



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

(11) BR 112015006535-0 B1



(22) Data do Depósito: 26/09/2013

(45) Data de Concessão: 29/11/2022

(54) Título: MÉTODO EXECUTADO POR UM DECODIFICADOR, DECODIFICADOR, MÉTODO EXECUTADO POR UM CODIFICADOR, CODIFICADOR, TERMINAL MÓVEL, E, NÓ DE REDE

(51) Int.Cl.: H04N 7/00.

(30) Prioridade Unionista: 28/09/2012 US 61/706,869.

(73) Titular(es): TELEFONAKTIEBOLAGET LM ERICSSON (PUBL).

(72) Inventor(es): JONATAN SAMUELSSON; RICKARD SJÖBERG.

(86) Pedido PCT: PCT EP2013070093 de 26/09/2013

(87) Publicação PCT: WO 2014/049066 de 03/04/2014

(85) Data do Início da Fase Nacional: 24/03/2015

(57) Resumo: DECODIFICAÇÃO E CODIFICAÇÃO DE IMAGENS DE UMA SEQUÊNCIA DE VÍDEO. A presente invenção refere-se a modalidades que reduzem o atraso na saída de imagens através da determinação, após uma imagem atual ter sido decodificada e armazenada em um armazenador temporário de imagens decodificadas, DPB, (125, 225, 325, 425, 525, 625), um número de imagens no DPB (125, 225, 325, 425, 525, 625) que estão marcadas como que precisam ser emitidas. Esse número é comparado, após uma imagem atual ter sido decodificada e armazenada no DPB (125, 225, 325, 425, 525, 625), com um valor derivado de ao menos um elemento de sintaxe presente ou a estar presente em um fluxo de bits (10) representando imagens de uma sequência de vídeo. Se esse número é maior do que o valor, uma imagem, que é a primeira imagem em ordem de saída, das imagens no DPB (125, 225, 325, 425, 525, 625) que estão marcadas como que precisam ser emitidas, é preferencialmente emitida e marcada como que não precisa ser emitida.

MÉTODO EXECUTADO POR UM DECODIFICADOR,
DECODIFICADOR, MÉTODO EXECUTADO POR UM CODIFICADOR,
CODIFICADOR, TERMINAL MÓVEL, E, NÓ DE REDE
CAMPO DA INVENÇÃO

[001] As presentes modalidades geralmente referem-se à decodificação e à codificação de imagens de uma sequência de vídeo, e em particular à emissão ou à transferência de imagens de um armazenador temporário de imagem decodificada em conjunto com a decodificação e codificação de imagens.

FUNDAMENTOS DA INVENÇÃO

COMPRESSÃO DE VÍDEO H.264

[002] H.264 (Codificação de Vídeo Avançada por Especialistas de Imagem em Movimento do Grupo 4) (MPEG-4 AVC)) é o padrão de codificação de vídeo mais moderno existente. Ele consiste de um esquema de codificação de vídeo híbrido baseado em blocos que explora as redundâncias temporais e espaciais. O padrão H.264/AVC é definido em um texto de especificação que contém muitos processos de decodificação que serão executados na sequência especificada, de modo que um decodificador seja compatível com o padrão. Não há exigências sobre o codificador, mas frequentemente há o caso em que o codificador também executa a maioria dos processos de modo a conseguir uma boa eficiência de compressão.

[003] H.264/AVC define um armazenador temporário de imagem decodificada (DPB) que armazena imagens decodificadas depois de terem sido decodificadas. Isso significa que exige-se que o decodificador use uma quantidade definida de memória de modo a decodificar uma sequência. O DPB contém imagens que são usadas como referência durante a decodificação de imagens futuras. "Usado para referência" aqui significa que uma imagem particular é utilizada para predição quando outra imagem é decodificada. Os valores de pixel da imagem que é utilizada para referência podem então ser

utilizados para prever os valores de pixel da imagem que está sendo atualmente decodificada. Isso é também chamado de predição Inter. O DPB contém adicionalmente imagens que estão à espera para a saída. "Saída" aqui significa a função onde um decodificador emite uma imagem fora do decodificados. A especificação H.264 descreve como um fluxo de bits é convertido em imagens que são então emitidas, ver Fig. 1. As imagens emitidas podem, por exemplo, ser exibidas ou gravadas em disco.

[004] Uma razão comum para uma imagem no DPB estar à espera de saída é que há uma imagem que não foi decodificada ainda que será emitida antes da imagem.

[005] A Figura 2 mostra um exemplo de três imagens: A, B, e C. A ordem de decodificação é a ordem pela qual as imagens em formato comprimido são fornecidas ao decodificador. Essa é tipicamente a mesma ordem na qual as imagens são codificadas pelo codificador. A Figura 2 mostra que a ordem de decodificação neste exemplo é A, B e C. A ordem de saída é a ordem na qual as imagens decodificadas são emitidas. A ordem de saída não tem que ser a mesma da ordem de decodificação como ilustrado no exemplo da Figura 2, onde a ordem de saída é A, C, B. As setas na figura mostram quais imagens são usadas como referência para cada imagem: imagem A é usada como referência para ambas imagens B e C.

[006] Na Figura 2, a imagem C é decodificada após a B, mas emitida antes dela. Quando a imagem B foi decodificada, ela pode não ser emitida imediatamente, visto que a imagem C não foi decodificada ainda e tem que ser emitida antes da imagem B. Portanto, a imagem B tem que ser armazenada no DPB depois de ter sido decodificada, mesmo que não seja usada como referência por qualquer outra imagem. Ao decodificar a imagem C, a imagem A também precisa estar presente no DPB, visto que imagem C usa imagem A como referência.

[007] A ordem de saída é controlada sinalizando um valor

PictureOrderCount (POC). Há elementos de sintaxe no fluxo de bits que transportam o POC de cada imagem e esses valores são usados para definir a ordem de saída das imagens.

[008] Para manter o rastreamento do DPB, o H.264/AVC contém três processos que ocorrem após a decodificação de uma imagem: o processo de marcação de imagem, o processo de emissão da imagem e do processo de liberação.

[009] O processo de marcação de imagem marca as imagens ou como "usadas como referência" ou como "não usadas como referência". Uma imagem marcada como "usada como referência" está disponível como referência, o que significa que uma imagem subsequente na ordem de decodificação pode usar a imagem como referência em seus processos de decodificação. A imagem marcada como "não usada como referência" não pode ser usada como referência por imagens subsequentes. Esse processo é controlado pelo codificador através do fluxo de bits. Há uma sintaxe opcional no fluxo de bits H.264/AVC que, quando presente, indica quais imagens marcar como "não usadas como referência". Essa operação é frequentemente chamada de a operação de controle de gerenciamento de memória (MMCO). Se não houver nenhuma sintaxe MMCO opcional, um mecanismo de primeiro que entra é o primeiro que sai é definido, chamado de o processo de "janela deslizante". O processo de janela deslizante significa que quando a última imagem decodificada resulta em muitas imagens no DPB, a imagem mais antiga na ordem de decodificação é marcada automaticamente como "não usada como referência".

[0010] O processo de emissão da imagem, que é executado após o processo de marcação de imagem, marca as imagens como "precisa ser emitida" ou como "não precisa ser emitida". Uma imagem marcada como "precisa ser emitida" não foi emitida ainda enquanto uma imagem marcada como "não precisa ser emitida" foi emitida e não está mais esperando por

saída. O processo de emissão de imagem também emite imagens. Isso significa que o processo seleciona as imagens que são marcadas como "precisam ser emitidas", emite-as e então as marca como "não precisam ser emitidas". O processo de emissão de imagem determina em qual ordem as imagens são emitidas. Nota-se que o processo de emissão de imagem pode emitir e marcar zero, uma ou muitas imagens após uma imagem particular ter sido decodificada.

[0011] Após esses dois processos terem sido invocados pelo decodificador, o processo de liberação é invocado. As imagens que são marcadas tanto como "não usadas como referência" quanto como "não precisam ser emitidas" são esvaziadas e removidas do DPB. Isso significa às vezes que uma das áreas de armazenamento de imagem DPB foi liberada.

[0012] O tamanho do DPB em H.264/AVC é limitado. Isso indica que o número de imagens que podem ser armazenadas porque estão esperando por saída e disponibilizadas como referência é limitado. O max_dec_frame_buffering variável denota o tamanho do DPB, às vezes chamado de o número de áreas de armazenamento de imagem que estão no DPB. O codificador tem que garantir que o tamanho do DPB nunca sobrecarregue.

[0013] Os três processos são descritos no padrão. Isso significa que o decodificador é controlado pelo codificador e então o decodificador não tem qualquer liberdade com relação à ordem de saída. Tudo é determinado pelo processo de emissão de imagem e pelos elementos relacionados no fluxo de bits enviado pelo codificador. É mostrado na Figura 3 um fluxograma simplificado para as etapas de decodificação de H.264/AVC.

[0014] O processo de emissão de imagem em H.264 define a ordem na qual as imagens devem ser emitidas. Um decodificador que emite imagens na ordem correta é compatível com a ordem de saída. Um decodificador pode seguir o processo de emissão de imagem descrito em H.264, mas é às vezes possível usar a variável num_reorder frames para emitir imagens mais

precoces do que o que é fornecido pelo processo de emissão de imagem. `num_reorder frames` indica o número máximo de imagens que precedem qualquer imagem na ordem de decodificação e seguem a ordem de saída.

[0015] A Figura 4 mostra um exemplo em que a imagem B foi decodificada. Mas a imagem B não pode ser emitida, uma vez que não se sabe se a imagem C será emitida antes ou após a imagem B. Se o codificador decidiu que a ordem de saída é a mesma da ordem de decodificação, ele pode indicar um valor `num_reorder frames` igual 0 ao decodificador. O codificador prometeu assim que a imagem C, no exemplo, será emitida após a imagem B e um decodificador pode emitir a imagem B imediatamente quando ela foi decodificada. Neste caso, quando `num_reorder frames` é igual a 0, não existe nenhum atraso de reordenamento adicional no decodificador. Se `num_reorder frames`, no exemplo, é configurado para ser igual a 1, é possível que a imagem C seja emitida antes da imagem B. Com `num_reorder frames` igual a 1, há um atraso de reordenamento adicional de 1 imagem, com `num_reorder frames` igual a 2, o atraso de reordenamento é de 2 imagens e assim por diante.

COMPRESSÃO DE VÍDEO HEVC

[0016] A Codificação de Vídeo de Alta Eficiência (HEVC), também chamada de H.265, é um padrão de codificação de vídeo desenvolvido em uma Equipe Colaborativa Conjunta — Codificação de Vídeo (JCT-VC). JCT-VC é um projeto colaborativo entre MPEG e a Sessão de Padronização de Telecomunicação da União Internacional de Telégrafos (ITU-T). HEVC inclui uma série de novas ferramentas e é consideravelmente mais eficiente do que H.264/AVC. HEVC também define um `temporal_id` para cada imagem, correspondente à camada temporal a que a imagem pertence. As camadas temporais são ordenadas e têm a propriedade de que uma camada temporal mais inferior nunca dependerá de uma camada temporal superior. Assim, as camadas temporais superiores podem ser removidas sem afetar as camadas temporais inferiores. A remoção de camadas temporais pode ser chamada de

escalonamento temporal. Um fluxo de bits HEVC contém um elemento de sintaxe, `max_sub_layers_minus1`, que especifica o número máximo de camadas temporais que podem estar presentes no fluxo de bits. Um decodificador pode decodificar todas as camadas temporais ou decodificar somente um subconjunto das camadas temporais. A camada temporal mais alta que o decodificador decodifica efetivamente é chamada de a subcamada temporal mais alta e pode ser igual ou menor do que o número máximo de camadas como especificado por `max_sub_layers_minus1`. O decodificador decodifica então todas as camadas que são iguais ou menores do que a subcamada temporal mais alta. A subcamada temporal mais alta pode ser configurada por dispositivos externos.

[0017] Nota-se que a descrição acima não é específica para camadas temporais, mas também é válida para outros tipos de camadas, tais como as camadas espaciais e as camadas de qualidade, etc. A camada temporal que o decodificador então decodifica é chamada de a camada decodificada mais alta.

[0018] O fluxo de decodificação de HEVC é ligeiramente diferente de H.264/AVC. HEVC tem um DPB, um processo de marcação de imagem que marca as imagens como "usadas como referência" e "não usadas como referência", um processo de emissão de imagem que marca as imagens como "precisam ser emitidas" e "não precisam ser emitidas" e um processo de liberação. Como H.264/AVC, HEVC também usa valores POC para definir a ordem de saída de imagem. Um valor POC está em HEVC representado pela variável `PicOrderCntVal`, onde as imagens são emitidas em ordem crescente `PicOrderCntVal`.

[0019] HEVC, entretanto, não tem MMCO ou processo de janela deslizante. Em vez disso, HEVC especifica que uma lista das imagens que são marcadas como "usadas como referência" é enviada explicitamente em cada cabeçalho de fatia. A marcação de imagem em HEVC usa essa lista e garante

que todas as imagens no DPB, que são listadas, são marcadas como "usadas como referência" e que todas as imagens no DPB que não são listadas são marcadas como "não usadas como referência". A lista é chamada de o conjunto de imagens de referência (RPS) e enviar uma em cada cabeçalho de fatia significa que o estado da marcação de referência no DPB é explícito e repetido em cada fatia, o que não é o caso em H.264/AVC.

[0020] Como os RPSs são usados em HEVC, o processo de marcação de imagem, o processo de emissão de imagem e o processo de liberação são todos executados após a análise do primeiro cabeçalho de fatia de uma imagem, ver Figura 5.

[0021] A funcionalidade `num_reorder frames`, como descrita para H.264/AVC, também está presente em HEVC. Um fluxo de bits HEVC contém um elemento de sintaxe para cada camada temporal, denotada `max num_reorder pics [i]`, onde `i` é a camada temporal. A função de `max num_reorder pics [i]` é a mesma função de `num_reorder frames [i]`, mas cada palavra de código indica o número máximo permitido de imagens na mesma camada temporal ou em camada temporal inferior que precede uma imagem na ordem de decodificação e sucede essa imagem na ordem de saída.

[0022] Considera-se o exemplo da Figura 6 em que a ordem de decodificação é A, B, C, D, E e a ordem de saída é A, D, C, E, B. Esta é uma estrutura de imagens que usa as camadas temporais onde as imagens A e B pertencem à camada temporal mais baixa (0), a imagem C pertence a uma camada temporal intermediária (camada 1) e as imagens D e E pertencem à camada temporal mais alta (camada 2). As setas na figura mostram quais imagens serão usadas como referência por outras imagens. Por exemplo, a imagem A é usada como referência pela imagem B, uma vez que há uma seta da imagem A para a imagem B. O melhor uso de `max num_reorder pics` em HEVC é configurá-lo para um valor mais baixo possível para reduzir ao máximo o atraso de emissão. Os valores mais baixos possíveis de `max`

num_reorder pics para cada camada temporal são mostrados na Figura 6. O motivo pelo qual ele é igual a 0 para a camada mais baixa é porque não há nenhuma imagem na camada 0 que preceda qualquer imagem na ordem de decodificação, mas segue na ordem de saída. Para a camada 1, tem-se a imagem B que precede a imagem C na ordem de decodificação, mas que a segue na ordem de saída, e para a camada 2 tem-se as imagens B e C que tanto precede a imagem D na ordem de decodificação, mas a segue na ordem de saída.

[0023] Se um decodificador sabe que ele somente decodificará a camada temporal 0, ele poderia emitir potencialmente a imagem B assim que tenha sido decodificada, mas se o decodificador decodifica todas as camadas, ele não pode. Ele poderia, então, ter que esperar até que existam duas imagens decodificadas que seguem B na ordem de saída.

[0024] JCTVC-K0030 v3, Proposed Editorial Improvement for High efficiency vídeo coding (HEVC) Text Specification Draft 8, B. Bross e outros, JCT-VC de ITU-T SG16 WP3 e ISO/IEC JTC11SC29/WG11 11¹ Reunião, Xangai, 10 a 19 de outubro de 2012, publicado em 12 de setembro de 2012 discute o uso de max num_reorder pics na seção 7.4.2.1 na página 62 e na seção 7.4.2.2 na página 64.

NO_OUTPUT_OF_PRIOR_PICS_FLAG

[0025] Ambos os fluxos de bits H.264 e HEVC especificam um sinalizador chamado no_output_of_prior pics flag. Esse sinalizador está presente no cabeçalho de fatia de imagens de acesso aleatório (RAP). As imagens de acesso aleatório são imagens a partir das quais é possível sintonizar em um fluxo. Eles garantem que a decodificação de imagens futuras pode ser feita corretamente se um decodificador começa a decodificação a partir do ponto de acesso aleatório. O decodificador não tem de ser alimentado com quaisquer dados contendo imagens que precedem a imagem de acesso aleatório em ordem de decodificação para sintonizar-se ao

trabalho.

[0026] O no-output – of prior pics_flag especifica como as imagens previamente decodificadas no armazenador temporário de imagem decodificada são tratadas após a decodificação de uma imagem de acesso aleatório. Em resumo, se no_output_of prior_pics flag é igual a 1, nenhuma imagem no DPB que é marcada como "precisa ser emitida" deveria ser emitida, mas se no_output_of prior_pics_flag é igual a 0, elas deveriam ser emitidas.

[0027] Considera-se a Figura 7 que mostra um exemplo onde max_num_reorder_pics é 0 e a imagem C é uma imagem de acesso aleatório com no_output_of prior_pics flag igual a 1. Em H.264, seria possível emitir a imagem B imediatamente após ela ter sido decodificada. Este não é o caso na especificação HEVC atual desde o decodificador não sabe imediatamente depois que a imagem B foi decodificada se a imagem C é uma imagem RAP com no_output_of prior_pics flag igual a 1 ou não. Se a imagem C não é tal imagem, a imagem B poderia ser emitida imediatamente após ela ter sido decodificada. Mas se a imagem C é de fato uma imagem RAP com no_output_of prior_pics_flag igual a 1, a imagem B não deveria ser emitida, uma vez que a imagem B é marcada como "precisa ser emitida" quando o cabeçalho de fatia da imagem C é decodificado.

[0028] Como o processo de emissão de imagem em HEVC é feito quando o cabeçalho de fatia é analisado e no_output_of prior_pics_flag é uma característica importante, há um maior atraso de emissão no padrão HEVC atual do que em H.264/AVC.

[0029] A informação de uso de no_output_of prior_pics flag é descrita na seção 7.4.7.1 na página 75 e na seção C.5.2 na página 26 em JCTVC-K0030 v3.

[0030] A vantagem usando RPSs em HEVC é que é muito mais resiliente a erro em comparação com o método H.264/AVC. também, a

escalabilidade temporal é mais simples. Um problema com a solução HEVC é que ela introduz atraso adicional em relação à emissão da imagem em relação a H.264/AVC. Em H.264/AVC, as imagens podem ser emitidas após a imagem ter sido decodificada. Em HEVC, o decodificador tem que esperar que o cabeçalho de fatia da próxima imagem seja analisado até que as imagens sejam emitidas. Isso provoca um atraso.

[0031] Portanto, existe uma necessidade de resolver as deficiências da técnica anterior de codificação de vídeo e problemas particulares de atraso que podem ocorrer na codificação de vídeo da técnica anterior.

SUMÁRIO DA INVENÇÃO

[0032] É um objetivo geral fornecer uma decodificação e codificação aprimoradas de imagens de uma sequência de vídeo.

[0033] É um objetivo particular fornecer tal decodificação e codificação que possibilitem baixo atraso de emissão para imagens.

[0034] Esses e outros objetivos são alcançados por modalidades descritas aqui.

[0035] Um aspecto das modalidades refere-se a um método executado por um decodificador. O método compreende determinar, após uma imagem atual ter sido decodificada e armazenada em um armazenador temporário de imagem decodificada (DPB), um número de imagens no DPB que estão marcadas como que precisam ser emitidas. O método também compreende comparar o número com um valor `sps_max_num_reorder_pics[HighestTid]`. `HighestTid` especifica uma camada mais alta que é decodificada pelo decodificador de uma sequência de vídeo. O método compreende ainda emitir uma imagem, que é uma primeira imagem em ordem de saída das imagens no DPB que são marcadas como que precisam ser emitidas, se o número é maior do que o valor. A imagem é também marcada como que não precisa ser emitida se o número é maior do que o valor.

[0036] Um aspecto relacionado das modalidades define um

decodificador configurado para determinar, após uma imagem atual de um fluxo de bits representando imagens de uma sequência de vídeo ter sido decodificada e armazenada em um DPB, um número de imagens no DPB que estão marcadas como que precisam ser emitidas. O decodificador é também configurado para comparar o número com um valor `sps_max_num_reorder pics[HighestTid]`. O decodificador é ainda configurado para emitir uma imagem, que é uma primeira imagem em ordem de saída, das imagens no DPB que estão marcadas como que precisam ser emitidas e marcar a imagem como que não precisa ser emitida se o número é maior do que o valor.

[0037] Outro aspecto relacionado das modalidades define um decodificador compreendendo um módulo de determinação de número para determinar, após uma imagem de um fluxo de bits representando as imagens de uma sequência de vídeo ter sido decodificada e armazenada no DPB, um número de imagens no DPB que estão marcadas como que precisam ser emitidas. O decodificador também compreende um módulo de comparação para comparar o número com um valor `sps_max_num_reorder pics[HighestTid]`. O decodificador compreende ainda um módulo de emissão para emitir uma imagem, que é uma primeira imagem em ordem de saída, das imagens no DPB que estão marcadas como que precisam ser emitidas se o número é maior do que o valor. O decodificador adicionalmente compreende um módulo de marcação para marcar a imagem como que não precisa ser emitida se o número é maior do que o valor.

[0038] Outro aspecto da modalidade refere-se a um método executado por um codificador. O método compreende determinar, após uma imagem atual ter sido decodificada e armazenada em um DPB, um número de imagens no DPB que estão marcadas como que precisam ser emitidas. O método também compreende comparar o número com um valor `sps_max_num_reorder pics[HighestTid]`. O método compreende ainda marcar uma imagem que é uma primeira imagem em ordem de saída das

imagens no DPB que estão marcadas como que precisam ser emitidas como que não precisam ser emitidas se o número é maior do que o valor.

[0039] Um aspecto relacionado das modalidades define um codificador configurado para determinar, após uma imagem atual ter sido decodificada e armazenada em um DPB, um número de imagens no DPB que estão marcadas como que precisam ser emitidas. O codificador é também configurado para comparar o número com um valor `sps_max_num_reorder_pics[HighestTid]`. O codificador é ainda configurado para marcar uma imagem, que é uma primeira imagem em ordem de saída, das imagens no DPB que estão marcadas como que precisam ser emitidas como que não precisam ser emitidas se o número é maior do que o valor.

[0040] Outro aspecto relacionado das modalidades define um codificador compreendendo um módulo de determinação de número para determinar, após uma imagem atual ter sido decodificada e armazenada em um DPB, um número de imagens no DPB que estão marcadas como que precisam ser emitidas. O codificador também compreende um módulo de comparação para comparar o número com um valor `sps_max_num_reorder_pics[HighestTid]`. O codificador compreende ainda um módulo de marcação para marcar uma imagem, que é uma primeira imagem em ordem de saída, das imagens no DPB que estão marcadas como que precisam ser emitidas como que não precisam ser emitidas se o número é maior do que o valor.

[0041] Um aspecto adicional das modalidades refere-se a um método executado por um decodificador. O método compreende analisar um cabeçalho de fatia de uma imagem atual a ser decodificada de uma sequência de vídeo. O método também compreende determinar um conjunto de imagens de referência (RPS) para a imagem atual com base no cabeçalho de fatia analisado. O método compreende ainda marcar todas as imagens em um DPB que não estão presentes no RPS como não usadas como referência. Zero, uma ou muitas imagens, marcadas como que precisam ser emitidas, das imagens

no DPB são emitidas e marcadas como que não precisam ser emitidas. O método também compreende esvaziar qualquer imagem, do DPB, marcada como não usada como referência e que não precisa ser emitida das imagens no DPB. O método compreende ainda decodificar a imagem atual e determinar um número de imagens no DPB que estão marcadas como que precisam ser emitidas. O número é comparado com um valor derivado de ao menos um elemento de sintaxe presente em um fluxo de bits representando imagens da sequência de vídeo. Se o número é maior do que o valor, uma imagem, que é uma primeira imagem em ordem de saída, das imagens no DPB que estão marcadas como que precisam ser emitidas é emitida e marcada como que não precisa ser emitida. Em uma modalidade, determinar o número de imagens, comparar o número, emitir a imagem e marcar a imagem são executados após decodificar a imagem atual.

[0042] Um aspecto relacionado das modalidades define um decodificador configurado para analisar um cabeçalho de fatia de uma imagem atual a ser decodificada de um fluxo de bits representando imagens de uma sequência de vídeo e determinar um RPS para a imagem atual com base no cabeçalho de fatia analisado. O decodificador é também configurado para marcar todas as imagens em um DPB que não estão presentes no RPS como não usadas como referência. O decodificador é ainda configurado para emitir zero, uma ou muitas imagens, marcadas como que precisam ser emitidas, das imagens no DPB e para marcar zero, uma ou muitas imagens como que não precisam ser emitidas. O decodificador é adicionalmente configurado para esvaziar, a partir do DPB, qualquer imagem marcada como não usada como referência e como que não precisa ser emitida das imagens no DPB. Nesta modalidade, o decodificador é também configurado para decodificar a imagem atual. O decodificador é ainda configurado para determinar um número de imagens no DPB que estão marcadas como que precisam ser emitidas. O decodificador é ainda configurado para comparar o

número com um valor derivado de ao menos um elemento de sintaxe presente no fluxo de bits. Nesta modalidade, o decodificador é configurado para emitir uma imagem, que é uma primeira imagem em ordem de saída, das imagens no DPB que estão marcadas como que precisam ser emitidas e marcar a imagem como que não precisa ser emitida se o número é maior do que o valor. O decodificador é preferencialmente configurado para determinar o número de imagens, comparar o número, emitir a imagem, e marcar a imagem após decodificar a imagem atual.

[0043] Outro aspecto relacionado das modalidades define um decodificador compreendendo um módulo de análise para analisar um cabeçalho de fatia de uma imagem atual a ser decodificada de um fluxo de bits representando imagens de uma sequência de vídeo. O decodificador compreende ainda um módulo de determinação de conjunto de imagens de referência para determinar um RPS para a imagem atual com base no cabeçalho de fatia analisado. O decodificador também compreende um módulo de marcação para marcar todas as imagens em um DPB que não estão presentes no RPS como não usadas como referência e um módulo de emissão para emitir zero, uma ou muitas imagens, marcadas como que precisam ser emitidas, das imagens no DPB. O módulo de marcação é ainda para marcar zero, uma ou muitas imagens como que não precisam ser emitidas. O decodificador compreende um módulo de esvaziamento de imagem para esvaziar, a partir do DPB, qualquer imagem marcada como não usada como referência e que não precisa ser emitida, das imagens no DPB. O decodificador ainda compreende um módulo de determinação de número para determinar um número de imagens no DPB que estão marcadas como que precisam ser emitidas e um módulo de comparação para comparar o número com um valor derivado de ao menos um elemento de sintaxe presente no fluxo de bits. Nesta modalidade, a unidade de saída é ainda para emitir uma imagem, que é uma primeira imagem em ordem de saída, das imagens no

DPB que estão marcadas como que precisam ser emitidas se o número é maior do que o valor. O módulo de marcação é ainda para marcar a imagem como que não precisa ser emitida se o número é maior do que o valor. Em uma modalidade preferencial, o módulo de determinação de número determina o número de imagens, o módulo de comparação compara o número, o módulo de emissão emite a imagem e o módulo de marcação marca a imagem após o módulo de decodificação decodificar a imagem atual.

[0044] Ainda outro aspecto das modalidades refere-se a um método executado por um codificador. O método compreende marcar todas as imagens em um DPB que não estão presentes em um RPS para uma imagem atual de uma sequência de vídeo como não usada como referência e marcar zero, uma ou muitas imagens, marcadas como que precisam ser emitidas, das imagens no DPB não precisa ser emitida. O método também compreende esvaziar, a partir do DPB, qualquer imagem marcada como não usada como referência e que não precisava ser emitida das imagens no DPB. O método compreende ainda decodificar a imagem atual. Um número de imagens no DPB que estão marcadas como que precisam ser emitidas é determinado e comparado com um valor derivado de ao menos um elemento de sintaxe definido. O método também compreende marcar uma imagem, que é uma primeira imagem em ordem de saída, das imagens no DPB que estão marcadas como que precisam ser emitidas, como que não precisa ser emitida se o número é maior do que o valor. Em uma modalidade preferencial, determinar o número de imagens, comparar o número e marcar a imagem são executados após decodificar a imagem atual.

[0045] Um aspecto relacionado das modalidades define um codificador configurado para marcar todas as imagens em um DPB que não estão presentes em um RPS para uma imagem atual de uma sequência de vídeo como não usada como referência. O codificador é também configurado para marcar zero, uma ou muitas imagens, marcadas como que precisam ser

emitidas, das imagens no DPB, como que não precisa ser emitidas e esvaziar, a partir do DPB, qualquer imagem marcada como não usada como referência e que não precisa ser emitida das imagens no DPB. O codificador é ainda configurado para decodificar a imagem atual. O codificador é adicionalmente configurado para determinar um número de imagens no DPB que estão marcadas como que precisam ser emitidas e comparar o número com um valor derivado de ao menos um elemento de sintaxe definido. O codificador é também configurado para marcar uma imagem, que é uma primeira imagem em ordem de saída, das imagens no DPB que estão marcadas como que precisam ser emitidas, como que não precisa ser emitida se o número é maior do que o valor. Em uma modalidade preferencial, o codificador é configurado para determinar o número de imagens, comparar o número e marcar a imagem após decodificar a imagem atual.

[0046] Outro aspecto relacionado das modalidades define um codificador compreendendo um módulo de marcação para marcar todas as imagens em um DPB que não estão presentes em um RPS para uma imagem atual de uma sequência de vídeo com não usadas para referência e marcar zero, uma ou muitas imagens, marcadas como que não precisam ser emitidas, das imagens no DPB, como que não precisam ser emitidas. O codificador também compreende um módulo de esvaziamento de imagem para esvaziar, a partir do DPB, qualquer imagem marcada como não usada como referência e que não precisa ser emitida das imagens no DPB. O codificador ainda compreende um módulo de decodificação para decodificar a imagem atual. O codificador também compreende um módulo de determinação de número para determinar um número de imagens no DPB que estão marcadas como que precisam ser emitidas e um módulo de comparação para comparar o número com um valor derivado de ao menos um elemento de sintaxe definido. Nesta modalidade, o módulo de marcação é ainda para marcar uma imagem, que é uma primeira imagem em ordem de saída, das imagens no DPB que estão

marcadas como que precisam ser emitidas, como que não precisa ser emitida se o número é maior do que o valor. Em uma modalidade preferencial, o módulo de determinação de número determina o número de imagens, o módulo de comparação compara o número e o módulo de marcação marca a imagem após o módulo de decodificação decodificar a imagem atual.

[0047] Aspectos adicionais das modalidades referem-se a um terminal móvel compreendendo um decodificador de acordo com o descrito acima e/ou um codificador de acordo com o descrito acima e um nó de rede compreendendo um decodificador de acordo com o descrito acima e/ou um codificador de acordo com o descrito acima.

BREVE DESCRIÇÃO DOS DESENHOS

[0048] As modalidades, junto com objetivos adicionais e suas vantagens, podem ser entendidas melhor fazendo referência à seguinte descrição tomada junto com os desenhos em anexo, nos quais:

A Figura 1 é uma visão geral esquemática de um decodificador H.264/AVC.

[0049] A Figura 2 ilustra a ordem de saída e decodificação para um exemplo de uma sequência de vídeo de multicamadas.

[0050] A Figura 3 é um fluxograma de decodificação simplificado de H.264/AVC.

[0051] A Figura 4 ilustra a ordem de saída e decodificação para um exemplo de uma sequência de vídeo de multicamadas.

[0052] A Figura 5 é um fluxograma de decodificação simplificado de HEVC.

[0053] A Figura 6 ilustra a ordem de saída e decodificação e camadas temporais para um exemplo de uma sequência de vídeo de multicamadas.

[0054] A Figura 7 ilustra a ordem de saída e decodificação para um exemplo de uma sequência de vídeo.

[0055] A Figura 8 é um fluxograma de um método executado por um

decodificados de acordo com uma modalidade.

[0056] A Figura 9 é um fluxograma de etapas opcionais adicionais do método da

[0057] Figura 8.

[0058] A Figura 10 é um fluxograma de uma etapa opcional adicional do método da Figura 8.

[0059] A Figura 11 é um fluxograma de um método executado por um decodificados de acordo com outra modalidade.

[0060] A Figura 12 é um fluxograma de etapas opcionais adicionais do método das Figuras 9 ou 11.

[0061] A Figura 13 é um fluxograma de uma etapa opcional adicional do método das Figuras 8 ou 11.

[0062] A Figura 14 é um fluxograma de decodificação simplificado de uma modalidade.

[0063] A Figura 15 é um fluxograma de decodificação simplificado de outra modalidade.

[0064] A Figura 16 é um diagrama de blocos esquemático de um decodificador de acordo com uma modalidade.

[0065] A Figura 17 é um diagrama de blocos esquemático de um decodificador de acordo com outra modalidade.

[0066] A Figura 18 é um diagrama de blocos esquemático de acordo com uma modalidade adicional.

[0067] A Figura 19 é um diagrama de blocos esquemático de um decodificador de acordo com ainda outra modalidade.

[0068] A Figura 20 é um diagrama de blocos esquemático de um decodificador de acordo com uma modalidade adicional.

[0069] A Figura 21 é um fluxograma de um método executado por um codificador de acordo com uma modalidade.

[0070] A Figura 22 é um fluxograma de um método executado por um

codificador de acordo com outra modalidade.

[0071] A Figura 23 é um diagrama de blocos esquemático de um codificador de acordo com uma modalidade.

[0072] A Figura 24 é um diagrama de blocos esquemático de um codificador de acordo com outra modalidade.

[0073] A Figura 25 é um diagrama de blocos esquemático de um codificador de acordo com uma modalidade adicional.

[0074] A Figura 26 é um diagrama de blocos esquemático de um codificador de acordo com ainda outra modalidade.

[0075] A Figura 27 é um diagrama de blocos esquemático de um codificador de acordo com uma modalidade adicional.

[0076] A Figura 28 é um diagrama de blocos esquemático de um terminal móvel de acordo com uma modalidade.

[0077] A Figura 29 é um diagrama de blocos esquemático de um nó de rede de acordo com uma modalidade.

[0078] A Figura 30 ilustra a ordem de saída e decodificação para um exemplo de uma sequência de vídeo de multicamadas.

DESCRIÇÃO DETALHADA

[0079] Por todos os desenhos, os mesmos números de referência são usados para elementos similares ou correspondentes.

[0080] As presentes modalidades referem-se geralmente à decodificação e codificação de imagens de uma sequência de vídeo, e em particular, à emissão ou transmissão de imagens a partir de um armazenador temporário de imagens decodificadas em conjunto com a decodificação e codificação de imagens. As modalidades possibilitam ainda uma redução no atraso de emissão e permitem uma emissão mais precoce de imagens do que o estado das soluções da técnica.

[0081] Em uma modalidade geral, quando uma imagem é decodificada, o decodificados ou codificador (a decodificação de imagens é

também executada durante a codificação de modo a obter imagens de referência reconstruídas a partir de imagens previamente codificadas) determina o número de imagens em um armazenador temporário de imagens decodificadas (DPB) que estão marcadas como que precisam ser emitidas e compara esse número com um valor que é derivado de elementos de sintaxe no fluxo de bits. Se o número de imagens no DPB marcadas como que precisam ser emitidas é maior do que o valor que é derivado de elementos de sintaxe no fluxo de bits, um processo de emissão modificado é executado ou a imagem das imagens no DPB marcadas como que precisam ser emitidas, que é a primeira imagem em ordem de saída, é emitida, denotado processo de emissão de imagem #2, e marcada como que não precisa ser emitida.

[0082] Várias modalidades de implementação particulares e aspectos não serão descritos aqui começando com o lado de decodificação e então continuando com o lado de codificação.

[0083] Marcada como que precisa ser emitida implica aqui que uma imagem está esperando ser emitida, tal como para exibição ou armazenamento. Correspondentemente, marcada como que não precisa ser emitida implica aqui que uma imagem não está esperando ser emitida e pode, por exemplo, já ter sido emitida. Correspondentemente, marcada como usada como referência implica em que a imagem é usada para predição quando outra imagem é decodificada. Marcada como não usada como referência implica em que a imagem não pode ser usada como referência ou predição por imagens subsequentes. A marcação, como mencionada acima, deveria não ser interpretada como literalmente marcando as imagens com uma etiqueta "precisa ser emitida", "não precisa ser emitida", "usada como referência" ou "não usada como referência". De preferência, a marcação deveria ser interpretada que a imagem é identificada de alguma forma que ela deveria não ser emitida e que ela poderia ser usada como imagem de referência/não usada como imagem de referência. A marcação poderia ser implementada em várias

modalidades. Por exemplo, a imagem poderia ser armazenada em uma parte particular do DPB dedicado por imagens que deveriam ser emitidas ou que deveriam não ser emitidas ou imagens que poderiam ser usadas como imagem de referência ou que deveriam não ser usadas como imagem de referência. Alternativamente, a imagem poderia ser marcada ou de outra forma associada com um sinalizador ou outra palavra de código identificando a imagem como que precisa ser emitida versus que não precisa ser emitida e usada como referência versus não usada como referência. No caso de uma implementação de sinalizador, o sinalizador poderia ter um primeiro valor (O_{bin} OU 1_{bin}) para indicar uma das alternativas e então ter um segundo valor (1_{bin} OU O_{bin}) para indicar a outra alternativa. Por exemplo, um primeiro sinalizador poderia ser usado com relação à emissão de imagens e um segundo sinalizador poderia ser usado com relação à referência.

[0084] Portanto, um aspecto geral refere-se a um método executado por um de-codificador. No método, o decodificador determinar o número de imagens no DPB que estão marcadas como que precisam ser emitidas e compara esse número com um valor que é derivado de elementos de sintaxe no fluxo de bits. Se o número de imagens no DPB marcadas como que precisam ser emitidas é maior do que o valor que é derivado de elementos de sintaxe no fluxo de bits, um processo de emissão modificado é executado ou a imagem das imagens no DPB marcadas como que precisam ser emitidas, que é a primeira imagem em ordem de saída é emitida, denotado processo de emissão de imagem #2 aqui, e marcada como que não precisa ser emitida.

[0085] Esse aspecto geral é preferencialmente implementado em um decodificados compatível com HEVC, também chamado de um decodificador compatível com H.265. Em tal caso, uma modalidade de implementação refere-se a um método executado por um decodificador compatível com HEVC e compreendendo as seguintes etapas. O decodificador compatível com HEVC determina um número de imagens em um DPB que estão

marcadas como que precisam ser emitidas. O decodificador compatível com HEVC também compara o número com um valor derivado de ao menos um elemento de sintaxe presente em um fluxo de bits representando imagens de uma sequência de vídeo. O decodificador compatível com HEVC emite ainda uma imagem, que é uma primeira imagem em ordem de saída, das imagens no DPB que estão marcadas como que precisam ser emitidas se o número é maior do que o valor. Em adição, o decodificador compatível com HEVC marca a imagem emitida como que não precisa ser emitida se o número é maior do que o valor.

[0086] A Figura 8 é um fluxograma de um método executado por um decodifica-dor de acordo com uma modalidade. O método compreende determinar, na etapa S1 e após uma imagem atual ter sido decodificada e armazenada em um DPB, um número de imagens no DPB que estão marcadas como que precisam ser emitidas.

[0087] Uma próxima etapa S2 compreende comparar esse número com um valor `sps_max_num_reorder_pics[HighestTid]`. Se o número é maior do que o valor, o método continua para a etapa S3. Essa etapa S3 compreende emitir uma imagem, que é uma primeira imagem em ordem de saída, das imagens no DPB que estão marcadas como que precisam ser emitidas. A imagem, que foi emitida na etapa S3, é então marcada como que não precisa ser emitida na etapa S4. Portanto, essa etapa S4 é executada se o número é maior do que o valor.

[0088] Nesta modalidade, `sps_max_num_reorder_pics[HighestTid]` é usado como a representação preferencial do valor derivado de ao menos um elemento de sintaxe presente em um fluxo de bits representando imagens de uma sequência de vídeo. Ademais, `HighestTid` especifica uma camada mais alta que é decodificada pelo decodificador da sequência de vídeo. Portanto, `HighestTid` indica a subcamada temporal mais alta, se a sequência de vídeo compreende uma ou mais camadas temporais, e indica a camada decodificada

mais alta, se a sequência de vídeo ao invés, compreende outro tipo de camadas, tal como uma ou mais camadas espaciais, camadas de qualidade, etc.

[0089] Em uma modalidade, `sps_max_num_reorder_pics[HighestTid]` indica o número máximo permitido de imagens que podem preceder qualquer imagem na sequência de vídeo codificada em ordem de decodificação e seguem essa imagem em ordem de saída quando `HighestTid` é igual a `i`. O prefixo `sps` de `max_num_reorder_pics[i]` indica que o elemento de sintaxe está preferencialmente presente em um Conjunto de Parâmetros de Sequência (SPS) do fluxo de bits.

[0090] Assim, nesta modalidade, poderia haver um valor derivado por camada no fluxo de bits e o valor para usar na comparação executada na etapa S2 é o valor que é derivado para a camada mais alta que é realmente decodificada pelo decodificador da sequência de vídeo. Nota-se que essa camada decodificada mais alta pode ser igual ou menor do que o número máximo de camadas na sequência de vídeo como especificado por um elemento de sintaxe `max_sub_layers_minus1`, tal como `vps_max_sub_layers_minus1` ou `sps_max_sub_layers_minus1`, dependendo se o elemento de sintaxe é restaurado a partir de um Conjunto de Parâmetros de Vídeo (VPS) ou SPS.

[0091] Portanto, em uma modalidade particular, a sequência de vídeo é uma sequência de vídeo de multicamadas compreendendo múltiplas, isto é, ao menos duas, camadas de imagens. Cada camada das múltiplas camadas tem preferencialmente um elemento de sintaxe associado definindo um respectivo valor. O valor usado na comparação da etapa S2 é então o valor obtido a partir do elemento de sintaxe associado com, nesta modalidade, a camada mais alta que é decodificada pelo decodificador das múltiplas camadas.

[0092] Em uma modalidade particular, `sps_max_num_reorder_pics[i`

] indica o número máximo permitido de imagens na mesma camada ou camada inferior, tal como a mesma camada ou camada temporal inferior, como comparado com a camada i , que precede uma imagem em ordem de decodificação e segue essa imagem em ordem de saída.

[0093] Em uma modalidade, se o número não é maior do que o valor determinado na comparação da etapa S2, o método preferencialmente termina e nenhuma imagem é emitida ou marcada. Assim, em tal caso, as etapas S3 e S4 são omitidas e não executadas, ver linha tracejada direita.

[0094] Em uma modalidade, as etapas S2-S4 poderiam ser executadas somente uma vez após uma imagem atual ter sido decodificada e armazenada no DPB. Alternativamente, o circuito formado pelas etapas S2-S4, ver linha tracejada esquerda, poderia ser executado até que o número de imagens no DPB que estão marcadas como que precisam ser emitidas não é maior do que o valor $\text{sps_max_num_reorder pics[HighestTid]}$. Cada vez que o circuito das etapas S2-S4 é executado, o número de imagens no DPB que estão marcadas como que precisam ser emitidas é reduzido por um até a marcação, na etapa S4, da imagem emitida na etapa S3, como que não precisa ser emitida. Isso significa que as etapas S3 ou S4 poderiam ser executadas, após a imagem atual ter sido decodificada e armazenada no DPB, zero vezes, se $\text{number} \leq \text{sps_max_num_reorder pics[HighestTid]}$; uma vez, se $\text{number} = \text{sps_max_num_reorder pics[HighestTid]} + 1$; ou mais do que uma vez se $\text{number} > \text{sps_max_num_reorder pics[HighestTid]} + 1$, isto é, n vezes se

[0095] $\text{number} = \text{sps_max_num_reorder pics[HighestTid]} + n$. Uma vez que a comparação na etapa S2 determina que o número não é mais maior do que o valor $\text{sps_max_num_reorder pics[HighestTid]}$, o método termina.

[0096] A determinação do número de imagens no DPB marcadas como que precisam ser emitidas na etapa S1 é preferencialmente executada após a imagem atual ter sido decodificada e armazenada no DPB.

[0097] O método descrito acima e ilustrado na Figura 8 é

preferencialmente executado uma vez para cada imagem da sequência de vídeo que é decodificada e armazenada no DPB. Assim, o método é preferencialmente executado instantaneamente uma vez que as imagens atuais são consideradas como decodificadas, isto é, após a última unidade de decodificação da imagem ser decodificada, e a imagem decodificada atual é armazenada em um armazenador temporário de imagem vazio, isto é, área de armazenamento de imagem, no DPB.

[0098] A Figura 9 é um fluxograma de etapas opcionais adicionais do método da Figura 8. O método começa na etapa S10, que compreende analisar um cabeçalho de fatia da imagem atual a ser decodificada da sequência de vídeo. Uma próxima etapa S11 determina um conjunto de imagens de referência (RPS) para a imagem atual com base no cabeçalho de fatia analisado. Todas as imagens no DPB que não estão presentes no RPS são marcadas como não usadas como referência na etapa S12. Uma etapa seguinte S13 compreende emitir zero, uma ou muitas, isto é, mais de uma imagem, que estão marcadas como que precisam ser emitidas, das imagens no DPB. Essa etapa S13 também compreende marcar zero, uma ou muitas imagens emitidas como que não precisam ser emitidas. A etapa S14 compreende esvaziar ou remover, do DPB, qualquer imagem marcada como não usada como referência e que não precisa ser emitida das imagens no DPB. A imagem atual é então decodificada na etapa S15. O método então continua para a etapa S1 na Figura 8. Portanto, nesta modalidade, determinar o número de imagens na etapa S1, comparar o número na etapa S2, emitir a imagem na etapa S3 e marcar a imagem na etapa S4 são executados após a decodificação da imagem atual na etapa S15.

[0099] Em adição à decodificação da imagem atual na etapa S15, essa etapa também compreende preferencialmente marcar a imagem atual como usada como referência, ou em uma modalidade opcional, marcar a imagem atual como usada como referência a curto prazo. A imagem atual é

preferencialmente também marcada como que precisa ser emitida ou que não precisa ser emitida de acordo com o valor da variável PicOutputFlag preferencialmente obtido na etapa S10.

[00100] Geralmente, uma sequência de vídeo codificada, isto é, fluxo de bits, compreende unidades de Camada de Abstração de Rede (NAL). Basicamente, uma unidade NAL compreende ou uma fatia com um cabeçalho de fatia correspondente incluindo informação de controle para essa fatia e dados de carga útil de vídeo ou a unidade NAL compreende um conjunto de parâmetros, tal como VPS, SPS e Conjunto de Parâmetros de Imagem (PPS). O conjunto de parâmetros compreende informação de controle. Uma imagem da sequência de vídeo pode consistir de uma única fatia ou múltiplas fatias. A etapa S10 da Figura 9 compreende assim analisar a parte de cabeça de fatia da unidade NAL compreendendo uma fatia da imagem. Se a imagem compreende múltiplas fatias e é assim distribuída dentre as múltiplas unidades NAL compreendendo um respectivo cabeçalho de fatia, então a etapa S10 é preferencialmente executada para cada fatia da imagem. Entretanto, as etapas S11 a S14 são preferencialmente somente executadas para uma das fatias na imagem, tipicamente a primeira fatia na imagem.

[00101] O cabeçalho de fatia analisado na etapa S10 compreende informação que possibilita que o decodificador gere um RPS. O RPS é um conjunto de imagens de referência associadas com a imagem atual e consistindo de todas as imagens de referência que estão antes da imagem atual em ordem de decodificação e que podem ser usadas para referência, isto é, predição inter, da imagem atual ou qualquer imagem na sequência de vídeo seguindo a imagem atual em ordem de decodificação.

[00102] A informação obtida na etapa S10 e usada para determinar o RPS na etapa S11 pode, por exemplo, compreender um identificador para uma estrutura de sintaxe RPS incluída em um conjunto de parâmetros, tal como SPS, aplicável à fatia atual. Um exemplo de tal identificador aplicável

para HEVC é `short term_ref pic_set_idx`. O conjunto de parâmetros compreende então elementos de sintaxe definindo um ou mais RPS, tal como `num_short term_ref pic_sets` definindo o número de elementos de sintaxe `short term_ref pic_set()` incluídos no SPS, onde o elemento de sintaxe `short term_ref pic_set()` define um RPS candidato para a imagem atual.

[00103] Alternativamente, a informação obtida na etapa S10 poderia ser usada diretamente para determinar o RPS na etapa S11. Essa informação pode, por exemplo, compreender o elemento de sintaxe `short term_ref pic_set()`, onde o elemento de sintaxe `short term_ref pic_set()` define o RPS da imagem atual.

[00104] Mais informação para determinar RPS pode ser encontrada em WO 20131002700 e seções 7.3.2.2, 7.3.51, 7.3.5.2, 7.422, 7.4.51, 7.4.5.2 e 8.3.2 em HCTVC-JI003_d7, High efficiency video coding (HEVC) text specification draft 8, B. Bross e outros, JCT-VC de ITU-T SG16 WP3 e ISO/IEC JTC11SC29/WG11 10th Meeting, Stockholm, 11 a 20 de julho de 2012.

[00105] O RPS especifica, como mencionado anteriormente, as imagens da sequência de vídeo que são mantidas no DPB, isto é, estão disponíveis para uso como imagens de referência quando decodificando a imagem atual e/ou quando decodificando imagens seguindo a imagem atual em ordem de decodificando. Isso significa que quaisquer imagens de referência armazenadas no DPB, mas não listadas no RPS, não serão usadas mais como imagem de referência. Portanto, a etapa S12 compreende assim marcar tais imagens no DPB, mas não identificadas no RPS como não usadas como referência.

[00106] A emissão de imagens na etapa S13 pode compreender emitir uma imagem marcada como que precisa ser emitida, emitir mais de uma imagem marcada como que precisa ser marcada ou de fato, não emitir qualquer imagem. Quaisquer imagens emitidas na etapa S13 são emitidas de

acordo com uma ordem de saída, preferencialmente emitidas na ordem do menor valor de Contagem de Ordem de Imagem (POC).

[00107] Qualquer imagem emitida na etapa S13 é então marcada como que não precisa ser emitida uma vez que a imagem foi então emitida. Se nenhuma imagem é emitida, certamente nenhuma marcação é executada na etapa S13.

[00108] As imagens que são marcadas como não usadas como referência e não precisam ser emitidas não são mais necessárias, nem como imagens de referência nem para emissão. Consequentemente, a etapa S14 esvazia ou remove quaisquer tais imagens marcadas do DPB para liberar assim um armazenador temporário de imagens, isto é, área de armazenamento de imagens, no DPB.

[00109] A decodificação da imagem atual é executada de acordo com os métodos bem conhecidos de decodificação de imagem, tal como especificado no padrão HEVC/H.265. Portanto, a decodificação na etapa S35 envolve gerar valores de pixels dos pixels ou amostras na imagem atual, tipicamente usando valores de pixel de uma ou mais imagens previamente decodificadas no DPB como imagens de referência.

[00110] Na modalidade mostrada na Figura 9, o método mostrado na Figura 8 e compreendendo as etapas S1-S4 poderia ser considerado como um processo de emissão de imagem adicional, que foi mencionado anteriormente como o processo de emissão de imagem #2. Esta modalidade possibilita então a emissão de imagens na etapa S3 mesmo após a imagem atual ter sido decodificada na etapa S15, mas antes de iniciar o processamento de uma próxima imagem no fluxo de bits, isto é, antes de executara etapa S10 para uma próxima imagem em ordem de decodificação.

[00111] A Figura 14 resume essa modalidade. Portanto, em uma modalidade, uma etapa de processo de emissão de imagem adicional é adicionada após a decodificação de imagem de modo a permitir a emissão

mais precoce de imagens, ver Figura 14. Um método de decodificação pode conter e/ou um decodificador pode ser configurado para executar as seguintes etapas ordenadas de acordo com a modalidade:

[00112] O cabeçalho de fatia, incluindo o RPS, do primeiro cabeçalho de fatia de uma imagem P é analisado.

[00113] Um processo de marcação de imagem é feito, por exemplo, imagens podem ser marcadas como não usadas como referência pelo RPS a partir do cabeçalho de fatia.

[00114] Um processo de emissão de imagem é feito no qual as imagens podem ser emitidas.

[00115] A imagem P é decodificada.

[00116] Após a imagem P ser decodificada, o decodificador determinar o número de imagens no DPB que estão marcadas como que precisam ser emitidas e compara esse número com o valor que é derivado de elementos de sintaxe, exemplificado por `sps_max_num_reorder_pics[HighestTid]`, no fluxo de bits.

[00117] Se o número de imagens no DPB marcadas como que precisam ser emitidas é maior do que o valor que é derivado dos elementos de sintaxe no fluxo de bits, a imagem das imagens no DPB marcadas como que precisam ser emitidas que é a primeira imagem em ordem de saída é emitida e marcada como que não precisa ser emitida. Assim, de acordo com a modalidade, o processo de emissão de imagem #2 é introduzido.

[00118] A próxima imagem Q é decodificada repetindo-se as etapas 1-6 acima para a imagem Q.

[00119] O método como descrito na Figura 8 poderia também ser implementado como parte de um processo de emissão modificado, tal como para HEVC. Nessa abordagem, a etapa de processo de emissão de imagem existente é modificada de modo a permitir a emissão precoce de imagens.

[00120] A modificação fornece uma solução para o problema com

no_output_of_prior_pics flag como discutido na seção Fundamento. A solução é similar às modalidades discutidas acima, mas com a mudança que, ao invés de adicionar um processo de emissão adicional, o processo de emissão HEVC existente é modificado quando no_output_of_prior_pics_flag é igual a 1.

[00121] A Figura 10 é um fluxograma de uma etapa opcional adicional do método da Figura 8 quando implementando a modalidade mencionada acima. A etapa S20 compreende analisar um cabeçalho de fatia de uma imagem de acesso aleatória

[00122] (RAP), também chamada de uma imagem de ponto de acesso aleatório intra (I RAP) na técnica, da sequência de vídeo para obter um valor de um sinalizador no_output_of_prior_pics flag. Se o valor do sinalizador no_output_of_prior_pics_flag é 1, o método continua para a etapa S1 da Figura 8. Portanto, nesta modalidade, determinar o número na etapa S1, comparar o número na etapa S2, emitir a imagem na etapa S3 e marcar a imagem na etapa S4 são executados se o valor de no_output_of_prior_pics flag é um.

[00123] Correspondentemente, se o valor de no_output_of_prior_pics flag é zero (0), o método terminar e o processo de emissão modificado não é executado. [00114] Quando esta modalidade é aplicada ao HEVC, o método compreende preferencialmente as etapas adicionais de determinar um RPS para a imagem de acesso aleatório com base no cabeçalho de fatia analisado na etapa S20 (comparar com a etapa S11 na Figura 9). O método também compreende marcar todas as imagens no DPB que não são listadas no RPS como não usadas como referência (comparar com a etapa S12 na Figura 9). A imagem de acesso aleatório é decodificada (comparar com a etapa S15 na Figura 9). Nesta modalidade, determinar o número na etapa S1, comparar o número na etapa S2, emitir a imagem na etapa S3 e marcar a imagem na etapa S4 são executados após marcar as imagens não listadas no RPS, mas antes de decodificar a imagem de acesso aleatório.

[00124] A Figura 15 é um fluxograma de decodificação simplificado

desta modalidade. Um método de decodificação pode conter e/ou um decodificador pode ser configurado para executar as seguintes etapas ordenadas de acordo com a modalidade:

[00125] O cabeçalho de fatia do primeiro cabeçalho de fatia de uma imagem P é analisado.

[00126] Um processo de marcação de imagem é feito.

[00127] Um processo de emissão de imagem é feito no qual as imagens podem ser emitidas. O processo de emissão de imagem é modificado de modo que se no_output_of_prior_pics flag é igual a 1, o seguinte se aplica:

O decodificador determina o número de imagens no DPB que estão marcadas como que precisam ser emitidas e compara esse número com um valor que é derivado de elementos de sintaxe no fluxo de bits.

[00128] Se o número de imagens no DPB marcadas como que precisam ser emitidas é maior do que o valor que é derivado de elementos de sintaxe no fluxo de bits, a imagem das imagens no DPB marcadas como que precisam ser emitidas que é a primeira em ordem de saída é emitida e marcada como que não precisa ser emitida.

[00129] Todas as imagens restantes no DPB que estão marcadas como que precisam ser emitidas são marcadas como que não precisam ser emitidas. Elas não são emitidas. 4) A imagem P é decodificada.

[00130] A Figura 11 é um fluxograma de um método executado por um decodificador de acordo com outra modalidade. O método compreende analisar um cabeçalho de fatia de uma imagem atual a ser decodificada de uma sequência de vídeo na etapa S30. Um RPS é determinado para a imagem atual com base no cabeçalho de fatia analisado na etapa S31. A seguinte etapa S32 compreende marcar todas as imagens no DPB que não estão presentes no RPS como não usadas como referência. Zero, uma ou muitas imagens marcadas como que precisam ser emitidas das imagens no DPB são emitidas e marcadas como que não precisam ser emitidas na etapa S33. A seguinte etapa

S34 compreende esvaziar ou remover qualquer imagem, do DPB, que é marcada como não usada como referência e não precisa ser emitida das imagens no DPB. A imagem atual é então decodificada na etapa S35.

[00131] Essas etapas S30 a S35 correspondem tipicamente às etapas S10-S15 discutidas anteriormente e ilustradas na Figura 9.

[00132] O método então continua para a etapa S36, que compreende determinar um número de imagens no DPB que estão marcadas como que precisam ser emitidas. Esse número é comparado na etapa S37 com um valor derivado de ao menos um elemento de sintaxe presente em um fluxo de bits representando as imagens da sequência de vídeo. Se o número é maior do que o valor, o método continua para as etapas S38 e S39. A etapa S38 compreende emitir a imagem, que é a primeira imagem em ordem de saída, das imagens no DPB que estão marcadas como que precisam ser emitidas. A etapa S39 compreende marcar a imagem, emitir na etapa S38, como que não precisa ser emitida.

[00133] Nesta modalidade, determinar o número de imagens na etapa S37, comparar o número na etapa S38, emitir a imagem na etapa S38 e marcar a imagem na etapa S39 são executados após decodificar a imagem atual na etapa S35. [00120] As etapas S36 a S39 correspondem às etapas S1 a S4 discutidas anteriormente e mostradas na Figura 8, mas com uma diferença pelo fato de que o valor com o qual o número determinado é comparado na etapa S37 não necessariamente tem que ser `sps_max_num_reorder_pics[HighestTid]`, mas poderia, ao invés, ser um valor derivado de um outro elemento(s) de sintaxe presente no fluxo de bits, que é ainda discutido abaixo.

[00134] Em uma modalidade, se o número não é maior do que o valor determinado na comparação da etapa S37, o método preferencialmente termina e nenhuma imagem é emitida ou marcada. Assim, em tal caso, as etapas S38 e S39 são omitidas e não executadas, ver linha tracejada direita.

[00135] Em uma modalidade, as etapas S37-S39 poderiam ser

executadas somente uma vez após uma imagem atual ter sido decodificada e preferencialmente armazenada no DPB. Alternativamente, o circuito formado pelas etapas S37-S39, ver linha tracejada esquerda, poderia ser executado até que o número de imagens no DPB que estão marcadas como que precisam ser emitidas não é mais maior do que o valor. Cada vez que o circuito das etapas S37-S39 é executado, o número de imagens no DPB que estão marcadas como que precisam ser emitidas, é reduzido em um através da marcação, na etapa S39, da imagem, emitida na etapa S38, como que não precisa ser emitida.

[00136] A Figura 12 é um fluxograma de etapas opcionais adicionais do método da Figura 11, mas também aplicável às modalidades como mostrado nas Figuras 8 e 9. O método continua a partir da etapa S35 na Figura 11 ou etapa S15 na Figura 9, no qual a imagem atual foi decodificada. Uma próxima etapa S40 compreende armazenar a imagem atual decodificada no DPB em um armazenador temporário de imagem vazio, isto é, área de armazenamento de imagem. A imagem atual decodificada é marcada na etapa S41 como que precisa ser emitida ou como que não precisa ser emitida. A imagem é opcionalmente também marcada como usada para referência a curto prazo, isto é, usada como imagem de referência. Essa etapa S41 é preferencialmente executada com base em um PicOutputFlag atribuído à imagem atual. Assim, se o sinalizador tem um valor de 1, então a imagem atual decodificada é marcada como que precisa ser emitida e de outra forma, isto é, o sinalizador tem um valor de 0, a imagem atual decodificada é marcada como que não precisa ser emitida. O PicOutputFlag pode ser derivado do elemento de sintaxe no fluxo de bits aplicável à imagem atual presente, tal como com base no elemento de sintaxe pic_output_flag que pode estar presente no cabeçalho de fatia da imagem atual.

[00137] O método então continua para a etapa S36 da Figura 11 ou etapa S1 da Figura 8, onde o número de imagens no DPB marcadas como que

precisam ser emitidas é determinado.

[00138] Aqui, várias modalidades de selecionar o valor usado na comparação na etapa S37 da Figura 11 e na etapa S2 da Figura 8 serão descritas.

[00139] Em uma modalidade, a sequência de vídeo é uma sequência de vídeo de multicamadas compreendendo múltiplas camadas de imagens. Cada camada das múltiplas camadas tem um elemento de sintaxe associado definindo um respectivo valor usado no processo de emissão. O método compreende ainda uma etapa opcional adicional como mostrada no fluxograma da Figura 13. O método continua da etapa S35 na Figura 11 ou da etapa S1 na Figura 8. Uma próxima etapa S50 compreende selecionar um valor derivado de um elemento de sintaxe associado com uma camada mais alta que é decodificada pelo decodificador das múltiplas camadas. O método então continua para a etapa S37 na Figura 11 ou a etapa S2 na Figura 8, onde esse valor selecionado é usado.

[00140] A etapa S37 e a etapa S2 compreendem preferencialmente, nesta modalidade, comparar o número determinado na etapa S36 ou S1 com um valor de `sps_max_num_reorder pics[HighestTid]`. Se o número é maior do que o valor `sps_max_num_reorder pics[HighestTid]`, o método continua para a etapa S38 ou S3, que compreende, nesta modalidade, emitir uma imagem no DPB com um valor menor de `PicOrderCntVal` de todas as imagens no DPB que estão marcadas como que precisam ser emitidas. `PicOrderCntVal` representa um valor de contagem de ordem de imagem da imagem, que, por sua vez, define preferencialmente a ordem de saída de imagens armazenadas no DPB.

[00141] Em um aspecto de implementação particular desta modalidade, a sequência de vídeo é uma sequência de vídeo de multicamadas compreendendo múltiplas camadas de imagens. Em tal caso, cada camada tem um respectivo valor, isto é, `sps_max_num_reorder pics[i]` para o número de

camada i . O valor a usar é então o valor associado com a camada decodificada mais alta, isto é, a camada mais alta que é decodificada pelo decodificador, tal como a subcamada mais alta se as camadas são camadas temporais diferentes.

[00142] Em outra modalidade, a sequência de vídeo é uma sequência de vídeo de multicamadas compreendendo múltiplas camadas de imagens. Cada camada das múltiplas imagens tem um elemento de sintaxe associado definindo um respectivo valor. O método compreende ainda uma etapa opcional adicional como mostrada no fluxograma da Figura 13. O método continua a partir da etapa S36 na Figura 11. Uma próxima etapa S50 compreende selecionar um valor derivado de um elemento de sintaxe associado com uma camada mais alta das múltiplas camadas. O método então continua para a etapa S37 onde esse valor selecionado é usado.

[00143] A etapa S37 compreende preferencialmente, nesta modalidade, comparar o número determinado na etapa S36 com um valor de `sps_max_num_reorder_pics[sps_max_sub_layers_minus1]`. Se o número é maior do que o valor `sps_max_num_reorder_pics[sps_max_sub_layers_minus1]`, o método continua para a etapa S38, que compreende, nesta modalidade, emitir uma imagem no DPB com um menor valor de `PicOrderCntVal` de todas as imagens no DPB que estão marcadas como que precisam ser emitidas. O elemento de sintaxe `sps_max_sub_layers_minus1` especifica um número máximo de camadas da sequência de vídeo.

[00144] Em um aspecto de implementação particular desta modalidade, a sequência de vídeo é uma sequência de vídeo de multicamadas compreendendo múltiplas camadas de imagens. Em tal caso, cada camada tem um respectivo valor, isto é, `sps_max_num_reorder_pics[i]` para o número de camada i . O valor a usar é então o valor associado com a camada mais alta no fluxo de bits.

[00145] Em uma camada adicional, a sequência de vídeo é uma

sequência de vídeo de multicamadas compreendendo múltiplas camadas de imagens. Cada camada das múltiplas camadas então tem um elemento de sintaxe associado definindo um respectivo valor. O método compreende ainda uma etapa opcional adicional como mostrado no fluxograma da Figura 13. O método continua a partir da etapa S36 na Figura 11. Uma próxima etapa S50 compreende selecionar um valor derivado de um elemento de sintaxe associado com uma camada das múltiplas camadas às quais a imagem atual decodificada pertence. O método então continua para a etapa S37 onde esse valor selecionado é usado.

[00146] Em ainda outra modalidade, a sequência de vídeo é uma sequência de vídeo de multicamadas compreendendo múltiplas camada de imagens. Cada camada das múltiplas camadas tem então um elemento de sintaxe associado definindo um respectivo valor. O método compreende ainda uma etapa opcional adicional como mostrado no fluxograma da Figura 13. O método continua a partir da etapa S36 na Figura 11. Uma próxima etapa S50 compreende selecionar um valor máximo dos respectivos valores.

[00147] O método como mostrado na Figura 11 pode ser aplicado a uma imagem atual que é uma imagem de acesso aleatório da sequência de vídeo. Em tal caso, a etapa S30 compreende preferencialmente analisar o cabeçalho de fatia da imagem de acesso aleatório para obter um valor de um sinalizador no-output-prior flag. Em tal caso, determinar o número na etapa S36, comparar o número na etapa S37, emitir a imagem na etapa S38 e marcar a imagem na etapa S39 poderiam ser condicionais com base no valor desse sinalizador. Portanto, em uma modalidade opcional, essas etapas S36-S39 são executadas se o valor do sinalizador no_output_prior flag é 1.

[00148] Em uma modalidade, marcar a imagem como executada na etapa S12 da Figura 9 e S32 da Figura 11 compreende preferencialmente marcar todas as imagens no DPB que não estão presentes no RPS como não usadas como referência. Essas etapas opcionalmente também compreendem

marcar todas as imagens no DPB que estão presentes no RPS como usadas como referência. Entretanto, geralmente uma imagem é marcada como usada como referência uma vez que é decodificada e armazenada no DPB. Portanto, a etapa S12 e S32 tipicamente envolvem remarcar imagens que não são mais necessárias como referência.

[00149] Em uma modalidade, emitir a imagem como executada na etapa S3 da Figura 8 e na etapa S38 da Figura 11 compreende preferencialmente emitir uma imagem com o menor valor de contagem de ordem de imagem das imagens no DPB que estão marcadas como que precisam ser emitidas se o número é maior do que o valor determinado na etapa S2 ou S37. O valor de contagem de ordem de imagem é preferencialmente representado pelo parâmetro PicOrderCntVal.

[00150] Em uma modalidade, emitir a imagem como executada na etapa S3 da Figura 8 e na etapa S38 da Figura 11 é preferencialmente executado antes de analisar um cabeçalho de fatia de uma próxima imagem a ser decodificada da sequência de vídeo codificada.

[00151] Em uma modalidade, o método executado na Figura 8 ou na Figura 11 compreende a etapa adicional de restaurar ao menos um elemento de sintaxe a partir de um conjunto de parâmetros associado com o fluxo de bits e selecionado a partir de um grupo que consiste de um PPS, um SPS, e um VPS. Um PPS é identificado com base em um identificador de PPS presente no cabeçalho de fatia da imagem atual. Um SPS que aplica à imagem atual é identificado por um identificador de SPS presente no PPS identificado pelo identificador de PPS presente no cabeçalho de fatia da imagem atual. Correspondentemente, um VPS que aplica à imagem atual é identificado por um identificador de VPS presente no SPS que aplica à imagem atual. Em uma modalidade particular, ao menos um elemento de sintaxe é restaurado a partir de um SPS.

[00152] Uma modalidade particular, que é aplicável a uma

implementação usando no_output_prior pics flag, compreende uma etapa adicional de preferencialmente marcar todas as imagens restantes no DPB marcadas como que precisam ser emitidas como que não precisam ser emitidas se o sinalizador no_output_prior_pics flag é um.

[00153] Em uma modalidade relacionada, o método compreende uma etapa adicional de esvaziar todas as imagens no DPB sem emitir quaisquer imagens se o sinalizador no_output_pics flag é um. A plenitude do DPB é então igual a zero para indicar que o DPB está vazio.

[00154] Aqui abaixo serão discutidas várias modalidades exemplificadas.

Modalidade Exemplificada 1

[00155] Em uma modalidade, uma etapa de processo de emissão de imagem adiciona é adicionada após a decodificação de imagem de modo a permitir a emissão mais precoce de imagens, ver Figura 14.

[00156] Um método de decodificação pode conter e/ou um decodificador pode ser configurado para executar as seguintes etapas ordenadas de acordo com a modalidade:

O cabeçalho de fatia do primeiro cabeçalho de fatia da imagem P é analisado.

[00157] Um processo de marcação de imagem é feito.

[00158] Um processo de emissão de imagem é feito no qual as imagens podem ser emitidas.

[00159] A imagem P é decodificada.

[00160] Após a imagem P ser decodificada, o decodificador determina o número de imagens no DPB que são marcadas como que "precisam ser emitidas" e compara esse número com um valor que é derivado de elementos de sintaxe (exemplificados pela modalidade 9) no fluxo de bits.

[00161] Se o número de imagens no DPB marcadas como que "precisam ser emitidas" é maior do que o valor que é derivado dos elementos

de sintaxe no fluxo de bits, a imagem das imagens no DPB marcadas como que "precisam ser emitidas" que é a primeira imagem em ordem de saída é emitida e marcada como que "não precisam ser emitidas". Assim, de acordo com a modalidade, o processo de emissão de imagem #2 é introduzido.

[00162] 7. Uma próxima imagem Q é decodificada

Modalidade Exemplificada 2

[00163] A mesma da modalidade exemplificada 1 onde o valor é representado por uma única palavra de código do elemento de sintaxe no fluxo de bits.

Modalidade Exemplificada 3

[00164] A mesma da modalidade exemplificada 2 onde o valor é representado por uma palavra de código em um conjunto de parâmetros de sequência, por exemplo, um conjunto de parâmetros de imagem (PPS), um conjunto de parâmetros de sequência (SPS) ou um conjunto de parâmetros de vídeo (VPS).

Modalidade Exemplificada 4

[00165] A mesma das modalidades 1-3 onde há um valor dependente de camada derivado por camada no fluxo de bits e o valor a usar na comparação é o valor que é derivado para a mesma camada da imagem P.

Modalidade Exemplificada 5

[00166] A mesma das modalidades 1-3 onde há um valor derivado por camada no fluxo de bits e o valor a usar na comparação é o valor que é derivado para a camada mais alta no fluxo de bits.

Modalidade Exemplificada 6

[00167] A mesma das modalidades 1-3 onde há um valor derivado por camada no fluxo de bits e o valor a usar na comparação é o valor máximo para todas as camadas.

Modalidade Exemplificada 7

[00168] A mesma da modalidade 5 onde a camada mais alta é definida

como a subcamada mais alta.

Modalidade Exemplificada 8

[00169] A mesma da modalidade 1-7 onde a primeira imagem em ordem de saída é definida como a imagem no DPB com o menor valor de PicOrderCntVal. PicOrderCntVal define o POC e é descrito acima.

Modalidade Exemplificada 9

[00170] A mesma da modalidade 4-8 onde as camadas são camadas temporais

Modalidade exemplificada 10

[00171] A combinação das modalidades 1, 2, 3, 5, 8 e 9 é uma modalidade preferencial. Expressar essa modalidade na especificação HEVC atual (JCTVCK0030 v3) poderia resultar nas seguintes mudanças de especificação HEVC na seção C.5.3 onde o texto em negrito significa texto adicionado:

C.5.3. Decodificação, marcação e armazenamento de imagem e possível emissão de uma imagem

[00172] O seguinte acontece instantaneamente quando a última unidade de decodificação da unidade de acesso n contendo a imagem atual é removida do CPB. [00165] A imagem atual é considerada como decodificada após a última unidade de decodificação da imagem ser decodificada. A imagem decodificada atual é armazenada em um armazenador temporário de imagem vazio no DPB, e o seguinte se aplica.

[00173] Se a imagem decodificada atual tem PicOutputFlag igual a 1, ela é marcada como que "não precisa ser emitida".

[00174] Ao contrário (a imagem decodificada atual tem PicOutputFlag igual a 0), ela é marcada como que "não precisa ser emitida".

[00175] Se a imagem decodificada atual é uma imagem de referência, ela é marcada como "usada como referência", ao contrário (a imagem decodificada atual é uma imagem de não referência), ela é marcada como

"não usada como referência".

[00176] Quando o número de imagens no DPB que são marcadas como que "precisam ser emitidas" é maior do que $\text{sps_max_num_reorder_pics}[\text{sps_max_sub_layers_minus1}]$ após a imagem decodificada atual ter sido armazenada no DPB, a imagem no DPB com o menor valor de PicOrderCntVal de todas as imagens no DPB é cortada, emitida e marcada como que "não precisa ser emitida".

Modalidade Exemplificada 11

[00177] A combinação das modalidades 1, 2, 3, 7, 8 e 9 é outra modalidade preferencial. Expressar essa modalidade na especificação HEVC atual (JCTVCK0030 v3) poderia resultar nas seguintes mudanças de especificação HEVC na seção C.5.3 onde o texto em negrito significa texto adicionado:

C.5.3. Decodificação, marcação e armazenamento de imagem e possível emissão de uma imagem

[00178] O seguinte acontece instantaneamente quando a última unidade de decodificação da unidade de acesso n contendo a imagem atual é removida do CPB

[00179] A imagem atual é considerada como decodificada após a última unidade de decodificação da imagem ser decodificada. A imagem decodificada atual é armazenada em um armazenador temporário de imagem vazio no DPB, e o seguinte se aplica.

[00180] Se a imagem decodificada atual tem PicOutputFlag igual a 1, ela é marcada como que "não precisa ser emitida".

[00181] Ao contrário (a imagem decodificada atual tem PicOutputFlag igual a 0), ela é marcada como que "não precisa ser emitida".

[00182] Se a imagem decodificada atual é uma imagem de referência, ela é marcada como "usada como referência", ao contrário (a imagem decodificada atual é uma imagem de não referência), ela é marcada como

"não usada como referência".

[00183] Quando o número de imagens no DPB que são marcadas como que "precisam ser emitidas" é maior do que `sps_max_num_reorder_pics[HighestTid]` após a imagem decodificada atual ter sido armazenada no DPB, a imagem no DPB com o menor valor de `PicOrderCntVal` de todas as imagens no DPB é cortada, emitida e marcada como que "não precisa ser emitida".

Modalidade Exemplificada 12

[00184] Em uma modalidade, a etapa de processo de emissão de imagem existente é modificada de modo a permitir a emissão mais precoce de imagens, ver Figura 15.

[00185] A modificação fornece uma solução para o problema com `no_output_of_prior_pics` flag como descrito acima. A solução é similar às modalidades exemplificadas 1-11, mas com a mudança de que ao invés de adicionar um processo de emissão adicional, o processo de emissão HEVC existente é modificado quando `no_output_of_prior_pics_flag` é igual a 1, como mostrado abaixo.

[00186] Um método de decodificação pode conter e/ou um decodificador pode ser configurado para executar as seguintes etapas ordenadas de acordo com a modalidade:

1. O cabeçalho de fatia do primeiro cabeçalho de fatia da imagem P é analisado.

[00187] 2. Um processo de marcação de imagem é feito.

[00188] 3. Um processo de emissão de imagem é feito no qual as imagens podem ser emitidas. O processo de emissão de imagem é modificado tal que se `no_output_of_prior_pics_flag` é igual a 1, o seguinte se aplica:

[00189] O decodificador determina o número de imagens no DPB que são marcadas como que "não precisam ser emitidas" e compara esse número com um valor que é derivado de elementos de sintaxe no fluxo de bits.

[00190] Se o número de imagens no DPB marcadas como que

"precisam ser emitidas" é maior do que o valor que é derivado dos elementos de sintaxe no fluxo de bits, a imagem das imagens no DPB marcadas como que "precisam ser emitidas" que é a primeira na ordem de saída é emitida e marcada como que "não precisa ser emitida".

[00191] Todas as imagens restantes no DPB que são marcadas como que "precisam ser emitidas" são marcadas como que "não precisam ser emitidas". Elas não são emitidas.

4. A imagem P é decodificada.

[00192] As modalidades exemplificadas 2-12 se aplicam também a essa modalidade exemplificada 12.

Modalidade Exemplificada 13

[00193] A combinação da modalidade exemplificada 12 e das modalidades exemplificadas 2, 3, 7, 8 e 9 é uma modalidade preferencial. Expressar essa modalidade na especificação HEVC atual (JCTVC-K0030 v3) poderia resultar nas seguintes mudanças de especificação HEVC na seção C.3.1 onde o texto em negrito significa texto adicionado:

C.3.1. Remoção de imagens do DPB

[00194] A remoção de imagens do DPB antes da decodificação da imagem atual (mas após analisar o cabeçalho de fatia da primeira fatia da imagem atual) acontece instantaneamente no tempo de remoção de CPB da primeira unidade de decodificação da unidade de acesso *ri* ~~~~~ a imagem atual) e prossegue como segue.

[00195] O processo de decodificação para o conjunto de imagens de referência como especificado na subcláusula 8.3.2 é invocado.

[00196] Se a imagem atual é uma imagem IDR ou uma imagem BLA, o seguinte se aplica:

[00197] Quando uma imagem IDR ou BLA não é a primeira imagem decodificada e o valor de `pic_width_in_luma_samples` ou `pic_height_in_luma_samples` ou `sps_max_dec_pic_buffering[i]` para

qualquer valor possível de i derivado do conjunto de parâmetros de sequência ativa é diferente do valor de `pic_width_in_luma_samples` ou `pic_height_in_luma_samples` ou `sps_max_dec_pic_buffering[i]` derivado do conjunto de parâmetros de sequência que estava ativo para a imagem anterior, respectivamente, `no_output_of_prior_pics_flag` é inferido como sendo igual a 1 pelo HRD, independentemente do valor real de `no_output_of_prior_pics_flag`.

[00198] Nota — Implementações do decodificador deveriam tentar manipular mudanças no tamanho da imagem ou do DPB mais elegantemente do que o HRD em relação às mudanças em `pic_width_in_luma_samples`, `pic_height_in_luma_samples`, ou `sps_max_dec_pic_buffering[i]`.

[00199] Quando `no_output_of_prior_pics_flag` é igual a 1 ou é inferido como sendo igual a 1, as seguintes etapas ordenadas se aplicam:

Quando o número de imagens no DPB que são marcadas como "que precisam ser emitidas" é maior do que `sps_max_num_reorder_pics[HighestTid]`, a imagem no DPB com o menor valor de `PicOrderCntVal` de todas as imagens no DPB é cortada, emitida e marcada como que "não precisa ser emitida".

[00200] Todos os armazenadores temporários de imagens no DPB são esvaziados sem emissão das imagens que eles contém, e a plenitude do DPB é configurada para 0.

[00201] Todas as imagens k no DPB, para as quais todas as seguintes condições são verdadeiras, são removidas do DPB:

- a imagem k é marcada como "não usada como referência",
- a imagem k tem `PicOutputFlag` igual a 0 ou seu tempo de saída do DPB é menor ou igual ao tempo de remoção de CPB da primeira unidade de decodificação (denotado como unidade de decodificação m) da imagem atual n ; isto é, $to_{dpb}(k) \leq tr(m)$.

[00202] Quando uma imagem é removida do DPB, a plenitude do DPB é diminuída em um.

[00203] Uma vantagem das modalidades descritas aqui sobre a técnica anterior, como representado pela seção C.5.1 nas páginas 216-217 em JCTVC-K0030 v3 será descrita abaixo com relação à Figura 30. A solução da técnica anterior em JCTVC-K0030 v3 usa o parâmetro `sps_max_num_reorder_pics[TemporalID]` para determinar se emite qualquer imagem. Em particular, JCTVC-K0030 v3 determina que quando o número de imagens no DPB que estão marcadas como que precisam ser emitidas é maior do que `sps_max_num_reorder_pics[TemporalID]`, então o processo de transmissão especificado na subcláusula C.5.2.1 é invocado repetidamente até que haja um armazenador temporário de imagem vazio para armazenar a imagem decodificada atual.

[00204] Os números apresentados na Figura 30 indicam a ordem de decodificação das imagens na sequência de vídeo. As imagens deveriam ser emitidas da esquerda para a direita, isto é, 10, b5, 133, b6, 132, b7, 134, 138, e P1. A figura também indica o número da camada, isto é, `TemporalID`, para as diferentes camadas e o parâmetro `sps_max_num_reorder_pics` para cada camada. Se um transmissor ou emissor emitiria imagens em conjunto com a análise de cabeçalhos de fatia usando `sps_max_num_reorder_pics[TemporalID]` como sugerido em JCTVC-K0030 v3, então o processo de emissão será de acordo com o que segue:

Imagem	Não emitir ainda	<code>sps_max_num_reorder_pics</code>	Ação
10	-	0	Sem ação
131	10	0	Emitir 10
132	131	1	Sem ação
133	131 132	2	Sem ação
134	131 132 133	2	Emitir 133 7

[00205] Entretanto, emitir a imagem 133 quando processando a imagem 134 é incorreto uma vez que a imagem b5 deveria ser emitida antes da imagem 133. [00194] Se usar uma modalidade como descrita aqui sem

transmitir, isto é, emitir, após a decodificação da imagem atual e usando HighestId = 3, então o seguinte resultado é alcançado.

[00206]	Imagem	Não emitir ainda	sps max num reorder	pics	Ação
	10 10	4			Sem ação
	131 10 132	4			Sem ação
	132 10 131 132	4			Sem ação
	133 10 131 132 133	4			Sem ação
	134 10 131 132 133 134	4			Emitir 10
	135 131 132 133 134 b5	4			Emitir b5

[00207] Se um usa escalonamento temporal e somente decodifica as camadas 0- 2, então HighestId é 2 e o seguinte resultado é alcançado de acordo com uma modalidade:

	Imagem	Não emitir ainda	sps max num reorder	pics	Ação
	10 10	2			Sem ação
	131 10 131	2			Sem ação
	132 10 131 132	2			Emitir 10
	133 131 132 133 2				Emitir 133
	134 131 132 134 2				Emitir 132

[00208] Portanto, as modalidades descritas aqui são capazes de emitir imagens na ordem correta no exemplo como ilustrado na Figura 30, sendo que a solução da técnica anterior em JCTVC-K0030 v3 não fornece a ordem de saída correta quando invocando o processo de transmissão após a análise dos cabeçalhos de fatia.

[00209] As etapas, funções, procedimentos, módulos e/ou blocos descritos acima em conjunto com as Figuras 8-15 podem ser implementadas em hardware usando qualquer tecnologia convencional, tal como tecnologia de circuito discreto ou de circuito integrado, incluindo circuito eletrônico de propósito geral e circuito de aplicação específica.

[00210] Exemplos particulares incluem um ou mais processadores de

sinal digital adequadamente configurados e outros circuitos eletrônicos conhecidos, por exemplo, portas lógicas discretas interconectadas para desempenhar uma função especializada, ou Circuitos Integrados de Aplicação Específica (ASICs).

[00211] Alternativamente, ao menos algumas das etapas, funções, procedimentos, módulos e/ou blocos descritos acima em conjunto com as Figuras 8-15 podem ser implementados em software tal como um programa de computador para execução por circuito de processamento adequado incluindo um ou mais processadores.

[00212] O processador é capaz de executar instruções de software contidas em um programa de computador armazenado em um produto de programa de computador, por exemplo, na forma de memórias. O respectivo produto de programa de computador pode ser uma memória sendo qualquer combinação de Memória de Acesso Aleatório (RAM) e Memória Somente de Leitura (ROM). A respectiva memória compreende armazenamento persistente, que, por exemplo, pode ser qualquer única memória ou combinação de memória magnética, memória óptica, memória de estado sólido ou até memória remotamente montada.

[00213] O fluxograma ou fluxogramas apresentados acima e mostrados nas Figuras 8-15 podem então ser considerados como um fluxograma ou fluxogramas de computador, quando executados por um ou mais processadores. Um aparelho correspondente pode ser definido como um grupo de módulos de função, onde cada etapa executada pelo processador corresponde a um módulo de função. Nesse caso, os módulos de função são implementados como um programa de computador rodando no processador.

[00214] Exemplos de circuito de processamento incluem, mas não estão limitados a um ou mais microprocessadores, um ou mais Processadores de Sinal Digital (DSPs), uma ou mais Unidades de Processamento Central (CPUs), hardware de aceleração de vídeo, e/ou qualquer circuito lógico

programável adequado tal como um ou mais Arranjos de Portas Programáveis em Campo (FPGAs), ou um ou mais Controladores Lógicos Programáveis (PLCs).

[00215] Dever-se-ia também entender que pode ser possível reutilizar as capacidades de processamento gerais de qualquer dispositivo ou unidade convencional na qual a tecnologia proposta é implementada. Pode também ser possível reutilizar software existente, por exemplo, reprogramando o software existente ou adicionando novos componentes de software.

[00216] De acordo com um aspecto, um decodificador configurado para executar um método de acordo com qualquer uma das modalidades descritas acima é fornecido. O decodificador é configurado para determinar, após uma imagem atual de um fluxo de bits representando imagens de uma sequência de vídeo ter sido decodificada e armazenada em um DPB, um número de imagens no DPB que estão marcadas como que precisam ser emitidas. O decodificador é também configurado para comparar o número com um valor `sps_max_num_reorder pics[HighestTid]`. O decodificador é ainda configurado para emitir uma imagem, que é uma primeira imagem em ordem de saída, das imagens no DPB que estão marcadas como que precisam ser emitidas se o número é maior do que o valor. O decodificador é adicionalmente configurado para marcar a imagem emitida como que não precisa ser emitida se o número é maior do que o valor.

[00217] O decodificador 100 compreende, em uma modalidade, um processador 110 configurado para executar as etapas de método previamente descritas aqui, ver Figura 8 e opcionalmente as Figuras 9, 10 e 12-14. O decodificador 100 pode também compreender uma memória 120 conectada ao processador 110, ver Figura 16. [00206] A Figura 16 é um diagrama de blocos esquemático de um decodificador 100 de acordo com uma modalidade. O decodificador 100 é configurado para receber um fluxo de bits 10 representando imagens de uma sequência de vídeo e decodificar o fluxo de

bits 10. O decodificador 100 compreende um processador 110 e uma memória 120 compreendendo um DPB 125. O processador 110 é configurado para determinar, após a imagem atual ter sido decodificada e armazenada no DPB 125, o número de imagens no DPB 125 que estão marcadas como que precisam ser emitidas. O processador 110 é também configurado para comparar o número com o valor `sps_max_num_reorder_pics[HighestTid] 14`. O processador 110 é ainda configurado para emitir a imagem, que é uma primeira imagem em ordem de saída, das imagens no DPB 125 que estão marcadas como que precisam ser emitidas se o número é maior do que o valor. O processador 110 é adicionalmente configurado para marcar a imagem como que não precisa ser emitida se o número é maior do que o valor.

[00218] Na Figura 16, o decodificador 100 foi ilustrado como compreendendo um processador 110. Esse processador 110 poderia ser implementado como um único processador ou múltiplos processadores, tal como na forma de um circuito de processamento.

[00219] A Figura 16 ilustra assim uma implementação de computador do decodificados 100. Neste exemplo particular, ao menos algumas das etapas, funções, procedimentos, módulos e/ou blocos descritos acima são implementados em um programa de computador, que é carregado na memória 120 para execução pelo processador 110. O processador 110 e a memória 120 são interconectados entre si para possibilitar a execução de software normal. Um dispositivo de entrada/saída opcional (não mostrado) pode também ser interconectado ao processador 110 e/ou à memória 120 para possibilitar a entrada de um fluxo de bits 10 de imagens codificadas e a saída de imagens decodificadas.

[00220] O termo `computador' deveria ser interpretado em um sentido geral como qualquer sistema, dispositivo ou aparelho capaz de executar código de programa ou instruções de programa de computador para executar uma tarefa particular de processamento, determinação ou computação.

[00221] Em uma modalidade, o processador 110 é preferencialmente configurado para analisar um cabeçalho de fatia 12 da imagem atual a ser decodificada da sequência de vídeo. O processador 110 é também configurado para determinar um RPS para a imagem atual com base no cabeçalho de fatia analisado 12. O processador 110 é ainda configurado para marcar todas as imagens no DPB 125 que não estão presentes no RPS como não usada como referência. O processador 110 é adicionalmente configurado para emitir zero, uma ou muitas imagens, marcadas como que precisam ser emitidas, das imagens no DPB 125 e marcar zero, uma ou muitas imagens como que não precisam ser emitidas. O processador 110 é preferencialmente também configurado para esvaziar ou remover, do DPB 125, qualquer imagem marcada como não usada como referência e que não precisa ser emitida das imagens no DPB 125. O processador 110 é ainda configurado para decodificar a imagem atual, preferencialmente usando dados de carga útil de vídeo 16 fornecido na forma codificada. Nesta modalidade, o processador 110 é configurado para determinar o número de imagens, comparar o número, emitir a imagem e marcar a imagem após decodificar a imagem atual.

[00222] Em uma modalidade, o processador 110 é opcionalmente configurado para analisar um cabeçalho de fatia 12 de uma imagem de acesso aleatório da sequência de vídeo para obter um valor de sinalizador no `output_prior_pics` flag. Em tal caso, o processador 110 é opcionalmente configurado para determinar o número, comparar o número, emitir a imagem e marcar a imagem se (e somente se) o valor do sinalizador no `output_prior_pics_flag` é um.

[00223] Outro aspecto das modalidades refere-se a um decodificador configurado para analisar um cabeçalho de fatia de uma imagem atual a ser decodificada de uma sequência de vídeo. O decodificador é também configurado para determinar um RPS para a imagem atual com base no cabeçalho de fatia analisado. O decodificador é ainda configurado para

marcar todas as imagens no DPB que não estão presentes no RPS como não usadas como referência. O decodificador é adicionalmente configurado para emitir zero, uma ou muitas imagens, marcadas como que precisam ser emitidas, das imagens no DPB e marcar zero, uma ou muitas imagens como que não precisam ser emitidas. O decodificador é preferencialmente também configurado para esvaziar ou remover, do DPB, qualquer imagem marcada como não usada para referência e que não precisa ser emitida das imagens no DPB. O decodificador é ainda configurado para decodificar a imagem atual. Nesta modalidade, o decodificador é configurado para determinar um número de imagens no DPB que estão marcadas como que precisam ser emitidas. O decodificador é também configurado para comparar o número com um valor derivado de ao menos um elemento de sintaxe presente no fluxo de bits. O decodificador é ainda configurado para emitir uma imagem, que é uma primeira imagem em ordem de saída, das imagens no DPB que estão marcadas como que precisam ser emitidas, se o número é maior do que o valor. O decodificador é adicionalmente configurado para marcar a imagem como que não precisa ser emitida se o número é maior do que o valor. Isso significa que nesta modalidade, o decodificador é configurado para determinar o número de imagens, comparar o número, emitir a imagem e marcar a imagem após decodificar a imagem atual.

[00224] O codificador 100 compreende, em uma modalidade, um processador 110 configurado para executar as etapas do método previamente descrito aqui, ver Figura 11 e opcionalmente as Figuras 12, 13 e 15. O codificador 100 pode também compreender uma memória 120 conectada ao processador 110, ver Figura 16.

[00225] Nesta modalidade, o decodificador 100 compreende então um processador 110 e uma memória 120 compreendendo um DPB 125. O processador 110 é configurado para analisar o cabeçalho de fatia 12 da imagem atual a ser decodificada da sequência de vídeo. O processador 110 é

também configurado para determinar o RPS para a imagem atual com base no cabeçalho de fatia analisado 12. O processador 110 é ainda configurado para marcar todas as imagens no DPB 125 que não estão presentes no RPS como não usadas como referência. O processador 110 é adicionalmente configurado para emitir zero, uma ou mais imagens, marcadas como que precisam ser emitidas, das imagens no DPB 125 e marcar zero, uma ou muitas imagens como que não precisam ser emitidas. O processador 110 é preferencialmente também configurado para esvaziar ou remover, do DPB 125, qualquer imagem marcada como não usada como referência e que não precisa ser emitida das imagens no DPB 125. O processador 110 é ainda configurado para decodificar a imagem atual. Nesta modalidade, o processador 110 é configurado para determinar o número de imagens no DPB 125 que estão marcadas como que precisam ser emitidas. O processador 110 é também configurado para comparar o número com o valor derivado de ao menos um elemento de sintaxe 14 presente no fluxo de bits 10. O processador 110 é ainda configurado para emitir a imagem, que é uma primeira imagem em ordem de saída, das imagens no DPB 125 que estão marcadas como que precisam ser emitidas se o número é maior do que o valor. O processador 110 é adicionalmente configurado para marcar a imagem como que não precisa ser emitida se o número é maior do que o valor. Isso significa que, nesta modalidade, o processador 110 é configurado para determinar o número de imagens, comparar o número, emitir a imagem e marcar a imagem após decodificar a imagem atual.

[00226] Em uma modalidade, o processador 110 é preferencialmente configurado para armazenar a imagem atual decodificada no DPB 125 e marcar a imagem atual decodificada como que precisa ser emitida ou como que não precisa ser emitida, como anteriormente descrito aqui.

[00227] O processador 110 é, em uma modalidade de implementação, preferencialmente configurado para comparar o número com um valor

sps_max_num_reorder pics [HighestTid]. Nesta modalidade de implementação, o processador 110 é configurado para emitir uma imagem no DPB 125 com um menor valor de PicOrderCntVal de todas as imagens no DPB 125 que estão marcadas como que precisam ser emitidas se o número é maior do que o valor sps_max_num_reorder pics[HighestTid].

[00228] Em outra modalidade de implementação, a imagem atual é uma imagem de acesso aleatório da sequência de vídeo. Em tal caso, o processador 110 é opcionalmente configurado para analisar o cabeçalho de fatia 12 da imagem de acesso aleatório para obter um valor de um sinalizador no_output_prior_pics flag. O processador 110 é então opcionalmente configurado para determinar o número, comparar o número, emitir a imagem e marcar a imagem se o valor do sinalizador no_output_prior_pics flag é um.

[00229] Se a sequência de vídeo é uma sequência de vídeo de multicamadas compreendendo múltiplas camadas de imagens, onde cada camada das múltiplas camadas tem um elemento de sintaxe associado definindo um respectivo valor, então o processador 110 é preferencialmente configurado para selecionar um valor derivado de um elemento de sintaxe, preferencialmente sps_max_num_reorder pics, associado com uma camada mais alta que é decodificada pelo decodificador 100 das múltiplas camadas.

[00230] Em uma modalidade, o processador 110 é preferencialmente configurado para emitir uma imagem com o menor valor de contagem de ordem de imagem das imagens no DPB 125 que estão marcadas como que precisam ser emitidas se o número é maior do que o valor.

[00231] Em uma modalidade, o processador 110 é preferencialmente configurado para emitir a imagem antes de analisar um cabeçalho de fatia de uma próxima imagem a ser decodificada da sequência de vídeo.

[00232] A Figura 17 é um diagrama de blocos esquemático de outro exemplo de implementação do decodificador 200. Esse exemplo é, em particular, adequado para uma implementação de hardware do decodificador

200. O decodificador 200 compreende uma unidade de entrada 210 configurada para receber o fluxo de bits representando imagens da sequência de vídeo e armazenar o fluxo de bits em uma memória conectada 220 compreendendo o DPB 225. O decodificador também compreende uma unidade de determinação de número 230 conectada à memória 220. Essa unidade de determinação de número 230 é configurada para determinar, após a imagem atual ter sido decodificada e armazenada no DPB 225, o número de imagens no DPB 225 que são marcadas como que não precisam ser emitidas. Um comparador 240 é conectado à unidade de determinação de número 230 e configurada para comparar o número com o valor `sps_max_num_reorder_pics[HighestTid]`. O decodificador 200 também compreende uma unidade de saída 250 conectada ao comparador 240 e preferencialmente à memória 220. A unidade de saída 250 é configurada para emitir a imagem, que é uma primeira imagem em ordem de saída, das imagens no DPB 225 que estão marcadas como que precisam ser emitidas se o número é maior do que o valor. Uma unidade de marcação 260 é conectada à memória 220 e configurada para marcar a imagem como que não precisa ser emitida se o número é maior do que o valor.

[00233] O comparador 240 é preferencialmente conectado à unidade de determinação de número 230 de modo a receber o número de imagens determinado pela unidade de determinação de número 230. Correspondentemente, a unidade de saída 250 é preferencialmente conectada ao comparador 240 de modo a receber informação de se o número determinado é maior do que o valor `sps_max_num_reorder_pics[HighestTid]` a partir do comparador 240. A unidade de marcação 260 é preferencialmente conectada à unidade de saída 250 de modo a receber informação de qual imagem que foi emitida pela unidade de saída 250.

[00234] A Figura 18 é um diagrama de blocos esquemático de ainda outro exemplo de implementação do decodificador 300. Esse exemplo é, em

particular, adequado para uma implementação de hardware do decodificador 300. O decodificador 300 compreende uma unidade de entrada 310 configurada para receber o fluxo de bits representando imagens da sequência de vídeo e armazenar o fluxo de bits em uma memória conectada 320 compreendendo o DPB 325. Uma unidade de análise 370 conectada à memória 320 é configurada para analisar o cabeçalho de fatia da imagem atual a ser decodificada da sequência de vídeo. O decodificador 300 compreende uma unidade de determinação de conjunto de imagens de referência 380 conectada à unidade de análise 370 e preferencialmente à memória 320. A unidade de determinação de conjunto de imagens de referência 380 é configurada para determinar o RPS para a imagem atual com base no cabeçalho de fatia analisado. Uma unidade de marcação 360 é conectada à memória 320 e configurada para marcar todas as imagens no DPB 325 que não estão presentes no RPS como não usadas como referência. O decodificador 300 também compreende uma unidade de saída 350 conectada à memória 320 e preferencialmente a um comparador 340. A unidade de saída 350 é configurada para emitir zero, uma ou muitas imagens, marcadas como que precisam ser emitidas, das imagens no DPB 325, onde a unidade de marcação 360 é configurada para marcar zero, uma ou muitas imagens como que não precisam ser emitidas. Uma unidade de esvaziamento de imagem 390 é conectada à memória 320 e configurada para esvaziar ou remover, do DPB 325, qualquer imagem marcada como não usada como referência e que não precisa ser emitida das imagens no DPB 325. A unidade de decodificação 305 é conectada à memória 320 e configurada para decodificar a imagem atual.

[00235] O decodificador 300 da Figura 18 também compreende uma unidade de determinação de número 330 conectada à memória 320 e preferencialmente também à unidade de análise 370 e ao comparador 340. A unidade de determinação de número 330 é configurada para determinar o número de imagens no DPB 325 que estão marcadas como que precisam ser

emitidas. O comparador previamente mencionado 340 é configurado para comparar o número com o valor derivado de ao menos um elemento de sintaxe presente no fluxo de bits. Nessa modalidade, a unidade de saída 350 é também configurada para emitir a imagem, que é uma primeira imagem em ordem de saída, das imagens no DPB 325 que estão marcadas como que precisam ser emitidas se o número é maior do que o valor. A unidade de marcação 360 é também configurada para marcar a imagem como que não precisa ser emitida se o número é maior do que o valor.

[00236] Em uma modalidade preferencial, a unidade de determinação de número 330 é configurada para determinar o número de imagens, o comparador 340 é configurado para comparar o número, a unidade de saída 350 é configurada para emitir a imagem e a unidade de marcação 360 é configurada para marcar a imagem após a unidade de decodificação 305 ter decodificado a imagem atual.

[00237] A unidade de determinação de conjunto de imagens de referência 380 é preferencialmente conectada à unidade de análise 370 de modo a receber informação presente no cabeçalho de fatia analisado e usada de modo a determinar o RPS. A unidade de determinação de conjunto de imagens de referência 380 é preferencialmente também conectada à unidade de marcação 360 de modo a fornecer o RPS ou informação das imagens listadas no RPS à unidade de marcação 360. O comparador 340 é preferencialmente conectado à unidade de determinação de número 330 de modo a receber o número de imagens determinado pela unidade de determinação de número 330. Correspondentemente, a unidade de saída 350 é preferencialmente conectada ao comparador 340 de modo a receber informação de se o número determinado é maior do que o valor do comparador 340. A unidade de marcação 360 é preferencialmente conectada à unidade de saída 350 de modo a receber informação de qual imagem que foi emitida pela unidade de saída 350.

[00238] Como indicado anteriormente, o decodificador pode alternativamente ser definido como um grupo de módulos de função, onde os módulos de função são implementados como um programa de computador rodando em um processador. [00228] A Figura 16 é um diagrama de blocos esquemático que ilustra um exemplo de um decodificador 100 compreendendo um processador 110 e uma memória associada 120.

[00239] O programa de computador residindo na memória 120 pode ser assim organizado como módulos de função apropriados configurados para executar, quando executados pelo processador 110, ao menos parte das etapas e/ou tarefas descritas acima. Um exemplo de tais módulos de função é ilustrado na Figura 19. A Figura 19 é assim um diagrama de blocos esquemático que ilustra um exemplo de um decodificador 400 compreendendo um grupo de módulos de função 410-440. Esses módulos compreendem um módulo de determinação de número 410 para determinar, após uma imagem atual de um fluxo de bits representando imagens de uma sequência de vídeo ter sido decodificada e armazenada em um DPB, um número de imagens no DPB que são marcadas como que não precisam ser emitidas. O decodificador 400 também compreende um módulo de comparação 420 para comparar o número, preferencialmente como recebido a partir do módulo de determinação de número 410, com um valor `sps_max_num_reorder pics[HighesTid]`. Um módulo de emissão 430 do decodificador 400 é para emitir uma imagem, que é uma primeira imagem em ordem de saída, das imagens no DPB que estão marcadas como que precisam ser emitidas se o número é maior do que o valor, opcionalmente, mas pre-

[00240] ferencialmente, como determinado pelo módulo de comparação 420. O decodificados 400 compreende ainda um módulo de marcação 440 para marcar a imagem como que não precisa ser emitida se o número é maior do que o valor, opcionalmente, mas preferencialmente como determinado pelo módulo de comparação 420.

[00241] A Figura 20 é um diagrama de blocos esquemático que ilustra outro exemplo de um decodificador 500 com um grupo de módulos de função 510-580. O decodificador 500 compreende um módulo de análise 510 para analisar um cabeçalho de fatia de uma imagem atual a ser decodificada de um fluxo de bits representando imagens de uma sequência de vídeo. Um módulo de determinação de conjunto de imagem de referência 520 do decodificador 500 é para determinar um RPS para a imagem atual com base no cabeçalho de fatia, preferencialmente como analisada pelo módulo de análise 510. O decodificador 500 também compreende um módulo de marcação 530 para marcar todas as imagens em um DPB que não estão presentes no RPS, opcionalmente, mas preferencialmente como determinado pelo módulo de determinação de conjunto de imagem de referência 520, como não usado como referência e um módulo de emissão 540 para emitir zero, uma ou muitas imagens, marcadas como que precisam ser emitidas, das imagens no DPB. Em uma modalidade, o módulo de marcação 530 é ainda para marcar zero, uma ou muitas imagens como que não precisam ser emitidas. O decodificador 500 ainda compreende um módulo de esvaziamento 550 para esvaziar ou remover, do DPB, qualquer imagem marcada como não usada como referência e como que não precisa ser emitida das imagens no DPB e um módulo de decodificação 560 para decodificar a imagem atual. O decodificador 500 adicionalmente compreende um módulo de determinação de número 560 para determinar um número de imagens no DPB que estão marcadas como que precisam ser emitidas e um módulo de comparação 570 para comparar o número, opcionalmente, mas preferencialmente como determinado pelo módulo de determinação de número 560, com um valor derivado de ao menos um elemento de sintaxe presente no fluxo de bits.

[00242] Em uma modalidade, o módulo de emissão 540 é ainda para emitir uma imagem, que é uma primeira imagem em ordem de saída, das imagens no DPB que estão marcadas como que precisam ser emitidas se o

número é maior do que o valor, opcionalmente, mas preferencialmente, como determinado pelo módulo de comparação 570. O módulo de marcação 530 é ainda para marcar a imagem como que não precisa ser emitida se o número é maior do que o valor, opcionalmente, mas preferencialmente como determinado pelo módulo de comparação 570.

[00243] Em uma modalidade, o módulo de determinação de número 570 determina o número de imagens, o módulo de comparação 580 compara o número, o módulo de emissão 540 emite a imagem e o módulo de marcação 530 marca a dita imagem após o módulo de decodificação 560 decodificar a imagem atual.

[00244] As modalidades do decodificador 400, 500, como mostrado nas Figuras 19 e 20, podem também ser opcionalmente operáveis para executar várias modalidades de implementação como previamente descrito aqui, tal como com relação às Figuras 9, 10, 12-15.

[00245] Em uma modalidade, o programa de computador compreende código de programa que quando executado por um processador 110, ver Figura 16, ou computador faz com que o processador 110 ou computador execute as etapas, funções, procedimentos, e/ou blocos descritos acima e mostrados nas Figuras 8-15.

[00246] O software ou programa de computador pode ser realizado como um produto de programa de computador, que é normalmente carregado ou armazenado em um meio legível por computador. O meio legível por computador pode incluir um ou mais dispositivos de memória removíveis ou não removíveis incluindo, mas não limitados a uma ROM, uma RAM, um Disco Compacto (CD), um Disco versátil Digital (DVD), uma memória de barramento serial universal (USB), um dispositivo de armazenamento de Unidade de Disco Rígido (HDD), uma memória rápida, ou qualquer outro dispositivo de memória convencional. O programa de computador pode ser assim carregado na memória de operação de um computador ou dispositivo de

processamento equivalente para execução pelo seu circuito de processamento.

[00247] O decodificador 100, 200, 300, 400, 500, como mostrado nas Figuras 16- 20, é preferencialmente um decodificador compatível com HEVC. Dever-se-ia, entretanto, notar que as modalidades não estão limitadas a HEVC.

[00248] De acordo com um aspecto, um método executado por um codificador é fornecido. No método, o codificador determinar o número de imagens no DPB que estão marcadas como que precisam ser emitidas e compara esse número com um valor que é representado por elementos de sintaxe no fluxo de bits. Se o número de imagens no DPB marcadas como que precisam ser emitidas é maior do que o valor que é derivado de elementos de sintaxe no fluxo de bits, um processo de marcação de emissão modificado é executado ou a imagem das imagens no DPB marcadas como que precisam ser emitidas que é a primeira imagem em ordem de saída é marcada como que não precisa ser emitida.

[00249] A Figura 21 é um fluxograma de um método executado por um codificador de acordo com uma modalidade. O método compreende determinar, na etapa S60, e após uma imagem atual ter sido decodificada e armazenada em um DPB, um número de imagens no DPB que estão marcadas como que precisam ser emitidas. Uma próxima etapa S61 compreende comparar o número com um valor `sps_max_num_reorder pics[HighestTid]`, onde `HighestTid` especifica uma camada mais alta que é decodificada pelo codificador de uma sequência de vídeo. Se o número é maior do que o valor, o método continua para a etapa S62, que compreende marcar uma imagem, que é uma primeira imagem em ordem de saída, das imagens em DPB que estão marcadas como que precisam ser emitidas como que não precisa ser emitida.

[00250] O método, como executado no codificador na Figura 21, é basicamente similar ao método correspondente executado em um

decodificador correspondente, ver Figura 8. Uma diferença é, então, que o codificador geralmente não emite quaisquer imagens, que é executado no decodificador. Portanto, a modalidade do método mostrado na Figura 21 carece preferencialmente da etapa de emissão S3 do método correspondente executado por um decodificador na Figura 8.

[00251] Em uma modalidade, quando a imagem atual foi decodificada, duas marcações ocorrem preferencialmente, uma para a saída e uma para referência. Portanto, em uma modalidade, essas marcações são preferencialmente feitas antes da etapa S60, uma vez que afeta o número de imagens que estão marcadas como que precisam ser emitidas. Se a imagem atual tem um PicOutputFlag = 1, a imagem atual é preferencialmente contada também.

[00252] Em uma modalidade, se o número não é maior do que o valor determinado na comparação da etapa S61, o método preferencialmente termina e nenhuma imagem é marcada. Assim, em tal caso, a etapa S62 é omitida e não executada, ver a linha tracejada direita.

[00253] Em uma modalidade, as etapas S61-S62 da Figura 21 poderia ser executadas somente uma vez após uma imagem atual ter sido decodificada e armazenada no DPB. Alternativamente, o circuito formado pelas etapas S61-S62, ver linha tracejada esquerda, poderia ser executado até que o número de imagens no DPB que estão marcadas como que precisam ser emitidas não é mais maior do que o valor `sps_max_num_reorder pics[HighestTid]`. Cada vez que o circuito das etapas S61- S62 é executado, o número de imagens no DPB que estão marcadas como que precisam ser emitidas é reduzido em um através da marcação na etapa S62.

[00254] A Figura 22 é um fluxograma de um método executado por um codificador de acordo com outra modalidade. O método compreende marcar, na etapa S70, todas as imagens em um DPB que não estão presentes em um RPS para uma imagem atual de uma sequência de vídeo configurada como

não usada como referência. O método também compreende marcar, na etapa S71, zero, uma ou muitas imagens, marcadas como que precisam ser emitidas, das imagens em DPB como que não precisam ser emitidas. Qualquer imagem marcada como não usada como referência e como que não precisa ser emitida das imagens no DPB é esvaziada ou removida na etapa S72 do DPB. Uma etapa seguinte S73 compreende decodificar a imagem atual. O método compreende ainda determinar, na etapa S74, um número de imagens no DPB que estão marcadas como que precisam ser emitidas e comparar, na etapa S75, o número com um valor derivado de ao menos um elemento de sintaxe definido. Se esse número é maior do que o valor, o método continua para a etapa S76, que compreende marcar uma imagem, que é uma primeira imagem em ordem de saída, das imagens no DPB que estão marcadas como que precisam ser emitidas como que não precisa ser emitida. Em uma modalidade preferencial, determinar o número de imagens na etapa S74, comparar o número na etapa S75, e marcar a imagem na etapa S76 são executados após decodificar a imagem atual na etapa S73.

[00255] O método, como executado no codificador na Figura 22, é basicamente similar ao método correspondente executado em um decodificador correspondente, ver Figura 11. Uma diferença é, então, que o codificador geralmente não emite quaisquer imagens, que é executado no decodificador. Portanto, a modalidade do método mostrado na Figura 22 preferencialmente carece das etapas de emissão S33 e S38 do método correspondente executado por um decodificador na Figura 11. Ademais, no método da Figura 22, o codificador não recebe qualquer fluxo de bits e assim não precisa analisar um cabeçalho de fatia de modo a obter informação para determinar o RPS para a imagem atual. Em claro contraste, o próprio codificador gera e determinar o RPS para a imagem atual. Portanto, as etapas S30 e S31 do método correspondente executado por um decodificador na Figura 11 são tipicamente não executadas pelo codificador.

[00256] Em uma modalidade, se o número não é maior do que o valor determinado na comparação da etapa S75, o método preferencialmente termina e nenhuma imagem é marcada. Assim, em tal caso, a etapa S76 é omitida e não executada, ver linha tracejada direita.

[00257] Em uma modalidade, as etapas S75-S76 poderiam ser executadas somente uma vez após uma imagem atual ter sido decodificada e armazenada no DPB. Alternativamente, o circuito formado pelas etapas S75-S76, ver linha tracejada esquerda, poderia ser executado até que o número de imagens no DPB que estão marcadas como que precisam ser emitidas não seja mais maior do que o valor. Cada vez que o circuito das etapas S75-S76 é executado, o número de imagens no DPB que estão marcadas como que precisam ser emitidas é reduzido em um através da marcação na etapa S76.

[00258] As modalidades discutidas anteriormente em conjunto com os métodos executados por um decodificador podem ser também executadas pelo codificador.

[00259] Por exemplo, um método de codificação pode conter e/ou um codificador pode ser configurado para executar as seguintes etapas ordenadas de acordo com uma modalidade:

A imagem P é decodificada.

[00260] Após a imagem P ser codificada, o codificador determina o número de imagens no DPB que estão marcadas como que "precisam ser emitidas" e compara esse número com o valor que pode ser derivado de elementos de sintaxe no fluxo de bits. Dever-se-ia notar que isso se refere ao DPB no codificador. O estado do DPB no codificador e decodificador é o mesmo. A comparação no decodificador será exatamente a mesma. A especificação HEVC especifica o que o decodificador fará, mas o codificador terá que manter o rastreamento disso, isto é, o codificador faz o que o decodificador faz.

[00261] Se o número de imagens no DPB marcadas como que "precisam ser emitidas" é maior do que o valor que pode ser derivado dos elementos de sintaxe no fluxo de bits, a imagem das imagens no DPB marcadas como que "precisam ser emitidas" que é a primeira imagem em ordem de saída é marcada como que "não precisa ser emitida". A imagem pode ser opcionalmente emitida a partir do decodificador.

[00262] A próxima imagem Q é codificada.

[00263] As etapas, funções, procedimentos, módulos e/ou blocos descritos acima em conjunto com as Figuras 21 e 22 podem ser implementados em hardware usando qualquer tecnologia convencional, tal como tecnologia de circuito discreto ou de circuito integrado, incluindo tanto circuito de propósito geral e circuito de aplicação específica.

[00264] Os exemplos particulares incluem um ou mais processadores de sinal digital adequadamente configurados e outros circuitos eletrônicos conhecidos, por exemplo, portas lógicas discretas interconectadas para executar uma função especializada, ou ASICs.

[00265] Alternativamente, ao menos algumas das etapas, funções, procedimentos, módulos e/ou blocos descritos acima em conjunto com as Figuras 21 e 22 podem ser implementados em software tal como um programa de computador para execução por circuito de processamento adequado incluindo um ou mais processadores.

[00266] O processador é capaz de executar instruções de software contidas em um programa de computador armazenado em um produto de programa de computador, por exemplo, na forma de memórias. O respectivo produto de programa de computador pode ser uma memória sendo qualquer combinação de RAM e ROM. A respectiva memória compreende armazenamento persistente, que, por exemplo, pode ser qualquer única memória ou uma combinação de memória magnética, memória óptica, memória de estado sólido ou até memória montada remotamente.

[00267] O fluxograma ou fluxogramas apresentados acima e mostrados nas Figuras 21 e 22 podem então ser considerados como um fluxograma ou fluxogramas de computador, quando executados por um ou mais processadores. Um aparelho correspondente pode ser definido como um grupo de módulos de função, onde cada etapa executada pelo processador corresponde a um módulo de função. Nesse caso, os módulos de função são implementados como um programa de computador rodando no processador.

[00268] Exemplos de circuito de processamento incluem, mas não estão limitados a um ou mais microprocessadores, um ou mais DSPs, uma ou mais CPUs, hardware de aceleração de vídeo, e/ou qualquer circuito lógico programável adequado tal como um ou mais FPGAs, ou um ou mais PLCs.

[00269] Dever-se-ia também entender que pode ser possível reutilizar as capacidades de processamento gerais de qualquer dispositivo ou unidade convencional na qual a tecnologia proposta é implementada. Pode também ser possível reutilizar software existente, por exemplo, reprogramando o software existente ou adicionando novos componentes de software.

[00270] De acordo com um aspecto, um codificador configurado para executar um método é fornecido. O codificador é configurado para determinar, após uma imagem atual ter sido decodificada e armazenada em um DPB, um número de imagens no DPB que estão marcadas como que precisam ser emitidas. O codificador é também configurado para comparar o número com um valor `sps_max_num_reorder_pics[HighestTid]`. O codificador é ainda configurado para marcar uma imagem, que é uma primeira imagem em ordem de saída, das imagens no DPB que estão marcadas como que precisam ser emitidas como que não precisa ser emitida se o número é maior do que o valor.

[00271] O codificador 600 compreende, em uma modalidade, um processador 610 configurado para executar as etapas de método previamente descritas aqui, ver Figura 21. O decodificador 600 pode também compreender

uma memória 620 conectada ao processador 610, ver Figura 23.

[00272] A Figura 23 é um diagrama de blocos esquemático de um codificador 600 de acordo com uma modalidade. O codificador 600 é configurado para codificar imagens de uma sequência de vídeo em um fluxo de bits 10. O codificador 600 compreende um processador 610 e uma memória 620. A memória 620 compreende um DPB 625. Em uma modalidade, o processador 610 é configurado para determinar, após a imagem atual ter sido decodificada e armazenada no DPB 625, o número de imagens no DPB 625 que estão marcadas como que precisam ser emitidas. O processador 610 é também configurado para comparar o número com o valor `sps_max_num_reorder_pics[HighestTid]`. O processador 610 é ainda configurado para marcar a imagem, que é uma primeira imagem em ordem de saída, das imagens no DPB 625 que estão marcadas como que precisam ser emitidas como que não precisa ser emitida se o número é maior do que o valor.

[00273] Na Figura 23, o codificador 600 foi ilustrado como compreendendo um processador 610. Esse processador 610 poderia ser implementado como um único processador ou múltiplos processadores, tal como na forma de um circuito de processamento.

[00274] A Figura 23 ilustra assim uma implementação em computador do codificador 600. Nesse exemplo particular, ao menos algumas das etapas, funções, procedimentos, módulos e/ou blocos descritos acima são implementados em um programa de computador, que é carregado na memória 620 para execução pelo processador 610. O processador 610 e a memória 620 são interconectados entre si para habilitar a execução de software normal. Um dispositivo de entrada/saída opcional (não mostrado) pode também ser interconectado ao processador 610 e/ou à memória 620 para possibilitar a entrada de imagens a serem codificadas e emitidas do fluxo de bits 10.

[00275] Outro aspecto das modalidades refere-se a um codificador

configurado para marcar todas as imagens em um DPB que não estão presentes em um RPS para uma imagem atual de uma sequência de vídeo como não usadas como referência. O codificador é também configurado para marcar zero, uma ou muitas imagens, marcadas como que precisam ser emitidas, das imagens no DPB como que não precisam ser emitidas e esvazia ou remove, do DPB, qualquer imagem marcada como não usada como referência e como que não precisa ser emitida das imagens no DPB. O codificador é ainda configurado para decodificar a imagem atual e determinar um número de imagens no DPB que estão marcadas como que precisam ser emitidas. O codificador é adicionalmente configurado para comparar o número com um valor derivado de ao menos um elemento de sintaxe definido e marcar uma imagem, que é uma primeira imagem em ordem de saída, das imagens no DPB que estão marcadas como que precisam ser emitidas como que não precisa ser emitida se o número é maior do que o valor. Em uma modalidade, o codificador é configurado para determinar o número de imagens, comparar o número e marcar a imagem após decodificar a imagem atual.

[00276] O codificador 600 compreende, em uma modalidade, um processador 610 configurado para executar as etapas do método previamente discutido aqui, ver Figura 22. O decodificador 600 pode também compreender uma memória 620 conectada ao processador 610, ver Figura 23.

[00277] A Figura 23 é um diagrama de blocos esquemático de um codificador 600 de acordo com uma modalidade. O codificador 600 é configurado para codificar imagens de uma sequência de vídeo em um fluxo de bits 10. O codificador 600 compreende um processador 610 e uma memória 620. A memória 620 compreende um DPB 625. Em uma modalidade, o processador 610 é configurado para marcar todas as imagens no DPB 625 que não estão presentes no RPS para a imagem atual da sequência de vídeo como não usadas como referência. O processador 610 é

também configurado para marcar zero, uma ou muitas imagens, marcadas como que precisam ser emitidas, das imagens no DPB 625 como que não precisam ser emitidas. O processador 610 é ainda configurado para esvaziar ou remover, do DPB 625, qualquer imagem marcada como não usada como referência e que não precisa ser emitida das imagens no DPB 625. Nesta modalidade, o processador 610 é também configurado para decodificar a imagem atual e determinar um número de imagens no DPB 625 que estão marcadas como que precisam ser emitidas. O processador 610 é ainda configurado para comparar o número com um valor derivado de ao menos um elemento de sintaxe definido. O processador 610 é adicionalmente configurado para marcar uma imagem, que é uma primeira imagem em ordem de saída, das imagens no DPB 625 que estão marcadas como que precisam ser emitidas como que não precisa ser emitida se o número é maior do que o valor. Nessa modalidade, o processador 610 é configurado para determinar o número de imagens, comparar o número e marcar a imagem após decodificar a imagem atual.

[00278] A Figura 24 é um diagrama de blocos esquemático de outro exemplo de implementação do codificador 700. Esse exemplo é, em particular, adequado para uma implementação em hardware do codificador 700. O codificador 700 compreende uma unidade de determinação de número 730 conectada a uma memória 720 compreende o DPB 725. A unidade de determinação de número 730 é configurada para determinar, após a imagem atual da sequência de vídeo ter sido decodificada e armazenada no DPB 725, o número de imagens no DPB 725 que estão marcadas como que precisam ser emitidas. O codificador 700 também compreende um comparador 740 conectado à unidade de determinação de número 730 e preferencialmente à memória 720. O comparador 740 é configurado para comparar o número com o valor `sps_max_num_reorder pics[HighestTid]`. A unidade de marcação 760 do codificador é conectada à memória 720 e configurada para marcar a

imagem, que é uma primeira imagem em ordem de saída, das imagens no DPB 725 que estão marcadas como que precisam ser emitidas, como que não precisa ser emitida se o número é maior do que o valor.

[00279] O comparador 740 é preferencialmente conectado à unidade de determinação de número 730 de modo a receber o número de imagens determinado pela unidade de determinação de número 730.

[00280] A Figura 25 é um diagrama de blocos esquemático de ainda outro exemplo de implementação do codificador 800, em particular, adequado para uma implementação em hardware. O codificador 800 compreende uma unidade de marcação 860 conectada a uma memória 820 compreendendo o DPB 825. A unidade de marcação 860 é configurada para (i) marcar todas as imagens no DPB 825 que não estão presentes no RPS para a imagem atual da sequência de vídeo como não usadas como referência e (ii) marcar zero, uma ou muitas imagens, marcadas como que precisam ser emitidas, das imagens no DPB 825 como que não precisam ser emitidas. Uma unidade de esvaziamento de imagem 890 é conectada à memória 820 e é configurada para esvaziar ou remover, do DPB 825, qualquer imagem marcada como não usada como referência e que não precisa ser emitida das imagens no DPB 825. O codificador 800 também compreende uma unidade de decodificação 805 conectada à memória 820 e configurada para decodificar a imagem atual. Uma unidade de determinação de número 830 do codificador 800 é conectada à memória 830 e é configurada para determinar o número de imagens no DPB 825 que são marcadas como que precisam ser emitidas. O codificador 800 compreende ainda um com-parador 840 conectado à unidade de determinação de número 830 e preferencialmente à memória 820. O comparador 840 é configurado para comparar o número com o valor derivado de ao menos um elemento de sintaxe definido. Nessa modalidade, a unidade de marcação 860 é também configurada para marcar a imagem, que é uma primeira imagem em ordem de saída, das imagens no DPB 825 que estão marcadas como que

precisam ser emitidas, como que não pode ser emitida se o número é maior do que o valor. Em uma modalidade preferencial, a unidade de determinação de número 830 é configurada para determinar o número de imagens, o comparador 840 é configurado para comparar o número e a unidade de marcação 840 é configurada para marcar a imagem após a unidade de decodificação 805 ter decodificado a imagem atual.

[00281] O comparador 840 é preferencialmente conectado à unidade de determinação de número 830 de modo a receber o número de imagens determinado pela unidade de determinação de número 830.

[00282] Como indicado anteriormente, o codificador pode ser alternativamente definido como um grupo de módulos de função, onde os módulos de função são implementados como um programa de computador rodando em um processador. [00269] A Figura 23 é um diagrama de blocos esquemático que ilustra um exemplo de um codificador 600 compreendendo um processador 610 e uma memória associada 620.

[00283] O programa de computador residindo na memória 620 pode ser assim organizado como módulos de função apropriados configurados para executar, quando executados pelo processador 610, ao menos parte das etapas e/ou tarefas descritas acima. Um exemplo de tais módulos de função é ilustrado na Figura 26. A Figura 26 é assim um diagrama de blocos esquemático que ilustra um exemplo de um codificador 900 compreendendo um grupo de módulos de função 910-930. Esses módulos compreendem um módulo de determinação de número 910 para determinar, após uma imagem atual de um fluxo de bits representando imagens de uma sequência de vídeo ter sido decodificada e armazenada em um DPB, um número de imagens no DPB que estão marcadas como que precisam ser emitidas. O codificador 900 também compreende um módulo de comparação 920 para comparar o número, opcionalmente, mas preferencialmente como determinar pelo módulo de determinação de número 910, com um valor `sps_max_num_reorder pics`[

HighestTid]. O codificador 900 compreende ainda um módulo de marcação 930 para marcar uma imagem, que é uma primeira imagem em ordem de saída, das imagens no DPB que estão marcadas como que precisam ser emitidas, como que não precisa ser emitida se o número é maior do que o valor, opcionalmente, mas preferencialmente como determinado pelo módulo de comparação 920.

[00284] A Figura 27 é um diagrama de blocos esquemático que ilustra outro exemplo de um codificador 1000 com um grupo de módulos de função 1000-1050. O codificador 1000 compreende um módulo de marcação 1010 para marcar todas as imagens em um DPB que não estão presentes em um conjunto RPS para uma imagem atual de uma sequência de vídeo como não usadas como referência e marcar zero, uma ou muitas imagens, marcadas como que precisam ser emitidas, das imagens no DPB como que não precisam ser emitidas. O codificador 1000 compreende ainda um módulo de esvaziamento 1020 para esvaziar ou remover, do DPB, qualquer imagem marcada como não usada como referência e que não precisa ser emitida das imagens no DPB e um módulo de decodificação 1030 para decodificar a imagem atual. O codificador 1000 adicionalmente compreende um módulo de determinação de número 1040 para determinar um número de imagens no DPB que estão marcadas como que precisam ser emitidas e um módulo de comparação 1050 para comparar o número, opcionalmente, mas preferencialmente como determinado pelo módulo de determinação de número 1040, com um valor derivado de ao menos um elemento de sintaxe presente no fluxo de bits.

[00285] Em uma modalidade, o módulo de marcação 1010 é ainda para marcar a imagem, que é uma primeira imagem em ordem de saída, das imagens no DPB que são marcadas como que precisam ser emitidas, como que não precisa ser emitida se o número é maior do que o valor, opcionalmente, mas preferencialmente como determinado pelo módulo de

comparação 1050.

[00286] Em uma modalidade, o módulo de determinação de número 1040 determina o número de imagens, o módulo de comparação 1050 compara o número e o módulo de marcação 1010 marca a dita imagem após o módulo de decodificação 1030 decodificar a imagem atual.

[00287] Em uma modalidade, o programa de computador compreende código de programa que quando executado por um processador 610, ver Figura 23, ou computador faz com que o processador 610 ou computador execute as etapas, funções, procedimentos, e/ou blocos descritos acima e mostrados nas Figuras 21 ou 22.

[00288] O software ou programa de computador pode ser realizado como um produto de programa de computador, que é normalmente carregado ou armazenado em um meio legível por computador. O meio legível por computador pode incluir um ou mais dispositivos de memória removíveis ou não removíveis incluindo, mas não limitados a uma ROM, uma RAM, um CD, um DVD, uma memória USB, um dispositivo de armazenamento HDD, uma memória rápida, ou qualquer outro dispositivo de memória convencional. O programa de computador pode ser assim carregado na memória operacional de um computador ou dispositivo de processamento equivalente para execução por seu circuito de processamento.

[00289] O codificador 600, 700, 800, 900, 1000, como mostrado nas Figuras 23- 27, é preferencialmente um codificador compatível com HEVC. Dever-se-ia notar, entretanto, que as modalidades não estão limitadas a HEVC.

[00290] O decodificador 100, 200, 300, 400, 500 em qualquer uma das Figuras 16-20 e o codificador 600, 700, 800, 900, 1000 em qualquer uma das Figuras 23-27 podem ser implementados, por exemplo, em um terminal móvel. Por exemplo, o de-codificador 100, 200, 300, 400, 500 pode, por exemplo, estar localizado em um receptor em uma câmera de vídeo oi

qualquer outro dispositivo para exibir um fluxo de vídeo. O codificador 600, 700, 800, 900, 1000 pode, por exemplo, estar localizado em um transmissor em uma câmera de vídeo, por exemplo, em um dispositivo móvel.

[00291] A Figura 28 é um diagrama de blocos esquemático de um terminal móvel 1100 de acordo com uma modalidade. O terminal móvel 1100 compreende um de-codificador 100, 200, 300, 400, 500 de acordo com as modalidades e/ou um codificador 600, 700, 800, 900, 1000 de acordo com as modalidades, tal como um decodificador como mostrado em qualquer uma das Figuras 16-20 e/ou um codificador como mostrado em qualquer uma das figuras 23-27.

[00292] O terminal móvel 1100 também compreende preferencialmente uma unidade de entrada e saída (110) 1110 para possibilitar a comunicação, tipicamente comunicação sem fio, mas alternativamente, ou em adição, comunicação por fio, com unidades externas. A unidade 110 1110 poderia ser implementada como um transmissor e um receptor, ou um transceptor, para comunicação sem fio. Alternativamente, a unidade 110 1110 poderia ser uma unidade 110 geral ou porta 1110 capaz de conduzir a comunicação sem fio. Se o terminal móvel 1110 é implementado com um codificador 600, 700, 800, 900, 1000, a unidade 110 1110 é preferencialmente configurada para transmitir ou emitir um fluxo de bits representando uma sequência de vídeo codificada como gerada pelo codificador 600, 700, 800, 900, 1000. Correspondentemente, se o terminal móvel 1100 compreende um decodificador 100, 200, 300, 400, 500, a unidade 110 1110 é preferencialmente configurada para receber ou inserir um fluxo de bits representando uma sequência de vídeo codificada.

[00293] O terminal móvel 1100 compreende uma memória 1120 configurada para armazenar imagens codificadas de uma sequência de vídeo codificada. Essas imagens codificadas podem ter sido geradas pelo próprio terminal móvel 1100. Em tal caso, o terminal móvel 1100 compreende

preferencialmente um mecanismo ou gravador de mídia (não mostrado) junto com um codificador conectado 600, 700, 800, 900, 1000. Alternativamente, a sequência de vídeo codificada é gerada por algum outro dispositivo e transmitida para o terminal móvel 1100.

[00294] As imagens codificadas são levadas da memória 1120 para um decodificador 100, 200, 300, 400, 500. O decodificador 100, 200, 300, 400, 500 então decodifica as imagens codificadas em imagens decodificadas. As imagens decodificadas são fornecidas a um reproduzidor de mídia 1130 que é configurado para renderizar as imagens decodificadas da sequência de vídeo em dados de vídeo que são exibidos em uma tela 1140 de/ou conectada ao terminal móvel 1100.

[00295] Na Figura 28, o terminal móvel 1100 foi ilustrado como compreendendo tanto o decodificador 100, 200, 300, 400, 500 quanto o reproduzidor de mídia 1130, com o decodificador 100, 200, 300, 400, 500 implementado como uma parte do reproduzidor de mídia 1130. Isso deveria, entretanto, ser visto meramente como um exemplo ilustrativo, mas não limitante de uma modalidade de implementação para o terminal móvel 1100. Implementações distribuídas são também possíveis quando o decodificador 100, 200, 300, 400, 500 e o reproduzidor de mídia 1130 são fornecidos em dois dispositivos fisicamente separados que são possíveis e dentro do escopo do terminal móvel 1100 como usado aqui. A tela 1140 poderia também ser fornecida como um dispositivo separado conectado ao terminal móvel 1100, onde o processamento de dados real está acontecendo.

[00296] O terminal móvel 1100 pode ser qualquer dispositivo tendo funções de decodificação de mídia que opera em uma sequência de vídeo codificada de imagens codificadas para decodificar assim as imagens e tornar os dados de vídeo disponíveis. Exemplos não limitantes de tais terminais móveis 1100 incluem telefones móveis e outros reproduzidores de mídia portáteis, computadores, decodificadores, consoles de jogos, etc.

[00297] O codificador 600, 700, 800, 900, 1000 das modalidades, tal como o mostrado em qualquer uma das Figuras 23-27, e/ou o decodificador 100, 200, 300, 400, 500 das modalidades, tal como o mostrado em qualquer uma das Figuras 16-20, podem ser implementados em um nó de rede 2 como mostrado na Figura 29.

[00298] Como ilustrado na Figura 29, o codificador 600, 700, 800, 900, 1000 e/ou o decodificador 100, 200, 300, 400, 500 podem ser implementados em um nó de rede 2 em uma rede de comunicação 1 entre uma unidade de envio 3 e uma unidade de recebimento 4. Tal nó de rede 2 pode ser um dispositivo para converter vídeo entre, por exemplo, diferentes resoluções de vídeo, taxas de quadro, qualidades, taxas de bits, e padrões de codificação. O nó de rede 2 pode estar na forma de uma estação rádio base, um W-13 ou qualquer nó de rede em uma rede de comunicação 1, tal como uma rede baseada em rádio.

[00299] O codificador e/ou decodificador das modalidades podem ser também fornecidos em qualquer elemento que opera em um fluxo de bits, tal como um Elemento de Rede Sensível à Mídia.

[00300] As modalidades não estão limitadas a HEVC, mas podem ser aplicadas a qualquer extensão de HEVC, tal como uma extensão escalável ou extensão de múltiplas vistas ou a um codec de vídeo de diferença.

[00301] As modalidades descritas acima são entendidas como alguns exemplos ilustrativos da presente invenção. Os versados na técnica entendem que várias modificações, combinações e mudanças podem ser feitas às modalidades sem abandonar o escopo da presente invenção. Em particular, diferentes soluções parciais nas diferentes modalidades podem ser combinadas em outras configurações, onde tecnicamente possível. O escopo da presente invenção é, entretanto, definido pelas reivindicações em anexo.

REIVINDICAÇÕES

1. Método executado por um decodificador (100, 200, 400), caracterizado pelo fato de que compreende:

determinar (S1), após uma imagem atual ter sido decodificada e armazenada em um armazenador temporário de imagem decodificada (125, 225), um número de imagens no armazenador temporário de imagem decodificada (125, 225) que estão marcadas como que precisam ser emitidas, a dita imagem atual pertencente a uma camada que não a camada mais alta que é decodificada pelo dito decodificador (100, 200, 400) de uma sequência de vídeo de multicamadas compreendendo múltiplas camadas de imagens;

selecionar (S50) um valor `sps_max_num_reorder_pics[HighestTid]`, onde `HighestTid` especifica a dita camada mais alta, e o dito valor `sps_max_num_reorder_pics[HighestTid]` é derivado de um elemento de sintaxe associado com a dita camada mais alta, cada camada *i* das ditas múltiplas camadas tem um elemento de sintaxe associado definindo um respectivo valor `sps_max_num_reorder_pics[i]`, e o dito valor `sps_max_num_reorder_pics[i]` indica um número máximo permitido de imagens que podem preceder qualquer imagem na dita sequência de vídeo multicamadas na ordem de decodificação e seguem essa imagem em ordem de saída quando `HighestTid` é igual a *i*;

comparar (S2) o dito número de imagens no dito armazenador temporário de imagem decodificada (125, 225) que são marcados como que precisam ser emitidas em relação ao dito valor `sps_max_num_reorder_pics[HighestTid]`;

emitir (S3) uma imagem, que é uma primeira imagem em ordem de saída das imagens no dito armazenador temporário de imagem decodificada (125, 225) que estão marcadas como que precisam ser emitidas, se o dito número de imagens no dito armazenador temporário de imagem decodificada (125, 225) que são marcadas como que precisam ser emitidas é

maior do que o dito valor `sps_max_num_reorder_pics[HighestTid]`; e

marcar (S4) a dita imagem como que não precisa ser emitida se o número de imagens no dito armazenador temporário de imagem decodificada (125, 225) que estão marcadas como que precisam ser emitidas é maior do que o dito valor `sps_max_num_reorder_pics[HighestTid]`.

2. Método, de acordo com a reivindicação 1, caracterizado adicionalmente pelo fato de que compreende:

analisar (S10) um cabeçalho de fatia da dita imagem atual a ser decodificada da dita sequência de vídeo de multicamadas;

determinar (S11) um conjunto de imagens de referência para a dita imagem atual com base no dito cabeçalho de fatia analisado;

marcar (S12) todas as imagens no dito armazenador temporário de imagens decodificadas (125, 225) que não estão presentes no dito conjunto de imagens de referência como não usadas como referência;

emitir (S13) zero, uma ou muitas imagens, marcadas como que precisam ser emitidas, das ditas imagens no dito armazenador temporário de imagens decodificadas (125, 225) e marcar as ditas zero, uma ou muitas imagens como que não precisam ser emitidas;

esvaziar (S14), do dito armazenador temporário de imagens decodificadas (125, 225), qualquer imagem marcada como não usada como referência e que não precisa ser emitida das ditas imagens no dito armazenador temporário de imagens decodificadas (125, 225); e

decodificar (S15) a dita imagem atual, onde determinar (S1) o dito número de imagens, comparar (S2) o dito número, emitir (S3) a dita imagem e marcar (S4) a dita imagem são executados após decodificar (S15) a dita imagem atual.

3. Método executado por um decodificador (100, 300, 500), caracterizado pelo fato de que compreende:

analisar (S30) um cabeçalho de fatia de uma imagem atual a

ser decodificada de uma sequência de vídeo de multicamadas compreendendo múltiplas camadas de imagens, a dita imagem atual pertencente a uma camada que não a camada mais alta que é decodificada pelo dito decodificador (100, 200, 400) da dita sequência de vídeo de multicamadas;

determinar (S31) um conjunto de imagens de referência para a dita imagem atual com base no dito cabeçalho de fatia analisado;

marcar (S32) todas as imagens em um armazenador temporário de imagens decodificadas (125, 325) que não estão presentes no dito conjunto de imagens de referência como não usadas como referência;

emitir (S33) zero, uma ou muitas imagens, marcadas como que precisam ser emitidas, das ditas imagens no dito armazenador temporário de imagens decodificadas (125, 325), e marcar as ditas zero, uma ou muitas imagens como que não precisam ser emitidas;

esvaziar (S34) qualquer imagem, a partir do dito armazenador temporário de imagens decodificadas (125, 325), marcada como não usada como referência e que não precisa ser emitida das ditas imagens no dito armazenador temporário de imagens decodificadas (125, 325);

decodificar (S35) a dita imagem atual;

determinar (S36) um número de imagens no dito armazenador temporário de imagens decodificadas (125, 325) que estão marcadas como que precisam ser emitidas;

selecionar (S50) um valor `sps_max_num_reorder_pics[HighestTid]`, onde `HighestTid` especifica a dita camada mais alta, e o dito valor `sps_max_num_reorder_pics[HighestTid]` é derivado de um elemento de sintaxe associado com a dita camada mais alta, cada camada `i` das ditas múltiplas camadas tem um elemento de sintaxe associado definindo um respectivo valor `sps_max_num_reorder_pics[i]`, e o dito valor `sps_max_num_reorder_pics[i]` indica um número máximo permitido de imagens que podem preceder qualquer imagem na dita sequência de vídeo

multicamadas na ordem de decodificação e seguem essa imagem em ordem de saída quando HighestTid é igual a i;

comparar (S37) o dito número de imagens no dito armazenador temporário de imagens decodificadas (125, 325) que estão marcadas como que precisam ser emitidas com o dito valor `sps_max_num_reorder_pics[HighestTid]`;

emitir (S38) uma imagem, que é uma primeira imagem em ordem de saída, das ditas imagens no dito armazenador temporário de imagens decodificadas (125, 325) que estão marcadas como que precisam ser emitidas se o dito número de imagens no dito armazenador temporário de imagens decodificadas (125, 325) que estão marcadas como que precisam ser emitidas é maior do que o dito valor `sps_max_num_reorder_pics[HighestTid]`; e

marcar (S39) a dita imagem como que não precisa ser emitida se o dito número de imagens no dito armazenador temporário de imagens decodificadas (125, 325) é maior do que o dito valor `sps_max_num_reorder_pics[HighestTid]`, em que determinar (S36) o dito número de imagens, comparar (S37) o dito número, emitir (S38) a dita imagem e marcar (S39) a dita imagem são executados após decodificar (S35) a dita imagem atual.

4. Método, de acordo com a reivindicação 3, caracterizado adicionalmente pelo fato de que compreende:

armazenar (S40) a dita imagem atual decodificada no dito armazenador temporário de imagens decodificadas (125, 325); e

marcar (S41) a dita imagem atual decodificada como que precisa ser emitida ou como que não precisa ser emitida, onde determinar (S1, S36) o dito número de imagens é executado após marcar (S41) a dita imagem atual decodificada.

5. Método, de acordo com qualquer uma das reivindicações 1 a

4, caracterizado pelo fato de que:

emitir (S3, S38) a dita imagem compreende emitir (S3, S38) uma imagem no dito armazenador temporário de imagens decodificadas (125, 225, 325) com o menor valor de PicOrderCntVal de todas as imagens no dito armazenador temporário de imagens decodificadas (125, 225, 325) que estão marcadas como que precisam ser emitidas, se o dito número de imagens no dito armazenador temporário de imagens decodificadas é maior do que o dito valor `sps_max_num_reorder_pics[HighestTid]`, onde PicOrderCntVal representa um valor de contagem de ordem de imagens.

6. Método, de acordo com qualquer uma das reivindicações 1 a 5, caracterizado pelo fato de que emitir (S3, S38) a dita imagem compreende emitir (S3, S38) uma imagem com o menor valor de contagem de ordem de imagens das ditas imagens no dito armazenador temporário de imagens decodificadas (125, 225, 325) que estão marcadas como que precisam ser emitidas se o dito número de imagens no dito armazenador temporário de imagens decodificadas (125, 225, 325) que estão marcadas como que precisam ser emitidas é maior do que o dito valor `sps_max_reorder_pics[HighestTid]`.

7. Método, de acordo com qualquer uma das reivindicações 1 a 6, caracterizado pelo fato de que emitir (S3, S38) a dita imagem é executado antes de analisar um cabeçalho de fatia de uma próxima imagem a ser decodificada da dita sequência de vídeo de multicamadas.

8. Decodificador (100, 200), caracterizado pelo fato de que é configurado para:

determinar, após uma imagem atual de um fluxo de bits (10) representando imagens de uma sequência de vídeo de multicamadas compreendendo múltiplas camadas de imagens ter sido decodificada e armazenada em um armazenador temporário de imagem decodificada (125, 225), um número de imagens no dito armazenador temporário de imagem

decodificada (125, 225) que estão marcadas como que precisam ser emitidas, a dita imagem atual pertencente a uma camada que não a camada mais alta que é decodificada pelo dito decodificador (100, 200) da dita sequência de vídeo de multicamadas;

selecionar um valor $sps_max_num_reorder_pics[HighestTid]$, onde $HighestTid$ especifica a dita camada mais alta, e o dito valor $sps_max_num_reorder_pics[HighestTid]$ é derivado de um elemento de sintaxe associado com a dita camada mais alta, cada camada i das ditas múltiplas camadas tem um elemento de sintaxe associado definindo um respectivo valor $sps_max_num_reorder_pics[i]$, e o dito valor $sps_max_num_reorder_pics[i]$ indica um número máximo permitido de imagens que podem preceder qualquer imagem na dita sequência de vídeo de multicamadas na ordem de decodificação e seguem essa imagem em ordem de saída quando $HighestTid$ é igual a i ;

comparar o dito número de imagens no dito armazenador temporário de imagens decodificadas (125, 325) que estão marcadas como que precisam ser emitidas com o dito valor $sps_max_num_reorder_pics[HighestTid]$;

emitir uma imagem, que é uma primeira imagem em ordem de saída, das ditas imagens no dito armazenador temporário de imagem decodificada (125, 225) que estão marcadas como que precisam ser emitidas, se o dito número de imagens no dito decodificador temporário de imagens decodificadas (125, 225) que estão marcadas como que precisam ser emitidas é maior do que o dito valor $sps_max_num_reorder_pics[HighestTid]$; e

marcar a dita imagem como que não precisa ser emitida se o dito número de imagens no dito armazenador temporário de imagens decodificadas (125, 225) que estão marcadas como que precisam ser emitidas é maior do que o dito valor $sps_max_num_reorder_pics[HighestTid]$.

9. Decodificador, de acordo com a reivindicação 8,

caracterizado pelo fato de que compreende:

uma unidade de entrada (210) configurada para receber o dito fluxo de bits (10) representando imagens da dita sequência de vídeo de multicamadas e armazenar o dito fluxo de bits (10) em uma memória conectada (220) compreendendo o dito armazenador temporário de imagens decodificadas (225);

uma unidade de determinação de número (230) conectada à dita memória (220) e configurada para determinar, após a dita imagem atual ter sido decodificada e armazenada no dito armazenador temporário de imagens decodificadas (225), o dito número de imagens no dito armazenador temporário de imagens decodificadas (225) que estão marcadas como que não precisam ser emitidas;

um comparador (240) conectado à dita unidade de determinação de número (230) e configurado para comparar o dito número de imagens no dito armazenador temporário de imagens decodificadas (225) que estão marcadas como que precisam ser emitidas com o dito valor `sps_max_num_reorder_pics[HighestTid]`;

uma unidade de saída (250) conectada ao dito comparador (240) e configurada para emitir a dita imagem, que é uma primeira imagem em ordem de saída, das imagens no dito armazenador temporário de imagens decodificadas (225) que estão marcadas como que precisam ser emitidas se o dito número de imagens no dito armazenador temporário de imagens decodificadas (225) que estão marcadas como que precisam ser emitidas é maior do que o dito valor `sps_max_reorder_pics[HighestTid]`; e

uma unidade de marcação (260) conectada à dita memória (220) e configurada para marcar a dita imagem como que não precisa ser emitida se o dito número de imagens no dito armazenador temporário de imagens decodificadas (225) que estão marcadas como que precisam ser emitidas é maior do que o dito valor `sps_max_reorder_pics[HighestTid]`.

10. Decodificador, de acordo com a reivindicação 8, caracterizado pelo fato de que compreende:

um processador (110); e

uma memória (120) compreendendo o dito armazenador temporário de imagens decodificadas (125), onde o dito processador (110) é configurado para:

determinar, após a dita imagem atual ter sido decodificada e armazenada no dito armazenador temporário de imagens decodificadas (125), o dito número de imagens no dito armazenador temporário de imagens decodificadas (125) que estão marcadas como que precisam ser emitidas;

selecionar o dito valor `sps_max_reorder_pics[HighestTid]`;

comparar o dito número de imagens no dito armazenador temporário de imagens decodificadas (125) que estão marcadas como que precisam ser emitidas com o dito valor `sps_max_num_reorder_pics[HighestTid]`;

emitir a dita imagem, que é uma primeira imagem em ordem de saída, das imagens no dito armazenador temporário de imagens decodificadas (125) que estão marcadas como que precisam ser emitidas se o dito número de imagens no dito armazenador temporário de imagens decodificadas (125) que estão marcadas como que precisam ser emitidas é maior do que o dito valor `sps_max_num_reorder_pics[HighestTid]`;

marcar a dita imagem como que não precisa ser emitida se o dito número

de imagens no dito armazenador temporário de imagens decodificadas (125) que estão marcadas como que precisam ser emitidas é maior do que o dito valor `sps_max_num_reorder_pics[HighestTid]`.

11 Decodificador, de acordo com a reivindicação 10, caracterizado pelo fato de que o dito processador (110) é configurado para:

analisar um cabeçalho de fatia da dita imagem atual a ser

decodificada da dita sequência de vídeo de multicamadas;

determinar um conjunto de imagens de referência para a dita imagem atual com base no dito cabeçalho de fatia analisado;

marcar todas as imagens no dito armazenador temporário de imagens decodificadas (125) que não estão presentes no dito conjunto de imagens de referência como não usadas como referência;

emitir zero, uma ou muitas imagens, marcadas como que precisam ser emitidas, das ditas imagens no dito armazenador temporário de imagens decodificadas (125) e marcar as ditas zero, uma ou muitas imagens como que não precisam ser emitidas;

esvaziar, a partir do dito armazenador temporário de imagens decodificadas (125), qualquer imagem marcada como não usada como referência e que não precisa ser emitida das ditas imagens no dito armazenador temporário de imagens decodificadas (125); e

decodificar a dita imagem atual, onde o dito processador (110) é configurado para determinar o dito número de imagens, comparar o dito número, emitir a dita imagem e marcar a dita imagem após decodificar a dita imagem atual.

12. Decodificador (100, 300), caracterizado pelo fato de que é configurado para:

analisar um cabeçalho de fatia de uma imagem atual a ser decodificada de um fluxo de bits (10) representando imagens de uma sequência de vídeo de multicamadas compreendendo múltiplas camadas de imagens, a dita imagem atual pertencente a uma camada que não a camada mais alta que é decodificada pelo dito decodificador (100, 300) da dita sequência de vídeo de multicamadas;

determinar um conjunto de imagens de referência para a dita imagem atual com base no dito cabeçalho de fatia analisado;

marcar todas as imagens em um armazenador temporário de

imagens decodificadas (125, 325) que não estão presentes no dito conjunto de imagens de referência como não usadas como referência;

emitir zero, uma ou muitas imagens, marcadas como que precisam ser emitidas, das ditas imagens no dito armazenador temporário de imagens decodificadas (125, 325) e marcar as ditas zero, uma ou muitas imagens como que não precisam ser emitidas;

esvaziar, do dito armazenador temporário de imagens decodificadas (125, 325), qualquer imagem marcada como não usada como referência e que não precisa ser emitida das ditas imagens no dito armazenador temporário de imagens decodificadas (125, 325);

decodificar a dita imagem atual;

determinar um número de imagens no dito armazenador temporário de imagens decodificadas (125, 325) que estão marcadas como que precisam ser emitidas;

selecionar um valor `sps_max_num_reorder_pics[HighestTid]`, onde `HighestTid` especifica a dita camada mais alta, e o dito valor `sps_max_num_reorder_pics[HighestTid]` é derivado de um elemento de sintaxe associado com a dita camada mais alta, cada camada `i` das ditas múltiplas camadas tem um elemento de sintaxe associado definindo um respectivo valor `sps_max_num_reorder_pics[i]`, e o dito valor `sps_max_num_reorder_pics[i]` indica um número máximo permitido de imagens que podem preceder qualquer imagem na dita sequência de vídeo de multicamadas na ordem de decodificação e seguem essa imagem em ordem de saída quando `HighestTid` é igual a `i`;

comparar o dito número de imagens no dito armazenador temporário de imagens decodificadas (125, 325) que estão marcadas como que precisam ser emitidas com um valor `sps_max_num_reorder_pics[HighestTid]`;

emitir uma imagem, que é uma primeira imagem em ordem de

saída, das ditas imagens no dito armazenador temporário de imagens decodificadas (125, 325) que estão marcadas como que precisam ser emitidas se o dito número de imagens no dito armazenador temporário de imagens decodificadas (125, 325) que estão marcadas como que precisam ser emitidas é maior do que o dito valor `sps_max_num_reorder_pics[HighestTid]`; e

marcar a dita imagem como que não precisa ser emitida se o dito número de imagens no dito armazenador temporário de imagens decodificadas (125, 325) que estão marcadas como que precisam ser emitidas é maior do que o dito valor `sps_max_num_reorder_pics[HighestTid]`, onde o dito decodificador (100, 300) é configurado para determinar o dito número de imagens, comparar o dito número, emitir a dita imagem e marcar a dita imagem após decodificar a dita imagem atual.

13. Decodificador, de acordo com a reivindicação 12, caracterizado pelo fato de que compreende:

uma unidade de entrada (310) configurada para receber o dito fluxo de bits (10) representando imagens da dita sequência de vídeo de multicamadas e armazenar o dito fluxo de bits (10) em uma memória conectada (320) compreendendo o dito armazenador temporário de imagens decodificadas (325);

uma unidade de análise (370) conectada à dita memória (320) e configurada para analisar o dito cabeçalho de fatia da dita imagem atual a ser decodificada da dita sequência de vídeo de multicamadas;

uma unidade de determinação de conjunto de imagens de referência (380) conectada à dita unidade de análise (370) e configurada para determinar o dito conjunto de imagens de referência para a dita imagem atual com base no dito cabeçalho de fatia analisado;

uma unidade de marcação (360) conectada à dita memória (320) e configurada para marcar todas as ditas imagens no dito armazenador temporário de imagens decodificadas (325) que não estão presentes no dito

conjunto de imagens de referência como não usadas como referência;

uma unidade de saída (350) conectada à dita memória (320) e configurada para emitir zero, uma ou muitas imagens, marcadas como que precisam ser emitidas, das ditas imagens no dito armazenador temporário de imagens decodificadas (325), onde a dita unidade de marcação (360) é configurada para marcar zero, uma ou muitas imagens como que não precisam ser emitidas;

uma unidade de esvaziamento de imagem (390) conectada à dita memória (320) e configurada para esvaziar, do dito armazenador temporário de imagens decodificadas (325), qualquer dita imagem marcada como não usada como referência e que não precisa ser emitida das imagens no dito armazenador temporário de imagens decodificadas (325);

uma unidade de decodificação (305) conectada à dita memória (320) e configurada para decodificar a dita imagem atual;

uma unidade de determinação de número (330) conectada à dita memória (320) e configurada para determinar o dito número de imagens no dito armazenador temporário de imagens decodificadas (325) que estão marcadas como que não precisam ser emitidas; e

um comparador (340) conectado à dita unidade de determinação de número (330) e configurado para comparar o dito número de imagens no dito armazenador temporário de imagens decodificadas (325) que estão marcadas como que precisam ser emitidas com o valor `sps_max_num_reorder_pics[HighestTid]`, onde

a dita unidade de saída (350) é também configurada para emitir a dita imagem, que é uma primeira imagem em ordem de saída, das ditas imagens no dito armazenador temporário de imagens decodificadas (325) que estão marcadas como que precisam ser emitidas se o dito número de imagens no dito armazenador temporário de imagens decodificadas (325) que estão marcadas como que precisam ser emitidas é maior do que o dito

valor `sps_max_num_reorder_pics[HighestTid]`; e a dita unidade de marcação (360) é também configurada para marcar a dita imagem como que não precisa ser emitida se o dito número de imagens no dito armazenador temporário de imagens decodificadas (325) que estão marcadas como que precisam ser emitidas é maior do que o dito valor `sps_max_num_reorder_pics[HighestTid]`, onde a dita unidade de determinação de número (330) é configurada para determinar o dito número de imagens, o dito comparador (340) é configurado para comparar o dito número, a dita unidade de saída (350) é configurada para emitir a dita imagem e a dita unidade de marcação (360) é configurada para marcar a dita imagem após a dita unidade de decodificação (305) ter decodificado a dita imagem atual.

14. Decodificador, de acordo com a reivindicação 12, caracterizado pelo fato de que compreende:

um processador (110); e

uma memória (120) compreendendo o dito armazenador temporário de imagens decodificadas (125), onde o dito processador (110) é configurado para: analisar o dito cabeçalho de fatia da dita imagem atual;

determinar o dito conjunto de imagens de referência para a dita imagem atual com base no dito cabeçalho de fatia analisado;

marcar todas as ditas imagens no dito armazenador temporário de imagens decodificadas (125) que não estão presentes no dito conjunto de imagens de referência como não usadas como referência;

emitir zero, uma ou muitas imagens, marcadas como que precisam ser emitidas, das ditas imagens no dito armazenador temporário de imagens decodificadas (125) e marcar zero, uma ou muitas imagens como que não precisam ser emitidas;

esvaziar, do dito armazenador temporário de imagens decodificadas (125), qualquer dita imagem marcada como não usada como referência e que não precisa ser emitida das ditas imagens no dito

armazenador temporário de imagens decodificadas (125);

decodificar a dita imagem atual;

determinar o dito número de imagens no dito armazenador temporário de imagens decodificadas (125) que estão marcadas como que precisam ser emitidas; selecionar o dito valor `sps_max_num_reorder_pics[HighestTid]`;

comparar o dito número de imagens no dito armazenador temporário de imagens decodificadas (125) que estão marcadas como que precisam ser emitidas é maior do que o dito valor `sps_max_num_reorder_pics[HighestTid]`;

emitir a dita imagem, que é uma primeira imagem em ordem de saída, das ditas imagens no dito armazenador temporário de imagens decodificadas (125) que estão marcadas como que precisam ser emitidas se o dito número de imagens no dito armazenador temporário de imagens decodificadas (325) que estão marcadas como que precisam ser emitidas é maior do que o dito valor `sps_max_num_reorder_pics[HighestTid]`; e

marcar a dita imagem como que não precisa ser emitida se o dito número de imagens no dito armazenador temporário de imagens decodificadas (325) que estão marcadas como que precisam ser emitidas é maior do que o dito valor `sps_max_num_reorder_pics[HighestTid]`, onde o dito processador (110) é configurado para determinar o dito número de imagens, comparar o dito número, emitir a dita imagem e marcar a dita imagem após decodificar a dita imagem atual.

15. Decodificador, de acordo com a reivindicação 11 ou 14, caracterizado pelo fato de que o dito processador (110) é configurado para:

armazenar a dita imagem atual decodificada no dito armazenador temporário de imagens decodificadas (125); e

marcar a dita imagem atual decodificada como que precisa ser emitida ou como que não precisa ser emitida.

16. Decodificador, de acordo com qualquer uma das reivindicações 10, 11, 14 ou 15, caracterizado pelo fato de que o dito processador (110) é configurado para:

emitir uma imagem no dito armazenador temporário de imagens decodificadas (125) com um menor valor de PicOrderCntVal de todas as imagens no dito armazenador temporário de imagens decodificadas (125) que estão marcadas como que precisam ser emitidas, se o dito número de imagens no dito armazenador temporário de imagens decodificadas (125) que estão marcadas como que precisam ser emitidas é maior do que o dito valor `sps_max_num_reorder_pics[HighestTid]`, onde PicOrderCntVal representa um valor de contagem de ordem de imagens.

17. Decodificador, de acordo com qualquer uma das reivindicações 10, 11, 14, 15 ou 16, caracterizado pelo fato de que o dito processador (110) é configurado para emitir uma imagem com o menor valor de contagem de ordem de imagens das ditas imagens no dito armazenador temporário de imagens decodificadas (125) que estão marcadas como que precisam ser emitidas se o dito número de imagens no dito armazenador temporário de imagens decodificadas (125) que estão marcadas como que precisam ser emitidas é maior do que o dito valor `sps_max_num_reorder_pics[HighestTid]`.

18. Decodificador, de acordo com qualquer uma das reivindicações 10, 11, 14, 15, 16, ou 17, caracterizado pelo fato de que o dito processador (110) é configurado para emitir a dita imagem antes de analisar um cabeçalho de fatia de uma próxima imagem a ser decodificada da dita sequência de vídeo de multicamadas.

19. Decodificador, de acordo com qualquer uma das reivindicações 8 a 18, caracterizado pelo fato de que o dito decodificador (100, 200, 300, 400, 500) é um decodificador (100, 200, 300, 400, 500) compatível com Codificação de Vídeo de Alta Eficiência, HEVC.

20. Método executado por um codificador (600, 700, 900), caracterizado pelo fato de que compreende:

determinar (S60), após uma imagem atual ter sido decodificada e armazenada em um armazenador temporário de imagens decodificadas (625, 725), um número de imagens no dito armazenador temporário de imagens decodificadas (625, 725) que estão marcadas como que precisam ser emitidas, a dita imagem atual pertencente a uma camada que não a camada mais alta que é decodificada pelo dito codificador (600, 700, 900) de uma sequência de vídeo de multicamadas compreendendo múltiplas camadas de imagens;

selecionar um valor `sps_max_num_reorder_pics[HighestTid]`, onde `HighestTid` especifica a dita camada mais alta, cada camada `i` das ditas múltiplas camadas tem um valor associado `sps_max_num_reorder_pics[i]`, e o dito valor `sps_max_num_reorder_pics[i]` indica um número máximo permitido de imagens que podem preceder qualquer imagem na dita sequência de vídeo de multicamadas na

ordem de decodificação e seguem essa imagem em ordem de saída quando `HighestTid` é igual a `i`;

comparar (S61) as ditas imagens no dito armazenador temporário de imagens decodificadas (625, 725) que estão marcadas como que precisam ser emitidas com o dito valor `sps_max_num_reorder_pics[HighestTid]`; e

marcar (S62) uma imagem, que é uma primeira imagem em ordem de saída, das ditas imagens no dito armazenador temporário de imagens decodificadas (625, 725) que estão marcadas como que precisam ser emitidas, como que não precisa ser emitida se o dito número de imagens no dito armazenador temporário de imagens decodificadas (625, 725) que estão marcadas como que precisam ser emitidas é maior do que o dito valor `sps_max_num_reorder_pics[HighestTid]`.

21. Método executado por um codificador (600, 800), caracterizado pelo fato de que compreende:

marcar (S70) todas as imagens em um dito armazenador temporário de imagens decodificadas (625, 825) que não estão presentes em um conjunto de imagens de referência para uma imagem atual de uma sequência de vídeo de multicamadas compreendendo múltiplas camadas de imagens como não usada como referência, a dita imagem atual pertencente a uma camada que não a camada mais alta que é decodificada pelo dito codificador (600, 800) da dita sequência de vídeo de multicamadas;

marcar (S71) zero, uma ou muitas imagens, marcadas como que precisam ser emitidas, das ditas imagens no dito armazenador temporário de imagens decodificadas (625, 825) como que não precisam ser emitidas;

esvaziar (S72), a partir do dito armazenador temporário de imagens decodificadas (625, 825), qualquer imagem marcada como não usada como referência e não precisa ser emitida das ditas imagens no dito armazenador temporário de imagens decodificadas (625, 825);

decodificar (S73) a dita imagem atual;

determinar (S74) um número de imagens no dito armazenador temporário de imagens decodificadas (625, 825) que estão marcadas como que precisam ser emitidas;

selecionar (S50) um valor `sps_max_num_reorder_pics[HighestTid]`, onde `HighestTid` especifica a dita camada mais alta, cada camada `i` das ditas múltiplas camadas tem um valor associado `sps_max_num_reorder_pics[i]`, e o dito valor `sps_max_num_reorder_pics[i]` indica um número máximo permitido de imagens que podem preceder qualquer imagem na dita sequência de vídeo de multicamadas na ordem de decodificação e seguem essa imagem em ordem de saída quando `HighestTid` é igual a `i`;

comparar (S75) o dito número de imagens no dito

armazenador temporário de imagens decodificadas (625, 825) que estão marcadas como que precisam ser emitidas com o dito valor `sps_max_num_reorder_pics[HighestTid]`; e

marcar (S76) uma imagem, que é uma primeira imagem em ordem de saída, das ditas imagens no dito armazenador temporário de imagens decodificadas (625, 825) que estão marcadas como que precisam ser emitidas, como que não precisa ser emitida, se o dito número de imagens no dito armazenador temporário de imagens decodificadas (625, 825) que estão marcadas como que precisam ser emitidas é maior do que o dito valor `sps_max_num_reorder_pics[HighestTid]`, onde determinar (S74) o dito número de imagens, comparar (S75) o dito número, e marcar (S76) a dita imagem são executados após decodificar (S73) a dita imagem atual.

22. Codificador (600, 700), caracterizado pelo fato de ser configurado para:

determinar, após uma imagem atual ter sido decodificada e armazenada em um armazenador temporário de imagem decodificada (625, 725), um número de imagens no dito armazenador temporário de imagem decodificada (625, 725) que estão marcadas como que precisam ser emitidas, a dita imagem atual pertencente a uma camada que não a camada mais alta que é decodificada pelo dito codificador (600, 700) de uma sequência de vídeo de multicamadas compreendendo múltiplas camadas de imagens;

selecionar um valor `sps_max_num_reorder_pics[HighestTid]`, onde `HighestTid` especifica a dita camada mais alta, cada camada `i` das ditas múltiplas camadas tem um valor associado `sps_max_num_reorder_pics[i]`, e o dito valor `sps_max_num_reorder_pics[i]` indica um número máximo permitido de imagens que podem preceder qualquer imagem na dita sequência de vídeo de multicamadas na ordem de decodificação e seguem essa imagem em ordem de saída quando `HighestTid` é igual a `i`;

comparar o dito número de imagens no dito armazenador

temporário de imagens decodificadas (625, 725) que estão marcadas como que precisam ser emitidas com o dito valor `sps_max_num_reorder_pics[HighestTid]`; e

marcar uma imagem, que é uma primeira imagem em ordem de saída, das ditas imagens no dito armazenador temporário de imagens decodificadas (625, 725) que são marcadas como que precisam ser emitidas, como que não precisa ser emitida se o dito número dito número de imagens no dito armazenador temporário de imagens decodificadas (625, 725) que estão marcadas como que precisam ser emitidas é maior do que o dito valor `sps_max_num_reorder_pics[HighestTid]`.

23. Codificador, de acordo com a reivindicação 22, caracterizado pelo fato de que compreende:

um processador (610); e

uma memória (620) compreendendo o dito armazenador temporário de imagens decodificadas (625), onde o dito processador (610) é configurado para:

determinar, após a dita imagem atual ter sido decodificada e armazenada no dito armazenador temporário de imagens decodificadas (625), o dito número de imagens no dito armazenador temporário de imagens decodificadas (625) que estão marcadas como que precisam ser emitidas;

selecionar o dito valor `sps_max_num_reorder_pics[HighestTid]`;

comparar o dito número de imagens no dito armazenador temporário de imagens decodificadas (625) que estão marcadas como que precisam ser emitidas com o dito valor `sps_max_num_reorder_pics[HighestTid]`; e

marcar a dita imagem, que é uma primeira imagem em ordem de saída, das ditas imagens no dito armazenador temporário de imagens decodificadas (625) que estão marcadas como que precisam ser emitidas,

como que não precisa ser emitida se o dito número de imagens no dito armazenador temporário de imagens decodificadas (625) que estão marcadas como que precisam ser emitidas é maior do que o dito valor `sps_max_num_reorder_pics[HighestTid]`.

24. Codificador, de acordo com a reivindicação 22, caracterizado pelo fato de que compreende:

uma unidade de determinação de número (730) conectada à dita memória (720) compreendendo o dito armazenador temporário de imagens decodificadas (725) e configurada para determinar, após a dita imagem atual da dita sequência de vídeo de multicamadas ter sido decodificada e armazenada no dito armazenador temporário de imagens decodificadas (725), o dito número de imagens no dito armazenador temporário de imagens decodificadas (725) que estão marcadas como que não precisam ser emitidas;

um comparador (540) conectado à dita unidade de determinação de número (730) e configurado para comparar o dito número de imagens no dito armazenador temporário de imagens decodificadas (725) que estão marcadas como que precisam ser emitidas é maior do que o dito valor `sps_max_num_reorder_pics[HighestTid]`; e

uma unidade de marcação (560) conectada à dita memória (720) e configurada para marcar a dita imagem, que é a primeira imagem em ordem de saída, das ditas imagens no dito armazenador temporário de imagens decodificadas (725) que estão marcadas como que precisam ser emitidas, como que não precisa ser emitida se o dito número de imagens no dito armazenador temporário de imagens decodificadas (725) que estão marcadas como que precisam ser emitidas é maior do que o dito valor `sps_max_num_reorder_pics[HighestTid]`.

25. Codificador (600, 800), caracterizado pelo fato de que é configurado para:

marcar todas as imagens em um armazenador temporário de imagens decodificadas (625, 825) que não estão presentes em um conjunto de imagens de referência para uma imagem atual de uma sequência de vídeo de multicamadas compreendendo múltiplas camadas de imagens como não usada como referência, a

dita imagem atual pertencente a uma camada que não a camada mais alta que é decodificada pelo dito codificador (600, 800) da dita sequência de vídeo de multicamadas;

marcar zero, uma ou muitas imagens, marcadas como que precisam ser emitidas, das ditas imagens no dito armazenador temporário de imagens decodificadas (625, 825) como que não precisam ser emitidas;

esvaziar, do dito armazenador temporário de imagens decodificadas (625, 825), qualquer imagem marcada como não usada como referência e que não precisa ser emitida das ditas imagens no dito armazenador temporário de imagens decodificadas (625, 825);

decodificar a dita imagem atual;

determinar um número de imagens no dito armazenador temporário de imagens decodificadas (625, 825) que são marcadas como que precisam ser emitidas;

selecionar um valor `sps_max_num_reorder_pics[HighestTid]`, onde `HighestTid` especifica a dita camada mais alta, cada camada `i` das ditas múltiplas camadas tem um valor associado `sps_max_num_reorder_pics[i]`, e o dito valor `sps_max_num_reorder_pics[i]` indica um número máximo permitido de imagens que podem preceder qualquer imagem na dita sequência de vídeo de multicamadas na ordem de decodificação e seguem essa imagem em ordem de saída quando `HighestTid` é igual a `i`;

comparar o dito número de imagens no dito armazenador temporário de imagens decodificadas (625, 825) que estão marcadas como que precisam ser emitidas com o dito valor `sps_max_num_reorder_pics[`

HighestTid]; e

marcar uma imagem, que é uma primeira imagem em ordem de saída, das ditas imagens no dito armazenador temporário de imagens decodificadas (625, 825) que estão marcadas como que precisam ser emitidas, como que precisa ser emitida se o dito número de imagens no dito armazenador temporário de imagens decodificadas (625, 825) que estão marcadas como que precisam ser emitidas é maior do que o dito valor `sps_max_num_reorder_pics[HighestTid]`, onde o dito codificador (600, 800) é configurado para determinar o dito número de imagens, comparar o dito número e marcar a dita imagem são executados após decodificar a dita imagem atual.

26. Codificador, de acordo com a reivindicação 25, caracterizado pelo fato de que compreende:

um processador (610); e

uma memória (620) compreendendo um armazenador temporário de imagens decodificadas (625), onde o dito processador (610) é configurado para:

marcar todas as ditas imagens no dito armazenador temporário de imagens decodificadas (625) que não estão presentes no dito conjunto de imagens de referência para a dita imagem atual da dita sequência de vídeo de multicamadas como não usadas como referência;

marcar zero, uma ou muitas imagens, marcadas como que precisam ser emitidas, das ditas imagens no dito armazenador temporário de imagens decodificadas (625) como que não precisam ser emitidas;

esvaziar, do dito armazenador temporário de imagens decodificadas (625), qualquer dita imagem marcada como não usada como referência e que não precisa ser emitida das ditas imagens no dito armazenador temporário de imagens decodificadas (625);

decodificar a dita imagem atual;

determinar o dito número de imagens no dito armazenador temporário de imagens decodificadas (625) que estão marcadas como que precisam ser emitidas;

selecionar o dito valor `sps_max_num_reorder_pics[HighestTid]`;

comparar o dito número de imagens no dito armazenador temporário de imagens decodificadas (625) que estão marcadas como que precisam ser emitidas com o dito valor `sps_max_num_reorder_pics[HighestTid]`; e

marcar a dita imagem, que é uma primeira imagem em ordem de saída, das ditas imagens no dito armazenador temporário de imagens decodificadas (625) que estão marcadas como que precisam ser emitidas, como que não precisa ser emitida se o dito número de imagens no dito armazenador temporário de imagens decodificadas (625) que estão marcadas como que precisam ser emitidas com o dito valor `sps_max_num_reorder_pics[HighestTid]`, onde o processador (610) é configurado para determinar o dito número de imagens, comparar o dito número e marcar a dita imagem após decodificar a dita imagem atual.

27. Codificador, de acordo com a reivindicação 25, caracterizado pelo fato de que compreende:

uma unidade de marcação (860) conectada a uma memória (820) compreendendo o dito armazenador temporário de imagens decodificadas (825) e configurada para: (i) marcar todas as ditas imagens no dito armazenador temporário de imagens decodificadas (825) que não estão presentes no dito conjunto de imagens de referência para a dita imagem atual da dita sequência de vídeo de multicamadas como não usadas como referência e (ii) marcar zero, uma ou muitas imagens, marcadas como que precisam ser emitidas, das ditas imagens no dito armazenador temporário de imagens decodificadas (825) como que não precisam ser emitidas;

uma unidade de esvaziamento de imagem (890) conectada à dita memória (820) e configurada para esvaziar, do dito armazenador temporário de imagens decodificadas (825), qualquer dita imagem marcada como não usada como referência e que não precisa ser emitida das ditas imagens no dito armazenador temporário de imagens decodificadas (825);

uma unidade de decodificação (805) conectada à dita memória (820) e configurada para decodificar a dita imagem atual;

uma unidade de determinação de número (830) conectada à dita memória (820) e configurada para determinar o dito número de imagens no dito armazenador temporário de imagens decodificadas (825) que estão marcadas como que precisam ser emitidas; e

um comparador (840) conectado à dita unidade de determinação de número (840) e configurado para comparar o dito número de imagens no dito armazenador temporário de imagens decodificadas (825) que estão marcadas como que precisam ser emitidas com o dito valor `sps_max_num_reorder_pics[HighestTid]`, onde a dita unidade de marcação (860) é também configurada para marcar a dita imagem, que é uma primeira imagem em ordem de saída, das ditas imagens no dito armazenador temporário de imagens decodificadas (825) que estão marcadas como que precisam ser emitidas, como que não precisa ser emitida se o dito número de imagens no dito armazenador temporário de imagens decodificadas (825) que estão marcadas como que precisam ser emitidas com o dito valor `sps_max_num_reorder_pics[HighestTid]`, onde a dita unidade de determinação de número (830) é configurada para determinar o dito número de imagens, o dito comparador (840) é configurado para comparar o dito número e a dita unidade de marcação (840) é configurada para marcar a dita imagem após a dita unidade de decodificação (805) ter decodificado a dita imagem atual.

28. Codificador, de acordo com qualquer uma das

reivindicações 22 a 27, caracterizado pelo fato de que o dito codificador (600, 700, 800, 900, 1000) é um codificador (600, 700, 800, 900, 1000) compatível com Codificação de Vídeo de Alta Eficiência, HEVC.

29. Terminal móvel (1100), caracterizado pelo fato de que compreende um decodificador (100, 200, 300, 400, 500) do tipo definido em qualquer uma das reivindicações 8 a 19 e/ou um codificador (600, 700, 800, 900, 1000) do tipo definido em qualquer uma das reivindicações 22 a 28.

30. Nó de rede (2), caracterizado pelo fato de que compreende um decodificador (100, 200, 300, 400, 500) do tipo definido em qualquer uma das reivindicações 8 a 19 e/ou um codificador (600, 700, 800, 900, 1000) do tipo definido em qualquer uma das reivindicações 22 a 28.

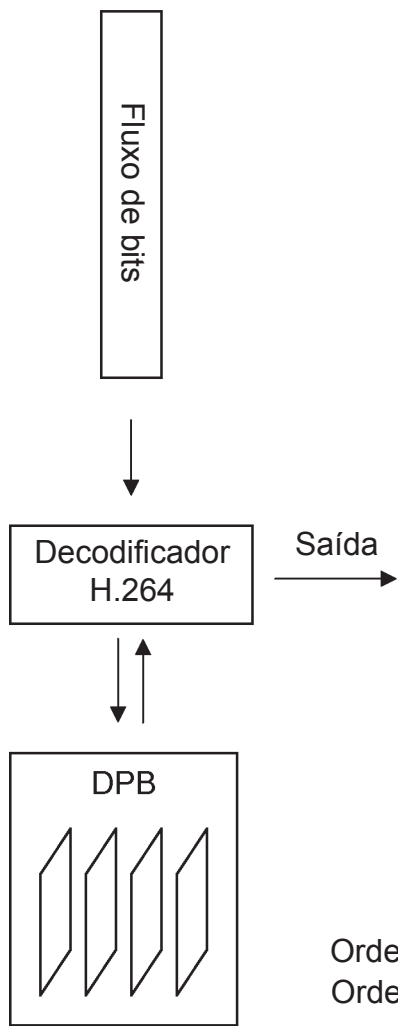
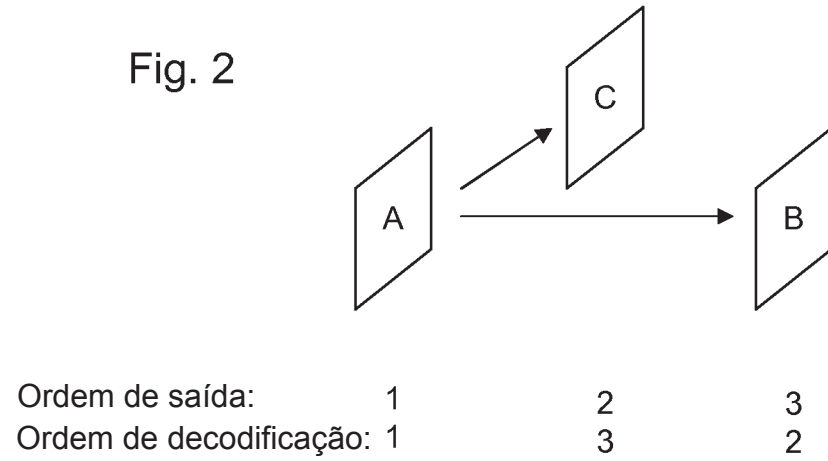


Fig. 1

Fig. 2



Ordem de saída: 1 2? 3 4?

Ordem de decodificação: 1 3 2

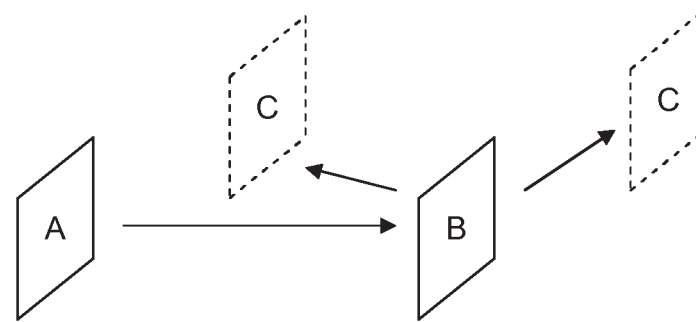


Fig. 4

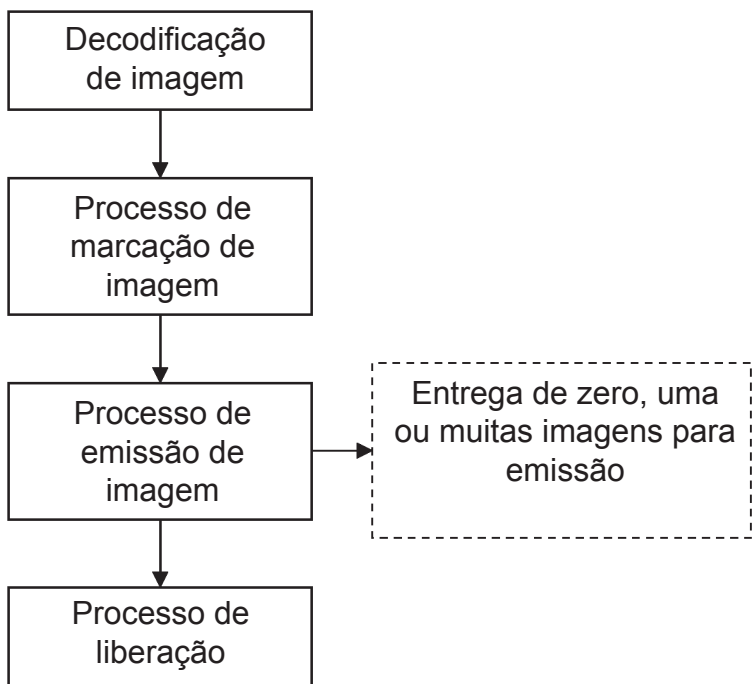


Fig. 3

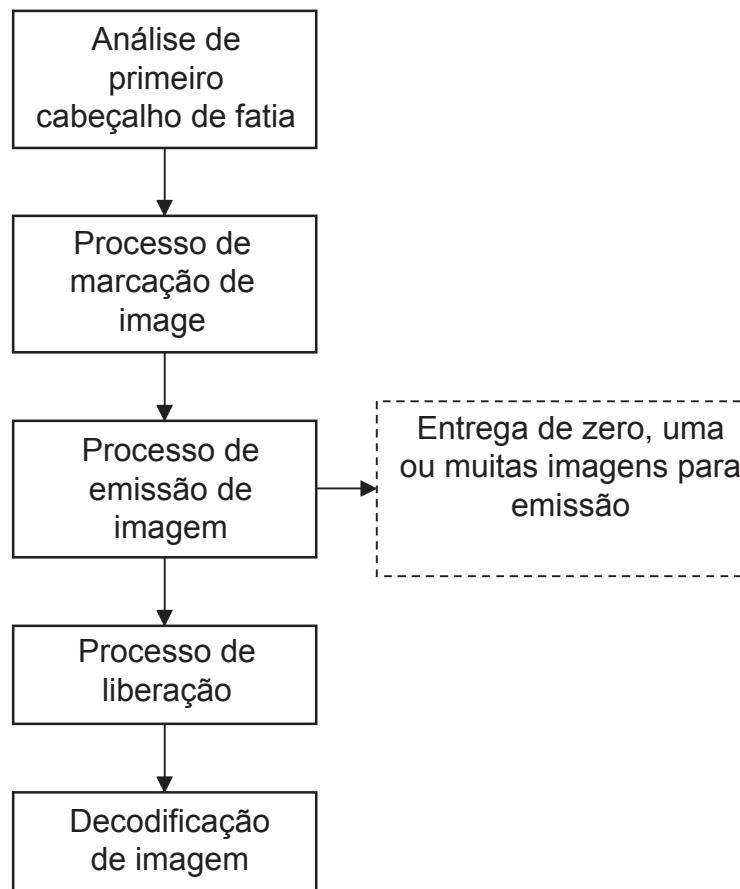


Fig. 5

MAX_NUM_REORDER_PICS[2]=2

MAX_NUM_REORDER_PICS[1]=1

MAX_NUM_REORDER_PICS[0]=0

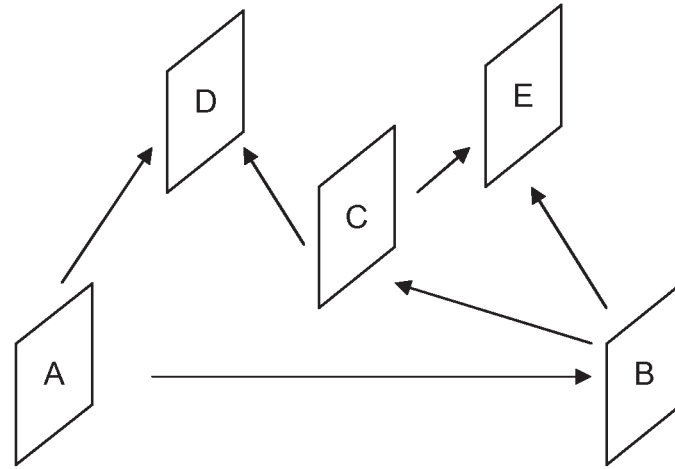


Fig. 6

Camada temporal:	0	2	1	2	0
Ordem de saída:	1	2	3	4	5
Ordem de decodificação:	1	4	3	5	2

MAX_NUM_REORDER_PICS[0]=0

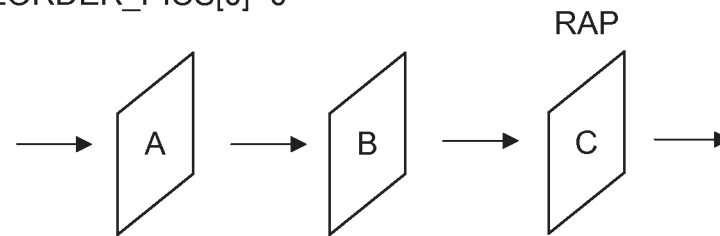
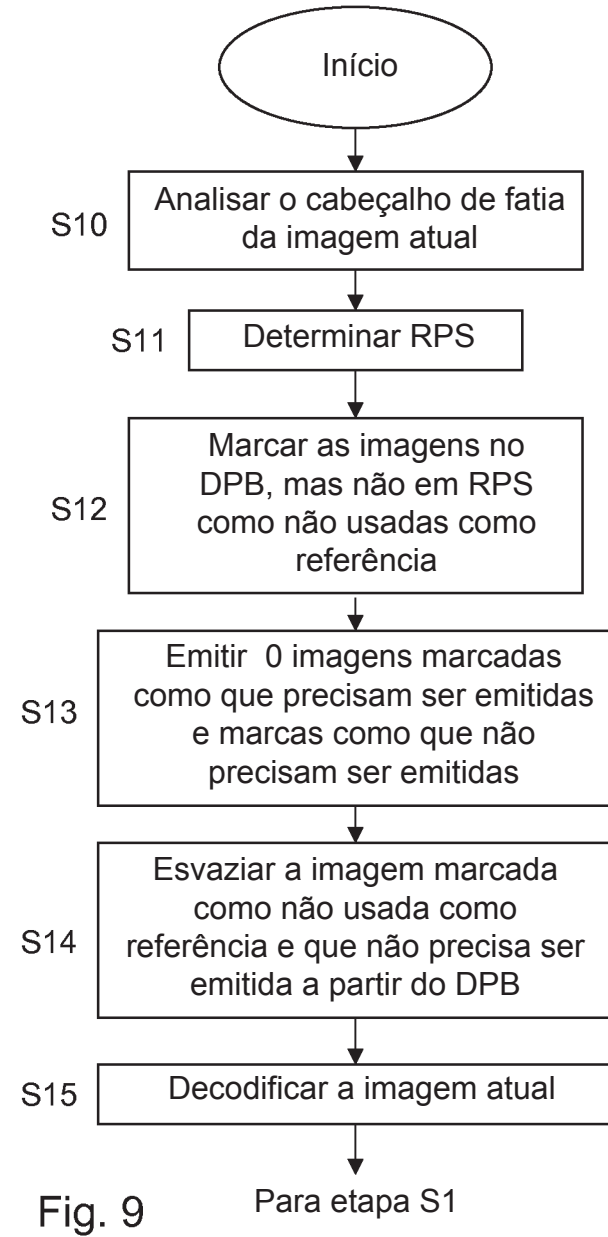
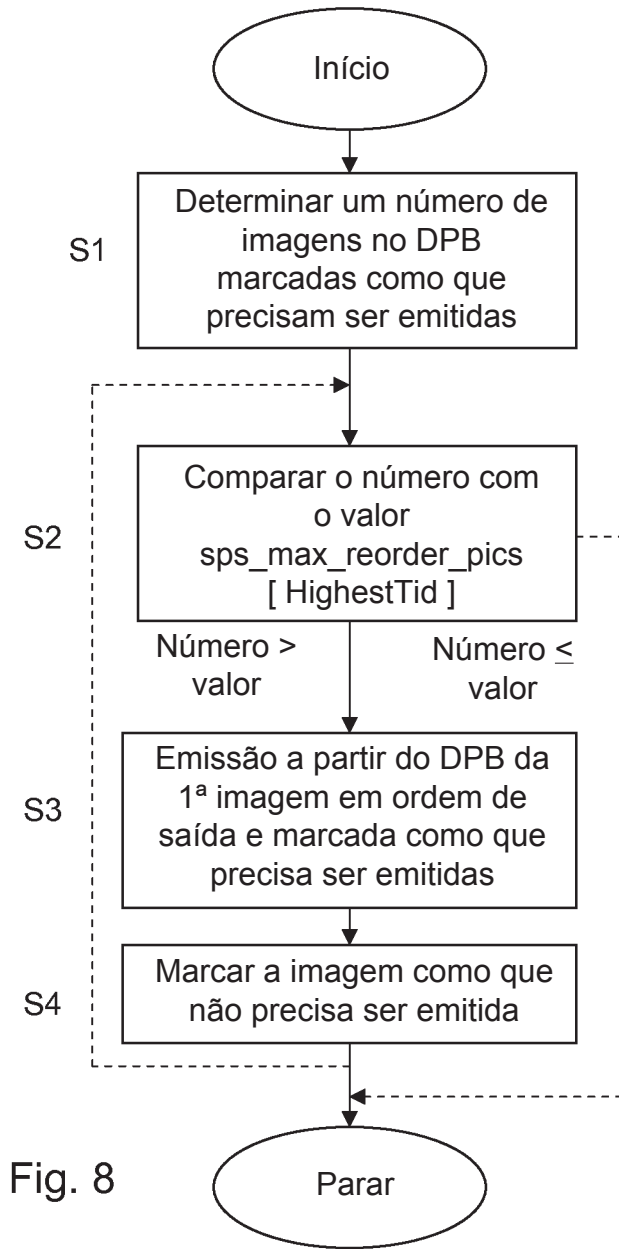


Fig. 7

Ordem de saída:	1	2	3
Ordem de decodificação:	1	2	3



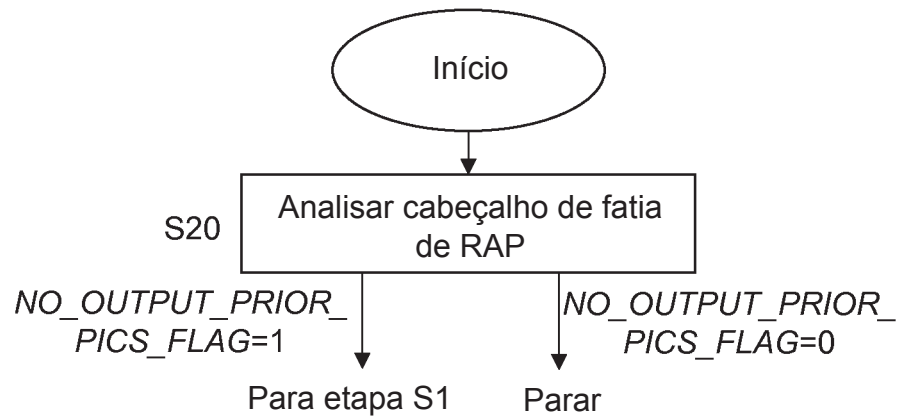


Fig. 10

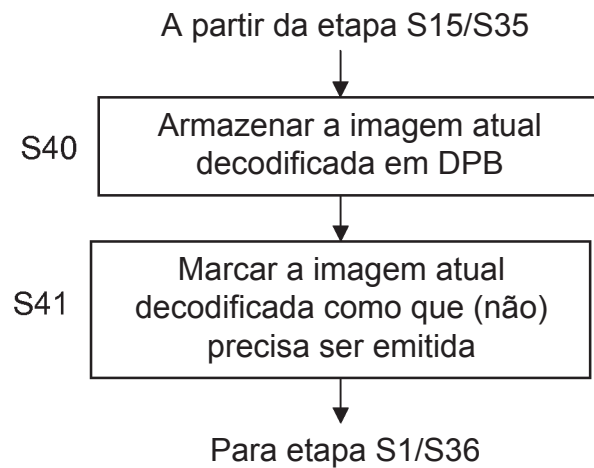


Fig. 12

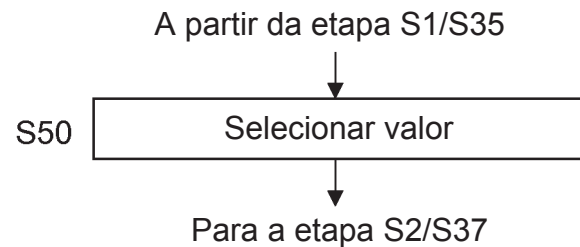


Fig. 13

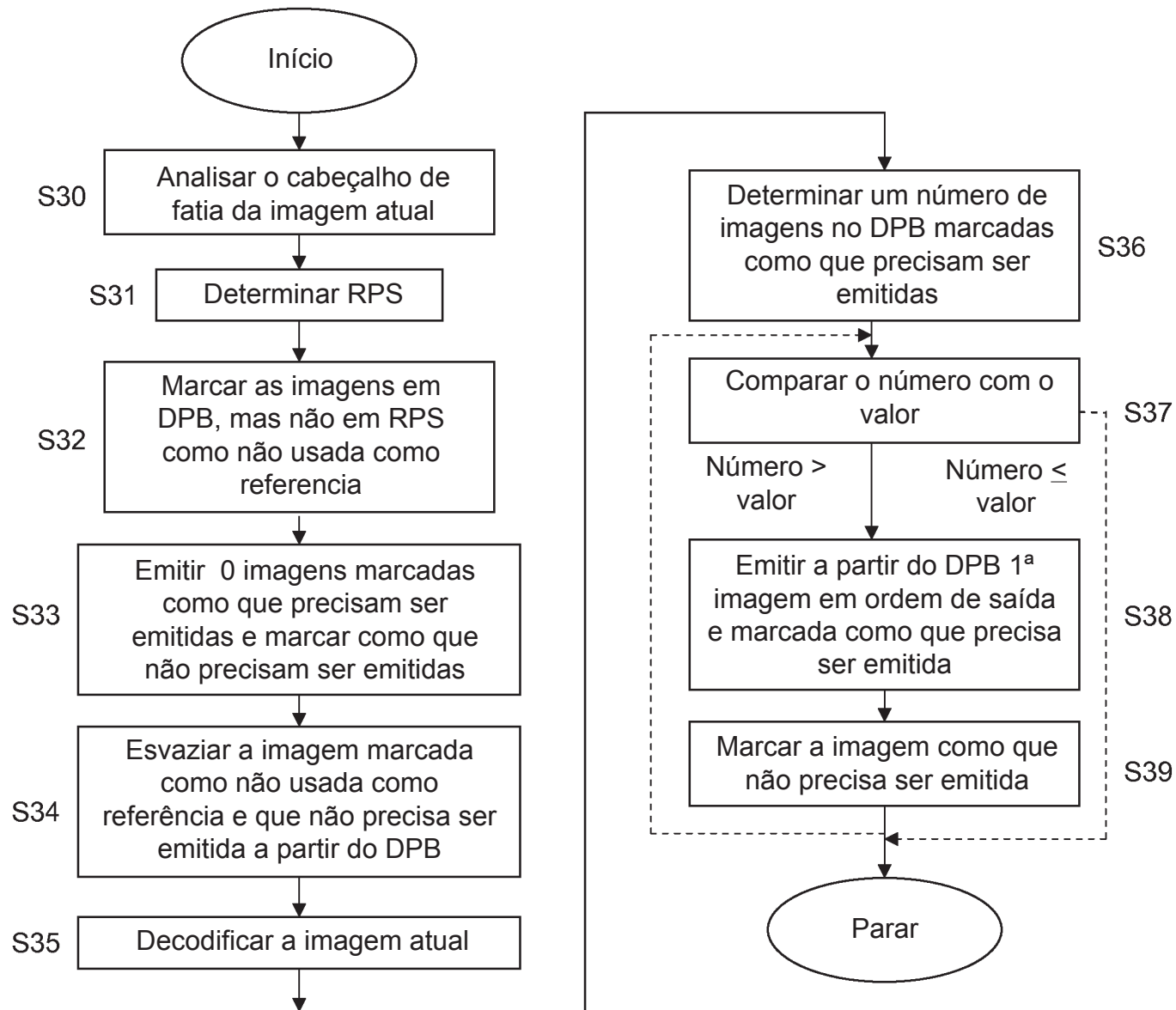


Fig. 11

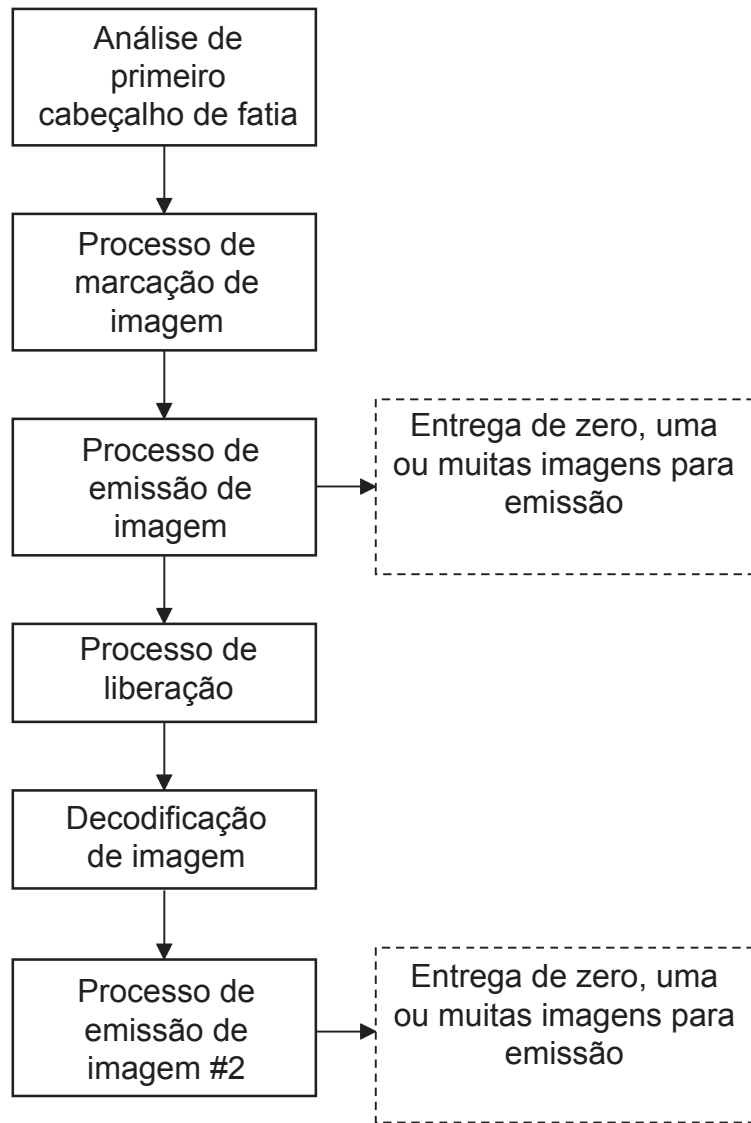


Fig. 14

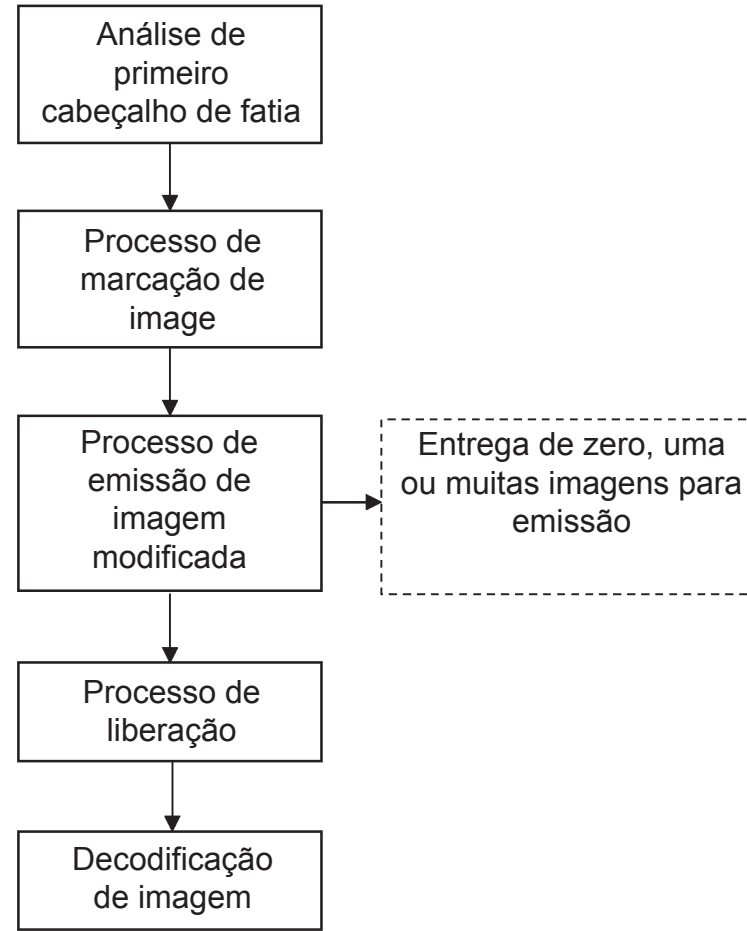


Fig. 15

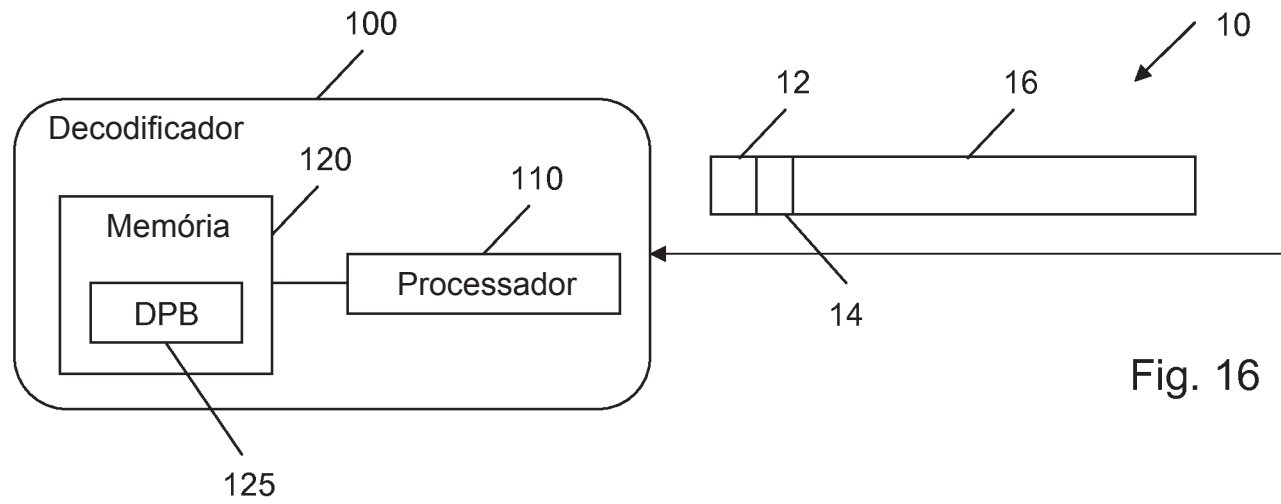


Fig. 16

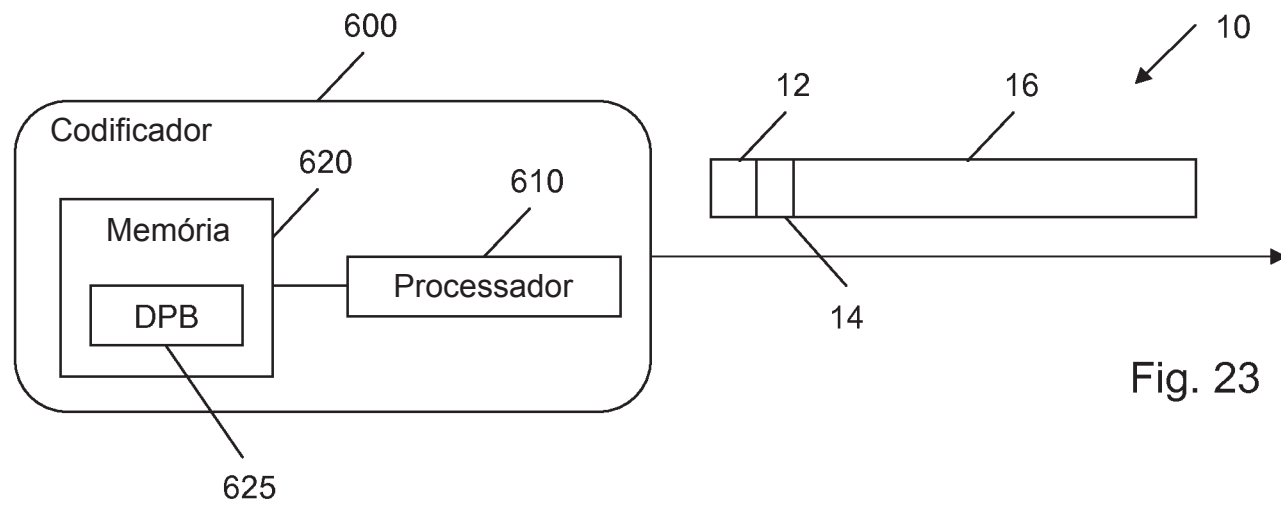


Fig. 23

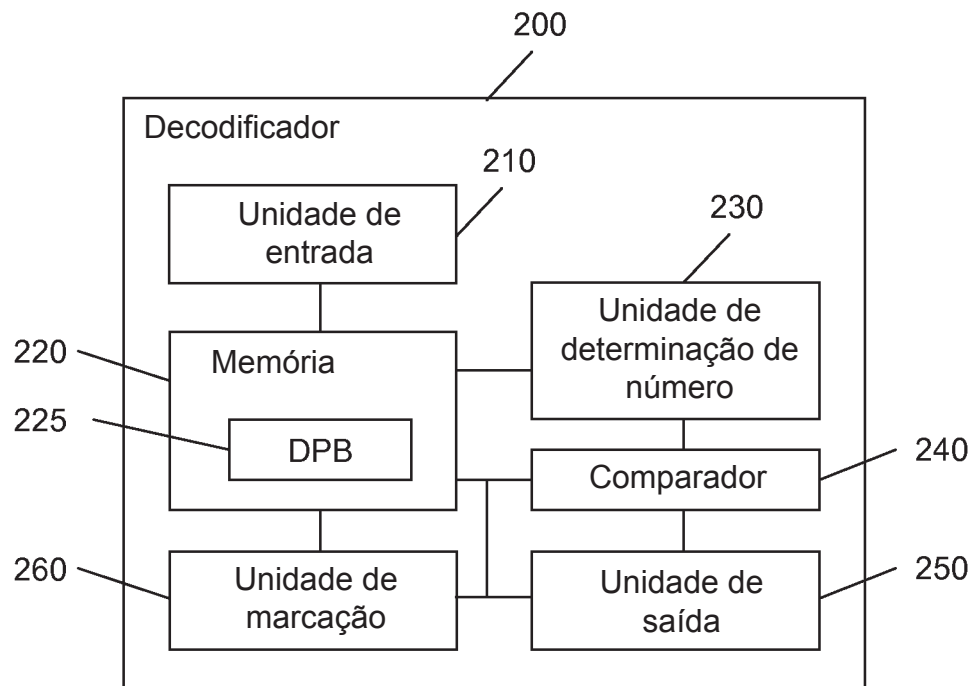


Fig. 17

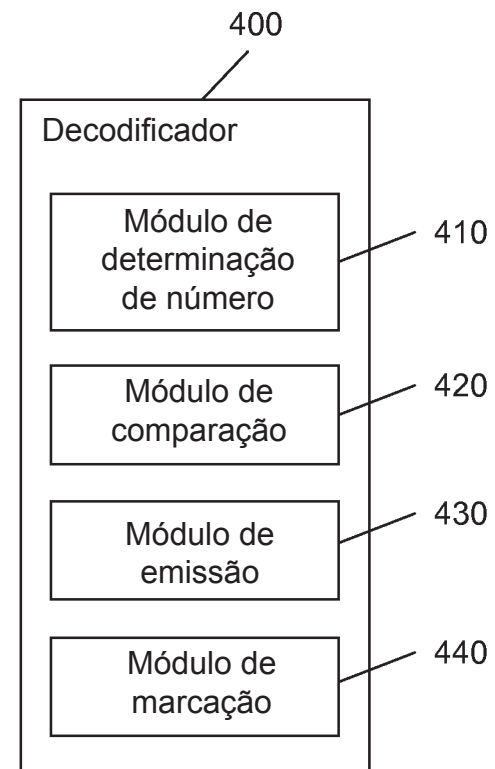


Fig. 19

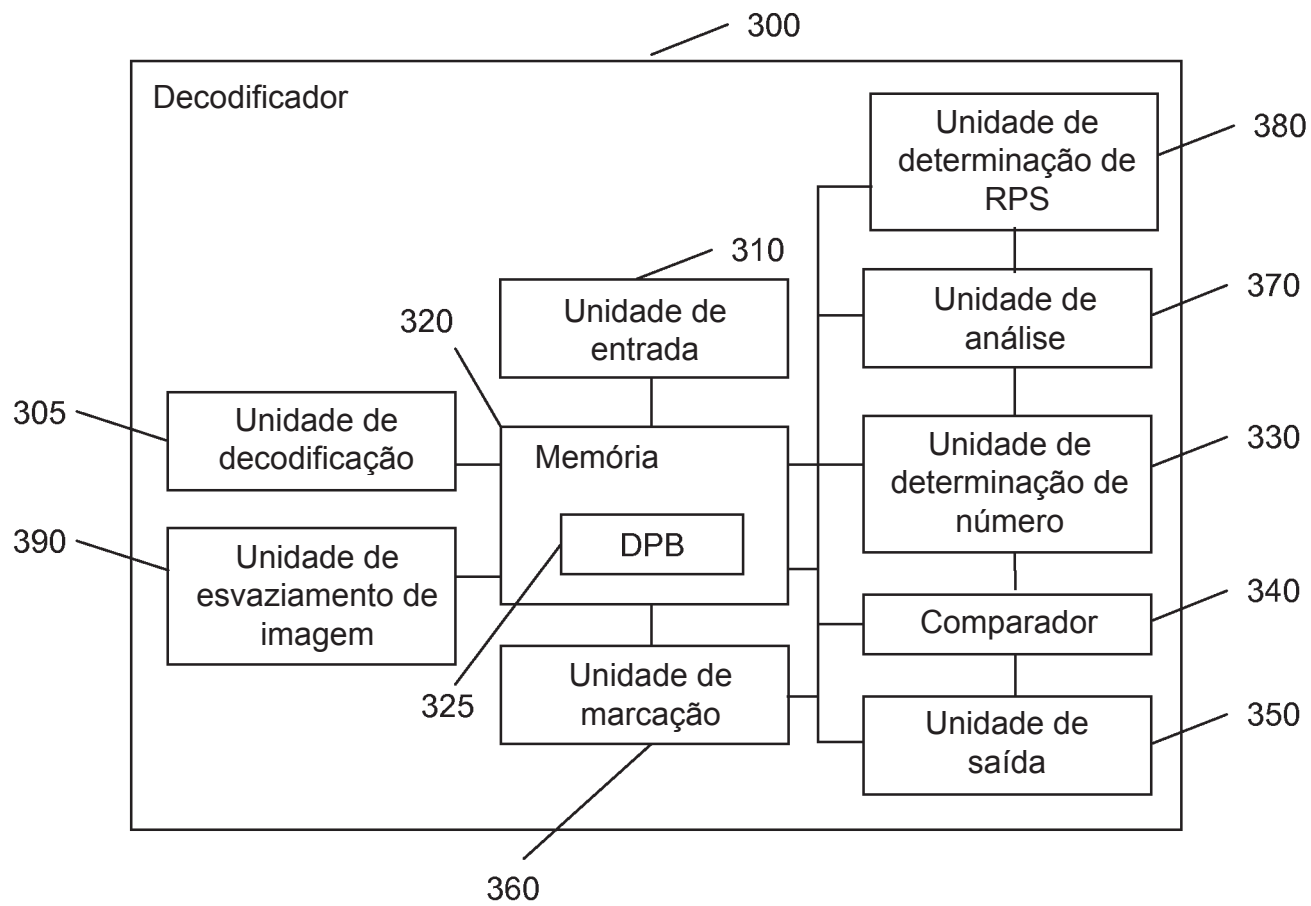


Fig. 18

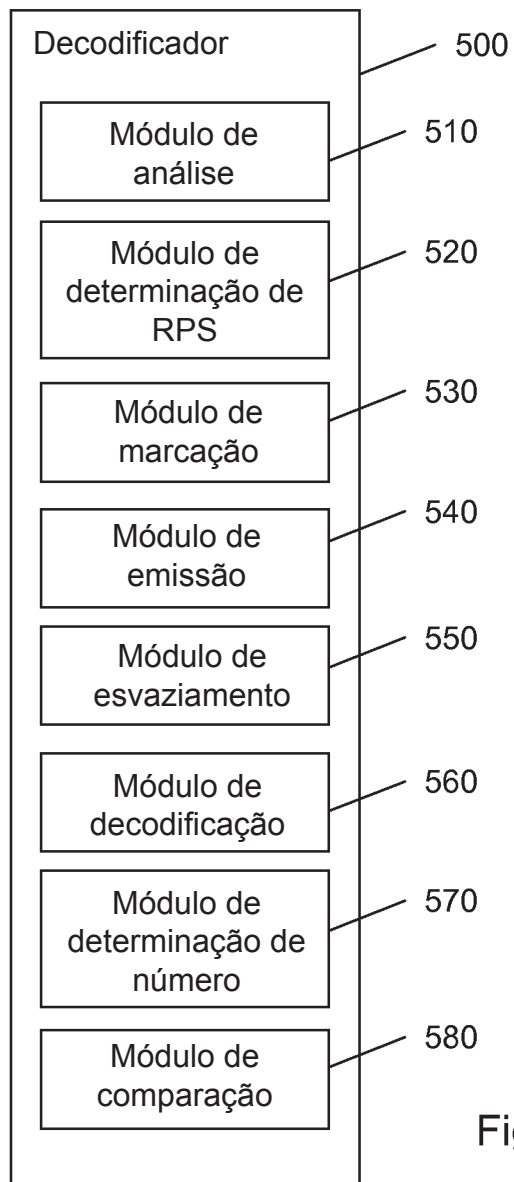


Fig. 20

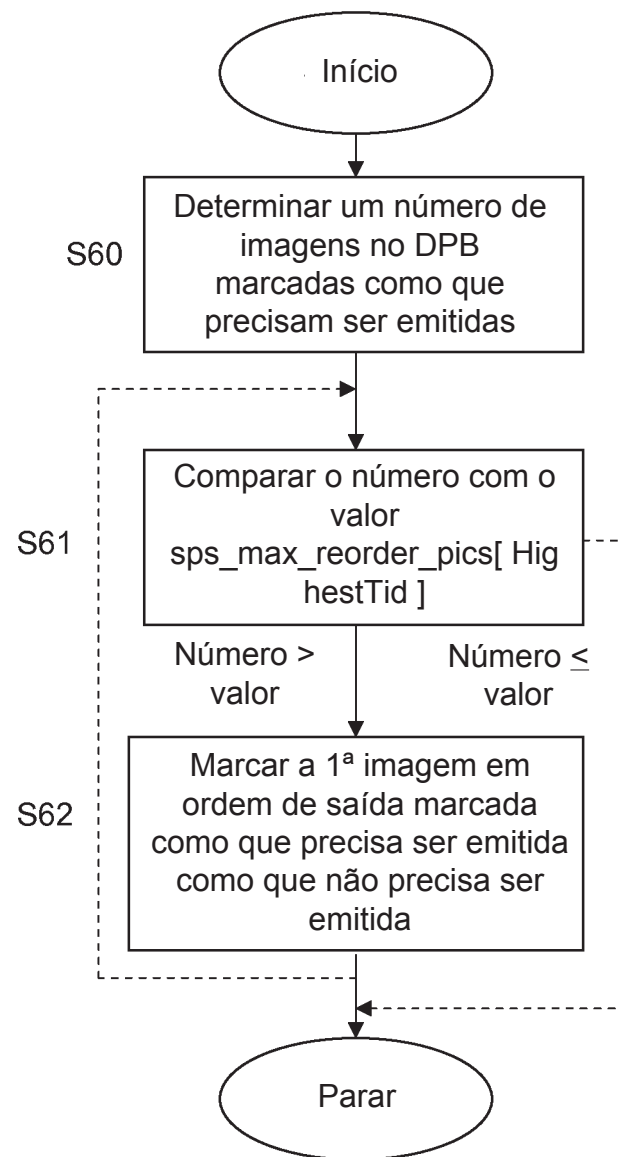


Fig. 21

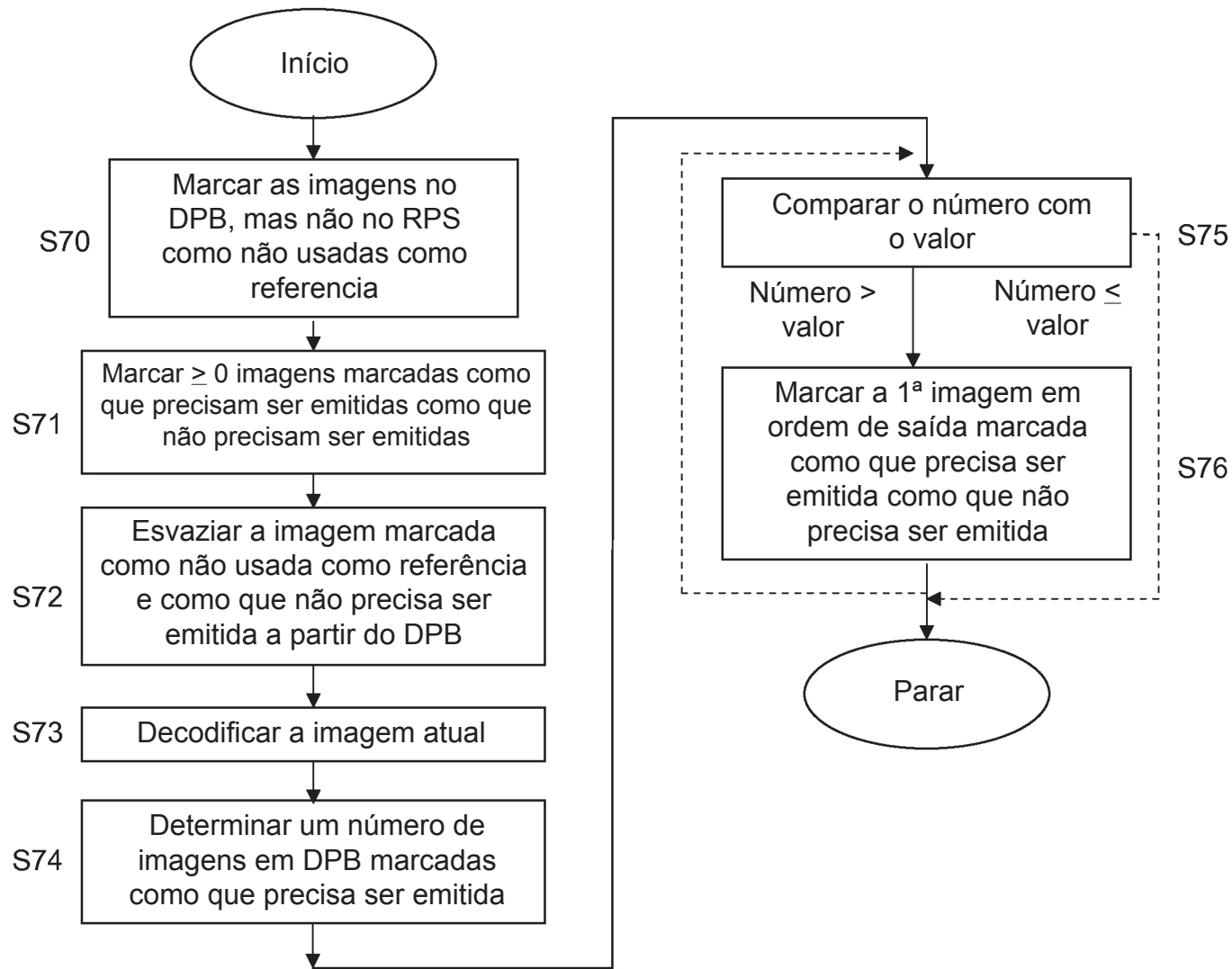


Fig. 22

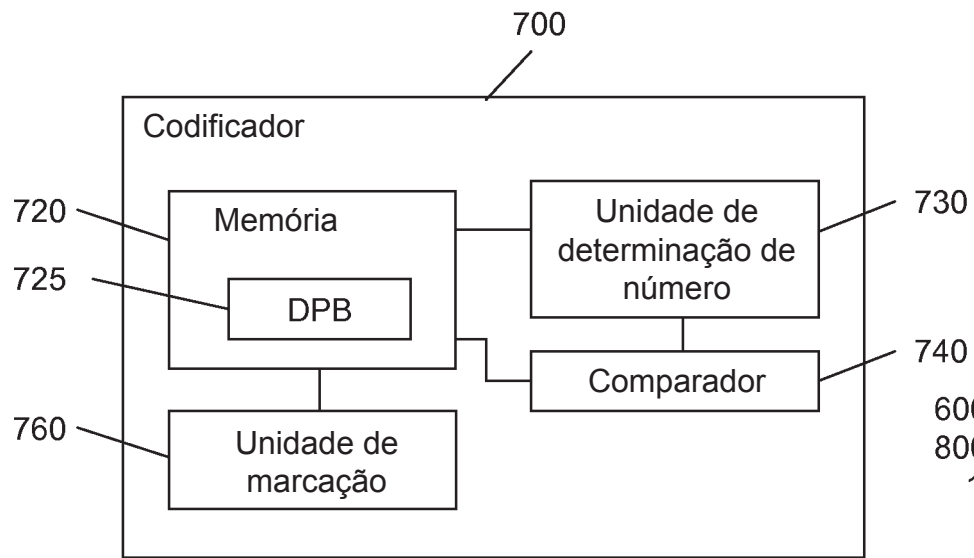


Fig. 24

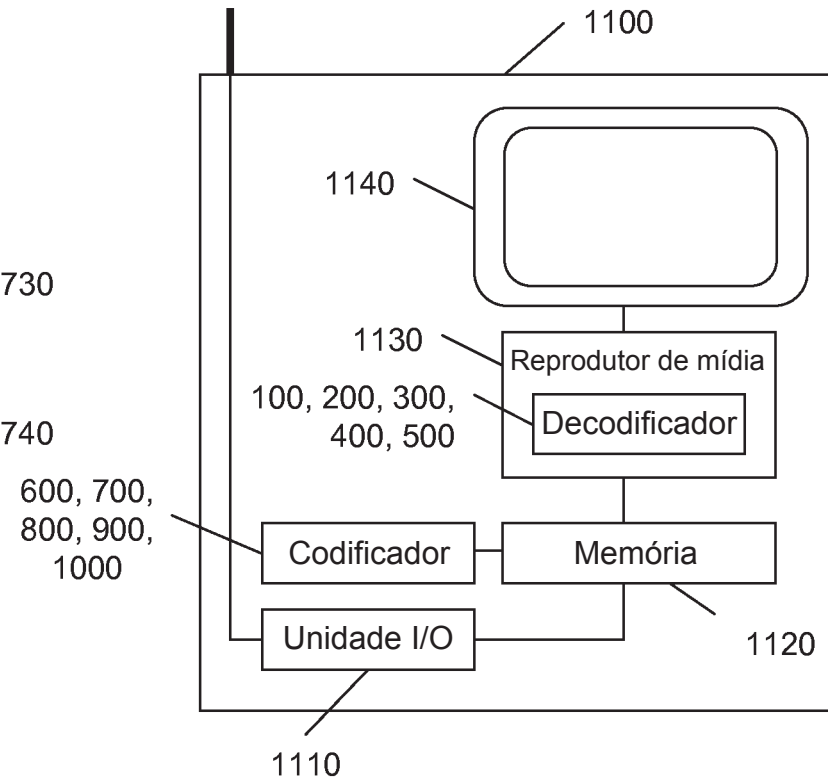


Fig. 28

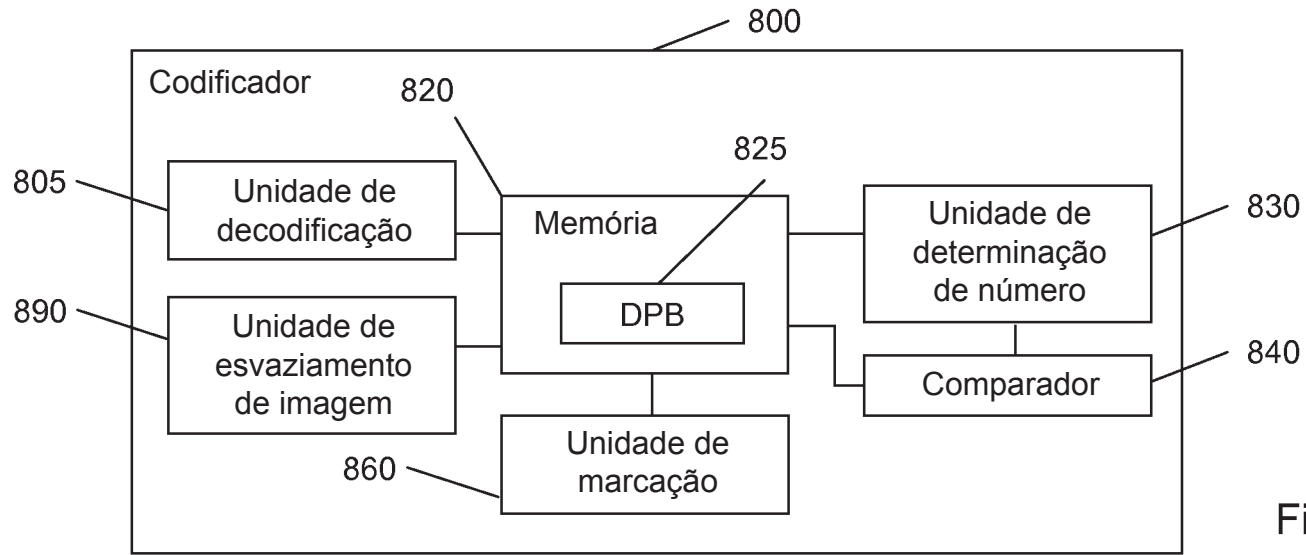


Fig. 25

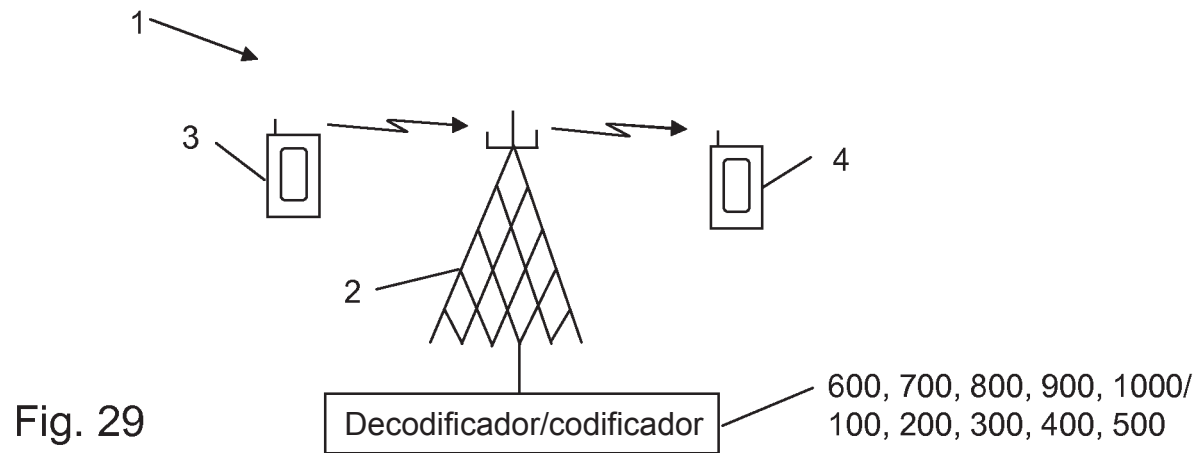


Fig. 29

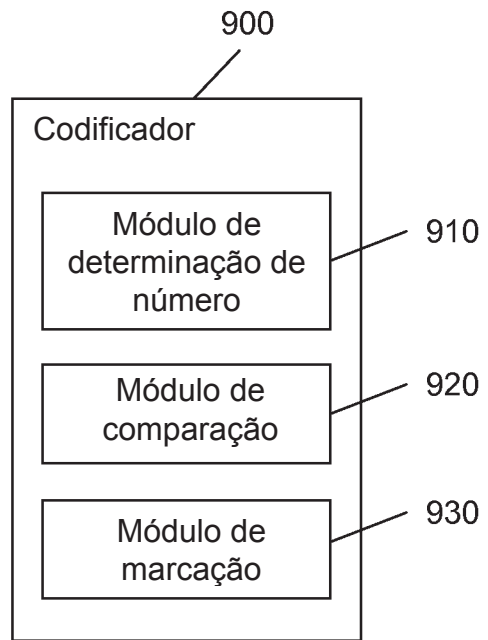


Fig. 26

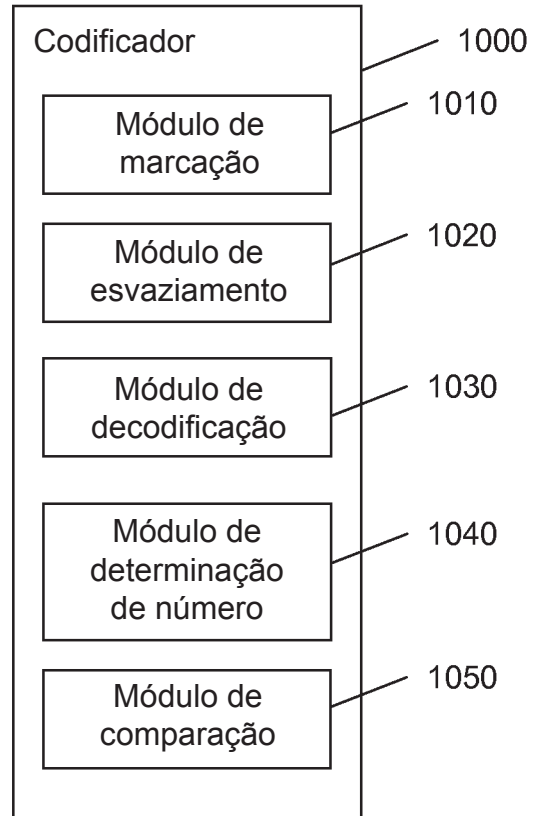


Fig. 27

Camada *sps_max_num_reorder_pic*

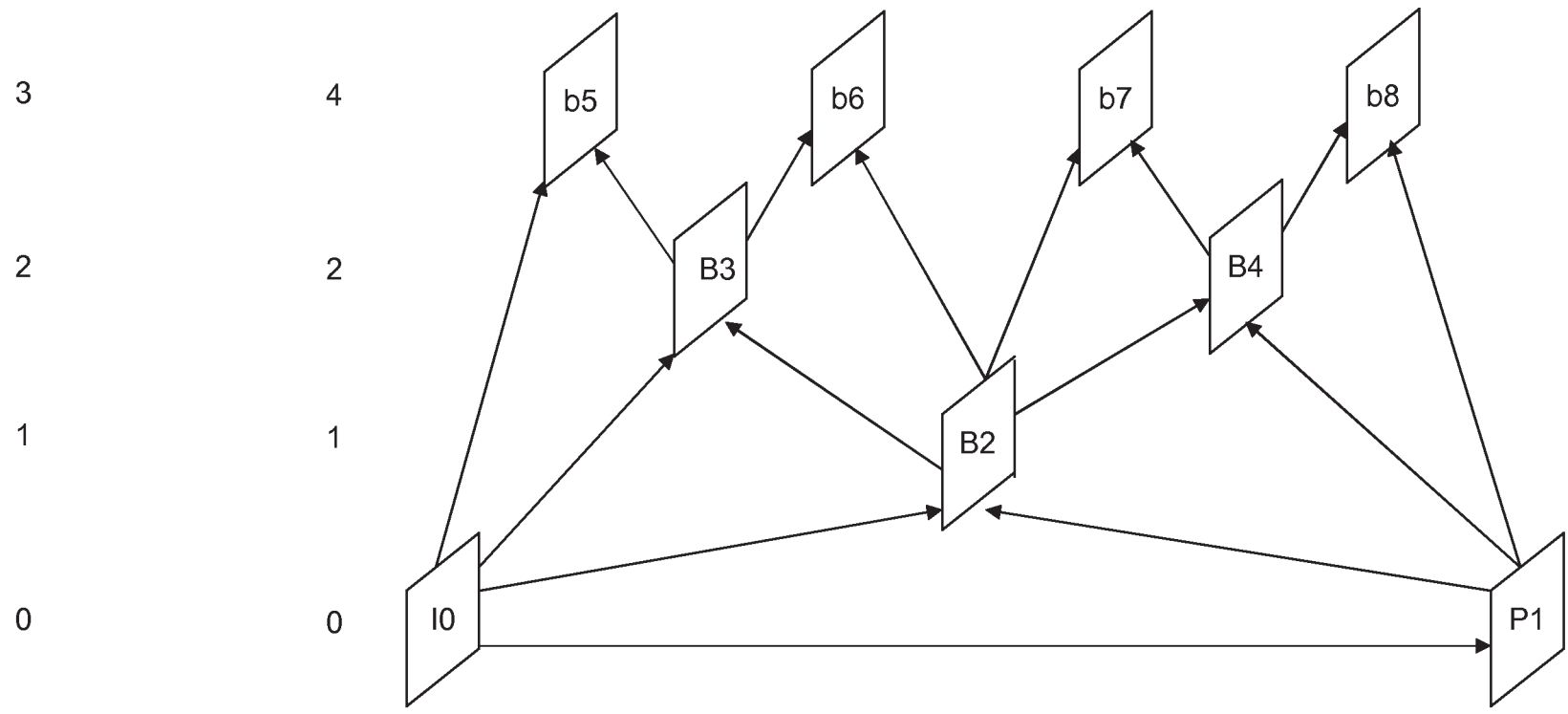


Fig. 30