

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第4709800号
(P4709800)

(45) 発行日 平成23年6月22日 (2011. 6. 22)

(24) 登録日 平成23年3月25日 (2011. 3. 25)

(51) Int. Cl.

H04L 12/403 (2006.01)

F I

H04L 12/403

請求項の数 4 (全 37 頁)

(21) 出願番号	特願2007-113395 (P2007-113395)	(73) 特許権者	500520743
(22) 出願日	平成19年4月23日 (2007. 4. 23)		ザ・ボーイング・カンパニー
(62) 分割の表示	特願2002-548925 (P2002-548925) の分割		The Boeing Company
原出願日	平成13年11月5日 (2001. 11. 5)		アメリカ合衆国、60606-1596
(65) 公開番号	特開2007-288791 (P2007-288791A)		イリノイ州、シカゴ、ノース・リバーサイド・プラザ、100
(43) 公開日	平成19年11月1日 (2007. 11. 1)	(74) 代理人	100109726
審査請求日	平成19年5月22日 (2007. 5. 22)		弁理士 園田 吉隆
(31) 優先権主張番号	60/254, 136	(74) 代理人	100101199
(32) 優先日	平成12年12月8日 (2000. 12. 8)		弁理士 小林 義敦
(33) 優先権主張国	米国 (US)	(72) 発明者	エラーブロック、フィリップ・ジェイ
(31) 優先権主張番号	09/735, 146		アメリカ合衆国、63376 ミズーリ州
(32) 優先日	平成12年12月12日 (2000. 12. 12)		、セント・ピーターズ、バッキントン・ストリート、351
(33) 優先権主張国	米国 (US)		

最終頁に続く

(54) 【発明の名称】 ネットワーク経由でデータチャネルをコントローラにデジタル方式でインターフェイスするためのネットワーク・デバイス・インターフェイス

(57) 【特許請求の範囲】

【請求項 1】

共通デジタルバスを介して、バスコントローラとセンサデータチャネルとアクチュエータデータチャネルとからなる複数のデータチャネルとの間の通信を容易にするためのシステムであって、

前記共通デジタルバスに接続されたバスコントローラと、前記共通デジタルバスと前記複数のデータチャネルとの間に接続されたネットワークデバイスインターフェイスとを備え、

前記バスコントローラは、前記ネットワークデバイスインターフェイスに、予め定められたビットレートで第1のメッセージを送信し、

前記ネットワークデバイスインターフェイスは、同期クロック信号に依存しない前記第1のメッセージを前記バスコントローラから受信すると、前記第1のメッセージが送信された、前記予め定められたビットレートを判断し、

前記ネットワークデバイスインターフェイスは、前記第1のメッセージに応答して、前記バスコントローラに対して、同一の予め定められたビットレートで第2のメッセージを送信し、

前記バスコントローラは、前記予め定められたビットレートを変更し、

前記バスコントローラと前記ネットワークデバイスインターフェイスとは、前記第1のメッセージの送信と、前記予め定められたビットレートの判断と、前記変更されたビットレートでの第2のメッセージの送信とを繰り返し、前記バスコントローラは、前記ネット

10

20

ワークデバイスインターフェイスに対して、次の第 1 のメッセージが送信されるビットレートを決定するポー選択コマンドを送信するシステム。

【請求項 2】

前記バスコントローラは、前記同一の予め定められたビットレートで、同期クロック信号に依存せずに、前記第 2 のメッセージを受信する、請求項 1 に記載のシステム。

【請求項 3】

前記バスコントローラは、前記予め定められたビットレートの変更の後に、かつ、前記バスコントローラによる前記第 1 のメッセージの送信の前に、前記ネットワークデバイスインターフェイスに対して、変更されたビットレートで、例としてのメッセージを送信し、

10

前記ネットワークデバイスインターフェイスによる前記例としてのメッセージが受信されると、前記変更されたビットレートが判断され得る、請求項 1 に記載のシステム。

【請求項 4】

前記ネットワークデバイスインターフェイスは、それに関連付けられる前記データチャネルと通信するために、前記予め定められたビットレートを使用する、請求項 1 に記載のシステム。

【発明の詳細な説明】

【技術分野】

【0001】

[連邦政府の支援を受けた研究または開発]

20

この発明は、NASA によって与えられた協力協定番号第 NCCW - 0076 号のもとに政府の支援を受けて行なわれた。政府は、この発明に或る権利を有する。

【0002】

[関連出願への相互参照]

この出願は、2000 年 12 月 8 日出願の「ネットワーク経由でデータチャネルをコントローラにデジタル方式でインターフェイスするためのネットワーク・デバイス・インターフェイス (NETWORK DEVICE INTERFACE FOR DIGITALLY INTERFACING DATA CHANNELS TO A CONTROLLER VIA A NETWORK)」と題される米国仮特許出願連続番号第 60 / 254, 136 号からの優先権を主張し、その内容をここに引用により援用する。

【0003】

30

この発明は一般に、ネットワーク・デバイス・インターフェイスに関し、特に、共通のネットワークバス上でデータチャネルをコントローラにデジタル方式でインターフェイスするための装置および方法に関する。

【背景技術】

【0004】

今日多くの産業において、システム故障の可能性、または関心のあるオブジェクトに対する環境およびその他の外部的な力の影響の評価を行なうためにモニタリングシステムが使用されている。たとえば、航空電子業界では、モニタリングシステムは、航空機上のさまざまな重要な構造的な場所における、歪み、加速度、圧力、腐食および温度などのパラメータを監視するために用いられている。同様に、そのようなモニタリングシステムは、オン / オフ乗員制御から動力伝達経路制御およびマルチメディアシステムに至るまで、あらゆるものを制御および監視するために自動車業界でも使用することができる。

40

【0005】

これら従来のモニタリングシステムの多くは、重要な場所で監視されているオブジェクトのまわりに位置付けられる、センサ、アクチュエータおよびサブシステムなどの複数の遠隔デバイスを使用する。さらに、これら従来のモニタリングシステムの多くは、遠隔デバイスからのデータを受取りかつ遠隔デバイスにコマンドを送信するために、遠隔デバイスの各々に接続される 1 つまたはいくつかのコントローラを含む。作動中、コントローラはさまざまなセンサからデータを取得する。コントローラは、オブジェクト上で機能を実現するためにアクチュエータの作動も行なう。

50

【発明の開示】

【発明が解決しようとする課題】

【0006】

これら従来のモニタリングシステムは、オブジェクトの重要な構造を監視するための方法を提供するが、いくつか欠点がある。たとえば、従来のモニタリングシステムの多くは、遠隔デバイスの各々をコントローラに接続するために専用の配線および信号調整を使用する。さらに、通常、遠隔デバイスの多くはアナログ方式であり、通常、遠隔デバイスから受取られるデータはアナログの形である。

【0007】

航空電子業界および自動車業界を含め、今日多くの業界では、ネットワークの複雑さによって、いくつかの理由のため、従来のモニタリングシステムは非実用的なものになり得る。特に、専用の配線および信号調整は、高価で、かさばり、重く、しかも設置および維持が難しくなり得る。これは、重量およびサイズを第1に考慮する航空機での応用において、特に重大な意味を持つ。さらに、自動車業界では、配線の追加は自動車の重量とコストを増やし得る。

【0008】

さらに上述のように、多くの従来のモデリングシステムは、アナログフォーマットでデータを伝送する。通常、アナログ信号は、データ伝送中に信号に取込まれる雑音の影響を受けやすい。伝送される信号の多くは、開始の振幅が低いいため、この雑音は、信号を劣化させ、信号の解像度の損失を引起こす信号対雑音比を低下させる。さらに、これら遠隔デバイスの多くは、コントローラからかなりの距離の所に散在するため、遠隔デバイスをコントローラに接続する電気線が、配線内のDC抵抗による信号劣化を引起こすほどに長くなり得る。

【0009】

このことを考慮すると、専用の配線およびアナログ伝送を共通のバスで置き換え、データのデジタル伝送を使用することが有利であろう。しかし、多くの従来のデジタルネットワークには、それ自身にさまざまな問題がある。たとえば、多くの既存のデジタルネットワークは、プロセッサを必要とし、かつ受容できないほどの大きいまたは費用のかかる遠隔デバイスを強要する、複雑なプロトコルを必要とする。プロセッサベースの検出装置も、正確に、正しい時間にアナログデータをサンプリングする際、またはアクチュエータに動作を行なわせる際に問題を有し得る。また、複雑なプロトコルは、データ取得および制御に必要なオーバーヘッドをバス上のメッセージに取込む。このオーバーヘッドによって、バス上で伝送可能なデータサンプルの数が大きく制限されることがある。これらのネットワークには他にも問題がある。たとえば、通常、それらは、取得および制御の両方をサポートせず、通常、短いネットワークのみをサポートする。さらに、これら従来のネットワークは、大抵、かさばるネットワーク・デバイス・インターフェイスを有し、ネットワークデータレートは遅く、またはネットワーク装置数が少ない。これに加えて、デジタルネットワークを含む多くのコンピュータシステムは、時間で決定された態様で作動しない。一般に、これらコンピュータシステムは、何らかの正しいタイミングで繰返しまたは解釈され実行される、ネットワーク構成要素へのトリガコマンドをスケジュールする能力に欠ける。

【0010】

上述のことを考慮すると、ネットワーク構成要素が、安価、単純かつ高速でありながら堅牢なネットワーク線上でデジタル形式で通信するのを可能にし、プロトコルが単純でオーバーヘッドメッセージが少なく、構成要素のサイズが小さく、しかも配線数の少ない、ネットワークシステムを提供すれば有利であろう。さらに、ネットワークシステムが、ネットワーク・デバイス用のマイクロコントローラまたはプロセッサを使用せずに作動すれば有利であろう。また、このネットワークシステムは、取得および制御の両方をサポートし、ネットワークに接続された複数の構成要素から同時にデータを取得または変換することができる。さらに、このネットワークシステムは、多くの構成要素、より長いネットワ

10

20

30

40

50

ーク線を可能にし、正確な態様で時間決定性を保証する。

【課題を解決するための手段】

【 0 0 1 1 】

〔 発 明 の 概 要 〕

理解を促し、この明細書での混乱を避けるため、ここでネットワークオブジェクトについて簡単に定義することが必要である。定義すべき第1のネットワークオブジェクトはバスコントローラである。バスコントローラは、ネットワークバス上でコマンドを送信するネットワーク・デバイスである。ネットワーク上の他のすべてのデバイスは、バスコントローラを聴き、バスコントローラのコマンドに基づいて動作を行なう。ネットワーク上の、バスコントローラでない装置はどれもネットワーク・デバイスである。ネットワーク・デバイスは、この開示を通じて、しばしば遠隔デバイスと称される。ネットワーク・デバイス・インターフェイス (Network Device Interface ; N D I) は、ネットワーク・デバイスの構成要素である。N D I は、バスコントローラおよびバス上の他のトラフィックを聞き、ネットワークバス上のトラフィックに応じて、動作を行なうか、またはネットワーク・デバイスに動作を行なわせる。大抵のN D I は、少なくとも1つまたは複数のデータチャンネルに接続される。データチャンネルは、センサ、アクチュエータ、センサおよび信号調整、アクチュエータおよび信号調整、またはその他アナログまたはデジタルシステムである。データチャンネルは、ネットワーク・デバイスの構成要素であるか、またはネットワーク・デバイスに接続される。

【 0 0 1 2 】

後に詳しく記載するように、この発明は、センサ、アクチュエータおよびサブシステムなどのさまざまなデータチャンネルを共通のコントローラに接続して、データチャンネルおよびコントローラからの、ならびにデータチャンネルおよびコントローラへの、コマンドならびにデータの伝送を行なうためのネットワーク・デバイス・インターフェイス (N D I) を提供することによって、これらの問題およびその他の問題を解決する。重要なことは、この発明のN D I デバイスは、共通のネットワークを介してさまざまなデータチャンネルをコントローラに接続することであり、このことによって、さまざまなデータチャンネルが同じ配線を共有してコントローラと通信可能となることである。さらに、この発明のN D I は、アナログ - デジタル、またはデジタル - アナログ、もしくはその他であり得る、異なる種類のデータチャンネルをインターフェイスすることができる。センサはアナログ - デジタルのデータチャンネルとしてN D I に接続され、アクチュエータはデジタル - アナログのデータチャンネルとしてN D I に接続される。この発明のN D I は、アナログ - デジタルのチャンネルからデジタルデータを取り、適切なプロトコルに従ってそれをフォーマットし、適切なプロトコルに従ってそれをネットワーク上で伝送することができる。この発明のN D I は、ネットワークからデジタルデータを取り、それをデジタルデータとしてデジタル - アナログのコンバータ (D / A) に与え、D / A にそのデータをアナログ信号に変換させることもできる。N D I がそのデータチャンネルから直接アナログ信号を受取り、またはそのデータチャンネルに直接アナログ信号を生成する、N D I の他の実施例も可能である。デジタルフォーマットでネットワーク上でデータを伝送することにより、コマンドおよびデータは、雑音および劣化の影響を受けにくい。

【 0 0 1 3 】

さらに、この発明のN D I デバイスは、ネットワーク上で一度に1つまたはいくつかのネットワーク・デバイスとコントローラが通信することを可能にするデータプロトコルとともに作動する。重要なのは、この発明のN D I デバイスによって使用されるデータプロトコルは、固定された低レベル命令セットを有することである。プロトコルが単純なため、この発明のN D I デバイスは、高レベルプロセッサである必要はない。その代わりに、好ましい一実施例では、この発明のN D I のデバイスは、特定用途向け集積回路 (A S I C) として実現されるステートマシンである。マイクロコントローラまたはプロセッサの代わりにステートマシンを使用してN D I デバイスを実現することの利点は、多くのプロセスを同時に行なえることであり、このことは、N D I デバイスが時間決定的かつ高速と

なるのを支援する。

【 0 0 1 4 】

有利には、一実施例において、この発明は、同期通信または非同期通信のいずれかを使用して、コントローラとデータチャネル間でコマンドおよびデータを通信可能なネットワーク・デバイス・インターフェイスを提供する。この実施例では、NDIデバイスは、共通のデジタルバスを介してコントローラからメッセージを受取るための受信機を有する。この実施例のNDIデバイスは、受信機によって受取られたメッセージに応答して、関連づけられるデータチャネルにコマンドを与えかつ関連づけられるデータチャネルからデータを受取るための、インターフェイスをさらに含む。さらに、NDIデバイスは、共通のデジタルバスを介してコントローラにメッセージを送送するための送信機も含む。重要な

10

【 0 0 1 5 】

作動中、コントローラからデータが受取られると、この発明のNDIデバイスのクロック検出器は、クロック信号がコントローラからのデータに付随するかを判断する。もしクロック信号が存在すれば、コントローラは同期モードで通信している。この場合、NDIデバイスは、クロック信号を使用して、データチャネルにコマンドおよびデータを与え、データチャネルからデータを受取る。さらに、この発明のNDIデバイスの送信機は、バスクロック信号を使用してデータをコントローラに伝送する。

【 0 0 1 6 】

しかしながら、NDIデバイスのクロック検出器がコントローラからのデータに関連づけられるクロック信号を検出しない場合、NDIデバイスは、バスコントローラが非同期モードで作動していると判断する。この場合、NDIデバイスは、バスクロックから独立して、非同期モードでデータをデータチャネルに与え、データチャネルからデータを受取る。さらに、この発明のNDIデバイスの送信機は、同期モードのバスクロックから独立して、非同期でコントローラにデータを伝送する。

20

【 0 0 1 7 】

一実施例において、コントローラは、共通のクロックバスを介して同期信号を与える。この場合、この発明のクロック検出器は、同期クロック信号を受取り、信号を分析して、それがデータビットと同じレートで送信されているかを判断する。もしそうであれば、ネットワーク・デバイス・インターフェイスのクロック検出器は、コントローラが同期モードで作動していると判断する。

30

【 0 0 1 8 】

さらに、いくつかの実施例では、この発明のネットワーク・デバイス・インターフェイスは、共通のデジタルバスに接続されるビットレート検出器をさらに含んでもよい。この実施例では、コントローラが非同期モードで作動している場合、コントローラは或るビットレートでコマンドおよびデータを伝送している。この発明のビットレート検出器はビットレートを検出し、NDIデバイスはこのビットレートを使用してデータチャネルにコマンドおよびデータを送信し、データチャネルからデータを受取る。さらにこの発明の送信機は、検出されたビットレートを使用してコントローラにデータを送り返す。

40

【 0 0 1 9 】

規定されたビットレートで非同期モードを使用してデータを伝送することに加え、コントローラは、通信中にビットレートを変更することもできる。この実施例では、まず、コントローラは、或る予め定められたビットレートで第1のメッセージをデータチャネルに送ることができる。この実施例では、クロック検出器は同期クロック信号を検出しないが、ビットレート検出器は、メッセージがコントローラによって伝送される第1のビットレートを検出する。この第1のビットレートに基づいて、この発明のネットワーク・デバイス・インターフェイスは、このビットレートを使用して、コマンドおよびデータをデータチャネルに送り、データチャネルからデータを受取る。さらに、この発明の送信機は、検出された第1のビットレートを使用してデータをコントローラに送り返す。

50

【 0 0 2 0 】

第1のメッセージまたはいくつかのメッセージが第1のビットレートで送られた後、コントローラは、ビットレートを変更して第2のメッセージを第2のビットレートで送ることができる。この実施例では、NDIデバイスのビットレート検出器は、メッセージがコントローラによって伝送される第2のビットレートを検出する。この第2のビットレートに基づいて、この発明のNDIデバイスは、このビットレートを使用して、バスコントローラからコマンドおよび引数を受取り、バスコントローラにデータを送り返す。

【0021】

一実施例では、コントローラは、コマンドおよびデータを送るために以前に使用されたビットレートから変更されたビットレートで、例としてのメッセージを送ってもよい。この実施例では、NDIデバイスのビットレート検出器は、ビットレートの変化を検出し、NDIデバイスの送信機は、新しいビットレートでコントローラにデータを送り返し、ビットレートが変更されたことを示す。さらに、別の実施例では、コントローラは、メッセージが伝送されるべきビットレートを規定するボー選択コマンドを送ることもできる。

【0022】

上述のように、この発明のNDIデバイスはプロトコルとともに作動する。一実施例では、プロトコルは、複数の異なるアドレスを使用して、同時に1つまたはいくつかのデータチャネルをアドレス指定する。たとえば、この発明の一実施例では、プロトコルは、論理アドレスを使用して個々のデータチャネルをアドレス指定し、グループアドレスを使用していくつかのデータチャネルをアドレス指定し、グローバルアドレスを使用して同時にすべてのデータチャネルをアドレス指定する。この実施例では、論理マスクおよびグループマスクは、この発明のNDIデバイスに関連づけられる記憶装置に記憶される。グループマスクは、NDIデバイスが各チャネルのためのグループアドレスのリストを記憶するのに効率のよい方法である。グループマスクは、各データチャネルについての複数のビットを含むものと解釈される。マスクの各ビットは、それぞれのグループに関連づけられ、それぞれのデータチャネルがグループのメンバーであることを示す第1の状態、およびそれぞれのデータチャネルがグループのメンバーでないことを示す第2の状態を有する。マスクは、ネットワーク・デバイス・インターフェイスに関連づけられるメモリにも記憶される。

【0023】

この実施例では、コマンドまたはデータがいつ送られようと、それは論理アドレス、グループアドレスまたはグローバルアドレスのどれかを含む。送られる各コマンドまたはデータメッセージについて、そのメッセージに関連づけられるアドレスがNDIデバイスによって分析される。アドレスがグローバルであれば、NDIデバイスはそのコマンドを実施する。同様に、アドレスが論理アドレスであり、NDIデバイスに関連づけられるデータチャネルの論理アドレスに対応する場合、NDIデバイスはコマンドを実施する。アドレスがグループアドレスである場合、この発明のNDIデバイスは、それぞれのデータチャネルに関連づけられるマスクを分析することによって、NDIデバイスに関連づけられるデータチャネルがそのグループアドレスによって規定されるグループのメンバーであるかを判断する。ネットワーク・デバイス・インターフェイスは、データチャネルがそのグループアドレスを有するグループのメンバーである場合のみ、コマンドを実施する。

【0024】

上述のように、この発明のNDIデバイスは、同期モードまたは非同期モードのどちらでも作動可能である。さらに、コントローラは、グループアドレスを与えて複数のデータチャネルに同時にコマンドを送ることができる。しかしながら、各データチャネルに接続されるNDIデバイスが非同期モードで作動している場合、NDIデバイスを同期させて、グループアドレスに関連づけられるコマンドを、NDIデバイスに接続されるそれぞれのデータチャネルにNDIデバイスが同時に適用するようにすることが困難だという点において、問題が発生する。このことを考慮して、一実施例では、NDIデバイスは、それらが非同期モードで作動している場合でも同期させることができる。

【0025】

特に、この発明の一実施例では、コントローラはコマンドを複数のデータチャンネルに伝送し、この場合、コマンドは、第1の状態と第2の状態との間の遷移によって規定される値を有する複数のビットを含む。この実施例では、この発明のNDIデバイスの各々は、複数のネットワーク・デバイス・インターフェイスが時間で決定される形でコマンドを同時に実行するように、コマンドのそれぞれのビットの値を規定する遷移に比例して予め定められた同じ時にコマンドの実施を開始する。さらに一実施例では、コントローラは、同期部、メッセージ本体およびパリティビットを含むコマンドを伝送する。この実施例では、この発明のNDIデバイスは、パリティビットの値を規定する遷移と同時にコマンドの実行を開始する。さらに別の実施例では、コントローラは、スタートビットと、コマンド領域と、0にセットされた未使用の最後のビットを有するアドレス領域と、1にセットされたストップビットとを含むコマンドを伝送する。この実施例では、この発明のNDIデバイスは、アドレス領域の未使用ビットからストップビットへの遷移と同時に、各データチャンネルでの機能の実現を開始する。

10

【0026】

さらに上述のように、同期モードでは、コントローラは同期クロック信号を共通のデジタルバス上で伝送する。この実施例では、この発明のNDIデバイスは、同期クロック信号を使用することによって、いくつかのネットワーク・デバイス・インターフェイス間で同期してコマンドを実行することも可能にし得る。具体的には、この実施例では、コントローラは、複数のデータチャンネルにコマンドを伝送し、メッセージは、第1の状態と第2の状態との間の遷移によって規定される値を有する複数のビットを含む。さらに、コントローラは、メッセージと同時に、複数のクロックパルスを含む同期クロック信号をコントローラから複数のデータチャンネルへ伝送する。この実施例では、複数のネットワーク・デバイス・インターフェイスは、それぞれのクロックパルスによって規定される予め定められた同じときにコマンドの実行を開始し、それぞれのクロックパルスは、メッセージのそれぞれのビットに対する予め定められた関係に基づいて規定される。

20

【0027】

たとえば、一実施例において、ネットワーク・デバイス・インターフェイスは、メッセージのそれぞれのビットに続く第1のクロックパルスに同期して、実行を開始する。具体的には、一実施例では、伝送されるメッセージは、第1の状態と第2の状態との間の遷移によって規定される値を有する複数のビットを有し、メッセージは、同期部、メッセージ本体およびパリティビットを規定する。この実施例では、ネットワーク・デバイス・インターフェイスは、それぞれのクロックパルスによって規定される予め定められた同じときにコマンドの実行を開始し、それぞれのクロックパルスは、メッセージのパリティビットの値を規定する遷移に対する予め定められた関係に基づいて規定される。

30

【0028】

上述のように、この発明のNDIデバイスは、簡略化したコントローラおよびネットワーク・デバイス・インターフェイスをネットワーク・デバイス上で使用することによって可能にする固定された低レベル命令セットを有するプロトコルとともに作動する。具体的には、一実施例では、この発明は、コンピュータ読取可能な記録媒体に記憶されるプロトコルを提供する。プロトコルは、共通のデジタルネットワークを通じてコントローラおよびネットワーク・デバイス・インターフェイス間でコマンドおよびデータを伝送するために使用される。重要なのは、プロトコルは、それぞれのコマンドおよび引数をデータチャンネルに送るために、コマンドおよびデータ構造のうちの少なくとも1つを含むことである。データ構造も、データチャンネルからバスコントローラにデータを送るために使用される。

40

【0029】

このことを考慮して、この発明は、ステートマシンを使用するシリアルな多重化された通信システムも提供する。特に、この発明の通信システムは、複数のコマンドを発行するためのコントローラ、およびコマンドに応答して予め規定された機能を実現するための複数のデータチャンネルを含む。コントローラとネットワーク・デバイス・インターフェイス

50

とを接続しているのは、それらの間の通信を支援するための共通のデジタルバスである。さらに、通信システムは、複数のネットワーク・デバイス・インターフェイスを含み、それらの1つは、各データチャンネルに関連づけられて、それぞれのデータチャンネルを共通のデジタルバスに相互接続する。この実施例では、各ネットワーク・デバイス・インターフェイスは、ステートマシンを含み、プロセッサから独立している。

【0030】

先に述べたように、同期モードでは、コントローラは、ネットワークバス上でネットワーク・デバイス・インターフェイスに同期クロック信号を与える。同期モードでは、同期クロックは、データを伝送するためのクロック信号として使用される。しかしながら、A / DコンバータおよびD / Aコンバータならびに信号調整デバイスの中には、同期クロックによってセットされるクロック速度で作動することができないものがある。このことを考慮して、この発明の一実施例では、ネットワーク・デバイス・インターフェイスはクロック分割器を含んでもよい。クロック分割器は、コントローラによって同期クロック信号を出力に接続されてもよいし、またはローカルの発振器に接続されてもよい。

【0031】

具体的には、この実施例では、バスコントローラがあり、1つの周波数で同期クロック信号およびコマンドをネットワーク・デバイスに送り、NDIデバイスは、バスコントローラからの同期クロックおよびデータを聞く。NDIデバイスは、バスコントローラからのデータおよびコマンドを、NDIデバイスに接続されるデータチャンネルによって期待される適切なフォーマットに変換し、同時にNDIデバイスは、分割された第2のクロック信号をデータチャンネルに与えて、コントローラおよびNDIデバイスが第1のクロック周波数と一致して作動して、コマンドおよびデータをネットワークバス上で送る一方で、データチャンネルが第2のクロック信号周波数と一致して作動しデータを変換するようにする。

【0032】

さらに別の実施例では、この発明のNDIデバイスは、各NDIデバイスに関連づけられる分割器を共通のデジタルバスに接続される他のすべてのNDIデバイスの分割器と同期させるための方法もさらに含む。特に、上述のように、いくつかの実施例では、コントローラは、コマンドが複数のデータチャンネル上で同時に実行されるように、グループアドレスを送る。しかしながら、これを行なうには、各NDIデバイスに配置される分割されたクロックを使用するデータチャンネルに設けられるクロック分割器のすべてを同期させなければならない。このことを考慮して、一実施例では、コントローラは共通のバスにわたって第1のクロック信号を送る。さらに、コントローラは、分割器の各々にそれらのそれぞれの第2のクロック信号の伝送を第1のクロック信号のエッジと同期させるように命令して、分割器の各々によって与えられる個々の第2のクロック信号が第1のクロック信号に対して同期し、このことによってコンバータの各々が同期するようにする。

【0033】

異なるクロック信号に従って作動することに加え、A / DおよびD / Aコンバータの中には、コントローラによって使用されるコマンドとは異なる特別なコマンドに従った作動も行なうものがある。このことを考慮して、この発明のネットワーク・デバイス・インターフェイスは、コンバータのための特別なコマンドを与えるために使用される特別な特徴を含み得る。この特徴はコマンド変換と呼ばれる。このため、作動中、この発明のNDIデバイスは、第1のセットのコマンドに従ってコントローラからコマンドおよびデータを受取り、第2のセットのコマンドに従ってそのコマンドを変換してコンバータに適用する。さらに、この発明のNDIデバイスは、第1のセットのコマンドに従って、コンバータから受取ったデータを共通のデジタルバスを通じて送ることもできる。

【0034】

NDIデバイスに好ましいプロトコルは、NDIデバイスの小型化を可能にするように、ネットワークデータビットのマンチェスター符号化を使用する。どのようなデバイスでも非同期シリアルデータを受取るには、シリアルデータストリームからデータシーケンス

10

20

30

40

50

のタイミングを取得できなければならないことを理解されたい。通常、シリアル非同期データの受信機は、その受信機を作動させ、シリアルデータからタイミング情報を回収するためにローカルの発振器を有しなければならない。一旦タイミング情報が抽出されると、ローカルの発振器の周波数があれば、非同期受信機は、或るレートでシリアルデータを受取ることができる。これらのレートから或る偏差分増加または減少したレートで受信することができる。シリアルデータのマンチェスター符号化によって、各ビットの中央でhighからlowへ、またはlowからhighへの遷移が起こる。このことによって、シリアルデータストリームから必要なタイミング情報を抽出することが容易になる。マンチェスター符号化されたシリアルデータストリームからタイミング情報を抽出することは容易であるため、予想されるデータレートからの比較的大きな偏差も、ローカルの発振器に基づいて許容され得る。予想されるデータレートからの比較的大きな偏差に対してこのように許容差があるため、各NDI受信機は、精度の低いローカルの発振器を使用してマンチェスター符号化されたデータを受取ることができる。精度の低いローカル発振器は、極めて小型化することができる。適切なローカル発振器の現在の実施例は、わずか約1×1.5ミリメートルである。このことはNDIデバイスの小型化に役立つ。

【発明を実施するための最良の形態】

【0035】

〔詳細な説明〕

ここからは、この発明の好ましい実施例が示される添付の図面を参照して、この発明をさらに十分に説明する。しかしながら、この発明は、多くの異なる形で実現可能であり、ここに示される実施例に限定されるものと解釈すべきではない。そうではなく、これらの実施例はこの開示を完璧かつ完全なものとし、当業者にこの発明の範囲を十分に伝えるために与えられるものである。同じ参照番号は、全体を通じて同じ要素を示す。

【0036】

上述のように、多くの従来のデータ取得および制御システムは、個々の配線を使用して、センサ、アクチュエータおよびサブシステムなどの遠隔デバイスを中央コントローラに接続し、データ取得および制御を行なっている。遠隔デバイスの数が多いため、これらのデバイスの各々のための個々の配線を合計すると、高価でかさばり、重く、しかも設置および維持が難しいものとなり得る。さらに、これら遠隔デバイスの多くは、アナログであるため、遠隔デバイスからの信号および遠隔デバイスへの信号は、雑音および信号の劣化の影響を受けやすい。

【0037】

後にさらに詳細に記載するように、この発明は、センサ、アクチュエータおよびサブシステムであり得るさまざまなデータチャネルを共通のコントローラに接続して、データチャネルおよびコントローラでコマンドならびにデータを伝送し合うためのネットワーク・デバイス・インターフェイス(NDI)を提供することによって、これらの問題および他の問題を解決する。重要なのは、この発明のNDIデバイスが共通のネットワークを介してさまざまな遠隔デバイスをコントローラに接続することであり、このことによって、さまざまな遠隔デバイスは同じ配線を共有してコントローラと通信することができる。さらに、この発明のNDIは、アナログからデジタル、またはデジタルからアナログまたはその他であり得る、異なる種類のデータチャネルにインターフェイスすることができる。センサは、アナログ・デジタル・データチャネルとしてNDIに接続され、アクチュエータは、デジタル・アナログ・データチャネルとしてNDIに接続される。この発明のNDIは、アナログ・デジタル・チャネルからデジタルデータを取り、適切なプロトコルに従ってそれをフォーマットし、プロトコルに従ってそれをネットワークに伝送することができる。この発明のNDIは、ネットワークからデジタルデータを取り、それをデジタルデータとしてデジタル・アナログ・コンバータ(D/A)に与え、D/Aにそのデータをアナログ信号に変換させることもできる。NDIの他の実施例は、そのデータチャネルにアナログ信号を直接生成し、またはそのデータチャネルからアナログ信号を直接受取ることでも可能である。デジタルフォーマットでネットワークでデータを伝送することにより、コマ

ンドおよびデータは、雑音および劣化の影響を受けにくい。

【 0 0 3 8 】

さらにこの発明のNDIは、コントローラがネットワーク上で1つまたはいくつかのネットワーク・デバイスと同時に通信することを可能にするデータプロトコルとともに作動する。重要なのは、この発明のNDIデバイスによって使用されるデータプロトコルは、固定された低レベル命令セットを有することである。プロトコルが単純であるため、この発明のNDIデバイスは高レベルプロセッサである必要はない。その代わりに、好ましい一実施例では、この発明のNDIデバイスは、特定用途向け集積回路(ASIC)として実現されるステートマシンである。マイクロコントローラまたはプロセッサの代わりにステートマシンを使用してNDIを実現する利点は、多くのプロセスを同時に行なえること

10

【 0 0 3 9 】

上述のように、この発明のNDIは、共通のコントローラと、共通のネットワークによってそのコントローラに接続されるさまざまなネットワーク・デバイスとの間のインターフェイスとして使用される。図1は、この発明のインターフェイスの実現の一実施例を示す。この図示は、この発明がより完全に理解されるようにするために与えられる。この発明は、この構成に限定されるのではなく、多くの異なるネットワークシステムで実現可能であることを理解されたい。NDIのこの実施例は、RHAMIS-HSプロトコルを使用するが、この開示によって企図される他の実施例は、他のプロトコルを使用し得る。

20

【 0 0 4 0 】

図1を参照すると、この発明が使用されるネットワークシステム30の一般的な実施例が示される。具体的には、このネットワークシステムは、高レベルプロセッサまたはパーソナルコンピュータなどのマスターコントローラ32を含み、これが、ネットワーク内の所望の点に位置するセンサ、アクチュエータおよびサブシステムなどのデータチャネル34からのデータを処理し、コマンドおよびデータをデータチャネル34に送る。重要なのは、ネットワークシステムがマスターコントローラ32とネットワークバス38との間に接続されるネットワークコントローラ36、およびネットワークバスとデータチャネルとの間に接続される1つまたはいくつかのNDIデバイス40をさらに含むことである。ネットワークコントローラおよびNDIデバイスをネットワークバスに接続するのは、それぞ

30

【 0 0 4 1 】

通常動作において、センサである遠隔デバイスは、検討中の特定のオブジェクトに接続され、温度、歪み、屈曲、速度、空気の流れ等のそのオブジェクトの特性を検知する。さらに、アクチュエータである遠隔デバイスは、機械的な部材およびその他の構造体に接続されて、テスト中のオブジェクト上で作動する。1つまたはいくつかの遠隔デバイスは、コンバータおよび信号調整デバイスを含む個々のデータチャネルを介して、この発明の1つのNDIデバイスに接続される。さらに、データおよびコマンドをネットワーク上でさまざまなネットワーク・デバイスに送信するように、マスターコントローラまたはネットワークコントローラのどちらかを構成することができる。これらコントローラの両方がそのような動作が可能であるため、バスコントローラという総称は、以下の議論において、マスターコントローラまたはネットワークコントローラのどちらかによって行なわれ得る動作を示すのに使用される。

40

【 0 0 4 2 】

図2を参照すると、センサからデータを取得するため、またはアクチュエータを作動させるため、コントローラは、ネットワークを通じて、コマンドおよびデータを遠隔デバイスにデジタル方式で送り、コマンドおよびデータは、遠隔デバイス上のデータチャネルの

50

1つまたはグループのために設計されている（ステップ200を参照）。コマンドおよびデータは、データプロトコルを使用して、ネットワーク上で伝送される。この発明のNDIデバイスは、データプロトコルの構造を使用して、データおよびコマンドを受取り、解釈する（ステップ210を参照）。さらに、この発明のNDIデバイスは、コマンドおよびデータがそこに接続されるデータチャネルのために指定されているかを判断する（ステップ220を参照）。もしそうであれば、NDIは、それがセンサであれば、指定されたデータチャネルからデータを取得し、またはそれがアクチュエータであれば、データチャネルに変換を行なうように命令する（ステップ230を参照）。センサチャネルから検索されるアナログデータは、まずデジタルデータに変換され（ステップ240を参照）、次に、データプロトコルに従って適切なフォーマットに変換される。さらにデジタルデータは、コントローラに伝送される（ステップ250を参照）。

10

【0043】

図1および2に示されるように、この発明のNDIデバイスは、バスコントローラとデータチャネルとの間のインターフェイスとして作動する。重要なのは、この発明のNDIデバイスは、データチャネルからデジタル化されたアナログデータ信号を受入れ、ネットワークバス上で伝送できることである。NDIは、バスコントローラからデジタルデータを受入れ、それをデータチャネルに与えることもできる。次にD/Aコンバータは、データをアナログ信号に変換することができる。データチャネルの中には、デジタルデータをアナログに変換することなく、直接それを受入れ使用することも可能であるものもある。NDIのいくつかの実施例は、NDIに一体化されたアナログ-デジタル（A/D）コンバータおよびD/Aコンバータを有してもよく、このことによって、アナログ信号をデータチャネルから受入れ、またはデータチャネルへ与えるように構成される。この発明のNDIデバイスは、選択されたデータプロトコルとともに作動して、ネットワークバスを介して適切にデータを受取りかつデコードし、またはフォーマットし、データを効率よく送信することができる。

20

【0044】

さらにこの発明のNDIデバイスは、データチャネルを構成する異なる種類のデバイスとコントローラが通信することを可能にする、たとえば、プログラム可能なトリガコマンド変換およびクロック信号などの、付加的な動作および特徴も提供する。これに加えて、この発明のNDIデバイスは、そこに接続されるデータチャネルを構成するための記憶された情報および手順を含む。A/DおよびD/Aコンバータは、プログラムする必要のあり得るまたは構成する必要のあり得るデータチャネルの構成要素の例である。この発明のNDIデバイスは、ローカルの発振器または同期バスクロックの何らかの断片または複数のローカル発振器または同期バスクロックであるローカルクロック信号をデータチャネルに与えることもあり得る。ネットワーク上の多くのまたはすべてのNDIデバイスのローカルクロック信号は、バスコントローラによって同期させることができる。さらにこの発明のNDIデバイスは、データチャネルを、個々に、同期グループで、またはすべてまとめてアドレス指定することができるように、データプロトコルとともに作動して、一意の論理アドレスおよびグループアドレスをデータチャネルの各々に与える。

30

【0045】

コントローラおよび遠隔デバイスならびに異なる構成を有するデータチャネルの間でのデータ通信を可能にすることに加え、この発明のNDIデバイスは、異なるデータ伝送モードを使用する、バスネットワーク上でのデータ通信も可能にする。具体的には、この発明の1つのモードでは、この発明のNDIデバイスは、同期モードでコントローラとともに作動し、コントローラによって与えられる同期クロック信号は、バスコントローラからコマンドおよびデータを受取るためにNDIデバイスによって使用される。この同じ同期クロック信号をNDIが使用して、データをバスコントローラまたはその他のネットワーク・デバイスに送信する。

40

【0046】

注意すべきことは、通常、NDIデバイスは、同期バスクロック信号をバスコントロー

50

ラに送り返さないことである。その代わり、通常、NDIデバイスは、同期モードでは同期バスクロック信号を受信したエッジでデータの出力時刻を記録する。一般に、バスコントローラのみが同期バスクロック信号を伝送する。さらに、通常、バスコントローラは、NDIデバイスからデータを受取るために非同期受信器を含む。

【0047】

同じ実施例または異なる実施例となり得る別のモードでは、NDIデバイスは、非同期モードでコントローラとともに作動する。この実施例では、この発明のNDIデバイスは、コントローラがネットワークバス上でデータを伝送しているレートを分析および判断して、このビットレートをを使用してバスコントローラからコマンドおよびデータを検索し、バスコントローラまたはその他のネットワーク・デバイスにデータを送る。非同期モードでも、NDIデバイスは、異なるNDIデバイス上にあるデータチャネルでのデータ変換を同期させることができる。データ変換は、バスコントローラからのコマンド内の特別なビットのエッジの変化で、またはその直後にそれを行なうことによって、異なるデータチャネル上で同期される。

【0048】

これらの利点および他の利点は、この発明のNDIデバイスによって実現され、その一実施例は、図3Aおよび図3Bに示される。特に、図3Aは、この発明の一実施例による、NDIデバイス40の一般的なブロック図である。図示のように、この発明のNDIデバイスは、先に図1に示したように、ネットワークバス38と遠隔デバイス34および36との間に接続される。この実施例では、遠隔デバイス34の1つはセンサであり、もう一方の遠隔デバイス36は、アクチュエータまたは同様の装置である。両方の遠隔デバイスとも、アナログ信号を調整するための信号調整デバイス58および60を含む。遠隔デバイス34に関しては、信号調整58はセンサ信号用である。アクチュエータのための信号調整は、60に示される。信号調整は、増幅器、フィルタ、減衰器等を含み得るが、これらに限定されない。

【0049】

重要なのは、遠隔デバイスと、この発明のNDIデバイスとの間に接続されるのは、それぞれA/Dコンバータ62およびD/Aコンバータ64であるということである。A/Dコンバータ62は、NDIデバイスとセンサ34との間に接続される。A/Dコンバータは、センサチャネルからのアナログ信号をデジタルデータに変換して、NDIデバイスに入力する。同様に、D/Aコンバータ64は、NDIデバイスとアクチュエータ装置36との間に接続され、NDIデバイスからのデジタル信号をアナログ信号に変換して、アクチュエータチャネルに入力する。センサおよびアクチュエータの中には、デジタル信号を直接生成または受信し得るものがあるため、A/D62またはD/A64が必要でない場合もあり得る。

【0050】

先に図1に示したように、この発明のNDIデバイスは、コントローラからコマンドおよびデータを受取る第1の受信器50を介して、ネットワークバスに接続される。第2の受信器52も、ネットワークが同期モードで作動している場合にコントローラから任意の同期クロック信号を受取るために設けられる。送信器46も、データをコントローラに伝送するために、この発明のNDIデバイスとネットワークバスとの間に接続される。さらに、記憶装置66およびローカルの発振器68は、この発明のNDIデバイスに接続される。NDIデバイスの異なる実施例は、受信器、送信器、ローカルの発振器およびメモリの一部またはすべてを組込み得る。

【0051】

図3Bは、この発明の一実施例によるNDIデバイス40のさまざまな制御論理構成要素を示す。特に、この発明のこの実施例のNDIデバイスは、データチャネル34および36に接続するためのポート70および72を含む。通常、これらポートはシリアルポートであるが、いくつかの実施例では、パラレルポートであってもよい。NDIデバイスのポートは、個々のポートコントローラ74および76によって制御される。各ポートに組

込まれるデータ線は、シリアルM O S I (master out slave in; マスター・アウト・スレーブ・イン) と称されるデータ出力線 78、C E と称されるチップ可能化またはチップ選択線 80、シリアルC L K と称されるクロック信号線 82 およびトリガ 84 を含む。図示のように、データ出力線 78 は、構成データ出力線 78 a およびデータ出力 / 特別コマンド出力線 78 b からなる。構成データ出力線 78 a は、後に説明するように、パワーアップ時にデータチャネルを構成するために使用される。さらに、データ出力 / 特別コマンド出力線は、バスコントローラからのデータをデータチャネルに与える。出力選択線 79 は、N D I デバイスがパワーアップモードであるか、または通常動作であるかに応じて、構成データ出力線とシリアル出力データ線との間で選択スイッチ 86 のトグル動作を行なう。

10

【0052】

上述のように、この発明のN D I デバイスは、メモリを作る複数のデータレジスタとして規定されるデータスタック 88 をさらに含む。データスタックは、データチャネルから取得されたデジタルデータを記憶するために使用される。データスタックは、データチャネルに送信するためのバスコントローラからのデータを記憶するためにも使用することができる。通常、データスタックは、後入れ先出し (last in first out; L I F O) デバイスとして作動し、ここでは、データスタックに最後に位置づけられたデータが最初にデータスタックから検索される値である。このようにして、スタックのサイズにかかわらず、データは、異なるN D I デバイスによって同じ順序でバスコントローラに戻される。新しいデータの値をスタックの最上部に置いてから、バスコントローラがそれを読取ることが可能になるまでの遅延は、最低限になる。しかしながら、バスコントローラがスタックの底部からデータを読取らなければならないのであれば、大きな遅延が生じるであろう。

20

【0053】

図示のように、データスタックに関連づけられるのは、スタック深さレジスタ 90 である。スタック深さレジスタは、その時点でのスタック内の有効なデータワード数を示す。

【0054】

さらに、この発明のN D I デバイスの現在の実施例の内部には、各データチャネルのための状態レジスタ 92 およびデータ選択マルチプレクサ 94 がある。重要なのは、状態レジスタが、たとえば、データチャネルがレディモードであるか、データチャネルがコマンドをサポートするか、またはメッセージ伝送エラーがあるか等の、データチャネルの状態に関する情報を含むことである。データ選択マルチプレクサは、要求されるデータに応じて、状態レジスタ、データスタックまたはスタック深さレジスタのいずれかを出力データマルチプレクサ 96 に接続する。各チャネルについてのデータ選択マルチプレクサ 94 は、それぞれのポートコントローラ 74 および 76 によって制御される。出力データマルチプレクサ 96 は、2つの遠隔デバイスの出力またはデバイス・インベントリ・レジスタ 98 の出力から選択する。N D I デバイスの異なる実施例は、N D I デバイス内に異なるマルチプレクサの配置を有し得るが、その効果は常に、バスコントローラが、必要とするN D I デバイス内のどのデータチャネルのためのレジスタにもアクセスできるようにすることである。

30

【0055】

デバイス・インベントリ (Device Inventory) ・ブロック 98 は、図 8 のフローチャートに示されるデバイス・インベントリ・オペレーションを実行するためにN D I デバイスによって使用される。

40

【0056】

さらにこの発明のN D I デバイスは、アドレスデコーダ 100 およびコマンドデコーダ 102 も含む。後に説明するように、これらデコーダは、コントローラによって伝送されるコマンドおよびデータを受取り、コマンドおよびデータをデコードし、そのコマンドおよびデータがN D I デバイスに接続されるデータチャネルの 1 つまたは複数にアドレス指定されているかを判断する。コマンドおよびデータがデータチャネルの 1 つについてアドレス指定されている場合、この発明のN D I デバイスは、コマンドに従ってそのデータチ

50

チャンネル上で作動する。上述の構成要素は、しばしばここでは、デバイス・インターフェイスと称される。

【 0 0 5 7 】

N D I デバイスは、U U I D、プロトコルバージョン、データチャンネルの数、論理アドレス、グループマスク、構成データ、および製造者またはユーザが規定し得るその他のデータを記憶するために N D I が使用する記憶装置 6 6 として、図 3 A に示される不揮発性メモリを含む。この記憶装置との通信は、構成レジスタ 1 0 4 から記憶装置への入出力線によって、図 3 B に示される。バスコントローラは、プロトコルに従ってメモリを読み出し、さまざまな記憶場所に書込むことができる。メモリ内の論理アドレスおよびグループマスク領域は特別である。これらは、図 8 のフロー図に従って N D I デバイスがデバイス・インベントリ競合 (Device Inventory Competition) に勝った直後に、バスコントローラによってのみ書込むことができる。このことによって、すべての N D I デバイスがバスコントローラによって一意に識別されることが可能となり、論理アドレスおよびグループマスクを割当てることが可能となる。これらの領域に書込む前にデバイス・インベントリ競合に勝たなければならないと命令することによって、これらの値を偶然変更することは事実上不可能となる。これと同じ種類のメモリ保護は、N D I デバイス製造者によって他のメモリ領域に適用することが可能である。

10

【 0 0 5 8 】

パワーアップ時、不揮発性メモリ 6 6 の内容の一部は、論理およびグループアドレスデコードレジスタ 1 0 0、シリアルポート 7 0 および 7 2 のための構成レジスタ 1 0 4、ポートコントローラ 7 4 および 7 6 内のコマンド変換レジスタにロードされ、一部の内容は、データチャンネルを構成するためにシリアルポートまたはその他のパラレルポートから送り出される。パワーアップ時、このメモリデータは、他にも使用され得る。このメモリは、ネットワーク・デバイス設置場所、較正データ等、ユーザが規定する情報を記憶するためにバスコントローラによって使用することもできる。通常、このメモリの内容は、トランスデューサ電子データシート (Transducer Electronic Data Sheet) を示す T E D と呼ばれる。

20

【 0 0 5 9 】

さらに、この発明の N D I デバイスは、コマンドを受取り、内蔵テスト、較正を行ない、かつ N D I デバイスをスリープモードおよびウエイクモードの間で遷移させるために、制御論理 1 0 6 を含んでもよい。

30

【 0 0 6 0 】

図 1 に示されるように、この発明の N D I デバイスは、ネットワークバスを通じてコントローラと通信する。以下に記載する、この発明のさまざまな動作の説明は、N D I デバイスに関するものである。マスターコントローラおよびネットワークコントローラの詳細な動作は、ここには記載しない。しかしながら、マスターコントローラおよびネットワークコントローラの動作の完全な詳細な開示は、2 0 0 0 年 1 2 月 8 日出願の「共通のバスを介して遠隔デバイスをデジタル方式で制御するためのネットワークコントローラ (NETWORK CONTROLLER FOR DIGITALLY CONTROLLING REMOTE DEVICES VIA A COMMON BUS)」と題される米国仮特許出願番号第 6 0 / 2 5 4 , 1 3 7 号に提供される。この特許出願の内容を、ここに引用により援用する。

40

【 0 0 6 1 】

上述のように、この発明の N D I デバイスは、いくつかの利点をもたらす。この発明の N D I デバイスの 1 つの重要な局面は、パワーアップ時に A / D および D / A コンバータならびに N D I デバイスに接続される遠隔デバイスが自己構成されることである。図 1 に示すように、ネットワークバスに接続される遠隔デバイスは、非常に多数でかつ非常に遠くに離れていることがあり得るため、中央の場所からデバイスを構成することは難しくなり得る。このことを考慮して、この発明の一実施例では、N D I デバイスは、信号調整デバイス 5 8 および 6 0 の利得、オフセット、フィルタ等に関するデータ、ならびに記憶装置 6 6 (図 3 A に示す) に記憶される A / D および D / A コンバータに関連する情報を含

50

む。具体的には、一実施例では、この発明のNDIデバイスは、パワーアップ時に、記憶装置から16の16ビットデジタルワードがポートの各々70および72を介して出力されるのを可能にする。この発明のNDIデバイスのこの局面は、市販のA/DコンバータおよびD/Aコンバータの自動構成を可能にしている。

【0062】

記憶装置に記憶される構成データは、コントローラによってプログラム可能である。16の16ビットワードは、32の8ビットバイトに分割して、ポートによってA/DコンバータおよびD/Aコンバータならびに信号調整に出力されるようにプログラム可能である。さらに、この発明のNDIデバイスは、構成データがポート70および72で出力されるときシリアルクロック82の位相およびシリアルクロック82の極性を変更するために、コントローラによってプログラム可能である。

10

【0063】

パワーアップ時にA/DコンバータおよびD/Aコンバータならびに信号調整デバイスを構成することに加え、この発明のNDIデバイスは、異なる種類のA/DコンバータおよびD/Aコンバータならびに信号調整デバイスと作動するように構成可能でもある。具体的には、逐次近似A/Dコンバータおよびシグマ/デルタオーバーサンプリングコンバータ等、多くの種類のコンバータがある。これらのコンバータは、クロックおよび動作遅延の面から見て、異なる動作をし得る。さらに、スイッチド・キャパシタ・フィルタおよびデジタル・アンチエイリアス・フィルタなどの信号調整デバイスの中には、クロックの面から見て異なる動作をするものがある。

20

【0064】

たとえば、図1に示されるように、同期モードでは、コントローラはネットワークバスを通じてネットワーク・デバイスに同期クロック信号を与える。同期モードでは、同期クロック信号は、ネットワークデータを伝送するためのクロック信号として使用される。アナログ・デバイスズ(Analog Devices)社のAD7714コンバータなどのA/Dコンバータの中には、正しく作動するために連続するクロック信号を必要とするものもある。通常、このクロックは、コントローラによって与えられる同期クロック信号よりも低い周波数である。このことを考慮して、この発明の一実施例では、この発明のNDIデバイスは、クロック分割器108を含んでもよい。クロック分割器は、図3Bに示されるようにコントローラによって出力される同期クロック信号に接続されていてもよいし、または図3Aに示されるようにローカルの発振器68に接続されていてもよい。

30

【0065】

NDIデバイスによって与えられるこのクロック信号は、図9に示されるバスコントローラによって同期させることができる。クロック信号は、バス上の、1つ、いくつかまたはすべてのNDIデバイスで、同時に同期させることができる。図9の例では、内部クロック周波数は、同期バスクロック周波数の1/4として示される。実際には、これは同期バスクロック周波数の何分の1であってもよい。

【0066】

図4は、バスコントローラからの1つの特定のコマンドに応答してNDIによって行なわれる動作のブロック図である。この図は、NDIデバイスが1度に2つ以上の作業を行なえることを示す。NDIデバイスの同時に複数の作業を行なえる能力によって、この発明のNDIデバイスは、同時に1つまたは複数のデータチャネルを取得または制御する一方で、同時にバスコントローラと通信することができる。

40

【0067】

図4を参照すると、作動中、この発明のNDIデバイスは、まずトリガおよび読出(Trigger and Read)コマンドなどのコマンドをコントローラから受取り(ステップ310を参照)、そのコマンドを解釈する。コマンドおよびアドレスがNDIデバイス上のデータチャネルのためのものであれば、NDIデバイスは、ステップ330および360に示される動作を並行して開始する。具体的には、NDIデバイスは、NDIに装着されるデータチャネルに変換信号を送る(ステップ330を参照)。このデータパルスの立上がりエ

50

ッジは、トリガおよび読出コマンドの終わりにパリティビットの中央エッジで起こる。変換信号は、アナログデータをサンプルにラッチし、A/Dコンバータの回路を保持するために与えられるか、または変換信号は、D/Aに変換プロセスを開始させるために使用することもできる。この信号の立上がりエッジの正確なタイミングによって、ネットワーク・デバイスが同期モードで作動していない場合でも、多くのデータチャネルは、いつアナログデータをサンプリングすべきか、または変換するべきかを知ることができる。

【0068】

変換信号が生成された後、短い休止が起こる（ステップ340を参照）。この休止によって、データチャネルは、そのサンプルにラッチされるアナログ信号を変換して、デジタル値に変換するように保持するための時間を有することができる。NDIデバイスのこの実施例では、この休止はプログラム可能である。2つの選択肢がある。これは僅か数百ナノ秒であってもよく、または6マイクロ秒にプログラムすることも可能である。

10

【0069】

休止の後、シリアル転送が行なわれる（ステップ350を参照）。このシリアル転送中、プログラム可能なワードの出力時刻がMOSE線上で記録される。このプログラム可能なワードは、特別なA/Dコンバータにデータを出力させるために使用される。例としては、AD7714がある。このシリアル転送が続く中で、データはデータチャネルからNDIデバイスに戻される。MOSE線上でNDIデバイスに戻されるデジタルデータは、データレジスタ89に記憶される。

20

【0070】

動作ステップ330、340および350が行なわれるのと同時に、動作ステップ360、370および380も行なわれる。動作ステップ360において、入力データレジスタ89の内容は、データスタックの最上部にシフトされる。次に、データスタックの最上部の内容は、送信機レジスタにロードされる（ステップ370を参照）。最後に、送信機レジスタの内容は、NDIデバイスの送信器46によってバスコントローラに送り返される（ステップ380を参照）。

【0071】

これら動作のすべての間、NDIインターフェイスは、連続したクロック信号をデータチャネルに与えている。すべてのデータチャネルがこのクロックを使用するわけではないが、使用可能である。このクロックの周波数はプログラム可能である。このクロック信号は、スイッチド・キャパシタ・フィルタ、デジタルフィルタまたはシグマ/デルタコンバータ等のデバイスを作動させるために有用である。このクロック信号は、動作ステップ300から380が行なわれていないときでも作動し続ける。

30

【0072】

汎用非同期送受信器（Universal Asynchronous Receiver Transmitter；UART）プロトコルが使用されている場合、コントローラは、スタートビット、コマンド領域、0にセットされる未使用の最後のビットを有するアドレス領域、および1にセットされるストップビットを含む、コマンドを伝送する。この実施例では、この発明のNDIデバイスは、アドレス領域の未使用ビットからストップビットへの遷移と同時に、各データチャネルでの関数の実行を開始する。

40

【0073】

クロックおよび遅延の発行に加え、A/DコンバータおよびD/Aコンバータの中には特別なコマンドも必要とするものがある。たとえば、A/DコンバータおよびD/Aコンバータの中には、センサからの異なる読取値をとるようにプログラム可能なものもある。たとえば、1つの応用例では、A/Dコンバータは、3次元で歪みを検知する歪みゲージに接続される。各読取可能な寸法は、別々の16ビットのアドレスでアドレス指定可能である。各寸法の測定値の1つまたはすべては、関連づけられる16ビットコマンドをコンバータに適用することによってアクセス可能である。ビットコマンドのすべてをコントローラに記憶して、それらをネットワーク上で伝送することは煩わしい場合があり得る。プロトコルの動作を簡略化するため、（以下に説明するように）、この発明のNDIは、こ

50

れら特別なコマンドを維持して、コントローラがそれらを維持したり、それらをネットワーク上で送る必要をなくす。このことを考慮して、一実施例では、この発明のNDIは、NDIに接続されるA/DコンバータまたはD/Aコンバータに関連づけられる特別なビットコマンドを含む。図3Aおよび3Bを参照すると、これらコマンドは、元々は記憶装置66に記憶され、プログラム可能である。パワーアップ時、これらの特別なビットコマンドは、コマンドデコーダ102に記憶することができる。

【0074】

図5を参照すると、作動中、特別なコマンドを備えるコンバータを有する遠隔デバイスがアドレス指定される場合、（センサからデータを取得するため、またはアクチュエータの場合は遠隔デバイスを作動させるため）、コントローラは、そのコンバータを備えるデータチャネルのアドレスとともに、適切にフォーマットされたトリガコマンドを送る（ステップ400を参照）。（コマンドのフォーマットは後述する）。データチャネルに関連づけられるNDIがコマンドおよびアドレスを受取ると（ステップ410を参照）、まず、NDIは、アドレス指定された遠隔デバイスが特別なコマンドを必要とするかを判断する。これは、受取ったアドレスをコマンドデコーダに記憶されるデータに関連づけられるアドレスと比較することによって行なう（ステップ420を参照）。アドレスに基づいて、この発明のNDIは、コマンド変換レジスタから適切な特別コマンドを検索する（ステップ430を参照）。次に、特別コマンドは、情報を受取るために、センサの場合はコマンドに対応するアクチュエータを作動させるために、コンバータに適用される（ステップ440を参照）。

【0075】

先に簡単に述べたように、コントローラおよびこの発明のNDIデバイスは、同期モードまたは非同期モードのどちらでも作動することができる。同期モードでは、コントローラは連続する同期クロック信号を与える。同期クロック信号は、バスコントローラからのデータの入力時刻を記録し、コントローラへのデータの出力時刻を記録するために、この発明のNDIデバイスによって使用される。このことによって、バスコントローラは、0ビット/秒から何らかの最大ビットレートまでのどのようなデータレートも選ぶことができる。

【0076】

この発明のNDIデバイスは、コントローラが非同期モードまたは同期モードのどちらで作動しているかを自動的に検出することができる。具体的には、図3A、3Bおよび図6を参照すると、この発明のNDIデバイスは、クロック検出器を使用して、第2の受信器52で受取られる信号を連続的に調べる（ステップ500を参照）。同期クロック信号が存在する場合（ステップ510を参照）、この発明のNDIデバイスは、同期モードで作動し（ステップ520を参照）、コントローラからの同期クロック信号を使用して、データの入力時刻および出力時刻を記録する。しかしながら、この発明のNDIデバイスが第2の受信器52からの同期クロック信号を検出しない場合（ステップ510を参照）、この発明のNDIデバイスは、非同期モードで作動する（ステップ530を参照）。

【0077】

上述のように、非同期モードでは、コントローラはさまざまなビットレートで作動し得る。このことを考慮して、一実施例において、この発明のNDIデバイスは、コントローラが作動しているビットレートを検出する。具体的には、図7を参照すると、この実施例では、この発明のNDIデバイスがコントローラが非同期モードで作動していると判断した後、この発明のNDIデバイスは、コントローラによって伝送されるコマンドおよびデータのビットを監視する（ステップ600を参照）。NDIデバイスは、ビットレートデコーダを使用して、各ビットの受取り間の時間を判断する。伝送間の時間が実質的に同じである予め定められた数のビットが受取られた後（ステップ610を参照）、この発明のNDIデバイスは、NDIデバイスに送られているデータのビットレートを選び、そのビットレートで作動する（ステップ620を参照）。重要なのは、NDIデバイスのビットレートを検出する能力は、ネットワークシステムに電力障害があった場合、またはコント

ローラが同期モードから非同期モードに遷移した場合に、すばやく復旧するために有利であることである。自動的に同期クロックを検出する特徴および自動的にビットレートを検出する特徴の第2の重要な局面は、それによって、1つの種類のNDIデバイスが異なるモードのネットワーク通信を使用してネットワーク上で通信することが可能となることである。ネットワークシステムの設計者は、ネットワークシステムの特定の用途に最適化したネットワーク通信のモードを選ぶことができる。

【0078】

さらに一実施例では、バスコントローラは、NDIデバイスに或る特定のビットレートでデータを送受信するように命令することができる。バスコントローラは、ビットレート変更コマンドを送り、それに続いてNDIデバイスがこれからそれに変更するビットレートを送ることができる。NDIデバイスは、バスコントローラによって指示されるビットレートで、データをバスコントローラから受取り、バスコントローラに伝送する。さらに、コントローラは、例としてのメッセージを新しいビットレートで送ることができ、この発明のNDIは、実際のコマンドおよびデータが送られる前にその新しいビットレートに変更する。

【0079】

この発明のNDIデバイスのもう1つの利点には、或るネットワークバス上の多くの異なるNDIデバイス上の多くのデータチャネルが同期または非同期モードでバスコントローラに通信している場合に、アナログデータを実質的に同時にサンプリングまたは変換することができる点である。非同期モードでのデータサンプリングまたは変換を同期させる1つの方法は、変換信号84の立上がりエッジ(図3Bを参照)を、コントローラからの変換コマンドの中央エッジが変化したとき、またはその直後にhighにすることによって達成される。プロトコルの中には、変換コマンドをトリガコマンドと呼ぶものがある。プロトコルおよびNDIデバイスが非同期モードでも同時にデータを取り、またはデータを変換する能力は、等時性と呼ばれる。

【0080】

同期データサンプリングの第2の方法は、同期クロック信号を各NDIデバイスから各NDIデバイスに関連づけられる各データチャネルに与えることによるものである。クロック信号は、同期コマンドを使用してバスコントローラによって同期され、NDIデバイスは、図9のタイミングに従ってコマンドに応答して同期する。そのようにして、同期バス信号の分割された一部である周波数を有するクロック信号を使用するデータチャネルのすべては、ほぼ完璧に同期して作動する。

【0081】

インターフェイスに異なる種類のA/DコンバータおよびD/Aコンバータならびに異なる信号調整システムを与え、同期モードおよび非同期モードの両方で作動することに加え、この発明のNDIデバイスは、ネットワークでのデータの伝送におけるオーバーヘッドを省くこともできる。特に、図3Bに示されるように、この発明のNDIデバイスは、データスタック88を含む。NDIデバイスがセンサ遠隔デバイスに接続される場合、データスタックは、入力データスタックである。入力データスタックは、データチャネルから受取ったデータを含む。この場合、入力データスタックに記憶されるデータは、コントローラによって1度に1つのワード(すなわち1度に1つのレジスタ)、またはデータのブロックとして(すなわち一度に複数のレジスタを)読出すことができる。データスタックから一度にデータのブロックを読出すことによって、ネットワークのオーバーヘッドが省かれる。

【0082】

さらに、この発明のNDIデバイスがアクチュエータ遠隔デバイスに接続される場合、データスタックは、出力データスタックである。この場合、出力データスタックは、コントローラによって転送されアクチュエータ・データチャネルに出力されるべきデータを含む。トリガコマンドがアクチュエータに送られると、アクチュエータはデータスタックの最上部のワードのデジタルからアナログへの変換を行ない、次に、NDIデバイスは、ス

10

20

30

40

50

タックをポップし、スタックの最上部の新しいワードをD/Aに伝送する。データワードは、コントローラによって個々にまたはワードのブロックとして出力データスタックに書込むことができる。出力データスタックに、一度に1つではなく、データのブロックを書込むことによって、ネットワークのオーバーヘッドが省かれる。

【0083】

上述の利点に加え、この発明のNDIデバイスにはさらなる利点がある。具体的には、一実施例において、この発明のNDIデバイスは、単純かつ高速でありながら堅牢なデジタル・マルチドロップ・ネットワーク上でデータチャネルが通信することを可能にするプロトコルとともに作動する。どのような適用可能なプロトコルでも、この発明のNDIデバイスとともに使用可能であることを理解されたい。しかしながら、以下に記載するのは、図1に示されるネットワークシステム30で使用される場合に、いくつかの利点を提供する、或る特定のプロトコルである。1つの重要な利点は、プロトコルが単純であるため、NDIデバイスを高レベルプロセッサではなく、ステートマシンとして実現することができることである。

10

【0084】

特に、プロトコルは、低レベル通信制御インターフェイス（すなわちネットワークコントローラおよびNDIデバイス）を保証するように設計される。プロトコルによって、ネットワーク内で高度に小型化されたコントローラおよびネットワーク・デバイス・インターフェイスを開発することが可能となる。ネットワークコントローラおよびNDIデバイスが、プロトコルとともに作動する特定用途向け集積回路（ASIC）またはFPGAで実現される場合、ネットワークコントローラおよびNDIデバイスは、素早く応答することができかつ最大限のバスの効率を効率良く実現する。バスコントローラまたはNDIデバイスがASICとして実現される場合、バスコントローラおよびNDIデバイスは、非常に小型化できる。

20

【0085】

この発明のプロトコルは、低オーバーヘッドのコマンド構造も有する。この発明のプロトコルは、固定長のメッセージを使用しない。メッセージの長さは、コマンドに応じて変り得る。このことによって、コマンドを実行するのに必要でない場合、不必要なデータが伝送されるのを排除することができる。さらに、コマンドセットは最低限かつ単刀直入であるため、ユーザは最低限の操作でネットワークバスを介してデータを容易に渡すことができる。

30

【0086】

上述のように、この発明のNDIデバイスは、固定された低レベル命令セットを有するため、簡略化されたコントローラおよびネットワーク・デバイス・インターフェイスをネットワーク・デバイス上で使用することを場合によっては可能にするプロトコルとともに作動する。特に、低レベルコマンドセットによって、NDIデバイスは、プロセッサまたはマイクロコントローラではなく、ステートマシンとして実現することができる。低レベルコマンドセットがどのように働くかの一例を示す。バスコントローラがNDIデバイスのメモリからデータワードを読出したい場合、これは、セットポインタと呼ばれるコマンドをNDIに送り、次にそのコマンドに続いてそのポインタの値を送る。NDIは、そのコマンドをデコードし、そのメモリアドレスポインタをバスコントローラによって送られた値にセットする。メモリアドレスポインタは、バスコントローラによって送られる値をラッチするレジスタでありさえすればよい。次に、バスコントローラは、メモリワード読出コマンドをNDIデバイスに発行する。NDIデバイスは、そのメモリアドレスポインタによって指し示されるメモリワードにアクセスし、それをバスコントローラに伝送することによって応答する。バスコントローラがNDIのメモリからデータのブロックを読出したい場合、このプロセスを複数回繰返す。

40

【0087】

コマンドセットが高レベルコマンドセットである場合、上述の読出プロセスは、特別メモリブロック読出コマンドとして実現される。NDIデバイスは、特別メモリブロックに

50

アクセスするためにそのポインタをどこにセットするかを知り、そのポインタをセットし、特別メモリブロックがどれくらい大きいかを知り、次に特別メモリブロックを送ることができる必要がある。このプロトコルは、どこに特別メモリ情報が記憶されているか、および特別メモリ情報のブロックサイズがどれくらい大きいか、どこに特別メモリデータを記憶するかを常に把握するなどのこれらの作業を、N D I デバイスから取除く。これらの作業は、メモリポインタと読書きできるメモリとをセットするように低レベルの命令をつなげる代わりに、バスコントローラによって行なう。

【 0 0 8 8 】

上述のように、この発明のN D I デバイスは、固定された低レベルかつ低オーバーヘッドの命令セットを有するプロトコルとともに作動し、このプロトコルは高性能なセンサおよびアクチュエータとともに使用されると、ネットワークバス上の実際のデータレートを向上する。センサおよびアクチュエータをネットワークするためのネットワーク・トラフィックは、通常のコンピュータネットワーク・トラフィックとは異なる。コンピュータネットワークは、1つのコンピュータから別のコンピュータへ大きなデータファイルまたはメッセージを伝送する必要がある。結果として、それらは、堅牢かつエラーのないデータ伝送を保証するため、メッセージに関連づけられるチェックサムおよびブロックサイズを有する。これらコンピュータネットワークプロトコルに関連づけられる、これらのチェックサムおよびブロックサイズ、またはその他のオーバーヘッドは、オーバーヘッドのサイズが合計のメッセージサイズと比較して小さいため、問題にはならない。センサおよびアクチュエータのネットワークトラフィックが異なるのは、多くの場合、ネットワーク上のメッセージの大半が、16ビットのデータのみを備える非常に小型のメッセージであるためである。ブロックサイズ、チェックサムまたはその他の不必要なオーバーヘッドが小型の16ビットデータメッセージに付加されると、チェックサム、ブロックサイズおよびその他のオーバーヘッドは、実際の16ビットデータメッセージよりも多くのビットを有し得る。このことによって、事実上、センサおよびアクチュエータ・ネットワークバスの帯域幅が低減する。この発明のN D I デバイスは、チェックサムおよびブロックサイズなど不必要なオーバーヘッドを短い16ビットのメッセージに付加しないメッセージプロトコルを使用することによって、この問題を解決する。これら短い16ビットのメッセージに付加される唯一のオーバーヘッドは、メッセージの開始を示す同期パターン、フラグビットおよびパリティビットである。パリティビットは、16ビットのメッセージにエラーがないかをチェックするために使用される。フラグビットは、送信するN D I デバイスまたは関連づけられるデータチャネルにエラー状況が存在するかどうかを示す。

【 0 0 8 9 】

この発明のプロトコルは、通常、マンチェスタ符号化されたフォーマットで伝送されるが、他のU A R T システムと通信するため、必要であれば、汎用非同期送受信器(U A R T)フォーマットプロトコルで実現され得る。たとえば、一実施例では、プロトコルは、Bi-Phase Sensor and System (BiSenSys) と称する、R S - 4 8 5 ベースの、マルチドロップ、マンチェスタ符号化されたプロトコルを使用する。BiSenSysは、18ビットの高速かつ高効率のプロトコルで、マンチェスタ符号化を使用するデジタルバス構造上の遠隔デバイスとサブシステムとを接続するために使用する。マンチェスタ符号化を使用する別のプロトコルの例は、M I L - S T D - 1 5 5 3 である。具体的には、データの各ビットは、1つの検出可能な遷移によって検出可能である(すなわち、「0」はlowからhighと規定され、「1」はhighからlowと規定される)。さらに、各メッセージは、同期パターン、メッセージ本体およびパリティフラグからなる。BiSenSys伝送プロトコルは、同期モードまたは非同期モードで作動可能であり、同期モードでは、1 H z ~ 1 0 M H z までのどのデータレートでも作動するように実現可能であり、または非同期モードでは、1 . 2 5 メガバイト毎秒、2 . 5 メガバイト毎秒、5 . 0 メガバイト毎秒、および1 0 . 0 メガバイト毎秒で作動するように実現可能である。さらにこの発明のプロトコルは、U A R T ベースのプロトコルで実現され得る。このプロトコルは、1 . 0 メガビット毎秒のデータレートで作動するように設計され、9ビットメッセージフォーマットおよび非ゼロ回帰(no

10

20

30

40

50

n-return-to-zero) ビット符号化を使用する。

【 0 0 9 0 】

ネットワークバスにわたって伝送されるデータフレームには、少なくとも3種類があり、データフレームは、BiSenSysプロトコルとUARTプロトコルとの間で異なる。BiSenSysデータフレームには3種類ある。第1の種類は、コマンドフレームである。コマンドフレームは、コマンド同期パターン、これに続くマンチェスタ符号化された10のアドレスビット、これに続くマンチェスタ符号化された7つのコマンドオペコードビット、およびマンチェスタ符号化された1つのパリティビットからなる。コマンド同期は、バス上の、1と2分の1バスビット期間のhighの状態と1と2分の1ビット期間lowの状態とからなる。

10

【 0 0 9 1 】

第2の種類のフレームは引数フレームである。引数フレームは、データ同期パターンで始まり、これにマンチェスタ符号化された16の引数ビット、マンチェスタ符号化された1つのフラグビット、最後にマンチェスタ符号化された1つのパリティビットが続く。データ同期パターンは、1と2分の1ビット期間のlowと、これに続く1と2分の1ビット期間のhighとからなる。BiSenSysフレームの最後の種類は、データフレームである。データフレームは、データ同期パターンで始まり、これに16ビットのマンチェスタ符号化されたデータが続き、マンチェスタ符号化された1つのフラットビット、マンチェスタ符号化された1つのパリティビットが続く。引数フレームとデータフレームとの唯一の違いは、引数フレームがバスコントローラによって伝送され、データフレームはネットワーク・デバイスによって伝送される点である。引数は、あるコマンドに続いてバスコントローラによってのみ伝送される。引数が続くことを必要とするコマンドは、RHAMIS / BiSenSysプロトコルで規定される。バスクロックがBiSenSys同期モードで使用される場合、これはマンチェスタ符号化されたビットストリームに直交して伝送される。

20

【 0 0 9 2 】

UARTプロトコルの場合、メッセージフォーマットは、3つ以上の11ビットフレームを含む。各メッセージは、発行されるコマンドに応じて、メッセージにつき2つの付加的なオプションのフレームを備えた、メッセージにつき3つの必要なフレームを有する。すべてのフレームの第1のビットは、スタートビット(0にセットされる)である。すべてのフレームの最後のビットは、ストップビット(1にセットされる)である。メッセージの第1のフレームは、UARTプロトコルのためのアドレスフレームである。これは、スタートビット、8ビットのアドレス領域、1にセットされるアドレスビット、およびストップビットの形である。UARTメッセージの第2のフレームは、コマンドフレームである。コマンドフレームは、スタートビット、7ビットのコマンド領域、未使用ビット、0にセットされるアドレスビットおよびストップビットからなる。引数フレームがUARTメッセージに属する場合、それらはコマンドフレームに続く。それらは、スタートビット、これに続く8つの引数ビット、0にセットされるアドレスビット、ストップビットからなる。UARTメッセージの最後のフレームは、チェックサムフレームである。これは、スタートビット、アドレス、コマンドおよび桁上りを伴わないデータビットの、2を法とした合計である8ビットチェックサム、0にセットされるアドレスビットおよびストップビットからなる。

30

40

【 0 0 9 3 】

同期モード、非同期モードのどちらでも作動しかつ異なる伝送プロトコルで作動する低レベル命令セットを有するプロトコルを提供することに加え、この発明は、各データチャネルに一意のアドレスを割当てするための方法も提供する。図1に示されるように、典型的なネットワークシステム30には、複数の異なるNDIデバイスに接続される多数のデータチャネルがあり、それらのすべてがバスコントローラと通信するために共通のネットワークバスを使用する。さらに、ネットワーク・デバイスの各々は、NDIデバイスによって命令され得るいくつかのデータチャネルまたはいくつかの作業を有し得る。この発明のプロトコルは、ネットワーク・デバイスの各データチャネルに3つの種類のアドレスを与

50

える。具体的には、各ネットワーク・デバイス上の各データチャンネルには、個々の論理アドレス、グローバルアドレス、および設定されている場合、グループマスクを割当てることが可能である。

【 0 0 9 4 】

論理アドレスは、ネットワーク・デバイス上の1つのデータチャンネルによって、またはネットワーク・デバイスの1つの制御可能な作業によって認識される、アドレスである。一方グローバルアドレスは、すべてのネットワーク・デバイスのすべてのデータチャンネルによって認識され、グループアドレスは、すべてのネットワーク・デバイスのすべてのデータチャンネルのサブセットによって認識される。データチャンネルにはグループアドレスは割当てられない。それらにはグループマスクが割当てられる。グループマスク内の各ビットはグループアドレスに対応する。このようにして、1つの16ビットマスクを使用して、データチャンネルを16のグループに割当てることができる。たとえば、グループマスクの第1および第3のビットが1にセットされている場合、データチャンネルは、それ自身がグループ1および3に属すると認識する。それは、1または3のいずれかのグループアドレスを有するコマンドメッセージに応答する。これら3つの異なるアドレスは、それらによって、コントローラが、同じネットワーク・デバイス内の、または複数のネットワーク・デバイス内の、1つのネットワーク・デバイスの個々のデータチャンネルまたはデータチャンネルのグループと通信することが可能となり、またはすべてのネットワーク・デバイスのすべてのデータチャンネルと通信することが可能となるため重要である。これらアドレスの各々は後に詳しく説明する。

【 0 0 9 5 】

論理アドレスおよびグループアドレスの判断は、どのような方法を選択して行なってもよい。好ましい方法は、2000年12月8日出願の「共通のバスを介して遠隔デバイスをデジタル方式で制御するためのネットワークコントローラ(NETWORK CONTROLLER FOR DIGITALLY CONTROLLING REMOTE DEVICES VIA A COMMON BUS)」と題される、米国仮特許出願番号第60/254,137号に記載される。この方法は、各ネットワーク・デバイスに関連づけられる汎用一意識別子(Universal Unique Identifier; UUI D)を使用する。UUI Dコードは、すべてのネットワーク・デバイスに対して一意である80ビットのコードであり、デバイスが製造された場所および日付に基づく。

【 0 0 9 6 】

図3Aを参照すると、通常、この発明のネットワーク・デバイスのためのUUI Dは、NDIデバイスの記憶装置66に記憶される。論理アドレスおよびグループ後の判断中、コントローラは、この発明のNDIデバイスにさまざまなコマンドを送り、それらにそれらのそれぞれの記憶装置にアクセスし、UUI Dの個々のビットを分析し、或るビットが1か0であるかに基づいて応答するかまたは沈黙したままでいるように命令する。このことによって、バスコントローラは、一度に1つのネットワーク・デバイスずつ、すべてのネットワーク・デバイスのUUI Dを発見することができる。ネットワーク・デバイスのUUI Dを見つけるとその直後に、バスコントローラは、そのネットワーク・デバイス上のすべてのデータチャンネルに論理アドレスおよびグループマスクを割当てて。

【 0 0 9 7 】

図8は、バスコントローラの制御の下、NDIデバイスを使用して、ネットワーク・デバイスがそれ自身をバスコントローラに対して一意に識別するために行なうステップのフローチャートである。デバイス・インベントリ・セッションを開始するため、NDIデバイスは、デバイス・インベントリ可能化(Device Inventory Enable)コマンドを使用して可能化される。これによって、NDIはデバイス・インベントリ・モードに入る(ステップ700を参照)。アドレス「0」(すべてのデバイスのために確保されたグローバルアドレス)がアドレス領域で指定されると、バス上のすべてのネットワーク・デバイスは、それ自身をデバイス・インベントリ・モードにする。現在割当てられている論理アドレスがアドレス領域で指定されると、そのネットワーク・デバイスは、それ自身をデバイス・インベントリ・モードにする。現在割当てられているグループアドレスがアドレス領域

で指定されると、そのグループに属するすべてのネットワーク・デバイスは、それら自身をデバイス・インベントリ・モードにする。

【 0 0 9 8 】

UUI Dワードサーチ (word search) 中、デバイス・インベントリ・コマンド内のさまざまな関数コードが使用される。新しいUUI Dワードサーチ関数コードを備えるデバイス・インベントリ・コマンドは、すべてのデバイス・インベントリ・セッション可能化装置をUUI Dワードサーチモードにセットする (ステップ 7 0 5 を参照)。バス上のネットワーク・デバイスは、UUI Dワードサーチモードでなければ、UUI Dワードサーチで競合しない。ワードサーチモードに入るようにと命令された直後、NDI デバイスは、UUI D の最下位ビットを第 1 のレジスタへロードし、かつ「1」を第 2 のレジスタにロードすることによって、自動的に進行する (ステップ 7 1 0 を参照)。NDI デバイスがUUI Dワードサーチモードである場合、それらは、2 つのUUI Dビット競合関数コードに応答する。

10

【 0 0 9 9 】

マスターコントローラは、UUI Dビット競合 (Bit Competition)、ノー・ドロップアウト (No Dropout) 関数コードを備えるデバイス・インベントリ・コマンドを発行する。すべてのNDI デバイスは、このコマンドに基づいて判断を行なう (ステップ 7 2 0 を参照)。NDI デバイスは、ステップ 7 3 0 に進み、レジスタ 1 内のそのビットに基づいて判断を行なう。このビットが「1」であれば (ステップ 7 3 0 を参照)、ネットワーク・デバイスは沈黙したままである。このビットが「0」である場合、ネットワーク・デバイスはUUI Dパルスを送送する (ステップ 7 4 0 を参照)。次に、ネットワーク・デバイスは、評価済みのビット (「1」または「0」のどちらか) をビットレジスタ 2 に移動し、その次のUUI Dビットにアクセスし、それをビットレジスタ 1 にロードする (ステップ 7 6 0 を参照)。

20

【 0 1 0 0 】

マスターコントローラは、バス上で伝送されるUUI Dパルスを聞く。マスターコントローラがUUI Dパルスを聞いた場合、少なくとも 1 つのネットワーク・デバイスがその現在のUUI Dビットに「0」を有することがわかる。マスターコントローラが少なくとも 1 つのUUI Dパルスを聞いた場合、それが発行する次のコマンドは、UUI Dビット競合、1 のドロップアウト (1's Dropout) 関数コードを備えるデバイス・インベントリ・コマンドである。NDI デバイスは、そのコマンドを解釈し、判断を行わなければならない (ステップ 7 2 0 を参照)。ネットワーク・デバイスがこのコマンドおよびコードを聞いた場合、レジスタ 2 のそのUUI Dビットを見る (ステップ 7 8 0 を参照)。このビットが「1」である場合、ネットワーク・デバイスは、現在のUUI Dビット競合モードを出て、UUI Dワードサーチでの競合を終了する。レジスタ 2 のビットが「0」である場合、ネットワーク・デバイスは、レジスタ 1 のビットを見る (ステップ 7 3 0 を参照)。それが「0」である場合、ネットワーク・デバイスはUUI Dパルスを送送する (ステップ 7 4 0 を参照)。このビットが「1」である場合、ネットワーク・デバイスは沈黙したままである。次に、ネットワーク・デバイスは、レジスタ 1 のビットをビットレジスタ 2 に移動し、その次のUUI Dビットにアクセスし、それをビットレジスタ 1 にロードする (ステップ 7 8 0 を参照)。

30

40

【 0 1 0 1 】

バス上でマスタコントローラにどのようなUUI Dパルスも聞こえない場合、マスタコントローラは、依然としてUUI Dワードサーチモードにあるすべてのネットワークデバイスがビットレジスタ 2 内に「1」を有することを認識する。これに応答して、マスタコントローラが発行する次のコマンドは、UUI D Bit Competition、No Dropout 関数コードを伴った Device Inventory コマンドである。ネットワークデバイスがこのコマンドを受取ると、そのレジスタ内に何があるかに関係なく、競合したままとなる。このコマンドに応答して、競合するすべてのデバイスは、ビットレジスタ 1 を見る (ステップ 7 3 0 参照)。それが「0」であると、ネットワークデバイスはUUI Dパルスを送信する (ス

50

テップ 740 参照)。それが「1」であると、ネットワークデバイスは使用されないままとなる。ネットワークデバイスは、レジスタ 1 内のビットをビットレジスタ 2 内に移し、その次の UUI D ビットにアクセスして、それをビットレジスタ 1 内にロードする (ステップ 760 参照)。

【0102】

このセクションで説明される UUI D Bit Competition は、80 の UUI D ビットのすべてに対して繰返される。マスタコントローラは、全部合わせて 81 の UUI D Bit Competition 関数コードを発行しなければならない (ステップ 750 参照)。マスタコントローラは、バス上の最後のネットワークデバイスが、勝った最終ビットを誰が有したかを決定できるように、81 番目の UUI D Bit Competition 関数コードを発行しなければなら

10

【0103】

UUI D Bit Competition 関数コードを伴った Device Inventory コマンドが、全部合わせて 81 回にわたって発行された後に、UUI D ワードサーチモードで残されるのは、1 つのネットワークデバイスのみである。このネットワークデバイスが勝者である (ステップ 770 参照)。このネットワークデバイスは、この時点でバスコントローラに一意に識別されており、特定の記憶場所の内容で応答する (ステップ 780 参照)。特定の記憶場所の上位のバイトはプロトコルの改訂数を保持し、下位のバイトはネットワークデバイス上のデータチャネル数である。勝ったネットワークデバイスおよびバスコントローラは、このネットワークデバイスが、全 81 ビットの競合に勝ったことによって勝利したことを認識している。

20

【0104】

勝ったデバイスが 81 番目の UUI D Bit Competition 関数コードを受取ると、このデバイスは、このデバイスがそのメモリ内に記憶したプロトコルのバージョン数およびチャネル数を含むバイフェイズ準拠データフィールドで応答する (ステップ 780 参照)。バスコントローラは、勝ったデバイスの論理アドレスチャネル割当を決定するために、この情報を用いることができる。勝ったネットワークデバイスは、その論理アドレスおよびグループマスクの記憶場所を保護しない (ステップ 790 参照)。バスコントローラは、すぐに引き続いて、論理アドレスおよびグループマスクを割当て (ステップ 800 参照)。データチャネルが複数のチャネルを含む場合、マスタコントローラは、各チャネルに論理アドレスとグループマスクとを割当て (ステップ 810 参照)。論理アドレスおよびグループマスクが割当てられた後に、ネットワークデバイスは、デバイスインベントリモードを終了する。バスコントローラは、すべてのネットワークデバイスが発見され、かつ、論理アドレスおよびグループマスクが各チャネルに割当てられるまで、UUI D ワードサーチを繰返す。バスコントローラは、バス内の UUI D が、仮に発見されるとしてすべて 1's である場合に、全デバイスが発見したことになると認識する。すべて 1 の UUI D を有するネットワークデバイスは存在しない。

30

【0105】

上述のように、各ネットワークデバイスに対する各チャネルには、データチャネルを一意に識別する論理アドレスが供給される。さらに、多数のデータチャネルをアドレス指定するグループアドレスと、全ネットワークデバイスの全チャネルをアドレス指定するグローバルアドレスとが存在する。グローバルアドレスは、グローバルデータおよびコマンドを交換するために用いられる。グループアドレスは、ネットワークデバイスの複数のデータチャネルによって認識されるアドレスである。グループアドレスにより、1 つ以上のデータチャネルは、同じコマンドに同時に応答する。1 つのデータチャネルのみをグループアドレスに割当てることができる。グループアドレスと関連するのが、アドレスデコーダ 100 内に記憶されるグループマスクである。グループマスクは、16 ビットワードであり、1 にセットされた各ビットは、データチャネルが属するグループを示す。

40

【0106】

一実施例における一例として、グローバルアドレスには 0000 hex が割当てられる

50

。この実施例において、アドレス 0 0 0 0 h e x が送信されると、N D I デバイスのすべてがこのコマンドに従う。これは、一般に、システムのリセットまたはシステムのテストのために用いられる。さらに、一実施例において、グループアドレスは、0 0 0 1 h e x から 0 0 0 f h e x の範囲で選択される。この実施例において、この範囲のアドレスが N D I デバイスによって受取られると、それは、グループアドレスと、アドレスデコードレジスタ 1 0 0 内に記憶されたグループマスクとを比較する。グループアドレスに対応するグループマスク内のビットがセットされると、この発明の N D I デバイスデータチャネルは、そのグループアドレスに関連するコマンドを解釈して、それに従う。たとえば、デバイスインベントリレジスタ内に記憶されたグループマスクが 1 0 0 0 0 0 0 0 1 1 0 0 b i n である場合、N D I デバイスは、グループアドレス 0 0 0 f h e x 、0 0 0 3 h e x 、および 0 0 0 2 h e x に属する。

10

【 0 1 0 7 】

グループアドレススキームは、さまざまなサンプル速度でセンサまたはアクチュエータのグループに対して、時間で決定されるトリガをユーザが設定できるように設計される。表 1 は、異なるサンプル速度を有するセンサのグループと、それらのグループ割当とを示し、表 2 は、デバイスをポーリングするためのシーケンスを示す。これらの表から分かるように、データチャネルは、それらが同時に、かつ異なったサンプル速度でトリガされ得るように、ともにグループ化され得る。

【 0 1 0 8 】

【表 1】

20

表 1

論理アドレス	デバイス	サンプルレート (サンプル/秒)	グループアドレス
16	温度 1	125	1
17	圧力 1	250	1, 2
18	ひずみ 1	500	1, 2, 3
19	ひずみ 2	500	1, 2, 3
20	ひずみ 3	500	1, 2, 3
21	ひずみ 4	500	1, 2, 3
22	加速度計 1	1000	1, 2, 3, 4
23	加速度計 2	1000	1, 2, 3, 4
24	加速度計 3	1000	1, 2, 3, 4
25	加速度計 4	1000	1, 2, 3, 4

30

【 0 1 0 9 】

【表 2】

表 2

コマンド実行時間	アクション
0	Trigger コマンドをアドレス 1 に発行する
	アドレス 16~25 をポーリングする
1 msec	Trigger コマンドをアドレス 4 に発行する
	アドレス 22~25 をポーリングする
1 msec	Trigger コマンドをアドレス 3 に発行する
	アドレス 18~25 をポーリングする
1 msec	Trigger コマンドをアドレス 4 に発行する
	アドレス 22~25 をポーリングする
1 msec	Trigger コマンドをアドレス 2 に発行する
	アドレス 17~25 をポーリングする
1 msec	Trigger コマンドをアドレス 4 に発行する
	アドレス 22~25 をポーリングする
1 msec	Trigger コマンドをアドレス 3 に発行する
	アドレス 18~25 をポーリングする
1 msec	Trigger コマンドをアドレス 4 に発行する
	アドレス 22~25 をポーリングする
1 msec	シーケンスを再開する

10

20

【 0 1 1 0 】

グループアドレスに関しては、コントローラがグループアドレスを送信すると、この発明のアドレスデコーダ 1 0 0 およびコマンドデコーダ 1 0 2 がコマンドのアドレス部分をデコードし、グループアドレスと、それ自体に記憶されたグループマスクとを比較する。ネットワークデバイスがグループアドレスの支配下にあることをグループマスクが示すと、そのグループアドレスに関連するコマンドを、適切なデータチャネルがネットワークデバイス上で実行する。

30

【 0 1 1 1 】

上述のように、プロトコルとともに用いられるアドレス指定方法により、コントローラは、コマンドおよびデータをネットワークデバイスの 1 つまたはいくつかのいずれかに送信することができ、データの異なった形式は、B i S e n S y s または U A R T のどちらが用いられているかに依存して、送信される。以下に、ネットワークバスを介して通信するために一般に用いられるコマンドを示す。上述のように、この発明のプロトコルは、コマンドおよび応答メッセージが、任意のコマンドを実行するのに必要とされるデータ量に依存して長さを変化させ得るように、効率を最大にするように設計される。このことを達成するために、この発明のプロトコルは、3 レベルのユーティリティを設ける。これらのコマンドを表 3 に示す。

40

【 0 1 1 2 】

【表 3】

表 3

コマンド(16進)	コマンド記述
サービスコマンド	
00	No Op
01	Built in Test
02	Reset
03	Read Status Register
04	Device Inventory Enable
05	Device Inventory
06	Control Pass
07	Wake
08	Sleep
09	E-Calibration
0A	Z-Calibration
0B	Synchronize
0C	Baud Select
0D-0F	Reserved
データコマンド	
20	Trigger
21	Trigger and Read
22	Read In-Data Register Word
23	Read In-Data Stack Word
24	Read In-Data Stack Block
25	Query In-Data/Out-Data Stack Depth
26	Write Out-Data Stack Word
27	Write Out-Data Stack Word/Acquire to In-Data Register
28	Write Out-Data Stack Block
29-2F	Reserved
メモリコマンド	
30	Set Memory Pointer
31	Read Memory Word with Current Pointer
32	Read Memory Block with Current Pointer
33	Write Memory Word with Current Pointer
34	Write Memory Block with Current Pointer
35	Read Memory Word with Passed Pointer
36	Read Memory Block with Passed Pointer
37	Write Memory Word with Passed Pointer
38	Write Memory Block with Passed Pointer
39-7F	Reserved

【0113】

コマンドの1つのレベルはサービスコマンドである。サービスコマンドは、ネットワークハウスキーピング、ネットワーク・デバイス・インターフェイス状態、電源管理、較正、およびバスマスタ裁定のためのものである。これらのコマンドを、以下に簡潔に説明する。たとえば、No Op コマンドは、この発明のNDIデバイスに対し、アクションを取らないように命令する。No Op コマンドは一般に、バス制御を維持するために、演算の初期化において、またはマスタもしくはネットワークコントローラによって、用いられる（

10

20

30

40

50

図 1 参照)。

【 0 1 1 4 】

Built-In-Test (B I T)、E - Calibration、および Z - Calibration コマンドは、ネットワークバスに接続された N D I デバイスにセルフテストを行なうために用いられる。たとえば、B I T コマンドは、N D I デバイスに対し、内部回路のチェックを行なうよう指令する。E - Calibration コマンドは、励起較正を強制して、センサネットワークデバイスからの入力基準電圧に置換えられ、N D I デバイスが基準電圧における N D I デバイスの較正值を測定するために読出しを行なう。同様に、Z - Calibration コマンドは、ゼロ較正の測定を開始し、センサデータチャネルに対する入力が短絡される。N D I デバイスは、データチャネルのオフセット値を測定するために読出しを行なう。

10

【 0 1 1 5 】

プロトコルは、2 種類のリセットコマンド、すなわち、Reset および Synchronize も含む。Reset コマンドは、ネットワークバスに接続された、この発明のネットワーク・デバイス・インターフェイスのすべてをパワーアップ状態に初期化する。この Reset コマンドは一般に、必要時にバスおよびネットワークデバイスをリセットして、ネットワークバスの制御を取り戻すために、コントローラによって用いられる。

【 0 1 1 6 】

Synchronize コマンドは、いくつかの N D I デバイス間で同期を確立して維持するために重要なコマンドである。特に、上述のように、A / D および D / A コンバータの中には、コントローラによって供給される同期クロック信号とは異なる、連続したクロック信号を必要とするものがある。このクロック信号は、ローカル発振器からのクロック信号または同期バスクロック信号のいずれかを分割する分割器によって供給される。この分割されたクロック信号は、データを変換するために、コンバータによって用いられる。分割器はコンバータによって必要とされる適切なクロック周波数を供給するが、異なった N D I デバイス上にこれらの異なった分割器が存在することにより、1 つの N D I デバイス上のコンバータが別の N D I デバイス上のコンバータと同期されないことがあり得る。特に、コンバータのすべては同一周波数で作動できるが、分割器は位相がずれていることがあり得る。

20

【 0 1 1 7 】

このことに照らして、Synchronize コマンドは、複数の N D I デバイス間で、分割されたクロック信号を同期させる。特に、図 9 を参照すると、分割器が同期クロック信号を用いており、かつ、Synchronize コマンドが発行されると、N D I デバイスは、そのクロック分割器を制御して、クロック信号の生成をリセットし、再開する。N D I デバイスのすべてが、同期コマンドで用いられるグループアドレスまたはグローバルアドレスに従って、それらのクロック分割器の同じリセットおよび再開を同時に行なう。

30

【 0 1 1 8 】

図 1 1 a、図 1 1 b、および図 1 1 c は、この発明の N D I デバイスが、どのように取付けられて異なるデータチャネルを制御することができるかの代替的实施例を示す。データチャネルの中には、図 1 1 A に示されるように、データを正確なタイミングで取得するために、N D I デバイスから変換信号のみを必要とするものがある。データチャネルの中には、図 1 1 B におけるように、データを正確なタイミングで取得するために、同期された分割クロック信号および可能性として同期信号のみを必要とするものがある。データチャネルの中には、図 1 1 C におけるように、正確なタイミングでデータを取得するために、変換信号および分割クロック信号の両方を必要とし得るものがある。

40

【 0 1 1 9 】

図 1 1 A ~ 図 1 1 C を参照すると、Synchronize コマンドの使用が、異なるネットワークデバイス間のデータチャネルにおいてスイッチド・キャパシタ・フィルタ 1 2 6、シグマ/デルタ A / D コンバータ 1 2 0、またはデジタルフィルタ 1 1 8 等をすべて同期に作動させることに関して示される。クロック分割器がクロック信号を生成するためにローカル発振器を分割する場合に、この分割器は、同期コマンドのパリティビットの中央エッジ

50

にてリセットし、再開する。これにより、複数のNDIデバイス間のクロック分割器は、ともに同期する。しかしながら、ローカル発振器の周波数の変化により、クロックが経時的に位相を外れてドリフトすることを認識されたい。クロック分割器に対するソースがバスクロックである場合、クロック分割器は図9のタイミングに従って同期し、分割されたクロックは、その後同期して稼働し続ける。NDIデバイスの本実施例は、シグマ/デルタA/Dコンバータ120、デジタルフィルタ118、または他のデバイスのリセットまたは同期を行なうために、同期信号116、124、および128をそれに取付けられたデータチャネルに与える。

【0120】

表3を参照すると、プロトコルは、Wake および Sleep コマンドも含む。Sleep コマンドは、ネットワークデバイスの特定の部分をパワーダウンさせ、Wake コマンドは、それらをパワーアップする。さらに、プロトコルは、Read Status Register コマンドを含む。このコマンドは、NDIデバイスの状態レジスタの読出しを生じ、ネットワークデバイスが忙しいか否か、それが特定のコマンドを指示するか否か、コントローラがデータスタックに要求したデータが多すぎるかまたは少なすぎるか、メッセージ送信エラーが生じたか否か、メモリが保護されていないか否か等の情報を供給する。Control Pass コマンドは、ネットワークバス上に2つ以上のバスコントローラがある場合に、どのバスコントローラがバスの制御下にあるかを確立するために、マスタおよびネットワークコントローラによって用いられる。さらに、Baud Select コマンドは、非同期モードで作動している際にネットワークバス上のボーレートを変更するために、コントローラによって用いられる。

【0121】

Device Inventory Enable コマンドおよび Device Inventory コマンドは、各ネットワークデバイスの一意的な識別、ならびに論理アドレスおよびグループマスクのデータチャネルへの割当に関して上で論じた。Device Inventory Enable コマンドは、一覧にされるべきネットワークデバイスのグループを選択する。Device Inventory コマンドは、ネットワークデバイスの識別ならびに論理アドレスおよびグループマスクの割当に関するほとんどのアクションを制御するために用いられる。このコマンドは、コマンドのアドレスフィールドが、さまざまなデバイスインベントリ関数をデバイス内に方向づけるために用いられる点において異なる。たとえば、これらの関数は、新規の Word Search 競合に入ること、ならびに No Dropout および 1's Dropout と関連するアクションを含む。さらに、このコマンドは、Word Search に勝ったデバイスのメモリから読出すこと、ならびに論理アドレスおよびグループマスクの書込みを含む。

【0122】

コマンドの第2の種類はデータタイプコマンドである。これらのコマンドは、時間で決定されるデータ取得および制御に適合される。ネットワーク効率は、NDIデバイスが、1つのデータポイントを直接動かすか、または規定された長さのブロック転送として2つ以上のデータポイントを動かすことを可能にすることによって最大にされる。たとえば、Trigger コマンドは、センサデータチャネルで入来するデータの測定を開始するか、または、アクチュエータのデータチャネルで物理量へのデータ変換を生じるために、コントローラによって用いられる。図3Bを参照すると、この発明のネットワーク・デバイス・インターフェイスのチャネルの各々は、入力データレジスタ89および入力データスタック88を含む。Trigger コマンドがNDIデバイスによって受取られると、入力データレジスタの内容は、その入力データスタックの上部にプッシュされる。次に、それはデータチャネルから新規の読出しを行なう。読出しが終了すると、新規の読出し値は、入力データレジスタ内に存在する。バスコントローラは通常、入力データスタックからデータを読出す。

【0123】

入力データスタックからデータを取得することにより、バスコントローラに送られるセンサ読出し値の1つ以上のサンプルのレイテンシが生じるが、この配置により、Read-in-

10

20

30

40

50

data-stack コマンドが、Trigger コマンドの直後にセンサデータチャネルに発行され得る。Read in-data stack コマンドは、入力データスタックからデータを読み出す。このことは、次いで、ネットワークバス帯域幅を最大限に用いることを可能にし、これは、センサデータチャネルが長い変換時間を有する場合に重要である。このことは、また、ユーザが書込まなければならないTrigger および Read コマンドの組を単純化する。なぜなら、入力データスタックから読み出す前に、休止を含む必要がないからである。

【 0 1 2 4 】

Read In-Data Register Word コマンドは、それがデータチャネルから利用可能になった直後に入力データレジスタからのデータの読み出しを可能にする。バスコントローラは、データが有効になる前に、このレジスタから読み出しを試みる必要がない。ネットワークデバイスがアクチュエータである場合、NDI デバイスは、出力データスタックの上部においてデータを値に変換することを指令し、次に、出力データスタックをポップする。ここに図示されていないNDI デバイスの現時点の実施例において、NDI デバイスに出力データスタックは存在しない。データはバスコントローラからNDI デバイスに送られて、データスタックを通過せずにD/Aコンバータまたはデータチャネルに直接至る。しかしながら、NDI デバイスに接続されたD/Aまたはデータチャネルは、1スタックワードのサイズの出力データスタックとして働く、データを保持するためのレジスタを有する。

【 0 1 2 5 】

上で論じたように、Trigger および Read コマンドは、データチャネルの測定サイクルを開始して、前の測定結果をネットワークバス上で直ちに送信するために用いられる。このコマンドに回答して、この発明のNDI デバイスは、同時に、入力データレジスタの内容を入力データスタック上にプッシュし、新規の測定サイクルを開始し、入力データスタックの上部の内容の送信を始める。入力データスタックへのプッシュは、コマンドのデータ同期パターンの送信前および送信中に生じる。このようにして、前の測定結果は、有効となって、NDI デバイスの送信機部分が送信のためにそれにアクセスする際に、データスタックに存在することになる。次に、新規の測定値が取られる間に、データサンプルがネットワークバスを介して送信される。

【 0 1 2 6 】

読み出しコマンドに関し、Read In-Data Register Word コマンドは、入力データレジスタから直接に読み出しを開始する。NDI デバイスは、入力データレジスタの内容をネットワークバス上に配置する。Read In-Data Stack Word コマンドは、入力データスタックの上部からデータワードを読み出すために用いられる。この場合、この発明のNDI デバイスは、データスタックから最新のデータワードをネットワークバス上に出力することによって回答し、より古いデータがスタックの上部にシフトされる。

【 0 1 2 7 】

図3Bに示されるように、この発明のNDI デバイスの各チャネルと関連するデータスタックは、いくつかのレジスタを含む。入力データまたは出力データスタック上にいくつかの有効なデータワードが存在するかを決定するために、Query In-Data/Out-Data Stack Depth コマンドが用いられる。この場合、この発明のNDI デバイスは、そのスタック深さレジスタ内の有効データワード数を記憶する。このレジスタの値は、バスコントローラによって指令されると、NDI デバイスによってバスコントローラに送信される。

【 0 1 2 8 】

Write Out-Data Stack Word コマンドは、この発明のNDI デバイスに対し、データワードを出力データスタックの上部に書込ませる。さらに、Write Out-Data Stack Word/Acquire To In-Data Register コマンドは、さらに、NDI デバイスに対し、同時に、データチャネルからデータを取得させ、それを入力データレジスタ内に入れさせる。データチャネルへの出力信号が、データチャネルからNDI への出力に接続される場合は、このコマンドを、アクチュエータに送信されたデータをバスコントローラにエコーバックするために用いることができる。Write Out-Data Stack Word と同様に、Write Out-Data Stack Block コマンドは、NDI デバイスに対し、複数のデータワードを出力データスタック

に書込ませる。

【 0 1 2 9 】

サービスおよびデータタイプのコマンドに加え、この発明のプロトコルは、メモリコマンドも含む。これらのコマンドは、それに対してデータの書込みまたは読出しが行われ得る、特定の規定された記憶場所または関数へのアクセスを可能にする。これにより、ランダムアクセスデータブロックは、オーバーヘッドをほとんど伴わずに、システム間で効率よく転送され得る。それはまた、直接メモリアクセスおよび/または1つ以上のデータバッファブロックの移動も可能にする。

【 0 1 3 0 】

たとえば、Set Memory Pointer コマンドは、N D I デバイス内にメモリアドレスポインタをセットする。Read Memory Word With Current Pointer コマンドは、メモリアドレスポインタによってポイントされたメモリから1つのワードを読出すために用いられる。Read Memory Word With Current Pointer は、メモリアドレスポインタの現在の値によってポイントされたメモリワードで開始するメモリからデータワードのブロックを読出すために用いられる。コマンドとともにN D I デバイスに渡される引数は、N D I デバイスに対し、いくつのデータワードが読出されるべきであることを教える。各メモリワードがN D I デバイスによってバスコントローラに送信された後に、メモリポインタは、自動的に1ずつインクリメントされる。次に、次のメモリワードが送信される。このプロセスは、バスコントローラによってリクエストされた数のメモリワードが送信されるまで繰返される。

【 0 1 3 1 】

この発明のプロトコルはまた、書込みメモリコマンドも含む。具体的に、Write Memory Word With Current Pointer コマンドは、メモリ内に1ワードを書込む。N D I デバイスは、コマンドを伴う引数を、メモリアドレスポインタによって現時点でポイントされた場所でメモリに書込む。その一方で、Write Memory Block With Current Pointer コマンドは、データワードのブロックをメモリ内に書込む。この発明のN D I デバイスは、コマンドを伴う引数を、メモリアドレスポインタによって現時点でポイントされた場所で始まるメモリ内に書込む。各ワードが書込まれると、メモリアドレスポインタは、1ずつインクリメントされる。このプロセスは、ワードのすべてがメモリ内に書込まれるまで繰返される。

【 0 1 3 2 】

上述のコマンドと同様に、この発明のプロトコルは、コマンドとともにコントローラによって送信されるポインタを用いて、データのブロックおよびワードをメモリから読出し、メモリに書込むためのコマンドを含む。これらのコマンドは、上述のものと同様に作用するが、例外として、ポインタは、コマンドに続く引数として供給される。これらのコマンドは、Read Memory Word With Passed Pointer、Read Memory Block With Passed Pointer、Write Memory Word With Passed Pointer、および Write Memory Block With Passed Pointer である。

【 0 1 3 3 】

上述のように、この発明のN D I デバイスは、コントローラとさまざまなネットワークデバイスとの間でネットワークバスを介した、コマンドおよびデータのデジタル通信を供給する。図10を参照すると、この発明のN D I デバイスを実現するネットワーク化システムが、航空機10において示される。この実施例では、重要な構造上のさまざまな場所を監視するために、ネットワークが用いられる。航空機には、翼付根、翼表面、尾翼付根、尾翼ケーブル、および着陸装置の歪み等の歪み12、ならびに主翼端および尾翼端の加速度等の加速度14を測定するためのネットワークデバイスが配置される。さらに、ネットワークは、音響疲労に対する重要な胴体圧に加え、レーダ、着陸装置、および着陸端部に対する重要な腐食箇所18等のさまざまな重要な構造上の場所における圧力、ならびにエンジンの外枠の温度20を監視するためのセンサ16を含む。この実施例において、ネットワークデバイスのすべては共通バスに接続され、それにより、過剰な配線をなくす。

さらに、データおよびコマンドは、デジタル式に送信されて、雑音による影響が低減される。

【 0 1 3 4 】

加えて、N D I デバイスに対する好ましいプロトコルは、N D I デバイスの小型化を可能にする助けとなるように、ネットワークデータビットにマンチェスター符号化を用いる。非同期のシリアルデータを受信するあらゆるデバイスにおいては、それが、シリアルなデータストリームからデータシーケンスのタイミングを取得する能力を有していなければならないことを理解されるべきである。通常、シリアルな非同期データの受信機はローカル発振器を有しており、その受信機に動作をさせ、かつ、シリアルなデータからタイミング情報を取り戻させなければならない。タイミング情報が抽出されると、非同期受信機は、或る速度で、これらの速度、すなわち所与のこのローカル発振器の周波数から或る一定のプラスまたはマイナスの偏差で、シリアルなデータを受信することができる。シリアルなデータのマンチェスター符号化により、各ビットの中心において、ハイからローまたはローからハイへの遷移を生じる。これにより、シリアルなデータストリームからの必要なタイミング情報の抽出が容易となる。マンチェスター符号化されたシリアルなデータストリームからのタイミング情報の抽出が、非常に容易であるために、ローカル発振器に基づいた、期待されたデータ速度からの比較的大きな偏差を許容することができる。期待されたデータ速度からの比較的大きな偏差に対するこのような許容性により、各N D I 受信機は、精度の低いローカル発振器を用いてもマンチェスター符号化されたデータを受取ることができる。精度の低いローカル発振器は、極めて小さく形成することができる。適切なローカル発振器の本実施例は、僅か約 1×1.5 ミリメートルである。これにより、小型のN D I デバイスを形成する助けとなる。

【 0 1 3 5 】

上述の説明および関連する図面に示された教示の恩恵を受ける、この発明が属する当該技術の当業者は、この発明の多くの変更例および他の実施例を思いつくであろう。したがって、この発明が、開示された特定の実施例に限定されず、変更例および他の実施例が前掲の請求項の範囲内に含まれることを意図するものと理解されたい。特定の用語がこの明細書で用いられているが、それらは限定するためではなく、一般的な説明のためにのみ用いられる。

【図面の簡単な説明】

【 0 1 3 6 】

【図 1】図 1 は、この発明の一実施例による、ネットワークバスを介してコントローラと複数のデータチャネルとの間でコマンドおよびデータをデジタル方式で伝送するためのネットワークシステムのブロック図である。

【図 2】この発明の一実施例による、ネットワークバスを介してコントローラと複数のデータチャネルとの間でコマンドおよびデータをデジタル方式で伝送するために行なわれる動作のブロック図である。

【図 3 A】この発明の一実施例による、ネットワークバス上でデータチャネルとコントローラとの間でデジタル方式でコマンドおよびデータを通信するためのN D I デバイスの一般的なブロック図である。

【図 3 B】この発明の一実施例による、ネットワークバス上でデータチャネルとコントローラとの間でデジタル方式でコマンドおよびデータを通信するためのN D I デバイスの一般的なブロック図である。

【図 4】この発明の一実施例による、遠隔デバイスからデータを検索するためにN D I が行なう動作のブロック図であり、N D I デバイスが同時に複数の作業を行なう一方、同時にバスコントローラと通信する能力を示す。

【図 5】この発明の一実施例による、コントローラから遠隔デバイスに送られたコマンドを、N D I に接続されるコンバータによって使用される特別なコマンドに変換して、コントローラが遠隔デバイスと通信できるようにするために行なわれる動作のブロック図である。

【図 6】この発明の一実施例による、コントローラが同期モードまたは非同期モードのどちらで作動しているかを判断するために行なわれる動作のブロック図である。

【図 7】この発明の一実施例による、コントローラがコマンドおよびデータを伝送しているビットレートを判断するために行なわれる動作のブロック図である。

【図 8】この発明の一実施例による、バスコントローラが論理アドレスおよびグループアドレスをネットワーク・デバイスに割当てているときに、NDI によって行なわれる動作のブロック図である。

【図 9】この発明の一実施例による、複数のNDI デバイスを通じてネットワークに装着される複数のデータチャネルのフリーランニングブロックを同期させるために、NDI デバイスによってデータチャネルに与えられる内部フリーランニングクロックの同期を示すグラフである。

10

【図 10】航空機で実現される、この発明の一実施例による、電氣的ネットワークシステムの概略図である。

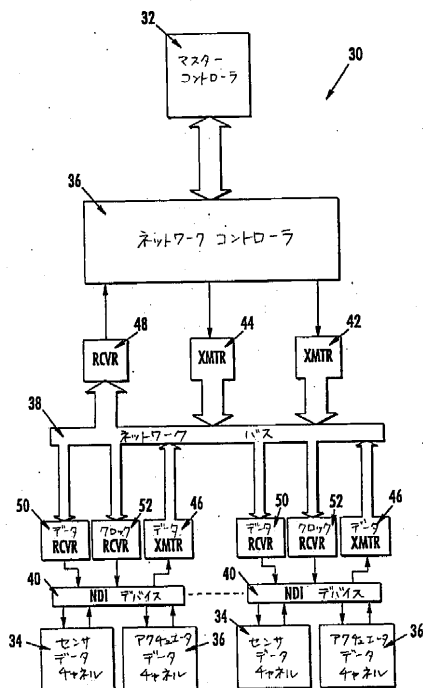
【図 11 A】この発明の一実施例による、NDI デバイスからの変換信号を使用して正しいタイミングでデータを取得する逐次近似 A/D への、この発明のNDI デバイスの接続のブロック図である。

【図 11 B】この発明の一実施例による、正しいタイミングでデータを取得するため、同期された分割クロックを使用し、同期信号を使用する可能性もあるデジタルフィルタおよびデシメータならびにシグマ/デルタ A/D への、この発明のNDI デバイスの接続のブロック図である。

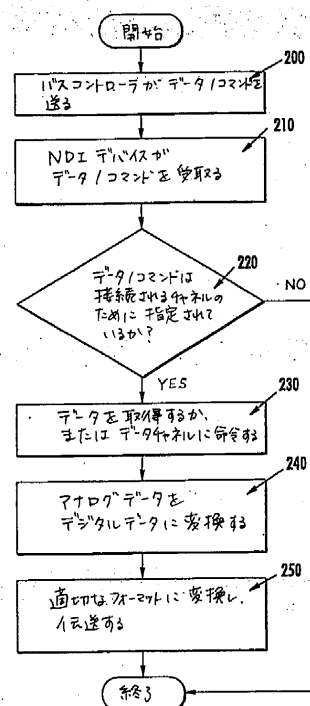
20

【図 11 C】この発明の一実施例による、スイッチド・キャパシタ・フィルタを備える逐次近似 A/D および信号調整への、この発明のNDI デバイスの接続のブロック図であり、スイッチド・キャパシタ・フィルタは、正確なタイミングでデータを取得するために変換信号および分割されたクロック信号の両方を必要とする。

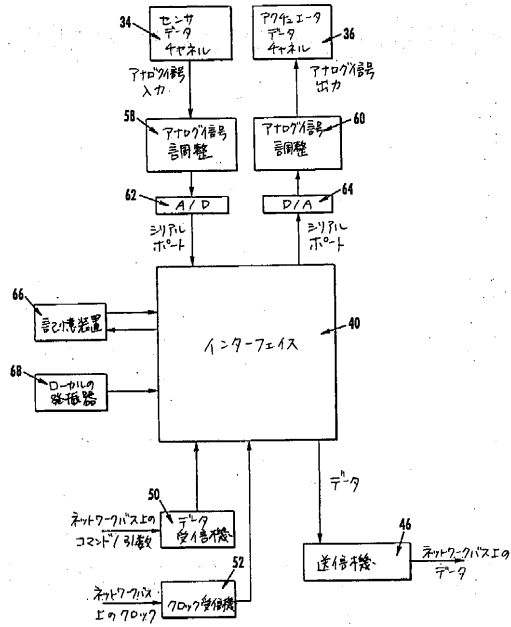
【図 1】



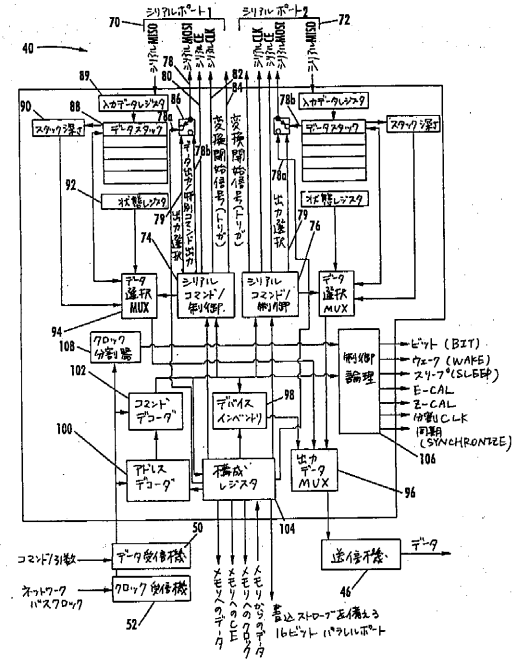
【図 2】



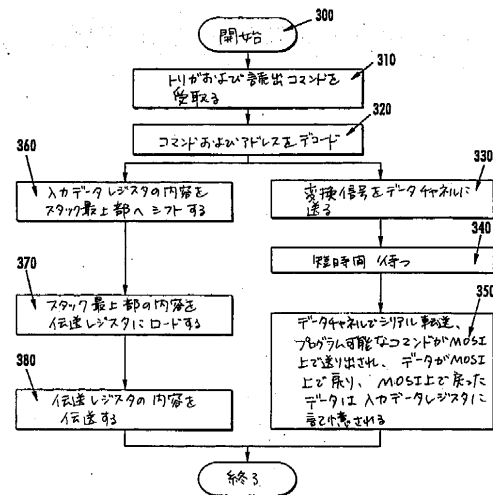
【図 3 A】



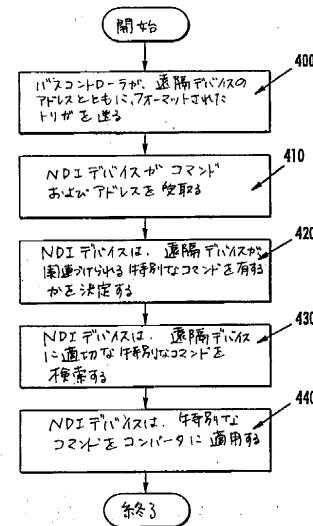
【図 3 B】



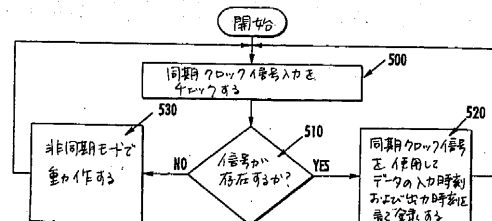
【図 4】



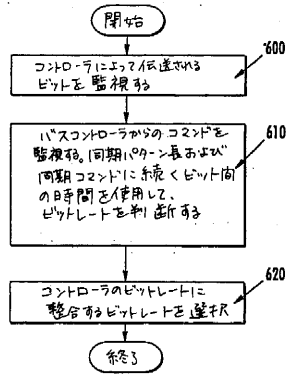
【図 5】



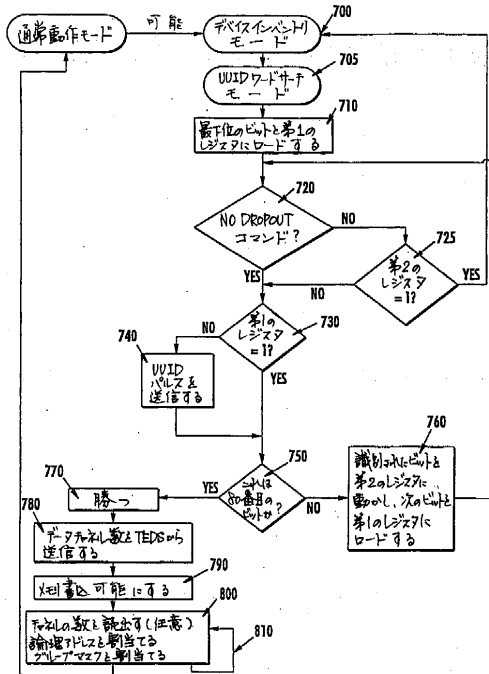
【図 6】



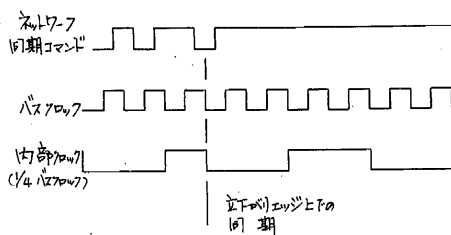
【 図 7 】



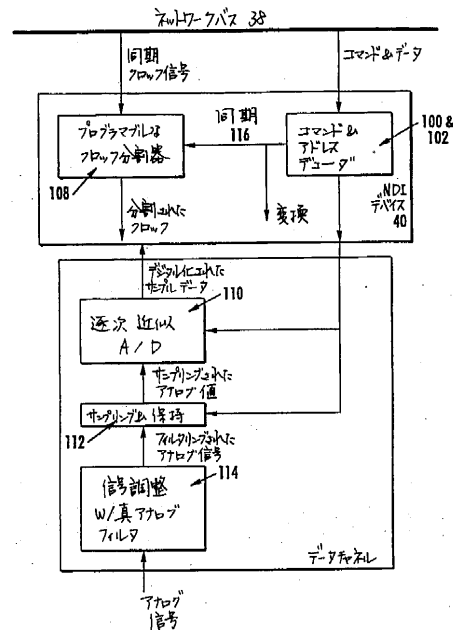
【 図 8 】



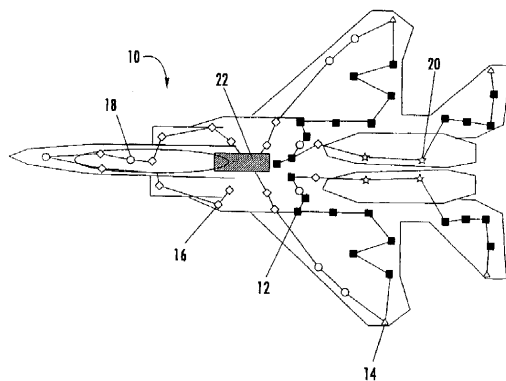
【圖 9】



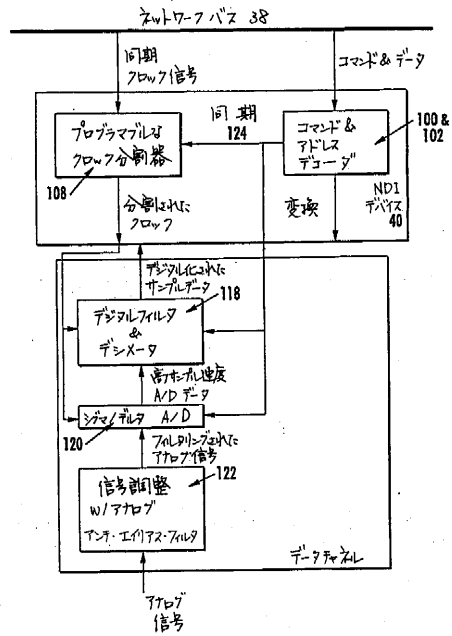
【 図 1 1 A 】



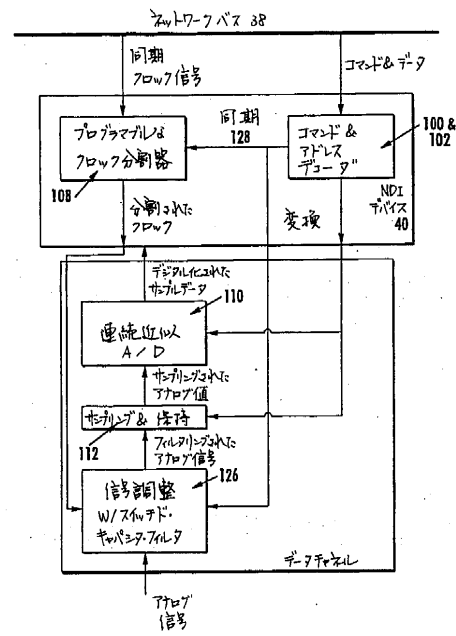
【 図 1 0 】



【図 11B】



【図 11C】



フロントページの続き

- (72)発明者 コンズ, ダニエル・ダブリュ
アメリカ合衆国、6 3 0 3 1 ミズーリ州、フロリサント、キュー・ガーデンズ、1 5 1 3
- (72)発明者 ウィンケルマン, ジョセフ・ピィ
アメリカ合衆国、6 3 3 7 6 ミズーリ州、セント・ピーターズ、エンゲルウッド・コート、3 8
- (72)発明者 グラント, ロバート・エル
アメリカ合衆国、6 3 3 7 6 ミズーリ州、セント・ピーターズ、ジャケッタ、1

審査官 岩田 玲彦

- (56)参考文献 特開2 0 0 0 - 2 3 1 4 0 3 (J P , A)
特開平0 2 - 2 9 5 3 2 5 (J P , A)
特表平0 4 - 5 0 3 5 9 4 (J P , A)

- (58)調査した分野(Int.Cl. , D B 名)
H 0 4 L 1 2 / 4 0 3