



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

(11) PI 0807951-0 B1



(22) Data do Depósito: 21/02/2008

(45) Data de Concessão: 14/05/2019

(54) Título: PROCESSADOR DE MULTIMÍDIA MULTI-ENCADEADO SOB DEMANDA

(51) Int.Cl.: G06F 9/38; G06F 9/455; G06F 12/08; G06F 12/10.

(30) Prioridade Unionista: 21/02/2007 US 11/677,362.

(73) Titular(es): QUALCOMM INCORPORATED.

(72) Inventor(es): YUN DU; CHUN YU; GUOFANG JIAO.

(86) Pedido PCT: PCT US2008054620 de 21/02/2008

(87) Publicação PCT: WO 2008/103854 de 28/08/2008

(85) Data do Início da Fase Nacional: 20/08/2009

(57) Resumo: PROCESSADOR DE MULTIMÍDIA MULTI-ENCADEADO SOB DEMANDA Um dispositivo inclui um processador de multimídia que pode suportar de forma simultânea múltiplos aplicativos para vários tipos de multimídia como gráficos, áudio, vídeo, câmera, jogos, etc. O processador de multimídia inclui recursos de armazenamento configuráveis para armazenar instruções, dados e informações de estado para os aplicativos e unidades de processamento atribuíveis para executar vários tipos de processamento para os aplicativos. Os recursos de armazenamento configuráveis podem incluir uma cache de instruções para armazenar instruções para os aplicativos, bancos de registrador para armazenar dados para os aplicativos, registradores de contexto para armazenar informações de estado para encadeamentos dos aplicativos, etc. As unidades de processamento podem incluir uma unidade de lógica aritmética (ALU), um núcleo de função elementar, um núcleo de lógica, um amostrador de textura, uma unidade de controle de carga, um controlador de fluxo, etc. O processador de multimídia aloca uma porção configurável dos recursos de armazenamento para cada aplicativo e dinamicamente atribui as unidades de processamento aos aplicativos, conforme solicitado por tais aplicativos.

"PROCESSADOR DE MULTIMÍDIA MULTI-ENCADEADO SOB DEMANDA"

FUNDAMENTOS

CAMPO

A presente revelação refere-se genericamente à
5 eletrônica, e mais especificamente a um processador.

FUNDAMENTOS

Processadores são amplamente utilizados para
várias finalidades como computação, comunicação,
funcionamento em rede (networking), etc. Um processador
10 pode ser um processador de propósito geral como uma unidade
de processamento central (CPU) ou um processador
especializado como um processador de sinais digitais (DSP)
ou uma unidade de processamento gráfico (GPU). Um
processador de propósito geral pode suportar um conjunto
15 genérico de instruções e funções genéricas, que podem ser
utilizadas por aplicativos de vários tipos. Um processador
de propósito geral pode ser ineficiente para certos
aplicativos com exigências específicas de processamento. Ao
contrário, um processador especializado pode suportar um
20 conjunto limitado de instruções e funções especializadas,
que podem ser customizadas para aplicativos específicos.
Isto permite que o processador especializado suporte
eficientemente os aplicativos para as quais é projetado.
Entretanto, a gama de aplicativos suportados pelo
25 processador especializado pode ser limitado.

Um dispositivo como um telefone celular, um
assistente digital pessoal (PDA), ou um computador laptop
pode suportar aplicativos de vários tipos. É desejável
executar esses aplicativos de forma tão eficientemente
30 quanto possível e com tão pouco hardware quanto possível
para reduzir custo, energia, etc.

SUMÁRIO

Um dispositivo incluindo um processador de multimídia que pode simultaneamente suportar múltiplos aplicativos é descrito aqui. Esses aplicativos podem ser para vários tipos de multimídia como gráficos, áudio, vídeo, câmera, jogos, etc. O processador de multimídia compreende recursos de armazenamento configuráveis para armazenar instruções, dados, e informações de estado para os aplicativos e unidades de processamento atribuíveis para executar vários tipos de processamento para os aplicativos.

Os recursos de armazenamento configuráveis podem incluir uma cache de instrução para armazenar instruções para os aplicativos, registrar bancos para armazenar dados para os aplicativos, registradores de contexto para armazenar informações de estado para encadeamentos dos aplicativos, etc. As unidades de processamento podem incluir um núcleo de unidade de lógica aritmética (ALU), um núcleo de função elementar, um núcleo de lógica, um amostrador de textura, uma unidade de controle de carga, um controlador de fluxo, etc. que podem operar como descrito abaixo. O processador de multimídia aloca uma porção configurável dos recursos de armazenamento para cada aplicativo e atribui dinamicamente as unidades de processamento aos aplicativos conforme solicitado por esses aplicativos. Cada aplicativo observa, desse modo, um processador virtual independente e não necessita estar ciente dos outros aplicativos que executam simultaneamente. O processador de multimídia pode incluir ainda uma unidade de interface de entrada para receber de forma assíncrona encadeamentos dos aplicativos, uma unidade de interface de saída para fornecer assincronamente resultados para os aplicativos, e uma unidade de controle de carga para buscar instruções e dados para os aplicativos, conforme necessário, a partir de uma memória cache e/ou uma memória principal.

O processador de multimídia pode determinar carregamento com base na percentagem de tempo que as unidades de processamento são atribuídas aos aplicativos. O processador de multimídia pode ajustar velocidade de relógio com base no carregamento para reduzir consumo de energia.

Vários aspectos e características da revelação são descritos em detalhe adicional abaixo.

BREVE DESCRIÇÃO DOS DESENHOS

10 A figura 1 mostra um diagrama de blocos de um sistema de multimídia.

A figura 2 mostra um diagrama de blocos de um processador de multimídia.

15 A figura 3 mostra alocação de recursos de armazenamento a N aplicativos.

A figura 4 mostra atribuição de unidades de processamento aos N aplicativos.

A figura 5 mostra um processador virtual para cada um dentre os N aplicativos.

20 A figura 6 mostra um diagrama de blocos de um programador de encadeamento.

A figura 7 mostra um desenho de uma unidade de armazenamento com uma arquitetura de memória virtual.

25 A figura 8 mostra tabelas de consulta física e lógica para a unidade de armazenamento na figura 7.

A figura 9 mostra um processo para suportar aplicativos de multimídia.

A figura 10 mostra um diagrama de blocos de um dispositivo de comunicação sem fio.

30 DESCRIÇÃO DETALHADA

A figura 1 mostra um diagrama de blocos de um sistema de multimídia 100. O sistema 100 pode ser um sistema independente ou parte de um sistema maior como um

sistema de computação (por exemplo, um computador laptop), um dispositivo de comunicação sem fio (por exemplo, um telefone celular), um sistema de jogos (por exemplo, um console de jogos), etc. O sistema 100 suporta N aplicativos de multimídia, que são mencionados como aplicativos 1 até N. Em geral, N pode ser qualquer valor inteiro. Um aplicativo também pode ser mencionado como um programa, um programa de software, etc. Um aplicativo de multimídia pode servir para qualquer tipo de multimídia como gráfico, áudio, vídeo, câmera, jogos, etc. Os aplicativos podem iniciar e terminar em tempos diferentes, e qualquer número de aplicativos pode ser executado em paralelo em qualquer momento dado.

O sistema 100 pode suportar gráfico bidimensional (2-D) e/ou tridimensional (3-D). Uma imagem 2-D ou 3-D pode ser representada com polígonos (tipicamente triângulos). Cada triângulo pode ser composto de elementos de imagem (pixels). Cada pixel pode ter vários atributos como coordenadas de espaço, valores de cor, coordenadas de textura, etc. Cada atributo pode ter até quatro componentes. Por exemplo, coordenadas de espaço podem ser dadas por três componentes x, y e z ou quatro componentes x, y, z e w, onde x e y são coordenadas horizontais e verticais, z é profundidade, e w é uma coordenada homogênea. Valores de cor podem ser dados por três componentes r, g, e b ou quatro componentes r, g, b e a, onde r é vermelho, g é verde, b é azul e a é um fator de transparência que determina a transparência de um pixel. Coordenadas de textura são tipicamente dadas por coordenadas horizontal e vertical, u e v. Um pixel também pode ser associado a outros atributos.

O sistema 100 inclui um processador de multimídia 120, uma máquina de textura 180, e uma memória cache

configurável 190. O processador de multimídia 120 pode executar vários tipos de processamento para aplicativos de multimídia, como descrito abaixo. A máquina de textura 180 pode executar operações de gráfico como mapeamento de
5 textura, que é uma operação complexa de gráfico envolvendo modificação da cor de pixels com a cor de uma imagem de textura. A memória cache 190 é uma memória rápida que pode armazenar instruções e dados para o processador de multimídia 120 e máquina de textura 180. O sistema 100 pode
10 incluir outras unidades.

O processador de multimídia 120 executa processamento para os N aplicativos. O processador de multimídia 120 pode dividir o processamento de cada aplicativo em uma série de encadeamentos, por exemplo,
15 automaticamente e transparente para o aplicativo. Um encadeamento (ou encadeamento de execução) pode indicar uma tarefa específica que pode ser executada com um conjunto de uma ou mais instruções. Encadeamentos permitem que um aplicativo tenha múltiplas tarefas executadas
20 simultaneamente por diferentes unidades e permite ainda que diferentes aplicativos compartilhem processamento e recursos de armazenamento.

No desenho mostrado na figura 1, o processador de multimídia 120 inclui uma unidade de interface de entrada
25 122, uma unidade de interface de saída 124, um programador de encadeamento 130, um controlador de fluxo 132, um controlador mestre 134, unidades de processamento atribuíveis 140, recursos de armazenamento configuráveis 150, e uma unidade de controle de carga 170. A unidade de
30 interface de entrada 122 recebe encadeamentos a partir dos N aplicativos e fornece esses encadeamentos ao programador de encadeamento 130. O programador de encadeamento 130 executa várias funções para programar e gerenciar execução

dos encadeamentos, como descrito abaixo. O controlador de fluxo 132 auxilia com controle de fluxo de programa/aplicativo. O controlador mestre 134 recebe informações como modo de processamento, formato de dados, etc. e configura a operação de várias unidades no processador de multimídia 120 de acordo. Por exemplo, o controlador mestre 134 pode decodificar comando, configurar estado para aplicativos, e controlar uma sequência de atualização de estado.

10 No desenho mostrado na figura 1, unidades de processamento atribuíveis 140 incluem um núcleo ALU 142, um núcleo de função elementar 144, um núcleo de lógica 146, e um amostrador de textura 148. Um núcleo se refere genericamente a uma unidade de processamento em um circuito integrado. Os termos "núcleo", "máquina", "motor", "processador", "unidade de processamento", "unidade de hardware", etc., pode ser utilizados de forma intercambiável. Em geral, unidades de processamento atribuíveis 140 podem incluir qualquer número de unidades de processamento e qualquer tipo de unidade de processamento. Cada unidade de processamento pode operar independentemente das outras unidades de processamento.

25 O núcleo ALU 142 pode executar operações aritméticas como adição, subtração, multiplicação, multiplicar e acumular, produto escalar, absoluto, negação, comparação, saturação, etc. O núcleo ALU 142 pode compreender uma ou mais ALUs escalares e/ou uma ou mais ALUs de vetor. Uma ALU escalar pode operar em uma componente em um tempo. Uma ALU de vetor pode operar em 30 múltiplas componentes (por exemplo, quatro) de uma vez. O núcleo de função elementar 144 pode computar funções elementares transcendentais como seno, cosseno, recíproco, logaritmo, exponencial, raiz quadrada, raiz quadrada

recíproca, etc., que podem ser amplamente utilizados por aplicativos de gráfico. O núcleo de função elementar 144 pode melhorar o desempenho por computar as funções elementares em bem menos tempo do que o tempo necessário para executar aproximações polinomiais das funções elementares utilizando instruções simples. O núcleo de função elementar 144 pode compreender uma ou mais unidades de função elementar. Cada unidade de função elementar pode computar uma função elementar para um componente de cada vez.

O núcleo de lógica 146 pode executar operações lógicas (por exemplo, AND, OR, XOR, etc.), operações no sentido de bit (por exemplo, deslocamentos para a esquerda e direita), operações de número inteiro, comparação, operações de gerenciamento de buffer de dados (por exemplo, empurrar, saltar, etc.), e/ou outras operações. O núcleo de lógica 146 pode executar também conversão de formato, por exemplo, de inteiros para números de ponto flutuante, e vice versa. O amostrador de textura 148 pode executar pré-processamento para máquina de textura 180. Por exemplo, o amostrador de textura 148 pode ler coordenadas de textura, anexar código e/ou outras informações, e enviar sua saída para a máquina de textura 180. O amostrador de textura 148 pode fornecer também instruções para a máquina de textura 180 e receber resultados da máquina de textura.

No desenho mostrado na figura 1, recursos de armazenamento configuráveis 150 incluem registradores de contexto 152, uma cache de instrução 154, bancos de registrador 156, e um buffer de constante 158. Em geral, recursos de armazenamento configuráveis 150 podem incluir qualquer número de unidades de armazenamento e qualquer tipo de unidade de armazenamento. Registradores de contexto 152 armazenam informações de estado ou contexto para

encadeamentos a partir de N aplicativos. A cache de instrução 154 armazena instruções para os encadeamentos. Essas instruções indicam operações específicas para executar para cada encadeamento. Cada operação pode ser uma
5 operação aritmética, uma função elementar, uma operação de lógica, uma operação de acesso de memória, etc. A cache de instrução 154 pode ser carregada com instruções provenientes da memória cache 190 e/ou uma memória principal (não mostrada na figura 1), conforme necessário,
10 via unidade de controle de carga 170. Os bancos de registrador 156 armazenam dados para os aplicativos bem como resultados intermediários e finais a partir de unidades de processamento 150. O buffer de constante 158 armazena valores constantes (por exemplo, fatores de
15 escala, pesos de filtro, etc.) utilizados pelas unidades de processamento 140 (por exemplo, núcleo de ALU 142 e núcleo de lógica 146).

A unidade de controle de carga 170 pode controlar o carregamento de instruções, dados e constantes para os N
20 aplicativos. A unidade de controle de carga 170 faz interface com a memória cache 190 e carrega cache de instrução 154, bancos de registrador 156, e buffer de constante 158 com instruções, dados, e constantes a partir da memória cache 190. A unidade de controle de carga 170
25 também grava os dados e resultados em bancos de registrador 156 para memória cache 190. A unidade de interface de saída 124 recebe os resultados finais para os encadeamentos executados a partir de bancos de registrador 156 e provê esses resultados para os aplicativos. As unidade de
30 interface de entrada 122 e unidade de interface de saída 124 podem fornecer interface assíncrona a unidades externas (por exemplo, câmera, unidade de exibição, etc.), associadas aos N aplicativos.

A figura 1 mostra um desenho de exemplo de processador de multimídia 120. Em geral, o processador de multimídia 120 pode incluir qualquer conjunto de unidades de processamento atribuíveis e qualquer conjunto de recursos de armazenamento configuráveis. Os recursos de armazenamento configuráveis podem armazenar instruções, dados, informações de estado, etc., para os aplicativos. As unidades de processamento podem executar qualquer tipo de processamento para os aplicativos. O controlador de fluxo 132 e a unidade de controle de carga 170 também podem ser considerados como unidades de processamento atribuíveis embora não estejam incluídos nas unidades 140. O processador de multimídia 120 pode incluir também outras unidades de processamento, armazenamento e controle não mostradas na figura 1. O processador de multimídia 120 pode alocar uma porção configurável dos recursos de armazenamento a cada aplicativo e atribuir dinamicamente as unidades de processamento aos aplicativos conforme solicitado por esses aplicativos.

O processador de multimídia 120 pode implementar uma ou mais interfaces de programação de aplicativo de gráfico (APIs), como Open Graphics Library (OpenGL), Direct3D, Open Vector Graphics (OpenVG), etc. Essas várias APIs de gráfico são conhecidas na técnica. O processador de multimídia 120 pode suportar também gráficos 2-D, ou gráficos 3-D, ou ambos.

A figura 2 mostra um diagrama de blocos de um desenho de processador de multimídia 120 na figura 1. Nesse desenho, o programador de encadeamento 130 faz interface com o núcleo de ALU 142, núcleo de função elementar 144, núcleo de lógica 146, e amostrador de textura 148 nas unidades de processamento atribuíveis 140. O programador de encadeamento 130 faz interface ainda com a unidade de

interface de entrada 122, controlador de fluxo 132, controlador mestre 134, registradores de contexto 152, cache de instrução 154, e unidade de controle de carga 170. Os bancos de registrador 156 fazem interface com o núcleo de ALU 142, núcleo de função elementar 144, núcleo de lógica 146, e amostrador de textura 148 nas unidades de processamento atribuíveis 140, unidade de controle de carga 170, e unidade de interface de saída 124. A unidade de controle de carga 170 faz interface ainda com a cache de instrução 154, buffer de constante 158, e memória cache 190. As várias unidades no processador de multimídia 120 podem fazer interface também entre si em outros modos.

Uma memória principal 192 pode fazer parte do sistema 100 ou pode ser externa ao sistema 100. A memória principal 192 é uma memória mais lenta, grande localizada mais distante (por exemplo, off-chip) do processador de multimídia 120. A memória principal 192 pode armazenar todas as instruções e dados para os N aplicativos sendo executadas pelo processador de multimídia 120. As instruções e dados na memória principal 192 podem ser carregadas na memória cache 190 quando e conforme necessário.

O processador de multimídia 120 pode ser projetado e operado de tal modo que pareça como um processador virtual independente para cada aplicativo sendo executado. A cada aplicativo podem ser atribuído recursos de armazenamento suficientes para instruções, dados, constante e informação de estado. Cada aplicativo pode ter seu próprio estado individual (por exemplo, contador de programa, formato de dados, etc.), que pode ser mantido pelo processador de multimídia 120. A cada aplicativo também podem ser atribuídas unidades de processamento atribuídas com base nas instruções a serem executadas para

tal aplicativo. Os N aplicativos podem executar simultaneamente sem interferir entre si e sem ter de estar ciente dos outros aplicativos. O processador de multimídia 120 pode ajustar o alvo de desempenho para cada aplicativo com base na demanda de aplicativo e/ou outros fatores, por exemplo, prioridade.

A figura 3 mostra uma alocação de exemplo de recursos de armazenamento para os N aplicativos. A cada aplicativo pode ser alocada uma porção de registradores de contexto 152, uma porção de cache de instrução 154, uma porção de bancos de registrador 156, e uma porção de buffer de constante 158. Para cada unidade de armazenamento, a porção alocada a um dado aplicativo pode ser zero ou não zero dependendo das exigências de armazenamento de tal aplicativo.

Os registradores de contexto 152 podem ser dinamicamente atribuídos a encadeamentos a partir dos N aplicativos e podem armazenar vários tipos de informações para os encadeamentos, como descrito abaixo. Os registradores de contexto 152 podem ser atualizados à medida que encadeamentos são aceitos, executados e concluídos. A cache de instrução 154 e os bancos de registrador 156 podem ser alocados a cada aplicativo no início de execução, por exemplo, com base nas exigências do aplicativo. Para cada aplicativo, as porções alocadas para cache de instrução 154 e /ou bancos de registrador 156 podem mudar durante a execução do aplicativo com base em sua demanda e outros fatores. O buffer de constante 158 pode armazenar constantes que podem ser utilizadas para qualquer aplicativo. Constantes para um dado aplicativo podem ser carregadas no buffer de constante 158 quando necessário e podem ficar então disponíveis para uso por todos os aplicativos.

As unidades de armazenamento podem ser projetadas para suportar alocação flexível de recursos de armazenamento para os aplicativos, conforme descrito abaixo. As unidades de armazenamento também podem ser projetadas para simplificar acesso à memória pelos aplicativos, como também descrito abaixo.

A figura 4 mostra uma atribuição de exemplo de unidades de processamento para os N aplicativos. Uma linha de tempo separada pode ser mantida para cada unidade de processamento como núcleo de ALU 142, núcleo de função elementar 144, núcleo de lógica 146, amostrador de textura 148, controlador de fluxo 132, e unidade de controle de carga 170. A linha de tempo para cada unidade de processamento pode ser dividida em partições de tempo. Uma partição de tempo é a menor unidade de tempo que pode ser alocada a um aplicativo e pode corresponder a um ou mais ciclos de relógio. As unidades de processamento podem ter partições de tempo da mesma duração ou durações diferentes.

As partições de tempo para o núcleo de ALU 142 podem ser atribuídas a qualquer um dos aplicativos. No exemplo mostrado na figura 4, o núcleo de ALU 142 é atribuído ao aplicativo 1 (App 1) em partições de tempo t e $t+1$, ao aplicativo 3 em partição de tempo $t+2$, aplicativo N em partição de tempo $t+3$, etc. De modo semelhante, as partições de tempo para núcleo de função elementar 144, núcleo de lógica 146, amostrador de textura 148, controlador de fluxo 132, e unidade de controle de carga 170 podem ser atribuídas a qualquer um dos aplicativos. O processador de multimídia 120 pode atribuir dinamicamente as unidades de processamento aos aplicativos em demanda com base nas exigências de processamento desses aplicativos.

A figura 5 mostra um processador virtual para cada um dos N aplicativos. Cada aplicativo observa um

processador virtual tendo todas as unidades de processamento utilizadas por tal aplicativo. A cada aplicativo podem ser atribuídas unidades de processamento com base na demanda de processamento de tal aplicativo, e

5 as unidades de processamento atribuídas podem ser mostradas em uma linha de tempo para tal aplicativo. No exemplo mostrado na figura 5, ao aplicativo 1 é atribuído o núcleo de ALU 142 em partições de tempo t e $t+1$, então o núcleo de lógica 144 na partição de tempo $t+2$, então a unidade de

10 controle de carga 170 na partição de tempo $t+3$, a seguir o núcleo de ALU 142 na partição de tempo $t+4$, a seguir a unidade de controle de carga 170 na partição de tempo $t+5$, então o núcleo de lógica 144 na partição de tempo $t+6$, etc. O aplicativo 1 não utiliza e a ele não é atribuído núcleo

15 de função elementar 144, amostrador de textura 148, e controlador de fluxo 132 durante as partições de tempo mostradas na figura 5. Aos aplicativos 2 até N são atribuídas unidades de processamento em sequências diferentes.

20 Como mostrado na figura 5, a cada aplicativo podem ser atribuídas unidades de processamento pertinentes no processador de multimídia 120. As unidades de processamento específicas a serem atribuídas a cada aplicativo podem mudar com o passar do tempo dependendo das

25 exigências de processamento. Cada aplicativo não necessita estar ciente do outro aplicativo nem as atribuições das unidades de processamento para os outros aplicativos.

O processador de multimídia 120 pode suportar múltiplos encadeamentos para obter execução paralela de

30 instruções e melhorar a eficiência geral. Múltiplos encadeamentos se refere à execução de múltiplos fluxos em paralelo por unidades de processamento diferentes. O programador de encadeamento 130 pode aceitar encadeamentos

a partir dos N aplicativos, determinar quais encadeamentos estão prontos para execução, e despachar esses encadeamentos para diferentes unidades de processamento. O programador de encadeamento 130 pode gerenciar a execução dos encadeamentos e utilização das unidades de processamento.

A figura 6 mostra um diagrama de blocos de um desenho de programador de encadeamento 130 nas figuras 1 e 2. Nesse desenho, o programador de encadeamento 130 inclui um programador de encadeamento central 610, um decodificador de nível elevado 612, uma unidade de monitor de uso de recurso 614, uma fila ativa 620 e uma fila de espera 622. Registradores de contexto 152 incluem t registradores de contexto 630a até 630t para t encadeamentos, onde t pode ser qualquer valor.

O programador de encadeamento central 610 pode comunicar-se com as unidades de processamento 132, 142, 144, 146, 148 e 170 e registradores de contexto 630a até 630t através de interfaces de concessão e solicitação (Req). O programador 610 pode emitir solicitações para cache de instrução 154 e receber indicações de acerto/erro em resposta. Em geral, a comunicação entre essas unidades pode ser obtida com vários mecanismos como sinais de controle, interrupções, mensagens, registradores, etc.

O programador de encadeamento central 610 pode executar várias funções para programar encadeamentos. O programador de encadeamento central 610 pode determinar se deve aceitar novos encadeamentos a partir dos N aplicativos, despachar encadeamentos que são para execução, e liberar/remover encadeamentos que são concluídos. Para cada encadeamento, o programador de encadeamento central 610 pode determinar se recursos (por exemplo, instruções, unidades de processamento, bancos de registrador, etc.)

exigidos por aquele encadeamento estão disponíveis, ativar o encadeamento e colocar o mesmo em fila ativa 620 se os recursos exigidos forem disponíveis, e colocar o encadeamento em fila de espera 622 se qualquer recurso não estiver disponível. A fila ativa 620 armazena encadeamentos que estão prontos para execução, e a fila de espera 622 armazena encadeamentos que não estão prontos para execução.

O programador de encadeamento central 610 pode gerenciar também execução de encadeamentos. Em cada intervalo de programação (por exemplo, cada partição de tempo), o programador de encadeamento central 610 pode selecionar um número de encadeamentos candidatos em fila ativa 620 para avaliação e possível despacho. O programador de encadeamento central 610 pode determinar as unidades de processamento a utilizar para os encadeamentos candidatos, verificar conflitos de leitura/gravação de unidades de armazenamento, e despachar diferentes encadeamentos para diferentes unidades de processamento para execução. O processador de multimídia 120 pode suportar execução de M encadeamentos simultaneamente, onde M pode ser qualquer valor apropriado (por exemplo, $M = 12$). Em geral, M pode ser selecionado com base no tamanho dos recursos de armazenamento (por exemplo, cache de instrução 154 e bancos de registrador 156), a latência ou retardo para operações de carga, os pipelines das unidades de processamento, e/ou outros fatores de modo que as unidades de processamento sejam utilizadas tão completamente quanto possível.

O programador de encadeamento central 610 pode atualizar o status e estado de encadeamentos conforme apropriado. O programador de encadeamento central 610 pode colocar um encadeamento em fila de espera 622 caso (a) a próxima instrução para o encadeamento não seja encontrada em cache de instrução 154, (b) a próxima instrução esteja

esperando por resultados a partir de uma instrução anterior, ou (c) algumas outras condições de espera sejam atendidas. O programador de encadeamento central 610 pode mover um encadeamento da fila de espera 622 para a fila
 5 ativa 620 quando as condições de espera não mais forem verdadeiras.

O programador de encadeamento central 610 pode manter um contador de programa para cada encadeamento e pode atualizar o contador de programa à medida que as
 10 instruções são executadas ou o fluxo de programa é alterado. O programador 610 pode solicitar a assistência a partir do controlador de fluxo 132 para controlar o fluxo de programa para os encadeamentos.

O controlador de fluxo 132 pode tratar instruções
 15 if/else, loops, chamadas de sub-rotina, ramificações, instruções de comutação, pixel kill e/ou outras instruções de alteração de fluxo. O controlador de fluxo 132 pode avaliar uma ou mais condições para cada tal instrução, indicar uma alteração no contador de programa de um modo se
 20 a(s) condição(ões) for(em) atendida(s), e indicar uma alteração no contador de programa de outro modo se a(s) condição(ões) não for(em) atendida(s). O controlador de fluxo 132 pode executar também outras funções relacionadas ao fluxo de programa dinâmico. O programador de
 25 encadeamento central 610 pode atualizar o contador de programa com base em resultados a partir do controlador de fluxo 132.

O programador de encadeamento central 610 pode também gerenciar registradores de contexto 152 e atualizar
 30 esses registradores à medida que encadeamentos são aceitos, executados e concluídos. Os registradores de contexto 152 podem armazenar vários tipos de informações para encadeamentos. Por exemplo, um registrador de contexto 630

para um encadeamento pode armazenar (1) um identificador de programa/aplicativo (ID) para o aplicativo ao qual o encadeamento pertence, (2) um contador de programa que aponta para a instrução atual para o encadeamento, (3) uma
5 máscara de cobertura que indica pixels válidos e inválidos para o encadeamento, (4) um indicador ativo que indica quais pixels operar em caso de uma instrução de alterar fluxo, (5) um ponteiro de instrução de reiniciar que indica quando um pixel será reativado se estiver inativo, (6) uma
10 pilha que armazena ponteiros de instrução de retornar para controle de fluxo dinâmico, (7) registradores de endereço para endereçamento relativo, (8) registradores de atributo que armazenam resultados de cálculo de condição de armazenamento, (9) um contador de referência de carga que
15 rastreia solicitações de carga e condições de retorno de dados, e/ou (10) outras informações. Os registradores de contexto 152 podem também armazenar menos informações, mais informações ou informações diferentes.

O processador de multimídia 120 pode suportar um
20 conjunto abrangente de instruções para vários aplicativos de multimídia. Esse conjunto de instruções pode incluir instruções aritméticas, função elementar, lógica, no sentido de bits, controle de fluxo e outras instruções.

A decodificação de dois níveis de instruções pode
25 ser executada para melhorar o desempenho. O decodificador de alto nível 612 pode executar decodificação de alto nível de instruções para determinar tipo de instruções, tipo de operando, identificadores de fonte e destino (IDs), e/ou outras informações utilizadas para programação. Cada
30 unidade de processamento pode incluir um decodificador de instrução separado que executa decodificação de nível baixo de instruções para aquela unidade de processamento. Por exemplo, um decodificador de instrução para o núcleo de ALU

142 pode tratar somente de instruções relacionadas a ALU,
 um decodificador de instrução para núcleo de função
 elementar 144 pode tratar somente de instruções para
 funções elementares, etc. A decodificação de dois níveis
 5 pode simplificar o desenho de programador de encadeamento
 central 610 bem como os decodificadores de instrução para
 as unidades de processamento.

A unidade de monitor de uso de recursos 614
 monitora o uso das unidades de processamento, por exemplo,
 10 por ficar de olho na percentagem de tempo que cada unidade
 de processamento é atribuída. A unidade de monitor 614 pode
 ajustar dinamicamente a operação das unidades de
 processamento para conservar energia de bateria enquanto
 fornece o desempenho desejado. Por exemplo, a unidade de
 15 monitor 614 pode ajustar a velocidade de relógio para o
 processador de multimídia 120 com base em carregamento do
 processador de multimídia para reduzir o consumo de
 energia. A unidade de monitor 614 pode ajustar também a
 velocidade de relógio para cada unidade de processamento
 20 individual com base em carregamento ou percentagem de
 utilização daquela unidade de processamento. A unidade de
 monitor 614 pode selecionar a velocidade mais elevada de
 relógio para carregamento total e pode selecionar
 velocidade de relógio progressivamente mais lenta para
 25 carregamento progressivamente menor. A unidade de monitor
 614 pode também desabilitar/desligar qualquer unidade de
 processamento que não seja atribuída a qualquer aplicativo
 e pode habilitar/ligar a unidade de processamento quando
 for atribuída.

30 Cada encadeamento pode ser baseado em pacote e
 pode operar em uma unidade de dados de até um tamanho
 predeterminado. O tamanho de unidade de dados pode ser
 selecionado com base no desenho das unidades de

armazenamento e processamento, as características dos dados sendo processados, etc. Em um desenho, cada encadeamento opera em até quatro pixels ou até quatro vértices em uma imagem. Os bancos de registrador 156 podem incluir quatro
 5 bancos de registrador que podem armazenar (a) até quatro componentes de cada atributo para pixels, um componente por banco de registrador, ou (b) componentes de atributos para pixels, um pixel por banco de registrador. O núcleo de ALU 142 pode incluir quatro ALUs escalares ou um ALU de vetor
 10 que pode operar em até quatro componentes de cada vez.

Uma unidade de armazenamento (por exemplo, cache de instrução 154 ou bancos de registrador 156) pode ser implementada com uma arquitetura de memória virtual que permite alocação eficiente de recursos de armazenamento
 15 para os aplicativos e fácil acesso dos recursos de armazenamento alocados pelos aplicativos. A arquitetura de memória virtual pode utilizar uma memória virtual e uma memória física. Os aplicativos podem ser seções alocadas da memória virtual e podem executar acesso de memória através
 20 do espaço de endereço virtual. Seções diferentes da memória virtual podem ser mapeadas para seções diferentes da memória física, que armazenam as instruções e/ou dados.

A figura 7 mostra um desenho de uma unidade de armazenamento 700 com uma arquitetura de memória virtual. A
 25 unidade de armazenamento 700 pode ser utilizada para cache de instrução 154, bancos de registrador 156, etc. Nesse desenho, a unidade de armazenamento 700 aparece como uma memória virtual 710 para os aplicativos. A memória virtual 710 pode ser dividida em múltiplos (S) tiles lógicos ou
 30 seções, que são mencionados como tiles lógicos 1 até S. Em geral, S pode ser qualquer valor inteiro igual ou maior que N. Os S tiles podem ter o mesmo tamanho ou tamanhos diferentes. A cada aplicativo pode ser alocado qualquer

número de tiles lógicos consecutivos com base no uso de memória de tal aplicativo e tiles disponíveis. No exemplo mostrado na figura 7, ao aplicativo 1 são alocados tiles lógicos 1 e 2, ao aplicativo 2 são alocados tiles lógicos 3 até 6, etc.

A unidade de armazenamento 700 implementa uma memória física 720 que armazena instruções e/ou dados para os aplicativos. A memória física 720 inclui S tiles físicos 1 até S. Cada tile lógico de memória virtual 710 é mapeado para um tile físico de memória física 720. Um mapeamento exemplar para alguns tiles lógicos é mostrado na figura 7. Nesse exemplo, o tile físico 1 armazena instruções e/ou dados para o tile lógico 2, tile físico 2 armazena instruções e/ou dados para o tile lógico S-1, etc.

O uso de tiles lógicos e tiles físicos pode simplificar a alocação de tiles para aplicativos e o gerenciamento dos tiles. Um aplicativo pode solicitar certas quantidades de recursos de armazenamento para instruções de dados. O processador de multimídia 120 pode alocar um ou mais tiles na cache de instrução 154 e um ou mais tiles em bancos de registrador 156 para o aplicativo. Ao aplicativo podem ser alocados tiles lógicos adicionais, em número menor ou diferentes, conforme necessário.

A figura 8 mostra um desenho de uma tabela de consulta (LUT) de tile lógico 810 e uma tabela de consulta de endereço físico 820 para a unidade de armazenamento 700 na figura 7. Nesse desenho, a tabela de consulta de tile lógico 810 inclui N entradas para os N aplicativos, uma entrada para cada aplicativo. As N entradas podem ser indexadas pelo ID de aplicativo. A entrada para cada aplicativo inclui um campo para o primeiro tile lógico alocado ao aplicativo e outro campo para o número de tiles lógicos alocados ao aplicativo. No exemplo mostrado na

figura 8, ao aplicativo 1 são alocados dois tiles lógicos que iniciam com o tile lógico 1, ao aplicativo 2 são alocados quatro tiles lógicos que iniciam com o tile lógico 3, ao aplicativo 3 são alocados oito tiles lógicos que iniciam com o tile lógico 7, etc. A cada aplicativo podem ser alocados tiles lógicos consecutivos para simplificar a geração de endereços para acesso de memória. Entretanto, aos aplicativos podem ser alocados tiles lógicos em qualquer ordem, por exemplo, o tile lógico 1 pode ser alocado a qualquer aplicativo.

No desenho mostrado na figura 8, a tabela de consulta de endereço físico 820 inclui S entradas para os S tiles lógicos, uma entrada para cada tile lógico. As S entradas da tabela 820 podem ser indexadas por endereço de tile lógico. A entrada para cada tile lógico indica o tile físico para o qual aquele tile lógico é mapeado. No exemplo mostrado nas figuras 7 e 8, o tile lógico 1 é mapeado para o tile físico 4, tile lógico 2 é mapeado para o tile físico 1, o tile lógico 3 é mapeado para o tile físico i, o tile lógico 4 é mapeado para o tile físico S-2, etc. Tabelas de consulta 810 e 820 podem ser atualizadas sempre que a um aplicativo forem alocados tiles lógicos adicionais, em número menor e/ou diferentes. Aos aplicativos podem ser alocadas quantidades diferentes de recursos de armazenamento por simplesmente atualizar as tabelas de consulta, sem ter de efetivamente transferir instruções ou dados entre os tiles físicos.

Uma unidade de armazenamento pode ser desse modo associada a uma memória virtual e uma memória física. A cada aplicativo pode ser alocada uma porção configurável da memória virtual. Pelo menos uma tabela pode ser utilizada para mapear a porção da memória virtual alocada a cada

aplicativo para uma porção correspondente da memória física.

A cada aplicativo são tipicamente alocadas quantidades limitadas de recursos de armazenamento em cache de instrução 154 e bancos de registrador 156 para armazenar instruções e dados, respectivamente, para tal aplicativo. A memória cache 190 pode armazenar instruções adicionais e dados para os aplicativos. Sempre que uma instrução para um aplicativo não estiver disponível em cache de instrução 154, um erro de cache pode ser retornado para o programador de encadeamento 130, que pode então emitir uma solicitação de instrução para a unidade de controle de carga 170. Similarmente, sempre que dados para um aplicativo não estiverem disponíveis em bancos de registrador 156 ou sempre que uma unidade de armazenamento transbordar dados, uma solicitação de dados pode ser emitida para a unidade de controle de carga 170.

A unidade de controle de carga 170 pode receber solicitações de instrução a partir do programador de encadeamento 130 e solicitações de dados a partir de outras unidades. A unidade de controle de carga 170 pode arbitrar essas várias solicitações e gerar solicitações de memória para (a) carregar as instruções desejadas e/ou dados a partir da memória cache 190 ou memória principal 192 e/ou (b) gravar dados para a memória cache 190 ou memória principal 192.

As unidades de armazenamento no processador de multimídia 120 podem armazenar porções pequenas de instruções e dados que são atualmente utilizadas para os aplicativos. A memória cache 190 pode armazenar porções maiores de instruções e dados que poderiam ser utilizadas para os aplicativos. O processador de multimídia 120 pode suportar instrução ilimitada e acesso de dados através da

memória cache 190. Essa capacidade permite que o processador de multimídia 120 suporte aplicativos de qualquer tamanho. O processador de multimídia 120 pode suportar também carga de memória genérica e carga de
5 textura entre memória cache 190 e memória principal 192.

A figura 9 mostra um processo 900 para suportar aplicativos de multimídia. Aplicativos de multimídia são suportados simultaneamente, por exemplo, por um processador de multimídia (bloco 912). Uma porção configurável de
10 recursos de armazenamento é alocada para cada aplicativo para armazenar instruções, dados e informações de estado para o aplicativo (bloco 914). Para o bloco 914, a cada aplicativo pode ser alocada uma porção configurável de uma cache de instrução para armazenar instruções para o
15 aplicativo de multimídia, uma porção configurável de bancos de registrador para armazenar dados para o aplicativo, um ou mais registradores de contexto para armazenar informações de estado para o aplicativo, etc. As unidades de processamento são dinamicamente atribuídas aos
20 aplicativos conforme solicitado por esses aplicativos (bloco 916). Encadeamentos podem ser recebidos de forma assíncrona dos aplicativos e programados para execução (bloco 918). Os resultados da execução dos encadeamentos podem ser fornecidos de forma assíncrona para os
25 aplicativos (bloco 920).

O processador de multimídia descrito aqui pode ser utilizado para dispositivos de comunicação sem fio, dispositivos portáteis, dispositivos de jogo, dispositivos de computação, dispositivos eletrônicos de consumidor,
30 computadores, etc. Um uso exemplar do processador de multimídia para um dispositivo de comunicação sem fio é descrito abaixo.

A figura 10 mostra um diagrama de blocos de um desenho de um dispositivo de comunicação sem fio 1000 em um sistema de comunicação sem fio. O dispositivo sem fio 1000 pode ser um telefone celular, um terminal, um aparelho telefônico, um assistente pessoal digital (PDA), ou algum
 5 outro dispositivo. O sistema de comunicação sem fio pode ser um sistema de Acesso Múltiplo por divisão de código (CDMA), um Sistema Global para comunicação móvel (GSM), ou algum outro sistema.

10 O dispositivo sem fio 1000 é capaz de fornecer comunicação bidirecional através de um percurso de recepção e um percurso de transmissão. No percurso de recepção, sinais transmitidos pelas estações base são recebidos por uma antena 1012 e fornecidos para um receptor (RCVR) 1014.
 15 O receptor 1014 condiciona e digitaliza o sinal recebido e provê amostras para uma seção digital 1020 para processamento adicional. No percurso de transmissão, um transmissor (TMTR) 1016 recebe dados a serem transmitidos a partir da seção digital 1020, processa e condiciona os
 20 dados, e gera um sinal modulado, que é transmitido através da antena 1012 para as estações base.

A seção digital 1020 inclui várias unidades de processamento, interface e memória como, por exemplo, um processador de modem 1022, um processador de sinais
 25 digitais (DSP) 1024, um processador de vídeo/áudio 1026, um controlador/processador 1028, um processador de exibição 1030, uma unidade de processamento central (CPU)/computador de conjunto de instruções reduzidas (RISC) 1032, um processador de multimídia 1034, um processador de câmera
 30 1036, uma memória cache/interna 1038, e uma interface de barramento externo (EBI) 1040. O processador de modem 1022 executa processamento para transmissão e recepção de dados (por exemplo, codificação, modulação, demodulação e

decodificação). O DSP 1024 pode executar processamento especializado para o dispositivo sem fio 1000. O processador de vídeo/áudio 1026 executa processamento em conteúdo de vídeo (por exemplo, imagens fixas, vídeos em movimento, e textos em movimento) para aplicativos de vídeo como camcorder, reprodução de vídeo e conferência de vídeo. O processador de vídeo/áudio 1026 também executa processamento para conteúdo de áudio (por exemplo, áudio sintetizado) para aplicativos de áudio. O controlador/processador 1028 pode orientar a operação de várias unidades na seção digital 1020. O processador de exibição 1030 executa o processamento para facilitar a exibição de vídeo, gráficos, e textos em uma unidade de exibição 1050. CPU/RISC 1032 pode executar processamento de propósito geral para o dispositivo sem fio 1000. O processador de multimídia 1034 executa processamento para aplicativos de multimídia e pode ser implementado como descrito acima para as figuras 1 até 8. O processador de câmera 1036 executa processamento para uma câmera (não mostrada na figura 10). A memória cache/interna 1038 armazena dados e/ou instruções para várias unidades na seção digital 1020 e pode implementar a memória cache 190 nas figuras 1 e 2. EBI 1040 facilita transferência de dados entre a seção digital 1020 (por exemplo, memória cache/interna 1038) e uma memória principal 1060. Aplicativos de multimídia podem ser executados por qualquer um dos processadores na seção digital 1020.

A seção digital 1020 pode ser implementada com um ou mais processadores, microprocessadores, DSPs, RISCs, etc. A seção digital 1020 pode ser fabricada também em um ou mais circuitos integrados de aplicação específica (ASICs) e/ou algum outro tipo de circuitos integrados (ICs).

O processador de multimídia descrito aqui pode ser implementado em vários dispositivos de hardware. Por exemplo, o processador de multimídia pode ser implementado em ASICs, DSPs, dispositivos de processamento de sinais digitais (DSPDs), dispositivos de lógica programáveis (PLDs), disposições de porta programáveis em campo (FPGAs), processadores, controladores, microcontroladores, microprocessadores, dispositivos eletrônicos, outras unidades eletrônicas. O processador de multimídia pode ou não incluir memória incorporada/integrada.

Um equipamento que implementa o processador de multimídia descrito aqui pode ser uma unidade independente ou pode fazer parte de um dispositivo. O dispositivo pode ser (i) um IC independente, (ii) um conjunto de um ou mais ICs que pode incluir ICs de memória para armazenar dados e/ou instruções, (iii) um ASIC como um modem de estação móvel (MSM), (iv) um módulo que pode ser incorporado em outros dispositivos, (v) um telefone celular, dispositivo sem fio, aparelho telefônico, ou unidade móvel, (vi) etc.

A descrição anterior da revelação é fornecida para permitir que qualquer pessoa versada na técnica faça ou utilize a revelação. Várias modificações na revelação serão prontamente evidentes para aqueles versados na técnica, e os princípios genéricos definidos aqui podem ser aplicados a outras variações sem se afastar do espírito ou escopo da revelação. Desse modo, a revelação não pretende ser limitada aos exemplos descritos aqui porém deve ser acordado escopo mais amplo compatível com os princípios e características novas aqui reveladas.

REIVINDICAÇÕES

1. Equipamento **caracterizado** pelo fato de que compreende:

5 meios para suportar simultaneamente múltiplos aplicativos de multimídia;

meios para alocar uma porção configurável de recursos de armazenamento para cada aplicativo de multimídia para armazenar instruções, dados, e informações de estado para o aplicativo de multimídia; e

10 meios para atribuir dinamicamente unidades de processamento para os múltiplos aplicativos de multimídia conforme solicitado pelos aplicativos de multimídia, em que as unidades de processamento compreendem unidades de processamento diferentes que realizam tipos diferentes de
15 processamento para os múltiplos aplicativos de multimídia, e em que as unidades de processamento compreendem pelo menos um núcleo de unidade lógica aritmética (ALU), pelo menos um núcleo de função elementar, pelo menos um núcleo lógico e pelo menos um amostrador de textura.

20 2. Equipamento, de acordo com a reivindicação 1, **caracterizado** pelo fato de que os meios para alocar compreendem:

meios para alocar uma porção configurável de uma cache de instrução para cada aplicativo de multimídia para
25 armazenar instruções para o aplicativo de multimídia.

3. Equipamento, de acordo com a reivindicação 1, **caracterizado** pelo fato de que os meios para alocar compreendem:

meios para alocar uma porção configurável de
30 bancos de registrador para cada aplicativo de multimídia para armazenar dados para o aplicativo de multimídia.

4. Equipamento, de acordo com a reivindicação 1, **caracterizado** pelo fato de que compreende adicionalmente:

meios para receber de forma assíncrona encadeamentos provenientes dos múltiplos aplicativos de multimídia; e

meios para prover de forma assíncrona resultados
5 para os múltiplos aplicativos de multimídia.

5. Equipamento, de acordo com a reivindicação 1, **caracterizado** pelo fato de que os meios para suportar simultaneamente múltiplos aplicativos de multimídia, os meios para alocar e os meios para atribuir dinamicamente
10 unidades de processamento são um processador multimídia.

6. Equipamento, de acordo com a reivindicação 5, **caracterizado** pelo fato de que os recursos de armazenamento configuráveis compreendem uma cache de instrução para armazenar instruções para os múltiplos aplicativos de
15 multimídia, em que a cada aplicativo de multimídia é alocada uma porção configurável da cache de instrução.

7. Equipamento, de acordo com a reivindicação 5, **caracterizado** pelo fato de que os recursos de armazenamento configuráveis compreendem bancos de registrador para
20 armazenar dados para os múltiplos aplicativos de multimídia, em que a cada aplicativo de multimídia é alocada uma porção configurável dos bancos de registrador.

8. Equipamento, de acordo com a reivindicação 5, **caracterizado** pelo fato de que os recursos de armazenamento
25 configuráveis compreendem uma unidade de armazenamento para armazenar instruções ou dados para os múltiplos aplicativos de multimídia, a unidade de armazenamento sendo associada a uma memória virtual e uma memória física, em que a cada aplicativo de multimídia é alocada uma porção configurável
30 da memória virtual; e

pelo menos uma tabela para mapear a porção da memória virtual alocada a cada aplicativo de multimídia para uma porção correspondente da memória física.

9. Equipamento, de acordo com a reivindicação 5, **caracterizado** pelo fato de que o processador de multimídia compreende adicionalmente:

5 uma unidade de controle de carga para buscar instruções para cada aplicativo de multimídia, conforme solicitado, a partir de uma memória cache ou uma memória principal para armazenamento na porção dos recursos de armazenamento alocados ao aplicativo de multimídia.

10 10. Equipamento, de acordo com a reivindicação 5, **caracterizado** pelo fato de que as unidades de processamento operam independentemente e cada unidade de processamento é atribuível a qualquer um dentre os múltiplos aplicativos de multimídia em uma dada partição de tempo.

15 11. Equipamento, de acordo com a reivindicação 1, **caracterizado** pelo fato de que o processador multimídia é um dispositivo sem fio.

12. Método, **caracterizado** pelo fato de que compreende:

20 suportar simultaneamente múltiplos aplicativos de multimídia;

alocar uma porção configurável de recursos de armazenamento para cada aplicativo de multimídia para armazenar instruções, dados, e informações de estado para o aplicativo de multimídia; e

25 atribuir dinamicamente unidades de processamento para os múltiplos aplicativos de multimídia conforme solicitado pelos aplicativos de multimídia, em que as unidades de processamento compreendem unidades de processamento diferentes que realizam tipos diferentes de processamento para os múltiplos aplicativos de multimídia,
30 e em que as unidades de processamento compreendem pelo menos um núcleo de unidade lógica aritmética (ALU), pelo

menos um núcleo de função elementar, pelo menos um núcleo lógico e pelo menos um amostrador de textura.

13. Método, de acordo com a reivindicação 12, **caracterizado** pelo fato de que alocar compreende alocar uma porção configurável de uma cache de instrução para cada aplicativo de multimídia para armazenar instruções para o aplicativo de multimídia.

14. Método, de acordo com a reivindicação 12, **caracterizado** pelo fato de que alocar compreende alocar uma porção configurável de bancos de registrador para cada aplicativo de multimídia para armazenar dados para o aplicativo de multimídia.

15. Método, de acordo com a reivindicação 12, **caracterizado** pelo fato de que compreende adicionalmente receber de forma assíncrona encadeamentos provenientes dos múltiplos aplicativos de multimídia; e prover de forma assíncrona resultados para os múltiplos aplicativos de multimídia.

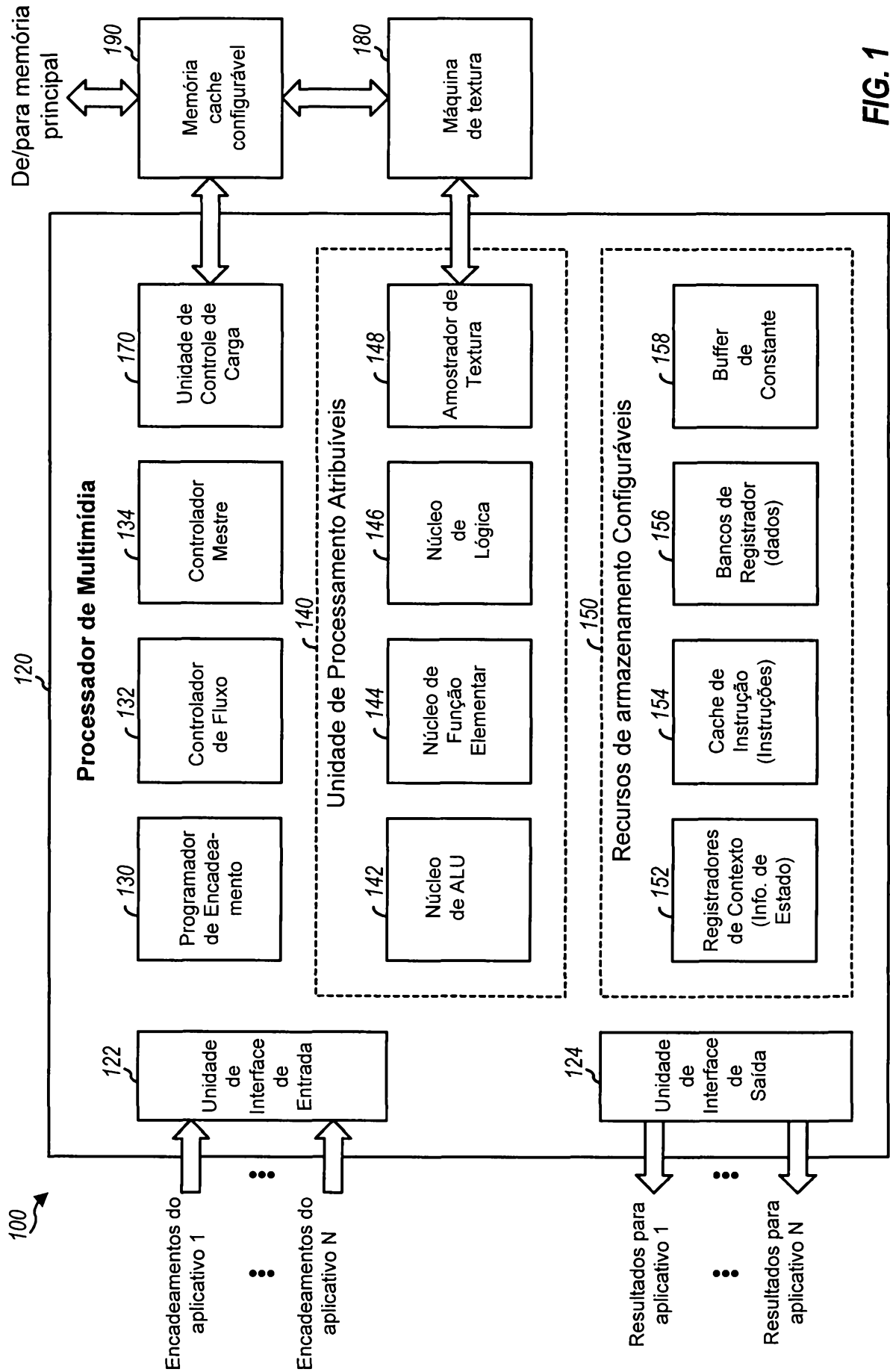


FIG. 1

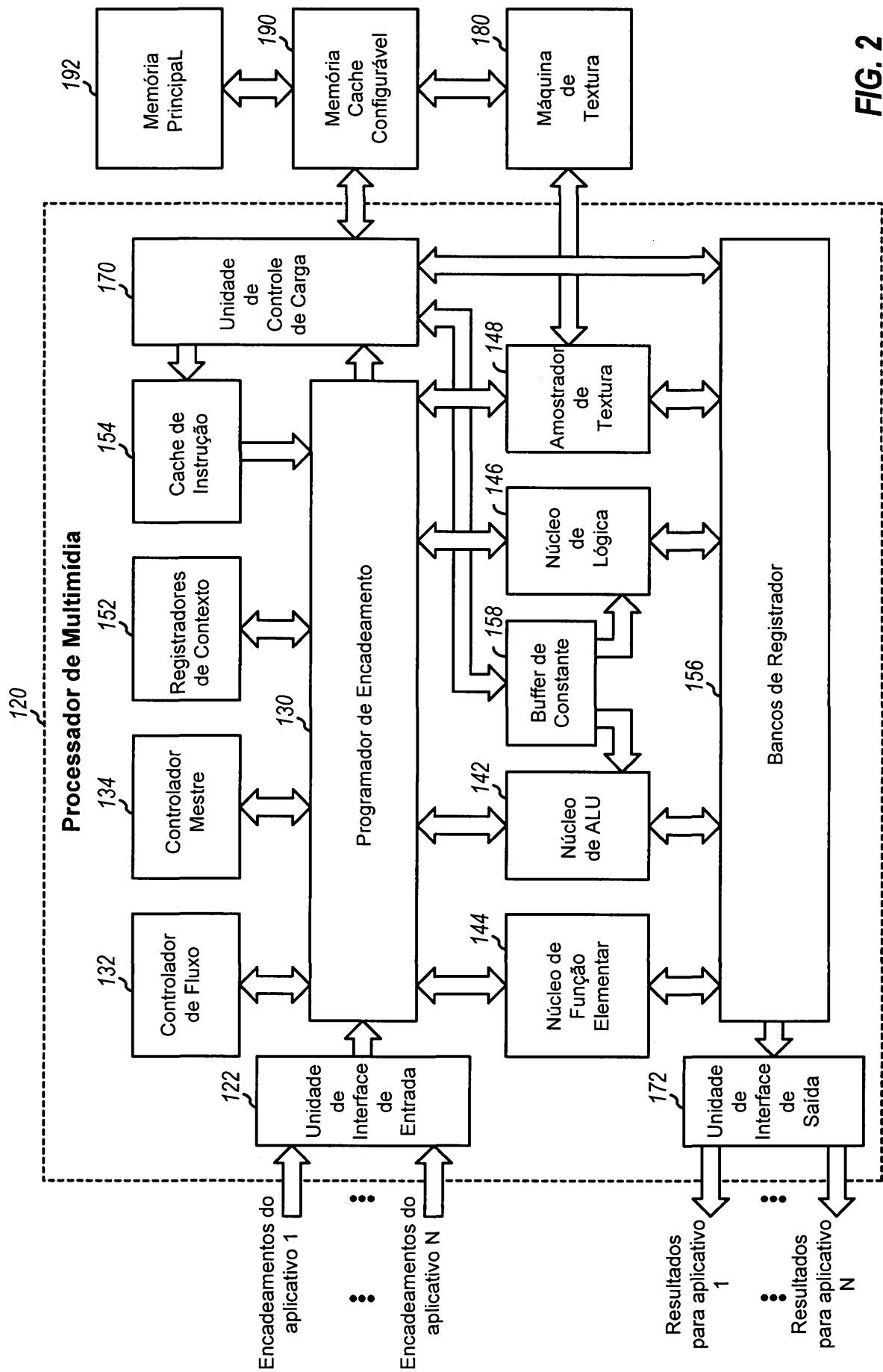


FIG. 2

150

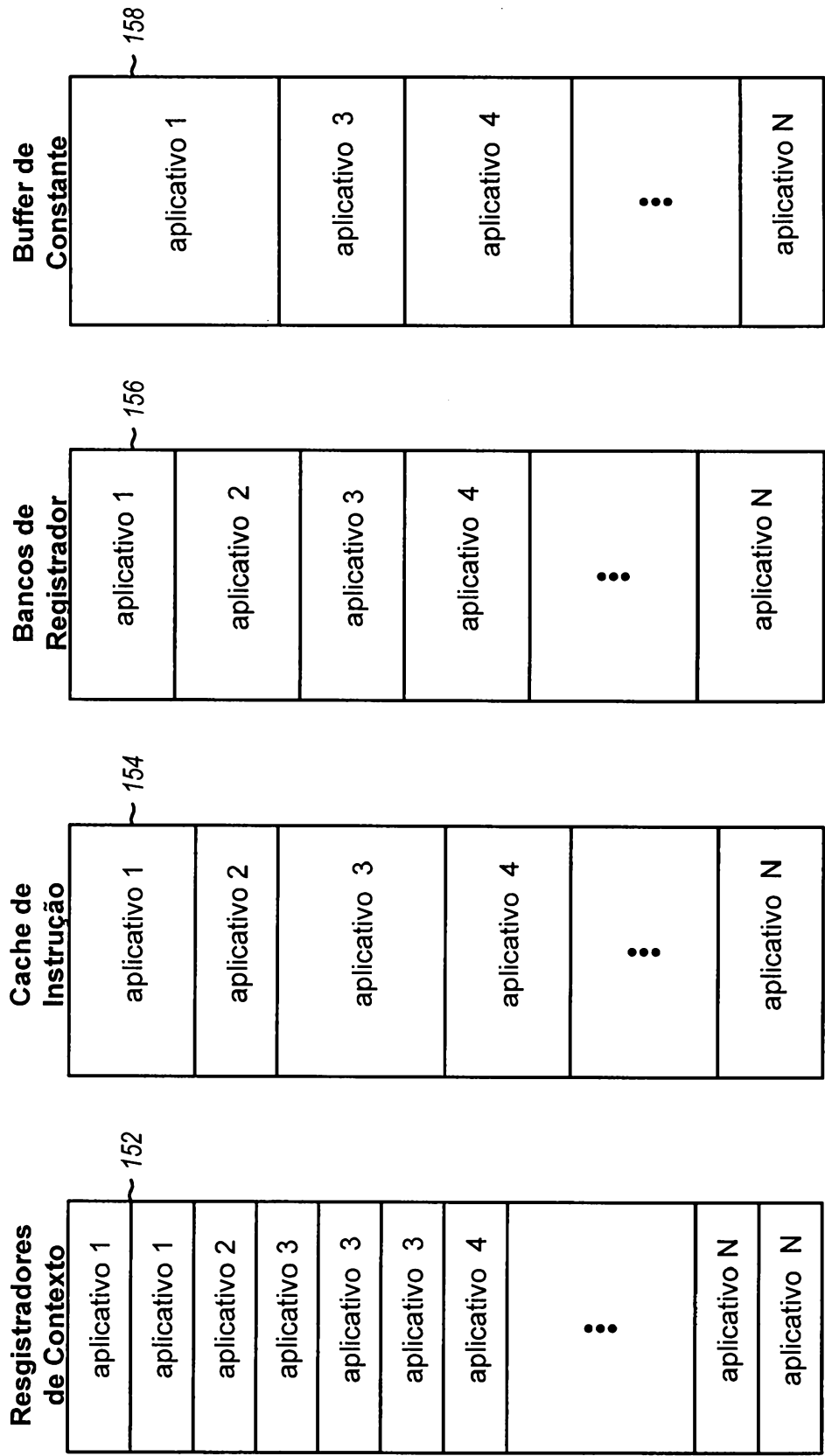


FIG. 3

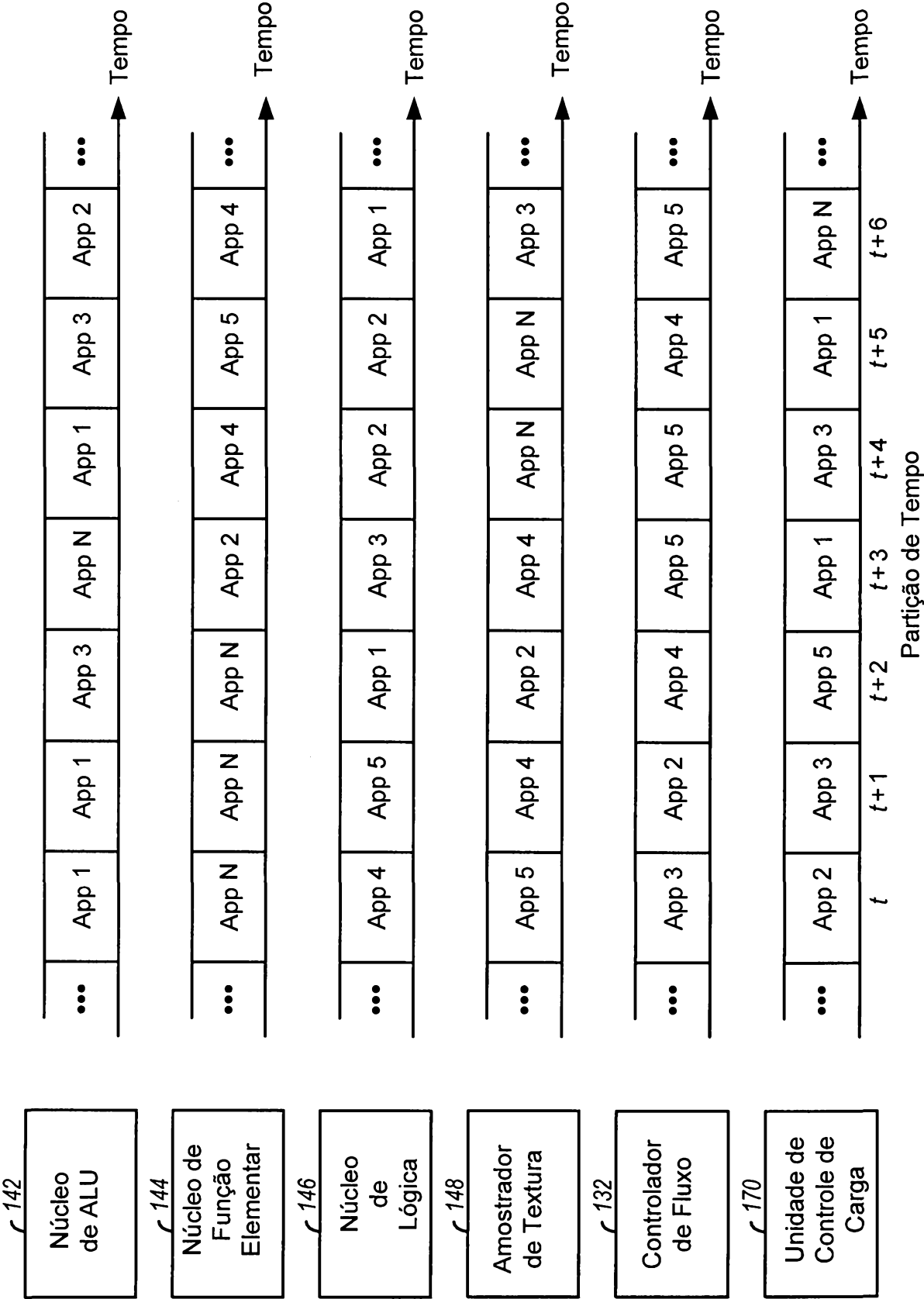
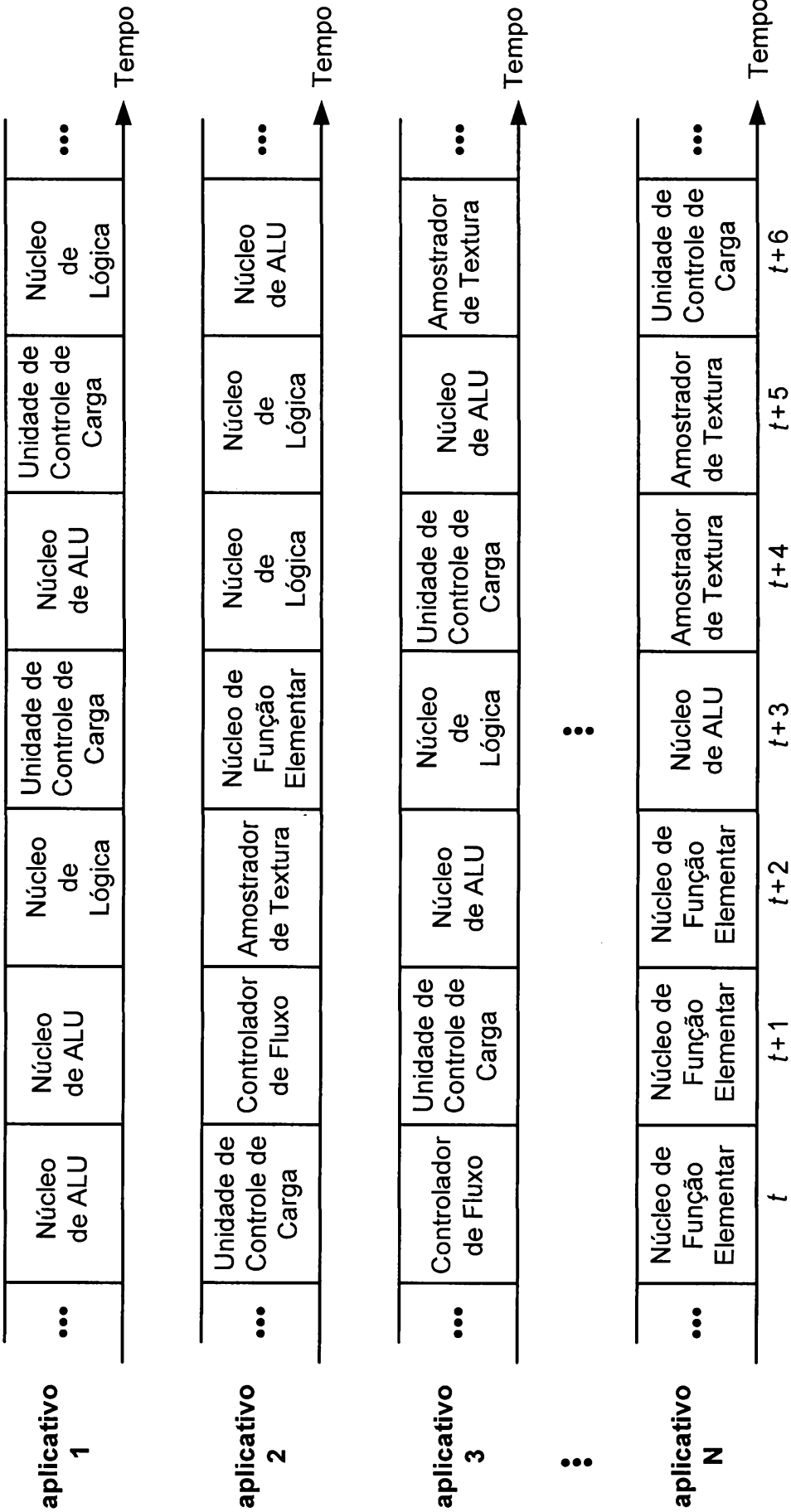


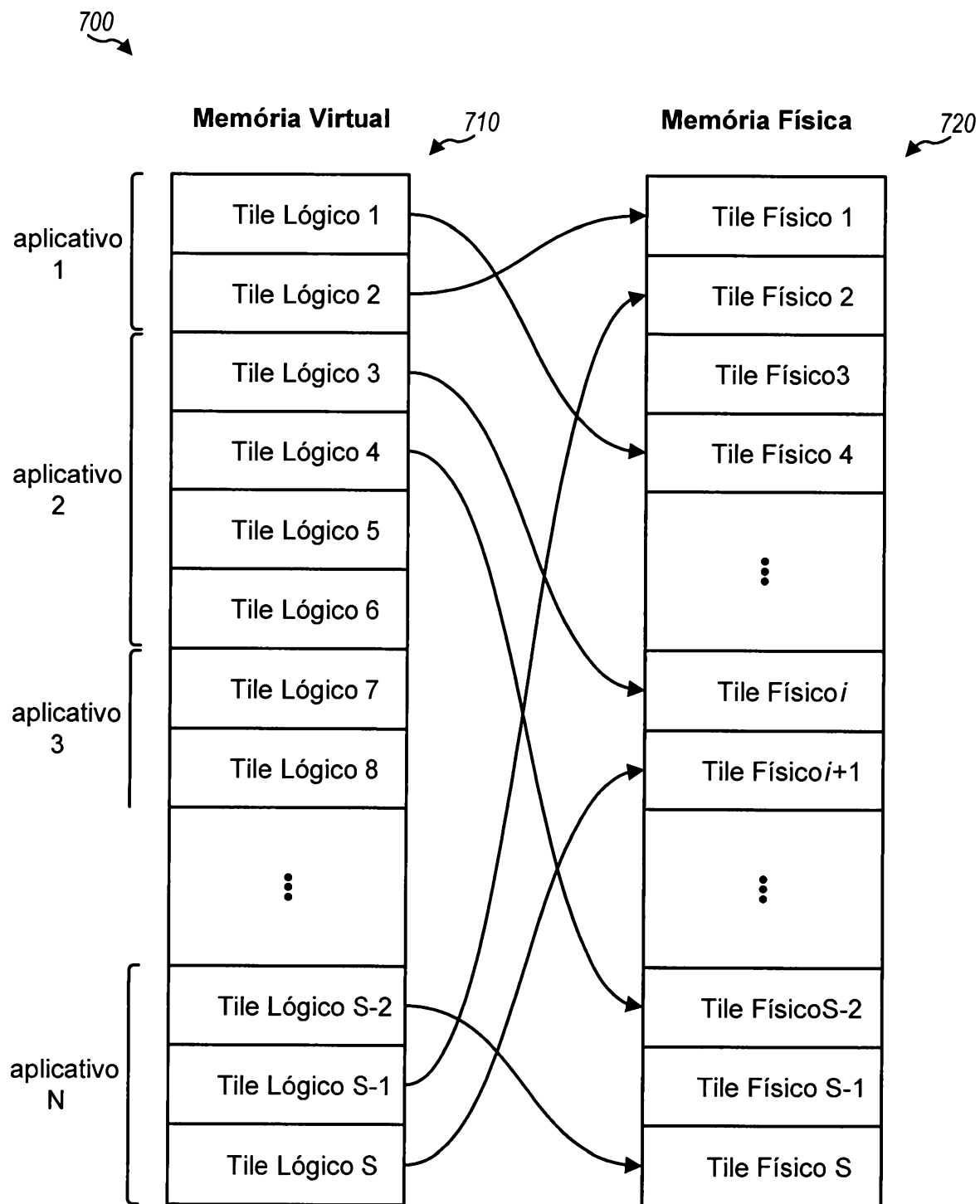
FIG. 4



Partição de Tempo

FIG. 5



**FIG. 7**

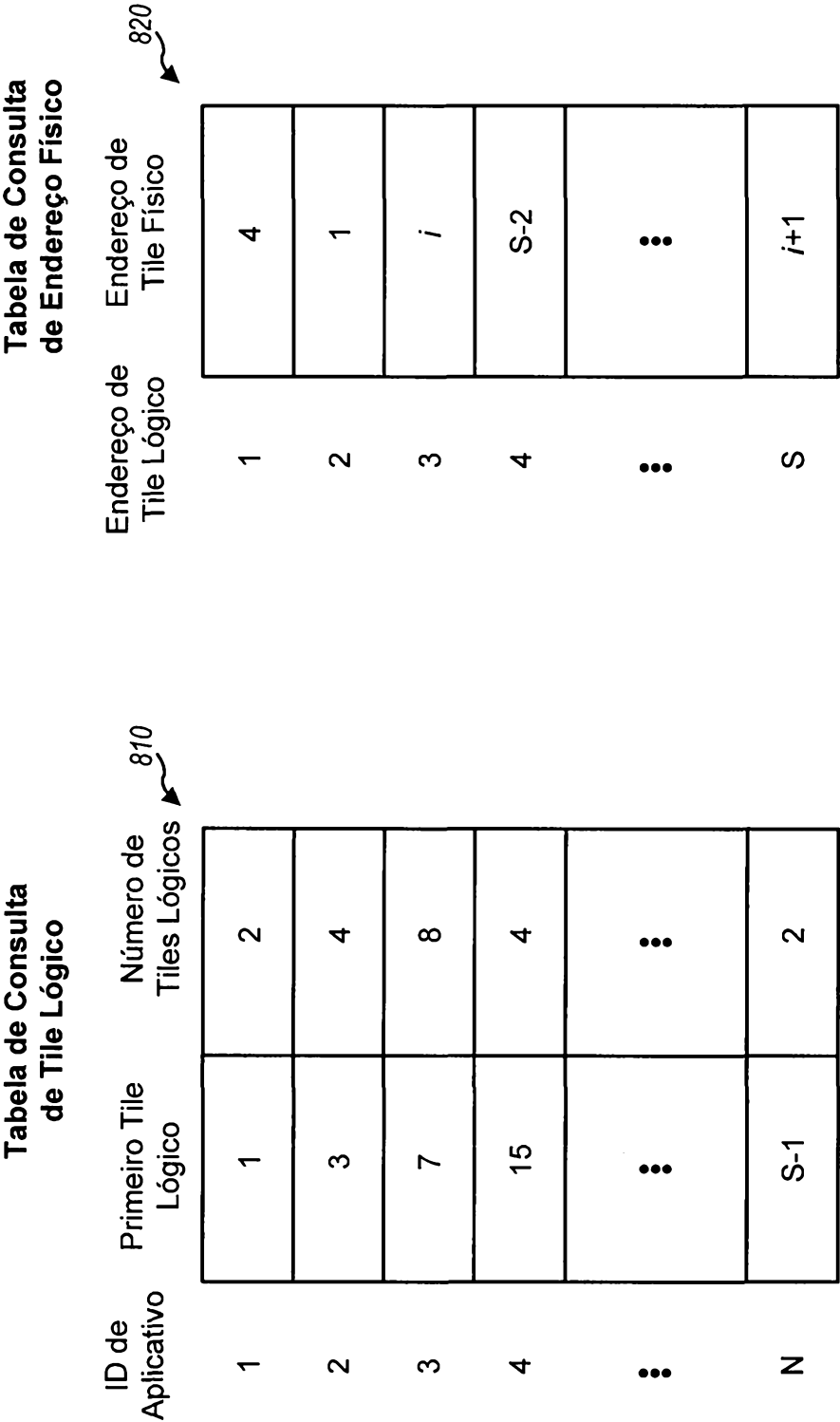
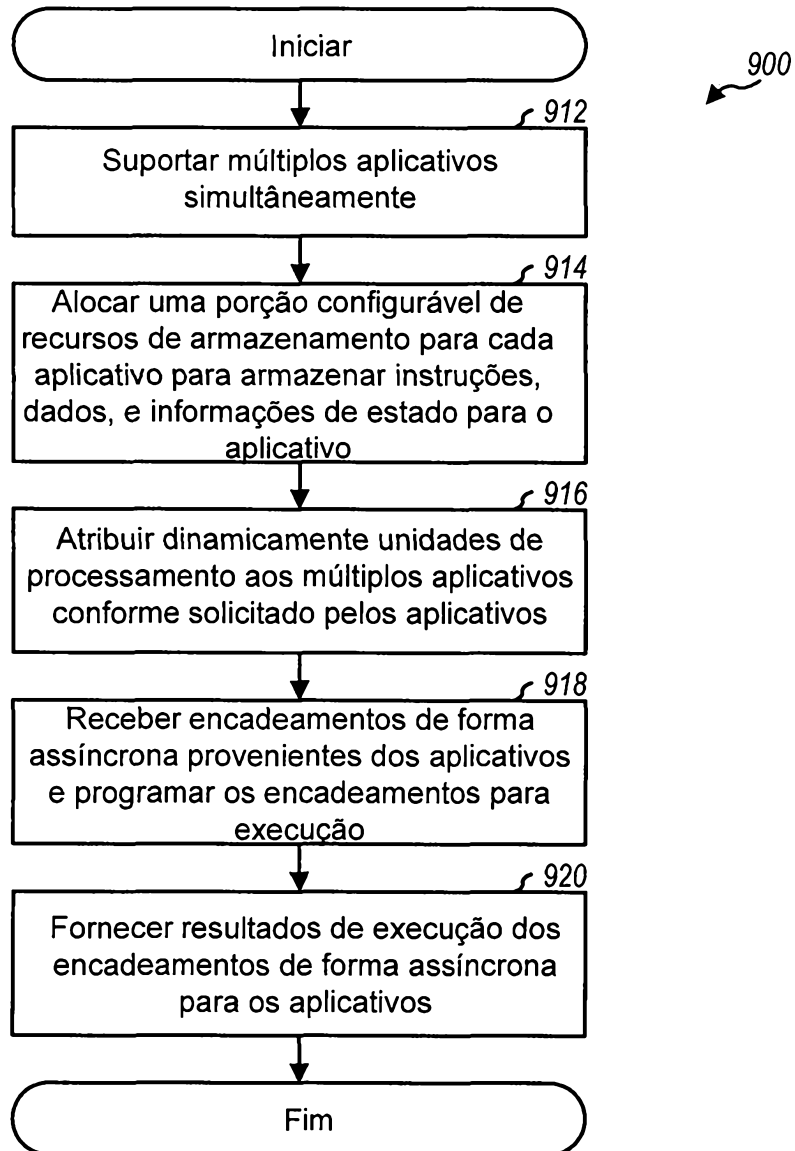


FIG. 8

**FIG. 9**

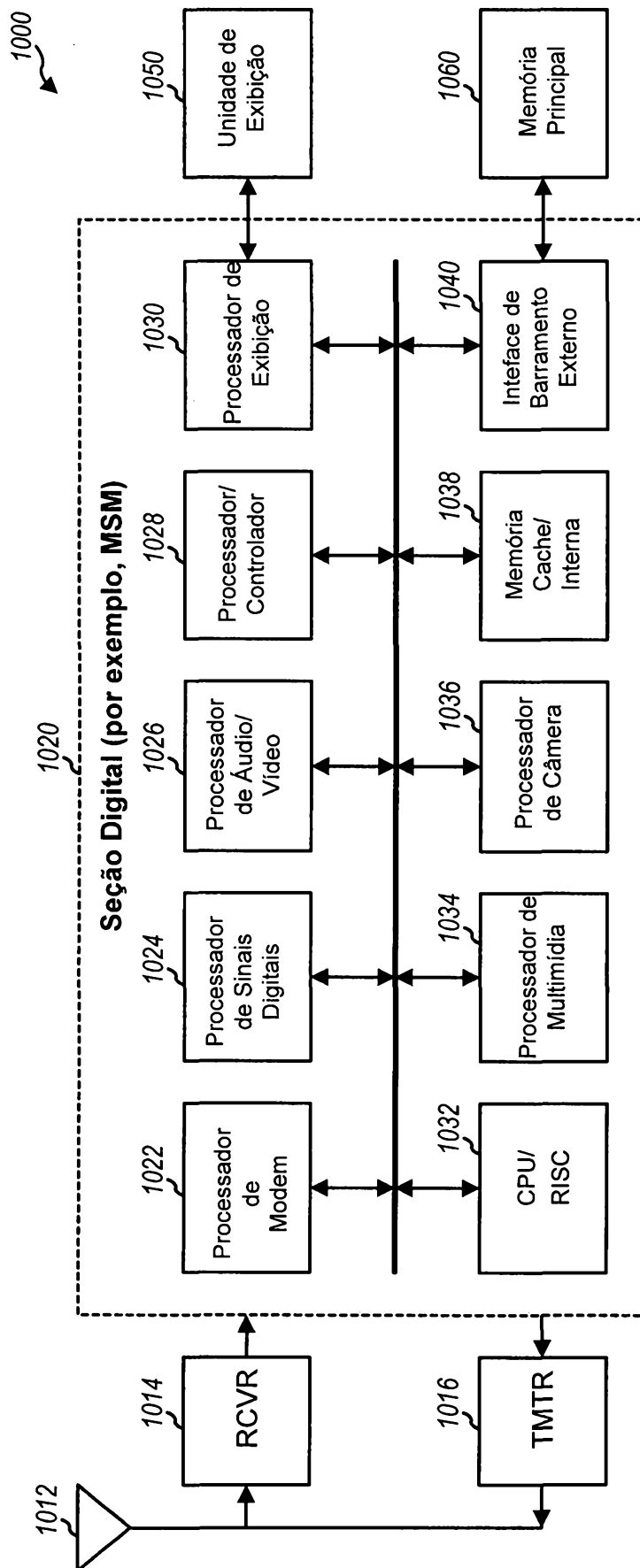


FIG. 10