República Federativa do Brasil

Ministério do Desenvolvimento, Indústria, Comércio e Serviços

Instituto Nacional da Propriedade Industrial

BR112022024916B1

(11) BR 112022024916-0 B1

(22) Data do Depósito: 24/05/2021

(45) Data de Concessão: 18/03/2025

(54) Título: MÉTODO E SISTEMA PARA O CONTROLE DE ACESSO A RECURSOS EM UM SISTEMA EM CHIP (SOC), E MEMÓRIA LEGÍVEL POR COMPUTADOR

(51) Int.Cl.: G06F 21/57; G06F 21/62.

(30) Prioridade Unionista: 17/06/2020 US 16/903,982.

(73) Titular(es): QUALCOMM INCORPORATED.

(72) Inventor(es): STEVEN HALTER; SAMAR ASBE; MIGUEL BALLESTEROS; GIRISH BHAT; MAHADEVAMURTY NEMANI.

(86) Pedido PCT: PCT US2021033930 de 24/05/2021

(87) Publicação PCT: WO 2021/257251 de 23/12/2021

(85) Data do Início da Fase Nacional: 06/12/2022

(57) Resumo: SISTEMA E MÉTODO DE CONTROLE DE ACESSO PARA ISOLAR DOMÍNIOS DE SEGURANÇA MUTUAMENTE DESCONFIADOS. O controle de acesso a recursos em um sistema em um chip (SoC) pode empregar um agente que é executado em um processador do SoC e um mecanismo de gerenciamento de confiança do SoC. O agente, tal como, por exemplo, um sistema operacional de alto nível ou um hipervisor, pode ser configurado para alocar um recurso, compreendendo uma região de memória, a um domínio de acesso e carregar uma imagem de software associada ao domínio de acesso na região de memória. O mecanismo de gerenciamento de confiança pode ser configurado para travar o recurso contra acesso por qualquer entidade além do domínio de acesso, para autenticar a imagem de software associada ao domínio de acesso e para inicializar o domínio de acesso em resposta a uma autenticação bemsucedida da imagem de software associada ao domínio de acesso.

"MÉTODO E SISTEMA PARA O CONTROLE DE ACESSO A RECURSOS EM UM SISTEMA EM CHIP (SOC), E MEMÓRIA LEGÍVEL POR COMPUTADOR"

DESCRIÇÃO DA TÉCNICA RELACIONADA

[0001] Dispositivos de computação portáteis ("PCDs") estão se tornando uma necessidade para as pessoas em níveis pessoais e profissionais. Esses dispositivos podem incluir telefones celulares, computadores tablet, computadores palmtop, assistentes digitais pessoais ("PDAs"), consoles de jogos portáteis, e outros dispositivos eletrônicos portáteis. PCDs contêm comumente circuitos integrados ou sistemas em um chip ("SoCs") que incluem inúmeros componentes projetados para funcionar em conjunto a fim de distribuir a funcionalidade para um usuário. Por exemplo, um SoC pode conter qualquer número de mecanismos de processamento, tal como modems, unidades de processamento central ("CPUs") com múltiplos núcleos, unidades de processamento gráfico ("GPUs"), etc. Um SoC pode ser acoplado a outros componentes dentro de um PCD, tal como memória de sistema, transceptores de comunicação sem fio (também referidos como modems), câmeras, microfones, alto falantes, etc. Os SoCs e outros componentes podem ser acoplados por um ou mais barramentos, ou outras interconexões, para fornecer a comunicação de dados entre os mesmos. O termo "recurso" pode ser utilizado para fazer referência a um componente ou parte de um componente, tal como uma região da memória, registro, porta, etc., ao qual um processador pode obter acesso através de uma transação de barramento.

[0002] A manutenção da segurança contra o acesso não autorizado dos recursos é uma consideração importante no projeto de SoC. A complexidade de um SoC dá

lugar a exigências de segurança conflitantes. Por exemplo, por um lado, um sistema operacional de alto nível ("HLOS") rodando em um processador no SoC pode precisar ser capaz de limitar o acesso a recursos das entidades rodando em outros processadores de SoC. Por outro lado, um vendedor de SoC pode desejar limitar o acesso pelo HLOS a tais outros processadores de SoC, visto que tal acesso pode expor potencialmente a propriedade intelectual do vendedor de SoC. A memória principal ou do sistema é geralmente o recurso mais substancial ou significativo e, comumente, o HLOS gerencia toda a memória do sistema. Uma necessidade também pode existir para que entidades em diferentes domínios de segurança colaborem, e a geração do mapa de memória de sistema para dedicar partes da memória de sistema a diferentes domínios de segurança pode ser indesejável.

[0003] A segurança é comumente fornecida em um dispositivo com base em SoC em uma das duas formas gerais. Uma forma é fornecer o hardware de SoC que controla a segurança através de todos os subsistemas. Uma desvantagem dessa abordagem é que a mesma não é escalonável. Não é flexível para adicionar novos domínios de segurança, e pode compreender a segurança de subsistemas. Outra forma é se permitir que HLOS controle a segurança através de todos os subsistemas pelo fornecimento de uma forma de entidade super usuário. Uma desvantagem dessa abordagem é que a segurança pode ser exposta à fonte aberta e pode ser comprometida. Nenhuma dessas soluções convencionais soluciona as preocupações de segurança conflitantes descritas acima.

SUMÁRIO DA DESCRIÇÃO

[0004] Sistemas, métodos e produtos de programa

de computador são descritos para controlar o acesso a recursos em um SoC.

[0005] Um método ilustrativo para o controle de acesso a recursos em um SoC pode incluir a alocação de um recurso compreendendo uma região de memória para um domínio de acesso. Um agente, tal como um HLOS, pode controlar ou realizar tal alocação. O método também pode incluir o carregamento de uma imagem de software associada ao domínio de acesso na região de memória. HLOS ou outro agente pode controlar ou realizar tal carregamento. O método pode incluir, adicionalmente, o travamento do recurso contra o acesso por qualquer entidade além do domínio de acesso. Um mecanismo de gerenciamento confiável pode controlar ou realizar tal travamento. O método pode ainda incluir adicionalmente a autenticação da imagem de software associada ao domínio de acesso. O mecanismo de gerenciamento confiável pode controlar ou realizar tal autenticação. O método pode incluir, adicionalmente, a inicialização do domínio de acesso em resposta a uma autenticação bem-sucedida da imagem de software. O mecanismo de gerenciamento confiável pode dar lugar à inicialização.

[0006] Um sistema ilustrativo para o controle de acesso a recursos em um SoC pode incluir um agente que é executado em um processador e um mecanismo de gerenciamento confiável. O agente pode ser configurado para alocar um recurso compreendendo uma região de memória para um domínio de acesso e para carregar uma imagem de software associada ao domínio de acesso na região de memória. O mecanismo de gerenciamento confiável pode ser configurado para travar o recurso contra acesso por qualquer entidade além do domínio

de acesso, para autenticar a imagem de software associada ao domínio de acesso, e inicializar o domínio de acesso em resposta a uma autenticação bem-sucedida da imagem de software associada ao domínio de acesso.

[0007] Outro sistema ilustrativo para o controle de acesso a recursos em um SoC pode incluir meios para a alocação de um recurso compreendendo uma região de memória para um domínio de acesso e para carregar uma imagem de software associada ao domínio de acesso na região de memória. O sistema ilustrativo pode incluir, adicionalmente, meios para se travar o recurso contra acesso por uma entidade além do domínio de acesso, para autenticar a imagem de software associada ao domínio de acesso, e para inicializar o domínio de acesso em resposta a uma autenticação bem-sucedida da imagem de software.

[0008] Um produto de programa de computador ilustrativo, para o controle de acesso a recursos em um SoC, pode compreender um meio legível por computador, possuindo armazenadas no mesmo, instruções que, quando executadas em um ou mais processadores do SoC, controlam um método. O método pode incluir a alocação de um recurso compreendendo uma região de memória para um domínio de acesso. O método também pode incluir o carregamento de uma imagem de software associada ao domínio de acesso na região de memória. O método pode incluir, adicionalmente, o travamento do recurso contra acesso por qualquer entidade além do domínio de acesso. O método pode, ainda, adicionalmente, incluir a autenticação da imagem de software associada ao domínio de acesso. O método pode incluir, adicionalmente, a inicialização do domínio de acesso em resposta a uma autenticação bem-sucedida

da imagem de software.

BREVE DESCRIÇÃO DOS DESENHOS

[0009] Nas figuras, referências numéricas similares se referem a partes similares por todas as várias vistas a menos que indicado o contrário. Para referências numéricas com designações de caractere alfabético tal como "102A" ou "102B", as designações de caractere alfabético podem diferenciar duas partes ou elementos similares presentes na mesma figura. As designações de caractere alfabético para referências numéricas podem ser omitidas quando se pretender que uma referência numérica englobe todas as partes que possuem a mesma referência numérica em todas as figuras.

[0010] A figura 1 é um diagrama em bloco de um SoC que inclui um sistema para o controle de acesso a recursos, de acordo com as modalidades ilustrativas.

[0011] A figura 2 é um diagrama em bloco ilustrando os mestres de barramento e os escravos de barramento em um sistema para o controle de acesso a recursos, de acordo com as modalidades ilustrativas.

[0012] A figura 3 é um mapa de memória de sistema ilustrando um exemplo de grupos de recursos, de acordo com as modalidades ilustrativas.

[0013] A figura 4 é um fluxograma ilustrando um método para o controle de acesso a recursos referentes à inicialização de um domínio de acesso, de acordo com as modalidades ilustrativas.

[0014] A figura 5 é outro fluxograma ilustrando um método de inicialização de um domínio de acesso, de acordo com as modalidades ilustrativas.

[0015] A figura 6 é um fluxograma ilustrando um método de controle de acesso a recursos referente à alocação de um recurso adicional, de acordo com as modalidades ilustrativas.

[0016] A figura 7 é outro fluxograma ilustrando um método de alocação de um recurso adicional, de acordo com as modalidades ilustrativas.

[0017] A figura 8 é um fluxograma ilustrando um método para o controle de acesso a recursos referentes à desalocação de um recurso adicional, de acordo com as modalidades ilustrativas.

[0018] A figura 9 é outro fluxograma ilustrando um método para a desalocação de um recurso adicional, de acordo com as modalidades ilustrativas.

[0019] A figura 10 é um diagrama em bloco de um PCD, de acordo com as modalidades ilustrativas.

[0020] A figura 11 é um diagrama em bloco de uma interface de programa de aplicativo para uma unidade de proteção, de acordo com as modalidades ilustrativas.

DESCRIÇÃO DETALHADA

[0021] O termo "exemplificativo" é utilizado aqui para significar "servindo como um exemplo, caso ou ilustração". O termo "ilustrativo" pode ser utilizado aqui de forma sinônima a "exemplificativo". Qualquer aspecto descrito aqui como "exemplificativo" não deve ser, necessariamente, considerado preferido ou vantajoso sobre outros aspectos.

[0022] Como ilustrado na figura 1, em uma modalidade ilustrativa ou exemplificativa, um PCD 100 pode incluir um SoC 102. O SoC 102 pode incluir um ou mais

processadores 104 e um ou mais outros mestres de barramento 106. Apesar de um barramento ou uma interconexão de sistema similar não ser ilustrado na figura 1 por motivos de clareza, o termo "mestre de barramento" é utilizado nessa descrição para significar qualquer componente que possa iniciar uma transação de barramento. Um grupo de um ou mais mestres de barramento pode ser referido nessa descrição como um domínio de acesso ou "AD". Por exemplo, o grupo de dois mestres de barramento 106 na figura 1 pode formar um AD 108.

[0023] O PCD 100 é pretendido apenas como um exemplo de um dispositivo no qual o SoC 102 pode ser incluído. Mais geralmente, exemplos de dispositivos que podem incluir um SoC, de acordo com a presente descrição, incluem: um sistema de computação (por exemplo, servidor, centro de dados, computador desktop), um dispositivo de computação móvel ou portátil (por exemplo, laptop, telefone celular, veículo, etc.), um dispositivo da Internet das Coisas ("IoT"), um sistema de realidade virtual ("VR"), um sistema de realidade aumentada ("AR"), etc.

[0024] O processador 104 também pode ser referido como uma CPU, um processador de aplicativo, etc., visto que, nessa modalidade ilustrativa, o processador 104 pode executar, dentre outros elementos de software (não ilustrados por motivos de clareza), um sistema operacional de alto nível ("HLOS") 110. O termo "HLOS" como utilizado nessa descrição deve englobar de forma ampla não apenas um HLOS, mas também um hipervisor, um HLOS em combinação com um hipervisor, etc. Uma ação realizada ou controlada pelo processador 104 sob o controle de, ou configurada pela execução do HLOS 110 pode, por motivos de brevidade de

descrição, ser referida nessa descrição como uma ação realizada pelo HLOS 110. De forma similar, uma ação realizada ou controlada por outro hardware de processamento sob o controle de, ou configurada pela execução de qualquer outra entidade de software pode, por motivos de brevidade de descrição, ser referida nessa descrição como uma ação realizada por tal outra entidade de software. Além disso, apesar de para fins de clareza o HLOS 110 ser ilustrado na figura 1 separadamente de outros mestres de barramento 106, deve-se notar que HLOS 110 também é um mestre de barramento (e, dessa forma, também é um tipo de AD).

[0025] O SoC 102 pode incluir recursos de sistema 112. Os recursos de sistema 112 incluem os componentes ou partes dos mesmos do PCD 100 ao qual um mestre de barramento, tal como o HLOS 110, outros mestres de barramento 106, etc. podem direcionar as transações de barramento. Em outras palavras, uma "transação de barramento", como o termo é utilizado nessa descrição, se refere a uma solicitação de um recurso de sistema 112, tal como uma solicitação de leitura ou uma solicitação de escrita. De acordo, os recursos de sistema 112 podem ser identificados por endereços ou faixas de endereço dentro de um espaço de endereço do sistema. Apesar de, para fins de clareza na figura 1, os recursos de sistema 112 serem apresentados como estando dentro do SoC 102, em outras modalidades todos, ou qualquer um dos recursos de sistema podem estar fora de um SoC. Os recursos de sistema 112 podem incluir, por exemplo, memória de acesso randômico dinâmica ("DRAM"), tal como DRAM de taxa de dados dupla ("DDR-DRAM"). Tal DDR-DRAM (ou, por motivos de brevidade, "DDR") pode

fornecer a memória de sistema principal do PCD 100, e uma CPU ou outro processador 104 pode executar os aplicativos (isso é, software) em tal DDR. Os recursos de sistema 112 podem incluir outros tipos de memória, tal como, por exemplo, memória flash, RAM estática ("SRAM"), etc. Os recursos de sistema 112 podem incluir vários registros, portas e outros componentes endereçáveis. Note-se que os recursos de sistema 112 são os sujeitos das solicitações de transação dos mestres de barramento, os recursos de sistema 112 também podem ser referidos como sujeitos de solicitações de transação dos ADs, visto que um AD é uma entidade que compreende um ou mais mestres de barramento.

[0026] Por exemplo, um primeiro recurso de sistema 112A pode compreender uma primeira região da memória, um segundo recurso de sistema 112B pode compreender uma segunda região de memória, etc. Apesar de apenas dois recursos de sistema 112A e 112B serem ilustrados para fins de ilustração, qualquer número de recursos de sistema 112 pode ser alocado no espaço de endereço de sistema. Exceto como descrito de outra forma nessa descrição, os recursos de sistema 112 podem ser solicitados, alocados e desalocados de acordo com os princípios de computação convencionais bem compreendidos pelos versados na técnica. Por exemplo, em resposta a determinadas condições operacionais, um mestre de barramento 106 pode iniciar uma solicitação de um recurso de sistema 112. A presente descrição se refere a sistemas e métodos para o controle do acesso a tal recurso de sistema 112, como indicado de forma conceitual pela seta de linhas interrompidas na figura 1.

[0027] Um mecanismo de gerenciamento confiável

("TME") 116 é um componente envolvido no sistema e método de controle de acesso. Para promover a segurança, o TME 116 pode ser basicamente consubstanciado em um hardware separado do processador 104 no qual o HLOS 110 é executado, tal como outro processador, memória de leitura apenas ("ROM"), etc. (não ilustrados separadamente na figura 1). O TME 116 pode empregar um agente de autenticação (não ilustrado separadamente) que utiliza os métodos criptográficos para autenticar uma imagem de software. Tal agente de autenticação pode ser de um tipo convencional e, portanto, não é descrito nessa descrição. Apesar de para fins de clareza, o TME 116 é ilustrado na figura 1 separadamente dos outros mestres de barramento 106 e o HLOS 110, deve-se notar que o TME 116 também é um mestre de barramento (e, dessa forma, também um tipo de AD).

[0028] Uma unidade de proteção 118 pode ser configurada para proteger seletivamente os recursos de sistema 112. Pode haver vários tipos de unidades de proteção, cada um associado a um componente capaz de servir como ou fornecer um recurso. O termo "xPU" pode ser utilizado nessa descrição para fazer referência de forma ampla a qualquer um dentre os vários tipos ("x") de unidades de proteção associadas aos tipos correspondentes de componentes. Por exemplo, uma xPU associada a uma memória pode ser referida como uma unidade de proteção de memória.

[0029] Como ilustrado na figura 2, um sistema 200 pode incluir dois ou mais mestres de barramento 202A, 202B, 202C, etc. Enquanto uma parte de cada um dos mestres de barramento 202A, 202B, 202C, etc., pode compreender hardware, um ou mais mestres de barramento 202A, 202B, 202C,

etc., também podem incluir uma parte de software. Por exemplo, um primeiro mestre de barramento 202A pode compreender uma CPU 204A, que pode executar um HLOS 206A. A CPU 204A pode ser um exemplo do processamento descrito acima 104 (figura 1). Um segundo mestre de barramento 202B pode, por exemplo, compreender um mecanismo de hardware 204B e uma parte de software que pode ser referida, nessa descrição, como uma entidade inteligente e confiável ("ITE") 206B. De forma similar, um terceiro mestre de barramento 202C pode compreender um mecanismo de hardware 204C e uma parte de software 206C. O segundo e o terceiro mestres de barramento 202B e 202C podem, juntos, formar um domínio de acesso ou AD 208B, tal como o AD 108 descrito acima com relação à figura 1. Em um AD que possui dois ou mais mestres de barramento, a parte de software do pelo menos um dos mestres de barramento é um ITE; partes de software de outros mestres de barramento não precisam ser ITEs. Por exemplo, o AD 208B pode ser um modem, onde os mestres de barramento 202B e 202C são núcleos de modem escalar e de vetor, respectivamente. Apesar de, nessa modalidade, a modalidade de um modem compreender dois mestres de barramento 202B e 202C, em outras modalidades, um modem pode, como outros ADs, consistir de apenas um mestre de barramento. Note-se que o primeiro mestre de barramento 202A também forma um AD 208A. Visto que o AD 208A é caracterizado pelo HLOS 206A na execução, tal AD pode ser referido, por motivos de conveniência nessa descrição, como o HLOS, de forma similar ao HLOS 110 na figura 1.

[0030] Os mestres de barramento 202A, 202B, 202C, etc., podem ser configurados para acessar os recursos através das unidades de gerenciamento de memória

correspondentes ("xMMUs") 210A, 210B, 210C, etc. O termo "xMMU" pode ser utilizado nessa descrição para fazer referência de forma ampla a uma MMU de qualquer um dos vários tipos ("x") de MMUs associadas aos tipos correspondentes de componentes, tal como, por exemplo, a memória de sistema (em contraste com outro nível ou tipo de memória). Apesar de para fins de clareza na figura 2 cada um dos mestres de barramento 202A, 202B, e 202C ser ilustrado acoplado exatamente a uma xMMU correspondente 210A, 210B e 210C, em outros exemplos (não ilustrados) pode haver mais elementos adicionais através dos quais um mestre de barramento pode acessar os recursos, tal como dois ou mais estágios de xMMUs.

[0031] Cada um dentre os vários escravos de barramento 212, tal como os escravos de barramento 212A e 212B, pode servir ou fornecer recursos de sistema correspondentes 112 (figura 1), tal como memória, registros, portas, etc. Apesar de dois escravos de barramento 212A e 212B serem ilustrados, o sistema 200 pode incluir qualquer número de escravos de barramento 212. Cada um dos escravos de barramento 212A, 212B, etc., pode ser acoplado a uma interconexão de sistema (por exemplo, um sistema de barramento) 216. As xMMUs descritas acima 210A, 210B, 210C, etc., também são acopladas à interconexão de sistema 216. Cada AD 208A, 208B, etc. pode, dessa forma, acessar um recurso pela participação em uma transação de barramento através da interconexão de sistema 216.

[0032] Cada um dos escravos de barramento 212 que pode fornecer recursos de sistema 112 (figura 1) é protegido por uma xPU correspondente 214, tal como xPUs 214A e 214B. Uma xPU 214 pode ser um exemplo da unidade de proteção

118 descrita acima com relação à figura 1. Uma xPU 214 pode ser configurada para proteger e evitar o acesso aos recursos de sistema 112 fornecidos pelos escravos de barramento 212 de uma forma descrita abaixo.

[0033] Como ilustrado na figura 3, um mapa de memória de sistema 300 representa regiões em uma memória do sistema, que pode ser referida nessa descrição como grupos de recurso ("RGs") 302. Uma região em uma memória de sistema (por exemplo, DDR), é um exemplo de um recurso de sistema 112 (figura 1). Note-se que os termos "recurso", "recurso de sistema" e "grupo de recursos" (ou "RG") podem ser utilizados essencialmente de forma sincronizada nessa descrição, com o termo "grupo de recursos" (ou "RG") sendo utilizado em alguns casos para fazer uma referência mais especificamente a um recurso que foi alocado para um AD. Como descrito abaixo em maiores detalhes, um RG pode ser alocado para um AD pela configuração de permissões de acesso que permitem que o AD acesse (isso é, complete uma transação de leitura ou escrita com) o RG. O termo "de propriedade de" (um AD) também pode ser utilizado nessa descrição para fazer referência a um RG que foi alocado para um AD. Um recurso que é de propriedade de ou alocado em um AD também pode ser referido como pertencente a, sendo parte, etc., do AD. Visto que o termo é utilizado nessa descrição, "para alocar" um RG para um AD significa não apenas alocar o RG em um sentido convencional (por exemplo, pela definição ou separação do espaço de memória), mas também tornar um AD um proprietário do RG. Um RG pode ser acessado apenas por um AD que é proprietário do RG. Uma tentativa de se acessar um RG por meio de qualquer entidade, além de um proprietário desse RG, é evitada ou

bloqueada pela xPU associada. Nas modalidades ilustrativas descritas aqui, um estado padrão ou inicial (por exemplo, mediante inicialização do PCD 100) pode ser que o HLOS 110 (figura 1) detenha todos os recursos de sistema 112; outros ADs, incluindo o TME 116, não podem acessar quaisquer recursos até o momento em que os recursos possam ser alocados para tais outros ADs.

[0034] Na figura 3, o mapa de memória de sistema 300 ilustra um exemplo no qual vários RGs 302A, 302B, etc., até 302N (coletivamente referidos como RGs 302) foram alocados para os ADs. O restante do mapa de memória de sistema 300 representa os recursos de sistema que não foram explicitamente alocados para os ADs e, portanto, de acordo com a modalidade ilustrativa descrita aqui, permanecem propriedade do HLOS. Os RGs 302 podem ser localizados em qualquer lugar na faixa de endereço de memória, e as localizações apresentadas na figura 3 servem apenas como exemplos.

[0035] Como ilustrado na figura 11, uma interface de programa de aplicativo xPU ("API") 1100, ilustrada de forma conceitual, representa um exemplo das várias configurações que um AD pode fornecer para a xPU para configurar ou controlar como a xPU opera. xPU API 1100 pode possuir duas partes: uma parte de configuração por RG 1102 através da qual um AP pode fornecer a xPU com as configurações (isso é, específicas para) para cada RG, e uma parte de configuração global 1104 através da qual aspectos da operação xPU não específicos de qualquer RG podem ser configurados.

[0036] A parte de configuração por RG 1102

fornece ou programa a xPU com uma configuração de RG 1106 que configura a xPU para um dentre vários RGs (por exemplo, "RG_0" a "RG_M", onde M é um número fixo que define um número máximo suportado (M + 1) de RGs). Cada configuração de RG 1106 pode incluir uma faixa de endereço 1108, permissões de acesso compreendendo permissões de leitura 1110 e permissões de escrita 1112, e uma configuração de ativação de RG 1114. Nessa modalidade ilustrativa, a xPU permite que apenas o HLOS configure uma configuração de RG 1106, e tal permissão está sujeita ao acessório de travamento descrito abaixo. A xPU nessa modalidade bloqueia qualquer tentativa de qualquer entidade, além de HLOS, de configurar uma configuração de RG 1106. Por exemplo, a xPU bloqueia qualquer tentativa de outro AD, TME, etc. de configurar uma faixa de endereço 1108, uma permissão de leitura 1110, uma permissão de escrita 1112, ou a configuração de ativação de RG 1114. Em outras modalidades (não descritas aqui), um acessório pode ser fornecido para permitir que outros ADs determinem as configurações de RG. Por exemplo, apesar de na modalidade ilustrativa descrita aqui, HLOS, por definição, deter todos os recursos de sistema não alocados protegidos pela xPU, em outras modalidades, uma configuração de modo pode ser fornecida na parte de configuração global 1104 através da qual o HLOS pode selecionar se tal propriedade ocorre por definição ou deve ocorrer pela alocação explícita de RGs para o HLOS.

[0037] Utilizando uma faixa de endereço 1108 fornecida por HLOS, xPU pode identificar ou definir um RG que a xPU deve proteger. Por exemplo, HLOS pode definir um dos RGs 32 descritos acima com referência à figura 3 pelo fornecimento de um endereço inicial e endereço final que o

RG 302 abrange.

[0038] A configuração de ativação de RG 1114 pode ser um único bit que o HLOS pode fornecer para ativar a xPU para começar a proteger o RG definido pela faixa de endereço 1106. Por exemplo, uma configuração de ativação de RG 1114 possuindo um valor igual a "1" pode ativar a xPU para aplicar as permissões 1110 e 1112 como descrito abaixo, enquanto uma configuração de ativação de RG 1114, possuindo um valor igual a "0", pode indicar para a xPU que o RG foi desativado, isso é, as permissões 1110 e 1112 não são (ou não são mais) aplicáveis. Além disso, o HLOS pode, efetivamente, retornar um RG para um grupo de recursos disponíveis pela configuração da configuração de ativação de RG 1114 para indicar que as permissões 1110 e 1112 não são mais aplicáveis. Na modalidade ilustrativa descrita aqui, um RG deve ser retornado para o grupo de recursos disponíveis antes de o HLOS poder alocar esse RG para outro AD.

[0039] As permissões de leitura 1110 e permissões de escrita 1112 podem ser fornecidas na forma de um bit de ativação de leitura ("RD_EN") por AD e um bit de ativação de escrita ("WR_EN") por AD, respectivamente. Por exemplo, as permissões de leitura 1110 para um RG podem incluir bits "AD_0:RD_EN" até "AD_N:RD_EN" para os ADs correspondentes (onde N é um número fixo que define um número máximo suportado (N + 1) de ADs). Da mesma forma, as permissões de escrita 1112 podem incluir bits "AD_0:WR_EN" a "AD_N:WR_EN" para os ADs correspondentes. Por exemplo, um bit de ativação de leitura, possuindo um valor igual a "1", pode configurar a xPU para ativar (ou não evitar) a finalização de uma transação de leitura iniciada pelo AD

correspondente direcionado ao RG, e um bit de ativação de leitura, possuindo um valor igual a "0", pode configurar a xPU para evitar a finalização de uma transação de leitura iniciada pelo AD correspondente direcionado para o RG. Da mesma forma, um bit de ativação de escrita, possuindo um valor igual a "1", pode configurar a xPU para permitir a finalização de uma transação de escrita iniciada pelo AD correspondente direcionada ao RG, e um bit de ativação de escrita, possuindo um valor igual a "0", pode configurar a xPU para evitar a finalização de uma transação de escrita iniciada pelo AD correspondente direcionada ao RG.

[0040] Um AD detém ou recebe um RG se pelo menos um bit nas permissões de leitura 1110, ou pelo menos um bit nas permissões de escrita 1112, indicar que o AD pode acessar o RG. Note-se que através das permissões de leitura 1110 e das permissões de escrita 1112 um AD pode ser fornecido com ambas uma permissão de leitura e escrita, uma permissão de leitura apenas, etc., controlando, assim, o tipo de acesso que o AD pode obter para um RG em particular.

[0041] Cada solicitação de transação que um AD gera pode incluir um identificador de AD, isso é, um valor que identifica de forma singular esse AD. Os bits das permissões de leitura e escrita 1110 e 1112 podem ser indexados pelos identificadores de AD. Quando a xPU recebe uma solicitação de transação, a xPU pode comparar o identificador de AD incluído em uma solicitação de transação com o valor de bit correspondente na permissão de leitura 1110 ou na permissão de escrita 1112, e, então, com base na comparação, permitir ou evitar a finalização da transação solicitada

[0042] A parte de configuração por RG 1102 fornece, adicionalmente, a xPU com os grupos de bit de travamento da configuração de RG 1116. Cada grupo de bit de travamento de configuração de RG 1116 corresponde a uma das configurações de RG 1106 descritas acima. Cada grupo de bit de travamento de configuração de RG 1116 pode consistir de vários bits de travamento 1118 igual ao número de ADs. Isso é, cada AD corresponde a um bit de travamento 1118. Por exemplo, um primeiro bit de travamento 1118 ("AD_0:LOCK") corresponde a um primeiro AD, um segundo bit de travamento 1118 ("AD_1:LOCK") corresponde a um segundo AD, etc., apesar de um bit de travamento N ("AD_N:LOCK"), corresponder a um AD N. Visto que HLOS e TME são tipos de ADs, um dos bits de travamento 1118, tal como, por exemplo, o primeiro bit de travamento 1118, pode corresponder a HLOS, e outros dos bits de travamento 1118, tal como, por exemplo, o segundo bit de travamento 1118, pode corresponder a TME.

[0043] A xPU permite que um bit de travamento 1118 seja configurado apenas pelo AD correspondente. Isso é, a xPU bloqueia ou ignora qualquer tentativa de se configurar um bit de travamento 1118 por qualquer entidade além do AD correspondente a esse bit de travamento 1118. Cada bit de travamento 1118 pode possuir um valor indicando um estado de "travado" ou um estado de "destravado". Os termos "travado" e "destravado" se referem à configuração de RG 1106; a configuração de um ou mais bits de travamento 1118 em um grupo de bits de travamento de configuração de RG 1116 para um valor que indica um estado "travado" configura a xPU para evitar que qualquer entidade modifique a configuração de RG correspondente 1106. Isso é, uma configuração de RG 1106 é

travada ou não pode ser modificada quando qualquer um ou mais bits de travamento 1118 no grupo 1116, correspondendo a essa configuração de RG 1106, é configurado. Pela configuração de seu bit de travamento correspondente 1118, um AD pode estabelecer a confiança em uma configuração de RG 1106 evitando que qualquer outro AD, incluindo HLOS, TME, etc. modifique, posteriormente, a configuração de RG 1106.

[0044] A xPU permite que um bit de travamento 1118 em um grupo 1116 seja configurada pelo AD correspondente apenas se esse AD detiver o RG correspondente para esse grupo 1116. Se um AD tentar travar uma configuração de RG 1106 na qual o AD não tem qualquer permissão de acesso ao RG, a xPU bloqueia o sucesso da tentativa.

[0045] O SoC 102 (figura 1) pode realizar uma inicialização a frio ou uma reconfiguração com base em hardware para se tornar pronto para a operação normal ou "modo de missão", que pode incluir fornecer controle de acesso a recursos da forma descrita abaixo. Tal inicialização a frio pode envolver entidades que não são diretamente relevantes para a presente descrição, tal como um ou mais carregadores de inicialização. De acordo, tais detalhes não são descritos aqui. Não obstante, pode ser útil se compreender que um ou mais carregadores de inicialização (que podem ser armazenados na ROM) são executados na CPU 104 para carregar o HLOS 110 em uma memória do sistema (isso é, nos recursos de sistema 112) e inicializar o HLOS 110. Um carregador de inicialização pode carregar o HLOS 110 na memória do sistema, por exemplo, através da cópia do HLOS 110 (imagem de software) a partir da memória flash ou outra memória não volátil. Como resultado disso, HLOS 110 começa

a ser executado. Em alguns exemplos, um carregador de inicialização pode carregar um ambiente de execução seguro ("SEE") ou outro software intermediário, que, por sua vez, carrega e inicializa o HLOS 110. Como compreendido pelos versados na técnica, um SEE (algumas vezes referidos como um ambiente de execução confiável ou "TEE") compreende software que, geralmente em conjunto com o hardware de processador seguro, isola os aplicativos confiáveis que estão sendo rodados no SEE dos aplicativos não confiáveis que estão sendo rodados no HLOS principal (não confiável). Não obstante, no contexto da presente descrição, um SEE é mencionado apenas como um exemplo de software intermediário que pode ser carregado antes de o HLOS ser carregado, e as modalidades não precisam incluir um SEE ou outro software intermediário. Então, como descrito abaixo, o HLOS 110, por sua vez, pode carregar e inicializar um ou mais outros ADs 108. O TME 116 pode ser inicializado simultaneamente com o HLOS 110. Deve-se notar que o TME 116 (e qualquer SEE) pode obter a propriedade dos recursos essencialmente da mesma forma que a descrita abaixo com relação a qualquer outro AD 108. Dessa forma, HLOS 110 pode estabelecer a confiança no TME 116 (e qualquer SEE).

[0046] Como ilustrado na figura 4, um método 400 para o controle de acesso a recursos em um SoC pode compreender inicialização de um AD, tal como, por exemplo, qualquer um dos ADs descritos acima 108 (figura 1) ou 208B (figura 2). Como indicado pelo bloco 402, o HLOS 110 ou outro agente pode alocar um recurso para o AD. Como descrito acima, o recurso pode ser, por exemplo, uma região da memória na qual o AD pode ser executado. Em uma modalidade ilustrativa,

a alocação de um recurso de acordo com o bloco 402 pode incluir a configuração de permissões de acesso em uma xPU protegendo o recurso para permitir que o AD acesse o recurso. Por exemplo, as permissões de acesso podem ser configuradas para fornecer uma permissão de leitura a um AD, permissão de escrita, ou ambas a permissão de leitura e escrita. A alocação do recurso também pode incluir a configuração de permissões de acesso de modo que o TME 116 (figura 1) tenha permissão de acesso suficiente (por exemplo, leitura apenas) para realizar a autenticação descrita abaixo. Com essa exceção para o TME 116, as permissões de acesso são configuradas de modo a impedirem o acesso ao recurso por qualquer outro AD além do AD que está sendo inicializado.

[0047] Apesar de, nas modalidades ilustrativas descritas aqui, o agente que realiza ou controla a alocação de recursos ser o HLOS 110, em outras modalidades um agente, além de um HLOS, pode realizar ou controlar a alocação de recursos ou uma parte de tal alocação. Por exemplo, em outras modalidades um HLOS pode realizar alguns aspectos da alocação de recursos descritos aqui, tal como alocação de faixas de endereço, enquanto outro agente configura as permissões de acesso.

[0048] Como indicado pelo bloco 404, o HLOS 110 ou outro agente pode carregar uma imagem de software associada ao AD na região de memória (recurso). A imagem de software pode ser um ITE e pode ser obtida a partir da ROM ou outra fonte.

[0049] Como indicado pelo bloco 406, o TME 116 (figura 1) pode, então, travar o recurso contra acesso por qualquer entidade além do AD que está sendo inicializado,

com a exceção de que o TME 116 pode ter permissão de acesso suficiente para realizar a autenticação descrita abaixo. Em uma modalidade ilustrativa, o travamento do recurso contra acesso, de acordo com o bloco 406, pode compreender o travamento de uma configuração de recursos pela configuração de um bit de travamento na xPU protegendo o recurso. Visto que as permissões de acesso podem ter sido configuradas pelo HLOS 110 ou outro agente, como descrito acima (bloco 402), o travamento da configuração de recurso, dessa forma, trava o recurso propriamente dito contra o acesso de acordo com essas permissões de acesso. Note-se que o travamento da configuração de recursos dessa forma trava as permissões de acesso contra a modificação pelo HLOS 110 (figura 1).

[0050] Como indicado pelo bloco 408, o TME 116 pode autenticar a imagem de software associada ao AD que foi carregado na memória e travado. Como indicado pelo bloco 410, mediante a autenticação bem-sucedida dessa imagem de software, o TME 116 pode iniciar (ou se a inicialização já tiver sido iniciada, não tomar qualquer ação para evitar a continuação de) a inicialização do AD. Como indicado pelo bloco 412, a inicialização o AD pode, então, travar o recurso contra acesso por qualquer entidade além de si mesmo. Dessa forma, o recurso pode ser travado contra acesso pelo TME 116 e o HLOS 110. Em um método ilustrativo descrito abaixo com relação à figura 5, o AD travando o recurso pode ser condicionado mediante a validação bem-sucedida do recurso pelo AD e mediante a remoção da permissão do TME em acessar o recurso.

[0051] Nos exemplos nos quais um AD inclui dois ou mais mestres de barramento, cada um possuindo uma parte

de software, as operações descritas acima com relação aos blocos 402-412 podem ser realizadas na imagem de software que é o ITE, tal como o ITE descrito acima 206B (figura 2). Em tais exemplos, a inicialização do AD pode resultar no ITE 206B começar a operar ou executar em seu mecanismo de hardware associado (onde o ITE 206B e o mecanismo de hardware associado 204B funcionam juntos como o primeiro mestre de barramento 202B), e o ITE, por sua vez, inicia a inicialização de um segundo mestre de barramento 202C pela realização das operações descritas acima com relação aos blocos 402-412 em uma segunda imagem de software. Em um exemplo no qual o AD inclui três mestres de barramento (não ilustrados), um segundo mestre de barramento pode inicializar um terceiro mestre de barramento, etc. Note-se que o segundo e o terceiro mestres de barramento em tal exemplo podem inicializar independentemente o TME; o TME só precisa inicializar o primeiro mestre de barramento.

[0052] Como ilustrado na figura 5, um método 500 para o controle de acesso a recursos em um SoC pode compreender a inicialização de um AD. O método 500 pode ser um exemplo do método descrito acima 400 (figura 4).

[0053] Como indicado pelo bloco 502, o HLOS pode ler as exigências de inicialização de AD, tal como, por exemplo, as exigências de memória para um AD (por exemplo, o número de partições e o tamanho de cada partição), as exigências de relógio/energia para um AD, etc. Como indicado pelo bloco 504, o HLOS pode alocar uma partição de memória para o AD. A partição de memória é de propriedade de HLOS nesse momento, e é fisicamente contígua. Como indicado pelo bloco 506, o HLOS pode carregar (isso é, copiar) a imagem do

software de AD de uma memória não volátil para a partição de memória de AD. HLOS também pode carregar qualquer carregador de inicialização associado ao AD na partição de memória.

[0054] Um AD pode ou não ser de um tipo que possui uma característica de privacidade. "Privacidade" nesse contexto se refere ao conteúdo de terceira parte que apenas o AD deve poder ler. Esse conteúdo pode ser armazenado na forma criptografada na memória flash, e o HLOS pode carregar o conteúdo criptografado para sua partição dedicada na memória do sistema. Essa região pode ser descriptografada pelo carregador de inicialização primário do AD depois que o AD foi inicializado. Se o AD possuir privacidade, então, como indicado pelo bloco 508, o HLOS pode criar dois RGs: um RG para o carregador de inicialização primária do AD, e outro RG para o resto da alocação de memória de AD. Se o AD não tiver privacidade, então, como indicado pelo bloco 510, o HLOS pode criar um RG para toda a alocação de memória do AD.

[0055] Como indicado pelo bloco 512, o HLOS pode, então, configurar o registro de vetor de inicialização do AD. O HLOS também pode configurar as xPUs protegendo o espaço de registro do AD. Por exemplo, o HLOS pode utilizar a xPU API 1100 (figura 11) para configurar as xPUs com uma configuração de RG 1106. A configuração de RG pode incluir permissões de acesso para que o AD tenha acesso de leitura e escrita ao RG. A configuração de RG também pode incluir permissões de acesso para que o TME tenha acesso de leitura ao RG de modo que o TME possa realizar a autenticação descrita abaixo. Como indicado pelo bloco 514, o HLOS pode, então, enviar uma solicitação para o TME inicializar o AD. Como descrito acima, a inicialização do AD engloba a

inicialização de todos os mestres de barramento no AD.

[0056] Como indicado pelo bloco 516, o TME trava as uma ou mais configurações RG do AD configurando seu bit de travamento nas xPUs correspondentes. No contexto do acessório de travamento, "configurar" um bit de travamento significa configurar a configuração de RG para um estado "travado", e "liberar" um bit de travamento significa configurar a configuração de RG para um estado "destravado". O TME pode ler a informação de alocação de memória e confirmar que as configurações de xPU estão corretas. Note-se que a configuração do bit de travamento do TME impede que qualquer entidade, incluindo o HLOS, o TME, etc. modifique a configuração de RG.

[0057] Como indicado pelo bloco 518, o TME pode, então, autenticar a imagem de software, tal como, por exemplo, pela autenticação da assinatura de imagem de software. A autenticação pode ser realizada de uma forma convencional utilizando as técnicas criptográficas (por exemplo, chaves), como compreendido pelos versados na técnica. Se o TME determinar que a autenticação foi bem-sucedida, o TME pode iniciar a inicialização do AD, como indicado pelo bloco 520. Isso pode incluir o TME ativando os recursos de inicialização e liberando o AD da reconfiguração, de modo que o AD possa inicializar. Como descrito acima, os ADs incluem, comumente, subsistemas com processadores dedicados e domínios de relógio/energia. "Liberação de um AD da reconfiguração" significa a realização de toda a configuração de hardware necessária para trazer o subsistema AD ou processador para fora de um estado de reconfiguração. Enquanto as partes de hardware do AD se tornam ativas da

forma acima, a imagem de software AD ou parte de software do AD começa a ser inicializado ou ser executado na região de memória ou partição dentro da qual foi carregado.

[0058] Como indicado pelo bloco 522, uma vez que o AD começa a ser executado, o mesmo pode validar seus um ou mais RGs pela inspeção da configuração de RG na xPU. Um AD tem sempre permissão para inspecionar (isso é, ler) as configurações de RF, tal como a configuração de RG descrita acima 1106 (figura 11), incluindo as permissões de acesso a RF. Se a validação de uma configuração de RG for bem-sucedida, o AD pode configurar seu bit de travamento correspondente. A validação é bem-sucedida se o AD confirmar que suas permissões de acesso a RG se conformam às expectativas do AD. Por exemplo, um AD pode, geralmente, esperar que as permissões de acesso a RG indiquem que o RG é de propriedade desse AD apenas, isso é, nenhuma entidade além desse AD tem permissão para acessar esse RG. Se as permissões de acesso ao RG permitirem que uma entidade inesperada acesse o RG, então, o AD rejeita a configuração de RG como inválida (isso é, a validação falha) e interrompe a inicialização. Visto que o AD, nesse exemplo, acabou de iniciar a inicialização (isso é, execução), o AD pode esperar que o TME ainda tenha acesso de leitura (mas não acesso de escrita) ao RG.

[0059] Na validação da configuração de RG bem-sucedida, e depois que o AD tenha configurado seu bit de travamento correspondente, o AD pode notificar o TME. Como indicado pelo bloco 523, o TME pode liberar seu bit de travamento quando recebe essa notificação do AD. A liberação do bit de travamento do TME para um RG pode acionar a xPU

para também remover, automaticamente (isso é, sem intervenção por parte do HLOS ou outra entidade), as permissões de acesso a TME para esse RG, visto que a liberação do TME de seu bit de travamento indica que o TME não exige mais acesso ao RG. Visto que o bit de travamento do AD permanece configurado, a configuração de RG é travada contra o acesso pelo TME, o HLOS, e qualquer outro AD que não detenha o RG. Como indicado pelo bloco 524, o AD pode, então, continuar com a inicialização.

[0060] Se o TME determinar (bloco 518) que a autenticação falhou, o TME pode destravar uma ou mais configurações de RG associadas ao RG contendo a imagem de software de AD, e sinalizar ao HLOS de que a operação de inicialização falhou, como indicado pelo bloco 526. O TME também pode controlar um sinal de reconfiguração de hardware (não ilustrado) aplicado ao mecanismo de hardware do AD, e só pode liberar o sinal de reconfiguração de hardware se a autenticação for bem-sucedida.

[0061] Como ilustrado na figura 6, um método 600 para o controle de acesso a recursos em um SOC pode compreender a alocação de um recurso adicional para um AD, tal como, por exemplo, qualquer um dos ADs descritos acima 108 (figura 1) ou 208B (figura 2). Como indicado pelo bloco 602, o AD pode transmitir uma solicitação para um ou mais recursos adicionais para o HLOS. "Adicional" nesse contexto significa em adição ao espaço de memória no qual a imagem de software do AD reside e que foi alocado para o AD quando o AD foi inicializado. Como indicado pelo bloco 604, o HLOS pode, então, alocar o recurso ou recursos adicionais para o AD solicitante. Como indicado pelo bloco 606, o AD pode,

então, pela configuração de um bit de travamento da xPU correspondente, travar a configuração de RG associada ao recurso adicional contra acesso por qualquer entidade além do AD solicitante. Note-se que o travamento do recurso dessa forma trava o recurso contra acesso pelo HLOS, o TME e qualquer outra entidade além do AD solicitante.

[0062] Como ilustrado na figura 7, um método 700 para o controle de acesso a recursos em um SoC pode compreender a alocação de um recurso adicional para um AD. O método 700 pode ser um exemplo do método descrito acima 600 (figura 6).

[0063] Como indicado pelo bloco 702, o AD pode transmitir uma solicitação de um ou mais recursos adicionais para o HLOS. A solicitação pode incluir, por exemplo, uma quantidade de espaço de memória sendo solicitada, além de permissões de acesso sendo solicitadas. Como indicado pelo bloco 704, o HLOS pode determinar se o recurso solicitado está disponível. Se o recurso solicitado não estiver disponível, o HLOS pode notificar o AD de que a solicitação foi rejeitada, como indicado pelo bloco 706. Se o recurso solicitado estiver disponível, o HLOS pode alocar um novo RG, como indicado pelo bloco 708. A alocação do novo RG pode incluir fornecer ou programar a xPU protegendo o recurso com uma configuração de RG, que pode incluir uma faixa de endereços de RG, permissões de acesso, etc. Por exemplo, o HLOS pode utilizar a xPU API 1100 (figura 11) para configurar a xPU com uma configuração de RG 1106. Então, o HLOS pode retornar um apontador para o RG recém-criado para o AD solicitante, como indicado pelo bloco 710.

[0064] Como indicado pelo bloco 712, o AD

solicitante pode travar o RG e inspecionar a configuração de RG para determinar se é "válida" ou "inválida". Como descrito acima com relação ao bloco 522 (figura 5), um AD pode considerar uma configuração de RG válida se as permissões de acesso a RG estiverem de acordo com as expectativas do AD. Por exemplo, se o AD esperar que nenhuma outra entidade tenha acesso ao recurso, então, as permissões de acesso a RG devem permitir apenas que o AD tenha acesso ao recurso. Se as permissões de acesso a RG permitirem que qualquer outra entidade acesse o recurso, então, o AD rejeita a configuração de RG como inválida. Se a configuração de RG for "inválida", o AD pode remover o travamento e notificar o HLOS de que o RG foi rejeitado, como indicado pelo bloco 716. Se a configuração de RG for "válida", o AD pode começar a utilizar o RG, como indicado pelo bloco 714.

[0065] Como ilustrado na figura 8, um método 800 para o controle de acesso a recursos em um SoC pode compreender a desalocação de um recurso adicional que foi alocado para um AD, tal como pelo método descrito acima 600 (figura 6) ou 700 (figura 7). Como indicado pelo bloco 802, um AD pode destravar o recurso. Como indicado pelo bloco 804, o AD pode, então, notificar o HLOS de que o recurso está, agora, livre.

[0066] Como ilustrado na figura 9, um método 900 para o controle de acesso a recursos em um SoC pode compreender a desalocação de um recurso adicional que foi alocado para um AD. O método 900 pode ser um exemplo do método descrito acima 800 (figura 8).

[0067] Como indicado pelo bloco 902, um AD pode zerar o espaço de memória que foi alocado para um RG. Zerar

o espaço de memória apaga os dados que, do contrário, seriam suscetíveis ao acesso não autorizado e, dessa forma, fornece a segurança ou privacidade adicional que pode ser desejável em algumas modalidades. Como indicado pelo bloco 904, o AD pode destravar o RG. Como indicado pelo bloco 906, o AD pode, então, notificar o HLOS de que o RG agora está livre e pode ser retornado para o grupo ou lista de RGs "não utilizados". Como indicado pelo bloco 908, o HLOS pode desativar o RG e retornar o mesmo para o grupo. "Desativar um RG" significa que a xPU não utiliza mais esse RG para controlar o acesso. O HLOS pode desativar o RG utilizando a configuração de ativação de RG 1114 da xPU API 1100 (figura 11).

[0068] Como ilustrado na figura 10, as modalidades ilustrativas dos sistemas e métodos de controle de acesso a recursos podem ser consubstanciadas em um PCD 1000. O PCD 1000 inclui um SoC 1002. O SoC 1002 pode incluir uma CPU 1004, uma GPU 1006, um DSP 1007, um processador de sinal analógico 1008 ou outros processadores. A CPU 1004 pode incluir múltiplos núcleos, tal como um primeiro núcleo 1004A, um segundo núcleo 1004B, etc. até um núcleo N 1004N. A CPU 1004 ou qualquer um de seus núcleos pode ser um exemplo do processador descrito acima 104 (figura 1) ou CPU 204A (figura 2).

[0069] Um controlador de exibição 1010 e um controlador de tela de toque 1012 podem ser acoplados à CPU 1004. Um monitor de tela de toque 1014 externo ao SoC 1002 pode ser acoplado ao controlador de exibição 1010 e ao controlador de tela de toque 1012. O PCD 1000 pode incluir, adicionalmente, um decodificador de vídeo 1016 acolado à CPU 1004. Um amplificador de vídeo 1018 pode ser acoplado ao

decodificador de vídeo 1016 e ao monitor de tela de toque 1014. Uma porta de vídeo 1020 pode ser acoplada ao amplificador de vídeo 1018. Um controlador de barramento serial universal ("USB") 1022 também pode ser acoplado à CPU 1004, e uma porta USB 1024 pode ser acoplada ao controlador USB 1022. Um cartão de módulo de identidade de assinante ("SIM") 1026 também pode ser acoplado à CPU 1004.

[0070] Uma ou mais memórias podem ser acopladas à CPU 1004. As uma ou mais memórias podem incluir ambas as memórias volátil e não volátil. Exemplos de memórias voláteis incluem a memória de acesso randômico estática ("SRAM") 1028 e RAMs dinâmicas ("DRAMs") 1030 e 1031. As DRAMs 1030 e 1031 ou partes das mesmas podem ser exemplos da memória de sistema descrita acima 112 (figura 1). Tais memórias podem estar fora do SoC 1002, tal como a DRAM 1030, ou dentro do SoC 1002, tal como a DRAM 1031. Um controlador DRAM 1032 acoplado à CPU 1004 pode controlar a escrita de dados em, e a leitura de dados a partir das DRAMs 1030 e 1031. Em outras modalidades, tal controlador DRAM pode ser incluído em um processador, tal como a CPU 1004. A CPU 1004 pode executar um HLOS ou outro software que é armazenado em qualquer uma das memórias mencionadas acima.

[0071] Um CODEC de áudio estéreo 1034 pode ser acoplado ao processador de sinal analógico 1008. Adicionalmente, um amplificador de áudio 1036 pode ser acoplado ao CODEC de áudio estéreo 1034. O primeiro e o segundo alto falantes estéreo 1038 e 1040, respectivamente, podem ser acoplados ao amplificador de áudio 1036. Adicionalmente, um amplificador de microfone 1042 pode ser acoplado ao CODEC de áudio estéreo 1034, e um microfone 1044

pode ser acoplado ao amplificador de microfone 1042. Um sintonizador de rádio de modulação de frequência ("FM") 1046 pode ser acoplado ao CODEC de áudio estéreo 1034. Uma antena FM 1048 pode ser acoplada ao sintonizador de rádio FM 1046. Adicionalmente, fones de ouvido estéreo 1050 podem ser acoplados ao CODEC de áudio estéreo 1034. Outros dispositivos que podem ser acoplados à CPU 1004 incluem uma ou mais câmeras digitais (por exemplo, CCD ou CMOS) 1052.

[0072] Um modem ou transceptor de RF 1054 pode ser acoplado ao processador de sinal analógico 1008. O modem ou transceptor de RF 1054, ou uma parte do mesmo, pode ser um exemplo do AD descrito acima 108 (figura 1) ou 208B (figura 2). Um comutador de RF 1056 pode ser acoplado ao transceptor de RF 1054 e a uma antena de RF 1058. Adicionalmente, um teclado 1060, um fone de ouvido mono com um microfone 1062, e um dispositivo de vibração 1064 podem ser acoplados ao processador de sinal analógico 1008.

[0073] Um suprimento de energia 1066 pode ser acoplado ao SoC 1002 através de um circuito integrado de gerenciamento de energia ("PMIC") 1068. O suprimento de energia 1066 pode incluir uma bateria recarregável ou um suprimento de energia DC que é derivado de um transformador AC-para-DC conectado a uma fonte de energia AC.

[0074] O SoC 1002 pode possuir um ou mais sensores térmicos em chip ou internos 1070A e pode ser acoplado a um ou mais sensores térmicos fora de chip ou externos 1070B. Um controlador de conversor de analógico para digital ("ADC") 1072 pode converter quedas de voltagem produzidas pelos sensores térmicos 1070A e 1070B em sinais digitais.

[0075] O monitor de tela de toque 1014, a porta de vídeo 1020, a porta USB 1024, a câmera 1052, o primeiro alto falante estéreo 1038, o segundo alto falante estéreo 1040, o microfone 1044, a antena FM 1048, os fones de ouvido estéreo 1050, o comutador RF 1056, a antena de RF 1058, o teclado 1060, o fone de ouvido mono 1062, o vibrador 1064, os sensores térmicos 1050B, o controlador ADC 1052, o PMIC 1068, o suprimento de energia 1066, a DRAM 1030, e o cartão SIM 1026 são externos ao SoC 1002 nessa modalidade ilustrativa. Será compreendido, no entanto, que em outras modalidades um ou mais desses dispositivos podem ser incluídos em tal SoC.

[0076] O SoC 1002 pode incluir um TME 1074, que pode ser um exemplo do TME descrito acima 116 (figura 1). O TME 1074 pode incluir hardware de processador e, de acordo, pode executar firmware ou software a fim de controlar partes dos métodos descritos acima.

[0077] Qualquer um dos componentes do SoC 1002, que são configurados como mestres de barramento, pode ser exemplo dos mestres de barramento descritos acima 106 (figura 1), 202A, 202B ou 202C (figura 2). O modem 1054 ou uma parte do mesmo é um exemplo de um componente que pode servir como um mestre de barramento. Os mestres de barramento podem incluir hardware de processador e, de acordo, podem executar firmware ou software para controlar as partes dos métodos descritos acima com relação aos ADs. De forma similar, qualquer um dos componentes do SoC 1002 que são configurados como escravos de barramento pode ser um exemplo dos escravos de barramento descritos acima 212 (figura 2) que servem como ou fornecem recursos. Apesar de não ilustrado na figura 10

por motivos de clareza, os escravos de barramento podem ser acoplados a xPUs. Apesar de, da mesma forma, não ser ilustrado na figura 10, o mestre de barramento, o escravo de barramento, e outros componentes do SoC 1002 podem se comunicar através de uma interconexão ou sistema de barramento, como descrito acima com relação à figura 2.

[0078] Firmware ou software pode ser armazenado em qualquer uma das memórias descritas acima, tal como DRAM 103 ou 1031, SRAM 1028, etc., ou pode ser armazenado em uma memória local diretamente acessível pelo hardware de processador no qual o software ou firmware é executado. Qualquer memória como essa, possuindo firmware ou software armazenado na mesma na forma legível por computador, para execução pelo hardware processador (por exemplo, CPU, TME, AD, etc.) pode ser um exemplo de um "produto de programa de computador", "meio legível por computador", etc., como tais termos são compreendidos no léxico de patente.

[0079] Modalidades alternativas se tornarão aparentes aos versados na técnica à qual a invenção pertence sem se distanciar de seu espírito e escopo. Portanto, apesar de aspectos selecionados terem sido ilustrados e descritos em detalhes, será compreendido que várias substituições e alterações podem ser realizadas aqui sem se distanciar do espírito e escopo da presente invenção, como definido pelas reivindicações em anexo.

REIVINDICAÇÕES

1. Método para o controle de acesso a recursos em um sistema em chip, SoC, **caracterizado** pelo fato de que compreende:

alocação (402), para um domínio de acesso (108), por um agente executado em um processador (104), de um recurso (112) compreendendo uma região de memória;

carregamento (404), pelo agente, de uma imagem de software associada ao domínio de acesso na região de memória;

travamento (406), por um mecanismo de gerenciamento de confiança (116), do recurso contra acesso por qualquer entidade além do domínio de acesso e do mecanismo de gerenciamento de confiança;

autenticação (408), pelo mecanismo de gerenciamento de confiança, da imagem de software associada ao domínio de acesso; e

inicialização de booting (410), pelo mecanismo de gerenciamento de confiança, do domínio de acesso em resposta a uma autenticação bem-sucedida da imagem de software associada ao domínio de acesso; e

o domínio de acesso, após booting, travando (412) o recurso contra o acesso pelo mecanismo de gerenciamento de confiança.

2. Método, de acordo com a reivindicação 1, **caracterizado** pelo fato de que o agente compreende um sistema operacional de alto nível (110).

3. Método, de acordo com a reivindicação 2, **caracterizado** pelo fato de que o domínio de acesso compreende um modem.

4. Método, de acordo com a reivindicação 1,

caracterizado pelo fato de que o domínio de acesso compreende um ou mais barramentos mestre (106).

5. Método, de acordo com a reivindicação 4, **caracterizado** pelo fato de que booting compreende um primeiro barramento mestre do domínio de acesso inicializando booting de um segundo barramento mestre do domínio de acesso, depois que o primeiro barramento mestre completa o booting, o segundo barramento mestre realizando o booting independentemente do mecanismo de gerenciamento de confiança.

6. Método, de acordo com a reivindicação 1, **caracterizado** pelo fato de que compreende adicionalmente:

o domínio de acesso transmitindo uma solicitação para alocar um recurso adicional que compreende uma região de memória adicional para um sistema operacional de alto nível, HLOS;

a alocação, pelo HLOS, do recurso adicional para o domínio de acesso;

o travamento, pelo domínio de acesso, do recurso adicional contra o acesso por qualquer entidade além do domínio de acesso.

7. Método, de acordo com a reivindicação 6, **caracterizado** pelo fato de que compreende adicionalmente:

o destravamento, pelo domínio de acesso, do recurso adicional contra o acesso por outra entidade; e

o domínio de acesso transmitindo para o HLOS uma indicação de que o recurso adicional está livre.

8. Sistema para o controle de acesso de recursos em um sistema em chip, SoC, (102) **caracterizado** pelo fato de que compreende:

um agente que é executado em um processador (104) e configurado para:

alocar um recurso (112) que compreende uma região de memória para um domínio de acesso (108); e

carregar uma imagem de software associada ao domínio de acesso na região da memória; e

um mecanismo de gerenciamento de confiança (116) configurado para:

travar o recurso contra acesso por qualquer entidade além do domínio de acesso e do mecanismo de gerenciamento de confiança;

autenticar a imagem de software associada ao domínio de acesso; e

inicializar booting do domínio de acesso em resposta a uma autenticação bem-sucedida da imagem de software associada ao domínio de acesso,

em que o domínio de acesso é adicionalmente configurado para, depois do booting, travar o recurso contra acesso pelo mecanismo de gerenciamento de confiança.

9. Sistema, de acordo com a reivindicação 8, **caracterizado** pelo fato de que o agente compreende um sistema operacional de alto nível (110).

10. Sistema, de acordo com a reivindicação 9, **caracterizado** pelo fato de que o domínio de acesso compreende um modem.

11. Sistema, de acordo com a reivindicação 8, **caracterizado** pelo fato de que o domínio de acesso compreende um ou mais barramentos mestre (106).

12. Sistema, de acordo com a reivindicação 11, **caracterizado** pelo fato de que booting compreende um primeiro

barramento mestre do domínio de acesso inicializando booting de um segundo barramento mestre do domínio de acesso, depois que o primeiro barramento mestre completa o booting, o segundo barramento mestre realizando o booting independentemente do mecanismo de gerenciamento de confiança.

13. Sistema, de acordo com a reivindicação 8, **caracterizado** pelo fato de que:

o domínio de acesso é adicionalmente configurado para transmitir uma solicitação de alocação de um recurso adicional compreendendo uma região de memória adicional para um sistema operacional de alto nível, HLOS, (110);

o HLOS é configurado para alocar o recurso adicional no domínio de acesso; e

o domínio de acesso é configurado para travar o recurso adicional contra acesso por qualquer entidade além do domínio de acesso.

14. Sistema, de acordo com a reivindicação 13, **caracterizado** pelo fato de que o domínio de acesso é adicionalmente configurado para:

destravar o recurso adicional contra acesso por outra entidade; e

transmitir para o HLOS uma indicação de que o recurso adicional está livre.

15. Memória legível por computador **caracterizada** pelo fato de que possui instruções, armazenadas na mesma, que quando executadas em um processador, executam o método de controle SoC conforme definido em qualquer uma das reivindicações 1 a 7.

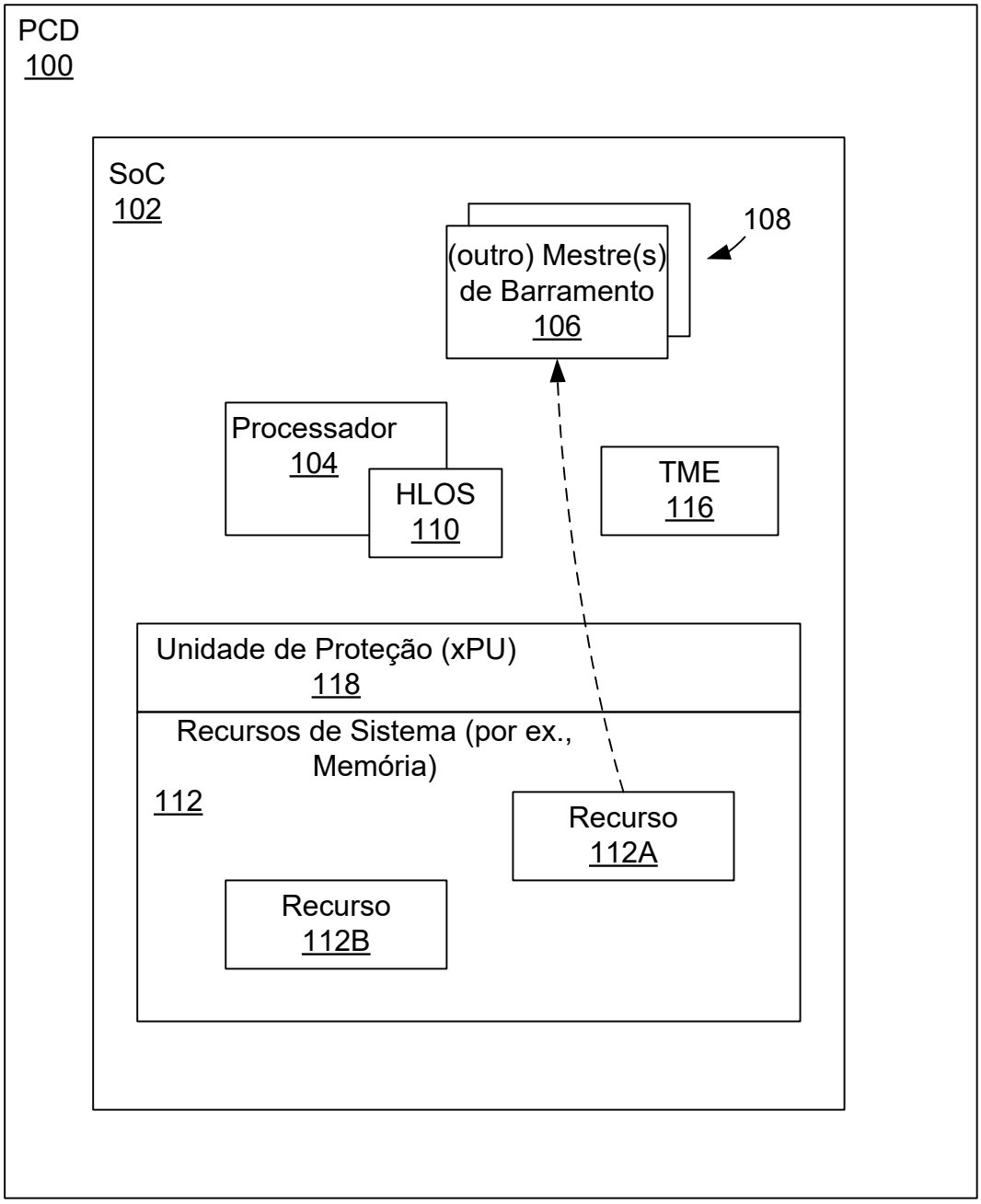
PCD
100
SoC
102
108
(outro) Mestre(s) de Barramento
106
Processador
104
HLOS
110
TME
116
Unidade de Proteção (xPU)
118
Recursos de Sistema (por ex., Memória)
112
Recurso
112A
Recurso
112B

FIG. 1

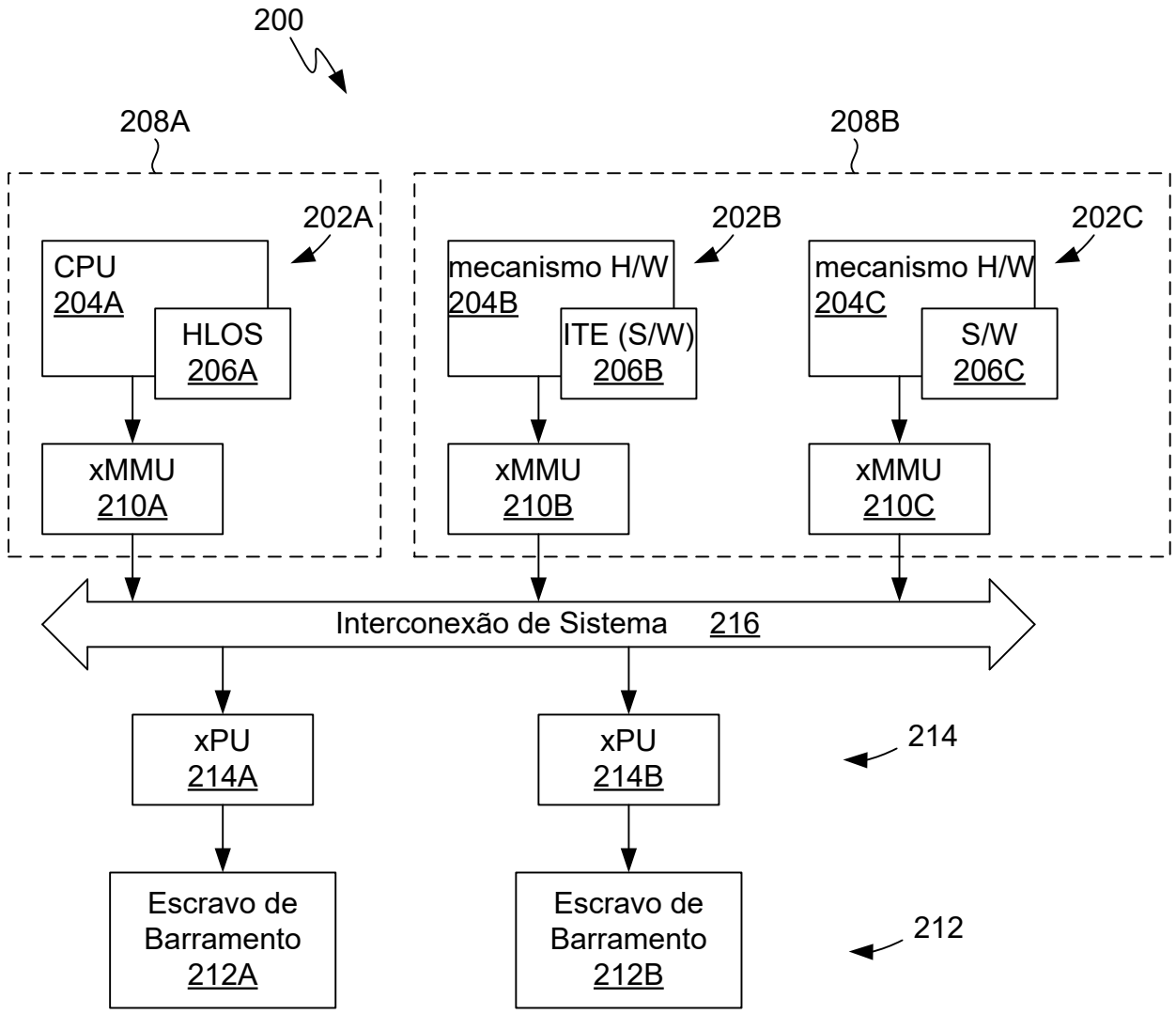
200
208A
208B
202A
202B
202C
CPU
204A
HLOS
206A
mecanismo H/W
204B
ITE (S/W)
206B
mecanismo H/W
204C
S/W
206C
xMMU
210A
xMMU
210B
xMMU
210C
Interconexão de Sistema 216
xPU
214A
xPU
214B
214
Escravo de
Barramento
212A
Escravo de
Barramento
212B
212

FIG. 2

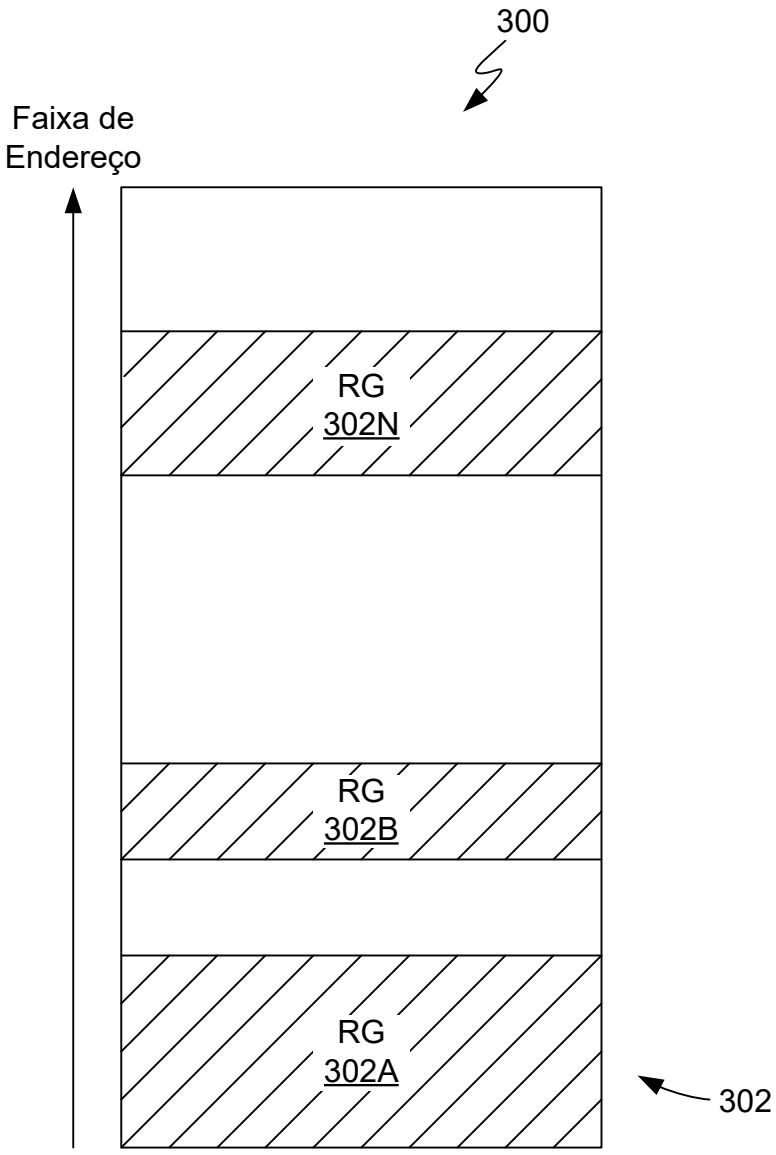

FIG. 3

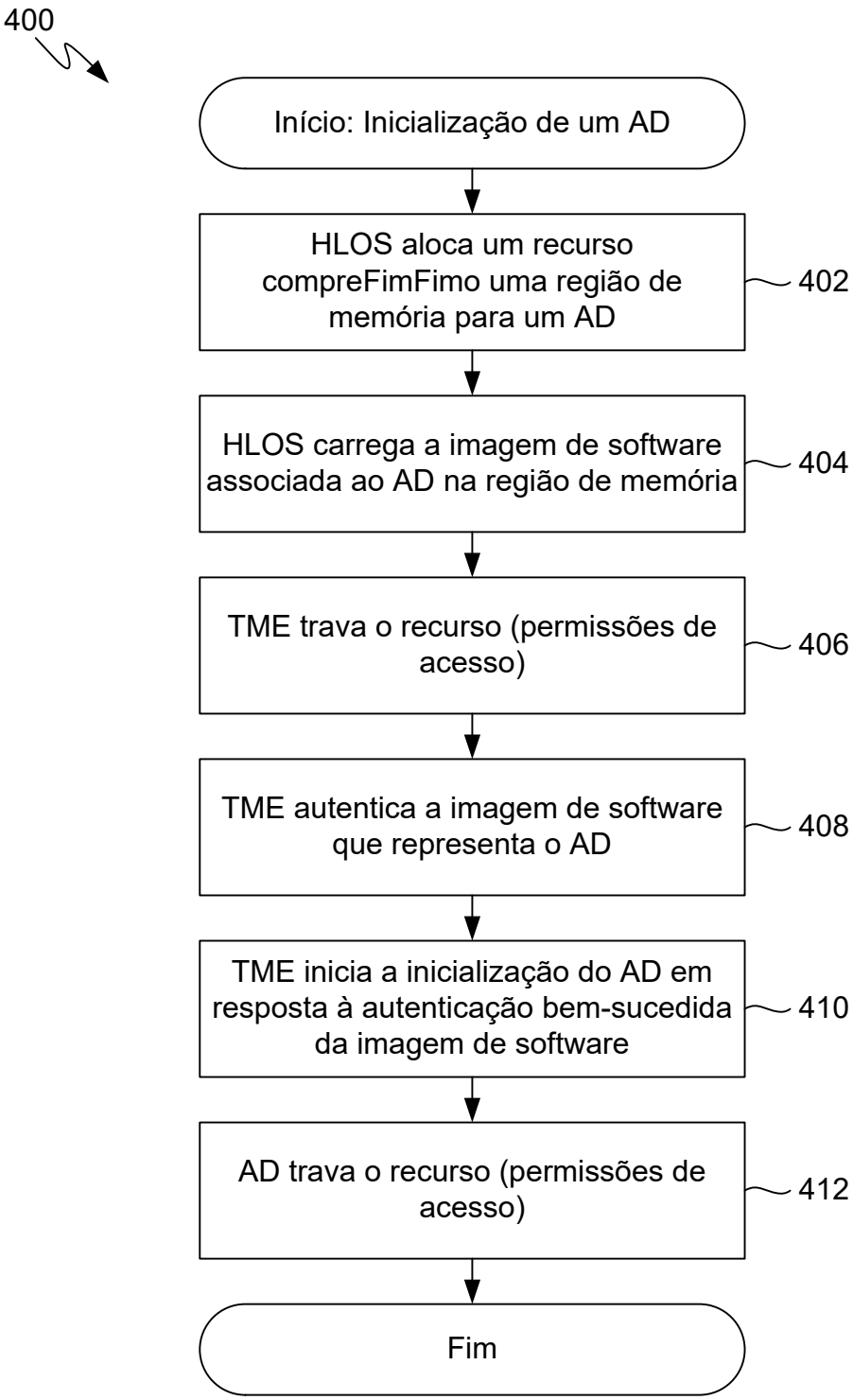
400
Início: Inicialização de um AD
HLOS aloca um recurso compreFimFimo uma região de memória para um AD
402
HLOS carrega a imagem de software associada ao AD na região de memória
404
TME trava o recurso (permissões de acesso)
406
TME autentica a imagem de software que representa o AD
408
TME inicia a inicialização do AD em resposta à autenticação bem-sucedida da imagem de software
410
AD trava o recurso (permissões de acesso)
412
Fim

FIG. 4

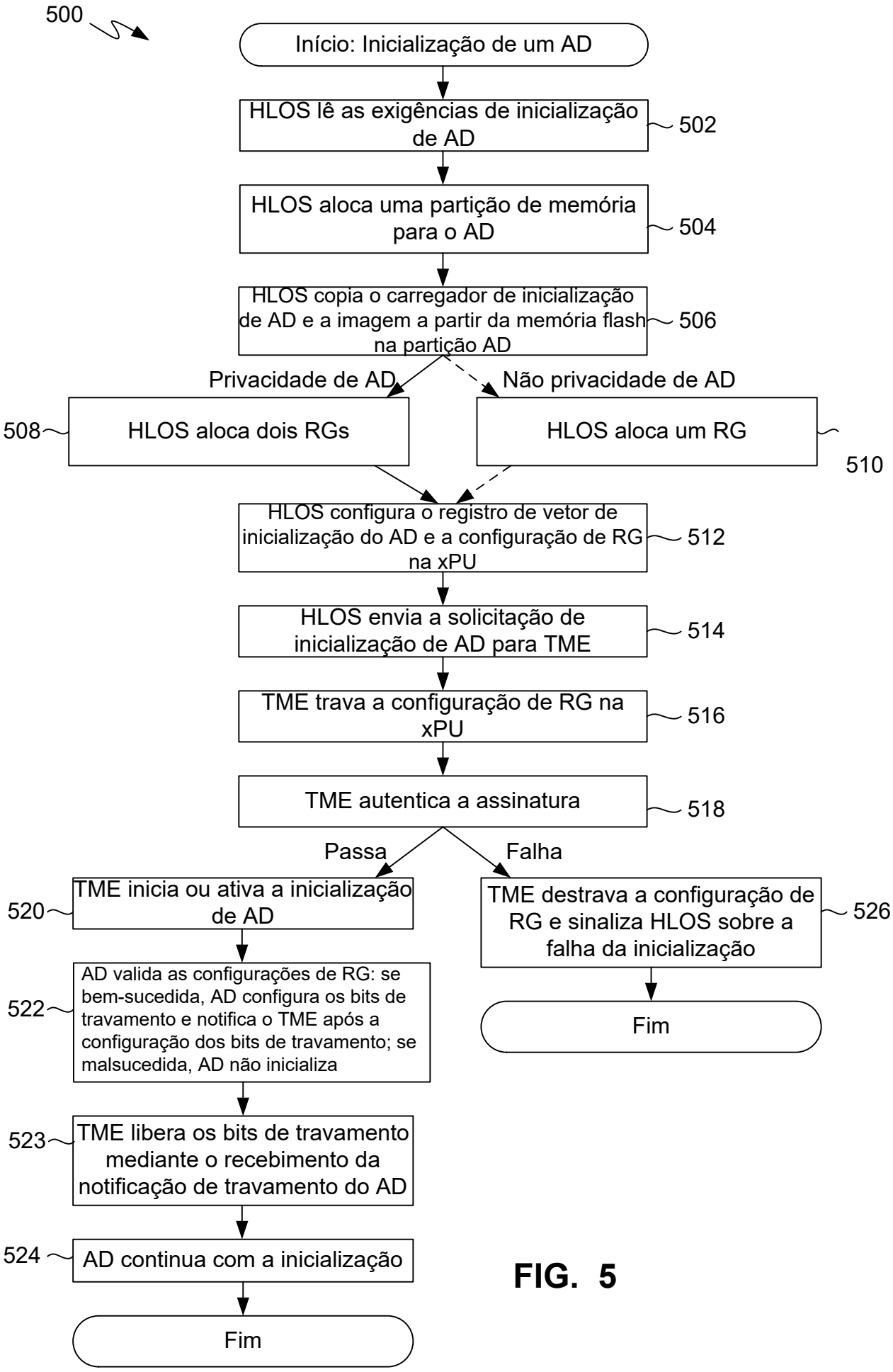
500
Início: Inicialização de um AD
HLOS lê as exigências de inicialização de AD
502
HLOS aloca uma partição de memória para o AD
504
HLOS copia o carregador de inicialização de AD e a imagem a partir da memória flash na partição AD
506
Privacidade de AD
Não privacidade de AD
508
HLOS aloca dois RGs
HLOS aloca um RG
510
HLOS configura o registro de vetor de inicialização do AD e a configuração de RG na xPU
512
HLOS envia a solicitação de inicialização de AD para TME
514
TME trava a configuração de RG na xPU
516
TME autentica a assinatura
518
Passa
Falha
520
TME inicia ou ativa a inicialização de AD
TME destrava a configuração de RG e sinaliza HLOS sobre a falha da inicialização
526
522
AD valida as configurações de RG: se bem-sucedida, AD configura os bits de travamento e notifica o TME após a configuração dos bits de travamento; se malsucedida, AD não inicializa
Fim
523
TME libera os bits de travamento mediante o recebimento da notificação de travamento do AD
524
AD continua com a inicialização
Fim

FIG. 5

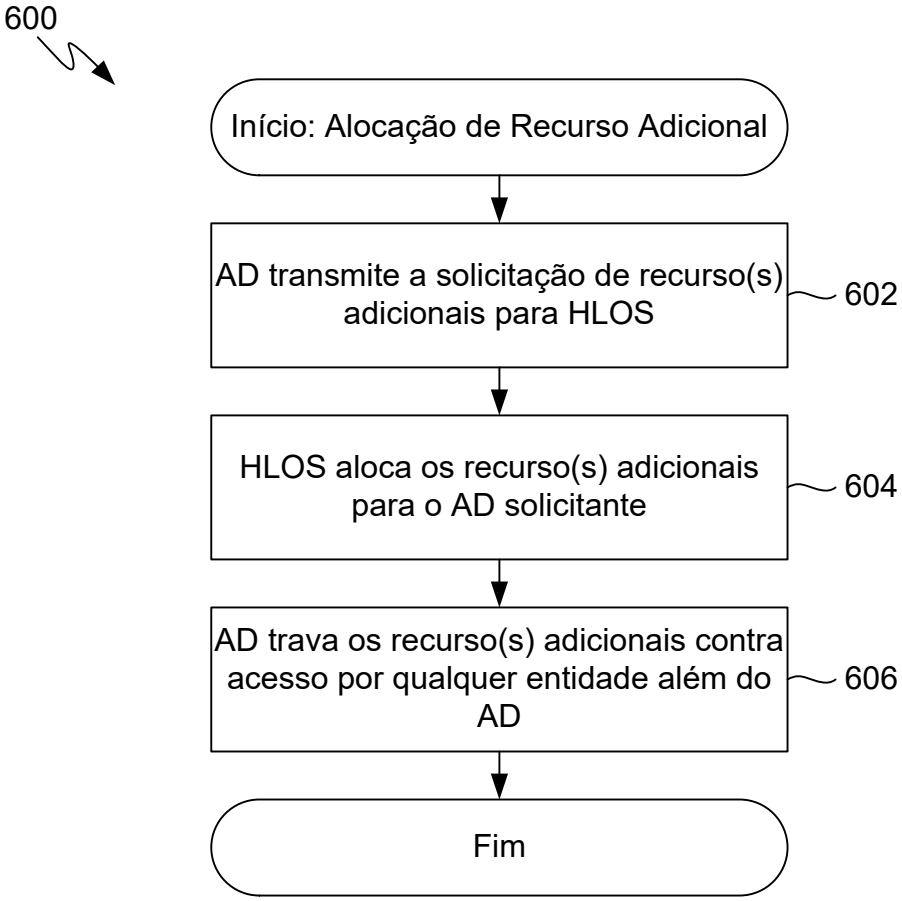
600
Início: Alocação de Recurso Adicional
AD transmite a solicitação de recurso(s) adicionais para HLOS
602
HLOS aloca os recurso(s) adicionais para o AD solicitante
604
AD trava os recurso(s) adicionais contra acesso por qualquer entidade além do AD
606
Fim

FIG. 6

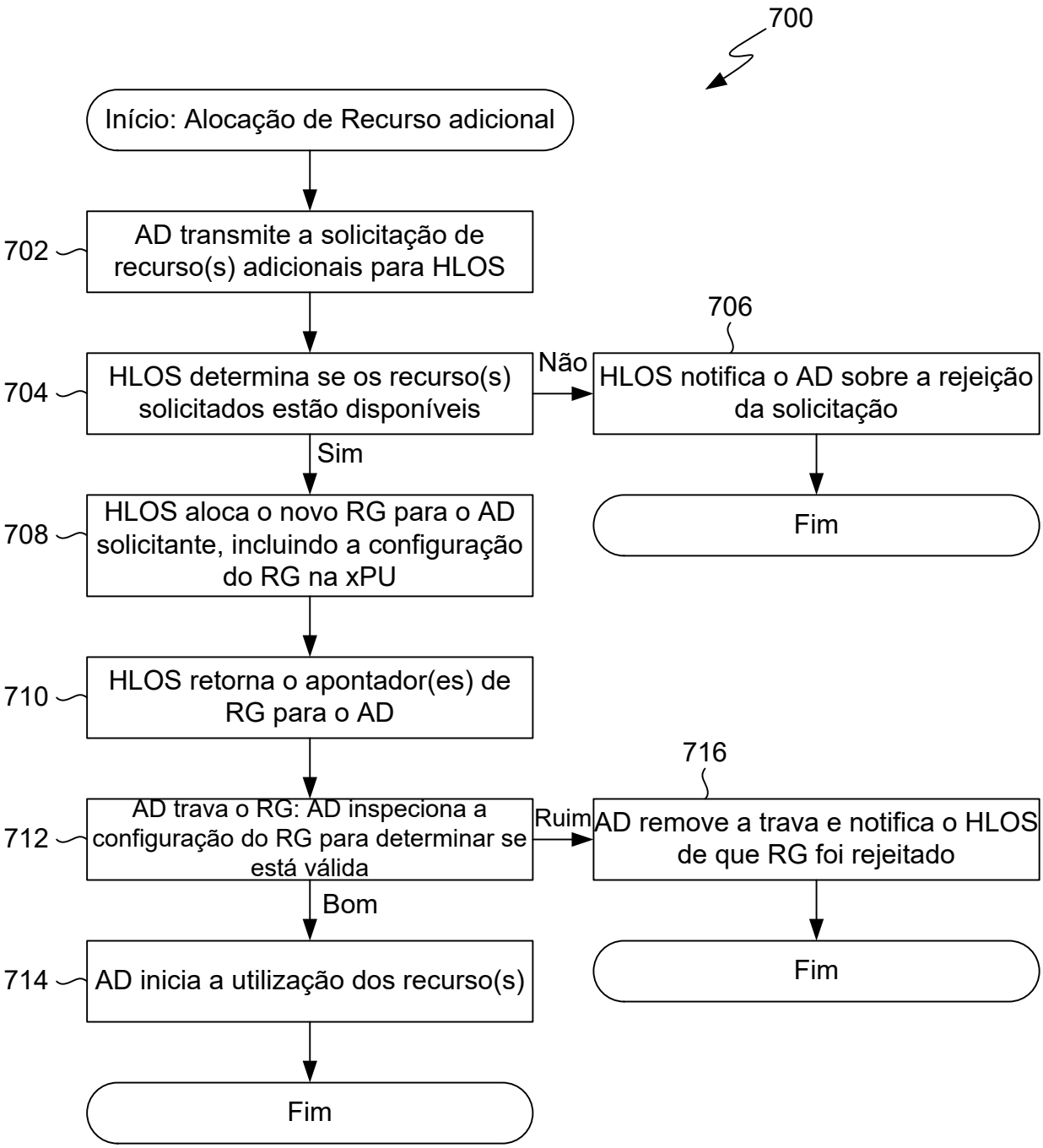
700
Início: Alocação de Recurso adicional
702
AD transmite a solicitação de recurso(s) adicionais para HLOS
704
HLOS determina se os recurso(s) solicitados estão disponíveis
Não
706
HLOS notifica o AD sobre a rejeição da solicitação
Fim
Sim
708
HLOS aloca o novo RG para o AD solicitante, incluindo a configuração do RG na xPU
710
HLOS retorna o apontador(es) de RG para o AD
712
AD trava o RG: AD inspeciona a configuração do RG para determinar se está válida
Ruim
716
AD remove a trava e notifica o HLOS de que RG foi rejeitado
Fim
Bom
714
AD inicia a utilização dos recurso(s)
Fim

FIG. 7

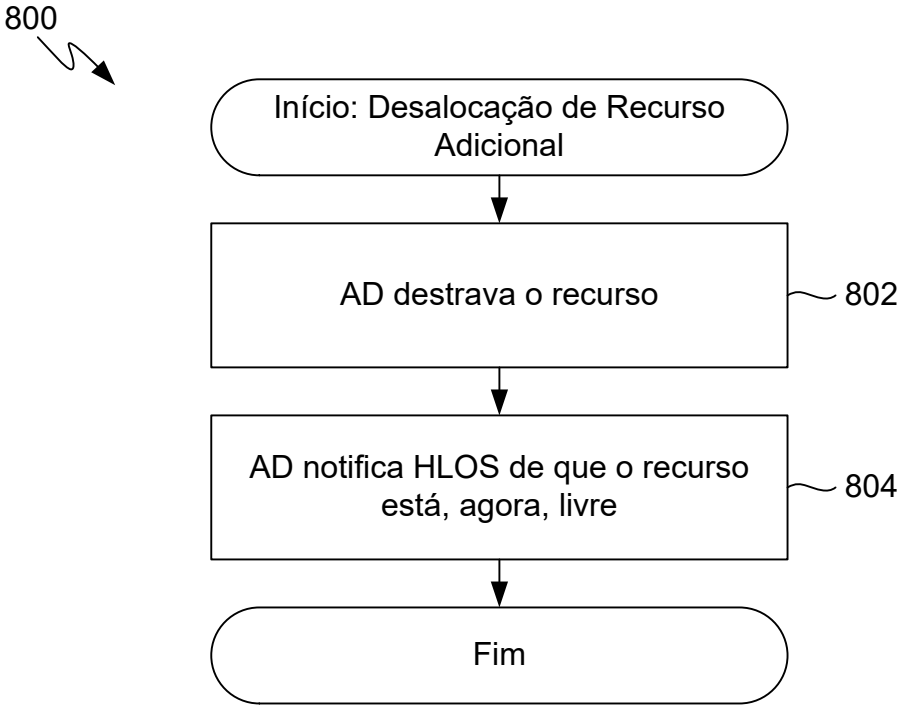
800
Início: Desalocação de Recurso Adicional
AD destrava o recurso
802
AD notifica HLOS de que o recurso está, agora, livre
804
Fim

FIG. 8

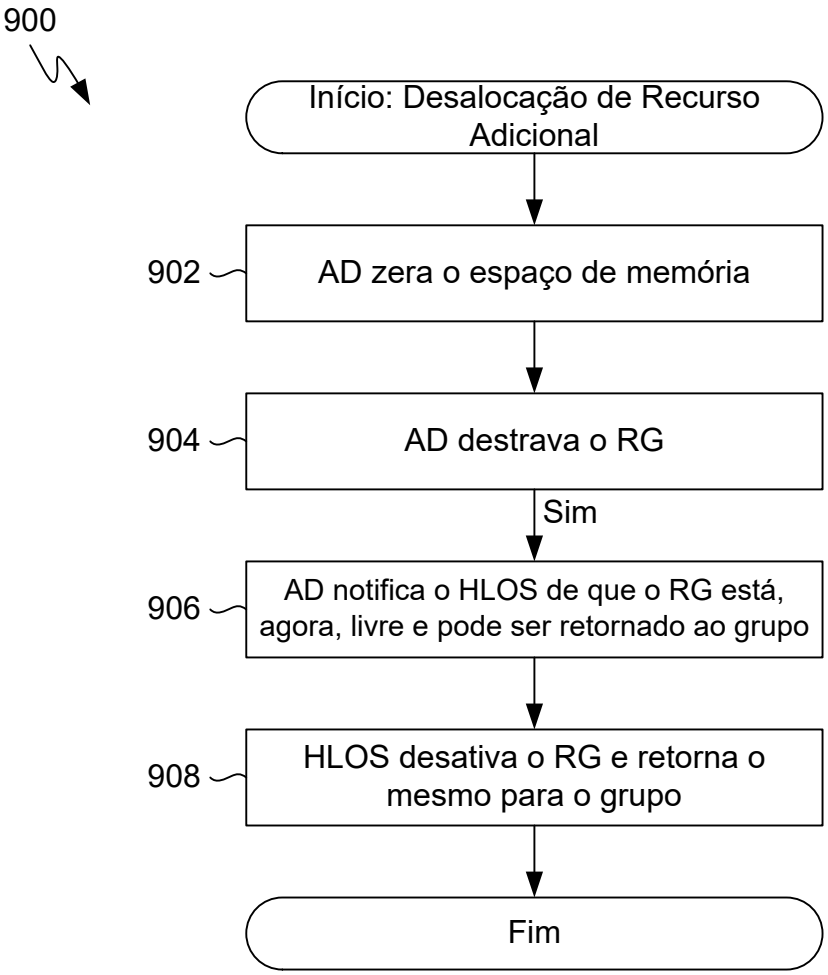
900
Início: Desalocação de Recurso Adicional
902
AD zera o espaço de memória
904
AD destrava o RG
Sim
906
AD notifica o HLOS de que o RG está, agora, livre e pode ser retornado ao grupo
908
HLOS desativa o RG e retorna o mesmo para o grupo
Fim

FIG. 9

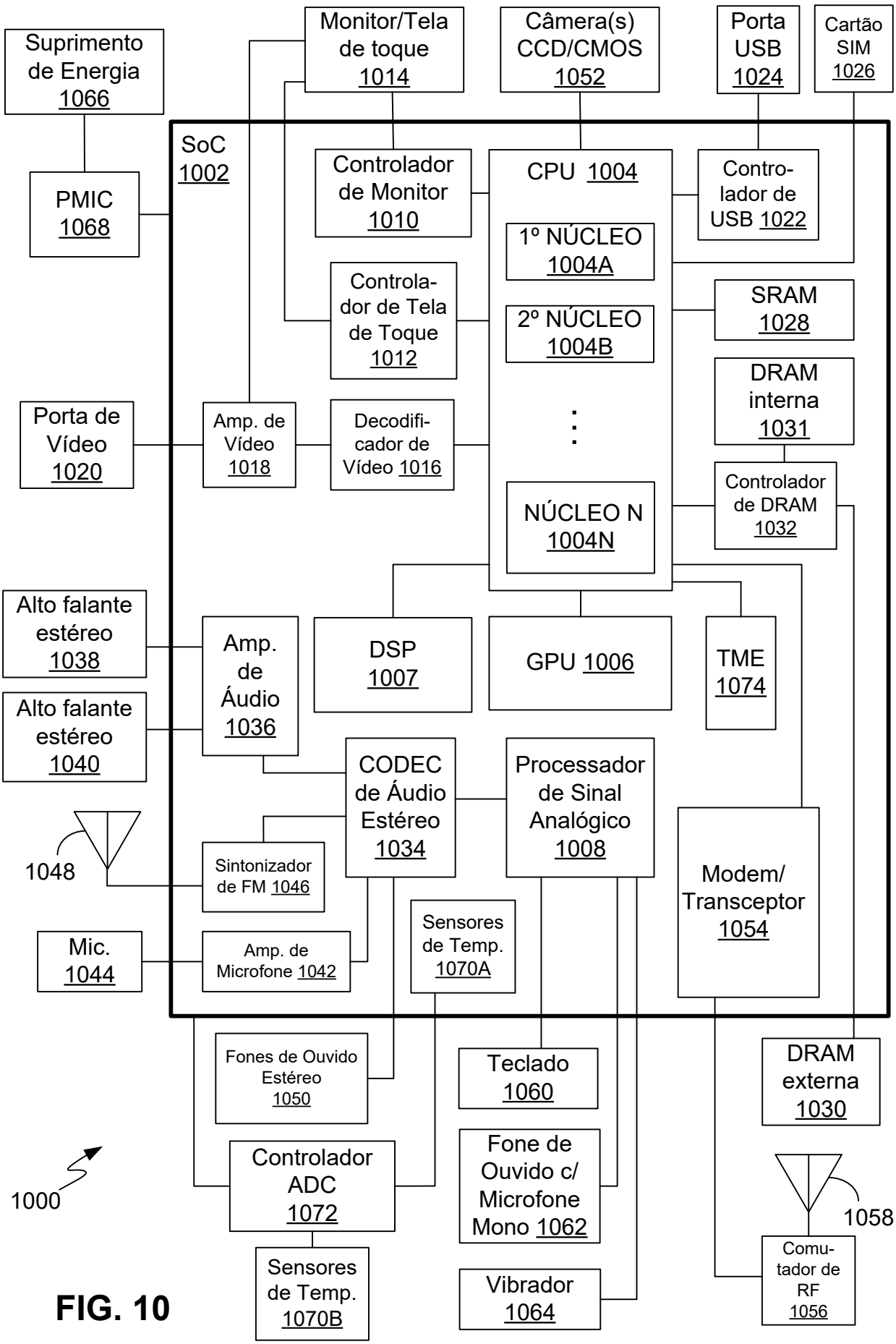
Suprimento de Energia 1066
Monitor/Tela de toque 1014
Câmera(s) CCD/CMOS 1052
Porta USB 1024
Cartão SIM 1026
SoC 1002
PMIC 1068
Controlador de Monitor 1010
CPU 1004
Contro-lador de USB 1022
1º NÚCLEO 1004A
Controla-dor de Tela de Toque 1012
2º NÚCLEO 1004B
SRAM 1028
DRAM interna 1031
Porta de Vídeo 1020
Amp. de Vídeo 1018
Decodifi-cador de Vídeo 1016
NÚCLEO N 1004N
Controlador de DRAM 1032
Alto falante estéreo 1038
Amp. de Áudio 1036
DSP 1007
GPU 1006
TME 1074
Alto falante estéreo 1040
CODEC de Áudio Estéreo 1034
Processador de Sinal Analógico 1008
1048
Sintonizador de FM 1046
Modem/ Transceptor 1054
Sensores de Temp. 1070A
Mic. 1044
Amp. de Microfone 1042
Fones de Ouvido Estéreo 1050
Teclado 1060
DRAM externa 1030
1000
Controlador ADC 1072
Fone de Ouvido c/ Microfone Mono 1062
1058
Comu-tador de RF 1056
Sensores de Temp. 1070B
Vibrador 1064

FIG. 10

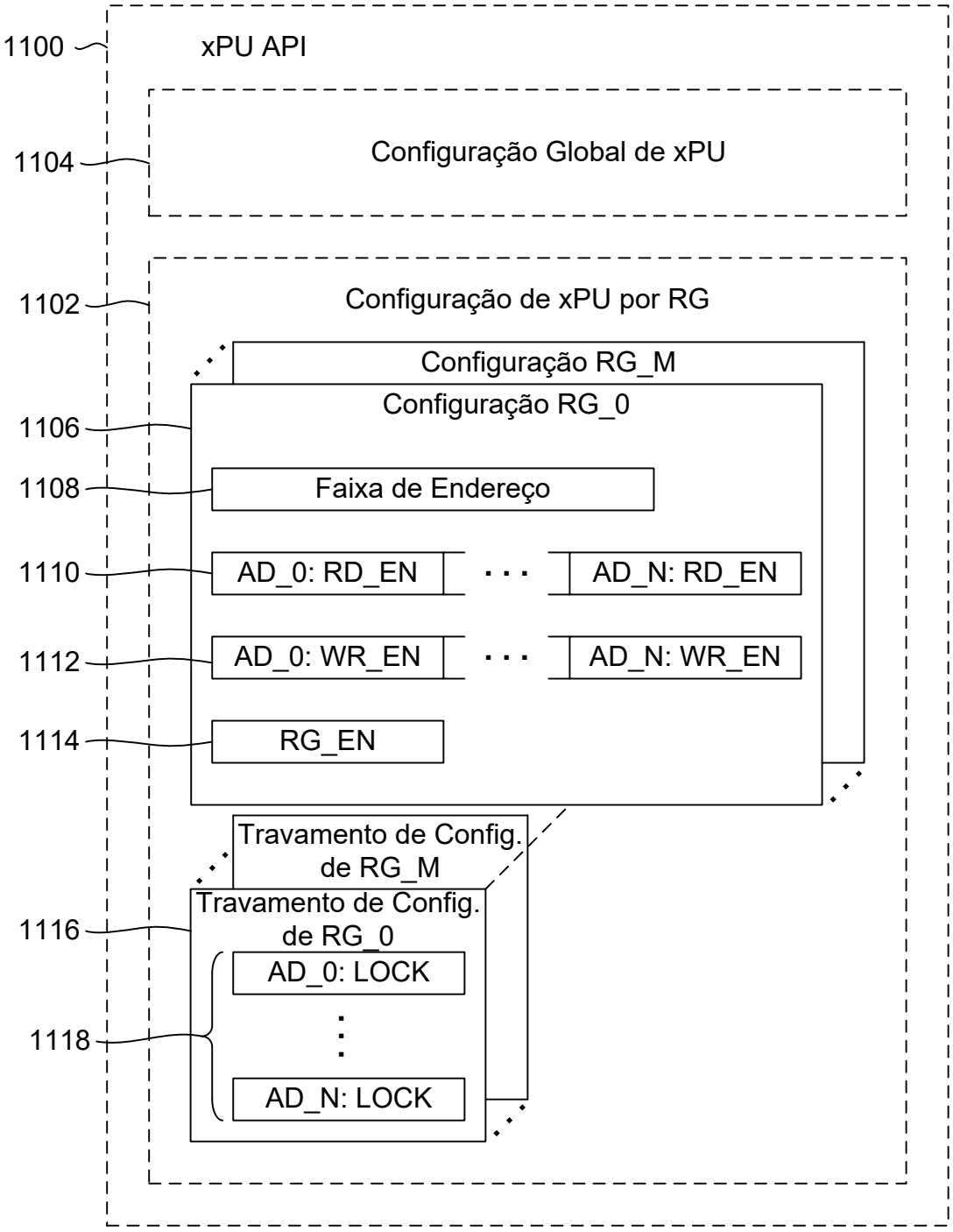
1100
xPU API
1104
Configuração Global de xPU
1102
Configuração de xPU por RG
Configuração RG_M
Configuração RG_0
1106
1108
Faixa de Endereço
1110
AD_0: RD_EN
AD_N: RD_EN
1112
AD_0: WR_EN
AD_N: WR_EN
1114
RG_EN
Travamento de Config. de RG_M
Travamento de Config. de RG_0
1116
AD_0: LOCK
1118
AD_N: LOCK

FIG. 11