



INSTITUTO NACIONAL
DA PROPRIEDADE INDUSTRIAL

(11) *Número de Publicação*: PT 728358 E

(51) *Classificação Internacional*: (Ed. 6)

G11B023/28 A H04L009/30 B
H04N007/16 B G11B020/00 B
G06F001/00 B G06F012/14 B

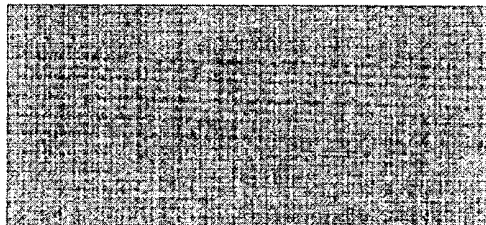
(12) *FASCÍCULO DE PATENTE DE INVENÇÃO*

(22) <i>Data de depósito</i> : 1994.10.13	(73) <i>Titular(es)</i> : TIME WARNER ENTERTAINMENT CO., L.P. 4000 WARNER BOULEVARD BURBANK, CA 91522 US
(30) <i>Prioridade</i> : 1993.10.29 US 144829	
(43) <i>Data de publicação do pedido</i> : 1996.08.28	(72) <i>Inventor(es)</i> : CHRISTOPHER J. COOKSON LEWIS S. OSTOVER US US
(45) <i>Data e BPI da concessão</i> : 2001.03.07	(74) <i>Mandatário(s)</i> : LUÍS MANUEL DE ALMADA DA SILVA CARVALHO RUA VITOR CORDON, N° 14 - 3° 1200 LISBOA PT

(54) *Epígrafe*: SISTEMA E MÉTODO PARA A AUTENTICAÇÃO DE PORTADORES DE SOFTWARE

(57) *Resumo*:

SISTEMA E MÉTODO PARA A AUTENTICAÇÃO DE PORTADORES DE SOFTWARE



428958

Wm. Suter Cumming

- 1 -

DESCRIÇÃO

"SISTEMA E MÉTODO PARA A AUTENTICAÇÃO DE PORTADORES DE *SOFTWARE*"

Esta invenção está relacionada com o tocar de portadores de *software* (por exemplo, filmes), e mais particularmente a uma técnica para a autenticação de discos ópticos tocáveis em gravadores compatíveis.

Antecedentes de invenção

O meio mais disseminado de distribuição de filmes é a cassete de vídeo. Todavia, discos ópticos codificados digitalmente são em teoria muito superiores para a distribuição de filmes e outras formas de apresentação. De especial vantagem é o uso de "vídeo comprimido", através do qual é possível codificar digitalmente um filme num disco que não é maior do que um CD de áudio dos dias de hoje. O armazenamento num disco óptico pode de facto ser tal densificação que o mesmo disco pode conter múltiplas versões (por exemplo, de classificação R e de classificação PG) do mesmo filme, múltiplas bandas sonoras em idiomas diferentes, e dados para outros sinais relacionados ou não relacionados.

Devido aos diferentes padrões utilizados na indústria televisiva em todo mundo, existem um igual número de padrões de cassetes de vídeo. Uma cassete de vídeo do tipo NTSC (*National Television Systems Committee*) vendida nos Estados Unidos, por exemplo, não tocará na maioria dos gravadores de cassetes de vídeo encontrados na Inglaterra. A fim de evitar que uma editora de

Very Sincerely,
[Signature]

software tenha de produzir discos ópticos em tantos formatos como são actualmente necessários para cassetes de vídeo, seria de longe preferível que os dados armazenados num disco óptico fossem convertidos para um padrão em particular pelo próprio gravador. Desta forma, o mesmo disco poderia ser vendidos em qualquer parte do mundo. Face às vantagens oferecidas pelo armazenamento de dados digitais em discos ópticos, é possível obter esta flexibilidade acompanhado do armazenamento de tantos sinais diferentes.

Todavia, o desenvolvimento de novos formatos de discos ópticos e gravadores complementares envolve custos muito elevados. Mais, o lançamento de um novo formato de portador requer esforços consideráveis, e pelo menos nos primeiros anos, até que a aceitação do consumidor cresça, os lucros poderão ser elusivos. É aceite que, conforme a popularidade de um novo formato cresce, outras editoras de *software* tirarão vantagem do formato e produzirão os seus portadores no novo formato a par do velho formato. Isto levanta dificuldades às editoras de *software* que são pioneiras do novo formato e não se sentam à espera para ver se o novo formato se estabelece.

A fim de assegurar a participação o mais cedo possível do maior número de editoras de *software* quanto possível, ou que aqueles que entrarem mais tarde paguem o devido, seria altamente vantajoso se existisse uma forma de um novo tipo de gravador poder tocar um novo tipo de disco apenas se o disco for "autorizado" pelos fabricantes de *hardware / software* que estão a tentar estabelecer o novo formato. Se existisse uma forma de evitar que editoras de *software* não autorizadas possam distribuir discos que poderiam tocar num novo tipo de gravador, então todas as editoras de *software* que desejassem tirar partido do novo formato seriam obrigadas a partilhar nos custos de desenvolvimento e do marketing inicial. É assim um objectivo da presente invenção proporcionar um sistema e método para prevenir o tocar de portadores de *software* não

autorizados num gravador.

Uma vez tornado possível prevenir qualquer tocar de um disco óptico não autorizado, por exemplo, torna-se possível levar o bloqueio (*lock-out*) um passo mais à frente. Por isto quer-se dizer que é possível prevenir certos tipos de tocar, enquanto se permite o tocar de outros tipos. Por exemplo, um disco óptico pode conter dados digitais num formato que permitirá que um gravador compatível possa gerar um sinal de vídeo em qualquer dos padrões de transmissões no mundo, NTSC, PAL, SECAM, etc. Se é possível prevenir o tocar de um disco na sua totalidade, também deverá ser possível permitir o tocar de um disco a fim de gerar um sinal de vídeo em um ou mais padrões autorizados, enquanto previne a geração de sinais de vídeo em outros padrões. Uma editora de *software* cujos trabalhos de áudio e vídeo são destinados a um território em particular pode estar interessado, por exemplo, em ter os seus discos a tocar apenas em receptores de televisão do tipo SECAM e em mais nenhum. Tal editora deveria talvez de pagar menos por uma licença de utilização do que editoras cujos discos tocam em receptores de televisão de todos os padrões. O que é necessário em tal caso é o proporcionar aos fabricantes de *software* o conhecimento necessário para fabricar discos ópticos que tocam apenas em padrões seleccionados. É assim outro objectivo da presente invenção proporcionar uma maneira em que o tocar de um portador de *software* pode ser autorizado apenas de acordo com padrões seleccionados.

É de se entender que os princípios da presente invenção não estão limitados a qualquer tipo em particular de portador ou a qualquer tipo em particular de *software*. É verdade que a utilização prevista de maior disseminação para a presente invenção está relacionada com o desenvolvimento de novos formatos de discos ópticos. Todavia, é de se entender que a invenção não está limitada a qualquer tipo em particular de meio (por exemplo, é aplicável a portadores de fitas e a todos os meios de armazenamento digital), nem está

limitada apenas à distribuição de filmes. Por exemplo, num caso extremo, a invenção é aplicável à distribuição de uma biblioteca de imagens fixas, em que não há qualquer "movimento". O termo "editora de *software*" abrange muito mais do que uma empresa cinematográfica, e o termo "portador" abrange muito mais do que um disco óptico de codificação digital.

WO92/12485 apresenta uma técnica de autenticação para permitir a verificação de um documento, incluindo discos ópticos.

Sumário da invenção

A invenção conforme definida nas reivindicações 1 e 8 utiliza técnicas criptográficas para assegurar que qualquer disco que é tocado tenha sido autorizado. A mesma técnica de base pode ser utilizada para autorizar o disco na sua totalidade, ou para autorizar padrões individuais. A secção de *lead-in* de um disco pode conter um código de autorização para o disco na sua totalidade. Na personificação ilustrativa da invenção, existe um código de autorização para cada padrão individualmente. O gravador verifica que o necessário código de autorização está no disco. Se não estiver, o gravador não irá tocar de todo o disco, ou irá bloquear alguns padrões em particular. Obviamente, que não é suficiente atribuir simplesmente um código de autorização para o disco na sua totalidade, ou códigos de autorização para cada padrão diferente, porque uma editora não autorizada poderia facilmente determinar os códigos simplesmente através da leitura destes a partir dos discos das editoras autorizadas.

Por esta razão, um código de autorização é tornado numa função dos dados contidos no disco propriamente dito. A uma editora de *software* autorizada é proporcionado um algoritmo secreto, conhecido apenas pelas editoras autorizadas (ou editoras que são autorizadas a publicar para tocar num

Very Sincerely
Larry

padrão respectivo), que é utilizado para processar os dados no disco e para gerar um código "resultado". Este resultado é o código de autorização que está na realidade armazenado no disco. Cada gravador lê certos e predeterminados dados no disco, antes dos dados serem utilizados para gerar um sinal de vídeo, e computa um "resultado" comparável. Se os dois resultados coincidem, então o disco (ou o padrão em particular) é assumido como estando autorizado. Desta forma, não é suficiente o simples copiar do código de autorização num disco verdadeiramente autorizado porque o código de autorização necessário para qualquer disco é uma função do conteúdo do material do programa no disco, e isso, obviamente, varia de disco para disco. (Em vez de proporcionar o algoritmo secreto a todas as editoras autorizadas, uma organização central de licenciamento poderá fazer toda a codificação.)

Mesmo isto poderá não ser o suficiente, todavia, sendo que com tempo e esforço suficiente será possível fazer a engenharia inversa de um gravador e de aprender o algoritmo de processamento dos predeterminados dados num disco a fim de derivar os códigos de autorização. Uma vez conhecido o algoritmo utilizado num gravador, esse algoritmo poderia ser utilizado por uma editora não autorizada para derivar o código necessário para qualquer *software*. Com o algoritmo a ser codificado no *software* para controlar milhões de gravadores, e com dúzias de fabricantes de gravadores, poderá ser difícil assegurar o secretismo. Por esta razão, é preferida a utilização de códigos encriptados e o que é conhecido como criptografia de chave pública.

Como em qualquer esquema criptográfico, são utilizadas duas chaves - uma para encriptar e uma para desencriptar. O que é único da criptografia de chave pública é que mesmo que a chave pública for conhecida, a chave secreta não pode ser determinada a partir desta com um esforço razoável. Assumamos que o algoritmo para o processamento dos predeterminados dados

para derivar um código de autorização é conhecido por todos. Uma editora de *software* autorizada utiliza o algoritmo para derivar um código de autorização para o seu disco, mas o código de autorização propriamente dito não é armazenado no disco. O que é armazenado é o código de autorização encriptado, com a chave secreta a ser utilizada para a encriptação. Todos os gravadores executa o mesmo algoritmo e deriva o mesmo resultado, um código calculado. O gravador utiliza então o par de chaves públicas para descriptar o código no disco, e verifica se o código de descriptação condiz com o código calculado. Apenas se estes condizem é que é assumido que o código de autorização correcto está no disco. Este esquema funciona porque mesmo que o algoritmo e a chave pública sejam conhecidos, os fabricantes não autorizados não terão qualquer forma de saber como fazer encriptação, e serão incapazes de armazenar códigos de autorização encriptados nos seus discos. Esta técnica tem elementos em comum com a "autenticação de assinatura" em criptografia de chave pública.

Todavia, existe ainda um problema em termos práticos com esta abordagem, pelo menos na personificação ilustrativa da presente invenção. Conforme será descrito, versões múltiplas do mesmo filme podem ser tocadas. Dados são armazenados em blocos discretos, e a forma como as versões múltiplas do mesmo filme podem ser contidas numa única faixa é através do controlo do saltar por cima de certos blocos de dados (aqueles necessários para o tocar de uma versão que não esteja a ser vista no momento). O algoritmo para derivar o código computado a partir dos dados armazenados no disco processa os dados em blocos particulares (blocos com endereços específicos e predeterminados). Isto iria permitir a uma editora não autorizada o copiar de blocos de dados específicos a partir de um disco não autorizado e utilizar o código de autorização encriptado para aquele disco no disco não autorizado. O algoritmo do gravador iria processar aqueles blocos de dados, e derivar um código computado que iria condizer com o código de autorização no disco pertencente à editora legítima depois deste ser

Very Sincerely
Larry

descriptado no gravador. Enquanto que esta cópia poderia levar à apresentação de imagens totalmente fora do contexto (para não falar nos problemas de infracção de *copyright*), isto poderia ser evitado através do armazenamento dos códigos apropriados no disco que iriam levar o gravador a saltar por cima destes blocos copiados, da mesma forma que os blocos são saltados por cima quando estes não são requeridos para a versão seleccionada.

Mesmo este problema pode ser evitado assegurando que os blocos de dados que são processados para derivar o código em primeiro lugar, tanto pela editora de *software* e por cada gravador, são blocos na secção de *lead-in* do disco, cujos blocos contêm dados que não podem ser copiados a partir do disco de outra editora. Isto é porque os dados de *lead-in* consiste em coisas tais como uma tabela de conteúdos, idiomas de apresentação disponíveis, e muitos outros itens que são peculiares ao *software* que está a ser tocado. Desta forma, não é viável a uma editora não autorizada copiar o código e os blocos de dados de uma editora autorizada.

A mesma técnica básica pode ser utilizada para autorizar o tocar de um disco como um todo, ou o tocar de acordo com um qualquer padrão de vídeo em particular. Na personificação ilustrativa da invenção, é proporcionado um par diferente de chave pública / chave privada para cada padrão. O algoritmo para o processamento dos blocos de dados podem ser comuns, mas cada padrão requer a encriptação de um "resultado" comum com uma chave privada única que é fornecida apenas às editoras que estão autorizadas a distribuir discos que irão tocar de acordo com aquele padrão. O *software* do gravador verifica o padrão de vídeo desejado pelo utilizador, utiliza o algoritmo padrão para derivar o código computado, e depois utiliza uma chave pública armazenada exclusiva àquele padrão para descriptar o código de encriptado no disco para ver se este condiz com o código computado. Se estes não coincidem, o tocar naquele padrão não é

Wm. Gates Company

permitido.

A invenção é apresentada no contexto de um sistema geral que oferece numerosas funções vantajosas. A totalidade do sistema é descrito apesar de as reivindicações anexadas serem dirigidas a funções específicas. A lista geral de funções que são de interesse particular na descrição abaixo inclui:

- Vídeo padrão e impedimento territorial.
- Execução em relações de aspectos múltiplos.
- Execução em versões múltiplas, por exemplo, classificação PG e classificação R, do mesmo filme no mesmo disco, com desactivação parental automática e selectiva de visualização de classificação R.
- Códigos de autorização encriptados que previnem que editoras não autorizadas produzam discos utilizáveis.
- Fornecimento de faixas de áudio em múltiplos idiomas e faixas de subtítulos em múltiplos idiomas num único disco, com o utilizador a escolher o idioma da sua escolha.
- Fornecimento de múltiplas "outras" faixas de áudio, por exemplo, cada uma contendo algum componente de música orquestral, com o utilizador a escolher a mistura desejada.
- A codificação em ritmo variado dos blocos de dados, e uso eficiente da capacidade de *bits* com comutação e/ou mistura de faixas, para permitir todas as capacidades acima referidas num único portador.

Outros objectivos, funções e vantagens da invenção serão aparentes após a consideração das seguintes descrições detalhadas em conjunto com os desenhos, em que:

A figura 1 mostra um sistema do estado da arte e tipifica a falta de flexibilidade na, e o pobre desempenho de, sistemas de *media* actualmente disponíveis;

A figura 2 mostra o enquadramento ilustrativo da invenção;

A figura 3 é uma tabela que lista os campos na porção de introdução da faixa de dados digitais de um disco óptico que pode ser executado no sistema da figura 2;

A figura 4 é uma tabela semelhante que lista os campos em cada um dos blocos de dados que seguem a secção de introdução da faixa da figura 3;

As figuras 5A-5E compreendem um gráfico de operação que ilustra o processamento pelo sistema da figura 2 dos dados contidos na secção de introdução da faixa de um disco óptico que esteja a ser tocado.

A figura 6 é um gráfico de operação que ilustra o processamento de blocos de dados, no formato apresentado na figura 4, que segue à secção de introdução da faixa;

A figura 7A é um diagrama de estado e legenda que caracteriza a forma como o aparelho da invenção lê apenas aqueles blocos de dados numa faixa de um disco que são requeridos para a execução de uma versão seleccionada de um filme ou outra apresentação de vídeo, e a figura 7B mostra a forma como uma de duas versões alternativas podem ser tocadas seguindo as regras ilustradas pelo diagrama de estado na figura 7A;

A figura 8 mostra simbolicamente uma técnica do estado da arte utilizada na compressão da representação digital de um sinal de vídeo; e

A figura 9 ilustra as relações entre três diferentes relações de

aspectos de imagens.

O estado da arte

As limitações do actual estado da arte são exemplificadas pelo sistema na figura 1. Tal sistema está presentemente disponível para executar uma única fonte de material de programa, usualmente uma cassette de vídeo VHS, para gerar um sinal de vídeo em conformidade com a selecção de um de entre múltiplos padrões. Um sistema deste tipo é referido como sendo um gravador de cassetes de vídeo (VCR) de múltiplos padrões, apesar de serem mostrados no desenho componentes independentes. Tipicamente, uma fita VHS 7 tem gravado em si um sinal (analógico) NTSC de vídeo, e a fita é tocada num gravador de cassetes de vídeo VHS 5. O sinal analógico é convertido em forma digital num conversor analógico-digital A/D 9, e as representações digitais de sucessivas imagens são escritas para o armazém 11 de imagens de vídeo. O circuito 13 apaga depois as imagens em excesso, ou estima e adiciona imagens adicionais, necessárias para estar em conformidade com o padrão seleccionado, por exemplo, PAL. Para converter de um padrão para outro, é geralmente necessário mudar o número de linhas horizontais num campo ou imagem (gradação de imagem). Isto é usualmente conseguido deixando cair algumas linhas e/ou repetindo algumas ou tirando a média de sucessivas linhas para derivar uma nova linha a ser inserida entre estas. A função principal do circuito 13, naturalmente, é o de converter uma representação de uma imagem digital na sua forma analógica como saída do vídeo.

Sistemas deste tipo mostrados na figura 1 degradam geralmente a saída do vídeo. Cassetes de vídeo convencionais fornecem uma qualidade de vídeo reduzida quando estas suportam mais do que um padrão de vídeo. Uma razão é porque existe uma dupla conversão de analógico para digital e vice-versa.

Outra é porque a graduação de imagens é usualmente executada de uma forma tosca (apagando linhas, repetindo linhas e tomando a média das linhas). Existem formas conhecidas, todavia, de executar a graduação de imagens no domínio digital sem degradar a imagem. Embora não seja geralmente utilizada, a técnica está no estado da arte e será resumidamente descrita uma vez que também é utilizada no enquadramento ilustrativo da invenção.

Para dar um exemplo concreto, o padrão PAL tem 625 linhas por imagem, enquanto que o padrão NTSC tem 525 linhas por imagem. Em virtude de nenhuma parte da imagem ser formada durante o retraçar vertical, nem todos os varrimentos das linhas horizontais em qualquer dos padrões são utilizáveis para a representação da informação da imagem. No padrão PAL existem normalmente 576 linhas por imagem com informação da imagem, e numa imagem de NTSC existem normalmente 483 linhas com informação da imagem.

Para converter de um padrão para outro, são primeiro desentrelaçados sucessivos campos. Depois 576 linhas são convertidas em 483, ou vice-versa, e entrelaçadas novamente. Como isto é feito é fácil de visualizar conceptualmente. Considere-se, por exemplo, um fatia vertical muito fina através de uma imagem PAL. A fatia é dividida na suas três componentes de cores. A graduação da imagem para converter de PAL para NTSC, de um ponto de vista conceptual, não é mais do que o desenhar de uma curva baseada em 576 pedaços PAL de dados de cor e depois o dividir da curva em 483 partes para derivar um pedaço de dados para cada linha horizontal do desejado sinal NTSC. Na realidade, isto é conseguido através de um processo de interpolação, e é feito digitalmente. (A graduação da imagem, em geral, pode também envolver uma alteração da relação de aspecto, por exemplo, ao ir de HDTV para NTSC, e pode requerer o tosquiar de informação em ambos os lados de cada linha horizontal.).

Embora os sistemas do estado da arte proporcionem de facto a conversão de padrões, este é quase o limite da sua flexibilidade. O sistema na figura 2, por outro lado, oferece uma flexibilidade sem precedentes em formas ainda não contempladas no estado da arte.

O sistema ilustrativo da invenção

O sistema da figura 2 inclui uma *drive* de disquetes 21 para executar um disco óptico 23. Dados digitais armazenados no disco aparecem no condutor 25 de SAÍDA DE DADOS. O funcionamento da *drive* do disco é governado pelo microprocessador 27 controlador da *drive* do disco. A cabeça de leitura é posicionada por comandos emitidos através do cabo condutor 29 do CONTROLO DA POSIÇÃO DA CABEÇA, e a velocidade da rotação do disco é governada por comandos emitidos através do condutor 31 de CONTROLO DO RITMO. Discos ópticos são usualmente conduzidos a velocidade linear constante ou a velocidade angular constante. (Outra possibilidade envolve o uso de um número discreto de velocidades angulares constantes.) Discos da invenção pode ser conduzidos a velocidade linear constante de forma a que o comprimento linear da faixa tomada por cada *bit* é o mesmo quer um *bit* seja gravado numa porção interna ou externa da faixa. Isto permite o armazenamento da maioria dos dados. Uma velocidade linear constante requer que o ritmo de rotação do disco diminua quando as faixas externas estão a ser lidas. Este tipo de controlo de disco óptico é convencional. Por exemplo, o padrão áudio CD também requer discos que são rodados a um ritmo linear constante.

O microprocessador 41 é o controlador mestre do sistema. Como tal, este emite comandos ao controlador da *drive* do disco através dos condutor 43 e determina o estado do controlador da *drive* do disco através dos condutor 45. O controlador da *drive* do disco é proporcionado com duas outras entradas. O

analisador 47 de número / apontador de bloco emite comandos ao controlador da *drive* do disco através dos condutor 49, e o condutor 51 de *BUFFER CHEIO* estende um sinal de controlo do *gate* OR 54 para o controlador da *drive* do disco. Estas duas entradas serão descritas mais abaixo. (Em geral, embora seja feita referência a condutores individuais, deverá ser entendido que neste contexto alguns destes condutores são na realidade cabos para estender os *bits* em paralelo. Por exemplo, enquanto que a saída do *gate* OR 54 pode ser estendida para o controlador da *drive* do disco através de um único condutor 51, o analisador 47 de número / apontador de bloco poderá ser ligado ao controlador da *drive* do disco através de um cabo 49 para que dados de múltiplos *bits* possam ser enviados em paralelo em vez de em série).

Uma característica importante do sistema na figura 2 é o de a informação de *bit* ser armazenada no disco a um ritmo que varia de acordo com a complexidade do material codificado. Quer com isto dizer que não são o número de *bits* por segundo que realmente aparecem no condutor 25 de SAÍDA DE DADOS que variam, mas antes que o número de *bits* que são utilizados por segundo é que variam. Informação de vídeo é armazenada em forma digital comprimida. A figura 8 mostra a maneira como as imagens de vídeo são codificadas de acordo com os padrões MPEG1 e MPEG2. Uma imagem I independente é codificado na sua totalidade. Imagens previstas ou imagens P são imagens que são previstas com base nas imagens independentes precedentes, e a informação digital que é realmente requerida para uma imagem P representa simplesmente a diferença entre a imagem efectiva e a sua previsão. Imagens B bidireccionais previstas são imagens que são previstas a partir de imagens I e/ou P, com a informação requerida para uma tal imagem representando mais uma vez a diferença entre a forma efectiva e a sua previsão. (Conforme poderá ser apreciado, as funções de avanço rápido e de retrocesso rápido, se desejadas, são melhor implementadas utilizando imagens I.) O número de *bits* requeridos para

representar qualquer imagem depende não só no seu tipo, mas também na informação visual efectiva que é para ser representada. Obviamente, isto requer muito menos *bits* para representar um céu azul do que requer para representar um campo de flores. Os padrões MPEG são desenvolvidos para permitir a codificação de imagens com um número mínimo de *bits*. A informação das imagens é requerida a um ritmo constante. Por exemplo, se um filme é representado em forma digital no disco, 24 imagens serão representadas para cada segundo de filmagem. O número de *bits* requeridos para uma imagem difere radicalmente de imagem para imagem. Uma vez que as imagens são processadas a um ritmo constante, é aparente que o número de *bits* que são processados (usados) por segundo podem variar desde valores muito baixos até valores muito altos. Assim quando *bits* são realmente lidos do disco, embora possam ser lidos do disco a um ritmo constante, estes não são necessariamente processados a um ritmo constante.

Aplicam-se considerações similares a qualquer áudio armazenado no disco. Qualquer bloco de dados poderá conter a informação de *bit* requerida para um número variável de imagens. Qualquer bloco de dados pode conter de forma semelhante a informação de *bit* requerida para um período de tempo variável de um número variável de até numerosas faixas de áudio. (Existe apenas uma faixa física. A referência a múltiplas faixas de áudio é para diferentes séries de fatias de divisão de tempo (*time-division*) contendo os respectivos materiais áudio.) As faixas de áudio contêm informação digital, que poderá também estar em forma comprimida. Isto significa que se houver informação armazenada em qualquer bloco de dados para uma faixa de áudio em particular, aqueles *bits* não representam necessariamente o mesmo período de tempo. Poder-se-á pensar que a duração do som gravado para qualquer faixa de áudio correspondente a qualquer imagem representada num bloco seria a duração das imagens. Todavia, isto não é necessariamente verdade. Isto significa que informação de áudio pode ser lida antes desta ser realmente necessária, com a leitura de mais informação de

áudio a poder fazer uma pausa quando uma quantidade suficiente tiver sido acumulada ou com a não inclusão do áudio em alguns blocos de dados para compensar pelo sobre fornecimento precedente. Isto leva ao conceito de *buffering*, a função dos *buffers* de áudio 53, o *buffer* de vídeo 55, o *buffer* de *pan scan* 57, o *buffer* de subtítulos 59, e o *gate* OR 54 que gera o sinal *BUFFER CHEIO*.

Conforme cada bloco de dados é lido do disco, este passa pelo *gate* 61, desde de que o *gate* esteja aberto, e os campos de *bits* são distribuídos pelo multiplexador 63 para os vários *buffers* e, através da linha COMANDO / DADOS 65, para o controlador mestre 41. Cada bloco de dados no enquadramento ilustrativo da invenção contém informação de *bit* de vídeo correspondente a um número variável de imagens. Conforme acima referido, pode haver um grande número de *bits*, ou um número pequeno, ou mesmo até nenhum *bit* (por exemplo, se o disco em particular que está a ser tocado não representa qualquer vídeo). Sucessivos grupos de dados de vídeo são armazenados no *buffer* de vídeo 55 separados por marcadores. O decodificador de vídeo 67 emite um comando através do condutor 69 quando este quer ser fornecido com uma nova fornada de dados através do condutor 71. Comandos são emitidos a um ritmo estável, embora o número de *bits* fornecidos em resposta variam de acordo com o número de *bits* requeridos para as imagens em particular que estão a ser processadas. O ritmo a que os *bits* estão a ser lidos da *drive* do disco é suficientemente elevada para acomodar as imagens que necessitam de informação máxima, mas a maioria das imagens não necessitam. Isto significa que o ritmo a que os blocos de dados são realmente lidos é superior ao ritmo a que são utilizados. Isto não significa, todavia, que um sistema bem desenvolvido deverá atrasar a leitura de um bloco de dados até que os dados sejam realmente requeridos para processamento. Para já, quando os dados são realmente requeridos, a cabeça de leitura pode não estar posicionada no início do desejado bloco de dados. É por esta razão que o

buffering é proporcionado. O *buffer* de vídeo 55 contém a informação de *bit* para um número de sucessivas imagens (o número efectivo dependendo do ritmo a que os *bits* são lidos, o ritmo a que as imagens são processadas, etc. conforme é sabido no estado da arte), e a informação de bloco de dados de vídeo é lida do *buffer* de vídeo a um ritmo constante de imagens determinado pelo decodificador 67 de vídeo. Dados de vídeo são fornecidos ao *buffer* apenas até que o *buffer* esteja cheio. Uma vez cheio o *buffer* não deverá ser enviada mais informação porque esta não poderá ser armazenada. Quando o *buffer* de vídeo estiver cheio, um sinal no condutor 69 leva a que a saída do *gate* OR 54 tenha um valor elevado para informar ao controlador 27 do *drive* do disco que um dos *buffers* está cheio.

Comentários semelhantes aplicam-se aos outros três tipos de *buffers*. (Existe um único *buffer* de subtítulos 59, um único *buffer* de *pan scan* 57, e numerosos *buffers* de áudio 53, as funções de cada um serão abaixo descritas). Quando qualquer destes *buffers* estiver cheio, a sua saída correspondente leva a que o *gate* OR 54 controle o condutor do *BUFFER CHEIO* para um valor elevado para assim informar ao controlador da *drive* do disco que um dos *buffers* está cheio. Os *buffers* de áudio 53 e o *buffer* de subtítulos 59 funcionam de uma forma comparável àquela descrita para o *buffer* de vídeo 55. O decodificador 71 do processador de áudio emite um comando ao *buffer* de áudio quando este requer dados da faixa de áudio, altura em que os *buffers* de áudio fornecem tais dados. Deste forma semelhante, o gerador de gráficos 73 extrai dados do *buffer* de subtítulos 59, e o processador / graduador vertical 87 de *pan scan* recebe dados do *buffer* de *pan scan* 57 conforme será abaixo descrito.

Quando qualquer um dos quatro *buffers* estiver cheio (que inclui qualquer um dos *buffers* individuais dentro do bloco 53), o controlador da *drive* do disco 27 leva a que o *drive* do disco pare de ler dados. Os dados não voltam a

ser lidos até que todos os *buffers* possam aceitá-los, isto é até que nenhum *buffer* esteja cheio e o condutor 51 tenha um valor baixo. (Dum modo oposto, se os *buffers* estiverem a ser esvaziados de dados rapidamente demais, um ajuste no sinal de RITMO DE CONTROLO no condutor 31 aumenta a velocidade do disco e conseqüentemente o ritmo a que os *buffers* são enchidos.)

Esta discussão sobre *buffering* surgiu da consideração da entrada 51 do *BUFFER CHEIO* para o controlador da *drive* do disco 27. A outra entrada que fica por descrever é aquela representada pelo cabo 49. Conforme será descrito mais abaixo, cada bloco de dados tem um número serial do bloco assim como informação do apontador no seu início. O circuito 47 lê o número serial do bloco e analisa a informação do apontador. O apontador, que é um número serial do bloco, aponta para o bloco de dados que deverá ser lido de seguida. Esta informação é fornecida ao controlador da *drive* do disco através do cabo 49. É desta forma que o controlador da *drive* do disco pode controlar o posicionamento da cabeça de leitura do *drive* do disco para que possa aceder ao bloco de dados desejado. Muitas vezes será lido o bloco errado -- isto é de se esperar no caso de haver um salto para um novo bloco, como é o caso, por exemplo, quando é dado um salto de uma faixa para outra quando se toca um disco áudio de CD. Se o *drive* do disco lê um bloco de dados cujo número serial do bloco é muito elevado ou muito baixo, isto é determinado pelo analisador 47 do número / apontador do bloco que depois emite um novo comando através do cabo 49 para o controlador da *drive* do disco para o levar a ler outro bloco com um número serial do bloco mais baixo ou mais elevado respectivamente. Durante o tempo que a cabeça de leitura se está a posicionar para ler um novo bloco, os dados que são lidos não são na realidade utilizados. O *gate* 61 permanece fechado para que a informação não seja entregue ao desmultiplexador 63 para distribuição aos quatro *buffers* e ao controlador mestre 41 através do cabo condutor de COMANDO / DADOS. É apenas quando chegar ao bloco de dados correcto, conforme determinado pelo

circuito 47 analisando o número serial do bloco no início do bloco, que o condutor 75 é impulsionado para um valor elevado para abrir o *gate* 61.

O remanescente do bloco é então entregue ao desmultiplexador. Os *bits* de dados lidos do disco são também entregues ao controlador mestre 41 do microprocessador através do condutor 77. Cada bloco de dados contém não só informação de *bit* que tem de ser distribuída aos vários *buffers*, mas também informação de controlo, por exemplo, *bits* que identificam o tipo de dados realmente encontrados no bloco. Os *bits* de identificação (bandeiras e semelhantes, conforme será descrito mais abaixo) são fornecidos ao controlador mestre para que esteja em controlo do sistema em todos os momentos. Os *bits* de identificação são utilizados pelo desmultiplexador para controlar a distribuição de dados aos vários *buffers*. (O controlador mestre emite comandos através do condutor 76 para o analisador 47 de número / apontador de bloco que exerce não só um controlo geral sobre este elemento, como também controlo específico levando o elemento 47 a desligar o sinal activo no condutor 75 conforme for apropriado para prevenir que blocos de dados completos entrem no desmultiplexador se estes não forem requeridos para processamento subsequente.)

O controlador mestre está no coração do sistema e de facto executa a maioria do processamento que será descrito mais abaixo. O utilizador do aparelho comunica com o controlador mestre via o interface 79, tipicamente um teclado. Ao utilizador é também proporcionado um mecanismo de chave e fechadura, mostrado simbolicamente pelo número 81, que é aqui referido como a opção "fechadura parental". Se a fechadura estiver fechada, não serão mostrados filmes de classificação R. A forma como isto é controlado por *bits* representados na realidade no disco será descrito mais abaixo. Se a fechadura estiver fechada, e se apenas estiver no disco um filme de classificação R, um sinal de desactivação

no condutor 83 de CONTROLO DA FECHADURA PARENTAL fecha o *gate* 61. Nenhum dos *bits* de dados são transmitidos através do *gate* e o disco não poderá ser tocado. Conforme será aparente mais abaixo, se o disco também tiver uma versão do filme que não seja de classificação R, este será mostrado se for seleccionado pelo espectador. Embora a funcionalidade da fechadura parental é mostrada como requerendo o uso de uma chave e fechadura física, é de se entender que esta funcionalidade pode ser implementada requerendo entradas no teclado conhecidas apenas pelos pais das crianças. A forma de informar o controlador mestre que as versões de classificação R de um filme não são para serem vistas não é restrita a qualquer forma específica. Assim como chaves físicas e chaves codificadas são utilizadas alternadamente para controlar o acesso a um computador, assim estas podem ser no sistema na figura 2. O que é importante é a forma como duas versões diferentes podem ser representadas no mesmo disco (sem requer uma versão completa de cada uma), e como o sistema determina se uma versão seleccionada pode à partida ser vista. Isto será descrito mais abaixo.

O controlador mestre 41 inclui várias outras saídas que ainda não foram descritas aqui. O condutor 85 representa um *bus* de RELÓGIO MESTRE que é estendido a todos os sub-sistemas mostrados na figura 2. Em qualquer sistema digital, é requerido um sinal relógio mestre para controlar o faseamento correcto dos vários circuitos. As outras seis saídas do controlador mestre são estendidas ao desmultiplexador 63, ao descodificador 71 do processador de áudio, ao processador / graduador vertical 87 de *pan scan*, ao armazém de imagens de vídeo, ao circuito de entrelaçamento e de *pulldown* 3:2 89, ao gerador de gráficos 73, e ao gerador de sincronização e conversor 92 de DVA. Estes são cabos condutores de controlo para gerir as operações dos blocos dos circuitos individuais.

O descodificador 71 do processador de áudio processa os dados nos

Wm. Suter Cumming

buffers 53 e obtém sinais analógicos individuais de áudio que são estendidos a um sistema de amplificador / altifalante mostrado simbolicamente pelo número 91. O decodificador de vídeo 67 obtém um sinal de VÍDEO DIGITAL no condutor 93 dos dados de vídeo comprimidos que são lidos do *buffer* 55. O vídeo digital é fornecido ao processador / graduador vertical 87 de *pan scan* imagem por imagem. A codificação / descodificação de um vídeo em particular que é empregue não é uma característica da presente invenção. Um padrão preferido seria na linha do MPEG1 e MPEG2, mas estes são apenas ilustrativos. O mesmo é verdade para a codificação da faixa de áudio. A presente invenção não é limitada a métodos de codificação específicos.

O funcionamento de circuitos 57 e 87 podem melhor ser compreendidos considerando primeiro o desenho simbólico na figura 9. A informação digital que é armazenada no disco óptico no enquadramento preferido da invenção caracteriza imagens que tenham uma relação de aspecto "mestre" de 16:9, a chamada imagem de "ecran largo". A relação de aspecto mestre é mostrado no canto superior esquerdo na figura 9. Se o sinal analógico final a ser mostrado no receptor de televisão do utilizador requer esta relação de aspecto, e o número de linhas horizontais de varrimento com informação de imagem (em oposição a linhas horizontais de varrimento que ocorrem durante a reconstituição vertical) corresponde ao número de linhas horizontais representadas pela informação *bit* de vídeo armazenada no disco, então o gerar do sinal analógico de vídeo é linear. Mas se o receptor de televisão do utilizador acomodar um sinal de TV tendo uma relação de aspecto de 4:3, e a relação de aspecto mestre no disco for de 16:9 em vez de 4:3, então existem duas escolhas. Uma é o de mostrar a imagem original em forma de "caixa de correio". Conforme apresentado no lado direito da figura 9, o que é feito neste caso é uma compressão vertical uniforme de uma imagem mestre para que a sua dimensão horizontal caiba nos limites do receptor de televisão. Isto resulta na dimensão vertical sendo encurtada ao mesmo

tempo para que esta encha menos do que a totalidade da altura da área do ecran da TV. O que isto significa é que os varrimentos das linhas horizontais em cima e em baixo de cada imagem em geral têm de ser postos sem dados, formando bandas escuras nos seus lugares -- mas a relação de aspecto original é preservado. A outra opção é para um *pan scan* de relação de aspecto reduzido. O que isto envolve é a sobreposição de uma caixa tendo uma relação de aspecto de 4:3 na imagem original de ecran largo. Como resultado, o lado esquerdo desta imagem, o lado direito, ou ambos os lados, são tosquiados. (Em todos os casos, mesmo que uma imagem de ecran largo correspondente a uma relação de aspecto mestre 16:9 é para ser mostrada, poderá ser necessário efectuar um número de varrimentos das linhas horizontais o que é diferente do número de linhas horizontais representados no disco. O número de linhas horizontais é uma função do padrão do sinal de vídeo ao qual a saída do vídeo tem de se conformar. Alterando o número de linhas é um processo conhecido como gradação vertical, conforme acima descrito.)

No que diz respeito ao processamento *pan scan*, será aparente da figura 9 que a fim de identificar aquela porção de uma imagem com relação de aspecto mestre 16:9 que deverá ser utilizada para formar uma imagem de *pan scan* de relação de aspecto reduzido, tudo o que é requerido é o especificar do ponto de partida ao longo de cada varrimento da linha horizontal da informação que deverá ser utilizada. Especificando um único número (por exemplo, coluna 200 de entre um total de 960 colunas) é suficiente para esta finalidade. A questão, todavia, é se a mesma coluna é sempre utilizada. Nalguns casos pode ser dito ao gravador que se for desejada uma relação de aspecto 4:3, este deverá sempre ser tirado do centro da imagem de ecran largo. Noutros casos, um ponto de partida de coluna variável pode ser desejado, e neste caso um bloco de dados contém realmente informação que representa o número da coluna inicial que deverá ser utilizada desde aquele ponto até que outra alteração seja efectuada.

Conforme será aparente mais abaixo, a informação de vídeo em cada bloco de dados inclui uma bandeira que representa se a informação de coluna do *pan scan* deverá ser actualizada. Se existir tal bandeira, o descodificador 67 de vídeo emite um comando através do condutor 95 para o *buffer* de *pan scan* 57. Nesta altura o *buffer* aceita uma actualização de *pan scan* vinda do desmultiplexador 63. Aquela actualização permanece no *buffer*, para uso pelo processador / graduador vertical 87 de *pan scan* com as imagens seguintes, até que outra alteração tenha lugar.

É no processador / graduador vertical 87 de *pan scan* que o número de linhas horizontais é ajustado e é alterada a relação de aspecto. O vídeo digital é fornecido pelo descodificador 67 de vídeo e a informação de *pan scan*, se esta for requerida, é providenciada pelo *buffer* 57. A saída do circuito 87 consiste em vídeo digital não comprimido, na relação de aspecto desejado e representado pelo número de linhas horizontais requeridas para o padrão de televisão seleccionado.

Uma vez armazenada em forma digital a informação da imagem vídeo no armazém de imagem 89, esta pode ser repartida em campos entrelaçados se o padrão seleccionado o requer. Também, o *pull-down* 3:2 é a técnica utilizada para converter filmes de 24-imagens-por-segundo em vídeo de 60-imagens-por-segundo (os valores nominais de 24 e 60 são na realidade 23,97 e 59,94); para converter dados representativos de um filme para um formato NTSC, a informação (bloco de dados) da imagem tem de ser lida a um ritmo de 24 por segundo. (Conforme é padrão no actual estado da arte, tal transformação aplica a imagem 1 do material da fonte aos campos 1, 2 e 3 do sinal de vídeo, imagem 2 do material da fonte aos campos 4 e 5 do sinal de vídeo, imagem 3 do material da fonte aos campos 6, 7 e 8, etc., assim produzindo 60 campos para 24 imagens de origem.) Por outro lado, a conversão para o padrão PAL é relativamente simples,

e o *pull-down* 3:2 não é requerido. O padrão PAL requer 50 campos por segundo. As imagens são processadas a um ritmo de 25 por segundo, e todas as imagens são utilizadas para formar dois campos. (Devido ao facto de os filmes serem gravados a um ritmo de 24 imagens por segundo embora sejam processados a um ritmo de 25 por segundo quando convertidos para PAL, tudo o que acontece num ecrã de TV ocorre 4% mais rápido na Europa do que nos Estados Unidos.) Se as imagens são processadas a um ritmo de 25 por segundo ou de 24 por segundo é controlado alterando a frequência do sinal do RELÓGIO MESTRE no condutor 85.

A saída do bloco 89 é digital, e é estendida ao gerador de sincronização e ao conversor digital-analógico 92. É neste elemento que impulsos apropriados de sincronização são inseridos nos campos, e a informação digital é convertida em analógica. Quaisquer subtítulos que são requeridos são contidos no *buffer* 59. Sob o controlo do microprocessador 41, os comandos são emitidos através do cabo condutor 97 de controlo para o gerador de gráficos 73. Este circuito convencional extrai informação codificada dos caracteres do *buffer* de subtítulos, e gera um sinal de VÍDEO no condutor 99 que descreve os subtítulos. O sinal de accionamento (KEY) é gerado no condutor 98, e os dois sinais são estendidos a um circuito convencional 96 de accionamento. Este dispositivo funde os subtítulos com a imagem de vídeo (utilizando accionamento rígido ou linear conforme a opção do fabricante, como é conhecido no estado da arte), e estende o sinal de vídeo composto para um ecrã 94 de TV convencional.

Campos de introdução da faixa

Antes de proceder com a descrição do processamento em detalhe, será de grande ajuda considerar a informação que é armazenada na porção de introdução da faixa do disco. Esta informação é armazenada em campos

individuais conforme mostrado na figura 3, e é esta informação que controla o subsequente processamento dos dados lidos do disco. O formato do bloco de dados é mostrado na figura 4, mas para uma compreensão de como os dados neste bloco são utilizados, é necessário apreciar a informação de preparação que é lida primeira.

Fazendo referência à figura 3, no início da faixa existem um número de *bits* de sincronização de introdução (*lead-in*). Embora para todas as outras entradas são mostrados números de *bits* mínimos e máximos nas colunas apropriadas, nenhum destes números são proporcionados para o *bits* de sincronização de introdução. O número de *bits* de sincronização requeridos no início da faixa depende do *hardware* utilizado. Dado um *hardware* em particular e o âmbito das velocidades de disco envolvidas, são proporcionados um número suficiente de *bits* de sincronização no início da faixa para permitir aos circuitos envolvidos na leitura do disco, incluindo o controlador da *drive* do disco 27 e o analisador 47 de número / apontador de bloco, para se sincronizarem ao fluxo de *bits* no condutor 25 de SAÍDA DE DADOS. A sincronização de *bits* é uma técnica bem conhecida em sistemas digitais.

O segundo campo consiste em 40 *bits* que representam territórios autorizados. Existem várias formas como as editoras de *software* podem bloquear a execução do seu *software*. O mais importante envolve o controlar se filmes com classificação R podem ser vistos (a opção de bloqueio parental), e se o sinal analógico de vídeo final de saída pode assumir o padrão seleccionado pelo utilizador. É desta forma, por exemplo, que uma editora de *software* poderá permitir que um filme seja visto num receptor NTSC mas não num receptor PAL. Mas desde que o aparelho seja proporcionado com este tipo de controlo de bloqueio, este pode ser estendido a territórios. Todos os aparelhos utilizados com os discos da invenção conformem com o mesmo conjunto de especificações. Uma

característica do desenvolvimento é de que cada aparelho é proporcionado com uma representação do território ou territórios para os quais este foi entendido para venda. Por exemplo, o território ou territórios podem ser representados pelas configurações de um interruptor DIP, um código armazenado no ROM de um microprocessador (por exemplo, no controlador mestre 41) ou semelhante. É presumido que existem um total de 40 territórios possíveis. Cada disco tem um campo de 40 *bits* na sua secção de introdução, sendo cada um destes *bits* associado a um dos 40 territórios. Um 1 em qualquer posição do *bit* é uma indicação de que o disco está autorizado a ser executado no respectivo território, e um 0 é uma indicação de que o disco não está autorizado. O aparelho cujo código indica que este está para venda na China, por exemplo, não executará um disco se existir um 0 no campo de território dos 40 *bits* na posição associada com a China.

Como exemplo do uso de tal característica, considere-se um aparelho destinado para venda num país em particular. Uma editora de *software* poderá distribuir um filme que por razões contratuais não pode ser distribuído naquele país. É por esta razão que um 0 seria armazenado na posição do *bit* associado com aquele país no campo dos territórios autorizados da secção de introdução da faixa. Ao detectar este *bit*, o controlador mestre 41 levará o circuito 47 a gerar um sinal de inibição no condutor 75 que iria levar o *gate* 61 a bloquear a passagem de todos os dados.

O terceiro campo é um *bit* único, uma bandeira que indica se existe qualquer informação no campo que lhe segue. Esta informação é aqui designada "*software* especial". O aparelho da figura 2 executa normalmente o mesmo código de *software*, tipicamente contido em memória de apenas de leitura. É este código que será descrito em relação aos gráficos de operação do desenho. Todavia, uma vez que o aparelho é controlado por microprocessador, não existe

razão para que este não seja utilizado para um propósito totalmente não relacionado, e isto pode ser feito simplesmente carregando *software* do disco. Se a bandeira do *software* especial for um 1, então o controlador mestre 41 lê através do condutor 77 o *software* que se segue imediatamente no campo 4. Assim, dependendo de a bandeira do *software* especial ser um 0 ou um 1, o quarto campo ou é vazio ou contém um *software* de comprimento indeterminado. No final do *software* existe uma palavra de sincronização que é única no sentido em que esta palavra não é permitida existir noutra local no fluxo geral de dados. Quando o padrão da palavra de sincronização aparece, isto é uma indicação de que o campo de dados precedentes chegou ao fim, e que se segue um novo campo. (Na eventualidade de dados tendo o padrão da palavra de sincronização aparecerem no fluxo de dados e serem mal interpretados como sendo uma palavra de sincronização, isto pode ser evitado utilizando técnicas conhecidas. Por exemplo, se a palavra de sincronização consistir em 32 *bits* de um determinado padrão, e alguma sequência geral de dados incluir nela este padrão, então depois da gravação de 31 *bits* dos dados de padrão, um *bit* extra, tendo um valor oposto ao valor do último *bit* do padrão da palavra de sincronização, pode ser inserido no fluxo de *bits*. Quando o aparelho vê este *bit*, este descarta-o e trata o seguinte *bit* como sendo um *bit* de dados em vez do último *bit* da palavra de sincronização.)

Um exemplo de *software* especial pode ser o *software* para controlar jogos de vídeo. Enquanto o aparelho é proporcionado com um sistema de operação desenhado para passar filmes e áudio de múltiplas faixas, é perfeitamente exequível que o aparelho execute funções adicionais e/ou diferentes envolvidas na utilização de jogos de vídeo. Isto é especialmente verdade se o interface do utilizador for separável e *joysticks* e semelhantes possam ser ligados no lugar de um teclado para acomodar o equipamento periférico de jogos. O sistema pode ser convertido num aparelho de jogos de vídeo simplesmente através do armazenamento do *software* necessário conforme

este é lido de um disco. Enquanto que nos gráficos de operação que serão descritos mais abaixo o *software* especial é mostrado como sendo contido em si mesmo e não envolvendo os passos de processamento padrão, o *software* especial pode certamente chamar subrotinas do sistema de operação para a execução a fim de tirar vantagem do código incorporado.

O quinto campo consiste em 12 posições de *bits*, cada um correspondendo a um padrão diferente. Os padrões incluem European HDTV de 1250 linhas, Japanese HDTV de 1125 linhas, o proposto American HDTV de 1050 linhas, (assim como 1080 linhas e 787 linhas dos padrões propostos), PAL de 625 linhas, NTSC de 525 linhas, SECAM de 625 linhas, "caixa de correio" de 360 linhas, etc. Até é possível acomodar padrões futuros, embora para formar um sinal de vídeo apropriado em tal caso seria necessário *software* diferente. Todavia, isso implica simplesmente proporcionar *software* num disco para complementar o sistema operativo incorporado.

Como exemplo simples, se a posição do primeiro *bit* de um campo de 12 *bits* corresponde ao padrão NTSC, e se o utilizador seleccionar um padrão NTSC para executar no seu receptor de TV, ou se este é a sua configuração *default* (conforme será descrito mais abaixo), então um sinal NTSC será gerado apenas se o primeiro *bit* no campo de padrões autorizados for um 1.

O campo 6 contém sempre 100 *bits*. Estes *bits* representam idiomas de áudio representativos -- diálogo -- para um filme. É raro que sejam preparadas tantas versões do mesmo filme, e não é contemplado que tantas versões sejam realmente incluídas num disco. De facto, existem um máximo de 16 faixas de áudio que podem conter diálogo em idiomas diferentes. Cada um dos 100 *bits*, exceptuando o primeiro, representa um de 99 idiomas. Se houver um 1 na posição de *bit* correspondente, isto é uma indicação de que existe uma faixa de áudio com diálogo no idioma correspondente.

A primeira das posições dos 100 *bits* não corresponde realmente a um idioma. Em vez disso, um 1 na primeira posição de *bit* significa que existe uma faixa de “música e efeitos” (“*M&E*”). (Por “efeitos” significa tais coisas como o som associado à trovada, tiros de armas e semelhantes.) Conforme indicado no campo de Comentários na figura 3, existem N “1”s no campo 6 da secção de introdução da faixa em geral, onde N tem um valor máximo de 16 (uma faixa de *M&E* e até 15 faixas de diálogo, ou até 16 faixas com diálogo sem *M&E*). Como exemplo único, suponhamos que a terceira posição de *bit* corresponde a francês, o quinto a grego, e o campo de 100 *bits* é 10101000....0. Isto significa que existe uma faixa de *M&E*, assim como faixas de diálogo em francês e em grego. Isto não significa que cada bloco de dados no disco inclui informação de *bit* que representa *M&E*, e diálogo em francês e em grego. O que isto de facto significa é que qualquer bloco de dados tem no máximo três faixas de áudio com *M&E* e/ou diálogo. Isto também significa que qualquer bloco de dados que tenha tal informação de faixa de áudio contém a informação na ordem *M&E*, francês, grego. Exactamente como o sistema determina que blocos de dados específicos contêm a informação áudio para aqueles idiomas representados no campo de 100 *bits* será descrito mais abaixo em relação aos campos contidos num bloco de dados.

Deverá ser entendido que as faixas de áudio de idioma não incluem necessariamente apenas diálogo. Conforme será descrito em breve, é possível misturar uma faixa de *M&E* com uma faixa de diálogo em francês, com o resultado sendo uma faixa de áudio completa adequada para executar em França. Mas é certamente possível que uma faixa de áudio em particular incluirá *M&E* pré-misturada e diálogo original. Por exemplo, se a posição do *bit* 10 do campo de 100 *bits* representar diálogo em inglês e existir um 1 lá armazenado, isto significa que existe uma versão de áudio no idioma inglês no disco. Todavia, é

possível que na correspondente faixa de áudio exista não só diálogo em inglês, mas uma banda sonora completa incluindo a M&E. Ao mesmo tempo, pode existir M&E numa faixa separada, se existir um 1 na primeira posição do campo de 100 *bits*. Como as várias faixas são processadas a fim de obter uma banda sonora completa para executar em qualquer dado idioma depende de informação subsequente. O campo 6 representa simplesmente quais os idiomas de áudio que estão disponíveis, assim como se existe uma faixa separada de M&E (sem qualquer diálogo).

Existe outro bocado de informação que é necessário para que o esquema áudio possa funcionar, e essa informação é representada no campo 7. Para cada uma das N faixas disponíveis para idiomas de áudio (até a um máximo de 16), existe um código de 3 *bits* no sétimo campo. Antes de descrever o significado dos códigos, tem de ser entendido como os códigos estão associados a uma faixa e idioma em particular. Suponhamos que o campo 6 é 101010000100...0 que é interpretado a significar que existe uma faixa de M&E, uma faixa em francês, uma faixa em grego e uma faixa em inglês. Desta informação por si, não há maneira de dizer se existe se quer alguma M&E na faixa francesa, grega ou inglesa. A única coisa que é sabida em termos de idiomas é que existe diálogo em apenas três idiomas. Para este exemplo, haveriam 12 *bits* no campo 7. Os primeiros três *bits* estão associados à faixa de M&E, os segundos três *bits* estão associados à faixa em francês, e os terceiros e quartos três *bits* estão associados à faixa em grego e em inglês respectivamente. Os códigos de três *bits* são como se seguem:

- 000 -- misturador mestre (M&E)
- 001 -- comutador mestre (M&E)
- 010 -- diálogo + (M&E), faixa de áudio completa
- 011 -- faixa a ser misturada com o misturador mestre
- 100 -- faixa a ser comutada com o comutador mestre

Estes cinco códigos são tudo o que é necessário para formar bandas sonoras completas em três idiomas disponíveis, francês, grego e inglês. Como as faixas são combinadas será descrito mais abaixo, mas o que tem de ser lembrado é que a razão de ser da totalidade do arranjo é o de proporcionar bandas sonoras em muitos idiomas (até 15), sem necessitar de aquilo que poderá ser uma gravação de duas horas para cada. De facto, se um filme tem uma duração de duas horas, mas o diálogo em si é de apenas 30 minutos, o objectivo é de gravar uma faixa completa (M&E ou banda sonora original), com apenas uma gravação de áudio de 30 minutos de diálogo para um idioma em particular.

O campo 8 contém $N \times 4$ bits, isto é, 4 bits para cada um dos N "1"s no campo 6. Existe assim um código de 4 bits no campo 8 para cada faixa de idioma de áudio que está disponível no disco. O código de 4 bits representa o tipo de faixa, e existe um máximo de 16 possibilidades. Tipos de faixas típicas são mono de canal único, *Dolby* de dois canais, Musicam de 5,1 canais ("*5.1-channel*"), etc. [O termo ("*5.1-channel*") refere-se a canais da esquerda, direita, centro, esquerda de trás e direita de trás em conjunto com um canal de *sub-woofer*.] Os códigos do tipo de faixa de 4 bits permitem ao controlador mestre determinar a forma como o descodificador 71 do processador de áudio opera nos dados das de-até-16 faixas de áudio para obter saídas analógicas para o sistema de altifalantes 91.

Considerando novamente o campo 7, existem várias formas como uma banda sonora completa, num idioma seleccionado, pode ser obtida de um disco. A operação de mistura envolve a mistura (o somar) de duas bandas sonoras. A operação de comutação envolve a comutação entre duas bandas sonoras, e a execução de uma delas em qualquer dado momento. A primeira faixa é sempre de M&F, se esta estiver disponível. O código para esta faixa é sempre 000 ou 001. Se o código for 000, isto significa que não existe diálogo na faixa e a

sua M&E é para ser misturado com a faixa do idioma seleccionado. Se o código 001 estiver associado com a faixa em francês, por exemplo, isto significa que a primeira e a terceira faixa devem sempre ser misturadas. O diálogo, quando este existe, aparece na faixa em francês, e misturando-a com o misturador mestre proporciona uma banda sonora completa em francês. Por outro lado, a primeira faixa pode ser um comutador mestre. O que isto significa é que música e efeitos são gravados nesta faixa, com ou sem diálogo. A faixa francesa neste caso seria representada pelo código 100. Este contém M&E e diálogo, mas apenas quando há diálogo. A faixa de M&E, a primeira, é tocada sozinha quando não há qualquer diálogo, mas a quinta faixa é tocada sozinha quando existe diálogo. As faixas são comutadas e não misturadas. A faixa francesa, quando diálogo é gravado nela, inclui não só diálogo mas também de M&E uma vez que isto seria a única fonte de M&E numa operação do tipo comutação.

A quinta possibilidade (010) é que uma faixa em particular contém por acaso a banda sonora original, M&E em conjunto com diálogo no idioma original. Se o diálogo estiver no idioma seleccionado, a faixa pode ser tocada do início ao fim, sozinha. Esta faixa pode também servir como um comutador mestre (código 001) para outros idiomas.

Quando se trata de misturar faixas, quaisquer áudios que estejam nas duas faixas especificadas (o misturador mestre e a faixa que é misturada com este) são simplesmente adicionadas em conjunto em todos os tempos; qualquer áudio que exista nas duas faixas será tocado. É apenas quando comutando entre o comutador mestre e a faixa com a qual esta está comutada que uma faixa é tocada em vez da outra. É verdade que cada faixa pode conter informação áudio apenas quando a outra não a contém (o que permitiria a mistura), mas é concebível que o comutador mestre também incluirá diálogo, isto é, se esta for uma gravação da banda sonora original do filme. É por isso que é empregue a comutação -- apenas

uma faixa é ouvida em qualquer dado momento. Conforme será descrito mais abaixo, cada bloco de dados inclui *bits* que informam ao controlador mestre quais faixas de áudio contêm na realidade dados naquele bloco. Se uma faixa de idioma de áudio seleccionada com um código original de faixa 100 tem dados em qualquer bloco de dados, então o descodificador 71 do processador de áudio processa os dados naquela faixa de áudio excluindo quaisquer dados que possam estar na faixa do comutador mestre.

O campo 9 na figura 3 contém seis *bits* que são codificados para representar um número M. Isto é o número de "outras" faixas de áudio, separadas e para além das até-16 faixas de idioma de áudio. A utilização usual destas faixas é a de representar, em forma digital comprimida, instrumentos individuais ou misturas de instrumentos, tendo o utilizador a opção de as combinar. Numa forma extrema, poderia haver 63 faixas de instrumentos, sendo o utilizador capaz de combinar qualquer faixa que este desejar, e de fixar os seus níveis relativos antes de fazer a mistura. Se uma das faixas contém à partida o som combinado, é possível apagar qualquer instrumento da mistura orquestral especificando que o conteúdo da sua informação deverá ser apagada, ou subtraída, da mistura orquestral. Isto iria permitir ao utilizador, por exemplo, tocar o seu piano acompanhado por uma orquestra a tocar um concerto da qual o som do piano terá sido eliminado. Iria também permitir a um utilizador de separar um instrumento em particular para facilitar o treino. O que o utilizador faz precisamente com as "outras" faixas de áudio é determinado pelas selecções de menu que lhe estão disponíveis. O campo 8 identifica simplesmente quantas "outras" faixas de áudio estão presentes no disco. (O termo "outras" faixas de áudio parece ser tanto ou quanto pouco descritivo, mas isto não é o caso. A intenção é de que o termo possa subsumir qualquer uso da faixa de áudio que não seja o proporcionar de faixas de áudio para filmes. Em vez de ter música orquestral nestas "outras" faixas de áudio, por exemplo, é possível ter vocalistas individuais, permitido a

um utilizador estudar diferentes harmonizações.).

É aparente que se existem de facto 63 “outras” faixas de áudio, então muita se não toda a capacidade do disco pode ser atribuída a dados de áudio. Mas esta é precisamente a razão porque são disponibilizadas tantas faixas de áudio. É certamente contemplado que alguns discos executáveis no sistema da figura 2 não incluirão vídeo. De facto, o campo 19, a ser descrito mais abaixo, é um campo de 1 *bit* que informa ao controlador mestre se existe quaisquer dados de vídeo no disco.

Uma vez determinado que existem M “outras” faixas de áudio, o campo seguinte especifica como cada faixa está codificada. À semelhança do caso do campo 8, um código de 4 *bits* é utilizado para cada uma das “outras” faixas de áudio. Assim o número de *bits* no campo 10 pode ser tão baixo como 0 (se não houverem “outras” faixas de áudio) ou tão elevado como 252 (63x4).

Enquanto o aparelho pode determinar pela leitura dos campos 9 e 10 quantas “outras” faixas de áudio existem, o utilizador tem de ser dito o que está nestas faixas a fim de este saber o que fazer com elas. Existe uma descrição de cada faixa, e está em múltiplos idiomas. A primeira coisa que o utilizador tem de ser dado é uma lista dos idiomas em que existem descrições das “outras” faixas de áudio. Um campo de 100 *bits* é utilizado para esta finalidade. Conforme indicado na figura 3, o campo 11 tem 100 *bits*. Um 1 em qualquer posição de *bit* é uma indicação de que estão disponíveis definições de faixas no respectivo idioma. A correspondência entre posições de *bits* e idiomas é a mesma no campo 11 como no campo 6. Será de lembrar que a primeira posição de *bit* no campo 6 corresponde à M&E, não a um idioma tradicional. A primeira posição de *bit* no campo 11 é assim não utilizada, e podem estar no máximo 99 “1”s no campo 11.

Antes das definições das faixas serem realmente lidas e processadas, o aparelho tem de determinar quais as opções de menu a proporcionar ao utilizador. Suponhamos, por exemplo, que existem dez “outras” faixas de áudio, cada uma tendo sons de diferentes instrumentos de orquestra. Uma vez disponibilizadas ao sistema operativo as definições das faixas no idioma seleccionado, este pode apresentar um menu padrão ao utilizador. O utilizador pode então escolher faixas específicas para serem tocadas em conjunto, faixas específicas para serem eliminadas os seus níveis relativos, e outras opções “padrão”. Todavia, na eventualidade das “outras” faixas de áudio não representarem música de orquestra, ou representam-na mas numa forma que requer selecções de menu fora do comum, o *software* padrão do sistema operativo de interface com o utilizador para que o sistema possa determinar o que é para ser feito com as “outras” faixas de áudio não será suficiente. Para acomodar situações fora do comum, o sistema operativo tem de ser proporcionado com *software* especial para a criação do menu, assim como para controlar como as faixas seleccionadas são misturadas / eliminadas depois das selecções do utilizador. A técnica utilizada é a mesma que a técnica acima descrita em relação ao carregamento de *software* especial para alterar a operação geral do aparelho (campos 3 e 4). O campo 12 é um *bit* único. Se este for um 1, é uma indicação de que existe um campo 13 que contém *software* especial de mistura / eliminação. Conforme indicado na figura 3, o campo 13 tem assim entre nenhum *bits* a um número indeterminado de *bits* que é dependente do comprimento do *software* especial a ser carregado na máquina vindo do disco. O *software* especial termina com uma palavra de sincronização para que o aparelho saiba quando se inicia o próximo campo.

O campo seguinte, campo 14, consiste nas definições das faixas propriamente ditas. Uma vez que existem M “outras” faixas de áudio, e existem P idiomas nos quais estas têm de ser definidas para o utilizador, estão representadas P x M sequências de caracteres no campo 14. Cada série é separada da seguinte

por um caractere de escapar. Primeiro existem M sequências de caracteres (definições das faixas) no primeiro idioma correspondendo à primeira posição no campo 11 que contém um 1, depois existem M sequências de caracteres no segundo idioma correspondendo à segunda posição de *bit* no campo 11 que contém um 1, etc. Conforme será descrito mais abaixo, o utilizador informa ao aparelho em qual dos idiomas disponíveis deverá ser mostrado o menu que inclui as definições das faixas. Enquanto que a totalidade da série de *bits* de DADOS DE SAÍDA da *drive* do disco é estendida ao controlador mestre no sistema na figura 2, apenas a sequência de caracteres correspondente ao idioma seleccionado é processada. Estas são processadas e mostradas de acordo com o *software* padrão, ou com o *software* especial de mistura / eliminação que fora só lido do campo 12 se tal *software* estiver incluído no disco. (Deverá ser notado que é a função do desmultiplexador 63 distribuir aos vários *buffers* apenas os respectivos *bits* de dados que lhes são destinados. É o controlador 41 que diz ao desmultiplexador o que fazer depois do controlador interpretar a informação tanto na secção de introdução das faixas como nos bloco de dados individuais.).

Conforme descrito em relação à figura 2, é feita provisão para a inserção de subtítulos. O idioma é seleccionado pelo utilizador conforme será descrito, mas o aparelho tem de ser dito em que idiomas estão disponíveis os subtítulos. Outro campo de 100 *bits* é utilizado para este fim. Conforme indicado na linha 15 da figura 3, os "1"s no campo representam os idiomas individuais disponíveis para subtítulos. Como no caso dos idiomas disponíveis para mostrar, existe um máximo de 99 uma vez que a primeira posição de *bits* corresponde a M&E que não é num sentido estrito um "idioma".

O campo 16 é um código de versões múltiplas de 4 *bits*. O aparelho é informado não apenas se existem duas versões da mesma apresentação de vídeo no disco, mas também quais são as escolhas em relação às mesmas. O primeiro

bit é um 0 se existir apenas uma versão no disco, e neste caso o segundo e quarto *bits* são ignorados. O *bit* 1 tem um valor de 1 se existirem duas versões no disco. O segundo *bit* no código diz ao aparelho se é para ser implementada a opção de fechadura parental, ou se um outro critério é para ser utilizado na selecção sobre qual a versão a ser tocada. A situação usual é quando é implementada a opção de fechadura parental, e neste caso o *bit* na segunda posição do código de 4 *bits* é um 0. Isto informa ao aparelho que este deve determinar se a opção de fechadura parental está “ligada”. Se estiver ligada, versões de classificação R (ou, no sentido mais lato, de classificação para adultos) não deverão ser tocadas. O *bit* na posição 3 do código é uma indicação de se a versão A (a primeira ou única versão) é de classificação R ou não (0 = não, 1 = sim), e o quarto *bit* no código proporciona a mesma informação para a versão B se existirem duas versões; se existir apenas uma versão então o quarto *bit* é ignorado. Isto é toda a informação de que o aparelho necessita para determinar se qualquer das duas ou ambas as versões podem ser tocadas. Quando existem duas versões do mesmo filme no disco, é perguntado ao utilizador para seleccionar uma delas. Mas se a opção de fechadura parental estiver “ligada” e uma das duas versões for de classificação R, o utilizador é apenas dada a possibilidade de tocar a versão de não-adulto, ou de tocar nenhuma, conforme será descrito mais abaixo. Se ambas as versões forem de classificação R e a opção de fechadura parental estiver “ligada”, então o utilizador não poderá ver qualquer versão.

Por outro lado, é possível que existam duas versões do mesmo material no disco, mas não havendo a questão de um deles ser de classificação para adultos e o outro não. Por exemplo, uma versão poderá ser um filme didáctico incluindo perguntas e respostas, e a outra poderá envolver um teste sobre a mesma matéria incluindo apenas as perguntas. Para a maior parte as duas versões seriam iguais. Nestes casos, o primeiro *bit* no campo 16 seria ainda um 1 para indicar que estão disponíveis duas versões, mas o segundo *bit* seria um 1 em

Very Selen Murray

vez de um 0, para indicar que a escolha entre as duas versões não depende de elas serem ou não de classificação R. Um 1 na segunda posição de *bit* é uma indicação de que o terceiro e quarto *bit* caracterizam as duas versões respectivamente em relação a uma característica que não é a classificação.

O que o terceiro e quarto *bit* realmente significam neste caso, e quais as escolhas de menu proporcionadas ao utilizador, tem de ser determinado recorrendo a critérios diferentes. A mesma técnica que foi anteriormente aqui utilizada é agora utilizada mais uma vez -- *software* especial é proporcionado a par dos códigos de versão. O campo 17 consiste num único *bit* que serve de bandeira para indicar se está disponível *software* especial de versão. Se o *bit* for um 1, então o campo 18 é lido para aceder ao *software*. Como no caso dos dois campos de *software* anteriores, o campo 18 termina com um palavra de sincronização para indicar o início do campo seguinte. O *software* especial controla uma apresentação no menu que é exclusiva do disco em particular.

O campo seguinte consiste num único *bit*. Conforme indicado na figura 3, este informa ao aparelho se estão disponíveis dados de vídeo. Se não estiverem, isto significa simplesmente que não existem campos de bloco de dados de vídeo na totalidade dos blocos de dados para serem descritos em relação à figura 4.

O campo 20 é um único *bit*, e identifica a relação de aspecto base ou mestre. Se o *bit* tiver um valor 0, isto é uma indicação de que qualquer vídeo no disco tem uma relação de aspecto de "ecran largo" de 16:9, conforme apresentado na figura 9. Por outro lado, se o *bit* tiver for um 1, isto é uma indicação de que a relação de aspecto do vídeo no disco é de 4:3.

Conforme acima descrito, se o vídeo original tiver uma relação de

Uma Escola Unimig

aspecto de "ecran largo", então existem duas maneiras de derivar uma relação de aspecto reduzido de 4:3. Uma maneira é a de formar a imagem de vídeo desde a parte central do original em "ecran largo". Outra maneira é a de fazer um *pan scan* no sentido em que a secção da imagem original que é realmente utilizada não é necessariamente sempre a parte central. De facto, a figura 9 mostra a utilização de mais informação no lado esquerdo do que no lado direito da imagem original. O campo 21 é um único *bit* que é indicativo da disponibilidade de *pan scan*. Se o campo 21 é um 1, a relação de aspecto de base é 4:3 para que a disponibilidade de *pan scan* seja irrelevante -- um *bit* único no campo 21 é simplesmente ignorado. Mas se a relação de aspecto de base for 16:9 (o campo 20 tem um 0), o valor do *bit* no campo 21 diz ao aparelho se os subsequentes blocos de dados proporcionam informação sobre a coluna de partida que possa ser carregada no *buffer* de *pan scan* 57 na figura 2. Se o *bit* no campo 21 é um 0, os blocos de dados não incluem informação sobre a coluna de partida, e se o vídeo é para ser tocado com a relação de aspecto 4:3 de um original de "ecran largo", então a imagem de vídeo é formada a partir da parte central de cada imagem original. Por outro lado, se a informação de *pan scan* estiver disponível nos blocos de dados, então o *buffer* 57 na figura 2 é actualizado conforme requerido e o vídeo finalmente formado terá um grau de variabilidade adicional.

O campo 22 é um número de 20 *bits* que representa o número total de blocos de dados no disco. Todavia, se existem duas versões diferentes, embora tenham muitos blocos de dados em comum, o restante número de blocos nas duas versões poderão ser diferentes. Por exemplo, uma cena poderá ser totalmente omitida de uma das versões, e neste caso teria um número total de blocos de dados inferior. Por esta razão, se o campo 16 indicar que existem duas versões de um filme ou de outro material de fonte no disco, o campo 23 proporciona o número total de blocos de dados na versão A, e o campo 24 proporciona o número total de blocos de dados na versão B. Ambos os campos são omitidos se

apenas existir uma versão no disco.

Cada bloco de dados pode incluir informação de vídeo para um número variável de imagens. O sistema poderia determinar o tempo total de excussão através do número de blocos de dados (tanto o número total se só existir uma versão, ou dois números diferentes se existirem duas versões), apenas se o sistema for informado sobre o ritmo original de imagens e do número médio de imagens representadas em cada bloco para o disco na sua totalidade. Dois discos com o mesmo número de blocos de dados terão tempos diferentes de excussão se o material original de fonte de um deles for filme cujas imagens foram geradas a um ritmo de 24 por segundo e o outro tinha um material original de fonte derivado de uma câmara de vídeo de 30 imagens por segundo. O campo 25 é um valor de 4 *bits* que identifica o ritmo original de imagens (24, 30, etc.), um número necessário para a correcta geração do sinal de vídeo. Embora o tempo representado por cada bloco de dados poderá ser determinado através do ritmo de imagens se cada bloco de dados conter apenas uma imagem. Também, pode não haver qualquer informação de imagem, isto é, a bandeira de vídeo disponível no campo 19 pode ser 0. Consequentemente, o campo 26 é proporcionado. Este campo contém um número de 10 *bits* que representa o factor temporal do bloco, isto é, a duração média representada por cada bloco. A multiplicação do factor temporal do bloco pelo número total de blocos (ou o número total numa versão em particular) resulta no tempo de execução. (Na prática, o factor temporal do bloco é quase o mesmo para ambas as versões no disco. Se desejado, podem ser proporcionados factores temporais de bloco independentes.)

Conforme é de prática comum em discos ópticos em geral, o disco da invenção pode ser proporcionado com uma tabela de conteúdo para permitir ao utilizador seleccionar uma parte em particular para tocar, ou simplesmente para informá-lo do que está precisamente no disco e quanto tempo cada parte leva a

tocar. O campo 27, se incluído, é uma tabela de conteúdo. Se apenas estiver uma versão do material fonte em disco, então existe apenas uma tabela de conteúdo. Caso contrário, existe um campo 28 adicional que consiste numa tabela de conteúdo para uma segunda versão. A figura 3 apresenta os sub-campos do campo 27.

Por falta de um termo melhor, a representação de vídeo é dividida naquilo que é chamado de "capítulos". Para cada capítulo a tabela de conteúdo inclui um número de capítulo de 8 *bits*, permitindo assim um máximo de 255 capítulos individuais. A seguir a cada número de capítulo existe um número serial do bloco de 20 *bits* do bloco de partida. Será de recordar que todos os blocos de dados no disco são numerados em série. Por outras palavras, enquanto que blocos de dados podem ser comuns a ambas as versões A e B, ou específicas a apenas uma versão, os números dos blocos de dados estão em ordem de série ao longo da faixa do disco. A tabela de conteúdo inclui o número serial do bloco do bloco de dados que é o bloco de partida para cada capítulo.

De igual modo, a fim de determinar o tempo de execução de cada capítulo, o sistema tem de saber quantos blocos estão incluídos em cada capítulo. Por esta razão, o seguinte pedaço de informação é um 20 *bit* de duração do bloco. Multiplicando este número pelo factor temporal do bloco permite a determinação do tempo de execução de cada capítulo. Alternativamente, o tempo de execução efectivo de cada capítulo poderá ser proporcionado em vez da duração do bloco. (Tal informação poderá ser proporcionada para diferentes versões e padrões.).

A fim de apresentar o título de cada capítulo, têm de ser proporcionados séries de idiomas. Mais uma vez, o sistema tem de ser avisado sobre os idiomas que estão disponíveis para mostrar os títulos dos capítulos para que o utilizador possa seleccionar um deles. A técnica usual de proporcionar um

Wm Suter Wmmy

bloco de 100 *bits* para a identificação dos idiomas disponíveis é empregue.

Finalmente, as séries de idioma propriamente ditas para a identificação individual dos capítulos são proporcionadas. Cada série termina com um caractere de escapar para a separar da próxima série. Esta é a mesma técnica utilizada em relação às definições das “outras” faixas de áudio acima descritas em relação ao campo 14.

O campo 29 tem um mínimo de 100 *bits* e um máximo de 1200 *bits*. É de lembrar que podem existir até 12 padrões autorizados, isto é, a saída final de vídeo pode ser em até 12 diferentes formatos. A fim de assegurar a conformidade com os padrões de qualidade acordados entre todos os fabricantes de aparelhos e todas as editoras de *software* que acordaram em suportar um conjunto comum de especificações, é possível prevenir que editoras de *software* desautorizadas publiquem discos que irão tocar em aparelhos da invenção.

Mais, é possível limitar a algumas editoras em particular o fabrico de discos que irão tocar de acordo com apenas um subconjunto dos 12 padrões. Por exemplo, se há direitos de autores a serem pagos por cada disco que é fabricado de acordo com umas das especificações acordadas, e os direitos de autores variam de acordo com o número de padrões para o qual um disco pode ser tocado, é possível limitar certos fabricantes de *software* a apenas o subconjunto de padrões para os quais estes concordaram pagar. Por esta razão, existe um código de autorização encriptado para cada padrão; os códigos são todos armazenados no campo 29. O disco tocará de acordo com um padrão em particular apenas se o código de autorização encriptado correcto estiver contido no disco. O campo 29 inclui 100 *bits* para cada um dos padrões autorizados no campo 5. Uma vez que pelo menos um padrão tem de ser autorizado existem pelo menos 100 *bits*. O número máximo de *bits* são 1200 se todos os 12 padrões forem

autorizados.

O esquema de encriptação é baseado em princípios de criptografia de chave pública. Criptografia de chave pública é presentemente muito conhecida, e uma exposição bastante clara sobre o assunto pode ser encontrada na edição de Agosto de 1979 do "*Scientific American*", num artigo de HellMan entitulado "*The Mathematics of Public-key Cryptography*". O uso do sistema de criptografia de chave pública permite que uma mensagem seja encriptada no local A de acordo com uma chave secreta, ser transmitida ao local B, e descriptada no local B de acordo com uma chave pública. A chave secreta para encriptar a mensagem é conhecida apenas pelo transmissor. Tal esquema é tipicamente utilizado para autenticar uma mensagem. Após a descriptação no local de recepção da mensagem encriptada transmitida, a mensagem será legível apenas se tiver sido encriptada com o par da chave privada. E uma vez que a chave privada é privada, se mensagem encriptada for legível, deve ter tido origem no detentor da chave privada.

Criptografia de chave pública é utilizado na invenção da seguinte maneira. Os dados propriamente ditos da faixa são processados pela editora de *software* de acordo com um algoritmo predeterminado. Os pormenores do processamento não são importantes. Qualquer processador não trivial que proporcione, por exemplo, um resultado de 100 *bits* baseado nos dados do disco será suficiente. Um resultado de 100 *bits* é uma "mensagem" a ser transmitida via o disco em qualquer uma de entre uma e doze formas de encriptação. Existem 12 pares de chaves no sistema de criptografia, cada um associado a um dos diferentes padrões. A chave privada para o primeiro padrão autorizado no disco é utilizada para encriptar a mensagem de 100 *bits* e a encriptação de 100 *bits* é armazenada no campo 29. Esta encriptação é o código de autorização para um padrão em particular. A mesma coisa é feita para todos os outros padrões

Wm. Suter Murray

autorizados para um disco em particular, com a chave privada associada a cada um dos padrões a ser utilizada em cada caso.

O sistema operativo do aparelho computa o mesmo resultado ou mensagem de 100 *bits* que foi originalmente computada pela editora de *software*. O *software* do aparelho usa depois a chave pública associada a cada um dos padrões autorizados no disco para descriptar o respectivo código de autorização encriptado para aquele padrão. A mensagem descriptada deverá condizer com a mensagem computada pelo sistema operativo depois de processar os dados do disco. Se não condizerem, é uma indicação de que a editora de *software* não tinha a chave privada para encriptação do código de autorização para o padrão em particular, e o aparelho não produzirá um sinal de vídeo de acordo com aquele padrão.

Para explicar isto de uma maneira diferente, vamos assumir que a chave privada para o padrão autorizado N no disco dá origem a uma mensagem encriptada $Pri_N(X)$, onde X é a mensagem a ser encriptada. Igualmente, a função $Pub_N(X)$ representa a descriptação de uma função X utilizando um par de chave pública. Admitemos ainda que seja assumido que o algoritmo pré-determinado para o processamento dos dados no disco é conhecido por todos os fabricantes de aparelhos e das editoras de *software*, e dá origem a um resultado de 100 *bits* que é tratado como um "mensagem" M cujo conteúdo (valor) depende dos dados no disco. Para o padrão N, a editora de *software*, depois de primeiro derivar M, armazena no disco o código $Pri_N(X)$ de autorização encriptada de 100 *bits*. O aparelho deriva primeiro o valor M da mesma forma que a editora de *software* o fez. O *software* do aparelho utiliza então a chave pública associada ao padrão N para descriptar o código de autorização encriptado. O sistema operativo deriva assim o $Pub_N(Pri_N(M))$. Uma vez que a descriptação de uma mensagem encriptada deverá resultar na mensagem original, o resultado desta

Very Selen Murray

descriptação deverá ser o mesmo valor M que o sistema operativo deriva ao processar os dados no disco. Se for, então o padrão em particular não só está autorizado, como a editora tem o direito de o autorizar. Por outro lado, se a descriptação do código de autorização encriptado M não condizer com o resultado algorítmico M derivado pelo aparelho (porque a editora de *software* não tinha a chave privada com a qual derivar $\text{Pri}_N(M)$), então aquele padrão em particular é trancado fora.

Embora um tal esquema funciona em abstracto, existe um problema prático que tem de ser ultrapassado. Suponhamos, por exemplo, que o algoritmo utilizado para derivar a “mensagem” original M envolve o processamento de 20 blocos de dados no disco com números de bloco disposto em série predeterminados. (O processamento poderá ser algo tão simples como a multiplicação por cada um dos sucessivos grupos de 100 *bits* cada, e utilizando como resultado de cada multiplicação -- para a próxima multiplicação -- apenas os 100 *bits* menos significantes.). Uma editora que não tem poder para autorizar o padrão N no disco pode todavia, desejar fazê-lo. Este não conhece a chave privada com a qual deve encriptar o valor M derivado que é aplicável ao seu *software*. Consequentemente, este não sabe qual código encriptado de 100 *bits* este deveria pôr no disco que irá descriptar para o valor M num aparelho. Mas o que este pode fazer é copiar os 20 blocos de dados predeterminados de outro disco legítimo e pô-los no seu próprio disco, e também copiar o código de autorização encriptado no campo 29. Esses 20 blocos de dados, quando processados num aparelho, resultam no valor M , e irão condizer com o código de autorização encriptado “roubado” depois de este ser descriptado no aparelho. Naturalmente, a editora de *software* pode ter cometido infracção do *copyright*, mas isto simplesmente agrava o crime. O problema prático que a editora de *software* enfrenta é que este terá blocos de dados que são “tocados” e que estarão totalmente fora de contexto no que diz respeito ao seu filme. Todavia, devido à

maneira como versões múltiplas de um filme podem ser armazenadas no mesmo disco em primeiro lugar é que o aparelho pode ser controlado para saltar por cima de certos blocos de dados, conforme será descrito mais abaixo, a editora de *software* pode codificar os seus próprios blocos de dados para que os blocos de dados copiados não sejam tocados. Desta maneira, a protecção encriptada pode ser tornada ineficaz.

A solução é que enquanto o algoritmo que deriva a “mensagem” M pode em primeiro lugar também operar em blocos de dados predeterminados, este poderia operar em pelo menos a secção de introdução da faixa. Não há maneira nenhuma que uma editora não autorizada possa copiar os campos de introdução da faixa de outro disco porque isso daria ao aparelho informações incorrectas sobre o conteúdo vídeo e áudio no disco da editora não autorizada. Os dados de introdução são uma função do assunto em particular no disco, e tem de aparecer na faixa a fim de permitir o correcto funcionamento do disco. Assim a informação apresentada na figura 3 pode ser tratada como uma “mensagem” M cujas encriptações, uma para cada padrão autorizado, são derivadas utilizando as respectivas chaves privadas e são armazenadas no campo 29 de introdução. (Em todo o rigor, a “mensagem” M é o resultado do processamento de todos os campos exceptuando o campo 29. Também, os campos mais compridos, como aqueles que contêm *software*, podem ser omitidos do processamento.). O aparelho deriva a mesma “mensagem”, desencripta um código de autorização encriptado com a chave pública associada ao respectivo padrão, e depois compara os dois. Se estes não condizerem, o aparelho determina que aquele padrão em particular não foi autorizado para a editora do disco em particular.

O campo do código de autorização encriptado é mostrado perto do fim da figura 3 e assim o correspondente processamento é descrito perto do fim do gráfico de operação das figuras 5A - 5C conforme será descrito mais abaixo. O posicionamento do campo do código de autorização encriptado conforme

apresentado facilita uma descrição do seu processamento, mas de facto o campo pode ser colocado com vantagens no início do processamento. É de recordar que pode ser lido *software* especial do disco para modificar a sequência do aparelho normal. É assim concebível que um falsificador poderia escrever *software* especial que permitisse passar ao lado do processamento do código de autorização. Fazendo o processamento antes até da leitura de qualquer *software* especial, impede que o código de autorização seja passado ao lado.

Voltando à descrição dos campos das faixas de introdução, o campo 30 é uma bandeira de comando / dados de um bloco de dados de 1 *bit*. Este *bit* informa ao sistema operativo se os blocos de dados incluem informação de comando ou dados que são para ser lidos durante a execução do disco. Como o sistema determina se um bloco de dados em particular contém comandos ou dados será descrito mais abaixo. O campo 30 indica simplesmente se tal informação existe de todo. Finalmente, os campos 31 e 32 são campos de "apanha tudo" para permitir ao disco controlar maneiras pouco usuais do aparelho processar a informação no disco. É de recordar que o campo 3 contém uma bandeira que indica se o campo 4 contém *software* especial que leve o aparelho a operar de acordo com um programa que é totalmente diferente daquele que é usualmente empregue, o campo 12 indica se o campo 13 contém *software* especial de mistura / eliminação para uso com as "outras" faixas de áudio, e o campo 17 contém uma bandeira que indica se o campo 18 contém *software* de versão especial para o processamento do código de versões múltiplas de 4 *bit*. O campo 31, indica se existe um *software* "suplementar" no campo 32. O *software* "suplementar" é diferente do *software* especial do campo 4 no sentido em que o *software* no campo 4 é basicamente um substituto do processamento que é normalmente utilizado, enquanto que o *software* "suplementar" trabalha geralmente com aquele código, em conjunto com comandos e dados que são encontrados nos blocos de dados.

Tipicamente, o *software* "suplementar" iria permitir a execução de um jogo de vídeo, com comandos e dados relacionados nos blocos de dados a determinar o percurso da execução. Mas existem outras utilizações para esta técnica. Como outro exemplo da maneira como o *software* "suplementar", e comandos e dados nos blocos de dados, podem ser utilizados, consideremos um disco desenvolvido para tocar / passar um filme clássico com subtítulos, mas que também é proporcionado com um comentário crítico que é para ser mostrado periodicamente em vez dos subtítulos, talvez durante momentos em que o ecran é lavado a fica em branco à excepção do comentário crítico. Para mostrar a flexibilidade que é possível, vamos até considerar um caso em que o comentário crítico é para ser num idioma diferente. O que é requerido em tal caso é que o *buffer* de subtítulos 59 na figura 2 seja carregado durante a execução de alguns blocos de dados com subtítulos num idioma e com subtítulos noutro idioma durante a execução de outros blocos de dados (alguns blocos de dados contendo subtítulos correspondentes ao filme original, e outros contendo comentário crítico noutro idioma). Em tais casos, tem de ser dito ao sistema de alguma maneira para comutar para frente e para trás entre os subtítulos dos idiomas, isto é, as diferentes faixas de subtítulos têm de ser processadas em diferentes blocos de dados. Isto pode ser controlado convenientemente através de emissão de comandos nos próprios blocos de dados. Da mesma forma, se for desejado pôr o ecran em branco e interromper o filme durante a apresentação do comentário, poderá ser incluído no bloco de dados um valor de dados que representa o tempo de duração do ecran em branco. Alternadamente, se for para fazer um comentário num idioma diferente, poderá ser escolhida uma faixa de áudio diferente para este propósito. Em qualquer caso, o *software* especial carregado do campo 32 iria controlar o processamento dos comandos e dados contidos nos blocos de dados, e iria trabalhar em conjunto com o sistema operativo do aparelho.

O processamento dos campos das faixas de introdução

O gráfico de operação das figuras 5A-5E representam o processamento da informação nos campos das faixas de introdução. A descrição deste processamento preliminar é agora apresentado, tendo em mente as funções dos campos individuais. Os campos nos blocos de dados, assim como o processamento dos blocos de dados, são descritos mais abaixo.

O processamento do sistema começa, conforme mostrado no topo da figura 5A, com a leitura das regulações *default*. Estas são regulações estabelecidas pelo interruptores DIP, códigos ROM, ou o uso de qualquer outro dispositivo ou técnica que configure o sistema quando este é ligado. É típico em sistemas baseados em microprocessadores a reconfiguração de todas as bandeiras e a leitura das regulações *default* quando o aparelho é ligado.

Existem quatro regulações *default* que são assim determinadas a fim de configurar o sistema. O primeiro é o padrão -- aparelhos vendidos nos Estados Unidos, por exemplo, serão tipicamente configurados, no estado *default*, para produzirem um sinal de vídeo NTSC.

A seguinte regulação *default* é o idioma -- o idioma do diálogo da banda sonora, o idioma dos subtítulos (se existem), e o idioma no qual o menu é para ser apresentado no ecran. Nos Estados Unidos, por exemplo, o idioma *default* seria o inglês. Se o utilizador não informar o aparelho de que outro idioma é desejado para uma ou mais destas funções, a faixa 10 de idioma de áudio será utilizada para gerar a banda sonora, e sequências de caracteres no idioma inglês serão utilizados na composição do menu de mistura / eliminação para as "outras" faixas de áudio e para a tabela de conteúdos. Quanto aos subtítulos, o *default* usual é "sem idioma".

O terceiro *default* é a relação de aspecto, 4:3 nos Estados Unidos. A relação de aspecto determina as dimensões relativas do ecrã representado pelo sinal de saída final de vídeo.

Finalmente, o estado da fechadura parental é determinada. No sistema da figura 2, isto implica simplesmente a determinação da regulação da fechadura 81. Mas também é possível dispensar o fecho e chave física, e de armazenar o estado da fechadura parental numa memória não-volátil depois de primeiro introduzir no teclado uma palavra-chave conhecida apenas pelas pessoas que exercitam o controlo sobre a função de fecho.

Conforme acontece em muitos dispositivos de grande consumo, o teclado pode ser utilizado pelo utilizador em qualquer momento para interrogar ou controlar o aparelho. Sequências de controlo rotineiro que são padrão no estado da arte não são mostradas nos gráficos de operação. Por exemplo, o teclado, ou um dispositivo de controlo remoto associado, pode ser utilizado para controlar o volume, avanço rápido, um salto para um capítulo específico, etc. O processamento normal pode ser interrompido para controlar um ecrã através da operação da tecla de menu, conforme é conhecido no estado da arte. No início do processamento da figura 5A, é mostrado um teste para determinar se a tecla de menu é utilizada. A razão para mostrar uma interrogação sobre se a tecla de menu é utilizada no início do processamento, em oposição a qualquer outro período durante a execução do disco, é porque este é o mecanismo pelo qual as regulações *default* podem ser alteradas. Se a tecla de menu for utilizada quando o aparelho é ligado, o sistema mostra o menu. Conforme indicado no gráfico de operação, o utilizador é dado a opção de alterar os *defaults*, de visualizar as tabelas de conteúdos do disco, e/ou (no caso da tecla de menu ter sido utilizada acidentalmente) retornar simplesmente ao processamento sem alterar nada. Conforme indicado, dependendo da selecção do menu, os *defaults* são alterados,

a totalidade do processo de selecção do menu é abortado, ou uma bandeira de TOC (tabela de conteúdos) é fixada em 1. Esta bandeira será examinada mais tarde para determinar se a tabela de conteúdos deverá ser apresentada.

Até aqui, nenhuma informação do disco foi processada. (Nesta descrição, são por vezes feitas referências à leitura de um campo e outras vezes ao processamento de um campo. Deverá ser compreendido que mesmo quando é dito que depois de um certo passo de processamento é lido um campo, o campo poderá na realidade ter sido lido antes mas armazenado num *buffer* para uso mais tarde. Dependendo do contexto, a leitura de um campo significa lê-lo na realidade para que os *bits* apareçam no condutor 25 de SAÍDA DE DADOS na figura 2, ou para fazer alguma coisa com os dados se estes foram anteriormente lidos e *buffered*.) Com referência à figura 3, o primeiro campo de informação que é lido da secção da faixa de introdução é um campo de 40 *bits* representando territórios autorizados. De seguida, é feita uma verificação para ver se o território no qual o aparelho foi destinado a ser utilizado é um daqueles autorizados no disco. O território do aparelho é também um tipo de configuração *default*, mas não é agrupado com os outros porque este não pode ser alterado pelo utilizador. (Para permitir a um utilizado mudar de um território para outro onde queira usar o seu aparelho, o território do aparelho pode ser alterado por um técnico autorizado.) Se o aparelho tiver sido desenvolvido para utilização na China, por exemplo, e a China não for um dos territórios autorizados no disco, a execução do disco será abortado.

Por outro lado, se o disco tiver sido autorizado para tocar no território do aparelho, o campo 3 é lido. Este *bit* único diz simplesmente ao sistema se *software* especial está presente. Conforme mostrado no gráfico de operação, se este estiver presente então o *software* especial é lido do campo 4 e é executado. O processamento termina com o passo "executar o *software* especial".

Isto é destinado a mostrar que o *software* especial no campo 4 substitui basicamente o sistema operativo incorporado. Tal *software* será empregue quando uma mudança radical estiver envolvida no uso geral do aparelho. (Conforme acima mencionado, isto não é para dizer que o *software* especial não pode chamar rotinas BIOS e semelhantes dos *chips* de ROM que contêm o sistema operativo.)

Se não estiver presente nenhum *software* especial, o sistema lê o padrão *default*, por exemplo, determina que um padrão NTSC é para ser empregue. Se o utilizador tiver alterado o padrão *default* através de uma selecção de menu, por exemplo, para PAL, então PAL será o novo padrão *default*. O sistema acede depois ao campo 5 que autoriza até 12 padrões. O teste que é efectuado é para determinar se o padrão *default* (o original, ou conforme alterado no início do processamento) é autorizado. Se este não for, é apresentado um ecran que mostra ao utilizador os padrões autorizados, e este selecciona depois um. Depois de ser feita uma selecção apropriada, ou se o padrão *default* for autorizado, o sistema processa os campos 6 e 7. A leitura do campo 6 informa ao aparelho sobre os idiomas de áudio disponíveis (até 16, incluindo a M&E e 15 idiomas.)

Mais uma vez, um valor *default* é testado contra um conjunto de opções permitidas. Anteriormente, foi o padrão *default* que fora testado contra os padrões autorizados lidos do disco. Desta vez é o idioma de áudio *default* (quer seja o idioma *default* no momento da ligação quer seja um idioma diferente seleccionado pelo utilizador se a tecla de menu fora operada) que é comparado com todos aqueles disponíveis. Conforme mostrado no gráfico de operação, se o idioma *default* não estiver disponível, é formada uma apresentação que lista os idiomas de áudio disponíveis, e o utilizador selecciona um deles. O sistema lê depois os tipos de faixas no campo 7. Este é o campo que informa ao sistema operativo se existe uma faixa de M&E, se esta é para ser utilizada como um

mestre de mistura ou de comutação, e se a faixa de idioma seleccionada é uma faixa de áudio completa, é para ser misturada com o misturador mestre, ou para ser comutada com o comutador mestre. De seguida, são lidos do campo 8 os códigos das faixas. Conhecendo o idioma seleccionado, e o seu tipo de faixa e código de faixa, assim como a informação sobre M&E, mistura e comutação, o sistema operativo tem toda a informação que necessita para gerar uma banda sonora para o filme que a acompanha que vem de encontro às necessidades do espectador.

A seguinte coisa que é feita é a leitura do campo 9 para determinar o número de "outras" faixas de áudio que estão no disco, sendo desde nenhum até 63. Se não existirem de facto "outras" faixas de áudio, todo o processamento para determinar o que é para ser feito com estas faixas é passado ao lado. Mas se existirem tais faixas, o campo 10 é primeiro lido para determinar como estas estão codificadas. Em virtude de o utilizador ter de ser informado sobre o que está nas faixas antes deste poder determinar o que é para ser feito com as faixas, o sistema tem seguidamente de determinar lendo o campo 11 quais os idiomas de menus das "outras" faixas que estão em disco. O tipo de verificação usual é então efectuado para ver se o menu está disponível no idioma *default*. Se não estiver, os idiomas disponíveis são apresentados e o utilizador selecciona um deles.

Conforme acima descrito, o sistema operativo pode executar uma rotina padrão para ler o menu, apresentá-lo, e interagir com o utilizador enquanto este determina o que deve ser feito com as "outras" faixas de áudio. Mas na eventualidade de requer uma mistura ou eliminação especial, será necessário *software* especial de mistura / eliminação. O campo 12 é lido para ver se tal *software* está disponível e, conforme indicado no gráfico de operação, qualquer *software* especial de mistura / eliminação que estiver no disco é lido do campo 13. Só então é que os itens do menu propriamente ditos (no idioma seleccionado)

são lidos do campo 14 e apresentados ao utilizador. Utilizando os menus disponibilizados pelo sistema operativo, o utilizador selecciona o modo de execução para as “outras” faixas de áudio. O utilizador pode, por exemplo, misturá-las em qualquer maneira autorizada, utilizar o que está numa faixa para eliminação (por inversão de fases) de outra faixa mais inclusiva, ajustar uma faixa para tocar exclusivamente, ajustar os níveis relativos de áudio, etc. O *software* especial de mistura / eliminação, pode, naturalmente, proporcionar estas opções assim como outras não comumente oferecidas.

Conforme mostrado na figura 5B, a informação dos subtítulos é agora processada de acordo com o modelo estabelecido. Primeiro, o sistema determina se subtítulos são de todo em todo desejados. No início mesmo do processamento na figura 5A, é de recordar que uma das regulações *default* é o idioma dos subtítulos. As regulações *default* usuais serão de que os subtítulos não são desejados. Se este for de facto o caso, o processamento dos subtítulos é simplesmente passado por cima. Mas se subtítulos são desejados, os idiomas de subtítulos disponíveis são lidos do campo 15. É depois feito um teste para ver se o idioma de subtítulos *default* está disponível. Se não estiver, são apresentados os idiomas de subtítulos disponíveis e o utilizador selecciona um deles.

De seguida, o código de 4 *bits* de versões múltiplas é lido no campo 16. O primeiro *bit* indica se existem duas versões, ou apenas uma. Não é feita aqui uma ramificação porque primeiro o sistema tem de determinar se está disponível um *software* especial de versão, e isto é determinado no campo 17. Se estiver disponível um *software* especial de versão, este é lido do campo 18 e executado. Até ao ponto em que este *software* tem de saber se são disponíveis versões múltiplas, e o que os códigos na terceira e quarta posição representam, já foi aqui determinado. Embora esteja indicado no gráfico de operação que as opções apresentadas ao utilizador são para seleccionar de entre as versões

autorizadas, ou para sair, é para ser entendido que as opções apresentadas serão geralmente diferentes se for executado *software* especial de versões. Também, é para ser entendido que pode existir *software* especial de versões mesmo que exista apenas uma versão que possa ser tocada. Por exemplo, pode ser apropriado avisar o utilizador de que um programa em particular pode ser extraordinariamente perturbador, e perguntar por uma resposta de "continuar" antes de iniciar a execução -- tudo isto sendo separado e à parte de uma classificação R.

Se *software* especial de versões não estiver disponível, então os *bits* 3 e 4 no código de 4 *bits* de versões múltiplas são utilizados por razões de classificação. É feito um teste para ver se a fechadura parental está ligada. Se não estiver, então não existem restrições na execução das versões A e B, e ambas as versões são autorizadas. Se tiver sido previamente determinado que existe apenas uma versão, então essa versão é considerada como sendo a versão A e esta é autorizada.

Por outro lado, se a fechadura parental estiver ligada, têm se ser feitos testes para ver se as versões no disco têm classificação R. Conforme mostrado na figura 5C, se a versão A for de classificação R, assim como a versão B, então é abortado a execução do sistema; embora não mostrada, pode ser apresentada uma mensagem apropriada para informar ao utilizador a razão porque o sistema parou de tocar. Se a versão A for de classificação R mas a versão B não for, então é apenas autorizada a versão B. Por outro lado, se a versão A não for de classificação R mas a versão B for, então é apenas autorizada a versão A. Finalmente, mesmo que a fechadura parental esteja ligada, se nenhuma das versões for de classificação R, ambas as versões são autorizadas.

O sistema apresenta de seguida as opções disponíveis ao utilizador.

Este pode escolher de entre as versões autorizadas, ou pode sair e parar de tocar o disco. (Este último caso poderá surgir, por exemplo, se uma criança tentar ver uma versão de classificação R, é lhe dito que esta não pode ser tocada, e é tomada uma decisão para avançar para uma coisa mais interessante.).

Se apenas estiver disponível uma versão, se esta não for de classificação R, e se não existir nenhum *software* especial de versões, então pode não haver necessidade para uma apresentação -- existe apenas um filme que possa ser tocado, e não existem restrições sobre quem o pode ver. Não obstante, conforme mostrado no gráfico de operação, ao utilizador ainda é dada a opção entre executar o que está no disco ou abortar a execução. O sistema poderia ser desenvolvido para saltar a apresentação nestes casos e simplesmente assumir que o utilizador quer ver a única versão do filme que está no disco. Por outro lado, o gerar da apresentação permite ao utilizador verificar que o disco que este meteu no aparelho é de facto o disco que este pretende.

Embora a invenção tenha sido descrita até agora em termos de uma ou duas versões de um filme num disco, é para ser entendido que podem existir três ou mais versões. Esta é uma das principais razões de proporcionar desde logo a capacidade de ler *software* especial de versões. Este *software* pode incluir toda a informação requerida sobre as várias versões donde são formados os ecrans dos menus para que o utilizador possa escolher o que é para ser executado. Conforme acima descrito, o *software* especial de versões pode permitir escolhas entre modos de ensinamento e de testes, e outras opções que nada têm a haver com se filmes em particular são de classificação para adultos.

O sistema lê de seguida o *bit* de disponibilidade de vídeo no campo 11, e assim determina se os blocos de dados que serão processados contêm subsequentemente dados de vídeo. Se estiverem presentes dados de vídeo, então a

relação de aspecto de base ou mestre no qual tenha sido armazenado no disco tem de ser determinado. O passo seguinte envolve assim a leitura do campo 20 para determinar se a relação de aspecto base ou mestre é 16:9 ou 4:3. Se a relação de aspecto mestre for 4:3, os seguintes cinco passos são passados por cima porque a disponibilidade de *pan scan* é irrelevante. Se a relação de aspecto *default* for de 4:3, então existe uma correspondência de um para um entre as imagens armazenadas e as apresentadas; se a relação de aspecto *default* for 16:9, então é apresentada uma imagem 4:3 num "ecran largo" com uma banda escura em cada lado. (Alternativamente, a imagem 4:3 poderia ser expandida para preencher o ecran 16:9, com a conseqüente perda de informação de cima e/ou de baixo.) mas se a relação de aspecto for 16:9, conforme mostrado na figura 9, existem várias possibilidades que têm de ser exploradas.

Um dos valores *default* que são determinados desde o início do processamento é a relação de aspecto. O sistema operativo verifica se a relação de aspecto *default* é *pan scan* de 4:3. Com referência à figura 9, se a relação de aspecto mestre é "ecran largo" (sendo processada a ramificação do gráfico de operação), então as possibilidades são "caixa de correio", *pan scan* centrada na imagem "ecran largo" (não mostrado na figura 9), ou *pan scan* variável (isto é com um número variável de coluna de partida). Se o *default* não for *pan scan* 4:3, então não existem escolhas para o utilizador fazer neste momento. O *default* é então "ecran largo" ou "caixa de correio", e o processamento subsequente é de acordo com o *default* que já tiver sido determinado.

Por outro lado, se o *default* for *pan scan* 4:3, a questão é se informação de *pan scan* variável existe no disco. O *bit* de disponibilidade de *pan scan* é então lido do disco no campo 21. Se *pan scan* estiver disponível, isto significa que os blocos de dados irão especificar ao sistema operativo quais os números da coluna de partida para o *pan scan* -- o utilizador não necessita de

escolher nada neste momento. Por outro lado, se *pan scan* não estiver disponível, e este era o *default* do utilizador, este terá de decidir de entre duas possibilidades -- um corte ao centro, em que a parte central de cada imagem de "ecran largo" pode ser vista, ou uma forma de "caixa de correio" em que a totalidade de todas as imagens pode ser vista, mas o ecran tem bandas escuras em cima e em baixo. É formada uma apresentação de menu, e o utilizador selecciona um dos dois modos.

Este uso de uma relação de aspecto comum no disco que não obstante permite ao utilizador escolher de entre vários tipos diferentes de apresentações exemplifica a abordagem de desenvolvimento da invenção. A ideia básica é a de proporcionar a máxima flexibilidade ao mesmo tempo que se armazena todos os dados requeridos num disco óptico sensivelmente do tamanho de um CD convencional. Depois de estar armazenado no disco um filme de "ecran largo", quase mais nenhum bem de raiz adicional é requerido para permitir ao utilizador gerar uma saída de vídeo tendo outra qualquer relação de aspecto. Embora possam existir até 15 idiomas no qual o dialogo possa ser ouvido, não existem nem de longe 15 bandas sonoras completas por causa das capacidades de mistura e comutação incluídas no aparelho e a maneira como informação redundante é eliminada das faixas de idioma de áudio. A mesma coisa se aplica aos padrão de vídeo. Enquanto que até à data o vídeo de elevada qualidade tem requerido um meio que possa ser tocado apenas em NTSC ou PAL, etc., a presente invenção permite que o mesmo disco dê origem a sinais de vídeo até 12 padrões. Uma das vantagens da invenção é que esta reduz grandemente o número de discos diferentes que têm de ser produzidos, por exemplo, por uma produtora de filmes que distribui filmes em todo mundo. Embora seja verdade que alguns campos terão de ser alterados de tempos a tempos, por exemplo, diferentes padrões têm de ser autorizados quando os vídeos forem lançados em NTSC e em PAL em alturas diferentes, tais alterações são relativamente triviais e são facilmente feitas.

Uma vez tomada a decisão sobre o modo do ecran, o campo 22 é lido para determinar o número total de blocos de dados no disco. Se existirem versões múltiplas, são também lidos os campos 23 e 24 a fim de determinar o número total de blocos de dados em cada versão. O campo 25 é então lido para determinar o ritmo original de imagens, e o campo 26 é lido para determinar o factor temporal do bloco.

O campo 27 é então processado. É de recordar da figura 3 que este é o campo que contém todas as informações necessárias para a apresentação da tabela de conteúdos. A tabela de conteúdos para a versão seleccionada (campo 27 se apenas existir uma versão, ou existem duas e a primeira foi seleccionada; ou o campo 28 se existem duas e a segunda foi seleccionada) inclui uma representação de 100 *bits* dos idiomas de apresentação dos capítulos disponíveis. O idioma de menu *default* é verificado contra aqueles que estão disponíveis. Se o idioma de menu *default* não estiver disponível, o utilizador é informado dos idiomas em que os títulos dos capítulos podem ser apresentados, e este selecciona de entre estes. Uma vez determinado o idioma de apresentação da informação dos capítulos, são calculadas as durações em tempo das várias tabelas de conteúdos. Uma vez que é conhecida quantos blocos de dados existem em cada capítulo, a duração de cada capítulo pode ser determinada multiplicando o número de blocos pelo factor temporal do bloco.

A tabela de conteúdos não é necessariamente apresentada. Esta é apenas apresentada se a bandeira TOC for activada no início do processamento, tendo o utilizador indicado que a tabela de conteúdos deveria ser apresentada. Se a bandeira TOC for 0, não há necessidade de apresentar a tabela de conteúdos. O sistema selecciona automaticamente o primeiro bloco de dados como o ponto de partida, isto é, a execução do disco começa no início. Por outro lado, se a bandeira TOC for um 1, a tabela de conteúdos é apresentada e o utilizador é dado

a opção de seleccionar o ponto de partida.

A seguir à tabela ou tabelas de conteúdos no disco estão os códigos de autorização encriptados para os padrões autorizados no campo 5. O sistema operativo lê o código de autorização encriptado para o padrão que está seleccionado. Este lê a seguir os dados predeterminados para o padrão que está seleccionado. É de recordar que para cada um dos 12 padrões possíveis, são processados dados predeterminados no disco para derivar uma "mensagem" M que serve como um código de autorização. É este código de autorização que é armazenado em forma encriptada no disco utilizando a chave privada associada a cada padrão. Os dados que são lidos do disco podem ser diferentes para cada padrão, desde que os mesmos dados são lidos e processados tanto durante o processo de encriptação como quando o aparelho deriva por si só a "mensagem" M. Conforme acima referido, é preferível que os dados incluam pelo menos parte dos campos de introdução porque seria auto-derrotismo para uma editora autorizada copiar estes dados.

Depois de estarem lidos os dados predeterminados para o padrão seleccionado, o código de autorização ("mensagem" M) é computado através dos dados. Utilizando a chave pública associada com o padrão seleccionado, cuja chave é imbuída no sistema operativo, o código de autorização armazenado no disco para o padrão seleccionado é descriptado. O teste para saber se a editora de *software* está autorizada a publicar discos que irão executar como sinais de vídeo no padrão seleccionado envolve a comparação o código de autorização de descriptação com o código de autorização computado. Se estes não combinarem, é abortado a execução.

Se estes dois códigos combinarem, é lido o campo 30. Este *bit* único informa simplesmente ao processador mestre se existem quaisquer

comandos ou dados armazenados nos blocos de dados para além do complemento normal mostrado na figura 4 e que será descrito mais abaixo. Se a bandeira é um 0, o sistema operativo nem procura tais comandos ou dados adicionais nos blocos de dados. Se a bandeira é um 1, isto significa que comandos ou dados podem estar presente num bloco de dados, mas não necessariamente.

Finalmente, o campo 31 é lido a fim de determinar se está disponível *software* "suplementar". Se estiver, este é lido do campo 32. O *software* "suplementar", conforme acima descrito, não é para ser utilizado em vez do *software* do sistema operativo, mas antes como suplemento a este. Esta é a diferença básica entre o *software* nos campos 4 e 32. Falando de uma forma geral, o *software* "suplementar" opera em comandos e dados incluídos nos blocos de dados num campo cuja presença é indicada (embora não necessariamente em todos os blocos de dados, conforme será aparente mais abaixo) pela bandeira do *software* "suplementar".

Com a leitura do campo 32 e a sua integração com o sistema operativo, a cabeça de leitura é levada a mover-se para o ponto de partida. Conforme acima descrito, o ponto de partida ou é o primeiro bloco de dados ou um bloco de dados determinado pelo utilizador se um capítulo diferente do primeiro tiver sido seleccionado. Os blocos de dados são lidos em sequência e o desmultiplexador 63 na figura 2 distribui os campos de dados a vários *buffers*. Conforme descrito no gráfico de operação, a leitura dos blocos de dados ocorre apenas se nenhum dos *buffers* estiver cheio. Mais, antes de ser lido um novo bloco de dados, o sistema verifica se existem algumas interrupções que têm de ser atendidas. O controlador 41 é a fonte de todas as interrupções. Por exemplo, se o utilizador tiver operado o teclado, o controlador gera uma interrupção na linha 43 na figura 2 que temporariamente pára a leitura dos blocos de dados. Depois das interrupções terem sido processadas, ou se não existir nenhuma interrupção a ser

atendida, é lido o bloco de dados seguinte. Conforme será descrito, o número serial do bloco é uma das primeiras coisa a serem lidas. O analisador 47 de número / apontador de bloco sabe o número do bloco seguinte que é requerido. Muitas vezes, isto será simplesmente o bloco que se segue na sequência da série. Todavia, o número do bloco pode estar fora de sequência, por exemplo, se tiver de ser dado um salto para outro capítulo, ou, conforme será aparente mais abaixo, certos blocos têm de ser saltados no disco quando se toca uma de múltiplas versões de um filme. Em qualquer caso, os sistemas verificam se o bloco que está a ser lido é o bloco correcto. Se não for, é feita um ramificação para o início do processo de leitura de blocos para que possa ser lido um bloco diferente. Também, o *gate* 61 na figura 2 é fechado para que os dados "incorrectos" no condutor 25 não sejam estendidos ao desmultiplexador 63.

Se o bloco lido for o bloco requerido, uma das primeiras coisas lidas imediatamente a seguir ao número do bloco são os dados do apontador. Os dados do apontador são utilizados pelo analisador 47 de número / apontador de bloco para determinar o número do bloco do bloco de dados seguinte que é requerido, conforme é indicado lá para o final do gráfico de operação. O número do bloco é transmitido através do cabo 49 para o microprocessador controlador 27 do *drive* do disco a fim de aceder a este bloco de dados ao completar a leitura do actual bloco de dados. Conforme indicado no final do gráfico de operação, o remanescente do bloco de dados que está a ser processado neste momento é lido e carregado nos vários *buffers*, depois do qual pode ser lido outro bloco de dados.

O gráfico de operação que se acabou de rever controla o processamento do aparelho. O que é realmente feito com os dados lidos dos blocos de dados é mostrado no gráfico de operação na figura 6, e este gráfico de operação será descrito depois dos campos num bloco de dados, conforme listado na figura 4, serem compreendidos. Mas a fim de apreciar a função dos dados do

apontador que estão incluídas num bloco de dados, serão primeiro descritas as figuras 7A e 7B. Estas figuras mostram como blocos de dados associados a versões individuais ou ambas as versões de um filme se interrelacionam uma com a outra, e como o sistema é controlado para saltar por cima de certos blocos de dados a fim de executar a versão seleccionada.

Figuras 7A e 7B -- A Função dos Dados do Apontador

No enquadramento ilustrativo da invenção, pode haver duas versões do mesmo filme no disco. A maioria dos blocos de dados irão representar vídeo e áudio que é comum às duas versões. Todavia, haverá outros blocos que são específicos a uma versão ou à outra. A questão é como controlar a leitura em sucessão dos blocos de dados que são requeridos para a versão seleccionada entre as duas versões.

Por questões de descrição, serão utilizadas as letras A, B e C para identificar respectivamente os blocos de dados que são específicos à versão A do filme, blocos de dados que são específicos à versão B, e blocos de dados que são comuns a ambas. A figura 7B ilustra uma porção da faixa com sucessivos blocos de dados sendo marcados A, B, ou C. É para ser entendido que na prática existem muitos milhares de blocos de dados em sucessão do mesmo tipo, com a maioria dos blocos de dados no disco sendo do tipo C. Todavia, para ilustrar a maneira como o sistema salta por cima de blocos de dados que não são requeridos, a figura 7B mostra no máximo dois blocos do mesmo tipo em sucessão.

A figura 7B mostra duas sequências, uma em cima para executar a versão B, e outra em baixo para executar a versão A. Se for seleccionada a versão B, e é assumido que de alguma forma o bloco B no lado esquerdo está a ser executado, é aparente que os dois blocos A que se seguem têm de ser saltados por cima a fim de ir para o quarto bloco, um bloco B. Depois deste bloco ser

executado, o bloco A que se segue tem de ser saltado por cima. São depois executados dois blocos comuns do tipo C, depois do qual o bloco A que se segue tem de ser saltado por cima para outro bloco C. O bloco seguinte, um bloco B, é então executado, seguido por blocos B, C e B. Finalmente, é dado um salto por cima de um bloco A para o último bloco mostrado na figura 7B, um bloco C.

Se estiver em execução a versão A, por outro lado, são executados dois sucessivos blocos A, depois há um salto por cima de um bloco B, são executados os seguintes cinco blocos -- A, C, C, A, C --, seguido de dois saltos por cima de dois blocos B para um bloco C, e finalmente há um salto por cima de outro bloco B para um bloco A seguido por um bloco C.

O padrão que emerge é que existem três tipos de transições de um bloco para outro. Primeiro, existe a execução de um bloco imediatamente a seguir à execução do bloco precedente. Existem sete exemplos disto mostrado na figura 7B -- AA, BB, CC, CA, CB, AC e BC. As duas possibilidades que são excluídas são AB e BA, uma vez que blocos específicos às duas versões nunca serão executados durante a mesma sessão de execução do disco, e muito menos uma a seguir à outra. Embora existam sete tipos de transição de um tipo de bloco para outro tipo de bloco, existem na realidade apenas três operações básicas -- ir de um bloco de um tipo qualquer para o seguinte bloco de um tipo qualquer; um salto de tanto um A para um A ou C, como de um B para um B ou C; ou uma ramificação de um bloco C tanto para um A ou B adjacente, como para um B ou A algures mais adiante na linha. A maioria das transições são do primeiro tipo. O segundo tipo ocorre quando um A é seguido de um B (cujos dois blocos nunca podem ser executados em sucessão); tem de ser dado um salto de A para outro A ou para um C. Reparos similares aplicam-se a um B seguido por um A. O terceiro tipo ocorre no final da execução de um bloco C, quando já não existe qualquer material comum para ser tocado e tem de ser feita uma comutação para uma versão ou para outra; o bloco seguinte é executado se este fizer parte da versão

seleccionada, ou senão alguns blocos terão de ser saltados por cima se a ramificação for para um bloco na outra versão.

A figura 7A mostra o diagrama de estado que define como e quando as transições são feitas de um bloco para outro. Conforme será abaixo descrito, todos os blocos de dados incluem uma bandeira apontadora de dois *bits*, possivelmente seguida de um campo que contém um apontador de 20 *bits*. (Quando o apontador está presente, este aponta sempre para o número serial do bloco de outro bloco de dados.). Fazendo referência ao código na figura 7A, se a bandeira apontadora de dois bits for 00, isto é uma indicação de que o processamento deve continuar com o bloco seguinte: neste caso, não há necessidade de um apontador. Se a bandeira apontadora de dois bits for um código 01, isto é uma indicação de que deverá ser dado um salto para um bloco na mesma versão a alguma distância mais abaixo, ou para um bloco C a alguma distância mais abaixo. Em qualquer destes casos, é necessário um apontador.

Os códigos 10 e 11 são utilizados quando é para ser feita uma ramificação de um bloco comum C. Qual o código que é utilizado depende do bloco que se segue ser um A ou um B. Se o bloco que segue ao C for um A, é utilizado o código 10 e o apontador é para um B ou um C mais abaixo na linha. Se o código for 11, isto significa que o bloco seguinte é um B, e o apontador é para um A ou um C mais abaixo na faixa. O sistema operativo sabe qual a versão que está a ser tocada. Se estiver a ser executada a versão A e o actual bloco tem uma bandeira apontadora de 10, isto significa que o bloco seguinte, um A, deverá ser executado depois do actual bloco. Não há necessidade de um apontador. O apontador é necessário no caso de estar a ser executada a versão B. Neste caso, uma vez que o bloco seguinte é um A, este não deverá ser executado. O aparelho deverá saltar para o bloco identificado pelo apontador -- ou para um outro C, ou para um B específico à versão B que está a ser executada.

Da mesma maneira, se estiver a ser executada a versão A e o actual bloco for um C com um código 11 na sua bandeira apontadora, isto significa que o bloco seguinte é um B. Uma vez que está a ser executada a versão A, o bloco seguinte não deverá ser executado depois do actual. Em vez disso, é dado um salto para o bloco A ou C identificado pelo apontador. Por outro lado, se estiver a ser executada a versão B o sistema continua simplesmente para o bloco seguinte.

A legenda na figura 7A mostra se é utilizado ou não o apontador quando são encontradas bandeiras apontadoras 10 e 11 num bloco C. A representação 10(P) é uma indicação de que o apontador deverá ser utilizado, e a representação 10 [P] é uma indicação de que o apontador deverá ser ignorado. É de recordar que o código 10 é utilizado para um bloco C quando o bloco seguinte for um A. Se estiver a ser executada uma versão A, o apontador não é necessário. É por isso que uma transição do bloco C para o bloco seguinte, um bloco A, é mostrada pelo símbolo 10[P]. Por outro lado, se estiver a ser executada uma versão B, uma vez que o bloco seguinte é um A este não pode ser executado depois do actual C. Em vez disso, tem de haver um salto para o bloco identificado pelo apontador e assim o uso da representação 10(P) -- o apontador aponta ou para o bloco B ou para outro C.

Argumentos semelhantes aplicam-se às representações 11(P) e 11[P]. Em ambos os casos, é o bloco C que está a ser executado e o bloco seguinte é um B. Se a versão A estiver a ser executada, o bloco seguinte não deverá ser executado e assim o símbolo 11(P) é requerido para mostrar uma transição de estado. Por outro lado, se a versão B estiver a ser executada, deverá ser executado o seguinte bloco B, e assim o símbolo 11[P] é apropriado.

Os quatro códigos, assim como as utilizações de (P) e [P], são

apresentadas na figura 7B. Fazendo referência à sequência de transição EXECUTAR B, a primeira transição mostrada é 01(P). É de recordar que o código 01 representa um salto de uma versão para um bloco da mesma versão ou para um bloco comum, e é requerido um apontador. A primeira transição mostrada é 01(P), um salto de um bloco B para outro bloco B. A transição seguinte na linha EXECUTAR B é 01(P), um salto de um B para um C. De seguida é um exemplo da transição mais comum, 00, a execução ordeira do bloco seguinte depois do actual bloco.

A quarta transição na linha EXECUTAR B é representada por um símbolo 10(P). O código 10 representa uma ramificação de um bloco C quando o bloco seguinte é um A, o exemplo ilustrado na figura 7B. Em tal caso, conforme indicado na figura 7A, se está a ser executada a versão B é dado um salto para o bloco identificado pelo apontador -- neste caso, o C seguinte.

O código 11 é utilizado para identificar uma ramificação de um bloco C quando o bloco seguinte é um B. Se estiver a ser executada a versão B, o caso em consideração, o apontador não é necessário porque o bloco seguinte é para ser executado. É por isso que o seguinte código mostrado é 11[P]. Seguem-se dois códigos 00 que representam transições óbvias para blocos adjacentes, seguidos por um código 11[P], uma ramificação de um bloco C para o bloco seguinte que é um B. Finalmente, é dado um salto deste bloco B por cima do seguinte bloco A para um bloco C. Isto requer um código 01(P) -- o código utilizado para saltar de um bloco de qualquer versão para um bloco da mesma versão ou para um bloco comum.

A sequência EXECUTAR A na figura 7B assume que é a versão A que está a ser executada. Os primeiros quatro códigos representam transições para blocos adjacentes, ou um salto de um bloco de uma versão para outro bloco

na mesma versão. O seguinte código, 10[P], é utilizado para mostrar uma ramificação de um bloco C para um bloco A adjacente. O apontador não é utilizado uma vez que a versão A está a ser executada, e o código 10 é empregue porque o seguinte bloco é um bloco A. O seguinte código 00 simboliza a transição do bloco A para o seguinte bloco C.

Segue-se um salto do bloco C para outro bloco C, saltando por cima de dois blocos B. O código 11 é utilizado porque este é o código empregue quando um bloco B segue um bloco C. O símbolo utilizado é 11(P), e não 11[P], porque o apontador é requerido na ida de um bloco C para outro bloco C mais abaixo na linha. Da mesma maneira, o código seguinte é mais uma vez um código 11(P) para simbolizar uma ramificação do bloco C para um bloco A mais abaixo na linha. A sequência na figura 7B termina com a transição de um bloco A para o bloco seguinte que é um C, para o qual é utilizado o código 00.

O diagrama de estado na figura 7A resume todas as possibilidades. Consideremos primeiro o estado em que um bloco A está a ser processado, representado pelo círculo com um A no seu interior no canto superior esquerdo. A bandeira apontadora de dois bits num bloco A é 00 se o seguinte bloco também for um A (mostrado pela transição de A de volta a A). Se o bloco seguinte for um B, por outro lado, então este claramente não deverá ser executado. Tem de haver um salto do bloco A por cima do B, ou para outro A ou para um C. Em qualquer caso, o código é 01(P). O desenho mostra tanto um salto por cima de B (para outro A), como um salto por cima de B para um C. A única outra transição de um bloco A é para o bloco seguinte se este for um C. Isto é mostrado pelo código 00.

Existem quatro transições semelhantes mostradas para o estado B, isto é quando um bloco de dados na versão B está a ser lida. O código 00 é utilizado se o bloco seguinte for um B ou um C. O código 01(P) é utilizado

quando o bloco seguinte for um A, e este é saltado por cima para que o sistema possa de seguida ler outro B ou um C.

As transições de um bloco C são mais complicadas porque existem sete destas, em vez de apenas quatro como é o caso de cada um dos blocos A e B. Se o bloco seguinte também for um C, o código é simplesmente 00 -- lê o bloco seguinte. Se o bloco seguinte for um B e tem de ser dado um salto para outro C, o código 10(P) controla o salto por cima do A. Da mesma maneira, o código 11(P) controla um salto por cima de um B para outro C. É de recordar que estes dois códigos são utilizados para controlar as ramificações de um bloco C, dependendo de o bloco seguinte ser um A ou B. Em quaisquer dos casos se o bloco seguinte não é para ser lido, este (e blocos iguais a este) têm de ser saltados por cima para o próximo C.

Todavia, depois de ler um bloco C, também é possível ler um A ou um B. Para ler um A, são utilizados um dos códigos 11(P) ou 10[P]. O código 11 é empregue quando o bloco seguinte é um B, e neste caso é requerido o apontador. O código 10 é utilizado quando o bloco seguinte é um A, e neste caso o apontador não é utilizado. Da mesma maneira, para ler de seguida um bloco B, ou é utilizado o código 10(P) ou o 11[P]. O primeiro é empregue quando o bloco seguinte no disco for um A, e o apontador é requerido porque este bloco tem de ser saltado por cima. Por outro lado, se o bloco seguinte for um B, o código 11 diz ao sistema para prosseguir para este bloco que se segue, e no processo ignorar o apontador porque este não é necessário.

Talvez o ponto mais importante a reconhecer é um que não é aparente dos desenhos, e é que a maioria dos blocos irão conter bandeiras apontadoras 00 e nenhum apontador. (O código 00 é o único sem um campo apontador a seguir.). Isso é porque uma vez que uma imagem de qualquer versão

esteja a ser executada, ou uma vez que uma imagem do material comum esteja a ser executada, o mais provável é que a imagem seguinte seja do mesmo tipo. Consequentemente, um código 00 por si só faz o trabalho. O resultado final é que as duas versões do mesmo filme podem ser armazenadas no disco, tendo o utilizador a opção de executar qualquer uma delas (desde que seja permitido pela fechadura parental), e apenas uma fracção muito pequena da totalidade dos bens de raiz do disco é "desperdiçado" por se estar a fazer a gestão de arquivo de *bits* que controlam as transições de um bloco para o bloco seguinte que é para ser lido depois deste. Mais uma vez, isto vai de encontro à filosofia subjacente ao desenvolvimento de proporcionar a máxima flexibilidade e o maior número de opção possíveis, sem desnecessariamente desperdiçar *bits* no processo.

Também deverá ser notado que a invenção não está limitada à colocação de apenas duas versões de um filme num disco. É possível usar a mesma técnica com três ou mais versões (embora a necessidade para tantas versões é menos provável). Em tal caso, os blocos comuns requeriam dois apontadores, não apenas um. Se existem três versões no disco, a seguir a um bloco C, o bloco seguinte poderá ser um A, B ou D. Dois apontadores seriam requeridos para apontar para os dois blocos que são para ser encontrados mais abaixo na linha. Obviamente, isto é apenas uma das alterações que teriam de ser feitas. A questão é que versões múltiplas podem ser acomodadas, embora seja à custa de mais *bits* de gestão de arquivo. Não obstante, o número total de *bits* de apontadores deste tipo é ainda inconsequente em comparação com o número total de *bits* áudio e vídeo.

Campos de Blocos de Dados

A figura 4 mostra os campos de um bloco de dados, e o formato é semelhante àquele que é mostrado para os campos da faixa de introdução na

figura 3. Todos os blocos de dados começam com uma palavra de sincronização. Conforme acima descrito, o padrão da palavra de sincronização não pode aparecer nos dados, e assim quando este é detectado o sistema operativo sabe que um novo bloco de dados está prestes a iniciar.

O segundo campo é um número serial do bloco de 20 *bits*. Todos os blocos no disco são numerados em ordem serial. O número do bloco é a primeira coisa lida porque este é utilizado pelo analisador 47 de número / apontador de bloco na figura 2. O número do bloco é essencial, por exemplo, para saltar de um bloco para outro. A cabeça de leitura será normalmente posicionada num ponto perto do bloco desejado, mas é muito improvável que o bloco correcto seja seleccionado na primeira tentativa. Isto é particularmente verdade uma vez que o número de *bits* nos blocos de dados é variável, e o sistema não tem maneira de saber quantos *bits* existem nos blocos que são saltados. Lendo o número do bloco no início do bloco de dados, o sistema pode rapidamente determinar se a cabeça tem de ser reposicionada.

O terceiro campo é um código de dois *bits* que representa se o bloco faz parte da versão A, da versão B, ou se é comum a ambas. (Apenas três de entre quatro códigos possíveis são utilizados.) Pode-se interrogar porque é que o sistema alguma vez teria de verificar a versão num bloco em particular, uma vez que quando começa a executar a versão A ou a versão B, os apontadores discutidos em relação às figuras 7A e 7B irão sempre identificar um bloco que seja ou comum ou parte da versão que está a ser executada. A resposta tem a ver com as operações de avanço rápido e recuo rápido. Embora estas operações não tenham sido discutidas extensivamente por serem técnicas inteiramente convencionais, quando avançando rapidamente, por exemplo, a cabeça de leitura pode ser posicionada mais ou menos arbitrariamente. O vídeo não deverá ser mostrado se este for da versão errada. Não é possível determinar a versão de um

bloco olhando simplesmente para o número ou para o apontador do bloco. Nenhum destes identifica a versão. É por esta razão que o sistema tem de determinar a versão do bloco quando este é primeiro lido.

Os campos 4 e 5 contêm a bandeira apontadora de dois *bits* e o apontador de 20 *bits* que têm de ser explicados em pormenor em relação às figuras 7A e 7B.

O campo 6 é uma bandeira de um *bit* que pode ou não estar presente. Com referência à figura 3, a bandeira de disponibilidade do vídeo no campo 19 diz ao sistema operativo se existe algum vídeo nos blocos de dados. Mesmo que exista, todavia, isto não significa que todos os blocos de dados contêm vídeo. Para um sistema em que existe uma única imagem representada em cada bloco de dados, e os bloco de dados são processados a um ritmo fixo, haveria vídeo em todos os blocos de dados, mesmo que seja um vídeo "minimalista" que consiste de um código representando um "sem alteração". Mas podem existir sistemas em que um bloco de dados possa representar mais ou menos que uma única imagem. Por exemplo, pode-se dar o caso de a informação de vídeo num bloco de dados, se de todo presente, seja sempre do mesmo número de *bits*. Dependendo da compressão, pode-se dar o caso de muitas imagens serem representadas num único bloco de dados. Em tal caso, alguns dos blocos seriam desprovidos de *bits* de vídeo. Dependendo do esquema de codificação empregue, o *bit* no campo 6 informa ao sistema operativo se existe de todo um campo 7. Se existe vídeo, o campo 7 contém a informação de vídeo, terminando com uma palavra de sincronização. Conforme acima referido, a codificação propriamente dita dos blocos de vídeo e áudio não faz parte da presente invenção. Embora esquemas de MPEG sejam preferidos, outros podem ser utilizados.

O campo 8 contém de entre nenhum até 16 *bits*. É de recordar que o

campo 6 da faixa de introdução contém 100 posições de *bits*, mas apenas N destes (onde o N máximo é 16) podem representar *bits* de valor 1 porque só podem existir no máximo 16 faixas de áudio no disco (das quais a M&E é considerada como sendo uma destas). Para cada uma destas N faixas, o campo 8 informa ao sistema operativo se existe algum áudio presente no bloco de dados. Existem assim X "1"s, até a um máximo de N. A primeira posição de *bits* de N do campo 8 corresponde à primeira faixa de idioma de áudio identificada no campo 6 da faixa de introdução. O segundo *bit* no campo 8 de um bloco de dados está associado ao segundo idioma de áudio representado no campo 6 da faixa de introdução, etc. A razão porque existem apenas N (máximo = 16) *bits* no campo 8 da figura 4, em vez de 100, é porque é sabido através da faixa de introdução quais os idiomas que podem estar presentes num bloco de dados. Não há razão para proporcionar 84 ou mais posições de *bits* em cada bloco de dados para indicar que os idiomas correspondentes não estão presente quando é sabido através da faixa de introdução que estes não se encontram em lugar algum no disco. É de recordar que o valor X na figura 4 não é igual ao valor N na figura 3. Esta última representa o número total de idiomas de áudio existentes em qualquer parte do disco, e o seu valor máximo é 16. O símbolo X representa quantos daqueles N estão realmente representados no actual bloco de dados.

O campo 9 contém X blocos de idioma de áudio. Suponhamos que existem 10 idiomas de áudio representados no disco, mas que apenas seis destes estão representados no actual bloco de dados. Neste caso, existiriam X sequências de *bits* correspondentes aos idiomas de áudio, cada um terminando com um caractere de escapar. O caractere de escapar é utilizado para separar os blocos de áudio uns dos outros. Se sempre que um bloco de áudio estiver presente este tiver uma duração fixa, então, uma vez que é sabido quantos blocos de áudio estão presente num bloco de dados através da informação no campo 8, não é necessário proporcionar uma palavra de sincronização no final do campo. O blocos de áudio

de comprimento variável requereriam uma palavra de sincronização no final do campo.

O campo 9 na faixa de introdução contém um valor de 0 a 63 que representa o número de "outras" faixas de áudio. Embora possam existir M tais "outras" faixas de áudio, conforme mostrado na figura 3, isto não significa que cada uma delas é representada no actual bloco de dados. O campo 10 em cada bloco de dados contém M *bits*, um para cada uma das "outras" faixas de áudio no disco. Se o actual bloco de dados contém realmente informação de *bits* para qualquer uma destas M faixas depende de se a correspondente posição de *bit* no campo 10 contém um 1. Se existirem Y "1"s e Y é inferior a M, isto significa que nem todas as "outras" faixas de áudio estão representadas no actual bloco de dados. O campo 11 contém Y "outras" faixas de áudio, cada uma terminando com um caractere de escapar. Será de apreciar que a maneira como as faixas de áudio e as "outras" faixas de áudio são representadas no bloco de dados são comparáveis.

Com referência novamente à figura 2, é de recordar que os *bits* de dados num bloco de dados são distribuídos para *buffers* de áudio, um *buffer* de vídeo, um *buffer* de *pan scan* e um *buffer* de subtítulos, assim como para o controlador mestre 41 através da linha 65 de COMANDO / DADOS. Até aqui, a representação dos blocos de áudio, "outras" faixas de áudio e um bloco de vídeo foram considerados na análise dos campos na figura 4. Antes de proceder com a representação dos dados dos subtítulos, todavia, é de entender que existe uma diferença na maneira como é representada a informação dos subtítulos, em oposição aos dados de áudio e vídeo. Estes últimos são representados numa base de bloco-por-bloco, e os *buffers* são continuamente atestados com novos dados de áudio e vídeo. Os subtítulos, por outro lado, não necessitam de mudar de imagem para imagem. De facto, um subtítulo nem será distinguido se este não ficar no

ecran por mais de uma imagem. Consequentemente, uma vez representados os dados do subtítulo no *buffer 59* se figura 2, este leva à formação de um subtítulo no ecran permanecendo lá até que nova informação de subtítulo for carregada no *buffer*. Para remover um subtítulo sem introduzir um novo, é carregado no *buffer* um novo subtítulo que consiste num campo em branco.

O campo 12 no bloco de dados consiste em P bits, cada um correspondendo a um dos P idiomas diferentes de subtítulos identificados no campo 15 da faixa de introdução. (É de recordar que a primeira posição em todos os campos de 100 bits correspondentes a idiomas não representam realmente um idioma, mas antes a M&E, de forma a que existem um máximo de 99 idiomas de subtítulos.) Qualquer subtítulo para o qual existe uma actualização no actual bloco de dados tem um 1 na sua posição correspondente no campo 12. Podem existir até Z "1"s, sendo o valor máximo de Z igual a P .

Para cada idioma de subtítulos para o qual exista uma actualização no actual bloco de dados, a actualização aparece no campo 13. Existem Z blocos de actualização, cada um terminando com um caractere de escapar. É importante compreender que um bloco de actualização pode ser um campo em branco. Esta é a maneira como um subtítulo é removido quando um novo subtítulo ainda não está para ocupar o seu lugar.

O campo 14 consiste num bit que pode ou não estar presente. O campo está presente apenas se o campo 21 na faixa de introdução for um 1. Em tal caso, a informação de *pan scan* está disponível no bloco de dados. Se a informação de *pan scan* estiver disponível, cada bloco de dados tem de dizer ao sistema operativo se contém realmente uma nova coluna de partida para o *pan scan*. O campo 14 é um único bit, uma bandeira, que indica se existe uma actualização de *pan scan*. Se o bit for um 1, o campo 15 é um número de coluna

de 9 bits, isto é, uma actualização de *pan scan*.

Finalmente, o campo 16 é um único *bit* que pode ou não estar presente, dependendo do valor do campo 30 na faixa de introdução. Esta bandeira de um *bit* na faixa de introdução diz ao sistema operativo se comandos “suplementares” e dados podem estar presentes no campo 17 de um bloco de dados. Se a bandeira de comando / dados presente for um 1, o bloco comando / dados é lido do campo 17. O campo termina com um caractere de escapar.

Um campo de bloco de dados contém assim até seis tipos diferentes de dados -- áudio, “outro” áudio, vídeo, informação de *pan scan*, subtítulos e um bloco de comando / dados. Estes são os seis tipos de informação que foram acima discutidos em relação com a figura 2, com desmultiplexador 63 a distribuir os diferentes blocos de informação para os *buffers* áudio, *buffer* vídeo, *buffer* de *pan scan*, *buffer* de subtítulos, e controlador mestre.

Processamento dos Campos dos Blocos de Dados

O processamento dos dados num bloco de dados é relativamente linear. O processamento mostrado no gráfico de operação da figura 6 ensambla com os campos dos blocos de dados propriamente ditos mostrados na figura 4.

Já foi descrito como o analisador 47 de número / apontador de bloco na figura 2 processa o número serial do bloco, versão, bandeira apontadora de dois bits e apontador contido nos campos 2-5 do bloco de dados. O campo seguinte é a bandeira de vídeo presente. Conforme mostrado na figura 6, se for determinado que dados de vídeo estão presentes, o *buffer* de vídeo 55 na figura 2 é carregado com o vídeo no campo 7. Se os dados de vídeo não estiverem presente, o *buffer* é simplesmente carregado com um marcador.

É importante perceber a necessidade de ter marcadores. A fim de o sistema operativo ser sempre capaz de sincronizar informação de vídeo, áudio, subtítulos, etc., este tem de ser capaz de dizer onde nos vários *buffers* diferentes está a informação do mesmo bloco de dados. Por outras palavras, o sistema operativo tem de saber qual parte dos dados áudio num *buffer* de áudio vai com qual parte dos dados vídeo num *buffer* de vídeo. Caso contrário os vários itens de informação não podem ser sincronizados uns com os outros. Proporcionando marcadores nos *buffers* para dados que não estão presente nos blocos de dados, o sistema operativo pode manter os vários itens de informação sincronizados uns com os outros.

De seguida, o sistema operativo olha para o campo 8 para determinar quantas das N faixas de áudio no disco (ver figura 3) estão realmente representadas no actual bloco de dados. O mesmo é verdade para as M "outras" faixas de áudio representadas o campo 10. Todos os dados de áudio e "outras" faixas de áudio são carregados nos seus respectivos *buffers*. O gráfico de operação mostra a sequência apenas para a primeira e última faixa de áudio. Em cada caso, é efectuado um teste para ver se a faixa de áudio ou "outras" faixas de áudio têm dados presentes no actual bloco de dados. Cada uma das faixas resulta em qualquer coisa ser carregado no seu respectivo *buffer* -- ou dados propriamente ditos seguidos de um marcador, ou somente um marcador.

Depois da informação áudio e vídeo, um bloco de dados contém actualizações de subtítulos. Se existir informação de actualização para os subtítulos num idioma seleccionado, este é carregado no *buffer* de subtítulos; caso contrário é apenas armazenado um marcador. Os três blocos que dizem respeito a subtítulos dizem respeito apenas a uma única faixa, esta correspondendo ao idioma do subtítulo seleccionado.

De seguida, é lida a bandeira de actualização de *pan scan* no campo 14. Se estiver presente informação de actualização de *pan scan*, este também é carregado, desta vez num *buffer* de *pan scan*. Se não estiver disponível qualquer informação nova, o marcador é simplesmente colocado no *buffer* de *pan scan* para indicar que outro bloco de dados passou sem qualquer nova informação de actualização de *pan scan*.


Finalmente, o sistema determina se estão disponíveis comandos ou dados (se o campo 30 da faixa de introdução disser que existem comandos ou dados nos blocos de dados). Se comandos / dados estiverem presente, i.e. o campo 16 no bloco de dados for um 1, este é carregado do campo 17 para a memória no controlador mestre 41 da figura 2. Se não existirem comandos ou dados disponíveis apenas é carregado um marcador na memória do microprocessador.

Deve ser notado que nenhuma das sequências de processamento da figura 6 mostram a verificação a ser feita sobre se o respectivo tipo de informação está disponível no disco em primeiro lugar. Mas é de entender que um teste como “está presente o comando / dados?” consiste na realidade em duas partes. Primeiro, a bandeira de comandos / dados do bloco de dados no campo 30 da faixa de introdução é um 0 ou um 1? Se for 0, os comandos e dados nem são procurados durante o processamento do bloco de dados. Por outro lado, se comandos ou dados podem estar presente num bloco de dados como resultado de a bandeira de dados no campo 30 da faixa de introdução ser um 1, então cada bloco de dados tem o seu campo 16 verificado para ver se a bandeira de comandos / dados presente é um 1. É o valor da bandeira no campo de bloco de dados que determina se apenas é carregado um marcador, ou um marcador seguido de *bits* de dados. Observações semelhantes aplicam-se a outras sequências. Por exemplo, não há nenhuma razão para verificar se uma actualização *pan scan* está presente se através da faixa de *lead-in* for determinado

- 78 -

que informação de *pan scan* não existe em qualquer lugar no disco.

Lisboa, 14 de Março de 2001


LUIS SILVA CARVALHO
Agente Oficial da Propriedade Industrial
RUA VICTOR CORDON, 14
1200 LISBOA

REIVINDICAÇÕES

1. Um sistema para tocar portadores de *software* que distingue entre portadores de *software* de editoras autorizadas e não autorizadas, contendo cada portador de *software* de uma editora autorizada uma pluralidade de blocos de dados a partir das quais um sinal pode ser gerado aquando do tocar do portador e um código de autorização do portador que é em função de informação predeterminada armazenada no portador, compreendendo meios para a leitura da dita informação predeterminada e, utilizando a dita função, derivando um código computado; meios para a comparação do dito código de autorização do portador e dito derivado código computado; meios para permitir selectivamente o tocar do dito portador de *software* de acordo com a operação dos ditos meios de comparação.

2. Um sistema de acordo com a reivindicação 1 em que o dito código de autorização do portador é uma encriptação com uma chave privada de um par de um sistema críptico de chave pública, e ditos meios de comparação descriptam o dito código de autorização do portador com o par de chaves públicas e compara o código descriptado com o dito código computado.

3. Um sistema de acordo com a reivindicação 2 em que o dito portador de *software* inclui uma secção de *lead-in* contendo informação pertinente ao conteúdo de *software* do portador, e a dita informação predeterminada inclui tal informação de *lead-in*.

4. Um sistema de acordo com a reivindicação 1 incluindo ainda meios de geração de um sinal aquando do tocar de um portador num de entre múltiplos modos disponíveis; o dito portador contendo um código de autorização

diferente, que é uma respectiva função da informação predeterminada armazenada no portador, para cada modo que é autorizado; ditos meios de leitura derivam um código computado utilizando a respectiva função para o modo seleccionado; e ditos meios de comparação operam para selectivamente permitir o tocar do dito portador de *software* apenas em modos autorizados.

5. Um sistema de acordo com a reivindicação 4 em que cada código de autorização do portador é uma encriptação com uma chave privada de um par de um sistema crítico de chave pública, e ditos meios de comparação descriptam cada dito código de autorização do portador com o par de chaves públicas e compara o código descriptado com o dito código computado.

6. Um sistema de acordo com a reivindicação 5 em que o dito portador de *software* inclui uma secção de *lead-in* contendo informação pertinente ao conteúdo de *software* do portador, e a dita informação predeterminada inclui tal informação de *lead-in*.

7. Um sistema de acordo com a reivindicação 1 em que o dito portador de *software* inclui uma secção de *lead-in* contendo informação pertinente ao conteúdo de *software* do portador, e a dita informação predeterminada inclui tal informação de *lead-in*.

8. Um método para tocar portadores de *software* que distingue entre portadores de *software* de editoras autorizadas e não autorizadas, contendo cada portador de *software* de uma editora autorizada uma pluralidade de blocos de dados a partir da qual um sinal pode ser gerado aquando do tocar do portador e um código de autorização que é uma função de informação predeterminada armazenada no portador, compreendendo os passos para a leitura da dita informação predeterminada e, utilizando a dita função, derivando um código

computado; comparando o dito código de autorização do portador e dito derivado código computado; e permitindo selectivamente o tocar do dito portador de *software* de acordo com a operação dos ditos meios de comparação.

9. Um método de acordo com a reivindicação 8 em que o dito código de autorização do portador é uma encriptação com uma chave privada de um par de um sistema críptico de chave pública, e dito passo de comparação descripta o dito código de autorização do portador com o par de chaves públicas e compara o código descriptado com o dito código computado.

10. Um método de acordo com a reivindicação 9 em que o dito portador de *software* inclui uma secção de *lead-in* contendo informação pertinente ao conteúdo de *software* do portador, e a dita informação predeterminada inclui tal informação de *lead-in*.

11. Um método de acordo com a reivindicação 8 incluindo ainda meios de geração de um sinal aquando do tocar de um portador num de entre múltiplos modos disponíveis; o dito portador contendo um código de autorização diferente, que é uma respectiva função da informação predeterminada armazenada no portador, para cada modo que é autorizado; e em que os ditos passos de leitura derivam um código computado utilizando a respectiva função para o modo seleccionado; e ditos passos de comparação operam para selectivamente permitir o tocar do dito portador de *software* apenas em modos autorizados.

12. Um método de acordo com a reivindicação 11 em que cada código de autorização do portador é uma encriptação com uma chave privada de um par de um sistema críptico de chave pública, e os ditos passos de comparação descriptam o dito código de autorização do portador com o par de chaves

públicas e compara o código descriptado com o dito código computado.

13. Um método de acordo com a reivindicação 12 em que o dito portador de *software* inclui uma secção de *lead-in* contendo informação pertinente ao conteúdo de *software* do portador, e a dita informação predeterminada inclui tal informação de *lead-in*.

14. Um método de acordo com a reivindicação 8 em que o dito portador de *software* inclui uma secção de *lead-in* contendo informação pertinente ao conteúdo de *software* do portador, e a dita informação predeterminada inclui tal informação de *lead-in*.

Lisboa, 14 de Março de 2001



LUIS SILVA CARVALHO
Agente Oficial da Propriedade Industrial
RUA VICTOR CORDON, 14
1200 LISBOA

FIG. 1

ESTADO DE ARTE
gravador VHS

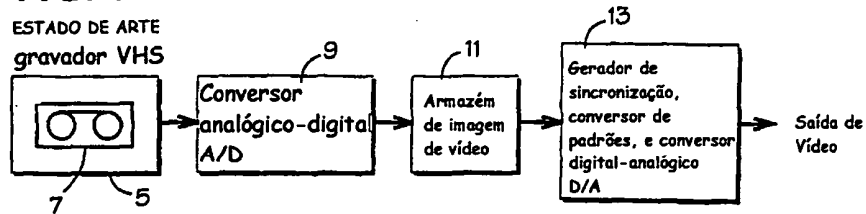


FIG. 8

ESTADO DE ARTE

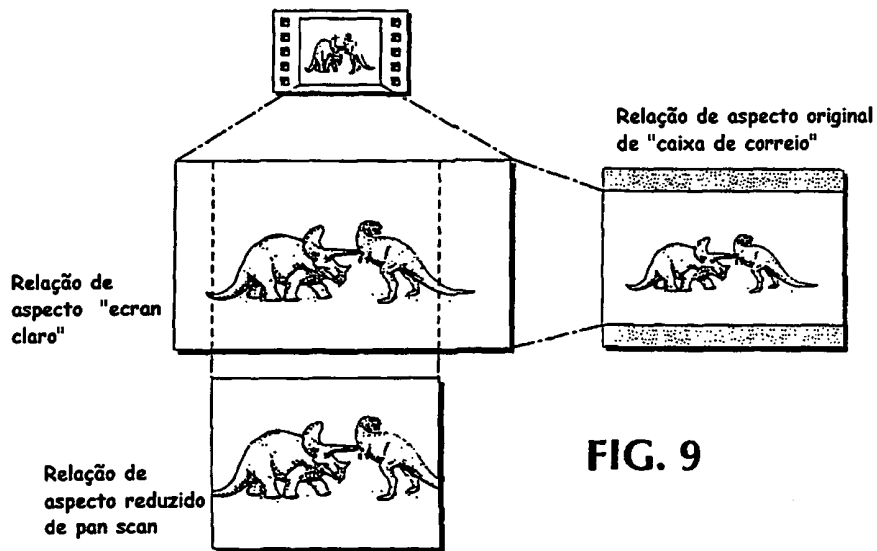
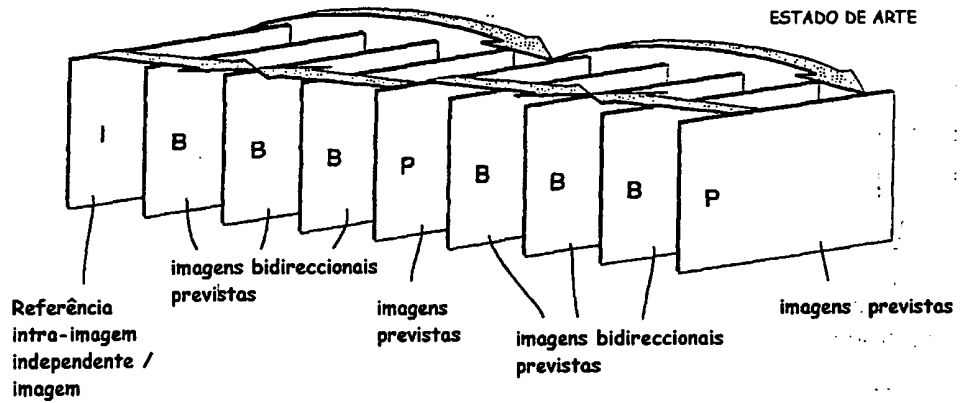


FIG. 9

Very Sales Company

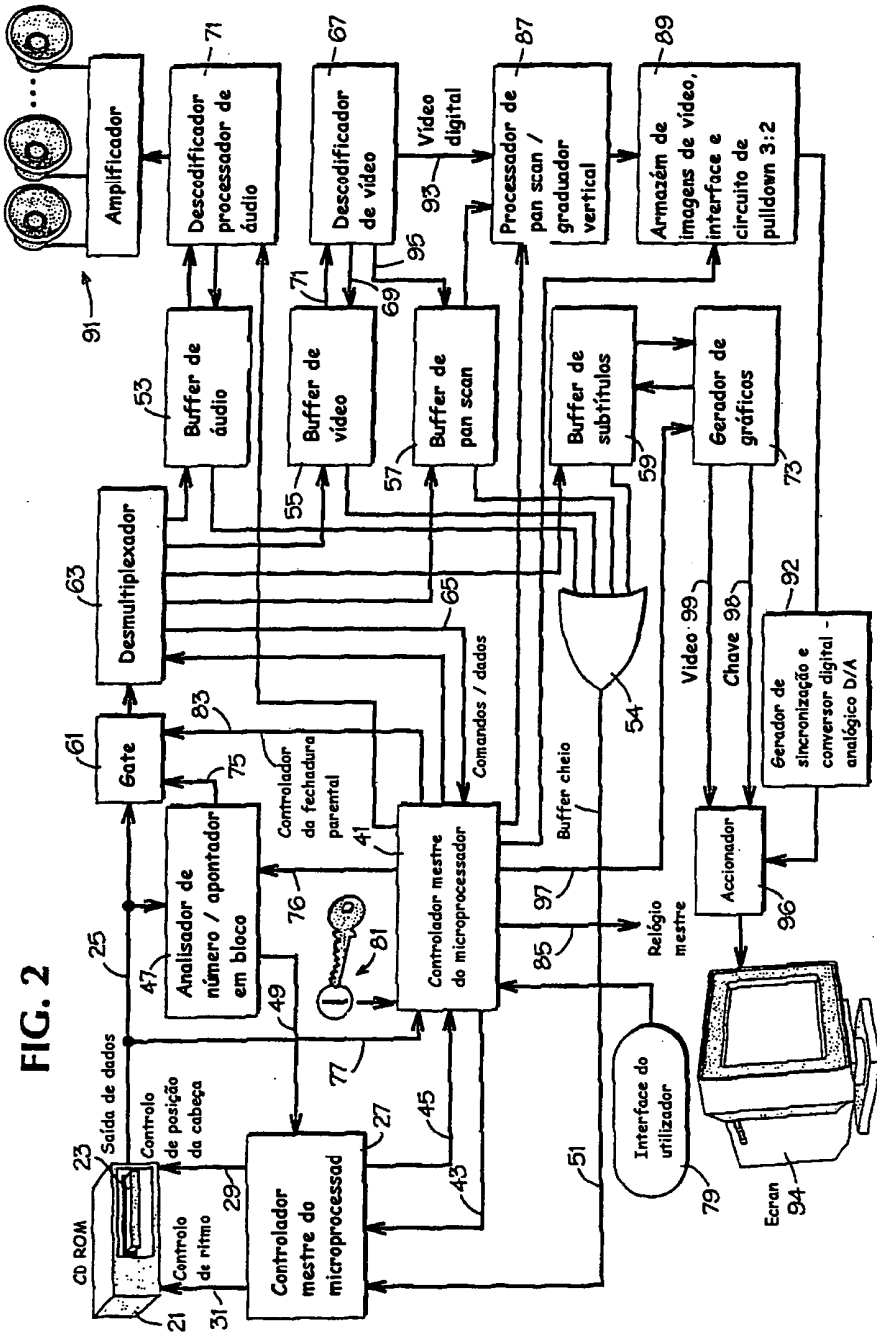


FIG. 2

FIG. 3

BITS			COMENTÁRIOS	
MIN	MAX			
1			bits de sincronização de introdução	
2	40	40	territórios autorizados	
3	1	1	bandeira de software especial	
4	0	UNDET	software especial com terminação em palavra de sincronização	
5	12	12	padrões autorizados	
6	100	100	idiomas de áudio disponíveis, M & E mais 99	N "1"s, MAX 16
7	0	48	tipos de faixa	N x 3
8	0	64	codificação das faixas	N x 4
9	6	6	número de "outras" faixas de áudio	M = 0 TO 63
10	0	252	codificação para as "outras" faixas de áudio	M x 4
11	100	100	idiomas de apresentação disponíveis	P "1"s, MAX 99
12	1	1	bandeira do software especial de mistura / eliminação	
13	0	UNDET	software especial de mistura / eliminação com terminação em palavra de sincronização	
14	0	UNDET	sequências P x M cada um terminando com caractere de escapar	
15	100	100	idiomas de subtítulos disponíveis	R "1"s, MAX 99
16	4	4	código de versão múltipla	
17	1	1	bandeira de software de versão especial	
18	0	UNDET	software de versão especial com terminação em palavra de sincronização	
19	1	1	bandeira de disponibilidade de vídeo	
20	1	1	relação de aspecto de base	
21	1	1	disponibilidade de vídeo	
22	20	20	número total de blocos de dados	
23	0	20	número de blocos de dados na versão A	
24	0	20	número de blocos de dados na versão B	
25	4	4	ritmo original de imagens	
26	10	10	factor temporal do bloco	
27	0	UNDET	tabela de conteúdos para a primeira versão para cada capítulo: número do capítulo 8 bits número serial do bloco de 20 bits do bloco de partida bloco de duração do capítulo de 20 bits idiomas de apresentação dos capítulos disponíveis (100 bits) sequências de idiomas identificando os capítulos cada um terminando caractere de escapar	
28	0	UNDET	tabela de conteúdos para a segunda versão	
29	100	1200	código de autorização encriptado para cada padrão com terminação em palavra de sincronização	
30	1	1	bandeira de comandos / dados de blocos de dados	
31	1	1	bandeira de software "suplementar"	
32	0	UNDET	software "suplementar" com terminação em palavra de sincronização	

FIG. 4

	BITS		COMENTÁRIOS
	MIN	MAX	
1	32	32	Palavra de sincronização
2	20	20	Número serial do bloco
3	2	2	Versão (A, B ou comum)
4	0	2	Bandeira apontadora de 2 bits
5	0	20	Apontador
6	0	1	Bandeira de vídeo presente
7	0	UNDET	Bloco de vídeo com terminação em palavra de sincronização
8	0	16	Faixas de áudio presente X "1"s, MAX=16
9	0	UNDET	X blocos de idiomas de áudio, cada um terminando com caractere de escapar
10	0	63	"outras" faixas áudio presente Y "1"s, MAX=63
11	0	UNDET	Y "outras" faixas de áudio presente cada um terminando com caractere de escapar
12	0	99	Actualização de subtítulos presente Z "1"s, MAX=99
13	0	UNDET	Z blocos de actualização de subtítulos, cada um terminando com caractere de escapar
14	0	1	Bandeira de actualização de pan scan
15	0	9	Actualização de pan scan
16	0	1	Bandeira de comandos / dados de presente
17	0	UNDET	Bandeira de comandos / dados terminando com caractere de escapar

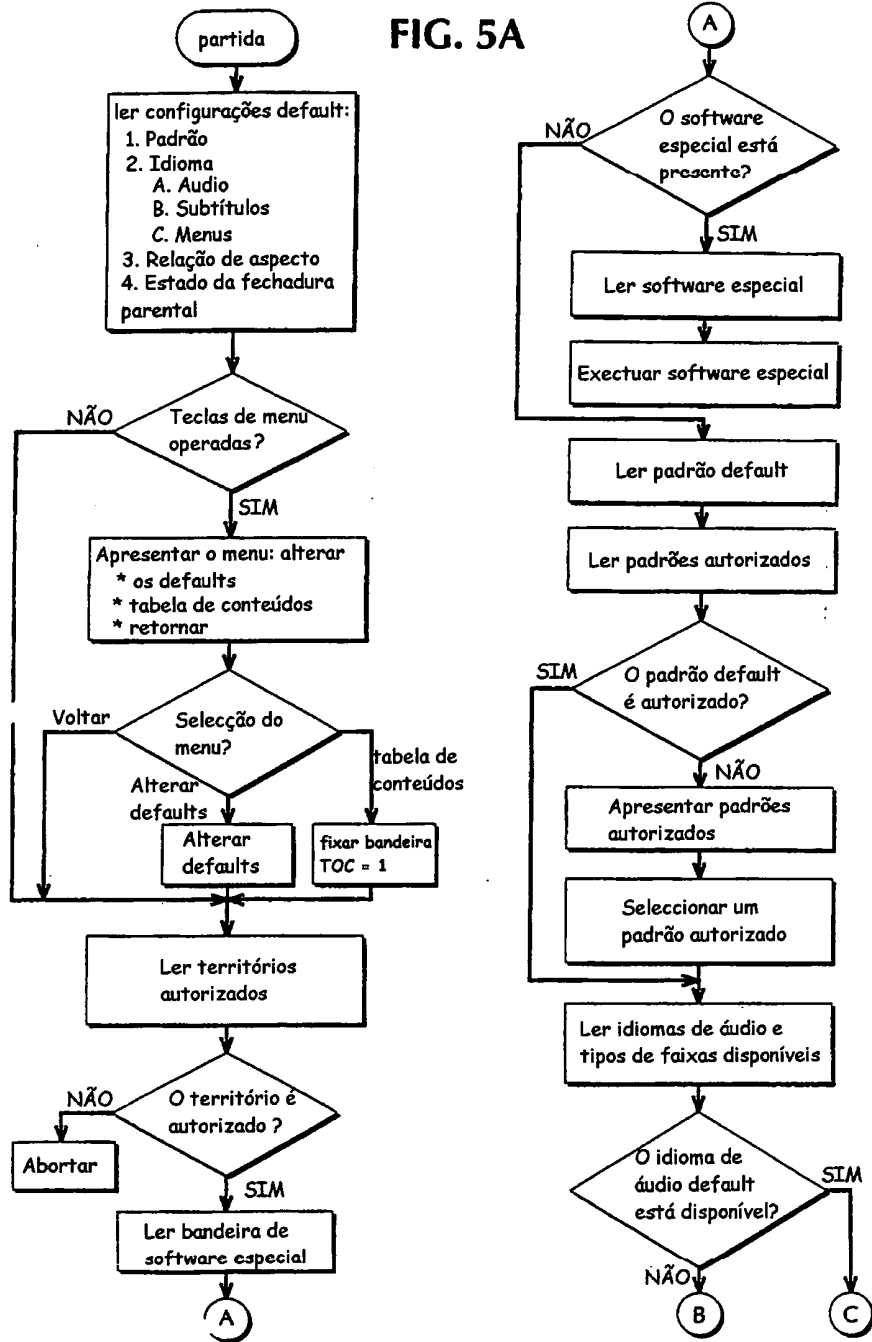


FIG. 5b

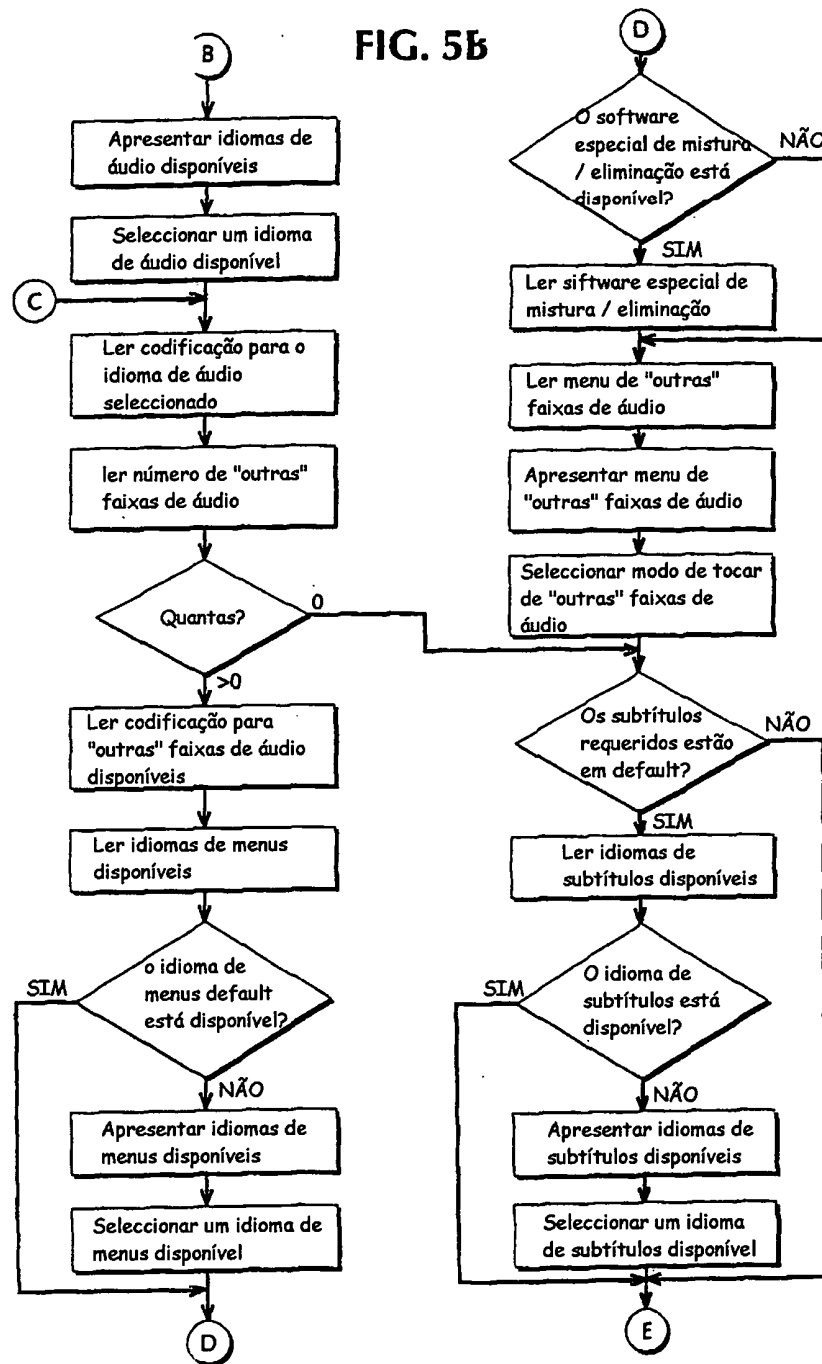
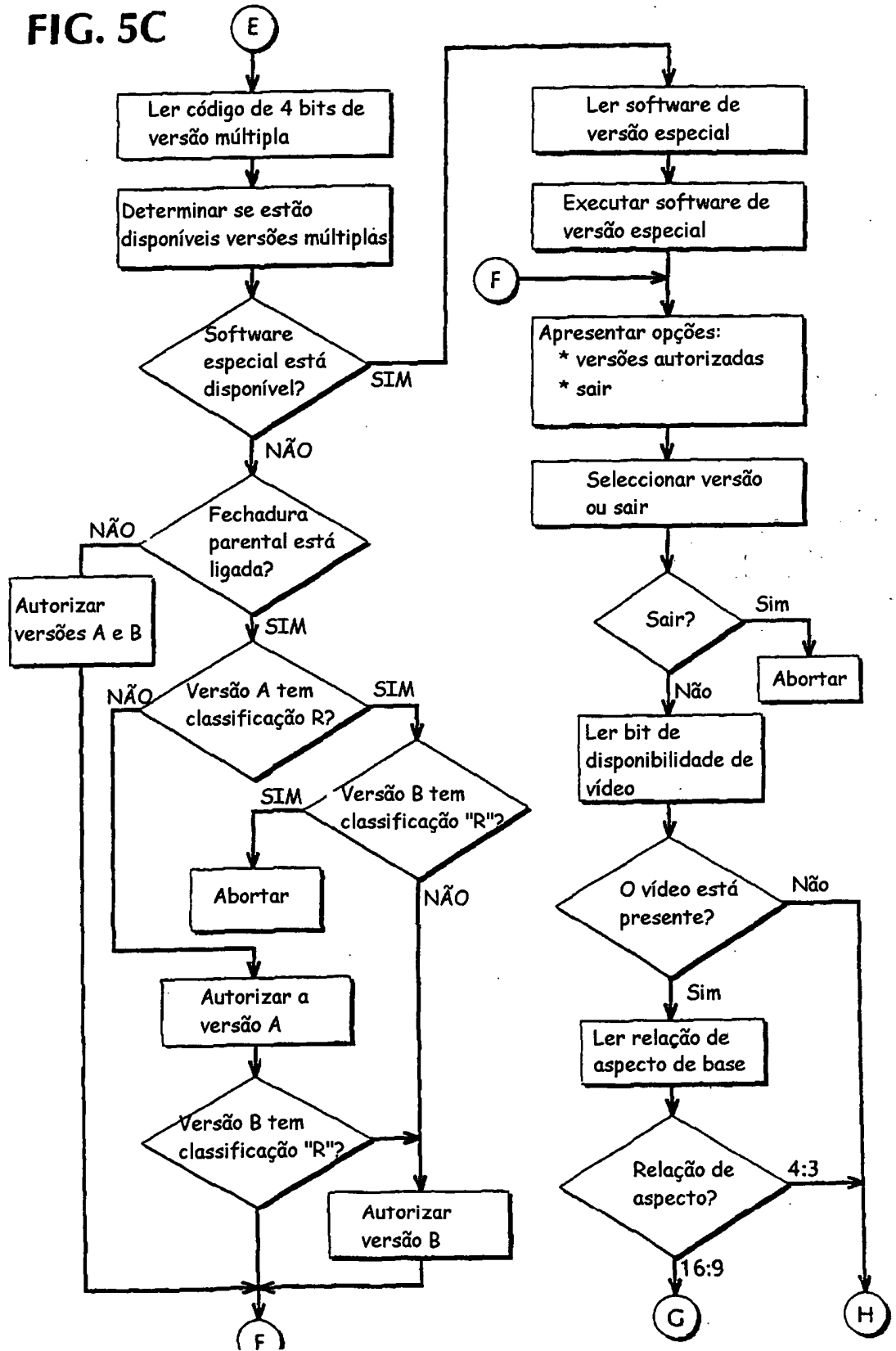


FIG. 5C



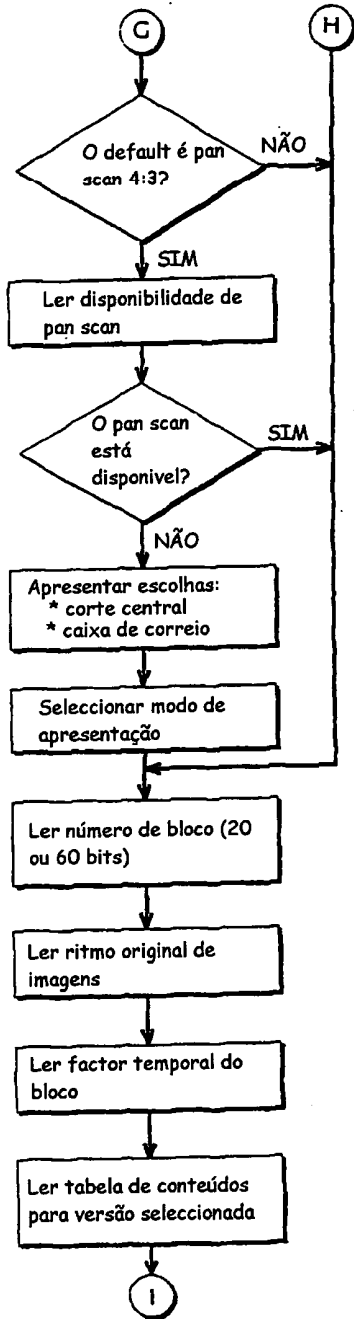


FIG. 5D

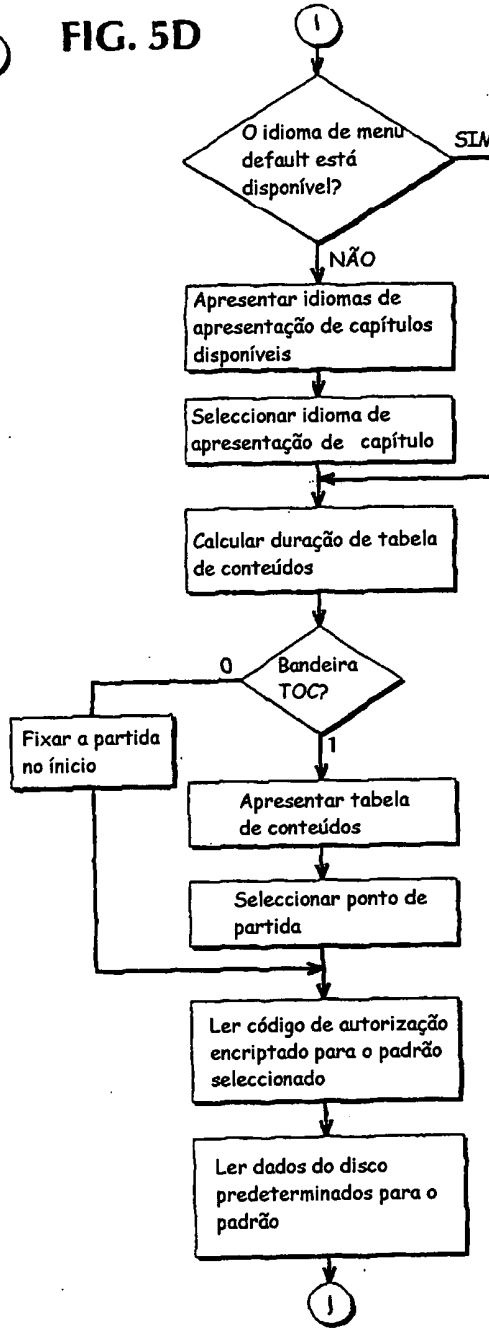


FIG. 5E

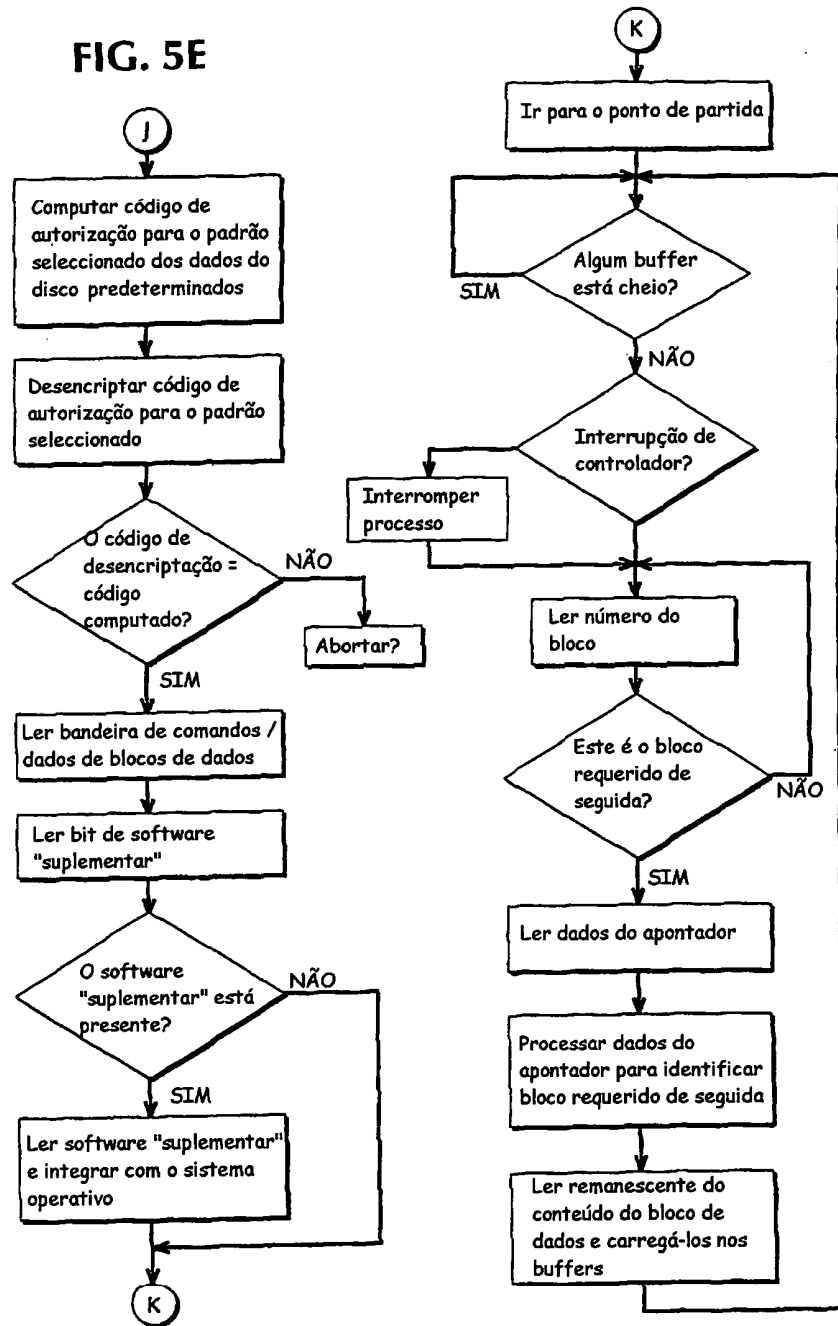
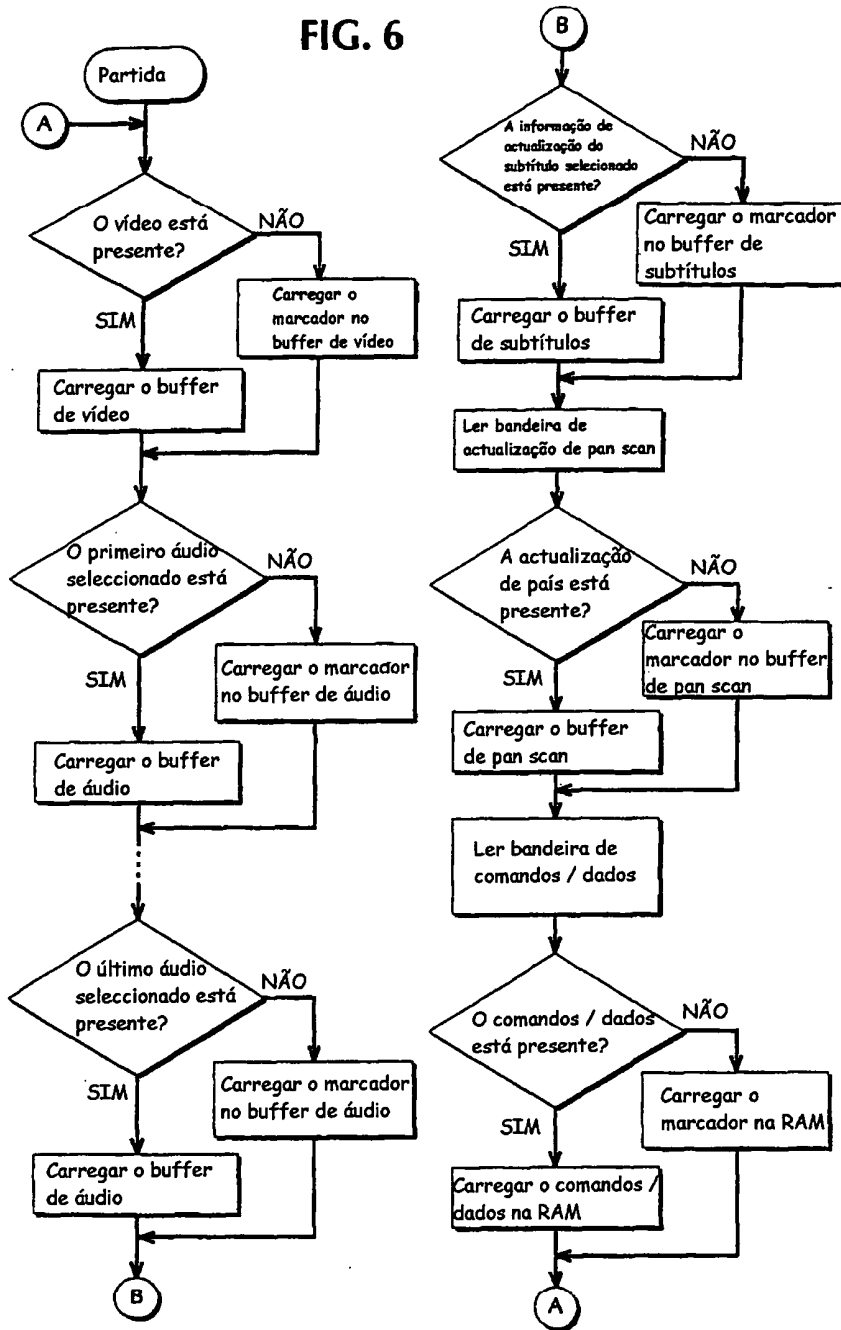


FIG. 6



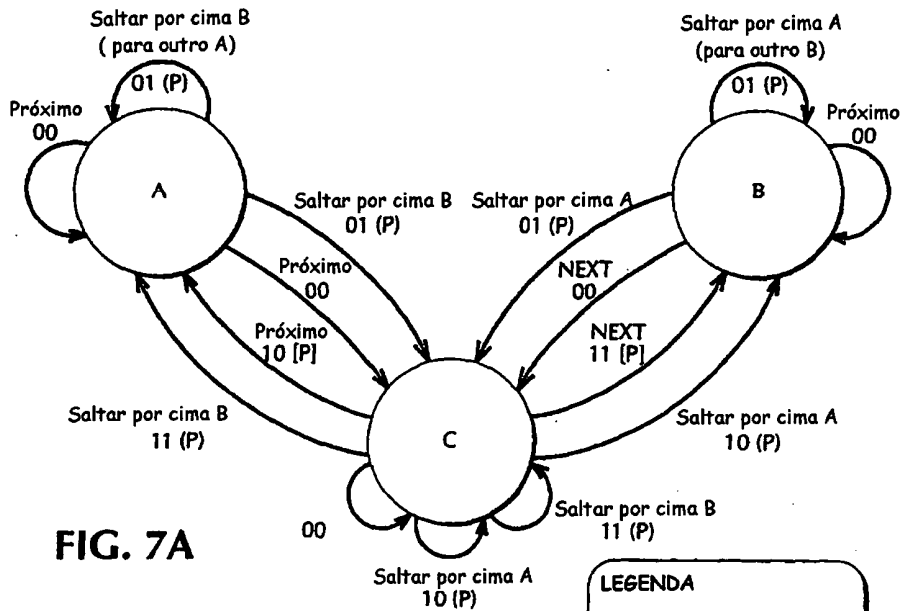


FIG. 7A

CODE

- 00 = Continuar para o bloco seguinte
- 01 = Saltar para a mesma versão ou comum, um apontador
- 10 = Ramificar de um comum
bloco seguinte é um A, apontador para um B ou um C
(se versão A está a ser tocada, continuar para o bloco seguinte)
(se versão B está a ser tocada, saltar para o bloco identificado pelo apontador)
- 11 = Ramificar de um comum
bloco seguinte é um B, apontador para um A ou um C
(se versão A está a ser tocada, saltar para o bloco identificado pelo apontador)
(se versão B está a ser tocada, continuar para o bloco seguinte)

LEGENDA

- 10 (P) } Utilizar
- 11 (P) } apontador P
- 10 [P] } Ignorar
- 11 [P] } apontador P

FIG. 7B

