



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

(21) **PI 1005259-3 A2**

(22) Data de Depósito: 22/12/2010
(43) Data da Publicação: 02/04/2013
(RPI 2204)



(51) *Int.Cl.:*
G06F 9/30
G06F 9/312

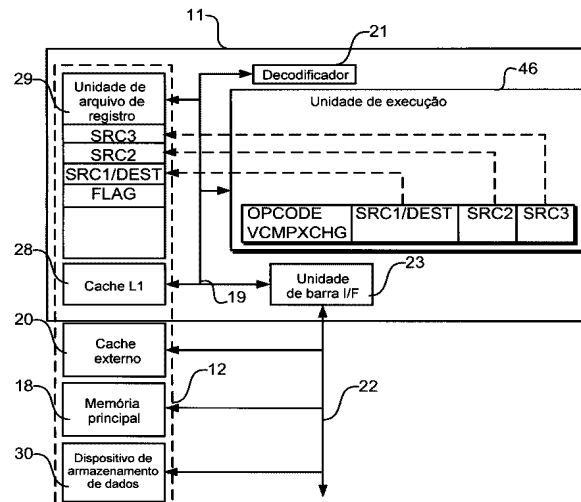
(54) **Título:** VETORES SIMD SINCRONIZADORES

(30) **Prioridade Unionista:** 22/12/2009 US 12/644.529

(73) **Titular(es):** Intel Corporation

(72) **Inventor(es):** Andrew T. Forsyth, Ravi Rajwar

(57) **Resumo:** VETORES SIMD SINCRONIZADORES. A presente invenção refere-se a uma operação de comparar trocar de vetor por meio da: decodificação por um decodificador em um dispositivo de processamento de uma única instrução que especifica uma operação de comparar-trocar de vetor para uma pluralidade de elementos de dados entre uma primeira localização de armazenamento, uma segunda localização de armazenamento, e uma terceira localização de armazenamento; emissão da única instrução para execução por uma unidade de execução no dispositivo de processamento; e, responsiva à execução da única instrução, comparando elementos de dados da primeira localização de armazenamento com os elementos de dados correspondentes na segunda localização de armazenamento; e, responsiva à determinação de que existe uma associação, substituindo elementos de dados da primeira localização de armazenamento pelos elementos de dados correspondentes da terceira localização de armazenamento.



Relatório Descritivo da Patente de Invenção para "VETORES SIMD SINCRONIZADORES".

A presente invenção refere-se a microprocessadores e a outros dispositivos de processamento e, mais particularmente, à sincronização de vetores SIMD.

Múltiplos "threads" e/ou unidades de processamento (adiante referidas como agentes), por exemplo, em sistemas que incorporam processadores "multi-thread", múltiplos dispositivos de processamento, e/ou processadores de múltiplos núcleos, podem muitas vezes precisar compartilhar recursos e dados armazenados dentro do sistema. Deve-se ter cuidado para assegurar que um agente acesse os dados mais recentes e atualizados e também para assegurar que um agente não acesse e modifique os dados correntemente associados com outro agente. Além de complicar este compartilhamento de dados e recursos, os dispositivos de processamento mais modernos incluem uma ou mais memórias "cache" dedicadas. Dentro de sistemas de múltiplos processadores e de múltiplos núcleos, os múltiplos caches em chip irão frequentemente e, na prática, geralmente conter múltiplas cópias de um item de dados. Consequentemente, quando um agente acessar uma cópia de um item de dados, será assegurado que um valor de dados atualizado ou válido será lido.

Desse modo, a "coerência de cache" é mantida nestes sistemas. A coerência de cache refere-se à sincronização de dados gravados e lidos na memória cache, de tal modo que qualquer item de dados armazenado em um cache que seja acessado por um thread ou processador seja a cópia mais recente desse item de dados. Além disso, qualquer valor de dados gravado do cache novamente na memória principal devem ser os dados mais correntes.

Um método de manter a coerência de cache e de assegurar que, quando um agente precisar de um item de dados, o valor mais atualizado para que o item de dados seja acessado é para implementar um semáforo (por exemplo, uma bandeira ou um bloqueio). Um bloqueio, por exemplo, compreende um processo que é executado em resposta a uma solicitação

para um item de dados específico da memória por um agente (por exemplo, em uma operação de carga) para assegurar a sincronização entre os processadores e/ou os threads. De modo geral, um bloqueio está associado com um conjunto de instruções, incluindo a instrução de ler/carregar, uma
5 instrução para modificar o item de dados, e uma instrução de gravar/armazenar. O bloqueio, também referido aqui como "sequência de bloqueios" ou "operação de bloqueio", pode, por exemplo, incluir a aquisição de propriedade de uma localização de memória que armazena dados, a execução de uma operação atômica nos dados enquanto impede que outros pro-
10 cessos operem nesses dados, e a liberação de propriedade da localização de memória depois que a operação atômica é executada. Uma operação atômica é aquela que é executada sequencialmente e de maneira ininterrupta e, além disso, que é garantida para ser completada ou não completada absolutamente (isto é, a operação é indivisível).

15 **DESCRIÇÃO DOS DESENHOS**

A figura 1 mostra um diagrama de bloco de um sistema de computação.

A figura 2 é um diagrama esquemático de um dispositivo de processamento, conforme mostrado na figura 1.

20 A figura 3 mostra um esquema de codificação de uma instrução de comparação e troca de vetor de única instrução múltipla de dados (SIMD).

A figura 4 é um diagrama de bloco de um primeiro sistema de computador exemplificativo para implementar o formato de instrução mos-
25 trado na figura 3.

A figura 5 é um diagrama de bloco de um segundo sistema de computador exemplificativo para implementar o formato de instrução mostrado na figura 3.

A figura 6 é um diagrama de bloco de um terceiro sistema de
30 computador exemplificativo para implementar o formato de instrução mostrado na figura 3.

A figura 7 é um diagrama de bloco de um quarto sistema de

computador exemplificativo para implementar o formato de instrução mostrado na figura 3.

Outras características e vantagens se tornarão evidentes a partir da descrição e dos desenhos, e a partir das reivindicações.

5 DESCRIÇÃO DETALHADA

Na seguinte descrição, inúmeros detalhes específicos são explicados, tais como instruções específicas, formatos de instrução, dispositivos, tais como registros e memória, etc., a fim de prover um completo entendimento dos exemplos providos na mesma. No entanto, será apreciado por 10 aquele versado na técnica que a presente invenção pode ser praticada sem estes detalhes específicos.

Uma maneira de determinar se um semáforo está bloqueado (e/ou para assim fazê-lo) é através do uso de uma sequência (ou operação) de ler-modificar-gravar. Contudo, uma preocupação com uma implementação 15 de ler-modificar-gravar é a aquisição e a liberação do próprio mecanismo de semáforo. Isto é, quando um processo tentar ter controle do espaço de memória compartilhado, ele primeiro irá ler o valor de bloqueio, verificar e modificar (caso permitido) o valor e gravar o valor de modificação novamente no bloqueio. Em geral, é desejável executar a operação de ler-modificar-gravar como uma operação atômica (isto é, completada sem interrupção, 20 uma vez que a execução tenha começado) para impedir que outros processos modifiquem o valor de bloqueio. Com o uso de uma operação atômica, um processo pode adquirir (ler) o semáforo, modificar o valor (caso permitido) e liberar o semáforo iniciando uma gravação para completar a operação 25 antes que outro processo tente adquirir o bloqueio.

Com referência agora à figura 1, o sistema de computador 10 é mostrado apresentando uma pluralidade de unidades de processamento 11 (por exemplo, processadores, núcleos, unidades de execução, etc.) acopladas a uma memória 12 (por exemplo, registros, cache, RAM. etc.) por uma 30 barra 13. Uma ou mais das unidades de processamento 11 são associadas com uma ou mais threads. Consequentemente, o sistema de computador 10 inclui qualquer número adequado de unidades de processamento 11, cada

qual apresentando qualquer número adequado de threads. Unidades de processamento 11 podem, cada qual, fazer parte de um dispositivo de circuito integrado separado ou, alternativamente, todas as unidades de processamento 11 (ou uma porção das mesmas) podem ser formadas em uma única matriz. Neste sistema de computador específico, quatro unidades de processamento 11 (indicadas como P1, P2, P3 e P4) são mostradas como parte do sistema 10. Todas as quatro unidades de processamento 11 são acopladas à memória 12 e especificamente a um espaço de memória compartilhado 15 dentro da memória 12.

É apreciado que a memória 12 pode ser configurada em uma variedade de maneiras. Embora ilustrada como uma memória única, a memória 12 pode compreender múltiplas memórias internas e/ou externas. No exemplo específico, todas as quatro unidades de processamento 11 acessam a memória 12, e uma porção da memória 12, indicada como espaço compartilhado 15, é acessada por mais de uma unidade de processamento 11. É possível que possa haver outras áreas compartilhadas dentro da memória 12, nas quais duas ou mais unidades de processamento 11 podem acessar tais áreas compartilhadas. As áreas não compartilhadas da memória 12 são geralmente relegadas ao acesso por uma unidade de processamento 11 apenas.

O sistema de computador 10, ilustrado na figura 1, se destina a ser um sistema de computador exemplificativo e pode incluir muitos componentes adicionais, que foram omitidos para fins de clareza. Por meio de exemplo, o sistema de computador 10 pode incluir um controlador DMA (acesso de memória direto), uma interface de rede (por exemplo, um cartão de rede), um conjunto de circuitos integrados associado com uma ou mais unidades de processamento 11, bem como linhas e barras de sinal adicionais. Também deve ser entendido que o sistema de computador 10 pode incluir todos os componentes mostrados na figura 1.

Na figura 1, os semáforos empregados são bloqueios (ou variáveis de bloqueios) 16 que são atribuídos para controlar os acessos a um ou mais respectivos espaços compartilhados 15 (conforme mostrado pela linha pontilhada 14). O bloqueio 16 é uma localização específica em memória que

é atribuído para conter um valor associado com a obtenção de acesso ao espaço compartilhado 15. Desse modo, a fim de que uma das unidades de processamento 11 acesse o espaço compartilhado 15, ela primeiro acessa o bloqueio correspondente 16 e testa o estado (valor) dos dados armazenados na localização de bloqueio 16. No formato mais simples, dois valores podem ser atribuídos ao bloqueio 16. Um primeiro valor indicando que o espaço compartilhado está disponível para acesso e um segundo valor indicando que o espaço compartilhado está atualmente sendo utilizado e, portanto, não está disponível para acesso. Novamente, na concretização mais simples, estados de bit 1 e 0 podem ser usados para os estados travado e destravado para o bloqueio 16.

É apreciado que os valores de bloqueio e os estados de bloqueio eficazes para o bloqueio 16 são uma escolha de desenho e muitas variações podendo ser idealizadas. Também, a localização do bloqueio 16 não precisa estar dentro da própria memória 12. Adicionalmente, em referência à figura 1, é apreciado que a memória 12 pode ser um de uma variedade de dispositivos de memória. Também é possível que uma ou mais unidades de processamento 11 possam ser substituídas por um dispositivo(s) de acesso de memória, (dispositivos, tais como controladores de acesso de memória direto), que também acessa(m) a memória. Nestes exemplos, estes dispositivos funcionariam similarmente às unidades de processamento 11 descritas aqui para obter acesso ao espaço compartilhado 15. Finalmente, embora apenas uma única barra 13 seja mostrada, pode haver uma pluralidade de barras no mesmo ou em diferentes níveis hierárquicos, tal como a barra 13 para acoplar os vários dispositivos.

O acesso da memória 12 pelas unidades de processamento 11 para a transferência de dados tipicamente envolve o uso de operações de carga e armazenamento. A operação de carga transfere o conteúdo de memória de uma localização na memória acessada e a operação de armazenamento transfere os dados para uma localização de memória acessada. Desse modo, operações de carregar/armazenar são usadas para acessar a memória 12 e o bloqueio 16 para transferência de dados entre unidades de

processamento 11 e a memória 12. Os acessos de carga e armazenamento são também referidos como acessos de leitura e gravação, respectivamente.

Com referência às figuras 1 e 2, o sistema de computador 10 inclui uma memória de leitura apenas (ROM) 31 e uma memória principal 18
5 acopladas -- através da barra de sistema 22 -- com as unidades de processamento 11, a memória principal 18 compreendendo, por exemplo, qualquer tipo adequado de memória de acesso aleatório (RAM). Unidades de processamento 11 também apresentam um dispositivo de armazenamento de dados 30 acoplado com as mesmas pela barra de sistema 22. O dispositivo de
10 armazenamento de dados 30 compreende qualquer memória não volátil adequada, tal como, por exemplo, uma unidade de disco rígido. O sistema de computador 10 adicionalmente inclui um meio de armazenamento removível 32, tal como uma unidade de disco flexível, uma unidade de CR ROM, e/ou uma unidade USB.

15 Na figura 2, a unidade de processamento 11 inclui inúmeros componentes que são interconectados por uma ou mais barras, e estas barras são ilustradas simbolicamente na figura 2 por uma barra local 19. A barra local 19, e, conseqüentemente, os componentes da unidade de processamento 11, são acoplados com uma unidade de interface de barra 23. A uni-
20 dade de interface de barra 23 acopla a unidade de processamento 11 com a barra de sistema 22, permitindo assim a comunicação entre a unidade de processamento 11 e a memória principal 18, bem como entre a unidade de processamento 11 e um cache externo 20.

A unidade de processamento 11 inclui um decodificador de ins-
25 trução 21 acoplado com a barra local 19. O decodificador 21 recebe uma instrução (ou instruções) associada com um programa ou pedaço de código que executa na unidade de processamento 11 e rompe a instrução em uma ou mais operações/instruções de nível de máquina (uops). Deve ser entendido que a unidade de processamento 11 pode receber uma ou mais instru-
30 ções associadas com um programa, ao passo que outra unidade de processamento 11 do sistema de computador 10 pode receber uma ou mais instruções associadas com o mesmo programa. Conseqüentemente, um progra-

ma pode ser executado em múltiplas unidades de processamento 11.

A unidade de processamento 11 adicionalmente inclui múltiplas unidades de execução, incluindo, por exemplo, unidade de controle de acesso de dados (DAC) 24, buffer de ordenação de memória (MOB) 26, uma unidade de arquivo de registro 29, e uma unidade funcional 27.

A unidade de arquivo de registro 29 inclui uma pluralidade de registros, cada qual apresentando 16, 32, 64, 128, 256, 512 bits de armazenamento. Além disso, o arquivo de registro 29 pode incluir um ou mais arquivos de registro, cada qual apresentando um ou mais registros. A unidade funcional 27 compreende uma ou mais unidades funcionais, tais como, por exemplo, uma unidade aritmética, lógica e/ou de ponto flutuante. O MOB 26 assegura a ordenação adequada de instruções de carregar e armazenar e, adicionalmente, provê as sequências adequadas destas transações dentro da hierarquia de memória (isto é, vários níveis de memória dentro do sistema de computador 10, incluindo o cache L0 25, o cache L1 28, o cache externo 20, a memória principal 18, e o dispositivo de armazenamento de dados 30). Cada cache, o cache L0 25 e o cache L1 28, pode armazenar dados recentemente acessados, ou a serem acessados, pela unidade funcional 27. Se um item de dados solicitado pela unidade funcional 27 estiver em uma das memórias cache 25, 28, um "acerto" de cache terá ocorrido; entretanto, se os dados solicitados não estiverem presentes no cache, então um "erro" de cache terá ocorrido. Uma ou mais das memórias cache (por exemplo, o cache L0 25) podem ser acopladas com o DAC 24. O DAC 24 controla todas as transações que resultaram em um erro de cache, bem como outras transações que exigem um controle especial. Um bloqueio, conforme descrito acima, é um tipo de transação que exige controle especial pelo DAC 24 e por outros componentes da unidade de processamento 11. Se uma uop corresponder, por exemplo, a uma operação aritmética, essa uop será despachada para a unidade funcional 27, que executa então a operação aritmética. Se uma uop corresponder a uma instrução de referência de memória, por exemplo, uma carga ou um armazenamento, essa uop será despachada para o MOB 26.

Deve ser entendido que a unidade de processamento 11 ilustrada na figura 2 se destina a representar um dispositivo de processamento exemplificativo e que, adicionalmente, tal unidade de processamento pode incluir muitos componentes adicionais que não são mostrados nestas figuras. Estes componentes foram omitidos para facilidade de entendimento. Por exemplo, a unidade de processamento 11 pode incluir uma unidade de geração de endereço, uma estação de reserva, um buffer de reordenação, um escalonamento, uma unidade de segmentação e translação de endereço, um buffer de tradução de endereços, uma rotina de tratamento de erro de página, e/ou circuitos de relógio internos. Também, embora ilustrados como elementos distintos, deve ser entendido que muitos dos componentes mostrados na figura 2 podem ser circuitos combinados e/ou compartilhados. Mais importante, as concretizações descritas aqui não são limitadas a qualquer arquitetura ou disposição específica -- bem como não são limitadas a qualquer terminologia específica usada para descrever tal arquitetura ou disposição -- e as concretizações descritas podem ser praticadas em qualquer tipo de dispositivo de processamento, independente de sua arquitetura ou da terminologia imputada a ela.

Qualquer uma ou mais das uops programadas para execução podem compreender uma uop travada. O bloqueio, conforme notado acima, corresponde a uma sequência de operações (por exemplo, carregar, modificar, e armazenar) que são executadas em uma maneira que assegure a sincronização entre os processadores e/ou os threads.

A figura 3 mostra uma instrução para executar uma operação de ler-modificar-gravar. A instrução 40 é uma instrução atômica única incluindo cinco operandos 41-45. O operando de código de operação 41 identifica que esta é uma instrução VCM PXCHG. Os operandos 42-44 correspondem aos operandos de fonte e destino associados com SRC1/DEST, SRC2, SRC3 e, em algumas implementações, a uma localização de armazenamento de máscara (MSK) e/ou a um operando de desvio (ou "imediato") 45. Este operando deslocado ou imediato é usado para prover um deslocamento de um endereço de base (tal como SRC1), quando do endereçamento da memória

12. As instruções notadas abaixo podem ter tais deslocamentos, mas não são ilustradas. Implementações que especificam uma localização de armazenamento de máscara 45 se referem a uma localização de memória ou registro que armazena elementos de máscara correspondendo aos respectivos
5 elementos de dados armazenados na localização de armazenamento referenciada pelo operando SRC1/DEST.

Em resposta à instrução 40, a unidade de processamento 11 lê primeiros dados de fonte, os compara a outros dados de fonte e, se a comparação satisfizer uma condição pré-indicada (tal como uma condição verdadeira ou de associação), algum valor modificado será gravado em uma
10 localização, que pode ser a localização original dos primeiros dados de fonte. Se a condição pré-indicada não for satisfeita, os dados originais na localização não serão mudados. A instrução utiliza três operandos de fonte (tais como SRC1, SRC2 e SRC3, conforme usado adiante) e um operando de
15 destino (tal como DEST, conforme usado adiante) para suprir a localização das várias informações usadas na execução da instrução. Os registros específicos de operação podem ser usados para prover um ou mais dos dados de fonte e/ou para armazenar os dados de destino, quando a instrução for executada, eliminando a necessidade de especificar os operandos explicitamente no atual formato de instrução. Além disso, neste exemplo, o operando
20 SRC1 e o operando DEST refere-se à mesma localização de armazenamento (SRC1/DEST).

Antes da execução da instrução 40, SRC1, SRC2 e SRC3 são carregados em registros na unidade de arquivo de registro 29. Por exemplo,
25 para atualizar com segurança um valor armazenado em uma localização especificada pelo operando SRC1/DEST, o valor é inicialmente lido em uma localização especificada pelo operando SRC2, e um valor de substituição é lido em uma localização especificada pelo operando SRC3. Depois, uma operação de comparar-trocar atômica é executada para comparar o valor
30 presente associado com o operando SRC1/DEST ao valor associado com o operando SRC2 (isto é, o valor presente pode ser diferente do valor inicialmente copiado devido à modificação por outro agente). Se o valor não tiver

mudado, ele será substituído pelo valor associado com o operando SCR3, e a bandeira zero será ajustada para indicar uma atualização de sucesso. Entretanto, se outro agente tiver modificado o valor entre a cópia inicial e a operação de comparar-trocar, o valor corrente não será substituído e a bandeira zero ficará livre para indicar uma atualização de falha.

O diagrama de bloco da figura 4 ilustra o fluxo de informação, quando da execução da instrução 40. A unidade de processamento 11 inclui uma unidade de execução 46 (por exemplo, DAC 24 da figura 2), arquivo de registro 29, BIU 23, e decodificador 21, que são todos acoplados entre si pela barra local 19. O arquivo de registro 29 inclui uma pluralidade de registros que são acessados pela unidade de execução 46 para executar as várias operações. Conforme notado na figura 4, a instrução VCMPXCHG 40 é mostrada dentro da unidade de execução 46 e linhas pontilhadas são mostradas a partir dos operandos da instrução para os registros correspondentes associados com SRC1, SRC2, SRC3 e DEST. Os registros estão dentro do arquivo de registro 29. O decodificador 21 é usado para decodificar as várias instruções (incluindo a instrução VCMPXCHG 40), a fim de que a unidade de execução 46 execute as operações.

A memória 12, anteriormente descrita nas figuras 1 e 2, é mostrada acoplada à BIU 23 pela barra 19 e/ou barra 22. Consequentemente, as transferências de dados entre a unidade de processamento 11 e a memória 12 podem ocorrer através da BIU 54a ou barra local 19. Deve ser apreciado que a rotina do programa que utiliza a instrução VCMPXCHG 40 pode estar dentro de alguma memória, que poderia também ser ou incluir a memória 12.

O pseudocódigo seguinte ilustra exemplos de como opera a instrução VCMPXCHG 40. Outros pseudocódigos, linguagens, operações, ordens de operações, e/ou números podem ser usados.

VCMPXCHG (versão codificada VEX.128)	VCMPXCHG (versão codificada VEX.256)
IF (DEST[127:0] == SRC2[127:0]) THEN DEST[127:0] ← SRC3[127:0] ZF ← 1	IF (DEST[255:0] == SRC2[255:0]) THEN DEST[255:0] ← SRC3[255:0] ZF ← 1

ELSE SRC2[127:0] ← DEST[127:0] ZF ← 0 FI SRC2[511:128] ← 0	ELSE SRC2[255:0] ← DEST[255:0] ZF ← 0 FI SRC2[511: 256] ← 0
--	---

Nos exemplos específicos VEX.128 e VEX.256 notados acima, os valores de bloqueio são armazenados em bits [127:0] e bits [255:0], respectivamente, de uma localização de armazenamento de 512 bits referenciada por SRC1/DEST (por exemplo, um registro ou linha de cache de 64 bytes). Em uma concretização, é provida uma correspondência de um para um entre os valores de bloqueio referenciados por SRC1/DEST e as localizações de armazenamento compartilhadas 15 às quais eles correspondem. Por exemplo, SRC1/DEST pode fazer referência a dezesseis valores de bloqueio de 8 bits (128 bits), cada qual correspondendo a uma das dezesseis localizações de armazenamento em um registro SIMD ou linha de cache. Alternativamente, SRC1/DEST pode fazer referência a trinta e dois valores de bloqueio de 8 bits (256 bits), cada qual correspondendo a uma respectiva das trinta e duas localizações de armazenamento em um registro SIMD ou linha de cache.

Com referência novamente aos exemplos acima, o resultado da comparação entre SRC1/DEST e SRC2 indica se os valores de trava foram modificados. Uma condição verdadeira indica que os bloqueios não foram modificados e que os bloqueios estão no estado destravado. Quando esta condição for atendida, os valores referenciados por SRC3 serão gravados em SRC1/DEST, modificando os valores de bloqueio para um estado travado, a fim de impedir que outros agentes tenham acesso ao(s) espaço(s) compartilhado(s). Em seguida, a bandeira zero (ZF) é ajustada para indicar uma operação de sucesso.

Uma falsa condição indica que um ou mais dos bloqueios foram modificados (travados) e que outro agente obteve a propriedade do espaço compartilhado. Quando a condição for falsa, os valores referenciados por SRC1/DEST (os valores de bloqueio correntes) serão armazenados em SRC2, e a bandeira zero será liberada para indicar uma operação mal sucedi-

da. Os bytes superiores de SRC2 são então liberados antes de retornar da operação.

Tipicamente, se o acesso for primeiramente recusado, o agente de interrogação continuará a tentar novamente o acesso até que este seja adquirido. Em algumas implementações, um laço externo irá incluir uma carga não atômica e teste antes para reexecutar a instrução VCMPXCHG 40. Uma vez que o processador complete seu(s) acesso(s) ao espaço de memória compartilhado 15, ele irá tipicamente liberar seu controle sobre o espaço de memória compartilhado 15 com um ciclo de gravação para que o bloqueio 16 seja destravado, de modo que outros agentes possam agora conseguir a entrada no espaço de memória compartilhado 15. Contudo, é apreciado que a maneira como o processador libera o espaço de memória compartilhado é uma escolha de desenho, que poderia ser ditada pela arquitetura do sistema.

Em algumas implementações, a instrução VCMPXCHG 40 inclui um vetor de máscara apresentando múltiplos elementos de máscara, cada qual correspondendo a um de uma pluralidade de elementos de dados referenciados por SRC1/DEST. A localização de armazenamento de vetor de máscara pode ser um registro em uma unidade de arquivo de registro 29, tal como, por exemplo, um registro de sombra, um registro de controle, um registro de bandeira, um registro de uso geral, um registro SIMD, ou outro registro apropriado. Em uma concretização, há uma correspondência de um para um entre os elementos de dados referenciados por SRC1/DEST e elementos de máscara correspondentes armazenados em um registro de máscara. Os elementos ou valores de máscara podem incluir bandeiras, marcadores, tabuladores, indicadores e/ou outros números, bits e/ou códigos para indicar se um elemento de dados correspondente (por exemplo, em uma localização de registro correspondente ou indicada) é comparado e/ou modificado. Por exemplo, um elemento de máscara apresentando um valor de "1" pode indicar que um elemento de dados correspondente é modificado; de outro modo, "0" pode ser usado. Outros números ou bandeiras podem ser usados.

Exemplos de instruções VCMPIXCHGD e VCMPIXCHGQ mascaradas são ilustrados no pseudocódigo abaixo para um vetor de largura 16 e de 512 bytes e um vetor de largura 8 e 512 bytes, respectivamente. Nas implementações de comparação mascaradas, apenas os elementos ativos são comparados e atualizados.

VCMPIXCHGD (versão codificada EVEX.512) ALL_CMPS_SUCCEED←1 FOR j←0 TO 15 i←j*32 IF k1[j] OR *no writemask* THEN IF(DEST[i+31:i]≠SRC2[i+31:i]) THEN ALL_CMPS_SUCCEED←0 FI ENDFOR IF(ALL_CMPS_SUCCEED == 1) THEN ZF←1 FOR j←0 TO 15 i←j*32 IF k1[j] OR *no writemask* THEN DEST[i+31:i] ←SRC3[i+31:i] FI ENDFOR ELSE ZF←0 FOR j←0 TO 15	VCMPIXCHGQ (versão codificada EVEX.512) ALL_CMPS_SUCCEED←1 FOR j←0 TO 7 i←j*64 IF k1[j] OR *no writemask* THEN IF(DEST[i+63:i]≠SRC2[i+63:i]) THEN ALL_CMPS_SUCCEED←0 FI ENDFOR IF(ALL_CMPS_SUCCEED == 1) THEN ZF←1 FOR j←0 TO 7 i←j*64 IF k1[j] OR *no writemask* THEN DEST[i+63:i] ←SRC3[i+63:i] FI ENDFOR THEN ZF←0 FOR j←0 TO 7
--	---

$i \leftarrow j * 32$ IF k1[j] OR *no writemask* THEN SRC2[i+31:i] $\leftarrow \text{DEST}[i+31:i]$ FI ENDFOR FI	$i \leftarrow j * 64$ IF k1[j] OR *no writemask* THEN SRC2[i+63:i] $\leftarrow \text{DEST}[i+63:i]$ FI ENDFOR FI
--	--

Nos exemplos VCMPXCHGD e VCMPXCHGQ específicos notados acima, uma variável ALL_CMPS_SUCCEED, é primeiro pré-ajustada em 1 (isto é, uma condição verdadeira). Uma vez ajustada, para cada elemento de máscara ativo (por exemplo, um elemento de máscara apresentando um valor específico armazenado no mesmo, incluindo, por exemplo, um binário 1 ou valores hexadecimais 0x01, 0xFF, ou 0x80), a localização de armazenamento correspondente referenciada por SRC1/DEST é comparada ao valor referenciado pelos bits correspondentes em SRC2. Se nenhuma máscara for usada, cada das localizações de armazenamento referenciadas pelo SRC1/DEST será comparada ao valor referenciado pelos bits correspondentes em SRC2.

Novamente, o resultado da comparação entre os valores correspondentes de SRC1/DEST e SRC2 indica se esse valor de trava específico foi modificado. Entretanto, nestes exemplos, uma condição verdadeira (isto, é, uma condição de não associação) indica que o bloqueio foi modificado e que outro agente obteve a propriedade da localização de armazenamento compartilhada. Quando esta condição for atendida para qualquer das localizações de armazenamento referenciadas, ALL_CMPS_SUCCEED será liberada, indicando que todas as comparações não tiveram sucesso. Depois disso, a bandeira zero é liberada e, para cada elemento de máscara ativo, o valor armazenado na localização de armazenamento correspondente referenciada por SRC1/DEST é carregado nos bits correspondentes em SRC2.

Quando o resultado da comparação for falso (isto é, para cada dos elementos de máscara ativos, um valor correspondente referenciado por SRC1/DEST associada a um valor correspondente em SRC2),

ALL_CMPS_SUCCEED permanecerá ajustado sem mudança. Depois disso, a bandeira zero (ZF) é ajustada e, para cada elemento de máscara ativo, o valor armazenado na localização de armazenamento correspondente em SRC3 é carregado em bits correspondentes em SRC1/DEST, modificando os valores de bloqueio para um estado travado, a fim de impedir que outros agentes obtenham acesso aos espaços compartilhados.

O diagrama de bloco da figura 5 ilustra outro exemplo do fluxo de informação, quando da execução da instrução 40. Conforme notado na figura 5, a instrução VCMXCHG 40 é mostrada dentro da unidade de execução 46 e linhas pontilhadas são mostradas a partir dos operandos da instrução para os registros correspondentes associados com SRC2, SRC3 e MSK. Neste exemplo, a localização de armazenamento de máscara (MSK) é um registro de máscara e a localização de armazenamento associada com SRC1/DEST é um cache L1. Os registros estão dentro da unidade de arquivo de registro 29.

Antes da execução da instrução 40, SRC1 é pré-buscado no cache L1 e os SRC2, SRC3 e os dados MSK são carregados nos registros na unidade de arquivo de registro 29. O registro de máscara armazena uma pluralidade de elementos de máscara correspondendo aos respectivos elementos de dados na localização de armazenamento associada com o operando SRC1/DEST. Além disso, um valor de comparação é inicialmente lido em uma localização especificada pelo operando SRC2, e um valor de substituição é lido em uma localização específica pelo operando SRC3. Depois, a instrução 40 é executada para fazer com que a unidade de execução 46 compare os elementos de dados correspondentes com SRC1/DEST e os operandos SRC2, e, se existir uma associação, para substituir os elementos de dados de SRC1/DEST pelos elementos de dados correspondentes de SRC3. Se não existir uma associação, a execução da instrução 40 fará com que a unidade de execução 46 substitua os elementos de dados SRC2 pelos elementos de dados SRC1/DEST correspondentes.

Em algumas implementações, a comparação entre os pares de SRC1/DEST e os elementos de dados SRC2 será apenas executada, se um

elemento de máscara correspondente estiver ativo. Em certas implementações, a unidade de execução 46 será adicionalmente configurada para ajustar uma bandeira, se houver uma associação entre cada par de elementos de dados correspondentes para os quais um elemento de máscara correspondente está ativo e para liberar a bandeira, se uma associação não existir entre qualquer dos pares para os quais está ativo um elemento de máscara correspondente. Além disso, em algumas implementações, a substituição dos elementos de dados SRC1/DEST pelos elementos de dados SRC3 correspondente é apenas executada se um elemento de máscara correspondendo ao respectivo elemento de dados SRC1/DEST estiver ativo. Além disso, em algumas implementações, a substituição dos elementos de SRC2 por um elemento de dados SRC1/DEST correspondente é apenas executada se um elemento de máscara correspondendo ao elemento de dados SRC1/DEST estiver ativo.

Em algumas concretizações, o valor de bloqueio indicando uma condição travada é igual ao valor de máscara que indica um elemento de máscara ativo (por exemplo, um 1 binário). Em tais casos, SRC3 pode ser usado tanto como um vetor de máscara como um vetor de substituição de valor de bloqueio.

Em algumas implementações, a operação de comparar-trocar completa a execução sem atualizar o valor associado com o operando SRC2. Depois disso, uma bandeira (por exemplo, a bandeira zero) é testada e, caso indique uma operação de atualização com falha com relação ao valor associado com o operando SRC1/DEST, as etapas bem antes da operação de comparar-trocar serão repetidas para atualizar os valores associados com SRC2 e SRC3 antes de repetir a operação de comparar-trocar.

Uma ou mais concretizações incluem um artigo de fabricação que inclui um meio tangível acessível à máquina e/ou legível à máquina apresentando no mesmo uma instrução SIMD que especifica uma operação de comparar-trocar de vetor para uma pluralidade de elementos de dados, cada elemento de dados apresentando um elemento de teste correspondente, um elemento de substituição e um elemento de máscara, que, caso exe-

cutado por uma máquina (por exemplo, uma unidade de execução), faz com que a máquina compare os elementos de dados com os elementos de teste correspondentes, se os respectivos elementos de máscara estiverem ativos; e responsiva à determinação de que todas as comparações indicam uma associação, ajusta uma bandeira e substitui os elementos de dados comparados pelos elementos de substituição correspondentes; e, responsiva à determinação de que todas as comparações não indicam uma associação, libera uma bandeira e substitui elementos de teste comparados com elementos de dados correspondentes. O meio tangível pode incluir um ou mais materiais sólidos. O meio pode incluir um mecanismo que apresenta, por exemplo, armazenamentos, informação em uma forma que é acessível pela máquina. Por exemplo, o meio pode opcionalmente incluir meios graváveis, tais como, por exemplo, disquete flexível, meio de armazenamento óptico, disco óptico, CR-ROM, disco magnético, disco magneto-óptico, memória de leitura apenas (ROM), ROM programável (PROM), ROM apagável e programável (EPROM), ROM eletricamente apagável e programável (EEPROM), memória de acesso aleatória (RAM), RAM estática (SRAM), RAM dinâmica (DRAM), memória "flash", e combinações dos mesmos.

Máquinas adequadas incluem, mas não são limitadas a unidades de execução, processadores de uso geral, processadores de uso especial (por exemplo, processadores gráficos e processadores criptográficos), aceleradores criptográficos, processadores de comunicações de rede, sistema de computador, dispositivos de rede, modems, assistentes digitais pessoas (PDAs), telefones celulares, e uma ampla variedade de outros dispositivos eletrônicos com uma ou mais unidades de execução, apenas para citar alguns. Ainda, outras concretizações se referem a um sistema de computador, sistema embutido, ou a outro dispositivo eletrônico apresentando uma unidade de execução e/ou executando um método, conforme descrito aqui.

A figura 6 ilustra um exemplo de um sistema de computador adequado 50 que inclui um processador 51. O processador inclui pelo menos uma unidade de execução 52 que é capaz de executar pelo menos uma ins-

trução de comparação e troca de vetor 53.

O processador é acoplado a um conjunto de circuitos integrados 54 através de uma barra (por exemplo, uma barra lateral frontal) ou outra interconexão 55. A interconexão pode ser usada para transmitir sinais de dados entre o processador e outros componentes no sistema através do conjunto de circuitos integrados.

O conjunto de circuitos integrados inclui um chip lógico de sistema conhecido como um cubo controlador de memória (CH) 56. O MCH é acoplado à barra lateral dianteira ou a outra interconexão 55.

Uma memória 58 é acoplada ao MCH. Em várias concretizações, a memória pode incluir uma memória de acesso aleatório (RAM). DRAM é um exemplo de um tipo de RAM usado em alguns, mas não em todos os sistemas de computador. Conforme mostrado, a memória pode ser usada para armazenar instruções 59, tais como uma ou mais instruções de multiplicação, e dados 60.

Uma interconexão de componentes 61 é também acoplada com o MCH. Em uma ou mais concretizações, a interconexão de componentes pode incluir uma ou mais interfaces expressas de interconexão de componentes periféricas (PCIe). A interconexão de componentes pode permitir que outros componentes sejam acoplados ao resto do sistema através do conjunto de circuitos integrados. Um exemplo de tais componentes é um chip gráfico ou outro dispositivo gráfico, embora este seja opcional e não exigido.

O conjunto de circuitos integrados também inclui um cubo controlador de entrada/saída (I/O) (ICH) 62. O ICH é acoplado ao MCH através da barra de interface de cubo ou outra interconexão 63. Em uma ou mais concretizações, a barra ou outra interconexão 63 pode incluir uma Interface de Mídia Direta (DMI).

Um armazenamento de dados 64 é acoplado ao ICH. Em várias concretizações, o armazenamento de dados pode incluir uma unidade de disco rígido, uma unidade de disco flexível, um dispositivo de CD-ROM, um dispositivo de memória flash, ou semelhante, ou uma combinação dos mesmos.

Uma segunda interconexão de componentes 65 também é acoplada com o ICH. Em uma ou mais concretizações, a segunda interconexão de componentes pode incluir uma ou mais interfaces expressas de interconexão de componentes periférica (PCIe). A segunda interconexão de componentes pode permitir que vários tipos de componentes sejam acoplados ao resto do sistema através do conjunto de circuitos integrados.

Uma porta de expansão serial 66 também é acoplada com o ICH. Em uma ou mais concretizações, a porta de expansão serial pode incluir uma ou mais portas de barra serial universal (USB). A porta de expansão serial pode permitir que vários outros tipos de dispositivos de entrada/saída sejam acoplados ao resto do sistema através do conjunto de circuitos integrados.

Alguns exemplos ilustrativos de outros componentes que podem opcionalmente ser acoplados com o ICH incluem, mas não são limitados a um controlador de áudio, um transceptor sem fio, e um dispositivo de entrada de usuário (por exemplo, um teclado, um mouse).

Um controlador de rede 67 é também acoplado ao ICH. O controlador de rede pode permitir que o sistema seja acoplado com uma rede.

Em uma ou mais concretizações, o sistema de computador pode executar uma versão do sistema de operação WINDOWS®, disponível pela Microsoft Corporation of Redmond, Washington. Alternativamente, podem ser usados outros sistemas de operação, tais como, por exemplo, UNIX, Linux, ou sistemas embutidos.

Este é apenas um exemplo específico de um sistema de computador adequado. Por exemplo, em uma ou mais concretizações alternativas, o processador pode ter múltiplos núcleos. Como outro exemplo, em uma ou mais concretizações alternativas, o MCH 56 pode ser fisicamente integrado na matriz com o processador 51 e o processador pode ser diretamente acoplado com uma memória 58 através do MCH integrado. Como um exemplo adicional, em uma ou mais concretizações alternadas, outros componentes podem ser integrados na matriz com o processador, tal como para prover um desenho de sistema em chip (SoC). Ainda como outro exemplo, em uma ou

mais concretizações alternativas, o sistema de computador pode ter múltiplos processadores.

A figura 7 é outro exemplo de um sistema de computador adequado 70. A segunda concretização exemplificativa apresenta certas similaridades ao sistema de computador 50 descrito acima. Para fins de clareza, a discussão tenderá a enfatizar as diferenças sem repetir todas as similaridades.

Similar ao sistema de computador 50, o sistema de computador 70 inclui um processador 71, e um conjunto de circuitos integrados 74 apresentando um cubo controlador I/O (ICH) 72. O sistema de computador 70 também inclui uma primeira interconexão de componentes 81 acoplada com o conjunto de circuitos integrados 74, uma segunda interconexão de componentes 85 acoplada com o ICH, uma porta de expansão serial 86 acoplada com o ICH, um controlador de rede 87 acoplado com o ICH, e um armazenamento de dados 84 acoplado com o ICH.

O processador 71 é um processador de múltiplos núcleos e inclui núcleos processadores 72-1 a 72-M, onde M pode ser um número inteiro igual ou maior do que dois (por exemplo, dois, quatro, sete ou mais). Cada núcleo pode incluir pelo menos uma unidade de execução que é capaz de executar pelo menos uma concretização de uma instrução, conforme descrito aqui. Conforme mostrado, o núcleo 1 inclui um cache 88 (por exemplo, um cache L1). Cada dos outros núcleos pode similarmente incluir um cache dedicado. Os núcleos de processador podem ser implementados em um único chip de circuito integrado (IC).

O processador também inclui pelo menos um cache compartilhado 89. O cache compartilhado pode armazenar dados (por exemplo, instruções) que são utilizados por um ou mais componentes do processador, tais como núcleos. Por exemplo, o cache compartilhado pode localmente colocar em cache os dados armazenados em uma memória 78 para um acesso mais rápido pelos componentes do processador. Em uma ou mais concretizações, o cache armazenado pode incluir um ou mais caches de nível médio, tais como o nível 2(L2), o nível 3(L3), o nível 4(L4) ou outros

níveis de cache, um cache de último nível (LLC), e/ou combinações dos mesmos.

Os núcleos de processador e o cache compartilhado são cada qual acoplados com uma barra ou outra interconexão 90. A barra ou outra
5 interconexão pode acoplar os núcleos e o cache compartilhado e permitir a comunicação.

O processador também inclui um cubo controlador de memória (MCH) 76. Conforme mostrado nesta concretização exemplificativa, o MCH é integrado com o processador 71. Por exemplo, o MCH pode ser em matriz
10 com os núcleos de processador. O processador é acoplado com a memória 78 através do MCH. Em uma ou mais concretizações, a memória pode incluir DRAM, embora isto não seja exigido.

O conjunto de circuitos integrados inclui um cubo de entrada/saída (I/O) 91. O cubo I/O é acoplado com o processador através de uma
15 barra (por exemplo, uma Interconexão QuickPath (QPI)) ou outra interconexão 75. A primeira interconexão de componentes 81 é acoplada com o cubo I/O 91.

Este é apenas um exemplo de um sistema adequado. Outros desenhos de sistema e configurações conhecidos na técnica para laptops,
20 desktops, PCs de mão, assistentes digitais pessoais, estações de trabalho de engenharia, servidores, dispositivos de rede, cubos de rede, comutadores, processadores embutidos, processadores de sinal digital (DSPs), dispositivos gráficos, dispositivos de vídeo game, conversores (set-top boxes), microcontroladores, telefones celulares, reprodutores de mídia portátil, dis-
25 positivos de mão, e vários outros dispositivos eletrônicos também são adequados. Em geral, uma enorme variedade de sistemas ou dispositivos eletrônicos capazes de incorporar um processador e/ou uma unidade de execução, conforme descrito aqui, é geralmente adequada.

Na descrição acima, para fins de explanação, inúmeros detalhes
30 específicos foram explicados a fim de prover um completo entendimento das concretizações da invenção. Ficará evidente, contudo, àquele versado na técnica, que uma ou mais outras concretizações podem ser praticadas sem

alguns destes detalhes específicos. As concretizações específicas descritas não são providas para limitar a invenção, mas para ilustrar as concretizações da invenção. O escopo da invenção não deve ser determinado pelos exemplos específicos providos acima, mas apenas pelas reivindicações abaixo.

5 Em outros exemplos, circuitos, estruturas, dispositivos e operações bem conhecidos foram mostrados na forma de diagrama de bloco ou sem detalhes a fim de impedir que o entendimento da descrição seja obscurecido. Quando considerados apropriados, numerais de referência ou porções de terminal de numerais de referência foram repetidas entre as figuras para indicar elementos correspondentes ou análogos, que podem opcionalmente ter características similares.

Certas operações podem ser executadas por componentes de hardware, ou podem ser concretizadas em instruções executáveis por máquina, que podem ser usadas para produzir, ou pelo menos resultar em um
15 circuito ou hardware programado com as instruções que executam as operações. O circuito pode incluir um processador de uso geral ou de uso especial, ou circuito lógico, apenas para citar alguns. As operações podem ser opcionalmente também executadas por uma combinação de hardware ou software. Uma unidade de execução e/ou um processador pode incluir circuitos
20 específicos ou particulares ou outra lógica responsiva a uma instrução de máquina ou a um ou mais sinais de controle derivados da instrução de máquina para armazenar um operando de resultado especificado por instrução.

Deve também ser apreciado que a referência por todo este relatório descritivo a "uma concretização", "a concretização", ou "uma ou mais
25 concretizações", por exemplo, indica que uma característica específica pode ser incluída na prática das concretizações da invenção. Similarmente, deve ser apreciado que, na descrição, várias características são muitas vezes agrupadas entre si em uma única concretização, figura, ou descrição da mesma para fins de aperfeiçoar a descrição e ajudar no entendimento dos
30 vários aspectos da invenção. Este método de descrição, contudo, não deve ser interpretado como refletindo uma intenção de que a invenção exige mais características do que são expressamente citadas em cada reivindicação.

De preferência, como refletem as seguintes reivindicações, os aspectos da invenção podem estar em menos do que todas as características de uma única concretização descrita. Desse modo, as reivindicações que seguem a Descrição Detalhada são aqui expressamente incorporadas nesta Descrição
5 Detalhada, com cada reivindicação sendo sustentada por si só como uma concretização separada da invenção.

Inúmeras concretizações da invenção foram descritas. Contudo, será entendido que várias modificações podem ser feitas sem se afastar do espírito e do escopo da invenção. Por exemplo, os sistemas de computador
10 precisam ser limitados a sistemas de computador apresentando múltiplos processadores ou dispositivos de acesso de memória. A presente invenção poderia ser prontamente utilizada em um único sistema de processador, onde uma instrução de leitura-modificação-gravação é implementada.

É adicionalmente apreciado que os controles de acesso para as
15 áreas compartilhadas da memória podem ser obtidos por meio que não a sequência de testar e ajustar descrita no exemplo acima. Por exemplo, um simples contador pode ser usado, no qual cada acesso é incrementado por uma contagem específica.

Também é apreciado que a instrução VCMPXCHG da concreti-
20 zação preferida executa a operação de ler-modificar-gravar, mas as fases de modificação e gravação são obtidas essencialmente como uma etapa única. Em vez de calcular os valores de modificação depois da leitura dos dados originais e então subsequentemente gravar os valores modificados, os valores modificados da instrução VCMPXCHG são preestabelecidos para uso
25 pela instrução. Embora seu uso dependa da decisão obtida, quando da comparação, estes valores de modificação preestabelecidos (SRC3) podem ser gravados imediatamente no destino para modificar o valor de destino.

Desse modo, é descrita uma técnica para implementar uma operação de comparar-trocar de vetor que utiliza uma máscara. É apreciado que
30 as instruções VCMPXCHG e implementações descritas aqui podem ser utilizadas em outras capacidades também e não precisam ficar limitadas à função de controlar o acesso a um espaço de memória compartilhado. Por e-

xemplo, as instruções VCMPXCHG podem ser usadas para a execução especulativa na qual uma operação SIMD é executada em uma pluralidade de elementos de dados, os resultados dos quais sendo apenas gravados no espaço de memória compartilhado, se os elementos de dados não tiverem sido modificados por outro agente durante a operação. Consequentemente, outras concretizações estão dentro do escopo das seguintes reivindicações.

REIVINDICAÇÕES

1. Método que compreende:

5 a decodificação por um decodificador em um dispositivo de processamento de uma única instrução que especifica uma operação de comparação e troca de vetor para uma pluralidade de elementos de dados entre uma primeira localização de armazenamento, uma segunda localização de armazenamento, e uma terceira localização de armazenamento;

10 a emissão da única instrução para execução por uma unidade de execução no dispositivo de processamento; e, responsiva à execução da única instrução,

a comparação dos elementos de dados da primeira localização de armazenamento aos elementos de dados correspondentes na segunda localização de armazenamento; e, responsiva à determinação de que existe uma associação,

15 a substituição dos elementos de dados da primeira localização de armazenamento pelos elementos de dados correspondentes da terceira localização de armazenamento.

20 2. Método, de acordo com a reivindicação 1, em que a única instrução adicionalmente especifica uma localização de armazenamento de máscara que armazena uma pluralidade de elementos de máscara correspondendo aos respectivos elementos de dados na primeira localização de armazenamento.

25 3. Método, de acordo com a reivindicação 2, em que a comparação dos elementos de dados da primeira localização de armazenamento com os elementos de dados correspondentes na segunda localização de armazenamento compreende:

30 a comparação de um elemento de dados da primeira localização de armazenamento com um elemento de dados correspondentes na segunda localização de armazenamento, quando um elemento de máscara correspondendo ao elemento de dados da primeira localização de armazenamento estiver ativo.

4. Método, de acordo com a reivindicação 2, em que a substitui-

ção dos elementos de dados da primeira localização de armazenamento pelos elementos de dados correspondentes da terceira localização de armazenamento compreende:

5 a substituição de um elemento de dados da primeira localização de armazenamento por um elemento de dados correspondentes da terceira localização de armazenamento, quando um elemento de máscara correspondendo ao elemento de dados da primeira localização de armazenamento estiver ativo.

10 5. Método, de acordo com a reivindicação 1, que adicionalmente compreende:

a substituição dos elementos de dados da segunda pluralidade de elementos de dados com os elementos de dados correspondentes da primeira localização de armazenamento, quando da não existência de uma associação.

15 6. Método, de acordo com a reivindicação 5, em que a única instrução adicionalmente especifica uma localização de armazenamento de máscara que armazena uma pluralidade de elementos de máscara correspondendo aos respectivos elementos de dados na primeira localização de armazenamento.

20 7. Método, de acordo com a reivindicação 6, em que a comparação dos elementos de dados da primeira localização de armazenamento com os elementos de dados correspondentes na segunda localização de armazenamento compreende:

25 a comparação de um elemento de dados da primeira localização de armazenamento com um elemento de dados correspondente na segunda localização de armazenamento, quando um elemento de máscara correspondendo ao elemento de dados da primeira localização de armazenamento estiver ativo.

30 8. Método, de acordo com a reivindicação 6, em que a substituição dos elementos de dados da primeira localização de armazenamento pelos elementos de dados correspondentes da terceira localização de armazenamento compreende:

a substituição de um elemento de dados da primeira localização de armazenamento por um elemento de dados correspondente da terceira localização de armazenamento, quando um elemento de máscara correspondendo ao elemento de dados da primeira localização de armazenamento estiver ativo.

9. Método, de acordo com a reivindicação 6, em que a substituição dos elementos de dados da segunda localização de armazenamento pelos elementos de dados correspondentes da primeira localização de armazenamento compreende:

a substituição de um elemento de dados da segunda localização de armazenamento por um elemento de dados correspondente da primeira localização de armazenamento, quando um elemento de máscara correspondendo ao elemento de dados da primeira localização de armazenamento estiver ativo.

10. Processador que compreende:

uma localização de armazenamento configurada para armazenar uma primeira pluralidade de elementos de dados, uma segunda pluralidade de elementos de dados, e uma terceira pluralidade de elementos de dados, cada elemento de dados da segunda e da terceira pluralidades de elementos de dados correspondendo a um elemento de dados da primeira pluralidade de elementos de dados;

um decodificador configurado para decodificar uma única instrução que especifica uma operação de comparar-trocar de vetor para a primeira, a segunda e a terceira pluralidades de elementos de dados; e

uma unidade de execução acoplada ao decodificador para receber instruções decodificadas e acoplada à localização de armazenamento para executar a operação de comparar-trocar de vetor;

em que, responsiva à execução da operação de comparar-trocar de vetor, a unidade de execução é configurada para:

comparar os elementos de dados correspondentes da primeira e da segunda pluralidade de elementos de dados; e, responsiva à determinação de que existe uma associação,

substituir os elementos de dados da primeira pluralidade de elementos de dados pelos elementos de dados correspondentes da terceira pluralidade de elementos de dados.

5 11. Processador, de acordo com a reivindicação 10, em que, responsiva à execução da operação de comparar-trocar de vetor, a unidade de execução é adicionalmente configurada para:

10 substituir os elementos de dados da segunda pluralidade de elementos de dados pelos elementos de dados correspondentes da primeira pluralidade de elementos de dados no caso da não existência de uma associação.

15 12. Processador, de acordo com a reivindicação 11, em que a única instrução adicionalmente especifica uma localização de armazenamento de máscara que armazena uma pluralidade de elementos de máscara correspondendo aos respectivos elementos de dados na primeira pluralidade de elementos de dados.

20 13. Processador, de acordo com a reivindicação 12, em que a unidade de execução é configurada para comparar os elementos de dados correspondentes da primeira e da segunda pluralidades de elementos de dados, quando um respectivo elemento de máscara estiver ativo.

25 14. Processador, de acordo com a reivindicação 12, em que a unidade de execução é configurada para substituir os elementos de dados da primeira pluralidade de elementos de dados pelos elementos de dados correspondentes da terceira pluralidade de elementos de dados, quando um respectivo elemento de dados estiver ativo.

30 15. Processador, de acordo com a reivindicação 12, em que a unidade de execução é configurada para substituir os elementos de dados da segunda pluralidade de elementos de dados pelos elementos de dados correspondentes da primeira pluralidade de elementos de dados, quando um respectivo elemento de máscara estiver ativo.

35 16. Processador, de acordo com a reivindicação 12, em que a unidade de execução executa a operação de comparar-trocar de vetor como uma operação atômica.

17. Processador, de acordo com a reivindicação 12, em que responsiva à execução da operação de comparar-trocar de vetor, a unidade de execução é adicionalmente configurada para:

5 ajustar uma bandeira, se existir uma associação entre cada par de elementos de dados correspondentes para os quais um elemento de máscara correspondente está ativo; e

 liberar a bandeira, caso não exista uma associação.

18. Sistema compreendendo:

10 um controlador de memória acoplado a uma primeira localização de armazenamento configurada para armazenar uma primeira pluralidade de elementos de dados; e

 um processador acoplado ao controlador de memória, o processador compreendendo:

15 um arquivo de registro configurado para armazenar uma segunda pluralidade de elementos de dados, e uma terceira pluralidade de elementos de dados, cada da segunda e da terceira pluralidades de elementos de dados correspondendo a um da primeira pluralidade de elementos de dados;

20 um decodificador configurado para decodificar uma única instrução que especifica uma operação de comparar-trocar de vetor para a primeira, a segunda e a terceira pluralidades de elementos de dados; e

25 uma unidade de execução acoplada ao decodificador para receber instruções decodificadas e acoplada à primeira localização de armazenamento e arquivo de registro para executar a operação de comparar-trocar de vetor;

 em que, responsiva à execução da operação de comparar-trocar de vetor, a unidade de execução é configurada para:

30 comparar os elementos de dados correspondentes da primeira e da segunda pluralidades de elementos de dados; e, responsiva à determinação de que existe uma associação,

 substituir os elementos de dados da primeira pluralidade de elementos de dados pelos elementos de dados correspondentes da terceira

pluralidade de elementos de dados; e, responsiva à determinação de que não existe uma associação,

substituir os elementos de dados da segunda pluralidade de elementos de dados pelos elementos de dados correspondentes da primeira pluralidade de elementos de dados.

19. Sistema, de acordo com a reivindicação 18, em que a única instrução adicionalmente especifica um registro de máscara que armazena uma pluralidade de elementos de máscara correspondendo aos respectivos elementos de dados na primeira pluralidade de elementos de dados.

20. Sistema, de acordo com a reivindicação 19, em que a unidade de execução é configurada para comparar pares de elementos de dados correspondentes da primeira e da segunda pluralidades de elementos de dados, quando um respectivo elemento de máscara estiver ativo, para ajustar uma bandeira, se cada comparação resultar em uma associação, e para liberar a bandeira, se cada comparação não resultar em uma associação.

21. Sistema, de acordo com a reivindicação 20, em que a unidade de execução executa a operação de comparar-trocar de vetor como uma operação atômica.

22. Meio legível por computador apresentando, armazenadas no mesmo, instruções operáveis para fazer com que o dispositivo de processador:

decodifique uma única instrução que especifica uma operação de comparar-trocar de vetor para uma pluralidade de elementos de dados, cada elemento de dados apresentando um elemento de teste correspondente, um elemento de substituição e um elemento de máscara;

compare os elementos de dados aos elementos de teste correspondentes, se os respectivos elementos de máscara estiverem ativos; e, responsiva à determinação de que todas as comparações indicam uma associação,

ajuste uma bandeira e substitua os elementos de dados comparados pelos elementos de substituição correspondentes; e, responsiva à determinação de que toda a comparação não indica uma associação,

libere a bandeira e substitua os elementos de teste comparados pelos elementos de dados correspondentes.

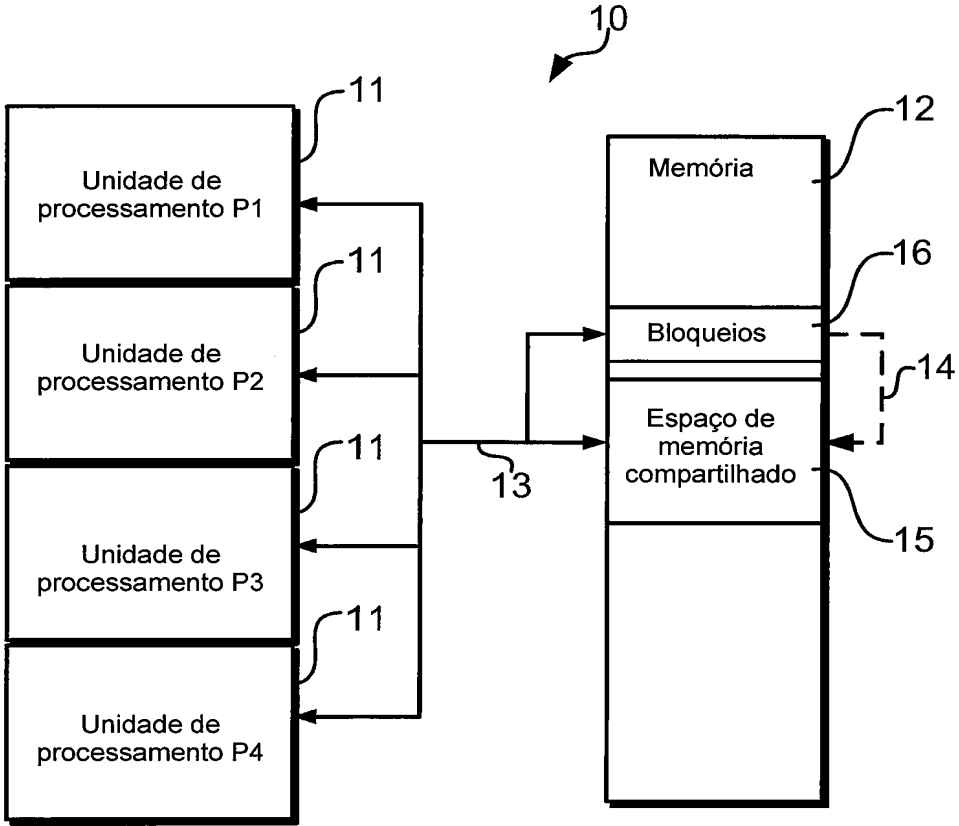


FIG. 1

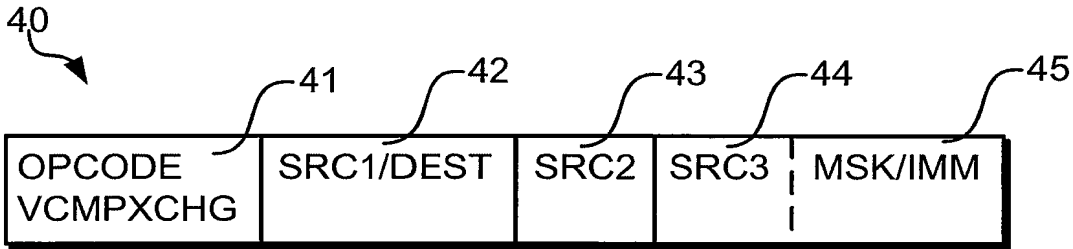


FIG. 3

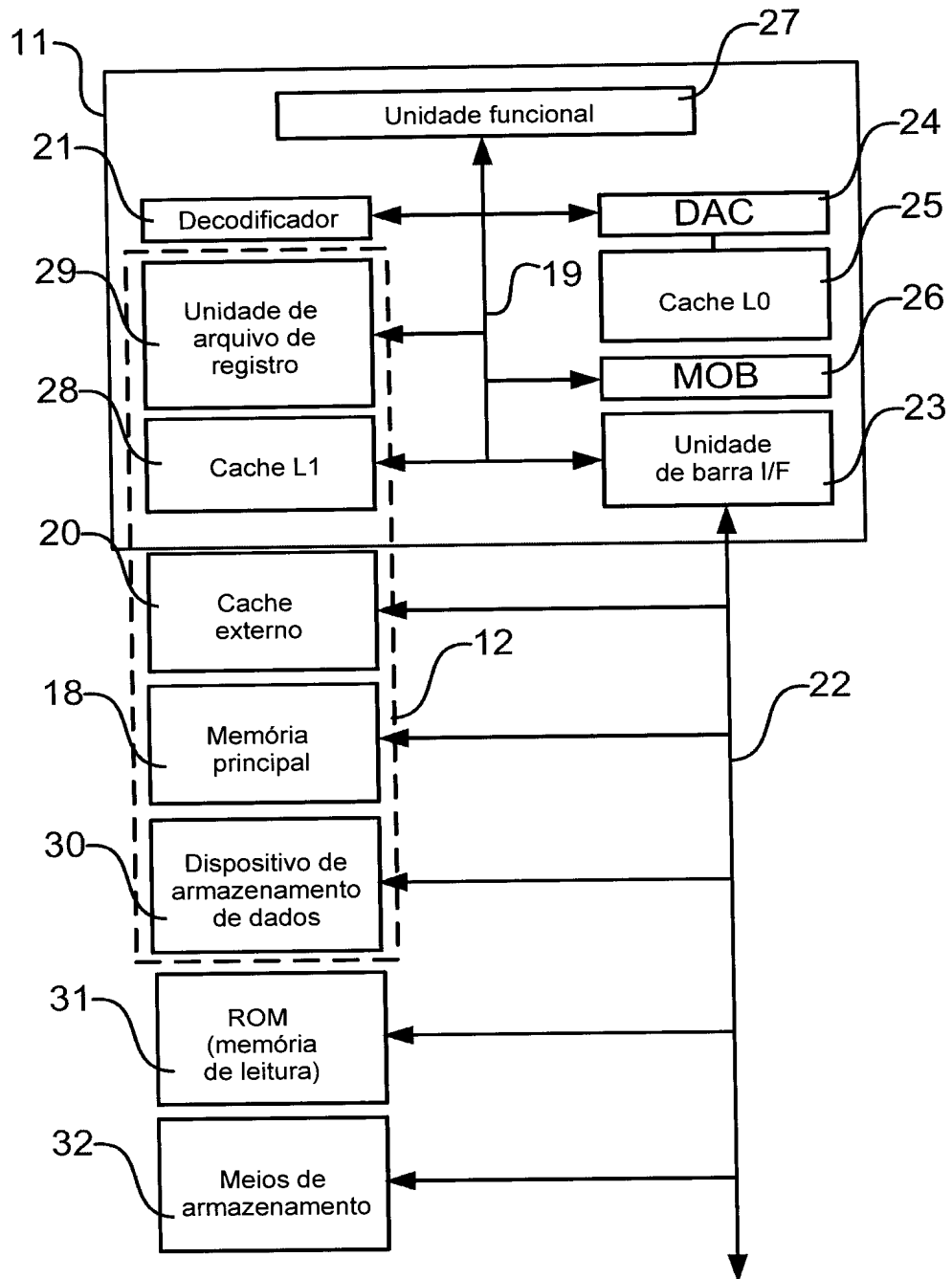


FIG. 2

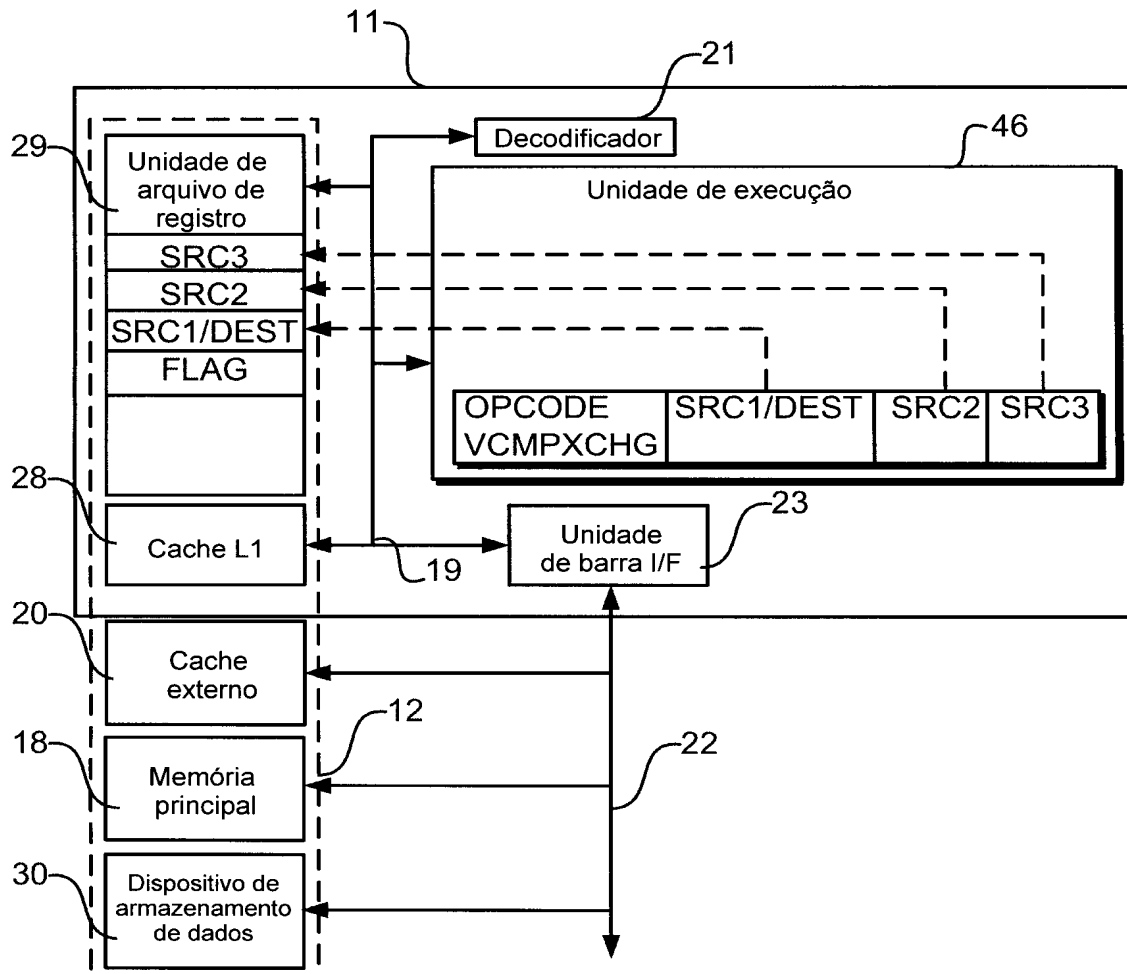


FIG. 4

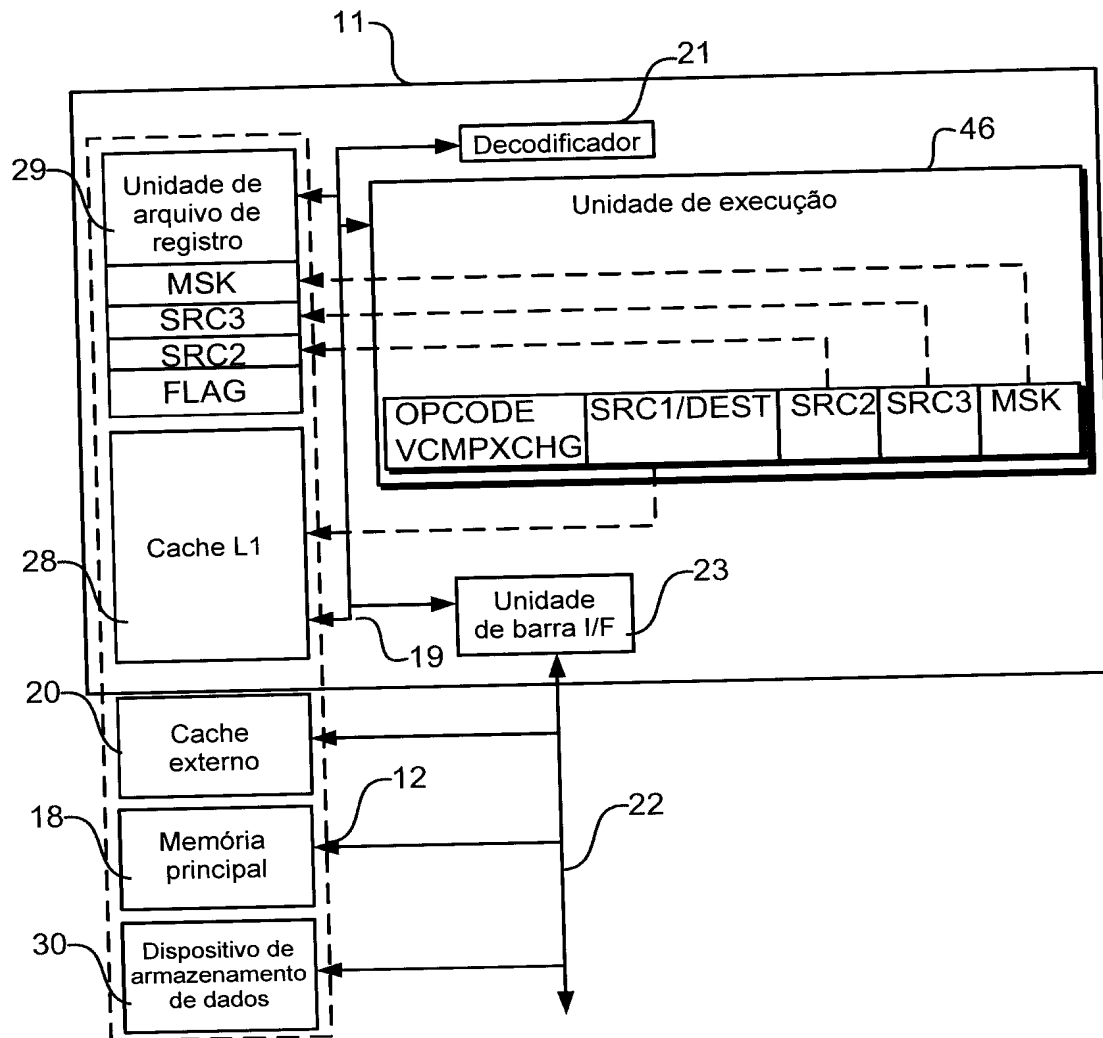
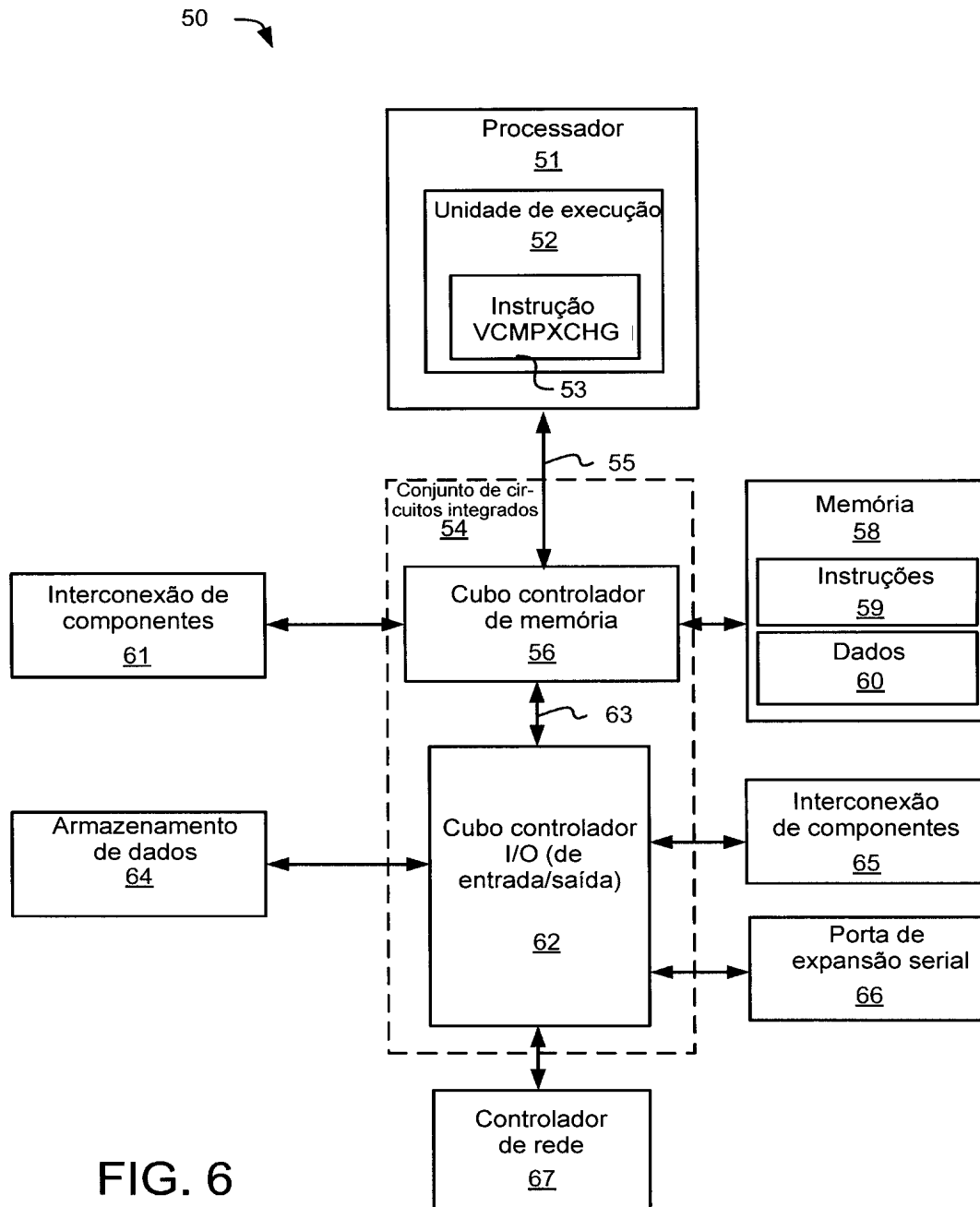


FIG. 5



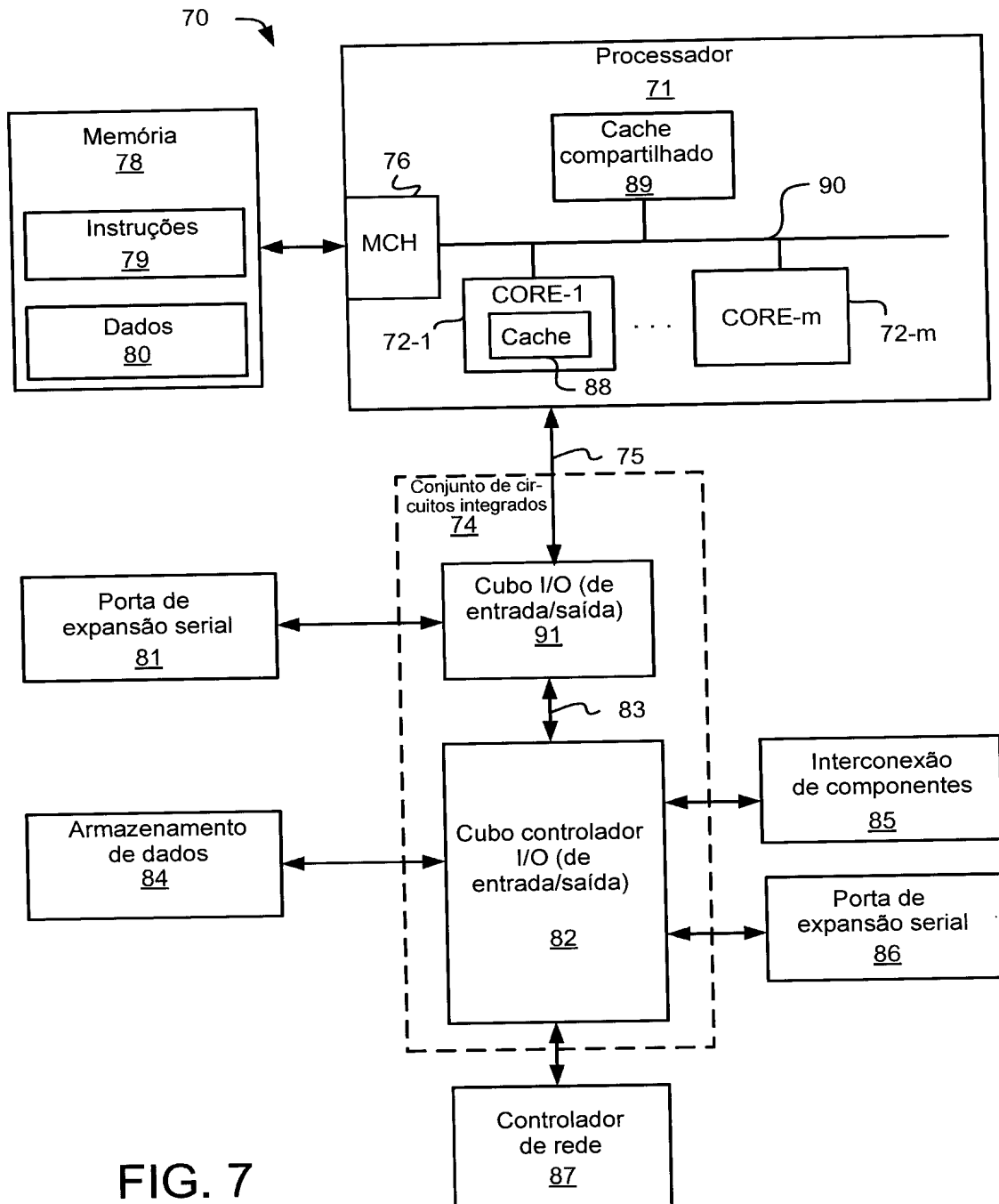


FIG. 7

RESUMO

Patente de Invenção: **"VETORES SIMD SINCRONIZADORES"**.

A presente invenção refere-se a uma operação de comparar-trocar de vetor por meio da: decodificação por um decodificador em um dispositivo de processamento de uma única instrução que especifica uma operação de comparar-trocar de vetor para uma pluralidade de elementos de dados entre uma primeira localização de armazenamento, uma segunda localização de armazenamento, e uma terceira localização de armazenamento; emissão da única instrução para execução por uma unidade de execução no dispositivo de processamento; e, responsiva à execução da única instrução, comparando elementos de dados da primeira localização de armazenamento com os elementos de dados correspondentes na segunda localização de armazenamento; e, responsiva à determinação de que existe uma associação, substituindo elementos de dados da primeira localização de armazenamento pelos elementos de dados correspondentes da terceira localização de armazenamento.