



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

(21) **PI0707149-3 A2**



\* B R P I O 7 0 7 1 4 9 A 2 \*

(22) Data de Depósito: 22/01/2007  
(43) Data da Publicação: 19/04/2011  
(RPI 2102)

(51) *Int.Cl.:*  
G06F 9/38  
G06F 12/08

(54) Título: **GERENCIAMENTO EFICIENTE DE HIERARQUIA DE MEMÓRIA**

(30) Prioridade Unionista: 20/01/2006 US 11/336,282

(73) Titular(es): QUALCOMM INCORPORATED

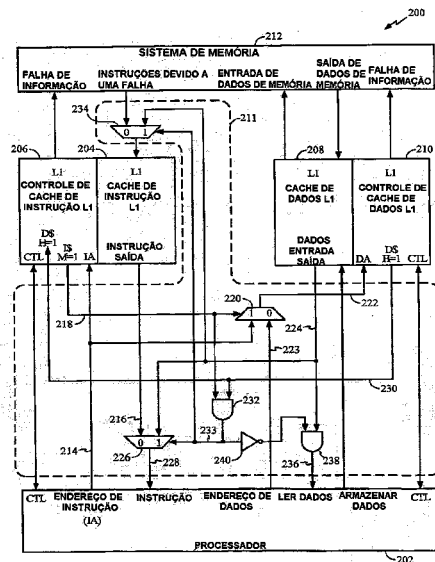
(72) Inventor(es): Michael William Morrow, Thomas Andrew Sartorius

(74) Procurador(es): MONTAURY PIMENTA MACHADO & LIOCE

(86) Pedido Internacional: PCT US2007060815 de 22/01/2007

(87) Publicação Internacional: WO 2007/085011 de 26/07/2007

(57) **Resumo:** GERENCIAMENTO EFICIENTE DE HIERARQUIA DE MEMÓRIA. Em um processador, existem situações onde instruções e certas partes de um programa podem residir em uma cache de dados antes da execução do programa. Técnicas de hardware e software são providas para buscar uma instrução na cache de dados após ter uma falha em uma cache de instrução para melhorar o desempenho do processador. Se uma instrução não estiver presente na cache de instrução, um endereço de busca de instrução é enviado como um endereço de busca de dados para a cache de dados. Se houver dados válidos presentes na cache de dados no endereço de busca de instrução fornecido, os dados na realidade são uma instrução e a entrada de cache de dados é buscada e fornecida como uma instrução para o complexo de processador. Um bit adicional pode ser incluído em uma tabela de página de instrução para indicar uma falha na cache de instrução e que a cache de dados deve ser verificada para a instrução.





PI0707149-3

## "GERENCIAMENTO EFICIENTE DE HIERARQUIA DE MEMÓRIA"

### CAMPO

A presente revelação se refere geralmente às técnicas para buscar instruções a partir da memória tendo  
5 uma cache de instrução e uma cache de dados e, mais especificamente, a uma abordagem aperfeiçoada para buscar uma instrução após uma falha na cache de instrução mediante ação de diretamente buscar a instrução a partir da cache de dados se a instrução residir nesse lugar.

10

### FUNDAMENTOS

Produtos comumente portáteis, tais como telefones celulares, computadores laptop, assistentes pessoais de dados (PDAs) ou semelhantes, requerem o uso de um processador executando programas, tal como, programas de  
15 comunicação e de multimídia. O sistema de processamento para tais produtos inclui um complexo de processador e memória para armazenar instruções e dados. Por exemplo, as instruções e os dados podem ser armazenados em uma memória hierárquica consistindo em múltiplos níveis de caches,  
20 incluindo, por exemplo, uma cache de instrução, uma cache de dados, e uma memória de sistema. O uso de uma cache de instrução separada e uma cache de dados separada é conhecido como uma arquitetura Harvard. Como a arquitetura Harvard isola a cache de instruções da cache de dados,  
25 podem surgir problemas quando as instruções são armazenadas na cache de dados.

Em geral os sistemas processando com uma arquitetura Harvard, existem situações que surgem nas quais as instruções podem ser armazenadas na cache de dados. Por  
30 exemplo, se um programa é criptografado ou está em uma forma compactada, ele deve ser decodificado/descompactado antes de permitir a execução pelo programa. O processo de decodificação/descompactação trata o programá

codificado/compactado como dados para processar os mesmos e armazena as instruções decodificadas/descompactadas como dados em uma cache de dados, por exemplo, uma cache de dados nível 1, em seu caminho para a memória do sistema. A  
5 geração de instruções a partir de códigos de byte Java é outra situação na qual as instruções são inicialmente tratadas como dados que são armazenados utilizando o caminho de dados, incluindo a cache de dados, para a memória de sistema. O estado inicial de um programa no qual  
10 as instruções de programa estão sendo tratadas como dados cria um problema de coerência dentro da hierarquia de memória, uma vez que pelo menos certas partes de um programa podem residir na cache de dados antes da execução do programa.

15 Para resolver o problema de coerência, uma abordagem de software é tipicamente realizada em que o programa ou os segmentos de programa na cache de dados são movidos para a memória de sistema sob controle do programa, a cache de instrução é tipicamente invalidada para limpar a  
20 cache de quaisquer segmentos do programa antigo, e as instruções compreendendo o programa são então buscadas a partir da memória de sistema. O movimento das instruções da cache de dados até a memória de sistema, e a ação de buscar as instruções, a partir da memória de sistema antes da  
25 execução, pode consumir vários ciclos, reduzindo o desempenho do processador devido ao overhead de tempo de processamento que deve ocorrer para acessar as instruções inicialmente residindo na cache de dados antes do programa executando no processador.

30

### SUMÁRIO

Entre seus vários aspectos, a presente revelação reconhece que o overhead relacionado às instruções em uma cache de dados pode limitar o desempenho do processador e,

possivelmente, limitar a qualidade de serviço que pode ser obtida. A presente invenção também reconhece que pode ser desejável acessar instruções que estejam em uma cache de dados.

5            Além disso, a presente revelação descreve equipamento, métodos, e meio legível por computador para ir buscar diretamente uma instrução a partir de uma cache de dados quando aquela instrução não foi encontrada na cache de instrução, uma falha da cache de instrução, e se  
10            determina que a instrução esteja na cache de dados. Mediante ação de buscar a instrução diretamente a partir da cache de dados, após uma falha da cache de instrução, o desempenho do processador pode ser aperfeiçoado.

            Com essa finalidade, uma modalidade da presente  
15            invenção inclui um método de encontrar uma instrução em uma cache de dados que é separado de uma cache de instrução. Em tal método, é determinado que uma tentativa de busca falhou na cache de instrução para a instrução em um endereço de busca de instrução. O endereço de busca de instrução é  
20            transformado em um endereço de busca de dados. Além disso, uma tentativa de busca na cache de dados é feita a partir da instrução no endereço de busca de dados transformado.

            Outra modalidade da invenção trata de um complexo de processador para buscar instruções. O complexo de  
25            processador pode incluir adequadamente uma cache de instrução, uma cache de dados, e um primeiro seletor. O primeiro seletor é usado para selecionar um endereço de busca de instrução ou um endereço de busca de dados. Um endereço de busca selecionado é aplicado a uma cache de  
30            dados pelo que instruções ou dados podem ser seletivamente buscados a partir da cache de dados.

            Um entendimento mais completo dos conceitos inventivos da presente invenção, aqui revelados, assim como

outras características, serão evidentes a partir da Descrição Detalhada a seguir e dos desenhos anexos.

#### **DESCRIÇÃO RESUMIDA DOS DESENHOS**

5 A Figura 1 é um diagrama de blocos de um sistema de comunicação sem fio exemplar no qual uma modalidade da revelação pode ser empregada;

A Figura 2 é um diagrama funcional de blocos de um complexo de processador e memória no qual a operação de cache de dados é adaptada para operações eficientes de  
10 memória de busca de instrução de acordo com uma modalidade da presente invenção;

A Figura 3 é um fluxograma de um método exemplar para buscar uma instrução armazenada em uma cache de dados, para reduzir o overhead de execução de falha associado com  
15 a instrução inicialmente armazenada como dados na cache de dados de acordo com a presente revelação;

A Figura 4 é um diagrama funcional de blocos de um complexo de processador e memória que inclui uma tabela de páginas de instrução na qual a operação de cache de  
20 dados é adaptada para busca de instrução eficiente de acordo com a presente revelação;

A Figura 5 é um fluxograma de um método exemplar para buscar uma instrução armazenada em uma cache de dados de acordo com a presente revelação; e

25 A Figura 6 é um fluxograma de um método exemplar para executar código que é gerado como dados e armazenado em uma cache de dados, de acordo com a presente revelação.

#### **DESCRIÇÃO DETALHADA**

30 Aspectos inventivos da presente revelação serão ilustrados mais completamente com referência aos desenhos anexos, nos quais várias modalidades da revelação são mostradas. A modalidade dessa invenção pode, contudo, ser incorporada em várias formas e não deve ser considerada

como limitada às modalidades aqui apresentadas. Mais propriamente, essas modalidades são providas de modo que essa revelação será completa e meticulosa, e transmitirá integralmente o escopo da revelação para aqueles versados na técnica.

Será considerado que a presente invenção pode ser incorporada como métodos, sistemas, ou produtos de programa de computador. Conseqüentemente, os conceitos inventivos da presente invenção, aqui revelados, podem assumir a forma de uma modalidade de hardware, uma modalidade de software ou uma modalidade combinando aspectos de software e de hardware. Adicionalmente, os conceitos inventivos da presente invenção, aqui revelados, podem assumir a forma de um produto de programa de computador em um meio de armazenamento utilizável por computador tendo código de programa utilizável por computador incorporado no meio. Qualquer meio legível por computador adequado pode ser utilizado incluindo discos rígidos, CD-ROMs, dispositivos de armazenamento ótico, memórias flash, ou dispositivos de armazenamento magnético.

Código de programa de computador que pode ser compilado, montado e carregado em um processador pode ser inicialmente escrito em uma linguagem de programação tal como C, C++, native Assembler, JAVA®, Smalltalk, JavaScript®, Visual Basic®, TSQL, Perl, ou em várias outras linguagens de programação de acordo com os ensinamentos da presente revelação. Código de programa ou meio legível por computador se refere ao código de linguagem de máquina tal como código e objeto cujo formato pode ser entendido por um processador. Modalidades de software da revelação não dependem de sua implementação com uma linguagem de programação específica. Quando o código de programa é executado, é criada uma nova tarefa que define o ambiente

de operação para o código de programa.

A Figura 1 mostra um sistema de comunicação sem fio exemplar 100 no qual uma modalidade da revelação pode ser empregada. Para fins de ilustração, a Figura 1 mostra  
5 três unidades remotas 120, 130, e 150 e duas estações base 140. Será reconhecido que sistemas típicos de comunicação sem fio podem ter unidades remotas e estações base. As unidades remotas 120, 130 e 150 incluem componentes de hardware, componentes de software, ou ambos conforme  
10 representado pelos componentes 125A, 125C, e 125B, respectivamente, os quais foram adaptados para incorporar a revelação conforme discutido adicionalmente abaixo. A Figura 1 mostra sinais de link direto 180 a partir das estações base 140 para as unidades remotas 120, 130 e 150 e  
15 sinais de link reverso 190 a partir das unidades remotas 120, 130 e 150 para as estações base 140.

Na Figura 1, a unidade remota 120 é apresentada como um telefone móvel; a unidade remota 130 é mostrada como um computador portátil; e a unidade remota 150 se  
20 apresenta como uma unidade remota de posição fixa em um sistema de loop local sem fio. Por exemplo, as unidades remotas podem ser telefones celulares, unidades de sistema de comunicação pessoal de mão (PCS), unidades portáteis de dados tais como assistentes pessoais de dados, ou unidades  
25 de dados de posição fixa tal como equipamento de leitura de medidor. Embora a Figura 1 ilustre unidades remotas de acordo com os ensinamentos da revelação, a revelação não é limitada a essas unidades exemplares ilustradas. A revelação pode ser empregada adequadamente em qualquer  
30 dispositivo tendo um processador com uma cache de instrução, uma cache de dados, e uma memória de sistema.

A Figura 2 é um diagrama funcional de blocos de um complexo de processador e memória 200 no qual a operação

normal da cache de dados é adaptada para busca de instrução mais eficiente conforme descrito adicionalmente aqui. O complexo de processador e memória 200 inclui um processador 202, uma cache de instrução de nível 1 (L1) 204, uma unidade de controle de cache de instrução L1 206, uma cache de dados L1 208, uma unidade de controle de cache de dados L1 210, uma seção de controle 211, e uma memória de sistema 212. A unidade de controle de cache de instrução L1 206 pode incluir uma memória endereçável de conteúdo de instrução para casamento de identificador de instrução, conforme pode ser usado em uma cache associativa estabelecida. A seção de controle 211 inclui elementos de multiplexação 220, 226, e 234, dispositivos de portas 232 e 238, e um inversor 240. Dispositivos periféricos, os quais podem se conectar ao complexo de processador, não são mostrados para clareza de discussão da presente revelação. O complexo de processador e memória 200 pode ser adequadamente empregado nos componentes 125A-C para executar códigos de programa que são armazenados na memória de sistema 212.

Para buscar uma instrução no complexo de processador e memória 200, o processador 202 gera um endereço de busca de instrução (IA) 214 da instrução desejada e envia o endereço de busca de instrução para a unidade de controle de cache de instrução L1 206. A unidade de controle de cache de instrução L1 206 verifica se a instrução está presente na cache de instrução L1 204. Essa verificação é realizada, por exemplo, através do uso de uma memória endereçável de conteúdo interno (CAM) em uma busca associativa para um casamento com o endereço de busca de instrução fornecido. Quando a instrução está presente, um casamento ocorre e a unidade de controle de cache de instrução L1 206 indica que a instrução está presente na

cache de instrução 204. Se a instrução não estiver presente, nenhum casamento será encontrado na busca associativa CAM e a unidade de controle de cache de instrução L1 206 indica que a instrução não está presente na cache de instrução 204.

Se a instrução estiver presente, a instrução no endereço de busca de instrução é selecionada a partir da cache de instrução 204. A instrução então é enviada no barramento de saída de instrução 216 através do elemento de multiplexação 226 para o processador 202.

Se a instrução não estiver presente na cache de instrução, um sinal de falha de cache de instrução (I $\bar{S}$ M=1) 218 é ativado indicando que ocorreu uma falha. A partir da detecção de uma falha na cache de instrução, o complexo de processador e memória 200 tenta buscar a instrução desejada a partir da cache de dados L1 208. Com essa finalidade, o elemento de multiplexação 220 é habilitado pelo sinal de falha (I $\bar{S}$ M=1) 218 para selecionar o endereço de busca de instrução 214. O endereço de busca de instrução 214 passa então através de um elemento de multiplexação 220 para um barramento Daddress 222 e é enviado para a unidade de controle de cache de dados L1 210 como um endereço de busca de dados. Observa-se que o complexo de processador e memória 200 representa uma visão lógica do sistema, uma vez que, por exemplo, a aplicação do endereço de busca de instrução 214 para o barramento Daddress 222 pode exigir uma arbitragem ou um período de espera antes que o acesso ao barramento Daddress 222 possa ser obtido. A abordagem realizada para multiplexar o endereço de busca de instrução 214 com o endereço de dados gerado pelo processador 223 pode ser variada e depende da abordagem específica realizada nos projetos de cache de instrução e de cache de dados.

A unidade de controle de cache de dados L1 210 verifica se existe um acerto na cache de dados L1 208 no endereço de busca de instrução fornecida, através de uma busca associativa interna, por exemplo, no endereço de  
5 busca de instrução, fornecido. Um acerto indica que existem dados presentes no endereço de busca de instrução, fornecido. Esses dados são na realidade uma instrução e a entrada de cache de dados é buscada a partir da cache de dados L1 208 e colocada no barramento de saída de dados  
10 224. Para fornecer os dados buscados a partir da cache de dados L1 208 como uma instrução para o processador, um elemento de multiplexação 226 pode ser adequadamente empregado. O barramento de saída de dados 224 é selecionado pelo elemento de multiplexação 226 colocando os dados  
15 buscados a partir da cache de dados no barramento de instrução 228 do processador, onde existe uma falha na cache de instrução seguido por um acerto na cache de dados no endereço de busca de instrução. A ocorrência da falha na cache de instrução, indicada pelo sinal de falha (I\$M=1)  
20 218 sendo ativa verdadeira seguida pelo acerto na cache de dados no mesmo endereço de busca de instrução, indicado por um sinal de acerto (D\$M=1) 230 sendo ativo verdadeiro, é logicamente representada pela porta E 232. A saída da porta E 232 é o sinal de seleção 233 para o elemento de  
25 multiplexação 226. A instrução encontrada na cache de dados também é multiplexada para carregamento na cache de instrução 204 pelo elemento de multiplexação 234 utilizando o sinal de seleção 233 logicamente provido pela porta E 232. Enquanto o barramento de saída de dados 224 está  
30 enviando a instrução para o processador, a entrada de dados de leitura 236 do processador é desativada pela porta E 238 utilizando o inversor 240 para prover um inverso do sinal de seleção 233.

Se foi determinado que houve uma falha na cache de dados no endereço de busca de instrução fornecido, a instrução não está na cache de dados e a instrução é buscada a partir da memória de sistema 212. O sinal de acerto ((D\$M=1) 230 também é enviado para a unidade de controle de cache de instrução L1 206 para indicar mediante seu estado inativo que uma falha ocorreu na tentativa de localizar a instrução na cache de dados 208. Observar que outro meio de sinalização pode ser usado para indicar que uma falha ocorreu na tentativa de localizar a instrução na cache de dados 208. Como a instrução não está na cache de instrução 204 e não está na cache de dados 208 ela deve ser buscada a partir da memória de sistema 212. Quando a instrução é obtida a partir da memória de sistema 212 ela é enviada para o processador 202. Observar que os caminhos a partir da memória de sistema para fornecimento de uma instrução devido a uma falha na cache de instrução ou na cache de dados e para fornecer dados devido a uma falha na cache de dados não são mostrados para ilustrar claramente a presente revelação.

A Figura 3 é um fluxograma exemplar de um método 300 para diretamente buscar uma instrução em uma cache de dados após ter uma falha na cache de instrução, para minimizar o overhead comumente associado com o manejo da instrução inicialmente armazenada como dados na cache de dados. Relações exemplares entre as etapas da Figura 3 e os elementos da Figura 2 são indicadas mediante descrição de como os elementos a partir do complexo de processador e memória 200 podem cooperar adequadamente para realizar as etapas do método 300.

Para buscar uma instrução, um endereço de busca de instrução é gerado na etapa 304. Por exemplo, um processador, tal como o processador 202, gera um endereço

de busca de instrução da instrução desejada e envia o endereço de busca de instrução 214 para o controlador de cache de instrução L1 206. Na etapa 308, é determinado se existe um acerto ou uma falha de cache de instrução. Por exemplo, o controlador de cache de instrução L1 206 verifica se a instrução está presente na cache de instrução 204. Se a instrução estiver presente, sua presença é indicada como um acerto. Se a instrução estiver presente, o método 300 prossegue para a etapa 312 e a instrução no endereço de busca de instrução é selecionada. Na etapa 316, a instrução é enviada para o processador. Por exemplo, a instrução selecionada é colocada no barramento de saída de instrução 216 e enviada para o processador 202 através do elemento de multiplexação 226.

Se a instrução não estiver presente na cache de instrução conforme determinado na etapa 308, é fornecida uma indicação de que uma falha ocorreu e é feita uma tentativa de buscar a instrução a partir da cache de dados na etapa 320. Por exemplo, o endereço de busca de instrução 214 é enviado através do elemento de multiplexação 220 como um endereço de busca de dados 222 para a cache de dados 208. Na etapa 324, é feita uma verificação, por exemplo, pelo controlador de cache de dados L1 210 para verificar se existem dados válidos presentes no endereço de busca de instrução fornecido. Se existem dados válidos presentes na cache de dados no endereço de busca de instrução fornecido, os dados efetivamente são uma instrução e a entrada de cache de dados é buscada na etapa 328. Na etapa 316, os dados buscados a partir da cache de dados são enviados como uma instrução para o processador. Por exemplo, os dados buscados no barramento de saída de dados 224 a partir da cache de dados 208 são enviados através do elemento de multiplexação 226 e fornecidos como uma instrução para o

processador 202 no barramento de instrução 228.

Na etapa 324, se houve uma falha na cache de dados no endereço de busca de instrução fornecido, a instrução não está na cache de dados e na etapa 332 a instrução é buscada a partir da memória de sistema. Por exemplo, o sinal de acerto de cache de dados D\$M=1 230 é enviado para a unidade de controle de cache de instrução L1 206 para indicar por intermédio de seu estado inativo que uma falha ocorreu na tentativa de localizar a instrução na cache de dados 208. Como a instrução não está na cache de instrução 204 e não está na cache de dados 208 ela deve ser buscada a partir da memória de sistema 212. Quando a instrução é obtida a partir da memória de sistema 212, a instrução é enviada para o processador 202, conforme indicado na etapa 316.

A Figura 4 é um diagrama funcional de blocos de um complexo de processador e memória 400 que inclui uma tabela de páginas de instrução na qual a operação de cache de dados normal é adaptada para operação eficiente de busca de instrução de acordo com a presente revelação. O complexo de processador e memória 400 inclui um processador 402, uma cache de instrução de nível 1 (L1) 404, uma unidade de gerenciamento de memória de instrução (IMMU) e controle de cache (IMMU/\$Control) 406, uma cache de dados L1 408, uma unidade de gerenciamento de memória de dados (DMMU) e controle de cache (DMMU/\$Control) 410, uma seção de controle 411, e uma hierarquia de memória 412. O IMMU/\$Control 406 pode incluir, por exemplo, um processo de conversão de endereço de instrução de virtual para física. A seção de controle 411 inclui elementos de multiplexação 432, 438, e 448, dispositivos de portas 428, 444, e 452, e um inversor 454. Dispositivos periféricos, os quais podem conectar o complexo de processador, não são mostrados para

clareza de discussão na presente revelação. O complexo de processador e memória 400 pode ser empregado adequadamente em componentes 125A-C para executar código de programa que é armazenado na memória de sistema 412.

5           A cache de instrução pode utilizar um armazenador de tradução look-aside (TLB) que contém uma tabela de página de instrução para melhorar o desempenho da cache de instrução. A tabela de páginas de instrução tendo, por exemplo, uma lista de números de páginas físicas associadas  
10 com números de páginas virtuais e informação adicional associada com cada entrada de número de página. Uma entrada de tabela de páginas de instrução é criada quando uma página da memória na faixa de endereços de instrução é carregada na cache de instrução ou na cache de dados. O  
15 carregamento de uma página de memória pode ocorrer sob a supervisão de um sistema operacional (OS). Em operação, a tabela de página de instrução é examinada no sentido de um casamento com um número de página virtual fornecido ao TLB. Embora um TLB tendo uma tabela de páginas de instrução seja  
20 descrito aqui como parte da MMU de instrução e controle de cache 406, será reconhecido que abordagens alternativas podem ser usadas.

Para buscar uma instrução no complexo de processador e memória 400, o processador 402 gera um  
25 endereço de busca de instrução (IA) 414 para a instrução desejada e envia o endereço de busca de instrução para o IMMU/\$Control 406. Uma entrada apropriada em uma tabela de página de instrução, tal com tabela de página 416 localizada no IMMU/\$Control 406 é selecionada com base em  
30 um número de página fornecido que é parte do IA 414. O endereço de instrução baseado na entrada de tabela de página selecionada é combinado com um endereço de página, também parte do IA 414, gerando um endereço de instrução

(GA) 418 que é aplicado internamente à cache de instrução L1 404. A entrada selecionada a partir da tabela de páginas 416 inclui informação adicional armazenada com aquela entrada. Um dos bits adicionais de informação que pode ser armazenado com cada entrada de tabela de página é um atributo de cache de dados de verificação, rotulado como bit D 420.

O bit D é ajustado para um "1" quando a entrada na tabela de página de instrução é criada devido ao carregamento de uma página de instrução na cache de dados ou ao gerar instruções que são armazenadas em uma página na cache de dados durante processamento. O bit D é tipicamente estabelecido pelo sistema operacional OS para indicar que um conteúdo da página pode ser usado como ambos, dados e instruções. Em um cenário exemplar, um programa, gerando dados que serão usados como instruções, chama o OS para solicitar que as páginas apropriadas sejam marcadas mediante ajuste do bit D nas entradas de tabela de página, associadas. Em outro cenário, um programa também pode solicitar páginas a partir do OS que já estão estabelecidas com o bit D ajustado. O bit D não precisa necessariamente ser explicitamente cancelado. Se um programa especifica que a cache de dados pode conter instruções mediante ação de fazer com que o bit D ou bits D apropriados sejam ajustados, então aquela especificação pode ser válida durante a vida do programa. O bit D ou bits D podem ser posteriormente cancelados quando a tabela de páginas for usada para um processo diferente.

O IMMU/\$Control 406 verifica se a instrução está presente na cache de instrução 404. Se a instrução estiver presente, essa presença é indicada como um acerto. Se a instrução estiver presente, a instrução no endereço de busca de instrução é selecionada a partir da cache de

instrução 404. A instrução então é enviada no barramento de saída de instrução 422 através de elemento de multiplexação 438 para o processador 402. Se a instrução não estiver presente, é fornecida uma indicação pelo IMMU/\$Control 406 de que ocorreu uma falha e um sinal de falha de cache de 5 instrução (I\$M=1) 424 é ativado indicando que ocorreu uma falha.

Ao detectar uma falha na cache de instrução em conjunto com o bit D selecionado sendo ajustado para um 10 "1", o complexo de processador e memória 400 tenta buscar a instrução desejada a partir da cache de dados L1 408. Essa tentativa pode ser realizada adequadamente, por exemplo, mediante uso do bit D selecionado em uma função de porta. O bit D 420 a partir da entrada de tabela de página selecionada é enviado como sinal de bit D 426y. Os sinais de bit D 426 são, por exemplo, ANDed, pela porta E 428, com 15 indicação de falha (I\$M=1) 424. A saída 430 da porta E 428 é, então, usada pelo elemento de multiplexação 432 para selecionar o endereço de instrução gerado (GA) 418 ou um 20 endereço de dados 433 a partir do processador 402. Quando selecionado, o GA 418 passa através do elemento de multiplexação 432 para o barramento Daddress (DA) 434 e é enviado para o MMU de dados e controle de cache 410 para determinar se a instrução reside na cache de dados 408 no 25 endereço de busca de dados. Observa-se que o complexo de processador e memória 400 representa uma visão lógica do sistema, uma vez que, por exemplo, a aplicação do endereço de instrução geral 418 no barramento Daddress 434 pode exigir uma arbitragem ou um período de espera antes de o 30 acesso ao barramento Daddress 434 poder ser obtido. A abordagem realizada para multiplexar o endereço de instrução gerado 418 com o endereço de dados gerado pelo processador 433 pode ser variada e depende da abordagem

específica realizada nos projetos de cache de instrução e de cache de dados.

A cache de dados então verifica se existem dados válidos presentes no endereço de busca de instrução  
5 fornecido. Se houver dados válidos presentes no endereço de busca de instrução fornecido, os dados na realidade são uma instrução e a entrada de cache de dados é buscada a partir da cache de dados L1 408 e colocada no barramento de saída de dados 436. Para fornecer a entrada de cache de dados  
10 como uma instrução para o processador, é usado um elemento de multiplexação 438, por exemplo. O elemento de multiplexação 438 é habilitado a passar o barramento de saída de dados 436 para o barramento de instrução 440 do processador quando houver uma falha na cache de instrução e  
15 o bit D selecionado é ajustado para um "1" seguido por um acerto na cache de dados no endereço de busca de instrução. A ocorrência da falha na cache de instrução, indicada pelo sinal de falha (I\$M=1) 424 sendo altamente ativo, e o sinal de bit D 426 ajustado para um "1" seguido pelo acerto na  
20 cache de dados no endereço de instrução gerado, indicado pelo sinal de acerto (D\$H=1) 442 sendo altamente ativo, é representado logicamente pela porta E 444. A saída da porta E 444 é o sinal de seleção 446 para o elemento de multiplexação 438. A instrução no barramento de saída de  
25 dados também é multiplexada para carregamento na cache de instrução pelo elemento de multiplexação 448 utilizando o sinal de seleção 446. Enquanto o barramento de saída de dados de cache de dados L1 436 está enviando a instrução para o processador 402, o barramento de saída de dados 436  
30 é fechado para transferências para a entrada de dados de leitura 450 do processador pela porta E 452 utilizando um inverso do sinal de seleção 446 provido pelo inversor 454.

Se foi determinado que havia uma falha na cache

de dados no endereço de busca de instrução fornecido, a instrução não está na cache de dados e a instrução é buscada a partir da memória de sistema 412. O sinal de acerto (D\$H=1) 442 também é enviado para o IMMU/\$Control 5 406 para indicar por intermédio de seu estado inativo que uma falha ocorreu na tentativa de localizar a instrução na cache de dados 408. Quando a instrução é obtida a partir da memória de sistema 412, ela é enviada para o processador 402. Observar que os caminhos a partir da hierarquia de 10 memória para fornecer uma instrução devido a uma falha na cache de instrução ou cache de dados e para fornecer dados devido a uma falha na cache de dados não são mostrados, porém qualquer uma de uma ampla variedade de abordagens de conexão pode ser empregada consistente com a aplicação e o 15 processador empregado.

A Figura 5 é um fluxograma exemplar de um método 500 para buscar uma instrução em uma cache de dados após ter uma falha na cache de instrução e um atributo de cache de dados de verificação indica que a cache de dados deve 20 ser verificado para a instrução. Relações exemplares entre as etapas da Figura 5 e os elementos da Figura 4 são indicadas mediante referência aos elementos exemplares a partir do complexo de processador e memória 400 que pode ser adequadamente empregado para realizar as etapas do 25 método 500 da Figura 5.

Para buscar uma instrução, um endereço de busca de instrução para a instrução desejada é gerado na etapa 502. Por exemplo, um processador, tal como o processador 402 gera um endereço de busca de instrução e envia o 30 endereço de busca de instrução 414 para o controlador de cache de instrução L1 406. O endereço de busca de instrução pode ser um endereço virtual composto de um número de página 504 e um endereço de página 506. Na etapa 508, uma

entrada apropriada em uma tabela de página de instrução, tal como tabela de página de instrução 416, é selecionada com base no número de página fornecido 504. O endereço gerado com base na entrada de tabela de página selecionada  
5 é combinado na etapa 509 com o endereço de página 506 para produzir um endereço de cache de instrução.

A entrada selecionada a partir da tabela de página de instrução 416 inclui informação adicional armazenada com aquela entrada. Um dos bits adicionais de  
10 informação que podem ser armazenados com cada entrada de tabela de página é um atributo de cache de dados de verificação, tal como o bit rotulado como bit D 420. Esse atributo é selecionado na etapa 510.

Na etapa 512, determina-se se existe um acerto de  
15 cache de instrução, ou uma falha. Por exemplo, a cache de instrução verifica se a instrução está presente. Se a instrução estiver presente, sua presença é indicada como um acerto. Se a instrução estiver presente, o método 500 prossegue para a etapa 514 e a instrução no endereço de  
20 busca de instrução é selecionada. Na etapa 516, a instrução é enviada para o processador. Por exemplo, a instrução selecionada é colocada no barramento de saída de instrução 422 e enviada através do elemento de multiplexação 438 para o barramento de instrução 440 do processador 402.

25 Se a instrução não estiver presente na cache de instrução conforme determinado na etapa 512, uma indicação é dada no sentido de que ocorreu uma falha e o método 500 prossegue para a etapa 518. Na etapa 518, o bit D que foi selecionado na etapa 510 é verificado no sentido de se ele  
30 está ajustado para um "1" indicando que a cache de dados deve ser verificado para a instrução. Se o bit D foi ajustado para um "1", processador tenta buscar a instrução a partir da cache de dados na etapa 520. Por exemplo, o

endereço de busca de instrução gerado 418 é enviado como um endereço de busca de dados 434 para a cache de dados.

Na etapa 524, a cache de dados verifica se existem dados válidos presentes no endereço de busca de instrução, fornecido. Se houver dados válidos presentes no endereço de busca de instrução, fornecido, os dados na realidade são uma instrução e a entrada de caixa de dados é buscada na entrada 528. Na etapa 516, os dados buscados a partir da cache de dados são enviados como uma instrução para o processador. Por exemplo, os dados buscados no barramento de saída de dados 436 são enviados através do elemento de multiplexação 438 e fornecidos como uma instrução para o processador 402 no barramento de instrução 440.

Retornando à etapa 518, se for determinado na etapa 518 que o bit D era um "0", sabe-se que a instrução não está presente na cache de dados e o método 500 prossegue para a etapa 522. A etapa 522 também é alcançada para a situação onde havia uma falha na cache de dados no endereço de busca de instrução fornecido, conforme determinado na etapa 524. Em qualquer caso, a instrução não é reconhecida como estando presente na cache de instrução ou na cache de dados e a instrução é buscada a partir da memória de sistema, conforme indicado na etapa 522. Por exemplo, a memória de sistema 412 será acessada para a instrução. Quando a instrução é obtida a partir da memória de sistema 412, a instrução é enviada para o processador 402, conforme indicado na etapa 516.

A Figura 6 é um fluxograma exemplar de um método 600 para executar código de programa que é gerado como dados e armazenado em uma cache de dados. Código de programa após esse método pode ser executado em um complexo de processador e memória tendo uma cache de instrução, uma

cache de dados, e uma memória de sistema, tal como, aqueles discutidos em conexão com as Figuras 2 e 4, e pode ser adequadamente empregado em componentes 125A-C da Figura 1.

Na etapa 602, um programa gera o código. Tal  
5 geração pode ocorrer, por exemplo, quando um programa gera código executável a partir de um programa compactado. O código gerado é inicialmente tratado como dados e armazenado na cache de dados após ele ser gerado. Antes de executar o programa, uma cache de instrução é invalidada na  
10 etapa 64. A etapa de invalidação garante que não existam instruções no mesmo endereço que o código gerado. Na etapa 606, o código gerado é executado pelo processador mediante busca das instruções a partir do espaço de endereço de programa na cache de instrução e pode incluir instruções  
15 que são armazenadas na cache de dados. Para aquelas instruções armazenadas na cache de dados, as técnicas da presente revelação são seguidas permitindo que a cache de dados seja verificado para instruções em uma ocorrência de uma falha na cache de instruções. Ao descobrir uma  
20 instrução na cache de dados, a instrução é diretamente buscada a partir da cache de dados para execução no processador.

Embora a presente revelação tenha sido revelada em um contexto atualmente preferido, será reconhecido que  
25 os presentes ensinamentos podem ser adaptados para uma variedade de contextos consistentes com essa revelação e com as reivindicações a seguir.

## REIVINDICAÇÕES

1. Método de achar uma instrução em uma cache de dados que é separado de uma cache de instruções, o método compreendendo:

5            determinar que uma tentativa de busca falhou na cache de instrução para a instrução em um endereço de busca de instrução;

             transformar o endereço de busca de instrução em um endereço de busca de dados; e

10           realizar uma tentativa de busca na cache de dados para a instrução no endereço de busca de dados transformado.

2. Método, de acordo com a reivindicação 1, compreendendo ainda:

15           ativar um atributo de cache de dados de verificação ao gerar instruções que são armazenadas como dados na cache de dados.

3. Método, de acordo com a reivindicação 2, compreendendo ainda:

20           determinar se um atributo de cache de dados de verificação está ativo; e

             habilitar a tentativa de busca na cache de dados para a instrução apenas se o atributo de cache de dados de verificação estiver ativo.

25           4. Método, de acordo com a reivindicação 2, em que o atributo de cache de dados de verificação é um bit armazenado em uma tabela de páginas de instrução.

30           5. Método, de acordo com a reivindicação 1, em que a etapa de transformar o endereço de busca de instrução compreende ainda:

             multiplexar o endereço de busca de instrução e um endereço de busca de dados; e

             selecionar o endereço de busca de instrução para

aplicação à cache de dados como o endereço de busca de dados transformado, em que o endereço de busca de instrução é selecionado após determinação de que a tentativa de busca de instrução falhou na cache de instrução.

5           6. Método, de acordo com a reivindicação 1, em que a etapa de fazer uma tentativa de busca na cache de dados compreende ainda:

          determinar o acerto de instrução na cache de dados; e

10           buscar a instrução a partir da cache de dados.

          7. Método, de acordo com a reivindicação 1, compreendendo ainda:

          determinar a tentativa de busca na cache de dados que falhou; e

15           informar um controle de memória de instrução de que a tentativa de busca na cache de dados falhou.

          8. Método, de acordo com a reivindicação 7, compreendendo ainda:

20           buscar a instrução a partir de uma memória de sistema.

          9. Complexo de processador compreendendo:

          uma cache de instrução;

          uma cache de dados; e

25           um primeiro seletor para selecionar um endereço de busca de instrução ou um endereço de busca de dados com base em um sinal de seleção, o sinal de seleção fazendo com que o endereço de busca de instrução ou o endereço de busca de dados seja aplicado à cache de dados pelo que instruções ou dados podem ser buscados seletivamente a partir da cache  
30 de dados.

          10. Complexo de processador, de acordo com a reivindicação 9, em que o sinal de seleção do primeiro seletor seleciona o endereço de busca de dados em resposta

a uma operação de acesso de dados.

11. Complexo de processador, de acordo com a reivindicação 9, em que o sinal de seleção do primeiro seletor seleciona o endereço de busca de instrução se um  
5 sinal de falha de instrução indicar que ocorreu uma falha em uma operação de busca de instrução na cache de instrução.

12. Complexo de processador, de acordo com a reivindicação 9, compreendendo ainda:

10 um segundo seletor para selecionar um barramento de saída de instrução a partir da cache de instrução ou um barramento de saída de dados a partir da cache de dados a ser aplicada a uma entrada de barramento de instrução do processador.

15 13. Complexo de processador, de acordo com a reivindicação 12, em que o segundo seletor seleciona o barramento de saída de dados a partir da cache de dados se uma falha ocorreu na cache de instrução e um acerto ocorreu na cache de dados no endereço de busca de instrução  
20 selecionada através do primeiro seletor.

14. Complexo de processador, de acordo com a reivindicação 12, e que o segundo seletor seleciona o barramento de saída de instrução se um acerto ocorrer na cache de instrução.

25 15. Complexo de processador, de acordo com a reivindicação 9, compreendendo ainda:

um terceiro seletor para selecionar um barramento de saída de memória a partir de uma memória de sistema ou um barramento de saída de dados a partir da cache de dados  
30 a ser aplicado à entrada de barramento de instrução da cache de instrução.

16. Complexo de processador, de acordo com a reivindicação 15, em que o terceiro seletor seleciona o

barramento de saída de dados a partir da cache de dados se uma falha ocorreu na cache de instrução e um acerto ocorreu na cache de dados no endereço de busca de instrução selecionada através do primeiro seletor.

5           17. Método para executar código de programa compreendendo:

          gerar instruções que são armazenadas como dados em uma cache de dados;

          invalidar uma cache de instrução antes da  
10 execução do código de programa que utiliza as instruções geradas; e

          buscar uma instrução diretamente a partir da cache de dados se a instrução não for encontrada na cache de instrução, pelo que o código de programa é executado.

15           18. Método, de acordo com a reivindicação 17, em que a geração da etapa de instrução inclui a operação de carregar instruções na cache de dados.

          19. Método, de acordo com a reivindicação 17, em que a invalidação da cache de instrução compreende ainda:

20           invalidar apenas uma porção da cache de instrução onde as instruções geradas são armazenadas.

          20. Método, de acordo com a reivindicação 17, compreendendo ainda:

25           estabelecer um atributo de cache de dados de verificação em uma tabela de página de instrução para indicar que uma instrução pode estar na cache de dados; e

          buscar a instrução a partir da cache de dados se a instrução não for encontrada na cache de instrução e o atributo de cache de dados de verificação estiver ativado.

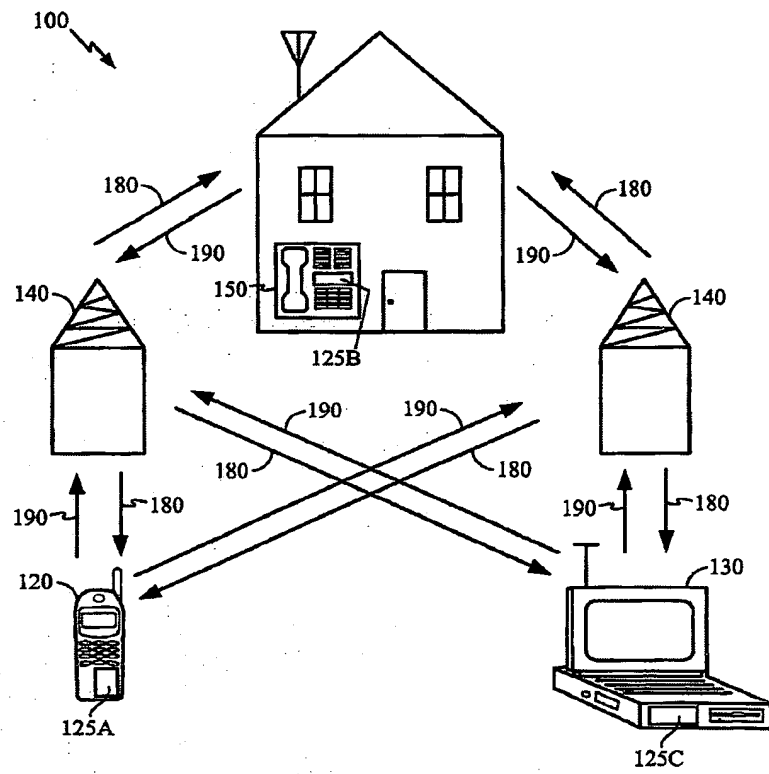


FIG. 1

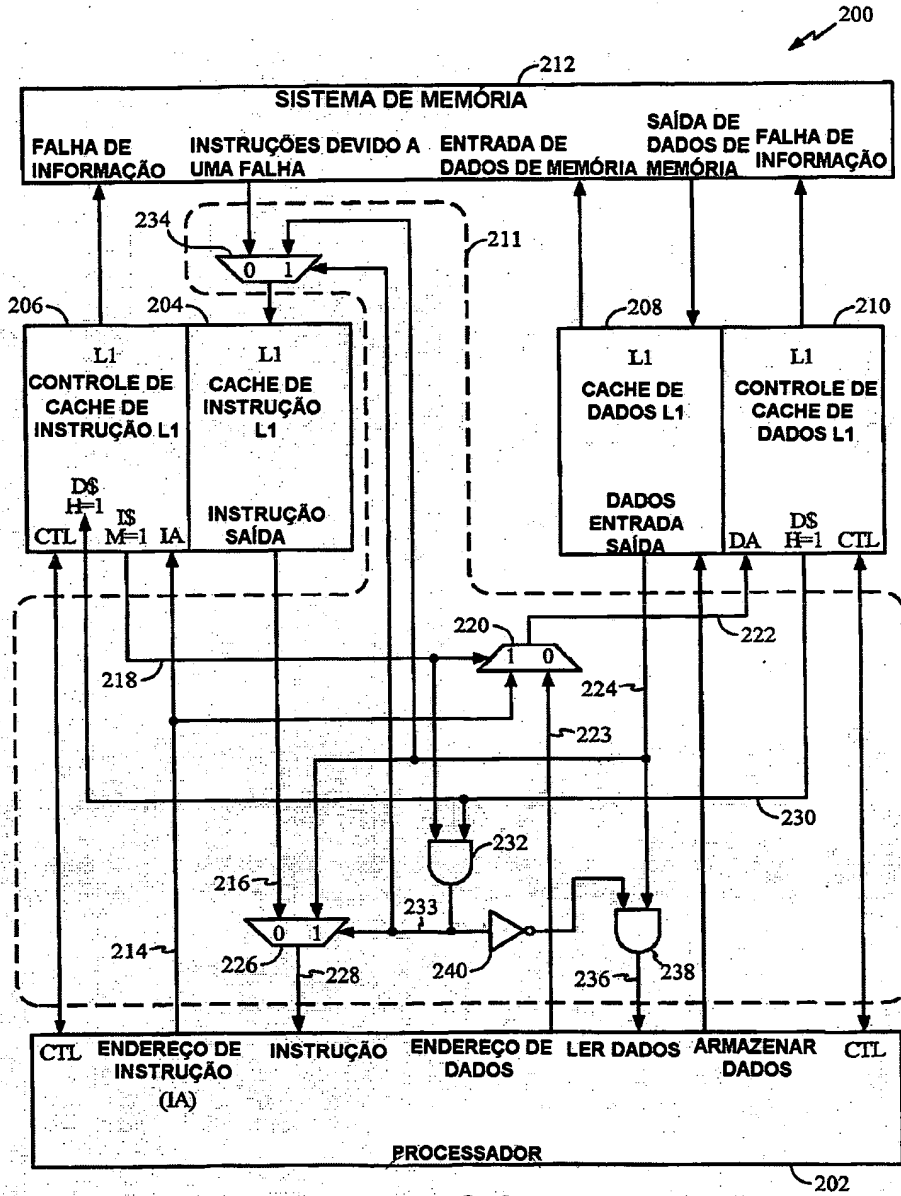


FIG. 2

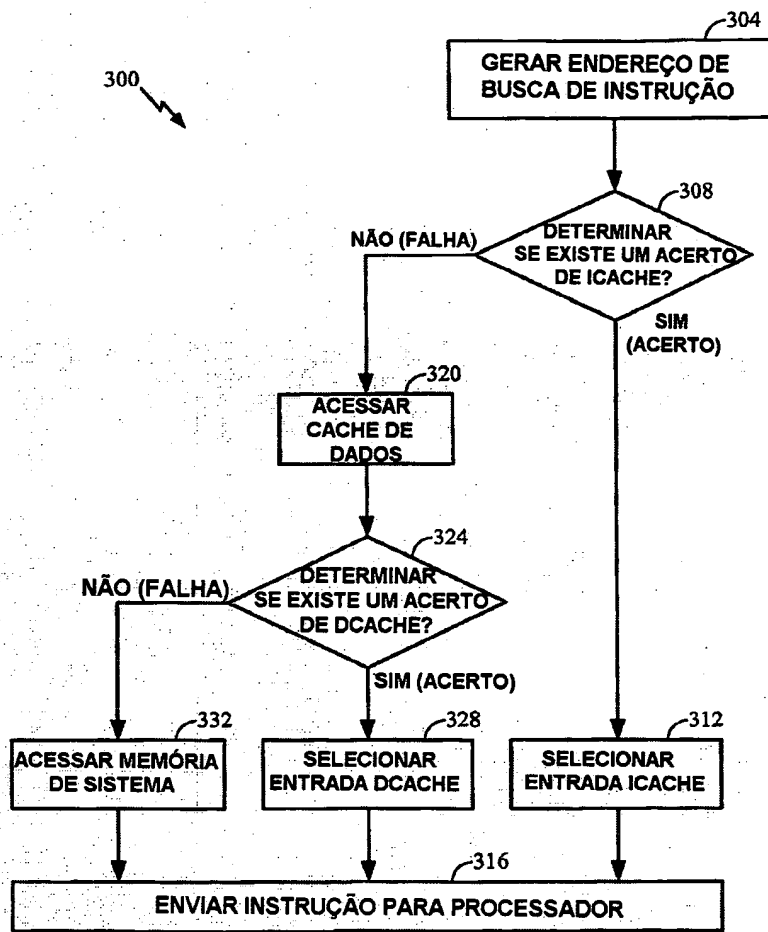


FIG. 3

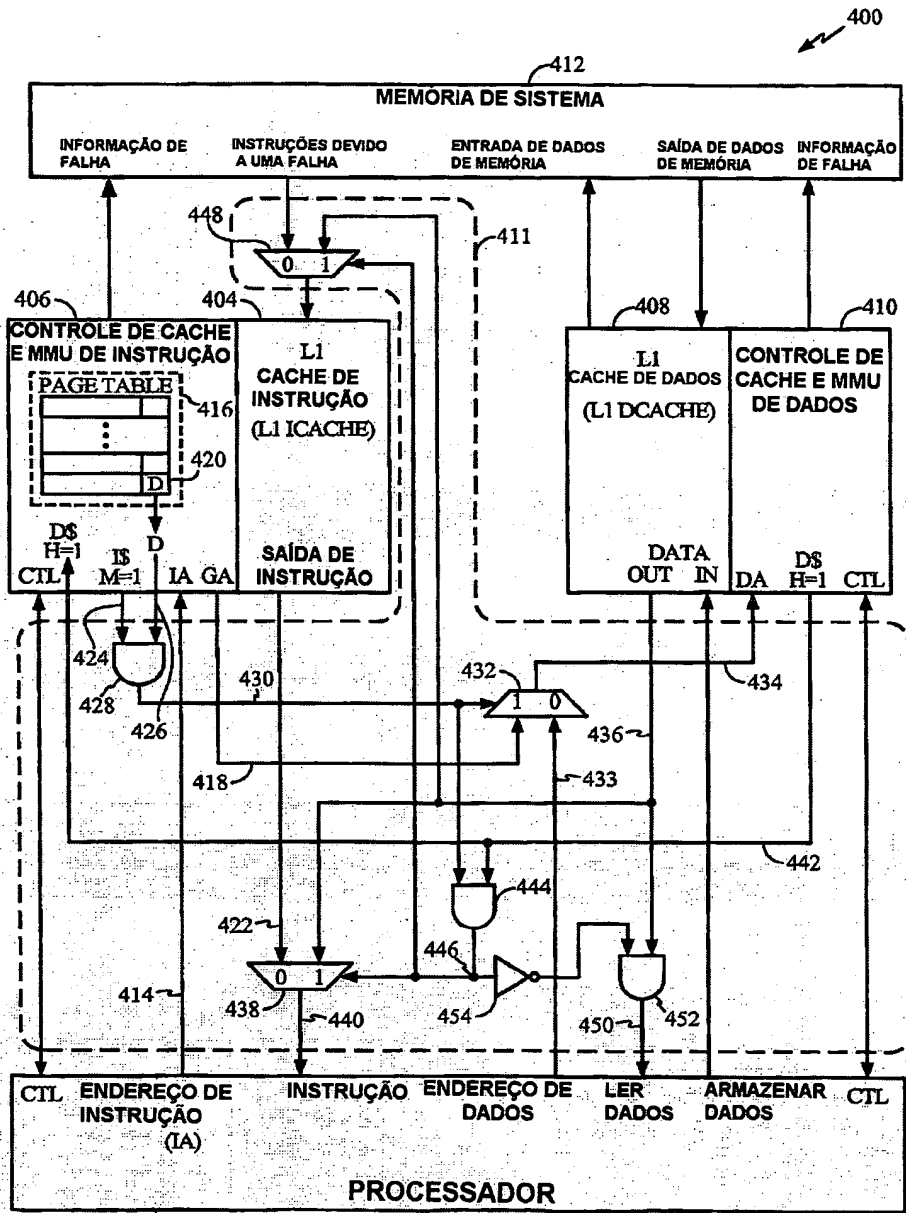


FIG. 4

402

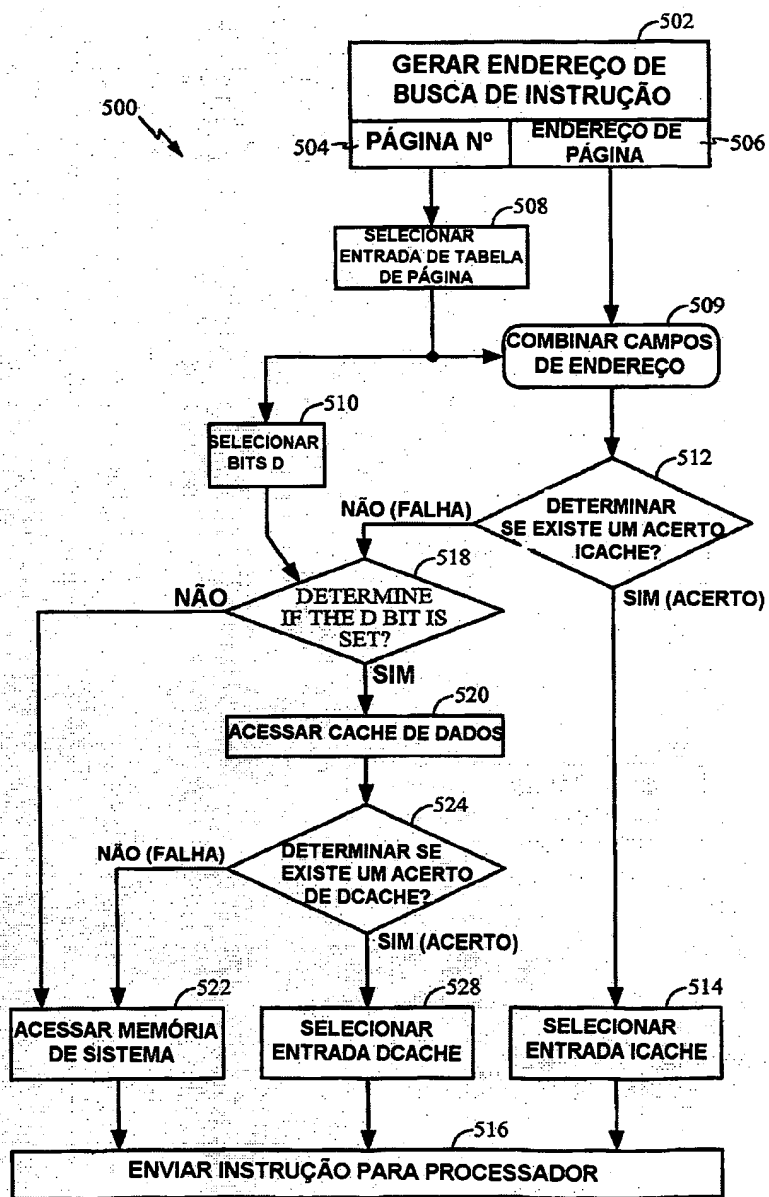


FIG. 5

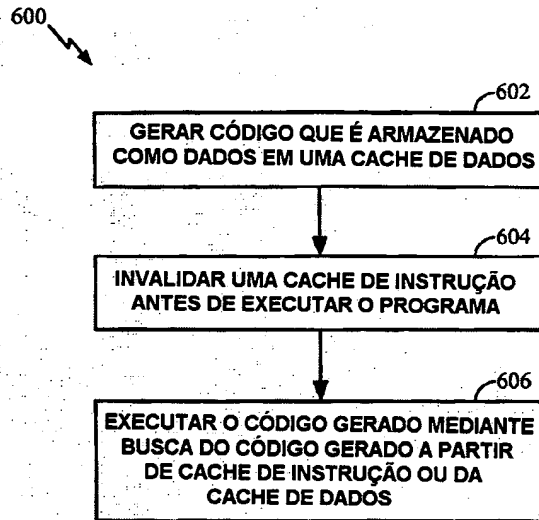


FIG. 6

RESUMO**"GERENCIAMENTO EFICIENTE DE HIERARQUIA DE MEMÓRIA"**

Em um processador, existem situações onde instruções e certas partes de um programa podem residir em uma cache de dados antes da execução do programa. Técnicas de hardware e software são providas para buscar uma instrução na cache de dados após ter uma falha em uma cache de instrução para melhorar o desempenho do processador. Se uma instrução não estiver presente na cache de instrução, um endereço de busca de instrução é enviado como um endereço de busca de dados para a cache de dados. Se houver dados válidos presentes na cache de dados no endereço de busca de instrução fornecido, os dados na realidade são uma instrução e a entrada de cache de dados é buscada e fornecida como uma instrução para o complexo de processador. Um bit adicional pode ser incluído em uma tabela de página de instrução para indicar uma falha na cache de instrução e que a cache de dados deve ser verificada para a instrução.