

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4058156号
(P4058156)

(45) 発行日 平成20年3月5日(2008.3.5)

(24) 登録日 平成19年12月21日(2007.12.21)

(51) Int.Cl.		F I			
G06F 3/12	(2006.01)	G06F 3/12			D
G06F 13/00	(2006.01)	G06F 13/00		3 5 1 B	
G06F 13/38	(2006.01)	G06F 13/00		3 5 7 A	
		G06F 13/38		3 5 0	

請求項の数 15 (全 34 頁)

(21) 出願番号	特願平10-63665	(73) 特許権者	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
(22) 出願日	平成10年3月13日(1998.3.13)	(74) 代理人	100090273 弁理士 園分 孝悦
(65) 公開番号	特開平11-259259	(72) 発明者	鈴木 尚久 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
(43) 公開日	平成11年9月24日(1999.9.24)	(72) 発明者	小林 真琴 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
審査請求日	平成16年11月30日(2004.11.30)	(72) 発明者	中村 敦 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

最終頁に続く

(54) 【発明の名称】 データ処理方法、データ処理装置、プリンタ、及び記憶媒体

(57) 【特許請求の範囲】

【請求項1】

共通シリアルバスを介して複数種類のデバイスのプロトコルを切り換えるためのデータ処理方法であって、

上記デバイスのプロトコルの種類に係わらない初期プロトコルを実行し、

上記初期プロトコルに続いて、上記複数種類のデバイス固有のプロトコルを実行し、

上記初期プロトコルは、上記共通シリアルバスによる同一ネットワーク上に存在する第1の種類のデバイスについての当該共通シリアルバスを用いた接続形態を示すトポロジ接続状態を調査する第1の調査処理を含むと共に、上記共通シリアルバスによる同一ネットワーク上に存在し、上記第1の種類のデバイスに対してデータを出力する第2の種類のデバイスについての当該共通シリアルバスを用いた接続形態を示すトポロジ接続状態を調査する第2の調査処理を含むことを特徴とするデータ処理方法。

10

【請求項2】

上記第1及び第2の調査処理は各々、デバイスの機能を調査する第3の調査処理を含み、

上記初期プロトコルは、上記第1の調査処理で得られたトポロジ接続状態及び上記第3の調査処理で得られた複数のプロトコルに基づいて上記第1の種類のデバイスに対する実行すべきプロトコルを決定する処理と、上記第2の調査処理で得られたトポロジ接続状態及び上記第3の調査処理で得られた複数のプロトコルに基づいて上記第2の種類のデバイスに対する実行すべきプロトコルを決定する処理とを含むことを特徴とする請求項1に記載

20

載のデータ処理方法。

【請求項 3】

上記第 1 の調査処理は、デバイスとしてのプリンタの状態及び能力を含む情報を得る処理を含み、

上記第 2 の調査処理は、デバイスとしてのイメージ入力装置の状態及び能力を含む情報を得る処理を含むことを特徴とする請求項 1 に記載のデータ処理方法。

【請求項 4】

上記プリンタに固有のプロトコルの実行後に、プリントすべき画像データを上記イメージ入力装置から上記プリンタに対して伝送することを特徴とする請求項 3 に記載のデータ処理方法。

10

【請求項 5】

上記共通シリアルバスは、IEEE 1394 規格に適合したバス、または USB 規格に適合したバスであることを特徴とする請求項 1 に記載のデータ処理方法。

【請求項 6】

上記初期プロトコルは、OSI モデルのデータリンク層よりも上位の層で実行されるプロトコルであることを特徴とする請求項 1 に記載のデータ処理方法。

【請求項 7】

共通シリアルバスを介して複数種類のデバイスのプロトコルを切り換えるためのデータ処理装置であって、

上記デバイスのプロトコルの種類に係わらない初期プロトコルを実行する第 1 の実行手段と、

20

上記初期プロトコルに続いて、上記複数種類のデバイス固有のプロトコルを実行する第 2 の実行手段とを備え、

上記初期プロトコルは、上記共通シリアルバスによる同一ネットワーク上に存在する第 1 の種類のデバイスについての当該共通シリアルバスを用いた接続形態を示すトポロジ接続状態を調査する第 1 の調査処理を含むと共に、上記共通シリアルバスによる同一ネットワーク上に存在し、上記第 1 の種類のデバイスに対してデータを出力する第 2 の種類のデバイスについての当該共通シリアルバスを用いた接続形態を示すトポロジ接続状態を調査する第 2 の調査処理を含むことを特徴とするデータ処理装置。

【請求項 8】

30

上記第 1 及び第 2 の調査処理は各々、デバイスの機能を調査する第 3 の調査処理を含み、

上記初期プロトコルは、上記第 1 の調査処理で得られたトポロジ接続状態及び上記第 3 の調査処理で得られた複数のプロトコルに基づいて上記第 1 の種類のデバイスに対する実行すべきプロトコルを決定する処理と、上記第 2 の調査処理で得られたトポロジ接続状態及び上記第 3 の調査処理で得られた複数のプロトコルに基づいて上記第 2 の種類のデバイスに対する実行すべきプロトコルを決定する処理とを含むことを特徴とする請求項 7 に記載のデータ処理装置。

【請求項 9】

上記第 1 の調査処理は、デバイスとしてのプリンタの状態及び能力を含む情報を得る処理を含み、

40

上記第 2 の調査処理は、デバイスとしてのイメージ入力装置の状態及び能力を含む情報を得る処理を含むことを特徴とする請求項 7 に記載のデータ処理装置。

【請求項 10】

上記第 2 の実行手段は、上記プリンタに固有のプロトコルの実行後に、プリントすべき画像データを上記イメージ入力装置から上記プリンタに対して伝送することを特徴とする請求項 9 に記載のデータ処理装置。

【請求項 11】

共通シリアルバスを介して複数のプロトコルをホストデバイスに対応して切り換えるプリンタであって、

50

上記プロトコルの種類に係わらない初期プロトコルを実行する実行手段と、

上記初期プロトコルに続いて、自機器に固有のプロトコルを実行した後、印字データを受信する受信手段とを備え、

上記初期プロトコルは、上記共通シリアルバスによる同一ネットワーク上に存在する自機器についての当該共通シリアルバスを用いた接続形態を示すトポロジ接続状態を調査する第1の調査処理を含むと共に、上記共通シリアルバスによる同一ネットワーク上に存在し、自機器に対して画像データを出力するイメージ入力装置についての当該共通シリアルバスを用いた接続形態を示すトポロジ接続状態を調査する第2の調査処理を含むことを特徴とするプリンタ。

【請求項12】

共通シリアルバスを介して複数種類のデバイスのプロトコルを切り換えるデータ処理ステップをコンピュータに実行させるためのプログラムを格納するコンピュータ読み取り可能な記憶媒体であって、

上記データ処理ステップは、上記デバイスのプロトコルの種類に係わらない初期プロトコルを実行する第1の実行ステップと、上記初期プロトコルに続いて、上記複数種類のデバイスに固有のプロトコルを実行する第2の実行ステップとを含み、

上記初期プロトコルは、上記共通シリアルバスによる同一ネットワーク上に存在する第1の種類のデバイスについての当該共通シリアルバスを用いた接続形態を示すトポロジ接続状態を調査する第1の調査処理を含むと共に、上記共通シリアルバスによる同一ネットワーク上に存在し、上記第1の種類のデバイスに対してデータを出力する第2の種類のデバイスについての当該共通シリアルバスを用いた接続形態を示すトポロジ接続状態を調査する第2の調査処理を含むことを特徴とするコンピュータ読み取り可能な記憶媒体。

【請求項13】

共通シリアルバスを介して複数種類のデバイスのプロトコルを切り換えるためのデータ処理方法であって、

上記デバイスのプロトコルの種類に係わらない初期プロトコルを実行し、上記共通シリアルバスによる同一ネットワーク上に存在し、上記複数種類のうちの少なくとも1種のデバイスに対してデータを出力する種類のデバイスについての当該共通シリアルバスを用いた接続形態を示すトポロジ接続状態を調査する調査処理を含むことを特徴とするデータ処理方法。

【請求項14】

共通シリアルバスを介して複数種類のデバイスのプロトコルを切り換えるためのデータ処理装置であって、

上記デバイスのプロトコルの種類に係わらない初期プロトコルを実行し、上記共通シリアルバスによる同一ネットワーク上に存在し、上記複数種類のうちの少なくとも1種のデバイスに対してデータを出力する種類のデバイスについての当該共通シリアルバスを用いた接続形態を示すトポロジ接続状態を調査する調査処理手段を備えることを特徴とするデータ処理装置。

【請求項15】

共通シリアルバスを介して複数種類のデバイスのプロトコルを切り換えるデータ処理ステップをコンピュータに実行させるためのプログラムを格納するコンピュータ読み取り可能な記憶媒体であって、

上記データ処理ステップは、上記デバイスのプロトコルの種類に係わらない初期プロトコルを実行し、上記共通シリアルバスによる同一ネットワーク上に存在し、上記複数種類のうちの少なくとも1種のデバイスに対してデータを出力する種類のデバイスについての当該共通シリアルバスを用いた接続形態を示すトポロジ接続状態を調査する調査処理ステップを含むことを特徴とするコンピュータ読み取り可能な記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

10

20

30

40

50

本発明は、IEEE 1394 - 1995等のインターフェースを介して、複数種類のデバイスのプロトコルを切り換えるためのデータ処理方法、データ処理装置、プリンタ、及びそれらを実施するための処理ステップをコンピュータが読出可能に格納した記憶媒体に関するものである。

【0002】

【従来の技術】

従来より、汎用インターフェースを介してプリンタにデータを送出するシステムとして、様々な種類のシステムが知られている。

例えば、SCSI (Small Computer System Interface) や、セントロニクス等、一般に広く用いられるようになったデファクトスタンダードのインターフェースを用いて、コンピュータからプリンタにデータを出力する技術が知られている。

10

【0003】

【発明が解決しようとする課題】

しかしながら、汎用インターフェースを介して、あるプリンタにプリントデータを転送するためのプリンタプロトコルは、そのプリンタの固有のものに限られていた。このように、従来では、ホストデバイスとターゲットデバイス間でデータ通信を行う際の通信プロトコルが、そのターゲットデバイスによって限定され、拡張性に欠ける、という問題が生じていた。

特に、様々な種類の機器を接続するインターフェース、例えば、IEEE 1394のようなシリアルインターフェースを介してデータ転送する場合には、かかる拡張性に欠けると

20

【0004】

そこで、本発明は、上記の欠点を除去するために成されたもので、ホストデバイスとターゲットデバイス間でデータ通信を行う際の通信プロトコルが、そのターゲットデバイスによって限定されることがない、拡張性の高いデータ処理方法、データ処理装置、プリンタ、及び記憶媒体を提供することを目的とする。

また、本発明は、IEEE 1394規格等のシリアルインターフェースを用いた好適なデータ処理方法、データ処理装置、プリンタ、及び記憶媒体を提供することを目的とする。

また、本発明は、コンピュータを介することなく、ホストデバイスからターゲットデバイスへ直接画像データ等を転送するのに好適なデータ処理方法、データ処理装置、プリンタ、及び記憶媒体を提供することを目的とする。

30

また、本発明は、最適なプリンタ出力を得ることができるデータ処理方法、データ処理装置、プリンタ、及び記憶媒体を提供することを目的とする。

また、本発明は、通信プロトコルの切り換えによる負荷を下げるることができるデータ処理方法、データ処理装置、プリンタ、及び記憶媒体を提供することを目的とする。

【0005】

【課題を解決するための手段】

本発明のデータ処理方法は、共通シリアルバスを介して複数種類のデバイスのプロトコルを切り換えるためのデータ処理方法であって、上記デバイスのプロトコルの種類に係わらない初期プロトコルを実行し、上記初期プロトコルに続いて、上記複数種類のデバイス固有のプロトコルを実行し、上記初期プロトコルは、上記共通シリアルバスによる同一ネットワーク上に存在する第1の種類のデバイスについての当該共通シリアルバスを用いた接続形態を示すトポロジ接続状態を調査する第1の調査処理を含むと共に、上記共通シリアルバスによる同一ネットワーク上に存在し、上記第1の種類のデバイスに対してデータを出力する第2の種類のデバイスについての当該共通シリアルバスを用いた接続形態を示すトポロジ接続状態を調査する第2の調査処理を含む。

40

【0006】

本発明のデータ処理方法における他の態様は、上記第1及び第2の調査処理は各々、デバイスの機能を調査する第3の調査処理を含み、上記初期プロトコルは、上記第1の調査処理で得られたトポロジ接続状態及び上記第3の調査処理で得られた複数のプロトコルに

50

基づいて上記第1の種類 of デバイスに対する実行すべきプロトコルを決定する処理と、上記第2の調査処理で得られたトポロジ接続状態及び上記第3の調査処理で得られた複数のプロトコルに基づいて上記第2の種類 of デバイスに対する実行すべきプロトコルを決定する処理とを含む。

【0007】

また、本発明のデータ処理方法におけるその他の態様は、上記第1の調査処理は、デバイスとしてのプリンタの状態及び能力を含む情報を得る処理を含み、上記第2の調査処理は、デバイスとしてのイメージ入力装置の状態及び能力を含む情報を得る処理を含む。

【0008】

また、本発明のデータ処理方法におけるその他の態様は、上記プリンタに固有のプロトコルの実行後に、プリントすべき画像データを上記イメージ入力装置から上記プリンタに対して伝送する。

【0009】

また、本発明のデータ処理方法におけるその他の態様は、上記共通シリアルバスは、IEEE 1394規格に適合したバス、またはUSB規格に適合したバスである。

【0010】

また、本発明のデータ処理方法におけるその他の態様は、上記初期プロトコルは、OSIモデルのデータリンク層よりも上位の層で実行されるプロトコルである。

【0011】

また、本発明のデータ処理方法におけるその他の態様は、共通シリアルバスを介して複数種類のデバイスのプロトコルを切り換えるためのデータ処理方法であって、上記デバイスのプロトコルの種類に係わらない初期プロトコルを実行し、上記共通シリアルバスによる同一ネットワーク上に存在し、上記複数種類のうちの少なくとも1種のデバイスに対してデータを出力する種類のデバイスについての当該共通シリアルバスを用いた接続形態を示すトポロジ接続状態を調査する調査処理を含む。

【0013】

本発明のデータ処理装置は、共通シリアルバスを介して複数種類のデバイスのプロトコルを切り換えるためのデータ処理装置であって、上記デバイスのプロトコルの種類に係わらない初期プロトコルを実行する第1の実行手段と、上記初期プロトコルに続いて、上記複数種類のデバイス固有のプロトコルを実行する第2の実行手段とを備え、上記初期プロトコルは、上記共通シリアルバスによる同一ネットワーク上に存在する第1の種類 of デバイスについての当該共通シリアルバスを用いた接続形態を示すトポロジ接続状態を調査する第1の調査処理を含むと共に、上記共通シリアルバスによる同一ネットワーク上に存在し、上記第1の種類 of デバイスに対してデータを出力する第2の種類 of デバイスについての当該共通シリアルバスを用いた接続形態を示すトポロジ接続状態を調査する第2の調査処理を含む。

【0014】

本発明のデータ処理装置における他の態様は、上記第1及び第2の調査処理は各々、デバイスの機能を調査する第3の調査処理を含み、上記初期プロトコルは、上記第1の調査処理で得られたトポロジ接続状態及び上記第3の調査処理で得られた複数のプロトコルに基づいて上記第1の種類 of デバイスに対する実行すべきプロトコルを決定する処理と、上記第2の調査処理で得られたトポロジ接続状態及び上記第3の調査処理で得られた複数のプロトコルに基づいて上記第2の種類 of デバイスに対する実行すべきプロトコルを決定する処理とを含む。

【0015】

また、本発明のデータ処理装置におけるその他の態様は、上記第1の調査処理は、デバイスとしてのプリンタの状態及び能力を含む情報を得る処理を含み、上記第2の調査処理は、デバイスとしてのイメージ入力装置の状態及び能力を含む情報を得る処理を含む。

【0016】

また、本発明のデータ処理装置におけるその他の態様は、上記第2の実行手段は、上記

プリンタに固有のプロトコルの実行後に、プリントすべき画像データを上記イメージ入力装置から上記プリンタに対して伝送する。

【0017】

また、本発明のデータ処理装置におけるその他の態様は、共通シリアルバスを介して複数種類のデバイスのプロトコルを切り換えるためのデータ処理装置であって、上記デバイスのプロトコルの種類に係わらない初期プロトコルを実行し、上記共通シリアルバスによる同一ネットワーク上に存在し、上記複数種類のうちの少なくとも1種のデバイスに対してデータを出力する種類のデバイスについての当該共通シリアルバスを用いた接続形態を示すトポロジ接続状態を調査する調査処理手段を備える。

【0019】

本発明のデータ処理装置は、共通シリアルバスを介して複数のプロトコルをホストデバイスに対応して切り換えるプリンタであって、上記プロトコルの種類に係わらない初期プロトコルを実行する実行手段と、上記初期プロトコルに続いて、自機器に固有のプロトコルを実行した後、印字データを受信する受信手段とを備え、上記初期プロトコルは、上記共通シリアルバスによる同一ネットワーク上に存在する自機器についての当該共通シリアルバスを用いた接続形態を示すトポロジ接続状態を調査する第1の調査処理を含むと共に、上記共通シリアルバスによる同一ネットワーク上に存在し、自機器に対して画像データを出力するイメージ入力装置についての当該共通シリアルバスを用いた接続形態を示すトポロジ接続状態を調査する第2の調査処理を含む。

【0020】

本発明の記憶媒体は、共通シリアルバスを介して複数種類のデバイスのプロトコルを切り換えるデータ処理ステップをコンピュータに実行させるためのプログラムを格納するコンピュータ読み取り可能な記憶媒体であって、上記データ処理ステップは、上記デバイスのプロトコルの種類に係わらない初期プロトコルを実行する第1の実行ステップと、上記初期プロトコルに続いて、上記複数種類のデバイスに固有のプロトコルを実行する第2の実行ステップとを含み、上記初期プロトコルは、上記共通シリアルバスによる同一ネットワーク上に存在する第1の種類のデバイスについての当該共通シリアルバスを用いた接続形態を示すトポロジ接続状態を調査する第1の調査処理を含むと共に、上記共通シリアルバスによる同一ネットワーク上に存在し、上記第1の種類のデバイスに対してデータを出力する第2の種類のデバイスについての当該共通シリアルバスを用いた接続形態を示すトポロジ接続状態を調査する第2の調査処理を含む。

【0021】

また、本発明の記憶媒体におけるその他の態様は、共通シリアルバスを介して複数種類のデバイスのプロトコルを切り換えるデータ処理ステップをコンピュータに実行させるためのプログラムを格納するコンピュータ読み取り可能な記憶媒体であって、上記データ処理ステップは、上記デバイスのプロトコルの種類に係わらない初期プロトコルを実行し、上記共通シリアルバスによる同一ネットワーク上に存在し、上記複数種類のうちの少なくとも1種のデバイスに対してデータを出力する種類のデバイスについての当該共通シリアルバスを用いた接続形態を示すトポロジ接続状態を調査する調査処理ステップを含む。

【0061】

【発明の実施の形態】

以下、本発明の実施の形態について図面を用いて説明する。

【0062】

以下に説明する第1及び第2の実施の形態では、各機器間を接続するデジタルインターフェース(I/F)として、例えば、IEEE1394-1995(High Performance Serial Bus、以下、単に「1394シリアルバス」と言う)を用いているため、まず、1394シリアルバスについて予めその概要を説明する。

【0063】

[1394シリアルバスの概要]

【0064】

10

20

30

40

50

民生用デジタルビデオカムレコーダ（VCR）やデジタルビデオディスク（DVD）プレーヤの登場に伴って、ビデオデータやオーディオデータ（以下、これらをまとめて「AVデータ」と言う）等、リアルタイムで且つ高情報量のデータを転送する必要が生じている。このようなAVデータをリアルタイムでパーソナルコンピュータ（PC、以下、単に「パソコン」と言う）や、その他のデジタル機器に転送して取り込ませるには、高速なデータ転送が可能なインターフェースが必要となる。そういった観点から開発されたインターフェースが、この1394シリアルバスである。

【0065】

図1は、1394シリアルバスを用いて構成されるネットワーク・システムの一例を示した図である。

このシステムは、機器A、B、C、D、E、F、G、Hを備えており、A-B間、A-C間、B-D間、D-E間、C-F間、C-G間、及びC-H間が、各々1394シリアルバス用のツイスト・ペア・ケーブルで接続された構成としている。

これらの機器A～Hの一例としては、パソコン等のホストコンピュータ装置、及びコンピュータ周辺機器である。

尚、ここでのコンピュータ周辺機器としては、デジタルVCR、DVDプレーヤ、デジタルスチルカメラ、ハードディスクや光ディスク等の記録媒体を用いる記憶装置、CRTやLCD等のモニタ、チューナ、イメージスキャナ、フィルムスキャナ、プリンタ、MODEM、ターミナルアダプタ（TA）等、コンピュータ周辺機器の全てが対象になる。

【0066】

各機器間の接続は、ディジーチェーン方式とノード分岐方式との混在が可能であり、自由度の高い接続を行うことができるようになされている。また、各機器は各自固有のIDを有し、互いにIDを認識し合うことによって、1394シリアルバスで接続された範囲にて、1つのネットワークを構成している。

例えば、各機器間を各々1本の1394シリアルバス用ケーブルでディジーチェーン接続するだけで、各々の機器が中継の役割を担うため、全体として1つのネットワークを構成することができる。

【0067】

また、1394シリアルバスは、Plug and Play機能に対応しており、ケーブルを機器に接続するだけで自動的に機器を認識し、接続状況を認識する機能を有している。

このため、上記図1のシステムにおいて、ネットワークから任意の機器が外されたり、或いは、新たに加えられたとき等には、自動的にバスがリセット、すなわちそれまでのネットワークの構成情報がリセットされ、新たなネットワークが再構築される。このような機能によって、その時々々のネットワークの構成を常時設定、認識することができる。

【0068】

1394シリアルバスのデータ転送速度は、100/200/400Mbpsが定義されており、上位の転送速度を持つ機器が下位の転送速度をサポートすることで、互換性を保つようになされている。

【0069】

データ転送モードとしては、コントロール信号等の非同期データを転送するAsynchronous転送モード（ATM）と、リアルタイムなAVデータの同期データを転送するIsochronous転送モードがある。

これらの転送モードにより、非同期データ及び同期データは、各サイクル（通常125μS/サイクル）の中で、サイクル開始を示すサイクル・スタート・パケット（CSP）の転送に続き、同期データの転送を優先しつつ、サイクル内で混在して転送される。

【0070】

図2は、1394シリアルバスの構成要素を示す図である。

この1394シリアルバスは、レイヤ構造で構成されている。上記図2に示すように、1394シリアルバス用のケーブル813の先端のコネクタには、1394コネクタ・ポート810が接続される。

10

20

30

40

50

【 0 0 7 1 】

1 3 9 4 コネクタ・ポート 8 1 0 の上位には、フィジカルレイヤ 8 1 1 及びリンクレイヤ 8 1 2 を含むハードウェア部 8 0 0 が位置づけられている。

ハードウェア部 8 0 0 は、インターフェース用チップで構成され、そのうちフィジカルレイヤ 8 1 1 は、符号化やコネクタ関連の制御等を行い、リンクレイヤ 8 1 2 は、パケット転送やサイクルタイムの制御等を行なう。

【 0 0 7 2 】

ハードウェア部 8 0 0 の上位には、トランザクションレイヤ 8 1 4 及びマネージメントレイヤ 8 1 5 を含むファームウェア部 8 0 1 が位置づけられている。そのトランザクションレイヤ 8 1 4 は、転送（トランザクション）すべきデータの管理を行ない、R e a d、W r i t e、L o c k の命令を出す。また、マネージメントレイヤ 8 1 5 は、1 3 9 4 シリアルバスに接続されている各機器の接続状況や I D の管理を行ない、ネットワークの構成を管理する。

【 0 0 7 3 】

これらのハードウェア部 8 0 0 及びファームウェア 8 0 1 までが、1 3 9 4 シリアルバスの実質的な構成である。

【 0 0 7 4 】

ファームウェア 8 0 1 の上位には、アプリケーションレイヤ 8 1 6 を含むソフトウェア部 8 0 2 が位置づけられている。そのアプリケーションレイヤ 8 1 6 は、利用されるソフトによって異なり、インタフェース上でどのようにしてデータを転送するかは、プリンタや A V / C プロトコル等のプロトコルによって定義されている。

【 0 0 7 5 】

図 3 は、1 3 9 4 シリアルバスにおけるアドレス空間の図を示す図である。

1 3 9 4 シリアルバスに接続された各機器（ノード）には必ず、上記図 3 に示すような、各ノード固有の 6 4 ビットアドレスを持たせておく。このアドレスは機器のメモリに格納されており、これにより自分や相手のノードアドレスを常時認識することで、通信相手を指定したデータ通信が行える。

【 0 0 7 6 】

1 3 9 4 シリアルバスのアドレッシングは、I E E E 1 2 1 2 規格に準じた方式で行われ、アドレス設定については、最初の 1 0 ビットがバスの番号（バス N o . ）の指定用に、次の 6 ビットがノード I D （ノード N o . ）の指定用に使用される。そして、残りの 4 8 ビットが、機器に与えられたアドレス幅になる。この 4 8 ビット領域は、各々固有のアドレス空間として使用できる。そのうちの最後の 2 8 ビットについては、機器に固有のデータの領域（固有データ領域）であり、各機器の識別や使用条件の指定の情報等が格納される。

【 0 0 7 7 】

以上が、1 3 9 4 シリアルバスについての概要である。

つぎに、1 3 9 4 シリアルバスの特徴をより詳細に説明する。

【 0 0 7 8 】

[1 3 9 4 シリアルバスの電氣的仕様]

【 0 0 7 9 】

図 4 は、1 3 9 4 シリアルバス用のケーブルの断面を示す図である。

この図 4 に示すように、1 3 9 4 シリアルバス用のケーブルでは、2 組のツイストペア信号線の他に、電源ラインが設けられている。このような構成によって、電源を持たない機器や、故障等により電圧低下した機器等にも、電力の供給が可能となる。

また、電源線により供給される直流電力の電圧は、8 ~ 4 0 V、その電流は、最大電流 1 . 5 A に規定されている。

尚、D V ケーブルと呼ばれる規格では、電源ラインを省いた四線で構成される。

【 0 0 8 0 】

[D S - L i n k 方式]

10

20

30

40

50

【 0 0 8 1 】

図 5 は、1 3 9 4 シリアルバスでデータ転送方式として採用されている、D S - L i n k (Data/Strobe Link) 方式を説明するための図である。

D S - L i n k 方式は、高速なシリアルデータ通信に適し、2 組の信号線を必要とする。すなわち、2 組の対線のうち一方の信号線でデータ信号を送り、他方の信号線でストロブ信号を送る構成となっている。したがって、受信側は、このデータ信号とストロブ信号を受信して排他的論理和をとることによって、クロックを再現することができる、という特徴がある。

このように、D S - L i n k 方式では、データ信号中にクロック信号を混入させる必要がない。したがって、D S - L i n k 方式では、他のシリアルデータ転送方式に比べ転送効率が高いクロック信号を生成できるため、位相ロックループ (P L L) 回路が不要になり、その分コントローラ L S I の回路規模を小さくすることができる。また、転送すべきデータが無いときに、アイドル状態であることを示す情報を送る必要が無いいため、各機器のトランシーバ回路をスリープ状態にすることができ、消費電力の低減が図れる。

【 0 0 8 2 】

[バスリセットのシーケンス]

【 0 0 8 3 】

1 3 9 4 シリアルバスに接続されている各機器 (ノード) には、ノード I D が与えられ、ネットワークを構成するノードとして認識される。

【 0 0 8 4 】

例えば、ネットワーク機器の接続分離や、電源の O N / O F F 等によるノード数の増減、すなわちネットワーク構成に変化があり、新たなネットワーク構成を認識する必要があるとき、その変化を検知した各ノードは、バス上にバスリセット信号を送信して、新たなネットワーク構成を認識するモードに入る。

このときのネットワーク構成の変化の検知は、1 3 9 4 コネクタ・ポート (上記図 2 、以下、単に「コネクタポート」と言う) において、バイアス電圧の変化を検知することによって行われる。

【 0 0 8 5 】

そこで、あるノードからバスリセット信号が送信されると、各ノードのフィジカルレイヤ 8 1 1 (上記図 2) は、送信されてきたバスリセット信号を受信すると同時に、リンクレイヤ 8 1 2 (上記図 2) にバスリセット信号の発生を伝達し、且つ他のノードに対してバスリセット信号を送信する。そして、最終的に全てのノードがバスリセット信号を受信した後、バスリセットのシーケンスが起動される。

【 0 0 8 6 】

尚、バスリセットのシーケンスは、ケーブルが抜き挿しされた場合や、ネットワークの異常等をハードウェアが検出した場合に起動されると共に、プロトコルによるホスト制御等、フィジカルレイヤ 8 1 1 (上記図 2) に直接命令を与えることによっても起動される。また、バスリセットのシーケンスが起動されると、データ転送は一時中断され、そのシーケンスの起動間は待機状態となり、バスリセット終了後、新しいネットワーク構成のもとで再開される。

【 0 0 8 7 】

[ノード I D 決定のシーケンス]

【 0 0 8 8 】

上述のようにして、バスリセットのシーケンスが起動され、バスリセットが終了した後、各ノードは、新しいネットワーク構成を構築するために、各ノードに I D を与える動作に入る。このときの、バスリセットからノード I D 決定までの一般的なシーケンスについてを、図 6 ~ 図 8 の各フローチャートを用いて説明する。

【 0 0 8 9 】

上記図 6 は、バスリセット信号の発生から、ノード I D が決定してデータ転送が行えるようになるまでの、一連のシーケンスを示すフローチャートである。

10

20

30

40

50

この図6において、まず、各ノードは、バスリセット信号を常時監視し(ステップS101)、バスリセット信号が発生したことを検知すると、ネットワーク構成がリセットされた状態において新たなネットワーク構成を得るために、互いに直結されている各ノード間で親子関係を宣言する(ステップS102)。

このステップS102の処理は、ステップS103の判定により、全てのノード間で親子関係が決定されたと判定されるまで繰り返される。そして、全てのノード間で親子関係が決定されると、次に、ルートを決定する(ステップS104)。

【0090】

ステップS104にてルートが決定されると、次に、各ノードにIDを与えるノードIDの設定作業を行う(ステップS105)。

このステップS105の処理は、ルートから所定のノード順にノードIDの設定を行う処理であり、ステップS106の判定により、全てのノードにIDが与えられたと判定されるで繰り返される。

そして、全てのノードへのノードIDの設定が終了すると、新しいネットワーク構成が全てのノードにおいて認識され、ノード間のデータ転送が行える状態となる。この状態にて、各ノードは、データ転送を開始し(ステップS107)、これと同時にステップS101へと戻り、再びバスリセット信号の発生を監視する。

【0091】

そこで、上記図7は、上述のバスリセット信号の監視(ステップS101)からルート決定(ステップS104)までの処理の詳細を示すフローチャートであり、上記図8は、上述のID設定(ステップS105、S106)の処理の詳細を示すフローチャートである。

【0092】

まず、上記図7において、各ノード(機器)は、バスリセット信号の発生を監視し(ステップS201)、バスリセット信号が発生したこと検知する。これにより、ネットワーク構成は、一旦リセットされる。

【0093】

次に、リセットされたネットワーク構成を再認識する作業の第一段階として、各ノードは、フラグFLをリーフ(ノード)であることを示すデータでリセットする(ステップS202)。

その後、各ノードは、ポート数、すなわち自分に接続されている他ノードの数を調べ(ステップS203)、その結果に応じて、これから親子関係の宣言を始めるために、未定義(親子関係が決定されていない)ポートの数を調べる(ステップS204)。

尚、ステップS204で検知される未定義ポート数は、バスリセットの直後はポート数に等しいが、親子関係が決定されていくに従って減少する。

【0094】

ここで、バスリセットの直後、親子関係の宣言を行えるのは、実際のリーフに限られている。リーフであるか否かは、ステップS203のポート数の確認結果から知ることができ、すなわちこのポート数が"1"であればリーフである。したがって、ステップS204で検知される未定義ポート数が"1"であった場合、リーフは、接続相手のノードに対して親子関係の宣言「自分は子、相手は親」を行い(ステップS205)、本シーケンスを終了する。

【0095】

また、ステップS203のポート数の確認結果が"2"以上であったノード、すなわちブランチ(ノード)は、バスリセットの直後はステップS204で検知される未定義ポート数が"未定義ポート数>1"となるため、フラグFLにブランチを示すデータをセットして(ステップS206)、他ノードから親子関係が宣言されるのを待つ(ステップS207)。

そして、他ノードから親子関係が宣言され、それを受けたブランチは、ステップS204に戻って未定義ポート数を確認する。このとき、もし未定義ポート数が"1"になってい

10

20

30

40

50

れば、残ポートに接続されている他ノードに対して、ステップS205で「自分は子、相手は親」の親子関係を宣言することができる。また、まだ未定義ポート数が"2"以上あるブランチは、再度ステップS207で再び他ノードから「親子関係」が宣言されるのを待つことになる。

【0096】

また、何れか1つのブランチ、又は、例外的に子宣言を行えるのにもかかわらずすばやく動作しなかったリーフの未定義ポート数が"0"になると、ネットワーク全体の親子関係の宣言が終了したことになる。このため、未定義ポート数が"0"になった唯一のノード、すなわち全てノードの親に決まったノードは、フラグFLにルート(ノード)を示すデータをセットする(ステップS208)。これにより、このノードは、ルートとして認識されることになり(ステップS209)、その後、本シーケンス終了となる。

10

【0097】

上述のようにして、バスリセットから、ネットワーク内のノード間における親子関係の宣言までの処理が終了する。

次に、各ノードにIDを与える処理を行うが、ここで、最初にIDの設定を行うことができるのは、リーフである。したがって、ここでは、リーフ ブランチ ルートの順に若い番号(ノード番号:0)からIDを設定する。

【0098】

すなわち、上記図8において、先ず、フラグFLに設定されたデータを基に、ノードの種類、すなわちリーフ、ブランチ、及びルートに応じた処理に分岐する(ステップS301)。

20

【0099】

ステップS301の結果が"リーフ"であった場合、ネットワーク内に存在するリーフの数(自然数)が変数Nに設定される(ステップS302)。その後、リーフは、ルートに対してノード番号を要求する(ステップS303)。この要求が複数ある場合、これを受けたルートは、アービトレーションを行い(ステップS304)、ある1つのノードにノード番号を与え、他のノードにはノード番号の取得失敗を示す結果を通知する(ステップS305)。

【0100】

ステップS306の判定により、ノード番号を取得できなかったリーフは、再びステップS303でノード番号の要求を繰り返す。

30

【0101】

一方、ステップS306の判定により、ノード番号を取得できたリーフは、取得したノード番号を含むID情報をブロードキャストすることで、全ノードに通知する(ステップS307)。このID情報のブロードキャストが終了すると、リーフの数を表す変数Nがデクリメントされる(ステップS308)。

その後、ステップS309の判定により、変数Nが"0"になるまで、ステップS303~ステップS308の処理が繰り返される。全てのリーフのID情報がブロードキャストされると、次のブランチのID設定処理(ステップS310~S317)に移る。

【0102】

40

このブランチのID設定処理は、ステップS301の結果がブランチであった場合にも実行される処理であり、上述のリーフのID設定時と同様に、先ず、ネットワーク内に存在するブランチの数(自然数)が変数Mに設定される(ステップS310)。その後、ブランチは、ルートに対してノード番号を要求する(ステップS311)。

この要求に対してルートは、アービトレーションを行い(ステップS312)、ある1つのブランチにはリーフに続く若い番号を与え、ノード番号を取得できなかったブランチには取得失敗を示す結果を通知する(ステップS313)。

【0103】

ステップS324の判定により、ノード番号の取得できなかったブランチは、再びステップS311でノード番号の要求を繰り返す。

50

【 0 1 0 4 】

一方、ノード番号を取得できたブランチは、取得したノード番号を含むID情報をブロードキャストすることで、全ノードに通知する（ステップS315）、。このID情報のブロードキャストが終了すると、ブランチ数を示す変数Mがデクリメントされる（ステップS316）。

その後、ステップS317の判定により、変数Mが"0"になるまで、ステップS311～ステップS316の処理が繰り返され、全てのブランチのID情報がブロードキャストされると、次のルートのID設定処理（ステップS318、S319）に移る。

【 0 1 0 5 】

ここまでの処理が終了すると、最終的にID情報を取得していないノードはルートのみであるため、次のルートのID設定処理では、ルートは、他のノードに与えていない最も若い番号を自分のノード番号に設定し（ステップS318）、そのノード番号を含むID情報をブロードキャストする（ステップS319）。尚、このルートのID設定処理は、ステップS301の結果がルートの場合にも実行される処理である。

【 0 1 0 6 】

上述のような処理により、全てのノードに対してIDが設定される。

そこで、図9に示すネットワーク例を用いて、ノードID決定のシーケンスの具体的な手順を説明する。

【 0 1 0 7 】

この図9に示すネットワークは、ルートであるノードBの下位にはノードAとノードCが直結され、ノードCの下位にはノードDが直結され、ノードDの下位にはノードEとノードFが直結された階層構造を有する。このようなネットワークでの階層構造やルートノード、ノードIDを決定する手順は、以下ようになる。

【 0 1 0 8 】

バスリセット信号が発生した後、各ノードの接続状況を認識するために、まず、各ノードの直接接続されているポート間において親子関係の宣言がなされる。ここでいう「親子」とは、階層構造の上位が「親」、下位が「子」という意味である。

【 0 1 0 9 】

上記図9では、バスリセットの後、最初に親子関係を宣言したのはノードAである。ここで、上述したように、1つのポートだけが接続されたノード（リーフ）から親子関係の宣言を開始することができる。これは、ポート数が"1"であれば、ネットワークの末端、すなわちリーフであるためである。これが認識されると、それらのリーフ中で最も早く動作を行なったノードから親子関係が決定されていくことになる。このようにして、親子関係の宣言を行なったノードのポートが互いに接続された2つのノードの「子」と設定され、相手ノードのポートが「親」と設定される。したがって、上記図9では、ノードA - B間、ノードE - D間、ノードF - D間で「子 - 親」と設定されることになる。

【 0 1 1 0 】

次に、階層が1つ上がって、複数のポートを持つノード、すなわちブランチのうち、他ノードから親子関係の宣言を受けたノードから順次、上位のノードに対して親子関係の宣言がなされる。

【 0 1 1 1 】

上記図9では、まず、ノードD - E間、D - F間の親子関係が決定された後、ノードDがノードCに対して親子関係を宣言し、その結果、ノードD - C間で「子 - 親」の関係が設定される。

ノードDからの親子関係の宣言を受けたノードCは、もう1つのポートに接続されているノードBに対して親子関係を宣言し、これによってノードC - B間で「子 - 親」の関係が設定される。

【 0 1 1 2 】

このようにして、上記図9に示すような階層構造が構成され、最終的に接続されている全てのポートにおいて親となったノードがルートと決定される（上記図9では、ノードBが

10

20

30

40

50

ルートとなる)。

尚、ルートは、1つのネットワーク構成中に1つしか存在しない。また、ノードAから親子関係を宣言されたノードBが速やかに他のノードに対して親子関係を宣言した場合は、例えば、ノードC等の他のノードがルートになる可能性もあり得る。すなわち、親子関係の宣言が伝達されるタイミングによっては、どのノードもルートとなる可能性があり、ネットワーク構成が同一であっても、特定のノードがルートになるとは限らない。

【0113】

ルートが決定されると、各ノードIDの決定モードに入る。全てのノードは、決定した自分のID情報を、他の全てのノードに通知するブロードキャスト機能を持っている。

尚、ID情報は、ノード番号、接続されている位置の情報、持っているポートの数、接続のあるポートの数、各ポートの親子関係の情報等を含む情報として、ブロードキャストされる。

10

【0114】

ノード番号の割当としては、上述したようにリーフから開始され、順に、ノード番号=0、1、2、・・・が割り当てられる。そして、ID情報のブロードキャストによって、そのノード番号は、割り当て済みであることが認識される。

全てのリーフがノード番号を取得し終ると、次はブランチへ移り、リーフに続くノード番号が割り当てられる。リーフと同様に、ノード番号が割り当てられたブランチから順にID情報がブロードキャストされ、最後にルートが自己のID情報をブロードキャストする。したがって、ルートは常に最大のノード番号を所有することになる。

20

【0115】

以上のようにして、階層構造全体のID設定が終了し、ネットワーク構成が再構築され、バスの初期化作業が完了することになる。

【0116】

[バスアービトレーション]

【0117】

1394シリアルバスは、データ転送に先立って必ずバスの使用権のアービトレーションを行なう。1394シリアルバスに接続された各機器(ノード)は、ネットワーク上で転送されるデータを各々中継することによって、ネットワーク内の全ての機器に同データを伝える論理的なバス型ネットワークを構成するため、パケットの衝突を防ぐ意味でバスアービトレーションが必要である。これによって、ある時間には、1つの機器だけがデータ転送を行なうことができる。

30

【0118】

図10(a)及び(b)は、アービトレーションを説明するための図であり、上記図10(a)は、バス使用権を要求する動作を示し、上記図10(b)は、バスの使用を許可する動作を示している。

【0119】

バスアービトレーションが始まると、1つ若しくは複数のノードが親ノードに向かって、各々バス使用権を要求する。上記図10(a)では、ノードCが、その親ノードであるノードBに対して、ノードFが、その親ノードであるノードAに対して、各々バス使用権を要求している。

40

この要求を受けた親ノードは、更に親ノードに向かって、バス使用権を要求することで、子ノードによるバス使用権の要求を中継する。この要求は最終的に調停を行なうルートに届けられる。上記図10(a)では、ノードFからの要求を受けたノードAが、その親ノードであるノードBに対してバス使用権を要求している。すなわち、ノードAがノードFによるバス使用権の要求を中継している。

【0120】

バス使用権の要求を受けたルートは、どのノードにバス使用権を与えるかを決定する。この調停作業はルートのみが行なえるものであり、調停に勝ったノードには、バス使用の許可が与えられる。上記図10(b)は、ノードCにバス使用許可が与えられ、ノードFの

50

バス使用権の要求は拒否された状態を示している。このとき、ルートは、バスアービトレーションに負けたノードに対してDP (data prefix) パケットを送り、そのバス使用権の要求が拒否されたことを知らせる。バスアービトレーションに負けたノードのバス使用権の要求は、次のバスアービトレーションまで待たされることになる。

一方、バスアービトレーションに勝ち、バス使用許可を得たノードは、以降、データ転送を開始することができる。

【0121】

ここで、バスアービトレーションの一連の流れのフローチャートを、図11に示して説明する。

【0122】

まず、ノードがデータ転送を開始できるためには、バスがアイドル状態であることが必要である。先に行われていたデータ転送が終了して、現在、バスがアイドル状態にあることを認識するためには、各転送モードで個別に設定されている所定のアイドル時間のギャップ長(例えば、サブアクション・ギャップ)の経過を検出すればよい。

【0123】

そこで、先ず、各ノードは、転送する非同期データ又は同期データに応じた所定のギャップ長が得られたか否かを判定する(ステップS401)。この所定のギャップ長が得られない限り、各ノードは、転送を開始するために必要なバス使用権を要求することはできない。したがって、各ノードは、所定のギャップ長が得られるまで待ち状態となる。

【0124】

ステップS401により、所定のギャップ長が得られると、そのノードは、転送すべきデータがあるか判定し(ステップS402)、転送データ有りの場合には、バス使用権を要求する信号をルートに対して送信する(ステップS403)。このバス使用権の要求を示す信号は、上記図10(a)に示したように、ネットワーク内の各機器に中継されながら、最終的にルートに届けられる。

尚、ステップS402において、転送データ無しと判定された場合、そのノードは、そのまま待機状態となる。

【0125】

ルートは、バス使用権を要求する信号を1つ以上受信すると(ステップS404)、そのバス使用権を要求したノードの数を調べる(ステップS405)。

このステップS405の結果、バス使用権を要求したノードが1つであった場合、ルートは、そのノードに直後のバス使用許可を与える(ステップS408)。

【0126】

一方、ステップS405の結果、バス使用権を要求したノードが複数であった場合、ルートは、直後のバス使用許可を与えるノードを1つに絞る調停作業を行う(ステップS406)。この調停作業は、毎回同じノードのみにバスの使用許可が与えられるという様なことはなく、平等にバス使用権が与えられるようにするための作業である(フェア・アービトレーション)。

【0127】

その結果、調停に勝った1つのノードと、敗れたその他のノードとに応じて、処理が分岐する(ステップS407)。

これにより、調停に勝った1つのノードには、直後のバス使用許可を示す許可信号が送られる(ステップS408)。したがって、この許可信号を受信したノードは、直後に転送すべきデータ(パケット)の転送を開始する。そして、そのデータ転送完了後、ステップS401へと戻る。

また、調停に敗れたノードには、バス使用権の要求が拒否されたことを示すDP (data prefix) パケットが送られる(ステップS409)。したがって、DPパケットを受け取ったノードは、再度バス使用権を要求するために、ステップS401へと戻る。

【0128】

[Asynchronous転送]

10

20

30

40

50

【 0 1 2 9 】

図 1 2 は、アシンクロナス転送における時間的な遷移状態を示した図である。この図 1 2 において、最初の " subaction gap " (サブアクション・ギャップ) は、バスのアイドル状態を示すものである。このアイドル状態の時間が所定値になった時点で、データ転送を希望するノードがバス使用権を要求できると判定し、したがって、上記図 1 1 で説明したようなバスアービトレーションが実行されることになる。

【 0 1 3 0 】

バスアービトレーションによりバスの使用が許可されると、データ転送がパケットされる。このデータを受信したノードは、" a s k g a p " という短いギャップの後、受信確認用の返送コード " a c k " を返して応答する (又は、応答パケットを送る) ことによ

10

【 0 1 3 1 】

図 1 3 は、アシンクロナス転送用のパケットフォーマットを示す図である。

パケットには、データ部及び誤り訂正用のデータ CRC の他にヘッダ部があり、そのヘッダ部には、目的ノード ID、ソースノード ID、転送データ長や各種コード等が書き込まれている。

ここで、アシンクロナス転送は、自己ノードから相手ノードへの 1 対 1 の通信である。したがって、転送元ノードから送り出されたパケットは、ネットワーク中の各ノードに行き渡

20

【 0 1 3 2 】

[Isochronous 転送]

【 0 1 3 3 】

1 3 9 4 シリアルバスの最大の特徴であるともいえるこのアイソクロナス転送は、特に、AV データ等のリアルタイム転送を必要とするマルチメディアデータの転送に適している。

また、アシンクロナス転送が 1 対 1 の転送であったのに対し、このアイソクロナス転送は、ブロードキャスト機能によって、1 つの転送元ノードから他の全てのノードへ様にデータ

30

【 0 1 3 4 】

図 1 4 は、アイソクロナス転送における時間的な遷移状態を示す図である。

アイソクロナス転送は、バス上で一定時間毎に実行される。この時間間隔は、アイソクロナスサイクルと呼ばれ、1 2 5 μ S とする。この各サイクルの開始時間を示し、各ノードの時間調整を行なう役割を担っているのが、サイクル・スタート・パケット (C S P) である。C S P を送信するのは、サイクル・マスタと呼ばれるノードであり、1 つ前のサイクル内の転送終了し、所定のアイドル期間 (サブアクションギャップ) を経た後、本サイクルの開始を告げる C S P を送信する。すなわち、C S P の送信される時間間隔が 1 2 5 μ S となる。

40

【 0 1 3 5 】

また、上記図 1 4 に " チャンネル A "、" チャンネル B "、及び " チャンネル C " と示すように、1 つの同期サイクル内において、複数種のパケットにチャンネル ID を各々与えることによって、それらのパケットを区別して転送することができる。これによって、複数のノード間で、略同時に、リアルタイム転送が可能となり、また、受信ノードは、自分が望むチャンネル ID のデータのみを受信すればよい。

尚、チャンネル ID は、受信ノードのアドレス等を表すものではなく、データに対する論理的な番号を与えているに過ぎないものである。したがって、送信されたパケットは、1 つの送信元ノードから他の全てのノードに行き渡る、すなわちブロードキャストされることになる。

50

【 0 1 3 6 】

アイソクロナス転送では、そのパケット送信に先立って、上述のアシクロナス転送と同様にバスアービトレーションが行われる。しかしながら、アイソクロナス転送はアシクロナス転送のように1対1の通信ではないため、上記図14に示すように、アイソクロナス転送には、受信確認用の返送コードである " a s k " は存在しない。

また、上記図14に示す " i s o g a p " (アイソクロナスギャップ) は、アイソクロナス転送を行なう前にバスがアイドル状態であることを認識するために必要なアイドル期間を表している。この所定のアイドル期間を経過すると、アイソクロナス転送を行ないたいノードに対するバスのアービトレーションが行なわれる。

【 0 1 3 7 】

図15は、アイソクロナス転送用のパケットフォーマットを示す図である。

各チャンネルに分けられた各種のパケットには、各々データ部及び誤り訂正用のデータCRCの他にヘッダ部があり、そのヘッダ部には、転送データ長やチャンネルNo.、その他各種コード及び誤り訂正用のヘッダCRC等が書き込まれている。

【 0 1 3 8 】

[バス・サイクル]

【 0 1 3 9 】

実際に、1394シリアルバスにおいては、アイソクロナス転送とアシクロナス転送が混在できる。その時のバス上の転送状態の時間的な遷移の様子を、図16に示す。

アイソクロナス転送は、アシクロナス転送より優先して実行される。その理由は、C S Pの後、アシクロナス転送を起動するために必要なアイドル期間のギャップ長(" s u b a c t i o n g a p " サブアクションギャップ)よりも短いギャップ長(" a c k g a p " アイソクロナスギャップ)で、アイソクロナス転送を起動できるからである。したがって、アシクロナス転送よりアイソクロナス転送が、優先して実行されることとなる。

【 0 1 4 0 】

上記図16示す一般的なバスサイクルでは、サイクル#mのスタート時にC S Pがサイクル・マスタから各ノードに転送される。このC S Pによって、各ノードの動作の同期がとられ、所定のアイドル期間(アイソクロナスギャップ)を待ってからアイソクロナス転送を行おうとするノードは、バスアービトレーションに参加して、パケット転送に入る。上記図16では、" チャンネルe"、" チャンネルs"、" チャンネルk" が順にアイソクロナス転送されている。

このバスアービトレーションからパケット転送までの動作を、与えられているチャンネル分繰り返し行なった後、サイクル#mにおけるアイソクロナス転送が全て終了すると、アシクロナス転送を行うことができるようになる。

【 0 1 4 1 】

すなわち、アイドル時間が、アシクロナス転送が可能なサブアクションギャップに達することによって、アシクロナス転送を行いたいノードは、バスアービトレーションに参加する。

尚、アシクロナス転送が行えるのは、アイソクロナス転送終了後から、次のC S Pを転送すべき時間(cycle synch)までの間に、アシクロナス転送を起動するためのサブアクションギャップが得られた場合に限られる。

【 0 1 4 2 】

上記図16のサイクル#mでは、3つのチャンネル分のアイソクロナス転送の後、アシクロナス転送により、" a c k " を含む2つのパケット(パケット1、パケット2)転送されている。この2つのパケット転送後は、サイクル#(m+1)をスタートすべき時間(cycle synch)に至るので、サイクル#mでの転送はこれ終了する。

このとき、非同期又は同期転送中に次のC S Pを送信すべき時間(cycle synch)に至った場合、転送を無理に中断せず、その転送が終了した後に、アイドル期間を経て、次サイクルのC S Pを送信する。すなわち、1つのサイクルが125 μ S以上続いた場合は、その延長分、次サイクルが基準の125 μ Sより短縮される。このように、アイソクロナス

10

20

30

40

50

・サイクルは、125 μ S を基準に超過或いは短縮し得るものである。
尚、アイソクロナス転送は、リアルタイム転送を維持するために、必要であれば毎サイクル実行され、アシンクロナス転送は、サイクル時間が短縮されたことによって、次以降のサイクルに延期されることもある。こういった遅延情報も、サイクル・マスタによって管理される。

【0143】

(第1の実施の形態)

【0144】

図17は、1394シリアルバスのインターフェースをLANでよく用いられるOSIモデルの各層と対比させた図である。

OSIモデルの物理層(Physical)1とデータリンク層(Data Link)2が、1394シリアルバスのインターフェースの下位層(IEEE1394 Layers)4であるフィジカル層811及びリンク層812(上記図2)に該当する。下位層4の上に存在する1394シリアルバスのインターフェースにおけるトランスポートプロトコル層(Transport Protocols)5とプレゼンテーション層(Presentation)6は、OSIモデルのネットワーク層(Network)、トランスポート層(Transport)、セッション層(Session)、及びプレゼンテーション層(Presentation)を含む上位層3に該当する。

また、本発明の特徴であるLOGINプロトコル(Login Protocol)(初期プロトコル)7は、1394シリアルバスのインターフェースの下位層4とトランスポートプロトコル層5の間で動作するものである。

【0145】

上記図17に示す例1(Example 1)では、プリンタ等の周辺機器用のシリアルバスプロトコル(SBP-2)(デバイス固有のプロトコル)8に準拠したデバイスに対して、LOGINプロトコル7を持たせる。これにより、相手のデバイスに対してSBP-2に準拠したプロトコルを使って、データのやり取りを行いたいことを通知することができる。

また、上記図17に示す例2(Example 2)では、1394シリアルバスのインターフェース上で特化されたデバイスプロトコル(Device Specific Protocol)(デバイス固有のプロトコル)9に対して、LOGINプロトコル7を持たせる。これにより、この例2においても、デバイスが互いに、互いのプロトコルをサポートしていることを判別することができる。

【0146】

図18は、LOGINプロトコルの基本動作を示す図であり、プリンタデバイスは、ホストデバイスからの印字タスク(Print task)10を実行する際に、まず、プリンタに用意されているプリンタプロトコルA、B、及びC(Protocol A、B、C)のうち、どれを選択して印字データのやり取りを行うかをLOGINプロトコル7による通信に基づいて決定し、その後、その決定したプリンタプロトコルに従って、印字データのやり取りを行う。

すなわち、いくつかのプリンタプロトコルをサポートしているデバイスは、ホストデバイスと接続する際に、まず、ホストデバイスに用意されているトランスポートプロトコル5をLOGINプロトコル7によって判別し、ホストデバイスのトランスポートプロトコル5に合ったプリンタプロトコルを選択し、選択したプリンタプロトコルに従った印字データやコマンドのやり取りを行うことで、印字タスク10の処理を行う。

【0147】

図19は、1394シリアルバスにおける接続形態を示す図であり、ここでは、複数のプリンタプロトコルに対応したプリンタ11に対してLOGINプロトコル7を実装したデバイス(PC12、スキャナ13、VCR14等)が接続された状態を示している。プリンタ11は、LOGINプロトコル7により判別した、接続を要求する相手デバイスのトランスポートプロトコル5に応じてプリンタプロトコルを切り替えることにより、各デバイス12~14からの印字タスクを問題なく処理することができる。

【 0 1 4 8 】

図 2 0 は、ログイン動作の流れを示す図である。

第 1 ステップ (STEP 1) において、

- ・ホストデバイス (HOST) は、ターゲットデバイス (DEVICE、この場合マルチプロトコルプリンタ) をロックする。
- ・ターゲットデバイスは、ホストデバイスのケーパビリティ (トランスポートプロトコル等を含む) を調べる。このケーパビリティは、後述するレジスタ 5 0 3 に格納される。
- ・ターゲットデバイスは、ホストデバイスのケーパビリティ (トランスポートプロトコル等を含む) をセットする。

第 2 ステップ (STEP 2) において、

- ・第 1 ステップで決定されたプロトコルで、プリントデータを通信する。

第 3 ステップ (STEP 3) において、

- ・ホストデバイスは、ターゲットデバイスとの接続を切断する。

【 0 1 4 9 】

図 2 1 は、LOG IN プロトコルのためにターゲットデバイスであるプリンタが備える 1 3 9 4 シリアルバスの CSR において、ロックレジスタ (lock) 5 0 1、プロトコルレジスタ (protocol) 5 0 2、ケーパビリティレジスタ (capability) 5 0 3 を示した図である。

これらのレジスタは、1 3 9 4 シリアルバスのアドレス空間における初期ユニット空間の定められたアドレスに配置される。すなわち、上記図 3 に示したように、機器に与えられたアドレス幅 4 8 ビットのうち、最初の 2 0 ビットにおける 0xFFFF がレジスタ空間と呼ばれ、その最初の 5 1 2 バイトに CSR アーキテクチャのコアになるレジスタ (CSR コア) が配置されている。

尚、このレジスタ空間には、バスに接続された機器間で共通な情報が置かれる。また、0 ~ 0xFFFFD はメモリ空間、0xFFFFE はプライベート空間と、各々呼ばれ、プライベート空間は、機器内で自由に利用できるアドレスであり、各機器間のコミュニケーションに使われる。

【 0 1 5 0 】

ロックレジスタ 5 0 1 は、リソースのロック状態 (専有状態) を示し、" 0 " でログイン可能な状態を表し、" 0 " 以外でロック状態で既にログインされていることを表す。

プロトコルレジスタ 5 0 2 には、実際に通信に使用するプロトコルが書き込まれる。したがって、このプロトコルレジスタ 5 0 2 は、現在設定されているプロトコルを示し、設定されたプロトコルに対応するケーパビリティレジスタ 5 0 3 のビットに相当するビットの値が " 1 " になる。

ケーパビリティレジスタ 5 0 3 には、各々のデバイスが対応可能なプロトコルを示す情報が格納され、各ビットの各々が独立に対応可能なデータトランスポートプロトコルを示す。すなわち、ケーパビリティレジスタ 5 0 3 は、複数ビットを有し、そのビット毎で設定可能なトランスポートプロトコルを各々独立に示す。例えば、" 1 " のビットに対応するプロトコルは設定可能であることを表し、" 0 " のビットに対応するプロトコルは設定不可能であることを表す。

【 0 1 5 1 】

図 2 2 は、1 3 9 4 シリアルバスにより構成されたネットワーク (1 3 9 4 ネットワーク) におけるプリンタマップのフォーマットを示した図である。

このプリンタマップには、応答を返してきたプリンタのユニーク ID (Unique ID)、ノード ID (Node ID)、ステータス (Status)、及びケーパビリティ (Capability) 等の情報が保存される。このとき、" ノード ID " は、図 2 3 に示すような CSR アーキテクチャにおけるフォーマットに従って保存される。

尚、ここでの " ステータス (Status) " は、例えば、上記図 2 1 に示したプロトコルレジスタ 5 0 2 の内容を示し、" ケーパビリティ (Capability) " は、例えば、上記図 2 1 に示したケーパビリティレジスタ 5 0 3 の内容を示す。

10

20

30

40

50

【 0 1 5 2 】

図 2 4 は、上記図 2 2 に示したようなプリンタマップの作成コマンドのフォーマットを示した図である。このコマンドは、非同期パケットのライトトランザクションにより、ターゲットデバイスに対して送信される。また、このコマンドは、1 3 9 4 シリアルバスのアドレス空間におけるユニット空間の定められたアドレスに配置される。

【 0 1 5 3 】

図 2 5 は、複数のプロトコルに対応可能なプリンタが 1 3 9 4 シリアルバスに接続されて構成されるネットワークにおいて、ホストデバイスでのプリンタマップ作成処理を示したフローチャートである。

【 0 1 5 4 】

ここで、ネットワーク上には、通常、様々なデバイスが接続されている。このような状況下で、イニシエータ（ホストデバイス）がターゲットデバイスであるプリンタとの通信を試みるとき、どのデバイスが目的とするプリンタに対応しているかを認識する必要がある。また、適切なプリンタ出力を得るためには、それに対応したプリンタの物理的な場所、その能力、その余力等を予め認識しておくこと非常に都合がよい。

【 0 1 5 5 】

そこで、ここでは、同一ネットワークに存在する各プリンタを予め調査する。

具体的には、ホストデバイスは、予め、ネットワーク上に存在する各プリンタを調査することで（第 1 の調査処理）、各々の物理的な場所、能力、及び余力等の情報（以下、トポロジー／能力情報とも言う）を得て、上記図 2 2 に示したようなプリンタマップを作成する。そして、そのプリンタマップに基づいて、目的とするプリンタを選択する。

【 0 1 5 6 】

プリンタマップの作成については、上記図 2 5 に示すように、先ず、ホストデバイスは、上記図 2 4 に示したようなプリンタマップ作成のためのコマンド（作成コマンド）を、ブロードキャストすることで（ステップ S 3 0 0 1）、ネットワーク上に存在する全てのデバイスに送信する。

そして、ホストデバイスは、各デバイスからの応答、すなわちネットワーク上に存在する各プリンタからの応答（応答コマンド）を待つ（ステップ S 3 0 0 2）。

【 0 1 5 7 】

次に、ホストデバイスは、各プリンタからの応答コマンドを受信すると、その応答コマンドを返してきたプリンタのプロトコルレジスタ及びケーパビリティレジスタの内容を読み取る（ステップ S 3 0 0 3）。これにより、ホストデバイスは、そのプリンタの能力や状態等を認識する。

【 0 1 5 8 】

そして、ホストデバイスは、ステップ S 3 0 0 3 で得た情報を基に、現在のネットワーク上のプリンタについてのプリンタマップを作成して（ステップ S 3 0 0 4）、本処理を終了する。

【 0 1 5 9 】

図 2 6 は、上述したホストデバイスから作成コマンドが与えられるターゲットデバイス、特に、プリンタでの処理を示すフローチャートである。

上記図 2 6 において、先ず、プリンタは、電源投入時からステータス及びケーパビリティを提示する（ステップ S 3 1 0 1）。

具体的には、例えば、現在の自プリンタの能力や状態等に応じて、プロトコルレジスタ及びケーパビリティレジスタの設定処理を行う。したがって、プリンタ内部でのステータス及びケーパビリティの変化は、このステップ S 3 1 0 1 でのステータス及びケーパビリティの提示にも反映することになる。

【 0 1 6 0 】

次に、プリンタは、ホストデバイスからの作成コマンドを待つ（ステップ S 3 1 0 2）。そして、プリンタは、ホストデバイス側からの作成コマンドを受信すると、その作成コマンドに対する応答コマンドをホストデバイスへ返送する（ステップ S 3 1 0 3）。

10

20

30

40

50

【 0 1 6 1 】

図 2 7 は、ホストデバイスにおけるログイン処理を示すフローチャートである。

ここで、ホストデバイスがターゲットデバイス（ここでは、プリンタ）へのログインを開始するためには、上記図 2 5 に示したプリンタマップ作成処理（ステップ S 6 0 0）の後、リード又はロックランザクションによって、ログインしようとするプリンタのロックレジスタ 5 0 1、プロトコルレジスタ 5 0 2、及びケーパビリティレジスタ 5 0 3 のデータを確認する必要がある。

【 0 1 6 2 】

そこで、先ず、ホストデバイスは、プリンタのケーパビリティレジスタ 5 0 3 のビット毎のデータから、自分が通信に用いようとしているプロトコルをプリンタがサポートしているかを確認する（ステップ 6 0 1）。このとき、もし、プリンタが自分のプロトコルをサポートしていない場合には、そのプリンタへのログインを中止とする。

10

【 0 1 6 3 】

次に、ホストデバイスは、プリンタのロックレジスタ 5 0 1 のデータから、現在ログイン可能であるかを確認する（ステップ 6 0 2）。すなわち、プリンタのロックレジスタ 5 0 1 のデータが " 0 " 以外であれば、他のデバイスがプリンタへログイン中であると見なし、ログイン中止（Login N.G.）とし、本処理を終了する。一方、プリンタのログインレジスタ 5 0 1 のデータが " 0 " であれば、現在ログイン可能（Login O.K.）と見なし、次のステップに進む。

尚、ステップ 6 0 1 において、プリンタがホストデバイスのプロトコルをサポートしておらず、ログインを中止とされた場合、このステップ 6 0 2 の判定により、本処理終了となる。

20

【 0 1 6 4 】

ステップ 6 0 2 の判定により、プリンタへのログインが可能であった場合、ホストデバイスは、ロックランザクションによって、プリンタのロックレジスタ 5 0 1 に " 1 " を書き込み、ログインを設定する（ステップ 6 0 3）。この状態でプリンタはロックされたことになり、他のデバイスからの制御は不可能となり、また、レジスタの変更も不可能となる。

【 0 1 6 5 】

上述のようにして、プリンタのリソースがロックされた状態となると、次に、ホストデバイスは、プロトコルの設定を行なう。

30

ここで、プリンタは、複数のプリンタプロトコルをサポートするため、プリントデータを受け取る前に、ホストデバイスが使用できるプロトコルを知る必要がある。

そこで、ホストデバイスは、ライトランザクションによって、プリンタのプロトコルレジスタ 5 0 2 の相当するビットを設定することで、これから使用するプロトコルをプリンタに通知する（ステップ 6 0 4）。

【 0 1 6 6 】

この時点で、ホストデバイスが通信に用いるプロトコルがプリンタに通知され、且つプリンタがロック状態であるため、現在、プリンタでバスにログインしているホストデバイスがデータ（ここでは、プリントデータ）の送信を行なう（ステップ 6 0 5）。

40

【 0 1 6 7 】

プリントデータの送信が終了すると、ホストデバイスは、プリンタのロックレジスタ 5 0 1、プロトコルレジスタ 5 0 2、及びケーパビリティレジスタ 5 0 3 をクリアすることにより、プリンタからログアウトする（ステップ 6 0 6）。

【 0 1 6 8 】

図 2 8 は、ターゲットデバイスであるプリンタのログイン処理を示す図である。

先ず、プリンタ側では、図 2 6 に示したプリンタマップ作成処理が実行される（ステップ S 7 0 0）。

ここで、通常、ホストデバイスからログインされるのを待つ状態にあるホストデバイスからのプリントリクエストは、プリンタのロックレジスタ 5 0 1、プロトコルレジスタ 5 0

50

2、及びケーパビリティレジスタ503の読み取りにより開始されるため、プリンタは、これらのレジスタを常に他のデバイスから読み出し可能な状態にしておく。この状態により、今、プリントアウトを実行しようとするホストデバイスからプリンタがロックされたとする(ステップS701)。

【0169】

次に、プリンタは、ホストデバイスから使用プロトコルが通知されるのを待つ(ステップS702)。このように、プリンタがロック状態になってから使用プロトコルの通知を待つのは、ログインの途中で他のデバイスからのリクエストにより、プロトコルレジスタ502の内容が書き換えられないようにするためである。

【0170】

そして、ホストデバイスからプリンタへ使用プロトコルの通知があると(ステップS703)、プリンタは、自分のプロトコルを通知された使用プロトコルに切り換えて、ホストデバイスのプロトコルに合わせた通信を行う(ステップS704~S709)。

【0171】

通信が終了したら、プリンタは、ロックレジスタ501及びケーパビリティレジスタ503がクリアされたのを確認し(ステップS710)、ログインを待つ状態(ステップ701)に戻る。

【0172】

(第2の実施の形態)

【0173】

上述した第1の実施の形態では、上記図25及び図26を用いて説明したように、1394シリアルバスに複数のプリンタを含む各デバイスが接続されてなるネットワーク上において、ホストデバイスが各プリンタについてのプリンタマップを作成し、そのプリンタマップに基づいて目的のプリンタを選択するようにしたが、本実施の形態では、ホストデバイスとプリンタが共に複数のプロトコルをサポートしており、さらにそのようなプリンタが複数存在する場合、ホストデバイスが各プリンタの使用可能なプロトコルを調べて多数決を取り、最も多くサポートされているプロトコルを決定する。

このように、多くのプロトコルが使用可能な状況下で多数決を取るように構成することで、実際に使用するプロトコルの種類を削減することを可能とし、ホストデバイスのプロトコルスイッチによる負荷を下げることを可能とする。

【0174】

尚、本実施の形態は、上記図25及び図26に示した処理以外については第1の実施の形態と同様であるため、その詳細な説明は省略し、ここでは、第1の実施の形態と異なる点についてのみ具体的に説明する。

【0175】

具体的には、まず、1394シリアルバスを用いたネットワーク上において、ホストデバイス(イニシエータ)とターゲットデバイス(プリンタ)が共に複数のプロトコルをサポートしており、さらにそのようなターゲットデバイスが同一ネットワークに複数接続されている場合、当然、ホストデバイスとターゲットデバイスは同一プロトコルを用いる必要がある。

そこで、ここでは、今、どのプロトコルを用いるかを決定するために、ホストデバイスが各ターゲットデバイスの使用可能なプロトコルを調べ、その中で多数決を取り、最も多くサポートされているプロトコルを決定する。

【0176】

このため、本実施の形態では、第1の実施の形態における上記図25及び図26の処理の代わりに、図29及び図30に示す処理を実行する。

尚、上記図29は、ホストデバイスのプリンタマップ作成処理(相対多数プロトコル決定プリンタマップ処理)を示すフローチャートであり、ホストデバイスのログイン処理(上記図27:ステップS600)で実行される。また、上記図30は、この処理時におけるプリンタの処理を示すフローチャートであり、プリンタのログイン処理(上記図29:ス

10

20

30

40

50

テップ S 7 0 0) で実行される。

【 0 1 7 7 】

すなわち、上記図 2 9 において、先ず、ホストデバイスは、上記図 2 2 に示したプリンタマップを作成するために、その作成コマンドをブロードキャストすることで、ネットワーク上に存在する全てのデバイスに送信する (ステップ S 3 2 0 1)。

そして、ホストデバイスは、各デバイスからの応答、すなわちネットワーク上に存在する各プリンタからの応答 (応答コマンド) を待つ (ステップ S 3 2 0 2)。

【 0 1 7 8 】

次に、ホストデバイスは、各プリンタからの応答コマンドを受信すると、その応答コマンドを返してきたプリンタのプロトコルレジスタ及びケーパビリティレジスタの内容を読み取る (ステップ S 3 2 0 3)。これにより、ホストデバイスは、そのプリンタの能力や状態等を認識する。

10

【 0 1 7 9 】

次に、ホストデバイスは、ステップ S 3 2 0 3 で得た情報を基に、現在のネットワーク上のプリンタについてのプリンタマップを作成する (ステップ S 3 2 0 4)。

【 0 1 8 0 】

次に、ホストデバイスは、ステップ S 3 2 0 4 で作成したプリンタマップより、現在ネットワークに接続されているプリンタ (複数のプロトコルをサポートしているマルチプロトコルプリンタ) の使用可能なプロトコルを調べ、それらのプロトコルの中から、最も多くサポートされているプロトコルを選択する (ステップ S 3 2 0 5)。

20

【 0 1 8 1 】

そして、ホストデバイスは、ステップ S 3 2 0 5 で選択したプロトコルを、プロトコル通知コマンドとして各プリンタに通知して (ステップ S 3 2 0 6)、本処理を終了する。

【 0 1 8 2 】

一方、プリンタは、上記図 3 0 に示すように、先ず、電源投入時からステータス及びケーパビリティを提示する (ステップ S 3 3 0 1)。

具体的には、例えば、現在の自プリンタの能力や状態等に応じて、プロトコルレジスタ及びケーパビリティレジスタの設定処理を行う。したがって、プリンタ内部でのステータス及びケーパビリティの変化は、このステップ S 3 3 0 1 でのステータス及びケーパビリティを提示にも反映することになる。

30

【 0 1 8 3 】

次に、プリンタは、ホストデバイスからの作成コマンドを待つ (ステップ S 3 3 0 2)。そして、プリンタは、ホストデバイスからの作成コマンドを受信すると、そのコマンドに対する応答コマンドをホストデバイスへ返送する (ステップ S 3 3 0 3)。

【 0 1 8 4 】

次に、プリンタは、ホストデバイスからのプロトコル通知コマンドを待つ (ステップ S 3 3 0 4)。

そして、プリンタは、ホストデバイスからのプロトコル通知コマンドを受信すると、そのコマンドにより示されるプロトコルに自分のプロトコルを切り換える (ステップ S 3 3 0 5)。

40

【 0 1 8 5 】

(第 3 の実施の形態)

【 0 1 8 6 】

図 3 1 は、本実施の形態における動作を示した図であり、上記図 1 8 に示した第 1 の実施の形態と比較すると、LOG IN プロトコル 7 を実装していないプロトコル D を持つデバイスについても対応している点が特徴である。

すなわち、LOG IN プロトコル 7 を持つデバイスだけでなく、既存のプロトコル D (例えば、AV / C プロトコル) にのみ対応しているデバイスに対しても印字動作の保証を行う為に、プリンタに、LOG IN プロトコル 7 を持たないデバイスに対応するプリンタプロトコルを追加する。

50

【 0 1 8 7 】

この動作について説明すると、接続の初めに行われるプリントリクエストによってホストデバイスが L O G I N プロトコル 7 に対応していないことをプリンタが認識した場合、プリンタは、プロトコル D を使ってホストデバイスとの通信を試み、通信が成立した場合は、そのプロトコル D に従って印字タスク 1 0 を実行する。

【 0 1 8 8 】

図 3 2 は、本実施の形態と、O S I モデルとを対比させた図であり、例 3 (Example 3) では、L O G I N プロトコル 7 の実装されていない現行の A V / C プロトコルに準拠した A V デバイス 1 5 をモデルとしている。例 4 (Example 4) では、L O G I N プロトコル 7 が実装されていないスキャナ用の非標準プロトコルが実装されているスキャナ 1 6 をモデルとしている。

10

すなわち、L O G I N プロトコル 7 を実装していないプロトコルを持つデバイスについても、そのデバイスが持つプロトコルにプリンタが対応することができれば、そのプリンタを利用することができるデバイスの種類を広げることができる。

【 0 1 8 9 】

(第 4 の実施の形態)

【 0 1 9 0 】

上述した第 1 の実施の形態では、上記図 2 5 及び図 2 6 を用いて説明したように、1 3 9 4 シリアルバスに複数のプリンタを含む各デバイスが接続されてなるネットワーク上において、ホストデバイス (イニシエータ) がターゲットデバイスである各プリンタについて
のプリンタマップを作成し、そのプリンタマップに基づいて目的のプリンタを選択する
ようにしたが、本実施の形態では、プリンタをネットワーク上の 1 出力デバイス (ホストデ
バイス) とし、デジタルビデオ、デジタルカメラ、スキャナ、モニタ、コンピュータ
等をイメージ入力装置 (ターゲットデバイス) とし、プリンタがイメージ入力装置にログ
インして画像情報を取得するために、プリンタがイメージ入力装置についてのマップ (上
述のプリンタマップに相当するマップ) を作成し、そのマップに基づいて目的のイメ
ージ入力装置を選択する。

20

【 0 1 9 1 】

尚、本実施の形態は、上記図 2 5 及び図 2 6 に示した処理以外については第 1 の実施の形
態と同様であるため、その詳細な説明は省略し、ここでは、第 1 の実施の形態と異なる点
についてのみ具体的に説明する。

30

【 0 1 9 2 】

まず、上述したように、ネットワーク上には、通常、様々なデバイスが接続されている。
このような状況下で、プリンタ (イニシエータ) がターゲットデバイスであるイメージ入
力装置との通信を試みる時、どのデバイスが目的とするイメージ入力装置に対応してい
るかを認識する必要がある。また、適切なプリンタ出力を得るためには、それに対応した
イメージ入力装置の物理的な場所、その能力、その余力等を予め認識しておく非常に都
合がよい。

【 0 1 9 3 】

そこで、ここでは、同一ネットワークに存在する各イメージ入力装置を予め調査する。

40

具体的には、プリンタは、予め、ネットワーク上に存在する各イメージ入力装置を調査
することで (第 2 の調査処理) 、各々の物理的な場所、能力、及び余力等の情報 (トポロ
ジー / 能力情報) を得て、上記図 2 2 に示したようなプリンタマップに相当するマップを
作成する。そして、そのマップに基づいて、目的とするイメージ入力装置を選択する。

【 0 1 9 4 】

イメージ入力装置についてのマップの作成については、図 3 3 に示すように、先ず、プリ
ンタは、マップ作成のためのコマンド (作成コマンド) を、ブロードキャストすることで
(ステップ S 3 4 0 1) 、ネットワーク上に存在する全てのデバイスに送信する。
そして、プリンタは、各デバイスからの応答、すなわちネットワーク上に存在する各イ
メージ入力装置からの応答 (応答コマンド) を待つ (ステップ S 3 4 0 2) 。

50

【0195】

次に、プリンタは、各イメージ入力装置からの応答コマンドを受信すると、その応答コマンドを返してきたイメージ入力装置のプロトコルレジスタ及びケーパビリティレジスタの内容を読み取る（ステップS3403）。これにより、プリンタは、そのイメージ入力装置の能力や状態等を認識する。

【0196】

そして、プリンタは、ステップS3403で得た情報を基に、現在のネットワーク上のイメージ入力装置についてのマップを作成して（ステップS3404）、本処理を終了する。

【0197】

図34は、上述したプリンタから作成コマンドが与えられるターゲットデバイス、すなわちイメージ入力装置での処理を示すフローチャートである。

上記図34において、まず、イメージ入力装置は、電源投入時からステータス及びケーパビリティを提示する（ステップS3501）。

具体的には、例えば、現在の自装置の能力や状態等に応じて、プロトコルレジスタ及びケーパビリティレジスタの設定処理を行う。したがって、装置内部でのステータス及びケーパビリティの変化は、このステップS3501でのステータス及びケーパビリティの提示にも反映することになる。

【0198】

次に、イメージ入力装置は、プリンタからの作成コマンドを待つ（ステップS3502）。

そして、イメージ入力装置は、プリンタからの作成コマンドを受信すると、その作成コマンドに対する応答コマンドをプリンタへ返送して（ステップS3503）、本処理を終了する。

【0199】

上述のようにして、プリンタにて、イメージ入力装置についてのマップが作成された後、上記図27及び図28に示したようなログイン処理が実行される。この結果、プリンタは、ネットワーク上に存在する各イメージ入力装置から選択した適切なイメージ入力装置からの画像情報をプリント出力することになる。

【0200】

尚、上述した各実施の形態では、IEEE1934シリアルバスを用いてネットワークを構成するものとしたが、本発明はこれに限定されるものではなく、例えば、Universal Serial Bus (USB) と呼ばれるシリアルインターフェース等、任意のシリアルインターフェースを用いて構成されるネットワークにも適用することができる。

【0201】

また、ホストデバイスとしては、例えば、コンピュータ、デジタルカメラ、スキャナ、DVD、Set-top-Box、デジタルテレビ、コンファレンスカメラ、デジタルビデオ、及びこれらを含む複合機等を用いることができる。一方、ターゲットデバイスとしては、モニタ、コンピュータ、外部記憶装置、Set-top-Box、プリンタ及びこれらを含む複合機等を用いることができる。

【0202】

また、上記図27のステップS600、上記図28のステップS700、及び上記図29のステップS3204で実行されるプリンタマップ作成処理や、上記図33のステップS3404で実行されるイメージ入力装置についてのマップ作成処理においては、トポロジ接続状態が上記図9のように調査されマップが作成される、かかるトポロジ接続状態が判定されることによって、単なる多数決ではなく、トポロジ接続状態を加味して実際に使用するプロトコルを決定することができる。

【0203】

また、上述した実施の形態では、ネットワーク上の各デバイスを、画像データを出力或いは入力する機器としたが、これに限らず、例えば、ネットワーク上のデバイスのうち任意

10

20

30

40

50

のデバイスを、画像データの出力或いは入力することなく、単にネットワーク上の状態を検出し、他の機器とネットワークの関係を調査する機能を有する機器としてもよい。

【0204】

また、本発明は、上記図19に示すような複数の機器から構成されるシステムに適用しても、1つの機器からなる装置内のデータ処理方法に適用してもよい。

【0205】

また、本発明の目的は、上述した各実施の形態のホスト及び端末の機能を実現するソフトウェアのプログラムコードを記憶した記憶媒体を、システム或いは装置に供給し、そのシステム或いは装置のコンピュータ（又はCPUやMPU）が記憶媒体に格納されたプログラムコードを読みだして実行することによっても、達成されることは言うまでもない。

10

【0206】

この場合、記憶媒体から読み出されたプログラムコード自体が各実施の形態の機能を実現することとなり、そのプログラムコードを記憶した記憶媒体は本発明を構成することとなる。

【0207】

プログラムコードを供給するための記憶媒体としては、例えば、フロッピーディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、磁気テープ、不揮発性のメモリカード、ROM等を用いることができる。

【0208】

また、コンピュータが読みだしたプログラムコードを実行することにより、各実施の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の指示に基づき、コンピュータ上で稼動しているOS等が実際の処理の一部又は全部を行い、その処理によって各実施の形態の機能が実現される場合も含まれることは言うまでもない。

20

【0209】

さらに、記憶媒体から読み出されたプログラムコードが、コンピュータに挿入された拡張機能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書き込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPUなどが実際の処理の一部又は全部を行い、その処理によって各実施の形態の機能が実現される場合も含まれることは言うまでもない。

【0210】

30

【発明の効果】

以上説明したように本発明によれば、ホストデバイスとターゲットデバイス間でデータ通信を行う際の通信プロトコルが、そのターゲットデバイスによって限定されることがない、拡張性の高いデータ処理方法、データ処理装置、プリンタ、及び記憶媒体を提供することができる。特に、複数種類のターゲットデバイス（プリンタやイメージ入力装置等）のプロトコルに対応可能であるので、拡張性が極めて高い。

【0211】

また、同一ネットワークに接続されたターゲットデバイスを調査することで、目的とするターゲットデバイスを選択することができるため、最適な出力（プリンタ出力等）を得ることができる。特に、プリンタ出力するのに適切なプリンタ（第1の種類のターゲットデバイス）を選択することができると共に、該プリンタに適切なイメージ入力装置（第1の種類のデバイスをホストデバイスとする第2の種類のターゲットデバイス）も選択することができるため、極めて良好なプリンタ出力を得ることができる。

40

【0212】

また、多くのプロトコルが使用可能な状況下であっても、それらのプロトコルのうちターゲットデバイスから最も多くサポートされているプロトコルを決定することで、実際に使用するプロトコルの種類を削減することができるため、ホストデバイスのプロトコルに切り替えによる負荷を下げるることができる。

【図面の簡単な説明】

【図1】IEEE1394シリアルインターフェースを用いて構成されるネットワークシ

50

システムの一例を示すブロック図である。

【図2】IEEE 1394シリアルインターフェースの構成を説明するための図である。

【図3】IEEE 1394シリアルインターフェースにおけるアドレス空間を説明するための図である。

【図4】IEEE 1394シリアルインターフェース用のケーブルの断面を説明するための図である。

【図5】DS-Link方式を説明するための図である。

【図6】IEEE 1394シリアルインターフェースにおけるネットワーク構築手順を説明するためのフローチャートである。

【図7】ルートの決定方法を説明するためのフローチャートである。

10

【図8】親子関係決定から全てのノードIDの設定までの手順を説明するためのフローチャートである。

【図9】ネットワークの一例を説明するための図である。

【図10】バスアービトレーションを説明するための図である。

【図11】アービトレーションの手順を説明するためのフローチャートである。

【図12】アシンクロナス転送における時間的な遷移状態を説明するための図である。

【図13】アシンクロナス転送用のパケットフォーマットを説明するための図である。

【図14】アイソクロナス転送における時間的な遷移状態を説明するための図である。

【図15】アイソクロナス転送用のパケットフォーマットを説明するための図である。

【図16】アシンクロナス転送とアイソクロナス転送が混在する場合のバス上の転送状態の時間的な遷移の様子を説明するための図である。

20

【図17】IEEE 1394シリアルインターフェースとOSIモデルの対比を説明するための図である。

【図18】LOGINプロトコルの基本動作を説明するための図である。

【図19】第1の実施の形態において、IEEE 1394シリアルインターフェースにおける接続形態を説明するための図である。

【図20】ログイン動作の流れを説明するための図である。

【図21】LOGINプロトコルのためにプリンタが備えるCSRを説明するための図である。

【図22】ホストデバイスで作成されるプリンタマップを説明するための図である。

30

【図23】CRCアーキテクチャにおけるノードIDを説明するための図である。

【図24】プリンタマップ作成のためのコマンドフォーマットを説明するための図である。

【図25】ホストデバイスで実行されるプリンタマップ作成処理を説明するためのフローチャートである。

【図26】上記プリンタマップ作成処理時にプリンタで実行される処理を説明するためのフローチャートである。

【図27】ホストデバイスにおけるLOGIN処理を説明するためのフローチャートである。

【図28】プリンタ(ターゲットデバイス)におけるLOGIN処理を説明するためのフローチャートである。

40

【図29】第2の実施の形態において、ホストデバイスで実行されるプリンタマップ作成処理を説明するためのフローチャートである。

【図30】上記プリンタマップ作成処理時にプリンタで実行される処理を説明するためのフローチャートである。

【図31】第3の実施の形態において、システム動作を説明するための図である。

【図32】OSIモデルとの対比を説明するための図である。

【図33】第4の実施の形態において、プリンタ(ホストデバイス)で実行されるイメージ入力装置についてのマップ作成処理を説明するためのフローチャートである。

【図34】上記マップ作成処理時にイメージ入力装置で実行される処理を説明するための

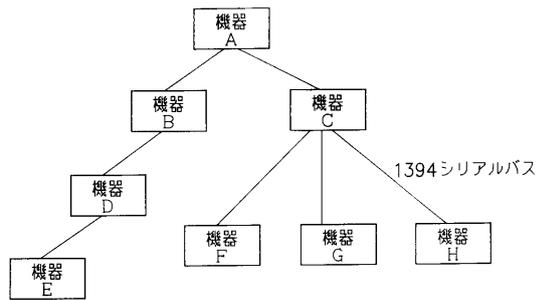
50

フローチャートである。

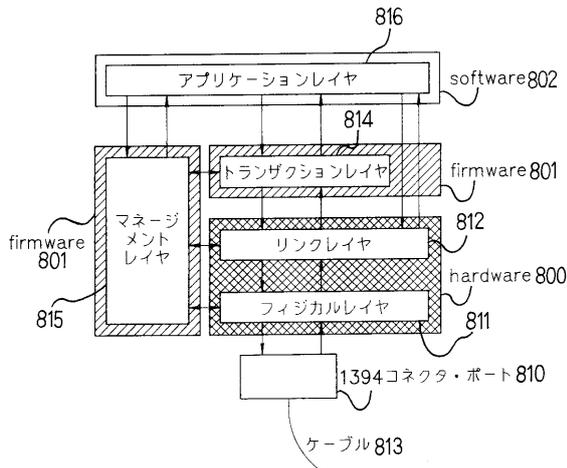
【符号の説明】

- 1 O S Iモデルの物理層
- 2 データリンク層
- 3 上位層
- 4 上位層
- 5 トラnsポートプロトコル層
- 6 プレゼンテーション層
- 7 LOG INプロトコル
- 8 シリアルバスプロトコル (S B P - 2)
- 9 デバイスプロトコル
- 1 1 プリンタ
- 1 2 P C
- 1 3 スキャナ
- 1 4 V C R

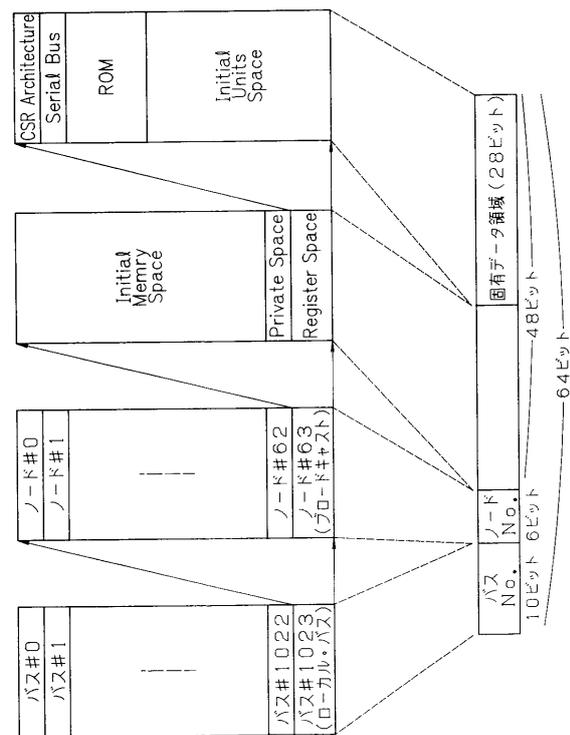
【図1】



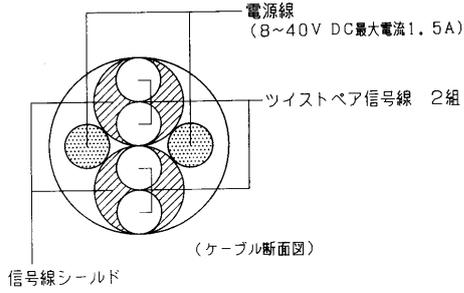
【図2】



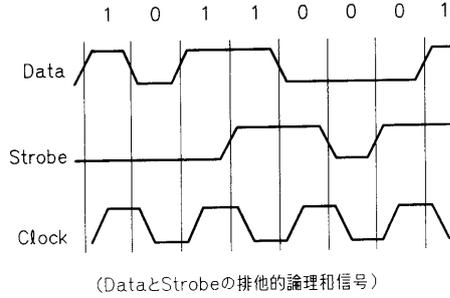
【図3】



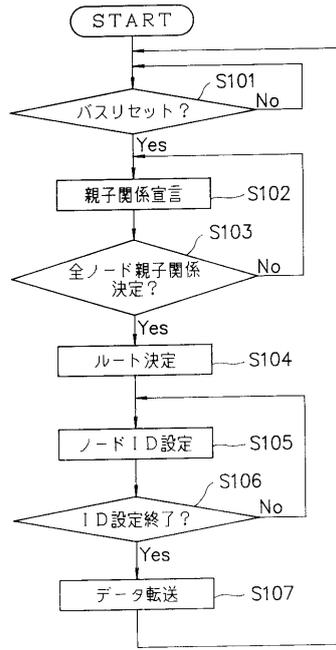
【図4】



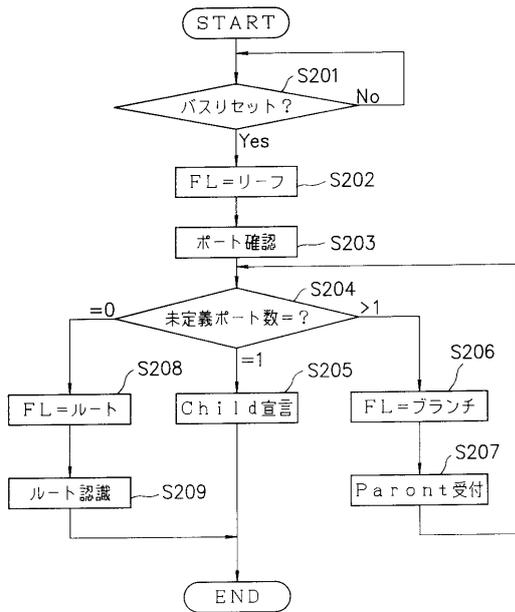
【図5】



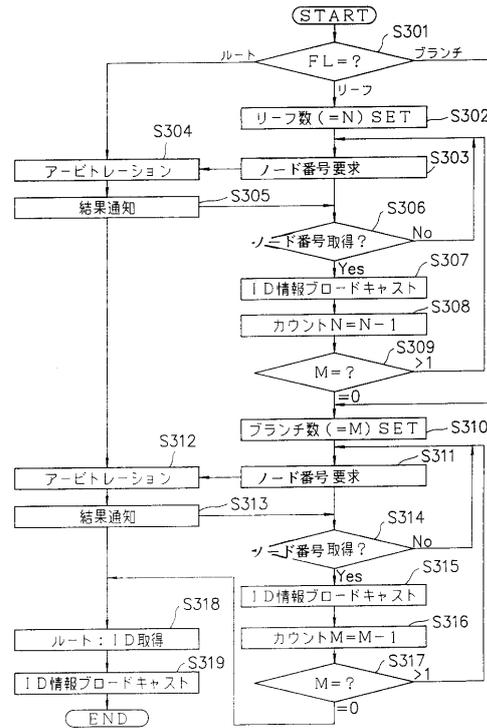
【図6】



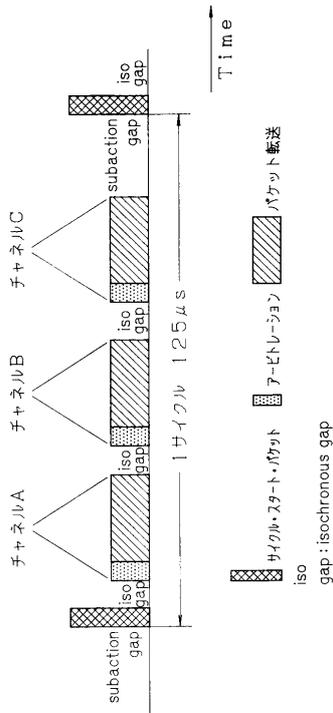
【図7】



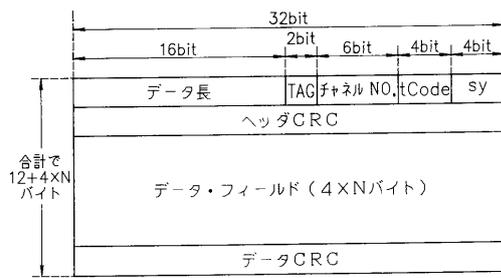
【図8】



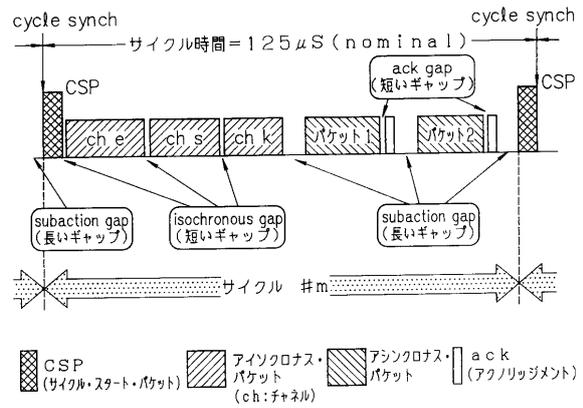
【 図 1 4 】



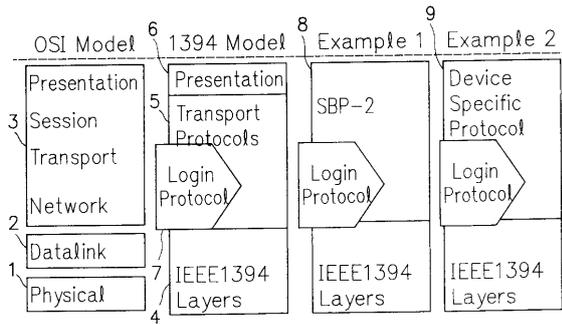
【 図 1 5 】



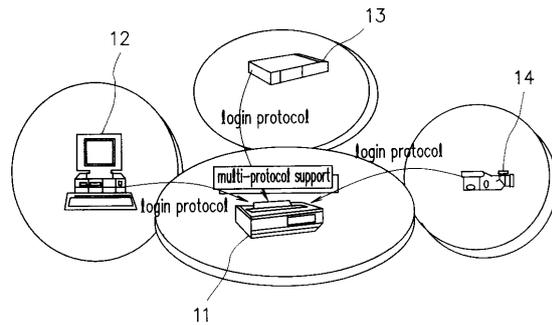
【 図 1 6 】



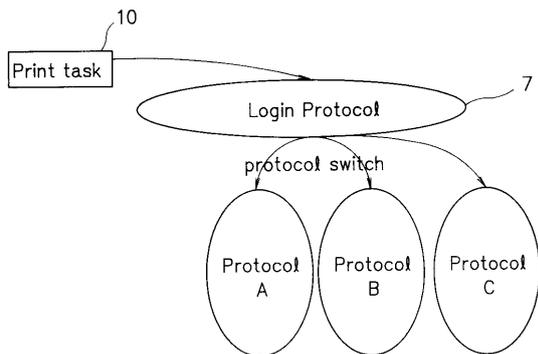
【 図 1 7 】



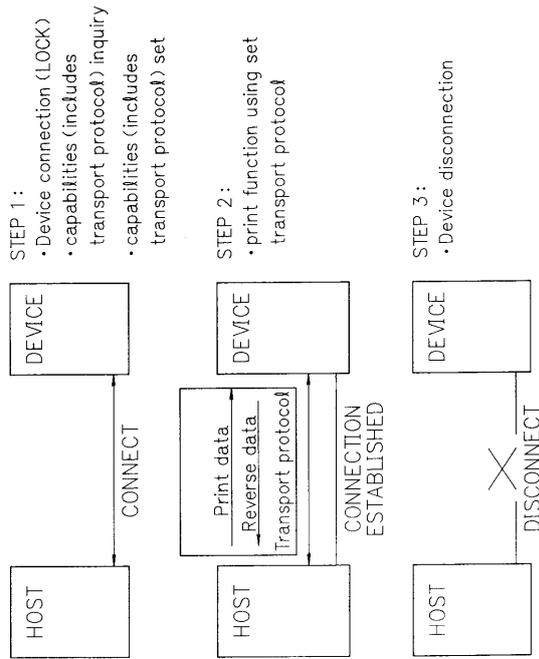
【 図 1 9 】



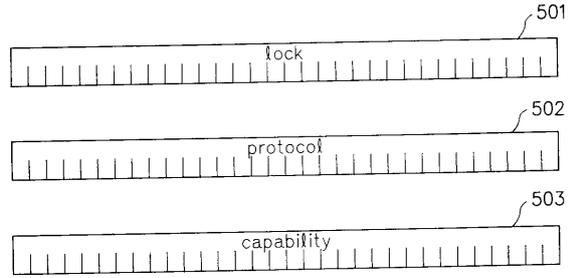
【 図 1 8 】



【図20】



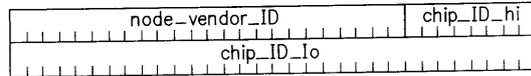
【図21】



【図22】

Unique ID	Node ID	Status	Capability

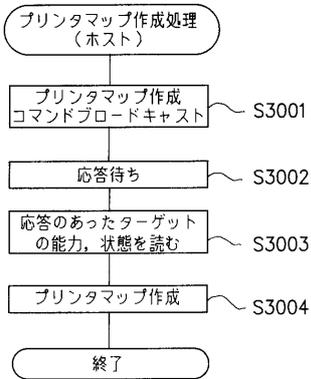
【図23】



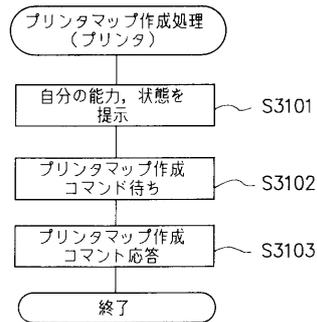
【図24】



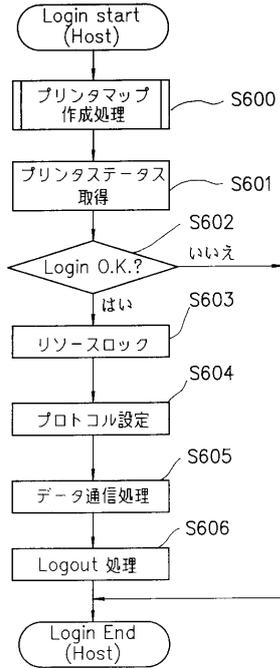
【図25】



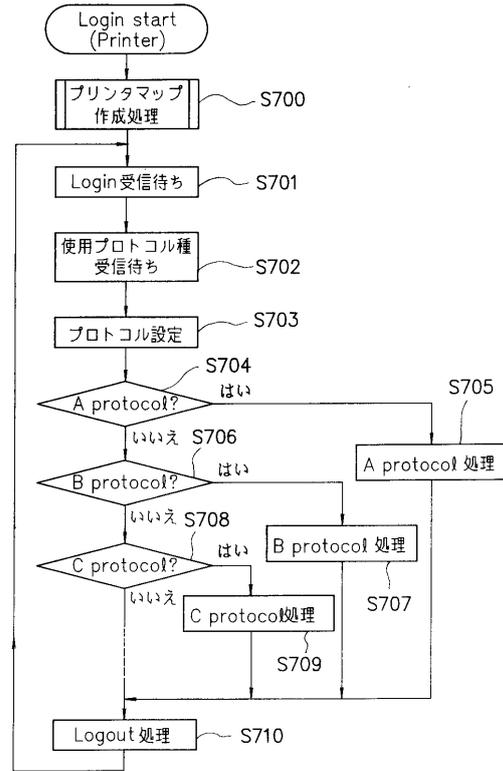
【図26】



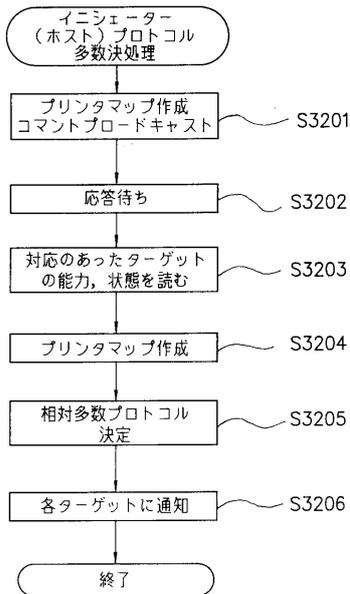
【図 27】



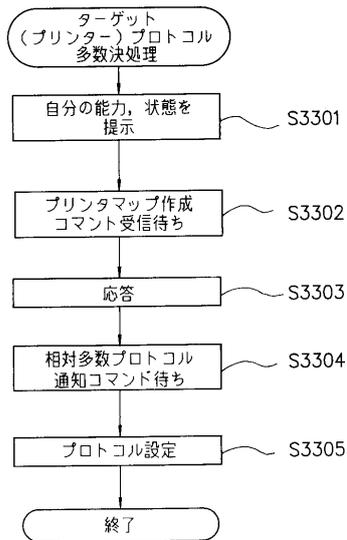
【図 28】



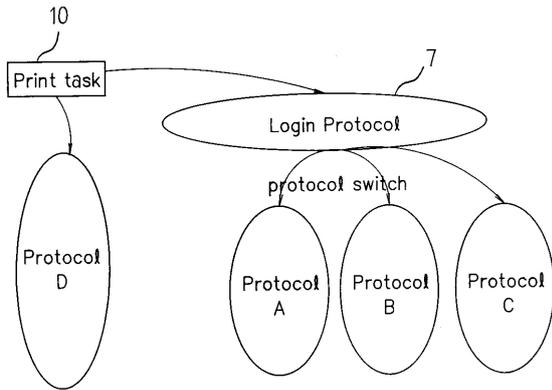
【図 29】



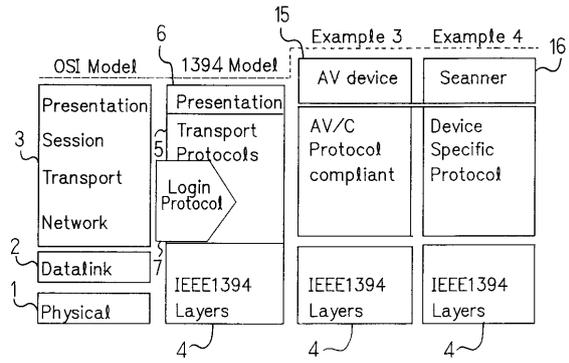
【図 30】



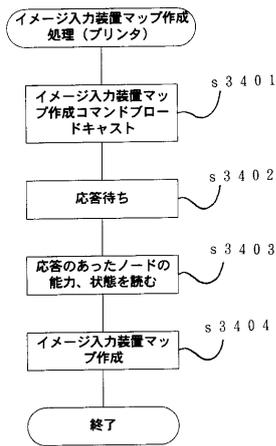
【図 3 1】



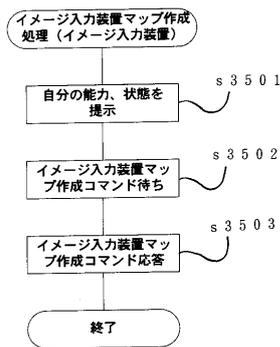
【図 3 2】



【図 3 3】



【図 3 4】



フロントページの続き

審査官 鈴木 匡明

- (56)参考文献 特開平02 - 206264 (JP, A)
特開平08 - 065325 (JP, A)
特表平05 - 506556 (JP, A)
特開昭62 - 049741 (JP, A)
特開昭62 - 248053 (JP, A)
特開平03 - 088539 (JP, A)
特開平02 - 090840 (JP, A)

(58)調査した分野(Int.Cl., DB名)

G06F 3/12

G06F 13/00~13/42

B41J 29/00~29/70