

(12) FASCÍCULO DE PATENTE DE INVENÇÃO

(22) Data de pedido: 2004.05.06	(73) Titular(es): INTERNATIONAL BUSINESS MACHINES CORPORATION NEW ORCHARD ROAD ARMONK, NY 10504 US
(30) Prioridade(s): 2003.05.12 US 435919	
(43) Data de publicação do pedido: 2006.05.03	
(45) Data e BPI da concessão: 2010.12.08 243/2010	(72) Inventor(es): TIMOTHY SLEGEL GB LISA HELLER GB ERWIN PFEFFER GB KENNETH PLAMBECK GB
	(74) Mandatário: ELSA MARIA MARTINS BARREIROS AMARAL CANHÃO PT RUA DO PATROCÍNIO 94 1399-019 LISBOA

(54) Epígrafe: **INVALIDAÇÃO DE ARMAZENAMENTO, ELIMINAÇÃO DE ENTRADAS DE UMA MEMÓRIA TAMPÃO**

(57) Resumo:

UNIDADES DE ARMAZENAMENTO SELECCIONADAS, TAIS COMO SEGMENTOS DE ARMAZENAMENTO OU REGIÕES DE ARMAZENAMENTO, SÃO INVALIDADAS. A INVALIDAÇÃO É FACILITADA PELA COLOCAÇÃO DE INDICADORES DE INVALIDAÇÃO SITUADOS EM ENTRADAS DE ESTRUTURAS DE DADOS CORRESPONDENDO ÀS UNIDADES DE ARMAZENAMENTO A INVALIDAR. ALÉM DISSO, ELIMINAM-SE ENTRADAS DE MEMÓRIAS TAMPÃO ASSOCIADAS COM AS UNIDADES DE ARMAZENAMENTO INVALIDADAS OU OUTRAS UNIDADES DE ARMAZENAMENTO ESCOLHIDAS. PROPORCIONA-SE UMA INSTRUÇÃO PARA EFECTUAR A INVALIDAÇÃO E/OU ELIMINAÇÃO. ALÉM DISSO, ELIMINAM-SE ENTRADAS DE MEMÓRIAS TAMPÃO ASSOCIADAS COM UM ESPAÇO DE ENDEREÇOS PARTICULAR, SEM QUALQUER INVALIDAÇÃO. ISTO TAMBÉM É EFECTUADO PELA INSTRUÇÃO. A INSTRUÇÃO PODE SER IMPLEMENTADA EM SOFTWARE, HARDWARE, FIRMWARE OU ALGUMA SUA COMBINAÇÃO OU PODE SER EMULADA.

DESCRIÇÃO

"INVALIDAÇÃO DE ARMAZENAMENTO, ELIMINAÇÃO DE ENTRADAS DE UMA MEMÓRIA TAMPÃO"

Esta invenção refere-se, de um modo geral, ao processamento num ambiente informático e, em particular, a um ou mais de invalidação de armazenamento, eliminação de entradas de uma memória tampão e uma instrução para tal.

Os ambientes informáticos existentes utilizam armazenamento virtual que é, normalmente, mantido em armazenamento auxiliar para aumentar o tamanho do armazenamento disponível. Quando se recorre a uma página de armazenamento virtual, o endereço virtual utilizado quando se recorre à página é traduzido por tradução de endereço dinâmica para um endereço de armazenamento real. Se a tradução for bem sucedida, então, o endereço virtual é válido; caso contrário, é inválido. Uma página de armazenamento virtual é indicada como válida ou inválida por um bit inválido numa entrada de tabela de páginas, em que a entrada inclui um endereço de armazenamento real se a página for válida.

Páginas de armazenamento podem ser invalidadas, uma página de cada vez, como divulgado pelo documento US 6119204. Por exemplo, num ambiente informático com base na arquitectura z/Architecture, proporcionada pela International Business Machines Corporation, uma instrução designada como instrução de Entrada de Tabela de Páginas Invalidada (IPTE), é utilizada para invalidar uma página de armazenamento. Invalidação inclui definir um indicador inválido numa estrutura de controlo,

associada com armazenamento virtual e situada num armazenamento físico principal, para indicar que um endereço virtual de um local no armazenamento virtual não pode ser traduzido por tradução de endereço dinâmica para um endereço de armazenamento físico principal, também denominado um endereço real. Além disso, a instrução é utilizada para eliminar entradas de memórias tampão de máquina interna associadas com a página, em que as entradas podem ser utilizadas durante uma tradução de endereço dinâmica para evitar o acesso às estruturas de controlo no armazenamento físico principal. Por exemplo, na arquitectura z/Architecture, há Memórias Tampão de Tradução de Endereço utilizadas durante a tradução de endereço dinâmica de endereços virtuais para endereços reais. Quando uma página de armazenamento é invalidada, as memórias tampão correspondentes também são desembaraçadas de entradas associadas com essa página.

Também há instruções que permitem a purga de Memórias Tampão de Tradução de Endereço completas. Por exemplo, na arquitectura z/Architecture, uma instrução designada como instrução de Comparar e Trocar e Purgar (CSP), purga todas as entradas de Memórias Tampão de Tradução de Endereço.

Assim, embora haja instruções para invalidar uma página de armazenamento e eliminar memórias tampão de entradas associadas a essa página e instruções para eliminar memórias tampão completas, existe ainda a necessidade de instruções que possam invalidar, selectivamente, vários tamanhos de unidades de armazenamento e/ou eliminar memórias tampão de entradas associadas às mesmas. Por exemplo, existe a necessidade de uma instrução que permita a especificação de uma unidade seleccionada de armazenamento maior do que uma página a

invalidar e/ou efectue a eliminação de entradas de memória tampão associadas à mesma. Existe ainda a necessidade de uma instrução que permita a eliminação de entradas de memória tampão para unidades seleccionadas de armazenamento sem afectar entradas de memória tampão para outras unidades de armazenamento.

A presente invenção proporciona um método de invalidação de armazenamento de um ambiente informático como reivindicado na reivindicação 1.

A presente invenção proporciona, de um modo preferido, um método de invalidação de armazenamento de um ambiente informático. O método inclui, por exemplo, a especificação de uma unidade de armazenamento a invalidar, incluindo, a unidade de armazenamento, uma pluralidade de páginas de armazenamento e, em que a especificação inclui proporcionar um descritor da unidade de armazenamento; e invalidação da unidade de armazenamento.

A presente invenção proporciona, de um modo preferido, um método para facilitar a eliminação de memórias tampão de um ambiente informático. O método inclui, por exemplo, a identificação de uma unidade de armazenamento na qual se vai efectuar a eliminação de uma ou mais entradas de uma ou mais memórias tampão, em que a unidade de armazenamento inclui uma pluralidade de páginas de armazenamento; e eliminar uma ou mais entradas da ou das memórias tampão associadas com a unidade de armazenamento identificada, em que uma ou mais entradas associadas com outra unidade de armazenamento não são eliminadas.

Proporciona-se, de um modo preferido, uma instrução a executar num ambiente informático. A instrução inclui, por exemplo, um código de operação para identificar uma instrução a executar; uma indicação de invalidação para especificar informação associada com uma unidade de armazenamento a invalidar, incluindo a unidade de armazenamento uma pluralidade de páginas de armazenamento; outra indicação para especificar informação adicional a utilizar pela instrução; e uma indicação de eliminação para especificar informação associada com uma de entre a unidade de armazenamento e outra unidade de armazenamento para as quais se pretende eliminar uma ou mais entradas de uma ou mais memórias tampão, em que um ou mais componentes de uma ou mais de entre a indicação de invalidação, a outra indicação e a indicação de eliminação podem ser configurados como não-operações.

Proporciona-se, de um modo preferido, uma instrução a executar num ambiente informático. A instrução inclui, por exemplo, um código de operação para identificar uma instrução a executar; e, pelo menos, uma de entre as seguintes: uma primeira indicação de invalidação para especificar informação associada com uma unidade de armazenamento a invalidar, incluindo a unidade de armazenamento uma aplicação de páginas de armazenamento; uma segunda indicação de invalidação para especificar informação adicional associada com a unidade de armazenamento a invalidar; e uma indicação de eliminação para especificar informação associada com uma de entre a unidade de armazenamento e outra unidade de armazenamento para as quais se pretende eliminar uma ou mais entradas de uma ou mais memórias tampão, em que a informação especificada pela indicação de eliminação não está associada com, pelo menos, uma outra unidade

de armazenamento para a qual não se pretende eliminar uma ou mais entradas.

Também se descrevem e reivindicam, aqui, produtos de sistema e de programa informático correspondentes aos métodos acima resumidos.

Características e vantagens adicionais são obtidas através das técnicas da presente invenção. Outras formas de realização e aspectos da invenção são descritos, neste documento, de modo pormenorizado e são considerados uma parte da invenção reivindicada.

O tema que é considerado como a invenção é, particularmente, salientado e distintamente reivindicado nas reivindicações na conclusão da descrição. Os objectivos acima mencionados e outros objectivos, características e vantagens da invenção resultam da descrição pormenorizada que se segue feita em associação com os desenhos anexos, nos quais:

A FIG. 1 representa uma forma de realização de um ambiente informático incorporando e utilizando um ou mais aspectos da presente invenção;

A FIG. 2 representa uma forma de realização de outros pormenores associados com um controlador da FIG. 1, de acordo com um aspecto da presente invenção;

A FIG. 3 representa uma forma de realização de um computador central que pode emular outro computador, de acordo com um aspecto da presente invenção;

A FIG. 4 representa uma forma de realização da lógica associada com a invalidação de armazenamento, de acordo com um aspecto da presente invenção;

A FIG. 5 representa uma forma de realização da lógica associada com a eliminação de entradas de memória tampão, de acordo com um aspecto da presente invenção;

A FIG. 6a representa uma forma de realização de um formato de uma instrução de Entrada de Tabela de Tradução de Endereço Dinâmica Invalidada (IDTE), de acordo com um aspecto da presente invenção;

A FIG. 6b representa uma forma de realização dos campos associados com o registo R1 geral da instrução IDTE da FIG. 6a, de acordo com um aspecto da presente invenção;

A FIG. 6c representa uma forma de realização dos campos associados com o registo R2 geral da instrução IDTE da FIG. 6a, de acordo com um aspecto da presente invenção;

A FIG. 6d representa uma forma de realização dos campos associados com o registo R3 geral da instrução IDTE da FIG. 6a, de acordo com um aspecto da presente invenção;

As FIGs. 7a-7b representam uma forma de realização da lógica associada com a instrução IDTE, de acordo com um aspecto da presente invenção;

A FIG. 8a representa uma forma de realização dos campos do registo R1 geral utilizado para uma operação alternativa da instrução IDTE, de acordo com um aspecto da presente invenção;

A FIG. 8b representa uma forma de realização dos campos do registo R2 geral utilizado para uma operação alternativa da instrução IDTE, de acordo com um aspecto da presente invenção;

A FIG. 8c representa uma forma de realização dos campos do registo R3 geral utilizado para uma operação alternativa da instrução IDTE, de acordo com um aspecto da presente invenção; e

A FIG. 9 representa uma forma de realização da lógica associada com a operação alternativa da instrução IDTE, de acordo com um aspecto da presente invenção;

De acordo com um aspecto da presente invenção, uma parte de armazenamento especificada (e. g., armazenamento virtual) é invalidada. A título de exemplos, selecciona-se um segmento de armazenamento, que inclui uma pluralidade de páginas de armazenamento ou uma região de armazenamento, que inclui uma pluralidade de segmentos de armazenamento, para invalidação. A invalidação é facilitada pela colocação de indicadores de invalidação situados em entradas de estrutura de dados correspondendo às unidades de armazenamento a invalidar. Num outro aspecto da presente invenção, entradas de memória tampão (e. g., entradas de Memórias Tampão de Tradução de Endereço) associadas com o armazenamento a invalidar ou outro armazenamento, são eliminadas. Ainda noutra aspecto da presente invenção, entradas de memória tampão associadas com espaços de endereços seleccionados são eliminadas de memórias tampão sem afectar entradas de memória tampão de espaços de endereços não seleccionados e sem execução de invalidação. Num exemplo, uma instrução, designada, neste documento, como instrução de Entrada de Tabela (IDTE) de Tradução de Endereço Dinâmica (DAT)

Invalidada, da arquitectura z/Architecture da IBM®, é utilizada para executar uma ou mais das operações acima mencionadas.

Uma forma de realização de um ambiente 100 informático incorporando e utilizando um ou mais aspectos da presente invenção é descrita recorrendo à FIG. 1. O ambiente 100 informático baseia-se, por exemplo, na arquitectura z/Architecture disponibilizada pela International Business Machines Corporation, Armonk, Nova Iorque. A arquitectura z/Architecture é descrita numa publicação da IBM® intitulada "z/Architecture Principles of Operation", Publicação IBM N° SA22-7832-00, de Dezembro de 2000 (IBM® é uma marca registada da International Business Machines Corporation, Armonk, Nova Iorque, E.U.A. Outros nomes aqui utilizados podem ser marcas registadas, marcas comerciais ou nomes de produtos da International Business Machines Corporation ou de outras companhias). Num exemplo, um ambiente informático com base na arquitectura z/Architecture inclui um servidor eServer zSeries disponibilizado pela International Business Machines Corporation, Armonk, Nova Iorque. Num exemplo, o ambiente 100 informático inclui um complexo 102 de processador central (CPC) acoplado a um controlador 120. O complexo 102 de processador central inclui, por exemplo, uma ou mais divisórias 104 (e. g., divisórias lógicas LP1-LPn), um ou mais processadores 106 centrais (e. g., CP1-CPm) e um monitor 108 de máquina virtual (e. g., um gestor de divisórias lógicas), todos descritos em seguida.

Cada divisória 104 lógica está apta a funcionar como um sistema distinto. Ou seja, cada divisória lógica pode ser reiniciada automaticamente, carregada inicialmente com um sistema operativo, se desejado e funcionar com programas

diferentes. Um sistema operativo ou programa de aplicação executado numa divisória lógica aparenta ter acesso a um sistema total e completo, mas, na realidade, apenas uma parte do mesmo está disponível. Uma combinação de hardware e Código Interno Licenciado (designado, normalmente, como microcódigo) impede que um programa numa divisória lógica interfira com um programa numa divisória lógica diferente. Isto permite o funcionamento de várias divisórias lógicas diferentes num único ou em múltiplos processadores físicos de modo dividido no tempo. Neste exemplo particular, cada divisória lógica tem um sistema 110 operativo residente, que pode diferir para uma ou mais divisórias lógicas. Numa forma de realização, o sistema 110 operativo é o sistema operativo z/OS, disponibilizado pela International Business Machines Corporation, Armonk, Nova Iorque.

Os processadores 106 centrais são recursos de processador físico que estão alocados às divisórias lógicas. Por exemplo, uma divisória 104 lógica inclui um ou mais processadores lógicos, cada um dos quais representa todos ou uma partilha de um recurso 106 de processador físico alocado à divisória. Os processadores lógicos de uma divisória 104 lógica particular podem ser dedicados à divisória, para que o recurso de processador subjacente fique reservado para essa divisória ou partilhados com outra divisória, para que o recurso de processador subjacente fique, potencialmente, disponível para outra divisória. As divisórias 104 lógicas são geridas por um monitor 108 de máquina virtual, implementado por microcódigo executado em processadores 106. Cada divisória 104 lógica e o monitor 108 de máquina virtual compreendem um ou mais programas residentes em partes respectivas do armazenamento central associado com os processadores centrais. Um exemplo de monitor 108 de máquina virtual é o Recurso de Processador/Gestor de

Sistemas (PR/SM) disponibilizado pela International Business Machines Corporation, Armonk, Nova Iorque.

O controlador 120, que está acoplado ao complexo de processador central, inclui lógica centralizada responsável pela arbitragem entre processadores diferentes emissores de solicitações. Por exemplo, quando o controlador 120 recebe uma solicitação, determina que o requerente é o processador mestre para essa solicitação e que os outros processadores são processadores escravos; difunde mensagens; e, por outro lado, trata solicitações. Um exemplo de um controlador é descrito na Patente U.S. Nº 6199219. Outros pormenores também são descritos recorrendo à FIG. 2. A FIG. 2 representa outro exemplo de um controlador 200 acoplado a uma pluralidade de processadores 201 centrais (CPU). Neste exemplo, representam-se dois processadores centrais. No entanto, deve compreender-se, que o controlador 200 pode estar acoplado a mais do que dois processadores.

O controlador 200 inclui vários controlos incluindo, por exemplo, controlos 202 de serialização de sistema. Os controlos de serialização de sistema são utilizados para assegurar que as operações a serializar, tais como instruções de invalidação, são serializadas, dado que apenas uma dessas instruções é executada de cada vez no ambiente informático. Também monitoriza a sequência de eventos para essa operação.

O controlador 200 está acoplado a cada processador central por meio de várias interfaces. Por exemplo, uma interface 204 é utilizada pelo Código Interno Licenciado num processador central para enviar comandos de “controlo” para o controlador, que especificam uma acção a tomar e para enviar comandos de “sentido”, que devolvem informação proveniente do controlador.

Outra interface é um barramento 206 de resposta, que é utilizado para devolver informação proveniente do controlador para os comandos de “sentido”. O barramento de resposta também é utilizado para comunicar um estado de comando para comandos de “controlo” e pode ser definido a partir de uma pluralidade de fontes no interior do controlador, incluindo os controlos de serialização de sistema. Um processador central pode utilizar esta interface para sentir o estado dos controlos de serialização de sistema no controlador 200.

Uma outra interface é a interface 208, que é utilizada pelo controlador para enviar comandos para cada CPU. Isto também pode ser controlado por uma pluralidade de fontes no interior do controlador, incluindo controlos 202 de serialização de sistema. Ainda uma outra interface é a interface 210, que fornece sinais a controlos 212 de cache do processador 201 central. Os controlos 212 de cache processam comandos em resposta aos sinais. Num exemplo, os controlos 212 de cache processam comandos que afectam uma ou mais memórias tampão, tais como Memórias Tampão 213 de Tradução de Endereço (TLB), como descrito em seguida em mais pormenor.

Além de controlos 212 de cache, o processador 201 central inclui vários outros controlos, incluindo, por exemplo, controlos 220 de interrupção e controlos 222 de execução. Em resposta a eventos particulares, os controlos 220 de interrupção fazem com que uma interrupção interna fique pendente no CPU, o que, por sua vez, faz com que os controlos 222 de execução suspendam o processamento de instruções de programa no ponto passível de interrupção seguinte. Em resposta à interrupção, os controlos 222 de execução invocam uma rotina de Código Interno Licenciado para fazer com que uma báscula 224 de operação de

difusão permitida autorize os controlos 212 de cache a processarem comandos pendentes.

O processador 201 central também inclui uma báscula 226 de CPU em repouso que indica se o processador central está ou não em repouso.

O ambiente informático descrito acima é apenas um exemplo. Por exemplo, uma ou mais divisórias podem estar a ser executadas em diferentes modos de arquitectura. Além disso, num outro exemplo, o ambiente não precisa de se basear na arquitectura z/Architecture, mas, em vez disso, pode basear-se noutras arquitecturas disponibilizadas pela Intel, Sun Microsystems, bem como por outras. Além disso, um ambiente pode incluir um emulador (e. g., software ou outros mecanismos de emulação), no qual se emula uma arquitectura particular ou seu subconjunto. Num ambiente deste tipo, uma ou mais funções de emulação do emulador podem implementar um ou mais aspectos da presente invenção, mesmo que um computador executando um emulador possa ter uma arquitectura diferente das capacidades a emular. Num exemplo, em modo de emulação, a instrução ou operação específica a emular é descodificada e uma função de emulação apropriada é construída para implementar a instrução ou operação individual.

Pormenores adicionais de um ambiente de emulação são descritos recorrendo à FIG. 3. Num exemplo, um computador 300 central está apto a emular outra arquitectura, computador e/ou capacidades de processamento de outro computador. Por exemplo, o computador 300 central baseia-se numa arquitectura Intel; arquitectura RISC, tal como PowerPC; uma arquitectura SPARC, disponibilizada pela Sun Microsystems; ou outra arquitectura e está apto a emular a arquitectura z/Architecture da IBM® ou outra

arquitectura da IBM® ou outra entidade. O computador 300 central inclui, por exemplo, uma memória 302 para armazenar instruções e dados; uma unidade 304 de obtenção de instruções para obter instruções da memória 302 e para, opcionalmente, proporcionar um armazenamento temporário local às instruções obtidas; uma unidade 306 de descodificação de instruções para receber instruções da unidade 304 de obtenção de instruções e para determinar o tipo de instruções que foram obtidas; e uma unidade 308 de execução de instruções para executar as instruções. A execução pode incluir o carregamento de dados num registo a partir da memória 302; armazenar, de novo, dados na memória provenientes de um registo; ou efectuar algum tipo de operação aritmética ou lógica, como determinado pela unidade de descodificação.

Num exemplo, cada unidade descrita acima é implementada em software. Por exemplo, as operações a executar pelas unidades são implementadas como uma ou mais subrotinas no software emulador. Noutro exemplo, uma ou mais das operações são implementadas em firmware, hardware, software ou alguma sua combinação.

Além disso, embora a FIG. 3 seja descrita relativamente à emulação, o ambiente da FIG. 3 não precisa de ser um ambiente de emulação. Noutro exemplo, executam-se instruções num ambiente nativo e as operações são implementadas em hardware, firmware, software ou alguma sua combinação.

Um ambiente informático pode incluir armazenamento virtual, assim como armazenamento principal. O armazenamento virtual pode exceder em muito o tamanho do armazenamento principal disponível na configuração e é, normalmente, mantido em armazenamento

auxiliar. Considera-se que o armazenamento virtual é constituído por blocos de endereços, denominados páginas. As páginas às quais se recorreu mais recentemente de armazenamento virtual são atribuídas para ocupar blocos de armazenamento físico principal. Quando um utilizador recorre a páginas de armazenamento virtual que não aparecem no armazenamento principal, as páginas virtuais substituem as páginas no armazenamento principal com menos probabilidade de serem utilizadas. A troca de páginas pode ser efectuada pelo sistema operativo sem o conhecimento do utilizador.

Os endereços utilizados para indicar locais no armazenamento virtual são designados como endereços virtuais. Um bloco de endereços virtuais sequenciais ocupando, por exemplo, até 4k bytes, é designado como página. De modo semelhante, um bloco de endereços virtuais sequenciais ocupando, por exemplo, até 1M bytes, é designado como segmento; e um bloco de endereços virtuais sequenciais ocupando, por exemplo, até 2G bytes, é designado como região. Além disso, uma sequência de endereços virtuais associados com armazenamento virtual indicada por um elemento de controlo de espaço de endereços (ASCE) é denominada um espaço de endereços. Podem utilizar-se espaços de endereços para proporcionar graus de isolamento entre utilizadores. Um espaço de endereços pode incluir uma ou mais regiões, um ou mais segmentos, uma ou mais páginas ou alguma sua combinação.

Existem estruturas de dados, associadas a diferentes tipos de unidades de armazenamento (e. g., regiões, segmentos, páginas), a utilizar em processamento associado com as unidades de armazenamento. Por exemplo, existem tabelas de regiões associadas com regiões; tabelas de segmentos associadas com segmentos; e tabelas de páginas associadas com páginas. Estas

tabelas são utilizadas, por exemplo, durante a tradução (e. g., Tradução de Endereço Dinâmica) de um endereço virtual para um endereço real, que é utilizado para aceder ao armazenamento principal. As tabelas a utilizar em tradução, designadas, aqui, como tabelas de tradução, são indicadas por um elemento de controlo de espaço de endereços (ASCE). Isto é descrito em mais pormenor numa publicação da IBM® intitulada "z/Architecture Principles of Operation", Publicação IBM Nº SA22-7832-00, de Dezembro de 2000. Uma unidade de armazenamento virtual que não esteja, actualmente, atribuída ao armazenamento principal é denominada inválida. O estado inválido de uma unidade de armazenamento virtual é indicado por um indicador inválido na estrutura de dados associada com a unidade.

O mecanismo de tradução de endereço dinâmica é implementado, numa forma de realização, para que a informação obtida a partir das tabelas de tradução (e. g., tabelas de regiões, tabelas de segmentos e/ou tabelas de páginas) através do processo DAT seja mantida numa ou mais memórias tampão situadas no interior dos processadores, designadas, aqui, como Memórias Tampão de Tradução de Endereço, de modo a melhorar o desempenho da tradução de endereços. Ou seja, durante a tradução, verifica-se se a informação necessária está presente nas memórias tampão e, depois, se a informação não estiver nas memórias tampão, acede-se a uma ou mais das tabelas de tradução.

Numa forma de realização, considera-se que uma entrada de memória tampão é uma de três tipos: uma entrada de tabelas de regiões e segmentos combinadas de TLB; uma entrada de tabela de páginas de TLB; ou uma entrada de espaço real de TLB, todas descritas em seguida.

Uma entrada de tabelas de regiões e segmentos combinadas de TLB (CRSTE) inclui a informação obtida a partir da entrada ou entradas de tabela em armazenamento real ou absoluto e os atributos utilizados para obter esta informação do armazenamento. Num exemplo, uma entrada de tabelas de regiões e segmentos combinadas de TLB (CRSTE) inclui os seguintes campos:

TO O início de tabela no elemento de controlo de espaço de endereços utilizado para formar a entrada;

RX O índice de região do endereço virtual utilizado para formar a entrada;

SX O índice de segmento do endereço virtual utilizado para formar a entrada;

PTO O início de tabela de páginas obtido a partir da entrada de tabela de segmentos em armazenamento real ou absoluto;

C O bit de segmento comum obtido a partir da entrada de tabela de segmentos em armazenamento real ou absoluto; e

P O bit de protecção de página obtido a partir da entrada de tabela de segmentos em armazenamento real ou absoluto.

V O bit de validade indica se a entrada na TLB é válida e, por conseguinte, pode ser utilizada para consultar entradas subsequentes.

Uma entrada de tabela de páginas de TLB inclui a informação obtida a partir da entrada ou entradas de tabela em armazenamento real ou absoluto e os atributos utilizados para

obter esta informação do armazenamento. Num exemplo, uma entrada de tabela de páginas de TLB inclui:

TO O início de tabela no elemento de controlo de espaço de endereços ou entrada de tabelas de regiões e segmentos combinadas de TLB utilizado para formar a entrada, consoante o modo de formação da entrada;

PTO O início de tabela de páginas utilizado para formar a entrada;

PX O índice de páginas do endereço virtual utilizado para formar a entrada;

PFRA O endereço real de página-trama obtido a partir da entrada de tabela de páginas em armazenamento real ou absoluto; e

P O bit de protecção de página obtido a partir da entrada de tabela de segmentos em armazenamento real ou absoluto.

V O bit de validade indica se a entrada na TLB é válida e, por conseguinte, pode ser utilizada para consultar entradas subsequentes.

Às vezes, pretende-se invalidar uma unidade de armazenamento. Para facilitar esta invalidação, emprega-se a tabela de tradução associada a essa unidade de armazenamento, como descrito aqui. Além disso, em resposta à invalidação do armazenamento, também se podem eliminar entradas de memória tampão correspondentes. Por exemplo, de modo a invalidar uma página de armazenamento, define-se um bit inválido de uma

entrada de tabela de páginas. Além disso, num exemplo, as entradas correspondentes numa ou mais TLB são eliminadas atomicamente. Num exemplo particular, a invalidação e eliminação de entradas associadas a páginas de armazenamento são efectuadas por uma instrução de Entrada de Tabela de Páginas Invalidada (IPTE).

Além de invalidar páginas de armazenamento, de acordo com um aspecto da presente invenção, segmentos de armazenamento e/ou regiões de armazenamento podem ser invalidadas. Num exemplo, utilizam-se tabelas de segmentos e/ou regiões nesta invalidação. Uma forma de realização da lógica associada com a invalidação de uma unidade de armazenamento particular é descrita recorrendo à FIG. 4.

Inicialmente, identifica-se uma tabela de regiões ou segmentos na qual se pretende invalidar uma ou mais entradas PASSO 400. Num exemplo, isto inclui proporcionar um início de tabela de regiões ou um início de tabela de segmentos. Em seguida, PASSO 402, identifica-se uma entrada no interior da tabela de regiões ou segmentos indicada para invalidação e a entrada é invalidada. Num exemplo, a identificação é feita ao proporcionar um índice de região ou índice de segmento (bits de um endereço virtual) que selecciona uma entrada no interior da tabela identificada e a invalidação inclui definir um indicador inválido no interior da entrada. Em resposta à colocação do indicador inválido, invalida-se uma unidade de armazenamento correspondente (e. g., uma região ou segmento). Subsequentemente, INTERROGAÇÃO 406, determina-se se há mais entradas a invalidar. Se houver mais entradas a invalidar, então, o processamento continua no PASSO 402. Caso contrário, o processamento está completo.

Além de invalidar a ou as entradas numa tabela de regiões ou segmentos e, assim, invalidar essas unidades de armazenamento, as entradas correspondentes (ou outras entradas) podem ser eliminadas ou purgadas de uma ou mais memórias tampão num ou mais processadores no interior do ambiente. Uma forma de realização da lógica associada a esta eliminação é descrita em mais pormenor recorrendo à FIG. 5.

Inicialmente, determina-se se um início de tabela de regiões ou segmentos está a ser especificado, INTERROGAÇÃO 500. Se um início de tabela de regiões ou segmentos estiver especificado, então, PASSO 502, uma ou mais memórias tampão são desembaraçadas de uma entrada especificada pelo índice de região ou segmento que coincide com o início de tabela de regiões ou segmentos. Subsequentemente, determina-se se há mais entradas a eliminar, INTERROGAÇÃO 504. Se for esse o caso, então, o processamento continua com o PASSO 502. Caso contrário, o processamento está completo.

Regressando à INTERROGAÇÃO 500, se o início de tabela de regiões ou segmentos não estiver especificado, então, uma ou mais memórias tampão são desembaraçadas de uma entrada especificada pelo índice de região ou segmento, PASSO 508. Subsequentemente, INTERROGAÇÃO 510, determina-se se há mais entradas a eliminar. Se houver mais entradas a eliminar, então, o processamento continua com o PASSO 508. Caso contrário, o processamento está completo.

Num exemplo, a invalidação e eliminação são efectuadas por uma instrução. A instrução invalida as entradas seleccionadas de uma tabela de regiões ou segmentos e purga, atomicamente, pelo

menos, as entradas correspondentes (ou outras entradas) das TLB nos processadores da configuração. Num exemplo, as entradas são purgadas de todas as TLB em todos os processadores, sejam físicos, virtuais ou emulados por software. Se existir uma estrutura TLM multinível, então, todos os níveis são purgados, num exemplo. Se existirem múltiplas divisórias lógicas no sistema, então, as entradas TLB neste e outros processadores formados para a divisória lógica na qual a actual instrução IDTE está a ser executada são purgadas.

Esta instrução pode ser implementada em muitas arquitecturas e pode ser emulada. Como exemplos, a instrução pode ser executada em hardware por um processador; ou por emulação do conjunto de instruções incluindo esta instrução ao executar software num computador tendo um conjunto de instruções nativas diferentes. Num exemplo particular, a instrução é implementada na arquitectura z/Architecture e é designada, aqui, como uma instrução de Entrada de Tabela (IDTE) de Tradução de Endereço Dinâmica (DAT) Invalidada.

A IDTE proporciona, por exemplo, uma opção de invalidação e eliminação, na qual uma ou mais regiões ou segmentos de armazenamento são invalidados por meio de uma ou mais entradas de uma tabela de regiões ou tabela de segmentos e, pelo menos, as entradas correspondentes (ou outras entradas) das TLB (e. g., todas as TLB) na configuração são purgadas. A instrução especifica um conjunto mínimo de entradas de TLB a purgar. Numa outra implementação, podem purgar-se mais ou mesmo todas as entradas de TLB após execução da instrução IDTE.

Uma forma de realização do formato de uma instrução (IDTE) de Entrada de Tabela de DAT Invalidada é descrita recorrendo à

FIG. 6. Num exemplo, uma instrução 600 IDTE inclui, por exemplo, um código 602 de operação que indica a instrução de Entrada de Tabela Invalidada DAT (e. g., 'B98E'x) e uma pluralidade de registos 604-608. Cada registo é descrito em mais pormenor, em seguida, recorrendo às FIGs. 6b-6d.

No que se refere à FIG. 6b, o registo 606, que é um registo R1 geral, tem, por exemplo, um formato de um elemento de controlo de espaço de endereços com um início de tabela (e. g., bits 0-51) e utiliza um controlo de tipo indicação (bits 60 e 61). Os bits 52-59, 62 e 63 são ignorados. Em particular, o R1 inclui, por exemplo, um inicio 601 de tabela de regiões ou tabela de segmentos para invalidação que indica a tabela de tradução para a qual se quer invalidar uma ou mais entradas; e um controlo 612 de tipo indicação (DT) que especifica o tipo dessa tabela (e. g., tabela de regiões ou tabela de segmentos).

O registo 608 (FIG. 6c), que é o registo R2 geral, inclui um primeiro índice 614 de região, especificando uma tabela de regiões de primeiro nível; um segundo índice 616 de região, especificando uma tabela de regiões de segundo nível; e um terceiro índice 618 de região, especificando uma tabela de regiões de terceiro nível. Ou seja, para acomodar grandes endereços virtuais, utilizam-se três níveis de tabelas de regiões. O registador R2 também inclui um índice 620 de segmento, que é um índice numa tabela de segmentos; um indicador 621 de modo, que determina a função a efectuar; e um indicador 622 de entradas adicionais, que indica quantas entradas se quer invalidar. Por exemplo, os bits 53-63 do registo R2 geral incluem um inteiro binário especificando o número de entradas de tabela adicionais a invalidar. Num exemplo, o número de entradas a invalidar é 1-2048, correspondendo a um valor em bits 53-63 de

0-2047. Os bits 44-51 do registo R2 geral são zeros; de outra forma, reconhece-se uma excepção de especificação. Assim, os bits 0-43 do registo R2 geral têm o formato do índice de região e índice de segmento de um endereço virtual. A parte de bits 0-43 utilizada pela DAT para seleccionar uma entrada no tipo de tabela indicada pelo registo R1 geral é denominada Índice de Invalidação Real e é descrito em mais pormenor abaixo.

O registo 604 (FIG. 6d), que é o registo R3 geral, tem, por exemplo, um formato de um elemento de controlo de espaço de endereços com o início de tabela, utilizando-se os bits 0-51 se não for nulo. Este início de tabela é utilizado para seleccionar entradas de TLB a eliminar e o tipo de tabela que indica é independente dos bits 60 e 61 do registo R1 geral. Os bits 52-63 do registo R3 geral são ignorados. Se R3 for zero, todo o conteúdo do registo 0 geral é ignorado. Num exemplo, as entradas a eliminar podem ser iguais ou diferentes das entradas que são invalidadas.

Uma forma de realização da lógica associada com uma operação de invalidação e eliminação é descrita recorrendo às FIGs. 7a-7b. Num exemplo, uma operação de invalidação e eliminação é especificada quando o bit 52 do registo R2 geral é zero. Com esta operação, a entrada de tabela de regiões indicada ou entrada de tabela de segmentos em armazenamento ou um leque de entradas começando com a entrada indicada, é invalidada e as memórias tampão de tradução de endereço (TLB) nos CPU da configuração são desembaraçadas das entradas associadas (ou outras entradas). Num exemplo, todos os níveis das memórias tampão, se forem multinível, são eliminados em todos os CPU.

No que se refere à FIG. 7a, inicialmente, especifica-se um início de tabela de regiões ou tabela de segmentos indicando uma tabela de regiões ou tabela de segmentos a partir da qual se pretende invalidar uma ou mais entradas, PASSO 700. Num exemplo, isto é especificado no registo R1 geral. Além disso, selecciona-se uma entrada de tabela em armazenamento a invalidar, PASSO 702. Num exemplo, indica-se uma entrada de tabela ao adicionar o início de tabela no registo R1 geral e o índice de invalidação real no registo R2 geral e seguindo várias regras, como aqui descrito.

Os bits 60 e 61 do registo R1 geral especificam um tipo de tabela e um índice de invalidação real, do seguinte modo:

Bits 60 e 61 do Reg. R1	Tipo de Tabela	Índice de invalidação Real no Reg. R2
11	Primeira tabela de regiões	Primeiro índice de região (bits 0-10)
10	Segunda tabela de regiões	Segundo índice de região (bits 11-21)
01	Terceira tabela de regiões	Terceiro índice de região (bits 22-32)
00	Tabela de segmentos	Índice de segmento (bits 33-43)

(A parte dos bits 0-43 para a direita do índice de invalidação real é ignorada)

No que se refere à tabela acima, quando os bits 60 e 61 têm o valor binário 11, a parte do primeiro índice de região de R2, em associação com o primeiro início de tabela de regiões de R1,

é utilizada para seleccionar uma entrada numa primeira tabela de regiões. O início de tabela é tratado como um endereço de 64 bits. O endereço de 64 bits da entrada da primeira tabela de regiões em armazenamento real ou absoluto é obtido pela anexação de doze zeros à direita dos bits 0-51 da indicação da primeira tabela de regiões e adição ao primeiro índice de região de três zeros mais à direita e cinquenta zeros mais à esquerda. Esta adição é efectuada de acordo com as regras de aritmética de um endereço de 64 bits, independentemente do modo de endereçamento actual, especificado pelos bits 31 e 32 da actual Palavra de Estado de Programa (PSW). Quando ocorre um transporte de bit de posição zero durante a adição, este é ignorada. O endereço formado a partir destes dois componentes é um endereço real ou absoluto. Todos os 64 bits do endereço são utilizados, independentemente da actual PSW especificar o modo de endereçamento de 24 bits, 31 bits ou 64 bits.

Num exemplo, a entrada completa de primeira tabela de regiões é obtida simultaneamente a partir do armazenamento, como observado por outros CPU. O acesso de obtenção à entrada está sujeito a protecção controlada por chave e o acesso de armazenamento está sujeito a protecção controlada por chave e baixa protecção de endereço. Quando o endereço de armazenamento gerado para obter a entrada de primeira tabela de regiões indica um local não disponível na configuração, reconhece-se uma excepção de endereçamento e a unidade de operação é suprimida.

Quando os bits 60 e 61 têm o valor binário 10, o segundo índice de região de R2, em associação com o início de segunda tabela de regiões de R1, é utilizado para seleccionar uma entrada a partir da segunda tabela de regiões. O processo de consulta da segunda tabela de regiões é semelhante ao processo

de consulta da primeira tabela de regiões. Quando não se reconhecem excepções, a entrada obtida a partir da segunda tabela de regiões indica o início e especifica a distância e comprimento da terceira tabela de regiões correspondente.

Quando os bits 60 e 61 têm o valor binário 01, o terceiro índice de região de R2, em associação com o início de terceira tabela de regiões de R1, é utilizado para seleccionar uma entrada a partir da terceira tabela de regiões. O processo de consulta da segunda tabela de regiões é semelhante ao processo de consulta da primeira tabela de regiões. Quando não se reconhecem excepções, a entrada obtida a partir da terceira tabela de regiões indica o início e especifica a distância e comprimento da tabela de segmentos correspondente.

Quando os bits 60 e 61 têm o valor binário 00, a parte de índice de segmento de R2, em associação com o início de tabela de segmentos de R1, é utilizada para seleccionar uma entrada a partir da tabela de segmentos. O processo de consulta da tabela de segmentos é semelhante ao processo de consulta da primeira tabela de regiões. Quando não se reconhecem excepções, a entrada obtida a partir da tabela de segmentos indica o início da tabela de páginas correspondente.

Num exemplo, a entrada de tabela completa é obtida simultaneamente a partir do armazenamento. Subsequentemente, a entrada é invalidada, PASSO 704. Num exemplo, a entrada é invalidada ao definir para o bit 58 da entrada o binário 1. Ao invalidar esta entrada, a unidade de armazenamento correspondente é invalidada.

Em seguida, determina-se se há mais entradas a invalidar, INTERROGAÇÃO 706. Num exemplo, esta determinação faz-se através da verificação dos bits 53-63 do registo R2 geral. Se nem todos os bits 53-63 do registo R2 geral forem zeros, então, há mais entradas a invalidar. Assim, adiciona-se um 1 ao valor anteriormente utilizado do índice de invalidação real e o processamento continua com o PASSO 700. Isto é repetido para que um número de entradas superior em um ao número especificado pelos bits 53-63 seja invalidado. Um transporte do bit de posição mais à esquerda do índice de invalidação real é ignorado e, neste caso, ocorre o rolamento (depois de se atingir o valor máximo, continua-se desde o início) na tabela. O conteúdo do registo R2 geral permanece inalterado.

Depois da invalidação da ou das entradas, de acordo com um outro aspecto da presente invenção, este CPU elimina entradas seleccionadas da sua TLB e ordena aos outros CPU na configuração que eliminem entradas seleccionadas das suas TLB, PASSO 708. Num exemplo, cada TLB é desembaraçada de, pelo menos, as entradas de Tabelas de Regiões e Segmentos Combinadas (CRSTE) que satisfaçam as seguintes condições:

Coincidência do índice de invalidação real e dos bits à esquerda do mesmo no registo R2 geral e da mesma parte do campo de índice de região e segmento na entrada de TLB. (Numa forma de realização, se estes campos incluírem campos de índice de segmento, podem eliminar-se CRSTE, independentemente dos seus campos de índice de região ou podem eliminar-se todas as CRSTE. Se os campos não incluírem campos de índice de segmento, podem eliminar-se todas as CRSTE.) Deve salientar-se que, quando múltiplas entradas de tabela são invalidadas devido aos bits 53-63 do registo R2 geral, então, o índice de invalidação real é

incrementado e perde-se um transporte do bit de posição mais à esquerda do índice.

Ou R3 é zero ou o campo de início de tabela no registo R3 geral coincide com o campo de início de tabela na entrada de TLB.

Se a entrada invalidada for uma entrada de tabela de segmentos, o campo de início de tabela de páginas na entrada invalidada coincide com o campo de início de tabela de páginas na entrada de TLB. (Numa forma de realização, uma entrada de TLB pode ser eliminada, independentemente do inicio de tabela de páginas na entrada.)

Cada TLB também é desembaraçada de, pelo menos, entradas de tabela de páginas nas quais o campo de início de tabela de páginas coincide com o campo de início de tabela de páginas na (1) entrada de tabela de segmentos invalidada, se uma entrada de tabela de segmentos foi invalidada ou (2), qualquer entrada de tabelas de regiões e segmentos combinadas que tenha sido eliminada (está no conjunto a eliminar) na mesma TLB. (Noutra forma de realização, a implementação pode eliminar uma entrada de tabela de páginas independentemente do inicio de tabela de páginas na entrada. Ou seja, pode eliminar todas as entradas de tabela de páginas.)

Uma forma de realização da lógica associada com eliminação de entradas a partir de uma TLB é descrita recorrendo à FIG. 7b. Como um exemplo, isto é uma continuação do processamento efectuado pela IDTE e é efectuado atomicamente em relação à invalidação, como observado pelo CPU e outros CPU no sistema.

No que se refere à FIG. 7b, inicialmente, determina-se, se R3 é zero, INTERROGAÇÃO 720. Se R3 não for zero, então, um início de tabela de regiões ou tabela de segmentos para eliminação é indicado no registo R3 e, para eliminar uma entrada de TLB, o início de tabela de regiões/segmentos indicado no registo R3 deve coincidir com o início de tabela de regiões/segmentos utilizado para formar a entrada, PASSO 722. Se R3 for zero, PASSO 724, o conteúdo do registo R3 (registo 0) é ignorado e uma entrada é eliminada independentemente do início de tabela de regiões/segmentos utilizado para formar a entrada.

A TLB é, eliminada em todos os processadores no sistema das entradas especificadas pelo índice de tabela de regiões/segmentos especificado no registo R2, PASSO 726. Num exemplo, a TLB é desembaraçada de, pelo menos, as entradas de tabelas de regiões e segmentos combinadas nas quais, por exemplo, o índice de invalidação real e os bits à esquerda do mesmo, no registo R2 geral, e a mesma parte da tabela de índices de região e segmento na entrada de TLB coincidem. Para eliminar uma entrada, num exemplo, o bit V é desligado.

Além disso, se uma entrada de tabela de segmentos estiver a ser invalidada, então, as TLB em todos os processadores no sistema são desembaraçadas de entradas de tabela de páginas nas quais o início de tabela de páginas na entrada é igual ao início de tabela de páginas na TLB, PASSO 728. Além disso, se uma entrada de tabela de segmentos estiver a ser invalidada, as TLB são desembaraçadas de, pelo menos, as entradas de tabela de páginas nas quais o campo de início de tabela de páginas coincide com o campo de início de tabela de páginas em qualquer CRSTE que tenha sido eliminada (e. g., necessariamente) na mesma TLB, PASSO 730. Este processo de eliminação é, PASSO 732,

repetido para um número de entradas superior em um ao número especificado no campo 622 de entradas adicionais da FIG. 6c.

Depois, determina-se se a operação de invalidação e eliminação está completa, INTERROGAÇÃO 734. Num exemplo, a instrução IDTE não está completa no processador emissor até que as entradas de TLB indicadas tenham sido eliminadas no CPU e todos os outros CPU no sistema tenham completado quaisquer acessos de armazenamento utilizando a entrada de TLB especificada nesta instrução. Num exemplo, é o controlador de repouso que indica a este processador emissor se os outros processadores completaram os seus acessos. Ou seja, considera-se que o processador emissor é o processador mestre e quaisquer processadores escravos afectados pela instrução IDTE indicam o seu estado ao controlador. Em resposta ao controlador que recebe o estado apropriado dos processadores escravos, este indica a finalização ao mestre. Se o processamento estiver completo, então, a instrução termina, PASSO 736. Caso contrário, o processador espera durante um predeterminado período de tempo e verifica o estado, mais uma vez, ou recebe informação quanto à finalização, INTERROGAÇÃO 734.

Numa forma de realização, o armazenamento na entrada de tabela de regiões ou segmentos e a eliminação de entradas de TLB pode ou não ocorrer se o bit inválido fosse já um na entrada de tabela de regiões ou segmentos.

Embora na forma de realização descrita acima a eliminação das entradas de TLB seja efectuada subsequentemente à invalidação das entradas desejadas, noutras formas de realização, a eliminação de entradas de TLB pode ocorrer depois de cada invalidação, após um número seleccionado de invalidações, etc.

Num outro aspecto da presente invenção, a instrução IDTE pode, em alternativa, ser utilizada para purgar entradas de TLB associadas a um espaço de endereços (e. g., indicado por um Elemento de Controlo de Espaço de Endereços (ASCE)). Num exemplo, esta opção é indicada ao definir para o bit 52 do registo R2 geral o valor 1. Com esta opção, o conteúdo do registo R1 geral é ignorado (ver FIG. 8a), bem como os bits 0-43 e 53-63 do registo R2 geral (ver FIG. 8b). O conteúdo do registo R3 geral tem o formato de um elemento de controlo de espaço de endereços com o início de tabela, sendo utilizados os bits 0-51 (FIG. 8c). Os bits 52-63 do registo R3 geral são ignorados. R3 pode ser zero ou diferente de zero; ou seja, qualquer registo geral incluindo o registo 0 pode ser indicado. Esta operação não efectua qualquer invalidação de armazenamento, mas elimina das TLB dos CPU na configuração, pelo menos, as entradas de tabelas de regiões e segmentos combinadas e entradas de tabela de páginas nas quais o campo de início de tabela coincide com o campo de início de tabela no registo R3 geral. (Numa outra forma de realização, a implementação pode eliminar uma entrada de tabela de páginas, independentemente do inicio de tabela na entrada. Ou seja, pode eliminar todas as entradas de tabela de páginas.)

Uma forma de realização da lógica associada com uma eliminação por operação ASCE é descrita recorrendo à FIG. 9. Inicialmente, especifica-se um início de tabela de regiões ou tabela de segmentos indicando uma tabela de regiões ou tabela de segmentos nas quais se pretende eliminar entradas de TLB correspondentes, PASSO 900. Num exemplo, isto é especificado em bits 0-51 do registo R3 geral. Em seguida, pelo menos, as entradas de tabelas de regiões e segmentos combinadas e entradas

de tabela de páginas correspondendo ao início de tabela especificado em R3 são eliminadas em todos os processadores no sistema, PASSO 902.

Subsequentemente, determina-se se a execução da instrução IDTE está completa, INTERROGAÇÃO 904. Numa forma de realização, a instrução IDTE não está completa no processador emissor até que as entradas de TLB correspondentes aos parâmetros especificados tenham sido eliminadas no CPU emissor e os outros CPU na configuração tenham completado quaisquer acessos de armazenamento, incluindo a actualização de bits de mudança e referência utilizando entradas de TLB correspondentes aos parâmetros especificados. Se, as entradas tiverem sido eliminadas e os acessos de armazenamento estiverem completos, então, a execução da instrução IDTE está completa, PASSO 906. Caso contrário, o processador espera durante um predeterminado período de tempo e verifica o estado, mais uma vez, ou recebe informação quanto à finalização, INTERROGAÇÃO 904.

Descreveu-se em pormenor, acima, o processamento associado a uma instrução IDTE. Numa forma de realização, as operações não têm, necessariamente, qualquer efeito em entradas de espaço real de TLB. Em seguida, proporciona-se informação adicional, melhoramentos e/ou formas de realização associadas com o processamento de uma forma de realização de uma instrução IDTE.

1. A eliminação selectiva de entradas de TLB pode ser implementada de formas diferentes, consoante o modelo ou forma de realização seleccionada. Por exemplo, de um modo geral, podem eliminar-se mais entradas do que o número mínimo especificado. Quando se efectua a operação de invalidação e eliminação, todas as entradas de tabelas de

regiões e segmentos combinadas (CRSTE) contendo um índice de segmento igual a um índice de invalidação real que é um índice de segmento podem ser eliminadas ou todas as CRSTE podem ser eliminadas quando o índice de invalidação real não é um índice de segmento. Além disso, uma CRSTE ou entrada de tabela de páginas pode ser eliminada independentemente do início de tabela de páginas na entrada. Quando a operação de eliminação por ASCE é efectuada, uma entrada de tabela de páginas pode ser eliminada, independentemente do início de tabela na entrada. Quando qualquer destas operações é efectuada, numa forma de realização, o número mínimo exacto de entradas requerido pode ser eliminado.

2. Um elemento de controlo de espaço de endereços pode conter um início de símbolo de espaço real em vez de um inicio de tabela de regiões ou tabela de segmentos. Um inicio de tabela no registo R3 geral é indistinguível de um inicio de símbolo no registo.

3. Quando a operação de invalidação e eliminação é especificada, a eliminação de entradas de TLB pode aproveitar o inicio de tabela de páginas numa entrada de tabela de segmentos. Por conseguinte, se a entrada de tabela de segmentos, quando se encontra no estado anexado, contiver um inicio de tabela de páginas diferente do valor actual, cópias de entradas contendo os valores anteriores podem permanecer na TLB.

4. Quando a operação de invalidação e eliminação é especificada, o endereço da entrada de tabela DAT para a IDTE é um endereço de 64 bits e a aritmética de endereços é efectuada seguindo as regras normais para a aritmética de

endereços de 64 bits, com rolamento em $2^{64} - 1$. De igual modo, os campos de distância e comprimento não são utilizados. Consequentemente, as tabelas DAT não são especificadas para continuar a operação de localização de armazenamento máxima para a localização 0 e a primeira entrada indicada e todas as entradas adicionais especificadas pelos bits 63-63 do registo R3 geral devem situar-se dentro da tabela indicada.

5. A instrução IDTE está sujeita a intercepção sujeita à SIE. Numa forma de realização, o sistema é serializado para que o processador que executa a instrução IDTE não possa iniciar a operação até que outros processadores tenham parado de utilizar as entradas de TLB a especificar. Além disso, o processador que executa a instrução IDTE não continua a prosseguir a execução até que as entradas apropriadas tenham sido eliminadas das TLB de todos os processadores no sistema.

6. Uma função de serialização é efectuada antes do início da operação e, de novo, depois de a operação estar completa. A serialização aplica-se a este CPU; outros CPU não são necessariamente serializados.

Descreveu-se em pormenor, acima, um exemplo de uma instrução que pode invalidar entradas de uma tabela de segmentos ou de regiões e eliminar, atomicamente, pelo menos, as entradas correspondentes (ou outras entradas) de uma ou mais memórias tampão de tradução de endereço ditas TLB. Embora seja aqui descrita uma instrução particular, podem fazer-se muitas variações sem divergir do espírito da presente invenção. Por exemplo, podem utilizar-se códigos de operação diferentes,

campos diferentes, registos diferentes, ou mesmo, não utilizar registos, bits diferentes, etc. Por exemplo, outras arquitecturas de conjuntos de instruções podem definir uma instrução equivalente em modos ligeiramente diferentes (e. g., códigos de operação diferentes, campos diferentes na instrução, registos diferentes utilizados, etc.), mas um ou mais aspectos da presente invenção ainda se aplicam. Além disso, como descrito acima, um ou mais aspectos da presente invenção aplicam-se igualmente bem a uma emulação de software de uma arquitectura visada, tal como a arquitectura z/Architecture da IBM. Neste caso, as TLB discutidas acima podem ser uma matriz ou outra estrutura de dados implementada pelo emulador de software para guardar mapeamentos recentes de endereços virtuais para reais. Também pode acontecer que o software emulador não implemente qualquer estrutura análoga a uma TLB. Neste caso, só se pode aplicar a invalidação de entradas em armazenamento.

Assim, proporciona-se, aqui, uma instrução que está apta a invalidar armazenamento, invalidar armazenamento e eliminar, atomicamente, entradas de memórias tampão ou eliminar entradas de memórias tampão. Por conseguinte, afirma-se que um ou mais registos da instrução podem ser configurados como não operações (*no-ops*). Ou seja, a informação nesses registos, se houver, associada com uma operação particular (e. g., invalidação, eliminação, etc.) que não está a ser proporcionada é ignorada ou indica que a operação não está a ser efectuada.

Embora se tenha descrito acima uma instrução particular na qual se proporciona uma invalidação, invalidação e eliminação ou eliminação, a instrução pode ser modificada ou podem implementar-se outras instruções que efectuem apenas um ou mais aspectos das operações implementadas por IDTE. Por exemplo, uma

instrução pode implementar apenas a invalidação de armazenamento. Num outro exemplo, a instrução pode apenas purgar entradas de memórias tampão que estejam associadas com tabelas de segmentos ou regiões particulares. Ainda noutro exemplo, pode implementar-se uma instrução que purgue apenas entradas com base no espaço de endereços.

Numa implementação comercial do formato de instrução de computador arquitectado da função de invalidação e/ou eliminação, as instruções são utilizadas por programadores, tais como programadores de sistema operativo que escrevem em linguagem *assembler*. Estes formatos de instrução armazenados num meio de armazenamento podem ser executados nativamente num Servidor IBM de arquitectura z/Architecture ou, em alternativa, em máquinas que executem outras arquitecturas. Podem ser emulados nos servidores de processadores centrais da IBM existentes e futuros e noutras máquinas da IBM (e. g., Servidores pSeries e Servidores xSeries). Podem ser executados em máquinas executando Linux num leque variado de máquinas utilizando hardware fabricado pela IBM, Intel, AMD, Sun Microsystems e outros. Para além da execução no hardware sujeito à arquitectura z/Architecture, também se pode utilizar Linux, bem como máquinas que utilizam emulação por Hercules, UMX, FXI ou Platform Solutions, nas quais a execução é feita, de um modo geral, num modo de emulação. Em modo de emulação, a instrução específica a ser emulada é descodificada e constrói-se uma subrotina para implementar a instrução individual, como numa subrotina ou programa de controlo C ou utiliza-se uma qualquer outra técnica para dotar o hardware específico com um programa de controlo, o que está ao alcance do especialista na técnica depois de compreender a descrição de uma forma de realização da invenção. Várias patentes de emulação de software e hardware

incluindo, sem limitação, a Patente U.S. Nº 5551013, Patente U.S. Nº 6009261, Patente U.S. Nº 5574873, Patente U.S. Nº 6308255, Patente U.S. Nº 6463582 e Patente U.S. Nº 5790825 ilustram uma variedade de modos conhecidos para obter emulação de um formato de instrução arquitectado para uma máquina diferente para uma máquina alvo disponível ao especialista na técnica, bem como as técnicas de software comercial utilizadas pelos referenciados acima.

De um modo vantajoso, um ou mais aspectos da presente invenção melhoram o desempenho do sistema. Por exemplo, ao purgar selectivamente entradas de memórias tampão em vez de purgar totalmente as memórias tampão, o desempenho do sistema é melhorado pelo facto de não requerer uma recriação desnecessária de entradas. Ou seja, requer 20+ a 100s ciclos para criar uma entrada de TLB e a purga selectiva evita a recriação desnecessária de entradas.

Como uma vantagem adicional, se se desejar uma purga total das TLB, é que a implementação permite uma purga em excesso. Além disso, a parte de TLB pode ser tratada como uma no-op (não operação) se, por exemplo, não se utilizar uma TLB. De um modo vantajoso, a instrução IDTE pode ser implementada em hardware, firmware, software ou uma sua combinação ou pode ser utilizada num emulador de um conjunto de instruções de processamento alvo.

São possíveis muitas variações do ambiente acima sem divergir do espírito da presente invenção. Por exemplo, um ou mais aspectos da presente invenção são igualmente aplicáveis a, por exemplo, uma emulação de máquina virtual, na qual uma ou mais entidades pagináveis (e. g., convidados) efectuam a execução num ou mais processadores. Num exemplo, convidados

pagináveis são definidos pela arquitectura de Execução de Início de Interpretação (SIE), cujo exemplo é descrito numa publicação da IBM® intitulada "IBM System/370 Extended Architecture", Publicação IBM, Nº SA22-7095 (1985).

Embora se mencione acima a SIE e arquitectura z/Architecture, um ou mais aspectos da presente invenção são igualmente aplicáveis a outras arquitecturas e/ou ambientes empregando entidades pagináveis ou construções semelhantes.

Além disso, as várias formas de realização descritas acima são apenas exemplos. Pode haver muitas variações destas formas de realização sem divergir do espírito da presente invenção. Por exemplo, embora seja aqui descrito um ambiente dividido logicamente, isto é apenas um exemplo. Aspectos da invenção são benéficos para muitos tipos de ambientes, incluindo outros ambientes que tenham uma pluralidade de zonas e ambientes não divididos. Além disso, podem não existir complexos de processadores centrais, mas, mesmo assim, múltiplos processadores acoplados entre si. Ainda, além disso, um ou mais aspectos da invenção são aplicáveis a ambientes de processadores individuais.

Embora se descrevam, aqui, ambientes particulares, podem implementar-se, de novo, muitas variações a estes ambientes sem divergir do espírito da presente invenção. Por exemplo, se o ambiente for dividido logicamente, então, podem incluir-se no ambiente mais ou menos divisórias lógicas. Além disso, podem existir múltiplos complexos de processamento central acoplados entre si. Estas são apenas algumas das variações que se podem fazer. Além disso, são possíveis outras variações. Por exemplo, embora o controlador aqui descrito serialize a instrução para

que uma instrução IDTE seja executada duma só vez, noutra forma de realização, múltiplas instruções podem ser executadas duma só vez. Além disso, o ambiente pode incluir múltiplos controladores. Ainda além disso, múltiplas solicitações de repouso (provenientes de um ou mais controladores) podem sobressair concorrentemente no sistema. Também são possíveis variações adicionais.

O termo “unidade de processamento”, como aqui utilizado, inclui entidades pagináveis, tais como convidados; processadores; emuladores; e/ou outros componentes semelhantes. Além disso, o termo “por uma unidade de processamento” inclui por conta de uma unidade de processamento. O termo “memória tampão” inclui uma área de armazenamento, bem como diferentes tipos de estruturas de dados incluindo, sem limitação, matrizes; e o termo “tabela” pode incluir outras estruturas para além de estruturas de dados de tipo tabela. Além disso, a instrução pode incluir regtos para além de regtos para indicar informação. Além disso, uma página, um segmento e/ou uma região podem ter tamanhos diferentes dos descritos aqui.

Uma ou mais das capacidades da presente invenção podem ser implementadas em software, firmware, hardware ou alguma sua combinação. Além disso, uma ou mais das capacidades podem ser emuladas.

Um ou mais aspectos da presente invenção podem ser incluídos num artigo de fabrico (e. g., um ou mais produtos de programas informáticos) tendo, por exemplo, meios utilizáveis por computador. O meio tem incorporados no seu interior, por exemplo, meios de códigos de programa legíveis por computador ou lógica (e. g., instruções, código, comandos, etc.) para

proporcionar e facilitar as capacidades da presente invenção. O artigo de fabrico pode ser incluído como uma parte de um sistema informático ou vendido à parte.

Além disso, pode proporcionar-se um dispositivo de armazenamento de programas legível por uma máquina incorporando, pelo menos, um programa de instruções executável pela máquina para efectivar as capacidades da presente invenção.

Os fluxogramas aqui ilustrados são apenas exemplos. Pode haver muitas variações destes diagramas ou dos passos (ou operações) aqui descritos. Por exemplo, os passos podem ser efectuados com uma ordem diferente ou podem adicionar-se, apagar ou modificar passos.

Lisboa, 9 de Dezembro de 2010

REIVINDICAÇÕES

1. Método para invalidar um conjunto de duas ou mais entradas de tabela de tradução de endereço num sistema informático tendo tabelas de tradução de endereço para traduzir, dinamicamente, endereços virtuais para endereços de armazenamento principal, compreendendo o método os passos:

determinar, a partir de um código de operação de uma instrução executável por máquina a executar, que a instrução está configurada para iniciar a execução de uma operação de invalidação e eliminação;

executar a instrução, compreendendo o passo de execução os seguintes passos:

determinar, a partir de informação fornecida pela instrução, um endereço de primeira entrada de tabela de tradução de uma primeira entrada de tabela de tradução do leque de duas ou mais entradas de tabela de tradução de endereço a invalidar;

determinar, a partir de informação de conjunto fornecida pela instrução, um número de entradas de tabela de tradução de endereço a invalidar; e

com base no endereço de primeira entrada de tabela de tradução de endereço, invalidar o conjunto de duas ou mais entradas de tabela de tradução de endereço indicado pela informação de conjunto.

2. Método de acordo com a reivindicação 1, em que o conjunto de duas ou mais entradas de tabela de tradução de endereço a invalidar compreende, para cada, um bit inválido de invalidação, em que, quando o bit inválido é 0, a utilização da entrada de tabela de tradução de endereço para tradução dinâmica é permitida, em que, quando o bit inválido é 1, a utilização da entrada de tabela de tradução de endereço para tradução dinâmica não é permitida.
3. Método de acordo com a reivindicação 1, em que o conjunto de duas ou mais entradas de tabela de tradução de endereço é constituído por qualquer uma de entre as entradas de tabela de segmentos ou entradas de tabela de regiões.
4. Método de acordo com a reivindicação 1, em que o sistema informático compreende processadores centrais (CPU), registos de utilização geral arquitectados, memórias tampão de tradução de endereço, tendo as memórias tampão de tradução de endereço entradas de memórias tampão de tradução de endereço contendo informação de tradução de endereço, consistindo as tabelas de tradução de endereço em qualquer um de:

uma ou mais tabelas de segmentos contendo entradas apontando para tabelas de páginas,

uma ou mais primeiras tabelas de regiões contendo entradas apontando para segundas tabelas de regiões, contendo as segundas tabelas de regiões entradas apontando para terceiras tabelas de regiões, contendo as terceiras tabelas de regiões entradas apontando

para tabelas de segmentos, contendo as tabelas de segmentos entradas apontando para tabelas de páginas,

uma ou mais segundas tabelas de regiões contendo entradas apontando para terceiras tabelas de regiões, contendo as terceiras tabelas de regiões entradas apontando para tabelas de segmentos, contendo as tabelas de segmentos entradas apontando para tabelas de páginas,

uma ou mais terceiras tabelas de regiões contendo entradas apontando para tabelas de segmentos, contendo as tabelas de segmentos entradas apontando para tabelas de páginas; e

compreendendo a determinação, a partir do passo de código de operação, o passo adicional seguinte:

recuperar uma segunda informação a partir de uma localização especificada pela instrução, compreendendo a segunda informação uma indicação da operação de invalidação e eliminação.

5. Método de acordo com a reivindicação 4, em que a segunda informação indica a função de invalidação e eliminação quando o bit 52 da segunda informação é 0.
6. Método de acordo com a reivindicação 1, em que o sistema informático compreende, ainda, uma ou mais memórias tampão de tradução de endereço para colocar na cache uma informação de tradução de endereço, compreendendo o método o seguinte passo adicional:

- d) desembaraçar as memórias tampão de tradução de endereço das entradas de memórias tampão de tradução de endereço associadas com entradas de tabela de tradução de endereço do conjunto de duas ou mais entradas de tabela de tradução de endereço.
7. Método de acordo com a reivindicação 6, em que as memórias tampão de tradução de endereço são constituídas por uma ou mais memórias tampão de tradução de endereço (TLB).
8. Método de acordo com a reivindicação 6, em que as memórias tampão de tradução de endereço compreendem qualquer um de:
- uma memória tampão de tradução de endereço associada com a unidade de processamento central,
- uma memória tampão de tradução de endereço associada com outra unidade de processamento central do sistema informático ou
- todas as memórias tampão de tradução de endereço associadas com o sistema informático.
9. Método de acordo com a reivindicação 1, em que o passo de determinação a partir de um código de operação compreende os passos adicionais de:
- obter a instrução do armazenamento principal, compreendendo a instrução:

a) um primeiro campo para localizar uma terceira informação, compreendendo a terceira informação uma localização de início de tabela de tradução de endereço para invalidação e

b) um segundo campo para localizar uma quarta informação, compreendendo a quarta informação uma parte de índice de tabela de tradução de endereço; e

formar o endereço da primeira entrada de tabela de tradução a partir da terceira informação em associação com a quarta informação.

10. Método de acordo com a reivindicação 9, em que o primeiro campo localiza a terceira informação num primeiro registo de utilização geral arquitectado, em que a localização do início de tabela de tradução de endereço para invalidação consiste em qualquer um de entre um endereço de uma tabela de regiões ou um endereços de uma tabela de segmentos.
11. Método de acordo com a reivindicação 9, em que a terceira informação compreende, ainda, um campo de controlo de tipo indicação para indicar qualquer um de entre uma região para invalidação ou uma tabela de segmentos para invalidação.
12. Método de acordo com a reivindicação 9, em que o segundo campo localiza a quarta informação, compreendendo a quarta informação um campo de primeiro índice de região, um campo de segundo índice de região, um campo de terceiro índice de região, um campo de índice de segmento e o conjunto.

13. Método de acordo com a reivindicação 1, em que a instrução é constituída por 32 bits contíguos, incluindo um campo de código de operação constituído por 16 bits contíguos, um primeiro campo operando constituído por 4 bits contíguos, um segundo campo operando constituído por 4 bits contíguos e um terceiro campo operando constituído por 4 bits contíguos, em que os bits contíguos do primeiro campo são contíguos com os bits contíguos do segundo campo e, em que os campos operandos especificam respectivos registo de utilização geral de um espaço de registo de computador arquitectado.
14. Método de acordo com a reivindicação 1, em que o passo de invalidação do conjunto de duas ou mais entradas de tabela de tradução de endereço compreende o passo adicional de marcar como inválido um bit inválido de cada entrada de tabela de tradução de endereço do conjunto de duas ou mais entradas de tabela de tradução de endereço.
15. Método de acordo com a reivindicação 1, em que o sistema informático compreende, ainda, tabelas de páginas, tendo as tabelas de páginas endereços de armazenamento principal reais, em que os endereços de armazenamento principal reais compreendem qualquer um de entre um endereço absoluto, um endereço real ou um endereço físico.
16. Programa de computador compreendendo instruções para realizar todos os passos do método de acordo com qualquer reivindicação de método anterior, quando o referido programa de computador é executado num sistema informático.

17. Sistema compreendendo meios adaptados para realizar todos os passos do método de acordo com qualquer reivindicação de método anterior.
18. Sistema da reivindicação 17 compreendendo:

um armazenamento principal;

um ou mais processadores centrais em comunicação com o armazenamento principal;

registos de utilização geral arquitectados acessíveis ao ou aos processadores centrais;

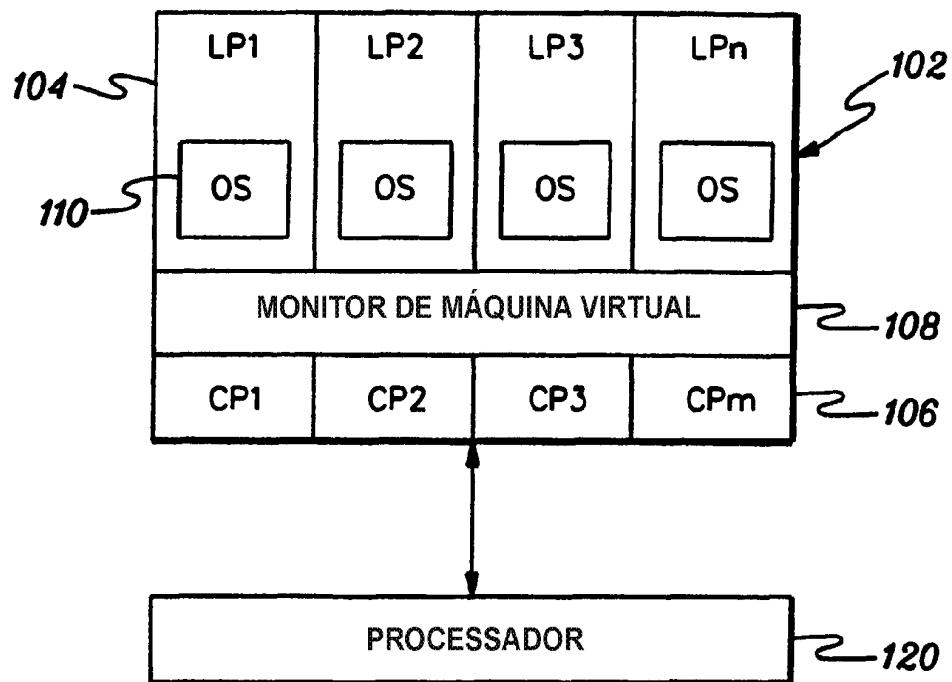
tabelas de tradução de endereço para traduzir, dinamicamente, endereços virtuais para endereços de armazenamento principal, sendo as tabelas de tradução de endereço acessíveis ao ou aos processadores centrais; e

uma ou mais unidades de execução de instruções acessíveis ao ou aos processadores centrais, estando as unidades de execução de instruções em comunicação com os registos de utilização geral arquitectados, executando as unidades de execução de instruções, instruções de máquina obtidas a partir do armazenamento principal.

Lisboa, 9 de Dezembro de 2010

100

COMPLEXO DE PROCESSADOR CENTRAL

*fig. 1*

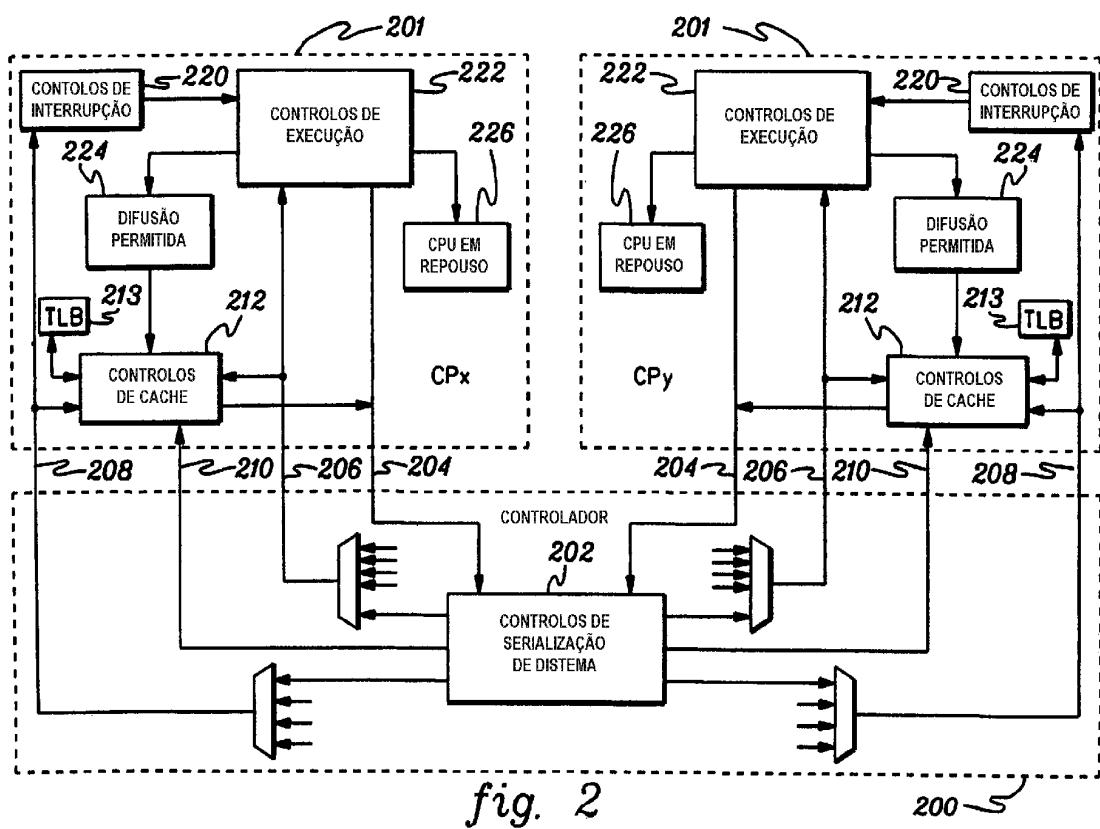


fig. 2

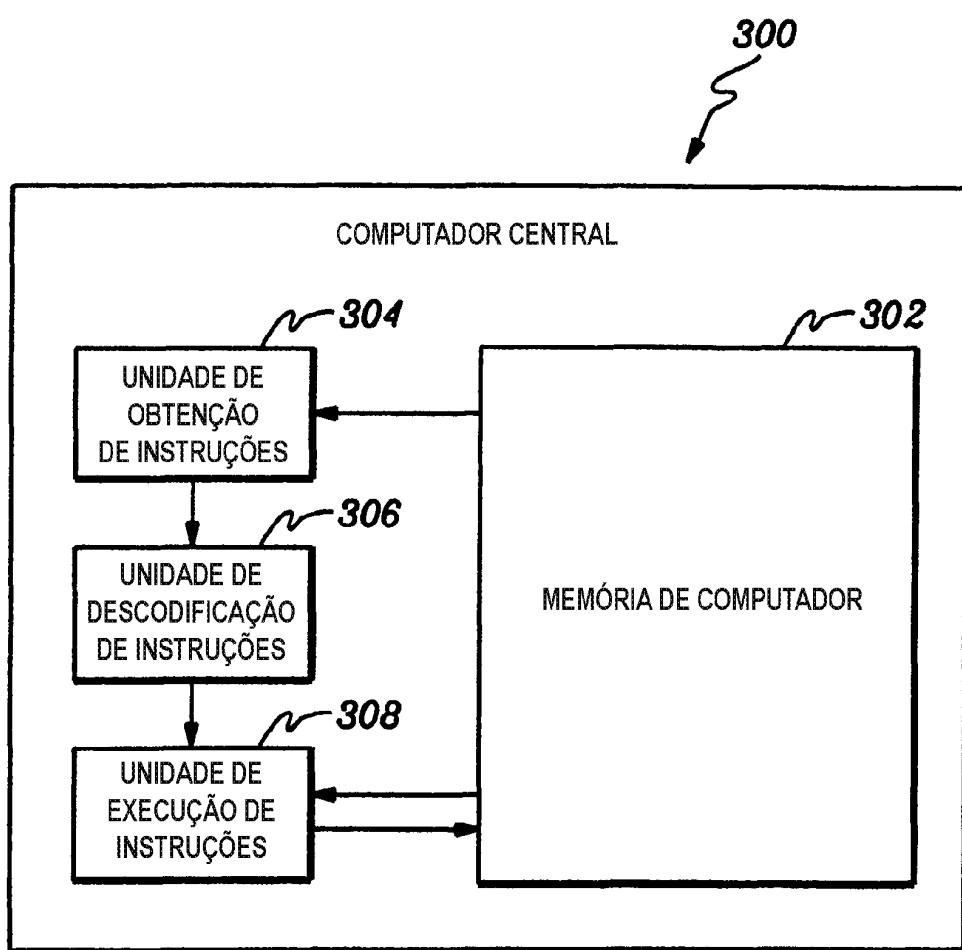


fig. 3

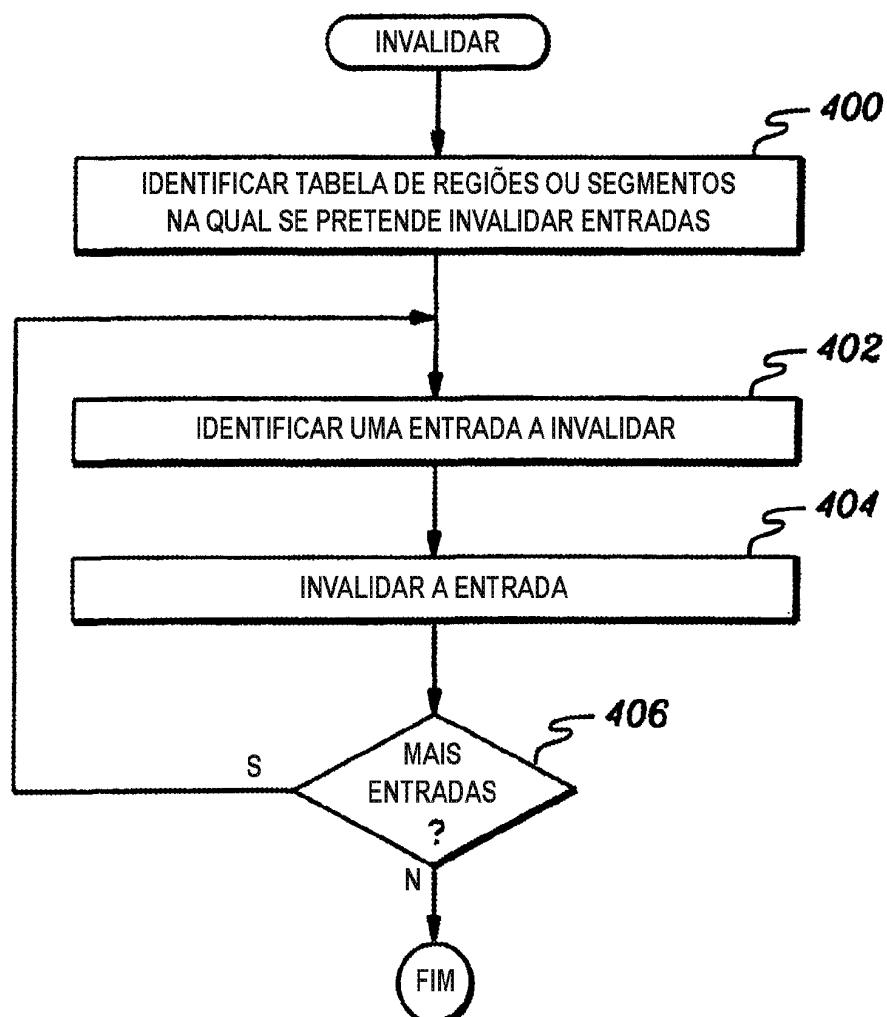


fig. 4

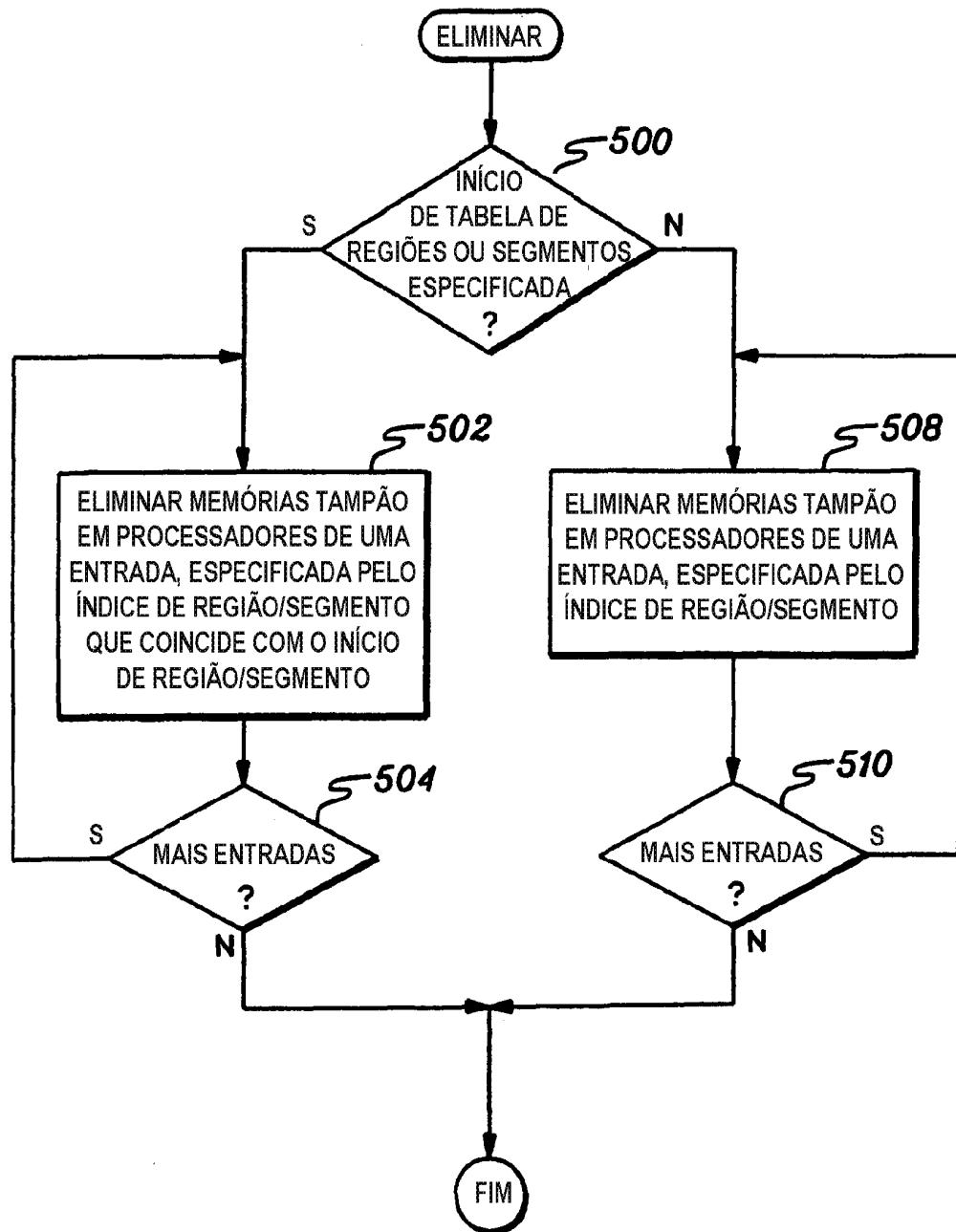


fig. 5

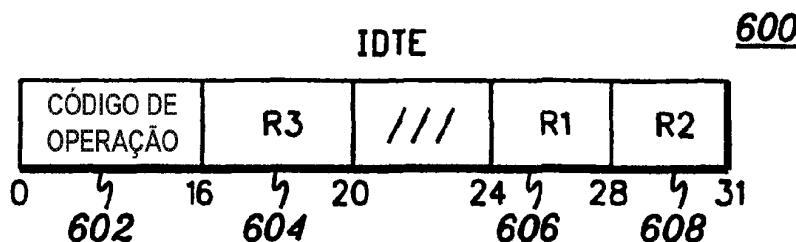


fig. 6A

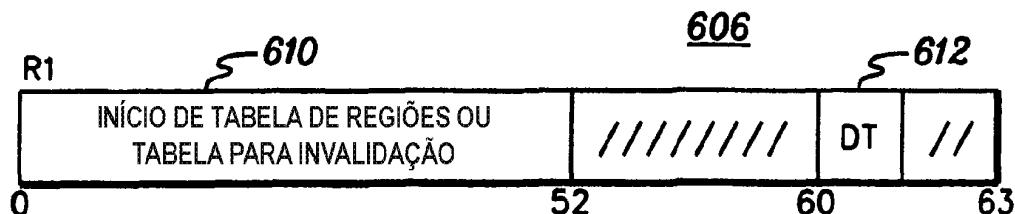


fig. 6B

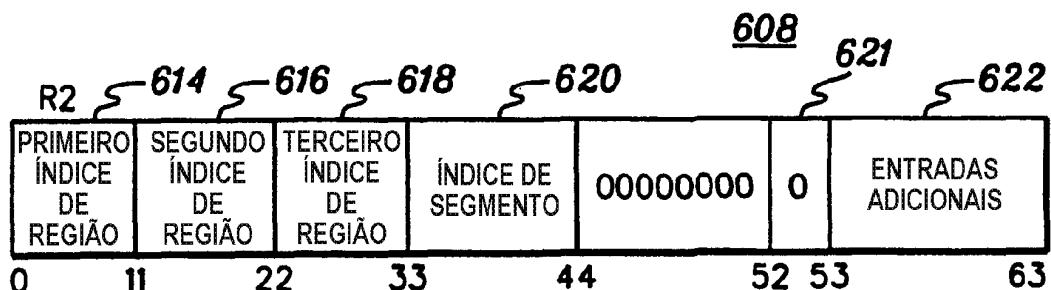


fig. 6C

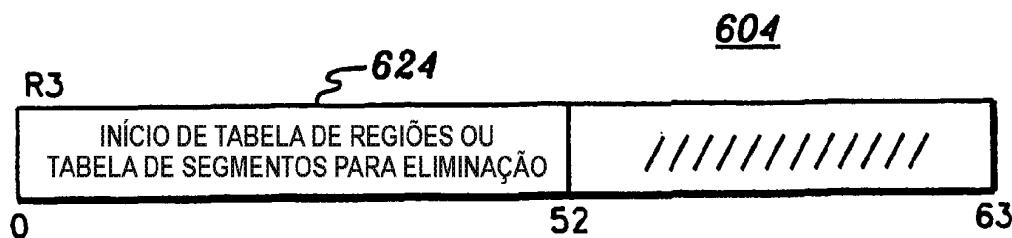


fig. 6D

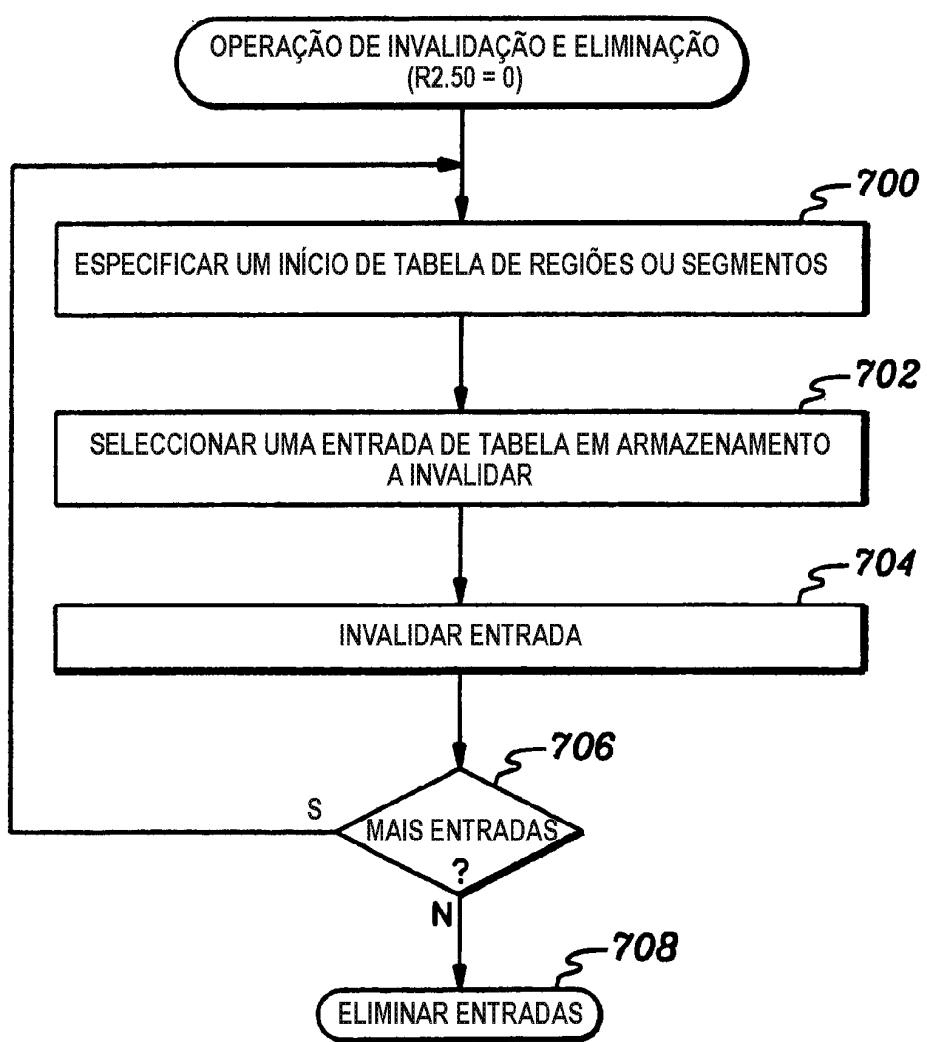
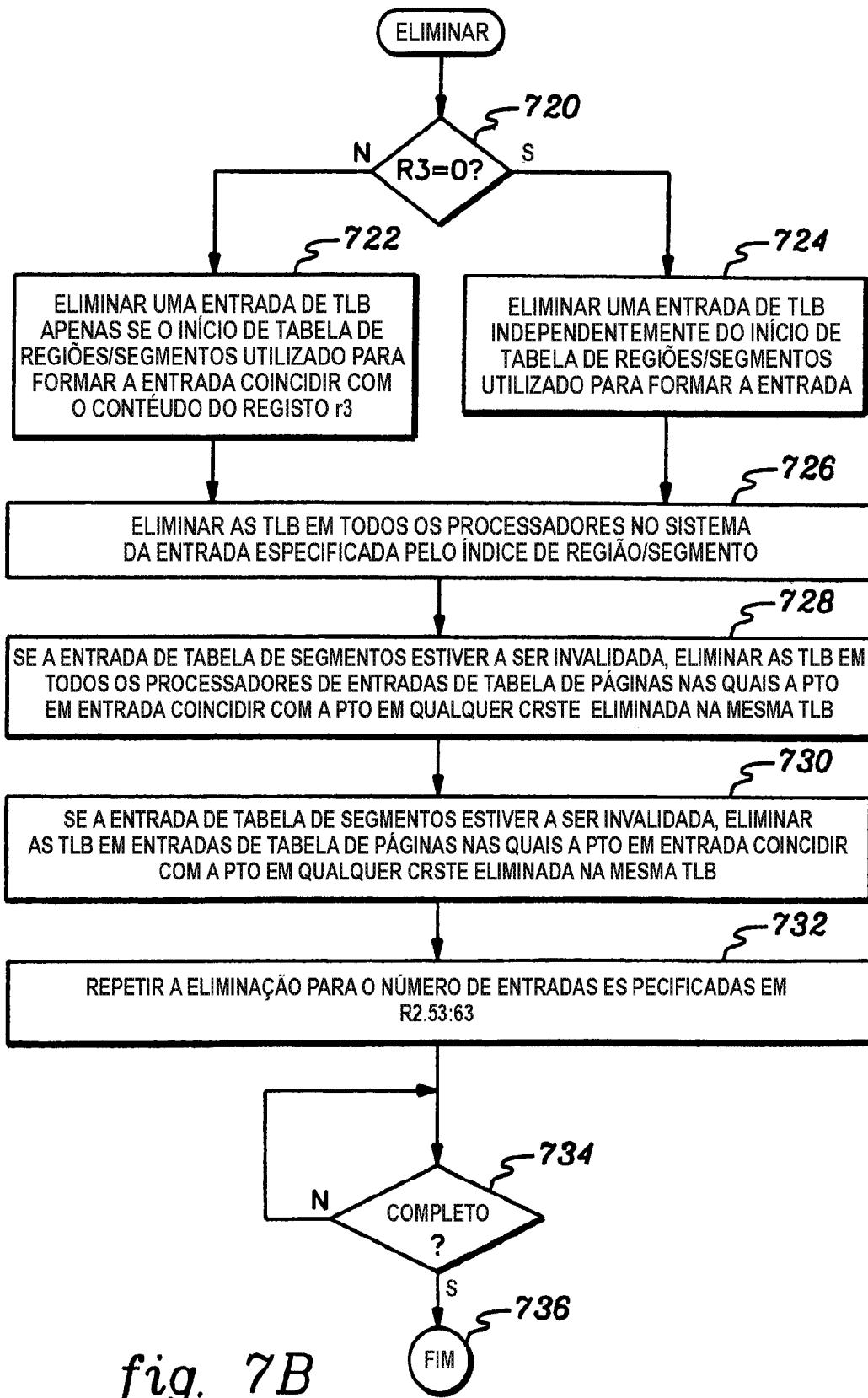


fig. 7A



R1



fig. 8A

R2



fig. 8B

R3

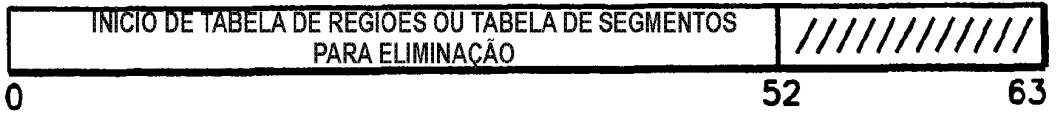


fig. 8C

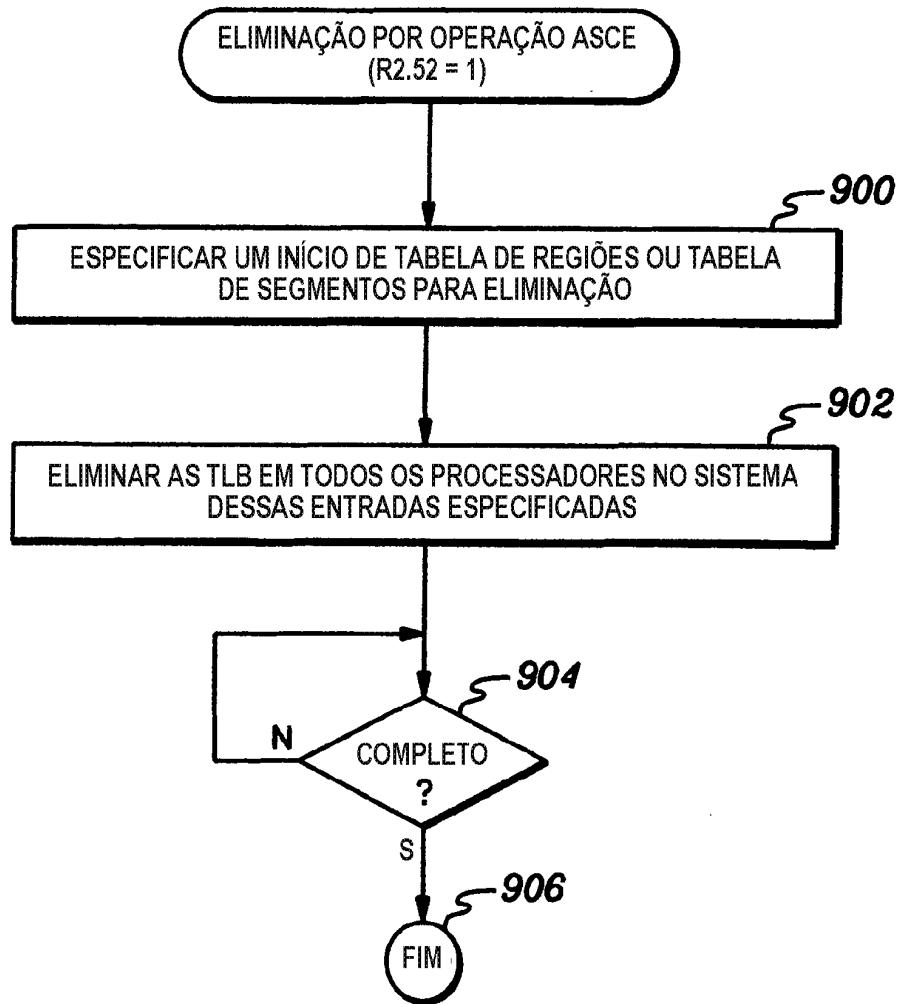


fig. 9

RESUMO

"INVALIDAÇÃO DE ARMAZENAMENTO, ELIMINAÇÃO DE ENTRADAS DE UMA MEMÓRIA TAMPÃO"

Unidades de armazenamento seleccionadas, tais como segmentos de armazenamento ou regiões de armazenamento, são invalidadas. A invalidação é facilitada pela colocação de indicadores de invalidação situados em entradas de estruturas de dados correspondendo às unidades de armazenamento a invalidar. Além disso, eliminam-se entradas de memórias tampão associadas com as unidades de armazenamento invalidadas ou outras unidades de armazenamento escolhidas. Proporciona-se uma instrução para efectuar a invalidação e/ou eliminação. Além disso, eliminam-se entradas de memórias tampão associadas com um espaço de endereços particular, sem qualquer invalidação. Isto também é efectuado pela instrução. A instrução pode ser implementada em software, hardware, firmware ou alguma sua combinação ou pode ser emulada.