



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

(11) PI 0918901-7 B1



(22) Data do Depósito: 16/09/2009

(45) Data de Concessão: 15/09/2020

(54) Título: MÉTODO E DISPOSITIVO PARA DECODIFICAR BLOCO DE TRANSPORTE, INSTRUÇÕES EXECUTÁVEIS POR PROCESSADOR DE ARMAZENAMENTO EM MEMÓRIA, E MÉTODO E DISPOSITIVO PARA DECODIFICAÇÃO ITERATIVA DE BLOCO DE TRANSPORTE

(51) Int.Cl.: H03M 13/00; H03M 13/09; H03M 13/29; H04L 1/00; H04L 1/18.

(52) CPC: H03M 13/00; H03M 13/09; H03M 13/29; H04L 1/00; H04L 1/18.

(30) Prioridade Unionista: 19/09/2008 US 12/234,067.

(73) Titular(es): OPTIS CELLULAR TECHNOLOGY, LLC.

(72) Inventor(es): RAJARAM RAMESH; HAVISH KOORAPATY; JUNG-FU CHENG; KUMAR BALACHANDRAN.

(86) Pedido PCT: PCT IB2009006856 de 16/09/2009

(87) Publicação PCT: WO 2010/032108 de 25/03/2010

(85) Data do Início da Fase Nacional: 18/03/2011

(57) Resumo: MÉTODO E DISPOSITIVO PARA DECODIFICAÇÃO ITERATIVA DE UM BLOCO RECEBIDO, MÉTODO E DISPOSITIVO PARA DECODIFICAR UM BLOCO DE TRANSPORTE, DISPOSITIVO PARA DECODIFICAÇÃO ITERATIVA DE UM BLOCO DE TRANSPORTE, E, INSTRUÇÕES EXECUTÁVEIS POR PROCESSADOR DE ARMAZENAMENTO DE MEMÓRIA. A decodificação iterativa de blocos pode ser continuada ou terminada com base na verificação de CRC. Em uma modalidade exemplo, uma iteração de um processo de decodificação iterativa é efetuada em um bloco cujos bits de informação são cobertos por um CRC. O processo de decodificação iterativa é parado se CRC é verificada para um número pré-determinado de iterações consecutivas. Em uma outra modalidade exemplo, uma iteração de decodificação é efetuada em um sub-bloco particular de sub-blocos múltiplos de um bloco de transporte, que inclui uma única CRC sobre a totalidade do bloco de transporte. A CRC é verificada usando bits decodificados obtidos a partir de uma iteração de decodificação no sub-bloco particular e bits decodificados obtidos a partir de iterações de decodificação anteriores nos outros sub-blocos dos sub-blocos múltiplos. A iteração de decodificação é então efetuada em um diferente sub-bloco se a CRC não é verificada. Também, as iterações de decodificação para os sub-blocos (...).

“MÉTODO E DISPOSITIVO PARA DECODIFICAR BLOCO DE TRANSPORTE, INSTRUÇÕES EXECUTÁVEIS POR PROCESSADOR DE ARMAZENAMENTO EM MEMÓRIA, E MÉTODO E DISPOSITIVO PARA DECODIFICAÇÃO ITERATIVA DE BLOCO DE TRANSPORTE”

REFERÊNCIA CRUZADA PARA O PEDIDO RELACIONADO

[0001] Este Pedido de Patente Não Provisório dos Estados Unidos reivindica o benefício de prioridade do Pedido de Patente Provisório U.S. de Nr. 60/973,500, depositado em 19 de setembro de 2007, e com o título de "*Method and Arrangement in a Telecommunication System*". Pedido de Patente Provisório U.S. de No. 60/973,500 é incorporado aqui para referência em sua totalidade.

CAMPO TÉCNICO

[0002] A presente invenção se refere geralmente aos sistemas de comunicações e em particular para redução de complexidade para decodificadores iterativos usando verificação de CRC.

FUNDAMENTO DA INVENÇÃO

[0003] Muitos termos especializados e abreviações são usados nas artes das comunicações. Pelo menos, alguns dos seguintes termos são referidos dentro do texto que segue, tal como neste background e / ou seções de descrição. Assim sendo, os seguintes termos e abreviações estão aqui definidos:

3GPP	Projeto de Parceria de Terceira Geração
CRC	Verificação de redundância cíclica
E-UTRAN	UTRAN Evoluído
LTE	Evolução a Longo Prazo
UMTS	Sistema de Telecomunicações de Terminais

Móveis Universal

UTRA	Acesso via Rádio Terrestre Universal
UTRAN	Rede de Acesso via Rádio Terrestre do UMTS
WiMAX	Interoperabilidade Mundial para Acesso de

Microondas

[0004] A telecomunicação forma o suporte da sociedade orientada à informação

de hoje e pode ser realizada usando transmissão e recepção com fio ou sem fio. A capacidade de um canal de comunicação é tipicamente limitada pela largura de banda. A largura de banda de um canal de comunicação é por sua vez limitada por limites naturais e artificiais impostos por, por exemplo, as propriedades do meio, o espectro alocado para comunicação através de restrições de projeto, e autoridades de regulação ou operadores.

[0005] A utilização de um canal de comunicações, mesmo dada uma alocação finita do espectro eletromagnético, pode ser aumentada adotando qualquer de um número de diferentes esquemas. Esses esquemas permitem mais informação a ser comunicada em uma dada alocação de espectro. A utilização eficiente do espectro pode reduzir o custo de serviços de comunicação sendo fornecidos, pode permitir serviços de comunicação mais ricos a serem fornecidos, ou ambos.

[0006] Um exemplo de esquema de comunicação envolve a codificação de informação. Informação é codificada em um dispositivo de transmissão, e a informação codificada é transmitida através de um canal. A informação codificada transmitida se propaga através do canal e é decodificada em um dispositivo de recepção para recuperar pelo menos, a aparência da informação original. Com codificação apropriada, a informação pode ser comprimida e / ou tornada mais resistente à interferência no canal de comunicações. Em outras palavras, esquemas de codificação podem ser empregados para aumentar a utilização eficiente de espectro eletromagnético.

[0007] Códigos de canal práticos que alcançam desempenho próximo da capacidade estão agora amplamente em uso nos padrões de comunicação sem fio correntes ou planejados para padrões futuros. Exemplos de tais códigos são os códigos turbo usados nos sistemas de UTRAN, E-UTRAN, e WiMAX. O decodificador ótimo para tais códigos não é prático para implementar. Conseqüentemente, decodificadores iterativos subótimos são usados. Tais decodificadores são conhecidos não obstante o desempenho ótimo de abordagem para variações de operação práticas. Esses decodificadores tipicamente requerem muitas iterações, com a complexidade do decodificador aumentando

proporcionalmente com o número de iterações empregado para alcançar uma taxa de erro de quadro residual.

[0008] Em sistemas de comunicação sem fio de terminais móveis, por exemplo, tais decodificadores são implementados nos terminais de comunicação móveis com restrições de tamanho, potência de processamento, e custo. Quando o sistema suporta altas taxa de dados, a complexidade do decodificador requerida coloca uma carga nos recursos limitados de um terminal de comunicação móvel. Tais restrições de complexidade também afetam a estação base onde os sinais para muitos usuários podem ter de ser decodificados concorrentemente. Ainda mais, tal um sistema pode usar segmentação do bloco de transporte em que um maior bloco de transporte é segmentado em múltiplos menores sub-blocos de código. Os sub-blocos são decodificados antes do bloco de transporte poder ser totalmente decodificado.

[0009] Por conseguinte, é de interesse reduzir a complexidade do decodificador iterativo enquanto também tentando manter um nível desejado de desempenho. Um tal método para reduzir complexidade envolve dinamicamente reduzir o número de iterações de decodificação empregado com base na palavra código recebida. Numerosas regras de parada para reduzir o número de iterações para menos do que um máximo pré-definido foram descritas na arte anterior.

[0010] A regra de parada para reduzir o número de iterações de decodificação fornece algum critério que quando encontrado, pode indicar com um alto grau de confiança relativo e / ou de aceitabilidade que o bloco de código foi decodificado com sucesso. Quando o critério esboçado pela regra de parada é encontrado, um processo de decodificação é parado. Tipicamente, o número de iterações usado para decodificar diferentes blocos de código pode diretamente ou indiretamente depender da proporção sinal para ruído do sinal recebido. De fato, seria benéfico para o decodificador aumentar a quantidade de esforço necessário para tomar uma decisão de decodificação como a proporção de sinal para ruído do sinal recebido diminui.

[0011] Muitas classes de códigos em uso, tal como códigos turbo, podem

mapear o esforço a ser empregado diretamente para o número de iterações usado pelo decodificador. Dado uma taxa de erro de quadro alvo que está para ser encontrada, o uso de uma regra de parada pode reduzir o número médio de iterações envolvido na decodificação quando comparado com um algoritmo que simplesmente usa um número de iterações fixo.

[0012] Assim sendo, a regra de parada pode ser usada para reduzir o número de iterações efetuado por um decodificador turbo iterativo. Essas regras podem ser classificadas amplamente em regras de decisão severas, regras de decisão leves, e uma regra de CRC. Regras de decisão severas são com base na falta de mudanças em decisões de bit entre iterações. Regras de decisão leves com base na comparação de uma métrica, que é derivada a partir dos valores de confiabilidade de bit de software gerados após uma iteração, com um limite.

[0013] A regra de CRC se baseia na ausência de erro detectado no final de uma iteração para decidir se um bloco de código decodificado está correto. Se a CRC indica uma falta de um erro de bloco de código, decodificação é parada. O uso da CRC como um critério de parada foi descrito no seguinte documento: 'Parar Rules for Turbo Decoders,' por Matache, Dolinar e Pollara, que foi publicado no NASA JPL TMO Progress Report 42-142, 15 de agosto de 2000, e correntemente disponível em http://tmo.jpl.nasa.gov/progress_report/42-142/titulo.htm.

[0014] Independente da regra de parada usada, um decodificador prático configura um limite no número máximo de iterações de decodificação (por exemplo, $I_{max}=8$). No final da última iteração permitida, a regra de parada pode ser usada para determinar se o bloco de código foi decodificado com sucesso. Contudo, a decodificação iterativa é então concluída independente se a regra de parada foi encontrada.

[0015] Em um moderno sistema de comunicação de taxa de dados alta, por exemplo, um bloco de transporte pode conter dezenas ou mesmo centenas de milhares de bits. Por razões práticas, um "grande" bloco de transporte é assim segmentado em sub-blocos múltiplos de código, cada um dos quais é usualmente menor do que um tamanho máximo pré-definido. Por exemplo, no LTE do sistema

UMTS (por exemplo, Especificação Técnica 3GPP TS 36.212 "Multiplexação e codificação de canal (Versão 8)" emitida pelo Projeto de Parceria de Terceira Geração), o tamanho máximo de sub-bloco de código é configurado para 6144 bits, e uma sequência de CRC é calculada e anexada a cada sub-bloco.

[0016] FIGS. 1A e 1B ilustram exemplo de esquemas de segmentação de bloco de código. Geralmente, cada um inclui um bloco de transporte 100 que é segmentado em sub-blocos múltiplos 102. Sequências de CRC 104 e / ou 106 são incluídas com cada sub-bloco 102. Mais especificamente, FIG. 1A ilustra um exemplo de segmentação de bloco de código e esquema de anexação de CRC de acordo com E-UTRAN. FIG. 1B ilustra um exemplo de segmentação de bloco de código e esquema de anexação de CRC de acordo com WiMAX / UTRA.

[0017] Na Fig. 1A, o bloco de transporte 100A inclui "n" sub-blocos 102A (e.g., sub-bloco 102A(1), sub-bloco 102A(2)... sub-bloco 102A(n)), com "n" representando um inteiro positivo. Cada sub-bloco 102 A inclui e está associado com uma CRC 104. CRCs 104 são denominadas CRCs internas. CRC 106 é denominada uma CRC externa. CRC externa 106 pode ser incluída como parte de um (ou dois ou uns poucos) sub-bloco 102 A, mas ela está associada com sub-blocos múltiplos 102 A, tal como cada um dos sub-blocos 102 A do bloco de transporte 100A. Assim sendo, com o LTE de UTRA, cada um dos sub-blocos 102 A é protegido por uma CRC interna 104, e o bloco de transporte 100A total é também protegido por uma outra sequência de verificação, a CRC externa 106.

[0018] Na Fig. 1B, o bloco de transporte 100B inclui "n" sub-blocos 102B (e.g., sub-bloco 102B(1), sub-bloco 102B(2)... sub-bloco 102B(n)), com "n" representando um inteiro positivo. Bloco de transporte 100B inclui e está associado com uma CRC, tal como a CRC externa 106. A CRC externa 106 pode ser incluída como parte de um dos sub-blocos, tal como o último sub-bloco 102B(n). Contudo, CRC externa 106 está associada com sub-blocos múltiplos 102B, tal como cada um dos sub-blocos 102B do bloco de transporte 100B. Sub-bloco 102B(n) pode também incluir bits de preenchimento. Deve ser notado que uma CRC externa 106 também pode ser dividida entre dois ou mais sub-blocos 102.

[0019] A codificação turbo é usada em muitos sistemas de comunicação sem fio devido às suas capacidades de desempenho. Tipicamente, o total de bloco de dados é dividido em muitos sub-blocos, cada um dos quais é codificado separadamente usando o codificador turbo. Esta divisão e codificação separada são pretendidas para negociar memória e especificações de desempenho do decodificador turbo. O decodificador turbo opera em um modo iterativo — cada iteração usa os resultados da iteração anterior como entrada e tipicamente melhora a confiabilidade da sequência de bits decodificados.

[0020] Geralmente, um bloco de transporte contendo sub-blocos múltiplos é decodificado decodificando os sub-blocos componentes na ordem sequencial em qual eles formam o bloco de transporte. Se um sub-bloco é encontrado ser correto, decodificação turbo para aquele sub-bloco pode ser parada, e o decodificador pode prosseguir para processar o seguinte sub-bloco. Por outro lado, se um sub-bloco é encontrado ser incorreto, decodificação do resto do bloco de transporte pode ser abortada. O efeito de ambos os tipos de conclusão de decodificação pode contribuir para uma redução na quantidade total de recursos de decodificação empregados.

[0021] Regras de parada antecipada podem ser implementadas para determinar quando decodificação pode ser concluída antes de um número máximo permitido de iterações. Há dois tipos de eventos de erro associados com regras de parada antecipada: Em um "evento perdido", um sub-bloco errôneo é declarado como correto. Em tal um evento, o receptor continua para decodificar o resto dos sub-blocos no bloco de transporte, que resulta em um aumento desnecessário em complexidade. Em um "evento de alarme falso", um sub-bloco decodificado de forma correta é declarado como estando em erro. Este tipo de evento é também incômodo. Quando um dos sub-blocos é declarado incorreto, ele é sensível para abortar uma decodificação do restante dos sub-blocos. Contudo, se tal detecção não é confiável, capacidade de transmissão de dados pode ser desnecessariamente reduzida, e os recursos de decodificação aplicados são concomitantemente aumentados.

[0022] Quando a capacidade de transmissão de dados é reduzida, a quantidade

de dados sendo comunicada por usuário é reduzida e / ou o número de usuários que pode ser atendido em um dado nível de serviço é reduzido. Quando recursos de decodificação são aumentados, especialmente com um dispositivo de terminal móvel, o nível de complexidade e quantidade de dreno de bateria também aumenta. Regras de parada existentes não tratam eficientemente essas questões na medida que elas são relativamente complexas.

[0023] Conseqüentemente, há uma necessidade de tratar essas deficiências no estado corrente da arte. Tais deficiências e outras necessidades são tratadas por um ou mais das várias modalidades da presente invenção.

SUMÁRIO

[0024] Modalidades de exemplos que são aqui descritas implicam em métodos de dinamicamente parar iterações de decodificação a fim de pelo menos, mitigar ou melhorar algumas das desvantagens de abordagens convencionais. Mais especificamente, modalidades exemplos descrevem métodos para reduzir complexidade de decodificação quando decodificação de sub-blocos que coletivamente formam um bloco de transporte.

[0025] É um objeto de determinada modalidade (s) da presente invenção para fornecer regras robustas com base em CRC para parar iterações de um processo de decodificação.

[0026] É um outro objeto de determinada modalidade (s) da presente invenção para reduzir a complexidade de decodificação de um bloco de transporte contendo sub-blocos múltiplos.

[0027] É ainda um outro objeto de determinada modalidade (s) da presente invenção, reduzir a complexidade de decodificação turbo usando uma regra de parada de verificação de CRC robusta para concluir um processo de decodificação iterativo.

[0028] Em uma modalidade de exemplo, há um método para decodificação iterativa de um bloco recebido cujos bits de informação são cobertos por uma CRC. O método inclui atos de realizar a iteração de decodificação, verificar uma CRC, e para um processo de decodificação se a CRC verificou para um número pré-

determinado de iterações consecutivas.

[0029] Em uma outra modalidade de exemplo, há um método para decodificar um bloco de transporte que inclui sub-blocos múltiplos, cada um do qual pode ser decodificado iterativamente. O bloco de transporte também inclui uma única CRC sobre uma totalidade do bloco de transporte. Uma iteração de decodificação é efetuada em um sub-bloco particular dos sub-blocos múltiplos. A CRC para a totalidade do bloco de transporte é verificada usando bits decodificados obtidos a partir da iteração de decodificação no sub-bloco particular e bits decodificados obtidos a partir das iterações de decodificação anteriores em outros sub-blocos dos sub-blocos múltiplos. A iteração de decodificação é então efetuada em um diferente sub-bloco se a CRC não verifica. Alternativamente, as iterações de decodificação para todos os sub-blocos dos sub-blocos múltiplos podem ser concluídas se a CRC verifica consecutivamente para um número pré-determinado de consecutivas iterações de decodificação.

[0030] Uma vantagem de determinadas modalidade (s) da presente invenção é que a complexidade de um dispositivo de terminal móvel, por exemplo, pode ser reduzida possibilitando conclusão apropriadamente antecipada de uma decodificação iterativa de sub-blocos de um bloco de transporte com impacto mínimo no desempenho de decodificação.

[0031] Uma outra vantagem de determinadas modalidade (s) da presente invenção é que operação mais rápida de um, e.g., processo de decodificação turbo, pode ser alcançada devido à conclusão antecipada. A conclusão antecipada pode permitir área de chip e consumo de potência inferior.

[0032] Modalidades adicionais são descritas e / ou aqui reivindicadas. Exemplo de modalidades adicionais incluem, a título de exemplo, mas não limitação, métodos, dispositivos, arranjos, memória, sistemas, e assim por diante. Aspectos adicionais da invenção são estabelecidos em parte na descrição detalhada, desenhos, e reivindicações que seguem, e em parte pode ser derivado a partir da descrição detalha e desenhos, ou pode ser aprendido através de prática da invenção. É para ser entendido que ambos a descrição geral precedente e a

seguinte descrição detalhada são exemplares e explanatórias somente e não são restritivas da invenção como divulgado ou como reivindicado.

DESCRIÇÃO BREVE DOS DESENHOS

[0033] Um mais completo entendimento da presente invenção pode ser obtido com referência a seguinte descrição detalhada quando considerada em conjunto com os desenhos anexos onde:

[0034] FIG. 1A ilustra um exemplo de segmentação de bloco de código e esquema de anexação de CRC de acordo com E-UTRAN.

[0035] FIG. 1B ilustra um exemplo de segmentação de bloco de código e esquema de anexação de CRC de acordo com WiMAX.

[0036] FIG. 2 é um diagrama em bloco de um exemplo de sistema de comunicações incluindo um dispositivo de transmissão e um dispositivo de recepção.

[0037] FIG. 3 ilustra um exemplo de composição da informação transmitida.

[0038] FIG. 4 ilustra um exemplo de estrutura para um decodificador iterativo.

[0039] FIG. 5 é um fluxograma de um exemplo geral de método para decodificar um bloco de transporte que é construído a partir de sub-blocos múltiplos tendo associados CRCs.

[0040] FIG. 6 ilustra um exemplo de algoritmo iterativo para decodificar um bloco de transporte quando cada sub-bloco está associado com uma CRC.

[0041] FIG. 7 ilustra um exemplo alternativo de algoritmo iterativo para decodificar um bloco de transporte quando cada sub-bloco está associada com uma CRC.

[0042] FIG. 8 é um fluxograma de um exemplo geral de método para decodificação iterativa de um bloco de transporte tendo uma CRC quando sub-blocos múltiplos estão associados com a CRC.

[0043] FIG. 9 é um diagrama em bloco ilustrando um exemplo de concatenação de bits decodificados entre iterações quando decodificando um bloco de transporte tendo uma CRC quando sub-blocos múltiplos estão associada com a CRC.

[0044] FIG. 10 é um diagrama em bloco de exemplo de dispositivos que podem

ser usados para implementar modalidades para decodificação iterativa de blocos com CRCs.

DESCRIÇÃO DETALHADA

[0045] Conforme descrito aqui acima, pode haver ambos "eventos perdidos" e "eventos de alarme falso" causados pelas regras de parada antecipada com a decodificação iterativa. É desejável manter as probabilidades para ambos o "evento perdido" e o "evento de alarme falso" tão baixo quanto factível em atribuir regras de parada antecipada. Probabilidades de "Evento perdido" estão presentes em todos os três tipos de parar regras: regras de decisão severas e leves assim como a regra de CRC. Contudo, a regra de CRC tem uma vantagem distintiva de exibir uma probabilidade de "evento de alarme falso" essencialmente zero.

[0046] Deve ser notado que, dado uma anexação de CRC para um bloco, há uma pluralidade de maneiras para conceber diferentes regras de parada com diferentes níveis de confiabilidade. Uma abordagem simples poderia resultar em um resultado em um limite inferior nas probabilidades de erro de bloco de código, que é em última instância indesejável. Isto é ainda exacerbado quando um bloco de transporte é dividido em sub-blocos múltiplos com uma regra de parada aplicada a uma decodificação iterativa de cada sub-bloco. Isto é um foco de determinadas modalidades descritas para evitar tal um resultado.

[0047] FIG. 2 é um diagrama em bloco de um exemplo de sistema de comunicações 200 incluindo um dispositivo de transmissão 202 e um dispositivo de recepção 204. Como ilustrado, o sistema de comunicações 200 opera sobre um canal 206 e inclui um sinal 208 em adição ao dispositivo de transmissão 202 e dispositivo de recepção 204. Dispositivo de transmissão 202 inclui um transmissor 210 e um codificador 212. Dispositivo de recepção 204 inclui um receptor 214 e um decodificador 216.

[0048] Em um exemplo de operação, o dispositivo de transmissão 202 transmite um sinal 208 sobre o canal 206 para o dispositivo de recepção 204. Dispositivo de recepção 204 recebe o sinal 208 a partir do dispositivo de transmissão 202 através do canal 206. Mais especificamente, o codificador 212 codifica bits de informação

em uma comunicação que são modulados e então transmitidos pelo transmissor 210 como sinal 208 sobre o canal 206. No dispositivo de recepção 204, o sinal 208 é recebido pelo receptor 214 e convertido para banda base. Decodificador 216 é adaptado para decodificar o sinal para recuperar os bits de informação originais. Um exemplo de composição do sinal 208 é aqui descrito abaixo com particular referência à Fig. 3.

[0049] Deve ser entendido que um único dispositivo pode funcionar como um dispositivo de transmissão 202 em um momento e / ou com relação para uma comunicação e como um dispositivo de recepção 204 em um outro momento e / ou com relação a uma outra comunicação. Exemplos de dispositivos de transmissão 202 e dispositivos de recepção 204 inclui, a título de exemplo, mas não limitação, nó de comunicação de rede, terminais remotos, e outros dispositivos que são capazes de se comunicar sobre um canal 206. Nó de comunicação de rede pode incluir, por exemplo, uma estação de transceptor base, uma estação rádio base, um nó B, um ponto de acesso, e assim por diante. Terminais remotos podem incluir, por exemplo, um terminal de comunicação móvel, uma estação terminal móvel, uma estação de assinante, um cartão ou módulo de comunicação, e assim por diante. Exemplos gerais de implementações de dispositivo para dispositivos de transmissão / de recepção 202 / 204 são aqui descritos abaixo com particular referência à Fig. 10.

[0050] Geralmente, o canal 206 pode ser um canal com fio ou um canal sem fio. Independentemente, o canal 206 afeta a transmissão, propagação, e / ou recepção de sinal 208. Por esta e outras razões tais como eficiência, o sinal 208 é codificado pelo codificador 212. A decodificação é efetuada pelo decodificador 216. Um exemplo de modalidade de decodificador iterativo é aqui descrito abaixo com particular referência à Fig. 4.

[0051] FIG. 3 ilustra um exemplo de composição 300 de informação transmitida. Composição 300 mostra ainda detalhes de exemplo de operações de codificação efetuadas no dispositivo de transmissão. Composição 300 inclui um bloco de bit de informação D, uma CRC associada, e um bloco de bit codificado V. O bloco de bit de informação D e a CRC associada são juntamente referidos como um bloco

concatenado U. Uma operação de codificação de canal 302 é também efetuado como parte da operação de codificação.

[0052] Para um exemplo de operação de codificação, o bloco de bit de informação D é para ser transmitido. Uma CRC é anexada ao bloco de bit de informação D para formar o bloco concatenado U. O bloco U é codificado com operação de codificação de canal 302 usando o codificador 212 (da Fig. 2). Operação de codificação de canal gera o bloco codificado V. Bloco V é então modulado e transmitido para um dispositivo de recepção tendo um decodificador 216 (das Figs. 2 e 4).

[0053] FIG. 4 ilustra um exemplo de estrutura para um decodificador iterativo 216. Como ilustrado, o decodificador iterativo 216 inclui uma iteração de decodificação 402 e uma verificação de regra de parada 404. Em um exemplo de operação geralmente, o decodificador iterativo 216 aceita como entrada, o bloco de símbolo recebido V' e produz como estimativa de saída U'. Como explicado ainda aqui abaixo, múltiplas iterações de decodificação 402 podem ser efetuadas como parte de um processo de decodificação iterativa. O processo de decodificação iterativa pode ser continuado ou concluído com base em se a regra de parada é verificada na verificação de regra de parada 404. Exemplos de modalidades empregam regras de parada que implicam em verificar CRCs sobre uma ou mais iterações. O número de iterações, assim como o número de verificações de CRC com sucesso ou falhos, pode ser monitorado.

[0054] Mais especificamente para o decodificador iterativo 216, o bloco de símbolo recebido severo ou leve, V', é aceito como entrada. O bloco de símbolo recebido V' corresponde ao bloco codificado V como entrada para o decodificador iterativo 216. A estrutura iterativa de alto nível do decodificador iterativo 216 é mostrada na Fig. 4. Após cada iteração de decodificação 402, uma estimativa de tentativa U'(i) do bloco U é gerada, onde "i" significa a i-ésima iteração. O decodificador iterativo 216 eventualmente emite a estimativa U', que inclui o bloco de bit de informação detectado D', se a decodificação é com sucesso. Por outro lado, ele sinaliza um erro de bloco ou quadro se a decodificação é sem sucesso.

[0055] O bloco de iteração de decodificação 402 pode incluir uma ou mais decodificadores. A título de exemplo, o bloco de iteração de decodificação 402 pode incluir pelo menos, um decodificador turbo. Com decodificação turbo, duas operações de decodificação são efetuadas por ciclo de decodificação. Cada iteração de um decodificador turbo pode, por conseguinte, ser considerado uma ou duas operações de decodificação. Em outras palavras, cada iteração pode efetivamente compreender uma ou duas "sub-iterações" dependendo da estrutura do codificador. Com codificação turbo, dois codificadores componentes são usados, assim duas operações de decodificação ou sub-iterações são usadas, um para cada codificador componente. A regra de parada para o bloco de verificação de regra de parada 404 pode então ser aplicada após cada sub-iteração ou após uma completa iteração incluindo duas sub-iterações.

[0056] Dois conjuntos de exemplos de modalidades são descritos abaixo. Um primeiro conjunto de exemplos de modalidade é direcionado aos blocos de transporte 100A (da Fig. 1A) tendo sub-blocos múltiplos 102 A com respectivas CRCs internas 104 cobrindo cada sub-bloco 102 A em adição a uma CRC externa 106 que cobre a totalidade do bloco de transporte. Um segundo conjunto de exemplos de modalidade é direcionado ao bloco de transportes 100B de (FIG. 1B) tendo sub-blocos múltiplos 102B e uma CRC externa 106 que cobre a totalidade do bloco de transporte 100B. FIGS. 5-7 ilustram exemplos de implementações do primeiro conjunto de modalidades. FIGS. 8 e 9 ilustram exemplos de implementações do segundo conjunto de modalidades.

[0057] De acordo com o primeiro conjunto de modalidades de exemplos, um método usa uma verificação de CRC interna em consecutivas iterações de decodificação como a regra de parada para concluir a operação do decodificador. Se a CRC interna verifica após cada um de um número pré-determinado de iterações consecutivas, um processo de decodificação iterativa é concluído para o sub-bloco, e a sequência de bits decodificados resultante é considerada estar correta. A CRC externa é verificada com relação à totalidade dos bits decodificados do bloco de transporte antes dos dados poderem ser declarados como válidos. Se o número de

iterações de decodificação em um dado sub-bloco excede um máximo selecionado sem passar a verificação de CRC interna, o dado sub-bloco é assumido errôneo, que implica que o inteiro bloco de transporte é errôneo também. Conclusão antecipada de umas operações de decodificação iterativa nos sub-blocos individuais usando a CRC interna ajuda a reduzir a complexidade do decodificador significativamente. Esta redução de complexidade pode reduzir a área de chip e consumo de potência para o decodificador iterativo.

[0058] De acordo com um primeiro aspecto, um sub-bloco não é declarado correto até verificações de CRC indicarem a falta de um erro para um número pré-determinado de iterações consecutivas (que é denotado por M no exemplo de algoritmos abaixo). Esta regra de parada de CRC consecutiva exibe uma probabilidade de alarme falsa que é eficazmente zero enquanto também, de forma significativa, reduzindo a probabilidade de evento perdido. De acordo com um segundo aspecto, o número pré-determinado de verificações consecutivas de CRC pode ser uma variável dependente, $M(i)$, que depende do número efetivo de iterações de decodificação que foi efetuado.

[0059] De acordo com um terceiro aspecto, os resultados de verificações de CRC das primeiras "poucas" iterações de decodificação, que é denotado por I_{min} , pode ser ignorado com relação a aplicar a regra de parada de CRC consecutiva. De acordo com um quarto aspecto, uma ordem de decodificação para os sub-blocos múltiplos em um bloco de transporte pode ser priorizada para ainda reduzir a complexidade de decodificação global de um bloco de transporte. Respectivos sub-blocos podem ser ordenados responsivos para os respectivos valores de confiabilidade do sub-bloco e então iterativamente decodificado com base na ordenação priorizada. À título de exemplo somente, eles podem ser ordenados responsivos as suas estimativas de variância de ruído e então sub-blocos com maiores estimativas de variância de ruído são primeiramente decodificados. Isto, também, pode resultar em uma redução em complexidade em média detectando um erro de bloco de transporte antecipadamente.

[0060] Para exemplo de modalidades gerais, um método é para decodificação

iterativa de um bloco recebido cujos bits de informação são cobertos por uma CRC. Iterações de um processo de decodificação iterativa são efetuadas no bloco. É apurado se a CRC verifica após cada iteração. Um processo de decodificação iterativa para o bloco é parado se a CRC verifica para um número pré-determinado de iterações consecutivas. Ainda mais, um processo de decodificação iterativa pode ser continuado se a CRC não verifica para o número pré-determinado de iterações consecutivas. Em um exemplo de implementação, o número pré-determinado de iterações consecutivas de verificações de CRC com sucesso é uma variável dependente que depende do número de iterações de decodificação que foi efetuado.

[0061] FIG. 5 é um fluxograma 500 de um exemplo de método geral para decodificar um bloco de transporte que é construído a partir de sub-blocos múltiplos tendo associadas CRCs. Como ilustrado, o fluxograma 500 inclui quatro blocos 502 - 508. O fluxograma 500 pode ser implementado através de um dispositivo, tal como um dispositivo de recepção 204 (da Fig. 2).

[0062] As ações do fluxograma 500, assim como os outros diagramas de fluxo aqui descritos, podem ser efetuadas com instruções executáveis pelo processador. Instruções executáveis pelo processador podem ser incorporadas como hardware, firmware, software, conjunto de circuitos de lógica fixa, combinações deles, e assim por diante. Exemplo de implementações operacionais de instruções executáveis pelo processador inclui, mas não são limitados à, uma memória acoplada a um processador, um circuito integrado de aplicação específica (ASIC), um processador de sinal digital e código associado, alguma combinação deles, e assim por diante.

[0063] Em uma modalidade de exemplo, o fluxograma 500 representa um método para decodificar um bloco de transporte construído a partir de sub-blocos múltiplos. Cada sub-bloco é uma palavra código que pode ser decodificada iterativamente com os bits de informação dela sendo cobertos por uma CRC. O método inclui ações de receber, realizar, parar e continuar. No bloco 502, o bloco de transporte é recebido. No bloco 504, uma iteração de um processo de decodificação iterativa em um sub-bloco particular dos sub-blocos múltiplos é efetuada. Isto pode ser apurado se a CRC para o sub-bloco particular verifica após cada iteração.

[0064] No bloco 506, um processo de decodificação iterativa para o sub-bloco particular é parado se a CRC associada com o sub-bloco particular verifica para um número pré-determinado de consecutivas iterações de decodificação. No bloco 508, o processo de decodificação iterativa para outros sub-blocos para os quais o processo de decodificação iterativa não foi parado é continuado até que as iterações de decodificação para todos os sub-blocos dos sub-blocos múltiplos possam ser paradas. Também, em um exemplo de implementação, um processo de decodificação iterativa é efetuado em um sub-bloco em um momento antes de prosseguir para o seguinte sub-bloco dos sub-blocos múltiplos.

[0065] FIG. 6 ilustra um exemplo de algoritmo iterativo 600 para decodificar um bloco de transporte quando cada sub-bloco está associado com uma CRC. Com algoritmo iterativo 600, a decodificação é iniciada na etapa 602. A contagem de iteração é configurada para $i=1$, e o corrente número de verificações com sucesso consecutivo é configurado para $CHK=0$. Na etapa 604, processamento para iteração i é efetuado. Na etapa 606, a CRC para o bloco emitido $U'(i)$ é verificada. Na etapa 608, é verificada se a CRC é satisfeita. Se não, então o contador de verificação com sucesso consecutivo é re-iniciado para $CHK=0$ na etapa 610.

[0066] Se a CRC não verifica (como verificado na etapa 608), então na etapa 612 o contador de verificação com sucesso consecutivo é incrementado com $CHK = CHK + 1$. Na etapa 614, é verificado se o contador de verificação com sucesso consecutivo se iguala ao número pré-determinado da variável (dependente) de verificações com sucesso consecutivas $M(i)$. Se a regra de parada de CRC consecutivo é verificada (na etapa 614), então na etapa 616 o bloco decodificado $U'(i)$ é emitido.

[0067] Após a etapa 610 ou após uma determinação negativa na etapa 614, o algoritmo continua na etapa 618. Na etapa 618, é determinado se o número máximo de iterações de decodificação foi efetuado. Se sim, então na etapa 620 uma indicação de erro de bloco é emitida. Se não, então a contagem de iteração é incrementada com $i = i+1$ na etapa 622.

[0068] Assim sendo, para determinadas modalidade de exemplos, o processo de

decodificação iterativa não é parado até verificações de CRC consecutivas de $M(i)$ terem sido satisfeitas. A variável $M(i)$ no algoritmo 600 é uma variável dependente. Isto se refere ao número de sucessivas iterações para as quais a CRC é para ser satisfeita na iteração i para iterações de decodificação para parar. Exemplos de implementações alternativas para essas modalidades são descritos abaixo:

[0069] Em uma implementação alternativa, o número pré-determinado de iterações consecutivas, que é denotado por $M(i)$ na Fig. 6, para a qual a CRC é para ser satisfeita não varia com o corrente número de iterações. Por exemplo, M pode ser selecionado como uma constante fixa tendo um pré-determinado valor fixo (e.g., 2, 3, etc.).

[0070] Em uma outra implementação alternativa, o número pré-determinado de verificações com sucesso consecutivas de $M(i)$ pode diminuir conforme a contagem de iteração i aumenta. Por exemplo, $M(i)$ pode ser configurado para 3 quando $i \leq 3$ (i.e., para as primeiras duas iterações), ser configurado para 2 quando $i > 3$, e possivelmente configurado para 1 quando $i = \text{Imax}$. A lógica para diminuir o número pré-determinado de verificações com sucesso consecutivas $M(i)$ é que a confiabilidade da saída do decodificador se torna sucessivamente maior com a contagem de iteração aumentando. Isto é, a probabilidade de um resultado para a verificação de CRC na iteração $i+1$ ser o mesmo que o resultado para a verificação de CRC na iteração i , aumenta conforme i aumenta. Isto conduz à verificações de CRC sucessivas se tornando progressivamente menos relevante na medida que elas fornecem pouca informação adicional.

[0071] Uma área alternativa de aplicação para a qual os princípios da invenção podem ser usados é um dispositivo de equalização de múltiplas passagens / decodificador no qual as operações de equalização e decodificação de canal são aplicadas iterativamente de modo a melhorar o desempenho global do receptor. Em tais aplicações, uma regra de parada de CRC consecutiva como aqui descrita pode ser usada para reduzir a probabilidade de falta enquanto também reduzindo complexidade.

[0072] FIG. 7 ilustra um exemplo alternativo de algoritmo iterativo 700 para

decodificar um bloco de transporte quando cada sub-bloco está associado com uma CRC. Com o algoritmo 700, a regra de parada de verificação de CRC consecutiva é aplicada após um número mínimo de iterações de decodificação, denotado por *Imin*, ter sido efetuado. Por exemplo, *Imin* pode ser configurado para 2, 3, etc. Algoritmo 700 difere do algoritmo 600 (da Fig. 6) pela adição de etapa 702. Na etapa 702, é determinado se pelo menos, o número mínimo de iterações de decodificação *Imin* foi efetuado. Se sim, o algoritmo continua a etapa 606 para verificar a CRC. Se pelo menos, *Imin* de iterações de decodificação não foram efetuadas, então o algoritmo prossegue para a etapa 622 para incrementar o contador de iteração *i*.

[0073] Em determinadas implementações do mundo real, os circuitos de verificação de CRC podem ser firmemente integrados em uma dada estrutura do decodificador iterativo. Por conseguinte, pode não ser possível completamente evitar realizar verificação de CRC para as iterações iniciais. Contudo, em tais configurações, o resultado de verificação de CRC pode ser ignorado até que o número de iterações de decodificação efetuadas alcança *Imin*.

[0074] Em determinados exemplos de modalidade, os sub-blocos múltiplos de um bloco de transporte recebido pode ser priorizado com base nos valores de confiabilidade do sub-bloco. O processo de decodificação iterativa nos sub-blocos múltiplos pode então ser efetuado em uma ordem que é responsiva aos valores de confiabilidade do sub-bloco. Isto pode reduzir a complexidade geral de um decodificador iterativo.

[0075] Quando um bloco de transporte é segmentado em sub-blocos múltiplos (conforme mostrado nas Figs. 1A e 1B), em erro em qualquer um dos sub-blocos considera o bloco de transporte completo inválido. Consequentemente, é desnecessário continuar a decodificação dos sub-blocos restantes do bloco de transporte quando um é confirmado estar em erro. Quando um bloco de transporte total está em erro, detectando um sub-bloco constituinte que está em erro antecipadamente ajuda a reduzir a complexidade.

[0076] De modo a aumentar a probabilidade que blocos de transportes em erro sejam detectados antecipadamente, os sub-blocos podem ser classificados em, e.g.,

diminuir ordem de correspondente confiabilidade de sinal. Em outras palavras, eles podem ser classificados e priorizados com base nos valores de confiabilidade do sub-bloco. Por exemplo, os sub-blocos podem ser classificados de acordo com suas relações de sinal-ruído ou suas relações de sinal-ruído e interferência. Como um outro exemplo, os sub-blocos podem ser classificados de acordo com suas estimativas de variância de ruído ou suas estimativas de potência de ruído e interferência.

[0077] O segundo conjunto de modalidades de exemplos é aplicável aos blocos de transporte nos quais os sub-blocos não têm sequências de verificação individuais (e.g., conforme mostrado na Fig. 1B). Em outras palavras, não há nenhuma CRCs internas. Há uma CRC externa que fornece uma soma lógica de verificação sobre a totalidade do bloco de dados. Uma abordagem direta para lidar este esquema de bloco de transporte é completar as operações de decodificação turbo em todos os sub-blocos (usando o número máximo de iterações), e finalmente realizar a verificação de quadro na sequência inteira para determinar se os dados do bloco de transporte foram recebidos corretamente ou erroneamente.

[0078] Ao contrário para tal uma abordagem "direta" e de acordo com o segundo conjunto de modalidades de exemplos, os métodos descritos podem reduzir a complexidade de um decodificador iterativo usando a CRC externa. Em um exemplo de implementação, um número limitado de iterações de decodificação é efetuado em cada sub-bloco antes de verificar a CRC. Um processo de decodificação iterativa é continuado se a CRC não verifica, mas pode ser parado se a CRC verifica.

[0079] Geralmente, exemplos de modalidades implicam em determinar se uma CRC verifica na totalidade de um bloco de transporte após cada iteração de decodificação em um sub-bloco. Após um número pré-determinado de corretos de verificações de CRC, o processo de decodificação iterativa pode ser concluído. Os sub-blocos também podem ser priorizados para determinar em quais sub-bloco a seguinte iteração de decodificação é para ser aplicada. Detalhes adicionais assim como implementações alternativas estão ainda descritos aqui abaixo.

[0080] Em determinadas modalidades de exemplos, há métodos para decodificar

um bloco de transporte que inclui sub-blocos múltiplos, cada um dos quais pode ser decodificado iterativamente. O bloco de transporte também inclui uma única CRC sobre uma totalidade do bloco de transporte. Os métodos incluem ações de realizar a iteração de decodificação em um sub-bloco particular e verificar a CRC para a totalidade do bloco de transporte. Um primeiro exemplo de método também inclui uma ação de realizar uma iteração de decodificação em um diferente sub-bloco. Um segundo exemplo de método também inclui uma ação de concluir as iterações de decodificação para todos os sub-blocos dos sub-blocos múltiplos.

[0081] Mais especificamente, uma iteração de decodificação é efetuada em um sub-bloco particular dos sub-blocos múltiplos. A CRC para a totalidade do bloco de transporte é verificada usando bits decodificados obtidos a partir da iteração de decodificação no sub-bloco particular e bits decodificados obtidos a partir das iterações de decodificação anteriores em outros sub-blocos dos sub-blocos múltiplos. No primeiro exemplo de método, uma iteração de decodificação é efetuada em um diferente sub-bloco se a CRC não verifica. Ainda mais, as iterações de decodificação para todos os sub-blocos dos sub-blocos múltiplos podem ser concluídas se a CRC verifica consecutivamente para um número pré-determinado de consecutivas iterações de decodificação. O número pré-determinado de iterações consecutivas pode ser configurado, por exemplo, tão baixo quanto um.

[0082] FIG. 8 é um fluxograma 800 de um exemplo geral de método para decodificação iterativa de um bloco de transporte tendo uma CRC quando sub-blocos múltiplos estão associados com a CRC. Como ilustrado, fluxograma 800 inclui seis etapas 802 - 812. O fluxograma 800 pode ser implementado por um dispositivo, tal como um dispositivo de recepção 204 (da Fig. 2). Em uma modalidade de exemplo, o fluxograma 800 representa um método para um dispositivo de recepção que iterativamente decodifica um bloco de transporte. O método inclui ações de receber, realizar, verificar, e realizar condicionalmente com a apuração.

[0083] Na etapa 802, um bloco de transporte tendo sub-blocos múltiplos é recebido. O bloco de transporte inclui uma CRC que está associado com os sub-

blocos múltiplos. Os sub-blocos múltiplos de número "n". Na etapa 804, uma iteração de decodificação é efetuada em um número de sub-blocos dos sub-blocos múltiplos, com o número sendo maior do que zero e menor do que "n".

[0084] Na etapa 808, é verificado se a CRC verifica com relação aos sub-blocos múltiplos. Se a verificação de CRC falha, os sub-blocos a serem decodificados são mudados na etapa 806. Assim sendo, na etapa 804, uma iteração de decodificação é efetuada em pelo menos, um diferente sub-bloco dos sub-blocos múltiplos se a CRC não é verificada para verificar (na etapa 808).

[0085] Se, por outro lado, a CRC é verificada para verificar (na etapa 808), o bloco de transporte pode ser declarado válido na etapa 812. Adicionalmente, a declaração de validade pode ser condicional em um histórico de verificação de CRC encontrando um critério. Quando verificações de CRC com sucesso repetidos são usados como parte de uma regra de parada, a ação(s) de etapa 810 é também efetuada após uma verificação de CRC afirmativa na etapa 808.

[0086] Na etapa 810, se a CRC é verificada para verificar, iterações de decodificação são repetidamente efetuadas em um ou mais sub-blocos até um histórico de verificação de CRC ser determinado como satisfazendo um critério pré-determinado. O critério pré-determinado pode ser, por exemplo, que uma CRC verifica um número pré-determinado total de vezes ou que a CRC verifica um número pré-determinado consecutivo de vezes.

[0087] Conforme um outro exemplo alternativo de implementação, o bloco de transporte pode ser declarado inválido se um número pré-determinado de iterações de decodificação é efetuado em cada sub-bloco dos sub-blocos múltiplos sem a verificação de CRC.

[0088] Em uma outra modalidade de exemplo, um dispositivo é adaptado para iterativamente decodificar um bloco de transporte tendo uma CRC. O dispositivo inclui um receptor, um decodificador, e um processador. O receptor recebe um bloco de transporte tendo sub-blocos múltiplos. O bloco de transporte inclui uma CRC que está associado com os sub-blocos múltiplos. O decodificador decodifica um subconjunto dos sub-blocos múltiplos. O subconjunto é um ou mais sub-blocos mas

menos do que todos os sub-blocos múltiplos. O processador é programado para verificar se a CRC verifica com relação aos sub-blocos múltiplos após o subconjunto ser decodificado pelo decodificador. O processador é também programado para forçar o decodificador a continuar a iterativamente decodificar subconjuntos dos sub-blocos múltiplos se a CRC não verifica após um subconjunto ser decodificado.

[0089] Em um exemplo de implementação, o processador é ainda programado para verificar se a CRC verifica com relação aos sub-blocos múltiplos verificando os bits decodificados que estão concatenados a partir de uma iteração de decodificação corrente no subconjunto dos sub-blocos múltiplos e bits decodificados a partir de uma ou mais iterações de decodificação anteriores em outros sub-blocos dos sub-blocos múltiplos. A concatenação de bits decodificados durante um processo de decodificação iterativa é descrita abaixo com particular referência à Fig. 9.

[0090] À título de exemplo, mas não limitação, operação de um codificador turbo e de um decodificador turbo são descritos. Cada sub-bloco de bits U é passado através do codificador turbo e uma sequência resultante de bits decodificados é obtida. Os bits decodificados para cada um dos sub-blocos são então intercalados, modulados, e enviado sobre o canal sem fio para o receptor. Um codificador turbo tipicamente inclui dois decodificadores constituintes. Os bits de dados compreendendo um sub-bloco são passados através do primeiro codificador constituinte para produzir um primeiro conjunto de bits decodificados. Os bits de dados são então intercalados e passados através do segundo codificador constituinte para produzir um segundo conjunto de bits decodificados. Finalmente, um processo de punção é usado para escolher dentre os bits de dados e os dois conjuntos de bits decodificados para selecionar o conjunto final de bits decodificados. Esta seleção é feita de acordo com uma taxa de codificação e regras de punção. O conjunto final de bits decodificados V é enviado sobre o canal.

[0091] O decodificador turbo funciona iterativamente com dois decodificadores constituintes. Ele primeiro decodifica de acordo com um dos decodificadores, e este processo produz uma versão refinada dos bits de dados. A versão refinada de bits de dados é então passada para o segundo decodificador, com um processo de

intercalação entre eles. Sucessivas iterações alternando entre o primeiro e segundo decodificadores constituintes continuamente refine a versão corrente dos bits de dados. Este processo pode ser continuado, e.g., para uma determinada pré-configuração número de iterações.

[0092] Em uma abordagem direta, o processo de decodificação turbo pode ser efetuado para o máximo número permitido de iterações para todos os sub-blocos. O bloco de dados assim sendo obtido, conforme a concatenação dos bits de dados emitidos a partir dos decodificadores turbo constituintes operando em cada sub-bloco, é então passado através do processo de verificação da CRC. Se a verificação é com sucesso, então o receptor pode reconhecer o bloco de dado recebido e o passa para camadas superiores. Se a verificação não é com sucesso, o receptor pode armazenar o bloco de dados recebido na memória e sinaliza um reconhecimento negativo para o transmissor, que pode retransmitir o bloco de dados.

[0093] Ao contrário, para determinados exemplos de modalidade que estão aqui descritas, uma verificação de CRC é efetuada no bloco de dados após que o decodificador turbo completa uma iteração em um dado sub-bloco. Assim sendo, o decodificador turbo é operado para a seguinte iteração programada no sub-bloco m , e a verificação de CRC é efetuada no bloco concatenado que tem os bits decodificados a partir do sub-bloco m a partir da iteração corrente, mas bits decodificados a partir de outros sub-blocos a partir das iterações anteriores.

[0094] FIG. 9 é um diagrama em bloco 900 ilustrando um exemplo de concatenação de bits decodificados entre iterações quando decodificando um bloco de transporte tendo uma CRC quando sub-blocos múltiplos estão associados com a CRC. Sub-blocos 102B(1), 102B(2)... 102B(n) de um bloco recebido de transporte são mostrados. Após as iterações de decodificação totais D , um conjunto concatenado de bits decodificados é produzido. Eles são ilustrados como bits decodificados 902(1), 902(2)... 902(n).

[0095] Uma outra iteração do decodificador (i.e., a $D+1$ ésima iteração) é efetuada nos bits decodificados 902(2) a partir do sub-bloco original 102B(2). Isto

produz bits decodificados 904(2). Os outros bits decodificados a partir das D iterações anteriores (e.g., bits decodificados 902(1) e 902(n)) são concatenadas com bits decodificados 904(2). A CRC é então verificada com relação a totalidade do bloco de transporte recebido usando os bits decodificados 902(1), 904(2),... 902(n). Sucessivas iterações de decodificação e verificações de CRC podem ser efetuadas de forma similar.

[0096] Se a CRC verifica, então há uma chance relativamente boa que o bloco de transporte inteiro foi corretamente recebido. Assim sendo, uma decodificação pode ser parada. Se a CRC não verifica, então uma outra iteração do decodificador para um diferente sub-bloco é efetuada. Após cada iteração, os bits decodificados concatenados a partir dos vários sub-blocos são verificados contra a CRC com bits decodificados a partir de uma iteração mais recente do decodificador em um sub-bloco sendo incluído com bits decodificados a partir das iterações anteriores nos outros sub-blocos. Assim sendo, os bits decodificados a partir da última iteração para cada sub-bloco são concatenados e verificados contra a CRC com o número de iterações efetuadas em cada um dos sub-blocos sendo potencialmente diferentes em qualquer dado momento.

[0097] A ordem também pode ser priorizada quando selecionando o seguinte sub-bloco a ser iterativamente decodificado. Cada respectivo sub-bloco pode estar associado com um correspondente valor de confiabilidade do sub-bloco. Antes de realizar uma iteração n, os blocos decodificados podem ser priorizados de modo a aumentar a confiabilidade. Sub-blocos com valores de confiabilidade inferior são priorizados para a operação de decodificação.

[0098] Porque sub-blocos relativamente menos confiáveis são mais prováveis de necessitar relativamente mais iterações, a eficiência pode ser aumentada se as seguintes iterações são aplicadas aos sub-blocos de modo a aumentar confiabilidade. Exemplos de valores de confiabilidade do sub-bloco incluem, mas não são limitados a diferentes meios (tal como a média ou o valor mínimo) do valor absoluto dos valores de probabilidade através dos bits decodificados para cada sub-bloco. Outros exemplos de confiabilidade de sub-bloco são métricas de decisão

ponderada.

[0099] Conforme notado acima, em vez de uma única verificação de CRC correta, múltiplas verificações de CRC corretas através de iterações de decodificação podem ser usadas como um critério para parar um processo de decodificação iterativa. Um histórico de verificações de CRC com sucesso e / ou falho pode ser mantido. Efetuando múltiplas verificações, a chance que um sub-bloco incorreto acione uma verificação de CRC com sucesso é, de forma significativa, reduzida.

[00100] Em um exemplo de implementação, uma verificação de CRC pode ser efetuada na estimativa dos bits antes de qualquer operação de decodificação ser efetuada, e uma outra verificação de CRC pode ser efetuada após uma iteração em cada um dos sub-blocos do bloco de transporte. Se a CRC verifica em ambos casos, então o bloco decodificado é altamente provável estar correto. Se a CRC não verifica, então o processo de decodificação de sub-blocos priorizados, junto com a verificação repetida da CRC, pode prosseguir. Um número de verificações de CRC correto consecutivas pode ser usado, por exemplo, como um critério para parar um processo de decodificação iterativa.

[00101] FIG. 10 é um diagrama em bloco 1000 de exemplo de dispositivos 1002 que pode ser usado para implementar modalidades para decodificação iterativa de blocos com CRCs. Como ilustrado, o diagrama de bloco 1000 inclui dois dispositivos 1002(1) e 1002(2), equipamento de interface homem – dispositivo 1012, e uma ou mais redes 1016. Como explicitamente mostrado com dispositivo 1002(1), cada dispositivo 1002 pode incluir pelo menos, um processador 1004, uma ou mais mídias 1006, uma ou mais interfaces de entrada / saída 1008, e pelo menos, uma interconexão 1014. Mídia 1006 pode incluir instruções executáveis pelo processador 1010. Rede (s) 1016 pode ser, à título de exemplo, mas não limitação, uma internet, uma intranet, uma Ethernet, uma rede pública, uma rede privada, uma rede a cabo, uma rede de linha de assinante digital (DSL), uma rede de telefone, uma rede com fio, uma rede sem fio, alguma combinação deles, e assim por diante. Dispositivo 1002(1) e dispositivo 1002(2) pode se comunicar através da rede (s) 1016.

[00102] Para modalidade de exemplos, o dispositivo 1002 pode representar qualquer dispositivo capaz de processamento. Processador 1004 pode ser implementado usando qualquer tecnologia capaz de processamento aplicável, e um pode ser realizado como um processador de propósito geral ou de propósito especial. Exemplos incluem, mas não são limitados à, uma unidade de processamento central (CPU), um processador de sinal digital (DSP), um microprocessador, alguma combinação deles, e assim por diante. Mídia 1006 pode ser qualquer mídia disponível que é incluída como parte de e / ou é acessível pelo dispositivo 1002. Ele inclui mídia volátil e não volátil, mídia removível e não removível, armazenagem (e.g., memória) e mídia de transmissão (e.g., canais de comunicação com fio ou sem fio), mídia lógica codificada por hardware (que pode incluir memória), combinações deles, e assim por diante. Mídia 1006 é mídia tangível quando ela é incorporada como uma fabricação e / ou como uma composição de material.

[00103] Interconexão 1014 interconecta os componentes do dispositivo 1002. Interconexão 1014 pode ser realizada como uma barra de comunicação ou outro mecanismo de conexão e pode diretamente ou indiretamente interconectar vários componentes. Interfaces de I / O 1008 pode incluir (i) uma interface de rede para monitorar e / ou comunicar através de rede 1016, (ii) um dispositivo de exibição interface para exibir informação em uma tela de exibição, (iii) uma ou mais interfaces homem - dispositivo, e assim por diante. Exemplo de interfaces de rede incluem, mas não são limitados a um rádio ou transceptor (e.g., um transmissor e / ou receptor), um modem, um cartão de rede, alguma combinação dele, e assim por diante. Equipamento de interface homem – dispositivo 1012 pode ser um teclado, uma tela de toque, um controle remoto, um mouse ou outro dispositivo de apontamento gráfico, uma tela, um alto-falante, e assim por diante. Equipamento de interface homem – dispositivo 1012 pode ser integrado com ou discreto a partir do dispositivo 1002

[00104] Geralmente, processador 1004 é capaz de executar, realizar, e / ou por outro lado realizar instruções executáveis pelo processador, tal como instruções

executáveis pelo processador 1010. Mídia 1006 é compreendida de uma ou mais mídia acessível pelo processador. Em outras palavras, mídia 1006 pode incluir instruções executáveis pelo processador 1010 que são executáveis pelo processador 1004 para realizar o desempenho de funções pelo dispositivo 1002. Instruções executáveis pelo processador 1010 podem ser incorporadas como software, firmware, hardware, conjunto de circuitos de lógica fixa, alguma combinação dele, e assim por diante. Processador 1004 e instruções executáveis pelo processador 1010 de mídia 1006 pode ser realizada separadamente (e.g., como um código de execução de DSP) ou integrada (e.g., como parte de um circuito integrado de aplicação específica (ASIC)).

[00105] Em implementações de exemplo, um dispositivo 1002 pode compreender um dispositivo de transmissão 202, e um outro dispositivo 1002 pode compreender um dispositivo de recepção 204 (ambos da Fig. 2). Instruções executáveis pelo processador 1010 podem compreender, por exemplo, o decodificador iterativo 216 da Fig. 4. Quando instruções executáveis pelo processador 1010 são executadas pelo processador 1004, as funções que estão aqui descritas podem ser efetuadas. Exemplos de funções incluem, mas não são limitados a aquelas ilustradas através de diagramas de fluxo 500 / 800 e algoritmos 600 / 700, assim como aqueles pertencendo aos recursos ilustrados pelos vários procedimentos, esquemas, e assim por diante.

[00106] Embora modalidades de exemplos foram descritas no contexto de processamento iterativo para decodificação de canal, é aparente que processamento iterativo ocorre em muitas outras situações em sistemas de comunicação sem fio e com fio. Assim sendo, os princípios que estão aqui descritos são também aplicáveis àquelas outras situações.

[00107] Diferente(s) modalidade(s) da invenção pode oferecer uma ou mais vantagens. Geralmente, a complexidade de um dispositivo de comunicação (e.g., um dispositivo de terminal móvel ou estação base) pode ser reduzida possibilitando a conclusão apropriadamente antecipada de decodificação iterativa de sub-blocos de um bloco de transporte. Também, operação mais rápida de um, e.g., processo de

decodificação turbo pode ser alcançado devida à conclusão antecipada. A conclusão antecipada pode permitir área de chip e consumo de potência inferiores.

[00108] Os dispositivos, ações, recursos, funções, métodos, esquemas, estruturas de dados, procedimentos, componentes, etc. das Figs. 1A - 10 são ilustrados nos diagramas que são divididos em múltiplo blocos e outros elementos. Contudo, uma ordem, interconexão, interconexões, inter-relações, esboço, etc. nos quais FIGS. 1A - 10 são descritos e / ou mostrados não são pretendidos serem interpretados como uma limitação, e qualquer número dos blocos e / ou outros elementos pode ser modificado, combinado, re-arrumado, aumentado, omitido, etc. em qualquer maneira para implementar um ou mais sistemas, métodos, dispositivos, mídia, aparelhos, arranjos, etc. para decodificação iterativa de blocos com controle de redundância cíclica.

[00109] Embora múltiplas modalidades da presente invenção foram ilustradas nos Desenhos anexos e descritas na discussão da Descrição Detalhada precedente, deve ser entendido que a invenção não é limitada às modalidades divulgadas, para isto é também capaz de numerosos re-arranjos, modificações e substituições sem fugir do escopo da invenção conforme estabelecido e definido pelas seguintes reivindicações.

REIVINDICAÇÕES

1. Método para decodificar um bloco de transporte construído a partir de sub-blocos múltiplos, cada sub-bloco sendo uma palavra de código que pode ser decodificada iterativamente com os bits de informação dela sendo cobertos por uma verificação de redundância cíclica, CRC, caracterizado pelo fato de compreender ações de:

receber o bloco de transporte;

realizar iterações de um processo de decodificação iterativa em um sub-bloco particular dos sub-blocos múltiplos;

verificar se a CRC para o sub-bloco particular verifica após cada iteração;

parar o processo de decodificação iterativa para o sub-bloco particular se a CRC associada com o sub-bloco particular verifica para um número pré-determinado de consecutivas iterações de decodificação; e

continuar o processo de decodificação iterativa para outros sub-blocos para os quais o processo de decodificação iterativa não foi parado até que as iterações de decodificação para todos os sub-blocos dos sub-blocos múltiplos possam ser paradas.

2. Método, de acordo com a reivindicação 1, caracterizado pelo fato de que o processo de decodificação iterativa é efetuado em um sub-bloco por vez antes de proceder para o seguinte sub-bloco dos sub-blocos múltiplos.

3. Método, de acordo com a reivindicação 1, caracterizado pelo fato de que as ações de verificar e parar são puladas até após um número pré-determinado mínimo de iterações do processo de decodificação iterativa ter sido efetuado.

4. Método, de acordo com a reivindicação 1, caracterizado pelo fato de que a ação de parar é pulada até após as ações de realizar e verificar terem sido implementadas um número pré-determinado mínimo de vezes.

5. Método, de acordo com a reivindicação 1, caracterizado pelo fato de que cada CRC de cada sub-bloco dos sub-blocos múltiplos compreende uma

CRC interna, e em que o bloco de transporte ainda inclui uma CRC externa cobrindo uma totalidade do bloco de transporte; o método ainda compreende ações de:

verificar se a CRC externa verifica para a totalidade do bloco de transporte; e

se a CRC externa é verificada para verificar, declarando a totalidade do bloco de transporte válido.

6. Método, de acordo com a reivindicação 1, caracterizado pelo fato de que o número pré-determinado de consecutivas iterações de decodificação é uma variável dependente que depende de um número de iterações de decodificação que foram efetuadas.

7. Método, de acordo com a reivindicação 1, ainda caracterizado pelo fato de compreender ações de:

priorizar alguns respectivos dos sub-blocos múltiplos com base nos respectivos valores de confiabilidade do sub-bloco; e

realizar o processo de decodificação iterativa nos sub-blocos múltiplos em uma ordem que é responsiva aos valores de confiabilidade do sub-bloco.

8. Dispositivo para decodificar um bloco de transporte construído a partir dos sub-blocos múltiplos, cada sub-bloco sendo uma palavra código que pode ser decodificada iterativamente com os bits de informação dela cobertos por uma verificação de redundância cíclica, CRC, caracterizado pelo fato de compreender:

um receptor para receber o bloco de transporte;

um decodificador para realizar iterações de um processo de decodificação iterativa em um sub-bloco particular dos sub-blocos múltiplos;

pelo menos um processador programado para verificar se a CRC para o sub-bloco particular verifica após cada iteração; o processador programado para parar o processo de decodificação iterativa para o sub-bloco particular se a CRC associada com o sub-bloco particular verifica para um número pré-determinado de consecutivas iterações de decodificação e para continuar o processo de decodificação iterativa para outros sub-blocos para os quais o processo de

decodificação iterativa não é parado até que as iterações de decodificação para todos os sub-blocos dos sub-blocos múltiplos possam ser paradas.

9. Dispositivo, de acordo com a reivindicação 8, caracterizado pelo fato de que o processador é ainda programado para fazer com que o decodificador realize o processo de decodificação iterativa de um sub-bloco por vez antes de proceder para o seguinte sub-bloco dos sub-blocos múltiplos.

10. Dispositivo, de acordo com a reivindicação 8, caracterizado pelo fato de que o número pré-determinado de consecutivas iterações de decodificação compreende um número pré-determinado fixo.

11. Dispositivo, de acordo com a reivindicação 8, caracterizado pelo fato de que o número pré-determinado de consecutivas iterações de decodificação compreende um número variável que é configurado responsivo a uma pré-determinada função que depende de um número das iterações de decodificação que foram efetuadas no sub-bloco particular.

12. Dispositivo, de acordo com a reivindicação 8, caracterizado pelo fato de que o processador é ainda programado para priorizar os sub-blocos múltiplos com base em valores de confiabilidade de sub-bloco e para fazer com que o decodificador realize o processo de decodificação iterativa nos diferentes sub-blocos em uma ordem que seja responsiva aos valores de confiabilidade do sub-bloco.

13. Dispositivo, de acordo com a reivindicação 12, caracterizado pelo fato de que os valores de confiabilidade do sub-bloco são com base em: estimativas de variância de ruído, relações de sinal-ruído, relações de sinal-ruído e interferência, estimativas de potência de ruído e interferência, pelo menos uma métrica de decisões ponderadas, ou valores de probabilidade através de bits decodificados.

14. Instruções executáveis por processador de armazenamento em memória para decodificar um bloco de transporte recebido construído a partir dos sub-blocos múltiplos, cada sub-bloco sendo uma palavra de código que pode ser decodificada iterativamente com os bits de informação dela sendo coberto por uma verificação de redundância cíclica, CRC; caracterizadas pelo fato de que, quando

executadas, direcionam um dispositivo a realizar ações compreendendo:

receber o bloco de transporte;

realizar iterações de um processo de decodificação iterativa em um sub-bloco particular dos sub-blocos múltiplos;

verificar se a CRC para o sub-bloco particular verifica após cada iteração;

parar o processo de decodificação iterativa para o sub-bloco particular se a CRC associada com o sub-bloco particular verifica para um número pré-determinado de consecutivas iterações de decodificação; e

continuar o processo de decodificação iterativa para outros sub-blocos para os quais o processo de decodificação iterativa não foi parado até que as iterações de decodificação para todos os sub-blocos dos sub-blocos múltiplos possam ser paradas.

15. Método para decodificar um bloco de transporte incluindo sub-blocos múltiplos, cada um dos quais pode ser decodificado iterativamente, onde o bloco de transporte também inclui uma única verificação de redundância cíclica, CRC, sobre uma totalidade do bloco de transporte; caracterizado pelo fato de compreender ações de:

realizar uma iteração de decodificação em um sub-bloco particular dos sub-blocos múltiplos;

verificar a CRC para a totalidade do bloco de transporte usando bits decodificados obtidos a partir da iteração de decodificação no sub-bloco particular e bits decodificados obtidos a partir de uma mais recente iteração de decodificação em outros sub-blocos dos sub-blocos múltiplos; e

realizar uma iteração de decodificação em um diferente sub-bloco se a CRC não é verificada;

parar o processo de decodificação iterativa para o sub-bloco particular se a CRC associada com o sub-bloco particular verifica para um número pré-determinado de consecutivas iterações de decodificação; e

continuar o processo de decodificação iterativa para outros sub-

blocos para os quais o processo de decodificação iterativa não foi parado até que as iterações de decodificação para todos os sub-blocos dos sub-blocos múltiplos possam ser paradas.

16. Método para decodificar um bloco de transporte incluindo sub-blocos múltiplos, cada um dos quais pode ser decodificado iterativamente, em que o bloco de transporte também inclui uma única verificação de redundância cíclica, CRC, sobre uma totalidade dos blocos de transporte; caracterizado pelo fato de compreender ações de:

realizar uma iteração de decodificação em um sub-bloco particular dos sub-blocos múltiplos;

verificar a CRC para a totalidade do bloco de transporte usando bits decodificados obtidos a partir da iteração de decodificação no sub-bloco particular e bits decodificados obtidos usando uma mais recente iteração de decodificação nos outros sub-blocos dos sub-blocos múltiplos; e

terminar as iterações de decodificação para todos dos sub-blocos dos sub-blocos múltiplos se a CRC verifica consecutivamente para um número pré-determinado de consecutivas iterações de decodificação;

parar o processo de decodificação iterativa para o sub-bloco particular se a CRC associada com o sub-bloco particular verifica para um número pré-determinado de consecutivas iterações de decodificação; e

continuar o processo de decodificação iterativa para outros sub-blocos para os quais o processo de decodificação iterativa não foi parado até que as iterações de decodificação para todos os sub-blocos dos sub-blocos múltiplos possam ser paradas.

17. Método, de acordo com a reivindicação 16, caracterizado pelo fato de que as consecutivas iterações de decodificação podem ser nos diferentes sub-blocos dos sub-blocos múltiplos.

18. Método, de acordo com a reivindicação 16, caracterizado pelo fato de que as ações de verificação e terminação são puladas até após um número pré-determinado mínimo das iterações de decodificação ter sido efetuado no sub-

bloco particular.

19. Método, de acordo com a reivindicação 16, caracterizado pelo fato de que a ação de terminar é pulada até após as ações de realizar e verificar terem sido implementadas um número pré-determinado mínimo de vezes.

20. Método para decodificação iterativa de um bloco de transporte tendo uma verificação de redundância cíclica, CRC, caracterizado pelo fato de compreender ações de:

receber um bloco de transporte tendo sub-blocos múltiplos, o bloco de transporte incluindo uma CRC associada com os sub-blocos múltiplos;

realizar uma iteração de decodificação em um número de sub-blocos dos sub-blocos múltiplos;

verificar se a CRC verifica com relação aos sub-blocos múltiplos; e

realizar a iteração de decodificação em pelo menos um diferente sub-bloco dos sub-blocos múltiplos se a CRC não é verificada para verificar;

parar o processo de decodificação iterativa para o sub-bloco particular se a CRC associada com o sub-bloco particular verifica para um número pré-determinado de consecutivas iterações de decodificação; e

continuar o processo de decodificação iterativa para outros sub-blocos para os quais o processo de decodificação iterativa não foi parado até que as iterações de decodificação para todos os sub-blocos dos sub-blocos múltiplos possam ser paradas.

21. Método, de acordo com a reivindicação 20, ainda caracterizado pelo fato de compreender uma ação de:

realizar a iteração de decodificação em cada um dos sub-blocos dos sub-blocos múltiplos pelo menos uma vez antes de realizar a ação de verificar.

22. Método, de acordo com a reivindicação 20, caracterizado pelo fato de que o número de sub-blocos para a ação de realizar uma iteração de decodificação em um número de sub-blocos iguais a um.

23. Método, de acordo com a reivindicação 20, caracterizado ainda pelo fato de compreender uma ação de:

se a CRC é verificada para verificar, declarar o bloco de transporte válido.

24. Método, de acordo com a reivindicação 20, caracterizado ainda pelo fato de compreender uma ação de:

se a CRC é verificada para verificar, repetidamente realizar as iterações de decodificação em um ou mais sub-blocos até um histórico de verificação de CRC ser determinado como satisfazendo um critério pré-determinado.

25. Método, de acordo com a reivindicação 24, caracterizado pelo fato de que:

o histórico de verificação de CRC é determinado para satisfazer o critério pré-determinado se a CRC verifica um número pré-determinado total de vezes; ou

o histórico de verificação de CRC é determinado para satisfazer o critério pré-determinado se a CRC verifica um pré-determinado consecutivo número de vezes.

26. Método, de acordo com a reivindicação 24, ainda caracterizado pelo fato de compreender uma ação de:

declarar o bloco de transporte inválido se o histórico de verificação de CRC não é determinado como satisfazendo o critério pré-determinado.

27. Método, de acordo com a reivindicação 20, ainda caracterizado pelo fato de compreender uma ação de:

declarar o bloco de transporte inválido se um número pré-determinado de iterações de decodificação são efetuadas em cada sub-bloco dos sub-blocos múltiplos sem a verificação de CRC.

28. Dispositivo para decodificação iterativa de um bloco de transporte tendo uma verificação de redundância cíclica, CRC, caracterizado pelo fato de compreender:

um receptor para receber um bloco de transporte tendo sub-blocos múltiplos, o bloco de transporte incluindo uma CRC associada com os sub-blocos múltiplos;

um decodificador para decodificar um subconjunto dos sub-blocos múltiplos; e

pelo menos um processador programado para verificar se a CRC verifica com relação aos sub-blocos múltiplos após o subconjunto ser decodificado pelo decodificador, o processador programado para fazer com que o decodificador continue a decodificar iterativamente outros subconjuntos dos sub-blocos múltiplos se a CRC não é verificada para verificar após cada um dos subconjuntos;

o processador programado ainda para parar o processo de decodificação iterativa para o sub-bloco particular se a CRC associada com o sub-bloco particular verifica para um número pré-determinado de consecutivas iterações de decodificação; e

o processador programado ainda para continuar o processo de decodificação iterativa para outros sub-blocos para os quais o processo de decodificação iterativa não foi parado até que as iterações de decodificação para todos os sub-blocos dos sub-blocos múltiplos possam ser paradas.

29. Dispositivo, de acordo com a reivindicação 28, caracterizado pelo fato de que o decodificador compreende um decodificador turbo que efetua duas operações de decodificação por ciclo de decodificação.

30. Dispositivo, de acordo com a reivindicação 29, caracterizado pelo fato de que cada iteração do decodificador turbo compreende uma operação de decodificação ou duas operações de decodificação.

31. Dispositivo, de acordo com a reivindicação 28, caracterizado pelo fato de que o processador é ainda programado (i) para priorizar respectivos alguns dos sub-blocos múltiplos com base nos respectivos valores de confiabilidade do sub-bloco e (ii) para fazer com que o decodificador decodifique os subconjuntos dos sub-blocos múltiplos responsivos à priorização do valor de confiabilidade do sub-bloco.

32. Dispositivo, de acordo com a reivindicação 28, caracterizado pelo fato de que o processador é ainda programado para verificar se a CRC verifica com relação aos sub-blocos múltiplos verificando os bits decodificados que são

concatenados a partir de uma iteração de decodificação corrente no subconjunto dos sub-blocos múltiplos e a partir de uma mais recente iteração de decodificação para um ou mais outros sub-blocos dos sub-blocos múltiplos.

33. Dispositivo, de acordo com a reivindicação 28, caracterizado pelo fato de que o processador é ainda programado para repetidamente decodificar subconjuntos dos sub-blocos múltiplos até um histórico de verificação de CRC ser determinado como satisfazendo um critério pré-determinado.

34. Dispositivo, de acordo com a reivindicação 33, caracterizado pelo fato de que:

o histórico de verificação de CRC é determinado como satisfazendo o critério pré-determinado se a CRC verifica um número pré-determinado total de vezes; ou

o histórico de verificação de CRC é determinado como satisfazendo o critério pré-determinado se a CRC verifica um pré-determinado consecutivo número de vezes.

35. Instruções executáveis por processador de armazenamento em memória para decodificação iterativa de um bloco de transporte tendo uma verificação de redundância cíclica, CRC, caracterizadas pelo fato de que, quando executadas, direcionam um dispositivo para realizar ações compreendendo:

receber um bloco de transporte tendo sub-blocos múltiplos, o bloco de transporte incluindo uma CRC associada com os sub-blocos múltiplos, os sub-blocos múltiplos numerados "n";

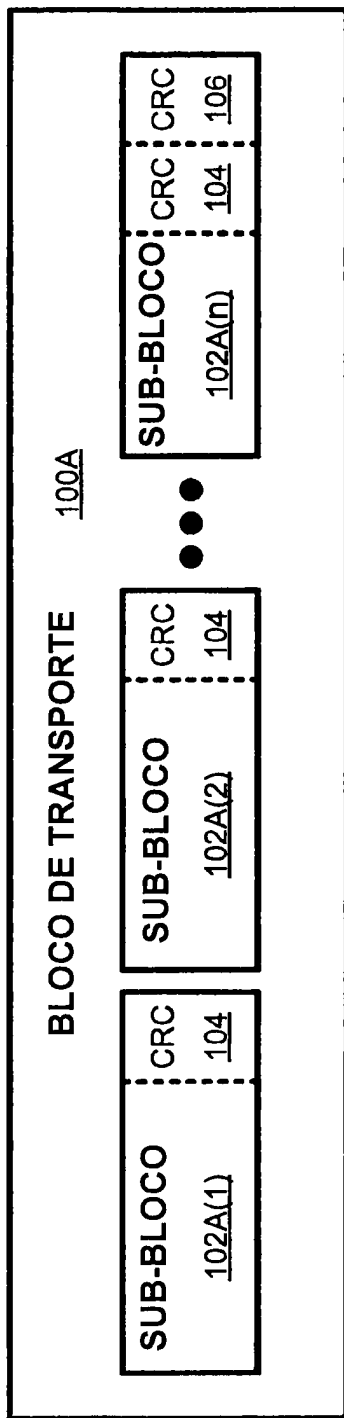
realizar a iteração de decodificação em um número de sub-blocos dos sub-blocos múltiplos, o número sendo maior do que zero e menor do que "n";

verificar se a CRC verifica com relação aos sub-blocos múltiplos;

realizar a iteração de decodificação em pelo menos um diferente sub-bloco dos sub-blocos múltiplos se a CRC não é verificada para verificar;

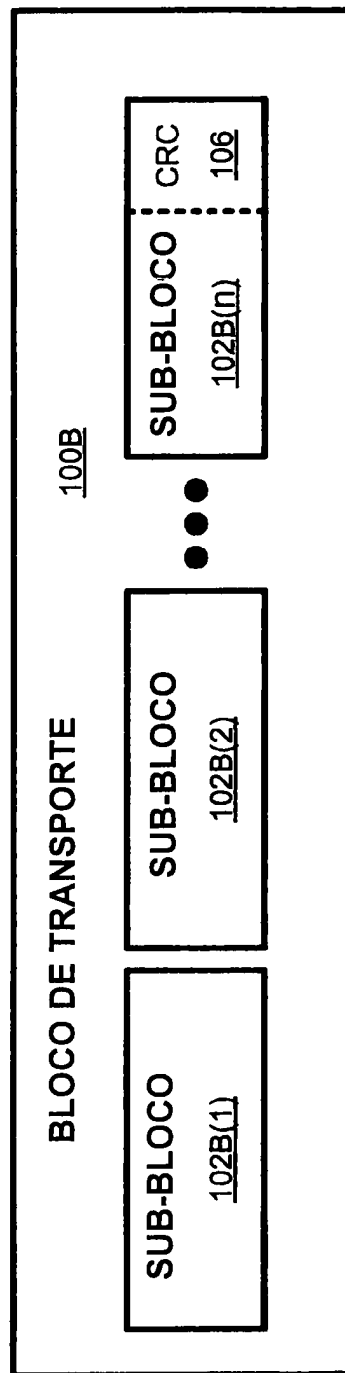
parar o processo de decodificação iterativa para o sub-bloco particular se a CRC associada com o sub-bloco particular verifica para um número pré-determinado de consecutivas iterações de decodificação; e

continuar o processo de decodificação iterativa para outros sub-blocos para os quais o processo de decodificação iterativa não foi parado até que as iterações de decodificação para todos os sub-blocos dos sub-blocos múltiplos possam ser paradas.



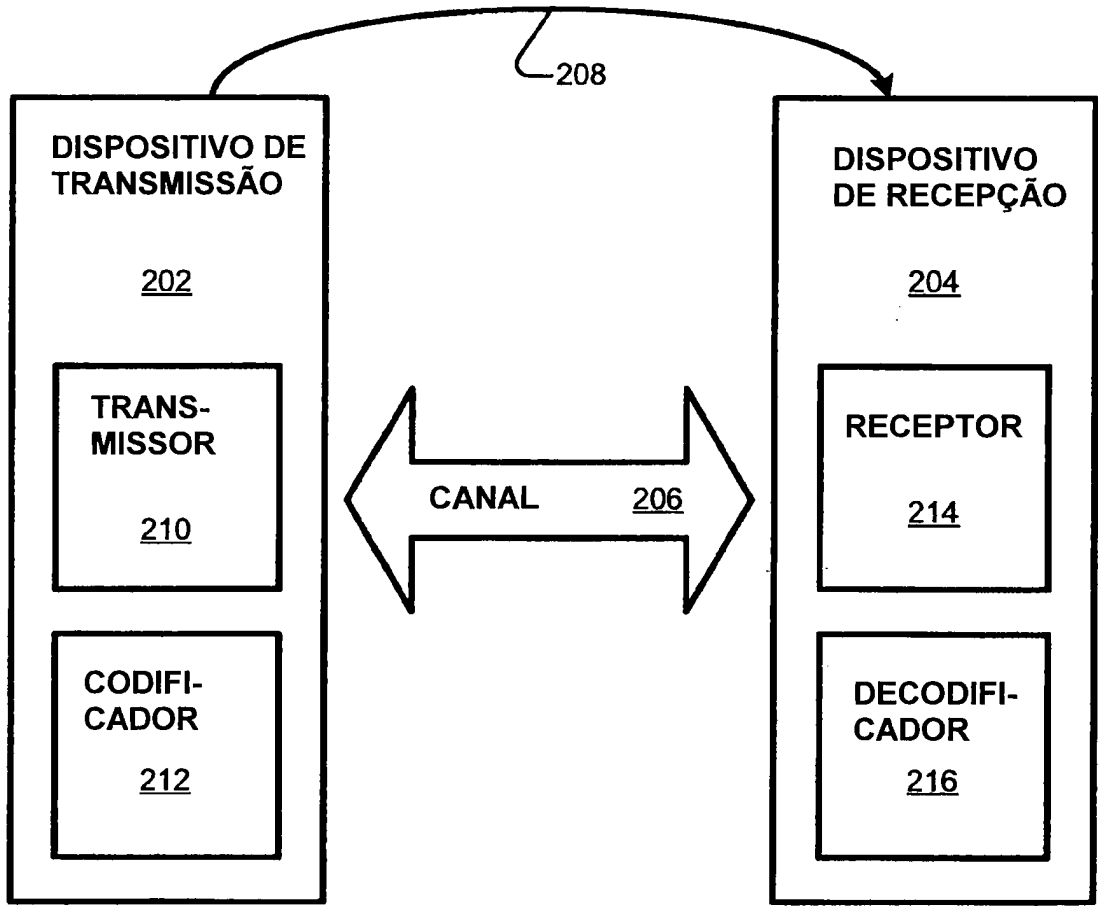
TÉCNICA ANTERIOR

FIG. 1A



TÉCNICA ANTERIOR

FIG. 1B



200 →

FIG. 2

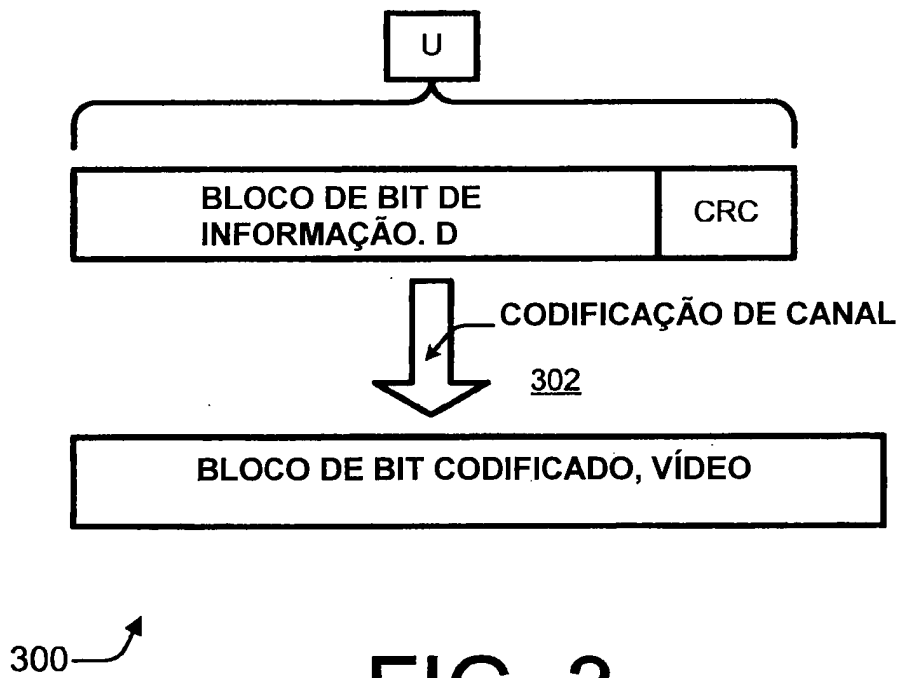
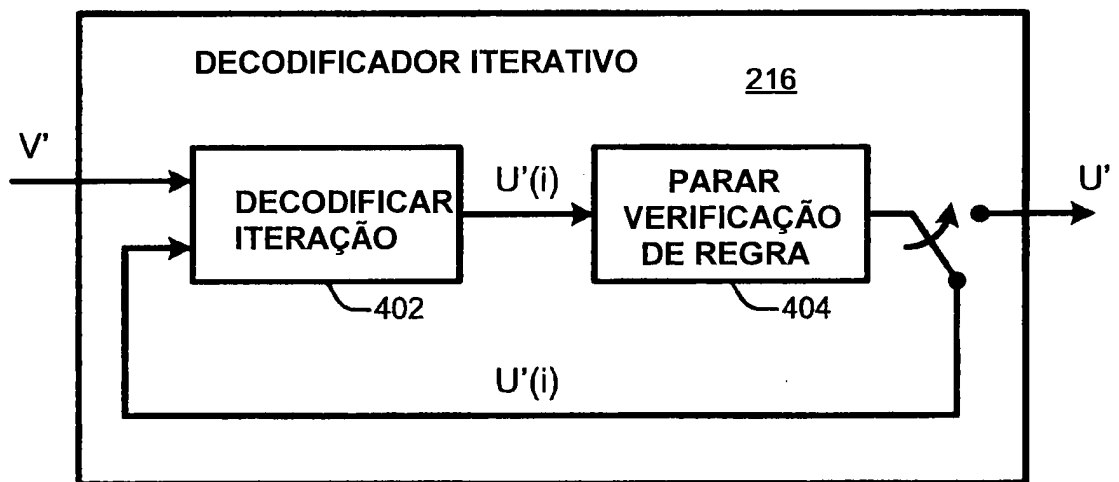


FIG. 3



400

FIG. 4

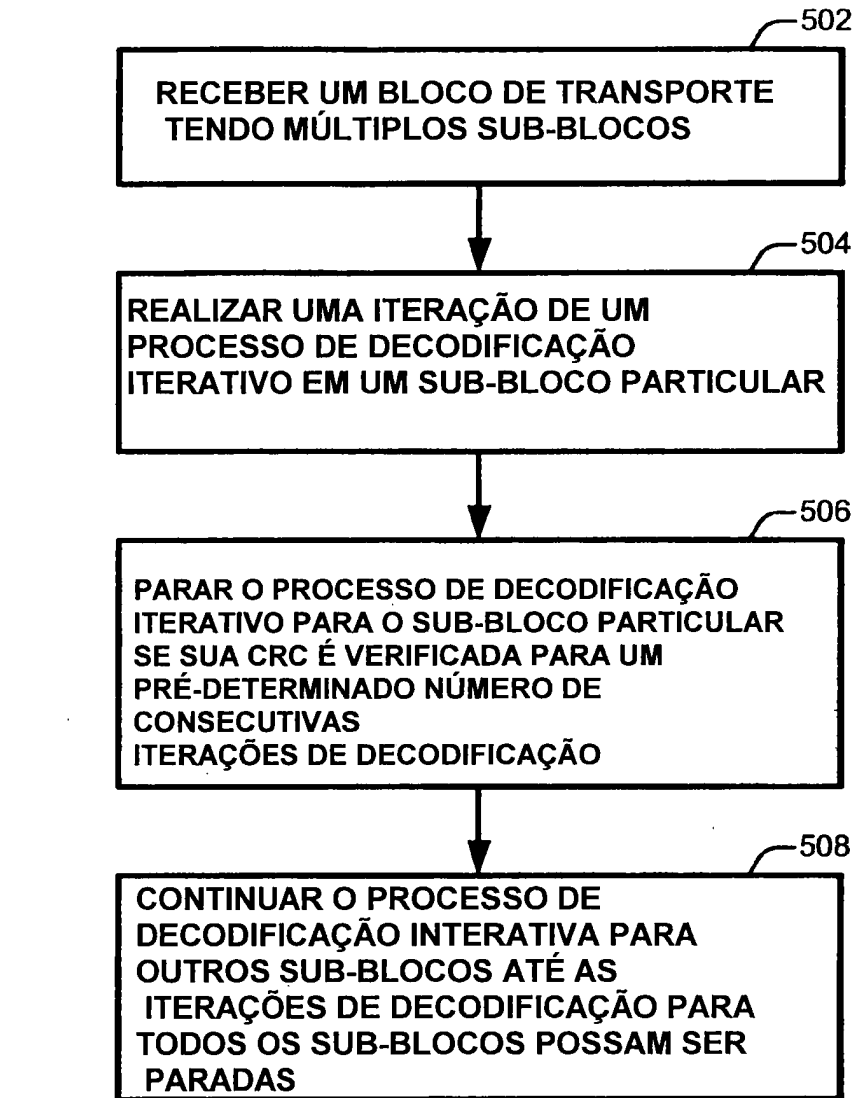
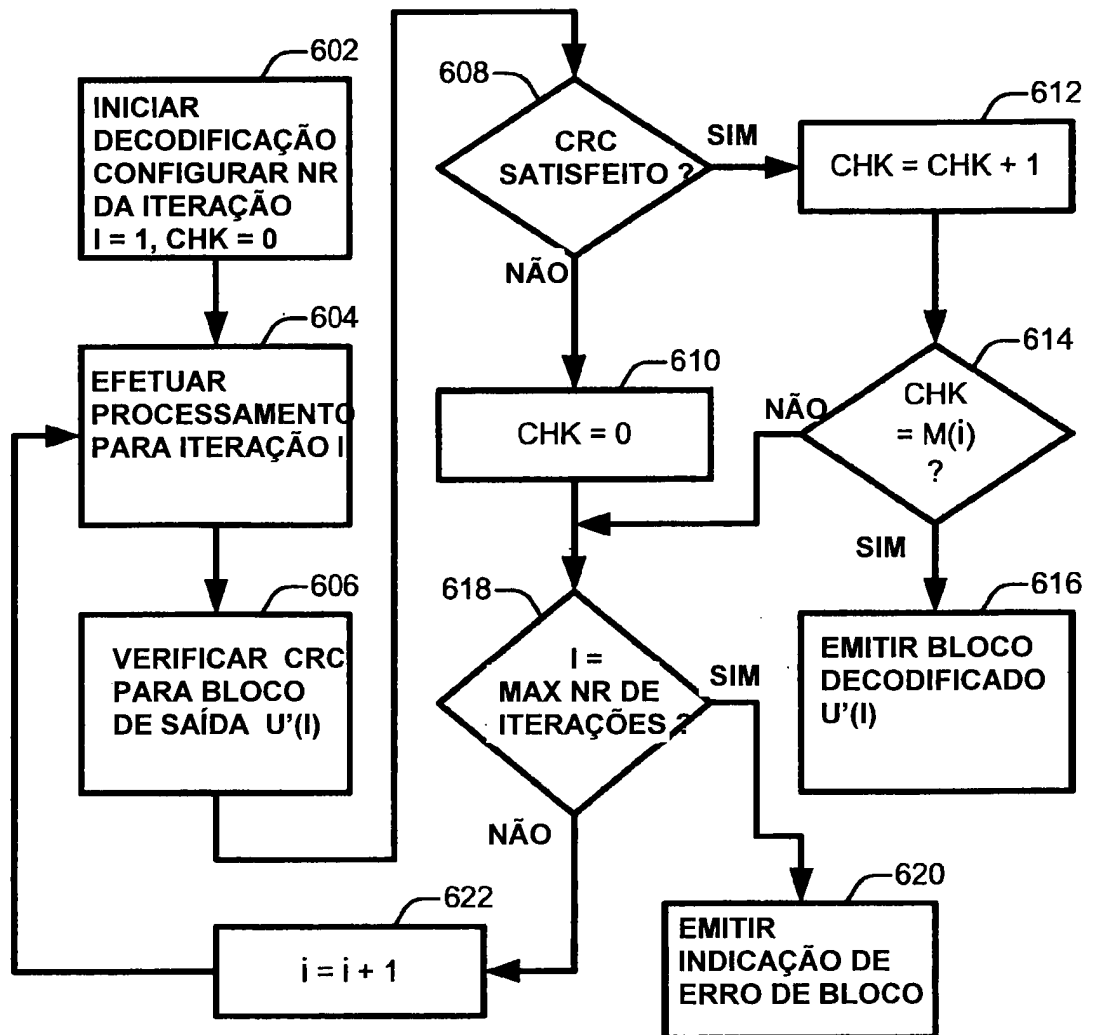
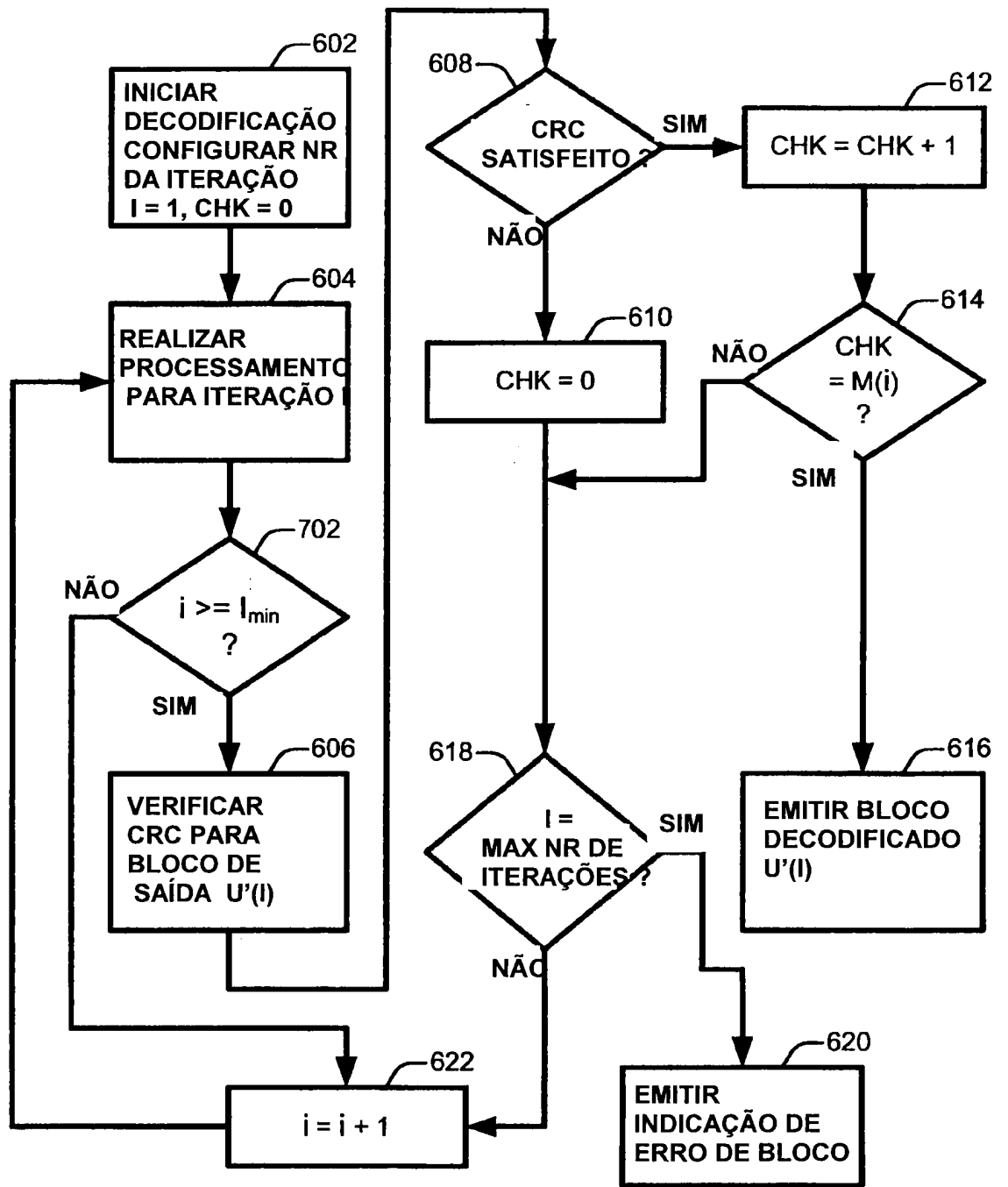


FIG. 5



600 ↗

FIG. 6



700 ↗

FIG. 7

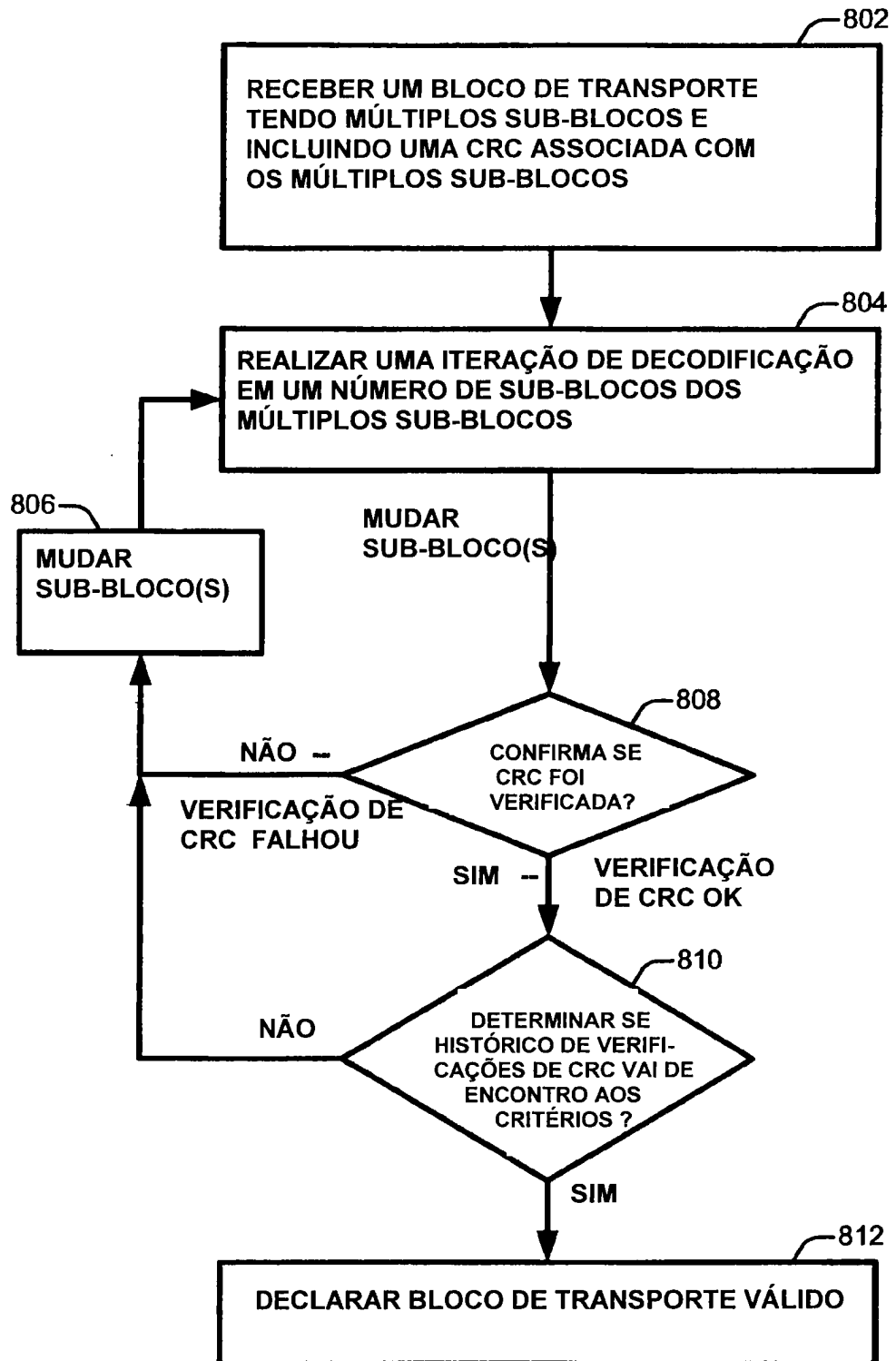
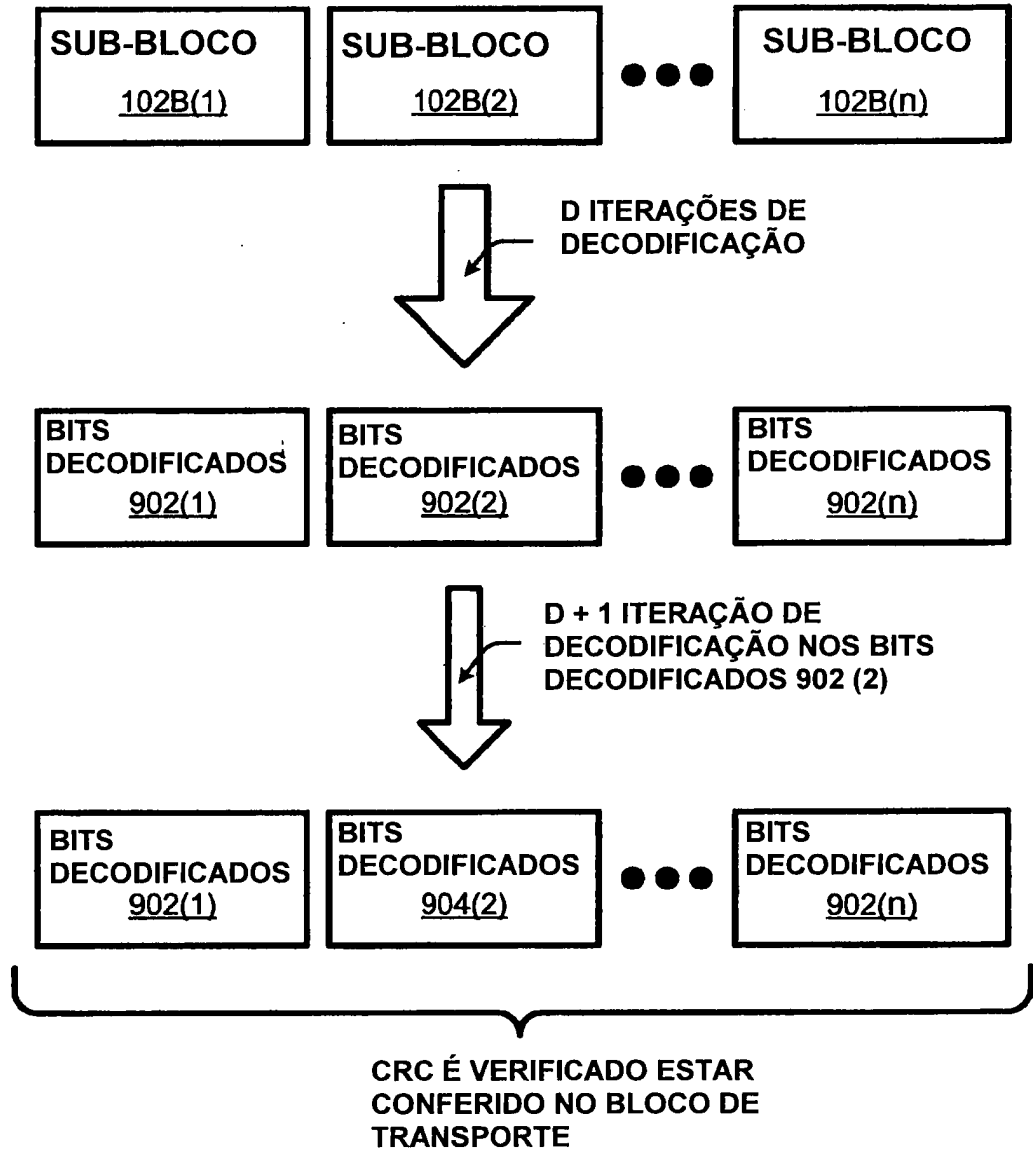
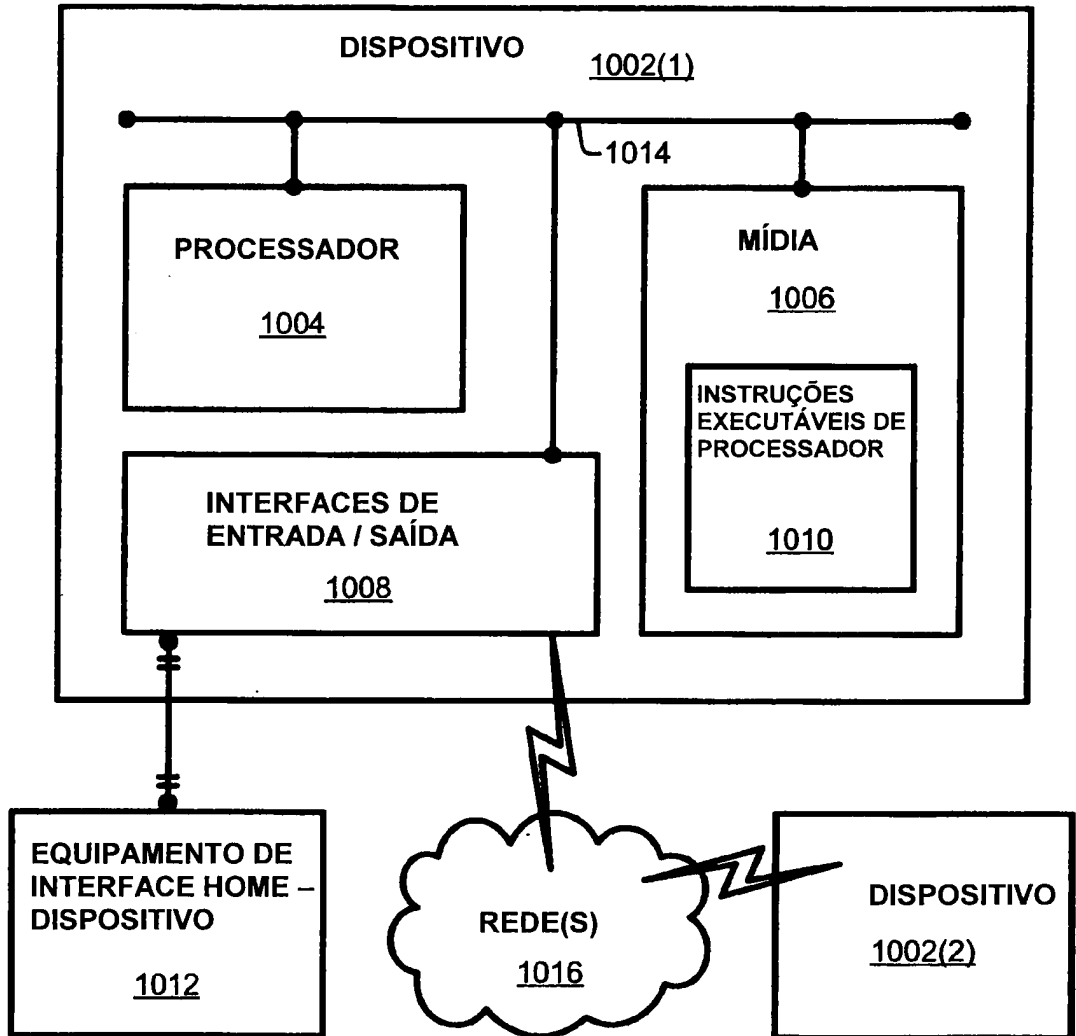


FIG. 8



900 ↗

FIG. 9



1000 ↗

FIG. 10