

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2005-123739

(P2005-123739A)

(43) 公開日 平成17年5月12日(2005.5.12)

(51) Int. Cl.⁷

H04L 12/28

F I

H04L 12/28 200M

H04L 12/28 400

テーマコード(参考)

5K033

審査請求 未請求 請求項の数 35 O L (全 44 頁)

(21) 出願番号 特願2003-354142 (P2003-354142)

(22) 出願日 平成15年10月14日(2003.10.14)

(71) 出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(74) 代理人 100090273

弁理士 國分 孝悦

(72) 発明者 新井田 光央

東京都大田区下丸子3丁目30番2号 キ

ヤノン株式会社内

Fターム(参考) 5K033 AA02 CB01 CC01 DA11 EA02

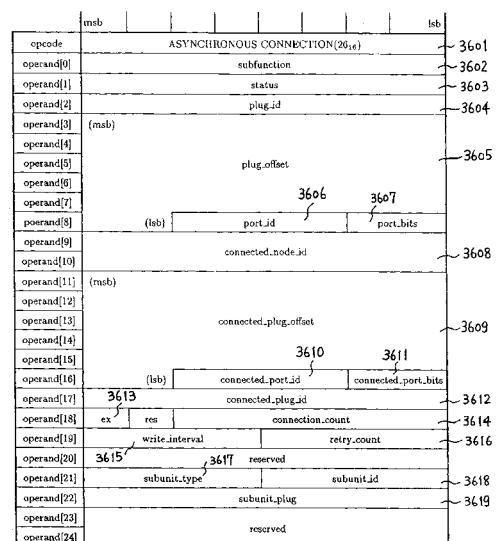
(54) 【発明の名称】 通信システム、端末装置、通信システムの制御方法、記録媒体およびプログラム

(57) 【要約】

【課題】 簡便な接続手順で、高速なデータ転送が可能であるとともにデータ転送の確実性が保証できる通信システムを提供する。

【解決手段】 ユニット間のアシンクロナスコネクションを確立するためのコマンドに、ユニット内のサブユニットプラグを指定するためのフィールドを拡張して追加し、ユニット間におけるアシンクロナスコネクション、およびユニット内でのサブユニットプラグとユニットが備えるポートとの間におけるコネクションを単一のコマンドにより、同時に接続あるいは切断することができるようにする。

【選択図】 図36



【特許請求の範囲】**【請求項 1】**

シリアルデータを送受信可能なシリアルバスを介して、データを送信する送信ノードと、上記送信ノードから送信されたデータを受信する受信ノードとが接続された通信システムであって、

任意の制御データを出力する制御手段を備え、

上記送信ノードは、上記シリアルバスを介してデータを送信するための送信手段と、

上記送信手段にデータを出力可能な出力手段を有するとともに、上記制御手段からの制御データにより制御可能な所定の機能を有する機能動作手段と、

上記送信手段と上記出力手段との接続状態を制御する通信路制御手段とを備え、

上記制御データは、単一の制御命令を用いて送信可能な上記送信手段と上記出力手段との接続状態を制御するための制御データを含むことを特徴とする通信システム。

10

【請求項 2】

上記制御データは、上記送信手段を識別するデータおよび上記出力手段を識別するためのデータを含むことを特徴とする請求項 1 に記載の通信システム。

【請求項 3】

上記制御データは、さらに上記送信手段と上記出力手段との接続状態の制御にて排他制御を行うためのデータを含むことを特徴とする請求項 1 または 2 に記載の通信システム。

【請求項 4】

上記機能動作手段は、データを記録するための記録手段であり、

上記送信手段は、上記記録手段に記録されたデータを送信可能であることを特徴とする請求項 1 ~ 3 の何れか 1 項に記載の通信システム。

20

【請求項 5】

上記記録手段は、ファイルデータおよびファイル属性データを記録することを特徴とする請求項 4 に記載の通信システム。

【請求項 6】

シリアルデータを送受信可能なシリアルバスを介して、データを送信する送信ノードと、上記送信ノードから送信されたデータを受信する受信ノードとが接続された通信システムであって、

任意の制御データを出力する制御手段を備え、

上記受信ノードは、上記シリアルバスを介してデータを受信するための受信手段と、

上記受信手段で受信したデータを入力可能な入力手段を有するとともに、上記制御手段からの制御データにより制御可能な所定の機能を有する機能動作手段と、

上記受信手段と上記入力手段との接続状態を制御する通信路制御手段とを備え、

上記制御データは、単一の制御命令を用いて送信可能な上記受信手段と上記入力手段との接続状態を制御するための制御データを含むことを特徴とする通信システム。

30

【請求項 7】

上記制御データは、上記受信手段を識別するデータおよび上記入力手段を識別するためのデータを含むことを特徴とする請求項 6 に記載の通信システム。

【請求項 8】

上記制御データは、さらに上記受信手段と上記入力手段との接続状態の制御にて排他制御を行うためのデータを含むことを特徴とする請求項 6 または 7 に記載の通信システム。

40

【請求項 9】

上記シリアルバスは、IEEE 1394 規格に準拠したシリアルバスであり、上記送信ノードと上記受信ノードとの間でのデータの送受信はアシンクロナス転送により行われることを特徴とする請求項 1 ~ 8 の何れか 1 項に記載の通信システム。

【請求項 10】

任意の制御データを送信する制御ノードと、上記制御データを受信する送信ノードおよび受信ノードとがシリアルデータを送受信するためのシリアルバスを介して接続された通信システムであって、

50

上記送信ノードは、上記制御データの一部により制御可能であるとともに、データを出力するための出力手段をそれぞれ備えた所定の機能を有する n 個 (n は自然数) の機能動作手段と、

上記シリアルバスを介してアシンクロナス転送によりデータを送信するための送信手段とを具備し、

上記受信ノードは、上記シリアルバスを介してアシンクロナス転送により上記データを受信するための受信手段を具備し、

上記制御ノードは、単一のデータ送信により上記出力手段と上記送信手段との間のデータ送信路を確保するためのデータを含む上記制御データを上記送信ノードに送信することを特徴とする通信システム。

10

【請求項 1 1】

上記制御ノードは、上記出力手段と上記送信手段との間のデータ送信路の確保に関する排他制御を行うためのデータを含む上記制御データを上記送信ノードに送信することを特徴とする請求項 1 0 に記載の通信システム。

【請求項 1 2】

任意の制御データを送信する制御ノードと、上記制御データを受信する送信ノードおよび受信ノードとがシリアルデータを送受信するためのシリアルバスを介して接続された通信システムであって、

上記送信ノードは、上記シリアルバスを介してアシンクロナス転送によりデータを送信するための送信手段を具備し、

20

上記受信ノードは、上記制御データの一部により制御可能であるとともに、データを入力するための入力手段をそれぞれ備えた所定の機能を有する m 個 (m は自然数) の機能動作手段と、

上記シリアルバスを介してアシンクロナス転送により上記データを受信するための受信手段とを具備し、

上記制御ノードは、単一のデータ送信により上記入力手段と上記受信手段との間のデータ受信路を確保するためのデータを含む上記制御データを上記受信ノードに送信することを特徴とする通信システム。

【請求項 1 3】

上記制御ノードは、上記入力手段と上記受信手段との間のデータ受信路の確保に関する排他制御を行うためのデータを含む上記制御データを上記受信ノードに送信することを特徴とする請求項 1 2 に記載の通信システム。

30

【請求項 1 4】

任意の制御データを送信する制御ノードと、上記制御データを受信する送信ノードおよび受信ノードとがシリアルデータを送受信するためのシリアルバスを介して接続された通信システムであって、

上記送信ノードは、データを記録するための記録手段と、

上記記録手段に記録されているデータを上記シリアルバスを介してアシンクロナス転送により送信するための送信手段とを具備し、

上記受信ノードは、上記送信ノードから送信されたデータを受信するための受信手段を具備し、

40

上記制御データは、単一のデータ送信により上記記録手段と上記送信手段との論理的な接続を行うための制御データを含むことを特徴とする通信システム。

【請求項 1 5】

上記記録手段は、ファイルデータおよびファイル属性データを記録することを特徴とする請求項 1 4 に記載の通信システム。

【請求項 1 6】

上記制御データは、上記記録手段と上記送信手段と上記受信ノードとの接続制御にて排他制御を行うための制御データを含むことを特徴とする請求項 1 4 または 1 5 に記載の通信システム。

50

【請求項 17】

任意の制御データを送信する制御ノードと、上記制御データを受信する送信ノードおよび受信ノードとがシリアルデータを送受信するためのシリアルバスを介して接続された通信システムであって、

上記送信ノードは、ファイルシステムを具備するとともに、当該ファイルシステムを上記シリアルバスを介してアシンクロナス転送により上記受信ノードに提供し、

上記制御データは、単一のデータ送信により上記ファイルシステムとの論理的な接続を行うための制御データを含むことを特徴とする通信システム。

【請求項 18】

上記制御データは、上記ファイルシステムと上記受信ノードとの接続制御にて排他制御を行うための制御データを含むことを特徴とする請求項 17 に記載の通信システム。

10

【請求項 19】

上記制御ノードと上記受信ノードとが、同一のノードであることを特徴とする請求項 10 ~ 18 の何れか 1 項に記載の通信システム。

【請求項 20】

シリアルデータを送受信可能なシリアルバスを介して、データを受信する受信ノードに接続された端末装置であって、

上記シリアルバスを介してデータを送信するための送信手段と、

上記送信手段にデータを出力可能な出力手段を有するとともに、制御データに基づいて制御可能な所定の機能を有する機能動作手段と、

20

上記送信手段と上記出力手段との接続状態を、単一の制御命令を用いて送信された制御データに基づいて制御可能な通信路制御手段とを備えることを特徴とする端末装置。

【請求項 21】

上記単一の制御命令を用いて送信される制御データは、上記送信手段と上記出力手段との接続状態の制御にて排他制御を行うためのデータを含むことを特徴とする請求項 20 に記載の端末装置。

【請求項 22】

シリアルデータを送受信可能なシリアルバスを介して、データを送信する送信ノードに接続された端末装置であって、

上記シリアルバスを介してデータを受信するための受信手段と、

30

上記受信手段で受信したデータを入力可能な入力手段を有するとともに、制御データに基づいて制御可能な所定の機能を有する機能動作手段と、

上記受信手段と上記入力手段との接続状態を、単一の制御命令を用いて送信された制御データに基づいて制御可能な通信路制御手段とを備えることを特徴とする端末装置。

【請求項 23】

上記単一の制御命令を用いて送信される制御データは、上記受信手段と上記入力手段との接続状態の制御にて排他制御を行うためのデータを含むことを特徴とする請求項 22 に記載の端末装置。

【請求項 24】

シリアルデータを送受信可能なシリアルバスを介して、データを送信する送信ノードと、上記送信ノードから送信されたデータを受信する受信ノードとが接続された通信システムの制御方法であって、

40

上記送信ノードが備える、所定の機能を有する機能動作手段からデータを出力するための出力手段と、上記シリアルバスを介してデータを外部に送信するための送信手段との接続状態を、単一の制御命令を用いて送信された制御データに基づいて制御することを特徴とする通信システムの制御方法。

【請求項 25】

上記送信手段と上記出力手段との接続状態の制御にて、上記単一の制御命令を用いて送信された制御データに基づき排他制御を行うことを特徴とする請求項 24 に記載の通信システムの制御方法。

50

【請求項 26】

シリアルデータを送受信可能なシリアルバスを介して、データを送信する送信ノードと、上記送信ノードから送信されたデータを受信する受信ノードとが接続された通信システムの制御方法であって、

上記受信ノードが備える、所定の機能を有する機能動作手段にデータを入力するための入力手段と、上記シリアルバスを介してデータを外部から受信するための受信手段との接続状態を、単一の制御命令を用いて送信された制御データに基づいて制御することを特徴とする通信システムの制御方法。

【請求項 27】

上記受信手段と上記入力手段との接続状態の制御にて、上記単一の制御命令を用いて送信された制御データに基づき排他制御を行うことを特徴とする請求項 26 に記載の通信システムの制御方法。

10

【請求項 28】

シリアルデータを送受信可能なシリアルバスを介してデータを受信する受信ノードに接続された送信ノードが備える、所定の機能を有する機能動作手段からデータを出力するための出力手段と、上記シリアルバスを介してデータを外部に送信するための送信手段との接続状態の制御を、単一の制御命令を用いて送信された制御データに基づいてコンピュータに実行させるためのプログラム。

【請求項 29】

上記単一の制御命令を用いて送信された制御データに基づいて、さらに上記送信手段と上記出力手段との接続状態の排他制御をコンピュータに実行させるための請求項 28 に記載のプログラム。

20

【請求項 30】

シリアルデータを送受信可能なシリアルバスを介してデータを送信する送信ノードに接続された受信ノードが備える、所定の機能を有する機能動作手段にデータを入力するための入力手段と、上記シリアルバスを介してデータを外部から受信するための受信手段との接続状態の制御を、単一の制御命令を用いて送信された制御データに基づいてコンピュータに実行させるためのプログラム。

【請求項 31】

上記単一の制御命令を用いて送信された制御データに基づいて、さらに上記受信手段と上記入力手段との接続状態の排他制御をコンピュータに実行させるための請求項 30 に記載のプログラム。

30

【請求項 32】

シリアルデータを送受信可能なシリアルバスを介してデータを受信する受信ノードに接続された送信ノードが備える、所定の機能を有する機能動作手段からデータを出力するための出力手段と、上記シリアルバスを介してデータを外部に送信するための送信手段との接続状態の制御を、単一の制御命令を用いて送信された制御データに基づいてコンピュータに実行させるためのプログラムを記録したことを特徴とするコンピュータ読み取り可能な記録媒体。

【請求項 33】

上記単一の制御命令を用いて送信された制御データに基づいて、さらに上記送信手段と上記出力手段との接続状態の排他制御をコンピュータに実行させるためのプログラムを記録したことを特徴とする請求項 32 に記載のコンピュータ読み取り可能な記録媒体。

40

【請求項 34】

シリアルデータを送受信可能なシリアルバスを介してデータを送信する送信ノードに接続された受信ノードが備える、所定の機能を有する機能動作手段にデータを入力するための入力手段と、上記シリアルバスを介してデータを外部から受信するための受信手段との接続状態の制御を、単一の制御命令を用いて送信された制御データに基づいてコンピュータに実行させるためのプログラムを記録したことを特徴とするコンピュータ読み取り可能な記録媒体。

50

【請求項 35】

上記単一の制御命令を用いて送信された制御データに基づいて、さらに上記受信手段と上記入力手段との接続状態の排他制御をコンピュータに実行させるためのプログラムを記録したことを特徴とする請求項 34 に記載のコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、通信システム、端末装置、通信システムの制御方法、記録媒体およびプログラムに関し、特に、制御信号とデータとを混在させて通信可能なデータ通信バスを介して、複数の電子機器を通信可能なように接続し、各機器間でデータ通信を行う通信システムに用いて好適なものである。

10

【背景技術】

【0002】

パーソナル・コンピュータ（以下、「PC」と称す。）の周辺機器の中で、ハードディスクやプリンタは利用頻度が高く、小型コンピュータ用汎用型インターフェースで代表的なデジタルインターフェース（以下、「デジタルI/F」と称す。）であるSCSI（Small Computer System Interface）等を用いてPCに対して接続され、データ通信が行われている。また、デジタルカメラやデジタルビデオカメラ等の画像記録再生装置も、PCとの間でデータ通信可能な周辺機器の1つであり、近年、画像記録再生装置により撮影した映像（静止画像や動画像）をPCに取り込んで、ハードディスクに記憶保存したり、またはPCを用いて編集した後、プリンタでカラー印刷したりする技術が発達し、画像記録再生装置により撮影した映像を保存、編集するユーザーも増加している。

20

【0003】

上述のように、取り込んだ画像データをPCからプリンタやハードディスクに出力する際などには、上記SCSI等のデジタルI/Fを介してデータ通信が行われるので、画像データのようにデータ量が多い情報を送るためにも、転送データレートが高く、かつ汎用性を有するデジタルI/Fが必要とされる。このような要望に応えるものとして、IEEE 1394規格に準拠したデータ通信バス（以下、「1394シリアルバス」と称す。）がある。

【0004】

30

以下に、1394シリアルバスの概要について説明する。

家庭用デジタルVTRやDVDの登場に伴い、ビデオデータやオーディオデータ等のリアルタイム性が要求され、かつ情報量が多いデータを、リアルタイムで転送して、PCに取り込んだり、その他のデジタル機器に転送したりするには、高速データ転送可能なデジタルI/Fが必要である。そのような観点から開発されたデジタルI/F規格の1つが、IEEE 1394 - 1995規格（High Performance Serial Bus）である。

【0005】

1394シリアルバスにおける各機器間の接続方式は、ダイジーチェーン方式とノード分岐方式とを混在可能としたものであり、高い自由度で各機器を接続することができる。また、各機器は、固有のノードIDをそれぞれ有し、互いに認識し合うことによって、1394シリアルバスを介して接続された範囲において1つのネットワークを構成する。各機器間をそれぞれ1本の1394シリアルバスケーブルで順次接続するだけで、それぞれの機器が中継の役割を担い、全体として1つのネットワークを構成するものである。

40

【0006】

また、機器の電源を入れたままケーブルを抜き差しできる、いわゆるホットスワップ機能や、1394シリアルバスのケーブルを機器に接続した時点で、機器や接続状況等を自動的に認識する、いわゆるプラグ・アンド・プレイ機能を有している。

【0007】

1394シリアルバスは、全体としてレイヤ（階層）構造で構成されている。1394シリアルバスにおいて、最もハードウェア的であるのが1394シリアルバスのケーブル

50

であり、そのケーブルのコネクタが接続されるコネクタポートがある。その上階層には、ハードウェアとして物理レイヤとリンク・レイヤとがある。さらに、その上には、通常はソフトウェアとして構成されるトランザクション・レイヤおよびアプリケーション・レイヤがある。

【 0 0 0 8 】

1394シリアルバスの接続ケーブルは、2組のツイストペア信号線に加え、電源線を設けている。これにより、電源を備えない機器や、故障等により電源電圧が低下した機器等にも電力を供給することが可能になっている。電源線内を流れる電源の電圧は8～40V、電流は最大電流DC1.5Aと規定されている。

【 0 0 0 9 】

1394シリアルバスにおけるハードウェア部は、実質的なインターフェースチップ（IC等）の部分であり、ハードウェア部の物理レイヤは、符号化やコネクタ関連の制御等を行い、リンク・レイヤはパケット転送やサイクルタイムの制御等を行う。また、ファームウェア部のトランザクション・レイヤは、転送（トランザクション）すべきデータの管理を行い、読み出し（Read）や書き込み（Write）等の命令を出す。シリアルバスマネジメントは、接続されている各機器の接続状況やノードIDの管理を行い、ノード制御やネットワーク構成を管理する部分である。

【 0 0 1 0 】

また、データ転送速度は100/200/400Mbpsが規定されており、データ転送速度が機器間で異なる場合には、上位の転送速度を有する機器が下位の転送速度をサポートして互換をとる。データ転送モードは、制御信号（制御データ）等のアシンクロナスデータを転送するアシンクロナス（Asynchronous）転送モード、およびリアルタイムなビデオデータやオーディオデータ等のアイソクロナスデータを転送するアイソクロナス（Isynchronous）転送モードがある。アシンクロナスデータとアイソクロナスデータとは、各サイクル（通常1サイクルは125μs）の開始を示すサイクル・スタート・パケット（CSP）の転送に続いて、アイソクロナスデータの転送を優先しつつ各サイクル内で混在して転送可能である。

【 0 0 1 1 】

上述したように1394シリアルバスでは、接続されている各機器（ノード）はノードIDを与えられ、ネットワーク構成として認識されている。ここで、例えば、ケーブルの挿抜や電源のON/OFF操作等によるノード数の増減などによりネットワーク構成に変化が生じ、ネットワーク構成を新たに認識する必要が生じたとする。このとき、変化を検知したノードは1394シリアルバス上にバスリセット信号を送信し、新たなネットワーク構成を認識する動作モードに入る。なお、ネットワーク構成の変化の検知方法は、1394ポート基板上でのバイアス電圧の変化を検知することにより行われる。

【 0 0 1 2 】

各ノードの物理レイヤは、1394シリアルバスを介してあるノードから伝達されたバスリセット信号を受信すると同時にリンク・レイヤにバスリセットの発生を伝達するとともに、他のノードにバスリセット信号を伝達する。最終的に、すべてのノードがバスリセット信号を検知した後、バスリセットが起動となる。

【 0 0 1 3 】

バスリセットは、上述したようなケーブルの挿抜や通信異常等のハードウェア検出による起動に限らず、プロトコルからのホスト制御等により物理レイヤに対して命令を直接出すことによっても起動する。また、バスリセットが起動すると、1394シリアルバスを介したデータ転送は、一時中断されてバスリセットの間のデータ転送は待たされる。そして、バスリセット終了後に、新しいネットワーク構成のもとでデータ転送が再開される。以上がバスリセットのシーケンスである。

【非特許文献1】IEEE Std 1394-1995, IEEE Standard for a High Performance Serial Bus, Institute of Electrical and Electronics Engineers, Inc.

【 発明の開示 】

10

20

30

40

50

【発明が解決しようとする課題】

【0014】

ここで、上述した従来例においては、記録または記憶されたファイルや、ファイルの持つ属性データを、アイソクロナスデータを用いて高速に転送することが可能である。しかしながら、このアイソクロナス転送の場合にはデータ転送の確実性は保証されない。また、データ転送の確実性が保証されているアシンクロナスデータを用いたアシンクロナス転送では、記録または記憶されたファイルや、ファイルの持つ属性データを転送するためのメカニズムが定義されていないという問題点があった。

【0015】

本出願人は、上記問題点を解決するための1つの方法を提案しているが、上記方法においては、機器内部の複数の機能ブロックと当該機器の入出力ポートとの接続手順、および機器間の接続手順を別にそれぞれ設ける必要があり、接続手順が煩雑であった。また、上記方法においては、上記機器内部の複数の機能ブロックと当該機器の入出力ポートとの接続、および機器間の接続に関して、排他制御が不可能であった。

10

【0016】

本発明は、上記問題点に鑑みて成されたものであり、簡便な接続手順で、高速なデータ転送が可能であるとともにデータ転送の確実性が保証できる通信システムを提供することを目的とする。また、機器内部の複数の機能ブロックと当該機器の入出力ポートとの接続、および機器間の接続にて排他制御できるようにすることを目的とする。

【課題を解決するための手段】

20

【0017】

本発明の通信システムは、シリアルデータを送受信可能なシリアルバスを介して、データを送信する送信ノードと、上記送信ノードから送信されたデータを受信する受信ノードとが接続された通信システムであって、任意の制御データを出力する制御手段を備え、上記送信ノードは、上記シリアルバスを介してデータを送信するための送信手段と、上記送信手段にデータを出力可能な出力手段を有するとともに、上記制御手段からの制御データにより制御可能な所定の機能を有する機能動作手段と、上記送信手段と上記出力手段との接続状態を制御する通信路制御手段とを備え、上記制御データは、単一の制御命令を用いて送信可能な上記送信手段と上記出力手段との接続状態を制御するための制御データを含むことを特徴とする。

30

【0018】

本発明の通信システムの他の特徴とするところは、シリアルデータを送受信可能なシリアルバスを介して、データを送信する送信ノードと、上記送信ノードから送信されたデータを受信する受信ノードとが接続された通信システムであって、任意の制御データを出力する制御手段を備え、上記受信ノードは、上記シリアルバスを介してデータを受信するための受信手段と、上記受信手段で受信したデータを入力可能な入力手段を有するとともに、上記制御手段からの制御データにより制御可能な所定の機能を有する機能動作手段と、上記受信手段と上記入力手段との接続状態を制御する通信路制御手段とを備え、上記制御データは、単一の制御命令を用いて送信可能な上記受信手段と上記入力手段との接続状態を制御するための制御データを含むことを特徴とする。

40

【0019】

本発明の通信システムのその他の特徴とするところは、任意の制御データを送信する制御ノードと、上記制御データを受信する送信ノードおよび受信ノードとがシリアルデータを送受信するためのシリアルバスを介して接続された通信システムであって、上記送信ノードは、上記制御データの一部により制御可能であるとともに、データを出力するための出力手段をそれぞれ備えた所定の機能を有する n 個(n は自然数)の機能動作手段と、上記シリアルバスを介してアシンクロナス転送によりデータを送信するための送信手段とを具備し、上記受信ノードは、上記シリアルバスを介してアシンクロナス転送により上記データを受信するための受信手段を具備し、上記制御ノードは、単一のデータ送信により上記出力手段と上記送信手段との間のデータ送信路を確保するためのデータを含む上記制御

50

データを上記送信ノードに送信することを特徴とする。

【0020】

本発明の通信システムのその他の特徴とするところは、任意の制御データを送信する制御ノードと、上記制御データを受信する送信ノードおよび受信ノードとがシリアルデータを送受信するためのシリアルバスを介して接続された通信システムであって、上記送信ノードは、上記シリアルバスを介してアシンクロナス転送によりデータを送信するための送信手段を具備し、上記受信ノードは、上記制御データの一部により制御可能であるとともに、データを入力するための入力手段をそれぞれ備えた所定の機能を有する m 個（ m は自然数）の機能動作手段と、上記シリアルバスを介してアシンクロナス転送により上記データを受信するための受信手段とを具備し、上記制御ノードは、単一のデータ送信により上記入力手段と上記受信手段との間のデータ受信路を確保するためのデータを含む上記制御データを上記受信ノードに送信することを特徴とする。

10

【0021】

本発明の通信システムのその他の特徴とするところは、任意の制御データを送信する制御ノードと、上記制御データを受信する送信ノードおよび受信ノードとがシリアルデータを送受信するためのシリアルバスを介して接続された通信システムであって、上記送信ノードは、データを記録するための記録手段と、上記記録手段に記録されているデータを上記シリアルバスを介してアシンクロナス転送により送信するための送信手段とを具備し、上記受信ノードは、上記送信ノードから送信されたデータを受信するための受信手段を具備し、上記制御データは、単一のデータ送信により上記記録手段と上記送信手段との論理的な接続を行うための制御データを含むことを特徴とする。

20

【0022】

本発明の通信システムのその他の特徴とするところは、任意の制御データを送信する制御ノードと、上記制御データを受信する送信ノードおよび受信ノードとがシリアルデータを送受信するためのシリアルバスを介して接続された通信システムであって、上記送信ノードは、ファイルシステムを具備するとともに、当該ファイルシステムを上記シリアルバスを介してアシンクロナス転送により上記受信ノードに提供し、上記制御データは、単一のデータ送信により上記ファイルシステムとの論理的な接続を行うための制御データを含むことを特徴とする。

【0023】

本発明の端末装置は、シリアルデータを送受信可能なシリアルバスを介して、データを受信する受信ノードに接続された端末装置であって、上記シリアルバスを介してデータを送信するための送信手段と、上記送信手段にデータを出力可能な出力手段を有するとともに、制御データに基づいて制御可能な所定の機能を有する機能動作手段と、上記送信手段と上記出力手段との接続状態を、単一の制御命令を用いて送信された制御データに基づいて制御可能な通信路制御手段とを備えることを特徴とする。

30

【0024】

本発明の端末装置の他の特徴とするところは、シリアルデータを送受信可能なシリアルバスを介して、データを送信する送信ノードに接続された端末装置であって、上記シリアルバスを介してデータを受信するための受信手段と、上記受信手段で受信したデータを入力可能な入力手段を有するとともに、制御データに基づいて制御可能な所定の機能を有する機能動作手段と、上記受信手段と上記入力手段との接続状態を、単一の制御命令を用いて送信された制御データに基づいて制御可能な通信路制御手段とを備えることを特徴とする。

40

【0025】

本発明の通信システムの制御方法は、シリアルデータを送受信可能なシリアルバスを介して、データを送信する送信ノードと、上記送信ノードから送信されたデータを受信する受信ノードとが接続された通信システムの制御方法であって、上記送信ノードが備える、所定の機能を有する機能動作手段からデータを出力するための出力手段と、上記シリアルバスを介してデータを外部に送信するための送信手段との接続状態を、単一の制御命令を

50

用いて送信された制御データに基づいて制御することを特徴とする。

【0026】

本発明の通信システムの制御方法の他の特徴とするところは、シリアルデータを送受信可能なシリアルバスを介して、データを送信する送信ノードと、上記送信ノードから送信されたデータを受信する受信ノードとが接続された通信システムの制御方法であって、上記受信ノードが備える、所定の機能を有する機能動作手段にデータを入力するための入力手段と、上記シリアルバスを介してデータを外部から受信するための受信手段との接続状態を、単一の制御命令を用いて送信された制御データに基づいて制御することを特徴とする。

【0027】

本発明のプログラムは、シリアルデータを送受信可能なシリアルバスを介してデータを受信する受信ノードに接続された送信ノードが備える、所定の機能を有する機能動作手段からデータを出力するための出力手段と、上記シリアルバスを介してデータを外部に送信するための送信手段との接続状態の制御を、単一の制御命令を用いて送信された制御データに基づいてコンピュータに実行させることを特徴とする。

【0028】

本発明のプログラムの他の特徴とするところは、シリアルデータを送受信可能なシリアルバスを介してデータを送信する送信ノードに接続された受信ノードが備える、所定の機能を有する機能動作手段にデータを入力するための入力手段と、上記シリアルバスを介してデータを外部から受信するための受信手段との接続状態の制御を、単一の制御命令を用いて送信された制御データに基づいてコンピュータに実行させることを特徴とする。

【0029】

本発明のコンピュータ読み取り可能な記録媒体は、シリアルデータを送受信可能なシリアルバスを介してデータを受信する受信ノードに接続された送信ノードが備える、所定の機能を有する機能動作手段からデータを出力するための出力手段と、上記シリアルバスを介してデータを外部に送信するための送信手段との接続状態の制御を、単一の制御命令を用いて送信された制御データに基づいてコンピュータに実行させるためのプログラムを記録したことを特徴とする。

【0030】

本発明のコンピュータ読み取り可能な記録媒体の他の特徴とするところは、シリアルデータを送受信可能なシリアルバスを介してデータを送信する送信ノードに接続された受信ノードが備える、所定の機能を有する機能動作手段にデータを入力するための入力手段と、上記シリアルバスを介してデータを外部から受信するための受信手段との接続状態の制御を、単一の制御命令を用いて送信された制御データに基づいてコンピュータに実行させるためのプログラムを記録したことを特徴とする。

【発明の効果】

【0031】

本発明によれば、単一のデータ送信によりノード内部の接続を制御する制御データを含む制御データを送信するようにしたので、単一のデータ送信で機器間の接続および機器内部での接続を同時に制御することができ、簡単な接続手順で、高速なデータ転送が可能であるとともにデータ転送の確実性が保証できる通信システムを実現することができる。

【0032】

また、単一のデータ送信により機器間の接続および機器内部での接続に関し排他制御を行うための制御データを送信するようにしたので、機器間の接続および機器内部での接続の双方の制御に関して、効率良く排他制御を行うことができる。

【発明を実施するための最良の形態】

【0033】

以下、本発明の実施形態を図面に基づいて説明する。

図1は、本発明の一実施形態による通信システムの一構成例を示すブロック図であり、各機能ブロックの電氣的接続構成をも示している。図1において、100はプリンタ、2

10

20

30

40

50

00は電子スチルカメラであり、IEEE1394シリアルバスのケーブル(以下、「1394ケーブル」とも称す。)10を介して通信可能なように接続されている。

【0034】

各機器の構成について説明する。プリンタ100は、通信部101、通信補助部102、画像処理部103、着脱可能記録部104、表示部105、制御部106、操作部107、メモリ108および印刷出力部109により構成され、各ブロックは制御を行うためのコントロールバス110とデータを伝送するためのデータバス111とにより接続されている。

【0035】

通信部101は、IEEE1394規格に準拠した通信方式により1394ケーブル10を介して外部の機器(例えば、電子スチルカメラ200)等との間で通信を行うためのものである。通信補助部102は、IEEE1394規格準拠の通信方式にプリンタ100内部の情報を変換するためのものである。

【0036】

画像処理部103は、1394ケーブル10を介してプリンタ100に送られてきたデータや、着脱可能記録部104に記録されているデータを印刷出力できるように、上記データに画像処理を施す。着脱可能記録部104は、プリンタ100に限らず、別の機器に接続してデータを記録したり、読み出したりすることも可能であり、表示部105は、ユーザーがプリンタ100を操作する際に操作の補助になる情報を提示する。

【0037】

制御部106は、プリンタ100全体の制御を行う。操作部107は、ユーザーがプリンタ100を実際に操作するためのものであり、メモリ108は画像処理部103にて印刷出力用に画像処理が施されたデータを一時的に記録しておき、印刷出力部109は、画像処理が施されたデータを印刷出力する。

【0038】

電子スチルカメラ200は、通信部201、通信補助部202、画像処理部203、着脱可能記録部204、表示部205、制御部206、操作部207、メモリ208および撮像部209により構成され、各ブロックは制御を行うためのコントロールバス210とデータを伝送するためのデータバス211とにより接続されている。

【0039】

通信部201は、IEEE1394規格に準拠した通信方式により1394ケーブル10を介して外部の機器(例えば、プリンタ100)等との間で通信を行うためのものである。通信補助部202は、IEEE1394規格準拠の通信方式に電子スチルカメラ200内部の情報を変換するためのものである。

【0040】

画像処理部203は、撮像部209によって得られたデータに画像処理を施し、画像として認識できる状態にする。着脱可能記録部204は、電子スチルカメラ200に限らず、別の機器に接続してデータを記録したり、読み出したりすることが可能であり、表示部205は、ユーザーが電子スチルカメラ200を操作する際に操作の補助になる情報を提示する。

【0041】

制御部206は、電子スチルカメラ200全体の制御を行う。操作部207は、ユーザーが電子スチルカメラ200を実際に操作するためのものであり、メモリ208は画像処理が施されたデータを一時的に記録するためのものである。撮像部209は、被写体像を撮影し電気的信号として得るためのものである。

【0042】

次に、1394シリアルバスを介して電子スチルカメラ200からプリンタ100に画像データを転送し、プリンタ100にて印刷出力する際の各機器の動作について説明する。

【0043】

10

20

30

40

50

電子スチルカメラ 200 にて、ユーザーが表示部 205 で被写体像を確認しながら操作部 207 によって撮影を促す操作を行うと、制御部 206 は被写体像を撮影するように撮像部 209 を制御する。撮像部 209 により得られた画像データは、データバス 211 を介してメモリ 208 に一時的に記録される。さらに、メモリ 208 に記録された画像データは、画像処理部 203 にて画像処理が施された後、データバス 211 を介して着脱可能記録部 204 にファイルとして記録される。

【0044】

着脱可能記録部 204 に記録された画像データに係る画像は、ユーザーが操作部 207 により再生操作を行うことで表示部 205 に表示することも可能であるし、電子スチルカメラ 200 から着脱可能記録部 204 を切り離して PC 等に接続することで再生（表示）

10

【0045】

また、着脱可能記録部 204 に記録された画像データは、通信部 201 を介して 1394 シリアルバス経由でプリンタ 100 に送信して印刷出力することも可能である。この場合には、電子スチルカメラ 200 は、データ伝送を保証するために IEEE 1394 規格に準拠した通信方式で通信を行わなければならない。本実施形態では、AV/C Compatible Asynchronous Serial Bus Connections 規格 (1394 Trade Association)、および AV/C Commands for Management of Asynchronous Serial Bus Connections 規格 (1394 Trade Association) を利用してデータ転送を行う。以下、上記二つの転送方法（規格）を総称して「アシンクロナスコネクション (Asynchronous Connection)」と称す。通信部 201 は、上記アシンクロナスコネクションを行う手段を有しており、これによりデータ転送が保証される。

20

【0046】

プリンタ 100 に転送し印刷する画像の画像データは、電子スチルカメラ 200 の表示部 205 に表示される着脱可能記録部 204 に記録された画像データの中から操作部 207 を用いて選択される。選択された画像データは、1394 シリアルバス経由でプリンタ 100 に伝送されるが、画像データを伝送する前に、プリンタ 100 を制御するためのコマンドが電子スチルカメラ 200 からプリンタ 100 に送信される。

【0047】

送信されたコマンドは、プリンタ 100 の通信部 101 を介して制御部 106 に転送され、制御部 106 はプリンタ 100 をデータ受信待ち状態に制御する。その後、選択された画像データが電子スチルカメラ 200 の通信部 201 を介してアシンクロナスコネクションによりプリンタ 100 に伝送される。データ受信待ち状態のプリンタ 100 は、電子スチルカメラ 200 から送信されてくる画像データを受信すると、データバス 111 を介してメモリ 108 に一時的に記録する。メモリ 108 に記録された画像データは、画像処理部 103 にて印刷出力用に画像処理が施され、印刷出力部 109 によって出力される。

30

【0048】

上述した説明は、電子スチルカメラ 200 からプリンタ 100 に画像データを伝送して印刷するプッシュモデルである。

以下に、プリンタ 100 が電子スチルカメラ 200 に記録された画像を選択して、1394 シリアルバス経由で取得し印刷する際の動作について説明する。

40

【0049】

電子スチルカメラ 200 の通信補助部 202 は、着脱可能記録部 204 にファイルとして記録されている画像データに関するファイル情報を読み出し、所定のファイルシステムにマッピングを行う。通信補助部 202 により表現される上記ファイルシステムは、1394 シリアルバスを介して外部の機器から参照することができる。すなわち、1394 シリアルバスを介して接続された各機器は、通信補助部 202 を各機器に実装しておくことで、各機器における記録媒体の記録方式が異なっていたとしても 1394 シリアルバス上では同様のファイルシステムとして取り扱うことが可能になる。

【0050】

50

図2は、本実施形態における上記ファイルシステムの構成例を示す図である。なお、この図2において、図1に示したブロック等と同一の機能を有するブロック等には同一の符号を付し、重複する説明は省略する。

【0051】

電子スチルカメラ200にて撮影された画像が、上記図1に示した着脱可能記録部204に記録形式A2-1で記録されている。通信補助部202は、記録形式A2-1で記録されているファイル情報を、図2に示すような予め定められた形式の情報形態(ファイルフォーマット)に変換して、変換により得られたファイル情報2-5を内部メモリ2-6に一時的に記録しておく。

【0052】

そして、プリンタ100の操作部107にて所定の操作を行い、電子スチルカメラ200が有するファイル情報を取得するためのコマンドをプリンタ100から電子スチルカメラ200に送信する。これにより、プリンタ100は、通信補助部202の内部メモリ2-6に保持するファイル情報2-5を1394シリアルバス経由で取得することができる。また、仮に記録形式A2-1とは異なる記録形式B2-2、記録形式C2-3、もしくは記録形式D2-4で電子スチルカメラ200の着脱可能記録部204にファイル情報が記録されていたとしても、プリンタ100は同一のファイルフォーマットのファイル情報2-5として認識することができる。

【0053】

上述のようにして取得したファイル情報は、プリンタ100の表示部105に表示され、ユーザーは操作部107を用いて印刷したい画像を選択する。上記選択に応じて、プリンタ100は、選択された画像の画像データを取得するためのコマンドを電子スチルカメラ200に送信する。同時に、プリンタ100内部では、1394シリアルバス経由で画像データを受信し印刷するためにデータ受信待ち状態になる。

【0054】

電子スチルカメラ200は、上記画像データの取得を要求するコマンドを受信すると、アシンクロナスコネクションによりプリンタ100に対して画像データの伝送を行う。プリンタ100は、電子スチルカメラ200からの画像データを受信すると、当該画像データをメモリ108に一時的に記録し、画像処理部103にて印刷出力用に画像処理を施した後、印刷出力部109によって出力する。

【0055】

次に、上記図1に示した着脱可能記録部104、204のような記録デバイス(記録媒体)に記録されたファイルへのアクセス手段について説明する。

【0056】

<ファイルシステムのデータ構造>

本実施形態における記録デバイスでは、指定されたファイルにアクセスする際にファイルパス(file_path)を用いる。上記ファイルパスは、可変長のデータであり、例えば、ルートディレクトリからのフルパス名である。また、上記ファイルパスでは、例えば、ディレクトリのデリミタとして"/"の文字("5C₁₆") (下付の添え字"16"は16進数表記であることを示す。以下についても同様である。)を用いるとともに、ターミネータとしてNULL(ヌル)文字("00₁₆")を用いる。

【0057】

上記ファイルパスのデータ長は、ファイルパス長(file_path_length)により示され、当該ファイルパス長には、最後のNULL文字の長さは含まれない。

上記ファイルパスデータおよびファイルパス長データは、後述するファイルアクセスコマンドセットにて使用される。

【0058】

本実施形態における記録デバイスからファイルの情報を取得する側のデバイス(以下、「コントローラ」と称す。)は、ファイルの情報を取得するために、記録デバイスに対してファイルリストを要求する。上記ファイルリストは、上記ファイルリストを要求するコ

10

20

30

40

50

マンドのレスポンスとして、記録デバイスからコントローラに転送される。ここで、ファイルリストは、複数のディレクトリエントリと、各ディレクトリエントリが保持しているファイルあるいはディレクトリの情報とを含む。また、ファイルリストのディレクトリエントリは、例えばサイズが20バイトであり、図3に示すようなデータ構造を有する。

【0059】

図3は、本実施形態におけるディレクトリエントリのデータ構造の一例を示す図である。図3に示したディレクトリエントリの各フィールドは、例えば、DOS (disk operating system) - FAT (file allocation table) システムとして知られているデータ構造と同様のデータ構造を有している。しかしながら、上記図3に示したディレクトリエントリでは、DOS - FATシステムのようなカレントディレクトリおよび親ディレクトリ情報は保持していない。

10

【0060】

図3において、最初の11バイト(アドレスオフセット値“00₁₆”~“0A₁₆”)のフィールドは、名前(name)フィールド301および拡張子(extension)フィールド302であり、ファイルの名前を示す。上記2つのフィールド301、302により、例えば、上述したDOS - FATシステムにおけるファイル名8文字と拡張子3文字とを形成することができる。上記記録デバイスに保持されているファイルの名前または拡張子が、上記2つのフィールド長よりも小さい場合には、例えば、パディングバイト(padding byte)が詰められる。上記パディングバイトの値は、例えば“20₁₆”である。

【0061】

次の1バイト(アドレスオフセット値“0B₁₆”)のフィールドは、属性(attribute_byte)フィールド303であり、ファイルの属性情報を示す。属性フィールド303は、図4に示すようなビットフィールドを含む。

20

【0062】

図4は、属性フィールド303の構成を示す図であり、ディレクトリ(directory)ビット401は、当該ディレクトリエントリがサブディレクトリか否かを示す。例えば、ディレクトリビット401が“1”にセットされている時には、当該ディレクトリエントリはサブディレクトリである。また、リード・オンリ(read_only)ビット402は、当該ディレクトリエントリ内のファイルに対して読み出しのみが可能であり、書き込み/消去が不可能であるか否かを示す。例えば、リード・オンリビット402が“1”にセットされている時には、ファイルは読み出し専用である。なお、図4において、複数のイグノア(ignored)ビットは、当該記録デバイスがいかなる値を書き込んでも、コントローラに無視されるようになっている。

30

【0063】

図3に戻り、更新時間(modification_time)フィールド304(アドレスオフセット値“0C₁₆”、“0D₁₆”)、および更新日付(modification_date)フィールド305(アドレスオフセット値“0E₁₆”、“0F₁₆”)は、ファイルの最新更新日時を示す。また、ファイルサイズ(file_size)フィールド306(アドレスオフセット値“10₁₆”~“13₁₆”)は、ファイル長(単位はバイト)を示す。

【0064】

本実施形態においては、複数の記録メディアと、当該記録メディア上の複数の分割されたいわゆるパーティションとを取り扱うことができる。上記記録メディアを、例えば「物理ボリューム」と称すると共に、上記パーティションを、例えば「論理ボリューム」と称する。

40

【0065】

本実施形態では、上記物理ボリュームの指定を物理ボリューム番号(physical_volume_number)データにて行い、上記論理ボリュームの指定を論理ボリューム番号(logical_volume_number)データにて行う。コントローラは、上記物理ボリューム番号および論理ボリューム番号により記録デバイス上の特定エリアを指定することができる。上記物理ボリューム番号および論理ボリューム番号は、後述するファイルアクセスコマンドセットにて

50

使用される。

【0066】

上記物理ボリューム番号データは、例えば“00₁₆”から“FE₁₆”の値をとり、最初の物理ボリューム番号は“00₁₆”の値をとる。“FF₁₆”の値は、例えば特別であり、上記物理ボリューム番号値は後述するファイルアクセスコマンドセットに応じて意味付けが異なる。

【0067】

上記論理ボリューム番号データは、例えば“00₁₆”から“FE₁₆”の値をとり、最初の論理ボリューム番号は“00₁₆”の値をとる。“FF₁₆”の値は、例えば特別であり、上記論理ボリューム番号値は後述するファイルアクセスコマンドセットに応じて意味付けが異なる。

10

【0068】

なお、本実施形態では、上記物理ボリューム番号および論理ボリューム番号の値が“FF₁₆”の場合を例外としているが、“00₁₆”から“FF₁₆”の値をとるように構成しても良い。

【0069】

本実施形態の記録デバイスでは、記録メディアの交換を通知するのにメディア・カウント(media_generation_count)を用い、記録メディアが記録デバイスに挿入された時に、メディア・カウントの値が“1”だけ増加される。なお、記録デバイスが複数の物理ボリュームをサポートしている場合には、各物理ボリュームに対して固有のメディア・カウントを有する。

20

【0070】

電源投入等により記録デバイスが初期化された時には、上記メディア・カウントは初期化される。なお、メディア・カウントの初期値は、記録デバイスに固有の値であり、例えば、零であっても良く、乱数であっても良い。また、記録デバイスが、電源切断時等にフラッシュメモリやその他の不揮発性メモリ等にメディア・カウントの値を記憶させることができる場合には、次の初期化の際の初期値として記憶したメディア・カウントの値を使用するようにしても良い。本実施形態では、上記メディア・カウントは1バイトのデータであり、バスリセットが発生した場合にはメディア・カウントの値は保持されるようになっている。

30

【0071】

コントローラは、記録デバイスに記録メディアが挿入され、メディア・カウントの値が増加したことを検出すると、上述したファイルリストを再取得する。本実施形態における記録デバイスでは、記録メディアの抜去によりメディア・カウントの値が増加することがないので、記録メディアの抜去により新たなファイルリストの取得を行う必要がなく、不要なトランザクションを発生させることがない。

【0072】

<ファイルアクセスコマンドセット>

次に、本実施形態のファイルシステムに関するコマンドについて説明する。本実施形態では、コマンドを用いることで、ファイル属性の取得やファイルの転送などのファイルに関する様々な制御を行うことが可能である。

40

本実施形態のファイルシステムに関するコマンドは、例えば、IEEE1394規格で定義されているFCP(Function Control Protocol)を用いている。

【0073】

以下、図5を参照してFCPについて説明する。

図5は、FCPを説明するための概念図である。FCPは、1394シリアルバスを介して接続されたデバイスを制御するためにIEEE1394規格に基づいて設計されており、種々のコマンドセットとコマンドトランザクションとが、FCP上で利用可能である。また、FCPではコマンドおよびレスポンスを送信する際に、IEEE1394規格のアシクロナスパケット(Asynchronous packet)が用いられる。

50

【 0 0 7 4 】

F C Pにおいて、他の（複数の）ノードを制御するノードを“コントローラ”と呼び、制御されるノードを“ターゲット”と呼ぶ。図5においては、500はコントローラ、510はターゲットとして動作するようになっている。

【 0 0 7 5 】

また、F C Pにおいては、コントローラからターゲットに送られるF C Pフレームを“コマンドフレーム”と呼び、ターゲットからコントローラに送られるF C Pフレームを“レスポンスフレーム”と呼ぶ。上記コマンドフレームを受け取る準備をしたレジスタを“コマンドレジスタ”と呼び、上記レスポンスフレームを受け取る準備をしたレジスタを“レスポンスレジスタ”と呼ぶ。図5においては、501および502は、コントローラ500上のコマンドレジスタおよびレスポンスレジスタであり、511および512は、ターゲット510上のコマンドレジスタおよびレスポンスレジスタである。

10

【 0 0 7 6 】

F C Pフレームを用いて送出されるアシンクロナスパケットのデータ構造を図6に示す。

図6において、601はデスティネーションIDフィールド、602はトランザクション・ラベル（tl）フィールド、603はリトライコード（rt）フィールド、604はトランザクション・コード（tcode）フィールド、605はプライオリティ（pri）フィールド、606はソースIDフィールド、607はデスティネーションオフセットフィールド、608はデータ長フィールド、609は拡張トランザクション・コードフィールド、610はヘッダCRC（header_CRC）フィールド、611はデータフィールド（F C Pフレーム）、612はデータCRC（data_CRC）フィールドである。上記アシンクロナスパケットは、例えば4バイト（32ビット、以下「クアドレッド」と称す。）を単位とするデータパケットである。

20

【 0 0 7 7 】

図6に示したアシンクロナスパケットにおいて、最初の16ビットのフィールドが、デスティネーションIDフィールド601であり、受信側（送信先）のノードIDを示す。次の6ビットのフィールドが、トランザクション・ラベルフィールド602であり、各トランザクション固有のタグである。次の2ビットのフィールドが、リトライコードフィールド603であり、パケットがリトライを試みるか否かを指定する。

30

【 0 0 7 8 】

次の4ビットのフィールドが、トランザクション・コードフィールド604であり、パケットのフォーマットや実行しなければならないトランザクションのタイプを指定する。本実施形態では、例えば、このフィールドの値が“0001₂”（下付の添え字“2”は2進数表記であることを示す。以下についても同様である。）である、データブロックの書き込みリクエストのトランザクションを用いる。

【 0 0 7 9 】

次の4ビットのフィールドが、プライオリティフィールド605であり、優先順位を指定する。本実施形態ではアシンクロナスパケットを用いているので、このフィールドの値は、例えば“0000₂”である。

40

【 0 0 8 0 】

次の16ビットのフィールドが、ソースIDフィールド606であり、送信側（送信元）のノードIDを示す。次の48ビットのフィールドが、デスティネーションオフセットフィールド607であり、パケットの受信側ノードアドレスの下位48ビットがこのフィールドによって指定される。

【 0 0 8 1 】

次の16ビットのフィールドが、データ長フィールド608であり、後述するデータフィールド611の長さをバイト単位で示す。次の16ビットのフィールドが、拡張トランザクション・コードフィールド609であり、本実施形態にて用いられるデータブロックの書き込みリクエストのトランザクションにおいては、この値は、例えば“0000₁₆”であ

50

る。

【0082】

次の32ビットのフィールドが、ヘッダCRCフィールド610であり、パケットヘッダ(上述したデスティネーションIDフィールド601から拡張トランザクション・コードフィールド609まで)のエラー検出に用いられる。

【0083】

次の可変長のフィールドが、データフィールド611であり、後述するCTS(Command/Transaction Set)にて用いられるコマンドフレームおよびレスポンスフレームが詰められる。上記データフィールド611を「ペイロード」と称する。本実施形態では、上記データフィールド611にてクアドレットの倍数に満たないビットには、“0”の値が詰められるようになっている。つまり、上記データ長フィールド608に格納されるデータ長がバイト(8ビット)単位で示されるとき、当該データ長フィールド608の値が“4”の倍数でない場合には、上記データフィールドは、クアドレットを満たすまで“00₁₆”の値のデータによって埋められる。

10

【0084】

最後の32ビットのフィールドが、データCRCフィールド612であり、上述したヘッダCRCフィールド610と同様に、上記データフィールド611のエラー検出に用いられる。

【0085】

FCPにおいて、データブロックの書き込みリクエスト(“Write request for data block packet”)のペイロードであるデータフィールド611は、“FCPフレーム”と呼ばれ、コマンドフレームは、ターゲット上のコマンドレジスタに書き込まれるとともに、レスポンスフレームは、コントローラ上のレスポンスレジスタに書き込まれる。ここで、コマンドレジスタとレスポンスレジスタとは切り離されており、これらのレジスタのデスティネーションオフセットアドレスは、図7に示すようにFCPで規定されている。

20

【0086】

本実施形態においては、デスティネーションオフセットアドレスとして、“FFFF F000 0B00₁₆”、および“FFFF F000 0D00₁₆”を含むライトトランザクション(Write transaction)のみが許されている。

【0087】

CTSは、コマンドセット、コマンドフィールドとレスポンスフィールドとの構造、およびコマンドとレスポンスとを送出する際に用いられるトランザクションの規則を指定したFCPフレームの一つのコンポーネントである。

30

【0088】

CTSで用いられるFCPフレームのデータ構造を図8に示す。

図8において、801はCTSフィールドである。CTSフィールド801は、4ビットのフィールドであり、CTSフィールド801の符号化の一例を図9に示す。本実施形態において、CTSフィールド801の値は、例えば“0000₂”を用いる。

【0089】

図10は、本実施形態におけるコマンドフレームおよびレスポンスフレームのデータ構造を示す図である。

40

図10(a)は、コマンドフレームのデータ構造を示しており、図10(a)において、1001はコマンドタイプ(ctype)フィールド、1002はサブユニットタイプ(subunit_type)フィールド、1003はサブユニットID(subunit_ID)フィールド、1004はオペコード(opcode)である。また、図10(a)において、オペコード1004以降は、1バイト毎に(n+1)個(nは整数)のオペランド(operand[0]、operand[1]、...、operand[n])が続く。

【0090】

4ビットからなるコマンドタイプフィールド1001は、コマンドのタイプを示す。

図11は、コマンドタイプフィールド1001の値とコマンドタイプとの関係の一例を

50

示す図である。

【0091】

図11に示すように、コマンドタイプフィールド1001の値が“0000₂”（コマンドタイプが“CONTROL”）のコマンドフレームは、コントローラが、ターゲットの制御を行うために用いる。制御内容は、後述するオペコードやオペランドによって指定される。また、コマンドタイプフィールド1001の値が“0001₂”（コマンドタイプが“STATUS”）のコマンドフレームは、コントローラが、ターゲットの現在の状態を問い合わせるために用いる。状態の指定は、後述するオペコードやオペランドによって行う。

【0092】

また、上記コマンドタイプフィールド1001の値が“0011₂”（コマンドタイプが“NOTIFY”）のコマンドフレームは、コントローラが、ターゲットの状態が変化したことをターゲットから通知させるために用いる。後述するオペコードやオペランドによって状態の指定を行うことは、上述した“STATUS”コマンドと同様である。

【0093】

さらに、コマンドタイプフィールド1001の値が“0010₂”あるいは“0100₂”（コマンドタイプが“SPECIFIC INQUIRY”あるいは“GENERAL INQUIRY”）のコマンドフレームは、同じオペコードを有する“CONTROL”コマンドが、ターゲットに実装されているか否かを確認するために用いる。ここで、“SPECIFIC INQUIRY”コマンドの場合には、オペコードと全てのオペランドとを指定しなければならないが、“GENERAL INQUIRY”コマンドの場合には、オペコードのみを指定する。これが、“SPECIFIC INQUIRY”コマンドと“GENERAL INQUIRY”コマンドとの相違点である。

【0094】

5ビットからなるサブユニットタイプフィールド1002と、3ビットからなるサブユニットIDフィールド1003とで、コマンドフレームが送られるサブユニット（subunit）を識別する。サブユニットは、AV/C Digital Interface Command Set General Specification（March 1998, 1394 Trade Association）規格（以下、「AV/Cコマンドセット規格」と称す。）等で定義されており、AVユニット（以下、単に「ユニット」とも称す。）の中で唯一つに識別されるとともに、首尾一貫した機能のセットを提供する仮想的なエン트리である。上記ユニットも、AV/Cコマンドセット規格にて同様に、定義されている。AVユニットは、1394シリアルバスに対して接続されているノードを有する電子デバイスを示す。

【0095】

上記AV/Cコマンドセット規格によれば、ユニットは、複数のサブユニットを有することができるようになってきている。そこで、サブユニットタイプフィールド1002とサブユニットIDフィールド1003とは、1394インターフェースに接続されるユニット中のサブユニットを識別するためのアドレスを示すようになってきている。サブユニットタイプフィールド1002の値とサブユニットのタイプとの関係の一例を図12に示す。

【0096】

上記サブユニットタイプフィールド1002とサブユニットIDフィールド1003とを総合して、「サブユニットアドレス」または「AV/Cアドレス」と称す。なお、上記サブユニットタイプフィールド1002の値が“1111₂”で、かつサブユニットIDフィールド1003の値が“0112”の場合には、サブユニットアドレスは、ユニットを示すようになってきている。

【0097】

本実施形態において、プリンタ100にコマンドフレームを送信する場合には、例えば、サブユニットタイプフィールド1002に“00010₂”の値を、サブユニットIDフィールド1003に“000₂”の値を指定する。また、本実施形態において、電子スチルカメラ200のカメラサブユニットにコマンドフレームを送信する場合には、例えば、サブユニットタイプフィールド1002に“00111₂”の値を、サブユニットIDフィールド1003に“000₂”の値を指定する。

【 0 0 9 8 】

さらに、本実施形態において、電子スチルカメラ 2 0 0 のファイルシステムを形成する通信補助部 2 0 2 は、図 1 2 におけるカメラ・ストレージ・サブユニット (Camera Storage Subunit) として動作する。このため、当該サブユニットにコマンドフレームを送信する場合には、例えば、サブユニットタイプフィールド 1 0 0 2 に “ 01011₂ ” の値を、サブユニット ID フィールド 1 0 0 3 に “ 000₂ ” の値を指定する。

【 0 0 9 9 】

オペコード 1 0 0 4 は、制御内容や後述するレスポンスフレームによって返される状態を定義する。オペコード 1 0 0 4 の後に続く、オペランドの数と意味付けは、上述したコマンドタイプ、サブユニットタイプ、あるいはオペコードに応じて異なる。

10

【 0 1 0 0 】

図 1 0 (b) は、レスポンスフレームのデータ構造を示す図である。図 1 0 (b) において、図 1 0 (a) に示したフィールドと同一の機能を有するフィールドには同一の符号を付し、重複する説明は省略する。図 1 0 (b) において、1 0 0 5 はレスポンス (response) フィールドであり、レスポンスのタイプを示す。

図 1 3 は、レスポンスフィールド 1 0 0 5 の値とレスポンスタイプとの関係の一例を示す図である。

【 0 1 0 1 】

本実施形態では、ターゲットとなるサブユニットは、コマンドタイプ、サブユニットアドレス、オペコード、およびオペランドにより構成されるコントローラから送出されたコマンドフレームに対して、適切なレスポンスフレームを発生させてコントローラに返すようになっている。上記レスポンスフレームは、受信したコマンドフレームに応じたレスポンスタイプ、サブユニットアドレス、オペコード、およびオペランドにより構成される。

20

【 0 1 0 2 】

次に、本実施形態におけるカメラ・ストレージ・サブユニットのコマンドについて説明する。図 1 4 は、カメラ・ストレージ・サブユニットのコマンドと、カメラ・ストレージ・サブユニットのオペコード値との関係の一例を示す図である。

【 0 1 0 3 】

カメラ・ストレージ・サブユニットのコマンドは、コントローラからコマンドフレームとして発行される時、およびレスポンスフレームとしてコントローラに返される時に、共通のヘッダ領域 (“ common frame header ”、以下、「共通フレームヘッダ」と称す。) を有する。制御コマンド (control command) (以下、単に「コマンド」とも称す。) における共通フレームヘッダのフォーマットを図 1 5 に示す。なお、図 1 5 においては、コマンドフレーム (command frame) における共通フレームヘッダのフォーマットを一例として示している。

30

【 0 1 0 4 】

図 1 5 において、オペコード (opcode) には、上記図 1 4 に示したオペコード値が入力される。また、レスポンスフレーム (response frame) におけるオペコードには、コマンドフレームのオペコードの値と同じ値が入力される。

【 0 1 0 5 】

第 0 のオペランド (operand[0]) は、1 バイトのサブファンクション (subfunction) フィールドであり、制御コマンドの動作モードを指定する。上記制御コマンドの動作モードとサブファンクションフィールド値との関係を図 1 6 に示す。カメラ・ストレージ・サブユニットが無効なサブファンクションフィールド値が入力されたコマンドフレームを受信した場合には、“ NOT IMPLEMENTED ” のレスポンス値 (レスポンスフィールドの値が “ 1 000₂ ”) を有するレスポンスフレームを返す。また、レスポンスフレームにおける第 0 のオペランドには、コマンドフレームの第 0 のオペランドの値と同じ値が入力される。

40

【 0 1 0 6 】

第 1 のオペランド (operand[1]) は、1 バイトのフィールドであり、固定値 “ FF₁₆ ” が入力される。また、レスポンスフレームにおける第 1 のオペランドには結果コード (resu

50

It code) が入力されて返される。上記結果コードの符号化の一例を図 17 に示す。ここで、カメラ・ストレージ・サブユニットが複数の理由により “REJECTED” レスポンスフレーム (レスポンスフィールドの値が “1010₂”) を返す場合には、図 17 に示した最も小さい値のデータが上記結果コードとして第 1 のオペランドに入力される。また、カメラ・ストレージ・サブユニットが “INTERIM” レスポンスフレーム (レスポンスフィールドの値が “1111₂”) を返す場合には、コントローラが発行した固定値と同じ “FF₁₆” が結果コードとして入力される。

【0107】

第 2 のオペランド (operand[2]) は、物理ボリューム番号フィールドであり、物理ボリュームの指定を行う。第 3 のオペランド (operand[3]) は、論理ボリューム番号フィールドであり、論理ボリュームの指定を行う。また、上述したようにコントローラがこれらのフィールドを用いないことを示す場合には、値 “FF₁₆” が入力される。

10

【0108】

ここで、コマンドフレーム内の物理ボリューム番号フィールドの値または論理ボリューム番号フィールドの値が無効な場合には、カメラ・ストレージ・サブユニットは “REJECTED” レスポンスフレームあるいは “NOT IMPLEMENTED” レスポンスフレームを返す。このとき本実施形態では、例えば、以下のように動作する。

【0109】

・コマンドフレームで指定された物理ボリューム番号 (物理ボリューム番号フィールド値) の記録メディアが存在しない場合には、カメラ・ストレージ・サブユニットは、結果コードとして “no media” (A0₁₆) の値が第 1 のオペランドに入力された “REJECTED” レスポンスフレームを返す。

20

【0110】

・コマンドフレームで指定された論理ボリューム番号 (論理ボリューム番号フィールド値) が物理ボリューム内に存在しない場合には、カメラ・ストレージ・サブユニットは、結果コードとして “invalid volume number” (91₁₆) の値が第 1 のオペランドに入力された “REJECTED” レスポンスフレームを返す。

【0111】

・カメラ・ストレージ・サブユニット内に存在しない物理ボリューム番号がコマンドフレームにより指定された場合には、カメラ・ストレージ・サブユニットは “NOT IMPLEMENTED” レスポンスフレームを返す。

30

【0112】

第 4 のオペランド (operand[4]) は、メディア・カウントフィールドであり、記録メディアの交換を通知するために用いられる。コントローラが上記図 14 に示したメディア情報 (“MEDIA INFO”) コマンドフレームおよびボリューム情報 (“VOLUME INFO”) コマンドフレーム以外のコマンドフレームを発行する場合には、コントローラが保持しているメディア・カウントの値が上記メディア・カウントフィールドに設定される。また、コントローラがメディア情報コマンドフレーム、あるいはボリューム情報コマンドフレームを発行する場合には、上記メディア・カウントフィールドに “FF₁₆” の値が設定される。

【0113】

カメラ・ストレージ・サブユニットは、上記メディア情報コマンドフレーム以外のコマンドフレームに対し、自らが保持しているメディア・カウント値を第 4 のオペランドに設定したレスポンスフレームを返す。このとき、受信したコマンドフレームがメディア情報コマンドフレーム、ボリューム情報コマンドフレーム以外のコマンドフレームであった場合には、カメラ・ストレージ・サブユニットは、自らが保持しているメディア・カウント値とコマンドフレーム内のメディア・カウント値とを比較する。その結果、メディア・カウント値が異なる場合には、カメラ・ストレージ・サブユニットは、結果コードとして “invalid generation count” (90₁₆) の値が第 1 のオペランドに入力された “REJECTED” レスポンスフレームを返す。

40

【0114】

50

また、通常は、コントローラがカメラ・ストレージ・サブユニットに任意の制御コマンドフレームを最初に送信する以前に、コントローラはボリューム情報コマンドフレームを発行して現在のメディア・カウント値を取得する。

【0115】

図18は、ファイルリストコマンド (FILE LIST control command) フレームのフォーマットを示す図である。

ファイルリストコマンドフレームにおいて、最初の6バイトのフィールドは、上述した共通フレームヘッダである。

【0116】

第5のオペランド (operand[5]) は、ファイルタイプ (file_type) フィールドであり、取得するディレクトリエントリのファイル型を指定する。ファイルタイプフィールド値の定義を図19に示す。ファイルタイプフィールド値が“00₁₆” (ファイルタイプ“any”) の場合には、返されるディレクトリエントリは任意のファイルあるいはディレクトリを含む。また、ファイルタイプフィールド値が“01₁₆” (ファイルタイプ“still image”) の場合には、返されるディレクトリエントリは静止画像ファイルのみを含む。

10

【0117】

第6のオペランド (operand[6]) の下位2ビットは、属性 (attribute) フィールドであり、取得するディレクトリエントリの型を指定する。属性フィールドの定義を図20に示す。

属性フィールドの下位ビット (“bit0”) が“1”に設定されている時には、カメラ・ストレージ・サブユニットはサブディレクトリを含んだディレクトリエントリを返す。一方、属性フィールドの下位ビットが“0”にクリアされている時には、カメラ・ストレージ・サブユニットはサブディレクトリを含んだディレクトリエントリを返さない。

20

【0118】

また、属性フィールドの上位ビット (“bit1”) が“1”に設定されている時には、カメラ・ストレージ・サブユニットはファイルを含んだディレクトリエントリを返す。一方、属性フィールドの上位ビットが“0”にクリアされている時には、カメラ・ストレージ・サブユニットはファイルを含んだディレクトリエントリを返さない。

さらに、属性フィールドの双方のビットがともに“1”に設定されている時には、カメラ・ストレージ・サブユニットはファイルとサブディレクトリとを含んだディレクトリエントリを返す。

30

【0119】

第7および第8のオペランド (operand[7]およびoperand[8]) は、開始番号 (start_number) フィールドを構成し、開始番号フィールドは、取得するディレクトリエントリの開始位置を指定する。開始番号フィールド値は、“0”から始まり、開始番号フィールド値により後述するページングパラメータを指定する。

【0120】

第9のオペランド (operand[9]) の最上位ビット (MSB) は、エンド・オブ・リスト (“eol” : end of list) ビットである。レスポンスフレーム内のエンド・オブ・リストビットは、レスポンスフレーム内の最後のディレクトリエントリが、コマンドフレームで指定されたディレクトリの中の最後のディレクトリエントリであるか否かを示す。

40

【0121】

具体的には、レスポンスフレームのエンド・オブ・リストビットの値が“0”にクリアされている場合には、レスポンスフレーム内の最後のディレクトリエントリが、コマンドフレームで指定されたディレクトリの最後のディレクトリエントリでないことを示す。一方、レスポンスフレームのエンド・オブ・リストビットの値が“1”に設定されている場合には、レスポンスフレーム内の最後のディレクトリエントリがコマンドフレームで指定されたディレクトリの最後のディレクトリエントリであることを示す。

【0122】

また、コントローラは、エンド・オブ・リストビットを“0”にクリアしてコマンドフ

50

レームを発行する。

【0123】

第9のオペランドの下位5ビットは、エン트리数(number_of_entries)フィールドであり、コマンドフレームでは取得するディレクトリエントリ数を指定し、レスポンスフレームではレスポンスフレーム内に実際に含まれるディレクトリエントリ数を示す。コマンドフレーム内のエン트리数フィールド値と、その応答としてのレスポンスフレーム内のエン트리数フィールド値とは一致しないことがあり得るが、レスポンスフレーム内のエン트리数フィールド値は、それに対応するコマンドフレーム内のエン트리数フィールド値以下になることが保証されている。

【0124】

上記開始番号フィールド、エンド・オブ・リストビット、およびエン트리数フィールドの値に基づいて、ページング機構が提供される。ここで、ページングは、カメラ・ストレージ・サブユニットがレスポンスフレームにより一度に返すことが可能なデータ長に制限があるため、複数のコマンド、レスポンスランザクションによりデータを授受する仕組みである。

【0125】

図21は、ページング動作を示すフローチャートである。なお、図21においては、コントローラにおけるファイルリスト取得時のページング動作の一例を示している。

図21において、ステップstep1にて、本実施形態におけるページング動作が開始されると、ステップstep2にて、コントローラは、上記開始番号の値を初期化する(例えば、開始番号の値に“0”を入力する)。

【0126】

次のステップstep3にて、コントローラは、上記エン트리数の値を初期化する。ここで、エン트리数の初期値は、コマンドフレームおよびレスポンスフレームの最大バイト数に応じて十分大きな値が入力される。例えば、コマンドフレームおよびレスポンスフレームの最大バイト数が256バイトの場合には、エン트리数の初期値として“128”等が入力される。

【0127】

次に、コントローラは、ステップstep4にて、ファイルリストコマンドフレームを送信し、ステップstep5にて、カメラ・ストレージ・サブユニットからのレスポンスフレームを受信する。コントローラは、レスポンスフレーム内の開始番号フィールド、エンド・オブ・リストビット、およびエン트리数フィールドの値をそれぞれ解析して、後に続く動作を決定する。

【0128】

まずステップstep6にて、エンド・オブ・リストビットが検査される。エンド・オブ・リストビットの値が“0”のときには、上述したようにディレクトリエントリが最後に達していないので、さらにディレクトリエントリを取得するために次のステップstep7に進む。ステップstep7では、開始番号フィールドの値と、レスポンスフレーム中のエン트리数フィールドの値とを加算して、新たな開始番号値を生成する。

【0129】

次のステップstep8にて、レスポンスフレームから抽出したエン트리数フィールドの値をコマンドフレーム中のエン트리数フィールドに入力する。上述したように、レスポンスフレーム内のエン트리数フィールドの値は、それに対応するコマンドフレーム内のエン트리数フィールドの値以下となることが保証されているとともに、上記ステップstep3ではコマンドフレーム中のエン트리数フィールドの値に十分に大きな値が入力されているので、ステップstep8にてエン트리数フィールドに入力される値はレスポンスフレームにて返すことができる最大エン트리数である確率が高い。

【0130】

上記ステップstep4～step8はループを構成しており、全てのディレクトリエントリを取得するまでステップstep4～step8が実行される。一方、ステップs

10

20

30

40

50

step 6にて、エンド・オブ・リストビットの値が“1”のときには、ディレクトリエントリが最後に達しているので、ステップstep 9に進み、ディレクトリエントリ取得動作を終了する。

上述したページング動作により、コントローラは着目したディレクトリに対して効率良くディレクトリエントリを取得することができる。

【0131】

図18に戻り、第10および第11のオペランド(operand[10]およびoperand[11])は、リザーブ(reserved)フィールドであり、当該フィールドにいかなる値を設定してもカメラ・ストレージ・サブユニットに無視される。コントローラは、当該フィールドに、例えば“FF₁₆”の値を設定してコマンドフレームを送信する。

10

【0132】

第12のオペランド(operand[12])は、リクエストパス長(request_path_length)フィールドであり、後に続くリクエストパス(request_path)フィールドのバイト長を示す。第13のオペランド(operand[13])以降は、可変長のリクエストパスフィールドであり、所望のディレクトリまたはファイルのパス名を指定する。上記リクエストパス長およびリクエストパスは、上述したファイルパス長およびファイルパスにそれぞれ相当する。

【0133】

上述したファイルリストコマンドフレームに対する応答としてのレスポンスフレームにおいては、ファイルリスト(file_list)フィールドが上述したリクエストパスフィールドの領域に設定される。ファイルリストフィールドは、可変長フィールドであり、上述したディレクトリエントリが保持されている。ファイルリストフィールドには複数のディレクトリエントリを含むことが可能であり、ファイルリストフィールドに含まれるディレクトリエントリ数は、上述したレスポンスフレーム内のエントリ数フィールドにより指示される。また、図3に示したように、各ディレクトリエントリのサイズは20バイトである。

20

【0134】

図22は、メディア情報コマンド(MEDIA INFO control command)フレームのフォーマットを示す図である。メディア情報コマンドは、カメラ・ストレージ・サブユニット内の記録メディアの情報を取得するために使用される。

メディア情報コマンドフレームにおいて、最初の6バイトのフィールドは、上述した共通フレームヘッダであり、物理ボリューム番号フィールド、論理ボリューム番号フィールド、およびメディア・カウントフィールドには固定値“FF₁₆”をそれぞれ指定する。

30

【0135】

図23は、メディア情報コマンドフレームに対するレスポンスフレームのフォーマットを示す図であり、最初の6バイトのフィールドは共通フレームヘッダに対する応答部分である。

第5のオペランド(operand[5])は、物理ボリューム数(number_of_physical_volume)フィールドであり、カメラ・ストレージ・サブユニット内の記録メディア数、すなわち物理ボリューム数が入力される。なお、記録メディアが着脱可能であって記録メディアが挿入されていない時にも、当該記録メディアが物理ボリューム数に加えられている。

40

【0136】

第6のオペランド(operand[6])以降は、論理ボリューム数(number_of_logical_volume)フィールドであり、物理ボリューム毎の論理ボリューム数が入力される。論理ボリューム数フィールドの総数は、上記物理ボリューム数フィールドの値に応じて異なり、物理ボリューム数フィールドの値がn(nは自然数)の場合には、第0の論理ボリューム数フィールド(number_of_logical_volume[0])から第(n-1)のボリューム数フィールド(number_of_logical_volume[n-1])までのn個の論理ボリューム数フィールドが存在する。なお、記録メディアが着脱可能であって記録メディアが挿入されていない時には、当該記録メディアに関連付けられている論理ボリューム数フィールドには“00₁₆”の値が入力される。

50

【 0 1 3 7 】

図 2 4 は、ファイル受信コマンド (RECEIVE FILE control command) フレームのフォーマットを示す図である。ファイル受信コマンドは、カメラ・ストレージ・サブユニット内の記録メディアにサブユニット・デスティネーションプラグ (subunit destination plug) から受信したファイルを記録するために使用される。

【 0 1 3 8 】

ファイル受信コマンドフレームにおいて、最初の 6 バイトのフィールドは、上述した共通フレームヘッダであり、物理ボリューム番号フィールドおよび論理ボリューム番号フィールドには記録される受信ファイルのボリュームを指定する。コントローラは、当該フィールドに“FF₁₆” (“any available volume”) の値を設定して、カメラ・ストレージ・サブユニットが選択した任意の記録可能な物理ボリュームおよび論理ボリュームにファイルを記録することができる。そして、カメラ・ストレージ・サブユニットは、記録時に選択した物理ボリューム番号および論理ボリューム番号を、上記物理ボリューム番号フィールドおよび論理ボリューム番号フィールドにそれぞれ設定してレスポンスフレームを返す。

10

【 0 1 3 9 】

なお、コントローラがコマンドフレームの物理ボリューム番号フィールドに“FF₁₆”の値を設定した場合には、カメラ・ストレージ・サブユニットは、コマンドフレーム内のメディア・カウントフィールドを無視するとともに、選択した物理ボリュームのメディア・カウント値をメディア・カウントフィールドに設定したレスポンスフレームを返す。

20

【 0 1 4 0 】

第 5 ~ 第 8 のオペランド (operand[5] ~ operand[8]) の 4 バイトは、受信サイズ (receive_size) フィールドであり、記録するファイルのバイト数を指定する。コントローラが、記録するファイルのバイト数を予め決定できない場合には、コントローラは受信サイズフィールドに零 (“00 00 00 00₁₆”) の値を設定して、カメラ・ストレージ・サブユニットにファイルを記録させることができる。

【 0 1 4 1 】

第 9 のオペランド (operand[9]) は、デスティネーションプラグ (destination_plug) フィールドであり、ファイルデータを入力するためのサブユニット・デスティネーションプラグを指定する。通常、コントローラは、ファイル受信コマンドフレームを発行するのに先立ち、カメラ・ストレージ・サブユニットのサブユニット・デスティネーションプラグと、ユニットプラグ (unit plug) または同じユニット内の他のサブユニットのサブユニット・ソースプラグ (subunit source plug) とのコネクションを確立する。

30

【 0 1 4 2 】

ここで、コントローラがデスティネーションプラグフィールドにて指定するサブユニット・デスティネーションプラグが既に使用されている場合には、カメラ・ストレージ・サブユニットは、結果コードとして“busy” (“80₁₆”) の値が第 1 のオペランドに入力された“REJECTED”レスポンスフレームを返す。

【 0 1 4 3 】

第 1 0 のオペランド (operand[10]) は、受信モード (receive_mode) フィールドであり、第 1 3 のオペランド (operand[13]) 以降のファイルパスフィールドで指定されたファイルパス名と同じファイルパス名のファイルが既に存在するときの動作を指定する。受信モードフィールドの符号化の一例を図 2 5 に示す。

40

【 0 1 4 4 】

第 1 1 のオペランド (operand[11]) は、ファイルタイプフィールドであり、受信されるファイルの型を指定する。ファイルタイプフィールドの符号化の一例を図 2 6 に示す。

【 0 1 4 5 】

ファイル受信コマンドフレームに対する応答としてのレスポンスフレームでは、例えば、上記受信サイズフィールド、デスティネーションプラグフィールド、受信モードフィールド、およびファイルタイプフィールドには、コマンドフレームと同じ値が入力されて返

50

される。

【 0 1 4 6 】

第 1 2 のオペランド (operand[12]) は、ファイルパス長フィールド、第 1 3 のオペランド以降は可変長のファイルパスフィールドであり、上述したファイルパス長およびファイルパスに相当する。上述したように記録された結果のファイルパス名は、上記ファイルパスフィールドに設定され、レスポンスフレームによりコントローラに返される。なお、ファイル受信コマンドフレームに対して “ REJECTED ” レスポンスフレームが返されるときは、ファイルパスフィールドには零の値が設定される。

【 0 1 4 7 】

図 2 7 は、ファイル送信コマンド (SEND FILE control command) フレームのフォーマットを示す図である。ファイル送信コマンドフレームは、ファイルデータの送信トリガとして使用される。 10

ファイル送信コマンドフレームにおいて、最初の 6 バイトの領域は、上述した共通フレームヘッダである。また、第 5 ~ 第 8 のオペランド (operand[5] ~ operand[8]) には固定値 “ FF₁₆ ” が設定される。

【 0 1 4 8 】

第 9 のオペランド (operand[9]) は、ソースプラグ (source_plug) フィールドであり、ファイルデータの送信に使用されるサブユニット・ソースプラグを指定する。通常、コントローラは、ファイル送信コマンドフレームを発行するのに先立ち、カメラ・ストレージ・サブユニットのサブユニット・ソースプラグと、ユニットプラグまたは同じユニット内の他のサブユニットのサブユニット・デスティネーションプラグとの接続を確立する。 20

【 0 1 4 9 】

ここで、カメラ・ストレージ・サブユニットがファイル送信コマンドフレームを受信したとき、当該コマンドフレームのソースプラグフィールドで指定されたサブユニット・ソースプラグが既に使用されている場合には、カメラ・ストレージ・サブユニットは、結果コードとして “ busy ” (“ 80₁₆ ”) の値が第 1 のオペランドに入力された “ REJECTED ” レスポンスフレームを返す。

【 0 1 5 0 】

また、ファイル送信コマンドで指定されたサブユニット・ソースプラグに接続されているユニットプラグまたは同じユニット内の他のサブユニットのサブユニット・デスティネーションプラグが指定されたファイルを受信できない場合には、カメラ・ストレージ・サブユニットは、結果コードとして “ invalid file type ” (“ A6₁₆ ”) の値が第 1 のオペランドに入力された “ REJECTED ” レスポンスフレームを返す。 30

【 0 1 5 1 】

上述したファイル受信コマンドフレームと同様に、第 1 2 のオペランド (operand[12]) はファイルパス長フィールド、第 1 3 のオペランド (operand[13]) 以降は可変長のファイルパスフィールドであり、上述したファイルパス長およびファイルパスに相当する。上述したように記録された結果のファイルパス名は、上記ファイルパスフィールドに設定され、レスポンスフレームによりコントローラに返される。なお、ファイル送信コマンドフレームに対して “ REJECTED ” レスポンスフレームが返されるときは、ファイルパスフィールドには零の値が設定される。 40

【 0 1 5 2 】

図 2 8 は、ボリューム情報コマンド (VOLUME INFO control command) フレームのフォーマットを示す図であり、図 2 9 は、ボリューム情報コマンドフレームに対するレスポンスフレームのフォーマットを示す図である。ボリューム情報コマンドは、指定されたボリュームの情報を取得するために使用される。

【 0 1 5 3 】

上記図 2 8 に示したボリューム情報コマンドフレームにおいて、最初の 6 バイトのフィールドは、上述した共通フレームヘッダであり、物理ボリューム番号フィールドおよび論 50

理ボリューム番号フィールドには、情報を取得する物理ボリュームおよび論理ボリュームをそれぞれ指定する。

【0154】

第4のオペランド (operand[4]) は、ボリューム情報コマンドフレームにおいては固定値 “FF₁₆” が設定され、レスポンスフレームにおいてはメディア・カウント値が設定される。

【0155】

第5のオペランド (operand[5]) は、ボリューム情報コマンドフレームにおいては固定値 “FF₁₆” が設定され、レスポンスフレームにおいては属性 (attributes) フィールドである。属性フィールドは、指定されたボリュームの状態を示す属性がビットアサインされる。

10

【0156】

属性フィールドのフォーマットを図30に示す。図30において、最上位ビットから6ビットのフィールドは、リザーブビットであり、当該領域に指定した値はカメラ・ストレージ・サブユニットにて無視される。最下位ビットから2ビット目のビット (“bit1”) は、書き込み禁止指示 (write_protected) ビットであり、指定されたボリュームが書き込み禁止の場合には “1” に設定される。最下位ビット (“bit0”) は、読み出し専用指示 (read_only_media) ビットであり、指定されたボリュームが読み出し専用の場合には、“1” に設定される。

【0157】

ボリューム情報コマンドフレームにおいて、第9～第16のオペランド (operand[9]～operand[16]) および第17～第24のオペランド (operand[17]～operand[24]) には、固定値 “FF FF FF FF FF FF FF₁₆” が設定される。

20

ボリューム情報コマンドフレームに対するレスポンスフレームにおいて、第9～第16のオペランド (operand[9]～operand[16]) は、最大容量 (maximum_capacity) フィールドであり、指定されたボリュームの最大容量がバイト単位で設定される。また、第17～第24のオペランド (operand[17]～operand[24]) は、空き容量 (free_space) フィールドであり、指定されたボリュームの現在の空き容量がバイト単位で設定される。

【0158】

<ファイル転送>

次に、コントローラとカメラ・ストレージ・サブユニットとの間での標準的な通信手順について説明する。図31は、コントローラとカメラ・ストレージ・サブユニットとの間でのコマンドの発行手順の概要を示す図である。

30

【0159】

まず、コントローラは、カメラ・ストレージ・サブユニットに対してメディア情報コマンドを発行して、カメラ・ストレージ・サブユニットが有する記録メディアのボリューム数の情報を取得する (S11)。コントローラは、メディア情報コマンドによりボリューム数の情報を取得した後、1つのボリュームを選択する。

【0160】

コントローラは、選択したボリュームにアクセスする前にボリューム情報コマンドを発行して、カメラ・ストレージ・サブユニットから現在のメディア・カウント値を取得する (S12)。当該ボリューム情報コマンドにより取得した値は、コントローラが選択したボリュームにアクセスするための各コマンドのコマンドフレームを発行する際に、メディア・カウントフィールドに設定される。

40

【0161】

その後、コントローラは、ファイルリストコマンドを発行して、カメラ・ストレージ・サブユニットからファイルリストを取得する (S13)。コントローラは、ファイルリストを取得した後、ファイル送信コマンドあるいはファイル受信コマンドを発行し、コントローラとカメラ・ストレージ・サブユニットとの間でのファイル転送を開始させる (S14)。

50

【0162】

次に、カメラ・ストレージ・サブユニットからのファイル（データ）転送手順について説明する。上記図31に示した通信手順により、ファイルリストを取得したコントローラは、ファイル送信コマンドを発行してデータ送信要求を行う。

このとき、コントローラは、ファイル送信コマンドの発行に先立って、カメラ・ストレージ・サブユニットのサブユニット・ソースプラグと、送信先（受信側）の装置であるデスティネーションとの間に、例えば上述したアシンクロナスコネクション（Asynchronous Connection）により、コネクションを確立する。なお、上記デスティネーションはコントローラであっても良いし、他の装置であっても良い。

【0163】

図32は、カメラ・ストレージ・サブユニットからファイルを転送させる際のコマンドフローの一例を示す図であり、縦方向（T軸方向）は概念的な時間経過を示している。

【0164】

図32において、例えばコントローラとカメラ・ストレージ・サブユニットとの間に、アシンクロナスコネクションによりコネクションを確立する（S21）。次に、コントローラは、カメラ・ストレージ・サブユニットに対してファイル送信コマンドを発行して（S22）、カメラ・ストレージ・サブユニットからのデータの転送（S23）を開始させる。本実施形態では、コントローラは、図32に示したように上記コネクションが確立されている間において、複数のファイル送信コマンドを発行して、複数のトランザクションを発生させることが可能である（S24、S25）。コントローラは、データ転送が終了すると上記ステップS21にて確立したコネクションを切断する（S26）。

【0165】

次に、カメラ・ストレージ・サブユニットにファイルを受信させる手順について説明する。図33は、カメラ・ストレージ・サブユニットにファイルを受信させる際のコマンドフローの一例を示す図であり、上記図32と同様に縦方向（T軸方向）は概念的な時間経過を示している。上記図31に示した通信手順により、ボリュームに関する情報を取得したコントローラは、ファイル受信コマンドを発行してファイルデータの受信要求を行う。

【0166】

このとき、コントローラは、ファイル受信コマンドの発行に先立って、カメラ・ストレージ・サブユニットのサブユニット・デスティネーションプラグと、送信元（送信側）の装置であるソースとの間にアシンクロナスコネクションによりコネクションを確立する（S31）。なお、上記ソースは、コントローラであっても良いし、他の装置であっても良く、例えば図33においては、コントローラをソースとしてコネクションを確立している。

【0167】

次に、コントローラは、カメラ・ストレージ・サブユニットに対してファイル受信コマンドを発行して（S32）、カメラ・ストレージ・サブユニットにファイルデータの受信（S33）を開始させる。図33に示したように上記コネクションが確立している間において、コントローラは、複数のファイル受信コマンドを発行して、複数のファイルを受信させることが可能である（S34、S35）。コントローラは、ファイルデータ転送が終了すると上記ステップS31にて確立したコネクションを切断する（S36）。

【0168】

上述したファイル転送の手順において、コネクションは確立されているが、コントローラからファイル転送のトリガになるコマンド、例えばファイル送信コマンドあるいはファイル受信コマンドが発行されていないにもかかわらずファイルデータが送受信された場合には、カメラ・ストレージ・サブユニットは当該ファイルデータを破棄する。

【0169】

以上説明したように、データの送信側であるプロデューサノード（以下、「プロデューサ」と称す。）と、データの受信側であるコンシューマノード（以下、「コンシューマ」と称す。）とのコネクションを管理するコントローラノード（以下、「コントローラ」と

10

20

30

40

50

称す。)が、プロデューサとコンシューマとを制御する場合には、コントローラは次の手順に従ってコマンドを発行する。

1)コントローラは、コネクションを確立するコマンドをコンシューマおよびプロデューサに発行する。

2)コントローラは、トリガコマンド(例えば、ファイル受信コマンド)をコンシューマに発行する。

3)コントローラは、コンシューマからのレスポンスを受け取った後、トリガコマンド(例えば、ファイル送信コマンド)をプロデューサに発行する。

【0170】

<アシンクロナスコネクション(Asynchronous Connection)>

上述したように本実施形態におけるファイルシステムでは、アシンクロナスコネクションにより、ファイルデータの転送およびファイル属性データの転送が行われる。アシンクロナスコネクションによりデータを送受信する通信システムは、例えば図34に示すように、データの送信側であるプロデューサ3401、データの受信側であるコンシューマ3402、およびプロデューサとコンシューマとのコネクションを管理するコントローラ3403から構成される。なお、図34においては、プロデューサ3401、コンシューマ3402、およびコントローラ3403をそれぞれ異なるノードとして示しているが、コントローラ3403を設けずに、プロデューサ3401がコントローラ3403の機能を有するようにしても良いし、コンシューマ3402がコントローラ3403の機能を有するようにしても良い。

【0171】

プロデューサ3401およびコンシューマ3402は、データの送受信を行うためのアシンクロナスプラグレジスタ(Asynchronous Plug Register)をそれぞれ所有する。上記アシンクロナスプラグレジスタは、図35に示すように構成されており、1つのアシンクロナス入力ポートレジスタ(iAPR:input Asynchronous Port Register)と、複数のアシンクロナス出力ポートレジスタ(oAPR:output Asynchronous Port Register)(図35に示した例においては、第1~第14のアシンクロナス出力ポートレジスタ(oAPR[1]~oAPR[14]))とで構成される。また、1つのアシンクロナス入力ポートレジスタおよび複数のアシンクロナス出力ポートレジスタは、IEEE1394インターフェースのアドレス空間にそれぞれ割り当てられる。

【0172】

ここで、プロデューサとしての機能のみを有するノードは、アシンクロナス出力ポートレジスタだけを備え、コンシューマとしての機能のみを有するノードは、アシンクロナス入力ポートレジスタとデータ受信用のセグメントバッファ(segment buffer)とを備える。また、プロデューサおよびコンシューマの双方の機能を有するノードは、アシンクロナス入力ポートレジスタとアシンクロナス出力ポートレジスタとセグメントバッファとを備える。

【0173】

ここで、プロデューサは、1つのコネクションで複数のコンシューマに対してデータを送信することができる場合には、そのデータ送信能力に応じて複数のアシンクロナス出力ポートレジスタを備え、1つのコンシューマにのみデータを送信することができない場合には、1つのアシンクロナス出力ポートレジスタだけを備えていれば良い。

【0174】

また、複数のコネクションを同時に形成することができるノードは、上記図35に示したアシンクロナスプラグレジスタを複数備える。このようにノードによっては複数のアシンクロナスプラグを有する場合もあり、各アシンクロナスプラグはプラグID(plug_id)により識別される。

【0175】

例えば、プロデューサ3401がライトトランザクションによりコンシューマ3402が備えるセグメントバッファに画像データ等を書き込むとする。このとき、書き込まれる

10

20

30

40

50

データをフレームと呼び、フレームのサイズがセグメントバッファのサイズより大きい場合には、複数回のセッションに分けて書き込みが行われる。この1回のセッションで書き込まれるデータをセグメントと呼ぶ。

【0176】

コントローラ3403は、コンシューマ3402のノードID、接続に使用するコンシューマ3402のプラグアドレス、プラグ番号等の情報をプロデューサ3401に対して送信する。同様に、コントローラ3403は、プロデューサ3401のノードID、接続に使用するプロデューサ3401のプラグアドレス、プラグ番号等の情報をコンシューマ3402に対して送信する。コントローラ3403からの情報を受信したプロデューサ3401およびコンシューマ3402は、受信した情報に基づいてデータの通信を開始し、プロデューサ3401からコンシューマ3402にデータが送信される。

10

【0177】

上述したコントローラ3403からプロデューサ3401あるいはコンシューマ3402に対しての通信、およびプロデューサ3401とコンシューマ3402との間でのデータの通信は、アシンクロナストランザクションを用いて行われる。このアシンクロナストランザクションに用いられるパケットフォーマットは、上記図6に示したデータ構造と同様である。

【0178】

上記アシンクロナス接続に用いられるコマンドフレームのフォーマットは、例えば図36に示すように構成されている。なお、コントローラ3403からプロデューサ3401あるいはコンシューマ3402に送信されるコマンドフレームと、プロデューサ3401あるいはコンシューマ3402からコントローラ3403に送信されるレスポンスフレームとは、同じフォーマットである。

20

【0179】

オペコード(opcode)3601は、コマンドの種類を示しており、アシンクロナス接続のコマンドフレームであることを示すコード値“ 26_{16} ”が設定される。

サブファンクションフィールド3602は、当該コマンドによりプロデューサやコンシューマが行う動作(プラグリソースの確保、接続の設定/解除等)を指定する。サブファンクションフィールドの符号化について図37に示す。

30

【0180】

なお、以下の説明では、アシンクロナス接続コマンド(ASYNCHRONOUS CONNECTION control command)フレームは、サブファンクションフィールドに基づいて呼称する。例えば、アシンクロナス接続コマンドフレームのサブファンクションフィールドの値が“ $E1_{16}$ ”(EX_ALLOCATE)の場合には、“EX_ALLOCATE”コマンドフレームと称する。

【0181】

ステータス(status)フィールド3603は、コマンド実行結果の状態を表すフィールドであり、レスポンスフレームにおいてデータが設定され、コントローラに状態が通知される。

40

【0182】

プラグID(plug_id)フィールド3604は、接続に使用するプラグ番号を指定する。プラグIDフィールドの符号化の一例を図38に示す。図38において、値“ BF_{16} ”は特別であり、コントローラが、カメラ・ストレージ・サブユニットのようなアシンクロナス接続をサポートしたデバイスに対して、現在使用していないプラグIDを用いて接続を確立するように指示するために使用される。

【0183】

プラグオフセット(plug_offset)フィールド3605は、42ビットのフィールドであり、プラグの先頭アドレス(図35に示したアシンクロナス入力ポートレジスタに相当するアドレス)が設定される。

50

【 0 1 8 4 】

ポートID (port_id) フィールド 3 6 0 6 は、プラグ内のどのポートが選択されているかを示す。ポートIDフィールドの符号化の一例を図 3 9 に示す。図 3 9 において、コンシューマポート (Consumer Port[0]) は、図 3 5 に示したアシンクロナス入力ポートレジスタ (iAPR) に対応し、第 1 ~ 第 1 4 のプロデューサポート (Producer Port[1] ~ Producer Port[14]) は、図 3 5 に示した第 1 ~ 第 1 4 のアシンクロナス出力ポート (oAPR[1] ~ oAPR[14]) にそれぞれ対応する。また、値 “0F₁₆” は特別であり、コントローラが、カメラ・ストレージ・サブユニットのようなアシンクロナスコネクションをサポートしたデバイスに対して、現在使用していないプロデューサポートを用いてコネクションを確立するように指示するために使用される。

10

【 0 1 8 5 】

したがって、コネクションに使用されるポートレジスタのオフセットアドレスは、例えば次式 (1) により計算される。

$$(\text{オフセットアドレス}) = (\text{plug_offset}) \quad 6 \mid (\text{port_id}) \quad 2 \quad \dots (1)$$

ここで、上記式 (1) において、“ ” は左ビットシフト演算を示し、“ | ” はビット毎の論理和 (OR) 演算を示す (以下についても同様である。)。

【 0 1 8 6 】

ポートビット (port_bits) フィールド 3 6 0 7 は、選択されているポートの能力を示す。ポートビットフィールドの符号化の一例を図 4 0 に示す。なお、図 4 0 において “ X ” は、考慮しないビット (Don't Care) であることを示す。

20

接続ノードID (connected_node_id) フィールド 3 6 0 8 は、コネクションを設定する相手のノード (以下、「接続ノード」と称す。) を示す。コントローラは、コンシューマに対しては接続されるプロデューサのノードIDを設定してコマンドフレームで通知し、プロデューサに対しては接続されるコンシューマのノードIDを設定してコマンドフレームで通知する。

【 0 1 8 7 】

接続プラグオフセット (connected_plug_offset) フィールド 3 6 0 9 は、接続ノードが有するプラグのアドレスを示す。

接続ポートID (connected_port_ID) フィールド 3 6 1 0 は、コネクションに使用する接続ノードのポート番号を示す。

30

【 0 1 8 8 】

したがって、接続ノードのポートレジスタのオフセットアドレスは、次式 (2) により計算される。

$$(\text{オフセットアドレス}) = (\text{connected_plug_offset}) \quad 6 \mid (\text{connected_port_id}) \quad 2 \quad \dots (2)$$

【 0 1 8 9 】

接続ポートビット (connected_port_bits) フィールド 3 6 1 1 は、接続ノードのポートの能力を示す。

接続プラグID (connected_plug_id) フィールド 3 6 1 2 は、コネクションに使用される接続ノードのプラグ番号を示す。

40

【 0 1 9 0 】

排他制御 (ex: exclusive) ビット 3 6 1 3 は、コントローラが、内部コネクションとユニットから他のユニットへの接続との双方の制御に関して、他のコントローラを排除することを指定するビットである。

【 0 1 9 1 】

具体的には、排他制御ビット 3 6 1 3 が “ 1 ” に設定されたアシンクロナスコネクションコマンドフレームを受け入れたターゲットノード (コントローラからのコマンドフレームを受信するノード) は、その後発行され、受信したアシンクロナスコネクションコマンドフレームを調べ、受信したアシンクロナスコネクションコマンドフレームを発行したコントローラのノードIDと、排他制御ビット 3 6 1 3 が “ 1 ” に設定されたアシンク

50

ナスコネクションコマンドフレームを発行したコントローラのノードIDとが一致するかどうかを検出する。その結果、ノードIDが不一致の場合には、ターゲットノードは受信したコマンドフレームに対して“REJECTED”レスポンスフレームを返す。

【0192】

また、排他制御ビット3613が“1”に設定されたアシンクロナスコネクションコマンドフレームは、次の場合にターゲットノードに受けつけられる。

- ・指定されたプラグがコンシューマのプラグの場合
- ・指定されたプロデューサのプラグが使用されていない場合

上述の場合には、ターゲットノードは、当該アシンクロナスコネクションコマンドフレームを発行したコントローラに対して、“ACCEPTED”レスポンスフレームを返す。

10

【0193】

コネクションカウント (connection_count) フィールド3614は、コンシューマのポートがいくつのプロデューサポートに接続されているかを示す。

ライトインターバル (write_interval) フィールド3615は、プロデューサがコンシューマのセグメントバッファにライトトランザクションを行う際のトランザクションの最小間隔を示す。

【0194】

リトライカウント (retry_count) フィールド3616は、シリアルバストランザクションに失敗した場合に必要なリトライ (再実行) 回数の値を示す。リトライカウントフィールドの値は、コンシューマからのレスポンスフレームで設定される。

20

サブユニットタイプ (subunit_type) フィールド3617とサブユニットID (subunit_id) フィールド3618とで、ソースまたはデスティネーションになるサブユニットを指定する。サブユニットタイプフィールド3617とサブユニットIDフィールド3618の定義は、上述したサブユニットアドレスと同様である。

サブユニットプラグ (subunit_plug) フィールド3619は、データを授受するための入出力プラグであるサブユニットプラグ番号を指定する。

【0195】

次に、上述したアシンクロナスコネクションにおけるコネクション確立手順について説明する。図41は、アシンクロナスコネクションにおけるコネクション確立手順を示す図である。

30

図41において、4100はプロデューサ、4101はプロデューサ4100が有する第1のサブユニット、4102はソースプラグ、4103はプロデューサポート、4110はコントローラ、4120はコンシューマ、4121はコンシューマ4120が有する第2のサブユニット、4122はデスティネーションプラグ、4123はコンシューマポートである。

【0196】

図41に示すように、本実施形態では、コントローラ4110は、まず、“EX_ALLOCATE”コマンド (EX_ALLOCATE control command) フレームをコンシューマ4120に対して発行する (1a)。このとき、図41に示した例では、コンシューマ4120内における第2のサブユニット4121のデスティネーションプラグ4122とコンシューマポート4123との内部コネクションが確立される。

40

【0197】

なお、デスティネーションプラグ4122とコンシューマポート4123との内部コネクションは、最初の“EX_ALLOCATE”コマンドフレームの発行後、後述する“EX_ATTACH”コマンド (EX_ATTACH control command) フレームが発行されるまでの間、当該内部コネクションが変更されないように排他的にロックされる。

【0198】

上記図37に示したように、“EX_ALLOCATE”コマンドフレームは、アシンクロナスコネクションにおけるコンシューマポートの取得要求、サブユニット・デスティネーションプラグの取得要求、およびコンシューマポートとサブユニット・デスティネーションプラ

50

グとの間の接続要求を行う。コンシューマ 4 1 2 0 は、サブユニットプラグ ID とコンシューマポートのアドレスとを、上記 “EX_ALLOCATE” コマンドフレームに対する “EX_ALLOCATE” レスポンスフレームとしてコントローラ 4 1 1 0 に返す (1 b)。

【 0 1 9 9 】

次に、コントローラ 4 1 1 0 は、“EX_ALLOCATE_ATTACH” コマンド (EX_ALLOCATE_ATTACH control command) フレームをプロデューサ 4 1 0 0 に対して発行し (2 a)、サブユニット・ソースプラグの取得要求、プロデューサポートの取得要求、およびサブユニット・ソースプラグとプロデューサポートとの接続要求を行う。

【 0 2 0 0 】

これにより、プロデューサ 4 1 0 0 内における第 1 のサブユニット 4 1 0 1 のソースプラグ 4 1 0 2 とプロデューサポート 4 1 0 3 との内部接続が確立される。また、プロデューサ 4 1 0 0 は、上記 “EX_ALLOCATE_ATTACH” コマンドフレームに対する “EX_ALLOCATE_ATTACH” レスポンスフレームをコントローラ 4 1 1 0 に返す (2 b)。

【 0 2 0 1 】

さらに、コントローラ 4 1 1 0 は、“EX_ATTACH” コマンド (EX_ATTACH control command) フレームをコンシューマ 4 1 2 0 に対して発行し (3 a)、プロデューサポート 4 1 0 3 とコンシューマポート 4 1 2 3 との接続要求を行う。コンシューマ 4 1 2 0 は、上記 “EX_ATTACH” コマンドフレームに対する “EX_ATTACH” レスポンスフレームをコントローラ 4 1 1 0 に返す (3 b)。

【 0 2 0 2 】

コンシューマ 4 1 2 0 は、プロデューサ 4 1 0 0 内のアシンクロナス出力ポートレジスタを更新して受信準備が終了したことを示す (4)。なお、プロデューサポート 4 1 0 3 は、コンシューマ 4 1 2 0 がプロデューサ 4 1 0 0 内のアシンクロナス出力ポートレジスタを更新するまでの間、インアクティブ状態 (非動作状態) に留まる。

【 0 2 0 3 】

上述のようにして接続の確立手順に従って動作することにより、本実施形態では、第 1 のサブユニット 4 1 0 1 と第 2 のサブユニット 4 1 2 1 との間でのアシンクロナス接続が確立される。

【 0 2 0 4 】

次に、上述したアシンクロナス接続における接続切断手順について説明する。図 4 2 は、アシンクロナス接続における接続切断手順を示す図である。なお、この図 4 2 において、図 4 1 に示したブロック等と同一の機能を有するブロック等には同一の符号を付し、重複する説明は省略する。

【 0 2 0 5 】

図 4 2 に示すように、まず、コントローラ 4 1 1 0 が “EX_DETACH” コマンド (EX_DETACH control command) フレームをコンシューマ 4 1 2 0 に対して発行し (1 a ')、接続切断手順が開始される。当該 “EX_DETACH” コマンドフレームは、接続カウンタ値を減少させる。

【 0 2 0 6 】

コンシューマ 4 1 2 0 が、上記 “EX_DETACH” コマンドフレームに対する応答として、コントローラ 4 1 1 0 に返す “EX_DETACH” レスポンスフレーム (1 b ') にて通知される接続カウンタ値が零であるときには、コンシューマ 4 1 2 0 が滞りなく接続切断手順に入ったことを示す。

【 0 2 0 7 】

一方、上記 “EX_DETACH” レスポンスフレームにて通知される接続カウンタ値が零でないときには、コンシューマ 4 1 2 0 は、まだ少なくとも 1 つの接続を有していることを示す。接続カウンタ値が零でない場合には、コントローラ 4 1 1 0 は、後述する “EX_DETACH_RELEASE” コマンド (EX_DETACH_RELEASE control command) フレームを発行しないようになっている。

【 0 2 0 8 】

上述のように、上記“EX_DETACH”レスポンスフレームにて通知される接続カウント値が零のとき、コンシューマ4120内の第2のサブユニット4121のデスティネーションプラグ4122とコンシューマポート4123とがインアクティブ状態になったままである。このインアクティブ状態のとき、コンシューマポート4123は、プロデューサ4100からのレジスタ更新、およびセグメントバッファへの書き込みを受け付ける。

【0209】

次に、コントローラ4110は、“EX_DETACH_RELEASE”コマンドフレームをプロデューサ4100に対して発行し、(2a')、プロデューサ4100内の第1のサブユニット4101のソースプラグ4102とプロデューサポート4103との接続が切断される。また、プロデューサ4100は、上記“EX_DETACH_RELEASE”コマンドフレームに対する“EX_DETACH_RELEASE”レスポンスフレームをコントローラ4110に返す(2b')。

10

【0210】

最後に、コントローラ4110は、“EX_RELEASE”コマンド(EX_RELEASE control command)フレームをコンシューマ4120に対して発行し(3a')、コンシューマ4120内における内部接続が切断される。コンシューマ4120は、上記“EX_RELEASE”コマンドフレームに対する“EX_RELEASE”レスポンスフレームをコントローラ4110に返す(3b')。

【0211】

上述のようにして接続の切断手順に従って動作することにより、本実施形態では、第1のサブユニット4101と第2のサブユニット4121との間でのアシンクロナス接続が切断される。

20

【0212】

以上、説明したように本実施形態によれば、アシンクロナス接続により接続を確立するための既存のコマンドに加え、ユニット(プロデューサやコンシューマ)内のサブユニットプラグ(サブユニット・ソースプラグ、サブユニット・デスティネーションプラグ)を指定するためのフィールドを拡張して追加する。これにより、ユニット間におけるアシンクロナス接続、およびユニット内でのサブユニットプラグとユニットが備えるポートとの間における接続を単一のコマンドにより、同時に接続あるいは切断することができ、簡単な接続手順で、高速なデータ転送が可能であるとともに、データ転送の確実性が保証できる通信システムを実現することができる。

30

【0213】

また、アシンクロナス接続における排他制御ビットを設けることで、当該排他制御ビットの値に応じて、ユニット内部での接続と、ユニットから他のユニットへの接続との双方の制御に関して他のコントローラを排除し、効率良く排他制御を行うことができる。

【0214】

なお、上述したアシンクロナス接続コマンドにおいては、AV/C Commands for Management of Asynchronous Serial Bus Connection規格と同じオペコード値(“26₁₆”)を用いているが、他の任意の値を用いても良い。

40

【0215】

[他の実施形態]

本発明は、単一の機器からなる装置にも適用可能であるとともに、例えば、ホストコンピュータ、インターフェース機器、リーダ、プリンタなどの複数の機器から構成されるシステムにも適用可能である。

また、上述した実施形態の機能を実現するべく各種のデバイスを動作させるように、該各種デバイスと接続された装置あるいはシステム内のコンピュータに対し、上記実施形態の機能を実現するためのソフトウェアのプログラムコードを供給し、そのシステムあるいは装置のコンピュータ(CPUあるいはMPU)に格納されたプログラムに従って上記各

50

種デバイスを動作させることによって実施したのも、本発明の範疇に含まれる。

【0216】

また、この場合、上記ソフトウェアのプログラムコード自体が上述した実施形態の機能を実現することになり、そのプログラムコード自体、およびそのプログラムコードをコンピュータに供給するための手段、例えばかかるプログラムコードを格納した記録媒体は本発明を構成する。かかるプログラムコードを記憶する記録媒体としては、例えばフレキシブルディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、磁気テープ、不揮発性のメモリカード、ROM（プログラムROM、EPROM、EEPROMなど）等を用いることができる。

【0217】

また、コンピュータが供給されたプログラムコードを実行することにより、上述の実施形態の機能が実現されるだけでなく、そのプログラムコードがコンピュータにおいて稼働しているOS（オペレーティングシステム）あるいは他のアプリケーションソフト等と共同して上述の実施形態の機能が実現される場合にもかかるプログラムコードは本発明の実施形態に含まれることは言うまでもない。

【0218】

さらに、供給されたプログラムコードがコンピュータの機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに格納された後、そのプログラムコードの指示に基づいてその機能拡張ボードや機能拡張ユニットに備わるCPU等が実際の処理の一部または全部を行い、その処理によって上述した実施形態の機能が実現される場合にも本発明に含まれることは言うまでもない。

【図面の簡単な説明】

【0219】

【図1】本発明の実施形態による通信システムの一構成例を示すブロック図である。

【図2】ファイルシステムの構成例を示す図である。

【図3】ディレクトリエントリのデータ構造の一例を示す図である。

【図4】属性フィールドの構成を示す図である。

【図5】FCPを説明するための概念図である。

【図6】アシンクロナスパケットのデータ構造を示す図である。

【図7】コマンドレジスタおよびレスポンスレジスタのデスティネーションオフセットアドレスを示す図である。

【図8】FCPフレームのデータ構造を示す図である。

【図9】CTSフィールドの符号化の一例を示す図である。

【図10】コマンドフレームおよびレスポンスフレームのデータ構造を示す図である。

【図11】コマンドタイプフィールドの値とコマンドタイプとの関係の一例を示す図である。

【図12】サブユニットタイプフィールドの値とサブユニットのタイプとの関係の一例を示す図である。

【図13】レスポンスフィールドの値とレスポンスタイプとの関係の一例を示す図である。

【図14】カメラ・ストレージ・サブユニットのコマンドと、カメラ・ストレージ・サブユニットのオペコード値との関係の一例を示す図である。

【図15】共通フレームヘッダのフォーマットを示す図である。

【図16】制御コマンドの動作モードとサブファンクションフィールド値との関係を示す図である。

【図17】結果コードの符号化の一例を示す図である。

【図18】ファイルリストコマンドフレームのフォーマットを示す図である。

【図19】ファイルタイプフィールド値の定義を示す図である。

【図20】属性フィールドの定義を示す図である。

【図21】ページング動作を示すフローチャートである。

10

20

30

40

50

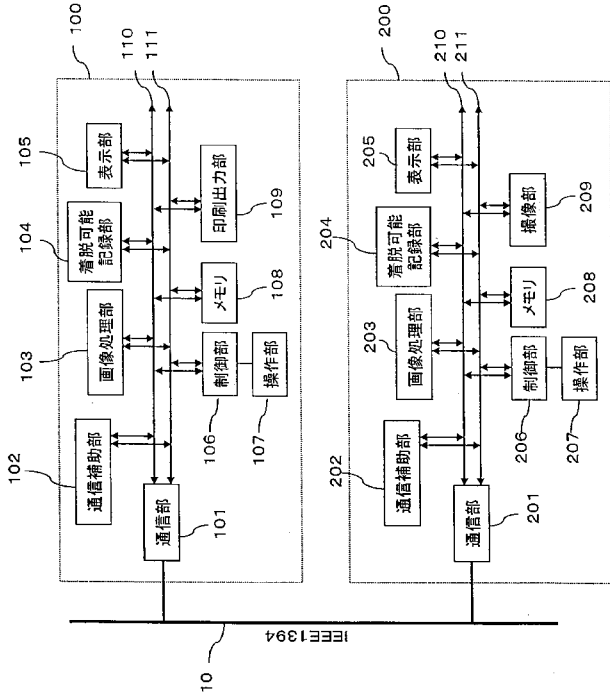
- 【図 2 2】メディア情報コマンドフレームのフォーマットを示す図である。
- 【図 2 3】メディア情報コマンドフレームに対するレスポンスフレームのフォーマットを示す図である。
- 【図 2 4】ファイル受信コマンドフレームのフォーマットを示す図である。
- 【図 2 5】受信モードフィールドの符号化の一例を示す図である。
- 【図 2 6】ファイルタイプフィールドの符号化の一例を示す図である。
- 【図 2 7】ファイル送信コマンドフレームのフォーマットを示す図である。
- 【図 2 8】ボリューム情報コマンドフレームのフォーマットを示す図である。
- 【図 2 9】ボリューム情報コマンドフレームに対するレスポンスフレームのフォーマットを示す図である。 10
- 【図 3 0】属性フィールドのフォーマットを示す図である。
- 【図 3 1】コントローラとカメラ・ストレージ・サブユニットとの間でのコマンドの発行手順の概要を示す図である。
- 【図 3 2】カメラ・ストレージ・サブユニットからファイルを転送させる際のコマンドフローの一例を示す図である。
- 【図 3 3】カメラ・ストレージ・サブユニットにファイルを受信させる際のコマンドフローの一例を示す図である。
- 【図 3 4】アシンクロナスコネクションによりデータを送受信する通信システムの構成例を示す図である。
- 【図 3 5】アシンクロナスプラグレジスタの構成例を示す図である。 20
- 【図 3 6】アシンクロナスコネクションに用いられるコマンドフレームのフォーマットを示す図である。
- 【図 3 7】サブファンクションフィールドの符号化の一例を示す図である。
- 【図 3 8】プラグIDフィールドの符号化の一例を示す図である。
- 【図 3 9】ポートIDフィールドの符号化の一例を示す図である。
- 【図 4 0】ポートビットフィールドの符号化の一例を示す図である。
- 【図 4 1】アシンクロナスコネクションにおけるコネクション確立手順を示す図である。
- 【図 4 2】アシンクロナスコネクションにおけるコネクション切断手順を示す図である。

【符号の説明】

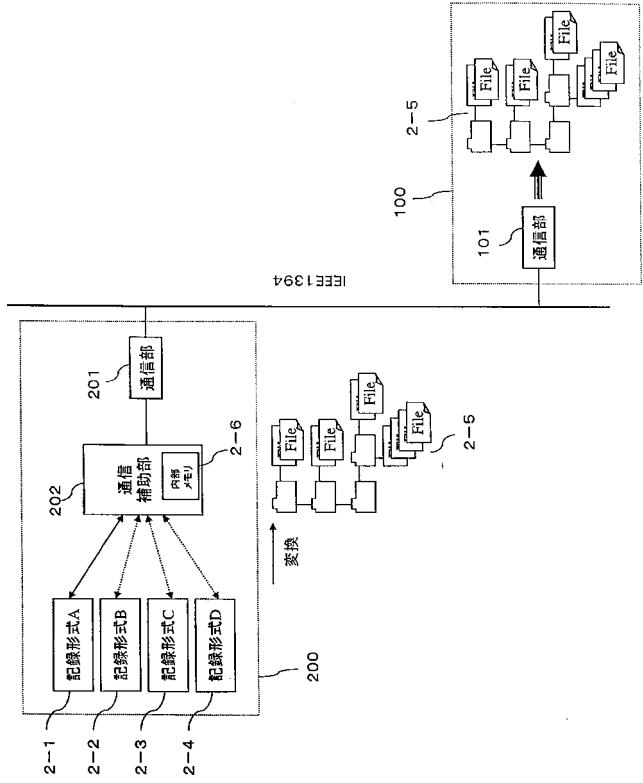
【0 2 2 0】 30

- 1 0 ケーブル
- 1 0 0 プリンタ
- 1 0 1、2 0 1 通信部
- 1 0 2、2 0 2 通信補助部
- 1 0 3、2 0 3 画像処理部
- 1 0 4、2 0 4 着脱可能記録部
- 1 0 5、2 0 5 表示部
- 1 0 6、2 0 6 制御部
- 1 0 7、2 0 7 操作部
- 1 0 8、2 0 8 メモリ 40
- 1 0 9 印刷出力部
- 1 1 0、2 1 0 コントロールバス
- 1 1 1、2 1 1 データバス
- 2 0 0 電子スチルカメラ
- 2 0 9 撮像部

【図1】



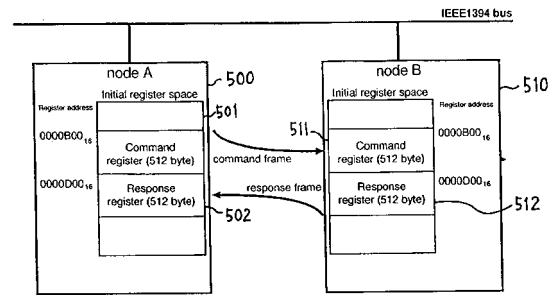
【図2】



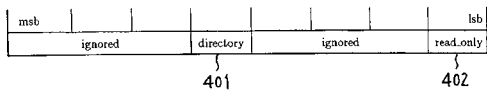
【図3】

address offset	contents
00 ₁₆	name
07 ₁₆	
08 ₁₆	extention
0A ₁₆	
0B ₁₆	attribute_byte
0C ₁₆	modification_time
0D ₁₆	
0E ₁₆	modification_date
0F ₁₆	
10 ₁₆	file_size
11 ₁₆	
12 ₁₆	
13 ₁₆	

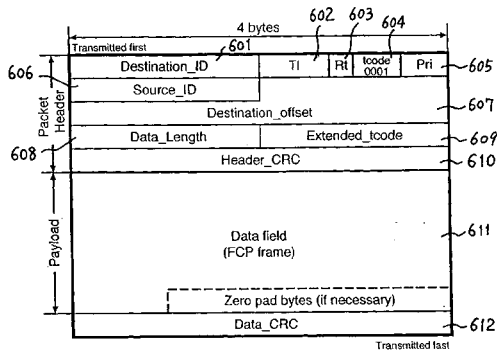
【図5】



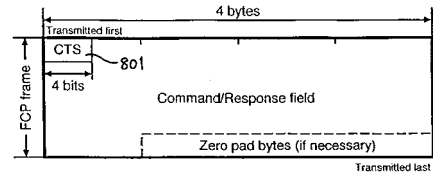
【図4】



【 図 6 】



【 図 8 】



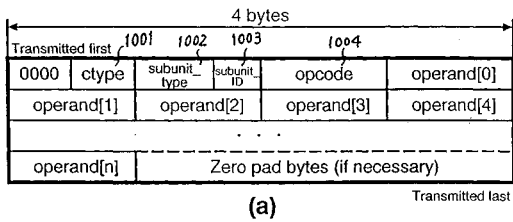
【 図 7 】

Offset base (top address of "Initial register space")	FFFF F000 0000 ₁₆
To address of command register (offset)	000 0B00 ₁₆
To address of response register (offset)	000 0D00 ₁₆

【 図 9 】

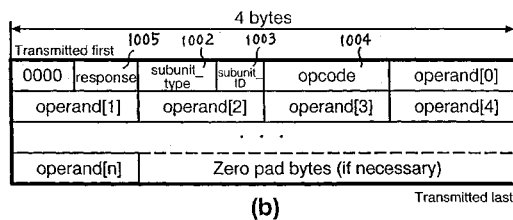
CTS code		CTS
msb	lsb	
0	0 0 0	AV/C
0	0 0 1	reserved for CAL
0	0 1 0	reserved for EHS
0	0 1 1	reserved
1	1 0 1	
1	1 1 0	Vendor Unique
1	1 1 1	Extended CTS

【 図 1 0 】



【 図 1 1 】

code (binary)	Command Type
MSB	LSB
0000	CONTROL
0001	STATUS
0010	SPECIFIC INQUIRY
0011	NOTIFY
0100	GENERAL INQUIRY
0101	(reserved)
0110	(reserved)
0111	(reserved)



【 図 1 2 】

code (binary)		Subunit Type
MSB	LSB	
0000		Video Monitor
0001		Audio
0010		Printer
0011		Disc Recorder/Player (audio or video)
0100		Tape Recorder/Player (audio or video)
0101		Tuner
0110		(reserved)
0111		Camera
1000		Changer
1001		Panel
1010		Bullitin Board
1011		Camera Storage
1100		(reserved)
1101		(reserved)
1110		Vendor unique
1111		(reserved)
1110		拡張 subunit.type
1111		Unit

【 図 1 3 】

code (binary)		Response
MSB	LSB	
1000		NOT IMPLEMENTED
1001		ACCEPTED
1010		REJECTED
1011		IN TRANSITION
1100		IMPLEMENTED/STABLE
1101		CHANGED
1110		(reserved)
1111		INTERIM

【 図 1 4 】

Name	Opecode Value	Comments
FILE LIST	43 ₁₆	ファイルリストを取得する
MEDIA INFO	40 ₁₆	記録メディアの情報を取得する
RECEIVE FILE	52 ₁₆	subunit デステイネーション plug からファイルを入力し、該ファイルを記録メディアに記録する
SEND FILE	50 ₁₆	指定されたファイルを subunit ソース plug へ出力する
VOLUME INFO	41 ₁₆	ボリューム情報を取得する

【 図 1 5 】

opcode	msb	lsb
opcode		
operand[0]		subfunction
operand[1]		FF ₁₆
operand[2]		physical.volume.number
operand[3]		logical.volume.number
operand[4]		media.generation.count

【 図 1 7 】

response	result	value	Comments
ACCEPTED	success	00 ₁₆	コマンドが成功した
STABLE	not excuting	01 ₁₆	コマンドが実行されていない
	excuting	03 ₁₆	コマンドは実行された
	suspended	04 ₁₆	コマンドはバスリセットから復帰した
	aborting	05 ₁₆	コマンドは停止中である
	REJECTED	busy	80 ₁₆
	aborted	81 ₁₆	コマンドは停止された
	retry	82 ₁₆	同じコマンドが実行中である
	no command	83 ₁₆	一致するコマンドがない
	disabled	84 ₁₆	コマンドの実行が許可されていない
	invalid generation count	90 ₁₆	無効な media.generation.count
	invalid volume number	91 ₁₆	無効なボリューム番号
	invalid subunit plug	92 ₁₆	無効な subunit plug 番号
	no media	A0 ₁₆	記録メディアがない
	no file	A1 ₁₆	指定されたファイルがない
	no drectory	A2 ₁₆	指定されたディレクトリがない
	no space	A3 ₁₆	空き容量がない
	file exist	A4 ₁₆	指定されたパスは既に使用済み
	unsupported format	A5 ₁₆	指定されたボリュームのフォーマットがサポートされていない
	invalid file type	A6 ₁₆	指定されたファイル型がサポートされていない
	volume protected	A7 ₁₆	指定されたボリュームは write プロテクトされている
	file protected	A8 ₁₆	指定されたファイルが読出し専用である
	no thumbnail	A9 ₁₆	サムネイルがない
	plug busy	B0 ₁₆	指定された plug が busy 状態である
	transport error	B1 ₁₆	データ転送でエラーが生じた
	unknown	FE ₁₆	その他のエラー

【 図 1 6 】

subfunction	value	meaning
execute	00 ₁₆	コントロールコマンドの実行
abort	01 ₁₆	実行中のコントロールコマンドの停止
resume	02 ₁₆	バスリセットからの復帰

【 図 18 】

	msb																		lsb
opcode	FILE LIST(43 ₁₆)																		
operand[0]	subfunction																		
operand[1]	FF ₁₆																		
operand[2]	physical.volume.number																		
operand[3]	logical.volume.number																		
operand[4]	media.generation.count																		
operand[5]	file.type																		
operand[6]	reserved														attribute				
operand[7]	start.number																		
operand[8]	start.number																		
operand[9]	eol	reserved			number.of.entries														
operand[10]	reserved																		
operand[11]	reserved																		
operand[12]	request.path.length																		
operand[13]	request.path																		
:																			
:																			

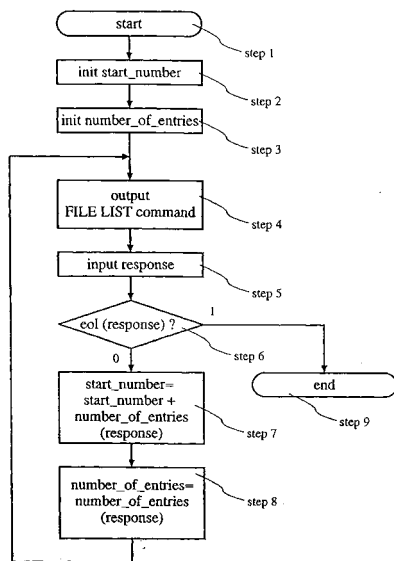
【 図 19 】

file.type	value	meaning
any	00 ₁₆	任意のファイル、及び、ディレクトリ
still image	01 ₁₆	静止画像ファイル

【 図 20 】

bit	description
bit 1	ファイル
bit 0 (lsb)	ディレクトリ

【 図 21 】



【 図 22 】

	msb																			lsb
opcode	MEDIA INFO(40 ₁₆)																			
operand[0]	subfunction																			
operand[1]	FF ₁₆																			
operand[2]	FF FF FF ₁₆																			
operand[3]	FF FF FF ₁₆																			
operand[4]	FF FF FF ₁₆																			

【 図 2 3 】

	msb								lsb
opcode	MEDIA INFO(40 ₁₆)								
operand[0]	subfunction								
operand[1]	result								
operand[2]	FFFF FF ₁₆ (same as command frame)								
operand[3]									
operand[4]									
operand[5]									
operand[5]	number.of.physical.volume{n}								
operand[6]	number.of.logical.volume[0]								
:	:								
operand[n+5]	number.of.logical.volume[n-1]								

【 図 2 4 】

	msb								lsb
opcode	RECEIVE FILE(52 ₁₆)								
operand[0]	subfunction								
operand[1]	FF ₁₆								
operand[2]	physical.volume.number								
operand[3]	logical.volume.number								
operand[4]	media.generation.count								
operand[5]	receive.size								
operand[6]									
operand[7]									
operand[8]									
operand[9]	destination.plug								
operand[10]	receive.mode								
operand[11]	file.type								
operand[12]	file.path.length								
operand[13]	file.path								
:									
:									

【 図 2 5 】

receive_mode	value	operation mode
protect	00 ₁₆	result code に "file exist" (A4 ₁₆) の値をセットした REJECTED response を返すコマンドを実行しない
over write	01 ₁₆	既存のファイルは受信ファイルで上書きされる
rename	02 ₁₆	受信ファイルは他のパス名を用いて記録される

【 図 2 7 】

	msb								lsb
opcode	SEND FILE(50 ₁₆)								
operand[0]	subfunction								
operand[1]	FF ₁₆								
operand[2]	physical.volume.number								
operand[3]	logical.volume.number								
operand[4]	media.generation.count								
operand[5]	FF FF FF FF ₁₆								
operand[6]									
operand[7]									
operand[8]									
operand[9]	source.plug								
operand[10]	reserved								
operand[11]	reserved								
operand[12]	file.path.length								
operand[13]	file.path								
:									
:									

【 図 2 6 】

file.type	value	operation mode
DCF	00 ₁₆	ファイルが JEIDA-49-2-1999, Design rule for Camera File System (DCF) 規格によって定められた命名規則に則ったファイル名を持つファイルであることを示す
unspecified	FF ₁₆	ファイルの型指定をしない

【 図 28 】

opcode	VOLUME INFO(41 ₁₆)
operand[0]	subfunction
operand[1]	FF ₁₆
operand[2]	physical_volume_number
operand[3]	logical_volume_number
operand[4]	FF ₁₆
operand[5]	FF ₁₆
operand[6]	reserved
operand[7]	
operand[8]	
operand[9]	FF FF FF FF FF FF FF FF ₁₆
:	
operand[16]	
operand[17]	FF FF FF FF FF FF FF FF ₁₆
:	
operand[24]	

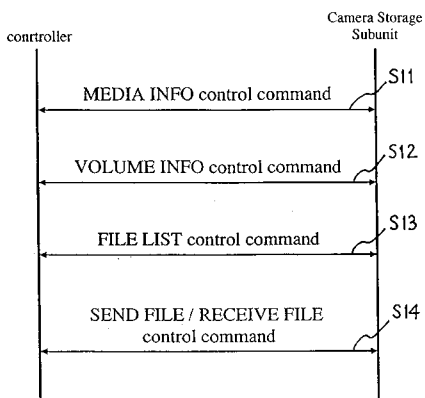
【 図 29 】

opcode	VOLUME INFO(41 ₁₆)
operand[0]	subfunction
operand[1]	result
operand[2]	physical_volume_number
operand[3]	logical_volume_number
operand[4]	media_generation_count
operand[5]	attributes
operand[6]	reserved (same as command frame)
operand[7]	
operand[8]	
operand[9]	maximum_capacity
:	
operand[16]	
operand[17]	free_space
:	
operand[24]	

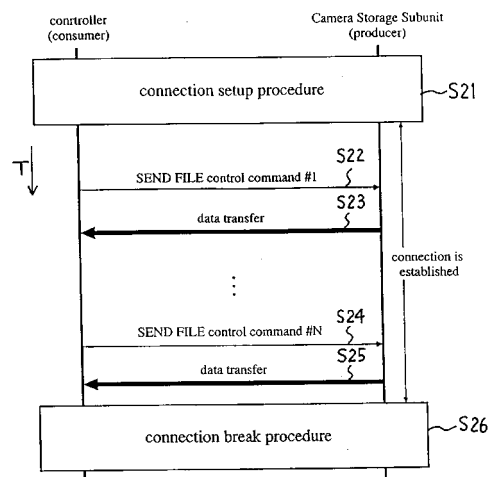
【 図 30 】

msb	reserved	write_protected	read_only_media	lsb
-----	----------	-----------------	-----------------	-----

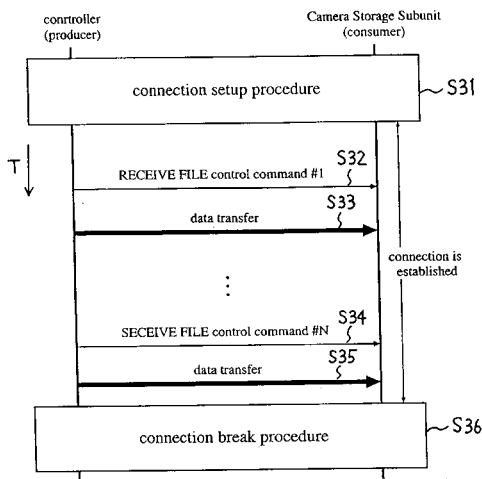
【 図 31 】



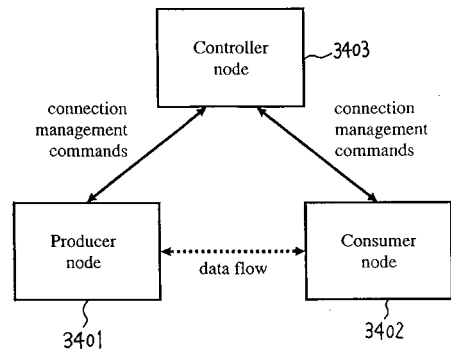
【 図 32 】



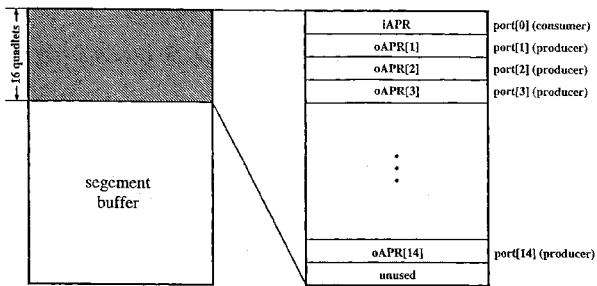
【 図 3 3 】



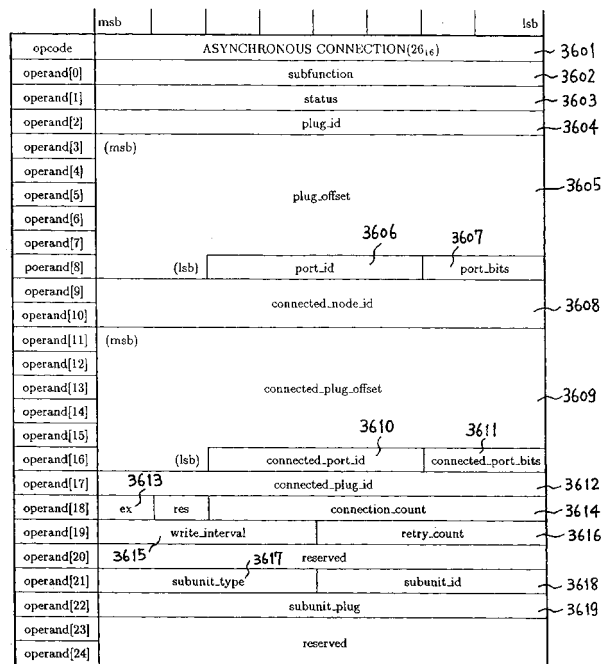
【 図 3 4 】



【 図 3 5 】



【 図 3 6 】



【 図 3 7 】

Symbol	value	meaning
EX.ALLOCATE	E1 ₁₆	コンシューマに対し、コンシューマ port と subunit plug との確保要求、及び、コンシューマ port と subunit plug との接続要求を行なう
EX.ATTACH	E2 ₁₆	コンシューマに対し、コンシューマ port とプロデューサ port との接続要求を行なう
EX.ALLOCATE.ATTACH	E3 ₁₆	プロデューサに対し、プロデューサ port と subunit plug との確保要求、及び、プロデューサ port と subunit plug との接続要求を行なうとともに、コンシューマがプロデューサ内の qAPR レジスタを更新するのを待機させる
EX.RELEASE	E5 ₁₆	コンシューマに対し、port と subunit plug との解放要求、及び、port と subunit plug 間の切断要求を行なう
EX.DETACH	E6 ₁₆	コンシューマに対し、コンシューマ port の切断要求を行なう
EX.DETACH.RELEASE	E7 ₁₆	プロデューサに対し、プロデューサ port の切断要求、プロデューサ port と subunit plug との間の切断要求、及び、プロデューサ port と subunit plug の解放要求を行なう

【 図 3 8 】

value	meaning
A0 ₁₆	Asynchronous Plug[0]
A1 ₁₆	Asynchronous Plug[1]
:	:
BE ₁₆	Asynchronous Plug[30]
BF ₁₆	Any available Asynchronous Plug

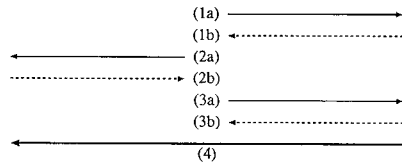
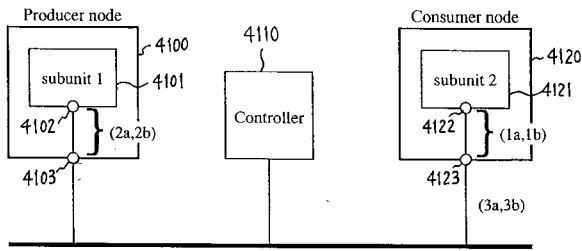
【 図 3 9 】

value	meaning
00 ₁₆	Consumer Port[0]
01 ₁₆	Producer Port[1]
:	:
0E ₁₆	Producer Port[14]
0F ₁₆	Any available Producer Port

【 図 4 0 】

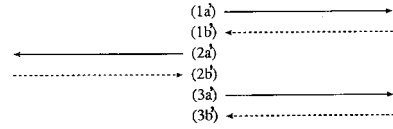
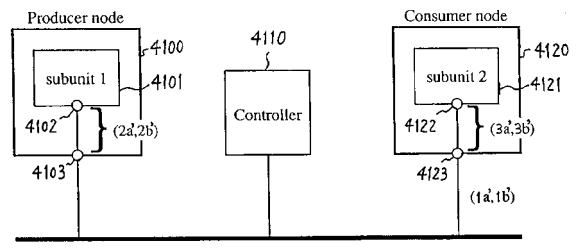
port.id value	port.bits value	symbol	meaning
00 ₁₆	X1 ₂	cf == 1	このコンシューマ port の segment buffer に対し、任意の順番での書き込みを許可する
	X0 ₂	cf == 0	このコンシューマ port の segment buffer に対して先頭アドレスからの順次書き込みのみを許可する
	1X ₂	m == 1	このコンシューマ port はマルチキャスト接続に対応
	0X ₂	m == 0	このコンシューマ port はマルチキャスト接続に非対応
01 ₁₆ ~ 0E ₁₆	00 ₂	FIXED	このコンシューマ port は固定サイズの segment buffer のみに対応
	01 ₂	FRAME	この port はフレーム毎の segment buffer のサイズ変更に対応
	10 ₂	SEGMENT	この port はセグメント毎の segment buffer のサイズ変更に対応

【 図 4 1 】



- (1a) EX_ALLOCATE command frame
- (1b) EX_ALLOCATE response frame
- (2a) EX_ALLOCATE_ATTACH command frame
- (2b) EX_ALLOCATE_ATTACH response frame
- (3a) EX_ATTACH command frame
- (3b) EX_ATTACH response frame, ++connection count == 1
- (4) Asynchronous Connection transport starts oAPR register update

【 図 4 2 】



- (1a) EX_DETACH command frame
- (1b) EX_DETACH response frame
- (2a) EX_DETACH_RELEASE command frame, --connection count == 0
- (2b) EX_DETACH_RELEASE response frame
- (3a) EX_RELEASE command frame
- (3b) EX_RELEASE response frame