

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第4642304号
(P4642304)

(45) 発行日 平成23年3月2日 (2011.3.2)

(24) 登録日 平成22年12月10日 (2010.12.10)

(51) Int. Cl.

F I

H03K 19/173 (2006.01)

H03K 19/173 101

G06F 17/50 (2006.01)

G06F 17/50 656A

H03K 19/177 (2006.01)

G06F 17/50 664P

H03K 19/177

請求項の数 4 (全 65 頁)

(21) 出願番号 特願2001-540820 (P2001-540820)
 (86) (22) 出願日 平成12年11月28日 (2000.11.28)
 (65) 公表番号 特表2003-516009 (P2003-516009A)
 (43) 公表日 平成15年5月7日 (2003.5.7)
 (86) 国際出願番号 PCT/IL2000/000797
 (87) 国際公開番号 W02001/039249
 (87) 国際公開日 平成13年5月31日 (2001.5.31)
 審査請求日 平成19年11月27日 (2007.11.27)
 (31) 優先権主張番号 60/167, 684
 (32) 優先日 平成11年11月29日 (1999.11.29)
 (33) 優先権主張国 米国 (US)

(73) 特許権者 502189649
 セロット・インコーポレーテッド
 アメリカ合衆国デラウェア州19810ウ
 イルミントン・レナーロード701
 (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 100108453
 弁理士 村山 靖彦
 (74) 代理人 100110364
 弁理士 実広 信哉
 (72) 発明者 オフエル, メジエド
 イスラエル・42384ネタンヤ・ジエバ
 ストリート30

最終頁に続く

(54) 【発明の名称】 汎用のハードウェアデバイス及び方法とそれと共に使用するツール

(57) 【特許請求の範囲】

【請求項 1】

汎用ハードウェアデバイス (100) に於いて、

データ記憶用の第1の複数のセルと、そして

前記複数のセルの少なくとも1組の入力及び出力と接続された第1のプログラム可能なマトリックス (109) とを具備しており、複数のハードウェア応用品は、前記セル内にデータを選択的に記憶すること及び前記セルの少なくとも1つを前記セルの少なくとも1つに接続するために前記マトリックスを選択的にプログラムすることにより実施され、

少なくとも1つの前記セルはデータ記憶用の第2の複数のセルと、そして

前記第2の複数のセルの入力及び出力に接続された第2のプログラム可能なマトリックス (102, 108) とを備えており、前記第1の複数のセル及び前記第2の複数のセルの各セルは該汎用ハードウェアデバイス全体と同様なアーキテクチャーを有してもよく、そして

前記第1の複数のセルの部分でない少なくとも1つのセルは該少なくとも1つのセルのポートを経由して直接アクセスされ得ることを特徴とする汎用ハードウェアデバイス。

【請求項 2】

請求項1の該デバイスに於いて、該第1のプログラム可能なマトリックス (109) と該第2のプログラム可能なマトリックス (102, 108) が一体化されていることを特徴とするデバイス。

【請求項 3】

10

20

請求項 1 又は 2 の該デバイスに於いて、該記憶されたデータの少なくとも幾つかはその実行中に該ハードウェア応用品により動的に変更可能であることを特徴とするデバイス。

【請求項 4】

請求項 1 から 3 の何れか 1 つの該デバイスが更に、該第 1 のプログラム可能なマトリックス (1 0 9) 及び / 又は該第 2 のプログラム可能なマトリックス (1 0 2 , 1 0 8) の制御設定をその中に記憶するためのメモリーを有することを特徴とするデバイス。

【発明の詳細な説明】

【 0 0 0 1 】

【本発明の分野】

本発明は回路の設計及びテストとデバイスアーキテクチャー (device architecture) とに関する。

【 0 0 0 2 】

【本発明の背景】

デーエスピーエス (DSPs) 及び特定用途向け集積回路 (Application Specific Integrated Circuits) { エイシック (ASIC) } を使用しての実施を含めたアイシーエス (ICs) の入った集積回路 { アイシーエス (ICs) } 及びモジュールの既知の設計及び製造過程は長い開発サイクルを要し、高価である。特に、該当応用品の長い設計、検証 (verification) そしてテストを要する開発期間の長さのために集積回路を市販するに要する期間は長い。該製品の検証及びテストの間、デバッグ及び修理を要する設計不良が検出されるかも知れず、これは開発コストと時間に大幅に付加され、設計不良が該過程の終わりで、例えば、製品発送後に検出された時は特に然りである。

【 0 0 0 3 】

市販までの時間 (time to market) はエンジニアリングコスト、市場喪失、等に影響するのでハードウェア製品は高価になる。又コストは、開発中に使用されたテスト用機器のオーバーヘッド (overhead) 、製造中に使用されたテスト機器の費用、在庫規模そして職業技術者の雇用に関するコストを帯びねばならない。

【 0 0 0 4 】

シミュレーションの精度は不足している。該応用品の望まれる機能をシミュレートするために 1 つのシミュレーションを要する一方、該実施の実行時動作 (run-time operation) をシミュレートするためにもう 1 つのシミュレーションが必要である。もし高分解能遅延 (high-resolution delays) のシミュレーションを要するならば、シミュレーションは厄介である。更に、シミュレーションは非常に遅く、実時間よりも数十万倍遅いことも多い。

【 0 0 0 5 】

既知の設計及び開発過程のなともう 1 つの欠点はプロトタイプと非エイシック電子的カード (non-ASIC electronic cards) の寸法が実際寸法を超えることである。チップ製品上のハイエンドシステムをテストする能力は集積回路の非常に高い密集度のために制限され、それは該回路内の関心の点にプローブを当てることを非常に難しくしている。高密度集積回路を含む完成製品をテストし、複雑な回路の完全な機能を検証することは同様に難しい。

【 0 0 0 6 】

応用品のより新しい技術との両立性を保証するように既設計製品を進歩したチップ生産技術へ適合させることは貧弱である (poor) 。これは該応用品開発者に打撃を与えるのみならずチップ製造者が開発者に彼等の現在の応用品を使用させながら新しいジオメトリー (geometries) を使うことを難しくさせる。

【 0 0 0 7 】

エイシックは、該応用品が専用化を正当化するに十分な量で販売される場合に、より小さい面積を使用して応用品を実施するために使用される。典型的に、該応用品は最初従来の設計方法を使用して開発され、該設計の正当性を確立した後、エイシックに変換される。これは時間を要し高価な過程である。

10

20

30

40

50

【 0 0 0 8 】

大抵のハードウェアの実施が特定応用品に専用化され拡張又は変更を受け入れないので、典型的に、応用速度は機能的柔軟性を犠牲にして該応用品をハードウェアで実施することにより高められる。

【 0 0 0 9 】

該工業に付随するなおもう1つの欠点は有資格者が比較的不足していることと、該開発サイクルが減らせるように幾人かのエンジニア間の共有に従えるように該設計及び開発をブレイクダウンすることの難しさである。

【 0 0 1 0 】

全てのこれらの制限は該開発及び製造サイクルの長さとコストの両者を増加させるよう組み合わされる。ロジック集積回路及びそれらを含むモジュールを実施するための従来の設計及び製造過程に付随する複雑さと努力を表現するために、種々の実施方法をここで説明する。

10

【 0 0 1 1 】

図1は電子的カード (electronic cards) 上の集積回路 { アイシー (IC) } チップのハードウェア実施用の従来の過程を示す流れ図である。カード開発過程7中で必要な応用を行うカードが開発される。この過程の後では、該カードの生産活動の準備が出来ている。該応用に関するアイデアと概念は応用の規定10の間に規定され、その後にインターフェースの規定12が続く。時には該インターフェースは該システム環境、例えばピーシー応用カード内で決定される。時には該カードを作る企業が該インターフェースを選ぶことが出来て、例えばカードで一杯のラック内では、大抵のカードは同じインターフェースを有する。時には該インターフェースはユニークであり、該カードの残りと同じ過程で設計されねばならない。

20

【 0 0 1 2 】

技術選択16は実時間のニーズ、柔軟性、寸法そしてタスク数 (number of tasks) に影響される。ハードウェアでの実施はその優れた実時間性能のための応用品に選択される。ハードウェアでの実施は最も速い (実時間) 解決策であるが、限られた数のタスクしか行えず、柔軟でなくそして大型になる。市販までの時間は長い。開発から製造までの過程は、数ヶ月から1年より長くまで、を要し、9ヶ月は非常に良い結果と考えられる。該開発及び製造過程は高価である。

30

【 0 0 1 3 】

該応用に利用可能な空間は該技術選択過程に重要なパラメーターである。普通は小さい程良い。もし寸法が重要なら、該カードはエイシックに変換されるがその場合は開発期間はエイシック変換のニーズのために数ヶ月延長される。9ヶ月が普通の延長期間である。実際、エイシック変換は上記の全ての問題をこうむる。

【 0 0 1 4 】

大抵の場合、複雑で、高速度の、デジタルハードウェア設計用には特に、該応用品のシミュレーション18が必要である。時には、該シミュレーションは該応用品を規定する過程の部分と考えられ、該ハードウェア開発過程の時間の部分としてはカウントされない。該シミュレーション過程の期間は、該応用の複雑さにより、数日から2, 3ヶ月続くかも知れない。該シミュレーションの目的は、考えられたアイデアと手段 (implementation) が実施可能であることを検証することである。例えば、もし通信リンク上で音声 (voice) を圧縮 (compress) するアイデアがあるとすれば、該圧縮アルゴリズムは "C++" の様な高級言語 (high level language) 及び "マットラブ (Matlab)" { マサチューセッツ州、ナチック、マスワーク (Mathworks, Natick, Massachusetts) の } の様な他のソフトウェアツールを使用して作られる。該シミュレーションと該手段の間に簡単な関連は無い。時には1つ以上のシミュレーションが行われ、該シミュレーション速度を増加させるためにワークステーション (workstation) が必要かも知れない。

40

【 0 0 1 5 】

次いで該インターフェースが設計されねばならない20。もし該インターフェースが例え

50

ばピーシーカードのピーシーアイインターフェースの様に、標準的であるならば、大抵の場合にそれを再設計する必要は無く、”在庫の(off the shelf)”既製品チップセット(ready-made chip set)が該実施で使用されてもよい。もし該インターフェースが固有(proprietary)であるならば、該ハードウェア開発者は、該インターフェースを、同じラック内の2, 3のカード用インターフェースの様な、他の応用品用にも同じく使用出来るように、該インターフェースの開発を該応用品の開発から分離するのが好ましい。この様な場合、市販までの時間は該インターフェースを開発し、実施するのに必要な時間により影響されないであろう。該インターフェースがユニークである時だけ、全体を実施する開発の部分となる。

【0016】

10

電子的設計22は該解決策を電子的集積回路デバイスのセットに変換する手順である。該開発者は或る関数を実行出来るこの様なデバイスのライブラリーを覚えていて、該応用を実施するために一緒に接続されるべき必要なアイシーエス(ICs)を選ぶ。該エンジニアは多数の異なる部品とそれらの使用法を記憶せねばならないので、該電子的設計過程は複雑である。該技術が改善されると、複雑な機能を有する部品が追加される。時には、該電子工学技術者は後に取り残される。高級設計言語(High-Level Design Languages)が実施されたが、該設計のなお大部分はモジュール的仕方で行われねばならない。該応用が複雑な程該設計に必要な時間は長い。

【0017】

図面は該応用技術者と該カードを作るために使用されるコンピュータ化ツール(computerized tool)の間のインターフェースである。これは該技術者が彼の実施アイデアを”書く(write)”ために使用する”言語(language)”である。図面の過程24は該設計を略図図面に変換する。該応用が複雑な程、該図面に必要な時間は長い。もしプログラム可能なデバイスが使用されれば、高級設計言語(High-level Design Language){エイチデーエル(HDL)}は該図面の1部を置き換えるため使われてもよい。

20

【0018】

実施が些細(trivial)でない時、該設計のシミュレーション25が行われ、タイミングがチェックされる。該設計者は該応用品のシミュレーション18の結果と該現在のシミュレーションとの間の相関を取るよう試みる。もし何等かの誤りが見出されるならば、該設計は修正され、過程22と24は繰り返される必要がある。

30

【0019】

該設計が仕上げられた後、部品26が得られる。時には、特定のデバイスを購入することは極めて長い時間を取り、それによりプロトタイプ製作を遅延させる。該応用品が複雑な程、より多くの部品を要し、製作までの時間は長く、そして部品の在庫は大きくなる。

【0020】

レイアウト28は該製図された設計を製造されるパッケージに変換する過程である。応用品が複雑な程、より多くの部品が使用され該レイアウト時間は長くなる。

【0021】

該基板が作られる30.各応用品用に異なる基板が作られねばならず、かくして投資される人的資源量と、カードの検証及び製造用の高価な機器のニーツとを増加させる。

40

【0022】

該部品は該製造されたカード上に設置される。該レイアウト内の欠陥は該設置で問題を引き起こす。例えば、該プロトタイプを設置するためのツールは製造用に使用されるそれらと普通異なる。特に、開発フェーズ(development phase)での設置はより少ない自動化を使用し、欠陥のあるカードの確率は大きい。該設置期間を短くするために、専用の、高価な器械、例えば、ハイピッチチップ挿入器械(high pitch chip insertion equipment)、が使用される。

【0023】

テスト及びデバッグ34は全体の開発過程の中で最も長い期間である。複雑な設計に費やされた時間の50から75%が検証(verification)に用いられる。検証は急速に、最大

50

の技術的障碍に成りつつある。

【 0 0 2 4 】

上記タスクの各 1 つで誤りが起こり得る。例えば、印刷回路のトラック間の狭い間隔は短絡を引き起こし得る。もしこの種の誤りが該デバッグ過程で発見されなければより悪くなり、何故ならば、後刻の、修理がより高価な時になって、見出されるかも知れぬからである。又誤りはカード規定や、その後も起こり得る。それは繰り返し過程が起こることを意味する。第 1 バッチの生産の前に 3 つのバージョンのプロトタイプを持つことが普通である。該電子的カードをテストするために高価なテスト機器、例えば、信号発生器 (signal generators)、雑音発生器 (noise generators)、ロジックアナライザー、オシロスコープ、デシベルメーター、加算器 (adders)、ラインシミュレーターその他が必要である。

10

【 0 0 2 5 】

アールアンドデー (R&D) から生産までの過程 3 6 では、該カードを製造するために要する全ての詳細を伴うドキュメンテーション (documentation) が創られる。この過程は該プロトタイプの最終バージョンが準備される前にスタートすることが出来るが、該過程は市販までの時間を延長させる。カード検証 { 釘のベッド (nails bed) の様な } 及び機能検証用の自動的ツールが創られる。自動部品挿入機がプログラムされること等が引き続く。

【 0 0 2 6 】

上記過程は該開発及び製造サイクルを長くさせる。誤りが見出されるのが遅い程修理は難しく、高価になる。従って、もしエイシック (ASIC) を要するなら、無謬の結果を保証するために可成りの努力が行われる。該電子部品をエイシックに変換する前に 2 , 3 バッチを製造するのが普通である。

20

【 0 0 2 7 】

一旦該カードのドキュメンテーションの準備が完了すると、全ての部品が購入され、チップ挿入機がプログラムされる等と進み、該カード生産過程 3 7 が始まることが出来る。次いで生産カード検証 3 8 が続くが、そこではその電子部品を有する又は有しない該カードがテストされる。機能検証 4 0 は該設計された応用品用に該カードをテストする過程である。該応用品の各機能をテストするための自動器械を創ることは極めて複雑で時間がかかる。該結果が満足な時、クライアントへの該カードの発送 4 2 が行われてもよい。

30

【 0 0 2 8 】

もし誤りが見出されるか又は該過程の終わりで増強部 (enhancement) が必要となるならば、修理又は増強 4 4 は非常に難しく、高価でそして時間が掛かる。最悪の場合、該過程の大部分は繰り返されねばならない。

【 0 0 2 9 】

上記の様に、実時間のニーズ、柔軟性、タスクのサイズと数は技術の選び出しに影響する。柔軟性が必要とされそして / 又は多数のタスクが行われるべきであるが、同時にではない時は、回路はデジタルシグナルプロセサーとして実施されてもよい。この解決策は等価なハードウェア手段より実時間では遅い。該デーエスピー手段は該ハードウェア手段と概略同じ寸法であるが、一般にデーエスピー手段はエイシックへの変換のオプションを可能にしない。

40

【 0 0 3 0 】

上記説明のハードウェア開発過程は又該デーエスピーカード用に実施されねばならないが、それは幾つかの理由で該市販までの時間の期間 (the time to market period) には稀にしか影響しない。第 1 に、該ハードウェア手段は簡単でありそれは該デーエスピーベンダーが該ハードウェア設計用の解決策を提供するからである。非標準のインターフェースだけが設計される必要がある。第 2 に、一旦カードが準備され該インターフェースが固定されると、該新しい応用品に既製品のカードが使用出来る。普通該プログラム用コードを開発する過程は該デーエスピーカードを開発する過程より多くの時間が掛かる。それにも拘わらず、該カードは少なくとも 1 回開発されねばならない。どの種類のデーエスピーを

50

選ぶか、該デーエスピーは次世代応用品のニーズを満足させようとしているか等の様なハードウェア手段に関するそれらとは異なる問題も取り組まれねばならない。

【 0 0 3 1 】

該産業は、特徴寸法 (feature size) (トランジスター及び相互接続) を収縮 (shrinking) させることにより 1 8 ヶ月毎にダイ (die) 内のトランジスター数が 2 倍になるゴードンムーアの法則 (Gordon Moor's Law) に従う (adheres) ので、製品は各新しい半導体世代と共に古くさくなりつつある。従って、新しいデーエスピー / シーピーユー開発は頻繁に必要な。非常に共通してベンダーはソフトウェア両立性を可能にするため最善を尽くすが、実際には変換時間が必要である。この変換過程は市販までの時間のみならずコストにも影響する。

10

【 0 0 3 2 】

デーエスピーでの実施はシミュレーション段階でハードウェアでの実施より有利であるがそれは該シミュレーションも該実施も両者共 " C " の様な高級言語で書かれ得るからである。実際には、該シミュレーションからデーエスピーコードへの変換の効率的手段はない。換言すれば、該シミュレーションは、特にもしアッセンブリ言語 (assembly language) が該デーエスピーコーディングで使用されれば、該精確な実施をシミュレートしない。又、アールアンドデーのコストは高い。該コスト計算は該カードの開発と該ソフトウェアの開発を考慮せねばならない。該市場ではデーエスピーエキスパート (DSP experts) が不足しているので該賃金費用が高い。

20

【 0 0 3 3 】

生産では、自動的機能検証は実施がなお難しく、只し発送後誤り又は増強部 (enhancement) が発見されるならば、実際はこれは些少 () からは程遠いが、顧客の構内 (at the customer's premises) でソフトウェアを変更することによりなお固定され得る。容易な部分は該実施品内へ改訂コードをロードすることである。

【 0 0 3 4 】

時には組合された実施が好ましい。例えば、もし該デーエスピー実施品内にフィルターが必要ならば、該フィルターはハードウェアで実施され、該応用品の残りは該デーエスピー内で実施される。各部分の有利な及び不利な点が留まっている。

【 0 0 3 5 】

プログラマブルロジックデバイス (Programmable Logic Devices) { ピーエルデーエス (PLDs) } は柔軟な実施を許容するが、与えられたチップ面積に対して応用能力で制限される。シミュレーション言語は設計者に実施を創生させる高級設計言語 (High-level Design Language) { エイチデーエル / ブイエイチデーエル (HDL/VHDL) } に変換される。それにも拘わらず、これらのソフトウェア言語はユーザーにモジュール形式で該ハードウェアを創生させるので、それらは C + + に似た言語より遙かに後れている。シミュレーションは不正確で、デバッグは複雑でそして製品は高価である。寸法が重要な時は、2 , 3 のハイエンドピーエルデーを " 促成プロトタイプ (fast prototype) " として使って該応用品を実施しそして次いで該応用品をエイシックに変換することが最も普通である。この場合、該エイシック開発用に 2 , 3 回の繰り返しを行うのが普通であり、それは価格と市販までの時間を増加させる。

30

40

【 0 0 3 6 】

該エイシック開発過程は、例えば、そのウェブアドレスが <http://www.ti.com/sc/docs/asic/cad/cad.htm> であるテキサスインスツルメント社 (Texas Instruments Incorporated) 、 { 米国、テキサス州、ダラス、7 5 3 8 0 - 9 0 6 6 (Dallas, Texas USA, 75380-9066) } により説明されている。

【 0 0 3 7 】

又、米国、カリフォルニア州、マウンテンビュー、のベリシテイデザイン社 (Verisity Design, Inc. Mountain View, CA USA) に属する <http://www.verisity.com/html/spechase.d.html> を参照する。同様に、電子的設計オートメーション (Electronic Design Automation) に関する更に進んだ情報は <http://www.wsdmag.com/library/penton/archives/wsd/Ja>

50

nuary1998/261.htmを参照することにより見出されるが、それは電子的設計オートメーション { イーデーエイ (EDA) } 技術が半導体製作の進歩の速さに対し後れたことを認めている。

【 0 0 3 8 】

該設計、製造そして検証過程に付随する欠点の幾つかは特許文書で述べられて来た。1998年9月29日刊行でアルテラ社 (Altera Corporation) に譲渡された " きめの粗いルックアップテーブルアーキテクチャー (Coarse-grained look-up table architecture) " の名称の米国特許第5、815、726号 { クリフ、リチャード ジー (Cliff; Richard G.) } はプログラム可能なロジックデバイスのアーキテクチャーを開示している。ロジック配列ブロックへのそしてそれからの相互接続信号用に、大域的相互接続源 (global interconnection resources) がスイッチボックス、長いライン、2重ライン、単一ライン、そして半分の及び部分的に密度を持つマルチプレキサー領域 (half- and partially populated multiplexer regions) を有する。該ロジック配列ブロックは2レベルの機能ブロックを有する。第1レベルには、8つの4入力機能ブロックがある。第2レベルには、2つの4入力機能ブロックと4つの第2の2入力機能ブロックがある。1実施例ではこれらの機能ブロックはルックアップテーブル { エルユーテーエス (LUTs) } を使用して実施される。該ロジック配列ブロックは組合せの、かつ、レジスターされた出力を有し、又シーケンシャルの又はレジスターされたロジック機能を実施するための記憶ブロックを含む。該ロジック配列ブロックはキャリービット (carry bits) を要するロジック機能を実施するためにキャリーチェーン (carry chain) を有し、又ランダムアクセスメモリーを

10

20

【 0 0 3 9 】

1999年6月1日に刊行されアルテラ社に譲渡された " デューター及びテスター間のピン接続をリコンフィギュアするツール (Tool to reconfigure pin connections between a DUT and a tester) " の名称の米国特許第5、909、450号 { ライト; アダム (Wright; Adam) } は集積回路のテストをシミュレートする方法を開示している。種々のダウンボンド (downbond) 用のテスターユニットとテスト中のデバイス { デューター (DUT) } との間の望まれる接続のデータベースが望ましい接続をセットアップするマルチプレキサーによりアクセスされる。該システムは、従来のシミュレーターシステムで要求されたユーザーからの手動式介入を要せずに、各ダウンボンド用の正しい接続を自動的に作る。

30

【 0 0 4 0 】

1998年10月13日刊行で、アルテラ社に譲渡された " 第2信号に対する入力の完全並べ換え可能性を有するルックアップテーブルベースのロジック要素 (Look-up table based logic element with complete permutability of the inputs to the secondary signals) " の名称の米国特許第5、821、773号 { ノーマン; ケビン エイ他 (Norman; Kevin A. et al.) } はプログラム可能なロジックデバイス用のロジック要素を開示している。該ロジック要素はロジック機能、プログラム可能な遅延ブロック、ラッチ又はフリップフロップとしてコンフィギュア可能な記憶ブロック、そして診断シャドーラッチ (diagnostic shadow latch) を実施するためのルックアップテーブルを含む。該記憶ブロックの第2機能を制御するために該ロジック要素への複数の入力及びこれらの入力の補数 (complements) が利用可能である。

40

【 0 0 4 1 】

2000年1月25日に刊行されアルテラ社に譲渡された " プログラム可能なロジック配列集積回路 (Programmable logic array integrated circuits) " の名称の米国特許第6、018、490号 { クリフ; リチャード ジー他 (Cliff; Richard G. et al.) } は、複数のロジック配列ブロック内に一緒にグループ化された多数のプログラム可能なロジックモジュールを有するプログラム可能なロジック配列集積回路を開示している。該ロジック配列ブロックは2次元配列内の該回路上に配置される。どのロジックモジュールもどれかの他のロジックモジュールに相互接続するよう導体ネットワークが提供される。加えて

50

、該全体的相互接続ネットワークを使用する必要なしに、ロジックモジュール間のキャリアチェーンを提供する様な特殊な目的用に及び/又はより複雑なロジック機能を提供するために2つ以上のモジュールと一緒に接続するために、隣接又は近隣ロジックモジュールは相互に接続可能である。クロック及びクリヤ信号の様な広く使用されるロジック信号を該回路全体に配分するためにいわゆる促成の又は汎用の導体のもう1つのネットワークが提供される。信号導体間に必要なプログラム可能な相互接続の数を減らすためにマルチプレキサーが種々の仕方で使用出来る。

【0042】

2000年5月2日に刊行されクイックターンデザインシステム社(Quickturn Design Systems, Inc.)に譲渡された"エミュレーション及びシミュレーションを使用する設計検証のための方法と装置(Method and apparatus for design verification using emulation and simulation)"の名称の米国特許第6、058、492号{サンプル; ステファン・ピー他(Sample; Stephen P. et al.)}はロジック設計のエミュレーション及びシミュレーションを組み合わせるための方法と装置を開示している。該方法と装置は、ゲートレベル記述(gate-level description)、動作表現(behavioral representation)、構造表現(structural representation)、又はそれらの組合せを含むロジック設計で使用され得る。該エミュレーション及びシミュレーション部分は該2部分間でのデータ転送時間を最小化する仕方で組み合わせられる。シミュレーションは1つ以上のマイクロプロセッサにより行われる一方エミュレーションはフィールドプログラマブルゲートアレイ(field programmable gate arrays)の様なリコンフィギュア可能な(reconfigurable)ハードウェアで行われる。多数のマイクロプロセッサが使われる時、該ロジック設計の独立部分は該多数の同期したマイクロプロセッサ上で行われるよう選択される。又リコンフィギュア可能なハードウェアは該シミュレーションを助け、そして処理時間を減じるためにイベント検出及びスケジュール動作を行う。

【0043】

1998年9月29日に刊行されモトローラ社(Motorola, Inc.)に譲渡された"ハードウェア及びソフトウェア部品を有する製品の設計方法とそれ用の製品(Method for designing a product having hardware and software components and product therefor)"の名称の米国特許第5、815、715号{ク・セディラ・ユーケー他(Ku.cedilla.uk et al.)}は計算システムとハードウェア及びソフトウェア部品を使用して該計算システムを設計する方法とを開示している。該計算システムは同じアーキテクチャーのスタイルを有するプログラム可能なコプロセッサ(coprocessor)を備える。各コプロセッサはシーケンサーとプログラム可能な相互接続ネットワークと可変数の機能ユニット及び記憶要素とを有する。該計算システムは、応用ソフトウェアコードの部分からホストマイクロプロセッサ(host microprocessor)コードを、そして該応用ソフトウェアコードの部分からコプロセッサコードを、発生するコンパイラを使用することにより設計される。該コンパイラは該ホストマイクロプロセッサの実行速度を決定するために該ホストマイクロプロセッサコードを、そして該コプロセッサの実行速度を決定するために該コプロセッサコードを、使用しそして該応用ソフトウェアコードの部分の実行用に該ホストマイクロプロセッサか又は該コプロセッサの1つを選択する。次いで該コンパイラは該ソフトウェアプログラムとして役立つコードを創る。

【0044】

2000年5月2日に刊行されアルテラ社に譲渡された"プログラム可能なロジックデバイス内でカム又はラムとしてコンフィギュア可能なメモリーセル(Memory cells configurable as CAM or RAM in programmable logic devices)"の名称の米国特許第6、058、452号{ランガサイー; クリシマ(Rangasayee; Krishna)}はコンテンツアドレス可能なメモリーを有するプログラム可能なロジックデバイスを開示している。該プログラム可能なロジックデバイスは第1モードではコンテンツアドレス可能なメモリーとして、そして第2モードではランダムアクセスメモリーとしての動作に好適なりコンフィギュア可能な2重モードメモリーを有する。ユーザーにコンテンツアドレス可能なメモリー

10

20

30

40

50

か又はランダムアクセスメモリーか何れかとして選択的に該２重モードメモリーをコンフィギュアさせるためにモード制御スイッチ回路が提供されてもよい。

【００４５】

２０００年６月２０日に刊行されザイリンクス社（Xilinx, Inc.）に譲渡された”動的にリコンフィギュア可能な計算のためにエフピージーエイエスを設計する方法（Method of designing FPGAs for dynamically reconfigurable computing）” {グッシオーネ；ステイブン エイ．（Guccione; Steven A.）}の名称の米国特許第６、０７８、７３６号はリコンフィギュア可能なコプロセッサ応用品用のソフトウエア環境を含むリコンフィギュア可能な計算のためのエフピージーエイエス（FPGAs）の設計法を開示している。この環境は標準高級言語コンパイラ {すなわちジャバ（Java）}及び１セットのライブラリーを含む。該エフピージーエイはホストプロセッサから直接コンフィギュアされ、コンフィギュレーション（configuration）、リコンフィギュレーション（reconfiguration）そしてホストの実行時動作（host run-time operation）はコードの単片（a single piece of code）でサポートされる。秒の桁の設計コンパイル時間とパラメータ化されたセル用に組み込まれたサポートとが該発明の方法の顕著な特徴である。

10

【００４６】

それぞれ２０００年２月２９日及び２０００年８月１日に刊行されアルテラ社に譲渡された、共に”コンフィギュレーションメモリー集積化回路（Configuration memory integrated circuit）”の名称の米国特許第６、０３１、３９１号及び第６、０９７、２１１号 {クーツ - マルチン；クリス他（Couts-Martin; Chris et al.）}はシステム内でプログラム可能な情報を記憶するコンフィギュレーションメモリーを開示している。該コンフィギュレーションメモリーのプログラム作業はジェイターエイジャー（JTAG） {アイイーイー標準１１４９．１（IEEE Standard 1149.1）}命令を使用して行われてもよい。更に、該コンフィギュレーションメモリー内の該コンフィギュレーションデータを使用するプログラム可能なロジックデバイスのコンフィギュレーションはジェイターエイジャー（JTAG）命令を用いて始動される。該コンフィギュレーションメモリーパッケージ内にプルアップ抵抗器が組み入れられる。

20

【００４７】

１９９９年４月１３日に刊行されアルテラ社に譲渡された”プログラム可能なロジックデバイス用のトライステート構造体（Tristate structures for programmable logic devices）” {レディ；スリニバス他（Reddy; Srinivas et al.）}の名称の米国特許第５、８９４、２２８号はトライステート構造体を有するプログラム可能なロジックデバイスアーキテクチャーを開示している。該プログラム可能なロジックデバイスアーキテクチャーはロジックに依るか又はプログラム可能式にか、又は両方で制御されてもよいトライステート構造体を提供する。これらのトライステート構造体を通して、該ロジック要素は該プログラム可能な相互接続に結合され、そこではそれらは該プログラム可能なロジックデバイスの他のロジック要素と結合されてもよい。これらのトライステート構造体を使用して該アーキテクチャーの信号通路は動的にリコンフィギュアされる。

30

【００４８】

２０００年２月１５日に刊行されアキシスシステムズ社（Axis Systems, Inc.）に譲渡された”メモリーシミュレーションシステムと方法（Memory simulation system and method）”の名称の米国特許第６、０２６、２３０号 {リン；シャロンシュー - ピング他（Lin; Sharon Sheau-Pyng et al.）}は４つのモードの動作、すなわち（１）ソフトウエアシミュレーション、（２）ハードウエア加速（Hardware Acceleration）を介したシミュレーション、（３）イン - サーキットエミュレーション {アイシーイー（）}、そして（４）ポストシミュレーション解析（Post-Simulation Analysis）を有するシステムを開示している。ハイレベルでは、該システムは上記４モード又はこれらのモードの種々の組合せの各々で具体化される。これらのモードのコアにはこのシステムの全体の動作を制御するソフトウエアカーネル（software kernel）がある。該カーネルの主制御ループは下記過程、すなわち：システムを初期化し、アクティブなテストベンチの過程 / 部品を評価し、ク

40

50

ロック部品を評価し、クロックエッジを検出し、レジスター及びメモリーを更新し、組合せ部品を広め、シミュレーション時刻を進め、そしてアクチブテストベンチ過程がある限り該ループを続ける、ことを実行する。該発明のメモリー写像(mapping)の側面は、該ユーザーの設計に付随する該多数のメモリーブロックが、該ユーザーの設計をコンフィギュアしモデル化するため使用される、該ロジックデバイスの内部の代わりに、該シミュレーションシステム内のエスラム(SRAM)メモリーデバイス内に写像される構造体及び企画を提供する。該メモリー写像又はメモリーシミュレーションシステムは、(1)該主計算システム及びその付随メモリーシステム、(2)該シミュレーションシステム内の該エフピージーエイバスに結合されたエスラムメモリーデバイス、そして(3)デバッグされつつある該コンフィギュアされ、プログラムされたユーザー設計を含む該エフピージーエイロジックデバイス、を制御しそれらとインターフェースするために、メモリー状態機械(memory state machine)、評価状態機械(evaluation state machine)、そしてそれらの付随ロジックを備える。

10

【0049】

2000年2月1日に刊行されアルテラ社に譲渡された"積項として構成可能なランダムアクセスメモリーを有するプログラム可能なロジック配列デバイス(Programmable logic array device with random access memory configurable as product terms)"の名称の米国特許第6、020、759号{ハイレ; フランシス・ビー. (Heile; Francis B.)}はランダムアクセスメモリー("ラム")としてか又は積項(product term){ "ピー-項(p-term)" }を行うためか何れかで動作させられ得るメモリー回路を備えた、ルックアップテーブルベースのプログラム可能なロジックデバイスを開示している。該メモリーからデータを読み出すために、該メモリーの各個別の行は、該メモリーヘデータを書き込めるようすなわちラムモードで個別にアドレス可能である。代わりに、該メモリーの多数行は該メモリーからピー-項を読み出すために並列にアドレス可能である。該発明のメモリー回路は、ルックアップテーブル型のプログラム可能なロジックデバイスへの追加として特に有用であり、何故なら、該メモリー回路の該ピー-項能力は、さもなければ多数のルックアップテーブルのツリーを要する広いファンイン(fan-in)ロジック機能を行う効率的な方法を提供するからである。

20

【0050】

2000年2月22日に刊行されアルテラ社に譲渡された"トライステート化可能なロジック配列ブロックを組み入れたプログラム可能なロジックデバイス(Programmable logic device incorporating a tri-stateable logic array block)"の名称の米国特許第6、028、809号{シライヒャー; ジェームス. (Schleicher; James.)}は複数の必須に接続された機能ユニットを有する多数機能ブロックを組み入れたプログラム可能なロジックを開示しており、そこでは該多数機能ブロック内の該機能ユニットの少なくとも1つはトライステートロジックユニットである。又該プログラム可能なロジックデバイスは該トライステートロジックユニットに動作的に接続されたトライステートバスを有しており、該トライステートロジックユニットは該トライステートバスにトライステートロジック信号を供給出来るのみならず該トライステートバスからトライステートロジック信号を受けることが出来る。該トライステートバスはトライステートデータ信号を担いそして該プログラム可能なロジックデバイス内の該トライステートロジックユニットの望まれる1つを選択するために動作する選択信号をアドレス指定する。

30

40

【0051】

2000年7月4日に刊行されアルテラ社に譲渡された"プログラム可能なロジックデバイスを使用するリコンフィギュア可能なコンピュータアーキテクチャー(Reconfigurable computer architecture using programmable logic devices)"の名称の米国特許第6、085、317号{スミス; ステファン・ジェイ. (Smith; Stephen J.)}はロジックデバイスを利用するリコンフィギュア可能なコンピュータアーキテクチャーを使用して計算するための方法とシステムを開示している。該計算は第1のプログラム可能なロジックユニットをシステム制御器としてコンフィギュアすることにより達成される。該システム

50

制御器は、該プログラム可能なロジックユニットの第3の1つをリコンフィギュアすると同時に該プログラム可能なロジックユニットの第2の1つ内にアルゴリズムの実施を導く。もう1つの側面では、該計算システムは、その各々が該システム制御器と該複数のプログラム可能なロジックデバイスを電氣的に相互接続するよう配置された、1対の独立した、双方向性バスを有する。この配置を用いると、第1のバスは該プログラム可能なロジックデバイスの選択された1つを該システム制御器により導かれるようリコンフィギュアするため使用されてもよい、一方該第2バスは該プログラム可能なロジックデバイスの動作する1つにより使用される。

【0052】

それぞれ2000年3月7日及び2000年7月18日に刊行されアルテラ社に譲渡された共に「ロジック回路用冗長回路 (Redundancy circuitry for logic circuits)」の名称の米国特許第6、034、536号及び第6、091、258号{マックリントック；キャメロン他 (McClintock; Cameron et al.)} はプログラム可能なロジックデバイスの様なロジック回路用の冗長回路を開示している。該冗長回路は該ロジック回路が該回路上の欠陥ロジック範囲を冗長ロジック回路で置き換えることにより修理されることを可能にしている。ロジック範囲の行及び列は行及び列の交換 (swapping) によりロジック的に再写像 (remapped) されてもよい。該ロジック回路は、冗長度コンフィギュレーションデータにより規定された順序でプログラミングデータを該回路上の種々のロジック範囲へ導くための動的制御回路を含む。冗長性は冗長ロジック範囲を全部か又は部分的にか何れかで使用して実現 (implemented) される。ロジック範囲は、欠陥を有するロジック範囲へ部分的に冗長のロジック範囲を再写像するよう交換されてもよい。該欠陥は次いで行又は列の交換又はシフトを使用して修理されてもよい。ロジック範囲のフォールドされた行 (folded rows) を有するロジック回路は欠陥のある半分の行 (half-row) を冗長な半分の行で置き換えることにより修理されてもよい。

【0053】

2000年3月30日に刊行されザイリンクス社に譲渡された「促成コンフィギュレーションメモリーデータリードバックを有するエフピージーエイ (FPGA having fast configuration memory data readback)」の名称の米国特許第6、069、489号{イワンツーク；ロマン他 (Iwanczuk; Roman et al.)} は各々がユニークなアドレスを有する縦のフレームに分けられたエフピージーエイコンフィギュレーションメモリーを開示している。コンフィギュレーションデータはコンフィギュレーションレジスター内にロードされ、該レジスターはコンフィギュレーションデータを、フレームづつ並列に、転送する。好ましい実施例では、入力レジスター、シャドー入力レジスター (shadow input register) そしてマルチプレキサー配列が、従来のエフピージーエイエスより多い数の入力ビットを使用して効率的なコンフィギュレーションデータ転送を可能にする。柔軟な外部インターフェースは予め決められた最大幅からその選択された分数まで下方へ変化するバスサイズとの接続を可能にする。コンフィギュレーションデータ転送は、この様なデータを、最小の遅延を有するフレームづつベースでメモリーセル内にドライブするためにシャドーレジスターを使用することによりそしてより広いコンフィギュレーションデータ転送バスを開発するためにマルチプレキサーを使うことによりより効率的になる。コンフィギュレーションリードバック (configuration read-back) の速度は、双方向データ転送をサポートするコンフィギュレーションレジスターロジックを使うことによりコンフィギュレーションデータ入力の速度と実質的に等しくなる。提案されたエフピージーエイコンフィギュレーションメモリーを使用すると、古いデバイス用に設計されたビットストリーム (bit stream) は追加的コンフィギュレーションメモリーセルを有する新デバイス用に使用出来る。

【0054】

1995年12月19日に刊行されクイックターンデザインシステム社に譲渡された「電氣的にリコンフィギュア可能なハードウェアエミュレーション装置を使用して回路設計をエミュレートするための方法 (Method for emulating a circuit design using an elect

10

20

30

40

50

rically reconfigurable hardware emulation apparatus) ” の名称の米国特許第 5、477、475 号 { サンプル ; ステファン ピー . (Sample; Stephen P.) } はユーザーが回路又はシステムコンフィギュレーションを表すデータを入力するデータエントリーワークステーションを有する電子回路又はシステムの物理的エミュレーション用システムを開示している。このデータは豊富に相互接続されたアーキテクチャーを備えたプログラム可能なゲート要素の配列をプログラムするに好適な形式に変換される。ユーザーの回路又はシステムのブイエルエスアイ (VLSI) デバイス又は他の部分を外部的に接続するよう用意されている。該プログラム可能なゲート配列内の未使用回路通路の利用により内部のブロープ作業用の相互接続のネットワークが利用可能である。

1998 年 10 月 13 日に刊行されアルテラ社 (Altera Corporation) 及びクイックターndeザインシステム社 (Quickturn Design Systems, Inc.) に譲渡された ” 第 2 信号への入力の完全な並べ換え能力を有するルックアップテーブルベースのロジック要素 (Look-up table based logic element with complete permutability of the inputs to the secondary signals) ” の名称の米国特許第 5、821、773 号 { ノーマン他 (Norma et al.) } はプログラム可能なロジックデバイス用ロジック要素を開示している。該ロジック要素はロジック機能実施用ルックアップテーブル、プログラム可能な遅延ブロック、ラッチ又はフリップフロップとしてコンフィギュア可能な記憶ブロックそして診断用シャドーラッチを含む。該ロジック要素への複数入力及びこれらの入力の補数が該記憶ブロックの第 2 機能を制御するために利用可能である。

1991 年 1 月 8 日に刊行されウルトラシステムズデフエンス社 (Ultrasystems Defense Inc.) に譲渡された ” 実時間データ処理を行うためにリコンフィギュレーションスイッチングネットワーク内に接続可能なプログラム可能な状態機械 (Programmable state machines connectable in a reconfiguration switching network for performing real-time data processing) ” の名称の米国特許第 4、984、192 号 { フリン (Flynn) } はプログラム可能なロジック回路を実施するためのプログラム可能な要素 { ピーイー (PE) } を開示している。該プログラム可能な要素はラン (run) アドレスレジスター、ロード (load) アドレスレジスターそしてランダムアクセスメモリー { ラム (RAM) } を含む。該ラムは該ロードアドレスレジスターを使用して状態遷移テーブルで初期化される。ラム初期化後そして各クロック遷移に際して、該ランアドレスレジスターは外部入力及び現在の状態をロードされる。現在の状態は該ラム出力から該ランアドレスレジスター入力への内部フィードバックラインからの該ラムの次の状態出力として受信される。該ラム出力は、該内部フィードバックライン内に記憶される次の状態と、該プログラム可能な要素の外の宛先 (destination) への要素データ出力とに分けられる。複雑なプログラム可能なロジック回路を実施するためのステートプロセサー { エスピー (SP) } は複数のプログラム可能な要素及びマトリックススイッチを有する。該マトリックススイッチはプログラム可能な要素用の外部フィードバックを提供する。加えて、何れかのプログラム可能な出力が何れかのプログラム可能な要素入力ヘルートを決められてもよい。プログラム可能なロジックシステムを実施するためのステートプロセサーの階層的ネットワーク (hierarchical network) は複数のステートプロセサーと複数のマトリックススイッチを含む。該階層的ネットワーク内では何れのステートプロセサー入力も何れかのステートプロセサーの出力を受けることが出来る。

1 つ以上のマトリックスがそれぞれのステートプロセサーに接続され、自身も中央マトリックスにより相互接続される図 13 を特に参照した。

主たる請求項の前文に適合するデバイスを示す、1980 年シュプリングerverlag (Springer-Verlag) 刊行で、ウー . ティーツエ、ツエーハー . シェンク (U. Tietze, CH. Schenk) 著 ” 半導体回路技術 (Halbleiter-Schaltungstechnik) ” の 491 及び 492 頁の図 20 . 1 も又参照した。

これらの参考文献のどれにも、セルが同様なセル内にコンフィギュアされ得て、それにより帰納的な仕方でプログラム可能なロジックデバイスとそのロジック要素との間に自己相似性 (self-similarity) を導入する様な仕方で、セルがプログラム可能なマトリックス

10

20

30

40

50

により相互接続されることを可能にするデバイスアーキテクチャーは開示されていない。

【 0 0 5 5 】

【本発明の概要】

本発明の目的は高い柔軟性を可能にし、設計から完成製品までの時間を減じる、デジタル回路の設計に特に適合した、改良されたデバイスアーキテクチャーを提供することである。

【 0 0 5 6 】

この目標に対して、本発明の広い側面に依れば、汎用ハードウェアデバイスが提供されるが、該汎用ハードウェアデバイスは、

データを記憶するための第 1 の複数のセルと、そして

前記複数のセルの少なくとも 1 組の入力及び出力と接続された第 1 のプログラム可能なマトリックスとを具備しており、前記セル内に選択的にデータを記憶しそして前記セルの少なくとも 1 つを前記セルの少なくとも 1 つと接続するために前記マトリックスを選択的にプログラムすることにより複数のハードウェア応用品が実施され、

少なくとも 1 つの前記セルはデータ記憶用の第 2 の複数のセルと、そして、

前記第 2 の複数のセルの入力及び出力に接続された第 2 のプログラム可能なマトリックスとを備えており、前記第 1 の複数のセル及び前記第 2 の複数のセルの各セルは該汎用ハードウェアデバイス全体と同様なアーキテクチャーを有してもよく、そして

前記第 1 の複数のセルの部分でない少なくとも 1 つのセルは該少なくとも 1 つのセルのポートを経由して直接アクセスされ得ることを特徴とする。

【 0 0 5 7 】

このようなデバイスアーキテクチャーは、より大きなセルを形成するよう自身が組み合わされ得るより大きなセルを形成するように、なおこの過程は必要なだけ繰り返されるが、そしてデータを該構成セルにダウンロードすることにより該組合わされたセルをハードウェア応用品としてコンフィギュアするように、セルが組み合わされることを可能にする。好ましくは、該セルはアドレス指定可能なメモリー位置を有するルックアップテーブルとしてコンフィギュア可能であるのがよく、そこでは記憶されたデータは該ルックアップテーブルにより実施される機能を規定している。該機能自身は、高級プログラム言語を使用してプログラムされ得てそして該セルの望ましい接続を実施するためのコードと一緒にフォーマットされてもよい。該フォーマットされたデータは次いで該デバイス内の該セルへダウンロードされる。一旦ダウンロードされると、該デバイスは、該望まれた機能を実施するために使用された高級プログラムコードに最早依らない仕方で予めプログラムされた機能を実行する。結果として該デバイスの動作は該高級プログラムコードの効率から独立している。同一コードが該デバイスをシミュレートするため使用されてもよく、かくして該デバイスの設計とシミュレーションを大幅に容易化しそして設計から市販までの時間を大幅に減じている。

【 0 0 5 8 】

又本発明は該ハードウェアデバイスの設計、シミュレーションそしてデバッグ用のツールを提供する。これらのツールは又、完成デバイスが所望のように動作することを確立した後該デバイスの全て又は部分をエイシック (ASIC) に変換するのに役立つ、尤も該製品の期待製品寿命が低落するとこの様な変換の価値は減少するが。

【 0 0 5 9 】

【本発明の詳細な記述】

図 2 は本発明のデバイス 100 の基本的アーキテクチャーを略図的に示す。該デバイスアーキテクチャーは少なくとも 1 つのプログラム可能なマトリックス 102 を介して相互接続されるセル 101 の集まりである。セル 103 はより小さいセル 101 から作られてもよい。同様に、該付随マトリックス 102 と一緒に該セル 101 又は 103 の各々は 104 及び 105 の様なブロックの部分形成してもよい。どんなブロックも全体のデバイス 100 と同じアーキテクチャーを有する。どんなブロックも 1 つのセルとしてコンフィギュアされ得る。1 つのブロックの出力と同じ又はもう 1 つのブロックの入力との間にどん

な接続も行い得るが、2つのブロックの内部セル間の特定の相互接続は必ずしも可能でなくてもよい。これはブロックが、該ブロックを含むブロックの数である " レベル (level) " に付随することを可能にする。かくして、例えば、レベル0のブロックは該デバイスそれ自身であり；レベル1のブロックは該デバイスを形成し；そしてレベル2のブロックはレベル1のブロックを形成する。これを云うことで、図2は略図的でありそして該プログラム可能なマトリックス102が該ブロックの境界の内に示されるか又は外にかは重要でなく、それは何れの場合もブロック内の全てのセルは少なくとも1つのプログラム可能なマトリックスに接続されねばならないからであることが気付かれるべきである。ブロックに対し中にあり該ブロック外への接続を可能にするセルの該接続が該ブロックのポートとして形成される。

10

【0060】

ブロックがセル又は2, 3のセルを形成することが更に気付かれるべきである。同様に、該デバイス100はそれ自身、プログラム可能なマトリックスにより相互接続された多数のセルを含むブロックでありかくしてどんなブロックもデバイス100の同様なアーキテクチャーを有しそして実際デバイスとして見なされてもよい。かくして該デバイス100は多数の同様なデバイスを含みそして多数の同様なセルから形成されたセルと見なされてもよい。

【0061】

該名前 " ブロック " は、例えこの区別が容易な明確化用の説明にだけ関係していても、完全なデバイス100とそれの、同様なアーキテクチャーを有する部品との間で区別が行われることを可能にする。請求項が関係する限りでは、該完全なデバイスと、同様なアーキテクチャーを有するその何等かの部品との間に区別はない。実際、本発明の本質的特徴は、該デバイスの部品のアーキテクチャーは全体としての該デバイスのアーキテクチャーと同様である事実に存在する。同じ特徴 (token) により、ブロックはそれ自身デバイスなのでそれは異なる方法で実現され得てかくしてデバイスは異なる構造を有する2つ以上のブロックを備えることが出来る。

20

【0062】

該マトリックス102は単一の実体 (single entity) である必要はなく部分に分割され得ることは気付かれるべきである。同様に、ブロック105はセルの多数グループを含み、該グループの2つは106及び107で識別され、各々はことによると種々の数のセル101を含みそして両者は単一のマトリックス108によりサービスされることが分かる。その構成セル、そして何等かの他の構成要素と一緒に該ブロック105は該ブロック105の外に示す第2マトリックス109に依っても又サービスされる。該マトリックス108及び109の各々は典型的に該マトリックス102と同一構造でありそしてそれが該ブロックの中に示されるか又はその外かは単に利便性の問題である。かくして、該マトリックスが該図内に描かれる仕方は図解のためだけの略図用である。又1つのマトリックス内で利用出来る接続の部分はもう1つのマトリックス内で複製されてもよいことも気付かれるべきである。何等かのこの様な複製は実施品内では除去されるであろう。実際には、図8を参照して下記で説明されるように、該マトリックスは単に、対応するフリップフロップ (Flip-Flop) ヘロジック " 1 " 又は " 0 " を書くことにより、該スイッチは閉じられ又は開かれ、それにより該セルが何等かの必要なトポロジーに依り接続されることを可能にするように、各々がフリップフロップにより制御されるスイッチ { シーモス (CMOS) スイッチの様な } の集まりである。該マトリックスの全ての該スイッチに関する該フリップフロップはグループで配置され、該フリップフロップの何れか1つがそれへデータを書き込む目的で選択されさせる補助回路に付随されている。かくして、該フリップフロップと付随補助回路はラムにより実施され、 " マトリックス制御メモリー (Matrix Control Memory) " として参照される。オプションでは、該マトリックス制御メモリー内のデータも読まれ得る。

30

40

【0063】

示された該デバイスアーキテクチャーは、自己類似度 (self-similarity) の特性を示す

50

複雑な幾何学的形状のクラスの何れかを説明するため数学で使用する " フラクタル (fractal) " 構造を思い出させる。

【 0 0 6 4 】

該デバイスの入力ピン及び出力ピンは該プログラム可能なマトリックスに；該入力ピンは該マトリックス入力へ；該出力ピンは該マトリックス出力から、の様に接続される。該入力から又は該出力から直接より低いレベルにアクセスするために、該ブロックのポートが使用されるべきである。

【 0 0 6 5 】

図 3 は第 1 実施例のセル 1 1 0 を略図的に示す。該セル 1 1 0 は (n + m) アドレスライン 1 1 2 を有するランダムアクセスメモリー { ラム (RAM) } 1 1 1 を備え、該ラインは、それらがそのアドレスビット数 (m + n) の最小が 1 であり得る 1 つのアドレスバスを機能させるが、2 つの別々のバスとして示されている。データバス 1 1 3 は該ラム 1 1 1 のアドレス可能なメモリー位置内に記憶されたデータが読み出されることを可能にし、その最小数も又 1 であるデータビット数 d を収容する。該データバス上に現れるデータはその出力 1 1 5 が該セル 1 1 0 の出力を構成するラッチ 1 1 4 によりラッチされる。該ラム 1 1 1 は該望まれるデータをロードされ得る。

10

【 0 0 6 6 】

図 4 は第 2 の実施例のセル 1 2 0 を略図的に示す。該セル 1 2 0 は (n + m) アドレスライン 1 2 2 を有するラム 1 2 1 を備え、該ラインも再び、それらはそのアドレスビット数 (m + n) の最小が 1 であり得る単一アドレスバスを機能させるが、2 つの別々のバスとして示されている。データバス 1 2 3 は該ラム 1 2 1 のアドレス可能なメモリー位置内に記憶されたデータが読み出されることを可能にし、そしてその最小数も又 1 であるデータビット数 d を収容する。この場合、該データバス 1 2 3 上に現れるデータは該セル 1 2 0 の出力を構成する。該アドレスバス 1 2 2 上に現れる該アドレスはそれぞれのラッチ 1 2 4 によりラッチされる。該ラム 1 2 1 は下記説明の仕方では望まれるデータをロードされ得る。該ラム 1 2 1 の入力に 2 つのラッチ 1 2 4 が示されるが、それらはセル 1 2 0 の " そのラッチ (the latch) " として参照され、該アドレスをラッチするため使用されるラッチの実際の数に対し区別はされない。

20

【 0 0 6 7 】

図 3 及び図 4 に示された該ラム 1 1 1 及び 1 2 1 のみならずそのラッチ 1 1 4 及び 1 2 4 も又図 2 に示す該デバイス 1 0 0 の部分である。実用化された特定の実施例では、該ラムは米国 9 5 0 5 4、カリフォルニア州、サンタクララ、インテグレートッドデバイステクノロジー (Integrated Device Technology, Santa Clara, California, USA 95054) のアイデーテー 6 1 1 6 (IDT6116) 上でモデル化され、該ラッチは米国 7 5 3 8 0 - 9 0 6 6、テキサス州、ダラス、テキサスインスツルメント社 (Texas Instruments Incorporated, Dallas, Texas USA, 75380-9066) のレジスターエスエヌ 7 4 エイチシー 3 7 4 (SN74HC374) 上でモデル化される。該セルの両コンフィギュレーションでは、n、m 及び d の値は該デバイスを使用して実施されるべき応用品により要求されるように割り当てられてもよい。

30

【 0 0 6 8 】

図 5 は図 4 のロジック的なセルをより詳細に略図で示す。該図は (m + n) ビットのアドレスバス 1 3 2 と d ビットのデータバス 1 3 3 を有するラム 1 3 1 を備えるセル 1 3 0 を示す。ラッチ 1 3 4 a 及び 1 3 4 b は該アドレスバス 1 3 2 上に該アドレスをラッチするために使用される。再び、該アドレスバスとそのラッチ 1 3 4 は唯図解により分割されて示されることは注意されるべきである。機能的には、単一バスがあるに過ぎず、そのラッチは単一ラッチとして考えられる

40

【 0 0 6 9 】

【 外 1 】

。ロジック信号はOE及び \overline{OE} はそのラッチ134の部分、又は延長部、として

【0070】

も考えられ得るラッチ136によりラッチされ、補助回路137を経由して該ラム131の出力イネーブル(OE)へ供給されそして該ラムをトライステート条

【0071】

【外2】

件にならしめる。ロジック信号CS及び \overline{CS} も又そのラッチ136によりラッチ

10

【0072】

され、該ラム131が選択されるか又は切断されることを可能にするよう該補助

【0073】

【外3】

回路137を経由して供給される。該CS及び \overline{CS} 信号の対の数は該全ブロック

【0074】

(又はデバイス)を単一セル内に写像することを可能にするような数である。クロックがそのラッチ134及び136に回され、又マトリックスを経由して回されるクロックイネーブル信号(CE)によりイネーブル又はデスイネーブルに

20

【0075】

【外4】

され得る。該信号OE、 \overline{OE} 、CS1、 $\overline{CS1}$ 、CS2、 $\overline{CS2}$ 、CS3、 $\overline{CS3}$ 、CS4、 $\overline{CS4}$ 等、そしてCEは、接続されない時は、アクチブロー信号が

【0076】

ロー(LOW)にセットされそしてアクチブハイ信号がハイ(HIGH)にセットされる、すなわちそれらのイネーブルとされる状態に、セットされるような、デフォルト値にセットされる信号となる。これに対する様に、接続されない時のWEのデフォルト値はそのデスイネーブル状態にセットされる。該ラムが選択されない(チップセレクトがアクチブでない)時は、それは両者共トライステート条件にあり、書き込み抑制される(write disabled)。

30

【0077】

図5のセルに基づいたデバイス用の起こり得るタイミング実施をここで説明する。

【0078】

(a)全てのセル用に単一"マスタークロック(master clock)"が使用される。

【0079】

(b)該マスタークロックが"書き込み"信号である時"書き込み"信号は無い。該書き込みは該"書き込みイネーブル(Write Enable)"信号に依りアクチブかそうでないかになる。

40

【0080】

(c)該"出力イネーブル"信号及び"書き込みイネーブル"信号も又該マスタークロックによってもラッチされるので、従って該"書き込み"信号と何等かの他の信号との間にタイミングレース又は衝突はない。

【0081】

(d)該ラッチ動作の開始は該書き込み動作の終わりであるので該ラッチ動作と該書き込み動作の間に衝突はない。これにも拘わらず、該ラッチへの2つの連続するクロックパル

50

スの間の時間に等価である、安全マージンを同じサイクルを保持しながら増加するために該書き込み信号のパルス幅を僅かに減じることが望ましい。

【 0 0 8 2 】

(e) 書き込みが起こると、データは該デバイスの入力から又は他のセルから、該マトリックスを経由して要求されたセルへ回らされる。後者の場合、該 2 つのセルのそれぞれのデータバスは相互接続される。データが書き込まれる該ラムの該出力イネーブルは該出力デイスエーブル状態 (トライステート) にあり、そこからデータが書かれる該ラムの該出力イネーブルはアクチブになっている。

【 0 0 8 3 】

(f) O E 及びチップセレクト信号が図 5 を参照した上記説明の様に提供される。該 O E 信号は書き込み動作、該セル間のマルチプレクス動作等に使用されてもよく、該チップセレクト信号は図 6 を参照して下記で説明される様により大きいセルの創生用に使用されてもよい。

【 0 0 8 4 】

書き込み動作の全く要しない応用に好適なデバイス用に、該クロックサイクルは、より速い (実時間の) 応用品を創るためにより短く出来てそして該セル出力が用意出来た時、該マスタークロックのローツーハイ遷移 (low-to-high transition) が起こり得る。これは図 1 5 及び 1 7 に示すラムサーバー組合せ (RAM-Server combination) の説明と図 1 6 及び 1 8 に示すタイミング線図とからより明らかになる。

【 0 0 8 5 】

図 6 は 2 倍のサイズを有する、すなわち (n + 1) ビットのアドレスバスによりアドレス指定されるアドレス可能位置の数が 2 倍の複合セル 1 4 0 を形成するよう各々が n ビットのアドレスを有する 2 つのセルを接続するための例を示す。各構成セルの部品が図 5 を参照して上記で説明したそれらと同一である程度に、図 6 では同様な参照数字が使用される。かくして、該複合セル 1 4 0 は両者がそれぞれ n ビットのアドレスバス 1 3 2 a 及び 1 3 2 b を有する 1 3 1 a 及び 1 3 1 b と識別される 2 つのラムを含む。かくして、組合せアドレスの該 n の最下位ビット (the n least significant bits) がそれぞれのラッチ 1 3 4 a 及び 1 3 4 b を経由して該対応するラム 1 3 1 a 及び 1 3 1 b に供給される。図解により、該組合せセルに供給された (n + 1) ビットアドレスは m ビットアドレスバス及び n + 1 ビットデータバスを有するラム 1 4 2 から引き出されており、 m ビットアドレスは m ビットラッチ 1 4 3 を経由してそれに供給されている。2 つのラム 1 3 1 a 及び 1 3 1 b のデータバス 1 3 3 は該マトリックスを介して接続され、各データ出力は、該ラムの選択された 1 つ上のデータのみが出力となるようにトライステートである。該 (n + 1) ビットアドレスバスの最上位ビット (MSB) は、該 2 つのラム 1 3 1 a 及び 1 3 1 b のどちらがデータを該データバス 1 3 3 に供給するかを制御するために使用される。この目的で、それは該ラム 1 3

【 0 0 8 6 】

【 外 5 】

1 a を制御するそのラッチ 1 3 6 a の該 $\overline{CS1}$ 入力と該ラム 1 3 1 b を制御する

【 0 0 8 7 】

そのラッチ 1 3 6 b の該 C S 1 入力とに接続されている。

【 0 0 8 8 】

該回路の動作は下記の様である。もし該組み合わせされた (n + 1) ビットアドレスの最上位ビットに回されるラム 1 4 2 のデータの最上位ビットが 0 ならば、該ラム 1 3 1 a はイネーブルとなり、該ラム 1 3 1 b はデイスエーブルとなる。逆に、もし該組み合わせされた (n + 1) ビットアドレスの該最上位ビットに回されるラム 1 4 2 のデータの最上位ビットが 1 ならば、該ラム 1 3 1 a はデイスエーブルとなり、該ラム 1 3 1 b はイネーブルとなる。図 5 に示す補助回路 1 3 7 を戻って参照すると、該 C S 1 入力は、その出力が全て

のその入力がいネーブルの場合のみアクティブな第 1 のロジックのアンドゲート 1 4 5 へ供給される。上記注意の様に、該マトリックスにより接続されないどんな入力も自動的にいネーブルになるので、ロジックのアンドゲート 1 4 5 の該出力は、もし $\overline{CS1}$ がいネーブルならアクティブにそしてもし $\overline{CS1}$ がデイスエーブルならインアクティブになる

【 0 0 8 9 】

【 外 6 】

。同様に、該 $\overline{CS1}$ 入力は、その出力が全てのその入力がいネーブルの場合（ア

【 0 0 9 0 】

10

クティブロー）のみアクティブな第 2 のロジックのアクティブローのアンドゲート 1 4 6 へ供給される。再び、該マトリックスにより接続されないどんな入力も自動的

【 0 0 9 1 】

【 外 7 】

にいネーブルになるので、ロジックのアンドゲート 1 4 6 の該出力は、もし $\overline{CS1}$ がいネーブルならアクティブ (ACTIVE) にそしてもし $\overline{CS1}$ がデイスエーブルな

【 0 0 9 2 】

20

らインアクティブ (INACTIVE) になる。かくして、もし該最上位ビットがローなら

【 0 0 9 3 】

【 外 8 】

、該ラム 1 3 1 a の $\overline{CS1}$ はいネーブルとなり、該ラム 1 3 1 a はオペラチブ（

【 0 0 9 4 】

operative) であり、もし該最上位ビットがハイなら、該ラム 1 3 1 b の $\overline{CS1}$ はいネーブルとなり、該ラム 1 3 1 b はオペラチブである。そこで該ラム 1 3 1 a がアクティブの時、該ラム 1 3 1 b はインアクティブであり、逆に該ラム 1 3 1 a がインアクティブの時、該ラム 1 3 1 b はアクティブである。

30

【 0 0 9 5 】

精確に同じ方法で、2 つのラム 1 4 0 が組み合わせられ得るが、その場合該 \overline{CS}

【 0 0 9 6 】

【 外 9 】

2 及び $\overline{CS2}$ 信号は該アドレスの 2 つの最上位ビットを収容するためにも使用さ

【 0 0 9 7 】

れる。この様な拡張は、特定の応用品により要求される多さのアドレス可能なメモリー位置を有するラムを作るために任意に繰り返すことが出来る。該 2 つのラム 1 3 1 a 及び 1 3 1 b は等しいサイズのアドレスバスを有するよう図 6 で示されることは注意されるべきである。しかしながら、これは必要ことでなく、応用品は異なるサイズのアドレスバスを有するトポロジが組み合わせられることを指示してもよく、それは普通に起こるのである。

40

【 0 0 9 8 】

それはデバッグの目的で設計中に主として使用されるが、該 " クロックいネーブル " 信号が該セルへの入力として考えられてもよい。

【 0 0 9 9 】

図 7 は D セル出力 1 5 1 を A セル入力 1 5 2 へ接続するマトリックス 1 5 0 を略図的に示

50

す。

【0100】

図8は図7の回路に使用されてもよい、4つの入力ラインと3つの出力ラインとを有する飽和型マトリックス(saturated matrix)155を略図的に示す。該マトリックス155は該ブロック内で該セル出力ライン151の各々を該セル入力ライン152の各々に接続出来ねばならない。各セル出力151はアルファベット文字a, b, c, dにより呼称される該マトリックス入力155のそれぞれの入力に接続される。該マトリックス155は各セル出力151の数字1, 2, 3により呼称される1つ以上のセル入力152への接続を可能にするよう働く。実際には各セル出力151を全ての起こり得るセル入力152に接続する能力を持つ理由はなく、かくして図9に示す様に飽和されないマトリックスの使用が可能である。しかしながら、図8に於ける様に該飽和型マトリックス155を使用するのがオートメーションをより簡単化し、動作は実時間に於いて、より速い。該セルはより小さいセルから作られ得て、幾つかの応用品ではより大きいセルは作られないが、より大きいセルの出力及び入力は予め決められる。

【0101】

該飽和型マトリックスの動作は下記の様である。入力a, b, c, dの各々は対応するスイッチを介して該出力1, 2, 3の各々に接続される。かくして、該入力a, b, c, dはスイッチa1, b1, c1, d1を介して出力1に接続される。同様に、該入力a, b, c, dはスイッチa2, b2, c2, d2を介して出力2に接続されそしてそれらはスイッチa3, b3, c3, d3を介して出力3に接続される。入力aを出力1に接続するために、該スイッチa1は閉じられる。cを3に接続するために、該スイッチc3は閉じられる。bを1及び3の両者に接続するためにスイッチb1及びb3の両者が閉じられる。bとdの両者を2と3に接続するためにスイッチb2, b3, d2, そしてd3が閉じられ、以下同様である。

【0102】

各スイッチは該スイッチを"閉じて"又は"開いて"セットする制御ライン(図示せず)を有し、該スイッチの状態を記憶する1ビットメモリーに接続されている。実際には非常に多くのスイッチがあるので、各スイッチ状態を記憶する全てのビットはメモリー構造体(memory structure)内に配置される。換言すれば、該スイッチの状態を記憶するメモリーユニットがある。該メモリー内の各ビットは1つの制御ラインに接続され、該マトリックス内のスイッチ数と同じ数のビットが該メモリー内にある。この様な手段により、該メモリーは各スイッチの状態を閉じか又は開きかと制御するためのマトリックス制御メモリー(Matrix Control Memory)として機能する。上記例では、該マトリックス155は4ビットデータバスを3ビットアドレスバスに接続する。しかしながら、該マトリックス155は、3ビットデータバスを4ビットアドレスバスへ接続するように出力を形成するラインa, b, c, dと該入力を形成するライン1, 2, 3とに等しく良好に接続され得ることは評価されるであろう。

【0103】

図9は、各々がそれらの各スイッチを制御するそれ自身のメモリーを有する、図8に示す複数の相互接続された飽和型マトリックス155を含む非飽和型マトリックス156の例を略図的に示す。全ての該メモリーは該マトリックス制御メモリーとして機能する1つの大きなメモリーとして組織される。該マトリックスをプログラムすることは、下記で説明される様に該マトリックス制御メモリーに適当なデータをロードすることにより達成され、該プログラム動作は該デバイスの望まれるトポロジーをセットする。

【0104】

限定されるが充分な数の接続を有するよう作られたこの様なマトリックス156は、リンク{ルーティング(routing)}を選ぶコードが僅かにより複雑であるが、それがダイ空間を節約するので、同じ数のスイッチ用接続を有する等価な飽和型マトリックスより好ましい。かくして、各マトリックス155は飽和型でありそして、

D = 該マトリックスへの入力ライン数、

10

20

30

40

50

A = 該マトリックスの出力ライン数、
 X = 入力マトリックス 1 5 5 の数、
 Y = 出力マトリックス 1 5 5 の数、
 Z = 中間列マトリックス (middle column matrixes) 1 5 5 の数、
 を表していると仮定して、
 X、YそしてZは次の様に計算され、
 【 0 1 0 5 】
 【 数 1 】

$$X = \text{ceiling} \left(\sqrt{\frac{D^2 + D \cdot A}{A}} \right)$$

$$Y = \text{ceiling} \left(\frac{A \cdot X}{D} \right)$$

$$Z \geq \frac{D}{X}$$

10

【 0 1 0 6 】

ここで "シーリング (ceiling)" は非整数 (non-integer) が次の最高整数 (the next highest integer) まで丸められることを表す。

20

【 0 1 0 7 】

該入力マトリックスの各々は該中間列マトリックスの各々に接続される。該出力マトリックスの各々は該中間列マトリックスの各々に接続される。交差接続 (cross connects) の限界を避けるために、Zを増加することは可能である。例えそうであっても、スイッチの数と付随するメモリーは同じ数の入力ピンと出力ピンとを有する飽和型マトリックス内のスイッチ数と付随メモリーよりずっと少ない。これは全てのレベルでの全てのセルを接続するために1つのマトリックスが使用される時特に重要である。より大きいセルを形成するためにセルを組み合わせようと意図する傾向のために、最終ユーザーが同じブロック内の2つのセルを接続する尤も度 (likelihood) は彼が異なるブロック内の2つのセルを接続する尤も度より大きいことは気付かれるだろう。従って、この様な1つのマトリックスが使用される時、該デバイスの設計中に、どの入力及び出力マトリックスに該セルのピンが接続されるかを斟酌することが勧められるが、何故ならば該最終ユーザーはこれらのセルからより大きいセルを形成することを選ぶからである。又、もし代わりに、各ブロック内で別々のマトリックスが提供されるならば、幾つかの接続の累積遅延はもし1つのマトリックスが使用されたとするより大きくなりそうであることも注意されるべきである。又該セルの出力と該セルの入力の間の相互接続に加えて該デバイスの入力及び出力への該マトリックス内の接続を提供する必要性も考慮されねばならない。

30

【 0 1 0 8 】

単に、必要なトポロジーを選択し、該セルの各の記憶要素内へデータをダウンロードすることにより種々のハードウェア応用品を実施するために如何に該デバイスが使用されるかを理解するために、ここで種々の例を説明する。該デバイスは図4のセル1 2 0を使用すると同じ仕方で作動するが、説明の容易化のために、幾つかの例は図3に示すセル1 1 0に基づいている。下記例では、図3に示すセルと図8に示すマトリックスとに共通な部品は同一参照数字により参照される。

40

【 0 1 0 9 】

【例 1 - カウンター】

図10は、その、該セルの出力を構成する、出力1 1 5が該マトリックス1 5 5の入力に接続されるn入力ラッチ1 1 4に供給されるnビットデータ出力バス1 1 3を有するラム1 1 1を備える1つのセル1 1 0を使用するカウンター1 6 0を略図的に示す。該セルの

50

n 出力ラインの各々は該マトリックス 1 5 5 を経由して該ラムのアドレスバス 1 1 2 のそれぞれのアドレスラインに接続される。該ラムは次のデータをロードされる。

【 0 1 1 0 】

アドレス	データ
0	1
1	2
2	3
.	.
K - 1	K
K	0

10

定常状態では、該ラム 1 1 1 の該 " アドレス " を規定するそのラッチ 1 1 4 の出力 1 1 5 に " 番号 (number) " がある。従って、該ラムの " データ " - 該ラッチの入力 - は該テーブルでセットされる。後者から離れた全てのアドレスについて、該ラムの何等かのアドレス可能な位置内のデータがそのアドレスより 1 大きいものに等しく、これは次のクロックパルス時新しいアドレスになる。かくして、該ラムがクロックされる度に、そのラッチ 1 1 4 は、そのデータが現在のデータプラス 1 に等しい次のアドレス可能な位置のアドレスをラッチする。遅延時間の後、該ラムは新クロック用に用意が出来て、該サイクルは繰り返され、該出力は逐次インCREMENTされる。

【 0 1 1 1 】

20

【例 2 - アップ - ダウンカウンタ】

(Up-Down Counter)

図 1 1 は該ラム 1 1 1 がそのそれぞれのアドレス可能な位置内に 2 つのテーブルを記憶するように作られていることを除くと、図 1 0 に示すカウンタと実質的に同一なアップ - ダウンカウンタ 1 6 5 を略図的に示す。勿論、これは該ラムが図 1 0 のカウンタで使用されるその 2 倍大きいこと、又は該カウンタの範囲が半分であることを要する。何れの場合も、該アドレスの 1 ビットは、該アドレスバスの残りのビットに供給される時、そのデータの値が該アドレスより 1 小さい新アドレスを指し示すデータを記憶するために該ラム内に新範囲をセットするために使用される。該ラム内の新範囲は次のデータでロードされる。

30

【 0 1 1 2 】

アドレス	データ
0	K
1	0
2	1
3	2
.	.
K - 1	K - 2
K	K - 1

40

該アップ / ダウン信号は又、該セルの全ての入力がそうであるように、該マトリックス 1 5 5 から回される。

【 0 1 1 3 】

【例 3 - 遅延】

遅延を達成するためには、該ラムは冗長である。従って、該ラム 1 1 1 は、例えば、アドレス " 0 " ではデータは " 0 " であり、アドレス " 1 " ではデータは 1 である等々の様に、簡単に該アドレスを該データに転送するようコードされ得る。各クロック信号で、該セルの入力のデータはラッチされ、直接該ラムを経由して外へ回され、1 クロックの遅延が達成される。又この様な遅延の接続は例 4 で下記に説明され、図 1 3 で示されるシフトレジスタでも使用される。

50

【 0 1 1 4 】

【例 4 - シフトレジスタ】

図 1 2 はブロック内の 3 つのセルを使用し、それぞれのラム 1 1 1 a、1 1 1 bそして 1 1 1 c を備えるシフトレジスタ 1 7 0 を略図的に示すが、該ラムの各々は、そのそれぞれの出力 1 1 5 a、1 1 5 bそして 1 1 5 c が該セルの出力を構成する 3 n ビット出力データバス 1 1 5 を形成するため組み合わせられる、それぞれの n 入力ラッチ 1 1 4 a、1 1 4 bそして 1 1 4 c に供給される n ビットデータ出力を有している。該マトリックス 1 5 5 は該出力データバス 1 1 5 a、1 1 5 bそして 1 1 5 c の各々の該 (n - 1) 最下位ビットを該対応するラム 1 1 1 a、1 1 1 bそして 1 1 1 c のそれぞれのアドレスビットへ接続するようプログラムされる。同様に、該出力データバス 1 1 5 a、1 1 5 bそして 1 1 5 c の各々の最上位ビットは該マトリックス 1 5 5 により次のラムの最下位ビットに供給され、例外になるのは、該応用品により使用されてもよい、該シフトレジスタにより簡単に捨てられる該データバス 1 1 5 c の最上位ビットである。各ラム内のデータは、上記例 3 で説明された該遅延で行われた様に、例えば、アドレス " 0 " では該データは " 0 " であり、アドレス " 1 " では該データは " 1 " である等々の様に、該アドレスを該データに簡単に転送するようコード化される。該シフトレジスタのトポロジは必要なシフトを作るよう設計され、該マトリックス 1 5 5 は該必要な接続を達成するようプログラムされる。

10

【 0 1 1 5 】

該マトリックス 1 5 5 の接続は明確なために図 1 2 には示されないことは理解されるであろう。しかしながら、該シフトレジスタ 1 7 0 の動作を明らかにするため、ラム 1 1 1 a のみの接続と動作を今図 1 3 を参照して説明するが、そこでは同一参照数字は図 1 2 のそれらの部品を表すため使用される。

20

【 0 1 1 6 】

図 1 3 は 8 ビットアドレスバス (すなわち $n = 8$) を有するラム 1 1 1 a を備える 1 つのセルのシフトレジスタ 1 7 5 を示し、そのデータビットの各々はラッチ 1 1 4 a によりラッチされそして該マトリックス (図示せず) により該ラムのアドレスバスの次ぎにより上位のビット (the next more significant bit) に接続される。かくして、 A_0 が最下位ビットで A_7 が最上位ビットである A_0 、 A_1 、 A_2 、 \dots 、 A_7 により該アドレスビットをそして D_0 、 D_1 、 D_2 、 \dots 、 D_7 により該データビットを表して、最下位データビット D_0 は該アドレスビット A_1 に接続され、データビット D_1 は該アドレスビット A_2 に接続され以下同様である。最上位データビット D_7 は捨てられるか又はもし幾つかの 1 セルシフトレジスタが図 1 2 に於ける様なカスケード (cascade) で接続されるなら該シフトレジスタの次段に供給される。

30

【 0 1 1 7 】

上記で注意されている様に、該ラム 1 1 1 a 内のデータは、例えば、アドレス " 0 " では該データは " 0 " であり、アドレス " 1 " では該データは " 1 " である等々、各アドレスビットを該対応するデータビットに簡単に転送するようコード化されている。この様な手段により各アドレスライン上の該データは該ラム 1 1 1 a により簡単に出力され、そのラッチ 1 1 4 a によりラッチされる。次のクロックパルス時に、各データビットは該マトリックスにより次のアドレスラインへ供給されそれにより引き続くクロックパルス時に、該最下位アドレスビット a_0 に供給されたデータは該シフトレジスタを通るよう波及 (ripples) する。

40

【 0 1 1 8 】

1 セルシフトレジスタを実施するため使用される代わりの取り組みは、 D_1 がアドレスビット A_1 に接続され、データビット D_2 がアドレスビット A_2 に接続され等々の様に該ラインを直接転送させるよう該マトリックスをプログラムすることである。この場合シフトは該ラムに次のデータをロードすることにより行われる。

【 0 1 1 9 】

アドレス データ

50

0	0
1	2
2	4
3	6
4	8
5	A _H

等々。もし逆シフト (reverse shift) が必要な時は該ラムは下記のデータをロードされる。

【 0 1 2 0 】

アドレス	データ
0	0
1	0
2	1
3	1
4	2
5	2

10

等々。

【 0 1 2 1 】

図 1 2 の該 3 セルシフトレジスタ (3-cell shift register) と図 1 3 の該 1 セルシフトレジスタとの両者では、最下位アドレスラインも又マトリックスに接続されることが理解されよう。しかしながら、該シフト動作を強調するために両図では " 切り離されて (detached) " 示されている。

20

【 0 1 2 2 】

下記点が注意されるべきである。このシフトレジスタの最大速度 (maximum rate) はその周波数が 1 セルの遅延に該マトリックスの遅延のみをプラスするよう斟酌せねばならないクロックにより決定される。又、" レベル " に関する前の議論を参照すると、該デバイスの入力ピン及び出力ピンは、該入力ピンが該マトリックス入力へ、該出力ピンが該マトリックス出力からとなるよう、該 " レベル 1 " プログラム可能なマトリックスに接続される。もし該応用品が " レベル 1 " で実施されるなら、該シフトレジスタの最下位ビットを担う入力ピンは該デバイスの入力ピンの 1 つから回され、そして該出力ピン 1 1 5 は該

30

【 0 1 2 3 】

【例 5 - ノイズ発生器】

ノイズ発生器が図 1 2 で図解された該シフトレジスタに基づいて作られてもよい。当業者には既知の様に、ノイズ発生器は、シフトレジスタの出力の幾つかをイーエックスノア演算する (XNOR-ing) ことと、該シフトレジスタを通して波及させるよう該イーエックスノアの出力 (output of the XNOR) を入力アドレス最下位ビットへ戻すよう供給することにより形成されてもよい。該ラム内のアドレス - データ関係は該イーエックスノア動作を達成するようプログラムされる。同じラムの 2 つのデータビットがイーエックスノアされるべき簡単な場合では、該デバイスの望まれる真理値表に依り該ラム内のデータをプログラムすることは率直である。しかしながら、もしイーエックスノアが 2 つの別々のラムのデータビット間で行われねばならないならば、これらのデータビットの 1 つは他のラムの入力も形成するように該マトリックスにより接続されねばならない。例えば、図 1 2 に示す該シフトレジスタ 1 7 0 で、ラム 1 1 1 a のデータビットの 1 つがラム 1 1 1 b の該ビットの 1 つとイーエックスノアされるべきことを考える。該マトリックス 1 5 5 は該ラム 1 1 1 b の出力 1 1 5 b 内の必要なデータビットを該ラム 1 1 1 a の入力 1 1 2 a に接続すべきであり、該イーエックスノアされた結果は該ラム 1 1 1 a のデータ出力ビット 1 1 5 a の 1 つへ渡されねばならない。結果として該ラム 1 1 1 a は該シフト機能を

40

50

行うべきより少ない残りビット (fewer remaining bits) を有する。

【 0 1 2 4 】

【例 6 - パターン発生器】

パターン発生器 / 信号発生器はもう 1 つの簡単な応用品であり、典型的に、1 つのセルをカウンタとして使用し、該カウンタは、そのデータが該カウンタの状態に依り必要なパターンを発生するようプログラムされる第 2 セルへ回される。かくして、計算されプロットされ得る何等かの関数 $f(t)$ を考えると、該カウンタは t の数列値を出力し、一方該第 2 セルはそのアドレスバスへ供給された t の各値について該関数 $f(t)$ の値を記憶する。該第 2 セルは異なる関数に関するデータをラムの異なる部分に記憶するようコンフィギュアされ得て、これらは該セルの他の空いているアドレスビットへ供給される選択コードにより要求される様な選択となる。プログラム可能な該カウンタセルは異なる信号 (パターン) 用に異なるサイクルをセット出来る。もし該セル内のデータビット量が必要なビット数より少ないなら、該信号のもう 1 つの部分が発生するためにもう 1 つのセルが使用され得て、1 つは該信号の最上位ビット部分が発生し、もう 1 つは最下位ビット部分が発生する。

【 0 1 2 5 】

今度は図 4 に図解された該セル 1 2 0 を使用して種々の例が提示される。

【 0 1 2 6 】

【例 7 - セルのトライステート能力】

図 1 4 は、両方共マトリックス (図示せず) により共通データバス 1 8 2 に接続されたそれぞれのデータ出力バス 1 8 2 a と 1 8 2 b を有する 2 つのラム 1 8 1 a と 1 8 1 b とを備える 1 対のセル 1 8 0 a と 1 8 0 b を略図的に示す。各セル出力はトライステート (tri-state) にドライブされ得る。マルチプレキサーが必要な時該セルのトライステート能力を使うことが勧められる。マルチプレキサーの実施品では、接続されるべき該セルの出力は、該マトリックスを介して、同じ点に回される。該ラム 1 8 1 a はそのそれぞれのアドレスラインが 1 対のラッチ 1 8 4 a と 1 8 4 b によりラッチされる ($n + m$) のアドレスバス 1 8 3 a と 1 8 3 b を有し、一方該ラム 1 8 4 b はそのそれぞれのアドレスラインがラッチ 1 8 4 c によりラッチされる k ビットのアドレスバス 1 8 3 c を有する。該 2 つのラムの該ラッチコンフィギュレーション (latch configuration) の間の外見の差は紙上のみにあると注意されるべきであり、何故なら k は $m + n$ に等しくてもよくて、それで該ラッチ 1 8 4 a と 1 8 4 b は該ラッチ 1 8 4 c と同様な 1 つの k ビットラッチとして動作するからである。該 2 つのラム 1 8 1 a と 1 8 1 b に印加される出力イネーブル信号はどちらの 1 つがアクティブであるかを選択する。該出力イネーブル信号も同様に該マトリックスを介して回される。

【 0 1 2 7 】

図 9 を参照すると、中間列のマトリックス 1 5 5 の数 Z は交差接続の制限を避けるために重要な数である。しかしながら、幾つかのセルの出力が同じ点に接続されるべき時に、ラインはこの中間列で自由にされる。かくして、該セル 1 8 0 a と 1 8 0 b は第 1 列のマトリックスを介してその共通の出力に接続され得て、このマトリックスの該共通な出力が、該第 2 列の 1 つのマトリックスのみを介して該マトリックス 1 5 6 の出力に接続されることを可能にして、かくしてそのより少ないラインを使用する。これは、該セル 1 8 0 a と 1 8 0 b の各々を該中間のマトリックスのそれぞれのラインに接続しそして次いで該交差接続をもたらすよりも好ましい。

【 0 1 2 8 】

該マトリックスは、例えば該セル 1 8 0 a の様な、どんなセルの出力も、例えば該セル 1 8 0 b 内の該ラムの様な、該ブロック内のどんな他のラム、のデータに接続出来るので (どんなラムのデータも又そのセルの出力である)、もし第 1 のセル 1 8 0 a がアクティブな間他のセル 1 8 0 b がトライステートであるなら、該セル 1 8 0 b 内の該ラム 1 8 1 b へ書き込み動作が行われ得る。該書き込みパルスは各サイクルで賦活されてもよいが、該マトリックスから回される該書き込みイネーブルが実際の書き込みをイネーブル又はデイス

10

20

30

40

50

エーブルにする。

【 0 1 2 9 】

【例 8 - ラム - " サーバー " 組合せ】

図 1 5 は、図 1 4 を参照して上記で説明された書き込み能力を開発するサーバー 1 9 1 とラム 1 9 2 を有する " ラム - サーバー " 組合せ 1 9 0 を示す。該サーバー 1 9 1 は 1 つの機能を実行するセルであり、一方該ラム 1 9 2 は次のサーバー動作で使用されるべき該サーバーの結果を記憶する。該サーバー 1 9 1 と該ラム 1 9 2 の間の必要な接続は該マトリックスによりもたらされるが、それは明確なために図では示されない。

【 0 1 3 0 】

該ラム - サーバー組合せ 1 9 0 の使用の例は次の様である。該ラム - サーバー 1 9 1 はバイトの入力ストリーム (input stream of bytes) を受け (起こり得る 2 5 6 の異なるバイトの) 各バイトが何回周期内に現れるかをカウントせねばならない { この例はハフマンコード (Huffman Code) 応用の部分である }。該入力ストリームは該 " ラム - サーバー " 組合せ 1 9 0 の入力 1 9 3 に回され該組合せは次ぎの様に動作する。該ラム 1 9 2 はバイト当たりの結果を記憶し、該サーバー 1 9 1 は該古い結果を 1 だけインクレメントする。この様な手段により、該ラム 1 9 2 は 1 連の 8 ビット数 (0 と 2 5 5 の間) を受けそして該ラム - サーバー 1 9 0 は各 8 ビット数の発生の頻度を示すヒストグラムをコンパイルして該ラム 1 9 2 内に記憶する。

【 0 1 3 1 】

次ぎの動作が起こる。

【 0 1 3 2 】

(a) 現在のバイトが該入力セル - ラム内にサンプルされ、該ラム内に記憶された数が準備される。

【 0 1 3 3 】

(b) この数は該セル - サーバー内にラッチされる。該サーバーは該数を増加するようプログラムされる (図 1 0 に示すカウンターと同様に)。

【 0 1 3 4 】

(c) 該増加したデータは該セル - ラム内に記憶される。

【 0 1 3 5 】

図 1 6 は上記実施用の起こり得るタイミング線図を示し、該動作が、該 " 書き込み " サイクルが新バイトがラッチされるのと同じサイクルで行われるので、唯 2 サイクル内で行われ得ることを示す。該マスタークロックは、 2 サイクル内の該書き込み動作をイネーブルにするために僅かに大きくあるべきなので、このタイミング解決策には実時間の浪費があるが、それにも拘わらずこれは正当化されるがそれは " ラム - サーバー " 組合せは応用品の実施を時には簡単化するからである。図 1 6 で示す全ての必要な信号は、上記で説明した様にパターン発生を行うようコンフィギュアされた他のセルにより発生され、該マトリックスを介して供給される。

【 0 1 3 6 】

もし該デバイスが、上記実施を可能にするには余りに短いクロックサイクルで設計されているならば、図 1 7 を参照してここで説明され、対応するタイミング線図が図 1 8 で示される様に 3 クロックサイクルのラム - サーバー組合せが使用されてもよい。かくして、図 1 7 を参照すると、図 1 5 を参照して上記で説明されたと同じ応用品を実行しそして図 5 のセルと同一のセルラム 2 0 2 に結合されたセル - サーバー 2 0 1 を含むラム - サーバー組合せ 2 0 0 が示されている。この場合、第 3 クロックサイクルで該データは、前のクロックサイクル中の該セル - サーバー 2 0 1 の出力と同じ値で留まるように、該書き込み動作中不変に保持されねばならない。さもなければ、該セル - サーバー 2 0 1 の該ラムの出力データは該セル - サーバー 2 0 1 のアドレスバスに戻るよう供給されるので、それはその中のカウンターにより再びインクレメントされる。従ってこの場合、該第 3 サイクルで該クロック信号はデイスエーブルにされる。しかしながら、この場合、該クロックイネーブル信号は不要にされ (dispensed with) そして代わりに、該第 3 サイクルで無変化をコ

10

20

30

40

50

マンドするよう、図 17 で " 1 を加算 (ADD One) " で示す 1 つの特別なアドレスビットが該セル - サーバ - 201 の入力に供給される。

【0137】

第3のタイミング計画は、書き込み期間が読み出し期間より短くなると期待されるよう該ラムアドレスのコード化が該書き込み動作の前に準備される事実に基づいている。従って、2サイクル動作の方が好ましい。しかしながら、図 18 で図解される様に3サイクル動作は実施出来る。該クロック " ローツーハイ (low-to-high) " 遷移が該読み出し動作の終了後にあるよう " 移され (moved) " 得て、より速いクロックサイクルを創る。上述の様に、該入力信号は他のセルにより発生され該マトリックスにより接続されることが出来る。

10

【0138】

図 17 を参照して上記で説明されたことから、該クロックイネーブルは該ラム - サーバ - 201 用には本質的でないこと従って例え該デバイス内にクロックイネーブルが提供されなくても、該ラム - サーバ - 201 はなお実施され得ることは明らかである。クロックイネーブルを提供するか否かに関する決定は該クロックイネーブル用に各セルにもう1つのロジックのアンドゲートを追加する必要にそれ程依らず、むしろ該プログラム可能なマトリックスのサイズを増加する必要に依る。

【0139】

より速いクロックサイクルを創ることは幾つかの応用品がより速く実行されるようにさせるが、もしこれが図 17 に示す該 " ラム - サーバ " 応用品の場合であるなら、該入力はその3サイクルクロックに同期化されねばならず、それでそれは実行がより遅くなると思われる。

20

【0140】

【例9 - タイムシェアリング】

応用品を実施するためには、普通幾つかのセルが必要である。タイムシェアリングでは、同じ動作が同じ入力ポート上でシーケンシャルに実施品に回される幾つかの独立な入力用に行われる。タイムシェアリング応用品を容易に実施するために、図 15 又は図 17 を参照して上記で説明された様な、該 " ラム - サーバ " 組合せは、該 " サーバ " がロジックを実行するセルとして使用され、該ラムが各スロットの状態を記憶するため使用される、ロジックを行う何等かの単一セルに置き換わるべきである。各ラムの該入力はタイムシェアリングを支配する応用品のスロット / チャンネルをカウントする1つの大域的カウンタに接続される。該カウンタは該セルの1つであるか、又は該タイムシェアリングされた実施を実行する該 " ラム - サーバ " 組合せの全ての該ラムへ該プログラム可能なマトリックスを経由して回される特製カウンタ (special purpose counter) とすることが出来る。好ましいアーキテクチャでは、特製カウンタは無くても、それでもタイムシェアリングが必要ならば、カウンタ応用品 (1つのセル又は2, 3のセル) が実施されるべきである。

30

【0141】

図 19 はタイムシェアリングカウンタ 211 を利用してタイムシェアリングモードで動作するシフトレジスタ 210 についての例である。該シフトレジスタ 210 は2つのラム - サーバ 212 及び 214 そして 213 及び 215 を含み、それらは図 12 を参照して上記で説明されたそれと同様な仕方で該組み合わせられたシフト動作の、それぞれ、下部及び上部部分を行い、該ラム 214 の最上位ビット該サーバ 213 の入力に供給される。それぞれのタイムシェアリングセル 214 及び 215 は、該シフトレジスタサンプルの各スロットに関するそれぞれロー及びハイ部分を記憶するために該サーバ 212 及び 213 に接続される。該ラム - サーバ 212, 214 及び 213, 215 は図 15 又は図 17 を参照し、そしてそれらのタイミング線図は図 16 及び図 18 を参照して上記で説明した様に作動する。該タイムシェアリングカウンタ 211 は、それぞれ図 15 及び図 17 で示すラムサーバ 190 及び 200 の入力バイトの接続と同様な仕方で該セル 214 及び 215 のラッチに接続される。全体の動作は図 16 を参照し上記で図解されそし

40

50

て図 20 を参照して下記で更に展示される様にもし好ましい " 2 サイクル " 動作が選ばれた場合のみ 2 サイクルを有する。

【 0 1 4 2 】

該応用品の動作は次の様である。該 2 つの " サーバー " セル 2 1 2 及び 2 1 3 は上記で説明した様にシフトレジスタ処理 (shift register implementation) を実施する (implementation)。該タイムシェアリングカウンタ 2 1 1 は該タイムシェアリングスロット / チャネルをカウントする大域的カウンタとして動作する。

【 0 1 4 3 】

図 20 はタイミング動作を示すタイミング線図であり、そこでは " N 1 " は " 1 だけシフト " をシンボル化し (symbolizes)、下記動作が起こることを示す。

10

【 0 1 4 4 】

(a) 同じクロックが該大域的カウンタ用にそして該残りの処理 (the rest of the implementation) 用のマスタークロック信号を提供するために使用されるので、該大域的カウンタはその最下位ビットを該タイムシェアリングセル 2 1 4 および 2 1 5 に回らせるべきでない。かくして、該大域的カウンタ上の該 ' n ' の最上位ラインは 2 サイクルカウンタを表す。該最下位ビットは該タイムシェアリングセル 2 1 4 及び 2 1 5 と該ラム - サーバー 2 1 2 及び 2 1 3 の ' OE ' を制御するため使用出来る。

【 0 1 4 5 】

(b) 図 10 を参照すると 1 つより多いセルが該カウンタを実施するため使用出来る (もしデータビットの数が必要な該 ' n + 1 ' ビットの数より少ないならば)。

20

【 0 1 4 6 】

(c) 第 1 サイクル時、該古いデータが読み出されるべき時、該タイムシェアリングセル 2 1 4 及び 2 1 5 の出力はイネーブルとなり、該ラム - サーバー 2 1 2 及び 2 1 3 の出力はデイスエーブルとなる。

【 0 1 4 7 】

(d) 次のサイクル時、該タイムシェアリングセル 2 1 4 及び 2 1 5 の出力はデイスエーブルであり一方それらの書き込みイネーブルはイネーブルとなり、該ラム - サーバー 2 1 2 及び 2 1 3 の出力はイネーブルとなる。結果として、それぞれのラム - サーバー 2 1 2 及び 2 1 3 の現在のステータスに対応するデータが今度は該タイムシェアリングカウンタ 2 1 1 の値によりアドレスされるそのメモリー位置で該タイムシェアリングセル 2 1 4 及び 2 1 5 に書き込まれる。

30

【 0 1 4 8 】

(e) 該大域的カウンタとしてサービスする該タイムシェアリングカウンタ 2 1 1 は両サイクル中イネーブルである。

【 0 1 4 9 】

【 例 10 - タイムシェアリング環境での " ラム - サーバー " 】

図 21 は図 15 を参照して上記で説明された例を実施するためにタイムシェアリング環境で動作する " ラム - サーバー " 組合せ 2 2 0 を示す。該ラム - サーバー組合せ 2 2 0 はセル - サーバー 2 2 1 と、図面の図 15 を参照して上記で説明された様に機能するラム 2 2 3 とラッチ 2 2 4 とを有するセル - ラム 2 2 2 とを備える。大域的タイムシェアリングカウンタ 2 2 5 は該セル - ラム 2 2 2 のみに回される。例として、タイムシェアリングは図面の図 15 を参照して上記で説明される該ラム - サーバー 1 9 0 に追加されるが、それは該セル - ラム 2 2 2 の該ラムの別の範囲がそれぞれのタイムシェアリングスロットに専用化されることを可能にするためである。

40

【 0 1 5 0 】

次の動作が行われる。

【 0 1 5 1 】

(a) 該大域的カウンタ 2 2 5 がインクレメントされ、その ' n ' 最上位ビットは、データが読み出されるべきラム 2 2 3 内の該範囲を規定する (すなわち該アドレスの ' n ' 最上位ビット)、現在のスロットを規定する。

50

【 0 1 5 2 】

(b) 該規定された範囲内の該メモリー位置の該ラム 2 2 3 内に記憶された番号は該ラム 2 2 3 へ該ラッチ 2 2 4 により供給されるアドレスの該 ' k ' 最下位ビットにより指摘されている。該 " サーバー " 2 2 1 は該番号をインCREMENTするようプログラムされている。

【 0 1 5 3 】

(c) 該INCREMENTされた番号は、該タイムシェアリングカウンタと該ラッチ入力の値によりアドレスされたそのメモリー位置で該セル - ラム 2 2 2 内に記憶される。

【 0 1 5 4 】

この繰り返し動作は該書き込みが新しいバイトと同じサイクルで行われ得るので 2 サイクル行われ得て、該タイムシェアリングカウンタはラッチされる。

10

【 0 1 5 5 】

【例 1 1 - より多量のデータビットを有するセルの創生】

セルのアドレスバスのサイズを増加するためにセルが如何に 2 つのセルから形成されるかを図 6 を参照して上記で説明した。図 2 2 はセルのデータバスのサイズを増加するためセル 2 3 0 が如何に 2 つのセル 2 3 1 と 2 3 2 から形成されるかを示す。該 2 つのセル 2 3 1 と 2 3 2 のアドレスバスは並列に接続され、それにより両セル 2 3 1 と 2 3 2 のアドレスは同一である。両セル 2 3 1 と 2 3 2 は望まれる機能の結果が該 2 つのセルの間に分かれるようにプログラムされる。必要ならば、2 つより多いセルのアドレスバスが並列に接続され、各々が機能出力の異なる部分を取り扱うようプログラムされる。該全ての接続は該プログラム可能なマトリックスを利用して作られる。もし図 9 に示すマトリックスが使用されれば、該マトリックスを利用して必要な接続を創る多くの方法があることを注意する。もしリンクされるセルが該 ' X ' 列の同じ飽和型マトリックスに接続されるなら、該接続はそのマトリックス内で直接行われ得て、該 ' Y ' 又は ' Z ' スイッチからのスイッチを使用しないことによりマトリックス源を節約する。又該接続は ' Z ' 列でも行われ得て、再び節約するか、又は該接続は該 ' Y ' 列で行われ得る。

20

【 0 1 5 6 】

厳密には、該データが実際は 1 つより多いセル間に分割されるので、そのデータビットの全てが既に使用されつつあるセルにデータビットを追加することは該セルサイズを増加することによっては達成され得ないことは議論されるかも知れない。しかしながら、複合セルと等価大型セルの構成セル間では区別は行われない。

30

【 0 1 5 7 】

【ロード動作】

セル内に該コードをロードするには幾つかの方法がある。たとえば、マトリックスは該全デバイスが 1 つのラムとしてコンフィギュアされる方法でプログラムされてもよい。次いでデータはこのラム内に、どんなメモリーでも書き込まれるべき同じ仕方でホストコンピュータ又は他の電子的実施品によりロードされ、その後該マトリックスは必様な接続をもたらすよう再度プログラムされてもよい。

【 0 1 5 8 】

図 2 3 は、ホストコンピュータ (host computer) が該ラムの全アドレス空間を写像することが出来ない場合にその中にデータをロードするために、又はデバッグする目的でコンフィギュアされたデバイス 2 4 0 を示す。この目的で、該デバイスは幾つかのラムに分けられており、その中の 3 つは 2 4 1 , 2 4 2 そして 2 4 3 及びそれぞれのラッチ 2 4 4 , 2 4 5 そして 2 4 6 で表されて示されそしてそれらはプログラム可能なマトリックス 2 5 0 を介して該ホストコンピュータに接続されている。補助回路 2 5 1 はそのラッチ 2 4 4 , 2 4 5 そして 2 4 6 に接続され、そのラッチへクロック - イネーブル (clock-enable) { シーイー (CE) } 信号を供給するための該ホストコンピュータによりそれへ供給されるロジック信号に応答する。

40

【 0 1 5 9 】

マトリックス制御メモリー (図示せず) のアドレスとデータビットはデバイス 2 4 0 のピ

50

ンアウト (pin-out) に接続される。この接続は該ホストコンピュータに、該ホストコンピュータが該マトリックス制御メモリの該メモリをそのメモリ範囲内に写像しそしてそれにプログラムされた接続をダウンロードする、一方該シーイー信号は該ラム間を " ページ (page) " するよう使用されるように、該マトリックス 250 にプログラム (制御) させる。

【0160】

該ロードに使用されるピンは、スイッチ範囲を含まない応用品用には、アイノオー用に使
用されると同じピンとすることが出来る。これは該マトリックス制御メモリへの書き込
みパルスが無い時は該スイッチ状態には変更が行われないからである。この様な場合、該
ロード中 (すなわちスイッチアウトされている) 該アイノオーピンは遮断されねばならな
い (すなわちスイッチアウトされる)。

10

【0161】

該補助回路 251 はデバッグ中該ラッチのクロックをイネーブル又はデイスエーブルにす
るため使用される。もしクロックイネーブル (シーイー) 信号が該マトリックス 250 に
接続されるなら、該クロックイネーブル (シーイー) 信号は該マトリックスを経由して回
され得て、各セルが別々にクロックされ、望まれる状態がロードされることを可能にする
ので、該補助回路 251 は不要である。

【0162】

典型的には実時間で作動するよう設計されることはないので、このロジック用に実時間の
考慮は行われない。しかしもし実時間の " クロックイネーブル " 信号が必要なら、これを
該 " レベル 1 マトリックス " 内に集積化することが勧められる。

20

【0163】

【好ましいアーキテクチャーを選ぶための考慮】

ダイのサイズは重要なパラメーターである。該セルが小さくそして該ブロックが大きい程
、柔軟性は大きく、ユーシジ (usage) は良くそして性能は良いが、該マトリックスはよ
り大きくなる。該マトリックスのサイズを減じるために、レベルが付加される。上記の様
に、1つのレベルの全てのセルがこの能力を有する必要はないことは注意されるべきであ
るが、セルはより低いレベルの他のセルから作られ得る。

【0164】

0.25 ミクロン (micron) 技術用の可能なアーキテクチャーの第 1 例はレベル 1 での 4
096 バイト (12 ビットアドレス、8 ビットデータ) の 300 セルと、レベル 2 での各
ブロックの 512 バイト (9 ビットアドレス、8 ビットデータ) の 64 セルとである。又
第 2 例は 2048 バイトの 64 セルから 512 バイトの 128 セルへレベル 2 のブロック
を形成することである。レベル 2 では 10 以下のブロックが実施された。レベル 0 では全
てのセルが 1つのセルに変換された (1つのメモリ又は 1つのルックアップテーブル)
。

30

【0165】

可能なアーキテクチャーの第 3 例は 1つのレベルだけの 64 キロバイトの 20 セルに基づ
いている。再び、レベル 0 では全てのセルが 1つのセルに変換されるので、この例では、
該マトリックスは小さくそして従って飽和され得る。これらの 3つの例は可能なアーキテ
クチャー間の幾つかの差を示す。これらの 3つの例では、ダイのサイズは概略同じである
。提案された最初の 2つのアーキテクチャーでは該レベル 1 は 64 キロバイトセルの約 1
8 メモリのみであるが、多数のより小さいセルがあるので、該ユーシジ (usage) はよ
り良く、全体の結果は大抵の応用品用により良いと期待される。

40

【0166】

【セル接続例を利用した実時間計算】

0.25 ミクロン技術では、1つのセルのサイクル時間は典型的に 10 ns より短いと期
待され、かくしてそれは下記計算での例として取られている。全てのセルは同期化され並
列に作動する。従って、10 ns の間に、該デバイス内のセルの各々は 1 コマンドを行う
。これは、該セルにより行われる平行コマンドの全てが 1 命令 (instruction) と考えれ

50

ば、該処理速度は100ミップス(MIPS)(秒当たり1億命令)である。

【0167】

図24は、8ビットコマンドを行うよう動作しそして普通に使用される計算に基づき実際の処理力(actual processing power)へのより良い参考事項を提供するため役立つデバイス260を略図的に示す。このコンフィギュレーションで、該デバイス260は8ビットコマンド、ラッチ261への入力、を行いそして前のコマンドの8ビット結果に関係することが出来る。

【0168】

64キロバイトアーキテクチャーで、サイズ4キロバイトの300セルは16ビット(すなわち64キロ)アドレスバスと8ビットデータバスを有するセルに変換されると仮定すると、タイムシェアリング及び他の小さなタスク用に使用出来る他のスペアのセルを残して18のこの様なセルがある。該18のセルの各々が8ビットコマンドを行うことが出来ると仮定すると、完全なデバイス260は1800ミップスを行うことが出来て、そこでは各命令は8ビットである。実際には、全てのセルが一般に64キロ×8の様に大きく必ずしもなく、そして又平行してコマンドを行う多数の小さなセルもあり、かくしてミップスの有効数をなお更に増加させる。

【0169】

動作速度(ミップス)が高い程、該多数のセルのために遅延も大きいことが注意されるべきである。例として、図12に示すシフトレジスターを参照して上記で説明されたノイズ発生器を考える。唯3n-1サイクル後に(すなわち、もしn=8なら、23サイクル後に)、該シフトレジスターは、その出力が必要なノイズ信号として使用されてもよいように、入力ビットで満ちる。かくして、各サイクルが10ns持続時間であると仮定すると、全遅延は230nsである。その後、10ns毎に新しいノイズサンプルが出力される。この種の遅延はハードウェア及びデーエスピー実施品、例えばフィルタの実施品では普通である。

【0170】

【異なるセルコンフィギュレーション】

該デバイスの前記説明では、全てのセルはラム及びラッチの両者を有した。それにも拘わらず、該デバイスアーキテクチャーは上記説明のそれらに対し異なる構造のセルを使用して実施されてもよい。例えば、フィードバックが必要でないならば、該デバイス100上で走る(running)応用品(図2参照)はそのラッチを有しない同じデバイス上でより速く走る。

【0171】

【例12-スタンドアローン(standalone)ラムセル】

図25はラム271と、図5を参照して上記で説明した付随補助回路272とを含むセル270を示す。図26はこの様なセルを使用し、そして2つのオペランド(operands)A及びBを加算するための1対の8ビット入力データバス276及び277と該加算の結果を担う9ビット出力データバスとを有する8ビット加算器275を示す。この様な応用品では、該出力は単に該2つの入力A及びBの状態を反映するのでタイミングは要せず従ってラッチは要しない。

【0172】

図26に示す該8ビット加算器275は、2つの8ビット入力の9ビット和は該マトリックスによりコンフィギュアされた必要サイズの1つのラムのみの中へプログラムされる、基本的解決策を図解する。この様なコンフィギュレーションでの該応用品はラムの9ビットワードの64キロの等価品を使用する。該解決策は使用されるラムの量を減じよう最適化され得る。

【0173】

図27はより良く最適化された加算器280を図解するが、それは各々が18ビットの2つの入力を加算し、そしてその出力に19ビット和を作る。図26に示す1つのラムを使用する代わりに、複数の第1段セル281, 282, 283そして284が、見られる様

10

20

30

40

50

に、5ビットと3ビットの要素オペランドに分割された18ビット入力を加算し、そして1つ又は2つの繰り上げ(carry-out)ビットを発生するようプログラムされる。該繰り上げ及び適当な第1段出力は次いで適当な第2段セル285, 286そして287を使用して加算される。この様な配置は、該加算をもたらしするために使用される該2段のために、該入力オペランドの供給と該結果の出力との間に2段の遅延がある。しかしながら、最下位ビットは該第1段加算器281を経由して直接供給されそして追加的遅延はこうむらない。もしそれが残りのビットと同じ2段遅延をこうむる必要があれば、それが1つの追加的遅延をこうむり、そしてそれにより残りの出力ビットの遅延に整合させるためにそれは該第1段加算器284を通して直接供給されることも出来る。

【0174】

10

この様な解決策は図5と図25を参照して上記で説明した両デバイス用に好適である。該18ビット加算器が図27で示されるようにコンフィギュアされる時は、応用品は、上記説明の理由で該非最適化解決策に於けるよりも1つ多い遅延があるに拘わらず、7キロバイトより少ない、すなわち、図26で示される該非最適化8ビット加算器より著しく少ない、ラムの等価物を使用する。

【0175】

【例13 - 構成部品から形成されるデバイス】

前の該遅延の例又は図12に示す該シフトレジスタでは、該セル内のラッチのみが利用され、該ラムのためのニーズはない。これと対照的に、図25を参照して上記で説明した様に、ラムは必要だが、ラッチは不要の応用品が存在する。従って、該セルの幾つかは各部品が別々に使用されるようそれらの部品に分解される。元のセル構造は該マトリックスを使用して該部品を組み合わせることによりなお実現され得る。もし望むならば、セルは追加的構成部品を含むことが出来る。

20

【0176】

図28は図25に示すラム270から独立に使用されてもよいが、図5に示される該セルを実現するために、もし必要ならば、それらと組み合わせられてもよいラッチ290の例を示す。

【0177】

【例14 - 独立なクロックを有するデバイス】

上記説明の様に、ブロックはデバイスと考えるてもよい。従って、完全なデバイス用の1つのクロックを有することは本質的でなく、各ブロックはそれ自身の独立なクロックを有してもよい。又前に説明した様に、該ブロックは、各セルがそれ自身の独立なクロックを有するよう、セル内へ変換されてもよい。

30

【0178】

もし異なる種類の2つのブロックが1つのデバイス内に組み合わせられたならば、1つのクロックは1つのブロック用に提供されるがもう1つ用ではない。従って、必要なクロック応答の十分な種類を提供するために多数のクロック入力が該マトリックスを経由して該デバイスの入力から何れかのブロック又はセルへ回されてもよい。

【0179】

【例15 - 単一ステップ動作】

40

単一ステップ動作のニーズはデバッグ目的用であり、該実施品の状態を動作の何れステップについてもトレースされることを可能にする。この様な特徴は本発明の該デバイスを使用したハードウェア応用品が、ソフトウェアプログラムがデバッガ(debugger)を使用してデバッグされる方法と同様な仕方でデバッグされることを可能にする。必要な時は何時でもソフトウェアプログラムが止められ、その変数が追跡、変更され、そしてそれが止められた所から又は該プログラム内の何等かの他の点からか何れかで再スタート(restarted)されると同じ方法で、同様な仕方で該ハードウェアデバイスをデバッグするためのツールを本発明は提供する。

【0180】

もしデバイスが、例えば、図27に示す加算器に於ける様にクロックを有しない種類であ

50

り、そして該実施品がフィードバックを有しないならば、該入力を単一ステップ化 (single stepping) することにより単一ステップ動作が達成される。もし該実施品がフィードバックを有するなら、該定常状態のみが追加的ツールを使用することなく追跡出来る。もう1つの可能性は該セルを現在のステータスが保持されるメモリーのもう1つの範囲へ導くためにアドレスビットを使うことである。

【0181】

もしデバイスがクロックを有しそして1つのクロック動作が必要ならば、ゆっくりした入力制御と速いハードウェア応答の間の同期化が達成されねばならない。1つの可能性は該デバイス用クロックとしてゆっくりした外部クロックを使用することである。

【0182】

図29は該クロックが外部的に供給され得ないもう1つの解決策を示し、該クロックイネーブル信号 (シーイー) は有効なクロック速度を調整するため使用される。該セルは次の状態を有するようプログラムされる。

【0183】

該入力 (x y) で: x は該外部シーイー信号、y は該フィードバック、
該出力 (a b) で: a は該内部シーイー信号、b = y は該フィードバックである。

【0184】

入力 出力

0 0	0 0	外部クロックイネーブルなし、内部クロックをイネーブルにせず
0 1	0 0	外部クロックイネーブルなし、内部クロックをイネーブルにせず
1 0	1 1	外部クロックイネーブルをスタート、内部クロックをイネーブルにする。

【0185】

1 1 0 1 外部クロックイネーブルを継続、内部クロックをイネーブルにせず。

【0186】

これは外部クロックイネーブルの各長いパルス用に、1つのクロックパルス用の等価の唯1つのパルスが発生されることを保証する。

【0187】

【例16 - 2重バッファアーメモリー】

上記で説明された大抵の実時間応用品は固定マトリックス交差接続に基づいている。回し方 (routing) を変更する時間は典型的デバイス目標の実時間の連続動作用には長過ぎる。しかしながら、新しい種類の応用品はホストに接続された1つより多いデバイスを利用することにより達成され得る。例えば、その各々が必要な動作を実行するようプログラムされるべき3つのデバイスがホストに接続されている状況を考える。該ホストコンピュータはメモリーを有し、該メモリーはそれが動作出来るようにするよう各デバイスにロードされねばならないコードを記憶している。1つのデバイスが該応用品を実行している間、第2は、前の結果がそこから読み出される又は新しいコードがそれにロードされることを可能にする普通メモリーとして該ホストに接続され、そして同時に該必要な接続をもたらしそして該必要なデータをロードするためのコードは第3のデバイスにロードされてもよい。現在走っているデバイスによる応用品が完了すると、各デバイスの状態は変化し、そしてホストは、結果を読み出しそして新しい実施品をロードするためにそのタスクを丁度完了したデバイスに接続される。この様な手段により、各実施品は、ソフトウェアを使用して普通のシーピーユー内で行われ得るより遙かに速いハードウェア内にもたらされ、一方時間のかかる、該デバイスへの新しいデータのダウンロード作業は異なるデバイスの動作で平行して行われ、それなので、実時間で何等のオーバーヘッドを要しない透明な動作を表す。

【0188】

上記実施では幾つかのデバイスが使用されるが、該実施は又同じデバイス内の2, 3グループのセルを使用しても行われ得る。もし2つのデバイスグループのみがアクティブなら、メモリーだけを "見る (sees)" 該ホストの遠望からは、それは2重バッファアーハードウェア実施 (double-buffer hardware implementation) に似ている。

【 0 1 8 9 】

この技術は、もし非常に速い実時間の計算が必要なら有用であり得る（コンフィギュア可能なコンピュータに於ける様に）。該ホストコンピュータは実施されるべきタスクを1つのデバイス上で必要な様に変更し、一方もう1つのデバイス内でそれは前の命令の結果をフエッチ（fetches）する。

【 0 1 9 0 】

この様な応用品の実際の実施品は各々がアクチブ（active）とイナクティブ（inactive）の状態を有する少なくとも1つのデバイスを備える。該ホストは各デバイスに結合され、メモリーを有するが、該メモリーは、該アクチブ状態にある時該それぞれのデバイスが必要な動作を実行出来るようにするか又は該イナクティブ状態にある時該ホストが該それぞれのデバイスにロードすることを可能にするように、該デバイスの各々内にロードされねばならないそれぞれのフォーマットされたデータをその中に記憶する。これはコンフィギュア可能なコンピュータが少なくとも2つのデバイスを使用して実施されることを可能にするが、そこでは該ホストは各タスクを実行するために必要な多さのデバイスを賦活することにより少なくとも1つのタスクを管理するよう適合されている。

【 0 1 9 1 】

【 例 1 7 - 多数 " エムシーエム (MCM) " アーキテクチャー 】

図9を参照して上記で説明された非飽和型マトリックス及び標題 " 好ましいアーキテクチャーを選ぶための考慮 " の下で上記議論された例には、約100万（1M）のスイッチがある。これは該マトリックスが全ダイ面積の約10%を占めることを意味する。図30は多数エムシーエム（multiple MCM）アーキテクチャーを略図的に示すが該アーキテクチャーは、2つのマトリックス制御メモリーの該ビット A_1 、 B_1 、 A_2 、 B_2 ... A_N 、 B_N の各々間を多重化する（multiplexing）ための1つの制御部299により全てが動作させられるスイッチモジュール298内のそれぞれの2路開閉器（two-way switches）297により選択される A_1 、 B_1 、 A_2 、 B_2 から A_N 、 B_N までとして表される該エムシーエム内のそれぞれの対のスイッチを経由して該マトリックス296内の各スイッチ295が開かれ又は閉じられるように追加的マトリックス制御メモリーを提供することにより作られる。該エムシーエムの各々は、それぞれの接続を規定するようデータでロードされ、それにより種々の予めコンフィギュアされた接続が望ましいエムシーエムを選択するように該制御を簡単に動作させることにより実質的に瞬間的に実施され得て、かくしてより速い実時間動作を可能にする。又このアーキテクチャーはデバイス内の同じセルの再使用を可能にするので、それらは応用品を実施するためそして該コンピュータのメモリーとして使用されてもよい。タイミング又はホストコンピュータの命令又は応用品の命令（例えば、該タスクが完了した時）は、該セルが該応用品にサービスする期間{ " ワーキング期間（working period） " }か、又は該ホストコンピュータにそのメモリーとしてサービスする期間の、それぞれの期間の間で切り替わってもよい。各ワーキング期間の後、該ホストコンピュータはそのメモリーにアクセスする許可を受ける。

【 0 1 9 2 】

このアーキテクチャーは該マトリックスと該マトリックス制御部品の集合したサイズを約4の倍数で増大させる。0.25ミクロン技術が使用され、そして0.18及び0.13ミクロンのジオメトリー（geometry）が現在使用可能と仮定すると、この例は実行可能な設計である。

【 0 1 9 3 】

このアーキテクチャーは例16で上記で説明した2重バッファアーメモリーと同じユーシジを達成出来る。上記説明した両例16及び17でホストコンピュータの使用が説明されたが、該デバイスを利用するモジュールは同じ機能を実現出来る。

【 0 1 9 4 】

【 デバッグポート 】

プログラム可能なマトリックスに接続されたポートはどんなセル出力も、該セル出力を該マトリックス経由で該ポートまで回すことにより見る事が出来る。セル入力を見るため

10

20

30

40

50

には、小さな自由セルが使われることが可能である（或いはセルとして作用する専用化されたラッチ）。図 1 2 のシフトレジスタで説明した様にそのセルでのデータは該入力を該出力に転送するようセットされ、該セル出力は次いで該ポートへ回送される。

【 0 1 9 5 】

このポートは、実時間動作で、該全体デバイスのどんな入力又は出力でも見られるようセットされ得る。

【 0 1 9 6 】

注意：

- 1．大抵の場合、該セルの入力は、恰もこの入力は何れかの出力に接続され、該出力が該ポートに回され得るなら、見る必要はない。もし該ポートが該デバイスのアイノオーピンに接続されていれば - それは該マトリックスにより該ポートへ回送され得る。
- 2．より低いレベルについては：該より低いレベルの " ポート " のみが該アイノオーピンで実時間で見られ得る。

【 0 1 9 7 】

【単一ステップ】

該デバイスをランニング（running）から停止させるために、全ての必要なことは該セルへのクロックを停止させることである。もし該ロジック操作の何れかが取り込まれ、そしてユーザーがこのロジックをインサーキットエミュレーションモード（In Circuit Emulation mode）で該ハードウェアの動作を停止させるようセットするなら、該応用品は停止する。もしユーザーが欲するなら、該マトリックス制御メモリーは該マトリックスにどれかのセルの該アイノオーピン内へのルートを決めさせるよう変更され得る。又それはメモリーに変換され、普通のメモリーとして読み出される得る。この方法で、該デバイス全体内の何等かの単一ロジックビットを見ることが出来る。該テストの後、該マトリックス制御メモリーはその元の状態へ戻るようセットされ得て、該動作はそれが停止した同じ点から継続してもよい。

注意：

該シミュレーションが停止する時、より低いレベルのそれらを含み、該デバイス全体内の何等かの単一ロジックビットを、恰もそのレベルがメモリーに変換され、全ての該出力セルは該レベルのポートに接続された様に、見ることが出来る。見られるべきビットは新しく創られたメモリーのアドレスにより選ばれる。

【 0 1 9 8 】

【技術の適合】

1つのチップ上に益々多くの機能を集積する半導体工業の能力はいつも増加している。該デバイスの技術はセルに基づき、使用されるセルが多い程、該応用品の能力は大きい。そのアーキテクチャーはセルがより大きいセルを形成するよう組み合わせられるようにされているので、下記方法は、より大きいダイ利用が可能になった時、何れの実施品も新技術上で自動的に作動することを保証する：

- 1．セルの数を減らさない。
- 2．該マトリックスの交差接続制限を増やさない（すなわち、図 9 で示す上記例で Z を減じない）。
- 3．小さいセルサイズはより大きいセルサイズで置き換えられてもよい。
- 4．より低いレベルが追加されてもよい（より小さいセル用にレベルを追加する）。該より低いレベルのセルはより高いセルへ戻るよう変換され得ることは注意されるべきである。
- 5．クロックサイクル周期を増やさない。
- 6．セルのブロックはより低いレベルからより高いレベルへ動かされてもよく、かくして該マトリックスサイズが増やされることが求められる。
- 7．入力ピン数を減らさない。
- 8．出力ピン数を減らさない。

【 0 1 9 9 】

事実、もし技術が改良されるならデバイス能力は減らされるべきでないことは上記から明らかである。これを考えることはどんな応用品も新しい技術と自動的に両立出来ることを保証する。又、シミュレーションで、機能はセルに分けられるので、該シミュレーション自身は変化せぬことは注意されるべきである。しかしながら、（マトリックス制御メモリーの）メモリー構造とデバイス全体の「メモリー画像（memory image）」との両者が変化するので該ロードされるプログラムは変化せねばならない。換言すれば、必要な全てのことはセルにロードされるデータが再フォーマット（reformatted）されねばならぬことである。

【 0 2 0 0 】

【ハードウェアのテスト】

実施品は該ハードウェアが適当に働くことを確認するためにテストされる必要はない。全てのラムをテストするためには、デバイス全体が1つのラムに変換され、全てのその位置（all its locations）は、従来型メモリーでなされる様に読み出し／書き込み動作を使用してチェックされる。例えば、「00」、「AA」、「55」、「FF」、「データ内アドレス」等。該マトリックス接続は今度は接続信号により全ての起こり得る接続を経由してチェックされる。同じ仕方で、全てのアイ／オーピンがチェックされる。

【 0 2 0 1 】

上記は応用のないハードウェアの全てをテストする能力の図解である。これを行う他の方法がある。例えば、信号はパターン発生器を使用して内部的に発生出来て該デバイス自身でテストされ得る。

【 0 2 0 2 】

【スタンドアロンのカード】

スタンドアロンのカードが必要な時、例えば、もしホストが利用出来なければ、デーエスピー分野に共通な同じ解決策がここでも良い。例えば：

消去可能でプログラム可能な読み出し専用メモリー { イーピーロム（EPROM） }

1．該マトリックス制御メモリー用及び該セル用のデータを有するイーピーロムが該カード上に設置される。一旦電力がオンに替わると、該デバイスは該マトリックス制御データを該イーピーロムから該マトリックス制御メモリーへコピーする。メモリーからメモリーへコピーする電子回路実施品は当該技術で知られている。

2．一旦最後のビットが該マトリックス制御メモリーにコピーされると、該デバイスは1つのメモリーの様に見える。該電子回路は今度は該イーピーロムから新しく創られたメモリーへコピーする。該セルはかくしてそれらのデータをロードされる。

3．最後に、該電子回路は該応用品用に必要な交差接続をセットするために、該イーピーロムから再び該マトリックス制御メモリーへコピーする。

【 0 2 0 3 】

もし使用される該デバイスアーキテクチャーが、該標題「好ましいアーキテクチャーを選ぶための考慮」の下の例で論じたと同じであるなら、該イーピーロムサイズは約10メガビットである。

他の解決策

該イーピーロム（EPROM）が使用されると同じ仕方で、当該技術で公知の他の解決策、例えば、バックアップバッテリーを有するイーイーピーロム（EEPROM）又はラムが使用出来る。該イーピーロムを該デバイス内に集積化することは、それが該デバイスの製造コストを増加させるので勧められない。

【 0 2 0 4 】

直接に該デバイス用となる、バッテリーバックアップは、もしそれが該デバイスがアクティブでない時だけ該バッテリーがアクティブであることを確認するならば、有用である。さもなければ、電力消費が多い。

【 0 2 0 5 】

【自己永久運動的応用品】

（Self-Perpetuating Application）開発

10

20

30

40

50

該ハードウェア応用品が記憶され、ソフトウェアコードとして再使用され得るので、再使用され得る各タスクは記憶され必要時にコールされることが可能である。例えば、もしフィルターが実施されデバッグされるなら、プログラムは該フィルターパラメーターを求めそして該デバイスコードを自動的に発生する。これは、該デバイス技術が多く使用される程、作業グループ (work-groups) に利用可能な在庫タスク実施品の数 (the number of "off-the-shelf" task implementations) が大きくなり、市販までの時間が短くなる、ことを意味する。更に、上記説明の様に該チップ製造技術が改善された時応用品のライブラリーはなお使用可能である。

【 0 2 0 6 】

【該デバイスの補足部品】

該デバイスアーキテクチャーに直接関係しない補足部品が、下記のために該デバイス上に提供されてもよい。

1. 同期化されたクロック及びタイミング用に内部クロックを発生する、
2. 外部クロックの使用をイネーブルにする、
3. "書き込みサイクル" 用タイミングを創る。これは固定タイミングであり、それは該書き込み動作が、該マトリックスから来る該 "書き込みイネーブル" 及び該セルの出力イネーブルの利用によりイネーブル又はデイスエーブルにされるからである。従って、それはマスタクロックからドライブされ得る。
4. 該アイ / オー信号をバッファする。

【 0 2 0 7 】

これらの機能を如何に実施するかは当該技術で公知である。

【 0 2 0 8 】

【該デバイスを実施する他の方法】

該デバイスアーキテクチャーの利点の幾つかは図 5 に示す様な該セル入力信号と該セル出力信号が同期化されている事実と関係する。しかしながら、幾つかの非同期動作も実施出来る。例えば、小さなセルは余りに速いのでそれらのサイクル周期は普通のセルのサイクル周期の半分であることを考える。これらのより速いセルはより速いクロックを有するより小さいブロック内に接続され得る。それらのサイクル周期は普通のセルのサイクル周期の半分であるので、これらのブロックの出力は残りのブロックに容易に同期化出来る。

【 0 2 0 9 】

より速いセルはより高いレベルにあるより小さいセル上で実施されることが出来る。従って、この種の接続は全体のアーキテクチャー内に適合出来る。事実、フィードバックが実施されない部分では、該セルのラッチは冗長である。フィードバックされないセルの終わりにある最後のラッチのみが必要である。

【 0 2 1 0 】

同様に、指定された遅延を提供するために多数のラッチがカスケードで使用され得る応用品が存在する。従って、該セルは部分的に、付随ラッチを有しないラム及び / 又は付随ラムを有しないラッチにより実現されてもよい。それにも拘わらずラッチをラムに直接付随させることが必要な場合、これは上記説明の様に該マトリックスを使用して行われ得る。それは使用するにより複雑であり、平均的エンジニアにとってエンジニアリング時間を費やすけれども、もしこの種のアーキテクチャーで働くエキスパートのグループが全ての利点用に新ライブラリーを作るため作業しつつあるならば、それは価値がある。

【 0 2 1 1 】

【簡単さ】

該応用品は小さな部分に分けられるので、それは応用品エンジニアを簡単な解決策へ導き、該解決策は実施するのに簡単でもある。

【 0 2 1 2 】

【人的資源の平行開発】

1 人のシステムエンジニアが該解決策のアウトライン (outline) を開発する時、彼又は彼女は該機能セルの中に作られねばならない機能を宣言出来る。次いで各セル実施品は独

10

20

30

40

50

立にコード化され得る。これは、もし市販までの時間が重要であるなら、各セル実施品のコードが異なるエンジニアにより開発され得ることを意味する。

【 0 2 1 3 】

【 流れ線図 】

図 3 1 は、本発明のデバイスを使用してハードウェア応用品の機能を規定する構造データ (construction data) を得るための第 1 の方法に付随する主要な動作ステップを示す流れ図である。 ” 構造データ ” は、セル及び / 又は構成要素間の接続を規定しその中に記憶するためのデータを含むトポロジを表すデータを意味する。見られる様に、トポロジは望ましい応用品を実施するために選択されるが、その後データは該選択されたトポロジにより規定されたセル及び / 又は構成要素内に記憶されるために定式化される。

10

【 0 2 1 4 】

図 3 2 は本発明のデバイスを使用して、望まれるハードウェア応用品の少なくとも部分的機能を規定する構造データを得るための代わりの取り組みを示す。各々がそれぞれの機能に関する構造データを含む予めコンフィギュアされたファイルを記憶するライブラリーがアクセスされ、該応用品の 1 つ以上の機能を実現するために 1 つ以上のファイルが該ライブラリーから選択される。オプションでは、該ライブラリーに含まれない機能を追加するか又はそこにある構造データを置き換えるか何れかを行う、前に得られた構造データが該ライブラリーにセーブされてもよい。かくして該デバイスと連携して該ライブラリー内に記憶された該構造データは該応用品の望ましい機能又は部分的機能を実現するデバイスアーキテクチャーを規定する。例えば、必要な追加的構造データと組み合わせられ、その構造データが該ライブラリーに記憶された 10 ビット加算器から、12 ビット加算器が作られてもよい。もし必要なら、最終 12 ビット加算器の構造データは、標準的特徴として次ぎに利用可能となるように、そこで該ライブラリーにセーブされてもよい。これは予め設計された構造データのライブラリーが再使用されることを可能にしてかくして各応用品用に繰り返して同じ構造データを創る必要を解消する。かくして上記例では、一旦創られた該 12 ビット加算器用構造データは記憶され、かくして 12 ビット加算器が必要な度に設計者がこの機能を再設計する必要を解消する。これは可成り設計時間を節約するのみならず、より重要なことは、エキスパート設計者が最適解決策を設計することそしてこれらの解決策を該設計者のコミュニティに利用可能にすることを可能にする。この設備は又、例えばインターネットを通してメンバーが設計を交換 (swapp) することを可能にする作業グループ (work-groups) に有用である。

20

30

【 0 2 1 5 】

図 3 3 は構造データを得るなおもう 1 つの方法を示すが、そこでは該少なくとも 1 つのセル及び / 又は該少なくとも 1 つの構成要素を実施するようそして該構造データを使用して予めコンフィギュアされたトポロジを実施するようコンピュータをプログラムすることによりハードウェア応用品のコンピュータシミュレーションが走る (is run) 。該構造データは、必要な様に、変更されてもよく、該コンピュータシミュレーションが満足されるまで該シミュレーションは繰り返される。次いで該ハードウェア応用品は該シミュレーション構造データから引き出されたデータを該デバイスにダウンロードすることにより実施される。上記注意の様に、該コンピュータは高級プログラム言語を使用してプログラムされてもよく、かくして従来はエキスパート設計エンジニアの領域であった熟練を今やプログラマーに利用可能にする。更に、該プログラムは該構造データを形成するためだけに役立ち、一旦該応用品の機能的実施が入手可能になり、該データがそのセルにダウンロードされると、該データが形成された仕方は最早関係なくなる。これは、該データを作りそれにダウンロードするため使用された該プログラムコードが最適化されたか、されなかったかに無関係に等しい効率で動作デバイスが機能することを意味する。

40

【 0 2 1 6 】

該シミュレーションを走らせるため該コンピュータをプログラムする時、該シミュレーション構造データは 2 つの方法で引き出されてもよい。 1 つの方策に依れば、該シミュレーション構造データの少なくとも 1 部分の瞬間的サンプルは該コンピュータにより実行中に

50

(on the fly) 連続的に計算される。しかしながら、異なる方策に依れば、該シミュレーション構造データの少なくとも 1 部分は予めコンフィギュアされそして記憶され、そして必要な時その瞬間的サンプルが該コンピュータによるフェッチ (fetched) される。このような方策は該シミュレーション構造データを実時間で計算する必要を避けるが、それは該データが予め決定されフェッチされ、かくして該構造データを引き出すために要するコンピュータ演算コード (computer opcodes) の数を著しく減らすからである。

【0217】

図 34 は本発明のデバイスを使用してハードウェア応用品を実施するための方法に付随する主要な動作ステップを示す流れ図である。その最も基本的な所で、フォーマットされたデータが得られ、該デバイスにダウンロードされる。該フォーマットされるデータは予め決められた構造データから引き出されて、次いで該デバイスへのダウンロード用にフォーマットされてもよい。該構造データ自身は図 31, 32 又は 33 を参照して上記で説明された方策のどれかを使用して得られてもよい。

10

【0218】

【シミュレーターとツール】

下記例はスタンドアロンの存在として各々同期化された複数のセルを取り扱う。これらの存在の集積化は当該技術で公知の様に達成される。

【0219】

【シミュレーター】

信号が該応用品に入力される普通の実施品では、該信号は操作されそして、結果として、信号が該応用品から出力される。シミュレーションでは、“仮想 (virtual)” 信号が、該実施品の機能をシミュレートするコンピュータプログラムに入力され、そして“仮想”出力信号を発生する。複雑な実施品をシミュレートすることは非常に複雑である。該難しい部分は該コンピュータ“時間 (time)”の分解能が固定されている時に該信号間の遅延をシミュレートすることである。該シミュレーターの分解能が良い程、該シミュレーションはゆっくりしている。

20

【0220】

全てのセルが同期化されているので、それが真に動作すると同じ仕方で該ハードウェアの精確な機能をシミュレートすることに問題はない。本発明のデバイスを使用して作られた応用品内の各セルはタスクを行う。該タスクは変換テーブル (ルックアップテーブル) として実施され得る。該テーブルデータにより表されるタスクは高級プログラム言語で書くことが出来る。種々のテーブル内に記憶された全てのデータの集まりはデバイスコードと呼ばれる。

30

【0221】

例：

今、擬似コード (pseud code) として“C”言語を使用した幾つかの例を下記するが、どんな他のプログラム言語も使用され得ることは理解される。

【0222】

どんなタスク実施品用の例も下記の様になるであろう：

```

//テーブル発生
.
.
#define CELL1_SIZE    256      // (何等かの利用可能なセルサイズ)
int iCell1(int iAddress);
.
.
int iTableOfCell1[CELL1_SIZE], iAddress;
.
.
for(iAddress=0; iAddress<CELL1_SIZE; iAddress++)
{
    iTableOfCell1[iAddress]=iCell1(iAddress);
}
//この点でiTableOfCell1[]はそのそれぞれの記憶要素内に記憶されたデータ
である該セル用のコードを保持する
.
.
    該セルのデータは該関数int iCell1(int iAddress)により別々に実施される。
例えば、図10を参照して上記で説明された該カウンター用該データは次の様に
、”C”で実施出来る：
//セルコード発生
#define MODULO  123
.
.
int iCell1(int iAddressForCell1)
{
    int iDataOfCell1;
    iDataOfCell1 = iAddressForCell1+1;

    if(iDataOfCell1==MODULO)
    {
        iDataOfCell1=0;
    }

    return iDataOfCell1;
}

```

10

20

30

該タスクがソフトウェアを使用して行われると、セル機能を実施する同じコードが該シミュレーター内でそして該デバイスへダウンロードされるべき該データの発生で使用出来る。該ハードウェア実施品の各サイクル内では唯1つの動作がセル内で行われるので、該動作は該コンピュータ上で”1対1(one to one)”でシミュレートされ得る。該動作をシミュレートするために、全ての該動作は、1つのハードウェアサイクルをシミュレートするループ内に含まれるべきである。

40

【0223】

例えば、該シミュレーター用コードは次の様に出来る：

```

//シミュレーション
.
.
#define CELL1_SIZE      256
int iDataOfCell1, iAddressForCell1;
BOOL bRunning;

iAddressForCell1=0;          //初期化
bRunning=TRUE;
while(bRunning)
{
    iDataOfCell1= iCell1(iAddressForCell1);
    //視点 (View point)。該データは準備完了しており、操作されスクリー
    ン上に示され得る。

    iAddressForCell1= iDataForCell1      //マトリックス動作
.
.
}
.
.

```

10

20

このコードと前のコードとの間の次の差が注意されるべきである：

- 1．該マトリックス動作が追加された；該 " アドレス " をインCREMENTする代わりに、該 " アドレス " は該新しい値を該データから得る。
- 2．該データはテーブル要素内には集められない。該データは1サイクル内に行われる単一動作を反映するのみである。
- 3．小数のセルは同じ仕方で、同じループ内で実施され得る。この例では、全てのセル用のマトリックス動作は該ループの終わりへ動かされるべきである。

【 0 2 2 4 】

30

従って、該過程を自動化すること、そして該シミュレーターに該デバイスコードにより求められるロード可能なデータを " 実行中に " 発生させることは全く容易である。該シミュレーションがこの仕方で作動する時、それは " コンパイルドモード (Compiled mode) " で動作すると云われる。

【 0 2 2 5 】

ユーザーにより変更されねばならない全てのコード、すなわち該セル用及び該マトリックス用のコード、は該シミュレーター用及び該ロードされるデータ用で同じであることを注意することは重要である。その結果として、該シミュレーターは該応用品の1対1のシミュレーションを行うことが出来るのみならず、更に、もし該シミュレーションが適当に作動すれば、それは追加的人的誤り無しに該ロードされるデータを創る自動的過程でもある。

40

【 0 2 2 6 】

該デバイスコードはアドレス - データ関係のテーブルである。該シミュレーターをより速く走らせるために、該テーブルにより表された該デバイスコードは予め計算され、該シミュレーターにより変換テーブルとして使用されることが出来て、かくして該シミュレーターが " 実行中に " で該テーブル値を計算する必要を解消し演算コードを節約する。該シミュレーションがこの仕方で作動する時、それは " リアルモード (Real Mode) " で動作すると云われる。この場合、該コードはこの様に作られる。

【 0 2 2 7 】


```

//シミュレーション
.
.
int iCell11(int iAddressForCell11);
.
.
#define CELL1_SIZE    256
int iDataOfCell11, iAddressForCell11;
int iTableOfCell11[CELL1_SIZE];

//該シミュレーションがスタートする前に該セルテーブルを創る
for(iAddressForCell11=0; iAddressForCell11<CELL1_SIZE;
iAddressForCell11++)
{
    iTableOfCell11[iAddressForCell11]= iCell11(iAddressForCell11);
}
.
.
//該シミュレーション

iAddressForCell11=0;                //初期化
while(bRunning)
{
    iDataOfCell11= iTableOfCell11[iAddressForCell11]; //関数の代わりのテー
ブル
    //視点 (view point)

    iAddressForCell11= iDataOfCell11                //マトリックス動作
.
.
}
.
.

```

10

20

30

該 2 つのシミュレーション方策は組合せて使用されてもよい。かくして、1 つのセルは " 実行中に " 該セルデータを計算することによりシミュレートされ得る、一方もう 1 つのセルは予め計算されたデータをフェッチすることが出来る。" コンパイルドモード " でデバッグすることはより容易であるが、それは " リアルモード " での方が速い。

【 0 2 2 8 】

例によると、該セルコード発生は次の様に現れる：

//セルサイズ宣言、ファイル名："CellSize.C":

```
.
.
#define CELL1_SIZE    256        // (何等かの利用可能なセルサイズ)
#define CELL2_SIZE    16384     // (何等かの利用可能なセルサイズ)
.
.
#define CELLn_SIZE    4096      // (何等かの利用可能なセルサイズ)
.
.
```

10

//iCell1用のセルコード発生、ファイル名："Cell1Generation.C":

```
.
.
int iCell1(int iAddressForCell1);
.
.
#include "Cellsize.C"
.
.
int iTableOfCell1[CELL1_SIZE], iAddressForCell1;
.
.
for(iAddressForCell1=0; iAddressForCell1<CELL1_SIZE;
iAddressForCell1++)
{
    iTableOfCell1[iAddressForCell1]= iCell1(iAddressForCell1);
}
.
.
.
```

20

30

//iCell2用セルコード発生、ファイル名："Cell2Generation.C":

```
.
.
int iCell2(int iAddressForCell2);
.
.
#include "Cellsize.C"
.
.
int iTableOfCell2[CELL2_SIZE], iAddressForCell2;
.
.
for(iAddressForCell2=0; iAddressForCell2<CELL2_SIZE;
iAddressForCell2++)
{
    iTableOfCell2[iAddressForCell2]= iCell2(iAddressForCell2);
}
.
.
```

40

50

このコードは各セル用を実施されるべきである一方、該関数コード (Cell1(int)、Cell2(int)...Celln(int)) は別々に実施される。

【 0 2 2 9 】

例により、該シミュレーターコード発生は次の様に現れる：

//シミュレーション：

.
.

//該シミュレーションがスタートする前に該セルテーブルを創る：

//セルコード発生：

#include "Cell1Generation.C"

10

#include "Cell2Generation.C"

.
.

#include "CellnGeneration.C"

.
.

int iDataOfCell1, iAddressForCell1;

int iDataOfCell2, iAddressForCell2;

.
.

int iDataOfCelln, iAddressForCelln;

20

.
.

BOOL bRunning=TRUE;

.
.

//該シミュレーション

iAddressForCell1=0; //初期化

iAddressForCell2=0; //初期化

.
.

iAddressForCelln=0; //初期化

30

while(bRunning)

```

{
    iDataOfCell11=iTableOfCell11[iAddressForCell11]; //リアルモード
    iDataOfCell12=iTableOfCell12[iAddressForCell12]; //リアルモード
    iDataOfCell13=iCell13(iAddressForCell12);      //コンパイルドモード
    .
    .
    iDataOfCell1n=iTableOfCell1n(iAddressForCell1n); //リアルモード

//視点
    iAddressForCell11=iDataForCell1k                //マトリックス動作
    (注意参照)
    iAddressForCell12=iDataForCell1m                //マトリックス動作
    (注意参照)
    .
    .
    iAddressForCell1n=iDataForCell1p                //マトリックス動作
    (注意参照)
    .
    .
}
.
.
注意：

```

10

20

該iAddressForCell1nは該セルの実際のアドレスの部分に過ぎなくてもよい。該マトリックスは該データの部分を該アドレスの部分に接続してもよい（ビット毎の接続）。これらのマトリックスラインが、このマトリックス動作を該”マトリックス制御メモリー”データに変換する、ホストにより使用される、種々の含まれたファイル、サブルーチン、又はクラスに対し分離され得ることは明らかである。

【 0 2 3 0 】

【シミュレーション速度】

30

下記議論は同じ機能を実施するデバイスの速度とシミュレーションの速度の間の比較に関する。該シミュレーターを走らせるコンピュータの速度要因を速度評価から除去するために、下記議論は該コンピュータは該デバイスと同じ速度で動作すると考える。

【 0 2 3 1 】

コンパイルドシミュレーターモード (Compiled simulator mode)

該応用品をシミュレートするために、指定された関数を実施するために専用化されたセルの各々は必要なアルゴリズムを実施するためにシミュレートされるべきである。下記では、”好ましいアーキテクチャーを選ぶための考慮”の標題の節で上記で説明されたアーキテクチャーに依り300セルが選択される。従って、該デバイスセルが平行に作動しつつありそして該シミュレーターがステップづつで作動しつつある時、該シミュレーターは該デバイスハードウェアより少なくとも300倍ゆっくりと作動しつつある。上記規定された”コンパイルドモード”では、該セルデータは該シミュレーターが走っている間に該指定された関数を実施することにより計算される。例え簡単な関数が典型的に少なくとも30の演算コードを使用して実施されても、かくして該ハードウェアに比して少なくとも9000の比率 (factor) で該シミュレーションを遅くさせる。ここで、該入力信号用の発生時間と該入力及び出力が見た信号の操作時間とを追加すると、該比率は更に高くなる。これは該シミュレーション速度を増加させるどんな技術も重要であることを意味する。

40

【 0 2 3 2 】

実際には、該コンピュータは該ハードウェアより遅くそして該セル関数内で求められるサブルーチンは非常に大きいので、該シミュレーションは遙かにもっとゆっくりと走る。

50

【 0 2 3 3 】

それにも拘わらず、シミュレートすべき限定された数の動作しかないので（上記で参照された好ましいアーキテクチャーで300セルを収容している）、該シミュレーターは今まで提案されたシミュレーターよりずっと速く走ると期待されている。

【 0 2 3 4 】

【リアルモード】

“リアルモード”では該セル内に記憶されるべき該データは該シミュレーションを走らせる前に計算され、該シミュレーション中該セルにより実施されるべき望ましいアルゴリズムを実行するために使用される。この様な手段により、該シミュレーションをもたらすに要する演算コードの数は大いに減少し、該シミュレーターは遙かに速く走る。この理由で、

10

1. 該セル関数用サブルーチンが大きい程、該速度の節約は大きい。
2. 該セルが大きい程、該節約は大きい。その理由はより大きいセルは多数のセルを組合せ、シミュレートされるべきセルの合計数を減らすことである。“リアルモード”では大きいセル又は小さいセルをシミュレートするために同じ時間を要する。
3. 少ない数のセルはコンパイルドモードでも該シミュレーター速度を増加させるが、大抵の場合より大きいセルはより小さいセルより多くのコードを使用するので、この利点は失われる。

【 0 2 3 5 】

20

結論では、該“リアルモード”は該デバイス上で走ると期待されるハイエンド応用品（high-end applications）で“コンパイルドモード”より数十から数百倍速く作動すると期待されてよい。この数は、該応用品でより大きいセルを創るため使用されたより小さいセルの平均数により掛け算されたセルの、タスク内の演算コードの平均数により計算される。

【 0 2 3 6 】

【エミュレーターモード】

エミュレーターモードでは該シミュレーションは“仮想信号（virtual signals）”発生により引き出されたクロック速度で走る。該応用品は実時間で走ることが出来るが、該シミュレーターにより発生される該“仮想信号”が最大速度をセットする。

30

【 0 2 3 7 】

【インサーキットエミュレーター】

（In Circuit Emulator）{アイシーイー（ICE）}モード

インサーキットエミュレーターモード内では、該シミュレーターは実時間で作動する。

【 0 2 3 8 】

【シミュレーター入、出力信号】

信号は、パターン発生を説明する例6での上記説明の様に該デバイス内セルにより発生される。等価な信号はコンピュータにより同等にうまく発生され操作され得て、該信号が何れかの場合に発生され又は使用される仕方の間に差は無い。それにも拘わらず、該コンピュータ化された信号は“仮想の”信号と引用され、かくして該コンピュータに接続されたディスプレイモニター上で見る事が出来る。

40

【 0 2 3 9 】

ユーザーが該“仮想”の信号を見て達成出来る幾つかの利点がある：

1. 該信号をそれらがあるが儘に示す（ロジックアナライザー流儀で）。
2. 該データのリストを示す（リストモード）。
3. 該信号を、見る前に変換する。例えば、パルスコード変調{ピーシーエム（）}では、“1つのロー（A Law）”{ピーシーエムの慣例（PCM convention）の1つ}信号は各第2ビットが反転される様なものである。ユーザーは該反転されたビットを見る前に反転出来るので、それは該結果をより理解し易い。ユーザーは該ピーシーエムコードを線形コードに変換してもよく、それによりそれをずっと理解し易くする。

50

4. ユーザーはバイト又はワード (words) をアナログ型に変換しそれらをスクリーン上
に示すことが出来る (スコープモード)。

【0240】

該出力信号は該応用品により発生される。それらは入力信号と同じ方法で操作され得る。

【0241】

図35は本発明のハードウェアデバイスをシミュレートするための方法に付随する主要な
動作ステップを抄録する流れ図である。ステップ式に走らせるよう適合されたエミュレ
ーションモジュールは該デバイス及びアイ/オーインターフェースを含んで作られる。望ま
れるハードウェア応用品を創るためにフォーマットされたデータが該エミュレーションモ
ジュール内の該デバイスにダウンロードされ、該アイ/オーインターフェースはコンピュ
ータに接続されるが、該コンピュータは制御デバイスを構成しそして又該エミュレーシ
ョンに参画する。入力サンプルが該コンピュータにより発生され該アイ/オーインターフ
ェースを経由して該エミュレーションモジュールにロードされる。出力サンプルは該アイ/
オーインターフェースを経由して該エミュレーションモジュールから解析用に集められ
る。1つの方策に依れば、該エミュレーションモジュールはステップ式に走らせるようセッ
トアップされるのでそれは、該前の出力サンプルが解析され、正しいと見出された後、該
コンピュータによりオーソライズされる、次の入力サンプルを待つ。もう1つの方策に依
ると、該エミュレーションモジュール内の該デバイスは、図面の図36を参照して下記で
説明される様に、インサーキットエミュレーション用のそれと同一の仕方で該出力サン
プルを解析するために幾つかの未使用セル (ことによるとこの目的に専用化されたもう1つ
のデバイス内の) を使用する。全過程は必要により、サンプル毎に繰り返される。代わり
のコンフィギュレーションでは、ロジックアナライザーが該コンピュータに取って代わり
、信号発生器が該入力サンプルを発生出来る。

【0242】

次の入力サンプルが印加される前に、該コンピュータ又は他のロジックアナライザーは現
在の出力サンプルを解析し、そして欠陥の場合に更に進んだ動作をことによるとアボ
ート (abort) するために時間を取るために該エミュレーションをステップ式に走らせるこ
とが必要である。該望ましいステップ式に走らせることは、該応用品にクロック信号を供給
するようクロックイネーブル信号を外部的に供給することにより、そして該応用品による
無効化まで該クロック信号の供給を避けるようクロックイネーブル信号を内部的にデイス
エーブルにすることにより達成出来る。

【0243】

【エミュレーター】

1つ以上の汎用ハードウェアデバイス (含む複数) を有する汎用エミュレーターピーシー
カード (general-purpose emulator PC card) が実施される。該カードは該デバイス内に
ロードされる。該コンピュータにより発生される " 仮想 " 入力信号は該ピーシーバス (PC
Bus) { 例えば、ピーシーアイ (PCI) バス } を経由して該カードの入力に転送される。
該デバイス又はもう1つのデバイスの自由な部分が、ロジックアナライザー用と同じロジ
ックだが、ユーザーが欲する様に柔軟な、望まれるロジック状態を取り込むためにプロ
グラムされる。該出力は戻って該ピーシーバス経由でサンプルされ該ユーザースクリーン上
で見られる。該スクリーン上で該信号を見るために、必要ならば、該エミュレーションカ
ードのクロックは操作され、スローダウンされ、停止されるか又は1ステップ化 (single
stepped) され得る。該クロック信号は最大の速度又は柔軟性を持つために該ソフトウ
ェアにより制御される該ピーシーから該デバイスの外部クロック信号内に注入され得る。

【0244】

もし数時間以上続く長い持続時間実信号を入力とする必要があれば、インサーキットエ
ミュレーターが使用されるべきである。該同じ汎用エミュレーターピーシーカードが使用さ
れるが、該ピーシーカード用ポートはユーザースクリーン上で実施されるべきである。このポ
ートは該アイ/オー信号を該エミュレーターカード内へ伝送する。該エミュレーターは該
入力信号を記憶し、該出力及び入力信号上でロジック操作を行う。一旦このロジック操作

の結果が取り込まれると、該過程は停止し（該クロックをデイスエーブルにすることにより）そして該応用品をデバッグするためにユーザーは該記憶された信号を該シミュレーター／エミュレーター内に注入（inject）する。

【 0 2 4 5 】

図 3 6 は該デバイスの実時間動作中に該エミュレーションモジュール中で該デバイスをテストするためのインサーキットエミュレーション方法に付随する主要な動作ステップを抄録する流れ図である。テスト中の該モジュールのアイノオーインターフェースの少なくとも 1 つは制御ユニットに接続され、そして該デバイスは該インターフェースの少なくとも 1 つを経由して該デバイスの特定点のサンプルを外へ移植するようコンフィギュアされる。該サンプルは該デバイスのサンプルの履歴を実時間で得るよう該制御ユニット内に採取され、そして該制御ユニットは最も最近のサンプルの少なくとも部分集合を解析するため使用される。もし必要なら、テスト中のデバイスの動作は次の何れか又は全部を可能にするように停止される：

- （ i ） 該デバイスのサンプルの履歴の検査、
- （ i i ） 該停止されたデバイスの瞬間的な現在の状態の検査、
- （ i i i ） 該停止されたデバイスへの異なる状態のダウンロード、
- （ i v ） 該デバイスの実時間動作の継続、そして
- （ v ） 該デバイスの実時間動作の再スタート。

【 0 2 4 6 】

【 単一開発環境 】

同じ開発環境は該過程の始めから該製造段階まで使用される。該開発環境は高価なテスト機器に取って代わり得るが、それは標題 “ シミュレーション ” の下で説明された様に該ピーシーソフトウェア環境（ PC software environment ）で “ 仮想 ” 信号を操作することは非常に容易だからである。それはシミュレーター、エミュレーター、そしてインサーキットエミュレーターとして使用出来て、そしてそれは製造部門で該カードをチェックするプログラムとして使用出来る。

【 0 2 4 7 】

【 顧客のコードの品質 】

応用品エンジニアは解決策、すなわち：各セル内に何がプログラムされるべきか、そして該セル間はどうな接続であるべきかを規定する。次いで、該セル用コードが書き込まれる。実時間に関しては、該コードの品質は、該デバイス内にロードされる “ コード ” の品質に対し重要でない。例えば、図 1 0 で示す該カウンター実施品が次のコードを使用すると仮定する：

```
int iCell1(int iAddress)
{
    int iData;
    iData=iAddress+1;
    if(iData)=MODULO)
    {
        iData=0;
    }
    return iData;
}
```

そして次の様に変えられる：

10

20

30

40

```

int iCell1(int iAddress)
{
    int iData;
    if(iData !=iAddress+1)          //??非効率的!!
    {
        iData=iAddress++;
    }
    if(iData)%MODULO
    {
        if(iData !=0)              //??非効率的!!
        {
            iData=0;
        }
    }
    return iData;
}

```

10

該第2の実施例のコードは遙かにゆっくりしているが、同一データは該デバイスにロードされる。かくして、最終顧客にリリースされる該製品は初心者プログラマーにより影響されない。

【0248】

20

上記特徴は、ソフトウェアエンジニアとハードウェアエンジニアと両者が該実施を行えるので、設計者の可用性 (availability) を増すために役立つ。システムエンジニア - 該解決策を開発する人 - 該シミュレーションを使用することにより容易にチェックされることが出来る。

【0249】

より貧弱なコードを有する "コンパイルモード" では該シミュレーターはよりゆっくり作動するが、該 "リアルモード" を利用するデバイスの部分では差が無いので組み合わせられたモードでは該差は気付かれない (該テーブルを作る時間は無視され得るので)。

【0250】

図37は所望機能を有する特定用途向けハードウェアデバイス (application-specific hardware device) を設計する方法に付随する主要な動作ステップを示す流れ図である。本発明のデバイスに連携して該所望機能を実現するデバイスアーキテクチャーを形成するフォーマットされたデータが得られる。該デバイスアーキテクチャーのプログラム可能なマトリックス又は複数マトリックスは該プログラム可能なマトリックス又は複数マトリックスの接続を実現する固定接続部により置換され、そして該デバイスアーキテクチャーのセルの少なくとも幾つか内の記憶要素はその中に記憶されたデータを実現するためのそれぞれの固定ドライブレベルにより置換される。この様な手段により、該アーキテクチャーは該特定用途向けハードウェアデバイスの直接的実施に好適にされる。

30

【0251】

この様な方策は特定用途向け集積回路 { エイシックス (ASICs) } の促成の開発を可能にし、該集積回路は多量生産されるよう意図されたハードウェア応用品用のチップ面積を減じるためにはプログラム可能なデバイスより好ましい。他方、本発明のデバイスは該エイシックスを設計するそして所要のフォーマットされたデータを得るための作用体 (agent) としてだけ使用され、そしてそれは次いで変型される。エイシックスはそれ自身汎用ハードウェアデバイスでなくそしてプログラム可能でない。

40

【0252】

かくして、エイシックス変換では、応用品ラムとして使用される各ラム内の各ビットは "1" 又は "0" ドライバーで置き換えられる。これは該ラムのサイズを減らし、該ラムをプログラム不可能なテーブルへ変換する。記憶用に使われる何れのラムも手を付けられず残される。より多くの節約が達成出来るが、該特別の節約は "エイシックスまでの時間 (time

50

to ASIC) ”を延ばし、より高価になるので、上記の様にだけすることが通常勤められる。上記は自動的ツールとして実施されてもよい。

【0253】

下記は応用品の機能の1部を実施するためにエイシックとして使用することにより達成され得る特別の節約の例を示す。8つのフリップフロップと幾つかのロジックを用い、従来の方法を使用して8ビットアップ・ダウンカウンタが実施され得る。本発明の同じ8ビットアップ・ダウンカウンタは9ビットアドレスで実施されそして8ビットデータセルは4096ビットのラムを要する。本発明は、実際複雑な応用品では、メモリーを非常な無駄にすると表面的にはこの例から見える事実にも拘わらず、本発明により達成される実施品全体は”ダイ節約品(die saver)”であり、カウンタ変換(counter conversion)は全体のエイシックダイサイズの約0.3%を節約する。例として、チャンネル毎の適合テーブル(adaptive tables)を保持する32チャンネルの各1つ用にTX及びRXアルゴリズムを実施する完全なイー1通信ライン(full E1 communication line)用に”ハフマンコード(Huffman Code)”が実施されることが評価された。この応用品は0.25ミクロン技術で約250mm²のダイ上に実施される。もう1つの例は完全なデバイスの3より少ないパーセントを消費する8ビット、8タップのエフアイアールフィルタ(FIR filter)である。本発明のエイシック変換方法は、上記カウンタの様な、該エイシックの幾つかの基本的機能が、ライブラリー形式での従来方法を使用して利用可能にされるよう拡張されてもよいことを、更に注意すべきである。この様な場合、予め変換された機能を使用することにより、ユーザーは該エイシックの回復不可能な消費{エヌアールイー(NRE)}時間を浪費することなくより良い変換を実施出来る。

【0254】

各々が種々の機能を有する1連の種々の回路基板上にエイシックが搭載されるべき時、該汎用ハードウェアデバイスの柔軟性をエイシックの利点と組み合わせることが望ましい。この場合、該デバイスアーキテクチャーの該プログラム可能なマトリックス又は複数マトリックスの1部のみが置換される接続部の接続を実施する固定接続により置き換えられ、そして該デバイスアーキテクチャーのセルの幾つかだけの記憶要素がその中に記憶されるデータを実現するためにそれぞれの固定ドライブレベルにより置換される。

【0255】

この様な技術は重要な部分(critical parts)の設計用又は実施品間の差の収容用に望ましい。例えば、種々の国の間で小さな差異のある電話通信システム信号動作(telephony signaling)が実施される場合である。

【0256】

【欠陥チップの利用】

チップの値段はダイのサイズに関係し、約250mm²は極めて大きいと考えられるので、下記の方法は該チップ価格を下げる事が出来る。該デバイスはセルベースであり、それは何等かの与えられたタスクを実施するためにセルが選ばれる応用品には関係しないので、デバイステストは行われるであろう。もしセルが欠陥ありと見出されるなら該セルは使用されないように欠陥ありとマークされる。

【0257】

次いで該マトリックスはテストされる。もしリンクが欠陥ありと見出されるなら、そのリンクは使用されるべきでない。時には欠陥接続通路はセルの消失(loss of a cell)を引き起こし、そして時には該リンクは放棄(spared)され得ることは注意されるべきである。

【0258】

2, 3百のセルしかなく、そして該マトリックスのスイッチは”マトリックス制御メモリー順”に記憶されるので、該欠陥部分をマーク付けするために小片のコードを作ることは全く簡単である。このコードはマトリックス制御メモリーの欠陥性アドレス及びデータと該欠陥性セル数から成る。

【0259】

もしタイミングが欠陥性であるなら、デバイス全体が欠陥性になる。該アイ/オーは同じ仕方でチェックされ得るが、もし該アイ/オーが欠陥性であれば該デバイスの使用は勧められない。

【 0 2 6 0 】

【より高い信頼性を達成する利用法】

そのアイデアはそのデバイス内にスペアのセル又は 2 , 3 のスペアのセルを持ち、それらを欠陥性セルの代わりに使用することである。マトリックスは既に " スペアリンク (spare links) " を有する。例えば :

もしデバイスに接続されたホストがあれば、上記方法は応用品で使用され得る :

- 1 . 1 つ又は幾つかのスペアセルが余される。
- 2 . 組み込まれたテストプログラム (Built-In Test program) がそれらの自由なセルをポートを経由してチェックする (ラムチェックと同様に) 。
- 3 . そのセルは該自由なセルと同じ接続を取ることが出来るもう 1 つのセルのタスクでロードされる。
- 4 . 該セルの入力は該もう 1 つのセルの同じ入力ヘルート付けする。
- 5 . 該セルの出力は該もう 1 つのセルの同じ出力ヘルート付けされる。今は両セルが並列で作動しているので衝突は起こらない (もし誤りがなければ) 。
- 6 . 該もう 1 つのセルはスペアセルであるようルート付けされる。

【 0 2 6 1 】

マトリックス誤りはセル誤りの " 様に見える (look like) " 。それをテストするために、もう 1 つのリンク通路が使用されてもよい。もし他のこの様なリンクが無ければ、該セルは " 欠陥性 " とマーク付けされる。タイムシェアリングコンフィギュレーション (time-sharing configuration) での " セル - ラム (Cell-RAM) " は全サイクルの後だけスイッチオフされ得る。

【 0 2 6 2 】

図 3 8 は、本発明のデバイスを使用してハードウェア応用品を実施する時の上記方法に付随する主要な動作ステップを抄録する流れ図である。該デバイスにフォーマットされたデータをダウンロードする前に、該フォーマットされたデータは、該フォーマットされたデータと何等かの欠陥性のセル及び / 又は接続の欠陥リストとを含む、デバイス特定の、拡張され、フォーマットされたデータ (device-specific extended formatted data) を創るために使用される。テストをする前は、該欠陥リストは空である。かくして、空の欠陥リストは欠陥性セル又は接続の無いことを示すが、簡単には、該デバイスが未だテストされてなかったことを意味する。もし必要なら、該デバイス特定の、拡張され、フォーマットされたデータは他のデバイス特定の情報を含んでもよい。該デバイス内のセルと接続はテストされ、従って欠陥リストは更新され、そして欠陥性と分かった何等かのテストされたセル又は接続を使用することを避けるよう、もし必要なら、該デバイス特定の、拡張されフォーマットされたデータ内の該フォーマットされたデータは変更される。

【 0 2 6 3 】

図 3 9 は本発明のデバイスの実時間自動欠陥検出及び修正方法 (method for real time automatic fault detection and correction) に付随する主要な動作ステップを示す流れ図である。図 3 8 を参照して上記で説明された様に、デバイス特定の、拡張され、フォーマットされたデータは、該欠陥リスト内にない未使用セルを位置検出するため使用される。もしこの様な未使用セルが位置検出されれば、該未使用セルとその接続はテストされそして、該未使用セル及び / 又はその接続の何等かが欠陥性であるなら、該デバイス特定の、拡張され、フォーマットされたデータは更新されそして欠陥性でない更に進んだ未使用セルが位置検出 (located) される。テストされ動作すると分かった自由なセルが位置検出された時、該デバイス特定の、拡張され、フォーマットされたデータは、該未使用セルと同じ接続を取ることが出来る、使用済みセルをテスト用に選択するために使用される。該デバイス特定の、拡張され、フォーマットされたデータは該選択されたセルとその接続とを該動作する自由なセル上へ複製するため使用される。該選択されたセルは今

度は遮断され、従って該デバイス特定の、拡張され、フォーマットされたデータは更新される。この様な手段により、テストされるべき使用済みセルの規定は動作するものと確立された自由なセルにコピーされる。結果として、該自由なセルは、今は遮断された元の使用済みセルとして今や動作する。該遮断の前に、両セルは並列に作動するがこれが起こることを許される時間は無視可能である。もし該元のセル又はその接続の何れかが欠陥性であるなら、それらの機能を該自由なセルにコピーする動作は、該自由なセルが動作すると知られているので、該欠陥を修正する。該全サイクルは必要な様に繰り返される。

【 0 2 6 4 】

図 4 0 は、利用可能な予め設計されたカードが無い場合又は新インターフェースが必要な時に、本発明のハードウェアデバイスを帯びる回路カードの初めての生産用の過程を抄録する流れ図である。一旦インターフェースが準備完了すると、最終カードは多くの異なる応用品用に使用出来るので、応用品の規定（それは該カード開発過程の 1 部でもある）は該設計には稀にしか影響しない。

10

【 0 2 6 5 】

かくして応用品の規定 3 0 0 は該応用品を一般的用語で規定し、それにシミュレーション 3 0 1 が続く。該シミュレーション過程は図 3 3 を参照して上記で詳細に説明された様に作られる応用品シミュレーションを要求する。該シミュレーションの目的は該アイデアが実行可能であることを確認し該デバイスを走らせ続けるためのコードを発生させることである。例えば、もし該デバイスが通信リンク上で音声を圧縮するよう設計されねばならないなら、“C++”の様な高級言語を使用するか又は上記説明の様にこの目的で供給されるツールを使用して該圧縮アルゴリズムが書かれる。もし望むなら、これらのツールは上記で参照した“マットラブ(Matlab)”の様な既知のツールと組み合わせて使用されることが出来る。一旦該シミュレーションが充たされると、該デバイス用コードは準備完了である。該シミュレーションは“1 対 1”シミュレーションであり、それは該関数をシミュレートする能力の他に、該コードが該デバイス上で走る精確な仕方をそれがシミュレートすることを意味している。従って、もし該応用品が該シミュレーター上で作動するなら、該応用品は該基板上で作動するであろうし、もし該応用品が該シミュレーター上で作動しないなら、それは該基板上で作動しないであろう。

20

【 0 2 6 6 】

該“1 対 1”のシミュレーション能力は該シミュレーションを行う該コンピュータ内に該デバイスの精確な仮想コピーを創生することを可能にする該デバイスアーキテクチャーに由来する(derives)。従って、該デバイスのコピーの入力を該コンピュータに入れる“人工の(sunthetic)”又は記録された“仮想の(virtual)”信号は該デバイスにより作られるであろう精確な“仮想の”出力を発生することになる。これらの“仮想”信号は操作され、テストされることが可能である。

30

【 0 2 6 7 】

該シミュレーターは次のモードで動作することが出来る：

- ・コンパイルモード、すなわちそこでは該セルにより実施されるべき望まれるアルゴリズムを実行するために高級言語で書かれたサブルーチンを利用して該シミュレーションが実行される。

40

【 0 2 6 8 】

- ・リアルモード、すなわちそこでは該セル内に記憶されるべきデータは該シミュレーションを走らせる前に計算され、そして該シミュレーション中、該セルにより実施されるべき望まれるアルゴリズムを実行するよう使われる。この様な手段により、該シミュレーションをもたらすに必要な演算コード数は大いに減らされ該シミュレーターは遙かに速く走る。

【 0 2 6 9 】

- ・上記の組合せ。テスト中である該応用品の部分は高級言語のコンパイルモードで走る、一方該残りはリアルモードにある。このモードはデバッグ用に最大速度を達成するよう意図されている。

50

【 0 2 7 0 】

・エミュレーションモード：該シミュレーターはハードウェアデバイスにより加速される。該エミュレーションモードでは、該シミュレーターは該デバイス内のセルを使用し、ロジックアナライザーに似た能力を有する。換言すれば、該エミュレーションはどのロジックパターン又はシーケンス上でも停止出来て、該デバイス内の各接続部の結果をスクリーン上に示すことが出来る。それは単一ステップでも同様に作動出来る。これはデバッガーを使用してソフトウェアプログラム中を追跡する能力に似たハードウェアである。

【 0 2 7 1 】

・"インサーキットエミュレーション(In Circuit Emulation)" {アイシーイー(ICE)}モードを使用して、該エミュレーションは入力信号をサンプルする能力と組み合わせられる。該カードはそのインサーキットデバイス(in-circuit device)の処理能力を使用して実時間で作動する。

10

【 0 2 7 2 】

該シミュレーターは"1対1"シミュレーションを実施するので、大抵のテストとデバッグとは該シミュレーション段階で行われる。テストとデバッグ302は数時間又は数日続く真の長持続時間信号(real long duration signals)をテストするよう要求され、従って該アイシーイーモードが使われることを必要とする。これは上記で説明された様に、該シミュレーション速度に影響を与える。

【 0 2 7 3 】

デバッグは次の様に実行されるが、すなわち、該デバイスの1部は、その原因が解析され、修正されるべき誤りを典型的に示す、何等かのロジック状態を取り込むようプログラムされる。同時に、該デバイスのもう1つの部分は入力連続サンプルを環状バッファ(circular buffer)内へ取り込む。誤りが検出されると、該エミュレーションは停止し、該誤りを引き起こした該記録された入力信号が、該問題を解くために該シミュレーター内に注入され得る。少なくとも1つは該環状バッファのカウンター用にそしてもう1つは該バッファ用に、該入力信号を取り込むために1つより多いセルが必要になる。

20

【 0 2 7 4 】

もう1つのデバッグ方法に依ると、該コンピュータ上の該仮想デバイスが該デバイス内にロードされたどんな望まれる状態をも発生させ、該テストはこの点から進むようエミュレーション又はアイシーイーモード内で続けられる。

30

【 0 2 7 5 】

一旦テストとデバッグとが完了すると、アールアンドデーから生産へ(R&D to Production)の段階303が始まる。予め設計され、多目的の、デバイスベースのカードを使用する応用品発生過程の場合、該アールアンドデーから生産の過程は、該シミュレーター301により発生される該デバイスコードをアールアンドデーから生産部門へ発送するより多くは要求しない。

【 0 2 7 6 】

カード生産304が続く。予め設計され、多目的の、デバイスベースのカードを使用した応用品発生過程の場合、カード生産は該応用品が仕上がる前にスタート出来る独立過程である。これは図41で略図的に示されるが、それは図41で該カード開発過程305が無いことを除けば他では図40と同一である。カード生産304に続いて、検証過程306が行われる。カード検証は電子機器を用いて該カードをテストする過程である。該新しいデバイスアーキテクチャーは応用コード無しに該応用品がその上で走り得ることを保証するよう該カードがテストされ得る様なものである。従って、該アールアンドデーグループの作動の終了と該生産部門での該カード検証過程の開始/終了の間にリンクは無い。

40

【 0 2 7 7 】

クライアントへのカード発送307は該応用品が準備完了する前に行われ得る。該応用品が準備完了する、より後の段階で、適合するようフォーマットされたデータの形式のデバイスコードが該デバイスへのダウンロード用に該顧客へ送られることが可能である。

【 0 2 7 8 】

50

該發送段階の後で見出された何等かの誤りは上記説明のシミュレーション及びデバッグツールを使用して固定され得て、新しいコードがその顧客に發送されることが可能である。増強（Enhancements）又はバグ（Bugs）段階 308 は該開発過程の 1 部として図 40 で示され、開発者が、發送後に変更が行われることを斟酌して、どの段階で該製品が發送されるべきかを決定出来ることを示している。

【0279】

本発明の開発過程は、現在使われているより普通な " 滝型（waterfall）" 過程と相対する、いわゆる " 螺旋型（spiral）" 開発過程モデルである。該螺旋型モデルとその利点は例えばウェブサイト <http://www.cstp.umkc.edu/personal/cjweber/spiral/html> に説明されている。

10

【0280】

図 40 に示される該 " カード開発 " 段階 305 は図 1 を参照して上記で説明されたそれと同様である。唯の差は実施されるべき応用が無く、それでその実施はより容易であり、よい短い時間しか取らないことである。従って唯該インターフェース、そして該デバイス（複数を含む）のピンアウト（pin-out）が設計される必要があるだけである。大抵の場合、該カード創生過程は次の理由のために市販までの時間に影響しない筈である：

- ・インターフェース、デバイス数（通常は 1 つ）そして該ピンアウトが設計される必要がある（非標準型であるなら）のみなので、実施は簡単である。換言すれば、
- ・一旦カードが準備完了し、インターフェースが固定されると、既製品（ready-made）のカード（又は既製品のカード設計）がその新応用品に使用出来る。

20

【0281】

それにも拘わらず、新応用品が非標準のユニークなインターフェースを用いて作られることになるなら、そのカード製造過程はその実施過程より長くなり、市販までの時間（time-to-market）の節約は減じられることが予想されるべきである。従って、既知インターフェース用の、汎用で、多目的で、在庫式で、既製品型ののカードが種々の応用品に準備完了であるよう用意する努力が行われるべきである。

【0282】

本発明はかくして莫大な数の種々の応用品の実施を受け入れそして容易な拡張と混じり合ったアーキテクチャーを許容する、高度に柔軟で再プログラム可能なデバイスアーキテクチャーを提供する。設計の信頼性と簡便性は " 1 対 1 " シミュレーションにより保証されるが、そこでは該応用品を実施するために該デバイスにダウンロードされると同一なデータが応用品のコンピュータシミュレーション中使用される。シミュレーションとエミュレーションとは良く知られた高級プログラム言語を使う普通のパーソナルコンピュータを使用して実行され、高価な、専用ワークステーションのニーズを解消する。ユニークなデバイスアーキテクチャーの結果として、該ハードウェア設計と応用は説明された様に並列動作に部分分割され得て、モジュール型作業ブルーprintsの方策にかなっている。この様な柔軟性は予め製作される構造とフォーマットされるデータのライブラリーへのアクセスの能力により高められる。

30

【0283】

種々の実施例が説明されたが、本発明の特に有利な特徴はその柔軟性とその使用法にあることは当業者には明らかであろう。かくして、該例は単に図解的であり、本発明の範囲は請求項の範囲内に入る全ての変型を含むよう意図されている。

40

【0284】

同様に、本発明は特にデジタル電子回路の設計に関連して説明されたが、本発明が、本発明の基本的部品、該セル及びプログラム可能なマトリックスが、適当な技術を使用して実施される、生物学的ハードウェア、空気圧的ハードウェア（pneumatic hardware）、機械的ハードウェアを創るためにも応用を見出してよいことは当業者には評価されるであろう。従って、請求項では、用語 " ハードウェア " は、電子的ハードウェア、生物学的ハードウェア、空気圧的ハードウェア、機械的ハードウェアそしてデータ記憶能力を有するセルが実施され得てそしてプログラム可能なマトリックスが実施され得るどんな他のハードウ

50

エアも含むよう使用される。

【0285】

又、本発明のシステムは適当にプログラムされたコンピュータであってもよいことは理解されよう。同様に、本発明は本発明の方法を実行するためにコンピュータにより読み出し可能なコンピュータプログラムを考慮する (contemplates)。本発明は更に、本発明の方法を実行するために機械により実行可能な命令のプログラムを明白に具体化する機械読み出し可能なメモリーを考慮する。

【0286】

続く方法請求項では、請求項過程を呼称するため使用されるアルファベット文字は利便性のためだけに提供され、過程を行う何等特定の順序を意味しない。

10

【0287】

本発明を理解し、それが実際に如何に実行されるかを見るために、ここでは好ましい実施例が、限定しない単なる例により、付属する図面を参照して説明される。

【図面の簡単な説明】

【図1】 電子的カード上でのアイシーチップのハードウェア実施品用の従来の過程を示す流れ図である。

【図2】 本発明によるデバイスアーキテクチャーを略図的に示す。

【図3及び4】 図2に示すデバイス内で使用するためのセルの代わりのコンフィギュレーションを略図的に示す。

【図5】 補助回路を有する、図4に示すセルの詳細を略図的に示す。

20

【図6】 より大きいアドレスバスを有するよう2つのセルから形成されたセルを創るに要する接続を略図的に示す。

【図7】 Dセル出力をAセル入力に接続するマトリックスを略図的に示す。

【図8】 図8の回路で使用されてもよい飽和マトリックスを略図的に示す。

【図9】 図7の回路で使用されてもよい不飽和マトリックスを略図的に示す。

【図10】 図3で示す1つのセルを使用するカウンターを略図的に示す。

【図11】 図3で示す1つのセルを使用するアップダウンカウンターを略図的に示す。

【図12】 図3で示す3つの相互接続されたセルを使用するシフトレジスターを略図的に示す。

【図13】 図12で示す該シフトレジスター内で使用される1つのセルのシフトレジスターの可能なトポロジーを略図的に示す。

30

【図14】 図5に示す種類の2つのセルが如何にトライステートに接続されるかを略図的に示す。

【図15】 図5に示す種類の2つのセルを使用する第1の実施例のラムサーバーを略図的に示す。

【図16】 図15に示すラムサーバー用の起こり得るタイミング線図を示す。

【図17】 図5に示す種類の2つのセルを使用する第2の実施例へのラムサーバーを略図的に示す。

【図18】 図17に示すラムサーバー用の起こり得るタイミング線図を示す。

【図19】 時分割応用モードで動作するシフトレジスターを略図的に示す。

40

【図20】 図19で示すシフトレジスター用のタイミング動作を示すタイミング線図である。

【図21】 時分割応用環境で動作するラムサーバー組合せを略図的に示す。

【図22】 より大きいデータバスを有するよう2つのセルから形成されたセルを創るに要する接続を略図的に示す。

【図23】 セルにデータをロードする間要する接続を略図的に示す。

【図24】 8ビットコマンドを行うようコンフィギュアされたデバイスを略図的に示す。

【図25】 ラッチを有しないスタンドアロンのラムセルを示す。

【図26】 図25に示すセルを使用する非最適化加算器を略図的に示す。

50

【図 27】 図 25 に示すセルを使用した改良された加算器を略図的に示す。

【図 28】 図 25 に示す該ラムから独立に使用されてもよいラッチを略図的に示す。

【図 29】 クロックイネーブル信号が有効クロック速度を調整するため使用されるデバイスを略図的に示す。

【図 30】 プログラム可能なマトリックスの種々の状態間を高速にスイッチすることを可能にする多数エムシーエムアーキテクチャーを略図的に示す。

【図 31】 本発明のデバイスを実施するための構造データを引き出す第 1 の方法により使用される主要動作過程を示す流れ図である。

【図 32】 構造データを抽出し記憶するようライブラリーを使用するための第 2 の方法により使用される主要な動作過程を示す流れ図である。

【図 33】 本発明のデバイスを実施するために構造データを引き出す第 3 の方法により使用される主要な動作過程を示す流れ図である。

【図 34】 本発明のデバイスを実施する方法により使用される主要な動作過程を示す流れ図である。

【図 35】 本発明のデバイスを使用する応用品をシミュレートする方法により使用される主要な動作過程を示す流れ図である。

【図 36】 本発明のデバイスを使用する応用品をエミュレートする方法により使用される主要な動作過程を示す流れ図である。

【図 37】 エイシック設計を容易にするため本発明のデバイスを使用する方法により使用される主要な動作過程を示す流れ図である。

【図 38】 該デバイスを使用する応用品を実現する間該デバイス内の欠陥セルの使用を避ける方法に使用される主要な動作過程を示す流れ図である。

【図 39】 該デバイスを使用する応用品の実時間動作の間該デバイス内の欠陥セルの欠陥修正のための方法により使用される主要な動作過程を示す流れ図である。

【図 40 及び 41】 電子的カード上でのアイシーチップのハードウェアでの実現用の本発明の過程を示す流れ図である。

10

20

【図 1】

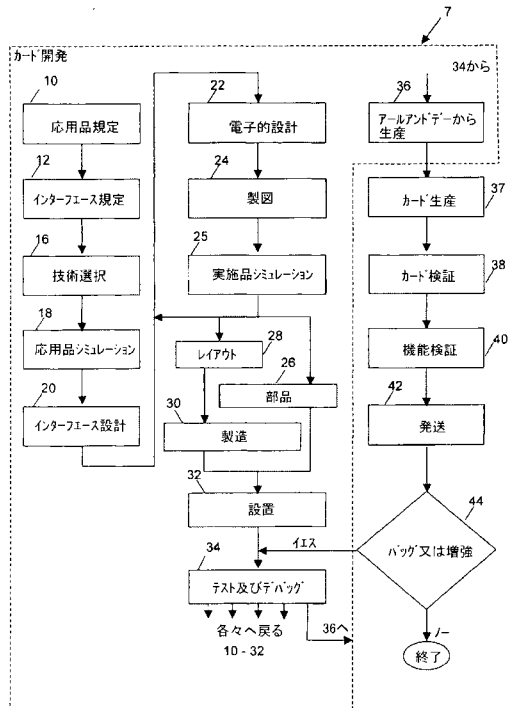


FIG. 1

【図 2】

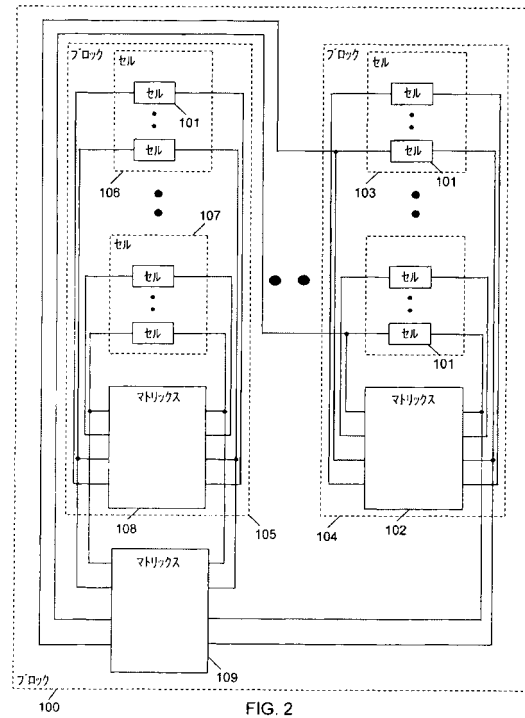


FIG. 2

【図 3】

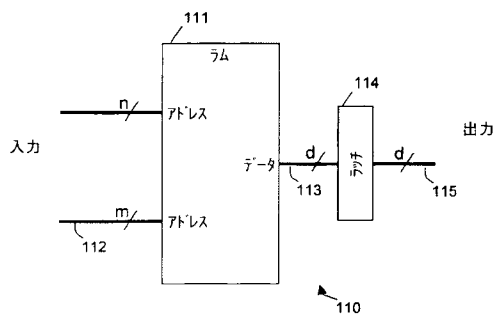


FIG. 3

【図 4】

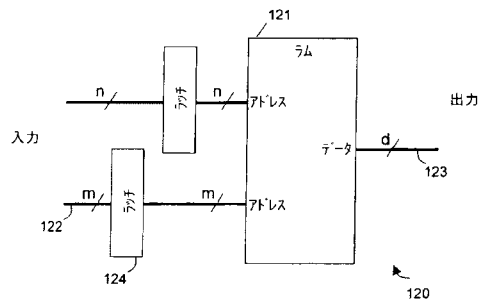


FIG. 4

【図 5】

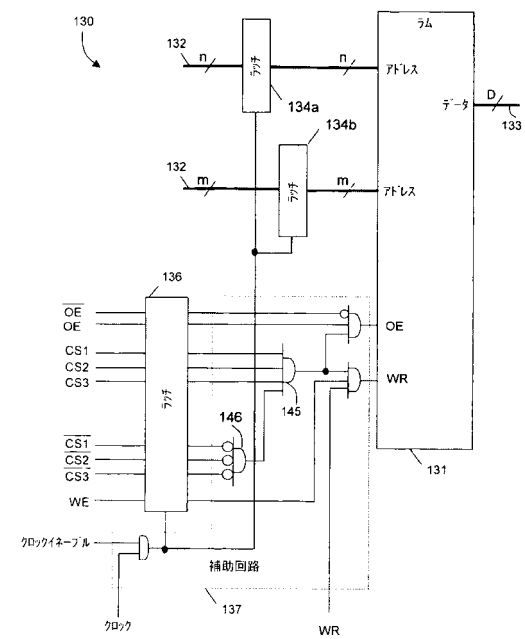
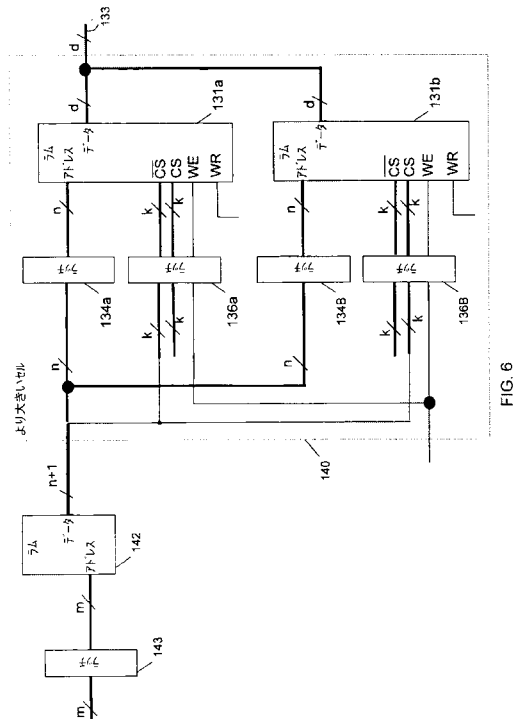
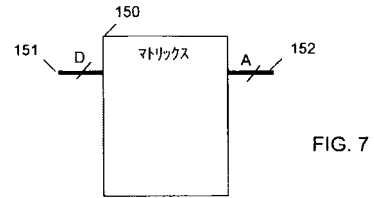


FIG. 5

【図 6】



【図 7】



【図 8】

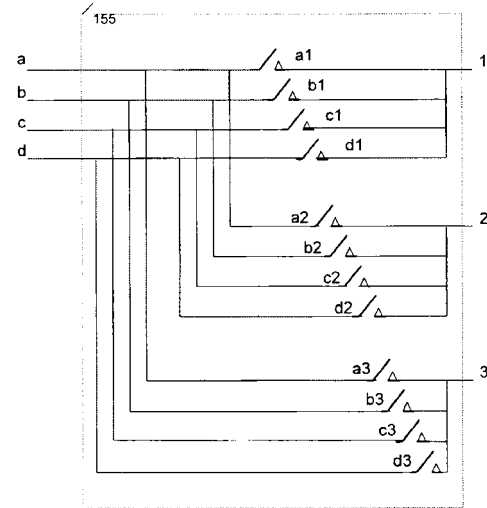


FIG. 8

【図 9】

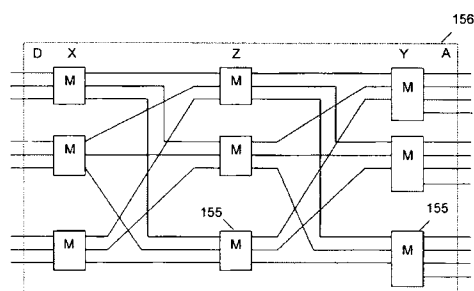


FIG. 9

【図 11】

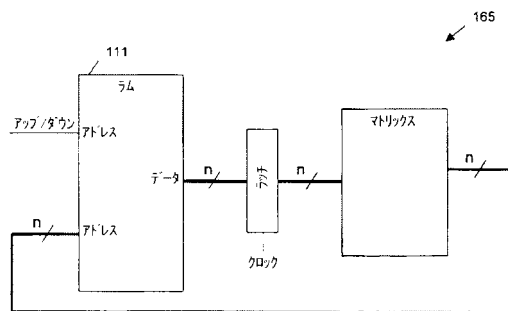


FIG. 11

【図 10】

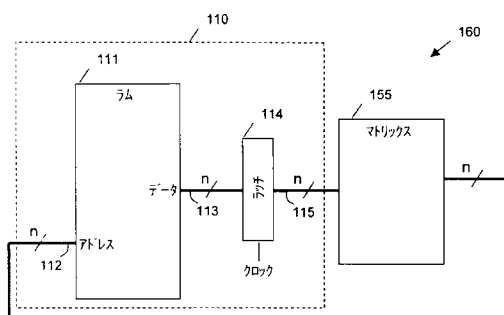


FIG. 10

【図 12】

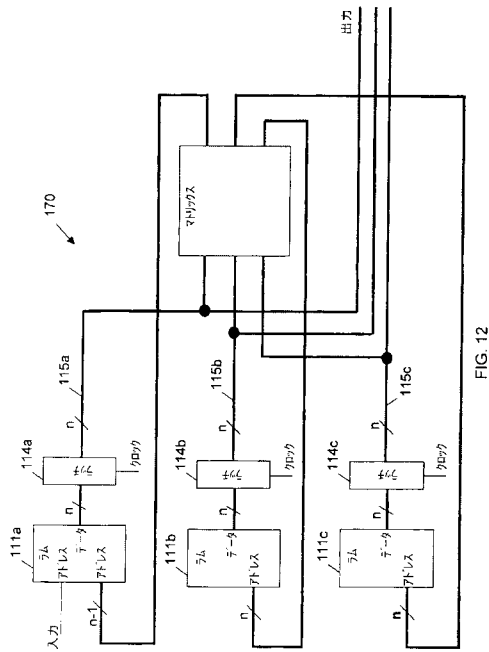


FIG. 12

【図 13】

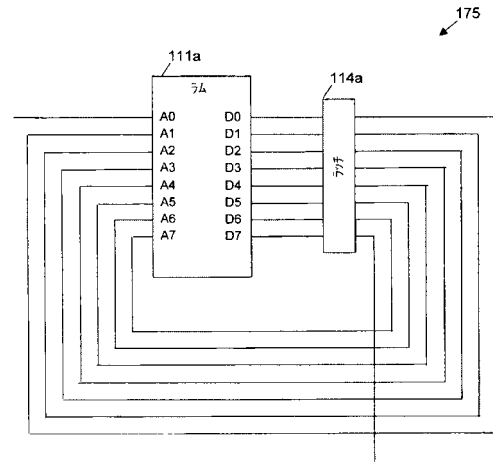


FIG. 13

【図 14】

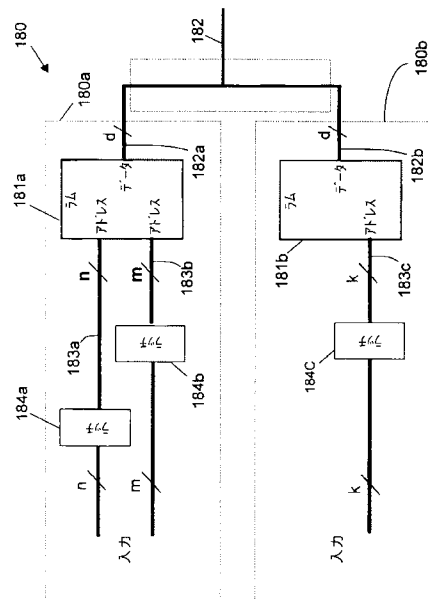


FIG. 14

【図 15】

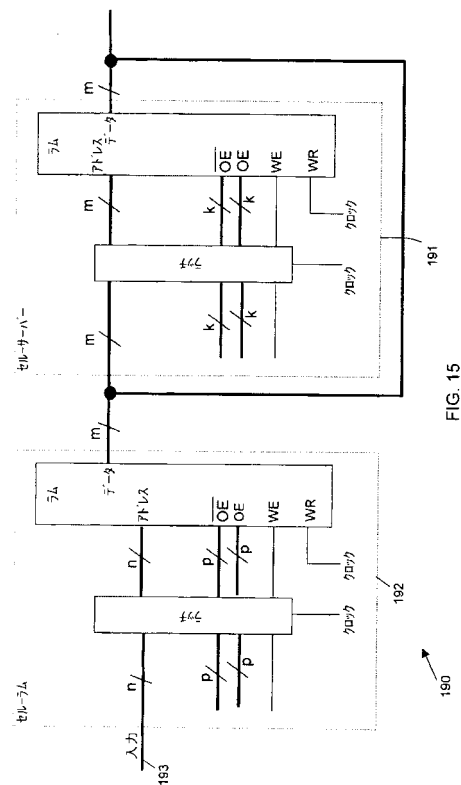


FIG. 15

【図 16】

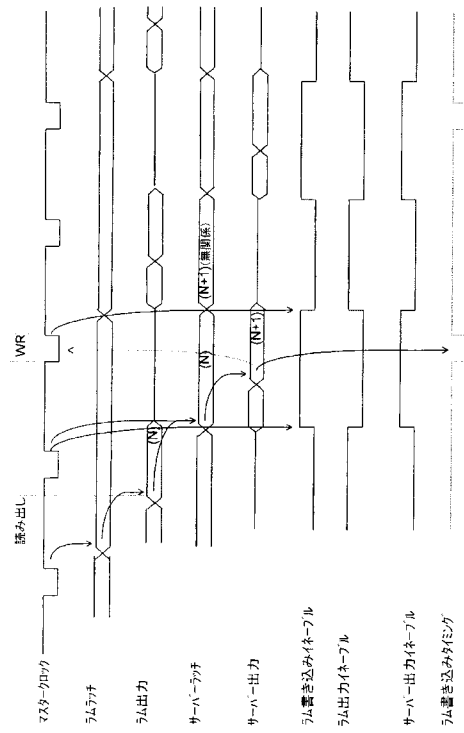


FIG. 16

【図 17】

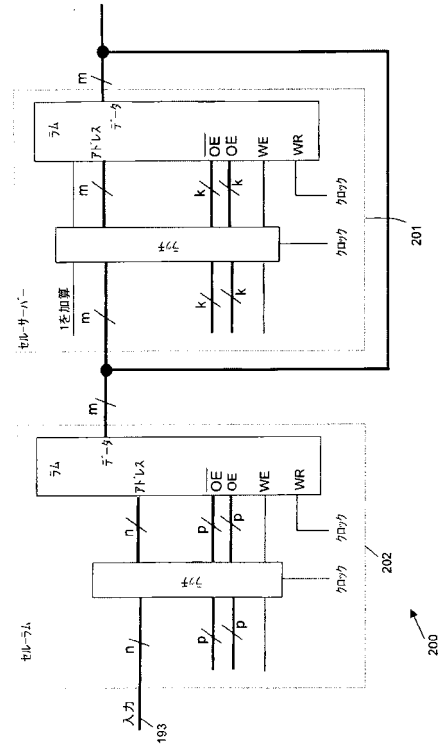


FIG. 17

【図 18】

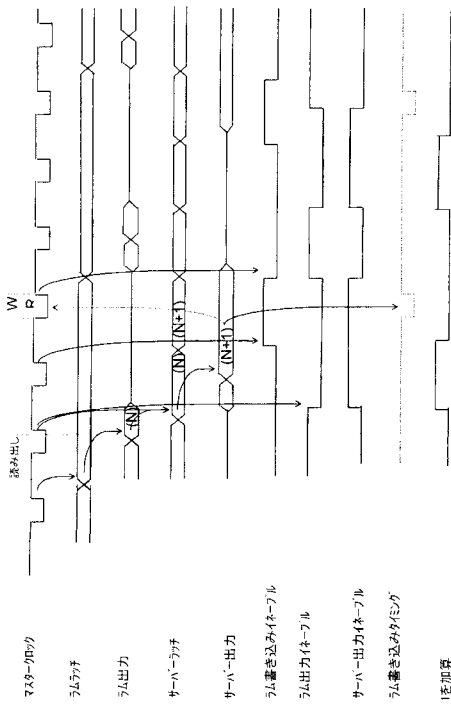


FIG. 18

【図 19】

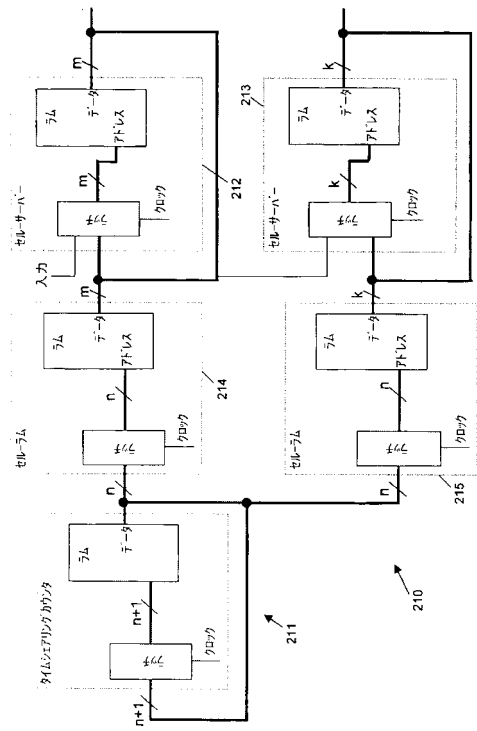


FIG. 19

【図 20】

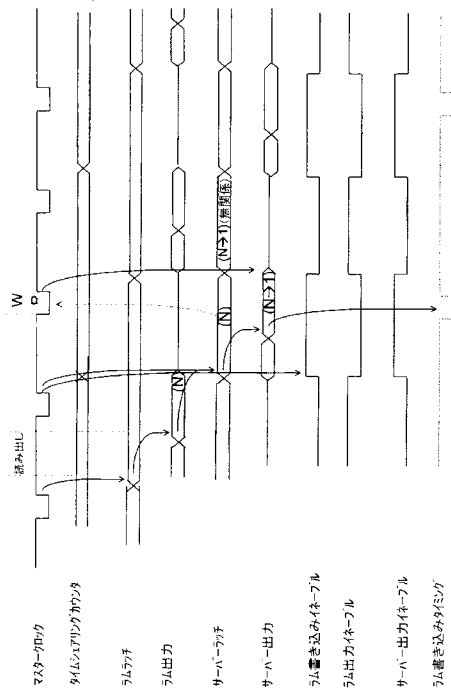


FIG. 20

【図 21】

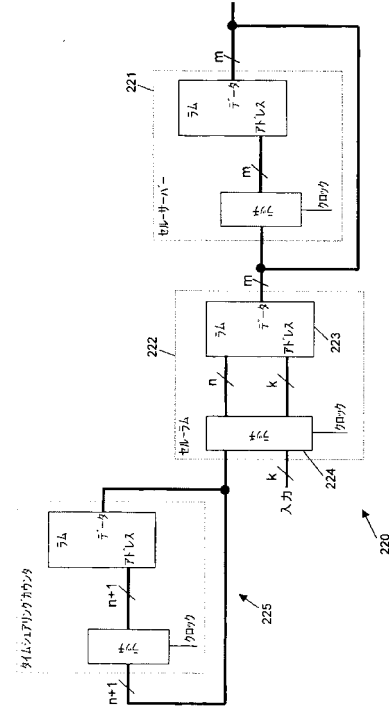


FIG. 21

【図 22】

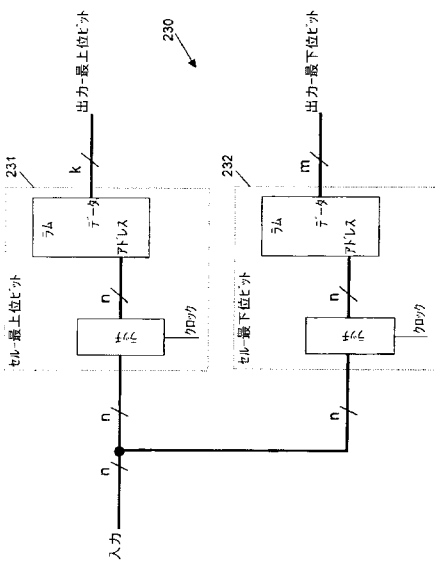


FIG. 22

【図 23】

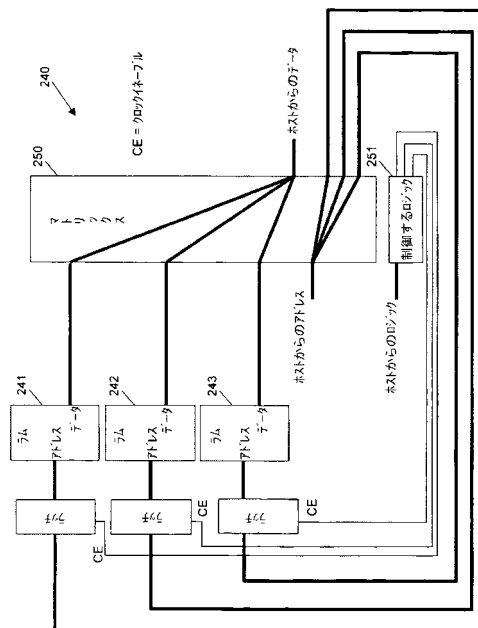


FIG. 23

【図 24】

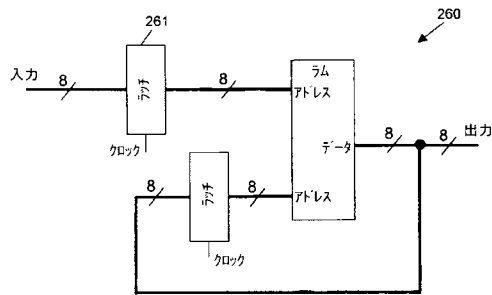


FIG. 24

【図 25】

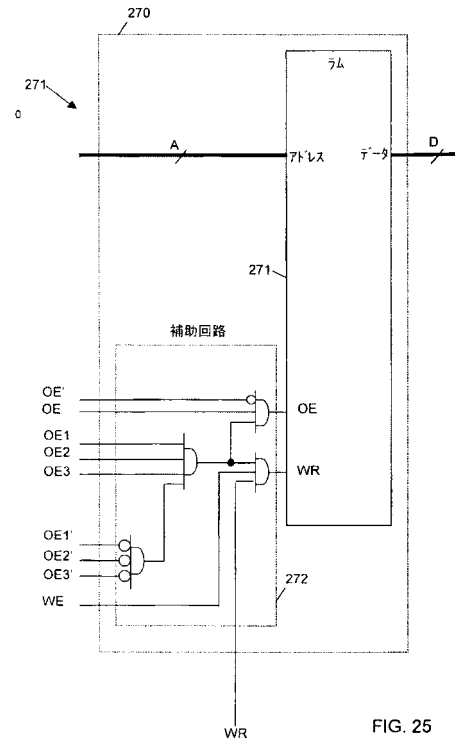


FIG. 25

【図 26】

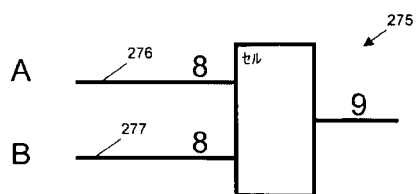


FIG. 26

【図 28】

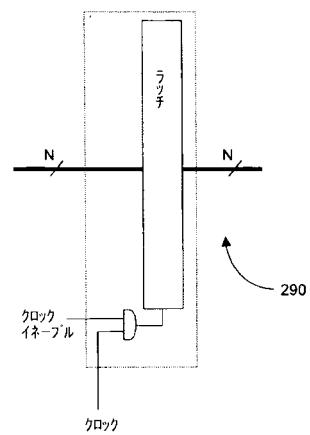


FIG. 28

【図 27】

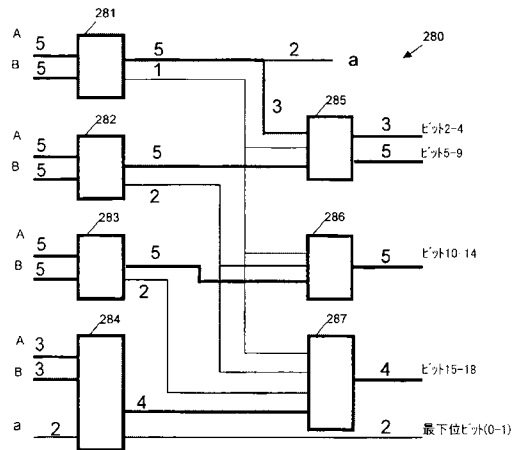


FIG. 27

【図 29】

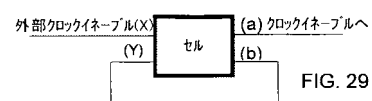


FIG. 29

【図30】

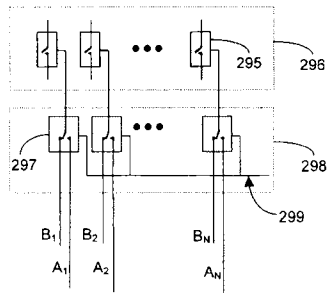


FIG. 30

【図32】

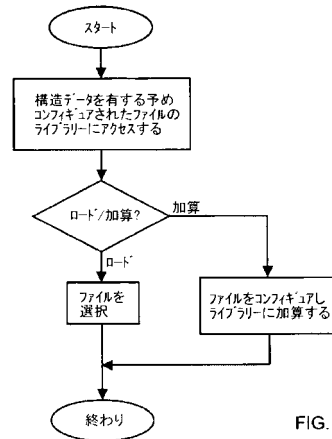


FIG. 32

【図31】

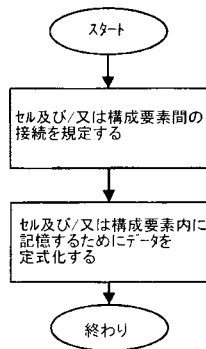


FIG. 31

【図33】

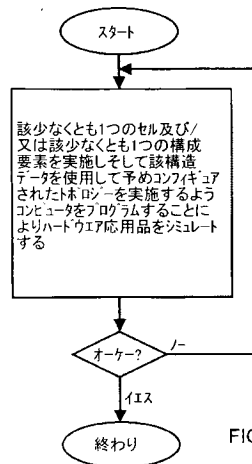


FIG. 33

【図34】

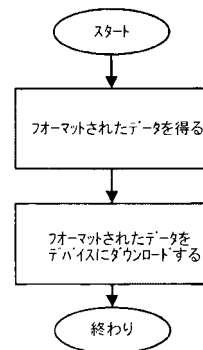


FIG. 34

【図 35】

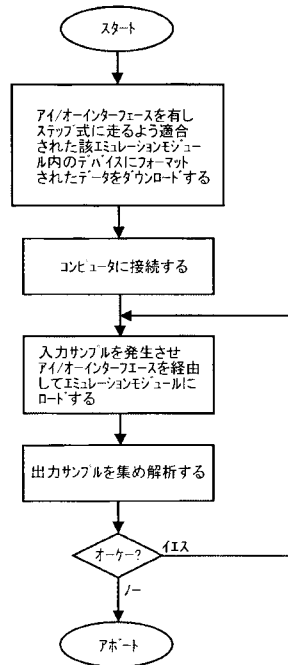


FIG. 35

【図 36】

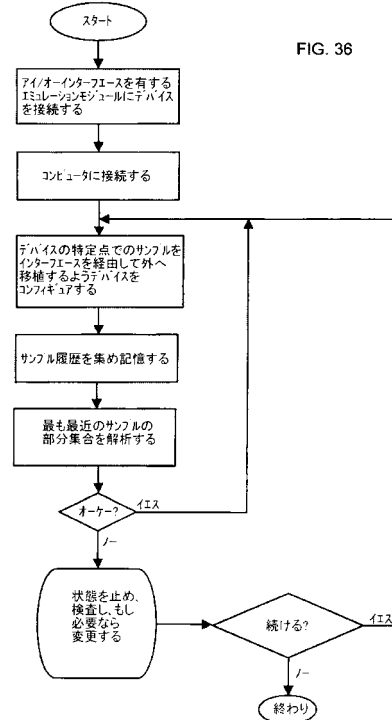


FIG. 36

【図 37】

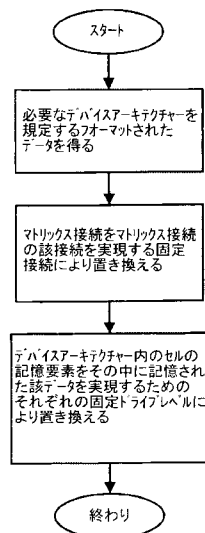


FIG. 37

【図 38】

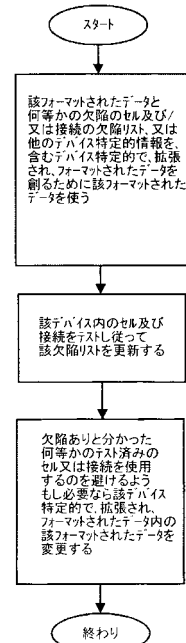
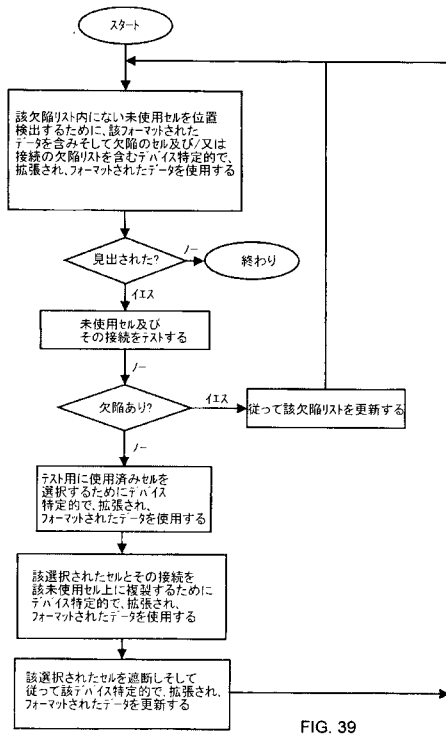
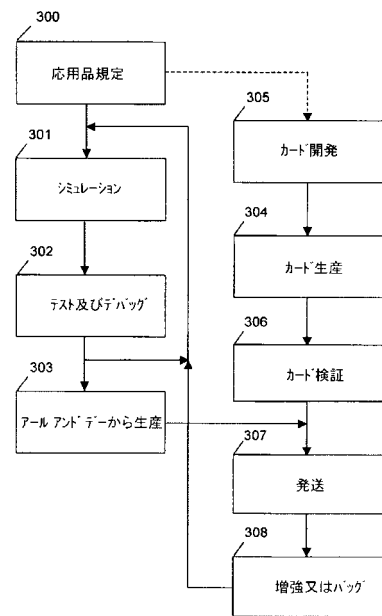


FIG. 38

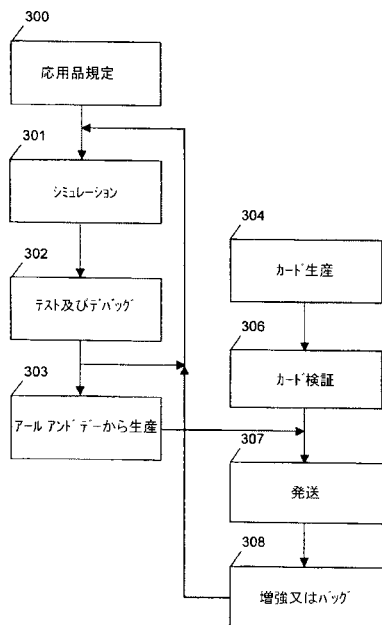
【図 39】



【図 40】



【図 41】



フロントページの続き

審査官 柳下 勝幸

- (56)参考文献 特開平 1 1 - 2 4 3 3 3 4 (J P , A)
特開平 0 8 - 3 3 0 9 4 5 (J P , A)
特開平 1 0 - 1 3 5 8 1 9 (J P , A)
米国特許第 0 5 8 2 1 7 7 3 (U S , A)
特開 2 0 0 0 - 0 3 1 2 8 4 (J P , A)

(58)調査した分野(Int.Cl. , D B 名)

H03K 19/173

G06F 17/50

H03K 19/177