(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2009-282976
(P2009-282976A)
(43)公開日　平成21年12月3日(2009.12.3)

(51)Int.Cl.		F I			テーマコード（参考）
G06F 17/50	(2006.01)	G06F	17/50	658E	5B046
		G06F	17/50	658U	
		G06F	17/50	658V	

審査請求 未請求　請求項の数 27　ＯＬ　外国語出願　（全 95 頁）

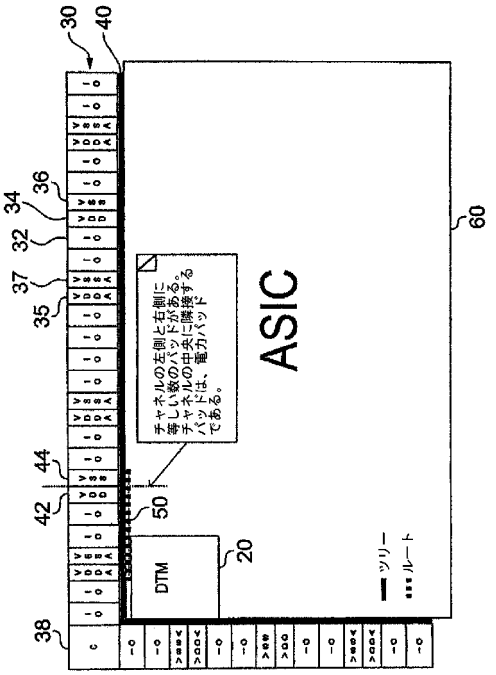
(21)出願番号　特願2009-120244 (P2009-120244)
(22)出願日　平成21年5月18日 (2009.5.18)
(31)優先権主張番号　12/153,448
(32)優先日　平成20年5月19日 (2008.5.19)
(33)優先権主張国　米国 (US)

(71)出願人　504394342
アーム・リミテッド
イギリス・ＣＢ１・９ＮＪ・ケンブリッジ・チェリー・ヒントン・フルボーン・ロード・１１０
(74)代理人　100108453
弁理士　村山　靖彦
(74)代理人　100064908
弁理士　志賀　正武
(74)代理人　100089037
弁理士　渡邊　隆
(74)代理人　100110364
弁理士　実広　信哉

最終頁に続く

(54)【発明の名称】相互接続回路内のデータ経路のルーティングを求める方法、システム、およびコンピュータプログラム製品

(57)【要約】
【課題】様々なデータ経路間の高精度の遅延整合を可能にする、集積回路の相互接続回路内のデータ経路のルーティングを求める改良型の技術を提供する。
【解決手段】集積回路についての相互接続回路内のデータ経路のルーティングを求めるシステム、方法、およびコンピュータプログラム製品が提供される。第1辺上の相互接続回路が、第1デバイスに対する接続のためのナローインターフェースを提供し、第2辺上の相互接続回路が、分散した複数の別のデバイスに対する接続のためのワイドインターフェースを提供する。各データ経路が、別のデバイスのうちの1つに関連付けられ、その関連付けられる別のデバイスと第1デバイスとの間の相互接続回路を通じて接続を実現する。
【選択図】図1



【特許請求の範囲】
【請求項１】
集積回路についての相互接続回路内のデータ経路のルーティングを求める方法において、第1辺上の前記相互接続回路が、第1デバイスに対する接続のためのナローインターフェースを提供し、第2辺上の前記相互接続回路が、分散した複数の別のデバイスに対する接続のためのワイドインターフェースを提供し、各データ経路が、前記別のデバイスのうちの1つに関連付けられ、関連付けられる前記別のデバイスと前記第1デバイスとの間の前記相互接続回路を通じて接続を実現する方法であって、
(i)前記ワイドインターフェースに沿って設けるべき複数のセルを定義するステップであって、前記別のデバイスのそれぞれが、前記セルのうちの少なくとも1つに関連付けられるステップと、
(ii)行および列として形成されたブロックのアレイとして前記相互接続回路を定義するステップであって、各セルが前記列のうちの1つに当接するステップと、
(iii)所定の1組のタイルを設けるステップであって、各タイルが所定の配線レイアウトを提供するステップと、
(iv)各ブロックについて、所定の規則を適用して、前記ブロックを実装するのに使用すべき前記タイルのうちの1つを決定するステップであって、前記所定の規則が、前記アレイ内の前記ブロックの位置、および前記複数の別のデバイスと前記複数のセルの間の関連付けを考慮に入れて、前記相互接続回路によって設けられる各データ経路が同一の伝播遅延を有することを保証するステップと
を含む方法。
【請求項２】
各データ経路が、タイルのシーケンスによって形成され、前記シーケンス中の各タイルが、前記シーケンスの隣接するタイル内の前記配線レイアウトと少なくとも部分的に当接する配線レイアウトを有するように選択される請求項1に記載の方法。
【請求項３】
前記所定の規則が、ツリー構造を生成するようにタイルを前記アレイ内の個々のブロックに割り振り、少なくとも1つの列が、ツリー構造のトランクとして形成されるナローインターフェースに当接し、残りの列内の前記行が、前記ツリー構造のブランチを形成し、
前記ステップ(iv)で、前記アレイ内の個々のブロックを実装するのに使用すべきタイルを決定するときに、前記所定の規則が、ブロックが前記ツリー構造の前記トランク内にあるか、それともツリー構造のブランチ内にあるかを考慮に入れる請求項1に記載の方法。
【請求項４】
前記所定のタイルの組が、トランクタイルの所定のサブセットと、ブランチタイルの所定のサブセットとを含み、前記ステップ(iv)中に、各ブロックについて、前記所定の規則が、前記ブロックが前記ツリー構造の前記トランク内にある場合にトランクタイルの前記サブセットからタイルを選択し、または前記ブロックが前記ツリー構造の前記ブランチ内にある場合にブランチタイルの前記サブセットからタイルを選択する請求項3に記載の方法。
【請求項５】
セルの数が、前記ワイドインターフェースに接続すべき別のデバイスの数を超過し、それによって、前記別のデバイスのうちの1つに関連付けられない少なくとも1つの非接続セルが存在し、
前記ツリー構造の前記トランクとして形成された各列について、前記列に当接するように選ばれた前記セルが非接続セルである請求項3に記載の方法。
【請求項６】
前記ツリー構造が、1つまたは複数のツリー区間を含み、各ツリー区間が、前記ナローインターフェースに当接して前記ツリー区間に対するトランクを形成する1つの列を有し、前記ツリー区間の残りの列内の前記行が、前記ツリー区間のブランチを形成する請求項3に記載の方法。

【請求項７】
少なくとも1つのツリー区間が、少なくとも1つの他のツリー区間とは異なる軸上に設けられる請求項6に記載の方法。
【請求項８】
各ツリー区間が、左ツリー区間または右ツリー区間を含み、前記ステップ(iv)で、適用される所定の規則が、タイルが左ツリー区間のブロックについて決定されるか、それとも右ツリー区間のブロックについて決定されるかに依存すると共に、左ツリー区間内であっても、右ツリー区間内であっても、前記相互接続回路によって与えられる各データ経路が同一の伝播遅延を有することを依然として保証する請求項6に記載の方法。
【請求項９】
前記分散した複数の別のデバイスが、少なくとも1つの軸をたどる分散経路に沿って分散する請求項1に記載の方法。
【請求項１０】
分散経路中の少なくとも1つの地点で、軸の変化が生じ、
前記ステップ(ii)で、軸の変化が生じる前記分散経路中の各地点に関連して、前記相互接続回路内にコーナ素子を定義し、それによって各コーナ素子が、前記アレイ内のブロックの2つの隣接する列間の軸の変化を定義するステップと、
前記ステップ(iv)で、前記所定の規則が、前記所定のタイルの組と共に含まれる所定のコーナタイルを使用して各コーナ素子を実装し、前記アレイ内の各ブロックについて使用すべきタイルを決定するときに、前記所定の規則が、前記相互接続回路によって設けられる各データ経路が同一の伝播遅延を有することを保証するように、前記相互接続回路内の任意のコーナ素子の位置を考慮に入れるステップと
をさらに含む請求項9に記載の方法。
【請求項１１】
前記ステップ(ii)で、前記コーナ素子のどちらかの側にフィラー素子が定義され、前記ステップ(iv)で、前記所定の規則が、前記所定のタイルの組に含まれる所定のフィラータイルを使用して各フィラー素子を実装する請求項10に記載の方法。
【請求項１２】
セルの数が、前記ワイドインターフェースに接続すべき別のデバイスの数を超過し、それによって、前記別のデバイスのうちの1つに関連付けられない少なくとも1つの非接続セルが存在し、前記相互接続回路に対する接続を必要としない前記集積回路の1つまたは複数の他の構成要素を、1つまたは複数の非接続セルに関連付けるステップをさらに含む請求項1に記載の方法。
【請求項１３】
前記他の構成要素が、前記分散した複数の別のデバイスに電力供給するのに使用される電力供給線を備える請求項12に記載の方法。
【請求項１４】
前記ステップ(iv)中に、前記所定の規則が、特定のブロックの位置で配線レイアウトが不要であると前記所定の規則が判定した場合、前記ブロックについてタイルが割り振られない請求項1に記載の方法。
【請求項１５】
前記所定のタイルの組が、配線レイアウトを有さない1つまたは複数のフィラータイルを含み、前記ステップ(iv)中に、特定のブロックの位置で配線レイアウトが不要であると前記所定の規則が判定した場合、前記フィラータイルのうちの1つが、前記ブロックに対して割り振られる請求項1に記載の方法。
【請求項１６】
前記ステップ(i)で、定義される前記複数のセルが、複数の異なる幅であり、前記所定のタイルの組が、各セルが前記列のうちの1つに当接することが可能となるように、対応する複数の異なる幅を有するタイルを含む請求項1に記載の方法。
【請求項１７】

前記ステップ(i)で、一定のセルが、初期セルとフィラーセルの組合せによって形成され、それによって前記アレイ内の列と当接する各セルが、等しい幅となる請求項1に記載の方法。

【請求項１８】

前記相互接続回路の前記ナローインターフェースから前記第1デバイスまで通るルートチャネルを定義し、それによって前記集積回路内の前記ナローインターフェース以外の位置への第1デバイスの配置を可能にするステップをさらに含む請求項1に記載の方法。

【請求項１９】

前記所定のタイルの組が、ルートタイルのサブセットを含み、前記サブセットからの1つまたは複数のルートタイルを使用して前記ルートチャネルを生成するステップをさらに含む請求項18に記載の方法。

【請求項２０】

前記分散した複数の別のデバイスが、前記集積回路の周辺に配置された入力/出力デバイスを含む請求項1に記載の方法。

【請求項２１】

前記第1デバイスが、前記複数の入力/出力デバイスを制御するのに使用されるメモリコントローラデバイスである請求項20に記載の方法。

【請求項２２】

前記別のデバイスのピンと、前記所定のタイルの組の中のタイルの配線レイアウトとの間のピンマッピングを実現する1つまたは複数のブリッジタイルを設けるステップと、

特定の別のデバイスについて、前記特定の別のデバイスのピンの、当接すべき前記相互接続回路内の前記タイルの前記配線レイアウトに対する直接的マッピングがないとき、前記特定の別のデバイスと前記相互接続回路内の前記タイルとの間に少なくとも1つのブリッジタイルを導入して、前記特定の別のデバイスのピンを前記タイルの前記配線レイアウトにマッピングするステップと

をさらに含む請求項1に記載の方法。

【請求項２３】

セルの数が、前記ワイドインターフェースに接続すべき別のデバイスの数を超過し、それによって、前記別のデバイスのうちの1つに関連付けられない少なくとも1つの非接続セルが存在し、ワイドインターフェースに沿った前記非接続セルの追加が、ナローインターフェースの幅に影響を及ぼさない請求項1に記載の方法。

【請求項２４】

前記ワイドインターフェースに沿った前記非接続セルの追加が、前記データ経路が前記ナローインターフェースに接続される順序に影響を及ぼさない請求項23に記載の方法。

【請求項２５】

前記所定の組の中の前記タイルのうちの少なくとも1つの配線レイアウトが、前記タイルのうちの少なくとも1つが前記アレイ内のブロックを実装するのに使用されるとき、前記タイル内の前記配線レイアウトのすべてが使用されるのではないようなものとなる請求項1に記載の方法。

【請求項２６】

コンピュータシステム上で実行されたときに、集積回路についての相互接続回路内のデータ経路のルーティングを求める方法を前記コンピュータシステムに実行させるコンピュータプログラムを含むコンピュータプログラム製品において、第1辺上の前記相互接続回路が、第1デバイスに対する接続のためのナローインターフェースを提供し、第2辺上の前記相互接続回路が、分散した複数の別のデバイスに対する接続のためのワイドインターフェースを提供し、各データ経路が、前記別のデバイスのうちの1つに関連付けられ、関連付けられる前記別のデバイスと前記第1デバイスとの間の前記相互接続回路を通じて接続を実現するコンピュータプログラム製品であって、前記方法が、

(i)前記ワイドインターフェースに沿って設けるべき複数のセルを定義するステップであって、前記別のデバイスのそれぞれが、前記セルのうちの少なくとも1つに関連付けら

れるステップと、
(ii)行および列として形成されたブロックのアレイとして前記相互接続回路を定義するステップであって、各セルが前記列のうちの1つに当接するステップと、
(iii)所定の1組のタイルを設けるステップであって、各タイルが所定の配線レイアウトを提供するステップと、
(iv)各ブロックについて、所定の規則を適用して、前記ブロックを実装するのに使用すべき前記タイルのうちの1つを決定するステップであって、前記所定の規則が、前記アレイ内の前記ブロックの位置、および前記複数の別のデバイスと前記複数のセルの間の関連付けを考慮に入れて、前記相互接続回路によって設けられる各データ経路が同一の伝播遅延を有することを保証するステップと
を含むコンピュータプログラム製品。
【請求項２７】
集積回路についての相互接続回路内のデータ経路のルーティングを求めるシステムにおいて、第1辺上の前記相互接続回路が、第1デバイスに対する接続のためのナローインターフェースを提供し、第2辺上の前記相互接続回路が、分散した複数の別のデバイスに対する接続のためのワイドインターフェースを提供し、各データ経路が、前記別のデバイスのうちの1つに関連付けられ、関連付けられる前記別のデバイスと前記第1デバイスとの間の前記相互接続回路を通じて接続を実現するシステムであって、
前記ワイドインターフェースに沿って設けるべき複数のセルを定義する手段であって、前記別のデバイスのそれぞれが、前記セルのうちの少なくとも1つに関連付けられる手段と、
行および列として形成されたブロックのアレイとして前記相互接続回路を定義する手段であって、各セルが前記列のうちの1つに当接する手段と、
所定の1組のタイルを設ける手段であって、各タイルが所定の配線レイアウトを提供する手段と、
各ブロックについて、所定の規則を適用して、前記ブロックを実装するのに使用すべき前記タイルのうちの1つを決定するデータ経路決定手段であって、前記所定の規則が、前記アレイ内の前記ブロックの位置、および前記複数の別のデバイスと前記複数のセルの間の関連付けを考慮に入れて、前記相互接続回路によって設けられる各データ経路が同一の伝播遅延を有することを保証するデータ経路決定手段と
を備えるシステム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、集積回路の相互接続回路内のデータ経路のルーティングを求める技術に関する。
【背景技術】
【０００２】
ASIC(特定用途向け集積回路)設計フロープロセスで使用されるデジタルレイアウトEDA(electronic design automation)ツールなどの現在の設計ツールは、ディープサブミクロンプロセスで高速ネットおよびバスに対して必要とされることがあるような、相互接続回路のタイミングクリティカルなデータ経路に関する遅延整合で高精度を達成する際の厳しい制限を有する。この制限は主に、EDAツールがどのように問題に取り組むかということによるものである。
【０００３】
現在のデジタルレイアウトEDAツールは、タイミング制約を、設計レイアウトに対する要件として使用する。これはしばしば、「タイミングドリブン位置およびルート(timing driven place and route)」と呼ばれる。これにより、ASICレイアウトツールには、実際には無制限の数のレイアウト実装が提示される。このジレンマに直面して、EDAツールはしばしば、発見的アルゴリズムを使用してASICレイアウトを自動的に生成する。発見的ア

ルゴリズムとは、最適解を求めるように設計されるのではなく、目標関数の点で最適解に近い、実現可能な解を見つけることを目標とするアルゴリズムである。設計の初期の配置およびルートが設計タイミング制約を満たすことはまれである。結果として、レイアウトツールは、設計タイミング制約を満たすことを望みながら、設計のタイミングクリティカルな部分の位置およびルートの反復に頼る。

【０００４】

例えば、高速平衡バス相互接続の場合に通常はそうであるように、設計要件が非常に限定的なものであるとき、レイアウト反復は往々にして収束せず、したがってツールは、使用可能なレイアウトの生成に失敗する。レイアウトツールが設計要件を満たすことに成功しないとき、生成されるレイアウトは、素子配置または信号ルーティングのどちらかで対称性がない。ネットはランダムな長さとなり、層から層へランダムにジャンプする。得られる各ネットのキャパシタンス、抵抗、カップリング、およびインダクタンスを正確に管理することができない。この結果、レイアウトのタイミング不確定性が生じる。

【０００５】

周知のデジタルレイアウトEDAツールは、タイミングを改善するいくつかの技術を使用する。こうした技術は、ある駆動素子を、同じ機能を有するが異なる駆動強度を有する別の駆動素子と置き換えること、クリティカルパス上にバッファを挿入して、遅延を加速または減速すること、および/または長いネットをレピータバッファで分割すること、重要な各セルを物理的に近くに移動することを含む。こうした手法によるタイミング変動が量子化され、タイミング不確定性を導入する。異なるタイミングクリティカルパス間の整合も、プロセスコーナにわたって非常に低減される。こうした経路は、異なる素子数、または異なる駆動特性を有する素子で構築されるからである。異なる素子のタイミング特性は、プロセス-温度-電圧コーナにわたって同一にスケーリングしない。あるコーナを合致させるとしばしば、別のコーナが不整合となる。

【０００６】

デジタルEDAレイアウトアルゴリズムの発見的な性質のために、異なるツールバージョン、または異なるベンダからのバージョンで、同一の設計が劇的に異なるレイアウトを生成する可能性がある。レイアウト手続きが反復されるとき、しばしば同一のツールが、同じ設計で異なるレイアウトを生成する。

【０００７】

現在のデジタルEDAレイアウトフローの別の制限は、ツールが、2つの別々のワイヤ上で送られる2つの補信号によって情報を電気的に送信する方法である差分信号を扱うことが不可能であることである。そのような差分信号技術は、例えばデジタルゲートのスイッチングによって生成されるような雑音のある環境を介する高速信号の正確な伝播を可能にすることができる。

【０００８】

結果として、周知のデジタルレイアウトEDAツールおよびASICフローは、相互接続回路内のクリティカルタイミング経路、ネット、およびバス(本明細書では「データ経路」と呼ぶ)の高度な遅延整合および雑音無影響性を必要とする設計レイアウトを生成することができない。

【０００９】

周知のデジタルレイアウトEDAツールの制限は、バスの異なるビットを物理的に接近していない設計ブロックにルーティングしなければならず、しかもタイミングの整合が必要とされるときに悪化する。DDR(double data rate)メモリインターフェースは、この制限の良い例である。DDRメモリは、ダブルポンピングする(クロック信号の立上りおよび立下り時にデータを転送する)ことにより、クロック周波数を上昇させることなく、非DDR(single data rate)メモリのほぼ2倍の帯域幅を達成する。DDRメモリコントローラは、それが接続しなければならない入出力(I/O)デバイスのスパンに対して比較的小型のレイアウトを有し、これにより、メモリコントローラをこうした分散I/Oデバイスと結合するのに使用される相互接続回路の設計に対して著しいタイミング問題が課される。

【００１０】
一例を挙げると、DDR3は、1600Mbsものデータ転送速度でオフチップSDRAMからのデータを読み書きするのにクロックの両端を使用するパラレルインターフェースである。データ信号DQ、ストローブ信号DQS、およびクロック信号CKの間のタイミング関係をしっかりと制御しなければならず、しかもこの設計に関するI/Oデバイスは、直線的に2mm超に及ぶことがある(すなわち、メモリコントローラブロックのサイズよりも何桁も大きい)。
【００１１】
クロックジッタ、オンチップ変動、プロセス-電圧-温度変動、プリント回路トレース変動、システム雑音、SDRAMセットアップおよび保持タイミングなどの制御不能タイミング不確定性を考慮に入れることにより、相互接続回路内のメモリコントローラとI/Oデバイスの間のデータ経路の伝播遅延を数十ナノ秒以内に整合させることが必要となる。2mm以上のネットについて数百ナノ秒の領域内にある相互接続遅延を仮定すると、そのような相互接続回路の設計は、現在のデジタルレイアウトEDAツールでは達成が困難である。
【発明の概要】
【発明が解決しようとする課題】
【００１２】
したがって、様々なデータ経路間の高精度の遅延整合を可能にする、集積回路の相互接続回路内のデータ経路のルーティングを求める改良型の技術を提供することが望ましい。
【課題を解決するための手段】
【００１３】
第1態様から見ると、本発明は、集積回路についての相互接続回路内のデータ経路のルーティングを求める方法において、第1辺上の相互接続回路が、第1デバイスに対する接続のためのナローインターフェースを提供し、第2辺上の相互接続回路が、分散した複数の別のデバイスに対する接続のためのワイドインターフェースを提供し、各データ経路が、前記別のデバイスのうちの1つに関連付けられ、その関連付けられる別のデバイスと第1デバイスとの間の相互接続回路を通じて接続を実現する方法であって、
(i)ワイドインターフェースに沿って設けるべき複数のセルを定義するステップであって、別のデバイスのそれぞれが、前記セルのうちの少なくとも1つに関連付けられるステップと、(ii)行および列として形成されたブロックのアレイとして相互接続回路を定義するステップであって、各セルが前記列のうちの1つに当接するステップと、(iii)所定の1組のタイルを設けるステップであって、各タイルが所定の配線レイアウトを提供するステップと、(iv)各ブロックについて、所定の規則を適用して、そのブロックを実装するのに使用すべき前記タイルのうちの1つを決定するステップであって、所定の規則が、アレイ内のブロックの位置、ならびに複数の別のデバイスと複数のセルの間の関連付けを考慮に入れて、相互接続回路によって設けられる各データ経路が同一の伝播遅延を有することを保証するステップとを含む方法を提供する。
【００１４】
本発明によれば、第1デバイスを分散した複数の別のデバイスと結合するのに使用される相互接続回路内のデータ経路のルーティングを求める方法が提供される。相互接続回路は、第1デバイスと結合するためのナローインターフェースと、分散した複数の別のデバイスと結合するためのワイドインターフェースとを有する。相互接続回路は、行および列として形成されたブロックのアレイとみなされ、所定の1組のタイルが、相互接続回路を構築する際に使用するために提供され、各タイルは所定の配線レイアウトを提供する。各ブロックについて、所定の規則が適用され、そのブロックを実装するのに使用すべきタイルのうちの1つが決定される。アレイ内のブロックの位置、および複数の別のデバイスとワイドインターフェースに沿って設けられる複数のセルとの間の関連付けを考慮に入れるように所定の規則を構成することにより、このタイル割振りプロセスは、相互接続回路によって設けられる各データ経路が同一の伝播遅延を有することを保証する。
【００１５】
したがって、本発明が、所定のタイルを使用する構造化ルーティング方法を提供し、相

互接続回路によって設けられる各データ経路が同一の伝播遅延を有することを保証するように、所定の規則が適用されて、相互接続回路を形成するブロックのアレイ内の各ブロックを実装するのにどのタイルが使用されるかが決定されることがわかる。

【００１６】

本発明を使用することにより、周知の従来技術配置ツールと比較して、いくつかの利点を得ることができる。まず、相互接続回路内の各データ経路が同一の伝播遅延を有する効果的な機構を提供することに加えて、プロセスは、カップリング、漏話、ローディング、雑音遮蔽などの他の物理的効果も整合させることができる。さらに、本発明の実施形態では、何ら反復する必要なしに方法を実装することができ、したがって、実行可能なルーティング設計を見つける目的で反復プロセスを実施する必要がある周知のEDAツールと比較したときに、著しい設計時間速度の利点が得られる。さらに、本発明は、所定の配線レイアウトを有するタイルを使用して、同一の伝播遅延を有するデータ経路を生成するものであるので、これにより、タイミングおよび整合を改善するために一定の経路中にバッファおよび/または他の構成要素をしばしば挿入する周知のEDAツールで設計されたものよりも、消費する電力が著しく低い相互接続回路を生み出すことができる。

【００１７】

一実施形態では、各データ経路が、タイルのシーケンスによって形成され、シーケンス中の各タイルは、シーケンスの隣接するタイル内の配線レイアウトと少なくとも部分的に当接する配線レイアウトを有するように選択される。したがって、各データ経路は、そのデータ経路を形成するのに使用されるタイルの関連するシーケンス内の様々な配線レイアウトの組合せによって形成される。各タイル内のワイヤの物理的レイアウトのために、各タイルに関連する伝播遅延が所定のものとなる。所定のタイルの組は、様々な異なる配線レイアウトに異なる伝播遅延を与え、これにより、伝播遅延があらゆるデータ経路について整合されることを保証するために、アレイ内のブロックの位置、および相互接続回路に接続すべき様々な別のデバイスの配置を考慮に入れて、所定の規則が、各ブロックを実装するのに使用すべき適切なタイルを決定することが可能となる。

【００１８】

一実施形態では、所定の規則は、ツリー構造を生成するようにタイルをアレイ内の個々のブロックに割り振り、少なくとも1つの列が、ツリー構造のトランクとして形成されるナローインターフェースに当接し、残りの列内の行が、ツリー構造のブランチを形成する。前記ステップ(iv)で、アレイ内の個々のブロックを実装するのに使用すべきタイルを決定するときに、前記所定の規則は、ブロックがツリー構造のトランク内にあるか、それともツリー構造のブランチ内にあるかを考慮に入れる。そのようなツリー構造の使用により、特に効率的なデータ経路のルーティングを達成することが可能となることが判明した。

【００１９】

一実施形態では、前記所定のタイルの組は、トランクタイルの所定のサブセットと、ブランチタイルの所定のサブセットとを含み、前記ステップ(iv)中に、各ブロックについて、所定の規則が、ブロックがツリー構造のトランク内にある場合にトランクタイルのサブセットからタイルを選択し、またはブロックがツリー構造のブランチ内にある場合にブランチタイルのサブセットからタイルを選択する。

【００２０】

一実施形態では、セルの数が、前記ワイドインターフェースに接続すべき別のデバイスの数を超過し、それによって、前記別のデバイスのうちの1つに関連付けられない少なくとも1つの非接続セルが存在し、ツリー構造のトランクとして形成された各列について、その列に当接するように選ばれたセルが非接続セルである。

【００２１】

したがって、そのような実施形態では、所定の規則が使用されて、ツリー構造のトランクを形成するブロックの列を実装するのに使用されるトランクタイルが選択されるとき、そうしたタイルが別のデバイスのうちの1つとの何らかの直接的接続を実現するための要件はなく、様々な別のデバイスとの接続が、ツリー構造のブランチを実装するのに使用さ

れるブランチタイルを介して行われる。

【００２２】

様々な方式でツリー構造を設計することができる。しかし、一実施形態では、ツリー構造は、1つまたは複数のツリー区間を含み、各ツリー区間は、ナローインターフェースに当接してそのツリー区間に対するトランクを形成する1つの列を有し、ツリー区間の残りの列内の行が、そのツリー区間のブランチを形成する。別々のツリー区間からツリー構造を形成することを可能にすることにより、データ経路ルーティングのレイアウトに関する柔軟性、および相互接続回路のワイドインターフェースに接続すべき別のデバイスの配置に関する柔軟性の改善が実現される。例えば、一実施形態では、少なくとも1つのツリー区間が、少なくとも1つの他のツリー区間とは異なる軸上に設けられ、それによって別のデバイスを複数の軸にわたって分散させることが可能となる。一実施形態では、そのような軸の変化をツリー区間自体の中に吸収することもできる。

【００２３】

様々なタイプのツリー区間を設けることができる。一実施形態では、各ツリー区間は、左ツリー区間または右ツリー区間を含み、前記ステップ(iv)で、適用される所定の規則が、タイルが左ツリー区間のブロックについて決定されるか、それとも右ツリー区間のブロックについて決定されるかに依存すると共に、左ツリー区間内であっても、右ツリー区間内であっても、相互接続回路によって与えられる各データ経路が同一の伝播遅延を有することを依然として保証する。一実施形態では、そのような手法を使用して、相互接続回路のデータ経路レイアウトを、概して中央のトランクを有し、トランクからブランチが対向する方向に形成されるツリー構造として生成することができる。一実施形態では、左ツリー区間と右ツリー区間を共に同じ列数および行数を有するように構成することができ、それによって、ツリー構造のトランクを中心とした対称的設計を生成することが可能となる。

【００２４】

分散した複数の別のデバイスを集積回路上で様々な方式で配置させることができる。しかし、一実施形態では、別のデバイスは、少なくとも1つの軸をたどる分散経路上に分散する。一実施形態では、別のデバイスをすべて線形経路に沿って配置することができ、別の実施形態では、互いに異なる向きでよい複数の線形経路に沿ってデバイスを設けることができる。

【００２５】

一実施形態では、分散経路中の少なくとも1つの地点で、アクセスの変化が生じ、この方法は、前記ステップ(ii)で、軸の変化が生じる分散経路中の各地点に関連して、相互接続回路内にコーナ素子を定義し、それによって各コーナ素子が、アレイ内のブロックの2つの隣接する列間の軸の変化を定義するステップと、前記ステップ(iv)で、所定の規則が、前記所定のタイルの組と共に含まれる所定のコーナタイルを使用して各コーナ素子を実装し、アレイ内の各ブロックについて使用すべきタイルを決定するときに、所定の規則が、相互接続回路によって設けられる各データ経路が同一の伝播遅延を有することを保証するように、相互接続回路内の任意のコーナ素子の位置を考慮に入れるステップとをさらに含む。

【００２６】

所定のコーナタイルを使用することにより、そのようなコーナタイルを使用することに伴う伝播遅延が所定のものとなり、したがって、所定の規則は、相互接続回路内の各データ経路が同一の伝播遅延を有することを保証するように、各ブロックについて相互接続回路内のそのブロックを実装するのに使用すべき適切なタイルを決定するときに、任意のコーナ素子の位置を考慮に入れることができる。

【００２７】

コーナ素子が比較的短い場合、これにより、レイアウトが密集することなく(トランクに関して)コーナ素子の反対側に設けることのできるブランチ数が限定される。一実施形態では、このことを、コーナ素子のどちらかの側にフィラー素子を定義し、次いで所定の

タイルの組内に提供される所定のフィラータイルを使用して各フィラー素子を実装することによって軽減することができる。コーナタイルと組み合わせたそのようなフィラータイルは、コーナ素子のサイズを実質上拡張し、コーナ素子の反対側で多数のブランチを必要とするツリー構造を構築することが可能となる。

【００２８】

非接続セルが存在する実施形態では、この方法は、相互接続回路への接続を必要としない集積回路の1つまたは複数の他の構成要素を、1つまたは複数の非接続セルに関連付けるステップをさらに含むことができる。非接続セルに関連付けることのできるそのような構成要素の一例を挙げると、そのような構成要素は、相互接続回路に接続すべき分散した複数の別のデバイスに電力供給するのに使用される電力供給線を備えることができる。

【００２９】

所定の規則を適用して、相互接続回路を定義するブロックのアレイ内の各ブロックにどのタイルを割り振るかを決定するとき、特定のブロックの位置で配線レイアウトが不要であると所定の規則が判定したとき、一実施形態では、そのブロックについてタイルが割り振られない。代替実施形態では、所定のタイルの組は、配線レイアウトを有さない1つまたは複数のフィラータイルを含むことができ、特定のブロックの位置で配線レイアウトが不要であると所定の規則が判定した場合、こうしたフィラータイルのうちの1つが、そのブロックに対して割り振られる。一実施形態では、そのようなフィラータイルは、与えられたプロセスに対する金属密度要件を満たすようにメタルフィルを有することができる。

【００３０】

一実施形態では、すべてのセルは等しい幅を有し、その結果として、ブロックの列のそれぞれは等しい幅となり、したがって所定のタイルの組を等しい幅で作成することができる。しかし、代替実施形態では、定義される複数のセルは、複数の異なる幅であり、所定のタイルの組は、各セルが前記列のうちの1つに当接することが可能となるように、対応する複数の異なる幅を有するタイルを含む。

【００３１】

しかし、代替実施形態では、フィラーセルの使用により、あらゆる可能な幅のセルに対するタイルを設ける必要を回避することができる。より具体的には、前記ステップ(i)で、一定のセルが、初期セルとフィラーセルの組合せによって形成され、それによってアレイ内の列と当接する各セルが、等しい幅となる。そのような手法により、ブロックの各列を同じ幅に構成することができ、したがって、所定のタイルの組が含む必要があるのは、その指定の幅のタイルだけとなり、それによって所定の組の中で提供する必要のあるタイルの数が削減される。

【００３２】

一実施形態では、第1デバイスが、相互接続回路のナローインターフェースに直接的に接続される。しかし、代替実施形態では、この方法は、相互接続回路のナローインターフェースから第1デバイスまで通るルートチャネルを定義し、それによって集積回路内のナローインターフェース以外の位置への第1デバイスの配置を可能にするステップをさらに含む。特定の一実施形態では、前記所定のタイルの組は、ルートタイルのサブセットを含み、この方法は、前記サブセットからの1つまたは複数のルートタイルを使用してルートチャネルを生成するステップをさらに含む。そのような手法により、分散した複数の別のデバイスの配置に対する第1デバイスの配置に関する自由が得られると共に、相互接続回路によって与えられる各データ経路が同一の伝播遅延を有することが依然として保証される。具体的には、一実施形態では、ルートチャネルを通る各データ経路の伝播遅延は同一であり、したがって、ルートチャネルを含めることにより、本発明の実施形態の技術の伝播遅延整合特性が維持される。

【００３３】

第1デバイスおよび別のデバイスは、様々な形態を取ることができる。一実施形態では、分散した複数の別のデバイスは、集積回路の周辺に配置された入力/出力デバイスを含む。そのようなデバイスでは、第1デバイスは、複数の入力/出力デバイスを制御するのに

使用されるメモリコントローラデバイスである。SDR、DDRなどのメモリインターフェース、およびPCIなどのパラレルバスアーキテクチャは、本発明の実施形態のルーティング方法から恩恵を受けることのできる技術の主な例である。より一般には、本発明の実施形態の技術は、ワイドインターフェースに沿って物理的に比較的離れた間隔を置いて配置されたデバイスにナローインターフェースを接続するデータ経路に対して高度なタイミング整合が必要となる、すべての相互接続回路に適用可能である。

【００３４】

一実施形態では、様々な別のデバイスが、相互接続回路内の関連する列内の隣接するタイルと直接的に接続される。しかし、代替実施形態では、この方法は、前記別のデバイスのピンと、前記所定のタイルの組の中のタイルの配線レイアウトとの間のピンマッピングを実現する1つまたは複数のブリッジタイルを設けるステップと、特定の別のデバイスについて、そのピンの、当接すべき相互接続回路内のタイルの配線レイアウトに対する直接的マッピングがないとき、前記特定の別のデバイスと相互接続回路内の前記タイルとの間に少なくとも1つのブリッジタイルを導入して、前記特定の別のデバイスのピンをそのタイルの配線レイアウトにマッピングするステップとをさらに含む。したがって、そのようなブリッジタイルを使用して、相互接続回路内の関係するタイルと当接するのに必要なピン位置に適合するように、別のデバイス上のピン位置を変換することができる。そのようなブリッジタイルはまた、相互接続回路内の配線の、相互接続回路を介する接続を必要としない別のデバイス内のピンとの偶発的当接を回避する役割も果たすことができる。

【００３５】

一実施形態では、非接続セルがワイドインターフェースに沿って設けられる場合、そのような非接続セルの追加は、ナローインターフェースに影響を及ぼさない。さらに、そのような実施形態では、ワイドインターフェースに沿ったそのような非接続セルの追加は、データ経路がナローインターフェースに接続される順序に影響を及ぼさない。したがって、セルがワイドインターフェースに沿って設けられる方式、および別のデバイスと個々のセルとの間の関連付けにおいて高度な柔軟性がある。

【００３６】

一実施形態では、所定の組の中のタイルのうちの少なくとも1つの配線レイアウトは、タイルのうちの少なくとも1つが前記アレイ内のブロックを実装するのに使用されるとき、そのタイル内の配線レイアウトのすべてが使用されるのではないようなものとなる。したがって、そのような実施形態では、各タイルの配線レイアウト内の個々のワイヤのすべてが、相互接続回路内のそのタイルのすべての具体化で使用されるわけではない。そのようにタイルを設計することにより、そうしたタイルの使用に関する柔軟性の改善が得られる。

【００３７】

第2態様から見ると、本発明は、コンピュータシステム上で実行されたときに、集積回路についての相互接続回路内のデータ経路のルーティングを求める方法をコンピュータシステムに実行させるコンピュータプログラムを含むコンピュータプログラム製品において、第1辺上の相互接続回路が、第1デバイスに対する接続のためのナローインターフェースを提供し、第2辺上の相互接続回路が、分散した複数の別のデバイスに対する接続のためのワイドインターフェースを提供し、各データ経路が、前記別のデバイスのうちの1つに関連付けられ、その関連付けられる別のデバイスと第1デバイスとの間の相互接続回路を通じて接続を実現するコンピュータプログラム製品であって、方法が、(i)ワイドインターフェースに沿って設けるべき複数のセルを定義するステップであって、別のデバイスのそれぞれが、前記セルのうちの少なくとも1つに関連付けられるステップと、(ii)行および列として形成されたブロックのアレイとして相互接続回路を定義するステップであって、各セルが前記列のうちの1つに当接するステップと、(iii)所定の1組のタイルを設けるステップであって、各タイルが所定の配線レイアウトを提供するステップと、(iv)各ブロックについて、所定の規則を適用して、そのブロックを実装するのに使用すべき前記タイルのうちの1つを決定するステップであって、所定の規則が、アレイ内のブロックの位置

、および複数の別のデバイスと複数のセルの間の関連付けを考慮に入れて、相互接続回路によって設けられる各データ経路が同一の伝播遅延を有することを保証するステップとを含むコンピュータプログラム製品を提供する。

【００３８】

第3態様から見ると、本発明は、集積回路についての相互接続回路内のデータ経路のルーティングを求めるシステムにおいて、第1辺上の相互接続回路が、第1デバイスに対する接続のためのナローインターフェースを提供し、第2辺上の相互接続回路が、分散した複数の別のデバイスに対する接続のためのワイドインターフェースを提供し、各データ経路が、前記別のデバイスのうちの1つに関連付けられ、その関連付けられる別のデバイスと第1デバイスとの間の相互接続回路を通じて接続を実現するシステムであって、ワイドインターフェースに沿って設けるべき複数のセルを定義する手段であって、別のデバイスのそれぞれが、前記セルのうちの少なくとも1つに関連付けられる手段と、行および列として形成されたブロックのアレイとして相互接続回路を定義する手段であって、各セルが前記列のうちの1つに当接する手段と、所定の1組のタイルを設ける手段であって、各タイルが所定の配線レイアウトを提供する手段と、各ブロックについて、所定の規則を適用して、そのブロックを実装するのに使用すべき前記タイルのうちの1つを決定するデータ経路決定手段であって、所定の規則が、アレイ内のブロックの位置、および複数の別のデバイスと複数のセルの間の関連付けを考慮に入れて、相互接続回路によって設けられる各データ経路が同一の伝播遅延を有することを保証するデータ経路決定手段とを備えるシステムを提供する。

【００３９】

添付の図面に示される本発明の実施形態を参照しながら、単なる例示として本発明をさらに説明する。

【図面の簡単な説明】

【００４０】

【図１】本発明の実施形態の技術を使用して生成される相互接続回路を組み込む集積回路を示す略図である。

【図２】相互接続回路内のデータ経路のルーティングを求めるために本発明の一実施形態で使用される方法を示す流れ図である。

【図３】本発明の実施形態での相互接続回路に対して使用されるツリー構造を示す略図である。

【図４】本発明の一実施形態に従って、相互接続回路内の様々なデータ経路を生成するのに、個々のタイル内の配線レイアウトがどのように使用されるかを示す略図である。

【図５】本発明の一実施形態に従って生成される相互接続回路を、左ツリー区間と右ツリー区間を共に含む平衡ツリー構造として示す略図である。

【図６】本発明の一実施形態に従って生成される左ツリー区間を示す略図である。

【図７】本発明の一実施形態に従って生成される右ツリー区間を示す略図である。

【図８】左ツリー区間と右ツリー区間がコーナ素子で分離される、本発明の一実施形態に従って生成される平衡ツリー構造を示す略図である。

【図９】右ツリー区間を生成するために本発明の一実施形態に従って使用される方法を概略的に示す流れ図である。

【図１０】左ツリー区間を生成するために本発明の一実施形態に従って使用される方法を概略的に示す流れ図である。

【図１１】本発明の一実施形態で使用される所定の1組のタイルを示す図である。

【図１２】本発明の一実施形態で使用される所定の1組のタイルを示す図である。

【図１３】本発明の一実施形態で使用される所定の1組のタイルを示す図である。

【図１４】本発明の一実施形態で使用される所定の1組のタイルを示す図である。

【図１５】本発明の一実施形態で使用される所定の1組のタイルを示す図である。

【図１６】本発明の一実施形態で使用される所定の1組のタイルを示す図である。

【図１７】本発明の一実施形態で使用される所定の1組のタイルを示す図である。

【図１８】本発明の一実施形態で使用される所定の1組のタイルを示す図である。
【図１９】本発明の一実施形態で使用される所定の1組のタイルを示す図である。
【図２０】本発明の一実施形態で使用される所定の1組のタイルを示す図である。
【図２１】本発明の一実施形態で使用される所定の1組のタイルを示す図である。
【図２２】本発明の一実施形態で使用される所定の1組のタイルを示す図である。
【図２３】本発明の一実施形態で使用される所定の1組のタイルを示す図である。
【図２４】本発明の一実施形態で使用される所定の1組のタイルを示す図である。
【図２５】本発明の一実施形態で使用される所定の1組のタイルを示す図である。
【図２６】本発明の一実施形態で使用される所定の1組のタイルを示す図である。
【図２７】本発明の一実施形態で使用される所定の1組のタイルを示す図である。
【図２８】本発明の一実施形態で使用される所定の1組のタイルを示す図である。
【図２９】本発明の一実施形態で使用される所定の1組のタイルを示す図である。
【図３０】本発明の一実施形態で使用される所定の1組のタイルを示す図である。
【図３１】本発明の一実施形態で使用される所定の1組のタイルを示す図である。
【図３２】本発明の一実施形態で使用される所定の1組のタイルを示す図である。
【図３３】本発明の一実施形態に従って、異なるタイルが当接によってどのように接続されるかを示す図である。
【図３４】本発明の一実施形態に従って、異なるタイルが当接によってどのように接続されるかを示す図である。
【図３５】本発明の一実施形態に従って、異なるタイルが当接によってどのように接続されるかを示す図である。
【図３６】本発明の一実施形態に従って、異なるタイルが当接によってどのように接続されるかを示す図である。
【図３７】本発明の一実施形態に従って、異なるタイルが当接によってどのように接続されるかを示す図である。
【図３８】本発明の一実施形態に従って、異なるタイルが当接によってどのように接続されるかを示す図である。
【図３９】本発明の一実施形態に従って、異なるタイルが当接によってどのように接続されるかを示す図である。
【図４０】本発明の一実施形態に従って、異なるタイルが当接によってどのように接続されるかを示す図である。
【図４１】本発明の一実施形態に従って、異なるタイルが当接によってどのように接続されるかを示す図である。
【図４２】本発明の一実施形態に従って、異なるタイルが当接によってどのように接続されるかを示す図である。
【図４３】本発明の一実施形態に従って、異なるタイルが当接によってどのように接続されるかを示す図である。
【図４４】本発明の一実施形態に従って、異なるタイルが当接によってどのように接続されるかを示す図である。
【図４５】本発明の一実施形態に従って、異なるタイルが当接によってどのように接続されるかを示す図である。
【図４６】本発明の一実施形態に従って、異なるタイルが当接によってどのように接続されるかを示す図である。
【図４７】本発明の一実施形態に従って、異なるタイルが当接によってどのように接続されるかを示す図である。
【図４８】本発明の一実施形態に従って、異なるタイルが当接によってどのように接続されるかを示す図である。
【図４９】本発明の一実施形態に従って、異なるタイルが当接によってどのように接続されるかを示す図である。
【図５０】本発明の一実施形態に従って、異なるタイルが当接によってどのように接続さ

れるかを示す図である。
【図５１】本発明の一実施形態に従って、異なるタイルが当接によってどのように接続されるかを示す図である。
【図５２】本発明の一実施形態に従って、異なるタイルが当接によってどのように接続されるかを示す図である。
【図５３】本発明の一実施形態に従って、異なるタイルが当接によってどのように接続されるかを示す図である。
【図５４】本発明の一実施形態に従って、異なるタイルが当接によってどのように接続されるかを示す図である。
【図５５】本発明の一実施形態に従って、異なるタイルが当接によってどのように接続されるかを示す図である。
【図５６】本発明の一実施形態に従って、異なるタイルが当接によってどのように接続されるかを示す図である。
【図５７】本発明の一実施形態に従って、異なるタイルが当接によってどのように接続されるかを示す図である。
【図５８】本発明の一実施形態に従って、異なるタイルが当接によってどのように接続されるかを示す図である。
【図５９】本発明の一実施形態に従って、異なるタイルが当接によってどのように接続されるかを示す図である。
【図６０】本発明の一実施形態に従って、異なるタイルが当接によってどのように接続されるかを示す図である。
【図６１】本発明の一実施形態に従って、異なるタイルが当接によってどのように接続されるかを示す図である。
【図６２】本発明の一実施形態に従って、異なるタイルが当接によってどのように接続されるかを示す図である。
【図６３】本発明の一実施形態に従って、異なるタイルが当接によってどのように接続されるかを示す図である。
【図６４】本発明の一実施形態に従って、異なるタイルが当接によってどのように接続されるかを示す図である。
【図６５】本発明の一実施形態に従って、異なるタイルが当接によってどのように接続されるかを示す図である。
【図６６】本発明の一実施形態に従って、異なるタイルが当接によってどのように接続されるかを示す図である。
【図６７】本発明の一実施形態に従って、異なるタイルが当接によってどのように接続されるかを示す図である。
【図６８】本発明の一実施形態に従って、異なるタイルが当接によってどのように接続されるかを示す図である。
【図６９】本発明の一実施形態に従って、異なるタイルが当接によってどのように接続されるかを示す図である。
【図７０】本発明の一実施形態に従って例示的相互接続構造を生成するのに使用されるタイル配置を示す図である。
【図７１】本発明の一実施形態に従って、相互接続回路に対する様々なデバイスの接続のために様々なタイルがどのように使用されるかを示す図である。
【図７２】本発明の一実施形態に従って、相互接続回路に対する様々なデバイスの接続のために様々なタイルがどのように使用されるかを示す図である。
【図７３】本発明の実施形態の技術を実装するコンピュータプログラムを実行することのできる汎用コンピュータのアーキテクチャを示す略図である。
【発明を実施するための形態】
【００４１】
図1は、本発明の実施形態の技術を使用して設計される相互接続回路40を組み込む集積

回路60の例示的実施例である。

【００４２】

図1に示されるように、集積回路60の2つの辺に広がる複数のデバイス/構成要素が、相互接続回路40を介して集積回路内のデバイス20に接続される。デバイス30は、相互接続回路40を介してデバイス20に対する接続を必要とするデバイスを含み、図1では、こうしたデバイスが、IO(入力/出力)デバイスと符号が付けられており、さらにデバイス30は、デバイス20に対する接続を必要としない1つまたは複数のデバイスをも含む。図1では、電源構成要素「VDD」34、「VDDA」35、「VSS」36、および「VSSA」37が、デバイス20に対する接続を必要としないそのような構成要素の例である。デバイスの2つの線形分散経路間の接合に、コーナ素子38も設けることができる。

【００４３】

相互接続回路40は、分散したデバイス30に対してワイドインターフェースを提供するが、デバイス20に対してナローインターフェースを提示する。図1に示されるように、デバイス20が集積回路60内に配置された場合に関する柔軟性を向上させるように、相互接続回路40のナローインターフェースからデバイス20に信号をルーティングするルートチャネル50を設けることができる。あるいは、そのようなルートチャネルを必要とすることなく、相互接続回路のナローインターフェースにデバイス20を直接的に接続することもできる。

【００４４】

後でより詳細に説明するように、相互接続回路を設計するとき、いくつかのセル(図1ではパッドとも呼ばれる)が、相互接続回路のワイドインターフェースに沿って設けられ、次いで様々なデバイス/構成要素30のそれぞれが、そうしたセルのうちの少なくとも1つに関連付けられる。一実施形態では、各デバイス30がセルのうちの1つに関連付けられるが、例えばデバイスが単一のセルに関連付けるには大き過ぎる場合、代替実施形態では、デバイスのうちのいくつかを複数のセルに関連付けることができる。

【００４５】

本発明の一実施形態では、相互接続回路内のデータ経路が、トランク部分およびいくつかのブランチ部分を含むツリー構造を採用するように設計される。ツリー構造は、1つまたは複数のツリー区間を含むことができ、例えば左ツリー区間および/または右ツリー区間を含むことができる。図1の実施例では、ツリー構造は、左ツリー区間と右ツリー区間を共に含み、左ツリー区間は、図1に示される点線の左側であり、右ツリー区間は、点線の右側である。各ツリー区間のトランクが、点線の一方の側のセルと位置合せされ、したがって、左ツリー区間のトランクがVDDセル42と位置合せされ、右ツリー区間のトランクがVSSセル44と位置合せされる。

【００４６】

図1に示される実施形態では、左ツリー区間内と右ツリー区間内に等しい数のセル/パッドがある。残りの図を参照しながらより詳細に論じるように、そのような制約を設計に課すことにより、様々なデータ経路が同一の伝播遅延を有する相互接続回路を生成することが容易となる。しかし、代替実施形態では、左ツリー区間と右ツリー区間に対してそれぞれ異なるタイプのタイルが設計される場合、相互接続回路内のすべてのデータ経路が同一の伝播遅延を有することを保証するように、この制約を取り除くことが可能となる。

【００４７】

図1では、デバイス20がメモリコントローラデバイスであり、かつ相互接続回路のワイドインターフェースに沿って設けられた分散した複数のデバイスが、メモリコントローラデバイスによって制御されるIOデバイス32であると仮定する。しかし、伝播遅延が相互接続を介するデータ経路のすべてについて整合される方式で、相互接続を介してデバイスを分散した複数の他のデバイスに結合する必要のある他の任意の相互接続構造に、本発明の実施形態の技術を適用することもできる。

【００４８】

図2は、相互接続回路40内のデータ経路のルーティングを求めるために本発明の一実施形態で使用される方法の略図である。まず、ステップ100で、相互接続回路のワイドイン

ターフェースに沿って設けるべきセルの数が求められる。セルの数は、相互接続回路を通じてデバイス20に接続しなければならないデバイスの数と、収容する必要のある非接続構成要素の数とを考慮して求められる。本発明の一実施形態では、ツリー構造の任意のトランク部分と位置合せされるセルに関連する非接続構成要素があることが要件である。

【００４９】

ステップ100に続いて、ステップ110で、相互接続を表現するブロックのアレイが形成され、アレイは、ブロックの行および列として形成される。次いで、各セルは、アレイ内の列のうちの1つと当接する。したがって、ステップ100で求めたセルの数が、アレイ内に設ける必要のある列数を決定することがわかる。一実施形態では、行数は、ツリー構造の任意の特定のツリー区間内で、列数よりも1つ少なくなるように選ばれる。したがって、全くの例として、ステップ100で求められた100個のセルがあり、相互接続回路が、左ツリー区間および右ツリー区間を有するツリー構造によって形成されるべき場合、各ツリー区間は、50個の列と49個の行を有することになる。

【００５０】

ステップ120で、アレイ内の各ブロックが考慮され、考慮される各ブロックについて、所定の規則が適用され、そのブロックを表現するのに使用すべき所定の複数のタイルからタイルが決定される。後でより詳細に論じるように、所定の1組のタイルが提供され、各タイルは、その中に所定の配線レイアウトを有する。異なるタイルが、相互接続回路内の異なる幅のセル、したがって異なる幅の列に対して提供される。所定の規則は、アレイ内ブロックの位置、および相互接続回路に接続されている複数のデバイス(例えば、図1のIOセル32)と複数のセルの間の関連付けを考慮に入れる(すなわち、そうした様々なデバイスがどのセルに関連付けられるかを考慮に入れる)。この情報に基づいて、所定の規則は、各ブロック位置について適切なタイルを選ぶことができ、その結果、各ブロックが考慮された後は、デバイス20と、IOデバイス32のそれぞれとの間の必要なデータ経路が、相互接続回路内に設けられ、それによって、そうしたデータ経路のそれぞれは同一の伝播遅延を有する。

【００５１】

したがって、ステップ130で示されるように、ステップ120で各ブロックが考慮された後は、相互接続回路についてのレイアウト設計が出力され、必要な各データ経路が、相互接続されたタイルのシーケンスによって形成され、各データ経路は同一の伝播遅延を有する。

【００５２】

図3は、相互接続回路40に対して使用されるツリー構造をより詳細に示す略図である。図示されるように、任意選択のルートチャネル50を介して相互接続ツリー構造40のナローインターフェース160にデバイス20を結合することができる。この例では、相互接続ツリー構造40が、列165によって形成されるトランクと、トランク165の左側に延びる複数のブランチ170、175、180、185、190とを有する左ツリー区間として形成される。この例では、ブロックのアレイが、5個の行および6個の列として構成され、6個の列は、デバイスを関連付けるべき様々なセルに対するワイドインターフェース150を提供する。したがって、この例では、ワイドインターフェースに沿って6個のセルがあり、相互接続に接続すべきデバイス、または相互接続に接続すべきではないデバイス/構成要素が各セルに関連付けられる。

【００５３】

図4は、相互接続ツリー構造40の同じ編成を示すが、タイルが各ブロックに割り振られた後にツリー構造内に生成することのできるデータ経路の略図を示す。前と同様に、任意選択のルートチャネル50が設けられ、デバイス20が相互接続40のナローインターフェースに接続される。図4から理解するであろうが、デバイス20とセル1から5の間に設けられたデータ経路は、それぞれ同一の長さであり、したがって同一の伝播遅延を有する。先に論じたように、相互接続への接続を必要とする任意のデバイスがツリー構造のトランクに当接する位置に配置されないという前述の制約を仮定すると、セル0 200に関連する任意の

デバイスを、相互接続40に対する接続を必要としないデバイスとなるように選ぶ必要がある。他のセル205、210、215、220、225に関して、これらはすべて、これらに関連する接続デバイスを有する可能性があり、またはこれらのうちの1つまたは複数が、これらに関連する非接続デバイスを有する可能性がある。

【００５４】

図5は、本発明の一実施形態を使用して生成される相互接続回路構成の一例である。NC(非接続)と符号が付けられた10個のセル260は、相互接続回路を介するデバイス20への接続を必要としない構成要素/デバイスに関連付けるべきセルである。この例は、この相互接続の主なサブコンポーネントを示す。

・ルート50-これは、デバイス20をツリー構造40に接続する。ルートはタイル10および11を使用して構築される。

・ツリー構造40-これは、ルート50を個々のIOセル260に接続する。ツリー構造40は、タイル0L、0R、1L、1R、2L、2R、3L、3R、4L、4R、5L、5R、6L、6R、7L、7R、8L、および8Rを使用して構築される。これが相互接続回路の本体を構成する。

【００５５】

ツリー自体は、2つの区間に分割される。左ツリー区間が、0L、0R、1L、2L、3L、4L、5L、6L、7L、および8Lタイルを使用して構築される。図5では、左ツリー区間は、図の中央の列C=0から、左末端の列C=12までに及ぶ。この区間は、列番号が与えられないコーナ素子250を含む。コーナ素子は、0Lおよび0Rタイルと共に構築される。それぞれの0Lおよび0Rタイルは、その行番号で索引付けされる。図5では、索引が角括弧内に0L[R]および0R[R]として示されており、角括弧内のRは、タイルに関連する行である。

【００５６】

右ツリー区間が、0L、0R、1R、2R、3R、4R、5R、6R、7R、および8Rタイルを使用して構築される。図5では、右ツリー区間は、図の中央の列C=0から、右末端の列C=12にまで及ぶ。図5のこの区間はコーナを含まないが、望むなら、コーナを右ツリー区間のレイアウトに組み込むことができる。

【００５７】

左ツリー区間および右ツリー区間はさらに、2つの部分、すなわちトランクとブランチに分割される。左ツリーブランチおよび右ツリーブランチは、0L、0R、1L、1R、2L、2R、3L、3R、4L、4R、5L、および5Rタイルを使用して構築される。これらは、値C>0を有する列内に含まれる。ツリーブランチは、ツリーの行Rと同じ番号を有する。行が、左ツリー区間と右ツリー区間についてそれぞれ、図の左側下部と右側に示されている。

【００５８】

ツリートランクは、6L、6R、7L、7R、8L、および8Rタイルを使用して構築される。図5では、ツリートランクは、列C=0内のタイルで作成される。

【００５９】

Xと符号が付けられたタイルは「don't care」タイルであり、相互接続回路を生成するときに「don't care」タイルを省略することができ、またはその代わりに、技術によるメタルフィル密度要件を満たすようにメタルフィルセルを含むタイルを使用することができる。

【００６０】

主要なタイプのツリー構成が3つある。左分岐ツリーは左ツリー区間のみを含み、右ツリー区間が省略される。左分岐ツリーはコーナ素子を含む。図6は、そのトランクのナローインターフェースでデバイス20に直接的に接続される左分岐ツリー270の一例を示す。この場合も、IOセル260が各列に関連付けられるように示されており、セルの一部が、相互接続回路のデータ経路に接続すべきデバイスに関連付けられており(IOLセル)、他のセルが、非接続デバイスに関連付けられている(NC IOLセル)。図6の例のこの左分岐ツリーは、コーナ素子またはルートを有さない。

【００６１】

右分岐ツリーは右ツリー区間を含み、左ツリー区間が省略される。左分岐ツリーの場合

と同じく、右分岐ツリーはコーナ素子を含むことができる。図7は、コーナ素子またはルートを有さない右分岐ツリー290の一例を示す。やはり、この例では、右分岐ツリー290が、そのトランクのナローインターフェースでデバイス20に直接的に接続され、セル260が各列に関連付けられるように示されている。

【００６２】

平衡ツリーは、例えば図5を参照しながら先に示したように、左ツリー区間と右ツリー区間のどちらも含む。平衡ツリーは、1つが左ツリー区間上、1つが右ツリー区間上の2つのコーナ素子を有することができる。

【００６３】

アングルド平衡ツリーは、左ツリー区間と右ツリー区間の間にコーナ素子を有する、平衡ツリーの特殊なケースである。図8は、アングルド平衡ツリーの一例を示す。アングルド平衡ツリーは、3つのコーナ素子、すなわち左ツリー区間上のコーナ素子、右ツリー区間上のコーナ素子、ならびに左ツリー区間と右ツリー区間の間のコーナ素子を有することができる。図8の例では、アングルド平衡ツリーは、コーナ280によって右ツリー区間290から分離される左ツリー区間270を有する。デバイス20は、タイル9、10、および11で形成されるルート50によってツリーから分離される。先の図5から7の場合と同じく、セル260は、接続されるデバイスまたは非接続デバイスとの関連付けのために、ツリーの各列に関連付けられるように示されている。

【００６４】

上記の例では、任意のツリー区間は、その中に多くとも1つのコーナ素子を有することができることが示されているが、代替実施形態では、各ツリー区間内に複数のコーナ素子を設けることができることを理解されよう。

【００６５】

図9は、右分岐ツリーを構築する一実施形態で使用されるアルゴリズムの流れ図を示す。このアルゴリズムは、平衡ツリーの右ツリー区間を構築するのにも使用される。図10は、左分岐ツリーを構築する一実施形態で使用されるアルゴリズムの流れ図を示す。このアルゴリズムは、平衡ツリーの左ツリー区間を構築するのにも使用される。どちらのアルゴリズムも、以下のパラメータおよび変数を使用する。

N-その値が相互接続回路を介して部分接続されるIOセルの数を表す、所与の構成に対する定数。

CCL-コーナセルがその後ろで具体化されるIOセル列の位置をその値が表す、所与の構成に対する定数。

c-タイルを具体化するために選択された列位置を含む変数。

r-タイルを具体化するために選択された行位置を含む変数。

【００６６】

図9および10では、タイルはGDSタイルと呼ばれる。アルゴリズムは、行rおよび列cの座標を使用することによってタイルのタイプおよびツリー内のタイルの位置を指定することにより、ツリーを構築する。図9および10に示されるアルゴリズムは、「don't care」タイルが使用される場合、「don't care」タイルを具体化しない。図6に示されるように、左分岐ツリーが右から左に構築される。このことは、列c=0が右端の列であり、列c=N-1が左端の列であることを意味する。図7に示されるように、右分岐ツリーが左から右に構築される。このことは、列c=0が左端の列であり、列c=N-1が右端の列であることを意味する。図5に示されるように左分岐ツリーと右分岐ツリーを列c=0で当接させ、または図8に示されるように90°の角度で当接させて共に配置することによって平衡ツリーが構築される。図9および10に示されるアルゴリズムを使用して相互接続回路用に生成されるツリー構造は、以下の制約を有する。

1.すべてのタイル、IOセル、およびデバイス20が、当接によって互いに接続される。

2.素子を当接するとき、相互接続バス(データ経路)に関するビット位置を維持しなければならない。

3.伝播遅延が整合されることを保証するために、左ツリー区間および右ツリー区間に対

する平衡ツリーを構築するときの列数が同一でなければならない。
4.列c=0に関連するIOセルを、相互接続を通じてデバイス20に対する接続を必要とするデバイスに関連付けなければならない。
5.コーナが、列番号が与えられないが、(c,r)グリッド位置を有するタイルの後に配置される0Lおよび0Rタイルと共に構築される。

【００６７】

図9および10で定義されるプロセスに続いて、右ツリー区間および左ツリー区間を、先の図で示されるように形成することができる。例えば、図9のプロセスを適用することにより、図7に示される右ツリー区間を生成することができる。この例では、N=11である。位置r=0、c=0で開始すると、プロセスがステップ300、305、310、320、325、330、405を通じてステップ415に進み、このツリー区間上にコーナがないと判定されることがわかる。したがって、プロセスはステップ420に進み、タイル6Rが行0、列0に挿入され、その後でプロセスはステップ400に進む。ステップ400で、列値cが増分され、その後でプロセスは、ステップ325、330、335、350、360を介してステップ370に進む。この時点で、ステップ370で指定される条件が満たされ、プロセスはステップ375に分岐し、タイル1Rが列1、行0に配置され、その後でプロセスはステップ400に戻る。

【００６８】

ステップ400で、値Nに等しい値まで列値が増分されると、プロセスはステップ450に分岐し、行値が増分され、その後で、ステップ320で列値が0にリセットされ、行1のブロック位置のすべてについてプロセスが反復される。ステップ310で、増分された行値がもはやN-1未満ではないと判定されたとき、プロセスはステップ315で終了する。

【００６９】

図10は、左ツリー区間を生成するのに使用される、対応するプロセスを示す。したがって、例えば、図6の左ツリー区間を考慮すると、図6に示されるこのタイル割振りが、図10のプロセスを実施することによって生成される。全くの例として、ブロック位置r=5、c=5を考慮すると、ステップ525から、プロセスは、地点ポイント560まで経路530、535、550をたどる。この地点で、r=5、c=5、およびN=11であるので、ステップ560で指定された条件が満たされる。したがって、プロセスはステップ565に分岐し、図6に明確に示されているように、タイル2Lがグリッド位置に配置される。

【００７０】

図5から8のいずれかに関連して図9および10を同様に検討にすることにより、こうした図に記載のプロセスが図5から8に示されるタイルレイアウトを生成することがわかる。こうしたタイルのそれぞれで設けられる配線レイアウトのために、図9および10のプロセスは、すべて同一の伝播遅延を有する相互接続構造内のデータ経路を生成する。さらに、図9および10から理解できるであろうが、この設計レイアウトプロセスは、タイミング制約を満たすかどうかを確認するための、生成されたレイアウトの評価を必要とせず、したがって、従来技術のEDAツールで一般に必要な反復のいずれも必要としない。

【００７１】

ルート構造は、あらゆる相互接続タイプに対して不要である。デバイス20は、図6および7に示されるように、ツリートランクに直接的に当接することができる。一般には、ルートは、タイル9、10、および11から構築される。左分岐ツリー、右分岐ツリー、および平衡ツリーは、タイル9、10、および11を使用して、ルートをツリーに接続することができる。アングルド平衡ツリーは、タイル9を使用して、ルートをツリーに接続する。図8に示されるように、デバイス20と当接する場所までルートを拡張するのにタイル10および11が使用される。

【００７２】

相互接続ツリー構造が、一緒に配置されたときに当接を通じて接続される1組のタイルから構築される。こうしたタイルを記述するいくつかの共通パラメータが存在する。
N-IO数。このパラメータは、IOリング260の一部となるIOセルの合計数を列挙する。
I-接続されるIO数。このパラメータは、相互接続を通じてデバイス20への接続を必要と

するIOセル数を指定する。
P-ポート幅。このパラメータは、IOセルが相互接続を通じてデバイス20に伝播する必要がある最大接続数または最大ビット数を指定する。
Z-最大必要水平金属長。このパラメータは、以下のうちの最長のものを指定する。
1.任意のタイルのイーストポートをウェストポートに接続するのに必要な最長のワイヤ。
2.任意の0Rまたは0Lタイルのイーストポートをウェストポートに接続するのに必要な最長のワイヤの2倍。
V-最大必要垂直金属長。このパラメータは、任意のタイルのノースポートをサウスポートに接続するための最長のワイヤを指定する。
H-タイル高さ。このパラメータは、任意のタイルのイースト側またはウェスト側の幅Pビットの2つのポートを支持するのに必要な最小の高さを指定する。

【００７３】

図11は、例示のためのタイル「0L」の一例である。大部分の実装は、複数の金属層およびバイアを使用して、ポートおよび相互接続配線を作成する。
・タイルは高さHを有する。
・タイル幅は、その行位置に応じて変動する。
・すべてのポート(2、3、4、および5)は、それぞれP個の接続(ビット)を有する。
・相互接続配線6および7は、長さMを有する。ただし
M=0.5*Z

【００７４】

図12は、例示のためのタイル「0R」の一例である。大部分の実装は、複数の金属層およびバイアを使用して、ポートおよび相互接続配線を作成する。
・タイルは高さHを有する。
・タイル幅は、その行位置およびコーナ素子のサイズに応じて変動する。
・すべてのポート(2、3、4、および5)は、それぞれP個の接続(ビット)を有する。
・相互接続配線6および7は、長さMを有する。ただし
M=0.5*Z

【００７５】

左側と右側のタイルが等しい幅を有する場合、水平フリップを使用することにより、所与の行rについてのタイプ0L[r]および0R[r]に対して同一のタイル配線設計を使用することができる。

【００７６】

図13は、例示のためのタイル「1L」の一例である。大部分の実装は、複数の金属層およびバイアを使用して、ポートおよび相互接続配線を作成する。
・タイルは高さHを有する。
・デバイス20への接続を有さないIOセルを含む相互接続回路内で使用される各IOセル幅について、1つのタイルが構築される。
・すべてのポート(2、3、4、5、6、および7)は、それぞれP個の接続(ビット)を有する。
・相互接続配線8および9は、長さMを有する。ただし
M=Z
・相互接続配線10は長さVを有する。

【００７７】

図14は、例示のためのタイル「1R」の一例である。大部分の実装は、複数の金属層およびバイアを使用して、ポートおよび相互接続配線を作成する。
・タイルは高さHを有する。
・相互接続回路内で使用される各IOセル幅について、1つのタイルが構築される。
・すべてのポート(2、3、4、5、6、および7)は、それぞれP個の接続(ビット)を有する。

・相互接続配線8および9は、長さMを有する。ただし
M=Z
・相互接続配線10は長さVを有する。

【００７８】

ノース側とサウス側のポートが垂直対称性を有する場合、水平フリップを使用することにより、タイプ1Lおよび1Rに対して同一のタイル配線設計を使用することができる。

【００７９】

図15は、例示のためのタイル「2L」の一例である。大部分の実装は、複数の金属層およびバイアを使用して、ポートおよび相互接続配線を作成する。

・タイルは高さHを有する。
・相互接続回路内で使用される各IOセル幅について、1つのタイルが構築される。
・すべてのポート(2、3、4、および5)は、それぞれP個の接続(ビット)を有する。
・相互接続配線6は長さMを有する。ただし
M=0.5*Z+0.5*V
・相互接続配線7は長さMを有する。ただし
M=Z

【００８０】

図16は、例示のためのタイル「2R」の一例である。大部分の実装は、複数の金属層およびバイアを使用して、ポートおよび相互接続配線を作成する。

・タイルは高さHを有する。
・相互接続回路内で使用される各IOセル幅について、1つのタイルが構築される。
・すべてのポート(2、3、4、および5)は、それぞれP個の接続(ビット)を有する。
・相互接続配線6は長さMを有する。ただし
M=0.5*Z+0.5*V
・相互接続配線7は長さMを有する。ただし
M=Z

【００８１】

ノース側のポートが垂直対称性を有する場合、水平フリップを使用することにより、タイプ2Lおよび2Rに対して同一のタイル配線設計を使用することができる。

【００８２】

図17は、例示のためのタイル「3L」の一例である。大部分の実装は、複数の金属層およびバイアを使用して、ポートおよび相互接続配線を作成する。

・タイルは高さHを有する。
・相互接続回路内で使用される各IOセル幅について、1つのタイルが構築される。
・すべてのポート(2および3)は、それぞれP個の接続(ビット)を有する。
・相互接続配線4は長さMを有する。ただし
M=2*Z+0.5*V

【００８３】

図18は、例示のためのタイル「3R」の一例である。大部分の実装は、複数の金属層およびバイアを使用して、ポートおよび相互接続配線を作成する。

・タイルは高さHを有する。
・相互接続回路内で使用される各IOセル幅について、1つのタイルが構築される。
・すべてのポート(2および3)は、それぞれP個の接続(ビット)を有する。
・相互接続配線4は長さMを有する。ただし
M=2*Z+0.5*V

【００８４】

水平フリップを使用することにより、タイプ3Lおよび3Rに対して同一のタイル配線設計を使用することができる。

【００８５】

図19は、例示のためのタイル「4L」の一例である。大部分の実装は、複数の金属層およ

びバイアを使用して、ポートおよび相互接続配線を作成する。
・タイルは高さHを有する。
・相互接続回路内で使用される各IOセル幅について、1つのタイルが構築される。
・すべてのポート(2および3)は、それぞれP個の接続(ビット)を有する。
・相互接続配線4は長さMを有する。ただし
M=Z+0.5*V

【００８６】

図20は、例示のためのタイル「4R」の一例である。大部分の実装は、複数の金属層およびバイアを使用して、ポートおよび相互接続配線を作成する。
・タイルは高さHを有する
・相互接続回路内で使用される各IOセル幅について、1つのタイルが構築される。
・すべてのポート(2および3)は、それぞれP個の接続(ビット)を有する。
・相互接続配線4は長さMを有する。ただし、
M=Z+0.5*V

【００８７】

水平フリップを使用することにより、タイプ4Lおよび4Rに対して同一のタイル配線設計を使用することができる。

【００８８】

図21は、例示のためのタイル「5L」の一例である。大部分の実装は、複数の金属層およびバイアを使用して、ポートおよび相互接続配線を作成する。
・タイルは高さHを有する。
・相互接続回路内で使用される各IOセル幅について、1つのタイルが構築される。
・すべてのポート(2および3)は、それぞれP個の接続(ビット)を有する。
・相互接続配線4は長さMを有する。ただし
M=1.5Z+0.5*V

【００８９】

図22は、例示のためのタイル「5R」の一例である。大部分の実装は、複数の金属層およびバイアを使用して、ポートおよび相互接続配線を作成する。
・タイルは高さHを有する。
・相互接続回路内で使用される各IOセル幅について、1つのタイルが構築される。
・すべてのポート(2および3)は、それぞれP個の接続(ビット)を有する。
・相互接続配線4は長さMを有する。ただし
M=1.5Z+0.5*V

【００９０】

ノース側のポートが垂直対称性を有する場合、水平フリップを使用することにより、タイプ5Lおよび5Rに対して同一のタイル配線設計を使用することができる。

【００９１】

図23は、例示のためのタイル「6L」の一例である。大部分の実装は、複数の金属層およびバイアを使用して、ポートおよび相互接続配線を作成する。
・タイルは高さHを有する。
・タイル幅は、ツリートランク上のIOセル幅と同一である。
・ポート2および3は、S個の接続を有する。ただし
S≧(I-1)*P
・ポート4および5は、S個の接続を有する。ただし
S=P
・相互接続配線6は長さMを有する。ただし
M=V
・相互接続配線7は長さMを有する。ただし
M=Z+0.5*V

【００９２】

図24は、例示のためのタイル「6R」の一例である。大部分の実装は、複数の金属層およびバイアを使用して、ポートおよび相互接続配線を作成する。

・タイルは高さHを有する。
・タイル幅は、ツリートランク上のIOセル幅と同一である。
・ポート2および3は、S個の接続を有する。ただし
S≧(I-1)*P
・ポート4および5は、S個の接続を有する。ただし
S=P
・相互接続配線6は長さMを有する。ただし
M=V
・相互接続配線7は長さMを有する。ただし
M=Z+0.5*V

【００９３】

水平フリップを使用することにより、タイプ6Lおよび6Rに対して同一のタイル配線設計を使用することができる。

【００９４】

図25は、例示のためのタイル「7L」の一例である。大部分の実装は、複数の金属層およびバイアを使用して、ポートおよび相互接続配線を作成する。

・タイルは高さHを有する。
・タイル幅は、ツリートランク上のIOセル幅と同一である。
・ポート2および3は、S個の接続を有する。ただし
S≧I*P
・相互接続配線4は長さMを有する。ただし
M=V

【００９５】

図26は、例示のためのタイル「7R」の一例である。大部分の実装は、複数の金属層およびバイアを使用して、ポートおよび相互接続配線を作成する。

・タイルは高さHを有する。
・タイル幅は、ツリートランク上のIOセル幅と同一である
・ポート2および3は、S個の接続を有する。ただし
S≧I*P
・相互接続配線4は長さMを有する。ただし
M=V

【００９６】

水平フリップを使用することにより、タイプ7Lおよび7Rに対して同一のタイル配線設計を使用することができる。

【００９７】

図27は、例示のためのタイル「8L」の一例である。大部分の実装は、複数の金属層およびバイアを使用して、ポートおよび相互接続配線を作成する。

・タイルは高さHを有する。
・タイル幅は、ツリートランク上のIOセル幅と同一である。
・ポート2および3は、S個の接続を有する。ただし
S≧(I-1)*P
・ポート4および5は、S個の接続を有する。ただし
S=P
・相互接続配線6は長さMを有する。ただし
M=V
・相互接続配線7は長さMを有する。ただし
M=2*Z+0.5*V

【００９８】

図28は、例示のためのタイル「8R」の一例である。大部分の実装は、複数の金属層およびバイアを使用して、ポートおよび相互接続配線を作成する。

・タイルは高さHを有する。
・タイル幅は、ツリートランク上のIOセル幅と同一である。
・ポート2および3は、S個の接続を有する。ただし
S≧(I-1)*P
・ポート4および5は、S個の接続を有する。ただし
S=P
・相互接続配線6は長さMを有する。ただし
M=V
・相互接続配線7は長さMを有する。ただし
M=2*Z+0.5*V

【００９９】

水平フリップを使用することにより、タイプ8Lおよび8Rに対して同一のタイル配線設計を使用することができる。

【０１００】

図29は、例示のためのタイル「9L」の一例である。大部分の実装は、複数の金属層およびバイアを使用して、ポートおよび相互接続配線を作成する。

・タイルは高さTを有する。ただし
Xがツリートランク上のIOセル幅である場合、
T=2*X
・タイルは幅Uを有する。ただし
Xがツリートランク上のIOセル幅である場合、
U=X
・ポート2、3、4、および5は、S個の接続を有する。ただし
S≧(I-1)*P
・相互接続配線6および7は、長さMを有する。ただしMは任意の値でよく、好ましくはタイルに対して可能な最小の長さである。

【０１０１】

図30は、例示のためのタイル「9R」の一例である。大部分の実装は、複数の金属層およびバイアを使用して、ポートおよび相互接続配線を作成する。

・タイルは高さTを有する。ただし
Xがツリートランク上のIOセル幅である場合、
T=2*X
・タイルは幅Uを有する。ただし
Xがツリートランク上のIOセル幅である場合、
U=X
・ポート2、3、4、および5は、S個の接続を有する。ただし
S≧(I-1)*P
・相互接続配線6および7は、長さMを有する。ただしMは任意の値でよく、好ましくはタイルに対して可能な最小の長さである。

【０１０２】

水平フリップを使用することにより、タイプ9Lおよび9Rに対して同一のタイル配線設計を使用することができる。

【０１０３】

図31は、例示のためのタイル「10」の一例である。大部分の実装は、複数の金属層およびバイアを使用して、ポートおよび相互接続配線を作成する。

・タイルは高さTを有する。ただし
Xがツリートランク上のIOセル幅である場合、
T=2*X

・タイルは幅Uを有する。ただし
Xがツリートランク上のIOセル幅である場合、
U=2*X
・ポート2、3、4、および5は、S個の接続を有する。ただし
S≧(I-1)*P
・相互接続配線6および7は、長さMを有する。ただしMは任意の値でよく、好ましくはタイルに対して可能な最小の長さである。

【０１０４】

応用例に適合するようにタイル10を回転し、裏返すことができる。

【０１０５】

図32は、例示のためのタイル「11」の一例である。大部分の実装は、複数の金属層およびバイアを使用して、ポートおよび相互接続配線を作成する。
・タイルは高さTを有する。ただし
Xがツリートランク上のIOセル幅である場合、
T=2*X
・タイル幅は任意の長さである。幅が短くなるほど、相互接続ルートは、ASIC内にデバイス20をより正確に配置することができる。タイル10の幅を短くすることには、デバイス20を相互接続ツリーから離して配置しなければならないときに、タイル10の多くを使用しなければならないという犠牲が伴う。配置精度をタイル具体化数の削減と組み合わせるために、複数の幅のタイル10を構築することができる。
・ポート2、3、4、および5は、S個の接続を有する。ただし
S≧(I-1)*P
・相互接続配線6および7は、長さMを有する。ただしMは任意の値でよく、好ましくはタイルに対して可能な最小の長さである。

【０１０６】

応用例に適合するようにタイル10を回転し、裏返すことができる。

【０１０７】

すべてのタイル、IOセル、およびデバイス20は、当接によって接続される。すべてのバスビット順序は、当接するときに常に維持される。図33から69は、相異なるタイルがどのように当接するかについての様々な例を示す。図のリストは網羅的なものではなく、レイアウト内でタイルがどのように共に配置されるかについての広範な概念的例である。

【０１０８】

図70は、左ツリー区間および右ツリー区間を有し、各ツリー区間がそれに関連付けられた5個のセルを有する平衡ツリーの比較的単純な例について生成された配線レイアウトを示す。典型的な実施形態では、ワイドインターフェースに沿ってさらに多くのセルが設けられることになり、図70は、例示のために与えられるものに過ぎない。図70からわかるように、様々なタイル内の個々の配線レイアウトが、相互接続を通るデータ経路を提供し、相互接続に接続すべき各IOデバイス32に接続点を提供する。図70の例に示されるように、左ツリー区間内にそのようなIOデバイス32が3個あり、右ツリー区間内にそのようなIOデバイス32が4個ある。個々のタイル内の配線レイアウトを使用して、接続されたIOデバイス32への接続を実現し、または非IOデバイスへの接続を回避することのできる方式が、図71および72に略図で示されている。図71に示されるように、接続する必要のあるIOデバイス32上のピン位置は、隣接するタイル上の必要な配線レイアウトと接続するようなものとなり、接続を必要としないIOデバイスは、そのピンと隣接するタイルの配線レイアウトとが位置合せされない。

【０１０９】

同様に、図72では、6Lタイルと6Rタイルの組合せを使用して、デバイス20に対する必要な接続をどのように提供するかを示す。

【０１１０】

図73は、上述の技術を実装するのに使用することのできるタイプの汎用コンピュータ70

0の略図を示す。汎用コンピュータ700は、中央演算処理装置702、ランダムアクセスメモリ704、読取り専用メモリ706、ネットワークインターフェースカード708、ハードディスクドライブ710、ディスプレイドライバ712およびモニタ714、ならびにすべて共通バス722を介して接続されるキーボード718およびマウス720を伴うユーザ入力/出力回路716を含む。動作の際に、中央演算処理装置702は、ランダムアクセスメモリ704、読取り専用メモリ706、およびハードディスクドライブ710のうちの1つまたは複数に格納することができ、またはネットワークインターフェースカード708を介して動的にダウンロードすることのできるコンピュータプログラム命令を実行する。ディスプレイドライバ712およびモニタ714を介して、実行された処理の結果をユーザに表示することができる。キーボード718またはマウス720からユーザ入力/出力回路716を介して、汎用コンピュータ700の動作を制御するユーザ入力を受け取ることができる。コンピュータプログラムを様々なコンピュータ言語で書くことができることを理解されよう。コンピュータプログラムを記録媒体上に格納および配布することができ、または汎用コンピュータ700に動的にダウンロードすることができる。適切なコンピュータプログラムの制御下で動作するとき、汎用コンピュータ700は、相互接続回路内のデータ経路のルーティングを求める上述の技術を実施することができ、汎用コンピュータ700が上述の技術を実施する装置を形成するとみなすことができる。汎用コンピュータ700のアーキテクチャはかなり変動する可能性があり、図73は一例に過ぎない。

【０１１１】

本発明の上述の実施形態では、配線レイアウトが不要である任意のブロック位置について、タイルが具体化されない(例えば、図5から8の「x」とマークされたタイルを参照)。しかし、代替実施形態では、所定のタイルの組は、そうしたブロックについてのタイルを具体化するのに使用されるフィラータイルを含むことができる。一実施形態では、こうしたフィラータイルは、所与の製造工程に対する金属密度要件を満たすようにメタルフィルを有する。

【０１１２】

さらに、ある実施形態では、IOセルライブラリが、レイアウトが密集することなく(トランクに関して)コーナセルの反対側に設けることのできるブランチ数を限定するショートコーナIOセルを有することがある。一実施形態では、このことを、コーナ素子のどちらかの側にフィラー素子を定義し、次いで所定のタイルの組内に提供される所定のフィラータイルを使用して各フィラー素子を実装することによって軽減することができる。コーナタイルと組み合わせたそのようなフィラータイルは、コーナ素子のサイズを実質上拡張し、コーナ素子の反対側で多数のブランチを必要とするツリー構造を構築することが可能となる。

【０１１３】

前述の実施形態では、様々なタイルが、あらゆる可能なセル幅について別々の具体化を有する。しかし、代替実施形態では、フィラーセルの使用により、あらゆる可能なセル幅についてタイルを設ける必要を回避することができる。より具体的には、各IOセルについて同じ幅を有する組合せIOセルライブラリを作成するために、フィラーセルを既存のIOセルと組み合わせることができる。これにより、相互接続回路を構築するために所定のタイルの組内に提供する必要のあるタイル幅の数が著しく削減される。

【０１１４】

前述の実施形態では、相互接続回路のツリー構造に接続すべき様々なデバイスが、相互接続回路の関連する列内の隣接するタイルに直接的に接続されるように構成される。しかし、代替実施形態では、こうしたデバイスのピンと、所定のタイルの組の中のタイルの配線レイアウトとの間のピンマッピングを実現する1つまたは複数のブリッジタイルを設けることができる。したがって、こうしたブリッジタイルは、IOデバイスを相互接続ツリーに当接するのに必要なピン位置に適合するように、IOデバイスのピン位置を変換する。こうしたセルはまた、相互接続回路を介する接続を必要としないIOデバイス/構成要素との偶発的当接を回避するように、相互接続回路をデバイスから分離する役割も果たすことが

できる。こうしたブリッジタイルを使用して、水平フリップによる左側ツリーと右側ツリーについてのタイル配線設計の再利用を可能にする垂直対称性を有するポートを有するIOデバイスインターフェースを作成することもできる。

【０１１５】

本発明の実施形態の上記の説明より、そのような実施形態が以下の利点を有することを理解されよう。

1.本発明の実施形態の技術を使用して設計された相互接続回路は、2つの物理ブロックのインターフェース間のサイズ差が非常に大きい場合にその2つの物理ブロックを接続することを可能にする。

2.本発明の実施形態の技術を使用して設計された相互接続回路は、差分信号を効果的に処理することができ、具体的には、本発明の実施形態の対称的ワイヤ長および非励振整合(物理的効果整合)の特性により、差分信号を処理することに特に適している。

3.この技術は、厳密に整合した伝播遅延を実現し、カップリング、漏話、ローディング、雑音遮蔽などの他の物理的側面も整合させることができる。

4.この技術は、様々な幅を有するIOデバイスの混合を扱うことができる。

5.この技術は、コアインターフェースピンおよび信号位置を修正することなく、相互接続を通じて接続されるIOデバイスと、相互接続を通じて接続されないIOデバイス/構成要素との様々な混合を効果的に扱うことができる。

6.前述のルートチャネルの使用により、IOデバイスの配置に関して、どんな場所にもコアインターフェースを配置することができる。

7.この技術は、左コアインターフェース配置、右コアインターフェース配置、および中心コアインターフェース配置に対して、別々の最適化されたトポロジを提供することができる。

8.この技術は、所定の配線レイアウトを有する所定のタイルを使用する構造化ルーティング方法を提供する。

9.少数の所定のレイアウト素子(タイル)のみを用いて多種多様な相互接続を構築することが可能である。

10.本発明の実施形態の技術を使用して構築された相互接続は、厳密な幾何学情報に基づく、事前に特徴付けられたタイミングを有する。

11.この技術はアルゴリズムベースのものであり、フィードフォワード手法に基づいて、様々なバス幅に対して即座にコンパイルすることができる。大部分の周知のASICツールとは異なり、プロセスの反復は不要である。

12.本発明の実施形態の技術を使用して設計された相互接続は通常、従来技術のツールを使用して設計されたものと比較して電力消費が削減される。本技術は、遅延のタイミングおよび整合を改善するためにバッファおよび他の構成要素を様々な経路中に挿入する必要を回避するからである。

【０１１６】

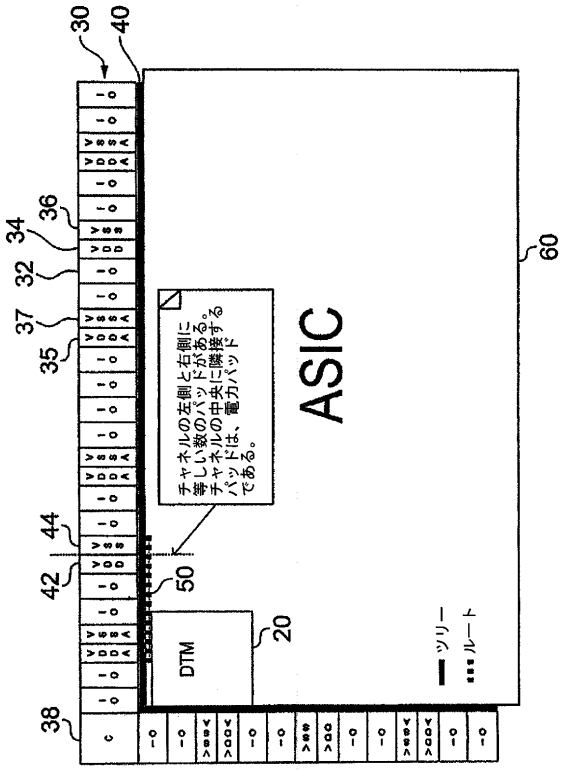
本発明の実施形態の技術を、物理的に離れたブロックにバスを接続するときに高度なタイミング整合を必要とするすべての高速バスに適用することができる。本発明の実施形態の技術は、データ経路の物理的幅のサイズが大きく異なる、局所化設計から集積回路上のIOセルにパラレルバスを接続するのに特に適している。SDR、DDRなどのメモリインターフェース、およびPCIなどのパラレルバスアーキテクチャは、本発明の実施形態の技術を使用することから非常に恩恵を受けることのできる技術の主な例である。こうしたアーキテクチャのデータ転送速度が増加するにつれて、本発明の実施形態の技術は特に有益である。

【０１１７】

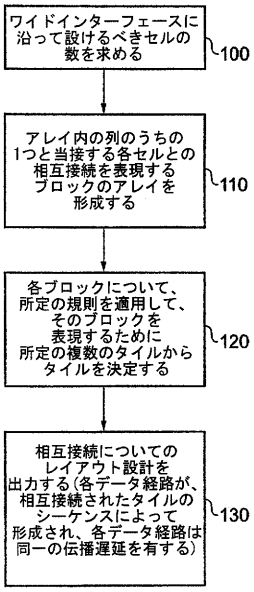
本明細書では本発明の特定の実施形態を説明したが、本発明はその特定の実施形態に限定されず、本発明の範囲内で多数の修正および追加を行えることは明らかであろう。例えば、本発明の範囲から逸脱することなく、以下の従属請求項の特徴の様々な組合せを、独立請求項の特徴と共に作成することができる。

【符号の説明】
【０１１８】
20 デバイス
30 デバイス
32 IOデバイス
34 電源構成要素「VDD」
35 電源構成要素「VDDA」
36 電源構成要素「VSS」
37 電源構成要素「VSSA」
38 電源構成要素コーナ素子
40 相互接続回路
42 VDDセル
44 VSSセル
50 ルートチャネル
60 集積回路
150 ワイドインターフェース
160 ナローインターフェース
165 トランク
170 ブランチ
175 ブランチ
180 ブランチ
185 ブランチ
190 ブランチ
200 セル
205 セル
210 セル
215 セル
220 セル
225 セル
250 コーナ素子
260 IOセル
270 左分岐ツリー
280 コーナ
290 右ツリー区間
700 汎用コンピュータ
702 中央演算処理装置
704 ランダムアクセスメモリ
706 読取り専用メモリ
708 ネットワークインターフェースカード
710 ハードディスクドライブ
712 ディスプレイドライバ
714 モニタ
716 ユーザ入力/出力回路
718 キーボード
720 マウス
722 共通バス

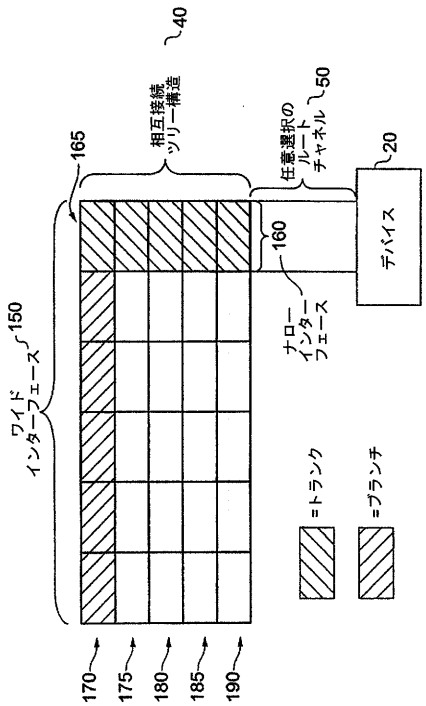
【図１】


30
40
34
36
32
37
35
44
42
38
60
ASIC
チャネルの左側と右側に等しい数のパッドがある。チャネルの中央に隣接するパッドは、電力パッドである。
50
DTM
20
ツリー
ルート

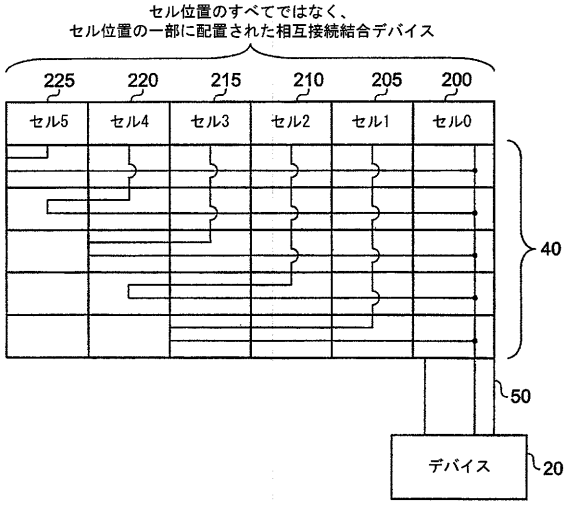
【図２】


ワイドインターフェースに沿って設けるべきセルの数を求める
100
アレイ内の列のうちの1つと当接する各セルとの相互接続を表現するブロックのアレイを形成する
110
各ブロックについて、所定の規則を適用して、そのブロックを表現するために所定の複数のタイルから1つのタイルを決定する
120
相互接続についてのレイアウト設計を出力する（各データ経路が、相互接続されたタイルのシーケンスによって形成され、各データ経路は同一の伝播遅延を有する）
130

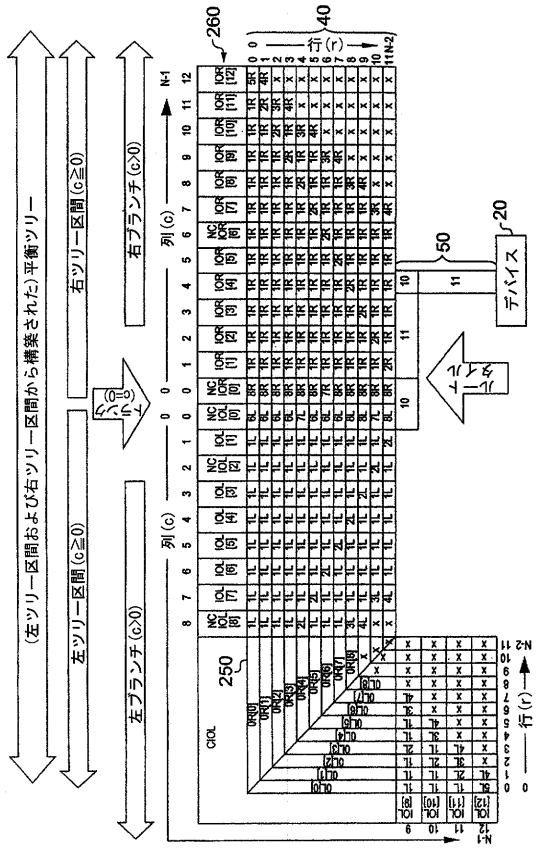
【図３】


40
相互接続ツリー構造
165
任意選択のルートチャネル
50
20
デバイス
160
ナローインターフェース
ワイドインターフェース
150
=トランク
=ブランチ
170
175
180
185
190

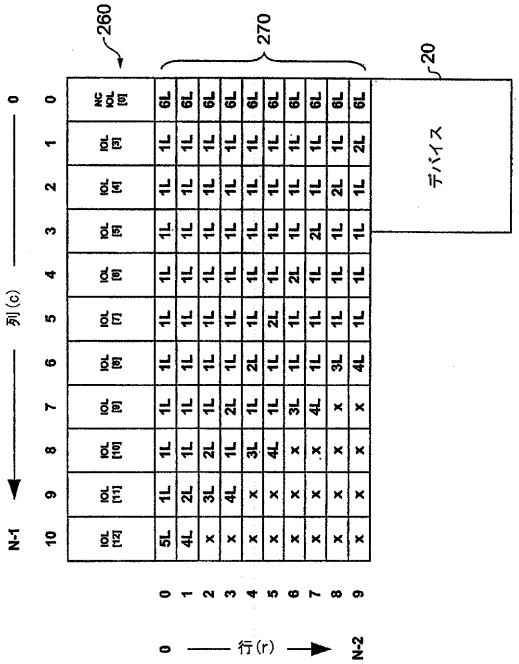
【図４】


セル位置のすべてではなく、
セル位置の一部に配置された相互接続結合デバイス
225
220
215
210
205
200
セル5
セル4
セル3
セル2
セル1
セル0
40
50
デバイス
20

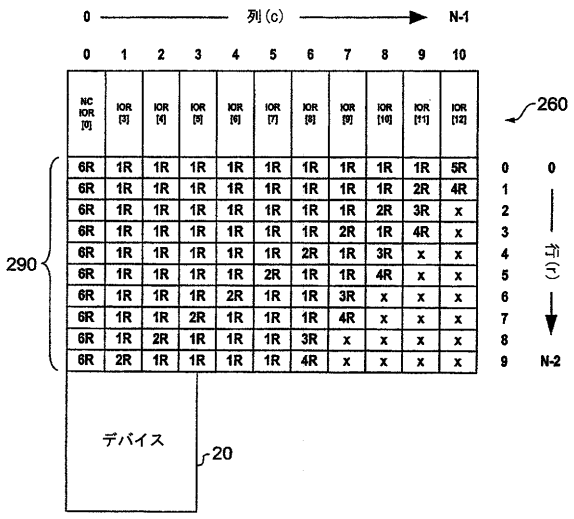
【図５】


(左ツリー区間および右ツリー区間から構築された)平衡ツリー
左ツリー区間(c≧0)
右ツリー区間(c≧0)
左ブランチ(c>0)
右ブランチ(c>0)
ルート
260
40
行(r)
列(c)
50
デバイス
20
250
CIOL

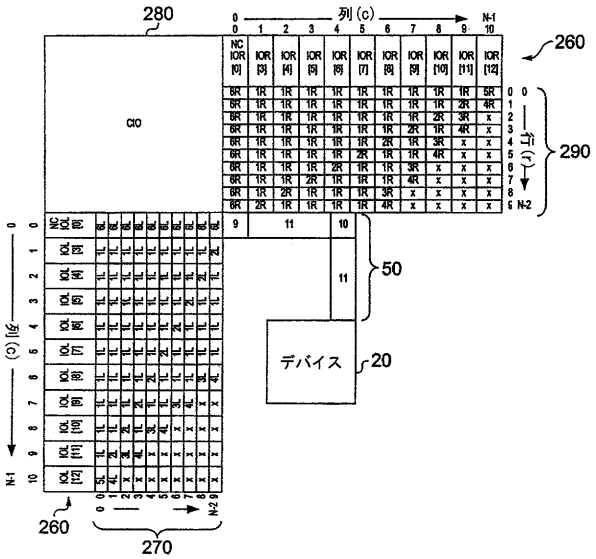
【図６】


260
270
20
デバイス
列(c)
行(r)

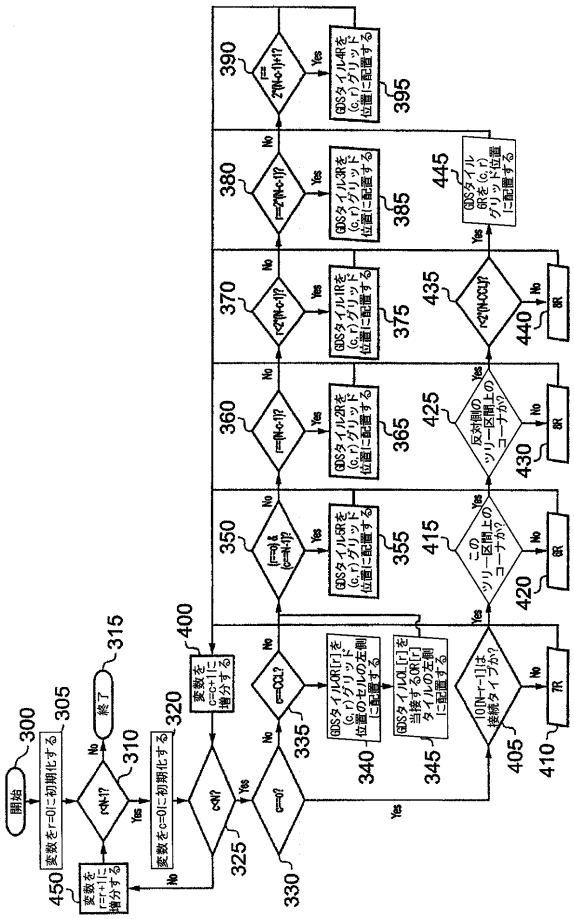
【図７】


列(c)
行(r)
260
290
デバイス
20

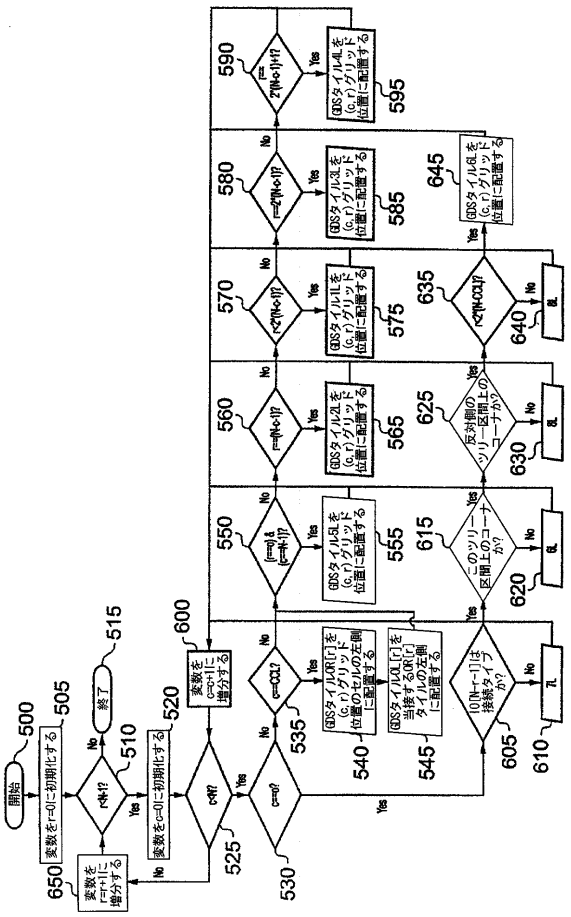
【図８】


280
CIO
260
290
50
デバイス
20
270
列(c)
行(r)

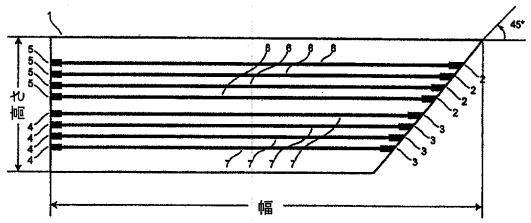
【図９】



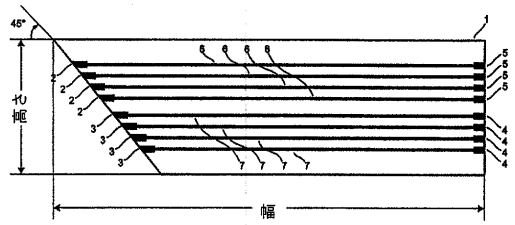
【図１０】



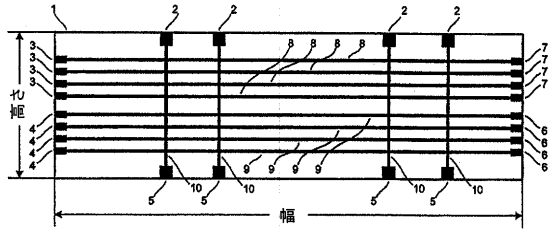
【図１１】



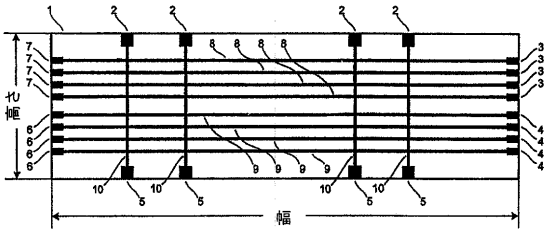
【図１２】



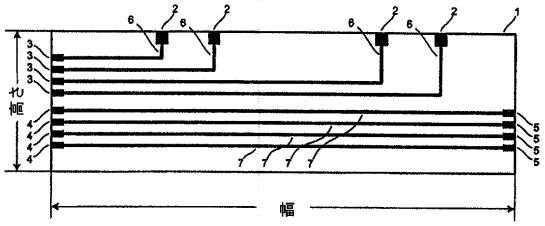
【図１３】



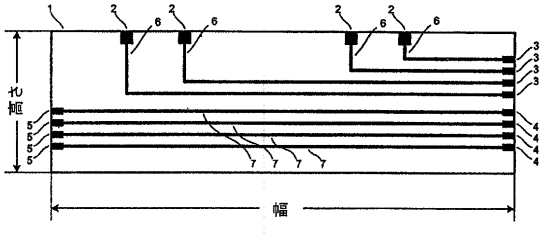
【図１４】



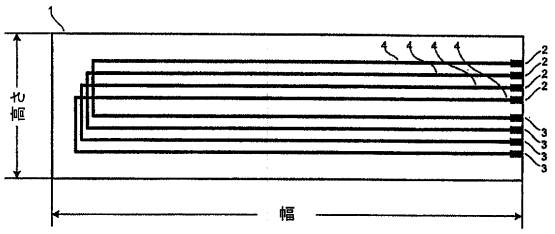
【図１５】



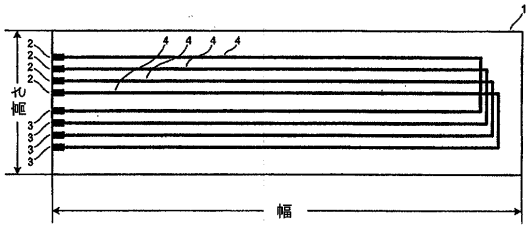
【図１６】



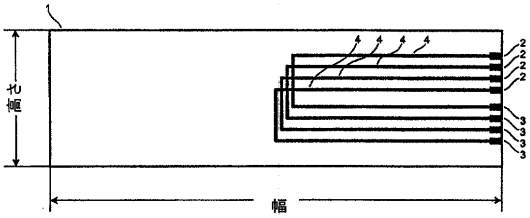
【図１７】



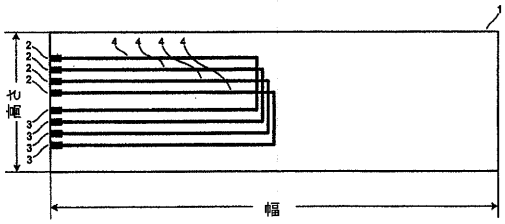
【図１８】



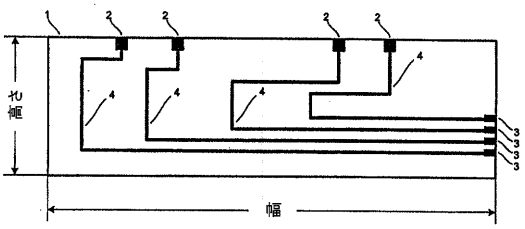
【図１９】



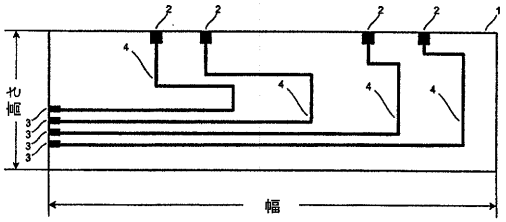
【図２０】



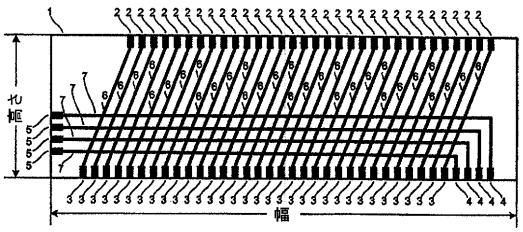
【図２１】



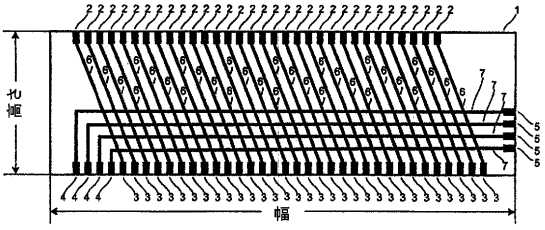
【図２２】



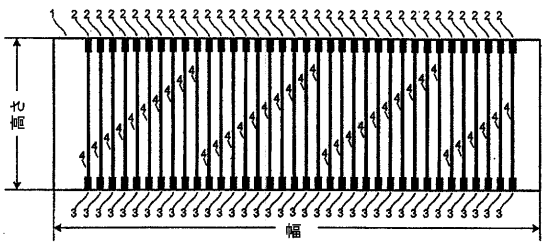
【図２３】



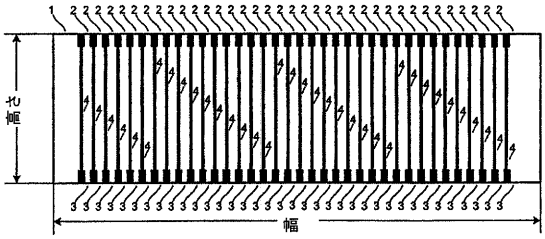
【図２４】



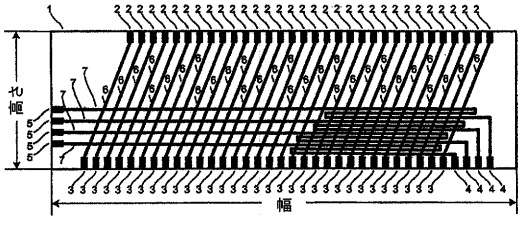
【図２５】



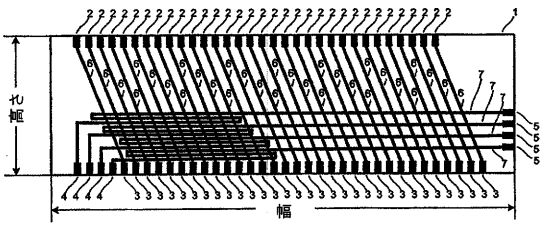
【図２６】



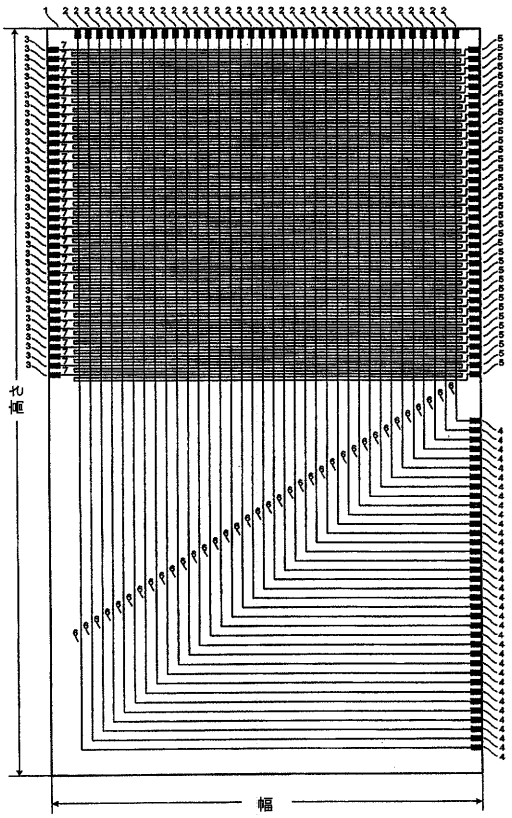
【図２７】



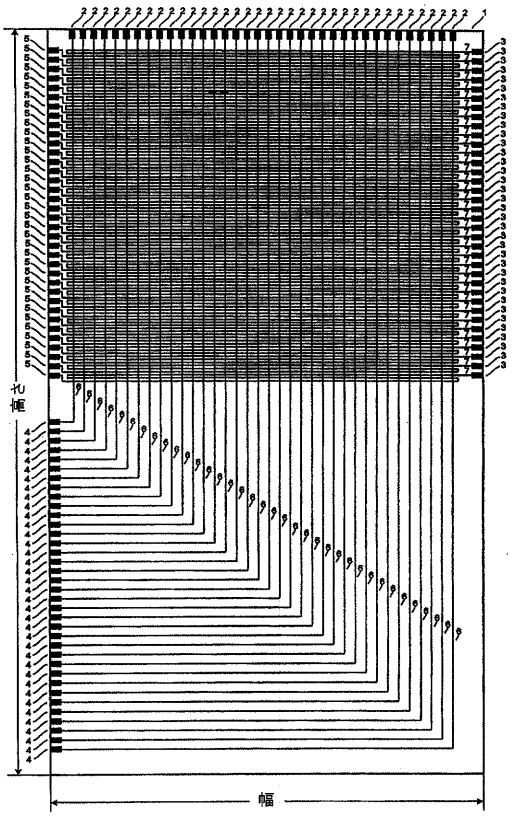
【図２８】



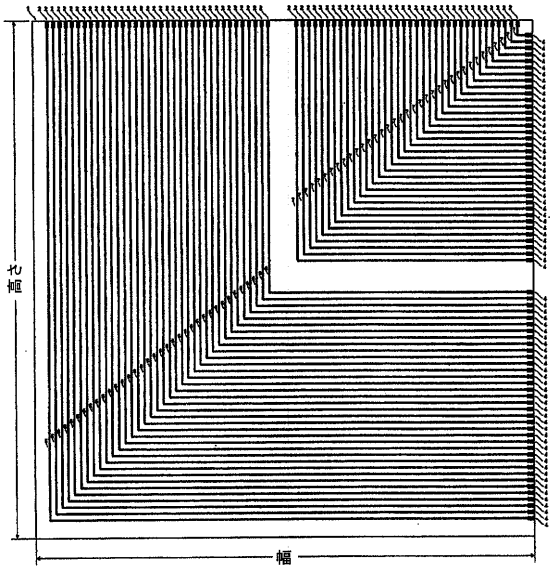
【図２９】



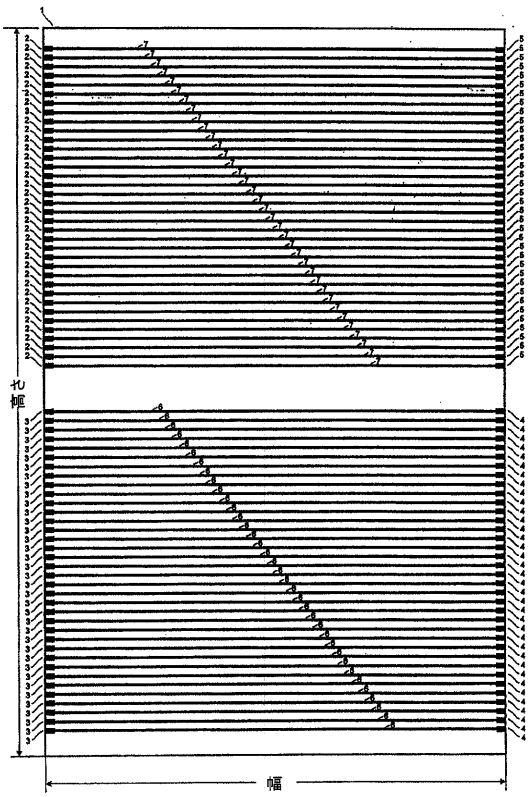
【図３０】



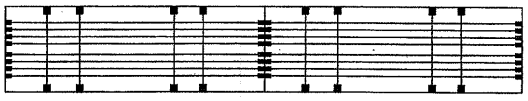
【図３１】



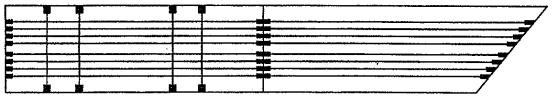
【図３２】



【図３３】



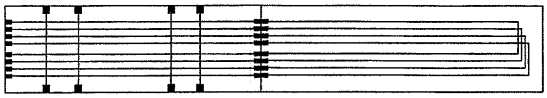
【図３４】



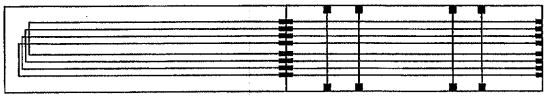
【図３５】



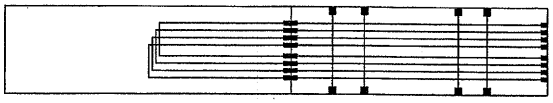
【図３６】



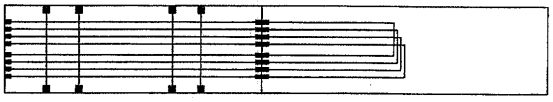
【図３７】



【図３８】



【図３９】



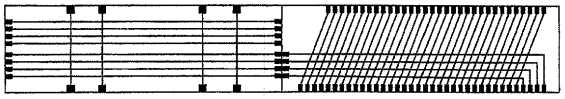
【図４０】



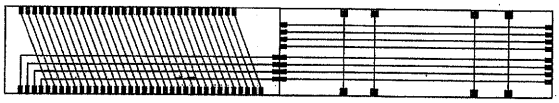
【図４１】



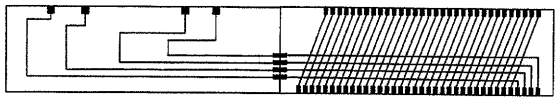
【図４２】



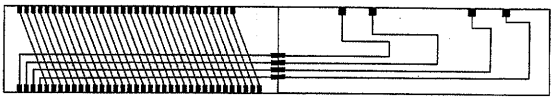
【図４３】



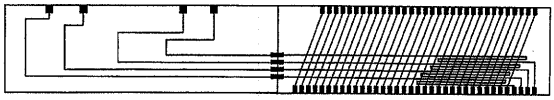
【図４４】



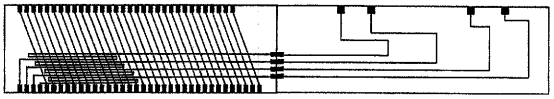
【図４５】



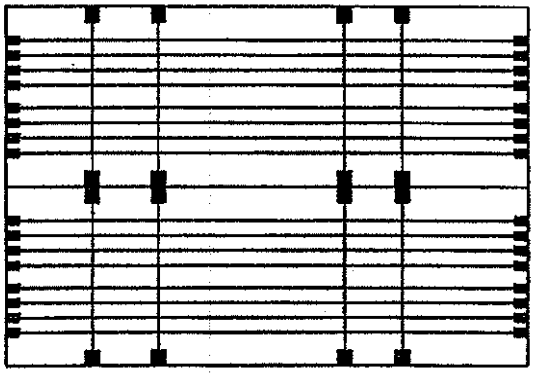
【図４６】



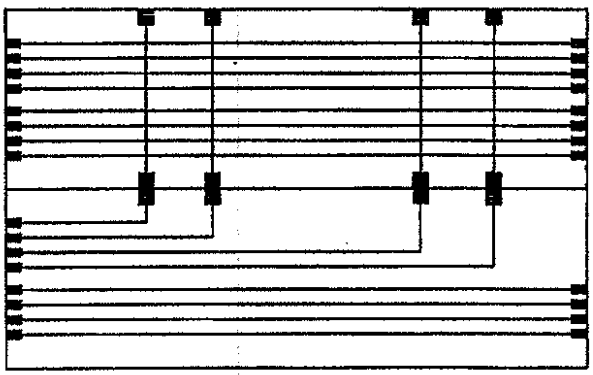
【図４７】



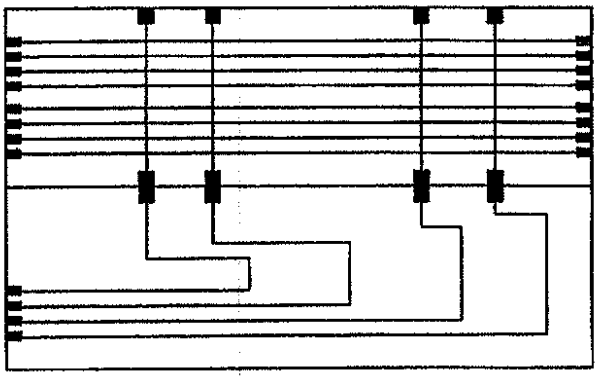
【図４８】



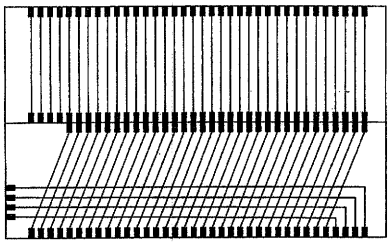
【図４９】



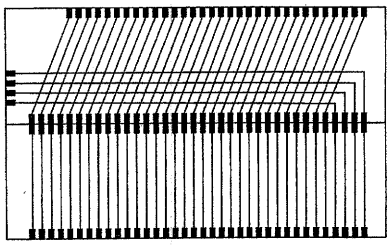
【図５０】



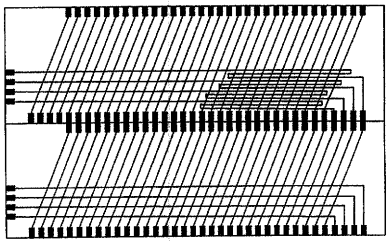
【図５１】



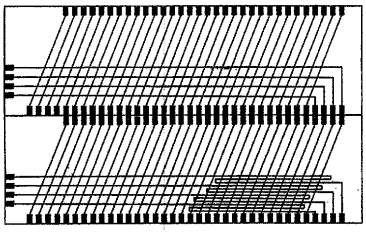
【図５２】



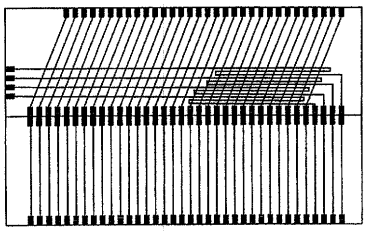
【図５３】



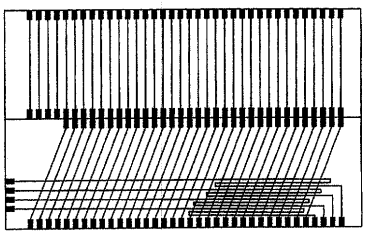
【図５４】



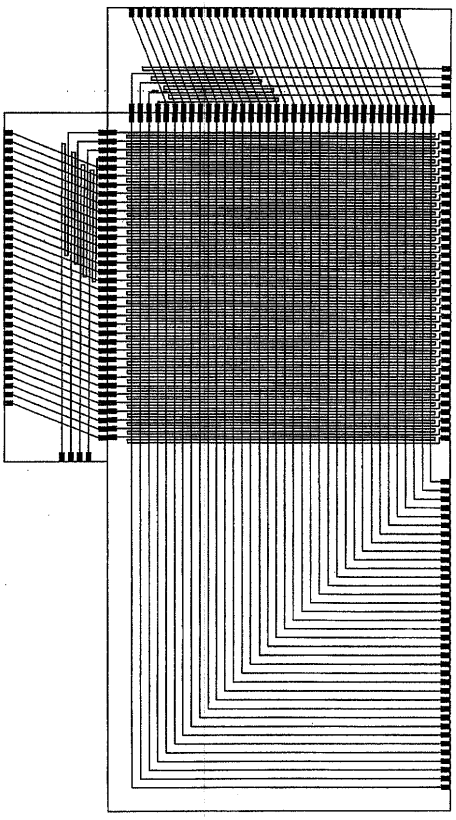
【図５５】



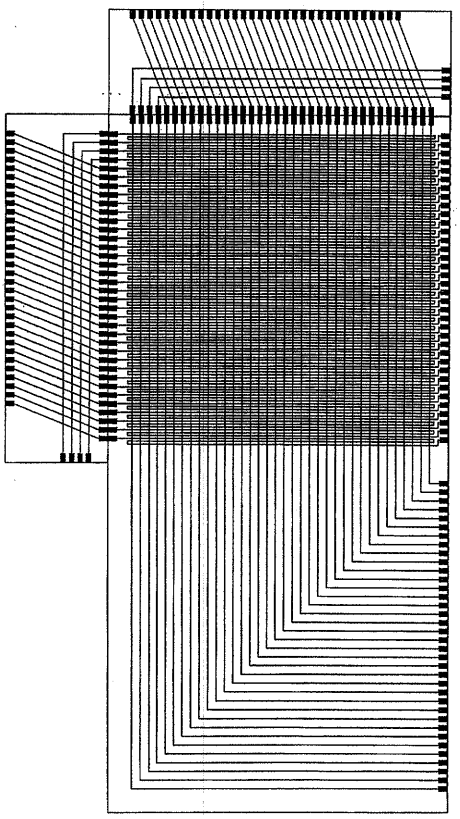
【図５６】



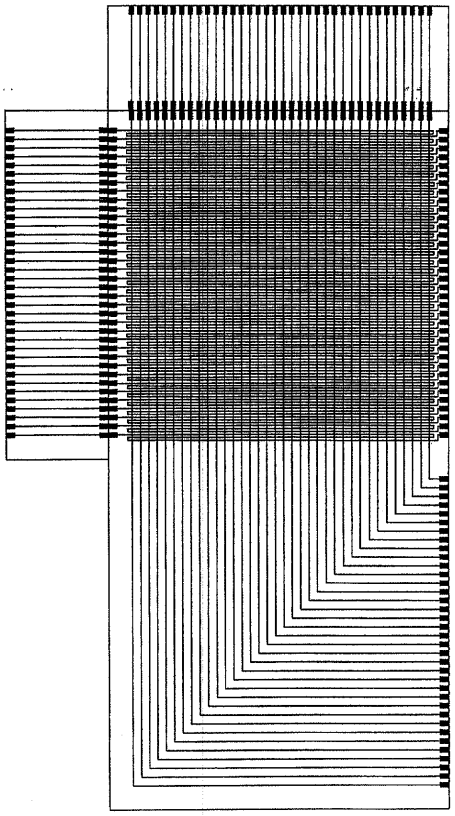
【図５７】



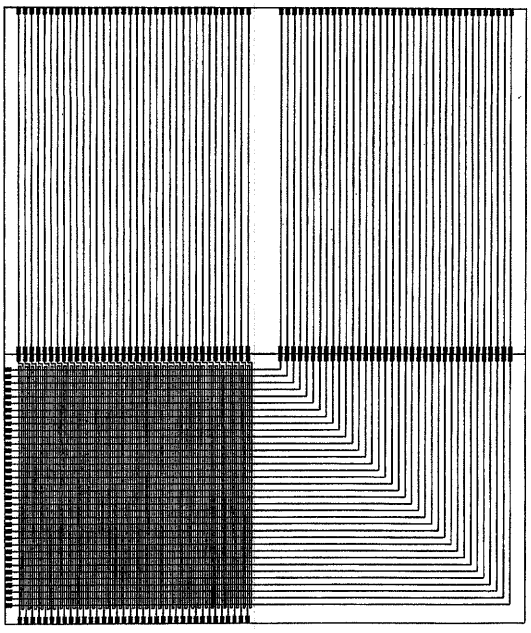
【図５８】



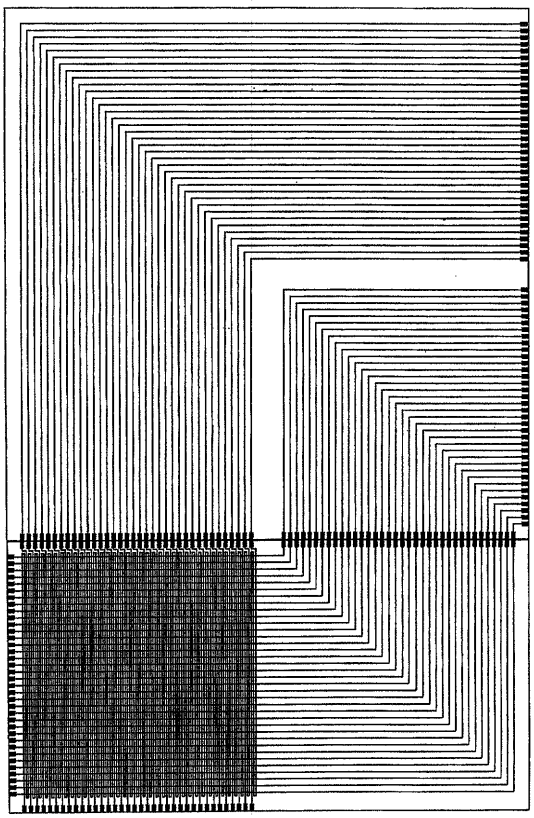
【図５９】



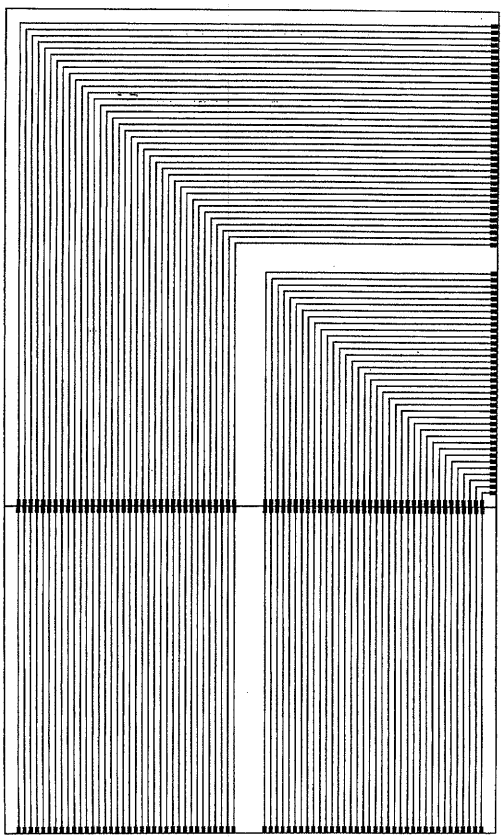
【図６０】



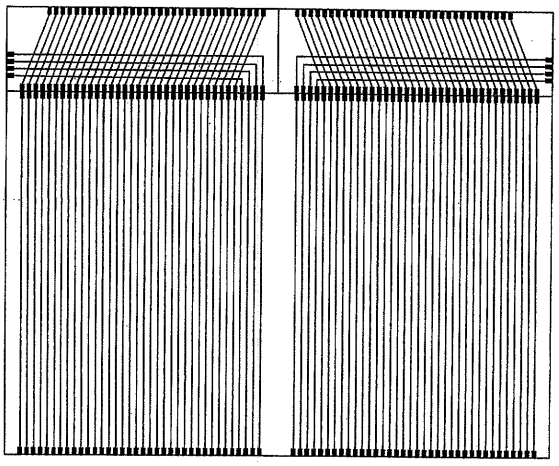
【図６１】



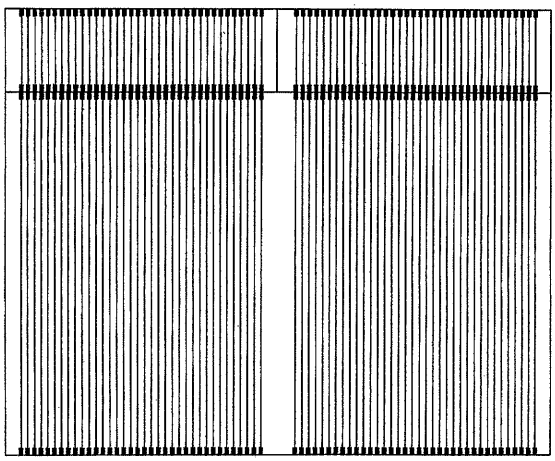
【図６２】



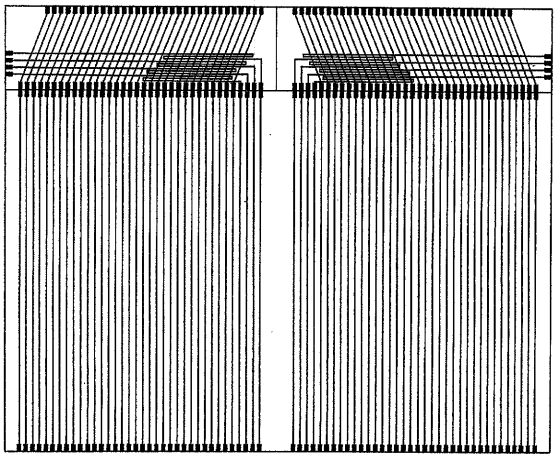
【図６３】



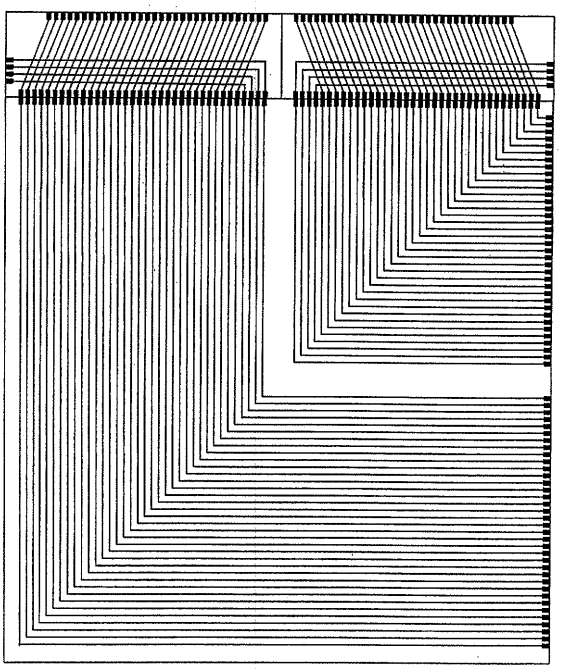
【図６４】



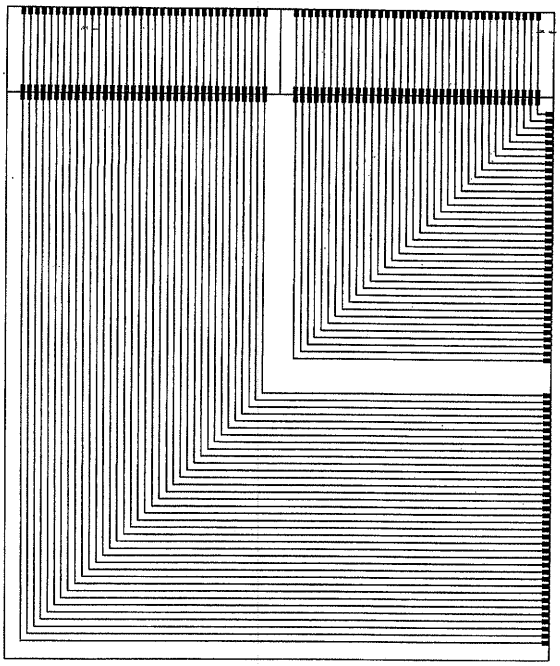
【図６５】



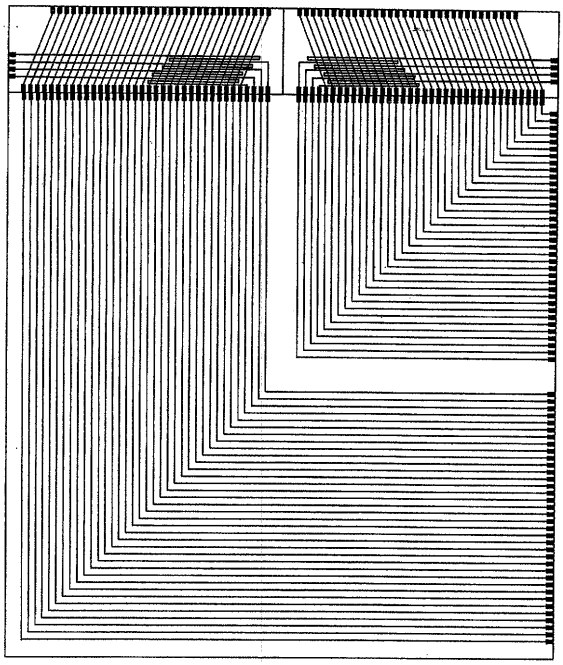
【図６６】



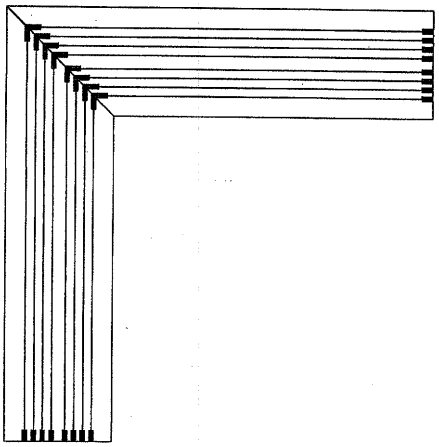
【図６７】



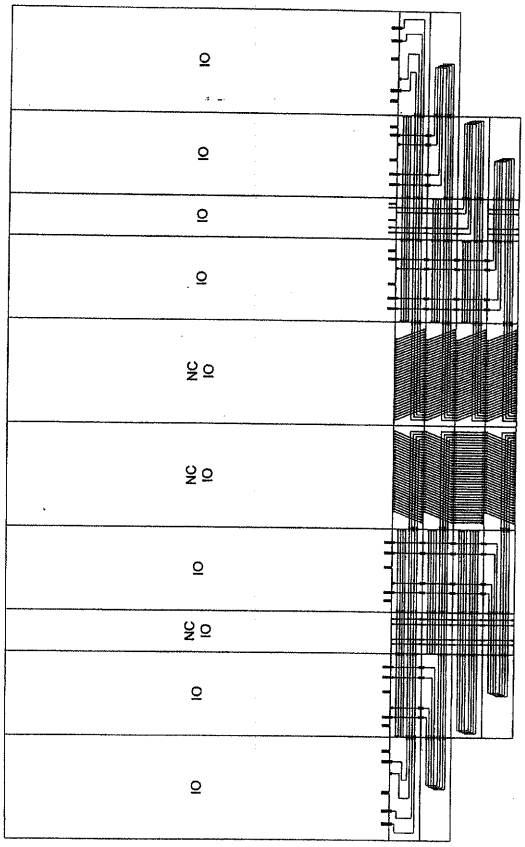
【図６８】



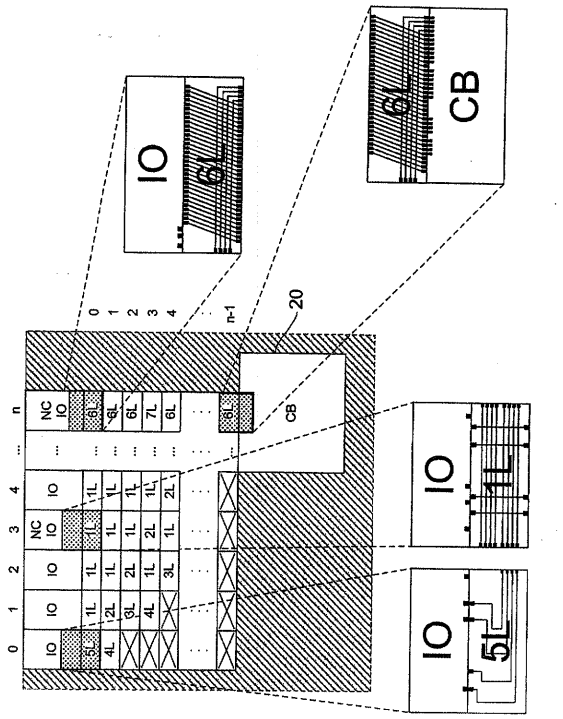
【図６９】



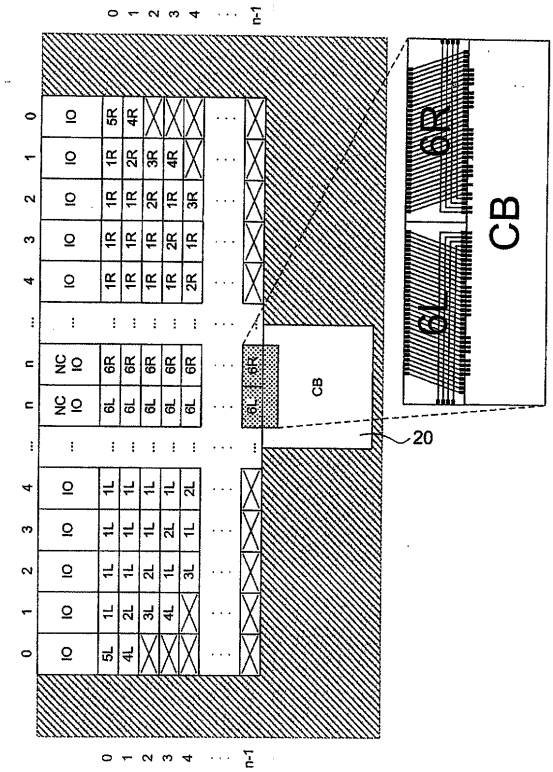
【図７０】



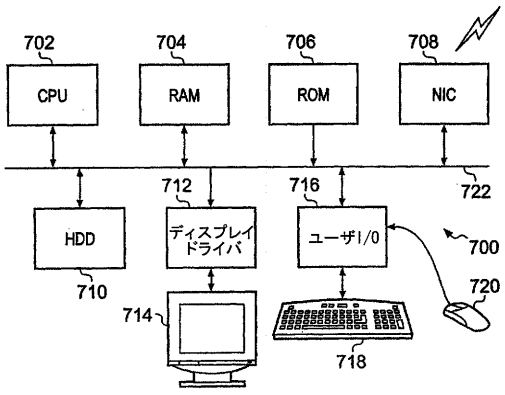
【図７１】


IO
6L
CB
20
1L
5L

【図７２】


IO
6R
CB
6L
20

【図７３】


702
CPU
704
RAM
706
ROM
708
NIC
722
712
ディスプレイドライバ
716
ユーザI/O
700
HDD
710
720
714
718

フロントページの続き

(72)発明者　コスタディン・ジチェフ
アメリカ合衆国・カリフォルニア・９５１３９・サン・ホセ・ペガサス・ウェイ・７４２２
(72)発明者　ビンダ・ブランドン・ワン
アメリカ合衆国・カリフォルニア・９５０３５・ミルピタス・サンドハースト・ドライブ・３８８

Ｆターム(参考) 5B046 AA08　BA04

【外国語明細書】

A METHOD, SYSTEM AND COMPUTER PROGRAM PRODUCT FOR DETERMINING ROUTING OF DATA PATHS IN INTERCONNECT CIRCUITRY

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to techniques for determining routing of data paths in interconnect circuitry of an integrated circuit.

Description of the Prior Art

Current design tools, such as digital layout EDA (electronic design automation) tools used in an ASIC (application specific integrated circuit) design flow process, have a severe limitation in achieving high accuracy in delay matching for timing critical data paths of interconnect circuitry, such as may be required for high speed nets and busses in deep submicron processes. This limitation is primarily due to how EDA tools approach the problem.

The current digital layout EDA tools use timing constraints as the requirement for the design layout. This is often referred to "timing driven place and route". This presents the ASIC layout tool with a practically unlimited number of layout implementations. Faced with this dilemma, EDA tool often use heuristic algorithms to automatically generate ASIC layouts, heuristic algorithms being algorithms which are not designed to determine the optimal solution, but instead are aimed at finding a feasible solution that, in objective function terms, is close to the optimal solution. The initial placement and route of the design rarely satisfies the design timing constraints. As a result the layout tools resort to iterations in place and route of the timing critical portions of the design with the hope of meeting the design timing constraints.

When the design requirements are very restrictive, as for example is typically the case for high speed balanced bus interconnects, the layout iterations often do not converge, and the tool hence fails to generate a usable layout. When the layout tool does succeed in meeting the design requirements, the layouts generated are without symmetry in either element placement or signal routing. Nets are of random length and randomly jump from layer to layer. The resulting capacitance, resistance,

coupling and inductance of each net cannot be accurately managed. This results in timing uncertainty in the layout.

Known digital layout EDA tools use several techniques for improving timing. Those techniques include: replacing one driving element with another with the same function but different drive strength; inserting buffers on critical paths to speed or slowdown delay; and/or breaking a long net with repeater buffers; moving critical cells physically closer. The timing variation from those approaches is quantized and introduces timing uncertainties. The matching between different timing critical paths is also greatly reduced over process corners, as those paths are built with different numbers of elements or with elements having different driving characteristics. The different elements' timing characteristics do not scale the same over process-temperature-voltage corners. Meeting one corner often mismatches other corners.

Due to the heuristic nature of the digital EDA layout algorithms, the same design can generate drastically different layouts with different tool versions or versions from different vendors. Often the same tool will produce a different layout with the same design when the layout procedure is repeated.

Another limitation of the current digital EDA layout flow is the inability of the tools to work with differential signalling, this being a method of transmitting information electrically by means of two complementary signals sent on two separate wires. Such differential signalling techniques can enable accurate propagation of fast signals though a noisy environment, such as for example are produced by the switching of digital gates.

As result, the known digital layout EDA tools and ASIC flows cannot produce design layouts that requiring a high degree of delay matching and noise immunity of the critical timing paths, nets and busses (referred to herein as "data paths") within interconnect circuitry.

The limitations of known digital layout EDA tools are exaggerated when different bits of a bus must be routed to design blocks which are not physically close together, yet require matched timing. DDR (double data rate) memory interfaces are a good example of this limitation. DDR memory achieves nearly twice the bandwidth of

non-DDR (single data rate) memory by double pumping (transferring data on the rising and falling edges of the clock signal) without increasing the clock frequency. The DDR Memory Controller has a relatively small layout with respect to the span of input/output (I/O) devices to which it must connect, and this places significant timing issues on the design of the interconnect circuitry to be used to couple the memory controller with those distributed I/O devices.

As an example, DDR3 is a parallel interface that uses both edges of the clock for reading and writing data from an off-chip SDRAM at data rates as high as 1600Mbs. The timing relationship between the data signals DQ, the strobe signal DQS, and the clock signal CK must be tightly controlled, yet the I/O devices for this design may span more than 2mm linearly (i.e. many orders of magnitude larger than the size of the memory controller block).

Budgeting for uncontrollable timing uncertainties such as clock jitter, on-chip-variation, process-voltage-temperature variations, printed circuit board trace variations, system noise and SDRAM setup and hold timing, requires propagation delay of the data paths within the interconnect circuitry between the Memory Controller and the I/O devices to be matched within tens of nanoseconds. Given interconnect delay which is in the region of hundreds of nanoseconds for nets of 2mm or more, the design of such interconnect circuitry is difficult to achieve with the current digital layout EDA tools.

Accordingly it would be desirable to provide an improved technique for determining routing of data paths in interconnect circuitry of an integrated circuit, which allows high accuracy in delay matching between the various data paths.

SUMMARY OF THE INVENTION

Viewed from a first aspect, the present invention provides a method of determining routing of data paths in interconnect circuitry for an integrated circuit, the interconnect circuitry on a first side providing a narrow interface for connection to a first device and on a second side providing a wide interface for connection to a distributed plurality of further devices, each data path being associated with one of said further devices and providing a connection through the interconnect circuitry between that associated further device and the first device, the method comprising the steps of: (i)

defining a plurality of cells to be provided along the wide interface, each of the further devices being associated with at least one of said cells; (ii) defining the interconnect circuitry as an array of blocks formed in rows and columns, each cell abutting one of said columns; (iii) providing a predetermined set of tiles, each tile providing a predetermined wiring layout; and (iv) for each block, applying predetermined rules to determine one of said tiles to be used to implement that block, the predetermined rules taking into account the location of the block in the array, and the association between the plurality of further devices and the plurality of cells, and ensuring that each data path provided by the interconnect circuitry has the same propagation delay.

In accordance with the present invention, a method is provided for determining the routing for data paths within interconnect circuitry used to couple a first device with a distributed plurality of further devices. The interconnect circuitry has a narrow interface for coupling with the first device and a wide interface for coupling with the distributed plurality of further devices. The interconnect circuitry is considered as an array of blocks formed in rows and columns, and a predetermined set of tiles is provided for use in constructing the interconnect circuitry, each tile providing a predetermined wiring layout. For each block, predetermined rules are applied to determine one of the tiles to be used to implement that block. By arranging the predetermined rules to take in to account the location of the block in the array, and the association between the plurality of further devices and the plurality of cells provided along the wide interface, this tile allocation process ensures that each data path provided by the interconnect circuitry has the same propagation delay.

Accordingly, it can be seen that the present invention provides a structured routing method using predetermined tiles, with predetermined rules being applied to determine which tiles are used to implement each block in the array of blocks forming the interconnect circuitry, so as to ensure that each data path provided by the interconnect circuitry has the same propagation delay.

Through use of the present invention, a number of benefits can arise when compared with the known prior art placement tools. Firstly, in addition to providing an effective mechanism for ensuring that each of the data paths within the interconnect

circuitry has the same propagation delay, the process can also match other physical effects such as coupling, cross-talk, loading, noise shielding, etc. Further, in embodiments of the present invention, the method can be implemented without the need for any iteration, hence providing significant design time speed benefits when compared with known EDA tools which need to perform iterative processes with the aim of finding a viable routing design. Furthermore, since the present invention involves using tiles having predetermined wiring layouts to produce data paths having the same propagation delay, this can give rise to interconnect circuits which will consume significantly less power than those designed with known EDA tools, which often insert buffers and/or other components within certain paths in order to improve timing and matching.

In one embodiment, each data path is formed by a sequence of tiles, each tile in the sequence being selected to have a wiring layout that at least partially abuts with the wiring layout in an adjacent tile of the sequence. Hence, each data path is formed by the combination of the various wiring layouts within the associated sequence of tiles used to form that data path. Due to the physical layout of the wires within each tile, the propagation delays associated with each tile will be predetermined. The predetermined set of tiles will provide a variety of different wiring layouts with different propagation delays, and this enables the predetermined rules to determine the appropriate tile to be used to implement each block taking in to account the location of the block within the array and the arrangement of the various further devices to be connected to the interconnect circuitry, in order to ensure that the propagation delays are matched for every data path.

In one embodiment, the predetermined rules allocate tiles to individual blocks in the array so as to produce a tree structure, at least one column abutting the narrow interface being formed as a trunk of the tree structure, and the rows in the remaining columns forming branches of the tree structures. At said step (iv), when determining tiles to be used to implement individual blocks in the array, said predetermined rules then take into account whether a block is in the trunk of the tree structure or a branch of the tree structure. It has been found that the use of such a tree structure enables a particularly efficient routing of the data paths to be achieved.

In one embodiment, said predetermined set of tiles comprise a predetermined subset of trunk tiles and a predetermined subset of branch tiles, during said step (iv), the predetermined rules selecting, for each block, a tile from the subset of trunk tiles if the block is in the trunk of the tree structure, or a tile from the subset of branch tiles if the block is in a branch of the tree structure.

In one embodiment, the number of cells exceeds the number of further devices to be connected to said wide interface, such that at least one non-connect cell exists that does not have one of said further devices associated therewith, and for each column formed as a trunk of the tree structure, the cell chosen to abut that column is a non-connect cell.

Hence, in such embodiments, when the predetermined rules are used to select the trunk tiles used to implement a column of blocks forming the trunk of the tree structure, there is no requirement for those tiles to provide any direct connection with one of the further devices, and instead the connection with the various further devices will occur via branch tiles used to implement the branches of the tree structure.

The tree structure can be designed in a variety of ways. However, in one embodiment, the tree structure comprises one or more tree sections, each tree section having one column abutting the narrow interface to form a trunk for that tree section, and the rows in the remaining columns of the tree section forming branches of that tree section. By enabling the tree structure to be formed from separate tree sections, this provides improved flexibility with regards to the layout of the data path routing and with respect to the placement of the further devices to be connected to the wide interface of the interconnect circuitry. For example, in one embodiment, at least one tree section is provided on a different axis to at least one other tree section, thereby allowing the further devices to be distributed along more then one axis. In one embodiment, such changes in axis may also be accommodated within a tree section itself.

Various types of tree section maybe provided. However, in one embodiment, each tree section comprises either a left tree section or a right tree section, at said step (iv) the predetermined rules applied being dependent on whether tiles are being determined for blocks of a left tree section or are being determined for blocks of a right

tree section, whilst still ensuring that each data path provided by the interconnect circuitry, whether in a left tree section or a right tree section, has the same propagation delay. In one embodiment such an approach can be used to produce the interconnect circuit's data path layout as a tree structure with a generally central trunk from which branches are formed in opposing directions. In one embodiment, both the left tree section and the right tree section can be arranged to have the same number of columns and rows, thereby enabling a symmetric design to be produced centred on the trunk of the tree structure.

The distributed plurality of further devices can be located on the integrated circuit in a variety of ways. However, in one embodiment the further devices are distributed on a distribution path following at least one axis. In one embodiment, the further devices may all be arranged along a linear path, whilst in other embodiments the devices may be provided along multiple linear paths which may be orientated differently with respect to each other.

In one embodiment, at at least one point in the distribution path a change in access takes place, and the method further comprises the steps of: at said step (ii), defining a corner element within the interconnect circuitry in association with each point in the distribution path where a change in axis take place, such that each corner element defines a change of axis between two adjacent columns of blocks in the array; at said step (iv), the predetermined rules implementing each corner element using predetermined corner tiles contained with said predetermined set of tiles, and when determining tiles to be used for each block in the array the predetermined rules taking into account the location of any corner elements within the interconnect circuitry so as to ensure that each data path provided by the interconnect circuitry has the same propagation delay.

By using predetermined corner tiles, the propagation delays involved in using such corner tiles will be predetermined, and accordingly the predetermined rules can take the location of any corner elements into account when determining for each block an appropriate tile to be used to implement that block within the interconnect circuitry, so as

to ensure that each data path within the interconnect circuitry has the same propagation delay.

If the corner elements are relatively short, this will limit the number of branches that can be provided on the far side of the corner element (with respect to the trunk) without layout congestion. In one embodiment, this can be alleviated by defining filler elements on either side of the corner element, and then implementing each filler element using predetermined filler tiles provided within the predetermined set of tiles. Such filler tiles in combination with the corner tiles will effectively extend the size of the corner element to allow for building tree structures that require a large number of branches on the far side of the corner element.

In embodiments where non-connect cells exist, the method may further comprise the step of associating with one or more non-connect cells one or more other components of the integrated circuit that do not require connection to the interconnect circuitry. As an example of such components that can be associated with non-connect cells, such components may comprise power supply lines used to power the distributed plurality of further devices that are to be connected to the interconnect circuitry.

When applying the predetermined rules to determine which tiles to allocate to each block in the array of blocks defining the interconnect circuitry, if the predetermined rules determine that no wiring layout is required at the location of a particular block, then in one embodiment no tile is allocated for that block. In an alternative embodiment, the predetermined set of tiles may include one or more filler tiles having no wiring layout, and if the predetermined rules determine that no wiring layout is required at the location of a particular block, one of those filler tiles is allocated for that block. In one embodiment, such filler tiles may have metal fill to satisfy the metal density requirements for the given process.

In one embodiment, all cells have equal width, as a result of which each of the columns of blocks will be of equal width and the predetermined set of tiles can hence be made of equal width. However, in alternative embodiment the plurality cells defined are of a plurality of different widths, and the predetermined set of tiles include tiles having a

corresponding plurality of different widths, so as to enable each cell to abut one of said columns.

However, in an alternative embodiment, the need to provide tiles for every possible width of cell can be avoided through use of filler cells. More particularly, at said step (i) certain cells are formed by a combination of an initial cell and a filler cell, such that each cell abutting with a column in the array is of equal width. By such an approach, each column of blocks can be arranged to be of the same width, and accordingly the predetermined set of tiles only needs to include tiles of that specified width, thereby reducing the number of tiles that need to be provided within the predetermined set.

In one embodiment, the first device is connected directly to the narrow interface of the interconnect circuitry. However, in an alternative embodiment the method further comprises the step of: defining a root channel passing from the narrow interface of the interconnect circuitry to the first device, thereby allowing placement of the first device within the integrated circuit at a location other than that of the narrow interface. In one particular embodiment, said predetermined set of tiles include a subset of root tiles, the method further comprising the step of producing the root channel using one or more root tiles from said subset. By such an approach, this provides freedom with regards to the placement of the first device with respect to the placement of the distributed plurality of further devices, whilst still ensuring that each data path provided by the interconnect circuitry has the same propagation delay. In particular, in one embodiment the propagation delay of each data path through the root channel is identical, and hence the inclusion of the root channel maintains the propagation delay matching properties of the technique of embodiments of the present invention.

The first device and the further devices can take a variety of forms. In one embodiment, the distributed plurality of further devices comprise input/output devices placed around the periphery of the integrated circuit. In one such device, the first device is a memory controller device used to control the plurality of input/output devices. Memory interfaces such as SDR, DDR and parallel bus architectures such as PCI are prime examples of technologies which can benefit from the routing methodology of

embodiments to the present invention. More generally, the technique of embodiments to the present invention is applicable to all interconnect circuitry where a high degree of timing matching is required for the data paths connecting a narrow interface to devices spaced physically relatively far apart along a wide interface.

In one embodiment, the various further devices connect directly to the adjacent tile in the associated column within the interconnect circuitry. However, in an alternative embodiment the method further comprises the steps of: providing one or more bridge tiles that provide pin mappings between pins of said further devices and wiring layouts of tiles in said predetermined set of tiles; and when, for a particular further device, there is not a direct mapping of its pins to the wiring layout of the tile in the interconnect circuitry that it is to abut to, introducing at least one bridge tile between said particular further device and said tile in the interconnect circuitry to map the pins of said particular further device to the wiring layout of that tile. Hence, such bridge tiles can be used to convert the pin locations on the further device to conform to the pin locations required for abutting with the relevant tile in the interconnect circuitry. Such bridge tiles can also serve to avoid accidental abutment of wiring within the interconnect circuitry with pins in the further devices which do not require connection through the interconnect circuitry.

In one embodiment, where non-connect cells are provided along the wide interface, addition of such non-connect cells has no effect on the narrow interface. Further, in such embodiments, addition of such non-connect cells along the wide interface has no effect on the order in which the data paths connect to the narrow interface. Hence, there is a great degree of flexibility in the manner in which the cells are provided along the wide interface, and the association between the further devices and the individual cells.

In one embodiment, the wiring layout of at least one of the tiles in the predetermined set is such that when that at least one of the tiles is used to implement a block within said array, not all of the wiring layout within that tile is utilised. Hence, in such embodiments, not all of the individual wires within the wiring layout of each tile will be utilised in all instantiations of that tile within the interconnect circuitry. By

designing tiles in such a way, this provides improved flexibility with regards to the use of those tiles.

Viewed from a second aspect, the present invention provides a computer program product comprising a computer program which when executed on a computer system causes the computer system to perform a method to determine routing of data paths in interconnect circuitry for an integrated circuit, the interconnect circuitry on a first side providing a narrow interface for connection to a first device and on a second side providing a wide interface for connection to a distributed plurality of further devices, each data path being associated with one of said further devices and providing a connection through the interconnect circuitry between that associated further device and the first device, the method comprising the steps of: (i) defining a plurality of cells to be provided along the wide interface, each of the further devices being associated with at least one of said cells;(ii) defining the interconnect circuitry as an array of blocks formed in rows and columns, each cell abutting one of said columns; (iii) providing a predetermined set of tiles, each tile providing a predetermined wiring layout; and (iv) for each block, applying predetermined rules to determine one of said tiles to be used to implement that block, the predetermined rules taking into account the location of the block in the array, and the association between the plurality of further devices and the plurality of cells, and ensuring that each data path provided by the interconnect circuitry has the same propagation delay.

Viewed from a third aspect, the present invention provides a system for determining routing of data paths in interconnect circuitry for an integrated circuit, the interconnect circuitry on a first side providing a narrow interface for connection to a first device and on a second side providing a wide interface for connection to a distributed plurality of further devices, each data path being associated with one of said further devices and providing a connection through the interconnect circuitry between that associated further device and the first device, the system comprising: means for defining a plurality of cells to be provided along the wide interface, each of the further devices being associated with at least one of said cells; means for defining the interconnect circuitry as an array of blocks formed in rows and columns, each cell abutting one of said

columns; means for providing a predetermined set of tiles, each tile providing a predetermined wiring layout; and data path determining means which, for each block, applies predetermined rules to determine one of said tiles to be used to implement that block, the predetermined rules taking into account the location of the block in the array, and the association between the plurality of further devices and the plurality of cells, and ensuring that each data path provided by the interconnect circuitry has the same propagation delay.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will be described further, by way of example only, with reference to embodiments thereof as illustrated in the accompanying drawings, in which:

Figure 1 is a diagram schematically illustrating an integrated circuit incorporating interconnect circuitry produced using the technique of embodiments of the present invention;

Figure 2 is a flow diagram illustrating the method used in one embodiment of the present invention to determine routing of data paths in the interconnect circuitry;

Figure 3 schematically illustrates the tree structure used for the interconnect circuitry in embodiments of the present invention;

Figure 4 schematically illustrates how the wiring layouts in individual tiles are used to produce the various data paths within the interconnect circuitry in accordance with one embodiment of the present invention;

Figure 5 schematically illustrates interconnect circuitry produced in accordance with one embodiment of the present invention as a balanced tree structure comprising both left and right tree sections;

Figure 6 schematically illustrates a left tree section produced in accordance with one embodiment of the present invention;

Figure 7 schematically illustrates a right tree section produced in accordance with one embodiment of the present invention;

Figure 8 schematically illustrates a balanced tree structure produced in accordance with one embodiment of the present invention, where the left tree section and right tree section are separated by a corner element;

Figure 9 is a flow diagram schematically illustrating the method employed in accordance with one embodiment of the present invention to produce right tree sections;

Figure 10 is a flow diagram schematically illustrating the method employed in accordance with one embodiment of the present invention to produce left tree sections;

Figures 11 to 32 illustrate a predetermined set of tiles used in one embodiment of the present invention;

Figures 33 to 69 illustrate how different tiles connect by abutment in accordance with one embodiment of the present invention;

Figure 70 shows the tile placement used to produce an example interconnect structure in accordance with one embodiment of the present invention;

Figures 71 and 72 illustrate how various tiles are used for connection of the various devices to the interconnect circuitry in accordance with one embodiment of the present invention; and

Figure 73 schematically illustrates the architecture of a general purpose computer which may execute a computer program to implement the techniques of embodiments to the present invention.

DESCRIPTION OF EMBODIMENTS

Figure 1 is an illustrative example of an integrated circuit 60 incorporating an interconnect circuit 40 designed using the techniques of embodiment of the present invention.

As shown in Figure 1, a plurality of devices/components spanning two sides of the integrated circuit 60 are connected to a device 20 within the integrated circuit via the interconnect circuitry 40. The devices 30 include devices which require connection to the device 20 via the interconnect circuitry 40, in Figure 1 these devices being labelled as IO (input/output) devices, and also may include one or more devices which do not require connection to the device 20. In Figure 1 the power supply components "VDD" 34, "VDDA" 35, "VSS" 36 and "VSSA" 37 are examples of such components which do not require connection to the device 20. A corner element 38 may also be provided at the junction between two linear distribution paths of devices.

The interconnect circuitry 40 provides a wide interface to the distributed devices 30, but presents a narrow interface to the device 20. As shown in Figure 1, a root channel 50 may be provided for routing signals from the narrow interface of the interconnect circuitry 40 to the device 20 so as to increase flexibility as to where the device 20 is positioned within the integrated circuit 60. Alternatively, the device 20 can be connected directly to the narrow interface of the interconnect circuitry without the need for such a root channel.

As will be discussed in more detail later, when designing the interconnect circuit, a number of cells (also referred to in Figure 1 as pads) are provided along the wide interface of the interconnect circuit, with each of the various devices/components 30 then being associated with at least one of those cells. In one embodiment, each device 30 is associated with one of the cells, but in alternative embodiments certain of the devices may be associated with more than one cell, for example if a device is too large to be associated with a single cell.

In one embodiment of the present invention, the data paths within the interconnect circuitry are designed so as to adopt a tree structure including a trunk portion and a number of branch portions. The tree structure can comprise one or more tree sections, and may for example include a left tree section and/or a right tree section. In the example of Figure 1, the tree structure comprises both a left tree section and a right tree section, with the left tree section being to the left of the dotted line shown in Figure 1 and the right tree section being to the right of the dotted line. The trunk of each tree section is aligned with the cell either side of the dotted line, and hence the trunk of the left tree section is aligned with the VDD cell 42 and the trunk of the right tree section is aligned with the VSS cell 44.

In the embodiments shown in Figure 1, there are an equal number of cells/pads in the left tree section and the right tree section. By placing such a constraint on the design, it is easier to produce an interconnect circuit where the various data paths have the same propagation delay, as will be discussed in more detail with reference to the remaining figures. However, in alternative embodiments, it would be possible to remove this constraint if different types of tiles were designed for the left tree section

and the right tree section, respectively, so as to ensure that all data paths within the interconnect circuitry have the same propagation delay.

In Figure 1, it is assumed that the device 20 is a memory controller device, and the distributed plurality of devices provided along the wide interface of the interconnect circuit are IO devices 32 controlled by the memory controller device. However, the technique of embodiments of the present invention can also be applied to any other interconnect structure where a device needs to be coupled via an interconnect to a distributed plurality of other devices in a manner where propagation delay is matched for all of the data paths through the interconnect.

Figure 2 schematically illustrates the method used in one embodiment of the present invention to determine the routing of data paths within the interconnect circuitry 40. Firstly, at step 100, the number of cells to be provided along the wide interface of the interconnect circuit is determined. This is determined having regards to the number of devices which must connect to the device 20 through the interconnect circuit, and the number of non-connecting components that need to be accommodated. In one embodiment of the present invention, it is a requirement that there is a non-connecting component associated with the cell that is aligned with any trunk portion of the tree structure.

Following step 100, at step 110, an array of blocks is formed to represent the interconnect, the array being formed into rows and columns of blocks. Each cell then abuts one of the columns in the array. Hence, it can be seen that the number of cells determined at step 100 determines the number of columns that need to be provided in the array. In one embodiment, the number of rows is chosen to be one less than the number of columns in any particular tree section of the tree structure. Hence, purely by way of example, if there are 100 cells determined at step 100, and the interconnect circuitry is to be formed by a tree structure having a left tree section and a right tree section, each tree section will have 50 columns, and 49 rows.

At step 120, each block within the array is considered in turn, and for each block considered, predetermined rules are applied to determine a tile from a predetermined plurality of tiles to be used to represent that block. As will be discussed

in more detail later, a predetermined set of tiles are provided, each tile having a predetermined wiring layout within it. Different tiles will be provided for different widths of cell, and accordingly different widths of columns within the interconnect circuitry. The predetermined rules take into account the location of the block in the array, and the association between the plurality of devices being connected to the interconnect circuitry (e.g. the IO cells 32 in Figure 1) and the plurality of cells (i.e. taking into account which cells those various devices are associated with). Based on this information, the predetermined rules can choose an appropriate tile for each block location, so that once each of the blocks has been considered, the required data paths between the device 20 and each of the IO devices 32 will be provided within the interconnect circuitry, such that each of those data paths has the same propagation delay.

Accordingly, as shown by step 130, once each block has been considered at step 120, a layout design is output for the interconnect circuit, with each required data path being formed by a sequence of interconnected tiles, and each data path having the same propagation delay.

Figure 3 schematically illustrates the tree structure used for the interconnect circuit 40 in more detail. As shown, the device 20 can be coupled via an optional root channel 50 to a narrow interface 160 of the interconnect tree structure 40. In this example, the interconnect tree structure 40 is formed as a left tree section having a trunk formed by the column 165, and a plurality of branches 170, 175, 180, 185, 190 emanating to the left of the trunk 165. In this example, it can be seen that the array of blocks is constituted as five rows and six columns, with the six columns providing the wide interface 150 to the various cells with which the devices are to be associated. In this example, there will accordingly be six cells along the wide interface and associated with each cell will be either a device to be connected to the interconnect, or a device/component which is not to be connected to the interconnect.

Figure 4 illustrates the same formation of interconnect tree structure 40, but illustrates schematically the data paths that may be produced within that tree structure once tiles have been allocated to each of the blocks. As before, the optional root

channel 50 is provided to connect the device 20 to the narrow interface of the interconnect 40. As will be appreciated from Figure 4, the data paths provided between the device 20 and the cells 1 to 5 are each of the same length, and will accordingly have the same propagation delays. As discussed earlier, any device associated with cell 0 200 needs to be chosen to be a device which is not requiring connection to the interconnect 40, given the earlier mentioned constraint that any device requiring connection to the interconnect is not placed in a position where it abuts the trunk of the tree structure. Regarding the other cells 205, 210, 215, 220, 225, these may all have connected devices associated with them, or one or more of them may have non-connected devices associated with them.

Figure 5 is an example of an interconnect circuit configuration produced using one embodiment of the present invention. Those IO cells 260 labelled NC (non-connect) are cells which are to be associated with components/devices that do not require connection to the device 20 via the interconnect circuitry. This example illustrates the main subcomponents of this interconnect:

- Root 50 - this connects device 20 to the tree structure 40. The root is built using tiles 10 and 11.
- Tree structure 40 – this connects the root 50 to the individual IO cells 260. It is built using the tiles 0L, 0R, 1L, 1R, 2L, 2R, 3L, 3R, 4L, 4R, 5L, 5R, 6L, 6R, 7L, 7R, 8L and 8R. This makes up the main body of the interconnect circuitry.

The tree itself is divided into two sections. The left tree section is built using 0L, 0R, 1L, 2L, 3L, 4L, 5L, 6L, 7L, and 8L tiles. In Figure 5, the left tree section spans from the middle of the figure, column C = 0, to the left ending with column C = 12. This section contains a corner element 250 which is not given a column number. A corner element is built with 0L and 0R tiles. Each 0L and 0R tile is indexed by its row number. In Figure 5, the index is shown in square brackets as 0L[R] and 0R[R], where R in the square brackets is the row associated with the tile.

The right tree section is build using 0L, 0R, 1R, 2R, 3R, 4R, 5R, 6R, 7R, and 8R tiles. In Figure 5, the right tree section spans from the middle of the figure, column

C=0, to the right ending with column C=12. This section in Figure 5 does not contain a corner, but corners can be incorporated into the layout of the right tree section if desired.

The left and right tree sections are again divided into two parts, namely the trunk and the branches. The left tree branches and the right tree branches are built using 0L, 0R, 1L, 1R, 2L, 2R, 3L, 3R, 4L, 4R, 5L, and 5R tiles. They are contained in columns with values C>0. The tree branches have the same numbers as the rows R of the tree. The rows are shown on the left bottom and right side of the figure for the left and right tree sections respectively.

The tree trunks are built using 6L, 6R, 7L, 7R, 8L and 8R tiles. In Figure 5, the tree trunks are made of the tiles in columns C=0.

Tiles which are labelled X are "don't care" tiles, and they can either be omitted when producing the interconnect circuitry, or instead a tile containing metal fill cell can be used to meet metal fill density requirements by the technology.

There are three type main types of tree configurations. A left branched tree contains only the left tree section while the right tree section is omitted. A left branched tree can contain a corner element. Figure 6 shows an example of a left branched tree 270 connected directly at the narrow interface of its trunk to the device 20. The IO cells 260 are again shown associated with each column, some of the cells being associated with devices to be connected to data paths of the interconnect circuit (the IOL cells) and other cells being associated with non-connected devices (the NC IOL cells). This left branched tree of the example of Figure 6 does not have a corner element or a root.

The right branched tree contains a right tree section while the left tree section is omitted. As with a left branched tree, a right branched tree can contain a corner element. Figure 7 shows an example of a right branched tree 290 which does not have a corner element or a root. Again, in this example, the right branched tree 290 is connected directly at the narrow interface of its trunk to the device 20, and the cells 260 are shown associated with each column.

A balanced tree contains both a left tree section and a right tree section as shown, by way of example, earlier with reference to Figure 5. A balanced tree can have two corner elements, one on the left tree section and one on the right tree section.

The angled balanced tree is a special case of the balanced tree which has a corner element between the left tree section and the right tree section. Figure 8 shows an example of an angled balanced tree. The angled balanced tree can have three corner elements, namely one on the left tree section, one on the right tree section, and one between the left tree section and the right tree section. In the Figure 8 example, the angled balanced tree has a left tree section 270 separated by a corner 280 from the right tree section 290. The device 20 is separated from the tree by a root 50 formed of tiles 9, 10 and 11. As with the earlier figures 5 to 7, cells 260 are shown associated with each column of the tree, for association with either connected devices or non-connected devices.

Whilst in the above examples, it is indicated that any tree section may have at most one corner element in it, it will be appreciated that in alternative embodiments more than one corner element may be provided within each tree section.

Figure 9 show the flow charts of the algorithm used in one embodiment for building right branched trees. The algorithm is also used to build right tree sections of balanced trees. Figure 10 shows the flowchart of the algorithm used in one embodiment for building left branched trees. The algorithm is also used to build left tree sections of balanced trees. Both algorithms use the following parameters and variables:

N – Constant for the given configuration whose value represents the number of IO cells that are part connected to via the interconnect circuit.

CCL – Constant for the given configuration whose value represents the IO cell column location after which a corner cell will be instantiated.

c – Variable that contains the selected column location to instantiate a tile.

r – Variable that contains the selected row location to instantiate the tile.

In figures 9 and 10 the tiles are referred to as GDS tiles. The algorithms build trees by specifying the type of tile and its location in the tree by using row r and column c coordinates. The algorithm illustrated in figures 9 and 10 does not instantiate "don't care" tiles if they are used. Left branched trees are built from right to left as shown in Figure 6. This means that column c = 0 is the right most column and column c = N-1 is the left most column. Right branched trees are built from left to right as shown in Figure 7. This means that column c = 0 is the left most column and column c = N-1 is the right most column. The balanced trees are built by placing together a left branched tree and a right branched tree abutted at columns c = 0 as shown in Figure 5, or at a 90° angle as shown in Figure 8. The tree structures produced for the interconnect circuit using the algorithm illustrated in figures 9 and 10 have the following constraints:

1. All the tiles, IO cells, and device 20 are connected together by abutment.
2. Bit positions for interconnect busses (data paths) must be maintained when abutting elements.
3. The number of columns when building balanced trees for the left tree section and the right tree section must be the same, to ensure the propagation delay is matched.
4. IO Cells associated with columns c = 0 must not be associated with devices that require connection to device 20 through the interconnect.
5. Corners are built with 0L and 0R tiles which are not given a column number, but are placed after the tile with (c,r) grid location.

By following the processes defined in Figures 9 and 10, right and left tree sections can be formed as shown in the earlier figures. For example, by applying the process of Figure 9, the right tree section shown in Figure 7 can be produced. In this example, N = 11. Starting at location r = 0, c = 0, it can be seen that the process proceeds through steps 300, 305, 310, 320, 325, 330, 405 to step 415, where it is determined that there are no corners on this tree section. Accordingly, the process proceeds to step 420, where the tile 6R is inserted at row 0, column 0, whereafter the process passes to step 400. At step 400, the column value c is incremented whereafter the process proceeds via steps 325, 330, 335, 350, 360 to step 370. At this point, the

condition specified in step 370 is met, causing the process to branch to step 375 where the tile 1R is placed at column 1, row 0, whereafter the process returns to step 400.

Once the column value has been incremented at step 400 to a value which equates with the value N, then the process branches to step 450 where the row value is incremented and thereafter at step 320 the column value is reset to 0, causing the process to be repeated for all of the block locations of row 1. When it is determined at step 310 that the incremented row value is no longer less than N - 1, then the process finishes at step 315.

Figure 10 illustrates the corresponding process used to produce left tree sections. Hence, for example considering the left tree section of Figure 6, this tile allocation as shown in Figure 6 is produced by performing the process of Figure 10. Purely by way of example, considering the block location at r = 5, c = 5, it will be seen that from step 525, the process follows path 530, 535, 550 to point 560. At this point, the condition specified at step 560 will be met, since r = 5, c = 5 and N = 11. Accordingly, the process branches to step 565 where the tile 2L is placed at that grid location, as indeed is clearly shown in Figure 6.

From a similar review of Figures 9 and 10 in association with any of Figures 5 to 8, it will be seen that the process as described in these figures will produce the tile layouts shown in Figures 5 to 8. Due to the wiring layouts provided in each of these tiles, the processes of Figures 9 and 10 produce data paths within the interconnect structure that all have the same propagation delay. Further, as will be appreciated from Figures 9 and 10, this design layout process requires no evaluation of the layout produced to see if it meets the timing constraints, and hence does not require any of the iterations that are commonly required with prior art EDA tools.

A root structure is not necessary for every interconnect type. Device 20 can directly abut to the tree trunk as shown in figures 6 and 7. In general, roots are built from tiles 9, 10, and 11. Left branched trees, right branched trees, and balanced trees can use tiles 9, 10 and 11 to connect the root to the tree. Angled balanced trees use tile 9 to connect the Root to the tree. Tiles 11 and 12 are used to extend the Root to the location where it abuts with the device 20 as illustrated in as illustrated in Figure 8.

The interconnect tree structure is built from a set of tiles which when placed together connect though abutment. There are a number of common parameters that describe those tiles:

N – Number of IO. This parameter lists the total number of IO cells which will be part of the IO ring 260.

I – Number of Connected IO. This parameter designates the number of IO cells which require connection to device 20 through the interconnect.

P – Port Width. This parameter designates the maximum number of connections, or bits, which an IO cell will need to propagate to device 20 through the interconnect.

Z – Maximum Required Horizontal Metal Length. This parameter designates the longest of the following:

1. The longest wire required to connect an east port to a west port of any tile.
2. Twice the longest wire required to connect an east port to a west port of any 0R or 0L tile.

V – Maximum Required Vertical Metal Length. This parameter designates the longest wire to connect a north port to a south port of any tile.

H – Tile Height. This parameter designates the minimum height required to support two ports of width P bits on the east or west side of any tile.

Figure 11 is an example of a tile "0L" for illustrative purpose. Most implementations will use multiple metal layers and vias to create ports and interconnect wiring.

- The tile has a height of H.
- The tile width varies depending on its row location.
- All the ports (2, 3, 4 and 5) have P number of connections (bits) each.
- Interconnect wiring 6 and 7 have a length of M where,

  $M = 0.5*Z$

Figure 12 is an example of a tile "0R" for illustrative purpose. Most implementations will use multiple metal layers and vias to create ports and interconnect wiring.

- The tile has a height of H.
- The tile width varies depending on its row location and the size of the corner element.
- All the ports (2, 3, 4, and 5) have P number of connections (bits) each.
- Interconnect wiring 6 and 7 have a length of M where,

  $M = 0.5*Z$

The same tile wiring design can be used for type 0L[r] and 0R[r] for a given row r, by using a horizontal flip, if the left and right tiles have equal width.

Figure 13 is an example of a tile "1L" for illustrative purpose. Most implementations will use multiple metal layers and vias to create ports and interconnect wiring.

- The tile has height of H.
- One tile will be built for each IO cell width used in the interconnect circuitry including IO cells with no connections to device 20.
- All the ports (2, 3, 4, 5, 6, and 7) have P number of connections (bits) each.
- Interconnect wiring 8 and 9 have a length of M where,

  $M = Z$

- Interconnect wiring 10 has a length of V.

Figure 14 is an example of a tile "1R" for illustrative purpose. Most implementations will use multiple metal layers and vias to create ports and interconnect wiring.

- The tile has height of H.
- One tile will be built for each IO cell width used in the interconnect circuitry.
- All the ports (2, 3, 4, 5, 6, and 7) have P number of connections (bits) each.
- Interconnect wiring 8 and 9 have a length of M where,

  $M = Z$

- Interconnect wiring 10 has a length of V.

The same tile wiring design can be used for type 1L and 1R by using a horizontal flip, if the ports on the north and south side have vertical symmetry.

Figure 15 is an example of a tile "2L" for illustrative purpose. Most implementations will use multiple metal layers and vias to create ports and interconnect wiring.

- The tile has height of H.
- One tile will be built for each IO cell width used in the interconnect circuitry.
- All the ports (2, 3, 4, and 5) have P number of connections (bits) each.
- Interconnect wiring 6 has a length of M where,

  $M = 0.5*Z + 0.5*V$
- Interconnect wiring 7 has a length of M where,

  $M = Z$

Figure 16 is an example of a tile "2R" for illustrative purpose. Most implementations will use multiple metal layers and vias to create ports and interconnect wiring.

- The tile has height of H.
- One tile will be built for each IO cell width used in the interconnect circuitry.
- All the ports (2, 3, 4, and 5) have P number of connections (bits) each.
- Interconnect wiring 6 has a length of M where,

  $M = 0.5*Z + 0.5*V$
- Interconnect wiring 7 has a length of M where,

  $M = Z$

The same tile wiring design can be used for type 2L and 2R by using a horizontal flip if the ports on the north side have vertical symmetry.

Figure 17 is an example of a tile "3L" for illustrative purpose. Most implementations will use multiple metal layers and vias to create ports and interconnect wiring.

- The tile has height of H.
- One tile will be built for each IO cell width used in the interconnect circuitry.
- All the ports (2 and 3) have P number of connections (bits) each.
- Interconnect wiring 4 has a length of M where,

  $M = 2*Z + 0.5*V$

Figure 18 is an example of a tile "3R" for illustrative purpose. Most implementations will use multiple metal layers and vias to create ports and interconnect wiring.

- The tile has height of H.
- One tile will be built for each IO cell width used in the interconnect circuitry.
- All the ports (2 and 3) have P number of connections (bits) each.
- Interconnect wiring 4 has a length of M where,

  $M = 2*Z + 0.5*V$

The same tile wiring design can be used for type 3L and 3R by using a horizontal flip.

Figure 19 is an example of a tile "4L" for illustrative purpose. Most implementations will use multiple metal layers and vias to create ports and interconnect wiring.

- The tile has height of H.
- One tile will be built for each IO cell width used in the interconnect circuitry.
- All the ports (2 and 3) have P number of connections (bits) each.
- Interconnect wiring 4 has a length of M where,

  $M = Z + 0.5*V$

Figure 20 is an example of a tile "4R" for illustrative purpose. Most implementations will use multiple metal layers and vias to create ports and interconnect wiring.

- The tile has height of H.
- One tile will be built for each IO cell width used in the interconnect circuitry.
- All the ports (2 and 3) have P number of connections (bits) each.
- Interconnect wiring 4 has a length of M where,

  $M = Z + 0.5*V$

The same tile wiring design can be used for type 4L and 4R by using a horizontal flip.

Figure 21 is an example of a tile "5L" for illustrative purpose. Most implementations will use multiple metal layers and vias to create ports and interconnect wiring.

- The tile has height of H.
- One tile will be built for each IO cell width used in the interconnect circuitry.
- All the ports (2 and 3) have P number of connections (bits) each.
- Interconnect wiring 4 has a length of M where,

  M = 1.5Z +0.5*V

Figure 22 is an example of a tile "5R" for illustrative purpose. Most implementations will use multiple metal layers and vias to create ports and interconnect wiring.

- The tile has height of H.
- One tile will be built for each IO cell width used in the interconnect circuitry.
- All the ports (2 and 3) have P number of connections (bits) each.
- Interconnect wiring 4 has a length of M where,

  M = 1.5Z +0.5*V

The same tile wiring design can be used for type 5L and 5R by using a horizontal flip if the ports on the north side have vertical symmetry.

Figure 23 is an example of a tile "6L" for illustrative purpose. Most implementations will use multiple metal layers and vias to create ports and interconnect wiring.

- The tile has height of H.
- The tile width is the same as the IO cell width on the tree trunk.
- Ports 2 and 3 have S number of connections where,

  S >= (I-1)*P
- Ports 4 and 5 have S number of connections where,

  S = P
- Interconnect wiring 6 has a length of M where,

  M = V
- Interconnect wiring 7 has a length of M where,

M = Z +0.5*V

Figure 24 is an example of a tile "6R" for illustrative purpose. Most implementations will use multiple metal layers and vias to create ports and interconnect wiring.

- The tile has height of H.
- The tile width is the same as the IO cell width on the tree trunk.
- Ports 2 and 3 have S number of connections where,

  S >= (I-1)*P
- Ports 4 and 5 have S number of connections where,

  S = P
- Interconnect wiring 6 has a length of M where,

  M = V
- Interconnect wiring 7 has a length of M where,

  M = Z +0.5*V

The same tile wiring design can be used for type 6L and 6R by using a horizontal flip.

Figure 25 is an example of a tile "7L" for illustrative purpose. Most implementations will use multiple metal layers and vias to create ports and interconnect wiring.

- The tile has height of H.
- The tile width is the same as the IO cell width on the tree trunk.
- Ports 2 and 3 have S number of connections where,

  S >= I*P
- Interconnect wiring 4 has a length of M where,

  M = V

Figure 26 is an example of a tile "7R" for illustrative purpose. Most implementations will use multiple metal layers and vias to create ports and interconnect wiring.

- The tile has height of H.
- The tile width is the same as the IO cell width on the tree trunk.

- Ports 2 and 3 have S number of connections where,

  S >= I*P

- Interconnect wiring 4 has a length of M where,

  M = V

The same tile wiring design can be used for type 7L and 7R by using a horizontal flip.

Figure 27 is an example of a tile "8L" for illustrative purpose. Most implementations will use multiple metal layers and vias to create ports and interconnect wiring.

- The tile has height of H.
- The tile width is the same as the IO cell width on the tree trunk.
- Ports 2 and 3 have S number of connections where,

  S >= (I-1)*P

- Ports 4 and 5 have S number of connections where,

  S = P

- Interconnect wiring 6 has a length of M where,

  M = V

- Interconnect wiring 7 has a length of M where,

  M = 2*Z +0.5*V

Figure 28 is an example of a tile "8R" for illustrative purpose. Most implementations will use multiple metal layers and vias to create ports and interconnect wiring.

- The tile has height of H.
- The tile width is the same as the IO cell width on the tree trunk.
- Port 2 and 3 have S number of connections where,

  S >= (I-1)*P

- Ports 4 and 5 have S number of connections where,

  S = P

- Interconnect wiring 6 has a length of M where,

  M = V

- Interconnect wiring 7 has a length of M where,

  $M = 2*Z + 0.5*V$

The same tile wiring design can be used for type 8L and 8R by using a horizontal flip.

Figure 29 is an example of a tile "9L" for illustrative purpose. Most implementations will use multiple metal layers and vias to create ports and interconnect wiring.

- The tile has height of T where,

  if X is the IO cell width on the tree trunk.

  $T = 2*X$

- The tile has width of U where,

  if X is the IO cell width on the tree trunk.

  $U = X$

- Ports 2, 3, 4 and 5 have S number of connections where,

  $S >= (I-1)*P$

- Interconnect wiring 6 and 7 have length of M, where M can be any value preferably the minimum length possible for the tile.

Figure 30 is an example of a tile "9R" for illustrative purpose. Most implementations will use multiple metal layers and vias to create ports and interconnect wiring.

- The tile has height of T where,

  if X is the IO cell width on the tree trunk.

  $T = 2*X$

- The tile has width of U where,

  if X is the IO cell width on the tree trunk.

  $U = X$

- Ports 2, 3, 4 and 5 have S number of connections where,

  $S >= (I-1)*P$

- Interconnect wiring 6 and 7 have length of M, where M can be any value preferably the minimum length possible for the tile.

The same tile wiring design can be used for type 9L and 9R by using a horizontal flip.

Figure 31 is an example of a tile "10" for illustrative purpose. Most implementations will use multiple metal layers and vias to create ports and interconnect.

- The tile has height of T where,
  if X is the IO cell width on the tree trunk.
  T = 2*X
- The tile has width of U where,
  if X is the IO cell width on the tree trunk.
  U = 2*X
- Ports 2, 3, 4 and 5 have S number of connections where,
  S >= (I-1)*P
- Interconnect wiring 6 and 7 have length of M, where M can be any value preferably the minimum length possible for the tile.

tile 10 can be rotated and flipped to fit the application.

Figure 32 is an example of a tile "11" for illustrative purpose. Most implementations will use multiple metal layers and vias to create ports and interconnect wiring.

- The tile has height of T where,
  if X is the IO cell width on the tree trunk.
  T = 2*X
- The tile width is of arbitrary length. The shorter the width, the more precisely the interconnect root can place device 20 within the ASIC. Shorter width 10 tiles come at the expense of having to use many of them when device 20 must be located away form the interconnect tree. Tile 10 of multiple widths can be built to combine placement precision with reduced number of tile instantiations.
- Ports 2, 3, 4 and 5 have S number of connections where,
  S >= (I-1)*P

• Interconnect wiring 6 and 7 have length of M, where M can be any value preferably the minimum length possible for the tile.

Tile 10 can be rotated and flipped to fit the application.

All tiles, IO cells and device 20 connect by abutment. All the bus bit order is always maintained when abutting. Figures 33 to 69 illustrate various examples of how different tiles abut. The list of figures is not exhaustive, but rather a broad conceptual example of how tiles are placed together in layout.

Figure 70 illustrates the wiring layout produced for the relatively simple example of a balanced tree having a left tree section and a right tree section, each tree section having five cells associated therewith. In a typical embodiment, there will be many more cells provided along the wide interface, and Figure 70 is merely provided for the purposes of illustration. As can be seen from Figure 70, the individual wiring layouts within the various tiles provide the data paths through the interconnect and provide points of connection to each of the IO devices 32 that are to be connected into the interconnect. As shown in the example of Figure 70, there are three such IO devices 32 in the left tree section and four such IO devices 32 in the right tree section. The manner in which the wiring layouts in individual tiles can be used to provide connections to connected IO devices 32, or to avoid connections to non-connected IO devices, is shown schematically in Figure 71 and 72. As shown in Figure 71, the pin locations on the IO devices 32 that need to be connected are such that they connect with the required wiring layout on the adjacent tile, whilst those IO devices which do not need connections do not have alignment of their pins with the wiring layouts of the adjacent tile.

Similarly, in Figure 72, this shows how a combination of a 6L and a 6R tile are used to provide the required connections to the device 20.

Figure 73 schematically illustrates a general purpose computer 700 of the type that may be used to implement the above described techniques. The general purpose computer 700 includes a central processing unit 702, a random access memory 704, a read only memory 706, a network interface card 708, a hard disk drive 710, a display driver 712 and monitor 714 and a user input/output circuit 716 with a keyboard 718

and mouse 720 all connected via a common bus 722. In operation the central processing unit 702 will execute computer program instructions that may be stored in one or more of the random access memory 704, the read only memory 706 and the hard disk drive 710 or dynamically downloaded via the network interface card 708. The results of the processing performed may be displayed to a user via the display driver 712 and the monitor 714. User inputs for controlling the operation of the general purpose computer 700 may be received via the user input output circuit 716 from the keyboard 718 or the mouse 720. It will be appreciated that the computer program could be written in a variety of different computer languages. The computer program may be stored and distributed on a recording medium or dynamically downloaded to the general purpose computer 700. When operating under control of an appropriate computer program, the general purpose computer 700 can perform the above described techniques for determining routing of data paths in an interconnect circuit, and can be considered to form an apparatus for performing the above described techniques. The architecture of the general purpose computer 700 could vary considerably and Figure 73 is only one example.

In the above described embodiments of the present invention, for any block locations where wiring layout is not required, no tiles are instantiated (see for example the tiles marked as "x" in Figures 5 to 8). However, in an alternative embodiment, the predetermined set of tiles could include filler tiles used to instantiate tiles for those blocks. In one embodiment, those filler tiles will have metal fill to satisfy the metal density requirements for the given manufacturing process.

Further, in some embodiments, the IO cell libraries may have short corner IO cells, which will limit the number of branches that can be provided on the far side of the corner cell (with respect to the trunk) without layout congestion. In one embodiment, this can be alleviated by defining filler elements on either side of the corner element, and then implementing each filler element using predetermined filler tiles provided within the predetermined set of tiles. Such filler tiles in combination with the corner tile will effectively extend the size of the corner element to allow for

building tree structures that require a large number of branches on the far side of the corner element.

In the earlier described embodiments, various of the tiles have separate instantiations for every possible width of cell. However, in an alternative embodiment, the need to provide tiles for every possible width of cell can be avoided through the use of filler cells. More particularly, the filler cells can be combined with existing IO cells in order to create a combined IO cell library which will have the same width for each IO cell. This will significantly reduce the number of tile widths that need to be provided within the predetermined set of tiles in order to build the interconnect circuit.

In the earlier described embodiments, the various devices that are to be connected to the tree structure of the interconnect circuit are arranged to directly connect to the adjacent tile in the associated column of the interconnect circuit. However, in an alternative embodiment, one or more bridge tiles may be provided that provide pin mappings between pins of those devices and the wiring layouts of the tiles in the predetermined set of tiles. These bridge tiles hence convert the IO device's pin location to conform to the pin location required for abutting the IO device to the interconnect tree. These cells can also serve to separate the interconnect circuit from the devices so as to avoid accidental abutting to IO devices/components which do not require connection through the interconnect circuit. Such a bridge tile can also be used to create an IO device interface which has ports with vertical symmetry allowing the reuse of the tile wiring design for left and right trees through a horizontal flip.

From the above description of embodiments of the present invention, it will be appreciated that such embodiments have the following advantages:

1. An interconnect circuit designed using the techniques of embodiments of the present invention enables two physical blocks to be connected where there are huge size differences between their interfaces;

2. An interconnect circuit designed using the techniques of embodiments of the present invention is able to handle differential signals effectively - in particular the symmetrical wire length and parasitic matching (physical effects matching) properties

of embodiments of the present invention makes it particularly suitable for handling differential signals;

3. The technique provides an exactly matched propagation delay, and can also match other physical aspects such as coupling, cross talk, loading, noise shielding, etc;

4. The technique can handle a mixture of IO devices with different widths;

5. The technique is able to effectively handle different mixtures of IO devices connecting through the interconnect and IO devices/components not connecting through the interconnect without modifying the core interface pin and signal location;

6. The core interface can be placed anywhere in relation to the IO device placement, through the use of the earlier-mentioned root channel.

7. The technique can provide separate optimized topologies, for left, right and centred core interface placement;

8. The technique provides a structured routing method using predetermined tiles with predetermined wiring layouts;

9. It is possible to build a wide variety of different interconnects using only a small number of predetermined layout elements (tiles);

10. Interconnects built using the techniques of embodiments of the present invention have pre-characterised timing based on exact geometrical information;

11. The technique is algorithm based, and can be instantly compiled for various bus widths based on a feed forward approach. Unlike most known ASIC tools, no iterations of the process are required;

12. Interconnects designed using the technique of embodiments of the present invention typically exhibit reduced power consumption compared with those designed using prior art tools, because the present technique avoids the need to insert buffers and other components in various paths in order to improve the timing and matching of delays.

The technique of embodiments of the present invention can be applied to all high speed busses which require a high degree of timing matching when connecting them to blocks which are physically far apart. The technique of embodiments of the present invention is especially suited for connecting parallel busses from a localised

design to IO cells on integrated circuits, where the data path physical widths differ greatly in size. Memory interfaces such SDR, DDR and parallel bus architectures such as PCI are prime examples of technologies which will greatly benefit from the use of the technique of embodiments of the present invention. As the data rate of those architectures increases, the technique of embodiments of the present invention will be particularly beneficial.

Although a particular embodiment of the invention has been described herein, it will be apparent that the invention is not limited thereto, and that many modifications and additions may be made within the scope of the invention. For example, various combinations of the features of the following dependent claims could be made with the features of the independent claims without departing from the scope of the present invention.

1. A method of determining routing of data paths in interconnect circuitry for an integrated circuit, the interconnect circuitry on a first side providing a narrow interface for connection to a first device and on a second side providing a wide interface for connection to a distributed plurality of further devices, each data path being associated with one of said further devices and providing a connection through the interconnect circuitry between that associated further device and the first device, the method comprising the steps of:

(i) defining a plurality of cells to be provided along the wide interface, each of the further devices being associated with at least one of said cells;

(ii) defining the interconnect circuitry as an array of blocks formed in rows and columns, each cell abutting one of said columns;

(iii) providing a predetermined set of tiles, each tile providing a predetermined wiring layout; and

(iv) for each block, applying predetermined rules to determine one of said tiles to be used to implement that block, the predetermined rules taking into account the location of the block in the array, and the association between the plurality of further devices and the plurality of cells, and ensuring that each data path provided by the interconnect circuitry has the same propagation delay.

2. A method as claimed in Claim 1, wherein each data path is formed by a sequence of tiles, each tile in the sequence being selected to have a wiring layout that at least partially abuts with the wiring layout in an adjacent tile of the sequence.

3. A method as claimed in Claim 1, wherein:

the predetermined rules allocate tiles to individual blocks in the array so as to produce a tree structure, at least one column abutting the narrow interface being formed as a trunk of the tree structure, and the rows in the remaining columns forming branches of the tree structures;

at said step (iv), when determining tiles to be used to implement individual blocks in the array, said predetermined rules taking into account whether a block is in the trunk of the tree structure or a branch of the tree structure.

4. A method as claimed in Claim 3, wherein:

said predetermined set of tiles comprise a predetermined subset of trunk tiles and a predetermined subset of branch tiles, during said step (iv), the predetermined rules selecting, for each block, a tile from the subset of trunk tiles if the block is in the trunk of the tree structure, or a tile from the subset of branch tiles if the block is in a branch of the tree structure.

5. A method as claimed in Claim 3, wherein:

the number of cells exceeds the number of further devices to be connected to said wide interface, such that at least one non-connect cell exists that does not have one of said further devices associated therewith; and

for each column formed as a trunk of the tree structure, the cell chosen to abut that column being a non-connect cell.

6. A method as claimed in Claim 3, wherein the tree structure comprises one or more tree sections, each tree section having one column abutting the narrow interface to form a trunk for that tree section, and the rows in the remaining columns of the tree section forming branches of that tree section.

7. A method as claimed in Claim 6, wherein at least one tree section is provided on a different axis to at least one other tree section.

8. A method as claimed in Claim 6, wherein each tree section comprises either a left tree section or a right tree section, at said step (iv) the predetermined rules applied being dependent on whether tiles are being determined for blocks of a left tree section or are being determined for blocks of a right tree section, whilst still ensuring that each data

path provided by the interconnect circuitry, whether in a left tree section or a right tree section, has the same propagation delay.

9. A method as claimed in Claim 1, wherein the distributed plurality of further devices are distributed along a distribution path following at least one axis.

10. A method as claimed in Claim 9, wherein at at least one point in the distribution path a change in axis takes place, the method further comprising the steps of:

at said step (ii), defining a corner element within the interconnect circuitry in association with each point in the distribution path where a change in axis take place, such that each corner element defines a change of axis between two adjacent columns of blocks in the array;

at said step (iv), the predetermined rules implementing each corner element using predetermined corner tiles contained with said predetermined set of tiles, and when determining tiles to be used for each block in the array the predetermined rules taking into account the location of any corner elements within the interconnect circuitry so as to ensure that each data path provided by the interconnect circuitry has the same propagation delay.

11. A method as claimed in Claim 10, wherein at said step (ii) a filler element is defined on either side of the corner element, and at said step (iv) the predetermined rules implement each filler element using predetermined filler tiles contained with said predetermined set of tiles.

12. A method as claimed in Claim 1, wherein the number of cells exceeds the number of further devices to be connected to said wide interface, such that at least one non-connect cell exists that does not have one of said further devices associated therewith, and the method further comprises associating with one or more non-connect cells one or more other components of the integrated circuit not requiring connection to the interconnect circuitry.

13. A method as claimed in Claim 12, wherein said other components comprise power supply lines used to power the distributed plurality of further devices.

14. A method as claimed in Claim 1, wherein during said step (iv), if the predetermined rules determine that no wiring layout is required at the location of a particular block, no tile is allocated for that block.

15. A method as claimed in Claim 1, wherein said predetermined set of tiles include one or more filler tiles having no wiring layout, and during said step (iv), if the predetermined rules determine that no wiring layout is required at the location of a particular block, one of said filler tiles is allocated for that block.

16. A method as claimed in Claim 1, wherein at said step (i) the plurality of cells defined are of a plurality of different widths, and the predetermined set of tiles include tiles having a corresponding plurality of different widths, so as to enable each cell to abut one of said columns.

17. A method as claimed in Claim 1, wherein at said step (i) certain cells are formed by a combination of an initial cell and a filler cell, such that each cell abutting with a column in the array is of equal width.

18. A method as claimed in Claim 1, further comprising the step of:

defining a root channel passing from the narrow interface of the interconnect circuitry to the first device, thereby allowing placement of the first device within the integrated circuit at a location other than that of the narrow interface.

19. A method as claimed in Claim 18, wherein said predetermined set of tiles include a subset of root tiles, the method further comprising the step of producing the root channel using one or more root tiles from said subset.

20. A method as claimed in Claim 1, wherein the distributed plurality of further devices comprise input/ouput devices placed around the periphery of the integrated circuit.

21. A method as claimed in Claim 20, wherein the first device is a memory controller device used to control the plurality of input/output devices.

22. A method as claimed in Claim 1, further comprising the steps of:

providing one or more bridge tiles that provide pin mappings between pins of said further devices and wiring layouts of tiles in said predetermined set of tiles; and

when, for a particular further device, there is not a direct mapping of its pins to the wiring layout of the tile in the interconnect circuitry that it is to abut to, introducing at least one bridge tile between said particular further device and said tile in the interconnect circuitry to map the pins of said particular further device to the wiring layout of that tile.

23. A method as claimed in Claim 1, wherein the number of cells exceeds the number of further devices to be connected to said wide interface, such that at least one non-connect cell exists that does not have one of said further devices associated therewith, addition of such non-connect cells along the wide interface having no effect on the width of the narrow interface.

24. A method as claimed in Claim 23, wherein addition of such non-connect cells along the wide interface further has no effect on the order in which the data paths connect to the narrow interface.

25. A method as claimed in Claim 1, wherein the wiring layout of at least one of the tiles in the predetermined set is such that when that at least one of the tiles is used to

implement a block within said array, not all of the wiring layout within that tile is utilised.

26. A computer program product comprising a computer program which when executed on a computer system causes the computer system to perform a method to determine routing of data paths in interconnect circuitry for an integrated circuit, the interconnect circuitry on a first side providing a narrow interface for connection to a first device and on a second side providing a wide interface for connection to a distributed plurality of further devices, each data path being associated with one of said further devices and providing a connection through the interconnect circuitry between that associated further device and the first device, the method comprising the steps of:

(i) defining a plurality of cells to be provided along the wide interface, each of the further devices being associated with at least one of said cells;

(ii) defining the interconnect circuitry as an array of blocks formed in rows and columns, each cell abutting one of said columns;

(iii) providing a predetermined set of tiles, each tile providing a predetermined wiring layout; and

(iv) for each block, applying predetermined rules to determine one of said tiles to be used to implement that block, the predetermined rules taking into account the location of the block in the array, and the association between the plurality of further devices and the plurality of cells, and ensuring that each data path provided by the interconnect circuitry has the same propagation delay.

27. A system for determining routing of data paths in interconnect circuitry for an integrated circuit, the interconnect circuitry on a first side providing a narrow interface for connection to a first device and on a second side providing a wide interface for connection to a distributed plurality of further devices, each data path being associated with one of said further devices and providing a connection through the interconnect circuitry between that associated further device and the first device, the system comprising:

means for defining a plurality of cells to be provided along the wide interface, each of the further devices being associated with at least one of said cells;

means for defining the interconnect circuitry as an array of blocks formed in rows and columns, each cell abutting one of said columns;

means for providing a predetermined set of tiles, each tile providing a predetermined wiring layout; and

data path determining means which, for each block, applies predetermined rules to determine one of said tiles to be used to implement that block, the predetermined rules taking into account the location of the block in the array, and the association between the plurality of further devices and the plurality of cells, and ensuring that each data path provided by the interconnect circuitry has the same propagation delay.

1 Abstract

A system, method and computer program product are provided for determining routing of data paths in interconnect circuitry for an integrated circuit. The interconnect circuitry on a first side provides a narrow interface for connection to a first device, and on a second side provides a wide interface for connection to a distributed plurality of further devices. Each data path is associated with one of the further devices and provides a connection through the interconnect circuitry between that associated further device and the first device. The method comprises the steps of defining a plurality of cells to be provided along the wide interface, each of the further devices being associated with at least one of the cells, and defining the interconnect circuitry as an array of blocks formed in rows and columns, with each cell abutting one of the columns. Further, the method includes the steps of providing a predetermined set of tiles, each tile providing a predetermined wiring layout, and for each block, applying predetermined rules to determine one of the tiles to be used to implement that block. The predetermined rules take into account the location of the block in the array and the association between the plurality of further devices and the plurality of cells, ensuring that each data path provided by the interconnect circuitry has the same propagation delay. By such an approach, a structured routing method is provided that uses predetermined tiles enabling a layout design for the interconnect circuitry to be readily produced whilst ensuring that the propagation delays are matched for each of the data paths within the interconnect circuitry.

2 Representative Drawing

Fig. 4

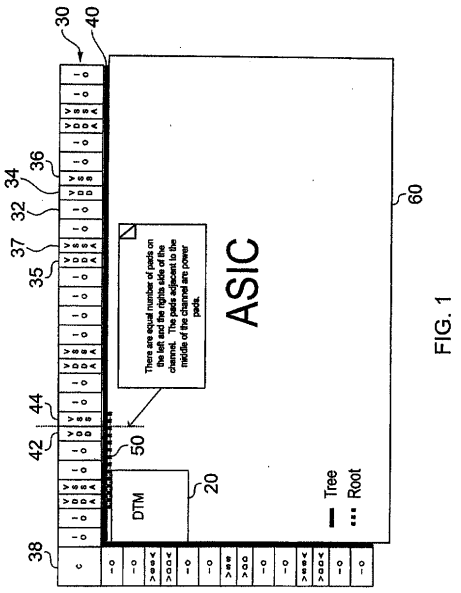


FIG. 1

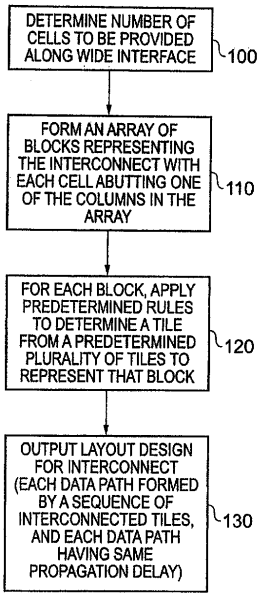


FIG. 2

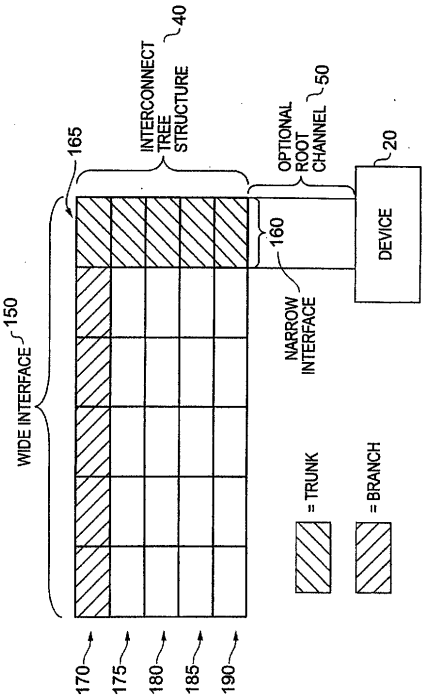


FIG. 3

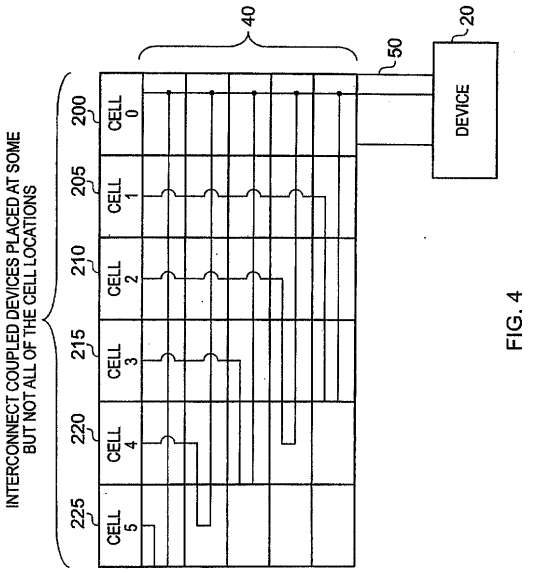


FIG. 4

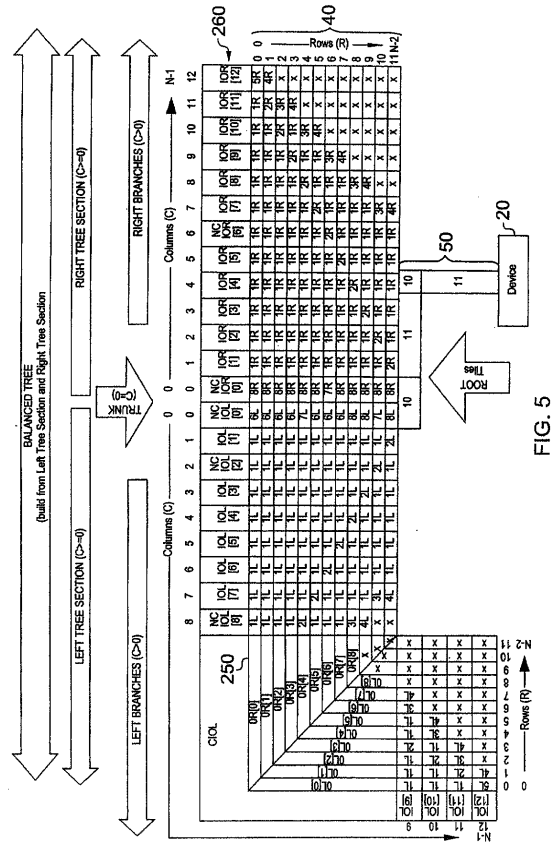


FIG. 5

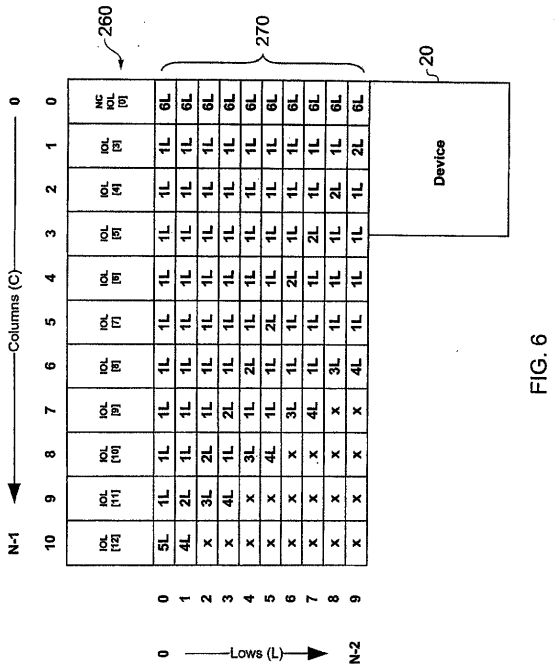


FIG. 6

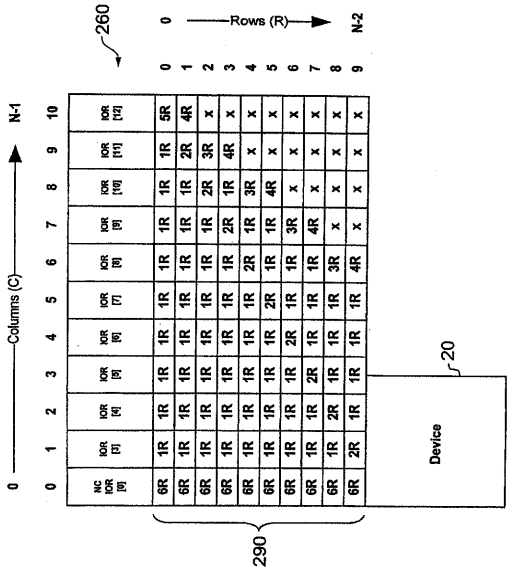


FIG. 7

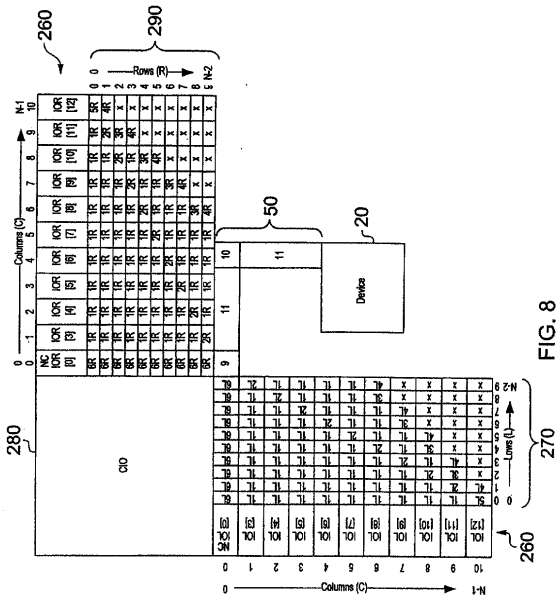


FIG. 8

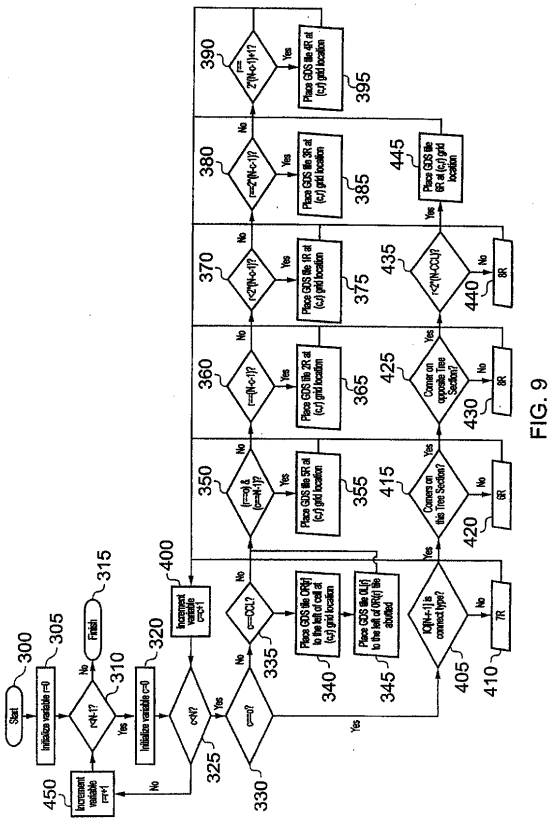


FIG. 9

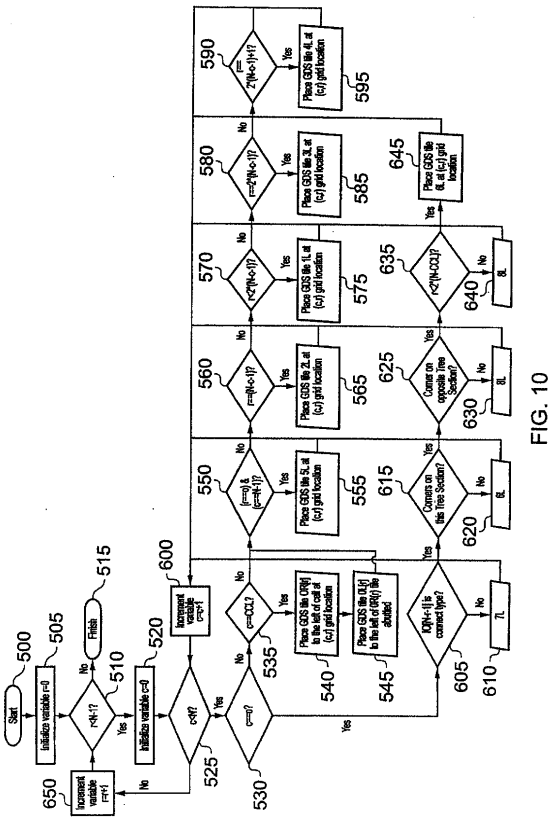


FIG. 10

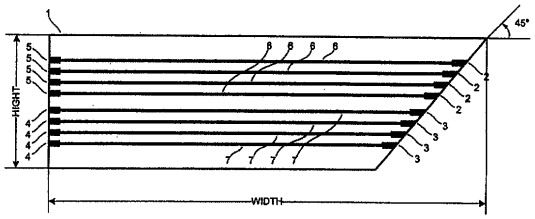


FIG. 11

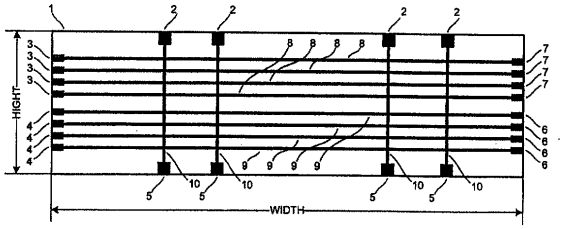


FIG. 13

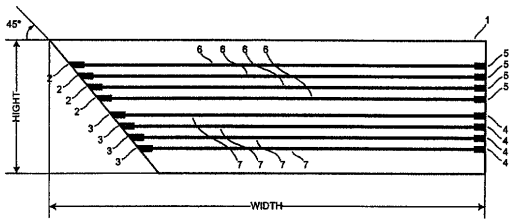


FIG. 12

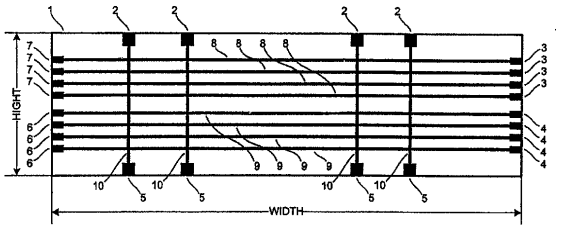


FIG. 14

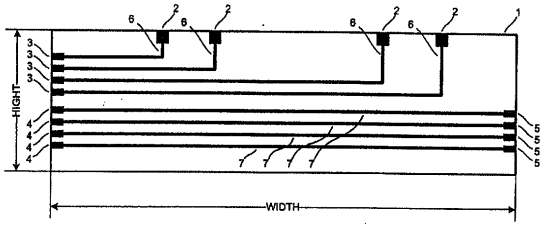


FIG. 15

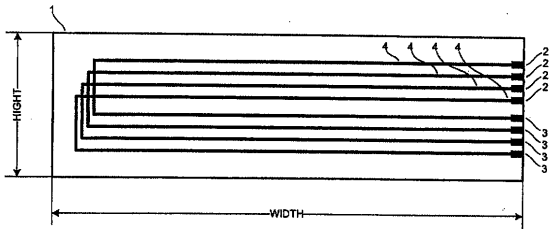


FIG. 17

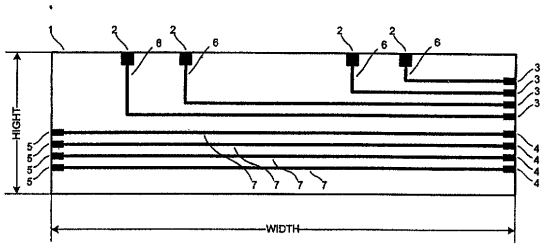


FIG. 16

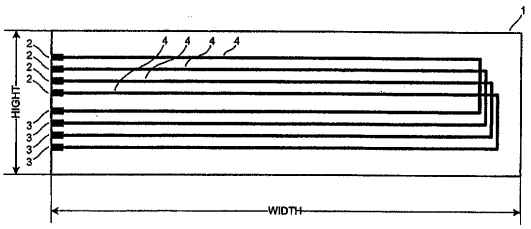


FIG. 18

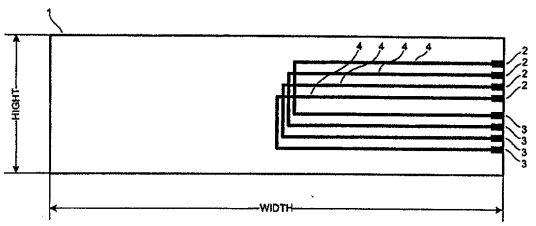


FIG. 19

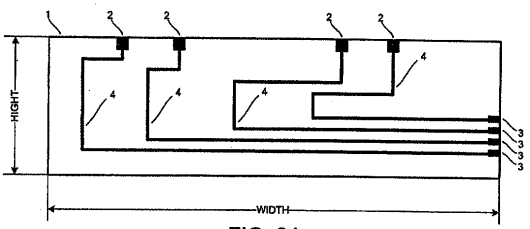


FIG. 21

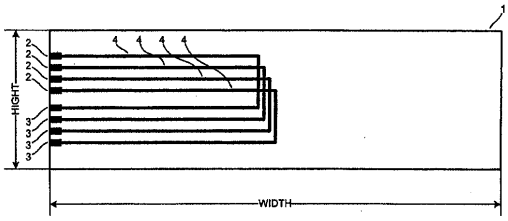


FIG. 20

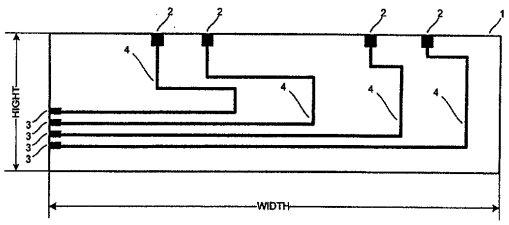


FIG. 22

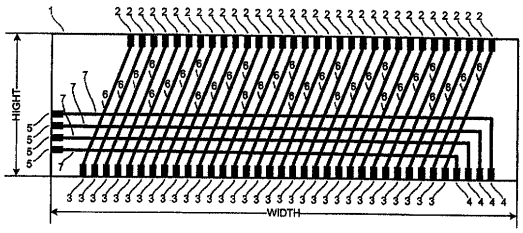


FIG. 23

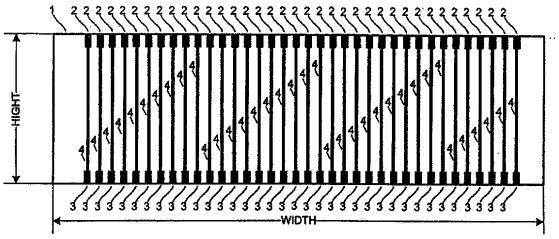


FIG. 25

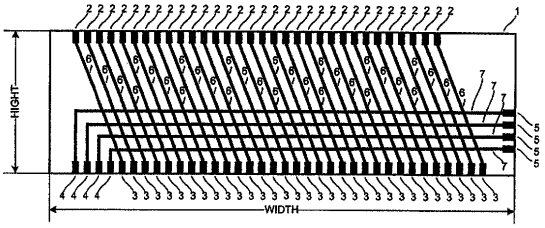


FIG. 24

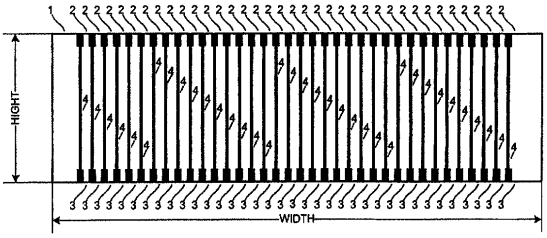


FIG. 26

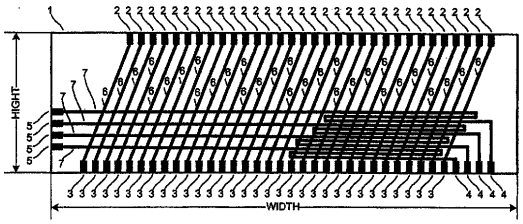


FIG. 27

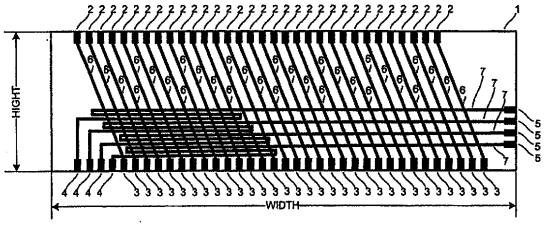


FIG. 28

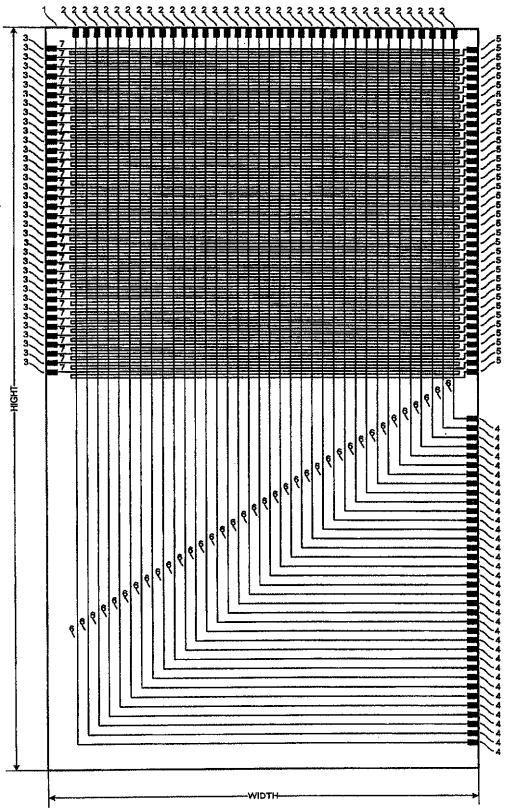


FIG. 29

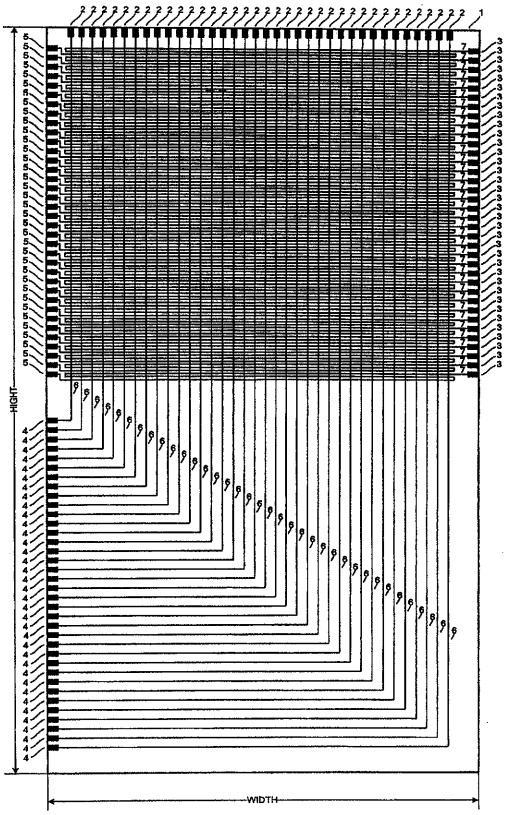


FIG. 30

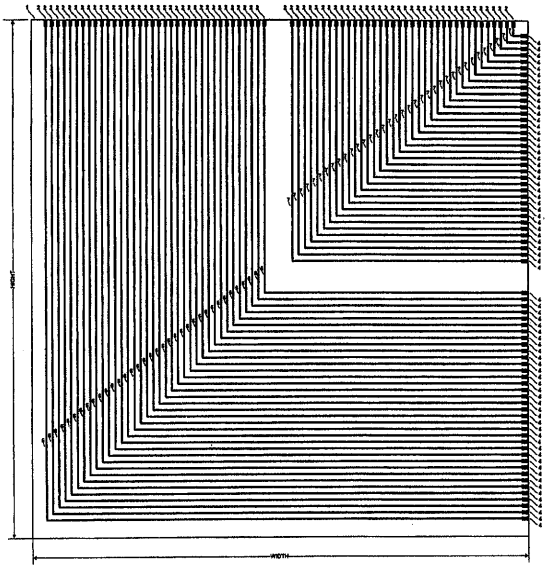


FIG. 31

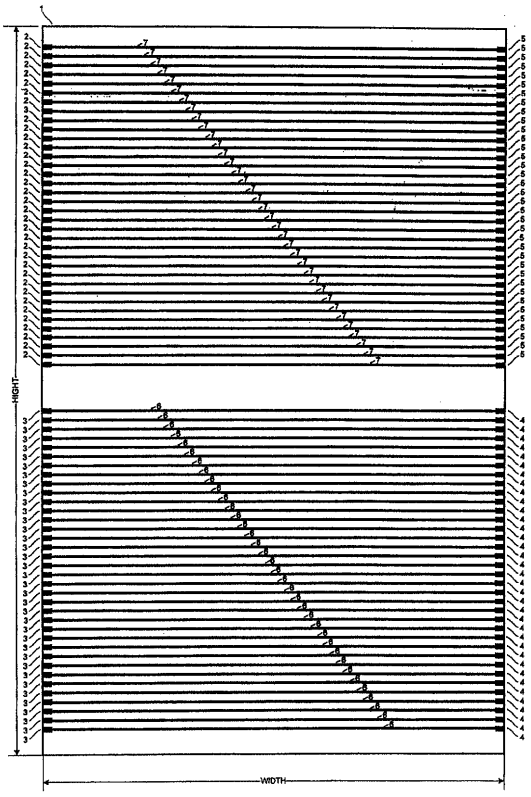


FIG. 32

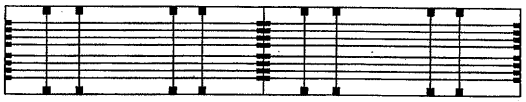


FIG. 33

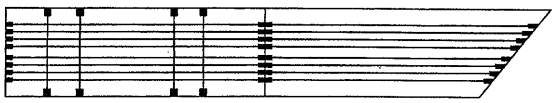


FIG. 34

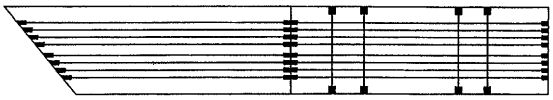


FIG. 35

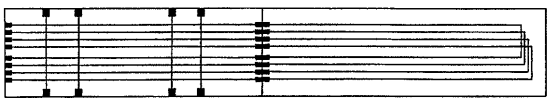


FIG. 36

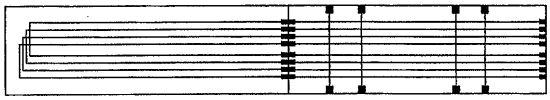


FIG. 37



FIG. 38



FIG. 39



FIG. 40



FIG. 41

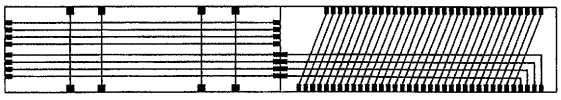


FIG. 42

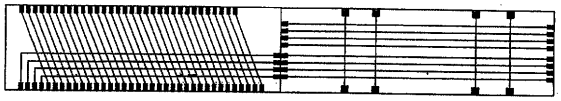


FIG. 43

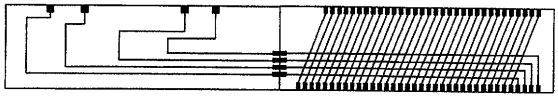


FIG. 44

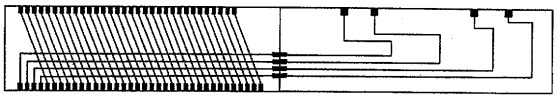


FIG. 45

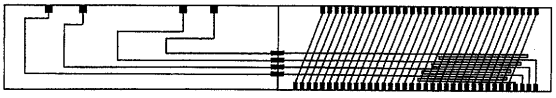


FIG. 46

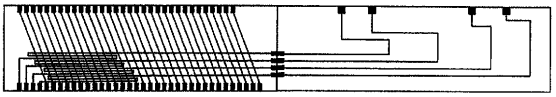


FIG. 47

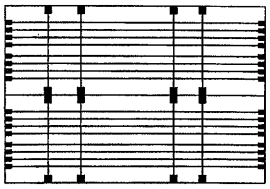


FIG. 48

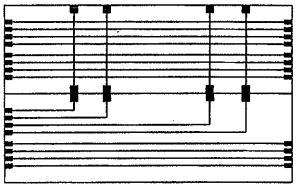


FIG. 49

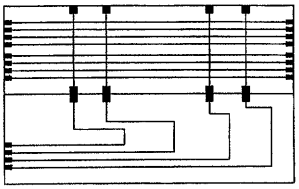


FIG. 50

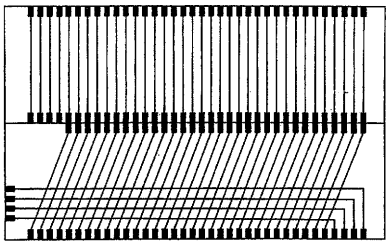


FIG. 51

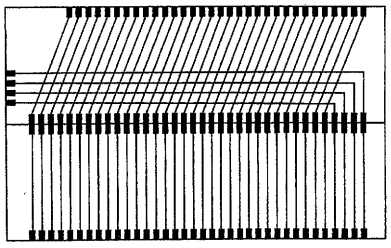


FIG. 52

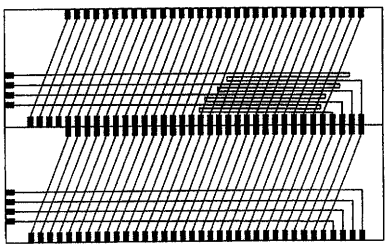


FIG. 53

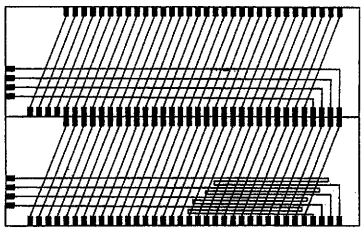


FIG. 54

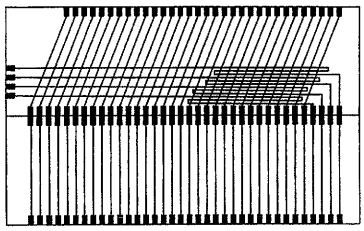


FIG. 55

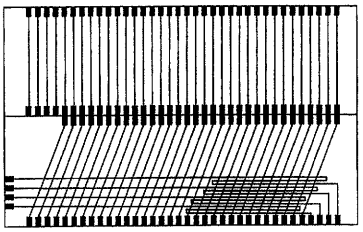


FIG. 56

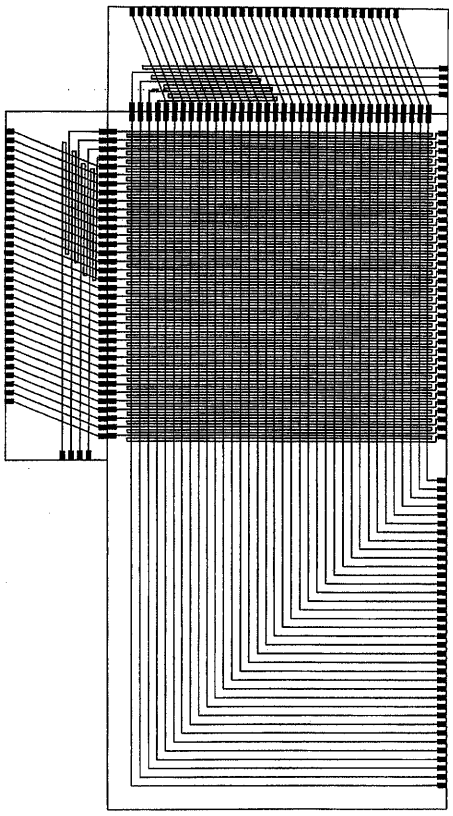


FIG. 57

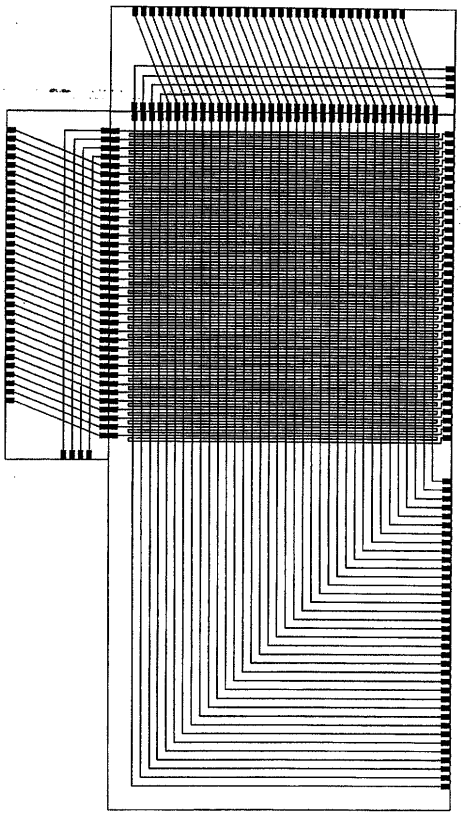


FIG. 58

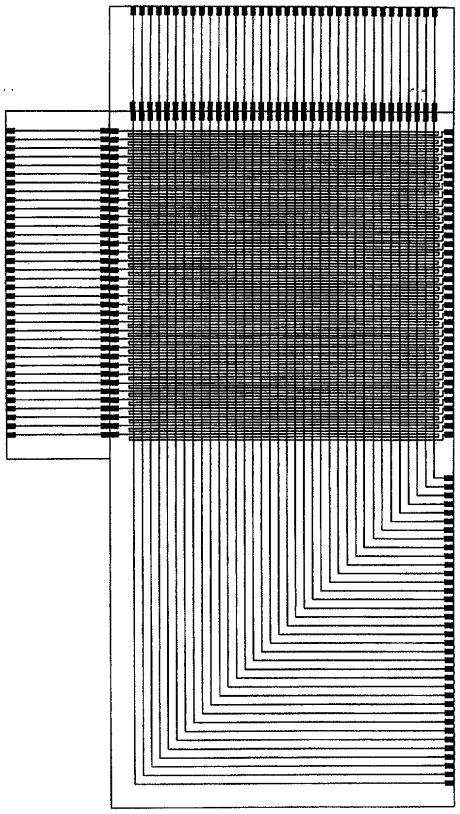


FIG. 59

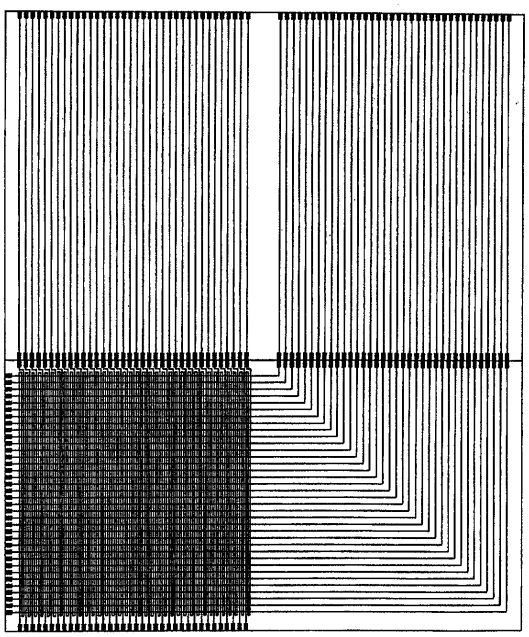


FIG. 60

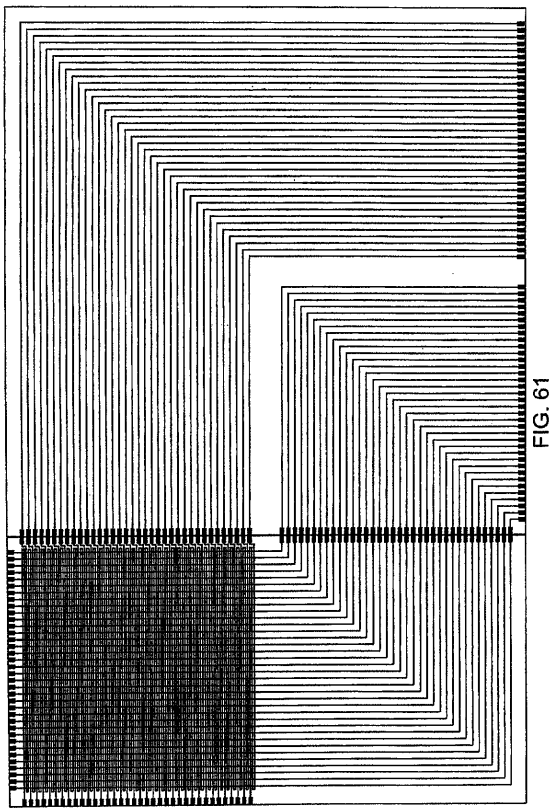


FIG. 61

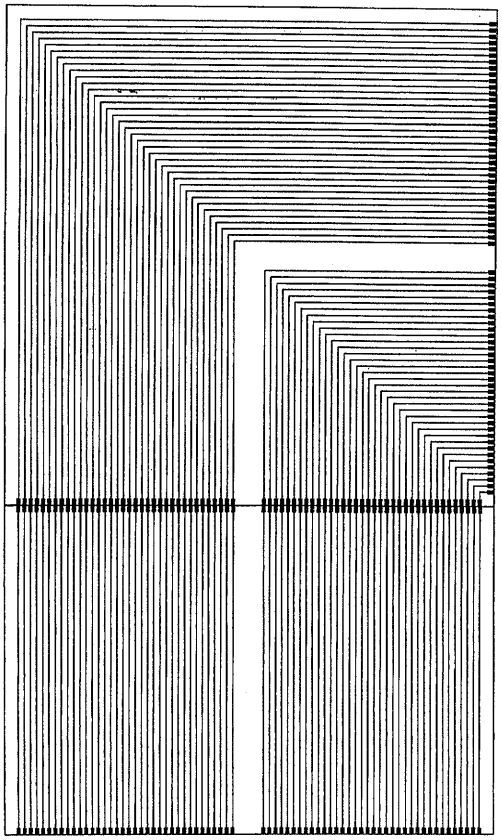


FIG. 62

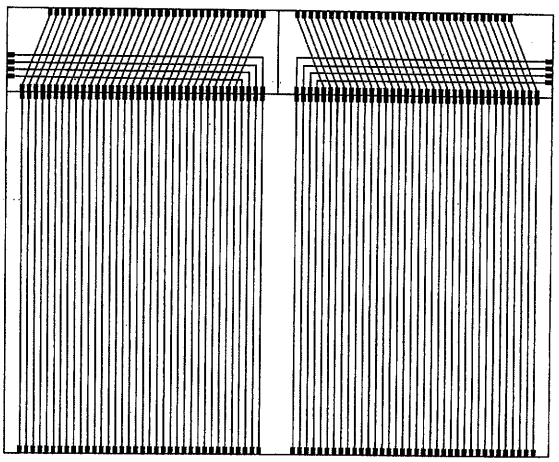


FIG. 63

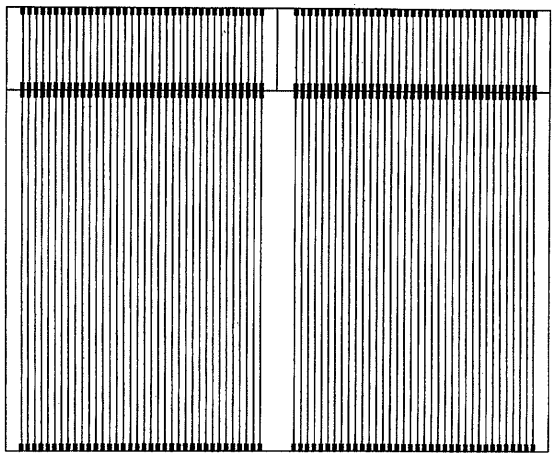


FIG. 64

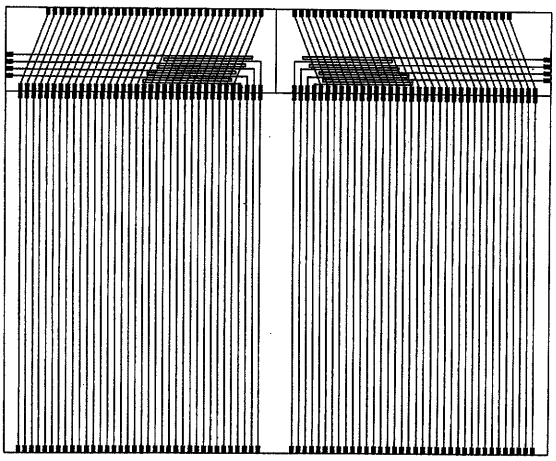


FIG. 65

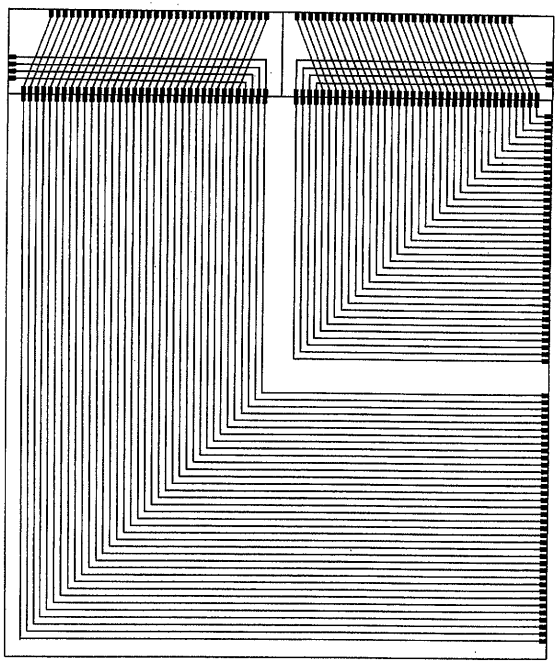


FIG. 66

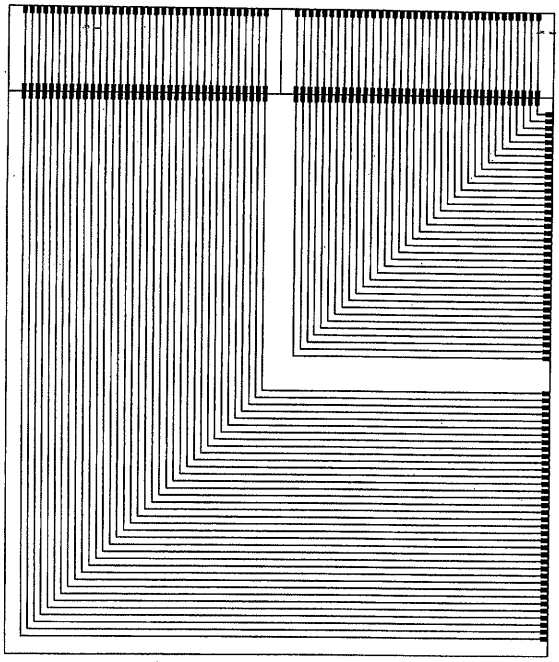


FIG. 67

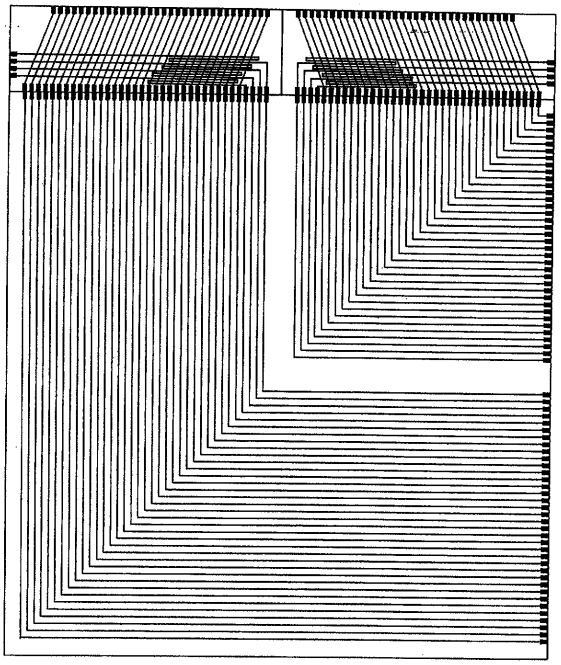


FIG. 68

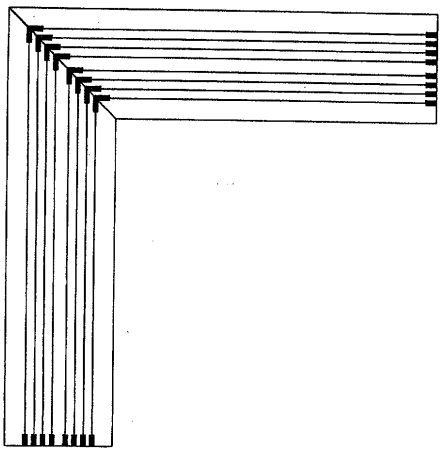


FIG. 69

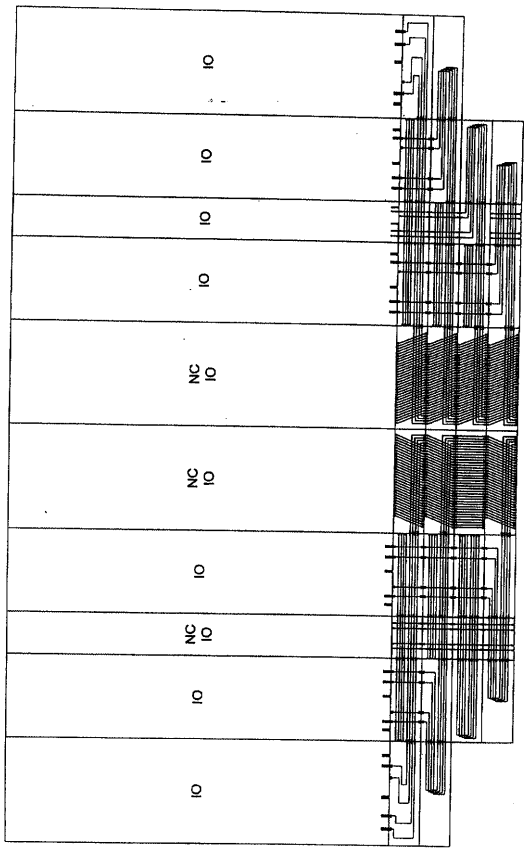


FIG. 70

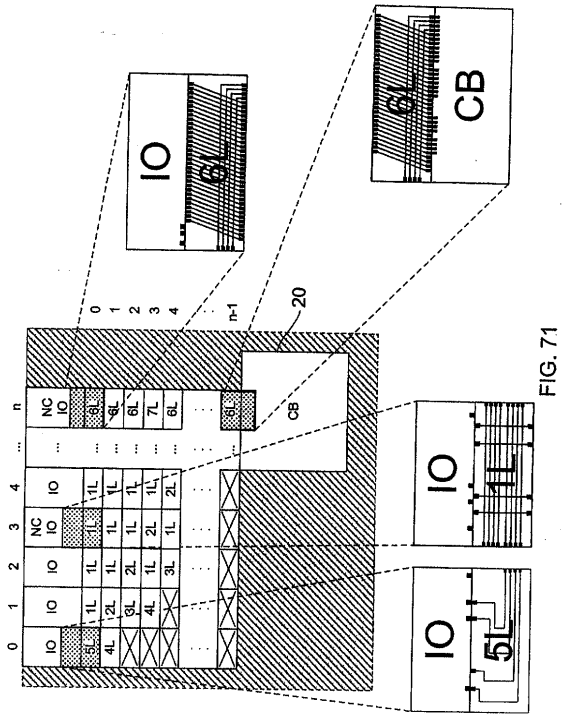


FIG. 71

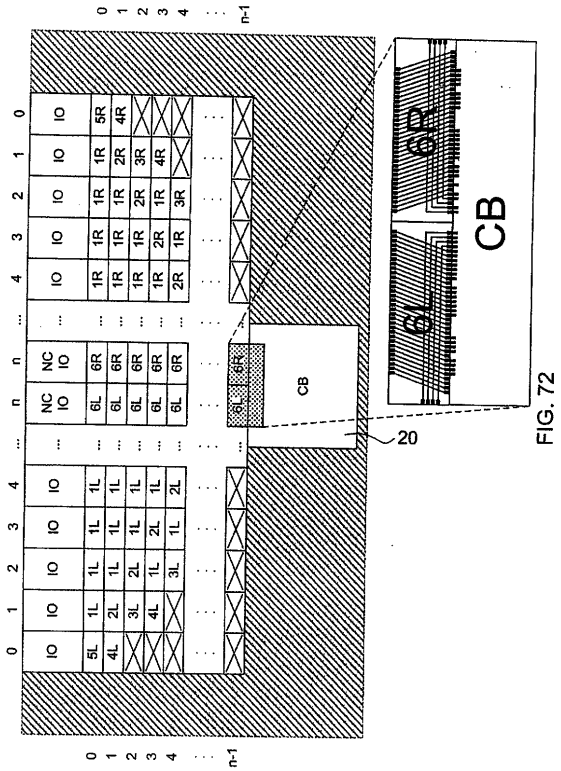

CB
CB
20

FIG. 72

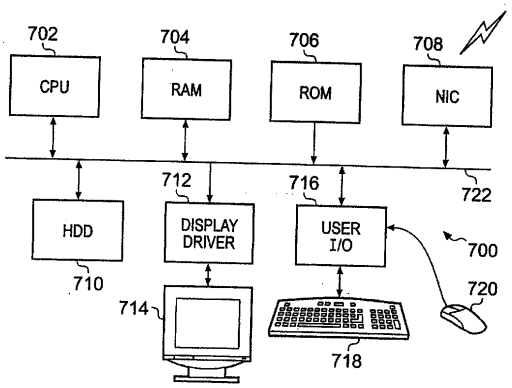

CPU
RAM
ROM
NIC
HDD
DISPLAY DRIVER
USER I/O
702
704
706
708
710
712
714
716
718
720
722
700

FIG. 73