



(12) BẢN MÔ TẢ SÁNG CHẾ THUỘC BẰNG ĐỘC QUYỀN SÁNG CHẾ

(19) Cộng hòa xã hội chủ nghĩa Việt Nam (VN)
CỤC SỞ HỮU TRÍ TUỆ

(11)



1-0042507

(51)^{2020.01} G06F 12/02

(13) B

(21) 1-2020-01479

(22) 14/08/2018

(86) PCT/US2018/046590 14/08/2018

(87) WO/2019/055164 21/03/2019

(30) 15/706,530 15/09/2017 US

(45) 27/01/2025 442

(43) 27/07/2020 388

(73) QUALCOMM INCORPORATED (US)

ATTN: International IP Administration, 5775 Morehouse Drive, San Diego, CA
92121-1714, United States of America

(72) RAMANUJAN, Raj (US); BAINS, Kuljit Singh (US); WANG, Liyong (US);
QUEEN, Wesley (US).

(74) Công ty TNHH Quốc tế D & N (D&N INTERNATIONAL CO.,LTD.)

(54) PHƯƠNG PHÁP VÀ THIẾT BỊ THỰC HIỆN CÁC THAO TÁC LIÊN TỤC

(21) 1-2020-01479

(57) Sáng chế đề cập đến phương pháp và thiết bị thực hiện các thao tác liên tục. Các hệ thống và phương pháp dùng cho các thao tác liên tục gồm máy chủ và hệ thống nhớ. Hệ thống nhớ, khi nhận lệnh Persistent Write và dữ liệu ghi liên quan từ máy chủ, thực hiện ghi liên tục dữ liệu ghi vào bộ nhớ bất biến trong hệ thống nhớ dựa trên lệnh Persistent Write. Hệ thống nhớ cũng có thể nhận mã định danh ghi (WID - write identification) liên quan đến lệnh Persistent Write từ máy chủ (host) và cung cấp, khi hoàn thành thành công Persistent Write, chỉ báo hoàn thành Persistent Write cùng với WID liên quan cho máy chủ (host).

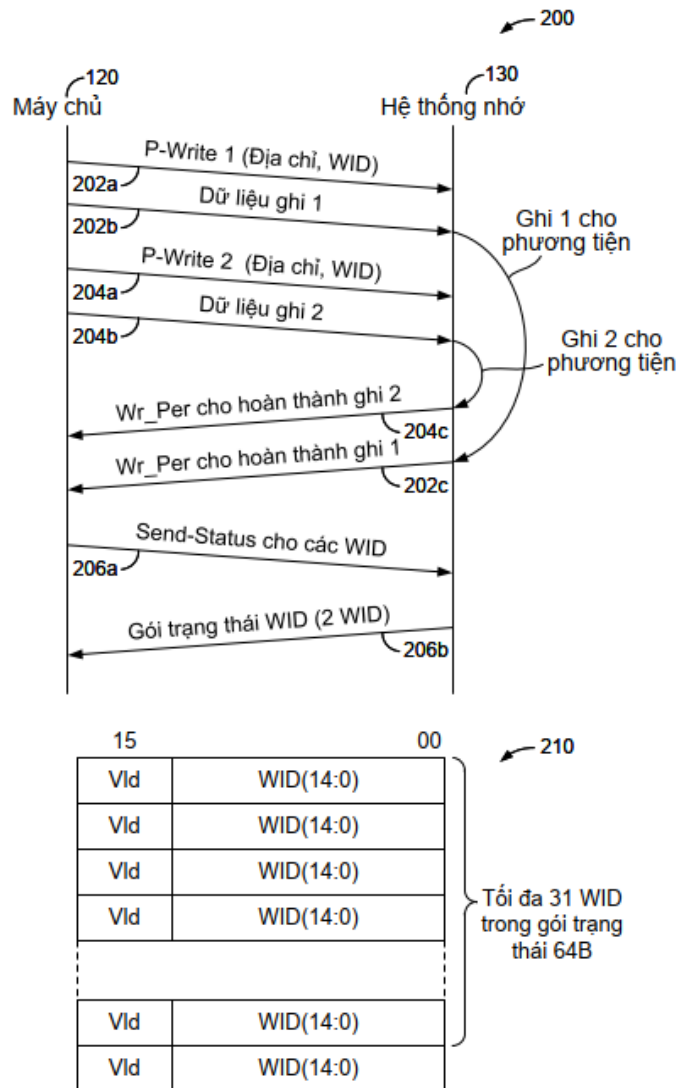


Fig.2A

Lĩnh vực kỹ thuật được đề cập

Các khía cạnh được mô tả đề cập đến hệ thống nhớ. Cụ thể hơn, các khía cạnh làm ví dụ đề cập đến các thao tác ghi liên tục (Persistent Write) và các giao thức của nó cho bộ nhớ bất biến.

Tình trạng kỹ thuật của sáng chế

Bộ nhớ lớp lưu trữ (SCM - Storage class memory) thường là bộ nhớ dung lượng cao cũng có thể có hiệu suất cao. Bộ nhớ SCM có thể được sử dụng trong các ứng dụng như các máy chủ hoặc các hệ thống xử lý khác trong đó tập hợp dữ liệu hoạt động đối với bộ xử lý hoặc thiết bị xử lý trung tâm có thể được lưu trữ trong bộ nhớ SCM, còn tập hợp dữ liệu hoàn chỉnh có thể được lưu trữ trong bộ nhớ phụ hoặc ổ cứng (HDD - hard disk drive). Bộ nhớ SCM rất được kỳ vọng ở sự liên tục ghi, điều đó có nghĩa là thông tin được ghi vào bộ nhớ SCM sẽ không bị mất nếu, giả sử, máy chủ gặp sự cố hoặc mất điện. Bộ nhớ bất biến thông thường, có thể đáp ứng các kỳ vọng như vậy liên quan đến sự liên tục, tuy nhiên, có thể không đáp ứng được các số đo về dung lượng và hiệu suất có thể mong muốn đối với bộ nhớ SCM. Do đó, các công nghệ như bộ nhớ thay đổi pha (Phase Change Memory - PCM), bộ nhớ truy cập ngẫu nhiên từ mômen truyền xoắn (Spin-Transfer Torque Magnetic Random Access Memory - STT MRAM), RAM điện trở (Resistive RAM - ReRAM), v.v., đang trở nên phổ biến hơn trong các phương án triển khai bộ nhớ SCM.

Khi sử dụng bộ nhớ SCM, một ứng dụng có thể sử dụng các thao tác ghi bộ nhớ để cập nhật bộ nhớ liên tục tương ứng. Để việc ghi vào bộ nhớ SCM được liên tục, ứng dụng yêu cầu thao tác ghi có thể kỳ vọng có sự xác nhận rõ ràng rằng thao tác ghi đã đến bộ nhớ liên tục. Ngược lại, các thao tác ghi vào bộ nhớ không liên tục (như bộ nhớ truy cập ngẫu nhiên động (DRAM - dynamic random access memory) hoặc bộ nhớ bất biến khác) thường được xem xét là đã được hoàn thành hoặc được gửi đi, từ góc độ của ứng dụng một khi thao tác ghi và dữ liệu liên quan đã được truyền đến bộ nhớ và không có sự xác nhận rõ ràng rằng dữ liệu đã được ghi là cần thiết. Vì vậy, đối với các ứng dụng sử dụng bộ nhớ SCM với sự kỳ vọng về tính liên tục, có mong muốn về các kỹ thuật hiệu suất cao cung cấp sự xác nhận rõ ràng về các thao tác ghi vào bộ nhớ liên tục, trong đó các kỹ thuật hiệu suất cao còn tương thích với các kích thước dữ liệu khác nhau để tối đa hóa hiệu suất.

Có hai loại sơ đồ thông thường đối với các thao tác bộ nhớ liên tục. Sơ đồ thứ nhất giả sử rằng toàn bộ hệ thống nhớ (ví dụ, môđun nhớ hai hàng chân (dual in-line memory module - DIMM) bao gồm một loạt mạch tích hợp DRAM, như đã biết trong lĩnh vực kỹ thuật này) được hỗ trợ năng lượng. Trong trường hợp này, thao tác ghi vào bộ đệm trung gian ở đầu nhận của DIMM có thể đủ đáp ứng sự kỳ vọng về tính liên tục. Theo một phương án thực hiện, một khi thao tác ghi qua giao diện kênh giữa ứng dụng yêu cầu thao tác ghi và DIMM được hoàn thành thành công, thao tác ghi có thể được xem là liên tục. Tuy nhiên, quá trình thực thi các sơ đồ như vậy có thể liên quan đến việc sử dụng các thiết bị lưu trữ năng lượng như các siêu tụ điện hoặc pin cung cấp năng lượng/điện tích để xóa sạch (flush) các bộ đệm trung gian trên DIMM khi phát hiện mất điện. Nhưng các thiết bị lưu trữ năng lượng như vậy có thể không có sẵn trên tất cả các DIMM, và hơn nữa, ngay cả khi có sẵn, chúng cũng có giá thành cao.

Trong sơ đồ thứ hai, tất cả các thao tác ghi trước có thể được xóa sạch cho bộ nhớ liên tục trong khi ứng dụng đợi trạng thái hoàn thành từ DIMM. Tuy nhiên, sơ đồ này có thể chịu chi phí hiệu suất đáng kể. Ví dụ, trong các trường hợp trong đó ứng dụng có thể yêu cầu các lệnh Persistent Write có độ chi tiết cao (fine granularity) vào DIMM nhưng có thể có các thao tác ghi đồng thời nhưng độc lập khác tạo dòng (streaming) vào DIMM, việc xóa sạch tất cả các thao tác ghi trước đó vào bộ nhớ liên tục đang chờ trạng thái hoàn thành có thể làm chậm không chỉ các yêu cầu Persistent Write mà cả các thao tác ghi đồng thời.

Do đó, lĩnh vực kỹ thuật này cần có các thao tác Persistent Write hiệu suất cao và hiệu quả cao hỗ trợ các độ chi tiết hoặc kích thước khác nhau của các lệnh Persistent Write, đồng thời tránh các nhược điểm nói trên của các phương pháp thông thường.

Bản chất kỹ thuật của sáng chế

Các khía cạnh làm ví dụ của sáng chế bao gồm các hệ thống và phương pháp đối với các thao tác liên tục. Hệ thống nhớ, khi nhận lệnh Persistent Write và dữ liệu ghi liên quan từ máy chủ (host), thực hiện ghi liên tục dữ liệu ghi vào bộ nhớ bất biến trong hệ thống nhớ dựa trên lệnh Persistent Write. Hệ thống nhớ cũng có thể nhận mã định danh ghi (WID - write identification) liên quan đến lệnh Persistent Write từ máy chủ (host) và cung cấp, khi hoàn thành thành công Persistent Write, chỉ báo hoàn thành Persistent Write cùng với WID liên quan cho máy chủ (host).

Theo một khía cạnh làm ví dụ, sáng chế đề cập đến phương pháp thực hiện các thao tác liên tục, phương pháp này bao gồm bước nhận, tại hệ thống nhớ, lệnh Persistent Write và dữ liệu ghi liên quan từ máy chủ (host), và thực hiện ghi liên tục dữ liệu ghi vào bộ nhớ bất biến trong hệ thống nhớ dựa trên lệnh Persistent Write.

Theo một khía cạnh làm ví dụ khác, sáng chế đề cập đến phương pháp thực hiện các thao tác liên tục, phương pháp này bao gồm bước cung cấp, từ máy chủ (host) đến hệ thống nhớ, lệnh Persistent Write và dữ liệu ghi liên quan, trong đó lệnh Persistent Write chỉ báo cho hệ thống nhớ để thực hiện ghi liên tục dữ liệu ghi vào bộ nhớ bất biến.

Theo một khía cạnh làm ví dụ khác, sáng chế đề cập đến thiết bị thực hiện các thao tác liên tục, thiết bị này bao gồm hệ thống nhớ được tạo cấu hình để nhận lệnh Persistent Write và dữ liệu ghi liên quan từ máy chủ (host), và thực hiện ghi liên tục dữ liệu ghi vào bộ nhớ bất biến trong hệ thống nhớ dựa trên lệnh Persistent Write.

Theo một khía cạnh làm ví dụ khác, sáng chế đề cập đến thiết bị bao gồm máy chủ (host) được tạo cấu hình để cung cấp lệnh Persistent Write và dữ liệu ghi liên quan cho hệ thống nhớ, trong đó lệnh Persistent Write chỉ báo cho hệ thống nhớ để thực hiện ghi liên tục dữ liệu ghi vào bộ nhớ bất biến.

Theo một khía cạnh làm ví dụ khác, sáng chế đề cập đến thiết bị bao gồm phương tiện lưu trữ dữ liệu, bao gồm phương tiện nhận lệnh Persistent Write và dữ liệu ghi liên quan từ máy chủ (host), và phương tiện thực hiện ghi liên tục dữ liệu ghi vào bộ nhớ bất biến trong phương tiện lưu trữ, dựa trên lệnh Persistent Write.

Theo một khía cạnh làm ví dụ khác, sáng chế đề cập đến thiết bị bao gồm phương tiện xử lý, bao gồm phương tiện cung cấp lệnh Persistent Write và dữ liệu ghi liên quan cho hệ thống nhớ, trong đó lệnh Persistent Write chỉ báo cho hệ thống nhớ để thực hiện ghi liên tục dữ liệu ghi vào bộ nhớ bất biến.

Theo một khía cạnh làm ví dụ khác, sáng chế đề cập đến phương tiện lưu trữ bất biến đọc được bằng máy tính bao gồm mã, mà khi được thực thi bởi bộ xử lý, khiến cho bộ xử lý thực hiện các thao tác liên tục, phương tiện lưu trữ khả biến đọc được bằng máy tính bao gồm mã để nhận, tại hệ thống nhớ, lệnh Persistent Write và dữ liệu ghi liên quan từ máy chủ (host), và mã để thực hiện ghi liên tục dữ liệu ghi vào bộ nhớ bất biến trong hệ thống nhớ dựa trên lệnh Persistent Write.

Theo một khía cạnh làm ví dụ khác, sáng chế đề cập đến phương tiện lưu trữ bất biến đọc được bằng máy tính bao gồm mã, mà khi được thực thi bởi bộ xử lý, khiến cho bộ xử lý thực hiện các thao tác liên tục, phương tiện lưu trữ khả biến đọc được bằng máy tính bao gồm mã để cung cấp, từ máy chủ (host) đến hệ thống nhớ, lệnh Persistent Write và dữ liệu ghi liên quan, trong đó lệnh Persistent Write chỉ báo cho hệ thống nhớ để thực hiện ghi liên tục dữ liệu ghi vào bộ nhớ bất biến.

Mô tả vắn tắt các hình vẽ

Các hình vẽ kèm theo được thể hiện để hỗ trợ việc mô tả theo các khía cạnh của sáng chế và được cung cấp chỉ để minh họa các khía cạnh và không nhằm mục đích giới hạn sáng chế.

Fig.1 minh họa hệ thống xử lý theo các khía cạnh của sáng chế.

Các hình vẽ từ Fig.2A đến Fig.2C minh họa các giao dịch đối với việc xử lý Persistent Write, theo một số khía cạnh của sáng chế.

Fig.3 minh họa việc mã hóa làm ví dụ cho lệnh Persistent Write theo sáng chế.

Các hình vẽ từ Fig.4A đến Fig.4B minh họa các chuỗi sự kiện liên quan đến các phương pháp làm ví dụ để thực hiện ghi liên tục, theo các khía cạnh của sáng chế.

Fig.5 mô tả thiết bị tính toán làm ví dụ trong đó một khía cạnh của sáng chế có thể được sử dụng một cách thuận lợi.

Mô tả chi tiết sáng chế

Các khía cạnh của sáng chế được mô tả trong phần mô tả sau đây và các hình vẽ liên quan đề cập đến các khía cạnh cụ thể của sáng chế. Các khía cạnh thay thế có thể được đưa ra mà không bị coi là vượt ra ngoài phạm vi của sáng chế. Ngoài ra, các thành phần đã biết của sáng chế sẽ không được mô tả chi tiết hoặc sẽ được lược bỏ để không gây khó hiểu cho các chi tiết liên quan của sáng chế.

Thuật ngữ “làm ví dụ” được sử dụng ở đây có nghĩa là “có vai trò làm ví dụ, mẫu hoặc minh họa”. Khía cạnh bất kỳ được mô tả ở đây là “làm ví dụ” không nhất thiết được hiểu là được ưu tiên hoặc có lợi hơn so với các khía cạnh khác”. Tương tự, thuật ngữ “các khía cạnh của sáng chế” không yêu cầu tất cả các khía cạnh của sáng chế đều bao gồm dấu hiệu, ưu điểm hoặc chế độ hoạt động đã được mô tả.

Thuật ngữ được sử dụng ở đây nhằm mục đích mô tả các khía cạnh cụ thể và không dự định giới hạn các khía cạnh của sáng chế. Như được sử dụng ở đây, các dạng số ít bao gồm cả dạng số nhiều trừ khi ngữ cảnh biểu thị khác một cách rõ ràng. Cũng cần hiểu rằng các thuật ngữ "gồm", "bao gồm", "gồm có" và/hoặc "kể cả", khi được sử dụng ở đây, chỉ rõ sự có mặt của các đặc điểm, số nguyên, bước, thao tác, thành phần, và/hoặc bộ phận được đề cập, nhưng không loại trừ sự có mặt hoặc sự thêm vào của một hoặc nhiều đặc điểm, số nguyên, bước, thao tác, thành phần, và/hoặc bộ phận khác, và/hoặc các nhóm của chúng.

Ngoài ra, nhiều khía cạnh được mô tả theo chuỗi các hành động sẽ được thực hiện bởi, ví dụ, các thành phần của thiết bị tính toán. Cần hiểu rằng một số hành động được mô tả ở đây có thể được thực hiện bởi các mạch riêng (ví dụ, mạch tích hợp chuyên dụng (application specific integrated circuit - ASIC), bởi các lệnh chương trình đang được thực thi bởi một hoặc nhiều bộ xử lý, hoặc bởi tổ hợp cả hai. Hơn nữa, các chuỗi hành động này được mô tả ở đây có thể được coi là được thể hiện toàn bộ trong dạng bất kỳ của phương tiện lưu trữ đọc được bằng máy tính có tập hợp lệnh máy tính được lưu trữ trong đó mà khi thực thi sẽ khiến cho bộ xử lý liên quan thực hiện chức năng được mô tả ở đây. Do đó, các khía cạnh khác nhau của sáng chế có thể được thể hiện ở nhiều dạng khác nhau, tất cả đều được dự tính là nằm trong phạm vi của đối tượng yêu cầu bảo hộ. Ngoài ra, với mỗi trong số các khía cạnh được mô tả ở đây, dạng tương ứng của khía cạnh bất kỳ như vậy có thể được mô tả ở đây là, ví dụ, "lôgic được tạo cấu hình để" thực hiện hành động đã mô tả.

Các khía cạnh làm ví dụ của sáng chế này đề cập đến các thao tác Persistent Write hiệu quả và hiệu suất cao cho bộ nhớ bất biến như DIMM bất biến (hoặc NVDIMM). Tương ứng, NVDIMM hoặc NVDIMM-P liên tục được mô tả là một hệ thống nhớ làm ví dụ hỗ trợ các thao tác Persistent Write theo các khía cạnh làm ví dụ. Thiết bị chủ có thể được tạo cấu hình để cung cấp các yêu cầu/lệnh làm ví dụ, ví dụ, cho các thao tác liên tục, và dữ liệu tương ứng cho hệ thống nhớ làm ví dụ, và hệ thống nhớ có thể được tạo cấu hình để thực hiện các thao tác Persistent Write được yêu cầu và cung cấp báo hiệu tương ứng cho thiết bị chủ như được mô tả chi tiết hơn trong các phần dưới đây.

Như được thể hiện trên Fig.1, hệ thống xử lý làm ví dụ 100 được thể hiện bao gồm máy chủ (host) 120 và hệ thống nhớ 130. Máy chủ 120 có thể bao gồm một hoặc nhiều phần tử xử lý như đơn vị xử lý trung tâm (central processing unit - CPU), bộ xử lý tín hiệu

số (digital signal processor - DSP), bộ xử lý đa phương tiện, bộ xử lý hệ thống, đơn vị xử lý đồ họa (graphics processing unit - GPU), bộ điều chế-giải điều chế (modem - modulator-demodulator), bộ xử lý ứng dụng, v.v., mặc dù các phần tử xử lý này không được minh họa một cách rõ ràng. Các phần tử xử lý này có thể tạo ra các yêu cầu để truy cập vào hệ thống nhớ 130. Bộ điều khiển bộ nhớ (không được thể hiện) có thể có mặt trong máy chủ 120 để điều khiển các yêu cầu truy cập này.

Hệ thống nhớ 130 có thể là bộ nhớ liên tục, ví dụ, NVDIMM-P theo sáng chế này. Hệ thống nhớ 130 được thể hiện bao gồm khối đầu vào/đầu ra (input/output - I/O) 132 và ngân hàng nhớ 134. Ngân hàng nhớ 134 có thể bao gồm bộ nhớ flash, DRAM, v.v.

Liên kết 110 được thể hiện giữa máy chủ 120 và hệ thống nhớ 130, với bus dữ liệu (DQ) 112, bus lệnh và địa chỉ (CA - command and address) 114, và đáp ứng 116 được xác định riêng. Máy chủ 120 có thể có khả năng cung cấp các lệnh và địa chỉ liên quan cho các yêu cầu truy cập bộ nhớ qua CA 114 và gửi/nhận dữ liệu qua DQ 112 (thể hiện là bus hai chiều). Đáp ứng 116, mặc dù được thể hiện riêng, có thể được tạo cấu hình như một phần CA 114 và có thể được thực thi như bus hai chiều trong một số trường hợp. Đáp ứng 116 có thể được sử dụng để cung cấp thông tin chẳng hạn như trạng thái của các lần Persistent Write trong một số khía cạnh làm ví dụ. Một số bus/dây khác cũng có thể có mặt trong liên kết 110 mặc dù chúng không được xác định riêng. Trong một số ví dụ, hệ thống nhớ 130 có thể sử dụng các bus riêng cho các đáp ứng xác định và không xác định, sẽ được mô tả chi tiết hơn dưới đây.

Theo một phương án thực hiện, trong đó hệ thống nhớ 130 có thể được tạo cấu hình như NVDIMM, có sự hỗ trợ thêm đối với cấu hình NVDIMM liên tục (NVDIMM-P - persistent NVDIMM) cho ít nhất một số thao tác, máy chủ 120 có thể có khả năng cung cấp một hoặc nhiều trong số các lệnh làm ví dụ tiếp theo sau cho hệ thống nhớ 130, ví dụ, trên CA 114:

- Lệnh READ (ví dụ, với sự mã hóa độ dài theo bội số 64B), cùng với định danh đọc (RID - read identification);
- Lệnh WRITE (ví dụ, lệnh ghi thông thường);
- Lệnh P-WRITE (ví dụ, lệnh Persistent Write, cùng với định danh ghi (WID - write identification) cho các lần Persistent Write, cùng với bit liên lục chỉ báo khi

ghi với WID nhất định cần tín hiệu hoàn thành Persistent Write (W_PER) (ví dụ, sẽ được cung cấp trên đáp ứng 116) từ hệ thống nhớ 130);

- Lệnh ADRx: Địa chỉ mở rộng;
- Lệnh SEND (ví dụ, lệnh cho hệ thống nhớ 130 để cung cấp trạng thái của yêu cầu dữ liệu đọc);
- Lệnh SEND_Status (ví dụ, lệnh cho hệ thống nhớ 130 để cung cấp chỉ báo lỗi, các WID, v.v. liên quan đến các thao tác liên tục từ hệ thống nhớ 130);
- Lệnh FLUSH (để xóa các thao tác ghi trước đó để được đẩy vào bộ nhớ liên tục)
- NOP (không hoạt động);
- Lệnh Speculative Read (ví dụ, được dùng để đọc bộ nhớ đệm); và
- Các lệnh Caching khác, các lệnh này có thể cụ thể cho từng phương án triển khai.

Như đã được đề cập trước đó, các bus riêng có thể được cung cấp trong liên kết 110 cho các đáp ứng xác định và không xác định từ hệ thống nhớ 130 đến máy chủ 120. Các đáp ứng xác định bao gồm siêu dữ liệu, thông tin lỗi/chặn lẻ như mã hóa điều khiển lỗi (error control coding-ECC) liên quan đến dữ liệu đọc được gửi trên DQ 112 đến máy chủ 120, v.v., có thể được ghép kênh trên các bus bắt nguồn từ các chốt ghép nối với I/O 132, như các chốt bit kiểm tra.

Trong số các bit ECC, có thể có ECC phương tiện dành riêng cho các phương án thực hiện của hệ thống nhớ 130 (ví dụ, như NVDIMM) và các bit ECC riêng của kênh trên DQ 112, ví dụ, có thể được chuẩn hóa để cho phép tương thích chéo trên một số phương án thực hiện.

Các bit siêu dữ liệu có thể bao gồm các RID trễ đối với các yêu cầu đọc gửi đi không theo thứ tự chương trình (trong đó, đối với các thao tác theo thứ tự, RID có thể được thiết lập thành trạng thái “không quan tâm”). Các bit siêu dữ liệu cũng có thể bao gồm tín dụng ghi (WC - write credit), tín dụng ghi này được dùng để chỉ định mức không sử dụng của các thao tác ghi phân bổ cho các máy chủ hoặc các phân tử xử lý nhất định của máy chủ 120. Các bit siêu dữ liệu cũng có thể bao gồm các bit đầu độc (poisoning) dữ liệu cho dữ liệu từ thiết bị người dùng như đã biết trong lĩnh vực kỹ thuật này, và các bit được xác định bởi người dùng khác.

Các đáp ứng không xác định theo sáng chế này có thể liên quan đến các thao tác liên tục và có thể được gửi qua báo hiệu dành riêng như đáp ứng 116 từ hệ thống nhớ 130 đến máy chủ 120, và có thể chỉ báo như sau:

- R_RDY: tín hiệu từ hệ thống nhớ 130 đến máy chủ 120 để chỉ báo rằng dữ liệu đọc là khả dụng;
- Wr_Per: tín hiệu từ hệ thống nhớ 130 đến máy chủ 120 để chỉ báo rằng việc Persistent Write đã hoàn thành; và
- ERROR: tín hiệu từ hệ thống nhớ 130 đến máy chủ 120 để chỉ báo các điều kiện lỗi như kiểm tra CRC, vi phạm tín dụng, hết thời gian chờ phương tiện, v.v.

Ví dụ, đối với một phương án thực hiện của hệ thống nhớ 130 như NVDIMM-P, các chốt giao thức sau đây có thể được xác định trong I/O 132. Bằng cách sử dụng quy trình điều chế độ rộng xung, các cấu hình tín hiệu đáp ứng và chốt sau đây có thể được thực hiện. Ví dụ, theo một phương án thực hiện tốc độ dữ liệu kép 5 (double-data rate 5 - DDR5) của NVDIMM-P, một dây được đánh dấu là RSP_n (một RSP_n được dành riêng cho mỗi kênh con) có thể được sử dụng để cung cấp báo hiệu sau đây: 2 xung đồng hồ thấp cho R_RDY, 4 xung đồng hồ thấp cho W_PER và 6 xung đồng hồ thấp cho MESSAGE. Mỗi xung thấp có thể theo sau là ít nhất 2 xung đồng hồ cao. Nếu cần tín hiệu ERROR riêng thì nó có thể được xác định là 8 xung đồng hồ thấp.

Đối với phương án thực hiện DDR4: hai chốt có thể được dùng để giải quyết các vấn đề về hiệu suất với một chốt (ODT1 và CKE1), trong đó ODT1 có 2 độ rộng xung đồng hồ thấp đối với R_RDY và 4 xung đồng hồ thấp cho MESSAGE, và CKE1 có 2 xung đồng hồ thấp cho W_PER. Mỗi xung thấp có thể theo sau là ít nhất 2 xung đồng hồ cao, và nếu cần tín hiệu ERROR riêng thì nó có thể được xác định là 6 xung đồng hồ thấp trên ODT1.

Theo một phương án thực hiện làm ví dụ của các lần Persistent Write, tổ hợp thích hợp bao gồm phần cứng, phần mềm, phần sụn, v.v. (ví dụ, các ứng dụng, bộ điều khiển, v.v.) có thể được tạo cấu hình để cho phép thông báo được cung cấp cho máy chủ 120 từ hệ thống nhớ 130 khi một hoặc nhiều yêu cầu ghi từ máy chủ 120 đến hệ thống nhớ 130 đạt được tính liên tục. Các thông báo này có thể cụ thể cho từng phương án triển khai, như được giải thích dưới đây.

Khi dữ liệu cần ghi cho thao tác ghi đến bộ đệm được bảo vệ khi mất điện trên bộ điều khiển phương tiện (ví dụ, bộ nhớ được bảo vệ khi mất điện của hệ thống nhớ 130), thao tác ghi có thể được coi là liên tục trong các thao tác ghi thông thường. Tuy nhiên đối với một số trường hợp ít khi xảy ra hoặc khi các bộ đệm của bộ điều khiển phương tiện không được bảo vệ khi mất điện, phần mềm sẽ đảm bảo rằng việc ghi được đầy hoàn toàn cho phương tiện NVM

Đối với một phương án thực hiện của hệ thống nhớ 130 như NVDIMM-P, các DIMM được hỗ trợ năng lượng bao gồm các cấu hình trong đó các bộ đệm nêu trên được bảo vệ khi mất điện, nghĩa là lệnh NVDIMM-P WRITE có thể được sử dụng ngay cả khi sự liên tục được là cần thiết cho các trường hợp bình thường. Ngoài ra, lệnh NVDIMM-P Flush, như được xác định ở đây, có thể được sử dụng để xóa tất cả việc ghi trong các bộ đệm của bộ điều khiển phương tiện cho bộ nhớ bất biến. Trong trường hợp theo lệnh Flush, chỉ có các lần ghi xảy ra trước lệnh Flush mới được đảm bảo là được làm cho liên tục đối với bộ nhớ bất biến. Các lệnh được thực thi bằng phần mềm như các thao tác chặn thích hợp có thể được sử dụng sau lần ghi cuối và trước lệnh Flush được đưa ra để đảm bảo duy trì đúng thứ tự ghi (ví dụ, khi máy chủ 120 có thể được tạo cấu hình để gửi yêu cầu Persistent Write ngoài thứ tự chương trình).

Mặc dù các DIMM không được hỗ trợ năng lượng có thể thường ít được sử dụng hơn các DIMM được hỗ trợ năng lượng đã nêu ở trên, lệnh Persistent Write NVDIMM-P có thể được sử dụng khi sự liên tục cần thiết cho các DIMM không được hỗ trợ năng lượng cũng được sử dụng. Bộ điều khiển bộ nhớ của máy chủ 120, như nêu trên, có thể được tạo cấu hình để xác định khi đưa ra lệnh Persistent Write. Trong trường hợp này, hệ thống nhớ 130 được kỳ vọng để cung cấp thông báo rõ ràng khi lệnh Persistent Write được hoàn thành, như sẽ được mô tả dựa vào Fig.2A. Hơn nữa, lệnh NVDIMM-P Flush có thể cũng được sử dụng như trước đó để xóa tất cả lần ghi (thậm chí cả các lần ghi không liên tục) trong bộ nhớ bất biến.

Như được thể hiện trên Fig.2A, tập hợp các giao dịch làm ví dụ được thể hiện giữa máy chủ 120 và hệ thống nhớ 130 để minh họa các khía cạnh của lệnh Persistent Write. Có một số tính năng của lệnh Persistent Write làm ví dụ (Wr_Per) (hoặc đơn giản là, “Persistent Write”) có thể phổ biến với lệnh READ mô tả ở trên từ máy chủ 120. Các tính năng này bao gồm thao tác định danh ghi (WID - write identification) chung, mà có thể là

mã định danh nhiều bit để nhận dạng các lệnh ghi cụ thể. Tập hợp WID làm ví dụ 210 được thể hiện, mỗi trong số đó có thể rộng tới 16-bit, bao gồm một bit hợp lệ “Vld” (do đó, ví dụ, có thể có tới 31 WID trong gói lệnh 64 byte được gửi trên CA 114). Lệnh Persistent Write cũng có thể có trường dự trữ trong quá trình mã hóa WID để trả lại trạng thái lệnh Flush mà sẽ được giải thích thêm trong các đoạn sau.

Theo một khía cạnh, máy chủ 120 có thể được tạo cấu hình để đưa ra lệnh Persistent Write chỉ khi máy chủ 120 có tín dụng Persistent Write liên quan có sẵn. Tín dụng Persistent Write (tương tự tín dụng đọc (Read credit) đã biết trong lĩnh vực kỹ thuật này) có thể được xác định trong khi tạo cấu hình và được quản lý bởi máy chủ 120, và có thể phản ánh số lượng các lần Persistent Write chưa giải quyết mà máy chủ 120 được phép đưa ra.

Khi được đưa ra, máy chủ 120 có thể được tạo cấu hình để theo dõi các lần Persistent Write chưa giải quyết dựa trên các WID 210 tương ứng của chúng. Trên Fig.2A (có tham chiếu kết hợp đến Fig.1), hai lần Persistent Write (P-Write 1 có địa chỉ thứ nhất và WID, và P-Write 2 có địa chỉ thứ hai và WID) được đánh dấu là 202a và 204a được thể hiện, được đưa ra từ máy chủ 120 đến hệ thống nhớ 130 trên CA 114, cùng với dữ liệu tương ứng, dữ liệu ghi 202b và dữ liệu ghi 204b trên DQ 112, ví dụ.

Hệ thống nhớ 130 được tạo cấu hình để đưa ra đáp ứng “Wr_Per” trên đáp ứng 116, đối với lệnh Persistent Write cụ thể, một khi tất cả dữ liệu cho lệnh Persistent Write đã được ghi vào bộ nhớ bất biến trong hệ thống nhớ 130. Wr_Per 202c và Wr_Per 204c được thể hiện đối với các lần Persistent Write tương ứng 202a và 204a. Tuy nhiên, Wr_Per 202c và Wr_Per 204c được thể hiện là sẽ được gửi theo các trình tự khác nhau so với các lần Persistent Write 202a và 204a đã được nhận bởi hệ thống nhớ 130 để minh họa rằng các đáp ứng không cần phải theo thứ tự chương trình hoặc theo thứ tự mà trong đó các yêu cầu Persistent Write được nhận từ máy chủ 120. Theo một khía cạnh, hệ thống nhớ 130 có thể xác nhận tín hiệu “Req” trên đáp ứng 116 cùng với việc mã hóa thích hợp cho bản tin “Write_Rdy” cho các đáp ứng Wr_Per.

Hơn nữa, máy chủ 120 cũng có thể được tạo cấu hình để đưa ra lệnh “Send-Status for WID” được chỉ định số tham chiếu là 206a, tại thời điểm bất kỳ, để xác định trạng thái của các lần Persistent Write chưa giải quyết của nó. Đáp lại, hệ thống nhớ 130 có thể được

tạo cấu hình để đưa ra gói trạng thái với các WID của các lần Persistent Write đã hoàn thành, ví dụ, trong khối có độ dài 8 hay “BL8” (burst length of 8) truyền trên DQ 112.

Như đã đề cập trước đó, tối đa 31 WID 210 có thể được gói trong mỗi gói trạng thái 64B, trong đó đối với mỗi WID 210 có thể là 16 bit được gán cho WID 15 bit và bit hợp lệ, được kết hợp. Hơn nữa, hệ thống nhớ 130 cũng có thể dùng trường siêu dữ liệu đã nêu để trả lại trạng thái cho các lệnh ghi khác. Máy chủ 120 có thể sử dụng các WID trả lại 210 trong gói trạng thái WID 206b, ví dụ, để kết thúc việc theo dõi các lần Persistent Write chưa giải quyết.

Theo một số khía cạnh, hai hoặc nhiều lệnh Persistent Write có thể được nhóm. Ví dụ, tập hợp lệnh Persistent Write 64B có thể được nhóm để chuyển (hoặc ghi vào bộ nhớ bất biến) trong trường hợp DIMM không được hỗ trợ năng lượng, ví dụ. Ví dụ, phương án thực hiện có thể bao gồm khối Persistent Write để được cấp cho hệ thống nhớ 130 từ máy chủ 120, trong đó hệ thống nhớ 130 có thể được tạo cấu hình để thu thập tối đa khối Persistent Write trong bộ đệm và chuyển tất cả các khối Persistent Write cùng một lúc, việc này có thể làm cho hiệu suất được cải thiện. Tuy nhiên, cần phải hiểu rằng việc nhóm các lần Persistent Write và chuyển chúng trong khối là không cần thiết cho các DIMM được hỗ trợ năng lượng trong đó các bộ đệm được bảo vệ khi mất điện.

Một số sửa đổi sau đây có thể được áp dụng cho lệnh Persistent Write để thực hiện chuyển nhóm mô tả ở trên. Máy chủ 120 có thể chọn một WID (từ các WID 210, ví dụ) cho tập hợp hai hoặc nhiều lệnh ghi. Ví dụ, bit bổ sung được gọi là “Persist” có thể được bổ sung vào lệnh Persistent Write khi gửi trên CA 114. Bit Persist có thể được sử dụng để xác định khi toàn bộ nhóm các lần Persistent Write đã được gửi đến hệ thống nhớ 130.

Ví dụ, ba lệnh Persistent Write 64B có thể được nhóm cùng nhau như sau bằng cách sử dụng WID = 5 trong một ví dụ minh họa. Lệnh Persistent Write thứ nhất (WID=5, Persist=0), lệnh Persistent Write thứ hai (WID=5, Persist=0), và lệnh Persistent Write thứ ba (WID=5, Persist=1) có thể được gửi trên CA 114. Hệ thống nhớ 130 có thể được tạo cấu hình để thu thập các lần Persistent Write với WID=5 trong bộ đệm đồng thời bit Persist là 0, và khi đến lần Persistent Write cuối cùng có bit Persist được đặt bằng 1 thì bắt đầu quy trình chuyển liên tục.

Theo một phương án thực hiện, chỉ một lệnh Persistent Write với một bit Persist được đặt bằng 1 có thể được tạo cấu hình để có đáp ứng Wr_Per từ hệ thống nhớ 130 (ví

dụ, chỉ lệnh Persistent Write thứ ba trong ví dụ ở trên) đối với nhóm Persistent Write. Việc này có thể giảm lưu lượng trên đáp ứng 116.

Theo một số khía cạnh, các lệnh Persistent Write với các WID khác nhau có thể được đan xen, ví dụ, trên CA 114. Do đó, việc nhóm các lệnh Persistent Write để chuyển liên tục không ngụ ý rằng các lệnh Persistent Write trong nhóm với cùng một WID được gửi liên tục từ máy chủ 120.

Theo một số khía cạnh, để giải quyết các điều kiện tranh đấu có thể phát sinh trong các đáp ứng Wr_Per của các lần Persistent Write, phương pháp trạng thái ID nhóm ghi (WGID - Write Group ID) có thể được sử dụng để nhóm các trạng thái của một hoặc nhiều lần Persistent Write, bằng cách sử dụng các phép ánh xạ bit khác nhau, như ánh xạ bit hoàn thành WGID và ánh xạ bit chờ WGID, như sẽ được mô tả dựa vào Fig.2B dưới đây. Xem xét các lệnh Persistent Write với các WID tương ứng, hệ thống nhớ 130 có thể khẳng định Wr_Per tương ứng (được gọi là “W_PER” đối với trường hợp này) cho mỗi lệnh Persistent Write với Persist = 1 và cho mỗi lần hoàn thành Flush. Máy chủ 120 có thể sử dụng lệnh Send-W_PER-Status khác sau khi nhận một hoặc nhiều W_PER (trong đó, máy chủ 120 cũng có thể duy trì tổng số các W_PER, được gọi là W_PER-Count). Hệ thống nhớ 130 có thể trả về trạng thái hoàn thành WGID với các bit được hoàn thành chỉ dựa trên các W_PER đã được khẳng định. Lần lượt, máy chủ 120 có thể cập nhật danh sách cho WGID, hoặc “danh sách WGID ” và giảm W_PER-Count dựa trên số lần hoàn thành.

Trong một số trường hợp, lỗi không thể sửa chữa (UE - uncorrectable error) có thể xảy ra trong các lần giao dịch, mà sẽ được mô tả dựa vào hình vẽ trên Fig.2C. Khi có UE trong lệnh Send-W_PER-Status, máy chủ 120 có thể dừng việc đưa ra các lệnh Persistent Write/Flush và Send-W_PER-Status. Máy chủ 120 có thể gửi lệnh đọc trạng thái được gọi là Xread-Status cho hệ thống nhớ 130. Hệ thống nhớ 130 lần lượt thu thập tất cả các lệnh Persistent Write trước khi nhận lệnh Xread-Status để trả lại lệnh WGID-Pending Status cho máy chủ 120 (các gói trạng thái bao gồm lệnh khẳng định W_PER trước khi nhận được lệnh RD_RDY) và hệ thống nhớ 130 có thể tiếp tục đưa ra W_PER trong lúc đọc trạng thái. Máy chủ 120 có thể cập nhật WGID List được duy trì bằng máy chủ 120 và giảm W_PER-Count dựa trên các lệnh ghi chưa hoàn thành Máy chủ 120 sau đó có thể bắt đầu lại đưa ra các lệnh Persistent Write/Flush.

Đối với một phương án thực hiện DIMM được hỗ trợ năng lượng của hệ thống nhớ 130, trong giao thức thông thường, máy chủ 120 có thể đưa ra các lệnh Persistent Write (với Persist = 0/1), và Flush, nhưng hệ thống nhớ 130 sẽ không khẳng định W_PER cho mỗi lệnh Persistent Write với Persist=1, nhưng hệ thống nhớ 130 sẽ khẳng định W_PER cho lệnh Flush khi lệnh Flush hoàn thành. Trong trường hợp thực hiện WGID, quy trình xử lý W_PER bởi hệ thống nhớ 130 vẫn giống như giao thức thông thường chỉ dành cho các lệnh Flush. Ánh xạ bit trạng thái hoàn thành WGID được cung cấp bởi hệ thống nhớ 130 sẽ có tập hợp các bit WGID Flush khi chúng hoàn thành. Khi có UE trong Send-W_PER-Status, thao tác vẫn giống như trường hợp thông thường, ngoại trừ trạng thái chờ WGID chỉ thích hợp cho các lệnh Flush.

Tín dụng của các phương án thực hiện WGID có thể được xử lý như sau. Tín dụng riêng có thể được duy trì cho các lần ghi trạng thái hay Xwrite và cho các lần Persistent Write, trong đó máy chủ 120 có thể xác định cách thức nhóm tín dụng có thể được phân bổ bởi hệ thống nhớ 130. Lệnh Incremental Credit Return có thể được cung cấp bởi lệnh Read Metadata, trong đó sơ đồ mã hóa để trả về Xwrite hoặc tín dụng Persistent Write có thể được sử dụng. Lệnh trả lại X-Read-Status có thể có sẵn cho lệnh Xwrite và các khe bộ đệm Persistent Write dựa trên sự phân bổ tín dụng.

Theo một phương án thực hiện, chẳng hạn, phương án này sẽ được mô tả dựa vào các hình vẽ từ Fig.2B đến Fig.2C, hệ thống nhớ 130 có thể hoàn thành lệnh Persistent Write (được gọi là PWRITE trong sáng chế này) và các lệnh Flush theo thứ tự bất kỳ. Để duy trì lệnh PWRITE riêng cho phương tiện, máy chủ 120 có thể đưa ra PWRITE cho một WGID nhất định có Persist = 1 hoặc đưa ra PWRITE có Persist = 0 theo sau là bất kỳ trong số các loại Flush. Hệ thống nhớ 130 có thể đưa ra W_PER cho mỗi PWRITE hoàn thành mà có Persist=1 trong lệnh cũng như mọi Flush hoàn thành. Nếu nhiều PWRITE được nhóm với một WGID có Persist=1 chỉ trong một PWRITE cuối kết thúc nhóm, thì hệ thống nhớ 130 có thể đưa ra W_PER chỉ khi toàn bộ nhóm PWRITE hoàn thành.

Như được thể hiện trên Fig.2B, W_PER xử lý sẽ được mô tả cho trường hợp thông thường. Hệ thống 250 được thể hiện với máy chủ 120 và hệ thống nhớ 130. Nếu có sẵn cả Write-Credit và WGID tự do, thì máy chủ 120 có thể đưa ra một hoặc nhiều lệnh PWRITE hoặc FLUSH được thể hiện là 252a, 254a. Máy chủ 120 có thể theo dõi các lệnh PWRITE

hoặc FLUSH đã đưa ra 252a, 254a trong danh sách Host-WGID-Pending (không được thể hiện, nhưng có thể được duy trì trong máy chủ 120).

Tương ứng, hệ thống nhớ 130 có thể chấp nhận và theo dõi các lệnh PWRITE hoặc FLUSH chờ hoàn thành 252a, 254a trong danh sách chờ hoàn thành DIMM-WGID (không được thể hiện). Hệ thống nhớ 130 có thể thực thi các lệnh PWRITE hoặc FLUSH chờ hoàn thành 252a, 254a và khẳng định các lệnh W_PER 254b và 252b tương ứng (lưu ý, được thể hiện theo thứ tự ngược của các lệnh PWRITE hoặc FLUSH 252a, 254a nhận được) cho máy chủ 120 sau khi hoàn thành mỗi lệnh nhận được tương ứng.

Hệ thống nhớ 130 có thể thu thập các lệnh PWRITE hoặc FLUSH 252a, 254a nhận được hoàn thành trong ánh xạ bit hoàn thành WGID 260, cho một số lần cập nhật 260a, 260b, 260c, v.v., được thể hiện. Hệ thống nhớ 130 còn có thể xóa các lệnh PWRITE hoặc FLUSH hoàn thành 252a, 254a khỏi danh sách DIMM-WGID-Pending.

Máy chủ 120 có thể duy trì tổng số sự kiện W_PER nhận được, ví dụ, để nhận các W_PER 254b, 252b, được gọi là W_PER-Count. Đồng thời, máy chủ 120 có thể xử lý các sự kiện W_PER nhận được như sau: Nếu W_PER-Count > 0, thì máy chủ 120 có thể đưa ra yêu cầu trạng thái thể hiện là Send-W_PER Status 256a. Sau thời gian định trước, được gọi là thời gian Tsend, hệ thống nhớ 130 có thể gửi ảnh chụp nhanh của ánh xạ bit WGID-Completed 260 vào thời điểm đó (trong trường hợp này là 260b) trong đáp ứng được thể hiện là WGID_Status 256b cho máy chủ 120. Ảnh chụp nhanh có thể gồm các lần hoàn thành của các W_PER được đưa ra đến tận lúc bắt đầu truyền WGID_Status 256b cho máy chủ 120.

Theo một số khía cạnh, giá trị hoàn thành 1 tại giá trị tối thiểu được ghi vào ảnh chụp nhanh. Hệ thống nhớ 130 dọn dẹp các vị trí bit trong ánh xạ bit WGID-Completed 260 dựa trên các lần hoàn thành được gửi đi trong WGID_Status 256b, được thể hiện bằng cách chuyển tiếp ánh xạ bit WGID-Completed 260b đến ánh xạ bit WGID-Completed 260c sau khi thiết lập lại hoặc dọn dẹp các vị trí bit.

Máy chủ 120 nhận WGID-Status 256b và có thể trích thông tin liên quan đến các WGID đã hoàn thành. Tương ứng, máy chủ 120 có thể giải phóng các WGID đã hoàn thành từ danh sách Host-WGID-Pending và giảm W_PER-Count bằng số lần hoàn thành nhận được trong ánh xạ bit WGID-Completed 260 (ví dụ, giảm số lượng 2 dựa trên hai W_PER nhận được như được chỉ báo bởi ánh xạ bit WGID-Completed 260b). Máy chủ

120 có thể lặp lại quy trình trên bắt đầu bằng việc giám sát W_PER-Count và nếu W_PER-Count>0, thì đưa ra yêu cầu trạng thái Send-W_PER Status 256a khác cho hệ thống nhớ 130.

Trong một số phương án thực hiện làm ví dụ, máy chủ 120 và hệ thống nhớ 130 có thể tiếp tục đưa ra và thực thi các lệnh PWRITE mới trong khi quá trình xử lý hiện tượng W_PER đang thực hiện. Mặc dù W_PER-Count và các danh sách chưa hoàn thành như danh sách Host-WGID-Pending, danh sách DIMM-WGID-Pending, v.v., đã được mô tả cho một phương án thực hiện làm ví dụ, các cấu trúc thay thế để đạt được chức năng mô tả ở trên có thể được sử dụng mà không vượt ra khỏi phạm vi của sáng chế.

Như được thể hiện trên Fig.2C, hệ thống 270 để xử lý lỗi không thể sửa chữa (Uncorrectable Error - UE) của kênh tiếp theo SEND-W_PER-Status từ máy chủ 120 dẫn đến mất các lần hoàn thành được gửi đi trong WGID_Status 256b (được giải thích trên Fig.2B ở trên) từ hệ thống nhớ 130 được thể hiện. Hơn nữa, lưu ý rằng hệ thống nhớ 130 có thể đã dọn dẹp các lần hoàn thành trước đó từ ánh xạ bit WGID-Completed 260 trên Fig.2B.

Do đó, trong giao thức để khôi phục từ các lỗi như vậy trong hệ thống 270, máy chủ 120 có thể bắt đầu quy trình khôi phục bằng cách dừng đưa ra các lệnh PWRITE hoặc FLUSH mới (ví dụ, PWRITE-3 hoặc FLUSH-3 272a không được đưa ra, thể hiện là các đường nét đứt để chỉ báo dòng thời gian mà chúng đã đưa ra có lỗi không xảy ra), còn hệ thống nhớ 130 có thể tiếp tục đưa ra các hiện tượng RD_RDY và/hoặc W_PER cho các lệnh đọc được hoàn thành hoặc lệnh PWRITE hoặc lệnh FLUSH (ví dụ, W_PER 254b được thể hiện sẽ được đưa ra trong khi 252b không được đưa ra đến khi phục hồi lỗi). Máy chủ 120 có thể còn tiếp tục đưa ra lệnh SEND và cập nhật W_PER-Count.

Sau thời gian trễ tối thiểu định trước cho tín hiệu có khả năng ghi, được gọi là TWE_Delay tiếp theo PWRITE cuối, máy chủ 120 cung cấp XREAD-STATUS 274a cho hệ thống nhớ 130, và hệ thống nhớ 130 có thể chuẩn bị gói trạng thái hoàn thành có ảnh chụp nhanh ánh xạ bit WGID-Pending 280, ánh xạ bit này là ánh xạ bit khác được tạo ra ngoài ánh xạ bit WGID-Completed 260 được mô tả ở trên, trong đó ánh xạ bit WGID-Pending 280 bao gồm trạng thái của tất cả các Pending PWRITE/Flush. Hệ thống nhớ 130 có thể khẳng định RD_RDY 276b, và máy chủ 120 có thể đưa ra SEND 278a để đáp lại.

Sau đó hệ thống nhớ 130 có thể trả lại gói trạng thái được chuẩn bị 278b từ đó máy chủ 120 có thể trích và xử lý ánh xạ bit WGID-Pending 280 nhận được trong gói trạng thái 278b. Máy chủ 120 có thể giải phóng các WGID thích hợp từ danh sách theo dõi Host-WGID-Pending và giảm số lượng WGID đã giải phóng cho W_PER-Count. Sau đó máy chủ 120 có thể lặp lại quy trình bắt đầu bằng cách đưa ra các lệnh PWRITE/FLUSH mới và xử lý các lệnh W_PER chưa hoàn thành tại thời điểm này cho mỗi trang trước đó.

Theo một số khía cạnh, gói trạng thái 278b được tạo cấu hình để chỉ báo liệu nó có ánh xạ bit WGID-Completed 260 hoặc ánh xạ bit WGID-Pending 280 hay không. Các gói trạng thái đáp ứng W_PER bao gồm ánh xạ bit WGID-Completed 260, trong khi tất cả các gói trạng thái khác bao gồm ánh xạ bit WGID-Pending 280. Thời gian TWE_Delay được tạo cấu hình để tính toán thời gian để thu được thông báo lỗi từ hệ thống nhớ 130 cho PWRITE cuối cùng đưa ra từ máy chủ 120 và thời gian chờ từ khi phát hiện UE trước khi XREAD-STATUS 274a đưa ra từ máy chủ 120 có thể thay đổi phụ thuộc khi PWRITE cuối được đưa ra.

Như được thể hiện trên Fig.3, việc mã hóa làm ví dụ đối với các lần Persistent Write, ví dụ, cho một phương án thực hiện DDR5 của hệ thống nhớ 130 được thể hiện. Trường CA1 thường được dùng để phân biệt giữa các lệnh 1UI và 2UI trong công nghệ DDR5 và có thể có thể được giữ lại để triển khai NVDIMM-P. CA 114, trong một số phương án thực hiện có thể được tạo cấu hình ở tốc độ DDR cho DDR5 chỉ với 7 chốt, và trong các trường hợp như vậy, quy trình mã hóa lệnh riêng biệt có thể được sử dụng cho lệnh Persistent Write, ví dụ, như được thể hiện trên Fig.3.

Trên Fig.3, nếu bit Persist = 1 thì việc này chỉ báo rằng hệ thống nhớ 130 sẽ đẩy tất cả các lần Persistent Write liên quan đến WID tương ứng cho bộ nhớ bất biến. Nếu có một lệnh 64B Persistent Write trong nhóm, thì bit Persist có thể được đặt bằng 1. Đối với các lần Persistent Write lớn hơn 64B, tất cả các lần Persistent Write có thể có cùng một WID, với lần Persistent Write cuối cùng có bit Persist của nó được đặt bằng 1 còn các lần Persistent Write còn lại có các bit Persist của chúng được đặt bằng 0.

Ngoài các giao dịch trên, như đã được giới thiệu trong các đoạn trước, lệnh khác cũng có thể được sử dụng kết hợp với các Persistent Write, được gọi là lệnh FLUSH. Lệnh FLUSH được tạo cấu hình để chỉ báo cho hệ thống nhớ 130 mà tất cả các thao tác ghi trước đó được đệm (ví dụ, trong bộ nhớ không liên tục hoặc bộ nhớ khả biến) sẽ được đẩy vào

bộ nhớ liên tục, lưu ý rằng các thao tác ghi trong tương lai có thể không bị ảnh hưởng một cách tương tự hoặc bị đẩy đến bộ nhớ liên tục khi sử dụng lệnh FLUSH.

Khi quá trình thi hành của FLUSH được hoàn thành, hệ thống nhớ 130 có thể một lần nữa xác nhận Wr_Per, ví dụ, trên đáp ứng 116 với máy chủ 120, tương tự trường hợp của lệnh Persistent Write được mô tả ở trên.

Hơn nữa, máy chủ 120 còn có thể cung cấp lệnh, Send-Status cho các WID (tương tự như các lệnh Persistent Write) cho hệ thống nhớ 130 trong trường hợp của lệnh FLUSH, trong đó hệ thống nhớ 130 có thể đáp ứng lại gói trạng thái WID với WID dự trữ duy nhất để chỉ báo việc hoàn thành thi hành lệnh FLUSH (ví dụ, WID có tất cả các bit được đặt bằng 1 có thể là WID dự trữ được dùng để chỉ báo việc hoàn thành thi hành FLUSH).

Theo một phương án thực hiện, duy nhất một lệnh FLUSH chưa giải quyết từ máy chủ 120 có thể được cho phép. Vì vậy, trong phương án thực hiện này, máy chủ 120 có thể phải chờ đáp ứng hoàn thành FLUSH từ hệ thống nhớ 130 trước khi gửi lệnh FLUSH khác. Theo các phương án thực hiện khác, các lệnh FLUSH có thể được đi cùng với các ID FLUSH (ví dụ, được chọn từ các trường WID dự trữ) và đáp ứng tương ứng với lệnh Send-Status có thể khiến hệ thống nhớ 130 trả lại các ID FLUSH mà quá trình thực thi FLUSH của nó đã được hoàn thành.

Cần phải hiểu rằng các khía cạnh bao gồm một số phương pháp để thực hiện các quy trình, các chức năng và/hoặc thuật toán được mô tả ở đây. Ví dụ, Fig.4A minh họa phương pháp 400 làm ví dụ để thực hiện các thao tác liên tục.

Khối 402 bao gồm bước nhận, tại hệ thống nhớ (ví dụ, hệ thống nhớ 130), lệnh Persistent Write (ví dụ, Persistent Write 202a) và dữ liệu ghi liên kết (ví dụ, dữ liệu 202b) từ máy chủ (ví dụ, máy chủ 120).

Khối 404 bao gồm bước thực hiện lệnh Persistent Write của dữ liệu ghi cho bộ nhớ bất biến (ví dụ, cho bộ nhớ bất biến trong hệ thống nhớ 130) trong hệ thống nhớ dựa trên lệnh Persistent Write. Mã định danh ghi (WID) kết hợp với lệnh Persistent Write có thể được nhận từ máy chủ và khi hoàn thành thành công lệnh Persistent Write, chỉ báo hoàn thành lệnh Persistent Write (Wr_Per) cùng với WID liên quan (ví dụ, Wr_Per 202c) có thể được cung cấp cho máy chủ.

Tương tự, Fig.4B minh họa phương pháp làm ví dụ khác 450 để thực hiện các thao tác liên tục.

Khối 452 bao gồm bước cung cấp, từ máy chủ (ví dụ, máy chủ 120) cho hệ thống nhớ (ví dụ, hệ thống nhớ 130), lệnh Persistent Write (ví dụ, Persistent Write 202a) và dữ liệu ghi liên kết (ví dụ, dữ liệu 202b) trong đó lệnh Persistent Write chỉ báo cho hệ thống nhớ để thực hiện lệnh Persistent Write của dữ liệu ghi cho bộ nhớ bất biến.

Khối 454 bao gồm việc cung cấp mã định danh ghi (WID) (ví dụ, WID 210) liên quan đến lệnh Persistent Write cho hệ thống nhớ từ máy chủ.

Thiết bị làm ví dụ trong đó các khía cạnh của sáng chế này có thể được dùng, sẽ được mô tả dựa vào Fig.5. Fig.5 là sơ đồ khối thể hiện thiết bị tính toán 500. Thiết bị tính toán 500 có thể tương ứng với một phương án thực hiện làm ví dụ của hệ thống xử lý 100 trên Fig.1, trong đó bộ xử lý 120' có thể là một trong các các phần tử xử lý của máy chủ 120. Bộ xử lý 120' được thể hiện một cách rõ ràng để được ghép nối với hệ thống nhớ 130 qua liên kết 110, các chi tiết khác về liên kết 110 được bỏ qua để dễ hiểu. Bộ xử lý 120', liên kết 110, và hệ thống nhớ 130 có thể được tạo cấu hình để thực hiện các phương pháp 400-450 như mô tả ở trên. Cần phải hiểu rằng các cấu hình bộ nhớ khác đã biết trong lĩnh vực kỹ thuật này như bao gồm một hoặc nhiều cấp bộ nhớ đệm, mặc dù không được thể hiện, có thể có mặt trong thiết bị tính toán 500.

Fig.5 cũng thể hiện bộ điều khiển hiển thị 526 được nối với bộ xử lý 120 và với màn hình 528. Trong một số trường hợp, thiết bị tính toán 500 có thể được dùng để truyền thông không dây và Fig.5 còn thể hiện các khối tùy chọn trong các đường nét đứt, như bộ mã hóa/bộ giải mã (CODEC - coder/decoder) 534 (ví dụ, CODEC âm thanh và/hoặc giọng nói) ghép nối với bộ xử lý 120' và loa 536 và micrô 538 có thể được ghép nối với CODEC 534; và anten không dây 542 ghép nối với bộ điều khiển không dây 540 được ghép nối với bộ xử lý 120'. Trong trường hợp có một hoặc nhiều trong số các khối tùy chọn này, theo một khía cạnh cụ thể, bộ xử lý 120', bộ điều khiển hiển thị 526, hệ thống nhớ 130, và bộ điều khiển không dây 540 được bao gồm trong thiết bị hệ thống trong gói hoặc hệ thống trên chip 522.

Do đó, theo một khía cạnh cụ thể, thiết bị đầu vào 530 và nguồn điện 544 được nối với thiết bị hệ thống trên chip 522. Hơn nữa, theo một khía cạnh cụ thể, như được thể hiện trên Fig.5, trong đó có một hoặc nhiều khối tùy chọn, màn hình 528, thiết bị đầu vào 530,

loa 536, micrô 538, anten không dây 542, và nguồn điện 544 ở bên ngoài thiết bị hệ thống trên chip 522. Tuy nhiên, mỗi màn hình 528, thiết bị đầu vào 530, loa 536, micrô 538, anten không dây 542, và nguồn điện 544 có thể được nối với một bộ phận của thiết bị hệ thống trên chip 522, chẳng hạn như giao diện hoặc bộ điều khiển.

Cần lưu ý rằng mặc dù Fig.5 nói chung mô tả thiết bị tính toán, bộ xử lý 120' và hệ thống nhớ 130, cũng có thể được tích hợp thành hộp giải mã, máy chủ, thiết bị nghe nhạc, thiết bị xem video, thiết bị giải trí, thiết bị dẫn đường, thiết bị hỗ trợ cá nhân (PDA - personal digital assistant), bộ dữ liệu định vị cố định, máy tính, máy tính xách tay, máy tính bảng, thiết bị truyền thông, điện thoại di động, hoặc các thiết bị tương tự khác.

Người có hiểu biết trung bình trong lĩnh vực kỹ thuật này hiểu rằng thông tin và tín hiệu có thể được biểu diễn bằng cách sử dụng công nghệ và kỹ thuật bất kỳ trong số nhiều công nghệ và kỹ thuật khác nhau. Ví dụ, dữ liệu, lệnh, chỉ lệnh, thông tin, tín hiệu, bit, ký hiệu, và chip mà có thể được mô tả trong suốt phần mô tả ở trên có thể được thể hiện bằng điện áp, dòng điện, sóng điện từ, các từ trường hoặc hạt từ, các trường hoặc hạt quang học, hoặc dạng kết hợp bất kỳ của chúng.

Người có hiểu biết trung bình trong lĩnh vực kỹ thuật này hiểu rằng các khối logic, mô đun, mạch, và các bước thuật toán minh họa khác nhau được mô tả liên quan đến các khía cạnh được bộc lộ trong bản mô tả này có thể được thực hiện dưới dạng phần cứng điện tử, phần mềm máy tính, hoặc kết hợp của cả hai. Để minh họa rõ tính hoán đổi của phần cứng và phần mềm, các thành phần, khối, mô đun, mạch và bước minh họa khác nhau đã được mô tả trên đây nhìn chung là về mặt chức năng của chúng. Chức năng như vậy có được thực thi dưới dạng phần cứng hay phần mềm hay không tùy thuộc vào ứng dụng cụ thể và ràng buộc thiết kế áp đặt lên toàn bộ hệ thống. Người có hiểu biết trung bình trong lĩnh vực kỹ thuật này có thể thực thi chức năng được mô tả theo nhiều cách khác nhau đối với từng ứng dụng cụ thể, nhưng các quyết định thực hiện này không nên được hiểu là vượt quá phạm vi của sáng chế.

Các phương pháp, trình tự và/hoặc thuật toán được mô tả liên quan đến các khía cạnh được bộc lộ ở đây có thể được thể hiện trực tiếp trong phần cứng, trong mô đun phần mềm được thực thi bởi bộ xử lý, hoặc kết hợp của cả hai. Mô đun phần mềm có thể nằm trong bộ nhớ RAM, bộ nhớ flash, bộ nhớ ROM, bộ nhớ EPROM, bộ nhớ EEPROM, thanh ghi, đĩa cứng, đĩa có thể tháo rời, CD-ROM, hoặc bất kỳ dạng phương tiện lưu trữ nào

khác được biết đến trong lĩnh vực này. Phương tiện lưu trữ làm ví dụ được ghép nối với bộ xử lý sao cho bộ xử lý có thể đọc và ghi thông tin vào phương tiện lưu trữ này. Theo cách khác, phương tiện lưu trữ có thể được tích hợp vào bộ xử lý.

Do đó, một khía cạnh của sáng chế có thể bao gồm phương tiện đọc được bằng máy tính gồm phương pháp thực hiện các lần Persistent Write. Do đó, sáng chế không bị giới hạn ở các ví dụ đã được minh họa và phương tiện bất kỳ để thực hiện chức năng được mô tả ở đây cũng được đề cập trong các khía cạnh của sáng chế.

Trong khi phần mô tả trên đây thể hiện các khía cạnh minh họa, cần lưu ý rằng những thay đổi và cải biến khác nhau có thể được thực hiện mà không nằm ngoài phạm vi của sáng chế như được xác định bởi bộ yêu cầu bảo hộ kèm theo. Các chức năng, bước và/hoặc hành động của các yêu cầu bảo hộ phương pháp theo các khía cạnh của sáng chế được mô tả ở đây không cần được thực hiện theo thứ tự cụ thể. Hơn nữa, mặc dù các thành phần sáng chế có thể được mô tả hoặc được nêu ở dạng số ít, điều này cũng dự tính bao gồm dạng số nhiều trừ khi giới hạn ở dạng số ít được nêu rõ ràng.

YÊU CẦU BẢO HỘ

1. Phương pháp thực hiện các thao tác liên tục, phương pháp này bao gồm các bước:

nhận, bởi modul nhớ hai hàng chân bất biến (non-volatile dual in-line memory module – NVDIMM) được tạo cấu hình để hỗ trợ các lần Persistent Write, lệnh Persistent Write và dữ liệu ghi liên quan từ máy chủ (host);

thực hiện ghi liên tục dữ liệu ghi vào bộ nhớ bất biến trong NVDIMM dựa trên lệnh Persistent Write;

nhận mã định danh ghi (write identification - WID) liên quan đến lệnh Persistent Write từ máy chủ khi hoàn thành thành công Persistent Write;

nhận, từ máy chủ, yêu cầu để gửi trạng thái cho nhiều lần Persistent Write cùng với các WID liên quan; và

nhận, bởi NVDIMM, một nhóm bao gồm hai hoặc nhiều lệnh Persistent Write có WID chung, với lệnh Persistent Write cuối cùng của nhóm có bit Persist được đặt bằng 1 và các lệnh Persistent Write còn lại có các bit Persist tương ứng được đặt bằng 0 và cung cấp chỉ báo hoàn thành Persistent Write cho lệnh Persistent Write cuối cùng.

2. Phương pháp theo điểm 1, trong đó phương pháp này còn bao gồm bước gửi, từ NVDIMM khi hoàn thành thành công lệnh Persistent Write, chỉ báo hoàn thành Persistent Write cùng với WID liên quan đến máy chủ.

3. Phương pháp theo điểm 2, trong đó bước gửi chỉ báo hoàn thành Persistent Write bao gồm gửi hai hoặc nhiều chỉ báo hoàn thành Persistent Write từ NVDIMM đến máy chủ theo thứ tự khác so với thứ tự trong đó hai hoặc nhiều lệnh Persistent Write tương ứng đã được nhận từ máy chủ.

4. Phương pháp theo điểm 1, trong đó phương pháp này còn bao gồm bước gửi gói trạng thái từ NVDIMM đến máy chủ, gói trạng thái bao gồm các WID cho các lệnh Persistent Write mà quá trình thực thi của nó đã được hoàn thành.

5. Phương pháp theo điểm 1, trong đó WID bao gồm mã định danh nhiều bit của Persistent Write và bit hợp lệ.

6. Phương pháp theo điểm 1, phương pháp còn bao gồm bước nhận, bởi NVDIMM, lệnh FLUSH từ máy chủ, trong đó lệnh FLUSH chỉ báo rằng tất cả lần ghi trước được đệm trong phương tiện khả biến sẽ được đẩy vào bộ nhớ bất biến hoặc bộ nhớ liên tục.

7. Phương pháp theo điểm 6, trong đó phương pháp này còn bao gồm bước gửi chỉ báo hoàn thành FLUSH khi hoàn thành việc thực thi lệnh FLUSH từ NVDIMM cho máy chủ.

8. Phương pháp theo điểm 1, trong đó phương pháp này còn bao gồm bước nhận, bởi NVDIMM, một hoặc nhiều lệnh Persistent Write, duy trì các trạng thái của một hoặc nhiều lệnh Persistent Write đã hoàn thành trong ánh xạ bit hoàn thành và các trạng thái của một hoặc nhiều lệnh Persistent Write chưa hoàn thành trong ánh xạ bit chưa hoàn thành, và khi có yêu cầu trạng thái từ máy chủ, cung cấp ánh xạ bit hoàn thành nếu không có lỗi không thể sửa chữa hoặc ánh xạ bit chưa hoàn thành nếu có lỗi không thể sửa chữa.

9. Phương pháp thực hiện các thao tác liên tục, phương pháp này bao gồm các bước:

gửi, từ máy chủ đến module nhớ hai hàng chân bất biến (non-volatile dual in-line memory module - NVDIMM) được tạo cấu hình để hỗ trợ các lần Persistent Write, lệnh Persistent Write và dữ liệu ghi liên quan;

cung cấp mã định danh ghi (write identification - WID) liên quan đến lệnh Persistent Write cho NVDIMM từ máy chủ khi hoàn thành thành công lệnh Persistent Write;

gửi, từ máy chủ đến NVDIMM, yêu cầu để gửi trạng thái của nhiều lệnh Persistent Write cùng với các WID liên quan; và

gửi từ máy chủ đến NVDIMM, nhóm gồm hai hoặc nhiều lệnh Persistent Write với WID chung, cùng với lệnh Persistent Write cuối cùng của nhóm có bit Persist được đặt bằng 1 và các lệnh Persistent Write còn lại có các bit Persist tương ứng được đặt bằng 0 và nhận từ NVDIMM, chỉ báo hoàn thành Persistent Write cho lệnh Persistent Write cuối cùng, trong đó lệnh Persistent Write chỉ báo cho NVDIMM thực hiện lệnh ghi liên tục dữ liệu ghi vào bộ nhớ bất biến.

10. Phương pháp theo điểm 9, phương pháp này còn bao gồm bước nhận tại máy chủ, chỉ báo hoàn thành Persistent Write cùng với WID liên quan từ NVDIMM khi hoàn thành thành công Persistent Write.

11. Phương pháp theo điểm 10, trong đó phương pháp này bao gồm bước nhận chỉ báo hoàn thành Persistent Write bao gồm việc nhận từ NVDIMM, hai hoặc nhiều chỉ báo hoàn

thành Persistent Write theo thứ tự khác với thứ tự trong đó hai hoặc nhiều lệnh Persistent Write tương ứng đã được gửi từ máy chủ đến NVDIMM.

12. Phương pháp theo điểm 11, trong đó phương pháp này còn bao gồm bước nhận gói trạng thái bởi máy chủ từ NVDIMM, gói trạng thái này bao gồm các WID cho các lệnh Persistent Write mà quá trình thực thi của nó đã được hoàn thành.

13. Phương pháp theo điểm 9, trong đó WID bao gồm mã định danh nhiều bit của Persistent Write và bit hợp lệ.

14. Phương pháp theo điểm 9, trong đó phương pháp này còn bao gồm bước gửi lệnh FLUSH từ máy chủ đến NVDIMM, trong đó lệnh FLUSH chỉ báo rằng tất cả các lần ghi trước được đệm trong phương tiện khả biến sẽ được đẩy vào bộ nhớ bất biến hoặc bộ nhớ liên tục bởi NVDIMM.

15. Phương pháp theo điểm 14, trong đó phương pháp này còn bao gồm bước nhận tại máy chủ, chỉ báo hoàn thành FLUSH khi NVDIMM hoàn thành quá trình thực thi lệnh FLUSH.

16. Thiết bị thực hiện các thao tác liên tục, thiết bị này bao gồm:

máy chủ, và

modun nhớ hai hàng chân bất biến (non-volatile dual in-line memory module - NVDIMM) được tạo cấu hình để hỗ trợ các lần Persistent Write và được tạo cấu hình để:

nhận lệnh Persistent Write và dữ liệu ghi liên quan từ máy chủ; và

thực hiện ghi liên tục dữ liệu ghi vào bộ nhớ bất biến trong NVDIMM dựa trên lệnh Persistent Write;

nhận mã định danh ghi (write identification - WID) liên quan đến lệnh Persistent Write từ máy chủ khi hoàn thành thành công lệnh Persistent Write;

nhận, từ máy chủ, yêu cầu để gửi trạng thái cho nhiều Persistent Write cùng với các WID liên quan; và

nhận nhóm bao gồm hai hoặc nhiều lệnh Persistent Write với WID chung, với lệnh Persistent Write cuối cùng của nhóm có bit Persist được đặt bằng 1 và các lệnh Persistent Write còn lại có các bit Persist tương ứng được đặt bằng 0 và cung cấp chỉ báo hoàn thành Persistent Write cho lệnh Persistent Write cuối cùng.

17. Thiết bị theo điểm 16, trong đó NVDIMM còn được tạo cấu hình để cung cấp, khi hoàn thành thành công Persistent Write, chỉ báo hoàn thành Persistent Write cùng với WID liên quan cho máy chủ.

18. Thiết bị theo điểm 17, trong đó NVDIMM còn được tạo cấu hình để cung cấp hai hoặc nhiều chỉ báo hoàn thành Persistent Write cho máy chủ theo thứ tự khác với thứ tự trong đó hai hoặc nhiều lệnh Persistent Write tương ứng đã được nhận từ máy chủ.

19. Thiết bị theo điểm 18, trong đó NVDIMM còn được tạo cấu hình để cung cấp gói trạng thái cho máy chủ, gói trạng thái này bao gồm các WID cho các lệnh Persistent Write mà việc thực thi của nó đã được hoàn thành.

20. Thiết bị theo điểm 16, trong đó WID bao gồm mã định danh nhiều bit của Persistent Write và bit hợp lệ.

21. Thiết bị theo điểm 16, trong đó NVDIMM còn được tạo cấu hình để nhận lệnh FLUSH từ máy chủ, trong đó lệnh FLUSH chỉ báo rằng tất cả các lần ghi trước được đệm trong phương tiện khả biến sẽ được đẩy vào bộ nhớ bất biến hoặc bộ nhớ liên tục.

22. Thiết bị theo điểm 21, trong đó NVDIMM còn được tạo cấu hình để cung cấp chỉ báo hoàn thành FLUSH khi hoàn thành quá trình thực thi lệnh FLUSH cho máy chủ.

23. Thiết bị theo điểm 16, trong đó NVDIMM còn được tạo cấu hình để nhận một hoặc nhiều lệnh Persistent Write, duy trì các trạng thái của một hoặc nhiều lệnh Persistent Write được hoàn thành trong ánh xạ bit hoàn thành và các trạng thái của một hoặc nhiều lệnh Persistent Write chưa hoàn thành trong ánh xạ bit chưa hoàn thành, và khi có yêu cầu về trạng thái từ máy chủ, cung cấp ánh xạ bit hoàn thành nếu không có lỗi không thể sửa chữa hoặc ánh xạ bit chưa hoàn thành nếu có lỗi không thể sửa chữa.

24. Thiết bị thực hiện các thao tác liên tục, thiết bị này bao gồm:

 máy chủ được tạo cấu hình để gửi lệnh Persistent Write và dữ liệu ghi liên quan; và
 modun nhớ hai hàng chân bất biến (non-volatile dual in-line module – NVDIMM) được tạo cấu hình để nhận lệnh Persistent Write và dữ liệu ghi liên quan được gửi từ máy chủ,

 trong đó lệnh Persistent Write chỉ báo cho NVDIMM thực hiện ghi liên tục dữ liệu ghi vào bộ nhớ bất biến của NVDIMM, và trong đó máy chủ còn được tạo cấu hình để cung cấp mã định danh ghi (write identification - WID) liên quan đến lệnh Persistent Write

cho NVDIMM và gửi tới NVDIMM khi hoàn thành thành công lệnh Persistent Write, yêu cầu để giữ trạng thái cho nhiều lần Persistent Write cùng với các WID liên quan, và gửi tới NVDIMM, nhóm bao gồm hai hoặc nhiều lệnh Persistent Write có WID chung, với lệnh Persistent Write cuối cùng của nhóm có bit Persist được đặt bằng 1 và các Persistent Write còn lại có các bit Persist tương ứng được đặt bằng 0, và nhận từ NVDIMM, chỉ báo hoàn thành Persistent Write cho Persistent Write cuối cùng.

25. Thiết bị theo điểm 24, trong đó máy chủ còn được tạo cấu hình để nhận chỉ báo hoàn thành Persistent Write cùng với WID liên quan từ NVDIMM khi hoàn thành thành công lệnh Persistent Write.

26. Thiết bị theo điểm 25, trong đó máy chủ còn được tạo cấu hình để nhận từ NVDIMM, hai hoặc nhiều chỉ báo hoàn thành Persistent Write theo thứ tự khác với thứ tự trong đó hai hoặc nhiều lệnh Persistent Write tương ứng được gửi cho NVDIMM.

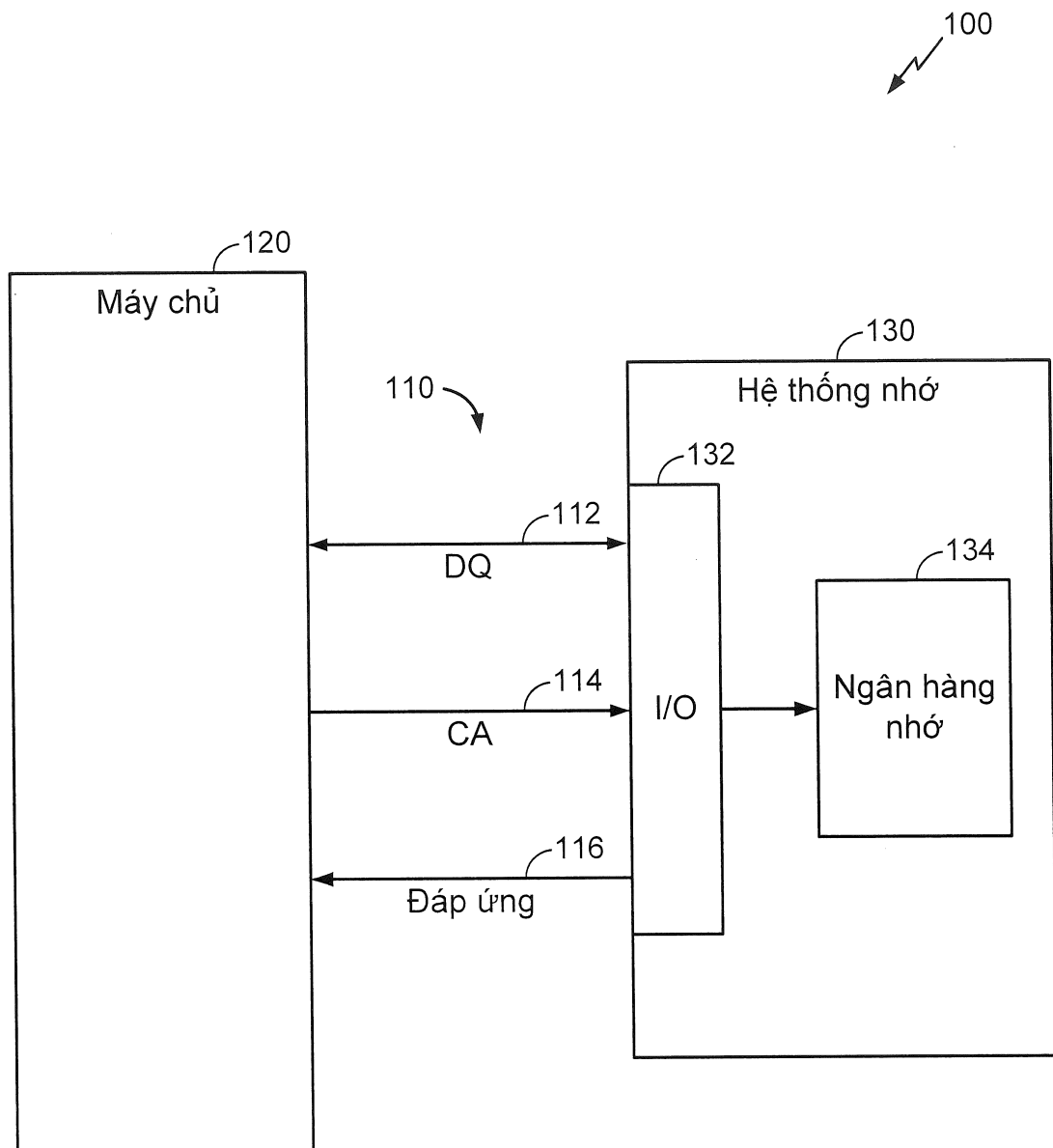
27. Thiết bị theo điểm 26, trong đó máy chủ còn được tạo cấu hình để nhận gói trạng thái từ NVDIMM, gói trạng thái này bao gồm các WID cho các lệnh Persistent Write mà quá trình thực thi của nó đã được hoàn thành.

28. Thiết bị theo điểm 27, trong đó WID bao gồm mã định danh nhiều bit của Persistent Write và bit hợp lệ.

29. Thiết bị theo điểm 24, trong đó máy chủ còn được tạo cấu hình để gửi lệnh FLUSH cho NVDIMM, trong đó lệnh FLUSH chỉ báo rằng tất cả các lần ghi trước được đệm trong phương tiện khả biến sẽ được đẩy vào bộ nhớ bất biến hoặc bộ nhớ liên tục bởi NVDIMM.

30. Thiết bị theo điểm 29, trong đó máy chủ còn được tạo cấu hình để nhận chỉ báo hoàn thành FLUSH khi NVDIMM hoàn thành việc thực thi lệnh FLUSH.

1/7

**Fig.1**

2/7

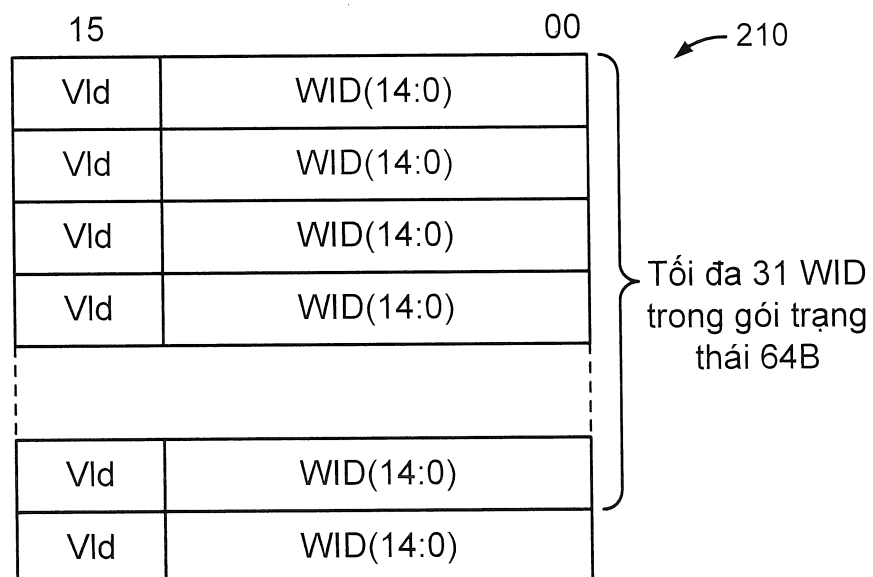
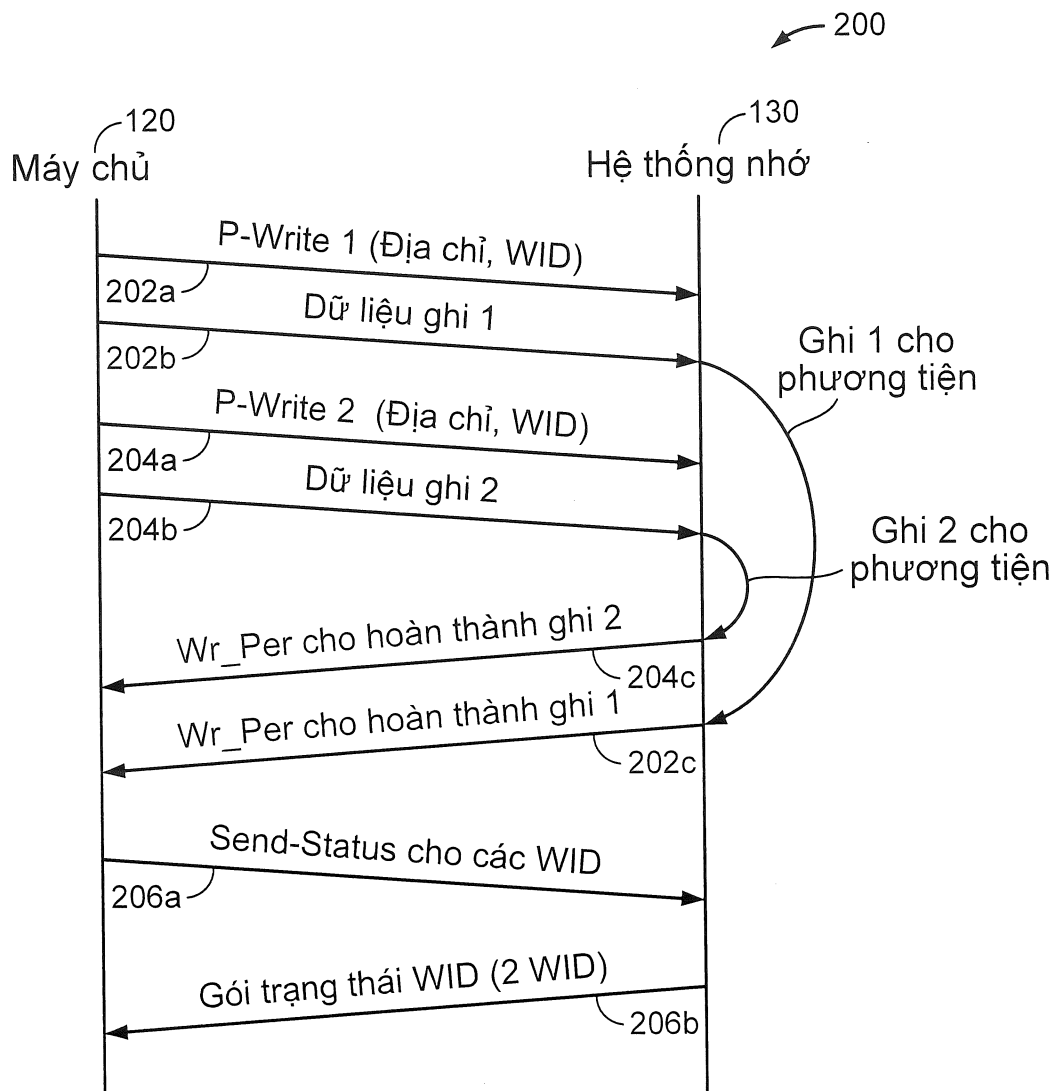


Fig.2A

3/7

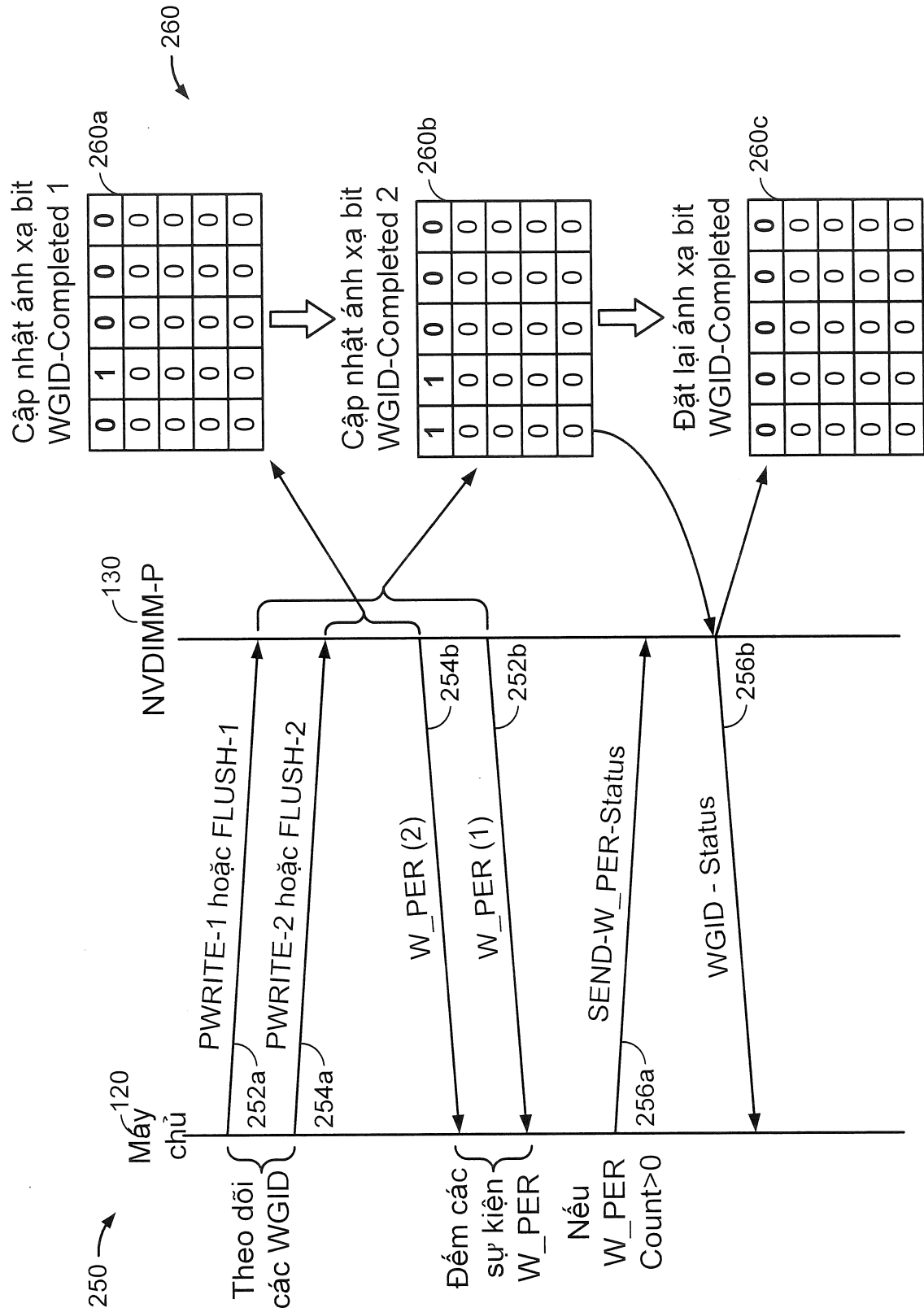


Fig.2B

4/7

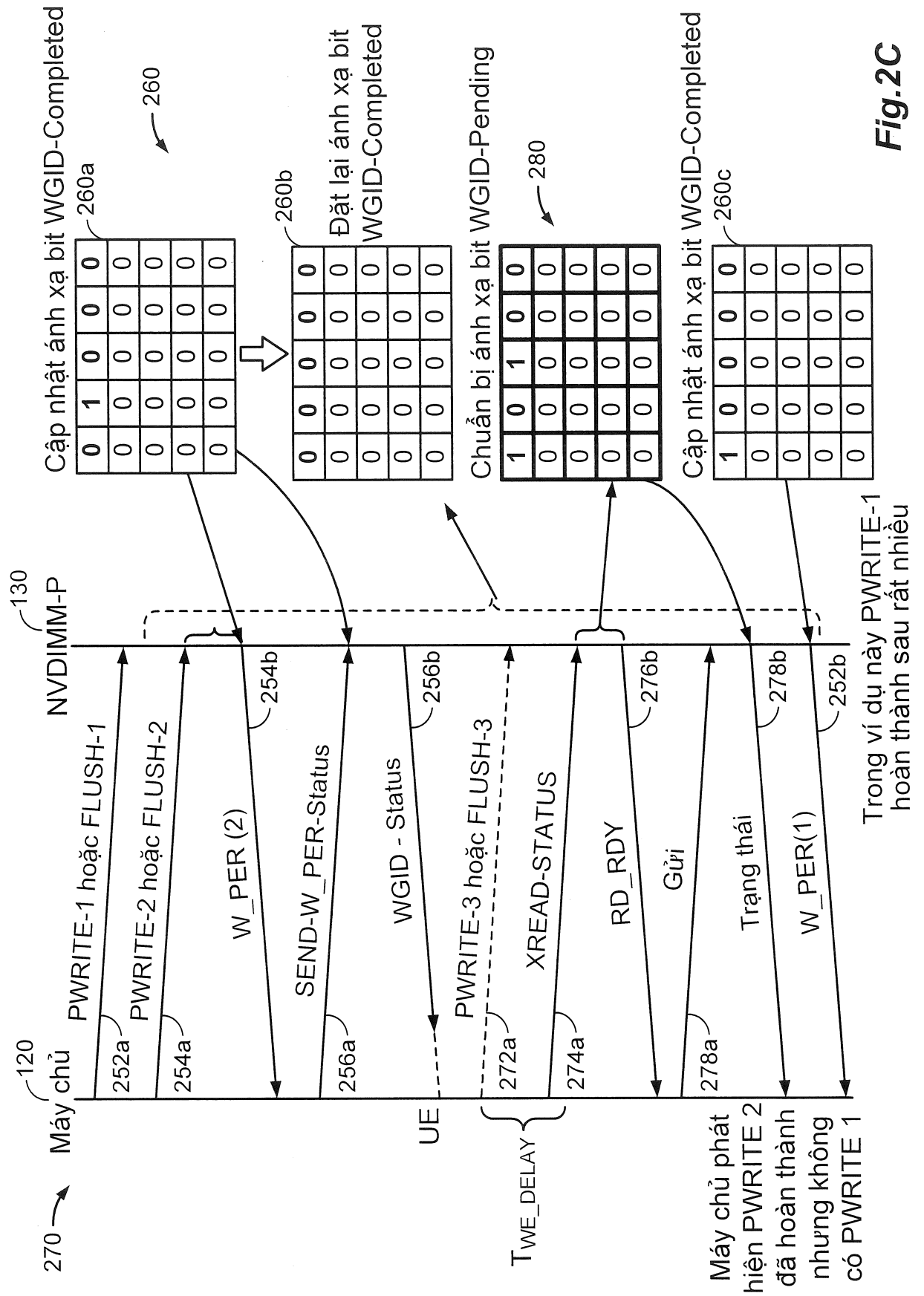


Fig.2C

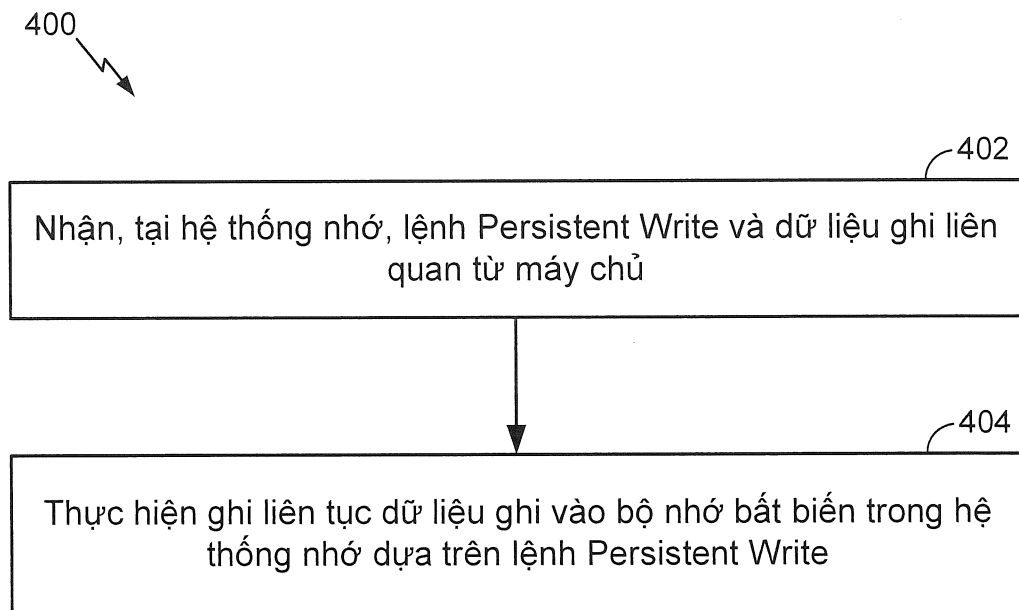
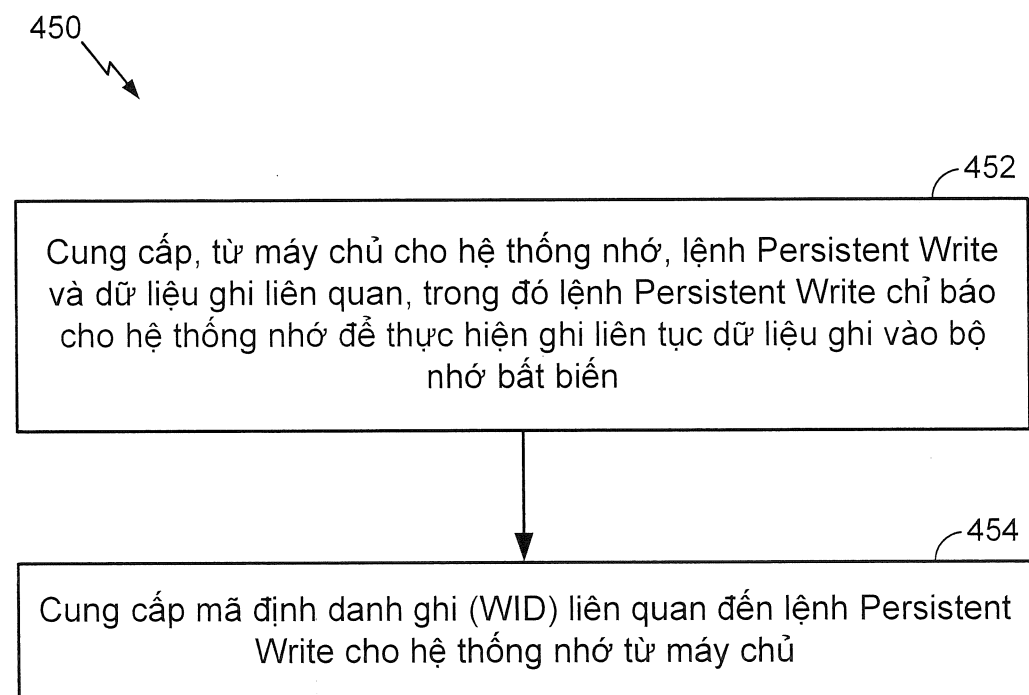
5/7

XPWRITE

CA0	CA1	CA2	CA3	CA4	CA5	CA6	CA7
H	L	L	H	L	Địa chỉ [39:37]		
Địa chỉ [36:29]							
Địa chỉ [28]	WID [9:3]						
WID [2:0]			Liên tục	RFU			

FIG.3

6/7

**Fig.4A****Fig.4B**

7/7

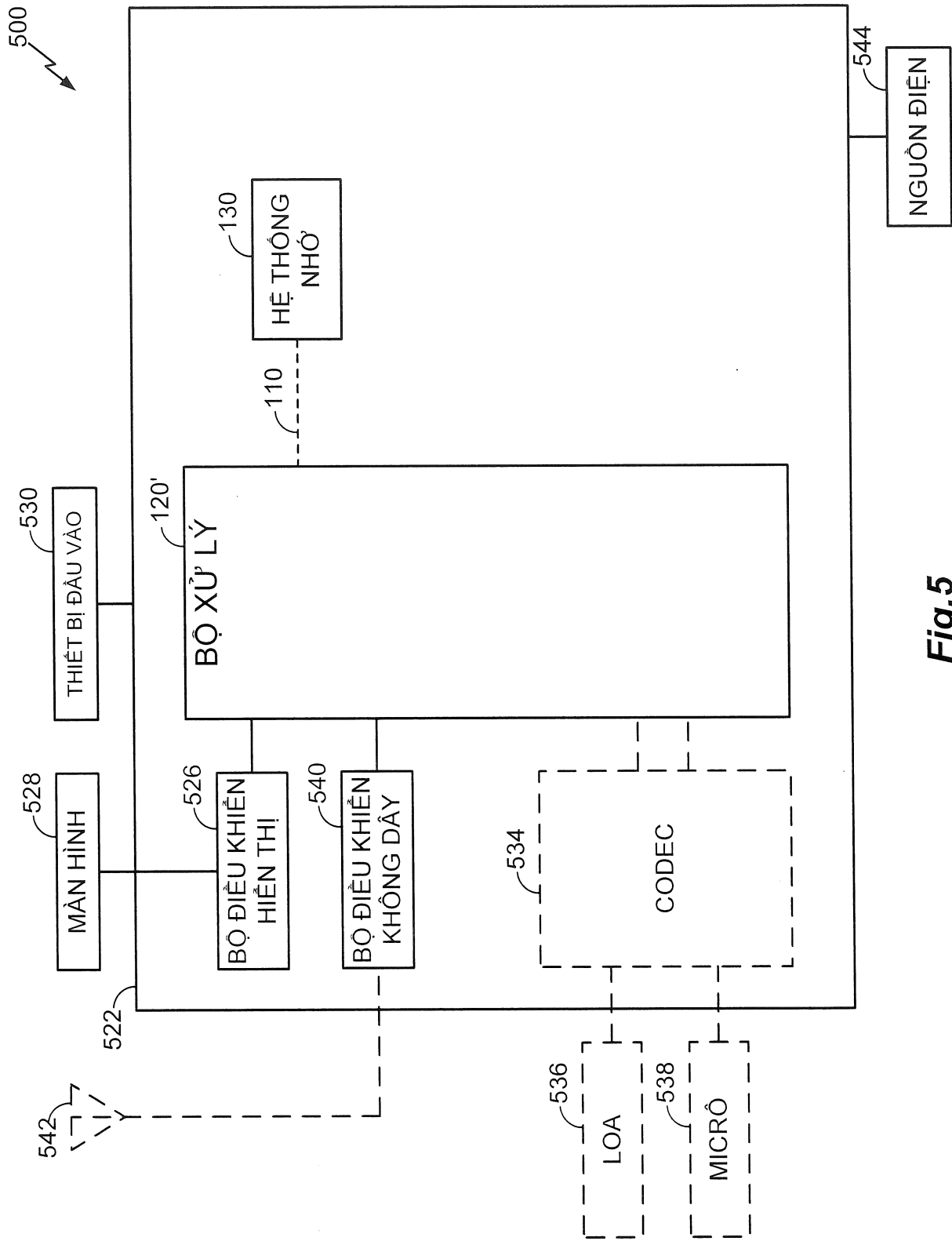


Fig.5