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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
09 June 2022 (09.06.2022)

WIPO | PCT

(10) International Publication Number
WO 2022/119704 A2

(51) International Patent Classification:
Not classified

(21) International Application Number:
PCT/US2021/059492

(22) International Filing Date:
16 November 2021 (16.11.2021)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
63/121,131 03 December 2020 (03.12.2020) US

(71) Applicant: **RAMBUS INC.** [US/US]; 4453 N First Street, Suite 100, San Jose, CA 95134 (US).

(72) Inventors: **HAMPEL, Craig, E.**; 4453 N First Street, Suite 100, San Jose, CA 95134 (US). **LINSTADT, John, Eric**; 4453 N First Street, Suite 100, San Jose, CA 95134 (US).

(74) Agent: **NEUDECK, Alexander, J.**; P.O. Box 1792, Fraser, CO 80442 (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, IT, 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, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, 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))*

Published:
— *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

(54) Title: DYNAMICALLY CONFIGURABLE MEMORY ERROR CONTROL SCHEMES

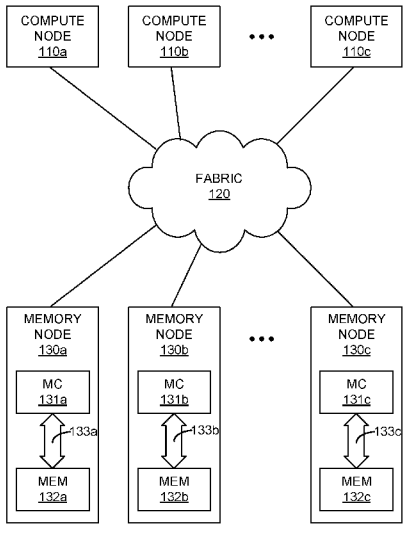

FIG. 1

(57) Abstract: A multi-host processing system may access memory devices (e.g., memory modules, memory integrated circuits, etc.) via memory nodes having memory controllers. The memory controllers may be configured to use more than one error control scheme when accessing the same memory devices. The selection of the error control scheme may be made based on the interface receiving the memory transaction request. The selection of the error control scheme may be made based on information in the memory transaction request. The selection of the error control scheme may be made based on the fabric physical address and a lookup table or address range registers. The selection of the error control scheme may be made based on the memory device physical address and a lookup table or address range registers.

WO 2022/119704 A2

DYNAMICALLY CONFIGURABLE MEMORY ERROR CONTROL SCHEMES

BRIEF DESCRIPTION OF THE DRAWINGS

[0001] Figure 1 is a block diagram illustrating a fabric interconnected processing system.

[0002] Figure 2 is a block diagram illustrating a multi-host processing system.

[0003] Figures 3A-3B are illustrations of address field assignments.

[0004] Figures 4A-4B are illustrations of example physical memory channel address field assignments.

[0005] Figures 5A-5B are illustrations of fabric physical address to physical memory channel address mappings.

[0006] Figure 6 is a flowchart illustrating a method of performing memory operations using a plurality of memory error control schemes.

[0007] Figure 7 is a flowchart illustrating a method of operating a memory controller.

[0008] Figure 8 is a flowchart illustrating a method of scheduling memory operations.

[0009] Figure 9 is a block diagram of a processing system.

DETAILED DESCRIPTION OF THE EMBODIMENTS

[0010] A multi-host processing system may access memory devices (e.g., memory modules, memory integrated circuits, etc.) via memory nodes having memory controllers. In an embodiment, the memory controllers may be configured to use more than one error control scheme (a.k.a., memory error resiliency policy) when accessing the same memory devices. For example, a memory controller may access one region of memory that is protected using a (39,32) Hamming code to provide single error correct double error detect protection/resiliency policy from one memory region and also access another region of the same memory device(s) using a more sophisticated non-binary block code (e.g., Reed-Solomon, BCH, etc.) that provides single symbol correct double symbol detect for 4-bit symbols.

[0011] In an embodiment, the selection of the error control scheme may be made based on the interface receiving the memory transaction request. In an embodiment, the selection of the error control scheme may be made based on information in the memory transaction request. In an embodiment, the selection of the error control scheme may be made based on the specific node interface that the transaction was received on. In an embodiment, the selection of the error control scheme may be made based on the fabric physical address and a lookup table or address range registers. In an embodiment, the selection of the error control

scheme may be made based on the memory physical address and a lookup table or address range registers.

[0012] Figure 1 is a block diagram illustrating a fabric interconnected processing system. In Figure 1, processing system 100 comprises compute nodes 110a-110c, fabric 120, and memory nodes 130a-130c. Memory nodes 130a-130c include memory controllers 131a-131c, respectively. Memory nodes 130a-130c include memory devices 132a-132c, respectively. Memory controllers 131a-131c are operatively coupled to memory devices 132a-132c by memory channels 133a-133c, respectively.

[0013] Compute nodes 110a-110c are each operatively coupled to fabric 120. Memory nodes 130a-130c are each operatively coupled to fabric 120. Each of compute nodes 110a-110c and memory nodes 130a-130c is operatively coupled to fabric 120 to communicate and/or exchange information, data, etc. with one or more other compute nodes 110a-110c and/or memory nodes 130a-130c.

[0014] Fabric 120 may be or comprise a switched fabric, point-to-point connections, and/or other interconnect architectures (e.g., ring topologies, crossbars, etc.) Fabric 120 may include links, linking, and/or protocols that are configured to be cache coherent. For example, fabric 120 may use links, linking, and/or protocols that include functionality described by and/or are compatible with one or more of Compute Express Link (CXL), Coherent Accelerator Processor Interface (CAPI), and Gen-Z standards, or the like. In an embodiment, compute nodes 110a-110c are operatively coupled to fabric 120 to load and/or store information from/to memory nodes 130a-130c that resides within memory devices 132a-132c.

[0015] In an embodiment, one or more memory controllers 131a-131c may be configured to use more than one error control scheme when accessing memory devices 132a-132c, respectively. For example, two or more error control policies can be implemented within a single memory node 130a-130c and in a single memory controller 131a-131c. Having memory controllers 131a-131c have an adaptable error control scheme for accesses to the same memory device 132a-132c allows applications, operating systems, and/or hypervisors executing on one or more compute nodes 110a-110c to select the error control scheme that best fits the needs of the application, operating system, and/or hypervisor. The error control scheme selected by applications, operating systems, and/or hypervisors may be based on a policy designed to improve and/or trade-off between bandwidth, latency, error recovery, and error detection. In the case of error detection, error recovery may be further provided by

system level solutions including recovering an error free copy of data from another location in the system including storage if it is available.

[0016] In an embodiment, the selection of the error control scheme used by memory controllers 131a-131c when accessing respective memory devices 132a-132c may be made based on the memory node 130a-130c interface receiving the memory transaction request. For example, if each compute node 110a-110c is associated, by memory node 130a-130c (and memory controllers 131a-131c, in particular) with a particular memory node 130a-130c interface to fabric 120, the identification of the fabric interface receiving the request may be used to determine the error control scheme used by memory controllers 131a-131c when accessing respective memory devices 132a-132c. Memory nodes 130a-130c may be configured by one or more compute nodes 110a-110c with an association between fabric interfaces and error control schemes.

[0017] In an embodiment, the selection of the error control scheme used by memory controllers 131a-131c when accessing respective memory devices 132a-132c may be made based on information in the memory transaction request received from a compute node 110a-110c. For example, a compute node 110a-110c sending a memory transaction request (e.g., read or write) to a memory node 130a-130c may include an indicator (e.g., bit or bits) of the error control scheme a memory controller 131a-131c is to use when storing/retrieving data from memory devices 132a-132c. In another example, a compute node 110a-110c sending a memory transaction request (e.g., read or write) to a memory node 130a-130c may include a processor core identification that is associated by memory controller 131a-131c with an error control scheme to be used when that processor core is storing/retrieving data from memory devices 132a-132c. In another example, a compute node 110a-110c sending a memory transaction request (e.g., read or write) to a memory node 130a-130c may include a virtual machine identification that is associated by memory controller 131a-131c with an error control scheme to be used when that virtual machine is storing/retrieving data from memory devices 132a-132c.

[0018] In an embodiment, the selection of the error control scheme may be made based on the fabric physical address of the memory transaction request and a lookup table and/or address range registers. For example, one or more of compute nodes 110a-110c may configure one or more memory controllers 131a-131c with a table that associate fabric physical address ranges to error control schemes.

[0019] In an embodiment, the selection of the error control scheme may be made based on the memory physical address and a lookup table or address range registers. For example,

one or more of compute nodes 110a-110c may configure one or more memory controllers 131a-131c with a table that associate memory physical address ranges used by memory channels 133a-133c to error control schemes.

[0020] In an embodiment, one or more of memory controllers 131a-131c, memory channels 133a-133c and memory devices 132a-132c are compatible with the double data rate 5 (DDR5) standard. Thus, memory controllers 131a-131c interface with memory devices 132a-132c via two independent memory channels 133a-133c, respectively. The two independent memory channels 133a-133c may have 40 bits or 32 bits of data width per independent channel. Example error control schemes for DDR5 memory channels 133a-133c are illustrated in Table 1.

Table 1

Error control scheme	Symbol Size (bits)	DIMM Data width	n=total bits/symbols k=data bits/symbols (n,k)	BL16 Transaction size	strength
Hamming	1	40	(39,32)	64 Byte	SECDED
BCH code GF(16)	4	40	(10,8)	64 Byte	SSC
Hamming	1	40 or 80	(72,64)	128 Byte	SECDED
BCH code GF(16)	4	40 or 80	(20,16)	128 Byte	SSCDSD

Notes:
SECDED = Single Error Correct, Double Error Detect
SSC = Single Symbol Correct
SSCDSD = Single Symbol Correct, Double Symbol Detect

[0021] In an embodiment, the extra 1 bit (40 bit DIMM width minus 39 total bits) for the Hamming (39,32) coding may be used as a spare bit (i.e., to replace a dead or faulty bit) or extra parity bit to improve on the protection of the (39,32) coding alone. In an embodiment, the extra 8 bits (2x40 or 80 bit DIMM width minus 72 total bits) for the Hamming (72,64) coding may be used as eight (8) spare bits, (e.g., to replace up to eight dead or faulty bits), two spare nibbles (e.g., to replace two dead or faulty 4-bit wide devices), or one (1) spare

byte (e.g., to replace a single dead or faulty 8-bit wide device) thereby improving on the protection of the (72,64) coding alone.

[0022] Figure 2 is a block diagram illustrating a multi-host processing system. In Figure 2, processing system 200 comprises compute nodes 210a-210c, controller 231, and memory devices 232. Controller 231 includes interfaces 225a-225c, address translation 226, multiplexor/demultiplexor (MUX/DEMUX) 227, subchannel 1 error control 235a (a.k.a., subchannel 1 ECC), subchannel 2 error control 235b (a.k.a., subchannel 2 ECC), interleaved error control 236 (a.k.a., interleaved ECC), subchannel 1 memory controller 237a, subchannel 2 memory controller 237b, and control circuitry 240. Control circuitry 240 includes scheduler 241 and optionally error control scheme map 242.

[0023] Subchannel 1 memory controller 237a is operatively coupled to memory devices 232. Subchannel 2 memory controller 237b is operatively coupled to memory devices 232. Compute nodes 210a-210c, controller 231, subchannel memory controllers 237a-237b, and memory devices 232 may be, or comprise, integrated circuit type devices, such as are commonly referred to as a "chips". Memory controllers, such as subchannel memory controllers 237a-237b, manage the flow of data going to and from memory devices and/or memory modules. Memory devices 232 may be one or more standalone devices, or may be one or more memory modules. A memory controller can be a separate, standalone chip, or integrated into another chip. For example, a memory controller may be included on a single die with a microprocessor, or included as part of a more complex integrated circuit system such as a block of a system on a chip (SOC) (e.g., controller 231.)

[0024] Compute nodes 210a-210c are operatively coupled to interfaces 225a-225c, respectively. Interfaces 225a-225c are operatively coupled to compute nodes 210a-210c to receive memory transaction requests (e.g., read and write operations/commands directed to memory devices 232) from compute nodes 210a-210c, respectively. Interfaces 225a-225c are operatively coupled to compute nodes 210a-210c to transmit memory transaction results (e.g., data retrieved from memory devices 232) to compute nodes 210a-210c, respectively.

[0025] Interfaces 225a-225c are operatively coupled to address translation 226. Address translation 226 receives memory transaction requests from compute nodes 210a-210c via interfaces 225a-225c. Address translation 226 receives memory transaction requests from compute nodes 210a-210c that include address information that is given in the form of a fabric physical address (FPA). A fabric physical address specifies an address that allows for routing transactions (memory or otherwise) among compute nodes 210a-210c, memory nodes (e.g., controller 231), and/or other devices. An example of an FPA type addressing scheme is

the CXL.mem transaction layer address. An example FPA field assignment is illustrated in Figure 3A.

[0026] Address translation 226 translates FPAs to device channel physical addresses (DPAs). Address translation 226 translates FPAs received from compute nodes 210a-210c to device DPAs that can be used to address memory devices 232. An example DPA field assignment is illustrated in Figure 3B.

[0027] Address translation 226 is operatively coupled to MUX/DEMUX 227. MUX/DEMUX 227 receives transaction information and DPAs from address translation 226 and extracts command, address, and data. MUX/DEMUX 227 receives memory operation information from subchannel 1 ECC 235a, subchannel 2 ECC, and/or interleaved ECC 236 and uses this information (e.g., data retrieved from memory devices 232) to create a return (result) transaction to be sent to a compute node 210a-210c.

[0028] MUX/DEMUX 227 is operatively coupled to subchannel 1 ECC 235a, subchannel 2 ECC 235b, and interleaved ECC 236. MUX/DEMUX 227 is operatively coupled to subchannel 1 ECC 235a, subchannel 2 ECC 235b, and interleaved ECC 236 to, based on the DPA and an indicator of the error control scheme to be applied to the transaction, route the command, address, and data information (if any) to the appropriate one of subchannel 1 ECC 235a, subchannel 2 ECC 235b, and interleaved ECC 236.

[0029] In an embodiment, the indicator of the error control scheme to be applied to the transaction when accessing memory devices 232 may be made based on the interface 225a-225c that received the memory transaction request. For example, if each compute node 210a-210c is associated with a particular interface 225a-225c, the identification of the interface 225a-225c receiving the request may be used to determine the error control scheme to be used when accessing memory devices 232. Based on the error control scheme to be used, and the DPA which determine which subchannel memory controller 237a-237b is to be used for non-interleaved transactions, MUX/DEMUX 227 routes the transaction information to the appropriate one of subchannel 1 ECC 235a (for non-interleaved error control scheme transactions directed to subchannel 1 by the DPA), subchannel 2 ECC 235b (for non-interleaved error control scheme transactions directed to subchannel 2 by the DPA), and interleaved ECC 236 (for interleaved error control scheme transactions). Control circuitry 240 may be configured by one or more compute nodes 210a-210c with associations between fabric interfaces 225a-225c and error control schemes. Such associations may be configured by, for example, storing the associations using an error control scheme map 242 or table.

[0030] In an embodiment, the indicator of the error control scheme to be applied to the transaction when accessing memory devices 232 may be made based on information in the memory transaction request received from a compute node 210a-210c. For example, a compute node 210a-210c sending a memory transaction request (e.g., read or write) to controller 231 may include an indicator (e.g., bit or bits) of the error control scheme controller 231 is to use when storing/retrieving data from memory devices 232. In another example, a compute node 210a-210c sending a memory transaction request (e.g., read or write) to controller 231 may include a processor core identification that is associated by controller 231 with an error control scheme to be used when that processor core is storing/retrieving data from memory devices 232. Control circuitry 240 may be configured by one or more compute nodes 210a-210c with associations between processor core identifications and error control schemes. Such associations may be configured by, for example, storing the associations using an error control scheme map 242 or table.

[0031] In another example, a compute node 210a-210c sending a memory transaction request (e.g., read or write) to controller 231 may include a virtual machine identification that is associated by controller 231 with an error control scheme to be used when that virtual machine is storing/retrieving data from memory devices 232. Control circuitry 240 may be configured by one or more compute nodes 210a-210c with associations between virtual machine identifications and error control schemes. Such associations may be configured by, for example, storing the associations using an error control scheme map 242 or table.

[0032] In an embodiment, the indicator of the error control scheme to be applied to the transaction when accessing memory devices 232 may be made based on the fabric physical address of the memory transaction request and a lookup table and/or address range registers. For example, one or more of compute nodes 210a-210c may configure controllers 231 with an error control scheme map 242 that associates fabric physical address ranges to error control schemes. Map 242 (a.k.a. a lookup table) can be modified from time to time to change the error control scheme for a given address region(s). For example, when a new application, process, and/or thread is allocated memory, the error control scheme may be updated for all or part of the memory assigned to that application, process, and/or thread. In an embodiment, map 242 may be modified/configured based on the type of data to be stored in the memory allocation. For example, map 242 may be configured such that memory allocated to hold executable code uses a very robust error control scheme while memory allocated to hold a JPEG image may, because JPEG data already includes an error control scheme, use a less robust error control scheme.

[0033] In an embodiment, the indicator of the error control scheme to be applied to the transaction when accessing memory devices 232 may be made based on the memory physical address and a lookup table or address range registers. For example, one or more of compute nodes 210a-210c may configure controller 231 with an error control scheme map 242 that associates memory physical address ranges used by subchannel memory controllers 237a-237b to identify error control schemes.

[0034] Subchannel 1 ECC 235a is operatively coupled to MUX/DEMUX 227 and subchannel 1 memory controller 237a. Subchannel 1 ECC 235a is operatively coupled to MUX/DEMUX 227 and subchannel 1 memory controller 237a to produce, check, and communicate error control information for transactions directed to subchannel 1 memory controller 237a that use non-interleaved error control schemes. Subchannel 2 ECC 235b is operatively coupled to MUX/DEMUX 227 and subchannel 1 memory controller 237b. Subchannel 2 ECC 235b is operatively coupled to MUX/DEMUX 227 and subchannel 2 memory controller 237b to produce, check, and communicate error control information for transactions directed to subchannel 2 memory controller 237b that use non-interleaved error control schemes.

[0035] Interleaved ECC 236 is operatively coupled to MUX/DEMUX 227, subchannel 1 memory controller 237a, and subchannel 2 memory controller 237b. Interleaved ECC 236 is operatively coupled to MUX/DEMUX 227, subchannel 1 memory controller 237a, and subchannel 2 memory controller 237b to produce, check, and communicate error control information for transactions directed to controller 231 that use interleaved error control schemes. Interleaved error control schemes use data and error control information that stored/retrieved across both subchannel 1 memory controller 237a, and subchannel 2 memory controller 237b (e.g., using both subchannels of memory devices 232 when memory devices 232 conforms to the DDR5 DIMM specifications.)

[0036] Scheduler 241 of control circuitry 240 coordinates the operations of subchannel 1 memory controller 237a and subchannel 2 memory controller 237b. Scheduler 241 of control circuitry 240 coordinates the operations of subchannel 1 memory controller 237a and subchannel 2 memory controller 237b to schedule the memory operations to/from memory devices 232. Scheduler 241 of control circuitry 240 also coordinates the operations of subchannel 1 ECC 235a, subchannel 2 ECC 235b, and interleaved ECC 236. Scheduler 241 of control circuitry 240 coordinates the operations of subchannel 1 ECC 235a, subchannel 2 ECC 235b, and interleaved ECC 236 subchannel 1 memory controller 237a, subchannel 2 memory controller 237b, subchannel 1 ECC 235a, subchannel 2 ECC 235b, and/or

interleaved ECC 236 to schedule and issue memory operations to memory devices 232 that have different error control schemes.

[0037] Scheduler 241 of control circuitry 240 coordinates the operations of subchannel 1 ECC 235a, subchannel 2 ECC 235b, and/or interleaved ECC 236, to schedule and issue memory operations to memory devices 232 that have different transaction sizes. For example, non-interleaved error control schemes may use transactions that are a first size (e.g., 64 bytes) and use only one of subchannel 1 memory controller 237a, subchannel 2 memory controller 237b. In contrast, interleaved error control schemes may use transactions that are a second size (e.g., 128 bytes) and use both subchannel 1 memory controller 237a and subchannel 2 memory controller 237b.

[0038] Figures 3A-3B are illustrations of address field assignments. Figure 3A illustrates an example fabric physical address field assignment that may be used by, for example, system 100, system 200, and/or their components. In Figure 3A, FPA 301 includes 55 bits. Bits 5-0 of FPA 301 specify which byte in a 64 byte transaction is addressed. Bits 6-51 of FPA 301 provide a transaction layer address (e.g., CXL.mem address.) Bits 54-52 of FPA 301 specify which host port is being addressed.

[0039] Figure 3B illustrates an example device physical address. Figure 3B illustrates an example device physical address compatible with a system having four (4) DDR5 DIMM channels with 2 subchannels per DIMM, where the DIMMs are 32 Gb x4 DDR5 DIMM, 16 H stack, 2R LRDIMM, 2DPC. Figure 3B illustrates an example fabric physical address field assignment that may be used by, for example, system 100, system 200, and/or their components. In Figure 3B, DPA 302 includes 44 bits. Bits 1-0 specify which byte in a subchannel is being addressed. Bits 5-2 of DPA 302 specify which four (4) bytes in a 64 byte transaction is addressed. Bits 12-6 specify a column address. Bits 17-13 specify bank address and bank group address. Bits 34-18 specify a row address. Bits 40-35 specify the slot, rank, and chip identification (CID). Bit 41 specifies the subchannel being addressed. Bits 43-42 specify the DIMM channel being addressed.

[0040] Figures 4A-4B are illustrations of example physical memory channel address field assignments. The address field assignments illustrated in Figures 4A-4B may be used on, for example, memory channels 133a-133c or by subchannel memory controllers 237a-237b and memory devices 232. Figure 4A illustrates an example subchannel physical address compatible with 64 byte transactions. Error control schemes that are compatible with 64 byte transactions include (39,32) Hamming and interleaved subchannel error control schemes. In Figure 4A, address 403 includes 41 bits. Bits 1-0 specify which byte in a subchannel is being

addressed. Bits 5-2 of DPA 302 specify which four (4) bytes in a 64 byte transaction is addressed. Bits 12-6 specify a column address. Bits 17-13 specify bank address and bank group address. Bits 34-18 specify a row address. Bits 40-35 specify the slot, rank, and chip identification (CID).

[0041] Figure 4B illustrates an example subchannel physical address compatible with 128 byte transactions. Error control schemes that are compatible with 128 byte transactions include (72,64) Hamming and (20,16) BCH-like codes over GF(16). In Figure 4B, address 404 includes 42 bits. Bits 1-0 specify which byte in a subchannel is being addressed. Bit 2 specifies which subchannel is being addressed. Bits 6-3 of DPA 403 specify which four (4) bytes in a 64 byte transaction is addressed. Bits 13-7 specify a column address. Bits 18-14 specify bank address and bank group address. Bits 35-19 specify a row address. Bits 41-36 specify the slot, rank, and chip identification (CID).

[0042] Figures 5A-5B are illustrations of fabric physical address to physical memory channel address mappings. Figures 5A-5B illustrate example fabric physical address to physical memory channel address mappings that may be used by, for example, system 100, system 200, and/or their components. In Figure 5A, a first fabric physical address range is mapped to a channel physical address range on subchannel 1. This is illustrated in Figure 5A by arrows 501-502 from a first range of fabric physical addresses to a first range of addresses on subchannel 1. Also, in Figure 5A, a second fabric physical address range is mapped to a channel physical address range on subchannel 2. This is illustrated in Figure 5A by arrows 503-504 from a second range of fabric physical addresses to a second range of addresses on subchannel 2. It should be understood that the error control schemes used to access the first range of addresses on subchannel 1 and second range of fabric physical addresses may be the same, or may be different, as describe herein. In an embodiment, the mapping illustrated in Figure 5A may be used to access, for example, data associated with non-interleaved (e.g., 64 byte transaction) error control schemes.

[0043] In Figure 5B, a fabric physical address range is interleaved between subchannel 1 and subchannel 2. This is illustrated in Figure 5B by the arrows running from addresses Q and Q+2 in the fabric physical address space to locations X and X+1, respectively in the subchannel 1 address space, and by the arrows running from addresses Q+1 and Q+3 in the fabric physical address space to locations Y and Y+1, respectively in the subchannel 1 address space. In an embodiment, Y=X. In another embodiment, $Y \neq X$. In an embodiment, the mapping illustrated in Figure 5B may be used to access, for example, data associated with interleaved (e.g., 128 byte transaction) error control schemes.

[0044] Figure 6 is a flowchart illustrating a method of performing memory operations using a plurality of memory error control schemes. One or more steps illustrated in Figure 6 may be performed by, for example, system 100, system 200, and/or their components. A first association between a first address range and a first error control scheme is received (602). For example, compute node 210a may configure (e.g., write) one or more entries in error control scheme map 242 to specify a first address range is to use a first error control scheme.

[0045] A second association between a second address range and a second error control scheme is received (604). For example, compute node 210b may configure (e.g., write) one or more entries in error control scheme map 242 to specify a second address range is to use a second error control scheme. In an embodiment, the first address range and the second address range are non-overlapping if the first error control scheme and the second error control scheme are different error control schemes.

[0046] A first memory transaction request is received that is directed to the first address range (606). For example, controller 231 may receive, from compute node 210c, a first memory transaction request (e.g., read or write) that is directed to a first address that is within the first address range. A first physical memory operation is scheduled that is associated with the first memory transaction request and includes information associated with the first error control scheme (610). For example, scheduler 241 may schedule a first memory operation (e.g., read or write) to be performed by subchannel 2 memory controller 237b that communicates (reads or writes) a 64 byte block of data with memory devices 232. This 64 byte block of data is to include error control information generated using the first error control scheme. For example, the 64 byte block of data to be read from, or written to, memory devices 232 via subchannel 2 memory controller 237b may include payload data and (39,32) Hamming code information.

[0047] A second memory transaction request is received that is directed to the second address range (612). For example, controller 231 may receive, from compute node 210c, a second memory transaction request (e.g., read or write) that is directed to a second address that is within the second address range. A second physical memory operation is scheduled that is associated with the second memory transaction request and includes information associated with the second error control scheme (614). For example, scheduler 241 may schedule a second memory operation (e.g., read or write) to be performed by subchannel 1 memory controller 237a that communicates (reads or writes) a 64 byte block of data with memory devices 232. This 64 byte block of data is to include error control information generated using the second error control scheme. For example, the 64 byte block of data to

be read from, or written to, memory devices 232 via subchannel 1 memory controller 237a may include payload data and (20,16) BCH code information.

[0048] The first and second physical memory operations are performed according to the schedule (616). For example, scheduler 241 may elect to perform the first memory operation and the second memory operation concurrently. Thus, for example, scheduler 241 issues the first memory operation at the same time as the second memory operation. In another example, scheduler 241 issues the second memory operation while the first memory operation is being performed.

[0049] Figure 7 is a flowchart illustrating a method of operating a memory controller. One or more steps illustrated in Figure 7 may be performed by, for example, system 100, system 200, and/or their components. A first memory transaction request is received (702). For example, controller 231 may receive, via interface 225a, a memory transaction (e.g., read or write) request from compute node 210a.

[0050] Based on a first indicator, first data associated with the first memory transaction request is communicated in association with first error control information generated using a first error control scheme (704). For example, based on an indicator of an error control scheme to be used when performing the first memory transaction request, subchannel 1 memory controller 237a, subchannel 2 memory controller 237b, or both may communicate data (e.g., read or write) with memory devices 232 in association with error control information generated in accordance with the indicator of the error control scheme. The indicator of the error control scheme may be based on, for example, one or more of: (1) the interface 225a receiving the first memory transaction request; (2) an indicator and/or other information (e.g., core ID or VM ID) included in the first memory transaction request; (3) the fabric physical address of the first memory transaction request and an FPA to error control scheme mapping; and/or (4) the device physical address of the first memory transaction request and a DPA to error control scheme mapping.

[0051] A second memory transaction request is received (706). For example, controller 231 may receive, via interface 225b, a memory transaction (e.g., read or write) request from compute node 210b. Based on a second indicator, second data associated with the second memory transaction request is communicated in association with second error control information generated using a second error control scheme (708). For example, based on an indicator of an error control scheme to be used when performing the second memory transaction request, subchannel 1 memory controller 237a, subchannel 2 memory controller 237b, or both, may communicate data (e.g., read or write) with memory devices 232 in

association with error control information generated in accordance with the indicator of the error control scheme. The indicator of the error control scheme may be based on, for example, one or more of: (1) the interface 225a receiving the second memory transaction request; (2) an indicator and/or other information (e.g., core ID or VM ID) included in the second memory transaction request; (3) the fabric physical address of the second memory transaction request and an FPA to error control scheme mapping; and/or (4) the device physical address of the second memory transaction request and a DPA to error control scheme mapping.

[0052] Figure 8 is a flowchart illustrating a method of scheduling memory operations. One or more steps illustrated in Figure 8 may be performed by, for example, system 100, system 200, and/or their components. A first memory transaction request is received (802). For example, controller 231 may receive, via interface 225a, a memory transaction (e.g., read or write) request from compute node 210a.

[0053] The first memory transaction request is associated with a first error control scheme (804). For example, control circuitry 240 may associate the first memory transaction request with one of the error control schemes detailed in Table 1. Control circuitry 240 may associate the first memory transaction request with an error control schemes based on, for example, one or more of: (1) the interface 225a receiving the first memory transaction request; (2) an indicator and/or other information (e.g., core ID or VM ID) included in the first memory transaction request; (3) the fabric physical address of the first memory transaction request and an FPA to error control scheme map 242; and/or (4) the device physical address of the first memory transaction request and a DPA to error control scheme map 242.

[0054] A second memory transaction request is received (806). For example, controller 231 may receive, via interface 225b, a memory transaction (e.g., read or write) request from compute node 210b. The second memory transaction request is associated with a second error control scheme (808). For example, control circuitry 240 may associate the second memory transaction request with one of the error control schemes detailed in Table 1. Control circuitry 240 may associate the second memory transaction request with an error control schemes based on, for example, one or more of: (1) the interface 225b receiving the second memory transaction request; (2) an indicator and/or other information (e.g., core ID or VM ID) included in the second memory transaction request; (3) the fabric physical address of the second memory transaction request and an FPA to error control scheme map 242; and/or

(4) the device physical address of the second memory transaction request and a DPA to error control scheme map 242.

[0055] A first physical memory operation is scheduled that is associated with the first memory transaction request and includes information associated with the first error control scheme (810). For example, scheduler 241 may schedule a first memory operation (e.g., read or write) to be performed by subchannel 1 memory controller 237a that communicates (reads or writes) a 64 byte block of data with memory devices 232. This 64 byte block of data is to include error control information generated using the first error control scheme. For example, the 64 byte block of data to be read from, or written to, memory devices 232 via subchannel 1 memory controller 237a may include payload data and (39,32) Hamming code information.

[0056] A second physical memory operation is scheduled that is associated with the second memory transaction request and includes information associated with the second error control scheme (812). For example, scheduler 241 may schedule a second memory operation (e.g., read or write) to be performed by subchannel 1 memory controller 237a and subchannel 2 memory controller 237b that communicates (reads or writes) a 128 byte block of data with memory devices 232. This 128 byte block of data is to include error control information generated using the second error control scheme. For example, the 128 byte block of data to be read from, or written to, memory devices 232 via subchannel 1 memory controller 237a and subchannel 2 memory controller 237b may include payload data and (20,16) BCH code information.

[0057] The first and second physical memory operations are performed according to the schedule (814). For example, scheduler 241 may elect to perform the second memory operation first, and the first memory operation second. Thus, because the second memory operation uses both subchannel 1 memory controller 237a and subchannel 2 memory controller 237b, scheduler 241 issues the second memory operation first and the first memory operation second.

[0058] The methods, systems and devices described above may be implemented in computer systems, or stored by computer systems. The methods described above may also be stored on a non-transitory computer readable medium. Devices, circuits, and systems described herein may be implemented using computer-aided design tools available in the art, and embodied by computer-readable files containing software descriptions of such circuits. This includes, but is not limited to one or more elements of system 100, system 200, and/or their components. These software descriptions may be: behavioral, register transfer, logic

component, transistor, and layout geometry-level descriptions. Moreover, the software descriptions may be stored on storage media or communicated by carrier waves.

[0059] Data formats in which such descriptions may be implemented include, but are not limited to: formats supporting behavioral languages like C, formats supporting register transfer level (RTL) languages like Verilog and VHDL, formats supporting geometry description languages (such as GDSII, GDSIII, GDSIV, CIF, and MEBES), and other suitable formats and languages. Moreover, data transfers of such files on machine-readable media may be done electronically over the diverse media on the Internet or, for example, via email. Note that physical files may be implemented on machine-readable media such as: 4 mm magnetic tape, 8 mm magnetic tape, 3-1/2 inch floppy media, CDs, DVDs, and so on.

[0060] Figure 9 is a block diagram illustrating one embodiment of a processing system 900 for including, processing, or generating, a representation of a circuit component 920. Processing system 900 includes one or more processors 902, a memory 904, and one or more communications devices 906. Processors 902, memory 904, and communications devices 906 communicate using any suitable type, number, and/or configuration of wired and/or wireless connections 908.

[0061] Processors 902 execute instructions of one or more processes 912 stored in a memory 904 to process and/or generate circuit component 920 responsive to user inputs 914 and parameters 916. Processes 912 may be any suitable electronic design automation (EDA) tool or portion thereof used to design, simulate, analyze, and/or verify electronic circuitry and/or generate photomasks for electronic circuitry. Representation 920 includes data that describes all or portions of system 100, system 200, and/or their components, as shown in the Figures.

[0062] Representation 920 may include one or more of behavioral, register transfer, logic component, transistor, and layout geometry-level descriptions. Moreover, representation 920 may be stored on storage media or communicated by carrier waves.

[0063] Data formats in which representation 920 may be implemented include, but are not limited to: formats supporting behavioral languages like C, formats supporting register transfer level (RTL) languages like Verilog and VHDL, formats supporting geometry description languages (such as GDSII, GDSIII, GDSIV, CIF, and MEBES), and other suitable formats and languages. Moreover, data transfers of such files on machine-readable media may be done electronically over the diverse media on the Internet or, for example, via email

[0064] User inputs 914 may comprise input parameters from a keyboard, mouse, voice recognition interface, microphone and speakers, graphical display, touch screen, or other type of user interface device. This user interface may be distributed among multiple interface devices. Parameters 916 may include specifications and/or characteristics that are input to help define representation 920. For example, parameters 916 may include information that defines device types (e.g., NFET, PFET, etc.), topology (e.g., block diagrams, circuit descriptions, schematics, etc.), and/or device descriptions (e.g., device properties, device dimensions, power supply voltages, simulation temperatures, simulation models, etc.).

[0065] Memory 904 includes any suitable type, number, and/or configuration of non-transitory computer-readable storage media that stores processes 912, user inputs 914, parameters 916, and circuit component 920.

[0066] Communications devices 906 include any suitable type, number, and/or configuration of wired and/or wireless devices that transmit information from processing system 900 to another processing or storage system (not shown) and/or receive information from another processing or storage system (not shown). For example, communications devices 906 may transmit circuit component 920 to another system. Communications devices 906 may receive processes 912, user inputs 914, parameters 916, and/or circuit component 920 and cause processes 912, user inputs 914, parameters 916, and/or circuit component 920 to be stored in memory 904.

[0067] Implementations discussed herein include, but are not limited to, the following examples:

[0068] Example 1: A memory buffer device, comprising: a first host interface to receive memory transaction requests; and, memory device interface to, based on a first indicator, communicate first data having a first memory error resiliency policy with a first memory device, and to, based on a second indicator, communicate first data having a second memory error resiliency policy with the first memory device.

[0069] Example 2: The memory buffer device of example 1, further comprising: a second host interface to receive memory transaction requests, the first indicator associated with memory transaction requests received via the first host interface, the second indicator associated with memory transaction requests received via the second host interface.

[0070] Example 3: The memory buffer device of example 1, further comprising: a storage element to associate the first indicator with a first address range, and to associate the second indicator with a second address range.

[0071] Example 4: The memory buffer device of example 1, wherein the first indicator is received via a first memory transaction request and the second indicator is received via a second memory transaction request.

[0072] Example 5: The memory buffer device of example 4, wherein a first source identification indicator in the first memory transaction request is associated with the first indicator, and a second source identification indicator in the second memory transaction request is associated with the second indicator.

[0073] Example 6: The memory buffer device of example 1, further comprising: a storage element to associate a first fabric physical address range with the first indicator and to associate a second fabric physical address range with the second indicator.

[0074] Example 7: The memory buffer device of example 1, further comprising: a storage element to associate a first memory physical address range with the first indicator and to associate a second memory physical address range with the second indicator.

[0075] Example 8: A memory node device, comprising: a first interface to receive memory commands; first circuitry to, based on a first indicator, communicate first data with a first memory channel in association with first error control information generated using a first error control scheme; and, second circuitry to, based on a second indicator, communicate second data with the first memory channel in association with second error control information generated using a second error control scheme.

[0076] Example 9: The memory node device of example 8, wherein, based on the second indicator, the second circuitry is to communicate third data with a second memory channel in association with third error control information generated using the second error control scheme.

[0077] Example 10: The memory node device of example 8, further comprising: a second interface to receive memory commands, the first indicator to be based on whether a first memory command was received via the first interface, the second indicator to be based on whether a second memory command was received via the second interface.

[0078] Example 11: The memory node device of example 8, wherein the first indicator is based on information received via a first memory command and the second indicator is based on information received via a second memory command.

[0079] Example 12: The memory node device of example 11, wherein the first indicator is associated with a first source identification indicator and the second indicator is associated with a second source identification indicator.

[0080] Example 13: The memory node device of example 8, further comprising: a storage element to associate a first fabric physical address range with the first indicator and to associate a second fabric physical address range with the second indicator.

[0081] Example 14: The memory node device of example 8, further comprising: a storage element to associate a first memory physical address range with the first indicator and to associate a second memory physical address range with the second indicator.

[0082] Example 15: A method, comprising: receiving a first memory transaction request; based on a first indicator, communicating first data associated with the first memory transaction request with a first memory channel in association with first error control information generated using a first error control scheme; receiving a second memory transaction request; and, based on a second indicator, communicating second data associated with the second memory transaction request with the first memory channel in association with second error control information generated using a second error control scheme.

[0083] Example 16: The method of example 15, wherein the first memory transaction request is received via a first host interface that is associated with the first indicator, and the second memory transaction request is received via a second host interface that is associated with the second indicator.

[0084] Example 17: The method of example 15, wherein the first memory transaction request includes first information to associate the first memory transaction request with the first error control scheme and the second memory transaction request includes second information to associate the second memory transaction request with the second error control scheme.

[0085] Example 18: The method of example 17, wherein the first information comprises a first source identifier and the second information comprises a second source identifier.

[0086] Example 19: The method of example 15, further comprising: receiving an association between a first fabric physical address and the first indicator; and, receiving an association between a second fabric physical address and the second indicator.

[0087] Example 20: The method of example 15, further comprising: receiving an association between a first memory physical address and the first indicator; and, receiving an association between a second memory physical address and the second indicator.

[0088] The foregoing description of the invention has been presented for purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form disclosed, and other modifications and variations may be possible in light of the above teachings. The embodiment was chosen and described in order to best explain the

principles of the invention and its practical application to thereby enable others skilled in the art to best utilize the invention in various embodiments and various modifications as are suited to the particular use contemplated. It is intended that the appended claims be construed to include other alternative embodiments of the invention except insofar as limited by the prior art.

CLAIMS

What is claimed is:

1. A memory buffer device, comprising:
    a first host interface to receive memory transaction requests; and,
    memory device interface to, based on a first indicator, communicate first data having a first memory error resiliency policy with a first memory device, and to, based on a second indicator, communicate first data having a second memory error resiliency policy with the first memory device.

2. The memory buffer device of claim 1, further comprising:
    a second host interface to receive memory transaction requests, the first indicator associated with memory transaction requests received via the first host interface, the second indicator associated with memory transaction requests received via the second host interface.

3. The memory buffer device of claim 1, further comprising:
    a storage element to associate the first indicator with a first address range, and to associate the second indicator with a second address range.

4. The memory buffer device of claim 1, wherein the first indicator is received via a first memory transaction request and the second indicator is received via a second memory transaction request.

5. The memory buffer device of claim 4, wherein a first source identification indicator in the first memory transaction request is associated with the first indicator, and a second source identification indicator in the second memory transaction request is associated with the second indicator.

6. The memory buffer device of claim 1, further comprising:
    a storage element to associate a first fabric physical address range with the first indicator and to associate a second fabric physical address range with the second indicator.

7. The memory buffer device of claim 1, further comprising:

a storage element to associate a first memory physical address range with the first indicator and to associate a second memory physical address range with the second indicator.

8. A memory node device, comprising:

a first interface to receive memory commands;

first circuitry to, based on a first indicator, communicate first data with a first memory channel in association with first error control information generated using a first error control scheme; and,

second circuitry to, based on a second indicator, communicate second data with the first memory channel in association with second error control information generated using a second error control scheme.

9. The memory node device of claim 8, wherein, based on the second indicator, the second circuitry is to communicate third data with a second memory channel in association with third error control information generated using the second error control scheme.

10. The memory node device of claim 8, further comprising:

a second interface to receive memory commands, the first indicator to be based on whether a first memory command was received via the first interface, the second indicator to be based on whether a second memory command was received via the second interface.

11. The memory node device of claim 8, wherein the first indicator is based on information received via a first memory command and the second indicator is based on information received via a second memory command.

12. The memory node device of claim 11, wherein the first indicator is associated with a first source identification indicator and the second indicator is associated with a second source identification indicator.

13. The memory node device of claim 8, further comprising:

a storage element to associate a first fabric physical address range with the first indicator and to associate a second fabric physical address range with the second indicator.

14. The memory node device of claim 8, further comprising:

a storage element to associate a first memory physical address range with the first indicator and to associate a second memory physical address range with the second indicator.

15. A method, comprising:

receiving a first memory transaction request;

based on a first indicator, communicating first data associated with the first memory transaction request with a first memory channel in association with first error control information generated using a first error control scheme;

receiving a second memory transaction request; and,

based on a second indicator, communicating second data associated with the second memory transaction request with the first memory channel in association with second error control information generated using a second error control scheme.

16. The method of claim 15, wherein the first memory transaction request is received via a first host interface that is associated with the first indicator, and the second memory transaction request is received via a second host interface that is associated with the second indicator.

17. The method of claim 15, wherein the first memory transaction request includes first information to associate the first memory transaction request with the first error control scheme and the second memory transaction request includes second information to associate the second memory transaction request with the second error control scheme.

18. The method of claim 17, wherein the first information comprises a first source identifier and the second information comprises a second source identifier.

19. The method of claim 15, further comprising:

receiving an association between a first fabric physical address and the first indicator; and,

receiving an association between a second fabric physical address and the second indicator.

20. The method of claim 15, further comprising:

receiving an association between a first memory physical address and the first indicator; and,

receiving an association between a second memory physical address and the second indicator.

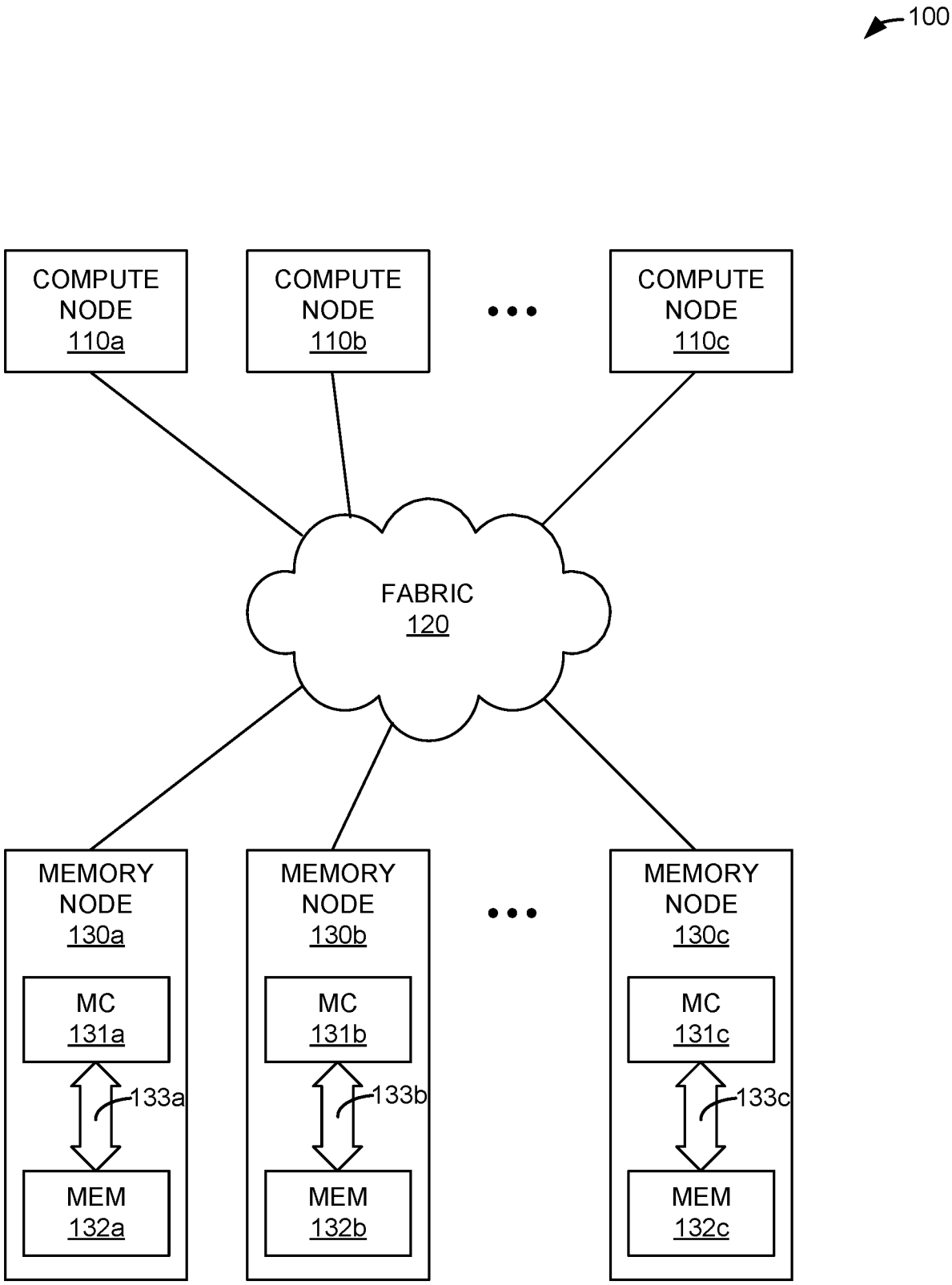
100
COMPUTE NODE 110a
COMPUTE NODE 110b
COMPUTE NODE 110c
FABRIC 120
MEMORY NODE 130a
MEMORY NODE 130b
MEMORY NODE 130c
MC 131a
MC 131b
MC 131c
133a
133b
133c
MEM 132a
MEM 132b
MEM 132c

FIG. 1

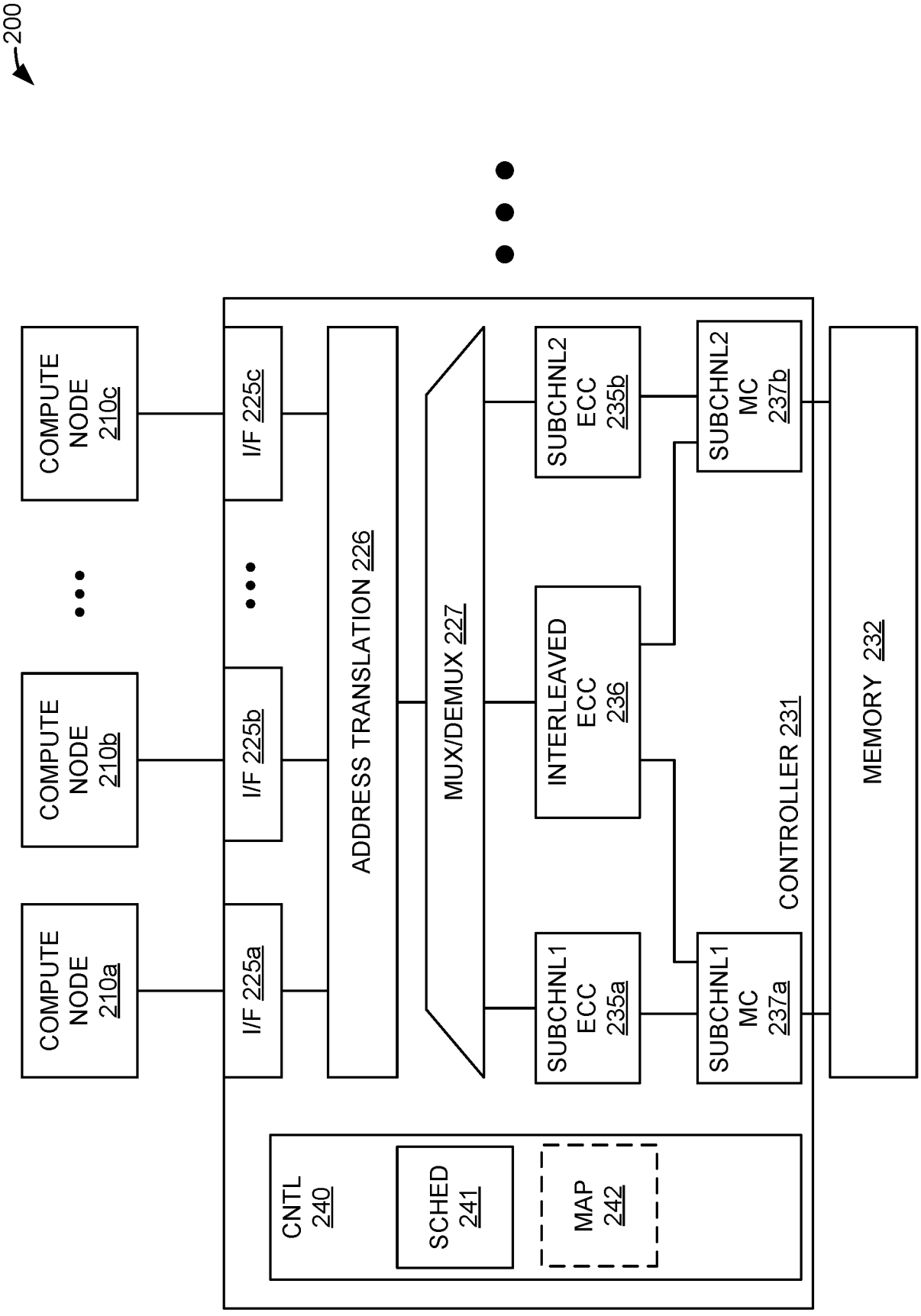
200
COMPUTE NODE 210a
COMPUTE NODE 210b
COMPUTE NODE 210c
I/F 225a
I/F 225b
I/F 225c
ADDRESS TRANSLATION 226
MUX/DEMUX 227
SUBCHNL1 ECC 235a
INTERLEAVED ECC 236
SUBCHNL2 ECC 235b
SUBCHNL1 MC 237a
SUBCHNL2 MC 237b
CONTROLLER 231
MEMORY 232
CNTL 240
SCHED 241
MAP 242

FIG. 2

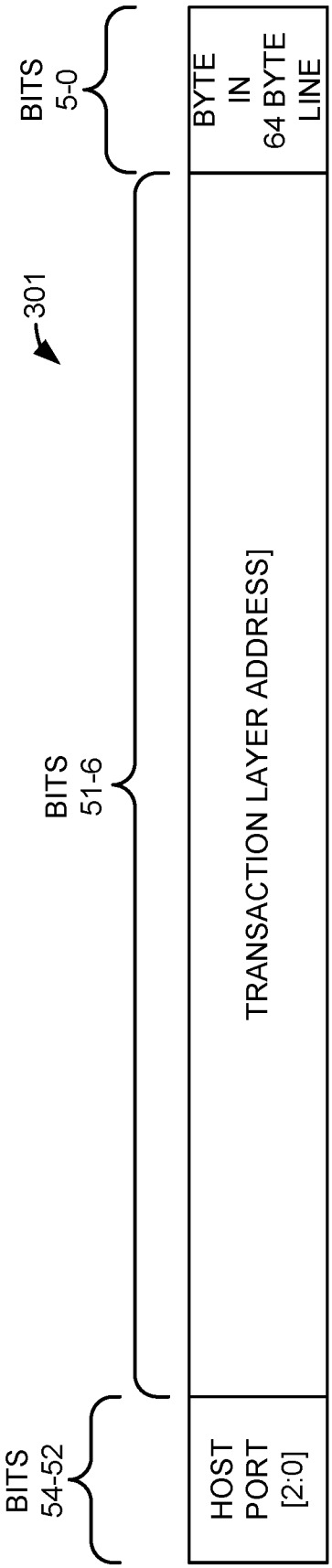
301
BITS 54-52
HOST PORT [2:0]
BITS 51-6
TRANSACTION LAYER ADDRESS]
BITS 5-0
BYTE IN 64 BYTE LINE

FIG. 3A

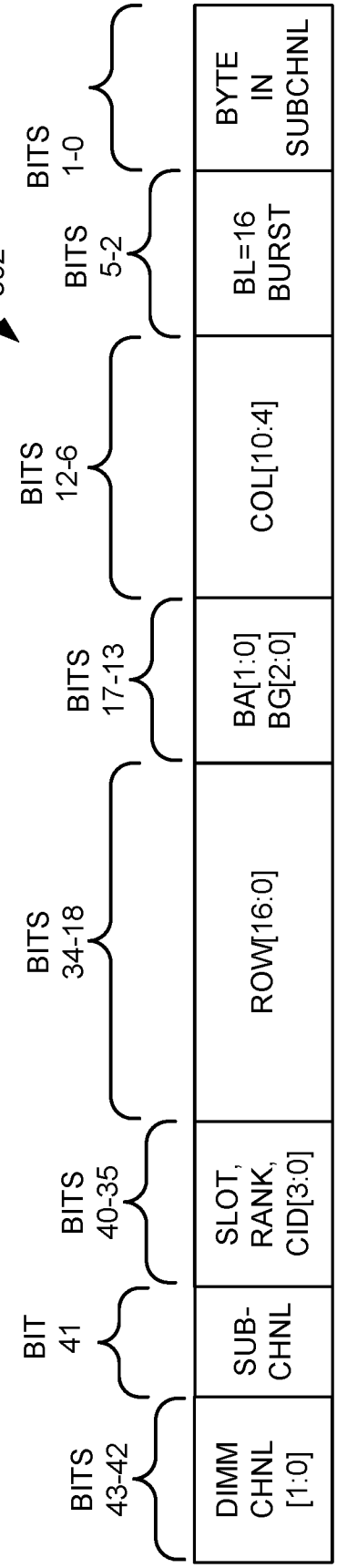
302
BITS 43-42
DIMM CHNL [1:0]
BIT 41
SUB-CHNL
BITS 40-35
SLOT, RANK, CID[3:0]
BITS 34-18
ROW[16:0]
BITS 17-13
BA[1:0] BG[2:0]
BITS 12-6
COL[10:4]
BITS 5-2
BL=16 BURST
BITS 1-0
BYTE IN SUBCHNL

FIG. 3B

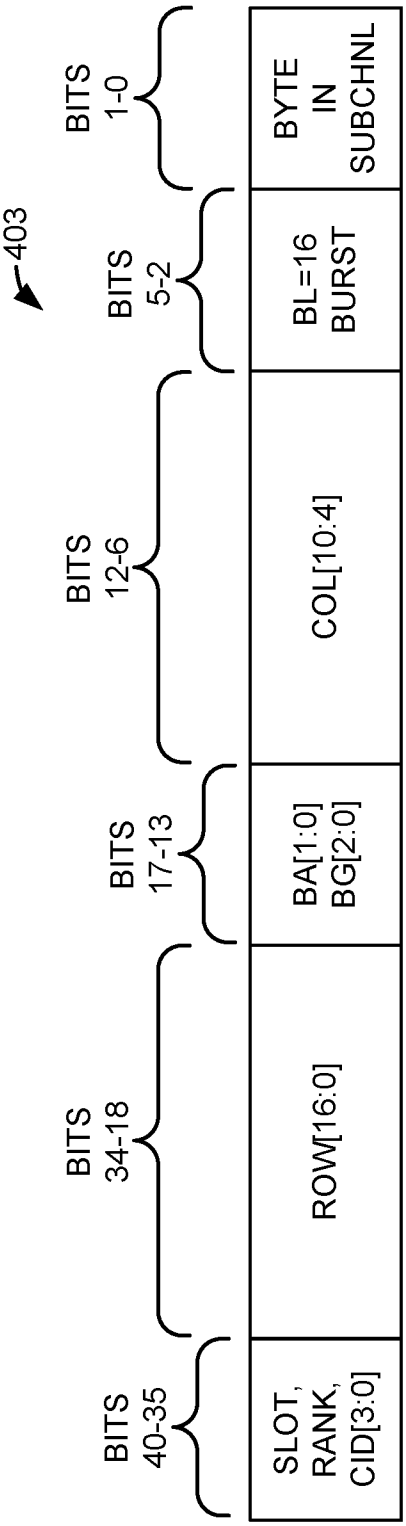
403
BITS 40-35
SLOT, RANK, CID[3:0]
BITS 34-18
ROW[16:0]
BITS 17-13
BA[1:0] BG[2:0]
BITS 12-6
COL[10:4]
BITS 5-2
BL=16 BURST
BITS 1-0
BYTE IN SUBCHNL

FIG. 4A

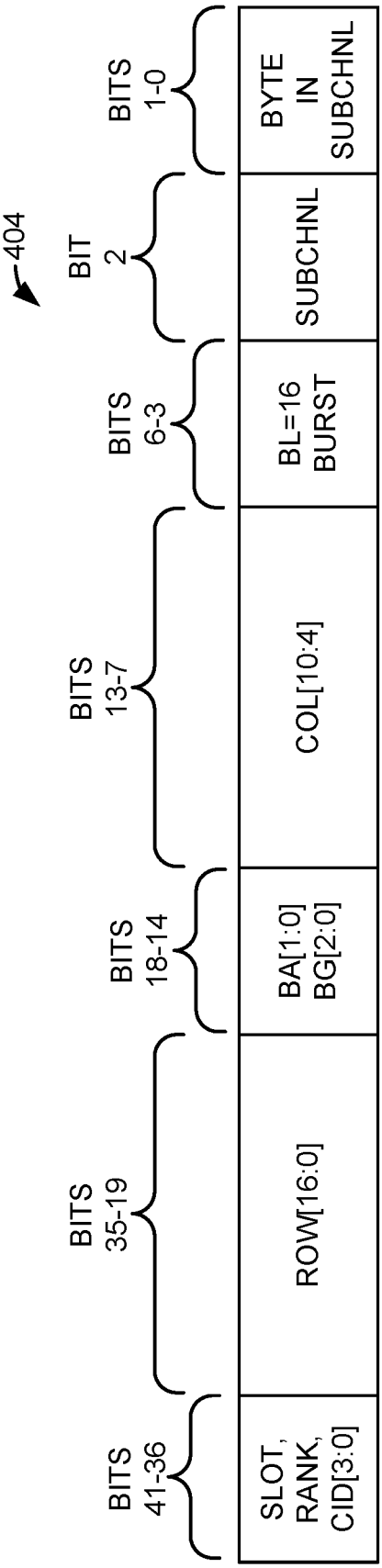
404
BITS 41-36
SLOT, RANK, CID[3:0]
BITS 35-19
ROW[16:0]
BITS 18-14
BA[1:0] BG[2:0]
BITS 13-7
COL[10:4]
BITS 6-3
BL=16 BURST
BIT 2
SUBCHNL
BITS 1-0
BYTE IN SUBCHNL

FIG. 4B

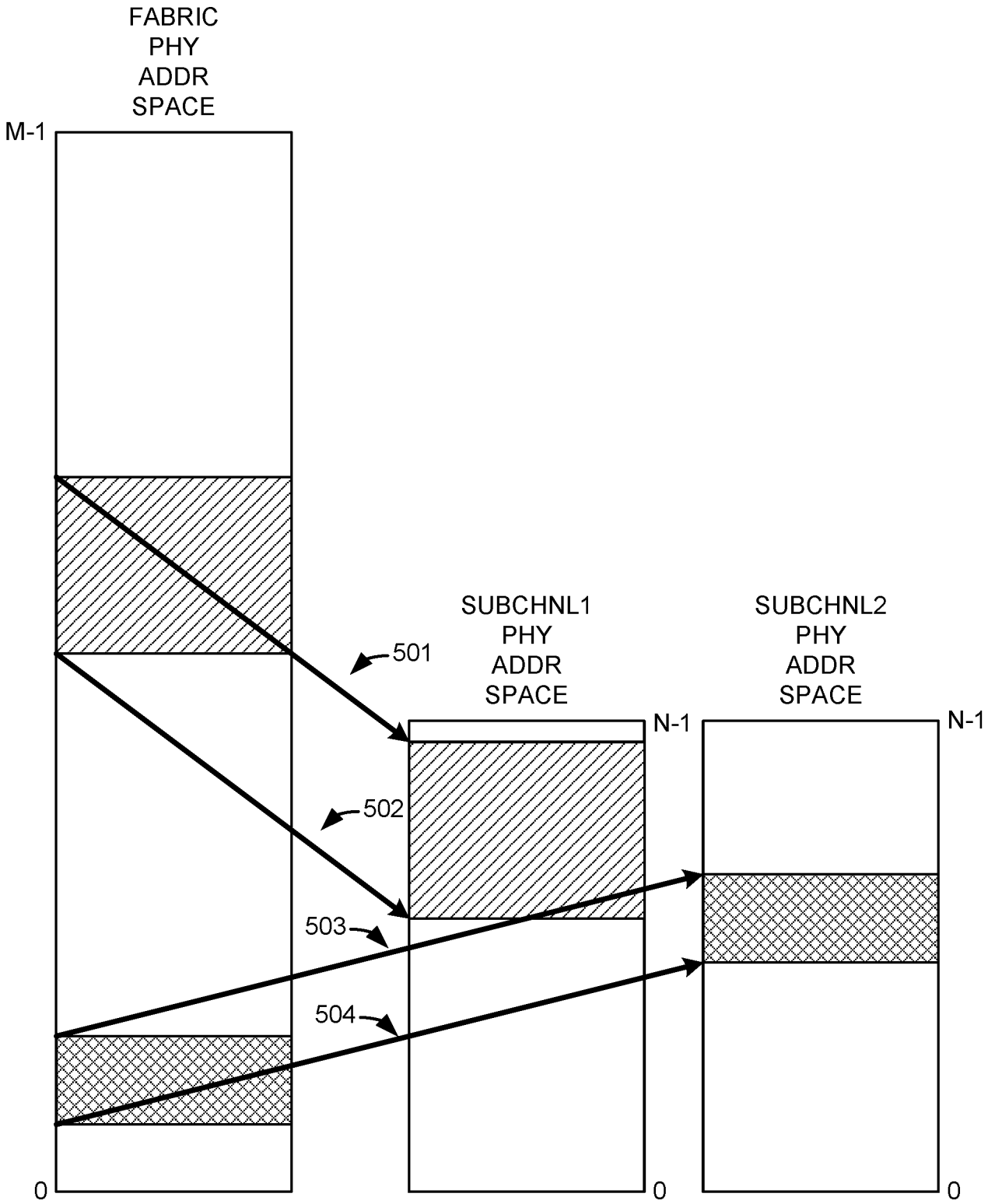
FABRIC
PHY
ADDR
SPACE
M-1
0
501
SUBCHNL1
PHY
ADDR
SPACE
N-1
0
502
503
504
SUBCHNL2
PHY
ADDR
SPACE
N-1
0

FIG. 5A

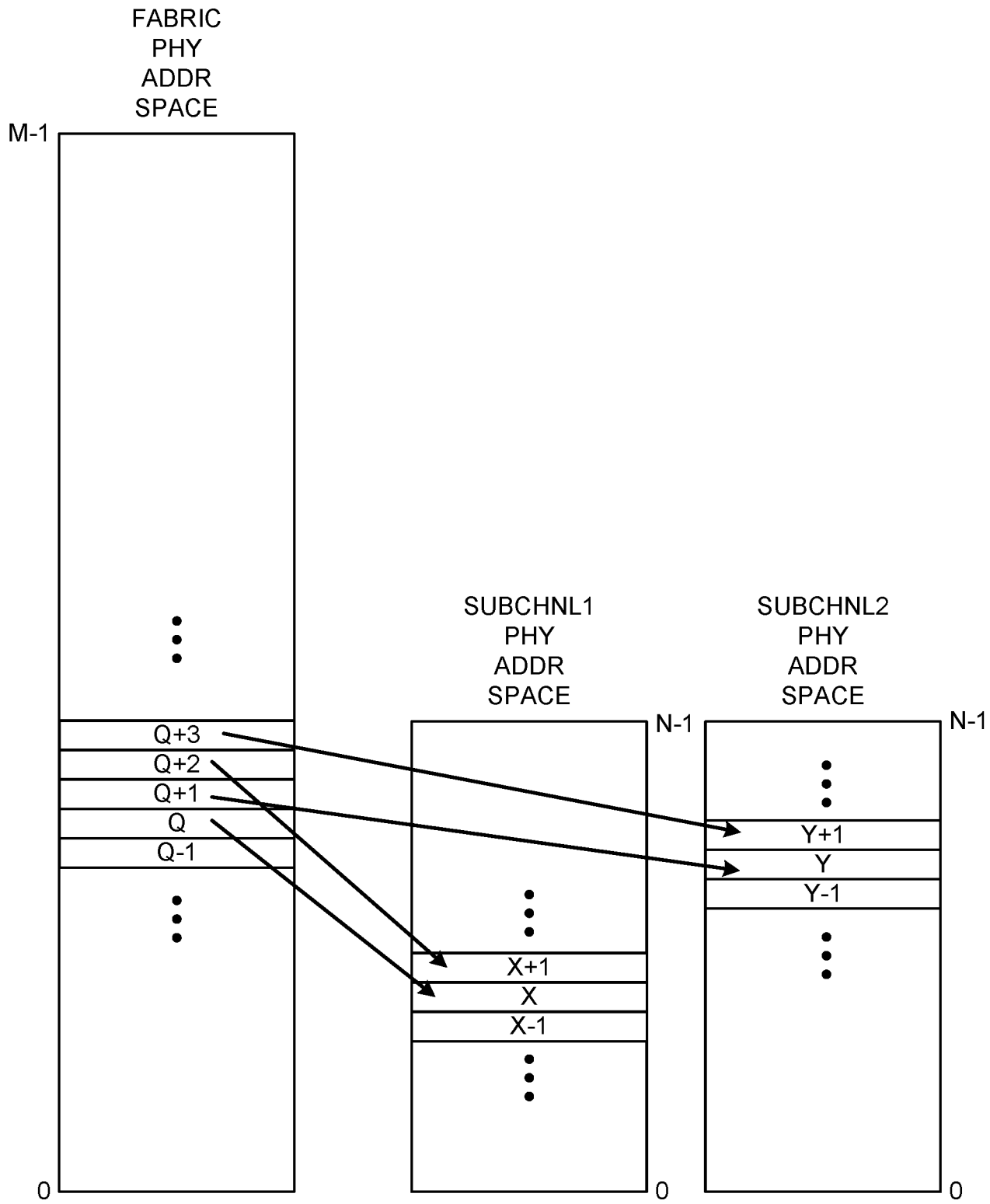
FABRIC
PHY
ADDR
SPACE
M-1
Q+3
Q+2
Q+1
Q
Q-1
0
SUBCHNL1
PHY
ADDR
SPACE
N-1
X+1
X
X-1
0
SUBCHNL2
PHY
ADDR
SPACE
N-1
Y+1
Y
Y-1
0

FIG. 5B

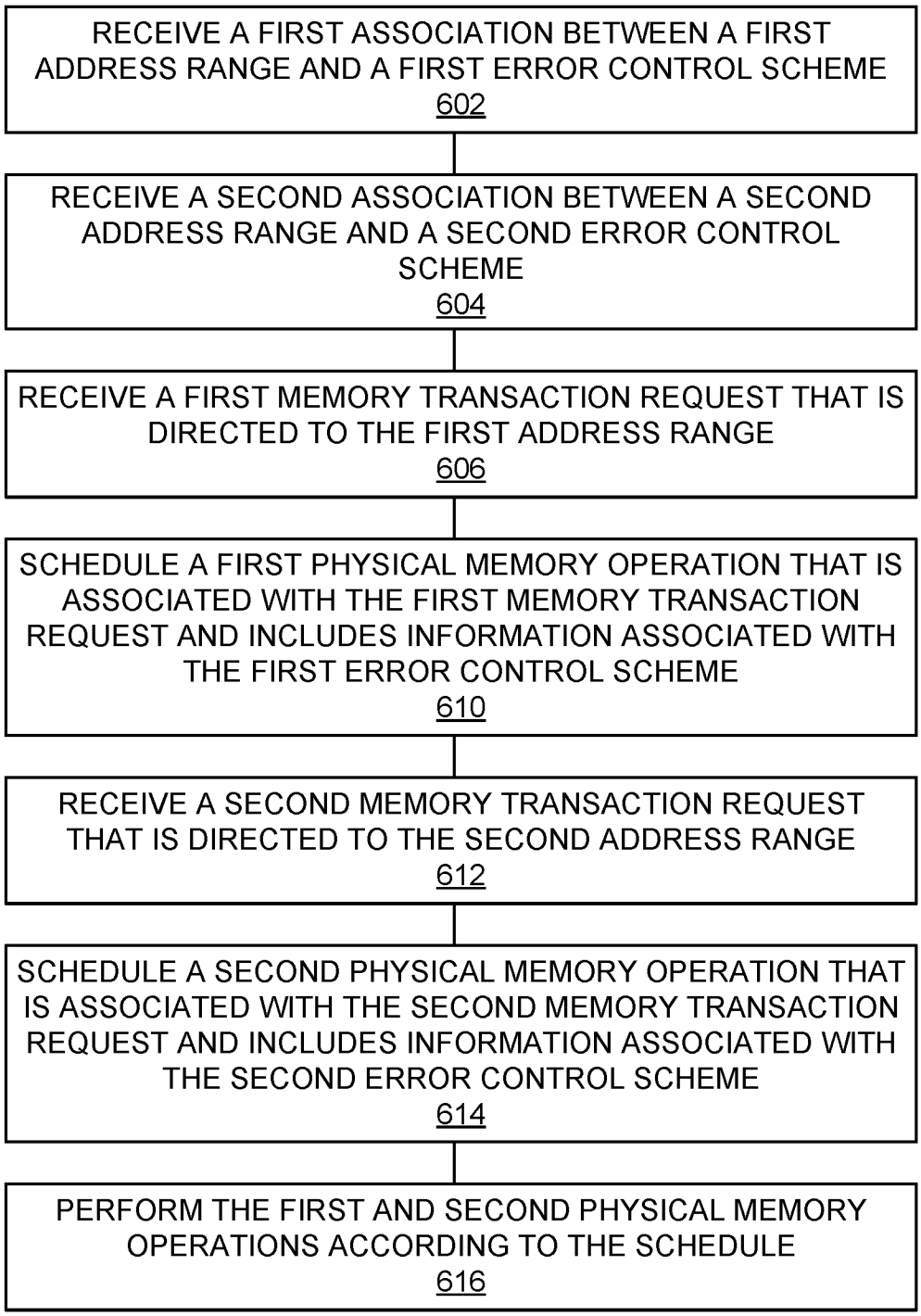
RECEIVE A FIRST ASSOCIATION BETWEEN A FIRST ADDRESS RANGE AND A FIRST ERROR CONTROL SCHEME
602
RECEIVE A SECOND ASSOCIATION BETWEEN A SECOND ADDRESS RANGE AND A SECOND ERROR CONTROL SCHEME
604
RECEIVE A FIRST MEMORY TRANSACTION REQUEST THAT IS DIRECTED TO THE FIRST ADDRESS RANGE
606
SCHEDULE A FIRST PHYSICAL MEMORY OPERATION THAT IS ASSOCIATED WITH THE FIRST MEMORY TRANSACTION REQUEST AND INCLUDES INFORMATION ASSOCIATED WITH THE FIRST ERROR CONTROL SCHEME
610
RECEIVE A SECOND MEMORY TRANSACTION REQUEST THAT IS DIRECTED TO THE SECOND ADDRESS RANGE
612
SCHEDULE A SECOND PHYSICAL MEMORY OPERATION THAT IS ASSOCIATED WITH THE SECOND MEMORY TRANSACTION REQUEST AND INCLUDES INFORMATION ASSOCIATED WITH THE SECOND ERROR CONTROL SCHEME
614
PERFORM THE FIRST AND SECOND PHYSICAL MEMORY OPERATIONS ACCORDING TO THE SCHEDULE
616

FIG. 6

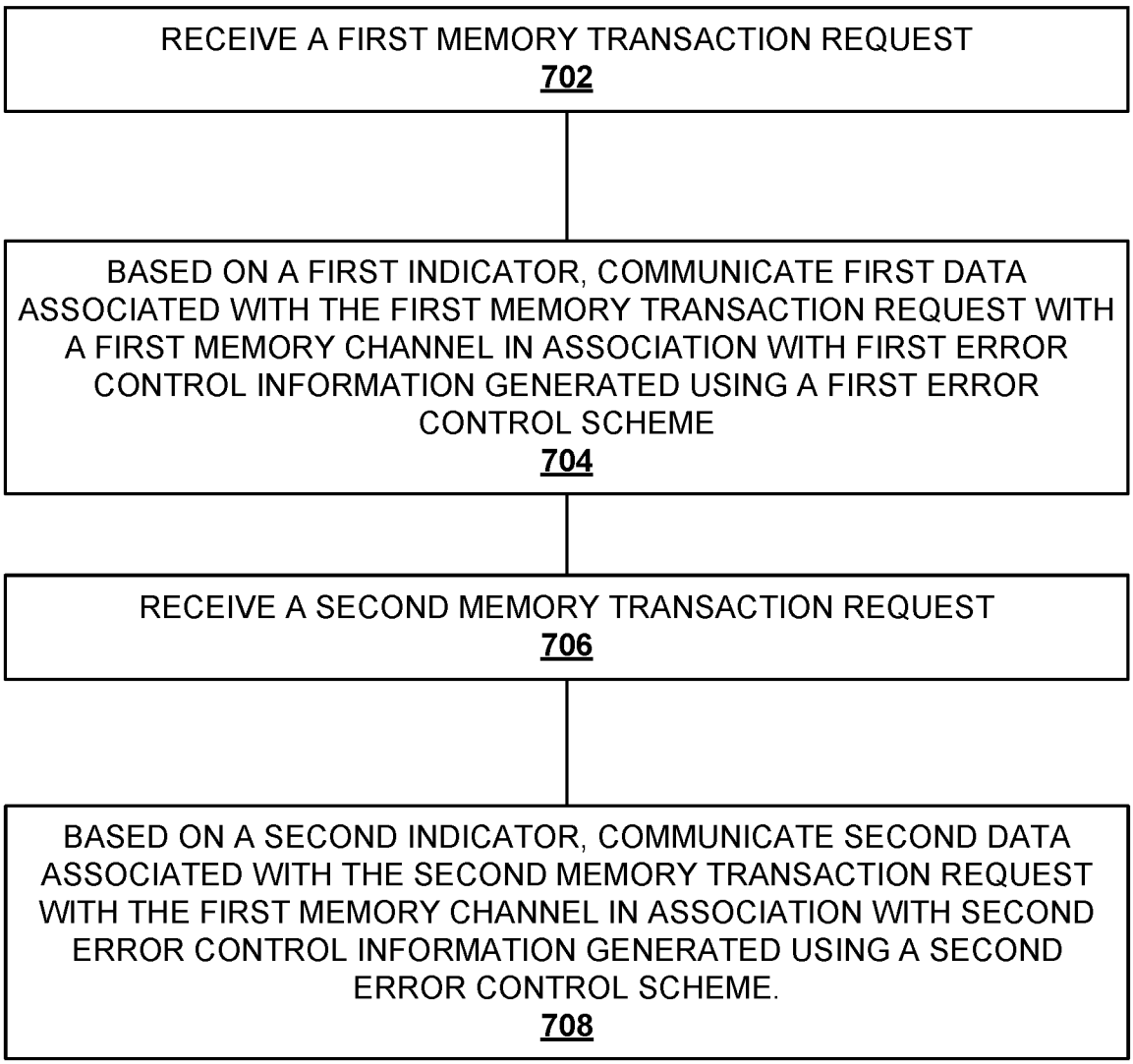
RECEIVE A FIRST MEMORY TRANSACTION REQUEST
702
BASED ON A FIRST INDICATOR, COMMUNICATE FIRST DATA ASSOCIATED WITH THE FIRST MEMORY TRANSACTION REQUEST WITH A FIRST MEMORY CHANNEL IN ASSOCIATION WITH FIRST ERROR CONTROL INFORMATION GENERATED USING A FIRST ERROR CONTROL SCHEME
704
RECEIVE A SECOND MEMORY TRANSACTION REQUEST
706
BASED ON A SECOND INDICATOR, COMMUNICATE SECOND DATA ASSOCIATED WITH THE SECOND MEMORY TRANSACTION REQUEST WITH THE FIRST MEMORY CHANNEL IN ASSOCIATION WITH SECOND ERROR CONTROL INFORMATION GENERATED USING A SECOND ERROR CONTROL SCHEME.
708

FIG. 7

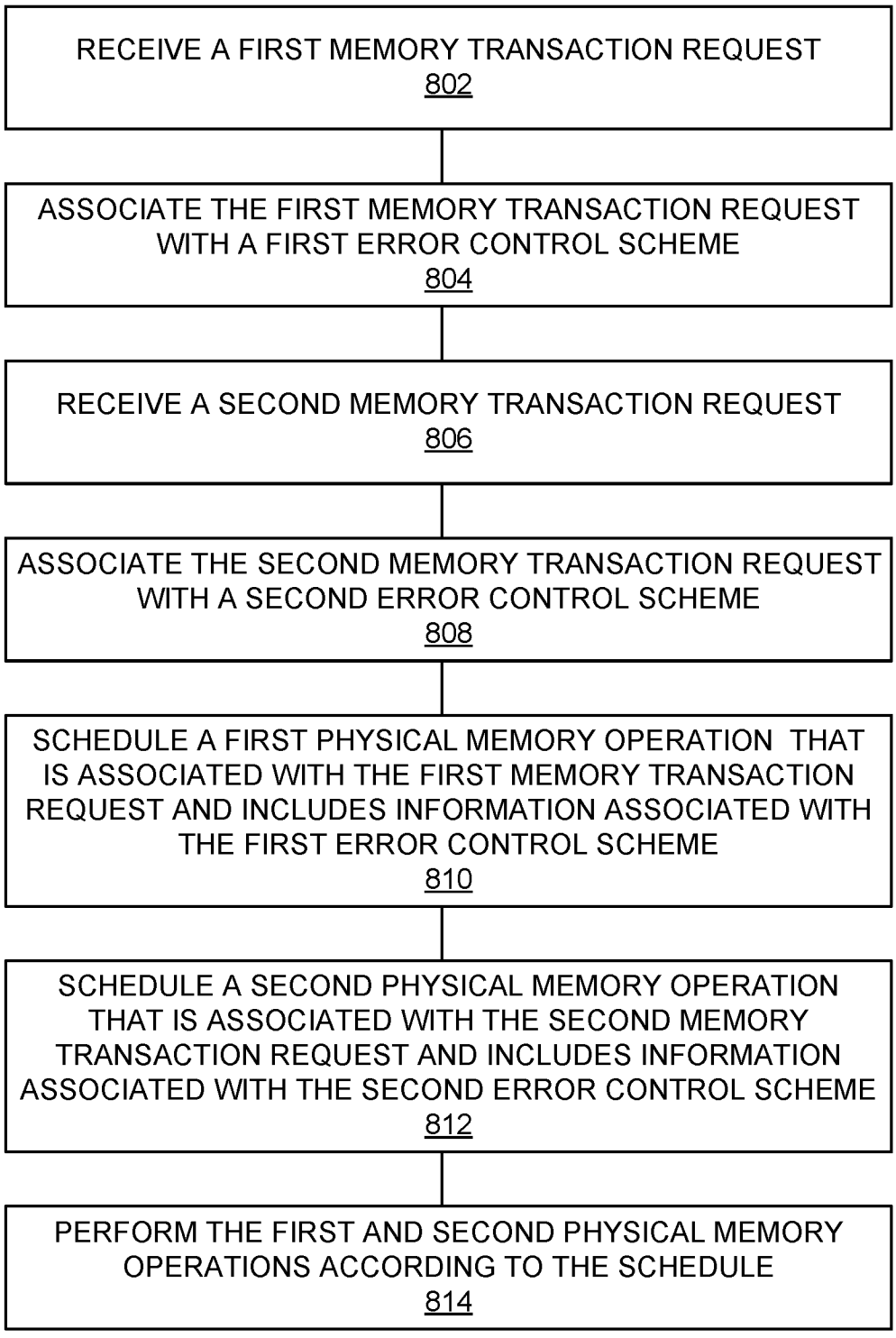
RECEIVE A FIRST MEMORY TRANSACTION REQUEST
802
ASSOCIATE THE FIRST MEMORY TRANSACTION REQUEST WITH A FIRST ERROR CONTROL SCHEME
804
RECEIVE A SECOND MEMORY TRANSACTION REQUEST
806
ASSOCIATE THE SECOND MEMORY TRANSACTION REQUEST WITH A SECOND ERROR CONTROL SCHEME
808
SCHEDULE A FIRST PHYSICAL MEMORY OPERATION THAT IS ASSOCIATED WITH THE FIRST MEMORY TRANSACTION REQUEST AND INCLUDES INFORMATION ASSOCIATED WITH THE FIRST ERROR CONTROL SCHEME
810
SCHEDULE A SECOND PHYSICAL MEMORY OPERATION THAT IS ASSOCIATED WITH THE SECOND MEMORY TRANSACTION REQUEST AND INCLUDES INFORMATION ASSOCIATED WITH THE SECOND ERROR CONTROL SCHEME
812
PERFORM THE FIRST AND SECOND PHYSICAL MEMORY OPERATIONS ACCORDING TO THE SCHEDULE
814

FIG. 8

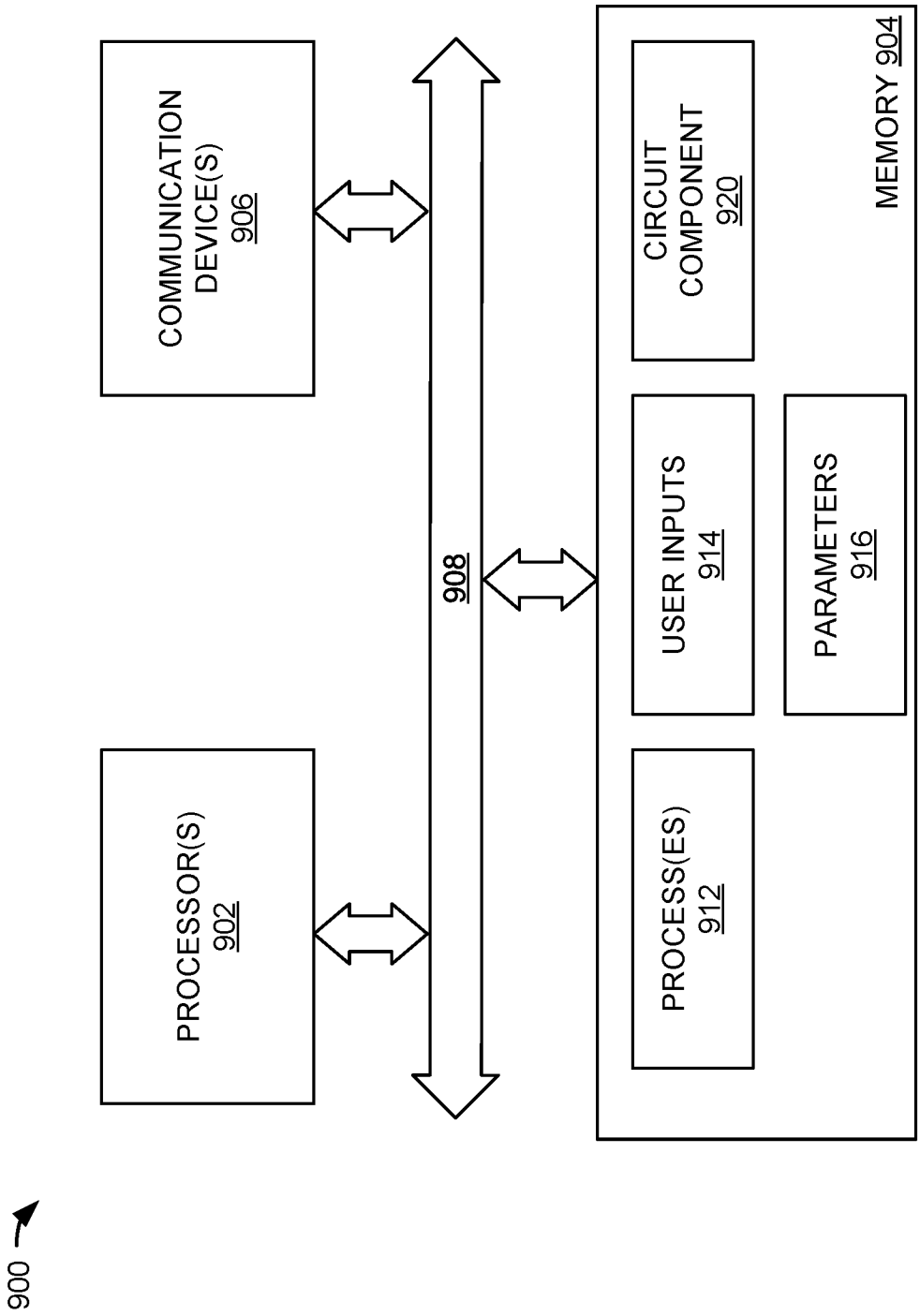
900
PROCESSOR(S)
902
COMMUNICATION
DEVICE(S)
906
908
MEMORY 904
PROCESS(ES)
912
USER INPUTS
914
PARAMETERS
916
CIRCUIT
COMPONENT
920

FIG. 9