



(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)



1-0047737

(51)^{2020.01} G06F 7/38; G06F 9/00

(13) B

(21) 1-2020-07568

(22) 25/12/2020

(30) 16/833,599 28/03/2020 US

(45) 25/06/2025 447

(43) 25/10/2021 403A

(73) INTEL CORPORATION (US)

2200 Mission College Boulevard, Santa Clara, California 95054, United States of America

(72) Shanbhogue, Vedvyas (US); Neiger, Gilbert (US); Gupta, Deepak K. (IN); Anvin, Peter H. (US).

(74) Công ty TNHH Tầm nhìn và Liên danh (VISION & ASSOCIATES CO.LTD.)

(54) BỘ XỬ LÝ, PHƯƠNG PHÁP QUẢN LÝ VÀ XỬ LÝ CÁC NGĂN XẾP BÓNG VÀ VẬT GHI ĐỌC ĐƯỢC BẰNG MÁY

(21) 1-2020-07568

(57) Thiết bị và phương pháp quản lý hiệu quả các ngăn xếp bóng. Ví dụ, một phương án của bộ xử lý bao gồm: các thanh ghi để lưu trữ các con trỏ ngăn xếp bóng (Shadow Stack Pointer, SSP), mỗi SSP được liên kết với mức ưu tiên sự kiện khác nhau; hệ mạch xử lý sự kiện để chọn SSP thứ nhất trong số các SSP từ thanh ghi thứ nhất trong số các thanh ghi đáp ứng lại việc tiếp nhận sự kiện thứ nhất được liên kết với mức ưu tiên sự kiện thứ nhất, SSP thứ nhất sử dụng được để nhận dạng đỉnh của ngăn xếp bóng thứ nhất; hệ mạch kiểm tra xác minh và sử dụng để xác định xem liệu SSP thứ nhất đã được xác minh trước đó hay chưa, trong đó nếu SSP thứ nhất chưa được xác minh trước đó thì khởi tạo tập hợp các hoạt động nguyên tử để xác minh SSP thứ nhất và xác nhận rằng SSP thứ nhất đang không được sử dụng, tập hợp các hoạt động nguyên tử này sử dụng hoạt động khóa để khóa dữ liệu cho đến khi hoàn thành tập hợp các hoạt động nguyên tử, và trong đó nếu SSP thứ nhất được xác minh trước đó, thì xác minh lại SSP thứ nhất và xác nhận rằng SSP thứ nhất đang không được sử dụng mà không sử dụng hoạt động khóa.

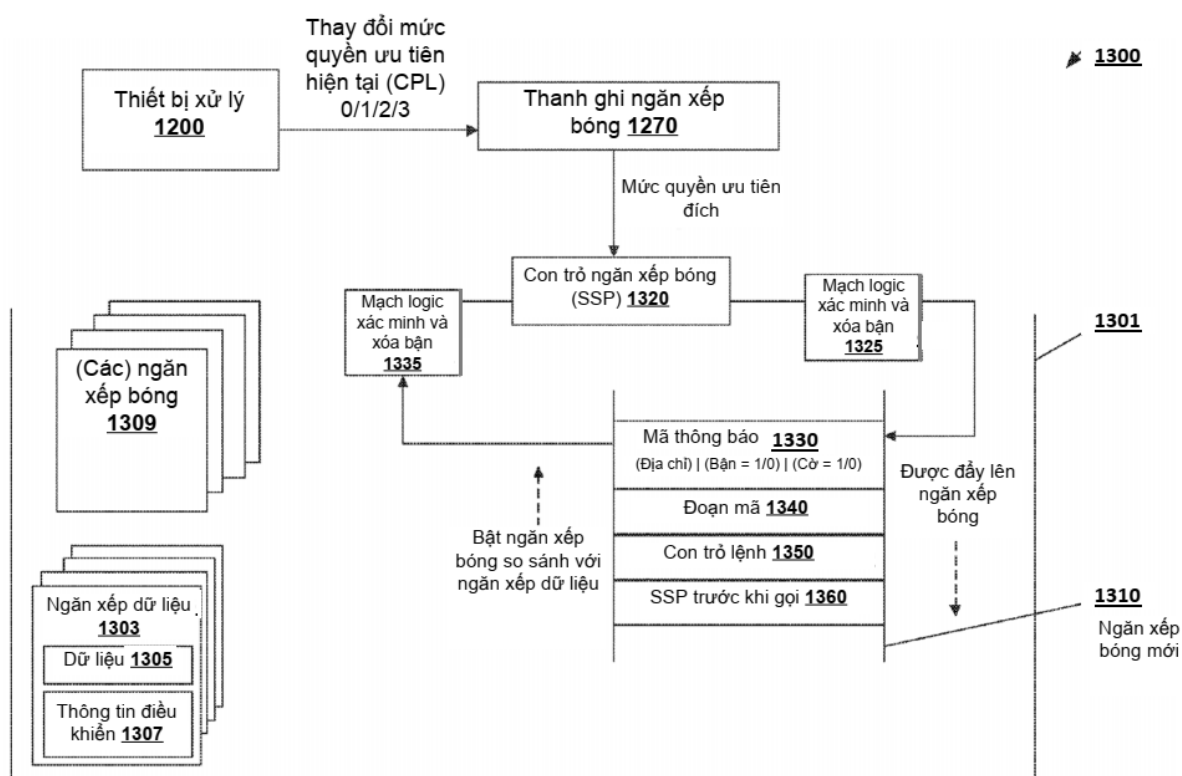


FIG. 13

Lĩnh vực kỹ thuật được đề cập

Nói chung, sáng chế đề cập đến lĩnh vực các bộ xử lý máy tính. Cụ thể là, các phương án đề cập đến thiết bị và phương pháp quản lý và xử lý hiệu quả các ngăn xếp bóng.

Tình trạng kỹ thuật của sáng chế

Các khai thác lập trình hướng trở lại (Return-Oriented Programming, ROP) là một hình thức ngày càng phổ biến của phần mềm độc hại (malicious software, malware) có thể phá vỡ các biện pháp bảo vệ nhất định mà đánh dấu các vị trí của bộ nhớ là không thể thực thi được. Khai thác ROP hoạt động bằng cách ghép thành chuỗi số lượng lớn đoạn mã thực thi được hiện có với nhau mà mỗi đoạn kết thúc bằng lệnh “trả lại” (được gọi là tiện ích). Mỗi tiện ích ROP thường ngắn, và thường không tương ứng với thủ tục hiện có hoặc thậm chí cả biên lệnh trong mã thực thi được. Kẻ tấn công xây dựng ngăn xếp độc hại bao gồm một loạt địa chỉ trả về trở đến chuỗi tiện ích mong muốn. Khai thác ROP được thực hiện bằng cách làm cho bộ xử lý của máy tính thực thi phần mềm sử dụng ngăn xếp độc hại thay vì ngăn xếp hệ thống hợp pháp. Ví dụ, ngăn xếp độc hại có thể được đưa vào bằng cách phá vỡ ngăn xếp, sử dụng khai thác tràn bộ đệm, chuyển hướng sang ngăn xếp mới, hoặc làm hỏng ngăn xếp hệ thống.

Các khai thác ROP nhất định có thể được ngăn chặn bằng cách duy trì “ngăn xếp bóng” song song với ngăn xếp hệ thống thông thường (cũng được gọi “ngăn xếp kế thừa”). Ngăn xếp bóng này duy trì bản sao của ngăn xếp kế thừa trong bộ nhớ không truy nhập được vào phần mềm thông thường, và có thể được sử dụng để xác định xem liệu ngăn xếp kế thừa có bị giả mạo bởi phần mềm độc hại hay không. Ngăn xếp bóng này có thể được thực hiện bằng cách sử dụng thiết bị đo nhị phân, điều này gây ra sự chậm lại hiệu suất đáng kể đối với một số cách sử dụng.

Các biện pháp khác có sẵn để giúp ngăn chặn việc khai thác ROP. Ví dụ, các giá trị “canary” có thể được chèn gần các địa chỉ trả về trong ngăn xếp và có thể được theo dõi

các thay đổi. Ví dụ khác, “các lệnh kết thúc chuyển giao điều khiển” có thể được chèn vào các mã nhị phân để xác định cụ thể các đích trả về hợp pháp. Tuy nhiên, các biện pháp đó có thể yêu cầu biên dịch lại hoặc sửa đổi phần mềm khách. Ngoài ra, một số kiến trúc bộ xử lý nhất định có thể cung cấp ngăn xếp lệnh gọi không thể truy nhập được cho một số phần mềm nhất định. Ví dụ, một số bộ vi điều khiển nhất định có thể duy trì ngăn xếp lệnh gọi không thể truy nhập được vào phần mềm. Ví dụ khác, một số kiến trúc bộ xử lý nhất định có thể duy trì thông tin ngăn xếp lệnh gọi trong một vùng bộ nhớ riêng biệt với các giá trị ngăn xếp khác, chẳng hạn như các biến tự động.

Mô tả vắn tắt các hình vẽ

Có thể hiểu sáng chế rõ hơn nhờ phần mô tả chi tiết tiếp theo cùng với các hình vẽ dưới đây, trong đó:

FIG.1A và FIG.1B là các sơ đồ khối minh họa định dạng lệnh tiện lợi vector chung và các mẫu lệnh của chúng theo các phương án của sáng chế;

FIG.2A-FIG.2C là các sơ đồ khối minh họa định dạng lệnh VEX làm ví dụ theo các phương án của sáng chế;

FIG.3 là sơ đồ khối của kiến trúc thanh ghi theo một phương án của sáng chế; và

FIG.4A là sơ đồ khối minh họa cả đường truyền bằng ống nạp, giải mã, ẩn theo thứ tự làm ví dụ và đường truyền bằng ống đổi tên thanh ghi, phát hành/thực thi không theo thứ tự làm ví dụ theo các phương án của sáng chế;

FIG.4B là sơ đồ khối minh họa cả phương án làm ví dụ về lỗi nạp, giải mã, ẩn theo thứ tự và lỗi kiến trúc đổi tên thanh ghi, phát hành/thực thi không theo thứ tự nằm trong bộ xử lý theo các phương án của sáng chế;

FIG.5A là sơ đồ khối của lõi bộ xử lý đơn, cùng với kết nối của nó với mạng kết nối trên khuôn;

FIG.5B minh họa hình vẽ phóng to một phần của lõi bộ xử lý trên FIG 5A theo các phương án của sáng chế;

FIG.6 là sơ đồ khối của bộ xử lý đơn lõi và bộ xử lý đa lõi có bộ điều khiển bộ nhớ được tích hợp và đồ họa theo các phương án của sáng chế;

FIG.7 minh họa sơ đồ khối của hệ thống theo một phương án của sáng chế;

FIG.8 minh họa sơ đồ khối của hệ thống thứ hai theo một phương án của sáng chế;

FIG.9 minh họa sơ đồ khối của hệ thống thứ ba theo một phương án của sáng chế;

FIG.10 minh họa sơ đồ khối của hệ thống trên chip (System on Chip, SoC) theo một phương án của sáng chế;

FIG.11 minh họa sơ đồ khối tương phản với việc sử dụng bộ biến đổi lệnh phần mềm để biến đổi các lệnh nhị phân trong tập hợp lệnh nguồn thành các lệnh nhị phân trong tập hợp lệnh đích theo các phương án của sáng chế;

FIG.12 minh họa thiết bị xử lý có các ngăn xếp bóng theo một phương án;

FIG.13 minh họa quản lý con trỏ ngăn xếp bóng theo một phương án;

FIG.14 minh họa một phương án xử lý các SSP dựa vào các lần ngắt;

FIG.15 minh họa chi tiết bổ sung được liên kết với quản lý con trỏ ngăn xếp bóng theo một phương án;

FIG.16 minh họa phương án trong đó SSP được chọn và xác minh bằng cách sử dụng các hoạt động nguyên tử hoặc phi nguyên tử;

FIG.17 minh họa một phương án về hoạt động xác minh/xác nhận;

FIG.18 minh họa phương pháp theo một phương án của sáng chế; và

FIG.19 minh họa chi tiết bổ sung của phương pháp chọn giữa các hoạt động xác minh nguyên tử và phi nguyên tử và kiểm tra bận.

Mô tả chi tiết sáng chế

Trong phần mô tả tiếp theo, nhằm mục đích giải thích, nhiều nội dung chi tiết cụ thể

được nêu ra để giúp hiểu toàn diện các phương án của sáng chế được mô tả dưới đây. Tuy nhiên, hiển nhiên với người có hiểu biết trung bình trong lĩnh vực kỹ thuật này rằng các phương án của sáng chế có thể được thực hiện mà không cần một số trong số các nội dung chi tiết cụ thể này. Trong các trường hợp khác, các cấu trúc và thiết bị đã biết được thể hiện ở dạng sơ đồ khối để tránh việc làm tối nghĩa các nguyên lý theo các phương án của sáng chế.

Kiến trúc bộ xử lý làm ví dụ,

Các định dạng lệnh, và các kiểu dữ liệu

Tập hợp lệnh bao gồm một hoặc nhiều định dạng lệnh. Định dạng lệnh đã cho xác định các trường khác nhau (số lượng bit, vị trí các bit) để chỉ định, trong số các nội dung khác, hoạt động cần được thực hiện (mã hoạt động) và (các) toán hạng mà hoạt động cần được thực hiện. Một số định dạng lệnh được chia nhỏ hơn qua định nghĩa của các mẫu (hoặc các định dạng con) lệnh. Ví dụ, các mẫu lệnh của định dạng lệnh đã cho có thể được xác định có các tập hợp con các trường của định dạng lệnh khác nhau (các trường được bao gồm thường theo cùng thứ tự, nhưng ít nhất một số có các vị trí bit khác nhau vì có ít trường hơn được bao gồm) và/hoặc được xác định có trường đã cho được hiểu theo cách khác. Do đó, mỗi lệnh của ISA được thể hiện bằng cách sử dụng định dạng lệnh đã cho (và, nếu được xác định, trong một trong số các mẫu lệnh đã cho của định dạng lệnh này) và bao gồm các trường để chỉ định hoạt động và (các) toán hạng. Ví dụ, lệnh ADD làm ví dụ có mã hoạt động cụ thể và định dạng lệnh bao gồm trường mã hoạt động để chỉ định các trường mã hoạt động và toán hạng để chọn các toán hạng (nguồn1/đích và nguồn2); và sự xuất hiện của lệnh ADD này trong luồng lệnh sẽ có các nội dung cụ thể trong các trường toán hạng chọn các toán hạng cụ thể.

Các phương án của (các) lệnh được mô tả trong bản mô tả này có thể được thể hiện ở các định dạng khác nhau. Ngoài ra, các hệ thống, các kiến trúc, và các đường truyền bằng ống làm ví dụ được mô tả chi tiết dưới đây. Các phương án của (các) lệnh có thể được thực thi trên các hệ thống, các kiến trúc, và các đường truyền bằng ống như vậy, nhưng không bị giới hạn ở nội dung chi tiết này.

Định dạng lệnh tiện lợi vector chung

Định dạng lệnh tiện lợi vector là định dạng lệnh được làm cho thích hợp với các lệnh vector (ví dụ, có các trường nhất định dành riêng cho các phép toán vector). Mặc dù các phương án được mô tả trong đó cả các phép toán vector và vô hướng được hỗ trợ qua định dạng lệnh tiện lợi vector, nhưng các phương án thay thế chỉ sử dụng các phép toán vector định dạng lệnh tiện lợi vector.

FIG.1A-FIG.1B là các sơ đồ khối minh họa định dạng lệnh tiện lợi vector chung và các mẫu lệnh của chúng theo các phương án của sáng chế. FIG.1A là sơ đồ khối minh họa định dạng lệnh tiện lợi vector chung và các mẫu lệnh lớp A của chúng theo các phương án của sáng chế; trong khi FIG.1B là sơ đồ khối minh họa định dạng lệnh tiện lợi vector chung và các mẫu lệnh lớp B của chúng theo các phương án của sáng chế. Cụ thể là, định dạng lệnh tiện lợi vector chung 100 được xác định là các mẫu lệnh lớp A và lớp B, cả hai đều không bao gồm các mẫu lệnh không truy nhập bộ nhớ 105 và các mẫu lệnh truy nhập bộ nhớ 120. Thuật ngữ chung trong ngữ cảnh của định dạng lệnh tiện lợi vector đề cập đến định dạng lệnh không bị ràng buộc với tập hợp lệnh cụ thể bất kỳ.

Mặc dù các phương án của sáng chế sẽ được mô tả trong đó định dạng lệnh tiện lợi vector hỗ trợ: chiều dài (hoặc kích thước) toán hạng vector 64 byte với chiều rộng (hoặc kích thước) phần tử dữ liệu 32 bit (4 byte) hoặc 64 bit (8 byte) (và do đó, vector 64 byte gồm 16 phần tử kích thước hai từ hoặc theo cách khác, 8 phần tử kích thước bốn từ); chiều dài (hoặc kích thước) toán hạng vector 64 byte với chiều rộng (hoặc kích thước) phần tử dữ liệu 16 bit (2 byte) hoặc 8 bit (1 byte); chiều dài (hoặc kích thước) toán hạng vector 32 byte với chiều rộng phần tử dữ liệu 32 bit (4 byte), 64 bit (8 byte), 16 bit (2 byte) hoặc 8 bit (1 byte); và chiều dài (hoặc kích thước) toán hạng vector 16 byte với chiều rộng (hoặc kích thước) phần tử dữ liệu 32 bit (4 byte), 64 bit (8 byte), 16 bit (2 byte) hoặc 8 bit (1 byte); các phương án thay thế có thể hỗ trợ các kích thước toán hạng vector lớn hơn, nhỏ hơn và/hoặc khác nhau (ví dụ, toán hạng vector 256 byte) với các độ rộng phần tử dữ liệu lớn hơn, nhỏ hơn hoặc khác nhau (ví dụ, các chiều rộng phần tử dữ liệu 128 bit (16 byte)).

Các mẫu lệnh lớp A trên FIG.1A bao gồm: 1) trong các mẫu lệnh không truy nhập

bộ nhớ 105 được thể hiện là mẫu lệnh không truy nhập bộ nhớ, hoạt động kiểu điều khiển làm tròn đầy đủ 110 và mẫu lệnh không truy nhập bộ nhớ, hoạt động kiểu biến đổi dữ liệu 115; và 2) trong các mẫu lệnh truy nhập bộ nhớ 120 được thể hiện là mẫu lệnh truy nhập bộ nhớ, theo thời gian 125 và mẫu lệnh truy nhập bộ nhớ, không theo thời gian 130. Các mẫu lệnh lớp B trên FIG.1B bao gồm: 1) trong các mẫu lệnh không truy nhập bộ nhớ 105 được hiển thị là mẫu lệnh không truy nhập bộ nhớ, điều khiển mặt nạ ghi, hoạt động kiểu điều khiển làm tròn một phần 112 và mẫu lệnh không truy nhập bộ nhớ, điều khiển mặt nạ ghi, hoạt động kiểu vsize 117; và 2) trong các mẫu lệnh truy nhập bộ nhớ 120 được thể hiện là mẫu lệnh truy nhập bộ nhớ, điều khiển mặt nạ ghi 127.

Định dạng lệnh tiện lợi vector chung 100 bao gồm các trường sau được liệt kê dưới đây theo thứ tự được minh họa trên FIG.1A-FIG.1B.

Trường định dạng 140 – giá trị cụ thể (giá trị mã định dạng định dạng lệnh) trong trường này nhận dạng duy nhất định dạng lệnh tiện lợi vector, và do đó xuất hiện các lệnh ở các định dạng lệnh tiện lợi vector trong các luồng lệnh. Như vậy, trường này là tùy chọn có nghĩa là nó không cần thiết cho tập hợp lệnh chỉ có định dạng lệnh tiện lợi vector chung.

Trường hoạt động cơ sở 142 – nội dung của nó phân biệt các hoạt động cơ sở khác nhau.

Trường chỉ số thanh ghi 144 – nội dung của nó, trực tiếp hoặc thông qua việc tạo địa chỉ, chỉ định vị trí của các toán hạng nguồn và đích, có thể là chúng trong thanh ghi hoặc trong bộ nhớ. Chúng bao gồm đủ số lượng bit để chọn N thanh ghi từ PxQ (ví dụ 32x512, 16x128, 32x1024, 64x1024) tệp thanh ghi. Mặc dù theo một phương án N có thể lên đến ba nguồn và một thanh ghi đích, nhưng các phương án thay thế có thể hỗ trợ nhiều hoặc ít nguồn và thanh ghi đích hơn (ví dụ, có thể hỗ trợ lên đến hai nguồn trong đó một trong số các nguồn này cũng hoạt động như đích, có thể hỗ trợ lên đến ba nguồn trong đó một trong số các nguồn này cũng hoạt động như đích, có thể hỗ trợ lên đến hai nguồn và một đích).

Trường biến đổi 146 – nội dung của nó phân biệt sự xuất hiện của các lệnh ở định dạng lệnh vector chung chỉ định truy nhập bộ nhớ từ những lệnh không có; tức là, giữa các mẫu lệnh không truy nhập bộ nhớ 105 và các mẫu lệnh truy nhập bộ nhớ 120. Các hoạt

động truy nhập bộ nhớ đọc và/hoặc ghi vào hệ thống phân cấp bộ nhớ (trong một số trường hợp chỉ định các địa chỉ nguồn và/hoặc đích sử dụng các giá trị trong các thanh ghi), trong khi các hoạt động không truy nhập bộ nhớ thì không (ví dụ, nguồn và các đích là các thanh ghi). Mặc dù theo một phương án, trường này cũng chọn giữa ba cách khác nhau để thực hiện các tính toán địa chỉ bộ nhớ, nhưng các phương án thay thế có thể hỗ trợ nhiều, ít cách hơn, hoặc các cách khác nhau để thực hiện các tính toán địa chỉ bộ nhớ.

Trường hoạt động tăng cường 150 – nội dung của nó phân biệt một trong số các hoạt động khác nhau cần được thực hiện ngoài hoạt động cơ sở. Trường này là ngữ cảnh cụ thể. Theo một phương án của sáng chế, trường này được chia thành trường lớp 168, trường alpha 152, và trường beta 154. Trường hoạt động tăng cường 150 cho phép các nhóm chung của các hoạt động được thực hiện trong một lệnh thay vì 2, 3, hoặc 4 lệnh.

Trường tỷ lệ 160 – nội dung của nó cho phép chia tỷ lệ nội dung của trường chỉ số để tạo địa chỉ bộ nhớ (ví dụ, để tạo địa chỉ sử dụng $2^{\text{tỷ lệ}} * \text{chỉ số} + \text{cơ sở}$).

Trường độ dịch 162A – nội dung của nó được sử dụng là một phần để tạo địa chỉ bộ nhớ (ví dụ, để tạo địa chỉ sử dụng $2^{\text{tỷ lệ}} * \text{chỉ số} + \text{cơ sở} + \text{độ dịch}$).

Trường hệ số độ dịch 162B (lưu ý rằng vị trí liền kề của trường độ dịch 162A trực tiếp trên trường hệ số độ dịch 162B chỉ báo trường nào được sử dụng) – nội dung của nó được sử dụng là một phần để tạo địa chỉ; nó chỉ định hệ số độ dịch cần được chia tỷ lệ theo kích thước của truy nhập bộ nhớ (N) – trong đó N là số lượng byte trong truy nhập bộ nhớ (ví dụ, để tạo địa chỉ sử dụng $2^{\text{tỷ lệ}} * \text{chỉ số} + \text{cơ sở} + \text{độ dịch được chia tỷ lệ}$). Các bit bậc thấp dư được bỏ qua và do đó, nội dung của trường hệ số độ dịch được nhân với kích thước tổng các toán hạng bộ nhớ (N) để tạo ra độ dịch cuối cùng được sử dụng trong quá trình tính địa chỉ hiệu dụng. Giá trị của N được xác định bởi phần cứng bộ xử lý ở thời gian chạy dựa vào trường mã hoạt động đầy đủ 174 (được mô tả sau đây) và trường thao tác dữ liệu 154C. Trường độ dịch 162A và trường hệ số độ dịch 162B là tùy chọn có nghĩa là chúng không được sử dụng cho các mẫu lệnh không truy nhập bộ nhớ 105 và/hoặc các phương án khác nhau có thể thực hiện chỉ một hoặc không thực hiện trường nào trong hai trường này.

Trường độ rộng phần tử dữ liệu 164 – nội dung của nó phân biệt một trong số số

lượng chiều rộng phần tử dữ liệu được sử dụng (theo một số phương án cho tất cả các lệnh; theo các phương án khác chỉ cho một số trong số các lệnh). Trường này là tùy chọn có nghĩa là không cần thiết nếu chỉ một chiều rộng phần tử dữ liệu được hỗ trợ và/hoặc các chiều rộng phần tử dữ liệu được hỗ trợ bằng cách sử dụng một số khía cạnh của các mã hoạt động.

Trường mặt nạ ghi 170 – nội dung của nó điều khiển, trên cơ sở mỗi vị trí phần tử dữ liệu, xem liệu vị trí phần tử dữ liệu đó trong toán hạng vector đích có phản ánh kết quả của hoạt động cơ sở và hoạt động tăng cường hay không. Các mẫu lệnh lớp A hỗ trợ gộp-ghi mặt nạ, trong khi các mẫu lệnh lớp B hỗ trợ cả gộp- và về không-ghi mặt nạ. Khi gộp, các mặt nạ vector cho phép tập hợp phần tử bất kỳ trong đích được bảo vệ khỏi các cập nhật trong quá trình thực thi hoạt động bất kỳ (được chỉ định bởi hoạt động cơ sở và hoạt động tăng cường); theo một phương án khác, lưu trữ giá trị cũ của mỗi phần tử của đích trong đó bit mặt nạ tương ứng bằng 0. Ngược lại, khi các mặt nạ vector về không cho phép tập hợp phần tử bất kỳ trong đích cần được đưa về không trong quá trình thực thi hoạt động bất kỳ (được chỉ định bởi hoạt động cơ sở và hoạt động tăng cường); theo một phương án, phần tử của đích được thiết lập bằng 0 khi bit mặt nạ tương ứng có giá trị bằng 0. Tập hợp con của chức năng này là khả năng điều khiển chiều dài vector của hoạt động được thực hiện (tức là, khoảng thời gian của các phần tử được sửa đổi, từ phần tử thứ nhất đến phần tử cuối cùng); tuy nhiên, không nhất thiết là các phần tử được sửa đổi phải liên tiếp. Do đó, trường mặt nạ ghi 170 cho phép các phép toán vector một phần, bao gồm tải, lưu trữ, số học, logic, v.v.. Mặc dù các phương án của sáng chế được mô tả trong đó nội dung của trường mặt nạ ghi 170 chọn một trong số các thanh ghi mặt nạ ghi chứa mặt nạ ghi được sử dụng (và do đó nội dung của trường mặt nạ ghi 170 gián tiếp nhận dạng mặt nạ được thực hiện), nhưng các phương án thay thế hoặc bổ sung cho phép nội dung của trường mặt nạ ghi 170 trực tiếp chỉ định mặt nạ được thực hiện.

Trường tức thời 172 – nội dung của nó cho phép mô tả sự tức thời. Trường này là tùy chọn có nghĩa là không có mặt theo phương án thực hiện của định dạng lệnh tiện lợi vector chung không hỗ trợ sự tức thời và không có mặt trong các lệnh không được sử dụng tức thời.

Trường lớp 168 – nội dung của nó phân biệt giữa các lớp lệnh khác nhau. Tham

chiều đến FIG.1A-FIG.1B, các nội dung của trường này chọn giữa các lệnh lớp A và lớp B. Trên FIG.1A-FIG.1B, các góc vuông được làm tròn được sử dụng để chỉ báo giá trị cụ thể có trong một trường (ví dụ, lớp A 168A và lớp B 168B cho trường lớp 168 lần lượt trên FIG.1A-FIG.1B).

Các mẫu lệnh của lớp A

Trong trường hợp các mẫu lệnh không truy nhập bộ nhớ 105 của lớp A, trường alpha 152 được hiểu là trường RS 152A, có nội dung phân biệt với một trong số các kiểu hoạt động tăng cường khác nhau cần được thực hiện (ví dụ, làm tròn 152A.1 và biến đổi dữ liệu 152A.2 lần lượt được chỉ định cho các mẫu lệnh không truy nhập bộ nhớ, hoạt động kiểu làm tròn 110 và không truy nhập bộ nhớ, hoạt động kiểu biến đổi dữ liệu 115), trong khi trường beta 154 phân biệt hoạt động nào trong số các hoạt động của kiểu được chỉ định được thực hiện. Trong các mẫu lệnh không truy nhập bộ nhớ 105, trường tỷ lệ 160, trường độ dịch 162A, và trường tỷ lệ độ dịch 162B không có mặt.

Các mẫu lệnh truy nhập không bộ nhớ – Hoạt động kiểu điều khiển làm tròn đầy đủ

Trong mẫu lệnh không truy nhập bộ nhớ hoạt động kiểu điều khiển làm tròn đầy đủ 110, trường beta 154 được hiểu là trường điều khiển làm tròn 154A, có (các) nội dung cung cấp khả năng làm tròn tĩnh. Mặc dù theo các phương án được mô tả của sáng chế trường điều khiển làm tròn 154A bao gồm trường loại trừ tất cả các ngoại lệ dấu phẩy động (SAE) 156 và trường điều khiển hoạt động làm tròn 158, nhưng các phương án thay thế có thể hỗ trợ có thể mã hóa cả các khái niệm này thành cùng trường hoặc chỉ có khái niệm/trường nào trong số các khái niệm/trường này (ví dụ, có thể chỉ có trường điều khiển hoạt động làm tròn 158).

Trường SAE 156 – nội dung của nó phân biệt xem liệu có tắt báo cáo sự kiện ngoại lệ hay không; khi nội dung của trường SAE 156 chỉ báo việc loại trừ được bật, thì lệnh nhất định không báo cáo loại trừ ngoại lệ dấu phẩy động bất kỳ và không đưa ra trình xử lý ngoại lệ dấu phẩy động bất kỳ.

Trường điều khiển hoạt động làm tròn 158 – nội dung của nó phân biệt một trong số

nhóm các hoạt động làm tròn để thực hiện (ví dụ, làm tròn lên, làm tròn xuống, làm tròn về không và làm tròn đến gần nhất). Do đó, trường điều khiển hoạt động làm tròn 158 cho phép thay đổi chế độ làm tròn trên cơ sở mỗi lệnh. Theo một phương án của sáng chế trong đó bộ xử lý bao gồm thanh ghi điều khiển để chỉ định các chế độ làm tròn, nội dung của trường điều khiển hoạt động làm tròn 150 ghi đè giá trị thanh ghi.

Các mẫu lệnh không truy nhập bộ nhớ – Hoạt động kiểu biến đổi dữ liệu

Trong mẫu lệnh không truy nhập bộ nhớ hoạt động kiểu biến đổi dữ liệu 115, trường beta 154 được hiểu là trường biến đổi dữ liệu 154B, có nội dung phân biệt một trong số số lượng biến đổi dữ liệu sẽ được thực hiện (ví dụ, không biến đổi dữ liệu, xáo trộn, phát rộng).

Trong trường hợp mẫu lệnh truy nhập bộ nhớ 120 của lớp A, trường alpha 152 được hiểu là trường gợi ý thu hồi 152B, có nội dung phân biệt một trong số các gợi ý thu hồi sẽ được sử dụng (trên FIG.1A, theo thời gian 152B.1 và không theo thời gian 152B.2 lần lượt được chỉ định cho mẫu lệnh truy nhập bộ nhớ, theo thời gian 125 và mẫu lệnh truy nhập bộ nhớ, không theo thời gian 130), trong khi trường beta 154 được hiểu là trường thao tác dữ liệu 154C, có nội dung phân biệt một trong số số lượng hoạt động thao tác dữ liệu (cũng được biết là các hàm nguyên thủy) sẽ được thực hiện (ví dụ, không thao tác; phát rộng; chuyển đổi lên nguồn; và chuyển đổi xuống đích). Các mẫu lệnh truy nhập bộ nhớ 120 bao gồm trường tỷ lệ 160, và tùy chọn bao gồm trường độ dịch 162A hoặc trường tỷ lệ dịch chuyển 162B.

Các lệnh bộ nhớ vector thực hiện tải vector từ và lưu trữ vector vào bộ nhớ, có hỗ trợ chuyển đổi. Là các lệnh vector thông thường, các lệnh bộ nhớ vector truyền dữ liệu từ/đến bộ nhớ theo kiểu từng phần tử dữ liệu, với các phần tử thực sự được truyền được quyết định bởi các nội dung của mặt nạ vector được chọn làm mặt nạ ghi.

Các mẫu lệnh truy nhập bộ nhớ – Theo thời gian

Dữ liệu theo thời gian là dữ liệu có khả năng được tái sử dụng đủ sớm để hưởng lợi từ bộ đệm. Tuy nhiên, đây là một gợi ý, và các bộ xử lý khác nhau có thể thực hiện nó theo các cách khác nhau, bao gồm bỏ qua toàn bộ gợi ý.

Các mẫu lệnh truy nhập bộ nhớ – Không theo thời gian

Dữ liệu không theo thời gian là dữ liệu có khả năng được tái sử dụng đủ sớm để hưởng lợi từ bộ đệm trong bộ đệm mức thứ nhất và phải được ưu tiên loại bỏ. Tuy nhiên, đây là một gợi ý, và các bộ xử lý khác nhau có thể thực hiện nó theo các cách khác nhau, bao gồm bỏ qua toàn bộ gợi ý.

Các mẫu lệnh của lớp B

Trong trường hợp các mẫu lệnh của lớp B, trường alpha 152 được hiểu là trường điều khiển mặt nạ ghi (Z) 152C, có nội dung phân biệt xem liệu mặt nạ ghi được điều khiển bởi trường mặt nạ ghi 170 nên gộp hay bằng 0.

Trong trường hợp các mẫu lệnh không truy nhập bộ nhớ 105 của lớp B, một phần của trường beta 154 được hiểu là trường RL 157A, có nội dung phân biệt một trong số các kiểu hoạt động tăng cường khác nhau sẽ được thực hiện (ví dụ, làm tròn 157A.1 và chiều dài vector (VSIZE) 157A.2 lần lượt được chỉ định cho mẫu lệnh không truy nhập bộ nhớ, điều khiển mặt nạ ghi, hoạt động kiểu điều khiển làm tròn một phần 112 và mẫu lệnh không truy nhập bộ nhớ, điều khiển mặt nạ ghi, hoạt động kiểu VSIZE 117), trong khi phần còn lại của trường beta 154 phân biệt các hoạt động trong số kiểu được chỉ định sẽ được thực hiện. Trong mẫu lệnh không truy nhập bộ nhớ 105, trường tỷ lệ 160, trường độ dịch 162A, và trường tỷ lệ độ dịch 162B không có mặt.

Trong mẫu lệnh không truy nhập bộ nhớ, điều khiển mặt nạ ghi, hoạt động kiểu điều khiển làm tròn một phần 110, phần còn lại của trường beta 154 được hiểu là trường hoạt động làm tròn 159A và báo cáo sự kiện ngoại lệ được tắt (lệnh nhất định không báo cáo kiểu cờ ngoại lệ dấu phẩy động bất kỳ và không đưa ra bất kỳ trình xử lý ngoại lệ dấu phẩy động nào).

Trường điều khiển hoạt động làm tròn 159A – như trường điều khiển hoạt động làm tròn 158, nội dung của nó phân biệt một trong số nhóm các hoạt động làm tròn để thực hiện (ví dụ, làm tròn lên, làm tròn xuống, làm tròn về không và làm tròn đến gần nhất). Do đó, trường điều khiển hoạt động làm tròn 159A cho phép thay đổi chế độ làm tròn trên cơ sở

mỗi lệnh. Theo một phương án của sáng chế trong đó bộ xử lý bao gồm thanh ghi điều khiển để chỉ định các chế độ làm tròn, nội dung của trường điều khiển hoạt động làm tròn 150 ghi đề giá trị thanh ghi.

Trong mẫu lệnh không truy nhập bộ nhớ, điều khiển mặt nạ ghi, hoạt động kiểu VSIZE 117, phần còn lại của trường beta 154 được hiểu là trường chiều dài vector 159B, có nội dung phân biệt một trong số số lượng chiều dài vector dữ liệu sẽ được thực hiện trên (ví dụ, 128, 256, hoặc 512 byte).

Trong trường hợp mẫu lệnh truy nhập 120 của lớp B, một phần của trường beta 154 được hiểu là trường phát rộng 157B, có nội dung phân biệt xem liệu hoạt động thao tác điều khiển kiểu phát rộng có được thực hiện hay không, trong khi phần còn lại của trường beta 154 được hiểu là trường chiều dài vector 159B. Các mẫu lệnh truy nhập bộ nhớ 120 bao gồm trường tỷ lệ 160, và một cách tùy ý trường độ dịch 162A hoặc trường tỷ lệ dịch chuyển 162B.

Đối với định dạng lệnh tiện lợi vector chung 100, trường mã hoạt động đầy đủ 174 được thể hiện bao gồm trường định dạng 140, trường hoạt động cơ sở 142, và trường chiều rộng phần tử dữ liệu 164. Trong khi một phương án được thể hiện trong đó trường mã hoạt động đầy đủ 174 bao gồm tất cả trong số các trường này, trường mã hoạt động đầy đủ 174 bao gồm ít hơn tất cả trong số các trường này theo các phương án không hỗ trợ tất cả các trường này. Trường mã hoạt động đầy đủ 174 tạo ra mã hoạt động (opcode).

Trường hoạt động tăng cường 150, trường chiều rộng phần tử dữ liệu 164, và trường mặt nạ ghi 170 cho phép các dấu hiệu này sẽ được chỉ định trên cơ sở mỗi lệnh ở định dạng lệnh tiện lợi vector chung.

Sự kết hợp của trường mặt nạ ghi và trường chiều rộng phần tử dữ liệu tạo ra các lệnh kiểu trong đó chúng cho phép mặt nạ sẽ được áp dụng dựa vào các chiều rộng phần tử dữ liệu khác nhau.

Các mẫu lệnh khác nhau được tìm thấy trong lớp A và lớp B đều có lợi trong các tình huống khác nhau. Theo một số phương án của sáng chế, các bộ xử lý khác nhau hoặc

các lỗi khác nhau trong bộ xử lý có thể hỗ trợ chỉ lớp A, chỉ lớp B, hoặc cả hai lớp. Ví dụ, lỗi không theo thứ tự đa năng hiệu suất cao dành cho điện toán đa năng có thể chỉ hỗ trợ lớp B, lỗi chủ yếu dành cho đồ họa và/hoặc tính toán khoa học (thông lượng) có thể chỉ hỗ trợ lớp A, và lỗi dành cho cả có thể hỗ trợ cả (tất nhiên, lỗi có một số kết hợp của các mẫu và các lệnh từ cả hai lớp nhưng không phải tất cả các mẫu và các lệnh từ cả hai lớp ở trong phạm vi của sáng chế). Ngoài ra, một bộ xử lý có thể bao gồm nhiều lỗi, tất cả đều hỗ trợ cùng lớp hoặc trong đó các lỗi khác nhau hỗ trợ lớp khác nhau. Ví dụ, trong bộ xử lý có các lỗi đồ họa và đa năng riêng biệt, một trong số các lỗi đồ họa chủ yếu dùng cho điện toán đồ họa và/hoặc khoa học có thể chỉ hỗ trợ lớp A, trong khi một hoặc nhiều trong số các lỗi đa năng có thể là các lỗi đa năng hiệu suất cao với thực thi không theo thứ tự và đổi tên thanh ghi dùng cho điện toán đa năng chỉ hỗ trợ lớp B. Bộ xử lý khác không có lỗi đồ họa riêng biệt, có thể bao gồm nhiều hơn một lỗi theo thứ tự hoặc không theo thứ tự đa năng hỗ trợ cả lớp A và lớp B. Tất nhiên, các dấu hiệu từ một lớp cũng có thể được thực hiện trong lớp khác theo các phương án khác nhau của sáng chế. Các chương trình được ghi ở ngôn ngữ mức cao sẽ được đặt (ví dụ, biên dịch đúng lúc hoặc biên dịch tĩnh) thành nhiều dạng thực thi khác nhau, bao gồm: 1) dạng chỉ có các lệnh của (các) lớp được hỗ trợ bởi bộ xử lý đích để thực thi; hoặc 2) dạng có các thường trình thay thế được ghi sử dụng các sự kết hợp khác nhau của các lệnh của tất cả các lớp và có mã dòng điều khiển chọn các thường trình để thực thi dựa vào các lệnh được hỗ trợ bởi bộ xử lý hiện đang thực thi mã này.

Định dạng lệnh VEX

Mã hóa VEX cho phép các lệnh có nhiều hơn hai toán hạng, và cho phép các thanh ghi vector SIMD dài hơn 28 bit. Việc sử dụng tiền tố VEX tạo ra cú pháp ba (nhiều hơn ba) toán hạng. Ví dụ, các lệnh hai toán hạng trước đó đã thực hiện các phép toán chẳng hạn như $A = A + B$, ghi đề toán hạng nguồn. Việc sử dụng tiền tố VEX cho phép các toán hạng thực hiện các phép toán không xóa chẳng hạn như $A = B + C$.

FIG.2A minh họa định dạng lệnh AVX làm ví dụ bao gồm tiền tố VEX 202, trường mã hoạt động thực 230, Mod R/M byte 240, SIB byte 250, trường độ dịch 262, và IMM8 272. FIG.2B minh họa các trường từ FIG.2A tạo thành trường mã hoạt động đầy đủ 274 và trường hoạt động cơ bản 241. FIG.2C minh họa các trường từ FIG.2A tạo ra trường chỉ số

thanh ghi 244.

Tiền tố VEX (các byte 0-2) 202 được mã hóa ở dạng ba byte. Byte thứ nhất là trường định dạng 290 (Byte VEX 0, các bit [7:0]), chứa giá trị byte C4 rõ ràng (giá trị duy nhất được sử dụng để phân biệt định dạng lệnh C4). Các bit thứ hai-thứ ba (các byte VEX 1-2) bao gồm số lượng trường bit cung cấp khả năng cụ thể. Cụ thể là, trường REX 205 (Byte VEX 1, các bit [7-5]) gồm trường bit VEX.R (Byte VEX 1, bit [7] – R), trường bit VEX.X (byte VEX 1, bit [6] – X), và trường bit VEX.B (byte VEX 1, bit[5] – B). Các trường khác của các lệnh mã hóa ba bit phía dưới của các chỉ số thanh ghi như đã biết trong tình trạng kỹ thuật (rrr, xxx, và bbb), để Rrrr, Xxxx, và Bbbb có thể được tạo thành bằng cách bổ sung VEX.R, VEX.X, và VEX.B. Trường ánh xạ mã hoạt động 215 (byte VEX 1, các bit [4:0] – mmmmm) bao gồm nội dung để mã hóa byte mã hoạt động hàng đầu ngụ ý. Trường W 264 (byte VEX 2, bit [7] – W) - được biểu diễn bằng ký hiệu VEX.W, và cung cấp các chức năng khác nhau phụ thuộc vào lệnh. Vai trò của VEX.vvvv 220 (Byte VEX 2, các bit [6:3]-vvvv) có thể bao gồm: 1) VEX.vvvv mã hóa toán hạng thanh ghi nguồn thứ nhất, được chỉ định ở dạng đảo ngược (bù 1s) và hợp lệ cho các lệnh có 2 hoặc nhiều hơn 2 toán hạng nguồn; 2) VEX.vvvv mã hóa toán hạng thanh ghi đích, được chỉ định ở dạng bù 1s cho các dịch chuyển vector nhất định; hoặc 3) VEX.vvvv không mã hóa toán hạng bất kỳ, trường này được dành riêng và nên chứa 1111b. Nếu trường kích thước VEX.L 268 (byte VEX 2, bit [2]-L) = 0, thì chỉ báo vector 28 bit; nếu VEX.L = 1, thì chỉ báo vector 256 bit. Trường mã hóa tiền tố 225 (byte VEX 2, các bit [1:0]-pp) cấp các bit bổ sung cho trường hoạt động cơ sở 241.

Trường mã hoạt động thực 230 (Byte 3) cũng được biết là byte mã hoạt động. Một phần của mã hoạt động được chỉ định trong trường này.

Trường MOD R/M 240 (Byte 4) bao gồm trường MOD 242 (các bit [7-6]), trường Reg 244 (các bit [5-3]), và trường R/M 246 (các bit [2-0]). Vai trò của trường Reg 244 có thể bao gồm: mã hóa toán hạng thanh ghi đích hoặc toán hạng thanh ghi nguồn (rrr của Rrrr), hoặc được xử lý ở dạng mở rộng mã hoạt động và không được sử dụng để mã hóa toán hạng lệnh bất kỳ. Vai trò của trường R/M 246 có thể bao gồm: mã hóa toán hạng lệnh tham chiếu địa chỉ bộ nhớ, hoặc mã hóa toán hạng thanh ghi đích hoặc toán hạng thanh ghi

nguồn.

Tỷ lệ, chỉ số, cơ sở (Scale, Index, Base, SIB) - Nội dung của trường tỷ lệ 250 (Byte 5) bao gồm SS252 (các bit [7-6]), được sử dụng để tạo địa chỉ bộ nhớ. Các nội dung của SIB.xxx 254 (các bit [5-3]) và SIB.bbb 256 (các bit [2-0]) được đề cập đến trước đây liên quan đến các chỉ số thanh ghi Xxxx và Bbbb.

Trường độ dịch 262 và trường tức thời (IMM8) 272 chứa dữ liệu.

Kiến trúc thanh ghi làm ví dụ

FIG.3 là sơ đồ khối của kiến trúc thanh ghi 300 theo một phương án của sáng chế. Theo phương án được minh họa, có 32 thanh ghi vector 310 có chiều rộng 512 bit; các thanh ghi này được gọi là zmm0 đến zmm31. 256 bit thứ tự thấp hơn của các thanh ghi 6 zmm phía dưới được phủ trên các thanh ghi ymm0-15. 128 bit thứ tự thấp hơn của các thanh ghi 6 zmm phía dưới (128 bit thứ tự thấp hơn của các thanh ghi ymm) được phủ trên các thanh ghi xmm0-15.

Các thanh ghi đa năng 325 - theo phương án được minh họa, có mười sáu thanh ghi đa năng 64-bit được sử dụng cùng với các chế độ địa chỉ x86 hiện có để giải quyết các toán hạng bộ nhớ. Các thanh ghi này được gọi bằng các tên RAX, RBX, RCX, RDX, RBP, RSI, RDI, RSP, và R8 qua R15.

Tập thanh ghi ngăn xếp dấu phẩy động vô hướng (ngăn xếp x87) 345, mà được viết tắt là tập thanh ghi phẳng số nguyên được đóng gói MMX 350 - theo phương án được minh họa, ngăn xếp x87 là ngăn xếp tám phần tử được sử dụng để thực hiện các hoạt động dấu phẩy động vô hướng đối với dữ liệu dấu phẩy động 32/64/80-bit sử dụng mở rộng tập hợp lệnh x87; trong khi các thanh ghi MMX được sử dụng để thực hiện các hoạt động đối với dữ liệu số nguyên được đóng gói 64-bit, cũng như để giữ các toán hạng cho một số hoạt động được thực hiện giữa các thanh ghi MMX và XMM.

Các phương án thay thế của sáng chế có thể sử dụng các thanh ghi rộng hơn hoặc hẹp hơn. Ngoài ra, các phương án thay thế của sáng chế có thể sử dụng nhiều hơn, ít hơn, hoặc các tập thanh ghi và các thanh ghi khác nhau.

Các kiến trúc lõi làm ví dụ, các bộ xử lý, và các kiến trúc máy tính

Các lõi bộ xử lý có thể được thực hiện theo các cách khác nhau, cho các mục đích khác nhau, và trong các bộ xử lý khác nhau. Ví dụ, các phương án thực hiện của các lõi này có thể bao gồm: 1) lõi theo thứ tự đa năng dùng cho điện toán đa năng; 2) lõi không theo thứ tự đa năng hiệu suất cao dùng cho điện toán đa năng; 3) lõi chuyên dụng chủ yếu dùng cho đồ họa và/hoặc tính toán khoa học (thông lượng). Các phương án thực hiện của các bộ xử lý khác nhau có thể bao gồm: 1) CPU bao gồm một hoặc nhiều lõi theo thứ tự đa năng dùng cho điện toán đa năng và/hoặc một hoặc nhiều lõi không theo thứ tự đa năng dùng cho điện toán đa năng; và 2) bộ đồng xử lý bao gồm một hoặc nhiều lõi chuyên dụng chủ yếu dùng cho đồ họa và/hoặc khoa học (thông lượng). Các bộ xử lý khác nhau như vậy dẫn đến các kiến trúc hệ thống máy tính khác nhau, có thể bao gồm: 1) bộ đồng xử lý trên chip tách biệt khỏi CPU; 2) bộ đồng xử lý trên khuôn riêng biệt trong cùng gói với CPU; 3) bộ đồng xử lý trên cùng khuôn với CPU (trong trường hợp, bộ đồng xử lý này đôi khi được gọi là mạch logic chuyên dụng, chẳng hạn như mạch logic đồ họa và/hoặc khoa học (thông lượng) được tích hợp, hoặc ở dạng các lõi chuyên dụng); và 4) hệ thống trên chip có thể bao gồm trên cùng khuôn CPU được mô tả (đôi khi được gọi là (các) lõi ứng dụng hoặc (các) bộ xử lý ứng dụng), bộ đồng xử lý được mô tả ở trên, và chức năng bổ sung. Các kiến trúc lõi làm ví dụ được mô tả tiếp theo, được tuân theo bởi các phần mô tả của các bộ xử lý và các kiến trúc máy tính làm ví dụ. Nội dung chi tiết trong bản mô tả này là các mạch (bộ phận) bao gồm các lõi, các bộ xử lý làm ví dụ, v.v..

Các kiến trúc lõi làm ví dụ

FIG.4A là sơ đồ khối minh họa cả đường truyền bằng ống theo thứ tự làm ví dụ và đường truyền bằng ống đổi tên thanh ghi, phát hành/thực thi không theo thứ tự làm ví dụ theo các phương án của sáng chế. FIG.4B là sơ đồ khối minh họa cả phương án làm ví dụ của lõi kiến trúc theo thứ tự và lõi kiến trúc đổi tên thanh ghi, phát hành/thực thi không theo thứ tự làm ví dụ nằm trong bộ xử lý theo các phương án của sáng chế. Các hộp đường nét liền trên FIG.4A-FIG.4B minh họa đường truyền bằng ống theo thứ tự và lõi theo thứ tự, trong khi các hộp đường nét đứt là bổ sung tùy chọn minh họa đường truyền bằng ống và lõi đổi tên thanh ghi, phát hành/thực thi không theo thứ tự. Cho rằng khía cạnh theo thứ tự

là tập hợp con của khía cạnh không theo thứ tự, khía cạnh không theo thứ tự sẽ được mô tả.

Trên FIG.4A, đường truyền bằng ống 400 của bộ xử lý bao gồm giai đoạn nạp 402, giai đoạn giải mã chiều dài 404, giai đoạn giải mã 406, giai đoạn cấp phát 408, giai đoạn đổi tên 410, giai đoạn lập lịch (cũng được biết là điều phối hoặc phát hành) 412, giai đoạn đọc thanh ghi/đọc bộ nhớ 414, giai đoạn thực thi 416, giai đoạn ghi lại/ghi bộ nhớ 418, giai đoạn xử lý ngoại lệ 422, và giai đoạn cam kết 424.

FIG.4B thể hiện lõi 490 của bộ xử lý bao gồm bộ phận đầu phía trước 430 được ghép nối với bộ phận công cụ thực thi 450, và đều được ghép nối với bộ phận nhớ 470. Lõi 490 có thể là lõi điện toán tập hợp lệnh giảm đi (Reduced Instruction Set Computing, RISC), lõi điện toán tập hợp lệnh phức tạp (Complex Instruction Set Computing, CISC), lõi từ lệnh rất dài (Very Long Instruction Word, VLIW), hoặc kiểu lõi lai hoặc thay thế. Là tùy chọn khác nữa, lõi 490 có thể là lõi chuyên dụng, chẳng hạn như, ví dụ, lõi mạng hoặc truyền thông, động cơ nén, lõi bộ đồng xử lý, lõi bộ phận xử lý đồ họa điện toán đa năng (General Purpose Computing Graphics Processing Unit, GPGPU), lõi đồ họa, hoặc tương tự.

Bộ phận đầu phía trước 430 bao gồm bộ phận dự đoán rẽ nhánh 432 được ghép nối với bộ phận đệm lệnh 434, được ghép nối với bộ đệm tra cứu dịch lệnh (Instruction Translation Lookaside Buffer, TLB) 436, được ghép nối với bộ nạp lệnh 438, được ghép nối với bộ phận giải mã 440. Bộ phận giải mã 440 (hoặc bộ giải mã) có thể giải mã các lệnh, và tạo ra ở dạng đầu ra một hoặc nhiều thao tác micro, các điểm mục nhập mã micro, các lệnh micro, các lệnh khác, hoặc các tín hiệu điều khiển khác, được giải mã từ, hoặc phản ánh, hoặc được suy ra từ, các lệnh ban đầu. Bộ phận giải mã 440 có thể được thực hiện bằng cách sử dụng các cơ chế khác nhau. Các ví dụ về các cơ chế thích hợp bao gồm, nhưng không bị giới hạn ở, các bảng tìm kiếm, các phương án thực hiện phần cứng, các mảng logic lập trình được (Programmable Logic Array, PLA), các bộ nhớ chỉ đọc (Read Only Memory, ROM) vi mã, v.v.. Theo một phương án, lõi 490 bao gồm ROM vi mã hoặc vật ghi khác lưu trữ vi mã cho các lệnh macro nhất định (ví dụ, trong bộ phận giải mã 440 hoặc trong bộ phận đầu phía trước 430). Bộ phận giải mã 440 được ghép nối với bộ phận đổi tên/cấp phát 452 trong bộ phận công cụ thực thi 450.

Bộ phận công cụ thực thi 450 bao gồm bộ phận đổi tên/cấp phát 452 được ghép nối với bộ phận ẩn 454 và tập hợp của một hoặc nhiều bộ phận lập lịch 456. (Các) bộ phận lập lịch 456 biểu diễn số lượng bộ lập lịch khác nhau bất kỳ, bao gồm trạm dành riêng, cửa sổ lệnh trung tâm, v.v.. (Các) bộ phận lập lịch 456 được ghép nối với (các) bộ phận (các) tệp thanh ghi vật lý 458. Mỗi trong số (các) tệp thanh ghi vật lý 458 biểu diễn một hoặc nhiều tệp thanh ghi vật lý, các tệp khác nhau lưu trữ một hoặc nhiều kiểu dữ liệu khác nhau, chẳng hạn như số nguyên vô hướng, dấu phẩy động vô hướng, số nguyên được đóng gói, dấu phẩy động được đóng gói, số nguyên vector, dấu phẩy động vector, trạng thái (ví dụ, con trỏ lệnh là địa chỉ của lệnh tiếp theo cần được thực thi), v.v.. Theo một phương án, bộ phận (các) tệp thanh ghi 458 bao gồm bộ phận các số nguyên vector và bộ phận các thanh ghi vô hướng. Các bộ phận thanh ghi này có thể cung cấp các thanh ghi vector kiến trúc, thanh ghi mặt nạ vector, và thanh ghi đa năng. (Các) bộ phận (các) tệp thanh ghi vật lý 458 được ngăn xếp lên bởi bộ phận ẩn 454 để minh họa các cách khác nhau trong đó thực thi đổi tên thanh ghi và không theo thứ tự có thể được thực hiện (ví dụ, sử dụng (các) bộ đệm sắp xếp lại và (các) tệp thanh ghi ẩn; sử dụng (các) tệp trong tương lai, (các) bộ đệm lịch sử, và (các) tệp thanh ghi ẩn; sử dụng các bản đồ thanh ghi và vùng trừ các thanh ghi; v.v.). Bộ phận ẩn 454 và (các) bộ phận (các) tệp thanh ghi 458 được ghép nối với (các) cụm thực thi 460. (Các) cụm thực thi 460 bao gồm tập hợp một hoặc nhiều bộ phận thực thi 462 và tập hợp một hoặc nhiều bộ phận truy nhập bộ nhớ 464. Các bộ phận thực thi 462 có thể thực hiện các phép toán khác nhau (ví dụ, dịch chuyển, cộng, trừ, nhân) và đối với các kiểu dữ liệu khác nhau (ví dụ, dấu phẩy động vô hướng, số nguyên được đóng gói, dấu phẩy động được đóng gói, số nguyên vector, dấu phẩy động vector). Mặc dù một số phương án có thể bao gồm số lượng bộ phận thực thi dành riêng cho các chức năng cụ thể hoặc tập hợp các chức năng, nhưng các phương án khác có thể chỉ bao gồm một bộ phận thực thi hoặc nhiều bộ phận thực thi thực hiện tất cả các chức năng. (Các) bộ phận lập lịch 456, (các) bộ phận (các) tệp thanh ghi vật lý 458, và (các) cụm thực thi 460 được thể hiện ở dạng số nhiều vì các phương án nhất định tạo ra các đường truyền bằng ống riêng biệt cho kiểu dữ liệu/hoạt động nhất định (ví dụ, đường truyền bằng ống số nguyên vô hướng, đường truyền bằng ống dấu phẩy động vô hướng/số nguyên được đóng gói/dấu phẩy động được đóng gói/số nguyên vector/dấu phẩy động vector, và/hoặc đường truyền bằng ống truy nhập bộ nhớ đều có bộ

phần lập lịch, bộ phận (các) tệp thanh ghi vật lý, và/hoặc cụm thực thi của chúng – và trong trường hợp đường truyền bằng ống truy nhập bộ nhớ riêng biệt, các phương án nhất định được thực hiện trong đó chỉ cụm thực thi của đường truyền bằng ống này có (các) bộ phận truy nhập bộ nhớ 464). Cũng cần hiểu rằng nơi mà các đường truyền bằng ống riêng biệt được sử dụng, một hoặc nhiều trong số các đường truyền bằng ống này có thể phát hành/thực thi không theo thứ tự và phần còn lại theo thứ tự.

Tập hợp các bộ phận truy nhập bộ nhớ 464 được ghép nối với bộ phận nhớ 470, bao gồm bộ phận TLB dữ liệu 472 được ghép nối với bộ phận đệm dữ liệu 474 được ghép nối với bộ phận đệm mức 2 (L2) 476. Theo một phương án làm ví dụ, các bộ phận truy nhập bộ nhớ 464 có thể bao gồm bộ phận tải, bộ phận địa chỉ lưu trữ, và bộ phận dữ liệu lưu trữ, mỗi trong số chúng được ghép nối với bộ phận TLB dữ liệu 472 trong bộ phận nhớ 470. Bộ phận đệm lệnh 434 còn được ghép nối với bộ phận đệm mức 2 (L2) 476 trong bộ phận nhớ 470. Bộ phận đệm L2 476 được ghép nối với một hoặc nhiều mức bộ nhớ đệm khác và thực tế với bộ nhớ chính.

Bằng cách lấy ví dụ, kiến trúc lõi đối tên thanh ghi, phát hành/thực thi không theo thứ tự làm ví dụ có thể thực hiện đường truyền bằng ống 400 như sau: 1) nạp lệnh 438 thực hiện giai đoạn nạp 402 và giai đoạn mã hóa chiều dài 404; 2) bộ phận giải mã 440 thực hiện giai đoạn giải mã 406; 3) bộ phận đổi tên/cấp phát 452 thực hiện giai đoạn cấp phát 408 và giai đoạn đổi tên 410; 4) (các) bộ phận lập lịch 456 thực hiện giai đoạn lập lịch 412; 5) (các) bộ phận (các) tệp thanh ghi vật lý 458 và bộ phận nhớ 470 thực hiện giai đoạn đọc thanh ghi/đọc bộ nhớ 414; cụm thực thi 460 thực hiện giai đoạn thực thi 416; 6) bộ phận nhớ 470 và (các) bộ phận (các) tệp thanh ghi vật lý 458 thực hiện giai đoạn ghi lại/ghi bộ nhớ 418; 7) các bộ phận khác nhau có thể tham gia vào giai đoạn xử lý ngoại lệ 422; và 8) bộ phận 454 và (các) bộ phận (các) tệp thanh ghi vật lý 458 thực hiện giai đoạn cam kết 424.

Lõi 490 có thể hỗ trợ một hoặc nhiều tập hợp lệnh (ví dụ, tập hợp lệnh x86 (có một số mở rộng được bổ sung bằng các phiên bản mới); tập hợp lệnh MIPS của MIPS Technologies of Sunnyvale, CA; tập hợp lệnh ARM (có các mở rộng bổ sung tùy chọn chẳng hạn như NEON) của ARM Holdings of Sunnyvale, CA), bao gồm (các) lệnh được

mô tả trong bản mô tả này. Theo một phương án, lõi 490 bao gồm mạch logic để hỗ trợ mở rộng tập hợp lệnh dữ liệu được đóng gói (ví dụ, AVX1, AVX2), nhờ đó cho phép các hoạt động được sử dụng bởi nhiều ứng dụng đa phương tiện để được thực hiện bằng cách sử dụng dữ liệu được đóng gói.

Cần hiểu rằng lõi có thể hỗ trợ đa luồng (thực thi hai hoặc nhiều hơn hai tập hợp hoạt động hoặc luồng song song), và có thể thực hiện như vậy theo các cách khác nhau bao gồm đa luồng lát thời gian, đa luồng đồng thời (trong đó một lõi vật lý cung cấp lõi logic cho mỗi trong số các luồng mà lõi vật lý thực hiện đa luồng đồng thời), hoặc sự kết hợp của chúng (ví dụ, nạp và giải mã lát thời gian và sau đó thực hiện đa luồng đồng thời chẳng hạn như trong công nghệ siêu luồng Intel®).

Mặc dù quá trình đổi tên thanh ghi được mô tả trong ngữ cảnh thực thi không theo thứ tự, nhưng cần hiểu rằng quá trình đổi tên thanh ghi có thể được sử dụng trong kiến trúc theo thứ tự. Mặc dù phương án được minh họa của bộ xử lý cũng bao gồm lệnh riêng biệt và các bộ phận đệm dữ liệu 434/474 và bộ phận đệm L2 được chia sẻ 476, nhưng các phương án thay thế có thể có một bộ đệm trong cho cả các lệnh và dữ liệu, chẳng hạn như, ví dụ, bộ đệm trong mức 1 (L1), hoặc nhiều mức của bộ đệm trong. Theo một số phương án, hệ thống này có thể bao gồm sự kết hợp của bộ đệm trong và bộ đệm ngoài ở bên ngoài lõi và/hoặc bộ xử lý. Theo cách khác, tất cả trong số bộ nhớ đệm có thể ở bên ngoài lõi và/hoặc bộ xử lý.

Kiến trúc lõi theo thứ tự làm ví dụ cụ thể

FIG.5A-FIG.5B minh họa sơ đồ khối của kiến trúc lõi theo thứ tự làm ví dụ cụ thể hơn, lõi sẽ là một trong số các khối logic (bao gồm các lõi khác của cùng kiểu và/hoặc các kiểu khác nhau) trong chip. Các khối logic truyền thông qua mạng kết nối băng tần cao (ví dụ, mạng vòng) với một số mạch logic chức năng cố định, giao diện I/O bộ nhớ và mạch logic I/O cần thiết khác, phụ thuộc vào ứng dụng.

FIG.5A là sơ đồ khối của một lõi bộ xử lý, cùng với kết nối của nó với mạng kết nối trên khuôn 502 và có tập hợp con cục bộ của nó của bộ đệm mức 2 (L2) 504, theo các phương án của sáng chế. Theo một phương án, bộ giải mã lệnh 500 hỗ trợ tập hợp lệnh x86

có mở rộng tập hợp lệnh dữ liệu được đóng gói. Bộ đệm L1 506 cho phép truy nhập độ trễ thấp vào bộ nhớ đệm vào các bộ phận vô hướng và vector. Mặc dù theo một phương án (để đơn giản hóa việc thiết kế), bộ phận vô hướng 508 và bộ phận vector 510 sử dụng các tập hợp thanh ghi riêng biệt (một cách tương ứng, các thanh ghi vô hướng 512 và các thanh ghi vector 514) và dữ liệu được truyền giữa chúng được ghi vào bộ nhớ và sau đó được đọc lại từ bộ đệm mức 1 (L1) 506, các phương án thay thế của sáng chế có thể sử dụng cách tiếp cận khác nhau (ví dụ, sử dụng một tập hợp thanh ghi hoặc bao gồm đường truyền thông cho phép dữ liệu cần được truyền giữa hai tập thanh ghi mà không được ghi và đọc lại).

Tập hợp con cục bộ của bộ đệm L2 504 là một phần của bộ đệm L2 toàn cầu được chia thành các tập hợp con cục bộ riêng biệt, một trên mỗi lõi bộ xử lý. Mỗi lõi bộ xử lý có đường dẫn truy nhập trực tiếp đến tập hợp con cục bộ của bộ đệm L2 504. Dữ liệu được đọc bởi lõi bộ xử lý được lưu trữ trong tập hợp con bộ đệm L2 504 của nó và có thể được truy nhập nhanh chóng, song song với các lõi bộ xử lý khác truy nhập các tập hợp con bộ đệm L2 cục bộ của chúng. Dữ liệu được ghi bởi lõi bộ xử lý được lưu trữ trong tập hợp con bộ đệm L2 504 của chính nó và còn từ các tập hợp con khác, khi cần. Mạng vòng đảm bảo tính nhất quán cho dữ liệu được chia sẻ. Mạng vòng là hai chiều để cho phép các đại lý chẳng hạn như các lõi bộ xử lý, các bộ đệm L2 và các khối logic khác truyền thông với nhau trong chip. Mỗi đường dẫn dữ liệu vòng rộng 1024-bit trên mỗi hướng theo một số phương án.

FIG.5B là hình vẽ phóng to của một phần của lõi bộ xử lý trên FIG.5A theo các phương án của sáng chế. FIG.5B bao gồm bộ đệm dữ liệu L1 506A một phần của bộ nhớ đệm L1 504, cũng như thông tin chi tiết hơn về bộ phận vector 510 và các thanh ghi vector 514. Cụ thể là, bộ phận vector 510 là bộ phận xử lý vector (Vector Processing Unit, VPU) rộng 6 (xem ALU rộng 16 528), thực thi một hoặc nhiều trong số các lệnh số nguyên, dấu phẩy động độ chính xác đơn, và dấu phẩy động độ chính xác kép. VPU hỗ trợ xáo trộn các đầu vào thanh ghi bằng bộ phận xáo trộn 520, chuyển đổi số bằng các bộ phận chuyển đổi số 522A-B, và sao chép bằng bộ phận sao chép 524 đối với đầu vào bộ nhớ.

Bộ xử lý có bộ điều khiển bộ nhớ và đồ họa được tích hợp.

FIG.6 là sơ đồ khối của bộ xử lý 600 có thể có nhiều hơn một lõi, có thể có bộ điều khiển bộ nhớ được tích hợp, và có thể có đồ họa được tích hợp theo các phương án của sáng chế. Các hộp đường nét liền trên FIG.6 minh họa bộ xử lý 600 có một lõi 602A, đại lý hệ thống 610, tập hợp một hoặc nhiều bộ phận điều khiển bus 616, trong khi các hộp đường nét đứt là bổ sung tùy chọn minh họa bộ xử lý thay thế 600 có nhiều lõi 602A-N, tập hợp một hoặc nhiều bộ phận điều khiển bộ nhớ được tích hợp 614 trong bộ phận đại lý hệ thống 610, và mạch logic chuyên dụng 608.

Do đó, các hoạt động khác nhau của bộ xử lý 600 có thể bao gồm: 1) CPU có mạch logic chuyên dụng 608 được tích hợp mạch logic đồ họa và/hoặc khoa học (thông lượng) (có thể bao gồm một hoặc nhiều lõi), và các lõi 602A-N là một hoặc nhiều lõi đa năng (ví dụ, các lõi theo thứ tự đa năng, các lõi không theo thứ tự đa năng, sự kết hợp của hai lõi này); 2) bộ đồng xử lý có các lõi 602A-N là số lượng lớn lõi chuyên dụng chủ yếu dùng cho đồ họa và/hoặc khoa học (thông lượng); và 3) bộ đồng xử lý có các lõi 602A-N là số lượng lớn lõi theo thứ tự đa năng. Do đó, bộ xử lý 600 có thể là bộ xử lý đa năng, bộ đồng xử lý hoặc bộ xử lý chuyên dụng, chẳng hạn như, ví dụ, bộ xử lý mạng hoặc truyền thông, động cơ nén, bộ xử lý đồ họa, GPGPU (bộ phận xử lý đồ họa đa năng), bộ đồng xử lý lõi được tích hợp nhiều (Many Integrated Core, MIC) thông lượng cao (bao gồm 30 lõi hoặc nhiều hơn 30 lõi), bộ xử lý nhúng, hoặc tương tự. Bộ xử lý có thể được thực hiện trên một hoặc nhiều chip. Bộ xử lý 600 có thể là một phần của và/hoặc có thể được thực hiện trên một một hoặc nhiều lớp nền sử dụng một bất kỳ trong số các công nghệ xử lý, chẳng hạn như, ví dụ, BiCMOS, CMOS, hoặc NMOS.

Hệ thống phân cấp bộ nhớ bao gồm một hoặc nhiều mức bộ đệm trong các lõi 604A-N, tập hợp hoặc một hoặc nhiều bộ phận đệm chia sẻ 606, và bộ nhớ ngoài (không được thể hiện trên hình vẽ) được ghép nối với tập hợp bộ phận điều khiển bộ nhớ được tích hợp 614. Tập hợp bộ phận đệm chia sẻ 606 có thể bao gồm một hoặc nhiều bộ nhớ đệm mức giữa, chẳng hạn như mức 2 (L2), mức 3 (L3), mức 4 (L4), hoặc các mức bộ đệm khác, bộ đệm mức cuối (Last Level Cache, LLC), và/hoặc các sự kết hợp của chúng. Mặc dù theo một phương án bộ phận kết nối dựa vào vòng 612 kết nối mạch logic đồ họa tích hợp 608, tập hợp bộ phận đệm chia sẻ 606, và bộ phận đại lý hệ thống 610/(các) bộ phận điều khiển

bộ nhớ được tích hợp 614, nhưng các phương án thay thế có thể sử dụng số lượng kỹ thuật đã biết bất kỳ để kết nối các bộ phận này. Theo một phương án, tính nhất quán được duy trì giữa một hoặc nhiều bộ phận đệm 606 và các lõi 602-A-N.

Theo một số phương án, một hoặc nhiều trong số các lõi 602A-N có khả năng đa luồng. Đại lý hệ thống 610 bao gồm các thành phần này điều phối và vận hành các lõi 602A-N. Bộ phận đại lý hệ thống 610 có thể bao gồm ví dụ bộ điều khiển công suất (Power Control Unit, PCU) và bộ phận hiển thị. PCU có thể là hoặc bao gồm mạch logic và các thành phần cần để điều chỉnh trạng thái nguồn của các lõi 602A-N và mạch logic đồ họa tích hợp 608. Bộ phận hiển thị để điều khiển một hoặc nhiều màn hiển thị được nối bên ngoài.

Các lõi 602A-N có thể đồng nhất hoặc không đồng nhất về mặt tập hợp lệnh kiến trúc; tức là, hai hoặc nhiều hơn hai lõi 602A-N có thể có khả năng thực thi cùng tập hợp lệnh, trong khi các lõi khác có thể chỉ có khả năng thực thi tập hợp con của tập hợp lệnh này hoặc tập hợp lệnh khác nhau.

Các kiến trúc máy tính làm ví dụ

FIG.7-FIG.10 là các sơ đồ khối của các kiến trúc máy tính làm ví dụ. Các thiết kế hệ thống và cấu hình khác đã biết trong tình trạng kỹ thuật cho máy tính xách tay, máy tính để bàn, PC cầm tay, thiết bị hỗ trợ cá nhân kỹ thuật số, trạm làm việc kỹ thuật, máy chủ, thiết bị mạng, trung tâm mạng, mạch chuyển đổi, bộ xử lý nhúng, bộ xử lý tín hiệu số (Digital Signal Processor, DSP), thiết bị đồ họa, thiết bị trò chơi video, hộp giải mã tín hiệu, bộ điều khiển micro, điện thoại di động, máy phát đa phương tiện cầm tay, thiết bị cầm tay, và các thiết bị điện tử khác nhau, cũng thích hợp. Nói chung, rất nhiều hệ thống hoặc thiết bị điện tử có khả năng kết hợp bộ xử lý và/hoặc mạch logic thực thi khác như được bộc lộ trong bản mô tả này nói chung là thích hợp.

Bây giờ tham chiếu đến FIG.7, được thể hiện là sơ đồ khối của hệ thống 700 theo một phương án của sáng chế. Hệ thống 700 có thể bao gồm một hoặc nhiều bộ xử lý 710, 715, được ghép nối với trung tâm bộ điều khiển 720. Theo một phương án, trung tâm bộ điều khiển 720 bao gồm trung tâm bộ điều khiển bộ nhớ đồ họa (Graphics Memory

Controller Hub, GMCH) 790 và trung tâm đầu vào/đầu ra (Input/Output Hub, IOH) 750 (có thể là trên các chip riêng biệt); GMCH 790 bao gồm các bộ điều khiển bộ nhớ và đồ họa được ghép nối với bộ nhớ 740 và bộ đồng xử lý 745; IOH 750 ghép nối các thiết bị đầu vào/đầu ra (Input/Output, I/O) 760 với GMCH 790. Theo cách khác, một hoặc cả hai trong số các bộ điều khiển bộ nhớ và đồ họa được tích hợp trong bộ xử lý (như được mô tả trong bản mô tả này), bộ nhớ 740 và bộ đồng xử lý 745 được ghép nối trực tiếp với bộ xử lý 710, và trung tâm bộ điều khiển 720 trong một chip với IOH 750.

Bản chất tùy chọn của các bộ xử lý bổ sung 715 được ký hiệu trên FIG.7 bằng các đường đứt nét. Mỗi bộ xử lý 710, 715 có thể bao gồm một hoặc nhiều lõi xử lý được mô tả trong bản mô tả này và có thể là một số phiên bản của bộ xử lý 600.

Bộ nhớ 740 có thể là, ví dụ, bộ nhớ truy nhập ngẫu nhiên động (Dynamic Random Access Memory, DRAM), bộ nhớ thay đổi pha (Phase Change Memory, PCM), hoặc sự kết hợp của cả hai bộ nhớ này. Đối với ít nhất một phương án, trung tâm bộ điều khiển 720 truyền thông với (các) bộ xử lý 710, 715 qua bus đa nhánh, chẳng hạn như bus phía trước (Frontside Bus, FSB), giao diện điểm đến điểm, hoặc kết nối tương tự 795.

Theo một phương án, bộ đồng xử lý 745 là bộ xử lý chuyên dụng, chẳng hạn như, ví dụ, bộ xử lý MIC thông lượng cao, bộ xử lý mạng hoặc truyền thông, động cơ nén, bộ xử lý đồ họa, GPGPU, bộ xử lý nhúng, hoặc tương tự. Theo một phương án, trung tâm bộ điều khiển 720 có thể bao gồm bộ gia tốc đồ họa tích hợp.

Có thể có nhiều sự khác biệt giữa các tài nguyên vật lý 710, 715 về mặt phổ đo lường giá trị bao gồm các đặc điểm kiến trúc, vi kiến trúc, nhiệt, tiêu thụ điện năng, và tương tự.

Theo một phương án, bộ xử lý 710 thực thi các lệnh để điều khiển các hoạt động xử lý dữ liệu của kiểu thông thường. Được nhúng trong các lệnh có thể là các lệnh của bộ đồng xử lý. Bộ xử lý 710 nhận diện các lệnh của bộ đồng xử lý này là kiểu cần được thực thi bởi bộ đồng xử lý đính kèm 745. Theo đó, bộ xử lý 710 phát hành các lệnh của bộ đồng xử lý này (hoặc các tín hiệu điều khiển biểu diễn các lệnh của bộ đồng xử lý) trên bus hoặc kết nối khác của bộ đồng xử lý, cho bộ đồng xử lý 745. (Các) bộ đồng xử lý 745 chấp nhận và

thực thi các lệnh được nhận của bộ đồng xử lý.

Bây giờ tham chiếu đến FIG.8, được thể hiện là sơ đồ khối của hệ thống làm ví dụ cụ thể thứ nhất 800 theo một phương án của sáng chế. Như được thể hiện trên FIG.8, hệ thống nhiều bộ xử lý 800 là hệ thống kết nối điểm đến điểm, và bao gồm bộ xử lý thứ nhất 870 và bộ xử lý thứ hai 880 được ghép nối qua kết nối điểm đến điểm 850. Mỗi trong số các bộ xử lý 870 và 880 có thể là một số phiên bản của bộ xử lý 600. Theo một phương án của sáng chế, các bộ xử lý 870 và 880 lần lượt là các bộ xử lý 710 và 715, trong khi bộ đồng xử lý 838 là bộ đồng xử lý 745. Theo một phương án khác, các bộ xử lý 870 và 880 lần lượt là bộ xử lý 710, bộ đồng xử lý 745.

Các bộ xử lý 870 và 880 được thể hiện lần lượt bao gồm bộ phận điều khiển bộ nhớ được tích hợp (Integrated Memory Controller, IMC) 872 và 882. Bộ xử lý 870 cũng bao gồm như một phần của các giao diện điểm đến điểm (Point-to-Point, P-P) bộ phận điều khiển bus 876 và 878; tương tự, bộ xử lý thứ hai 880 bao gồm các giao diện P-P 886 và 888. Các bộ xử lý 870, 880 có thể trao đổi thông tin qua giao diện điểm đến điểm (Point-to-Point, P-P) 850 sử dụng các mạch giao diện P-P 878, 888. Như được thể hiện trên FIG.8, các IMC 872 và 882 ghép nối các bộ xử lý với các bộ nhớ tương ứng, cụ thể là bộ nhớ 832 và bộ nhớ 834, có thể là các phần của bộ nhớ chính được gắn cục bộ với các bộ xử lý tương ứng.

Các bộ xử lý 870, 880 có thể trao đổi thông tin với bộ chip 890 qua các giao diện P-P riêng lẻ 852, 854 sử dụng các mạch giao diện điểm đến điểm 876, 894, 886, 898. Bộ chip 890 có thể tùy chọn trao đổi thông tin với bộ đồng xử lý 838 qua giao diện hiệu suất cao 892. Theo một phương án, bộ đồng xử lý 838 là bộ xử lý chuyên dụng, chẳng hạn như, ví dụ, bộ xử lý MIC thông lượng cao, bộ xử lý mạng hoặc truyền thông, động cơ nén, bộ xử lý đồ họa, GPGPU, bộ xử lý nhúng, hoặc tương tự.

Bộ đệm chia sẻ (không được thể hiện trên hình vẽ) có thể nằm trong bộ xử lý hoặc nằm ngoài cả hai bộ xử lý, được nối với các bộ xử lý qua kết nối P-P, sao cho thông tin bộ đệm cục bộ của một hoặc cả hai bộ xử lý có thể được lưu trữ trong bộ đệm chia sẻ nếu bộ xử lý được đặt ở chế độ công suất thấp.

Bộ chip 890 có thể được ghép nối với bus thứ nhất 816 qua giao diện 896. Theo một phương án, bus thứ nhất 816 có thể là bus kết nối thành phần ngoại vi (Peripheral Component Interconnect, PCI), hoặc bus chẳng hạn như bus biểu thị PCI hoặc bus kết nối I/O khác, mặc dù phạm vi của sáng chế không bị giới hạn.

Như được thể hiện trên FIG.8, các thiết bị I/O 814 khác nhau có thể được ghép nối với bus thứ nhất 816, cùng với cầu bus 818 ghép nối bus thứ nhất 816 với bus thứ hai 820. Theo một phương án, một hoặc nhiều bộ xử lý bổ sung 815, chẳng hạn như các bộ đồng xử lý, các bộ xử lý MIC thông lượng cao, các bộ gia tốc của GPGPU (chẳng hạn như, ví dụ, các bộ gia tốc đồ họa hoặc các bộ phận xử lý tín hiệu số (Digital Signal Processing, DSP)), các mảng cổng lập trình được dạng trường, hoặc bộ xử lý khác bất kỳ, được ghép nối với bus thứ nhất 816. Theo một phương án, bus thứ hai 820 có thể là bus đếm chân thấp (Low Pin Count, LPC). Các thiết bị khác nhau có thể được ghép nối với bus thứ hai 820 bao gồm, ví dụ, bàn phím và/hoặc chuột 822, các thiết bị truyền thông 827 và bộ phận lưu trữ 828 chẳng hạn như ổ đĩa hoặc thiết bị lưu trữ lớn khác có thể bao gồm các lệnh/mã và dữ liệu 830, theo một phương án. Ngoài ra, I/O âm thanh 824 có thể được ghép nối với bus thứ hai 816. Lưu ý rằng các kiến trúc khác là khả dụng. Ví dụ, thay vì kiến trúc điểm đến điểm trên FIG.8, hệ thống có thể khai triển kiến trúc bus đa nhánh hoặc kiến trúc khác.

Bây giờ tham chiếu đến FIG.9, được thể hiện là sơ đồ khối của hệ thống làm ví dụ cụ thể thứ hai 900 theo một phương án của sáng chế. Các phần tử tương tự trên FIG.8 và FIG.9 mang các số chỉ dẫn tương tự, và các khía cạnh nhất định của FIG.8 được lược bỏ khỏi FIG.9 để tránh làm tối nghĩa các khía cạnh khác của FIG.9.

FIG.9 minh họa rằng các bộ xử lý 870, 880 có thể lần lượt bao gồm bộ nhớ tích hợp và mạch logic điều khiển (Control Logic, “CL”) I/O 972 và 982. Do đó, CL 972, 982 bao gồm các bộ phận điều khiển bộ nhớ được tích hợp và bao gồm mạch logic điều khiển I/O. FIG.9 minh họa rằng không chỉ các bộ nhớ 832, 834 được ghép nối với CL 872, 882, mà các thiết bị I/O 914 cũng được ghép nối với mạch logic điều khiển 872, 882. Các thiết bị I/O kế thừa 915 được ghép nối với bộ chip 890.

Bây giờ tham chiếu đến FIG.10, được thể hiện là sơ đồ khối của SoC 1000 theo một

phương án của sáng chế. Các phần tử tương tự trên FIG.6 mang các số chỉ dẫn tương tự. Ngoài ra, các hộp đường nét đứt là các bộ phận tùy chọn trên các SoC nâng cao hơn. Trên FIG.10, (các) bộ phận kết nối 1002 được ghép nối với: bộ xử lý ứng dụng 1010 bao gồm tập hợp một hoặc nhiều lõi 102A-N, các bộ phận đệm 604A-N, và (các) bộ phận đệm chia sẻ 606; bộ phận đại lý hệ thống 610; (các) bộ phận điều khiển bus 616; (các) bộ phận điều khiển bộ nhớ được tích hợp 614; tập hợp hoặc một hoặc nhiều bộ đồng xử lý 1020 có thể bao gồm mạch logic đồ họa tích hợp, bộ xử lý hình ảnh, bộ xử lý âm thanh, và bộ xử lý video; bộ phận nhớ truy nhập ngẫu nhiên tĩnh (Static Random Access Memory, SRAM) 1030; bộ phận truy nhập bộ nhớ trực tiếp (Direct Memory Access, DMA) 1032; và bộ phận hiển thị 1040 để ghép nối với một hoặc nhiều màn hiển thị bên ngoài. Theo một phương án, (các) bộ đồng xử lý 1020 bao gồm bộ xử lý chuyên dụng, chẳng hạn như, ví dụ, bộ xử lý mạng hoặc truyền thông, động cơ nén, GPGPU, bộ xử lý MIC thông lượng cao, bộ xử lý nhúng, hoặc tương tự.

Các phương án về các cơ chế được bộc lộ trong bản mô tả này có thể được khai triển trong phần cứng, phần mềm, phần sụn, hoặc sự kết hợp các cách tiếp cận khai triển này. Các phương án theo sáng chế có thể được khai triển ở dạng các chương trình máy tính hoặc mã chương trình thực thi trên các hệ thống lập trình được bao gồm ít nhất một bộ xử lý, hệ thống lưu trữ (bao gồm bộ nhớ khả biến và bất khả biến và/hoặc các phần tử lưu trữ), ít nhất một thiết bị đầu vào, và ít nhất một thiết bị đầu ra.

Mã chương trình, chẳng hạn như mã 830 được minh họa trên FIG.8, có thể được áp dụng để nhập vào các lệnh để thực hiện các chức năng được mô tả trong bản mô tả này và tạo ra thông tin đầu ra. Thông tin đầu ra có thể được áp dụng cho một hoặc nhiều thiết bị đầu ra, theo cách đã biết. Đối với các mục đích của sáng chế, hệ thống xử lý bao gồm hệ thống bất kỳ có bộ xử lý, chẳng hạn như, ví dụ; bộ xử lý tín hiệu số (Digital Signal Processor, DSP), bộ vi điều khiển, mạch tích hợp chuyên dụng (Application Specific Integrated Circuit, ASIC), hoặc bộ vi xử lý.

Mã chương trình có thể được triển khai bằng ngôn ngữ lập trình hướng đối tượng hoặc thủ tục mức cao để truyền thông với hệ thống xử lý. Mã chương trình cũng có thể được triển khai bằng hợp ngữ hoặc ngôn ngữ máy, nếu cần. Thực tế, các cơ chế được mô

tả trong bản mô tả này không bị giới hạn ở phạm vi ngôn ngữ lập trình cụ thể bất kỳ. Trong trường hợp bất kỳ, ngôn ngữ này có thể là ngôn ngữ biên dịch hoặc ngôn ngữ diễn dịch.

Một hoặc nhiều khía cạnh theo ít nhất một phương án có thể được thực hiện bởi các lệnh biểu diễn được lưu trữ trên vật ghi đọc được bằng máy biểu diễn mạch logic khác nhau trong bộ xử lý, khi được đọc bởi máy làm cho máy tạo ra mạch logic để thực hiện các kỹ thuật được mô tả trong bản mô tả này. Các biểu diễn này, được biết đến là “các lõi IP” có thể được lưu trữ trên vật ghi hữu hình, đọc được bằng máy và được cấp cho các khách hàng khác nhau hoặc các nhà máy chế tạo để tải vào các máy chế tạo thực sự tạo ra mạch logic hoặc bộ xử lý.

Vật ghi lưu trữ đọc được bằng máy như vậy có thể bao gồm, không giới hạn, các cách bố trí lâu dài, hữu hình của các sản phẩm được chế tạo hoặc được tạo thành bởi máy hoặc thiết bị, bao gồm vật ghi lưu trữ chẳng hạn như các đĩa cứng, kiểu đĩa khác bất kỳ bao gồm các đĩa mềm, đĩa quang, bộ nhớ chỉ đọc đĩa nén (Compact Disk Read-Only Memory, CD-ROM), đĩa nén ghi lại được (Compact Disk Rewritable, CD-RW), và các đĩa từ quang, các thiết bị bán dẫn chẳng hạn như bộ nhớ chỉ đọc (Read-Only Memory, ROM), các bộ nhớ truy nhập ngẫu nhiên (Random Access Memory, RAM) chẳng hạn như các bộ nhớ truy nhập ngẫu nhiên động (Dynamic Random Access Memory, DRAM), bộ nhớ truy nhập ngẫu nhiên tĩnh (Static Random Access Memory, SRAM), bộ nhớ chỉ đọc lập trình được xóa được (Erasable Programmable Read-Only Memory, EPROM), bộ nhớ nhanh, bộ nhớ chỉ đọc lập trình được xóa được bằng tín hiệu điện (Electrically Erasable Programmable Read-Only Memory, EEPROM), bộ nhớ thay đổi pha (Phase Change Memory, PCM), các thẻ từ hoặc quang, hoặc kiểu vật ghi khác bất kỳ thích hợp để lưu trữ các lệnh điện tử.

Theo đó, các phương án của sáng chế cũng bao gồm vật ghi lâu dài, hữu hình đọc được bằng máy chứa các lệnh hoặc chứa dữ liệu thiết kế, chẳng hạn như ngôn ngữ mô tả phần cứng (Hardware Description Language, HDL), xác định các dấu hiệu của các cấu trúc, mạch, thiết bị, các bộ xử lý và/hoặc hệ thống được mô tả trong bản mô tả này. Các phương án này cũng có thể được gọi là các sản phẩm chương trình.

Mô phỏng (bao gồm dịch nhị phân, biên hình mã, v.v.)

Trong một số trường hợp, bộ chuyển đổi lệnh có thể được sử dụng để biến đổi lệnh từ tập hợp lệnh nguồn thành tập hợp lệnh đích. Ví dụ, bộ chuyển đổi lệnh có thể dịch (ví dụ, sử dụng dịch nhị phân tĩnh, dịch nhị phân động bao gồm biên dịch động), biến hình, mô phỏng, hoặc chuyển đổi lệnh thành một hoặc nhiều lệnh khác sẽ được xử lý bởi lõi. Bộ chuyển đổi lệnh có thể được thực hiện trong phần mềm, phần cứng, phần sụn, hoặc sự kết hợp của chúng. Bộ chuyển đổi lệnh có thể ở trên bộ xử lý, tắt bộ xử lý, hoặc bật một phần và tắt một phần bộ xử lý.

FIG.11 là sơ đồ khối tương phản việc sử dụng bộ biến đổi lệnh phần mềm để biến đổi các lệnh nhị phân trong tập hợp lệnh nguồn thành các lệnh nhị phân trong tập hợp lệnh đích theo các phương án của sáng chế. Theo phương án được minh họa, bộ chuyển đổi lệnh là bộ biến đổi lệnh phần mềm, mặc dù theo cách khác bộ chuyển đổi lệnh có thể được thực hiện trong phần mềm, phần sụn, phần cứng, hoặc các sự kết hợp khác nhau của chúng. FIG.11 thể hiện chương trình trong ngôn ngữ mức cao 1102 có thể được biên dịch sử dụng trình biên dịch thứ nhất 1104 để tạo ra mã nhị phân thứ nhất (ví dụ, x86) 1106 có thể được thực thi nguyên bản bởi bộ xử lý bằng ít nhất một lõi thiết lập lệnh thứ nhất 1116. Theo một số phương án, bộ xử lý có ít nhất một lõi thiết lập lệnh thứ nhất 1116 biểu diễn bộ xử lý bất kỳ gần như có thể thực hiện cùng các chức năng với bộ xử lý Intel có ít nhất một lõi thiết lập lệnh x86 bằng cách thực thi tương thích hoặc xử lý (1) một phần quan trọng của tập hợp lệnh của lõi thiết lập lệnh x86 Intel hoặc (2) các phiên bản mã đối tượng của các ứng dụng hoặc phần mềm khác nhằm chạy trên các bộ xử lý Intel có ít nhất một lõi thiết lập lệnh x86, để gần như đạt được cùng kết quả với bộ xử lý Intel bằng ít nhất một lõi thiết lập lệnh x86. Trình biên dịch thứ nhất 1104 biểu diễn trình biên dịch hoạt động được để tạo ra mã nhị phân của tập hợp lệnh thứ nhất 1106 (ví dụ, mã đối tượng) có thể, có hoặc không có quá trình xử lý liên kết bổ sung, được thực thi trên bộ xử lý có ít nhất một lõi thiết lập lệnh thứ nhất 1116. Tương tự, FIG.11 thể hiện chương trình trong ngôn ngữ mức cao 1102 có thể được biên dịch sử dụng trình biên dịch tập hợp lệnh thay thế 1108 để tạo ra mã nhị phân tập hợp lệnh thay thế 1110 có thể được thực thi nguyên bản bởi bộ xử lý mà không có ít nhất một lõi thiết lập lệnh thứ nhất 1114 (ví dụ, bộ xử lý có các lõi thực thi tập hợp lệnh MIPS của MIPS Technologies of Sunnyvale, CA và/hoặc thực thi tập hợp lệnh ARM của ARM Holdings of Sunnyvale, CA). Bộ chuyển đổi lệnh 1112 được sử dụng để biến đổi mã

nhị phân thứ nhất 1106 thành mã có thể được thực thi nguyên bản bởi bộ xử lý mà không có lỗi thiết lập lệnh thứ nhất 1114. Mã được chuyển đổi này có thể không giống với mã nhị phân tập hợp lệnh thay thế 1110 vì khó tạo ra một bộ chuyển đổi lệnh có khả năng thực hiện điều này; tuy nhiên, mã được chuyển đổi sẽ thực hiện hoạt động chung và tạo thành các lệnh từ tập hợp lệnh thay thế. Do đó, bộ chuyển đổi lệnh 1112 biểu diễn phần mềm, phần sụn, phần cứng, hoặc sự kết hợp của chúng, qua mô phỏng, mô phỏng hoặc quy trình khác bất kỳ, cho phép bộ xử lý hoặc thiết bị điện tử khác không có bộ xử lý hoặc lỗi thiết lập lệnh thứ nhất để thực thi mã nhị phân thứ nhất 1106.

THIẾT BỊ VÀ PHƯƠNG PHÁP QUẢN LÝ VÀ XỬ LÝ HIỆU QUẢ NGĂN XẾP BÓNG

Một phương án theo sáng chế thực hiện các kỹ thuật để tránh các hoạt động nguyên tử để xác minh mã thông báo được sử dụng để xác nhận con trỏ ngăn xếp bóng (Shadow Stack Pointer, SSP) trong khi vẫn giữ lại các thuộc tính bảo mật cơ bản của cơ chế mã thông báo. Ngoài ra, các phương án nhất định bao gồm các phần mở rộng ảo hóa để hỗ trợ kiến trúc xác minh mã thông báo hiệu quả.

Trong nhiều trường hợp, kiến trúc tập hợp lệnh (Instruction Set Architecture, ISA), chẳng hạn như kiến trúc x86, cung cấp các mức quyền ưu tiên mà các lệnh có thể được thực thi. Các mức quyền ưu tiên được sử dụng để cải thiện độ tin cậy của hệ điều hành (ví dụ, nhân) được liên kết với nền tảng phần cứng bằng cách bảo vệ các tài nguyên khỏi sự truy cập trực tiếp của các thực thể ít tin cậy hơn. Theo một số phương án, các mức quyền ưu tiên này trong khoảng từ vòng 0, mức quyền ưu tiên được tin tưởng nhất (ví dụ, chế độ giám sát), đến vòng 3, hoặc mức không ưu tiên (ví dụ, chế độ ứng dụng hoặc người dùng). Mỗi mức quyền ưu tiên yêu cầu một số tài nguyên cần được cấp phát trước và được lưu trữ để sử dụng sau khi chuyển vòng sang mức quyền ưu tiên mới.

Mức quyền ưu tiên hiện tại của bộ xử lý điều khiển việc thực thi lệnh. Bộ xử lý có thể thay đổi mức quyền ưu tiên hiện tại của nó để điều khiển khả năng truy cập bởi ứng dụng vào các tài nguyên hệ thống chẳng hạn như các thanh ghi hệ thống, các lệnh hệ thống, và các vùng bộ nhớ hệ thống được chỉ định. Khi thực thi ứng dụng, bộ xử lý sử dụng ngăn

xếp dữ liệu lưu trữ cả dữ liệu (ví dụ, các tham số và dữ liệu khác được chuyển đến các thường trình con, v.v.) và thông tin điều khiển để hướng dòng điều khiển của các lệnh được thực thi. Thông thường, thông tin điều khiển bao gồm một hoặc nhiều địa chỉ trả về mà dòng điều khiển sẽ trả về khi thủ tục được gọi hoàn thành.

Trong cuộc tấn công lập trình hướng trở lại (Return-Oriented Programming, ROP), những kẻ tấn công tìm cách giành quyền điều khiển ngăn xếp dữ liệu để chiếm đoạt dòng điều khiển chương trình. Ví dụ, kẻ tấn công có thể thực hiện các sửa đổi ngăn xếp không được ủy quyền của địa chỉ trả về của thủ tục được lưu trữ trong ngăn xếp dữ liệu. Các sửa đổi ngăn xếp không được ủy quyền được sử dụng để chuyển hướng dòng điều khiển của quy trình hiện tại đến một vị trí bộ nhớ tùy ý nhằm nâng cao mục tiêu của cuộc xâm nhập. Một yếu tố làm cho ngăn xếp dữ liệu dễ bị tổn thương hơn trước ROP và các kiểu tấn công lật đổ dòng điều khiển khác là ngăn xếp thường lưu trữ cả dữ liệu và thông tin điều khiển và do đó ngăn xếp dữ liệu cần viết được bởi phần mềm. Các yếu tố khác làm cho ngăn xếp dữ liệu dễ bị tổn thương trước các tấn công như vậy là việc chuyển đổi ngăn xếp dữ liệu thường được thực hiện như một hoạt động không có mức quyền ưu tiên.

Để đáp lại một số sự kiện (ví dụ, ngắt/ngoại lệ), bộ xử lý có thể chuyển quyền điều khiển chuỗi lệnh được thực thi ở mức người dùng không ưu tiên sang mức quyền ưu tiên giám sát để truy nhập các tài nguyên cần để xử lý sự kiện đó. Các yếu tố này được thảo luận ở trên có thể có xu hướng làm tăng khả năng hệ thống gặp phải các cuộc tấn công lật đổ dòng điều khiển do các lỗi bị khai thác cho phép sửa đổi con trỏ ngăn xếp và/hoặc thông tin dòng điều khiển (ví dụ, các địa chỉ trả về) (ví dụ, để trỏ đến bộ nhớ được điều khiển độc hại/kẻ tấn công).

Các phương án của sáng chế bao gồm các kỹ thuật để bảo vệ các ngăn xếp khỏi các cuộc tấn công lật đổ dòng điều khiển khi chuyển giao quyền ưu tiên. Theo một phương án, số lượng ngăn xếp “bóng” được tạo ra để bảo vệ thông tin điều khiển trên ngăn xếp dữ liệu khỏi các cố gắng giả mạo. Khi được sử dụng trong bản mô tả này, ngăn xếp bóng là ngăn xếp thứ hai tách khỏi ngăn xếp dữ liệu và được sử dụng để lưu trữ và bảo vệ thông tin từ ngăn xếp dữ liệu được sử dụng trong các hoạt động chuyển điều khiển. Ví dụ, ngăn xếp bóng có thể lưu trữ thông tin điều khiển nhưng không phải dữ liệu (ví dụ, không phải các

tham số và dữ liệu khác của kiểu được lưu trữ trên ngăn xếp mà các chương trình ứng dụng mức người dùng ghi và sửa đổi).

Theo một số phương án, ngăn xếp bóng chỉ ghi được bởi các lệnh nhất định, chẳng hạn như “GOI” (là ví dụ về các cơ chế gọi thủ tục sử dụng ngăn xếp dữ liệu) và “GOI XA” (ví dụ về lệnh nhánh thực hiện cả các lệnh phân nhánh micro và phân nhánh macro) thực hiện các hoạt động chuyển điều khiển và luồng (ví dụ, phân phối ngắt/ngoại lệ). Trong các trường hợp này, ngăn xếp bóng có thể không ghi được bởi các cửa hàng được khởi tạo phần mềm khác qua các lệnh khác, chẳng hạn như “MOV” truyền các byte trong một hoạt động và “XSAVE” được sử dụng để lưu trữ trạng thái ngữ cảnh vào bộ nhớ và phục hồi trạng thái ngữ cảnh từ bộ nhớ đến các thanh ghi của bộ xử lý. Việc bảo vệ ghi này đạt được bằng cách đánh dấu bộ nhớ được sử dụng cho các ngăn xếp bóng trong các bảng trang là bộ nhớ ngăn xếp bóng sao cho các cửa hàng được khởi tạo bởi phần mềm sử dụng các lệnh như “MOV”, “XSAVE”, v.v. không hoàn thành và lỗi. Tương tự, các lệnh chẳng hạn như “GOI”, khi chúng tạo ra các lệnh lưu trữ dành cho bộ nhớ ngăn xếp bóng, sử dụng các hoạt động lưu trữ cụ thể sao cho chúng chỉ hoàn thành nếu bộ nhớ mà việc lưu trữ được thực hiện được đánh dấu trong bảng trang là bóng ngăn xếp bộ nhớ. Nếu không, lỗi xảy ra.

Theo một số phương án, các ngăn xếp bóng có thể tùy chọn được tạo cấu hình để hoạt động riêng biệt ở chế độ mức người dùng không có quyền ưu tiên (ví dụ, mức quyền ưu tiên vòng 3) hoặc ở chế độ mức quyền ưu tiên được ưu tiên hoặc giám sát (mức quyền ưu tiên vòng 0, vòng 1, hoặc vòng 2). Để hỗ trợ chuyển đổi các ngăn xếp bóng trên các chuyển mức quyền ưu tiên từ mức quyền ưu tiên nguồn sang mức quyền ưu tiên mới, một phương án thực hiện bao gồm các thanh ghi được liên kết với mỗi mức quyền ưu tiên.

Theo một số phương án, phần mềm hệ thống (ví dụ nhân hệ thống hoặc hệ điều hành) tải con trỏ ngăn xếp bóng (Shadow Stack Pointer, SSP) vào một hoặc nhiều thanh ghi ở mỗi mức quyền ưu tiên. Bộ xử lý sau đó truy nhập SSP qua thanh ghi được liên kết với mức quyền ưu tiên mới để nhận dạng ngăn xếp bóng cho mức quyền ưu tiên này. Khi hoạt động, khi bộ xử lý chọn SSP cho mức quyền ưu tiên mới, thì bộ xử lý xác nhận/xác minh SSP để sử dụng bởi bộ xử lý ở mức mới. Nếu việc xác nhận thành công, thì bộ xử lý thiết lập ngăn xếp bóng ở mức quyền ưu tiên mới là bận và đẩy thông tin địa chỉ trả về cho

mức quyền ưu tiên nguồn vào ngăn xếp bóng. Sau đó, các phương án của sáng chế đảm bảo rằng chuyển tiếp bất kỳ của bộ xử lý trở lại mức quyền ưu tiên nguồn được xác minh trước khi ngăn xếp bóng được đánh dấu là không bận. Việc xác minh này bao gồm đảm bảo rằng địa chỉ trả về của mức quyền ưu tiên nguồn được đẩy vào các ngăn xếp bóng trùng khớp địa chỉ hiện tại được liên kết với ngăn xếp dữ liệu. Để thực hiện việc xác minh này, bộ xử lý sử dụng các hoạt động tải cụ thể (ví dụ, “shadowStackPop” hoặc tải bằng các ngữ nghĩa ngăn xếp bóng) sao cho nếu hoạt động tải không vào bộ nhớ được đánh dấu là bộ nhớ ngăn xếp bóng trong các bảng trang sau đó việc tải này không hoàn thành (ví dụ, tạo ra điều kiện lỗi). Do đó cố gắng bất kỳ để trở con trở ngăn xếp bóng đến bộ nhớ không phải ngăn xếp bóng được phát hiện và được ngăn chặn.

Theo các phương án khác, các kỹ thuật được mô tả trong bản mô tả này có thể được sử dụng, ví dụ, để bảo vệ chuyển đổi ngăn xếp bóng là một phần của việc phân phối ngắt/ngoại lệ hoặc một phần của chuyển đổi tác vụ từ tác vụ hiện tại ở mức quyền ưu tiên nguồn sang tác vụ mới ở mức quyền ưu tiên mới, để thực hiện các lệnh gọi hệ thống nhanh (ví dụ, chuyển quyền ưu tiên chế độ người dùng sang chế độ điều khiển hoàn toàn của ủy quyền phần cứng) bằng các ngăn xếp bóng, để bảo vệ các ngăn xếp bóng khỏi các cuộc tấn công xuyên lòng, chẳng hạn như tấn công từ quy trình quyền ưu tiên nhỏ đến quy trình quyền ưu tiên lớn, cũng như để bảo vệ các ngăn xếp bóng trong các chuyển vòng từ các kiểu sửa đổi ngăn xếp không được ủy quyền được sử dụng bởi những kẻ tấn công.

FIG.12 minh họa sơ đồ khối của thiết bị xử lý để hỗ trợ các mở rộng bộ xử lý để bảo vệ các ngăn xếp trong quá trình chuyển vòng theo một phương án. Thiết bị xử lý có thể thường được gọi là “bộ xử lý” hoặc “CPU”. “Bộ xử lý” hoặc “CPU” trong bản mô tả này sẽ đề cập đến thiết bị có khả năng thực thi các lệnh mã hóa các phép toán số, logic, hoặc I/O. Trong một ví dụ minh họa, bộ xử lý có thể bao gồm bộ phận logic số học (Arithmetic Logic Unit, ALU), bộ phận điều khiển, và các thanh ghi. Theo một khía cạnh khác, bộ xử lý có thể bao gồm một hoặc nhiều lõi xử lý, và do đó có thể là bộ xử lý đơn lõi thường có khả năng xử lý một đường truyền bằng ống lệnh, hoặc bộ xử lý đa lõi có thể xử lý đồng thời nhiều đường truyền bằng ống lệnh. Theo một khía cạnh khác, bộ xử lý có thể được thực hiện ở dạng một mạch tích hợp, hai hoặc nhiều hơn hai mạch tích hợp, hoặc có thể là

thành phần của môđun nhiều chip (ví dụ, trong đó các khuôn bộ vi xử lý riêng lẻ nằm trong một gói mạch tích hợp và do đó chia sẻ một ổ cắm).

Như được thể hiện trên FIG.12, thiết bị xử lý có thể bao gồm các thành phần khác nhau. Theo một phương án, thiết bị xử lý có thể bao gồm một hoặc nhiều lõi bộ xử lý 1210 và bộ phận điều khiển bộ nhớ 1220, trong số các thành phần khác, được ghép nối với nhau như được thể hiện. Thiết bị xử lý cũng có thể bao gồm thành phần truyền thông (không được thể hiện trên hình vẽ) có thể được sử dụng để truyền thông điểm đến điểm giữa các thành phần khác nhau của thiết bị xử lý 1200. Thiết bị xử lý có thể được sử dụng trong hệ thống tính toán (không được thể hiện trên hình vẽ) bao gồm, nhưng không giới hạn ở, máy tính để bàn, máy tính dạng bảng, máy tính xách tay, netbook, máy tính notebook, thiết bị hỗ trợ cá nhân kỹ thuật số (Personal Digital Assistant, PDA), máy chủ, trạm làm việc, điện thoại di động, thiết bị điện toán di động, điện thoại thông minh, thiết bị Internet hoặc kiểu thiết bị điện toán khác bất kỳ. Theo một phương án khác, thiết bị xử lý có thể được sử dụng trong hệ thống hệ thống trên chip (System on Chip, SoC). Theo một phương án, SoC có thể bao gồm thiết bị xử lý và bộ nhớ. Bộ nhớ cho một hệ thống như vậy là bộ nhớ DRAM. Bộ nhớ DRAM có thể được đặt trên cùng chip với bộ xử lý và các thành phần hệ thống khác. Ngoài ra, các khối logic khác chẳng hạn như bộ điều khiển bộ nhớ hoặc bộ điều khiển đồ họa cũng có thể được đặt trên chip.

(Các) lõi bộ xử lý 1210 có thể thực thi các lệnh cho thiết bị xử lý 1200. Các lệnh này có thể bao gồm, nhưng không bị giới hạn ở, mạch logic nạp trước để nạp các lệnh, mạch logic giải mã để giải mã các lệnh, mạch logic thực thi để thực thi các lệnh và tương tự. Hệ thống tính toán có thể là biểu diễn của các hệ thống xử lý dựa vào họ Intel x86 của các bộ xử lý và/hoặc các bộ vi xử lý có sẵn từ Intel® Corporation of Santa Clara, Calif. (ví dụ, các bộ xử lý Intel Core i5, i7, v.v.), mặc dù các hệ thống khác bao gồm các thiết bị điện toán có các bộ vi xử lý khác, trạm làm việc kỹ thuật, hộp giải mã tín hiệu và tương tự cũng có thể được sử dụng. Theo một phương án, hệ thống điện toán ví dụ có thể thực thi phiên bản của hệ điều hành, phần mềm được nhúng, và/hoặc các giao diện người dùng đồ họa. Do đó, các phương án của sáng chế không bị giới hạn ở sự kết hợp cụ thể bất kỳ của hệ mạch phần cứng và phần mềm.

Trong một ví dụ minh họa, lõi xử lý 1210 có thể có kiến trúc micro bao gồm mạch logic bộ xử lý và các mạch. Các lõi bộ xử lý có các kiến trúc micro khác nhau có thể chia sẻ ít nhất một phần của tập hợp lệnh chung. Ví dụ, các kiến trúc thanh ghi tương tự có thể được thực hiện theo các cách khác nhau trong các kiến trúc micro khác nhau sử dụng các kỹ thuật khác nhau, bao gồm các thanh ghi vật lý chuyên dụng, một hoặc nhiều thanh ghi vật lý được cấp phát động sử dụng cơ chế đổi tên thanh ghi (ví dụ, sử dụng bảng viết tắt thanh ghi (Register Alias Table, RAT), bộ đệm sắp xếp lại (Reorder Buffer, ROB) và tệp thanh ghi ẩn).

Bộ điều khiển bộ nhớ 1220 có thể thực hiện các chức năng mà cho phép thiết bị xử lý truy nhập và truyền thông với bộ nhớ (không được thể hiện trên hình vẽ) bao gồm bộ nhớ khả biến và/hoặc bộ nhớ bất khả biến. Theo một số phương án, bộ điều khiển bộ nhớ 1220 có thể được đặt trên khuôn bộ xử lý được liên kết với thiết bị xử lý 1200, trong khi bộ nhớ được đặt ngoài khuôn bộ xử lý. Theo một số phương án, thiết bị xử lý bao gồm bộ phận đệm 1230 để đệm các lệnh và/hoặc dữ liệu. Bộ phận đệm 1230 bao gồm, nhưng không giới hạn ở, mức một (L1) 1232, mức hai (L2) 1234, và bộ đệm mức cuối cùng (Last Level Cache, LLC) 1236, hoặc cấu hình khác bất kỳ của bộ nhớ đệm trong thiết bị xử lý 1200. Theo một số phương án, bộ đệm L1 1232 và bộ đệm L2 1234 có thể chuyển dữ liệu đến và từ LLC 1236. Theo một phương án, bộ điều khiển bộ nhớ 1220 có thể được nối với LLC 1236 để chuyển dữ liệu giữa bộ phận đệm 1230 và bộ nhớ. Như được thể hiện, bộ phận đệm 1230 có thể được tích hợp vào các lõi xử lý 1210. Bộ phận đệm 1230 có thể lưu trữ dữ liệu (ví dụ, bao gồm các lệnh) được sử dụng bởi một hoặc nhiều thành phần của thiết bị xử lý 1200.

Theo một số phương án, thiết bị xử lý có thể thực thi một hoặc nhiều ứng dụng mức người dùng 1240 (ví dụ, ứng dụng đa luồng mức người dùng). Các ứng dụng mức người dùng 1240 có thể được thực thi bởi phần mềm hệ thống (không được thể hiện trên hình vẽ) được cài đặt ở thiết bị xử lý 1200. Các ví dụ về phần mềm hệ thống bao gồm, nhưng không bị giới hạn ở, một hoặc nhiều hệ điều hành, bộ giám sát máy ảo (Virtual Machine Monitor, VMM), siêu giám sát, và tương tự, và các sự kết hợp của chúng. Các ứng dụng mức người dùng 1240 có thể sử dụng các lệnh (ví dụ, các lệnh 1250) để điều khiển thiết bị xử lý như

được bọc lộ trong bản mô tả này. Các lệnh 1250 có thể biểu diễn các lệnh macro, các lệnh hợp ngữ, hoặc các lệnh mức máy được cấp cho lõi xử lý 1210 để thực thi.

Theo một số phương án, lõi xử lý 1210 hoạt động để thực hiện phương án về lệnh thủ tục gọi 1252 và phương án về lệnh thủ tục trả về 1254 được chỉ định bởi các lệnh 1250. Lệnh thủ tục gọi 1252 sẽ thường được gọi đơn giản là lệnh gọi, và lệnh thủ tục trả về 1254 sẽ thường được gọi đơn giản là lệnh trả về. Các lệnh gọi và trả về có thể biểu diễn các lệnh hoặc tín hiệu điều khiển của một tập hợp lệnh cụ thể cho thiết bị xử lý để thực thi ở mức quyền ưu tiên nhất định. Trong quá trình hoạt động, lõi xử lý 1210 có thể nhận lệnh gọi 1252. Ví dụ, lệnh gọi 1252 có thể được nhận từ thủ tục gọi được liên kết với các ứng dụng mức người dùng 1240 đang gọi thường trình con, thường trình, trình xử lý ngắt hoặc tương tự. Theo một số phương án, địa chỉ của lệnh trả về 1254 có thể được đẩy vào ngăn xếp dữ liệu được lưu trữ trong bộ nhớ khi các hàm hoặc thủ tục được gọi.

Nếu lệnh gọi 1252 làm cho thay đổi mức quyền ưu tiên hiện tại của thiết bị xử lý 1200, thì thiết bị xử lý thực hiện chuyển ngăn xếp từ ngăn xếp hiện tại sang ngăn xếp được xác định cho mức quyền ưu tiên mới. Ví dụ, lệnh gọi 1252 có thể làm cho thiết bị xử lý thay đổi mức quyền ưu tiên hiện tại để truy nhập các tài nguyên hệ thống nhất định không thể truy nhập ở mức quyền ưu tiên hiện tại. Theo một số phương án, một hoặc nhiều ngăn xếp bóng được lưu trữ trong bộ nhớ được sử dụng để giúp bảo vệ ngăn xếp dữ liệu bị giả mạo và/hoặc giúp làm tăng bảo mật máy tính. Thông tin được lưu trữ trên (các) ngăn xếp bóng có thể biểu diễn thông tin liên quan đến địa chỉ trả về liên quan đến lệnh trả về 1254 (ví dụ, địa chỉ trả về thực tế, thông tin để xác thực/xác minh các địa chỉ trả về, thông tin địa chỉ trả về khác).

Để nhận dạng ngăn xếp bóng hiện tại ở mức quyền ưu tiên cụ thể, thiết bị xử lý bao gồm số lượng thanh ghi ngăn xếp bóng 1270 mà mỗi thanh ghi được liên kết với mức quyền ưu tiên nhất định. Là ví dụ minh họa, thiết bị xử lý có thể bao gồm bốn (4) thanh ghi được gọi là IA32_PLx_SSP trong đó x biểu diễn mức quyền ưu tiên (ví dụ, 0, 1, 2 hoặc 3). Theo một số phương án, các thanh ghi ngăn xếp bóng 1270 có thể là các thanh ghi mô hình cụ thể (Model Specific Register, MSR) hoặc các thanh ghi đa năng (General Purpose Register, GPR). Theo các phương án khác, các kiểu cấu trúc khác nhau có thể được sử dụng làm

thanh ghi 1270 miễn là chúng có khả năng lưu trữ và cung cấp dữ liệu như được mô tả trong bản mô tả này.

Khi khởi động thiết bị xử lý 1200, phần mềm hệ thống được liên kết với thiết bị có thể lập trình con trỏ ngăn xếp (SSP) 1275 vào mỗi trong số các thanh ghi ngăn xếp bóng 1270. Con trỏ ngăn xếp (SSP) 1275 hoạt động để nhận dạng ngăn xếp bóng cụ thể đối với mức quyền ưu tiên được nhận dạng bởi thanh ghi. Ví dụ, SSP 1275 có thể là địa chỉ tuyến tính trong bộ nhớ trỏ đến đỉnh của ngăn xếp bóng. Theo một phương án, thiết bị xử lý có thể bao gồm mạch logic bảo vệ ngăn xếp bóng 1280 để điều khiển việc truy nhập vào các ngăn xếp bóng qua các thanh ghi ngăn xếp bóng 1270 trong quá trình chuyển vòng.

Mạch logic bảo vệ ngăn xếp bóng 1280 cũng có thể bắt buộc việc tải và lưu trữ dành cho ngăn xếp bóng tạo ra tình trạng lỗi nếu địa chỉ mà tải hoặc lưu trữ này được thực hiện không được đánh dấu trong bảng trang là thuộc kiểu ngăn xếp bóng. Mạch logic bảo vệ ngăn xếp bóng 1280 cũng có thể ngăn chặn việc lưu trữ vào các bộ nhớ ngăn xếp bóng bằng cách lưu trữ phần mềm khởi tạo thông qua các lệnh, chẳng hạn như “MOV” và “XSAVE”. Theo một số phương án, mạch logic bảo vệ ngăn xếp bóng 1280 cũng có thể bao gồm chế độ ngăn xếp bóng tạo cấu hình được tùy chọn 1285. Chế độ ngăn xếp bóng 1285 có thể là bộ chỉ báo bit được tạo cấu hình để cho phép và vô hiệu hóa theo cách khác việc sử dụng các ngăn xếp bóng bởi thiết bị xử lý 1200. Theo cách khác, các ngăn xếp bóng có thể luôn được cho phép và chế độ ngăn xếp bóng 1285 có thể được lược bỏ một cách tùy ý. Theo cách khác, có thể có chế độ ngăn xếp bóng tương ứng với mỗi trong số các mức quyền ưu tiên.

FIG.13 minh họa hệ thống 1300 bao gồm bộ nhớ 1301 để hỗ trợ các phần mở rộng của bộ xử lý nhằm bảo vệ ngăn xếp trong quá trình chuyển vòng theo một phương án. Trong ví dụ này, bộ nhớ 1301 lưu trữ một hoặc nhiều ngăn xếp dữ liệu, chẳng hạn như ngăn xếp dữ liệu 1303. Các ngăn xếp dữ liệu thường được gọi là ngăn xếp lệnh gọi, ngăn xếp dữ liệu, hoặc chỉ là ngăn xếp. Ngăn xếp dữ liệu 1303 có thể biểu diễn cấu trúc dữ liệu kiểu ngăn xếp hoạt động để lưu trữ cả dữ liệu 1305 và thông tin điều khiển 1307 để hướng dẫn điều khiển dòng lệnh được thực thi. Thông thường, có ngăn xếp dữ liệu 1303 đối với mỗi mức quyền ưu tiên được liên kết với thiết bị xử lý 1200. Như vậy, khi thiết bị xử lý chuyển mức quyền

ưu tiên thì nó cũng chuyển ngăn xếp dữ liệu 1303 giống như đối với ngăn xếp bóng như được mô tả trong bản mô tả này. Dữ liệu 1305 đối với mỗi ngăn xếp dữ liệu có thể biểu diễn kiểu dữ liệu bất kỳ trong số nhiều kiểu dữ liệu có thể được đẩy trên ngăn xếp dữ liệu 1301 (ví dụ, các tham số và dữ liệu khác chuyển đến các thường trình con, v.v.). Nói chung, thông tin điều khiển 1307 có thể bao gồm một hoặc nhiều địa chỉ trả về cho một hoặc nhiều lệnh gọi thủ tục được thực hiện trước đó. Các địa chỉ trả về này có thể biểu diễn các địa chỉ lệnh mà thủ tục được gọi là để trả về dòng điều khiển khi thủ tục được gọi kết thúc và trả về.

Như được thể hiện trên FIG.13, bộ nhớ 1301 cũng bao gồm các ngăn xếp bóng 1309. Các ngăn xếp bóng 1309 được sử dụng để giúp bảo vệ ngăn xếp dữ liệu 1303 từ kiểu tấn công dòng điều khiển nhất định. Các ngăn xếp bóng 1309 có thể biểu diễn các cấu trúc dữ liệu kiểu ngăn xếp bổ sung tách biệt khỏi ngăn xếp dữ liệu 1303. Theo một số phương án, các ngăn xếp bóng 1309 có thể được sử dụng để lưu trữ thông tin được liên kết với thông tin điều khiển 1307 của lệnh được thực thi mà không phải dữ liệu 1305. Theo một ví dụ minh họa, các ngăn xếp bóng 1309 có thể lưu trữ bộ mô tả đoạn mã (Code Segment, CS), chẳng hạn như CS 1340, chỉ định đoạn mã trong bộ nhớ 1301 đang được thực thi và con trỏ lệnh (EIP), chẳng hạn như EIP 1350, có thể nhận dạng địa chỉ trả về liên quan đến lệnh trong ngăn xếp dữ liệu 1303 cũng như thông tin khác. Theo một số phương án, bộ nhớ được sử dụng để tạo các ngăn xếp bóng có thể được đánh dấu là kiểu bộ nhớ ngăn xếp bóng trong các bảng trang, sao cho các tải và thiết bị lưu trữ vào lỗi ngăn xếp bóng nếu chúng không thuộc bộ nhớ kiểu ngăn xếp bóng. Tương tự, các thiết bị lưu trữ không dành cho lỗi ngăn xếp bóng nếu chúng được tạo cho bộ nhớ được đánh dấu là bộ nhớ ngăn xếp bóng.

Mặc dù hệ thống 1300 có thể bao gồm các ngăn xếp bóng 1309, nhưng chỉ một ngăn xếp bóng ở một thời điểm có thể được chọn là ngăn xếp bóng hiện tại 1310. Về vấn đề này, các ngăn xếp bóng 1309 có thể hoạt động riêng rẽ ở chế độ mức người dùng không có quyền ưu tiên (ví dụ, mức quyền ưu tiên vòng 3) hoặc ở chế độ mức quyền ưu tiên được ưu tiên hoặc giám sát (mức quyền ưu tiên vòng 0, vòng 1, hoặc vòng 2). Theo một số phương án, mỗi trong số các ngăn xếp bóng 1309 bao gồm con trỏ ngăn xếp bóng (Shadow Stack Pointer, SSP), chẳng hạn như SSP 1320, hoạt động để nhận dạng đỉnh của ngăn xếp bóng.

Ví dụ, SSP 1320 có thể là địa chỉ tuyến tính trong bộ nhớ 1301 trở đến đỉnh của ngăn xếp bóng hiện tại. Như được lưu ý trên đây, SSP 1320 có thể được lưu trong một trong số các thanh ghi ngăn xếp bóng 1270 được liên kết với mức quyền ưu tiên cụ thể.

Như được thể hiện, hệ thống 1300 cũng có thể bao gồm thiết bị xử lý trên FIG.12. Theo một số phương án, thiết bị xử lý có thể có mức quyền ưu tiên hiện tại để điều khiển việc thực thi ứng dụng trong hệ thống 1300. Trong trường hợp nhất định, thiết bị xử lý có thể thay đổi mức quyền ưu tiên hiện tại của nó sang mức quyền ưu tiên mới để điều khiển khả năng truy nhập ứng dụng vào các tài nguyên hệ thống 1300. Ví dụ, thiết bị xử lý có thể thay đổi mức quyền ưu tiên hiện tại của nó để đáp lại các kiểu hoạt động nhất định, chẳng hạn như thủ tục được gọi, trình xử lý ngắt/ngoại lệ hoặc để đáp lại kiểu hoạt động khác. Theo một số phương án, thiết bị xử lý có thể sử dụng mạch logic bảo vệ ngăn xếp bóng 1280 để bảo vệ ngăn xếp này trong quá trình chuyển mức quyền ưu tiên.

Trong một ví dụ về chuyển mức quyền ưu tiên, thiết bị xử lý có thể chuyển từ mức quyền ưu tiên hiện tại (Current Privilege Level, CPL) ở chế độ người dùng (ví dụ, CPL3) sang mức quyền ưu tiên mới ở chế độ giám sát (ví dụ, CPL<3). Kiểu chuyển mức quyền ưu tiên này có thể được khởi tạo, ví dụ, bằng lệnh gọi qua cổng gọi yêu cầu thay đổi mức quyền ưu tiên cao hơn hoặc bằng cách gọi trình xử lý ngắt/ngoại lệ. Để đáp lại quá trình chuyển mức quyền ưu tiên chế độ người dùng sang chế độ giám sát, thiết bị xử lý kiểm tra xem liệu các ngăn xếp bóng 1309 có cho phép thiết bị xử lý ở mức quyền ưu tiên mới hay không. Ví dụ, thiết bị xử lý có thể kiểm tra thiết lập của bộ chỉ báo chế độ ngăn xếp bóng 1285 để xác định xem liệu các ngăn xếp bóng 1309 được cho phép hay vô hiệu hóa. Nếu các ngăn xếp bóng 1309 được cho phép, thì SSP hiện tại được liên kết với chế độ người dùng được lưu vào một trong số các thanh ghi ngăn xếp bóng 1270. Bởi vậy, ngăn xếp bóng mới 1310 được thiết lập cho chế độ giám sát sử dụng SSP, chẳng hạn như SSP 1320, được lưu trữ trong thanh ghi được liên kết với mức quyền ưu tiên mới. Theo một số phương án, chế độ người dùng SSP được lưu vào thanh ghi ngăn xếp bóng để thiết bị xử lý ở chế độ giám sát có thể sửa đổi chế độ người dùng SSP khi cần, chẳng hạn như thực hiện mở cuộn ngăn xếp bóng để xóa các mục nhập khỏi ngăn xếp và các kiểu sửa lỗi khác trước khi trở lại chế độ người dùng.

Khi thiết bị xử lý truy xuất SSP 220 từ thanh ghi được liên kết mức quyền ưu tiên mới, thì thiết bị xử lý tự động thực hiện một số kiểm tra để xác minh SSP được lập trình trong thanh ghi bằng cách sử dụng mạch logic nhất định của mạch logic bảo vệ ngăn xếp bóng 1280, chẳng hạn như mạch logic xác minh và thiết lập mạch logic bận 1325. Việc kiểm tra tự động này bao gồm, nhưng không bị giới hạn ở, xác minh rằng SSP 1320 được lập trình trong thanh ghi trở đến đỉnh của ngăn xếp bóng giám sát (ví dụ, ngăn xếp bóng 1310, ngăn xếp bóng được nhận dạng bởi SSP 1320 có thể được sử dụng cho chế độ hiện tại của nền tảng phần cứng được liên kết với thiết bị xử lý và ngăn xếp bóng không được tải lên luồng xử lý khác bất kỳ. Các kiểm tra này có thể được sử dụng để ngăn chặn các cuộc tấn công nhất định, chẳng hạn như một luồng chéo trong đó kẻ tấn công có thể trở thành ghi trên hai luồng bộ xử lý khác nhau đến cùng một ngăn xếp bóng để thao tác địa chỉ trả về được sử dụng trên một bộ xử lý bằng cách thực hiện các lệnh gọi trên bộ xử lý thứ hai.

Để thực hiện kiểm tra để xác minh SSP 1320, phần mềm hệ thống (ví dụ, nhân hoặc OS) của thiết bị xử lý nhận dạng mã thông báo 1330 ở đỉnh của ngăn xếp bóng 1310 được trả bởi SSP. Theo một số phương án, thiết bị xử lý có thể (ví dụ, tự động) tải (ví dụ, 8 byte) mã thông báo 1330 từ địa chỉ được chỉ định bởi SSP 1320. Theo một phương án, việc tải mã thông báo 1330 khóa mã thông báo và/hoặc vị trí (ví dụ, dòng đệm) để ngăn chặn mã thông báo bị sao chép hoặc sửa đổi bởi lỗi bộ xử lý hoặc thiết bị xử lý khác. Theo một phương án, việc tải mã thông báo được thực hiện bởi thiết bị xử lý sử dụng hoạt động ngăn xếp bóng cụ thể, sao cho nếu địa chỉ mà việc tải hoặc lưu trữ được thực hiện không bộ nhớ được đánh dấu là bộ nhớ ngăn xếp bóng trong bảng trang thì nó gây ra lỗi.

Mã thông báo 1330 bao gồm các bit được sử dụng để xác minh SSP 1320 được lập trình trong thanh ghi. Theo một số phương án, thiết bị xử lý phân tích các bit để nhận dạng các bit được thiết lập biểu diễn địa chỉ tuyến tính của mã thông báo 1330, bộ chỉ báo bận có thể là bit được xác định của mã thông báo được thiết lập thành giá trị (ví dụ, 1 hoặc 0) chỉ báo xem liệu ngăn xếp bóng 1310 được tải lên bộ xử lý logic bất kỳ hay chưa, bộ chỉ báo chờ có thể là một bit xác định khác của mã thông báo được thiết lập thành giá trị (ví dụ, 1 hoặc 0) chỉ báo xem liệu ngăn xếp bóng 1310 có được sử dụng ở chế độ bit nhất định hay

không (ví dụ, 32-bit hoặc 64-bit) cũng như các thông tin liên quan khác.

Theo mạch logic xác minh 1325, thiết bị xử lý kiểm tra bộ chỉ báo bận không được thiết lập (ví dụ, 0) chỉ báo rằng ngăn xếp bóng 1310 không được tải lên bộ xử lý logic bất kỳ. Thiết bị xử lý cũng kiểm tra nền tảng phần cứng của hệ thống 1300 đang hoạt động ở chế độ trùng với chế độ bit được biểu diễn bởi bộ chỉ báo chờ. Ví dụ, dựa vào bộ chỉ báo chờ này, cờ thiết bị xử lý có thể xác định rằng ngăn xếp bóng 1310 để sử dụng trên máy 32-bit hoặc 64 bit. Thiết bị xử lý có thể sau đó kiểm tra, ví dụ, thanh ghi nền tảng được liên kết với nền tảng phần cứng để xác minh rằng nền tảng này ở chế độ hoạt động tương ứng (ví dụ, 32-bit hoặc 64-bit). Thiết bị xử lý cũng kiểm tra địa chỉ tuyến tính được ghi trong mã thông báo 1330 trùng với địa chỉ tuyến tính của SSP 1320.

Việc kiểm tra này trùng với các địa chỉ tuyến tính xác minh rằng thanh ghi 1270 đang trở đến đỉnh của ngăn xếp bóng hợp lệ do địa chỉ tuyến tính của vị trí 8 trên ngăn xếp bóng sẽ không xuất hiện trong nội dung của 8 byte này trên ngăn xếp bóng trừ khi được tạo ra bởi phần mềm hệ thống để khởi tạo mã thông báo 1303. Như vậy, mã thông báo 1330 tạo thành bộ chỉ báo mà con trỏ ngăn xếp bóng được lưu trữ trong thanh ghi 1270 đang trở đến đỉnh của ngăn xếp bóng hợp lệ. Theo một số phương án, thiết bị xử lý bắt buộc ngăn xếp bóng được căn chỉnh thành 4 byte hoặc 8 byte phụ thuộc vào việc xem liệu thiết bị xử lý ở chế độ hoạt động 32 bit hay 64 bit, và do đó, 2 bit thấp hơn của con trỏ ngăn xếp bóng 1320 luôn bằng 0 và có thể được sử dụng để lưu bộ chỉ báo bận và cờ của mã thông báo 1330. Nếu tất cả các kết quả kiểm tra ở trên chỉ báo chính xác, sau đó bộ chỉ báo bận của mã thông báo có thể được thiết lập thành giá trị (ví dụ, 1) để chỉ báo rằng ngăn xếp bóng được trở bởi SSP 1320 khi sử dụng ở mức quyền ưu tiên mới. Cần lưu ý rằng trong khi các kiểm tra này đang được thực hiện, vị trí bộ nhớ (ví dụ dòng đệm) chứa mã thông báo 1330 được khóa bởi thiết bị xử lý để không thể sửa đổi nó (ví dụ đặt bận bởi bộ xử lý logic khác). Sau khi thiết lập bit bận, bộ xử lý sẽ giải phóng khóa trên vị trí bộ nhớ này. Thực hiện các hoạt động kiểm tra này với khóa được giữ trên mã thông báo đảm bảo rằng không có vấn đề về thời gian kiểm tra đến thời điểm sử dụng. Nếu không, nếu bất kỳ kiểm tra nào ở trên không thành công, thì thiết bị xử lý có thể tạo ra lỗi hoặc cảnh báo có thể được truyền, ví dụ, đến hệ điều hành được liên kết với thiết bị xử lý 1200. Theo một số phương án, nếu lỗi

được tạo ra, khóa trên vị trí bộ nhớ chứa mã thông báo 1330 sẽ được giải phóng mà không sửa đổi nội dung của vị trí đó.

Khi quay trở lại mức quyền ưu tiên chế độ người dùng từ chế độ giám sát, chẳng hạn như chuyển đổi chế độ CPL 0/1/2-> CPL 3, thiết bị xử lý thực hiện một số kiểm tra bằng cách sử dụng một số mạch logic nhất định, chẳng hạn như xác minh và xóa logic bận 1325. Kiểu chuyển mức quyền ưu tiên này có thể được khởi tạo, ví dụ, bằng lệnh trả về ngắt (Interrupt Return, IRET) hoặc lệnh "FAR RET" chuyển địa chỉ tiếp theo sẽ được thực thi đến con trỏ lệnh của CPU hoặc các kiểu lệnh trả về khác.

Là một phần của quá trình chuyển trở lại mức quyền ưu tiên chế độ người dùng, thiết bị xử lý theo chỉ dẫn của mạch logic xác minh và xóa bận 1335 kiểm tra xem liệu bộ chỉ báo bận trong mã thông báo 1330 có được thiết lập hay không và xem liệu địa chỉ tuyến tính được ghi trong mã thông báo có trùng với địa chỉ của SSP 1320 hay không và nếu SSP 1320 đang trỏ đến địa chỉ được căn chỉnh 8 byte. Nếu tất cả các kiểm tra ở trên dẫn đến chỉ báo chính xác, thì thiết bị xử lý sẽ đánh dấu mã thông báo 1330 là "rảnh" bằng cách xóa chỉ báo bận trong mã thông báo 1330. Cần lưu ý rằng mã thông báo được tải từ địa chỉ bộ nhớ được trỏ đến bởi SSP 1320 sử dụng tải được khóa sao cho vị trí (ví dụ dòng đệm) được khóa khỏi các sửa đổi tiếp theo. Nếu việc xác minh thành công sau đó bit bận được xóa và khóa được giải phóng. Nếu việc xác minh không thành công sau đó khóa sẽ được giải phóng mà không cần sửa đổi nội dung của vị trí đó. Sau đó, thiết bị xử lý truy xuất SSP đối với chế độ người dùng từ thanh ghi ngăn xếp bóng thích hợp 1270 được liên kết với mức quyền ưu tiên. Nếu không, thiết bị xử lý có thể tạo ra lỗi chỉ báo rằng mã thông báo 1330 không hợp lệ.

Ví dụ khác về chuyển mức quyền ưu tiên, thiết bị xử lý có thể chuyển từ mức quyền ưu tiên chế độ giám sát hiện tại sang chế độ giám sát mới cao hơn, chẳng hạn như chuyển chế độ CPL2->CPL1, CPL1->CPL0 hoặc CPL2->CPL0. Kiểu chuyển mức quyền ưu tiên này có thể được khởi tạo, ví dụ, bằng lệnh gọi qua cổng gọi yêu cầu thay đổi mức quyền ưu tiên cao hơn hoặc bằng cách gọi và trình xử lý ngắt/ngoại lệ. Đối với việc chuyển này, thiết bị xử lý chọn SSP 1320 cho mức quyền ưu tiên mới từ một thích hợp trong số các thanh ghi ngăn xếp bóng 1270 đối với mức này. Khi thiết bị xử lý truy xuất SSP 1320 từ

thanh ghi được liên kết mức quyền ưu tiên mới, thì thiết bị xử lý thực hiện một số kiểm tra để xác minh SSP được lập trình trong thanh ghi.

Theo một số phương án, thiết bị xử lý thực hiện kiểm tra bằng cách sử dụng mạch logic xác minh và thiết lập bận 1325 như được mô tả ở trên. Nếu việc kiểm tra thành công, thì thiết bị xử lý đánh dấu ngăn xếp bóng 1310 là bận bằng cách thiết lập bộ chỉ báo bận trong mã thông báo 1330. Sau đó, thiết bị xử lý đẩy bộ chọn CS 1340 (ví dụ, bộ chọn đoạn mã) nhận dạng đoạn mã trong bộ nhớ 1301 đang được thực thi, EIP 1350 nhận dạng thông tin địa chỉ trả về và SSP 1360 được liên kết với mức quyền ưu tiên hiện tại trên ngăn xếp bóng có mức quyền ưu tiên mới. Theo một số phương án, thiết bị xử lý có thể đẩy một con trỏ lệnh tuyến tính (Linear Instruction Pointer, LIP) (là EIP cộng với cơ sở đoạn mã) trên ngăn xếp bóng của mức quyền ưu tiên mới thay vì EIP 1350. Điều này có thể được thực hiện để phát hiện nỗ lực bất kỳ để thay đổi cơ sở đoạn mã giữa lệnh gọi và trả về từ lệnh gọi này.

Khi quay trở lại chế độ giám sát trước đó từ chế độ giám sát mới, thiết bị xử lý đảm bảo rằng những lần trở lại đó sẽ đến cùng địa chỉ nơi lệnh gọi trước đó được thực hiện. Ví dụ, thiết bị xử lý xác minh rằng CS 1340 và EIP 1350 (hoặc LIP) từ ngăn xếp bóng 1310 trùng với các giá trị tương ứng trên ngăn xếp dữ liệu 1303. Nếu trùng khớp, thì sau đó thiết bị xử lý loại bỏ hoặc bật CS 1340, EIP 1350 và SSP 1360 khỏi ngăn xếp bóng 1310 để thiết lập SSP mới được sử dụng ở mức quyền ưu tiên mới. Theo một số phương án, các hoạt động đẩy và bật được thực hiện bằng cách sử dụng các hoạt động tải và lưu trữ ngăn xếp bóng cụ thể, chẳng hạn như nếu các hoạt động đối với bộ nhớ không được đánh dấu là bộ nhớ ngăn xếp bóng trong các bảng trang thì các hoạt động sẽ gây ra lỗi.

Là một phần của quá trình truyền, thiết bị xử lý theo chỉ dẫn của mạch logic xác minh và xóa bận 1335 kiểm tra xem liệu bộ chỉ báo bận trong mã thông báo 1330 của ngăn xếp bóng được trỏ đến SSP 1320 có được thiết lập hay không và địa chỉ tuyến tính được ghi trong mã thông báo 1330 có trùng với địa chỉ của SSP 1320 hay không. Nếu tất cả các kiểm tra ở trên dẫn đến chỉ báo chính xác, thì thiết bị xử lý sẽ đánh dấu mã thông báo 1330 là “rảnh” bằng cách xóa chỉ báo bận trong mã thông báo 1330.

Theo một phương án khác nữa về chuyển mức quyền ưu tiên có thể dựa vào sự chuyển chế độ của thiết bị xử lý từ chế độ người dùng sang mức quyền ưu tiên được tin tưởng nhất, chẳng hạn như chuyển chế độ CPL3->CPL0. Kiểu chuyển mức quyền ưu tiên này có thể được khởi tạo, ví dụ, bằng cách sử dụng lệnh nhất định như SYSCALL hoặc SYSETER được thiết kế để gọi ra các lệnh gọi hàm hệ thống trong đó một số lệnh gọi hệ thống không sử dụng ngăn xếp để thực hiện các hoạt động gọi hệ thống. Trong quá trình chuyển này, thiết bị xử lý lưu SSP hiện tại trong thanh ghi được liên kết với chế độ người dùng và chuyển sang ngăn xếp bóng được liên kết với mức quyền ưu tiên CPL0 bằng cách truy xuất SSP trong thanh ghi tương ứng. Vì thiết bị xử lý có thể đang thực hiện hoạt động quan trọng của hệ thống nên nó không xác minh SSP, nhưng vẫn chọn một trong số các ngăn xếp bóng trong tập hợp các ngăn xếp bóng là SSP hiện tại đối với mức quyền ưu tiên mới. Ví dụ, kiểu chuyển mức quyền ưu tiên này có thể được khởi tạo, ví dụ, bởi lệnh gọi hệ thống nhanh không sử dụng ngăn xếp để thực hiện các hoạt động gọi hệ thống. Theo một số phương án, nếu lệnh gọi hệ thống cần sử dụng ngăn xếp để thực hiện các lệnh gọi sau đó nó có thể được tạo cấu hình để sử dụng lệnh mới được cấp bởi thiết bị xử lý để thực hiện xác minh và đánh dấu ngăn xếp bóng là bận, chẳng hạn như thực hiện trong mạch logic xác minh và thiết lập bận 1325. Hoạt động này của mạch logic xác minh và thiết lập bận 1325 được thực hiện trên ngăn xếp bóng được chọn, nhưng không được xác minh và thiết lập bận trước đó.

Khi quay trở lại chế độ người dùng trước đó từ mức quyền ưu tiên được tin tưởng nhất, chẳng hạn như chuyển chế độ CPL0->CPL3, thiết bị xử lý phục hồi ngăn xếp bóng chế độ người dùng bằng cách truy nhập SSP được lưu trữ trong thanh ghi tương ứng với mức quyền ưu tiên chế độ người dùng. Kiểu chuyển mức quyền ưu tiên này có thể được khởi tạo, ví dụ, bằng các lệnh trở lại hoặc thoát khỏi hệ thống (ví dụ, SYSRET hoặc SYSEXIT). Trước khi trở lại CPL3, nếu lệnh gọi hệ thống sử dụng lệnh mới được cấp bởi thiết bị xử lý để xác minh và đánh dấu ngăn xếp bóng là bận, thì lệnh bổ sung được cấp bởi thiết bị xử lý để xác minh và đánh dấu ngăn xếp bóng là rảnh, chẳng hạn như được sử dụng trong mạch logic xác minh và xóa bận 1335.

FIG.14 minh họa hệ thống 1400 bao gồm bộ nhớ 1301 trên FIG.13 để chuyển các

tác vụ là một phần của ngắt hoặc ngoại lệ phân phối theo một phương án. Theo một số phương án, sáng chế đề xuất các kỹ thuật để bù chuyển đổi ngăn xếp bóng trong quá trình phân phối các lần ngắt và thực thi nhất định bằng cách sử dụng bảng con trỏ ngăn xếp bóng ngắt 1410 thay vì các thanh ghi ngăn xếp bóng 1270 đang được sử dụng để chọn con trỏ ngăn xếp bóng cho mức quyền ưu tiên mà trình xử lý ngắt hoặc ngoại lệ này cần được thực thi. Phần mềm hệ thống của thiết bị xử lý 1210 có thể lập trình địa chỉ của bảng 1410 vào thanh ghi, chẳng hạn như thanh ghi địa chỉ bảng SSP ngắt 1420. Bảng 1410 được sử dụng để lưu trữ SSP 1320 được nhận dạng bởi thiết bị xử lý trong quá trình chuyển vòng như được mô tả ở trên đối với hệ thống 1300 hoặc làm cho chuyển tác vụ thậm chí không chuyển vòng nếu ngắt hoặc ngoại lệ được tạo cấu hình để hoạt động trên ngăn xếp bóng mới. Về vấn đề này, việc chọn ngăn xếp bóng mới từ một trong số các con trỏ ngăn xếp bóng trong bảng ngăn xếp bóng ngắt 1410 bằng cách sử dụng bộ chọn 1430 được tạo cấu hình cho ngắt trong bảng mô tả ngắt (Interrupt Descriptor Table, IDT).

Khi hoạt động, khi ngoại lệ hoặc ngắt xảy ra, thì thiết bị xử lý có thể phản hồi bằng cách chuyển điều khiển sang chuỗi mã nhất định nhằm đối phó với sự kiện này. Chuỗi mã này thường chạy ở chế độ giám sát. Địa chỉ của chuỗi mã đặc biệt này, hoặc thường trình, được truy xuất bởi thiết bị xử lý từ bảng mô tả ngắt (Interrupt Descriptor Table, IDT), sử dụng chỉ số vào IDT cụ thể cho kiểu ngoại lệ đã xảy ra (ví dụ, một chỉ số có thể tương ứng với lỗi trang trong khi một trang khác tương ứng với ngoại lệ tràn). Mục nhập IDT này có thể còn có chỉ số vào bảng ngăn xếp bóng ngắt 1410, sao cho khi giá trị chỉ số này khác không thì chỉ báo rằng việc chuyển đổi ngăn xếp bóng phải được thực hiện để ngắt hoặc ngoại lệ ngay cả khi không cần sự thay đổi quyền ưu tiên để phân phối ngắt hoặc ngoại lệ. Thiết bị xử lý có thể sử dụng chỉ số được liên kết với IDT để chỉ bảng con trỏ ngăn xếp bóng ngắt 1410 để nhận dạng mục nhập lưu trữ SSP 1320. Ví dụ, mục nhập trong bảng con trỏ ngăn xếp bóng ngắt 1410 cũng có thể dành riêng cho kiểu ngoại lệ đã xảy ra. Sau đó SSP 1320 có thể được sử dụng bởi thiết bị xử lý ở mức quyền ưu tiên mới được liên kết với ngắt. Sau khi lựa chọn SSP, thiết bị xử lý gọi ra mạch logic nhất định (ví dụ, mạch logic xác minh và thiết lập bận 1325, để đảm bảo rằng SSP 1320 được lưu trữ trong bảng ngăn xếp bóng ngắt 1410 chỉ đến đỉnh của ngăn xếp bóng hợp lệ cho chế độ hoạt động đó và ngăn xếp bóng không bận).

FIG.15 minh họa hệ thống 1500 bao gồm bộ nhớ 1301 trên FIG.13 để chuyển các tác vụ là một phần chuyển đổi tác vụ theo một phương án. Trong hoạt động chuyển đổi tác vụ, thiết bị xử lý có thể thực thi lệnh chuyển điều khiển làm cho chuyển đổi tác vụ từ tác vụ đang chạy hiện tại sang tác vụ mới. Theo một số phương án, các tác vụ có thể biểu diễn các đơn vị công việc mà thiết bị xử lý có thể điều động, thực hiện hoặc tạm dừng. Ví dụ, các tác vụ có thể được sử dụng để thực thi một hoặc nhiều trong số chương trình, tác vụ hoặc quy trình, tiện ích dịch vụ hệ điều hành, trình xử lý ngắt hoặc ngoại lệ hoặc một tiện ích nhân hoặc thực thi. Theo một khía cạnh, chuyển đổi tác vụ có thể được thực hiện khi các lệnh thủ tục gọi được thực hiện, hoặc khi ngắt hoặc ngoại lệ xảy ra, hoặc ngược lại.

Khi thiết bị xử lý thực hiện chuyển đổi tác vụ từ tác vụ hiện tại sang tác vụ mới, thì thiết bị này lưu thông tin trạng thái của tác vụ hiện tại trong cấu trúc dữ liệu trong bộ nhớ 1301 được gọi là đoạn trạng thái tác vụ (Task State-Segment, TSS), chẳng hạn như TSS hiện tại 1520, tương ứng với tác vụ hiện tại, và thông tin trạng thái của tác vụ mới 1527 có thể được tải hoặc được truy xuất từ TSS khác (ví dụ, TSS mới 1525) tương ứng với tác vụ mới 1527. Thông tin trạng thái này có thể bao gồm, nhưng không bị giới hạn ở, thông tin được yêu cầu để thực hiện chuyển đổi tác vụ, chẳng hạn như các con trỏ ngăn xếp và lệnh và các địa chỉ cơ sở ánh xạ 1210.

TSS biểu diễn cấu trúc dữ liệu được quản lý phần cứng trong bộ nhớ trên máy tính mà có thể được sử dụng để lưu trữ thông tin về các tác vụ. Ví dụ, TSS bao gồm thông tin cần để thiết bị xử lý 100 thực hiện chuyển đổi tác vụ, chẳng hạn như các con trỏ ngăn xếp. Mỗi TSS cho các tác vụ khác nhau có thể bao gồm các con trỏ ngăn xếp đến các ngăn xếp khác nhau (ví dụ, các ngăn xếp dữ liệu được sử dụng để lưu trữ cả thông tin điều khiển và dữ liệu) cho các mức quyền ưu tiên khác nhau. Các con trỏ ngăn xếp được sử dụng để nhận dạng ngăn xếp phù hợp khi các thay đổi mức quyền ưu tiên được thực hiện do hoạt động chuyển tác vụ. Ví dụ, hoạt động chuyển tác vụ có thể bao gồm chuyển đổi tác vụ hiện tại ở mức quyền ưu tiên hiện tại sang tác vụ mới ở mức quyền ưu tiên mới.

Theo một số phương án, các ngăn xếp bóng như được mô tả trong bản mô tả này có thể được sử dụng để bảo vệ các ngăn xếp dữ liệu trong quá trình chuyển mức quyền ưu tiên từ tác vụ hiện tại sang tác vụ mới. Như được lưu ý ở trên, ngăn xếp bóng là ngăn xếp thứ

hai tách biệt khỏi ngăn xếp dữ liệu để lưu trữ và bảo vệ thông tin từ ngăn xếp dữ liệu được sử dụng trong các hoạt động chuyển điều khiển. Theo một phương án, thông tin chọn ngăn xếp bóng, chẳng hạn như con trỏ ngăn xếp bóng 1530, có thể được lưu trữ trong cấu trúc TSS trong kiến trúc bộ xử lý nhất định (ví dụ, lõi Intel i7, i5, và các bộ xử lý Xeon).

Tham chiếu đến FIG.15, thiết bị xử lý đang hoạt động phản hồi chuyển đổi tác vụ từ tác vụ hiện tại sang tác vụ mới bằng cách nhận dạng TSS 1520 được liên kết với tác vụ mới. Về vấn đề này, chuyển tác vụ có thể được khởi tạo, ví dụ, bởi lệnh gọi, ngắt/ngoại lệ hoặc khác. Theo một số phương án, thiết bị xử lý truy xuất SSP 1530 được lưu trữ trong TSS 1520. SSP 1530 nhận dạng ngăn xếp bóng 1510 để sử dụng tác vụ mới.

Theo một số phương án, thiết bị xử lý xác minh SSP 1530 bằng cách thực hiện các kiểm tra sử dụng mạch logic xác minh và thiết lập bận 1325 như được mô tả ở trên. Nếu các kiểm tra thành công, thì thiết bị xử lý đánh dấu ngăn xếp bóng 1310 là bận bằng cách thiết lập bộ chỉ báo bận trong mã thông báo 1540. Sau đó, thiết bị xử lý đẩy con trỏ CS 1550 nhận dạng đoạn mã của tác vụ hiện tại đang được thực thi, EIP 1560 nhận dạng thông tin địa chỉ trả về liên quan đến tác vụ hiện tại và SSP 1570 được liên kết với mức quyền ưu tiên hiện tại của tác vụ hiện tại vào ngăn xếp bóng mới.

Khi quay trở lại từ tác vụ mới đến tác vụ trước đó, thiết bị xử lý đảm bảo rằng các kết quả trả về đó đến cùng địa chỉ của tác vụ trước đó. Ví dụ, thiết bị xử lý xác minh rằng CS 1550 và EIP 1560 từ ngăn xếp bóng 1510 trùng với các giá trị tương ứng trong TSS được liên kết với tác vụ trước đó. Nếu trùng khớp, thì thiết bị xử lý sẽ xóa hoặc bật CS 1550, EIP 1560 và SSP 1570 ra khỏi ngăn xếp bóng 1510 để thiết lập SSP hiện tại được sử dụng trong tác vụ trước đó. Là một phần của quá trình chuyển trở lại tác vụ trước đó, thiết bị xử lý theo chỉ dẫn của mạch logic xác minh và xóa bận 1335 kiểm tra xem liệu bộ chỉ báo bận trong mã thông báo 1540 có được thiết lập hay không và xem liệu địa chỉ tuyến tính của mã thông báo 1540 có khớp với địa chỉ của SSP 1530 hay không. Nếu tất cả các kiểm tra ở trên dẫn đến chỉ báo chính xác, thì thiết bị xử lý đánh dấu mã thông báo 1540 là “rảnh” bằng cách xóa bộ chỉ báo bận trong mã thông báo 1540.

Mặc dù các kiến trúc nhất định hỗ trợ hoạt động ở mức quyền ưu tiên 1 hoặc 2 (CPL

== 1 hoặc CPL == 2), nhưng các vòng ưu tiên này không được sử dụng bởi một số hệ điều hành. Hơn nữa, các kiến trúc nhất định không hỗ trợ phân phối ngắt/ngoại lệ với trình xử lý hoạt động ở các mức quyền ưu tiên lớn hơn 0. Trong khi phần còn lại của phần mô tả chi tiết này tập trung vào các tương tác trong vòng quyền ưu tiên 0, các nguyên tắc cơ bản của sáng chế không giới hạn ở hoạt động ở mức quyền ưu tiên cụ thể bất kỳ.

Một phương án của thiết bị xử lý hỗ trợ triển khai trả về nhanh và phân phối sự kiện (Fast Return and Event Delivery FRED) nhằm mục tiêu tăng tốc đáng kể kiến trúc phân phối ngắt/ngoại lệ cho các hệ điều hành 64 bit hiện đại. Cụ thể là, phương án này thay thế việc phân phối thông qua bảng mô tả ngắt (Interrupt Descriptor Table, IDT), do đó loại bỏ các truy nhập bộ nhớ và tải bảng mô tả cần thiết để định vị mã và bộ chọn phân đoạn ngăn xếp và con trỏ cho trình xử lý sự kiện. Ngoài ra, việc ngừng phân phối sự kiện/ngoại lệ cho các vòng 1 và 2 làm giảm độ trễ vì vi mã và/hoặc hệ mạch phân phối sự kiện không cần xem xét các đường dẫn này. Cuối cùng, một phương án đề xuất các lệnh tách biệt để trở lại không gian người dùng và không gian giám sát, để triển khai vi mã tối ưu có độ trễ thấp.

Theo một phương án, khi nhiều ngăn xếp bóng được sử dụng, chỉ một ngăn xếp bóng (ngăn xếp bóng hiện tại) khả dụng ở thời điểm đã cho. Như được mô tả ở trên, con trỏ đến đỉnh của ngăn xếp bóng hiện tại được giữ trong con trỏ ngăn xếp bóng (Shadow Stack Pointer, SSP) 1320. Để hỗ trợ chuyển đổi các ngăn xếp bóng trên các chuyển đổi quyền ưu tiên, một phương án của bộ xử lý bao gồm các thanh ghi mô hình cụ thể sau (Model Specific Register, MSR):

1. IA32_PL3_SSP
2. IA32_FRED_SSP0
3. IA32_FRED_SSP1
4. IA32_FRED_SSP2
5. IA32_FRED_SSP3

Theo một phương án, IA32_PL3_SSP MSR giữ giá trị SSP được thiết lập khi chuyển sang CPL 3. IA32_FRED_SSPi giữ giá trị SSP được thiết lập khi chuyển sang CPL 0 để

phân phối ngắt và/hoặc sự kiện.

FIG.16 minh họa phương án trong đó hệ mạch/mạch logic xử lý sự kiện 1690 chọn SSP MSR 1600-1604 với con trỏ đến ngăn xếp bóng 1640-1644, một cách tương ứng, dựa vào mức ưu tiên của ngắt/ngoại lệ 1694 và mức quyền ưu tiên hiện tại 1696. Theo phương án này, các ngắt và các ngoại lệ 1694 được phân loại thành một trong số bốn mức ưu tiên, 0 đến 3, trong đó 0 là mức ưu tiên thấp nhất và 3 là cao nhất. Khi sự kiện cần được xử lý, thì hệ mạch/mạch logic xử lý sự kiện 1690 phân phối ở mức ngăn xếp được tạo cấu hình cho mức ('i') bằng cách chuyển SSP sang giá trị ở IA32_FRED_SSPi MSR tương ứng. Ví dụ, để đáp lại việc ngắt hoặc ngoại lệ 1694 có mức ưu tiên là 2, con trỏ ngăn xếp bóng từ FRED SSP2 MSR 1602 được chọn, nhận dạng ngăn xếp bóng giám sát SS2 1642 (phụ thuộc vào mức quyền ưu tiên hiện tại 1696 như được mô tả dưới đây). Tuy nhiên, nếu bộ xử lý đã xử lý sự kiện được phân loại cần được phân phối ở mức ngăn xếp cao hơn mức ngăn xếp hoạt động hiện tại, sau đó bộ xử lý vẫn ở trên ngăn xếp hiện tại.

Phần còn lại của phần mô tả này sẽ giả sử chuyển đổi sang ngăn xếp bóng cho mức ngăn xếp 'i' mà sự kiện/ngắt cần được phân phối. Theo một phương án, mỗi trong số các FRED SSPi MSR 1600-1603 lưu trữ các định nghĩa trường bit sau:

- Bit 0 – Được xác minh (V) – chỉ báo xem liệu MSR có chỉ đến mã thông báo ngăn xếp bóng giám sát được xác minh (được mô tả sau)
- Bit 2:1 – Dành riêng (thiết lập bằng 0)
- Bit 63:3 – Địa chỉ tuyến tính của mã thông báo ngăn xếp bóng giám sát hợp lệ trên ngăn xếp bóng giám sát.

Theo một phương án, hệ mạch/mạch logic xử lý sự kiện 1690 chọn con trỏ từ một trong số các SSP MSR 1600-1603 nhận dạng ngăn xếp bóng giám sát mới 1640-1643, một cách tương ứng, nếu sự kiện 1694 đến khi bộ xử lý đang thực thi ở chế độ người dùng ở CPL3 (tức là, như được chỉ báo bởi mức quyền ưu tiên hiện tại 1694) hoặc nếu bộ xử lý đang thực thi ở chế độ giám sát ở CPL0 và sự kiện 1694 cần được phân phối ở mức ngăn xếp cao hơn. Do đó, giả sử các ngăn xếp bóng được cho phép ở CPL3 (chế độ người dùng)

và CLP0 (chế độ giám sát), SSP được lưu vào PL3 SSP 1604 (tức là, khi bắt đầu ở chế độ người dùng) và ngăn xếp bóng được thiết lập cho chế độ giám sát sử dụng địa chỉ SSP trong FRED SSPi MSR 1600-1603 được chọn trong đó i là mức ngăn xếp đang được chuyển đổi như được nhận dạng bởi mức ưu tiên của sự kiện 1694 được phân phối.

Khi SSP thăm dò cho chế độ giám sát được chọn làm PL0 SSP 1650 từ một trong số các FRED SSPi MSR 1600-1603, thì hệ mạch/mạch logic kiểm tra xác minh & sử dụng 1695 xác nhận rằng (1) SSP được lập trình trong SSP MSR được chọn 1600-1603 đang chỉ đến đỉnh của ngăn xếp bóng giám sát tương ứng 1640-1643, một cách tương ứng, và (2) ngăn xếp bóng giám sát 1640-1643 được chỉ đến bởi SSP thăm dò này 1602 không hoạt động trên bộ xử lý logic khác bất kỳ, như được chỉ báo bởi bit bận (B). Nếu vượt qua cả hai lần kiểm tra sau đó FRED SSPi MSR 1602 trở thành mức quyền ưu tiên mới 0 (PL0) SSP 1650.

Các kiểm tra này được thực hiện để ngăn chặn cuộc tấn công luồng chéo trong đó kẻ tấn công có thể trở MSR trên hai bộ xử lý logic đến cùng ngăn xếp bóng và do đó có thể thao tác các địa chỉ trả về được sử dụng trên một bộ xử lý logic bằng cách thực hiện các lệnh gọi trên bộ xử lý logic thứ hai.

Theo một phương án, để thực hiện các kiểm tra này, phần mềm hệ thống đặt mã thông báo ngăn xếp bóng giám sát 1650-1653 ở đỉnh của mỗi ngăn xếp bóng giám sát. Theo một phương án thực hiện cụ thể, mã thông báo ngăn xếp bóng giám sát 1650-1653 được định dạng như sau:

- Các Bit 63:3 – Địa chỉ tuyến tính của mã thông báo trên ngăn xếp bóng
- Bit 2:1 – Dành riêng (thiết lập bằng 0)
- Bit 0 – Bận (B); nếu bit này là 0, thì ngăn xếp bóng được chọn 1640-1643 không được tải trên bộ xử lý logic bất kỳ.

FIG.17 minh họa ngăn xếp bóng giám sát 1642 với mã thông báo ngăn xếp bóng giám sát 1652. PL0_SSP 1650 giữ địa chỉ tuyến tính 0x1008 chỉ đến đỉnh của ngăn xếp bóng giám sát 1642 để giữ mã thông báo ngăn xếp bóng giám sát 1652. Khi phần mềm hệ

thông tạo ra ngăn xếp bóng này thì cờ bận 1701 – bit 0 – được khởi tạo từ 0.

Phương pháp theo một phương án của sáng chế được minh họa trên FIG.18. Phương pháp có thể được thực hiện trên bộ xử lý và các kiến trúc hệ thống được mô tả trong bản mô tả này, nhưng không giới hạn ở kiến trúc cụ thể.

Ở bước 1801, sự kiện hệ thống (ví dụ, ngắt) được nhận ở mức quyền ưu tiên hiện tại có mức ưu tiên là i . Ở bước 1802 việc xác định được thực hiện để xem liệu sự thay đổi có được yêu cầu đối với con trỏ ngăn xếp bóng mức quyền ưu tiên 0 hay không. Như được mô tả ở trên, sự thay đổi được yêu cầu nếu sự kiện đến khi bộ xử lý thực thi ở chế độ người dùng ở CPL3 hoặc nếu bộ xử lý đang thực thi ở chế độ giám sát ở CPL0 và sự kiện này cần được phân phối ở mức ngăn xếp cao hơn. Nếu không yêu cầu sự thay đổi nào, sau đó quy trình giữ nguyên ở mức quyền ưu tiên hiện tại với SSP hiện tại ở bước 1810.

Nếu sự thay đổi được yêu cầu, sau đó ở bước 1803, SSP thăm dò được chọn dựa vào mức ưu tiên (i) của sự kiện. Ví dụ, SSP thăm dò có thể được chọn từ SSP MSR_i . Ở bước 1804 các kiểm tra xác minh và bận được thực hiện đối với SSP thăm dò. Ví dụ, mã thông báo ở đỉnh của ngăn xếp bóng có thể được so sánh với SSP thăm dò và bit bận có thể được kiểm tra để xác nhận rằng ngăn xếp bóng được nhận dạng bởi SSP thăm dò không được tải trên bộ xử lý logic khác bất kỳ. Nếu một trong số các kiểm tra này thất bại, thì được xác định ở bước 1805, sau đó lỗi/ngoại lệ được tạo ra ở bước 1811.

Nếu vượt qua cả hai kiểm tra ở bước 1805, sau đó ở bước 1806 con trỏ ngăn xếp bóng cho mức quyền ưu tiên 0 được thiết lập bằng SSP thăm dò. Mức quyền ưu tiên hiện tại cũng có thể thay đổi. Ở bước 1807 sự kiện được xử lý ở mức quyền ưu tiên hiện tại và sử dụng giá trị SSP PL0 mới. Quy trình này sau đó trở lại bước 1801 khi sự kiện hệ thống tiếp theo được nhận.

Theo một phương án, hệ mạch/mạch logic kiểm tra xác nhận & sử dụng 1690 và các kiểm tra xác nhận & bận 1804 hoạt động để đảm bảo rằng nếu MSR được đánh dấu là chưa được xác minh sau đó ngăn xếp bóng (ví dụ, FRED SSP2 1602) có thể được tải chỉ trên một bộ xử lý logic ở thời điểm bất kỳ (“Kiểm tra 1” trong chuỗi mã ở trên). Ngoài ra, các phương án này đảm bảo rằng con trỏ ngăn xếp bóng trong MSR chỉ đến mã thông báo ngăn

xếp bóng giám sát hợp lệ ở đỉnh của ngăn xếp (“Kiểm tra 2” trong chuỗi mã ở trên).

Theo một phương án, chuỗi mã sau được thực thi để xác minh mã thông báo và thiết lập trở ngăn xếp bóng SSP thành địa chỉ trong mã thông báo:

```
IF ShadowStackEnabled(CPL)
```

```
tempSSP = FRED_SSPI; (* trong đó i là mức ngăn xếp *)
```

```
tempSSP.B = 0
```

```
If FRED_SSPI.V == 0
```

```
    // Bit được xác minh chưa được thiết lập
```

```
    Fault = 0
```

```
    Atomic Start
```

```
    SSPToken = 8 bytes loaded with shadow stack semantics from tempSSP
```

```
        // Kiểm tra 1: NẾU BIT BẠN được thiết lập sau đó gây ra lỗi
```

```
        IF (SSPToken.B == 1) THEN fault <-1; FI;
```

// Kiểm tra 2: Nếu địa chỉ trong mã thông báo không trùng với SSP thăm dò sau đó lỗi

```
        IF SSPToken != tempSSP) THEN fault <-1; FI;
```

```
        // Nếu không có lỗi được phát hiện sau đó thiết lập mã thông báo bạn
```

```
        IF fault = 0 THEN SSPToken.B = 1; FI;
```

```
Lưu trữ 8 byte của SSPToken bằng các ngữ nghĩa ngăn xếp bóng vào SSP;
```

```
Atomic End
```

```
If fault = 1 THEN #GP(0); FI;
```

```
    // Nếu lỗi bất kỳ được phát hiện sau đó gây ra ngoại lệ #GP
```

```

    FRED_SSPI.V = 1 // Thiết lập bit được xác minh trong MSR

ELSE

    // MSR có tập hợp bit được xác minh; kiểm tra nếu địa chỉ MSR trùng với mã
    thông báo

    SSPToken = 8 bytes loaded with shadow stack semantics from tempSSP

    // Kiểm tra 1: NẾU BIT BẬN được thiết lập sau đó gây ra lỗi

    IF (SSPToken.B == 1) THEN fault <-1; FI;

    // Kiểm tra 2: Nếu địa chỉ trong mã thông báo không trùng với SSP thăm dò sau đó
    lỗi

    IF SSPToken != tempSSP THEN fault <-1; FI;

    If fault = 1 THEN #GP(0); FI; // Nếu lỗi bất kỳ được phát hiện sau đó gây ra ngoại
    lệ #GP

    ENDIF

    SSP = tempSSP

FI

```

Lưu ý rằng khi MSR không được đánh dấu là được xác minh thì các kiểm tra này có thể được thực hiện nguyên tử bằng cách sử dụng đọc-sửa-đổi-ghi được khóa như được chỉ báo bởi Nguyên tử-Bắt đầu trong chuỗi mã ở trên lấy khóa trên dòng bộ đệm và Nguyên tử-Kết thúc mà phát hành khóa được lấy. Nếu các kiểm tra này thành công thì mã thông báo được đánh dấu bận và MSR được đánh dấu là được xác minh.

Theo một phương án, khi MSR được đánh dấu là được xác minh thì trong các lần gọi tiếp theo của các hoạt động này, không có hoạt động đọc-sửa-ghi bị khoá nào được thực hiện nữa. Thay vào đó, nếu MSR được đánh dấu là được xác minh, thì hai lần kiểm tra giống nhau có thể được thực hiện bằng cách tải mã thông báo từ bộ nhớ mà không lấy khóa

trên dòng bộ đệm giữ mã thông báo. Theo một phương án, để tránh bit được xác minh được thiết lập sai hoặc có hại trong MSR, các lệnh được sử dụng để ghi MSR này luôn bỏ qua bit được xác minh (ví dụ, bit 0) trong giá trị được ghi và xóa vô điều kiện bit được xác minh thành 0. Các lệnh này bao gồm, nhưng không bị giới hạn ở, lệnh WRMSR (ghi MSR kiểu riêng) và lệnh XRSTORS (khôi phục các trạng thái mở rộng chế độ giám sát bộ xử lý từ bộ nhớ).

FIG.19 minh họa một phương án của phương pháp để lựa chọn giữa các hoạt động nguyên tử và các hoạt động phi nguyên tử dựa vào bit được xác minh. Ở bước 1901 sự kiện hệ thống chẳng hạn như sự ngắt được nhận có mức ưu tiên là i . Ở bước 1902, việc kiểm tra được thực hiện để xác định xem liệu bit được xác minh (ví dụ, $SSPi = 0$) có bằng 0 hay không (tức là, chưa được xác minh). Nếu bit được xác minh bằng 0, thì ở bước 1903 tập hợp hoạt động nguyên tử được khởi tạo. Bit bận được kiểm tra ở bước 1904. Nếu nó được thiết lập bằng 0, thì sau đó ở bước 1905 mã thông báo ngăn xếp bóng được so sánh với SSP tạm thời. Nếu chúng trùng khớp, sau đó các hoạt động nguyên tử được hoàn thành ở bước 1906. Ở bước 1907 bit được xác minh được thiết lập bằng 1 và ở bước 1908 con trỏ ngăn xếp bóng tạm thời (Temporary Shadow Stack Pointer, $SSPi$) được đặt là con trỏ ngăn xếp bóng (Shadow Stack Pointer, SSP) hiện tại.

Trở lại bước 1904, nếu bit bận được thiết lập bằng 1 (nghĩa là ngăn xếp bóng sử dụng bộ xử lý logic khác) hoặc, ở bước 1905, nếu mã thông báo không trùng với SSP tạm thời, sau đó điều kiện lỗi được tạo ra ở bước 1913. Ngoài ra, ở bước 1902, nếu bit được xác minh được thiết lập bằng 1 (nghĩa là SSP đã được xác minh), sau đó tập hợp các hoạt động nguyên tử không cần thiết nữa. Hai lần kiểm tra giống nhau có thể được thực hiện ở bước 1914 (kiểm tra bit bận) và ở bước 1915 (so sánh mã thông báo). Tuy nhiên, các hoạt động này được thực hiện mà không lấy khóa trên dòng bộ đệm giữ dữ liệu cần thiết. Ví dụ, không có khóa nào được lấy trên dòng bộ đệm lưu trữ mã thông báo, do đó cải thiện hiệu suất bằng cách cho phép các hoạt động khác tiến hành đồng thời.

Bộ xử lý có thể chuyển đổi khỏi ngăn xếp bóng giám sát (ví dụ, $PL0\ SSP\ 1650$) khi trở lại xử lý ngắt/ngoại lệ nếu trở về chế độ người dùng (CPL3) hoặc khi trở lại quy trình bị gián đoạn ở chế độ giám sát nhưng quy trình bị ngắt đang thực thi ở mức ngăn xếp thấp

hơn mức ngăn xếp mà sự kiện được xử lý.

Hệ điều hành như một phần của việc xử lý sự kiện/ngắt cũng có thể thực hiện chuyển đổi ngữ cảnh. Theo một phương án, chuyển đổi ngữ cảnh gây ra việc ghi vào FRED_SS*Pi* MSR để cập nhật con trỏ ngăn xếp bóng được liên kết với một hoặc nhiều mức ưu tiên. Trong trường hợp này, việc ghi vào các FRED_SS*Pi* MSR làm cho bit được xác minh trong MSR bị xóa thành 0.

Do đó, theo một phương án, ba trường hợp này có thể xảy ra khi trả về từ trình xử lý sự kiện:

(1) $\text{FRED_SSPi} = \text{SSP} \mid 1$:

- SSP trùng với giá trị MSR (SSP ở đáy ngăn xếp) và MSR được xác minh
- Không làm gì

(2) $\text{FRED_SSPi} = \text{SSP}$

- SSP trùng với giá trị MSR nhưng MSR không được xác minh
- Kiểm tra mã thông báo để thấy nếu MSR có thể được xác minh lại:

(3) $\text{FRED_SSPi}[63:1] \neq \text{SSP}[63:1]$

– SSP không trùng MSR; khởi động lại mã thông báo để làm cho sự xác minh MSR tự do và rõ ràng

- Hoạt động này yêu cầu đọc-sửa-ghi nguyên tử như được mô tả ở trên.

IF SSP == FRED_SS*Pi* // trong đó i là mức ngăn xếp hiện tại

// SSP trùng với giá trị MSR nhưng bit được xác minh trong MSR bằng 0

// Cố gắng xác minh lại MSR – sự xác minh lại này không yêu cầu khóa bất kỳ

// Kiểm tra 1: Trước tiên kiểm tra nếu có bit bận trong mã thông báo

IF SSPToken.B = 0 (* Nếu bit bận không được thiết lập sau mã thông báo không

hợp lệ*)

```
invalidToken <-1; FI;
```

```
// Kiểm tra 2: Xác minh địa chỉ trong mã thông báo trùng với SSP
```

```
IF ((SSPToken AND ~0x1) !=SSP)
```

```
invalidToken <-1; FI;
```

```
// Tìm thấy mã thông báo hợp lệ; xóa bit bận của chúng
```

```
IF invalidToken = 0
```

```
FRED_SSPI.V = 1
```

```
ELSE
```

```
IF SSP != FRED_SSPI & ~0x01
```

```
Atomic Start
```

```
SSPToken <- Load 8 bytes with shadow stack semantics from SSP
```

```
nvalidToken <- 0
```

```
// Kiểm tra 1: Đầu tiên kiểm tra nếu có bit bận trong mã thông báo
```

```
IF SSPToken.B = 0 (* If busy bit not set then invalid token*)
```

```
invalidToken <- 1; FI;
```

```
// Kiểm tra 2: Xác minh địa chỉ trong mã thông báo trùng với SSP
```

```
IF ((SSPToken AND ~0x7) !=SSP)
```

```
invalidToken <- 1; FI;
```

```
// Tìm thấy mã thông báo hợp lệ; xóa bit bận của chúng
```

```
IF invalidToken = 0
```

THEN SSPToken.B = 1;

Lưu trữ 8 byte của SSPToken bằng các ngữ nghĩa ngăn xếp bóng vào SSP;

Atomic End

ENDIF

ENDIF

Do đó, khi MSR đang chỉ đến mã thông báo ngăn xếp bóng giám sát hợp lệ và bận, thì MSR được đánh dấu là được xác minh. Hoạt động này là trường hợp thường xuyên nhất và không yêu cầu các hoạt động được khóa – nhờ đó cải thiện hiệu suất. Nếu MSR đang không chỉ đến mã thông báo ngăn xếp bóng giám sát hợp lệ và bận trên ngăn xếp bóng hiện tại, và nếu có mã thông báo ngăn xếp bóng giám sát bận trên ngăn xếp bóng hiện tại thì mã thông báo này được tạo tự do. Hoạt động này được thực hiện bằng cách sử dụng hoạt động được khóa để đảm bảo tính nhất quán và bảo mật.

Sau khi xóa mã thông báo này, nếu bộ xử lý đang trở lại chế độ người dùng (CPL3), thì bộ xử lý thiết lập con trỏ ngăn xếp bóng cho chế độ người dùng từ PL3_SSP 1604. Nếu bộ xử lý đang quay trở lại quy trình bị ngắt ở chế độ giám sát (CPL0) thì con trỏ ngăn xếp bóng của quy trình bị gián đoạn được khôi phục (ví dụ, sử dụng con trỏ FRED SSP0).

Lưu ý rằng lệnh WRMSR hoặc lệnh XRSTORS ghi vào FRED_SSPi MSR làm cho bit được xác minh trong MSR bị xóa thành 0. Tuy nhiên trình quản lý máy ảo (Virtual Machine Manager, VMM) hoặc siêu giám sát cần phải chuyển đổi ngữ cảnh trạng thái của các MSR này trên bộ chuyển đổi máy ảo (Virtual Machine, VM). Theo phương án này, chuyển đổi ngữ cảnh VM không được làm mất trạng thái được xác minh trong các MSR này.

Theo một phương án, để giải quyết trường hợp này khi xảy ra lỗi thoát VM, bộ xử lý lưu các FRED_SSPi MSR - bao gồm bit được xác minh - vào cấu trúc điều khiển máy ảo (Virtual Machine Control Structure, VMCS). Một mục nhập VM tiếp theo, bộ xử lý khôi phục trạng thái của các MSR này, bao gồm bit được xác minh, từ VMCS.

Các ví dụ

Phần tiếp theo là các phương án thực hiện ví dụ theo các phương án khác nhau của sáng chế.

Ví dụ 1. Bộ xử lý bao gồm: các thanh ghi để lưu trữ các con trỏ ngăn xếp bóng (Shadow Stack Pointer, SSP), mỗi SSP được liên kết với mức ưu tiên sự kiện khác nhau; hệ mạch xử lý sự kiện để chọn SSP thứ nhất trong số các SSP từ thanh ghi thứ nhất trong số các thanh ghi đáp ứng lại việc tiếp nhận sự kiện thứ nhất được liên kết với mức ưu tiên sự kiện thứ nhất, SSP thứ nhất sử dụng được để nhận dạng đỉnh của ngăn xếp bóng thứ nhất; hệ mạch kiểm tra xác minh và sử dụng để xác định xem liệu SSP thứ nhất có được xác minh trước đó hay không, trong đó nếu SSP thứ nhất chưa được xác minh trước đó thì khởi tạo tập hợp các hoạt động nguyên tử để xác minh SSP thứ nhất và xác nhận rằng SSP thứ nhất đang không được sử dụng, tập hợp các hoạt động nguyên tử sử dụng hoạt động khóa để khóa dữ liệu cho đến khi hoàn thành tập hợp các hoạt động nguyên tử, và trong đó nếu SSP thứ nhất được xác minh trước đó, thì xác minh lại SSP thứ nhất và xác nhận rằng SSP thứ nhất đang không được sử dụng mà không sử dụng hoạt động khóa.

Ví dụ 2. Bộ xử lý theo ví dụ 1 trong đó các SSP bao gồm các SSP giám sát, bao gồm SSP thứ nhất, trỏ đến các ngăn xếp bóng giám sát tương ứng được liên kết với các mức quyền ưu tiên giám sát.

Ví dụ 3. Bộ xử lý theo ví dụ 2 trong đó các SSP còn bao gồm ít nhất một SSP mức người dùng chỉ đến ngăn xếp bóng mức người dùng được liên kết với mức quyền ưu tiên người dùng.

Ví dụ 4. Bộ xử lý theo điểm 1 trong đó sự kiện thứ nhất bao gồm ngắt thứ nhất hoặc ngoại lệ thứ nhất được liên kết với mức ưu tiên thứ nhất.

Ví dụ 5. Bộ xử lý theo ví dụ 4 trong đó các SSP được liên kết với các mức ưu tiên sự kiện tương ứng.

Ví dụ 6. Bộ xử lý theo ví dụ 1 trong đó xác nhận rằng SSP thứ nhất đang không được sử dụng bao gồm kiểm tra giá trị bit bận được liên kết với SSP thứ nhất.

Ví dụ 7. Bộ xử lý theo ví dụ 6 trong đó bước xác minh và xác minh lại SSP thứ nhất bao gồm bước so sánh SSP thứ nhất với giá trị mã thông báo thứ nhất được lưu trữ ở đỉnh của ngăn xếp bóng thứ nhất.

Ví dụ 8. Bộ xử lý theo ví dụ 7 còn bao gồm: hệ mạch thực thi để thực thi một hoặc nhiều lệnh sử dụng ngăn xếp dữ liệu thứ nhất được liên kết với ngăn xếp bóng thứ nhất ở mức quyền ưu tiên giám sát thứ nhất được liên kết với ngăn xếp dữ liệu thứ nhất.

Ví dụ 9. Bộ xử lý theo ví dụ 8 trong đó hệ mạch xử lý sự kiện và hệ mạch kiểm tra xác minh và sử dụng bao gồm các phần của đường truyền bằng ống thực thi lệnh của bộ xử lý, hệ mạch kiểm tra xác minh và sử dụng bao gồm phần thứ nhất của đường truyền bằng ống thực thi lệnh để thực thi tập hợp lệnh thứ nhất liên quan đến việc kiểm tra xác minh và sử dụng SSP và hệ mạch xử lý sự kiện bao gồm phần thứ hai của đường truyền bằng ống thực thi lệnh để thực thi tập hợp lệnh thứ hai liên quan đến việc xử lý sự kiện.

Ví dụ 10. Phương pháp bao gồm các bước: lưu trữ các con trỏ ngăn xếp bóng (Shadow Stack Pointer, SSP) trong các thanh ghi tương ứng, mỗi SSP được liên kết với mức ưu tiên sự kiện khác nhau; chọn SSP thứ nhất trong số các SSP từ thanh ghi thứ nhất trong số các thanh ghi đáp ứng lại việc tiếp nhận sự kiện thứ nhất được liên kết với mức ưu tiên sự kiện thứ nhất, SSP thứ nhất sử dụng được để nhận dạng đỉnh của ngăn xếp bóng thứ nhất; xác định xem liệu SSP thứ nhất đã được xác minh trước đó hay chưa, trong đó nếu SSP thứ nhất chưa được xác minh trước đó thì khởi tạo tập hợp các hoạt động nguyên tử để xác minh SSP thứ nhất và xác nhận rằng SSP thứ nhất đang không được sử dụng, tập hợp các hoạt động nguyên tử sử dụng hoạt động khóa để khóa dữ liệu cho đến khi hoàn thành tập hợp các hoạt động nguyên tử, và trong đó nếu SSP thứ nhất được xác minh trước đó, thì xác minh lại SSP thứ nhất và xác nhận rằng SSP thứ nhất đang không được sử dụng mà không sử dụng hoạt động khóa.

Ví dụ 11. Phương pháp theo ví dụ 10 trong đó các SSP bao gồm các SSP giám sát, bao gồm SSP thứ nhất, trỏ đến các ngăn xếp bóng giám sát tương ứng được liên kết với các mức quyền ưu tiên giám sát.

Ví dụ 12. Phương pháp theo ví dụ 11 trong đó các SSP còn bao gồm ít nhất một SSP

mức người dùng chỉ đến ngăn xếp bóng mức người dùng được liên kết với mức quyền ưu tiên người dùng.

Ví dụ 13. Phương pháp theo điểm 10 trong đó sự kiện thứ nhất bao gồm ngắt thứ nhất hoặc ngoại lệ thứ nhất được liên kết với mức ưu tiên thứ nhất.

Ví dụ 14. Phương pháp theo ví dụ 13 trong đó các SSP được liên kết với các mức ưu tiên sự kiện tương ứng.

Ví dụ 15. Phương pháp theo điểm 10 trong đó bước xác nhận rằng SSP thứ nhất đang không được sử dụng bao gồm bước kiểm tra giá trị bit bận được liên kết với SSP thứ nhất.

Ví dụ 16. Phương pháp theo ví dụ 15 trong đó bước xác minh và xác minh lại SSP thứ nhất bao gồm bước so sánh SSP thứ nhất với giá trị mã thông báo thứ nhất được lưu trữ ở đỉnh của ngăn xếp bóng thứ nhất.

Ví dụ 17. Phương pháp theo ví dụ 16 còn bao gồm các bước: thực thi một hoặc nhiều lệnh sử dụng ngăn xếp dữ liệu thứ nhất được liên kết với ngăn xếp bóng thứ nhất ở mức quyền ưu tiên giám sát thứ nhất được liên kết với ngăn xếp dữ liệu thứ nhất.

Ví dụ 18. Vật ghi đọc được bằng máy có mã chương trình được lưu trữ trên đó mà, khi được thực thi bởi máy, làm cho máy này thực hiện các hoạt động trong số: lưu trữ các con trỏ ngăn xếp bóng (Shadow Stack Pointer, SSP) trong các thanh ghi tương ứng, mỗi SSP được liên kết với mức ưu tiên sự kiện khác nhau; chọn SSP thứ nhất trong số các SSP từ thanh ghi thứ nhất trong số các thanh ghi đáp ứng lại việc tiếp nhận sự kiện thứ nhất được liên kết với mức ưu tiên sự kiện thứ nhất, SSP thứ nhất sử dụng được để nhận dạng đỉnh của ngăn xếp bóng thứ nhất; xác định xem liệu SSP thứ nhất đã được xác minh trước đó hay chưa, trong đó nếu SSP thứ nhất chưa được xác minh trước đó thì khởi tạo tập hợp các hoạt động nguyên tử để xác minh SSP thứ nhất và xác nhận rằng SSP thứ nhất đang không được sử dụng, tập hợp các hoạt động nguyên tử sử dụng hoạt động khóa để khóa dữ liệu cho đến khi hoàn thành tập hợp các hoạt động nguyên tử, và trong đó nếu SSP thứ nhất được xác minh trước đó, thì xác minh lại SSP thứ nhất và xác nhận rằng SSP thứ nhất đang

không được sử dụng mà không sử dụng hoạt động khóa.

Ví dụ 19. Vật ghi đọc được bằng máy theo ví dụ 18 trong đó các SSP bao gồm các SSP giám sát, bao gồm SSP thứ nhất, trở đến các ngăn xếp bóng giám sát tương ứng được liên kết với các mức quyền ưu tiên giám sát.

Ví dụ 20. Vật ghi đọc được bằng máy theo ví dụ 19 trong đó các SSP còn bao gồm ít nhất một SSP mức người dùng chỉ đến ngăn xếp bóng mức người dùng được liên kết với mức quyền ưu tiên người dùng.

Ví dụ 21. Vật ghi đọc được bằng máy theo ví dụ 18 trong đó sự kiện thứ nhất bao gồm ngắt thứ nhất hoặc ngoại lệ thứ nhất được liên kết với mức ưu tiên thứ nhất.

Ví dụ 22. Vật ghi đọc được bằng máy theo ví dụ 21 trong đó các SSP được liên kết với các mức ưu tiên sự kiện tương ứng.

Ví dụ 23. Vật ghi đọc được bằng máy theo ví dụ 18 trong đó xác nhận rằng SSP thứ nhất đang không được sử dụng bao gồm kiểm tra giá trị bit bận được liên kết với SSP thứ nhất.

Ví dụ 24. Vật ghi đọc được bằng máy theo ví dụ 23 trong đó bước xác minh và xác minh lại SSP thứ nhất bao gồm bước so sánh SSP thứ nhất với giá trị mã thông báo thứ nhất được lưu trữ ở đỉnh của ngăn xếp bóng thứ nhất.

Ví dụ 25. Vật ghi đọc được bằng máy theo ví dụ 24 còn bao gồm mã chương trình làm cho máy này thực hiện các hoạt động trong số: thực thi một hoặc nhiều lệnh sử dụng ngăn xếp dữ liệu thứ nhất được liên kết với ngăn xếp bóng thứ nhất ở mức quyền ưu tiên giám sát thứ nhất được liên kết với ngăn xếp dữ liệu thứ nhất.

Trong phần mô tả nêu trên, các phương án của sáng chế được mô tả với tham chiếu đến các phương án làm ví dụ cụ thể của sáng chế. Tuy nhiên, hiển nhiên là các cải biến và thay đổi khác nhau có thể được thực hiện mà không vượt ra khỏi bản chất và phạm vi rộng hơn của sáng chế như được thể hiện trong yêu cầu bảo hộ kèm theo. Theo đó, bản mô tả này và hình vẽ chỉ mang tính chất minh họa thay vì mang ý nghĩa giới hạn.

Các phương án của sáng chế có thể bao gồm các bước khác nhau, được mô tả ở trên. Các bước này có thể được thể hiện ở các lệnh thực thi được bằng máy có thể được sử dụng để làm cho bộ xử lý đa năng hoặc chuyên dụng thực hiện các bước này. Theo cách khác, các bước này có thể được thực hiện bởi các thành phần phần cứng cụ thể mà chứa các mạch logic được nối cứng để thực hiện các bước này, hoặc bởi sự kết hợp bất kỳ của các thành phần máy tính được lập trình và các thành phần phần cứng tùy chỉnh.

Như được mô tả trong bản mô tả này, các lệnh có thể đề cập đến các cấu hình phần cứng cụ thể chẳng hạn như các mạch tích hợp chuyên dụng (Application Specific Integrated Circuit, ASIC) được tạo cấu hình để thực hiện các hoạt động nhất định hoặc có chức năng hoặc các lệnh phần mềm định trước được lưu trữ trong bộ nhớ được thể hiện ở dạng vật ghi lâu dài đọc được bằng máy tính. Do đó, các kỹ thuật được thể hiện trên các hình vẽ có thể được thực hiện bằng cách sử dụng mã và dữ liệu được lưu trữ và được thực thi trên một hoặc nhiều thiết bị điện tử (ví dụ, trạm cuối, phần tử mạng, v.v.). Các thiết bị điện tử này lưu trữ và truyền (nội bộ và/hoặc với các thiết bị điện tử khác qua mạng) mã và dữ liệu bằng cách sử dụng vật ghi đọc được bằng máy tính, chẳng hạn như vật ghi lưu trữ lâu dài đọc được bằng máy tính (ví dụ, các đĩa từ; các đĩa quang; bộ nhớ truy nhập ngẫu nhiên; bộ nhớ chỉ đọc; các thiết bị nhớ nhanh; bộ nhớ chuyển pha) và vật ghi truyền thông đọc được bằng máy tính tạm thời (ví dụ, các tín hiệu điện, quang, âm hoặc các dạng tín hiệu lan truyền khác – chẳng hạn như các sóng mang, các tín hiệu hồng ngoại, các tín hiệu số, v.v.). Ngoài ra, các thiết bị điện tử này thường bao gồm tập hợp trong số một hoặc nhiều bộ xử lý được ghép nối với một hoặc nhiều thành phần khác, chẳng hạn như một hoặc nhiều thiết bị lưu trữ (vật ghi lưu trữ lâu dài đọc được bằng máy), các thiết bị đầu vào/đầu ra người dùng (ví dụ, bàn phím, màn hình, và/hoặc màn hiển thị), và các kết nối mạng. Việc ghép nối tập hợp bộ xử lý và các thành phần khác thường qua một hoặc nhiều bus và cầu nối (cũng được gọi là các bộ điều khiển bus). Thiết bị lưu trữ và các tín hiệu mang lưu lượng mạng lần lượt biểu diễn một hoặc nhiều vật ghi lưu trữ đọc được bằng máy và vật ghi truyền thông đọc được bằng máy. Do đó, thiết bị lưu trữ của thiết bị điện tử đã cho thường lưu trữ mã và/hoặc dữ liệu để thực thi trên tập hợp một hoặc nhiều bộ xử lý của thiết bị điện tử. Tất nhiên, một hoặc nhiều phần của một phương án của sáng chế có thể được thực hiện bằng cách sử dụng các sự kết hợp khác nhau của phần mềm, phần sụn, và/hoặc phần cứng. Trong toàn bộ phần

mô tả chi tiết này, với các mục đích giải thích, nhiều nội dung chi tiết cụ thể được nêu để hiểu trọn vẹn sáng chế. Tuy nhiên, hiển nhiên với người có hiểu biết trung bình trong lĩnh vực kỹ thuật này rằng sáng chế có thể được thực hiện mà không cần một số trong số các nội dung chi tiết cụ thể này. Trong các trường hợp nhất định, các cấu trúc và chức năng đã biết không được mô tả chi tiết để tránh làm tối nghĩa đối tượng của sáng chế. Theo đó, phạm vi và bản chất của sáng chế cần được đánh giá dựa vào yêu cầu bảo hộ sau.

YÊU CẦU BẢO HỘ

1. Bộ xử lý bao gồm:

các thanh ghi để lưu trữ các con trỏ ngăn xếp bóng (Shadow Stack Pointer, SSP), mỗi SSP được liên kết với mức ưu tiên sự kiện khác nhau;

hệ mạch xử lý sự kiện để chọn SSP thứ nhất trong số các SSP từ thanh ghi thứ nhất trong số các thanh ghi đáp ứng lại việc tiếp nhận sự kiện thứ nhất được liên kết với mức ưu tiên sự kiện thứ nhất, SSP thứ nhất sử dụng được để nhận dạng đỉnh của ngăn xếp bóng thứ nhất;

hệ mạch kiểm tra xác minh và sử dụng để xác định xem liệu SSP thứ nhất đã được xác minh trước đó hay chưa,

trong đó nếu SSP thứ nhất chưa được xác minh trước đó thì khởi tạo tập hợp các hoạt động nguyên tử để xác minh SSP thứ nhất và xác nhận rằng SSP thứ nhất đang không được sử dụng, tập hợp các hoạt động nguyên tử này sử dụng hoạt động khóa để khóa dữ liệu cho đến khi hoàn thành tập hợp các hoạt động nguyên tử, và

trong đó nếu SSP thứ nhất đã được xác minh trước đó, thì xác minh lại SSP thứ nhất và xác nhận rằng SSP thứ nhất đang không được sử dụng mà không sử dụng hoạt động khóa.

2. Bộ xử lý theo điểm 1 trong đó các SSP bao gồm các SSP giám sát, bao gồm SSP thứ nhất, trỏ đến các ngăn xếp bóng giám sát tương ứng được liên kết với các mức quyền ưu tiên giám sát.

3. Bộ xử lý theo điểm 2 trong đó các SSP còn bao gồm ít nhất một SSP mức người dùng chỉ đến ngăn xếp bóng mức người dùng được liên kết với mức quyền ưu tiên người dùng.

4. Bộ xử lý theo điểm 1 trong đó sự kiện thứ nhất bao gồm ngắt thứ nhất hoặc ngoại lệ thứ nhất được liên kết với mức ưu tiên thứ nhất.

5. Bộ xử lý theo điểm 4 trong đó các SSP được liên kết với các mức ưu tiên sự kiện tương ứng.

6. Bộ xử lý theo điểm 1 trong đó bước xác nhận rằng SSP thứ nhất đang không được sử dụng bao gồm bước kiểm tra giá trị bit bận được liên kết với SSP thứ nhất.

7. Bộ xử lý theo điểm 6 trong đó bước xác minh và xác minh lại SSP thứ nhất bao gồm bước so sánh SSP thứ nhất với giá trị mã thông báo thứ nhất được lưu trữ ở đỉnh của ngăn xếp bóng thứ nhất.

8. Bộ xử lý theo điểm 7 còn bao gồm:

hệ mạch thực thi để thực thi một hoặc nhiều lệnh bằng cách sử dụng ngăn xếp dữ liệu thứ nhất được liên kết với ngăn xếp bóng thứ nhất ở mức quyền ưu tiên giám sát thứ nhất được liên kết với ngăn xếp dữ liệu thứ nhất này.

9. Bộ xử lý theo điểm 8 trong đó hệ mạch xử lý sự kiện và hệ mạch kiểm tra xác minh và sử dụng bao gồm các phần của đường truyền bằng ống thực thi lệnh của bộ xử lý, hệ mạch kiểm tra xác minh và sử dụng bao gồm phần thứ nhất của đường truyền bằng ống thực thi lệnh để thực thi tập hợp lệnh thứ nhất được liên kết với việc kiểm tra xác minh và sử dụng SSP và hệ mạch xử lý sự kiện bao gồm phần thứ hai của đường truyền bằng ống thực thi lệnh để thực thi tập hợp lệnh thứ hai được liên kết với việc xử lý sự kiện.

10. Phương pháp quản lý và xử lý các ngăn xếp bóng bao gồm các bước:

lưu các con trỏ ngăn xếp bóng (Shadow Stack Pointer, SSP) trong các thanh ghi tương ứng, mỗi SSP được liên kết với mức ưu tiên sự kiện khác nhau;

chọn SSP thứ nhất trong số các SSP từ thanh ghi thứ nhất trong số các thanh ghi đáp ứng lại việc tiếp nhận sự kiện thứ nhất được liên kết với mức ưu tiên sự kiện thứ nhất, SSP thứ nhất sử dụng được để nhận dạng đỉnh của ngăn xếp bóng thứ nhất;

xác định xem liệu SSP thứ nhất đã được xác minh trước đó hay chưa,

trong đó nếu SSP thứ nhất chưa được xác minh trước đó thì khởi tạo tập hợp các hoạt động nguyên tử để xác minh SSP thứ nhất và xác nhận rằng SSP thứ nhất đang không được sử dụng, tập hợp các hoạt động nguyên tử này sử dụng hoạt động khóa để khóa dữ liệu cho đến khi hoàn thành tập hợp các hoạt động nguyên tử, và

trong đó nếu SSP thứ nhất được xác minh trước đó, thì xác minh lại SSP thứ nhất và xác nhận rằng SSP thứ nhất đang không được sử dụng mà không sử dụng hoạt động khóa.

11. Phương pháp theo điểm 10 trong đó các SSP bao gồm các SSP giám sát, bao gồm SSP thứ nhất, trở đến các ngăn xếp bóng giám sát tương ứng được liên kết với các mức quyền ưu tiên giám sát.

12. Phương pháp theo điểm 11 trong đó các SSP còn bao gồm ít nhất một SSP mức người dùng chỉ đến ngăn xếp bóng mức người dùng được liên kết với mức quyền ưu tiên người dùng.

13. Phương pháp theo điểm 10 trong đó sự kiện thứ nhất bao gồm ngắt thứ nhất hoặc ngoại lệ thứ nhất được liên kết với mức ưu tiên thứ nhất.

14. Phương pháp theo điểm 13 trong đó các SSP được liên kết với các mức ưu tiên sự kiện tương ứng.

15. Phương pháp theo điểm 10 trong đó bước xác nhận rằng SSP thứ nhất đang không được sử dụng bao gồm bước kiểm tra giá trị bit bận được liên kết với SSP thứ nhất.

16. Phương pháp theo điểm 15 trong đó bước xác minh và xác minh lại SSP thứ nhất bao gồm bước so sánh SSP thứ nhất với giá trị mã thông báo thứ nhất được lưu trữ ở đỉnh của ngăn xếp bóng thứ nhất.

17. Phương pháp theo điểm 16, trong đó phương pháp này còn bao gồm bước:

thực thi một hoặc nhiều lệnh sử dụng ngăn xếp dữ liệu thứ nhất được liên kết với ngăn xếp bóng thứ nhất ở mức quyền ưu tiên giám sát thứ nhất được liên kết với ngăn xếp dữ liệu thứ nhất.

18. Vật ghi đọc được bằng máy có mã chương trình được lưu trữ trên đó mà, khi được thực thi bởi máy, làm cho máy này thực hiện các hoạt động trong số:

lưu trữ các con trỏ ngăn xếp bóng (Shadow Stack Pointer, SSP) trong các thanh ghi tương ứng, mỗi SSP được liên kết với mức ưu tiên sự kiện khác nhau;

chọn SSP thứ nhất trong số các SSP từ thanh ghi thứ nhất trong số các thanh ghi đáp ứng lại việc tiếp nhận sự kiện thứ nhất được liên kết với mức ưu tiên sự kiện thứ nhất, SSP thứ nhất sử dụng được để nhận dạng đỉnh của ngăn xếp bóng thứ nhất;

xác định xem liệu SSP thứ nhất đã được xác minh trước đó hay chưa,

trong đó nếu SSP thứ nhất chưa được xác minh trước đó thì khởi tạo tập hợp các hoạt động nguyên tử để xác minh SSP thứ nhất và xác nhận rằng SSP thứ nhất đang không được sử dụng, tập hợp các hoạt động nguyên tử này sử dụng hoạt động khóa để khóa dữ liệu cho đến khi hoàn thành tập hợp các hoạt động nguyên tử, và

trong đó nếu SSP thứ nhất đã được xác minh trước đó, thì xác minh lại SSP thứ nhất và xác nhận rằng SSP thứ nhất đang không được sử dụng mà không sử dụng hoạt động khóa.

19. Vật ghi đọc được bằng máy theo điểm 18 trong đó các SSP bao gồm các SSP giám sát, bao gồm SSP thứ nhất, trở đến các ngăn xếp bóng giám sát tương ứng được liên kết với các mức quyền ưu tiên giám sát.

20. Vật ghi đọc được bằng máy theo điểm 19 trong đó các SSP còn bao gồm ít nhất một SSP mức người dùng trở đến ngăn xếp bóng mức người dùng được liên kết với mức quyền ưu tiên người dùng.

21. Vật ghi đọc được bằng máy theo điểm 18 trong đó sự kiện thứ nhất bao gồm ngắt thứ nhất hoặc ngoại lệ thứ nhất được liên kết với mức ưu tiên thứ nhất.

22. Vật ghi đọc được bằng máy theo điểm 21 trong đó các SSP được liên kết với các mức ưu tiên sự kiện tương ứng.

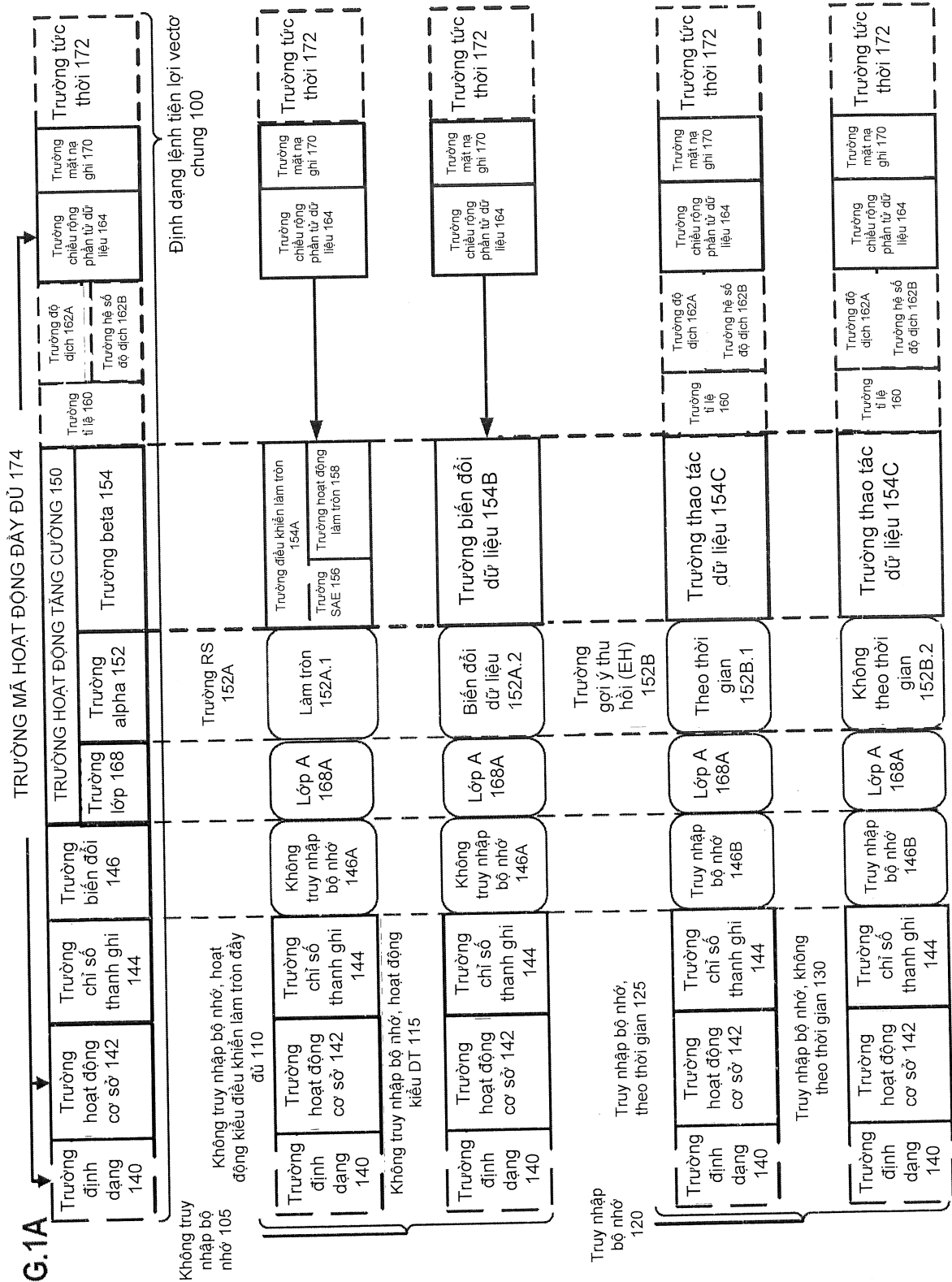
23. Vật ghi đọc được bằng máy theo điểm 18 trong đó bước xác nhận rằng SSP thứ nhất đang không được sử dụng bao gồm kiểm tra giá trị bit bận được liên kết với SSP thứ nhất.

24. Vật ghi đọc được bằng máy theo điểm 23 trong đó bước xác minh và xác minh lại SSP thứ nhất bao gồm bước so sánh SSP thứ nhất với giá trị mã thông báo thứ nhất được lưu trữ ở đỉnh của ngăn xếp bóng thứ nhất.

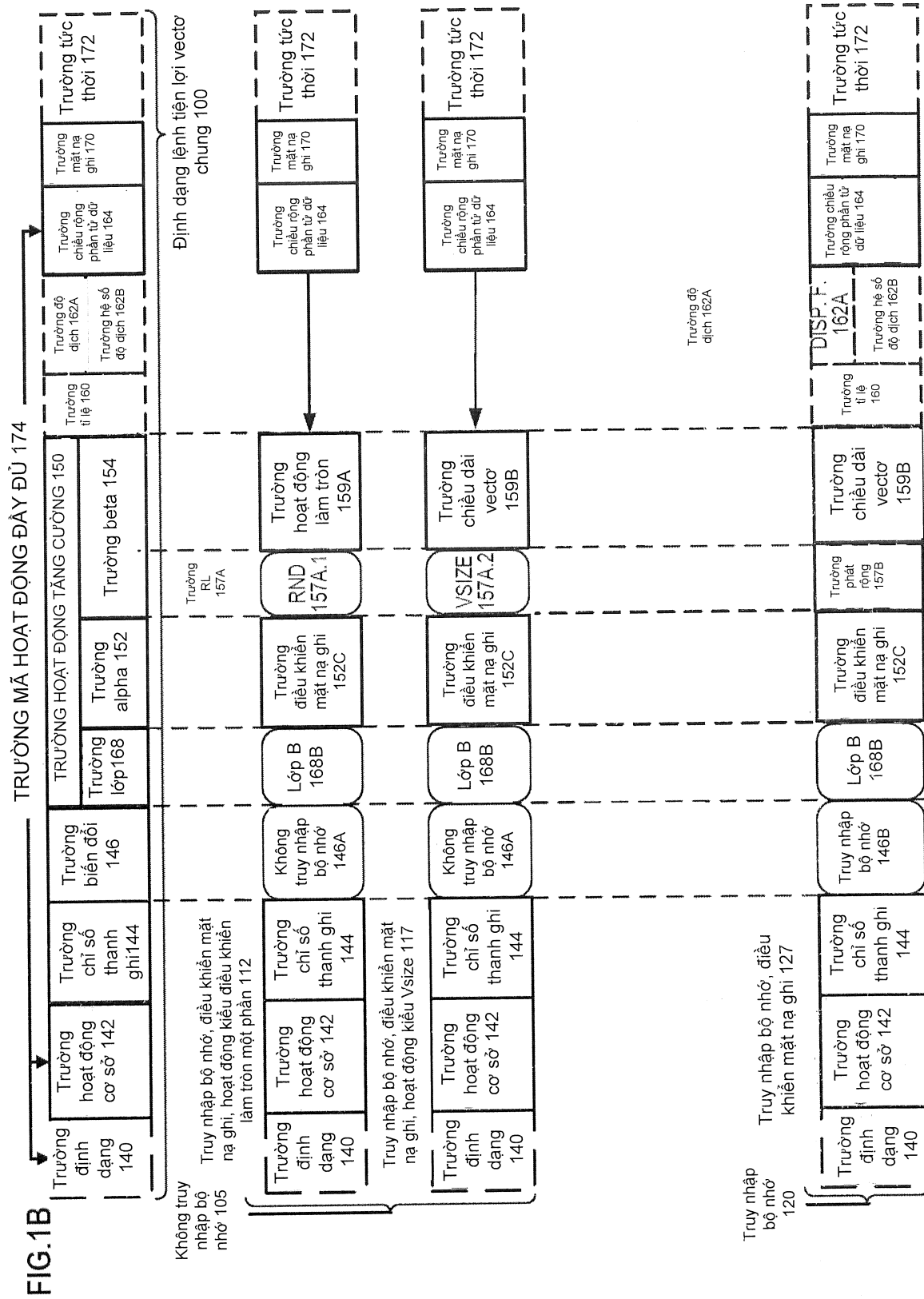
25. Vật ghi đọc được bằng máy theo điểm 24 còn bao gồm mã chương trình làm cho máy thực hiện các hoạt động trong số:

thực thi một hoặc nhiều lệnh sử dụng ngăn xếp dữ liệu thứ nhất được liên kết với ngăn xếp bóng thứ nhất ở mức quyền ưu tiên giám sát thứ nhất được liên kết với ngăn xếp dữ liệu thứ nhất.

FIG. 1A



2/20



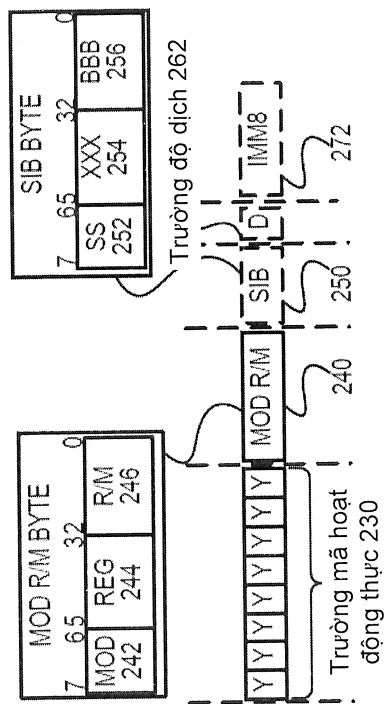


FIG. 2C

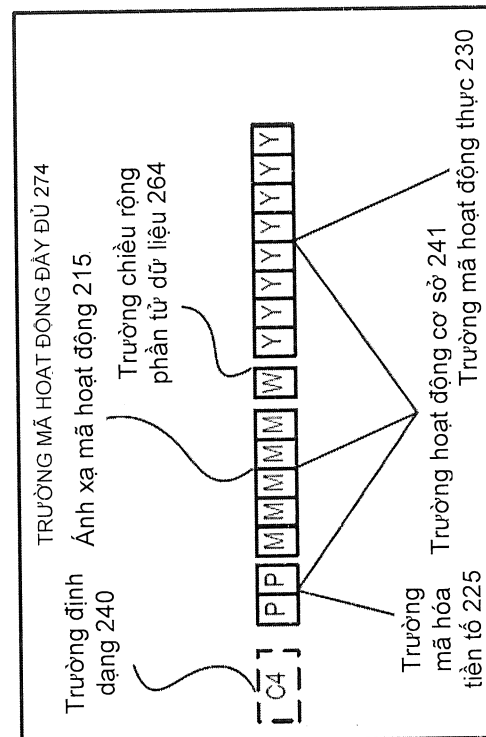
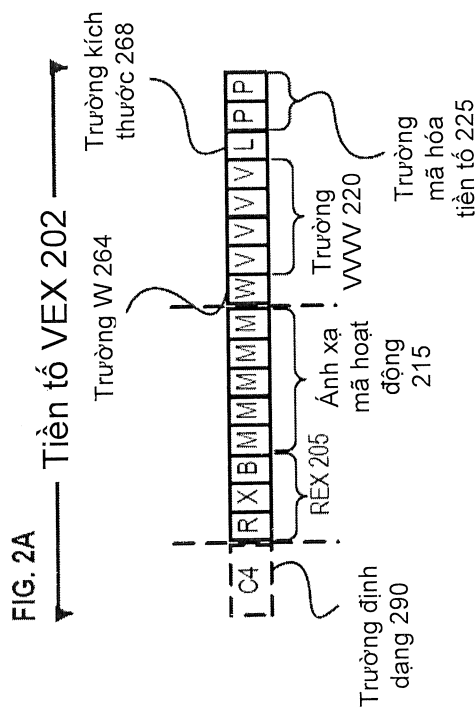
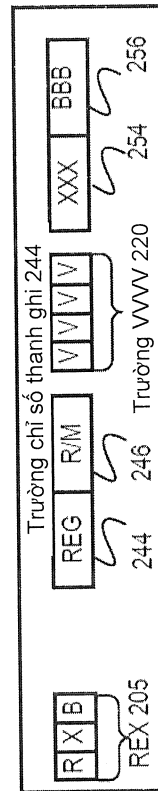
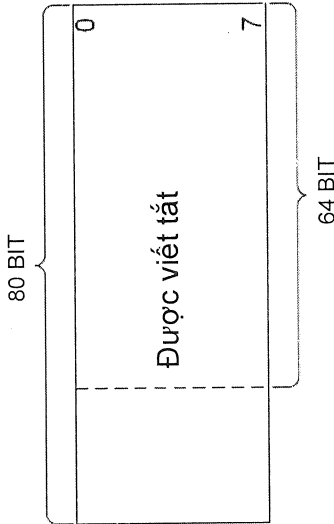


FIG. 2B

FIG. 3

Tập thanh ghi ngăn xếp số thực
345 (X87FP)



Tập thanh ghi phẳng INT
được đóng gói MMX 350

Kiến trúc thanh ghi 300

Thanh ghi đa năng 325
16 X 64 BIT

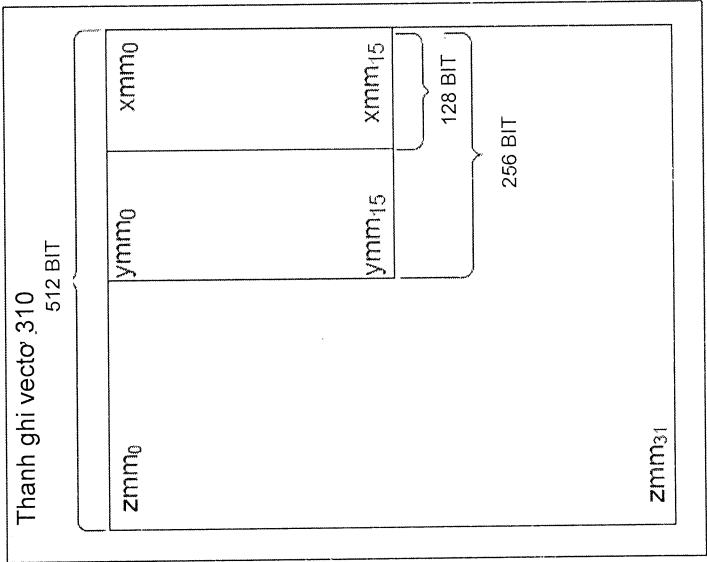


FIG. 4A

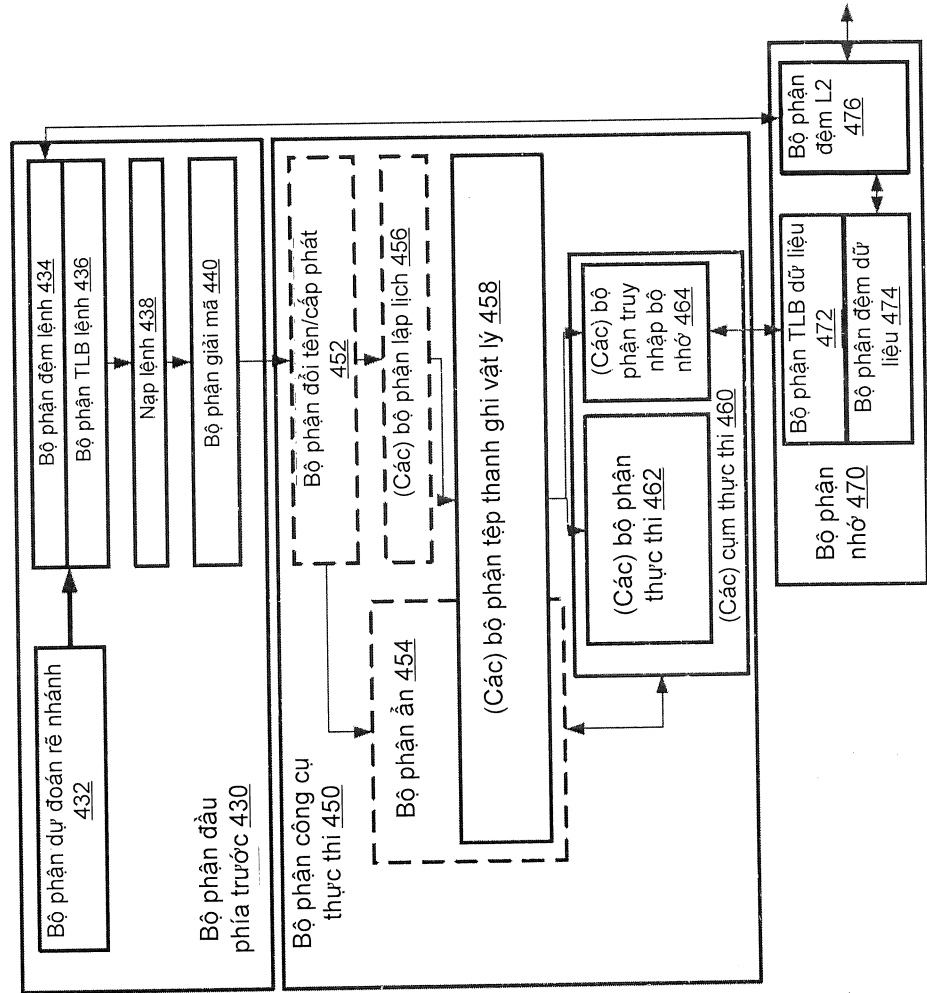
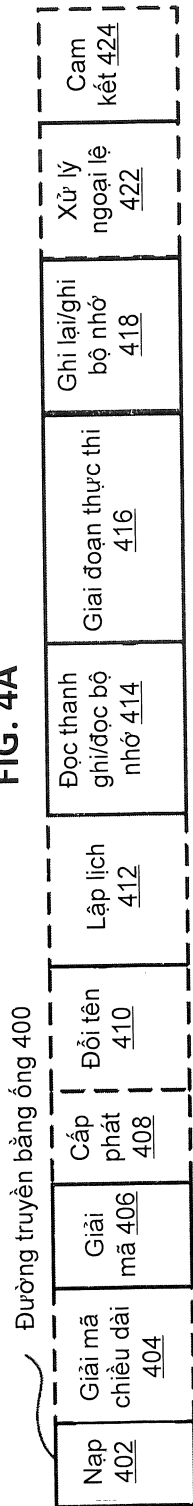


FIG. 4B

FIG. 5A

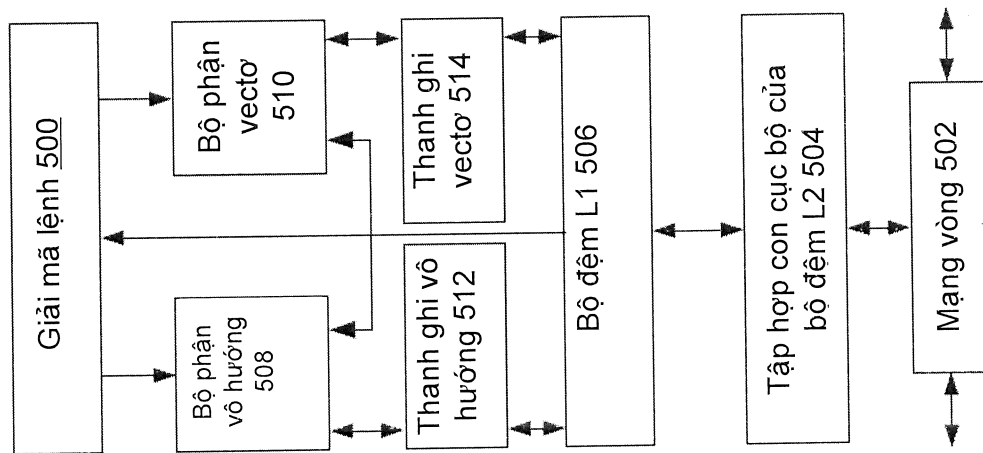
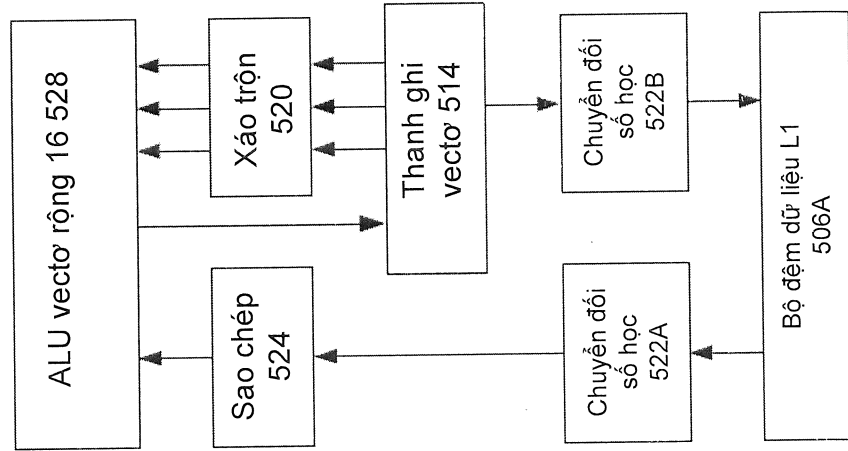
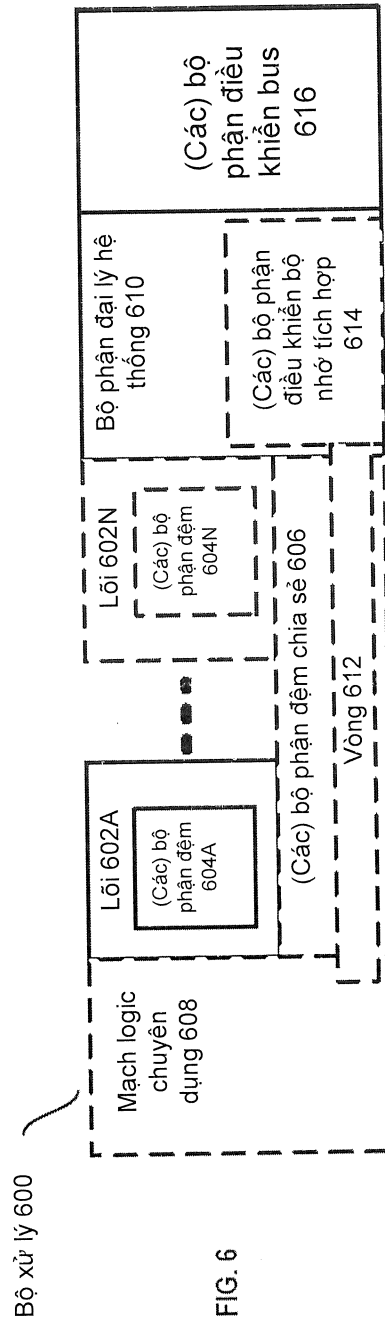


FIG. 5B



7/20



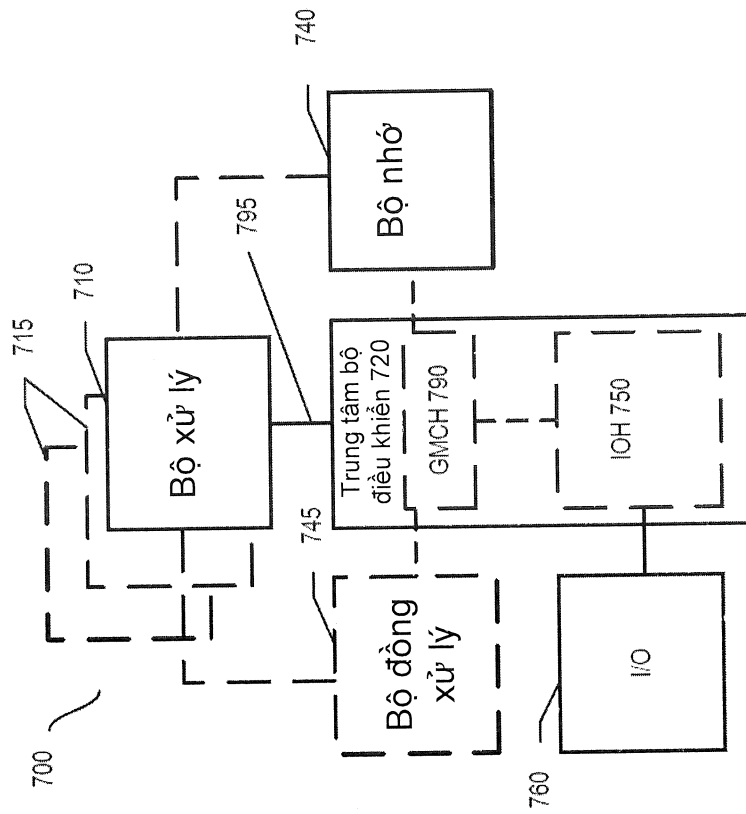
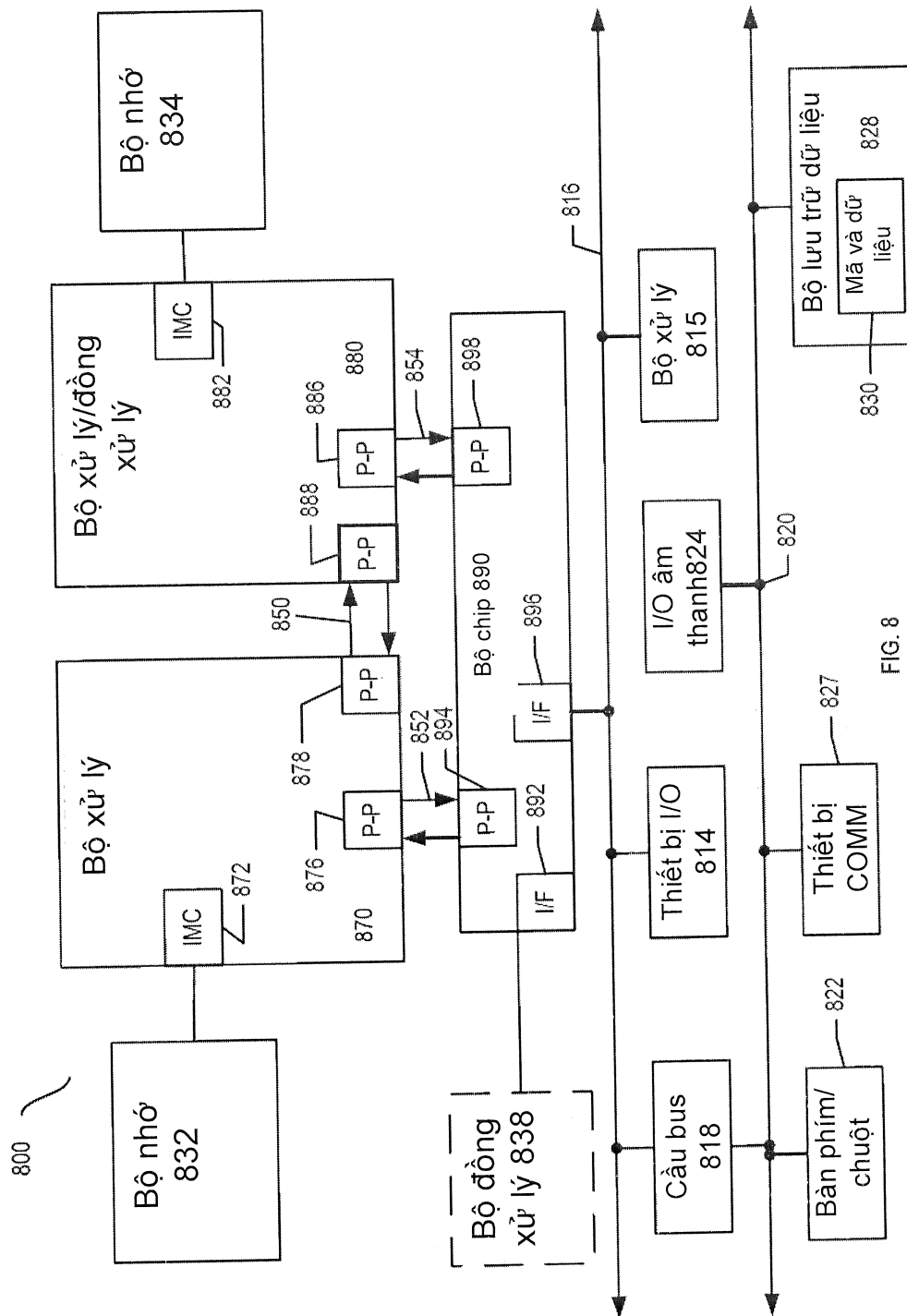


FIG. 7

9/20



10/20

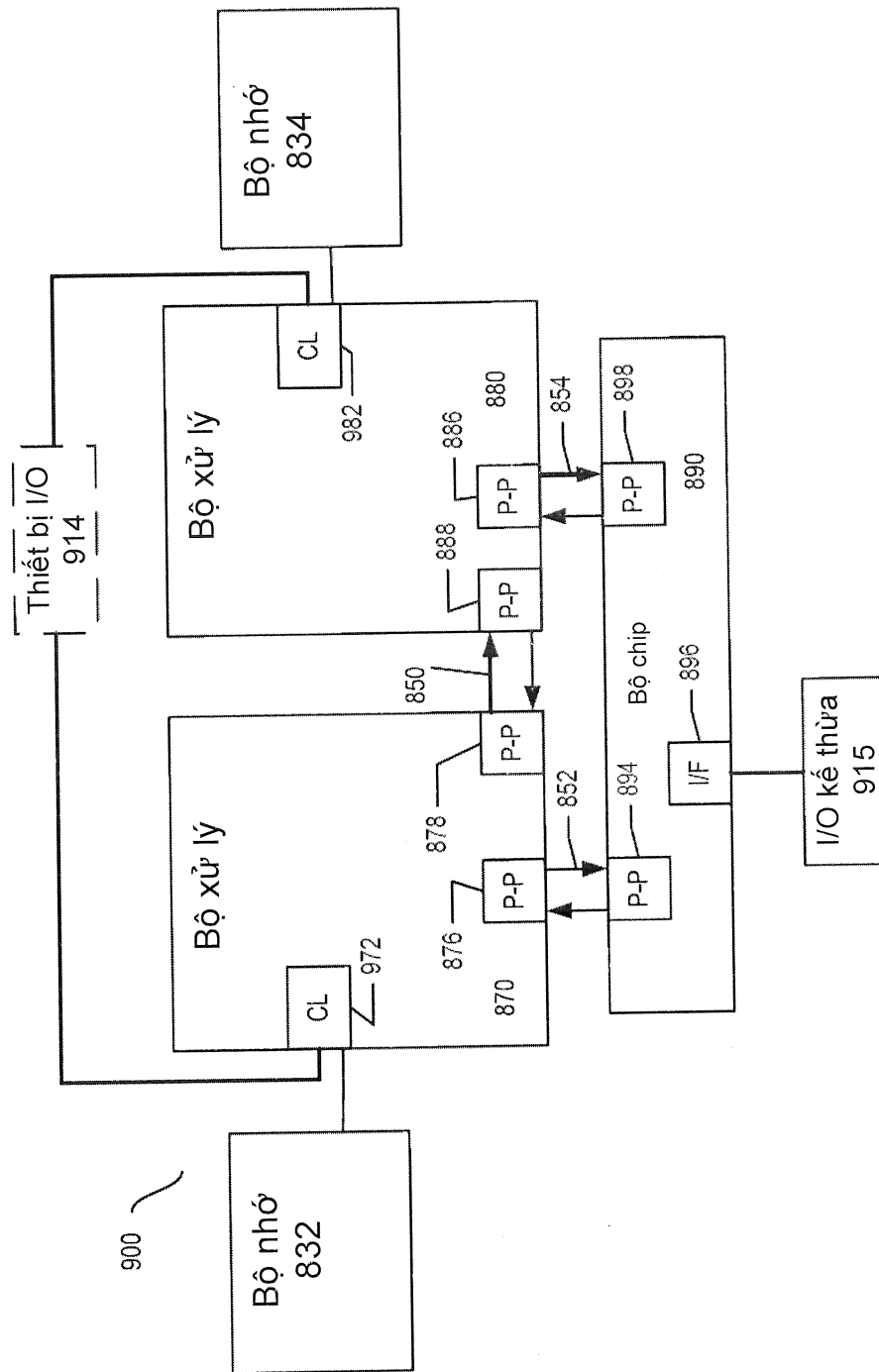


FIG. 9

11/20

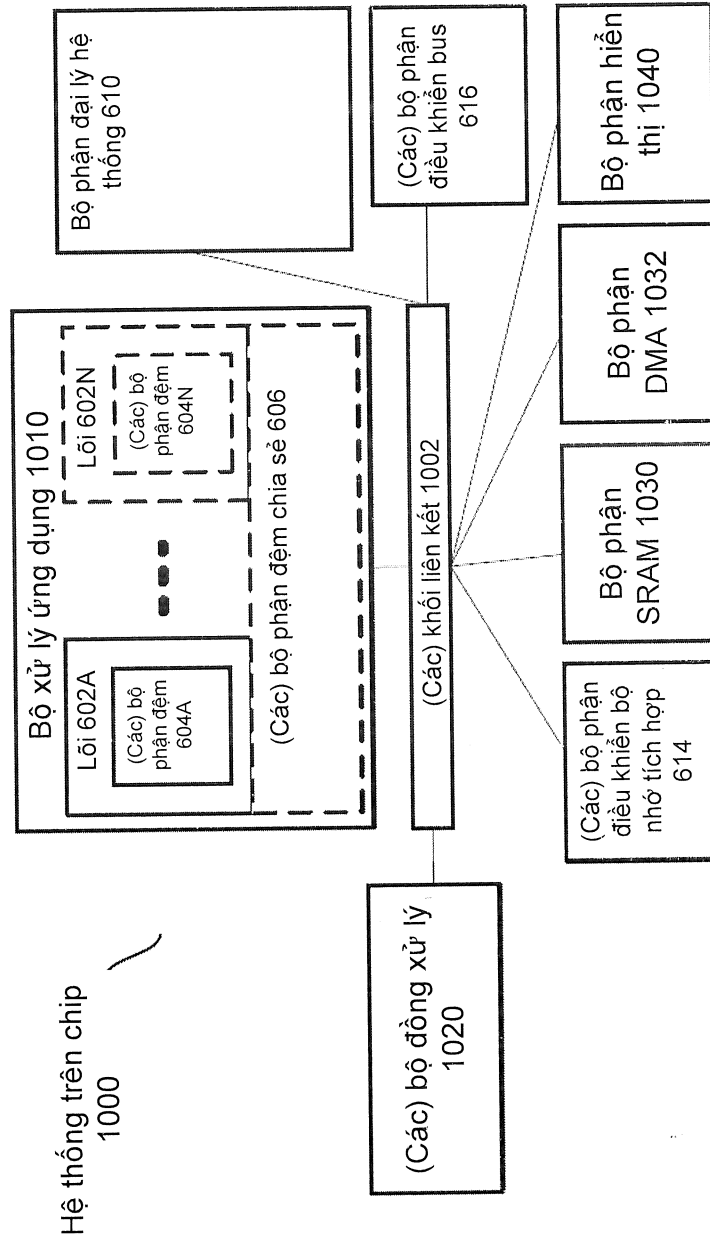
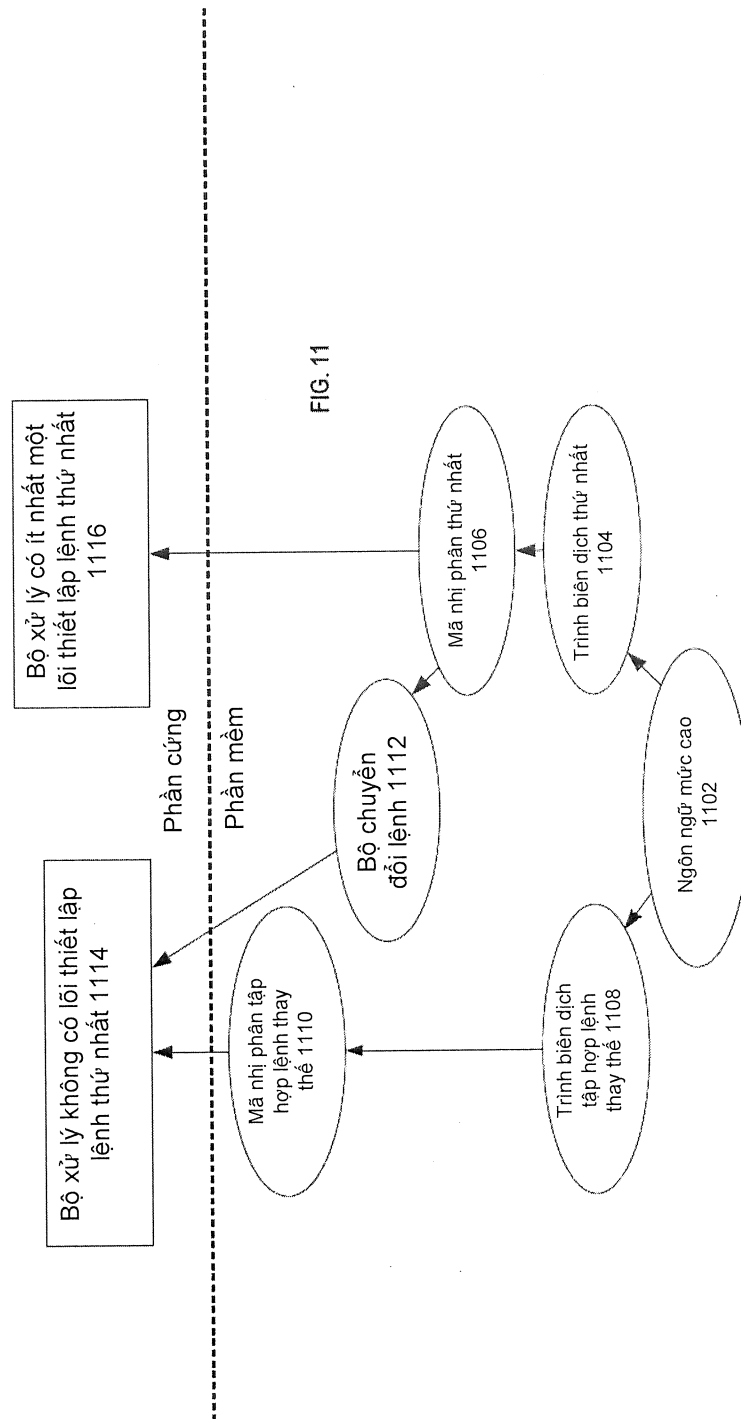


FIG. 10

12/20



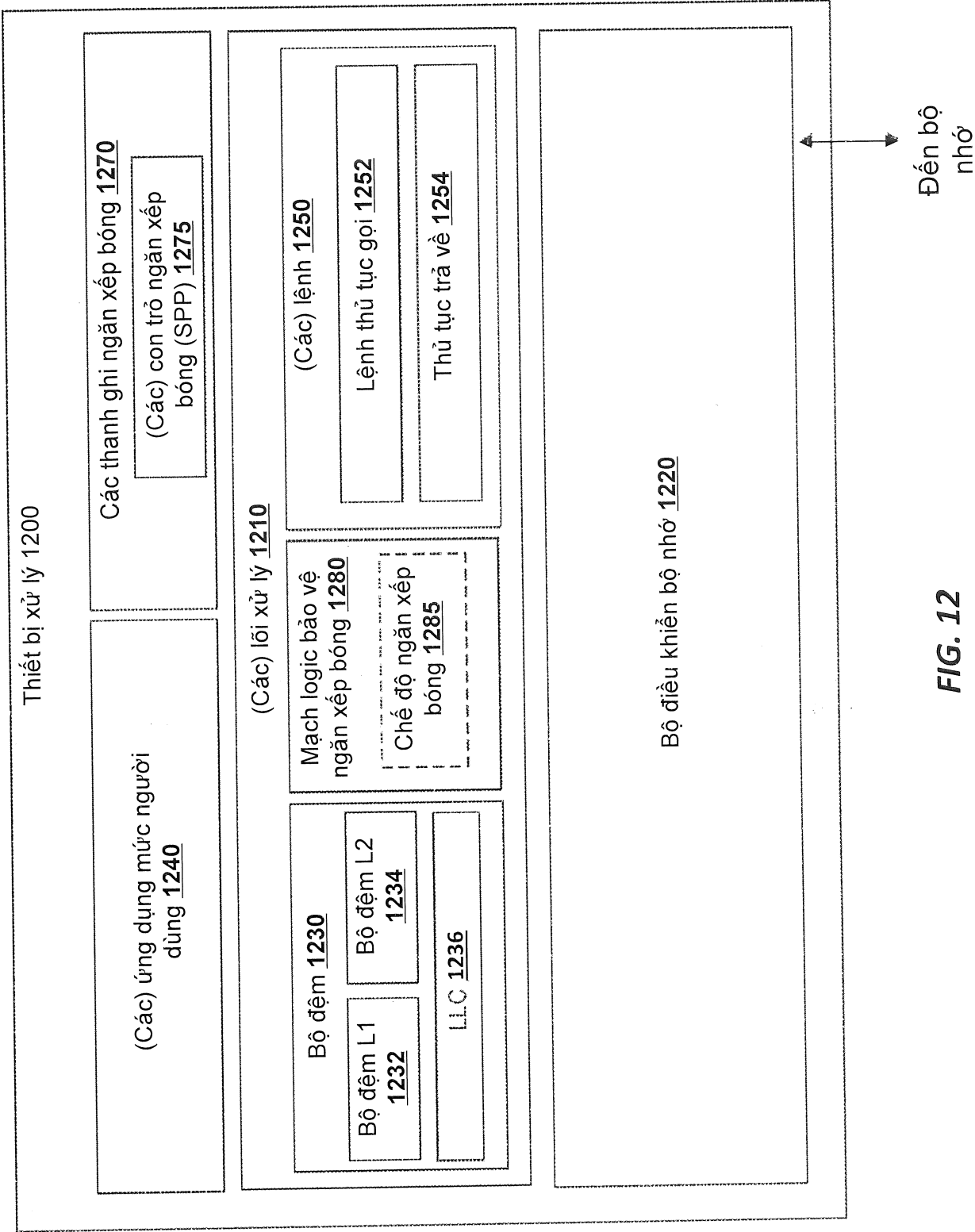


FIG. 12

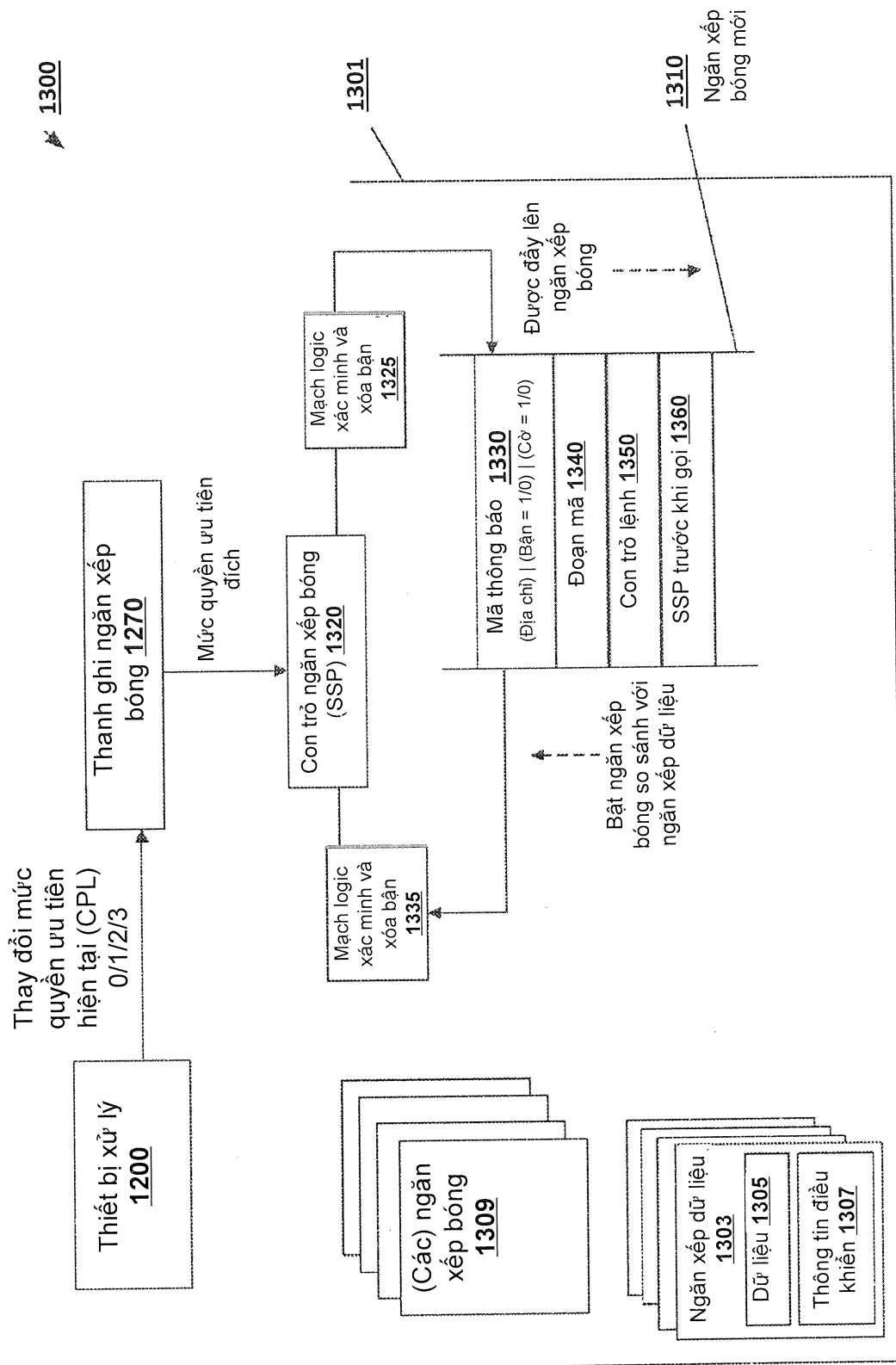


FIG. 13

15/20

1400

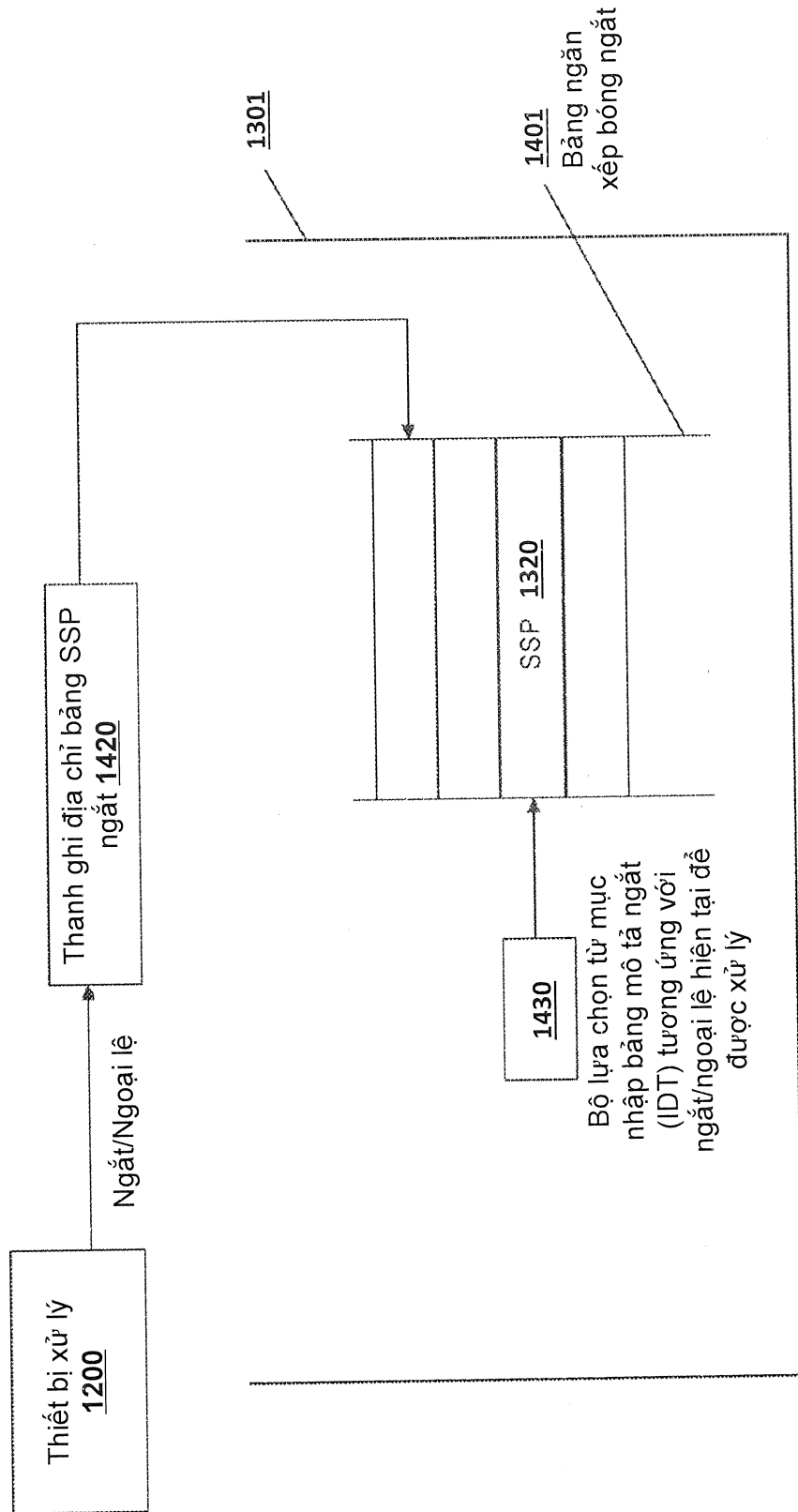


FIG. 14

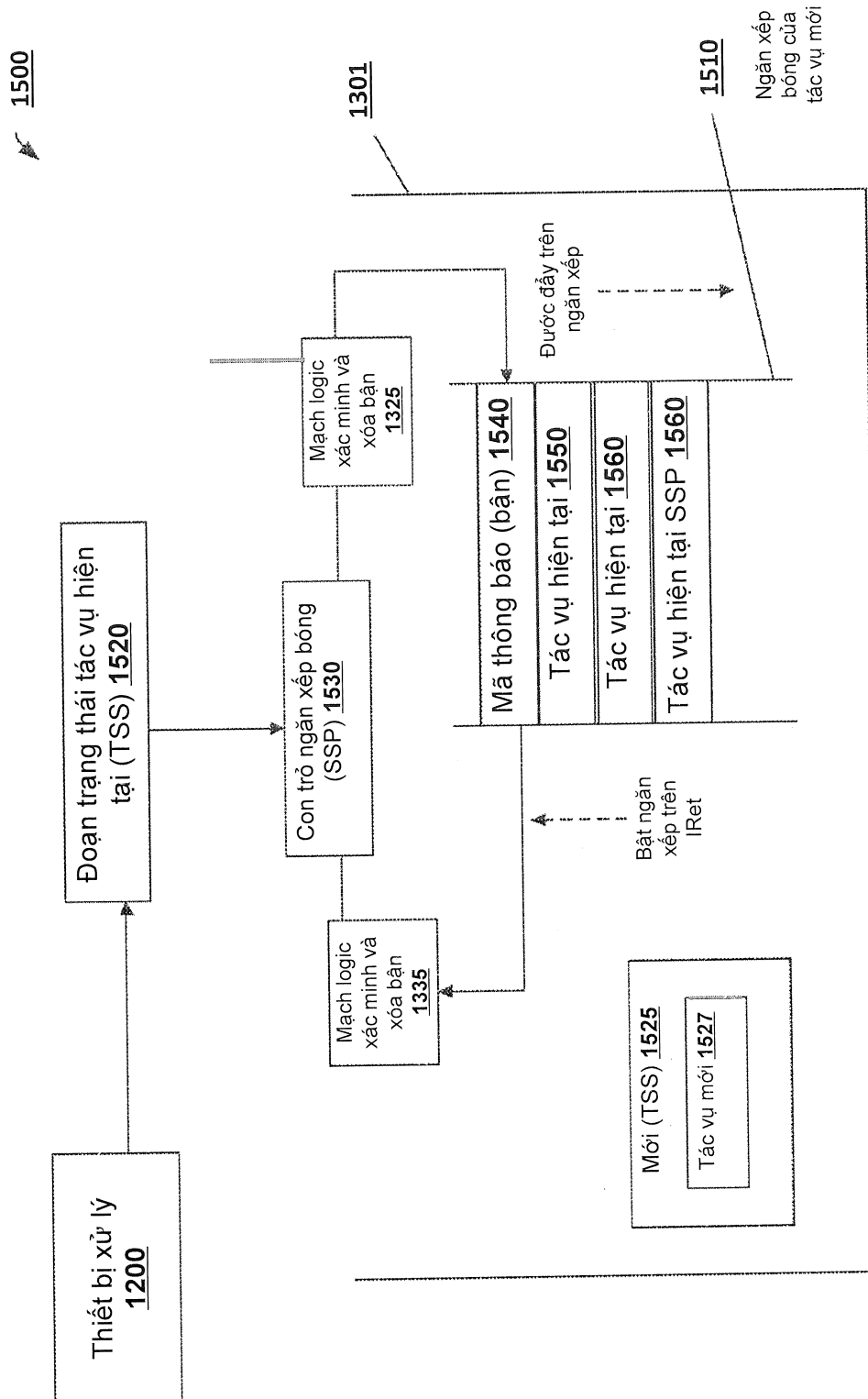


FIG. 15

17/20

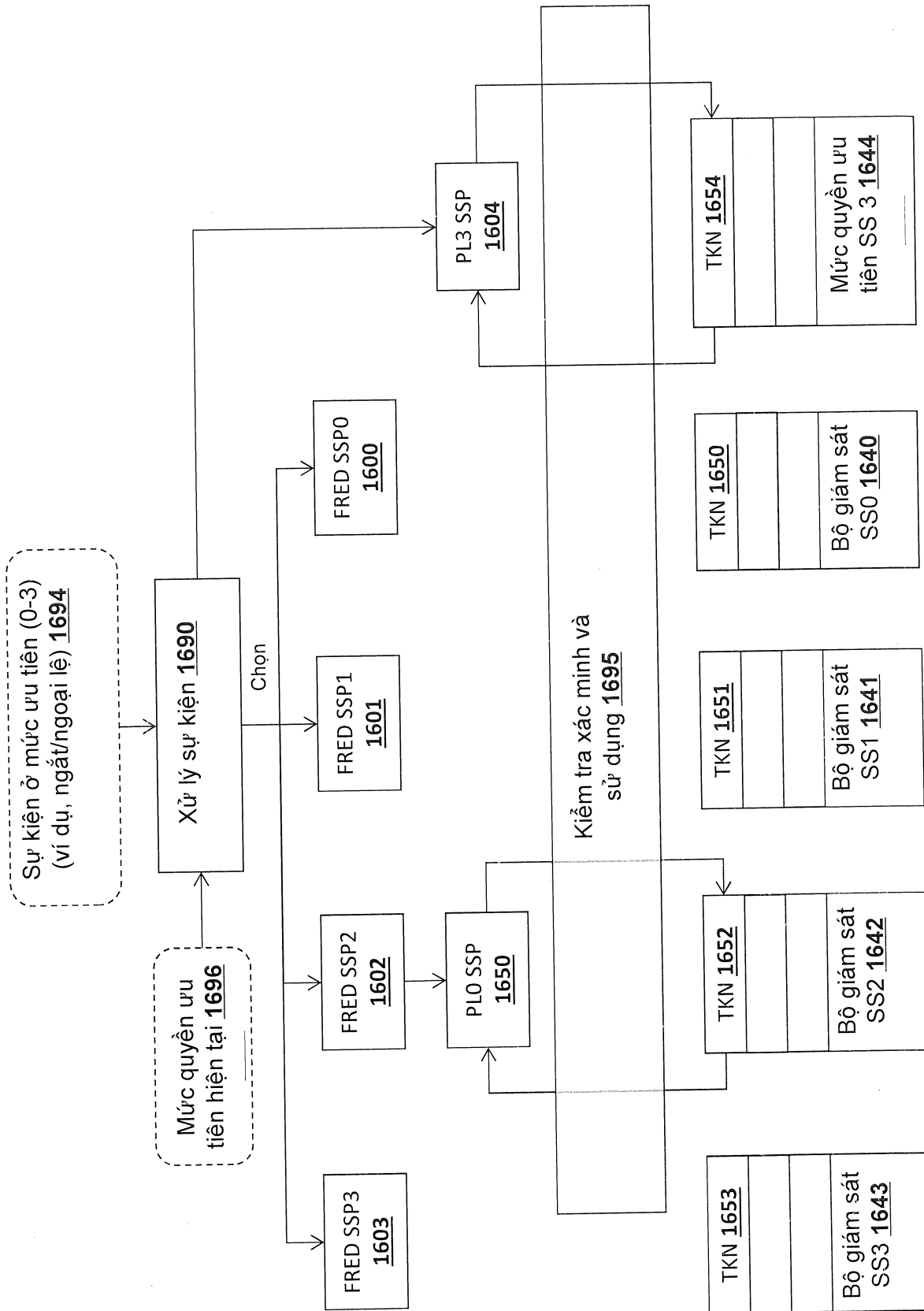


FIG. 16

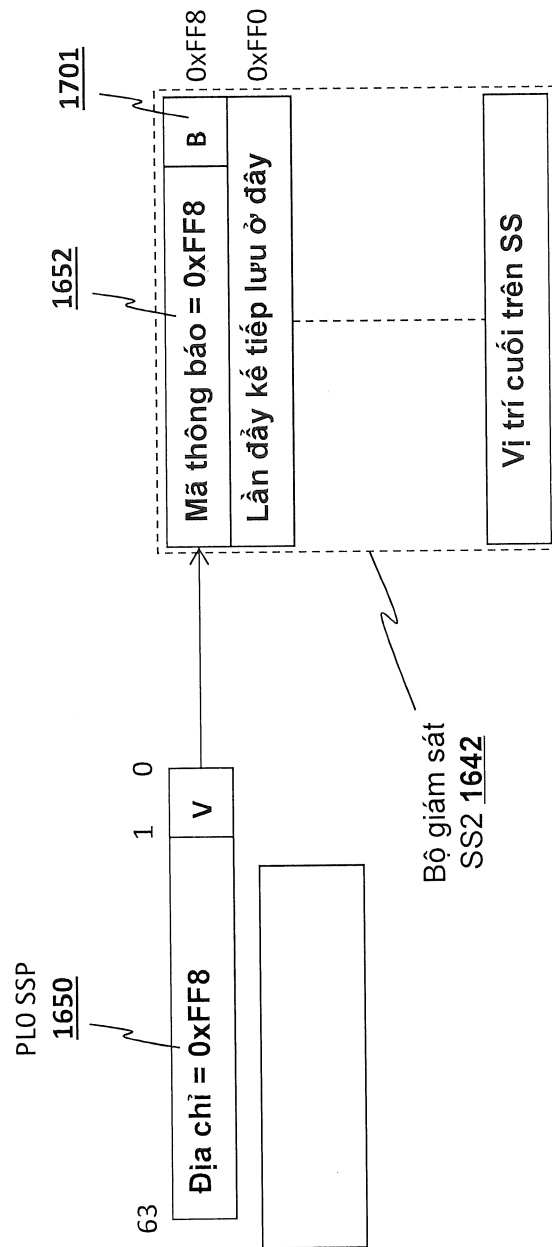


FIG. 17

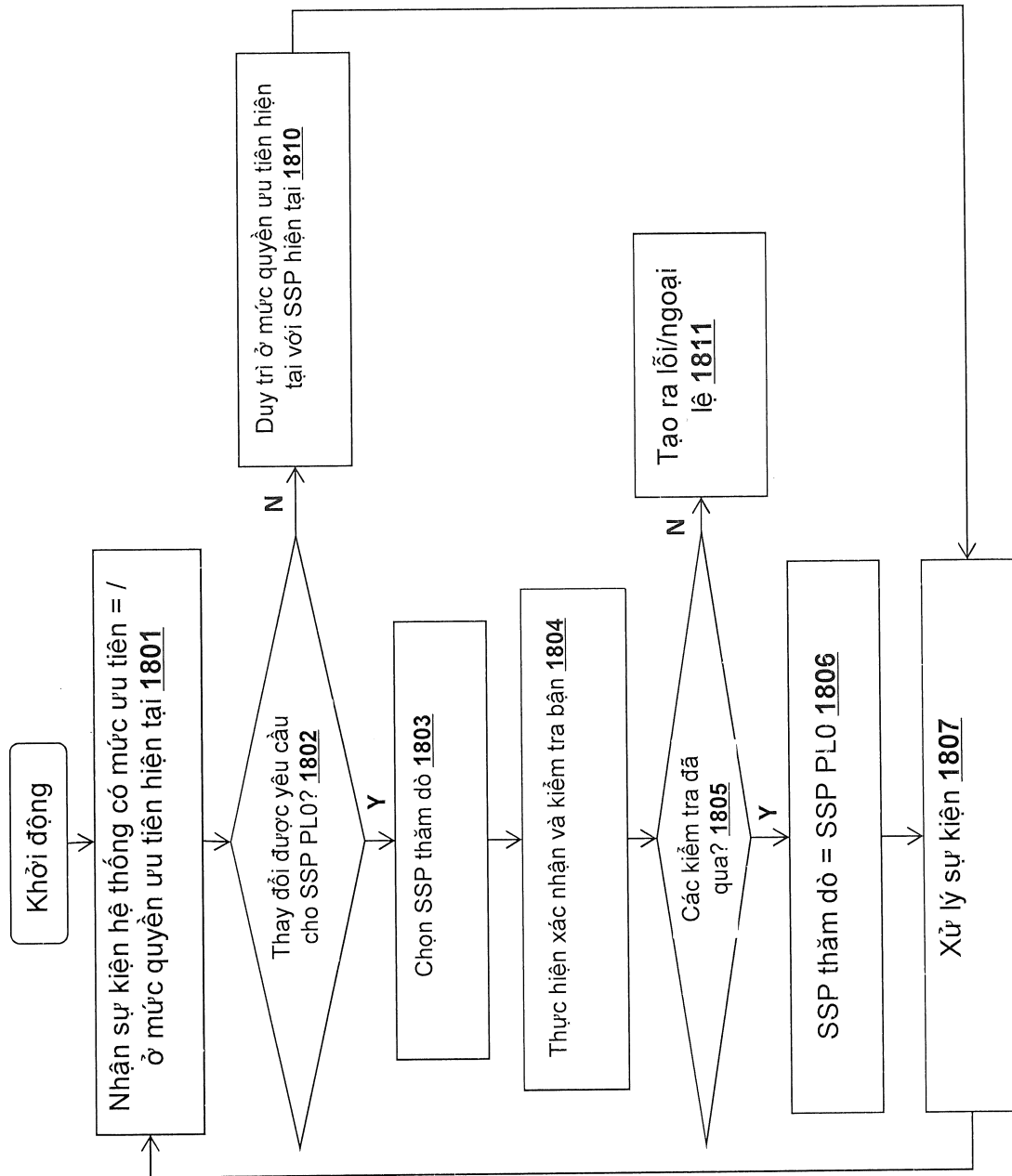


FIG. 18

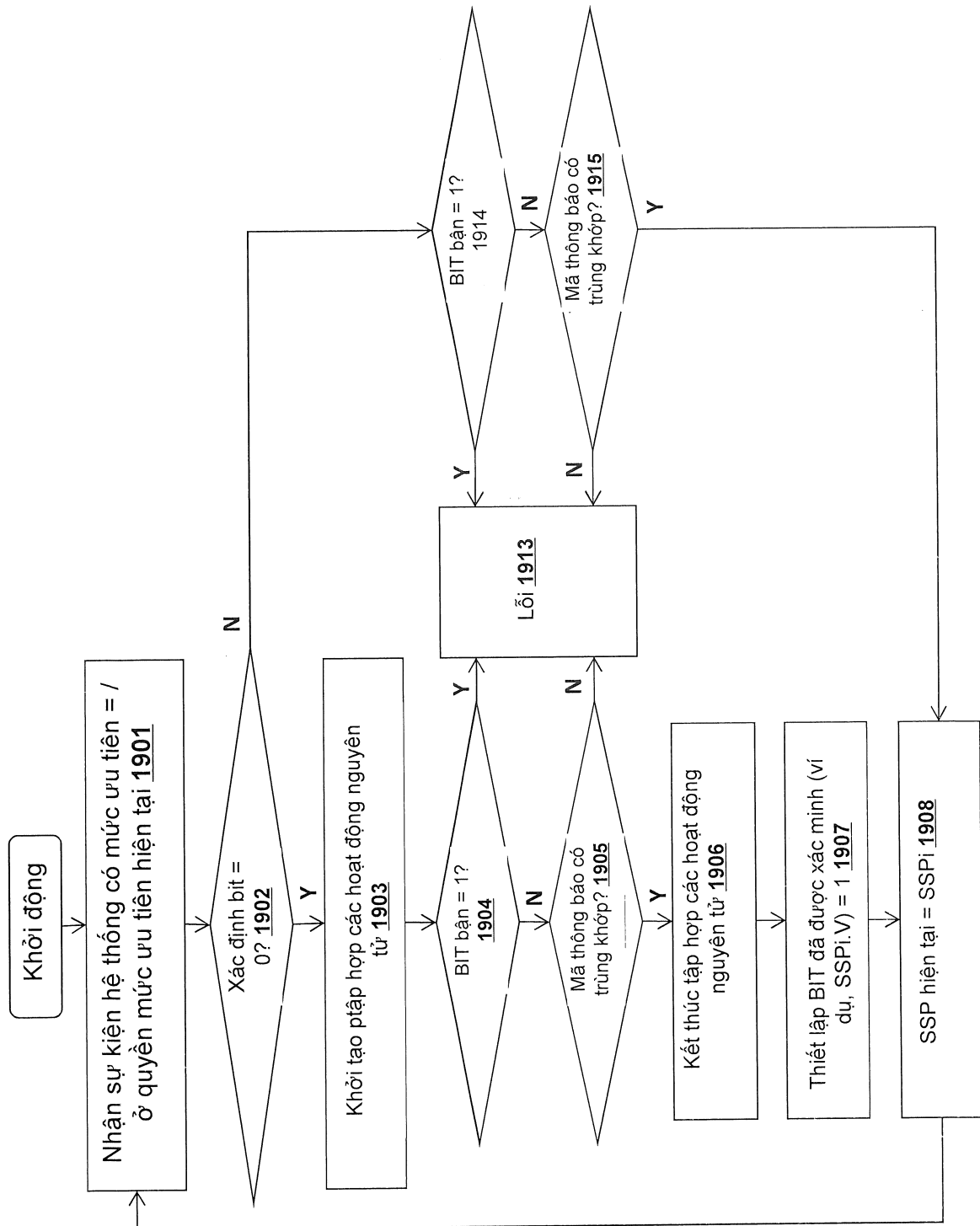


FIG. 19