

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4721502号
(P4721502)

(45) 発行日 平成23年7月13日(2011.7.13)

(24) 登録日 平成23年4月15日(2011.4.15)

(51) Int.Cl.	F I
H O 4 L 12/56 (2006.01)	H O 4 L 12/56 1 O O Z
G O 6 F 15/163 (2006.01)	G O 6 F 15/163

請求項の数 8 (全 18 頁)

(21) 出願番号	特願2000-341119 (P2000-341119)	(73) 特許権者	398038580
(22) 出願日	平成12年11月8日(2000.11.8)		ヒューレット・パッカード・カンパニー
(65) 公開番号	特開2001-189759 (P2001-189759A)		HEWLETT-PACKARD COMPANY
(43) 公開日	平成13年7月10日(2001.7.10)		アメリカ合衆国カリフォルニア州パロアルト
審査請求日	平成19年6月26日(2007.6.26)		ハノーバー・ストリート 3000
(31) 優先権主張番号	09/436062	(74) 代理人	100087642
(32) 優先日	平成11年11月8日(1999.11.8)		弁理士 古谷 聡
(33) 優先権主張国	米国 (US)	(74) 代理人	100076680
前置審査			弁理士 溝部 孝彦
		(74) 代理人	100121061
			弁理士 西山 清春
		(72) 発明者	クレイグ・ダブリュー・ワーナー
			アメリカ合衆国テキサス州75001, ア
			ディソン, アスペン・コート・14589
			最終頁に続く

(54) 【発明の名称】 ソース定義パケットルーティングシステム及びその方法

(57) 【特許請求の範囲】

【請求項 1】

システムの処理速度を向上させるために、ある特定のソフトウェアアプリケーションをマルチプロセッサによって並列処理させるデータ通信システムであって、

ネットワーク(100)内において相互接続された複数のノードを備え、

前記複数のノードが、ソースノード(203)と、宛先ノード(209)と、少なくとも1つの中間ノード(206)とを含み、

前記ソースノード(203)は、前記マルチプロセッサのうちの1つのプロセッサに接続されたCPUエージェントを含み、前記中間ノード(206)は、少なくとも1つのクロスバと、その他のプロセッサに接続されたCPUエージェントのうちの少なくとも1つを含み、前記宛先ノード(209)は、複数のメモリコントローラのうちの少なくとも1つを含み、該複数のメモリコントローラの各々は、複数のメモリユニットのうちの対応する1つのメモリユニットと、前記CPUエージェントのうちの対応する1つのCPUエージェントとに接続されており、

前記CPUエージェント及び前記メモリコントローラの各々が、

前記ソースノード(203)から前記少なくとも1つの中間ノード(206)を介して前記宛先ノード(209)までのデータ経路を識別するための、前記ソースノード(203)内のソースロジック(SL)であって、前記ソースノード(203)から前記宛先ノード(209)へと伝達されることとなるデータパケット(136)に付与される少なくとも1つの宛先ポート値(149)とカレント

10

20

ホップカウント(146)とのシーケンスによって前記データ経路が指定されていることからなる、ソースロジックと、

前記データパケット(136)の最終的な宛先を検出するための宛先ロジック(DL)とを含み、及び、

前記CPUエージェントの各々が、

前記データ経路に沿って前記データパケット(136)をルーティングするための、前記少なくとも1つの中間ノード(206)内のルーティングロジック(RL)を含み、

前記データパケット(136)内に前記少なくとも1つの中間ノード(206)の少なくとも1つのソースポート値を記録するための戻りルーティングロジック(RRL)を、前記少なくとも1つの中間ノード(206)が含み、

10

前記クロスバが、前記ルーティングロジック(RL)と前記戻りルーティングロジック(RRL)とを含み、及び、

前記データパケット(136)内の前記少なくとも1つの宛先ポート値(149)を前記少なくとも1つの中間ノード(206)のソースポート値に置換するロジックを、前記戻りルーティングロジック(RRL)が更に含むことからなる、データ通信システム。

【請求項2】

前記データパケット(136)に合計ホップ値(143)が付与されることからなる、請求項1に記載のデータ通信システム。

【請求項3】

20

前記ソースノード(203)内に配置されたルーティングテーブル(RT0~6, RTa~RTh)を更に備え、該ルーティングテーブルが、前記ソースノード(203)から前記宛先ノード(209)までの少なくとも1つのデータ経路を含むことからなる、請求項1に記載のデータ通信システム。

【請求項4】

前記カレントホップカウント(146)をデクリメントするロジックを前記ルーティングロジック(RL)が更に含む、請求項1に記載のデータ通信システム。

【請求項5】

システムの処理速度を向上させるために、ある特定のソフトウェアアプリケーションをマルチプロセッサによって並列処理させるデータ通信方法であって、

30

ネットワーク(100)内においてソースノード(203)から少なくとも1つの中間ノード(206)を介して宛先ノード(209)へと伝達されるデータパケット(136)を生成し、

前記ソースノード(203)から前記少なくとも1つの中間ノード(206)を介して前記宛先ノード(209)までのデータ経路をソースロジック(SL)によって識別し、該データ経路が、前記ソースノード(203)から前記宛先ノード(209)へと伝達されることとなる前記データパケット(136)に付与される少なくとも1つの宛先ポート値(149)とカレントホップカウント(146)とのシーケンスによって指定されており、

前記少なくとも1つの中間ノード(206)内において前記データ経路に沿って前記データパケット(136)をルーティングロジック(RL)によってルーティングし、

前記少なくとも1つの中間ノード(206)内において、該少なくとも1つの中間ノード(206)の少なくとも1つのソースポート値を前記データパケット(136)内に戻りルーティングロジック(RRL)によって記録し、及び、

40

前記宛先ノード(209)内において前記データパケット(136)の到着を宛先ロジック(DL)によって検出することを含み、

前記ソースノード(203)は、前記マルチプロセッサのうちの1つのプロセッサに接続されたCPUエージェントを含み、前記中間ノード(206)は、少なくとも1つのクロスバと、その他のプロセッサに接続されたCPUエージェントのうちの少なくとも1つとを含み、前記宛先ノード(209)は、複数のメモリコントローラのうちの少なくとも1つを含み、該複数のメモリコントローラの各々は、複数のメモリユニットのうちの対応する1つのメ

50

モリユニットと、前記CPUエージェントのうちの対応する1つのCPUエージェントとに接続されており、

前記クロスバが、前記ルーティングロジック(RL)と前記戻りルーティングロジック(RRL)とを含み、及び、

前記少なくとも1つの中間ノード(206)内において、該少なくとも1つの中間ノード(206)の少なくとも1つのソースポート値を前記データパケット(136)内に前記戻りルーティングロジック(RRL)によって記録することが、前記データパケット(136)内の前記少なくとも1つの宛先ポート値(149)を、前記少なくとも1つの中間ノード(206)の前記少なくとも1つのソースポート値に置換することを更に含むことからなる、データ通信方法。

【請求項6】

前記データパケット(136)に合計ホップ値(143)を付与することを更に含むことからなる、請求項5に記載のデータ通信方法。

【請求項7】

前記ソースノード(203)から前記少なくとも1つの中間ノード(206)を介して前記宛先ノード(209)までのデータ経路を前記ソースロジック(SL)によって識別することが、前記ソースノード(203)内に配置されたルーティングテーブル(RT0~6, RTa~RTh)を検査することを更に含み、該ルーティングテーブルが、前記ソースノード(203)から前記宛先ノード(209)までの少なくとも1つのデータ経路を含むことからなる、請求項5に記載のデータ通信方法。

【請求項8】

前記少なくとも1つの中間ノード(206)内において前記データ経路に沿って前記データパケット(136)を前記ルーティングロジック(RL)によってルーティングすることが、前記カレントホップカウンタ(146)をデクリメントすることを更に含むことからなる、請求項5に記載のデータ通信方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、一般にデータ通信分野に関し、特にネットワークにおけるパケットの前方及び戻りルーティングのためのデータ経路を規定するシステム及び方法に関する。

【0002】

【従来の技術】

集積化マイクロプロセッサ回路の出現以来、コンピューティングテクノロジーはますます複雑になってきている。最近まで、プロセッサ回路は、単一のプロセッサを含み、データバス及び制御バスに電氣的に結合された付随するメモリを有していた。かかるマシン上で実行されるソフトウェアアプリケーションは、今日の複雑なアプリケーションと比べて比較的単純であった。特に、かかるプロセッサ回路が動作するクロック速度は比較的低速であり、このため、かかるプロセッサ回路上で実行することが可能な種々のアプリケーションのサイズは制限されていた。

【0003】

【発明が解決しようとする課題】

しかし、集積回路テクノロジーが改良された結果としてプロセッサ速度が向上することとなった。これと同時に、より高速なプロセッサを使用して実行されるソフトウェアアプリケーションのサイズ及び複雑さもまた増大した。しかし、コンピュータチップメーカーは、集積回路の速度を更に高速化する新たな挑戦に直面してきた。特に、集積回路内の導体の浮遊容量その他の電氣的特性に関する問題が、高速プロセッサ回路内での高周波数信号のルーティングを困難にしている。

【0004】

上記問題にも関わらず、コンピュータシステムの速度を向上させる1つの方法として、並列処理の採用が挙げられる。即ち、複数のプロセッサ回路が同時に動作して特定のソフトウェアアプリケーションの様々な側面を処理することを可能にするものである。かかるシ

10

20

30

40

50

システムは、1つ又は2つ以上の共有メモリを用いて、様々なアプリケーションに共通のデータについて動作することが可能である。かかる共有メモリを可能にするために、コンピュータシステムは、様々なネットワーク要素を用いて、1つ又は2つ以上のメモリ要素間のデータを複数のプロセッサ回路等に振分ける。しかし、1つ又は2つ以上のネットワーク要素が故障した場合に問題が生じる。一般に、かかる故障が生じると、コンピュータシステムは、他の並列処理要素が依然として動作可能な場合であっても動作を停止する。

【0005】

【課題を解決するための手段】

上述に鑑み、本発明は、マルチプロセッサ相互接続ネットワークで通信を行うためのデータ通信システム及び方法を提供する。一実施形態では、本システムは、マルチプロセッサ相互接続ネットワークにおいて相互接続された複数のノードに配置された複数の論理回路を含む。ここで、該ノードには、少なくとも1つのソースノードと、少なくとも1つの宛先ノードと、少なくとも1つの中間ノードとが含まれる。また前記論理回路は、ソースノード内に配置されたソースロジックを含み、該ソースロジックは、1つ又は2つ以上の中間ノードを介した個々のソースノードと宛先ノードとの間のデータ経路を識別する。該データ経路は、個々のソースノードと宛先ノードとの間で送信されるデータパケットに添付される少なくとも1つの宛先ポート値とカレント（即ち現在の）ホップカウントとによって指定される。論理回路はまた、該データ経路に沿ってデータパケットをルーティングする中間ノード内のルーティングロジックと、データパケットの最終的な宛先を検出する宛先ノード内の宛先ロジックとを含む。また、本発明の様々な実施形態は、データパケット中に戻り経路を記録する戻りルーティングロジックを中間ノード内に含む。

【0006】

別の実施形態では、本発明は、データ通信方法として見ることができる。一般に、該方法は、ネットワークにおいてソースノードから少なくとも1つの中間ノードを介して宛先ノードへと送信するデータパケットを生成し、ソースノードから少なくとも1つの中間ノードを介した宛先ノードまでのデータ経路であって、前記ソースノードから前記宛先ノードへと送信されるデータパケットに添付される少なくとも1つの宛先ポート値とカレントホップカウントとのシーケンスによって指定されるデータ経路を識別し、前記少なくとも1つの中間ノードで前記データ経路に沿って前記データパケットをルーティングし、前記宛先ノードで前記データパケットの到着を検出する、という各ステップを含むものである。

【0007】

本発明の様々な実施形態は、ソースノードが、データ経路を制御し、ひいては個々のソースノードから個々の宛先ノードへのデータパケットの送信にマイクロプロセッサ相互接続ネットワークの何れの要素が採用されるかを制御し、これにより、ネットワーク要素に障害が発生した場合であってもソースノードがデータパケットを宛先へ確実にルーティングすることが可能となる、という点で様々な利点を提供するものとなる。

【0008】

データパケットのヘッダに戻りデータ経路を記録することにより、宛先ノードは、例えば該データパケットがとったのと同じデータ経路で、応答パケットをソースノードへとルーティングすることができる。これにより、ソースノードは、ネットワーク要素に障害が発生した場合であっても、宛先ノードのルーティングテーブルの状態にかかわらず、データパケット及びそれに対応する応答パケットを確実にルーティングすることが可能となる。その結果として、宛先ノードのルーティングテーブルがまだ初期化されていないという事実が、応答パケットのルーティングに影響を与える、ということがなくなる。

【0009】

更に、本発明の様々な実施形態は、低レイテンシのネットワークを介してデータパケットを操作する方法を提供する。各中間ノード毎の宛先ポート値がルーティング対象となるデータパケットのヘッダに配置されているため、中間ノードでのルーティングテーブルのルックアップは必要ない。

【0010】

10

20

30

40

50

更に、本発明の様々な実施形態は、個々のノードに含まれる機能に応じて中間ノード又は宛先ノードにデータパケットを送信する能力を提供する。

【0011】

本発明のその他の特徴及び利点については、当業者であれば以下に示す詳細な説明及び図面を考察することにより明らかとなろう。かかる更なる特徴及び利点の全てを本発明の範囲内に包含させることが意図されている。

【0012】

本発明は、図面を参照することにより一層良好に理解することができる。同図中の構成要素は必ずしも正しいスケールで描かれたものではなく、本発明の原理を明確に説明するために強調が施されたものである。更に、同図中では、同様の符号は、幾つかの図面全体を通して対応する構成要素を示している。

【0013】

【発明の実施の形態】

ここで図1を参照する。同図には、本発明の一実施形態によるマルチプロセッサ相互接続ネットワーク100（以下「ネットワーク100」と称す）が示されている。該ネットワーク100は、並列に動作する多数のプロセッサ103を含んでいる。該プロセッサ103は、例えば、命令を翻訳して実行する中央処理装置又はマイクロプロセッサ等から構成することが可能なものである。プロセッサ103は、図示のように多数の中央処理装置（CPU）エージェントA0～A7に電気的に接続されている。各CPUエージェントA0～A7には最大で4つのプロセッサ103がリンクされているが、個々のCPUエージェントA0～A7に5つ以上のプロセッサをリンクさせることも可能である。各CPUエージェントA0～A7は、それぞれ2つのメモリコントローラMC0～MC7にリンクされている。例えば、CPUエージェントA0,A1は共にメモリコントローラMC0,MC1にリンクされている。同様に、図示のように、CPUエージェントA2,A3は共にメモリコントローラMC2,MC3にリンクされ、CPUエージェントA4,A5はメモリコントローラMC4,MC5にリンクされ、CPUエージェントA6,A7はメモリコントローラMC6,MC7にリンクされている。また、ネットワーク100は、各メモリコントローラMC0～MC7にリンクされた複数のメモリM0～M7を含んでいる。

【0014】

ネットワーク100はまた、第1のクロスバ X_0 及び第2のクロスバ X_1 も含んでいる。図示のように、CPUエージェントA0～A7並びに第1及び第2のクロスバ X_0 , X_1 は共に、複数のデータポート106を含んでいる。データポート106は、各CPUエージェントA0～A7及びクロスバ X_0 , X_1 において番号が付されている。CPUエージェントA0～A7及びクロスバ X_0 , X_1 は、様々なパラレルバス等を用いて様々なデータポート106を介して互いにリンクされている。更に、各CPUエージェントA0～A7は、同様に各データポートを介して各メモリコントローラMC0～MC7にリンクされている。プロセッサ103と、CPUエージェントA0～A7と、メモリコントローラMC0～MC7と、メモリM0～M7との間の相互接続によって、各プロセッサ103は、特定のソフトウェアアプリケーションの実行中にメモリM0～M7のうちの任意の1つからデータを取得することが可能となる。メモリM0～M7に格納されプロセッサ103によりアクセスされるデータは、複数のキャッシュラインへと編成される。各キャッシュラインは、例えば、16～128バイトのデータから構成することができ、当業界で周知のように、各メモリM0～M7には多数のキャッシュラインが存在する。該キャッシュラインは、ネットワーク100の特定の設計等に応じて、上記範囲よりも多い（又は少ない）バイトのデータから構成することも可能であることが理解されよう。

【0015】

CPUエージェントA0～A7の各々は、ソースロジックSL、ルーティングロジックRL、宛先ロジックDL、戻りルーティングロジックRRL、戻り経路再構成ロジックRRR（以下「再構成ロジックRRR」と称す）、及び個々のルーティングテーブルRT0～7を有している。これらの要素は、ネットワーク100内でのデータパケットの転送を容易にするために用いられる。また、メモリコントローラMC0～MC7は、ソースロジックSL、宛先ロジックDL、再構成ロジックRRR、及び個々のルーティングテーブルRTa～hを有している。最後に、クロスバ X_0 , X_1

は、ルーティングロジックRL及び戻りルーティングロジックRRLを有している。ソースロジックSL、宛先ロジックDL、ルーティングロジックRL、戻りルーティングロジックRRL、再構成ロジックRRR、及びルーティングテーブルRT0～7/RTa～hの機能については、以降で詳述することとする。

【0016】

ネットワーク100の動作を説明するために、キャッシュライン要求の一例について説明する。以下の説明では、プロセッサ103のうちの1つがメモリM0～M7のうちの1つに格納されている特定のキャッシュラインを取得する必要があるものとする。第1に、プロセッサ103のうちの1つが、メモリM0～M7のうちの1つに格納されている特定のキャッシュラインの要求を、各プロセッサ103とインタフェースする対応するCPUエージェントA0～A7へ送信する。次いで各CPUエージェントA0～A7は、ソースロジックSLを使用して、データパケットという形でキャッシュライン要求を生成する。該データパケットは、所望のキャッシュラインを格納している特定のメモリM0～M7とインタフェースするメモリコントローラMC0～MC7の各1つに送信されることになる。各CPUエージェントA0～A7は、ネットワーク100を介した各メモリコントローラMC0～MC7への適当なデータ経路を決定するために参照するルーティングテーブルRT0～7を有している。かかる特定のデータ経路は、ソースロジックSLに従ってデータパケットのヘッダに含まれるルーティングテーブルRT0～7中の複数のパラメータによって指定される。

【0017】

次いで該データパケットは、クロスバ X_0 又は X_1 の一方とCPUエージェントA1～A7のうちの別の1つとを介して各メモリコントローラMC0～MC7へと送信される。このため、クロスバ X_0, X_1 は、該クロスバ X_0, X_1 の各々に存在するルーティングロジックRLに従って、第1のCPUエージェントA0～A7から第2のCPUエージェントA0～A7へと特定のデータパケットをルーティングする機能を実行する。送信用のデータパケットの生成に加えて、CPUエージェントA0～A7はクロスバとしても作用し、該CPUエージェントA0～A7の各々は、クロスバ X_0, X_1 間及びメモリコントローラMC0～MC7に対するデータパケットのルーティングを行うルーティングロジックRLを有している。このため、CPUエージェントA0～A7は、それら各々のプロセッサ103に対するサービスを行い、必要に応じてクロスバとして作用する、という二重の能力で作用するものとなる。

【0018】

クロスバ X_0, X_1 及びCPUエージェントA0～A7はまた、データパケットがデータ経路に沿って各宛先へと進行する際に該データパケット中に戻り経路を記録する働きをする戻りルーティングロジックRRLも有している点に留意されたい。

【0019】

各メモリコントローラMC0～MC7は、各CPUエージェントA0～A7からキャッシュラインの要求を受信すると、宛先ロジックDLを使用して、該要求が実際にデータパケットの最終的な宛先であることを決定する。次いで、各メモリコントローラMC0～MC7は、応答パケットを生成することにより応答を行う。該応答パケットは、戻り経路に沿って送信側のCPUエージェントA0～A7に返信され、これにより、特定のキャッシュラインの要求を含むデータパケットがその最終的な宛先に到達したことが送信側のCPUエージェントA0～A7に通知される。

【0020】

しかし、他の情報を中継するデータパケットをCPUエージェントA0～A7と同様にメモリコントローラMC0～MC7において発生させることが可能である点に留意されたい。特に、メモリコントローラMC0～MC7は、ネットワーク100を介して送信されるデータパケットの特定のデータ経路を決定するために参照するルーティングテーブルRTa～hを有している。このため、ネットワーク100は、任意の2つのCPUエージェントA0～A7間、任意の2つのメモリコントローラMC0～MC7間、及びCPUエージェントA0～A7とメモリコントローラMC0～MC7との間等の通信を容易にする。また、後に明らかとなるように、クロスバ X_0, X_1 がソースロジックSL、宛先ロジックDL、及び再構成ロジックRRRを有する場合には、クロスバ X_0, X_1 と

CPUエージェントA0～A7又はメモリコントローラMC0～MC7のうちの何れか1つとの間に通信を確立することが可能である。

【0021】

次に図2を参照する。同図には、ルーティングテーブルRT0～7/RTa～hの全ての一例として、CPUエージェントA0内に存在するルーティングテーブルRT0が示されている。該ルーティングテーブルRT0は、宛先カラム123、クロスバカラム126、複数の宛先ポートカラム129、及び合計ホップ値カラム133を有している。ルーティングテーブルRT0中の各データ経路エントリ毎に、宛先カラム123は、データパケットがCPUエージェントA0から送信されることになるネットワーク100(図1)中の特定の要素を示す。ネットワーク100中の任意の2つの要素間には2つ以上のデータ経路が存在し得るが、他のパスに優先するパスを使用することが好ましい。特に、ネットワーク100のパスレイテンシを低減させるよう一般に最短のデータ経路が好ましい。したがって、ルーティングテーブルRT0～7/RTa～hにおける類似したパスのうち特定の要素に対する最も好ましいパスを最初に列挙することが可能である。

10

【0022】

クロスバカラム126は、最終的な宛先である要素に到達するためにデータ経路中で用いられる特定のクロスバ X_0 又は X_1 を示す。宛先ポートカラム129は、各データパケットがデータ経路に沿って特定のクロスバ X_0 若しくは X_1 又は特定のCPUエージェントA0～A7を出る際に通過する特定の宛先ポートを示す。最後に、合計ホップ値カラム133は、特定のデータパケットがその最終宛先に到達するためにネットワーク100を介して一要素から一要素へと通過する際のホップの総数を示す。1つの特定のホップは、ネットワーク100における1つの要素から別の要素への1つの特定のデータパケットの移動により構成されるものである。

20

【0023】

ここで図3を参照する。同図には、ネットワーク100中の1つの要素から次の要素へとデータを送信するために用いることが可能なデータパケット136の一例が示されている。該データパケット136は、合計ホップ値143、カレントホップカウント146、第1宛先ポート値149a、及び第2宛先ポート値149bを含む。該データパケット136はまたデータペイロード153を含む。一般に、該データペイロード153は、データパケット136の実質的なデータを含むものである。合計ホップ値143、カレントホップカウント146、第1宛先ポート149a、及び第2宛先ポート149bは、ネットワーク100内のソース要素から宛先要素へとデータパケット136をルーティングするためにネットワーク100により用いられる。なお、ネットワーク100の特性に応じて、1つの特定のデータパケット内には、図示よりも多数又は少数の宛先ポート値149が存在することが可能である(ここでは2つの宛先ポート値149a, 149bが一例として示されている)。

30

【0024】

ここで図4を参照する。同図には、本発明の一実施形態によるデータ経路200の特定の例が示されている。該データ経路200は、図示のようにCPUエージェントA0からメモリコントローラMC7まで延びている。以下の説明を容易にするために、ネットワーク100(図1)は、データパケット136(図3)の通信を容易にする複数ノードからなるネットワークであると考えることができる。データ経路200に関し、CPUエージェントA0はソースノード203として機能し、クロスバ X_0 及びCPUエージェントA6は中間ノード206として機能し、メモリコントローラMC7は宛先ノード209として機能するものとなる。

40

【0025】

一般に、本発明の目的のため、ソースノード203を、CPUエージェントA0～A7(図1)又はメモリコントローラMC0～MC7(図1)のうちの何れか1つから構成することが可能である。更に、中間ノード206は、CPUエージェントA0～A7のうちの何れか1つ又はクロスバ X_0 , X_1 のうちの何れか1つから構成することができる。最後に、宛先ノード209は、CPUエージェントA0～A7のうちの何れか1つ又はメモリコントローラMC0～MC7のうちの何れか1つから構成することができる。所与のデータ経路について、CPUエージェントA0～A7、クロス

50

バ X_0 若しくは X_1 、又はメモリコントローラMC0～MC7の、ソースノード203、中間ノード206、又は宛先ノード209の何れかとしての状態は、特定のデータ経路200がどこで開始し終了するかによって決まる。

【0026】

しかし、最終的には、メモリコントローラMC0～MC7、CPUエージェントA0～A7、及びクロスバ X_0 、 X_1 の何れかが、それらに含まれる論理的な機能に依存して、ソースノード203、中間ノード206、又は宛先ノード209として作用することになる点に留意されたい。

【0027】

以下の説明は、本発明を更に説明するために、図4に示すようにソースノード203から中間ノード206を介して宛先ノード209へと移動する特定のデータパケット136を追跡したものである。ソースノードでは、データパケット136が、宛先ノード209に送信されるように生成される。例えば、ソースノード203がCPUエージェントA0であると仮定すると、それにリンクされているプロセッサ103は、宛先ノード209（メモリコントローラMC7）にリンクされているメモリM7に格納されているキャッシュラインを要求することができる。この場合、CPUエージェントA0は、ソースノード203として作用し、メモリM7から所望のキャッシュラインを取り出すために宛先ノード209に送信されるデータパケット136を生成する。

【0028】

最初に、ソースノード203は、データパケット136の実質的なデータを構成するキャッシュライン要求を生成する。その後、ソースノード203におけるソースロジックSLは、（CPUエージェントA0内にある）ルーティングテーブルRT0を検査して、宛先ノード209に到達するようデータパケット136が送信されることになるデータ経路200の適切なパラメータを識別する。このため、データ経路200は、ソースノード203から中間ノード206aへの第1のホップと、該中間ノード206aから中間ノード206bへの第2のホップと、該中間ノード206bから宛先ノード209への最終ホップとを含むものとなる。

【0029】

ルーティングテーブルRT0においてデータ経路200のパラメータが識別されると、ソースロジックSLは、該パラメータを、宛先ノード209に送信されることになるデータパケット136中に包含させる。特に、合計ホップ値143、カレントホップカウンタ146、並びに第1及び第2の宛先ポート値149a / 149bが、データパケット136中の所定の位置に付加される。データパケット136（図2）内におけるこれらパラメータの位置は、ルーティングロジックRL、戻りルーティングロジックRRL、再構成ロジックRRR、及び宛先ロジックDLにより該パラメータを識別することができるヘッダその他の所定の位置である。

【0030】

その後、該データパケット136は、ソースノード203から第1の中間ノード206aに送信される。パケット136が第1の中間ノード206aにより受信されると、該第1の中間ノード206aのルーティングロジックRLが、データパケット136を検査して、カレントホップカウンタ146により保持されている値を決定する。次いで、該ルーティングロジックRLは、宛先ポート値149a / 149bを介して索引付けを行い、前記カレントホップカウンタ146の値とそれに対応する宛先ポート値149a / 149bとの関係を引き出す。本質的には、カレントホップカウンタ146は、データパケット136を受信したカレント（即ち現在の）中間ノード206a / 206bと関連付けされた特定の宛先ポート値149a / 149bを指すものであり、これにより、データパケット136を送って次のホップを達成するための適切な宛先ポートが決定される。

【0031】

図4に示すように、第1の中間ノード206aにより受信されたデータパケット136のカレントホップカウンタ146は「2」に等しい。このため、データ経路200の最初の時点では、カレントホップカウンタ146は、合計ホップフィールド143に配置された総ホップ数「2」に等しい。ルーティングロジックRLは、カレントホップカウンタ146に基づき様々な宛先ポート値149a / 149bを介して戻る方向に索引付けを行う。図4の例では、カレントホップカウンタ146は「5」に等しい第1の宛先ポート値149を示している。

【0032】

10

20

30

40

50

したがって、図4の例によれば、「2」のカレントホップカウント146は、「5」の宛先ポート値149aを示すが、カレントホップカウント146は、他の順序に従って宛先ポート値149a / 149bを介して索引付けすることも可能である。例えば、宛先ポート値149の位置は、データパケット136内でカレントホップカウント146と交換することが可能である。ルーティングロジックRLの索引付け機能は、マルチプレクサの出力に宛先ポート値149a / 149bの一方を生成するマルチプレクサの選択入力にカレントホップカウント146を適用することにより達成することが可能である点に留意されたい。この場合、宛先ポート値149a / 149bは、複数の特定のマルチプレクサ入力のうちの1つ等にルーティングされる。次いで、「5」の宛先ポート値により、ルーティングロジックRLがクロスバX₀の第5の宛先ポート106からデータパケット136を送信することになる。

10

【0033】

このクロスバX₀の第5の宛先ポート106からのデータパケット136の送信に加え、ルーティングロジックRLは、カレントホップカウント146を値1でデクリメントすることにより、データパケット136が第1のホップを通過したことを示す。このデータパケット136のカレントホップカウント146の変化は、該データパケット136が第1の中間ノード206aを離れる際に該データパケット136に現れる。

【0034】

第1の中間ノード206aからのデータパケット136は、図示のように第2の中間ノード206bにより受信される。該第2の中間ノード206bにおける宛先ロジックDLは、カレントホップカウント146を検査し、それが「0」に等しくないため、該データパケット136を無視する。また、第2の中間ノード206bにおけるルーティングロジックRLは、カレントホップカウント146を検査して、該第2の中間ノード206bによりデータパケット136が送信されるべき適当な宛先ポート149a / 149bを決定する。カレントホップカウント146が「1」であるため、第2の中間ノード206bにおけるルーティングロジックRLは、「0」の宛先ポート値149bを適当な宛先ポート106として識別し、該宛先ポートを介して宛先ノード209へとデータパケット136を送信する。ルーティングロジックRLはまた、カレントホップカウント146をもう1回デクリメントし、その結果としてカレントホップカウント146が「0」となる。

20

【0035】

宛先ノード209は、データパケット136を受信すると、宛先ロジックDLを用いてカレントホップカウント146を検査して、該宛先ノード209が実際にデータパケット136の最終的な宛先であるか否かを決定する。カレントホップカウント146が「0」に等しいため、宛先ノード209における宛先ロジックDLは、これに応じて、宛先ノード209内の他の機能に対し、該宛先ノード209が実際にデータパケット136の最終的な宛先であると通知する。次いで、宛先ノード209は、その中のデータペイロード153にアクセスして、データパケット136が中継する情報に関連する機能を実行する。後述するように、宛先ノード209において再構成ロジックRRRが用いられて、ソースノード203に返信する応答パケットが生成される。

30

【0036】

また、中間ノード206a, 206bは、逆方向にデータ経路200を辿る戻り経路を生成するために用いられる戻りルーティングロジックRRLを含む。該戻りルーティングロジックRRLにより生成される戻り経路を識別するパラメータは、好ましくは、データパケット136が中間ノード206a, 206bを通過して移動する際に該データパケット136中に格納される。したがって、データパケット136が宛先ノード209により受信されると、その戻り経路は、既に該データパケット中に含まれており、該データパケット中の戻り経路を、宛先ノード209が、ソースノード203等に返信される応答パケット又は他のパケットを生成する際に使用する。

40

【0037】

戻り経路を生成するために、各中間ノード206a / 206bにおける戻りルーティングロジックRRLは、本質的に、データパケット136を受信した各ソースポート206a / 206bを指定する。次いで、該ソースポートの番号指定が、データパケット136中に記録される。この番号指定は、特定の中間ノード206a / 206bに関連付けされた宛先ポート値149a / 149b上にソースポートを単に書き込むことにより行われる。代替的には、データパケット136のヘッダそ

50

他の部分中の新たな場所に上記と同じソースポート値を書き込むことも可能である。

【 0 0 3 8 】

更なる説明のため、データパケット136が第1の中間ノード206aに入ったものと仮定すると、戻りルーティングロジックRRLは、データパケット136を受信したソースポート106の追跡を継続する。カレントホップカウンタ146が「0」に等しくないと仮定すると、ルーティングロジックRLにより、データパケット136について、第1の中間ノード206aにおける適当な宛先ポート106が決定される。次いで、戻り経路ロジックRRLは、ソースポート（この場合には第0ポート）を、上述のようにして識別されたデータパケット136中の宛先ポート値149a（「5」）上に書き込む。これは、データパケット136がルーティングされる際に通過する中間ノード206a / 206bで宛先ポート値149a / 149bが識別された場合には、特定のデータパケット136が特定の中間ノード206a / 206bの宛先ポート値149a / 149bに関する情報を保持する必要がないからである。第2の中間ノード206bにおいて「0」の宛先ポート値149bに「3」が上書きされる点に留意されたい。

10

【 0 0 3 9 】

更に、特定の中間ノード206a / 206bのソースポートをデータパケット136中の関連する宛先ポート値149a / 149b上に書き込むことは、戻り経路を記録するためにそれ以上のオーバーヘッドがデータパケット136に追加されないため、有利である。これは、ネットワーク100（図1）全体のデータトラフィック量を低減させるために重要であり、これにより、ネットワーク100のデータ転送レイテンシが改善される。代替的に、ソースポートは、図示のようにデータパケット136内における宛先ポート値149a / 149bとは別個の場所には書き込むことが可能であるが、この場合にはデータパケット136に更なるオーバーヘッドが追加されることになる。戻りルーティングロジックRRLは、中間ノード206a / 206bの全てにおいて同様に動作する。

20

【 0 0 4 0 】

このため、上記説明は、ソースロジックSL、ルーティングロジックRL、戻りルーティングロジックRRL、宛先ロジックDL、及びルーティングテーブルRT0 ~ 7 / RTa ~ hが、ネットワーク100中の任意の1つのソースノード203から任意の宛先ノード209へとデータパケット136を転送するために如何に用いられるかを示したものである。同時に、宛先ノード209からソースノード203へと戻る逆方向にデータ経路200を辿る応答パケットその他のデータパケットについて、戻り経路が識別される。

30

【 0 0 4 1 】

ここで図5を参照する。同図は、宛先ノード209で再構成ロジックRRRの機能が用いられることを説明するためにデータパケット136及び応答パケット136aを示したものである。特に、再構成ロジックRRRは、図示のように宛先ポート149a / 149bの順序を逆にすると共にカレントホップカウンタ146をリセットすることにより、ソースノード203に送信すべき応答パケット136aをデータパケット136から生成する。その結果、応答パケット136aは、第2の中間ノード206bに送信される際に、データパケット136がソースノード203から宛先ノード209へとルーティングされたのと同じ態様でソースノード203へと戻る。

【 0 0 4 2 】

ここで図6を参照する。同図には、本発明の一実施形態によるソースロジックSLのフローチャートが示されている。ソースロジックSLは、CPUエージェントA0 ~ 7及びメモリコントローラMC0 ~ 7内に存在するが、別のネットワーク構成で他の要素がソースロジックSLを採用することも可能である。ソースロジックSLは、一般に、合計ホップ値143（図3）、カレントホップカウンタ146（図3）及び宛先ポート値149（図3）といった適当なデータ経路パラメータを有するデータパケット136（図3）を生成するために用いられる。ソースロジックSLは、ブロック303から開始して、特定の宛先ノード209（図4）にデータパケット136を送信するようトリガされる。次いでソースロジックSLは、ブロック306に移行して、ルーティングテーブルRT0 ~ 7 / RTa ~ h内の適当なデータ経路200を検索する。次いでソースロジックSLはブロック309へ移行し、ソースノード203で、ルーティングテーブルRT0 ~ 7 / RTa ~ hからの適当なデータ経路パラメータを用いてデータパケット136（図3）を生

40

50

成する。その後、ブロック313で、上述したようにデータ経路200（図4）の第1のホップに従ってデータパケット136を中間ノード206a / 206bに送信する。

【0043】

次いでブロック316で、ソースロジックSLは、応答パケットのタイムアウトが発生したか否かを判定する。応答パケットのタイムアウトは、データパケット136が宛先ノード209により正しく受信されたことを示す応答パケット136a（図5）がソースノード203により宛先ノード209から受信されなければならない所定時間の通知(tolling)を含む。応答パケット136aがタイムアウト時間内に受信された場合には、ソースロジックSLは適当に終了する。一方、ブロック316で応答パケットのタイムアウトが存在する場合には、ソースロジックSLはブロック319へと移行し、データパケット136の送信に用いられた特定のデータ経路を、適当なルーティングテーブルRT0~7 / RTa~h内に使用不能なものとして記録する。その後、ソースロジックSLはブロック323へ移行し、宛先ノード209に到達するための次の試行でデータパケット136のコピーを送信するために次の最良のデータ経路200を決定する。その後、ソースロジックSLはブロック309に戻って、後続のデータ経路にデータパケット136を送信する、といった具合である。

【0044】

ここで図7を参照する。同図には、本発明の別の実施形態によるルーティングロジックRLのフローチャートが示されている。該ルーティングロジックRLが中間ノード206a / 206bにより実行されて上述のように特定のデータ経路200に沿ってデータパケット136が案内される。ブロック333から開始して、ルーティングロジックRLは、パケット136（図3）が中間ノード206a / 206bにより受信されたか否かを判定する。パケット136が中間ノード206a / 206bにより受信された場合には、ルーティングロジックRLはブロック336に進み、カレントホップカウント146（図4）を検査して該カレントホップカウント146が「0」に等しい（データパケット136が宛先ノード209に到達したということを示す）か否かを判定する。カレントホップカウント146が「0」に等しい場合には、データパケット136は、それ以上ルーティングされる必要がないため、ルーティングロジックRLは終了する。一方、カレントホップカウント146が「0」よりも大きい場合には、ルーティングロジックRLはブロック339に進む。なお、この場合に、実際の値は必ずしも「0」である必要はなく、ルーティングロジックRLは他の値に応答するよう設計することが可能である。

【0045】

ブロック339で、特定の中間ノード206a / 206bに対応する宛先ポート値が、データパケット136のカレントホップカウント146に基づき同データパケット136中の宛先ポート値149a / 149bを介して索引付けを行うことにより決定される。次いでブロック343で、カレントホップカウント146が値「1」だけデクリメントされる。その後、ルーティングロジックRLはブロック346に進み、次の中間ノード206a / 206b又は宛先ノード209へとデータパケット136を適宜送信する。その後、ルーティングロジックRLは図示のように終了する。

【0046】

ここで図8を参照する。同図には、本発明の別の実施形態による戻りルーティングロジックRRLのフローチャートが示されている。該戻りルーティングロジックRRLは、中間ノード206a / 206b中に配置され、元のデータ経路200に関連する戻り経路のパラメータを生成する。戻りルーティングロジックRRLは、ブロック373から開始し、各中間ノード206a / 206bによりデータパケット136が受信されたか否かを判定する。各中間ノード206a / 206bによりデータパケット136が受信された場合には、戻りルーティングロジックRRLはブロック375に進み、データパケット136を受信したソースポートを一時的に格納する。次いでブロック376で、図7に関して上述したルーティングロジックRLの場合と同様に、カレントホップカウント146を検査してそれが「0」又は他のしきい値と等しいか否かを判定する。

【0047】

カレントホップカウント146が「0」に等しい場合には、図示のように戻りルーティングロジックRRLは終了する。またカレントホップカウント146が「0」に等しくない場合には、戻りルーティングロジックRRLはブロック379に進み、特定の中間ノード206a / 206bに対

応する宛先ポート値149a / 149bを識別する。その後、ブロック383で、図4に関して上述したように、データパケット136を受信した特定の間ノード206a / 206bのソースポートが宛先ポート値149a / 149b上に書き込まれる。その後、戻りルーティングロジックRRLは終了する。

【0048】

ここで図9を参照する。同図には、本発明の別の実施形態による宛先ロジックDLのフローチャートが示されている。該宛先ロジックDLは、宛先ノード209内に配置され、宛先ノード209が実際にデータパケット136の最終的な宛先であるか否かを判定する。宛先ロジックDLは、ブロック403から開始して、データパケット136が宛先ノード209により受信されたか否かを判定する。データパケット136が宛先ノード209により受信された場合には、宛先ロジックDLはブロック406に進み、カレントホップカウント146(図3)を検査する。カレントホップカウント146が「0」に等しい場合には、図示のように宛先ロジックDLは終了する。またカレントホップカウント146が「0」に等しくない場合には、宛先ロジックDLはブロック409に進む。該ブロック409で、カレントホップカウント146が「0」又は他の値に等しいと仮定すると、宛先ロジックDLは、宛先ノード209内の別の論理回路等に対し、該宛先ノード209がデータパケット136中のデータの意図するノードであることを通知する。その後、宛先ロジックDLは終了する。

【0049】

最後に図10を参照する。同図には、本発明の別の実施形態による再構成ロジックRRRのフローチャートが示されている。該再構成ロジックRRRは、宛先ノード209(図4)内に配置され、データパケット136(図3)が宛先ノード209へとルーティングされる際に中間ノード206a / 206b(図4)により記録された戻り経路に沿って応答パケット136a(図5)その他のデータパケットを返信するためのパラメータを生成する。再構成ロジックRRRは、ブロック433から開始して、データパケット136が宛先ノード209により受信されたか否かを判定する。データパケット136が宛先ノード209により受信された場合には、再構成ロジックRRRは、ブロック436に進み、カレントホップカウント146(図3)を検査する。カレントホップカウント146が「0」に等しい場合には、図示のように再構成ロジックRRRは終了する。またカレントホップカウント146が「0」に等しくない場合には、再構成ロジックRRRはブロック439に進む。

【0050】

ブロック439で、カレントホップカウント146は、合計ホップ値143(図3)に等しくリセットされ、その後、ブロック443で、宛先ポート値149a / 149bの再順序付けを行って、応答パケット136が元のソースノード203に戻るよう正しくルーティングされるようにする。その後、再構成ロジックRRRは終了する。

【0051】

上記説明に加えて、本発明のソースロジックSL、ルーティングロジックRL、宛先ロジックDL、戻りルーティングロジックRRL、及び再構成ロジックRRRは、ハードウェア、ソフトウェア、ファームウェア又はそれらの組合せによって実施することが可能である。好ましい実施形態では、ロジックSL,RL,DL,RRL,RRRは、最大速度のために及びレイテンシを最小化するために、ハードウェアで実現される。ソフトウェアで実施される場合には、ロジックSL,RL,DL,RRL,RRRは、メモリに格納することが可能であり、適当な命令実行システムにより実行される。また本実施形態に従ってハードウェアで実施される場合には、ロジックSL,RL,DL,RRL,RRRは、以下の技術(その全ては当業界で周知のものである)のうちの何れか又は組合せにより実施することが可能である。即ち、データ信号に対して論理機能を実施する論理ゲートを有する個別の論理回路、適当な論理ゲートを有する特定用途向け集積回路、プログラマブルゲートアレイ(PGA)、フィールドプログラマブルゲートアレイ(FPGA)等である。ハードウェアによる実施は、ネットワーク100(図1)のより高速な動作及びより低いレイテンシを提供するために好ましいものとなる。

【0052】

また、図6ないし図10のフローチャートは、ロジックSL,RL,DL,RRL,RRRの考え得る実施

形態のアーキテクチャ、機能性、及び動作を示している。ここで、各ブロックは、指定された論理的機能を実施するために1つ又は2つ以上の実行可能命令を含む論理回路、モジュール、セグメント、又はコードの一部を表している。また、代替的な実施形態によっては、各ブロックで示す機能が、図6ないし図10に示す順序以外の順序で発生することが可能である。例えば、図6ないし図10で連続的に示されている2つのブロックは実際には実質的に同時に実行することが可能であり、また、付随する機能に応じてそれらブロックを逆の順序で実行することが可能である。

【0053】

最後に、ロジックSL,RL,DL,RRL,RRRは、論理機能を実施するための実行可能な命令を順序付けして列挙したものから構成することが可能であり、これは、命令実行システム又は装置からの命令をフェッチして該命令を実行することができるコンピュータベースシステムやプロセッサ内蔵システムその他のシステムにより使用され又はかかるシステムに接続して使用されるあらゆるコンピュータ読み出し可能媒体で実施することが可能である。ここで、「コンピュータ読み出し可能媒体」とは、命令実行システム又は装置により又はそれに接続されて使用されるプログラムを含み、格納し、通信し、伝播し、又は転送することが可能なあらゆる手段とすることが可能なものである。例えば、コンピュータ読み出し可能媒体は、電子的、磁氣的、光学的、電磁氣的、赤外線、又は半導体的なシステム、装置、又は伝達媒体とすることが可能である（但しこれらに限定されるものではない）。コンピュータ読み出し可能媒体のより具体的な例には、以下のものが含まれる（但しこれは全てを網羅するものではない）。即ち、1つ又は2つ以上のワイヤを有する電氣的接続（電子的なもの）、ポータブルコンピュータディスク（磁氣的なもの）、ランダムアクセスメモリ（RAM）（磁氣的なもの）、リードオンリメモリ（ROM）（磁氣的なもの）、消去可能プログラマブルリードオンリメモリ（EPROM又はフラッシュメモリ）（磁氣的なもの）、光ファイバ（光学的なもの）、及びポータブルコンパクトディスクリードオンリメモリ（CDROM）（光学的なもの）である。なお、コンピュータ読み出し可能媒体は、プログラムがプリントされた用紙その他の適当な媒体とすることも可能である。この場合には、例えば用紙その他の媒体を光学走査することによりプログラムを電子的に捕捉し、その後、必要に応じて適当な方法でコンパイルし、翻訳し、又はその他の方法で処理した後に、コンピュータメモリに格納することが可能である。

【0054】

本発明の思想及び原理を実質的に逸脱することなく、本発明の上述の実施形態に対し多くの変形及び変更を行うことが可能である。かかる変更及び変形は全て、本開示内及び本発明の範囲内に含まれることが意図されている。

【0055】

以下においては、本発明の様々な構成要件の組み合わせからなる例示的な実施態様を示す。

1. ソースノード(203)、宛先ノード(209)、及び少なくとも1つの中間ノード(206)を含む、ネットワーク(100)内で相互接続された複数のノードと、
前記ソースノード(203)から前記少なくとも1つの中間ノード(206)を介して前記宛先ノード(209)へのデータ経路を識別する、前記ソースノード(203)内のソースロジック(SL)であって、前記データ経路が、前記ソースノード(203)から前記宛先ノード(209)へと送信されるべきデータパケット(136)に付与される少なくとも1つの宛先ポート値(149)とカレントホップカウンタ(146)とのシーケンスによって指定されるものである、前記ソースノード(203)内のソースロジック(SL)と、
前記データ経路に沿って前記データパケット(136)をルーティングする、前記少なくとも1つの中間ノード(206)内のルーティングロジック(RL)と、
前記データパケット(136)の最終的な宛先を検出する、前記宛先ノード(209)内の宛先ロジック(DL)と
を備えている、データ通信システム。

2. 前記データパケット(136)中に前記少なくとも1つの中間ノード(206)の少なくとも1

10

20

30

40

50

つのソースポート値を記録する、前記少なくとも1つの中間ノード(206)内の戻りルーティングロジック(RRL)を更に備えており、

前記データパケット(136)に合計ホップ値(143)が付与される、前項1に記載のシステム。

3. 前記ソースノード(203)内に配置されたルーティングテーブル(RT0~6, RTa~RTh)を更に備えており、該ルーティングテーブル(RT0~6, RTa~RTh)が、前記ソースノード(203)から前記宛先ノード(209)への少なくとも1つのデータ経路を含む、前項1に記載のシステム。

4. 前記ルーティングロジック(RL)が、前記カレントホップカウント(146)をデクリメントするロジックを更に備えている、前項1に記載のシステム。

5. 前記戻りルーティングロジック(RRL)が、前記データパケット(136)中の前記少なくとも1つの宛先ポート値(149)を前記少なくとも1つの中間ノード(206)のソースポート値に置換するロジックを更に備えている、前項2に記載のシステム。

6. ネットワーク(100)内でソースノード(203)から少なくとも1つの中間ノード(206)を介して宛先ノード(209)へと送信するデータパケット(136)を生成し、前記ソースノード(203)から前記少なくとも1つの中間ノード(206)を介して前記宛先ノード(209)へのデータ経路を識別し、該データ経路が、前記ソースノード(203)から前記宛先ノード(209)へと送信される前記データパケット(136)に付与される少なくとも1つの宛先ポート値(149)とカレントホップカウント(146)とのシーケンスによって指定されるものであり、

前記少なくとも1つの中間ノード(206)で前記データ経路に沿って前記データパケット(136)をルーティングし、

前記宛先ノード(209)で該データパケット(136)の到着を検出する、
という各ステップを含む、データ通信方法。

7. 前記データパケット(136)に合計ホップ値(143)を付与し、前記少なくとも1つの中間ノード(206)において該少なくとも1つの中間ノード(206)の少なくとも1つのソースポート値を前記データパケット(136)中に記録する、
という各ステップを更に含む、前項6に記載の方法。

8. 前記ソースノード(203)から前記少なくとも1つの中間ノード(206)を介して前記宛先ノード(209)へのデータ経路を識別する前記ステップが、前記ソースノード(203)内に配置されたルーティングテーブル(RT0~6, RTa~RTh)を検査するステップ更に含み、該ルーティングテーブル(RT0~6, RTa~RTh)が、前記ソースノード(203)から前記宛先ノード(209)への少なくとも1つのデータ経路を含む、前項6に記載の方法。

9. 前記少なくとも1つの中間ノード(206)で前記データ経路に沿って前記データパケット(136)をルーティングする前記ステップが、前記カレントホップカウント(146)をデクリメントするステップを更に含む、前項6に記載の方法。

10. 前記少なくとも1つの中間ノード(206)で該少なくとも1つの中間ノード(206)の少なくとも1つのソースポート値を前記データパケット(136)中に記録する前記ステップが、該データパケット(136)中の前記少なくとも1つの宛先ポート値(149)を前記少なくとも1つの中間ノード(206)の少なくとも1つのソースポート値に置換するステップを更に含む、前項7に記載の方法。

【図面の簡単な説明】

【図1】本発明の一実施形態によるマルチプロセッサ相互接続ネットワークの概要を示す説明図である。

【図2】図1のマルチプロセッサ相互接続ネットワークの様々な要素で用いられるデータ経路テーブルを示す説明図である。

【図3】図1のマルチプロセッサ相互接続ネットワークで用いられるデータパケットを示すブロック図である。

【図4】図1のマルチプロセッサ相互接続ネットワークで用いられるデータ経路の概要を示す説明図である。

【図5】図1のマルチプロセッサ相互接続ネットワークで用いられる応答パケットと比較

10

20

30

40

50

して図3のデータパケットを示す説明図である。

【図6】図1のマルチプロセッサ相互接続ネットワークで用いられるソースロジックを示すフローチャートである。

【図7】図1のマルチプロセッサ相互接続ネットワークで用いられるルーティングロジックを示すフローチャートである。

【図8】図1のマルチプロセッサ相互接続ネットワークで用いられる戻りルーティングロジックを示すフローチャートである。

【図9】図1のマルチプロセッサ相互接続ネットワークで用いられる宛先ロジックを示すフローチャートである。

【図10】図1のマルチプロセッサ相互接続ネットワークで用いられる戻り経路再構成ロジックを示すフローチャートである。

10

【符号の説明】

100 マルチプロセッサ相互接続ネットワーク

136 データパケット

143 合計ホップ値

146 カレントホップカウント

149 宛先ポート値

203 ソースノード

206 中間ノード

209 宛先ノード

DL 宛先ロジック

RL ルーティングロジック

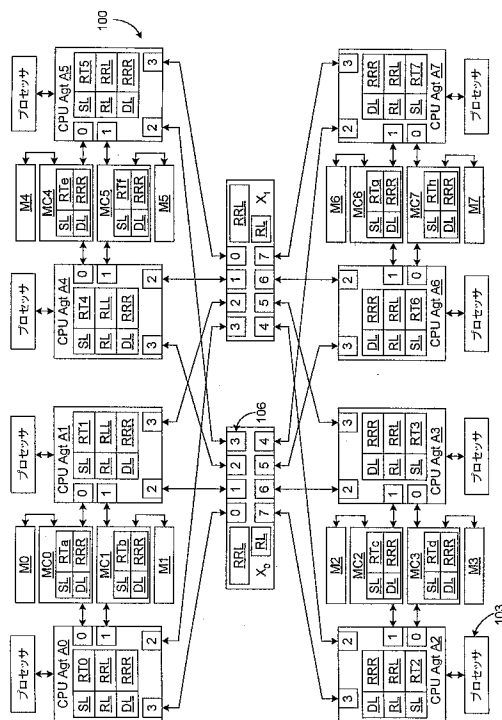
RRL 戻りルーティングロジック

SL ソースロジック

RT0 ~ 6, RTa ~ RTh ルーティングテーブル

20

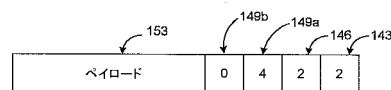
【図1】



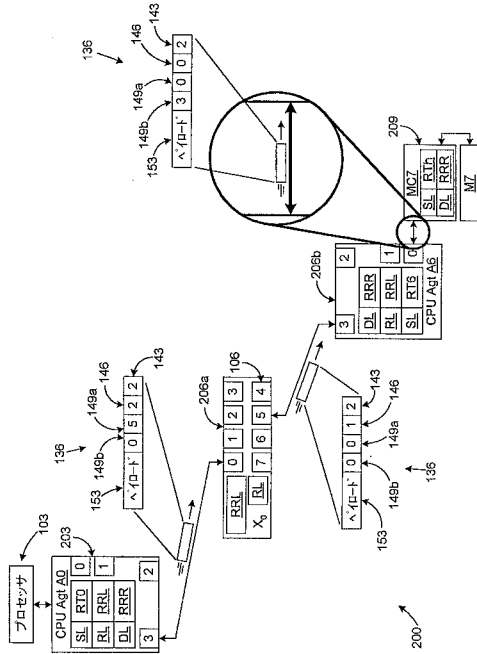
【図2】

ルーティングテーブル RT0						
宛先	クロスバ	宛先ポート1	宛先ポート2	...	宛先ポートN	合計ホップ値
MC7	X ₀	5	0	...		2
MC7	X ₀	4	0	...		2
MC7	X ₀	7	0	...		2
MC7	X ₁	6	0	...		2
...

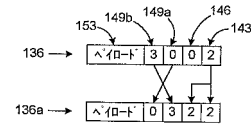
【図3】



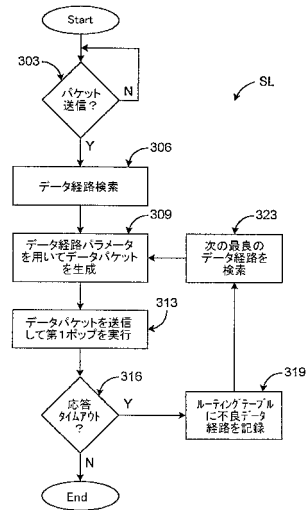
【図4】



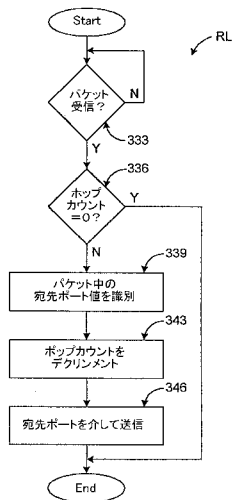
【図5】



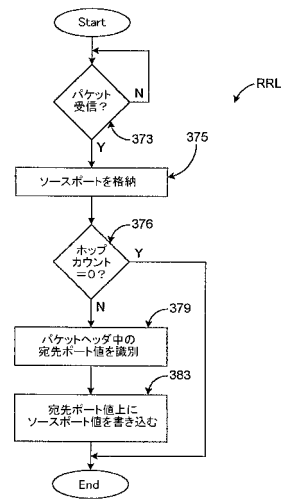
【図6】



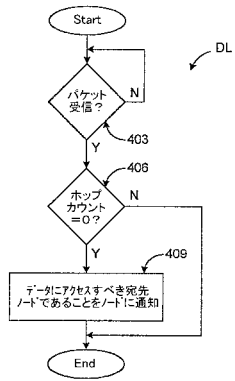
【図7】



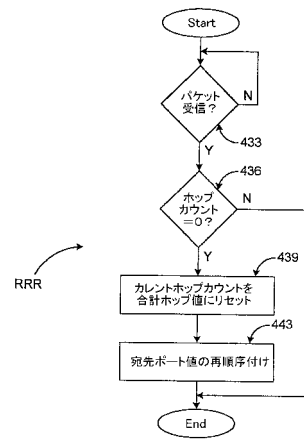
【図8】



【図 9】



【図 10】



フロントページの続き

審査官 保田 亨介

(56)参考文献 特開平 0 6 - 0 6 9 9 2 9 (J P , A)
国際公開第 9 9 / 0 0 0 9 3 8 (W O , A 1)

(58)調査した分野(Int.Cl. , D B 名)
H04L 12/00-12/66
G06F 15/163