(19) INSTITUTO NACIONAL DA PROPRIEDADE INDUSTRIAL
PORTUGAL

(11) *Número de Publicação:* **PT 745307 E**

(51) *Classificação Internacional:* (Ed. 6)
H04N007/025 A H04N005/445 B

(12) *FASCÍCULO DE PATENTE DE INVENÇÃO*

(22) *Data de depósito:* 1995.12.12

(30) *Prioridade:* 1994.12.14 EP 94203642

(43) *Data de publicação do pedido:*
1996.12.04

(45) *Data e BPI da concessão:*
1999.12.29

(73) *Titular(es):*
KONINKLIJKE PHILIPS ELECTRONICS N.V.
GROENEWOUDSEWEG 1, 5621 BA EINDHOVEN, HOLANDA NL

PHILIPS NORDEN AB
KOTTBYGATAN 7, KISTA S-164 85 STOCKHOLM SE

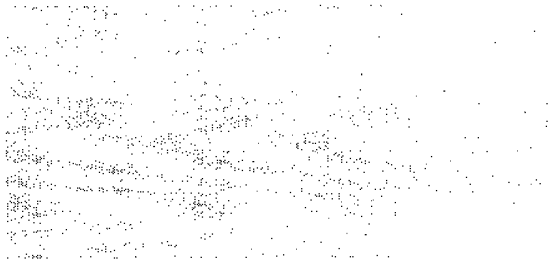
(72) *Inventor(es):*

RICHARD CEES SPIERO NL
JAN VAN DER MEER NL
ARMAND VICTOR WEMELSFELDER NL
WIEBE DE HAAN NL

(74) *Mandatário(s):*
MANUEL GOMES MONIZ PEREIRA
RUA DO ARCO DA CONCEIÇÃO 3, 1º AND. 1100 LISBOA PT

(54) *Epígrafe:* SISTEMA DE TRANSMISSÃO DE SUBSTITULAÇÃO

(57) *Resumo:*



Nº 745307

RESUMO

SISTEMA DE TRANSMISSÃO DE SUBTITULAÇÃO

Descreve-se um método para transmitir simultaneamente um sinal de vídeo e dados codificados que representam imagens gráficas. A invenção é particularmente aplicável para transmitir subtítulos em várias línguas com um programa de vídeo. As imagens gráficas são regiões rectangulares dentro da área activa do vídeo. São transmitidas sob a forma de aplicações de bits. A invenção permite não só a exibição de qualquer tipo ou tamanho de caracteres como também a definição de, por exemplo, um logotipo do fornecedor do programa. Os dados codificados incluem um selo de tempo para especificar o momento em que um subtítulo deve ser exibido. Formas de realização preferidas da invenção incluem a transmissão de dados de uma tabela de consulta de cores (colour-look-up-table – CLUT) e de um código de compatibilidade que indica um número mínimo exigido de entradas da referida tabela de consulta de cores. Para receptores com uma CLUT com mais entradas que as necessárias, é transmitida uma tabela de aplicação para aplicar a largura dos dados dos pixels à largura de entrada da CLUT em causa. O método é aplicável a sistemas de Difusão Directa de Vídeo (Direct Video Broadcast) em que o sinal de vídeo é codificado em MPEG2 e as imagens gráficas são instaladas num fluxo privado de dados de um Fluxo de Transporte (Transport Stream) MPEG2.

DESCRIÇÃO

SISTEMA DE TRANSMISSÃO DE SUBTITULAÇÃO

ÂMBITO DA INVENÇÃO

A invenção abrange um método para transmitir dados codificados que definem uma imagem gráfica sob a forma de uma região rectangular dentro de umaa área activa do vídeo. A invenção abrange também um método para receber as referidas imagens gráficas, e um transmissor e receptor preparados para executar os referidos métodos.

ANTECEDENTES DA INVENÇÃO

Um método de transmissão de imagens gráficas da técnica anterior é geralmente conhecido pelo nome de «teletexto». O padrão do Sistema Mundial de Teletexto (World System Teletext) foi adoptado em muitos países para a transmissão simultânea de informações textuais e gráficas com um sinal analógico convencional de televisão. O sistema de teletexto é particularmente usado para a transmissão de subtítulos associados a um programa de televisão.

Entretanto, o padrão digital de televisão MPEG2 (ISO/IEC 13818) será aplicado à escala mundial para a difusão de programas digitais de televisão ao utilizador final por redes de satélite, por cabo e terrestres e por meios apresentados em pacotes de programas, como por exemplo fita ou disco. Existe no mercado uma enorme variedade de circuitos integrados para a descompressão e desmultiplexagem de MPEG2, que serão aplicados em descodificadores de receptores com o objectivo de receber e descodificar programas difundidos pela televisão ou de descodificar multi(média) apresentados em conjuntos de programas.

A tecnologia de compressão e multiplexagem MPEG2 proporcionará ao utilizador final uma gama muito vasta de programas (normalmente na ordem das centenas) que exigirão uma «interface homem-máquina» fácil de utilizar. Espera-se que o utilizador final associe «digital» a qualidade elevada. A qualidade elevada tem portanto de ser manifesta em todos os aspectos do serviço que será fornecido. É característica da difusão por satélite a grande pegada da sua área de recepção, que abrange normalmente áreas de linguagens

múltiplas. Portanto, muitos dos programas de televisão a serem difundidos serão também traduzidos em áreas de linguagens múltiplas, o que requer capacidades de subtitulação em várias línguas. A atracção e a capacidade de leitura de subtítulos terá um impacto destacado no desejo do utilizador final de ver um programa de televisão numa linguagem falada numa língua diferente da sua.

O sistema de teletexto da técnica anterior é muito limitado na qualidade e no desempenho. Trata-se de um sistema de transmissão baseado em caracteres, em que são transmitidos dados codificados para definir quais os caracteres que serão exibidos. O aparecimento dos subtítulos (em termos de tipo dos caracteres, espacejamento, possibilidades gráficas, cores) é fixado no hardware. Além disso, o presente sistema de teletexto não consegue suportar conjuntos complexos de caracteres como os que são usados na República Popular da China e no Japão. O sistema com base em caracteres exigiria geradores de caracteres muito complexos e dispendiosos no caso de suportar conjuntos múltiplos de caracteres ou no caso de um conjunto de caracteres único mas muito complexo. Assim, o sistema de teletexto da técnica anterior apenas proporciona uma solução básica para a subtitulação em várias línguas.

A Patente norte-americana US-A-5.089.899 correspondente ao preâmbulo da reivindicação 1 descreve um leitor de discos em que é gravado um sinal de vídeo e imagens gráficas que podem ser exibidas simultaneamente com o sinal de vídeo gravado. A imagem gráfica tem um tamanho fixo e compreende uma grelha de tipos de 16*48 cada um deles com pixels de 6*12. A imagem é codificada por instruções que especificam os valores dos pixels para cada posição dos tipos. Deste modo, podem definir-se pixels individuais. Por defeito, a imagem é exibida no meio do ecrã mas o utilizador pode deslocar a posição da imagem para outras posições do ecrã. Podem ser pré-gravadas até 16 imagens. O utilizador pode seleccionar uma delas por playback.

O Pedido de Registo de Patente europeia EP-A 0 488 676 descreve um formato de armazenagem para guardar uma apresentação de multimédia num CD-ROM. Os dados guardados no disco incluem dados de aplicação que definem uma imagem e dados de items que definem items do menu a serem exibidos simultaneamente com a referida aplicação. A cada item do menu estão associados dados da posição de exibição que indicam a sua posição de exibição no ecrã. Os items do menu são items de texto. A publicação é omissa acerca da maneira como o referido texto é codificado.

OBJECTIVO E SUMÁRIO DA INVENÇÃO

É um objectivo da invenção ultrapassar, de modo a aperfeiçoá-lo, o sistema de transmissão da técnica anterior. É outro objectivo proporcionar uma solução universal para subtitulação em várias línguas, em particular para um ambiente MPEG2.

Com este objectivo, a invenção proporciona um método para transmitir dados codificados que define uma imagem gráfica sob a forma de uma região rectangular para exibição dentro de uma região de um sinal activo de vídeo, sendo os pixels que formam a referida região individualmente definidos pelos dados codificados, caracterizado por os dados codificados incluírem o tamanho e a posição da referida região e um selo de tempo que representa o momento em que a referida região deve ser exibida.

A invenção proporciona também um método correspondente para receber dados codificados que definem uma imagem gráfica, bem como o transmissor, o receptor e o sinal de imagem correspondentes.

O método possibilita aos fornecedores do programa controlarem o aspecto das imagens gráficas segundo os pixels. O sistema «Gráficos com base na Região» («Region based Graphics») proporciona regiões com a aplicação de bits em vez de regiões codificadas por caracteres sobrepostas ao sinal de vídeo. Não existem restrições no que se refere ao tamanho dos caracteres e ao tipo dos caracteres e existe a possibilidade de sobrepor outros gráficos para além do texto, por exemplo o logotipo do fornecedor do programa. A transmissão de um selo de tempo representando o momento em que a referida região será exibida permite que sejam antecipadamente transmitidos títulos em várias línguas e que sejam exibidos em rigoroso sincronismo com um sinal de vídeo. Nos sistemas da técnica anterior, os subtítulos são exibidos logo que são recebido. Só é possível uma sincronização mais ou menos rigorosa se todos os subtítulos em todas as línguas forem transmitidos durante um período limitado de tempo.

Os dados codificados subsequentes para uma determinada região podem incluir um código de visibilidade para desactivar a exibição da referida região. Isto permite apagar efectivamente uma região gráfica.

Noutras formas de realização da invenção, os pixels de uma região são entradas codificadas de uma tabela de consulta de cores. São instaladas tabelas de aplicação para aplicar códigos de n pixels a uma tabela de consulta de cores com mais de 2n entradas.

Numa forma de realização particularmente vantajosa, os dados codificados são instalados num fluxo de transporte privado de acordo com o padrão MPEG2 ou similar. A arrumação dos dados codificados em pacotes PES de um fluxo de transporte privado MPEG2 tem a vantagem de permitir uma fácil sincronização das aplicações de bits a um fluxo de vídeo MPEG associado.

BREVE DESCRIÇÃO DAS FIGURAS

As Figs. 1-3 mostram exemplos de um ecrã de exibição que compreende uma ou mais regiões de imagens gráficas.

A Fig. 4 mostra uma região com os seus parâmetros de tamanho e posição relevantes.

A Fig. 5 mostra um diagrama esquemático de um transmissor preparado para transmitir simultaneamente um sinal de vídeo e imagens gráficas de acordo com a invenção.

A Fig. 6 mostra um diagrama esquemático de um receptor que recebe simultaneamente um sinal de vídeo e imagens gráficas de acordo com a invenção.

As Figs. 7 e 8 mostram formas de realização de um circuito gerador de cor do receptor apresentado na Fig. 6.

DESCRIÇÃO DE FORMAS DE REALIZAÇÃO

A invenção será agora descrita em primeiro lugar em termos do formato de transmissão. A sintaxe tal como é descrita abaixo é semelhante à sintaxe utilizada no padrão MPEG2 (ISO/IEC 13818). Os símbolos {and} desta sintaxe foram omitidos dado que não são relevantes para a legibilidade. Foram também omitidos os elementos de dados que não são essenciais para a invenção ou que apenas são úteis para o alinhamento de bytes.

Os dados codificados que definem as imagens gráficas são apresentados em pacotes PES (Fluxos elementares em Pacotes – Packetized elementary Streams) e instalados num Fluxo de Transporte (Transport Stream) MPEG2. Tal como determinado pelo padrão MPEG2, os pacotes PES compreendem um registo de cabeçalho e uma carga útil.

O registo de cabeçalho compreende uma pluralidade de elementos de dados, sendo os seguintes relevantes para a invenção:

- O elemento *stream_id* é colocado em "1011 1101" de modo a indicar que a imagem gráfica e transmitida como *private_data_stream_1*.
- O elemento *PES_packet_length* é colocado num valor tal que cada pacote PES é alinhado com um pacote de transporte MPEG2.
- O elemento *data_alignment_indicator* é colocado em "1", indicando que os campos dos dados da região são alinhados com os pacotes PES.
- O *PTS* (Selo de Apresentação do Tempo – Presentation Time Stamp) indica o início do tempo de apresentação de cada conjunto de dados (relacionado com a região) contidos no pacote PES. Os PTS's das regiões dentro de uma multipágina hão-de diferir mais de 40 msec.

A carga útil do pacote PES que define as imagens gráficas é identificada como *PES_data_field()*. O quadro 1 mostra a sintaxe deste campo.

Quadro 1

PES_data_field()	
data_identifier	8 uimsbf
RbG_stream_id	16 bslbf
stream_level_of_compatibility	3 bslbf
single_page_display_flag	1 bslbf
para (i = 0; N;i + +)	
region_id	8 uimsbf
region_level_of_compatibility	3 bslbf
region_data_length	16 uimsbf
region_data_field()	

Aplica-se a seguinte definição semântica aos elementos dos dados apresentados neste Quadro:

- *data_identifier* – Os dados dos Gráficos com base na Região (RbG – Region based Graphics) são identificados pelos valores 0x20-0x2F do campo *data_identifier*.
- *RbG_stream_id* – Identifica o fluxo dos Gráficos com base da Região a partir do qual os dados são guardados neste pacote PES.
- *stream_level_of_compatibility* – Indica a Tabela de Consulta de Cores (Colour-Look-Up-Table – CLUT) mínima necessária para descodificar regiões deste fluxo. O valor 0x01 refere-se a uma CLUT de 2 bits, o valor 0x02 refere-se a uma CLUT de 4 bits e o valor 0x03 refere-se a uma CLUT de 8 bits.
- *single_page_display_flag* – Se for colocado em "1", indica que este fluxo RbG contém apenas uma página para exibição.
- *region_id* – identifica a região a partir da qual os dados são contidos no *region_data_field()* imediatamente a seguir.
- *region_level_of_compatibility* – indica a CLUT mínima que é necessária para descodificar esta região. São válidos os valores acima definidos.
- *region_data_length* – Especifica o número de bytes no *region_data_field()* imediatamente subsequente, cuja sintaxe é apresentada no Quadro II.

O Quadro II apresenta a sintaxe do *region_data_field()*.

A definição semântica que se segue aplica-se aos elementos dos dados apresentados neste Quadro:

- *region_data_type* – identifica o tipo de dados da região. O valor 0x10 refere-se a uma intra-região (região I – I-region), o valor 0x11 refere-se aos atributos da região (região A – A-region), o valor 0xFF indica uma região que será apagada.

Quadro II

region_data_field()	
region_data_type	4 bslbf
visibility_flag	1 bslbf
horizontal_address_flag	1 bslbf
vertical_address_flag	1 bslbf
number_of_pixels_flag	1 bslbf
number_of_lines_flag	1 bslbf
se (horizontal_address_flag = = "1")	
region_horizontal_address	16 uimsbf
se (vertical_address-flag = = "1")	
region_vertical_address	16 uimsbf
se (number_of_pixels_flag = = "1")	
number_of_pixels	16 uimsbf
se (number_of_lines_flag = = "1")	
number_of_lines	16 uimsbf
number_of_subregions	4 uimsbf
para (i = 0;i<number_of_sub_regions;i + +)	
sub_region_id	4 bslbf
number_of_sub_region_lines_flag	1 bslbf
clut_flag	1 bslbf
attribute_extension_flag	1 bslbf
coded_data_flag	1 bslbf
coded_data_extension_flag	1 bslbf
se (number_of_sub_region_lines_flag = = "1")	
number_of_sub_region_lines	16 uimsbf
se (clut_flag = = "1")	
colour_look_up_table_data()	
se (attribute_extension_flag = = "1")	
attribute_extension_length	8 uimsbf
para (i = 1;i<attribute_extension_length;i + +)	
reservado	8 bslbf
se (coded_data_flag = = "1")	
coded_sub_region_data()	
se (coded_data_extension_flag = = "1")	
coded_data_extension_length	8 uimsbf
para (i=0;i<coded_data_extension_length;i + +)	
reservado	8 bslbf

- *visibility_flag* – Se for colocado em "1", especifica que esta região será exibida tal como definida pelos dados codificados para esta região. Se for colocado em "0", esta região será transparente.
- *horizontal_address_flag* – Indica a presença do campo *region_horizontal_address*. Nas regiões I, este sinalizador será "1".
- *vertical_address-flag* – Indica a presença do campo *region_vertical_address*. Nas regiões I, este sinalizador será "1".
- *number_of_pixels_flag* – Indica a presença do campo *number_of_pixels*. Nas regiões I, este sinalizador será "1".
- *number_of_lines_flag* – Indica a presença do campo *number_of_lines*. Nas regiões I, este sinalizador será "1".
- *region_horizontal_address* – Especifica o endereço horizontal do pixel superior esquerdo desta região. Na configuração de 720*576 pixels, só os pixels pares serão endereçados. O pixel mais à esquerda dos 720 pixels activos tem índice zero e o endereço dos pixels aumenta da esquerda para a direita. O endereço horizontal permitirá que cada pixel da região se situe entre os 720 pixels activos.
- *region_vertical_address* – Especifica o endereço vertical da linha de cima desta região. A linha de cima da configuração de 720*576 é a linha zero e o índice das linhas aumenta uma dentro da configuração de cima para baixo. O endereço vertical permitirá que cada linha da região se situe entre as 576 linhas activas.
- *number_of_pixels* – Especifica o número de pixels horizontais desta região. É interdito um valor zero.
- *number_of_lines* – Especifica o número de linhas verticais desta região. É interdito um valor zero.
- *number_of_sub_regions* – Especifica o número de sub-regiões dentro desta região. É interdito um valor zero.
- *sub_region_id* – identifica a sub-região.
- *number_of_sub_region_lines_flag* – Indica a presença do campo *number_of_sub_region_lines*.
- *clut_flag* – Indica a presença do campo *colour_look_up_data_field()*, cuja sintaxe é apresentada no Quadro III.
- *attribute_extension_flag* – Se for colocado em "1", indica a presença do campo *attribute_extension_length* e dos campos reservados a ele associados.
- *coded_data_flag* – Indica a presença do campo *coded_sub_region_data()* que é definido no Quadro IV.

- *coded_data_extension_flag* – Se for colocado em "1", identifica a presença do campo *coded_data_extension_length* e os campos reservados a ele associados.
- *number_of_sub_region_lines* – Especifica o número de linhas de uma sub-região. O número total de linhas de todas as sub-regiões incluídas na região será igual ao número de linhas da região.
- *attribute_extension_length* – Especifica o número de bytes reservados imediatamente a seguir.
- *coded_data_extension_length* – Especifica o número de bytes reservados imediatamente a seguir.

O Quadro III apresenta a sintaxe do *colour_look_up_data_field()*.

Quadro III

Colour_look_up_table_data	
Number_of_entries	9 uimsbf
Para (i = 1;i<number_of_entries;i + +)	
2_bit_clut_destination_flag	1 bslbf
4_bit_clut_destination_flag	1 bslbf
8_bit_clut_destination_flag	1 bslbf
full_range_flag	1 bslbf
entry_value	8 uimsbf
se (full_range_flag = = "1")	
Y_value	8 bslbf
U_value	8 bslbf
V_value	8 bslbf
T_value	8 bslbf
Nos outros casos	
Y_value	4 bslbf
U_value	4 bslbf
V_value	4 bslbf
T_value	4 bslbf

Aos elementos dos dados apresentados neste Quadro aplica-se a seguinte definição semântica:

- *number_of_clute_updates* – Especifica o número de ciclos de actualização a fazer.
- *number_of_entries* – Especifica o número de entradas da CLUT que são actualizadas.
- *2_bit_clut_destination_flag* – Se for colocado em "1", indica que esta entrada se destina a uma CLUT de 2 bits.
- *4_bit_clut_destination_flag* – Se for colocado em "1", indica que esta entrada se destina a uma CLUT de 4 bits.
- *8_bit_clut_destination_flag* – Se for colocado em "1", indica que esta entrada se destina a uma CLUT de 8 bits.
- *full_range_flag* – Se for colocado em "1", indica que os campos do valor Y, do valor U, do valor V e do valor T têm uma resolução total de 8 bits. Se for colocado em "0", nesse caso estes campos contêm apenas os 4 bits mais significativos.
- *entry_value* – Especifica o valor de entrada da CLUT. O valor de entrada da primeira entrada da CLUT é zero.
- *Y_value* – O valor de saída da luminância da CLUT para esta entrada.
- *U_value* – O primeiro valor de saída da crominância da CLUT para esta entrada.
- *V_value* – O segundo valor de saída da crominância da CLUT para esta entrada.
- *T_value* – O valor de saída da transparência da CLUT para esta entrada. Um valor zero identifica a ausência de transparência. Um valor de 255 identifica uma transparência total. Para todos os outros valores, o nível de transparência é definido por interpolação linear.

O Quadro IV mostra a sintaxe dos *coded_sub_region_data()*.

Aos elementos dos dados apresentados neste Quadro aplica-se a seguinte definição semântica:

- *data_block_length* – Especifica o número de bytes do campo *coded_sub_region_data* imediatamente a seguir a este campo.
- *coding_mode* – Especifica o modo de codificação aplicado. Se for colocado em "00", identificando a presença do *mode_extension_field.*
- *pixels_in_mode* – Especifica o número de pixels codificados por um código de 2, 4 ou 8 bits de acordo com a especificação do campo *coding_mode*.

- *2_bit_code* – Um código de dois bits que especifica a cor de um pixel.
- *2_stuff_bits* – Dois bits de enchimento que serão codificados com "11".
- *4_bit_code* – Um código de quatro bits que especifica a cor de um pixel.
- *4_stuff_bits* – Quatro bits de enchimento que serão codificados com "1111".
- *8_bit_code* – Um código de oito bits que especifica a cor de um pixel.
- *mode_extension* – Para identificar o modo de extensão aplicado. Se for colocado em "00", identifica a presença de um comando. Se for colocado noutro valor, identifica que se aplica codificação de passagem de extensão.
- *10_bit-run* – Para especificar o número de pixels subsequentes com a cor identificada pelo campo *2_bit_code* imediatamente a seguir.
- *8_bit_run* – Para especificar o número de pixels subsequentes com a cor identificada pelo campo *4_bit_code* imediatamente a seguir.
- *4_bit_run* – Para especificar o número de pixels subsequentes com a cor identificada pelo campo *8_bit_code* imediatamente a seguir.
- *command_length* – Especifica o número de bytes deste campo *coded_sub_region_data_field()* imediatamente a seguir ao campo *command_length*. Se for colocado num valor de 15, este número não é especificado pelo campo *command_length*.
- *long_length* – Especifica o número de bytes deste campo *coded_sub_region_data_field()* imediatamente a seguir ao campo *long_length*.

Quadro IV

```
Coded_sub_region_data()
  data_block_length                                                    16 uimsbf
  para (i = 1; i<data_block_length;i + + )
    coding_mode                                                         2 bslbf
    se (coding_mode = "00")
      pixels_in_mode                                                    6 bslbf
     se (coding_mode = = "01")
        para (i = 1 ;i<(pixels_in_mode + 3)/4 ;i + + )
          para (i = 1 ;i<pixels_in_mode ;i + + )
            2_bit_code                                                  2 bslbf
          para i + 1;i<(4*((pixels_in_mode + 3)/4)-pixels_in_mode);i + + )
            2_stuff_bits                                                2 bslbf
      se (coding_mode = = "10")
        para (i = 1 ;i<(pixels_in_mode + 1)/2 ;i + + )
          para (i = 1;i<pixels_in_mode;i + +)
            4_bit_code                                                  4 bslbf
          para (i = 1;i<(2*((pixels_in_mode + 1)/2-pixels_in_mode);i + + )
            4_stuff_bits                                                4 bslbf
        se (coding_mode = = "11")
          para (i = 1;i<pixels_in_mode;i + + )
            8_bit_code                                                  8 bslbf

nos outros casos
        mode_extension                                                  2 bslbf
        se (mode_extension = = "00")
          se (mode_extension = = "01")
            10_bit_run                                                 10 uimsbf
            2_bit_code                                                  2 bslbf
        se (mode_extension = = "10")
          8_bit_run                                                     8 uimsbf
          4_bit_code                                                    4 bslbf
        se (mode_extension = = "11")
          4_bit_run                                                     4 uimsbf
          8_bit_code                                                    8 bslbf

nos outros casos
        command_length                                                  4 uimsbf
        se (command_length = = 15)
          long_length                                                  16 uimsbf
        command_id                                                      8 bslbf
        se (command_id = = load_map_table)
          map_table_data()

nos outros casos
        se (command_length = = 15)
          para (i = 1;i<command_length;i + + )
            reservado                                                   8 bslbf

nos outros casos
          para (i = 1;i<long_length;i + + )
            reservado                                                   8 bslbf
```

- *command_id* – Identifica o tipo de comando. O valor 0x10 denota Tabela de Aplicação de Carregamento (Load Map Table). Este comando é seguido por um campo *map_table_data()* que é definido no Quadro V. O valor 0x11 denota Fim da Linha (End of Line).

O Quadro V apresenta a sintaxe dos *map_table_data()*.

Quadro V

map_table_data	
número_de_entradas	8 uimsbf
para (i = 1;i<number_of_entries;i + +)	
2_to_4_map_table_flag	1 bslbf
2_to_8_map_table_flag	1 bslbf
4_to_8_map_table_flag	1 bslbf
entry_value	4 uimsbf
map_table_output_value	8 uimsbf

Aplica-se a seguinte definição semântica aos elementos dos dados apresentados neste Quadro:

- *number_of_entries* – Especifica o número de entradas da Tabela de Aplicação (Map Table) que são carregadas.
- *2_to_4_map_table_flag* – Se for colocado em "1", indica que esta entrada foi carregada na "Tabela de Aplicação 2 a 4".
- *2_to_8_map_table_flag* – Se for colocado em "1", indica que esta entrada foi carregada na "Tabela de Aplicação 2 a 8".
- *4_to_8_map_table_flag* – Se for colocado em "1", indica que esta entrada foi carregada na "Tabela de Aplicação 4 a 8".
- *map_table_type* – Identifica o tipo de Tabela de Aplicação que foi carregado. O valor "01" denota uma "Tabela de Aplicação 2 a 4", o valor "10" denota uma "Tabela de Aplicação 2 a 8", o valor "11" denota uma "Tabela de Aplicação 4 a 8".
- *entry_value* – Especifica a entrada da Tabela de Aplicação. O valor da entrada da primeira entrada da Tabela de Aplicação é zero.

- *map_table_output_value* – Especifica o valor de saída da Tabela de Aplicação para esta entrada.

A sintaxe e a semântica acima serão agora elucidadas numa linguagem mais corrente. O *PES_data_field()* (Quadro I) especifica pelo menos uma "página" que compreende uma ou mais regiões. As Figs. 1-3 mostram alguns exemplos de uma área activa de vídeo que compreende uma ou mais regiões. A Fig. 1 mostra o logotipo A de um fornecedor de programas e um subtítulo corrente B. A Fig. 2 mostra um subtítulo adicional C para espectadores com deficiência auditiva (as pessoas com deficiência auditiva precisam com frequência de textos explicativos adicionais, como por exemplo «aplausos da assistência», «música em crescendo»). A Fig. 3 mostra o logotipo A e um subtítulo Bi, em que a referência i indica uma língua que pode ser seleccionada entre uma variedade de línguas.

Os dados para cada região são colocados no *region_data_field()* (Quadro II). Este campo especifica o tamanho e a posição da região respectiva. Com referência à Fig. 4, é denotada a posição de uma região (x, y), sendo x especificado por *region_horizontal_address* e y por *region_vertical_address*. Estes parâmetros de posição podem ser alterados num prazo que permite o rolamento (scrolling), pela retransmissão de dados codificados para uma determinada região (especificada pela sua *region_id*) sem retransmissão de códigos de pixels e de dados da tabela de consulta de cores. O tamanho da região é p*n, sendo p e n especificados respectivamente por *number_of_pixels* e *number_of_lines*. Cada região compreende um máximo de 16 sub-regiões. Na Fig. 4 estão representadas duas sub-regiões que compreendem n_1 e n_2 linhas, respectivamente. O *region_data_field()* especifica os dados da tabela de consulta de cores e/ou dados dos pixels para cada sub-região.

Nem todos os parâmetros e dados acima referidos são obrigatórios. A sua presença é indicada pelos sinalizadores respectivos. Isto permite actualizar ou modificar regiões e/ou sub-regiões já definidas por referência à sua *region_id* e/ou *sub_region_id*, respectivamente. Por exemplo: a posição de uma região pode ser modificada (o que permite o rolamento (scrolling)); uma região já exibida pode tornar-se invisível; ou uma sub-região também pode ser consultada por outra região e assim exibida noutro local do ecrã sem que todos dos dados das imagens sejam retransmitidos.

A tabela de consulta de cores é carregada ou actualizada pelo campo *colour_look_up_data()* (Quadro III). Este campo especifica que valores de saída de luminância, crominância e transparência serão guardados em que entradas da CLUT. A transparência define até que ponto os pixels do vídeo e gráficos devem ser misturados no ecrã.

Os dados dos pixels para uma sub-região (ou para uma região inteira, se o número de sub-regiões for 1) são instalados no campo *coded_sub_region_data()* (Quadro IV). Os pixels são definidos como entradas da tabela de consulta de cores. Os valores dos pixels podem ser definidos em 2, 4 ou 8 bits. Existem três modos de codificação. Num primeiro modo de codificação (o *coding_mode* é "01", "10", "11"), são especificados individualmente pixels consecutivos. O número de pixels assim transmitidos é definido pelo *pixels_in_mode*. Num segundo modo de codificação (o *coding_mode* é "00", a *coding_mode_extension* é "01", "10", "11"), séries de pixels idênticos consecutivos são codificadas em série. Os primeiro e segundo modos de codificação podem misturar-se dentro de uma região. Para minimizar a complexidade da descodificação, houve o cuidado de alinhar os dados codificados em série segundo os bytes. Cada código em série compreende dois bytes: 2 bits para o modo de codificação, 2 bits para a extensão do modo de codificação e 12 bits para o código em série. Este último define uma série de 10 bits de pixels de 2 bits, uma série de 8 bits de pixels de 4 bits ou uma série de 4 bits de pixels de 8 bits, como é especificado pela extensão do modo de codificação. Um terceiro modo de codificação (o *coding_mode* é "00", a *coding_mode_extension* é "00") especifica comandos como as tabelas de aplicação e indicadores de fim-de-linha que funcionam nos dados descodificados em série. O conceito de tabelas de aplicação é abaixo descrito mais pormenorizadamente.

Considera-se que os chips Exibição-no-Ecrã (On-Screen-Display) existentes são usados para exibir as imagens gráficas. Alguns destes chips permitem 2 bits por pixel, outros 4 bits por pixel ou 8 bits por pixel. Surge um problema potencial se os pixels incluídos numa região forem codificados usando um número de bits por pixel diferente da largura de entrada da tabela de consulta de cores. Para resolver este problema, introduziu-se o conceito de tabelas de aplicação. As tabelas de aplicação também aumentam a eficiência da codificação. Permitem que os pixels sejam codificados em menos bits por pixel que a largura de entrada da tabela de consulta de cores. Estes aspectos serão agora elucidados através de um exemplo. Suponhamos uma (sub)região em que se usam apenas quatro

cores diferentes. As cores respectivas são guardadas, em termos de luminância e crominância, em quatro entradas de uma CLUT. Os dados dos pixels relativos à referida região são codificados em 2 bits/pixel. Contudo, o descodificador pode ter uma CLUT de 4 bits, exigindo assim quatro bits por pixel. Ora, a tabela de aplicação proporciona a aplicação dos códigos de 2 bits/pixel transmitidos a uma entrada da CLUT de 4 bits. Assim, a tabela de aplicação informa um descodificador quais as 4 das 16 entradas que devem ser usadas. O facto de a codificação de 2 bits/pixel melhorar realmente ou não a eficiência depende do investimento para alterar o modo de codificação ou a tabela de aplicação. Assim, instalaram-se tabelas de aplicação para aplicar uma codificação de n bits/pixel a uma CLUT de m bits, em que n e m são números diferentes. Considerando esta elucidação do conceito de tabelas de aplicação, o campo *map_table_data ()* (Quadro V) não precisa de mais explicações. É definido um valor de saída para cada valor de entrada da tabela de aplicação.

A Fig. 5 mostra um diagrama esquemático de um transmissor de acordo com a invenção. O transmissor está preparado para transmitir simultaneamente um sinal de vídeo e imagens gráficas. O transmissor recebe um sinal de vídeo em banda de base em formato YUV e um sinal áudio AUD associado e compreende um codificador de vídeo MPEG2 **101** e um codificador de áudio MPEG2 **102** para codificar o referido sinal de vídeo e áudio formando fluxos elementares em pacotes respectivamente PSV e PSA. O transmissor compreende ainda um terminal de edição **103** para editar subtítulos e/ou outras imagens gráficas como por exemplo um logotipo do fornecedor do programa. O terminal de edição é acoplado a um dispositivo de processamento e armazenamento **104**. Parte-se do princípio de que uma pluralidade de subtítulos associados ao programa de televisão em causa, possivelmente numa pluralidade de línguas diferentes, é previamente guardada no dispositivo de processamento e armazenamento. O sinal de vídeo é também aplicado a um gerador de tempo **105** que aplica um tempo t ao dispositivo de processamento e armazenamento **104**. O tempo t pode ser o tempo que decorreu desde o início do programa de televisão em causa ou pode ser um código de tempo associado ao programa e guardado na mesma fita que o sinal de televisão. Nos sistemas MPEG, o tempo t corresponderá à Referência do Relógio do Sistema (System Clock Reference) SCR do MPEG. O dispositivo de processamento e armazenamento compara o tempo t com um selo de tempo associado a cada subtítulo guardado. Quando chega o momento de um subtítulo, os dados relevantes são aplicados a um codificador e formador de pacotes **106** que forma um fluxo elementar em pacote PSG que define dados codificados de acordo

com o formato da transmissão anteriormente analisado. Os fluxos elementares em pacotes PSV, PSA e PSG são multiplexados pelo multiplexador MPEG **107** de modo a formar um fluxo de transporte TS. O fluxo de transporte é transmitido a um receptor através de um meio de transmissão ou armazenamento (não representado).

A Fig. 6 mostra um diagrama esquemático de um receptor de acordo com a invenção. O receptor compreende um desmultiplexador MPEG2 **201** que recebe o fluxo de transporte TS do MPEG2 e selecciona o pacote do fluxo de áudio PSA, o fluxo de vídeo PSV e os dados gráficos codificados PSG em conjunto, formando um programa de televisão seleccionado. O fluxo de áudio é descodificado por um descodificador de áudio MPEG2 **202**. O sinal de áudio AUD descodificado é aplicado a um locutor **203** para ser reproduzido. O fluxo de vídeo é descodificado por um descodificador de vídeo MPEG2 **204** para obter o sinal de vídeo descodificado em formato YUV para ser exibido num ecrã de exibição **205**. O fluxo codificado de dados gráficos é aplicado a um analisador gramatical (parser) de dados **206**. Este circuito analisa os dados instalados nos pacotes PES PSG e fornece os seguintes sinais de saída:

- o selo do tempo de apresentação PTS colocado no registo de cabeçalho PES, indicando o momento em que a região será exibida;
- o valor do *visibility_flag* (Quadro II), denotado como VF;
- os dados codificados dos pixels tal como são definidos pelos campos *n_bit_code* no Quadro IV (n=2,4,8); os código dos pixels de 8 bits são denotados como PX8, os códigos dos pixels de 4 bits são denotados como PX4 e os códigos dos pixels de 2 bits são denotados como PX2;
- a posição (x,y) e o tamanho (p,n) da região recebida no momento tal como são definidos pelos campos *region_horizontal_address*, *region_vertical_address*, *number_of_pixels* e *number_of_lines*, respectivamente (Quadro II, Fig. 4);
- os dados da tabela de aplicação denotados como MTD definidos no campo *map_table_data* (Quadro V) (este sinal não é usado na presente forma de realização);
- um sinal denotado como LoC derivado do campo *region_level_of_compatibility* (Quadro I); e
- dados da tabela de consulta de cores denotados como CTD, definidos pelo campo *colour_look_up_table_data* (Quadro III).

Os dados dos pixels PXn (n=2,4,8) recebidos do analisador gramatical **206** são guardados (quando são codificados em série por intermédio de um descodificador da sequência em série **207**) numa memória de pixels **208**, sendo as localizações relevantes da memória determinadas pela posição (x,y) e pelo tamanho (p,n) recebidos do analisador gramatical. A memória de pixels **208** é lida por sinais de regulação do tempo de exibição (não representados). Os pixels PX2, PX4 ou PX8 (uma zona pode compreender pixels tanto de 2 como 4 ou 8 bits, mas só um deles está activo no tempo de duração de um pixel) são assim aplicados a um circuito gerador de cor **209** em sincronismo com o sinal de vídeo recebido.

O circuito gerador de cor **209** compreende uma tabela de consulta de cores **2092**. Nesta forma de realização, parte-se do princípio de que a referida CLUT tem quatro entradas apenas, isto é, que a sua entrada tem 2 bits de largura. A CLUT tem, para cada entrada, um sinal de saída denotado como OSD em termos de um valor de luminância Y de 8 bits e de valores de crominância U e V de 8 bits, e de um valor de transparência T de 8 bits. Os valores YUV e T relevantes são guardados na CLUT em reacção aos dados CTD da tabela de consulta de cores recebidos do analisador gramatical.

Como a tabela de consulta de cores **2092** tem uma entrada de 2 bits, os pixels de 2 bits PX2 são directamente aplicados nela. Os pixels de 4 bits PX4 e os pixels de 8 bits PX8 são aplicados à CLUT através de um circuito de redução de 4 para 2 bits **2093** e de um circuito de redução de 8 para 2 bits **2094**, respectivamente. Numa forma de realização simples, os referidos circuitos de redução limitam-se a pôr de lado os bits de entrada menos significativos. Numa forma de realização alternativa, o bit mais significativo não sofre alteração (ou alguns dos bits mais significativos são tornados OR) para obter o primeiro bit de saída, enquanto os bits de entrada menos significativos são tornados OR para obter o segundo bit de saída.

Os circuitos de redução **2093** e **2094** permitem que dados da CLUT de nível superior (> 2 bits) sejam descodificados de maneira compatível por CLUT's de nível inferior (2 bits de entrada). Contudo, um fornecedor de programas pode impedir que as zonas definidas por mais de 4 cores sejam descodificadas por CLUT's de 2 bits. Isto consegue-se por meio de um interruptor **2091** controlado pelo sinal LoC recebido do analisador gramatical, isto é, pelo campo *region_level_of_compatibility* (ver Quadro I). Se o interruptor estiver na posição denotada como 4 ou 8 (isto é, se o fornecedor do programa exigir pelo menos

uma CLUT de 4 bits ou uma CLUT de 8 bits, respectivamente), a região não será exibida. Só se o interruptor estiver na posição denotada como 2 (isto é, se o fornecedor do programa aceitar que sejam exibidos pixels em apenas quatro cores) é que os dados dos pixels são realmente aplicados à CLUT.

Em reacção aos códigos dos pixels de 2 bits, a CLUT **2092** emite o sinal de saída de exibição OSD e o sinal de transparência T. O sinal de saída OSD é aplicado a um activador de mistura **210** onde é misturado, numa base de pixel por pixel, com o sinal de vídeo YUV. O sinal de transparência T determina a quantidade de transparência dos pixels gráficos. Se o valor de T for 0, o sinal de vídeo é completamente apagado e os pixels gráficos não são afectados. Se o valor de T for 255 (isto é, 100%), o pixel gráfico relevante é totalmente transparente. Quando T tem outros valores, o nível de transparência é linearmente interpolado. Isto permite que o sinal de vídeo se mantenha visível através da imagem gráfica.

O sinal combinado é então exibido no ecrã de exibição **205**. Contudo, o facto de a imagem gráfica ser de facto exibida ou não depende de um interruptor **212** que é controlado por um circuito de activação da exibição **213**. Este circuito recebe o selo de tempo de apresentação PTS e o sinalizador de visibilidade VF do analisador gramatical bem como o tempo actual t (não representado) tal como é transmitido pela System Clock Reference do MPEG. O circuito de activação da exibição **213** está preparado para activar a exibição da região gráfica logo que t> PTS e para desactivar a sua exibição quando o *visibility_flag* da zona em causa é colocado em "0".

A Fig. 7 mostra outra forma de realização do circuito gerador de cor **209** do receptor. Nesta forma de realização, a CLUT **2092** do receptor tem uma entrada de 4 bits, isto é, a CLUT compreende dezasseis entradas. Agora, os pixels de 4 bits PX4 são aplicados directamente ao interruptor **2091** enquanto os pixels de 8 bits PX8 passam através de um circuito de redução de 8 para 4 bits **2096** que, por exemplo, rejeita os quatro bits de entrada menos significativos. Os pixels de 2 bits passam através de uma tabela de aplicação de 2 para 4 bits **2095**, que é na realidade uma memória com quatro localizações de 4 bits. A tabela de aplicação **2095** aplica os pixels de 2 bits em entradas de 4 bits da CLUT, tal como é especificado pelos dados da tabela de aplicação MTD recebidos do analisador gramatical. O conceito de tabelas de aplicação já foi anteriormente descrito. Se o interruptor **2091** estiver na posição denotada como 8 (isto é, se o fornecedor do

programa exige pelo menos uma CLUT de 8 bits), a região não será exibida. Se o interruptor estiver na posição denotada como 2 ou 4 (isto é, se o fornecedor do programa aceitar que pixels de 8 bits sejam exibidos em apenas dezasseis cores), os dados dos pixels são de facto aplicados à CLUT.

A Fig. 8 mostra uma forma de realização do circuito gerador de cor **209** em que a CLUT **2092** tem uma entrada de 8 bits, isto é, a CLUT compreende 256 entradas. Agora, os pixels de 8 bits PX8 são aplicados directamente ao interruptor **2091**, enquanto os pixels de 2 bits PX2 e os pixels de 4 bits PX4 passam através de uma tabela de aplicação de 2 para 8 bits **2097** e de uma tabela de aplicação de 4 para 8 bits **2098**, respectivamente. A tabela de aplicação **2097** aplica a informação dos pixels de 2 bits nas entradas de 8 bits da CLUT, tal como é especificado pelos dados da tabela de aplicação MTD recebidos do analisador gramatical. A tabela de aplicação **2098** faz o mesmo em relação aos pixels de 4 bits. Rigorosamente falando, o interruptor **2091** é redundante nesta forma de realização, dado que este receptor trata de todas as transmissões, independentemente das exigências da CLUT transmitidas.

Como se explicou acima, os circuitos de redução (**2093**, **2094**, **2096**) permitem que dados de uma CLUT de nível superior sejam decodificados de modo compatível por CLUT's de nível inferior, enquanto as tabelas de aplicação (**2095**, **2097**, **2098**) permitem que CLUT's de nível inferior sejam dinamicamente aplicadas em CLUT's de nível superior. O Quadro VI sintetiza as várias opções e indica os vários níveis de desempenho.

Quadro VI

CLUT de 2 bits	0	0	-	0	-	-
CLUT de 4 bits	0	+	+ +	+	+ +	-
CLUT de 8 bits	0	+	+ +	+ +	+ + +	+ + + +
	LoC = 2	LoC = 2	LoC = 4	LoC = 2	LoC = 4	LoC = 8
	Dados de 2 bits	Dados de 4 bits		Dados de 8 bits		

O Quadro VI apresenta três filas que indicam o desempenho de receptores que têm respectivamente uma CLUT de 2 bits, de 4 bits e de 8 bits. Três colunas indicam se uma

região compreende pixels de 2 bits, 4 bits ou 8 bits, respectivamente. LoC denota o nível de compatibilidade incluído nos dados codificados.

Se uma região é definida em 2 bits/pixel apenas, podem ser exibidas quatro cores (desempenho denotado como 0) independentemente de a entrada da CLUT ter 2, 4 ou 8 bits de largura. Não faz sentido que um fornecedor de programas exija, para essa região, uma CLUT melhor que de 2 bits. Isto está indicado por LoC=2 na coluna respectiva.

Se uma região compreende pixels de 4 bits, o fornecedor do programa pode permitir que essa região seja exibida usando uma CLUT de 2 bits (LoC=2). Os referidos receptores de 2 bits exibem a região em quatro cores (desempenho denotado como 0), enquanto os receptores de 4 bits e de 8 bits exibem 16 cores (desempenho denotado como +). Contudo, o fornecedor do programa pode exigir também que o receptor tenha uma CLUT de 4 bits (LoC=4). Nesse caso, um receptor de 2 bits não pode exibir a região (desempenho denotado como -), enquanto os receptores de 4 bits e de 8 bits exibirão 16 cores (desempenho denotado como ++).

Se uma região compreende pixels de 8 bits, o fornecedor do programa pode permitir que essa região seja exibida usando uma CLUT de 2 bits (LoC=2). Nesse caso, os receptores de 2 bits exibem 4 cores (desempenho denotado como 0), os receptores de 4 bits exibem 16 cores (desempenho denotado como +) e os receptores de 8 bits exibem a gama completa de cores (desempenho denotado como ++). O fornecedor do programa pode exigir também que o receptor tenha pelo menos uma CLUT de 4 bits (LoC=4). Nesse caso, os receptores de 2 bits não podem exibir a região em causa (desempenho denotado como -), os receptores de 4 bits exibem 16 cores (desempenho denotado como ++) e os receptores de 8 bits exibem a gama completa de cores (desempenho denotado como +++). Se o fornecedor do programa exigir que o receptor tenha uma CLUT de 8 bits (LoC=8), nem os receptores de 2 bits nem os de 4 bits podem exibir a região em causa (desempenho denotado como -), enquanto os receptores de 8 bits têm um desempenho ideal (denotada como ++++). Este último desempenho é ideal porque o fornecedor do programa não precisa de ter em conta as restrições dos receptores de nível inferior.

Resumindo a invenção, descreve-se um método para a transmissão simultânea de um sinal de vídeo e de dados codificados que representam imagens gráficas. A invenção é particularmente aplicável para a transmissão de subtítulos em várias línguas com um

programa de vídeo. As imagens gráficas são regiões rectangulares dentro da área activa do vídeo. São transmitidas sob a forma de aplicações de bits. A invenção permite não só a exibição de qualquer tipo ou tamanho de caracteres mas também a definição de por exemplo o logotipo de um fornecedor de programas. Os dados codificados incluem um selo de tempo para especificar o momento em que um subtítulo será exibido. As formas de realização preferidas da invenção incluem a transmissão de dados da tabela de consulta de cores (CLUT) e de um código de compatibilidade que indica um número mínimo requerido de entradas da referida tabela de consulta de cores. Para os receptores com uma CLUT com mais entradas que as necessárias, é transmitida uma tabela de aplicação para a aplicação da largura dos dados dos pixels à largura de entrada da CLUT em causa. O método é aplicável aos sistemas de Direct Video Broadcast (Difusão Directa de Vídeo) em que o sinal de vídeo é codificado em MPEG2 e as imagens gráficas são instaladas num fluxo de dados privado de um Fluxo de Transporte (Transport Stream) MPEG2.

Lisboa, 29 FEV. 2000

Por Koninklijke Philips Electronics N.V.

ENG.º MANUEL MONIZ PEREIRA

Agente Oficial da Propriedade Industrial

Arco da Conceição, 3, 1.º - 1100 LISBOA

REIVINDICAÇÕES

1. Método para transmitir dados codificados que definem uma imagem gráfica sob a forma de uma região rectangular para exibir dentro de uma área de um sinal activo de vídeo, em que os pixels que formam a referida região são individualmente definidos pelos dados codificados, caracterizado por os dados codificados incluírem o tamanho e a posição da referida região e um selo de tempo que representa o momento em que a referida região será exibida.

2. Método tal como reivindicado na reivindicação 1, em que is dados codificados subsequentes para determinada região incluem um código de visibilidade para desactivar a exibição da referida região.

3. Método tal como reivindicado na reivindicação 1, em que uma região compreende uma pluralidade de sub-regiões que incluem um código identificador para permitir que uma imagem gráfica definida pela referida sub-região seja consultada por outra região.

4. Método tal como reivindicado na reivindicação 1, em que os pixels para uma região são codificados como entradas de uma tabela de consulta de cores, em que os dados codificados incluem um código de compatibilidade que indica um número mínimo exigido de entradas da referida tabela de consulta de cores.

5. Método tal como reivindicado na reivindicação 4, em que os pixels são definidos por dados de pixels de n bits, em que os dados codificados incluem ainda uma tabela de aplicação para aplicação dos referidos dados dos pixels de n bits a uma tabela de consulta de cores com mais de 2^n entradas.

6. Método tal como reivindicado em qualquer uma das reivindicações anteriores, em que os dados codificados são colocados num fluxo privado de transporte de acordo com o padrão MPEG2 ou sua extensão.

7. Método de recepção de dados codificados que definem uma imagem gráfica sob a forma de uma região rectangular para exibição dentro de uma área de um sinal activo de

vídeo, o qual compreende os passos de descodificar os referidos dados codificados em pixels individuais que constituem a referida região, armazenar os referidos pixels e gerar sinais de exibição que representam os referidos pixels, caracterizado pelos passos de descodificar, dos referidos dados codificados, o tamanho e a posição da referida região e um selo de tempo e de exibir a região com os referidos tamanho e posição a partir de um momento representado pelo referido selo de tempo.

8. Método tal como reivindicado na reivindicação 7, que compreende ainda os passos de descodificar um código de visibilidade incluído em dados codificados subsequentes para uma determinada região e de desactivar a geração dos referidos sinais de exibição para a referida região em reacção ao referido código de visibilidade.

9. Método tal como reivindicado na reivindicação 7, em que uma região compreende uma pluralidade de sub-regiões que incluem um código identificador, o qual compreende os passos de incluir a referida sub-região numa outra região reportada à referida sub-região.

10. Método tal como reivindicado na reivindicação 7, em que os pixels para uma região são codificados como entradas de uma tabela de consulta de cores e os dados codificados incluem um código de compatibilidade que indica um número mínimo exigido de entradas da referida tabela de consulta de cores, o qual compreende o passo de desactivar a geração de sinais de exibição se o número de entradas exigido for inferior ao disponível.

11. Método tal como reivindicado na reivindicação 10, que compreende ainda os passos de receber dados da tabela de aplicação e de guardar os referidos dados da tabela de aplicação numa tabela de aplicação para aplicação dos dados dos pixels com n bits a uma tabela de consulta de cores com mais de 2^n entradas.

12. Método tal como reivindicado em qualquer uma das reivindicações 7-11, em que o sinal de vídeo é codificado de acordo com um padrão MPEG2 ou sua extensão e em que os dados codificados que definem a imagem gráfica são colocados em pacotes de um fluxo privado de transporte MPEG2.

13. Transmissor para transmitir dados codificados que definem uma imagem gráfica sob a forma de uma região rectangular dentro de uma área de um sinal activo de vídeo, o qual compreende meios para codificar os pixels individuais que formam a referida região, caracterizado por compreender ainda meios para incluir nos dados codificados o tamanho e a posição da referida região e um selo de tempo que representa o momento em que a referida região será exibida.

14. Receptor acoplado a um ecrã de exibição para receber dados codificados que definem uma imagem gráfica sob a forma de uma região rectangular dentro de uma área de um sinal activo de vídeo, que compreende meios para descodificar os referidos dados codificados em pixels individuais que constituem a referida região, meios para guardar os referidos pixels e meios para gerar sinais de exibição que representam os referidos pixels, caracterizado por compreender ainda meios para descodificar, dos referidos dados codificados, o tamanho e a posição da referida região e um selo de tempo e meios para exibir a região com os referidos tamanho e posição a partir de um momento representado pelo referido selo de tempo.

15. Sinal de imagem que compreende dados codificados que definem uma imagem gráfica sob a forma de uma região rectangular dentro de uma área de um sinal activo de vídeo, em que os pixels que formam a referida região são definidos individualmente pelos dados codificados, caracterizado por os dados codificados incluírem o tamanho e a posição da referida região e um selo de tempo que representa o momento em que a referida região será exibida.

Lisboa, 29 FEV. 2000

Por Koninklijke Philips Electronics N.V.

ENGº MANUEL MOÑIZ PEREIRA

Agente Oficial da Propriedade Industrial

Arco da Conceição, 3, 1º - 1100 LISBOA

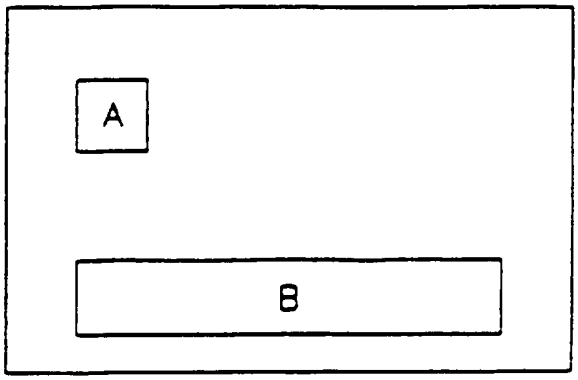


FIG. 1

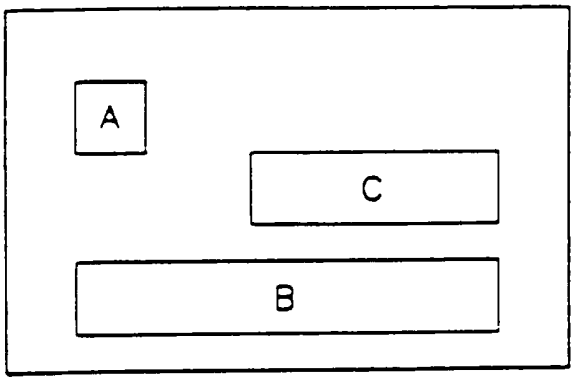


FIG. 2

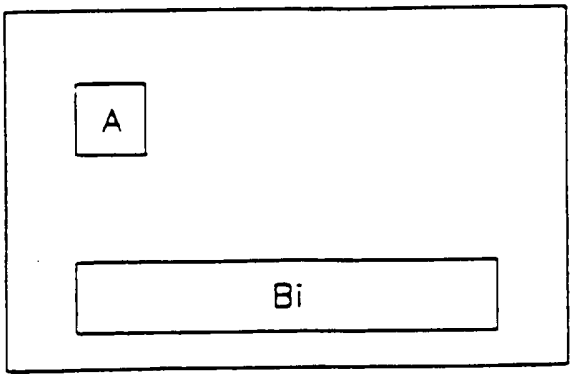


FIG. 3

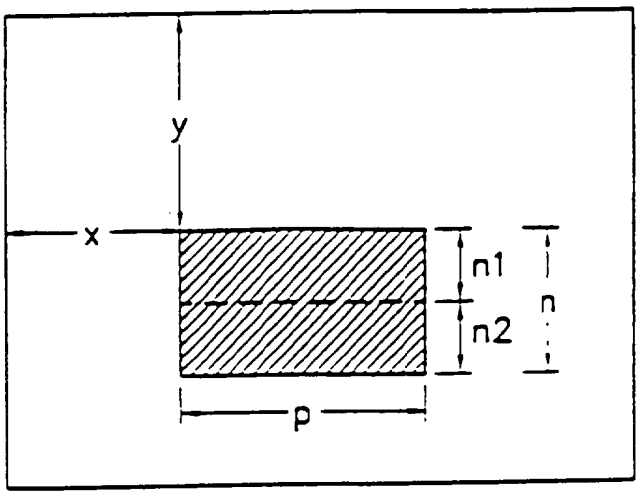


FIG. 4

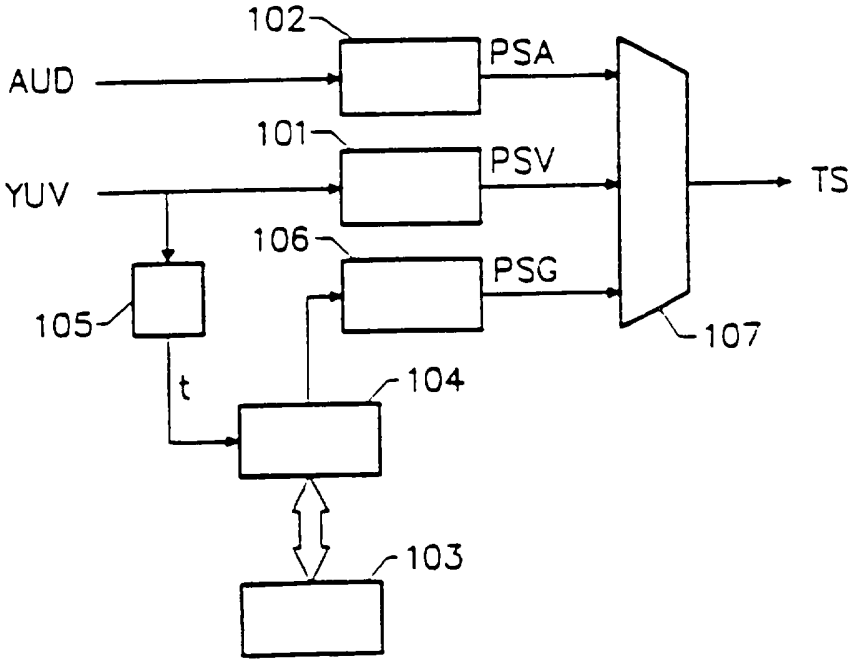


FIG. 5

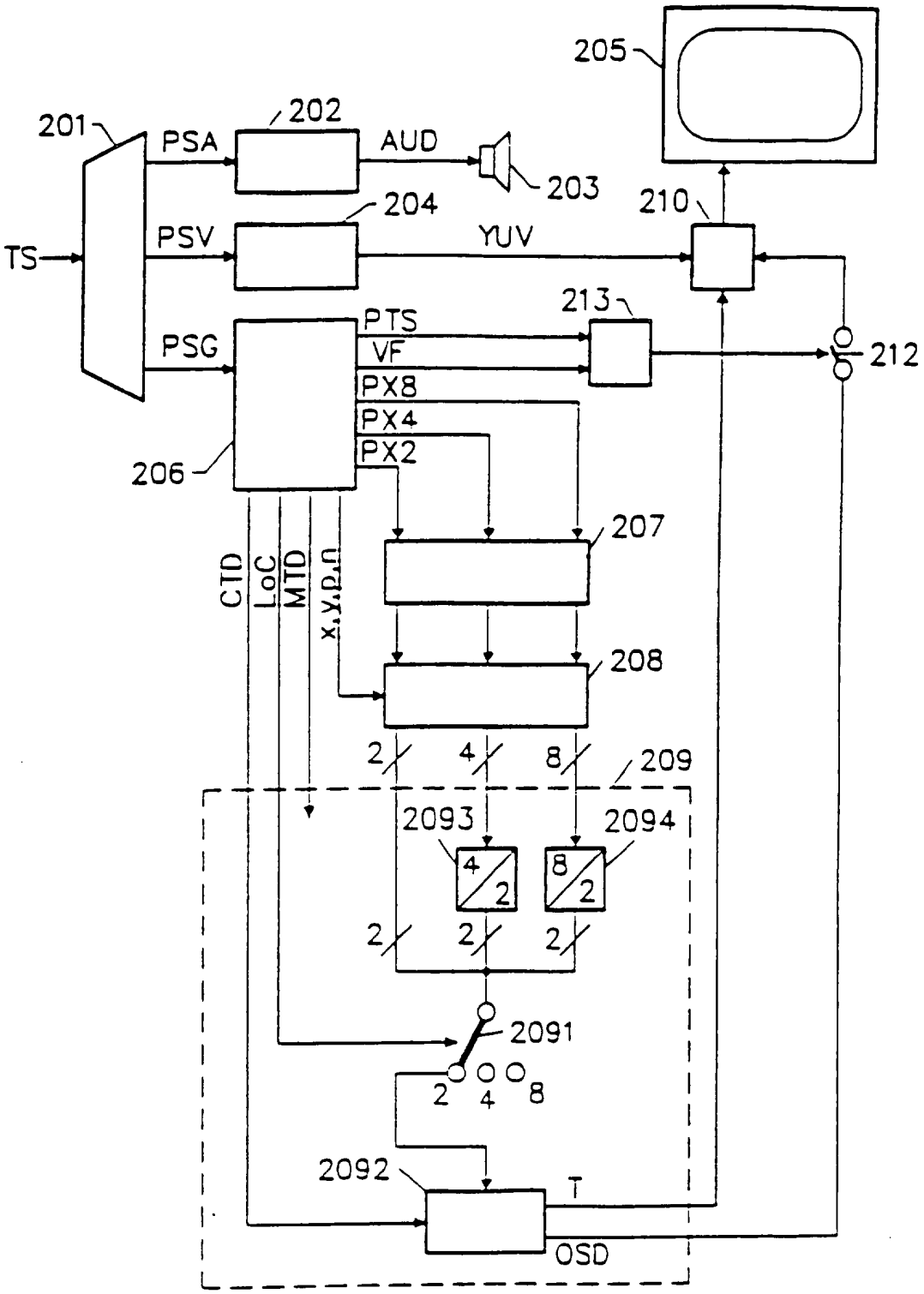

205
201
202
PSA
AUD
203
210
TS
204
PSV
YUV
213
PTS
PSG
VF
212
PX8
PX4
PX2
206
207
CTD
LoC
MTD
x,y,p,n
208
2
4
8
209
2093
2094
4
2
8
2
2
2
2
2091
2
4
8
2092
T
OSD

FIG. 6

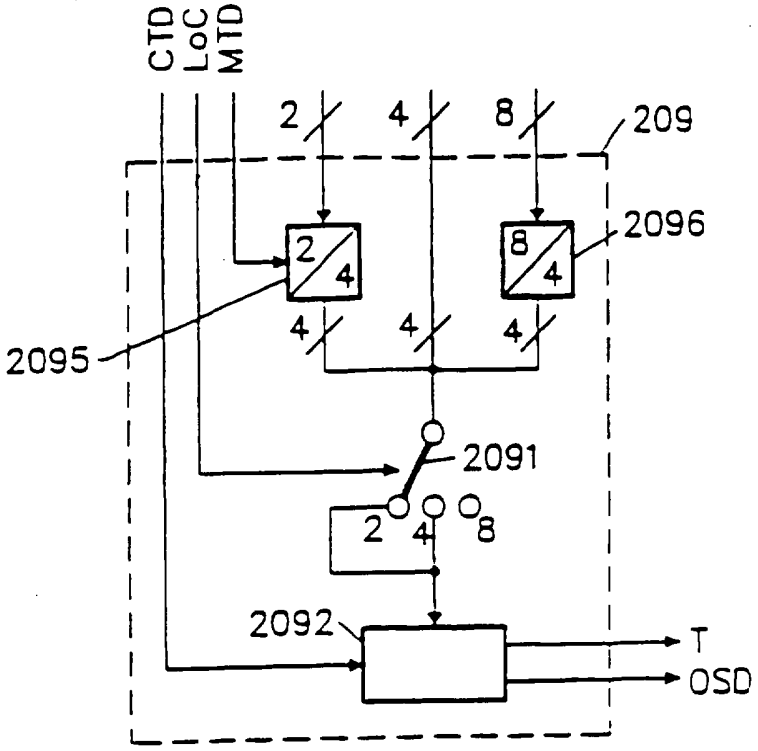

CTD
LoC
MTD
2
4
8
209
2096
2095
2091
2092
T
OSD

FIG. 7

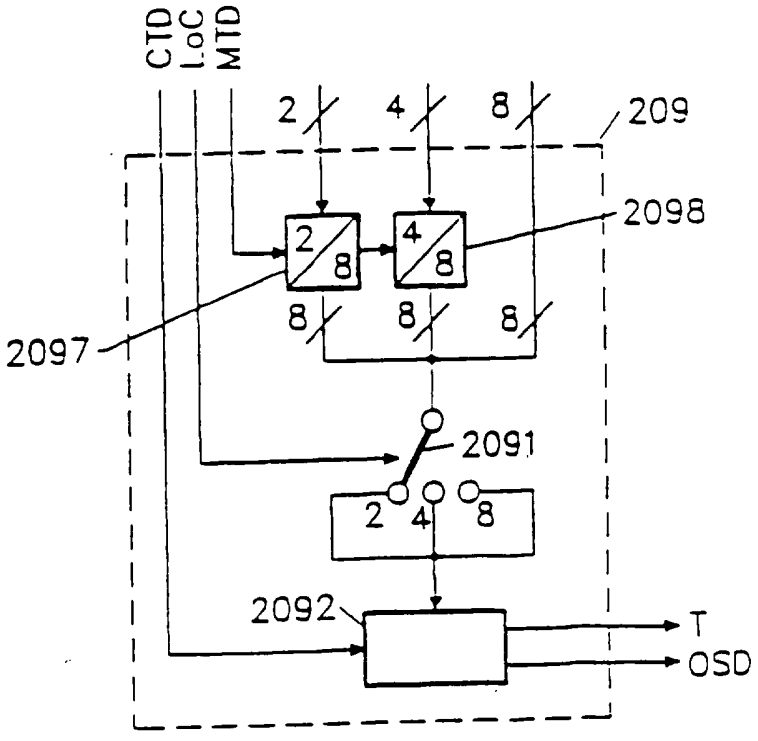

CTD
LoC
MTD
2
4
8
209
2098
2097
2091
2092
T
OSD

FIG. 8