

(12) FASCÍCULO DE PATENTE DE INVENÇÃO

(22) Data de pedido: 2009.06.03	(73) Titular(es): LG ELECTRONICS INC. 20 YOIDO-DONG, YEONGDUNGPO-GU SEOUL 150-721 KR
(30) Prioridade(s): 2009.01.29 US 148357 P	
(43) Data de publicação do pedido: 2010.08.04	
(45) Data e BPI da concessão: 2012.09.12 247/2012	(72) Inventor(es): WOO SUK KO KR SANG CHUL MOON KR HO TAEK HONG KR
	(74) Mandatário: ALBERTO HERMÍNIO MANIQUE CANELAS RUA VÍCTOR CORDON, 14 1249-103 LISBOA PT

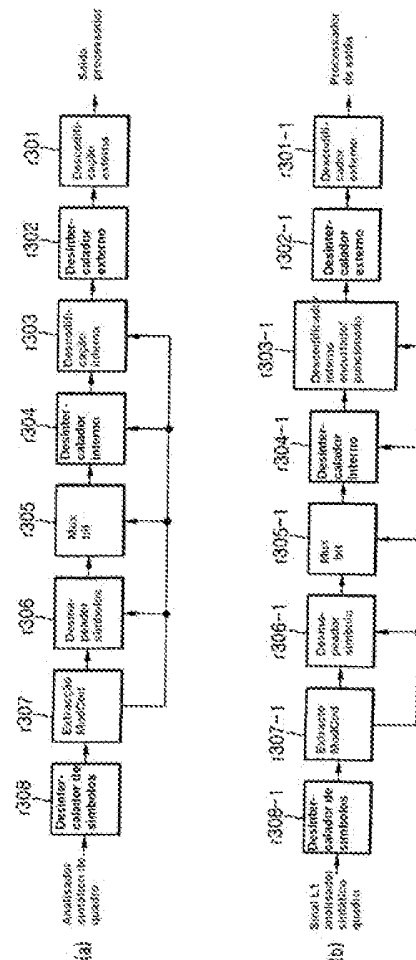
(54) Epígrafe: **DISPOSITIVO E PROCESSO PARA A TRANSMISSÃO E RECEPÇÃO DE UM SINAL**

(57) Resumo:

A PRESENTE INVENÇÃO REFERE-SE A UM PROCESSO PARA A TRANSMISSÃO E RECEPÇÃO DE SINAIS E A UM DISPOSITIVO CORRESPONDENTE. UM ASPECTO DA PRESENTE INVENÇÃO REFERE-SE A UMA SINALIZAÇÃO EFICIENTE DA CAMADA 1 (L1) E A UM RECEPTOR QUE UTILIZA A SINALIZAÇÃO L1 EFICIENTE PARA UMA RADIODIFUSÃO POR CABO EFICIENTE.

RESUMO**"DISPOSITIVO E PROCESSO PARA A TRANSMISSÃO E RECEPÇÃO DE UM SINAL"**

Fig. 54



A presente invenção refere-se a um processo para a transmissão e recepção de sinais e a um dispositivo correspondente. Um aspecto da presente invenção refere-se a uma sinalização eficiente da camada 1 (L1) e a um receptor que utiliza a sinalização L1 eficiente para uma radiodifusão por cabo eficiente.

DESCRIÇÃO**"DISPOSITIVO E PROCESSO PARA A TRANSMISSÃO E RECEPÇÃO DE UM
SINAL"****Antecedentes da invenção****Campo da invenção**

A presente invenção refere-se a um processo para a transmissão e recepção de um sinal e a um dispositivo para a transmissão e recepção de um sinal, e mais particularmente, a um processo para a transmissão e recepção de um sinal e a um dispositivo para transmissão e recepção de um sinal, que são passíveis de melhorar a eficiência da transmissão de dados.

Descrição da técnica relacionada

Com o desenvolvimento da tecnologia de transmissão digital, os utilizadores passaram a receber uma imagem em movimento de alta definição (*HD*). Com o desenvolvimento contínuo de um algoritmo de compressão e alto desempenho de hardware, será proporcionado aos utilizadores um melhor ambiente no futuro. Um sistema de televisão digital (*DTV*) pode receber um sinal de radiodifusão digital e fornecer uma variedade de serviços

adicionais aos utilizadores, bem como um sinal de vídeo e um sinal de áudio. Os documentos DVB ORGANIZATION "en_302xxxv010101rev3.3.doc" e "tm3980.DVB-T2.V0.5.5" descrevem processos da técnica anterior para a radiodifusão de sinais de vídeo digital.

A radiodifusão de vídeo digital (*Digital Video Broadcasting (DVB) - C2*) é a terceira especificação a unir-se à família *DVB* de sistemas de transmissão de segunda geração. Desenvolvida em 1994, a *DVB-C* actual encontra-se implantada em mais de 50 milhões de sintonizadores de cabo em todo o mundo. Em consonância com os outros sistemas de segunda geração *DVB*, o *DVB-C2* utiliza uma combinação de códigos de baixa densidade com controlo de paridade (*Low-density parity-check - LDPC*) e *BCH*. Este potente código de correcção de erros (*Forward Error correction - FEC*) proporciona cerca de 5 dB de melhoramento de rácio de portadora-ruído em relação a *DVB-C*. Esquemas de intercalação de bits apropriados optimizam a robustez global do sistema *FEC*. Prolongado por um cabeçalho, esses quadros são denominados de condutas de camada física (*Physical Layer Pipes - PLP*). Uma ou mais dessas *PLPs* são multiplexadas transformando-se numa fatia de dados. Dois intercalamentos dimensionais (nos domínios do tempo e da frequência) são aplicados a cada parcela permitindo ao receptor eliminar o impacto das deficiências das rajadas e da interferência selectiva da frequência tal como entrada de uma única frequência.

Com o desenvolvimento dessas tecnologias de radiodifusão digital, aumentou a exigência por um serviço, tal como um sinal de vídeo e um sinal de áudio e a dimensão dos dados desejados pelos utilizadores ou o número de canais de transmissão aumentou gradualmente.

Sumário da invenção

Sendo assim, a presente invenção refere-se a um processo para a transmissão e recepção de um sinal e a um dispositivo para a transmissão e recepção de um sinal que substancialmente evita um ou mais problemas devido às limitações e desvantagens da técnica relacionada.

Um objectivo da presente invenção é o de proporcionar um processo para a transmissão e recepção de um sinal e um dispositivo para a transmissão e recepção de um sinal, que são passíveis de melhorar a eficiência da transmissão de dados.

Um outro objectivo da presente invenção é o de proporcionar um processo para a transmissão e recepção de um sinal e um dispositivo para a transmissão e recepção de um sinal, que são passíveis de melhorar a capacidade de correcção de erro de bits que configuram um serviço.

As vantagens adicionais, objectos e características da invenção serão apresentados em parte na descrição que se segue e em parte se tornarão evidentes

para o técnico após análise do seguinte. Os objectivos e outras vantagens da invenção podem ser realizados e alcançados pela estrutura especialmente assinalada na descrição e reivindicações da mesma bem como nos desenhos anexos.

Breve descrição dos desenhos

Os desenhos anexos, que se encontram incluídos para proporcionar uma maior compreensão da invenção e se encontram incorporados e constituem uma parte deste pedido, ilustram forma(s) de realização da invenção e juntamente com a descrição servem para explicar o princípio da invenção. As figuras representam:

Figura 1 exemplo de modulação de amplitude em quadratura 64 (QAM) utilizada no DVB-T europeu.

Figura 2 processo do código binário Gray reflectido (BRGC).

Figura 3 saída próxima de gaussiana modificando a 64-QAM utilizada em DVB-T.

Figura 4 distância de Hamming entre o par reflectido no BRGC.

Figura 5 características em QAM, onde existe par reflectido para cada eixo I e eixo Q.

Figura 6 processo para modificar a QAM utilizando o par reflectido do BRGC.

Figura 7 exemplo de 64/256/1024/4096-QAM modificado.

Figuras

8-9 exemplo de 64-QAM modificada utilizando o par
reflectido do BRGC.

Figuras

10-11 exemplo de 256-QAM modificada utilizando o par
reflectido do BRGC.

Figuras

12-13 exemplo de 1024-QAM modificada utilizando o par
reflectido do BRGC(0~511).

Figuras

14-15 exemplo de 1024-QAM modificada utilizando o par
reflectido do BRGC(512~1023).

Figuras

16-17 exemplo de 4096-QAM modificada utilizando o par
reflectido do BRGC(0~511).

Figuras

18-19 exemplo de 4096-QAM modificada utilizando o par
reflectido do BRGC(512~1023).

Figuras

20-21 exemplo de 4096-QAM modificada utilizando o par
reflectido do BRGC (1024~1535).

Figuras

22-23 exemplo de 4096-QAM modificada utilizando o par
reflectido do BRGC (1536~2047).

Figuras

24-25 exemplo de 4096-QAM modificada utilizando o par
reflectido do BRGC(2048~2559).

Figuras

26-27 exemplo de 4096-QAM modificada utilizando o par
reflectido do BRGC(2560~3071).

Figuras

28-29 exemplo de 4096-QAM modificada utilizando o par reflectido do BRGC (3072~3583).

Figuras

30-31 exemplo de 4096-QAM modificada utilizando o par reflectido do BRGC(3584~4095).

Figura 32 exemplo do mapeamento de bits da QAM modificada onde 256-QAM é modificada utilizando o BRGC.

Figura 33 exemplo de transformação de MQAM numa constelação não uniforme.

Figura 34 exemplo de sistema de transmissão digital.

Figura 35 exemplo de um processador de entrada.

Figura 36 informação que pode ser incluída na banda de base (BB).

Figura 37 exemplo de BICM.

Figura 38 exemplo de codificador encurtado/puncionado.

Figura 39 exemplo de aplicação de várias constelações.

Figura 40 outro exemplo de casos onde é considerada a compatibilidade entre os sistemas convencionais.

Figura 41 estrutura de quadro que compreende o preâmbulo para a sinalização *L1* e símbolo de dados para dados *PLP*.

Figura 42 exemplo de construtor de quadros.

Figura 43 exemplo de inserção piloto (404) apresentada na figura 4.

Figura 44 estrutura de *SP*.

Figura 45 nova estrutura *SP* ou padrão piloto (*PP*) 5'.

Figura 46 estrutura *PP5'* sugerida.

Figura 47 relacionamento entre símbolo de dados e preâmbulo.

Figura 48 outro relacionamento entre símbolo de dados e preâmbulo.

Figura 49 exemplo de perfil de atrasos de canal por cabo.

Figura 50 estrutura piloto dispersa que utiliza $z=56$ e $z=112$.

Figura 51 exemplo de modulador baseado em *OFDM*.

Figura 52 exemplo de estrutura de preâmbulo.

Figura 53 exemplo de descodificação do preâmbulo.

Figura 54 processo para conceber preâmbulo mais otimizado.

Figura 55 outro exemplo de estrutura de preâmbulo

Figura 56 outro exemplo de descodificação de preâmbulo.

Figura 57 exemplo de estrutura de preâmbulo.

Figura 58 exemplo de descodificação *L1*.

Figura 59 exemplo de processador analógico.

Figura 60 exemplo de sistema de recepção digital.

Figura 61 exemplo de processador analógico utilizado no receptor.

Figura 62 exemplo de desmodulador.

Figura 63 exemplo de analisador sintáctico de quadros.

Figura 64 exemplo de desmodulador *BICM*.

Figura 65 exemplo de descodificação *LDPC* utilizando encurtamento / punção.

Figura 66 exemplo de processador de saída.

Figura 67 exemplo de taxa de repetição de bloco *L1* de 8 MHz.

Figura 68 exemplo de taxa de repetição de bloco *L1* de 8 MHz.

Figura 69 nova taxa de repetição de bloco *L1* de 7.61 MHz.

Figura 70 exemplo de sinalização *L1* que é transmitida no cabeçalho do quadro.

Figura 71 resultado do preâmbulo e simulação da estrutura *L1*.

Figura 72 exemplo de intercalador de símbolos.

Figura 73 exemplo de uma transmissão de bloco *L1*.

Figura 74 outro exemplo de sinalização *L1* transmitida dentro de um cabeçalho de quadro.

Figura 75 exemplo de intercalamento/desintercalamento de frequência ou no tempo.

Figura 76 quadro analisando a informação complementar da sinalização *L1*, que é transmitida no cabeçalho *FECFRAME* no módulo de inserção de cabeçalho *ModCod* (307) no trajecto de dados do módulo *BICM* apresentado na figura 3.

Figura 77 estrutura para cabeçalho *FECFRAME* para minimizar a informação complementar.

Figura 78 desempenho de uma taxa de erro nos bits (*BER*) da protecção *L1* acima mencionada.

Figura 79 exemplos de um quadro de transmissão e estrutura de quadro *FEC*.

Figura 80 exemplo de sinalização *L1*.

Figura 81 exemplo de pré-sinalização *L1*.

Figura 82 estrutura do bloco de sinalização *L1*.

Figura 83 intercalação no tempo *L1*.

Figura 84 exemplo de extracção de informação de modulação e código.

Figura 85 outro exemplo de pré-sinalização *L1*.

Figura 86 exemplo de agendamento de bloco de pré-sinalização *L1* que é transmitido no preâmbulo.

Figura 87 exemplo de pré-sinalização *L1* onde é considerado um reforço da potência.

Figura 88 exemplo de pré-sinalização *L1*.

Figura 89 outro exemplo de extracção de informação de modulação e código.

Figura 90 outro exemplo de extracção de informação de modulação e código.

Figura 91 exemplo de pré-sincronização *L1*.

Figura 92 exemplo de pré-sinalização *L1*.

Figura 93 exemplo de sinalização *L1*.

Figura 94 exemplo de trajecto de sinalização *L1*.

Figura 95 exemplo de sinalização *L1* transmitida dentro de um cabeçalho de quadro.

Figura 96 exemplo de sinalização *L1* transmitida dentro de um cabeçalho de quadro.

Figura 97 exemplo de sinalização *L1* transmitida dentro de um cabeçalho de quadro.

Figura 98 exemplo de sinalização *L1*.

Figura 99 exemplo de intercalador de símbolos.

Figura

100 desempenho de intercalação do intercalador no tempo da figura 99.

Figura

101 exemplo de intercalador de símbolos.

Figura

102 desempenho de intercalação do intercalador no tempo da figura 101.

Figura

103 exemplo de desintercalador de símbolos.

Figura

104 exemplo de intercalador no tempo.

Figura

105 resultado de intercalação utilizando o processo
mostrado na figura 104.

Figura

106 exemplo de processo de endereçamento da figura
105.

Figura

107 exemplo de intercalador no tempo $L1$.

Figura

108 exemplo de desintercalador de símbolos.

Figura

109 exemplo de desintercalador.

Figura

110 exemplo de desintercalador de símbolos

Figura

111 exemplo de endereços de linha e coluna para
desintercalação no tempo

Figura

112 exemplo de intercalação geral no bloco num
domínio de símbolo de dados onde os pilotos não
são utilizados.

Figura

113 exemplo de um transmissor *OFDM* que utiliza fatias
de dados.

Figura

114 exemplo de um receptor *OFDM* que utiliza fatias de
dados

Figura

115 exemplo de intercalador no tempo e exemplo de desintercalador no tempo.

Figura

116 exemplo de formação de símbolos *OFDM*.

Figura

117 exemplo de um intercalador no tempo (*TI*).

Figura

118 exemplo de um intercalador no tempo (*TI*).

Figura

119 exemplo de uma estrutura de preâmbulo de um transmissor e um exemplo de um processo num receptor.

Figura

120 exemplo de um processo num receptor para obter *L1_XFEC_FRAME* do preâmbulo.

Figura

121 exemplo de uma estrutura de preâmbulo num transmissor e um exemplo de um processo num receptor.

Figura

122 exemplo de um intercalador no tempo (*TI*).

Figura

123 exemplo de um transmissor *OFDM* que utiliza fatia de dados.

Figura

124 exemplo de um receptor *OFDM* que utiliza fatias de dados.

Figura

125 exemplo de intercalador no tempo (*Time Interleaver - TI*).

Figura

126 exemplo de um desintercalador no tempo (*Time De-Interleaver - TDI*).

Figura

127 exemplo de intercalador no tempo (*Time Interleaver - TI*).

Figura

128 exemplo de fluxo de intercalação e desintercalação no tempo de preâmbulos.

Figura

129 parâmetro de profundidade de intercalação no tempo na sinalização *L1* de cabeçalho.

Figura

130 exemplo de uma sinalização *L1* de cabeçalho, estrutura *L1*, e um processo de zona de preenchimento.

Figura

131 exemplo de sinalização *L1*.

Figura

132 exemplo de *dslice_ti_depth*.

Figura

133 exemplo de *dslice_type*.

Figura

134 exemplo de *Plp_type*.

Figura

135 exemplo de *Plp_payload_type*.

Figura

136 exemplo de *Plp_modcod*.

Figura

137 exemplo de *GI*.

Figura

138 exemplo de *PAPR*.

Figura

139 exemplo de sinalização *L1*.

Figura

140 exemplo de *plp_type*.

Figura

141 exemplo de sinalização *L1*.

Figura

142 exemplo de uma sinalização *L1* de cabeçalho, estrutura *L1*, e um processo de zona de preenchimento.

Descrição das formas de realização preferidas

Será feita agora referência pormenorizada às formas de realização preferidas da presente invenção, exemplos esses que se encontram ilustrados nos desenhos anexos. Sempre que possível, serão utilizados os mesmos números de referência em todos os desenhos para designar as mesmas ou peças semelhantes.

Na descrição a seguir, o termo "serviço" é indicativo de qualquer conteúdo difundido que pode ser transmitido/recebido pelo dispositivo de transmissão/recepção de sinal.

A modulação de amplitude por quadratura (*QAM*) utilizando código binário *Gray* reflectido (*BRGC*) é utilizada como modulação num ambiente de transmissão de radiodifusão onde é utilizada a modulação codificada de intercalamento de bits (*BICM*). A figura 1 mostra um exemplo de 64-*QAM* utilizada no *DVB-T* europeu.

O *BRGC* pode ser realizado utilizando o processo mostrado na figura 2. Um *BRGC* de n bits pode ser realizado pela adição de um código inverso de *BRGC* ($n-1$) bit (ou seja código reflectido) para uma traseira de ($n-1$) bit, adicionando 0s a uma frente de *BRGC* ($n-1$) bit original, e adicionando 1s a uma frente de código reflectido. O código *BRGC* feito por este processo tem uma distância *Hamming*, entre códigos contíguos, de um (1). Adicionalmente, quando *BRGC* é aplicado a *QAM*, a distância *Hamming* entre um ponto e os quatro pontos que se encontram mais próximas contíguas ao ponto, é um (1) e a distância *Hamming* entre o ponto e outros quatro pontos que são os segundos mais próximos contíguas ao ponto, é dois (2). Tais características de distância *Hamming* entre um ponto de constelação específico e outros pontos contíguos podem ser denominadas como regra de mapeamento *Gray* em *QAM*.

Para tornar um sistema robusto contra ruído aditivo gaussiano branco (*Additive White Gaussian Noise - AWGN*), a distribuição de sinais transmitidos de um transmissor pode ser feita perto da distribuição Gaussiana.

Para ser passível de fazer isso, podem ser modificadas as localizações dos pontos na constelação. A figura 3 mostra uma saída próxima da gaussiana modificando 64-QAM utilizado em DVB-T. Tal constelação pode ser denominada como QAM não uniforme (NU-QAM).

Para tornar uma constelação QAM não-uniforme, pode ser utilizada uma função gaussiana de distribuição cumulativa (*Cumulative Distribution Function - CDF*). No caso de 64, 256, ou 1024 QAM, i.e., 2^N AMs, a QAM pode ser dividida em duas N-PAM independentes. Ao dividir a CDF gaussiana em N secções de probabilidade idêntica e ao permitir a um ponto de sinal em cada secção que represente a secção, pode ser realizada uma constelação que apresenta a distribuição gaussiana. Por outras palavras, a coordenada x_j da N-APM não uniforme recém-definida, pode ser definida da seguinte forma:

$$\int_{-\infty}^{x_j} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx = p_j, \quad p_j \in \left\{ \frac{1}{2N}, \frac{3}{2N}, \dots, \frac{2N-1}{2N} \right\} \quad (\text{Eq. 1})$$

A figura 3 é um exemplo de transformação 64QAM de DVB-T em NU-64QAM utilizando os processos acima. A figura 3 representa o resultado da modificação das coordenadas de cada eixo I e eixo Q utilizando os processos acima e correspondendo os pontos de constelação anteriores às recém-definidas coordenadas. No caso de 32, 128, ou 512 QAM, i.e., QAM transversal, que não é 2^N QAM, ao modificar p_j apropriadamente, pode ser encontrada uma nova coordenada.

Uma forma de realização da presente invenção pode modificar *QAM* utilizando *BRGC* empregando as características de *BRGC*. Tal como apresentado na figura 4, a distância de *Hamming* entre o par reflectido em *BRGC* é um porque difere somente em um bit que é adicionado à frente de cada código. A figura 5 mostra as características em *QAM*, onde existe o par reflectido para cada eixo I e eixo Q. Nesta figura, existe o par reflectido em cada lado da linha preta pontilhada.

Ao utilizar pares reflectidos existentes na *QAM*, pode ser reduzida uma potência média de uma constelação *QAM* mantendo simultaneamente a regra de mapeamento *Gray* em *QAM*. Por outras palavras, numa constelação onde uma potência média é normalizada como 1, pode ser aumentada na constelação a distância euclidiana mínima. Quando esta *QAM* modificada é aplicada aos sistemas de transmissão ou de comunicação, é possível implementar um sistema mais robusto ao ruído utilizando a mesma energia que um sistema convencional ou um sistema com o mesmo desempenho que um sistema convencional, mas que consome menos energia.

A figura 6 mostra um processo para modificar *QAM* utilizando um par reflectido do *BRGC*. A figura 6a mostra uma constelação e a figura 6b mostra um fluxograma para modificar *QAM* utilizando um par reflectido do *BRGC*. Em primeiro lugar, é necessário ser encontrado entre os pontos da constelação um ponto alvo que tem a potência mais

elevada. Os pontos candidatos são os pontos onde esse ponto alvo pode mover-se e são os pontos contíguos mais próximos do par reflectido do ponto alvo. Depois tem que ser encontrado entre os pontos candidatos um ponto vazio (ou seja, um ponto que ainda não está tomado por outros pontos) que tiver a menor potência, sendo comparadas a potência do ponto-alvo e a potência de um ponto candidato. Se a potência do ponto candidato for menor, o ponto-alvo move-se o ponto candidato. Estes processos são repetidos até que uma potência média de pontos na constelação atinja um mínimo, mantendo a regra de mapeamento *Gray*.

A figura 7 mostra um exemplo de 64/256/1024/4096-QAM modificado. Os valores *Gray* mapeados, correspondem às figuras 8~31 respectivamente. Adicionalmente a estes exemplos, podem ser realizados outros tipos de QAM modificada que permitem que possa ser realizada a optimização idêntica da potência. Isto porque um ponto de destino pode mover-se para vários pontos candidatos. A QAM modificada sugerida pode ser aplicada a, não somente a 64/256/1024/4096-QAM, mas também a QAM transversal, uma QAM de tamanho maior, ou modulações utilizando outro *BRGC* diferente de QAM.

A figura 32 mostra um exemplo de mapeamento de bits de QAM modificada onde 256-QAM é modificada utilizando *BRGC*. A figura 32a e figura 32b mostram o mapeamento dos bits mais significativos (*MSB*). Os pontos designados como círculos cheios representam os mapeamentos de uns e os

pontos designados como círculos em branco representam os mapeamentos de zeros. De um mesmo modo, cada bit é correspondido tal como apresentado nas figuras de (a) a (h) na figura 32, até que os bits menos significativos (*Least Significant Bits - LSB*) estejam correspondidos. Tal como apresentado na figura 32, a QAM modificada pode habilitar a decisão de bit utilizando somente os eixos I e Q como QAM convencional, excepto para um bit que se encontra próximo de *MSB* (figura 32c e figura 32d). Ao utilizar estas características, pode ser realizado um receptor simples modificando parcialmente um receptor para QAM. Pode ser implementado um receptor eficiente verificando ambos os valores I e Q somente na determinação do bit próximo do *MSB* e calculando somente I ou Q para o resto dos bits. Este processo pode ser aplicado para *LLR* aproximado, *LLR* exacto, ou decisão difícil.

Ao utilizar o QAM modificado ou MQAM, que utiliza as características do BRGC acima, pode ser realizada a constelação não uniforme ou NU-MQAM. Na equação acima onde é utilizada a CDF gaussiana, P_j pode ser modificado para caber MQAM. Assim como a QAM, em MQAM, podem ser considerados dois PAMs que apresentam o eixo I e eixo Q. No entanto, ao contrário de QAM onde vários pontos que correspondem a um valor de cada eixo PAM são idênticos, o número de pontos altera-se na MQAM. Se um número de pontos que corresponde ao j° valor de PAM se encontra definido como n_j num MQAM onde existe um total de M pontos de constelação, então P_j pode ser definido tal como se segue:

$$\int_{-\infty}^{x_i} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx = p_i$$

$$p_i = \frac{\sum_{j=0}^{i-1} n_j + \frac{n_i}{2}}{M}, \quad n_0 = 0 \quad (\text{Eq. 2})$$

Ao utilizar o p_j recém-definido, a $MQAM$ pode ser transformada em constelação não uniforme. p_j pode ser definido tal como se segue para o exemplo de 256- $MQAM$.

$$p_j \in \left\{ \frac{2.5}{256}, \frac{10}{256}, \frac{22}{256}, \frac{36}{256}, \frac{51}{256}, \frac{67}{256}, \frac{84}{256}, \frac{102}{256}, \frac{119.5}{256}, \frac{136.5}{256}, \frac{154}{256}, \frac{172}{256}, \frac{189}{256}, \frac{205}{256}, \frac{220}{256}, \frac{234}{256}, \frac{246}{256}, \frac{253.5}{256} \right\}$$

A figura 33 é um exemplo de transformação de $MQAM$ numa constelação não uniforme. A $NU-MQAM$ realizada utilizando estes processos pode reter características dos receptores $MQAM$ com coordenadas modificadas de cada PAM . Deste modo, pode ser implementado um receptor eficiente. Adicionalmente, pode ser implementado um sistema mais robusto ao ruído do que o anterior $NU-QAM$. Para um sistema mais eficiente de transmissão de radiodifusão, é possível a hibridação $MQAM$ e $NU-MQAM$. Por outras palavras, pode ser implementado um sistema mais robusto ao ruído utilizando $MQAM$ para um ambiente onde um código de correcção de erro com elevada taxa de código é utilizado e empregando de outro modo $NU-MQAM$. Para um tal caso, um transmissor pode permitir que um receptor possua informação da taxa de código de um código de correcção de erro actualmente utilizado e um género de modulação actualmente utilizado de tal forma que o receptor pode desmodular de acordo com a modulação actualmente utilizada.

A figura 34 mostra um exemplo de sistema de transmissão digital. As entradas podem incluir um número de fluxos *MPEG-TS* ou fluxos *GSE* (*General Stream Encapsulation*). Um módulo 101 de processador de entrada pode adicionar parâmetros de transmissão ao fluxo de entrada e executar o agendamento para um módulo *BICM* 102. O módulo *BICM* 102 pode adicionar redundância e intercalar dados para correcção de erros no canal de transmissão. Um construtor de quadros 103 pode construir quadros adicionando a informação de sinalização da camada física e pilotos. Um modulador 104 pode executar a modulação nos símbolos de entrada em processos eficientes. Um processador analógico 105 pode executar vários processos para converter sinais de entrada digitais em sinais de saída analógicos.

A figura 35 mostra um exemplo de um processador de entrada. O fluxo de entrada *MPEG-TS* ou *GSE* pode ser transformado pelo pré-processador de entrada num total de n fluxos que serão processados independentemente. Cada um destes fluxos pode ser um quadro *TS* completo que inclui componentes de vários serviços ou um quadro *TS* mínimo que inclui componente de serviço (site é vídeo ou áudio). Além disso, cada um destes fluxos pode ser um fluxo *GSE* que transmite vários serviços ou um único serviço.

O módulo de interface de entrada 202-1 pode atribuir vários bits de entrada iguais à capacidade máxima do campo de dados de um quadro de banda de base (*BB*). Uma

zona de preenchimento pode ser inserida para completar a capacidade do bloco de código *LDPC/BCH*. O módulo de sincronismo do fluxo de entrada 203-1 pode proporcionar um mecanismo para regenerar, no receptor, o relógio do fluxo de transporte (ou fluxo genérico em pacotes), a fim de garantir taxas e atrasos de bits constantes extremidade-a-extremidade.

De modo a permitir a recombinação do fluxo de transporte sem exigir memória adicional no receptor, os fluxos de transporte de entrada são atrasados por compensadores de atraso 204-1~n considerando os parâmetros da intercalação dos *PLPs* de dados num grupo e o *PLP* comum correspondente. O módulo de apagar 205-1~n de pacote de tamanho zero pode aumentar a eficiência de transmissão através da remoção de pacotes de tamanho zero inseridos em caso de serviço *VBR* (*variable bit rate* - taxa de bits variável). Os módulos codificadores de verificação cíclica de redundância (*Cyclic Redundancy Check* - *CRC*) 206-1~n podem adicionar paridade *CRC* para aumentar a confiabilidade da transmissão de quadro *BB*. Os módulos 207-1~n de inserção de cabeçalho *BB* podem adicionar o cabeçalho do quadro numa parte inicial do quadro *BB*. A informação que pode ser incluída no cabeçalho *BB* encontra-se apresentada na figura 36.

Um módulo de fusão/corte 208 pode executar o corte em fatias do quadro *BB* de cada *PLP*, fundindo quadros *BB* de vários *PLPs*, e agendando cada quadro *BB* dentro de um

quadro de transmissão. Por isso, o módulo de fusão/corte 208 pode emitir informação de sinalização *L1* que se refere à atribuição de *PLP* no quadro. Por último, um módulo misturador *BB* 209 pode randomizar fluxos de bits de entrada para minimizar a correlação entre os bits dentro de fluxos de bits. Os módulos a sombreado na figura 35 são módulos usados quando o sistema de transmissão utiliza um único *PLP*, sendo os outros módulos na figura 35 módulos usados quando o dispositivo de transmissão utiliza vários *PLPs*.

A figura 37 mostra um exemplo do módulo *BICM*. A figura 37a mostra o trajecto dos dados e a figura 37b mostra o trajecto *L1* do módulo *BICM*. Um módulo codificador externo 301 e um módulo codificador interno 303 podem adicionar redundância a fluxos de bits de entrada para correcção de erros. Um módulo intercalador externo 302 e um módulo intercalador interno 304 podem intercalar bits para evitar erro de rajada. O módulo intercalador externo 302 pode ser omitido se o *BICM* for especificamente para *DVB-C2*. Um módulo desmultiplexador de bits 305 pode controlar a fiabilidade de cada bit emitido pelo módulo intercalador interno 304. Um módulo de mapeamento de símbolo 306 pode corresponder os fluxos de bits de entrada a fluxos de símbolos. Nesse momento, é possível utilizar qualquer de uma *QAM* convencional, uma *MQAM* que utiliza o *BRGC* acima descrito para a melhoria do desempenho, uma *NU-QAM* que utiliza modulação não-uniforme, ou uma *NU-MQAM* que utiliza *BRGC* aplicado de modulação não-uniforme para melhoria do desempenho. Para construir um sistema que é mais robusto

contra o ruído, podem ser consideradas as combinações de modulações que utilizam *MQAM* e/ou *NU-MQAM* dependendo da taxa de código do código de correcção de erros e da capacidade de constelação. Nesse momento, o módulo de mapeamento de símbolos 306 pode utilizar uma constelação adequada de acordo com a taxa de código e capacidade da constelação. A figura 39 mostra um exemplo de tais combinações.

O caso 1 mostra um exemplo de utilização exclusiva de *NU-MQAM* à taxa de código baixa para implantação simplificada do sistema. O caso 2 mostra um exemplo de utilização de constelação optimizada a cada taxa de código. O transmissor pode enviar informação sobre a taxa de código do código de correcção de erros e sobre a capacidade da constelação para o receptor de tal forma que o receptor pode utilizar uma constelação adequada. A figura 40 mostra um outro exemplo de casos em que a compatibilidade entre os sistemas convencionais é considerada. Adicionalmente aos exemplos são possíveis combinações adicionais para optimizar o sistema.

O módulo de inserção 307 do cabeçalho *ModCod* apresentado na figura 37 pode obter informação de retorno de codificação e modulação adaptativa (*Adaptive coding and modulation - ACM*) / de codificação e modulação variável (*Variable coding and modulation - VCM*) e adicionar informações de parâmetro utilizadas na codificação e modulação para um bloco *FEC* como cabeçalho. O cabeçalho do

tipo de modulação/taxa de código (*ModCod*) pode incluir a seguinte informação:

- * Tipo de *FEC* (1 bit) - *LDPC* longo ou curto
- * Taxa de código (3 bits)
- * Modulação (3 bits) - até 64K *QAM*
- * Identificador *PLP* (8 bits)

O módulo intercalador de símbolos 308 pode realizar a intercalação no domínio do símbolo para obter efeitos adicionais de intercalação. Processos semelhantes realizados no trajecto dos dados podem ser realizados no trajecto de sinalização *L1* mas possivelmente com parâmetros diferentes 301-1 ~ 308-1. Neste ponto pode ser utilizado, um módulo de código encurtado/puncionado (303-1) para código interno.

A figura 38 mostra um exemplo de codificação *LDPC* utilizando encurtamento / puncionagem. O processo de encurtamento pode ser realizado em blocos de entrada que apresentam menos bits do que um número necessário de bits para a codificação *LDPC* dado que tantos zero bits necessários para a codificação *LDPC* podem ser colocados em zonas de preenchimento (301c). Fluxos de bits de entrada de zonas de preenchimento a zero podem apresentar bits de paridade através da codificação *LDPC* (302c). Nesse momento, para fluxos de bits que correspondem a fluxos de bits originais, os zeros podem ser removidos (303c) e para fluxos de bits de paridade, o puncionamento (304C) pode ser

executado de acordo com as taxas de código. Estes fluxos processados de bits de informação e fluxos de bits de paridade podem ser multiplexados em sequências originais e emitidos (305c).

A figura 41 mostra uma estrutura de quadro que compreende o preâmbulo para a sinalização *L1* e símbolo de dados para dados *PLP*. Pode-se observar que o preâmbulo e os símbolos de dados são gerados ciclicamente, utilizando um quadro como uma unidade. Os símbolos de dados incluem *PLP* tipo 0 que é transmitido utilizando uma modulação/codificação fixa e *PLP* tipo 1 que é transmitido utilizando uma modulação/codificação variável. Para *PLP* tipo 0, a informação tal como a modulação, tipo de *FEC*, e taxa de código *FEC* são transmitidos no preâmbulo (veja figura 42 inserção de cabeçalho do quadro 401). Para *PLP* tipo 1, a informação correspondente pode ser transmitida num cabeçalho de bloco *FEC* de um símbolo de dados (veja figura 37 inserção do cabeçalho *ModCod* 307). Pela separação de tipos *PLP*, a informação complementar do *ModCod* pode ser reduzida em 3~4% de uma taxa de transmissão total, para *PLP* type0 que é transmitido a uma taxa de bits fixa. Num receptor, para a modulação fixa/codificação *PLP* do *PLP* tipo 0, o removedor de cabeçalho do quadro r401 apresentado na figura 63 pode extrair informação sobre a modulação e taxa do código *FEC* e fornecer a informação extraída a um módulo de descodificação *BICM*. Para modulação/codificação variável *PLP* do *PLP* tipo 1, os módulos de extracção *ModCod*, r307 e r307-1 apresentados na figura 64 pode extrair e fornecer os parâmetros necessários para a descodificação *BICM*.

A figura 42 mostra um exemplo de um construtor de quadros. Um módulo de inserção de cabeçalho de quadro 401 pode construir um quadro de fluxos de símbolos de entrada e pode adicionar o cabeçalho do quadro na frente de cada quadro transmitido. O cabeçalho do quadro pode incluir a seguinte informação:

- * *Number of bonded channels (4 bits)*
- * *Guard interval (2 bits)*
- * *PAPR (2 bits)*
- * *Pilot pattern (2 bits)*
- * *Digital System identification (16 bits)*
- * *Frame identification (16 bits)*
- * *Frame length (16 bits) - number of Orthogonal Frequency Division Multiplexing (OFDM) symbols per frame*
- * *Superframe length (16 bits) - number of frames per superframe*
- * *number of PLPs (8 bits)*
- * *for each PLP*
 - PLP identification (8 bits)*
 - Channel bonding id (4 bits)*
 - PLP start (9 bits)*
 - PLP type (2 bits) - common PLP or others*
 - PLP payload type (5 bits)*
 - MC type (1 bit) - fixed/variable modulation & coding*
 - if MC type == fixed modulation & coding*
 - FEC type (1 bits) - long or short LDPC*

```
Coderate (3 bits)
Modulation (3 bits) - up-to 64K QAM
end if;
Number of notch channels (2 bits)
for each notch
    Notch start (9 bits)
    Notch width (9 bits)
end for;
PLP width (9 bits) - max number of FEC blocks of PLP
PLP time interleaving type (2 bits)
end for;
* CRC-32 (32 bits)
```

O ambiente de ligação de canal é assumido para informação *L1* transmitida no cabeçalho do quadro, sendo os dados que correspondem a cada fatia de dados definidos como *PLP*. Por isso, a informação tal como o identificador *PLP*, identificador de ligação de canal, e endereço de início de *PLP* é necessária para cada canal utilizado na ligação. Uma forma de realização da presente invenção sugere transmitir o campo *ModCod* no cabeçalho do quadro *FEC* se o tipo *PLP* suportar modulação variável/codificação e transmitir campo *ModCod* no cabeçalho do quadro se o tipo *PLP* suportar modulação fixa/codificação para reduzir a informação complementar da sinalização. Adicionalmente, se existir uma banda de entalhe para cada *PLP*, ao transmitir o endereço de início do entalhe e a sua largura, podem tornar-se desnecessárias as portadoras correspondentes de descodificação no receptor.

A figura 43 mostra um exemplo de padrão piloto 5 (PP5) aplicado num ambiente de ligação de canal. Tal como apresentado, se as posições *SP* forem coincidentes com as posições do preâmbulo piloto, pode surgir uma estrutura piloto irregular.

A figura 43a mostra um exemplo de módulo de inserção de piloto 404 tal como apresentado na figura 42. Tal como representado na figura 43, se for utilizada uma banda de uma só frequência (por exemplo 8 MHz), a largura de banda disponível é de 7,61 MHz, mas se se encontrarem ligadas várias bandas de frequência, as bandas de guarda pode ser removidas, sendo que, deste modo, a eficiência da frequência pode aumentar consideravelmente. A figura 43b é um exemplo do módulo de inserção do preâmbulo 504 tal como apresentado na figura 51 que é transmitido na parte frontal do quadro e mesmo com ligação de canal, o preâmbulo apresenta uma taxa de repetição de 7,61 MHz, que é a largura de banda do bloco *L1*. Esta é uma estrutura considerando a largura de banda de um sintonizador que executa o varrimento inicial do canal.

Existem padrões piloto para ambos, o preâmbulo, e os símbolos de dados. Para símbolo de dados, podem ser usados padrões piloto difundidos (*scattered pilot - SP*). Os padrões piloto 5 (PP5) e padrões piloto 7 (PP7) de T2 podem ser bons candidatos para a interpolação somente de frequência. PP5 tem $x=12$, $y=4$, $z=48$ para $GI=1/64$ e PP7 tem

$x=24$, $y=4$, $z=96$ para $GI=1/128$. É também possível interpolação adicional no tempo para uma melhor estimativa de canal. Os padrões piloto para preâmbulo podem cobrir todas as posições piloto possíveis para a aquisição inicial do canal. Adicionalmente, as posições do preâmbulo piloto devem ser coincidentes com as posições *SP*, sendo desejado um padrão piloto único para ambos, o preâmbulo e o *SP*. Os preâmbulos piloto podem também ser utilizados para a interpolação no tempo e todos os preâmbulos podem apresentar um padrão piloto idêntico. Estes requisitos são importantes para a detecção de C2 no varrimento e necessários para a estimativa offset da frequência com correlação da sequência de codificação. Num ambiente de ligação de canal, a coincidência nas posições piloto deve também ser mantida para a ligação do canal porque a estrutura piloto irregular pode prejudicar o desempenho da interpolação.

Em pormenor, se uma distância z entre pilotos difundidos (*SPs*) num símbolo *OFDM* for de 48 e se uma distância y entre *SPs* que correspondem a uma portadora *SP* específica ao longo do eixo do tempo for de 4, uma distância efectiva x após a interpolação no tempo torna-se 12. Isto é quando a fracção do intervalo de guarda (*GI*) é de $1/64$. Se a fracção *GI* for de $1/128$, pode ser utilizado $x=24$, $y=4$, e $z=96$. Se a fracção *GI* for de $1/128$, pode ser utilizado $x=24$, $y=4$, e $z=96$. Se for utilizada a ligação de canal, as posições *SP* podem ser realizadas coincidentes com as posições do preâmbulo piloto gerando pontos não-contínuos na estrutura piloto difundida.

Nesse momento, as posições do preâmbulo piloto podem ser coincidentes com todas as posições de *SP* do símbolo de dados. Quando é utilizada a ligação do canal, a fatia de dados onde um serviço é transmitido, pode ser determinada independentemente da granularidade da largura de banda de 8 MHz. Contudo, para reduzir a informação complementar para o endereçamento da fatia de dados, pode ser escolhida a transmissão que inicia na posição *SP* e que termina na posição *SP*.

Quando um receptor recebe tais *SPs*, se necessário, o módulo da estimativa de canal r501 apresentado na figura 62 pode executar a interpolação no tempo para obter os pilotos apresentados em linhas a ponteados na figura 43 e executar a interpolação da frequência. Nesse momento, para os pontos não-contínuos dos quais os intervalos se encontram assinalados como 32 na figura 43, pode ser implementada a execução de interpolações à esquerda e direita separadamente ou executar interpolações em apenas um lado, executando depois a interpolação no outro lado utilizando as posições piloto já interpoladas, em que pode ser implementado o intervalo de 12 como ponto de referência. Nesse momento, a largura da fatia de dados pode variar dentro de 7,61 MHz, sendo que deste modo um receptor pode minimizar o consumo de energia executando a estimativa de canal e descodificando apenas as subportadoras necessárias.

A figura 44 mostra um outro exemplo de *PP5* aplicado no ambiente de ligação de canal ou uma estrutura de *SP* para manter a distância *eFECTiva* x como 12 para evitar a estrutura *SP* irregular apresentada na figura 43 quando é utilizada a ligação de canal. A figura 44a é uma estrutura de *SP* para o símbolo de dados e a figura 44b é uma estrutura de *SP* para o símbolo de preâmbulo.

Tal como apresentado, se a distância *SP* for mantida consistente em caso de ligação de canal, não haverá qualquer problema na interpolação da frequência mas as posições piloto entre os símbolos de dados e preâmbulo podem não ser coincidentes. Por outras palavras, esta estrutura não necessita de estimativa adicional de canal para uma estrutura *SP* irregular, no entanto, as posições *SP* utilizadas na ligação de canal e posições do preâmbulo piloto tornam-se diferentes para cada canal.

A figura 45 mostra uma nova estrutura *SP* ou *PP5'* para proporcionar uma solução aos dois problemas acima mencionados no ambiente de ligação de canal. Especificamente, uma distância piloto de $x=16$ pode solucionar aqueles problemas. Para preservar a densidade piloto ou para manter a mesma informação complementar, um *PP5'* pode apresentar $x=16$, $y=3$, $z=48$ para $GI=1/64$ e um *PP7'* pode apresentar $x=16$, $y=6$, $z=96$ para $GI=1/128$. A capacidade de interpolação somente da frequência pode ainda ser mantida. As posições piloto encontram-se representadas na figura 45 para comparação com a estrutura *PP5*.

A figura 46 mostra um exemplo de um novo padrão *SP* ou estrutura *PP5* em ambiente de ligação de canal. Tal como apresentado na figura 46, se for utilizado um canal simples ou ligação de canal, pode ser proporcionada uma distância piloto $x=16$ eficaz. Adicionalmente, porque as posições *SP* podem ser realizadas coincidentes com as posições do preâmbulo piloto, pode ser evitada a deterioração da estimativa do canal provocada pela irregularidade *SP* ou posições *SP* não coincidentes. Por outras palavras, não existe qualquer posição *SP* irregular para o interpolador de frequência, encontrando-se proporcionada a coincidência entre o preâmbulo e as posições *SP*.

Consequentemente, os novos padrões *SP* propostos podem ser vantajosos em que um único padrão *SP* pode ser utilizado para ambos o canal único e canal ligado; não pode ser provocada qualquer estrutura piloto irregular, sendo deste modo possível uma boa estimativa de canal; ambas as posições do preâmbulo e as posições piloto *SP* podem ser mantidas coincidentes, a densidade dos pilotos pode ser mantida a mesma que para *PP5* e *PP7* respectivamente, podendo também ser preservada a capacidade de interpolação somente da frequência.

Adicionalmente, a estrutura do preâmbulo pode corresponder aos requisitos de modo que as posições do preâmbulo piloto devem cobrir todas as posições *SP*

possíveis para aquisição inicial do canal; o número máximo de portadoras deve ser de 3409 (7,61 MHz) para o varrimento inicial; exactamente os mesmos padrões piloto e sequência de codificação devem ser utilizados para detecção de C2, não sendo necessário nenhum preâmbulo específico de detecção tal como P1 em T2.

Em termos de relação com a estrutura do quadro, a granularidade da posição da fatia de dados pode ser alterada para 16 portadoras em vez de 12, sendo que deste modo pode surgir menos informação complementar para endereçamento de posição, não se esperando qualquer outro problema no que se refere à condição da fatia de dados, podendo ser esperada condição de ranhura zero, etc.

Por isso, no módulo de estimativa de canal r501 da figura 62, os pilotos em todos os preâmbulos podem ser utilizados quando é executada a interpolação no tempo *SP* dos símbolos de dados. Por isso, a aquisição de canal e estimativa de canal nos limites do quadro podem ser melhoradas.

Agora, no que se refere aos requisitos relacionados com o preâmbulo e a estrutura piloto, existe consenso em que as posições dos preâmbulos piloto e *SPs* devem coincidir independentemente da ligação do canal; o número total de portadoras no bloco *L1* deve ser divisível pela distância piloto para evitar a estrutura irregular na extremidade da banda; os blocos *L1* devem ser repetidos no

domínio da frequência; e os blocos *L1* devem ser sempre descodificáveis na posição da janela de sintonia arbitrária. Os requisitos adicionais deveriam ser que as posições piloto e padrões devem ser repetidos por períodos de 8 MHz; o desvio correcto da frequência portadora deve ser estimado sem o conhecimento da ligação de canal; e a descodificação *L1* (reordenamento) impossível antes de o desvio da frequência ser compensado.

A figura 47 mostra um relacionamento entre o símbolo de dados e preâmbulo quando são utilizadas as estruturas do preâmbulo tal como apresentado na figura 52 e figura 53. O bloco *L1* pode ser repetido por períodos de 6 MHz. Para a descodificação *L1*, devem ser encontrados tanto o desvio da frequência como o padrão da deslocação do preâmbulo. A descodificação *L1* não é possível na posição arbitrária do sintonizador sem informação sobre a ligação de canal e um receptor não consegue diferenciar entre o valor de deslocação do preâmbulo e desvio da frequência.

Deste modo, um receptor, especificamente para o removedor r401 do cabeçalho do quadro apresentado na figura 63 para executar a descodificação de sinal *L1*, tem que ser obtida a estrutura de ligação de canal. Dado que é conhecida a quantidade esperada de deslocação do preâmbulo em duas regiões sombreadas verticalmente na figura 47, o módulo de sincronização tempo/frequência r505 na figura 62 pode estimar o desvio da frequência portadora. Com base na estimativa, o trajecto de sinalização *L1* (r308-1- r301-1) na figura 64 pode descodificar *L1*.

A figura 48 mostra um relacionamento entre o símbolo de dados e o preâmbulo quando é utilizada a estrutura de preâmbulo tal como apresentada na figura 55. O bloco *L1* pode ser repetido por períodos de 8 MHz. Para a descodificação *L1* é necessário ser encontrado somente o desvio da frequência, sendo que pode não ser necessário o conhecimento da ligação do canal. O desvio da frequência pode ser facilmente estimado utilizando uma sequência binária pseudo-aleatória (*Pseudo Random Binary Sequence - PRBS*). Tal como apresentado na figura 48, o preâmbulo e símbolos de dados encontram-se alinhados, sendo que deste modo a procura da sincronização adicional pode tornar-se desnecessária. Por isso, para um receptor, especificamente para o módulo removedor r401 do cabeçalho do quadro apresentado na figura 63, é possível que tenha que ser obtido somente o pico de correlação com sequência de codificação do piloto para executar a descodificação do sinal *L1*. O módulo de sincronização tempo/frequência r505 na figura 62 pode estimar o desvio da frequência portadora da posição pico.

A figura 49 mostra um exemplo de perfil de atraso de canal por cabo.

Do ponto de vista da concepção piloto, o *GI* actual já sobreprotege a dispersão dos tempos de propagação do canal de cabo. No pior dos casos, redesenhar o modelo de canal pode ser uma opção. Para repetir o padrão exactamente

a cada 8 MHz, a distância piloto deve ser um divisor de 3584 portadoras ($z=32$ ou 56). Uma densidade piloto de $z=32$ pode aumentar a informação complementar piloto, deste modo pode ser escolhido $z=56$. Uma cobertura do tempo de propagação ligeiramente menor pode não ser importante no canal de cabo. Por exemplo, pode ser $8\text{ }\mu\text{s}$ para $PP5'$ e $4\text{ }\mu\text{s}$ para $PP7'$ comparado com $9,3\text{ }\mu\text{s}$ ($PP5$) e $4,7\text{ }\mu\text{s}$ ($PP7$). Atrasos significativos podem ser cobertos por ambos os padrões piloto no pior dos casos. Para a posição do preâmbulo piloto, não são necessários mais do que todas as posições SP no símbolo de dados.

Se o trajecto de atraso de -40 dB puder ser ignorado, a distribuição actual dos tempos de propagação pode tornar-se $2,5\text{ }\mu\text{s}$, $1/64\text{ GI} = 7\text{ }\mu\text{s}$, ou $1/128\text{ GI} = 3,5\text{ }\mu\text{s}$. Isso mostra que o parâmetro da distância piloto, $z=56$ pode ser um valor suficientemente bom. Adicionalmente, $z=56$ pode ser um valor conveniente para estruturar o padrão piloto que permite a estrutura preâmbulo apresentada na figura 48.

A figura 50 mostra uma estrutura piloto dispersa que utiliza $z=56$ e $Z=112$ que é construída no módulo de inserção de piloto 404 na figura 42. São propostos $PP5'$ ($x=14$, $y=4$, $z=56$) e $PP7'$ ($x=28$, $y=4$, $z=112$). Podem ser inseridas portadoras de extremidade para o $FEcho$ da extremidade.

Tal como apresentado na figura 50, os pilotos

encontram-se alinhados a 8 MHz de cada extremidade da banda, podendo cada posição piloto e estrutura piloto ser repetida a cada 8 MHz. Deste modo, esta estrutura pode suportar a estrutura do preâmbulo apresentada na figura 48. Adicionalmente, pode ser utilizada uma estrutura piloto comum entre o preâmbulo e símbolos de dados. Por isso, o módulo de estimativa de canal r501 na figura 62 pode executar a estimativa de canal utilizando a interpolação no preâmbulo e símbolos de dados porque não pode surgir qualquer padrão piloto irregular, independentemente da posição da janela que é decidida pelos locais das fatias de dados. Nesse momento, utilizar somente a interpolação na frequência pode ser suficiente para compensar a distorção de canal da dispersão dos tempos de propagação. Se a interpolação no tempo for adicionalmente executada, pode ser realizada uma estimativa de canal mais precisa.

Por conseguinte, no novo padrão piloto proposto, a posição piloto e padrão podem ser repetidos com base num período de 8 MHz. Pode ser utilizado um único padrão piloto para ambos, o preâmbulo e os símbolos de dados. A descodificação *L1* pode sempre ser possível sem o conhecimento da ligação do canal. Adicionalmente, o padrão-piloto proposto pode não afetar vulgarmente com T2 porque pode ser utilizada a mesma estratégia piloto de padrão piloto difundido; T2 já utiliza 8 padrões piloto diferentes, sendo que nenhuma complexidade significativa do receptor pode ser aumentada por padrões piloto modificados. Para uma sequência de codificação piloto, o período de *PRBS*

pode ser 2047 (sequência-m); a geração de *PRBS* pode ser reinicializada a cada 8 MHz, em que o período é de 3584; a taxa de repetição piloto de 56 pode ser também co-primo com 2047; não sendo esperado qualquer problema *PAPR* .

A figura 51 mostra um exemplo de um modulador baseado em *OFDM*. Os fluxos de símbolos de entrada podem ser transformados em domínio do tempo pelo módulo *IFFT* 501. Se necessário, a relação potência de pico/potência média (*peak-to-average power ratio - PAPR*) pode ser reduzida no módulo 502 redutor de *PAPR* . Para os processos *PAPR* , podem ser utilizadas a extensão da constelação activa (*ACE*) ou reserva de tom. O módulo insersor de *GI* 503 pode copiar uma última parte do símbolo *OFDM* eFECTivo para preencher o intervalo de guarda numa forma de prefixo cíclico.

O módulo insersor de preâmbulo 504 pode inserir preâmbulo na frente de cada quadro transmitido de modo que um receptor pode detectar o sinal digital, quadro e adquirir a aquisição do desvio de tempo/frequência. Nesse momento, o sinal do preâmbulo pode executar a sinalização da camada física tal como dimensão *FFT* (3 bits) e dimensão de intervalo de guarda (3 bits). O módulo de inserção do preâmbulo 504 pode ser omitido se o modulador for especificamente para *DVB-C2*.

A figura 52 mostra um exemplo de uma estrutura de preâmbulo para ligação de canal, gerado no módulo de inserção do preâmbulo 504 na figura 51. Um bloco *L1*

completo deve ser "sempre descodificável" em qualquer posição arbitrária de janela de sintonia 7,61 MHz e não deve ter lugar qualquer perda da sinalização *L1* independentemente da posição da janela do sintonizador. Tal como apresentado, os blocos *L1* podem ser repetidos no domínio da frequência por períodos de 6 MHz. O símbolo de dados pode ser ligado por canal a cada canal de 8 MHz. Se, para a descodificação *L1*, um receptor utilizar um sintonizador tal como o sintonizador r603 representado na figura 61 que utiliza uma largura de banda de 7,61 MHz, o removedor de cabeçalho r401 de quadro na figura 63 precisa de reorganizar o bloco recebido *L1* deslocado ciclicamente (figura 53) para a sua forma original. Este rearranjo é possível porque o bloco *L1* é repetido para cada bloco de 6MHz. A figura 53a pode ser reordenada transformando-se na figura 53b.

A figura 54 mostra um processo para conceber um preâmbulo mais optimizado. A estrutura de preâmbulo da figura 52 utiliza apenas 6MHz de largura de banda total de 7,61 MHz do sintonizador para descodificação *L1*. Em termos de eficiência de espectro, a largura de banda de 7,61 MHz não é totalmente utilizada. Portanto, pode haver uma maior optimização na eficiência espectral.

A figura 55 mostra um outro exemplo da estrutura preâmbulo ou estrutura preâmbulo dos símbolos para a eficiência completa do espectro, gerada no módulo 401 de inserção do cabeçalho no quadro na figura 42. Tal como o

símbolo de dados, os blocos *L1* podem ser repetidos no domínio da frequência por períodos de 8 MHz. Um bloco *L1* completo encontra-se ainda "sempre descodificável" em qualquer posição arbitrária da janela de sintonia de 7,61 MHz. Após a sintonia, os dados a 7,61 MHz podem ser vistos como um código virtualmente puncionado. Tendo exactamente a mesma largura de banda para ambos o preâmbulo e os símbolos de dados e exactamente a mesma estrutura piloto para ambos o preâmbulo e símbolos de dados, pode maximizar a eficiência do espectro. Outras características tais como a propriedade deslocada ciclicamente e não enviar o bloco *L1* em caso de não haver fatias de dados, podem ser mantidas inalteradas. Por outras palavras, a largura de banda dos símbolos preâmbulo pode ser idêntica à largura de banda dos símbolos de dados ou, tal como apresentado na figura 57, a largura de banda dos símbolos preâmbulo pode ser a largura de banda do sintonizador (aqui é de 7,61 MHz). A largura de banda do sintonizador pode ser definida como uma largura de banda que corresponde a um número de portadoras activas totais quando é utilizado um único canal. Quer dizer, a largura de banda do símbolo preâmbulo pode corresponder ao número de portadoras activas totais (aqui é de 7,61 MHz).

A figura 56 mostra um código virtualmente puncionado. Os dados de 7,61 MHz entre o bloco *L1* de 8 MHz pode ser considerado como código puncionado. Quando um sintonizador r603 mostrado na figura 61 utiliza uma largura de banda de 7,61 MHz para a descodificação *L1*, o removedor de cabeçalho r401 de quadro na figura 63 precisa de

reorganizar o bloco *L1* recebido, deslocado ciclicamente para a sua forma original tal como apresentado na figura 56. Nesse momento, é executada a descodificação *L1* utilizando toda a largura de banda do sintonizador. Assim que o bloco *L1* estiver rearranjado, um espectro do bloco *L1* reorganizado pode apresentar uma região em branco dentro do espectro tal como apresentado na parte superior direita da figura 56 porque um tamanho original do bloco *L1* apresenta 8 MHz de largura de banda.

Assim que a região vazia se encontrar preenchida com zeros, após o desintercalamento no domínio dos símbolos pelo desintercalador de frequência r403 na figura 63 ou pelo desintercalador de símbolos r308-1 na figura 64 ou após o desintercalamento no domínio de bits pelo desmapeador de símbolos r306-1, multiplexador de bits 305-1, e desintercalador interno r304-1 na figura 64, o bloco pode apresentar uma forma que parece ser puncionada tal como apresentado no lado direito inferior da figura 56.

Este bloco *L1* pode ser descodificado no módulo de descodificação puncionado/encurtado r303-1 na figura 64. Ao utilizar esta estrutura preâmbulo, pode ser utilizada toda a largura de banda do sintonizador, podendo deste modo ser aumentados a eficiência do espectro e o ganho da codificação. Adicionalmente pode ser utilizada uma largura de banda idêntica e estrutura piloto comum para o preâmbulo e símbolos de dados.

Adicionalmente, se a largura de banda do preâmbulo ou a largura de banda dos símbolos do preâmbulo se encontrar definida como uma largura de banda de sintonizador como apresentado na figura 58, (é de 7,61 MHz no exemplo), pode ser obtido um bloco *L1* completo após a reorganização mesmo sem o funcionamento. Por outras palavras, para um quadro que apresenta símbolos de preâmbulo, onde os símbolos do preâmbulo apresentam pelo menos um bloco de camada 1 (*L1*), pode-se dizer que o bloco *L1* tem 3408 subportadoras activas e as 3408 subportadoras activas correspondem a 7,61 MHz de 8MHz de banda de frequência de rádio (RF).

Deste modo, a eficiência do espectro e desempenho da descodificação *L1* pode ser maximizada. Por outras palavras, no receptor, a descodificação pode ser realizada no módulo descodificador punctionado/encurtado r303-1 na figura 64, após a realização de apenas desintercalamento no domínio dos símbolos.

Consequentemente, a nova estrutura preâmbulo proposta pode ser vantajosa por ser totalmente compatível com o preâmbulo utilizado anteriormente excepto que a largura de banda é diferente; os blocos *L1* são repetidos por períodos de 8 MHz; o bloco *L1* pode ser sempre descodificável independentemente da posição da janela do sintonizador; pode ser utilizada para descodificação *L1* a largura de banda completa do sintonizador; a eficiência máxima do espectro pode garantir mais ganho de código; o

bloco *L1* incompleto pode ser considerado como codificado punccionado; pode ser utilizada uma estrutura piloto simples e a mesma para ambos, o preâmbulo, e os dados; e largura de banda idêntica pode ser utilizada para ambos o preâmbulo e dados.

A figura 59 mostra um exemplo de um processador analógico. Um módulo DAC 601 pode converter uma entrada de sinal digital em sinal analógico. Após a largura de banda da frequência de transmissão ter sido convertida para cima (602) e filtrada analogicamente (603) o sinal pode ser transmitido.

A figura 60 mostra um exemplo de um sistema de recepção digital. O sinal recebido é convertido em sinal digital num módulo r105 de processo analógico. Um desmodulador r104 pode converter o sinal em dados no domínio da frequência. Um analisador de quadros r103 pode remover pilotos e cabeçalhos e activar a selecção de informação de serviço que necessita de ser descodificada. Um desmodulador *BICM* r102 pode corrigir erros no canal de transmissão. Um processador de saída r101 pode restaurar o fluxo de serviço transmitido originalmente e informação de temporização.

A figura 61 mostra um exemplo de processador analógico utilizado no receptor. Um módulo/sintonizador AGC r603 pode seleccionar a largura de banda de frequência desejada a partir do sinal recebido. Um módulo de conversão

para baixo r602 pode restaurar a banda de base. Um módulo ADC r601 pode converter um sinal analógico em sinal digital.

A figura 62 mostra um exemplo de desmodulador. Um módulo de detecção de quadro r506 pode detectar o preâmbulo, verificar se existe um sinal digital correspondente, e detectar o início de um quadro. Um módulo de sincronização de tempo/frequência r505 pode executar a sincronização nos domínios do tempo e frequência. Nesse momento, para a sincronização no domínio do tempo, pode ser utilizada uma correlação de intervalo de guarda. Para a sincronização no domínio da frequência, pode ser utilizada a correlação ou o desvio pode ser estimado a partir da informação da fase de uma subportadora que é transmitida no domínio da frequência. Um módulo removedor de preâmbulo r504 pode remover o preâmbulo da frente do quadro detectado. Um módulo de remoção de *GI*, r503, pode remover o intervalo de guarda. Um módulo *FFT*, r501, pode transformar o sinal no domínio do tempo em sinal no domínio da frequência. Um módulo de estimativa/igualização de canal r501 pode compensar erros estimando a distorção no canal de transmissão utilizando o símbolo piloto. O módulo de remoção do preâmbulo r504 pode ser omitido se o desmodulador for especificamente para *DVB-C2*.

A figura 63 mostra um exemplo de analisador de quadros. Um módulo removedor de piloto r404 pode remover símbolo piloto. Um módulo de desintercalamento de

frequência r403 pode executar o desintercalamento no domínio da frequência. Um concentrador de símbolos *OFDM* r402 pode restaurar o quadro de dados de fluxos de símbolos transmitidos em símbolos *OFDM*. Um módulo r401 removedor de cabeçalho de quadro pode extrair sinalização de camada física do cabeçalho de cada quadro transmitido e remover o cabeçalho. A informação extraída pode ser utilizada como parâmetros para os seguintes processos no receptor.

A figura 64 mostra um exemplo de um desmodulador *BICM*. A figura 64a mostra um trajecto de dados e a figura 64b mostra um trajecto de sinalização *L1*. Um desintercalador de símbolos r308 pode executar o desintercalamento no domínio dos símbolos. Um extracto *ModCod* r307 pode extrair parâmetros *ModCod* da frente de cada quadro *BB* e tornar os parâmetros disponíveis para a seguinte desmodulação adaptativa/variável e processos de decodificação. Um desmapeador r306 de símbolo pode desmapear fluxos símbolos de entrada transformando em fluxos de bits *Log-Likelihood Ratio (LLR)*. Os fluxos *LLR* de bits de saída podem ser calculados utilizando uma constelação utilizada num mapeador de símbolos 306 do transmissor como ponto de referência. Neste ponto, quando é utilizado o anteriormente referido *MQAM* ou *NU-MQAM*, ao calcular ambos o eixo I e eixo Q quando se calcula o bit mais próximo de *MSB* e ao calcular o eixo I ou eixo Q quando se calcula os bits residuais, pode ser implementado um desmapeador de símbolos eficiente. Este processo pode ser aplicado a, por exemplo, *LLR* aproximado, *LLR* exacto, ou decisão difícil.

Quando é utilizada uma constelação otimizada de acordo com a capacidade da constelação e taxa de código do código de correcção de erro no mapeador de símbolos 306 do transmissor, o desmapeador de símbolos r306 do receptor pode obter uma constelação utilizando a taxa de código e informação sobre a capacidade da constelação transmitida do transmissor. O multiplexador de bits r305 do receptor pode executar uma função inversa do demultiplexador de bits 305 do transmissor. O desintercalador interno r304 e desintercalador externo r302 do receptor pode executar funções inversas do intercalador interno 304 e intercalador externo 302 do transmissor, para obter respectivamente a corrente de bits na sua sequência original. O desintercalador externo r302 pode ser omitido se o desmodulador *BICM* for especificamente para *DVB-C2*.

O descodificador interno r303 e o descodificador externo r301 do receptor podem executar os processos de descodificação correspondentes para o codificador interno 303 e código externo 301 do transmissor, respectivamente, para corrigir erros no canal de transmissão. Processos semelhantes realizados no trajecto dos dados podem ser realizados no trajecto de sinalização *L1* mas com parâmetros diferentes 308-1 ~ 301-1. Neste ponto, tal como explicado na parte do preâmbulo, pode ser utilizado um módulo de código encurtado/puncionado r303-1 para descodificação do sinal *L1*.

A figura 65 mostra um exemplo de descodificação *LDPC* utilizando encurtamento / puncionagem. Um desmultiple-xador r301a a pode emitir separadamente parte de informação e parte de paridade de código sistemático a partir de fluxos de bits de entrada. Para a parte de informação, pode ser realizada uma zona de preenchimento a zero (r302a) de acordo com uma série de fluxos de bits de entrada do descodificador *LDPC*, para a parte da paridade, fluxos de bits de entrada para (r303a) o descodificador *LDPC* podem ser gerados por despuncionagem da parte puncionada. A descodificação *LDPC* (r304a) pode ser executada em fluxos de bits gerados, podendo ser removidos os zeros na parte da informação e emitidos (r305a).

A figura 66 mostra um exemplo de processador de saída. Um descodificador *BB*, r209, pode restaurar fluxos de bits codificados (209) no transmissor. Um divisor r208 pode restaurar quadros *BB* que correspondem a vários *PLP*'s que são multiplexados e transmitidos a partir do transmissor de acordo com o trajecto *PLP*. Para cada trajecto *PLP*, um removedor de cabeçalho *BB* r207-1~n pode remover o cabeçalho que é transmitido na frente do quadro *BB*. Um descodificador CRC r206-1~n pode executar a descodificação CRC e tornar quadros *BB* fiáveis, disponíveis para selecção. Um módulo de inserção de pacote zero r205-1~n pode restaurar pacotes zero que foram removidos para uma maior eficiência de transmissão no seu local original. Um módulo de recuperação de atraso r204-1~n pode restaurar um tempo de propagação que existe entre cada trajecto *PLP*.

Um módulo de recuperação de relógio de saída $r203-1 \sim n$ pode restaurar a temporização original do fluxo de serviço a partir da informação de temporização transmitida do módulo de sincronização $203-1 \sim n$ do fluxo de entrada. Um módulo de interface de saída $r202-1 \sim n$ pode restaurar dados em pacote *TS/GS* de fluxos de bits de entrada que se encontram divididos em parcelas no quadro *BB*. Um módulo de pós-processamento de saída $r201-1 \sim n$ pode restaurar vários fluxos *TS/GS* convertendo-os num fluxo *TS/GS* completo, se necessário. Os blocos sombreados mostrados na figura 66 representam módulos que podem ser usados quando um único *PLP* é processado num período e o resto dos blocos representam os módulos que podem ser utilizados quando vários *PLPs* são simultaneamente processados.

Os padrões do preâmbulo piloto foram cuidadosamente projectados para evitar o aumento *PAPR*, sendo que, deste modo, se a taxa de repetição *L1* aumentar é necessário ser considerado o *PAPR*. O número de bits de informação *L1* varia dinamicamente de acordo com a ligação de canal, o número de *PLPs*, etc. Em pormenor, é necessário considerar coisas tais como o tamanho fixo do bloco *L1* pode introduzir informação complementar desnecessária; a sinalização *L1* deve ser protegida mais fortemente do que os símbolos de dados; e a intercalação no tempo do bloco *L1* pode melhorar a robustez em relação a danos no canal tal como a necessidade de ruído impulsivo.

Para uma taxa de repetição de bloco $L1$ de 8 MHz, tal como apresentado na figura 67 é exibida a eficiência do espectro completo (26,8% de aumento de BW) com funcionamento virtual mas o $PAPR$ pode ser ampliado dado que a largura de banda $L1$ é a mesma que a dos símbolos de dados. Para a taxa de repetição de 8 MHz, pode ser utilizado a intercalação de frequência 4K-FFT DVB-T2 para a uniformização, podendo o mesmo padrão repetir-se a um período de 8 MHz após a intercalação.

Para uma taxa de repetição de bloco $L1$ de 6 MHz, tal como apresentado na figura 68, pode ser exibida uma eficiência reduzida de espectro sem funcionamento virtual. Pode ocorrer um problema semelhante de $PAPR$ como para o caso de 8MHz dado que as larguras de banda de $L1$ e dos símbolos de dados partilham $LCM=24$ MHz. Para a taxa de repetição de 6 MHz, pode ser utilizado a intercalação de frequência 4K-FFT DVB-T2 para a uniformização, podendo o mesmo padrão repetir-se a um período de 24 MHz após a intercalação.

A figura 69 mostra uma nova taxa de repetição de bloco $L1$ de 7.61 MHz ou uma largura de banda completa de sintonizador. Uma eficiência de espectro largo (aumento de 26.8% de BW) pode ser obtida sem funcionamento virtual. Não pode haver qualquer problema com o $PAPR$ dado que $L1$ e larguras de banda de símbolo de dados partilham LCM 1704 MHz. Para a taxa de repetição de 7.61 MHz, pode ser utilizado a intercalação de frequência 4K-FFT DVB-T2 para a

uniformização, podendo o mesmo padrão repetir-se a um período de cerca de 1704 MHz após a intercalação.

A figura 70 é um exemplo de sinalização *L1* que é transmitida no cabeçalho do quadro. Cada informação na sinalização *L1* pode ser transmitida para o receptor e pode ser utilizada como um parâmetro de decodificação. Em especial, a informação pode ser utilizada no trajecto de sinal *L1* apresentado na figura 64, podendo os *PLPs* ser transmitidos em cada fatia de dados. Pode ser obtido um aumento da robustez para cada *PLP*.

A figura 72 é um exemplo de um intercalador de símbolos 308-1 tal como apresentado no trajecto de sinalização *L1* na figura 37 e também pode ser um exemplo do seu desintercalador de símbolos r308-1 correspondente tal como apresentado no trajecto de sinalização *L1* na figura 64. Os blocos com linhas inclinadas representam blocos *L1* e os blocos sólidos representam portadoras de dados. Os blocos *L1* podem ser transmitidos não somente dentro de um único preâmbulo, mas também podem ser transmitidos dentro de vários blocos *OFDM*. Dependendo do tamanho do bloco *L1*, o tamanho do bloco de intercalamento pode variar. Por outras palavras, *num_L1_sym* e amplitude *L1* podem ser diferentes umas das outras. Para minimizar a informação complementar desnecessária, podem ser transmitidos dados no resto das portadoras dos símbolos *OFDM* onde o bloco *L1* é transmitido. Neste ponto, pode ser garantida a eficiência completa do espectro porque o ciclo de repetição do bloco *L1* é ainda um

sintonizador de largura de banda completa. Na figura 72, os números em blocos com linhas inclinadas representam a ordem dos bits dentro de um único bloco *LDPC*.

Consequentemente, quando os bits são escritos numa memória de intercalamento no sentido da linha de acordo com um índice de símbolo tal como apresentado na figura 72 e lidos na direcção da coluna de acordo com um índice de portadora, pode ser obtido um efeito de intercalamento de bloco. Por outras palavras, um bloco *LDPC* pode ser intercalado no domínio do tempo e no domínio da frequência e depois ser transmitido. Num_ *L1*_sym pode ser um valor predeterminado, por exemplo um número entre 2-4 pode ser definido como uma série de símbolos *OFDM*. Neste ponto, para aumentar a granularidade da dimensão do bloco *L1*, pode ser utilizado para protecção do *L1* um código *LDPC* punccionado/encurtado que apresenta um comprimento mínimo da palavra-chave.

A figura 73 é um exemplo de uma transmissão de bloco *L1*. A figura 73 ilustra a figura 72 no domínio do quadro. Tal como apresentado na figura 73a, os blocos *L1* podem ser gerados em largura de banda completa de sintonizador ou tal como apresentado na figura 73b, os blocos *L1* podem ser parcialmente gerados e o resto das portadoras pode ser utilizado para o transporte de dados. Em ambos os casos, pode ser verificado que a taxa de repetição do bloco *L1* pode ser idêntico a uma largura de banda completa do sintonizador. Adicionalmente, para os

símbolos *OFDM* que utilizam a sinalização *L1* incluindo o preâmbulo, só pode ser executado a intercalação de símbolos enquanto não se permite a transmissão de dados nesses símbolos *OFDM*. Por conseguinte, para o símbolo *OFDM* utilizado para a sinalização *L1*, um receptor pode decodificar *L1* realizando o desintercalamento sem a decodificação de dados. Neste ponto, o bloco *L1* pode transmitir a sinalização *L1* do quadro actual ou sinalização *L1* de um quadro subsequente. No lado do receptor, os parâmetros *L1* decodificados, do trajecto de decodificação da sinalização *L1* apresentado na figura 64, podem ser utilizados para o processo de decodificação para o trajecto de dados do analisador de quadros do quadro subsequente.

Em resumo, num transmissor, os blocos de intercalação da região *L1* podem ser executados por blocos de escrita numa memória numa direcção em linha e leitura dos blocos escritos da memória no sentido da coluna. Num receptor, os blocos de desintercalamento da região *L1* podem ser executados escrevendo blocos numa memória numa direcção de coluna e a leitura dos blocos escritos da memória no sentido da linha. As indicações de leitura e escrita do transmissor e receptor podem ser intercambiadas.

Quando a simulação é realizada com suposições tais como serem realizados $CR=1/2$ para protecção *L1* e para a uniformização *T2*; a correspondência de símbolo 16-QAM; densidade piloto de 6 no preâmbulo; o número de *LDPC* curto

implica que seja eFECtuada a quantidade necessária de punção/ encurtamento, resultados ou conclusões tais como somente preâmbulo para a transmissão *L1* podem não ser suficientes; o número de símbolos *OFDM* depende da quantidade da dimensão do bloco *L1*; a palavra-chave *LDPC* mais curta (por exemplo 192 bits de informação) entre o código encurtado/punçado pode ser utilizada para flexibilidade e granularidade fina; podendo ser adicionada a zona de preenchimento, se necessário, com informações complementares negligenciáveis. O resultado encontra-se resumido na figura 71.

Consequentemente, para uma taxa de repetição de bloco *L1*, a largura de banda completa do sintonizador sem punção virtual pode ser uma boa solução e ainda não surgir qualquer problema do *PAPR* com a eficiência total do espectro. Para a sinalização *L1*, uma estrutura de sinalização eficiente pode permitir a configuração máxima num ambiente de 8 ligações de canal, 32 entalhes, 256 fatias de dados, e 256 *PLPs*. Para a estrutura de bloco *L1*, pode ser implementada a sinalização *L1* flexível de acordo com a dimensão do bloco *L1*. A intercalação no tempo pode ser executada para melhor robustez para a uniformização T2. Menos informação complementar pode permitir a transmissão de dados no preâmbulo.

Para uma melhor robustez pode ser executado a intercalação de blocos do bloco *L1*. A intercalação pode ser realizada com um número pré-definido fixo de símbolos *L1*

(*num_L1_sym*) e um número de portadoras, gerado por *L1* como um parâmetro (*L1_span*). A mesma técnica é utilizada para a intercalação do preâmbulo P2 em DVB-T2.

Pode ser utilizado um bloco *L1* de dimensão variável. O tamanho pode ser adaptável à quantidade de bits de sinalização *L1*, resultando numa informação complementar reduzida. A eficiência do espectro completo pode ser obtida sem nenhum problema *PAPR*. Uma repetição de menos do que 7,61 MHz pode significar que mais redundância pode ser enviada mas sem ter sido utilizada. Não pode surgir qualquer problema *PAPR* devido à taxa de repetição de 7,61 MHz para o bloco *L1*.

A figura 74 é um outro exemplo de sinalização *L1* transmitida dentro de um cabeçalho de quadro. Esta figura 74 é diferente da figura 70 em que o campo *L1_span* que apresenta 12 bits se encontra dividido em dois campos. Por outras palavras, o campo *L1_span* encontra-se dividido em uma *L1_column* que apresenta 9 bits e um *L1_row* que apresenta 3 bits. A *L1_column* representa o índice da portadora que *L1* gera. Dado que a fatia de dados inicia e termina a cada 12 portadoras, que é a densidade piloto, os 12 bits da informação complementar podem ser reduzidos em 3 bits para chegar a 9 bits.

L1_row representa o número de símbolos *OFDM* em que *L1* é gerado quando a intercalação no tempo é aplicada. Consequentemente, a intercalação no tempo pode ser

realizada dentro de uma área de *L1_columns* multiplicada por *L1_rows*. Alternativamente, um tamanho total de blocos *L1* pode ser transmitido de tal forma que *L1_span* mostrado na figura 70 pode ser utilizado quando a intercalação no tempo não é realizada. Para esse caso, o tamanho do bloco *L1* é de 11.776 x 2 bits no exemplo, sendo deste modo 15 bits o suficiente. Consequentemente, o campo *L1_span* pode ser composto por 15 bits.

A figura 75 é um exemplo de intercalação/desintercalação de frequência ou no tempo. A figura 75 mostra uma parte de um quadro completo de transmissão. A figura 75 mostra também a ligação de várias larguras de banda de 8 MHz. Um quadro pode ser composto por um preâmbulo que transmite blocos *L1* e um símbolo de dados que transmite dados. Os diferentes tipos de símbolos de dados representam fatias de dados para serviços diferentes. Tal como apresentado na figura 75, o preâmbulo transmite blocos *L1* por cada 7,61 MHz.

Para o preâmbulo, a intercalação na frequência ou tempo é realizada dentro de blocos *L1* e não realizada entre blocos *L1*. Ou seja, para o preâmbulo, pode-se dizer que a intercalação é realizada ao nível do bloco *L1*. Isto permite a descodificação dos blocos *L1* transmitindo blocos *L1* dentro de uma largura de banda de janela de sintonizador mesmo quando a janela do sintonizador se tiver movimentado para um local aleatório dentro de um sistema de ligação de canal.

Para a descodificação de símbolos de dados numa largura de banda aleatória de janela de sintonizador, não deve ter lugar a intercalação entre fatias de dados. Quer dizer, para fatias de dados, pode-se dizer que a intercalação é realizada ao nível da fatia de dados. Consequentemente, a intercalação na frequência e intercalação no tempo devem ser realizadas dentro de uma fatia de dados. Por isso, um intercalador de símbolos 308 num trajecto de dados de um módulo *BICM* do transmissor tal como apresentado na figura 37 pode realizar a intercalação de símbolos para cada fatia de dados. Um intercalador de símbolos 308-1 num trajecto de sinal *L1* pode executar a intercalação de símbolos para cada bloco *L1*.

Um intercalador de frequência 403 apresentado na figura 42 precisa de realizar separadamente a intercalação no preâmbulo e símbolos de dados. Especificamente, para o preâmbulo, a intercalação na frequência pode ser realizada para cada bloco *L1* e para o símbolo de dados, a intercalação na frequência pode ser realizada para cada fatia de dados. Neste ponto, a intercalação no tempo no trajecto de dados ou trajecto de sinal *L1* pode não ser realizada considerando o modo de baixa latência.

A figura 76 é um quadro que analisa a informação complementar da sinalização *L1* que é transmitida no cabeçalho *FECFRAME* no módulo de inserção de cabeçalho *ModCod* 307 no trajecto de dados do módulo *BICM* apresentado

na figura 37. Tal como apresentado na figura 76, para o bloco *LDPC* curto (tamanho=16.200), pode ocorrer uma informação complementar máxima de 3,3% que pode não ser insignificante. Na análise são assumidos 45 símbolos para protecção *FECFRAME*, sendo o preâmbulo uma sinalização *L1* específica do quadro C2 e o cabeçalho *FECFRAME* uma sinalização *L1* específica de *FECFRAME*, isto é identificador *Mod*, *Cod*, e *PLP*.

Para reduzir a informação complementar *L1*, podem ser consideradas abordagens de acordo com dois tipos de fatias de dados. Para o tipo *ACM/VCM* e vários casos *PLP*, o quadro pode ser mantido o mesmo que para o cabeçalho *FECFRAME*. Para o tipo *ACM/VCM* e casos *PLP* individuais, o identificador *PLP* pode ser removido do cabeçalho *FECFRAME*, resultando numa redução da informação complementar de até 1,8%. Para o tipo *CCM* e vários casos *PLP*, o campo *Mod/Cod* pode ser removido do cabeçalho *FECFRAME*, resultando numa redução da informação complementar de até 1,5%. Para o tipo *CCM* e casos *PLP* individuais, não é necessário qualquer cabeçalho *FECFRAME*, podendo ser obtido deste modo até 3,3% de redução.

Numa sinalização *L1* encurtada, pode ser transmitido identificador *Mod/Cod* (7 bits) ou *PLP* (8 bits), mas pode ser demasiado curto para obter novamente qualquer codificação. No entanto é possível não requerer sincronização porque os *PLPs* podem estar alinhados com o quadro de transmissão C2; todos os *ModCod* de cada *PLP* podem

ser conhecidos a partir do preâmbulo; e um cálculo simples pode permitir a sincronização com o *FECFRAME* específico.

A figura 77 mostra uma estrutura para o cabeçalho *FECFRAME* para minimizar a informação complementar. Na figura 77, os blocos com linhas inclinadas e o construtor *FECFRAME* representam um diagrama de blocos pormenorizado do módulo de inserção do cabeçalho *ModCod* 307 no trajecto de dados do módulo *BICM* tal como apresentado na figura 37. Os blocos sólidos representam um exemplo do módulo de codificação interno 303, intercalador interno 304, desmultiplicador de bits 305, e mapeador de símbolos 306 no trajecto de dados do módulo *BICM* tal como mostrado na figura 37. Neste ponto, pode ser executada a sinalização *L1* encurtada porque *CCM* não requiere um campo *Mod/Cod* e *PLP* único não requiere um identificador de *PLP*. Neste sinal *L1* com um número reduzido de bits, o sinal *L1* pode ser repetido três vezes no preâmbulo e modulação *BPSK* pode ser realizada, sendo assim possível uma sinalização muito robusta. Finalmente, o módulo de inserção de cabeçalho *ModCod* 307 pode inserir o cabeçalho gerado em cada quadro *FEC*. A figura 84 mostra um exemplo do módulo extractor *ModCod* r307 no trajecto de dados do módulo de desmodulação *BICM* apresentado na figura 64.

Tal como apresentado na figura 84, o cabeçalho *FECFRAME* pode ser analisado no analisador sintáctico (r301b), sendo que depois símbolos que transmitem informações idênticas em símbolos repetidos podem ser

atrasados, alinhados, e depois combinados no módulo de combinação *Rake* r302b. Finalmente, quando é realizada a desmodulação *BPSK* (r303b), o campo de sinal *L1* recebido pode ser restaurado, sendo que este campo de sinal *L1* restaurado pode ser enviado para o controlador de sistema para ser utilizado como parâmetros para descodificação. O *FECFRAME* analisado sintacticamente pode ser enviado para o desmapeador de símbolos.

A figura 78 mostra o desempenho de uma taxa de erro nos bits (*BER*) da protecção *L1* acima mencionada. Pode-se observar que cerca de 4,8 dB de ganho de *SNR* é obtido através de uma repetição por três vezes. A *SNR* necessária é de 8,7 dB a *BER* = 1E-11.

A figura 79 apresenta exemplos de um quadro de transmissão e estruturas de quadro *FEC*. As estruturas de quadro *FEC* apresentadas no lado superior direito da figura 79 representam o cabeçalho *FECFRAME* inserido pelo módulo de inserção 307 de cabeçalho *ModCod* apresentado na figura 37. Pode ser visto que, dependendo de várias combinações de condições, ou seja, tipo *CCM* ou *ACM/VCM* e *PLP* simples ou múltiplo, podem ser inseridas diferentes dimensões de cabeçalhos. Ou, nenhum cabeçalho pode ser inserido. Os quadros de transmissão formados de acordo com tipos de fatia de dados e apresentado na parte inferior do lado esquerdo da figura 79 pode ser formado pelo módulo de inserção 401 de cabeçalho de quadro do construtor de quadros tal como apresentado na figura 42 e o mistura-

dor/cortador 208 do processador de entrada apresentado na figura 35. Neste ponto, o *FECFRAME* pode ser transmitido de acordo com diferentes tipos de fatias de dados. Utilizando este processo, pode ser reduzida um máximo de 3,3% de informação complementar. No lado superior direito da figura 79 são apresentados quatro diferentes tipos de estruturas, mas um técnico irá compreender que estes são apenas exemplos, e qualquer um destes tipos ou as suas combinações podem ser utilizados para a fatia de dados.

No lado do receptor, o módulo removedor de cabeçalho de quadro r401 do módulo analisador sintáctico de quadros tal como apresentado na figura 63 e o módulo extractor *ModCod* r307 do módulo demod *BICM* apresentado na figura 64 pode extrair um parâmetro de campo *ModCod* que é necessário para descodificação. Neste ponto, de acordo com o tipo de fatia de dados de transmissão, podem ser extraídos parâmetros de quadro de transmissão. Por exemplo, para o tipo *CCM*, os parâmetros podem ser extraídos da sinalização *L1* que é transmitida no preâmbulo e para o tipo *ACM/VCM*, os parâmetros podem ser extraídas do cabeçalho *FECFRAME*.

Tal como apresentado no lado superior direito da figura 79, a estrutura *FECFRAME* pode ser dividida em dois grupos, em que o primeiro grupo são as estruturas dos três quadros superiores com cabeçalho e o segundo grupo é a última estrutura de quadro sem cabeçalho.

A figura 80 apresenta um exemplo de sinalização *L1* que pode ser transmitida dentro do preâmbulo pelo módulo de inserção 401 do cabeçalho do quadro do módulo de construção de quadro apresentado na figura 42. Esta sinalização *L1* é diferente da sinalização anterior *L1* em que o tamanho do bloco *L1* podem ser transmitido em bits (*L1_size*, 14 bits); é possível ligar/desligar a intercalação no tempo na fatia de dados (*dslice_time_intrlv*, 1 bit); sendo que ao definir o tipo de fatia de dados (*dslice_type*, 1 bit), é reduzida a informação complementar da sinalização. Neste ponto, quando o tipo de fatia de dados é *CCM*, o campo *Mod/Cod* pode ser transmitido dentro do preâmbulo em vez de dentro do cabeçalho *FECFRAME* (*PLP_mod* (3 bits), *PLP_FEC_type* (1 bit), *PLP_cod* (3 bits)).

No lado do receptor, o decodificador interno r303-1 encurtado/perfurado do *BICM* demod tal como apresentado na figura 64 pode obter o primeiro bloco *LDPC*, que tem um tamanho de bloco *L1* fixo, transmitido dentro do preâmbulo, através da descodificação. Podem também ser obtidos os números e a dimensão do resto dos blocos *LDPC*.

A intercalação no tempo pode ser utilizada quando vários símbolos *OFDM* são necessários para a transmissão de *L1* ou quando existe uma fatia de dados intercalada no tempo. Um ligar/desligar flexível da intercalação do tempo é possível com uma marca de intercalação. Para intercalação do tempo do preâmbulo, pode ser necessária uma marca de intercalação do tempo (1 bit) e podem ser necessários

vários símbolos *OFDM* intercalados (3 bits), deste modo, um total de 4 bits podem ser protegidos de um modo semelhante a um *FECFRAME* encurtado.

A figura 81 mostra um exemplo de pré-sinalização *L1* que pode ser executada no módulo de inserção de cabeçalho *ModCod* 307-1 no trajecto de dados do módulo *BICM* apresentado na figura 37. Os blocos com as linhas inclinadas e construtor de preâmbulo são exemplos do módulo de inserção de cabeçalho *ModCod* 307-1 na sinalização *L1* do módulo *BICM* apresentado na figura 37. Os blocos sólidos são exemplos do módulo de inserção 401 de cabeçalho de quadro do construtor de quadros tal como apresentado na figura 42.

Deste modo, os blocos sólidos podem ser exemplos do módulo de código interno 303-1 encurtado/puncionado, intercalador interno 304-1, demux de bit 305-1, e mapeador de símbolos 306-1 no trajecto de sinalização *L1* do módulo *BICM* apresentado na figura 37.

Tal como apresentado na figura 81, o sinal *L1* que é transmitido no preâmbulo pode ser protegido utilizando codificação *LDPC* encurtada/puncionada. Os parâmetros relacionados podem ser inseridos no cabeçalho numa forma de uma pré-sinalização *L1*. Neste ponto, somente parâmetros de intercalação do tempo podem ser transmitidos no cabeçalho do preâmbulo. Para garantir maior robustez, pode ser executada uma repetição de quatro vezes. No lado do receptor, para ser passível de descodificar o sinal *L1* que

é transmitido no preâmbulo, o módulo de extracção *ModCod* r307-1 no trajecto de sinalização *L1* do demod *BICM* tal como mostrado na figura 64 precisa de utilizar o módulo de descodificação apresentado na figura 84. Neste ponto, porque existe uma repetição de quatro vezes ao contrário do anterior cabeçalho *FECFRAME* de descodificação, é necessário um processo de recebimento de *Rake* que sincroniza os símbolos repetidos quatro vezes e adicionando os símbolos.

A figura 82 apresenta uma estrutura do bloco de sinalização *L1* que é transmitido do módulo de inserção 401 do cabeçalho do quadro do módulo de construção de quadro tal como apresentado na figura 42. É apresentado um caso onde não é utilizada qualquer intercalação no tempo num preâmbulo. Tal como apresentado na figura 82, podem ser transmitidos diferentes géneros de blocos *LDPC* na ordem das portadoras. Assim que estiver formado e transmitido um símbolo *OFDM* é formado e transmitido um símbolo *OFDM* seguinte. Para que o último símbolo *OFDM* seja transmitido, se sobrar alguma portadora, estas portadoras podem ser utilizadas para a transmissão de dados ou podem ser colocadas em zonas de preenchimento fictícias. O exemplo na figura 82 mostra um preâmbulo que compreende três símbolos *OFDM*. No lado do receptor, para este caso de não intercalação, o desintercalador de símbolos r308-1 no trajecto de sinalização *L1* do demod *BICM* tal como apresentado na figura 64 pode ser ignorado.

A figura 83 mostra um caso onde é executada a

intercalação no tempo *L1*. Tal como apresentado na figura 83, a intercalação em blocos pode ser executada de modo a formar um símbolo *OFDM* para índices de portadora idênticos, formando depois um símbolo *OFDM* para os próximos índices de portadora. Tal como no caso onde não é executada qualquer intercalação, se sobrar alguma portadora, estas portadoras podem ser utilizadas para a transmissão de dados ou podem ser colocadas em zonas de preenchimento fictícias. No lado do receptor, para este caso de não intercalação, o desintercalador de símbolos *r308-1* no trajecto de sinalização *L1* do demod *BICM* apresentado na figura 64 pode executar a desintercalação em blocos lendo blocos *LDPC* em ordem crescente de números de blocos *LDPC*.

Adicionalmente, podem existir pelo menos dois tipos de fatias de dados. A fatia de dados do tipo 1 tem *dslice_type* = 0 nos campos de sinalização *L1*. Este tipo de fatia de dados não tem nenhum cabeçalho de *XFECFrame* e tem os seus valores *Mod/Cod* em campos de sinalização *L1*. A fatia de dados do tipo 2 tem *dslice_type* = 1 nos campos de sinalização *L1*. Este tipo de fatia de dados tem cabeçalho de *XFECFrame* e tem os seus valores *Mod/Cod* em cabeçalho de *XFECFrame*.

XFECFrame significa *XFEC(complex Forward Error correction)Frame* e *Mod/Cod* significa tipo de modulação /taxa de código (*modulation type/coderate*).

Num receptor, um analisador sintáctico de quadro

pode construir um quadro de sinais desmodulados. O quadro apresenta símbolos de dados e os símbolos de dados podem ter um primeiro tipo de fatias de dados que apresenta uma *XFECFrame* e um cabeçalho de *XFECFrame* e um segundo tipo de fatia de dados que apresenta *XFECFrame* sem cabeçalho de *XFECFrame*. Além disso, um receptor pode extrair um campo para indicar se é para executar o tempo desintercalação nos símbolos do preâmbulo ou não para executar desintercalação no tempo nos símbolos do preâmbulo, a partir do *L1* dos símbolos preâmbulo.

Num transmissor, um construtor de quadros pode construir um quadro. Os símbolos de dados compreendem um primeiro tipo de fatias de dados que apresentam uma *XFECFrame* e um cabeçalho de *XFECFrame* e um segundo tipo de fatias de dados que apresenta *XFECFrame* sem cabeçalho de *XFECFrame*. Adicionalmente, um campo para indicar se deve ou não executar a intercalação no tempo em símbolos preâmbulo para executar a intercalação no tempo em símbolos de preâmbulo pode ser inserido no *L1* dos símbolos de preâmbulo.

Por fim, para código encurtado/puncionado para o módulo de inserção 401 do cabeçalho do quadro do construtor de quadros apresentado na figura 42, pode ser determinada uma dimensão mínima de palavra-chave que pode obter ganho de codificação e pode ser transmitida num primeiro bloco *LDPC*. Desta forma, para o resto das dimensões de bloco *LDPC* podem ser obtidas a partir daquela dimensão de bloco *L1* transmitida.

A figura 85 mostra um outro exemplo de pré-sinalização *L1* que pode ser transmitida do módulo de inserção de cabeçalho *ModCod* 307-1 no trajecto de sinalização do módulo *BICM* apresentado na figura 37. A figura 85 é diferente da figura 81 em que o mecanismo de protecção da parte do cabeçalho foi modificado. Tal como apresentado na figura 85, a informação sobre a dimensão do bloco *L1*, *L1_size* (14 bits), não é transmitida no bloco *L1*, mas transmitida no cabeçalho. No cabeçalho, a informação sobre a intercalação no tempo de 4 bits pode também ser transmitida. Para um total de 18 bits de entrada, o código *BCH* (45, 18) que gera 45 bits são utilizados e copiados para os dois trajectos e, finalmente, a *QPSK* é mapeada. Para o trajecto *Q*, pode ser executada uma deslocação cíclica de 1 bit para ganho de diversidade, podendo ser executada a modulação *PRBS* de acordo com a palavra de sincronização. Um total de 45 símbolos *QPSK* podem ser emitidos destas entradas de trajecto *I/Q*. Neste ponto, se a profundidade da intercalação no tempo estiver definida como um número de preâmbulos que é necessário para transmitir o bloco *L1*, pode não ser necessário transmitir *L1_span* (3bits) que indica a profundidade da intercalação no tempo. Por outras palavras, só pode ser transmitida a marca (1 bit) de ligar/desligar intercalação no tempo. Num lado do receptor, ao verificar apenas um número de preâmbulos transmitidos, sem utilizar *L1_span*, pode ser obtida a profundidade da desintercalação no tempo.

A figura 86 apresenta um exemplo de agendamento de bloco de pré-sinalização $L1$ que é transmitido no preâmbulo. Se uma dimensão de informação $L1$ que pode ser transmitida num preâmbulo for N_{max} , quando a dimensão de $L1$ é menor do que N_{max} , um preâmbulo pode transmitir a informação. No entanto, quando a dimensão $L1$ é maior do que N_{max} , a informação $L1$ pode ser igualmente dividida de modo que o bloco secundário $L1$ dividido é menor do que N_{max} , então o bloco secundário $L1$ dividido pode ser transmitido num preâmbulo. Neste ponto, para uma portadora que não é utilizada porque a informação $L1$ é menor do que N_{max} , não são transmitidos quaisquer dados.

Em vez disso, tal como apresentado na figura 88, a potência das portadoras onde os blocos $L1$ são transmitidos, pode ser aumentada para manter uma potência do sinal total de preâmbulo idêntica à potência do símbolo de dados. O factor de reforço de potência pode variar dependendo da dimensão $L1$ transmitida e um transmissor e um receptor pode ter um valor definido deste factor de reforço de potência. Por exemplo, se forem utilizadas somente metade das portadoras totais, o factor de reforço de potência pode ser dois.

A figura 87 mostra um exemplo de pré-sinalização $L1$ onde é considerado um reforço da potência. Quando comparada com a figura 85, pode-se ver que a potência do símbolo $QPSK$ pode ser reforçada e enviada para o construtor de preâmbulo.

A figura 89 apresenta outro exemplo do módulo de extracção *ModCod* r307-1 no trajecto de sinalização *L1* do módulo demod *BICM* apresentado na figura 64. Do símbolo de preâmbulo de entrada, a sinalização *L1 FECFRAME* pode ser emitida para dentro do desmapeador de símbolos e somente a parte do cabeçalho pode ser decodificada.

Para o símbolo do cabeçalho de entrada, o desmapeamento *QPSK* pode ser executado e pode ser obtido o valor do relatório do rácio de probabilidade (*Log-Likelihood Ratio - LLR*). Para o trajecto *Q*, a desmodulação *PRBS* de acordo com a palavra de sincronização pode ser executada, podendo ser realizado para restauração um processo inverso da deslocação cíclica de 1-bit.

Estes dois valores alinhados de trajecto *I/Q* podem ser combinados, podendo ser obtido o ganho *SNR*. A saída de decisão difícil pode ser entrada para o decodificador *BCH*. O decodificador *BCH* pode restaurar 18 bits de *L1* pré a partir da entrada de 45 bits.

A figura 90 mostra uma parte contrária, extractor *ModCod* de um receptor. Quando comparado com a figura 89, o controlo da potência pode ser executado nos símbolos de entrada *QPSK* do desmapeador para restaurar do nível de potência reforçado pelo transmissor para o seu valor original. Neste ponto, o controlo de potência pode ser executado considerando um número de portadoras utilizadas

para a sinalização *L1* num preâmbulo e tomando o inverso do factor de reforço de potência obtido de um transmissor. O factor de reforço de potência configura a potência do preâmbulo e potência dos símbolos de dados idênticos uns aos outros.

A figura 91 apresenta um exemplo da pré-sincronização *L1* que pode ser executada no módulo de extracção *ModCod* r307-1 no trajecto de sinalização *L1* do módulo de desmodulação *BICM* apresentado na figura 64. Este é um processo de sincronização para obter uma posição inicial de cabeçalho num preâmbulo. Os símbolos de entrada podem ser *QPSK* desmapeado do que para a saída trajecto *Q*, um inverso de uma deslocação cíclica de 1 bit pode ser executada, podendo ser realizado o alinhamento. Dois valores de trajectos *I/Q* podem ser valores multiplicados e modulados por pré-sinalização *L1* podem ser desmodulados. Assim, a saída do multiplicador pode expressar apenas *PRBS* que é uma palavra de sincronização. Quando a saída é correlacionada com uma sequência conhecida *PRBS*, pode ser obtido um pico de correlação no cabeçalho. Deste modo, pode ser obtida uma posição inicial de cabeçalho num preâmbulo. Se necessário, o controlo de potência que é executado para restaurar o nível de potência original, tal como apresentado na figura 90, pode ser executado na entrada do desmapeador *QPSK*.

A figura 92 mostra um outro exemplo de campo de cabeçalho de bloco *L1* que é enviado para o módulo de

inserção de cabeçalho *ModCod* 307-1 no trajecto de sinalização *L1* do módulo *BICM* tal como apresentado na figura 37. Esta figura 92 é diferente da figura 85 em que *L1_span* que representa a profundidade da intercalação no tempo é reduzida a 2 bits e os bits reservados são aumentados em 1 bit. Um receptor pode obter parâmetro de intercalação no tempo do bloco *L1* do *L1_span* transmitido.

A figura 93 apresenta processos de divisão igual de um bloco *L1* em várias partes quanto um número de preâmbulos, inserindo depois um cabeçalho em cada um dos blocos *L1* divididos e em seguida atribuindo os blocos *L1* com cabeçalho inserido num preâmbulo. Isto pode ser executado quando uma intercalação do tempo é executada com vários preâmbulos onde o número de preâmbulos é maior do que um número mínimo de preâmbulos que é necessário para a transmissão de blocos *L1*. Isto pode ser executado no bloco *L1*, no trajecto de sinalização *L1* do módulo *BICM* tal como apresentado na figura 37. O resto das portadoras, após a transmissão de blocos *L1* podem ter padrões de repetição cíclica ao invés de ser munidas com zonas de preenchimento de zeros.

A figura 94 apresenta um exemplo do desmapeador de símbolos r306-1 do módulo de desmodulação *BICM*, tal como apresentado na figura 64. Para um caso onde os blocos *L1 FEC* são repetidos tal como apresentado na figura 93, cada ponto de partida do bloco *L1 FEC* pode ser alinhado, combinado (r301f), e depois desmapeado em *QAM* (r302f) para

obter ganho de diversidade e ganho de *SNR*. Neste ponto, o combinador pode incluir processos de alinhamento e adicionar cada bloco *L1 FEC* e dividir o bloco *L1 FEC* adicionado. Para o caso onde somente parte do último bloco *FEC* é repetido tal como apresentado na figura 93, apenas a parte repetida pode ser dividida em até um número de cabeçalho de bloco *FEC* e a outra parte pode ser dividida por um valor que é um a menos do que um número de cabeçalho de bloco *FEC*. Por outras palavras, o número divisor corresponde a um número de portadoras que é adicionado a cada portadora.

A figura 98 apresenta um outro exemplo de agendamento de bloco *L1*. A figura 98 é diferente da figura 93 em que, em vez de executar a zona de preenchimento a zeros ou repetição quando os blocos *L1* não enchem um símbolo *OFDM*, o símbolo *OFDM* pode ser preenchido com redundância de paridade executando menos punção em código encurtado/puncionado no transmissor. Por outras palavras, quando a punção de paridade (304c) é executada na figura 38, a taxa de código *eFEC*tiva pode ser determinada de acordo com o rácio de punção, deste modo, puncionando dado que menos bits têm que ter zonas de preenchimento zeros, a taxa de código *eFEC*tiva pode ser reduzida podendo ser obtido um melhor ganho de codificação. O módulo de despuncionagem de paridade r303a de um receptor tal como apresentado na figura 65 pode executar a despuncionagem considerando a redundância de paridade menos puncionada. Neste ponto, porque um receptor e um

transmissor podem ter informações da dimensão total do bloco *L1*, o rácio de punccionagem pode ser calculado.

A figura 95 apresenta um outro exemplo de campo de sinalização *L1*. A figura 95 é diferente da figura 74 em que, para um caso em que o tipo de fatia de dados é *CCM*, pode ser transmitido um endereço inicial (21 bits) da *PLP*. Isso pode possibilitar que o *FECFRAME* de cada *PLP* forme um quadro de transmissão, sem o *FECFRAME* estar alinhado com uma posição inicial de um quadro de transmissão. Deste modo, a informação complementar da zona de preenchimento, que pode ocorrer quando a largura de uma fatia de dados é estreita, pode ser eliminada. Um receptor, quando um tipo de fatia de dados é *CCM*, pode obter informação *ModCod* do preâmbulo no trajecto de sinalização *L1* do desmodulador *BICM* tal como apresentado na figura 64, em vez de obter a mesma do cabeçalho do *FECFRAME*. Além disso, mesmo quando ocorre uma passagem com velocidade num local aleatório do quadro de transmissão, a sincronização *FECFRAME* pode ser realizada sem atraso porque o endereço de início da *PLP* já pode ser obtido a partir do preâmbulo.

A figura 96 apresenta um outro exemplo de campos de sinalização *L1* que podem reduzir a informação complementar do endereçamento *PLP*.

A figura 97 apresenta os números de símbolos *QAM* que correspondem a um *FECFRAME* dependendo dos tipos de modulação. Neste ponto, um divisor comum maior do símbolo

QAM é 135, sendo que deste modo pode ser reduzida uma informação complementar de $\log_2(135)$ 7 bits. Deste modo, a figura 96 é diferente da figura 95 em que um número de bits de campo *PLP_start* pode ser reduzido de 21 bits para 14 bits. Este é um resultado de considerar 135 símbolos como um único grupo e endereçar o grupo. Um receptor pode obter um índice de portadora *OFDM* onde a *PLP* começa num quadro de transmissão após a obtenção do valor do campo *PLP_start* e multiplicá-lo por 135.

A figura 99 e figura 101 mostram exemplos de intercalador de símbolos 308 o qual pode intercalar no tempo símbolos de dados que são enviados do módulo de inserção de cabeçalho *ModCod* 307 no trajecto de dados do módulo *BICM* tal como apresentado na figura 37.

A figura 99 é um exemplo do intercalador de blocos que pode operar numa base de fatias de dados. O valor da linha quer dizer um número de células de carga útil em quatro dos símbolos *OFDM* dentro de uma fatia de dados. A intercalação com base no símbolo *OFDM* pode não ser possível porque o número de células pode variar entre células contíguas *OFDM*. A coluna valor *K* significa uma profundidade de intercalação no tempo, que pode ser de 1, 2, 4, 8 ou 16... A sinalização de *K* para cada fatia de dados pode ser executada dentro da sinalização *L1*. A intercalação de frequência 403 tal como apresentado na figura 42 pode ser executada antes da intercalação no tempo 308 tal como apresentado na figura 37.

A figura 100 mostra um desempenho de intercalação do intercalador no tempo tal como apresentado na figura 99. Supõe-se que um valor de coluna é 2, um valor de linha é 8, uma largura de fatia de dados é 12 células de dados, e que não se encontram na fatia de dados pilotos contínuos. A figura de topo na figura 100 é uma estrutura de símbolo *OFDM* quando a intercalação no tempo não é executada e a figura inferior da figura 100 é uma estrutura de símbolo *OFDM* quando a intercalação no tempo é executada. As células negras representam um piloto disperso e as células não-negras representam as células de dados. O mesmo tipo de células de dados representa um símbolo *OFDM*. Na figura 100 as células de dados que correspondem a um único símbolo *OFDM* são intercaladas em dois símbolos. É utilizada uma memória de intercalação que corresponde a oito símbolos *OFDM* mas a profundidade da intercalação corresponde a apenas dois símbolos *OFDM*, sendo que desse modo não é obtida a profundidade completa de intercalação.

A figura 101 é sugerida para alcançar uma profundidade de intercalação total. Na figura 101, as células negras representam pilotos dispersos e as células não-negras representam as células de dados. O intercalador no tempo tal como apresentado na figura 101 pode ser implementado numa forma de intercalador de bloco e pode intercalar fatias de dados. Na figura 101, um número de coluna, K representa uma largura de fatia de dados, um número de linha, N representa a profundidade da

intercalação no tempo e o valor, K pode ser valores aleatórios i.e., $K=1,2,3, \dots$. O processo de intercalação inclui escrever célula de dados de um modo de retorção da coluna e leitura na direcção da coluna, excluindo as posições piloto. Quer dizer, pode-se dizer que a intercalação é executada de um modo torcido de linha-coluna.

Além disso, num transmissor, as células que são lidas de um modo torcido em coluna da memória de intercalação correspondem a um único símbolo *OFDM* e as posições piloto dos símbolos *OFDM* podem ser mantidas enquanto se intercala as células.

Além disso, num transmissor, as células que são lidas de um modo torcido em coluna da memória de desintercalação correspondem a um único símbolo *OFDM* e as posições-piloto dos símbolos *OFDM* podem ser mantidas enquanto se desintercala no tempo as células.

A figura 102 mostra um desempenho de intercalação no tempo da figura 101. Para comparação com a figura 99, supõe-se que um número de colunas é 8, uma largura de fatia de dados é 12 células de dados, e que não se encontram na fatia de dados pilotos contínuos. Na figura 102, as células de dados que correspondem a um único símbolo *OFDM* são intercaladas em oito símbolos *OFDM*. Tal como apresentado na figura 102, é utilizada uma memória de intercalação que corresponde a oito símbolos *OFDM*, correspondendo a

profundidade da intercalação resultante a oito símbolos *OFDM*, sendo desde modo obtida a profundidade completa de intercalação.

O intercalador no tempo tal como apresentado na figura 101 pode ser vantajoso em que a profundidade total de intercalação pode ser obtida utilizando memória idêntica; a profundidade da intercalação pode ser flexível, ao contrário da figura 99; consequentemente, um comprimento de quadro de transmissão pode também ser flexível, i.e., as linhas não precisam ser múltiplos de quatro. Além disso, o intercalador no tempo utilizado para a fatia de dados, pode ser idêntico ao processo de intercalação utilizado para o preâmbulo e também pode ter afinidade com um sistema de transmissão digital que usa *OFDM* geral. Especificamente, o intercalador no tempo 308 tal como apresentado na figura 37 pode ser utilizado antes do intercalador de frequência 403 tal como apresentado na figura 42 ser utilizado. A respeito da complexidade de um receptor, nenhuma memória adicional pode ser necessária a não ser a lógica de controlo de endereço adicional que pode exigir uma complexidade muito pequena.

A figura 103 mostra um desintercalador de símbolos correspondente r308 num receptor. Pode executar a desintercalação depois de receber a saída do módulo removedor de cabeçalho de quadro r401. Nos processos de desintercalação comparados com a figura 99, os processos de escrita e leitura da intercalação de blocos encontram-se

invertidos. Ao utilizar informação de posição-piloto, o desintercalador no tempo pode executar a desintercalação virtual ao não escrever para ou ler a partir de uma posição piloto na memória do intercalador e ao escrever para ou ler a partir de uma posição de célula de dados na memória do intercalador. A informação desintercalada pode ser emitida para o módulo extractor *ModCod* r307.

A figura 104 mostra um outro exemplo de intercalação no tempo. Pode ser executada a escrita no sentido diagonal e a leitura linha a linha. Tal como na figura 101, a intercalação é realizada tendo em conta as posições piloto. A leitura e escrita não é realizada para as posições piloto, mas a memória de intercalação é acedida considerando apenas as posições da célula de dados.

A figura 105 mostra um resultado de intercalação utilizando o processo apresentado na figura 104. Quando comparadas com a figura 102, as células com os mesmos padrões encontram-se dispersas, não somente no domínio do tempo, mas também no domínio da frequência. Por outras palavras, a profundidade completa da intercalação pode ser obtida em ambos os domínios do tempo e frequência.

A figura 108 mostra um desintercalador de símbolos r308 de um receptor correspondente. A saída do módulo removedor do cabeçalho de quadro r401 pode ser desintercalada. Quando comparada com a figura 99, a desintercalação mudou a ordem de leitura e escrita. A

desintercalação no tempo pode utilizar informação da posição-piloto para realizar desintercalação virtual tal como não ser executada qualquer leitura ou escrita em posições-piloto mas de forma que a leitura ou a escrita possa ser realizada apenas em posições de células de dados. Os dados desintercalados podem ser emitidos para o módulo extractor *ModCod* r307.

A figura 106 mostra um exemplo do processo de endereçamento da figura 105. *NT* significa profundidade da intercalação no tempo e *ND* significa largura da fatia de dados. Supõe-se que um valor de linha, *N* é 8, uma largura de fatia de dados é 12 células de dados, e que não se encontram na fatia de dados pilotos contínuos. A figura 106 representa um processo de geração de endereços para escrever dados numa memória de intercalação no tempo, quando um transmissor realiza a intercalação no tempo. O endereçamento começa a partir de um primeiro endereço com endereço de linha (*Row Address - RA*) = 0 e endereço de coluna (*Column Address - CA*) = 0. A cada ocorrência de endereçamento, *RA* e *CA* são aumentados. Para *RA*, pode ser executada uma operação de módulo com os símbolos *OFDM* utilizados no intercalador no tempo. Para *CA*, pode ser executada uma operação de módulo com um número de transportadoras que corresponde a uma largura de fatia de dados. *RA* pode ser incrementado em 1 quando portadoras que correspondem a uma fatia de dados são gravadas numa memória. Escrever numa memória pode ser executada somente quando uma localização do endereço actual não é uma

localização de um piloto. Se a localização do endereço actual for uma localização de um piloto, somente o valor do endereço pode ser aumentado.

Na figura 106, um número de coluna, K representa a largura de fatia de dados, um número de linha, N representa a profundidade da intercalação no tempo e o valor, K pode ser valores aleatórios ou seja, $K=1,2,3, \dots$. O processo de intercalação pode incluir escrever células de dados num modo de torção da coluna e ler na direcção da coluna, excluindo as posições piloto. Por outras palavras, a memória de intercalação virtual pode incluir posições piloto mas posições-piloto podem ser excluídas na intercalação actual.

A figura 109 mostra a desintercalação, um processo inverso da intercalação no tempo tal como apresentado na figura 104. Escrever linha a linha e ler na direcção diagonal pode restaurar células para as suas sequências originais.

O processo de endereçamento utilizado num transmissor pode ser utilizado num receptor. O receptor pode escrever dados recebidos numa memória de desintercalação no tempo linha a linha e pode ler os dados escritos utilizando valores de endereço gerados e informação de localização piloto que podem ser gerados de forma semelhante àquele de um transmissor. Como uma forma alternativa, os valores de endereço gerados e informação

piloto que foram utilizados para a escrita podem ser utilizados para a leitura linha a linha.

Estes processos podem ser aplicados num preâmbulo que transmite *L1*. Dado que cada símbolo *OFDM* que compreende preâmbulo pode ter pilotos em locais idênticos, pode ser executada a intercalação referente a valores de endereço tendo em conta as localizações piloto ou intercalação referente a valores de endereço sem ter em conta os localizações piloto. Para o caso de se referir ao valores de endereço sem levar em conta as localizações piloto, o transmissor armazena de cada vez dados numa memória de intercalação no tempo. Para um caso deste género, um tamanho de memória necessário para executar a intercalação/desintercalação de preâmbulos num receptor ou num transmissor torna-se idêntico a uma série de células de carga útil existentes nos símbolos *OFDM* utilizados para a intercalação no tempo.

A figura 107 é outro exemplo de intercalador no tempo *L1*. Neste exemplo, a intercalação no tempo pode colocar portadoras em todos os símbolos *OFDM* enquanto que as portadoras estariam todas localizadas num único símbolo *OFDM* se não tiver sido executada qualquer intercalação no tempo. Por exemplo, para dados localizados num primeiro símbolo *OFDM*, a primeira portadora do primeiro símbolo *OFDM* estará localizada na sua localização original. A segunda portadora do primeiro símbolo *OFDM* estará localizada num índice de portadora do segundo símbolo *OFDM*. Por outras

palavras, a i -ésima portadora de dados que se encontra localizada no n -ésimo símbolo *OFDM* estará localizada num i -ésimo índice portador do $(i+n) \bmod N$ -ésimo símbolo *OFDM*, em que $i = 0, 1, 2, \dots$ número de portadora -1 , $n=0, 1, 2, \dots, N-1$, e N é um número de símbolos *OFDM* utilizados na intercalação no tempo $L1$. Neste processo de intercalação no tempo $L1$, pode-se dizer que a intercalação para todos os símbolos *OFDM* é executada de um modo distorcido tal como apresentado na figura 107. Apesar de as posições piloto não estarem ilustradas na figura 107, tal como mencionado acima, a intercalação pode ser aplicada a todos os símbolos *OFDM* incluindo aos símbolos piloto. Ou seja, pode-se dizer que a intercalação pode ser realizada para todos os símbolos *OFDM* sem considerar posições piloto ou independentemente de os símbolos *OFDM* serem símbolos piloto ou não.

Se um tamanho de um bloco *LDPC* utilizado em $L1$ for menor do que um tamanho de um único símbolo *OFDM*, as portadoras restantes podem ter cópias de peças do bloco *LDPC* ou podem estar preenchidas com zeros. Neste ponto, pode ser realizada uma intercalação simultânea como acima. De modo semelhante, na figura 104, um receptor pode executar a desintercalação armazenando todos os blocos utilizados na intercalação no tempo $L1$ numa memória e lendo os blocos na ordem em que foram intercalados, ou seja, na ordem de números escritos em blocos apresentados na figura 107.

Quando é utilizado um intercalador de bloco tal como apresentado na figura 103 são utilizadas duas memórias tampão. Especificamente, enquanto uma memória tampão se encontra a armazenar símbolos de entrada, símbolos de entrada anteriores podem ser lidos da outra memória tampão. Uma vez que estes processos tenham sido realizados por um bloco de intercalação de um símbolo, a desintercalação pode ser realizada comutando a ordem de leitura e escrita, para evitar conflitos de acesso à memória. Esta desintercalação do estilo "pingue-pongue" pode ter uma simples lógica de geração de endereço. No entanto, a complexidade do hardware pode ser aumentada quando se utilizam duas memórias tampão de intercalação de símbolos.

A figura 110 mostra um exemplo de um desintercalador de símbolos r308 ou r348-1 tal como apresentado na figura 64. Esta forma de realização da invenção proposta pode utilizar somente uma única memória tampão para realizar a desintercalação. Uma vez que um valor de endereço seja gerado pela lógica de geração de endereço, o valor do endereço pode ser emitido da memória tampão, podendo ser executada a operação de reposicionamento pode ser realizada através do armazenamento de um símbolo que é introduzido no mesmo endereço. Por estes processos, pode ser evitado um conflito de acesso à memória durante a leitura e escrita. Além disso, a desintercalação do símbolo pode ser realizada utilizando apenas uma única memória tampão. Podem ser definidos parâmetros para explicar esta regra de geração de endereço. Tal como

apresentado na figura 106, um número de linhas de uma memória de desintercalação pode ser definido como a profundidade da intercalação no tempo, D e um número de colunas da memória de desintercalação pode ser definido como largura da fatia de dados, W . Em seguida o gerador de endereços pode gerar os seguintes endereços.

i-ésima amostra no *j*-ésimo bloco, incluindo o piloto.

$i=0,1,2,\dots,N-1;$

$N=D*W;$

$C_{i,j}=i \bmod W;$

$T_w=((C_{i,j} \bmod D) * j) \bmod D;$

$R_{i,j}=((i \div W)+T_w) \bmod D;$

$L_{i,j}(1)=R_{i,j}*W+C_{i,j};$

Ou

$L_{i,j}(2)=C_{i,j}*D+R_{i,j};$

Os endereços incluem posições piloto, sendo que deste modo assume-se que os símbolos de entrada incluem posições piloto. Se os símbolos de entrada que incluem apenas símbolos de dados precisam ser processados, podem ser necessária a lógica de controlo adicional que ignora os endereços correspondentes. Neste ponto, i representa um índice de símbolo de entrada, j representa um índice de bloco de intercalação de entrada, e $N=D*W$ representa um comprimento de bloco de interligação. Operação *Mod* representa operação do módulo que emite o restante após a divisão. A operação *Div* representa a operação de divisão que emite o quociente após a divisão. $R_{i,j}$ e $C_{i,j}$ representam o endereço da linha e endereço da coluna da i -

ésima entrada de símbolo do j -ésimo bloco de intercalação, respectivamente. Tw representa valor de retorção da coluna para endereços onde os símbolos estão localizados. Por outras palavras, cada coluna pode ser considerada como uma memória tampão onde a retorção independente é realizada de acordo com valores Tw . Li, j representa um endereço quando a memória tampão individual é implementada numa memória sequencial de uma dimensão sequencial, e não de duas dimensões. Li, j podem ter valores de 0 a $(N-1)$. São possíveis dois processos diferentes. Li, j (1) é utilizado quando a matriz da memória se encontra ligada linha a linha e Li, j (2) é utilizado quando a matriz da memória se encontra ligada em coluna a coluna.

A figura 111 mostra um exemplo de endereços de linha e coluna para a desintercalação no tempo quando D é 8 e W é 12. J tem início a partir de $j=0$ e para cada valor j , uma primeira linha pode representar o endereço da linha e uma segunda linha pode representar o endereço da coluna. A figura 111 mostra apenas endereços dos primeiros 24 símbolos. Cada índice de coluna pode ser idêntico ao índice i do símbolo de entrada.

A figura 113 mostra um exemplo de um transmissor OFDM que utiliza fatias de dados. Tal como apresentado na figura 113, o transmissor pode incluir um trajecto PLP de dados, um trajecto de sinalização $L1$, um construtor de quadro, e uma parte de modulação OFDM. O trajecto PLP de dados é indicado por blocos com linhas horizontais e linhas

verticais. O trajecto de sinalização *L1* é indicado por blocos com linhas inclinadas. Os módulos de processamento de entrada 701-0, 701-N, 701-K, e 701-M podem incluir blocos e sequências de módulo de interface de entrada 202-1, módulo sincronizador de fluxo de entrada 203-1, módulo compensador de atraso 204-1, módulo de eliminação de pacotes nulos 205-1, codificador CRC 206-1, módulo de inserção de cabeçalho *BB* 207-1, e codificador *BB* 209 executado para cada *PLP* tal como apresentado na figura 35. Os módulos *FEC* 702-0, 702-N, 702-K, e 702-M podem incluir blocos e sequências de codificador externo 301 e codificador interno 303 tal como apresentado na figura 37. Um módulo *FEC* 702-*L1* utilizado no trajecto *L1* pode incluir blocos e sequências do codificador externo 301-1 e codificador interno 303-1 encurtado/puncionado tal como apresentado na figura 37. O módulo de sinal *L1* 700-*L1* *L1* pode gerar informação *L1* necessária para compreender um quadro.

Os módulos de intercalação de bits 703-0, 703-N, 703-K, e 703-M podem incluir blocos e sequências de intercalador interno 304 e desmultiplicador de bits 305 tal como apresentado na figura 37. O intercalador de bits 703-*L1* utilizado no trajecto *L1* pode incluir blocos e sequências de intercalador interno 304-1 e desmultiplicador de bits 305-1 tal como apresentado na figura 37. Os módulos mapeadores de símbolo 704-0, 704-N, 704-K, e 704-M podem executar funções idênticas às funções do mapeador de símbolos 306 apresentado na figura 37. O módulo mapeador de

símbolos 704-*L1* utilizado no trajecto *L1* pode executar funções idênticas às funções do mapeador de símbolos 306-1 apresentado na figura 37. Os módulos de cabeçalho *FEC* 705-0, 705-N, 705-K, e 705-M podem executar funções idênticas às funções do módulo de inserção de cabeçalho *ModCod* 307 apresentado na figura 37. O módulo de cabeçalho *FEC* 705-*L1* para o trajecto *L1* pode executar funções idênticas às funções do módulo de inserção de cabeçalho *ModCod* 307-1 apresentado na figura 37.

Os módulos mapeadores de fatia de dados 706-0 e 706-K podem agendar blocos *FEC* a fatias de dados correspondentes e podem transmitir os blocos agendados *FEC*, onde os blocos *FEC* correspondem a *PLPs* que se encontram atribuídos a cada fatia de dados. O bloco mapeador de preâmbulo 707-*L1* pode agendar blocos *FEC* de sinalização *L1* para preâmbulos. Blocos *FEC* de sinalização *L1* são transmitidos em preâmbulos. Os módulos de intercalação no tempo 708-0 e 708-K podem executar funções idênticas às funções do intercalador de símbolos 308 apresentado na figura 37 que pode intercalar fatias de dados. O intercalador no tempo 708-*L1* utilizado no trajecto *L1* pode executar funções idênticas às funções do intercalador de símbolos 308-1 apresentado na figura 37.

Alternativamente, o intercalador no tempo 708-*L1* utilizado no trajecto *L1* pode executar funções idênticas ao intercalador de símbolos 308-1 apresentado na figura 37, mas apenas nos símbolos de preâmbulo.

Os intercaladores de frequência 709-0 e 709-K podem executar a intercalação de frequência em fatias de dados. O intercalador de frequência 709-L1 utilizado no trajecto L1 pode executar a intercalação de frequência de acordo com a largura de banda do preâmbulo.

O módulo gerador piloto 710 pode gerar pilotos que são adequados para piloto contínuo (CP), piloto disperso (SP), bordo de fatia de dados, e preâmbulo. Um quadro pode ser formado (711) do agendamento da fatia de dados, preâmbulo, e piloto. O módulo *IFFT* 712 e blocos de módulo de inserção *GI* 713 podem executar funções idênticas às funções do módulo *IFFT* 501, e blocos de módulo de inserção *GI* 503 apresentados na figura 51, respectivamente. Por último, o módulo *DAC* 714 pode converter sinais digitais em sinais analógicos e os sinais convertidos podem ser transmitidos.

A figura 114 mostra um exemplo de um receptor *OFDM* que utiliza fatia de dados. Na figura 114, o sintonizador r700 pode executar as funções do módulo sintonizador/AGC r603 e as funções do módulo conversor descendente r602 apresentado na figura 61. O ADC r701 pode converter sinais analógicos recebidos em sinais digitais. O módulo sincronizador de tempo/frequência r702 pode executar funções idênticas às funções do módulo sincronizador de tempo/frequência r505 apresentado na figura 62. O módulo detector de quadro r703 pode executar funções idênticas às

funções do módulo detector de quadro r506 apresentado na figura 62.

Neste ponto, após a sincronização do tempo/frequência ter sido realizada, a sincronização pode ser melhorada utilizando preâmbulo em cada quadro que é enviado a partir do módulo detector de quadro r703 durante o processo de localização.

O módulo removedor de *GI* r704 e módulo *FFT* r705 podem executar funções idênticas às funções do módulo removedor *GI* r503 e o módulo *FFT* r502 apresentado na figura 62, respectivamente.

O módulo de estimativa de canal r706 e módulo de equalização de canal r707 podem executar uma parte de estimativa de canal e uma parte de equalização de canal do módulo de canal *Est/Eq* r501 tal como apresentado na figura 62. Analisador sintático de quadros r708 pode produzir uma fatia de dados e preâmbulo onde os serviços seleccionados por um utilizador são transmitidos. Os blocos assinalados por linhas inclinadas processam um preâmbulo. Os blocos assinalados por linhas horizontais que podem incluir *PLP* comum, processam fatias de dados. O desintercalador de frequência r709-*L1* utilizado no trajecto *L1* pode executar a desintercalação de frequência dentro da largura de banda do preâmbulo. O desintercalador de frequência r709 utilizado no trajecto da fatia de dados pode executar a desintercalação de frequência dentro da fatia de dados. O

descodificador de cabeçalho *FEC* r712-L1, desintercalador no tempo r710-L1, e desmapeador de símbolo r713 L1 utilizado no trajecto L1 pode executar funções idênticas às funções do módulo extractor *ModCod* r307-1, desintercalador de símbolo r308-1, e desmapeador de símbolo r306-1 apresentado na figura 64.

O desintercalador de bits r714-L1 pode compreender blocos e sequências de demux de bits r305-1 e desintercalador interno 304-1 tal como apresentado na figura 64. O descodificador *FEC* r715-L1 pode compreender blocos e sequências de módulo de codificador interno encurtado/puncionado r301-1 e descodificador externo r303-1 apresentado na figura 64. Neste ponto, a saída do trajecto L1 pode ser informação de sinalização L1 e pode ser enviada para um controlador de sistema para restaurar dados *PLP* que são transmitidos em fatias de dados.

O desintercalador no tempo r710 utilizado no trajecto da fatia de dados pode executar funções idênticas às funções do desintercalador de símbolos r308 apresentado na figura 64. O analisador de fatia de dados r711 pode emitir *PLP* seleccionado pelo utilizador das fatias de dados e, se necessário, *PLP* comum associado à *PLP* seleccionada pelo utilizador. Os descodificadores de cabeçalho *FEC* r712-C e r712-K podem executar funções idênticas às funções do módulo extractor *ModCod* r307 apresentado na figura 64. Os desmapeadores de símbolos r713-C e r713-K podem executar funções idênticas às funções do desmapeador de símbolos r306 apresentado na figura 64.

O desintercalador de bits r714-C e r714-K pode compreender blocos e sequências de desmultiplicação de bits r305 e desintercalador interno r304 tal como apresentado na figura 64. Os descodificadores *FEC* r715-C e r715-K podem compreender blocos e sequências do descodificador interno r303 e descodificador externo r301 tal como apresentado na figura 64. Por fim, os módulos de processo de saída r716-C e r716-K podem compreender blocos e sequências de descodificador *BB* r209, módulo removedor de cabeçalho *BB* r207-1. Descodificador *CRC* r206-1, módulo de inserção de pacotes nulos r205-1, recuperação de atraso r204-1, recuperação de relógio de saída r203-1, e uma interface de saída r202-1 que são executados para cada *PLP* na figura 35. Se for utilizada uma *PLP* comum, a *PLP* comum e *PLP* de dados associada à *PLP* comum podem ser transmitidas para um recombinaor *TS* e podem ser transformadas numa *PLP* seleccionada por utilizador.

Deverá ser notado da figura 114, que num receptor, os blocos no trajecto *L1* não se encontram simetricamente sequenciados para um transmissor ao contrário ao trajecto de dados onde os blocos se encontram simetricamente posicionados ou na sequência oposta de um transmissor. Por outras palavras, para o trajecto de dados, encontram-se posicionados o desintercalador de frequência r709, desintercalador no tempo r710, analisador sintático de fatia de dados r711, e descodificador de cabeçalho *FEC* r712-C e r712-K. Contudo, para o trajecto *L1*, encontram-se

posicionados o desintercalador de frequência $r709-L1$, módulo de descodificação de cabeçalho $FEC\ r712-L1$, e desintercalador no tempo $r710-L1$.

A figura 112 mostra um exemplo de intercalação geral de bloco num domínio de símbolo de dados onde os pilotos não são utilizados. Tal como apresentado na figura 112a, a memória de intercalação pode ser preenchida sem pilotos negros. Para formar uma memória rectangular, se necessário podem ser utilizadas células de preenchimento. Na figura 112a, as células de preenchimento encontram-se indicadas como células com linhas inclinadas. No exemplo, porque um piloto contínuo pode se sobrepor a um tipo de padrão piloto disperso, é necessário um total de três células de preenchimento durante quatro durações de símbolo *OFDM*. Finalmente, na figura 112b, é apresentado o conteúdo da memória intercalada.

Tal como na figura 112a, pode ser realizado a escrita linha a linha e executada a torção da coluna; ou a escrita de uma forma torcida desde o início. A saída do intercalador pode incluir a leitura linha a linha da memória. Os dados de saída que forma lidos podem ser colocados tal como apresentados na figura 112c quando é considerada a transmissão *OFDM*. Neste momento, por simplicidade, o intercalamento da frequência pode ser ignorado. Tal como apresentado na figura 112, a diversidade da frequência não é tão elevada como aquela da figura 106, mas é mantida num nível semelhante. Acima de tudo, pode ser

vantajoso na medida em que a memória necessária para executar a intercalação e desintercalação pode ser otimizada. No exemplo, o tamanho da memória pode ser reduzido de $W \cdot D$ para $(W-1) \cdot D$. Conforme a largura da fatia de dados se torna maior, o tamanho da memória pode ser ainda mais reduzida.

Para entradas de desintercalador no tempo, um receptor deve restaurar o conteúdo da memória tampão da memória numa forma da figura do meio da figura 112 considerando as células de zona de preenchimento. Basicamente, os símbolos *OFDM* podem ser lidos símbolo a símbolo e podem ser guardados linha a linha. Pode ser executada a distorção correspondente à torção da coluna. A saída do desintercalador pode ser saída numa forma de leitura linha a linha a partir da memória da figura 112a. Desta forma, quando comparado com o processo apresentado na figura 106, a informação complementar piloto pode ser minimizada e, conseqüentemente, a memória de intercalação/desintercalação pode ser minimizada.

A figura 115 mostra a intercalação no tempo (figura 115a) e desintercalação no tempo (figura 115b).

A figura 115a mostra um exemplo de um intercalador no tempo 708-*L1* para trajecto *L1* da figura 113. Tal como apresentado na figura 115a, a intercalação no tempo para o preâmbulo onde *L1* é transmitida, pode incluir células de dados *L1* de intercalação, excluindo os pilotos

que são geralmente transmitidos no preâmbulo. O processo de intercalação pode incluir a escrita de dados de entrada numa direcção diagonal (linhas sólidas) e leitura dos dados linha a linha (linhas a pontilhado), utilizando processos idênticos aos que são apresentados em referência à figura 106.

A figura 115b mostra um exemplo de um desintercalador no tempo $r712-L1$ no trajecto $L1$ tal como apresentado na figura 114. Tal como apresentado na figura 115b, para um preâmbulo onde $L1$ é transmitido, pode ser realizada a desintercalação da célula de dados $L1$, excluindo os pilotos que são regularmente transmitidos no preâmbulo. O processo de desintercalação pode ser idêntico ao processo apresentado na figura 109, onde os dados de entrada são escritos linha a linha (linhas sólidas) e lidos na diagonal (linha pontilhada). Os dados de entrada não incluem qualquer piloto, consequentemente, os dados de saída apresentam células de dados $L1$ que também não incluem piloto. Quando um receptor utiliza uma única memória tampão num desintercalador no tempo para o preâmbulo, a estrutura do gerador de endereços que apresenta uma memória desintercaladora tal como apresentado na figura 110 pode ser utilizada.

A desintercalação ($r712-L1$) pode ser realizada utilizando operações de endereço tal como se segue:

i -ésima amostra no j -ésimo bloco, incluindo o piloto
 $i=0,1,2, \dots, N-1;$

```

 $N = D * W;$ 
 $C_{i,j} = i \bmod W;$ 
 $T_w = ((C_{i,j} \bmod D) * j) \bmod D;$ 
 $R_{i,j} = ((i \div W) + T_w) \bmod D;$ 
 $L_{i,j}(1) = R_{i,j} * W + C_{i,j};$ 
Ou
 $L_{i,j}(2) = C_{i,j} * D + R_{i,j};$ 

```

Nas operações acima, um comprimento de uma linha, W é um comprimento de uma linha de uma memória de intercalação tal como apresentado na figura 115. O comprimento da coluna D é uma profundidade de intercalação no tempo de preâmbulo, que é um número de símbolos *OFDM* que são necessários para a transmissão de preâmbulos.

A figura 116 mostra um exemplo de formação de símbolos *OFDM* agendando pilotos e preâmbulos de entrada do construtor de quadros 711 tal como apresentado na figura 113. As células em branco formam um cabeçalho $L1$ que é um sinal de saída do cabeçalho $FEC\ 705-L1$ no trajecto $L1$, tal como apresentado na figura 113. As células cinzentas representam os pilotos contínuos para o preâmbulo que são gerados pelo módulo de geração de piloto 710 tal como apresentado na figura 113. As células com padrões representam as células de sinalização $L1$ que são um sinal de saída do mapeador de preâmbulos 707- $L1$ tal como apresentado na figura 113. A figura 116a representa símbolos *OFDM* quando a intercalação no tempo se encontra desligada e a figura 116b representa símbolos *OFDM* quando a

intercalação no tempo se encontra ligada. O cabeçalho *L1* pode ser excluído da intercalação no tempo porque o cabeçalho *L1* transmite um comprimento de campo de sinalização *L1* e uma informação de marca de ligar/desligar a intercalação no tempo. É porque o cabeçalho *L1* é adicionado antes da intercalação no tempo. Tal como referido acima, a intercalação no tempo é realizada excluindo células piloto. O restante das células de dados *L1* pode ser intercalado tal como apresentado na figura 115, podendo ser atribuído a subportadoras *OFDM*.

A figura 117 mostra um exemplo de intercaladores no tempo 708-0 708-K que podem intercalar símbolos de dados que são enviados dos mapeadores de fatias de dados 706-0 706-K no trajecto de dados de um transmissor *OFDM* utilizando fatia de dados apresentado na figura 113. O intercalamento no tempo pode ser realizado para cada fatia de dados. Símbolos intercalados no tempo podem ser emitidos em intercaladores de frequência 709-0~709-K.

A figura 117 mostra também um exemplo de um intercalador no tempo simples que utiliza uma única memória tampão. A figura 117a mostra uma estrutura de símbolos *OFDM* antes da intercalação no tempo. Os blocos com os mesmos padrões representam o mesmo tipo de símbolos *OFDM*. A figura 117b e figura 117c mostram uma estrutura de símbolos *OFDM* após a intercalação no tempo. O processo de intercalação no tempo pode ser dividido em Tipo 1 e Tipo 2. Cada tipo pode ser realizado como alternativa para símbolos pares e

símbolos ímpares. Um receptor pode executar a desintercalação em conformidade. Uma das razões para utilizar alternativamente o tipo 1 e tipo 2 é o de reduzir a memória necessária num receptor utilizando uma única memória tampão durante a desintercalação no tempo.

A figura 117b mostra uma intercalação no tempo utilizando a intercalação tipo 1. Os símbolos de entrada podem ser escritos na direcção diagonal descendente e pode ser lidos num sentido de fila. A figura 117c mostra uma intercalação no tempo utilizando a intercalação tipo 2. Os símbolos de entrada podem ser escritos na direcção diagonal descendente e pode ser lidos num sentido de fila. A diferença entre tipo 1 e tipo 2 é se uma direcção de escrita de símbolo de entrada é para cima ou para baixo. Os dois processos são diferentes num modo de escrever símbolos, contudo os dois processos são idênticos em termos de exibir profundidade de intercalação em tempo integral e diversidade de frequência integral. No entanto utilizar estes processos pode provocar um problema durante a sincronização num receptor devido à utilização de dois esquemas de intercalação.

Podem existir duas soluções possíveis. A primeira solução pode ser sinalizar 1 bit de um tipo de intercalação de um primeiro bloco de intercalação que surge em primeiro lugar após cada preâmbulo, através de sinalização *L1* de preâmbulo. Este processo é a execução de uma intercalação correcta através de sinalização. A segunda solução pode ser

a formação de um quadro para apresentar um comprimento de um número par de blocos de intercalação. Utilizando este processo, um primeiro bloco de intercalação de cada quadro pode ter um tipo idêntico, portanto, o problema de sincronização do bloco de intercalação pode ser resolvido. Por exemplo, o problema de sincronização pode ser resolvido aplicando a intercalação do tipo 1 a um primeiro bloco de intercalação e aplicando sequencialmente aos próximos blocos de intercalação dentro de cada quadro, terminando depois um último bloco de intercalação de cada quadro com a intercalação tipo 2. Este processo requer que um quadro seja composto por dois blocos de intercalação mas pode ser vantajoso em que não é necessária qualquer sinalização adicional tal como no primeiro processo.

A figura 122 mostra uma estrutura de um desintercalador no tempo $r710$ de um receptor apresentado na figura 114. A desintercalação no tempo pode ser realizada nas saída do desintercalador de frequência $r709$. O desintercalador no tempo da figura 122 representa um esquema de desintercalação que é um processo inverso de uma intercalação no tempo apresentado figura 117. A desintercalação, em comparação com a figura 117, terá uma maneira oposta de leitura e escrita. Por outras palavras, o desintercalador de tipo 1 pode escrever símbolos de entrada numa direcção de linha e pode ler os símbolos escritos no sentido diagonal descendente. O desintercalador de tipo 2 pode escrever símbolos de entrada no sentido diagonal descendente e pode ler os símbolos escritos no sentido da

fila. Estes processos podem permitir a escrita dos símbolos recebidos onde os símbolos são previamente lidos ao realizar um sentido de escrita dos símbolos do desintercalador do tipo 2 idêntico a um sentido de símbolos de leitura de desintercalador do tipo 1. Deste modo, um receptor pode executar a desintercalação utilizando uma única memória tampão. Além disso, pode ser realizada uma implementação simples devido ao facto dos processos de desintercalação do tipo 1 e tipo 2 serem realizados por símbolos de escrita e leitura num sentido diagonal ou num sentido em linha.

Contudo utilizar estes processos pode provocar um problema na sincronização num receptor por causa da utilização de dois esquemas de intercalação. Por exemplo, desintercalando símbolos intercalados do tipo 1 de um modo do tipo 2 pode provocar a deterioração no desempenho. Podem existir duas soluções possíveis. A primeira solução pode ser determinar um tipo de um bloco de intercalação que surge após um preâmbulo, utilizando 1 bit de um tipo de intercalação de uma parte de sinalização *L1* transmitida. A segunda solução pode ser a realização da desintercalação utilizando um tipo de acordo com um primeiro bloco de intercalação dentro de um quadro, se um número de blocos de intercalação dentro de um quadro for um número par. O símbolo desintercalado pode ser emitido para dentro do analisador sintático de fatias de dados *r711*.

A figura 118 mostra uma lógica de geração de endereço que é idêntica a uma lógica de geração de endereço

de uma única memória tampão, quando um intercalador de blocos utiliza duas memórias tampão de memória tal como na figura 106. A lógica da geração de endereço pode executar funções idênticas às funções apresentadas na figura 106. Ao definir uma profundidade D de intercalação no tempo como um número de linhas de uma memória de desintercalação e definindo uma largura W de fatia de dados como um número de coluna, os endereços apresentados na figura 118 podem ser gerados por um gerador de endereços. Os endereços podem incluir posições piloto. Para intercalar no tempo os símbolos de entrada que incluem apenas símbolos de dados, pode ser necessária uma lógica de controlo que ignora os endereços. Os endereços utilizados nos preâmbulos de intercalação podem não exigir posições piloto e a intercalação pode ser realizada utilizando blocos $L1$. O i representa um índice de um símbolo de entrada, $N=D*W$ representa um comprimento de bloco de interligação. R_i e C_i representa um endereço de linha e um endereço de coluna de um i -ésimo símbolo de entrada, respectivamente. T_w representa um valor de retorção de coluna ou parâmetro de retorção de um endereço onde os símbolos se encontram localizados. L_i representa endereços quando uma memória dimensional com uma única memória tampão é implementada. Os valores de L_i podem ser de 0 a $(N-1)$. Nesta memória unidimensional são possíveis pelo menos dois processos. $L_i(1)$ encontra-se a acoplar uma matriz de memória linha a linha e $L_i(2)$ encontra-se a acoplar uma matriz de memória coluna a coluna. Um receptor pode utilizar a lógica de geração de endereços na leitura de símbolos durante uma desintercalação.

A figura 119 mostra um outro exemplo de um preâmbulo. Para um caso em que é utilizado um símbolo *OFDM* que apresenta um tamanho de 4K-FFT na largura de banda de 7,61MHz e uma sexta portadora dentro de um símbolo *OFDM* e portadoras em ambas as extremidades são utilizadas como pilotos, pode-se assumir que o número de portadoras que podem ser utilizadas na sinalização *L1* sejam de 2840. Quando vários canais se encontram ligados, podem existir várias larguras de banda de preâmbulo. O número de portadoras pode mudar dependendo de um tipo de pilotos a ser utilizado, um tamanho de *FFT*, um número de canais ligados, e outros factores. Se um tamanho de um *L1_XFEC_FRAME* que inclui *L1_header* (H) que deve ser atribuído a um único símbolo *OFDM* e bloco *L1 FEC (L1_FEC1)* for menor do que um único símbolo *OFDM* ($5w-a-1$), *L1_XFEC_FRAME* incluindo *L1_header* pode ser repetido para preencher uma parte restante do único símbolo *OFDM* ($5w-a-2$). Isto é semelhante à estrutura do preâmbulo da figura 93. Para que um receptor receba uma fatia de dados que se encontra localizada numa determinada largura de banda de canais ligados, uma janela de sintonizador do receptor pode estar localizada numa determinada largura de banda.

Se uma janela de sintonizador de um receptor se encontrar localizada como $5w-a-3$ da figura 119, pode ocorrer um resultado incorrecto durante a fusão de *L1_XFEC_FRAMES* repetidos. O caso 1 da figura 119 pode ser um tal exemplo. Um receptor encontra *L1_Header* (H) para

localizar uma posição inicial de um *L1_Header* (H) dentro de uma janela de sintonizador, mas o *L1_Header* encontrado pode ser um cabeçalho de um *L1_XFEC_FRAME* incompleto (5w-a-4). A informação de sinalização *L1* pode não ser obtida correctamente se for obtido um comprimento de *L1_XFEC_FRAME* com base naquele *L1_Header* e um resto da parte (5w-a-5) for adicionado a uma posição inicial daquela da *L1_Header*. Para evitar um caso do género, um receptor pode precisar de operações adicionais para encontrar um cabeçalho de um *L1_XFEC_FRAME* completo. A figura 120 mostra tais operações. No exemplo, para encontrar um cabeçalho de um *L1_XFEC_FRAME* completo, se existir um *L1_XFEC_FRAME* incompleto num preâmbulo, um receptor pode utilizar pelo menos dois *L1_Headers* para encontrar um local de início de *L1_Header* para fusão de *L1_XFEC_FRAME*. Primeiro, um receptor pode encontrar *L1_Header* a partir de um símbolo *OFDM* de preâmbulo (5w-b-1). Em seguida, utilizando um comprimento de um *L1_XFEC_FRAME* dentro do *L1_Header* encontrado, o receptor pode verificar se todos os *L1_XFEC_FRAME* dentro de um símbolo *OFDM* actual é um bloco completo (5w-b-2). Se não for, o receptor pode encontrar outro *L1_Header* do símbolo de preâmbulo actual (5w-b-3). A partir de uma distância calculada entre um *L1_Header* recém descoberto e um *L1_Header* anterior, pode ser determinado se um certo *L1_XFEC_FRAME* é um bloco completo (5w-b-4). Então, pode ser utilizado um *L1_Header* de um *L1_XFEC_FRAME* completo como um ponto de exposição para fusão. Usando o ponto de exposição, *L1_XFEC_FRAME* pode ser integrado (5w-b-5). Utilizando estes processos, caso 2 ou fusão correcta apresentada na

figura 119 podem ser esperados num receptor. Estes processos podem ser realizados no descodificador de cabeçalho *FEC* r712-L1 no trajecto do sinal *L1* da figura 114.

A figura 121 é um exemplo de uma estrutura de preâmbulo que pode eliminar as operações adicionais acima descritas num receptor. Ao contrário da estrutura de preâmbulo anterior, quando uma parte restante de um símbolo *OFDM* se encontra preenchida, apenas *L1_FEC1* de um *L1_XFEC_FRAME*, excluindo *L1_Header* (H) pode ser várias vezes cheia (5w-c-2). Deste modo, quando um receptor encontra uma posição de início de um *L1_Header* (H) para fundir *L1_XFEC_FRAME*, só pode ser encontrado (5w-c-4) *L1_Header* de somente *L1_XFEC_FRAME* completo, deste modo, sem operações adicionais, *L1_XFEC_FRAME* pode ser fundido utilizando o *L1_Header* encontrado. Por isso, os processos tais como 5w-b-2), 5w-b-3, e 5w-b-4 apresentados na figura 120 podem ser eliminados num receptor. Estes processos e processos da parte contrária dos processos podem ser executados no descodificador de cabeçalho *FEC* r712-L1 no trajecto do sinal *L1* de um receptor da figura 114 e no cabeçalho *FEC* 705 *L1* no trajecto do sinal *L1* de um transmissor da figura 113.

O desintercalador no tempo r712-L1 no trajecto *L1* de um receptor da figura 114 pode desintercalar células de bloco *L1* ou células com padrões, excluindo outras células tais como cabeçalho de preâmbulo e células piloto. As

células de bloco *L1* são representadas por células com padrões tal como apresentado na figura 116. A figura 123 mostra um outro exemplo de um transmissor *OFDM* que utiliza fatias de dados. Este transmissor pode ter estrutura idêntica e pode realizar função idêntica ao transmissor da figura 113, exceto os blocos adicionados e modificados. O mapeador de preâmbulo 1007-*L1* pode mapear blocos *L1* e cabeçalhos de bloco *L1* que são saídas de cabeçalho de *FEC* 705-*L1* em símbolos de preâmbulo utilizados num quadro de transmissão. Especificamente, o cabeçalho de bloco *L1* pode ser repetido para cada preâmbulo e o bloco *L1* pode ser dividido tanto quanto um número de preâmbulos utilizados. O intercalador no tempo 1008-*L1* pode intercalar blocos *L1* que são divididos em preâmbulos. Neste ponto, o cabeçalho de bloco *L1* pode ser incluído na intercalação ou não incluído na intercalação. Se o cabeçalho de bloco *L1* se encontra incluído ou não, pode não alterar mudar uma estrutura de sinal de um cabeçalho de bloco *L1*, mas pode alterar uma ordem de intercalação e transmissão de blocos *L1*. O módulo repetidor *L1_XFEC*, 1015-*L1*, pode repetir os blocos *L1_XFEC* intercalados no tempo dentro de uma largura de banda de preâmbulo. Neste ponto, o cabeçalho de bloco *L1* pode ser repetido dentro de um preâmbulo ou não repetido dentro de um preâmbulo.

A figura 124 mostra um outro exemplo de um receptor *OFDM* que utiliza fatias de dados. Este receptor apresenta estrutura idêntica e pode realizar função idêntica ao receptor da figura 114, excepto os blocos

adicionados e modificados. O descodificador de cabeçalho *FEC* r1012-L1 pode sincronizar cabeçalhos *L1* dentro de um preâmbulo. Se os cabeçalhos *L1* forem repetidos, os cabeçalhos *L1* podem ser combinados para obter um ganho SNR. Então, descodificador de cabeçalho *FEC* r712-L1 da figura 114 pode executar uma descodificação *FEC*. O processo de sincronização pode dar uma localização de um cabeçalho correlacionando a palavra de sincronização de um cabeçalho e preâmbulos. Para deslocamentos de frequência de múltiplos de um inteiro, pode ser determinada uma gama de correlação a partir de endereçamento circular.

O combinador r1017-L1 *L1_XFEC* pode combinar blocos *L1_XFEC* para obter um ganho *V*, quando blocos *L1* divididos são recebidos dentro de um preâmbulo. O desintercalador no tempo r1010-L1 pode desintercaladar no tempo blocos *L1* dentro de um preâmbulo. Dependendo se cabeçalhos de bloco *L1* são intercalados no tempo num transmissor ou não, os cabeçalhos de bloco *L1* podem ser desintercalados de modo correspondente num receptor. Uma ordem de desintercalação dos blocos *L1* pode ser alterada dependendo se os cabeçalhos de bloco *L1* são intercalados no tempo num transmissor ou não. Por exemplo, quando a intercalação no tempo se encontra ligada tal como na figura 116, uma localização da célula número 33 que é uma primeira célula de bloco *L1* dentro de um primeiro preâmbulo, pode mudar. Por outras palavras, quando os cabeçalhos de bloco *L1* não são incluídos numa intercalação, será recebido o sinal intercalado que tem as localizações das células tal

como apresentado na figura 116. Se os cabeçalhos de bloco *L1* forem incluídos numa intercalação, uma localização da célula número 33 precisa de ser alterada para desintercalar células que se encontram intercalados na diagonal, utilizando uma primeira célula de um primeiro cabeçalho de bloco *L1* dentro de um primeiro preâmbulo considerando como referência. O misturador *L1_FEC* *r1018-L1* pode misturar blocos *L1* que se encontram divididos em muitos preâmbulos num único bloco *L1* para descodificação *FEC*.

Com um 1 bit adicional, o campo *PLP_type* de campos de sinalização *L1* que são transmitidos num preâmbulo pode ter os seguintes valores.

PLP_type = 00 (*PLP* comum)

PLP_type = 01 (*PLP* de dados normais)

PLP_type = 10 (*PLP* de dados desmultiplexados)

PLP_type = 11 (reservado)

Uma *PLP* de dados normais representa uma *PLP* de dados quando um único serviço é transmitido numa única fatia de dados. Um *PLP* de dados desmultiplexados representa um *PLP* de dados quando um único serviço é desmultiplexado em várias fatias de dados. Quando um utilizador muda de serviço, se a sinalização *L1* e sinalização *L2* estiverem armazenados num receptor, pode ser eliminada a espera de uma informação de sinalização *L1* dentro de um próximo quadro. Por isso, um receptor pode alterar os serviços de forma eficiente e um utilizador pode beneficiar de menos

atraso durante uma troca de serviço. A figura 128 mostra as estruturas do sinal do bloco *L1* que é transmitido num preâmbulo, para o fluxo de intercalação no tempo e fluxo de desintercalação no tempo. Tal como se pode ver na figura 128, a intercalação e a desintercalação pode ser realizada não numa largura de banda completa de preâmbulo, mas num bloco *L1* dividido.

A figura 129 é um exemplo de um campo de intercalação no tempo *L1* de campos de sinalização *L1*, processado pelo módulo de cabeçalho *FEC 705-L1* no trajecto *L1* apresentado na figura 123. Tal como apresentado na figura 129, podem ser utilizados um bit ou dois bits para parâmetro de intercalação no tempo. Se for utilizado um bit, a intercalação não é executada quando o valor de bit é 0, sendo que a intercalação tendo uma profundidade de símbolos *OFDM* utilizados nos símbolos do preâmbulo pode ser realizada quando o valor do bit é 1. Se forem utilizados dois bits, a intercalação com uma profundidade de intercalação de 0 ou não intercalação é executada quando o valor de bit é 00, podendo a intercalação tendo uma profundidade de símbolos *OFDM* utilizados em símbolos de preâmbulo ser executada quando o valor do bit é 01. A intercalação tendo uma profundidade de quatro símbolos *OFDM* pode ser executada quando o valor do bit é 10. A intercalação que apresenta uma profundidade de oito símbolos *OFDM* pode ser executada quando o valor de bits é 11.

Um receptor, especificamente, decodificador de cabeçalho *FEC* r1012-*L1* no trajecto *L1* mostrado na figura 124 pode extrair parâmetros de intercalação no tempo (TI) apresentados na figura 129. Usando os parâmetros, o desintercalador no tempo r1010-*L1* pode realizar a desintercalação de acordo com a profundidade de entrelaçamento. Os parâmetros que são transmitidos no cabeçalho *L1* são o tamanho da informação *L1* (15bits), o parâmetro da intercalação no tempo (máximo de 2 bits), e CRC (max. 2 bits). Se for utilizado o código Reed-Muller RM (16, 32) para codificação do campo de sinalização do cabeçalho *L1*, porque os bits que podem ser transmitidos são 16 bits, não existe um número suficiente de bits. A figura 130 mostra um exemplo de campo de sinalização *L1* que pode ser utilizado para um tal caso e um processo de preenchimento.

A figura 130 mostra processos executados num módulo de cabeçalho *FEC* 705-*L1* no trajecto *L1* da figura 123. Na figura 130a, *L1* () nos campos de sinalização a coluna representa tamanho *L1* e TI() representa tamanho para os parâmetros de intercalação no tempo. Para o primeiro caso ou quando o tamanho *L1* (15 bits) e TI (1 bit) são transmitidos, o preenchimento adicional pode não ser necessário, podendo ser obtido um desempenho de descodificação substancial do cabeçalho *L1*, sendo que porque é transmitida a informação se se deseja realizar uma intercalação no tempo ou não, para um bloco *L1* curto não pode ser obtido o efeito de intercalação.

Para o segundo caso ou quando o tamanho *L1* é reduzido a 1/8 do tamanho original, torna-se possível a transmissão de informação com números de bits como *L1* (12 bits), *TI* (2 bits) e *CRC* (2 bits). Deste modo, para o segundo caso, pode ser esperado um melhor desempenho da descodificação *L1* e efeito de intercalação no tempo. No entanto, o segundo caso requiere processo de preenchimento adicional para fazer do tamanho *L1* um múltiplo de oito se o tamanho *L1* não for um múltiplo de oito. A figura 130b representa o processo de preenchimento que pode ser executado no sinal *L1* (700-*L1*) da figura 123. Mostra que o preenchimento se encontra localizado após o bloco *L1* e coberto com a codificação *CRC*. Consequentemente, num receptor, o módulo de descodificação *BCH/LDPC* *r715-L1* no trajecto *L1* da figura 124 pode executar a descodificação *FEC*, porque se não existir qualquer erro quando o campo *CRC* é verificado, pode ser executada a análise sintática de bit de acordo com o campo de sinalização *L1*, sendo então necessário um processo que define o resto dos bits como preenchimento ou *CRC32* e excluindo o resto de bits a partir de parâmetros.

Para o terceiro caso ou quando o tamanho *L1* é expresso como um número de células mapeadas *QAM*, não um número de bits, o número de bits pode ser reduzido. Para o quarto caso, o tamanho *L1* é expresso não como um tamanho de um bloco *L1* completo, mas como um tamanho *L1* por cada símbolo *OFDM*. Assim, para que um receptor obtenha um tamanho de um bloco *L1* completo é necessário multiplicar o

tamanho do bloco *L1* num símbolo *OFDM* por uma série de símbolos *OFDM* utilizados no preâmbulo. Neste caso, o tamanho *L1* actual tem que excluir o preenchimento.

Para o quinto caso, ao expressar o bloco *L1* não como um número de bits mas como um número de células mapeadas QAM, é possível maior redução de bits. Para o terceiro até ao quinto caso são apresentados os parâmetros *TI*, *CRC*, e um número de bits de preenchimento necessários. Para um caso em que o tamanho do bloco *L1* é expresso como um número de células, para que um receptor obtenha o tamanho *L1* em bits, o receptor precisa de multiplicar um número de bits onde apenas as células são transmitidas por um tamanho *L1* recebido. Adicionalmente, tem que ser excluído um número de bits de preenchimento.

O último caso mostra um número total crescente de bits para 32 bits utilizando dois blocos de código RM no cabeçalho. Campos *CRC* totais transformam-se em quatro bits porque cada bloco de código RM precisa de dois bits de campo *CRC*. Um receptor ou decodificador *r1012-L1* de cabeçalho *FEC* no trajecto *L1* da figura 124, precisa de obter os parâmetros necessários ao executar a decodificação *FEC* num total de dois blocos *FEC*. Utilizando os parâmetros obtidos, um receptor, em especial o desintercalador no tempo *r1010-L1* no trajecto *L1* da figura 124, pode determinar se é para realizar a desintercalação ou não, e pode obter uma profundidade de desintercalação, se for determinado que deve ser executada a desinter-

calação. Além disso, o módulo *BCH/LDPC* de descodificação *FEC* *r715-L1* pode obter o comprimento de bloco *LDPC* necessário para realizar a descodificação *FEC* e parâmetros de encurtamento/puncionagem. Os campos de preenchimento desnecessários exigidos para enviar o sinal *L1* para um controlador de sistema, podem ser removidos.

A figura 125 mostra um exemplo de um intercalador no tempo (TI) de fatias de dados. O processo de TI assume que todas as posições-piloto são conhecidas. O TI pode produzir apenas células de dados, excluindo os pilotos. Saber as posições piloto habilita o número correto de células de saída para cada símbolo *OFDM*. Além disso, o TI pode ser implementado por uma única memória tampão num receptor.

A figura 126 mostra um exemplo de uma implementação eficiente da intercalação no tempo num receptor. A figura 126a mostra quatro esquemas diferentes de desintercalação de acordo com uma forma de realização da presente invenção. A figura 126b mostra uma única memória tampão que realiza a desintercalação. A figura 126c mostra um exemplo de um esquema para endereçar os blocos *L1* numa matriz 2D ou uma sequência 1D.

Tal como apresentado na 126a-c, utilizar um algoritmo de memória tampão individual pode ser uma implementação mais eficiente do desintercalador no tempo. O algoritmo pode ser caracterizado primeiro pela leitura das

células de saída da memória, e depois escrever células de entrada onde as células de saída são lidas. O endereçamento diagonal pode ser considerado como um endereçamento circular em cada coluna.

Mais especificamente, em relação à figura 126a, estes quatro processos de leitura e escrita aplicam-se sequencialmente aos quadros C2 que são recebidos num receptor. O primeiro quadro recebido num receptor é escrito na memória do desintercalador na figura 126b no caminho para o bloco 0 na figura 126a e lido no caminho para o primeiro bloco. O segundo quadro recebido é escrito na memória do desintercalador na figura 126b no caminho para o primeiro bloco e lido para o segundo bloco. O terceiro quadro recebido é escrito na memória do desintercalador na figura 126b no caminho para o segundo bloco e lido no caminho para o terceiro bloco. O quarto quadro recebido é escrito na memória do desintercalador na figura 126b no caminho para o terceiro bloco e lido no caminho para o 0° bloco, etc. Quer dizer, os processos de escrita e leitura na figura 126a podem ser sequencialmente e ciclicamente aplicados aos quadros C2 que são recebidos sequencialmente.

O processo de intercalação no tempo (TI) pode ser executado em preâmbulos tal como apresentado na figura 127. As posições piloto são removidas periodicamente e facilmente, não sendo necessária qualquer intercalação para o cabeçalho do bloco L1. É porque o cabeçalho do preâmbulo carrega parâmetros TI e tanto a intercalação como a não-

intercalação têm os mesmos resultados devido a repetição. Deste modo, somente as células de sinalização *L1* são intercaladas. Pode ser aplicada a única memória tampão utilizada na fatia de dados TI.

A figura 128 mostra o fluxo do preâmbulo da intercalação/desintercalação no tempo. A intercalação pode ser realizada dentro de um bloco *L1*, em vez de um preâmbulo completo. Num transmissor, tal como apresentado na figura 128a, o bloco *L1* pode ser codificado ① então pode ser realizada uma intercalação dentro do bloco *L1* ②, e o bloco *L1* intercalado pode ser repetido dentro de um preâmbulo. Num receptor, tal como apresentado na figura 128b, de um preâmbulo recebido ①, o bloco *L1* pode ser combinado ou sincronizado, podendo ser obtido ② um único período de bloco *L1*, podendo o bloco *L1* combinado ser desintercalado ③.

A figura 129 mostra parâmetros de profundidade de intercalação no tempo em sinalização de cabeçalho *L1*. Para a estrutura de cabeçalho *L1*, RM (16, 32) tem uma capacidade de 16 bits. Um máximo de 2 bits de CRC pode melhorar o desempenho RM BER. Os campos de sinalização necessários do cabeçalho *L1* são *L1_info_size* (15 bits) que podem exigir um máximo de 5 símbolos *OFDM* e *TI_depth* (2 bits ou 1 bit). No entanto, um total de 18 ou 19 bits excede a capacidade do cabeçalho *L1*.

A figura 131 mostra um exemplo de uma sinalização

L1 transmitida num cabeçalho de quadro. A informação de sinalização *L1* pode ser utilizada como parâmetros de decodificação num receptor. Em especial, os módulos no trajecto do sinal *L1* da figura 124 pode executar decodificação de sinalização *L1*, sendo que os módulos no trajecto *PLP* da figura 124 pode usar parâmetros, portanto, os serviços podem ser decodificados. Um receptor pode obter parâmetros da sinalização *L1* a partir de sinais do trajecto *L1* que são decodificados de acordo com uma ordem de cada campo e comprimento de campo. O seguinte explica o significado de cada campo e a sua utilização. Um nome de cada campo, um número de bits para cada campo, ou um exemplo de cada campo podem ser modificados.

Num_chbon: Este campo indica um número de canais utilizados numa ligação de canal. Utilizando este campo, um receptor pode obter uma largura de banda total de canais utilizados. O canal pode ter 6MHz, 7MHz, 8MHz, ou outros valores de largura de banda. *Num_dslice*: Este campo indica um número de fatias de dados existentes num canal ligado. Depois da decodificação da sinalização *L1*, um receptor acede a um circuito fechado onde a informação das fatias de dados se encontra contida, para obter a informação da fatia de dados. Utilizando este campo, um receptor pode obter um tamanho do circuito fechado para decodificação.

Num_notch: Este campo indica um número de bandas de entalhe existentes num canal ligado. Depois da decodificação da sinalização *L1*, um receptor acede a um

circuito fechado onde a informação da banda de entalhe se encontra contida, para obter a informação da banda de entalhe. Utilizando este campo, um receptor pode obter um tamanho do circuito fechado para descodificação.

Para cada fatia de dados, podem ser transmitidos *dslice_id*, *dslice_start*, *dslice_width*, *dslice_ti_depth*, *dslice_type*, *dslice_pwr_allocation* e informações *PLP* num preâmbulo de um cabeçalho de quadro. A fatia de dados pode ser considerada como uma largura de banda específica que contém um ou mais *PLPs*. Os serviços podem ser transmitidos nos *PLPs*. Um receptor precisa de aceder às fatias de dados que contêm um *PLP* específico, para descodificar um serviço.

Dslice_id: Este campo pode ser utilizado para identificação de fatia de dados. Cada fatia de dados num canal ligado pode ter um valor único. Quando um receptor acessa uma das *PLPs* para descodificar serviços, este campo pode ser utilizado para o receptor diferenciar uma fatia de dados onde a *PLP* se encontra localizada, de outras fatias de dados.

Dslice_start: Este campo indica um local de início de uma fatia de dados dentro de um canal ligado. Utilizando este campo, um receptor pode obter uma frequência onde a fatia de dados começa. Adicionalmente, a sintonização para aceder a uma fatia de dados pode ser executada utilizando este campo.

Dslice_width: Este campo indica uma largura de banda de uma fatia de dados. Utilizando este campo, um receptor pode obter um tamanho de uma fatia de dados. Em especial, este campo pode ser utilizado em desintercalação no tempo permitir a descodificação. Juntamente com o campo *dslice_start*, um receptor pode determinar que frequência descodificar dos sinais *RF* recebidos. Este processo pode ser executado em sintonizador r700 da figura 124. As informações tais como *dslice_start* e *dslice_width* podem ser utilizadas como sinal de controlo do sintonizador r700.

Dslice_ti_depth: Este campo indica uma profundidade de intercalador no tempo utilizada na intercalação no tempo de fatias de dados. Juntamente com *dslice_width*, um receptor pode obter uma largura e uma profundidade de um desintercalador no tempo e pode realizar a desintercalação no tempo. A figura 132 mostra um exemplo de um *dslice_ti_depth*. No exemplo, 1, 4, 8 ou 16 de símbolos *OFDM* são utilizados na intercalação no tempo. Isto é executado no desintercalador no tempo r710 da figura 124. *Dslice_width* e *dslice_ti_depth* podem ser utilizados como sinal de controlo.

Dslice_type: Este campo indica um tipo de fatia de dados. Fatia de dados do tipo 1 tem um único *PLP* dentro do mesmo e o *PLP* é uma *CCM* (*constant coding and modulation* - codificação e modulação constantes) aplicada. A fatia de dados do tipo 2 representa todos os outros géneros de fatias de dados. Utilizando este campo, um receptor pode

realizar a decodificação de acordo com *PLP*. A *PLP* do tipo 1 não tem cabeçalho *FECFRAME*, sendo que assim um receptor não procura pelo cabeçalho *FECFRAME*. Para o tipo 2, um receptor procura pelo cabeçalho *FECFRAME* da *PLP* para obter informação *MODCOD*. A figura 133 mostra um exemplo de um *dslice_type*. Utilizando este campo, o analisador sintático de fatias de dados r711 da figura 124 pode controlar os descodificadores de cabeçalho *FEC* r712-c, k.

Dslice_pwr_allocation: Este campo indica uma potência de uma fatia de dados. Cada fatia de dados pode ter uma potência diferente de outras fatias de dados. É para adaptação da ligação no sistema de cabo. Um receptor pode utilizar este campo para controlar a potência da fatia de dados recebida. O sintonizador r700 da figura 124 pode ajustar o ganho do sinal utilizando este campo.

Num_PLP: Este campo indica um número de *PLPs* numa fatia de dados. Após a descodificação da sinalização *L1*, um receptor acede a um circuito fechado que inclui informação *PLP*. Utilizando este campo, um receptor pode obter um tamanho do circuito fechado e descodificar *PLPs*. Para cada *PLP*, *PLP_id*, *PLP_type*, reproprocessamento *PSI/SI*, *PLP_payload_type*, *PLP_modcod* e *PLP_start_addr* podem ser transmitidos num cabeçalho de quadro (preâmbulo). Cada *PLP* pode transmitir um ou mais fluxos ou pacotes tais como *TS* e *GSE*. Um receptor pode obter serviços descodificando *PLPs* onde os serviços são transmitidos.

PLP_id: Este campo é um identificador de *PLP* e tem um valor único para cada *PLP* num canal ligado. Utilizando este campo, um receptor pode aceder à *PLP* onde existe um serviço para descodificar. Este campo pode servir a um propósito idêntico à *PLP_id* transmitida num cabeçalho *FECFRAME*. Os descodificadores de cabeçalho *FEC r712-c, k* da figura 124 podem aceder à *PLP* necessária utilizando este campo.

PLP_type: Este campo indica se um tipo de *PLP* é uma *PLP* comum ou uma *PLP* de dados. Utilizando este campo, um receptor pode encontrar *PLP* comum e pode obter informação necessária para decodificar um pacote *TS* da *PLP* comum. Além disso, o receptor pode descodificar um pacote *TS* dentro de uma *PLP* de dados. A figura 134 mostra um exemplo de um *PLP_type*.

Reprocessamento *PSI/SI*: Este campo indica se um *PSI/SI* de um sinal recebido é reprocessado ou não. Utilizando este campo, um receptor pode determinar referir *PSI/SI* de um serviço específico de um serviço transmitido. Se um receptor não pode referir *PSI/SI* de um serviço específico de um serviço transmitido, *PSI/SI* que pode ser referido por um serviço específico pode ser transmitido através de *PLP* comum, por exemplo. Utilizando esta informação, um receptor pode descodificar serviços.

PLP_payload_type: Este campo indica o tipo de dados de carga útil que a *PLP* transmite. Um receptor pode

utilizar este campo antes de descodificar dados dentro de *PLPs*. Se um receptor não pode descodificar um tipo específico de dados, pode ser impedida a descodificação de uma *PLP* que contém este tipo específico de dados. A figura 135 mostra um exemplo de um *PLP_payload_type*. Se uma fatia de dados tem uma única *PLP* e uma *CCM* é aplicada à fatia de dados fatia, isto é, fatia de dados tipo 1, os campos tais como *PLP_modcod* e *PLP_start_addr* podem ser adicionalmente transmitidos.

PLP_modcod: Este campo indica um tipo de modulação e taxa de código *FEC* utilizado na *PLP*. Utilizando este campo, um receptor pode realizar a desmodulação *QAM* e descodificação *FEC*. A figura 136 mostra um exemplo de um *PLP_modcod*. Os valores apresentados na figura 136 podem ser utilizados em *modcod* que é transmitido num cabeçalho de um *FECFRAME*. O desmapeador de símbolos *r713-c, k* e módulo descodificador *FEC BCH/LDPC r715-c, k* da figura 124 podem utilizar este campo para descodificação.

PLP_start_addr: Este campo indica onde um primeiro *FECFRAME* de uma *PLP* aparece num quadro de transmissão. Utilizando este campo, um receptor pode obter uma localização inicial de *FECFRAME* e realizar a descodificação *FEC*. Utilizando este campo, o analisador sintático de fatias de dados *r711* da figura 127 pode sincronizar *FECFRAMEs* para *PLPs* do tipo 1. Para cada banda de entalhe, as informações tais como *notch_start* e *notch_width* podem ser transmitidas num cabeçalho de quadro (preâmbulo).

Notch_start: Este campo indica um local de início de uma banda de entalhe. *Notch_width*: Este campo indica uma largura de uma banda de entalhe. Utilizando *notch_start* e *notch_width*, um receptor pode obter um local e um tamanho de uma banda de entalhe dentro de um canal ligado. Além disso, pode ser obtido um local de sintonia para uma decodificação de serviço correcta e pode ser verificada uma existência de um serviço dentro de uma determinada largura de banda. O sintonizador r700 da figura 124 pode executar a sintonia utilizando esta informação.

GI: Este campo indica a informação do intervalo de guarda utilizado num sistema. Um receptor pode obter informação de intervalo de guarda utilizando este campo. O módulo r702 de sincronização de hora/freq e módulo removedor de *GI* r704 da figura 124 pode usar este campo. A figura 137 mostra um exemplo.

Num_data_symbols: Este campo indica um número de símbolos de dados *OFDM*, exceto preâmbulo, utilizados num quadro. Um comprimento de quadro de transmissão pode ser definido por este campo. Utilizando este campo, um receptor pode prever a localização de um preâmbulo a seguir, assim, este campo pode ser utilizado para decodificar a sinalização *L1*. O analisador sintático de quadros r708 da figura 124 pode utilizar este campo e prever símbolos *OFDM* que são preâmbulo e enviar sinal para o trajecto de decodificação de preâmbulo.

Num_c2_frames: Este campo indica um número de quadros existentes num canal ligado. Utilizando este campo, um receptor pode obter um limite de um super quadro e pode prever informação repetida por cada super quadro.

Frame_idx: Este campo é um índice de quadros e é reinicializado para cada super quadro. Utilizando este campo, um receptor pode obter um número actual de quadros e encontrar uma localização do quadro actual dentro de um super quadro. Usando este campo, o analisador sintático de quadros r708 da figura 124 pode descobrir quantos quadros estão à frente de um quadro actual num super quadro. Juntamente com *num_c2_frames*, pode ser prevista a mudança que ocorre numa sinalização *L1* e a descodificação pode ser controlada.

PAPR: Este campo indica se é utilizada ou não uma reserva de tom para reduzir uma PAPR. Utilizando este campo, um receptor pode processar em conformidade. A figura 138 mostra um exemplo. Por exemplo, se for utilizada uma reserva de tom, um receptor pode excluir portadoras utilizadas numa reserva de tom, a partir da descodificação. Especificamente, o analisador sintático de quadros r711 da figura 124 pode utilizar este campo para excluir portadoras da descodificação.

Reservado: Este campo é para bits adicionais reservados para utilização futura.

A figura 139 mostra um outro exemplo de uma sinalização *L1* transmitida num cabeçalho de quadro. Na figura 139, a informação adicionada adicionalmente à figura 131 pode tornar a descodificação de serviço por um receptor mais eficiente. Os campos a seguir explicam apenas a informação adicional. Os outros campos são os mesmos que os da figura 131.

Network_id: Este campo indica uma rede onde o sinal transmitido pertence. Utilizando este campo, um receptor pode descobrir uma rede actual. Quando um receptor sintoniza para outra rede para encontrar um serviço na rede, o receptor pode processar mais rápido porque utilizando apenas a descodificação *L1* é o suficiente para tomar a decisão se a rede sintonizada é uma rede desejada ou não.

C2_system_id: Este campo identifica um sistema onde o sinal transmitido pertence. Utilizando este campo, um receptor pode descobrir o sistema actual. Quando um receptor sintoniza para outra rede para encontrar um serviço no sistema, o receptor pode processar mais rápido porque utilizar apenas a descodificação *L1* é o suficiente para tomar a decisão se a rede sintonizada é um sistema desejado ou não.

C2_signal_start_frequency: Este campo indica uma frequência inicial de canais ligados.

C2_signal_stop_frequency: Este campo indica uma frequência final de canais ligados. Utilizando *c2_signal_start_frequeny* e *c2_signal_stop_frequency*, as larguras de banda RF de todas as fatias de dados podem ser encontradas por descodificação de *L1* de determinada largura de banda dentro de canais ligados. Além disso, este campo pode ser utilizado para obter um valor de deslocamento de frequência exigido na sincronização de *L1_XFEC_FRAMEs*. O combinador *L1 XFEC r1017-L1* da figura 124 pode usar este campo. Além disso, quando um receptor recebe fatias de dados localizadas em ambas as extremidades de um canal ligado, este campo pode ser utilizado para sintonizar para uma frequência adequada. O sintonizador *r700* da figura 124 pode utilizar esta informação.

PLP_type: Este campo indica se um tipo de *PLP* é uma *PLP* comum, uma *PLP* de dados normal, ou *PLP* de dados agrupada. Utilizando este campo, um receptor pode identificar *PLP* comum e pode obter informação necessária para descodificar um pacote *TS* da *PLP* comum, depois pode descodificar o embalagem *TS* dentro de uma *PLP* de dados agrupada. A figura 140 mostra um exemplo deste campo. A *PLP* de dados normais é uma *PLP* de dados que não tem *PLP* comum. Neste caso, um receptor não precisa de encontrar uma *PLP* comum. A *PLP* comum ou *PLP* agrupada pode transmitir informação tal como *PLP_group_id*. Para os outros tipos de *PLP* é possível uma transmissão mais eficiente porque não é necessário transmitir nenhuma informação adicional.

PLP_group_id: Este campo indica um grupo onde uma *PLP* atual pertence. *PLP* de dados agrupados pode transmitir parâmetros *TS* comuns utilizando *PLP* comum. Utilizando este campo, se uma *PLP* atualmente descodificada é uma *PLP* agrupada, um receptor pode encontrar uma *PLP* necessária comum, obter parâmetros necessários para pacote *TS* de *PLP* agrupada, e formar um pacote *TS* completo.

Reserved_1/reserved_2/reserved_3: Estes campos são bits adicionais reservados respectivamente para utilização futura para um circuito fechado de fatia de dados, um circuito fechado *PLP*, e um quadro de transmissão.

A figura 141 mostra um outro exemplo de uma sinalização *L1* transmitida num cabeçalho de quadro. Comparado com a figura 139, mais informações otimizadas podem ser transmitidas, assim, pode ocorrer menos sinalização de informação complementar. Correspondentemente, um receptor pode descodificar serviços de forma eficiente. Em especial, os módulos no trajecto do sinal *L1* da figura 124 pode executar decodificação de sinalização *L1*, sendo que os módulos no trajecto *PLP* da figura 124 pode usar parâmetros, portanto, os serviços podem ser decodificados. Um receptor pode obter parâmetros da sinalização *L1* a partir de sinais do trajecto *L1* que são descodificados de acordo com uma ordem de cada campo e comprimento de campo. Um nome de cada campo, um número de bits para cada campo, ou um exemplo de cada campo podem ser modificados. As descrições dos campos, excepto *dslice_width* são idênticas

às descrições acima descritas dos campos. Uma função de *dslice_width* de acordo com um exemplo é a seguinte.

Dslice_width: Este campo indica uma largura de banda de uma fatia de dados Utilizando este campo, um receptor pode obter um tamanho de uma fatia de dados. Em especial, este campo pode ser utilizado em desintercalação no tempo permitir a descodificação. Juntamente com o campo *dslice_start*, um receptor pode determinar que frequência descodificar dos sinais *RF* recebidos. Este processo pode ser executado em sintonizador r700 da figura 124. As informações tais como *dslice_start* e *dslice_width* podem ser utilizadas como sinal de controlo do sintonizador r700. Neste ponto, a largura de uma fatia de dados pode-se estender até 64MHz usando 12 bits para este campo *dslice_width*. Usando este campo, um receptor pode determinar se um sintonizador atualmente disponível pode descodificar a fatia de dados atual. Se a largura de uma fatia de dados for maior do que uma largura de banda de um sintonizador herdado de um receptor, para descodificar uma tal uma fatia de dados, um receptor pode usar, pelo menos, dois sintonizadores herdados ou um sintonizador com uma largura de banda suficientemente grande. No exemplo, uma granularidade de valores utilizada no *dslice_start*, *dslice_width*, *notch_start* e *notch_width* pode ser 12 portadoras *OFDM* (células). Por outras palavras, um receptor pode encontrar um local de uma célula *OFDM* actual multiplicando os valores transmitidos por 12. No exemplo, para uma granularidade de *PLP_start_addr*, pode ser utilizada uma portadora *OFDM* (célula). Por outras palavras,

um receptor pode descobrir quantos símbolos *OFDM* e células *OFDM* estão à frente de um local de início de uma *PLP* dentro de um símbolo *OFDM*. *Dslice_start* e *dslice_width* podem ser usadas para esta finalidade. O analisador sintático de fatia de dados *r711* da figura 124 pode realizar tal processo.

A figura 142 mostra um exemplo de processos no módulo de cabeçalho *FEC 705-L1* no caminho *L1* da figura 123. A figura 142a mostra a estrutura de cabeçalho *FEC* e a figura 142b mostra exemplos de profundidade *TI* explicada na figura 129. Um total de 16 bits podem ser transmitidos no cabeçalho do *FEC* de um caminho *L1*. Quatorze bits podem ser atribuídos para *L1_info_size*. Se *L1_info_size* tiver um valor que é metade do comprimento do bloco *L1* transmitido actualmente, um receptor pode multiplicar *L1_info_size* recebido por dois e obter o comprimento actual do bloco *L1* e começar a decodificar *L1*. Este comprimento obtido do bloco *L1* é um comprimento que inclui zona de preenchimento.

Para o bloco *L1* que se encontra determinado em não apresentar qualquer erro apesar da verificação CRC, um receptor pode considerar o resto de bits após a decodificação *L1* como zona de preenchimento. Os dois últimos bits, semelhantes tal como nos processos anteriores, podem ser utilizados para indicar a profundidade da intercalação no tempo de preâmbulos. O mapeador de preâmbulo 1007-*L1* da figura 123 pode determinar os símbolos *OFDM* necessários para transmitir blocos *L1*. Após isto, o intercalador no tempo 1008-*L1* da figura 123 pode realizar a

intercalação no tempo. Usando a informação de profundidade de intercalação no tempo e *L1_info_size*, um receptor pode descobrir que tamanho do bloco *L1* é transmitido em quantos símbolos *OFDM*. Combinar, fundir, e desintercalar no tempo blocos *L1* pode ser realizado no combinador *L1 XFEC* 1017-*L1*, unificador *L1_FEC* 1018-*L1*, e desintercalador no tempo 1010-*L1* da figura 124, respectivamente.

Num receptor na figura 124, um comprimento de um bloco *L1 XFEC* dentro de um símbolo *OFDM* pode ser obtido dividindo um comprimento total de bloco *L1* por um número de símbolos *OFDM* usados num preâmbulo. O número de símbolos *OFDM* pode ser obtido a partir de um valor definido em *ti_depth*. O combinador *L1 XFEC* 1017-*L1* de um receptor pode obter o bloco *L1 XFEC*. Então, a desintercalação no tempo 1010-*L1* pode ser realizada utilizando *ti_depth*. Finalmente, os blocos *L1 XFEC* podem ser fundidos para obter um bloco *L1_FEC*. Após a fusão *L1_FEC* 1018-*L1*, desintercalação de bits *r714-L1*, e descodificação *LDPC/BCH* *r715-L1*, pode ser obtido o bloco *L1*. *L1_info_size* pode ser multiplicado por dois, o bloco *L1* pode ser verificado com *CRC*, e *L1* pode ser descodificado. Pode ser desconsiderada zona de preenchimento desnecessária.

Utilizando os processos e dispositivos sugeridos, entre outras vantagens, é possível implementar um transmissor digital, receptor e sinalização eficiente da estrutura da camada física.

Ao transmitir a informação *ModCod* em cada

cabeçalho de quadro *BB* que é necessário para *ACM/VCM* e transmitir o resto da sinalização da camada física num cabeçalho de quadro, pode ser minimizada a informação complementar da sinalização.

Pode ser implementado o *QAM* modificado para uma transmissão mais eficiente em termos de energia ou um sistema de radiodifusão digital mais resistente ao ruído. O sistema pode incluir o transmissor e receptor para cada exemplo descrito e as suas combinações.

Pode ser implementado um *QAM* não uniforme melhorado para uma transmissão mais eficiente em termos de energia ou um sistema de radiodifusão digital mais robusto ao ruído. É também descrito um processo de utilização de uma taxa de código de código de correcção de erros de *NU-MQAM* e *MQAM*. O sistema pode incluir o transmissor e receptor para cada exemplo descrito e as suas combinações.

O processo de sinalização *L1* sugerido pode reduzir a informação adicional em 3~4% minimizando a informação complementar de sinalização durante a ligação do canal.

Será evidente para os técnicos que podem ser efectuadas várias modificações e variações na presente invenção, sem fugir do escopo da invenção.

Lisboa, 6 de dezembro de 2012

REIVINDICAÇÕES

1. Transmissor para a transmissão de dados de radiodifusão para um receptor, caracterizado por compreender:

um meio de processamento de entrada (101) configurado para receber sequências de entrada, para adicionar pelo menos um cabeçalho de banda de base às sequências de entrada recebidas, e compor pelo menos um quadro de banda de base; e

um primeiro meio *BICM*, *Bit Interleaved Coding & Modulation*, (102; 301 a 308) configurado para codificar em *BCH* os dados do quadro de banda base, para codificar os dados codificados *LDPC BCH*, e para bit intercalar os dados codificados *LDPC*;

um segundo meio *BICM* (301-1 a 308-1) configurado para codificar em *BCH* dados de sinalização de camada 1, para codificar em *LDPC* os dados de sinalização de camada 1 codificados em *BCH*, e para bit intercalar os dados de sinalização de camada 1 codificados em *LDPC*; sendo o transmissor caracterizado por compreender adicionalmente:

um meio de intercalação no tempo (1008-*L1*) configurado para intercalar no tempo os dados de sinalização da camada 1 com uma profundidade de intercalação no tempo;

um construtor de quadros (103) configurado para reunir os dados bit intercalados e os dados de sinalização de camada 1 intercalados no tempo para formar um quadro de sinal incluindo pelo menos um símbolo de preâmbulo que transporta os dados de sinalização de camada 1; e

um meio de geração *OFDM* (104) configurado para modular em *OFDM* o quadro sinal formado,

em que o transmissor se encontra configurado para processar pelo menos um símbolo de preâmbulo, incluindo o pelo menos um símbolo preâmbulo um cabeçalho de camada 1, sendo que o cabeçalho de camada 1 inclui informação *L1_INFO_Size* que indica a metade dos dados de sinalização *L1* e dados de preenchimento de bloco *L1* e informação de profundidade de intercalação no tempo *L1* que indica a profundidade da intercalação no tempo dos dados de sinalização da camada 1.

2. Transmissor de acordo com a reivindicação 1, caracterizado por o primeiro meio *BICM* compreender:

um primeiro codificador *BCH* (702-0) configurado para codificar em *BCH* os dados do quadro de banda base para gerar dados protegidos de erro;

um primeiro codificador *LDPC* (702-0) configurado para codificar em *LDPC* os dados codificados em *BCH*;

um primeiro intercalador de bits (703-0) configurado para intercalar em bits os dados codificados em *LDPC*;

um primeiro mapeador *QAM* (306-1, 704-0) configurado para mapear os dados bit intercalados em palavras de células e mapear as palavras de célula em valores de constelação; e

um intercalador de frequência no tempo (308, 708-0) configurado para intercalar no tempo/frequência os dados mapeados.

3. Transmissor de acordo com as reivindicações 1 ou 2, caracterizado por o segundo meio *BICM* compreender:

um segundo codificador *BCH* (702-L1) configurado para codificar em *BCH* os dados de sinalização de camada 1;

um segundo codificador *LDPC* (702-L1) configurado para codificar em *LDPC* os dados de sinalização de camada 1 codificados em *BCH* para gerar pelo menos um bit de paridade *LDPC*;

um meio de punção configurado para executar a punção no bit de paridade *LDPC* gerado;

um segundo intercalador de bits (703-L1) configurado para bit intercalar os dados de sinalização da camada 1 codificados em *LDPC* e o bit de paridade *LDPC* punção; e

um segundo mapeador *QAM* (704-L1) configurado para mapear os dados de sinalização de camada 1 intercalados em palavras de células e mapear as palavras de célula em valores de constelação.

4. Transmissor de acordo com a reivindicação 1, caracterizado por se encontrar adicionalmente configurado para processar os dados de preenchimento do bloco da camada 1, assegurando os dados de preenchimento do bloco de camada 1 que um comprimento dos dados de sinalização de camada 1 e os dados de preenchimento do bloco de camada 1 é um múltiplo de 2.

5. Receptor para processar dados de transmissão, compreendendo o receptor:

meios de recepção (r700) configurados para receber os

dados de radiodifusão, incluindo um quadro de sinal que inclui pelo menos um símbolo de preâmbulo que transporta dados de sinalização de camada 1; sendo o receptor caracterizado por compreender adicionalmente:

um meio de desintercalação no tempo (r710) configurado para desintercalar no tempo os dados de sinalização da camada 1 utilizando uma profundidade de intercalação no tempo;

um desmapeador *QAM* (r713-L1) configurado para desmapear valores de constelação que correspondem aos dados de sinalização da camada 1 em palavras de célula e para multiplexar as palavras da célula desmapeadas em dados de sinalização de camada 1;

um desintercalador de bits (r714-L1) configurado para bit desintercalar os dados de sinalização da camada 1 multiplexados;

um meio de despuncionagem configurado para executar a despuncionagem nos dados de sinalização de camada 1;

um decodificador *LDPC* (r715-L1) configurado para decodificar em *LDPC* os dados de sinalização da camada 1; e

um decodificador *BCH* (r715-L1) configurado para decodificar em *BCH* os dados de sinalização da camada 1 decodificados em *LDPC*,

em que o receptor se encontra configurado para processar o pelo menos um símbolo de preâmbulo, incluindo o pelo menos um símbolo de preâmbulo um cabeçalho de camada 1, sendo que o cabeçalho de camada 1 inclui informação *L1_INFO_Size* que indica a metade dos dados de sinalização *L1* e dados de preenchimento de bloco *L1* e informação de

profundidade de intercalação no tempo *L1* que indica a profundidade da intercalação no tempo dos dados de sinalização da camada 1.

6. Receptor de acordo com a reivindicação 5, caracterizado por compreender adicionalmente:

um desintercalador no tempo/frequência (r709) configurado para desintercalas no tempo/frequência valores de constelação que correspondem a dados *PLP*, *Physical Layer Pipe*;

um desmapeador QAM (r713-C) configurado para desmapear os valores de constelação desintercalados em palavras de célula e para multiplexar as palavras de célula desmapeadas em dados *PLP*;

um desintercalador de bits (r714-C) configurado para bit desintercalas os dados *PLP* multiplexados;

um decodificador *LDPC* (r715-C) configurado para decodificar em *LDPC* os dados *PLP* desintercalados em bits;
e

um decodificador *BCH* (r715-C) configurado para decodificar em *BCH* os dados *PLP* decodificados em *LDPC*.

7. Receptor de acordo com as reivindicações 5 ou 6, adicionalmente configurado para processar o pelo menos um símbolo de preâmbulo, caracterizado por o pelo menos um símbolo de preâmbulo incluir os dados de sinalização de camada 1, os dados de preenchimento de bloco de camada *L1*, e dados CRC para a detecção de erros.

8. Receptor de acordo com a reivindicação 7, caracterizado por se encontrar adicionalmente configurado para processar os dados de preenchimento do bloco da camada 1, assegurando os dados de preenchimento do bloco de camada 1 que um comprimento dos dados de sinalização de camada 1 e os dados de preenchimento do bloco de camada 1 são um múltiplo de 2.

9. Processo para transmitir dados de radiodifusão para um receptor, compreendendo o processo:
receber sequências de entrada;

adicionar pelo menos um cabeçalho de banda base para as sequências de entrada recebidas para compor pelo menos um quadro de banda base das sequências de entrada recebidas;

codificar dados em *BCH* do quadro de banda base;

codificar em *LDPC* os dados codificados em *BCH*;

bit intercalar os dados codificados em *LDPC*;

codificar em *BCH* os dados de sinalização de camada 1;

codificar em *LDPC* os dados de sinalização de camada 1 codificados em *BCH*;

bit intercalar os dados de sinalização de camada 1 codificados em *LDPC*; sendo o processo caracterizado por compreender adicionalmente:

intercalar no tempo os dados de sinalização da camada 1 bit intercalados com uma profundidade de intercalação no tempo;

reunir os dados bit intercalados do quadro da banda de base e os dados de sinalização de camada 1 intercalados no

tempo para formar um quadro de sinal incluindo pelo menos um símbolo de preâmbulo que transporta os dados de sinalização de camada 1; e

modular em *OFDM* o quadro de sinal formado,

em que o pelo menos um símbolo de preâmbulo inclui um cabeçalho de camada 1, sendo que o cabeçalho de camada 1 inclui informação *L1_INFO_Size* que indica a metade dos dados de sinalização *L1* e dados de preenchimento de bloco *L1* e informação de profundidade de intercalação no tempo *L1* que indica a profundidade da intercalação no tempo dos dados de sinalização *L1*.

10. Processo para receber dados de radiodifusão, caracterizado por compreender:

receber dados de radiodifusão incluindo um quadro de sinal que inclui pelo menos um símbolo de preâmbulo que transporta dados de sinalização de camada 1; sendo o processo caracterizado por compreender adicionalmente:

desintercalar no tempo os dados de sinalização da camada 1 utilizando uma profundidade de intercalação no tempo;

desmapear, por meio de um processo de desmapeamento QAM, valores de constelação que correspondem aos dados de sinalização da camada 1 em palavras de células e multiplexar as palavras de células desmapeadas transformando-as em dados de sinalização de camada 1;

bit desintercalar os dados de sinalização de camada 1 multiplexados; e

despuncionar os dados de sinalização de camada 1;

descodificar por meio de um processo de descodificação *LDPC* os dados de sinalização de camada 1; e

descodificar por meio de um processo de descodificação *BCH* os dados de sinalização de camada 1 descodificados em *LDPC*,

em que o pelo menos um símbolo de preâmbulo inclui um cabeçalho de camada 1, sendo que o cabeçalho de camada 1 inclui informação *L1_INFO_Size* que indica a metade dos dados de sinalização *L1* e dados de preenchimento de bloco *L1* e informação de profundidade de intercalação no tempo *L1* que indica a profundidade da intercalação no tempo dos dados de sinalização *L1*.

11. Processo de acordo com a reivindicação 10, caracterizado por compreender adicionalmente:

desintercalação no tempo/frequência de valores de constelação que correspondem a dados *PLP*, *Physical Layer Pipe* (pipe de camada física);

desmapear os valores de constelação desintercalados em palavras de célula;

multiplexar as palavras de célula desmapeadas em dados *PLP*;

bit desintercalar os dados *PLP* multiplexados;

descodificar em *LDPC* os dados *PLP* bit desintercalados;

e

descodificar em *BCH* os dados *PLP* descodificados em *LDPC*.

12. Processo de acordo com as reivindicações 10

ou 11, caracterizado por compreender adicionalmente: processar o pelo menos um símbolo de preâmbulo, incluindo o pelo menos um símbolo de preâmbulo os dados de sinalização de camada 1, os dados de preenchimento de bloco de camada *L1*, e dados *CRC* para a detecção de erros.

13. Processo de acordo com a reivindicação 12, caracterizado por os dados de preenchimento do bloco da camada 1 serem para assegurar que um comprimento dos dados de sinalização de camada 1 e os dados de preenchimento do bloco de camada 1 são um múltiplo de 2.

Lisboa, 6 de dezembro de 2012

Fig. 1

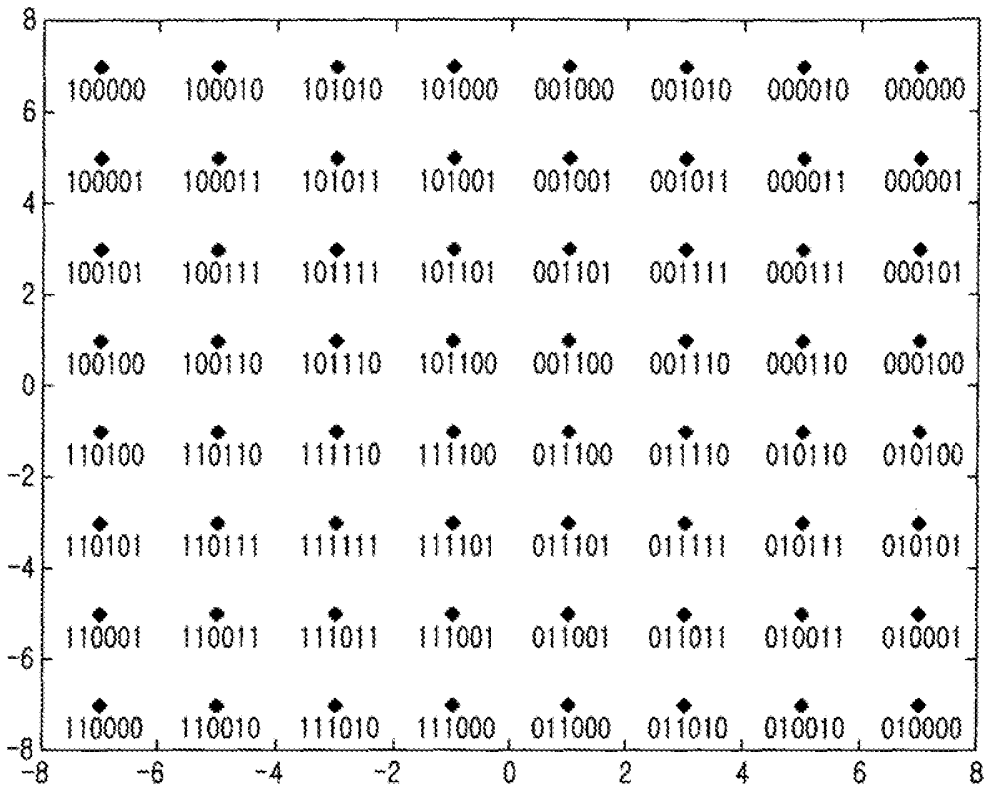


Fig. 2

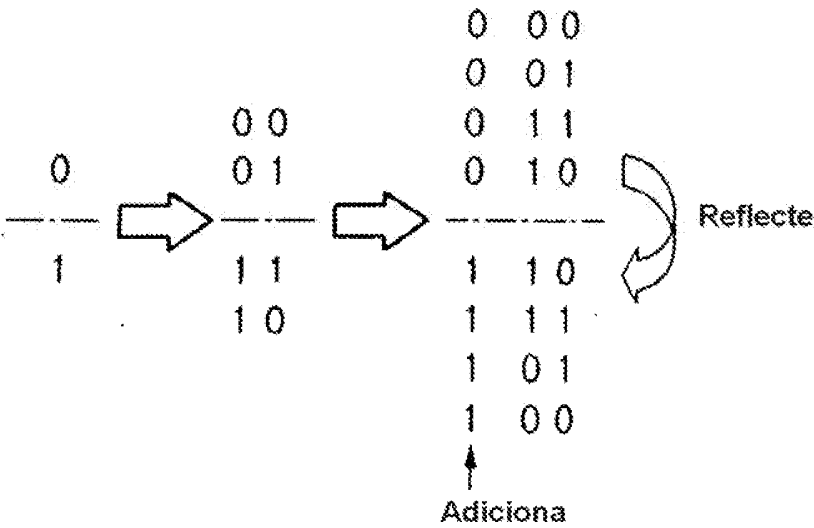


Fig. 3

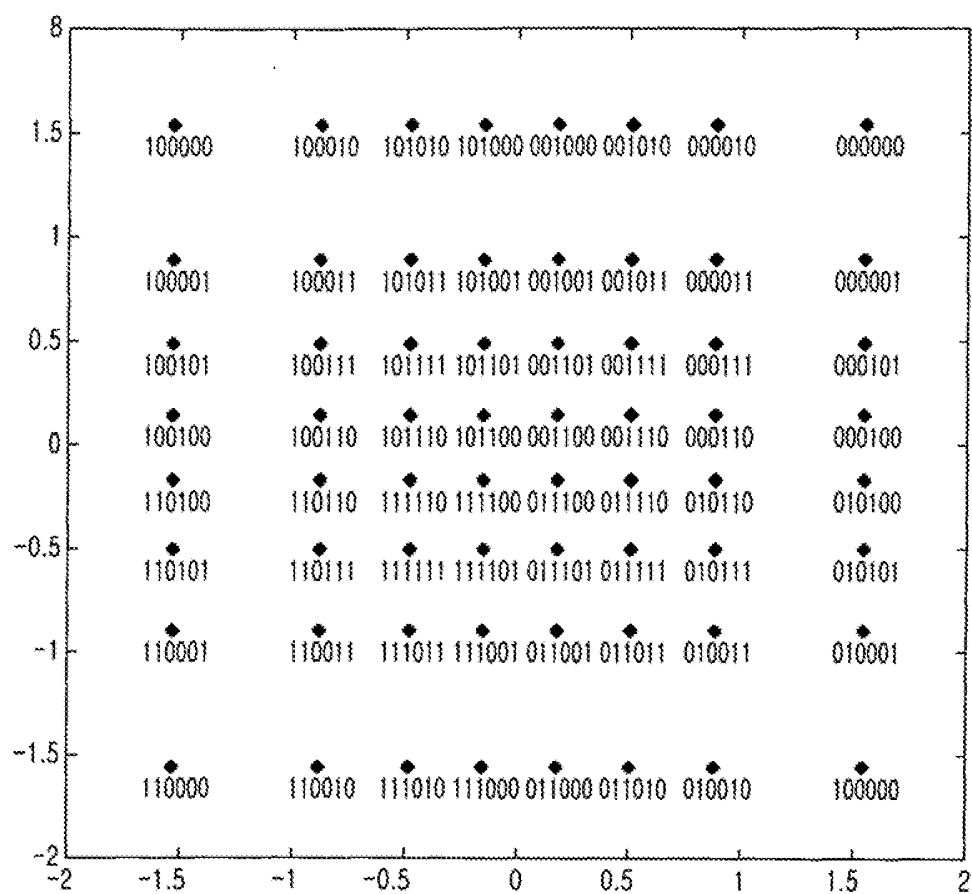


Fig. 4

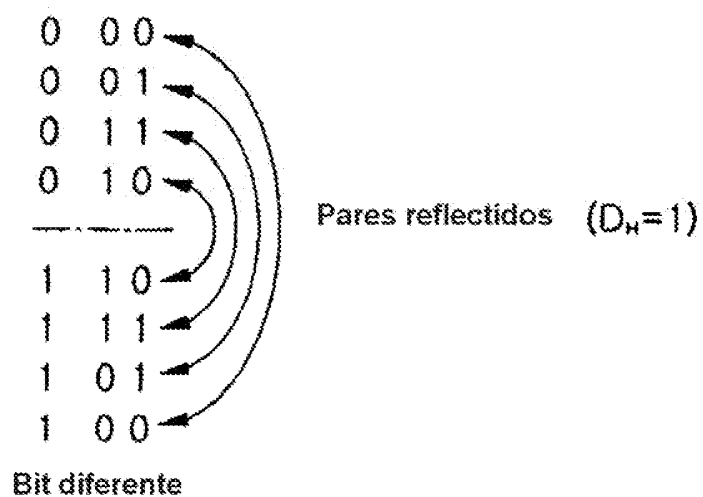


Fig. 5

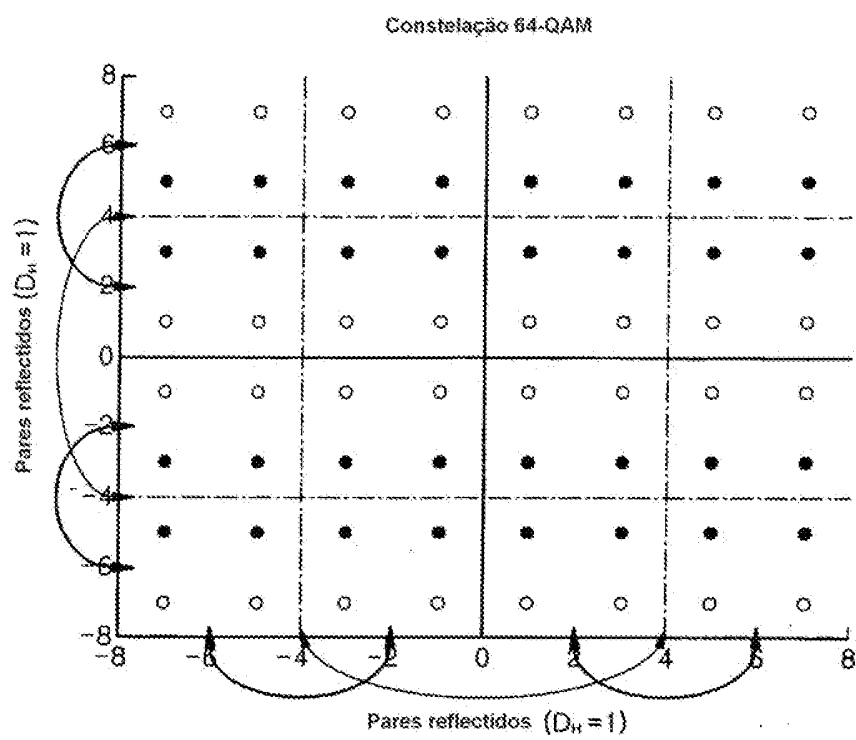
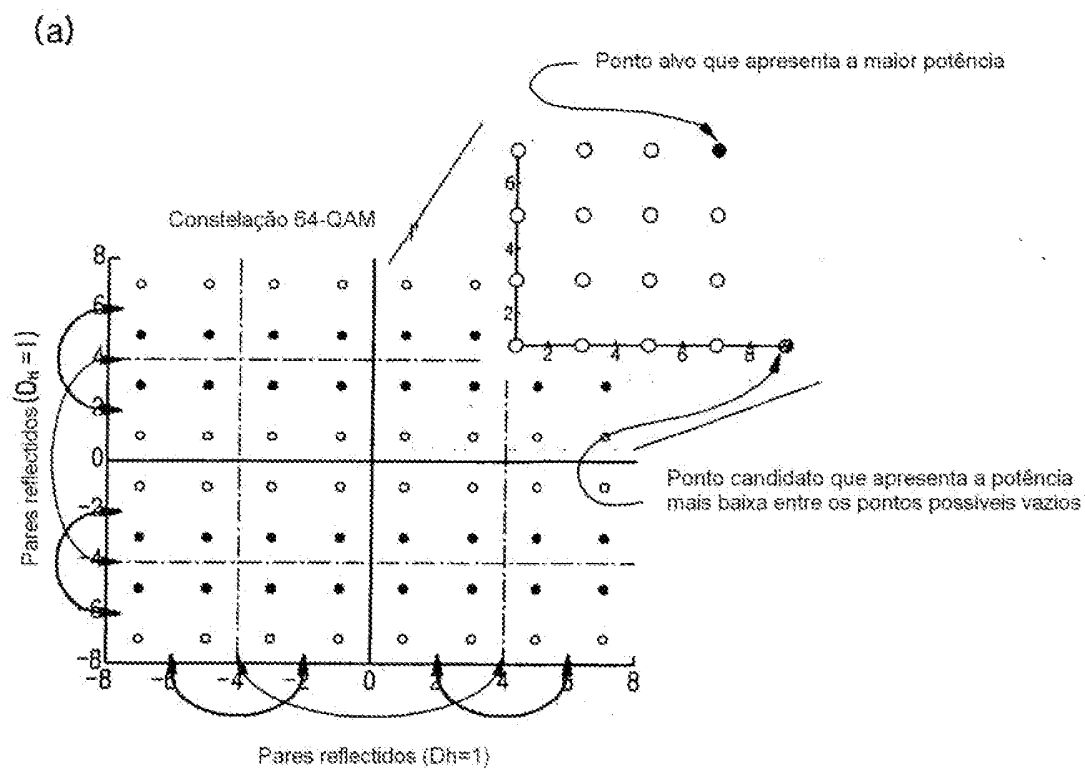


Fig. 6



(b)

- Ponto candidato no ponto vizinho mais próximo do par reflectido do ponto do alvo

Pt : Potência do ponto alvo
 Pc : Potência do ponto candidato
 Ps : Potências de cada ponto difundido

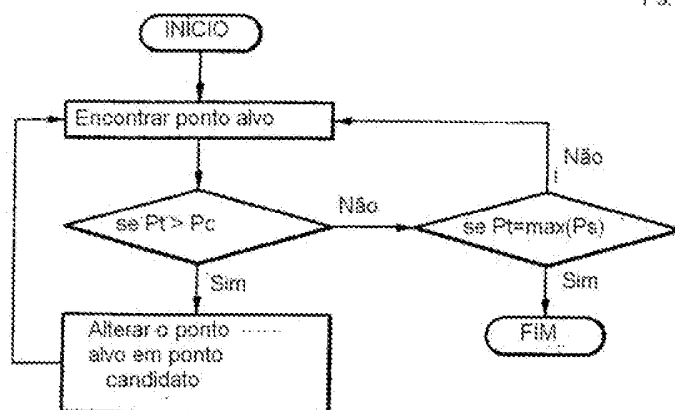


Fig. 7

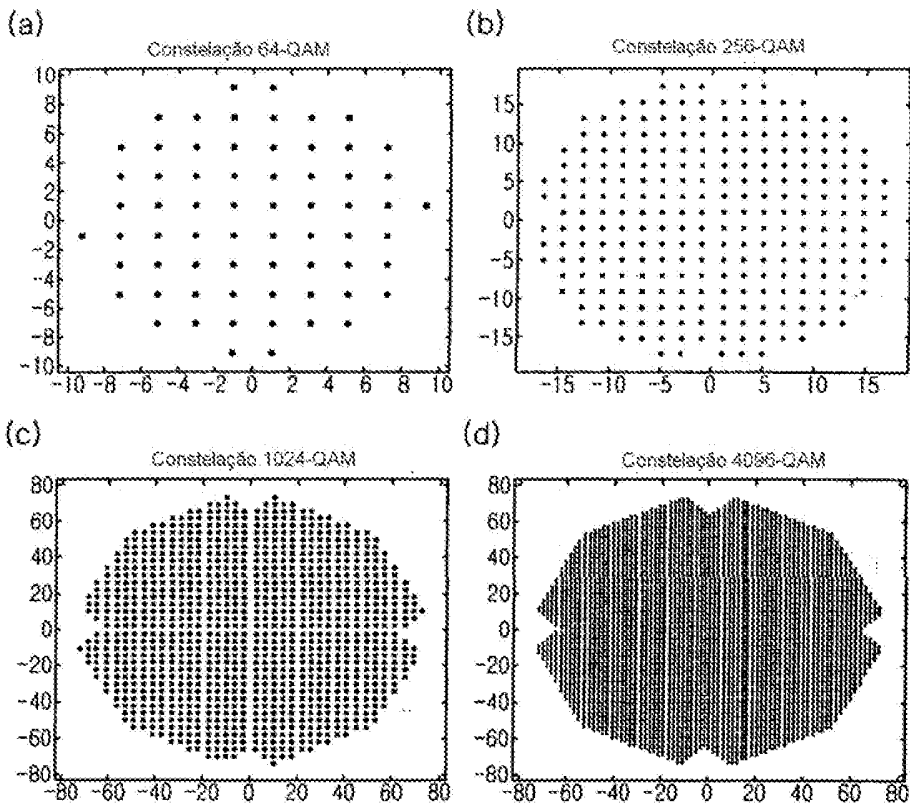


Fig. 8

Value	Scatter
0	9 + 1i
1	7 + 5i
2	5 + 7i
3	5 + 5i
4	7 + 1i
5	7 + 3i
6	5 + 1i
7	5 + 3i
8	1 + 7i
9	1 + 5i
10	3 + 7i
11	3 + 5i
12	1 + 1i
13	1 + 3i
14	3 + 1i
15	3 + 3i
16	1 - 9i
17	7 - 5i
18	5 - 7i
19	5 - 5i
20	7 - 1i
21	7 - 3i
22	5 - 1i
23	5 - 3i
24	1 - 7i
25	1 - 5i
26	3 - 7i
27	3 - 5i
28	1 - 1i
29	1 - 3i
30	3 - 1i

Fig. 9

31	3 - 3i
32	-1 + 9i
33	-7 + 5i
34	-5 + 7i
35	-5 + 5i
36	-7 + 1i
37	-7 + 3i
38	-5 + 1i
39	-5 + 3i
40	-1 + 7i
41	-1 + 5i
42	-3 + 7i
43	-3 + 5i
44	-1 + 1i
45	-1 + 3i
46	-3 + 1i
47	-3 + 3i
48	-9 - 1i
49	-7 - 5i
50	-5 - 7i
51	-5 - 5i
52	-7 - 1i
53	-7 - 3i
54	-5 - 1i
55	-5 - 3i
56	-1 - 7i
57	-1 - 5i
58	-3 - 7i
59	-3 - 5i
60	-1 - 1i
61	-1 - 3i
62	-3 - 1i
63	-3 - 3i

Value = Valor, Scatter = Difusão

Fig. 10

Value	Scatter	Value	Scatter	Value	Scatter	Value	Scatter
0	17 + 11	64	1 - 17	128	-1 + 17	192	-17 - 11
1	17 + 3	65	17 - 3	129	-17 + 3	193	-17 - 3
2	3 + 17	66	3 - 17	130	-3 + 17	194	-3 - 17
3	13 + 13	67	13 - 13	131	-13 + 13	195	-13 - 13
4	15 + 9	68	15 - 9	132	-15 + 9	196	-15 - 9
5	17 + 5	69	17 - 5	133	-17 + 5	197	-17 - 5
6	13 + 9	70	13 - 9	134	-13 + 9	198	-13 - 9
7	13 + 11	71	13 - 11	135	-13 + 11	199	-13 - 11
8	9 + 15	72	9 - 15	136	-9 + 15	200	-9 - 15
9	9 + 13	73	9 - 13	137	-9 + 13	201	-9 - 13
10	5 + 17	74	5 - 17	138	-5 + 17	202	-5 - 17
11	11 + 13	75	11 - 13	139	-11 + 13	203	-11 - 13
12	9 + 9	76	9 - 9	140	-9 + 9	204	-9 - 9
13	9 + 11	77	9 - 11	141	-9 + 11	205	-9 - 11
14	11 + 9	78	11 - 9	142	-11 + 9	206	-11 - 9
15	11 + 11	79	11 - 11	143	-11 + 11	207	-11 - 11
16	15 + 1	80	15 - 1	144	-15 + 1	208	-15 - 1
17	15 + 3	81	15 - 3	145	-15 + 3	209	-15 - 3
18	13 + 1	82	13 - 1	146	-13 + 1	210	-13 - 1
19	13 + 3	83	13 - 3	147	-13 + 3	211	-13 - 3
20	15 + 7	84	15 - 7	148	-15 + 7	212	-15 - 7
21	15 + 5	85	15 - 5	149	-15 + 5	213	-15 - 5
22	13 + 7	86	13 - 7	150	-13 + 7	214	-13 - 7
23	13 + 5	87	13 - 5	151	-13 + 5	215	-13 - 5
24	9 + 1	88	9 - 1	152	-9 + 1	216	-9 - 1
25	9 + 3	89	9 - 3	153	-9 + 3	217	-9 - 3
26	11 + 1	90	11 - 1	154	-11 + 1	218	-11 - 1
27	11 + 3	91	11 - 3	155	-11 + 3	219	-11 - 3
28	9 + 7	92	9 - 7	156	-9 + 7	220	-9 - 7
29	9 + 5	93	9 - 5	157	-9 + 5	221	-9 - 5
30	11 + 7	94	11 - 7	158	-11 + 7	222	-11 - 7

Fig. 11

31	11 + 5	95	11 - 5	159	-11 + 5	223	-11 - 5
32	1 + 15	96	1 - 15	160	-1 + 15	224	-1 - 15
33	1 + 13	97	1 - 13	161	-1 + 13	225	-1 - 13
34	3 + 15	98	3 - 15	162	-3 + 15	226	-3 - 15
35	3 + 13	99	3 - 13	163	-3 + 13	227	-3 - 13
36	1 + 9	100	1 - 9	164	-1 + 9	228	-1 - 9
37	1 + 11	101	1 - 11	165	-1 + 11	229	-1 - 11
38	3 + 9	102	3 - 9	166	-3 + 9	230	-3 - 9
39	3 + 11	103	3 - 11	167	-3 + 11	231	-3 - 11
40	7 + 15	104	7 - 15	168	-7 + 15	232	-7 - 15
41	7 + 13	105	7 - 13	169	-7 + 13	233	-7 - 13
42	5 + 15	106	5 - 15	170	-5 + 15	234	-5 - 15
43	5 + 13	107	5 - 13	171	-5 + 13	235	-5 - 13
44	7 + 9	108	7 - 9	172	-7 + 9	236	-7 - 9
45	7 + 11	109	7 - 11	173	-7 + 11	237	-7 - 11
46	5 + 9	110	5 - 9	174	-5 + 9	238	-5 - 9
47	5 + 11	111	5 - 11	175	-5 + 11	239	-5 - 11
48	1 + 1	112	1 - 1	176	-1 + 1	240	-1 - 1
49	1 + 3	113	1 - 3	177	-1 + 3	241	-1 - 3
50	3 + 1	114	3 - 1	178	-3 + 1	242	-3 - 1
51	3 + 3	115	3 - 3	179	-3 + 3	243	-3 - 3
52	1 + 7	116	1 - 7	180	-1 + 7	244	-1 - 7
53	1 + 5	117	1 - 5	181	-1 + 5	245	-1 - 5
54	3 + 7	118	3 - 7	182	-3 + 7	246	-3 - 7
55	3 + 5	119	3 - 5	183	-3 + 5	247	-3 - 5
56	7 + 1	120	7 - 1	184	-7 + 1	248	-7 - 1
57	7 + 3	121	7 - 3	185	-7 + 3	249	-7 - 3
58	5 + 1	122	5 - 1	186	-5 + 1	250	-5 - 1
59	5 + 3	123	5 - 3	187	-5 + 3	251	-5 - 3
60	7 + 7	124	7 - 7	188	-7 + 7	252	-7 - 7
61	7 + 5	125	7 - 5	189	-7 + 5	253	-7 - 5
62	5 + 7	126	5 - 7	190	-5 + 7	254	-5 - 7
63	5 + 5	127	5 - 5	191	-5 + 5	255	-5 - 5

Value = Valor, Scatter = Difusão

Fig. 12

Value	Scatter	Value	Scatter	Value	Scatter	Value	Scatter	Value	Scatter	Value	Scatter	Value	Scatter	Value	Scatter	Value	Scatter	Value	Scatter	Value	Scatter	Value	Scatter	Value	Scatter	Value	Scatter	Value	Scatter
0	33 + 11	64	31 + 11	128	1 + 311	192	1 + 31	256	1 - 331	320	31 - 11	384	1 - 311	448	1 - 11														
1	33 + 31	65	31 + 31	129	1 + 291	193	1 + 31	257	13 - 31	321	31 - 31	385	1 - 291	449	1 - 31														
2	3 + 331	66	29 + 11	130	3 + 311	194	3 + 11	258	3 - 331	322	29 - 11	386	3 - 311	450	3 - 11														
3	35 + 31	67	29 + 31	131	3 + 291	195	3 + 31	259	3 - 351	323	29 - 31	387	3 - 291	451	3 - 31														
4	33 + 71	68	31 + 71	132	1 + 251	196	1 + 71	260	13 - 71	324	31 - 71	388	1 - 251	452	1 - 71														
5	33 + 51	69	31 + 51	133	1 + 271	197	1 + 51	261	13 - 51	325	31 - 51	389	1 - 271	453	1 - 51														
6	35 + 71	70	29 + 71	134	3 + 251	198	3 + 71	262	35 - 71	326	29 - 71	390	3 - 251	454	3 - 71														
7	35 + 51	71	29 + 51	135	3 + 271	199	3 + 51	263	35 - 51	327	29 - 51	391	3 - 271	455	3 - 51														
8	7 + 331	72	25 + 11	136	7 + 311	200	7 + 11	264	7 - 331	328	25 - 11	392	7 - 311	456	7 - 11														
9	7 + 351	73	25 + 31	137	7 + 291	201	7 + 31	265	7 - 351	329	25 - 31	393	7 - 291	457	7 - 31														
10	5 + 331	74	27 + 11	138	5 + 311	202	5 + 11	266	5 - 331	330	27 - 11	394	5 - 311	458	5 - 11														
11	5 + 351	75	27 + 31	139	5 + 291	203	5 + 31	267	5 - 351	331	27 - 31	395	5 - 291	459	5 - 31														
12	25 + 251	76	25 + 71	140	7 + 251	204	7 + 71	268	25 - 251	332	25 - 71	396	7 - 251	460	7 - 71														
13	25 + 271	77	25 + 51	141	7 + 271	205	7 + 51	269	25 - 271	333	25 - 51	397	7 - 271	461	7 - 51														
14	27 + 251	78	27 + 71	142	5 + 251	206	5 + 71	270	27 - 251	334	27 - 71	398	5 - 251	462	5 - 71														
15	37 + 51	79	27 + 51	143	5 + 271	207	5 + 51	271	5 - 371	335	27 - 51	399	5 - 271	463	5 - 51														
16	31 + 171	80	31 + 151	144	1 + 171	208	1 + 151	272	31 - 171	336	31 - 151	400	1 - 171	464	1 - 151														
17	33 + 131	81	31 + 131	145	1 + 191	209	1 + 131	273	33 - 131	337	31 - 131	401	1 - 191	465	1 - 131														
18	29 + 171	82	29 + 151	146	3 + 171	210	3 + 151	274	29 - 171	338	29 - 151	402	3 - 171	466	3 - 151														
19	29 + 191	83	29 + 131	147	3 + 191	211	3 + 131	275	29 - 191	339	29 - 131	403	3 - 191	467	3 - 131														
20	33 + 91	84	31 + 91	148	1 + 231	212	1 + 91	276	33 - 91	340	31 - 91	404	1 - 231	468	1 - 91														
21	33 + 111	85	31 + 111	149	1 + 211	213	1 + 111	277	33 - 111	341	31 - 111	405	1 - 211	469	1 - 111														
22	35 + 91	86	29 + 91	150	3 + 231	214	3 + 91	278	35 - 91	342	29 - 91	406	3 - 231	470	3 - 91														
23	29 + 211	87	29 + 111	151	3 + 211	215	3 + 111	279	29 - 211	343	29 - 111	407	3 - 211	471	3 - 111														
24	25 + 171	88	25 + 151	152	7 + 171	216	7 + 151	280	25 - 171	344	25 - 151	408	7 - 171	472	7 - 151														
25	25 + 191	89	25 + 131	153	7 + 191	217	7 + 131	281	25 - 191	345	25 - 131	409	7 - 191	473	7 - 131														
26	27 + 171	90	27 + 151	154	5 + 171	218	5 + 151	282	27 - 171	346	27 - 151	410	5 - 171	474	5 - 151														
27	27 + 191	91	27 + 131	155	5 + 191	219	5 + 131	283	27 - 191	347	27 - 131	411	5 - 191	475	5 - 131														
28	25 + 231	92	25 + 91	156	7 + 231	220	7 + 91	284	25 - 231	348	25 - 91	412	7 - 231	476	7 - 91														
29	25 + 211	93	25 + 111	157	7 + 211	221	7 + 111	285	25 - 211	349	25 - 111	413	7 - 211	477	7 - 111														
30	27 + 231	94	27 + 91	158	5 + 231	222	5 + 91	286	27 - 231	350	27 - 91	414	5 - 231	478	5 - 91														

Value = Valor, Scatter = Difusão

Fig. 13

31	27 + 21	95	27 + 11	159	5 + 21	223	5 + 11	287	27 - 21	351	27 - 11	415	5 - 21	479	5 - 11
32	17 + 31	96	17 + 11	160	15 + 31	224	15 + 11	288	17 - 31	352	17 - 11	416	15 - 31	480	15 - 11
33	17 + 29	97	17 + 31	161	15 + 29	225	15 + 31	289	17 - 29	353	17 - 31	417	15 - 29	481	15 - 31
34	13 + 33	98	19 + 11	162	13 + 31	226	13 + 11	290	13 - 33	354	19 - 11	418	13 - 31	482	13 - 11
35	19 + 29	99	19 + 31	163	13 + 29	227	13 + 31	291	19 - 29	355	19 - 31	419	13 - 29	483	13 - 31
36	17 + 25	100	17 + 71	164	15 + 25	228	15 + 71	292	17 - 25	356	17 - 71	420	15 - 25	484	15 - 71
37	17 + 27	101	17 + 51	165	15 + 27	229	15 + 51	293	17 - 27	357	17 - 51	421	15 - 27	485	15 - 51
38	19 + 25	102	19 + 71	166	13 + 25	230	13 + 71	294	19 - 25	358	19 - 71	422	13 - 25	486	13 - 71
39	19 + 27	103	19 + 51	167	13 + 27	231	13 + 51	295	19 - 27	359	19 - 51	423	13 - 27	487	13 - 51
40	9 + 33	104	23 + 11	168	9 + 31	232	9 + 11	296	9 - 33	360	23 - 11	424	9 - 31	488	9 - 11
41	9 + 35	105	23 + 31	169	9 + 29	233	9 + 31	297	9 - 35	361	23 - 31	425	9 - 29	489	9 - 31
42	11 + 33	106	21 + 11	170	11 + 31	234	11 + 11	298	11 - 33	362	21 - 11	426	11 - 31	490	11 - 11
43	21 + 29	107	21 + 31	171	11 + 29	235	11 + 31	299	21 - 29	363	21 - 31	427	11 - 29	491	11 - 31
44	23 + 25	108	23 + 71	172	9 + 25	236	9 + 71	300	23 - 25	364	23 - 71	428	9 - 25	492	9 - 71
45	23 + 27	109	23 + 51	173	9 + 27	237	9 + 51	301	23 - 27	365	23 - 51	429	9 - 27	493	9 - 51
46	21 + 25	110	21 + 71	174	11 + 25	238	11 + 71	302	21 - 25	366	21 - 71	430	11 - 25	494	11 - 71
47	21 + 27	111	21 + 51	175	11 + 27	239	11 + 51	303	21 - 27	367	21 - 51	431	11 - 27	495	11 - 51
48	17 + 17	112	17 + 15	176	15 + 17	240	15 + 15	304	17 - 17	368	17 - 15	432	15 - 17	496	15 - 15
49	17 + 19	113	17 + 13	177	15 + 19	241	15 + 13	305	17 - 19	369	17 - 13	433	15 - 19	497	15 - 13
50	19 + 17	114	19 + 15	178	13 + 17	242	13 + 15	306	19 - 17	370	19 - 15	434	13 - 17	498	13 - 15
51	19 + 19	115	19 + 13	179	13 + 19	243	13 + 13	307	19 - 19	371	19 - 13	435	13 - 19	499	13 - 13
52	17 + 23	116	17 + 91	180	15 + 23	244	15 + 91	308	17 - 23	372	17 - 91	436	15 - 23	500	15 - 91
53	17 + 21	117	17 + 11	181	15 + 21	245	15 + 11	309	17 - 21	373	17 - 11	437	15 - 21	501	15 - 11
54	19 + 23	118	19 + 91	182	13 + 23	246	13 + 91	310	19 - 23	374	19 - 91	438	13 - 23	502	13 - 91
55	19 + 21	119	19 + 11	183	13 + 21	247	13 + 11	311	19 - 21	375	19 - 11	439	13 - 21	503	13 - 11
56	23 + 17	120	23 + 15	184	9 + 17	248	9 + 15	312	23 - 17	376	23 - 15	440	9 - 17	504	9 - 15
57	23 + 19	121	23 + 13	185	9 + 19	249	9 + 13	313	23 - 19	377	23 - 13	441	9 - 19	505	9 - 13
58	21 + 17	122	21 + 15	186	11 + 17	250	11 + 15	314	21 - 17	378	21 - 15	442	11 - 17	506	11 - 15
59	21 + 19	123	21 + 13	187	11 + 19	251	11 + 13	315	21 - 19	379	21 - 13	443	11 - 19	507	11 - 13
60	23 + 23	124	23 + 91	188	9 + 23	252	9 + 91	316	23 - 23	380	23 - 91	444	9 - 23	508	9 - 91
61	23 + 21	125	23 + 11	189	9 + 21	253	9 + 11	317	23 - 21	381	23 - 11	445	9 - 21	509	9 - 11
62	21 + 23	126	21 + 91	190	11 + 23	254	11 + 91	318	21 - 23	382	21 - 91	446	11 - 23	510	11 - 91
63	21 + 21	127	21 + 11	191	11 + 21	255	11 + 11	319	21 - 21	383	21 - 11	447	11 - 21	511	11 - 11

Fig. 14

Value	Scatter	Value	Scatter	Value	Scatter	Value	Scatter	Value	Scatter	Value	Scatter	Value	Scatter
512	-1 + 33i	576	-31 + 1i	640	-1 + 31i	704	-1 + 1i	768	-33 - 1i	832	-31 - 1i	896	-1 - 31i
513	-33 + 3i	577	-31 + 3i	641	-1 + 29i	705	-1 + 3i	769	-33 - 3i	833	-31 - 3i	897	-1 - 29i
514	-3 + 33i	578	-29 + 1i	642	-3 + 31i	706	-3 + 1i	770	-3 - 33i	834	-29 - 1i	898	-3 - 31i
515	-3 + 35i	579	-29 + 3i	643	-3 + 29i	707	-3 + 3i	771	-35 - 3i	835	-29 - 3i	899	-3 - 29i
516	-33 + 7i	580	-31 + 7i	644	-1 + 25i	708	-1 + 7i	772	-33 - 7i	836	-31 - 7i	900	-1 - 25i
517	-33 + 5i	581	-31 + 5i	645	-1 + 27i	709	-1 + 5i	773	-33 - 5i	837	-31 - 5i	901	-1 - 27i
518	-35 + 7i	582	-29 + 7i	646	-3 + 25i	710	-3 + 7i	774	-35 - 7i	838	-29 - 7i	902	-3 - 25i
519	-35 + 5i	583	-29 + 5i	647	-3 + 27i	711	-3 + 5i	775	-35 - 5i	839	-29 - 5i	903	-3 - 27i
520	-7 + 33i	584	-25 + 1i	648	-7 + 31i	712	-7 + 1i	776	-7 - 33i	840	-25 - 1i	904	-7 - 31i
521	-7 + 35i	585	-25 + 3i	649	-7 + 29i	713	-7 + 3i	777	-7 - 35i	841	-25 - 3i	905	-7 - 29i
522	-5 + 33i	586	-27 + 1i	650	-5 + 31i	714	-5 + 1i	778	-5 - 33i	842	-27 - 1i	906	-5 - 31i
523	-5 + 35i	587	-27 + 3i	651	-5 + 29i	715	-5 + 3i	779	-5 - 35i	843	-27 - 3i	907	-5 - 29i
524	-25 + 25i	588	-25 + 7i	652	-7 + 25i	716	-7 + 7i	780	-25 - 25i	844	-25 - 7i	908	-7 - 25i
525	-25 + 27i	589	-25 + 5i	653	-7 + 27i	717	-7 + 5i	781	-25 - 27i	845	-25 - 5i	909	-7 - 27i
526	-27 + 25i	590	-27 + 7i	654	-5 + 25i	718	-5 + 7i	782	-27 - 25i	846	-27 - 7i	910	-5 - 25i
527	-5 + 37i	591	-27 + 5i	655	-5 + 27i	719	-5 + 5i	783	-37 - 5i	847	-27 - 5i	911	-5 - 27i
528	-31 + 17i	592	-31 + 15i	656	-1 + 17i	720	-1 + 15i	784	-31 - 17i	848	-31 - 15i	912	-1 - 17i
529	-33 + 13i	593	-31 + 13i	657	-1 + 19i	721	-1 + 13i	785	-33 - 13i	849	-31 - 13i	913	-1 - 19i
530	-29 + 17i	594	-29 + 15i	658	-3 + 17i	722	-3 + 15i	786	-29 - 17i	850	-29 - 15i	914	-3 - 17i
531	-29 + 19i	595	-29 + 13i	659	-3 + 19i	723	-3 + 13i	787	-29 - 19i	851	-29 - 13i	915	-3 - 19i
532	-33 + 9i	596	-31 + 9i	660	-1 + 23i	724	-1 + 9i	788	-33 - 9i	852	-31 - 9i	916	-1 - 23i
533	-33 + 11i	597	-31 + 11i	661	-1 + 21i	725	-1 + 11i	789	-33 - 11i	853	-31 - 11i	917	-1 - 21i
534	-35 + 9i	598	-29 + 9i	662	-3 + 23i	726	-3 + 9i	790	-35 - 9i	854	-29 - 9i	918	-3 - 23i
535	-29 + 21i	599	-29 + 11i	663	-3 + 21i	727	-3 + 11i	791	-29 - 21i	855	-29 - 11i	919	-3 - 21i
536	-25 + 17i	600	-25 + 15i	664	-7 + 17i	728	-7 + 15i	792	-25 - 17i	856	-25 - 15i	920	-7 - 17i
537	-25 + 19i	601	-25 + 13i	665	-7 + 19i	729	-7 + 13i	793	-25 - 19i	857	-25 - 13i	921	-7 - 19i
538	-27 + 17i	602	-27 + 15i	666	-5 + 17i	730	-5 + 15i	794	-27 - 17i	858	-27 - 15i	922	-5 - 17i
539	-27 + 19i	603	-27 + 13i	667	-5 + 19i	731	-5 + 13i	795	-27 - 19i	859	-27 - 13i	923	-5 - 19i
540	-25 + 23i	604	-25 + 9i	668	7 + 23i	732	-7 + 9i	796	-25 - 23i	860	-25 - 9i	924	-7 - 23i

Value = Valor, Scatter = Difusão

Fig. 16

31	69 + 11	95	69 + 21	159	37 + 53	223	37 + 43	287	59 + 11	351	59 + 21	415	37 + 11	479	37 + 21
32	15 + 65	96	49 + 33	160	17 + 65	224	47 + 33	288	49 + 11	352	49 + 31	416	47 + 11	480	47 + 31
33	15 + 67	97	49 + 35	161	17 + 67	225	47 + 35	289	49 + 31	353	49 + 29	417	47 + 31	481	47 + 29
34	13 + 65	98	51 + 33	162	19 + 65	226	45 + 33	290	51 + 11	354	51 + 31	418	45 + 11	482	45 + 31
35	13 + 67	99	51 + 35	163	19 + 67	227	45 + 35	291	51 + 31	355	51 + 29	419	45 + 31	483	45 + 29
36	15 + 71	100	49 + 39	164	17 + 71	228	47 + 39	292	49 + 71	356	49 + 25	420	47 + 71	484	47 + 25
37	15 + 69	101	49 + 37	165	17 + 69	229	47 + 37	293	49 + 51	357	49 + 27	421	47 + 51	485	47 + 27
38	13 + 71	102	51 + 39	166	45 + 57	230	45 + 39	294	51 + 71	358	51 + 25	422	45 + 71	486	45 + 25
39	13 + 69	103	51 + 37	167	19 + 69	231	45 + 37	295	51 + 51	359	51 + 27	423	45 + 51	487	45 + 27
40	9 + 65	104	55 + 33	168	23 + 65	232	41 + 33	296	55 + 11	360	55 + 31	424	41 + 11	488	41 + 31
41	9 + 67	105	55 + 35	169	23 + 67	233	41 + 35	297	55 + 31	361	55 + 29	425	41 + 31	489	41 + 29
42	11 + 65	106	53 + 33	170	21 + 65	234	43 + 33	298	53 + 11	362	53 + 31	426	43 + 11	490	43 + 31
43	11 + 67	107	53 + 35	171	21 + 67	235	43 + 35	299	53 + 31	363	53 + 29	427	43 + 31	491	43 + 29
44	9 + 71	108	55 + 39	172	41 + 57	236	41 + 39	300	55 + 71	364	55 + 25	428	41 + 71	492	41 + 25
45	9 + 69	109	55 + 37	173	41 + 59	237	41 + 37	301	55 + 51	365	55 + 27	429	41 + 51	493	41 + 27
46	11 + 71	110	53 + 39	174	43 + 57	238	43 + 39	302	53 + 71	366	53 + 25	430	43 + 71	494	43 + 25
47	11 + 69	111	53 + 37	175	21 + 69	239	43 + 37	303	53 + 51	367	53 + 27	431	43 + 51	495	43 + 27
48	49 + 49	112	49 + 47	176	47 + 49	240	47 + 47	304	49 + 151	368	49 + 171	432	47 + 151	496	47 + 171
49	49 + 51	113	49 + 45	177	47 + 51	241	47 + 45	305	49 + 131	369	49 + 191	433	47 + 131	497	47 + 191
50	51 + 49	114	51 + 47	178	45 + 49	242	45 + 47	306	51 + 171	370	51 + 171	434	45 + 151	498	45 + 171
51	51 + 51	115	51 + 45	179	45 + 51	243	45 + 45	307	51 + 151	371	51 + 191	435	45 + 131	499	45 + 191
52	49 + 53	116	49 + 41	180	47 + 53	244	47 + 41	308	49 + 91	372	49 + 231	436	47 + 91	500	47 + 231
53	49 + 53	117	49 + 43	181	47 + 53	245	47 + 43	309	49 + 111	373	49 + 211	437	47 + 111	501	47 + 211
54	13 + 73	118	51 + 41	182	45 + 55	246	45 + 41	310	51 + 91	374	51 + 231	438	45 + 91	502	45 + 231
55	51 + 53	119	51 + 43	183	45 + 53	247	45 + 43	311	51 + 111	375	51 + 211	439	45 + 111	503	45 + 211
56	55 + 49	120	55 + 47	184	41 + 49	248	41 + 47	312	55 + 151	376	55 + 171	440	41 + 151	504	41 + 171
57	73 + 13	121	55 + 45	185	41 + 51	249	41 + 45	313	55 + 131	377	55 + 191	441	41 + 131	505	41 + 191
58	53 + 49	122	53 + 47	186	43 + 49	250	43 + 47	314	53 + 151	378	53 + 171	442	43 + 151	506	43 + 171
59	53 + 51	123	53 + 45	187	43 + 51	251	43 + 45	315	53 + 131	379	53 + 191	443	43 + 131	507	43 + 191
60	73 + 9	124	55 + 41	188	41 + 55	252	41 + 41	316	55 + 91	380	55 + 231	444	41 + 91	508	41 + 231
61	73 + 11	125	55 + 43	189	41 + 53	253	41 + 43	317	55 + 111	381	55 + 211	445	41 + 111	509	41 + 211
62	11 + 73	126	53 + 41	190	43 + 53	254	43 + 41	318	53 + 91	382	53 + 231	446	43 + 91	510	43 + 231
63	53 + 53	127	53 + 43	191	43 + 53	255	43 + 43	319	53 + 111	383	53 + 211	447	43 + 111	511	43 + 211

Fig. 17

31	69 + 11i	95	69 + 21i	159	37 + 53i	223	37 + 43i	287	59 + 11i	351	59 + 21i	415	37 + 11i	479	37 + 21i
32	15 + 65i	96	49 + 33i	160	17 + 65i	224	47 + 33i	288	49 + 11i	352	49 + 31i	416	47 + 11i	480	47 + 31i
33	15 + 67i	97	49 + 35i	161	17 + 67i	225	47 + 35i	289	49 + 3i	353	49 + 29i	417	47 + 3i	481	47 + 29i
34	13 + 65i	98	51 + 33i	162	19 + 65i	226	45 + 33i	290	51 + 11i	354	51 + 31i	418	45 + 11i	482	45 + 31i
35	13 + 67i	99	51 + 35i	163	19 + 67i	227	45 + 35i	291	51 + 3i	355	51 + 29i	419	45 + 3i	483	45 + 29i
36	15 + 69i	100	49 + 39i	164	17 + 71i	228	47 + 39i	292	49 + 7i	356	49 + 25i	420	47 + 7i	484	47 + 25i
37	15 + 69i	101	49 + 37i	165	17 + 69i	229	47 + 37i	293	49 + 5i	357	49 + 27i	421	47 + 5i	485	47 + 27i
38	13 + 71i	102	51 + 39i	166	45 + 57i	230	45 + 39i	294	51 + 7i	358	51 + 25i	422	45 + 7i	486	45 + 25i
39	13 + 69i	103	51 + 37i	167	19 + 69i	231	45 + 37i	295	51 + 5i	359	51 + 27i	423	45 + 5i	487	45 + 27i
40	9 + 65i	104	55 + 33i	168	23 + 65i	232	41 + 33i	296	55 + 11i	360	55 + 31i	424	41 + 11i	488	41 + 31i
41	9 + 67i	105	55 + 35i	169	23 + 67i	233	41 + 35i	297	55 + 3i	361	55 + 29i	425	41 + 3i	489	41 + 29i
42	11 + 65i	106	53 + 33i	170	21 + 65i	234	43 + 33i	298	53 + 11i	362	53 + 31i	426	43 + 11i	490	43 + 31i
43	11 + 67i	107	53 + 35i	171	21 + 67i	235	43 + 35i	299	53 + 3i	363	53 + 29i	427	43 + 3i	491	43 + 29i
44	9 + 71i	108	55 + 39i	172	41 + 57i	236	41 + 39i	300	55 + 7i	364	55 + 25i	428	41 + 7i	492	41 + 25i
45	9 + 69i	109	55 + 37i	173	41 + 59i	237	41 + 37i	301	55 + 5i	365	55 + 27i	429	41 + 5i	493	41 + 27i
46	11 + 71i	110	53 + 39i	174	43 + 57i	238	43 + 39i	302	53 + 7i	366	53 + 25i	430	43 + 7i	494	43 + 25i
47	11 + 69i	111	53 + 37i	175	21 + 69i	239	43 + 37i	303	53 + 5i	367	53 + 27i	431	43 + 5i	495	43 + 27i
48	49 + 49i	112	49 + 47i	176	47 + 49i	240	47 + 47i	304	49 + 15i	368	49 + 17i	432	47 + 15i	496	47 + 17i
49	49 + 51i	113	49 + 45i	177	47 + 51i	241	47 + 45i	305	49 + 13i	369	49 + 19i	433	47 + 13i	497	47 + 19i
50	51 + 49i	114	51 + 47i	178	45 + 49i	242	45 + 47i	306	51 + 15i	370	51 + 17i	434	45 + 15i	498	45 + 17i
51	51 + 51i	115	51 + 45i	179	45 + 51i	243	45 + 45i	307	51 + 13i	371	51 + 19i	435	45 + 13i	499	45 + 19i
52	49 + 53i	116	49 + 41i	180	47 + 53i	244	47 + 41i	308	49 + 9i	372	49 + 23i	436	47 + 9i	500	47 + 23i
53	49 + 53i	117	49 + 43i	181	47 + 53i	245	47 + 43i	309	49 + 11i	373	49 + 21i	437	47 + 11i	501	47 + 21i
54	13 + 73i	118	51 + 41i	182	45 + 55i	246	45 + 41i	310	51 + 9i	374	51 + 23i	438	45 + 9i	502	45 + 23i
55	51 + 53i	119	51 + 43i	183	45 + 53i	247	45 + 43i	311	51 + 11i	375	51 + 21i	439	45 + 11i	503	45 + 21i
56	55 + 49i	120	55 + 47i	184	41 + 49i	248	41 + 47i	312	55 + 15i	376	55 + 17i	440	41 + 15i	504	41 + 17i
57	73 + 13i	121	55 + 45i	185	41 + 51i	249	41 + 45i	313	55 + 13i	377	55 + 19i	441	41 + 13i	505	41 + 19i
58	53 + 49i	122	53 + 47i	186	43 + 49i	250	43 + 47i	314	53 + 15i	378	53 + 17i	442	43 + 15i	506	43 + 17i
59	53 + 51i	123	53 + 45i	187	43 + 51i	251	43 + 45i	315	53 + 13i	379	53 + 19i	443	43 + 13i	507	43 + 19i
60	73 + 9i	124	55 + 41i	188	41 + 55i	252	41 + 41i	316	55 + 9i	380	55 + 23i	444	41 + 9i	508	41 + 23i
61	73 + 11i	125	55 + 43i	189	41 + 53i	253	41 + 43i	317	55 + 11i	381	55 + 21i	445	41 + 11i	509	41 + 21i
62	11 + 73i	126	53 + 41i	190	43 + 55i	254	43 + 41i	318	53 + 9i	382	53 + 23i	446	43 + 9i	510	43 + 23i
63	53 + 53i	127	53 + 43i	191	43 + 53i	255	43 + 43i	319	53 + 11i	383	53 + 21i	447	43 + 11i	511	43 + 21i

Fig. 19

541	7 + 53i	605	7 + 43i	669	25 + 53i	733	25 + 43i	797	7 + 11i	861	7 + 21i	925	25 + 11i	989	25 + 21i
542	5 + 55i	606	5 + 41i	670	27 + 55i	734	27 + 41i	798	5 + 9i	862	5 + 23i	926	27 + 9i	990	27 + 23i
543	5 + 53i	607	5 + 43i	671	27 + 53i	735	27 + 43i	799	5 + 11i	863	5 + 21i	927	27 + 11i	991	27 + 21i
544	15 + 63i	608	15 + 33i	672	17 + 63i	736	17 + 33i	800	15 + 1i	864	15 + 31i	928	17 + 1i	992	17 + 31i
545	15 + 61i	609	15 + 35i	673	17 + 61i	737	17 + 35i	801	15 + 3i	865	15 + 29i	929	17 + 3i	993	17 + 29i
546	13 + 63i	610	13 + 33i	674	19 + 63i	738	19 + 33i	802	13 + 1i	866	13 + 31i	930	19 + 1i	994	19 + 31i
547	13 + 61i	611	13 + 35i	675	19 + 61i	739	19 + 35i	803	13 + 3i	867	13 + 29i	931	19 + 3i	995	19 + 29i
548	15 + 57i	612	15 + 39i	676	17 + 57i	740	17 + 39i	804	15 + 7i	868	15 + 25i	932	17 + 7i	996	17 + 25i
549	15 + 59i	613	15 + 37i	677	17 + 59i	741	17 + 37i	805	15 + 5i	869	15 + 27i	933	17 + 5i	997	17 + 27i
550	13 + 57i	614	13 + 39i	678	19 + 57i	742	19 + 39i	806	13 + 7i	870	13 + 25i	934	19 + 7i	998	19 + 25i
551	13 + 59i	615	13 + 37i	679	19 + 59i	743	19 + 37i	807	13 + 5i	871	13 + 27i	935	19 + 5i	999	19 + 27i
552	9 + 63i	616	9 + 33i	680	23 + 63i	744	23 + 33i	808	9 + 1i	872	9 + 31i	936	23 + 1i	1000	23 + 31i
553	9 + 61i	617	9 + 35i	681	23 + 61i	745	23 + 35i	809	9 + 3i	873	9 + 29i	937	23 + 3i	1001	23 + 29i
554	11 + 63i	618	11 + 33i	682	21 + 63i	746	21 + 33i	810	11 + 1i	874	11 + 31i	938	21 + 1i	1002	21 + 31i
555	11 + 61i	619	11 + 35i	683	21 + 61i	747	21 + 35i	811	11 + 3i	875	11 + 29i	939	21 + 3i	1003	21 + 29i
556	9 + 57i	620	9 + 39i	684	23 + 57i	748	23 + 39i	812	9 + 7i	876	9 + 25i	940	23 + 7i	1004	23 + 25i
557	9 + 59i	621	9 + 37i	685	23 + 59i	749	23 + 37i	813	9 + 5i	877	9 + 27i	941	23 + 5i	1005	23 + 27i
558	11 + 57i	622	11 + 39i	686	21 + 57i	750	21 + 39i	814	11 + 7i	878	11 + 25i	942	21 + 7i	1006	21 + 25i
559	11 + 59i	623	11 + 37i	687	21 + 59i	751	21 + 37i	815	11 + 5i	879	11 + 27i	943	21 + 5i	1007	21 + 27i
560	15 + 49i	624	15 + 47i	688	17 + 49i	752	17 + 47i	816	15 + 15i	880	15 + 17i	944	17 + 15i	1008	17 + 17i
561	15 + 51i	625	15 + 45i	689	17 + 51i	753	17 + 45i	817	15 + 13i	881	15 + 19i	945	17 + 13i	1009	17 + 19i
562	13 + 49i	626	13 + 47i	690	19 + 49i	754	19 + 47i	818	13 + 15i	882	13 + 17i	946	19 + 15i	1010	19 + 17i
563	13 + 51i	627	13 + 45i	691	19 + 51i	755	19 + 45i	819	13 + 13i	883	13 + 19i	947	19 + 13i	1011	19 + 19i
564	15 + 55i	628	15 + 43i	692	17 + 55i	756	17 + 43i	820	15 + 9i	884	15 + 23i	948	17 + 9i	1012	17 + 23i
565	15 + 53i	629	15 + 41i	693	17 + 53i	757	17 + 41i	821	15 + 11i	885	15 + 21i	949	17 + 11i	1013	17 + 21i
566	13 + 55i	630	13 + 41i	694	19 + 55i	758	19 + 41i	822	13 + 9i	886	13 + 23i	950	19 + 9i	1014	19 + 23i
567	13 + 53i	631	13 + 43i	695	19 + 53i	759	19 + 43i	823	13 + 11i	887	13 + 21i	951	19 + 11i	1015	19 + 21i
568	9 + 49i	632	9 + 47i	696	23 + 49i	760	23 + 47i	824	9 + 15i	888	9 + 17i	952	23 + 15i	1016	23 + 17i
569	9 + 51i	633	9 + 45i	697	23 + 51i	761	23 + 45i	825	9 + 13i	889	9 + 19i	953	23 + 13i	1017	23 + 19i
570	11 + 49i	634	11 + 47i	698	21 + 49i	762	21 + 47i	826	11 + 15i	890	11 + 17i	954	21 + 15i	1018	21 + 17i
571	11 + 51i	635	11 + 45i	699	21 + 51i	763	21 + 45i	827	11 + 13i	891	11 + 19i	955	21 + 13i	1019	21 + 19i
572	9 + 55i	636	9 + 43i	700	23 + 55i	764	23 + 43i	828	9 + 9i	892	9 + 23i	956	23 + 9i	1020	23 + 23i
573	9 + 53i	637	9 + 41i	701	23 + 53i	765	23 + 41i	829	9 + 11i	893	9 + 21i	957	23 + 11i	1021	23 + 21i
574	11 + 55i	638	11 + 41i	702	21 + 55i	766	21 + 41i	830	11 + 9i	894	11 + 23i	958	21 + 9i	1022	21 + 23i
575	11 + 53i	639	11 + 43i	703	21 + 53i	767	21 + 43i	831	11 + 11i	895	11 + 21i	959	21 + 11i	1023	21 + 21i

Value = Valor, Scatter = Difusão

[illegible]

Fig. 21

1056	15 - 65	1120	49 - 33	1184	17 - 65	1248	47 - 33	1312	49 - 11	1376	49 - 31	1440	47 - 11	1504	47 - 31
1057	15 - 67	1121	49 - 35	1185	17 - 67	1249	47 - 35	1313	49 - 31	1377	49 - 29	1441	47 - 31	1505	47 - 29
1058	13 - 65	1122	51 - 33	1186	19 - 65	1250	45 - 33	1314	51 - 11	1378	51 - 31	1442	45 - 11	1506	45 - 31
1059	13 - 67	1123	51 - 35	1187	19 - 67	1251	45 - 35	1315	51 - 31	1379	51 - 29	1443	45 - 31	1507	45 - 29
1060	15 - 71	1124	49 - 39	1188	17 - 71	1252	47 - 39	1316	49 - 71	1380	49 - 25	1444	47 - 71	1508	47 - 25
1061	15 - 69	1125	49 - 37	1189	17 - 69	1253	47 - 37	1317	49 - 51	1381	49 - 27	1445	47 - 51	1509	47 - 27
1062	13 - 71	1126	51 - 39	1190	45 - 57	1254	45 - 39	1318	51 - 71	1382	51 - 25	1446	45 - 71	1510	45 - 25
1063	13 - 69	1127	51 - 37	1191	19 - 69	1255	45 - 37	1319	51 - 51	1383	51 - 27	1447	45 - 51	1511	45 - 27
1064	9 - 65	1128	55 - 33	1192	23 - 65	1256	41 - 33	1320	55 - 11	1384	55 - 31	1448	41 - 11	1512	41 - 31
1065	9 - 67	1129	55 - 35	1193	23 - 67	1257	41 - 35	1321	55 - 31	1385	55 - 29	1449	41 - 31	1513	41 - 29
1066	11 - 65	1130	53 - 33	1194	21 - 65	1258	43 - 33	1322	53 - 11	1386	53 - 31	1450	43 - 11	1514	43 - 31
1067	11 - 67	1131	53 - 35	1195	21 - 67	1259	43 - 35	1323	53 - 31	1387	53 - 29	1451	43 - 31	1515	43 - 29
1068	9 - 71	1132	55 - 39	1196	41 - 57	1260	41 - 39	1324	55 - 71	1388	55 - 25	1452	41 - 71	1516	41 - 25
1069	9 - 69	1133	55 - 37	1197	41 - 59	1261	41 - 37	1325	55 - 51	1389	55 - 27	1453	41 - 51	1517	41 - 27
1070	11 - 71	1134	53 - 39	1198	43 - 57	1262	43 - 39	1326	53 - 71	1390	53 - 25	1454	43 - 71	1518	43 - 25
1071	11 - 69	1135	53 - 37	1199	21 - 69	1263	43 - 37	1327	53 - 51	1391	53 - 27	1455	43 - 51	1519	43 - 27
1072	49 - 49	1136	49 - 47	1200	47 - 49	1264	47 - 47	1328	49 - 15	1392	49 - 17	1456	47 - 15	1520	47 - 17
1073	49 - 51	1137	49 - 45	1201	47 - 51	1265	47 - 45	1329	49 - 13	1393	49 - 19	1457	47 - 13	1521	47 - 19
1074	51 - 49	1138	51 - 47	1202	45 - 49	1266	45 - 47	1330	51 - 15	1394	51 - 17	1458	45 - 15	1522	45 - 17
1075	51 - 51	1139	51 - 45	1203	45 - 51	1267	45 - 45	1331	51 - 13	1395	51 - 19	1459	45 - 13	1523	45 - 19
1076	49 - 55	1140	49 - 43	1204	47 - 55	1268	47 - 43	1332	49 - 9	1396	49 - 23	1460	47 - 9	1524	47 - 23
1077	49 - 53	1141	49 - 43	1205	47 - 53	1269	47 - 43	1333	49 - 11	1397	49 - 21	1461	47 - 11	1525	47 - 21
1078	13 - 53	1142	51 - 43	1206	45 - 55	1270	45 - 43	1334	51 - 9	1398	51 - 23	1462	45 - 9	1526	45 - 23
1079	51 - 53	1143	51 - 43	1207	45 - 53	1271	45 - 43	1335	51 - 11	1399	51 - 21	1463	45 - 11	1527	45 - 21
1080	55 - 49	1144	55 - 47	1208	41 - 49	1272	41 - 47	1336	55 - 15	1400	55 - 17	1464	41 - 15	1528	41 - 17
1081	73 - 13	1145	55 - 45	1209	41 - 51	1273	41 - 45	1337	55 - 13	1401	55 - 19	1465	41 - 13	1529	41 - 19
1082	53 - 49	1146	53 - 47	1210	43 - 49	1274	43 - 47	1338	53 - 15	1402	53 - 17	1466	43 - 15	1530	43 - 17
1083	53 - 51	1147	53 - 45	1211	43 - 51	1275	43 - 45	1339	53 - 13	1403	53 - 19	1467	43 - 13	1531	43 - 19
1084	9 - 73	1148	55 - 43	1212	41 - 55	1276	41 - 43	1340	55 - 9	1404	55 - 23	1468	41 - 9	1532	41 - 23
1085	73 - 11	1149	55 - 43	1213	41 - 53	1277	41 - 43	1341	55 - 11	1405	55 - 21	1469	41 - 11	1533	41 - 21
1086	11 - 73	1150	53 - 41	1214	43 - 55	1278	43 - 41	1342	53 - 9	1406	53 - 23	1470	43 - 9	1534	43 - 23
1087	53 - 53	1151	53 - 43	1215	43 - 53	1279	43 - 43	1343	53 - 11	1407	53 - 21	1471	43 - 11	1535	43 - 21

Fig. 22

Value	Scatter	Value	Scatter	Value	Scatter	Value	Scatter	Value	Scatter	Value	Scatter	Value	Scatter	Value	Scatter	Value	Scatter	Value	Scatter
1536	1-63	1600	1-33	1664	31-63	1728	31-33	1792	1-11	1856	1-31	1920	31-11	1984	31-31	2048	31-11	2112	31-31
1537	1-61	1601	1-35	1665	31-61	1729	31-35	1793	1-3	1857	1-29	1921	31-3	1985	31-29	2049	31-3	2113	31-29
1538	3-63	1602	3-33	1666	29-63	1730	29-33	1794	3-11	1858	3-31	1922	29-11	1986	29-31	2050	29-11	2114	29-31
1539	3-61	1603	3-35	1667	29-61	1731	29-35	1795	3-3	1859	3-29	1923	29-3	1987	29-29	2051	29-3	2115	29-29
1540	1-57	1604	1-39	1668	31-57	1732	31-39	1796	1-7	1860	1-25	1924	31-7	1988	31-25	2052	31-7	2116	31-25
1541	1-59	1605	1-37	1669	31-59	1733	31-37	1797	1-5	1861	1-27	1925	31-5	1989	31-27	2053	31-5	2117	31-27
1542	3-57	1606	3-39	1670	29-57	1734	29-39	1798	3-7	1862	3-25	1926	29-7	1990	29-25	2054	29-7	2118	29-25
1543	3-59	1607	3-37	1671	29-59	1735	29-37	1799	3-5	1863	3-27	1927	29-5	1991	29-27	2055	29-5	2119	29-27
1544	7-63	1608	7-33	1672	25-63	1736	25-33	1800	7-11	1864	7-31	1928	25-11	1992	25-31	2056	25-11	2120	25-31
1545	7-61	1609	7-35	1673	25-61	1737	25-35	1801	7-3	1865	7-29	1929	25-3	1993	25-29	2057	25-3	2121	25-29
1546	5-63	1610	5-33	1674	27-63	1738	27-33	1802	5-1	1866	5-31	1930	27-1	1994	27-31	2058	27-1	2122	27-31
1547	5-61	1611	5-35	1675	27-61	1739	27-35	1803	5-3	1867	5-29	1931	27-3	1995	27-29	2059	27-3	2123	27-29
1548	7-57	1612	7-39	1676	25-57	1740	25-39	1804	7-7	1868	7-25	1932	25-7	1996	25-25	2060	25-7	2124	25-25
1549	7-59	1613	7-37	1677	25-59	1741	25-37	1805	7-5	1869	7-27	1933	25-5	1997	25-27	2061	25-5	2125	25-27
1550	5-57	1614	5-39	1678	27-57	1742	27-39	1806	5-7	1870	5-25	1934	27-7	1998	27-25	2062	27-7	2126	27-25
1551	5-59	1615	5-37	1679	27-59	1743	27-37	1807	5-5	1871	5-27	1935	27-5	1999	27-27	2063	27-5	2127	27-27
1552	1-49	1616	1-47	1680	31-49	1744	31-47	1808	1-15	1872	1-17	1936	31-15	2000	31-17	2064	31-15	2128	31-17
1553	1-51	1617	1-45	1681	31-51	1745	31-45	1809	1-13	1873	1-19	1937	31-13	2001	31-19	2065	31-13	2129	31-19
1554	3-49	1618	3-47	1682	29-49	1746	29-47	1810	3-15	1874	3-17	1938	29-15	2002	29-17	2066	29-15	2130	29-17
1555	3-51	1619	3-45	1683	29-51	1747	29-45	1811	3-13	1875	3-19	1939	29-13	2003	29-19	2067	29-13	2131	29-19
1556	1-55	1620	1-41	1684	31-55	1748	31-41	1812	1-9	1876	1-23	1940	31-9	2004	31-23	2068	31-9	2132	31-23
1557	1-53	1621	1-43	1685	31-53	1749	31-43	1813	1-11	1877	1-21	1941	31-11	2005	31-21	2069	31-11	2133	31-21
1558	3-55	1622	3-41	1686	29-55	1750	29-41	1814	3-9	1878	3-23	1942	29-9	2006	29-23	2070	29-9	2134	29-23
1559	3-53	1623	3-43	1687	29-53	1751	29-43	1815	3-11	1879	3-21	1943	29-11	2007	29-21	2071	29-11	2135	29-21
1560	7-49	1624	7-47	1688	25-49	1752	25-47	1816	7-15	1880	7-17	1944	25-15	2008	25-17	2072	25-15	2136	25-17
1561	7-51	1625	7-45	1689	25-51	1753	25-45	1817	7-13	1881	7-19	1945	25-13	2009	25-19	2073	25-13	2137	25-19
1562	5-49	1626	5-47	1690	27-49	1754	27-47	1818	5-15	1882	5-17	1946	27-15	2010	27-17	2074	27-15	2138	27-17
1563	5-51	1627	5-45	1691	27-51	1755	27-45	1819	5-13	1883	5-19	1947	27-13	2011	27-19	2075	27-13	2139	27-19
1564	7-53	1628	7-41	1692	25-53	1756	25-41	1820	7-9	1884	7-23	1948	25-9	2012	25-23	2076	25-9	2140	25-23
1565	7-53	1629	7-43	1693	25-53	1757	25-43	1821	7-11	1885	7-21	1949	25-11	2013	25-21	2077	25-11	2141	25-21
1566	5-55	1630	5-41	1694	27-55	1758	27-41	1822	5-9	1886	5-23	1950	27-9	2014	27-23	2078	27-9	2142	27-23
1567	5-53	1631	5-43	1695	27-53	1759	27-43	1823	5-11	1887	5-21	1951	27-11	2015	27-21	2079	27-11	2143	27-21
1568	15-63	1632	15-33	1696	17-63	1760	17-33	1824	15-11	1888	15-31	1952	17-11	2016	17-31	2080	17-11	2144	17-31
1569	15-61	1633	15-35	1697	17-61	1761	17-35	1825	15-3	1889	15-29	1953	17-3	2017	17-29	2081	17-3	2145	17-29
1570	13-63	1634	13-33	1698	19-63	1762	19-33	1826	13-11	1890	13-31	1954	19-11	2018	19-31	2082	19-11	2146	19-31

Value = Valor, Scatter = Difusão

Fig. 23

1571	13 - 61	1635	13 - 35	1699	19 - 61	1763	19 - 35	1827	13 - 3	1891	13 - 29	1955	19 - 3	2019	19 - 29
1572	15 - 57	1636	15 - 39	1700	17 - 57	1764	17 - 39	1828	15 - 7	1892	15 - 25	1956	17 - 7	2020	17 - 25
1573	15 - 59	1637	15 - 37	1701	17 - 59	1765	17 - 37	1829	15 - 5	1893	15 - 27	1957	17 - 5	2021	17 - 27
1574	13 - 57	1638	13 - 39	1702	19 - 57	1766	19 - 39	1830	13 - 7	1894	13 - 25	1958	19 - 7	2022	19 - 25
1575	13 - 59	1639	13 - 37	1703	19 - 59	1767	19 - 37	1831	13 - 5	1895	13 - 27	1959	19 - 5	2023	19 - 27
1576	9 - 63	1640	9 - 33	1704	23 - 63	1768	23 - 33	1832	9 - 1	1896	9 - 31	1960	23 - 1	2024	23 - 31
1577	9 - 61	1641	9 - 35	1705	23 - 61	1769	23 - 35	1833	9 - 3	1897	9 - 29	1961	23 - 3	2025	23 - 29
1578	11 - 63	1642	11 - 33	1706	21 - 63	1770	21 - 33	1834	11 - 1	1898	11 - 31	1962	21 - 1	2026	21 - 31
1579	11 - 61	1643	11 - 35	1707	21 - 61	1771	21 - 35	1835	11 - 3	1899	11 - 29	1963	21 - 3	2027	21 - 29
1580	9 - 57	1644	9 - 39	1708	23 - 57	1772	23 - 39	1836	9 - 7	1900	9 - 25	1964	23 - 7	2028	23 - 25
1581	9 - 59	1645	9 - 37	1709	23 - 59	1773	23 - 37	1837	9 - 5	1901	9 - 27	1965	23 - 5	2029	23 - 27
1582	11 - 57	1646	11 - 39	1710	21 - 57	1774	21 - 39	1838	11 - 7	1902	11 - 25	1966	21 - 7	2030	21 - 25
1583	11 - 59	1647	11 - 37	1711	21 - 59	1775	21 - 37	1839	11 - 5	1903	11 - 27	1967	21 - 5	2031	21 - 27
1584	15 - 49	1648	15 - 47	1712	17 - 49	1776	17 - 47	1840	15 - 15	1904	15 - 17	1968	17 - 15	2032	17 - 17
1585	15 - 51	1649	15 - 45	1713	17 - 51	1777	17 - 45	1841	15 - 13	1905	15 - 19	1969	17 - 13	2033	17 - 19
1586	13 - 49	1650	13 - 47	1714	19 - 49	1778	19 - 47	1842	13 - 15	1906	13 - 17	1970	19 - 15	2034	19 - 17
1587	13 - 51	1651	13 - 45	1715	19 - 51	1779	19 - 45	1843	13 - 13	1907	13 - 19	1971	19 - 13	2035	19 - 19
1588	15 - 55	1652	15 - 41	1716	17 - 55	1780	17 - 41	1844	15 - 9	1908	15 - 23	1972	17 - 9	2036	17 - 23
1589	15 - 53	1653	15 - 43	1717	17 - 53	1781	17 - 43	1845	15 - 11	1909	15 - 21	1973	17 - 11	2037	17 - 21
1590	13 - 55	1654	13 - 41	1718	19 - 55	1782	19 - 41	1846	13 - 9	1910	13 - 23	1974	19 - 9	2038	19 - 23
1591	13 - 53	1655	13 - 43	1719	19 - 53	1783	19 - 43	1847	13 - 11	1911	13 - 21	1975	19 - 11	2039	19 - 21
1592	9 - 49	1656	9 - 47	1720	23 - 49	1784	23 - 47	1848	9 - 15	1912	9 - 17	1976	23 - 15	2040	23 - 17
1593	9 - 51	1657	9 - 45	1721	23 - 51	1785	23 - 45	1849	9 - 13	1913	9 - 19	1977	23 - 13	2041	23 - 19
1594	11 - 49	1658	11 - 47	1722	21 - 49	1786	21 - 47	1850	11 - 15	1914	11 - 17	1978	21 - 15	2042	21 - 17
1595	11 - 51	1659	11 - 45	1723	21 - 51	1787	21 - 45	1851	11 - 13	1915	11 - 19	1979	21 - 13	2043	21 - 19
1596	9 - 55	1660	9 - 41	1724	23 - 55	1788	23 - 41	1852	9 - 9	1916	9 - 23	1980	23 - 9	2044	23 - 23
1597	9 - 53	1661	9 - 43	1725	23 - 53	1789	23 - 43	1853	9 - 11	1917	9 - 21	1981	23 - 11	2045	23 - 21
1598	11 - 55	1662	11 - 41	1726	21 - 55	1790	21 - 41	1854	11 - 9	1918	11 - 23	1982	21 - 9	2046	21 - 23
1599	11 - 53	1663	11 - 43	1727	21 - 53	1791	21 - 43	1855	11 - 11	1919	11 - 21	1983	21 - 11	2047	21 - 21

Fig. 25

2081	-15 + 67	2145	-49 + 35	2209	-17 + 67	2273	-47 + 35	2337	-49 + 31	2401	-49 + 29	2465	-47 + 31	2529	-47 + 29
2082	-13 + 65	2146	-51 + 33	2210	-19 + 65	2274	-45 + 33	2338	-51 + 31	2402	-51 + 31	2466	-45 + 31	2530	-45 + 31
2083	-13 + 67	2147	-51 + 35	2211	-19 + 67	2275	-45 + 35	2339	-51 + 31	2403	-51 + 29	2467	-45 + 31	2531	-45 + 29
2084	-15 + 71	2148	-49 + 39	2212	-17 + 71	2276	-47 + 39	2340	-49 + 71	2404	-49 + 25	2468	-47 + 71	2532	-47 + 25
2085	-15 + 69	2149	-49 + 37	2213	-17 + 69	2277	-47 + 37	2341	-49 + 51	2405	-49 + 27	2469	-47 + 51	2533	-47 + 27
2086	-13 + 71	2150	-51 + 39	2214	-45 + 57	2278	-45 + 39	2342	-51 + 71	2406	-51 + 25	2470	-45 + 71	2534	-45 + 25
2087	-13 + 69	2151	-51 + 37	2215	-19 + 69	2279	-45 + 37	2343	-51 + 51	2407	-51 + 27	2471	-45 + 51	2535	-45 + 27
2088	-9 + 65	2152	-55 + 33	2216	-23 + 65	2280	-41 + 33	2344	-55 + 11	2408	-55 + 31	2472	-41 + 11	2536	-41 + 31
2089	-9 + 67	2153	-55 + 35	2217	-23 + 67	2281	-41 + 35	2345	-55 + 31	2409	-55 + 29	2473	-41 + 31	2537	-41 + 29
2090	-11 + 65	2154	-53 + 33	2218	-21 + 65	2282	-43 + 33	2346	-53 + 11	2410	-53 + 31	2474	-43 + 11	2538	-43 + 31
2091	-11 + 67	2155	-53 + 35	2219	-21 + 67	2283	-43 + 35	2347	-53 + 31	2411	-53 + 29	2475	-43 + 31	2539	-43 + 29
2092	-9 + 71	2156	-55 + 39	2220	-41 + 57	2284	-41 + 39	2348	-55 + 71	2412	-55 + 25	2476	-41 + 71	2540	-41 + 25
2093	-9 + 69	2157	-55 + 37	2221	-41 + 59	2285	-41 + 37	2349	-55 + 51	2413	-55 + 27	2477	-41 + 51	2541	-41 + 27
2094	-11 + 71	2158	-53 + 39	2222	-43 + 57	2286	-43 + 39	2350	-53 + 71	2414	-53 + 25	2478	-43 + 71	2542	-43 + 25
2095	-11 + 69	2159	-53 + 37	2223	-21 + 69	2287	-43 + 37	2351	-53 + 51	2415	-53 + 27	2479	-43 + 51	2543	-43 + 27
2096	-49 + 49	2160	-49 + 47	2224	-47 + 49	2288	-47 + 47	2352	-49 + 15	2416	-49 + 17	2480	-47 + 15	2544	-47 + 17
2097	-49 + 51	2161	-49 + 45	2225	-47 + 51	2289	-47 + 45	2353	-49 + 13	2417	-49 + 19	2481	-47 + 13	2545	-47 + 19
2098	-51 + 49	2162	-51 + 47	2226	-45 + 49	2290	-45 + 47	2354	-51 + 15	2418	-51 + 17	2482	-45 + 15	2546	-45 + 17
2099	-51 + 51	2163	-51 + 45	2227	-45 + 51	2291	-45 + 45	2355	-51 + 13	2419	-51 + 19	2483	-45 + 13	2547	-45 + 19
2100	-49 + 55	2164	-49 + 41	2228	-47 + 55	2292	-47 + 41	2356	-49 + 9	2420	-49 + 23	2484	-47 + 9	2548	-47 + 23
2101	-49 + 53	2165	-49 + 43	2229	-47 + 53	2293	-47 + 43	2357	-49 + 11	2421	-49 + 21	2485	-47 + 11	2549	-47 + 21
2102	-13 + 73	2166	-51 + 41	2230	-45 + 53	2294	-45 + 41	2358	-51 + 9	2422	-51 + 23	2486	-45 + 9	2550	-45 + 23
2103	-51 + 53	2167	-51 + 43	2231	-45 + 53	2295	-45 + 43	2359	-51 + 11	2423	-51 + 21	2487	-45 + 11	2551	-45 + 21
2104	-55 + 49	2168	-55 + 47	2232	-41 + 49	2296	-41 + 47	2360	-55 + 15	2424	-55 + 17	2488	-41 + 15	2552	-41 + 17
2105	-73 + 13	2169	-55 + 45	2233	-41 + 51	2297	-41 + 45	2361	-55 + 13	2425	-55 + 19	2489	-41 + 13	2553	-41 + 19
2106	-53 + 49	2170	-53 + 47	2234	-43 + 49	2298	-43 + 47	2362	-53 + 15	2426	-53 + 17	2490	-43 + 15	2554	-43 + 17
2107	-53 + 51	2171	-53 + 45	2235	-43 + 51	2299	-43 + 45	2363	-53 + 13	2427	-53 + 19	2491	-43 + 13	2555	-43 + 19
2108	-9 + 73	2172	-55 + 41	2236	-41 + 55	2300	-41 + 41	2364	-55 + 9	2428	-55 + 23	2492	-41 + 9	2556	-41 + 23
2109	-73 + 11	2173	-55 + 43	2237	-41 + 53	2301	-41 + 43	2365	-55 + 11	2429	-55 + 21	2493	-41 + 11	2557	-41 + 21
2110	-11 + 73	2174	-53 + 41	2238	-43 + 55	2302	-43 + 41	2366	-53 + 9	2430	-53 + 23	2494	-43 + 9	2558	-43 + 23
2111	-53 + 53	2175	-53 + 43	2239	-43 + 53	2303	-43 + 43	2367	-53 + 11	2431	-53 + 21	2495	-43 + 11	2559	-43 + 21

Value = Valor, Scatter = Difusão

Value	Scatter	Value	Scatter	Value	Scatter	Value	Scatter	Value	Scatter	Value	Scatter	Value	Scatter	Value	Scatter
2550	-1 + 63i	2624	-1 + 33i	2688	-31 + 63i	2752	-31 + 33i	2816	-1 + 1i	2880	-1 + 31i	2944	-31 + 1i	3008	-31 + 31i
2561	-1 + 61i	2625	-1 + 35i	2689	-31 + 61i	2753	-31 + 35i	2817	-1 + 3i	2881	-1 + 29i	2945	-31 + 3i	3009	-31 + 29i
2562	-3 + 63i	2626	-3 + 33i	2690	-29 + 63i	2754	-29 + 33i	2818	-3 + 1i	2882	-3 + 31i	2946	-29 + 1i	3010	-29 + 31i
2563	-3 + 61i	2627	-3 + 35i	2691	-29 + 61i	2755	-29 + 35i	2819	-3 + 3i	2883	-3 + 29i	2947	-29 + 3i	3011	-29 + 29i
2564	-1 + 57i	2628	-1 + 39i	2692	-31 + 57i	2756	-31 + 39i	2820	-1 + 7i	2884	-1 + 25i	2948	-31 + 7i	3012	-31 + 25i
2565	-1 + 59i	2629	-1 + 37i	2693	-31 + 59i	2757	-31 + 37i	2821	-1 + 5i	2885	-1 + 27i	2949	-31 + 5i	3013	-31 + 27i
2566	-3 + 57i	2630	-3 + 39i	2694	-29 + 57i	2758	-29 + 39i	2822	-3 + 7i	2886	-3 + 25i	2950	-29 + 7i	3014	-29 + 25i
2567	-3 + 59i	2631	-3 + 37i	2695	-29 + 59i	2759	-29 + 37i	2823	-3 + 5i	2887	-3 + 27i	2951	-29 + 5i	3015	-29 + 27i
2568	-7 + 63i	2632	-7 + 33i	2696	-25 + 63i	2760	-25 + 33i	2824	-7 + 1i	2888	-7 + 31i	2952	-25 + 1i	3016	-25 + 31i
2569	-7 + 61i	2633	-7 + 35i	2697	-25 + 61i	2761	-25 + 35i	2825	-7 + 3i	2889	-7 + 29i	2953	-25 + 3i	3017	-25 + 29i
2570	-5 + 63i	2634	-5 + 33i	2698	-27 + 63i	2762	-27 + 33i	2826	-5 + 1i	2890	-5 + 31i	2954	-27 + 1i	3018	-27 + 31i
2571	-5 + 61i	2635	-5 + 35i	2699	-27 + 61i	2763	-27 + 35i	2827	-5 + 3i	2891	-5 + 29i	2955	-27 + 3i	3019	-27 + 29i
2572	-7 + 57i	2636	-7 + 39i	2700	-25 + 57i	2764	-25 + 39i	2828	-7 + 7i	2892	-7 + 25i	2956	-25 + 7i	3020	-25 + 25i
2573	-7 + 59i	2637	-7 + 37i	2701	-25 + 59i	2765	-25 + 37i	2829	-7 + 5i	2893	-7 + 27i	2957	-25 + 5i	3021	-25 + 27i
2574	-5 + 57i	2638	-5 + 39i	2702	-27 + 57i	2766	-27 + 39i	2830	-5 + 7i	2894	-5 + 25i	2958	-27 + 7i	3022	-27 + 25i
2575	-5 + 59i	2639	-5 + 37i	2703	-27 + 59i	2767	-27 + 37i	2831	-5 + 5i	2895	-5 + 27i	2959	-27 + 5i	3023	-27 + 27i
2576	-1 + 49i	2640	-1 + 47i	2704	-31 + 49i	2768	-31 + 47i	2832	-1 + 15i	2896	-1 + 17i	2960	-31 + 15i	3024	-31 + 17i
2577	-1 + 51i	2641	-1 + 45i	2705	-31 + 51i	2769	-31 + 45i	2833	-1 + 13i	2897	-1 + 19i	2961	-31 + 13i	3025	-31 + 19i
2578	-3 + 49i	2642	-3 + 47i	2706	-29 + 49i	2770	-29 + 47i	2834	-3 + 15i	2898	-3 + 17i	2962	-29 + 15i	3026	-29 + 17i
2579	-3 + 51i	2643	-3 + 45i	2707	-29 + 51i	2771	-29 + 45i	2835	-3 + 13i	2899	-3 + 19i	2963	-29 + 13i	3027	-29 + 19i
2580	-1 + 53i	2644	-1 + 43i	2708	-31 + 53i	2772	-31 + 43i	2836	-1 + 9i	2900	-1 + 23i	2964	-31 + 9i	3028	-31 + 23i
2581	-1 + 51i	2645	-1 + 45i	2709	-31 + 51i	2773	-31 + 45i	2837	-1 + 11i	2901	-1 + 21i	2965	-31 + 11i	3029	-31 + 21i
2582	-3 + 53i	2646	-3 + 43i	2710	-29 + 53i	2774	-29 + 43i	2838	-3 + 9i	2902	-3 + 23i	2966	-29 + 9i	3030	-29 + 23i
2583	-3 + 51i	2647	-3 + 45i	2711	-29 + 51i	2775	-29 + 45i	2839	-3 + 11i	2903	-3 + 21i	2967	-29 + 11i	3031	-29

Fig. 27

2591	-5 + 53	2655	-5 + 43	2739	-27 + 53	2783	-27 + 43	2847	-5 + 11	2911	-5 + 21	2975	-27 + 11	3039	-27 + 21
2592	-15 + 63	2656	-15 + 33	2720	-17 + 63	2784	-17 + 33	2848	-15 + 11	2912	-15 + 31	2976	-17 + 11	3040	-17 + 31
2593	-15 + 61	2657	-15 + 35	2721	-17 + 61	2785	-17 + 35	2849	-15 + 3	2913	-15 + 29	2977	-17 + 3	3041	-17 + 29
2594	-13 + 63	2658	-13 + 33	2722	-19 + 63	2786	-19 + 33	2850	-13 + 1	2914	-13 + 31	2978	-19 + 1	3042	-19 + 31
2595	-13 + 61	2659	-13 + 35	2723	-19 + 61	2787	-19 + 35	2851	-13 + 3	2915	-13 + 29	2979	-19 + 3	3043	-19 + 29
2596	-15 + 57	2660	-15 + 39	2724	-17 + 57	2788	-17 + 39	2852	-15 + 7	2916	-15 + 25	2980	-17 + 7	3044	-17 + 25
2597	-15 + 59	2661	-15 + 37	2725	-17 + 59	2789	-17 + 37	2853	-15 + 5	2917	-15 + 27	2981	-17 + 5	3045	-17 + 27
2598	-13 + 57	2662	-13 + 39	2726	-19 + 57	2790	-19 + 39	2854	-13 + 7	2918	-13 + 25	2982	-19 + 7	3046	-19 + 25
2599	-13 + 59	2663	-13 + 37	2727	-19 + 59	2791	-19 + 37	2855	-13 + 5	2919	-13 + 27	2983	-19 + 5	3047	-19 + 27
2600	-9 + 63	2664	-9 + 33	2728	-23 + 63	2792	-23 + 33	2856	-9 + 1	2920	-9 + 31	2984	-23 + 1	3048	-23 + 31
2601	-9 + 61	2665	-9 + 35	2729	-23 + 61	2793	-23 + 35	2857	-9 + 3	2921	-9 + 29	2985	-23 + 3	3049	-23 + 29
2602	-11 + 63	2666	-11 + 33	2730	-21 + 63	2794	-21 + 33	2858	-11 + 1	2922	-11 + 31	2986	-21 + 1	3050	-21 + 31
2603	-11 + 61	2667	-11 + 35	2731	-21 + 61	2795	-21 + 35	2859	-11 + 3	2923	-11 + 29	2987	-21 + 3	3051	-21 + 29
2604	-9 + 57	2668	-9 + 39	2732	-23 + 57	2796	-23 + 39	2860	-9 + 7	2924	-9 + 25	2988	-23 + 7	3052	-23 + 25
2605	-9 + 59	2669	-9 + 37	2733	-23 + 59	2797	-23 + 37	2861	-9 + 5	2925	-9 + 27	2989	-23 + 5	3053	-23 + 27
2606	-11 + 57	2670	-11 + 39	2734	-21 + 57	2798	-21 + 39	2862	-11 + 7	2926	-11 + 25	2990	-21 + 7	3054	-21 + 25
2607	-11 + 59	2671	-11 + 37	2735	-21 + 59	2799	-21 + 37	2863	-11 + 5	2927	-11 + 27	2991	-21 + 5	3055	-21 + 27
2608	-15 + 49	2672	-15 + 47	2736	-17 + 49	2800	-17 + 47	2864	-15 + 15	2928	-15 + 17	2992	-17 + 15	3056	-17 + 17
2609	-15 + 51	2673	-15 + 45	2737	-17 + 51	2801	-17 + 45	2865	-15 + 13	2929	-15 + 19	2993	-17 + 13	3057	-17 + 19
2610	-13 + 49	2674	-13 + 47	2738	-19 + 49	2802	-19 + 47	2866	-13 + 15	2930	-13 + 17	2994	-19 + 15	3058	-19 + 17
2611	-13 + 51	2675	-13 + 45	2739	-19 + 51	2803	-19 + 45	2867	-13 + 13	2931	-13 + 19	2995	-19 + 13	3059	-19 + 19
2612	-15 + 55	2676	-15 + 41	2740	-17 + 55	2804	-17 + 41	2868	-15 + 9	2932	-15 + 23	2996	-17 + 9	3060	-17 + 23
2613	-15 + 53	2677	-15 + 43	2741	-17 + 53	2805	-17 + 43	2869	-15 + 11	2933	-15 + 21	2997	-17 + 11	3061	-17 + 21
2614	-13 + 55	2678	-13 + 41	2742	-19 + 55	2806	-19 + 41	2870	-13 + 9	2934	-13 + 23	2998	-19 + 9	3062	-19 + 23
2615	-13 + 53	2679	-13 + 43	2743	-19 + 53	2807	-19 + 43	2871	-13 + 11	2935	-13 + 21	2999	-19 + 11	3063	-19 + 21
2616	-9 + 49	2680	-9 + 47	2744	-23 + 49	2808	-23 + 47	2872	-9 + 15	2936	-9 + 17	3000	-23 + 15	3064	-23 + 17
2617	-9 + 51	2681	-9 + 45	2745	-23 + 51	2809	-23 + 45	2873	-9 + 13	2937	-9 + 19	3001	-23 + 13	3065	-23 + 19
2618	-11 + 49	2682	-11 + 47	2746	-21 + 49	2810	-21 + 47	2874	-11 + 15	2938	-11 + 17	3002	-21 + 15	3066	-21 + 17
2619	-11 + 51	2683	-11 + 45	2747	-21 + 51	2811	-21 + 45	2875	-11 + 13	2939	-11 + 19	3003	-21 + 13	3067	-21 + 19
2620	-9 + 55	2684	-9 + 41	2748	-23 + 55	2812	-23 + 41	2876	-9 + 9	2940	-9 + 23	3004	-23 + 9	3068	-23 + 23
2621	-9 + 53	2685	-9 + 43	2749	-23 + 53	2813	-23 + 43	2877	-9 + 11	2941	-9 + 21	3005	-23 + 11	3069	-23 + 21
2622	-11 + 55	2686	-11 + 41	2750	-21 + 55	2814	-21 + 41	2878	-11 + 9	2942	-11 + 23	3006	-21 + 9	3070	-21 + 23
2623	-11 + 53	2687	-11 + 43	2751	-21 + 53	2815	-21 + 43	2879	-11 + 11	2943	-11 + 21	3007	-21 + 11	3071	-21 + 21

Fig. 28

Value	Scatter	Value	Scatter	Value	Scatter	Value	Scatter	Value	Scatter	Value	Scatter	Value	Scatter	Value	Scatter
3072	-65 - 11	3136	-63 - 33	3200	-33 - 63	3264	-33 - 33	3328	-63 - 11	3392	-63 - 31	3456	-33 - 11	3520	-33 - 31
3073	-65 - 31	3137	-65 - 29	3201	-33 - 61	3265	-33 - 35	3329	-63 - 31	3393	-63 - 29	3457	-33 - 31	3521	-33 - 29
3074	-3 - 65	3138	-61 - 33	3202	-29 - 65	3266	-35 - 35	3330	-61 - 11	3394	-61 - 31	3458	-35 - 11	3522	-35 - 31
3075	-67 - 31	3139	-61 - 35	3203	-35 - 61	3267	-35 - 35	3331	-61 - 31	3395	-61 - 29	3459	-35 - 31	3523	-35 - 29
3076	-65 - 71	3140	-65 - 25	3204	-33 - 57	3268	-33 - 39	3332	-63 - 71	3396	-63 - 25	3460	-33 - 71	3524	-33 - 25
3077	-65 - 51	3141	-65 - 27	3205	-33 - 59	3269	-33 - 37	3333	-63 - 51	3397	-63 - 27	3461	-33 - 51	3525	-33 - 27
3078	-67 - 71	3142	-67 - 25	3206	-35 - 57	3270	-35 - 39	3334	-61 - 71	3398	-61 - 25	3462	-35 - 71	3526	-35 - 25
3079	-67 - 51	3143	-61 - 37	3207	-35 - 59	3271	-35 - 37	3335	-61 - 51	3399	-61 - 27	3463	-35 - 51	3527	-35 - 27
3080	-7 - 65	3144	-57 - 33	3208	-25 - 65	3272	-39 - 33	3336	-57 - 11	3400	-57 - 31	3464	-39 - 11	3528	-39 - 31
3081	-7 - 67	3145	-57 - 35	3209	-25 - 67	3273	-39 - 35	3337	-57 - 31	3401	-57 - 29	3465	-39 - 31	3529	-39 - 29
3082	-5 - 65	3146	-59 - 33	3210	-27 - 65	3274	-37 - 33	3338	-59 - 11	3402	-59 - 31	3466	-37 - 11	3530	-37 - 31
3083	-5 - 67	3147	-59 - 35	3211	-37 - 61	3275	-37 - 35	3339	-59 - 31	3403	-59 - 29	3467	-37 - 31	3531	-37 - 29
3084	-71 - 71	3148	-57 - 39	3212	-39 - 57	3276	-39 - 39	3340	-57 - 71	3404	-57 - 25	3468	-39 - 71	3532	-39 - 25
3085	-7 - 69	3149	-57 - 37	3213	-39 - 59	3277	-39 - 37	3341	-57 - 51	3405	-57 - 27	3469	-39 - 51	3533	-39 - 27
3086	-69 - 71	3150	-59 - 39	3214	-37 - 57	3278	-37 - 39	3342	-59 - 71	3406	-59 - 25	3470	-37 - 71	3534	-37 - 25
3087	-69 - 51	3151	-59 - 37	3215	-37 - 59	3279	-37 - 37	3343	-59 - 51	3407	-59 - 27	3471	-37 - 51	3535	-37 - 27
3088	-65 - 15	3152	-65 - 17	3216	-33 - 49	3280	-33 - 47	3344	-63 - 15	3408	-63 - 17	3472	-33 - 15	3536	-33 - 17
3089	-65 - 13	3153	-65 - 19	3217	-33 - 51	3281	-33 - 45	3345	-63 - 13	3409	-63 - 19	3473	-33 - 13	3537	-33 - 19
3090	-67 - 15	3154	-67 - 17	3218	-35 - 49	3282	-35 - 47	3346	-61 - 15	3410	-61 - 17	3474	-35 - 15	3538	-35 - 17
3091	-67 - 13	3155	-67 - 19	3219	-35 - 51	3283	-35 - 45	3347	-61 - 13	3411	-61 - 19	3475	-35 - 13	3539	-35 - 19
3092	-65 - 9	3156	-65 - 23	3220	-33 - 55	3284	-33 - 41	3348	-63 - 9	3412	-63 - 23	3476	-33 - 9	3540	-33 - 23
3093	-65 - 11	3157	-65 - 21	3221	-33 - 53	3285	-33 - 43	3349	-63 - 11	3413	-63 - 21	3477	-33 - 11	3541	-33 - 21
3094	-67 - 9	3158	-67 - 23	3222	-35 - 55	3286	-35 - 41	3350	-61 - 9	3414	-61 - 23	3478	-35 - 9	3542	-35 - 23
3095	-67 - 11	3159	-67 - 21	3223	-35 - 53	3287	-35 - 43	3351	-61 - 11	3415	-61 - 21	3479	-35 - 11	3543	-35 - 21
3096	-71 - 15	3160	-71 - 17	3224	-39 - 49	3288	-39 - 47	3352	-57 - 15	3416	-57 - 17	3480	-39 - 15	3544	-39 - 17
3097	-71 - 13	3161	-57 - 45	3225	-39 - 51	3289	-39 - 45	3353	-57 - 13	3417	-57 - 19	3481	-39 - 13	3545	-39 - 19
3098	-69 - 15	3162	-69 - 17	3226	-37 - 49	3290	-37 - 47	3354	-59 - 15	3418	-59 - 17	3482	-37 - 15	3546	-37 - 17
3099	-69 - 13	3163	-69 - 19	3227	-37 - 51	3291	-37 - 45	3355	-59 - 13	3419	-59 - 19	3483	-37 - 13	3547	-37 - 19
3100	-71 - 9	3164	-57 - 41	3228	-39 - 55	3292	-39 - 41	3356	-57 - 9	3420	-57 - 23	3484	-39 - 9	3548	-39 - 23

Value = Valor, Scatter = Difusão

Fig. 29

3101	-71 - 11	3165	-57 - 43	3229	-39 - 53	3293	-39 - 43	3357	-57 - 11	3421	-57 - 21	3485	-39 - 11	3549	-39 - 21
3102	-69 - 9	3166	-59 - 41	3230	-37 - 55	3294	-37 - 41	3358	-59 - 9	3422	-59 - 23	3486	-37 - 9	3550	-37 - 23
3103	-69 - 11	3167	-69 - 21	3231	-37 - 53	3295	-37 - 43	3359	-59 - 11	3423	-59 - 21	3487	-37 - 11	3551	-37 - 21
3104	-15 - 65	3168	-49 - 33	3232	-17 - 65	3296	-47 - 33	3360	-49 - 11	3424	-49 - 31	3488	-47 - 11	3552	-47 - 31
3105	-15 - 67	3169	-49 - 35	3233	-17 - 67	3297	-47 - 35	3361	-49 - 3	3425	-49 - 29	3489	-47 - 3	3553	-47 - 29
3106	-13 - 65	3170	-51 - 33	3234	-19 - 65	3298	-45 - 33	3362	-51 - 11	3426	-51 - 31	3490	-45 - 11	3554	-45 - 31
3107	-13 - 67	3171	-51 - 35	3235	-19 - 67	3299	-45 - 35	3363	-51 - 3	3427	-51 - 29	3491	-45 - 3	3555	-45 - 29
3108	-15 - 71	3172	-49 - 39	3236	-17 - 71	3300	-47 - 39	3364	-49 - 7	3428	-49 - 25	3492	-47 - 7	3556	-47 - 25
3109	-15 - 69	3173	-49 - 37	3237	-17 - 69	3301	-47 - 37	3365	-49 - 5	3429	-49 - 27	3493	-47 - 5	3557	-47 - 27
3110	-13 - 71	3174	-51 - 39	3238	-45 - 57	3302	-45 - 39	3366	-51 - 7	3430	-51 - 25	3494	-45 - 7	3558	-45 - 25
3111	-13 - 69	3175	-51 - 37	3239	-19 - 69	3303	-45 - 37	3367	-51 - 5	3431	-51 - 27	3495	-45 - 5	3559	-45 - 27
3112	-9 - 65	3176	-55 - 33	3240	-23 - 65	3304	-41 - 33	3368	-55 - 11	3432	-55 - 31	3496	-41 - 11	3560	-41 - 31
3113	-9 - 67	3177	-55 - 35	3241	-23 - 67	3305	-41 - 35	3369	-55 - 3	3433	-55 - 29	3497	-41 - 3	3561	-41 - 29
3114	-11 - 65	3178	-53 - 33	3242	-21 - 65	3306	-43 - 33	3370	-53 - 11	3434	-53 - 31	3498	-43 - 11	3562	-43 - 31
3115	-11 - 67	3179	-53 - 35	3243	-21 - 67	3307	-43 - 35	3371	-53 - 3	3435	-53 - 29	3499	-43 - 3	3563	-43 - 29
3116	-9 - 71	3180	-55 - 39	3244	-41 - 57	3308	-41 - 39	3372	-55 - 7	3436	-55 - 25	3500	-41 - 7	3564	-41 - 25
3117	-9 - 69	3181	-55 - 37	3245	-41 - 59	3309	-41 - 37	3373	-55 - 5	3437	-55 - 27	3501	-41 - 5	3565	-41 - 27
3118	-11 - 71	3182	-53 - 39	3246	-43 - 57	3310	-43 - 39	3374	-53 - 7	3438	-53 - 25	3502	-43 - 7	3566	-43 - 25
3119	-11 - 69	3183	-53 - 37	3247	-21 - 69	3311	-43 - 37	3375	-53 - 5	3439	-53 - 27	3503	-43 - 5	3567	-43 - 27
3120	-49 - 49	3184	-49 - 47	3248	-47 - 49	3312	-47 - 47	3376	-49 - 15	3440	-49 - 17	3504	-47 - 15	3568	-47 - 17
3121	-49 - 51	3185	-49 - 45	3249	-47 - 51	3313	-47 - 45	3377	-49 - 13	3441	-49 - 19	3505	-47 - 13	3569	-47 - 19
3122	-51 - 49	3186	-51 - 47	3250	-45 - 49	3314	-45 - 47	3378	-51 - 15	3442	-51 - 17	3506	-45 - 15	3570	-45 - 17
3123	-51 - 51	3187	-51 - 45	3251	-45 - 51	3315	-45 - 45	3379	-51 - 13	3443	-51 - 19	3507	-45 - 13	3571	-45 - 19
3124	-49 - 55	3188	-49 - 41	3252	-47 - 55	3316	-47 - 41	3380	-49 - 9	3444	-49 - 23	3508	-47 - 9	3572	-47 - 23
3125	-49 - 53	3189	-49 - 43	3253	-47 - 53	3317	-47 - 43	3381	-49 - 11	3445	-49 - 21	3509	-47 - 11	3573	-47 - 21
3126	-13 - 73	3190	-51 - 41	3254	-45 - 55	3318	-45 - 41	3382	-51 - 9	3446	-51 - 23	3510	-45 - 9	3574	-45 - 23
3127	-51 - 53	3191	-51 - 43	3255	-45 - 53	3319	-45 - 43	3383	-51 - 11	3447	-51 - 21	3511	-45 - 11	3575	-45 - 21
3128	-55 - 49	3192	-55 - 47	3256	-41 - 49	3320	-41 - 47	3384	-55 - 15	3448	-55 - 17	3512	-41 - 15	3576	-41 - 17
3129	-73 - 13	3193	-55 - 45	3257	-41 - 51	3321	-41 - 45	3385	-55 - 13	3449	-55 - 19	3513	-41 - 13	3577	-41 - 19
3130	-53 - 49	3194	-53 - 47	3258	-43 - 49	3322	-43 - 47	3386	-53 - 15	3450	-53 - 17	3514	-43 - 15	3578	-43 - 17
3131	-53 - 51	3195	-53 - 45	3259	-43 - 51	3323	-43 - 45	3387	-53 - 13	3451	-53 - 19	3515	-43 - 13	3579	-43 - 19
3132	-73 - 9	3196	-55 - 41	3260	-41 - 55	3324	-41 - 41	3388	-55 - 9	3452	-55 - 23	3516	-41 - 9	3580	-41 - 23
3133	-73 - 11	3197	-55 - 43	3261	-41 - 53	3325	-41 - 43	3389	-55 - 11	3453	-55 - 21	3517	-41 - 11	3581	-41 - 21
3134	-11 - 73	3198	-53 - 41	3262	-43 - 55	3326	-43 - 41	3390	-53 - 9	3454	-53 - 23	3518	-43 - 9	3582	-43 - 23
3135	-53 - 53	3199	-53 - 43	3263	-43 - 53	3327	-43 - 43	3391	-53 - 11	3455	-53 - 21	3519	-43 - 11	3583	-43 - 21

Fig. 31

3616	-15 - 63	3680	-15 - 33	3744	-17 - 63	3808	-17 - 33	3872	-15 - 11	3936	-15 - 31	4000	-17 - 11	4064	-17 - 31
3617	-15 - 61	3681	-15 - 35	3745	-17 - 61	3809	-17 - 35	3873	-15 - 31	3937	-15 - 29	4001	-17 - 31	4065	-17 - 29
3618	-13 - 63	3682	-13 - 33	3746	-19 - 63	3810	-19 - 33	3874	-13 - 11	3938	-13 - 31	4002	-19 - 11	4066	-19 - 31
3619	-13 - 61	3683	-13 - 35	3747	-19 - 61	3811	-19 - 35	3875	-13 - 31	3939	-13 - 29	4003	-19 - 31	4067	-19 - 29
3620	-15 - 57	3684	-15 - 39	3748	-17 - 57	3812	-17 - 39	3876	-15 - 71	3940	-15 - 25	4004	-17 - 71	4068	-17 - 25
3621	-15 - 59	3685	-15 - 37	3749	-17 - 59	3813	-17 - 37	3877	-15 - 51	3941	-15 - 27	4005	-17 - 51	4069	-17 - 27
3622	-13 - 57	3686	-13 - 39	3750	-19 - 57	3814	-19 - 39	3878	-13 - 71	3942	-13 - 25	4006	-19 - 71	4070	-19 - 25
3623	-13 - 59	3687	-13 - 37	3751	-19 - 59	3815	-19 - 37	3879	-13 - 51	3943	-13 - 27	4007	-19 - 51	4071	-19 - 27
3624	-9 - 63	3688	-9 - 33	3752	-23 - 63	3816	-23 - 33	3880	-9 - 11	3944	-9 - 31	4008	-23 - 11	4072	-23 - 31
3625	-9 - 61	3689	-9 - 35	3753	-23 - 61	3817	-23 - 35	3881	-9 - 31	3945	-9 - 29	4009	-23 - 31	4073	-23 - 29
3626	-11 - 63	3690	-11 - 33	3754	-21 - 63	3818	-21 - 33	3882	-11 - 11	3946	-11 - 31	4010	-21 - 11	4074	-21 - 31
3627	-11 - 61	3691	-11 - 35	3755	-21 - 61	3819	-21 - 35	3883	-11 - 31	3947	-11 - 29	4011	-21 - 31	4075	-21 - 29
3628	-9 - 57	3692	-9 - 39	3756	-23 - 57	3820	-23 - 39	3884	-9 - 71	3948	-9 - 25	4012	-23 - 71	4076	-23 - 25
3629	-9 - 59	3693	-9 - 37	3757	-23 - 59	3821	-23 - 37	3885	-9 - 51	3949	-9 - 27	4013	-23 - 51	4077	-23 - 27
3630	-11 - 57	3694	-11 - 39	3758	-21 - 57	3822	-21 - 39	3886	-11 - 71	3950	-11 - 25	4014	-21 - 71	4078	-21 - 25
3631	-11 - 59	3695	-11 - 37	3759	-21 - 59	3823	-21 - 37	3887	-11 - 51	3951	-11 - 27	4015	-21 - 51	4079	-21 - 27
3632	-15 - 49	3696	-15 - 47	3760	-17 - 49	3824	-17 - 47	3888	-15 - 15	3952	-15 - 17	4016	-17 - 15	4080	-17 - 17
3633	-15 - 51	3697	-15 - 45	3761	-17 - 51	3825	-17 - 45	3889	-15 - 13	3953	-15 - 19	4017	-17 - 13	4081	-17 - 19
3634	-13 - 49	3698	-13 - 47	3762	-19 - 49	3826	-19 - 47	3890	-13 - 15	3954	-13 - 17	4018	-19 - 15	4082	-19 - 17
3635	-13 - 51	3699	-13 - 45	3763	-19 - 51	3827	-19 - 45	3891	-13 - 13	3955	-13 - 19	4019	-19 - 13	4083	-19 - 19
3636	-15 - 55	3700	-15 - 41	3764	-17 - 55	3828	-17 - 41	3892	-15 - 9	3956	-15 - 23	4020	-17 - 9	4084	-17 - 23
3637	-15 - 53	3701	-15 - 43	3765	-17 - 53	3829	-17 - 43	3893	-15 - 11	3957	-15 - 21	4021	-17 - 11	4085	-17 - 21
3638	-13 - 55	3702	-13 - 41	3766	-19 - 55	3830	-19 - 41	3894	-13 - 9	3958	-13 - 23	4022	-19 - 9	4086	-19 - 23
3639	-13 - 53	3703	-13 - 43	3767	-19 - 53	3831	-19 - 43	3895	-13 - 11	3959	-13 - 21	4023	-19 - 11	4087	-19 - 21
3640	-9 - 49	3704	-9 - 47	3768	-23 - 49	3832	-23 - 47	3896	-9 - 15	3960	-9 - 17	4024	-23 - 15	4088	-23 - 17
3641	-9 - 51	3705	-9 - 45	3769	-23 - 51	3833	-23 - 45	3897	-9 - 13	3961	-9 - 19	4025	-23 - 13	4089	-23 - 19
3642	-11 - 49	3706	-11 - 47	3770	-21 - 49	3834	-21 - 47	3898	-11 - 15	3962	-11 - 17	4026	-21 - 15	4090	-21 - 17
3643	-11 - 51	3707	-11 - 45	3771	-21 - 51	3835	-21 - 45	3899	-11 - 13	3963	-11 - 19	4027	-21 - 13	4091	-21 - 19
3644	-9 - 55	3708	-9 - 41	3772	-23 - 55	3836	-23 - 41	3900	-9 - 9	3964	-9 - 23	4028	-23 - 9	4092	-23 - 23
3645	-9 - 53	3709	-9 - 43	3773	-23 - 53	3837	-23 - 43	3901	-9 - 11	3965	-9 - 21	4029	-23 - 11	4093	-23 - 21
3646	-11 - 55	3710	-11 - 41	3774	-21 - 55	3838	-21 - 41	3902	-11 - 9	3966	-11 - 23	4030	-21 - 9	4094	-21 - 23
3647	-11 - 53	3711	-11 - 43	3775	-21 - 53	3839	-21 - 43	3903	-11 - 11	3967	-11 - 21	4031	-21 - 11	4095	-21 - 21

Fig. 32

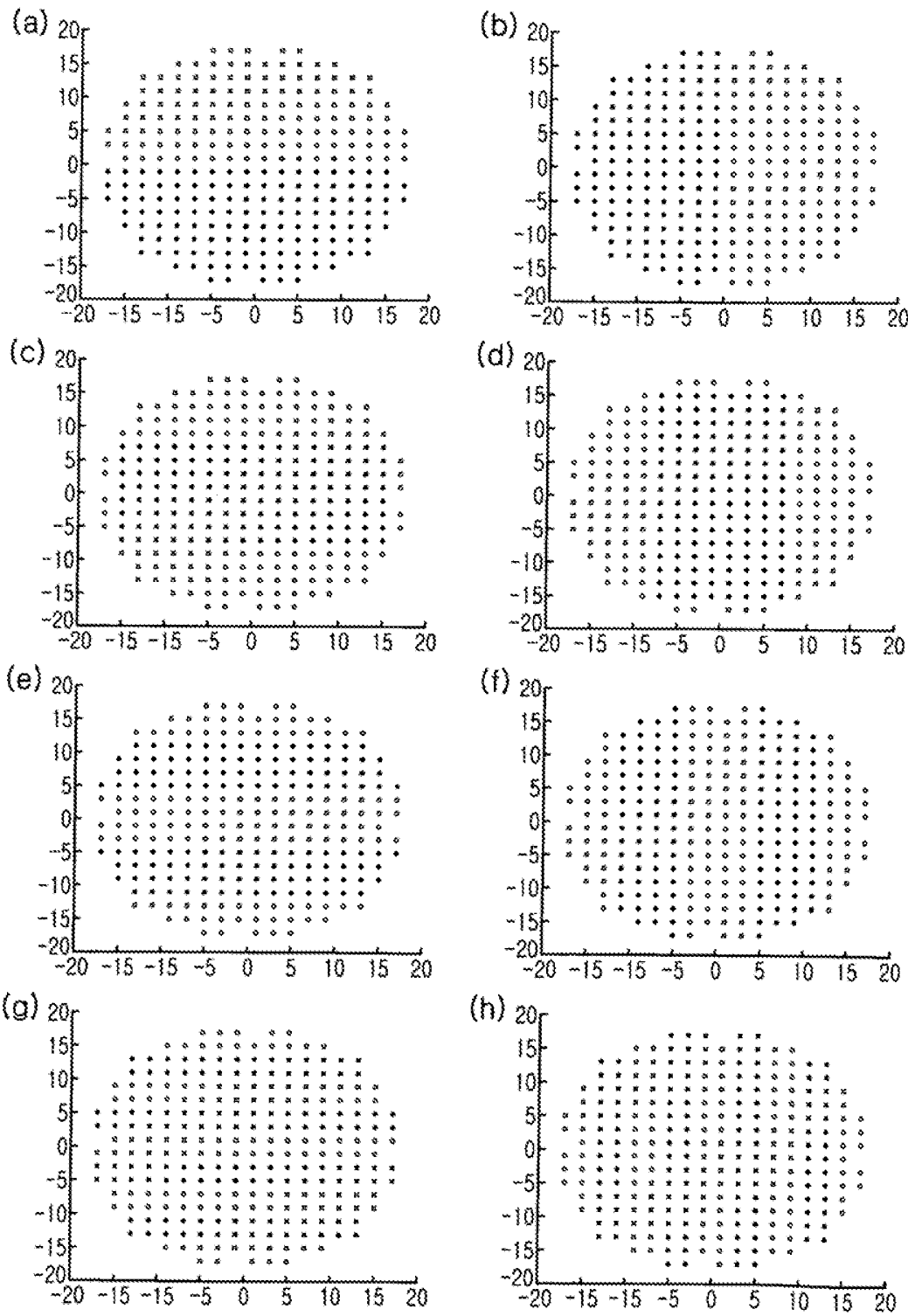


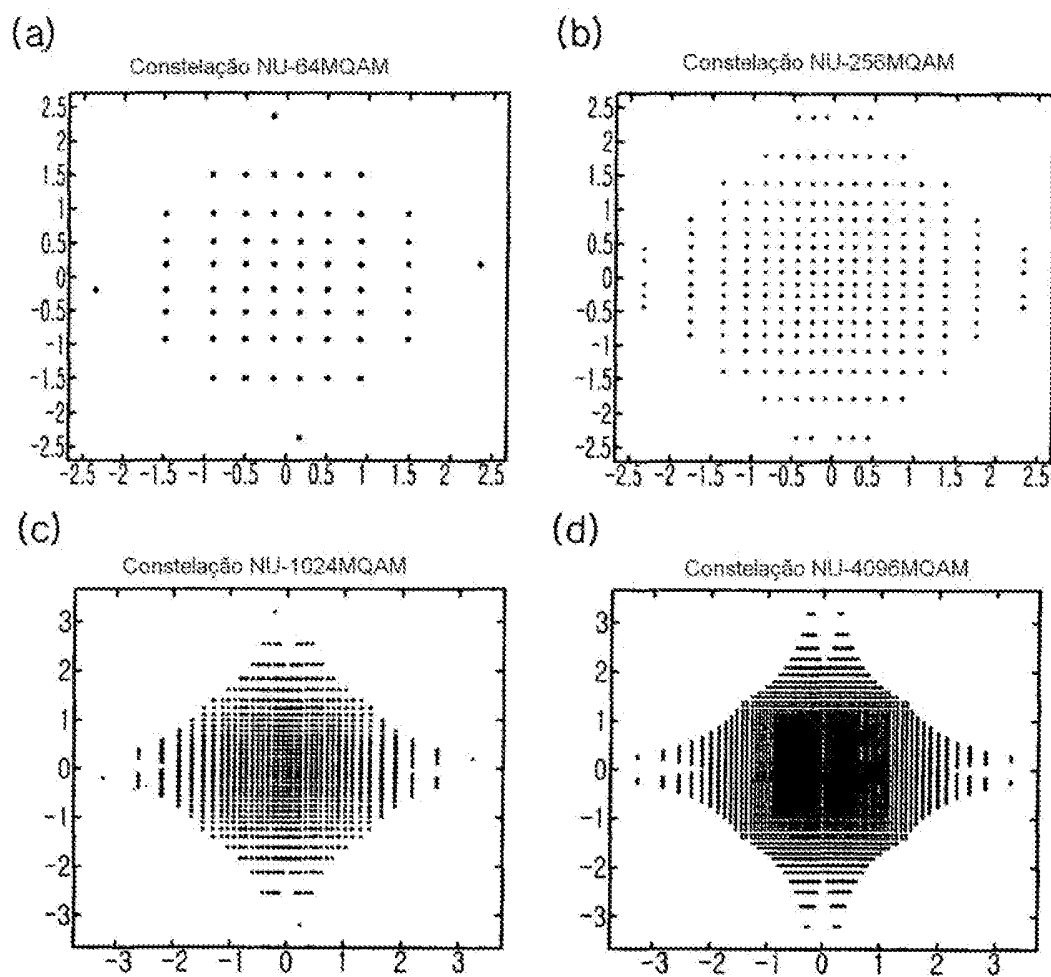
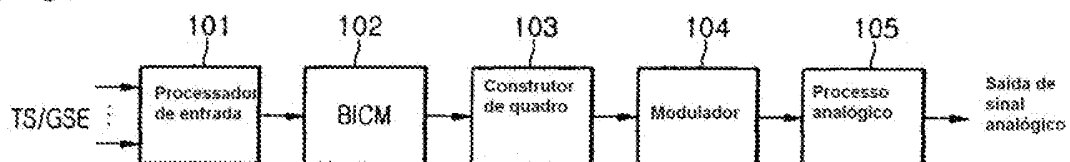
Fig. 33**Fig. 34**

Fig. 35

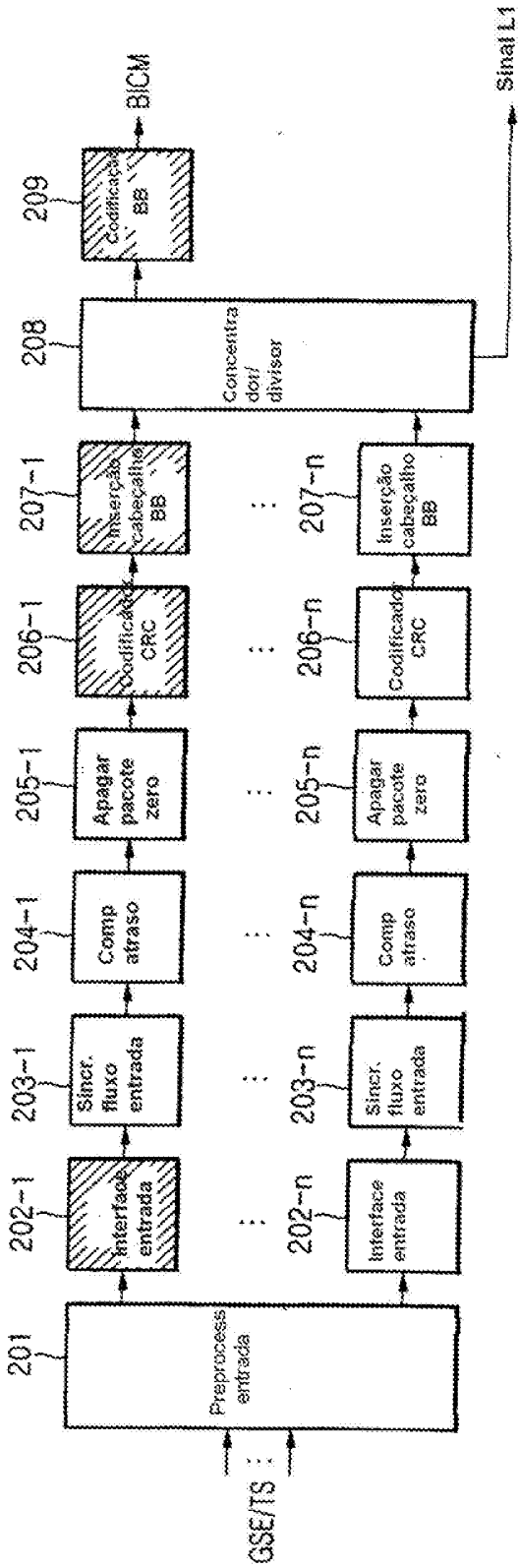


Fig. 36

TS/GS (2 bits)	SIS/MIS (1 bit)	CCM/ACM (1 bit)	ISSYI (1 bit)	NPD (1bit)	EXT (2bits)
00 = GFPS 11 = TS 01 = GCS 10 = GSE	1 = individual 0 = múltiplo	1 = CCM 0 = ACM	1 = activo 0 = não-activo	1 = activo 0 = não-activo	Reservado para utilização futura

Campo	Dimensão (Bytes)	Descrição
MATYPE	2	Tal como descrito acima
UPL	2	Comprimento do pacote do utilizador em bits, na gama [0,65535]
DFL	2	Comprimento do campo de dados, na gama [0,53760]
SYNC	1	Uma cópia do byte de sincronização do pacote de utilizador
SYNCD	2	A distância em bits do início do campo de dados (DATA FIELD) para o primeiro UP completo do campo de dados. SYNCD=0 quer dizer que o primeiro UP se encontra alinhado com o início do campo de dados. SYNCD=65535 quer dizer que nenhum UP tem início no DATA FIELD.
CRC-8 MODE	1	O XOR do campo CRC-8 (1-byte) com o campo MODE (1-byte). CRC-8 é o código de detecção de erro aplicado aos primeiros 9 bytes do BBHEADER. MODE (8 bits) deve ser: * 0: Modo normal * 1: Modo de alta eficiência * Outros valores: reservado para utilização futura

Fig. 37

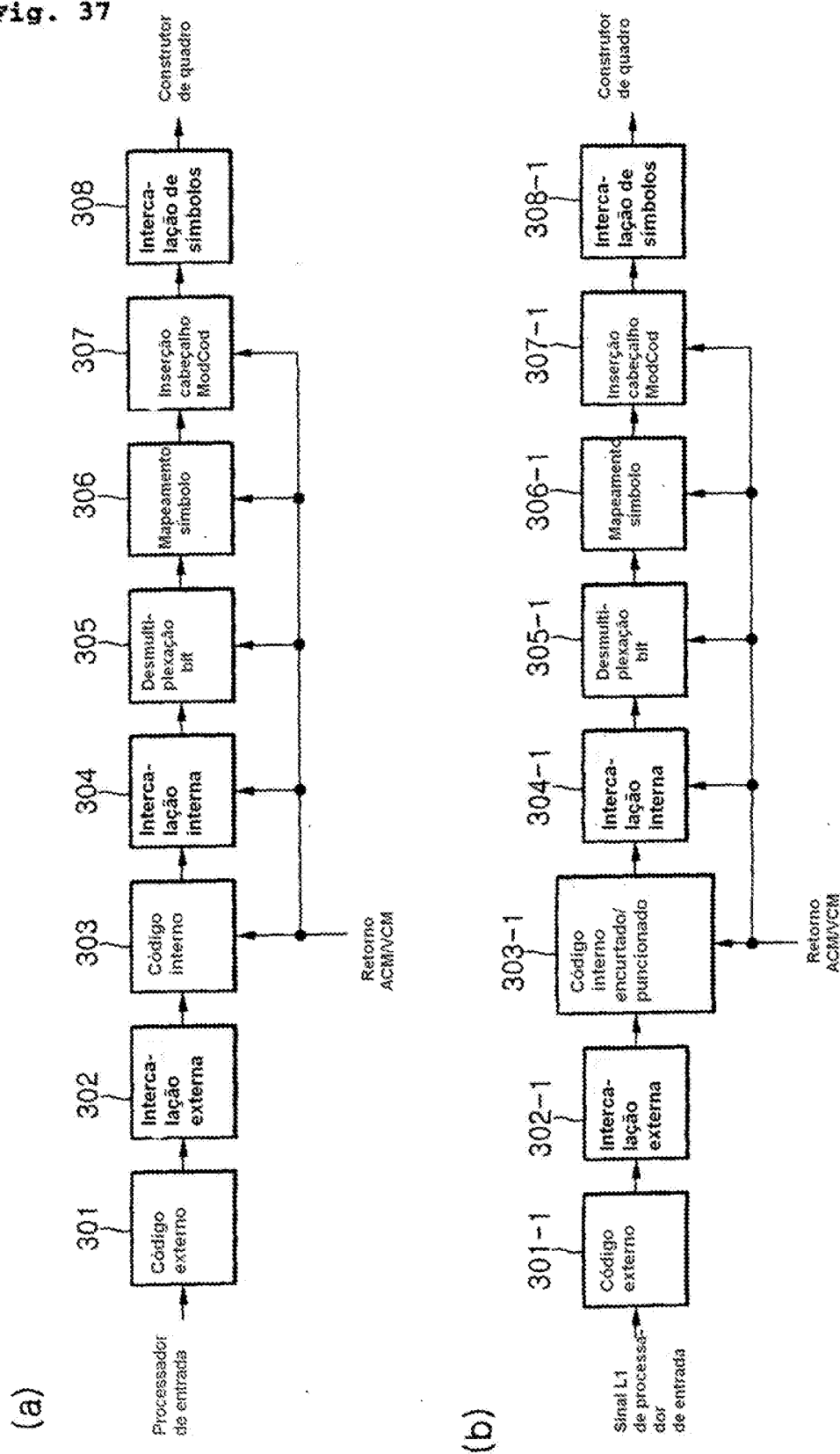


Fig. 38

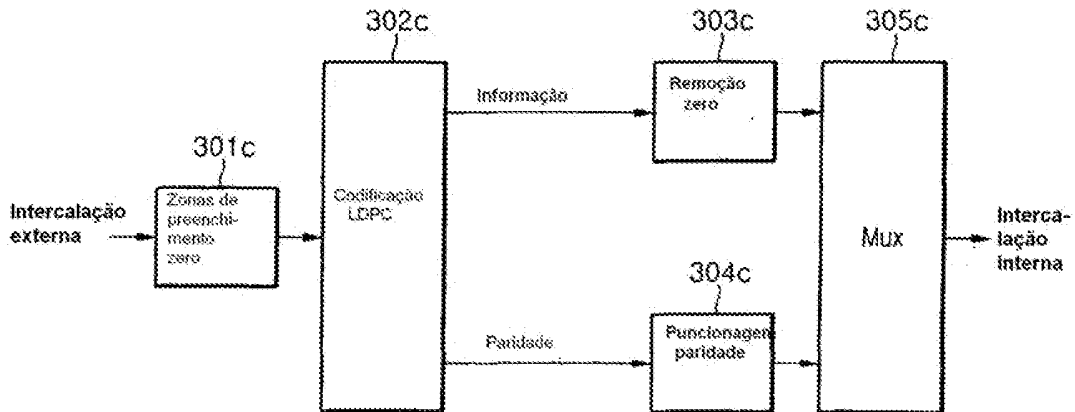


Fig. 39

			Caso 1	Caso 2
Capacidade (bit/s/Hz)			Modulação	Modulação
6	1/2	3.0	NU-MQAM	NU-QAM
	2/3	4.0	NU-MQAM	NU-QAM
	3/4	4.5	NU-MQAM	NU-MQAM
	4/5	4.8	MQAM	MQAM
	5/6	5.0	MQAM	MQAM
	8/9	5.3	MQAM	MQAM
	9/10	5.4	MQAM	MQAM
8	1/2	4.0	NU-MQAM	NU-QAM
	2/3	5.3	NU-MQAM	NU-QAM
	3/4	6.0	NU-MQAM	NU-MQAM
	4/5	6.4	NU-MQAM	NU-MQAM
	5/6	6.7	MQAM	MQAM
	8/9	7.1	MQAM	MQAM
	9/10	7.2	MQAM	MQAM
10	1/2	5.0	NU-MQAM	NU-QAM
	2/3	6.7	NU-MQAM	NU-QAM
	3/4	7.5	NU-MQAM	NU-MQAM
	4/5	8.0	NU-MQAM	NU-MQAM
	5/6	8.3	NU-MQAM	NU-MQAM
	8/9	8.9	MQAM	MQAM
	9/10	9.0	MQAM	MQAM
12	1/2	6.0	NU-MQAM	NU-QAM
	2/3	8.0	NU-MQAM	NU-QAM
	3/4	9.0	NU-MQAM	NU-MQAM
	4/5	9.6	NU-MQAM	NU-MQAM
	5/6	10.0	NU-MQAM	NU-MQAM
	8/9	10.7	MQAM	MQAM
	9/10	10.8	MQAM	MQAM

Fig. 40

			Caso 1	Caso 2	Caso 2
Capacidade (bit/s/Hz)			Modulação	Modulação	Modulação
2	1/2	3.0	QAM	QAM	QAM
	2/3	4.0	QAM	QAM	QAM
	3/4	4.5	QAM	QAM	QAM
	4/5	4.8	QAM	QAM	QAM
	5/6	5.0	QAM	QAM	QAM
	8/9	5.3	QAM	QAM	QAM
	9/10	5.4	QAM	QAM	QAM
4	1/2	3.0	QAM	QAM	QAM
	2/3	4.0	QAM	QAM	QAM
	3/4	4.5	QAM	QAM	QAM
	4/5	4.8	QAM	QAM	QAM
	5/6	5.0	QAM	QAM	QAM
	8/9	5.3	QAM	QAM	QAM
	9/10	5.4	QAM	QAM	QAM
6	1/2	3.0	QAM	QAM	QAM
	2/3	4.0	QAM	QAM	QAM
	3/4	4.5	QAM	QAM	QAM
	4/5	4.8	QAM	QAM	QAM
	5/6	5.0	QAM	QAM	QAM
	8/9	5.3	QAM	QAM	QAM
	9/10	5.4	QAM	QAM	QAM
8	1/2	4.0	QAM	QAM	QAM
	2/3	5.3	QAM	QAM	QAM
	3/4	6.0	QAM	QAM	QAM
	4/5	6.4	QAM	QAM	QAM
	5/6	6.7	QAM	QAM	QAM
	8/9	7.1	QAM	QAM	QAM
	9/10	7.2	QAM	QAM	QAM
10	1/2	5.0	NU-MQAM	NU-QAM	MQAM
	2/3	6.7	NU-MQAM	NU-QAM	MQAM
	3/4	7.5	NU-MQAM	NU-MQAM	MQAM
	4/5	8.0	NU-MQAM	NU-MQAM	MQAM
	5/6	8.3	NU-MQAM	NU-MQAM	MQAM
	8/9	8.9	MQAM	MQAM	MQAM
	9/10	9.0	MQAM	MQAM	MQAM
12	1/2	6.0	NU-MQAM	NU-QAM	MQAM
	2/3	8.0	NU-MQAM	NU-QAM	MQAM
	3/4	9.0	NU-MQAM	NU-MQAM	MQAM
	4/5	9.6	NU-MQAM	NU-MQAM	MQAM
	5/6	10.0	NU-MQAM	NU-MQAM	MQAM
	8/9	10.7	MQAM	MQAM	MQAM
	9/10	10.8	MQAM	MQAM	MQAM

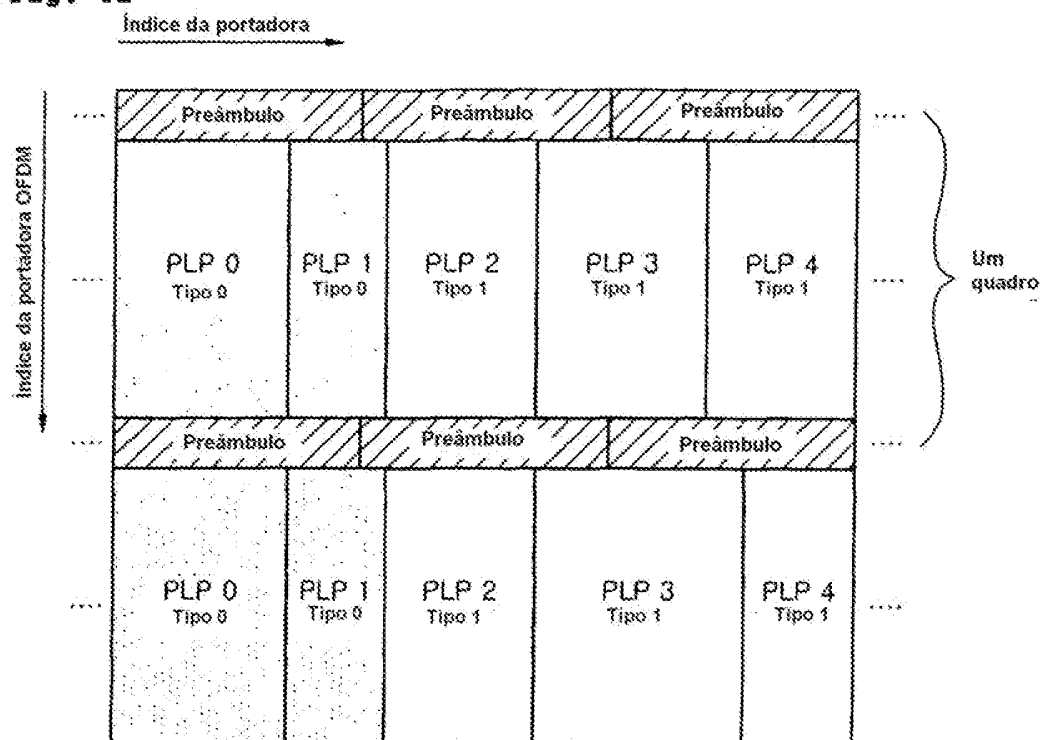
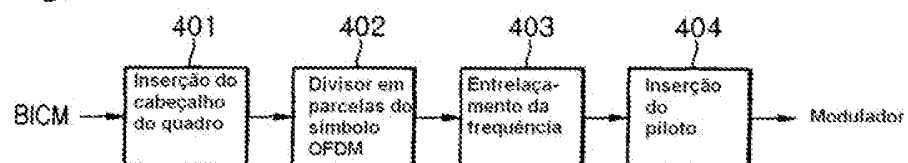
Fig. 41**Fig. 42**

Fig. 43

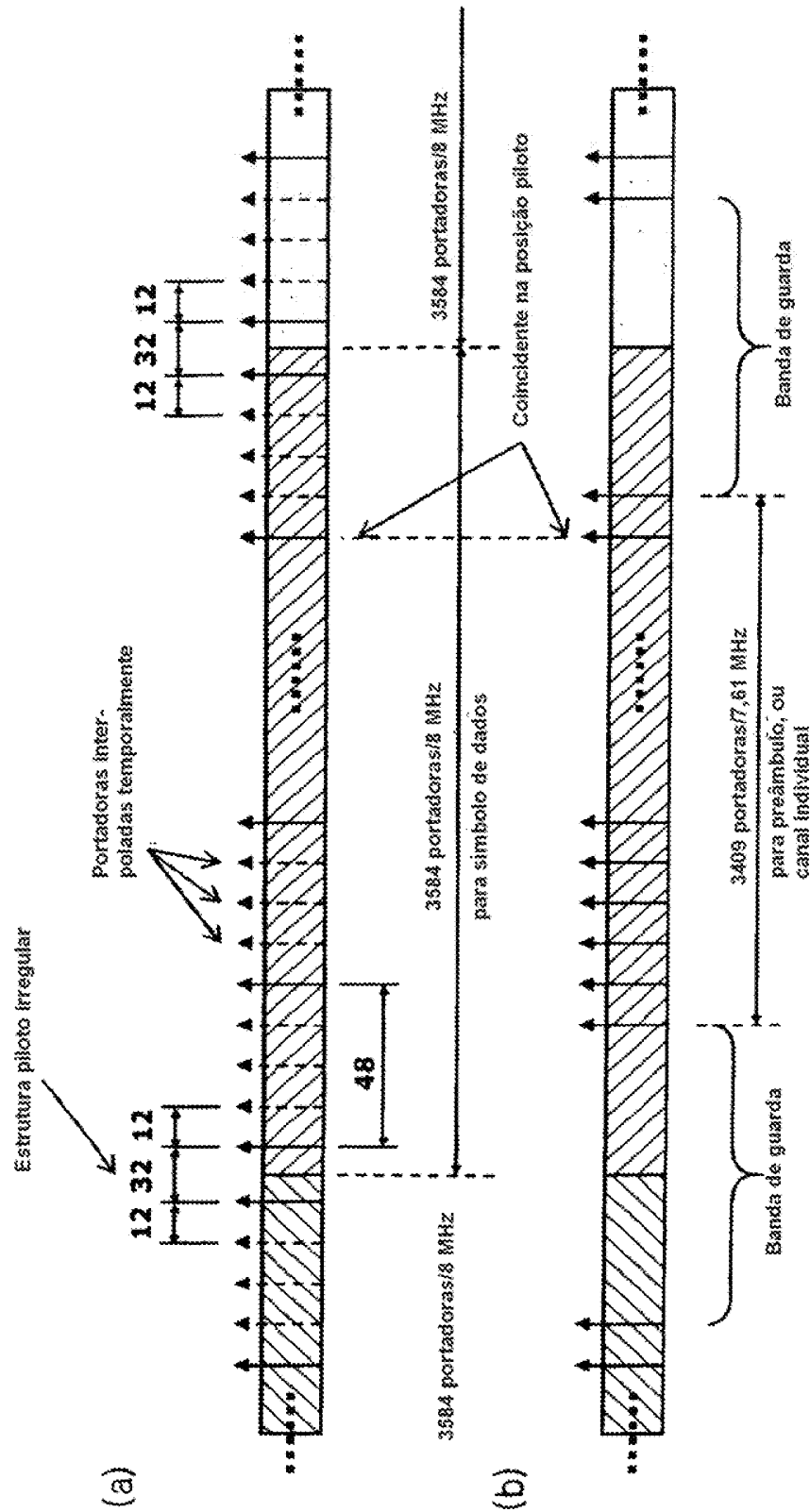


Fig. 44

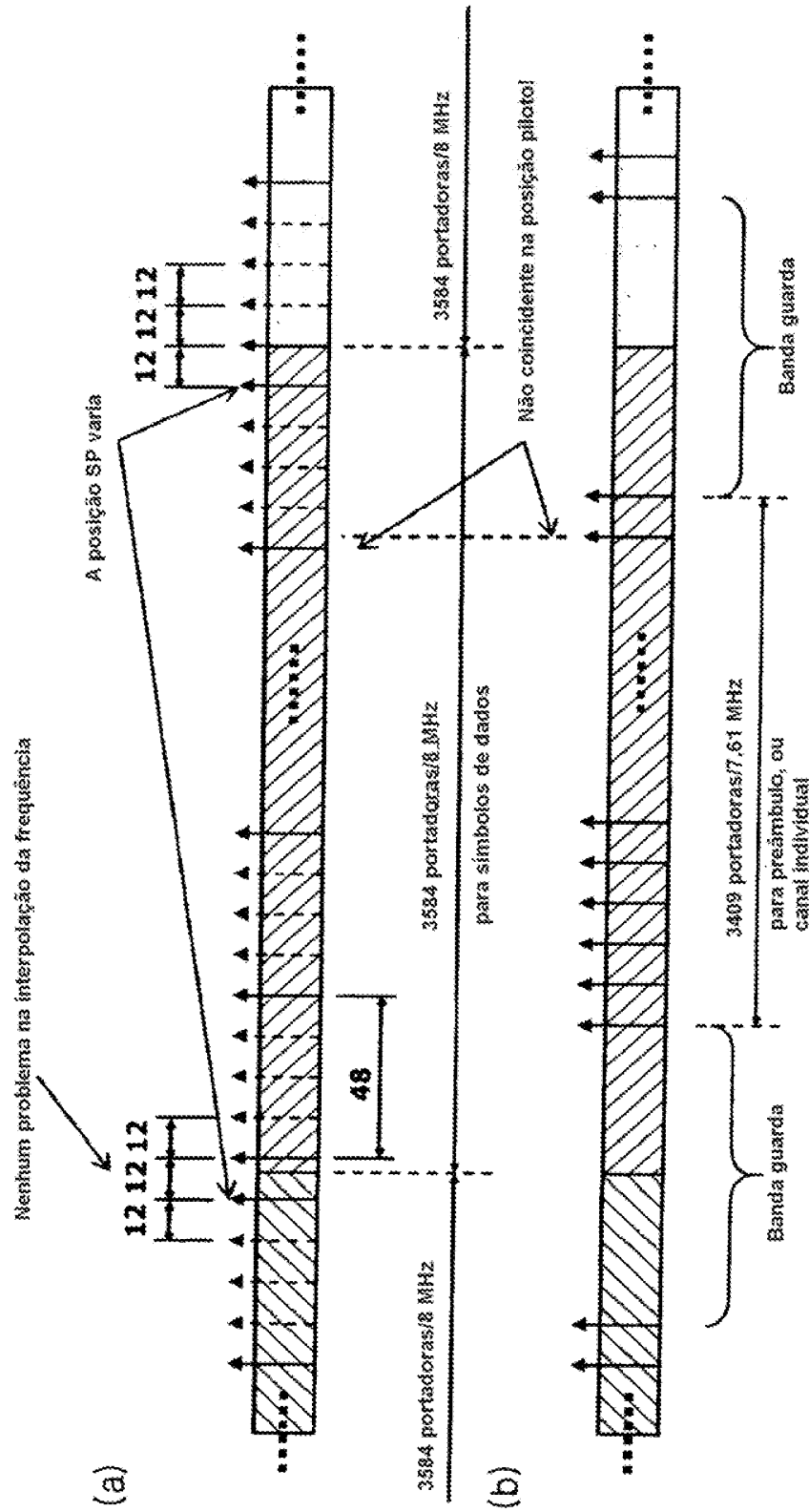


Fig. 45

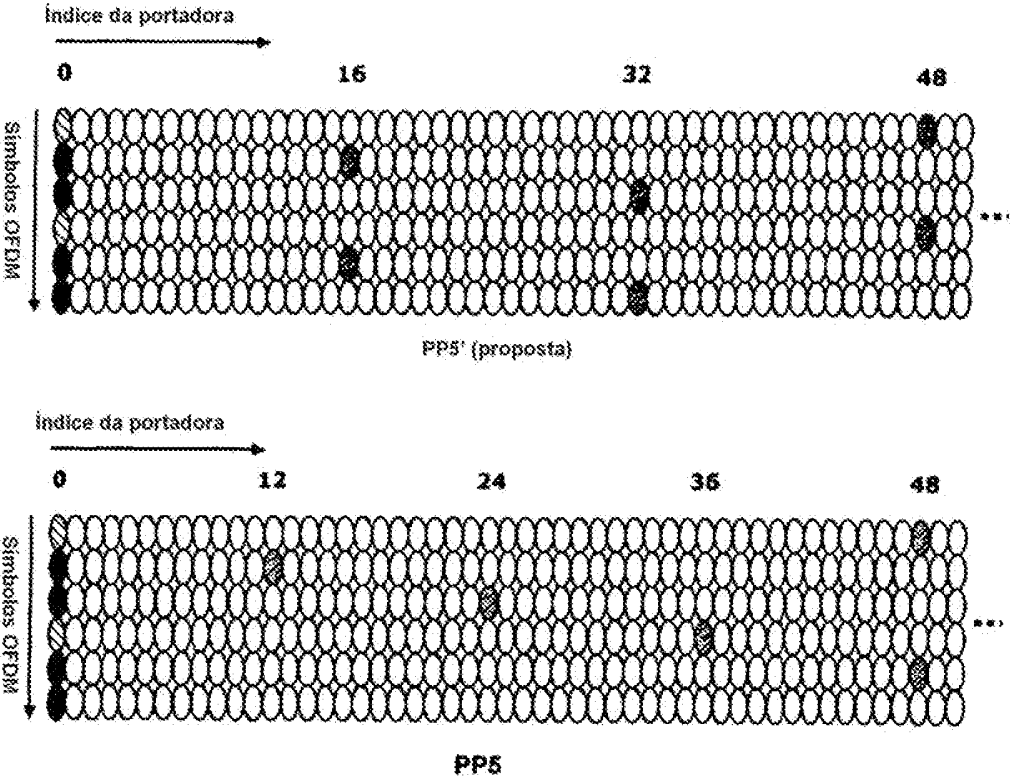


Fig. 46

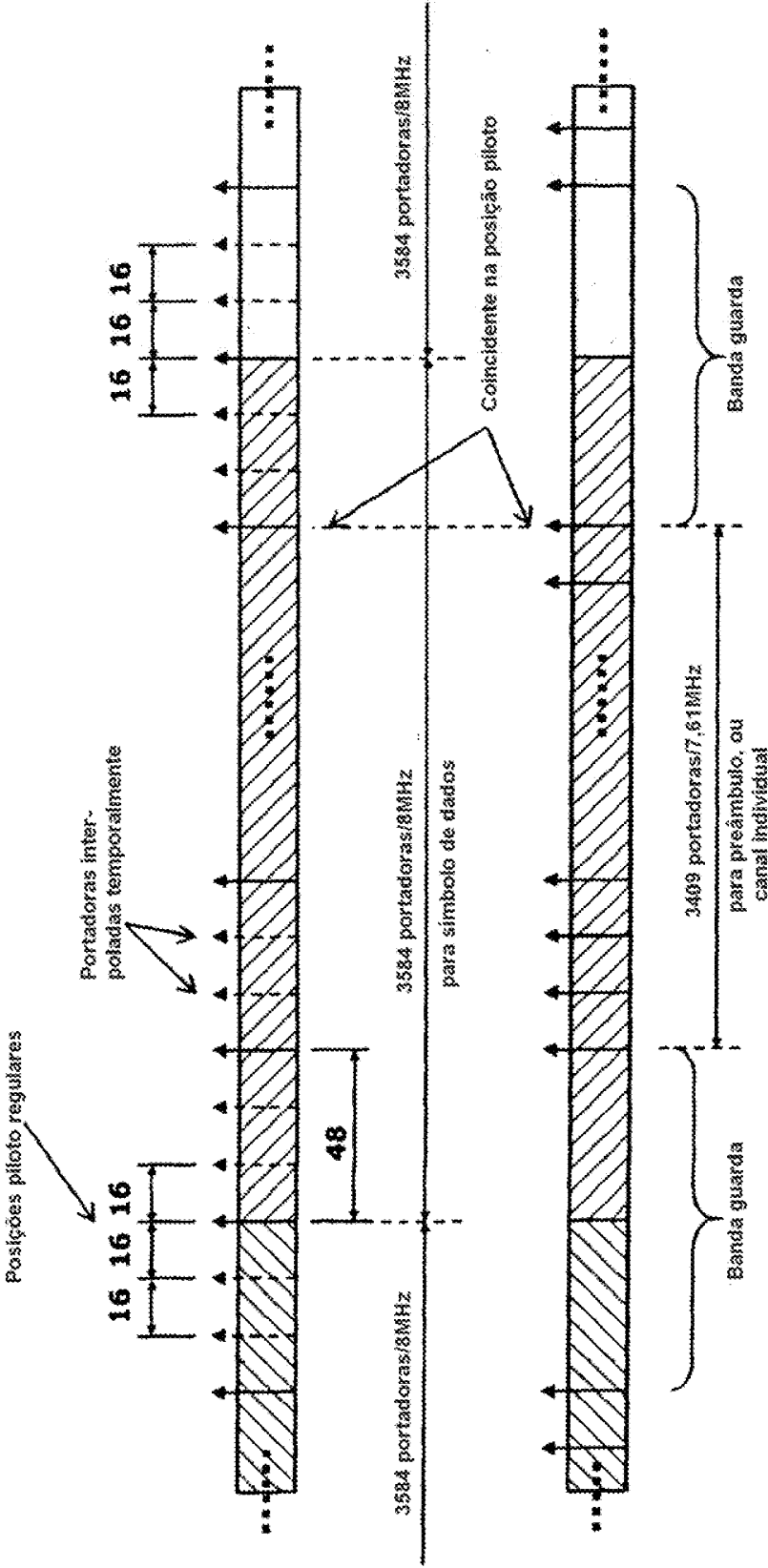


Fig. 47

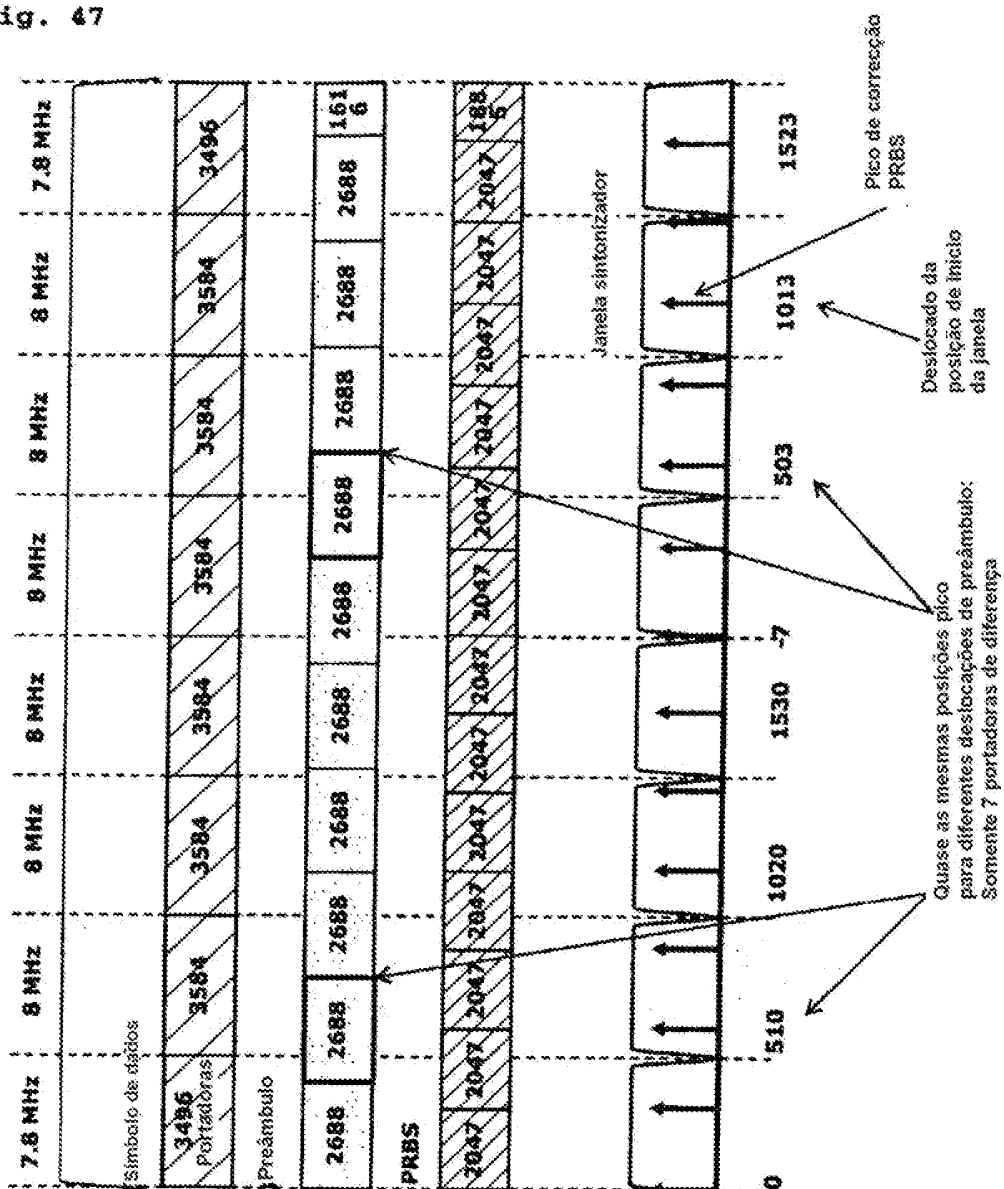


Fig. 48

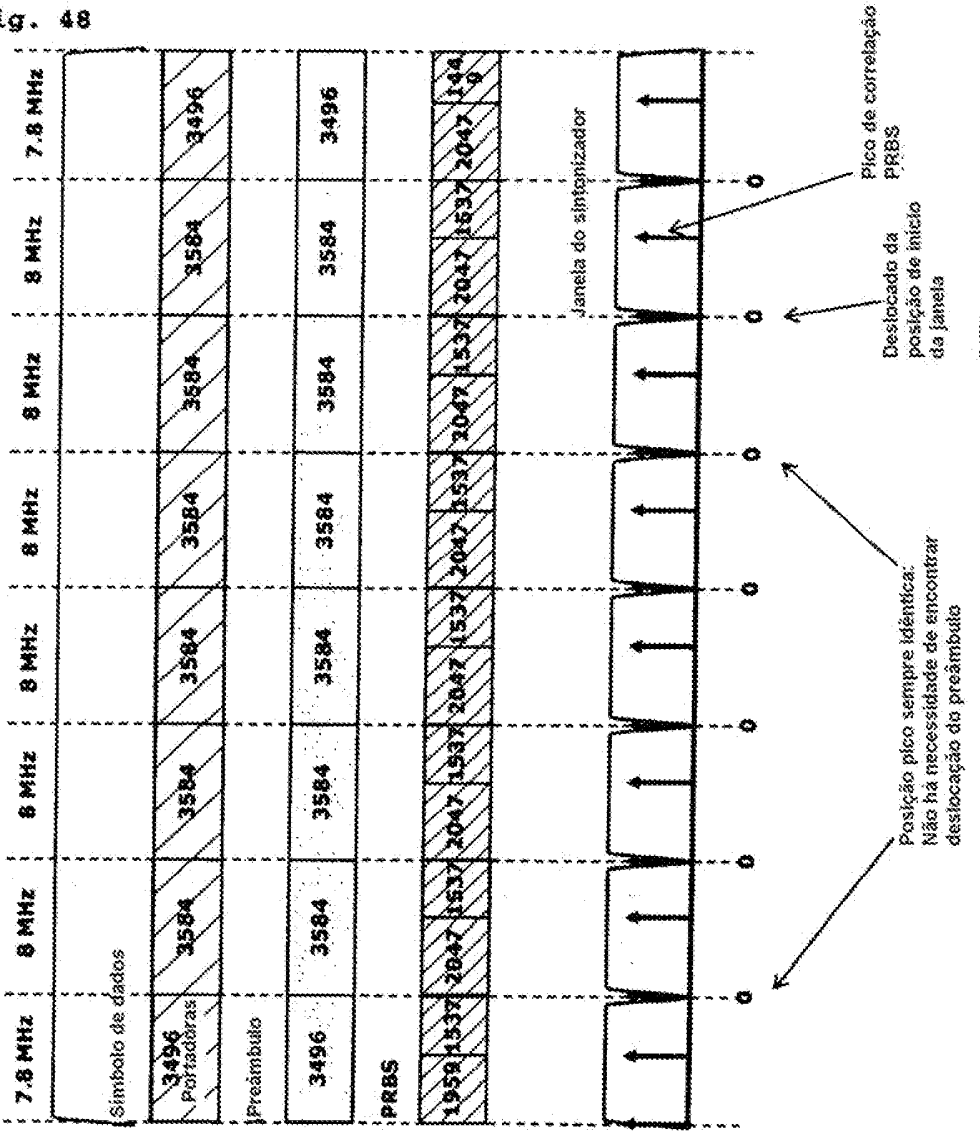


Fig. 49

Atraso (µs)	0.2	0.4	0.8	1.2	2.5	15
Atenuação (dB)	-11	-14	-17	-23	-32	-40

Trajecto de atraso negligenciável

Fig. 50

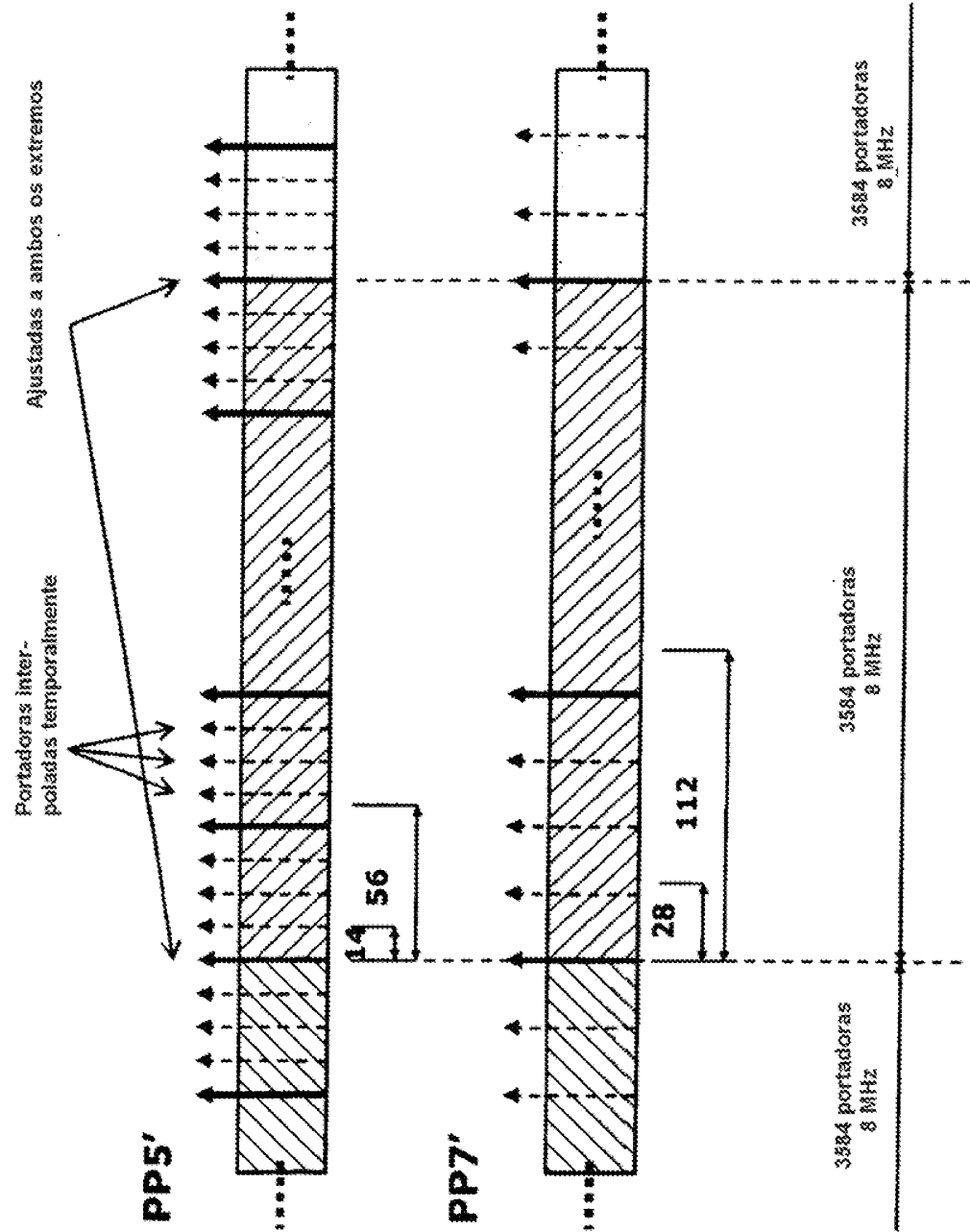


Fig. 51

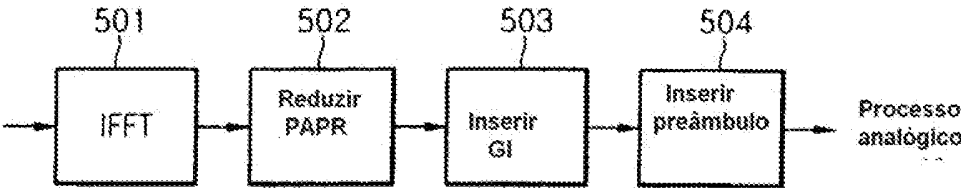


Fig. 52

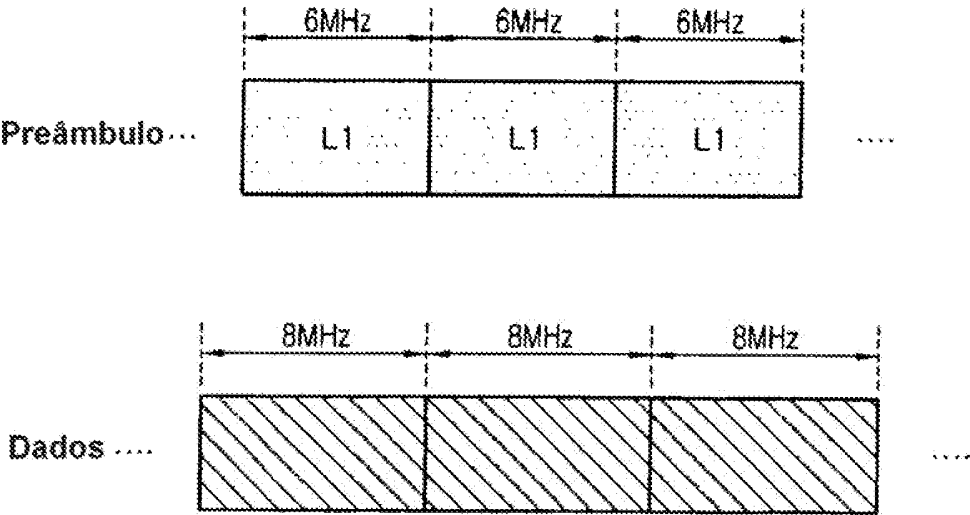


Fig. 53

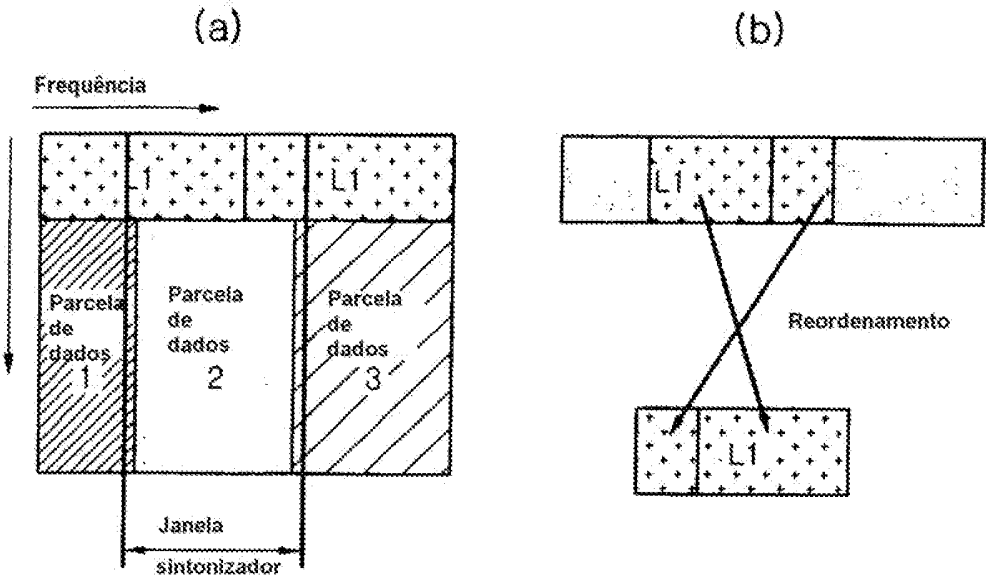


Fig. 54

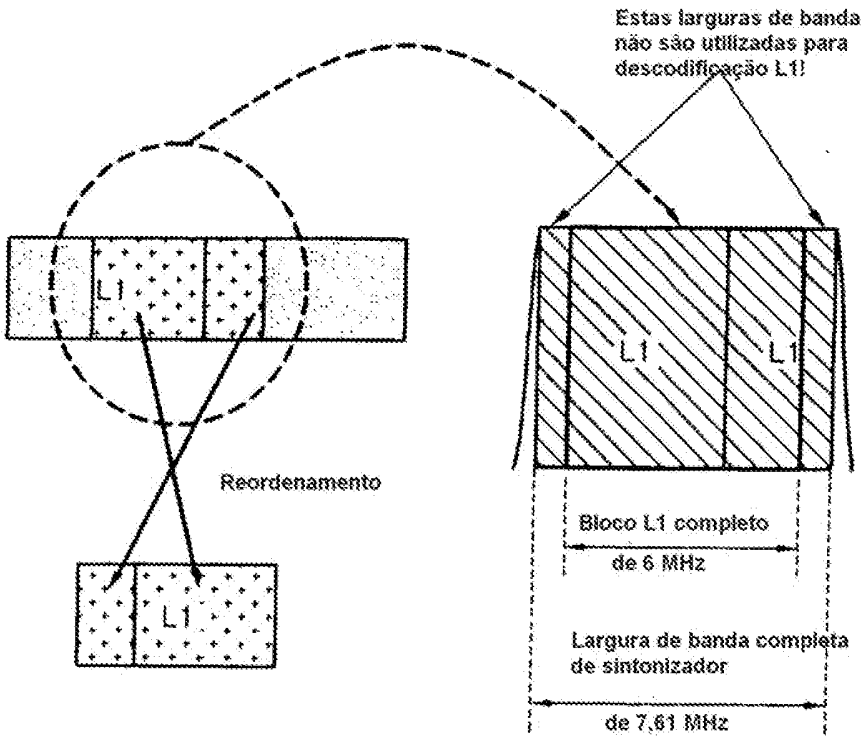


Fig. 55

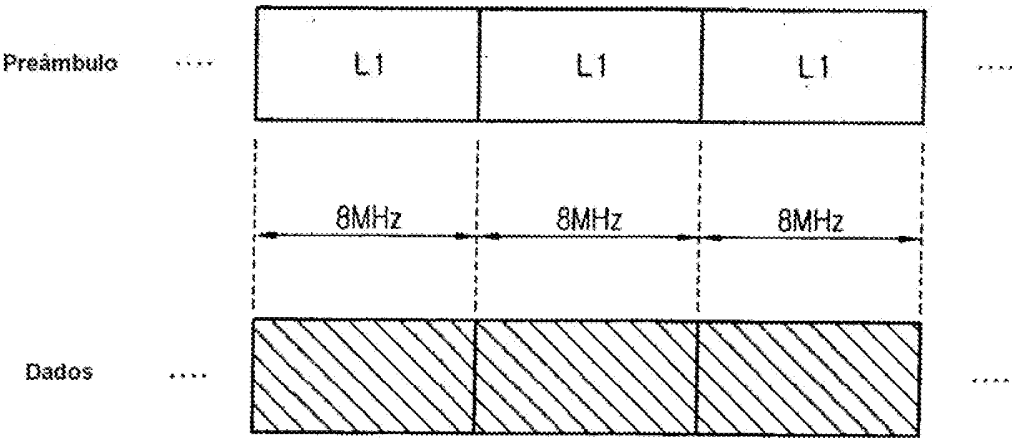


Fig. 56

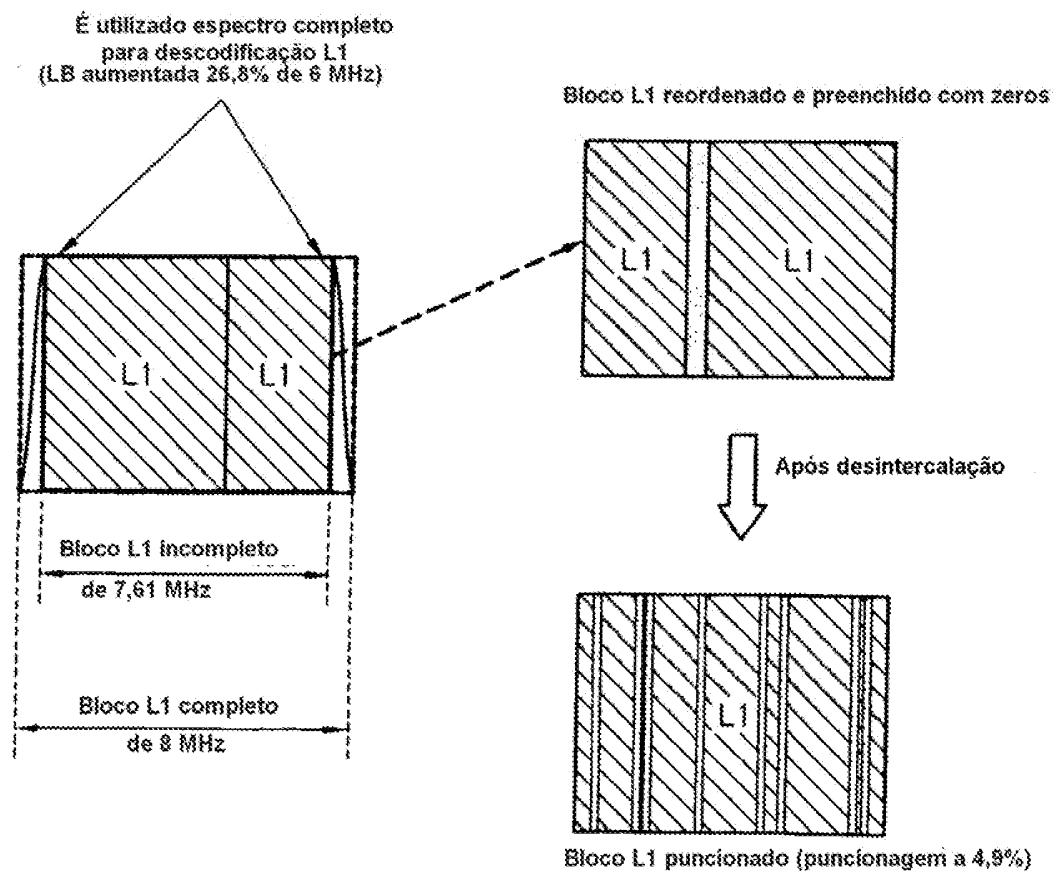


Fig. 57

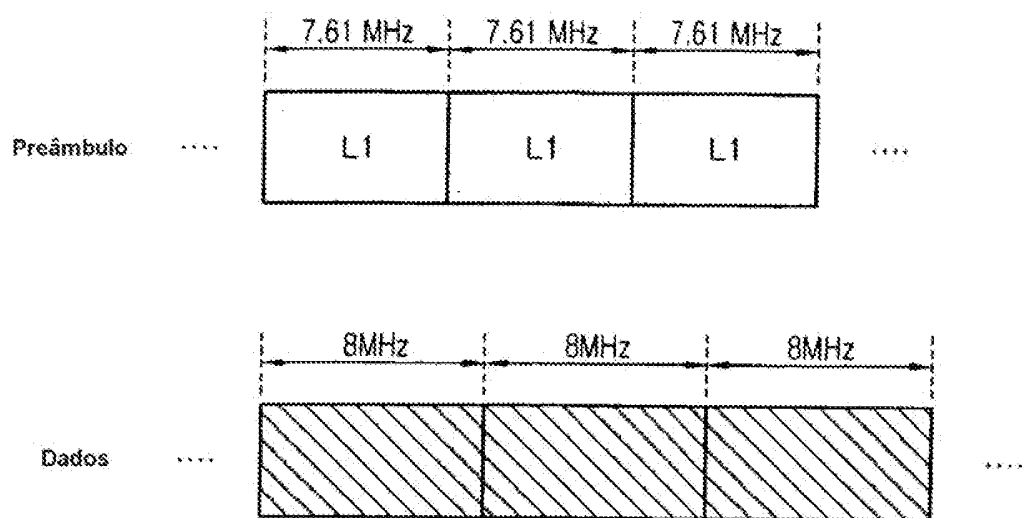


Fig. 58

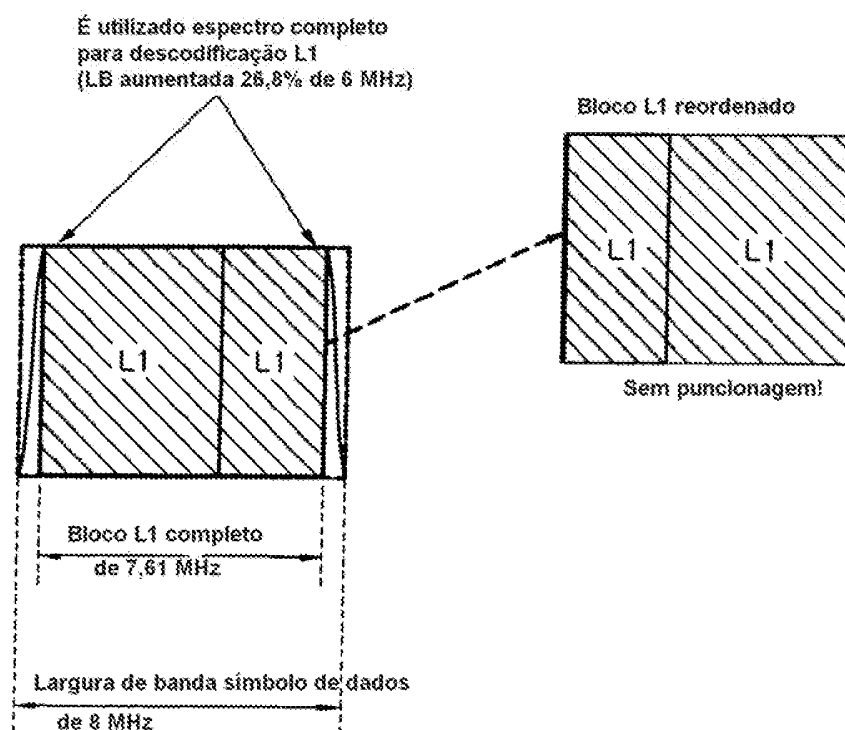


Fig. 59

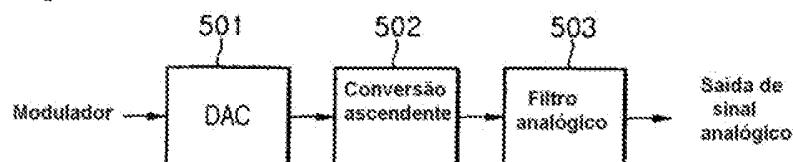


Fig. 60



Fig. 61

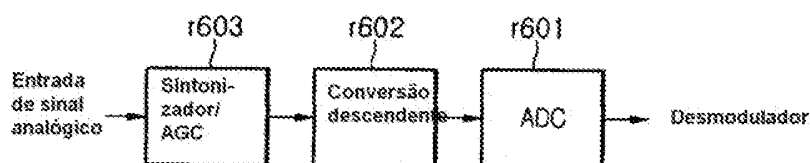


Fig. 62

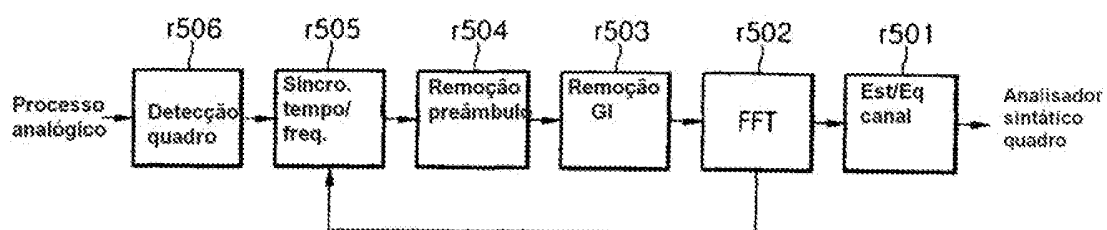


Fig. 63

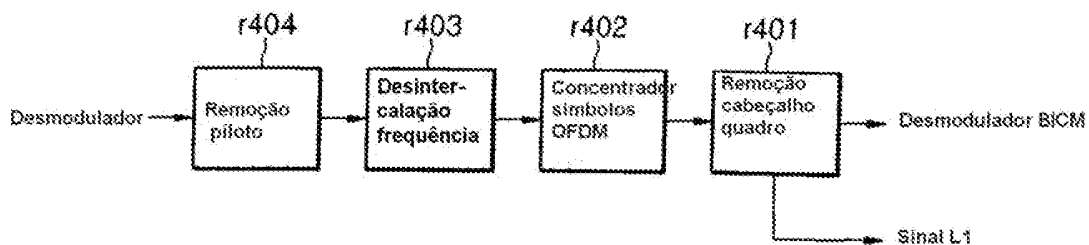


Fig. 64

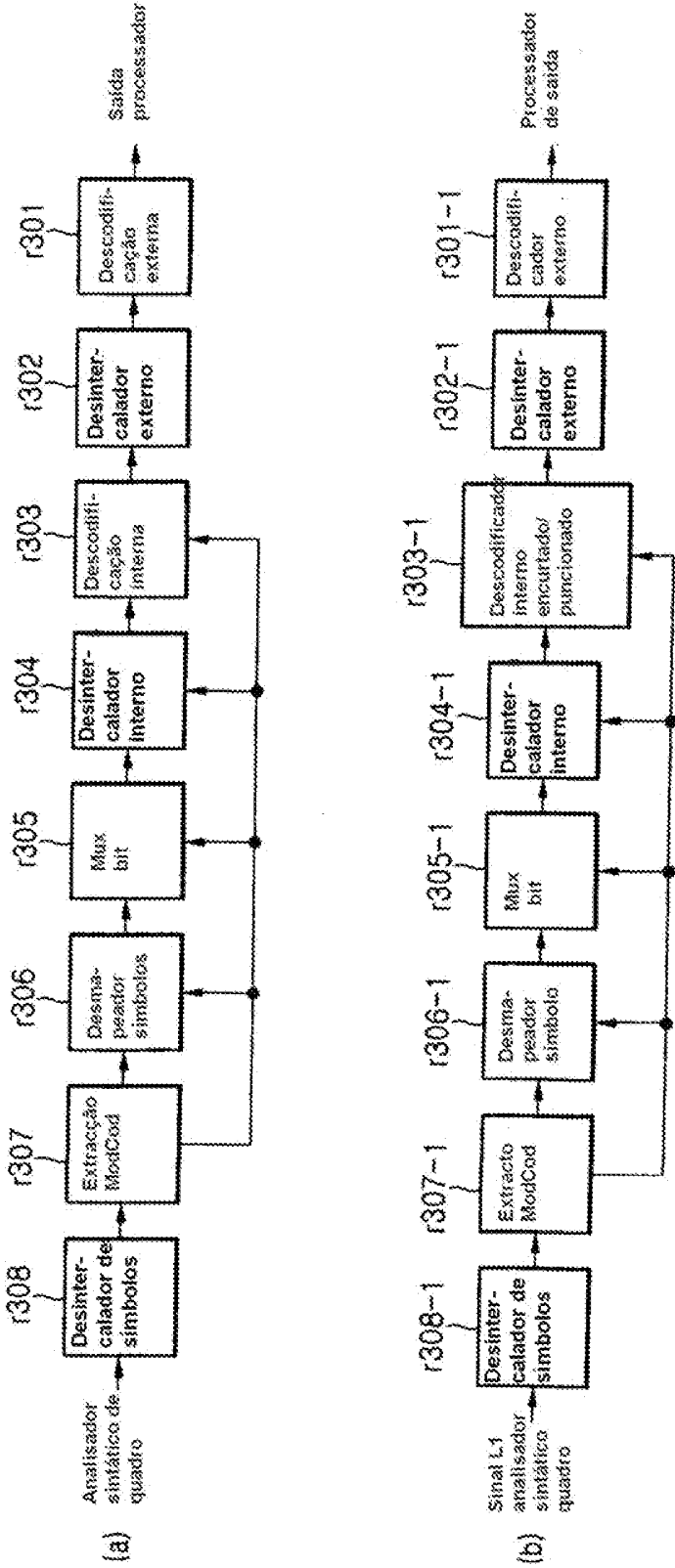


Fig. 65

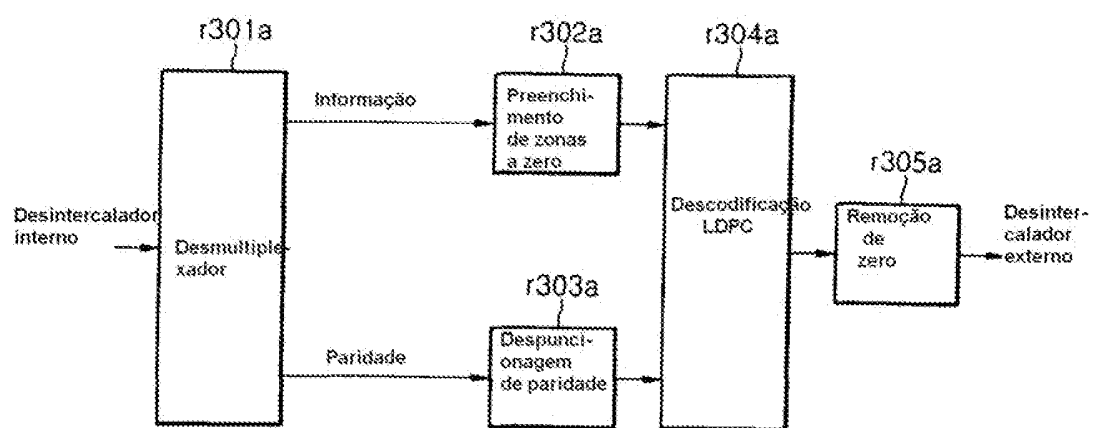


Fig. 66

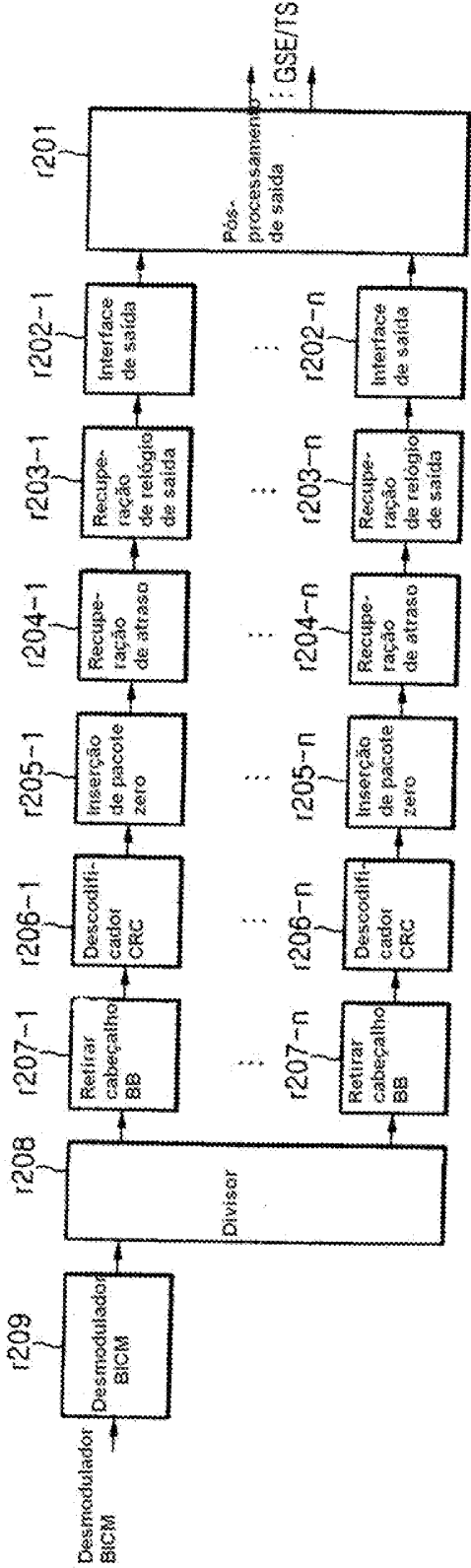


Fig. 67

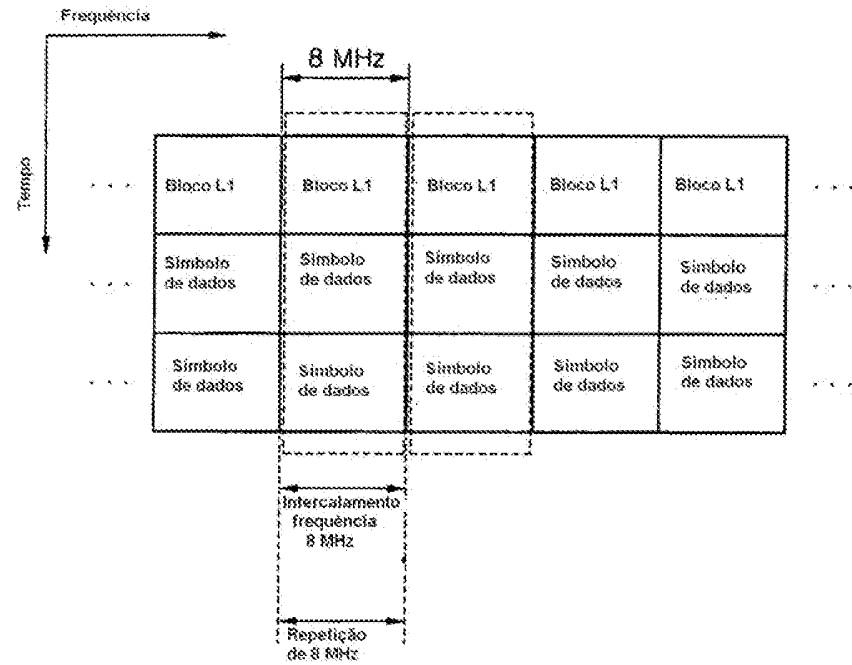


Fig. 68

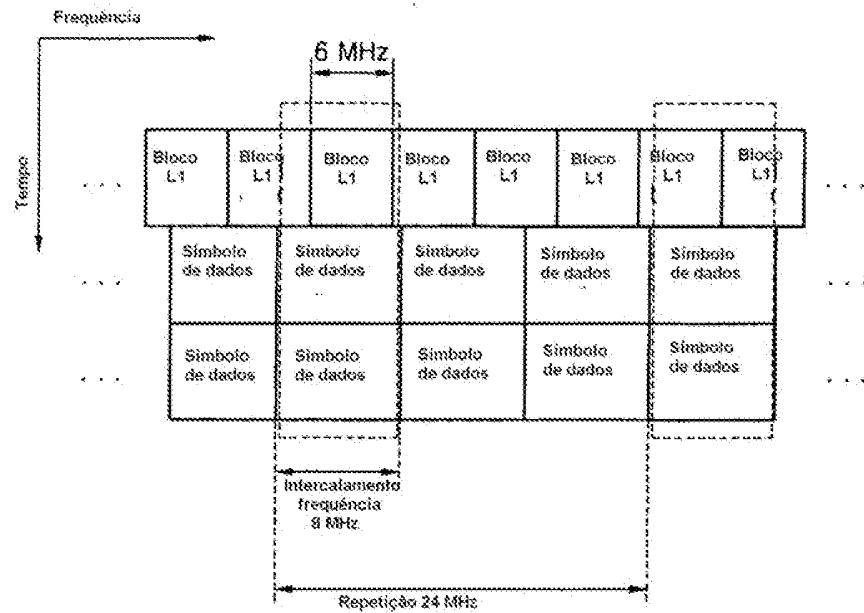


Fig. 69

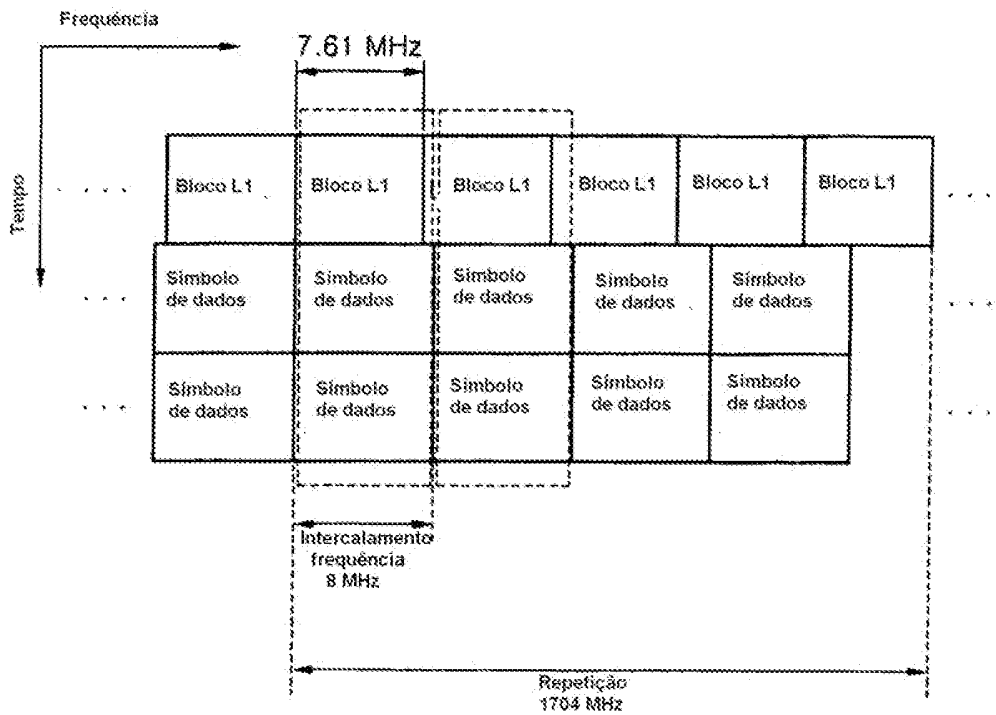


Fig. 70

Campo	Bits	
L1_span	12	Número de portadoras fracionadas por bloco L1 dentro de um símbolo OFDM (Max = 7,61 MHz)
num_chbon	3	número de canais ligados
num_dslice	8	número de parcelas de dados
num_plp	8	número de PLPs
num_notch	5	número de bandas de entalhe
for dslice {		
chbon_index	3	índice de canal ligado
dslice_start	9	início de parcela de dados dentro de um canal (8 MHz)
dslice_width	9	largura da parcela de dados
}		
for plp {		
dslice_id	8	ID parcelado de dados
plp_id	8	PLP ID
plp_type	1	tipo de PLP (comum/dados)
plp_payload_type	5	tipo de carga útil PLP (TS, GS,...)
}		
for notch {		
chbon_index	3	índice de canal ligado
notch_start	9	início da banda de entalhe dentro de um canal (8 MHz)
notch_width	9	largura da banda de entalhe
}		
gi	1	modo de intervalo de guarda
sframe_id	16	ID do superquadro
frame_id	16	ID do quadro
reserved	0	rfu
crc32	32	CRC32
Total	11760	

O número de bits de informação L1 varia de acordo com várias configurações/condições

Fig. 71

Info L1 (bits)	11760		
Bloco L1 (bits)	23520		
Símbolos 16-QAM	5880		
Portadoras totais	3408		
Distância SP do preâmbulo	6		
Portadoras de dados	2840		
LDPC curto	1.45		
Símbolos OFDM	2.07		

Dimensão máx.

Fig. 72

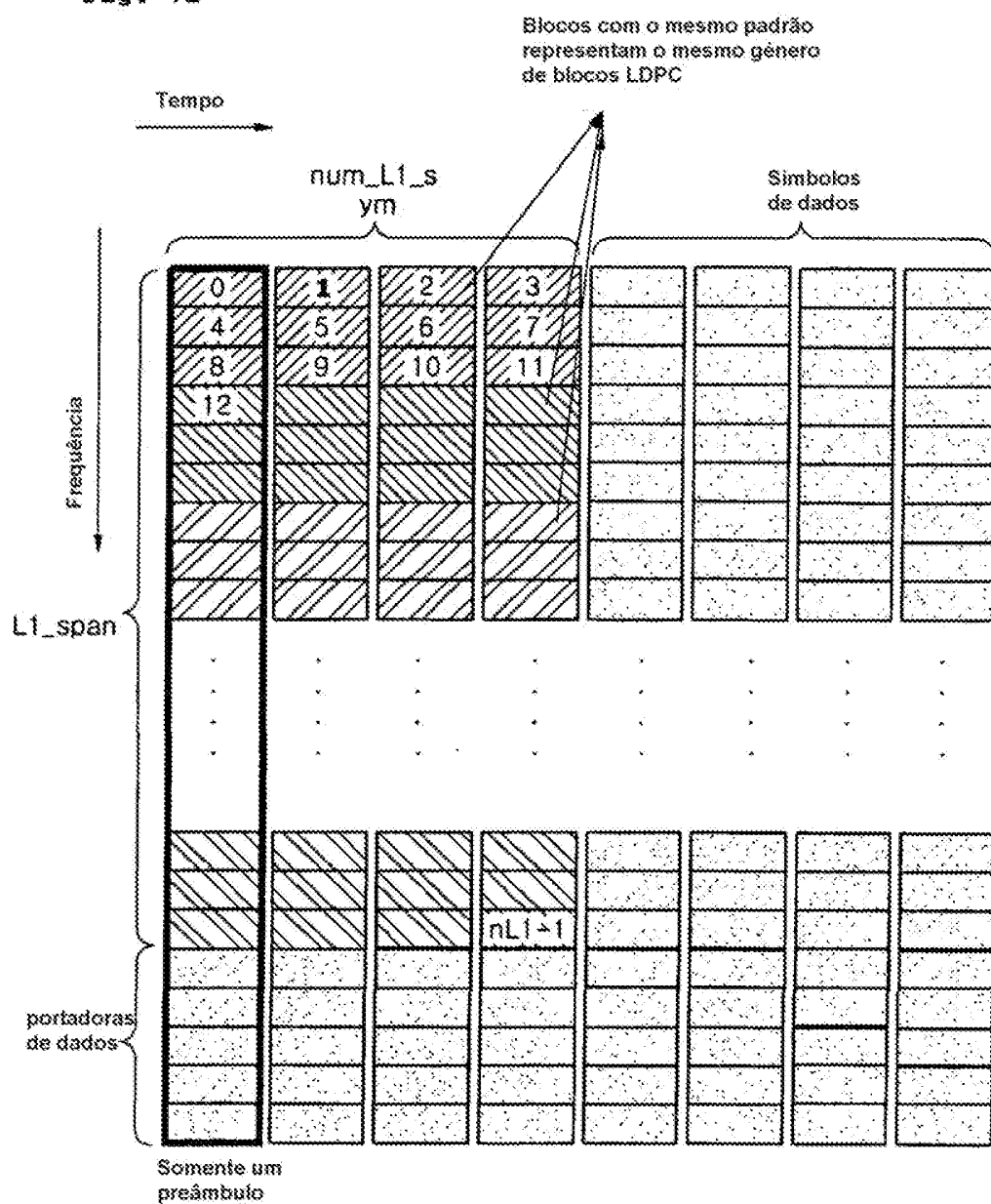


Fig. 73

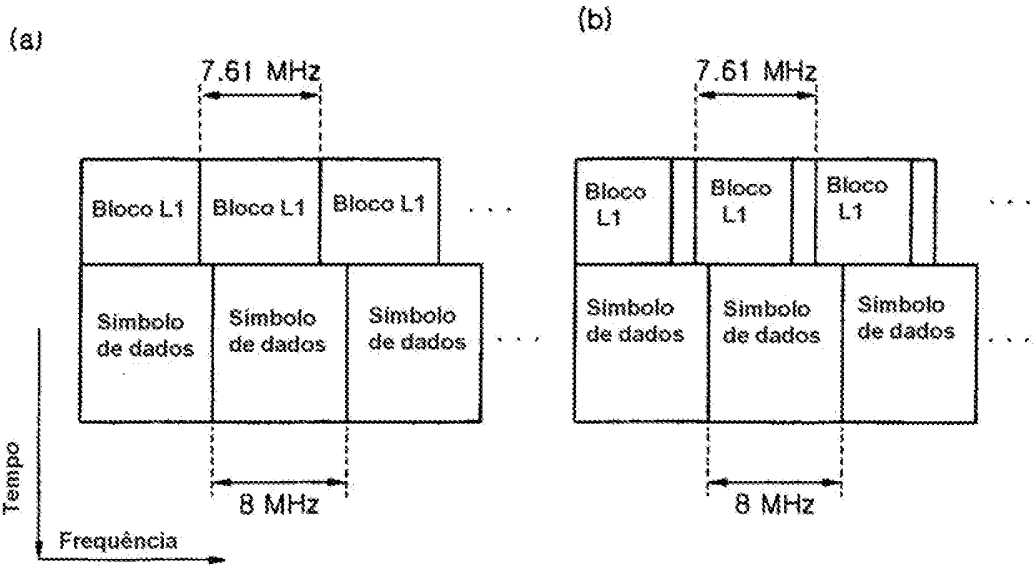


Fig. 74

Field	Bits	
L1_column	9	Número de portadoras fracionadas por bloco L1 dentro de um símbolo OFDM (máx = 7,68 MHz)
L1_row	3	número de símbolos OFDM fracionados por bloco L1
num_chbon	3	número de canais ligados
num_plp	8	número de parcelas de dados
num_dslice	8	número de PLP's
num_notch	5	número de bandas de entalhe
for dslice {		
chbon_index	3	índice de canal ligado
dslice_start	9	início da parcela de dados dentro de um canal (8 MHz)
dslice_width	9	largura da parcela de dados
}		
for plp {		
dslice_id	8	ID parcelado de dados
plp_id	8	PLP ID
plp_type	1	tipo de PLP (comunidades)
plp_payload_type	5	tipo de carga útil PLP (TS, GS,...)
}		
for notch {		
chbon_index	3	índice de canal ligado
notch_start	9	início da banda de entalhe dentro de um canal (8 MHz)
notch_width	9	largura da banda de entalhe
}		
gi	1	modo de intervalo de guarda
slrame_id	16	ID do superquadro
frame_id	16	ID do quadro
reserved	16	rfu
crc32	32	CRC32
Total	11776	

O número de bits de informação L1 varia de acordo com várias configurações/condições

Fig. 75

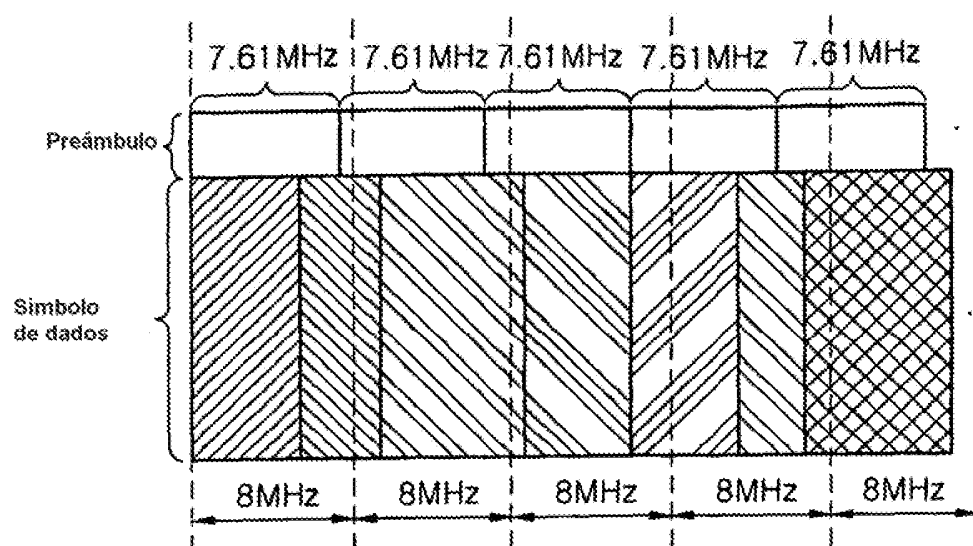


Fig. 76

Tamanho bloco LDPC	Modulação (bps/Hz)	Símbolos QAM	Informação complementar ModCod
64800	4	16200	0.28%
	6	10800	0.42%
	8	8100	0.56%
	10	6480	0.69%
	12	5400	0.83%
16200	4	4050	1.11%
	6	2700	1.67%
	8	2025	2.22%
	10	1620	2.78%
	12	1350	3.33%

Fig. 77

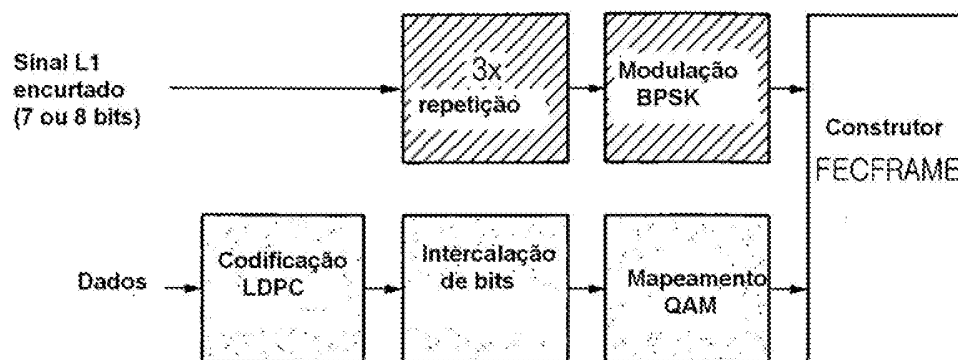


Fig. 78

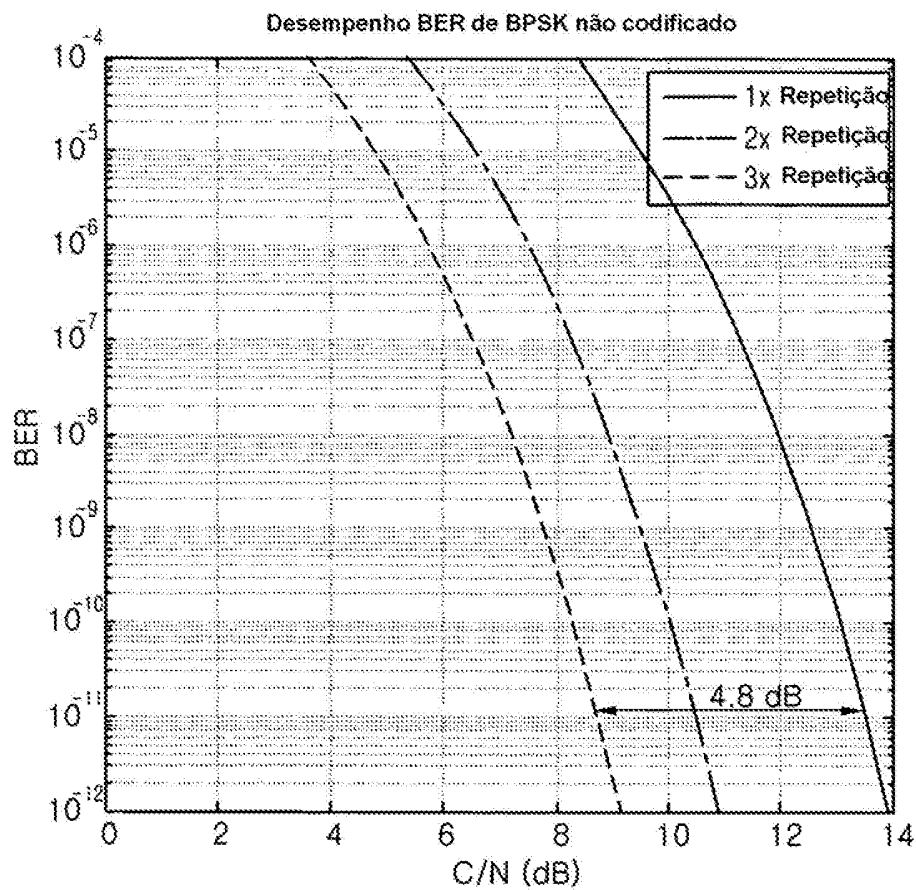


Fig. 79

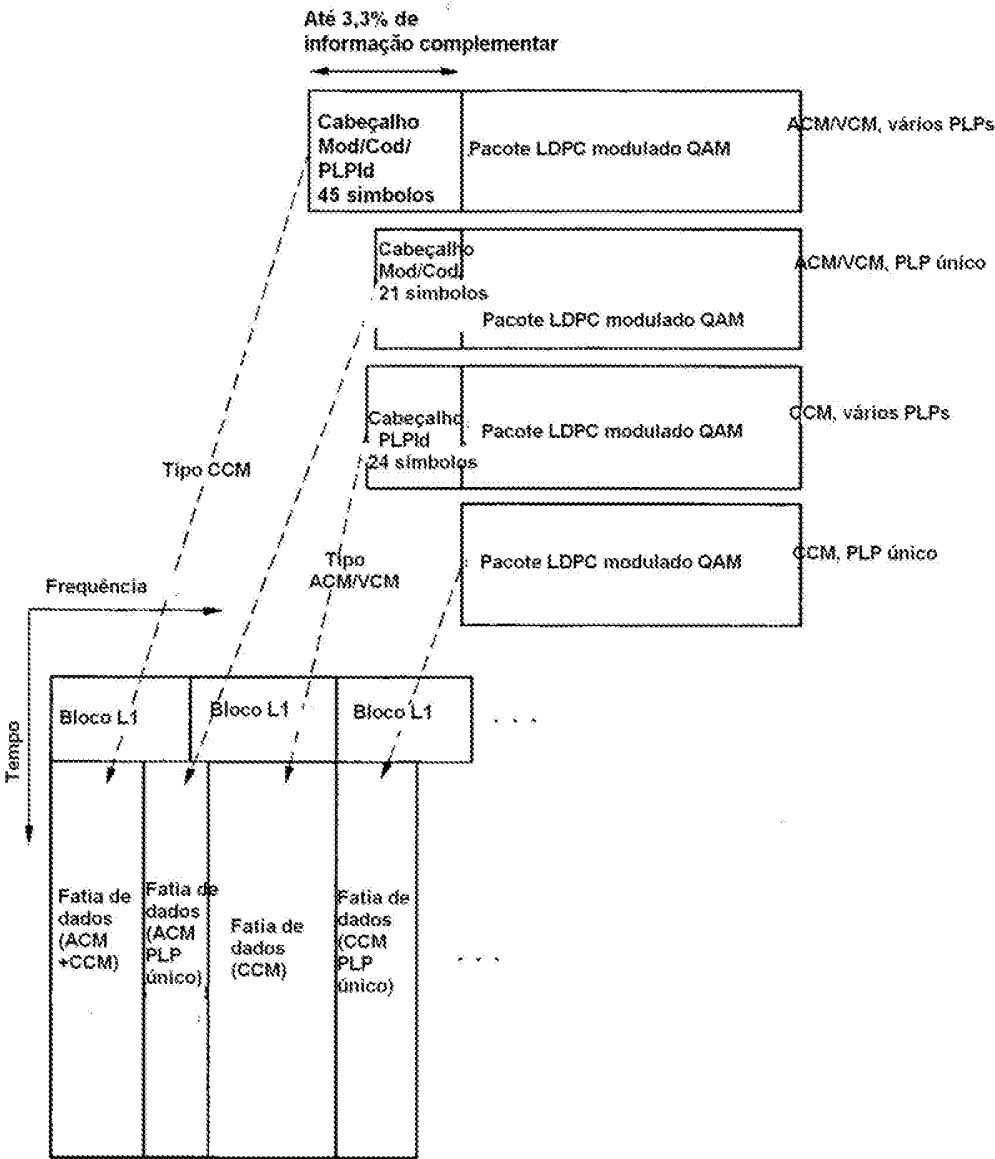


Fig. 80

	Campo	Bits	
	L1_size	14	Tamanho do bloco L1 (bits)
	num_chbon	3	número de canais ligados
	num_dslice	8	número de fatias de dados
	num_plp	8	número de PLPs
	num_notch	5	número de bandas de entalhe
	for dslice {		
	chbon_index	3	índice de canal ligado
	dslice_start	9	início de fatia de dados dentro de um canal (8 MHz)
	dslice_width	9	largura da fatia de dados
	dslice_time_intrlv	1	intercalação no tempo ligado/destligado
	dslice_type	1	tipo de fatia de dados (CCM, ACM/VCN)
	for plp {		
	dslice_id	8	ID em fatias de dados
	plp_id	8	ID PLP
	plp_type	1	tipo PLP
	plp_payload_type	5	carga útil PLP
	if dslice_type=CCM {		
	plp_mod	3	tipo de modulação PLP
	plp_fec_type	1	tipo PLP FEC (longo/curto)
	plp_cod	3	taxa de código PLP
	}		
	}		
	for notch {		
	chbon_index	3	índice de canal ligado
	notch_start	9	início da banda de entalhe dentro de um canal (8 MHz)
	notch_width	9	largura da banda de entalhe
	}		
	gi	1	modo de intervalo de guarda
	sframe_id	16	ID de superquadro
	frame_id	16	ID do quadro
	reserved	16	rfu
	crc32	32	CRC32
	Total	14082	

O tamanho do bloco L1 é transmitido no primeiro bloco LDPC (comprimento mais curto fixo = 192 bits)

Intercalação no tempo ligado/destligado para suportar modo de latência curto

Tipo de fatia de dados para redução da informação complementar da sinalização L1

Os campos Mod/Cod são transmitidos no preâmbulo somente para tipo CCM

Fig. 81

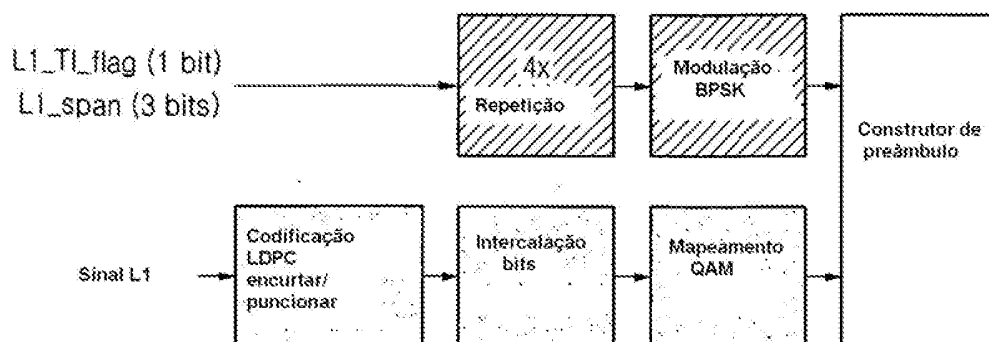


Fig. 82

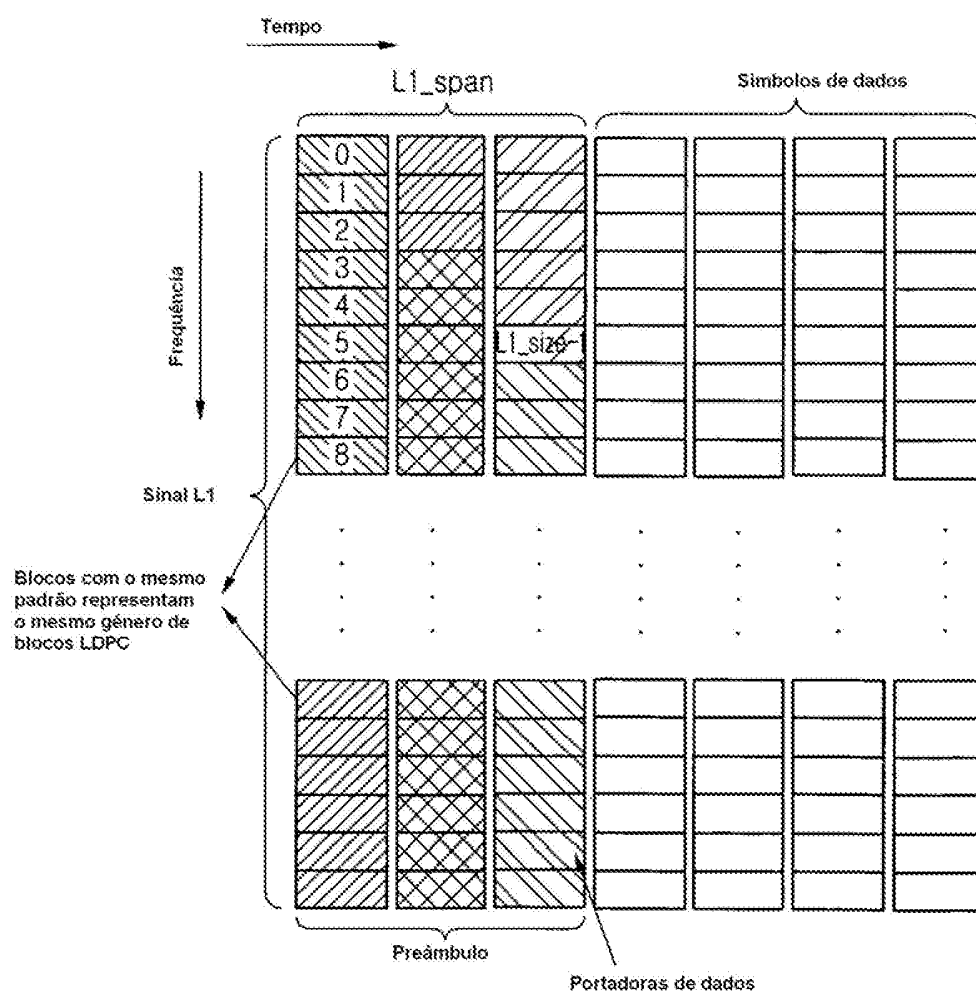


Fig. 83

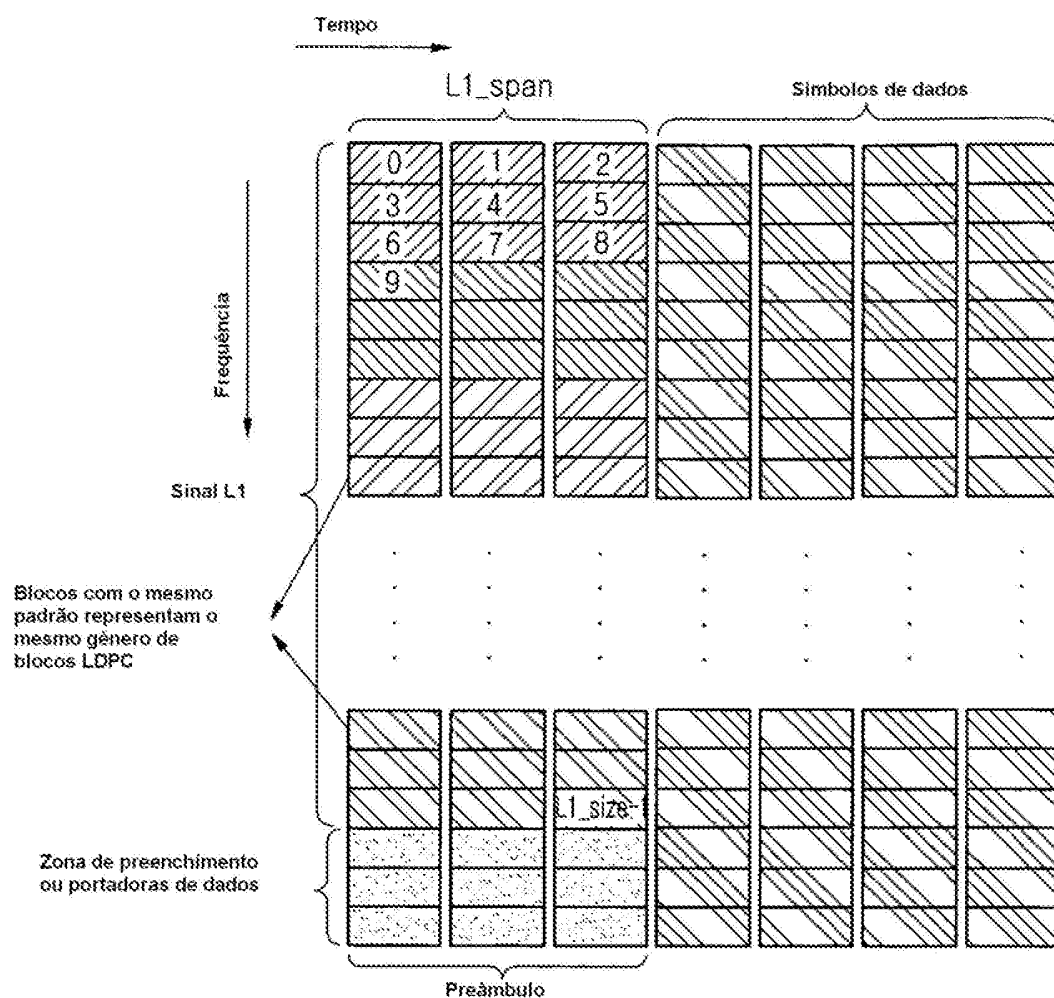


Fig. 84

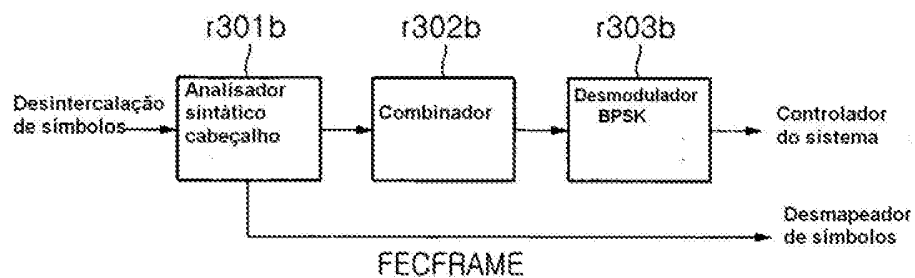


Fig. 85

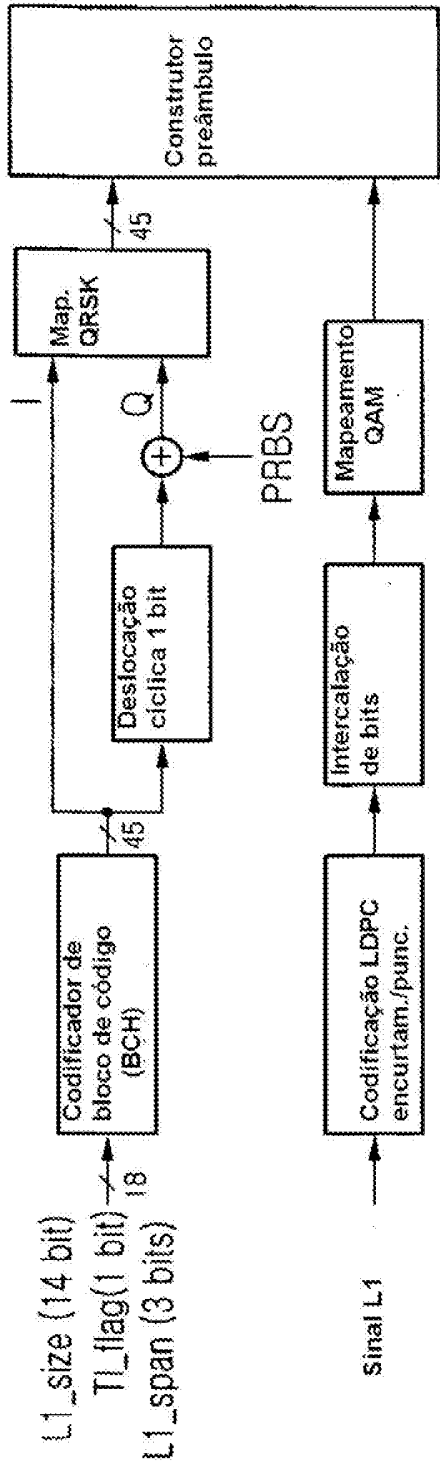


Fig. 86

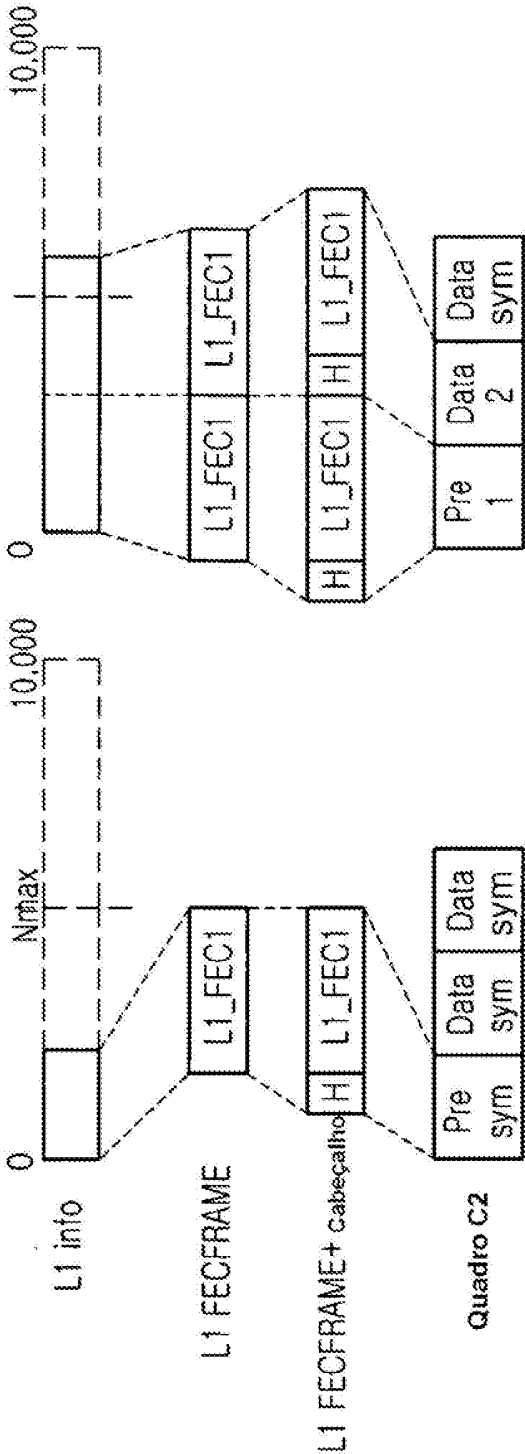


Fig. 87

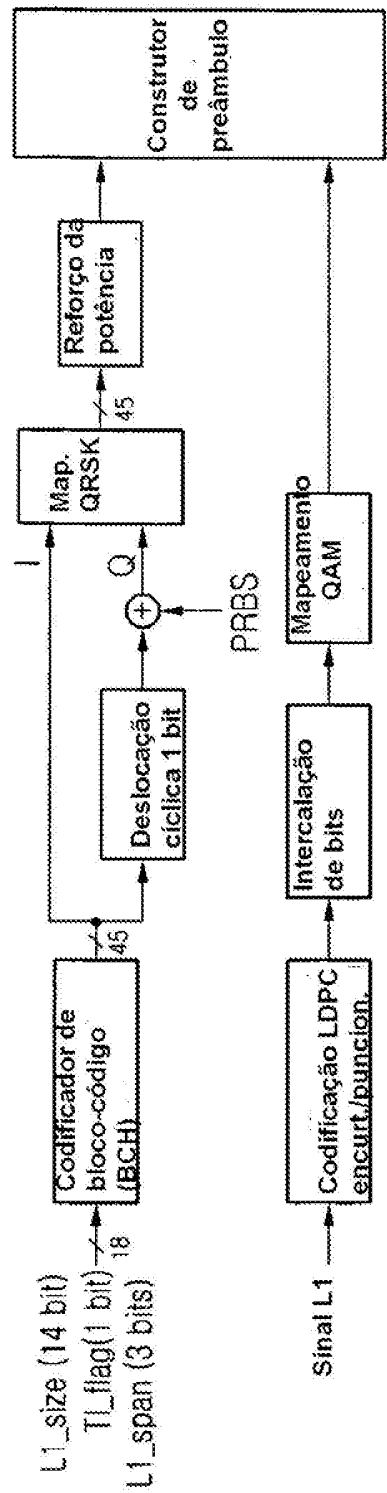


Fig. 88

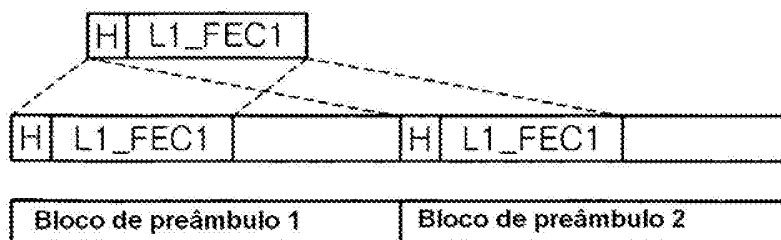


Fig. 89

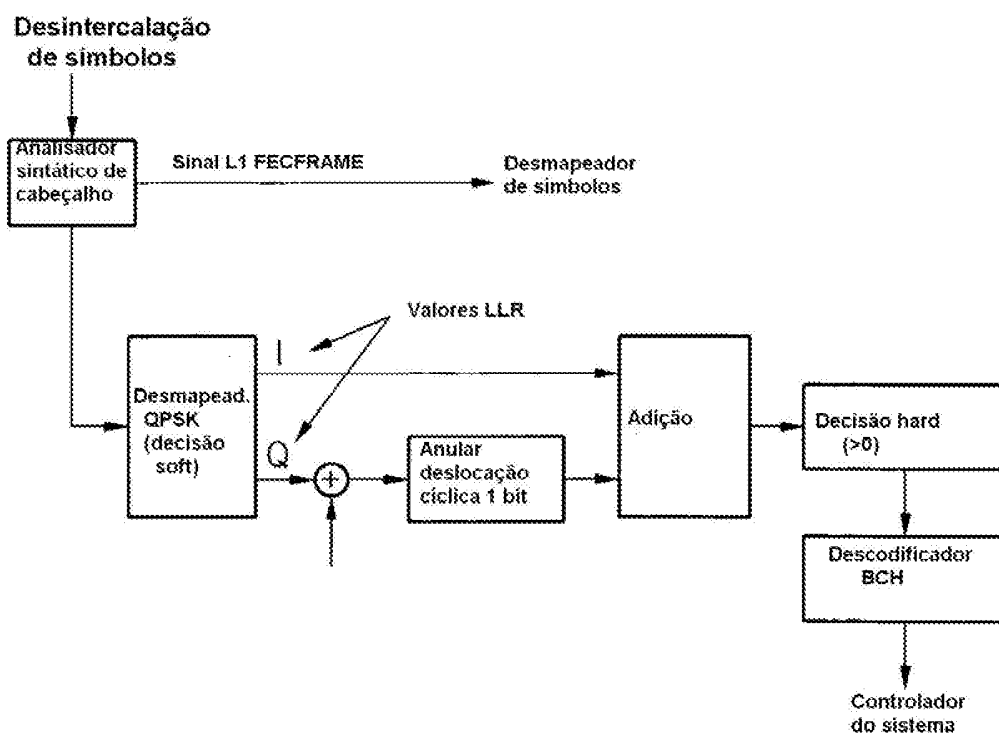


Fig. 90

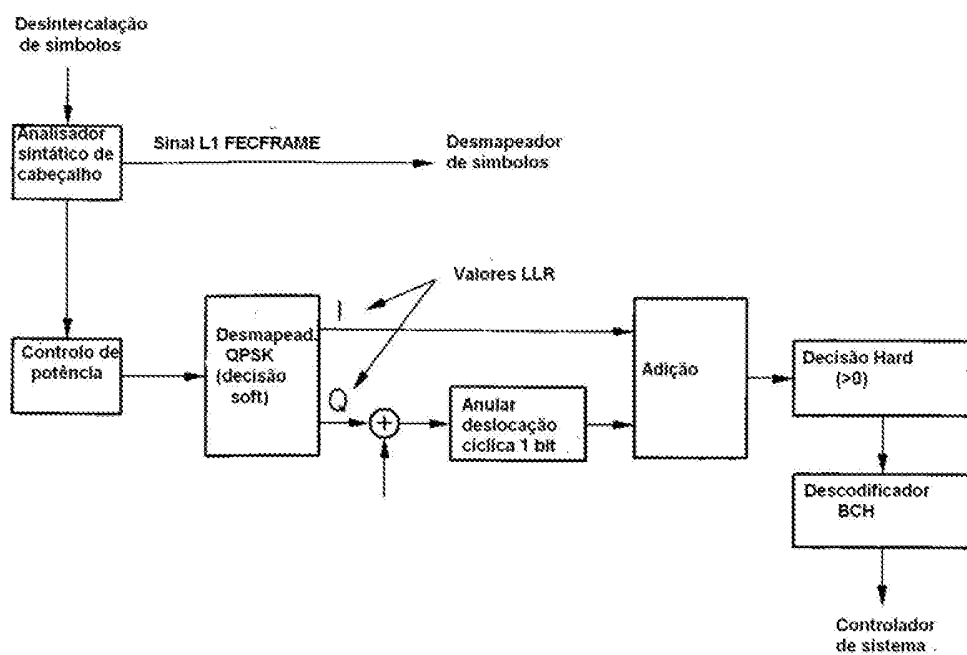


Fig. 91

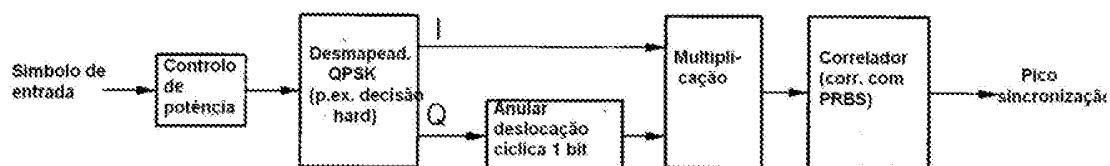


Fig. 92

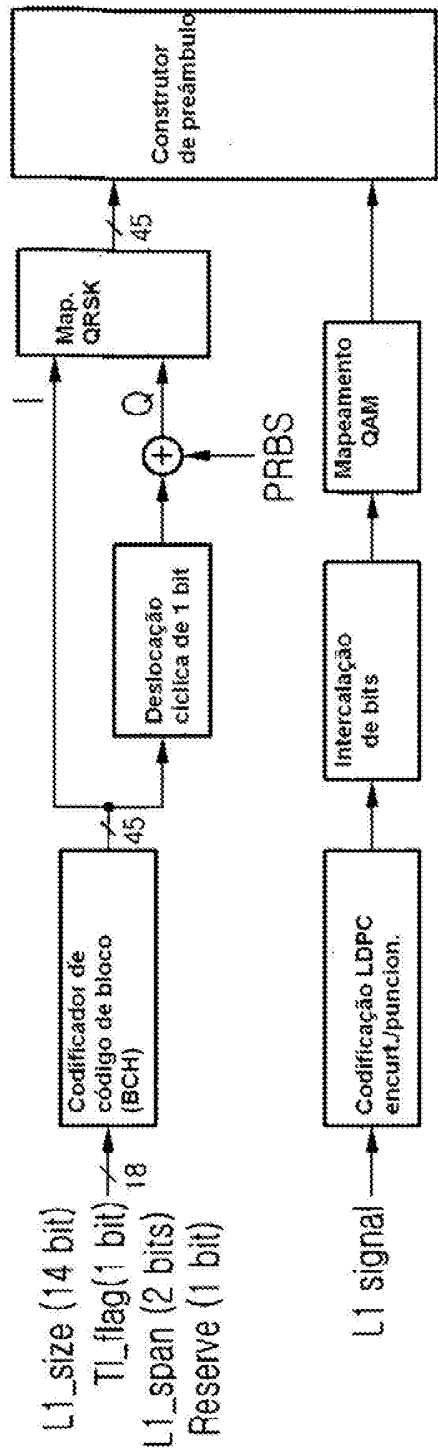


Fig. 93

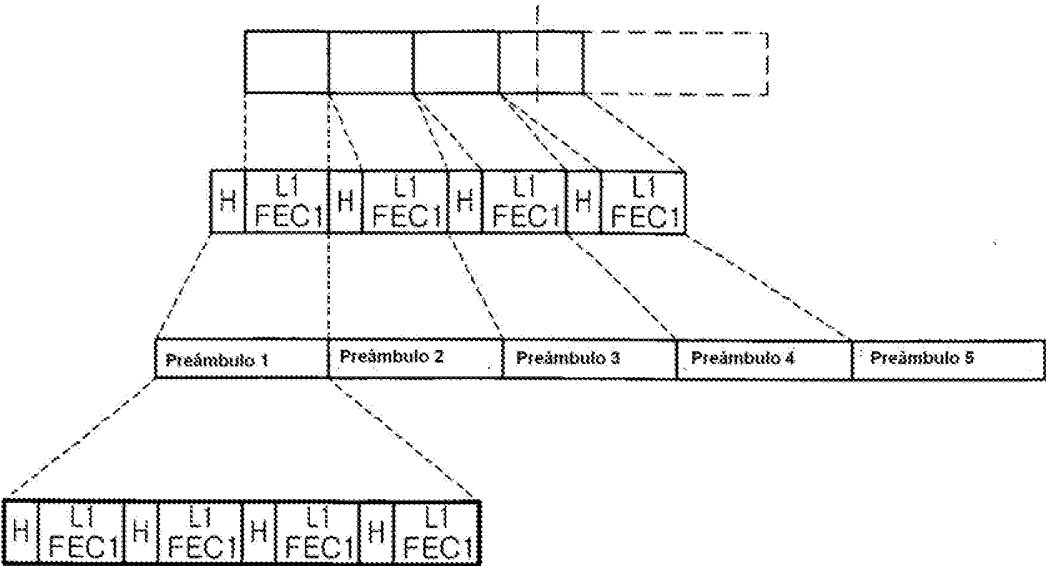


Fig. 94

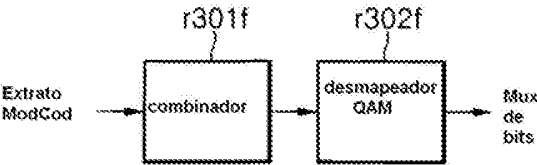


Fig. 95

L1_size	14	Tamanho do bloco L1 (bits)
num_chbon	3	número de canais ligados
num_dslice	8	número de fatias de dados
num_plp	8	número de PLPs
num_notch	5	número de bandas de entalhe
for dslice {		
chbon_index	3	índice de canal ligado
dslice_start	9	início de fatia de dados dentro de um canal (8 MHz)
dslice_width	9	largura da fatia de dados
dslice_time_intrlv	1	intercalação no tempo ligado/destigado
dslice_type	1	tipo de fatia de dados (CCM, ACM/VCM)
}		
for plp {		
dslice_id	8	ID em fatias de dados
plp_id	8	ID PLP
plp_type	1	tipo PLP (comum/dados)
plp_payload_type	5	carga útil PLP (TS, GS, ...)
if dslice_type=CCM {		
plp_mod	3	tipo de modulação PLP
plp_fec_type	1	tipo PLP FEC (longo/curto)
plp_cod	3	taxa de código PLP
plp_start	21	endereço de início PLP
}		
}		
for notch {		
chbon_index	3	índice de canal ligado
notch_start	9	início da banda de entalhe dentro de um canal (8 MHz)
notch_width	9	largura da banda de entalhe
}		
gi	1	modo de intervalo de guarda
sframe_id	16	ID de superquadro
frame_id	16	ID do quadro
reserved	16	rfu
crc32	32	CRC32
Total	19458	

O número de bits de informação L1
varia de acordo com varias configurações/condições

Fig. 96

L1_size	14	Tamanho do bloco L1 (bits)
num_chbon	3	número de canais ligados
num_dslice	8	número de fatias de dados
num_plp	8	número de PLPs
num_notch	5	número de bandas de entalhe
for dslice {		
chbon_index	3	índice de canal ligado
dslice_start	9	início de fatia de dados dentro de um canal (8 MHz)
dslice_width	9	largura da fatia de dados
dslice_time_intrlv	1	intercalação no tempo ligado/destigado
dslice_type	1	tipo de fatia de dados (CCM, ACM/VCM)
}		
for plp {		
dslice_id	8	ID em fatias de dados
plp_id	8	ID PLP
plp_type	1	tipo PLP (comum/dados)
plp_payload_type	5	carga útil PLP (TS, GS, ...)
if dslice_type=CCM {		
plp_mod	3	tipo de modulação PLP
plp_fec_type	1	tipo PLP FEC (longo/curto)
plp_cod	3	taxa de código PLP
plp_start	14	endereço de início PLP
}		
}		
for notch {		
chbon_index	3	índice de canal ligado
notch_start	9	início da banda de entalhe dentro de um canal (8 MHz)
notch_width	9	largura da banda de entalhe
}		
gi	1	modo de intervalo de guarda
sframe_id	16	ID de superquadro
frame_id	16	ID do quadro
reserved	16	rfu
crc32	32	CRC32
Total	17666	

O número de bits de informação L1
varia de acordo com varias configurações/condições

Fig. 97

Comprimento LDPC	Tipo QAM	Simbolos QAM
64800	16 QAM	16200
64800	64 QAM	10800
64800	256 QAM	8100
64800	1024 QAM	6480
64800	4096 QAM	5400
16200	16 QAM	4050
16200	64 QAM	2700
16200	256 QAM	2025
16200	1024 QAM	1620
16200	4096 QAM	1350

Fig. 98

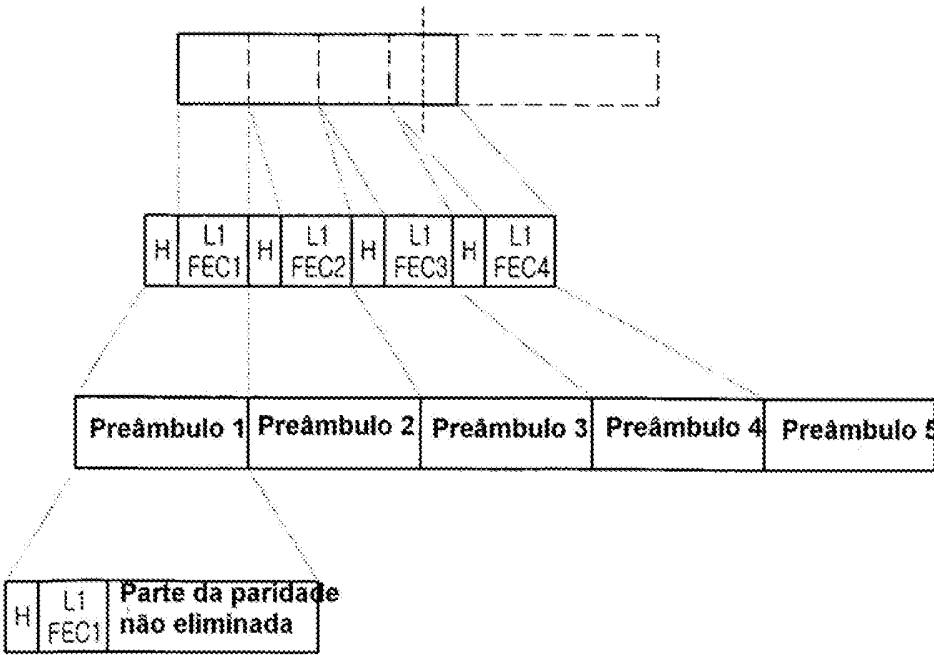


Fig. 99

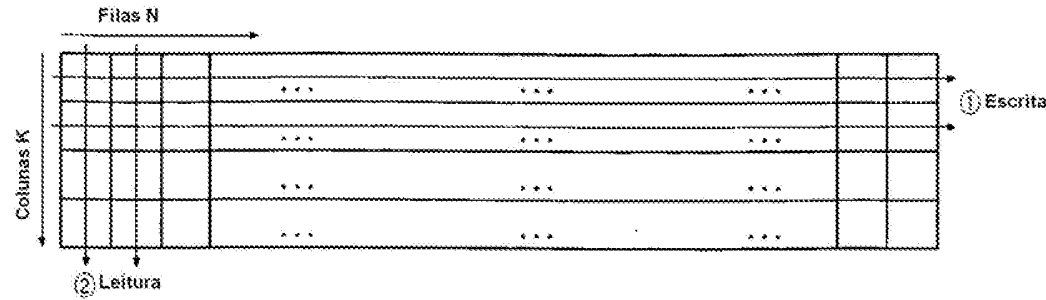


Fig. 100

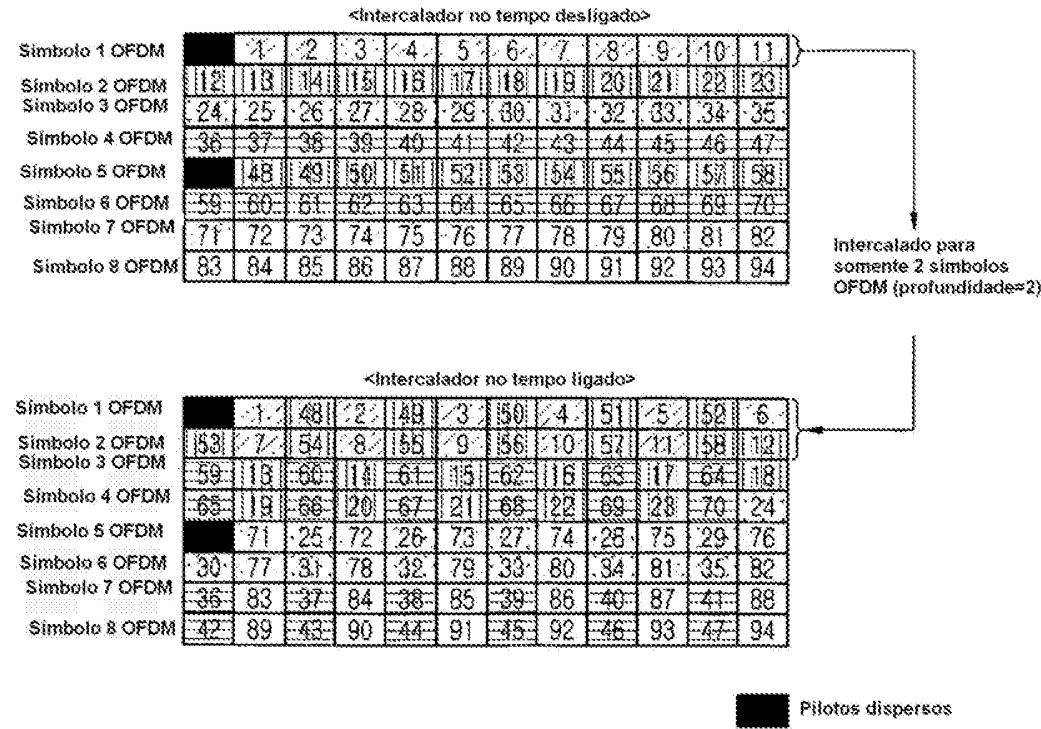


Fig. 101

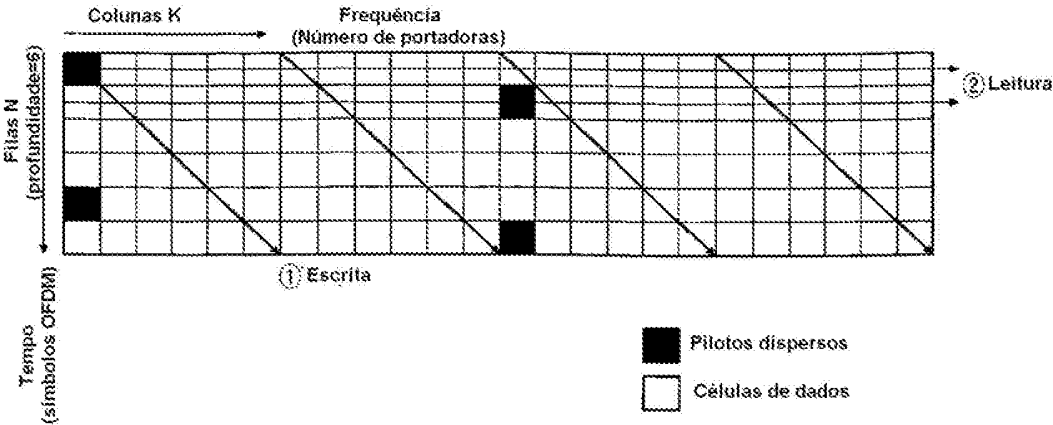


Fig. 102

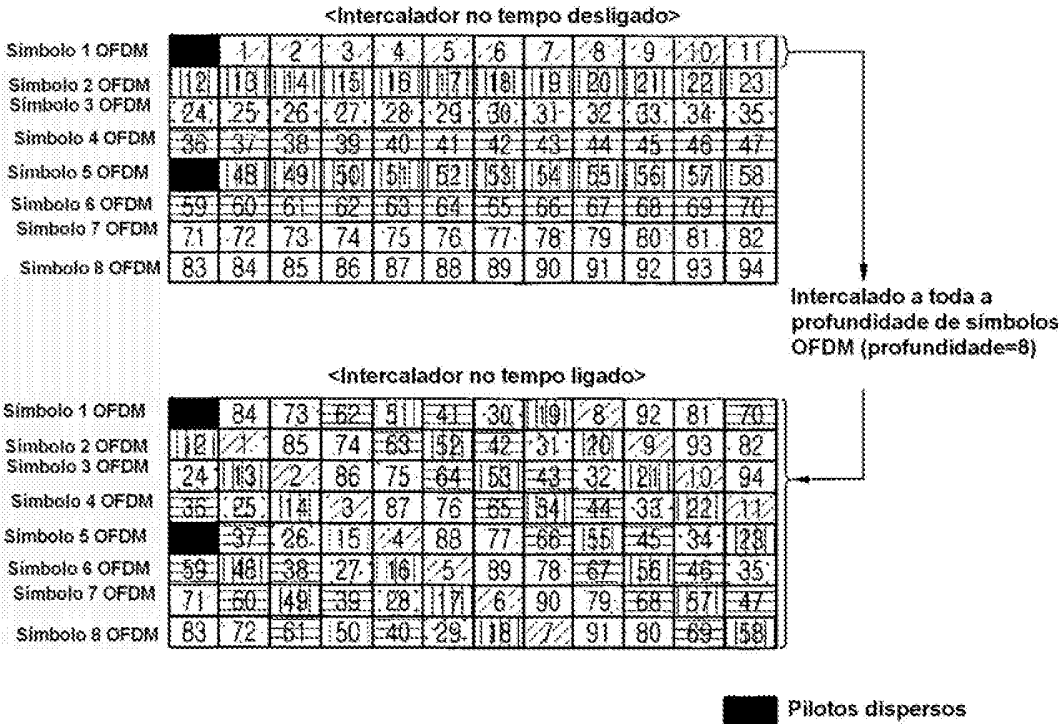


Fig. 103

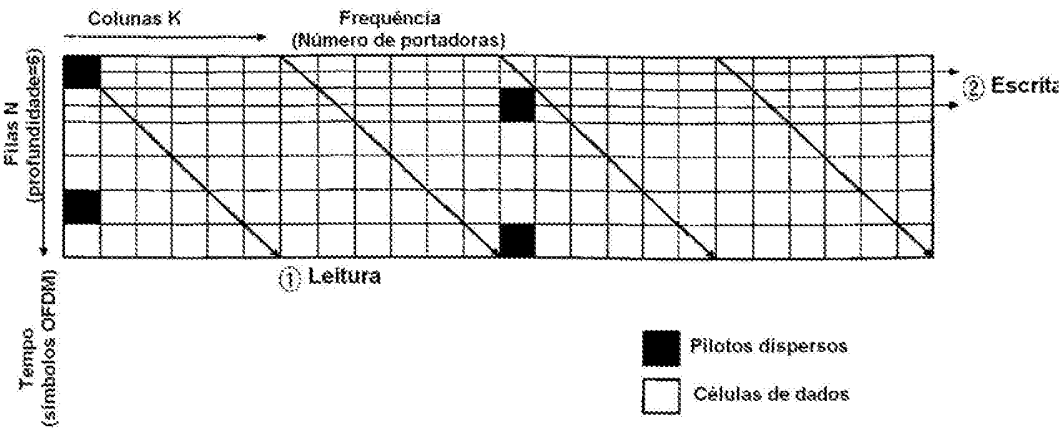


Fig. 104

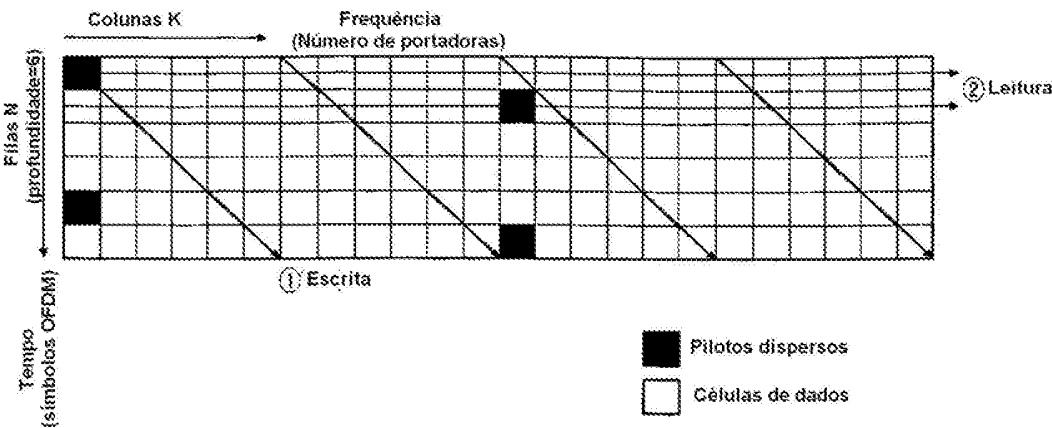


Fig. 105

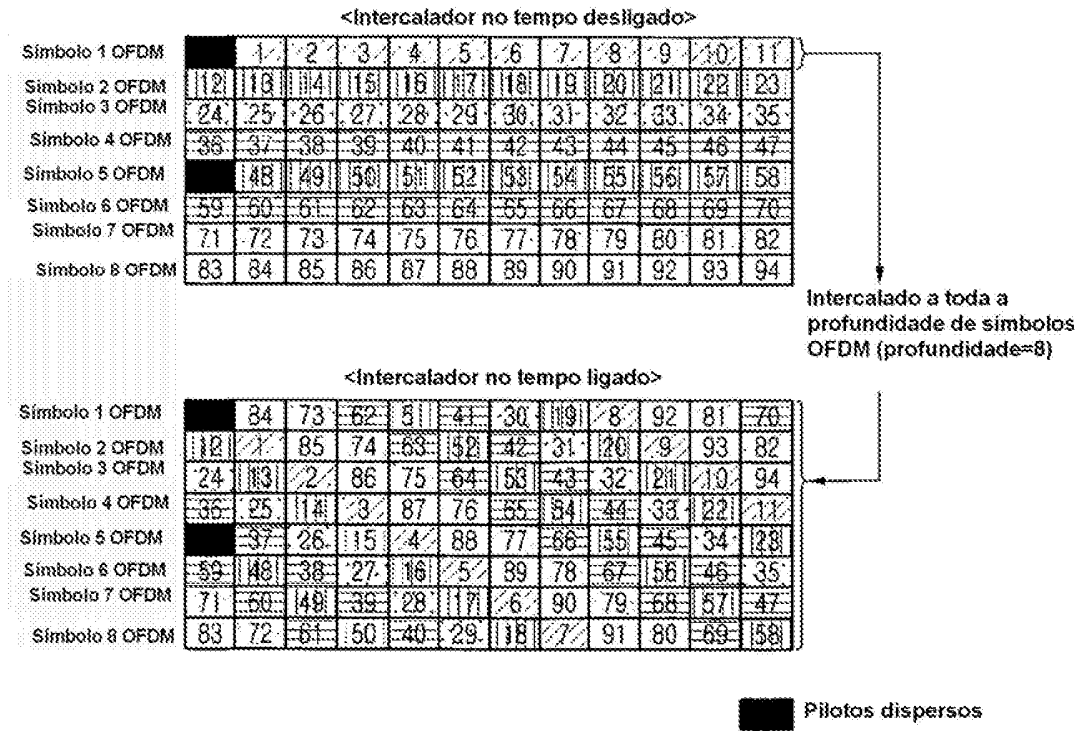
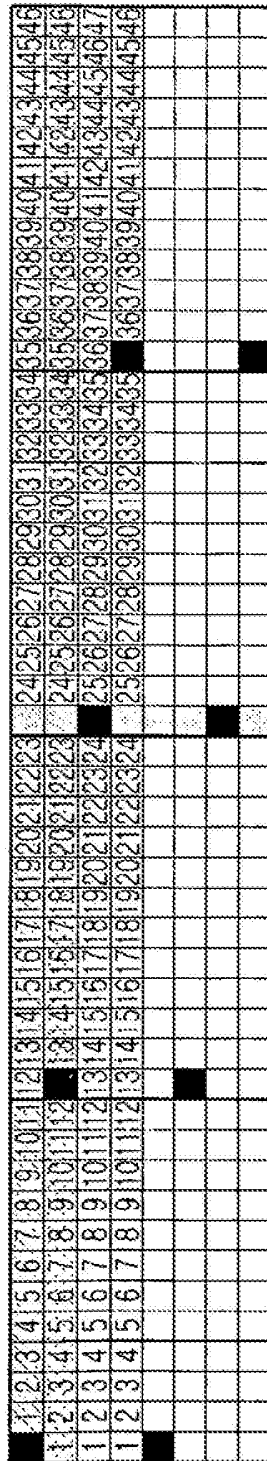
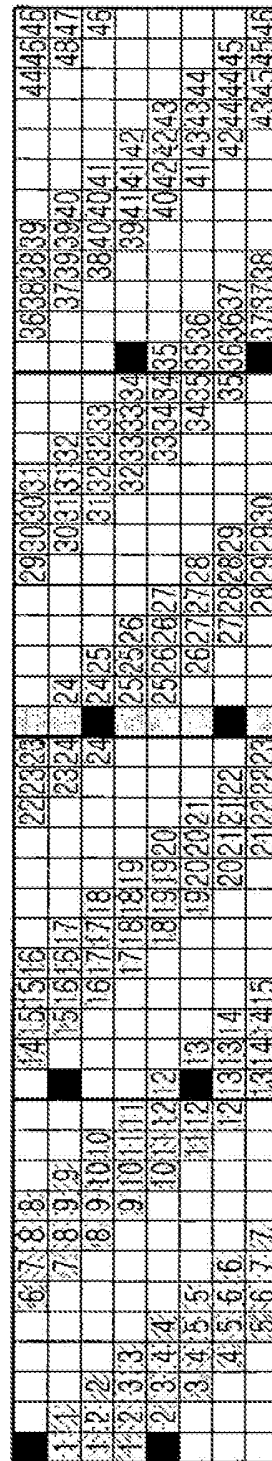


Fig. 106

Após aplicar intercalação
no tempoAntes de aplicar intercalação
no tempo

RA=0; CA=0;

while K<nCELL loop

 If addr = pilot position

 RA = [RA+1] mod NT;

 CA = [CA+1] mod ND;

 end if ;

WRITE:

RA = [RA+1] mod NT;

CA = [CA+1] mod ND;

 if CA = 0

 RA = RA+1;

 end if ;

end loop;

■ Piloto disperso

□ Piloto contínuo

RA = Endereço da linha

CA = Endereço da coluna

nCell = # de células dentro do
bloco TI

Informação complementar
da memória = 2%

Fig. 107

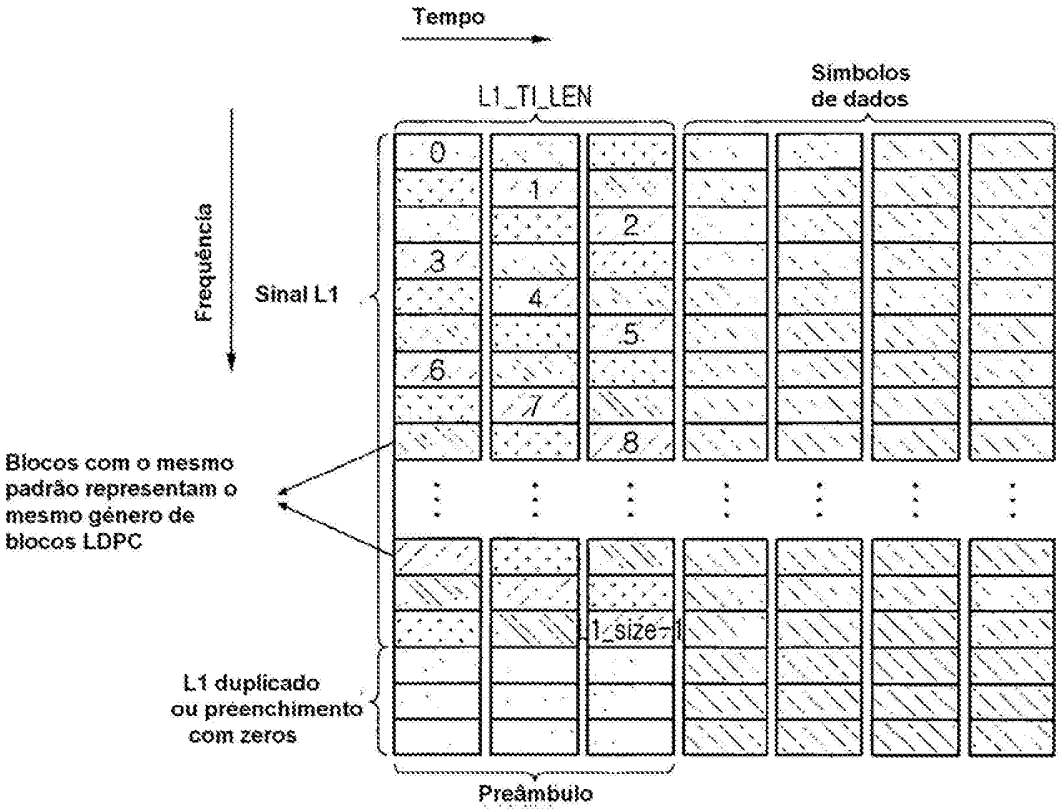


Fig. 108

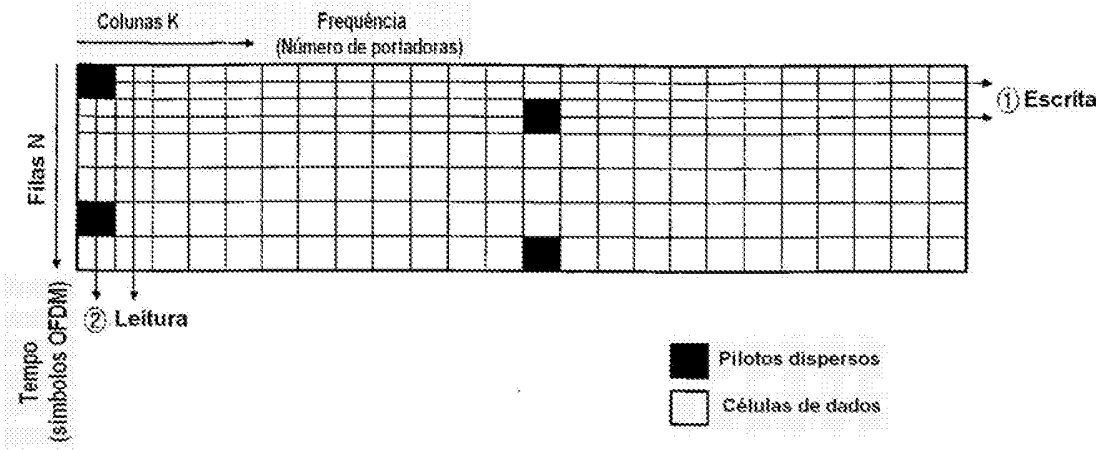


Fig. 109

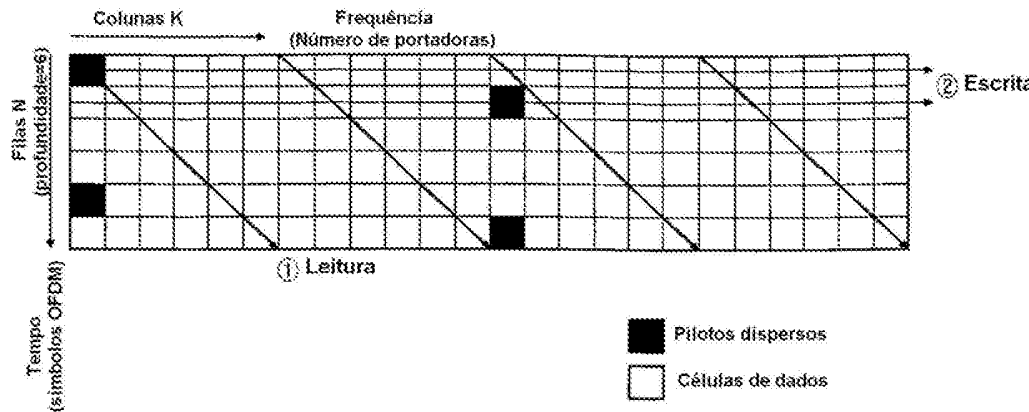


Fig. 110

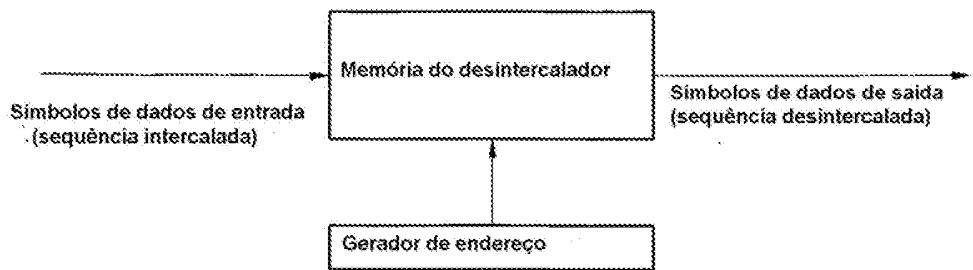


Fig. 111

	i=0		i=12		i=23	
i=0	R _{i,j}	0 0 0 0 0 0 0 0 0 0 0 0	R _{i,j}	1 1 1 1 1 1 1 1 1 1 1 1	R _{i,j}	1 1 1 1 1 1 1 1 1 1 1 1
	C _{i,j}	0 1 2 3 4 5 6 7 8 9 10 11		C _{i,j}	0 1 2 3 4 5 6 7 8 9 10 11	
i=1	R _{i,j}	0 1 2 3 4 5 6 7 0 1 2 3	R _{i,j}	1 2 3 4 5 6 7 0 1 2 3 4	R _{i,j}	1 2 3 4 5 6 7 0 1 2 3 4
	C _{i,j}	0 1 2 3 4 5 6 7 8 9 10 11		C _{i,j}	0 1 2 3 4 5 6 7 8 9 10 11	
i=2	R _{i,j}	0 2 4 6 0 2 4 6 0 2 4 6	R _{i,j}	1 3 5 7 1 3 5 7 1 3 5 7	R _{i,j}	1 3 5 7 1 3 5 7 1 3 5 7
	C _{i,j}	0 1 2 3 4 5 6 7 8 9 10 11		C _{i,j}	0 1 2 3 4 5 6 7 8 9 10 11	
i=3	R _{i,j}	0 3 6 1 4 7 2 5 0 3 6 1	R _{i,j}	1 4 7 2 5 0 3 6 1 4 7 2	R _{i,j}	1 4 7 2 5 0 3 6 1 4 7 2
	C _{i,j}	0 1 2 3 4 5 6 7 8 9 10 11		C _{i,j}	0 1 2 3 4 5 6 7 8 9 10 11	

Fig. 112

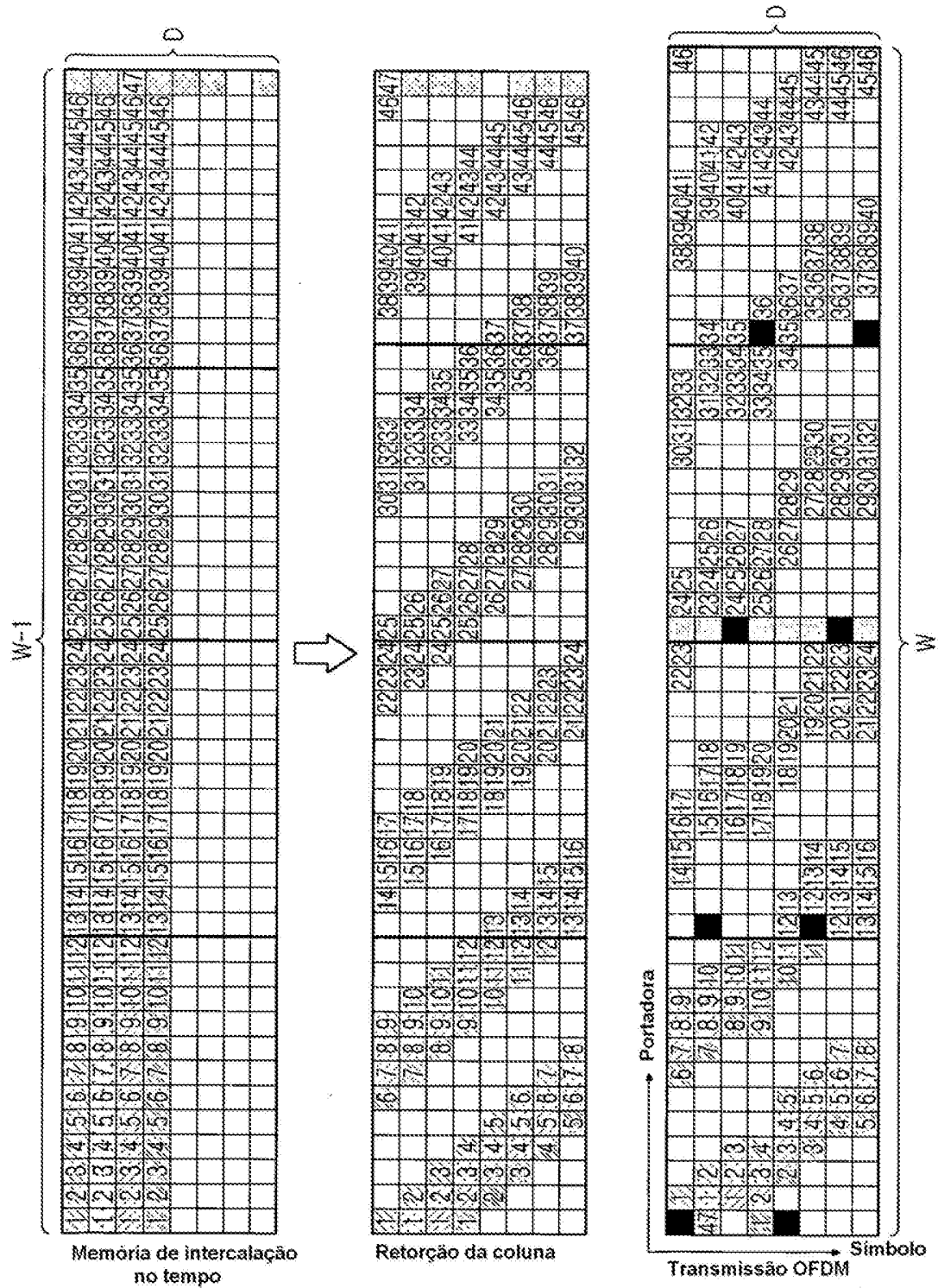


Fig. 113

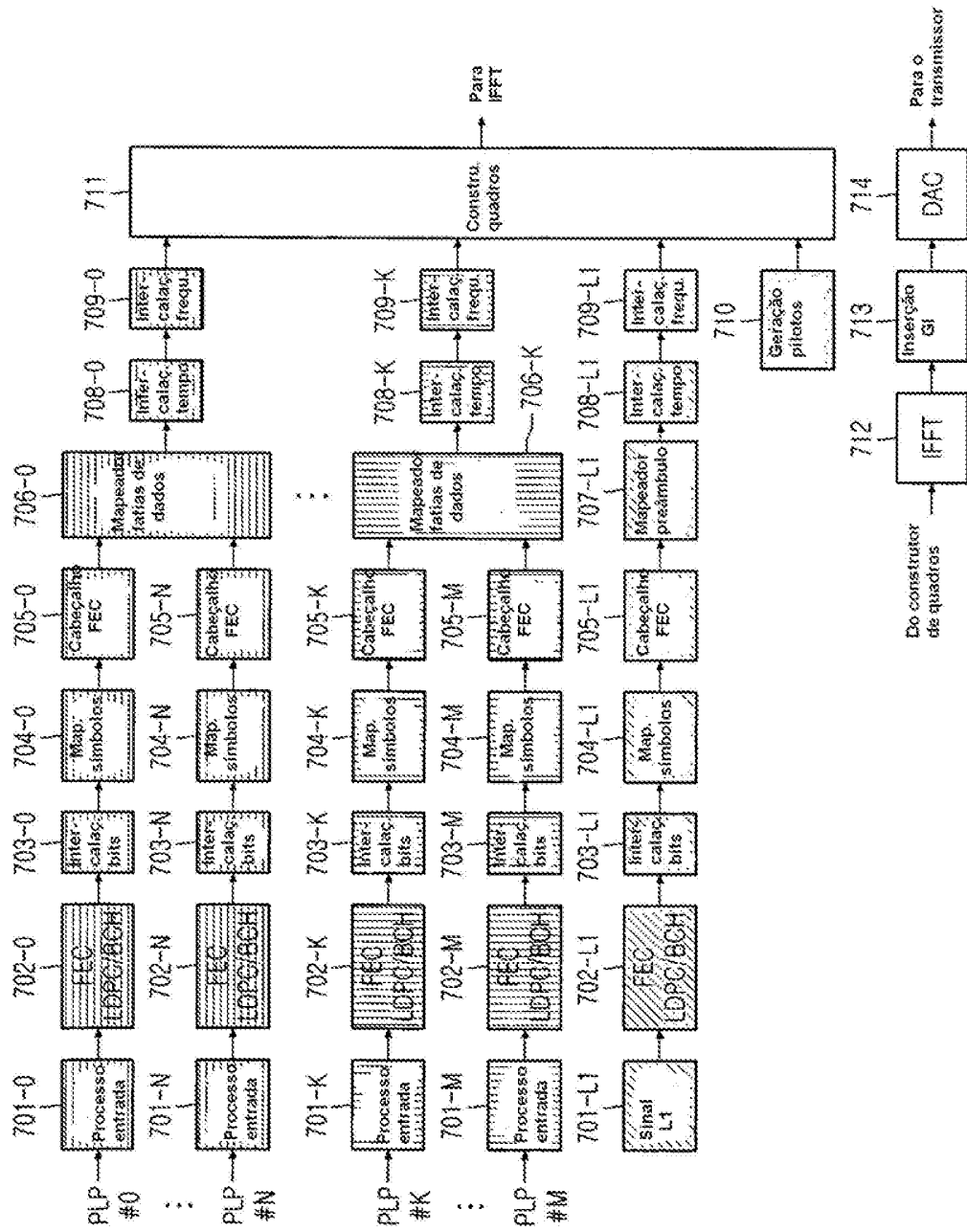


Fig. 114

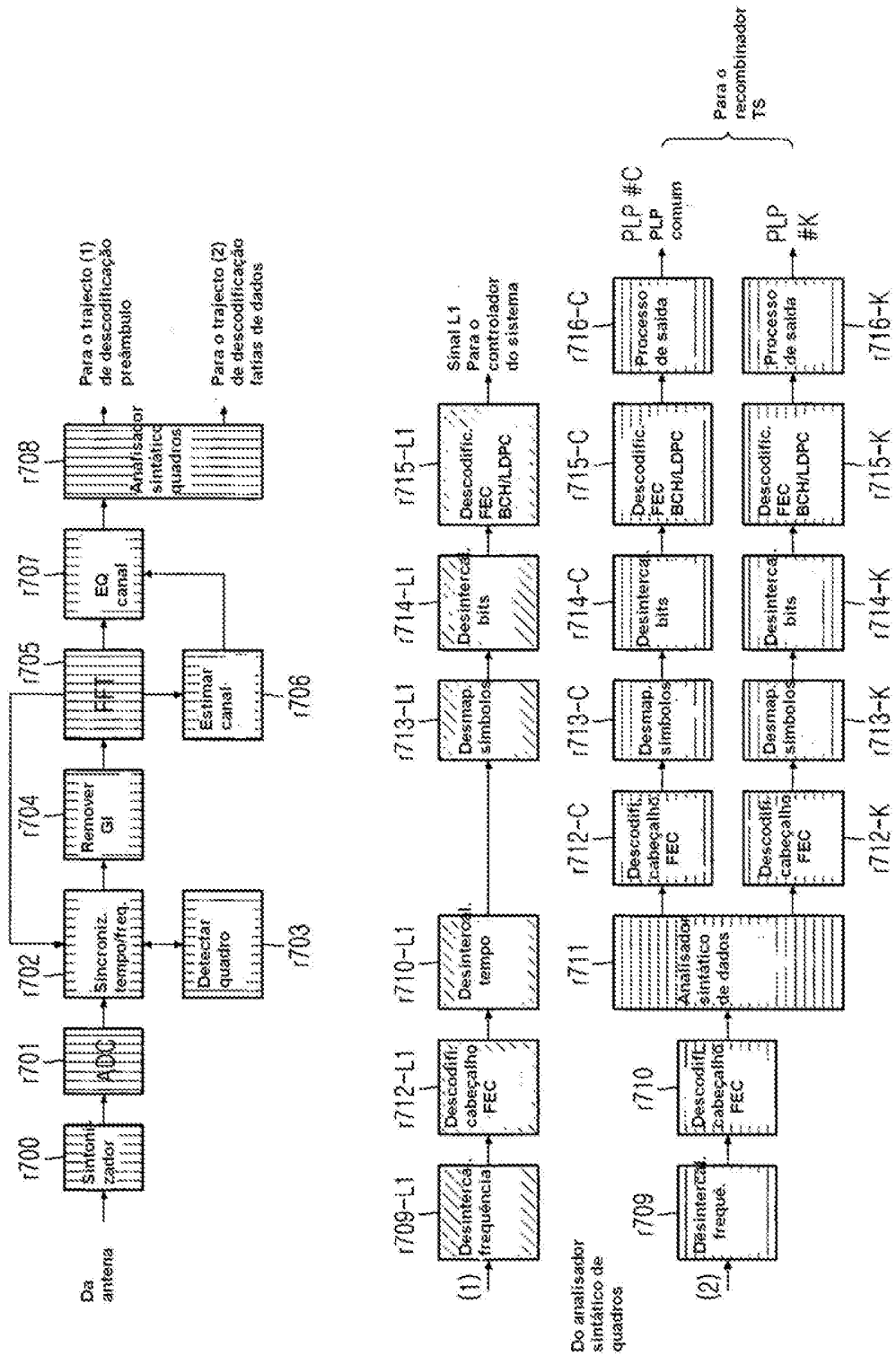


Fig. 115

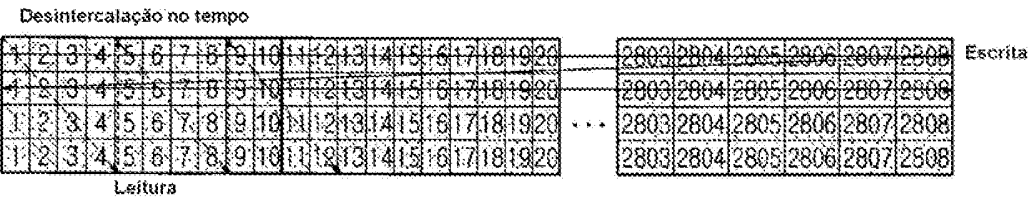
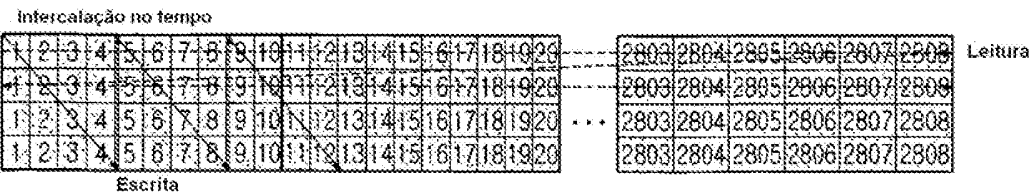


Fig. 116

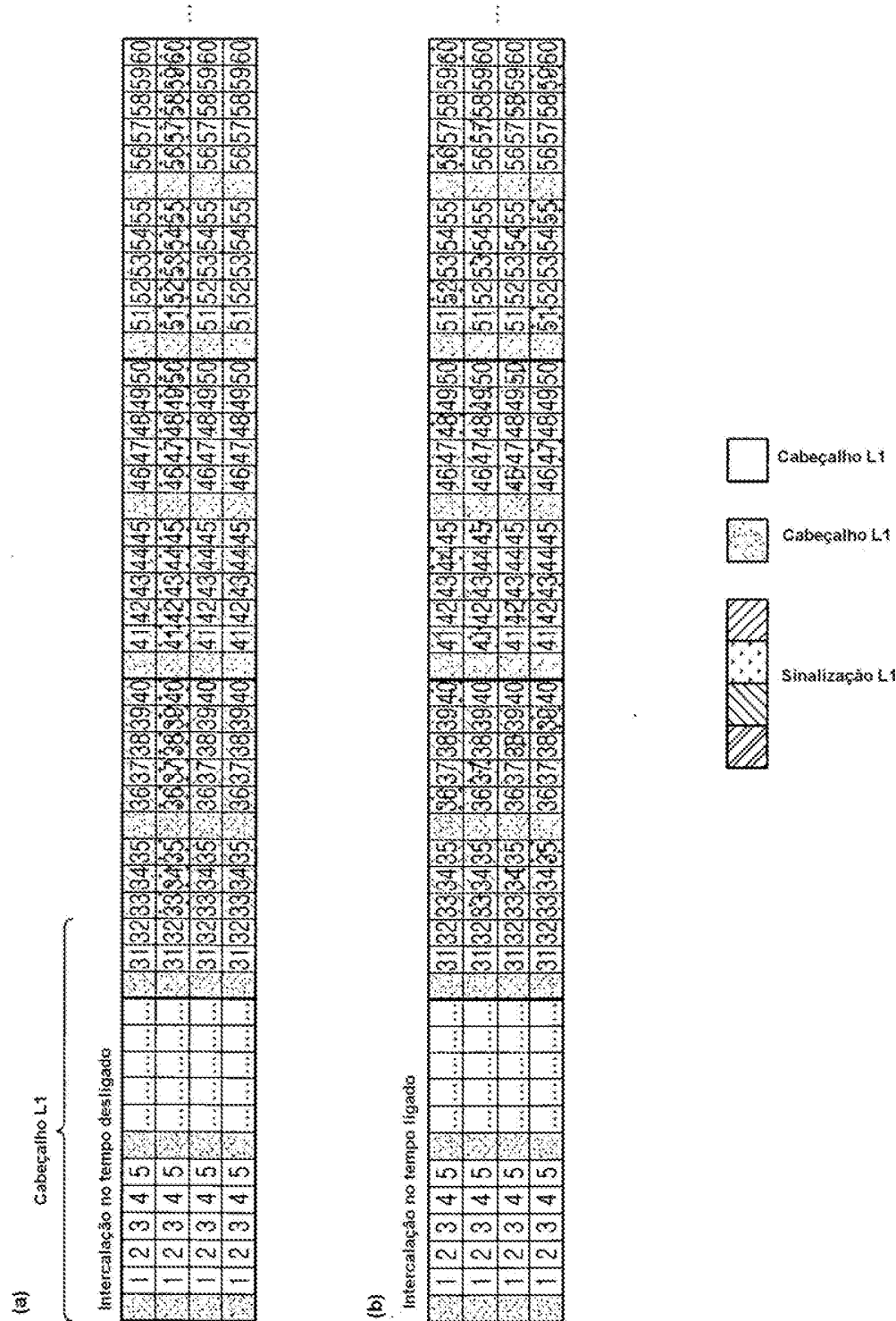


Fig. 118

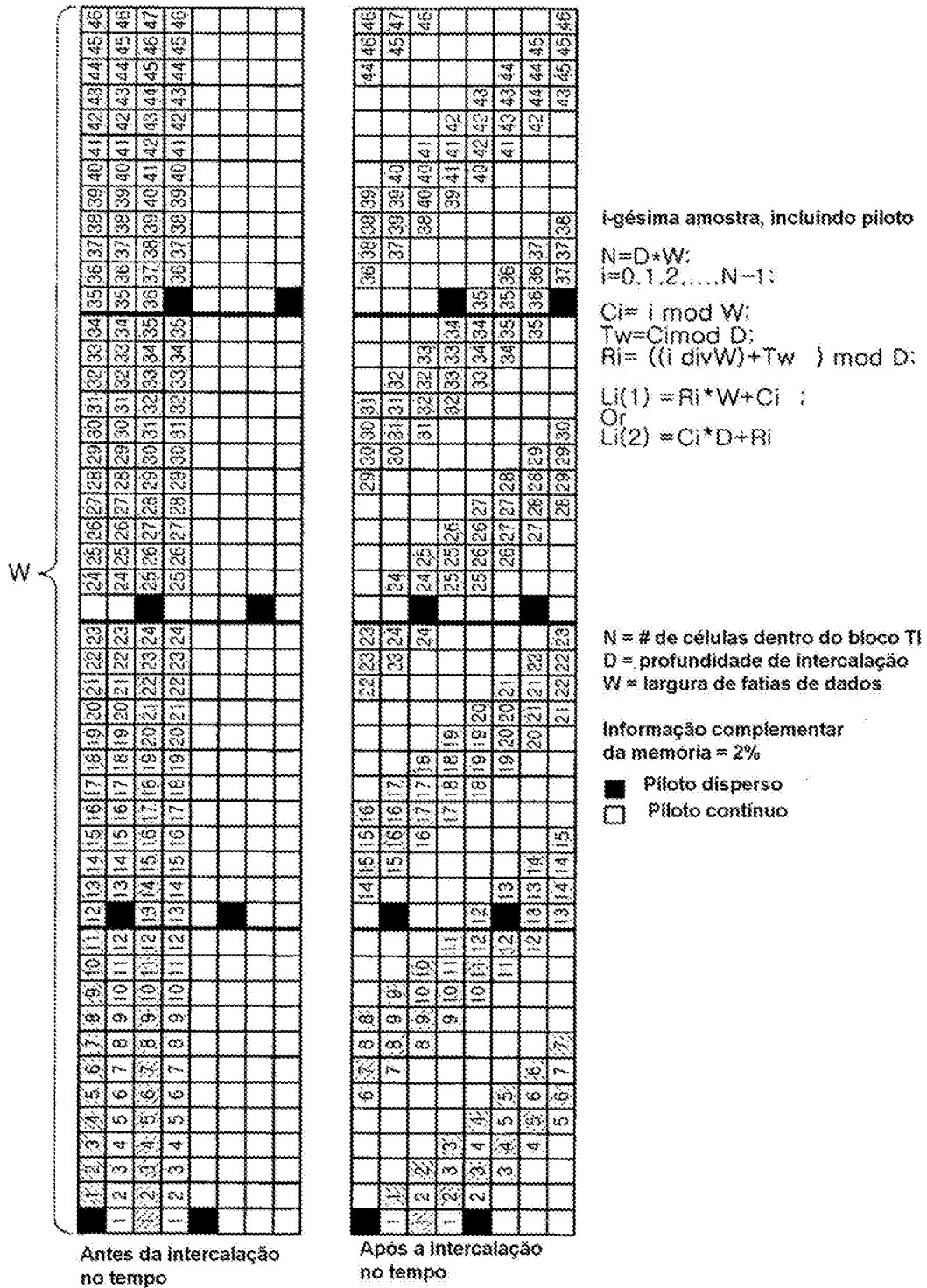


Fig. 119

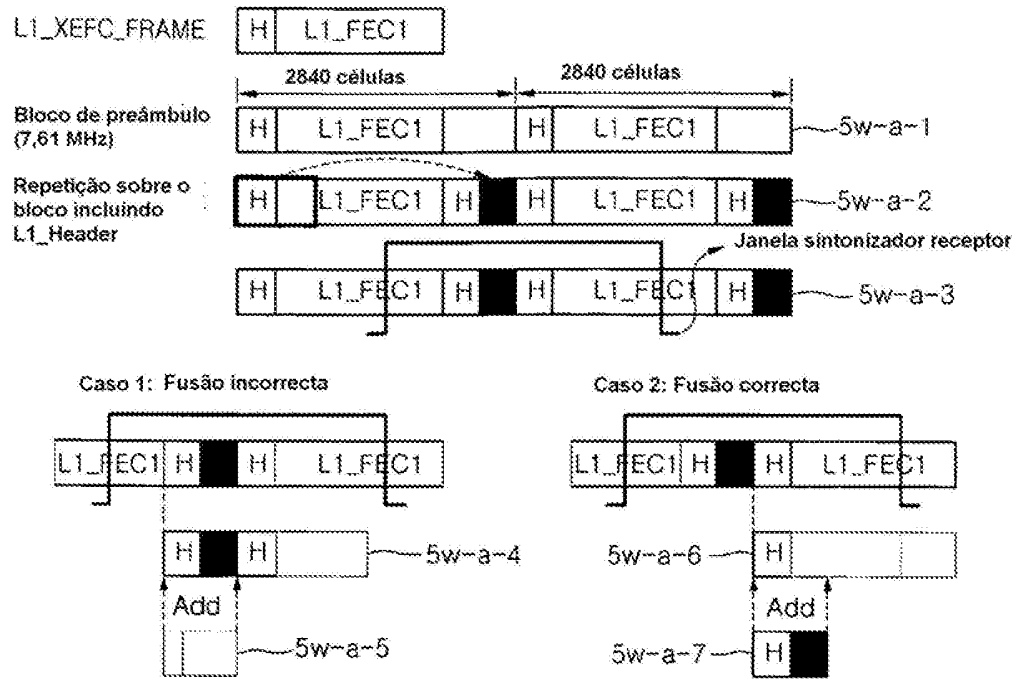


Fig. 120

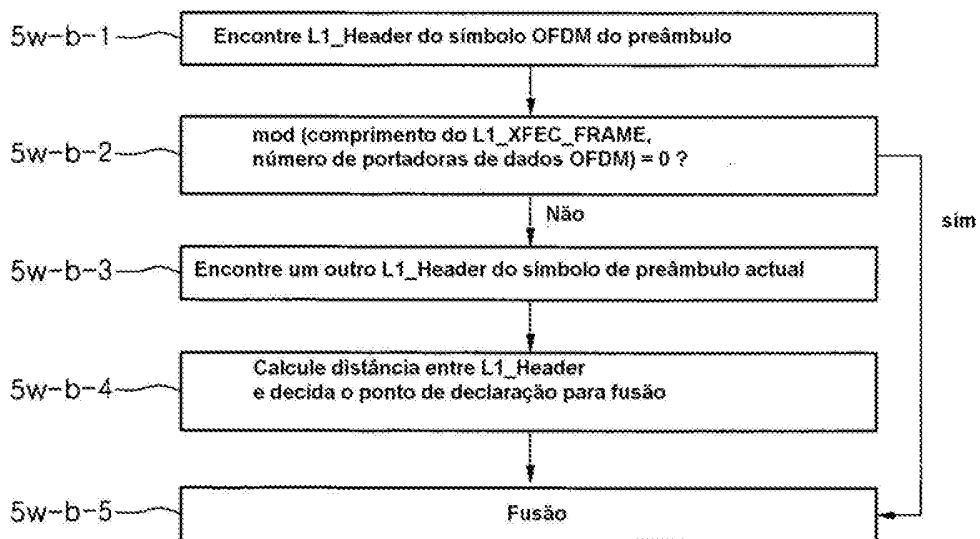


Fig. 121

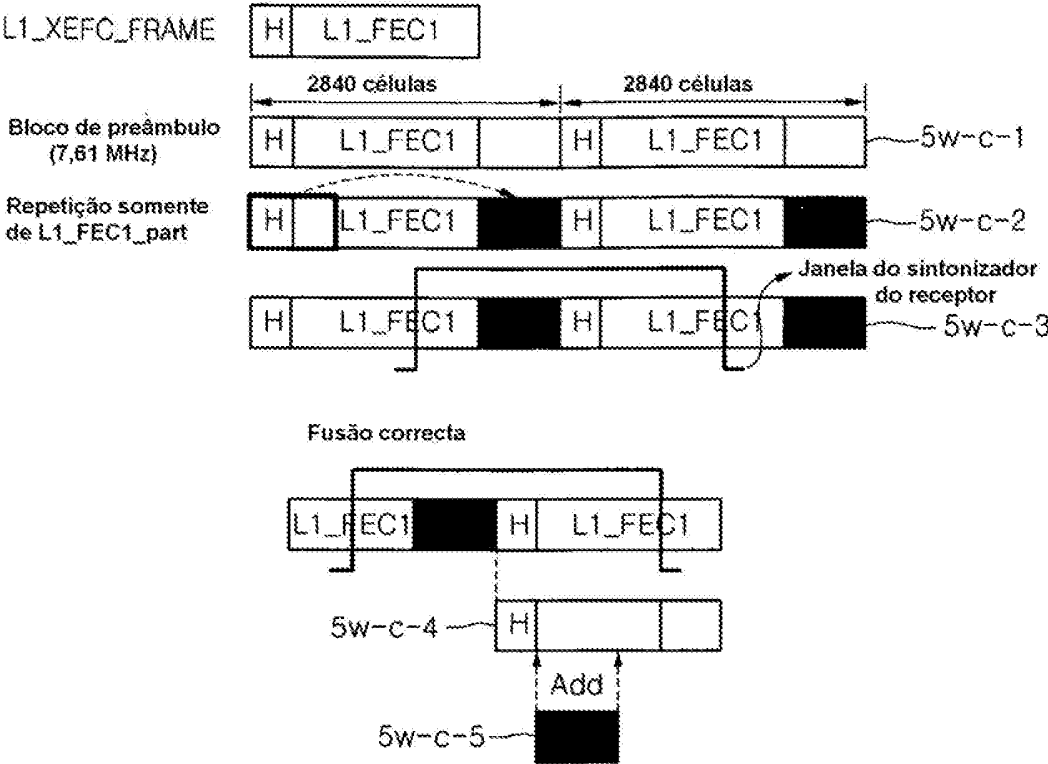


Fig. 122

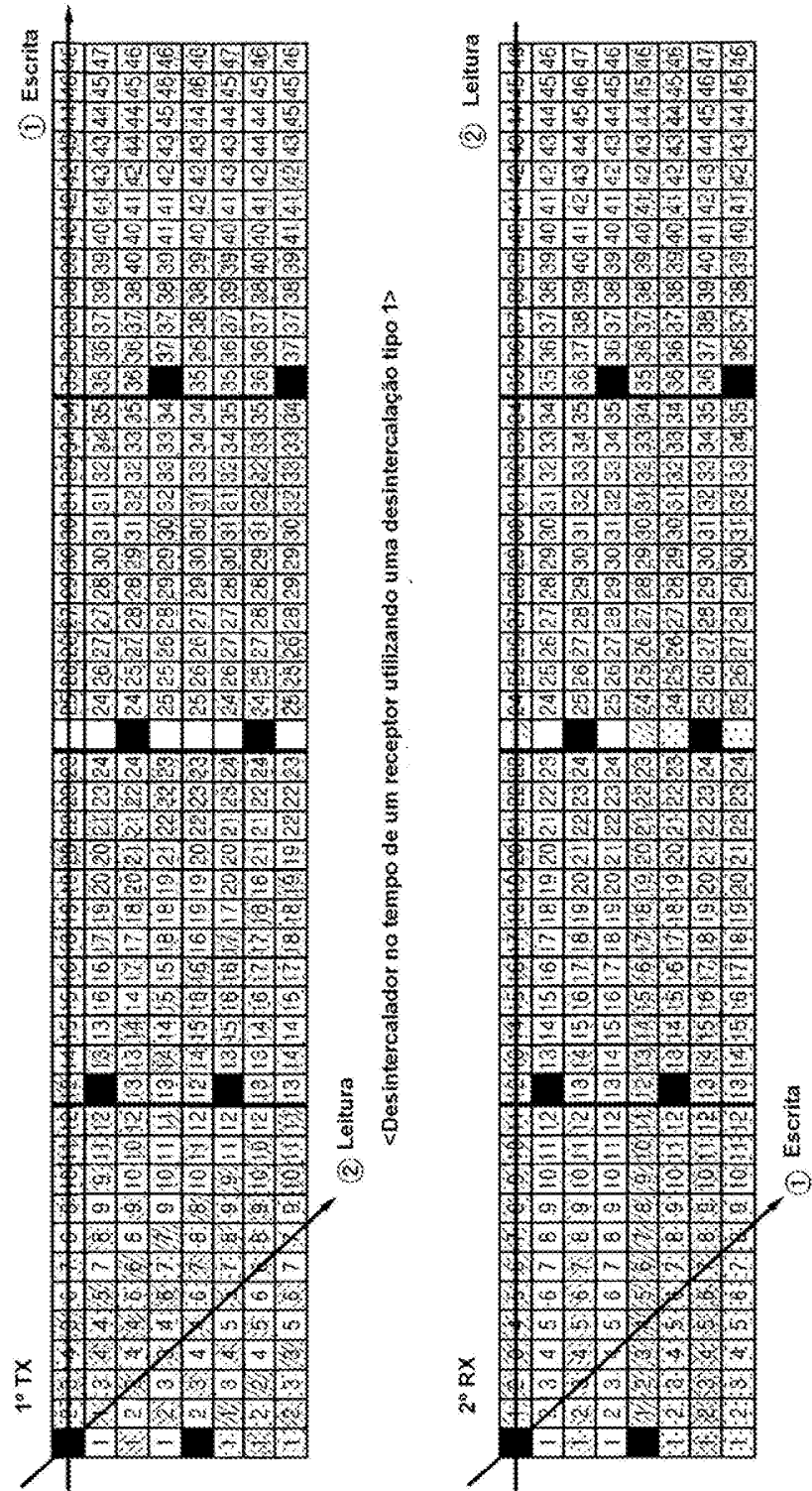


Fig. 123

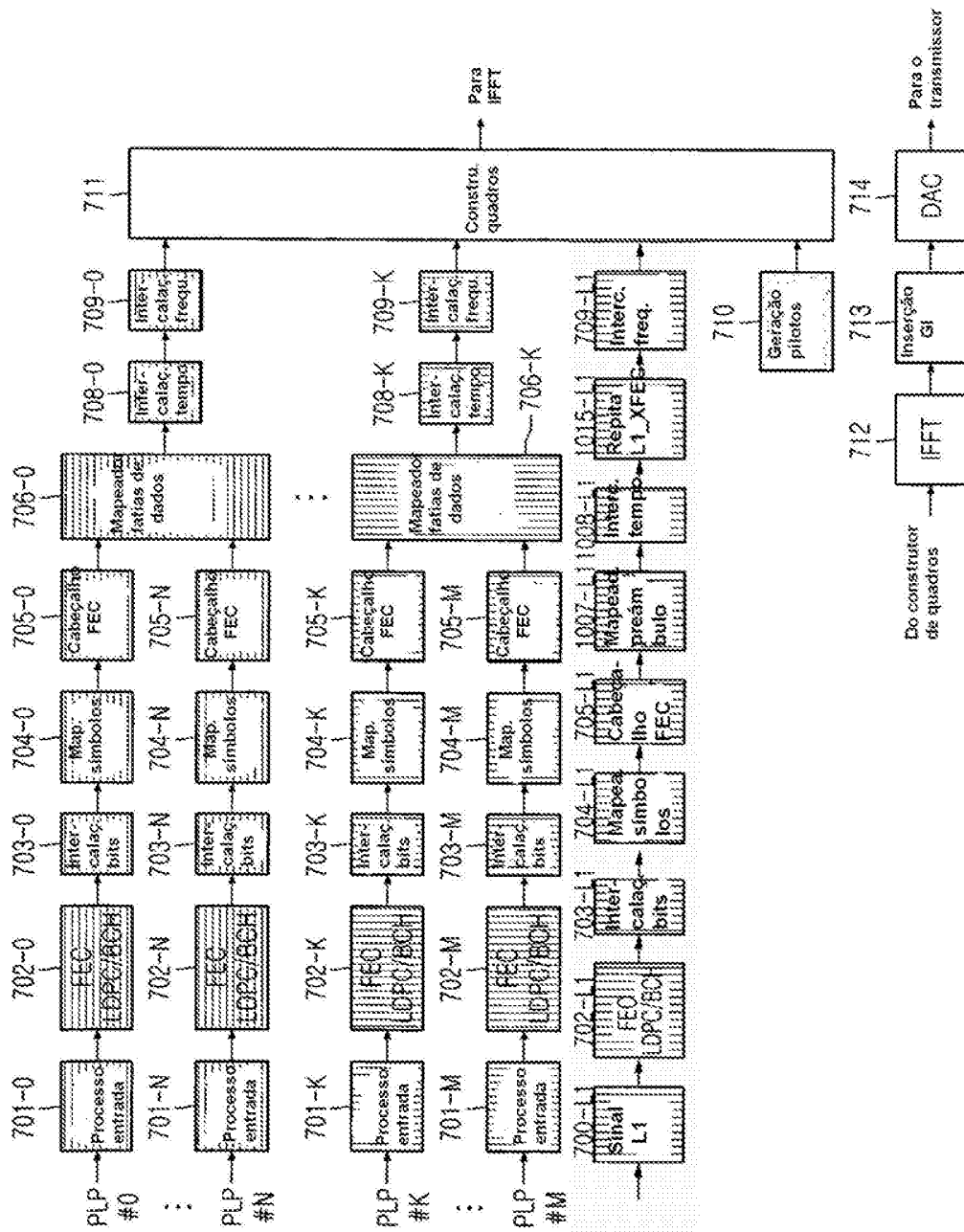


Fig. 124

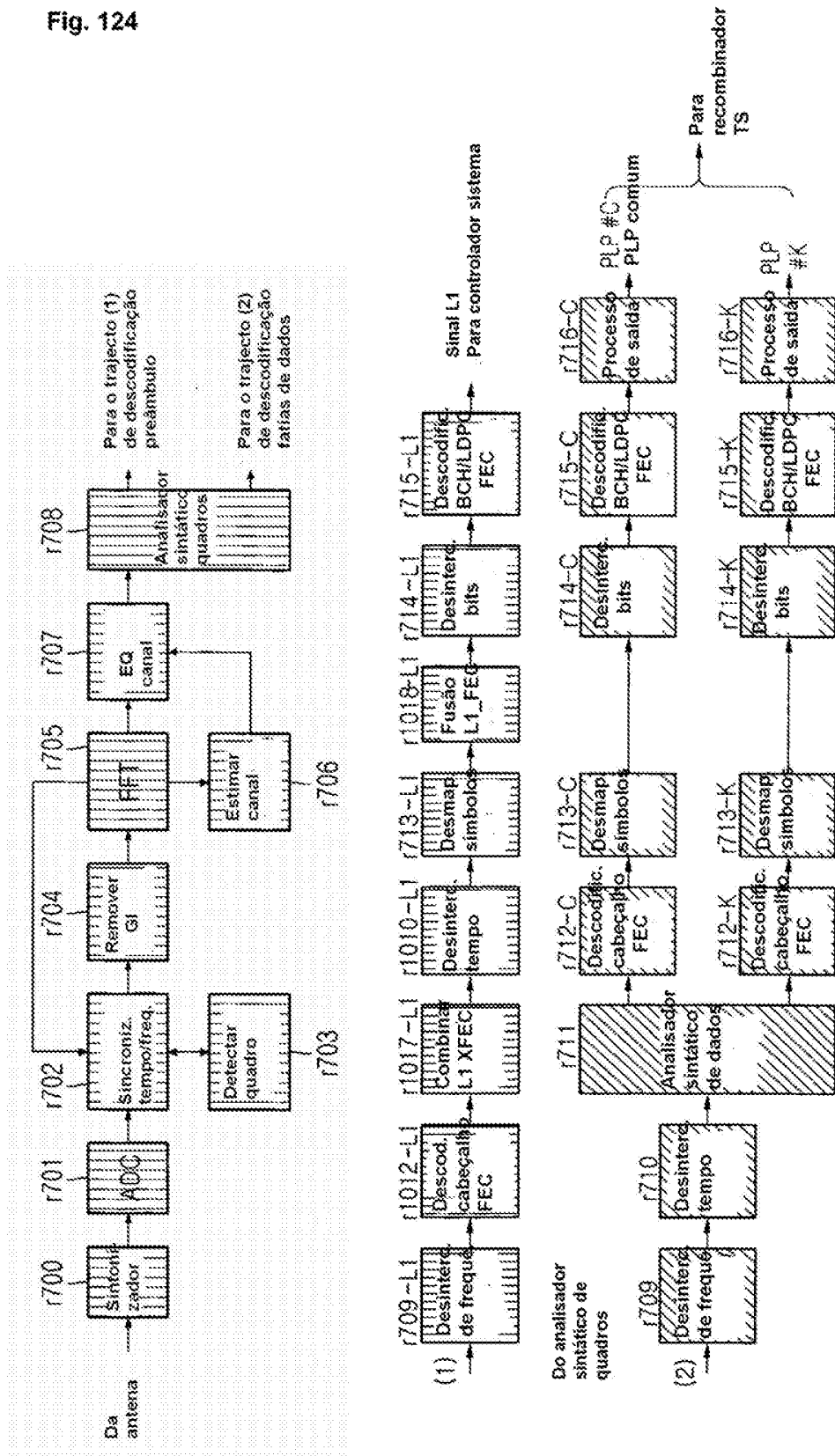
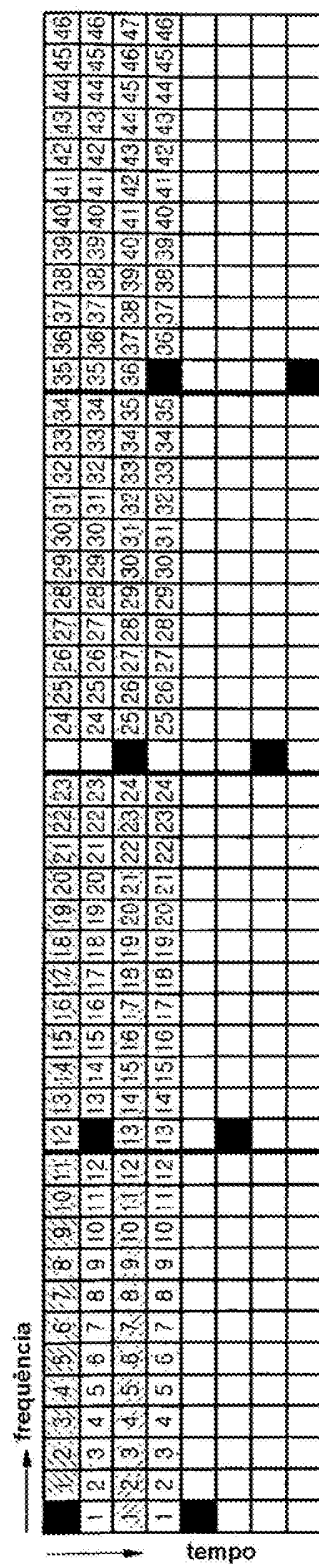
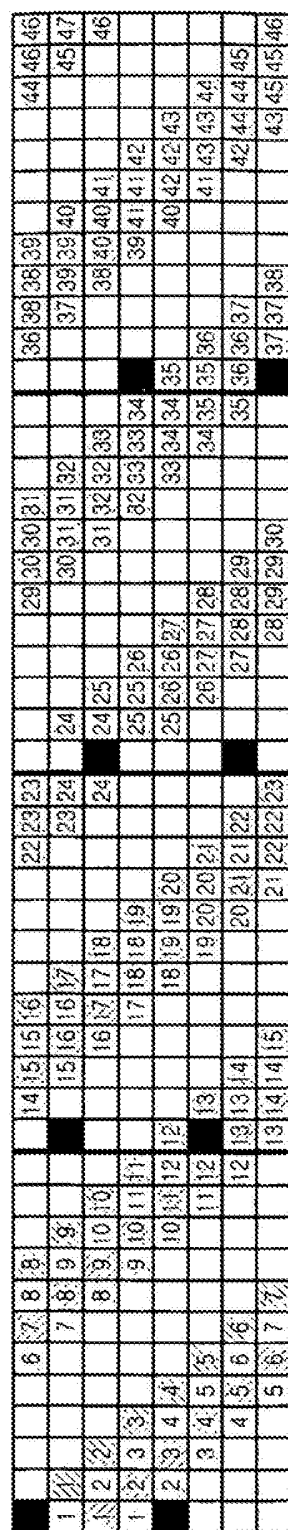


Fig. 125



Intercalação no tempo DESLIGADA



Intercalação no tempo LIGADA

Fig. 126

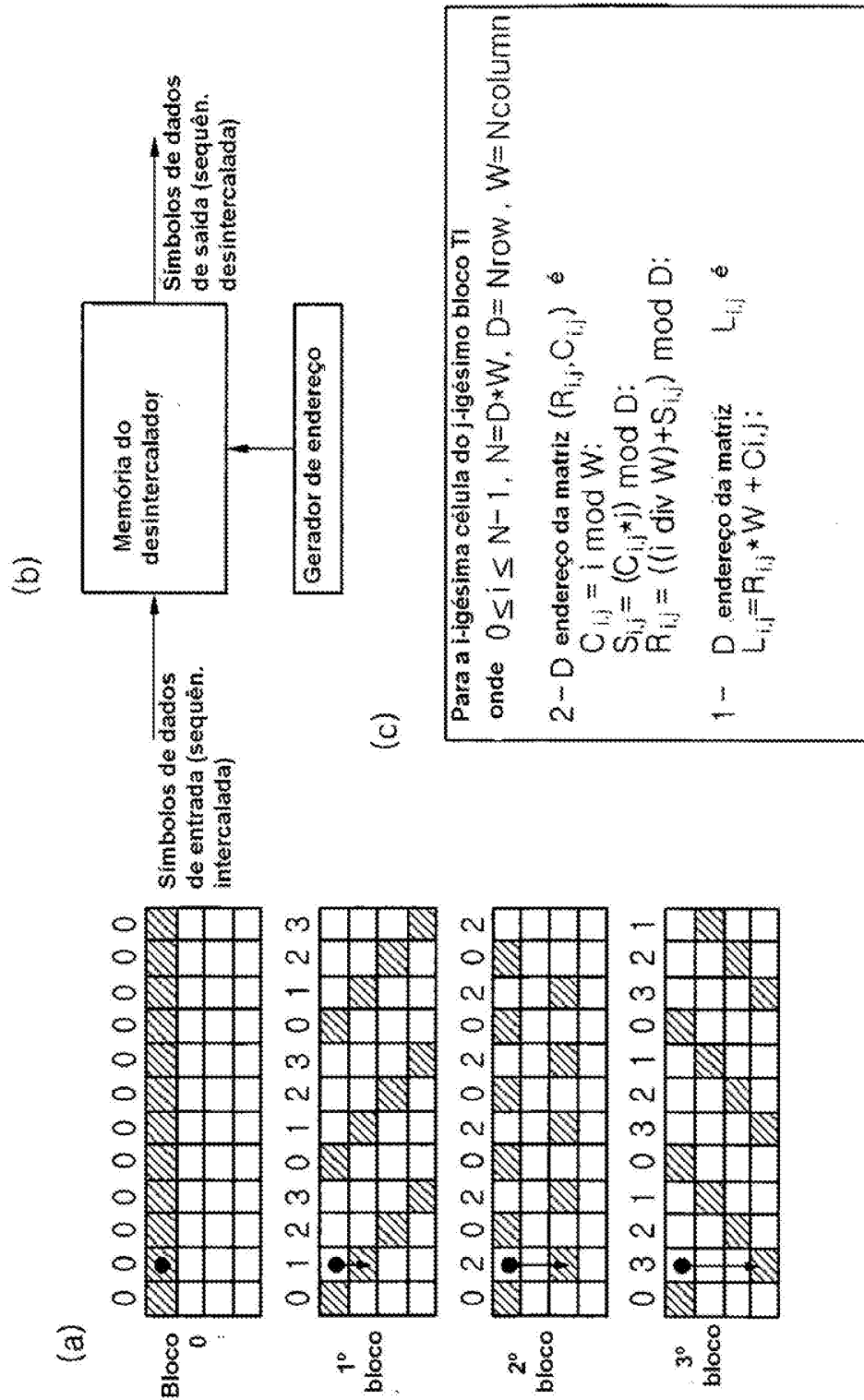


Fig. 127

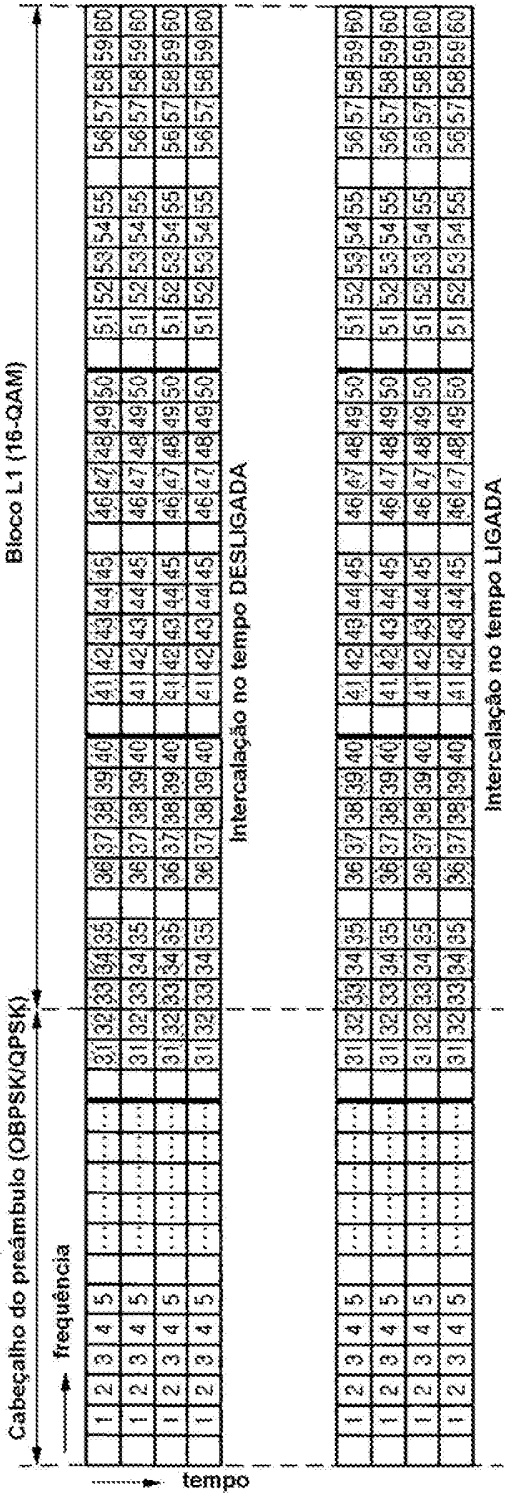


Fig. 128

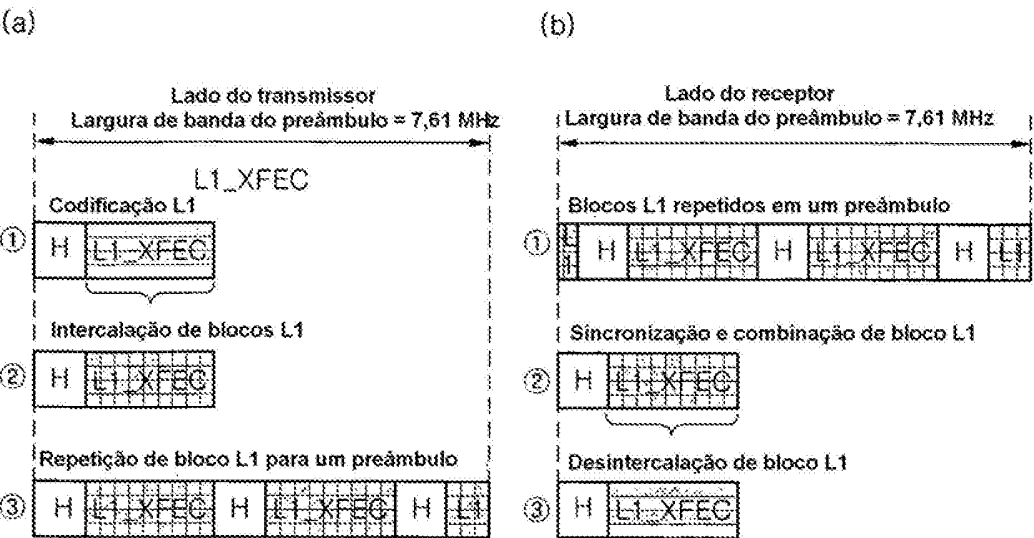


Fig. 129

Valor	Profundidade TI
00	Sem TI
01	Mais adequado
10	4 símbolos OFDM
11	8 símbolos OFDM

Quando TI_flag(1 bit) é utilizado

TI para bloco L1 curto

Para melhor desempenho TI

Fig. 130

(a) Sinalização e estrutura do cabeçalho

Nº	Características chave	Campos de sinalização	Preenchimento a zeros	Cabeçalho L1 por desempenho	Desempenho bloco L1 TI	Inform. complement. adicional
1	Somente marca TI lig./deslig.	L1 (15), TI (1), no CRC	Não	Normal	Mau para bloco L1 curto	Não
2	L1_size_bits / 8	L1 (12), TI (2), CRC (2)	Max 7 bits	Melhor	Bom	Não
3	L1_size_cells	L1 (13), TI (2), CRC (1)	Max 3 bits	Bom	Bom	Não
4	L1_size_bits por símbolo	L1 (13), TI (2), CRC (1)	Max 4 bits	Bom	Bom	Não
5	L1_size_cells por símbolo	L1 (11), TI (2), CRC (2), Rfu (1)	Max 19 bits	Melhor	Bom	Não
6	Dois cabeçalhos (32 bits)	L1 (15), TI (2), CRC (4), Rfu (11)	Não	Melhor	Bom	0,01% (negligenciável)

(b) Método de preenchimento a zeros

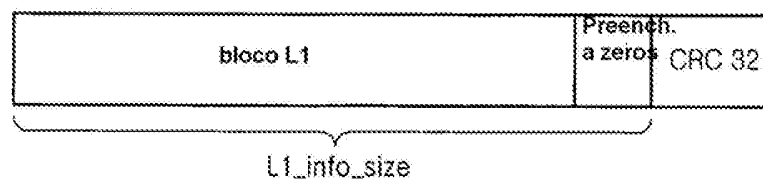


Fig. 131

Campo	Bits	Descrição
num_chbon	3	número de canais ligados
num_dslice	8	número de fatias de dados
num_notch	4	número de bandas de entalhe
for i=0..num_dslice-1 {		
dslice_id	8	identificador de fatia de dados
dslice_start	12	início da fatia de dados dentro do canal C2 (Max 64 MHz)
dslice_width	9	largura da fatia de dados (Max 7,61 MHz)
dslice_ti_depth	2	profundidade do intercalador no tempo
dslice_type	1	tipo de fatia de dados (tipo 1, tipo 2)*
dslice_pwr_allocation	2	atribuição de potência de fatia de dados
num_plp	8	número de PLPs
for i=0..num_plp-1 {		
plp_id	8	identificador PLP
plp_type	1	tipo de PLP (comum/dados)
PSI/SI reprocessing	1	marca de reprocessamento PSP/SI
plp_payload_type	5	tipo de carga útil PLP (TS, GS, ...)
if dslice_type=='0' {		
plp_modcod	5	modulação/codificação PLP
plp_start_addr	14	endereço de início PLP
}		
}		
}		
for i=0..num_notch-1{		
notch_start	12	início da banda de entalhe dentro do canal C2 (Max 64 MHz)
notch_width	9	largura da banda de entalhe (Max. 7,61 MHz)
}		
gi	2	modo de intervalo de guarda
num_data_symbols	10	número de dados símbolos OFDM por quadro C2
num_c2_frames	8	número de quadros C2 por superquadro
frame_idx	8	índice do quadro C2 dentro de um superquadro
PAPR	1	marca PAPR
reserved	0	reservado para utilização futura
crc32	32	CRC32

*tipo 1: PLP único com CCM

*tipo 2: Outros

Fig. 132

dslice_ti_depth	Profundidade TI
"00"	1
"01"	4
"10"	8
"11"	16

Fig. 133

dslice_type	tipo
"0"	PLP único com CCM
"1"	Outros

Fig. 134

plp_type	tipo
"0"	plp comum
"1"	plp de dados

Fig. 135

plp_payload_type	tipo de carga útil
"00000"	GFPS
"00001"	GCS
"00010"	GSE
"00011"	TS
others	reservado para utilização futura

Fig. 136

plp_modcod	Modulação	Taxa de código	Bloco LDPC
"00000"	16QAM	4/5	longo
"00001"			curto
"00010"		9/10	longo
"00011"		8/9	curto
"00100"	64QAM	2/3	longo
"00101"			curto
"00110"		4/5	longo
"00111"			curto
"01000"	256QAM	9/10	longo
"01001"		8/9	curto
"01010"		3/4	longo
"01011"			curto
"01100"	1024QAM	5/6	longo
"01101"			curto
"01110"		9/10	longo
"01111"		8/9	curto
"10000"	4096QAM	3/4	longo
"10001"			curto
"10010"		5/6	longo
"10011"			curto
"10100"		9/10	longo
"10101"		8/9	curto
"10110"		5/6	longo
"10111"			curto
"11000"		9/10	longo
"11001"		8/9	curto
outros	reservado para uso futuro	reservado para uso futuro	reservado para uso futuro

Fig. 137

gi	valor
"00"	1/128
"01"	1/64
outros	reservado para uso futuro

Fig. 138

PAPR	valor
"0"	desativar
"1"	ativar

Fig. 139

Campo	Bits	Descrição
network_id	16	ID da rede DVB-C2 actual
c2_system_id	16	ID do sistema C2 dentro da rede DVB-C2
C2_signal_start_frequency	32	Frequência do início do sinal C2
C2_signal_stop_frequency	32	Frequência da paragem do sinal C2
num_chbon	3	número de canais ligados
num_dslice	8	número de fatias de dados
num_notch	4	número de bandas de entalhe
for i=0..num_dslice-1 {		
dslice_id	8	identificador da fatia de dados
dslice_start	12	início de fatia de dados dentro do canal C2 (max. 64 MHz)
dslice_width	9	largura da fatia de dados (max. 7,61 MHz)
dslice_ti_depth	2	profundidade do intercalador no tempo
dslice_type	1	tipo de fatia de dados (tipo 1, tipo 2)*
dslice_pwr_allocation	2	atribuição da potência da fatia de dados
reserved_1	0	reservado para uso futuro
num_plp	8	número de PLPs
for i=0..num_plp-1 {		
plp_id	8	identificador PLP
plp_type	2	tipo de PLP (comum/dados normais/dados agrupados)
if plp_type=='common' or 'grouped' {		
plp_group_id	8	identificador do grupo PLP
}		
plp_payload_type	5	tipo de carga útil PLP (TS, GS, ...)
PSI/SI reprocessing	1	marca de reprocessamento PSP/SI
reserved_2	0	reservado para uso futuro
if dslice_type=='0' {		
plp_modcod	5	modulação/codificação PLP
plp_start_addr	14	endereço de início PLP
}		
}		
}		
for i=0..num_notch-1 {		
notch_start	12	início da banda de entalhe dentro do canal C2 (max. 64 MHz)
notch_width	9	largura da banda de entalhe (max. 7,61 MHz)
}		
qi	2	modo de intervalo de guarda
num_data_symbols	10	número de dados de símbolos OFDM por quadro C2
num_c2_frames	8	número de quadros C2 por superquadro
frame_idx	8	índice de quadro C2 dentro de um super quadro
PAPR	1	marca PAPR
reserved_3	0	reservado para uso futuro
crc32	32	CRC32

*tipo 1: PLP único com CCM

*tipo 2: Outros

Fig. 140

plp_type	tipo
"00"	plp comum
"01"	plp de dados normais
"10"	plp de dados agrupados
outros	reservado para uso futuro

Fig. 141

Campo	Bits	Descrição
network_id	16	ID da rede DVB-C2 actual
c2_system_id	16	ID do sistema C2 dentro da rede DVB-C2
C2_signal_start_frequency	32	frequência do início do sinal C2
num_chbon	3	número de canais ligados
num_dslice	8	número de fatias de dados
num_notch	4	número de bandas de entalhe
for i=0..num_dslice-1 {		
dslice_id	8	identificador de fatia de dados
dslice_start	12	início da fatia de dados dentro do canal C2 (max. 64 M
dslice_width	12	largura da fatia de dados (Max. 64 MHz)
dslice_ti_depth	2	profundidade do intercalador no tempo
dslice_type	1	tipo de fatia de dados (tipo 1, tipo 2)*
num_plp	8	número de PLPs
reserved_1	0	reservado para uso futuro
for i=0..num_plp-1 {		
plp_id	8	identificador PLP
plp_type	2	tipo PLP (comum/dados normais/dados agrupados)
if plp_type=='common' or 'grouped' {		
plp_group_id	8	identificador de grupo PLP
}		
PSP/SI reprocessing	1	marca de reproprocessamento PSP/SI
plp_payload_type	5	tipo de carga útil PLP (TS, GS, ...)
reserved_2	0	reservado para uso futuro
if dslice_type=='0' {		
plp_modcod	5	modulação/codificação PLP
plp_start_addr	14	endereço de início PLP
}		
}		
}		
for i=0..num_notch-1 {		
notch_start	12	início da banda de entalhe dentro do canal C2 (max 64
notch_width	9	largura da banda de entalhe (max. 7,61 MHz)
}		
gi	2	modo de intervalo de guarda
PAPR	1	marca PAPR
reserved_3	0	reservado para uso futuro
crc32	32	CRC32
Total	221	

*tipo 1: PLP único com CCM

*tipo 2: Outros

Fig. 142

