



(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)



1-0053827

(51)^{2022.01} G06F 3/06

(13) B

(21) 1-2023-02008

(22) 18/08/2021

(86) PCT/US2021/046575 18/08/2021

(87) WO 2022/072077 A1 07/04/2022

(30) 17/061,889 02/10/2020 US

(45) 25/11/2025 452

(43) 25/07/2023 424A

(73) QUALCOMM INCORPORATED (US)

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

(72) KUMAR, Vikrant (IN); CHANDRASEKAR, Karthik (IN).

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

(54) PHƯƠNG PHÁP VÀ HỆ THỐNG LẬP LỊCH CÁC GIAO DỊCH BỘ NHỚ TRUY
CẬP NGẪU NHIÊN ĐỘNG TRONG THIẾT BỊ ĐIỆN TOÁN

(21) 1-2023-02008

(57) Sáng chế đề cập đến phương pháp và hệ thống lập lịch các giao dịch bộ nhớ truy cập ngẫu nhiên động trong thiết bị điện toán. Các giao dịch bộ nhớ trong thiết bị điện toán có thể được lập lịch bằng cách thành lập các tập hợp con của tập hợp các giao dịch bộ nhớ tương ứng với các yêu cầu giao dịch bộ nhớ được hướng đến DRAM. Mỗi tập hợp con có thể gồm các giao dịch được nhận diện bởi cùng một kết hợp của hướng (đọc hoặc ghi) và nhóm DRAM giống nhau. Các giao dịch được lựa chọn để đưa vào mỗi tập hợp con có thể được xác định dựa trên hiệu quả. Một trong các tập hợp con có thể được lựa chọn dựa trên chỉ số được áp dụng cho mỗi tập hợp con và các giao dịch trong tập hợp con được lựa chọn có thể được gửi tới DRAM.

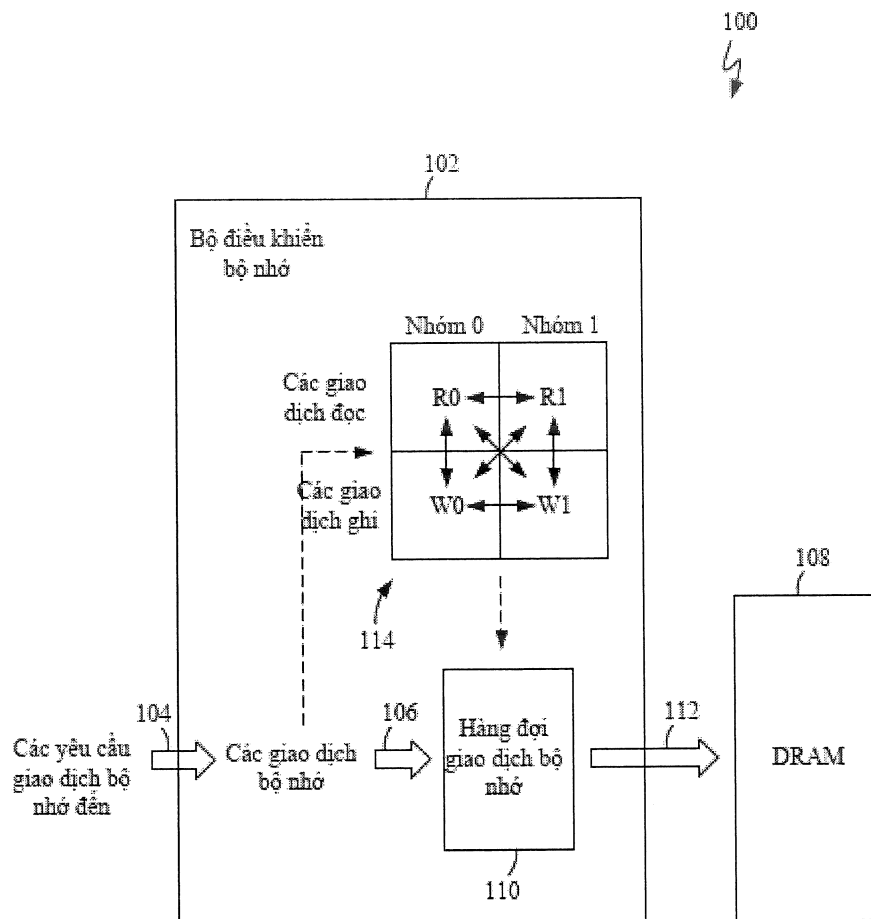


FIG.1

Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến truyền thông không dây, cụ thể là đến phương pháp và hệ thống lập lịch các giao dịch bộ nhớ truy cập ngẫu nhiên động trong thiết bị điện toán.

Tình trạng kỹ thuật của sáng chế

Các thiết bị điện toán di động (“Portable computing device - PCD”) đang trở thành nhu cầu thiết yếu đối với con người ở các cấp độ cá nhân và chuyên nghiệp. Các thiết bị này có thể gồm điện thoại di động, máy tính bảng, máy tính cầm tay, thiết bị hỗ trợ kỹ thuật số cầm tay (“portable digital assistant - PDA”), máy chơi game cầm tay, và các thiết bị điện tử cầm tay khác. PCD thường chứa các mạch tích hợp hoặc các hệ thống trên chip (“system-on-a-chip - SoC”) mà gồm nhiều thành phần được thiết kế để hoạt động cùng nhau nhằm cung cấp chức năng cho người dùng. Ví dụ, SoC có thể chứa số lượng bất kỳ các công cụ xử lý, chẳng hạn như đơn vị xử lý trung tâm (“central processing unit - CPU”) có nhiều lõi, đơn vị xử lý đồ họa (“graphical processing unit - GPU”), đơn vị xử lý neuron (“neural processing unit - NPU”), bộ xử lý tín hiệu số (“DSP”), bộ thu phát không dây (còn được gọi là môđem), v.v.. SoC có thể được ghép nối với các thành phần khác trong PCD, chẳng hạn như bộ nhớ hệ thống. Trong thiết bị điện toán, gồm nhưng không giới hạn ở các PCD, các giao dịch bộ nhớ có thể được thực hiện giữa các công cụ xử lý khác nhau và bộ nhớ hệ thống. Bộ nhớ hệ thống trong các PCD và các thiết bị điện toán khác thường bao gồm bộ nhớ truy cập ngẫu nhiên động (“dynamic random access memory - DRAM”).

Công cụ xử lý có thể bắt đầu ghi dữ liệu vào hoặc đọc dữ liệu từ bộ nhớ hệ thống bằng cách yêu cầu giao dịch ghi hoặc giao dịch đọc, tương ứng. Bộ điều khiển bộ nhớ có thể dịch các yêu cầu giao dịch như vậy từ công cụ xử lý thành giao dịch bộ nhớ bao gồm tập hợp các lệnh DRAM và dữ liệu được liên kết cũng như các địa chỉ bộ nhớ đích. Để đáp ứng các ràng buộc định thời của DRAM, bộ điều khiển bộ nhớ phải giãn cách các lệnh nhất định mà nó cung cấp cho chip DRAM theo các lượng thời gian hoặc các quãng thời gian được chỉ định. Hiệu ứng tích lũy của các quãng định thời như vậy góp phần vào độ trễ hoặc thời gian cần thiết để hoàn thành giao dịch bộ nhớ. Có mong muốn giảm thiểu độ trễ giao dịch bộ nhớ và cải thiện hiệu quả truy cập bộ nhớ, vì cả hai đều có khả năng tác động xấu đến hiệu suất của thiết bị điện toán, và cuối cùng là, trải nghiệm người dùng.

Bộ điều khiển bộ nhớ có thể xử lý nhiều yêu cầu giao dịch bộ nhớ mà được đưa ra bởi nhiều công cụ xử lý. Bộ điều khiển bộ nhớ có thể nhận được các yêu cầu giao dịch bộ nhớ bổ sung từ một hoặc nhiều công cụ xử lý trước khi hoàn thành các yêu cầu giao dịch mà bộ điều khiển bộ nhớ đã nhận trước đó. “Lập lịch” là thuật ngữ thường được sử dụng để chỉ các kỹ thuật được sử dụng nhằm giúp cân bằng các mục tiêu đối lập đó là đáp ứng các ràng buộc định thời của DRAM trong khi giảm thiểu độ trễ và tối đa hóa hiệu quả của DRAM. Nói một cách khái quát, quá trình lập lịch liên quan đến việc lựa chọn thứ tự hoặc mức độ ưu tiên mà trong đó bộ điều khiển bộ nhớ gửi các giao dịch tới DRAM. Các thuật toán lập lịch giao dịch bộ nhớ thông thường có xu hướng không linh hoạt (được định nghĩa tĩnh) và không thích ứng động với các đặc điểm của các yêu cầu giao dịch trong thời gian chạy.

Bản chất kỹ thuật của sáng chế

Các hệ thống, phương pháp, sản phẩm chương trình máy tính, và các phương án khác được bộc lộ để lập lịch các giao dịch bộ nhớ trong thiết bị điện toán. Trong các phương án làm ví dụ, thiết bị điện toán có thể gồm bộ điều khiển bộ nhớ được ghép nối với DRAM.

Phương pháp làm ví dụ để lập lịch các giao dịch DRAM trong thiết bị điện toán có thể gồm bước xác định nhiều giao dịch DRAM tương ứng với nhiều yêu cầu giao dịch nhận được từ một hoặc nhiều bộ xử lý. Phương pháp làm ví dụ có thể còn gồm bước thành lập các tập hợp con giao dịch DRAM. Phương pháp làm ví dụ cũng có thể còn gồm bước lựa chọn một trong các tập hợp con. Phương pháp làm ví dụ có thể còn gồm bước gửi tất cả các giao dịch trong tập hợp con được lựa chọn tới DRAM. Khi xác định nhiều giao dịch DRAM, mỗi giao dịch DRAM có thể được nhận diện bởi sự kết hợp của việc liệu giao dịch là đọc hay ghi và nhóm DRAM (DRAM rank) mà giao dịch được hướng tới. Mỗi tập hợp con có thể bao gồm một hoặc nhiều giao dịch DRAM được nhận diện bởi cùng một kết hợp của hướng và nhóm DRAM giống nhau.

Hệ thống làm ví dụ để lập lịch các giao dịch DRAM trong thiết bị điện toán có thể gồm hàng đợi giao dịch và logic lập lịch. Logic lập lịch có thể được tạo cấu hình để xác định nhiều giao dịch DRAM tương ứng với nhiều yêu cầu giao dịch nhận được từ một hoặc nhiều bộ xử lý. Logic lập lịch có thể còn được tạo cấu hình để thành lập các tập hợp con giao dịch DRAM. Logic lập lịch cũng có thể còn được tạo cấu hình để lựa chọn một

trong các tập hợp con. Logic lập lịch có thể còn được tạo cấu hình để gửi tất cả các giao dịch trong tập hợp con đã lựa chọn tới DRAM. Khi xác định các giao dịch DRAM, logic lập lịch có thể nhận diện mỗi giao dịch bởi sự kết hợp của việc liệu giao dịch đó là đọc hay ghi và nhóm DRAM mà giao dịch được hướng tới. Mỗi tập hợp con có thể bao gồm một hoặc nhiều giao dịch DRAM được nhận diện bởi cùng một kết hợp của hướng và nhóm DRAM giống nhau.

Một hệ thống làm ví dụ khác để lập lịch các giao dịch DRAM trong thiết bị điện toán có thể gồm phương tiện để xác định nhiều giao dịch DRAM tương ứng với nhiều yêu cầu giao dịch nhận được từ một hoặc nhiều bộ xử lý. Hệ thống làm ví dụ có thể còn gồm phương tiện để thành lập các tập hợp con giao dịch DRAM. Hệ thống làm ví dụ cũng có thể còn gồm phương tiện để lựa chọn một trong các tập hợp con. Hệ thống làm ví dụ có thể còn gồm phương tiện để gửi tất cả các giao dịch trong tập hợp con được lựa chọn tới DRAM. Phương tiện để xác định nhiều giao dịch DRAM trong hệ thống làm ví dụ có thể nhận diện mỗi giao dịch bộ nhớ bởi sự kết hợp của việc liệu giao dịch là đọc hay ghi và nhóm DRAM mà giao dịch được hướng tới. Mỗi tập hợp con có thể bao gồm một hoặc nhiều giao dịch DRAM được nhận diện bởi cùng một kết hợp của hướng và nhóm DRAM giống nhau.

Sản phẩm chương trình máy tính làm ví dụ để lập lịch các giao dịch DRAM trong thiết bị điện toán có thể bao gồm phương tiện bất biến đọc được bằng máy tính có các lệnh được lưu trữ trên đó dưới dạng các lệnh thực thi được bằng máy tính mà khi được thực thi bởi hệ thống xử lý của bộ điều khiển bộ nhớ thì tạo cấu hình bộ điều khiển bộ nhớ để: xác định nhiều giao dịch DRAM tương ứng với nhiều yêu cầu giao dịch nhận được từ một hoặc nhiều bộ xử lý; thành lập các tập hợp con giao dịch DRAM; lựa chọn một trong các tập hợp con; và gửi tất cả các giao dịch trong tập hợp con được lựa chọn tới DRAM. Trong sản phẩm chương trình máy tính làm ví dụ, giao dịch DRAM có thể được nhận diện bởi sự kết hợp của việc liệu giao dịch được đọc hay ghi và nhóm DRAM mà giao dịch được hướng tới. Mỗi tập hợp con có thể bao gồm một hoặc nhiều giao dịch DRAM được nhận diện bởi cùng một kết hợp của hướng và nhóm DRAM giống nhau.

Mô tả vắn tắt các hình vẽ

Trên các hình vẽ, các số tham chiếu giống nhau đề cập đến các phần giống nhau trong các hình chiếu khác nhau trừ khi có chỉ dẫn khác. Đối với các số tham chiếu có ký

hiệu ký tự chữ, chẳng hạn như “102A” hoặc “102B”, ký hiệu ký tự chữ có thể phân biệt hai phần hoặc phần tử giống nhau có trong cùng một hình vẽ. Các ký hiệu ký tự chữ cho các số tham chiếu có thể được bỏ qua khi dự định rằng số tham chiếu bao hàm tất cả các phần có cùng một chữ số tham chiếu trong tất cả các hình vẽ.

Fig.1 là sơ đồ khối minh họa thiết bị điện toán có bộ điều khiển bộ nhớ được ghép nối với DRAM, theo các phương án làm ví dụ.

Fig.2 là sơ đồ khối minh họa hệ thống để lập lịch các giao dịch DRAM trong thiết bị điện toán, theo các phương án làm ví dụ.

Fig.3 là lưu đồ minh họa phương pháp lập lịch các giao dịch DRAM trong thiết bị điện toán, theo các phương án làm ví dụ.

Fig.4 là lưu đồ minh họa phương pháp thành lập tập hợp con giao dịch, theo các phương án làm ví dụ.

Fig.5A là một phần của lưu đồ minh họa phương pháp loại bỏ các giao dịch không hiệu quả khỏi tập hợp con giao dịch, theo các phương án làm ví dụ.

Fig.5B là một phần khác của lưu đồ minh họa phương pháp loại bỏ các giao dịch không hiệu quả khỏi tập hợp con giao dịch, tiếp tục từ Fig.5A, theo các phương án làm ví dụ.

Fig.6 là lưu đồ quy trình minh họa phương pháp thêm các giao dịch tiếp theo sau khi tập hợp con đã bắt đầu được gửi tới DRAM, theo các phương án làm ví dụ.

Fig.7 là dòng thời gian minh họa các khía cạnh định thời của phương pháp trên Fig.6, theo các phương án làm ví dụ.

Fig.8 là sơ đồ khái niệm minh họa ví dụ về việc thành lập tập hợp con giao dịch bằng cách lựa chọn các giao dịch từ một số khối, theo các phương án làm ví dụ.

Fig.9 là lưu đồ minh họa phương pháp lựa chọn một trong các tập hợp con giao dịch, theo các phương án làm ví dụ.

Fig.10 là dòng thời gian minh họa khoảng thời gian chồng lấn giữa bước gửi giao dịch từ tập hợp con dẫn đầu và bước gửi giao dịch từ tập hợp con theo sau, theo các phương án làm ví dụ.

Fig.11 là lưu đồ minh họa một khía cạnh của phương pháp gửi tập hợp con được lựa chọn tới DRAM, theo các phương án làm ví dụ.

Fig.12 là lưu đồ minh họa một khía cạnh khác của phương pháp gửi tập hợp con được lựa chọn tới DRAM, theo các phương án làm ví dụ.

Fig.13 là dòng thời gian minh họa các khía cạnh định thời của phương pháp trên Fig.12, theo các phương án làm ví dụ.

Fig.14 là sơ đồ khối của thiết bị điện toán, theo các phương án làm ví dụ.

Mô tả chi tiết sáng chế

Từ “làm ví dụ” được sử dụng ở đây có nghĩa là “đóng vai trò là ví dụ, trường hợp, hoặc minh họa”. Từ “minh họa” ở đây có thể được sử dụng đồng nghĩa với “làm ví dụ”. Bất kỳ khía cạnh nào được mô tả ở đây là “làm ví dụ” không nhất thiết phải được hiểu là được ưu tiên hoặc có lợi hơn các khía cạnh khác.

Như được minh họa ở dạng khái niệm trên Fig.1, trong thiết bị điện toán 100, có thể thuộc bất kỳ loại nào, bộ điều khiển bộ nhớ 102 có thể, trong số các chức năng khác, dịch các yêu cầu giao dịch bộ nhớ đến 104 từ một hoặc nhiều công cụ xử lý (không được thể hiện) thành các giao dịch bộ nhớ tương ứng 106. Yêu cầu giao dịch bộ nhớ 104 có thể là, ví dụ, yêu cầu đọc, tức là, yêu cầu đọc dữ liệu từ DRAM 108, hoặc yêu cầu ghi, tức là yêu cầu ghi dữ liệu vào DRAM 108. Giao dịch 106 là tập hợp các phần tử dữ liệu mà có thể gồm một hoặc nhiều lệnh được hướng đến DRAM 108, các từ dữ liệu được ghi vào DRAM 108, và các địa chỉ đích trong DRAM 108.

Một chức năng khác của bộ điều khiển bộ nhớ 102 là lập lịch các giao dịch 106. Lập lịch giao dịch là phương pháp hoặc quá trình mà bộ điều khiển bộ nhớ 102 xác định và theo cách khác điều khiển việc định thời mà theo đó các giao dịch 106 (tức là, các tập hợp của các lệnh, dữ liệu, và các địa chỉ) được gửi đến DRAM 108. Một số khía cạnh của bước định thời có thể liên quan đến các giao dịch 106 khác, chẳng hạn như thứ tự hoặc mức độ ưu tiên mà các giao dịch 106 được gửi đến DRAM 108. Các khía cạnh khác của bước định thời có thể liên quan đến các ràng buộc định thời được áp đặt bởi các khả năng của DRAM 108. Bộ điều khiển bộ nhớ 102 có thể sắp đặt các giao dịch 106 bằng cách sử dụng hàng đợi giao dịch 110 (còn được gọi là hàng đợi lệnh) và điều khiển thời gian mà

các lệnh DRAM cấu thành 112 của các giao dịch 106 được gửi từ hàng đợi giao dịch 110 đến DRAM 108.

Quá trình lập lịch giao dịch có thể dựa trên kỹ thuật được đề cập trong sáng chế là “lập lịch giao dịch thích ứng”. Theo phương án làm ví dụ được thể hiện trên Fig.1, DRAM 108 có thể gồm hai nhóm. Trong các phương án DRAM nhóm kép như vậy, quá trình lập lịch tập hợp con thích ứng có thể được gọi là “lập lịch góc phần tư thích ứng”, dựa trên cấu trúc góc phần tư 114. Trên Fig.1 cấu trúc góc phần tư 114 được thể hiện ở dạng khái niệm; cấu trúc góc phần tư 114 có thể được thể hiện trong logic điều khiển (không được thể hiện riêng trên Fig.1) của bộ điều khiển bộ nhớ 102 theo bất kỳ cách nào. Ví dụ, cấu trúc góc phần tư 114 có thể liên quan đến một hoặc nhiều cấu trúc dữ liệu hoặc máy trạng thái hữu hạn. Trong ví dụ được thể hiện trên Fig.1, cấu trúc góc phần tư 114 được tổ chức dạng khái niệm dưới dạng lưới hoặc bảng trong đó hai hàng tương ứng với hướng của giao dịch (tức là, đọc hoặc ghi), và hai cột tương ứng với hai nhóm của DRAM 108. Trong quá trình lập lịch góc phần tư thích ứng, các giao dịch có thể được nhóm lại hoặc được nhận diện theo cấu trúc góc phần tư 114 là thuộc về hoặc được liên kết với một trong bốn nhóm hoặc góc phần tư: “R0,” đề cập đến các giao dịch đọc (“R”) từ nhóm thứ nhất (nhóm “0”) của DRAM 108; “R1,” đề cập đến giao dịch đọc từ nhóm thứ hai (nhóm “1”) của DRAM 108; “W0,” đề cập đến giao dịch ghi (“W”) vào nhóm thứ nhất của DRAM 108; và “W1”, đề cập đến giao dịch ghi vào nhóm thứ hai của DRAM 108.

Tổng quát hơn, trong quá trình lập lịch tập hợp con thích ứng, các giao dịch có thể được nhóm hoặc được nhận diện bởi sự kết hợp của hướng (tức là, đọc hoặc ghi) và nhóm DRAM. Các hệ thống, các phương pháp và các phương án khác của quá trình lập lịch tập hợp con thích ứng được mô tả ở đây có thể được áp dụng cho DRAM có số lượng (N) nhóm bất kỳ. Mặc dù không được thể hiện nhằm mục đích rõ ràng, nhưng các giao dịch bộ nhớ lập lịch thích ứng được hướng đến DRAM có N nhóm có thể dựa trên cấu trúc được tổ chức dạng khái niệm dưới dạng lưới hoặc bảng có hai hàng và N cột (hoặc, theo cách khác, hai cột và N hàng), xác định $2N$ tập hợp con đơn trị.

Theo cách được mô tả dưới đây đối với các phương án làm ví dụ, quá trình lập lịch tập hợp con thích ứng điều chỉnh một cách linh hoạt (tức là, theo thời gian chạy) quá trình lập lịch bộ nhớ được đề cập ở trên dựa trên tập hợp các giao dịch trong một hoặc nhiều tập hợp con. Ví dụ, theo khía cạnh của quá trình lập lịch góc phần tư thích ứng được chỉ ra

trên Fig.1 bằng các mũi tên hai đầu giữa mỗi cặp góc phần tư trong cấu trúc góc phần tư 114, thứ tự mà các góc phần tư được lựa chọn để gửi các giao dịch 106 của chúng đến DRAM 108 có thể được điều chỉnh động. Theo một khía cạnh khác của quá trình lập lịch góc phần tư thích ứng, các giao dịch 106 trong mỗi tập hợp con có thể được lựa chọn để gửi tới DRAM 108. Theo một khía cạnh khác của quá trình lập lịch tập hợp con thích ứng, thứ tự mà các giao dịch 106 trong tập hợp con đã lựa chọn được gửi đến DRAM 108 có thể được điều chỉnh động. Các khía cạnh khác nữa của quá trình lập lịch tập hợp con thích ứng được mô tả dưới đây.

Như được minh họa trên Fig.2, theo phương án minh họa hoặc làm ví dụ, bộ điều khiển bộ nhớ 202 có thể nhận các yêu cầu giao dịch bộ nhớ từ một hoặc nhiều công cụ xử lý, chẳng hạn như đơn vị xử lý trung tâm (“central processing unit - CPU”) 204. Các yêu cầu giao dịch bộ nhớ có thể bao gồm các yêu cầu đọc và các yêu cầu ghi được hướng đến, ví dụ, DRAM 206 nhóm kép. DRAM 206 có thể là, ví dụ, DRAM đồng bộ tốc độ dữ liệu kép (“double-data rate synchronous DRAM - DDR-SDRAM”). Bộ điều khiển bộ nhớ 202 có thể dịch các yêu cầu giao dịch bộ nhớ thành các giao dịch DRAM (đôi khi được viết tắt là các “txn”) theo cách thông thường. Vì chức năng thông thường này của bộ điều khiển bộ nhớ 202 và các chức năng liên quan được hiểu rõ bởi người có hiểu biết trung bình trong lĩnh vực kỹ thuật, nên các khía cạnh như vậy của bộ điều khiển bộ nhớ 202 không được mô tả ở đây.

DRAM 206 có thể có cấu trúc thông thường và hoạt động theo cách thông thường. Mặc dù cấu trúc và hoạt động như vậy được hiểu rõ bởi người có hiểu biết trung bình trong lĩnh vực kỹ thuật, phần mô tả ngắn gọn sau đây được cung cấp làm tình trạng kỹ thuật. Thuật ngữ “nhóm kép” có nghĩa là DRAM 206 có thể gồm hai nhóm 208, mà có thể được gọi là nhóm_0 và nhóm_1. Vì hai nhóm 208 giống hệt nhau, nên mô tả sau đây áp dụng cho mỗi nhóm 208. Nhóm 208 bao gồm hai hoặc nhiều khối 210. Ví dụ, mỗi nhóm 208 có thể gồm tám khối 210. Mỗi khối 210 được tổ chức thành một mảng hai chiều của các vị trí lưu trữ hoặc các ô được đánh chỉ số theo các hàng và các cột, trong đó hàng làm ví dụ và cột làm ví dụ được đánh dấu bằng dấu gạch chéo trên Fig.2 nhằm mục đích minh họa. Mặc dù DRAM 206 nhóm kép được mô tả ở đây nhằm mục đích rõ ràng, nên hiểu rằng trong các phương án khác, số lượng (N) nhóm trong DRAM như vậy có thể là một số bất kỳ lớn hơn hoặc bằng hai.

Mỗi giao dịch có thể gồm một hoặc nhiều lệnh 212, địa chỉ đích 214, và dữ liệu 216. Các lệnh 212 có thể gồm các lệnh đọc và các lệnh ghi, trong số các lệnh khác. Các địa chỉ đích 214 có thể gồm các địa chỉ hàng, các địa chỉ cột, và các địa chỉ khối. Mặc dù không được thể hiện nhằm mục đích rõ ràng, nhưng để đáp lại địa chỉ nhóm (ví dụ, bit lựa chọn chip được gồm trong lệnh đọc hoặc ghi) được cung cấp bởi bộ điều khiển bộ nhớ, logic giải mã địa chỉ nhóm có thể lựa chọn một trong các nhóm 208. Tương tự như vậy, mặc dù không được thể hiện nhằm mục đích rõ ràng, để đáp lại địa chỉ khối được cung cấp bởi bộ điều khiển bộ nhớ, logic giải mã địa chỉ khối có thể lựa chọn một trong các khối 210 ở nhóm được lựa chọn 208. Để đáp lại địa chỉ hàng được cung cấp bởi bộ điều khiển bộ nhớ, bộ giải mã địa chỉ hàng 218 có thể lựa chọn một trong các hàng trong khối được lựa chọn 210 trong nhóm được lựa chọn 208. Tương tự, để đáp lại địa chỉ cột được cung cấp bởi bộ điều khiển bộ nhớ, bộ giải mã địa chỉ cột 220 có thể lựa chọn một trong các cột trong khối được lựa chọn 210 trong nhóm được lựa chọn 208. Dữ liệu 216 có thể gồm dữ liệu (gửi đi hoặc “đọc”) mà DRAM 206 gửi đến bộ điều khiển bộ nhớ 202 và dữ liệu (đến hoặc “ghi”) mà DRAM 206 nhận được từ bộ điều khiển bộ nhớ 202. Các lệnh 212, các địa chỉ 214, v.v., được thể hiện riêng biệt và ở mức khái niệm trên Fig.2 nhằm mục đích rõ ràng. Ở cấp độ vật lý, bus bộ nhớ (không được thể hiện) giữa bộ điều khiển bộ nhớ 202 và DRAM 206 có thể ghép kênh hoặc sử dụng lại một số đường tín hiệu vật lý giống nhau cho các loại tín hiệu khác nhau tại các thời điểm khác nhau, theo cách hiểu của một người có hiểu biết trung bình trong lĩnh vực kỹ thuật.

Mỗi nhóm 208 có thể có chốt đọc 222 để đệm dữ liệu đọc, và chốt ghi 224 để đệm dữ liệu ghi. Mỗi nhóm 208 cũng có thể có logic đầu vào/đầu ra (“input/output - I/O”) 226 được tạo cấu hình để hướng dữ liệu đọc và ghi từ và đến các vị trí bộ nhớ được lựa chọn.

Mỗi khối 210 có thể có vùng đệm hàng 228. Vùng đệm hàng 228 lưu trữ các nội dung của hàng được lựa chọn (đôi khi được gọi là “trang”). Hàng phải được lựa chọn hoặc “được mở” trước khi nó có thể được ghi vào hoặc được đọc từ đó. DRAM 206 mở hàng, tức là, lưu trữ các nội dung của hàng đó trong vùng đệm hàng 228, để đáp lại lệnh kích hoạt (“activate - ACT”). Địa chỉ hàng có thể được cung cấp với lệnh ACT. Sau khi hàng được kích hoạt hoặc được mở, DRAM 206 có thể đọc từ hoặc ghi vào bất kỳ số lượng cột nào trong vùng đệm hàng 228 để đáp lại các lệnh đọc hoặc ghi, còn được gọi là các lệnh lựa chọn địa chỉ cột (“column address select - CAS”). Sau lệnh đọc hoặc ghi, dữ liệu được truyền nối tiếp giữa bộ điều khiển bộ nhớ 202 và DRAM 206 theo các đơn vị được gọi là

"cụm (burst)", ví dụ, có thể là tám bit trên mỗi đường tín hiệu dữ liệu. Hàng phải được khôi phục hoặc "được đóng" sau khi ghi vào hoặc đọc từ vùng đệm hàng 228. DRAM 206 đóng hàng, để đáp lại lệnh nạp trước ("pre-charge - PRE"). Trong quá trình dịch yêu cầu đọc hoặc yêu cầu ghi, bộ điều khiển bộ nhớ 202 xác định trình tự và sự định thời của một hoặc nhiều lệnh cần thiết để thực hiện yêu cầu. Bộ điều khiển bộ nhớ 202 cũng thực hiện các thao tác làm mới định kỳ trên tất cả các hàng trong tất cả các khối 210 của cả hai nhóm 208 để duy trì tính toàn vẹn dữ liệu, và có các lệnh được liên kết với các thao tác làm mới đó.

Bộ điều khiển bộ nhớ 202 có thể gồm bộ lập lịch giao dịch 230 mà điều khiển các phương pháp lập lịch được mô tả ở đây. Bộ lập lịch giao dịch 230 có thể gồm logic lập lịch 232 và hàng đợi giao dịch 234. Như được mô tả dưới đây đối với các phương pháp làm ví dụ, logic lập lịch 232 của bộ điều khiển bộ nhớ 202 có thể được tạo cấu hình để thành lập các giao dịch bộ nhớ thành các tập hợp con góc phần tư 236 sử dụng cấu trúc góc phần tư 114 được mô tả ở trên (Fig.1). Tổng quát hơn, đối với DRAM có N nhóm (trong đó N là số nguyên lớn hơn hoặc bằng một), logic lập lịch 232 có thể được tạo cấu hình để thành lập các giao dịch bộ nhớ thành 2N tập hợp con. Mỗi tập hợp con thứ nhất đến thứ N có thể bao gồm các giao dịch được nhận diện là đọc được hướng đến các tập hợp tương ứng trong nhóm thứ nhất đến thứ N, và mỗi tập hợp con thứ (N+1) đến thứ 2N có thể bao gồm các giao dịch được nhận diện là ghi được hướng đến các tập hợp tương ứng trong nhóm thứ nhất đến thứ N. Mặc dù không được thể hiện nhằm mục đích rõ ràng, logic lập lịch 232 có thể bao gồm hệ thống xử lý (ví dụ, kết hợp của một hoặc nhiều bộ xử lý và bộ nhớ được liên kết), máy trạng thái hữu hạn, hoặc logic khác có thể tạo cấu hình được để thực hiện hoặc nếu không thì điều khiển các phương pháp được mô tả ở đây. Trong các phương án trong đó logic lập lịch 232 bao gồm hệ thống xử lý, hệ thống xử lý có thể được tạo cấu hình bằng cách thực thi các lệnh firmware.

Bộ điều khiển bộ nhớ 202 có thể phân loại hoặc xếp ngăn mỗi giao dịch liên quan đến hiệu quả (ví dụ, độ trễ hoặc băng thông) mà giao dịch có thể được hoàn thành. Bộ điều khiển bộ nhớ 202 có thể nhận diện mỗi giao dịch là "thành công (hit)", "không thành công (miss)" hoặc "xung đột (conflict)" trên cơ sở trạng thái hiện tại của các khối DRAM. Hit là giao dịch đọc hoặc ghi vào hàng (trong khối 210) mà bộ điều khiển bộ nhớ 202 xác định là mở tại thời điểm bộ điều khiển bộ nhớ 202 dịch yêu cầu giao dịch. Miss là giao dịch đọc hoặc ghi vào hàng (trong khối 210) mà bộ điều khiển bộ nhớ 202 xác định được đóng hoặc

nạp trước tại thời điểm bộ điều khiển bộ nhớ 202 dịch yêu cầu giao dịch. Nếu bộ điều khiển bộ nhớ 202 nhận diện giao dịch đọc hoặc ghi là giao dịch miss, thì bộ điều khiển bộ nhớ 202 trước tiên phải mở hàng trước khi đọc từ hoặc ghi vào hàng đó. Conflict là giao dịch đọc hoặc ghi vào hàng (trong khối 210) mà bộ điều khiển bộ nhớ 202 xác định là đóng tại thời điểm bộ điều khiển bộ nhớ 202 dịch yêu cầu giao dịch, trong khi bộ điều khiển bộ nhớ 202 xác định rằng một hàng khác có cùng nhóm và khối (mà giao dịch bộ nhớ không được định hướng đến) đang mở tại thời điểm đó. Nếu bộ điều khiển bộ nhớ 202 nhận diện giao dịch đọc hoặc ghi là giao dịch conflict, thì bộ điều khiển bộ nhớ 202 trước tiên phải nạp trước (tức là, đóng) hàng mở trong khối đó, sau đó mở hàng mà giao dịch được hướng đến trước khi đọc từ hoặc ghi vào hàng đó. Để cho ngắn gọn trong phần mô tả sau đây, giao dịch được nhận diện là giao dịch hit có thể được gọi tắt là "giao dịch hit", giao dịch được nhận diện là giao dịch miss có thể được gọi tắt là "giao dịch miss" và giao dịch được nhận diện là giao dịch conflict có thể được gọi tắt là "giao dịch conflict".

Việc hoàn thành giao dịch phát sinh các độ trễ hoặc chi phí thời gian khác nhau, có thể thay đổi tùy thuộc vào việc liệu giao dịch là giao dịch hit, giao dịch miss, hay giao dịch conflict, liệu giao dịch có liên quan đến việc truyền dữ liệu theo hướng ngược lại với giao dịch trước đó hay không, liệu giao dịch có liên quan đến việc truyền dữ liệu trong nhóm khác với giao dịch trước đó hay không, và các yếu tố khác. Các độ trễ này thường được chỉ định bởi nhà sản xuất DRAM 206. Ví dụ về các độ trễ thường được chỉ định gồm: tRCD (tức là độ trễ hàng-cột), là số lượng các chu kỳ xung nhịp tối thiểu phải trôi qua giữa bước gửi lệnh ACT tới DRAM 206 để mở hàng và bước gửi lệnh CAS tới DRAM 206 để lựa chọn cột trên hàng đang mở; tRP (tức là, đọc sau khi nạp trước), là số lượng các chu kỳ xung nhịp tối thiểu phải trôi qua giữa bước gửi lệnh PRE tới DRAM 206 để đóng hàng và bước gửi lệnh ACT tới DRAM 206 để mở một hàng khác trên cùng một khối; tRRD (tức là, độ trễ đọc-đọc), là số lượng các chu kỳ xung nhịp tối thiểu phải trôi qua giữa các lần kích hoạt của hai hàng trong các khối khác nhau trong cùng một nhóm; tRD2PRE, là ràng buộc định thời đại diện biểu thị số lượng các chu kỳ xung nhịp tối thiểu phải trôi qua giữa bước gửi lệnh đọc (CAS) và gửi lệnh PRE; và tWR2PRE, là ràng buộc định thời đại diện biểu thị số lượng các chu kỳ xung nhịp tối thiểu phải trôi qua giữa bước gửi lệnh ghi (CAS) và bước gửi lệnh PRE.

Như được minh họa ở dạng lưu đồ trên Fig.3, phương pháp 300 để lập lịch giao dịch DRAM trong thiết bị điện toán có thể gồm các bước hoặc các hành động được mô tả

dưới đây liên quan đến các khối 302-308. Ví dụ, phương pháp 300 có thể được thực hiện hoặc nếu không thì được điều khiển bởi, bộ điều khiển bộ nhớ 202 được mô tả ở trên (Fig.2). Bộ điều khiển bộ nhớ 202 hoặc các phần của chúng (ví dụ, logic lập lịch 232, hàng đợi giao dịch 234, v.v.) có thể là các ví dụ về phương tiện để thực hiện các chức năng được mô tả dưới đây liên quan đến phương pháp 300.

Như được chỉ ra bởi khối 302, phương pháp 300 có thể gồm bước xác định nhiều giao dịch DRAM tương ứng với nhiều yêu cầu giao dịch nhận được từ một hoặc nhiều bộ xử lý. Bước xác định các giao dịch DRAM tương ứng với các yêu cầu giao dịch có thể được thực hiện theo cách thông thường, như được hiểu bởi một người có hiểu biết trung bình trong lĩnh vực kỹ thuật. Ngoài ra, mỗi giao dịch có thể được nhận diện bởi sự kết hợp của hướng (tức là, đọc hoặc ghi) và nhóm trong số N nhóm của DRAM mà giao dịch được hướng đến. Ví dụ, theo phương án DRAM nhóm kép như được mô tả ở trên liên quan đến Fig.2, mỗi giao dịch có thể được nhận diện là đọc từ nhóm thứ nhất (“R0”), đọc từ nhóm thứ hai (“R1”), ghi vào nhóm thứ nhất (“W0”), hoặc ghi vào nhóm thứ hai (“W1”). Về mặt khái niệm, mỗi giao dịch có thể được mô tả là “trong” hoặc “được liên kết với” một trong bốn góc phần tư, tùy thuộc vào hướng (tức là, đọc hoặc ghi) và nhóm của giao dịch.

Bước xác định nhiều giao dịch DRAM có thể còn gồm nhận diện mỗi giao dịch là giao dịch hit, giao dịch miss, giao dịch conflict, hoặc giao dịch hit-dưới-miss/conflict. “Giao dịch hit-dưới-miss/conflict” có nghĩa là một giao dịch ban đầu là giao dịch miss nhưng sẽ trở thành giao dịch hit ngay khi một giao dịch miss khác được hướng đến cùng một hàng trong cùng một khối đã kích hoạt hàng đó, hoặc một giao dịch ban đầu là giao dịch conflict nhưng sẽ trở thành giao dịch hit như ngay sau khi một giao dịch conflict khác được hướng đến cùng một hàng trong cùng một khối đã kích hoạt hàng đó. Các giao dịch được hướng đến DRAM có thể gồm số lượng giao dịch hit bất kỳ (tức là, không hoặc nhiều hơn), số lượng giao dịch miss bất kỳ, số lượng giao dịch conflict bất kỳ, và số lượng giao dịch hit-dưới-miss/conflict bất kỳ.

Như được chỉ ra bởi khối 304, phương pháp 300 có thể gồm bước thành lập các giao dịch DRAM thành nhiều tập hợp con. Mỗi tập hợp con có thể bao gồm các giao dịch DRAM mà được nhận diện bởi cùng một kết hợp của hướng (tức là, đọc hoặc ghi) và nhóm DRAM mà chúng được hướng tới. Trong các phương án trong đó DRAM có N nhóm, mỗi tập hợp con thứ nhất đến thứ N có thể bao gồm các giao dịch được nhận diện là đọc được

hướng đến các tập hợp tương ứng trong nhóm thứ nhất đến thứ N , và mỗi tập hợp con thứ $(N+1)$ đến thứ $2N$ có thể bao gồm các giao dịch được nhận diện là ghi được hướng đến các tập hợp tương ứng trong nhóm thứ nhất đến thứ N . Ví dụ, trong các phương án DRAM nhóm kép (tức là, $N=2$), bốn tập hợp con có thể bao gồm tập hợp con góc phần tư thứ nhất gồm một số hoặc tất cả các giao dịch được nhận diện là đọc từ nhóm thứ nhất (tức là, tập hợp con giao dịch trong R_0), tập hợp con góc phần tư thứ hai gồm một số hoặc tất cả các giao dịch được nhận diện là đọc từ nhóm thứ hai (tức là, tập hợp con giao dịch trong R_1), tập hợp con góc phần tư thứ ba gồm một số hoặc tất cả các giao dịch được nhận diện là ghi vào nhóm thứ nhất (tức là, tập hợp con giao dịch trong W_0), và tập hợp con góc phần tư thứ tư gồm một số hoặc tất cả các giao dịch được nhận diện là ghi vào nhóm thứ hai (tức là, tập hợp con giao dịch trong W_1).

Tham chiếu ngắn gọn một lần nữa đến Fig.2, các tập hợp con có thể được thành lập (về mặt khái niệm), ví dụ, trong hàng đợi giao dịch 234 bởi logic lập lịch 232. Tuy nhiên, cần hiểu rằng các giao dịch có thể được thành lập thành các tập hợp con, được lưu trữ (trong hàng đợi 234 hoặc nơi khác), hoặc nếu không thì được xử lý theo cách cho phép chúng được nhận diện là tập hợp con theo bất kỳ cách nào. Các phương pháp làm ví dụ để thành lập các tập hợp con được mô tả dưới đây.

Như được chỉ ra bởi khối 306, phương pháp 300 có thể gồm bước lựa chọn một trong các tập hợp con. Các phương pháp làm ví dụ để lựa chọn một trong các tập hợp con được mô tả dưới đây.

Như được chỉ ra bởi khối 308, phương pháp 300 có thể gồm bước gửi các giao dịch trong tập hợp con được lựa chọn tới DRAM, chẳng hạn như, ví dụ, DRAM 206 nhóm kép được mô tả ở trên (Fig.2). Như được mô tả dưới đây, bước gửi các giao dịch tới DRAM có thể gồm lập lịch hoặc lựa chọn thứ tự trong đó các giao dịch trong tập hợp con được lựa chọn được gửi sau đó tới DRAM. Các phương pháp làm ví dụ để gửi các giao dịch trong tập hợp con được lựa chọn tới DRAM được mô tả dưới đây.

Cần hiểu rằng phương pháp 300 có thể được thực hiện lặp đi lặp lại. Ví dụ, bước xác định các giao dịch bộ nhớ như được mô tả ở trên liên quan đến khối 302, bước thành lập các giao dịch thành các tập hợp con như được mô tả ở trên liên quan đến khối 304, và bước gửi các giao dịch đến DRAM như được mô tả ở trên liên quan đến khối 306 về cơ bản có thể được thực hiện liên tục khi các yêu cầu giao dịch được nhận và các giao dịch

tương ứng được xác định. Ví dụ, đối với phương án làm ví dụ được mô tả ở trên liên quan đến Fig.2, logic lập lịch 232 có thể ở trạng thái nhàn rỗi khi không có giao dịch nào được xác định và do đó logic lập lịch 232 không lựa chọn, gửi, v.v., các tập hợp con giao dịch. Ngay sau khi ít nhất một giao dịch đã được xác định và do đó có mặt để logic lập lịch 232 hành động theo, logic lập lịch 232 có thể thay đổi từ trạng thái nhàn rỗi sang trạng thái khởi tạo.

Như được mô tả chi tiết hơn dưới đây, một số hoạt động liên quan đến việc gửi tập hợp con được lựa chọn tới DRAM có thể diễn ra đồng thời hoặc về cơ bản là song song với các hoạt động khác liên quan đến một tập hợp con khác. Ví dụ, các lệnh ACT, PRE và CAS có thể được gửi liên quan đến bước gửi tập hợp con có thể được gọi là tập hợp con "dẫn đầu", trong khi chỉ các lệnh ACT và PRE được gửi liên quan đến bước gửi tập hợp con có thể được gọi là tập hợp con "theo sau". Khi ít nhất một giao dịch đã được xác định và do đó có mặt để logic lập lịch 232 hành động theo, thì logic lập lịch 232 có thể thay đổi từ trạng thái nhàn rỗi sang trạng thái khởi tạo và có thể chỉ định tập hợp con (ví dụ, góc phần tư) mà giao dịch được liên kết là tập hợp con dẫn đầu. Ngay sau khi ít nhất một giao dịch khác được liên kết với tập hợp con (ví dụ, góc phần tư) khác có mặt để logic lập lịch 232 hành động theo, logic lập lịch có thể chỉ định tập hợp con (ví dụ, góc phần tư) khác đó làm tập hợp con theo sau. Các khía cạnh khác liên quan đến các tập hợp con dẫn đầu và theo sau được mô tả dưới đây.

Như được minh họa ở dạng lưu đồ trên Fig.4, phương pháp ví dụ 400 để thành lập tập hợp con có thể gồm các bước hoặc các hành động được mô tả dưới đây liên quan đến các khối 402-414. Phương pháp 400 có thể là ví dụ của khối 304 được mô tả ở trên của phương pháp 300 (Fig.3). Phương pháp 400 có thể được áp dụng cho các giao dịch được nhận diện như đã mô tả ở trên liên quan đến khối 302. Nghĩa là, phương pháp 400 có thể được áp dụng cho các giao dịch được nhóm hoặc được nhận diện bởi sự kết hợp của hướng (tức là, đọc hoặc viết) và nhóm. Ví dụ, trong các phương án DRAM nhóm kép, phương pháp 400 có thể được thực hiện cho mỗi góc phần tư, và tập hợp con bao gồm các giao dịch được lựa chọn trong góc phần tư. Nói chung, phương pháp 400 liên quan đến việc lựa chọn các tập hợp con hiệu quả của các giao dịch. Hiệu quả có thể liên quan đến, ví dụ, độ trải rộng khối, số lượng giao dịch, độ dài cụm giao dịch, hoặc các yếu tố khác.

Như được chỉ ra bởi khối 402, tất cả các giao dịch hit có thể được lựa chọn, bất kể giao dịch hit có thể ở khối nào. Mặc dù không được thể hiện nhằm mục đích rõ ràng, nhưng các giao dịch hit được lựa chọn (H) có thể được tổ chức vào một danh sách. Như được chỉ ra bởi khối 404, một giao dịch miss cho mỗi khối trong các khối không có các giao dịch hit (tức là, khối không có bất kỳ giao dịch hit nào và do đó không có giao dịch được lựa chọn trong khối 402) có thể được lựa chọn. Mặc dù không được thể hiện nhằm mục đích rõ ràng, nhưng các giao dịch miss được lựa chọn (M) có thể được tổ chức vào một danh sách.

Như được chỉ ra bởi khối 406, ngoài các giao dịch được lựa chọn trong các khối 402 và 404, một giao dịch conflict cho mỗi khối trong các khối không có các giao dịch hit hoặc các giao dịch miss (tức là, các khối không có bất kỳ giao dịch hit hoặc giao dịch miss nào và do đó không có các giao dịch được lựa chọn trong khối 402) có thể được lựa chọn. Như được chỉ ra bởi khối 408, tất cả các giao dịch hit-dưới-miss/conflict ở dưới các giao dịch được lựa chọn trong các khối 404 và 406 đều có thể được lựa chọn. Mặc dù không được thể hiện nhằm mục đích rõ ràng, nhưng các giao dịch conflict (C) được lựa chọn cũng như các giao dịch hit dưới miss (Mx) được lựa chọn và các giao dịch hit dưới conflict (Cx) được lựa chọn (gọi chung là, các giao dịch hit dưới miss/conflict ($X=Mx+Cx$) được lựa chọn) có thể được tổ chức vào các danh sách tương ứng.

Như được chỉ ra bởi khối 409, các danh sách của các giao dịch miss (M), các giao dịch hit dưới miss (Mx), các giao dịch conflict (C), và các giao dịch hit dưới conflict (Cx) được lựa chọn có thể được sắp xếp theo thứ tự giảm dần số lượng chu trình dữ liệu của chúng. Số lượng chu kỳ dữ liệu cho giao dịch là số lượng các chu kỳ xung nhịp mà sẽ được sử dụng trên bus dữ liệu DRAM để truyền dữ liệu. Ví dụ, đối với bus dữ liệu 16 bit để truyền 64 byte (512 bit) dữ liệu, sẽ mất 16 chu kỳ xung nhịp DRAM hoặc chu kỳ dữ liệu.

Như được chỉ ra bởi khối 410, các giao dịch được hướng đến các khối cũng đến hạn đối với các hoạt động làm mới có thể được loại bỏ trong một số trường hợp. Ví dụ, theo phương án nhóm kép, nếu bộ điều khiển bộ nhớ cần lập lịch làm mới mỗi khối ("per-bank refresh - PBR") cho khối tại thời điểm bộ điều khiển bộ nhớ thực hiện phương pháp 400 trên góc phần tư mà có các giao dịch đến cùng khối với PBR ở trên, thì bộ điều khiển bộ nhớ có thể bỏ qua (tức là, không lựa chọn) tất cả các giao dịch đến khối đó mà phương pháp 400 có thể lẽ ra đã chỉ định để lựa chọn. Tương tự, ví dụ, nếu bộ điều khiển bộ nhớ

cần lập lịch làm mới toàn bộ khối ("all-bank refresh - ABR") cho góc phần tư tại thời điểm bộ điều khiển bộ nhớ thực hiện phương pháp 400 trên góc phần tư đó, thì bộ điều khiển bộ nhớ có thể không lựa chọn tất cả các giao dịch đến cả hai góc phần tư trong cùng nhóm với ABR ở trên, mà phương pháp 400 có thể lẽ ra đã chỉ định để lựa chọn. Nói cách khác, nếu bộ điều khiển bộ nhớ cần lập lịch ABR cho góc phần tư, thì bộ điều khiển bộ nhớ có thể không thực hiện phương pháp 400 trên cả hai góc phần tư có cùng nhóm với ABR, dẫn đến cả hai tập hợp con góc phần tư trống hoặc không có giá trị.

Như được chỉ ra bởi khối 411, mọi giao dịch không hiệu quả mà đã được lựa chọn trong các khối 402-408 có thể được loại bỏ khỏi tập hợp con. Ví dụ, một hoặc nhiều giao dịch miss, giao dịch conflict, hoặc giao dịch hit-dưới-miss/conflict trong khối có thể được loại bỏ nếu việc lập lịch cho chúng sẽ dẫn đến không hiệu quả. Phương pháp làm ví dụ để loại bỏ các giao dịch không hiệu quả được mô tả chi tiết hơn dưới đây.

Như được chỉ ra bởi các khối 412 và 414, khi các giao dịch được lựa chọn còn lại thành lập tập hợp con bắt đầu được gửi (khối 412) đến DRAM, các giao dịch tiếp theo hoặc bổ sung có thể được lựa chọn (khối 414), tức là, được thêm vào tập hợp con. Như được mô tả chi tiết hơn dưới đây, các giao dịch tiếp theo như vậy có thể được lựa chọn khi các ràng buộc về thời gian có thể cho phép (mà không ảnh hưởng tiêu cực đến hiệu quả). Việc lựa chọn các giao dịch tiếp theo hoặc bổ sung để thêm vào tập hợp con có thể được gọi là "đánh giá lười biếng" (tức là, đánh giá bị trì hoãn miễn là thời gian cho phép). Lưu ý rằng trong một số trường hợp, bộ điều khiển bộ nhớ có thể thêm các giao dịch sau khi nó đã bắt đầu thực hiện phương pháp 400 (ví dụ, để đáp lại các yêu cầu giao dịch đến bổ sung). Trong những trường hợp như vậy, giao dịch mới được thêm vào có thể được đánh giá theo cách được mô tả dưới đây để xác định xem có lựa chọn giao dịch đó để đưa vào tập hợp con hay không.

Như được minh họa ở dạng lưu đồ trên các hình vẽ Fig.5A-5B, phương pháp làm ví dụ 500 để bỏ chọn hoặc loại bỏ các giao dịch không hiệu quả khỏi tập hợp con có thể gồm các bước hoặc các hành động được mô tả dưới đây liên quan đến các khối 502-526. Phương pháp 500 có thể là ví dụ của khối 411 được mô tả ở trên của phương pháp 400 (Fig.4). Các giao dịch đã được lựa chọn trong các khối 402-408 có thể được loại bỏ khỏi tập hợp con nếu chúng không hiệu quả để hoàn thành. Nói một cách khái quát, phương

pháp 500 có thể loại bỏ khỏi tập hợp con các giao dịch miss, giao dịch conflict, và các giao dịch hit-dưới-miss/conflict bất kỳ mà thiếu số lượng chu kỳ dữ liệu còn lại để hoàn thành.

Như được chỉ ra bởi khối 502, có thể xác định xem tổng số hoặc số lượng các giao dịch (“subset_txn_cnt”) trong tập hợp con có thấp hơn ngưỡng giao dịch tối thiểu (“min_txn_threshold”) hay không. Min_txn_threshold là số lượng cố định hoặc được xác định trước mà giá trị phù hợp (ví dụ, 4 hoặc 8) sẽ dễ dàng xuất hiện đối với một người có hiểu biết trung bình trong lĩnh vực kỹ thuật theo các mô tả và các ví dụ sau đây. Tổng số hoặc số lượng các giao dịch có thể là tổng của số lượng giao dịch trong danh sách H, M, C và X được mô tả ở trên liên quan đến các khối 402-408 (Fig.4). Lưu ý rằng có thể có 0 hoặc nhiều giao dịch trong mỗi trong số các danh sách H, M, C và X. Nếu trong khối 502 xác định được rằng tổng số các giao dịch trong tập hợp con nhỏ hơn ngưỡng giao dịch tối thiểu, thì phương pháp 500 có thể kết thúc mà không có bất kỳ giao dịch nào được loại bỏ khỏi tập hợp con.

Nếu khối 502 được xác định rằng tổng số các giao dịch trong tập hợp con không nhỏ hơn ngưỡng giao dịch tối thiểu, thì tất cả các giao dịch hit (ví dụ, trong danh sách H được mô tả ở trên liên quan đến Fig.4) có thể được đánh dấu là không thể loại bỏ, như được chỉ ra bởi khối 504. Nghĩa là, phần còn lại của phương pháp 500 bỏ qua các giao dịch đó trong quá trình xem xét thêm liệu có nên loại bỏ chúng khỏi tập hợp con hay không. Số lượng các giao dịch hit được đánh dấu là không thể loại bỏ được thêm vào số lượng (ban đầu là 0) của các giao dịch không thể loại bỏ.

Như được chỉ ra bởi khối 506, sau khi các giao dịch hit (nếu có) được đánh dấu là không thể loại bỏ, thì một hoặc nhiều giao dịch miss (ví dụ, trong danh sách M được đề cập ở trên) có thể được đánh dấu là không thể loại bỏ. Mặc dù không được thể hiện rõ ràng dưới dạng các bước phụ của khối 506 nhằm mục đích rõ ràng, miễn là có một hoặc nhiều giao dịch trong danh sách M, và miễn là số lượng giao dịch không thể loại bỏ trong tập hợp con chưa đạt đến ngưỡng giao dịch tối thiểu, phương pháp 500 có thể tiếp tục thông qua danh sách M theo từng giao dịch, đánh dấu giao dịch miss tiếp theo là không thể loại bỏ, và theo đó tăng số lượng giao dịch không thể loại bỏ trong tập hợp con.

Như được chỉ ra bởi khối 508, sau khi các giao dịch hit (nếu có) và giao dịch miss (nếu có) được đánh dấu là không thể loại bỏ, thì một hoặc nhiều giao dịch conflict (ví dụ, trong danh sách C được đề cập ở trên) có thể được đánh dấu là không thể loại bỏ. Mặc dù

không được thể hiện rõ ràng dưới dạng các bước phụ của khối 508 nhằm mục đích rõ ràng, miễn là có một hoặc nhiều giao dịch trong danh sách C, và miễn là số lượng giao dịch không thể loại bỏ trong tập hợp con chưa đạt đến ngưỡng giao dịch tối thiểu, phương pháp 500 có thể tiếp tục thông qua danh sách C theo từng giao dịch, đánh dấu mỗi giao dịch conflict là không thể loại bỏ, và theo đó tăng số lượng giao dịch không thể loại bỏ trong tập hợp con.

Như được chỉ ra bởi khối 510, sau khi các giao dịch hit (nếu có), giao dịch miss (nếu có), và các giao dịch conflict (nếu có) đã được đánh dấu là không thể loại bỏ, có thể xác định liệu có bất kỳ giao dịch miss nào còn lại trong tập hợp con hay không (ví dụ, trong danh sách M được đề cập ở trên) mà chưa được đánh dấu là không thể loại bỏ và do đó vẫn có khả năng loại bỏ được. Nếu xác định được trong khối 510 rằng một hoặc nhiều giao dịch miss vẫn có khả năng loại bỏ được, thì có thể xác định được liệu chênh lệch giữa số lượng các chu kỳ dữ liệu khả dụng (“available data cycles - avail_data_cycles”) và số lượng các chu kỳ ACT đang chờ xử lý (“pending ACT cycles - pending_act_cycles”) có lớn hơn tổng của tRRD và tRCD hay không, như được chỉ ra bởi khối 512.

Số lượng ban đầu các chu kỳ dữ liệu khả dụng (available data cycles) (khi khối 512 được thực hiện) là tổng của số lượng các chu kỳ dữ liệu đang chờ xử lý và số lượng các chu kỳ dữ liệu nhàn rỗi (idle data cycles). Điều này có thể được biểu diễn là, ban đầu:

$$\text{avail_data_cycles} = \text{pending_data_cycles} + \text{idle_data_cycles},$$

trong đó pending_data_cycles là số lượng các chu kỳ dữ liệu cần thiết để hoàn thành tất cả các giao dịch không thể loại bỏ trong tập hợp con hiện đang được thành lập (theo bước 304 trên Fig.3). Điều này có thể được biểu diễn là:

$$\text{pending_data_cycles} = \text{data_cycles (giao dịch không được loại bỏ)}.$$

Idle_data_cycles là giá trị lớn hơn (tức là, hàm “max”) của thời gian cần thiết để lập lịch giao dịch conflict trang (trường hợp xấu nhất là tRCD + tRP) và thời gian cần thiết để lập lịch các giao dịch đang chờ xử lý của tập hợp con trước đó, chi phí chuyển đổi (switch cost) sang tập hợp con hiện hành và các chu kỳ dữ liệu không hiệu quả được phép (allowable inefficiency data cycles). Điều này có thể được biểu diễn là:

$$\text{idle_data_cycles} = \max (\text{prev_subset_pending_data_cycles} + \text{switch_cost} +$$

allowable_inefficiency_data_cycles, tRCD + tRP),

trong đó prev_subset_pending_data_cycles là số lượng các chu kỳ dữ liệu cần thiết để hoàn thành tất cả các giao dịch đang chờ xử lý trong tập hợp con được lựa chọn trước đó. Switch_cost là số lượng các chu kỳ dữ liệu mà trôi qua khi chuyển đổi từ bước gửi giao dịch CAS cuối cùng trong một tập hợp con (ví dụ, trong một góc phần tư) sang bước gửi giao dịch CAS thứ nhất trong một tập hợp con khác (ví dụ, trong một góc phần tư khác). (Bước gửi các giao dịch từ bộ điều khiển bộ nhớ đến DRAM được mô tả chi tiết hơn dưới đây). Nói cách khác, switch_cost là thời gian phạt để chuyển đổi giữa các hướng (tức là, từ đọc sang ghi, hoặc từ ghi sang đọc), để chuyển đổi nhóm, hoặc cả hai. Lưu ý rằng cả prev_subset_pending_data_cycles và switch_cost sẽ có giá trị bằng 0 khi không có chuyển đổi giữa tập hợp con dẫn đầu và theo sau (vì logic lập lịch nhàn rỗi). Allowable_inefficiency_data_cycles là số lượng ngưỡng cố định hoặc được xác định trước của các chu kỳ bus dữ liệu mà nhận diện phép đo về sự không hiệu quả được phép để giữ lại theo kiểu cơ hội nhiều giao dịch hơn trong tập hợp con hiện hành, và có thể dễ dàng được lựa chọn bởi một người có hiểu biết trung bình trong lĩnh vực kỹ thuật. Số lượng ban đầu của pending_act_cycles (khi khối 512 được thực hiện) là thời gian cần thiết để lập lịch cho tất cả các ACT đang chờ xử lý trong tập hợp con dẫn đầu hiện hành và các giao dịch không thể loại bỏ trong tập hợp con đang được xem xét. Điều này có thể được biểu diễn là:

$$\text{pending_act_cycles} = \text{pending_act_cycles_in}(\text{tập hợp con dẫn đầu} + \text{các txn không thể loại bỏ ở trên}) * \text{tRRD}.$$

Như được chỉ ra bởi khối 514, nếu xác định được trong khối 512 rằng chênh lệch giữa số lượng các chu kỳ dữ liệu khả dụng và số lượng các chu kỳ ACT đang chờ xử lý lớn hơn tổng của tRRD và tRCD, thì giao dịch miss (được nhận diện trong khối 510) và bất kỳ giao dịch hit nào dưới nó (tức là, giao dịch hit-dưới-miss) trong các danh sách M và Mx sẽ không được loại bỏ mà được phép giữ lại trong tập hợp con (ví dụ, trong các danh sách M và Mx tương ứng của chúng). Ngoài ra, trong khối 514, số lượng các chu kỳ dữ liệu khả dụng có thể được cập nhật bằng cách tăng số lượng các chu kỳ dữ liệu khả dụng theo số lượng các chu kỳ dữ liệu được sử dụng để hoàn thành giao dịch miss đó và bất kỳ giao dịch hit nào dưới nó. Tương tự như vậy, trong khối 514, số lượng các chu kỳ

ACT đang chờ xử lý có thể được cập nhật bằng cách tăng nó theo tRRD. Những cập nhật này có thể được biểu diễn là:

avail_data_cycles += data_cycles(giao dịch miss và bất kỳ giao dịch hit nào dưới nó), và

pending_act_cycles += tRRD.

Tiếp theo khối 514, phương pháp 500 có thể quay lại khối 510 để xác định xem tập hợp con có gồm một giao dịch miss khác mà chưa được đánh dấu là không thể loại bỏ hay không.

Nếu xác định được trong khối 512 đã mô tả ở trên rằng chênh lệch giữa số lượng các chu kỳ dữ liệu khả dụng và số lượng các chu kỳ ACT đang chờ xử lý nhỏ hơn tổng của tRRD và tRCD, thì tất cả các giao dịch miss, các giao dịch conflict, và các giao dịch hit-dưới- miss/conflict còn lại trong tập hợp con (ví dụ, trong các danh sách M, C và X tương ứng), tức là vẫn có thể loại bỏ, có thể được loại bỏ khỏi tập hợp con, như được chỉ ra bởi khối 516. Tiếp theo khối 516, phương pháp 500 có thể kết thúc và không có giao dịch nào khác được loại bỏ khỏi tập hợp con.

Nếu xác định được trong khối 510 rằng không có giao dịch miss nào có khả năng loại bỏ được còn lại trong tập hợp con (ví dụ, trong danh sách M được đề cập ở trên), thì phương pháp 500 có thể tiếp tục ở khối 518 (Fig.5B). Như được chỉ ra bởi khối 518, có thể xác định liệu có bất kỳ giao dịch conflict nào còn lại trong tập hợp con (ví dụ, trong danh sách C được đề cập ở trên) mà chưa được đánh dấu là có thể loại bỏ và do đó vẫn có khả năng loại bỏ được.

Nếu xác định được trong khối 518 rằng không có giao dịch conflict nào mà vẫn có khả năng loại bỏ được, thì phương pháp 500 có thể kết thúc, và không có giao dịch nào khác được loại bỏ khỏi tập hợp con. Nếu xác định được trong khối 518 rằng một hoặc nhiều giao dịch conflict vẫn có khả năng loại bỏ được, thì có thể xác định liệu chênh lệch giữa số lượng các chu kỳ dữ liệu khả dụng và số lượng các chu kỳ ACT đang chờ xử lý có lớn hơn tổng của tRRD và tRCD hay không, và liệu số lượng chu kỳ dữ liệu khả dụng có lớn hơn tổng của tRP và tRCD hay không, như được chỉ ra bởi các khối 520 và 522, tương ứng.

Như được chỉ ra bởi khối 524, nếu xác định được trong các khối 520 và 522 rằng chênh lệch giữa số lượng các chu kỳ dữ liệu khả dụng và số lượng các chu kỳ ACT đang chờ xử lý lớn hơn tổng của tRRD và tRCD, và số lượng các chu kỳ dữ liệu khả dụng lớn hơn tổng của tRP và tRCD, thì giao dịch conflict (được nhận diện trong khối 518) và bất kỳ giao dịch hit nào dưới nó (tức là, giao dịch hit-dưới-conflict) trong các danh sách C và Cx không được loại bỏ mà được phép giữ lại trong tập hợp con (ví dụ, trong các danh sách C và Cx tương ứng của chúng). Ngoài ra, trong khối 524, số lượng các chu kỳ dữ liệu khả dụng có thể được cập nhật bằng cách tăng số lượng các chu kỳ dữ liệu khả dụng theo số lượng các chu kỳ dữ liệu được sử dụng để hoàn thành giao dịch conflict đó và bất kỳ giao dịch hit nào dưới nó. Tương tự như vậy, trong khối 524, số lượng các chu kỳ ACT đang chờ xử lý có thể được cập nhật bằng cách tăng nó theo tRRD. Tiếp theo khối 524, phương pháp 500 có thể quay lại khối 518 để xác định xem tập hợp con có gồm một giao dịch conflict khác mà chưa được đánh dấu là không thể loại bỏ hay không.

Nếu xác định được trong khối 520 đã mô tả ở trên rằng chênh lệch giữa số lượng các chu kỳ dữ liệu khả dụng và số lượng các chu kỳ ACT đang chờ xử lý không lớn hơn tổng của tRRD và tRCD, hoặc nếu xác định được trong khối 522 đã mô tả ở trên rằng số lượng chu kỳ dữ liệu khả dụng không lớn hơn tổng của tRP và tRCD, thì tất cả các giao dịch conflict và các giao dịch hit-dưới-conflict còn lại trong tập hợp con (ví dụ, trong các danh sách C, C và Cx, tương ứng), tức là vẫn có thể loại bỏ, có thể được loại bỏ khỏi tập hợp con, như được chỉ ra bởi khối 526. Tiếp theo khối 526, phương pháp 600 có thể kết thúc và không có giao dịch nào khác được loại bỏ khỏi tập hợp con.

Như được minh họa ở dạng lưu đồ trên Fig.6, phương pháp ví dụ 600 để thêm các giao dịch tiếp theo vào tập hợp con dẫn đầu sau khi bộ điều khiển bộ nhớ đã bắt đầu gửi tập hợp con dẫn đầu đến DRAM có thể gồm các bước hoặc các hành động được mô tả dưới đây liên quan đến các khối 601-618. Phương pháp 600 có thể là ví dụ của khối 414 được mô tả ở trên của phương pháp 400 (Fig.4). Nói một cách khái quát, các giao dịch có thể làm tăng hiệu quả của tập hợp con có thể được thêm vào. Các giao dịch mà phương pháp 600 liên quan có thể gồm các giao dịch mà không có trong bộ điều khiển bộ nhớ tại thời điểm bộ điều khiển bộ nhớ bắt đầu gửi tập hợp con dẫn đầu tới DRAM tương ứng với khối 412, tức là, khi lệnh CAS thứ nhất được đưa ra. Các giao dịch như vậy có thể được gọi là các giao dịch “mới”. Các giao dịch mà phương pháp 600 liên quan có thể gồm các giao dịch mà không được lựa chọn hoặc được loại bỏ tương ứng với các khối 402-411

(Fig.4). Các giao dịch như vậy có thể được gọi là các giao dịch “còn lại”. Phương pháp 600 có thể được thực hiện trên mọi chu kỳ dữ liệu mà lệnh DRAM được liên kết với tập hợp con dẫn đầu có thể được gửi tới DRAM, bắt đầu ngay khi lệnh CAS thứ nhất được liên kết với tập hợp con dẫn đầu được gửi tới DRAM. Sau khi lệnh CAS thứ nhất được liên kết với tập hợp con dẫn đầu được gửi đến DRAM, phương pháp 600 có thể lặp lại (bắt đầu lại ở khối 601) trên mọi chu kỳ dữ liệu mà lệnh DRAM được liên kết với tập hợp con dẫn đầu có thể được gửi đến DRAM, cho đến khi lệnh CAS cuối cùng được liên kết với tập hợp con dẫn đầu được gửi tới DRAM, vì giao dịch hit mới có thể được thêm vào bất kỳ lúc nào trước khi CAS cuối cùng được gửi.

Như được chỉ ra bởi khối 602, có thể xác định liệu có bất kỳ giao dịch hit mới nào hay không. Nếu xác định được trong khối 602 rằng có một hoặc nhiều giao dịch hit mới, thì như được chỉ ra bởi khối 606, giao dịch hit mới có thể được lựa chọn, tức là, được thêm vào tập hợp con vì nó làm tăng hiệu quả của tập hợp con, và các chu kỳ dữ liệu khả dụng (available_data_cycles) được cập nhật để gồm các chu kỳ dữ liệu của các giao dịch hit mới được thêm vào. Tiếp theo khối 606, phương pháp 600 có thể quay lại khối 601 và, trong chu kỳ dữ liệu tiếp theo mà lệnh DRAM có thể được gửi, một lần nữa xác định xem CAS cuối cùng đã được gửi hay chưa. Nếu xác định được trong khối 601 rằng CAS cuối cùng được liên kết với tập hợp con dẫn đầu hiện hành đã được gửi, thì phương pháp 600 có thể kết thúc và không được thực hiện lại cho đến khi CAS thứ nhất được liên kết với tập hợp con dẫn đầu tiếp theo được gửi đi.

Nếu xác định được trong khối 602 rằng không có giao dịch hit mới nào, thì có thể xác định xem có bất kỳ giao dịch hit mới hoặc giao dịch miss còn lại nào trên bất kỳ khối "chưa sử dụng" nào không, như được chỉ ra bởi khối 608. Khối "chưa sử dụng" đề cập đến khối mà không có giao dịch nào trong tập hợp con dẫn đầu hiện hành được hướng đến. Nếu xác định được trong khối 608 rằng có một hoặc nhiều giao dịch miss mới hoặc còn lại trên một hoặc nhiều khối chưa sử dụng, thì khối 610 sẽ xác định liệu chênh lệch giữa số lượng các chu kỳ dữ liệu khả dụng và số lượng các chu kỳ ACT đang chờ xử lý có lớn hơn không hơn tổng của tRRD và tRCD hay không, như được chỉ ra bởi khối 610. Trong phương pháp 600, số lượng "chu kỳ dữ liệu đang chờ xử lý" là số lượng "chu kỳ dữ liệu khả dụng" mà còn lại sau khi phương pháp 500 được mô tả ở trên (các Fig.5A- 5B) được thực hiện. Như được chỉ ra bởi khối 612, nếu xác định được trong khối 610 rằng chênh lệch giữa số lượng các chu kỳ dữ liệu khả dụng và số lượng các chu kỳ ACT đang chờ xử

lý lớn hơn tổng của tRRD và tRCD, thì giao dịch miss mới hoặc còn lại có thể được lựa chọn, tức là, được thêm vào tập hợp con dẫn đầu. Ngoài ra, trong khối 612, số lượng các chu kỳ dữ liệu khả dụng có thể được cập nhật bằng cách tăng số lượng các chu kỳ dữ liệu khả dụng theo số lượng các chu kỳ dữ liệu được sử dụng để hoàn thành giao dịch miss đó. Tương tự như vậy, trong khối 618, số lượng các chu kỳ ACT đang chờ xử lý có thể được cập nhật bằng cách tăng nó theo tRRD. Những cập nhật này có thể được biểu diễn là:

avail_data_cycles += data_cycles(giao dịch conflict), và

pending_act_cycles += tRRD.

Tiếp theo khối 612, phương pháp 600 có thể quay lại khối 601 và, trong chu kỳ xung nhịp tiếp theo mà lệnh DRAM có thể được gửi, một lần nữa xác định xem CAS cuối cùng đã được gửi hay chưa.

Nếu xác định được trong khối 608 rằng không có các giao dịch miss mới hoặc còn lại trên bất kỳ khối chưa sử dụng nào, hoặc nếu xác định được trong khối 610 rằng chênh lệch giữa số lượng các chu kỳ dữ liệu khả dụng và số lượng các chu kỳ ACT đang chờ xử lý không lớn hơn tổng của tRRD và tRCD, thì có thể xác định liệu có bất kỳ giao dịch conflict mới hoặc còn lại nào trên bất kỳ khối chưa sử dụng nào hay không, như được chỉ ra bởi khối 614.

Nếu xác định được trong khối 614 rằng có một hoặc nhiều giao dịch conflict mới hoặc còn lại trên một hoặc nhiều khối chưa sử dụng, thì có thể xác định liệu chênh lệch giữa số lượng các chu kỳ dữ liệu khả dụng và số lượng các chu kỳ ACT đang chờ xử lý có lớn hơn tổng của tRRD và tRCD hay không, và liệu số lượng chu kỳ dữ liệu khả dụng có lớn hơn tổng của tRP và tRCD hay không, như được chỉ ra bởi các khối 616 và 617, tương ứng. Như được chỉ ra bởi khối 618, nếu xác định được trong các khối 616 và 617 rằng chênh lệch giữa số lượng các chu kỳ dữ liệu khả dụng và số lượng các chu kỳ ACT đang chờ xử lý lớn hơn tổng của tRRD và tRCD, và số lượng các chu kỳ dữ liệu khả dụng lớn hơn tổng của tRP và tRCD, thì giao dịch conflict (được nhận diện trong khối 614) có thể được lựa chọn, tức là, được thêm vào tập hợp con dẫn đầu. Ngoài ra, trong khối 618, số lượng các chu kỳ dữ liệu khả dụng có thể được cập nhật bằng cách tăng số lượng các chu kỳ dữ liệu khả dụng theo số lượng các chu kỳ dữ liệu được sử dụng để hoàn thành giao dịch conflict đó. Tương tự như vậy, trong khối 618, số lượng các chu kỳ ACT đang chờ

xử lý có thể được cập nhật bằng cách tăng nó theo tRRD. Những cập nhật này có thể được biểu diễn là:

$avail_data_cycles += data_cycles(\text{giao dịch conflict}), \text{ và}$

$pending_act_cycles += tRRD.$

Phương pháp 600 có thể quay lại khối 601 trong chu kỳ xung nhịp tiếp theo mà lệnh DRAM có thể được gửi, một lần nữa xác định xem CAS cuối cùng đã được gửi hay chưa và nếu không, liệu một giao dịch khác có thể được thêm vào một cách hiệu quả hay không.

Nếu xác định được trong khối 616 đã mô tả ở trên rằng chênh lệch giữa số lượng các chu kỳ dữ liệu khả dụng và số lượng các chu kỳ ACT đang chờ xử lý không lớn hơn tổng của tRRD và tRCD, hoặc nếu xác định được trong khối 617 đã mô tả ở trên rằng số lượng các chu kỳ dữ liệu khả dụng không lớn hơn tổng của tRP và tRCD, thì phương pháp 600 có thể quay lại khối 601 và, trong chu kỳ xung nhịp tiếp theo mà lệnh DRAM có thể được gửi, một lần nữa xác định xem CAS cuối cùng đã được gửi chưa.

Như được minh họa bởi dòng thời gian 700 trên Fig.7, theo phương pháp 600 được mô tả ở trên (Fig.6), các giao dịch hit, các giao dịch miss, hoặc các giao dịch conflict mới hoặc còn lại có thể được thêm vào tập hợp con dẫn đầu sau khi bộ điều khiển bộ nhớ đã đưa ra lệnh ACT hoặc PRE được liên kết với tập hợp con. Các giao dịch hit, giao dịch miss hoặc giao dịch conflict mới hoặc còn lại có thể được thêm vào sau khi CAS (“CAS_0”) thứ nhất được liên kết với tập hợp con được gửi tới DRAM và trước khi CAS cuối cùng (“CAS_n”) được liên kết với tập hợp con được gửi tới DRAM. Giao dịch hit có thể được thêm vào bất kỳ lúc nào trước khi CAS cuối cùng được gửi. Giao dịch miss có thể được thêm vào miễn là ít nhất các chu kỳ tRCD vẫn còn trước CAS cuối cùng. Giao dịch conflict có thể được thêm vào miễn là ít nhất các chu kỳ tRCD + tRP vẫn còn trước CAS cuối cùng. Trong ví dụ được thể hiện trên Fig.7, tại thời điểm 702, bộ điều khiển bộ nhớ đang đánh giá giao dịch mới hoặc còn lại theo phương pháp 600 (Fig.6). Vì thời gian 702 nhỏ hơn các chu kỳ tRCD + tRP trước CAS cuối cùng, nên bộ điều khiển bộ nhớ có thể xác định rằng giao dịch mới hoặc còn lại có thể được thêm vào nếu nó là giao dịch hit hoặc giao dịch miss nhưng không thể được thêm vào nếu nó là giao dịch conflict.

Trên Fig.8, sơ đồ khái niệm 800 minh họa ví dụ về việc áp dụng các phương pháp 400 (Fig.4), 500 (Fig.5), và 600 (Fig.6) được mô tả ở trên cho tập hợp các giao dịch. Các giao dịch đã được xác định là các giao dịch hit (“H”), các giao dịch miss (“M”), các giao

dịch conflict (“C”), và các giao dịch hit-dưới-conflict/miss (“X”) được thể hiện liên quan đến các khối mà giao dịch được hướng đến. Các giao dịch trong mỗi khối cũng được thể hiện theo thứ tự thời gian trong đó bộ điều khiển bộ nhớ đã tạo ra các giao dịch tương ứng với các yêu cầu giao dịch đến, từ giao dịch được tạo ra cũ nhất hoặc sớm nhất đến giao dịch được tạo mới nhất hoặc gần đây nhất. Trong ví dụ minh họa: các giao dịch trong khối thứ nhất (“khối_0”) gồm, theo thứ tự từ cũ nhất đến mới nhất, giao dịch hit 802, giao dịch hit 804, và giao dịch conflict 806; các giao dịch tại khối thứ hai (“khối_1”) gồm giao dịch conflict 808; các giao dịch tại khối thứ ba (“khối_2”) gồm, theo thứ tự từ cũ nhất đến mới nhất, hai giao dịch conflict 810 và 812; các giao dịch tại khối thứ tư (“khối_3”) gồm, theo thứ tự từ cũ nhất đến mới nhất, hai giao dịch conflict 814 và 816 và giao dịch hit 818; các giao dịch tại khối thứ năm (“khối_4”) gồm, theo thứ tự từ cũ nhất đến mới nhất, giao dịch conflict 820 và giao dịch hit-dưới-conflict 822 (tức là, giao dịch ban đầu được nhận diện là giao dịch conflict nhưng giao dịch đó sẽ trở thành giao dịch hit ngay khi giao dịch conflict 820 được hướng đến cùng một hàng trong cùng một khối kích hoạt nó); các giao dịch trong khối thứ sáu (“khối_5”) gồm giao dịch hit 832 mà bộ điều khiển bộ nhớ đã tạo ra một thời gian sau khi bắt đầu gửi tập hợp con tới DRAM; các giao dịch tại khối thứ bảy (“khối_6”) gồm, theo thứ tự từ cũ nhất đến mới nhất, giao dịch miss 824 và giao dịch miss 826; các giao dịch tại khối thứ tám (“khối_7”) gồm, theo thứ tự từ cũ nhất đến mới nhất, giao dịch miss 828 và giao dịch hit-dưới-miss 830 (tức là, giao dịch ban đầu được xác định là giao dịch miss nhưng giao dịch đó sẽ trở thành giao dịch hit ngay khi giao dịch miss 828 được hướng đến cùng một hàng trong cùng một khối được hoàn thành).

Tập hợp con 834 (được biểu thị bằng đường nét đứt trên Fig.8) được gửi đến DRAM có thể được thành lập bằng cách áp dụng các phương pháp 400 (Fig.4) và 500 (Fig.5) được mô tả ở trên cho các giao dịch 802-830. Khi thành tập hợp con 834, tất cả các giao dịch hit 802, 804, và 818 có thể được lựa chọn (tương ứng với khối 402 trên Fig.4). Giao dịch hit 832 mà có thể không có mặt vào thời điểm bộ điều khiển bộ nhớ bắt đầu gửi tập hợp con 832 tới DRAM có thể được thêm vào sau (tương ứng với khối 414 trên Fig.4). Ngoài ra, khi thành lập tập hợp con 832, các giao dịch miss 824 và 828 lần lượt trong khối_6 và khối_7 có thể được lựa chọn (tương ứng với khối 404 trên Fig.4), vì không có giao dịch hit nào có mặt để lựa chọn trong các khối đó. Các giao dịch conflict 808, 810, và 820 trong khối_1, khối_2, và khối_4, tương ứng, có thể được lựa chọn (tương ứng với khối 406 trên Fig.4), vì không có giao dịch hit hoặc miss nào có mặt để lựa chọn trong các khối đó. Khi

giao dịch conflict 820 được lựa chọn, thì giao dịch hit-dưới-conflict 822 cũng có thể được lựa chọn (tương ứng với khối 408 trên Fig.4). Tương tự, khi giao dịch miss 828 đã được lựa chọn, thì giao dịch hit-dưới-miss 830 cũng có thể được lựa chọn (tương ứng với khối 408 trên Fig.4). Lưu ý trong ví dụ này rằng các giao dịch không nằm trong tập hợp con 834 gồm các giao dịch conflict 806, 812, 814, 816 và giao dịch miss 826.

Như được minh họa ở dạng lưu đồ trên Fig.9, phương pháp ví dụ 900 để lựa chọn một trong bốn tập hợp con để gửi tới DRAM có thể gồm đặc điểm sau đây. Phương pháp 900 có thể là ví dụ của khối 306 được mô tả ở trên của phương pháp 300 (Fig.3). Nói chung, tập hợp con có thể được lựa chọn dựa trên kết quả chỉ số được tạo ra bằng cách áp dụng chỉ số cho mỗi tập hợp con. Trong một số phương án, có thể áp dụng các ngoại lệ đối với việc lựa chọn tập hợp con dựa trên chỉ số.

Như được chỉ ra bởi khối 902, có thể xác định liệu có bất kỳ tập hợp con nào chưa được lựa chọn trong số lượng ngưỡng (“max_skip”) của các giao dịch gần đây nhất hay các tập hợp con gần đây nhất đã được gửi tới DRAM. Nếu xác định được trong khối 902 rằng có một tập hợp con chưa được lựa chọn trong số lượng ngưỡng của các giao dịch gần đây nhất, thì tập hợp con đó có thể được lựa chọn, như được chỉ ra bởi khối 904.

Như được chỉ ra bởi khối 906, nếu xác định được trong khối 902 rằng không có tập hợp con nào chưa được lựa chọn trong số lượng ngưỡng của các giao dịch gần đây nhất, thì chỉ số có thể được đánh giá hoặc được tính toán cho mỗi tập hợp con. Ví dụ, theo phương án DRAM nhóm kép, chỉ số có thể được tính toán cho mỗi tập hợp con trong số bốn tập hợp con góc phần tư, tạo ra kết quả chỉ số tương ứng cho mỗi góc phần tư. Nói một cách khái quát, chỉ số có thể dựa trên lợi ích hoặc hiệu quả đạt được bằng cách hoàn thành số lượng các giao dịch trong tập hợp con và có thể gồm phần bù dựa trên các chi phí. Các chi phí có thể gồm, ví dụ, chi phí trong các chu kỳ chuyển đổi giữa các tập hợp con. Như được mô tả dưới đây liên quan đến một ví dụ, các tập hợp con có thể được đánh giá bằng cách sử dụng phương pháp 900 trong khi tập hợp con đang được gửi tới DRAM. Tập hợp con có các yêu cầu dữ liệu đang được gửi đến DRAM (tức là, bằng lệnh CAS) là tập hợp con dẫn đầu, và tập hợp con được lựa chọn bằng cách sử dụng phương pháp 900 để trở thành tập hợp con dẫn đầu tiếp theo sau khi tập hợp con dẫn đầu hiện hành đã gửi xong các yêu cầu dữ liệu của nó tới DRAM là tập hợp con theo sau. Lưu ý rằng các lệnh kích hoạt và nạp trước của các góc phần tư của tập hợp con dẫn đầu và tập hợp con theo

sau có thể được xen kẽ miễn là thứ tự thực hiện của tập hợp con dẫn đầu không bị ảnh hưởng. Tuy nhiên, các CAS của tập hợp con dẫn đầu sẽ luôn được lập lịch trước các CAS của tập hợp con theo sau. Có chi phí liên quan đến việc chuyển đổi giữa các khối, và cũng có chi phí liên quan đến việc chuyển đổi hướng truyền dữ liệu (tức là, từ đọc sang ghi, hoặc từ ghi sang đọc).

Khi định nghĩa chỉ số, các lợi ích có thể được thể hiện theo nhiều cách khác nhau, chẳng hạn như: các chu kỳ dữ liệu (“data_cycles(H,M,C,X)”), biểu diễn số lượng các chu kỳ dữ liệu cần thiết để hoàn thành tất cả các giao dịch hit (“H”), giao dịch miss (“M”), giao dịch conflict (“C”), và các giao dịch hit-dưới-miss/conflict (“X”) trong tập hợp con; độ trải rộng khối (“bank_spread”), biểu diễn số lượng các khối đơn trị đã đi qua để hoàn thành các giao dịch trong tập hợp con; các byte trung bình trên mỗi trang (“average_bytes_per_page(H,M,C,X)”), biểu diễn số lượng byte trung bình trên mỗi trang mà sẽ được truy cập để hoàn thành tất cả các giao dịch hit, giao dịch miss, giao dịch conflict, và giao dịch hit-dưới-miss/conflict trong tập hợp con; số lượng tương đương độ dài cụm tối thiểu (“minBL_equivalent_count(H,M,C,X)”), là số lượng các độ dài cụm DRAM tối thiểu mà sẽ được truy cập để hoàn thành tất cả các giao dịch hit, giao dịch miss, giao dịch conflict, và giao dịch hit-dưới-miss/conflict trong tập hợp con; số lượng giao dịch (“transaction_count(H,M,C,X)”), là số lượng các giao dịch trong tập hợp con; và những lợi ích khác mà có thể thấy được đối với người có hiểu biết trung bình trong lĩnh vực kỹ thuật dựa trên các mô tả và các ví dụ được đưa ra ở đây.

Khi định nghĩa các chỉ số, các chi phí có thể được biểu thị theo nhiều cách khác nhau, chẳng hạn như, chi phí chuyển đổi trong các chu kỳ (“switch_cost_in_cycles”), biểu diễn độ trễ tối thiểu để chuyển đổi giữa các tập hợp con; các chi phí cộng nhóm giống nhau trong các chu kỳ (“same_rank_adder_in_cycles”), biểu diễn sự không hiệu quả của việc gửi các giao dịch đến các khối giống nhau dưới dạng tập hợp con dẫn đầu từ tập hợp con đang xem xét; và các chi phí khác mà có thể thấy được đối với người có hiểu biết trung bình trong lĩnh vực kỹ thuật dựa trên các mô tả và các ví dụ được đưa ra ở đây. Same_rank_adder_in_cycles có thể được định nghĩa là chênh lệch giữa chi phí lập lịch các giao dịch conflict đến các khối giống với tập hợp con dẫn đầu và lợi ích của các chu kỳ dữ liệu chồng lấn từ các giao dịch đến các khối khác với tập hợp con dẫn đầu. Giá trị này xấp xỉ bằng: Same_bank_conflict_cost – Diff_bank_data_cycles, trong đó

Same_bank_conflict_cost là $(tRD2PRE \text{ hoặc } tWR2PRE) + tRP + tRCD$, và Diff_bank_data_cycles là số lượng các chu kỳ dữ liệu chồng lấn được mô tả ở trên.

Khi định nghĩa các chỉ số, các lợi ích và các chi phí được mô tả ở trên có thể được kết hợp theo nhiều cách khác nhau. Ví dụ thứ nhất về chỉ số có thể là: $data_cycles(H,M,C,X) - switch_cost_in_cycles$. Ví dụ thứ hai về chỉ số có thể là: $data_cycles(H,M,C,X) - switch_cost_in_cycles - same_rank_adder_in_cycles$. Ví dụ thứ ba về chỉ số có thể là: $bank_spread \times average_bytes_per_page(H,M,C,X)$. Ví dụ thứ tư về chỉ số có thể là: $bank_spread \times average_bytes_per_page(H,M,C,X) - switch_cost_in_cycles$. Ví dụ thứ năm về chỉ số có thể là: $min_BL_equivalent_count(HM,C,X)$. Ví dụ thứ sáu về chỉ số có thể là: $transaction_count(HM,C,X)$. Các chỉ số khác nữa có thể thấy được đối với người có hiểu biết trung bình trong lĩnh vực kỹ thuật khi xem xét các mô tả và các ví dụ ở trên.

Sau khi chỉ số được áp dụng cho mỗi tập hợp con trong khối 906, có thể xác định xem có ràng buộc hay không, tức là, liệu chỉ số đã đánh giá hoặc đã tạo ra cùng một kết quả cho hai hoặc nhiều tập hợp con, như được chỉ ra bởi khối 908. Nếu xác định được rằng chỉ số đã tạo ra cùng một kết quả cho hai hoặc nhiều tập hợp con (tức là, có sự ràng buộc giữa hai hoặc nhiều tập hợp con), thì tập hợp con mà chỉ số đã tạo ra kết quả cao nhất trên một nhóm khác so với tập hợp con dẫn đầu có thể được lựa chọn, như được chỉ ra bởi khối 910. Nếu vẫn có sự ràng buộc giữa hai tập hợp con trên nhóm khác đó, thì lựa chọn tập hợp con cùng hướng và khác nhóm so với tập hợp con dẫn đầu. Nếu không (tức là, nếu chỉ số đã tạo ra kết quả cho một tập hợp con cao hơn so với kết quả mà chỉ số tạo ra cho các tập hợp con khác), thì tập hợp con mà chỉ số tạo ra giá trị cao nhất có thể được lựa chọn, như được chỉ ra bởi khối 912. Các chính sách khác nữa về việc sử dụng bất kỳ chỉ số nào ở trên có thể thấy được đối với người có hiểu biết trung bình trong lĩnh vực kỹ thuật khi xem xét các mô tả và các ví dụ ở trên.

Như được minh họa bằng dòng thời gian mẫu 1000 trên Fig.10, có khoảng thời gian chồng lấn trong đó một trong các tập hợp con là tập hợp con dẫn đầu và tập hợp con khác là tập hợp con theo sau. Ví dụ, theo phương án DRAM nhóm kép, một trong các góc phần tư (ví dụ, R0) là góc phần tư dẫn đầu trong khi một trong các góc phần tư khác (ví dụ, R1) là góc phần tư theo sau. Trong ví dụ này, trong khoảng thời gian chồng lấn, bộ điều khiển bộ nhớ đang gửi các lệnh đọc và ghi (ví dụ, CAS), ACT, và PRE được liên kết với góc

phần tử dẫn đầu, và bộ điều khiển bộ nhớ chỉ gửi các lệnh ACT và PRE được liên kết với góc phần tư theo sau. Sau khi CAS cuối cùng (“CAS_m”) được liên kết với góc phần tư dẫn đầu được gửi, góc phần tư theo sau sẽ trở thành góc phần tư dẫn đầu. Vì bộ điều khiển bộ nhớ đã lập lịch lệnh ACT hoặc PRE cho góc phần tư đã trở thành góc phần tư dẫn đầu, sau tSwitch bộ điều khiển bộ nhớ có thể ngay lập tức bắt đầu gửi các lệnh CAS, bắt đầu với CAS thứ nhất (“CAS_0”) và kết thúc với CAS cuối cùng (“CAS_n”). Do đó, các chi phí được liên kết với việc mở hoặc đóng hàng (trang) trong một góc phần tư chồng lấn một cách hiệu quả với các lệnh CAS trên một góc phần tư khác.

Như được minh họa ở dạng lưu đồ trên Fig.11, phương pháp 1100 để gửi các giao dịch trong tập hợp con được lựa chọn tới DRAM có thể gồm đặc điểm sau đây. Phương pháp 1100 có thể là ví dụ của khối 308 được mô tả ở trên của phương pháp 300 (Fig.3). Phương pháp 1100 cũng có thể được gọi là ví dụ về lập lịch trong tập hợp con. Như được mô tả ở trên, khi các lệnh đọc hoặc ghi (tức là, các lệnh CAS) được gửi tới DRAM, tập hợp con được liên kết có thể được gọi là tập hợp con dẫn đầu. Phương pháp 1100 có thể xác định thứ tự hoặc mức độ ưu tiên trong đó các giao dịch trong tập hợp con dẫn đầu được gửi tới DRAM. Nói chung, thứ tự hoặc mức độ ưu tiên trong đó các giao dịch trong tập hợp con dẫn đầu được gửi tới DRAM có thể dựa trên các giao dịch trong một tập hợp con khác trên cùng nhóm (theo hướng giao dịch ngược lại). Ví dụ, theo phương án nhóm kép trong đó có bốn tập hợp con hoặc góc phần tư, R0, R1, W0 và W1: nếu R0 là góc phần tư dẫn đầu, thì thứ tự hoặc mức độ ưu tiên trong đó các giao dịch trong R0 được gửi có thể dựa trên các giao dịch trong W0; nếu R1 là góc phần tư dẫn đầu, thì thứ tự hoặc mức độ ưu tiên trong đó các giao dịch trong R1 được gửi có thể dựa trên các giao dịch trong W1; nếu W0 là góc phần tư dẫn đầu, thì thứ tự hoặc mức độ ưu tiên trong đó các giao dịch trong W0 được gửi có thể dựa trên các giao dịch trong R0; và nếu W1 là góc phần tư dẫn đầu, thì thứ tự hoặc mức độ ưu tiên trong đó các giao dịch trong W1 được gửi có thể dựa trên các giao dịch trong R1. Thứ tự hoặc mức độ ưu tiên trong đó các giao dịch trong góc phần tư dẫn đầu được gửi có thể dựa trên các khối mà có các giao dịch chung với các khối của các giao dịch trong góc phần tư khác trên cùng nhóm.

Tiếp tục ví dụ này liên quan đến bốn góc phần tư, nếu xác định được trong khối 1102 rằng góc phần tư dẫn đầu là góc phần tư thứ nhất (R0), thì thứ tự hoặc mức độ ưu tiên mà các giao dịch của nó được gửi có thể dựa trên việc các giao dịch đó có các khối nào chung với các giao dịch trong góc phần tư thứ ba (W0), như được chỉ ra bởi khối 1104.

Nếu khối 1106 xác định rằng góc phần tư dẫn đầu là góc phần tư thứ hai (R1), thì thứ tự hoặc mức độ ưu tiên mà các giao dịch của nó được gửi có thể dựa trên việc các giao dịch đó có các khối nào chung với các giao dịch trong góc phần tư thứ tư (W1), như được chỉ ra bởi khối 1108. Nếu xác định được trong khối 1110 rằng góc phần tư dẫn đầu là góc phần tư thứ ba (W0), thì thứ tự hoặc mức độ ưu tiên mà các giao dịch của nó được gửi có thể dựa trên việc các giao dịch đó có các khối nào chung với các giao dịch trong góc phần tư thứ nhất (R0), như được chỉ ra bởi khối 1112. Nếu xác định được (bằng các xác định phụ định trong các khối 1102, 1106, và 1110) rằng góc phần tư dẫn đầu là góc phần tư thứ tư (W1), thì thứ tự hoặc mức độ ưu tiên mà các giao dịch của nó được gửi có thể dựa trên việc các giao dịch đó có các khối nào chung với các giao dịch trong góc phần tư thứ nhất (R0), như được chỉ ra bởi khối 1114.

Như được minh họa ở dạng lưu đồ trên Fig.12, phương pháp 1200 để ưu tiên thứ tự trong đó các giao dịch trong tập hợp con dẫn đầu (ví dụ, góc phần tư) được gửi tới DRAM dựa trên các khối chung với một tập hợp con khác (ví dụ, góc phần tư khác) trên cùng nhóm có thể gồm đặc điểm sau. Phương pháp 1200 có thể là ví dụ của mỗi trong số các khối 1104, 1108, 1112, và 1114 của phương pháp 1100 (Fig.11).

Như được chỉ ra bởi khối 1202, các giao dịch trong góc phần tư dẫn đầu có thể được nhóm thành ba tập hợp: giao dịch hit, giao dịch miss, và giao dịch conflict. Như được chỉ ra bởi các khối 1204, 1206, và 1208, tương ứng, các giao dịch hit, các giao dịch miss, và các giao dịch conflict sau đó được phân loại theo các khối mà chúng có chung với các giao dịch trong góc phần tư khác trên cùng nhóm. “Được phân loại” nghĩa là các lệnh DRAM của giao dịch tới các khối chung được lập lịch trước các lệnh DRAM của các giao dịch tới các khối khác. Nếu không có giao dịch nào trong số giao dịch hit, giao dịch miss, hoặc giao dịch conflict có bất kỳ khối nào chung với giao dịch trong góc phần tư khác trên cùng nhóm, thì chúng có thể được sắp xếp thứ tự ưu tiên là, ví dụ, tất cả giao dịch hit trước, sau đó là tất cả giao dịch miss (gồm mọi giao dịch hit-dưới-miss), và sau đó là tất cả giao dịch conflict (gồm mọi giao dịch hit-dưới-conflict).

Như được chỉ ra bởi khối 1210, các lệnh CAS, ACT, và PRE có thể được gửi tới DRAM ngay khi chúng thích hợp, tức là, đáp ứng các ràng buộc định thời tương ứng của chúng. Nếu lệnh CAS, ACT, và PRE đều thích hợp trên cùng một chu kỳ xung nhịp, thì

chúng có thể được sắp xếp thứ tự ưu tiên là, ví dụ: CAS trước tiên, tiếp theo là ACT, và sau đó là PRE.

Theo ví dụ về cách phương pháp 1200 có thể được áp dụng, góc phân tử dẫn đầu có thể là R0, trong đó có thể có giao dịch đọc được hướng đến khối_0 và giao dịch đọc hướng đến khối_4. Trong ví dụ này, góc phân tử khác trên cùng nhóm là W0, mà có thể có giao dịch ghi được hướng đến khối_4 và giao dịch ghi được hướng đến khối_5. Áp dụng phương pháp 1200, giao dịch đọc được hướng đến khối_4 được gửi trước giao dịch đọc được hướng đến khối_0 vì khối_4 là khối chung giữa giao dịch đó và các giao dịch trong W0, và khối_0 không phải là khối chung giữa giao dịch đó và các giao dịch trong W0. Thứ tự này của các giao dịch dẫn đến PRE/ACT đối với giao dịch ghi được hướng đến khối_4 xảy ra song song với các CAS của giao dịch đọc được hướng đến khối_0, ngay sau khi giao dịch đọc được hướng đến khối_4 hoàn tất. Ngược lại, nếu phương pháp 1200 không được áp dụng, giao dịch đọc được hướng đến khối_0 có thể được gửi trước giao dịch đọc được hướng đến khối_4. Nếu các giao dịch đã được gửi theo thứ tự này, thì PRE/ACT được liên kết với giao dịch ghi được hướng đến khối_4 sẽ chỉ xảy ra sau các CAS được liên kết với giao dịch đọc đến khối_0 và giao dịch đọc đến khối_4.

Như được minh họa bằng dòng thời gian mẫu 1300 trên Fig.13, giữa thời điểm 1302 và thời điểm 1304 của một trong các góc phân tử (ví dụ, R0) là góc phân tử dẫn đầu trong khi một trong các góc phân tử khác (ví dụ, R1) là góc phân tử theo sau. Đây có thể là trạng thái khởi tạo được mô tả ở trên liên quan đến Fig.2. Giữa thời điểm 1302 và thời điểm 1304, bộ điều khiển bộ nhớ có thể gửi các lệnh đọc (tức là, CAS), ACT, và PRE được liên kết với R0, và bộ điều khiển bộ nhớ có thể chỉ gửi các lệnh ACT và PRE được liên kết với R1. CAS cuối cùng ("CAS_m") được liên kết với R0 được gửi tại thời điểm 1304. Tại thời điểm 1304, R1 trở thành góc phân tử dẫn đầu. Cũng tại thời điểm 1304, góc phân tử (ví dụ, W0) được xác định bằng phương pháp 900 được mô tả ở trên (Fig.9) trở thành góc phân tử theo sau. Đây có thể là trạng thái ổn định được mô tả ở trên liên quan đến Fig.2. Bắt đầu tại thời điểm 1306 tức là tSwitch chu kỳ sau thời điểm 1304, bộ điều khiển bộ nhớ có thể gửi các lệnh đọc (tức là, CAS), ACT, và PRE được liên kết với R1 làm góc phân tử dẫn đầu mới. CAS cuối cùng ("CAS_n") được liên kết với R1 được gửi tại thời điểm 1308. Giữa thời điểm 1304 và thời điểm 1308, bộ điều khiển bộ nhớ có thể chỉ gửi các lệnh ACT và PRE được liên kết với W0 làm góc phân tử theo sau được xác định mới. Tại thời điểm 1308, W0 trở thành góc phân tử dẫn đầu. Bắt đầu tại thời điểm 1310 tức là tSwitch chu kỳ

sau thời điểm 1308, bộ điều khiển bộ nhớ có thể gửi các lệnh ghi (tức là, CAS), ACT và PRE được liên kết với W0 làm góc phần tư dẫn đầu mới. CAS cuối cùng (“CAS_p”) được liên kết với W0 được gửi tại thời điểm 1312. Mẫu được thể hiện trên Fig.13 có thể lặp lại, với việc bộ điều khiển bộ nhớ lựa chọn góc phần tư theo sau tiếp theo, và các góc phần tư chuyển đổi giữa các vai trò dẫn đầu và theo sau mỗi khi bộ điều khiển bộ nhớ hoàn thành bước gửi tất cả các giao dịch được lựa chọn trong tập hợp con của góc phần tư tập hợp con của góc phần tư dẫn đầu.

Như được minh họa trên Fig.14, các phương án ví dụ về các hệ thống và các phương pháp lập lịch giao dịch DRAM có thể được cung cấp trong thiết bị điện toán di động (“portable computing device - PCD”) 1400. PCD 1400 có thể là ví dụ về thiết bị điện toán 100 được mô tả ở trên (Fig.1), như đã nêu ở trên có thể là bất kỳ loại thiết bị điện toán nào.

PCD 1400 có thể gồm SoC 1402. SoC 1402 có thể gồm CPU 1404, GPU 1406, DSP 1407, bộ xử lý tín hiệu tương tự 1408, hoặc các bộ xử lý khác. CPU 1404 có thể gồm nhiều lõi, chẳng hạn như lõi thứ nhất 1404A, lõi thứ hai 1404B, v.v., cho đến lõi thứ N 1404N.

Bộ điều khiển hiển thị 1410 và bộ điều khiển màn hình cảm ứng 1412 có thể được ghép nối với CPU 1404. Màn hình cảm ứng 1414 bên ngoài SoC 1402 có thể được ghép nối với bộ điều khiển hiển thị 1410 và bộ điều khiển màn hình cảm ứng 1412. PCD 1400 có thể còn gồm bộ giải mã video 1416 được ghép nối với CPU 1404. Bộ khuếch đại video 1418 có thể được ghép nối với bộ giải mã video 1416 và màn hình cảm ứng 1414. Cổng video 1420 có thể được ghép nối với bộ khuếch đại video 1418. Bộ điều khiển bus nối tiếp đa năng (“universal serial bus - USB”) 1422 cũng có thể được ghép nối với CPU 1404, và cổng USB 1424 có thể được ghép nối với bộ điều khiển USB 1422. Thẻ module định danh thuê bao (“subscriber identity module - SIM”) 1426 cũng có thể được ghép nối với CPU 1404.

Một hoặc nhiều bộ nhớ có thể được ghép nối với CPU 1404. Một hoặc nhiều bộ nhớ có thể gồm cả bộ nhớ khả biến và bộ nhớ không khả biến. Các ví dụ về bộ nhớ khả biến gồm bộ nhớ truy cập ngẫu nhiên tĩnh (“static random access memory - SRAM”) 1428 và các RAM động (“dynamic RAM - DRAM”) 1430 và 1431. Các bộ nhớ như vậy có thể nằm bên ngoài SoC 1402, chẳng hạn như DRAM 1430, hoặc bên trong SoC 1402, chẳng hạn như DRAM 1431. Bộ điều khiển DRAM 1432 được ghép nối với CPU 1404 có thể

điều khiển việc ghi dữ liệu vào, và đọc dữ liệu từ, các DRAM 1430 và 1431. Theo các phương án khác, bộ điều khiển DRAM như vậy có thể được gồm trong bộ xử lý, chẳng hạn như CPU 1404. Bộ điều khiển DRAM 1432 có thể là ví dụ về bộ điều khiển bộ nhớ 102 (Fig.1) hoặc 202 (Fig.2). Các DRAM 1430 và 1431 có thể là ví dụ về DRAM 108 (Fig.1) hoặc 206 (Fig.2).

CODEC âm thanh nổi 1434 có thể được ghép nối với bộ xử lý tín hiệu tương tự 1408. Hơn nữa, bộ khuếch đại âm thanh 1436 có thể được ghép nối với CODEC âm thanh nổi 1434. Các loa âm thanh nổi thứ nhất và thứ hai 1438 và 1440, tương ứng, có thể được ghép nối với bộ khuếch đại âm thanh 1436. Ngoài ra, bộ khuếch đại micrô 1442 có thể được ghép nối với CODEC âm thanh nổi 1434 và micrô 1444 có thể được ghép nối với bộ khuếch đại micrô 1442. Bộ dò sóng vô tuyến điều chế tần số (“frequency modulation - FM”) 1446 có thể được ghép nối với CODEC âm thanh nổi 1434. Anten FM 1448 có thể được ghép nối với bộ dò sóng vô tuyến FM 1446. Ngoài ra, các tai nghe headphone âm thanh nổi 1450 có thể được ghép nối với CODEC âm thanh nổi 1434. Các thiết bị khác có thể được ghép nối với CPU 1404 gồm một hoặc nhiều camera kỹ thuật số (ví dụ, CCD hoặc CMOS) 1452.

Modem của bộ thu phát RF có thể được ghép nối với bộ xử lý tín hiệu tương tự 1408. Chuyển mạch RF 1456 có thể được ghép nối với bộ thu phát RF 1454 và anten RF 1458. Ngoài ra, bàn phím 1460, tai nghe headset đơn âm (mono) với micrô 1462 và thiết bị rung 1464 có thể được ghép nối với bộ xử lý tín hiệu tương tự 1408.

SoC 1402 có thể có một hoặc nhiều cảm biến nhiệt bên trong hoặc trên chip 1470A và có thể được ghép nối với một hoặc nhiều cảm biến nhiệt bên ngoài hoặc bên ngoài chip 1470B. Bộ điều khiển bộ chuyển đổi tương tự-số (“analog-to-digital converter - ADC”) 1472 có thể chuyển đổi sự sụt giảm điện áp được tạo ra bởi các cảm biến nhiệt 1470A và 1470B thành các tín hiệu số. Nguồn cấp điện 1474 và mạch tích hợp quản lý cấp điện (“power management integrated circuit - PMIC”) 1476 có thể cung cấp điện cho SoC 1402.

Firmware hoặc phần mềm có thể được lưu trữ trong bất kỳ bộ nhớ nào được mô tả ở trên, chẳng hạn như DRAM 1430 hoặc 1431, SRAM 1428, v.v., hoặc có thể được lưu trữ trong bộ nhớ cục bộ mà có thể truy cập trực tiếp bởi phần cứng bộ xử lý mà trên đó phần mềm hoặc firmware thực thi. Việc thực thi firmware hoặc phần mềm như vậy có thể

điều khiển các khía cạnh của bất kỳ phương pháp nào được mô tả ở trên hoặc tạo cấu hình các khía cạnh của bất kỳ hệ thống nào được mô tả ở trên. Bất kỳ bộ nhớ hoặc phương tiện lưu trữ bất biến như vậy nào có firmware hoặc phần mềm được lưu trữ trong đó ở dạng đọc được bằng máy tính để thực thi bởi phần cứng bộ xử lý có thể là ví dụ về "sản phẩm chương trình máy tính", "phương tiện đọc được bằng máy tính", v.v., do các thuật ngữ như vậy được hiểu theo thuật ngữ sáng chế.

Các phương án thay thế sẽ trở nên rõ ràng đối với một người có hiểu biết trung bình trong lĩnh vực kỹ thuật sáng chế có liên quan mà không nằm ngoài phạm vi của của sáng chế. Do đó, mặc dù các khía cạnh được lựa chọn đã được minh họa và được mô tả chi tiết, nhưng có thể hiểu rằng các thay thế và sửa đổi khác nhau có thể được thực hiện trong đó mà không nằm ngoài phạm vi của sáng chế, như được xác định bởi các yêu cầu bảo hộ sau đây.

YÊU CẦU BẢO HỘ

1. Phương pháp lập lịch các giao dịch bộ nhớ truy cập ngẫu nhiên động (dynamic random access memory - DRAM) trong thiết bị điện toán, phương pháp này bao gồm các bước:

xác định, bởi bộ điều khiển bộ nhớ, nhiều giao dịch bộ nhớ tương ứng với nhiều yêu cầu giao dịch nhận được từ một hoặc nhiều bộ xử lý, trong đó bước xác định nhiều giao dịch bộ nhớ bao gồm nhận diện mỗi giao dịch bộ nhớ là một trong số giao dịch đọc và giao dịch ghi và được hướng đến nhóm được nhận diện trong số nhiều nhóm DRAM;

thành lập, bởi bộ điều khiển bộ nhớ, nhiều giao dịch bộ nhớ thành nhiều tập hợp con, mỗi tập hợp con bao gồm một hoặc nhiều giao dịch bộ nhớ được nhận diện bởi cùng một kết hợp của hướng và nhóm DRAM giống nhau;

lựa chọn, bởi bộ điều khiển bộ nhớ, một trong số các tập hợp con;

gửi, bởi bộ điều khiển bộ nhớ, tất cả các giao dịch trong tập hợp con được lựa chọn trong số các tập hợp con tới DRAM; và

trong đó bước chọn một trong các tập hợp con bao gồm:

tính toán chỉ số cho mỗi tập hợp con để tạo ra các kết quả chỉ số tương ứng; và

lựa chọn một trong các tập hợp con tương ứng với kết quả chỉ số cao nhất.

2. Phương pháp theo điểm 1, trong đó

bước xác định nhiều giao dịch bộ nhớ bao gồm nhận diện mỗi giao dịch là một trong số giao dịch hit (thành công), giao dịch miss (không thành công), giao dịch conflict (xung đột) và giao dịch hit-dưới-miss/conflict, mỗi trong số nhiều khối của DRAM không có hoặc có nhiều giao dịch bộ nhớ.

3. Phương pháp theo điểm 2, trong đó bước thành lập nhiều giao dịch bộ nhớ thành nhiều tập hợp con còn bao gồm, đối với mỗi tập hợp con:

lựa chọn mỗi giao dịch được nhận diện là giao dịch hit trong một khối của DRAM; và

lựa chọn mỗi giao dịch được nhận diện là giao dịch miss trong bất kỳ khối nào của DRAM trong đó không có giao dịch nào được nhận diện là giao dịch hit được lựa chọn; và

lựa chọn mỗi giao dịch được nhận diện là giao dịch conflict trong bất kỳ khối nào của DRAM trong đó không có giao dịch nào được nhận diện là giao dịch hit được lựa chọn và không có giao dịch nào được nhận diện là giao dịch miss được lựa chọn.

4. Phương pháp theo điểm 3, trong đó bước thành lập nhiều giao dịch bộ nhớ thành nhiều tập hợp con còn bao gồm, đối với mỗi tập hợp con:

lựa chọn mỗi giao dịch được nhận diện là giao dịch hit-dưới-miss/conflict dưới mỗi giao dịch đã lựa chọn được nhận diện là giao dịch miss và dưới mỗi giao dịch đã lựa chọn được nhận diện là giao dịch conflict.

5. Phương pháp theo điểm 4, trong đó bước thành lập nhiều giao dịch bộ nhớ thành nhiều tập hợp con còn bao gồm, đối với mỗi tập hợp con, loại bỏ một hoặc nhiều giao dịch đã lựa chọn được nhận diện là giao dịch miss, giao dịch conflict, hoặc giao dịch hit-dưới-miss/conflict nếu việc lập lịch cho chúng sẽ dẫn đến vượt quá số lượng ngưỡng của các chu kỳ bus dữ liệu trống.

6. Phương pháp theo điểm 1, trong đó chỉ số gồm phần lợi ích đáp ứng ít nhất một trong số: số lượng các chu kỳ dữ liệu trong tất cả các giao dịch được lựa chọn trong tập hợp con; độ trải rộng khối trong tất cả các giao dịch được lựa chọn trong tập hợp con; số lượng byte trung bình trên mỗi trang trong tất cả các giao dịch được lựa chọn trong tập hợp con; số lượng tương đương độ dài cụm tối thiểu trong tất cả các giao dịch được lựa chọn trong tập hợp con; và số lượng giao dịch trong tất cả các giao dịch được lựa chọn trong tập hợp con.

7. Phương pháp theo điểm 1, trong đó chỉ số gồm phần chi phí đáp ứng ít nhất một trong số: chi phí chuyển đổi tập hợp con; và chi phí thêm nhóm giống nhau.

8. Phương pháp theo điểm 1, trong đó bước gửi tất cả các giao dịch trong tập hợp con được lựa chọn trong số các tập hợp con tới DRAM bao gồm:

xác định thứ tự giao dịch cho tập hợp con được lựa chọn trong số các tập hợp con dựa trên các khối chung giữa tập hợp con được lựa chọn trong số các tập hợp con và một tập hợp con khác ở cùng nhóm với tập hợp con được lựa chọn trong số các tập hợp con; và

gửi các giao dịch bộ nhớ trong tập hợp con được lựa chọn trong số các tập hợp con tới DRAM theo thứ tự giao dịch đã xác định.

9. Phương pháp theo điểm 1, trong đó bước gửi tất cả các giao dịch trong tập hợp con được lựa chọn trong số các tập hợp con còn bao gồm thêm một hoặc nhiều giao dịch vào tập hợp con được lựa chọn trong số các tập hợp con trước khi lệnh lựa chọn địa chỉ cột (column address select - CAS) cuối cùng được liên kết với tập hợp con được lựa chọn trong số các tập hợp con được gửi đến DRAM.

10. Hệ thống lập lịch các giao dịch bộ nhớ trong thiết bị điện toán có bộ điều khiển bộ nhớ và bộ nhớ truy cập ngẫu nhiên động (DRAM), hệ thống này bao gồm:

hàng đợi giao dịch trong bộ điều khiển bộ nhớ; và

logic lập lịch trong bộ điều khiển bộ nhớ, logic lập lịch được tạo cấu hình để:

xác định nhiều giao dịch bộ nhớ tương ứng với nhiều yêu cầu giao dịch nhận được từ một hoặc nhiều bộ xử lý, trong đó mỗi giao dịch bộ nhớ được nhận diện là một trong số giao dịch đọc và giao dịch ghi và được hướng đến nhóm được nhận diện trong số nhiều nhóm DRAM;

thành lập nhiều giao dịch bộ nhớ thành nhiều tập hợp con, mỗi tập hợp con bao gồm một hoặc nhiều giao dịch bộ nhớ được nhận diện bởi cùng một kết hợp của hướng và nhóm DRAM giống nhau;

lựa chọn một trong các tập hợp con;

gửi tất cả các giao dịch trong tập hợp con được lựa chọn tới DRAM thông qua hàng đợi giao dịch; và

trong đó logic lập lịch được tạo cấu hình để lựa chọn một trong các tập hợp con bằng việc được tạo cấu hình để:

tính toán chỉ số cho mỗi tập hợp con để tạo ra kết quả chỉ số tương ứng; và

lựa chọn một trong các tập hợp con tương ứng với kết quả chỉ số cao nhất.

11. Hệ thống theo điểm 10, trong đó:

logic lập lịch được tạo cấu hình để nhận diện mỗi giao dịch là một trong số giao dịch hit, giao dịch miss, giao dịch conflict, và giao dịch hit-dưới-miss/conflict, mỗi trong số nhiều khối của DRAM không có hoặc có nhiều giao dịch bộ nhớ.

12. Hệ thống theo điểm 11, trong đó logic lập lịch được tạo cấu hình để thành lập nhiều giao dịch bộ nhớ thành nhiều tập hợp con bằng cách được tạo cấu hình thêm để, đối với mỗi tập hợp con:

lựa chọn mỗi giao dịch được nhận diện là giao dịch hit trong một khối của DRAM; và

lựa chọn mỗi giao dịch được nhận diện là giao dịch miss trong bất kỳ khối nào của DRAM mà trong đó không có giao dịch nào được nhận diện là giao dịch hit được lựa chọn; và

lựa chọn mỗi giao dịch được nhận diện là giao dịch conflict trong bất kỳ khối nào của DRAM trong đó không có giao dịch nào được nhận diện là giao dịch hit được lựa chọn và không có giao dịch nào được nhận diện là giao dịch miss được lựa chọn.

13. Hệ thống theo điểm 12, trong đó logic lập lịch được tạo cấu hình để thành lập nhiều giao dịch bộ nhớ thành nhiều tập hợp con bằng cách được tạo cấu hình thêm để, đối với mỗi tập hợp con:

lựa chọn mỗi giao dịch được nhận diện là giao dịch hit-dưới-miss/conflict dưới mỗi giao dịch đã lựa chọn được nhận diện là giao dịch miss và dưới mỗi giao dịch đã lựa chọn được nhận diện là giao dịch conflict.

14. Hệ thống theo điểm 13, trong đó logic lập lịch được tạo cấu hình để thành lập nhiều giao dịch bộ nhớ thành nhiều tập hợp con bằng cách được tạo cấu hình thêm để, đối với mỗi tập hợp con, loại bỏ một hoặc nhiều giao dịch đã lựa chọn được nhận diện là giao dịch miss, giao dịch conflict, hoặc giao dịch hit-dưới-miss/conflict, nếu việc lập lịch cho chúng sẽ dẫn đến vượt quá số lượng ngưỡng của các chu kỳ bus dữ liệu trống.

15. Hệ thống theo điểm 10, trong đó chỉ số gồm phần lợi ích đáp ứng ít nhất một trong số: số lượng các chu kỳ dữ liệu trong tất cả các giao dịch được lựa chọn trong tập hợp con; độ trải rộng khối trong tất cả các giao dịch được lựa chọn trong tập hợp con; số lượng byte trung bình trên mỗi trang trong tất cả các giao dịch được lựa chọn trong tập hợp con; số lượng tương đương độ dài cụm tối thiểu trong tất cả các giao dịch được lựa chọn trong tập hợp con; và số lượng giao dịch trong tất cả các giao dịch được lựa chọn trong tập hợp con.

16. Hệ thống theo điểm 10, trong đó chỉ số gồm phần chi phí đáp ứng ít nhất một trong số: chi phí chuyển đổi tập hợp con; và chi phí thêm nhóm giống nhau.

17. Hệ thống theo điểm 10, trong đó logic lập lịch được tạo cấu hình để gửi tất cả các giao dịch trong tập hợp con được lựa chọn trong số các tập hợp con tới DRAM bằng cách được tạo cấu hình để:

xác định thứ tự giao dịch cho tập hợp con được lựa chọn trong số các tập hợp con dựa trên các khối chung giữa tập hợp con được lựa chọn trong số các tập hợp con và một tập hợp con khác ở cùng nhóm với tập hợp con được lựa chọn trong số các tập hợp con; và

gửi các giao dịch bộ nhớ trong tập hợp con được lựa chọn trong số các tập hợp con tới DRAM theo thứ tự giao dịch đã xác định.

18. Hệ thống theo điểm 10, trong đó logic lập lịch được tạo cấu hình để gửi tất cả các giao dịch trong tập hợp con được lựa chọn trong số các tập hợp con tới DRAM bằng cách được tạo cấu hình thêm để thêm một hoặc nhiều giao dịch vào tập hợp con được lựa chọn trong số các tập hợp con trước khi lệnh lựa chọn địa chỉ cột (CAS) cuối cùng được liên kết với tập hợp con được lựa chọn trong số các tập hợp con được gửi tới DRAM.

19. Hệ thống lập lịch các giao dịch bộ nhớ truy cập ngẫu nhiên động (DRAM) trong thiết bị điện toán, hệ thống này bao gồm:

phương tiện để xác định nhiều giao dịch bộ nhớ tương ứng với nhiều yêu cầu giao dịch nhận được từ một hoặc nhiều bộ xử lý, trong đó phương tiện để xác định nhiều giao dịch bộ nhớ nhận diện mỗi giao dịch bộ nhớ là một trong số giao dịch đọc và giao dịch ghi và được hướng đến nhóm được nhận diện trong số nhiều nhóm DRAM;

phương tiện để thành lập nhiều giao dịch bộ nhớ thành nhiều tập hợp con, mỗi tập hợp con bao gồm các giao dịch bộ nhớ được nhận diện bởi cùng một kết hợp của hướng và nhóm DRAM giống nhau;

phương tiện để lựa chọn một trong các tập hợp con;

phương tiện để gửi tất cả các giao dịch trong tập hợp con được lựa chọn trong số các tập hợp con tới DRAM; và

trong đó phương tiện để lựa chọn một trong các tập hợp con bao gồm:
phương tiện để tính toán chỉ số cho mỗi tập hợp con để tạo ra kết quả chỉ số tương ứng; và
phương tiện để lựa chọn một trong các tập hợp con tương ứng với kết quả chỉ số cao nhất.

20. Hệ thống theo điểm 19, trong đó:

phương tiện để xác định nhiều giao dịch bộ nhớ bao gồm phương tiện để nhận diện mỗi giao dịch là một trong số giao dịch hit, giao dịch miss, giao dịch conflict và giao dịch hit-dưới-miss/conflict, mỗi trong số nhiều khối của DRAM không có hoặc có nhiều giao dịch bộ nhớ.

21. Hệ thống theo điểm 20, trong đó phương tiện để thành lập nhiều giao dịch bộ nhớ thành nhiều tập hợp con bao gồm, đối với mỗi tập hợp con:

phương tiện để lựa chọn mỗi giao dịch được nhận diện là giao dịch hit trong một khối của DRAM

phương tiện để lựa chọn mỗi giao dịch được nhận diện là giao dịch miss trong bất kỳ khối nào của DRAM trong đó không có giao dịch nào được nhận diện là giao dịch hit được lựa chọn; và

phương tiện để lựa chọn mỗi giao dịch được nhận diện là giao dịch conflict trong bất kỳ khối nào của DRAM trong đó không có giao dịch nào được nhận diện là giao dịch hit được lựa chọn và không có giao dịch nào được nhận diện là giao dịch miss được lựa chọn.

22. Hệ thống theo điểm 21, trong đó phương tiện để thành lập nhiều giao dịch bộ nhớ thành nhiều tập hợp con còn bao gồm, đối với mỗi tập hợp con:

phương tiện để lựa chọn mỗi giao dịch được nhận diện là giao dịch hit-dưới-miss/conflict dưới mỗi giao dịch đã lựa chọn được nhận diện là giao dịch miss và dưới mỗi giao dịch đã lựa chọn được nhận diện là giao dịch conflict.

23. Hệ thống theo điểm 22, trong đó phương tiện để thành lập nhiều giao dịch bộ nhớ thành nhiều tập hợp con còn bao gồm, đối với mỗi tập hợp con, phương tiện để loại bỏ một hoặc

nhiều giao dịch đã lựa chọn được nhận diện là giao dịch miss, giao dịch conflict, hoặc giao dịch hit-dưới-miss/conflict nếu việc lập lịch cho chúng sẽ dẫn đến vượt quá số lượng ngưỡng của các chu kỳ bus dữ liệu trống.

24. Hệ thống theo điểm 19, trong đó:

chỉ số gồm phần lợi ích đáp ứng ít nhất một trong số: số lượng chu kỳ dữ liệu trong tất cả các giao dịch được lựa chọn trong tập hợp con; độ trải rộng khối trong tất cả các giao dịch được lựa chọn trong tập hợp con; số lượng byte trung bình trên mỗi trang trong tất cả các giao dịch được lựa chọn trong tập hợp con; số lượng tương đương độ dài cụm tối thiểu trong tất cả các giao dịch được lựa chọn trong tập hợp con; và số lượng giao dịch trong tất cả các giao dịch được lựa chọn trong tập hợp con; và

chỉ số gồm phần chi phí đáp ứng ít nhất một trong số: chi phí chuyển đổi tập hợp con; và chi phí thêm nhóm giống nhau.

25. Hệ thống theo điểm 19, trong đó phương tiện để gửi tất cả các giao dịch trong tập hợp con được lựa chọn trong số các tập hợp con tới DRAM bao gồm:

phương tiện để xác định thứ tự giao dịch cho tập hợp con được lựa chọn trong số các tập hợp con thứ nhất, thứ hai, thứ ba, và thứ tư dựa trên các khối chung giữa tập hợp con được lựa chọn trong số các tập hợp con và một tập hợp con khác ở cùng nhóm với tập hợp con được lựa chọn trong số các tập hợp con; và

phương tiện để gửi các giao dịch bộ nhớ trong tập hợp con được lựa chọn trong số các tập hợp con gửi các giao dịch bộ nhớ tới DRAM theo thứ tự giao dịch đã xác định.

26. Hệ thống theo điểm 19, trong đó phương tiện để gửi tất cả các giao dịch trong tập hợp con được lựa chọn trong số các tập hợp con còn bao gồm phương tiện để thêm một hoặc nhiều giao dịch vào tập hợp con được lựa chọn trong số các tập hợp con trước khi lệnh lựa chọn địa chỉ cột (CAS) cuối cùng được liên kết với tập hợp con được lựa chọn trong số các tập hợp con được gửi đến DRAM.

27. Phương tiện bất biến đọc được bằng máy tính có các lệnh được lưu trữ trên đó dưới dạng các lệnh thực thi được bằng máy tính mà khi được thực thi bởi hệ thống xử lý của bộ điều khiển bộ nhớ tạo cấu hình bộ điều khiển bộ nhớ để:

xác định nhiều giao dịch bộ nhớ tương ứng với nhiều yêu cầu giao dịch nhận được từ một hoặc nhiều bộ xử lý, trong đó mỗi giao dịch bộ nhớ được nhận diện là một trong số giao dịch đọc và giao dịch ghi và được hướng đến nhóm được nhận diện trong số nhiều nhóm DRAM;

thành lập nhiều giao dịch bộ nhớ thành nhiều tập hợp con, mỗi tập hợp con bao gồm các giao dịch bộ nhớ được nhận diện bởi cùng một kết hợp của hướng và nhóm DRAM giống nhau;

lựa chọn một trong các tập hợp con bằng cách:
 tính toán chỉ số cho mỗi tập hợp con để tạo ra các kết quả chỉ số tương ứng; và
 lựa chọn một trong các tập hợp con tương ứng với kết quả chỉ số cao nhất; và
 gửi tất cả các giao dịch trong tập hợp con được lựa chọn trong số các tập hợp con tới DRAM.

28. Phương tiện bất biến đọc được bằng máy tính theo điểm 27, trong đó:

các lệnh tạo cấu hình bộ điều khiển bộ nhớ để nhận diện mỗi giao dịch là một trong số giao dịch hit, giao dịch miss, giao dịch conflict, và giao dịch hit-dưới-miss/conflict, mỗi trong số nhiều khối của DRAM không có hoặc có nhiều giao dịch bộ nhớ.

29. Phương tiện bất biến đọc được bằng máy tính theo điểm 28, trong đó:

các lệnh tạo cấu hình bộ điều khiển bộ nhớ để thành lập nhiều giao dịch bộ nhớ thành nhiều tập hợp con bằng cách tạo cấu hình bộ điều khiển bộ nhớ để, đối với mỗi tập hợp con:

lựa chọn mỗi giao dịch được nhận diện là giao dịch hit trong một khối của DRAM;
 lựa chọn mỗi giao dịch được nhận diện là giao dịch miss trong bất kỳ khối nào của DRAM mà trong đó không có giao dịch nào được nhận diện là giao dịch hit được lựa chọn;
 và

lựa chọn mỗi giao dịch được nhận diện là giao dịch conflict trong bất kỳ khối nào của DRAM trong đó không có giao dịch nào được nhận diện là giao dịch hit được lựa chọn và không có giao dịch nào được nhận diện là giao dịch miss được lựa chọn.

30. Phương tiện bất biến đọc được bằng máy tính theo điểm 29, trong đó các lệnh tạo cấu hình bộ điều khiển bộ nhớ để thành lập nhiều giao dịch bộ nhớ thành nhiều tập hợp con bằng cách tạo cấu hình thêm bộ điều khiển bộ nhớ để, đối với mỗi tập hợp con:

lựa chọn mỗi giao dịch được nhận diện là giao dịch hit-dưới-miss/conflict dưới mỗi giao dịch đã lựa chọn được nhận diện là giao dịch miss và dưới mỗi giao dịch đã lựa chọn được nhận diện là giao dịch conflict.

31. Phương tiện bất biến đọc được bằng máy tính theo điểm 30, trong đó các lệnh tạo cấu hình bộ điều khiển bộ nhớ để thành lập nhiều giao dịch bộ nhớ thành nhiều tập hợp con bằng cách tạo cấu hình thêm bộ điều khiển bộ nhớ để, đối với mỗi tập hợp con, loại bỏ một hoặc nhiều giao dịch đã lựa chọn được nhận diện là giao dịch miss, giao dịch conflict, hoặc giao dịch hit-dưới-miss/conflict, nếu việc lập lịch cho chúng sẽ dẫn đến vượt quá số lượng ngưỡng của các chu kỳ bus dữ liệu trống.

32. Phương tiện bất biến đọc được bằng máy tính theo điểm 27, trong đó:

chỉ số gồm phần lợi ích đáp ứng ít nhất một trong số: số lượng chu kỳ dữ liệu trong tất cả các giao dịch được lựa chọn trong tập hợp con; độ trải rộng khối trong tất cả các giao dịch được lựa chọn trong tập hợp con; số lượng byte trung bình trên mỗi trang trong tất cả các giao dịch được lựa chọn trong tập hợp con; số lượng tương đương độ dài cụm tối thiểu trong tất cả các giao dịch được lựa chọn trong tập hợp con; và số lượng giao dịch trong tất cả các giao dịch được lựa chọn trong tập hợp con; và

chỉ số gồm phần chi phí đáp ứng ít nhất một trong số: chi phí chuyển đổi góc phần tư; và chi phí thêm nhóm giống nhau.

33. Phương tiện bất biến đọc được bằng máy tính theo điểm 27, trong đó các lệnh tạo cấu hình bộ điều khiển bộ nhớ để gửi tất cả các giao dịch trong tập hợp con được lựa chọn trong số các tập hợp con tới DRAM bằng cách tạo cấu hình bộ điều khiển bộ nhớ để:

xác định thứ tự giao dịch cho tập hợp con được lựa chọn trong số các tập hợp con dựa trên các khối chung giữa tập hợp con được lựa chọn trong số các tập hợp con và một tập hợp con khác ở cùng nhóm với tập hợp con được lựa chọn trong số các tập hợp con; và

gửi các giao dịch bộ nhớ trong tập hợp con được lựa chọn trong số các tập hợp con tới DRAM theo thứ tự giao dịch đã xác định.

34. Phương tiện bất biến đọc được bằng máy tính theo điểm 27, trong đó các lệnh tạo cấu hình bộ điều khiển bộ nhớ để gửi tất cả các giao dịch trong tập hợp con được lựa chọn trong số các tập hợp con tới DRAM bằng cách tạo cấu hình thêm bộ điều khiển bộ nhớ để thêm một hoặc nhiều giao dịch vào tập hợp con được lựa chọn trong số các tập hợp con trước khi lệnh lựa chọn địa chỉ cột (CAS) cuối cùng được liên kết với tập hợp con được lựa chọn trong số các tập hợp con được gửi tới DRAM.

1/15

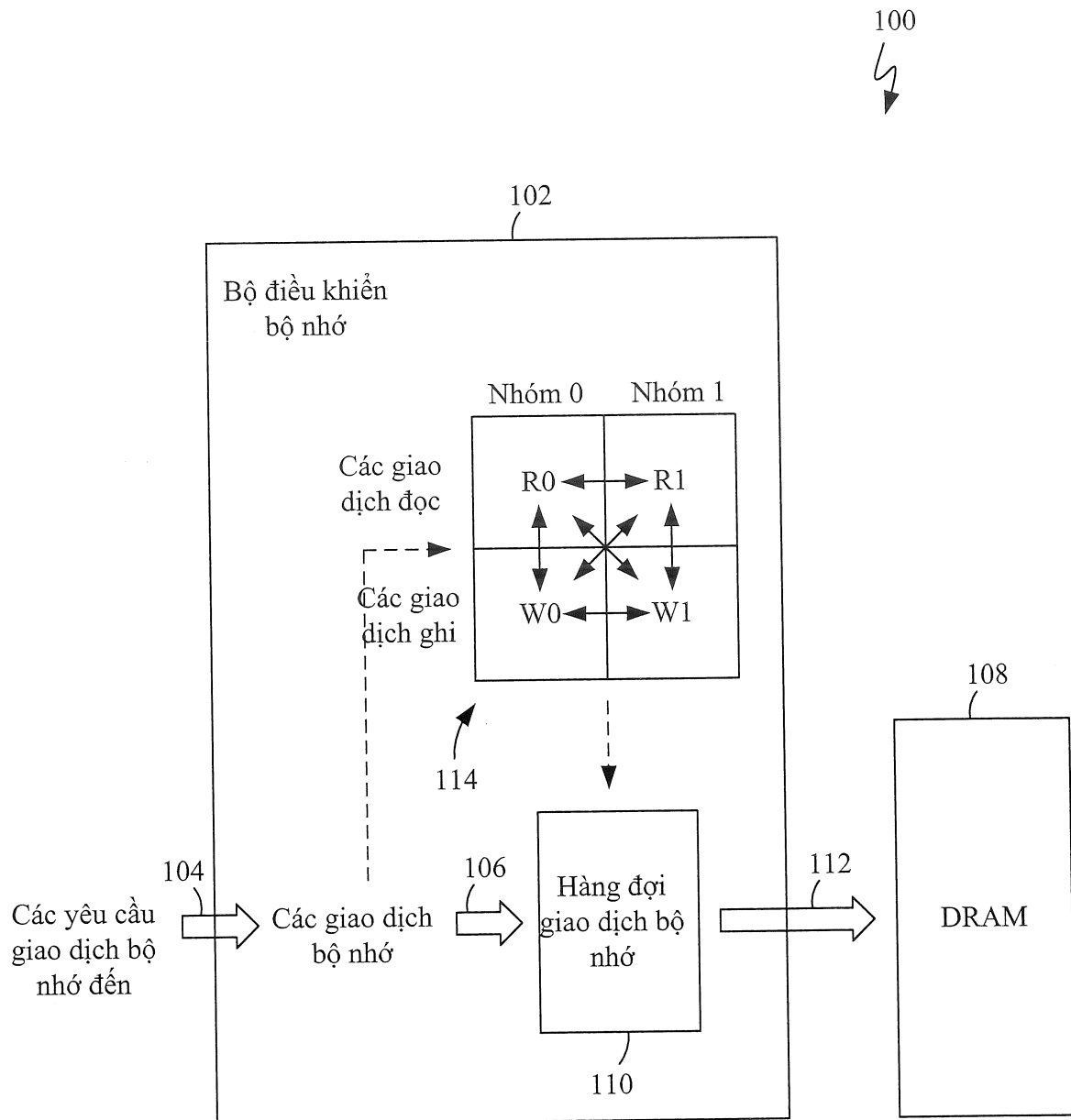


FIG.1

2/15

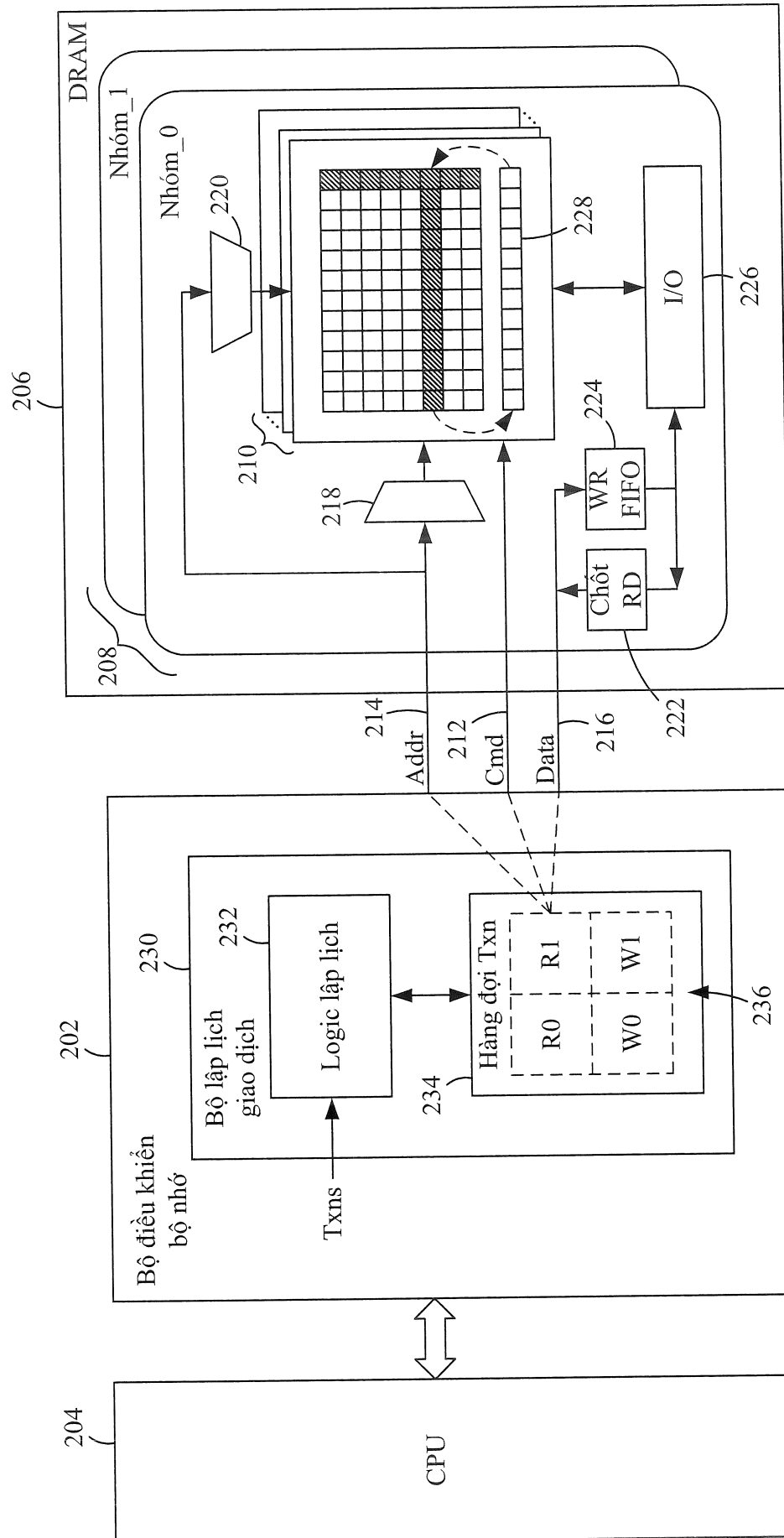


FIG. 2

3/15

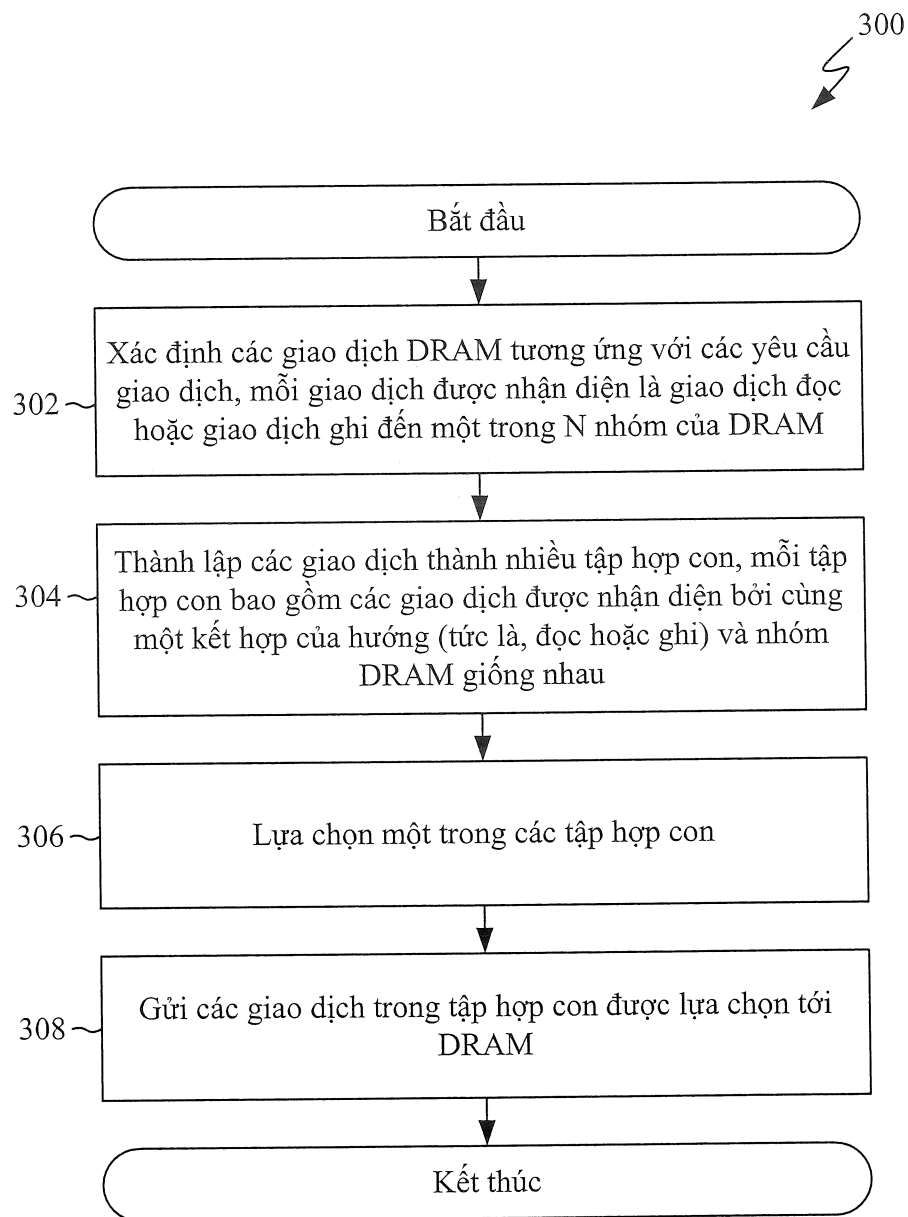


FIG.3

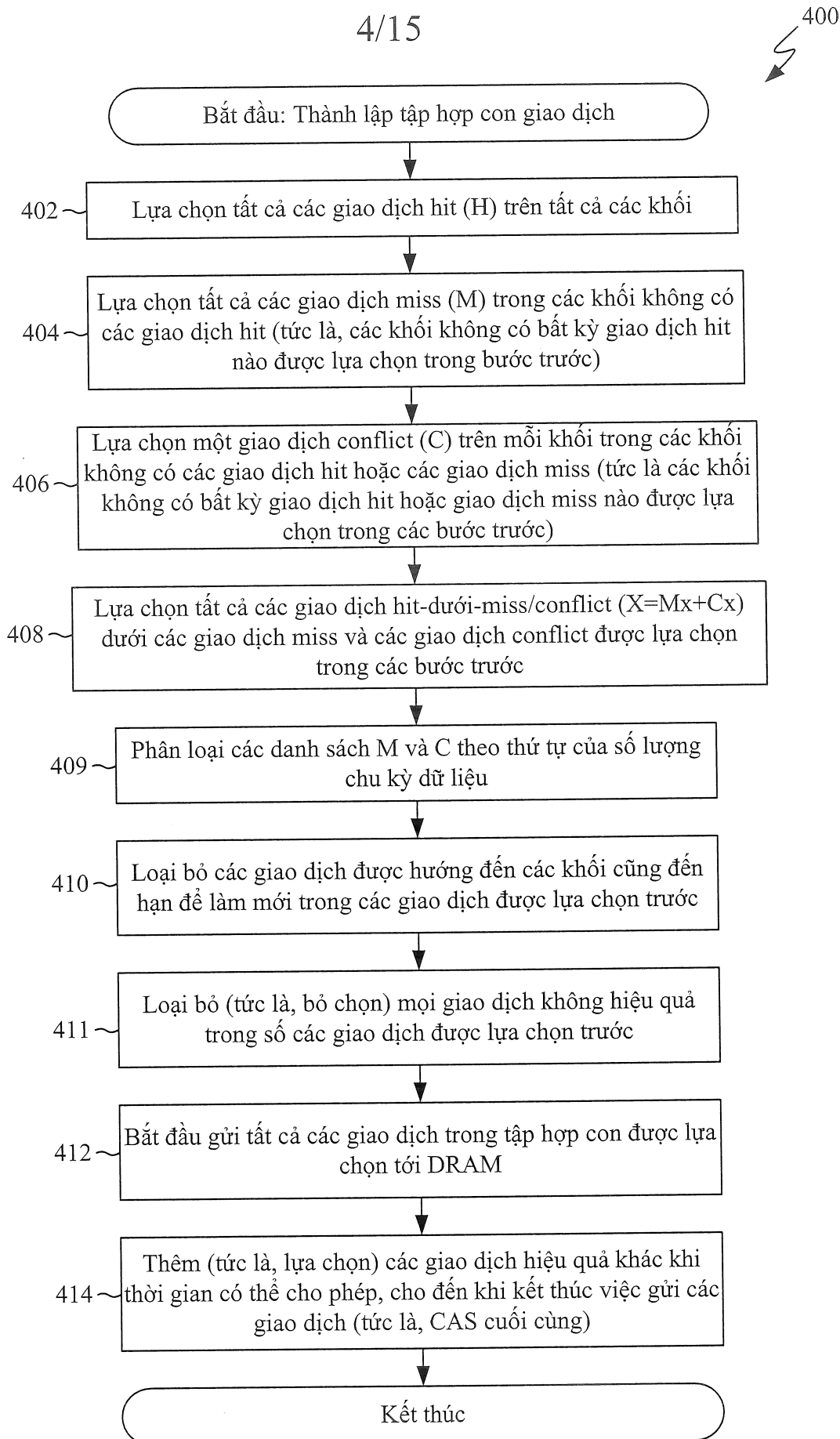


FIG.4

5/15

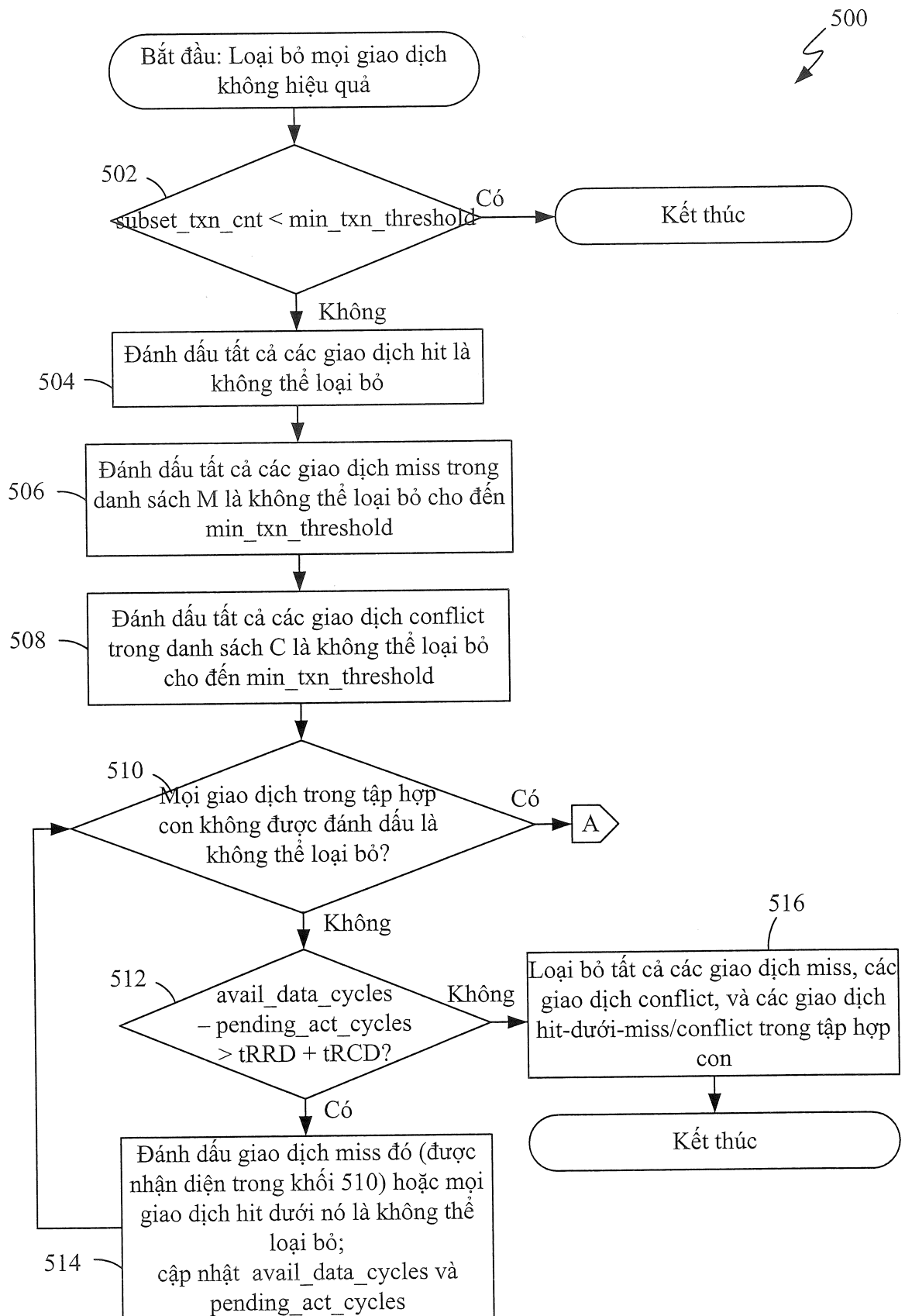


FIG.5A

6/15

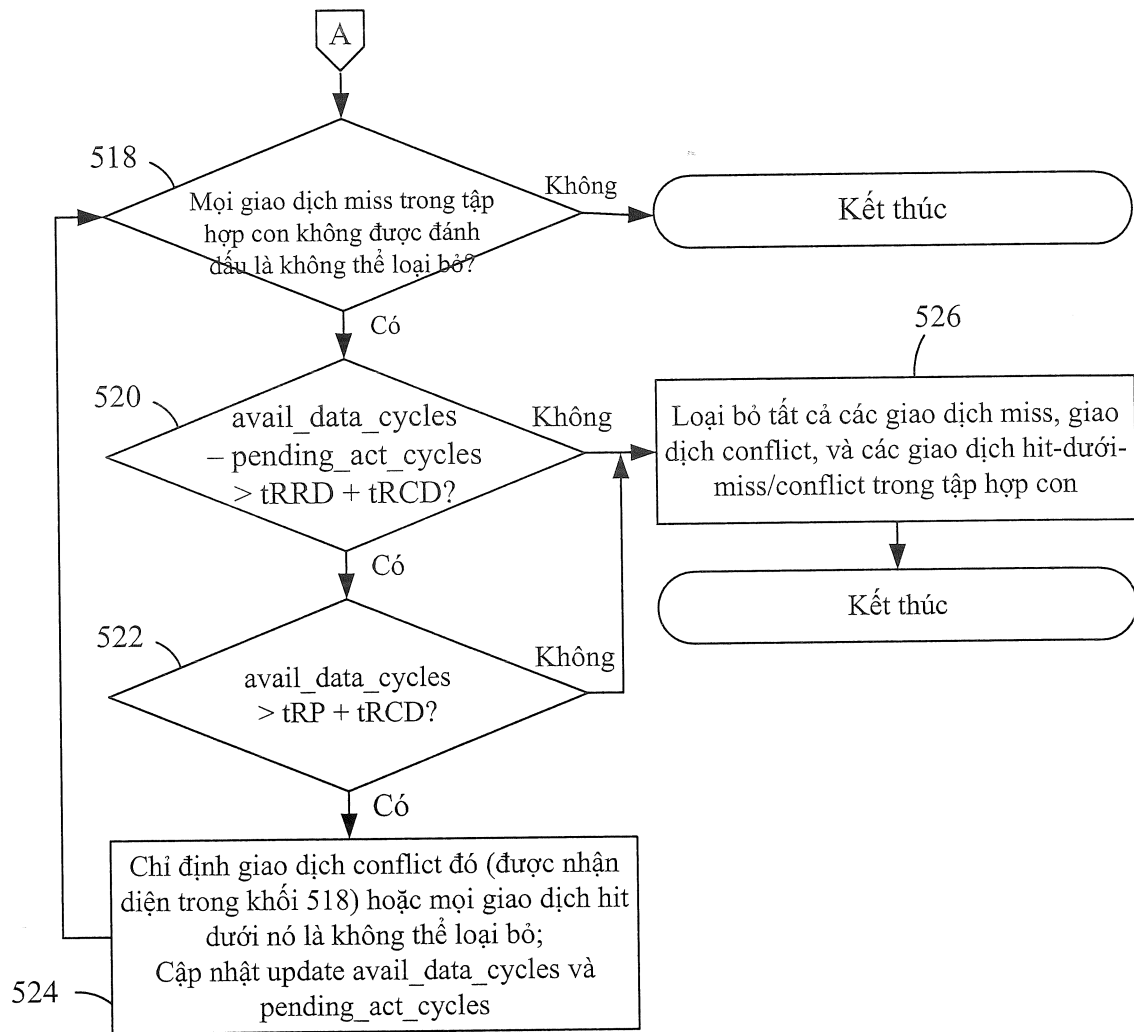
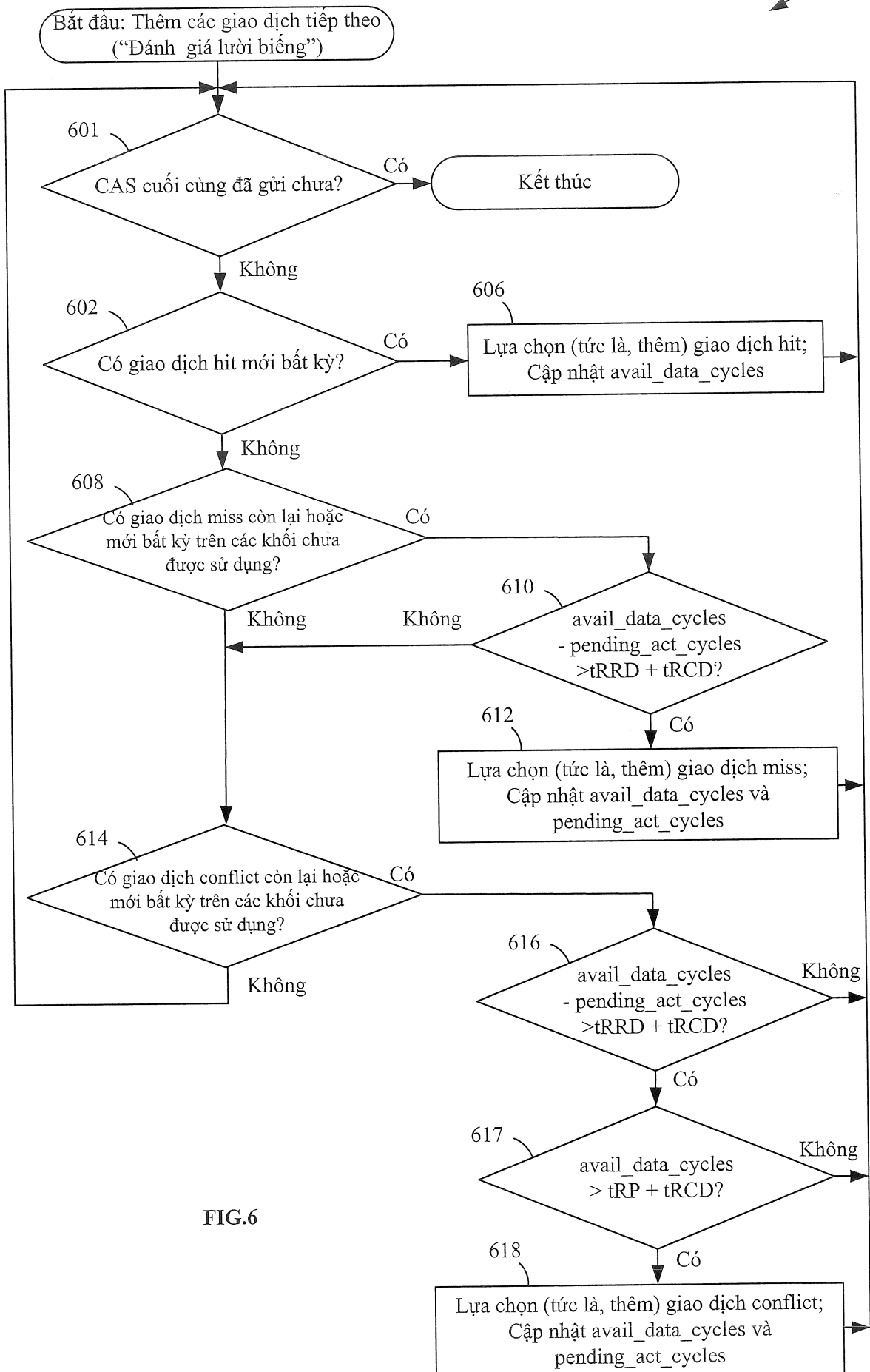


FIG.5B

7/15

600



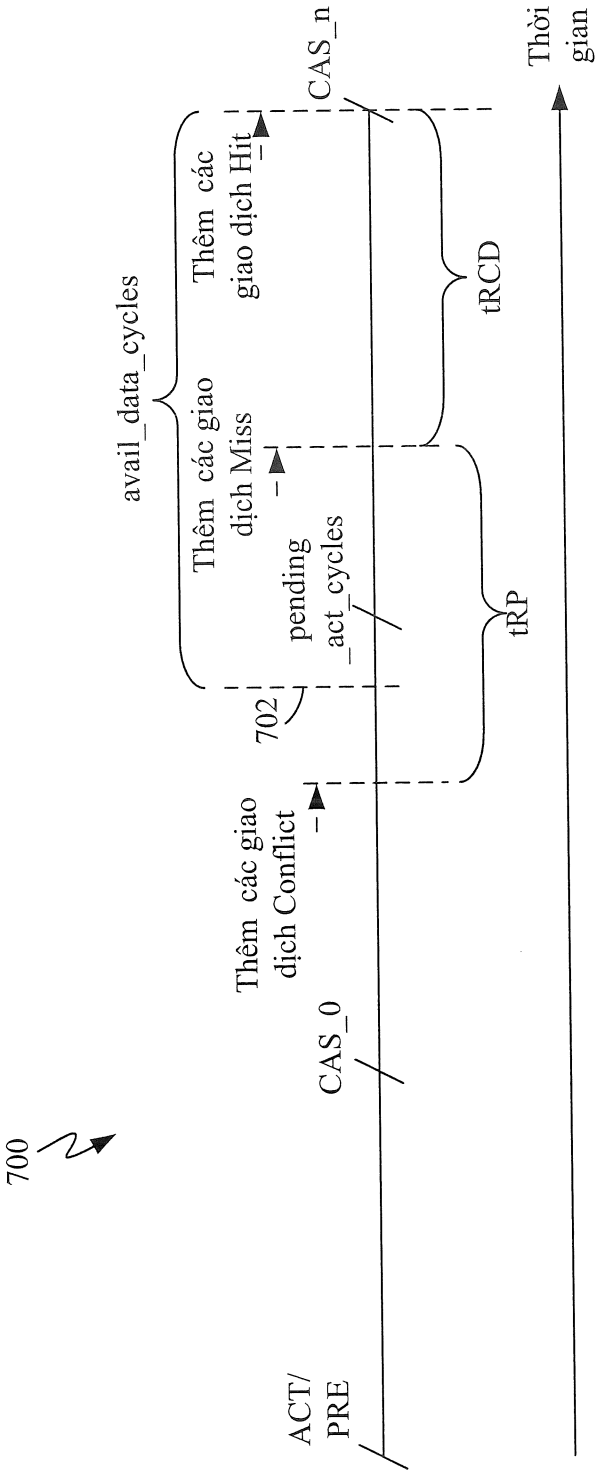


FIG.7

800 ↘

9/15

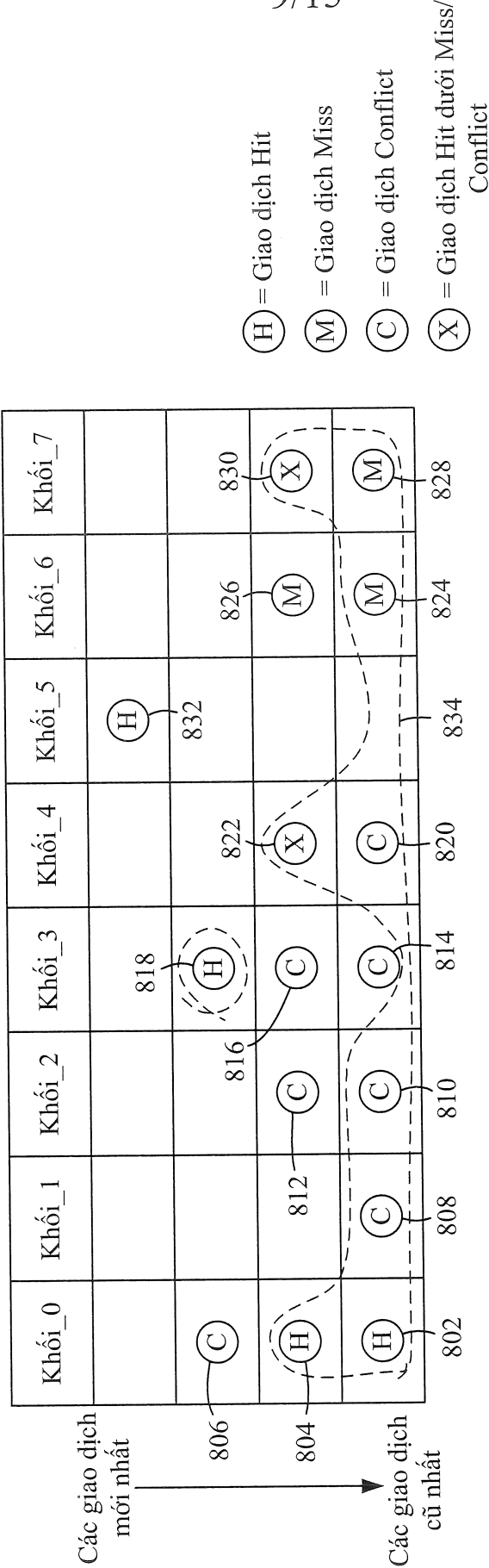


FIG.8

10/15

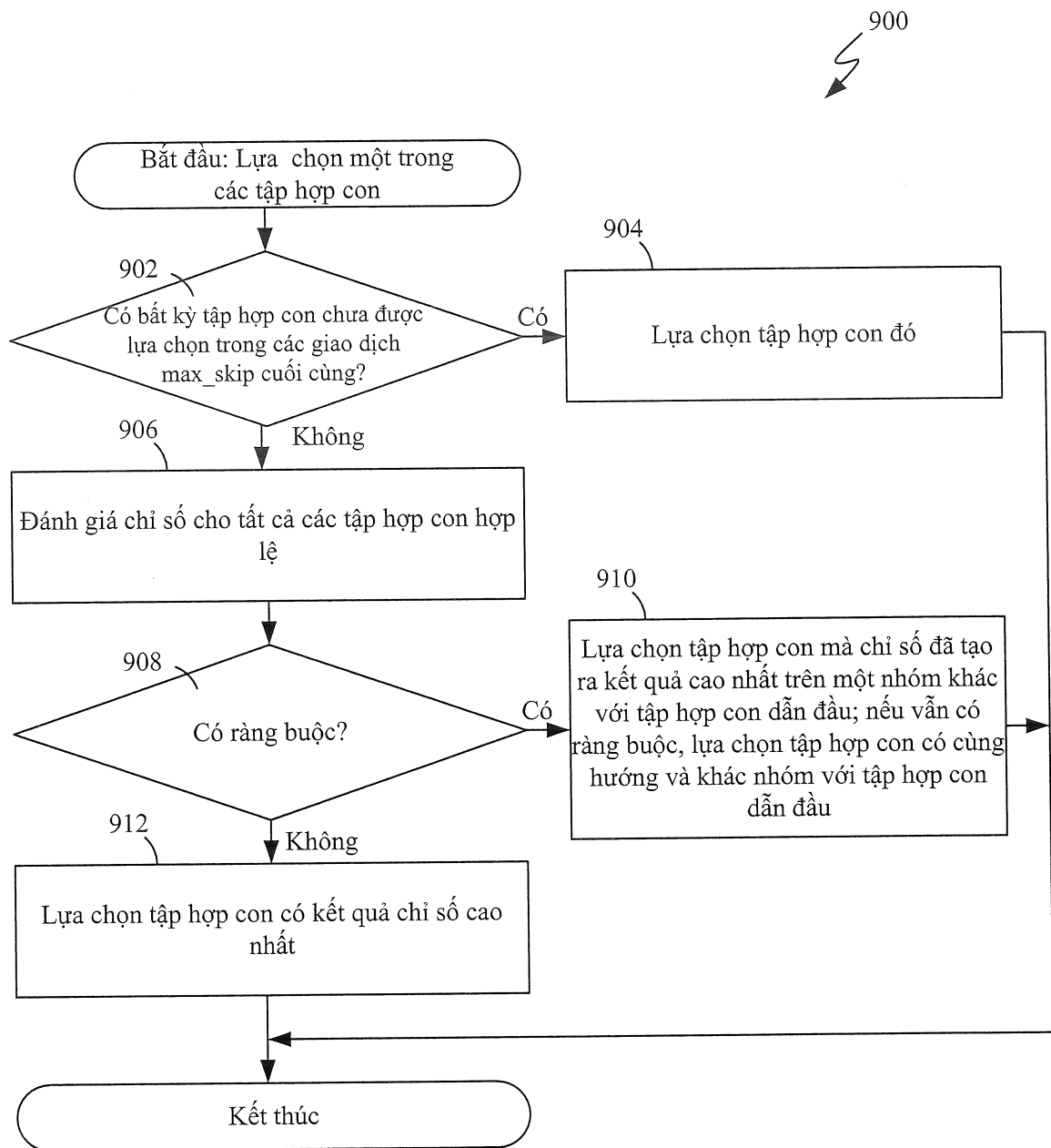


FIG.9

11/15

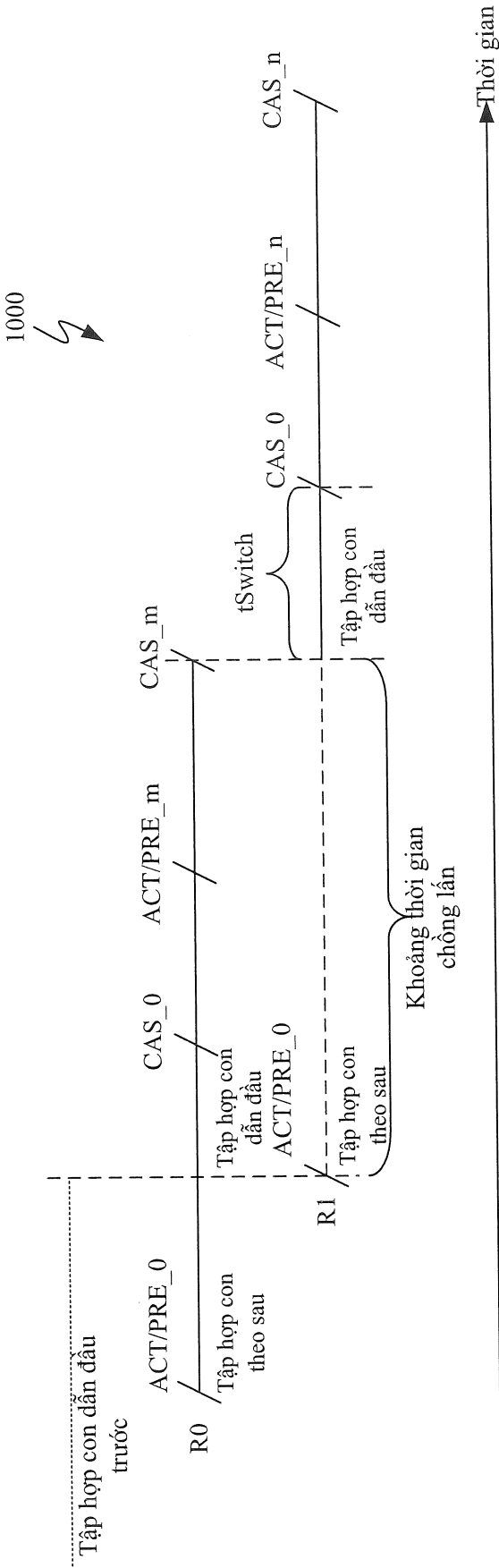


FIG.10

12/15

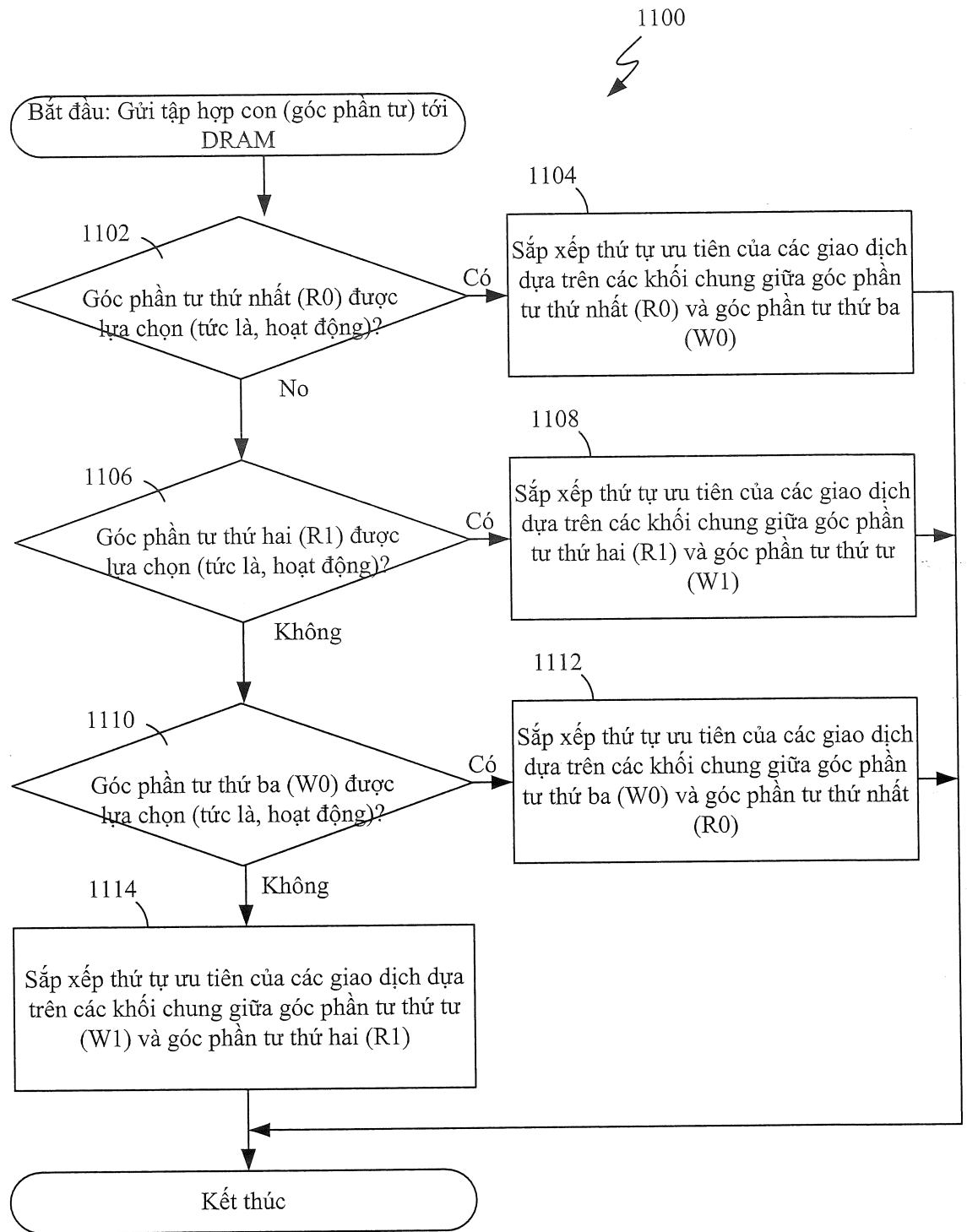


FIG.11

13/15

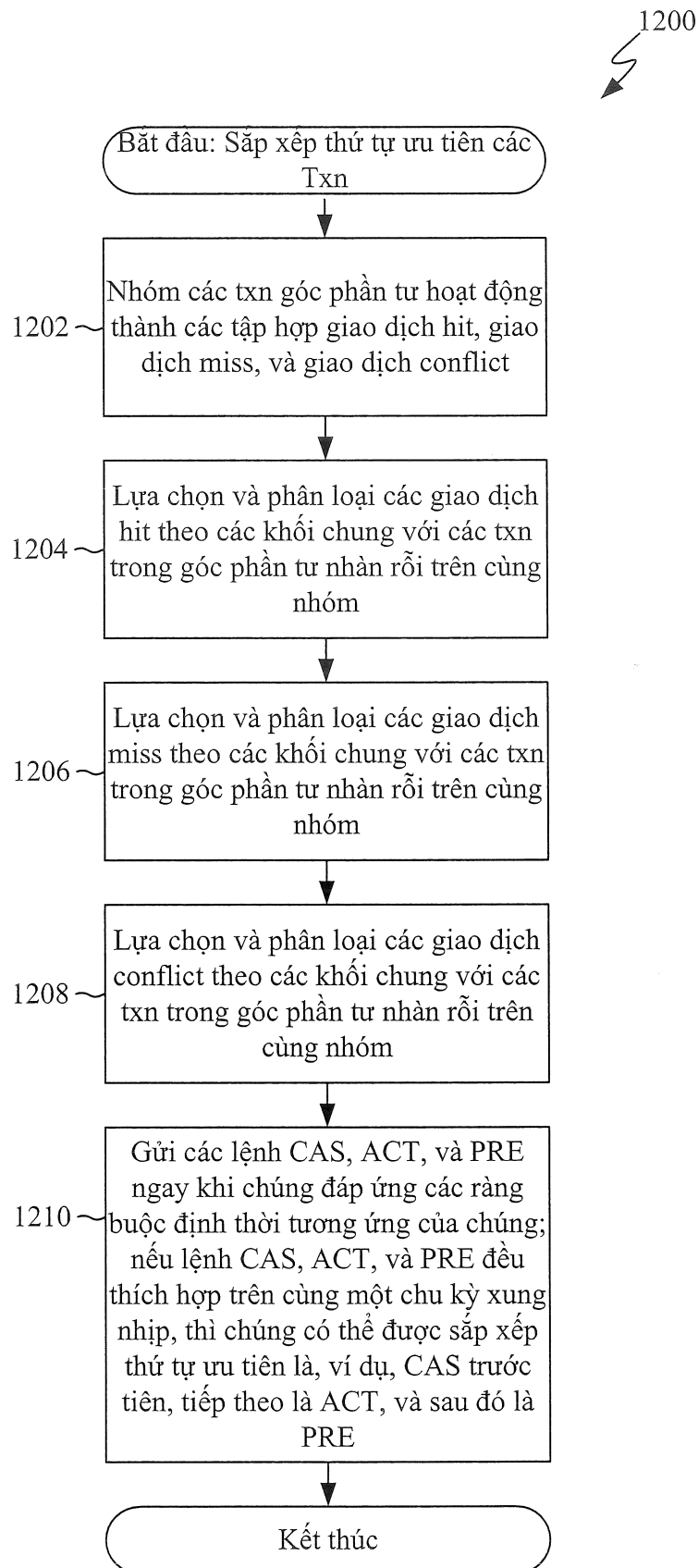


FIG.12

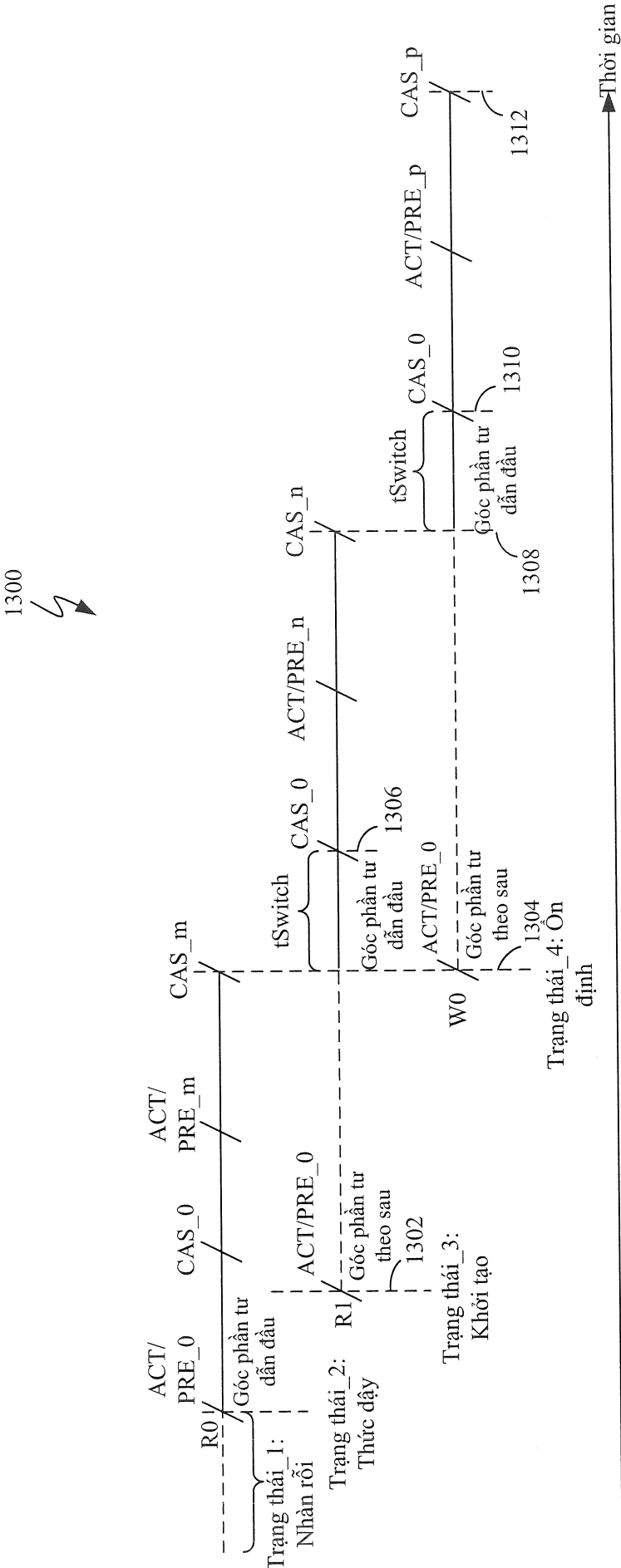


FIG.13

15/15

