



República Federativa do Brasil
Ministério do Desenvolvimento, Indústria
e do Comércio Exterior
Instituto Nacional da Propriedade Industrial.

(21) **PI0710171-6 A2**



(22) Data de Depósito: 17/04/2007
(43) Data da Publicação: 23/08/2011
(RPI 2120)

(51) *Int.Cl.:*
G06F 12/08 2006.01
G06F 12/10 2006.01

(54) Título: **CACHE DE INSTRUÇÕES MARCADA VIRTUALMENTE COM COMPORTAMENTO MARCADO FISICAMENTE**

(30) Prioridade Unionista: 31/08/2006 US 11/468,850, 19/04/2006 US 60/793,015, 19/04/2006 US 60/793,016, 31/08/2006 US 11/468,850

(73) Titular(es): Qualcomm Incorporated

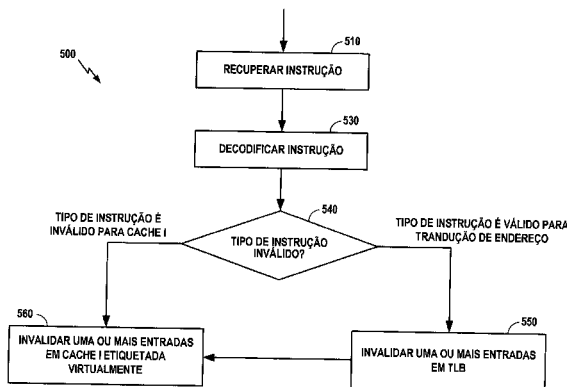
(72) Inventor(es): Daren Eugene Streett, Rodney Wayne Smith, Thomas Andrew Sartorius

(74) Procurador(es): Montauray Pimenta, Machado & Lloce

(86) Pedido Internacional: PCT US2007066802 de 17/04/2007

(87) Publicação Internacional: WO 2007/124307 de 01/11/2007

(57) **Resumo:** CACHE DE INSTRUÇÕES MARCADA VIRTUALMENTE COM COMPORTAMENTO MARCADO FISICAMENTE É descrito um sistema de cache de instruções possuindo uma cache de instruções marcada virtualmente que, do ponto de vista de um programa de software, opera como se fosse uma cache de instruções marcada fisicamente. O sistema de cache de instruções inclui também um dispositivo para endereçar translação que responde a uma instrução de invalidar endereçar translação e um circuito lógico de controle. O circuito lógico de controle está configurado para invalidar uma entrada na cache de instruções marcada virtualmente em resposta à instrução de invalidar endereçar translação.





PI0710171-6

**"CACHE DE INSTRUÇÕES MARCADA VIRTUALMENTE COM COMPORTAMENTO
MARCADO FISICAMENTE"**

PEDIDOS CORRELACIONADOS

O presente pedido de patente reivindica a
5 prioridade do Pedido Provisório de Patente U.S. Nº de Série
60/793 016, depositado em 19 de abril de 2006, e do Pedido
Provisório de Patente U.S. Nº de Série 60/793 015,
depositado em 19 de abril de 2006, ambos aqui incorporados
em sua totalidade pela presente referência.

10 CAMPO

A presente invenção está de um modo geral
relacionada a sistemas de processamento e mais
especificamente a sistemas de cache marcada virtualmente.

FUNDAMENTOS

15 Os sistemas de processador comum utilizam o
conceito de tradução de endereços em um encadeamento
(pipeline) de processador para modificar o endereço de uma
página de dados ou instruções de seu endereço virtual de
armazenamento para seu endereço de armazenamento real,
20 físico. Um dispositivo de tradução de endereço utiliza um
buffer de previsão de tradução (TLB) que armazena
convencionalmente várias entradas, em que cada entrada
inclui um identificador de espaço de aplicativo (ASID),
também designado como um identificador de processo (PID),
25 um marcador (tag) de endereço virtual (VT) e um número de
página física (PPN). Dessa forma, o buffer de previsão de
tradução contém o mapeamento entre um endereço virtual e um
endereço físico. Em sua forma mais simples, a linha do
processador recebe como entrada um endereço virtual de uma
30 instrução, compara uma parte do endereço virtual com as
entradas no TLB para encontrar uma entrada correspondente e
substitui a parte do endereço virtual pelo número de página
física associada à entrada correspondente para formar um
endereço físico. Um dado mapeamento de um endereço físico

para um endereço virtual de um modo geral cobre uma faixa do espaço de endereços físico e virtual, tal faixa sendo designada como uma "página", com o tamanho de uma página sendo de um modo geral maior do que o tamanho de uma linha de cache em uma cache de instruções, de tal forma que várias linhas de cache podem estar todas associadas à mesma página do espaço de endereço virtual ou físico.

Os sistemas de processador comum incluem também uma ou mais caches de instruções que armazenam instruções recém usadas em memória rápida "on-chip" para minimizar o retardo resultante da recuperação de instruções a partir de memórias "off-chip" mais lentas. Uma cache de instruções pode estar indexada para pesquisa eficiente de uma entrada. O termo "indexado", tal como usado com relação a caches de instruções, significa um conjunto de bits em um endereço virtual ou um endereço físico que é utilizado para especificar um conjunto em uma cache de instruções associativo a conjuntos ou uma fila em uma cache de instruções mapeada diretamente. Uma cache de instruções indexada por bits dentro de um endereço virtual é conhecida como uma cache indexada virtualmente. Uma cache de instruções indexada por bits dentro de um endereço físico é conhecida como uma cache indexada fisicamente.

As entradas em uma cache de instruções podem estar marcadas com uma chave a qual é comparada com uma parte de um endereço físico ou uma parte de um endereço virtual. Uma cache de instruções marcada com uma parte de um endereço físico é conhecida como uma cache de instruções marcada fisicamente. Uma cache de instruções marcada com uma parte de um endereço virtual é conhecida como uma cache de instruções marcada virtualmente. Um projetista de um processador escolhe como uma cache de instruções será indexada e marcada. As caches de instruções convencionais podem ser indexadas virtualmente e marcadas virtualmente (VIVT), indexadas virtualmente e marcadas fisicamente

(VIPT), ou indexadas fisicamente e marcadas fisicamente (PIPT). Através de tradução de endereços, um endereço virtual é convertido para um endereço físico, o qual pode ser utilizado para procurar por uma entrada em uma cache de instruções marcada fisicamente, ou para acessar uma memória real no caso de uma perda ou erro de cache de instruções.

Além de um marcador de endereço virtual, as caches de instruções marcadas virtualmente convencionais comumente incluem um ASID ou um marcador PID. Tal marcador permite ao sistema processador diferenciar se uma entrada de endereço é válida para o processo de software ativo. Como exemplo, caso o sistema processador esteja executando instruções associadas a um processo de software possuindo o ASID "x", uma comparação de cache bem sucedida iria requerer a correspondência tanto do marcador ASID como do marcador de instrução de endereço virtual. Devido a várias situações, tais como a troca de um ASID ativo, isto é, a substituição de um processo que usa um dado valor de ASID por um novo processo usando o mesmo valor de ASID, as caches de instruções marcadas virtualmente podem possuir uma ou mais entradas que se tornam obsoletas e não são mais válidas. Dado que acertos ou "hits" de cache economizam tempo do processador para recuperar instruções a partir da memória, os projetistas de sistemas processadores desejam coerência da cache, uma cache cujo conteúdo reflete instruções válidas, não obsoletas. Para manter a coerência de cache, as entradas em uma cache são comumente invalidadas em resposta a instruções para invalidar cache de instruções comuns emitidas por um aplicativo de software. A quantidade de gerenciamento de uma cache de instruções requerida pelo aplicativo de software depende de se a cache de instruções é marcada virtualmente ou marcada fisicamente. Dado que um ou mais endereços virtuais podem estar mapeados para um dado endereço físico, podem ocorrer

problemas de alias e sinônimo em uma cache de instruções marcada virtualmente convencional.

O problema de alias ocorre quando duas ou mais entradas de cache I contendo diferentes combinações de endereço virtual/ASID são mapeadas para o mesmo endereço físico. Tais combinações diferentes podem ocorrer devido às três razões que se seguem. Primeira, o endereço virtual de duas ou mais entradas de cache I podem ser literalmente diferentes porém fazem parte do mesmo processo de software possuindo o mesmo valor do campo ASID. Segunda, os endereços virtuais de duas ou mais entradas de cache I podem ser iguais, porém estarem associados a diferentes processos de software e, portanto, associados a diferentes valores do campo ASID. Terceira, os endereços virtuais de duas ou mais entradas de cache I podem ser todos diferentes e estar associados a diferentes processos de software, estando, portanto, associados a diferentes valores de ASID. Em qualquer dos três casos, as diferentes combinações podem ser mapeadas para o mesmo endereço físico. Como resultado de tal problema de alias, as caches de instruções marcadas virtualmente convencionais impõem uma carga sobre um aplicativo de software para emitir instruções para invalidar caches I para mais situações do que o requerido por uma cache de instruções marcada fisicamente. Como exemplo, um aplicativo de software escrito para caches de instruções marcadas virtualmente teria que emitir instruções para invalidar caches de instruções para invalidar cada endereço virtual na cache de instruções que possa estar associada a um endereço físico que se tornou por si não válido ou que tenha sido modificado.

O problema de sinônimos diz respeito a duas ou mais entradas de cache I que estão localizadas em diferentes índices virtuais da cache I, mas que estão associadas ao mesmo endereço físico. O problema de sinônimos pode ocorrer em caches I indexadas virtualmente

convencionais independentemente de se tais caches I são marcadas virtual ou fisicamente.

As caches de instruções marcadas virtualmente também impõem outra carga adicional sobre o software por
5 invalidar entradas de cache I, em comparação a caches de instruções marcadas fisicamente. Especificamente, quando o mapeamento para a combinação de um dado ASID e página do espaço de endereço virtual é modificado de um endereço físico para outro, não mais são válidas todas as entradas
10 de cache I em uma cache I marcada virtualmente que estão associadas a tal combinação de ASID e página do espaço de endereço virtual, dado que a cache I pode ter armazenado o conteúdo dos locais de memória física anteriores associados ao mapeamento anterior, em lugar do conteúdo dos novos
15 locais de memória física associados ao novo mapeamento. Tal permanece verdadeiro mesmo que o conteúdo real dos locais de memória física antigos e novos possa não ter mudado. Portanto, um software escrito para gerenciar uma cache de instruções marcada virtualmente deve executar operações de
20 invalidar uma cache de instruções para invalidar todas as entradas de cache de instruções que possam estar associadas à combinação de ASID e página do espaço de endereço virtual para o qual foi modificado o mapeamento.

Em um processador com uma cache de instruções
25 marcada virtualmente, a tradução de endereços pode ser efetuada em um encadeamento em paralelo com a consulta a instruções provenientes da cache de instruções marcada virtualmente. Dessa forma, tal paralelismo propicia uma poderosa potência, frequência e capacidade de transmissão
30 de instruções quando são utilizadas caches de instruções marcadas virtualmente. Portanto, existe uma demanda por um sistema e métodos de caches de instruções que propiciem as vantagens de uma cache de instruções marcada virtualmente convencional, eliminando porém a carga adicional exercida
35 sobre aplicativos de software por caches de instruções

marcadas virtualmente convencionais com referência ao gerenciamento de entradas e que não imponha demandas adicionais a aplicativos de software.

SUMÁRIO

5 Um aspecto da invenção descreve um sistema de cache de instruções que apresenta as vantagens de uma cache de instruções marcada virtualmente e de uma cache de instruções marcada fisicamente. Tal modalidade da presente invenção utiliza uma cache de instruções marcada
10 virtualmente que, do ponto de vista de um programa de software, opera como uma cache de instruções marcada fisicamente.

Outro aspecto da invenção descreve que endereços virtuais ou partes dos mesmos são utilizados como
15 marcadores tanto em um dispositivo de tradução de endereços como em uma cache de instruções marcada virtualmente. Em tal modalidade, instruções não válidas direcionadas ao dispositivo de tradução de endereço podem ser adicionalmente utilizadas para invalidar entradas na cache
20 de instruções marcada virtualmente.

Em outro aspecto da invenção, um sistema de cache de instruções inclui uma cache de instruções marcada virtualmente, um dispositivo para tradução de endereços e um circuito lógico de controle. O dispositivo para tradução
25 de endereços responde a uma instrução de invalidar a tradução de endereço. O circuito lógico de controle está configurado para invalidar uma entrada na cache de instruções marcada virtualmente em resposta à instrução de invalidar a tradução de endereço.

30 Em mais outro aspecto, é descrito um método para que uma cache de instruções marcada virtualmente para operar tal como se ele fosse uma cache de instruções marcada fisicamente do ponto de vista de um programa de software. O método inclui receber uma instrução de
35 invalidar uma tradução de endereço e invalidar uma entrada

em uma cache de instruções marcada virtualmente em resposta à instrução de invalidar tradução de endereço.

Em mais outro aspecto, é descrito um circuito lógico de controle para ordenar a uma cache de instruções marcada virtualmente a operar como se ele fosse uma instrução marcada fisicamente. O circuito lógico de controle inclui uma primeira entrada para receber uma instrução de invalidar uma tradução de endereço e um dispositivo para gerar sinais de controle para invalidar uma entrada na cache de instruções marcada virtualmente em resposta à instrução de invalidar a tradução de endereço.

Deve ficar claro que outras modalidades da presente invenção ficarão prontamente claras para os técnicos na área através da descrição detalhada que se segue, em que várias modalidades da invenção são apresentadas e descritas como ilustração. Como será notado, a invenção pode assumir outras modalidades e seus diversos detalhes podem ser modificados com relação a vários outros pontos, sempre sem constituir um afastamento da presente invenção. Assim sendo, os desenhos e a descrição detalhada devem ser considerados como sendo de natureza ilustrativa e não restritiva.

BREVE DESCRIÇÃO DOS DESENHOS

A Figura 1 é um diagrama de blocos funcional de uma modalidade de um processador.

A Figura 2 é uma descrição mais detalhada do sistema de cache de instruções da Figura 1.

A Figura 3 é um diagrama de blocos ilustrando a invalidação seletiva de um conjunto de entradas de cache I marcadas virtualmente que poderiam ser associadas a um endereço físico em uma cache I mapeada diretamente.

A Figura 4 é um diagrama de blocos ilustrando a invalidação seletiva de um conjunto de entradas de cache I virtualmente marcadas que poderiam estar associadas a um

endereço físico em uma cache I associativa de conjunto 2-via.

A Figura 5 é um fluxograma ilustrando um método para que uma cache de instruções marcada virtualmente opere como se ele fosse uma cache de instruções marcada fisicamente do ponto de vista de um programa de software.

DESCRIÇÃO DETALHADA

A descrição detalhada apresentada a seguir em conjunto com os desenhos anexos tenciona ser uma descrição de várias modalidades da presente invenção, e não a representar as únicas modalidades em que a presente invenção pode ser praticada. A descrição detalhada inclui detalhes específicos com o propósito de prover uma completa compreensão da presente invenção. No entanto, ficará claro para os técnicos na área que a presente invenção pode ser praticada sem o recurso a tais detalhes específicos. Em alguns casos, estruturas e componentes bem conhecidos são apresentados em forma de diagrama de blocos de modo a evitar obscurecer desnecessariamente os conceitos da presente invenção.

A Figura 1 apresenta um diagrama de blocos funcional de um processador 100. O processador 100 executa instruções em um encadeamento 120 de execução de instruções de acordo com o circuito lógico de controle 110. O processador 100 pode ser qualquer tipo de componente de processador linear, incluindo, por exemplo, um microprocessador, um processador de sinais digitais (DSP), uma ponte, uma lógica programável, uma lógica de transistores ou de porta individual, ou qualquer outro componente de processamento de informações. Em algumas modalidades, o encadeamento 120 pode ser um esquema super escalar, com múltiplos encadeamentos paralelos. O encadeamento 120 inclui vários registradores ou latches 160a a d, organizados em estágios de linha, e uma ou mais unidades de execução 180. Apesar de estarem representados

cinco estágios de linha na Figura 1, será notado pelos técnicos na área que um número de estágios de linha maior ou menor que os estágios de linha ilustrados podem ser incluídos no processador 100. Um arquivo de registrador de propósito geral (GPR) 130 provê registradores constituindo o topo da hierarquia de memória. O processador 100 inclui um sistema de cache de instruções 127 que inclui uma cache de instruções marcada virtualmente (cache I ou I-cache) 122, um buffer de previsão de tradução (TLB) 128 e um circuito de controle 110.

O encadeamento 120 recupera instruções a partir da cache I 122 provendo um endereço virtual para a cache I 122. A tradução de endereço de memória é gerenciada pelo TLB 128. Em paralelo ao provimento de um endereço virtual para a cache I 122, a linha 120 provê o endereço de instrução virtual para o TLB 128 para determinar o endereço físico correspondente que seria utilizado para acessar a memória 32 através da interface de memória 30 caso um endereço virtual não seja encontrado (perda de cache) na cache I 122. O circuito de lógica de controle 110 recebe instruções de invalidação provenientes da linha 120 e gerencia a invalidação de uma ou mais entradas na cache I 122 e no TLB 128. A estrutura e operação do sistema de cache de instruções 127 serão mais completamente descritas em conexão com a descrição da Figura 2.

Os dados são acessados a partir de uma cache de dados (cache D) 126, com a tradução de endereço de memória e permissões gerenciadas pelo buffer de previsão de tradução (TLB) 128. Apesar de a Figura 1 apresentar o TLB 128 como um TLB integrado, em várias modalidades, o TLB 128 pode ser segregado em múltiplos TLBs em que um está dedicado a transladar acessos à cache D 126 e o outro está dedicado a transladar acessos à cache I 122. As perdas na cache I 122 e/ou na cache D 126 causam um acesso à memória principal (off-chip) 150, sob o controle de uma interface

de memória 140. Tais acessos à memória principal utilizam o endereço físico recuperado a partir do TLB 128.

O processador 100 pode incluir uma interface de entrada/saída (I/O) 134, que controla o acesso a vários dispositivos periféricos 136 e 138. Os técnicos na área 5 notarão que são possíveis inúmeras variações do processador 100. Como exemplo, o processador 100 pode incluir uma cache de segundo nível (L2) para uma das, ou para ambas as, caches I 122 e D 126. Além disso, um ou mais dos blocos 10 funcionais apresentados no processador 100 pode ser omitido de uma modalidade específica.

A Figura 2 é uma representação mais detalhada do sistema de cache de instruções 127 da Figura 1. Em tal modalidade exemplar, o TLB 128 está configurado com três 15 colunas 210a a 210c. A coluna 210a contém os identificadores espaciais de aplicativo (ASIDs) associados a processos de software. O registrador 215 contém o ASID do processo de software atualmente em execução, o qual pode estar armazenado em um registrador dedicado ou no arquivo 20 GPR 130. A coluna 210b contém marcadores virtuais TLB. Um marcador virtual TLB consiste de uma parte de um endereço virtual. Como exemplo, o endereço virtual 205 é um endereço de 32 bits armazenado em um registrador tal como um registrador de contador de programa (PC). Neste exemplo, as 25 posições de bits 31 a 12 definem a parte do endereço virtual que define o marcador virtual TLB.

A coluna 210c contém números de página física (PPN) de 20 bits correspondentes a uma combinação de marcador virtual ASID/TLB. Um número de página física é 30 retornado a partir do TLB 128 quando um ASID ativo (associado a um processo de software em execução) e um marcador virtual TLB proveniente de um endereço virtual 205 estão de acordo com a combinação do marcador ASID e marcador virtual TLB armazenados em uma fileira do TLB 128. 35 O PPN retornado é concatenado com o offset de página para

definir o endereço físico correspondente. Em particular, durante a concatenação, o PPN retornado é colocado nas posições de bits 31 a 12, enquanto a parte de offset de página do endereço virtual continua a ocupar as posições de bits 11 a 0.

Nesta modalidade exemplar, a cache I marcada virtualmente 122 é uma cache de mapeamento direto de 32 KB com 1024 "conjuntos de 1 linha" e linhas de 32 bytes. Em tal modalidade, a cache I marcada virtualmente 122 está configurado com quatro colunas 220a a 220d. A coluna 220a contém os identificadores espaciais de aplicativo (ASID) associados a processos de software. A coluna 220b contém os marcadores virtuais de cache I (I\$). A coluna 220c contém uma linha de instrução de 32 bytes associada a uma combinação específica de marcador virtual de cache I e ASID. Opcionalmente, a coluna 220d contém indicadores indicando sob quais condições foi armazenada uma entrada associada na cache I de modo a limitar o escopo da invalidação de entradas subsequentes.

O endereço virtual 205 é mostrado apresentando dois significados superpostos para suas posições de bits. Quando o endereço virtual 205 é utilizado pela cache I marcada virtualmente 122, as posições de bits 31 a 15 definem o marcador virtual de cache I (I\$); as posições de bits 14 a 5 definem um índice 207 para seleção de uma linha ou conjunto dentre os 1024 "conjuntos de 1 linha" na cache I 122; e as posições de bits 4 a 0 definem um offset ou deslocamento em uma coluna de linha de instrução 220c. Uma fila na cache I mapeada diretamente 122 é também designada como um "conjunto de 1 linha". Apesar de, por exemplo, oito instruções de 32 bits poderem residir em uma única linha de instrução, para maior simplicidade, é ilustrada apenas uma instrução por marcador virtual I\$. Deve ficar claro que outros tamanhos de instruções são contemplados pela presente invenção. Quando o endereço virtual 205 é

utilizado pelo TLB 128 para tradução de endereço, as posições de bits 31 a 12 definem um marcador virtual TLB e as posições de bits 11 a 0 definem um offset de página.

5 Durante uma consulta à cache I por um endereço virtual especificado, o índice 207 proveniente do endereço virtual seleciona uma linha correspondente na cache I 122. O ASID ativo 215 e as posições de bits 31 a 15 do endereço virtual especificado são a seguir comparados com o tag ASID e o marcador virtual I\$ armazenado na linha de cache
10 selecionada.

De forma similar, durante uma translação de TLB para um endereço virtual especificado, o ASID virtual 215 e as posições de bits 31 a 12 do endereço virtual especificado são comparados com os tags ASID e marcadores
15 virtuais TLB armazenados no TLB 128 para encontrar uma igualdade ou correspondência e, portanto, um PPN correspondente.

Apesar de não ser ilustrado aqui, em certas modalidades a parte de um endereço virtual compreendendo um
20 marcador virtual TLB pode ser a mesma parte do endereço virtual compreendendo um marcador virtual de cache I. Os técnicos na área notarão que, apesar de a cache I marcada virtualmente 122 ser ilustrado como uma cache mapeada diretamente, a cache I 122 pode também ser incorporada na
25 forma de uma cache associativa em conjunto de modo a acessar grupos de entradas na forma de conjuntos de múltiplas linhas.

Voltando ao TLB 128, estão ali representadas uma relação "alias" e uma relação de diferenciação. Uma relação
30 alias ocorre quando o mesmo número de página física (PPN) está localizado em duas ou mais entradas no TLB 128. Como exemplo, as entradas 211 e 213 fazem ambas referência ao PPN 0x80000. Neste exemplo, as entradas 211 e 213 possuem o mesmo marcador virtual de TLB, porém estão associadas a um
35 processo de software diferente para tornar a combinação de

tab ASID/marcador virtual TLB exclusiva. As entradas 212 e 214 são também aliases entre si, pois ambas fazem referência ao PPN 0x90000. As entradas 212 e 214, uma em relação à outra, possuem marcadores virtuais TLB exclusivos e tags ASID exclusivos, de modo a definir uma combinação de marcador virtual ASID/TLB exclusiva. Uma relação de diferenciação ocorre quando o mesmo marcador virtual TLB é mapeado para dois ou mais números de página física diferentes. As entradas 211 e 212 apresentam uma relação de diferenciação pois o mesmo marcador virtual TLB (TLBtag_A) é mapeado para dois números de página física diferentes, 0x80000 e 0x90000, respectivamente.

Considerando-se que os marcadores virtuais TLB e marcadores virtuais I\$ são compostos por posições de bits em um endereço virtual (algumas das quais são posições de bits comuns), as entradas no TLB 128 estão relacionadas a entradas na cache I 122. Conseqüentemente, com a finalidade de nomenclatura, o uso de um sub-escrito na referência a um marcador virtual I\$ específico (por exemplo, I\$tag_A) indica que o valor do marcador virtual I\$ em questão está correlacionado ao valor de um marcador virtual TLB possuindo o mesmo sub-escrito (por exemplo, TLB tag_A).

Na modalidade exemplar da Figura 2, os marcadores virtuais TLB contêm mais bits do que o marcador virtual I\$. Conseqüentemente, um marcador virtual TLB pode corresponder a um ou mais marcadores virtuais I\$. Dessa forma, uma entrada TLB possuindo um número de página física pode corresponder a uma ou mais entradas de cache I, o que significa que a instrução ou instruções na cache I 122 podem também estar fisicamente localizadas no número de página física da entrada TLB correspondente. Para maior simplicidade, deve ser notado que nem todas as todas as entradas de cache I relacionadas a uma entrada TLB específica estão ilustradas. Dada a relação de uma para várias acima descrita entre as entradas TLB e as entradas

de cache I, caso uma dada fila de cache I esteja associada ao mesmo tag ASID e a um valor de tag de endereço virtual em uma dada fila do TLB, então a instrução ou instruções na fila de cache I em questão podem estar adicionalmente localizadas dentro da página do espaço de endereços físicos correspondente à página do espaço de endereços virtuais associado à fila TLB em questão.

A medida que passa o tempo, as entradas do TLB 128 e da cache I 122 podem se tornar obsoletas. Como exemplo, caso um ASID seja reutilizado por um programa de software subsequente não correlacionado e os mapeamentos virtuais para físicos do programa subsequente sejam diferentes daqueles do programa anterior, então as entradas no TLB e na cache I contendo o ASID reutilizado não mais seriam válidas pois elas seriam mapeadas para os endereços físicos associados ao programa de software anterior. Para manter a coerência de TLB e cache, é utilizado o circuito lógico de controle 110 para gerenciamento de entradas obsoletas tanto no TLB 128 como na cache I 122. De um modo geral, os programas de software desenvolvidos para uma cache I marcada fisicamente podem ser mais simples do que aqueles desenvolvidos para uma cache I marcada virtualmente, com relação ao gerenciamento de entradas obsoletas na cache I por várias razões, incluindo o fato de que o problema de cognomes não ocorre em uma cache I marcada fisicamente e que a mera mudança dos mapeamentos de endereço virtuais para físicos não requer a invalidação de cache I em uma cache I marcada fisicamente. No entanto, os programas de software escritos para caches I marcados fisicamente e aqueles escritos para caches I marcados virtualmente devem incluir instruções explícitas de invalidação de TLB quando da mudança do mapeamento de um dado endereço virtual para uma dada ID de processo de um endereço físico para outro.

Existem duas classes principais de situações em que as entradas de uma cache I devem ser invalidadas com uma cache de instruções marcada virtualmente. A primeira classe de situações envolve mudanças de mapeamentos de endereço em que, por exemplo, um endereço virtual passa a ser mapeado para um endereço físico diferente. Em implementações convencionais de cache I marcada virtualmente, são utilizadas instruções de invalidação de cache I para invalidar a cache I em tal classe de situações. Além disso, na primeira classe de situações, são utilizadas instruções de invalidação de TLB para invalidar um TLB tanto em implementações convencionais de cache I marcada virtualmente como de cache I marcada fisicamente. A segunda classe de situações envolve o caso em que o conteúdo da memória física é modificado. Na segunda classe de situações, são utilizadas instruções de invalidação de cache I para invalidar a cache I tanto em implementações convencionais de cache I marcada virtualmente como em implementações convencionais de cache I marcada fisicamente. No entanto, em implementações convencionais de cache I marcada fisicamente, uma instrução de invalidação de cache I deve apenas identificar o endereço físico da memória modificada a ser invalidada, ao contrário de todos os endereços virtuais possíveis que poderiam dizer respeito a tal endereço físico em uma implementação convencional de cache I marcada virtualmente.

O circuito lógico de controle 110 está configurado para reconhecer que a primeira classe de situações que iria convencionalmente requerer programas de software para invocar invalidação de cache I explícita para uma cache I marcada virtualmente também requer programas de software para invocar invalidação de TLB mesmo para uma cache I marcada fisicamente. Como exemplo, uma situação inclui o caso em que uma combinação de marcador virtual TLB/ASID é mapeada para um novo número de página física. Em

tal caso, o circuito lógico de controle 110, em resposta a uma instrução de invalidação TLB explícita, invalida as entradas no TLB 128 envolvendo a combinação de marcador virtual TLB/ASID e também invalida entradas na cache I
5 marcada virtualmente 122 envolvendo a combinação de marcador virtual de cache I/ASID. Em certas modalidades, a instrução de invalidação TLB explícita causa a invalidação flash de todas as entradas na cache I 122 independentemente da combinação de marcador virtual TLB/ASID.

10 Com referência à segunda classe de situações, referente a modificações efetuadas no conteúdo de um endereço físico especificado, um programa de software emite uma instrução de invalidação de cache I explícita para levar a cache I 122 a ser atualizado. O conteúdo da
15 instrução de invalidação de cache I explícita pode ou não especificar um endereço virtual associado ao endereço físico modificado ou o endereço físico modificado.

Em uma modalidade, o circuito lógico de controle 110, em resposta a tal instrução de invalidação de cache I
20 explícita, é configurado para invalidar em flash todas as entradas na cache I marcada virtualmente 122. Em tal modalidade, um endereço virtual associado ao endereço físico em questão, ou, alternativamente, o endereço físico em questão, pode ser especificado na instrução de
25 invalidação de cache I explícita. No entanto, dado que todas as entradas de cache I são invalidadas em flash, o endereço virtual ou físico especificado da instrução de invalidação de cache I explícita é simplesmente ignorado.

Em uma modalidade alternativa, o circuito lógico
30 de controle 110, em resposta à instrução de invalidação de cache I explícita, é configurado para invalidar entradas seletivamente na cache I marcada virtualmente 122. dado que as entradas de cache I, de um modo geral, possuem marcadores virtuais que correspondem a um endereço físico,
35 as entradas de cache I que possam corresponder ao endereço

físico especificado deveriam ser invalidadas. É feita referência às entradas nas filas 212 e 214 no TLB 128 para um exemplo da identificação de múltiplos endereços virtuais para os quais as correspondentes entradas de cache I
5 marcadas virtualmente deveriam ser invalidadas, como resultado de uma mudança do conteúdo de um endereço físico. Tais entradas são entradas "alias" e são mapeadas para o mesmo número de página física 0x90000.

De forma similar, as entradas de cache I 225 e
10 224 contêm instruções que são também encontradas em um endereço físico definido pelos primeiros 17 bits do número de página física, pois o valor dos marcadores virtuais I\$ (I\$ tagA e I\$ tagB) estão relacionados ao valor dos marcadores virtuais TLB (TLB tagA e TLB tagB),
15 respectivamente, tal como ilustrado pelo significado duplo superposto das posições de bits do endereço virtual 205 acima descrito. Conseqüentemente, caso, por exemplo, o conteúdo do endereço físico 0x9000_0000 mude, as entradas 225 e 224, juntamente com quaisquer outras entradas não
20 ilustradas cujas instruções são também encontradas no endereço físico 0x9000_0000, serão seletivamente invalidadas pelo circuito lógico de controle 110. Mais particularmente, nesta modalidade de invalidação seletiva de entradas na cache I marcada virtualmente 122, o circuito
25 lógico de controle 110 está configurado para invalidar todas as entradas associadas a índices virtuais de cache I específicos em que o endereço físico especificado possa residir, tal como todas as entradas nos conjuntos de cache I selecionados.

30 A Figura 3 é um diagrama de blocos ilustrando a invalidação seletiva de um conjunto de entradas de cache I marcadas virtualmente que poderiam ser associadas a um endereço físico em uma cache I mapeada diretamente 122. Como mostrado no endereço virtual 205, as posições de bits
35 14 a 12, ou a referência 305, fazem parte dos bits de

índice, também designados como os bits de índice de cache I, quando o endereço virtual 205 é utilizado pela cache I marcada virtualmente 122. As posições de bits 14 a 12 fazem também parte do marcador virtual TLB. No entanto, dado que
5 as posições de bits 14 a 12 são bits traduzidos durante a tradução de endereço, estes mesmos bits fazem parte do número de página física correspondente ao endereço virtual 205. Dado que as posições de bits 14 a 12 não são utilizadas pelo marcador virtual I\$, e que um número de
10 página física pode ser especificado para invalidação, cada combinação de bits 14 a 12 define limites 0, 128, 256, ..., 896, dentro da cache I marcada virtualmente 122, onde poderia estar localizada uma entrada de cache I associada correspondente ao número de página física. Os bits de
15 índice remanescentes, os bits 11 a 5, ou a referência 315, não são traduzidos durante a tradução de endereço e, portanto, são comuns ao endereço virtual 205 e seu endereço físico transladado. Conseqüentemente, os bits 11 a 5 especificam um offset em relação aos limites definidos para
20 localização de entradas de cache I correspondentes potencialmente relacionadas à página física. Dado que a referência 305 contém três bits, o circuito lógico 110, em resposta à invalidação de um endereço físico, pode ser configurado para invalidar simultaneamente oito entradas na
25 cache I marcada virtualmente 122 por concatenação de cada combinação de bits 14 a 12 com os bits 11 a 5 do endereço físico.

De um modo geral, dado que os bits 11 a 0 não são traduzidos pela tradução de endereços, eles não variam
30 entre um endereço físico especificado e quaisquer endereços virtuais que poderiam ser traduzidos para tal endereço físico. Posto de outra forma, todos os endereços virtuais que possam ser mapeados para o endereço físico especificado devem estar de acordo com o endereço físico nos bits 11 a
35 0. Portanto, quaisquer dos bits dentre aqueles nas posições

11 a 0 que sejam usados como parte do índice de cache I para seleção de um conjunto de cache I que possa conter entradas correspondentes ao endereço físico especificado, devem possuir o mesmo valor que os bits correspondentes
5 provenientes do endereço físico. São apenas aqueles bits de índice de cache I nas posições 14 a 12, por exemplo, que poderiam assumir um valor diferente dos bits correspondentes provenientes do endereço físico, mas que ainda identificam um conjunto de entradas possivelmente
10 contendo entradas correspondentes ao endereço físico especificado.

Em resposta a uma instrução de invalidação de cache I explícita identificando um endereço físico a ser invalidado na cache I devido ao conteúdo de tal endereço
15 físico ter sido modificado, o circuito lógico de controle utiliza as posições de bits 11 a 5 para determinar os offsets a partir dos limites definidos pelos bits 14 a 12 para invalidar todas as entradas no conjunto transladado de entradas qualificadas na cache I, permitindo que a cache I
20 seja invalidado com base em um conjunto em lugar de com base em todo uma cache I. Com uma única instrução especificando um endereço físico a ser invalidado ou, alternativamente, um endereço virtual associado ao endereço físico inválido, tal modalidade permite a um software
25 escrito para um modelo de cache I marcada fisicamente invocar automaticamente a operação de invalidação de cache I em uma cache I marcada virtualmente para assegurar que todos os cognomes de endereços virtuais possíveis para o endereço físico especificado sejam removidos da cache. Como
30 descrito, tal remoção pode ocorrer com base em uma linha, conjunto, ou em toda uma cache. Dessa forma, a cache I marcada virtualmente é gerenciada apropriadamente sem qualquer carga adicional sobre o software que foi desenvolvido para o modelo de cache I marcada fisicamente.

O número de índices simultâneos para a cache I marcada virtualmente 122 pelo circuito lógico de controle 110 depende da associação de conjunto de uma cache I marcada virtualmente. A Figura 4 é um diagrama de blocos 5 ilustrando a invalidação seletiva de um conjunto de entradas de cache I virtualmente marcadas que poderiam estar associadas a um endereço físico em uma cache I associativa de conjunto 2-via 422. Abstraindo-se sua associação, a cache I associativa 2-via 422 é uma cache I 10 marcada virtualmente de 32 KB com linhas de 32 bytes e que pode ser adequadamente similar à cache I 222. O endereço virtual 405 é um endereço de 32 bits armazenado em um registrador tal como um contador de programa (PC) e, de forma semelhante ao endereço virtual 205, é representado 15 apresentando dois significados superpostos para suas posições de bits. A cache I 422 possui um índice 418 possuindo um tamanho de nove bits e, portanto, um marcador virtual I\$ possuindo um tamanho de 18 bits. Para tal modalidade, as posições de bits 13 a 12, ou as posições de 20 bits traduzidas 410, definirão quatro limites, resultando em quatro índices simultâneos, enquanto as posições de bits 11 a 5, ou as posições de bits não traduzidas 415 definirão um offset a partir de cada um dentre quatro limites para invalidar seletivamente filas na cache I marcada 25 virtualmente. Em uma cache associativa de conjuntos 2-via, cada fila inclui duas linhas de cache. Conseqüentemente, o circuito lógico de controle 110 indexa simultaneamente quatro filas de cache e, portanto invalida oito entradas de cache I marcada virtualmente.

30 Deve ficar claro que vários tamanhos e configurações de caches I marcados virtualmente são contemplados como modalidades adicionais. Como exemplo, uma modalidade associativa de conjunto 4-via de uma cache I marcada virtualmente de 32 KB com linhas de 32 bytes iria 35 resultar em um índice possuindo um tamanho de oito bits e,

portanto, um marcador virtual I\$ possuindo um tamanho de 19 bits. Para uma cache I associativa 4-via, a posição de bit 12 irá definir dois limites e as posições de bits 11 a 5 definirão o offset a partir dos dois limites para invalidar seletivamente as entradas na cache I 4-via.

Como outro exemplo, uma modalidade associativa de conjunto 8-via de uma cache I marcada virtualmente de 32 kb com linhas de 32 bytes irá resultar em um índice possuindo um tamanho de sete bits e, portanto, um marcador virtual I\$ possuindo um tamanho de 20 bits, o mesmo tamanho de um marcador virtual TLB e, logo, de um número de página física. Para uma cache I associativa de conjunto 8-via, não existem limites definidos e as posições de bits 11 a 5 irão indexar para a cache 8-via. No entanto, note-se que cada fila em uma cache 8-via compreende 8 linhas de cache, sendo seletivamente invalidadas oito linhas de cache. Além disso, dado que neste exemplo o tamanho de marcador virtual TLB é igual ao tamanho de marcador virtual I\$, é garantido que quaisquer entradas na cache I que possam ser associadas ao endereço físico especificado estejam localizadas na via de cache I selecionada pelo único índice de cache I, o qual por sua vez é o mesmo que as posições de bits de índice provenientes do endereço físico especificado, uma vez que tais posições de bits não são traduzidas. Dito de outra forma, quando todos os bits de índices de cache I estão em posições de bits dentro de um offset de página, portanto, os bits de índice de cache I não são traduzidos pelo mecanismo de tradução de endereço, é garantido que quaisquer entradas na cache I que possam estar associadas ao endereço físico especificado estejam localizadas na via de cache I selecionada por tal índice único de cache I.

Os exemplos acima demonstram como uma cache I associativa virtualmente indexada e virtualmente marcada de acordo com os ensinamentos da presente invenção afeta a invalidação seletiva. Em particular, quanto mais alta a

associatividade de conjunto da cache virtualmente indexada e virtualmente marcada, mais baixo o número de bits que são traduzidos pelo processo de tradução de endereço, mas que são utilizados em sua forma não traduzida no índice da cache I, resultando em um número mais baixo de filas de cache I sendo seletivamente invalidadas. De forma similar, o tamanho de uma página física afeta o número de bits de endereço que são traduzidos. Portanto, quanto maior o tamanho da página, menor o número de bits que são traduzidos pelo processo de tradução de endereço, mas que são utilizados em sua forma não traduzida para o índice da cache I, resultando novamente em um menor número de filas/conjuntos de cache I sendo seletivamente invalidados.

No caso em que o número de bits no número de página física for o mesmo que o número de bits no marcador virtual de cache I, não existem quaisquer bits no índice de cache I que finalmente sofrerão translação pelo processo de tradução de endereço. Dessa forma, o conjunto de vias de cache I potenciais é reduzido para uma via garantida. De um modo geral, o aumento no tamanho de página resulta em menos bits de translação e, portanto, é reduzido o tamanho de entradas de cache I potenciais associadas ao endereço físico.

Fazendo novamente referência à Figura 2, o circuito lógico de controle 110 recebe, como entrada, instruções de invalidação, tais como instruções de invalidação de TLB e instruções de invalidação de cache I, provenientes de um programa de software, independentemente de se o programa de software foi desenvolvido com uma cache I marcada fisicamente ou uma cache I marcada virtualmente. Tais instruções de invalidação podem ser providas a partir de um estágio de decodificação na linha 120, tal como o estágio de linha 160b por exemplo. O circuito lógico de controle 110 responde a um conjunto de instruções para invalidar TLB para invalidar uma ou mais entradas na cache

I 122, utilizando mecanismos conhecidos, tais como a geração de sinais de controle de cache I para invalidar uma ou mais entradas de uma cache I em resposta a instruções para invalidar cache I conhecidas. Pela resposta ao conjunto de instruções para invalidar TLB para gerenciar 5 adicionalmente a cache I e responder à instrução de invalidar cache I para o caso que indica que o conteúdo de um endereço físico mudou, o circuito lógico de controle 110 alivia a carga de um programa de software de ter de emitir 10 instruções para invalidar cache I separadamente para cada situação que uma cache I marcada virtualmente convencional requiera. Conseqüentemente, os programas de software que são escritos para gerenciar a cache I fisicamente marcado podem agora ser vantajosamente executados no processador 15 100 gerenciando a cache I virtualmente indexado e virtualmente marcado de tal forma que ele se comporte como se ele fosse fisicamente indexado e fisicamente marcado, do ponto de vista do software.

O circuito lógico de controle 110 também responde 20 a instruções de invalidação de cache I conhecidas para invalidar uma ou mais entradas na cache I marcada virtualmente 122 para que os programas de software existentes que são desenvolvidos para gerenciar uma cache I marcada virtualmente. Adicionalmente, em certas 25 modalidades, o circuito lógico de controle 110 pode receber um sinal de habilitação opcional 237 proveniente do processador 100, que pode ser utilizado caso estejam sendo executados programas de software especificamente desenvolvidos para uma cache I marcada virtualmente. O 30 sinal de habilitação 237 pode ser invocado como resultado de um bit de estado de processador, um bit de configuração e similares. Quando o sinal de habilitação 237 é invocado, o circuito lógico de controle 110 opera tal como acima descrito. Quando o sinal de habilitação 237 não for 35 invocado (desabilitado), o circuito lógico de controle 110

irá responder a instruções de invalidação de TLB para invalidar uma ou mais entradas provenientes do TLB 128 e irá responder a instruções para invalidar cache I para invalidar uma ou mais entradas na cache I marcada virtualmente 122. Quando o sinal de habilitação 237 não é invocado (desabilitado), o circuito lógico de controle 110 não irá invalidar entradas na cache I marcada virtualmente 122 em resposta às instruções para invalidar TLB e não irá necessariamente invalidar mais entradas de cache I do que aquelas especificamente associadas a um endereço virtual especificado, em resposta às instruções para invalidar cache I.

Em certas modalidades, o circuito lógico de controle 110 pode incluir um gerador de sinais de controle TLB 235 e um gerador de sinais de controle de cache I 245. As alimentações 241 e 243 para o gerador de sinais de controle de cache I 245 acoplam respectivamente instruções para invalidar TLB e instruções para invalidar cache I para o gerador de sinais de controle de cache I 245. O gerador de sinais de controle de cache I 245 está configurado para gerar sinais de controle para invalidar uma ou mais entradas na cache I marcada virtualmente 122 com base na instrução de invalidar recebida. O gerador de sinais de controle TLB 235 gera sinais de controle de uma maneira conhecida para invalidar uma ou mais entradas no TLB 128 com base no tipo específico de instrução de invalidar TLB recebida. Quando a entrada ao gerador de sinais de controle de cache I 245 for resultado de uma instrução de invalidar cache I, o gerador de sinais de controle de cache I 245 gera sinais de controle de uma maneira conhecida para invalidar uma ou mais entradas na cache I marcada virtualmente 122. Adicionalmente, o gerador de sinais de controle de cache I gera sinais de controle para invalidar uma ou mais entradas na cache I marcada virtualmente 122 em resposta a instruções para invalidar TLB. A tabela que se

segue descreve o comportamento funcional do gerador de sinais de controle de cache I 245 resultante da recepção das seguintes instruções para invalidar TLB:

Instruções para invalidar TLB	Comportamento do gerador de sinais de controle de cache I
Invalidar todas as entradas TLB (TLBIALL)	Gera sinais de controle para invalidar em flash todo a cache I 122. Como exemplo, gera os mesmos sinais de controle que uma instrução ICIALL conhecida.
Invalidar entradas TLB associadas a um ASID (TLBIASID)	Gera sinais de controle para invalidar todas as entradas na cache I 122 associadas ao ASID portado na invalidação de TLB por instrução ASID. Alternativamente, gera sinais para invalidar em flash todo a cache I 122.
Invalidar entradas TLB por endereços virtuais (TLBIVA)	Gera sinais de controle para invalidar em flash todo a cache I 122.

Tabela 1

5 A coluna 220d constitui uma modalidade para assegurar que não ocorra "concordância" errônea em uma entrada na cache I 122 com base em uma comparação de tag virtual, quando a entrada foi estabelecida em um momento em que o mecanismo de translação estava em um estado de
10 habilitação, porém a concordância está ocorrendo em um momento posterior quando o mecanismo de translação está em outro estado de habilitação. Como exemplo, no caso em que um mecanismo de translação, tal como o TLB 128, possui um recurso de habilitação que permita ao TLB 128 estar em um
15 estado habilitado ou desabilitado, as entradas armazenadas na cache I 122 podem conter um valor de indicador indicando

se o TLB estava ou não operacional no momento em que a entrada foi armazenada. Quando uma entrada é armazenada na cache I durante um TLB 128 não operacional, a entrada armazenada é associada a um endereço físico que é o mesmo
5 que o endereço virtual não transladado.

No caso em que o TLB passa de desabilitado para habilitado, as entradas da cache I que forem armazenadas durante o TLB não operacional são invalidadas pelo circuito lógico de controle 110 em resposta ao sinal de transição
10 TLB 239. Além disso, no caso em que o TLB passa de habilitado para desabilitado, as entradas de cache I que forem armazenadas durante o TLB operacional são invalidadas pelo circuito lógico de controle 110 em resposta ao sinal de transição TLB 239. O sinal de transição TLB 239 é
15 comumente gerado quando o TLB passa de desabilitado para habilitado e vice versa. Em particular, o gerador de sinais de controle de cache I 245, em resposta ao sinal de transição TLB 239, gera sinais para invalidar as entradas na cache I 122 que estão de acordo com o campo do indicador
20 indicando as entradas que foram inicialmente armazenadas enquanto o TLB 128 estava no estado do qual ela acabou de sair. Em uma modalidade alternativa, os indicadores da coluna 220d podem ser ajustados quando o estado da tabela de translação é habilitado. Em tal modalidade, quando é
25 efetuada uma consulta à cache I, os indicadores são considerados para determinar se ocorre uma concordância com o estado atual da tabela de translação.

Em outra modalidade alternativa, o problema de assegurar que não ocorra uma "concordância" errônea em uma
30 entrada da cache I estabelecida em um estado inconsistente anterior é solucionado sem a coluna 229d opcional. Em tal modalidade, o circuito de controle lógico 110 invalida em flash todo a cache 122 quando o sinal de transição TLB habilitado 239 indica que ocorreu uma transição no
35 mecanismo de habilitação de TLB.

Será notado pelos técnicos na área que o tamanho do endereço virtual 205 e as dimensões do TLB 128 e a cache I 122 podem ser aplicados a outras modalidades sem constituir um afastamento dos ensinamentos da invenção.

5 A Figura 5 é um fluxograma ilustrando um método 500 para que uma cache de instruções marcada virtualmente opere como se ele fosse uma cache de instruções marcada fisicamente do ponto de vista de um programa de software. No bloco de função 510, o método recupera uma instrução.

10 A instrução recuperada passa à decodificação de instruções 530 onde a instrução recuperada é decodificada. No bloco 540, o método determina o tipo de instrução. Caso o tipo de instrução seja uma instrução de invalidar tradução de endereço, o método passa ao bloco 50 que
15 invalida uma ou mais entradas em um TLB, tal como o TLB 128, de acordo com a instrução de invalidar tradução de endereço. O método a seguir prossegue para o bloco 560. No bloco 560, o método também invalida uma ou mais entradas na
20 cache I 122 de acordo com um mapeamento de instruções para invalidar tradução de endereço tal como definido na Tabela 1. Os técnicos na área notarão que a ordem dos blocos 550 e 560 pode ser invertida sem afetar o processo de invalidação. Voltando ao bloco 540, caso o tipo de instrução seja uma instrução de invalidar cache I, o método
25 passa ao bloco 560 para invalidar uma ou mais entradas na cache I 122 de acordo com a instrução de invalidar cache I. Voltando ao bloco 540, caso o tipo de instrução não seja uma instrução de invalidação de qualquer tipo, então volta-se ao bloco 510 para recuperar a próxima instrução.

30 Os vários exemplos de blocos lógicos, módulos, circuitos, elementos e/ou componentes aqui descritos em conexão com as modalidades aqui apresentadas podem ser implementados ou efetivados por meio de um processador de uso geral, um processador de sinais digitais (DSP), um
35 circuito integrado específico para aplicação (ASIC),

arranjos de porta programáveis no campo (FPGA) ou outros dispositivos lógicos programáveis, portas individuais ou lógica de transistores, componentes de hardware individuais, ou quaisquer combinações de tais projetadas para efetuar as funções aqui descritas. Um processador de uso geral pode ser um microprocessador, porém como alternativa o processador pode ser qualquer processador, controlador, micro controlador, ou máquina de estado convencionais. Um processador pode também ser implementado na forma de uma combinação de dispositivos de computação, por exemplo, uma combinação de um DSP e um microprocessador, uma pluralidade de microprocessadores, um ou mais microprocessadores em conjunto com um núcleo DSP, ou qualquer outra configuração similar.

Os métodos ou algoritmos descritos em conexão com as modalidades aqui apresentadas podem ser efetivadas diretamente em hardware, em um módulo de software executado por um processador, ou em uma combinação de ambos. Um módulo de software pode residir em uma memória RAM, memória flash, memória ROM, memória EPROM, memória EEPROM, registradores, um disco rígido, um disco removível, um CD-ROM, ou qualquer outra forma de meio de armazenamento conhecido pelos técnicos na área. Um meio de armazenamento pode ser acoplado ao processador de tal forma que o processador possa ler informações provenientes do, e gravar informações no, meio de armazenamento. Como alternativa, o meio de armazenamento pode estar integrado ao processador.

Apesar de a invenção ser descrita no contexto de modalidades, ficará claro que pode ser empregada uma ampla variedade de implementações pelos técnicos na área consistentes com a descrição acima e com as reivindicações que se seguem.

REIVINDICAÇÕES

1. Um sistema de cache de instruções, compreendendo:

uma cache de instruções marcada virtualmente;

5 um dispositivo para tradução de endereço, o dispositivo respondendo a uma instrução de invalidar tradução de endereço; e

um circuito lógico de controle configurado para invalidar uma entrada na cache de instruções marcada virtualmente em resposta à instrução de invalidar tradução de endereço.

2. O sistema de cache de instruções, de acordo com a reivindicação 1, no qual o dispositivo para tradução de endereço consiste de um buffer de previsão de tradução (TLB).

3. O sistema de cache de instruções, de acordo com a reivindicação 1, no qual o circuito lógico de controle está configurado para invalidar em flash a cache de instruções marcada virtualmente em resposta à instrução de invalidar tradução de endereço.

4. O sistema de cache de instruções, de acordo com a reivindicação 1, no qual o circuito lógico de controle está configurado para invalidar uma ou mais entradas na cache de instruções marcada virtualmente correspondentes a um identificador de serviço de aplicativo (ASID) especificado pela instrução de invalidar tradução de endereço.

5. O sistema de cache de instruções, de acordo com a reivindicação 1, no qual o circuito lógico de controle está configurado para invalidar um conjunto seletivo de entradas na cache de instruções marcada virtualmente correspondente a um índice especificado na instrução de invalidar tradução de endereço.

6. O sistema de cache de instruções, de acordo com a reivindicação 1, no qual o circuito lógico de

controle está configurado para invalidar em flash a cache de instruções marcada virtualmente em resposta aos dispositivos para estado de mudança de tradução de endereço.

5 7. O sistema de cache de instruções, de acordo com a reivindicação 1, no qual o circuito lógico de controle compreende uma entrada/entrada para receber um sinal de habilitação, o circuito lógico de controle estando configurado para invalidar a entrada na cache de instruções
10 marcada virtualmente somente em resposta à instrução de invalidar tradução de endereço após receber o sinal de habilitação.

 8. O sistema de cache de instruções, de acordo com a reivindicação 1, no qual o circuito lógico de
15 controle está configurado para invalidar em flash a cache de instruções marcada virtualmente em resposta a uma instrução de invalidar tradução de endereço especificando a invalidação de um endereço físico específico.

 9. O sistema de cache de instruções, de acordo com a reivindicação 5, no qual o índice compreende posições
20 de bits traduzidas e posições de bits não traduzidas.

 10. O sistema de cache de instruções, de acordo com a reivindicação 1, o qual está disposto em um
processador.

25 11. Um método para que uma cache de instruções marcada virtualmente opere tal como se ele fosse uma cache de instruções marcada fisicamente do ponto de vista de um programa de software, o método compreendendo:

 receber uma instrução de invalidar uma tradução
30 de endereço; e

 invalidar uma entrada em uma cache de instruções marcada virtualmente em resposta à instrução de invalidar uma tradução de endereço.

12. O método, de acordo com a reivindicação 11, no qual o invalidar a entrada em uma cache de instruções marcada virtualmente compreende também:

5 invalidar em flash a cache de instruções marcada virtualmente.

13. O método, de acordo com a reivindicação 11, no qual o invalidar a entrada na cache de instruções marcada virtualmente compreende também:

10 invalidar uma ou mais entradas correspondentes a um identificador de serviço de aplicativo (ASID) especificado pela instrução de invalidar tradução de endereço.

14. O método, de acordo com a reivindicação 11, no qual o invalidar a entrada na cache de instruções marcada virtualmente compreende também:

15 invalidar um conjunto seletivo de entradas correspondente a um endereço especificado pela instrução de invalidar tradução de endereço.

15. O método, de acordo com a reivindicação 11, no qual o invalidar a entrada na cache de instruções marcada virtualmente compreende também:

20 invalidar em flash a cache de instruções marcada virtualmente em resposta a um dispositivo para estado de mudança de tradução de endereço.

25 16. O método, de acordo com a reivindicação 11, no qual o invalidar a entrada na cache de instruções marcada virtualmente compreende também:

30 receber um sinal de habilitação; e
invalidar a entrada na cache de instruções marcada virtualmente somente após a recepção do sinal de habilitação.

17. O método, de acordo com a reivindicação 11, no qual o invalidar a entrada na cache de instruções marcada virtualmente compreende também:

receber uma instrução de invalidar cache de instruções especificando a invalidação de um endereço físico específico; e

5 invalidar em flash a cache de instruções marcada virtualmente em resposta à instrução de invalidar cache de instruções.

18. Um circuito lógico de controle para ordenar a uma cache de instruções marcada virtualmente que opere como se ele fosse uma cache de instruções marcada fisicamente do ponto de vista de um programa de software, compreendendo:

uma primeira entrada/entrada para receber uma instrução de invalidar uma tradução de endereço; e

15 um dispositivo para gerar sinais de controle para invalidar uma entrada na cache de instruções marcada virtualmente em resposta à instrução de invalidar uma tradução de endereço.

19. O circuito lógico de controle, de acordo com a reivindicação 18, compreendendo também:

20 uma segunda entrada/entrada para receber uma instrução de invalidar cache de instruções em que o dispositivo para gerar sinais de controle para invalidar uma entrada na cache de instruções marcada virtualmente responde também à instrução de invalidar a cache de instruções.

20. O circuito lógico de controle, de acordo com a reivindicação 19, compreendendo também:

30 uma terceira entrada para desabilitar o dispositivo para geração de sinais de controle para invalidar uma entrada na cache de instruções marcada virtualmente em resposta à instrução de invalidar uma tradução de endereço.

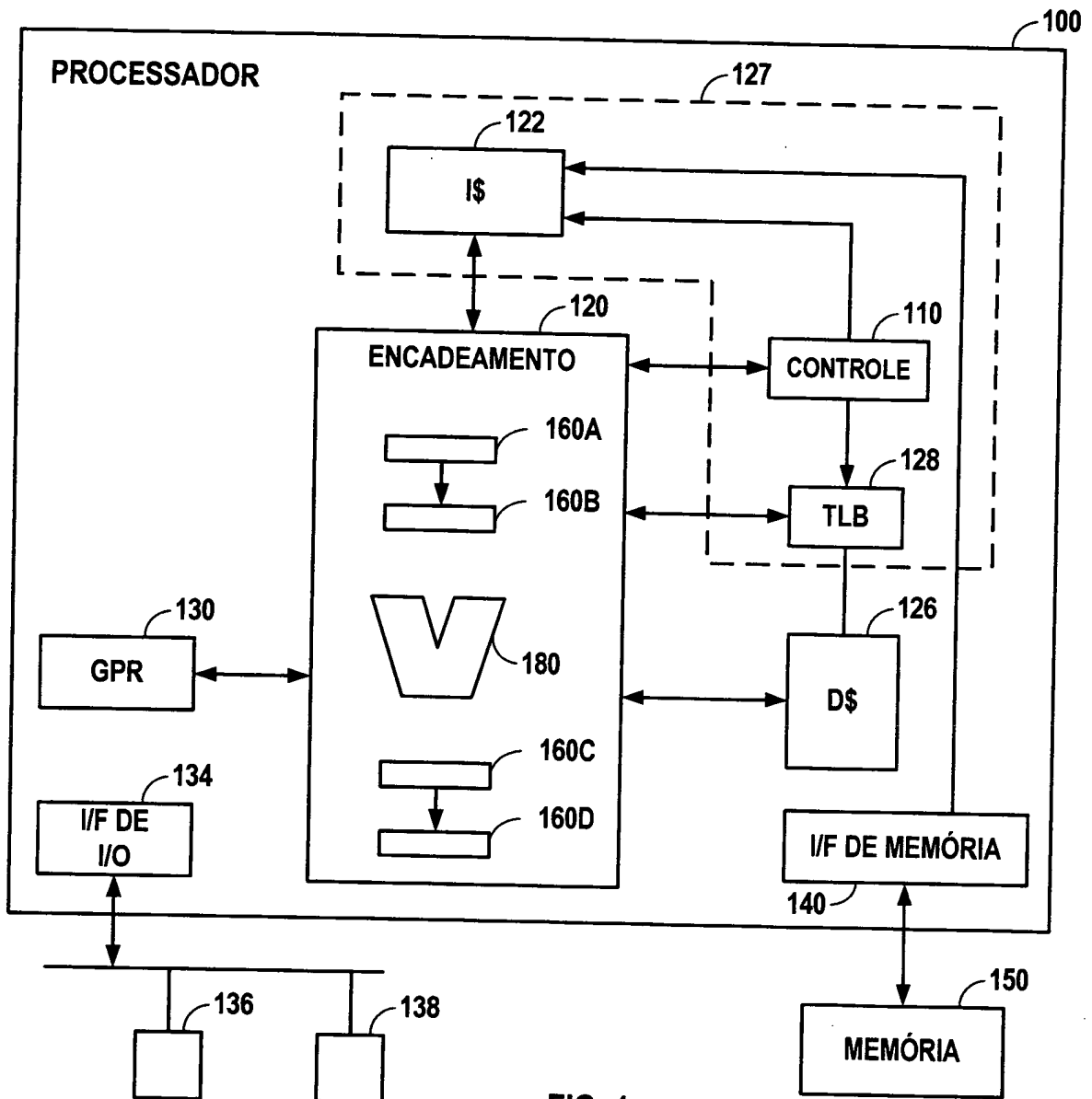


FIG. 1

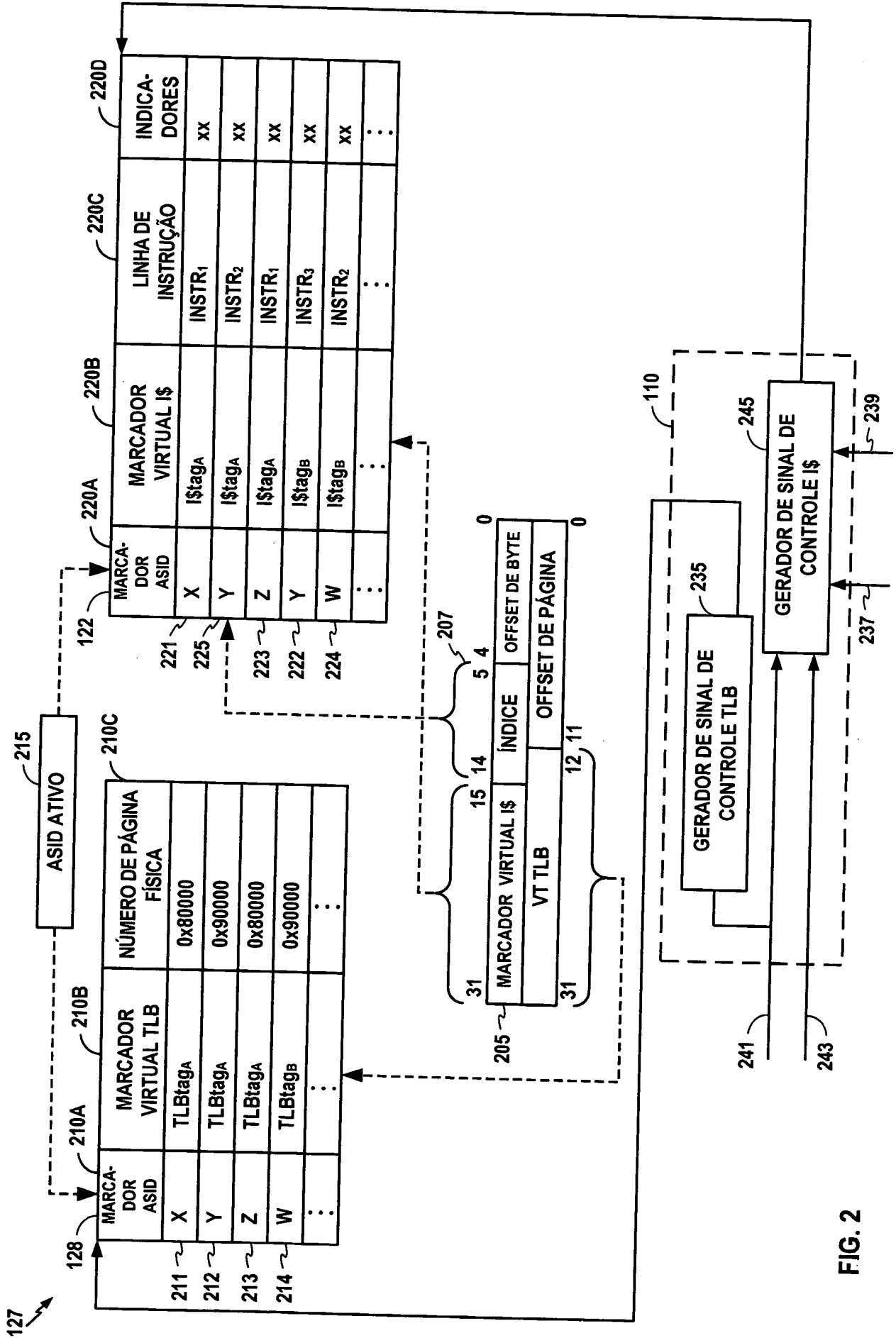


FIG. 2

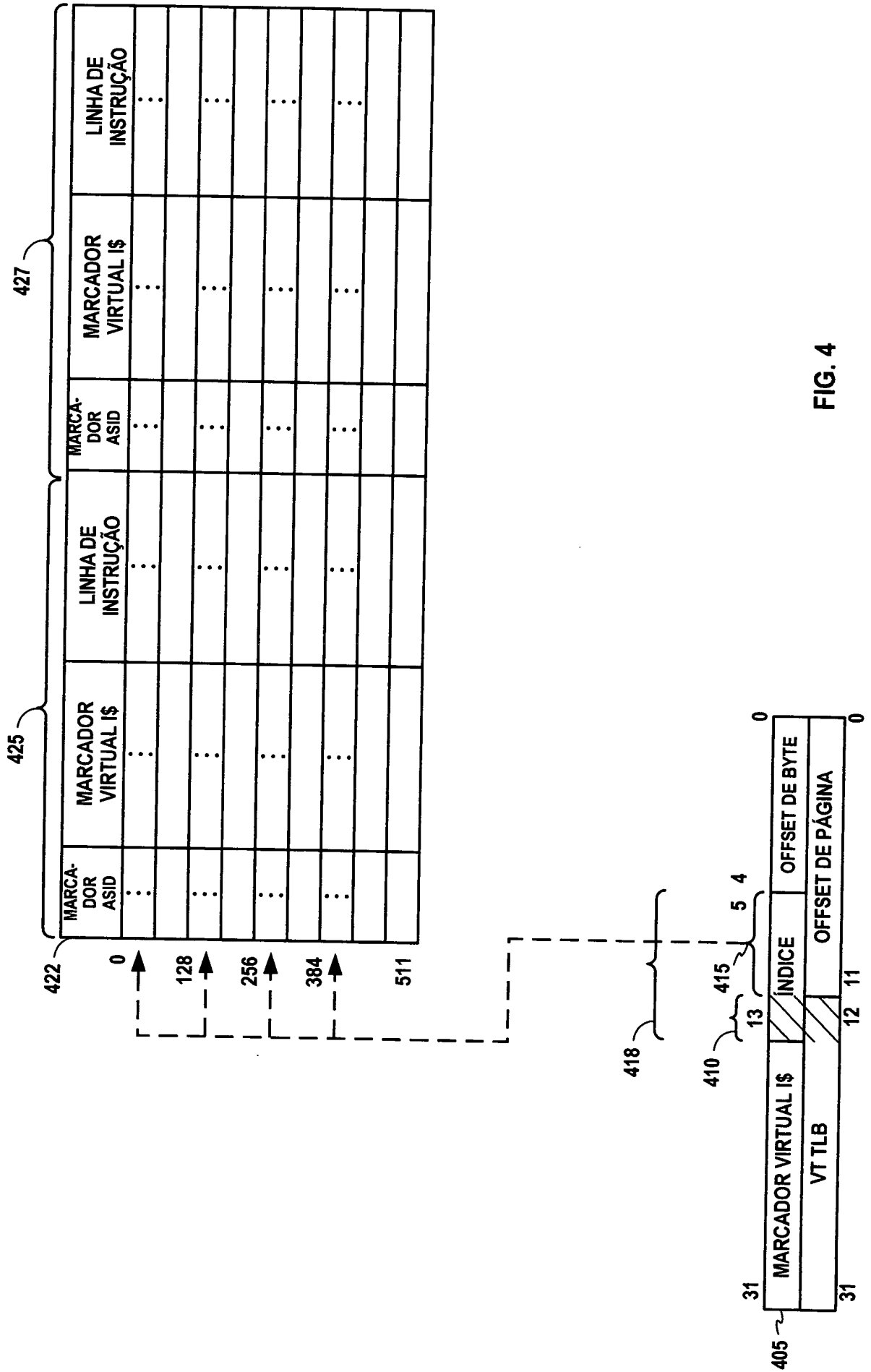


FIG. 4

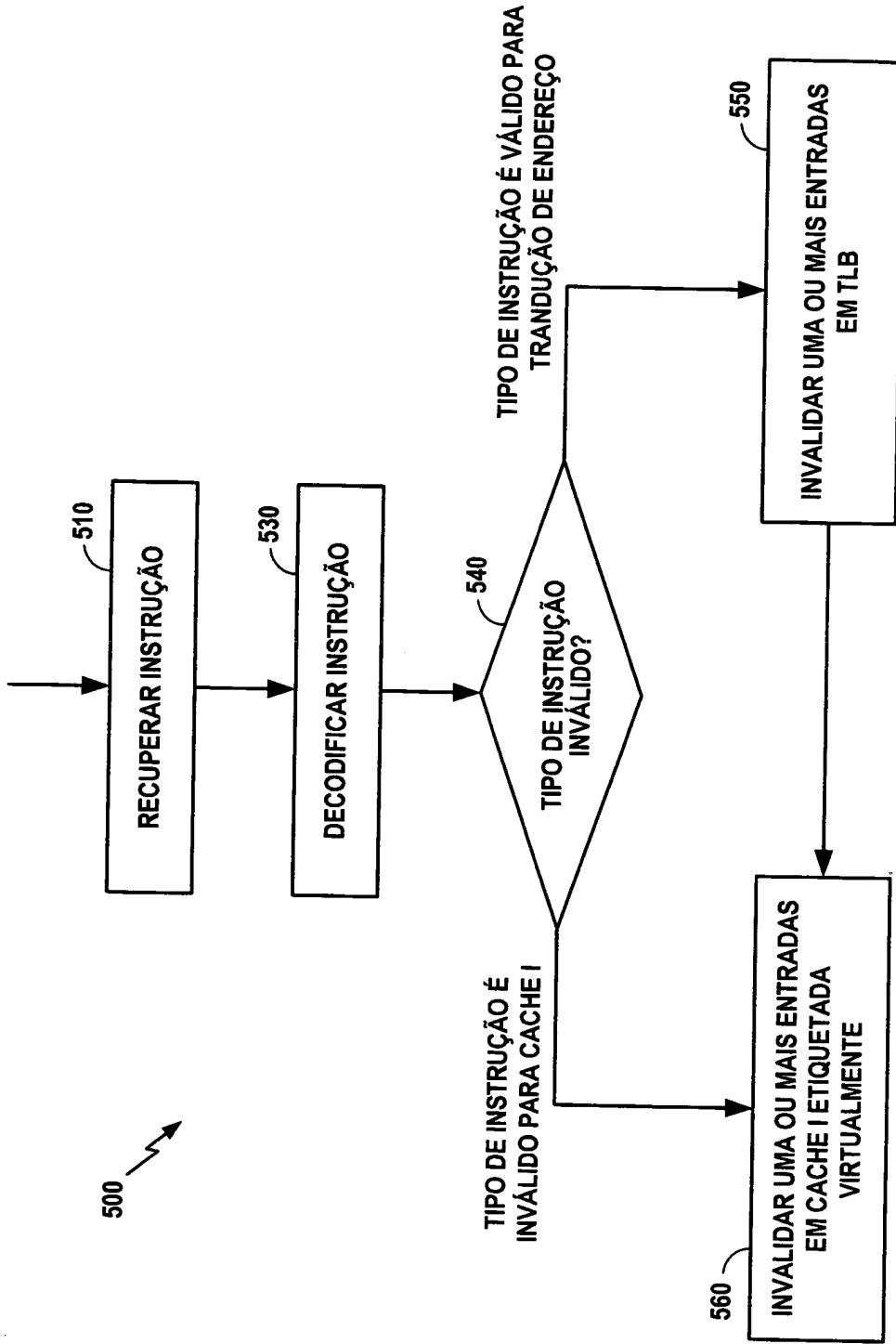


FIG. 5

PI0710171-6

RESUMO

**"CACHE DE INSTRUÇÕES MARCADA VIRTUALMENTE COM COMPORTAMENTO
MARCADO FISICAMENTE"**

É descrito um sistema de cache de instruções
5 possuindo uma cache de instruções marcada virtualmente que,
do ponto de vista de um programa de software, opera como se
fosse uma cache de instruções marcada fisicamente. O
sistema de cache de instruções inclui também um dispositivo
para endereçar translação que responde a uma instrução de
10 invalidar endereçar translação e um circuito lógico de
controle. O circuito lógico de controle está configurado
para invalidar uma entrada na cache de instruções marcada
virtualmente em resposta à instrução de invalidar endereçar
translação.