(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第4287368号
(P4287368)

(45)発行日　平成21年7月1日(2009.7.1)　(24)登録日　平成21年4月3日(2009.4.3)

(51)Int.Cl.　F I
G06F 17/50 (2006.01)　G06F 17/50 654K
G06F 17/50 654D
G06F 17/50 654M

請求項の数 17　(全 17 頁)

(21)出願番号　特願2004-506220 (P2004-506220)
(86)(22)出願日　平成15年5月14日 (2003.5.14)
(65)公表番号　特表2005-526327 (P2005-526327A)
(43)公表日　平成17年9月2日 (2005.9.2)
(86)国際出願番号　PCT/US2003/015216
(87)国際公開番号　WO2003/098848
(87)国際公開日　平成15年11月27日 (2003.11.27)
審査請求日　平成18年5月11日 (2006.5.11)
(31)優先権主張番号　10/146,989
(32)優先日　平成14年5月15日 (2002.5.15)
(33)優先権主張国　米国 (US)

(73)特許権者　500107083
ソニックス・インコーポレーテッド
アメリカ合衆国・９４０４０・カリフォルニア州・マウンテン　ビュー・ウエスト　エル　カミノ　リアル・２４４０・スイート　６２０
(74)代理人　100064621
弁理士　山川　政樹
(74)代理人　100098394
弁理士　山川　茂樹
(72)発明者　メイヤー，マイケル・ジェイ
アメリカ合衆国・９４３０３・カリフォルニア州・パロアルト・エッジウッド　ドライブ・１５７５

最終頁に続く

(54)【発明の名称】分散型多重化バス相互接続を最適化する方法と装置

(57)【特許請求の範囲】

【請求項１】

分散型多重化バス相互接続に接続されたＩＰコアが前記多重化バス相互接続から信号を送信するかまたは受信するかに基づいて個々のシグナリング配線を除去することで前記分散型多重化バス相互接続内に存在するシグナリング配線の量を最適化する相互接続最適化方法であって、

前記多重化バス相互接続が該多重化バス相互接続を介して信号をルーティングする１以上のマルチプレクサを含み、

前記分散型多重化バス相互接続に接続した第１分散ノードは他のノードからの入力信号を組み合わせ、かつその入力ポートに存在するよりも少数の配線を有する次ノードへの出力ポートを有するコンバイニング・ロジックを含み、

該コンバイニング・ロジックおよび前記第１ノードと前記入力信号を供給する前記他のノードとの間のシグナリング配線を、該入力信号を供給する該他のノードが前記分散型多重化バス相互接続へ信号を送信しないときに除去する、

ことを特徴とする方法。

【請求項２】

前記コンバイニング・ロジックおよび前記第１ノードと前記入力信号を供給する前記他のノードとの間の前記シグナリング配線を、該入力信号を供給する該他のノードに接続されたサブノードもまた前記分散型多重化バス相互接続へ信号を送信しないときに限り除去する、請求項１に記載の方法。

【請求項３】
コンバイニング・ロジックをもたない前記ノードを前記最適化を行なった後に木構造へ併せ接続する、請求項１に記載の方法。
【請求項４】
前記不要な信号配線の除去は前記木の最下部から上へ実行する、請求項３に記載の方法。
【請求項５】
前記分散型多重化バス相互接続へ接続する第１分散ノードは他のノードからの入力信号を組み合わせて次のノードへの出力端を有するコンバイニング・ロジックを含み、該コンバイニング・ロジックおよび前記第１ノードと前記入力信号を供給する前記他のノードとの間の前記シグナリング配線を、該他のノードが送信する前記入力信号が一定の信号であるときに除去する、請求項１に記載の方法。
【請求項６】
前記分散型多重化バス相互接続へ接続する第１分散ノードは受信信号を第２ノードへ再送出するリピータ・ロジックを含み、該リピータ・ロジックおよび前記第１分散ノードと前記第２分散ノードとの間のシグナリング配線を、前記第２ノードと該第２ノードに接続したサブ・ノードが前記分散型多重化バス相互接続から信号を受信しないときに除去する、請求項１に記載の方法。
【請求項７】
前記最適化はさらにコンバイニング・ロジックとリピータ・ロジックと被接続ノードへのシグナリング配線とルート・ロジックとからなる群から選択される分散型多重化バス相互接続へ接続されたノードからの除去を含む、請求項１に記載の方法。
【請求項８】
前記最適化は前記分散型多重化バス相互接続を含むデバイスの製造前の所定時点で行なう、請求項１に記載の方法。
【請求項９】
一群の命令の実行時に請求項１に記載の方法を実施するプロセッサを含む、ことを特徴とする処理システム。
【請求項１０】
実行時に請求項１に記載の方法を実施する命令をそこに記憶した、ことを特徴とする機械読み取り可能な媒体。
【請求項１１】
バス相互接続配線およびロジックを最適化する方法であって、
（ａ）相互接続最適化プログラムを用いて可能性のある全てのバス信号に非最適化済みと標識付けするステップと、
（ｂ）前記全てのバス信号が最適化されたかどうかを判定し、そうである場合に停止するステップと、そうでない場合に、
（ｃ）第１の非最適化バス信号を選択するステップと、
（ｄ）２以上の入力信号を組み合わせない前記第１の非最適化バス信号の信号路内のノードごとにシグナリング配線を除去するとともにコンバイナ・ロジックをリピータ・ロジックでもって置換するステップと、
（ｅ）前記第１の非最適化バス信号を別のノードへ再送出する必要のない前記第１の非最適化バス信号の前記信号路内のノードごとに配線およびリピータ・ロジックを除去するステップと、
（ｆ）前記第１の非最適化バス信号を最適化済みと標識付けするステップと、
（ｇ）前記ステップ（ｂ）へ戻るステップと
を含む、
ことを特徴とする方法。
【請求項１２】
コンバイナ・ロジック木とリピータ・ロジック木の間の接続をもって第１ノードを特定

するステップで、前記第１ノード内の第１のコンバイナ・ロジックへの唯一の入力が第２ノードからの第２のコンバイナ・ロジックからの出力配線である前記ステップと、前記第２ノードからの前記出力配線を前記第２ノード内の第１のリピータ・ロジックの入力端へ接続するステップと、前記第１ノード内の前記第１のコンバイナ・ロジックを除去し、前記第１のコンバイナロジックの入力配線を前記第１ノード内の第２のリピータ・ロジックの入力端へ接続するステップとをさらに含む、請求項１１に記載の方法。

【請求項１３】

バスロジックを最適化する方法であって、

第１ノードが多重化バス相互接続に信号を送信する機能を有するかどうか判定するステップと、

前記第１ノードの送信信号路内に接続したサブ・ノードが前記多重化バス相互接続へ信号を送信する機能を有するかどうか判定するステップと、

前記第１ノードまたは前記サブ・ノードが前記多重化バス相互接続へ信号を送信する機能を有すると判定された場合に、コンバイニング・ロジックを付加して入力信号を前記第１ノードへ組み合わせるステップと、

前記送信信号路内の第２ノードが送信信号路を受信信号路へ接続するルート・ロジックを含むまで、前記送信信号路内のノードへコンバイニング・ロジックを付加するステップと

を含むことを特徴とする方法。

【請求項１４】

分散型多重化バス相互接続に接続されたＩＰコアが前記多重化バス相互接続から信号を送信するかまたは受信するかに基づいて個々のシグナリング配線を除去することで前記分散型多重化バス相互接続内に存在するシグナリング配線の量を最適化する手段を備える装置であって、

前記前記多重化バス相互接続が該多重化バス相互接続を介して信号をルーティングする１以上のマルチプレクサと、

前記分散型多重化バス相互接続に接続した第１分散ノードは他のノードからの入力信号を組み合わせ、かつその入力ポートに存在するよりも少数の配線を有する次ノードへの出力ポートを有するコンバイニング・ロジックと

を含み、

該コンバイニング・ロジックおよび前記第１ノードと前記入力信号を供給する前記他のノードとの間のシグナリング配線を、該入力信号を供給する該他のノードが前記分散型多重化バス相互接続へ信号を送信しないときに除去する、

ことを特徴とする装置。

【請求項１５】

前記信号結線量の最適化は前記装置の製造前に行なう、請求項１４に記載の装置。

【請求項１６】

前記最適化手段は前記多重化バス相互接続を含むデバイスの前記製造前の所定時点に最適化する、請求項１４に記載の装置。

【請求項１７】

請求項１４に記載の装置を表わす情報をそこに記憶させた、ことを特徴とする機械可読媒体。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は相互接続に関する。具体的に言うと、本発明は、分散型多重化バス相互接続を最適化する方法と装置に関する。

【背景技術】

【０００２】

情報を通信する必要があるコンピュータ・ネットワーク、インターネットワーキング、

通信、集積回路などでは、しばしば、情報の転送を容易にするために確立される相互接続がある。１つの手法が、情報を転送する専用の通信「信号線」またはリンクを使用することである。バスは、通常、２つのデバイスが通信する必要がある時に使用される。バスを実現する従来の形は、あるデバイスがバスを駆動し、他のドライバがディスエーブルされるトライステート・バス・ドライバである。もう１つの手法が、各デバイスに異なる組のワイヤを使用させ、マルチプレクサを使用して、イネーブルされるデバイスのワイヤの組を選択することである。

【０００３】

しかし、バスを多重化する際に、バスのすべての機能を必要としない可能性がある通信ポイントがある場合がある。したがって、バス全体をこれらの実体まで延長することは、空間、電力などのリソースの浪費になる。これが問題になる可能性がある。

【発明を実施するための最良の形態】

【０００４】

本発明を、添付図面の図で、制限ではなく例として示す。図面では、類似する符号が類似する要素を示す。

【０００５】

分散型多重化バス相互接続を最適化する方法と装置を説明する。

【０００６】

この文書で使用される用語ＩＰは、Ｉｎｔｅｌｌｅｃｔｕａｌ　Ｐｒｏｐｅｒｔｙを示す。用語ＩＰは、単独で使用することができ、あるいは、機能性を有する設計を示すために、コアなどの他の用語と共に使用することができる。たとえば、ＩＰコアまたは略してＩＰは、回路、バス、通信リンク、マイクロプロセッサなどからなるものとすることができる。さらに、ＩＰをさまざまな形で実施することができ、集積回路などで製造することができる。

【０００７】

バスは、伝統的に、図１Ａまたは図１Ｂに示された形で接続された一列のブロックまたはコネクタと考えられてきた。この手法は、単純であり、実装が簡単である。集積回路（ＩＣ）設計では、複数のドライバとバスを有するワイヤにリピータを追加することの難しさのゆえに、３ステート（トライステート）（図１Ａ）が望ましくない場合があり、バスは、一般にマルチプレクサを介して実装される。マルチプレクサの実施形態は、各トランスミッタから各レシーバへのワイヤを必要とし、これによって、配線輻輳の可能性が生じる場合がある。もう１つの手法は、分散マルチプレクサとすることであり、この場合に、各ブロックのロジックによって、他のブロックからの信号がマージされ、おそらくは次デバイスへのより少数のワイヤがマージされる。伝統的に、バス実施形態では、バスの信号が、レシーバ、トランスミッタ、またはトランシーバであると仮定されてきた。

【０００８】

分散型多重化バス・トポロジの記述は、さまざまな形で行うことができる。単純な線形トポロジについて、リストによって、ブロックが接続される順番を表すことができる。しかし、複数のノードがあるノードでマージされるより複雑なトポロジについて、木が、トポロジを記述するよりよい形である。ユーザは、下記のように、解析木風の構文でバスのトポロジを記述することができる。

```
多重化木　＜部分木＞
＜部分木＞　：：＝　＜ルート＞　＜枝＞　＜枝＞
＜ルート＞　：：＝　＜インスタンス名＞
＜枝＞　：：＝　｛＜部分木＞｝
　　｜　＜インスタンス名＞
　　｜　０（接続なしを示す標識）
```

【０００９】

例の木を、

```
多重化木　Ａ｛ＢＣ｛ＤＥ｝｝｛ＦＧＨ｝
```

とすることができる。上のこの行によって、図２Ａのようなトポロジを有するブロックの間の接続性が記述される。この設計が、ＩＣで実装される時に、ブロックは、図２Ｂのブロックのように見える可能性がある。木の各ノードは、ＩＰコア、任意選択として親ノード、任意選択として子ノードの組と通信することができる。各ノードは、図２Ｃに示されているように、ＩＰコアとエージェントからなる。分散マルチプレクサ（および潜在的に他の機能）を、エージェント内で実現することができる。多重化バスが接続される方法に応じて、エージェントを変更する必要がある場合があるが、コアは影響を受けない可能性がある。これによって、バスが変更された時に変更する必要なしでコアを再利用できるようになる可能性がある。図３Ａに、この例の各エージェントのレシーバとトランシーバを示す。この例のバスの信号について、エージェントＡ、Ｃ、Ｇは、信号を受け取る必要があり、エージェントＢ、Ｃ、Ｄ、Ｅは信号を送ることができる。エージェントに、これらのレシーバとトランスミッタのインターフェースと、実際のＩＰコアとの間の他のプロトコル変換を実行する追加ロジックを設けることができる。このトランスミッタとレシーバの構成、さらには上で説明したｍｕｘ＿ｔｒｅｅ仕様が与えられると、エージェントが、図３Ｂに示されているように一緒に配線される。ｍｕｘ　ｔｒｅｅルートの出力が、ｍｕｘ　ｔｒｅｅルート・リピータの入力に接続されていることに留意されたい。コンバイナ機能（図７Ａ）は、入力として、コアまたは他のエージェントのロジックからのデータおよびイネーブルならびに部分木のコンバイナ出力をとる。図７Ｂに、コンバイナの「ａｎｄ－ｏｒ」実施形態を示す。

【００１０】

木構造は、単純な線形配線より少ない配線および／または短いエンドツーエンド・パスとなる。構造をユーザが指定することができ、かつ／またはプログラムが最小スパニング木（minimal spanning tree）を見つけることができる。同一のトポロジを使用して信号をルーティングすることによって、予測可能な配線遅延を作成し、かつ／または輻輳を減らすことができる。２組のワイヤが、トポロジ的に隣接する２つのノードの間で使用され、第１の組は、結果を組み合わせるのに使用することができ、第２の組は、結果をすべてのノードに分配するのに使用することができる。

【００１１】

分散型多重化バスの信号配線の最適化は、ノードを調べることによって行うことができる。たとえば、ある信号を生成（送出）しないノードがあり、ある信号を使用し（受け取ら）ないノードがある場合がある。すべての信号をすべてのノードにルーティングするには、より多くの配線が必要になり、信号のエンドツーエンド・パス長が増える可能性がある。信号を駆動しないノードからのコンバイニング・ワイヤおよび／または結果を使用しないノードへの分配ワイヤを除去することによって、より少数のドライバおよび／またはより少ないワイヤを必要とすることによる面積の削減、クリティカル・パスの短縮によるチップ・タイミングの潜在的な改善、および／または同一タイミングを達成するのにより小さいドライバを使用することによる電力削減を可能にすることができる。

【００１２】

図６Ａに、バス信号配線の最適化の高水準アルゴリズムの本発明の一実施形態を示す。シーケンスを特定の順序で実行する必要はないが、図示の順序によって実装が簡単になることに留意されたい。各信号を分析し、不要なトランスミッタやレシーバを除去する。バスの信号は、各信号がコア機能とバス・プロトコルに基づく異なるトランスミッタとレシーバのトポロジを有するので、個別に最適化される。これらの不要なワイヤを除去するアルゴリズムを、下に示す。ルートとｂｕｓ＿ｓｉｇｎａｌｓを有する木に対して：

```
foreach signal in bus_signals{
      optimaize_combiners(root,signal);
      optimaize_repeaters(root,signal);
      optimaize_root(root,signal);
}
```

【００１３】

不要なコンバイナの除去によって、チップのトップレベルでブロックを接続するのに使用される配線の量を減らすことができ、かつ／または、タイミング、面積、電力などについてよりよく最適化できるように一部の信号のパスを短縮することができる。コンバイナの最適化は、本発明の一実施形態で、特定の信号に関する不要なコンバイナのボトム・アップ除去によって行うことができる。エージェントに接続されたコアと他のエージェントのロジックが、トランスミッタを有しておらず、部分木の子のどれもがコンバイナを有しない場合に、エージェント内でコンバイナは不要である。図６Ｂに、流れ図で、コンバイナの挿入の最適化の本発明の一実施形態を示す。コンバイナの最適化は、図６Ｂに示されている加法的および／または下で示す再起アルゴリズムに示されているように減算的のいずれかとすることができる。

```
procedure optimize_combiners(sub_tree,signal){
      has_combiner=node_needs_combiner(sub_tree.node,signal)
      foreach child in sub_tree.children
            if(optimize_combiners(child,signal){
                  has_combiner=true
            }
      if(!has_combiner)
            remove_combiner(sub_tree.node,signal)
      return has_combiner
}
```

【００１４】

図３Ｃに、コンバイナ最適化の効果を示す。コンバイナが、エージェントＦ、Ｇ、Ｈから除去されている。コンバインが１つの入力を有する場合に、「ｏｒ」機能を、入力が部分木から来る場合にはバッファによって置換でき、入力がトランスミッタから来る場合にはコンバイナのデータとイネーブルの「ａｎｄ」に置換することができる。

【００１５】

未使用のリピータは、特定の信号に関して不要なリピータを除去することによって最適化することができる。これは、タイミング、面積、電力などの利益を有する。エージェントに接続されたコアとエージェント内の他のロジックが、リピータを有しておらず、部分木のどの子にもリピータがない場合に、エージェントにリピータは不要である。図６Ｃに、本発明の一実施形態の、ｍｕｘ　ｔｒｅｅへのリピータの追加の流れ図を示す。リピータ最適化処理は、図６Ｃに示されているように加法的とすることができ、かつ／または下のアルゴリズムに示されているように減算的にすることができる。

```
procedure optimize_repeaters(sub_tree,signal){
      has_repeater=node_needs_repeater(sub_tree,signal)
      foreach child in sub_tree.children
            if(optimize_repeaters(child,signal){
                  has_repeater=true
            }
      if(!has_repeater)
            remove_repeater(sub_tree.node,signal)
      return has_repeater
}
```

【００１６】

図３Ｄに、リピータ最適化の効果を示す。リピータが、エージェントＤ、Ｅ、Ｈから除去されている。

【００１７】

すべてのトランスミッタが１つの部分木の中にある場合に、木全体のルートからその部分木のルートへの結果を返すのに使用されるワイヤは、このノードが部分木およびルートに直接に結果を駆動できるので、除去することができる。図６Ｄに、信号に関するルート

の最適化の流れ図を示し、例の擬似コードを下に示す。

```
procedure optimize_root(sub_tree,signal){
    if node_needs_transmitter(sub_tree,signal)
          connect_combiner_to_repeater(sub_tree.root);
          return;
    new_root=NULL
    foreach child in sub_tree.childrem{
        if(node_needs_transmitter(child,signal)){
            if(new_root!=NULL){
                return
            }
            new_root=child;
        }
    }
   if(new_root!=NULL)
        optimize_root(child,signal)
 }
```

【００１８】

図３Ｅに、ルートの最適化の効果を示す。この例では、ＡとＢを接続するリピータ・ワイヤが除去され、エージェントＢがこの信号のルートとして働く。

【００１９】

分散型多重化バスのタイミングの最適化は、電力、面積などの削減をもたらす。論理合成は、式を、最適化された論理ゲートに変換するプログラムである。論理式のほかに、合成は制約も受け入れることができる。タイミング制約によって、入力が使用可能である時と出力が必要な時を記述することができ、論理合成は、これらの制約を最もよく満たすように論理ゲートの最適化を試みる。以前の手法では、タイミング制約で、制約を生成する時にバス・トポロジ内の位置が考慮されないことがあった。これは、過大な制約につながる可能性があり、その結果、面積および／または電力に関する準最適設計につながる可能性がある。本発明の一実施形態では、生成を制約する時に、バス・トポロジ内の各エージェントの位置を考慮する。制約は、信号配線を最適化した後に生成される。従来の特性記述に基づいて、全体的なバス遅延を構成するタイミングの各構成要素について推定を行う。構成要素には、スケーラブルな構成要素と固定構成要素が含まれる。スケーラブル構成要素には、レジスタ－バス出力遅延（図５－Ａ）、バス入力－バス出力遅延（図５－ＣおよびＦ）、バス入力－レジスタ遅延（図５－Ｅ、Ｈ）を含めることができる。固定構成要素は、エージェントの位置に基づく、出力ポートと入力ポートの間の配線に起因する遅延（図５－Ｂ、Ｄ、Ｇ）とすることができる。これらの構成要素を使用して、バスを構成するパスのすべてに対して遅延計算を行うことができる。図４Ａに示された多重化バス・トポロジに対して、図４－Ｂの多重化バスが使用される。上の最適化を適用することによって、ＣからＡへのＢ２を介する論理的に同等のパスよりよいタイミングについて最適化されたＤからＡへのＢ１を介するパスがもたらされる。さらに、ノードＣ内のドライバのサイズは、パスがＥから始まるパスよりクリティカルでないので、減らされる。

【００２０】

図８に、タイミング制約の最適化に関する本発明の一実施形態を示す。遅延計算は、出力ポートからすべての可能なパスをサーチし、パスを構成するタイミング要素を加算することによって行われる。各独自のパスのリストが、その総合的な遅延と共に記憶される。パスの構成要素のそれぞれが、スケーリング処理用のリストに保管される。

【００２１】

次の手順は、バスが必要とするタイミングを満たすための各パスのスケーリングである。各スケーラブル・タイミング構成要素を減らすスケール・ファクタを計算することによって、タイミングを超えるパスをタイミングを満たすようにスケール・ダウンする。必要

より短いタイミングを有するパスは、各スケーラブル・タイミング構成要素を増やすスケール・ファクタを計算することによって、タイミングを満たすようにスケール・アップされる。スケーリングは、最長遅延のパスから始め、各構成要素にパス固有のスケーリング・ファクタを適用し、各構成要素をスケーリング済みとしてマークすることによって進行する。この処理は、上で生成されたパスごとに継続される。順序付けとマークは、より長いパスが必要とするタイミング構成要素の遅延を増やさないようにするのに重要である。より長い時間を可能にすること（タイミング構成要素のスケール・アップ）によって、論理合成で、より小さく、より少ない電力を使用するより低速のセルを選択できるようになる。最終的な結果は、総合的な設計に関するより小さい面積要件および／またはより低い電力要件とすることができる。

【００２２】

図５に、この処理の効果を示す。この構成では、Ａ、Ｂ、Ｃ、Ｄ、Ｆ、Ｇ、Ｈを介するパスのタイミングがより長いパスなので、パスＡ、Ｂ、Ｃ、Ｄ、Ｅの制約が、よりゆるい。その結果、論理合成では、Ｄ、ＥからよりもＧ、Ｈからのタイミングが最適化される。

【００２３】

同一信号の複数同時ドライバは、ある信号（エラー信号または割込み信号など）について正しいが、他の信号（アドレスなど）について不正である場合がある。シミュレータは、衝突する値（あるドライバが１を駆動し、別のドライバが０を駆動する）を駆動している時に複数同時トライステート・ドライバを検出し、設計エラーを検出するのを助けるためにＸを生成することができる。分散型多重化バスで複数ドライバを検出するのは、困難である。というのは、コンバイニング機能（たとえば「ａｎｄ－ｏｒ」）によって、シミュレータがこの設計エラーを把握できなくなり、実施形態の分散的な性質によって、単一チェッカの追加が困難になるからである。もう１つの手法は、分散されたマルチプレクサの各コンバイナ機能に、検査を分散することである。

【００２４】

Ｎビット・コンバイナｖｅｒｉｌｏｇ論理式の「ｏｒ」実施形態を下に示す。

```
Output=(core_input & {N{core_enable}})|left_input|right_input|
```

【００２５】

衝突を検出するチェッカは、コアがイネーブルされ（ｃｏｒｅ＿ｉｎｐｕｔが非０）、左または右の入力が０でない場合を探すことができる。さらに、左入力と右入力の両方が非０の場合に、複数のドライバがある。表１（図７Ｃ）に、バス衝突を検出する真理値表を示す。これのｖｅｒｉｌｏｇ論理式は次の通りである。

```
error=(Core_enable &&|(left_input|right_input))||(|left_input|| |right_input)
```

【００２６】

これによって、複数のコアが０を駆動している場合が把握されない可能性があるが、複数ビット信号に関するこれの可能性は、比較的低く、したがって、この検査は、コア・イネーブル信号のすべての、より複雑な検査とほとんど同程度によい。これを使用して、下に示すように、シミュレーションを停止し、設計エラーを報告することができる。

```
if(error)begin
    $display("multiple drivers");
    $finish;
end
```

【００２７】

したがって、開示されたものは、分散型多重化バス相互接続を最適化する方法と装置である。

【００２８】

図９に、説明した技法を実施できるネットワーク環境９００を示す。ネットワーク環境９００は、Ｓ個のサーバ９０４－１から９０４－ＳとＣ個のクライアント９０８－１から９０８－Ｃを接続するネットワーク９０２を有する。図からわかるように、Ｓ個のサーバ９０４－１から９０４－ＳとＣ個のクライアント９０８－１から９０８－Ｃの形の複数の

システムが、ネットワーク１９０２を介して互いに接続され、このネットワークは、たとえば、オンチップ通信ネットワークとすることができる。代替案では、ネットワーク９０２を、チップ間通信、光ネットワーク、インターネット、ローカル・エリア・ネットワーク（ＬＡＮ）、広域ネットワーク（ＷＡＮ）、衛星リンク、ファイバ・ネットワーク、ケーブル・ネットワーク、あるいはこれらおよび／または他の組合せとするか、その１つまたは複数を含めることができることに留意されたい。サーバは、たとえば、チップ上のマスタ・デバイス；メモリ；マイクロプロセッサ、通信インターフェースなどのＩＰコア；ディスク・ストレージ・システム；および／またはコンピューティング・リソースを表すことができる。同様に、クライアントは、コンピューティング機能、ストレージ機能、表示機能を有することができる。本明細書に記載の方法と装置は、ローカルであれリモートであれ、ＬＡＮ、ＷＡＮ、システム・バス、オンチップ・バスなど、本質的にすべてのタイプの通信手段または通信デバイスに適用することができる。さらに、用語クライアントとサーバの使用が、どちらが通信を開始し（クライアント）、どちらが応答する（サーバ）かを指定する際の明瞭さのためのものであることを諒解されたい。明示的に示されない限り、階層は暗示されない。両方の機能を、単一の通信するデバイス内に設けることができ、その場合に、クライアント－サーバとサーバ－クライアント関係を、ピアツーピアとみなすことができる。したがって、サーバ９０８－１および９０４－Ｓなどの２つのデバイスの両方が、通信を開始し、通信に応答する場合に、その通信を、ピアツーピアとみなすことができる。同様に、９０４－１と９０４－Ｓ、９０８－１と９０８－Ｃの間の通信は、そのような通信デバイスが通信を開始でき、応答できる場合に、ピア・ツー・ピアとみなすことができる。

【００２９】

図１０に、図９に示されたクライアントおよび／またはサーバのいずれかを表すことができる、コンピュータ・システム１０００をブロック図形式で示す。このブロック図は、高水準の概念表現であり、さまざまな形でさまざまなアーキテクチャによって実装することができる。バス・システム１００２によって、中央処理装置（ＣＰＵ）１００４、読取専用メモリ（ＲＯＭ）１００６、ランダム・アクセス・メモリ（ＲＡＭ）１００８、ストレージ１０１０、ディスプレイ１０２０、オーディオ１０２２、キーボード１０２４、ポインタ１０２６、各種入出力（Ｉ／Ｏ）デバイス１０２８、通信１０３０が相互接続されている。バス・システム１００２は、たとえば、オンチップ・バス、システム・バス、Ｐｅｒｉｐｈｅｒａｌ Ｃｏｍｐｏｎｅｎｔ Ｉｎｔｅｒｃｏｎｎｅｃｔ（ＰＣＩ）、Ａｄｖａｎｃｅｄ Ｇｒａｐｈｉｃｓ Ｐｏｒｔ（ＡＧＰ）、Ｓｍａｌｌ Ｃｏｍｐｕｔｅｒ Ｓｙｓｔｅｍ Ｉｎｔｅｒｆａｃｅ（ＳＣＳＩ）、Ｉｎｓｔｉｔｕｔｅ ｏｆ Ｅｌｅｃｔｒｉｃａｌ ａｎｄ Ｅｌｅｃｔｒｏｎｉｃｓ Ｅｎｇｉｎｅｅｒｓ（ＩＥＥＥ）標準規格番号１３９４（ＦｉｒｅＷｉｒｅ）、Ｕｎｉｖｅｒｓａｌ Ｓｅｒｉａｌ Ｂｕｓ（ＵＳＢ）などの１つまたは複数とすることができる。ＣＰＵ１００４は、単一、複数、または分散コンピューティング・リソースとすることができる。ストレージ１０１０は、コンパクト・ディスク（ＣＤ）、ディジタル多用途ディスク（ＤＶＤ）、ハード・ディスク（ＨＤ）、光ディスク、テープ、フラッシュ、メモリ・スティック、ビデオ・レコーダなどとすることができる。ディスプレイ１０２０は、陰極線管（ＣＲＴ）、液晶ディスプレイ（ＬＣＤ）、プロジェクション・システム、テレビジョン（ＴＶ）などとすることができる。システムの実際の実施形態に応じて、システムに、このブロック図の構成要素の一部、すべて、より多く、または再配置を含めることができることに留意されたい。たとえば、集積回路上のオンチップ通信システムでは、ディスプレイ１０２０、キーボード１０２４、ポインタ１０２６をなくすことができる。もう１つの例が、シン・クライアントであり、これは、たとえば従来のキーボードがない無線ハンド・ヘルド・デバイスからなるものとすることができる。したがって、図１０のシステムに対する多数の変形形態が可能である。

【００３０】

本発明の議論と理解において、技法と手法を説明するために、当業者によってさまざま

な用語が使用されることを理解されたい。さらに、説明で、説明のために、本発明の完全な理解を提供するために、多数の具体的な詳細を示す。しかし、本発明を、これらの具体的な詳細なしで実践できることは、当業者に明白である。いくつかの場合に、周知の構造やデバイスが、本発明を不明瞭にしないために、詳細にではなくブロック図形式で示される。これらの実施形態は、当業者が本発明を実践できるようにするのに十分に詳細に説明され、他の実施形態を使用できることと、本発明の範囲から逸脱せずに論理的変更、機械的変更、電気的変更、他の変更を行えることを理解されたい。

【００３１】

この説明のいくつかの部分は、たとえばコンピュータ・メモリ内のデータ・ビットに対する動作のアルゴリズム表現または記号表現に関して提示される場合がある。これらのアルゴリズム的な説明と表現は、他の当業者に成果の実質を最も効果的に伝えるために当業者によって使用される手段である。アルゴリズムは、本明細書で、一般的に、所望の結果につながる動作の自己完結型のシーケンスと考えられる。作用は、物理的な量の物理的操作を必要とする作用である。通常、必ずではないが、これらの量は、保管され、転送され、組み合わされ、比較され、他の形で操作されることが可能な電気信号、磁気信号、または光信号の形をとる。時々、主に一般的な使用のために、これらの信号を、ビット、値、要素、記号、文字、項、番号、または類似物と称することが便利であることがわかっている。

【００３２】

しかし、これらおよび類似する用語のすべてが、適当な物理的量に関連し、これらの量に適用される単に便利なラベルであることに留意されたい。議論から明白として他の形で特に述べられないかぎり、この説明全体を通じて、「処理」または「コンピューティング」または「計算」または「判定」または「通信」または「表示」または類似物などの用語を使用する議論が、電子デバイスまたはコンピュータ・システムのレジスタやメモリ内の物理的（電子的）量として表されるデータを操作し、電子デバイスおよび／またはコンピュータ・システムのメモリまたはレジスタあるいはそのような情報の保管デバイス、伝送デバイス、または表示デバイス内の物理的量として同様に表される他のデータに変換するコンピュータ・システムまたは電子デバイスの作用または処理を指すことができることを諒解されたい。

【００３３】

本発明を、本明細書の動作を実行する装置によって実現することができる。この装置は、必要な目的のために特に構成することができ、あるいは、コンピュータに保管されたコンピュータ・プログラムによって選択的にアクティブ化または再構成される汎用コンピュータを含めることができる。そのようなコンピュータ・プログラムは、フロッピ・ディスク、ハード・ディスク、光ディスク、コンパクト・ディスク読取専用メモリ（ＣＤ－ＲＯＭ）、ディジタル多用途ディスク（ＤＶＤ）、光磁気ディスクを含むすべてのタイプのディスク、読取専用メモリ（ＲＯＭ）、ランダム・アクセス・メモリ（ＲＡＭ）、電気的プログラム可能読取専用メモリ（ＥＰＲＯＭ）、電気的消去可能プログラム可能読取専用メモリ（ＥＥＰＲＯＭ）、フラッシュ・メモリ、磁気カード、光カード、あるいはコンピュータにローカルまたはコンピュータからリモートのいずれかの電子命令を保管するのに適するすべてのタイプの媒体を含むがこれに制限されないコンピュータ可読記憶媒体に保管することができる。

【００３４】

本明細書で提示されたアルゴリズムおよびディスプレイは、特定のコンピュータまたは他の装置に固有に関連するものではない。さまざまな汎用システムを、本明細書の教示に従ってプログラムと共に使用することができ、あるいは、より特殊化された装置を構成して必要な方法を実行することが便利であるとわかる場合がある。たとえば、本発明による方法のいずれかを、ハードワイヤード回路内で、汎用プロセッサをプログラムすることによって、またはハードウェアやソフトウェアの組合せによって、実現することができる。当業者は、ハンドヘルド・デバイス、マルチプロセッサ・システム、マイクロプロセッサ

ベースまたはプログラム可能な民生用電子機器、ディジタル信号プロセッサ（ＤＳＰ）デバイス、セットトップ・ボックス、ネットワークＰＣ、ミニコンピュータ、メインフレーム・コンピュータ、類似物を含む、本明細書に記載されたもの以外のコンピュータ・システム構成を用いて本発明を実践できることを、即座に諒解するであろう。本発明は、通信ネットワークを介してリンクされたリモート処理デバイスによってタスクが実行される分散コンピューティング環境で時も実践することができる。この通信ネットワークは、サイズによって制限されず、たとえば、オンチップ通信からインターネットなどのＷＡＮまでの範囲とすることができる。

【００３５】

本発明の方法を、コンピュータ・ソフトウェアを使用して実施することができる。認識可能な標準規格に従うプログラミング言語で記述される場合に、この方法を実施するように設計された命令のシーケンスを、さまざまなプラットフォームでの実行とさまざまなオペレーティング・システムへのインターフェースのためにコンパイルすることができる。さまざまなプログラミング言語を使用して、本明細書に記載の発明の教示を実施できることを諒解されたい。さらに、当技術分野で、動作を行うか結果を引き起こすものとして、ある形または別の形（たとえば、プログラム、プロシージャ、アプリケーション、ドライバなど）でソフトウェアについて話すことが一般的である。そのような表現は、コンピュータによるソフトウェアの実行が、コンピュータのプロセッサに動作を実行させるか結果を生じさせることを話す単に速記の形である。

【００３６】

通信、プロトコル、アプリケーション、実施形態、機構など、さまざまな用語および技法が、当業者によって使用されることを理解されたい。そのような技法の１つが、アルゴリズムまたは数学的表現に関する技法の実施形態の説明である。すなわち、この技法を、たとえばコンピュータ上の実行コードとして実施することができるが、その技法の表現を、式、アルゴリズム、または数学的表現としてより適切かつ簡潔に伝え、通信することができる。したがって、当業者は、ハードウェアおよび／またはソフトウェアでの実施形態が２つの入力（ＡおよびＢ）をとり、合計出力（Ｃ）を作る加算機能としてＡ＋Ｂ＝Ｃを示すブロックを理解するであろう。したがって、説明としての式、アルゴリズム、または数学的表現の使用は、少なくともハードウェアおよび／またはソフトウェアでの物理的実施形態（本発明の教示を実施形態として実践されると同時に実施することができるコンピュータ・システムなど）を有するものとして理解されなければならない。

【００３７】

機械可読媒体は、機械（たとえばコンピュータ）によって可読の形で情報を保管または伝送するすべての機構を含むと理解される。たとえば、機械可読媒体に、読取専用メモリ（ＲＯＭ）；ランダム・アクセス・メモリ（ＲＡＭ）；磁気ディスク記憶媒体；光記憶媒体；フラッシュ・メモリ・デバイス；電子、光、音響、または他の形の伝搬される信号（たとえば、搬送波、赤外線信号、ディジタル信号など）などを含めることができる。

【００３８】

したがって、分散型多重化バス相互接続を最適化する方法と装置を説明した。

【図面の簡単な説明】

【００３９】

【図１】従来のバスを示す図である。

【図２】トポロジを示す、本発明の実施形態を示す図である。

【図３Ａ】特定のバス信号のレシーバおよびトランスミッタを示す本発明の一実施形態を示す図である。

【図３Ｂ】最適化されていない接続を示す本発明の一実施形態を示す図である。

【図３Ｃ】コンバイナ最適化の後のトポロジを示す本発明の一実施形態を示す図である。

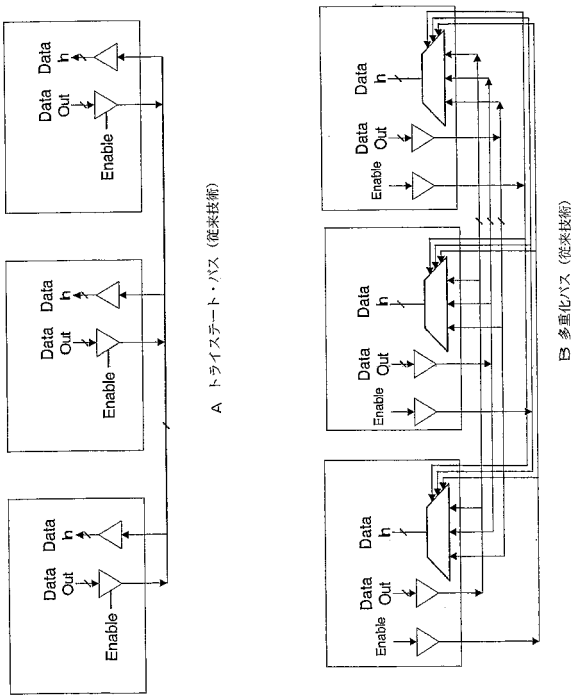
【図３Ｄ】リピータ最適化の後のトポロジを示す本発明の一実施形態を示す図である。

【図３Ｅ】ルート最適化の後のトポロジを示す本発明の一実施形態を示す図である。

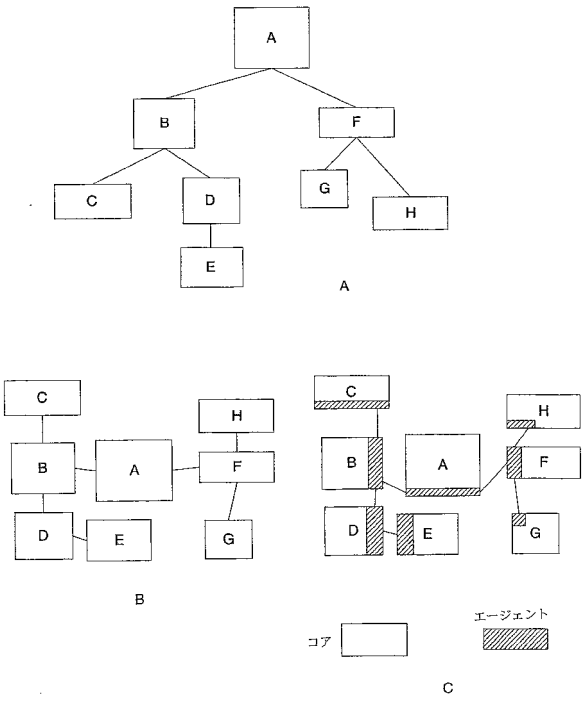
【図４】相互接続およびロジックの詳細を示す本発明の一実施形態を示す図である。

【図５】タイミング・パスを示す本発明の一実施形態を示す図である。
【図６Ａ】バス信号配線の最適化の流れ図を示す本発明の一実施形態を示す図である。
【図６Ｂ】コンバイナ最適化の流れ図を示す本発明の一実施形態を示す図である。
【図６Ｃ】バス・リピータ最適化の流れ図を示す本発明の一実施形態を示す図である。
【図６Ｄ】ルート最適化の流れ図を示す本発明の一実施形態を示す図である。
【図７】コンバイナ・ブロックを示す本発明の一実施形態を示す図（Ａ）、コンバイナ・ブロックを詳細に示す本発明の一実施形態を示す図（Ｂ）、バス衝突の検査に関する真理値表１を示す本発明の一実施形態を示す図（Ｃ）である。
【図８】タイミング制約の最適化を示す本発明の一実施形態を示す図である。
【図９】本発明の方法と装置を実施できるネットワーク環境を示す図である。
【図１０】コンピュータ・システムを示すブロック図である。

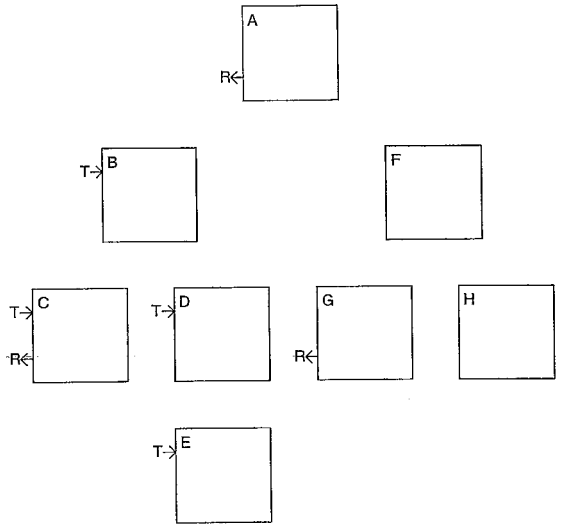
【図１】



【図２】



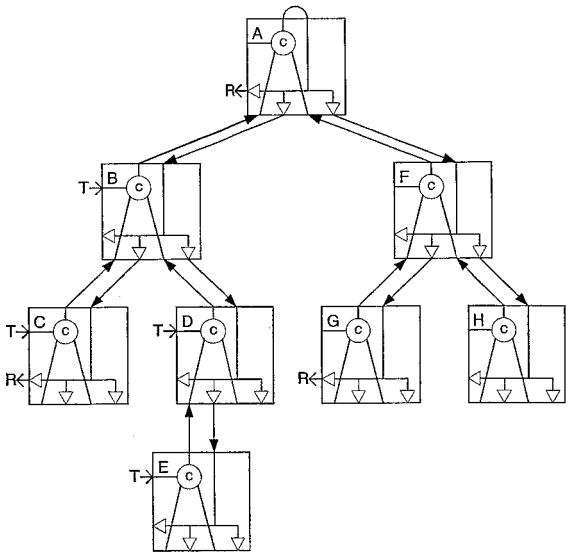
【図３Ａ】



前提：
エージェントA、C、Gはレシーバを有する
エージェントB、C、D、Eはトランスミッタを有する

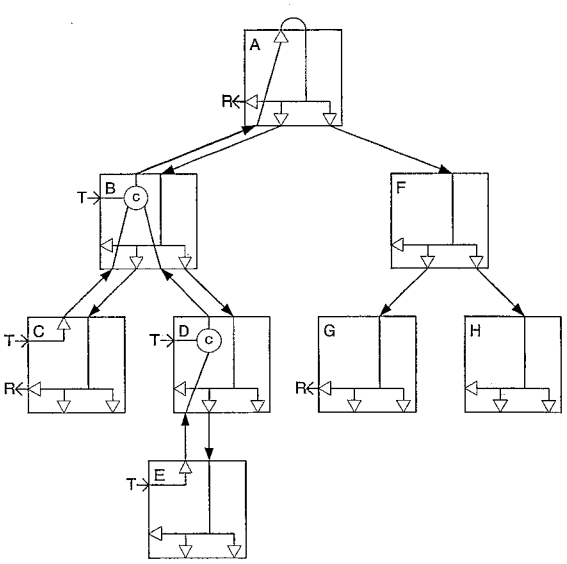
特定のバス信号に関するレシーバおよびトランスミッタ

【図３Ｂ】



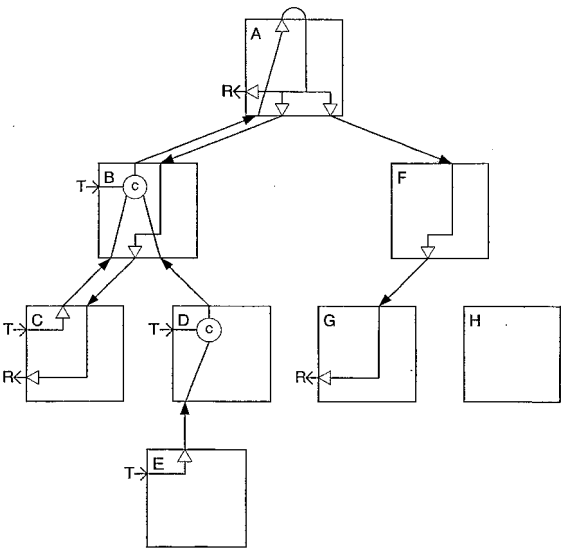
ｍｕｘ＿ｔｒｅｅ Ａ{ＢＣ{ＤＥ０}} {ＦＧＨ} の最適化されていない接続

【図３Ｃ】



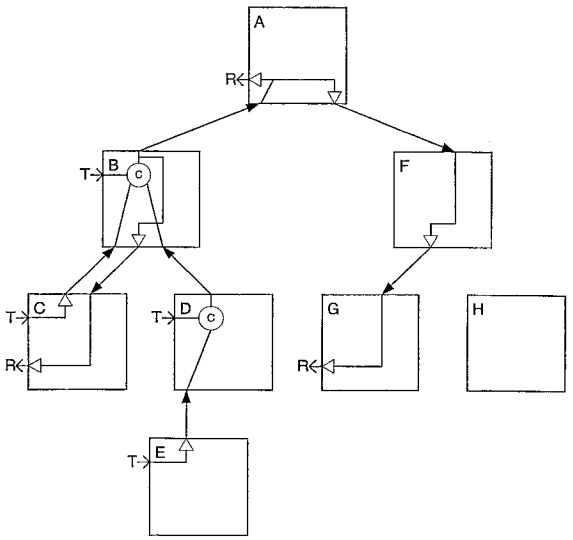
コンバイナ最適化の後

【図３Ｄ】



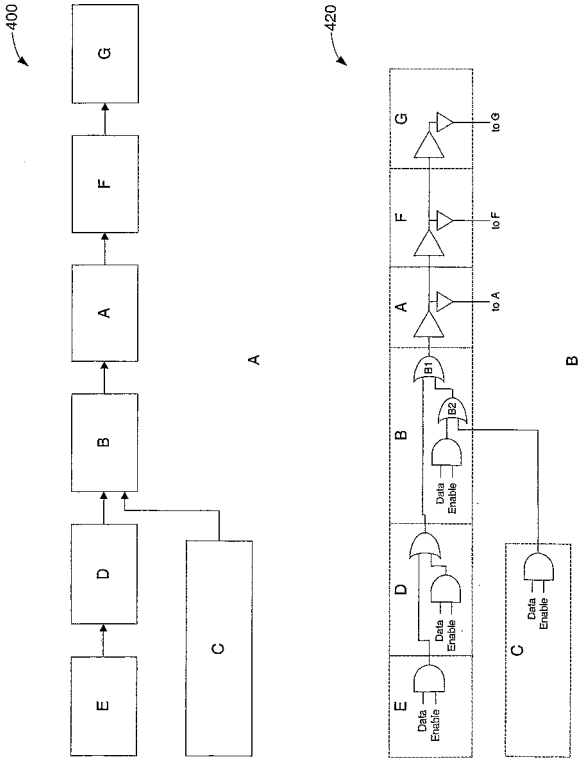
リピータ最適化の後

【図３Ｅ】

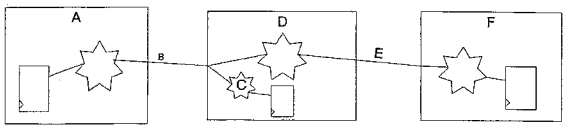


ルート最適化の後

【図４】

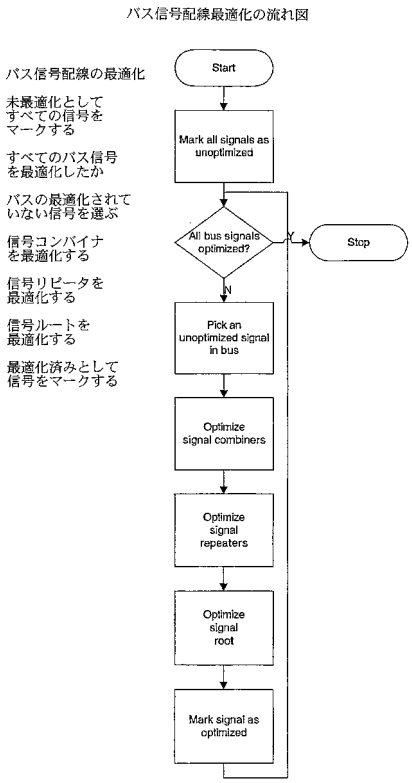


【図５】

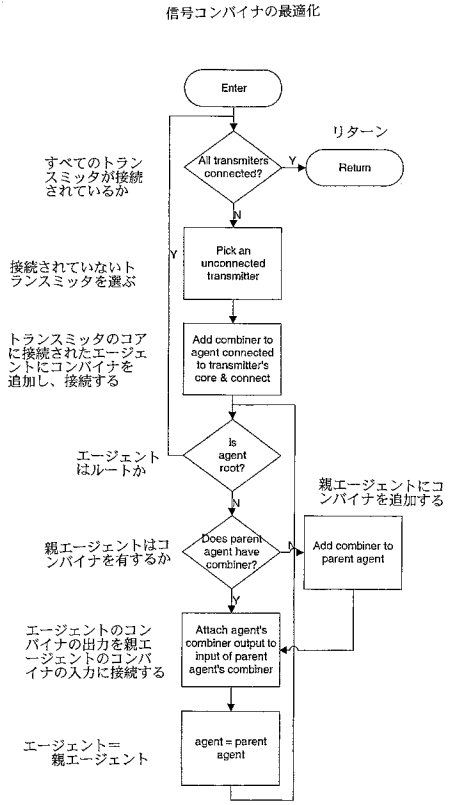


パスA、B、D、E、Fは、クリティカル・パスであり、クロック期間に合う必要がある。パスA、B、Cは、はるかに短いが、セグメントAは、タイミング要件を満たす必要がある。セグメントCだけをスケーリングすることができる。

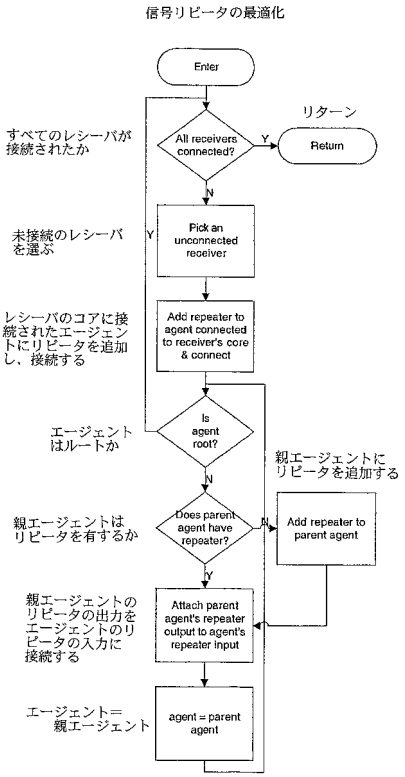
【図６Ａ】



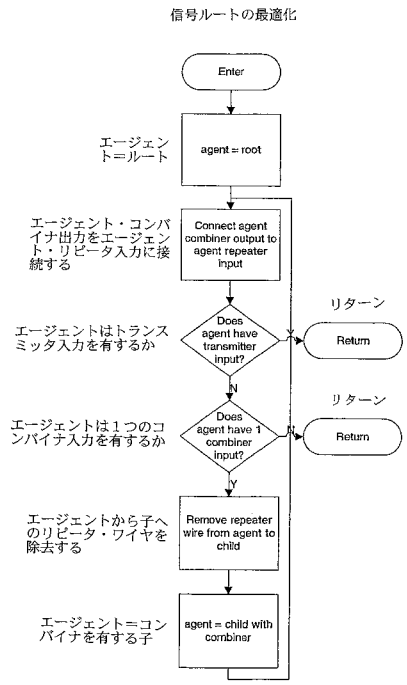
【図６Ｂ】



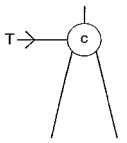
【図６Ｃ】



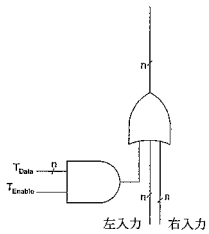
【図６Ｄ】



【図７】



A: コンバイナ・ブロック



Ｔｅｎａｂｌｅは単一ビットであり、
それ以外はｎビット信号である

B: コンバイナ・ブロックの詳細

left input: 左入力
right input: 右入力
bus state: バス状態

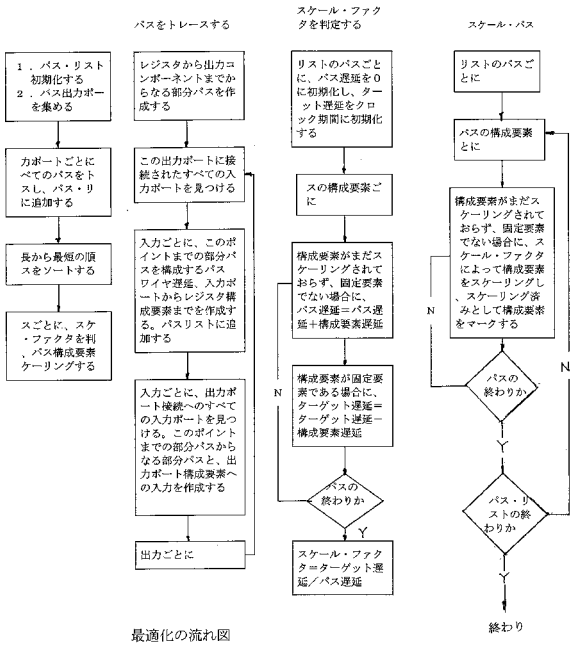
No conflict: 衝突なし
Conflict: 衝突

Tenable	left input	right input	bus state
0	0	0	No conflict
1	!=0	X	Conflict
1	X	!=0	Conflict
X	!=0	!=0	Conflict

表１：衝突検出

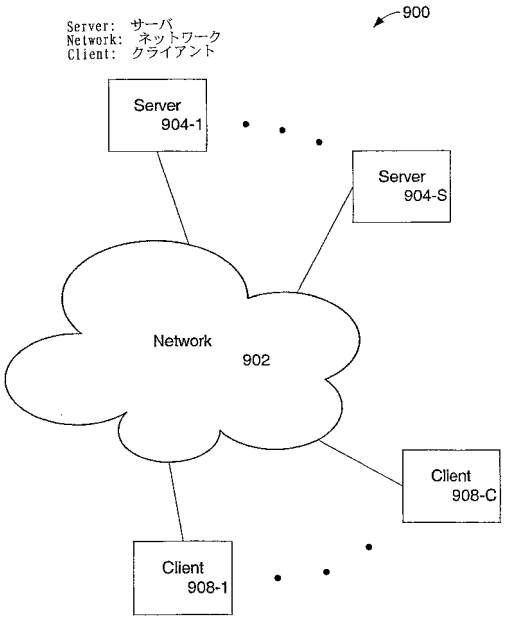
C

【図８】

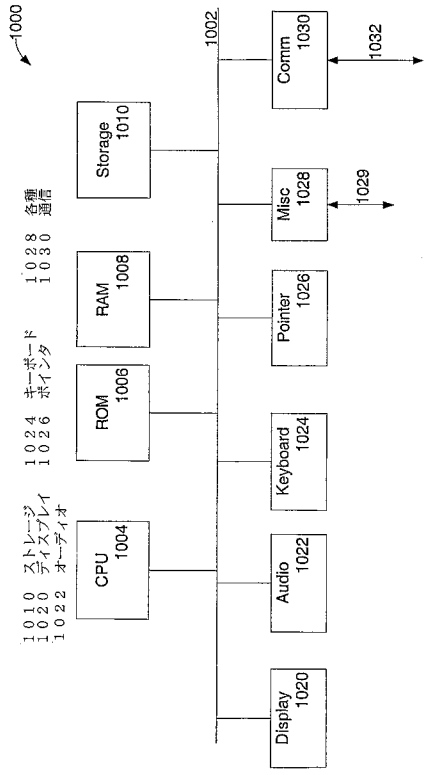


最適化の流れ図

【図９】



【図１０】



フロントページの続き

(72)発明者　エヴァンス，スコット・シイ
アメリカ合衆国・９５０５１・カリフォルニア州・サンタクララ・ヘイズ　アベニュ・２５２
(72)発明者　サイネック，カミル
アメリカ合衆国・９４０８７・カリフォルニア州・サニイベイル・マラード　ウエイ・１５１３

審査官　早川　学

(56)参考文献　特開平０９－０３４９２７（ＪＰ，Ａ）
国際公開第００／０１９３４３（WO，Ａ１）
特表２００５－５３９４０８（ＪＰ，Ａ）
Pierre Guerrier, et al., A Generic Architecture for On-Chip Packet-Switched Interconnections, Design, Automation, and Test in Europe, ACM, ２０００年, pp.250-256

(58)調査した分野(Int.Cl.，ＤＢ名)
G06F 17/50