110; the MVCs 104, 106; the SCLIs 112, 113, 114, 115, 164; the processor(s) 114; the switch 5 116; the register 117; the packets 300, 400, 122, 160; the packet decoders 118, 120; the fields 310, 320, 410; the DPSPs 128, 174; the deserializer 130; the SEDPs 134, 142, 166, 170; the demultiplexer 138; the interface 146; the packet encoders 154, 158; the multiplexer 168; the serializer 172; the PVCL 510; the memory sequencer 514; the refresh controller 515; the buffers 516, 517; the out- 10 of-order request queue 518; the MVRL 524; the array repair logic 526; and the TWI repair logic 528 may all be characterized as “modules” herein.

The modules may include hardware circuitry, optical components, single or multi-processor circuits, memory circuits, software program modules and objects, firmware, and combinations thereof, as desired by the architect of the 15 memory system 100 and as appropriate for particular implementations of various embodiments.

The apparatus and systems of various embodiments may be useful in applications other than a high-density, multi-link, high-throughput semiconductor memory subsystem. Thus, various embodiments of the invention 20 are not to be so limited. The illustrations of the memory system 100 are intended to provide a general understanding of the structure of various embodiments. They are not intended to serve as a complete description of all the elements and features of apparatus and systems that might make use of the structures described herein.

The novel apparatus and systems of various embodiments may comprise 25 or be incorporated into electronic circuitry used in computers, communication and signal processing circuitry, single-processor or multi-processor modules, single or multiple embedded processors, multi-core processors, data switches, and application-specific modules including multilayer, multi-chip modules.

30 Such apparatus and systems may further be included as sub-components within a variety of electronic systems, such as televisions, cellular telephones, personal computers (e.g., laptop computers, desktop computers, handheld computers, tablet computers, etc.), workstations, radios, video players, audio players (e.g.,

MP3 (Motion Picture Experts Group, Audio Layer 3) players), vehicles, medical devices (e.g., heart monitor, blood pressure monitor, etc.), set top boxes, and others. Some embodiments may include a number of methods.

FIGS. 6A and 6B are flow diagrams illustrating a method 600 according to various example embodiments. The method 600 includes substantially concurrently transferring a plurality of outbound streams of commands, addresses, and/or data from one or more originating devices (e.g., the processor(s) 114 of FIG. 1) to a plurality of memory vaults (e.g., the memory vaults 102 of FIG. 1). The streams may be packetized and transported from the originating device(s) across a plurality of outbound SCLIs (e.g., the outbound SCLIs 113 of FIG. 1) to a set of packet decoders (e.g., the packet decoders 118 of FIG. 1). The command, address, and data streams may then be switched to corresponding MVCs (e.g., the MVCs 104) for execution and/or writing to or reading from the memory vaults.

The method 600 may commence at block 606 with segmenting an outbound packet into a set of first data rate sub-packet portions at the originating device. In some embodiments, the originating device may include one or more processors. In some embodiments, the originating device may include a category of devices capable of direct memory access (DMA) such as a graphics controller. The packet may carry one or more outbound memory subsystem commands, addresses, or data fields to be written to one or more memory subsystem locations.

The method 600 may continue at block 610 with sending each of the first data rate sub-packets from the originating device (e.g., from a selected processor) to a deserializer (e.g., the deserializer 130 of FIG. 1). The first data rate sub-packets may be sent across a plurality of DPSPs (e.g., the DPSPs 128 of FIG. 1) corresponding to a selected outbound SCLI (e.g., the outbound SCLI 114 of FIG. 1) at a first data rate. The method 600 may also include segmenting each of the first data rate sub-packets into a plurality of second data rate sub-packets at the deserializer, at block 612.

The method 600 may further include sending each of the second data rate sub-packets from the deserializer to a demultiplexer (e.g., the demultiplexer 138 of FIG. 1) at a second data rate slower than the first data rate, at block 614. At

the demultiplexer, each of the second data rate sub-packets may be segmented into a set of third data rate sub-packets, as depicted at block 616. The method 600 may also include sending the third data rate sub-packets to a packet decoder at a third data rate slower than the second data rate, at block 618.

5 The method 600 may continue at block 622 with receiving the third data rate sub-packets at the packet decoder from the selected SCLI. The method 600 may include assembling the set of third data rate sub-packets into the outbound packet, at block 626. The method 600 may also include extracting at least one of the outbound command, the outbound address, or the outbound data from the
10 packet, at block 628.

 The method 600 may also include presenting the outbound command, address, or data to the switch, at block 632. The method 600 may further include concurrently switching an outbound command, address, and/or data associated with each stream at the switch, at block 636. The outbound command, address,
15 and/or data associated with each stream is switched to a destination MVC (e.g., the MVC 106 of FIG. 1) associated with a corresponding memory vault (e.g., the memory vault 110 of FIG. 1).

 The method 600 may continue at block 640 with buffering the outbound command, address, and/or data at a write buffer component of the MVC (e.g.,
20 the write buffer 516 of FIG. 1). The method 600 may include presenting the outbound command, address, and/or data to a memory sequencer (e.g., the memory sequencer 514 of FIG. 1) for processing, at block 644.

 In some embodiments, the method 600 may optionally include determining whether the memory subsystem has been configured to operate in a
25 synchronous parallel mode, at block 645. If so, the method 600 may include operating a synchronous subset of the memory vaults in response to a single memory request, at block 646. Such operation may be used to decrease access latency by synchronously transferring a wide data word of a width that is a multiple of a single memory vault word length. The resulting wide data word
30 width corresponds to the number of memory vaults in the synchronous subset of vaults.

 The method 600 may optionally include ordering read and/or write operations to a plurality of memory banks associated with a corresponding

memory vault at an out-of-order request queue component of the memory sequencer (e.g., the out-of-order request queue 518 of FIG. 5), at block 648. The ordering may operate to avoid multiple sequential reads and/or writes to any single memory bank and may thereby reduce bank conflicts and decrease read-to-write turnaround times.

The method 600 may conclude at block 650 with performing data write operations to write the outbound data to the corresponding memory vault, data read operations to read data from the corresponding memory vault, and/or memory vault housekeeping operations. The data write operations, data read operations, and/or housekeeping operations may be performed independently from concurrent operations associated with other MVCs coupled to other memory vaults.

FIGS. 7A and 7B are flow diagrams illustrating a method 700 according to various example embodiments. The method 700 includes substantially concurrently transferring a plurality of inbound streams of packetized commands, addresses, and/or data from a plurality of memory vaults (e.g., the memory vaults 102 of FIG. 1) to one or more destination devices (e.g., the processor(s) 114 of FIG. 1). The command, address, and/or data streams may be read from the memory vaults by a set of MVCs associated with the memory vaults (e.g., the MVCs 104 of FIG. 1) and/or may originate at the MVCs. The streams may be switched through a switch (e.g., the switch 116 of FIG. 1) to a set of packet encoders (e.g., the packet encoders 154 of FIG. 1). The streams may then be packetized and transported to the destination device(s) across a plurality of inbound SCLIs (e.g., the inbound SCLIs 115 of FIG. 1).

The method 700 may commence at block 706 with receiving a read command from a processor at an MVC (e.g., the MVC 106 of FIG. 1) corresponding to a selected memory vault (e.g., the memory vault 110 of FIG. 1). It is noted that in some embodiments, the processor and the destination device may be the same device; however this need not be the case. The method 700 may continue at block 710 with accessing an inbound data word from a selected memory bank associated with the memory vault using a memory sequencer (e.g., the memory sequencer 514 of FIG. 1) associated with the MVC.

The method 700 may include presenting the inbound data word to the switch, at block 714.

5 The method 700 may also include switching the inbound data word to a packet encoder (e.g., the packet encoder 158 of FIG. 1) associated with a selected SCLI (e.g., the inbound SCLI 164) using the switch, at block 718. The method 700 may further include packetizing the inbound data word into an inbound packet using the packet encoder, at block 722.

10 The method 700 may continue at block 726 with segmenting the inbound packet into a plurality of third data rate inbound sub-packets. The method 700 may include sending the plurality of third data rate inbound sub-packets to a multiplexer (e.g., the multiplexer 168 of FIG. 1) at a third data rate, at block 734. The method 700 may also include multiplexing each of a plurality of subsets of the third data rate inbound sub-packets into a second data rate inbound sub-packet using the multiplexer, at block 738. The method 700 may further include
15 sending the second data rate inbound sub-packets to a serializer (e.g., the serializer 172 of FIG. 1) at a second data rate, at block 742.

The method 700 may continue at block 746 with aggregating each of a plurality of subsets of the second data rate inbound sub-packets into a first data rate inbound sub-packet using the serializer. The method 700 may include
20 presenting the first data rate inbound sub-packets to the destination device(s), at block 754. The method 700 may also include assembling the first data rate inbound sub-packets into the inbound packet, at block 758. The method 700 may conclude with extracting the inbound data word from the inbound packet, at block 762, and presenting the inbound data word to an operating system
25 associated with the destination device(s), at block 768.

It is noted that the activities described herein may be executed in an order other than the order described. The various activities described with respect to the methods identified herein may also be executed in repetitive, serial, and/or parallel fashion.

30 A software program may be launched from a computer-readable medium in a computer-based system to execute functions defined in the software program. Various programming languages may be employed to create software programs designed to implement and perform the methods disclosed herein. The

programs may be structured in an object-oriented format using an object-oriented language such as Java or C++. Alternatively, the programs may be structured in a procedure-oriented format using a procedural language, such as assembly or C. The software components may communicate using well-known
5 mechanisms, including application program interfaces, inter-process communication techniques, and remote procedure calls, among others. The teachings of various embodiments are not limited to any particular programming language or environment.

The apparatus, systems, and methods described herein may operate to
10 substantially concurrently transfer a plurality of streams of commands, addresses, and/or data between one or more originating and/or destination devices (e.g., one or more processors) and a set of stacked-array memory vaults. Increased memory system density, bandwidth, parallelism, and scalability may result.

15 By way of illustration and not of limitation, the accompanying figures show specific embodiments in which the subject matter may be practiced. The embodiments illustrated are described in sufficient detail to enable those skilled in the art to practice the teachings disclosed herein. Other embodiments may be used and derived therefrom, such that structural and logical substitutions and
20 changes may be made without departing from the scope of this disclosure. This Detailed Description, therefore, is not to be taken in a limiting sense.

Such embodiments of the inventive subject matter may be referred to herein individually or collectively by the term "invention" merely for
25 convenience and without intending to voluntarily limit this application to any single invention or inventive concept, if more than one is in fact disclosed. Thus, although specific embodiments have been illustrated and described herein, any arrangement calculated to achieve the same purpose may be substituted for the specific embodiments shown. This disclosure is intended to cover any and all adaptations or variations of various embodiments. Combinations of the
30 above embodiments and other embodiments not specifically described herein will be apparent to those of skill in the art upon studying the above description.

The Abstract of the Disclosure is provided to comply with 37 C.F.R. §1.72(b) requiring an abstract that will allow the reader to quickly ascertain the

nature of the technical disclosure. It is submitted with the understanding that it will not be used to interpret or limit the scope or meaning of the claims. In the foregoing Detailed Description, various features are grouped together in a single embodiment for the purpose of streamlining the disclosure. This method of
5 disclosure is not to be interpreted to require more features than are expressly recited in each claim. Rather, inventive subject matter may be found in less than all features of a single disclosed embodiment. Thus the following claims are hereby incorporated into the Detailed Description, with each claim standing on its own as a separate embodiment.

Claims

What is claimed is:

1. A memory system, comprising:
a plurality of memory vaults comprising a stacked plurality of tiled memory arrays, each memory vault comprising a stacked plurality of memory array die sections;
5 a plurality of memory vault controllers (MVCs), each MVC coupled to a corresponding one of the plurality of memory vaults in a one-to-one relationship and capable of communicating with the corresponding memory vault independently from communications between others of the plurality of MVCs and memory vaults corresponding to the others of the plurality of MVCs; and
10 a plurality of configurable serialized communication link interfaces (SCLIs) capable of concurrent operation to communicatively couple the plurality of MVCs to at least one of an originating device or a destination device.
2. The memory system of claim 1, wherein the stacked plurality of memory array die sections comprises a single tile from each of the stacked plurality of tiled memory arrays.
15
3. The memory system of claim 1, wherein the plurality of memory array die sections corresponding to each memory vault are interconnected using through-wafer interconnects (TWIs).
4. The memory system of claim 1, further comprising:
20 a switch coupled to the plurality of SCLIs and to the plurality of MVCs to cross-connect a selected SCLI to a selected MVC.
5. The memory system of claim 4, further comprising:
a packet decoder coupled to the switch to receive an outbound packet sent from
25 the originating device across an outbound one of the plurality of SCLIs, to extract at least one of an outbound memory command, an outbound memory address, or an

outbound memory data field from the outbound packet and to present a set of memory vault select signals to the switch; and

5 a packet encoder coupled to the switch to receive at least one of an inbound memory command, an inbound memory address, or inbound memory data from one of the plurality of MVCs and to encode the inbound memory address or the inbound memory data into an inbound packet for transmission across an inbound one of the plurality of SCLIs to the destination device.

6. The memory system of claim 4, further comprising:

10 a control register coupled to the switch to accept configuration parameters from a configuration source and to configure the switch to operate according to a selectable mode.

7. The memory system of claim 6, wherein the selectable mode includes at least one of operating the plurality of memory vaults synchronously in response to at least one request or operating the plurality of memory vaults asynchronously in response to at least one request.

8. The memory system of claim 1, an outbound one of the plurality of SCLIs further comprising:

20 a plurality of outbound differential pair serial paths (DPSPs) coupled to the originating device to transport an outbound packet containing at least one of command information, address information, or data across the outbound SCLI, each of the plurality of outbound DPSPs to transport a first data rate outbound sub-packet portion of the outbound packet at a first data rate;

25 a deserializer coupled to the plurality of outbound DPSPs to convert each first data rate outbound sub-packet portion of the outbound packet to a plurality of second data rate outbound sub-packets for transmission across a first plurality of outbound single-ended data paths (SEDPs) at a second data rate, the second data rate slower than the first data rate; and

30 a demultiplexer communicatively coupled to the deserializer to convert each of the plurality of second data rate outbound sub-packets to a plurality of third data rate outbound sub-packets for transmission across a second plurality of outbound SEDPs

to a packet decoder at a third data rate, the third data rate slower than the second data rate.

9. The memory system of claim 1, further comprising:

5 a packet encoder communicatively coupled to the switch to packetize at least one of command information, address information, or data received from an MVC associated with the selected memory vault for transmission across an inbound one of the plurality of SCLIs to the destination device, to segment a resulting inbound packet into a plurality of third data rate inbound sub-packets, and to send the third
10 data rate inbound sub-packets across a first plurality of inbound single-ended data paths (SEDPs) at a third data rate;

a multiplexer communicatively coupled to the packet encoder to multiplex each of a plurality of subsets of the third data rate inbound sub-packets into a second data rate inbound sub-packet and to send the second data rate inbound sub-packets across
15 a second plurality of inbound SEDPs at a second data rate faster than the third data rate; and

a serializer communicatively coupled to the multiplexer to aggregate each of a plurality of subsets of the second data rate inbound sub-packets into a first data rate inbound sub-packet and to send the first data rate inbound sub-packets to the
20 destination device across a plurality of inbound differential pair serial paths (DPSPs) at a first data rate faster than the second data rate.

10. The memory system of claim 1, each MVC further comprising:

a programmable vault control logic (PVCL) component to interface the MVC
25 to the corresponding memory vault and to generate at least one of bank control signals or timing signals associated with the corresponding memory vault;

a memory sequencer coupled to the PVCL, the memory sequencer to perform at least one of command decode operations, memory address multiplexing operations, memory address demultiplexing operations, memory refresh operations, memory
30 vault training operations, or memory vault prefetch operations associated with the corresponding memory vault;

a write buffer coupled to the PVCL to buffer data arriving at the MVC from the originating device; and

a read buffer coupled to the PVCL to buffer data arriving at the MVC from the corresponding memory vault.

11. The memory system of claim 10, wherein the memory sequencer comprises a
5 dynamic random access memory (DRAM) sequencer.

12. The memory system of claim 10, wherein the PVCL is configured to interface
the MVC to the corresponding memory vault and to generate the bank control signals
or timing signals according to specifications associated with a memory vault of at
10 least one of a selected configuration or a selected technology.

13. The memory system of claim 10, wherein the memory sequencer is configured
to perform the command decode operations, memory address multiplexing
operations, memory address demultiplexing operations, memory refresh operations,
15 memory vault training operations, or memory vault prefetch operations according to
specifications associated with a memory vault of at least one of a selected
configuration or a selected technology.

14. The memory system of claim 10, further comprising:
20 an out-of-order request queue associated with each MVC, the out-of-order
request queue to order at least one of read operations to a plurality of memory banks
or write operations to the plurality of memory banks to avoid sequential operations to
any single memory bank.

25 15. The memory system of claim 1, each MVC further comprising:
a memory vault repair logic (MVRL) component to perform at least one of
defective memory array address remapping operations or defective through-wafer
interconnect (TWI) repair operations associated with the corresponding memory
vault.

30 16. A method, comprising:
concurrently transferring a plurality of streams of commands, addresses, or
data;

concurrently switching at least one of a command, an address, or data associated with each stream to a corresponding destination memory vault controller (MVC) associated with a corresponding memory vault; and

5 performing at least one of data write operations to write the outbound data to the corresponding memory vault, data read operations to read data from the corresponding memory vault, or memory vault housekeeping operations associated with the corresponding memory vault, the data write operations, data read operations, or housekeeping operations to be performed independently from concurrent operations associated with others of a plurality of MVCs coupled to a corresponding
10 plurality of memory vaults.

17. The method of claim 16, further comprising:

segmenting a packet into a plurality of first data rate sub-packet portions of the packet, the packet containing at least one memory subsystem command, at least one
15 memory subsystem address, or data to be written to at least one memory subsystem location or to be read from at least one memory subsystem location;

sending each of the first data rate sub-packets at a first data rate;

segmenting each of the first data rate sub-packets into a plurality of second data rate sub-packets;

20 sending each of the second data rate sub-packets at a second data rate slower than the first data rate;

segmenting each of the second data rate sub-packet into a plurality of third data rate sub-packets; and

25 sending the third data rate sub-packets to a packet decoder at a third data rate slower than the second data rate.

18. The method of claim 16, further comprising:

receiving a set of sub-packets corresponding to a packet;

assembling the set of sub-packets into the packet;

30 extracting at least one of the outbound command, the outbound address, or the outbound data from the packet; and

presenting the outbound command, the outbound address, or the outbound data.

19. The method of claim 16, further comprising:
buffering at least one of the outbound command, the outbound address, or the
outbound data; and
presenting the outbound command, the outbound address, or the outbound data
5 for processing.
20. The method of claim 19, further comprising:
ordering at least one of read operations to a plurality of memory banks
associated with the corresponding memory vault or write operations to the plurality
10 of memory banks.
21. The method of claim 16, further comprising:
operating a synchronous subset of the corresponding plurality of memory vaults
in response to a single request to decrease access latency by synchronously
15 transferring a wide data word of a width that is a multiple of a single memory vault
word length, the multiple corresponding to a number of memory vaults in the
synchronous subset.
22. A method, comprising:
20 creating a plurality of inbound data streams from command, address, or data
read from each of a plurality of memory vaults;
switching the plurality of inbound data streams to a corresponding plurality of
serialized communication link interfaces (SCLIs); and
concurrently transferring the plurality of inbound streams across the plurality of
25 SCLIs to at least one destination device.
23. The method of claim 22, further comprising:
accessing an inbound data word from a selected memory bank associated with
the corresponding memory vault in response to a read command; and
30 presenting the inbound data word.

24. The method of claim 23, further comprising:
switching the inbound data word to a packet encoder associated with a selected one of the plurality of SCLIs;
packetizing the inbound data word into an inbound packet;
5 segmenting the inbound packet into a plurality of third data rate inbound sub-packets; and
sending the plurality of third data rate inbound sub-packets to a multiplexer at a third data rate.
- 10 25. The method of claim 24, further comprising:
multiplexing each of a plurality of subsets of the third data rate inbound sub-packets into a second data rate inbound sub-packet,
sending the second data rate inbound sub-packets at a second data rate;
aggregating each of a plurality of subsets of the second data rate inbound sub-
15 packets into a first data rate inbound sub-packet;
presenting the first data rate inbound sub-packets;
assembling the first data rate inbound sub-packets into the inbound packet; and
extracting the inbound data word from the inbound packet.
- 20 26. The method of claim 25, further comprising:
presenting the inbound data word to an operating system associated with the at least one destination device.

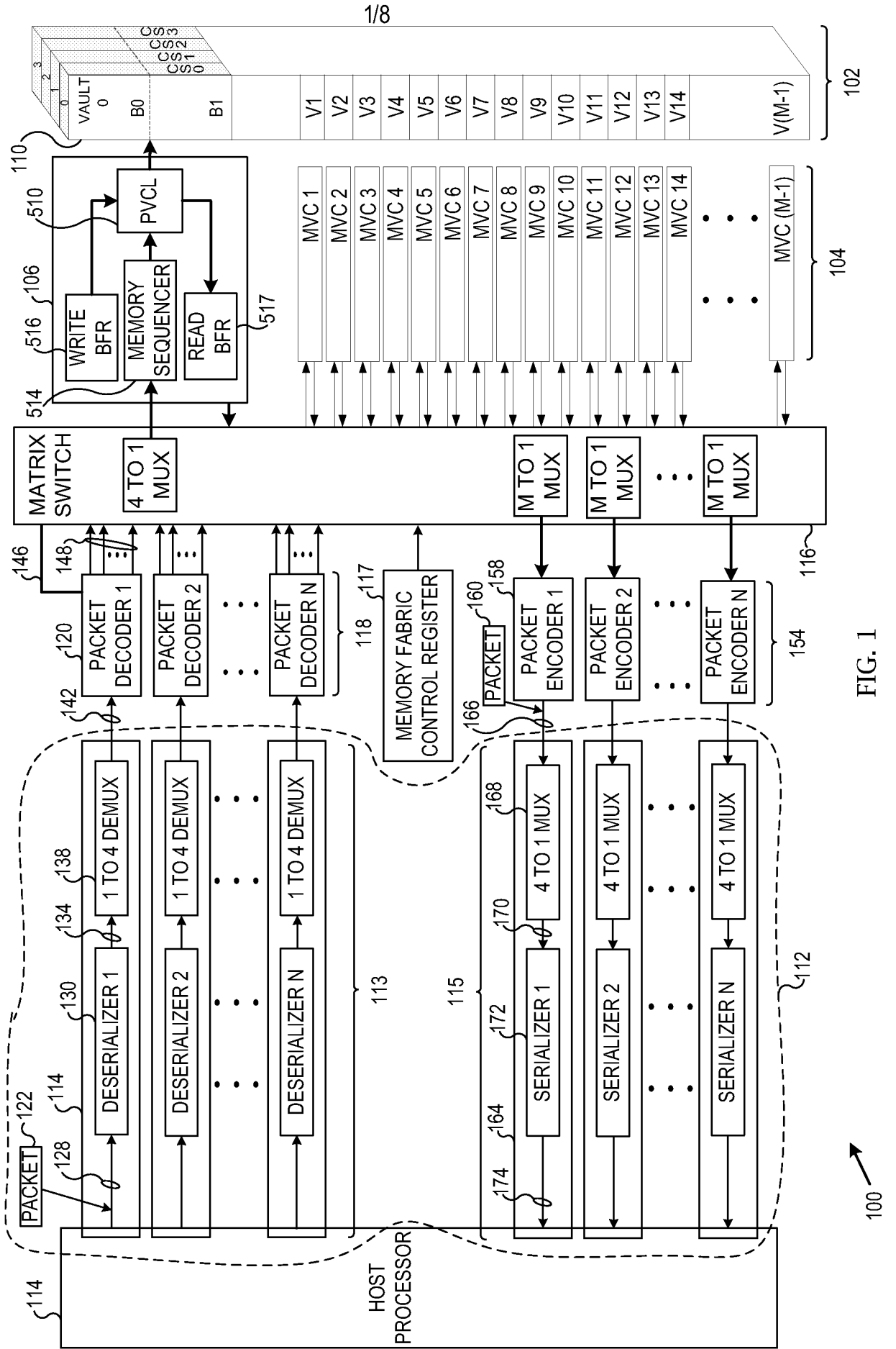


FIG. 1

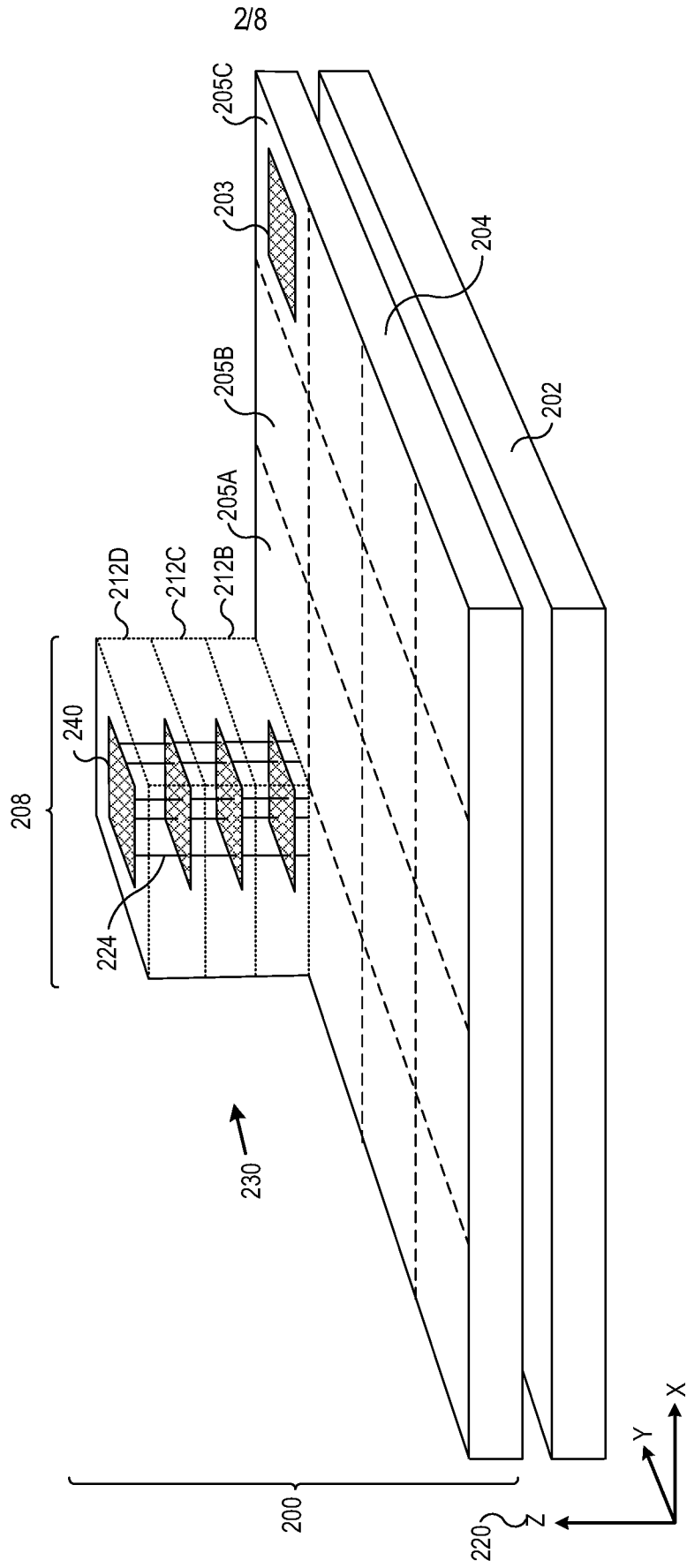


FIG. 2

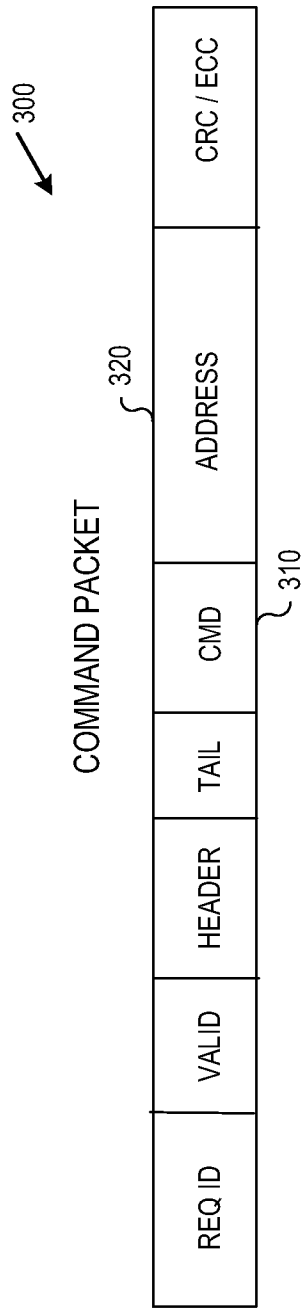


FIG. 3

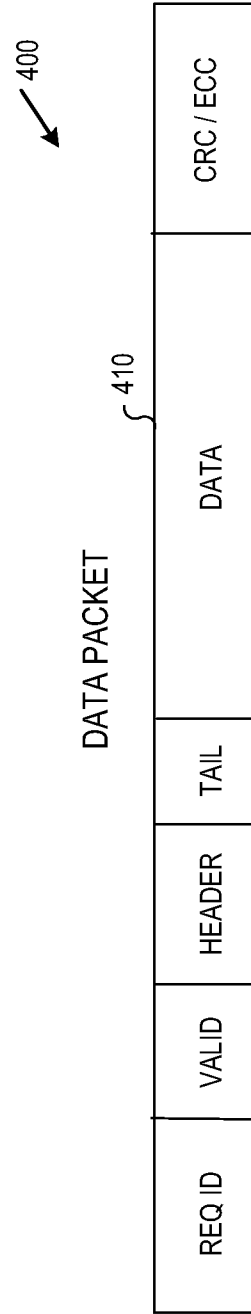


FIG. 4

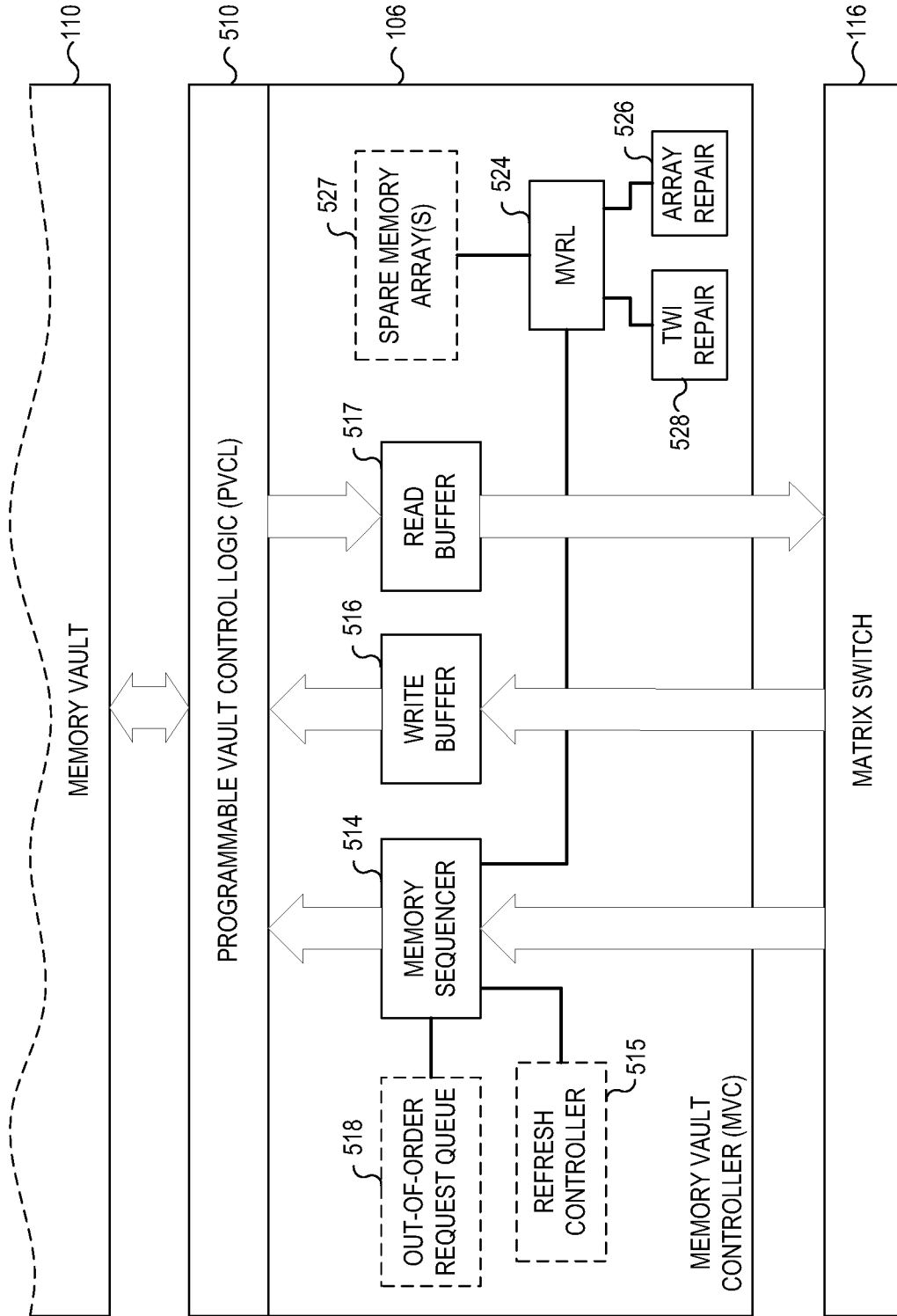


FIG. 5

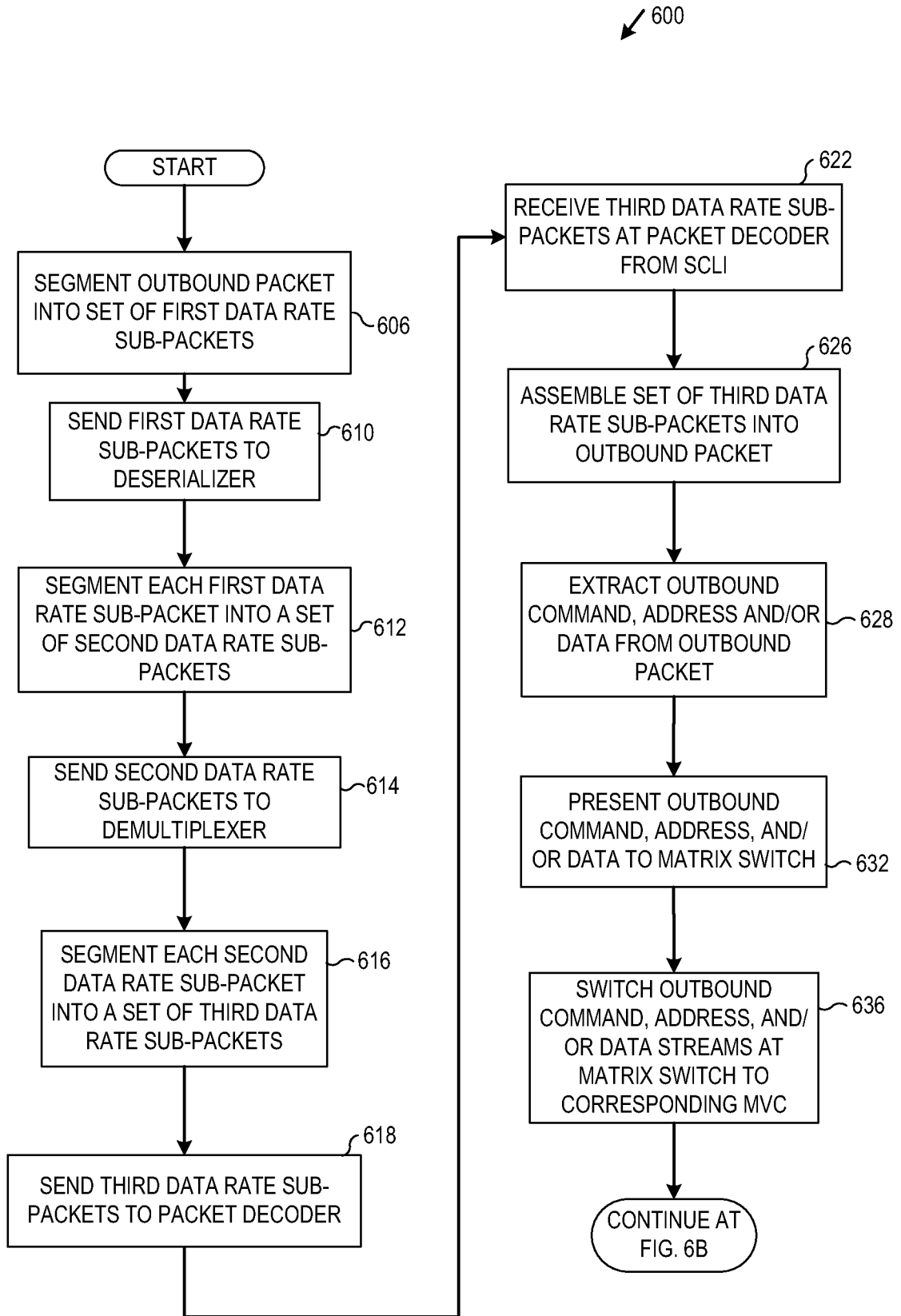


FIG. 6A

6/8

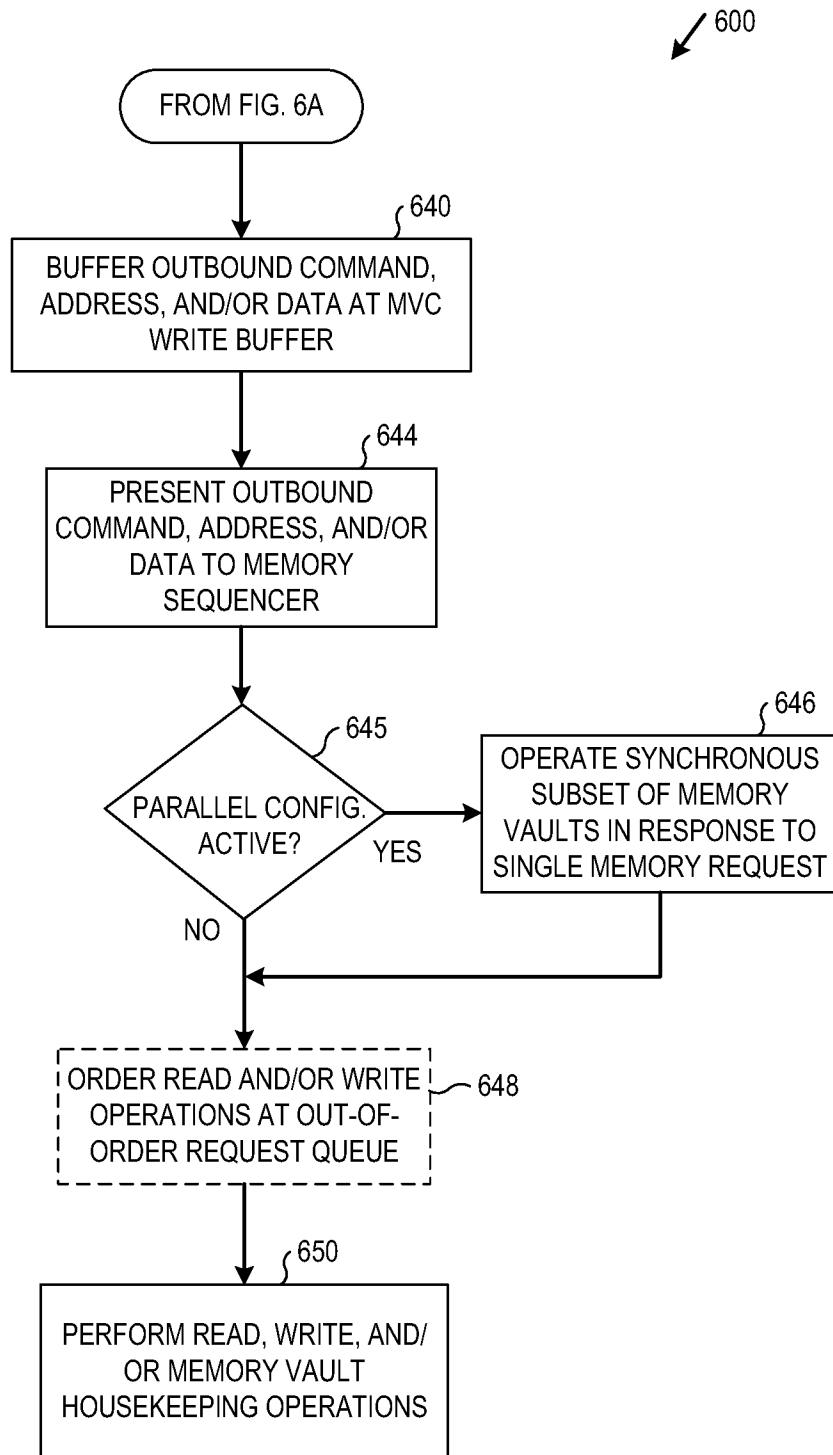


FIG. 6B

700

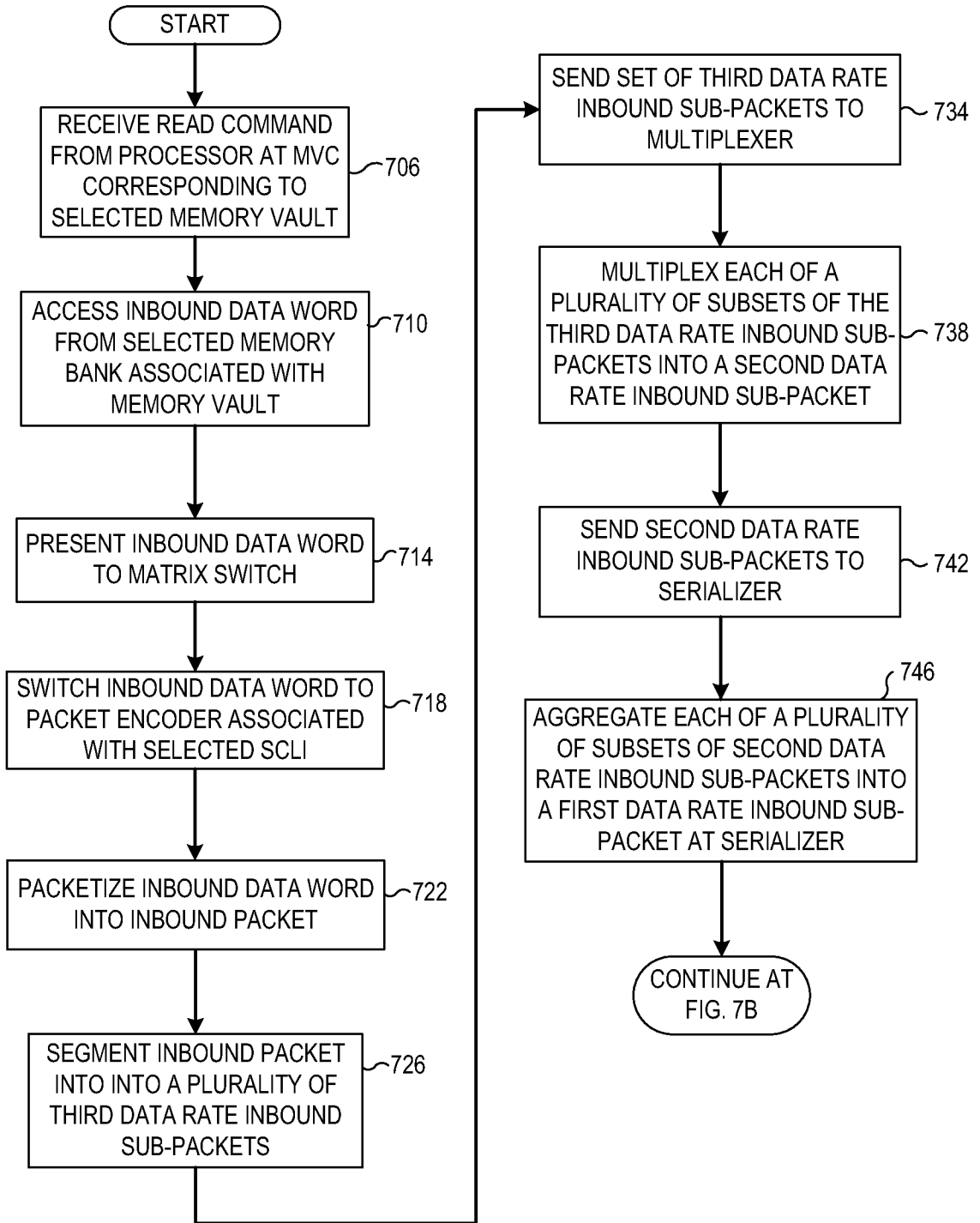


FIG. 7A

700

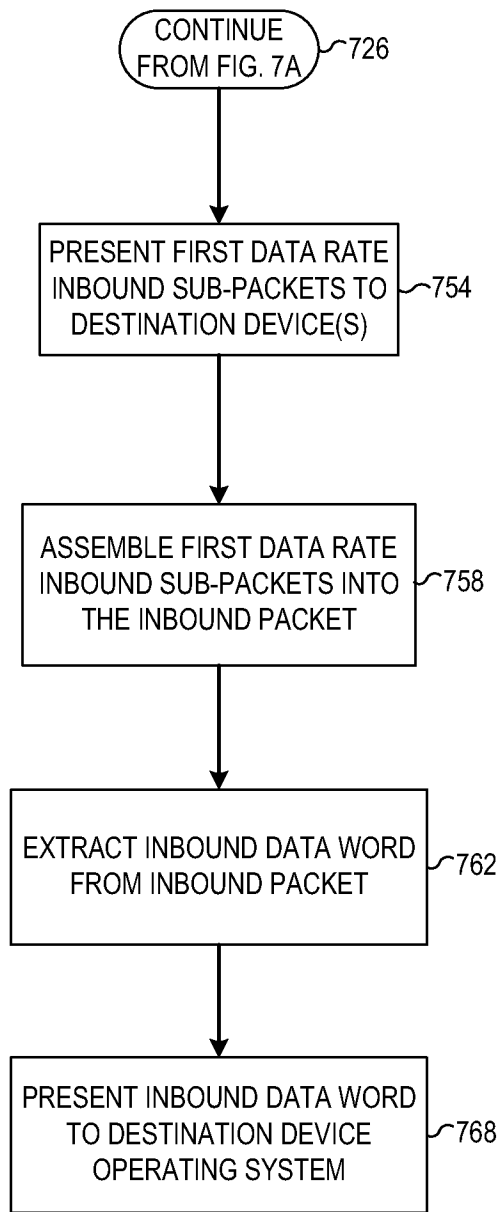


FIG. 7B

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2009/062799**A. CLASSIFICATION OF SUBJECT MATTER****G06F 12/14(2006.01)i, G06F 12/16(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 12/14; G06F 12/00; G11C 5/02; H01L 23/48

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
(Chinese Patents and application for patent)

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

eKOMPASS(KIPO internal) & Keywords: "memory", "transfer"

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2007-0070669 A1 (TSERN, E.) 29 March 2007 See Abstract; Figures 1, 10-12, 14, 18, 19; and their descriptions.	1-26
A	US 2006-0036827 A1 (DELL, T. J. et al.) 16 February 2006 See Abstract; Figures 4, 5, 7-9, 14, 15; and their descriptions.	1-26
A	US 2005-0081012 A1 (GILLINGHAM, P. et al.) 14 April 2005 See Abstract; Figures 1a, 6a, 7, 12; and their descriptions.	1-26
A	US 2005-0189639 A1 (TANIE, H. et al.) 01 September 2005 See Abstract; Figures 1, 3; and Claims 1, 7.	1-26

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

* Special categories of cited documents:

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search

26 FEBRUARY 2010 (26.02.2010)

Date of mailing of the international search report

03 MARCH 2010 (03.03.2010)

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
Government Complex-Daejeon, 139 Seonsa-ro, Seo-
gu, Daejeon 302-701, Republic of Korea

Facsimile No. 82-42-472-7140

Authorized officer

LEE, Sang Hun

Telephone No. 82-42-481-5914



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2009/062799

Patent document cited in search report	Publication date	Patent family member(s)	Publication date		
US 2007-0070669 A1	29.03.2007	CN 101310338 A	19.11.2008		
		CN 101310338 C0	19.11.2008		
		EP 1929479 A2	11.06.2008		
		JP 2009-510562 A	12.03.2009		
		US 2007-0088995 A1	19.04.2007		
		US 2008-0080261 A1	03.04.2008		
		US 2008-0144411 A1	19.06.2008		
		US 7464225 B2	09.12.2008		
		US 7562271 B2	14.07.2009		
		WO 2007-038225 A2	05.04.2007		
		WO 2007-038225 A3	14.06.2007		
		WO 2008-124503 A1	16.10.2008		
		US 2006-0036827 A1	16.02.2006	EP 1622020 A2	01.02.2006
				EP 1622020 A3	10.09.2008
JP 2006-048688 A	16.02.2006				
KR 10-0843538 B1	04.07.2008				
KR 10-2006-0049975 A	19.05.2006				
US 7539800 B2	26.05.2009				
US 2005-0081012 A1	14.04.2005	US 2001-0047450 A1	29.11.2001		
		US 2003-0065877 A1	03.04.2003		
		US 2008-0065820 A1	13.03.2008		
		US 2008-0120457 A1	22.05.2008		
		US 2008-0120458 A1	22.05.2008		
		US 6510503 B2	21.01.2003		
		US 6779097 B2	17.08.2004		
		US 7299330 B2	20.11.2007		
US 2005-0189639 A1	01.09.2005	CN 1665027 A	07.09.2005		
		CN 1665027 C0	13.08.2008		
		JP 2005-244143 A	08.09.2005		
		JP 4205613 B2	07.01.2009		
		KR 10-0602106 B1	19.07.2006		
		KR 10-2005-0088917 A	07.09.2005		
		TW 249238 B	11.02.2006		
		US 7119428 B2	10.10.2006		