

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4049863号
(P4049863)

(45) 発行日 平成20年2月20日(2008.2.20)

(24) 登録日 平成19年12月7日(2007.12.7)

(51) Int.Cl.

H04N 7/26 (2006.01)

F 1

H04N 7/13

Z

請求項の数 4 (全 18 頁)

(21) 出願番号 特願平9-339253
 (22) 出願日 平成9年11月4日(1997.11.4)
 (65) 公開番号 特開平10-210461
 (43) 公開日 平成10年8月7日(1998.8.7)
 審査請求日 平成16年11月4日(2004.11.4)
 (31) 優先権主張番号 030108
 (32) 優先日 平成8年11月1日(1996.11.1)
 (33) 優先権主張国 米国(US)

(73) 特許権者 590000879
 テキサス インスツルメンツ インコーポ
 レイテッド
 アメリカ合衆国テキサス州ダラス、ノース
 セントラルエクスプレスウェイ 135
 OO
 (74) 代理人 100066692
 弁理士 浅村 皓
 (74) 代理人 100072040
 弁理士 浅村 肇
 (74) 代理人 100091339
 弁理士 清水 邦明
 (74) 代理人 100094673
 弁理士 林 銘三

最終頁に続く

(54) 【発明の名称】MPEG2トランSPORT・ストリーム・パケット・パーザ・システム

(57) 【特許請求の範囲】

【請求項1】

パケットタイプを識別するパケット・アイデンティティ(PID)を含むパケットのヘッダデータからデータ部分を抽出し、データ入力ストリームから更なるデータ(データ1)を抽出するシフトレジスタと、

シフトレジスタに接続され、前記データ部分によってアドレス指定される連想メモリであって、該連想メモリは、前記データ部分に対応するパケットタイプがパケットタイプの所定の部分集合と一致するかどうかを示す一致データ(VALID-PID)を格納し、前記一致データが一致を示すなら、対応するパケットタイプに対する基準データ(CAM-AD)をさらに格納し、前記データ部分によってアドレス指定された時、もし前記一致データが一致を示すなら、前記一致データと前記基準データを出力するために前記データ部分を出力する、連想メモリと、

前記連想メモリと接続され、該連想メモリが基準データを出力するなら、前記基準データに応答して、前記基準データに対応するアドレス(SRAM-AD)を発生するアドレス・ゼネレータと、

前記アドレス・ゼネレータと接続され、前記基準データの各々に対する属性データ(データ2)を格納するメモリ(SRAM)と、

前記メモリと前記シフトレジスタとに接続され、前記一致した基準データに関連して選択された属性データ(データ2)と前記シフトレジスタにより供給される更なるデータ(データ1)に応答して前記パケットを処理する方法を決定する論理回路

とを含むパケット認識処理回路。

【請求項 2】

前記シフトレジスタから前記論理回路へのデータパス（データ 1，パケット・アナライザ制御レジスタ）を含み，前記基準データに応答して前記更なるデータを前記論理回路に供給する請求項 1 に記載のパケット認識処理回路。

【請求項 3】

前記データパスが，シフトレジスタに接続されたデータ入力と前記連想メモリに接続された負荷イネーブル入力（A F - ロード）とを有する更なるレジスタ（パケット・アナライザ制御レジスタ）を含み，前記基準データおよび負荷イネーブル信号に応答して前記更なるデータ（データ 1）をロードし，前記更なるレジスタは前記論理回路に接続されたデータ出力を有する，請求項 2 に記載のパケット認識処理回路。

10

【請求項 4】

前記更なるレジスタは前記メモリと接続された更なるデータ入力（データ 2）を有し，前記選択された属性データを受け取る，請求項 3 に記載のパケット認識処理回路。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、M P E G 2 デコーダに関し、特に、デコーダなどの中で使うための I M P E G 2 トランスポート・ストリーム・パケット・パーザ（p a r s e r）に関する。

20

【0002】

【従来の技術、及び、発明が解決しようとする課題】

ディジタル T V デコーダは次の 5 つのモジュールに分割することができる。

復調

M P E G トランスポート・パケット・パーザ（T P P）

M P E G オーディオ

M P E G ビデオ

R I S C マイクロコントローラ

M P E G 2 トランスポート・ストリーム・パーザ（T P P）はM P E G のトランスポート・ストリームを受信し、ビデオ、オーディオまたはサービス情報のパケットを選択する。

30

デコードの後、そのパケットはメモリ・バッファに格納されてデータ・ストリームを形成する。オーディオ・デコーダはM P E G のオーディオ・ストリームを処理し、アナログのオーディオ信号を生成する。ビデオ・デコーダはM P E G のビデオを復元（d e c o m p r e s s）し、ビデオ・シーケンスを発生する。

【0003】

<トランスポート・パケットのフォーマット>

トランスポート・パケット（図 1）は 188 個のバイトを含む。このパケットは 2 つの部分、すなわち、32 ビットのヘッダおよび 184 バイトのペイロード（p a y l o a d）に分けられる。トランスポート・ヘッダは異なるフィールド（P I D、ペイロード・ユニット・スタート・インジケータ、アダプテーション・フィールド・フラグ、連続性カウンタのインデックス）を含み、それによってトランスポート・パケットのパーザが粗いフィルタリングを行う。トランスポート・ヘッダの後、パケットは可変長のアダプテーション・フィールドまたはペイロードを含む可能性がある。これらのペイロードのいくつかはそれ自身ヘッダ（P E S ヘッダ）から始まり、それも可変長である（図 1 参照）。M P E G 2 の標準規格における P E S ヘッダは 1 つのパケットより大きくなる可能性がある（次のパケットの上にオーバフローする）。

40

【0004】

現代水準のM P E G 2 トランスポート・ストリーム・システム・アナライザは完全なトランスポート・ストリーム・ハードウェア・アナライザおよびマイクロプロセッサを含んでいるチップ・セット・ソリューションから構成される。ビデオおよびオーディオのデコーダは、図 2 に示されているように別のデバイスの中にある。トランスポート・ストリーム

50

・パーザは現在の方式によってトランスポート・ストリームのフル解析（約70kゲート）を担当し、CPUに対してはデータを正しい方向に向けるタスクおよび異なるモジュールを制御するタスクを残している。もう1つのソリューションはすべてのトランスポート解析をソフトウェアで実装することである。これはずっと高いCPUパワーの処理を必要とする。

【0005】

【課題を解決するための手段】

プリプロセッシングの後に関連するフラグが付いているパケットを含めるための中間バッファと、送信フラグによって選択されたパケットをさらにソフトウェアで処理するためのプロセッサとを含んでいる、トランスポート・ストリーム・パーザ・システム。

10

【0006】

【発明の実施の形態】

本発明のトランスポート・ストリーム・パーザは解析をより効果的にするために、トランスポート・ストリームおよびMPEG2の解析の負荷をハードウェアとソフトウェアとの間で分割する。ハードウェアのフィルタリングによってPIDの認識に基づいて早期の決定ができるので、データ・ストリームを非常に効果的に削減することができ、一方、ファームウェアはその削減されたデータ・ストリーム上で、より複雑なフィルタリングのアルゴリズムを実行する。この方式はリソースを最適化する。ハードウェアは単純になり、一方、ファームウェアはCPUパワーがより少なくて済む。というのは、受け取るビット・レートがハードウェアによってあらかじめ処理されているからである。

20

パケットのパーサリング (parsing) およびルーティングは、各PIDに対してローカル・メモリの中に格納されている一組の属性と結合されているPID認識機構によって行われる。これらの属性によって異なるヘッダから受信された情報が完成される。

【0007】

<トランスポート・パケット・パーザの機能>

このモジュールの機能はアプリケーションによって選択されないすべてのパケットを同時に捨てる所以であり、そして組み込みのソフトウェアからの「リアル・タイム」の支援なしに、できるだけ多くのパケットを回送することである。

有効なPIDを持つパケットが検出されると、パーザはそのパケットがCPUによってさらに処理される必要があるかどうかを決定する条件の集合をチェックする（「EOP割込み発生」の詳細説明参照）。これらの条件の1つが満足されていた場合、そのパケットは付加されたフラグと一緒に一時バッファへ送られ（さらに処理が必要である）、そしてEOP割込みがCPUへ送られて新しいパケットが受信されていることを示す。この一時バッファの中に記憶されたデータは、図3のバス（2）の中で示されているように回送されるか、あるいは捨てられるかが行われる前に、組込みソフトウェアによって処理される。データが処理されると、RISCはそれらを自分自身で転送するか、あるいはシステム・バスへのDMA転送を起動する。また、そのソフトウェアは或る場合においてPID属性（例えば、ストリーム-idフィルタ）のいくつかを変更し、それに続くパケット（同じPIDの）のルーティングを変更しようとする。どの条件も満足されなかった場合、付加されたフラグは追加の処理が不要であることを示す。そのパケットは自動のDMA転送を経由してその最終の宛先へ自動的に転送される（バス（1））。例えば、圧縮されたビデオ情報を含んでいるパケットは直接システム・ビデオ・バスへ送ることができる。この場合、EOP割込みがRISCへ送られることなく、パケットの最終宛先に対する転送はCPUのリソースを必要としない。

30

【0008】

図3は2つのデータ・フローを示している。

【0009】

このメカニズム（図3）はハードウェア/ソフトウェア解析の間のトレードオフのためのキーである。トランスポート・ストリーム・パーザは非常に容易にそのパケットを回送することができ、そして「トランスポート・ストリーム・ソフトウェア解析」の説明の中で

40

50

示されているように、そのビット・ストリームの或る部分について C P U により複雑な解析を行わせることができる。

その選択はヘッダ (P I D 、 A F フラグ、 P E S / P S I スタート・フラグ [ペイロード・ユニット・スタート・インジケータ] 、トランスポート・エラー・フラグ、カウンタの連続性インデックス) を解析することおよび、 P I D 属性を解析することによって行われる。というのは、その選択はパケット自身から情報を抽出することによって必ずしも常に行われるとは限らないからである (図 4) 。場合によっては、以前のパケット (同じ P I D の) の性質によって、そのパケットが追加の処理 (P E S ヘッダが 2 つのパケットの間で分離する) を必要とするかどうかが決定される。場合によっては、それは P I D 番号にリンクされる。 P I D が、例えば、常に C P U の処理を必要とするテーブル情報を含んでいる場合、その属性 S p e i がセットされる (この P I D を持つパケットは常に C P U へ送られることを意味する) 。 E O P 割込みを発生するために使われる情報のリストが以下の図 4 の中に要約されている。

この図は情報がどこから来るかを示している。 E O P を発生するための条件は後で詳細に説明される。

【 0 0 1 0 】

< パケット・アナライザ制御レジスタのビットの説明 >

V a l i d _ P I D (有効 P I D) : このビットは P I D 認識機能の結果を示す。その P I D 番号は各パケットのトランスポート・ストリーム・ヘッダの中で受信されたパケット識別番号である。それはパケットがシステムに対して転送されなければならないかどうかを決定する。 P I D 番号が選択された場合、この V a l i d _ P I D ビットはこの P I D 番号を持つパケットが受信された時にアクティブ「 1 」になる。それは第 1 レベルのパケット・フィルタリングであり、そして他の解析ステップに対する必要条件である。

【 0 0 1 1 】

T e (トランスポート・エラー・インジケータ) : これはトランスポート・ストリーム・パケット・ヘッダから検索される。トランスポート・エラー指示ビットは受信されたパケットが何らかのエラーを含んでいる時にアクティブになる。

P e s / P s i _ s t a r t フラグ (トランスポート・ストリーム・ペイロードのスタート・ユニット・インジケータ) : これはトランスポート・ストリーム・パケットのヘッダから検索される。

トランスポート・ストリームのペイロードが P E S パケット・データを含んでいる時。「 1 」はこのトランスポート・ストリーム・パケットのペイロードが P E S パケットの第 1 バイト (P E S ヘッダ) から始まることを示し、「 0 」はトランスポート・ストリーム・パケットの中で P E S パケットが開始しないことを示す。

【 0 0 1 2 】

トランスポート・ストリーム・パケットのペイロードが P S I データを含んでいる時。「 1 」はそのペイロードが少なくとも 1 つの P S I セクションの先頭を含んでいることを意味し、そして「 0 」はこのトランスポート・ストリーム・パケットの中で P S I セクションのスタートが含まれていないことを意味する。

A F _ f r a g (トランスポート・ストリーム・アダプテーション・フィールド) : トランスポート・ストリーム・ヘッダの中のこの 2 ビットのフィールドは、そのトランスポート・ストリーム・ヘッダの後にアダプテーション・フィールドが続いているかどうかを示す。

【 表 1 】

10

20

30

40

AFの値	説明
00	ISO/IECによって将来使うために予約されている
01	アダプテーション・フィールドなし、ペイロードのみ
10	アダプテーション・フィールドのみ、ペイロードなし
11	アダプテーション・フィールドの後にペイロードが続く

【0013】

10

CC-error (CCのエラー) : このビットは連続性カウンタ (CC) におけるハードウェアの不連続性カウンタ検出機能の結果である。その検出はCurrent_CC (現在のCC) と呼ばれるCC (4ビットがトランスポート・ストリーム・ヘッダの中に含まれている) と同じPID (Previous_CC (前のCC) と呼ばれる) の前のパケットからのCCとの間の比較によって行われる。Previous_CCはハードウェアによって変更された属性である。その比較が行われると、ハードウェアはそのPrevious_CCを次のパケットの受信のためにCurrent_CCで書き換える。以下同様に行われる。その比較が、Current_CC = Previous_CCまたはCurrent_CC = Previous_CC + 1の条件にマッチしなかった場合、CC_errorビットがアクティブになる。

20

【0014】

Duplicate-packet (パケット複製) フラグ : 比較がcurrent_CC = Previous_CCであることを示した場合、このDuplicate-packet フラグがアクティブになる。

Hpei (ハードウェアのパケット・エンド・イネーブル割込み)。これはハードウェアによってセットされる属性ビットである。新しいパケットの受信時 (パケット・ユニット・スタート・インジケータがアクティブである時およびそれがプログラム・サービス情報PSIを持つパケットでない時)、Hpeiがアクティブになり、同じPIDの次のパケットにおいてリセットされる。それはPESヘッダが2つのパケット間で分割される場合に、さらに解析されるようにPESヘッダを持つパケットに続くパケットを一時的にバッファの中に記憶させる。このビットは入力ビット・ストリームのビット・レートが高過ぎてEOPサービス・ルーチンのラテンシー (latency) の範囲内にフィットしない時に必要である。

30

【0015】

Spei (ソフトウェアのパケットの終りイネーブル割込み) : この属性ビットはソフトウェアによってセットされ、2つの異なる目的をカバーする。第1の使用では、このビットはCPU処理を常に必要とするPIDに対して初期化時にセットされる (例えば、テーブル情報を持つPID)。それはそのケースにおいてはこのPIDのすべてのパケットがCPUによって解析されるように強制するために単純に使われる。

第2の使用法はPESヘッダが2つのパケット間で分割されるオーディオおよびビデオのパケットに対する場合である。これはソフトウェアがそのビット・ストリーム・レートとそのラテンシー応答がコンパチブルであると判定した時にいくつかの与えられたシステムに対してだけ有効である。その場合、ソフトウェアはPESヘッダを含んでいるパケットのEOP割込みを受け取ると直ぐにSpeiをセットする。後でソフトウェアはその次のパケットにおいてPESヘッダがオーバフローするかどうかを判定する。オーバフローしない場合、ソフトウェアはSpei属性ビットをリセットする。この場合、Speiは効率を良くするためにディスエーブルされるHpeiを置き換えることができる (En_Hpei参照)。

40

【0016】

PSI (プログラム・サービス情報パケット・フラグ) : MPEG2のトランスポート・

50

ストリームにおいて、トランスポート・ストリーム・ヘッダの中の1つのフラグ（ペイロード・ユニット・スタート・インジケータ）が新しいPESパケットのスタートまたはプログラム・サービス情報のセクションのスタートを示す。これらの2つのタイプのデータを区別するために、別の属性（PSI）が必要である。この属性はプログラム・サービス情報（PSI）を含んでいるPIDに対する初期化時に、ファームウェアによってセットされる。

それはHpeiを発生するために使われるが、さらに重要なことは、暗号化されたバイト/クリーンなバイトの間の境界の検出のために使われる。PIDに密に結合されているこの情報なしでは、コントローラは暗号化フラグを見つけることができず、あるいはビット・ストリーム・インターフェース上で意味のあるいくつかの入力バッファなしでオン・タイムで、暗号化されたバイト/クリーンなバイトの間の境界を見つけることもできない。

Stream-id-filter（ストリームidフィルタ）；このソフトウェア属性ビットはソフトウェアが選択されていないストリームidを持つPESヘッダを含んでいるパケットを受信した時に、ソフトウェアによってセットされる。この属性がセットされると、ハードウェアは新しいPESヘッダに達するまでペイロードを含んでいるこのPID（AF = “1x”的すべてのパケットを捨てる。そのパケットはCPUへ送られ、CPUはPESヘッダを解析して、その新しいストリームidが有効であるかどうかを判定する。有効であった場合、CPUはそれに続くパケットに対するストリームidフィルタをリセットする。

【0017】

<グローバル制御ビットの説明>

En_Hpei : Hpeiのメカニズムはシステムの制約によってスイッチ・オフ/オンすることができる。次のパケットを受信する前にソフトウェアが反応してSpei属性ビットを変更するのに十分な時間があると判定した場合、Hpei属性をディスエーブルすることができる。その場合にはそのビット・ストリームのパーシングはより効率的である。というのは、PESヘッダが付いているパケットに続くパケットは、PESヘッダが1つのパケットの中に完全には含まれていない時にCPUにさらに解析させるためにCPUへ送られるだけだからである。

En_erroneous_p : このグローバル制御ビットはエラーのあるパケットを一時バッファへ転送することができる。これは将来において何らかの新しいエラー隠蔽（concealment）対策が発見される可能性がある場合において、或る程度に柔軟性を追加するのが主な目的である。

En_dupliccate_p : このグローバル制御ビットはペイロードだけを含んでいる複製パケットを送信できるようにする。これはテストの目的だけに使われる。

【0018】

<TPPハードウェアの実装>

次の図（5）はPIDの認識に基づいたハードウェアの実装を示す。トランスポート・パケットのヘッダからのPIDは連想メモリの中にある一組のPID（32）と並列に比較される。ヘッダの中のPIDが連想メモリの中のPIDに格納されているPIDに対応している場合、CAMエンコーダは単独のシステム・クロック期間において「マッチ」信号（Valid-PID）およびCAMアドレス（cam-ad）を出力する。valid-PIDはトランスポート・システム・パケットの解析を開始するために使われ、それによってプロセスがトランスポート・ストリーム・ヘッダから入ってくる他のビット、AF-load（トランスポート・エラー、ペイロード・ユニット・スタート・インジケータ、アダプテーション・フィールド・フラグ、連続性カウンタ・インデックスおよび他の同様な暗号化フラグが解読モジュールの制御のために予約されている）をロードすることができる。有効なPIDが検出されると直ぐに、cam-adがその関連する属性を探すために使われる。シーケンサがRead-SRAM属性を発生してSRAMのアドレス・ゼネレータを制御するステート・マシンをさせる。そのステート・マシンはcam-adからSRAMのアドレスを生成し、そのSRAMを二度読む。一度はハードウェア属性を検

10

20

30

40

50

探し、そして二度目はソフトウェア属性（スプリットしているハードウェアによって修正された属性およびソフトウェアによって修正された属性は複雑な保護メカニズムを使うことを避ける）を読む。両方から、EOP割込みを発生するための条件である属性およびビット・ストリーム情報（ヘッダからのビット）が評価される。ハードウェア属性はその後、次のパケット（同じPIDの）を受信するために更新される。ソフトウェア属性はEOP割込みサービス・ルーチンの一部として更新することができるが、それは或る非常に正確な機能（例えば、ストリームidフィルタ）のためだけである。次のパケットのスタートの前に属性が変更される必要がある場合、ビット・レート、パケット間のギャップおよび割込みサービス・ルーチンのラテンシーがその変更を許すためのいくつかの制約条件を満足していなければならない。

パケット・アナライザの制御ビット（図4）に依存して、パケットは捨てられるか、自動的にその最終の宛先へ送信されるか、あるいは組み込まれているソフトウェアによって事前に処理される。EOP割込みを発生するためのこれらの条件の1つが満足された時、その割込みを制御しているビットがアクティブになり、そして次のパケットの第1バイトにおいてリセットされる。EOP割込みによって、パケットの終りに対して相対的な割込みの位置を制御することができる。EOP割込みを発生する各種の条件（図5：論理プロック）が説明される。

【0019】

```
< 1 - エラーのパケット >
I f ( T e = 1 )
i f ( E n _ e r r o r n e o u s _ p = 1 )
それを EOP する
e l s e
「パケットを捨てる」
e n d i f
```

普通はエラーを含んでいるパケットは捨てられるが、或る場合には完全なパケットを削除するよりも小さなエラーを許容するほうがシステムにとって妨害が少なくなる可能性がある。その場合、そのパケットはバッファへ転送され、ファームウェアがそのパケットのタイプ（ビデオ、オーディオなど...）によって変わる最終の決定を行う。

【0020】

```
< 2 - 不連続性の検出 >
e l s i f ( C u r r e n t _ C C = P r e v i o u s _ C C )
「パケットを複製する」
i f ( ( A F = " 0 1 " ) a n d ( E n _ d u p l i c a t e _ p = 0 )
「パケットを捨てる」
e l s e
それを EOP する
e n d i f
e l s i f ( C u r r e n t _ C C / = P r e v i o u s _ C C + 1 )
それを EOP する
```

EOPパケット割込みは不連続性カウンタ検出の後に発生される。MPLEG2においては、CCの不連続性はアダプテーション・フィールドの中の不連続性フラグがセットされている時に可能である。後で示されるように、アダプテーション・フィールド付きのパケットはすべてEOP割込みを発生する。普通はペイロードだけを含んでいる（AF = "01"）の複製パケットは捨てられる。グローバル制御ビットEn_duplicat e_pがセットされている場合は、それらを組込みのソフトウェアへ転送してに最終の決定を行わせることができる。

【0021】

```
< 3 - 他の条件のリスト >
e l s i f ( A F = " 1 x " )
```

10

20

30

40

50

それを EOP する

```
else if (Payload_unit_start_indicator = 1)
```

それを EOP する

```
else if (Hpei = 1) and (En_Hpei = 1)
```

それを EOP する

EOP 割込みは次の場合に発生される。

- アダプテーション・フィールドを含んでいるすべてのパケット
- ペイロード・ユニット・スタート・インジケータがセットされているすべてのパケット
- ペイロード・ユニット・スタート・インジケータがセットされているパケットに続くすべてのパケット。これは PES ヘッダが 2 つ以上のパケットを取ることができるので重要である。Spei を十分早期にセットできる場合、この機能をイネーブルまたはディスエーブルすることができる（以下の図 6 参照）。

```
else if (Spei = 1)
```

それを EOP する

- 組み込まれているソフトウェアがさらに解析が必要と判断したすべてのパケット

```
else if (AF = "01" and (((Hpei = 0) and (En_Hpei = 1)) or (Spei = 0)) and (stream_id = 1))
```

「パケットを捨てる」

```
else
```

「パケットは自動的に転送される」

```
endif
```

- 組み込まれているソフトウェアによってパケットが解析されることを強制する属性がセットされていない限り、ストリーム id がアクティブである場合は、ペイロードだけを含んでいるすべてのパケットが捨てられる（前のケース）。

注：このアルゴリズムの中で順次に評価されているこれらの条件は、実際にはすべてハードウェアによって並列に計算される。

EOP の割込み条件が満足されると、組み込まれている RISC プロセッサがその解析のタスクを行う。

【0022】

<トランスポート・ストリームのソフトウェア解析>

マイクロプロセッサはトランスポート・パケット・バーザのハードウェア・モジュールよりさらに複雑な解析を実行することができ、ハードウェアのプリプロセッシングを完成する。さらに、その処理はそのパケットに対応するデータのタイプにしたがって異なる可能性がある。実際に、使われるべき特定のアルゴリズムにそのパケットを関連付けるソフトウェア・テーブルを、PID の値に基づいて定義することは非常に簡単である。

トランスポート・パケットのソフトウェア処理は割込みによって駆動される。EOP 割込みが発行されると、そのパケットを解析するプロセスが開始される。CPU の latency 応答およびビット・ストリームのレートにしたがって、2 種類の処理を選択することができる。

第 1 のケースにおいてソフトウェアは PES ヘッダを含んでいるパケットの EOP 割込みを受信するとすぐに、TPP ハードウェア・モジュールの中の Spei フラグをセットする。そのパケットを処理した後、ソフトウェアはその PES ヘッダがその次のトランスポート・パケットの中に続いているかどうかを検出することができる。PES ヘッダが現在のパケットで終っている場合、CPU はパケット・アナライザの制御レジスタの中の Spei ビットをリセットする。というのは、次のパケットを解析する必要がないからである。それ以外の場合、Spei ビットはそのまま変わらずに残される。この方法が使えるのは、入力のビット・レートと比較して CPU の latency に余裕があって、次のパケットが到着する前に Spei をセットできる場合だけである。

【0023】

反対に、第 2 の方法はどんな場合においても使うことができる。En_Hpei ビットが

10

20

30

40

50

パケット・アナライザの制御レジスタの中でセットされている場合、CPUは各PESヘッダの後のSpeiビットをセットする必要はない。というのは、PESヘッダの連續性を含んでいるパケットをTPPのハードウェアが既にARMプロセッサに対して送信することを提供しているからである。

その後、TPPハードウェア・モジュールのレジスタの中のフラグを読むことによってその割込みの理由が調べられる。その場合、アダプテーション・フィールドが処理される。アダプテーション・フィールドの解析の後、そのペイロードのうちのどのペイロードも前のパケットからの複製されたものでない場合に、そのパケットを捨てることができる。この時点で、ソフトウェア・プロセスの実行はパケットのタイプ（オーディオ、ビデオ、プライベート・データ、サービス情報、...）にしたがって、別のルーチンへ分岐する。次に、そのデータがDMAチャネルを経由して特定のバッファへ転送される。

これはパケットを解析するために使うことができるアルゴリズムである（図6参照）。次のリストの中の最初のポイント（Pesi/Psiスタート・フラグのチェックおよびSpeiフラグのセット）はオプションである。

【0024】

実際、前に説明されたように、ソフトウェアのラテンシーが十分に短くて、TPPハードウェア・モジュールが次のトランスポート・パケットを解析する前にソフトウェアがSpeiビットをセットすることができる場合、この動作によってCPUの介入が最小化される。PESヘッダの開始に続くパケットが実際に処理されなければならない場合だけ、マイクロプロセッサに対してEOP割込みが送られる。

それ以外の場合、到着しているストリームのビット・レートとCPUのラテンシーが高過ぎる場合、Hpeiメカニズムをイネーブルすることができる（En_Hpei = 1）。PESヘッダの開始点に続く同じPIDの次のパケットは常にCPUへ転送される。

【0025】

パケット・アナライザ制御レジスタ：パケット・ステータス・ワード（図4参照）の中のPesi/Psiスタート・フラグを読むことによって、現在のトランスポート・パケットの中でPESヘッダがスタートするかどうかチェックする。PESヘッダが現在のトランスポート・パケットの中でスタートする場合、パケット・アナライザ制御レジスタの中のSpeiビットが、同じPIDの次のパケットを割り込ませるためにCPUによってセットされなければならない。実際問題として、PESヘッダは次のトランスポート・パケットの中に継続する可能性がある。

【0026】

パケット・アナライザ制御レジスタ：パケット・ステータス・ワード（図4参照）の中のAFフラグを読むことによって、アダプテーション・フィールドの存在をチェックする。アダプテーション・フィールドがそのパケットの中に転送される場合、その関連するデータが処理されなければならない。特に、基準クロックの再構築を制御するためにPCRが抽出されてフィルタされなければならない。別の重要な部分は不連続性ステータスの解析から構成される。アダプテーション・フィールドの中の不連続性ステータス・ビットがセットされていた場合、その連続性カウンタの値における不連続性は欠落しているパケットには対応しない。不連続性のステータスがセットされていた場合、PCRの値（存在している場合）に対してローカル・クロックを初期化し直さなければならない。

【0027】

パケット・アナライザ制御レジスタ（図4参照）の中の複製パケット・フラグを読むことによって、そのパケットが複製されているかどうかをチェックする。

そのパケットが複製されていた場合、それは捨てられる。というのは、他の処理は不要だからである。PESヘッダが存在していてHpeiの方法が使われないのでパケットの開始時点でのSpeiビットが変更されていた場合、Speiビットはそのプロセスの終る前にクリアされなければならない。

【0028】

パケット・アナライザ制御レジスタ（図4参照）の中のAFフラグ読むことによって、そ

10

20

30

40

50

のパケットがトランスポート・ペイロードを含んでいるかどうかチェックする。
ペイロードがない場合（アダプテーション・フィールドだけの場合）そのパケットは捨てられる。というのは、他の処理は不要だからである。

【0029】

<パケット・タイプをチェックする>

この部分のアルゴリズムは、実行されるべき特定の処理に対して選択された各P I Dを関連付けるテーブルに基づいている。この方法で、そのパケットは関連のルーチンによって処理される。そのテーブルは32個のエントリーを含んでいる。それぞれが可能な選択された各P I Dに対応している各エントリーによって、そのパケットのタイプ（すなわち、オーディオ・パケット、ビデオ・パケット、P E Sプライベート・データ、P S Iなど）およびそのデータが格納されなければならないメモリのアドレスを識別することができる。特に、そのパケットのタイプがビデオ・ストリームまたはオーディオ・ストリームに対応していて（一般に自動D M A転送に対応していて）、H p e iのメカニズムが使われない場合、そのパケットを処理した後、S p e iビットはP E Sヘッダが終了している場合はクリアされなければならない。

10

【0030】

<D M A転送を開始する>

処理後のデータは最後に特定のメモリ・バッファへ転送される。

完全なアルゴリズムが図6に示されている。黒のモジュールはH p e iメカニズムがディスエーブルされている場合にだけ実行されるアルゴリズムの論理ブロックに対応している。

20

例：P E Sシンタックスを解析しないビデオ・デコーダに対するトランスポート・ストリーム。

【0031】

この場合、ビデオのエレメンタリ・ストリームだけがビデオ・デコーダへ送られる。P T Sの値が抽出されて、ビデオ・デコーダのレジスタの中に書き込まれる。P E Sヘッダの中に含められている可能性がある他のフィールドはどれも考慮されない。というのは、それはD V B勧告によって許されているからである。図7はP E Sパケットの構造を示している。

次のアルゴリズムはP E Sパケットを処理するために使われる。P E Sヘッダの最初の部分はその対象とするフィールドを処理するために一時バッファの中に格納される。異なるパケットの中にヘッダが分割されている場合、第2のパケットが受信された後そのプロセスは継続し、そしてそのヘッダの最初のバイトが一時バッファの中に書き込まれなければならない。

30

【0032】

この例はP E S解析に関連する部分だけを説明する。C P U処理の他の部分については、上記のアルゴリズムが適用される。

T P Pハードウェア・モジュールのP I D選択をイネーブルする前に、T P P制御レジスタのフラグが次のように初期化される。

E n _ e r r o n e o u s _ p = 0 エラーを含んでいるパケットが捨てられる。
E n _ d u p l i c a t e _ p = 0 アダプテーション・フィールドを含んでいない複製されたパケットが捨てられる。この場合、そのペイロードは前のパケットのペイロードと同じである。

40

S p e i = 0 C P Uの介入なしの自動転送：そのパケットは前のセクションの中で説明された条件が発生した時だけ、C P Uによって横取りされる。

s t r e a m - i d f i l t e r フラグ = 0 ストリームidフィールドについてのフィルタは不要である。

P S I フラグ = 0 ビデオ・ストリームはP S Iセクションによって転送される、P E Sシンタックスを使うことによって転送される。

E n _ H p e i = 1 H p e iメカニズムがイネーブルされている。

50

【0033】

このアルゴリズムは次の擬似コードによって説明することができる。

PesStatus は前のステータスの PES ヘッダ処理のステータスを示す (0 = PES ヘッダが終了している ; 1 = PES ヘッダが前のパケットから継続していて、まだ解析されていない ; 2 = PES ヘッダは前のパケットから継続していて、すべての有用なフィールドは既に解析されている)。

PreviousSize は前のパケットの中で解析された PES ヘッダの長さを示し、そして更新の後、前のパケットの中で解析された PES パケットのサイズに現在のトランスポート・パケットの中に含まれている PES パケットのサイズを加えたサイズを示す。

if (CC-error フラグ = 1) and

(不連続性ステータス = TRUE)

「エラーの隠蔽 (concealment) を開始する」

【0034】

連続性カウンタのエラーがあるが、その不連続性は許されない。前のパケットのいくつかが欠落している。この場合、エラーの隠蔽の適切な方法が開始されなければならない。

if (PES/PSI_start フラグ = 0) and

(PesStatus = 0)

「転送されるデータの開始点をアレンジする」

このパケットの中には PES ヘッダはない：すべてのペイロードが転送されなければならない。

else if (PesStatus = 2)

PES ヘッダの連続性があるが、すべての有用なフィールドが前のパケットの中で解析されている。

PayloadLength = 184 - AfLength

バイト単位でのペイロードの長さを計算する：ペイロード・サイズはトランスポート・パケット・サイズ (184 バイト) からアダプテーション・フィールドの長さを引いた値である。

PreviousSize = PreviousSize + PayloadLength
サイズを更新する。

if (PreviousSize < PES パケットの長さ)

「ノー・オペレーション」

PES ヘッダはこのパケットの中では終了しない：というのは、PES ペイロードがまだ始まっておらず、データが転送される必要がないからである。

else

PesStatus = 0

「転送されるべきデータの開始点をアレンジする」

PES ヘッダはこのパケットにおいて終了する；PES ペイロードが転送されなければならない。

else PesStatus は 2 に等しくない：PES ヘッダはまだ解析されていない。
。

PayloadLength = 184 - AfLength

バイト単位でペイロードの長さを計算する：ペイロードのサイズはトランスポート・パケットのサイズ (184 バイト) からアダプテーション・フィールドの長さを引いた値である。

【0035】

「一時バッファへ N バイトを付加する」 (N = 14 - PreviousSize)

PreviousSize = PreviousSize + PayloadLength
サイズを更新する。

if (PreviousSize < 9)

PesStatus = 1 ;

10

20

30

40

50

P E S ヘッダの固定部分はこのパケットの中には完全には含まれていない。

else PreviousSize は 9 より大きいかまたは等しい： P E S ヘッダの固定部分のすべてのフラグがこのパケットの中にある。

if (pts_flag = 0) PTS は P E S ヘッダの中にはない

if (PreviousSize < P E S ヘッダの長さ)

PesStatus = 2 ; 有用なフィールドはこれ以上ないが、 P E S ヘッダはこのパケットの中では終了しない。

else

PesStatus = 0 ;

「転送されるべきデータの開始点をアレンジする」

10

P E S ヘッダはこのパケットの中で終了する： P E S ペイロードが転送されなければならない。

else P E S ヘッダは PTS を含んでいる。

if (PreviousSize < 14) PTS は次のパケットの中にある。

PrsStatus = 1 ;

else PTS はこのパケットの中にある。

「 PTS をビデオ・デコーダへ書き込む」

if (PreviousSize < P E S ヘッダの長さ)

ヘッダはこのパケットの中では終了しない。

PesStatus = 2 ;

20

else ヘッダは終了している。

PesStatus = 0 ;

「転送されるべきデータの開始点をアレンジする」

P E S ヘッダはこのパケットの中で終了する： P E S ペイロードが転送されなければならない。

【 0036 】

この場合、パケットは P E S ヘッダが存在している限り C P U へ送られる。その発生はトランスポート・パケット・ヘッダの中の P E S インジケータの値をチェックすることによって簡単に知ることができる。

ビデオ・ストリームが 15 M b p s のビット・レートでフレームにコード化されていると仮定する。 C P U が各パケットを処理しなければならない場合、 C P U は約 100 μ s ごとに割り込まれる。

30

反対に、 T P P ハードウェアによって実行される事前のフィルタおよびルーティングを用ることによって、 C P U は P E S パケットごとに割り込まれる。その P E S パケットの長さが少なくともフレーム・サイズに等しい場合、割込みは最悪の場合 40 m s ごとに発生される。

例： P E S シンタックスを解析するオーディオ・デコーダに対するトランスポート・ストリーム、ストリーム i d フィールドについてのフィルタが実行される。

この例では、オーディオ・デコーダは P E S パケットを解析することができるが、 C P U は特定の値とは異なるストリーム i d を持つすべての P E S パケットをフィルタしなければならない（そのストリーム i d が同じである場合）。

40

【 0037 】

図 6 に示されていて前のページで説明された一般的なアルゴリズムが使われる。次の部分はオーディオの P E S パケットに関連している処理の特定の部分だけを説明する。特に、 H p e i メカニズムはディスエーブルされていて、したがって、 C P U は P E S ヘッダの開始点を含んでいるトランスポート・パケットが受信されるたびに S p e i フラグをセットする。

T P P ハードウェア・モジュールの P I D 選択をイネーブルする前に、 T P P 制御レジスタのフラグが次のように初期化される。

E n _ e r r o n e o u s _ p = 0 エラーを含んでいるパケットが捨てられる。

50

`En_duplicat e_p = 0` アダプテーション・フィールドを含んでいない複製されたパケットが捨てられる：この場合、そのペイロードは前のパケットのペイロードと同じである。

`Spei = 0` C P U の介入なしの自動転送：そのパケットは前のセクションで説明されている条件が発生した時だけ C P U によって横取りされる。

`stream_id` フィルタ・フラグ = 1 ストリーム `id` フィールドについてのフィルタが必要である。

`PSI` フラグ = 0 オーディオ・ストリームは `PSI` セクションによってではなく、P E S シンタックスを使うことによって転送される。

`En_Hpei = 0` `Hpei` メカニズムがディスエーブルされており、C P U は `Spei` をセットおよびリセットしなければならない。 10

初期化後、T P P ハードウェア・モジュールは最初の P E S ヘッダが受信されるまで、すべてのパケットを捨てる。この時点で、E O P 割込みが発行され、C P U がそのパケットを解析する。

ストリーム `id` の値が特定のパターンにマッチした場合、すべての P E S パケットはオーディオ・デコーダへ送られなければならない。結果として、マイクロプロセッサはパケット・アナライザ制御レジスタの中のストリーム `id` フィルタ・フラグをクリアし、すべてのトランスポート・パケットはオーディオ・デコーダへ自動的に転送される。別の P E S ヘッダがT P P ハードウェアによって見つけられると（そのトランSPORT・ヘッダの中パケット・ユニット・スタート・インジケータがセットされている）、割込みが発生され、そしてC P U はその新しいP E S パケットのストリーム `id` 値について別のフィルタ操作を実行することができる。 20

ストリーム `id` の値が特定のパターンにマッチしなかった場合、C P U はT P P モジュールの中のストリーム `id` フラグをセットし、同じ P I D を持つそれ以降のすべてのトランSPORT・パケットは、次の P S ヘッダまでハードウェアによって自動的に捨てられる。

【0038】

このアルゴリズムは次の擬似コードによって説明することができる。C P U はすべての P E S ヘッダを解析せず、ストリーム `id` フィールドだけを解析する。 `PesStatus` は前のステータスの P E S ヘッダ処理のステータスを示す（0 = P E S ヘッダが終了しているか、あるいは P E S ヘッダは前のパケットから継続しているが、ストリーム `id` 値が既にフィルタされている；1 = その P E S ヘッダが前のパケットから継続していて、そのストリーム `id` 値がまだフィルタされていない）。 30

`PreviousSize` は前のパケットの中で読まれた P E S ヘッダの長さを示す。

`if (ストリーム id フィルタ・フラグ = 0) and`

`(PES / PSI スタート・フラグ = 0)`

「そのパケットを捨てる」

現在のパケットはストリーム `id` フィルタ・パターンにマッチせず、そして P E S ヘッダがない：そのパケットは捨てられなければならない。

`else`

`if (CC-error フラグ = 1) and`

`(不連続性ステータス = TRUE)`

「エラーの隠蔽を開始する」

連続性カウンタのエラーが存在するが、その不連続性は許されない。前のパケットのいくつかが欠落している。この場合、エラー隠蔽の適切な方法が開始されなければならない。

`if (PES / PSI スタート・フラグ = 0) and`

`(PesStatus = 0)`

「転送されるべきデータの開始点をアレンジする」

このパケットには P E S ヘッダがない；すべてのペイロードが転送されなければならない。 40

else PESヘッダまたはPESの継続がある。

PayloadLength = 184 - AfLength

バイト単位でペイロードの長さを計算する；ペイロード・サイズはトランSPORT・パケットのサイズ（184バイト）からアダプテーション・フィールドの長さを引いた値である。

【0039】

PreviousSize = PreviousSize + PayloadLength

サイズを更新する。

if (PreviousSize < 4)

PesStatus = 1;

10

ストリームidはこのパケットには含まれていない。次のトランSPORT・パケットが解析されなければならない。

else PreviousSizeが4より大きいかまたは4に等しい：ストリームidフィールドがこのパケットの中に含まれている。

PesStatus = 0；CPUは次のパケットを解析する必要はない。

if (ストリームid = 「パターン」) ストリームidフィールドをフィルタする；その値が特定のパターンに等しかった場合、PESパケットが選択される。

「ストリームidフィルタ・フラグをクリアする」 TPPハードウェアによるパケットの自動棄却をディスエーブルする。

「転送されるべきデータの開始点をアレンジする」

20

PESヘッダはこのパケットで終了する；PESペイロードが転送されなければならない。

else ストリームidフィールドが指定されたパターンと異なっている；PESパケットは捨てられなければならない。

「ストリームidフィルタ・フラグをセットする」 TPPハードウェアによるパケットの自動棄却をイネーブルする。

「パケットを捨てる」

【0040】

この時点でSpeiビットはPesStatusが0に等しい場合だけクリアされる。というのは、PESヘッダの考慮される部分が既に処理されているからである。

30

この場合、ストリームid値をフィルタするためにPESヘッダが存在する時だけCPUに対して割込みが送られる。ストリームidが選択された値とマッチした場合、次のパケットは宛先のバッファへ自動的に転送される。反対に、ストリームidがその選択された値とマッチしなかった場合、次のPESヘッダまで、それに続くすべてのパケットはTPPハードウェア・モジュールによって捨てられる。

このメカニズムなしでは、CPUはストリームidの値が望ましくないPESパケットの中に含まれているすべてのトランSPORT・パケットを捨てるための割込みを強制的に受信させられる。

【0041】

以上の説明に関して更に以下の項を開示する。

40

(1) トランSPORT・ストリーム・パーザ・システムであって、1つの関連するフラグによって事前処理された後のパケットを含めるための中間バッファと、送信フラグによって選択されたパケットをさらにソフトウェアで処理するためのプロセッサとを含むシステム。

【図面の簡単な説明】

【図1】188個のバイトを含んでいるトランSPORT・パケットを示す図。

【図2】従来技術のデコーダ・システムを示す図。

【図3】入力データ・パケットに対する2つのデータ・フロー・パスを示す図。

【図4】割込みを発生するための情報および条件を示す図。

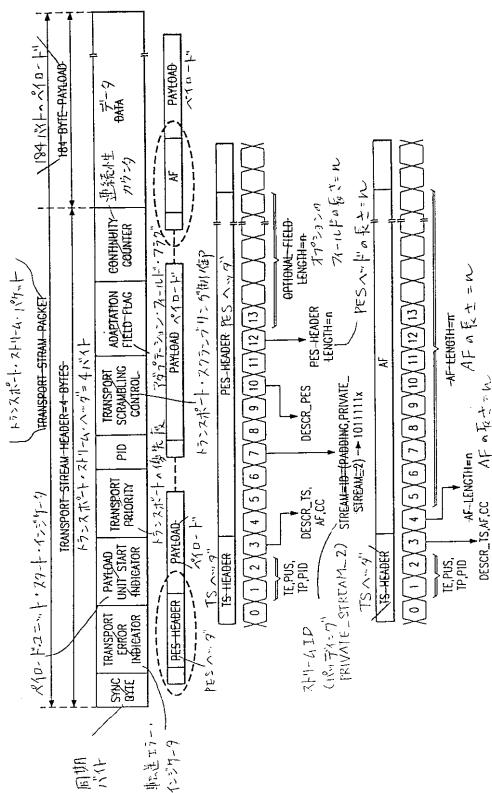
【図5】本発明のシステムのブロック図を示す図。

50

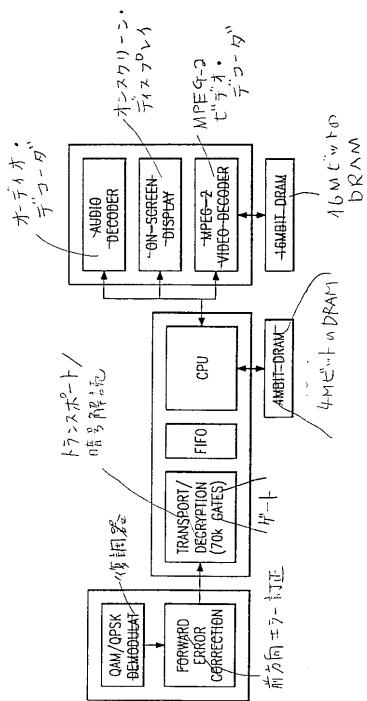
【図6】パケットを解析するために使われるフロー・チャート

【図7】PESパケットの構造を示す図。

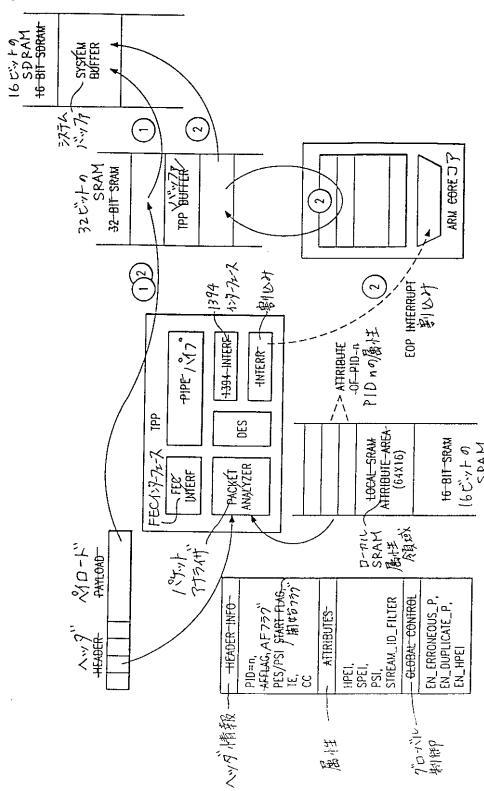
【図1】



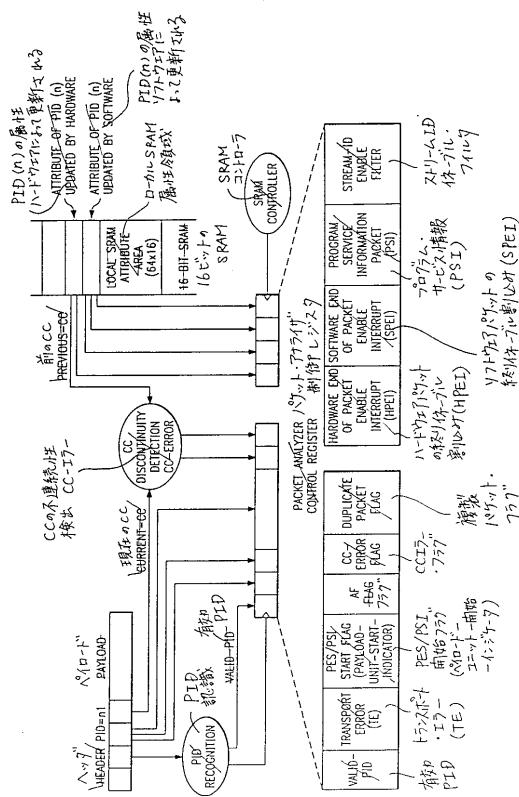
【図2】



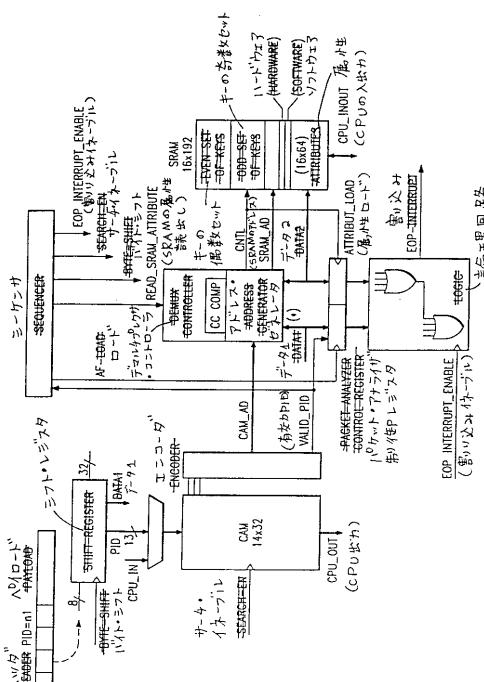
【図3】



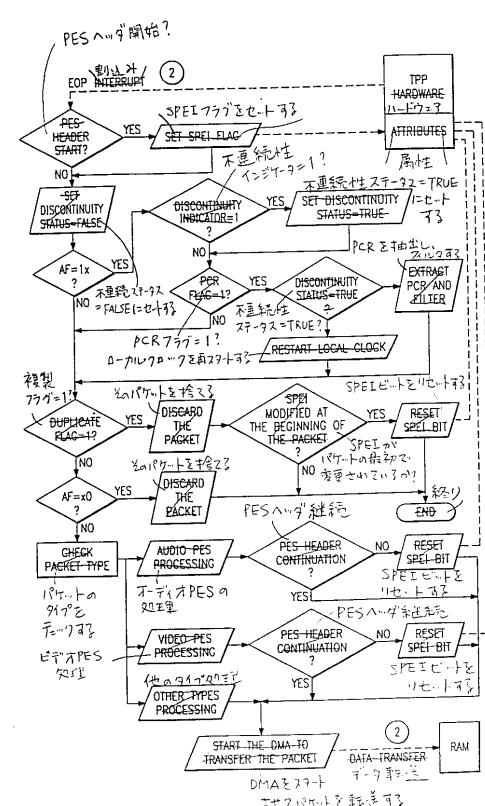
【 図 4 】



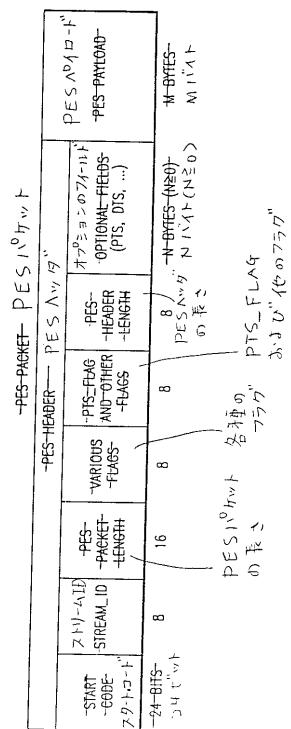
【 囧 5 】



【 四 6 】



【図7】



フロントページの続き

(72)発明者 ジエラール ショベル

フランス国アンティーブ, シュマン ド ラ シュゲット 300, レ ベルジェル ド バル
コンスタンス

(72)発明者 セルジュ ラサール

フランス国フルジュ, サン ジャン ド カンヌ, リュ ドュ マルソー 29

(72)発明者 マリオ ジアニ

フランス国カーニュ シュル メール, ルート ド グラッセ, レジダンス ラオシス - パチ
マン ピー, 73

(72)発明者 ティエメン スピッツ

アメリカ合衆国テキサス州ノース リッチランド ヒルズ, エリス ドライブ 8528

審査官 菅原 道晴

(56)参考文献 特開平08-265746 (JP, A)

特開平08-214312 (JP, A)

(58)調査した分野(Int.Cl., DB名)

H04N 7/26-7/68