



República Federativa do Brasil
Ministério da Indústria, Comércio Exterior
e Serviços
Instituto Nacional da Propriedade Industrial

(11) PI 0411541-4 B1

(22) Data do Depósito: 16/06/2004

(45) Data de Concessão: 18/09/2018



(54) Título: MÉTODO E APARELHO PARA PROCESSAR PACOTES NULOS EM UM RECEPTOR DE MÍDIA DIGITAL

(51) Int.Cl.: H04N 5/04; H03M 13/09

(30) Prioridade Unionista: 18/06/2003 US 60/479.397

(73) Titular(es): THOMSON LICENSING

(72) Inventor(es): WEIXIAO LIU; IVONETE MARKMAN; MATTHEW THOMAS MAYER

(85) Data do Início da Fase Nacional: 16/12/2005

"MÉTODO E APARELHO PARA PROCESSAR PACOTES NULOS EM
UM RECEPTOR DE MÍDIA DIGITAL"

CAMPO DA INVENÇÃO

A presente invenção diz respeito a transmissão e
5 recebimento de dados multimídia, incluindo vídeo e áudio di-
gital, e, mais particularmente, a um método e aparelho para
sincronizar e entregar de forma confiável um fluxo MPEG-2
difundido em um sistema de transmissão digital como esse à
camada de transporte do receptor.

10 ANTECEDENTES DA INVENÇÃO

Sistemas de transmissão digital oferecem aos con-
sumidores dados multimídia de alta qualidade, incluindo flu-
xos de áudio e vídeo compactados. Para difusores, a compres-
são de dados permite que diversos programas sejam entregues
15 na mesma largura de banda analógica. Os componentes de áudio
e vídeo de um programa são compactados na fonte e multiple-
xados com o tempo com outros programas e informação de sis-
tema necessária para recriar o programa original. O multi-
plexado digital é processado por uma camada física e trans-
20 mitido ao consumidor. No terminal do consumidor, o receptor
processa o sinal para recuperar os fluxos digitais multiple-
xados, extrai o programa de interesse, e decodifica o áudio
e vídeo compactado para apresentação em uma exibição de ví-
deo/áudio, tal como uma televisão.

25 Para promover o desenvolvimento de componentes in-
teroperáveis de diferentes fabricantes, o padrão de compres-
são internacional MPEG-2 foi desenvolvido. O padrão não es-
pecifica as técnicas para codificação, multiplexação e deco-

dificação dos fluxos de bits, mas somente o formato dos dados. Isto deixa uma oportunidade para os fabricantes diferenciarem seus produtos pela maneira na qual eles usam recursos, tais como silício, energia de processador e memória, e pela suas capacidades de cancelar e recuperar erros. O padrão é composto de três partes primárias que cobrem sistemas, vídeo e áudio. As partes de vídeo e áudio especificam o formato dos dados de vídeo e áudio compactados, enquanto que a parte dos sistemas especifica os formatos para multiplexação dos dados de áudio e vídeo para um ou mais programas, bem como informação necessária para a recuperação dos programas.

Os padrões ANSI/SCTE 07 2000 (antigamente SCTE DVS 031) e ITU-T J.83B, que são praticamente idênticos, descrevem um sistema de transmissão digital para distribuição a cabo de serviços de vídeo, som e dados. Em particular, o padrão ANSI/SCTE 07 2000 descreve o padrão adotado para transmissão a cabo digital nos Estados Unidos. Em ambos os padrões, o formato de dados que entra na camada física (codificação e modulação de canal) é considerado transporte MPEG-2.

Na camada física, o enquadramento MPEG é a camada externa de processamento. Na transmissão, o bloco de quadros MPEG é seguido pelo codificador de Correção de Erro Direta (FEC) e o Modulador de Amplitude pela Quadratura (QAM) 64 ou 256. Um sistema FEC é uma classe de métodos para controlar erros em um sistema de comunicação unidirecional, tal como um fluxo MPEG-2. Um codificador FEC transmite informação ex-

P I D 1 5 4

qüentes contendo identificação de serviço, informação de
cruzamento de sinais e controle. 24

O cabeçalho do pacote de transporte de quatro by-
tes é seguido por 184 bytes de dados Mpeg-2 ou auxiliares. O
5 cabeçalho de pacote de transporte é disposto na seguinte or-
dem:

a) Byte de sincronismo: 8 bits consistindo de um
valor fixo de 0x47 (47hex);

b) Transport_error_indicator: 1 bit indicando um
10 erro de bit incorrigível no pacote de transporte atual. Esta
informação pode ser estabelecido pelo transmissor ou pelo
receptor.

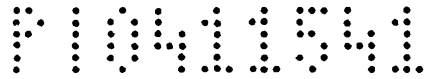
c) Payload_unit_start_indicator: 1 bit indicando a
presença de um novo pacote PES (Fluxo Elementar Empacotado)
15 ou uma nova seção TS_PSI (Informação Específica de Fluxo de
Transporte - Programa).

d) Transport_priority: 1 bit indicando uma maior
prioridade do que outros pacotes.

e) PID: ID de pacote de 13 bits. Valores 0 e 1 são
20 pré-atribuídos, enquanto os valores 2 a 15 são reservados.
Valores 0x0010 a 0x1FFE podem ser atribuídos pela Informação
Específica do Programa (PSI). O valor 0x1FFF é reservado pa-
ra pacotes nulos.

f) Controle Transport_scrambling: 2 bits indicando
25 o modo cruzamento de sinais da carga útil do pacote.

g) Adaptation_field_control: 2 bits indicando a
presença de um campo de adaptação ou campo de dados de carga
útil.



h) Continuity_counter: 4 bits, representando um continuity_counter por valor de PID. Ele incrementa com cada pacote de fluxo de transporte não repetido que tem o valor PID correspondente. 25

5 Um fluxo MPEG-2 difundido pode conter diversos programas multiplexados de dados de áudio e vídeo, juntamente com os dados de sistema necessários, e cada pacote de dados é identificado por uma etiqueta ou PID exclusiva dentro do cabeçalho do pacote.

10 Uma seção do fluxo de dados MPEG-2 poderia ser fortemente predisposta em um PID particular, ou uniformemente ponderada por muitos PIDs, ou poderia conter uma alta porcentagem de pacotes nulos. Erros, conhecidos e desconhecidos, são inerentes na entrega de fluxo de transporte, e 15 podem ocorrer a qualquer momento. Erros desconhecidos, tais como corrompimento de bit ou perda de dados, podem ocorrer em qualquer posição de bit do fluxo, e podem levar a erro no transporte para comportamento não usual.

O byte de sincronismo de MPEG-2 é destinado a facilitar delimitação de pacotes em um decodificador. Entretanto, diferente de muitos outros padrões de transmissão digital, o método usado para sincronização MPEG-2 na camada física do sistema de transmissão a cabo digital é desacoplado da sincronização de Correção de Erro Direta (FEC). Primeiro, 20 o pacote MPEG-2 não contém um número inteiro de quadros FEC, ou mesmo palavras código Reed Solomon (RS). Codificação Reed-Solomon (RS), usando um código (128, 122), fornece codificação e decodificação de bloco para corrigir até três sím-



bolos de 7 bit dentro de um bloco RS. Conseqüentemente, os pacotes MPEG-2 e os quadros FEC, ou os pacotes MPEG-2 e as palavras código RS, são assíncronas uns em relação aos outros. Segundo, o byte de sincronismo é substituído dentro do bloco de enquadramento MPEG no sítio de transmissão por meio de uma soma de verificação de paridade que é um co-conjunto de um código de bloco linear de verificação de paridade FIR.

Conseqüentemente, o bloco de enquadramento MPEG no sítio receptor precisa decodificar este código de bloco de verificação de paridade a fim de recuperar o byte de sincronismo e em seguida travá-lo. Ele então entrega sincronização de pacotes MPEG aos blocos receptores à jusante, incluindo o bloco de transporte. A saída deste bloco pode incluir um relógio de saída, incluindo o bloco de transporte. A saída deste bloco pode incluir um relógio de saída, o fluxo de dados, em formato serial ou paralelo, um sinal de sincronismo identificando a posição do byte de sincronismo no fluxo de dados, um sinal válido identificando quando dados estão presentes no fluxo de dados de saída e um sinal de erro identificando se o pacote é considerado inválido (erros incorrigíveis) ou livre de erros.

Este recurso de desacoplamento de sincronização é incorporado com uma camada adicional de processamento para fazer uso da capacidade de levar informação do byte de sincronismo. No transmissor, uma soma de verificação de paridade que é um co-conjunto de um código de bloco linear de verificação de paridade (LBC ou FIR-PCC) de FIR (resposta de impulso finito) é usado em substituição a este byte de sin-

cronismo, com o propósito de melhor funcionalidade de deli-
 neação de pacotes, e capacidade de detecção de erro indepen-
 dente da camada FEC. A soma de verificação de paridade é
 computada nos 187 bytes adjacentes, que constituem o conteú-
 5 do de pacotes MPEG-2 imediatamente anterior (byte de sincro-
 nismo menos). As verificações de paridade do código de bloco
 são computadas no receptor observando a saída de uma respos-
 ta de impulso finita (FIR), filtro linear invariante com o
 tempo (binário). A estrutura de verificação de paridade é
 10 baseada em uma seqüência PN gerada por um polinômio primiti-
 vo (binário).

No lado do transmissor, a soma de verificação é
 computada passando os 1496 bits de carga útil através de um
 registro de deslocamento de realimentação (LFSR linear) des-
 crito pela equação seguinte:
 15

$$f(X) = [1 + X^{1497} b(X)] / g(X), \text{ onde } g(X) = 1 + X + X^5 + X^6 + X^8 \text{ e } b(X) = 1 + X + X^3 + X^7.$$

Um deslocamento de 67Hex é adicionado a este re-
 sultado da soma de verificação para melhores propriedades de
 autocorrelação, e faz com que um resultado 47Hex seja produ-
 zido durante uma operação de decodificação de síndrome,
 20 quando uma palavra código válida está presente. Esta estru-
 tura permite uma implementação computacionalmente eficiente
 do filtro FIR de verificação de paridade, de uma maneira re-
 cursiva, que é em geral auto-sincronizante, e, portanto, su-
 porta sincronização de pacotes e detecção de erro simultâ-
 25 nea. O decodificador computa uma soma de verificação corre-

diça no fluxo de dados serial, usando a detecção de uma palavra código válida para detectar o início de um pacote. 28

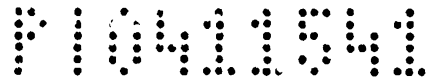
Uma matriz de verificação de paridade é usada pelo decodificador para identificar uma soma de verificação válida. Um gerador de síndrome pode também ser empregado. O código foi designado de maneira tal que, quando os 188 bytes apropriados do fluxo de bits (incluindo a soma de verificação) forem multiplicados contra a matriz de verificação de paridade, um casamento positivo é indicado, quando o produto calculado produzir um resultado 47Hex. (Note que a soma de verificação é calculada com base nos 187 bytes anteriores, e não nos 187 bytes por ser recebidos pelo decodificador de sincronismo MPEG-2. Isto é o contrário da sintaxe convencional de um pacote MPEG, no qual o byte de sincronismo é normalmente descrito como o primeiro byte de um pacote recebido).

A Figura 1 é um diagrama de blocos que representa um bloco de enquadramento MPEG da tecnologia anterior 200 no terminal do receptor. O fluxo de dados é serializado e o Fluxo de Dados Serial é transmitido através do gerador de síndrome 210 que é especificado no padrão ANSI/SCTE 07 2000 (ver, por exemplo, o gerador de síndrome ilustrado na Figura 3 desse padrão). O detector de síndrome 220 compara a saída do gerador de síndrome 210 com 47Hex para diversos pacotes, NP, e um limiar programável, Sync_thresh, e estabelece se um byte de sincronismo foi realmente detectado. Por exemplo, se durante NP pacotes, o número de saídas de síndrome igual a 47Hex for maior ou igual a Sync_thresh, então um byte de

sincronismo foi detectado. Um Lock_flag indica se o byte de sincronismo foi ou não detectado dentro do fluxo de dados, por exemplo, sendo um 1 ou 0 lógico, respectivamente. Um Sync_flag indica a posição do byte de sincronismo dentro do
5 fluxo de dados, por exemplo, sendo 1 lógico durante o byte de sincronismo e, caso contrário, 0. Uma vez que uma condição de alinhamento travado seja estabelecida, a ausência de uma palavra código válida no local esperado indicará um erro de pacote. O Error_flag do pacote anterior pode então ser
10 estabelecido em 1; caso contrário, o pacote é considerado livre de erros, e o Error_flag é estabelecido em 0.

Em um caminho paralelo, o Fluxo de Dados Serial original é devidamente atrasado (pelo bloco de Atraso 230) e transmitido ao bloco reinserir sincronismo de MPEG 240, em
15 que o byte de sincronismo é reinserido no lugar da soma de verificação de paridade (que foi criada no bloco de enquadramento MPEG do transmissor, não mostrado). Conseqüentemente, a saída do fluxo de dados para a camada de transporte é um fluxo de transporte MPEG-2 padrão rearmazenado. Estes da-
20 dos podem sair tanto no modo serial como paralelo. Dois sinais adicionais não mostrados na Figura 1 são também transmitidos à camada de transporte: o relógio e o sinal válido ou habilitado associado com os dados.

O recurso de desacoplamento de sincronização de
25 MPEG-2 foi destinado a introduzir a flexibilidade, por exemplo, para permitir que o sistema carregue pacotes do Modo de Transferência Assíncrona (ATM) facilmente sem interferir com sincronização ATM. Entretanto, uma consequência desprezível-



osa deste recurso é a maior probabilidade de "falsa trava" ³⁰
no detector de síndrome da tecnologia anterior dentro do
bloco de enquadramento MPEG na Figura 1.

Isto acontece em virtude de o código de bloco de
5 soma de paridade codificado no lado do transmissor não ser
muito potente, e seu decodificador (gerador síndrome 210) no
terminal receptor poder indicar diversos lugares em um pacote
onde um possível byte de sincronismo poderia ser encontrado,
quando somente um for o correto. Isto pode ocorrer
10 mesmo quando o FEC estiver perfeitamente travado e entregar
um fluxo de dados livre de erros. Felizmente, o fluxo de dados
típico em geral não apresenta uma característica periódica e,
depois de ser processado pelo decodificador de bloco de
verificação de paridade, diferentes pacotes tenderão ter
15 as posições de sincronismo corretas em comum. Entretanto, no
caso de um fluxo de dados periódico, tal como, por exemplo,
um fluxo de dados que tem um número considerável de pacotes
nulos, este problema se torna grave, e o Detector de trava
de sincronismo (Detector de Síndrome 220) da tecnologia an-
20 terior pode travar de maneira falsa em uma das diversas po-
sições de byte de sincronismo erradas no pacote identificado
pelo decodificador de blocos de soma de paridade, e transmitir
pacotes inválidos ao bloco de transporte, mesmo quando o
FEC for perfeitamente travado, e entregar um fluxo de dados
25 livre de erros. Desde que haja pacotes suficientes multiple-
xados no fluxo de dados em uma base regular, isto pode ser o
bastante para manter o detector de trava falsamente travado
por um período prolongado.

Depois de uma detecção de travamento (sincronização) o reinsersor de sincronismo MPEG 240 dentro do bloco de enquadramento MPEG da tecnologia anterior insere o byte de sincronismo na posição do byte de sincronismo identificado pelo decodificador de bloco de verificação de paridade, sai com o sinal Sync_flag, o sinal Error_flag, os sinais válidos e de relógio, e transmite o fluxo de dados à camada de transporte. No caso de um travamento falso, a camada de transporte não pode identificar facilmente um pacote errado, uma vez que ela está recebendo 188 bytes de dados, com um primeiro byte na expectativa de ser o byte de sincronismo, um sinal válido em linha com os bytes, e sinal de erro indicando um pacote livre de erros.

Em difusão, o fluxo de dados pode conter pacotes nulos repetitivos, que podem fazer com que o Detector de Síndrome 220 dentro do bloco de enquadramento MPEG da tecnologia anterior trave na posição de byte errada (sincronização) produzindo assim saída de pacotes MPEG-2 inválidas para o bloco de transporte, mesmo quando o FEC for perfeitamente travado e entregar um fluxo de dados livre de erros.

SUMÁRIO DA INVENÇÃO

A presente invenção fornece um método e aparelho para detectar confiavelmente posições de byte de sincronismo recebidas por um sistema de transmissão digital no caso de um fluxo de pacotes contendo uma pluralidade de pacotes nulos ou uma pluralidade de pacotes contendo um padrão de dados repetitivos fixo e para sincronizar e entregar confiavelmente o fluxo MPEG-2 difundido à camada de transporte do

receptor. Um circuito do Detector de Pacote Nulos pode ser
32
provido para comparar o conteúdo do pacote atual com um pa-
drão de bits fixo (ou pré-determinado) para detectar um paco-
te nulo MPEG-2 e para identificar confiavelmente o local de
5 um byte de sincronismo codificado pela soma de verificação
dentro de um pacote nulo. Uma posição do byte de sincronismo
é identificada com base na posição do padrão de bits fixo
pré-determinado na parte do cabeçalho de um ou de uma plura-
lidade de pacotes nulos no fluxo.

10 DESCRIÇÃO RESUMIDA DOS DESENHOS

Os recursos apresentados da presente invenção fi-
carão mais aparentes pela descrição de suas modalidades e-
xemplares com detalhes com referência aos desenhos anexos,
em que:

15 A Figura 1 mostra um diagrama de blocos de um blo-
co de enquadramento MPEG da tecnologia anterior do terminal
receptor de um sistema de transmissão digital;

A Figura 2 é um diagrama de blocos de um bloco de
enquadramento MPEG no terminal receptor de um sistema de
20 transmissão digital de acordo com uma modalidade da presente
invenção;

A Figura 3 é um fluxograma que descreve o método
realizado para determinar a posição do byte de sincronização
no caso de pacotes nulos de acordo com uma modalidade da
25 presente invenção; e

A Figura 4 é um fluxograma que descreve mais pre-
cisamente um algoritmo usado para detectar um fluxo de dados
contendo pacotes nulos de acordo com uma outra modalidade da

presente invenção.

DESCRIÇÃO DAS MODALIDADES PREFERIDAS

A Figura 2 é um diagrama de blocos de um bloco de enquadramento MPEG 299 no terminal receptor de um sistema de transmissão digital de acordo com uma modalidade da presente invenção. O bloco de enquadramento MPEG 299 da presente invenção é similar ao bloco de enquadramento MPEG 200 da tecnologia anterior representado na Figura 1, mas pode compreender três blocos adicionais: o detector de pacotes nulos 250, a máquina de estado 255 e um bloco lógico de decisões 260.

O detector de pacotes nulos 250 detecta pacotes nulos recebidos no Fluxo de Dados Serial.

Pacotes nulos normalmente têm um preâmbulo pré-determinado seguido por um padrão de dados fixo (por exemplo, zeros). Em um pacote nulo, o preâmbulo contém bits específicos dentro do cabeçalho do pacote que são exclusivos de um pacote nulo. Esses são:

- a) carga útil_unit_start_indicator="0".
- b) PID=0x1FFF
- c) controle de cruzamento de sinais de transporte-
"00"
- d) campo de adaptação-'01'

Em alguns casos, pacotes nulos podem também conter o preâmbulo predeterminado seguido por dados de carga útil contendo um padrão de dados predeterminado (todos zeros, ou uma seqüência de bits sem ser todos zeros).

O detector de pacotes nulos 250 primeiramente de-



34
tecta o preâmbulo de pacotes Nulos predeterminado (por exemplo, PID igual a 0x1FFFhex (8191 dec)) e, mediante detecção do preâmbulo do pacotes Nulo predeterminado, começa determinar se os bits de dados subseqüentes é igual ao padrão de dados fixos esperado (predeterminado) (por exemplo, uma série de zeros). Se o preâmbulo do pacote Nulo predeterminado for detectado e o padrão de dados predeterminado (fixo) subseqüente for detectado (por exemplo, igual a zero), então considera-se que um pacote nulo foi encontrado ("detectado").

Nas modalidades alternativas (por exemplo, para aumentar a robustez da detecção de pacotes nulos contra ruído e interferência), o número de bits de diferença (erros de bits) entre o padrão de dados fixo esperado (predeterminado) e os dados da carga útil recebidos são contados, e a contagem diferença de bit (erro de bits) é comparada com um valor de patamar programável, Null_thresh. Se o preâmbulo for detectado, mas o número de bits não casados (erros de bits contados) for maior do que Null_thresh, então o pacote não é considerado um pacote nulo; caso contrário (se o preâmbulo for detectado), o pacote é considerado um pacote nulo. Mediante detecção do preâmbulo do pacote Nulo e subseqüentemente a contagem de diversos erros de bits (por exemplo, aqueles onde o padrão de dados predeterminado é "todos zeros") que são menores ou iguais a Null_thresh, o detector de pacotes nulos irá gerar um "sinal detectar nulo" (Null_flag). O Null-flag pode ser definido, por exemplo para ser um "1" lógico, quando um pacote nulo for detectado e, caso contrário,

co. Além do mais, o detector de pacote nulo 250 cria um sinal Null-sync que indica a posição do byte de sincronismo MPEG dentro do pacote (nulo) (no fluxo de pacotes) indicado pela detecção de um preâmbulo de pacote nulo e padrão de dados predeterminado.

Para aumentar a confiabilidade de detecção, a função do detector de pacotes nulos pode ser moderada pela função de filtragem passa baixa de uma característica histerética. Uma máquina de estado 255 pode ser provida após o detector de pacotes nulos 250 para permitir que o circuito de detecção tenha uma característica histerética que filtra flutuação de frequência mais alta na seleção da posição do byte de sincronismo com base na saída do detector de nulo 250. Este recurso pode ser implementado por contagem durante um número predeterminado ou programável de pacotes, Npackets1, se o Null_flag for 1 para uma contagem maior ou igual a um número designado (ou programável) de pacotes limitantes, Lock_In_thresh, antes de declarar uma trava do detector de pacotes nulos (mediante o que Null_lock é estabelecido em 1). Por outro lado, uma vez que Null_lock é 1, o circuito conta as ocorrências de Null_lock igual a 0 durante um número programável de pacotes, Npackets (por exemplo, Npackets0 = Npackets1 = Npackets) para determinar que o Null_flag foi 0 para uma contagem maior ou igual a um número designado (ou programável) de pacotes limitantes, Lock_Out_thresh, antes de declarar uma perda de trava (mediante o que Null_lock é estabelecido em 0). Em algumas modalidades, o número programável de pacotes, Npacket0 e Npackets1, (representando as "ja-

nelas" de busca que cobrem um número, por exemplo, N_{packets} de pacotes recebidos) pode ser igual ao respectivo número designado de pacotes limitantes, (Lock_Out_thresh e Lock_In_thresh), mas preferivelmente será maior, de maneira
5 tal que Null_lock seja estabelecido em 1 (ou restabelecido em 0), quando uma parte selecionável (por exemplo, três quartos) dos pacotes recebidos em uma janela de busca forem pacotes nulos (ou não forem pacotes nulos).

Outras variações da característica histerética são
10 possíveis sem perda de generalidade e, alternativamente, o detector de pacotes nulos 250 e a função da característica histerética (da máquina de estado 255) poderiam ser implementados por um microprocessador ou ASIC.

O bloco lógico de decisões 260 opera da seguinte
15 maneira: quando o detector de pacotes nulos 250 não detectar um pacote nulo (por exemplo, $\text{Null_sync} = 0$ e $\text{Null_lock} = 0$), a saída de sincronização MPEG (Sincronismo) é gerada pela soma de verificação com base no detector de sincronização (blocos gerador de síndrome 210 e detector 220) como na tec-
20 nologia anterior (isto é, a posição do byte de sincronismo está indicada pelo sinal Sync_flag determinado pelo detector baseado na soma de verificação 220). Mas, quando pacotes nulos forem detectados (e o Null_lock for 1), a sincronização MPEG é determinada pelo detector de pacotes nulos 250, e sua
25 saída Null_sync será usada como a saída de sincronização MPEG (Sincronismo) para ser usada no bloco de reinserção de Sincronismo MPEG 240. A saída de erro do bloco lógico de decisões 260 é a mesma que o Error_flag recebido do Detector

de Síndrome 220. A saída de Trava da lógica de decisão é o 37
Null_lock, se Null_lock for 1, e o Lock_flag, se Null_lock
for 0. O bloco lógico de decisões 260 pode ser implementado
como um circuito que efetivamente inclui um multiplexor que
5 multiplexa o Sync_flag convencional e a saída do sinal
Null_sync do Detector de Nulo 250, selecionado um ou o ou-
tro, de acordo com o método de seleção do bloco lógico de
decisões 260, da maneira explicada antes. Em modalidades al-
ternativas da invenção, a função da Máquina de Estado 255
10 (por exemplo, filtro de característica Histerética) pode ser
fundida e incorporada no bloco lógico de decisões 260.

As demais partes operantes do bloco de enquadra-
mento MPEG 299 em geral operam da mesma maneira que na tec-
nologia anterior da Figura 1.

15 Em algumas modalidades desta invenção, o detector
de pacotes nulos 250 do aparelho de enquadramento MPEG (por
exemplo, 299) pode ser compreendido de um detector adaptado
para detectar o preâmbulo predeterminado associado com um
pacote nulo (ou, mais simplesmente, adaptado para detectar
20 somente o campo PID igual a 0x1FFFhex (8191 dec)). Em outras
modalidades desta invenção, o detector de pacotes nulos 250
pode primeiramente detectar o preâmbulo e em seguida prosse-
guir para verificar se os bits de dados subseqüentes (na
carga útil dos pacotes) correspondem a um padrão de dados
25 fixo (por exemplo, uma série de zeros), ou todos os zeros
associados com um pacote nulo. Em algumas modalidades da in-
venção, a saída (Null_flag) do detector de pacotes nulos 250
é filtrada (por exemplo, por uma função histerética executa-

P I N T I B A I

da por uma máquina de estado 255) para produzir uma saída filtrada (Null_lock). Em outras modalidades da invenção, a saída do detector de pacotes nulos 250 não é filtrada (ver Figura 3).

5 Operações de um bloco de enquadramento MPEG (por exemplo, 299 da Figura 2) de acordo com modalidades da invenção são descritas adicionalmente com referência às Figuras 3 e 4.

10 A Figura 3 é um fluxograma que descreve o método geral realizado para selecionar a posição do byte de sincronização de acordo com uma modalidade da presente invenção. Na etapa S1, um Fluxo de Dados Serial compreendido de pacotes de comprimento fixo (por exemplo, byte 188) é recebido como sinais codificados digitalmente (MPEG-2) conforme um
15 protocolo predeterminado (por exemplo, conforme os padrões ANSI/SCTE 07 2000 e ITU-T J.83B), o protocolo definindo múltiplos tipos de pacotes (por exemplo, pacotes nulos), cada pacote incluindo uma parte de cabeçalho (que pode incluir um preâmbulo que identifica um pacote como um pacote nulo) e
20 uma parte de dados de carga útil, a parte do cabeçalho contendo um byte de sincronismo codificado pela soma de verificação.

Na etapa S2, o Fluxo de Dados Serial recebido é transmitido pelo gerador de síndrome, que é especificado no
25 padrão ANSI/SCTE 07 2000 (ver, por exemplo, o gerador de síndrome ilustrado na Figura 3 desse padrão) e em seguida um detector de síndrome 220 compara a saída do gerador de síndrome com a soma de verificação de sincronização predetermi-

nada para estabelecer se um byte de sincronismo (possível) foi detectado.

Se um byte de sincronismo for detectado pelo método da soma de verificação convencional realizado na etapa S2, então a etapa S3, e qualquer uma das etapas S4A ou Etapa S4B e, em seguida, as etapas S5 e S6, são executadas.

Na Etapa S3, é determinado se o pacote atualmente recebido é um pacote nulo. Se for determinado que o pacote atualmente recebido é um pacote nulo, então um sinalizador nulo (Null_flag) é estabelecido (por exemplo, em 1), e a etapa S4A e etapas S5 e S6 serão executadas; se for determinado que o pacote atualmente recebido é um pacote nulo, então o sinalizador nulo (Null_flag) é restabelecido (por exemplo, em 0) e a etapa S4B e as etapas S5 e S6 serão executadas.

Em seguida, na etapa S5, o valor do byte de sincronismo (47hex) é reinserido na posição do byte de sincronismo selecionada por uma das etapas S4A e S4B e, na etapa S6, o fluxo de dados contendo o byte de sincronismo da maneira restaurada antes sai para a camada de transporte como um fluxo de transporte MPEG-2 do padrão restaurado.

A Figura 4 é um fluxograma que descreve mais precisamente um algoritmo usado para detectar se o fluxo de dados contém múltiplos pacotes nulos, de acordo com uma outra modalidade da presente invenção.

A etapa S3B de detecção de pacotes nulos é uma alternativa a uma etapa S3 de detecção de pacotes nulos mais simples da Figura 3, e pode ser realizada pelo Detector de

P I D I S I S I

Nulo 250 e Máquina de Estado 255 da Figura 2, e é compreendido de subetapas B1, B2 e B3. Se um byte de sincronismo for detectado pelo método da soma de verificação convencional (executado na etapa S2 da Figura 3), então a etapa S3B é executada (e, em seguida, qualquer uma das etapas S4A ou etapa S4B e, em seguida, as etapas S5 e S6 da Figura 3, são executadas).

Na subetapa B1, é determinado se o pacote atualmente recebido contém o preâmbulo predeterminado (por exemplo, incluindo um PID igual a 0x1FFFhex (8191 dec)) associado com um pacote nulo (por exemplo, na posição do cabeçalho (PID) indicada pela sincronização baseada na soma de verificação). Se o preâmbulo do pacote Nulo predeterminado for detectado (na posição esperada) então um sinalizador nulo (Null_flag) pode ser tentativamente estabelecido (por exemplo, em 1) sujeito à determinação feita na subetapa B2, e a subetapa B2 é em seguida executada. Se o preâmbulo do pacote Nulo predeterminado (um primeiro padrão de bits fixo) não for detectado (na posição esperada) então um sinalizador nulo (*Null_flag) é restabelecido (por exemplo, em 0) e a determinação (Null_flag) é transmitida à subetapa de Característica Histerética B3.

Na subetapa B2, é determinado se a carga útil de dados (subseqüente) do pacote atualmente recebido contém o padrão de dados fixo predeterminado (um segundo padrão de bits fixo, por exemplo, uma série de zeros, ou todos os zeros). A diferença de bits (contagem de bit de erro) entre os dados recebidos e o padrão de dados predeterminado (fixo) é

contado e comparado com um limite programável, `Null_thresh`. Se o padrão de dados predeterminado (fixo) for detectado (por exemplo, bits de dados igual a zero ou contagem de bits de erro for menor ou igual a `Null_thresh`), então considera-se que um pacote nulo é "encontrado" (detectado) e o sinalizador nulo (`Null_flag`) é estabelecido (por exemplo, em 1) e a subetapa B3 é em seguida executada.

Se o padrão de dados fixo predeterminado (por exemplo, uma série de zeros, ou todos os zeros) não for detectado (dentro do limite `Null_thresh`), então um sinalizador nulo (`Null_flag`) é restabelecido (por exemplo, em 0), e essa determinação (`Null_flag`) é transmitida à Subetapa de Característica Histerética B3. Assim, se o padrão de dados predeterminado (fixo) não for detectado (por exemplo, para muitos bits de erro serem contados), então considera-se que o pacote nulo não é "encontrado" (detectado), e o sinalizador nulo (`Null_flag`) é restabelecido (por exemplo, em 0) e a subetapa B3 é em seguida executada.

Na subetapa B3, o sinal `Null_flag` resultante tanto da subetapa B1 como B2 é filtrado (por exemplo, por uma característica histerética). Esta filtração pode ser implementada por uma máquina de estado (por exemplo, 255 na Figura 2) para reduzir rápidas flutuações na sinalização indicando uma detecção de um pacote nulo (por exemplo, pelo Detector de Nulo 250 da Figura 2) que é usado para identificar a verdadeira posição dos bytes de sincronismo MPEG-2 nos pacotes recebidos pelo Fluxo de Dados Serial. Os detalhes da realização de um filtro de característica histerética como esse

podem ser os mesmos daqueles anteriormente notados com relação à descrição apresentada da operação da Máquina de Estado 255 da Figura 2, em que o estado do pacote nulo determinado de um número (por exemplo, Npackets é maior do que um) de 5 pacotes são examinados para determinar a saída de decisão do estado da máquina (para selecionar o desempenho tanto da etapa S4A como S4B da Figura 3). O resultado da subetapa B3 de filtração é então usado para selecionar o desempenho tanto da etapa S4A como da etapa S4B da Figura 3, e em seguida 10 as etapas S5 e S6 serão executadas.

Modalidades exemplares da invenção foram explicadas anteriormente e estão mostradas nas Figuras. Entretanto, a presente invenção não está limitada às modalidades exemplares supradescritas, e fica aparente que variações e modificações podem ser realizadas pelos versados na técnica de 15 acordo com o espírito e escopo da presente invenção. Portanto, as modalidades exemplares não devem ser entendidas como limitações, mas exemplos. O escopo da presente invenção não é determinado pela descrição apresentada, mas pelas reivindicações anexas, e variações e modificações podem ser feitas 20 nas modalidades da invenção sem fugir do escopo da invenção na forma definida pelas reivindicações anexas e equivalentes.

42

REIVINDICAÇÕES

1. Aparelho (299) para processar um fluxo de pacotes de comprimento fixo recebido como sinais codificados digitalmente e tendo múltiplos tipos de pacotes, cada pacote incluindo uma parte de cabeçalho e uma parte de dados, a parte do cabeçalho contendo um byte de sincronismo codificado pela soma de verificação, o aparelho sendo **CARACTERIZADO** pelo fato de que compreende:

um Detector de Síndrome (220) para detectar o byte de sincronismo codificado pela soma de verificação e para gerar um sinal Sync_flag para indicar o local do byte de sincronismo codificado pela soma de verificação;

um Detector de Pacotes Nulos (250) adaptado para detectar se um pacote recebido é um pacote nulo, e adaptado para identificar o local do byte de sincronismo de um pacote nulo detectado; e

um circuito de reinserção de byte de sincronismo MPEG (240) para inserir um valor predeterminado no local do byte de sincronismo indicado por um sinal de sincronização MPEG.

2. Aparelho, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que o Detector de Pacotes Nulos (250) é adicionalmente adaptado para sair com um sinal Null_sync para indicar o local do byte de sincronismo de um pacote nulo detectado.

3. Aparelho, de acordo com a reivindicação 2, **CARACTERIZADO** pelo fato de que compreende adicionalmente:

um multiplexador, em que a saída Sync_flag do De-

tector de Síndrome e a saída Null_sync do Detector de Pacotes Nulos são multiplexadas e alternativamente têm saída pelo multiplexador, para ser usadas pelo circuito de reinserção do byte de sincronismo MPEG, de acordo com a condição de pacotes nulos terem sido detectados.

4. Aparelho, de acordo com a reivindicação 3, **CARACTERIZADO** pelo fato de que compreende adicionalmente um circuito lógico de decisões (260) conectado operativamente ao multiplexador e adaptado para controlar o multiplexador de maneira tal que, quando o Detector de Pacotes Nulos detectar pacotes nulos, a saída Null_sync do Detector de Pacotes Nulos tem saída pelo multiplexador para ser usada como o sinal de sincronização MPEG pelo circuito de reinserção do byte de sincronismo MPEG.

5. Aparelho, de acordo com a reivindicação 2, **CARACTERIZADO** pelo fato de que é adaptado de maneira tal que, quando pacotes nulos forem detectados, a saída Null_sync do detector de Pacotes Nulos é usada como sinal de sincronização MPEG usado pelo circuito de reinserção de sincronismo MPEG.

6. Aparelho, de acordo com a reivindicação 5, **CARACTERIZADO** pelo fato de que, quando pacotes nulos não forem detectados, a saída Null_Sync do detector de Pacotes Nulos não é usada como o sinal de sincronização MPEG usado pelo circuito de reinserção de sincronismo MPEG.

7. Aparelho, de acordo com a reivindicação 6, **CARACTERIZADO** pelo fato de que, quando pacotes nulos não forem detectados, a saída Sync_flag do Detector de Síndrome é

usada como o sinal de sincronização MPEG usado pelo circuito de resincronização de sincronismo MPEG.

8. Método para processar um fluxo de pacotes de comprimento fixo recebido como sinais codificados digitalmente e tendo múltiplos tipos de pacotes, cada pacote incluindo uma parte de cabeçalho e uma parte de dados, a parte de cabeçalho contendo um byte de sincronismo codificado pela soma de verificação, o método sendo **CARACTERIZADO** pelo fato de que compreende:

10 realizar uma primeira etapa de detecção de decodificação da soma de verificação para detectar uma posição candidata do byte de sincronismo codificado pela soma de verificação no pacote atual dos pacotes de comprimento fixo; e

15 realizar uma segunda etapa de detecção para detectar o primeiro padrão de bits fixo na parte do cabeçalho no pacote atual dos pacotes de comprimento fixo;

20 se o primeiro padrão de bits fixo for detectado no fluxo de pacotes de comprimento fixo, então identificar a posição do byte de sincronismo no byte de sincronismo de cada um dos pacotes de comprimento fixo baseado na detecção do primeiro padrão de bits fixo;

 inserir um valor de byte de sincronismo predeterminado na posição do byte de sincronismo identificado.

25 9. Método, de acordo com a reivindicação 8, **CARACTERIZADO** pelo fato de que a segunda etapa de detecção é realizada somente se for detectado uma posição do byte de sincronismo candidato codificado pela soma de verificação na primeira etapa de detecção.

10. Método, de acordo com a reivindicação 8, **CARACTERIZADO** pelo fato de que, enquanto o primeiro padrão de bits fixo não for detectado no fluxo de pacotes de comprimento fixo, então identificar a posição do byte de sincronismo do byte de sincronismo de cada um dos pacotes de comprimento fixo no fluxo baseado no resultado da primeira etapa de detecção;

e inserir o valor do byte de sincronismo predeterminado na posição do byte de sincronismo candidato baseado no resultado da primeira etapa de detecção.

11. Método, de acordo com a reivindicação 8, **CARACTERIZADO** pelo fato de que, se um byte de sincronismo candidato codificado pela soma de verificação for detectado na primeira etapa de detecção, mas o primeiro padrão de bits fixo não for detectado na segunda etapa de detecção, então inserir um valor de byte de sincronismo predeterminado na posição de byte de sincronismo candidato com base no resultado da primeira etapa de detecção.

12. Método, de acordo com a reivindicação 8, **CARACTERIZADO** pelo fato de que a segunda etapa de detecção é realizada mesmo se um byte de sincronismo candidato codificado pela soma de verificação não for detectado na primeira etapa de detecção.

13. Método, de acordo com a reivindicação 8, **CARACTERIZADO** pelo fato de que o primeiro padrão de bit fixo é um padrão de bits predeterminado na parte do cabeçalho de cada pacote dentro da pluralidade de pacotes.

14. Método, de acordo com a reivindicação 13,

CARACTERIZADO pelo fato de que as ditas partes do cabeçalho compreendem cabeçalhos de transporte de um Fluxo de Transporte MPEG-2.

15. Método, de acordo com a reivindicação 8, **CARACTERIZADO** pelo fato de que o primeiro padrão de bits fixo é um padrão predeterminado que inclui pelo menos um dos seguintes valores de campo do cabeçalho de ligação de fluxo de transporte MPEG-2: `payload_unit_start_indicator = "0"`, `PID=0x1FFF`, controle de cruzamento de sinais de transporte = `"00"` e campo de adaptação = `'01'`.

16. Método, de acordo com a reivindicação 8, **CARACTERIZADO** pelo fato de que cada um da pluralidade de pacotes é um pacote nulo MPEG-2.

17. Método, de acordo com a reivindicação 8, **CARACTERIZADO** pelo fato de que cada pacote na pluralidade de pacotes que contém cada qual um primeiro padrão de bits fixo em uma parte de cabeçalho de cada pacote contém adicionalmente um segundo padrão de bits fixo dentro da parte de dados de cada pacote; e compreende adicionalmente:

20 realizar uma terceira etapa de detecção para detectar o segundo padrão de bits fixo na pluralidade de pacotes dentro do fluxo de pacotes de comprimento fixo, e, se o segundo padrão de bits fixo for detectado no fluxo de pacotes de comprimento fixo, então inserir o valor do byte de sincronismo predeterminado na posição do byte de sincronismo com base no resultado da terceira etapa de detecção.

18. Método, de acordo com a reivindicação 16, **CARACTERIZADO** pelo fato de que a terceira etapa de detecção

é realizada somente se a posição do byte de sincronismo candidato codificado pela soma de verificação for detectada na primeira etapa de detecção.

19. Método, de acordo com a reivindicação 16,
5 **CARACTERIZADO** pelo fato de que, se nem a primeira nem a segunda etapa de detecção tiver identificado uma posição do byte de sincronismo, então nenhum valor do byte de sincronismo predeterminado é inserido no fluxo de pacotes de comprimento fixo.

10 20. Método, de acordo com a reivindicação 16,
CARACTERIZADO pelo fato de que a terceira etapa de detecção é realizada somente se o primeiro padrão de bits fixo for detectado no fluxo de pacotes de comprimento fixo na segunda etapa de detecção.

15 21. Método, de acordo com a reivindicação 16,
CARACTERIZADO pelo fato de que, se o segundo padrão de bits fixo não for detectado no fluxo de pacotes de comprimento fixo, então identificar a posição do byte de sincronismo do byte de sincronismo de cada um dos pacotes de comprimento
20 fixo com base no resultado da primeira etapa de detecção.

200 →

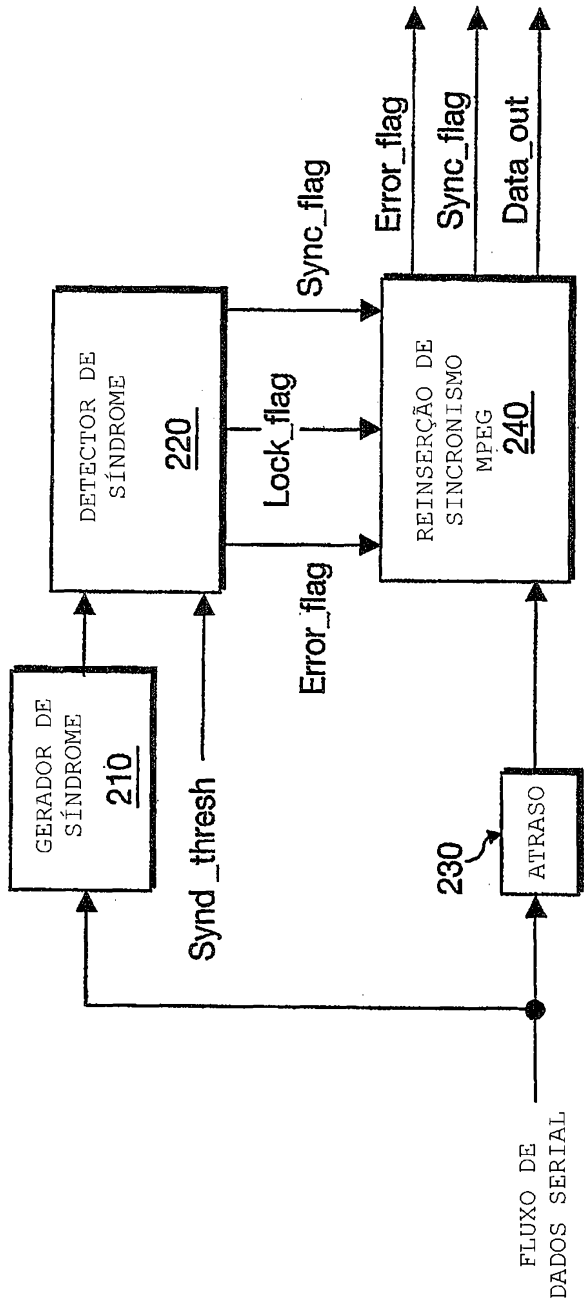
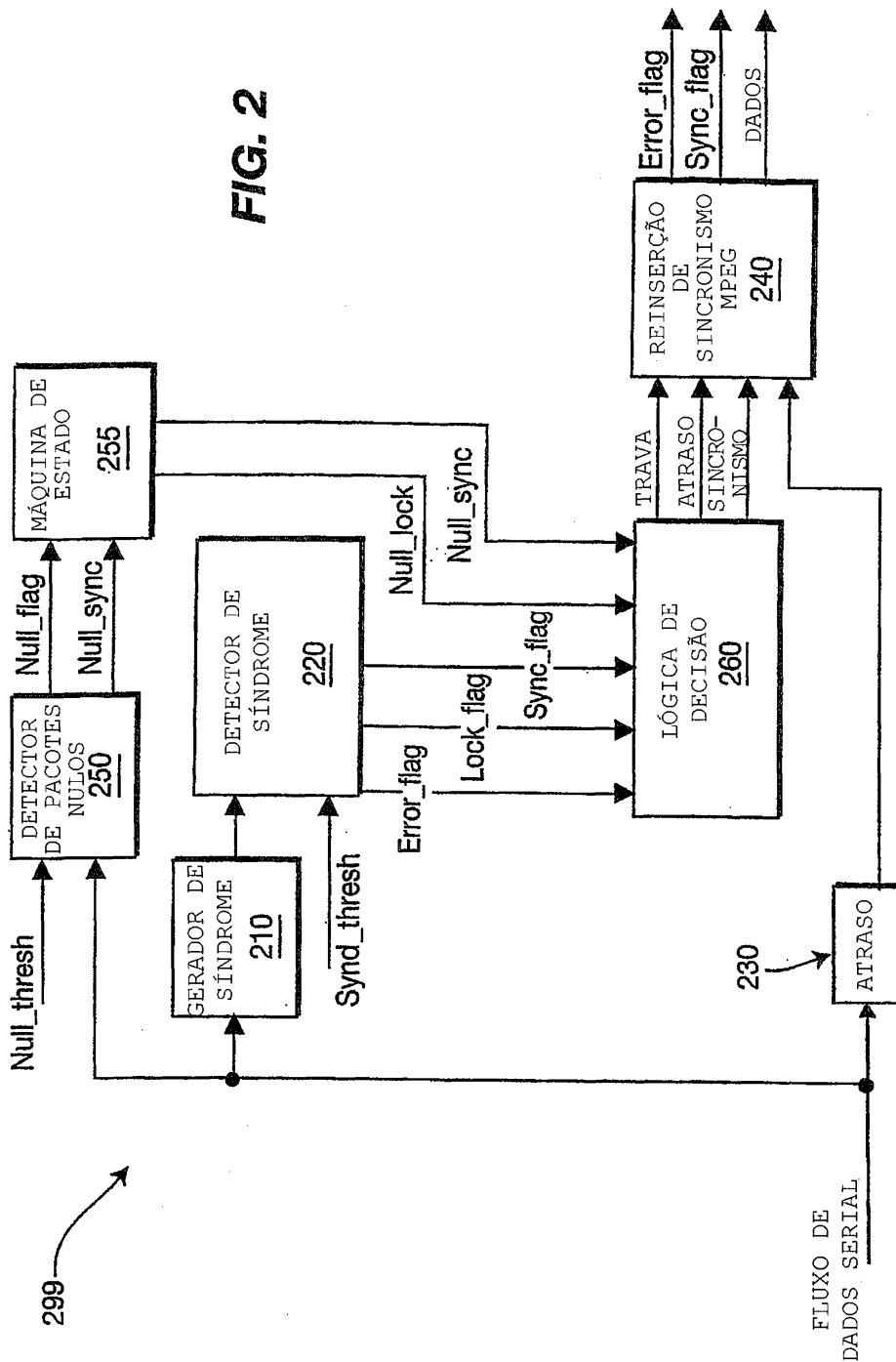
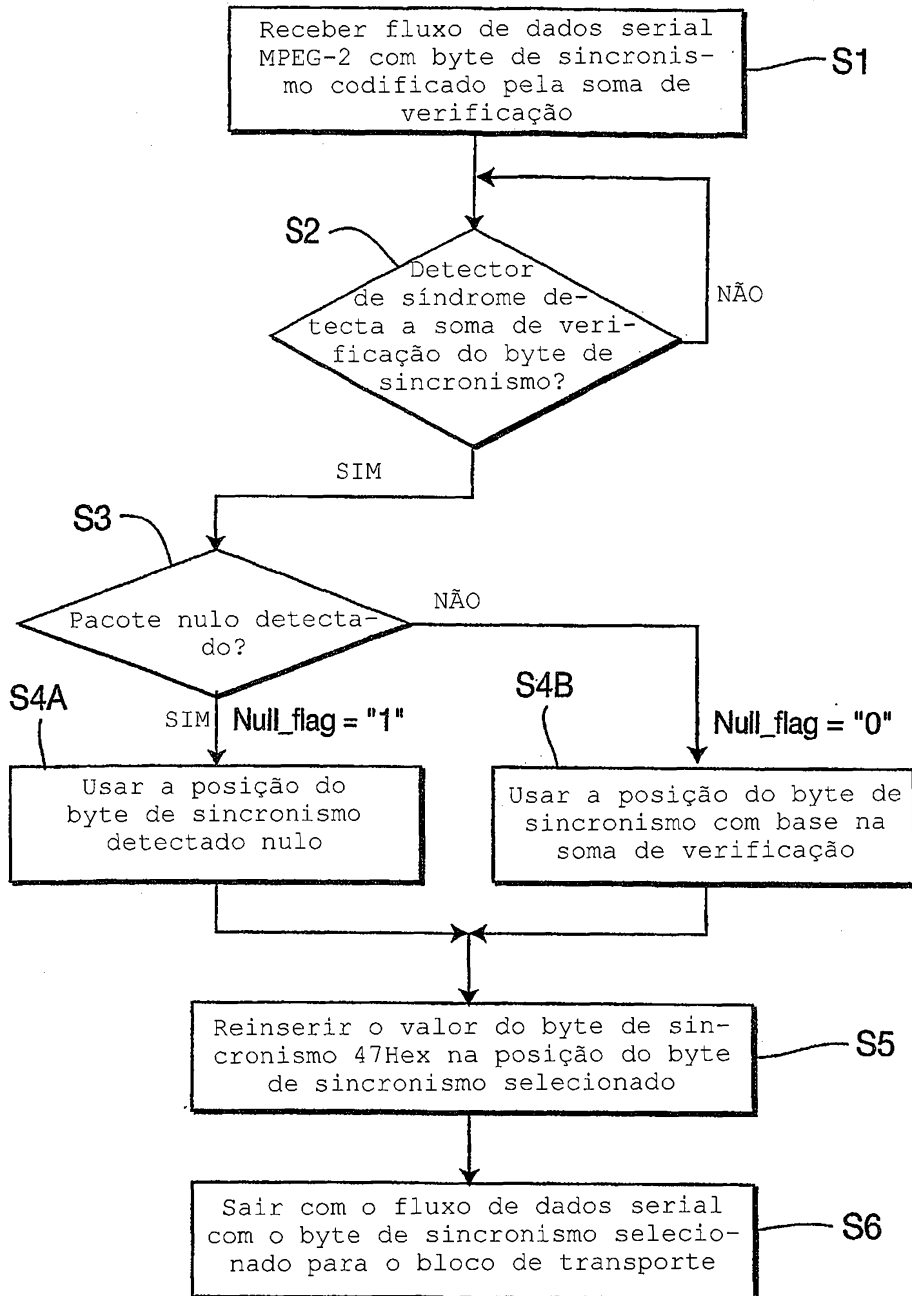


FIG. 1

FIG. 2



**FIG. 3**

56

FIG. 4

