



República Federativa do Brasil
Ministério da Economia
Instituto Nacional da Propriedade Industrial

(11) PI 0409710-6 B1



(22) Data do Depósito: 14/04/2004

(45) Data de Concessão: 14/01/2020

(54) Título: MÉTODO E LÓGICA CONTÁBIL PARA DETERMINAR A UTILIZAÇÃO DE RECURSO DE PROCESSADOR PER-THREAD EM UM PROCESSADOR DE MÚLTIPLOS THREADS SIMULTÂNEOS (SMT)

(51) Int.Cl.: G06F 11/34; G06F 9/38.

(30) Prioridade Unionista: 23/04/2003 US 10/422,025.

(73) Titular(es): INTERNATIONAL BUSINESS MACHINES CORPORATION.

(72) Inventor(es): WILLIAM JOSEPH ARMSTRONG; MICHAEL STEPHEN FLOYD; LARRY SCOTT LEITNER; RONALD NICK KALLA; BALARAM SINHARROY.

(86) Pedido PCT: PCT GB2004001586 de 14/04/2004

(87) Publicação PCT: WO 2004/095282 de 04/11/2004

(85) Data do Início da Fase Nacional: 24/10/2005

(57) Resumo: "MÉTODO E LÓGICA CONTÁBIL PARA DETERMINAR A UTILIZAÇÃO DE RECURSO DE PROCESSADOR PER-THREAD EM UM PROCESSADOR DE MÚLTIPLOS THREADS SIMULTÂNEOS (SMT)". Um método contábil e lógica para determinar utilização de recurso de processador por-thread em um processador de múltiplos threads (SMT) provê um mecanismo para contabilizar o uso do recurso de processador por programas e threads dentro de programas. O uso de recurso relativo é determinado pela detecção de despacho de instruções para múltiplos thread ativos dentro do processador, que podem incluir threads ociosos que ainda estão ocupando recursos do processador. Se instruções são despachadas para todos os threads ou nenhum thread, o ciclo de processador é contabilizado igualmente para todos os threads. Alternativamente, se nenhum thread está no estado de despacho, a contabilização pode ser feita usando um estado anterior ou em conformidade com taxas de níveis de prioridade de threads. Se apenas um thread é despachado, aquele thread é contabilizado pelo ciclo de processador completo. Se múltiplos threads são despachados, mas menos que todos os threads são despachados (em um processador suportando mais que dois threads), o ciclo de processador é faturado uniformemente pelos threads despachados. Múltiplos threads podem ser detectados para os threads e um (...).

Relatório Descritivo da Patente de Invenção para: "MÉTODO E LÓGICA CONTÁBIL PARA DETERMINAR A UTILIZAÇÃO DE RECURSO DE PROCESSADOR PER-THREAD EM UM PROCESSADOR DE MÚLTIPLOS THREADS SIMULTÂNEOS (SMT)".

5 Antecedentes da Invenção

Campo Técnico

A presente invenção se relaciona, em geral, a processadores e sistemas de computação, e mais particularmente, a um processador de múltiplos threads
10 simultâneos (SMT). A presente invenção relaciona-se também aos sistemas contábeis de uso do processador.

Descrição do Estado da Técnica

Os processadores dos dias atuais de alta velocidade incluem a capacidade de execução simultânea de instruções, execução especulativa e o carregamento de instruções e
15 operação simultânea de vários recursos dentro de um processador. Particularmente, encontrou-se desejável gerenciar a execução de um ou mais threads dentro de um processador, de modo que mais de um thread de execução
20 possa usar o processador e de modo que os recursos sejam mais eficazmente usados do que eles são tipicamente, por um único thread.

Os projetos de processador anteriores trataram do problema de controlar múltiplos threads através de um

switch de estado de hardware da uma execução de um thread até a execução de um outro thread. Tais processadores são conhecidos como os processadores multithread de um switch de hardware entre a execução de um ou outro thread. Um
5 processador de HMT supera as limitações de esperar um thread bloqueado, permitindo que o hardware comute a execução para um outro thread. A execução de ambos os threads pode ser feita não simultaneamente, mas por meio da alocação de partes de execução para cada thread quando cada
10 thread passa por uma condição de bloqueio, tal como uma falta de cache.

Os processadores multithread simultâneos (SMT) fornecem um uso ainda mais eficiente de recursos do processador, à medida que os múltiplos threads podem
15 simultaneamente usar recursos do processador. Os múltiplos threads são executados simultaneamente em um processador de SMT de modo que as múltiplas unidades de execução do processador, tais como unidades de ponto flutuante, as unidades de instrução de ponto fixo, as unidades de
20 carregamento / armazenamento e outras possam executar tarefas para um (ou mais dependendo das capacidades das unidades de execução) dos múltiplos threads simultaneamente. Os recursos de armazenamento e registo

P10409710

22

também podem ser alocados com base por-thread de modo que a permuta do estado interno completa do HMT seja evitada.

A contabilidade do uso de tempo do processador é necessária para a administração de vendas de serviços de
5 computação, bem como para o gerenciamento de contabilidade de custo interno, quando, para o exemplo, algumas rodadas de processadores são para as atividades de pesquisa e desenvolvimento que permitem que o hardware seja capitalizado em uma maneira diferente para finalidades de
10 impostos do que outros usos. Um servidor pode ser particionado e o tempo de processador dito aos múltiplos usuários baseado em "em demanda" ou "à medida que usado". Adicionalmente, o tempo de processador pode ser utilizado por proprietários ou arrendadores de hardwares e também
15 subcontractado para entidades que pagam por serviços. Conseqüentemente, a contabilidade exata do tempo de execução do processador é uma necessidade nos modelos de software e de arquitetura computacional. Também, o uso de tempo do processador, especialmente em uma base por-thread,
20 é útil para perfilar programas para otimização.

Em sistemas de processamento de thread único, a contabilidade é geralmente direta. Uma contagem do uso de ciclo de processador ou mesmo a medida do tempo de "wall clock" pode ser fornecida para funcionamentos de trabalho

completos, como se múltiplos threads dentro de programas múltiplos fossem executados, eles não seriam executados simultaneamente, mas seqüencialmente. Um registro de tempos de ciclo está mantido até que um trabalho esteja
5 completo e o total estiver apresentado para finalidades de contabilidade. O tempo medido correlaciona diretamente à utilização do recurso do processador.

Em uma máquina de HMT, a tarefa é similar pelo fato que, dentro do hardware do processador, apenas um thread é
10 executa por vez e os contadores múltiplos são usados para rastrear o uso do tempo de processador de cada thread, quando, rodando um contador somente enquanto seu thread associado estiver ativo. Entretanto, um processador de SMT, dois ou mais thread podem ser, simultaneamente,
15 executados dentro de um único núcleo processador e o uso dos recursos por cada thread não é determinado facilmente por uma contagem de execução simples ou medida de tempo.

É, conseqüentemente, desejável fornecer um método e um aparelho que possam contabilizar o uso do tempo de
20 processador em um processador de SMT. É mais desejável fornecer um método para a contabilidade do uso de recurso dentro de um uso do processador de SMT entre os threads que executam dentro de tal processador.

Sumário da Invenção

O objetivo de contabilizar o uso do tempo de processador dentro de um processador de múltiplos threads (SMT) simultâneos é obtido em um processador que tem contadores de utilização do processador e um método para a

5 contabilizar o uso do tempo de processador.

O processador incorpora contadores de ciclo múltiplos do uso do processador, um associado com cada thread que está executando dentro de um processador para fornecer uma contagem que corresponde ao uso do recurso de processador

10 por cada thread. O uso relativo do recurso é a detecção no com base por-thread e é usado atualizar os contadores de ciclo. A detecção do uso de recurso pode ser realizada por meio da detecção da presença de um estado de ciclo particular que indica a instrução ativa que processa para

15 um thread. O estado do ciclo é detectado pela amostragem periódica, que pode ser a cada ciclo de clock, e os contadores são incrementados (ou decrescidos alternativamente) em conformidade com o estado de ciclo detectado. O estado detectado é escolhido para ser

20 indicativo do uso do recurso de processador relativo por cada thread.

O estado de ciclo detectado pode ser uma expedição de uma instrução para thread. Os ciclos para os quais nenhum thread está no estado de ciclo particular podem ser

P10409710

25

carregados uniformemente para cada thread, ou o ciclo pode ser carregado em conformidade com um último estado da expedição para todas os thread. Alternativamente, ciclos não indicativos podem ser carregados em conformidade com uma prioridade de thread ordenada, que dá uma indicação do uso relativo do recurso de thread. Ciclos, nos quais um thread está no estado de ciclo particular e o outro thread não está, podem ser atribuídos inteiramente ao thread no estado de ciclo particular, fornecendo, assim, atribuição para threads ativos diante de threads ociosos. Ademais, os ciclos podem ser atribuídos de acordo com os valores fracionários que indicam o número de recursos usados em um dado ciclo por cada thread.

O processador pode ser capaz de ambos modos de operação de múltiplos thread simultâneos (SMT) e thread único (ST), e a contabilidade pode ser seletivamente responsiva a um modo selecionado, atribuindo todos os ciclos a um único thread que executa no modo ST.

A publicação de patente norte-americana US 2001/0056456 divulga uma arquitetura simultânea baseada em prioridade combinando informação de prioridade de sistema operacional com heurística de thread a fim de fornecer prioridades dinâmicas para selecionar instruções de thread para processamento.

P 10409710

384

Os objetivos anteriores e outros, as características, e as vantagens da invenção serão aparentes da seguinte, mais particular, descrição da modalidade preferida da invenção, como ilustrado nos desenhos anexos.

5 Descrição Breve dos Desenhos

As novas características, consideradas como caracteriza da invenção, estão definidas nas reivindicações anexas. A própria invenção, entretanto, assim como uma modalidade preferida de uso, objetivos adicionais, e
10 vantagens da mesma, serão melhor compreendidos por referência à seguinte descrição detalhada de uma modalidade ilustrativa, quando lida conjuntamente com os desenhos anexos, nos quais números de referência semelhantes indicam componentes análogos.

15 A figura 1 é um diagrama de blocos de um sistema de acordo com uma modalidade da invenção.

A figura 2 é um diagrama de blocos de um núcleo processador de acordo com uma modalidade da invenção.

A figura 3 é um diagrama de blocos de detalhes das
20 unidades funcionais dentro do núcleo processador 10 da figura 2.

A figura 4 é um fluxograma que descreve um método de acordo com uma modalidade da invenção atual.

Descrição da Modalidade ilustrativa

P 10409710

38

Com referência agora às figuras, e particularmente com referência à figura 1, está representado um diagrama de blocos de um sistema de acordo com uma modalidade da invenção atual. O sistema inclui um processador de grupo 5 que pode ser conectado a outros processadores de grupo através de uma ponte 37 que forma um processador super-escalar. O processador de grupo 5 é conectado a uma unidade de cache L3, memória local de sistema 36 e vários periféricos 34, bem como a dois processadores de serviço 34A e 34B. Os processadores de serviço fornecem a supervisão da falha, assistência de inicialização e teste de capacidade do processador de grupo 5 e podem ter suas próprias vias de interconexão para outros processadores de grupo, bem como conectar todos os processadores 30A-D.

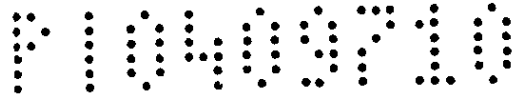
Dentro do processador de grupo, está uma pluralidade de processadores 30A-D, fabricado geralmente em uma única unidade e que incluem uma pluralidade de núcleos 10A e 10B processadores acoplados a uma cache L2 32 e um controlador de memória 4. Os núcleos 10A e 10B fornecem a execução de instruções e operações em valores de dados para funções de processamento de propósito geral. A ponte 37, bem como as outras pontes dentro do sistema fornecem comunicação através de barramentos *wides* com outros processadores de grupo e o barramento 35 fornece a conexão dos processadores

P 10409710

240

30A-D, a ponte 37, os periféricos 34, a cache L3 36 e a memória local do sistema 38. A outra memória de sistema global pode ser acoplada externa a ponte 37 para o acesso simétrico por todos os processadores de grupo.

5 Os núcleos processadores 10A e 10B são processadores de múltiplos threads simultâneos (SMT) capazes de execução simultânea de múltiplos threads. Os núcleos processadores 10A e 10B ainda suportam um único modo de operação de único thread para a execução eficiente de um único thread, quando
10 as condições de execução de programa impõem operação de thread único, por exemplo, quando a execução de programa de alta prioridade deve ser completada por um tempo conhecido, ou quando um thread em um processador de múltiplos threads é conhecido como sendo inativa. Os múltiplos threads
15 introduzem algumas ineficiências na execução completa de um único thread, mas no total, existe uma vantagem de eficiência de sistema como threads são freqüentemente espera ociosa em outras tarefas para completar. Conseqüentemente, a transição entre os modos de múltiplos
20 threads e de único thread fornece uma vantagem em adaptação de uma ou mais das condições acima-descritas, e as concretizações da presente invenção fornecem a contabilidade por tempo de processador em uma maneira



consistente com um processador que forneça a contabilidade de tempo do processador responsivo a tais transições.

Consultando agora à figura 2, os detalhes de um núcleo processador 10 que tem as características idênticas aos 5 núcleos processadores 10A e 10B são descritos. Uma unidade de interface de barramento 23 conecta o núcleo processador 10 para outros processadores de SMT e a periféricos e conecta a cache D L1 22 para armazenar valores de dados, a cache I L1 20 para armazenar as instruções de programa e a 10 unidade de interface de cache 21 à memória externa, processador e outros dispositivos. A cache I L1 20 fornece o carregamento de fluxos de instrução conjuntamente com a unidade de busca de instrução IFU 16, que pré-busca instruções e pode incluir o carregamento especulativo e 15 capacidade de predição de ramo. Uma unidade de seqüenciador de instrução (ISU) 12 controla a seqüência de instruções emitidas para as várias unidades internas, tais como uma unidade de ponto fixo (FXU) 14 para executar operações gerais e uma unidade de ponto flutuante (FPU) 15 20 para executar operações de ponto flutuando. Tabelas de complexão global (GCT) 13 rastreiam as instruções emitidas pela ISU 12 através de tags até que a unidade de execução particular objetivada pela instrução indique que as instruções terminaram a execução.

P 10409710

A unidade de ponto fixo 14 e a unidade de ponto flutuante 15 são acopladas a vários recursos, tais como os registradores de propósito geral (GPR) 18A, registradores de ponto flutuante (FPR) 18B, registradores de condição (CR) 18C, buffers de renomeação 18D, registradores de conta / registradores de ligação (CTR/LR) 18E e os registradores de exceção (XER) 18F. O GPR 18A e o FPR 18B fornecem o armazenamento de valor de dados para valores de dados carregados e armazenados a partir de cache D L1 22 por meio da unidade de armazenamento de carga (LSU) 19. O CR 18 C armazena informação de ramificação condicional e os buffers de renomeação 18D (que podem compreender algumas unidades de renomeação associadas com as várias unidades internas de execução) fornecem o armazenamento do operando e do resultado para as unidades de execução. O XER 18F armazena informação de exceção de ponto fixo e ramo e CTR/LR 18E armazena informação de ligação de ramificação e a informação de contagem para a execução do ramo do programa. GPR 18A, FPR 18B, CR 18C, buffers de renomeação 18D, CTR/LR 18E e XER 18F são os recursos que incluem alguns registradores fixos (arquitetados) que armazenam a informação durante a execução de um programa e devem ser fornecido como um conjunto fixo para cada thread em execução, outros registradores não arquitetados dentro dos

P10409710

Lu

recursos acima estão livre para uso de renomeação. A lógica de controle 11 é acoplada a várias unidades e recursos da execução dentro do núcleo processador 10, e é usado fornecer o controle pervasivo das unidades de execução e recursos. A unidade de interface SCOM/XSCOM 25 fornece uma conexão para os processadores de serviço externos 34A-B.

Fazendo referência agora à figura 3, detalhes de blocos funcionais dentro do núcleo processador 10 são descritos. A ISU 12 indica para a lógica de controle 11 quando as instruções são despachadas para threads particulares em execução com núcleo processador 10. A lógica de controle 11 atualiza os contadores de meio-ciclo 42A e 42B seletivamente dependendo de se as instruções foram despachadas pela ISU 12 para um thread associado com um dado contador (thread 0 para o contador de meio-ciclo 42A e thread 1 para o contador de meio-ciclo 42B) em um dado ciclo como detectado por um circuito de detecção de expedição 44. As saídas dos contadores de meio ciclo 42A e 42B são travadas por latches 43A e 43B que são acoplados ao contador de ciclo de base-tempo 47. No oitavo ciclo de processador do contador de ciclo de base-tempo 47, a saída dos latches 43A e 43B são fornecidas para incrementar os contadores de uso de thread 41A e/ou 41B, fornecendo uma

P 10409710

10

atualização dos contadores de uso de thread 41A e/ou 41B todo oitavo ciclo de processador que o contador de ciclo de base-tempo 47 está ativo (que impossibilitará operações especiais onde a base de tempo estiver inativa para ser atribuída para ativar threads). O contador de ciclo de base-tempo 47 reseta os latches 43A e 43B, após o incremento. A ação acima descrita fornece contagens nas saídas dos contadores de uso de threads 41A e/ou 41B que são frações do contador de base tempo para o processador e podem ser usados diretamente para computar o uso relativo do recurso de thread em um valor de "tempo usado".

De acordo com concretizações alternativas da invenção, outros estados de ciclo particulares podem ser usados pela lógica de controle 11 e ISU 12 (ou outra unidade de execução dentro do núcleo processador 10) para disparar uma atualização do contador de meio-ciclo 42A e/ou 42B em cada ciclo, mas a expedição da instrução é escolhida na ilustração exemplificativa da invenção, porque a expedição da instrução fornece um mecanismo para medir o começo de cada atividade de processamento usando o núcleo processador 10, em vez do progresso ou da conclusão de tais atividades, que são dependentes da eficiência do código. Por exemplo, um programa codificado de forma ineficiente pode gerar muitas predições de ramificação incorretas e conseqüentes

P10409710

pré-buscas e flushes, usando recursos críticos do processador ao gerar poucas conclusões. Conseqüentemente, o uso de conclusões de instrução ao invés de expedições de instrução geraria uma medida de uso de tempo de processador mais baixa para o thread em execução de código ineficiente. Conseqüentemente, o uso da expedição de instrução como o evento de disparo particular para atualizar os contadores de meio-ciclo 41A e 41B fornece uma estimativa do uso do recurso do processador que não recompensa código ineficiente.

Quando os modelos prévios de contabilidade foram capazes de alocar essencialmente todo o tempo do núcleo processador 10 para um thread particular, devido à falta de execução simultânea de thread em um dado processador, em um processador de SMT, o "faturamento" do uso de tempo do processador em base por-thread deve ser distribuído através de dois ou mais threads que podem estar simultaneamente em execução dentro do núcleo processador 10. Quando a ISU 12 não está despachando as instruções para qualquer thread que usa o núcleo processador 10, ela ainda precisa contabilizar o uso do núcleo processador 10 por todas os threads. Os thread podem não estar começar um novo trabalho na forma de instruções despachadas, mas ainda se ligando ao núcleo processador 10, ou podem ser ativos nas

P10409710

44

fases de execução ao invés de expedindo instrução. Por exemplo, um thread pode estar causando longas pré-buscas e um outro thread pode estar executando uma operação de ponto flutuante longa. Os recursos do processador e as unidades de execução estão sendo usados ativamente por ambos os threads, enquanto somente um pequeno número de conclusões de instrução seria associado com os intervalos de processamento de acordo com o exemplo acima. Um outro estado particular de instrução que é preferível às conclusões são as buscas de instrução, que forneceria uma indicação similar àquela da expedição da instrução, mas fornecendo contagens do uso inclinadas mais para o uso de IFU 16 e cache I L1 20, ao invés do uso de unidades de execução incluindo FXU 14, FPU 15 e seus recursos associados.

Conseqüentemente, a modalidade exemplificativa aqui descrita fornece um mecanismo para a contabilidade de todos os ciclos do núcleo processador baseada em uma estimativa da unidade de execução e do uso de recurso, mesmo quando nenhuma expedição ocorre em um dado ciclo (ou em concretizações alternativas, outros estados de ciclo particulares ou em usos de recurso detectados pela lógica de controle 11). Nos ciclos onde nenhuma expedição de instrução ocorre para qualquer thread, o ciclo é

P10409710

contabilizado como uma divisão igual do valor do ciclo para cada thread que executa dentro do núcleo processador 10 (incluindo thread ociosos que não foram descarregados do núcleo processador 10 e que estão ainda conectando os recursos do processador). Mas, em uma modalidade alternativa ou em um comportamento selecionável da lógica de controle 11, a lógica de controle 11 pode recordar o último estado de expedição e pode usar isso para atualizar, preferivelmente, o contador de meio-ciclo 42A e/ou 42B ao invés. Uma outra alternativa é para a lógica de controle 11 para usar o conjunto de prioridades de threads relativas no núcleo processador 10 para dividir em partes o valor do ciclo de clock usado para atualizar o contador de meio-ciclo 42A e/ou 42B a fim de alocar corretamente um ciclo em que nenhuma expedição ocorre para qualquer linha.

Similarmente, quando as instruções são despachadas em um dado ciclo para mais de um thread, o ciclo é contabilizado como uma divisão igual do valor de ciclo de cada thread para os quais instruções foram despachadas. Ou mais uma vez, a lógica de controle 11 pode usar o conjunto de prioridades de threads relativas no núcleo processador 10 para dividir em partes o valor do ciclo de clock usado para atualizar o contador de meio-ciclo 42A e 42B de forma

a alocar adequadamente um ciclo no qual a expedição ocorre para todos os threads.

Conseqüentemente, na metodologia de divisão idêntica (e não no esquema de ciclo anterior ou prioridade listado como alternativas acima), em um núcleo processador que 5 suporte a execução de um ou dois threads, um valor de um meio é contabilizado para cada thread para ciclos em que nenhuma expedição ocorre ou nos ciclos em que uma expedição de instrução ocorre para cada thread. Nos ciclos onde uma 10 expedição de instrução ocorre para somente um thread, o thread para o qual a expedição de instrução ocorre é faturado para o ciclo inteiro. A lógica de controle 11 detecta um dos quatro estados potenciais: nenhuns thread sendo despachado, thread 0 sendo despachado, thread 1 sendo 15 despachado, ou ambos threads sendo despachados, e atualiza o contador de meio ciclo de thread 0 42A e o contador de meio ciclo de thread 0 42B em conformidade com a distribuição de faturamento de ciclo acima descrita.

Uma vez que o valor mais baixo de distribuição 20 descrito acima é para uma alocação de incrementos de ciclo de um-meio, os contadores de meio-ciclo 42A e 42B são atualizados por um incremento (ou por decremento) de 1 para o meio-ciclo ou de 2 para a alocação de ciclo completo. Na prática, cada um dos contadores de uso de thread 41A e 41B

P 10409710

são atualizados em um oitavo da taxa de expedição de instrução possível para a área do chip, potência e razões de sincronismo (ou devido à taxa do contador de ciclo de base tempo 47, que é escolhida como 1/8 do tempo de ciclo do processador para as mesmas razões), assim os contadores de meio-ciclo de 4 bits 42A e 42B que não fazem parte dos contadores de uso de thread "arquitetados" thread fornecem uma contagem eficaz de uso de 8 ciclos por excesso de fluxo (overflow) (uma vez que um valor dos dois será alocado para o incremento total dos contadores de meio-ciclo 42A e 42B). Os contadores de meio-ciclo 42A e 42B excedem sua capacidade para a porção de contadores de uso de thread 41A e 41B que são lidos pela unidade de ponto fixo 14, quando uma rotina de contabilidade (programa) recupera a informação de uso de tempo de ciclo do processador. Os contadores de meio-ciclo 42A e 42B são implementados como somadores que adicionam duas sub-contas para uma alocação de ciclo completa ou uma sub-conta para meia alocação de ciclo.

Em uma modalidade alternativa da presente invenção, a unidade de uso de tempo de ciclo do processador 40 usa uma adição para incrementar (ou subtração para decrementar) os contadores de uso de thread 41A e 42. O valor adicionado é determinado pelo circuito de detecção de expedição 44 que

detecta que as expedições múltiplas estão sendo feitas para um dado thread, e o sub-contador associado é incrementado (tem adicionado a ele) por um valor proporcional à relação do número de instruções despachadas para o thread com o número total de instruções despachadas para todas os threads para o ciclo. Por exemplo, quando o núcleo processador 10 está executando dois threads e em um dado ciclo, três instruções são despachadas para um primeiro thread e uma instrução é despachada para o segundo thread, um valor de $3/4$ é alocado ao contador de uso de thread para o primeiro thread e $1/4$ é alocado para o contador de uso de thread para o segundo thread. Se nenhuma expedição for feita em um dado ciclo, o ciclo ainda está alocado uniformemente entre os threads.

Consultando agora a figura 4, um método de acordo com uma modalidade da presente invenção é descrito em um fluxograma. O método descrito é para a contabilidade dentro de um processador de SMT que tem um modo de SMT e um modo de ST e é capaz de executar simultaneamente múltiplos threads na modalidade de SMT. Se o processador estiver no modo de ST (decisão 50), o contador de ciclo de thread para o thread em execução é atualizado por 1 valor do ciclo todo ciclo (etapa 51). Se o processador estiver na modalidade de SMT (decisão 50), se somente um thread for despachado

P 10409710

14

(decisão 52), o contador de ciclo de thread associado é atualizado por 1 valor de ciclo (etapa 53). Se todas os threads são despachados (decisão 54), os contadores de ciclo de thread para todos os threads são atualizados igualmente de acordo com uma fração de um ciclo determinado pelo número de threads ou de acordo com o número das expedições emitidas (ou dos recursos usados) para cada thread (etapa 55). Se nenhum thread for despachado, os contadores de ciclo de thread podem ser atualizados igualmente, ou em proporção à última amostragem da expedição, ou em proporção à sua prioridade (etapa 57). Finalmente, se nenhum thread for despachado, os contadores de ciclo de thread podem, do mesmo modo, ser atualizados igualmente, ou em proporção à última amostragem da expedição, ou em proporção a sua prioridade (etapa 58).

Enquanto a invenção foi particularmente mostrada e descrita em referência à modalidade preferida da mesma, será compreendido que aqueles hábeis na arte que as mudanças antecedentes e outras na forma, e detalhes podem ser feitos nisso sem fugir do espírito e do escopo da invenção.

REIVINDICAÇÕES

1. Método de contabilidade de uso de tempo de processador em um processador (10) **caracterizado por** compreender:

em intervalos periódicos, determinar incrementos a serem aplicados aos circuitos de contador de uso de recursos dentro do núcleo do processador (10A, 10B) de acordo com valores de uso de recursos internos do núcleo do processador (10A, 10B) por uma pluralidade de threads em execução dentro do processador (10), cada um dos threads correspondendo a um dos circuitos de contador de uso de recursos;

no fim do intervalo periódico, incrementar os circuitos de contador de uso de recursos pelos incrementos determinados pela determinação, em que os circuitos de contador são incrementados por valores diferentes de acordo com diferenças nos valores de uso dos recursos internos do núcleo de processador (10A, 10B) por seus threads correspondentes;

repartir o uso de tempo do processador através de um ciclo de base temporal (47) em respectivos threads da pluralidade de threads em proporção aos valores contidos nos circuitos de contador de uso de recursos nos finais dos intervalos periódicos, em que threads da pluralidade de threads que usam mais dos recursos são contabilizados um valor maior do uso de tempo de processador.

2. Método, de acordo com a reivindicação 1, **caracterizado pelo** fato de que o intervalo periódico é cada ciclo de clock do núcleo do processador (10A, 10B), em que o método ainda compreende determinar se cada thread está ou não em um estado de ciclo particular durante um determinado ciclo de clock, e em que os incrementos da determinação determinam os incrementos em conformidade com um resultado da detecção se cada thread está ou não em um estado de ciclo particular.

3. Método, de acordo com a reivindicação 2, **caracterizado pelo** fato de que os incrementos de determinação ainda determinam valores dos incrementos a partir de um número de recursos para cada thread que é detectado estando em um estado de ciclo particular.

4. Método, de acordo com a reivindicação 2, **caracterizado pelo** fato de que o estado de ciclo particular é um estado indicando uma expedição de instrução para um thread.

5. Método, de acordo com a reivindicação 2, **caracterizado pelo** fato de que o estado de ciclo particular é um estado indicando um término de instrução para um thread.

6. Método, de acordo com a reivindicação 2, **caracterizado pelo** fato de que em resposta à detecção ter detectado que uma ou mais da pluralidade de threads está em

um estado de ciclo particular, os incrementos de determinação atribuem incrementos iguais a serem aplicados aos circuitos de contador de uso de recurso correspondendo a um ou mais threads.

7. Método, de acordo com a reivindicação 2, **caracterizado pelo** fato de que o incremento incrementa cada um dos circuitos de contador de uso de recursos igualmente em resposta à detecção ter detectado que nenhuma da pluralidade de threads está no estado de ciclo particular.

8. Método, de acordo com a reivindicação 2, **caracterizado pelo** fato de que os incrementos de determinação determinam os incrementos de acordo com números fracionários de recursos internos usados pelos threads correspondentes a partir de um anterior dos intervalos periódicos, em resposta à detecção ter detectado que nenhuma da pluralidade de threads está no estado de ciclo particular.

9. Método, de acordo com a reivindicação 2, **caracterizado pelo** fato de que os incrementos de determinação determinam os incrementos em conformidade com um nível de prioridade de seus threads correspondentes, em resposta à detecção ter detectado que nenhuma da pluralidade de threads está no estado de ciclo particular.

10. Núcleo de processador suportando a execução simultânea de uma pluralidade de threads, o núcleo de processador (10A, 10B) **caracterizado por** compreender:

uma pluralidade de recursos internos usados pela pluralidade de threads e integrados dentro do núcleo do processador (10A, 10B);

uma unidade de controle de instrução (12) acoplada à pluralidade de recursos internos, em que a unidade de controle de instrução (12) controla um número dos recursos internos usados pelo processador (10) em intervalos periódicos;

uma pluralidade de circuitos de contador de uso de recursos integrados dentro do núcleo do processador (10A, 10B), cada um dos circuitos de contador de uso de recursos correspondendo a uma particular dentre a pluralidade de threads;

um contador de base temporal (47) para gerar uma base temporal de ciclo de medição de uso do processador; e

um circuito lógico (11) acoplado à unidade de controle de instrução (12), o contador de base temporal (47), e os circuitos de contador de uso de recursos, em que o circuito lógico (11) atualiza cada um dos circuitos de contador de uso de recursos de acordo com incrementos diferentes determinados a partir de um valor de uso de recursos internos

de núcleo de processador pelos threads correspondentes, em que o uso de tempo de processador relativo dentre a pluralidade de threads através de um ciclo do contador de base temporal (47) é representado em valores contidos nos circuitos de contador de uso de recursos, em que threads dentre a pluralidade de threads que usam mais da pluralidade de recursos internos são contabilizadas com um valor maior do uso de tempo de processador.

11. Núcleo de processador, de acordo com a reivindicação 10, **caracterizado pelo** fato de que os intervalos periódicos são ciclos de clock individuais do núcleo de processador (10A, 10B), e em que o circuito lógico (11) determina se cada thread está ou não em um estado do ciclo particular.

12. Núcleo de processador, de acordo com a reivindicação 11, **caracterizado pelo** fato de que a lógica de controle (11) ainda determina um número de recursos internos usados por cada thread que é determinado como sendo no estado de ciclo particular.

13. Núcleo de processador, de acordo com a reivindicação 12, **caracterizado pelo** fato de que a lógica de controle (11) incrementa de forma fracionária cada um dos recursos circuitos de contador de uso de recursos em

conformidade com o número determinado de recursos internos usados pelos seus threads correspondentes.

14. Núcleo de processador, de acordo com a reivindicação 11, **caracterizado pelo** fato de que o estado de ciclo particular é um estado indicando uma expedição de instrução para um thread.

15. Núcleo de processador, de acordo com a reivindicação 11, **caracterizado pelo** fato de que a lógica de controle (11), em resposta a determinação de que uma ou mais da pluralidade de threads estão no estado de ciclo particular, incrementa igualmente os circuitos de contador de uso de recursos que correspondem com uma ou mais dos threads.

16. Núcleo de processador, de acordo com a reivindicação 11, **caracterizado pelo** fato de que a lógica de controle (11) incrementa igualmente os circuitos de contador de uso de recursos em resposta a determinação de que nenhuma da pluralidade de threads está no estado de ciclo particular.

17. Núcleo de processador, de acordo com a reivindicação 11, **caracterizado pelo** fato de que a lógica de controle (11) incrementa os circuitos de contador de uso de recursos de acordo com um valor fracionário de recursos internos usados por threads correspondentes em um anterior dos intervalos periódicos, em resposta a determinação de que

nenhuma dentre a pluralidade de threads está no estado de ciclo particular.

18. Núcleo de processador, de acordo com a reivindicação 11, **caracterizado pelo** fato de que a lógica de controle (11) atualiza de forma fracionária os circuitos de contador de uso de recursos em conformidade com um nível de prioridade dos threads correspondentes, em resposta a determinação de que nenhuma da pluralidade de threads está no estado de ciclo particular.

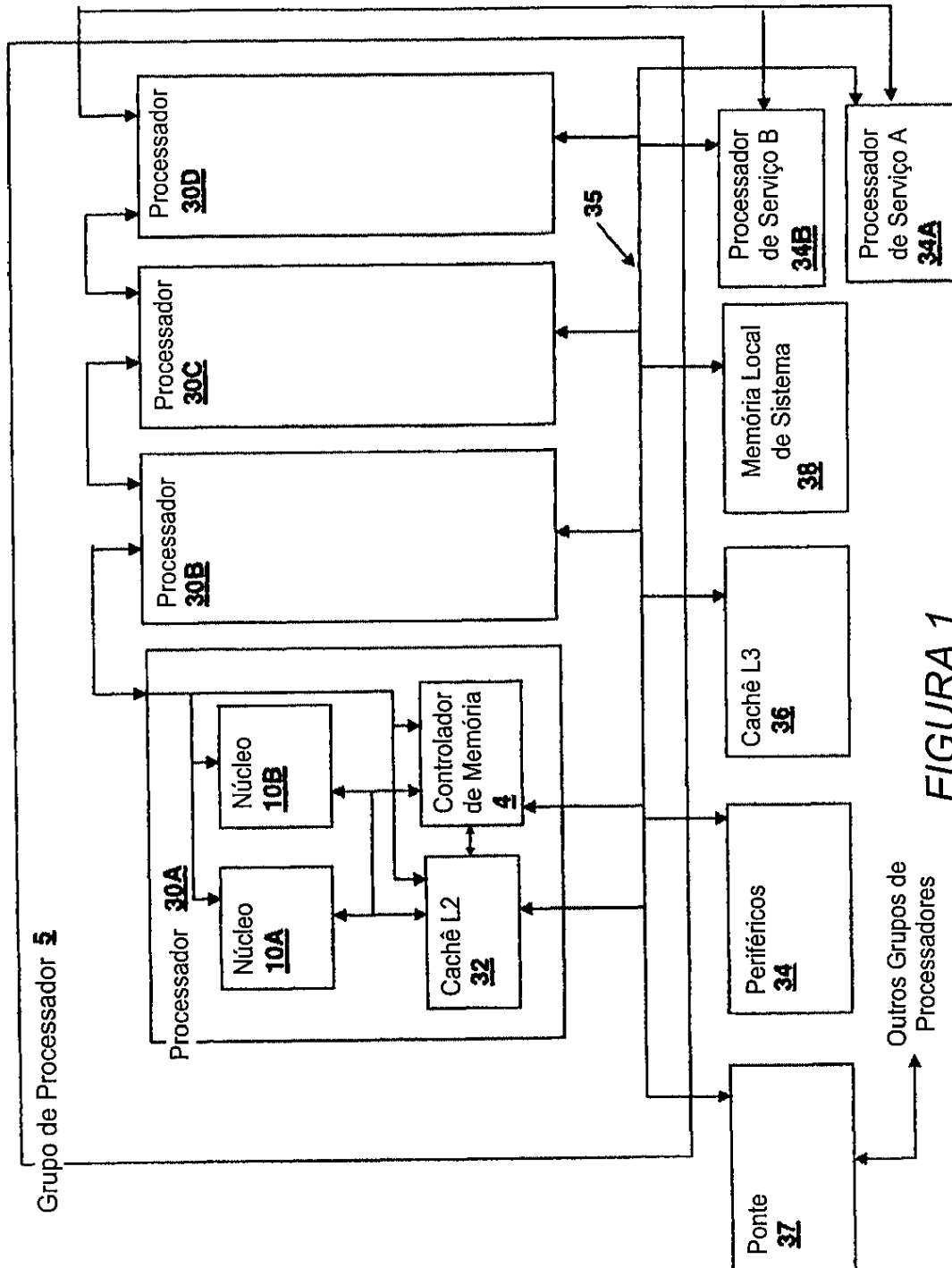


FIGURA 1

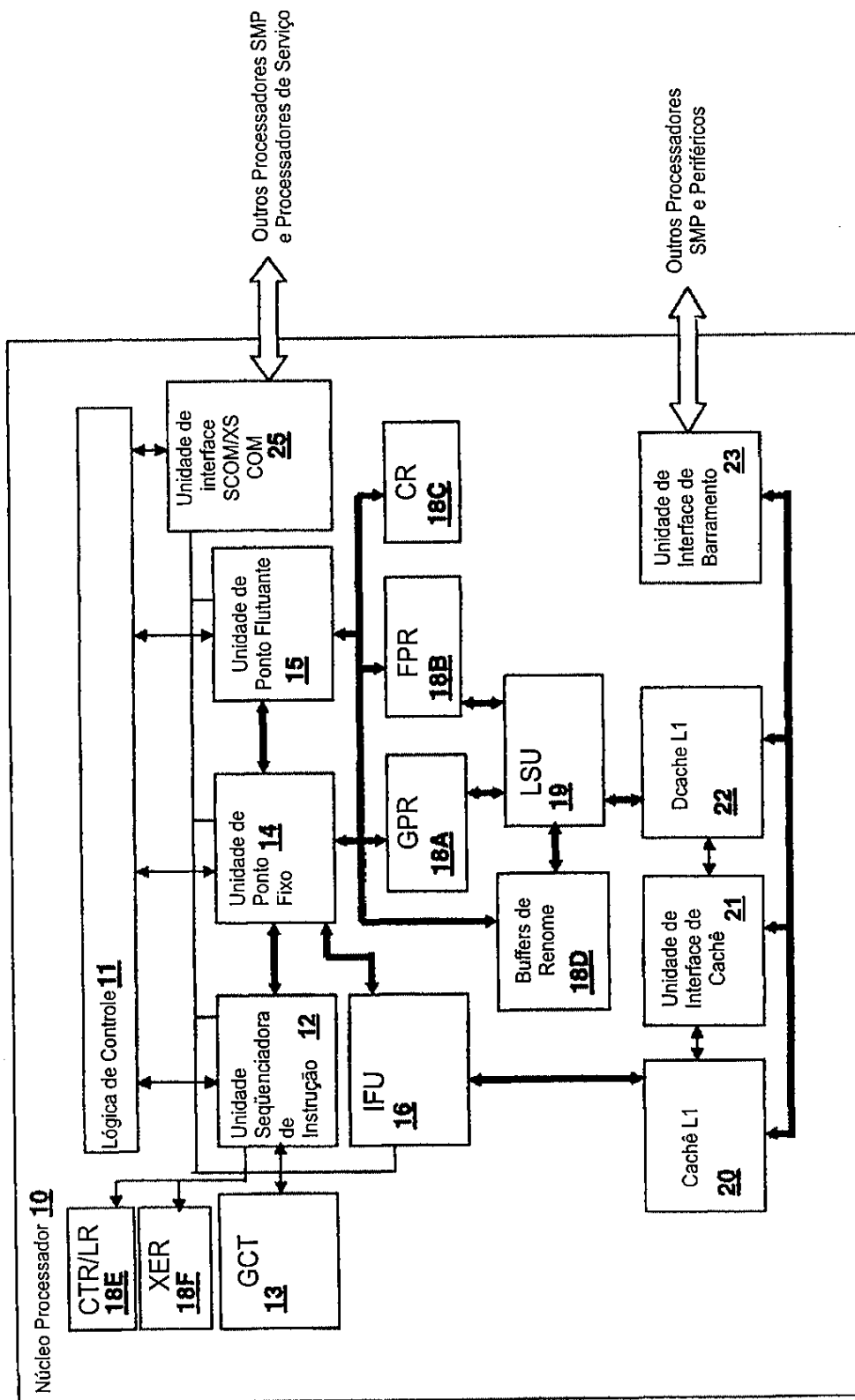


FIGURA 2

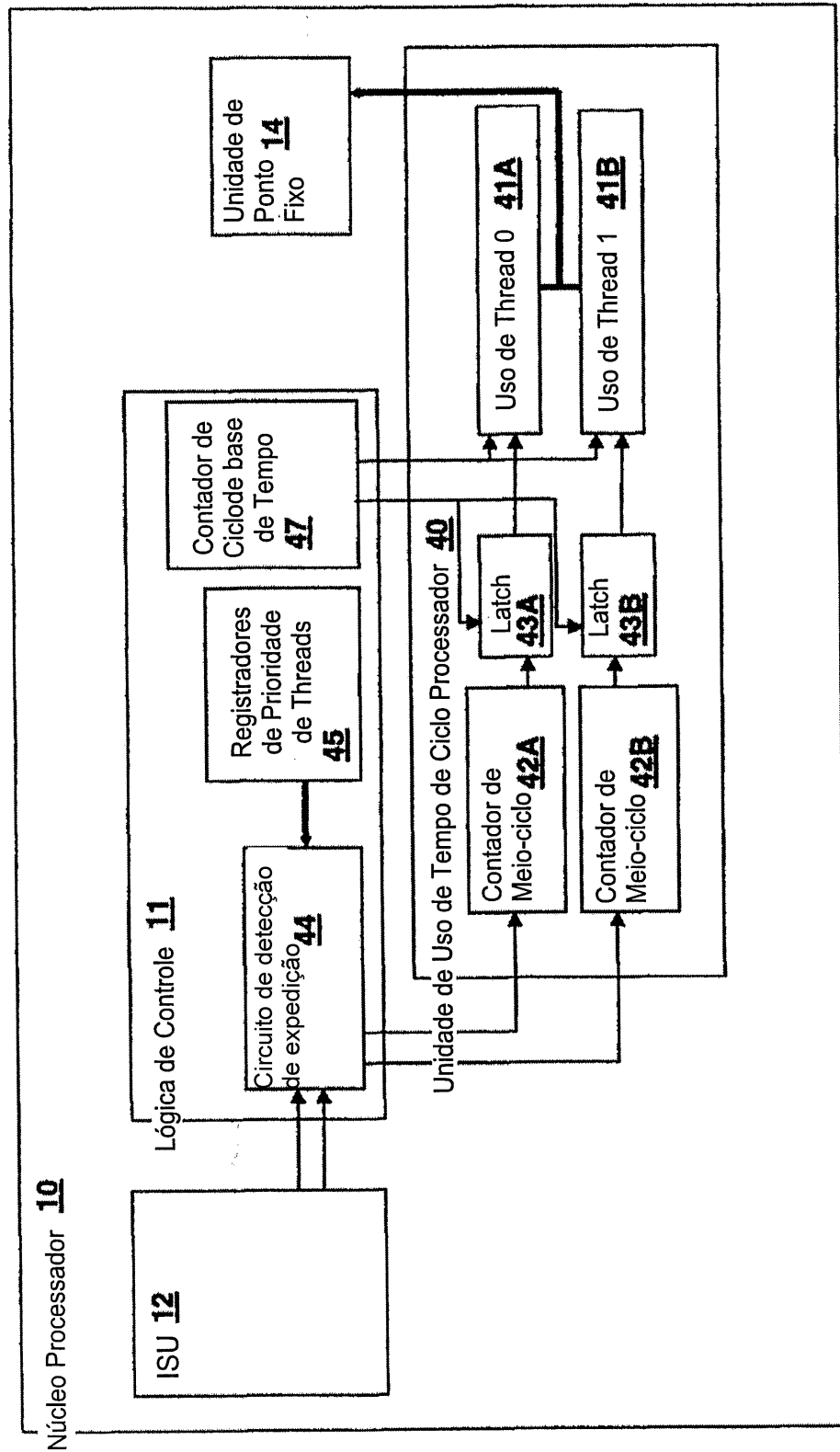
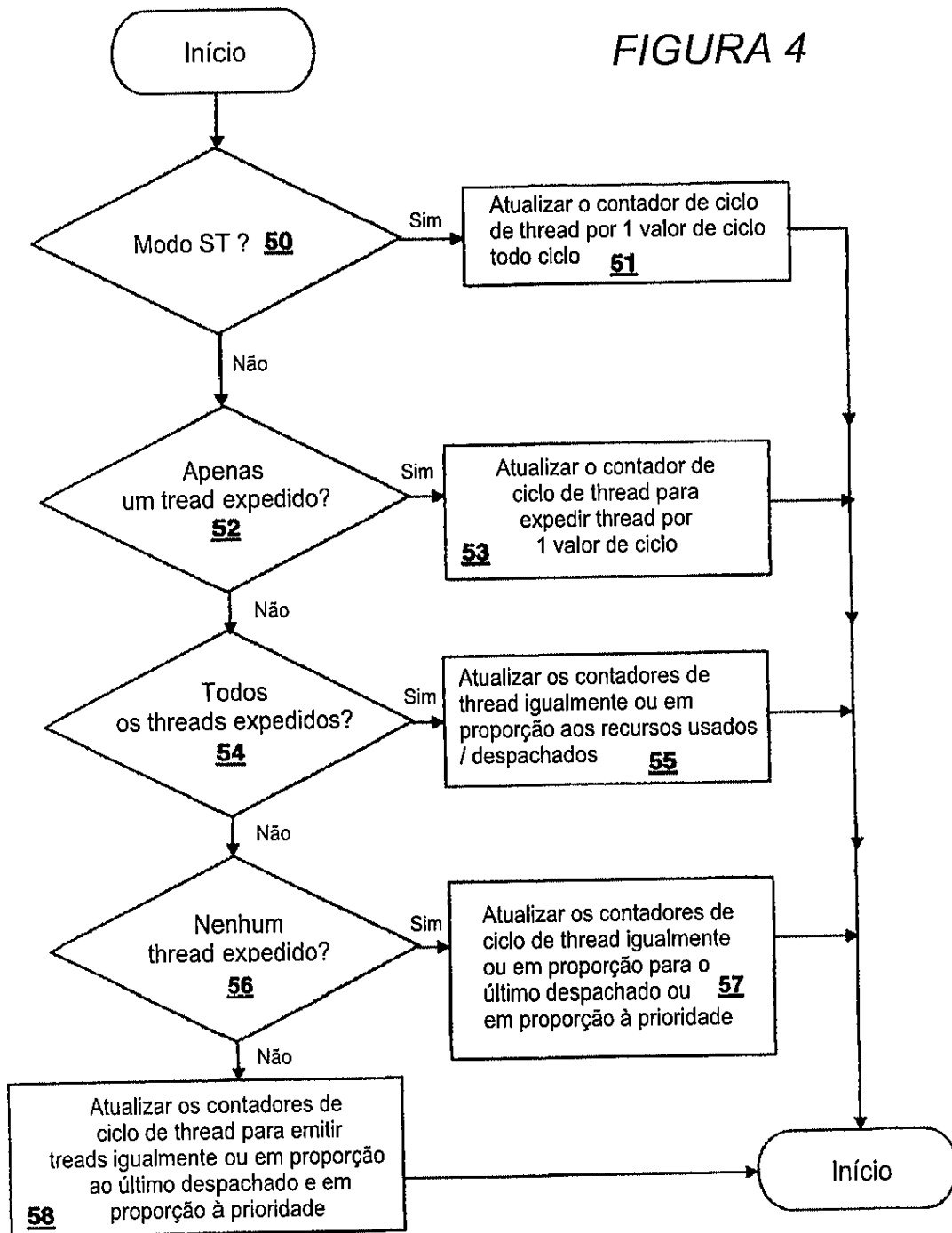


FIGURA 3

FIGURA 4



Resumo da Patente de Invenção para: "MÉTODO E LÓGICA CONTÁBIL PARA DETERMINAR A UTILIZAÇÃO DE RECURSO DE PROCESSADOR PER-THREAD EM UM PROCESSADOR DE MÚLTIPLOS THREADS SIMULTÂNEOS (SMT)".

5 Um método contábil e lógica para determinar utilização de recurso de processador por-thread em um processador de múltiplos threads (SMT) provê um mecanismo para contabilizar o uso do recurso de processador por programas e threads dentro de programas. O uso de recurso relativo é
10 determinado pela detecção de despacho de instruções para múltiplos thread ativos dentro do processador, que podem incluir threads ociosos que ainda estão ocupando recursos do processador. Se instruções são despachadas para todos os threads ou nenhum thread, o ciclo de processador é
15 contabilizado igualmente para todos os threads. Alternativamente, se nenhum thread está no estado de despacho, a contabilização pode ser feita usando um estado anterior ou em conformidade com taxas de níveis de prioridade de threads. Se apenas um thread é despachado,
20 aquele thread é contabilizado pelo ciclo de processador completo. Se múltiplos threads são despachados, mas menos que todos os threads são despachados (em um processador suportando mais que dois threads), o ciclo de processador é faturado uniformemente pelos threads despachados. Múltiplos

P10409710

Sp

threads podem ser detectados para os threads e um uso de recurso fracionado para cada thread e os contadores podem ser atualizados de acordo com os seus usos fracionados.