(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) **1-0052452**

(51)[2022.01] **G06F 21/57**; G06F 21/62

(13) **B**

(21) 1-2022-08095

(22) 24/05/2021

(86) PCT/US2021/033930 24/05/2021

(87) WO 2021/257251 A1 23/12/2021

(30) 16/903,982 17/06/2020 US

(45) 27/10/2025 451

(43) 25/04/2023 421A

(73) Qualcomm Incorporated (US)
Attn: International IP Administration, 5775 Morehouse Drive, San Diego, CA 92121-1714, United States of America

(72) HALTER, Steven (US); ASBE, Samar (IN); BALLESTEROS, Miguel (US); BHAT, Girish (IN); NEMANI, Mahadevamurty (US).

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

(54) PHƯƠNG PHÁP, HỆ THỐNG VÀ PHƯƠNG TIỆN ĐỌC ĐƯỢC BẰNG MÁY TÍNH ĐỂ ĐIỀU KHIỂN TRUY CẬP TÀI NGUYÊN TRONG HỆ THỐNG TRÊN CHIP

(21) 1-2022-08095

(57) Sáng chế đề cập đến điều khiển truy cập tài nguyên trong hệ thống trên chip (system-on-chip - SoC) có thể sử dụng thành phần thực thi trên bộ xử lý của SoC và công cụ quản lý độ tin cậy của SoC. Thành phần, chẳng hạn, ví dụ, hệ điều hành cấp cao hoặc bộ quản lý siêu dữ liệu, có thể được tạo cấu hình để phân bổ tài nguyên bao gồm vùng bộ nhớ cho miền truy cập và để tải hình ảnh phần mềm liên quan đến miền truy cập vào vùng bộ nhớ. Công cụ quản lý độ tin cậy có thể được tạo cấu hình để khóa tài nguyên chống lại truy cập bởi bất kỳ thực thể nào không phải miền truy cập, để xác thực hình ảnh phần mềm liên quan đến miền truy cập và để bắt đầu khởi động miền truy cập để đáp lại xác thực thành công hình ảnh phần mềm liên quan đến miền truy cập. Cụ thể, sáng chế đề cập đến phương pháp, hệ thống, và phương tiện đọc được bằng máy tính để điều khiển truy cập tài nguyên trong hệ thống trên chip.

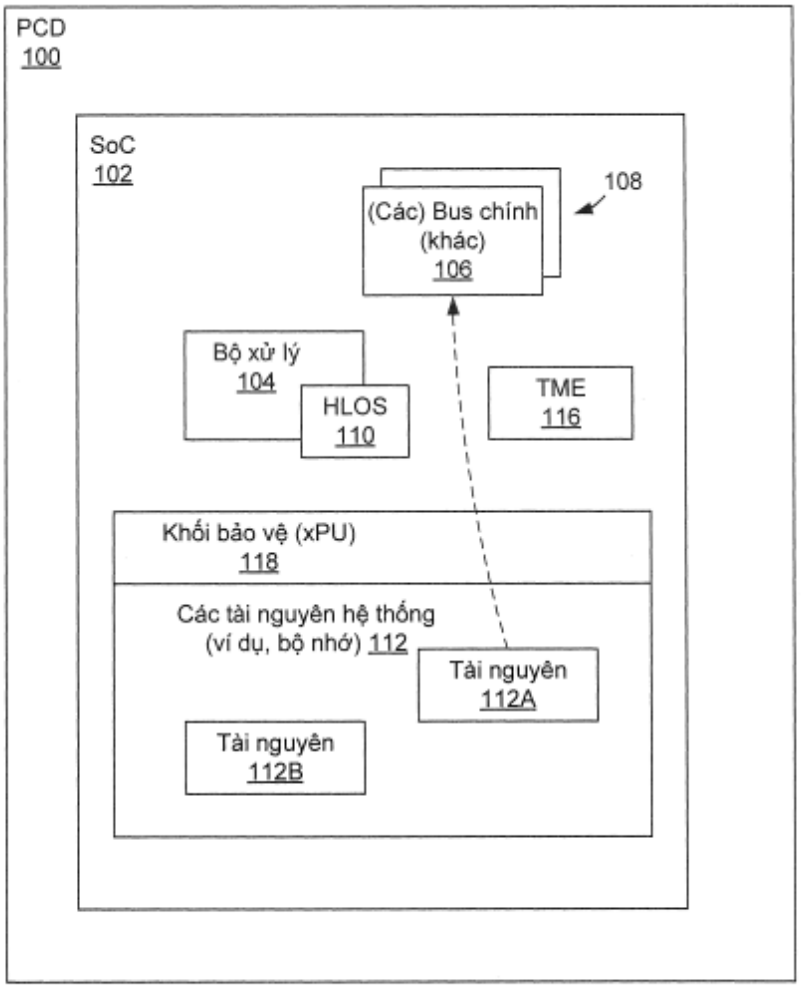


Fig.1

Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến phương pháp và hệ thống điều khiển truy cập tài nguyên trong hệ thống trên chip.

Tình trạng kỹ thuật của sáng chế

Thiết bị máy tính cầm tay (portable computing device - PCD) đang trở thành vật dụng thiết yếu với con người ở cấp độ cá nhân và cấp độ chuyên nghiệp. Các thiết bị này có thể bao gồm điện thoại di động, máy tính bảng, máy tính cầm tay, thiết bị trợ giúp số cầm tay (portable digital assistant - PDA), bàn giao tiếp trò chơi cầm tay và các thiết bị điện tử cầm tay khác. Các PCD thường chứa các mạch tích hợp hoặc hệ thống trên chip (systems on a chip - SoC) bao gồm nhiều thành phần được thiết kế để làm việc cùng nhau nhằm cung cấp chức năng cho người dùng. Ví dụ, SoC có thể chứa số lượng máy xử lý bất kỳ như các mođem, khối xử lý trung tâm (central processing unit - CPU) với nhiều lõi, các khối xử lý đồ họa (central processing unit - GPU), v.v. SoC có thể được ghép nối với các thành phần khác trong PCD, như bộ nhớ hệ thống, bộ thu phát truyền thông không dây (còn được gọi là các mođem), các camera, micro, loa v.v. SoC và các thành phần khác có thể được ghép nối bởi một hoặc nhiều bus hoặc các kết nối khác để cung cấp truyền thông dữ liệu giữa chúng. Thuật ngữ "tài nguyên" có thể được sử dụng để chỉ một thành phần hoặc một phần của thành phần như vùng của bộ nhớ, thanh ghi, cổng, v.v. mà bộ xử lý có thể thu được quyền truy cập vào đó thông qua giao dịch bus.

Duy trì bảo mật chống lại truy cập tài nguyên trái phép là một điểm quan trọng cần cân nhắc trong thiết kế SoC. Độ phức tạp của SoC làm phát sinh các yêu cầu bảo mật xung đột. Ví dụ, một mặt, hệ điều hành cấp cao ("high-level operating system - HLOS") chạy trên một bộ xử lý trong SoC có thể cần có khả năng giới hạn quyền truy cập vào các tài nguyên từ các thực thể chạy trên các bộ xử lý SoC khác. Mặt khác, nhà cung cấp SoC có thể mong muốn giới hạn quyền truy cập của HLOS vào các bộ xử lý SoC khác như vậy vì sự truy cập như vậy có thể có tiềm năng làm lộ tài sản trí tuệ của nhà cung cấp SoC. Nói chung, bộ nhớ chính hay bộ nhớ hệ thống là tài nguyên cơ bản và quan trọng nhất, và thông thường HLOS quản lý toàn bộ bộ nhớ hệ thống. Cũng có thể tồn tại nhu

cầu về các thực thể trong các miền bảo mật khác nhau để phối hợp và việc chia bản đồ bộ nhớ hệ thống để dành riêng các phần của bộ nhớ hệ thống cho các miền bảo mật khác nhau có thể là điều không mong muốn.

Bảo mật thông thường được cung cấp trong thiết bị dựa trên SoC theo một trong hai cách sau. Một cách là cung cấp phần cứng SoC để điều khiển bảo mật trên tất cả các hệ thống con. Hạn chế của phương pháp này là ở chỗ nó không thể điều chỉnh được. Nó không linh hoạt cho việc bổ sung các miền bảo mật mới và nó có thể bao gồm bảo mật cho hệ thống con. Một cách khác là cho phép HLOS điều khiển bảo mật trên tất cả các hệ thống con bằng cách cung cấp dạng thực thể siêu người dùng. Nhược điểm của cách này là ở chỗ bảo mật có thể bị lộ với nguồn mở và có thể bị gây tổn hại. Cả hai giải pháp truyền thống này đều không giải quyết được các mối quan ngại về bảo mật xung đột như nêu trên đây.

Bản chất kỹ thuật của sáng chế

Sáng chế đề xuất hệ thống, phương pháp và sản phẩm chương trình máy tính để điều khiển truy cập tài nguyên trong SoC.

Phương pháp ví dụ để điều khiển truy cập tài nguyên trong SoC có thể bao gồm phân bổ tài nguyên bao gồm vùng bộ nhớ cho miền truy cập. Một thành phần, chẳng hạn như HLOS, có thể điều khiển hoặc thực hiện việc phân bổ như vậy. Phương pháp này cũng bao gồm tải hình ảnh phần mềm liên quan đến miền truy cập vào vùng bộ nhớ. HLOS hoặc thành phần khác có thể điều khiển hoặc thực hiện việc tải này. Phương pháp này có thể còn bao gồm khóa tài nguyên chống lại việc truy cập của bất kỳ thực thể nào không phải miền truy cập. Công cụ quản lý độ tin cậy có thể điều khiển hoặc thực hiện việc khóa như vậy. Phương pháp này có thể vẫn còn bao gồm xác thực hình ảnh phần mềm liên quan đến miền truy cập. Công cụ quản lý độ tin cậy có thể điều khiển hoặc thực hiện việc xác thực như vậy. Phương pháp này có thể còn bao gồm thêm bắt đầu khởi động miền truy cập để đáp lại xác thực thành công hình ảnh phần mềm. Công cụ quản lý độ tin cậy có thể bắt đầu khởi động.

Hệ thống ví dụ để điều khiển truy cập tài nguyên trong SoC có thể bao gồm thành phần thực thi trên bộ xử lý và công cụ quản lý độ tin cậy. Hệ thống có thể được tạo cấu hình để phân bổ tài nguyên bao gồm vùng bộ nhớ tới miền truy cập và tải hình ảnh phần mềm liên quan đến miền truy cập vào trong vùng bộ nhớ. Công cụ quản lý độ tin cậy có

thể được tạo cấu hình để khóa tài nguyên chống lại truy cập bởi bất kỳ thực thể nào không phải miền truy cập, để xác thực hình ảnh phần mềm liên quan đến miền truy cập và để bắt đầu khởi động miền truy cập để đáp lại xác thực thành công hình ảnh phần mềm liên quan đến miền truy cập.

Một hệ thống ví dụ khác để điều khiển truy cập tài nguyên trong SoC có thể bao gồm phương tiện để phân bổ tài nguyên bao gồm vùng bộ nhớ tới miền truy cập và để tải hình ảnh phần mềm liên quan đến miền truy cập vào trong vùng bộ nhớ. Hệ thống ví dụ có thể còn bao gồm phương tiện để khóa tài nguyên chống lại truy cập của bất kỳ thực thể nào không phải miền truy cập, để xác thực hình ảnh phần mềm liên quan đến miền truy cập, và để bắt đầu khởi động miền truy cập để đáp lại xác thực thành công phần mềm hình ảnh.

Sản phẩm chương trình máy tính ví dụ để điều khiển truy cập tài nguyên trong SoC có thể bao gồm phương tiện đọc được bằng máy tính lưu trữ trên đó các lệnh mà khi được thực thi trên một hoặc nhiều bộ xử lý của SoC sẽ điều khiển phương pháp. Phương pháp bao gồm phân bổ tài nguyên bao gồm vùng bộ nhớ tới miền truy cập. Phương pháp này có thể còn bao gồm tải hình ảnh phần mềm liên quan đến miền truy cập vào trong vùng bộ nhớ. Phương pháp này có thể còn bao gồm khóa tài nguyên chống lại truy cập của bất kỳ thực thể nào không phải miền truy cập. Phương pháp này vẫn có thể còn bao gồm xác thực hình ảnh phần mềm liên quan đến miền truy cập. Phương pháp có thể còn bao gồm bắt đầu khởi động miền truy cập để đáp lại xác thực thành công hình ảnh phần mềm.

Mô tả vắn tắt các hình vẽ

Trên các hình vẽ, các số tham chiếu giống nhau chỉ các phần giống nhau trên các hình khác nhau trừ khi được chỉ dẫn khác. Đối với các số tham chiếu có các ký hiệu chữ cái như "102A" hoặc "102B", các ký hiệu chữ cái có thể phân biệt hai phần hoặc phần tử giống nhau có trong cùng một hình vẽ. Các ký hiệu chữ cái cho các số tham chiếu có thể được bỏ qua khi dự định rằng số tham chiếu bao gồm tất cả các phần có cùng một số tham chiếu trong tất cả các hình.

Fig.1 là sơ đồ khối của SoC bao gồm hệ thống điều khiển truy cập tài nguyên, theo các phương án làm ví dụ.

Fig.2 là sơ đồ khối thể hiện các bus chính và bus phụ trong hệ thống điều khiển

truy cập tài nguyên theo các phương án làm ví dụ.

Fig.3 là bản đồ bộ nhớ hệ thống thể hiện ví dụ về nhóm tài nguyên, theo các phương án làm ví dụ.

Fig.4 là lưu đồ minh họa phương pháp điều khiển truy cập tài nguyên liên quan đến khởi động miền truy cập, theo các phương án làm ví dụ.

Fig.5 là một lưu đồ khác minh họa phương pháp khởi động miền truy cập, theo các phương án làm ví dụ.

Fig.6 là lưu đồ minh họa phương pháp điều khiển truy cập tài nguyên liên quan đến phân bổ tài nguyên bổ sung, theo các phương án ví dụ.

Fig.7 là một lưu đồ khác minh họa phương pháp phân bổ tài nguyên bổ sung, theo các phương án làm ví dụ.

Fig.8 là lưu đồ minh họa phương pháp điều khiển truy cập tài nguyên liên quan đến hủy phân bổ tài nguyên bổ sung, theo các phương án làm ví dụ.

Fig.9 là một lưu đồ khác minh họa phương pháp hủy phân bổ tài nguyên bổ sung, theo các phương án làm ví dụ.

Fig.10 là sơ đồ khối của PCD, theo các phương án làm ví dụ.

Fig.11 là sơ đồ khối của giao diện chương trình ứng dụng cho khối bảo vệ, theo các phương án làm ví dụ.

Mô tả chi tiết sáng chế

Từ "làm ví dụ" được sử dụng ở đây có nghĩa là "dùng làm ví dụ, trường hợp, hoặc minh họa". Từ "minh họa" được dùng ở đây đồng nghĩa với "ví dụ". Bất kỳ khía cạnh nào được mô tả ở đây là "ví dụ" không nhất thiết phải được hiểu là ưu tiên hoặc có lợi so với các khía cạnh khác.

Như minh họa trên Fig.1, theo một phương án minh họa hoặc ví dụ, PCD 100 có thể bao gồm SoC 102. SoC 102 có thể bao gồm một hoặc nhiều bộ xử lý 104 và một hoặc nhiều bus chính 106 khác. Mặc dù bus hoặc kết nối hệ thống tương tự không được thể hiện trên Fig. 1 nhằm mục đích rõ ràng, nhưng thuật ngữ "bus chính" được sử dụng trong sáng chế có nghĩa là bất kỳ thành phần nào có khả năng khởi tạo giao dịch bus. Nhóm một hay nhiều bus chính có thể được gọi trong sáng chế này là miền truy cập hoặc

"AD" (access domain). Ví dụ, nhóm hai bus chính 106 như trên Fig. 1 có thể hình thành AD 108.

PCD 100 được dự định chỉ là một ví dụ về thiết bị mà SoC 102 có thể được bao gồm trong đó. Nói một cách tổng quát hơn, các ví dụ về các thiết bị mà có thể bao gồm SoC theo sáng chế bao gồm: hệ thống máy tính (ví dụ, máy chủ, trung tâm dữ liệu, máy tính để bàn), thiết bị máy tính cầm tay hoặc di động (ví dụ, máy tính xách tay, điện thoại di động, phương tiện giao thông v.v.), thiết bị internet vạn vật (Internet of Things - IoT), hệ thống thực tế ảo (virtual reality - VR), hệ thống thực tế tăng cường (augmented reality - AR), v.v..

Bộ xử lý 104 cũng có thể được gọi là CPU, bộ xử lý ứng dụng, v.v., bởi vì theo các phương án làm ví dụ này, bộ xử lý 104 có thể thực thi, trong số các phần tử phần mềm khác (không được thể hiện nhằm mục đích rõ ràng), hệ điều hành cấp cao ("HLOS") 110. Thuật ngữ "HLOS" như được sử dụng trong sáng chế này được dự định bao gồm theo nghĩa rộng không chỉ HLOS mà còn bao gồm cả bộ quản lý siêu dữ liệu, một HLOS kết hợp với một bộ quản lý siêu dữ liệu, v.v. Một hành động được thực hiện hoặc điều khiển bởi bộ xử lý 104 dưới sự điều khiển của, hoặc được tạo cấu hình bởi, việc thực thi của HLOS 110 có thể, để mô tả ngắn gọn, được gọi trong sáng chế này là hành động được thực hiện bởi HLOS 110. Tương tự, một hành động được thực hiện hoặc điều khiển bởi phần cứng xử lý khác dưới sự điều khiển của, hoặc được tạo cấu hình bởi, việc thực thi của bất kỳ thực thể phần mềm nào khác có thể, để mô tả ngắn gọn, được gọi trong sáng chế này là hành động được thực hiện bởi thực thể phần mềm khác đó. Ngoài ra, mặc dù nhằm mục đích rõ ràng, HLOS 110 được thể hiện trên Fig.1 tách riêng với các bus chính 106 khác, nhưng cần lưu ý rằng HLOS 110 cũng là bus chính (và vì vậy cũng là một loại AD).

SoC 102 có thể bao gồm tài nguyên hệ thống 112. Tài nguyên hệ thống 112 bao gồm các thành phần và các phần của PCD 100 mà bus chính, chẳng hạn như HLOS 110, bus chính 106 khác, v.v., có thể gửi các giao dịch bus tới đó. Nói cách khác, "giao dịch bus", như thuật ngữ được sử dụng trong sáng chế này, liên quan đến yêu cầu về tài nguyên hệ thống 112, chẳng hạn như yêu cầu đọc hoặc yêu cầu ghi. Theo đó, các tài nguyên hệ thống 112 có thể được nhận dạng bởi các địa chỉ hoặc dải địa chỉ trong không gian địa chỉ hệ thống. Mặc dù nhằm mục đích rõ ràng trên Fig.1 tài nguyên hệ thống 112

được thể hiện là nằm trong SoC 102, nhưng theo các phương án khác, bất kỳ hoặc tất cả các tài nguyên hệ thống như vậy có thể nằm bên ngoài SoC. Tài nguyên hệ thống 112 có thể bao gồm, ví dụ, bộ nhớ truy cập ngẫu nhiên động (dynamic random-access memory - DRAM) như DRAM tốc độ dữ liệu kép (double data-rate DRAM - DDR-DRAM). DDR-DRAM như vậy (hoặc gọi ngắn gọn là "DDR") có thể cung cấp bộ nhớ hệ thống chính của PCD 100 và CPU hoặc bộ xử lý 104 khác có thể thực thi các ứng dụng (tức là, phần mềm) trong DDR như vậy. Tài nguyên hệ thống 112 có thể bao gồm các loại bộ nhớ khác như, ví dụ, bộ nhớ flash, RAM tĩnh (static RAM - SRAM), v.v. Tài nguyên hệ thống 112 có thể bao gồm các thanh ghi khác nhau, cổng và các thành phần khả lập địa chỉ khác. Lưu ý rằng vì tài nguyên hệ thống 112 là đối tượng của yêu cầu giao dịch từ bus chính, nên tài nguyên hệ thống 112 cũng có thể được gọi là đối tượng của yêu cầu giao dịch từ AD, vì AD là thực thể bao gồm một hoặc nhiều bus chính.

Ví dụ, tài nguyên hệ thống thứ nhất 112A có thể bao gồm vùng bộ nhớ thứ nhất, tài nguyên hệ thống thứ hai 112B có thể bao gồm vùng bộ nhớ thứ hai, v.v.. Mặc dù chỉ có hai tài nguyên hệ thống 112A và 112B được thể hiện nhằm mục đích ví dụ, nhưng số lượng bất kỳ các tài nguyên hệ thống 112 có thể được phân bổ trong không gian địa chỉ hệ thống. Trừ khi được mô tả khác trong sáng chế, tài nguyên hệ thống 112 có thể được yêu cầu, phân bổ và hủy phân bổ theo các nguyên lý điện toán thông thường được biết rõ bởi người có hiểu biết trung bình trong lĩnh vực kỹ thuật. Ví dụ, để đáp lại các điều kiện hoạt động nhất định, bus chính 106 có thể khởi tạo yêu cầu về tài nguyên hệ thống 112. Sáng chế liên quan đến hệ thống và phương pháp điều khiển truy cập vào tài nguyên hệ thống 112 như vậy, được chỉ ra về mặt khái niệm bằng đường mũi tên nét đứt trên Fig.1.

Công cụ quản lý độ tin cậy ("TME") 116 là thành phần có liên quan đến hệ thống và phương pháp điều khiển truy cập. Để tăng cường bảo mật, TME 116 có thể chủ yếu được thực hiện trong phần cứng tách biệt với bộ xử lý 104 mà HLOS 110 thực thi trên đó, như một bộ xử lý khác, bộ nhớ chỉ đọc (read-only memory - "ROM"), v.v. (không được thể hiện riêng trên Fig.1). TME 116 có thể sử dụng một thành phần xác thực (không được thể hiện riêng) sử dụng các phương pháp mã hóa để xác thực hình ảnh phần mềm. Hệ thống xác thực như vậy có thể thuộc loại thông thường và do đó không được mô tả trong sáng chế này. Mặc dù nhằm mục đích rõ ràng, TME 116 được thể hiện trên Fig.1 tách riêng với các bus chính 106 khác và HLOS 110 nhưng cần lưu ý rằng TME 116 cũng là bus chính (và do đó cũng là một loại AD).

Khối bảo vệ 118 có thể được tạo cấu hình để bảo vệ có chọn lọc các tài nguyên hệ thống 112. Có thể có các loại khối bảo vệ khác nhau, mỗi khối kết hợp với một thành phần có khả năng phục vụ như hoặc cung cấp tài nguyên. Thuật ngữ "xPU" có thể được sử dụng trong sáng chế này để đề cập theo nghĩa rộng đến loại bất kỳ trong số các loại ("x") khối bảo vệ kết hợp với các loại thành phần tương ứng. Ví dụ, xPU kết hợp với bộ nhớ có thể được gọi là khối bảo vệ bộ nhớ.

Như minh họa trên Fig.2, hệ thống 200 có thể bao gồm hai hoặc nhiều bus chính 202A, 202B, 202C, v.v. Mặc dù một phần của mỗi trong số các bus chính 202A, 202B, 202C, v.v., có thể bao gồm phần cứng, nhưng một hoặc nhiều trong số các bus chính 202A, 202B, 202C, v.v., cũng có thể bao gồm một phần phần mềm. Ví dụ, bus chính thứ nhất 202A có thể bao gồm CPU 204A, mà có thể thực thi HLOS 206A. CPU 204A có thể là ví dụ về bộ xử lý 104 được mô tả ở trên (Fig.1). Ví dụ, bus chính thứ hai 202B có thể bao gồm công cụ phần cứng 204B và một phần phần mềm mà có thể được gọi trong sáng chế này là thực thể thông minh và đáng tin cậy (intelligent and trusted entity - "ITE") 206B. Tương tự, bus chính thứ ba 202C có thể bao gồm công cụ phần cứng 204C và một phần phần mềm 206C. Bus chính thứ hai và thứ ba 202B và 202C có thể cùng nhau tạo thành miền truy cập hoặc AD 208B, như AD 108 được mô tả ở trên liên quan đến Fig.1. Trong AD có hai hoặc nhiều bus chính, phần phần mềm của ít nhất một trong các bus chính là ITE; phần phần mềm của các bus chính khác không cần phải là ITE. Ví dụ, AD 208B có thể là một modem, trong đó bus chính 202B và 202C lần lượt là các lõi modem vectơ và vô hướng. Mặc dù theo phương án ví dụ này, modem bao gồm hai bus chính 202B và 202C, nhưng theo các phương án khác, modem có thể, giống như các AD khác, chỉ bao gồm một bus chính. Lưu ý rằng bus chính thứ nhất 202A cũng tạo thành AD 208A. Vì AD 208A được đặc trưng bởi HLOS 206A khi thực thi, nên AD như vậy có thể được gọi cho thuận tiện trong sáng chế này là HLOS, tương tự với HLOS 110 trên Fig.1.

Các bus chính 202A, 202B, 202C, v.v., có thể được tạo cấu hình để truy cập tài nguyên thông qua các khối quản lý bộ nhớ tương ứng (memory management unit - "xMMUs") 210A, 210B, 210C, .v.v. Thuật ngữ "xMMU" có thể được sử dụng trong sáng chế để đề cập theo nghĩa rộng đến MMU thuộc loại bất kỳ trong số các loại ("x") MMU kết hợp với các loại thành phần tương ứng, chẳng hạn như bộ nhớ hệ thống (ngược lại với cấp độ hoặc loại bộ nhớ khác). Mặc dù nhằm mục đích rõ ràng trên Fig.2

mỗi trong số các bus chính 202A, 202B và 202C được thể hiện là ghép nối với chính xác một xMMU 210A, 210B và 210C tương ứng nhưng trong các ví dụ khác (không được thể hiện) có thể có các phần tử bổ sung mà thông qua đó bus chính có thể truy cập tài nguyên, chẳng hạn như hai hoặc nhiều tầng xMMU.

Mỗi trong số một số bus phụ 212, chẳng hạn như các bus phụ 212A và 212B, có thể phục vụ như hoặc cung cấp các tài nguyên hệ thống 112 tương ứng (Fig. 1), chẳng hạn như bộ nhớ, thanh ghi, cổng, v.v. Mặc dù hai bus phụ 212A và 212B được thể hiện, nhưng hệ thống 200 có thể bao gồm số lượng bus phụ 212 bất kỳ. Mỗi trong số các bus phụ 212A, 212B, v.v., có thể được ghép nối với kết nối hệ thống (ví dụ, hệ thống bus) 216. Các xMMU 210A, 210B, 210C, v.v. được mô tả trên đây cũng được ghép nối với kết nối hệ thống 216. Do đó, mỗi AD 208A, 208B, v.v., có thể truy cập tài nguyên bằng cách tham gia vào giao dịch bus thông qua kết nối hệ thống 216.

Mỗi trong số các bus phụ 212 mà có thể cung cấp tài nguyên hệ thống 112 (Fig. 1) được bảo vệ bởi xPU 214 tương ứng, chẳng hạn các xPU 214A và 214 B. xPU 214 có thể là ví dụ về khối bảo vệ 118 được mô tả trên đây cùng với Fig.1. xPU 214 có thể được tạo cấu hình để bảo vệ hoặc phòng chống truy cập vào tài nguyên hệ thống 112 được cung cấp bởi bus phụ 212 theo cách được mô tả dưới đây.

Như minh họa trên Fig.3, bản đồ bộ nhớ hệ thống 300 thể hiện các vùng trong bộ nhớ hệ thống, có thể được gọi trong sáng chế này là các nhóm tài nguyên (resource group - "RG") 302. Vùng trong bộ nhớ hệ thống (ví dụ, DDR) là ví dụ về tài nguyên hệ thống 112 (Fig. 1). Lưu ý rằng các thuật ngữ "tài nguyên", "tài nguyên hệ thống" và "nhóm tài nguyên" (hoặc "RG") có thể được sử dụng về cơ bản đồng nghĩa trong sáng chế, với thuật ngữ "nhóm tài nguyên" (hoặc "RG") được sử dụng trong một số trường hợp để đề cập cụ thể hơn đến tài nguyên đã được phân bổ cho AD. Như được mô tả dưới đây chi tiết hơn, RG có thể được phân bổ cho AD bằng cách thiết lập các quyền truy cập cho phép AD truy cập (tức là, hoàn thành một trao đổi đọc hoặc ghi với) RG. Thuật ngữ "sở hữu bởi" (AD) cũng có thể được sử dụng trong sáng chế để chỉ RG được phân bổ cho AD. Tài nguyên được sở hữu bởi hoặc được phân bổ cho AD cũng có thể được gọi là thuộc về, là một phần của, v.v., AD. Vì thuật ngữ được sử dụng trong sáng chế này, "để phân bổ" RG cho AD có nghĩa không chỉ là phân bổ RG theo nghĩa thông thường (ví dụ, bằng cách xác định hoặc thiết lập riêng không gian bộ nhớ) mà còn khiến cho AD trở

thành chủ sở hữu của RG. RG có thể chỉ truy cập được bởi AD sở hữu RG đó. Nỗ lực truy cập RG của bất kỳ thực thể nào không phải là chủ sở hữu RG đó bị ngăn chặn hoặc bị chặn bởi xPU liên quan. Theo các phương án làm ví dụ được mô tả ở đây, trạng thái mặc định hay ban đầu (ví dụ, sau khi khởi động PCD 100) có thể là trạng thái mà HLOS 110 (Fig.1) sở hữu tất cả các tài nguyên hệ thống 112; các AD khác, bao gồm cả TME 116 đều không thể truy cập vào bất kỳ tài nguyên nào cho đến khi các tài nguyên có thể được phân bổ cho các AD khác đó.

Trên Fig.3 bản đồ bộ nhớ hệ thống 300 minh họa ví dụ trong đó các RG khác nhau 302A, 302B, v.v., đến 302N (được gọi chung là các RG 302) đã được phân bổ cho các AD. Phần còn lại của bản đồ bộ nhớ hệ thống 300 thể hiện các tài nguyên hệ thống mà chưa được phân bổ rõ ràng cho các AD và do đó, theo phương án ví dụ được mô tả ở đây, vẫn thuộc sở hữu của HLOS. Các RG 302 có thể được định vị ở bất kỳ đâu trong dải địa chỉ bộ nhớ và các vị trí được thể hiện trên Fig.3 được dự định chỉ làm các ví dụ.

Như minh họa trên Fig.11, giao diện chương trình ứng dụng (application program interface - "API") xPU 1100, được thể hiện ở dạng khái niệm, là ví dụ về các cài đặt cấu hình khác nhau mà AD có thể cung cấp cho xPU để tạo cấu hình hoặc điều khiển cách xPU hoạt động. API xPU 1100 có thể có hai phần: phần cấu hình trên mỗi RG 1102 mà qua đó AD có thể cung cấp cho xPU các thiết lập cấu hình cho (tức là, cụ thể cho) mỗi RG và phần cấu hình tổng thể 1104 mà qua đó các khía cạnh của hoạt động xPU không riêng cho bất kỳ RG nào có thể được tạo cấu hình.

Phần cấu hình trên mỗi RG 1102 cung cấp cho hoặc lập trình xPU với một cấu hình RG 1106 mà tạo cấu hình xPU cho một trong số các RG (ví dụ, "RG_0" đến "RG_M", trong đó M là số cố định xác định số RG được hỗ trợ tối đa (M + 1)). Mỗi cấu hình RG 1106 có thể bao gồm dải địa chỉ 1108, các quyền truy cập bao gồm quyền đọc 1110 và quyền ghi 1112, và thiết lập cho phép RG 1114. Theo phương án làm ví dụ này, xPU chỉ cho phép HLOS thiết lập cấu hình RG 1106, và sự cho phép đó phụ thuộc vào tính năng khóa được mô tả dưới đây. xPU trong phương án này ngăn chặn mọi nỗ lực từ bất kỳ thực thể nào không phải HLOS thiết lập cấu hình RG 1106. Ví dụ, xPU chặn bất kỳ nỗ lực nào của một AD khác, TME, v.v., thiết lập dải địa chỉ 1108, quyền đọc 1110, quyền ghi 1112 hoặc thiết lập cho phép RG 1114. Theo các phương án khác (không được mô tả ở đây), tính năng có thể được cung cấp để cho phép các AD khác thiết lập cấu hình

RG. Ví dụ, mặc dù theo phương án làm ví dụ được mô tả ở đây, HLOS mặc định sở hữu tất cả các tài nguyên hệ thống chưa phân bổ được bảo vệ bởi xPU, nhưng theo các phương án khác, cài đặt chế độ có thể được cung cấp trong phần cấu hình tổng thể 1104 mà qua đó HLOS có thể chọn xem quyền sở hữu đó xảy ra theo mặc định hay phải xảy ra bằng cách phân bổ rõ ràng các RG cho HLOS.

Sử dụng dải địa chỉ 1108 cung cấp bởi HLOS, xPU có thể nhận dạng hoặc xác định RG mà xPU sẽ bảo vệ.Ví dụ, HLOS có thể xác định một trong số các RG 302 được mô tả trên đây liên quan đến Fig.3 bằng cách cung cấp địa chỉ bắt đầu và địa chỉ kết thúc mà RG 302 trải trên.

Thiết lập cho phép RG 1114 có thể là một bit mà HLOS có thể cung cấp để cho phép xPU bắt đầu bảo vệ RG được xác định bởi dải địa chỉ 1106. Ví dụ, thiết lập cho phép RG 1114 có giá trị là "1" có thể cho phép xPU áp dụng các quyền 1110 và 1112 như được mô tả dưới đây, trong khi thiết lập cho phép RG 1114 có giá trị là "0" có thể chỉ báo cho xPU biết rằng RG đã bị vô hiệu hóa, tức là, các quyền 1110 và 1112 không (hoặc không còn) áp dụng được. Ngoài ra, HLOS có thể, trên thực tế, trả lại RG về vùng tài nguyên có sẵn bằng cách thiết lập thiết lập cho phép RG 1114 để chỉ báo rằng các quyền 1110 và 1112 không còn áp dụng được nữa. Theo phương án làm ví dụ được mô tả ở đây, RG phải được trả lại về vùng tài nguyên sẵn trước khi HLOS có thể phân bổ RG đó cho một AD khác.

Quyền đọc 1110 và quyền ghi 1112 có thể được cung cấp lần lượt dưới hình thức một bit cho phép đọc (read-enable - RD_EN) trên mỗi AD và một bit cho phép ghi (write-enable - WR_EN) trên mỗi AD. Ví dụ, quyền đọc 1110 đối với RG có thể bao gồm các bit "AD_0:RD_EN" đến "AD_N:RD_EN" cho các AD tương ứng (trong đó N là một số cố định xác định số lượng hỗ trợ tối đa (N + 1) AD. Tương tự như vậy, quyền ghi 1112 có thể bao gồm các bit "AD_0: WR_EN" đến "AD_N: WR_EN" cho các AD tương ứng. Ví dụ, bit cho phép đọc có giá trị là "1" có thể tạo cấu hình xPU để cho phép (hoặc không ngăn chặn) hoàn thành giao dịch đọc khởi tạo bởi AD tương ứng hướng đến RG, và bit cho phép đọc có giá trị là "0" có thể tạo cấu hình xPU để ngăn chặn hoàn thành giao dịch đọc khởi tạo bởi AD tương ứng hướng đến RG. Tương tự như vậy, bit cho phép ghi có giá trị là "1" có thể tạo cấu hình xPU để cho phép hoàn thành giao dịch ghi khởi tạo bởi AD tương ứng hướng đến RG, và bit cho phép ghi có giá trị là "0" có thể

tạo cấu hình xPU để ngăn chặn việc hoàn thành một giao dịch ghi khởi tạo bởi AD tương ứng hướng đến RG.

AD sở hữu hoặc đã được phân bổ RG nếu ít nhất một bit trong quyền đọc 1110 hoặc ít nhất một bit trong quyền ghi 1112 chỉ báo rằng AD được phép truy cập RG. Lưu ý rằng thông qua quyền đọc 1110 và quyền ghi 1112, AD có thể được cung cấp cả quyền đọc và quyền ghi, quyền chỉ đọc v.v, do đó điều khiển kiểu truy cập mà AD có thể có vào RG cụ thể.

Mỗi yêu cầu giao dịch mà AD có thể tạo ra có thể bao gồm ký hiệu nhận dạng AD, tức là giá trị nhận dạng duy nhất AD đó. Các bit quyền đọc và quyền ghi 1110 và 1112 có thể được lập chỉ mục bởi các ký hiệu nhận dạng AD. Khi xPU nhận yêu cầu giao dịch, thì xPU có thể so sánh ký hiệu nhận dạng AD có trong yêu cầu giao dịch với các giá trị bit tương ứng trong quyền đọc 1110 và quyền ghi 1112, và sau đó dựa trên so sánh này cho phép hoặc ngăn chặn việc hoàn thành giao dịch được yêu cầu.

Phần cấu hình trên mỗi RG 1102 còn cung cấp cho xPU các nhóm bit khóa cấu hình RG 1116. Mỗi nhóm bit khóa cấu hình RG 1116 tương ứng với một trong các cấu hình RG 1106 được mô tả ở trên. Mỗi nhóm bit khóa cấu hình RG 1116 có thể bao gồm một số bit khóa 1118 bằng với số lượng AD. Nghĩa là, mỗi AD tương ứng với một bit khóa 1118. Ví dụ, bit khóa 1118 thứ nhất (“AD_0: LOCK”) tương ứng với AD thứ nhất, và bit khóa 1118 thứ hai (“AD_1: LOCK”) tương ứng với AD thứ hai v.v., cho đến bit khóa thứ N (“AD_N: LOCK”) tương ứng với AD thứ N. Vì HLOS và TME là các loại AD, nên một trong các bit khóa 1118, chẳng hạn như, ví dụ, bit khóa 1118 thứ nhất, có thể tương ứng với HLOS và một bit khóa khác trong số các bit khóa 1118, chẳng hạn như, bit khóa 1118 thứ hai, có thể tương ứng với TME.

xPU cho phép bit khóa 1118 được thiết lập chỉ bởi AD tương ứng. Có nghĩa là, xPU khóa hoặc bỏ qua bất kỳ nỗ lực thiết lập bit khóa 1118 của bất kỳ thực thể nào không phải AD tương ứng với bit khóa 1118 đó. Mỗi bit khóa 1118 có thể có giá trị chỉ báo trạng thái "đã khóa" hoặc trạng thái "đã mở khóa". Thuật ngữ “đã khóa “ hay “đã mở khóa" chỉ cấu hình RG 1106; thiết lập một hoặc nhiều bit khóa 1118 trong nhóm bit khóa cấu hình RG 1116 bằng một giá trị chỉ báo trạng thái “đã khóa” tạo cấu hình xPU để ngăn chặn bất kỳ thực thể nào sửa đổi cấu hình RG 1106 tương ứng. Tức là, cấu hình RG 1106 bị khóa hoặc không thể sửa đổi được khi một hoặc nhiều bit khóa 1118 bất kỳ trong

nhóm 1116 tương ứng với cấu hình RG 1106 đó được thiết lập. Bằng cách thiết lập bit khóa 1118 tương ứng của nó, AD có thể thiết lập sự tin cậy trong cấu hình RG 1106 bằng cách ngăn chặn bất kỳ AD khác nào, bao gồm HLOS, TME, v.v. sau đó sửa đổi cấu hình RG 1106.

xPU cho phép bit khóa 1118 trong nhóm 1116 được thiết lập bởi AD tương ứng chỉ nếu AD đó sở hữu RG tương ứng với nhóm 1116 đó. Nếu AD nỗ lực khóa cấu hình RG 1106 trong đó AD không có quyền truy cập RG, thì xPU ngăn chặn việc hoàn thành nỗ lực đó.

SoC 102 (Fig.1) có thể thực hiện thiết lập lại dựa trên phần cứng hoặc khởi động nguội để trở nên sẵn sàng với hoạt động bình thường hoặc "chế độ nhiệm vụ", mà có thể bao gồm cung cấp điều khiển truy cập tài nguyên theo cách mô tả dưới đây. Việc khởi động nguội như vậy có thể bao gồm các thực thể mà không liên quan trực tiếp đến sáng chế, như một hoặc nhiều bộ nạp khởi động. Do đó, các chi tiết như vậy không được mô tả ở đây. Tuy nhiên, có thể hữu ích khi hiểu rằng một hoặc nhiều bộ nạp khởi động (có thể được lưu trữ trong ROM) được thực hiện trên CPU 104 để tải HLOS 110 vào bộ nhớ hệ thống (tức là, trong tài nguyên hệ thống 112) và khởi động HLOS 110. Bộ nạp khởi động có thể tải HLOS 110 vào bộ nhớ hệ thống bằng cách, chẳng hạn, sao chép HLOS 110 (hình ảnh phần mềm) từ bộ nhớ flash hoặc bộ nhớ không khả biến khác. Kết quả là, HLOS 110 bắt đầu thực thi. Trong một số trường hợp, bộ nạp khởi động có thể tải môi trường thực thi an toàn (secure execution environment - SEE) hay phần mềm trung gian khác, lần lượt tải và khởi động HLOS 110. Như được hiểu bởi người có hiểu biết trung bình trong lĩnh vực kỹ thuật này, SEE (đôi khi được gọi là môi trường thực thi đáng tin cậy hoặc "TEE" (trusted execution environment)) bao gồm phần mềm mà, thường cùng với phần cứng bộ xử lý an toàn, cách ly các ứng dụng đáng tin cậy chạy trong SEE với các ứng dụng không đáng tin cậy chạy trên HLOS chính (không đáng tin cậy). Tuy nhiên, trong ngữ cảnh của sáng chế, SEE được đề cập chỉ như một ví dụ về phần mềm trung gian mà có thể được tải trước khi HLOS được tải, và các phương án không cần bao gồm SEE hoặc phần mềm trung gian khác. Sau đó, như được mô tả dưới đây, đến lượt HLOS 110 có thể tải và khởi động một hay nhiều AD 108 khác. TME 116 có thể khởi động đồng thời với HLOS 110. Cần lưu ý rằng TME 116 (và bất kỳ SEE nào) có thể có quyền sở hữu các tài nguyên theo cách về cơ bản giống như cách được mô tả dưới đây

đối với bất kỳ AD 108 khác nào. Theo cách này, HLOS 110 có thể thiết lập sự tin cậy trong TME 116 (và SEE bất kỳ).

Như minh họa trên Fig.4, phương pháp 400 để điều khiển truy cập tài nguyên trong SoC có thể bao gồm khởi động AD, như, ví dụ, AD bất kỳ trong số các AD 108 được mô tả trên đây (Fig.1) hoặc 208B (Fig.2). Như được chỉ báo bởi khối 402, HLOS 110 hoặc thành phần khác có thể phân bổ tài nguyên cho AD. Như được mô tả trên đây, tài nguyên có thể là, ví dụ, vùng bộ nhớ mà ở đó AD sẽ thực thi. Theo phương án làm ví dụ, phân bổ tài nguyên theo khối 402 có thể bao gồm tạo cấu hình các quyền truy cập trong một xPU bảo vệ tài nguyên để cho phép AD truy cập tài nguyên. Ví dụ, các quyền truy cập có thể được tạo cấu hình để trao cho AD quyền đọc, quyền ghi hoặc cả quyền đọc và quyền ghi. Việc phân bổ tài nguyên có thể còn bao gồm tạo cấu hình quyền truy cập sao cho TME 116 (Fig.1) có đủ quyền truy cập (ví dụ, chỉ đọc) để thực hiện việc xác thực được mô tả dưới đây. Cùng với ngoại lệ này dành cho TME 116, các quyền truy cập được tạo cấu hình để ngăn chặn truy cập vào tài nguyên bởi bất kỳ AD nào không phải AD đang được khởi động.

Mặc dù theo các phương án làm ví dụ được mô tả ở đây, thành phần thực hiện hoặc điều khiển việc phân bổ các tài nguyên là HLOS 110, nhưng theo các phương án khác, thành phần không phải HLOS có thể thực hiện hoặc điều khiển việc phân bổ tài nguyên hoặc một phần của việc phân bổ đó. Ví dụ, theo các phương án khác, HLOS có thể thực hiện phân bổ tài nguyên theo một số khía cạnh được mô tả ở đây, chẳng hạn như phân bổ các dải địa chỉ, trong khi một thành phần khác tạo cấu hình các quyền truy cập.

Như được chỉ ra bởi khối 404, HLOS 110 hoặc thành phần khác có thể tải hình ảnh phần mềm liên quan đến AD vào vùng bộ nhớ (tài nguyên). Hình ảnh phần mềm có thể là ITE và có thể thu được từ ROM hoặc một nguồn khác.

Như được chỉ ra bởi khối 406, TME 116 (Fig.1) sau đó có thể khóa tài nguyên chống lại truy cập của bất kỳ thực thể nào không phải AD mà đang được khởi động, ngoại trừ TME 116 có thể có đủ quyền truy cập để thực hiện việc xác thực được mô tả dưới đây. Theo một phương án làm ví dụ, khóa tài nguyên chống lại truy cập theo khối 406 có thể bao gồm khóa cấu hình tài nguyên bằng cách thiết lập một bit khóa trong xPU bảo vệ tài nguyên. Vì quyền truy cập có thể được tạo cấu hình bởi HLOS 110 hoặc thành phần khác như được mô tả ở trên (khối 402), nên việc khóa cấu hình tài nguyên sẽ khóa

bản thân tài nguyên chống lại truy cập theo các quyền truy cập đó. Lưu ý rằng khóa cấu hình tài nguyên theo cách này là khóa quyền truy cập chống lại sửa đổi của HLOS 110 (Fig.1).

Như được chỉ báo bởi khối 408, TME 116 có thể xác thực hình ảnh phần mềm liên quan đến AD mà đã được tải vào bộ nhớ và được khóa. Như được chỉ ra bởi khối 410 sau khi xác thực thành công hình ảnh phần mềm đó, TME 116 có thể khởi tạo (hoặc nếu quá trình khởi động đã bắt đầu, thì không tiến hành ngăn chặn sự tiếp tục của) khởi động AD. Như được chỉ ra bởi khối 412, AD khởi động sau đó có thể khóa tài nguyên chống lại truy cập của bất kỳ thực thể nào ngoại trừ chính nó. Do đó, tài nguyên có thể bị khóa để chống lại truy cập của TME 116 và HLOS 110. Theo phương pháp làm ví dụ được mô tả dưới đây liên quan đến Fig.5, AD khóa tài nguyên có thể được điều chỉnh sau khi AD hợp lệ hóa thành công tài nguyên và sau khi loại bỏ quyền truy cập tài nguyên của TME.

Trong các ví dụ mà AD bao gồm hai hoặc nhiều bus chính, mỗi bus có một phần phần mềm, các hoạt động được mô tả ở trên liên quan đến các khối 402-412 có thể được thực hiện dựa trên hình ảnh phần mềm là ITE, chẳng hạn như ITE 206B mô tả trên đây (Fig.2). Trong các ví dụ như vậy, việc khởi động AD có thể dẫn đến việc ITE 206B bắt đầu hoạt động hoặc thực thi trên công cụ phần cứng liên quan của nó (trong đó ITE 206B và công cụ phần cứng liên quan 204B cùng nhau hoạt động như bus chính thứ nhất 202B) và đến lượt ITE bắt đầu khởi động bus chính thứ hai 202C bằng cách thực hiện các thao tác được mô tả ở trên liên quan đến các khối 402-412 trên hình ảnh phần mềm thứ hai. Trong ví dụ mà AD bao gồm ba bus chính (không được thể hiện), bus chính thứ hai có thể khởi động bus chính thứ ba, v.v. Lưu ý rằng bus chính thứ hai và thứ ba trong ví dụ này có thể khởi động độc lập với TME; TME chỉ cần khởi động bus chính thứ nhất.

Như minh họa trên Fig.5, phương pháp 500 để điều khiển truy cập tài nguyên trong SoC bao gồm khởi động AD. Phương pháp 500 có thể là ví dụ về phương pháp 400 được mô tả trên đây (Fig.4).

Như được chỉ ra bởi khối 502, HLOS có thể đọc các yêu cầu khởi động AD, chẳng hạn như yêu cầu bộ nhớ đối với AD (ví dụ, số lượng phân vùng và kích thước của mỗi phân vùng), yêu cầu về xung nhịp/nguồn cho một AD, v.v. Như được chỉ ra bởi khối 504, HLOS có thể phân bổ một phân vùng bộ nhớ cho AD. Phân vùng bộ nhớ thuộc sở

hữu của HLOS tại thời điểm này và nằm liền kề về mặt vật lý. Như được chỉ ra bởi khối 506, HLOS có thể tải (tức là, sao chép) hình ảnh phần mềm AD từ bộ nhớ không khả biến vào phân vùng bộ nhớ AD. HLOS có thể cũng tải bất kỳ bộ nạp khởi động nào liên quan đến AD vào phân vùng bộ nhớ.

AD có thể hoặc có thể không thuộc kiểu có đặc tính quyền riêng tư. "Quyền riêng tư" trong ngữ cảnh này đề cập đến nội dung của bên thứ ba mà chỉ AD mới được đọc. Nội dung này có thể được lưu trữ dưới dạng mã hóa trong bộ nhớ flash, và HLOS có thể tải nội dung được mã hóa vào phân vùng riêng cho nó trong bộ nhớ hệ thống. Vùng này có thể được giải mã bởi bộ nạp khởi động chính của AD sau khi AD được khởi động. Nếu AD có quyền riêng tư, thì như được chỉ ra bởi khối 508, HLOS có thể tạo hai RG: một RG cho bộ nạp khởi động chính của AD và RG còn lại cho phần phân bổ bộ nhớ còn lại của AD. Nếu AD không có quyền riêng tư, thì như được chỉ ra bởi khối 510, HLOS có thể tạo một RG cho toàn bộ phân bổ bộ nhớ của AD.

Như được chỉ báo bởi khối 512, HLOS sau đó có thể tạo cấu hình cho thanh ghi vectơ khởi động của AD. HLOS có thể cũng tạo cấu hình (các) xPU bảo vệ không gian thanh ghi của AD. Ví dụ, HLOS có thể sử dụng API xPU 1100 (Fig.11) để tạo cấu hình (các) xPU cùng với cấu hình RG 1106. Cấu hình RG có thể bao gồm các quyền truy cập cho AD để truy cập đọc và ghi vào RG. Cấu hình RG có thể cũng bao gồm các quyền truy cập cho TME để truy cập đọc vào RG sao cho TME có thể thực hiện xác thực như mô tả dưới đây. Như chỉ ra bởi khối 514, HLOS sau đó có thể gửi yêu cầu khởi động AD đến TME. Như mô tả trên đây, việc khởi động AD bao gồm khởi động tất cả các bus chính trong AD.

Như được chỉ ra bởi khối 516, TME khóa một hoặc nhiều cấu hình RG của AD bằng cách thiết lập bit khóa của nó trong các xPU tương ứng. Trong ngữ cảnh của tính năng khóa, "thiết lập" một bit khóa có nghĩa là thiết lập cấu hình RG ở trạng thái "đã khóa" và "xóa" một bit khóa có nghĩa là thiết lập cấu hình RG ở trạng thái "đã mở khóa". TME có thể đọc thông tin phân bổ bộ nhớ và xác nhận rằng các cấu hình xPU là đúng. Lưu ý rằng việc thiết lập bit khóa của TME ngăn chặn bất kể thực thể nào, bao gồm cả HLOS, TME v.v. sửa đổi cấu hình của RG.

Như được chỉ ra bởi khối 518, TME sau đó có thể xác thực hình ảnh phần mềm, chẳng hạn, ví dụ, bằng cách xác thực chữ kí của hình ảnh phần mềm. Việc xác thực có

thể được thực hiện theo cách thông thường bằng cách sử dụng các kỹ thuật mã hóa (ví dụ, các khóa), như được hiểu bởi người có hiểu biết trung bình trong lĩnh vực kỹ thuật này. Nếu TME xác nhận rằng việc xác thực đã thành công, thì TME có thể bắt đầu khởi động AD như được chỉ báo bởi khối 520. Điều này có thể bao gồm việc TME cho phép các tài nguyên khởi động và giải phóng AD khỏi việc thiết lập lại, sao cho AD được cho phép hoặc được phép khởi động. Như mô tả trên đây, các AD nói chung bao gồm các hệ thống con với bộ xử lý chuyên dụng và các miền xung nhịp/nguồn. "Giải phóng AD khỏi việc thiết lập lại" có nghĩa là thực hiện tất cả các cấu hình phần cứng cần thiết để đưa hệ thống con và bộ xử lý AD ra khỏi trạng thái thiết lập lại. Tuy các phần phần cứng của AD trở nên hoạt động theo cách nêu trên, nhưng hình ảnh phần mềm AD hoặc phần phần mềm của AD bắt đầu khởi động hoặc thực thi trong vùng hoặc phân vùng bộ nhớ mà nó được tải vào.

Như được chỉ ra bởi khối 522, ngay khi AD bắt đầu thực thi, nó có thể xác thực tính hợp lệ một hoặc nhiều RG của nó bằng cách kiểm tra cấu hình RG trong xPU. AD luôn luôn được phép kiểm tra (tức là, đọc) các cấu hình RG như cấu hình RG 1106 mô tả trên đây (Fig.11) bao gồm các quyền truy cập RG. Nếu việc xác nhận cấu hình RG thành công thì AD có thể thiết lập bit khóa tương ứng của nó. Việc xác thực tính hợp lệ là thành công nếu AD xác nhận rằng các quyền truy cập RG phù hợp với kỳ vọng của AD. Ví dụ, AD thường có thể kỳ vọng các quyền truy cập RG chỉ ra rằng RG chỉ thuộc sở hữu của AD đó, tức là không thực thể nào không phải AD đó được phép truy cập RG đó. Nếu các quyền truy cập RG cho phép thực thể không được kỳ vọng truy cập vào RG thì AD từ chối cấu hình RG do không hợp lệ (tức là, xác nhận tính hợp lệ không thành công) và dừng lại việc khởi động. Vì AD trong ví dụ này vừa mới bắt đầu khởi động (tức là, thực thi), nên AD có thể kỳ vọng TME vẫn có quyền truy cập đọc (nhưng không có quyền truy cập ghi) vào RG.

Khi xác thực tính hợp lệ cấu hình RG thành công và sau khi AD đã thiết lập bit khóa tương ứng của nó, AD có thể thông báo cho TME. Như được chỉ ra bởi khối 523, TME có thể xóa bit khóa của nó sau khi nhận được thông báo này từ AD. Việc xóa bit khóa của TME đối với RG có thể kích hoạt xPU cũng tự động (tức là, không có sự can thiệp của HLOS hoặc thực thể khác) loại bỏ các quyền truy cập của TME vào RG đó, vì việc xóa bit khóa của TME chỉ báo TME không còn yêu cầu quyền truy cập vào RG nữa. Khi bit khóa của AD vẫn còn được thiết lập, thì cấu hình RG bị khóa chống lại truy cập

của TME, HLOS và bất kỳ AD nào khác không sở hữu RG này. Như được chỉ ra bởi khối 524, AD sau đó có thể tiếp tục khởi động.

Nếu TME xác định (khối 518) rằng xác thực không thành công, TME có thể mở khóa một hoặc nhiều cấu hình RG liên quan đến RG chứa hình ảnh phần mềm của AD và báo hiệu cho HLOS rằng hoạt động khởi động không thành công, như được chỉ ra bởi khối 526. TME cũng có thể điều khiển tín hiệu thiết lập lại phần cứng (không được thể hiện trên hình vẽ) được áp dụng cho công cụ phần cứng của AD, và có thể chỉ giải phóng tín hiệu thiết lập lại nếu việc xác thực thành công.

Như minh họa trên Fig.6, phương pháp 600 để điều khiển truy cập tài nguyên trong SoC có thể bao gồm bước phân bổ tài nguyên bổ sung cho AD, chẳng hạn, ví dụ, AD bất kỳ trong số các AD 108 (Fig.1) hay 208B (Fig.2) được mô tả trên đây. Như được chỉ báo bởi khối 602, AD có thể truyền yêu cầu về một hoặc nhiều tài nguyên bổ sung đến HLOS. "Bổ sung" trong ngữ cảnh này có nghĩa là bổ sung vào không gian bộ nhớ mà hình ảnh phần mềm của AD nằm trong đó và được phân bổ cho AD khi AD được khởi động. Như được chỉ ra bởi khối 604, HLOS sau đó có thể phân bổ tài nguyên hoặc các tài nguyên bổ sung này đến AD yêu cầu. Như được chỉ ra bởi khối 606, AD sau đó có thể, bằng cách thiết lập bit khóa của xPU tương ứng, khóa cấu hình RG liên quan đến tài nguyên bổ sung chống lại truy cập của bất kỳ thực thể nào không phải AD yêu cầu. Lưu ý rằng việc khóa tài nguyên theo cách này sẽ khóa tài nguyên chống lại truy cập của HLOS, TME và bất kỳ thực thể khác nào không phải AD yêu cầu.

Như minh họa trên Fig.7, phương pháp 700 để điều khiển truy cập tài nguyên trong SoC có thể bao gồm bước phân bổ một tài nguyên bổ sung cho AD. Phương pháp 700 có thể là ví dụ về phương pháp 600 (Fig.6) đã mô tả trên đây.

Như chỉ ra ở khối 702, AD có thể truyền yêu cầu về một hoặc nhiều tài nguyên bổ sung đến HLOS. Yêu cầu có thể bao gồm, ví dụ, lượng không gian bộ nhớ được yêu cầu, cũng như các quyền truy cập được yêu cầu. Như được chỉ ra bởi khối 704, HLOS có thể xác định xem tài nguyên yêu cầu có sẵn hay không. Nếu tài nguyên yêu cầu không có sẵn, thì HLOS sẽ thông báo cho AD rằng yêu cầu bị từ chối như chỉ ra bởi khối 706. Nếu yêu cầu tài nguyên có sẵn, thì HLOS có thể phân bổ RG mới như được chỉ báo bởi khối 708. Phân bổ RG mới có thể bao gồm cung cấp cho và lập trình xPU bảo vệ tài nguyên với một cấu hình RG, mà có thể bao gồm dải địa chỉ RG, các quyền truy cập v.v. Ví dụ,

HLOS có thể dùng API xPU 1100 (Fig.11) để tạo cấu hình xPU với cấu hình RG 1106. Sau đó HLOS có thể trả con trỏ tới RG mới được tạo cho AD yêu cầu, như được chỉ ra bởi khối 710.

Như được chỉ ra bởi khối 712, AD yêu cầu có thể khóa RG và kiểm tra cấu hình RG để xác định xem nó "hợp lệ" hay "không hợp lệ". Như đã mô tả ở trên liên quan đến khối 522 (Fig.5), AD có thể coi cấu hình RG là hợp lệ nếu các quyền truy cập RG là theo kỳ vọng của AD. Ví dụ, nếu AD mong muốn không có thực thể nào khác truy cập vào tài nguyên, thì các quyền truy cập RG chỉ nên cho phép AD đó quyền truy cập vào tài nguyên. Nếu các quyền truy cập RG cho phép bất kỳ thực thể nào khác truy cập vào tài nguyên thì AD sẽ từ chối cấu hình RG đó vì không hợp lệ. Nếu cấu hình RG là "không hợp lệ", thì AD có thể loại bỏ khóa và thông báo cho HLOS biết rằng RG đã bị từ chối như chỉ dẫn bởi khối 716. Nếu cấu hình RG là "hợp lệ", thì AD có thể bắt đầu sử dụng RG như chỉ dẫn bởi khối 714.

Như minh họa trên Fig.8, phương pháp 800 để điều khiển truy cập tài nguyên trong SoC có thể bao gồm bước hủy phân bổ tài nguyên bổ sung mà trước đó đã được phân bổ cho AD, như bởi phương pháp 600 (Fig.6) hoặc 700 (Fig.7) mô tả trên đây. Như được chỉ báo bởi khối 802, AD có thể mở khóa tài nguyên. Như được chỉ báo bởi khối 804, AD sau đó có thể thông báo cho HLOS biết rằng tài nguyên hiện đang rỗi.

Như minh họa trên Fig.9, phương pháp 900 để điều khiển truy cập tài nguyên trong SoC có thể bao gồm hủy phân bổ tài nguyên bổ sung mà trước đó đã được phân bổ tới AD. Phương pháp 900 có thể là ví dụ về phương pháp 800 (Fig.8) đã mô tả trên đây.

Như được chỉ ra bởi khối 902, AD có thể xóa không gian bộ nhớ mà trước đó đã được phân bổ cho RG. Xóa không gian bộ nhớ là xóa dữ liệu mà nếu không có thể dễ bị truy cập trái phép và vì vậy cung cấp thêm bảo mật hoặc quyền riêng tư mà có thể được mong muốn theo một số phương án. Như được chỉ báo trong khối 904, AD có thể mở khóa RG. Như được chỉ báo bởi khối 906, AD sau đó có thể thông báo cho HLOS biết rằng RG hiện đang rỗi và có thể được trả lại vùng hay danh sách các RG "không sử dụng". Như được chỉ báo bởi khối 908, HLOS có thể vô hiệu hóa RG và trả nó về lại vùng. "Vô hiệu hóa RG" nghĩa là xPU không còn sử dụng RG này để điều khiển truy cập. HLOS có thể vô hiệu hóa RG bằng cách sử dụng thiết lập cho phép RG 1114 của API xPU 1100 (Fig.11).

Như minh họa trên Fig.10, các phương án làm ví dụ về hệ thống và phương pháp để điều khiển truy cập tài nguyên có thể được thể hiện trong PCD 1000. PCD 1000 bao gồm SoC 1002. SoC 1002 có thể bao gồm CPU 1004, GPU 1006, DSP 1007 và bộ xử lý tín hiệu tương tự 1008 hoặc các bộ xử lý khác. CPU 1004 có thể bao gồm nhiều lõi, như lõi thứ nhất 1004A, lõi thứ hai 1004B v.v đến lõi thứ N 1004N. CPU 1004 hoặc lõi bất kỳ trong số các lõi của nó có thể là ví dụ về bộ xử lý 1004 (Fig.1) hoặc CPU 204A (Fig. 2) mô tả trên đây.

Bộ điều khiển màn hình 1010 và bộ điều khiển màn hình cảm ứng 1012 có thể được ghép nối với CPU 1004. Màn hình cảm ứng 1014 bên ngoài SoC 1002 có thể được ghép nối với bộ điều khiển màn hình 1010 và bộ điều khiển màn hình cảm ứng 1012. PCD 1000 có thể còn bao gồm bộ giải mã viđeo 1016 được ghép nối với CPU 1004. Bộ khuếch đại viđeo 1018 được ghép nối với bộ giải mã viđeo 1016 và màn hình cảm ứng 1014. Cổng viđeo 1020 có thể được ghép nối với bộ khuếch đại viđeo 1018. Bộ điều khiển bus nối tiếp đa năng (universal serial bus - USB) 1022 có thể cũng được ghép nối với CPU 1004, và cổng USB 1024 có thể được ghép nối với bộ điều kiển USB 1022. Thẻ mođun nhận dạng thuê bao (subscriber identity module - SIM) 1026 cũng có thể được ghép nối với CPU 1004.

Một hoặc nhiều bộ nhớ có thể được ghép nối với CPU 1004. Một hoặc nhiều bộ nhớ bao gồm cả bộ nhớ khả biến và bộ nhớ không khả biến. Ví dụ về bộ nhớ khả biến bao gồm bộ nhớ truy cập ngẫu nhiên tĩnh (static random access memory - SRAM) 1028 và các RAM động (dynamic RAM - DRAM) 1030 và 1031. Các DRAM 1030 và 1031 hoặc các phần của chúng có thể là các ví dụ về bộ nhớ hệ thống 112 (Fig.1) được mô tả trên đây. Các bộ nhớ như vậy có thể ở bên ngoài SoC 1002 như DRAM 1030 hoặc ở bên trong SoC 1002 như DRAM 1031. Bộ điều khiển DRAM 1032 được ghép nối với CPU 1004 có thể điều khiển việc ghi dữ liệu vào, và đọc dữ liệu từ, các DRAM 1030 và 1031. Theo các phương án khác, bộ điều khiển DRAM như vậy có thể được bao gồm trong bộ xử lý, chẳng hạn như CPU 1004. CPU 1004 có thể thực thi HLOS hoặc phần mềm khác mà được lưu trữ ở bộ nhớ bất kỳ trong số các bộ nhớ nêu trên.

CODEC âm thanh lập thể 1034 có thể được ghép nối với bộ xử lý tín hiệu tương tự 1008. Ngoài ra, bộ khuếch đại âm thanh 1036 có thể được ghép nối với CODEC âm thanh lập thể 1034. Các loa lập thể thứ nhất và thứ hai 1038 và 1040 lần lượt có thể được

ghép nối với bộ khuếch đại âm thanh 1036. Ngoài ra, bộ khuếch đại micrô 1042 có thể được ghép nối với CODEC âm thanh lập thể 1034, micrô 1044 có thể được ghép nối với bộ khuếch đại micrô 1042. Bộ dò sóng vô tuyến điều chế tần số (frequency modulation - FM) 1046 có thể được ghép nối với CODEC âm thanh lập thể 1034. Anten FM 1048 được ghép nối với bộ dò sóng FM 1046. Ngoài ra, các tai nghe lập thể 1050 có thể được ghép nối với CODEC âm thanh lập thể 1034. Các thiết bị khác mà có thể được ghép nối với CPU 1004 bao gồm một hoặc nhiều camera số (ví dụ, CCD hoặc CMOS) 1052.

Modem hoặc bộ thu phát RF 1054 có thể được ghép nối với bộ xử lý tín hiệu tương tự 1008. Modem hoặc bộ thu phát RF 1054 hoặc một phần của nó có thể là ví dụ về AD 108 (Fig.1) hay 208B (Fig.2) được mô tả ở trên. Bộ chuyển đổi RF 1056 có thể được ghép nối với bộ thu phát RF 1054 và anten RF 1058. Ngoài ra, bàn phím 1060, tai nghe đơn âm có micrô 1062 và thiết bị rung 1064 có thể được ghép nối với bộ xử lý tín hiệu analog 1008.

Nguồn cấp điện 1066 có thể được ghép nối với SoC 1002 thông qua mạch tích hợp quản lý nguồn (power management integrated circuit - PMIC) 1068. Nguồn cấp điện 1066 có thể bao gồm pin sạc lại được hoặc nguồn cấp điện một chiều (DC) mà được lấy từ bộ biến đổi AC sang DC kết nối với nguồn điện xoay chiều (AC).

SoC 1002 có thể có một hoặc nhiều cảm biến nhiệt bên trong hoặc trên chip 1070A và có thể được ghép nối với một hoặc nhiều cảm biến nhiệt bên ngoài hoặc ngoài chip 1070B. Bộ điều khiển chuyển đổi tương tự sang số (analog-to-digital converter - ADC) 1072 có thể chuyển đổi các sụt áp tạo ra bởi các cảm biến nhiệt 1070A và 1070B thành các tín hiệu số.

Màn hình cảm ứng 1014, cổng viđeo 1020, cổng USB 1024, camera 1052, loa lập thể thứ nhất 1038, loa lập thể thứ hai 1040, micrô 1044, anten FM 1048, tai nghe lập thể 1050, bộ chuyển đổi RF 1056, anten RF 1058, bàn phím 1060, tai nghe mono 1062, thiết bị rung 1064, cảm biến nhiệt 1050B, bộ điều khiển ADC 1052, PMIC 1068, nguồn cấp điện 1066, DRAM 1030 và thẻ SIM 1026 là bên ngoài SoC 1002 theo phương án làm ví dụ này. Tuy nhiên, cần hiểu rằng theo các phương án khác, một hoặc nhiều trong số các thiết bị này có thể được bao gồm trong một SoC như vậy.

SoC 1002 có thể bao gồm TME 1074, mà có thể là ví dụ về TME 116 (Fig.1) được mô tả trên đây. TME 1074 có thể bao gồm phần cứng bộ xử lý và do vậy có thể

thực thi firmware hoặc phần mềm để điều khiển các phần của các phương pháp được mô tả trên đây.

Bất kể thành phần nào trong số các thành phần của SoC 1002 mà được tạo cấu hình làm các bus chính có thể là các ví dụ về các bus chính 106 (Fig.1), 202A, 202B hoặc 202C (Fig.2) được mô tả trên đây. Modem 1054 hoặc một phần của nó là ví dụ về thành phần mà có thể đóng vai trò như bus chính. Bus chính có thể bao gồm phần cứng bộ xử lý và do đó có thể thực thi firmware hoặc phần mềm để điều khiển các phần của các phương pháp được mô tả trên đây liên quan đến các AD. Tương tự, thành phần bất kỳ trong số các thành phần của SoC 1002 mà được tạo cấu hình như bus phụ có thể là ví dụ về bus phụ 212 (Fig.2) mô tả trên đây mà phục vụ như hoặc cung cấp các tài nguyên. Mặc dù không được thể hiện trên Fig.10 nhằm các mục đích rõ ràng, nhưng các bus phụ có thể được ghép nối với các xPU. Mặc dù điều tương tự không được thể hiện trên Fig. 10, nhưng các bus chính, bus phụ và các thành phần khác của SoC 1002 có thể truyền thông qua hệ thống kết nối hoặc bus như được mô tả trên đây liên quan tới Fig. 2.

Firmware hoặc phần mềm có thể được lưu trữ trong bất kỳ bộ nhớ nào được mô tả ở trên, chẳng hạn như DRAM 1030 hoặc 1031, SRAM 1028, v.v., hoặc có thể được lưu trữ trong bộ nhớ cục bộ mà có thể truy cập trực tiếp bởi phần cứng bộ xử lý mà phần mềm hoặc firmware thực thi trên đó. Bộ nhớ bất kỳ như vậy có firmware hay phần mềm lưu trữ trên đó ở dạng đọc được bằng máy tính để thực thi bằng phần cứng bộ xử lý (ví dụ, CPU, TME, AD v.v.) có thể là ví dụ về "sản phẩm chương trình máy tính", "phương tiện đọc được bằng máy tính" v.v., vì các thuật ngữ như vậy được hiểu trong từ điển sáng chế.

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

YÊU CẦU BẢO HỘ

1. Phương pháp điều khiển truy cập tài nguyên trong hệ thống trên chip, system-on-chip - SoC, phương pháp này bao gồm các bước:

phân bổ (402) đến miền truy cập (108), bởi thành phần thực thi trên bộ xử lý (104), tài nguyên (112) bao gồm vùng bộ nhớ;

tải (404), bởi thành phần, hình ảnh phần mềm liên quan đến miền truy cập vào vùng bộ nhớ;

khóa (406), bởi công cụ quản lý độ tin cậy (116), tài nguyên chống lại truy cập bởi thực thể bất kỳ không phải miền truy cập và công cụ quản lý độ tin cậy;

xác thực (408), bởi công cụ quản lý độ tin cậy, hình ảnh phần mềm liên quan đến miền truy cập;

bắt đầu khởi động (410), bởi công cụ quản lý độ tin cậy, miền truy cập để đáp lại xác thực thành công hình ảnh phần mềm liên quan đến miền truy cập; và

miền truy cập, sau khi khởi động, khóa (412) tài nguyên chống lại truy cập bởi công cụ quản lý độ tin cậy.

2. Phương pháp theo điểm 1, trong đó thành phần bao gồm hệ điều hành cấp cao (110).

3. Phương pháp theo điểm 2, trong đó miền truy cập bao gồm modem.

4. Phương pháp theo điểm 1, trong đó miền truy cập bao gồm một hoặc nhiều bus chính (106).

5. Phương pháp theo điểm 4, trong đó bước khởi động bao gồm bus chính thứ nhất của miền truy cập bắt đầu khởi động bus chính thứ hai của miền truy cập sau khi bus chính thứ nhất hoàn thành khởi động, bus chính thứ hai khởi động độc lập với công cụ quản lý độ tin cậy.

6. Phương pháp theo điểm 1, phương pháp còn bao gồm các bước:

miền truy cập truyền yêu cầu phân bổ tài nguyên bổ sung bao gồm vùng bộ nhớ bổ sung cho hệ điều hành cấp cao, high-level operating system - HLOS;

phân bổ, bởi HLOS, tài nguyên bổ sung cho miền truy cập;

khóa, bởi miền truy cập, tài nguyên bổ sung chống lại truy cập bởi thực thể bất kỳ không phải miền truy cập.

7. Phương pháp theo điểm 6, phương pháp còn bao gồm các bước:

mở khóa, bởi miền truy cập, tài nguyên bổ sung chống lại truy cập bởi một thực thể khác; và

miền truy cập truyền đến HLOS chỉ báo rằng tài nguyên bổ sung đang rỗi.

8. Hệ thống điều khiển truy cập tài nguyên trong hệ thống trên chip, SoC (102), hệ thống này bao gồm:

thành phần thực thi trên bộ xử lý (104) và được tạo cấu hình để:

phân bổ tài nguyên (112) bao gồm vùng bộ nhớ vào miền truy cập (108); và

tải hình ảnh phần mềm liên quan đến miền truy cập vào vùng bộ nhớ; và

công cụ quản lý độ tin cậy (116) được tạo cấu hình để:

khóa tài nguyên chống lại truy cập bởi thực thể bất kỳ không phải miền truy cập và công cụ quản lý độ tin cậy;

xác thực hình ảnh phần mềm liên quan đến miền truy cập; và

bắt đầu khởi động miền truy cập để đáp lại xác thực thành công hình ảnh phần mềm liên quan đến miền truy cập, trong đó miền truy cập còn được tạo cấu hình để, sau khi khởi động, khóa tài nguyên chống lại truy cập bởi công cụ quản lý độ tin cậy.

9. Hệ thống theo điểm 8, trong đó thành phần bao gồm hệ điều hành cấp cao (110).

10. Hệ thống theo điểm 9, trong đó miền truy cập bao gồm modem.

11. Hệ thống theo điểm 8, trong đó miền truy cập bao gồm một hoặc nhiều bus chính (106).

12. Hệ thống theo điểm 11, trong đó việc khởi động bao gồm bus chính thứ nhất của miền truy cập bắt đầu khởi động bus chính thứ hai của miền truy cập sau khi bus chính thứ nhất hoàn thành khởi động, bus chính thứ hai khởi động độc lập với công cụ quản lý độ tin cậy.

13. Hệ thống theo điểm 8, trong đó:

miền truy cập còn được tạo cấu hình để truyền yêu cầu phân bổ tài nguyên bổ sung bao gồm vùng bộ nhớ bổ sung cho hệ điều hành cấp cao, HLOS (110);

HLOS được tạo cấu hình để phân bổ tài nguyên bổ sung cho miền truy cập; và

miền truy cập được tạo cấu hình để khóa tài nguyên bổ sung chống lại truy cập bởi thực thể bất kỳ không phải miền truy cập.

14. Hệ thống theo điểm 13, trong đó miền truy cập còn được tạo cấu hình để:

mở khóa tài nguyên bổ sung chống lại truy cập bởi một thực thể khác; và

truyền đến HLOS chỉ báo rằng tài nguyên bổ sung đang rỗi.

15. Phương tiện đọc được bằng máy tính để điều khiển truy cập tài nguyên trong hệ thống trên chip, SoC, phương tiện đọc được bằng máy tính này có các lệnh được lưu trữ trên đó mà khi được thực thi trên bộ xử lý của SoC điều khiển phương pháp theo các điểm từ 1 đến 7.

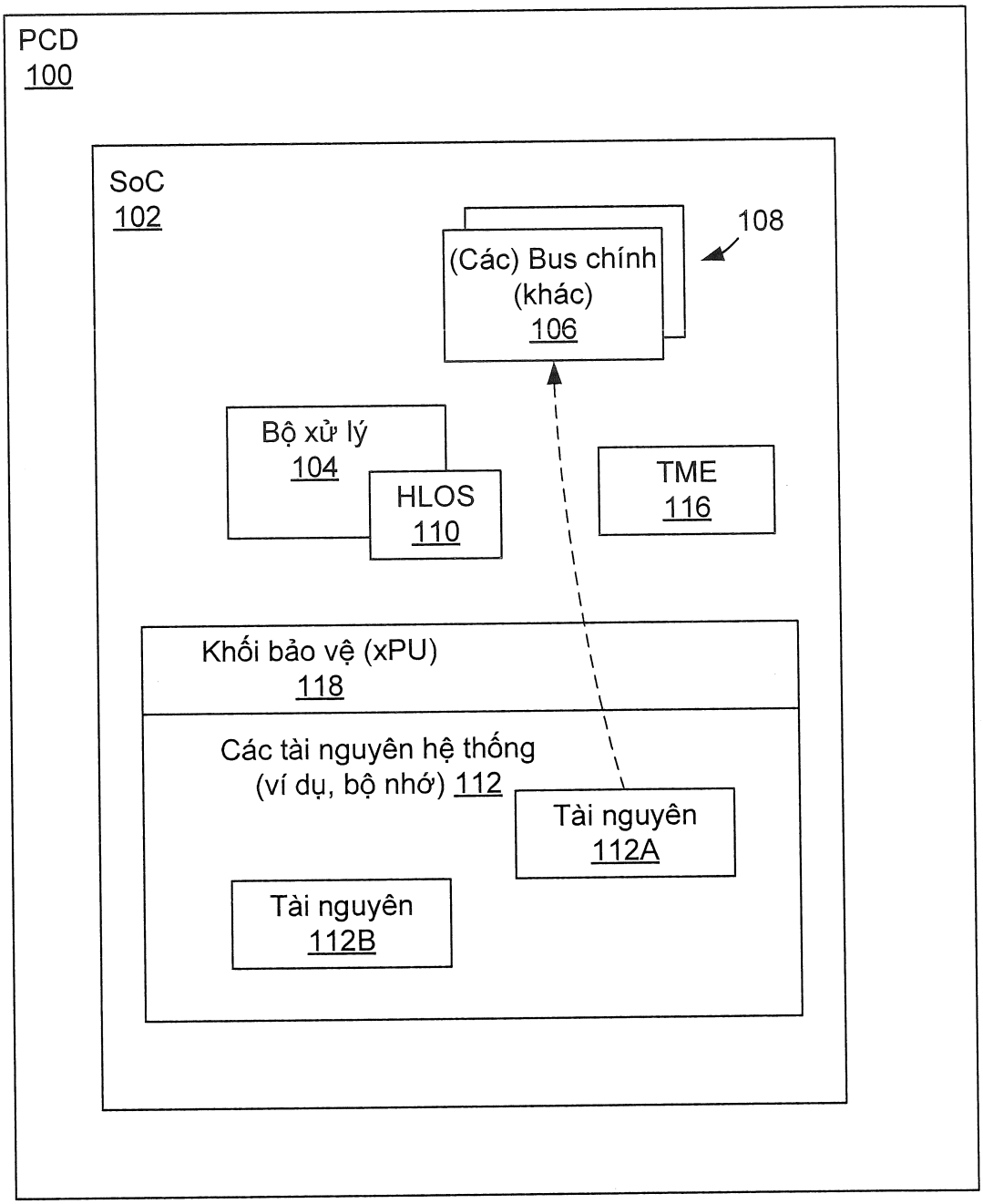

PCD
100
SoC
102
108
(Các) Bus chính
(khác)
106
Bộ xử lý
104
HLOS
110
TME
116
Khối bảo vệ (xPU)
118
Các tài nguyên hệ thống
(ví dụ, bộ nhớ) 112
Tài nguyên
112A
Tài nguyên
112B

Fig.1

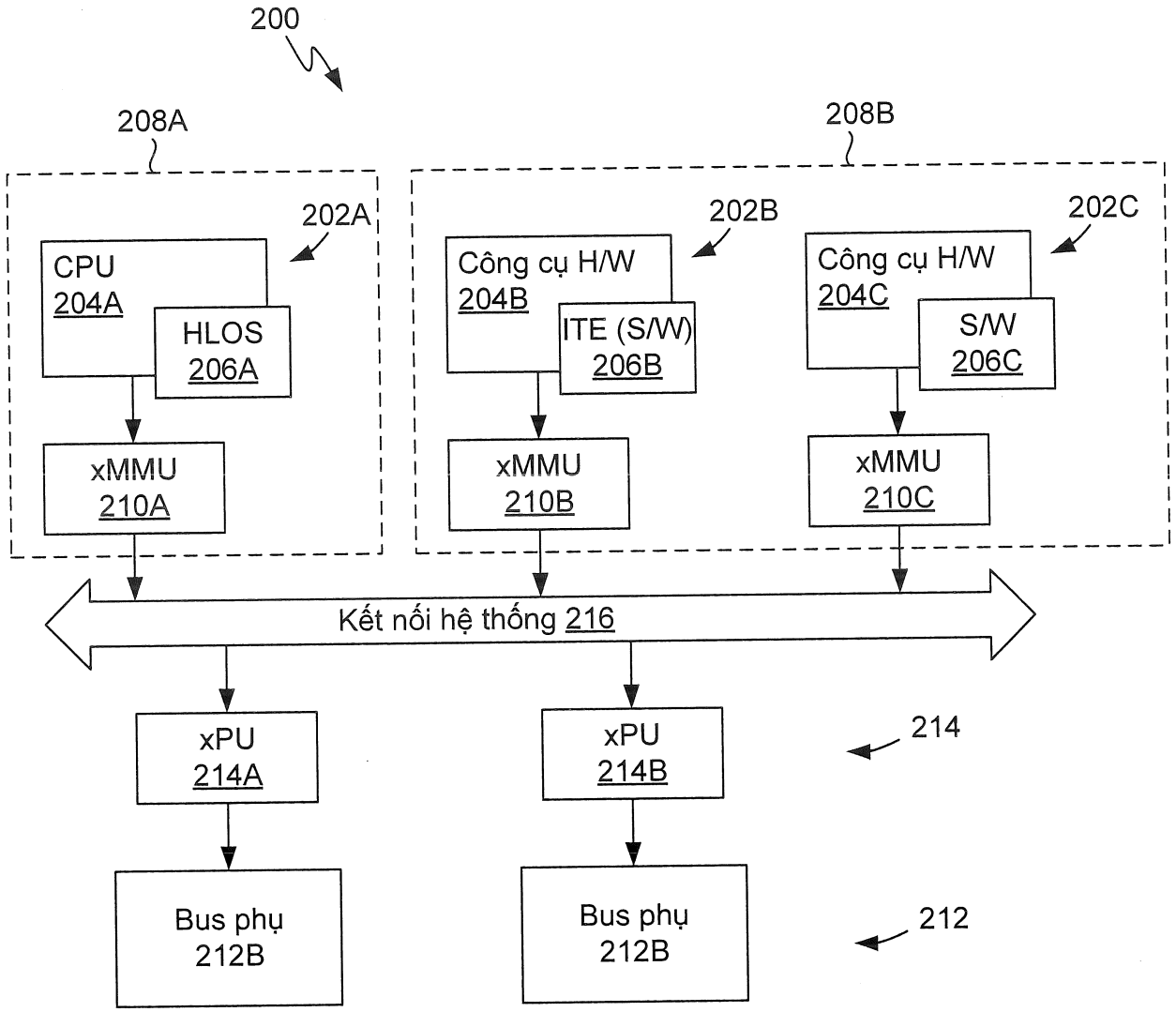
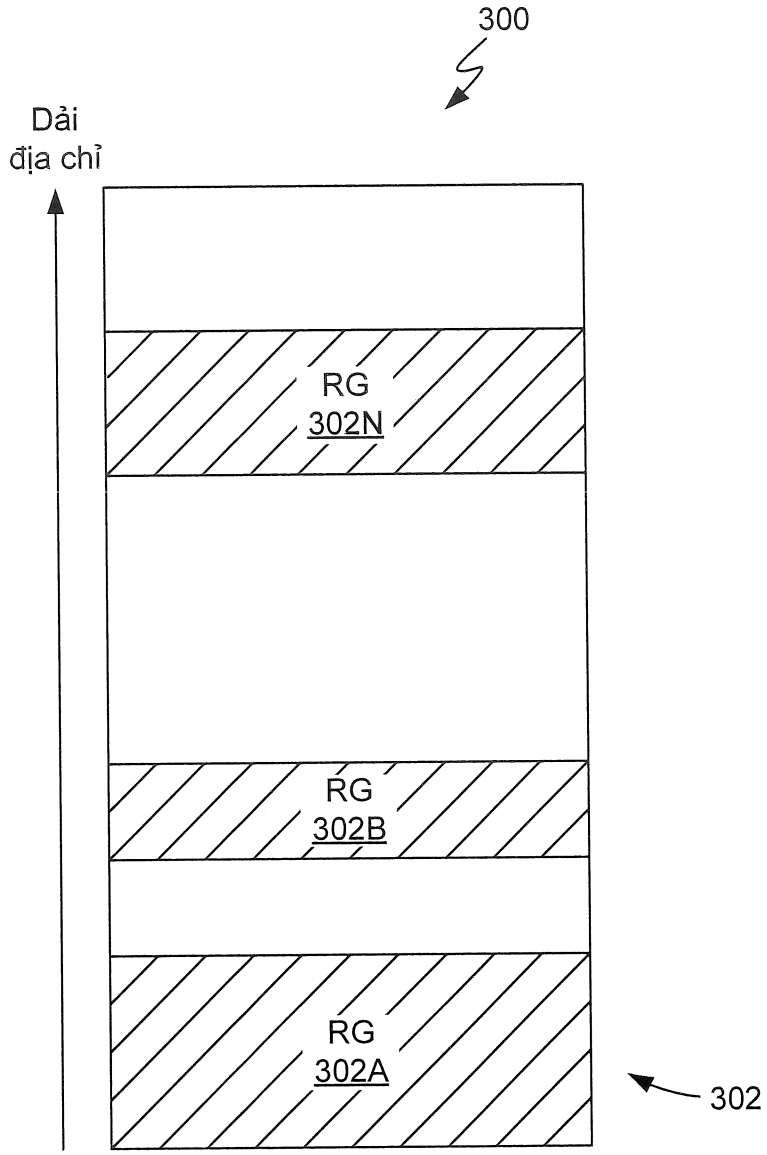
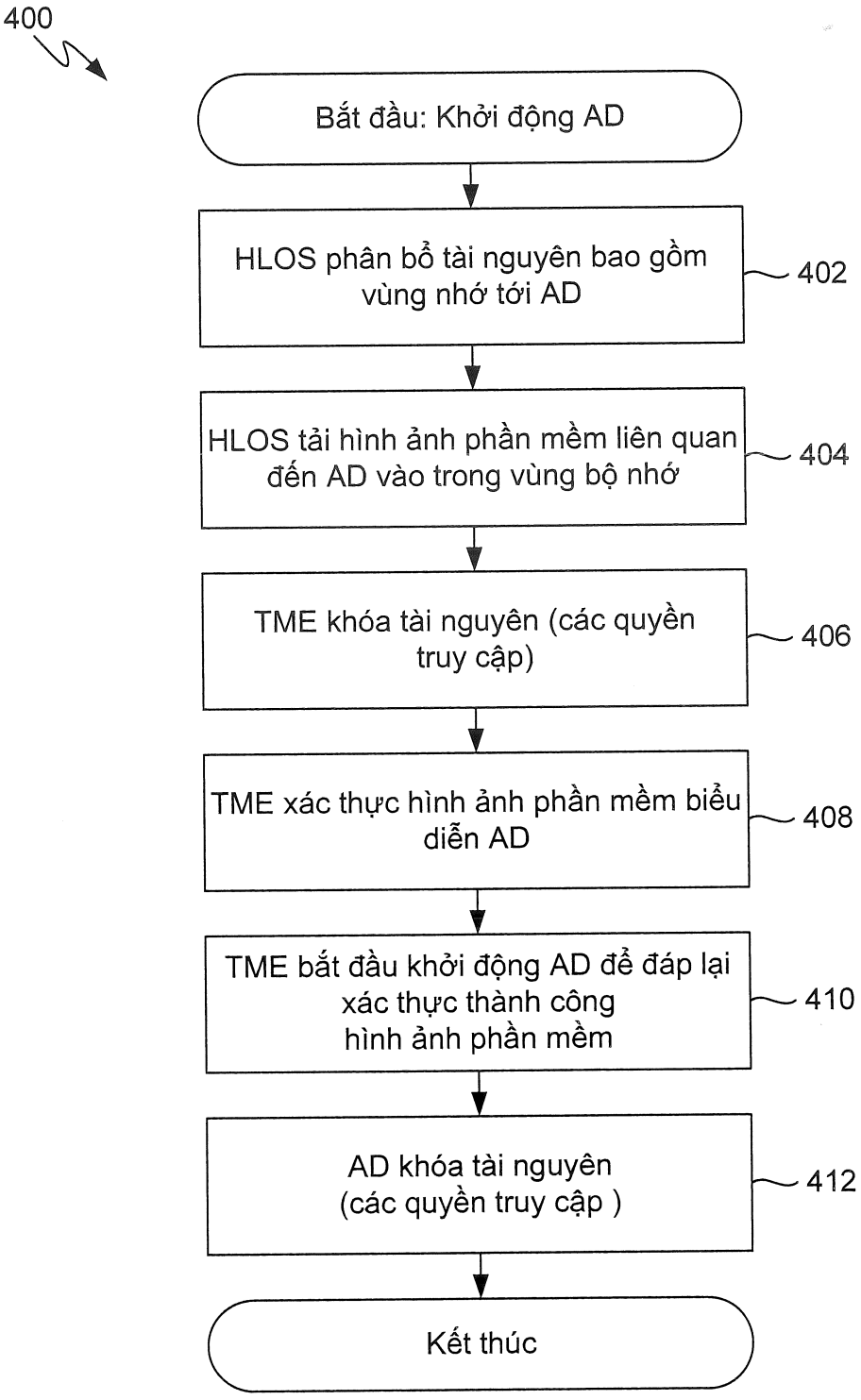

200
208A
208B
202A
202B
202C
CPU
204A
HLOS
206A
Công cụ H/W
204B
ITE (S/W)
206B
Công cụ H/W
204C
S/W
206C
xMMU
210A
xMMU
210B
xMMU
210C
Kết nối hệ thống 216
xPU
214A
xPU
214B
214
Bus phụ
212B
Bus phụ
212B
212

Fig.2

3/11



Hình 3


400
Bắt đầu: Khởi động AD
HLOS phân bổ tài nguyên bao gồm vùng nhớ tới AD
402
HLOS tải hình ảnh phần mềm liên quan đến AD vào trong vùng bộ nhớ
404
TME khóa tài nguyên (các quyền truy cập)
406
TME xác thực hình ảnh phần mềm biểu diễn AD
408
TME bắt đầu khởi động AD để đáp lại xác thực thành công hình ảnh phần mềm
410
AD khóa tài nguyên (các quyền truy cập)
412
Kết thúc

HÌNH 4

5/11

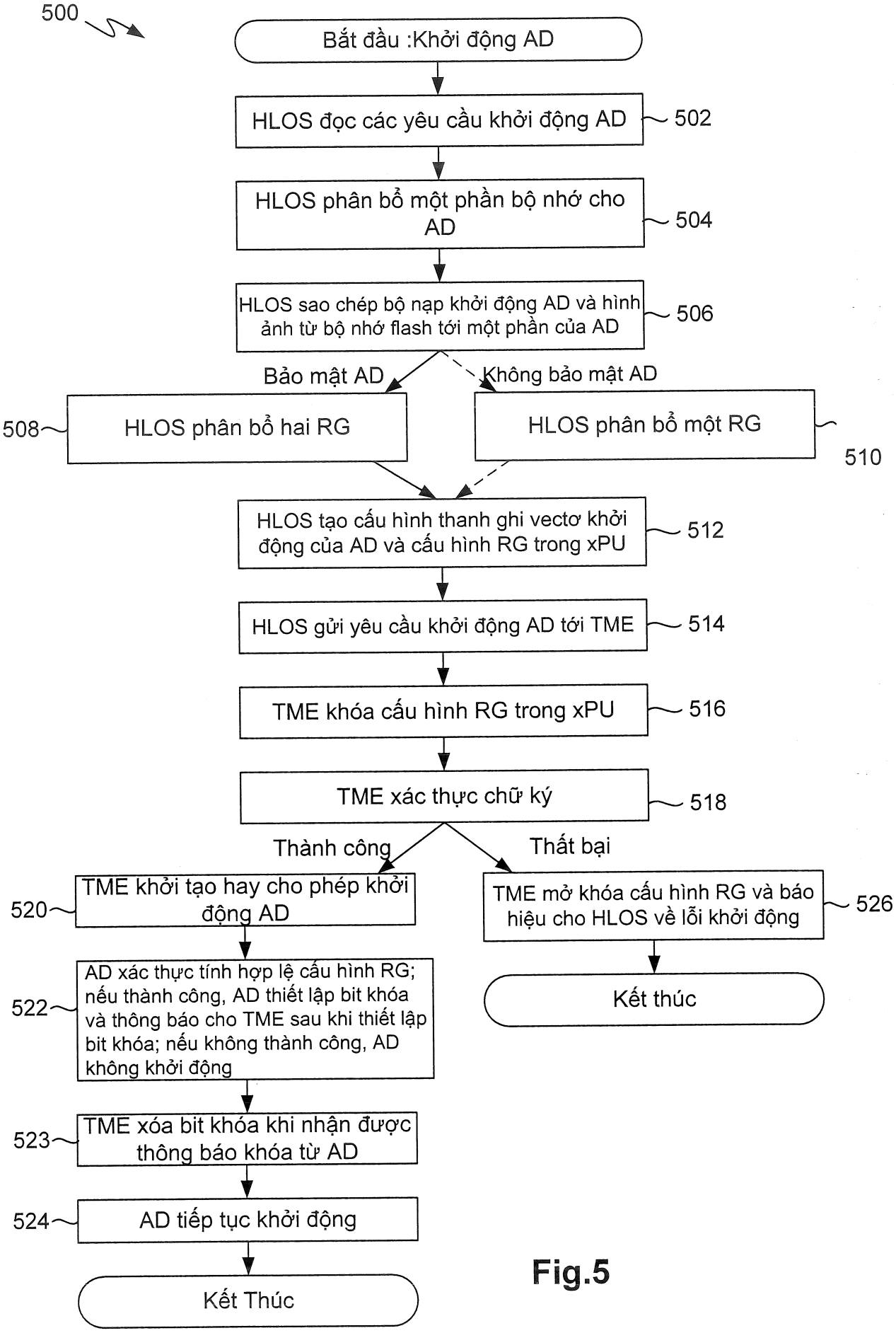

500
Bắt đầu :Khởi động AD
HLOS đọc các yêu cầu khởi động AD
502
HLOS phân bổ một phần bộ nhớ cho AD
504
HLOS sao chép bộ nạp khởi động AD và hình ảnh từ bộ nhớ flash tới một phần của AD
506
Bảo mật AD
Không bảo mật AD
508
HLOS phân bổ hai RG
HLOS phân bổ một RG
510
HLOS tạo cấu hình thanh ghi vectơ khởi động của AD và cấu hình RG trong xPU
512
HLOS gửi yêu cầu khởi động AD tới TME
514
TME khóa cấu hình RG trong xPU
516
TME xác thực chữ ký
518
Thành công
Thất bại
520
TME khởi tạo hay cho phép khởi động AD
TME mở khóa cấu hình RG và báo hiệu cho HLOS về lỗi khởi động
526
522
AD xác thực tính hợp lệ cấu hình RG; nếu thành công, AD thiết lập bit khóa và thông báo cho TME sau khi thiết lập bit khóa; nếu không thành công, AD không khởi động
Kết thúc
523
TME xóa bit khóa khi nhận được thông báo khóa từ AD
524
AD tiếp tục khởi động
Kết Thúc

Fig.5

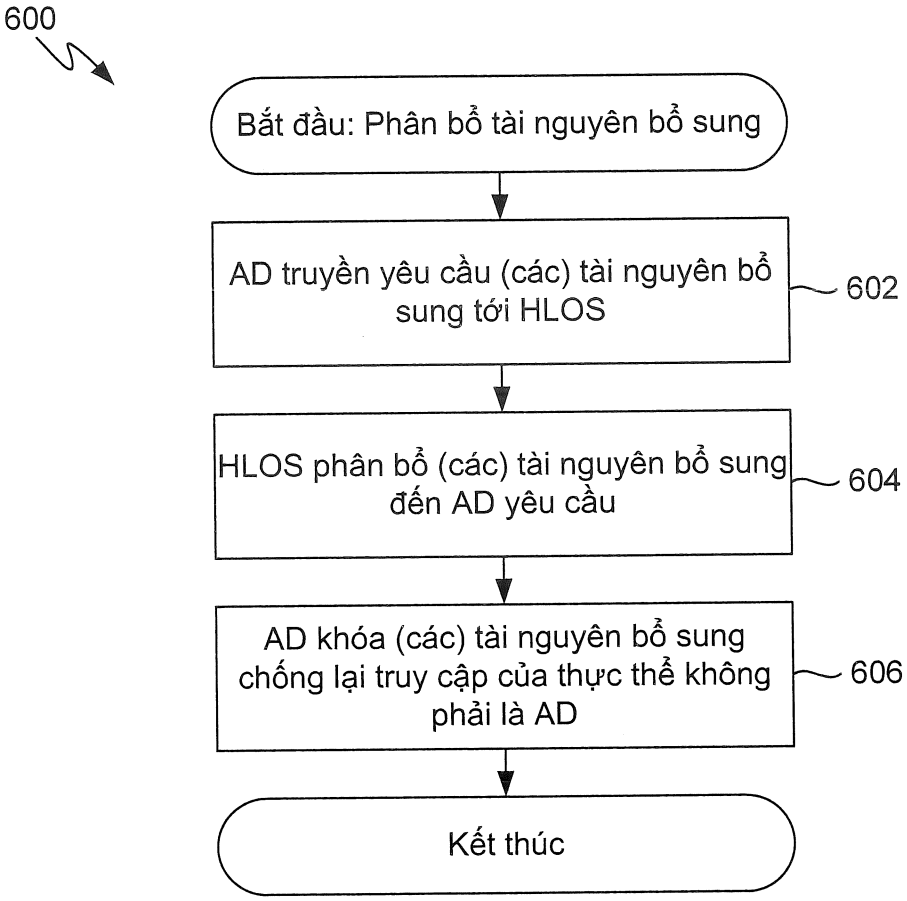

600
Bắt đầu: Phân bổ tài nguyên bổ sung
AD truyền yêu cầu (các) tài nguyên bổ sung tới HLOS
602
HLOS phân bổ (các) tài nguyên bổ sung đến AD yêu cầu
604
AD khóa (các) tài nguyên bổ sung chống lại truy cập của thực thể không phải là AD
606
Kết thúc

Fig.6

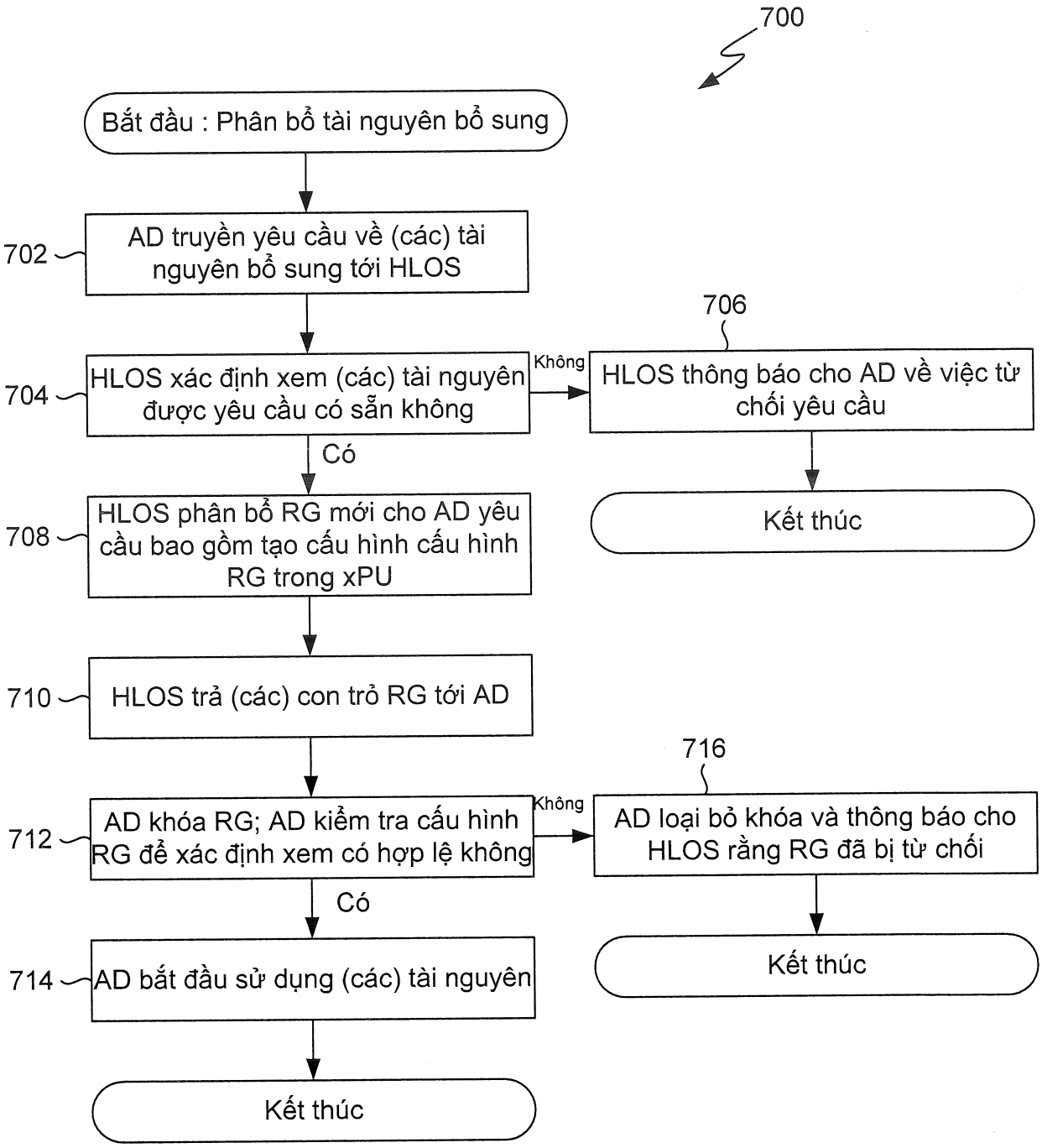

700
Bắt đầu : Phân bổ tài nguyên bổ sung
702
AD truyền yêu cầu về (các) tài nguyên bổ sung tới HLOS
706
704
HLOS xác định xem (các) tài nguyên được yêu cầu có sẵn không
Không
HLOS thông báo cho AD về việc từ chối yêu cầu
Có
Kết thúc
708
HLOS phân bổ RG mới cho AD yêu cầu bao gồm tạo cấu hình cấu hình RG trong xPU
710
HLOS trả (các) con trỏ RG tới AD
716
712
AD khóa RG; AD kiểm tra cấu hình RG để xác định xem có hợp lệ không
Không
AD loại bỏ khóa và thông báo cho HLOS rằng RG đã bị từ chối
Có
714
AD bắt đầu sử dụng (các) tài nguyên
Kết thúc
Kết thúc

Fig.7

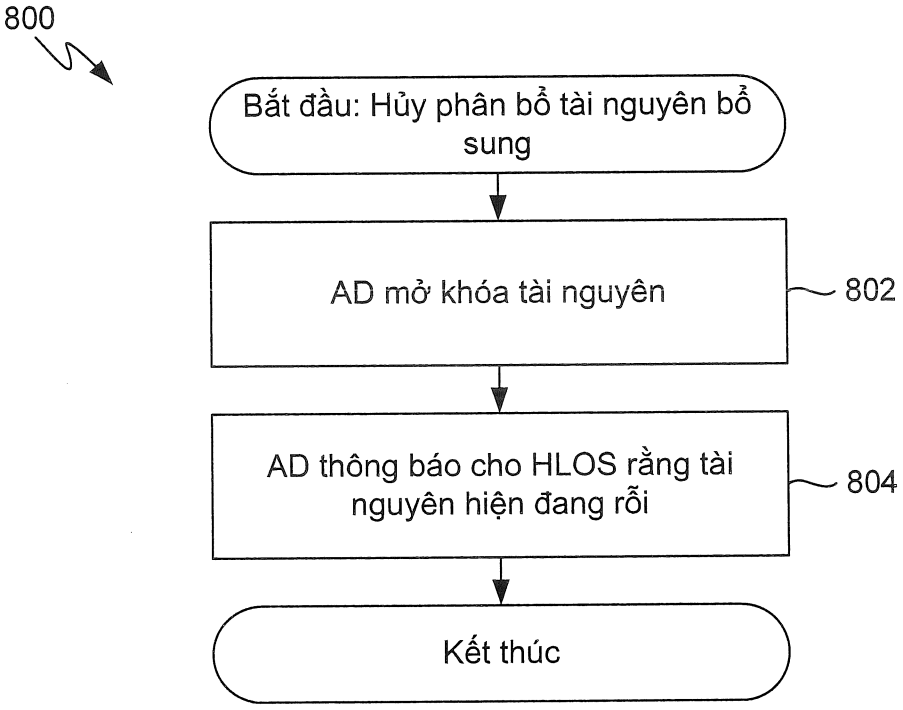

800
Bắt đầu: Hủy phân bổ tài nguyên bổ sung
AD mở khóa tài nguyên
802
AD thông báo cho HLOS rằng tài nguyên hiện đang rỗi
804
Kết thúc

Fig.8

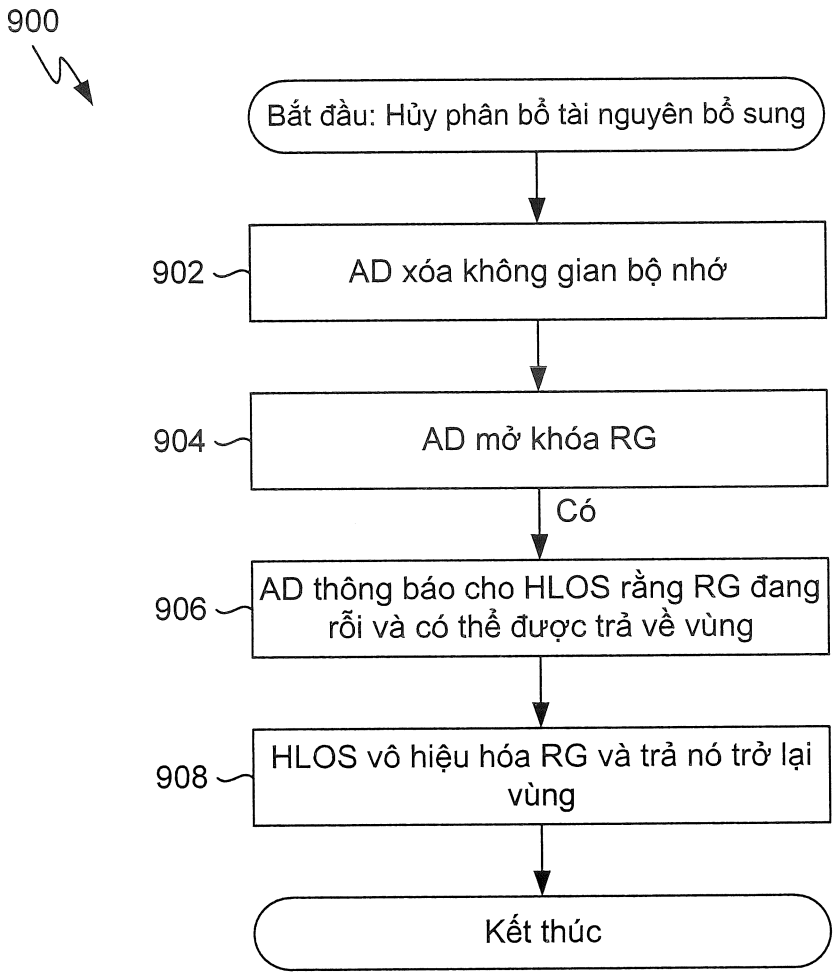

900
Bắt đầu: Hủy phân bổ tài nguyên bổ sung
902
AD xóa không gian bộ nhớ
904
AD mở khóa RG
Có
906
AD thông báo cho HLOS rằng RG đang rỗi và có thể được trả về vùng
908
HLOS vô hiệu hóa RG và trả nó trở lại vùng
Kết thúc

Fig.9

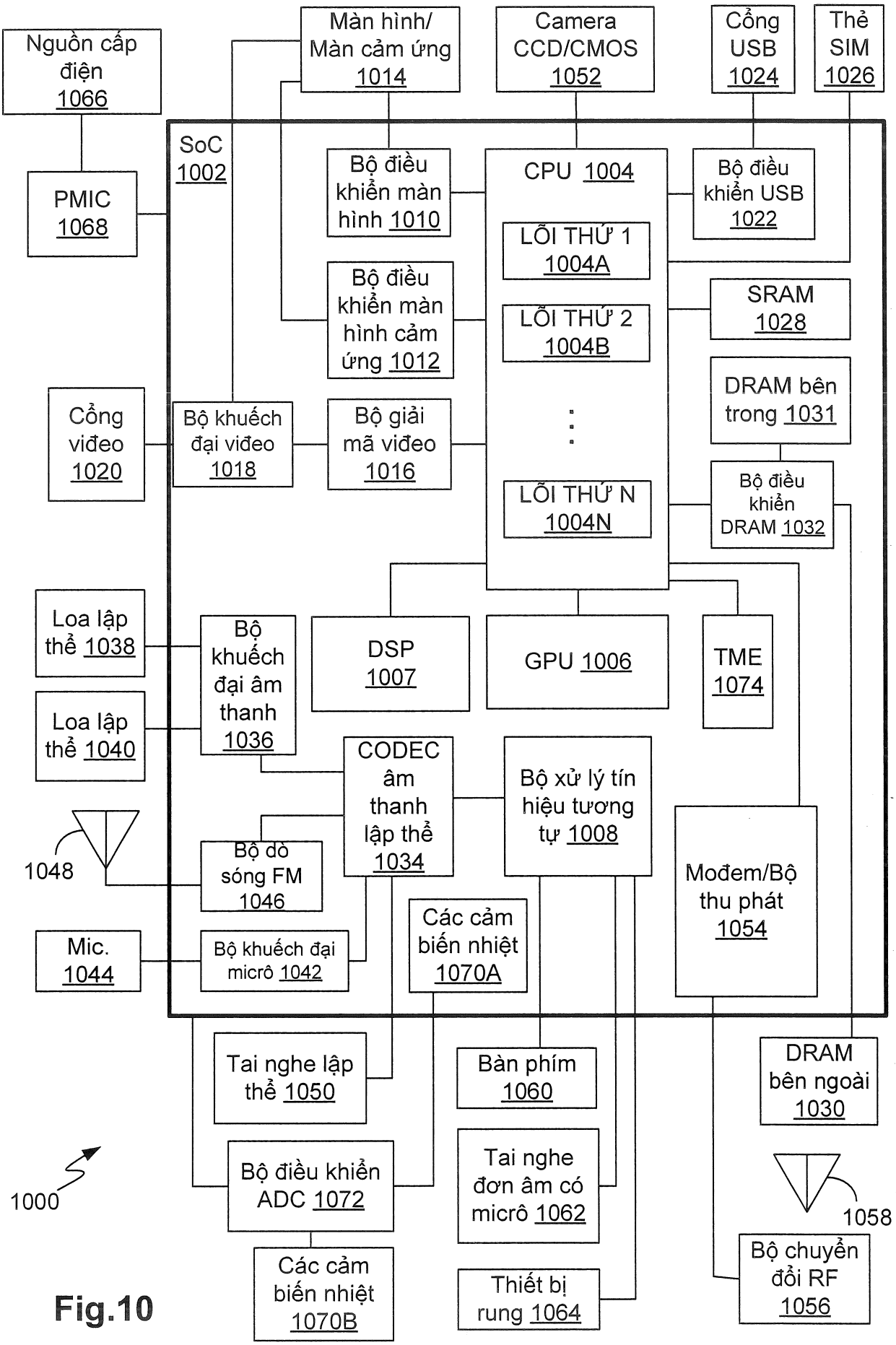

Nguồn cấp điện 1066
Màn hình/ Màn cảm ứng 1014
Camera CCD/CMOS 1052
Cổng USB 1024
Thẻ SIM 1026
SoC 1002
PMIC 1068
Bộ điều khiển màn hình 1010
CPU 1004
Bộ điều khiển USB 1022
LÕI THỨ 1 1004A
Bộ điều khiển màn hình cảm ứng 1012
SRAM 1028
LÕI THỨ 2 1004B
DRAM bên trong 1031
Cổng viđeo 1020
Bộ khuếch đại viđeo 1018
Bộ giải mã viđeo 1016
LÕI THỨ N 1004N
Bộ điều khiển DRAM 1032
Loa lập thể 1038
Bộ khuếch đại âm thanh 1036
DSP 1007
GPU 1006
TME 1074
Loa lập thể 1040
CODEC âm thanh lập thể 1034
Bộ xử lý tín hiệu tương tự 1008
1048
Bộ dò sóng FM 1046
Modem/Bộ thu phát 1054
Mic. 1044
Bộ khuếch đại micrô 1042
Các cảm biến nhiệt 1070A
Tai nghe lập thể 1050
Bàn phím 1060
DRAM bên ngoài 1030
1000
Bộ điều khiển ADC 1072
Tai nghe đơn âm có micrô 1062
1058
Các cảm biến nhiệt 1070B
Thiết bị rung 1064
Bộ chuyển đổi RF 1056

Fig.10

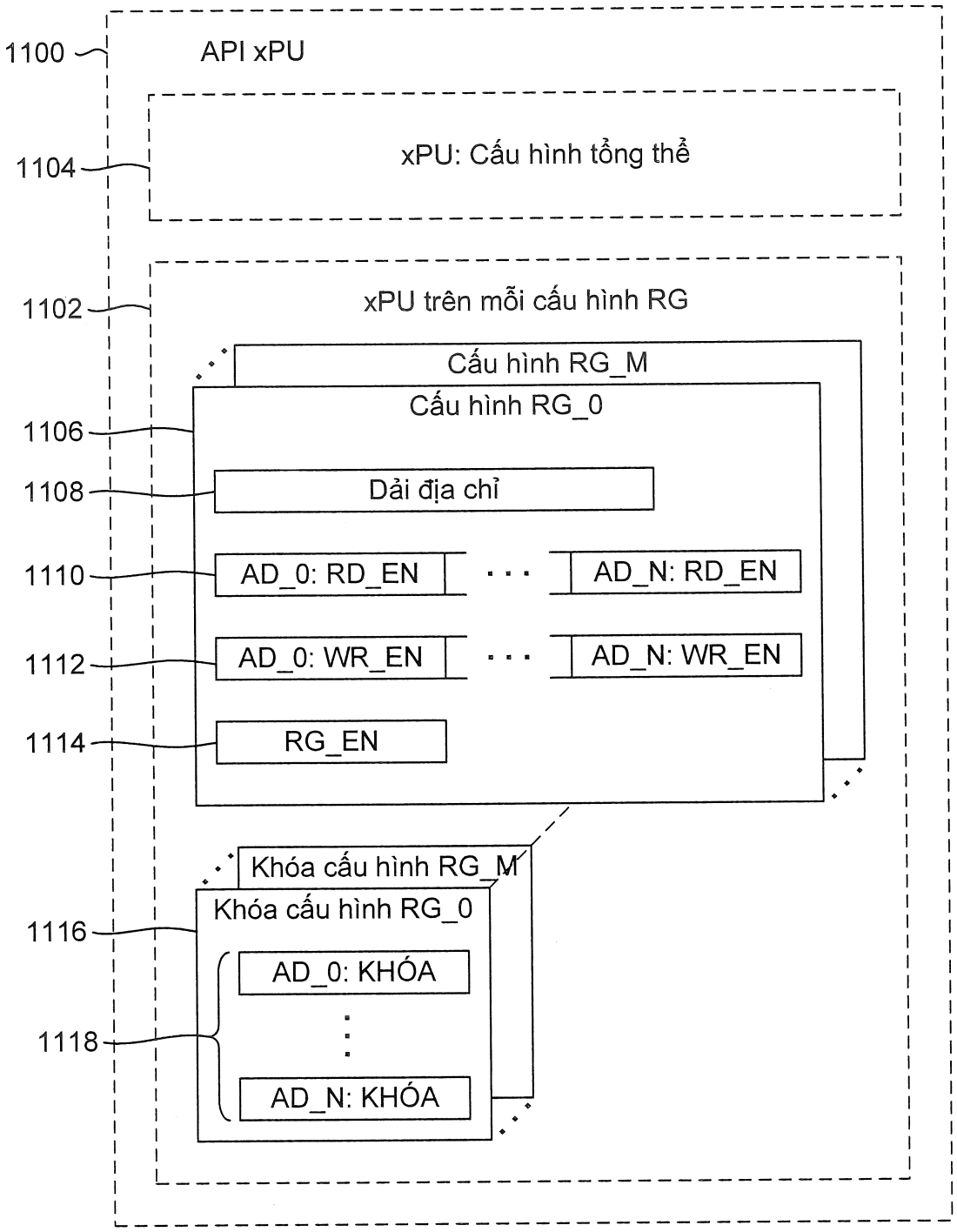

1100
API xPU
1104
xPU: Cấu hình tổng thể
1102
xPU trên mỗi cấu hình RG
Cấu hình RG_M
Cấu hình RG_0
1106
1108
Dải địa chỉ
1110
AD_0: RD_EN
AD_N: RD_EN
1112
AD_0: WR_EN
AD_N: WR_EN
1114
RG_EN
Khóa cấu hình RG_M
Khóa cấu hình RG_0
1116
AD_0: KHÓA
1118
AD_N: KHÓA

Fig.11