

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第6207605号
(P6207605)

(45) 発行日 平成29年10月4日(2017.10.4)

(24) 登録日 平成29年9月15日(2017.9.15)

(51) Int.Cl.

F 1

G 06 F 15/82 (2006.01)
G 06 F 15/173 (2006.01)G 06 F 15/82 620 C
G 06 F 15/82 610 G
G 06 F 15/82 650 A
G 06 F 15/173 683 C

請求項の数 9 (全 24 頁)

(21) 出願番号 特願2015-523127 (P2015-523127)
 (86) (22) 出願日 平成25年7月9日 (2013.7.9)
 (65) 公表番号 特表2015-534659 (P2015-534659A)
 (43) 公表日 平成27年12月3日 (2015.12.3)
 (86) 國際出願番号 PCT/US2013/049755
 (87) 國際公開番号 WO2014/014712
 (87) 國際公開日 平成26年1月23日 (2014.1.23)
 審査請求日 平成28年6月23日 (2016.6.23)
 (31) 優先権主張番号 13/552,492
 (32) 優先日 平成24年7月18日 (2012.7.18)
 (33) 優先権主張国 米国(US)

早期審査対象出願

前置審査

(73) 特許権者 595168543
 マイクロン テクノロジー, インク.
 アメリカ合衆国, アイダホ州 83716
 -9632, ポイズ, サウス フェデ
 ラル ウエイ 8000
 (74) 代理人 100074099
 弁理士 大菅 義之
 (74) 代理人 100106851
 弁理士 野村 泰久
 (72) 発明者 ノイズ, ハロルド ビー.
 アメリカ合衆国, アイダホ州 83713
 , ポイズ, ウエスト アルバニー ストリ
 ート 10836

最終頁に続く

(54) 【発明の名称】状態機械エンジンをプログラミングするための方法およびデバイス

(57) 【特許請求の範囲】

【請求項1】

状態機械エンジンであって、
 バスインターフェースと、
 設定データは、最初に状態機械ラティスをプログラミングするための初期状態ベクトル
 データを含み、前記状態機械ラティスの外部にあり、前記バスインターフェースを介して
 設定データを受信するように構成されるプログラムバッファと、

前記状態機械ラティスの外部にあり、前記バスインターフェースを介して命令を受信す
 るように構成される命令バッファと、

前記命令が前記設定データに対応する命令であると前記命令バッファが判断した場合に
 、前記プログラムバッファに受信された前記設定データによって前記状態機械ラティスを
 プログラミングするプログラミングシステムと、

前記状態機械ラティスの外部にあり、前記プログラムバッファと並列に前記バスインターフェースを介して解析データを受信するように構成されるデータバッファと、

前記バスインターフェースを介して外部デバイスに前記状態機械ラティスの修復要素に
 関連した修復マップデータを提供するように構成され、前記状態機械ラティスの外部にあ
 る修復マップバッファと

を備え、

前記状態機械ラティスが、ブロックに組織化された複数のプログラム可能要素を備え、
 各プログラム可能要素が、前記解析データを解析し、前記解析の結果を出力するように構

10

20

成される複数のメモリセルを備え、前記状態機械ラティスが、ブロック間スイッチと、前記ブロック間スイッチとやり取りする信号のタイミングを制御するための第3のバッファを含み、

前記ブロック間スイッチは、前記複数のブロックの少なくとも2つに結合するように構成される、状態機械エンジン。

【請求項2】

前記プログラムバッファが、圧縮された設定データを前記プログラミングシステムに提供するように構成される、請求項1に記載の状態機械エンジン。 10

【請求項3】

前記プログラミングシステムが、前記圧縮された設定データを受信した後、前記設定データを復元するように構成される、請求項2に記載の状態機械エンジン。 10

【請求項4】

前記状態機械ラティスの外部にあり、前記バスインターフェースを介して追加の解析データを受信するように構成される追加のデータバッファを更に備え、前記解析データは前記追加のデータバッファには受信されず、前記追加の解析データは前記データバッファには受信されないように構成させる、請求項1に記載の状態機械エンジン。 10

【請求項5】

前記バスインターフェースは、前記データバッファのデータ許容量がフルになるまで解析されるべきデータを提供する請求項1に記載の状態機械エンジン。 10

【請求項6】

前記修復マップバッファが、圧縮された修復マップデータを提供するように構成される、請求項1に記載の状態機械エンジン。 20

【請求項7】

前記修復マップバッファが、非圧縮の修復マップデータを提供するように構成される、請求項1に記載の状態機械エンジン。 20

【請求項8】

前記外部デバイスが、前記修復マップバッファから提供される修復マップデータを受信し、前記修復マップデータを解析し、設定データの修正版を構築し、前記設定データの修正版を前記プログラムバッファに提供するように構成される、請求項1に記載の状態機械エンジン。 30

【請求項9】

前記プログラムバッファが、設定データをラティスプログラミングシステムに提供して、前記状態機械ラティスを設定するように構成される、請求項1に記載の状態機械エンジン。

【発明の詳細な説明】

【技術分野】

【0001】

本発明の実施形態は、概して、電子デバイスに関し、より詳細には、特定の実施形態では、データ解析のための並列デバイスを備える電子デバイスに関する。

【背景技術】

【0002】

複雑なデータ解析（例えば、パターン認識）は、従来のノイマン型コンピュータで実行するには能率が悪い場合がある。しかしながら、生物の脳、特に人間の脳は、複雑なデータ解析を行うのに長けている。最近の研究によれば、人間の脳は、新皮質内において階層性を持って組織化された一連のニューロンの層を用いてデータ解析を行っていることが示されている。階層の下層にあるニューロンは、例えば、感覚器からの「生の信号」を解析し、他方、上層のニューロンは、下位レベルのニューロンからの信号出力を解析する。この新皮質における階層型システムが、場合により脳のその他の領域と共同して、空間推論、意識的な思考、および複雑な言語などの高次の機能を人間が行うことを可能にする、複雑なデータ解析を達成する。 50

【0003】

コンピューティングの分野では、例えば、パターン認識タスクは、益々難しくなってきている。従来よりも大量のデータがコンピュータ間で送信され、ユーザが検出することを望むパターンの数は増加している。例えば、スパムまたはマルウェアの検出は、しばしば、特定のフレーズまたはコードの一部などのデータストリーム内のパターンを検索することによってなされる。パターンの数は、スパムおよびマルウェアの多様性と共に増加する。それは、新しい変種を検索するために新しいパターンが実装され得るためである。データストリームでこれらのパターンのそれぞれを検索することは、コンピューティングボトルネックを形成し得る。しばしば、データストリームが受信されると、各パターンが1度に1つずつ検索される。システムがデータストリームの次の部分を検索できるようになるまでの遅延は、パターンの数と共に増大する。このように、パターン認識は、データの受信を遅くし得る。10

【発明の概要】**【発明が解決しようとする課題】****【0004】**

データストリームでパターンを検索するハードウェアが設計されているが、このハードウェアは、所与の時間内に十分な量のデータを処理することができないことがしばしばである。データストリームを検索するように構成される一部のデバイスは、データストリームを複数の回路に分散することによって検索を行う。回路はそれぞれ、データストリームがパターンの一部分に一致するかどうかを判定する。しばしば、並列に動作する多数の回路が、それぞれ、ほぼ同時にデータストリームを検索する。しかしながら、生物の脳の拳動により近い拳動で複雑なデータ解析を行うことを事実上可能にするシステムはまだない。そのようなシステムの開発が望まれている。20

【図面の簡単な説明】**【0005】**

【図1】本発明の様々な実施形態による、状態機械エンジンを有するシステムの例を示す図である。

【図2】本発明の様々な実施形態による、図1の状態機械エンジンの有限状態機械(FSM)ラティスの例を示す図である。

【図3】本発明の様々な実施形態による、図2のFSMラティスのブロックの例を示す図である。30

【図4】本発明の様々な実施形態による、図3のブロックの行の例を示す図である。

【図5】本発明の様々な実施形態による、図4の行の2つ一組の例を示す図である。

【図6】本発明の様々な実施形態による、有限状態機械図の例を示す図である。

【図7】本発明の様々な実施形態による、FSMラティスが実装されている2階層の例を示す図である。

【図8】本発明の様々な実施形態による、図2のFSMラティスのプログラミングのために、コンパイラがソースコードをバイナリファイルに変換するための方法の例を示す図である。

【図9】本発明の様々な実施形態による、状態機械エンジンを示す図である。

【図10】本発明の様々な実施形態による、状態機械エンジンにおけるFSMラティスをプログラミングする方法を示す流れ図である。

【発明を実施するための形態】**【0006】**

ここで図面を参照すると、図1は、参照数字10で概して示される、プロセッサベースのシステムの実施形態を示している。システム10(例えば、データ解析システム)は、デスクトップコンピュータ、ノートパソコン、ポケットベル、携帯電話、個人用端末(personal organizer)、携帯音楽プレーヤー、制御回路、カメラなどの様々なタイプのいずれであってもよい。システム10はまた、ルータ、サーバ、またはクライアント(例えば、先述のタイプのコンピュータのうちの1つ)などのネットワークノード50

ドであってもよい。システム10は、コピー機、スキャナ、プリンタ、ゲーム機、テレビ、セットトップビデオ分配または録画システム、ケーブルテレビ用チューナ、パーソナルデジタルメディアプレーヤ、工場自動化システム、自動車用コンピュータシステムまたは医療機器などの何らかのその他の種類の電子デバイスであってもよい。(これらの種々のシステムの例を記述するために使用される用語は、本明細書で使用されるその他の多くの用語と同様に、いくつかの指示対象を共有する場合があり、そのため、列挙されたその他の項目によって、狭く解釈されるべきではない。)

【0007】

システム10などの典型的なプロセッサベースのデバイスでは、マイクロプロセッサなどのプロセッサ12が、システム10におけるシステム機能および要求の処理を制御する。さらに、プロセッサ12は、システム制御を分担する複数のプロセッサを備え得る。プロセッサ12は、システム10における要素のそれぞれに、直接的または間接的に接続されて、プロセッサ12が、システム10内またはシステム10の外部に記憶され得る命令を実行することによってシステム10を制御するようにし得る。

10

【0008】

本明細書に説明される実施形態によれば、システム10は、状態機械エンジン14を含み、これは、プロセッサ12の制御下で動作し得る。本明細書で用いる場合、状態機械エンジン14は、単一のデバイス(例えば、シングルチップ)を指す。状態機械エンジン14は、任意のオートマトン理論を用い得る。例えば、状態機械エンジン14は、限定されるものではないが、ミーリ型アーキテクチャ、ムーア型アーキテクチャ、有限状態機械(FSM)、決定性FSM(DFSM)、ビットパラレル状態機械(BPSM)などを含む、いくつかの状態機械アーキテクチャのうちの1つを用い得る。様々なアーキテクチャが用いられ得るが、説明の目的のために、本出願はFSMについて言及する。しかしながら、当業者であれば、説明される技法は、様々な状態機械アーキテクチャのうちのどれを用いても利用され得ることを理解するであろう。

20

【0009】

以下に詳述するように、状態機械エンジン14は、いくつかの(例えば、1つまたは2つ以上の)有限状態機械(FSM)ラティス(例えば、チップのコア)を含み得る。本出願の目的上、「ラティス」の語は、要素(例えば、ブールセル、カウンタセル、状態機械要素、状態遷移要素)の組織化されたフレームワーク(例えば、ルーティングマトリクス、ルーティングネットワーク、フレーム)を指す。さらに、「ラティス」は、任意の好適な形、構造、または階層構成(例えば、グリッド、キューブ、球、カスケード接続)を有し得る。各FSMラティスは、複数のFSMを実装し得、複数のFSMのそれぞれは、同じデータを並行して受信し、解析する。さらに、FSMラティスは、グループ(例えば、クラスタ)として構成され得て、FSMラティスのクラスタは、同じ入力データを並行して解析し得る。さらに、状態機械エンジン14のFSMラティスのクラスタは、階層構造として構成され得、階層構造の下位レベルにある状態機械ラティスからの出力が、上位レベルにある状態機械ラティスへの入力として用いられ得る。状態機械エンジン14のFSMラティスのクラスタを、階層構造にわたって直列にカスケード接続することによって、益々複雑なパターンを解析(例えば、評価、検索など)し得る。

30

【0010】

さらに、状態機械エンジン14の階層的並列構成に基づいて、状態機械エンジン14は、高い処理速度を利用するシステムにおける複雑なデータ解析(例えば、パターン認識)のために使用され得る。例えば、本明細書に説明される実施形態は、処理速度が1Gバイト/秒であるシステムに組み込まれ得る。したがって、状態機械エンジン14を利用して、高速メモリデバイスまたはその他の外部デバイスからのデータを迅速に解析し得る。状態機械エンジン14は、いくつかの基準(例えば、検索ターム)にしたがって、ほぼ同時に、例えば、单一デバイスサイクルの間、データストリームを解析し得る。状態機械エンジン14のあるレベルにあるFSMのクラスタ内のFSMラティスのそれぞれは、ほぼ同時にデータストリームから検索タームをそれぞれ受信し得、並列FSMラティスのそれぞれ

40

50

は、そのタームが、状態機械エンジン14を処理基準における次の状態に進めるかどうかを判定し得る。状態機械エンジン14は、例えば、100を越える、110を越える、または10000を越える比較的多数の基準にしたがってタームを解析し得る。それらが並列に動作することから、それらは、データストリームを遅延させることなく、比較的高帯域幅を有するデータストリーム、例えば、1Gバイト/秒より大きいかほぼ等しいデータストリームに対して基準を適用し得る。

【0011】

一実施形態では、状態機械エンジン14は、データストリームにおいて多数のパターンを認識（例えば、検出）するように構成され得る。例えば、状態機械エンジン14は、ユーザまたはその他のエンティティが解析を望む様々なタイプのデータストリームのうちの1つまたは2つ以上においてパターンを検出するために利用され得る。例えば、状態機械エンジン14は、インターネットを介して受信されるパケット、またはセルラーネットワークを介して受信される音声もしくはデータなどのネットワークを介して受信されるデータのストリームを解析するように構成され得る。一例では、状態機械エンジン14は、データストリームにおいてスパムまたはマルウェアを解析するように構成され得る。データストリームは、シリアルデータストリームとして受信され、ここでデータは、時間的、語彙的、または意味的に重要視される順序などの、意味のある順序で受信される。あるいは、データストリームは、並列もしくはバラバラの順序で受信され得、その後、例えば、インターネットを介して受信されるパケットを順序づけることによって、シリアルデータストリームへと変換される。一部の実施形態では、データストリームは、タームをシリアルに提示し得るが、タームのそれぞれを表すビットはパラレルに受信され得る。データストリームは、システム10の外部にあるソースから受信され得る、またはメモリ16などのメモリデバイスに問い合わせし、かつメモリ16に記憶されているデータからデータストリームを形成することによって、形成され得る。その他の例では、状態機械エンジン14は、特定の単語を綴る文字シーケンス、遺伝子を特定する遺伝子塩基対配列、画像の一部を形成する画像ファイルもしくはビデオファイルにおけるビットのシーケンス、プログラムの一部を形成する実行可能ファイルにおけるビットのシーケンス、または歌もしくは話された言葉の一部を形成するオーディオファイルにおけるビットのシーケンスを認識するように構成され得る。データのストリームは、バイナリフォーマットまたは例えば、10進数、ASCIIなどのその他のフォーマットの複数ビットのデータを含み得る。ストリームは、一桁または複数桁で、例えば、いくつかの2進数でデータを符号化し得る。

【0012】

理解されるように、システム10は、メモリ16を含む。メモリ16には、ダイナミックランダムアクセスメモリ(DRAM)、スタティックランダムアクセスメモリ(SRAM)、同期型DRAM(SDRAM)、倍データレートDRAM(DDR SDRAM)、DDR2 SDRAM、DDR3 SDRAMなどの揮発性メモリが含まれ得る。メモリ16はまた、読み出し専用メモリ(ROM)、PC-RAM、シリコン-酸化物-窒化物-酸化物-シリコン(SONOS)メモリ、金属-酸化物-窒化物-酸化物-シリコン(MONOS)メモリ、ポリシリコンフローティングゲートベースのメモリ、および/または揮発性メモリと共に使用されるその他のタイプの様々なアーキテクチャのフラッシュメモリ(例えば、NANDメモリ、NORメモリなど)などの不揮発性メモリを含み得る。メモリ16は、DRAMデバイスなどの1つまたは2つ以上のメモリデバイスを含み得、これは、状態機械エンジン14によって解析されるデータを提供し得る。本明細書で用いる場合、「提供する」の語は、一般的に、向ける、入力する、挿入する、送る、転送する、送信する、生成する、与える、出力する、配置する、書き込むなどを指し得る。このようなデバイスは、ソリッドステートドライブ(SSD)、マルチメディアメディアカード(MMC)、セキュアデジタル(SD)カード、コンパクトフラッシュ(CF)カード、または任意のその他の好適なデバイスと呼ばれ得る、あるいはこれを含み得る。さらに、このようなデバイスは、ユニバーサルシリアルバス(USB)、ペリフェラルコンボネントインターフェクト(PCI)、PCIエクスプレス(PCI-E)、スマートコン

10

20

30

40

50

ピュータシステムインターフェース(ＳＣＳＩ)、ＩＥＥＥ1394(ファイアーウイヤ)、または任意のその他の好適なインターフェースなどの任意の好適なインターフェースを介してシステム10に接続され得ることを理解されたい。フラッシュメモリデバイスなどのメモリ16の動作を助けるために、システム10は、メモリコントローラ(図示せず)を含み得る。理解されるように、メモリコントローラは、独立のデバイスであり得る、またはプロセッサ12と一体であり得る。加えて、システム10は、磁気記憶装置などの外部記憶装置18を含み得る。外部記憶装置は、入力データを状態機械エンジン14に提供し得る。

【0013】

システム10は、いくつかの追加的な要素を含み得る。例えば、コンパイラ20は、図8に関連して詳述するように、状態機械エンジン14を設定(例えば、プログラム)するために用いられ得る。入力デバイス22もまたプロセッサ12に接続されて、ユーザがデータをシステム10内に入力できるようにし得る。例えば、入力デバイス22は、状態機械エンジン14により後で解析するために、データをメモリ16内に入力するのに用いられ得る。入力デバイス22は、例えば、ボタン、スイッチング要素、キーボード、ライトペン、スタイルスペン、マウス、および/または音声認識システムを含み得る。ディスプレイなどの出力デバイス24もまた、プロセッサ12に接続され得る。ディスプレイ24は、例えば、ＬＣＤ、ＣＲＴ、ＬＥＤ、および/または聴覚ディスプレイを含み得る。システムはまた、インターネットなどのネットワークとインターフェースをとるために、ネットワークインターフェースカード(ＮＩＣ)などのネットワークインターフェースデバイス26を含み得る。理解されるように、システム10は、システム10のアプリケーションに応じて、多くのその他の構成要素を含み得る。

【0014】

図2～図5は、ＦＳＭラティス30の例を示している。一例では、ＦＳＭラティス30は、ブロック32のアレイを備える。説明するように、各ブロック32は、複数の選択的に接続可能なハードウェア要素(例えば、設定可能な要素および/または特定目的要素)を含み得、これは、ＦＳＭの複数の状態に対応する。ＦＳＭにおける状態と同様に、ハードウェア要素は入力ストリームを解析し、入力ストリームに応じて下流のハードウェア要素をアクティブ化し得る。

【0015】

設定可能な要素は、多くの異なる機能を実装するように設定(例えば、プログラム)され得る。例えば、設定可能な要素は、状態機械要素(ＳＭＥ)34、36(図5に示す)を含み得、これは、階層性を持って行38(図3および図4に示す)およびブロック32(図2および図3に示す)に組織化されている。ＳＭＥはまた、状態遷移要素(ＳＴＥ)とみなされ得る。階層性を持って組織化されたＳＭＥ34、36の間ににおいて信号をルーティングするために、ブロック間スイッチング要素40(図2および図3に示す)、ブロック内スイッチング要素42(図3および図4に示す)、および行内スイッチング要素44(図4に示す)を含む、設定可能なスイッチング要素の階層構造が用いられ得る。

【0016】

後述するように、スイッチング要素は、ルーティング構造と、バッファとを含み得る。ＳＭＥ34、36は、ＦＳＭラティス30によって実装されたＦＳＭの状態に対応し得る。ＳＭＥ34、36は、後述のように、設定可能なスイッチング要素を用いて互いに接続され得る。したがって、状態の関数に対応するようにＳＭＥ34、36を設定することによって、またＦＳＭにおける状態間の遷移に対応するようにＳＭＥ34、36を互いに選択的に接続することによって、ＦＳＭは、ＦＳＭラティス30に実装され得る。

【0017】

図2は、ＦＳＭラティス30の例の全体図を示している。ＦＳＭラティス30は、複数のブロック32を含み、これは、設定可能なブロック間スイッチング要素40と、互いに選択的に接続され得る。ブロック間スイッチング要素40は、導体46(例えば、ワイヤ、線など)と、バッファ48および50とを含み得る。一例では、バッファ48および50

10

20

30

40

50

0が含まれているのは、接続と、ブロック間スイッチング要素40との間ににおける信号のタイミングとを制御するためである。以下に詳述するように、バッファ48は、ブロック32間で送信されるデータをバッファリングするために提供され得、他方、バッファ50は、ブロック間スイッチング要素40間で送信されるデータをバッファリングするために提供され得る。加えて、ブロック32は、信号(例えば、データ)を受信し、データをブロック32に提供するための入力ブロック52(例えば、データ入力ポート)に選択的に接続され得る。ブロック32はまた、信号をブロック32から外部デバイス(例えば、別のFSMラティス30)に提供するための出力ブロック54(例えば、出力ポート)に選択的に接続され得る。FSMラティス30はまた、プログラミングインターフェース56を含んで、FSMラティス30を(例えば、画像、プログラムを介して)設定し得る。画像は、SME34、36の状態を設定(例えば、セット)し得る。つまり、画像は、SME34、36を設定して、入力ブロック52における所与の入力に対して特定の方法で反応し得る。例えば、SME34、36は、入力ブロック52で文字「a」を受信した場合、ハイ(high)信号を出力するようにセットされ得る。
10

【0018】

一例では、入力ブロック52、出力ブロック54、および/またはプログラミングインターフェース56は、レジスタとして実装されて、レジスタへの書き込みまたはレジスタからの読み出しが、それぞれの要素にデータを提供する、またはそれぞれの要素からデータを提供するようにし得る。このようにして、プログラミングインターフェース56に対応するレジスタに記憶された画像からのビットは、SME34、36にロードされ得る。
20 図2では、ブロック32、入力ブロック52、出力ブロック54、およびブロック間スイッチング要素40の間ににおいて特定数の導体(例えば、ワイヤ、線)が示されているが、その他の例では、より少数、またはより多数の導体が用いられ得ることを理解されたい。

【0019】

図3は、ブロック32の例を示している。ブロック32は、複数の行38を含み、これは、設定可能なブロック内スイッチング要素42と、互いに選択的に接続され得る。加えて、行38は、ブロック間スイッチング要素40により、別のブロック32内の別の行38と選択的に接続され得る。行38は、要素の対として組織された複数のSME34、36を含み、要素の対は、本明細書において、2つの要素からなる組(groups of two), (GOT)60と呼ぶ。一例では、ブロック32は、16の行38を備える。
30

【0020】

図4は、行38の例を示している。GOT60は、設定可能な行内スイッチング要素44によって、その他のGOT60および行38内の任意のその他の要素(例えば、特定目的要素58)に選択的に接続され得る。GOT60はまた、ブロック内スイッチング要素42により、他の行38にある他のGOT60と、あるいはブロック間スイッチング要素40により、他のブロック32にある他のGOT60と接続され得る。一例では、GOT60は、第1および第2の入力62、64と、出力66とを有する。図5と参照して詳述するように、第1の入力62は、GOT60の第1のSME34に接続され、第2の入力64は、GOT60の第2のSME36に接続される。
40

【0021】

一例では、行38は、第1および第2の複数の行相互接続導体68、70を含む。一例では、GOT60の入力62、64は、1つまたは2つ以上の行相互接続導体68、70に接続され得、出力66は、1つまたは2つ以上の行相互接続導体68、70に接続され得る。一例では、第1の複数の行相互接続導体68は、行38内の各GOT60の各SME34、36に接続され得る。第2の複数の行相互接続導体70は、行38内の各GOT60の1つのSME34、36のみに接続され得るが、GOT60の他のSME34、36には接続されない。一例では、図5に関連して詳述されるように、第2の複数の行相互接続導体70の最初の半分は、行38内のSME34、36の最初の半分(各GOT60から一方のSME34)に接続し得、第2の複数の行相互接続導体70の残りの半分は、行38内のSME34、36の残りの半分(各GOT60から他方のSME34、3
50

6)に接続し得る。第2の複数の行相互接続導体70と、SME34、36との間における制限された接続性は、本明細書において、「パリティ」と呼ばれる。一例では、行38はまた、カウンタ、設定可能なプール論理要素、ルックアップテーブル、RAM、現場で設定可能なゲートアレイ(FPGA)、特定用途向け集積回路ASIC)、設定可能なプロセッサ(例えば、マイクロプロセッサ)、または特定目的機能を実行するためのその他の要素などの特定目的要素58を含み得る。

【0022】

一例では、特定目的要素58は、カウンタ(本明細書において、カウンタ58とも呼ばれる)を備える。一例では、カウンタ58は、12ビットの設定可能な減算カウンタを備える。12ビットの設定可能なカウンタ58は、計数入力と、リセット入力と、ゼロカウント出力とを有する。計数入力は、アサートされると、カウンタ58の値を1つずつ減らす。リセット入力は、アサートされると、カウンタ58に関連するレジスタから初期値をロードさせる。12ビットのカウンタ58では、初期値として最大12ビットの数までロードされ得る。カウンタ58の値がゼロ(0)までデクリメントされると、ゼロカウント出力がアサートされる。カウンタ58はまた、少なくとも2つのモード、すなわち、パルスとホールドとを有する。カウンタ58がパルスモードに設定されると、カウンタ58がゼロに達し、かつクロックが供給されている場合、ゼロカウント出力がアサートされる。カウンタ58の次のクロックサイクルの間、ゼロカウント出力はアサートされる。その結果、カウンタ58は、クロックサイクルから時間的に外れる。その次のクロックサイクルでは、ゼロカウント出力は、アサートされた状態ではなくなる。カウンタ58がホールドモードに設定されると、ゼロカウント出力は、カウンタ58がゼロに向けてデクリメントするクロックサイクルの間、アサートされ、そして、リセット入力がアサートされることによってカウンタ58がリセットされるまで、アサートされた状態のままでいる。

【0023】

別の例では、特定目的要素58は、プール論理を備える。例えば、プール論理は、AND、OR、NAND、NOR、積和(SOP)、積和否定出力(NSOP)、和積否定出力(NPOS)、および和積(PoS)の関数などの論理関数を実行するために用いられ得る。このプール論理は、FSMラティス30における(本明細書で後述するように、FSMの終端ノードに対応する)終端状態のSMEからデータを抽出するために用いられ得る。抽出されたデータが用いられて、状態データを他のFSMラティス30に提供する、および/またはFSMラティス30を再設定するために用いられる、あるいは別のFSMラティス30を再設定するために用いられる設定データを提供する。

【0024】

図5は、GOT60の例を示している。GOT60は、第1のSME34と、第2のSME36とを含み、これらは、入力62、64を有し、それらの出力72、74をORゲート76および3入力マルチプレクサ78に接続されている。3入力マルチプレクサ78は、GOT60の出力66を、第1のSME34、第2のSME36、またはORゲート76のいずれかに接続するように設定され得る。ORゲート76は、両方の出力72、74と一緒に接続して、GOT60の共通出力66を形成するために用いられ得る。一例では、第1および第2のSME34、36が上述のようなパリティを呈し、そこでは、第1のSME34の入力62が行相互接続導体68の一部に接続され得、かつ第2のSME36の入力64がその他の行相互接続導体70に接続され得るのであるが、共通出力66が生成され得て、パリティ問題を克服し得る。一例では、GOT60内の2つのSME34、36は、スイッチング要素79のいずれかまたは両方を設定することにより、カスケード接続され得る、および/またはそれら自身にループバックされ得る。SME34、36は、SME34、36の出力72、74を他方のSME34、36の入力62、64に接続することにより、カスケード接続され得る。SME34、36は、出力72、74をそれら自身の入力62、64に接続することにより、それら自身にループバックされ得る。このように、第1のSME34の出力72は、第1のSME34の入力62および第2のSME36の入力64のどちらにも接続されなくてもよいし、それらの一方または両方に

10

20

30

40

50

接続されてもよい。

【0025】

一例では、状態機械要素34、36は、検出線82に並列に接続される複数のメモリセル80、例えば、ダイナミックランダムアクセスメモリ(DRAM)でしばしば用いられるものなどを備える。1つのこのようなメモリセル80は、ハイ(high)値またはロー(low)値(例えば、1または0)のいずれかに対応するものなどのデータ状態に設定され得るメモリセルを備える。メモリセル80の出力は、検出線82に接続され、メモリセル80への入力は、データストリーム線84上のデータに基づく信号を受信する。一例では、入力ブロック52における入力が復号されて、メモリセル80のうちの1つまたは2つ以上を選択する。選択されたメモリセル80は、その記憶されたデータ状態を出力として検出線82上に提供する。例えば、入力ブロック52で受信されたデータは、デコーダ(図示せず)に提供され得て、デコーダは、データストリーム線84のうちの1つまたは2つ以上を選択し得る。一例では、デコーダは、8ビットのACSI文字を256本のデータストリーム線84のうちの対応する1つに変換し得る。10

【0026】

したがって、あるメモリセル80は、そのメモリセル80がハイ値に設定され、かつデータストリーム線84上のデータがそのメモリセル80を選択している場合、ハイ信号を検出線82に出力する。データストリーム線84上のデータがそのメモリセル80を選択し、かつそのメモリセル80がロー値に設定されている場合、そのメモリセル80は、ロー信号を検出線82に出力する。メモリセル80からの検出線82への出力は、検出セル86によってセンスされる。20

【0027】

一例では、入力線62、64上の信号は、それぞれの検出セル86をアクティブまたは非アクティブ状態のいずれかに設定する。非アクティブ状態に設定されると、検出セル86は、それぞれの検出線82上の信号と関わりなく、ロー信号をそれぞれの出力72、74上に出力する。アクティブ状態に設定されると、検出セル86は、それぞれのSME34、36のメモリセル82のうちの1つからハイ信号が検出された場合、ハイ信号をそれぞれの出力線72、74上に出力する。アクティブ状態にある場合、検出セル86は、それぞれのSME34、36のメモリセル82のすべてからの信号がローである場合、ロー信号をそれぞれの出力線72、74上に出力する。30

【0028】

一例では、あるSME34、36は、256個のメモリセル80を含み、各メモリセル80は、異なるデータストリーム線84に接続される。このようにして、SME34、36は、データストリーム線84のうちの選択された1つまたは2つ以上が、そこにハイ信号を有する場合、ハイ信号を出力するようにプログラムされ得る。例えば、SME34は、第1のメモリセル80(例えば、ビット0)をハイに設定させ、すべての他のメモリセル80(例えば、ビット1~255)をローに設定され得る。それぞれの検出セル86がアクティブ状態にある場合、ビット0に対応するデータストリーム線84がそこにハイ信号を有するならば、SME34は、ハイ信号を出力72上に出力する。その他の例では、SME34は、複数のデータストリーム線84のうちの1つがそこにハイ信号を有するならば、適切なメモリセル80をハイ値に設定することにより、ハイ信号を出力するよう設定され得る。40

【0029】

一例では、メモリセル80は、関連するレジスタからビットを読み取ることによって、ハイ値またはロー値に設定され得る。このように、SME34は、コンパイラ20により形成された画像をレジスタ内に記憶して、レジスタ内のビットを関連するメモリセル80内にロードすることによって設定され得る。一例では、コンパイラ20により形成された画像は、ハイおよびロー(例えば、1および0)のビットからなる二値画像を含む。画像が、FSMラティス30を設定して、SME34、36をカスケード接続することによって、FSMを実装し得る。例えば、第1のSME34は、検出セル86をアクティブ状態50

に設定することによって、アクティブ状態に設定され得る。第1のSME34は、ビット0に対応するデータストリーム線84がそこにハイ信号を有するならば、ハイ信号を出力するように設定され得る。第2のSME36は、最初非アクティブ状態に設定され得るが、アクティブの際には、ビット1に対応するデータストリーム線84がそこにハイ信号を有するならば、ハイ信号を出力するように設定され得る。第1のSME34および第2のSME36は、第1のSME34の出力72を第2のSME36の入力64に接続するように設定することによって、カスケード接続され得る。このように、ビット0に対応するデータストリーム線84上でハイ信号がセンスされると、第1のSME34が、ハイ信号を出力72に出力して、第2のSME36の検出セル86をアクティブ状態に設定する。ビット1に対応するデータストリーム線84上でハイ信号がセンスされると、第2のSME36が、別のSME36をアクティブにするために、またはFSMラティス30からの出力のために、ハイ信号を出力74に出力する。

【0030】

一例では、単一のFSMラティス30が、単一の物理デバイス上に実装されるが、その他の例では、2つまたは3つ以上のFSMラティス30が、単一の物理デバイス（例えば、物理チップ）上に実装され得る。一例では、各FSMラティス30は、別個のデータ入力ブロック52と、別個の出力ブロック54と、別個のプログラミングインターフェース56と、別個の一式の設定可能な要素とを含み得る。さらに、それぞれの一式の設定可能な要素は、それらが対応するデータ入力ブロック52にあるデータに対して反応（例えば、ハイ信号またはロー信号を出力）し得る。例えば、第1のFSMラティス30に対応する第1の一式の設定可能な要素は、第1のFSMラティス30に対応する第1のデータ入力ブロック52にあるデータに対して反応し得る。第2のFSMラティス30に対応する第2の一式の設定可能な要素は、第2のFSMラティス30に対応する第2のデータ入力ブロック52に対して反応し得る。このように、各FSMラティス30は一式の設定可能な要素を含み、異なる入力データに対しては、それに応じて反応し得る一式の設定可能な要素も異なる。同様に、各FSMラティス30、およびそれぞれに対応する一式の設定可能な要素は、別個の出力を提供し得る。一部の例では、第1のFSMラティス30の出力ブロック54は、第2のFSMラティス30の入力ブロック52に接続されて、第2のFSMラティス30のための入力データが、第1のFSMラティス30からの出力データを、一連のFSMラティス30の階層的配列で含み得るようにし得る。

【0031】

一例では、FSMラティス30上にロードするための画像は、FSMラティス30内における設定可能な要素と、設定可能なスイッチング要素と、特定目的要素とを設定するための複数のデータビットを備える。一例では、画像は、FSMラティス30を設定するためにFSMラティス30上にロードされて、特定の入力に基づいて所望の出力を提供する。出力ブロック54は、データ入力ブロック52におけるデータに対する設定可能な要素の反応に基づくFSMラティス30からの出力を提供し得る。出力ブロック54からの出力は、所与のパターンの一一致を示す单一のビットと、複数のパターンに対する一致および不一致を示す、複数のビットを備えるワードと、所与の時点におけるすべてまたは特定の設定可能な要素の状態に対応する状態ベクトルとを含み得る。前述のように、いくつかのFSMラティス30は、状態機械エンジン14などの状態機械エンジンに含まれて、パターン認識（例えば、音声認識、画像認識など）、信号処理、画像処理、コンピュータビジョン、暗号作成法などのデータ解析を行い得る。

【0032】

図6は、FSMラティス30によって実装され得る有限状態機械(FSM)のモデル例を示している。FSMラティス30は、FSMの物理的な実装として構成（例えば、プログラム）され得る。FSMは、ダイアグラム90（例えば、有向グラフ、無向グラフ、擬グラフ）として表され得、これは、1つまたは2つ以上のルートノード92を含む。ルートノード92に加えて、FSMは、いくつかの標準的なノード94および終端ノード96から構成され得、これらは、ルートノード92およびその他の標準的なノード94に1つ

10

20

30

40

50

または2つ以上の枝98を介して連結される。ノード92、94、96は、FSMにおける状態に対応する。枝98は、状態間における遷移に対応する。

【0033】

ノード92、94、96のそれぞれは、アクティブ状態または非アクティブ状態のいずれかであり得る。非アクティブ状態にある場合、ノード92、94、96は、入力データに対して反応（例えば、応答）しない。アクティブ状態にある場合、ノード92、94、96は、入力データに対して反応し得る。入力データが上流のノード92、94と、下流のノード94、96との間ににおける枝98により指定される基準に一致する場合、上流のノード92、94は、そのノードに対して下流にあるノード94、96をアクティブにすることによって、入力データに対して反応し得る。例えば、文字「b」を指定する第1のノード94は、第1のノード94がアクティブであり、文字「b」が入力データとして受信される場合、第1のノード94に枝98により連結される第2のノード94をアクティブにする。本明細書で用いる場合、「上流」は、1つまたは2つ以上のノード間における関係を指し、そこでは、1つまたは2つ以上の他のノードの上流（または、ループまたはフィードバック構成の場合、それ自身の上流）にある第1のノードとは、第1のノードが、1つまたは2つ以上の他のノードをアクティブにし得る（または、ループの場合、自らをアクティブにし得る）状況を指す。同様に、「下流」とは、1つまたは2つ以上の他のノードの下流（または、ループの場合、それ自身の下流）にある第1のノードが、1つまたは2つ以上の他のノードによってアクティブにされ得る（または、ループの場合、自らによってアクティブにされ得る）関係を指す。このように、本明細書において「上流」および「下流」の語は、1つまたは2つ以上のノード間における関係を指すために用いられるが、これらの語は、ループまたは他の非線形のノード間経路の使用を排除するものではない。10

【0034】

ダイアグラム90では、ルートノード92は、最初にアクティブにされ得、また入力データがルートノード92からの枝98に一致する場合、下流のノード94をアクティブにし得る。ノード94は、入力データがノード94からの枝98に一致する場合、ノード96をアクティブにし得る。ダイアグラム90にわたりノード94、96は、入力データが受信されると、このようにしてアクティブにされ得る。終端ノード96は、入力データによる着目するシーケンスの一一致に対応する。したがって、終端ノード96のアクティブ化は、着目するシーケンスが、入力データとして受信されたことを示す。パターン認識機能を実装するFSMラティス30の文脈においては、終端ノード96に到達したということは、特定の着目するパターンが入力データにおいて検出されたことを示し得る。30

【0035】

一例では、それぞれのルートノード92、標準的なノード94、および終端ノード96は、FSMラティス30における設定可能な要素に対応し得る。各枝98は、設定可能な要素間の接続に対応し得る。このように、別の標準的なノード94または終端ノード96に遷移する（例えば、連結する枝98を有する）標準的なノード94は、別の設定可能な要素に遷移する（例えば、出力を提供する）設定可能な要素に対応する。一部の例では、ルートノード92は、対応する設定可能な要素を持たない。40

【0036】

理解されるように、ノード92をルートノードとして説明し、ノード96を終端ノードとして説明したが、必ずしも特定の「開始部」またはルートノードが存在しない場合もあるし、必ずしも特定の「終了部」または出力ノードが存在しない場合もある。換言すると、任意のノードが開始点であってもよく、任意のノードが出力を提供するのであってもよい。

【0037】

FSMラティス30がプログラムされる場合、設定可能な要素のそれぞれはまた、アクティブ状態または非アクティブ状態のいずれかであり得る。所与の設定可能な要素は、非アクティブの場合、対応するデータ入力ブロック52における入力データに対して反応し50

ない。アクティブな設定可能な要素は、データ入力ブロック 52 における入力データに反応し得、また入力データが設定可能な要素の設定に一致する場合、下流の設定可能な要素をアクティブにし得る。設定可能な要素が終端ノード 96 に対応する場合、設定可能な要素は、出力ブロック 54 に接続されて、一致の指標を外部デバイスに提供し得る。

【0038】

プログラミングインターフェース 56 を介して FSM ラティス 30 上にロードされた画像は、データ入力ブロック 52 におけるデータに対する反応に基づいてノードを順次アクティブにすることによって所望の FSM が実装されるように、設定可能な要素および特定目的要素、ならびに設定可能な要素および特定目的要素の間の接続を設定し得る。一例では、設定可能な要素は、单一のデータサイクル（例えば、単一の文字、文字 1 セット、単一のクロックサイクル）の間、アクティブに保たれ、次いで、上流の設定可能な要素によって再度アクティブにされない限り、非アクティブになる。

【0039】

終端ノード 96 は、圧縮された過去のイベントの履歴を記憶するとみなすことができる。例えば、終端ノード 96 に達するのに必要とされる 1 つまたは 2 つ以上の入力データのパターンは、その終端ノード 96 のアクティブ化により表され得る。一例では、終端ノード 96 が提供する出力は 2 進数であり、つまり、出力は、着目するパターンが一致したか否かを示す。ダイアグラム 90 における終端ノード 96 の標準的なノード 94 に対する比は、非常に小さいものであり得る。換言すると、FSM は非常に複雑であり得るが、FSM の出力は、比べれば小さい。

【0040】

一例では、FSM ラティス 30 の出力は、状態ベクトルを備え得る。状態ベクトルは、FSM ラティス 30 の設定可能な要素の状態（例えば、アクティブ化された、またはされていない）を備える。別の例では、状態ベクトルは、設定可能な要素が終端ノード 96 に対応するかを問わず、設定可能な要素のすべてまたはサブセットの状態を含み得る。一例では、状態ベクトルは、終端ノード 96 に対応する設定可能な要素のための状態を含む。このように、出力は、ダイアグラム 90 のすべての終端ノード 96 によって提供される指標の集まりを含み得る。状態ベクトルは、ワードで表現され得、ただし、各終端ノード 96 により提供される 2 進表示は、そのワードのうちの 1 ビットを備える。この終端ノード 96 の符号化は、FSM ラティス 30 のための検出状態の有効な指標を提供し得る（例えば、着目するシーケンスが検出されたかどうか、また、どの着目するシーケンスが検出されたのか）。

【0041】

上述のように、FSM ラティス 30 は、パターン認識機能を実装するようにプログラムされ得る。例えば、FSM ラティス 30 は、入力データにおける 1 つまたは 2 つ以上のデータシーケンス（例えば、シグネチャ、パターン）を認識するように構成され得る。着目するデータシーケンスが FSM ラティス 30 により認識されると、その認識の指標が出力ブロック 54 で提供され得る。一例では、パターン認識は、記号列（例えば、ASCII 文字）を認識して、例えば、ネットワークデータ内のマルウェアまたはその他のデータを識別し得る。

【0042】

図 7 は、階層構造 100 の例を示しており、ここで、2 つのレベルの FSM ラティス 30 が直列に接続され、データを解析するために用いられている。具体的には、図示の実施形態では、階層構造 100 は、第 1 の FSM ラティス 30A と、直列に配置された第 2 の FSM ラティス 30B とを含む。各 FSM ラティス 30 は、データ入力を受信するためのそれぞれのデータ入力ブロック 52 と、設定信号を受信するためのプログラミングインターフェースブロック 56 と、出力ブロック 54 とを含む。

【0043】

第 1 の FSM ラティス 30A は、入力データ、例えば、データ入力ブロックにおける生データを受信するように構成される。第 1 の FSM ラティス 30A は、上述のように入力

10

20

30

40

50

データに対して反応し、出力を出力ブロックに提供する。第1のF S Mラティス3 0 Aからの出力は、第2のF S Mラティス3 0 Bのデータ入力ブロックに送られる。次いで、第2のF S Mラティス3 0 Bは、第1のF S Mラティス3 0 Aによって提供された出力に基づいて反応し、対応する階層構造1 0 0 の出力信号1 0 2を提供し得る。直列接続する2つのF S Mラティス3 0 Aおよび3 0 Bのこの階層的な接続は、過去のイベントに関するデータを圧縮されたワードの形態で第1のF S Mラティス3 0 Aから第2のF S Mラティス3 0 Bに提供する手段を提供する。提供されるデータは、事実上、第1のF S Mラティス3 0 Aにより記録された複雑なイベント（例えば、着目するシーケンス）のサマリであり得る。

【0 0 4 4】

10

図7に示されるF S Mラティス3 0 A、3 0 Bの2階層構造1 0 0は、2つの独立なプログラムが同じデータストリームに基づいて動作するのを可能にする。2階層構造は、別領域としてモデル化されている生物の脳における視覚認識に類似するものであり得る。このモデルの下では、これらの領域は、事実上異なるパターン認識エンジンであり、それぞれが類似の計算機能（パターンマッチング）を実行するが、異なるプログラム（シグネチャ）を用いる。複数のF S Mラティス3 0 A、3 0 Bを一緒に接続することによって、データストリーム入力に関するさらなる知識が取得され得る。

【0 0 4 5】

階層構造の第1のレベル（第1のF S Mラティス3 0 Aにより実装される）は、例えば、生のデータストリームに対して直接、処理を実行し得る。つまり、生のデータストリームは、第1のF S Mラティス3 0 Aの入力ブロック5 2で受信され得、第1のF S Mラティス3 0 Aの設定可能な要素が、生のデータストリームに対して反応し得る。階層構造の第2のレベル（第2のF S Mラティス3 0 Bにより実装される）は、第1のレベルからの出力を処理し得る。つまり、第2のF S Mラティス3 0 Bは、第1のF S Mラティス3 0 Aの出力ブロック5 4からの出力を第2のF S Mラティス3 0 Bの入力ブロック5 2で受信し、第2のF S Mラティス3 0 Bの設定可能な要素は、第1のF S Mラティス3 0 Aの出力に反応し得る。このように、この例では、第2のF S Mラティス3 0 Bは、生のデータストリームを入力として受信するのではなく、第1のF S Mラティス3 0 Aが判定するところにより、生のデータストリームによって一致した着目するパターンの指標を受信する。第2のF S Mラティス3 0 Bは、第1のF S Mラティス3 0 Aからの出力データストリーム内のパターンを認識するF S Mを実装し得る。第2のF S Mラティス3 0 Bは、F S Mラティス3 0 Aからの出力を受信するだけでなく、複数のその他のF S Mラティスからの入力を受信し得ることを認識されたい。同様に、第2のF S Mラティス3 0 Bは、その他のデバイスからの入力を受信し得る。第2のF S Mラティス3 0 Bは、これらの複数の入力を組み合わせて、出力を作成し得る。

20

【0 0 4 6】

30

図8は、コンパイラが、ソースコードをラティス3 0などのF S Mラティスを設定するように構成された画像に変換して、F S Mを実装するための方法1 1 0の例を示している。方法1 1 0は、ソースコードを構文解析して構文木にするステップ（ブロック1 1 2）と、構文木をオートマトンに変換するステップ（ブロック1 1 4）と、オートマトンを最適化するステップ（ブロック1 1 6）と、オートマトンをネットリストに変換するステップ（ブロック1 1 8）と、ネットリストをハードウェア上に配置するステップ（ブロック1 2 0）と、ネットリストをルーティングするステップ（ブロック1 2 2）と、結果として生じる画像を発行するステップ（ブロック1 2 4）とを含む。

40

【0 0 4 7】

一例では、コンパイラ2 0は、ソフトウェア開発者がF S Mラティス3 0上でF S Mを実装するための画像を作成できるようにするアプリケーションプログラミングインターフェース（A P I）を含む。コンパイラ2 0は、ソースコードにおける正規表現の入力セットを、F S Mラティス3 0を設定するように構成される画像に変換するための方法を提供する。コンパイラ2 0は、ノイマン型アーキテクチャを有するコンピュータ用の命令によ

50

って実装され得る。これらの命令は、コンピュータ上のプロセッサ12に、コンパイラ20の機能を実装させ得る。例えば、命令は、プロセッサ12によって実行される場合、プロセッサ12に、プロセッサ12がアクセス可能なソースコード上のブロック112、114、116、118、120、122、および124で説明した動作を実行させ得る。

【0048】

一例では、ソースコードは、一群の記号の中で記号のパターンを識別するための検索文字列を記述する。検索文字列を記述するために、ソースコードは、複数の正規表現（`regular expression`）を含み得る。正規表現は、記号検索パターンを記述するための文字列であり得る。正規表現は、プログラミング言語、テキストエディタ、ネットワークセキュリティなどの様々なコンピュータドメイン内で広く使用されている。一例では、コンパイラでサポートされる正規表現は、非構造化データの解析のための基準を含む。非構造化データは、自由な形式のデータであって、データ内のワードに適用されたインデックスのないデータを含み得る。ワードは、データ内に、印字可能および印字不能の双方であるバイトの任意の組み合わせを含み得る。一例では、コンパイラは、Perl、（例えば、Perl互換の正規表現（PCRE））、PHP、Java、およびNET言語を含む、正規表現を実装するための複数の異なるソースコード言語に対応し得る。

【0049】

ブロック112で、コンパイラ20は、ソースコードを構文解析して、関連付けられて接続された演算子の配置を形成し、ここで、演算子のタイプが異なれば、ソースコードで実装される対応する関数も異なる（例えば、ソースコードの正規表現で実装される関数が異なる）。ソースコードを構文解析すると、ソースコードの一般的な表現を作成することができる。一例では、一般的な表現は、構文木として知られるツリーグラフの形で、ソースコードにおける正規表現の符号化表現を備える。本明細書で説明される例は、その配置を構文木（「抽象構文木」としても知られる）と呼ぶが、その他の例では、具象構文木またはその他の配置も用いられ得る。

【0050】

上述のように、コンパイラ20は、複数の言語のソースコードに対応し得るため、構文解析するステップは、言語にかかわらず、ソースコードを構文木などの非言語固有表現に変換する。こうして、コンパイラ20によるさらなる処理（ブロック114、116、118、120）は、ソースコードの言語にかかわらず、共通の入力構造から機能し得る。

【0051】

上述のように、構文木は、関連付けられて連結された複数の演算子を含む。構文木は、複数の異なるタイプの演算子を含み得る。つまり、演算子が異なれば、ソースコードにおける正規表現によって実装された対応する関数も異なり得る。

【0052】

ブロック114では、構文木は、オートマトンに変換される。オートマトンは、FSMのソフトウェアモデルを備え、それに応じて、決定性または非決定性として分類され得る。決定性オートマトンは、所与の時間に、単一の実行パスを有するが、非決定性オートマトンは、複数の並列実行パスを有する。オートマトンは、複数の状態を備える。構文木をオートマトンに変換するためには、演算子および構文木内での演算子間の関係が、状態間での遷移を伴って状態に変換される。一例では、オートマトンは、FSMラティス30のハードウェアに部分的に基づいて、変換され得る。

【0053】

一例では、オートマトンの入力記号は、アルファベット、数字0～9、およびその他の印字可能文字からなる記号を含む。一例では、入力記号は、0と255を含めて0から255までのバイト値によって表される。一例では、オートマトンは、グラフのノードが一式の状態に対応する有向グラフとして表され得る。一例では、入力記号に関する、状態pから状態qへの遷移、すなわち（p, q）が、ノードpからノードqへの有向連結によって示される。一例では、オートマトンを反転すると、何らかの記号に関する各遷移p → qが、同じ記号に関するq → pと反転される、新しいオートマトンが生成される。反

10

20

30

40

50

転では、開始状態が最終状態になり、最終状態が開始状態になる。一例では、オートマトンによって認識された（例えば、一致した）言語は、オートマトンに順次して入力する場合に、最終状態に達する、一式のすべての可能な文字列である。オートマトンによって認識された言語における各文字列は、開始状態から1つまたは2つ以上の最終状態へのパスを辿る。

【0054】

ブロック116では、オートマトンが構築された後、オートマトンは、とりわけ、その複雑性およびサイズを低減するように最適化される。オートマトンは、重複する状態を組み合わせることにより最適化され得る。

【0055】

ブロック118では、最適化されたオートマトンは、ネットリストに変換される。オートマトンのネットリストへの変換は、オートマトンの各状態をFSMラティス30上のハードウェア要素（例えば、SME34、36、その他の要素）へマッピングし、ハードウェア要素間の接続を判定する。

【0056】

ブロック120では、ネットリストが配置されて、ネットリストの各ノードに対応するターゲットデバイスの特定のハードウェア要素（例えば、SME34、36、特定目的要素58）を選択する。一例では、配置により、FSMラティス30に対する一般的な入力および出力制約に基づいて各特定のハードウェア要素が選択される。

【0057】

ブロック122では、選択されたハードウェア要素を、ネットリストによって記述される接続を達成するように互いに連結するために、配置されたネットリストがルーティングされて、設定可能なスイッチング要素（例えば、ブロック間スイッチング要素40、ブロック内スイッチング要素42、および行内スイッチング要素44）に対する設定を判定する。一例では、設定可能なスイッチング要素に対する設定は、選択されたハードウェア要素を接続するために用いられるFSMラティス30の特定の導体の判定、および設定可能なスイッチング要素に対する設定によって決定される。ルーティングは、ブロック120での配置よりも、ハードウェア要素間の接続に関するより具体的な制限を考慮に入れることができる。したがって、ルーティングは、FSMラティス30上の導体の実際の制限を前提として、適切な接続を行うために、グローバル配置によって決定されるように、いくつかのハードウェア要素の位置を調整し得る。

【0058】

ネットリストが配置され、ルーティングされると、配置され、ルーティングされたネットリストは、FSMラティス30の設定のために、複数のビットに変換され得る。複数のビットは、本明細書では画像と呼ばれる。

【0059】

ブロック124では、画像がコンパイラ20によって発行される。画像は、FSMラティス30の特定のハードウェア要素を設定するための複数のビットを備える。画像が複数のビット（例えば、0および1）を備える実施形態では、画像は、二値画像と呼ばれ得る。ビットは、SME34、36、特定目的要素58、および設定可能なスイッチング要素の状態を設定するために、FSMラティス30上にロードされて、プログラムされたFSMラティス30が、ソースコードによって記述された機能性を有するFSMを実装するようにし得る。配置（ブロック120）およびルーティング（ブロック122）は、FSMラティス30内の特定の位置における特定のハードウェア要素をオートマトンにおける特定の状態にマッピングし得る。したがって、画像内のビットは、特定のハードウェア要素を設定して、所望の機能（複数可）を実装し得る。一例では、画像は、機械コードをコンピュータ可読媒体に記憶することによって発行され得る。別の例では、画像は、画像をディスプレイ装置上に表示することによって発行され得る。さらに別の例では、画像は、その画像をFSMラティス30上にロードするための設定デバイスなどの別のデバイスにその画像を送ることにより発行され得る。さらにまた別の例では、画像は、その画像をFS

10

20

30

40

50

Mラティス(例えば、FSMラティス30)上にロードすることによって発行され得る。

【0060】

一例では、画像は、ビット値を画像からSME34、36およびその他のハードウェア要素に直接ロードするか、あるいは画像を1つまたは2つ以上のレジスタにロードしてから、ビット値をレジスタからSME34、36およびその他のハードウェア要素に書き込むかのいずれかによって、FSMラティス30上にロードされ得る。一例では、FSMラティス30のハードウェア要素(例えば、SME34、36、特定目的要素58、設定可能なスイッチング要素40、42、44)は、設定デバイスおよび/またはコンピュータが、画像を1つまたは2つ以上のメモリアドレスに書き込むことにより、画像をFSMラティス30上にロードし得るようにメモリマップされる。

10

【0061】

本明細書に説明される方法例は、少なくとも部分的に機械またはコンピュータで実装され得る。一部の例は、上述の例で説明したような方法を実行するように電子デバイスを設定するように操作可能な命令で符号化されたコンピュータ可読媒体または機械可読媒体を含み得る。このような方法の実装には、マイクロコード、アセンブリ言語コード、高水準言語コードなどのコードが含まれ得る。このようなコードは、様々な方法を実行するためのコンピュータ可読命令を含み得る。コードは、コンピュータプログラム製品の一部を形成し得る。さらに、コードは、実行中またはその他の時に、1つまたは2つ以上の揮発性もしくは不揮発性のコンピュータ可読媒体上に有形的に記憶され得る。これらのコンピュータ可読媒体は、限定されるものではないが、ハードディスク、取り外し可能磁気ディスク、取り外し可能光ディスク(例えば、コンパクトディスクおよびデジタルビデオディスク)、磁気カセット、メモリカードまたはスティック、ランダムアクセスメモリ(RAM)、読み出し専用メモリ(ROM)などを含み得る。

20

【0062】

ここで図9を参照すると、状態機械エンジン14の実施形態(例えば、シングルチップ上の単一のデバイス)が示されている。先に説明したように、状態機械エンジン14は、データをメモリ16などのソースからデータバスを介して受信するように構成され得る。図示の実施形態では、データは、ダブルデータレートスリー(DDR3)バスインターフェース130などのバスインターフェースを介して状態機械エンジン14に送信され得る。DDR3バスインターフェース130は、1Gバイト/秒を上回る、またはこれに等しい速度で、データを交換(例えば、提供および受信)可能であり得る。このようなデータ交換速度は、データが状態機械エンジン14によって解析される速度よりも速い場合もある。理解されるように、解析するデータのソースに応じて、バスインターフェース130は、状態機械エンジン14に対するデータソースへのデータ、およびデータソースからのデータを交換するための任意の好適なバスインターフェース、例えば、NANDフラッシュインターフェース、ペリフェラルコンポーネントインターフェース(PCIE)インターフェース、ギガビットメディア非依存インターフェース(GMMI)などであり得る。先述のように、状態機械エンジン14は、データを解析するように構成される1つまたは2つ以上のFSMラティス30を含む。各FSMラティス30は、2つのハーフラティスに分割され得る。図示の実施形態では、各ハーフラティスは、24KのSME(例えば、SME34、36)を含んで、ラティス30が48KのSMEを含むようにし得る。ラティス30は、図2～図5に関して先述したように配置される、任意の望ましい数のSMEを備え得る。また、1つのFSMラティス30のみが図示されているが、状態機械エンジン14は、先述のように、複数のFSMラティス30を含み得る。

30

【0063】

解析されるデータは、バスインターフェース130で受信され、いくつかのバッファおよびバッファインターフェースを介して、FSMラティス30に提供される。図示の実施形態では、データバスには、データバッファ132と、命令バッファ133と、プロセスバッファ134と、ランク間(I/R)バスおよびプロセスバッファインターフェース136とが含まれる。データバッファ132は、解析されるデータを受信し、一時的に記憶す

40

50

るよう構成される。一実施形態では、2つのデータバッファ132（データバッファAおよびデータバッファB）が存在する。データは、FSMラティス30による解析のために、2つのデータバッファ132のうちの1つに記憶され得るが、データは、他方のデータバッファ132から除去される。バスインターフェース130は、解析されるデータをデータバッファ132に、データバッファ132がフルになるまで提供するように構成され得る。データバッファ132がフルになった後、バスインターフェース130は、その他の目的のために用いることができるよう、解放されるように構成され得る（例えば、データバッファ132がさらなる解析されるデータを受信するために利用できるようになるまで、データストリームからその他のデータを提供する）。図示の実施形態では、データバッファ132は、それぞれ32Kバイトであり得る。命令バッファ133は、解析されるデータに対応する命令および状態機械エンジン14の設定に対応する命令などの命令を、プロセッサ12からバスインターフェース130を介して受信するように構成される。10 IRバスおよびプロセスバッファインターフェース136は、データをプロセスバッファ134に提供するのを助け得る。IRバスおよびプロセスバッファインターフェース136は、確実にデータがFSMラティス30によって順番に処理されるようにするのに用いられ得る。IRバスおよびプロセスバッファインターフェース136は、データ、タイミングデータ、パッキング命令などの交換を調整して、データが受信され、正確に解析されるようにし得る。概して、IRバスおよびプロセスバッファインターフェース136は、20 FSMラティス30の論理ランクにわたる複数のデータセットの並列解析を可能にする。例えば、複数の物理デバイス（例えば、状態機械エンジン14、チップ、別個のデバイス）は、ランクに構成され得、また互いにIRバスおよびプロセスバッファインターフェース136を介してデータを提供し得る。本出願の目的上、「ランク」の語は、同じチップセレクトに接続された一式の状態機械エンジン14を指す。図示の実施形態では、IRバスおよびプロセスバッファインターフェース136は、32ビットデータバスを含み得る。その他の実施形態では、IRバスおよびプロセスバッファインターフェース136は、128ビットデータバスなどの任意の好適なデータバスを含み得る。

【0064】

図示の実施形態では、状態機械エンジン14はまた、デコンプレッサ138と、コンプレッサ140とを含んで、状態機械エンジン14中に状態ベクトルデータを提供するのに役立つ。コンプレッサ140およびデコンプレッサ138は、連動して働いて、状態ベクトルデータが圧縮されて、データ提供時間を最小化し得るようにする。状態ベクトルデータを圧縮することにより、バス利用時間が最小限に抑えられ得る。コンプレッサ140およびデコンプレッサ138はまた、様々なバースト長の状態ベクトルデータを扱うように構成され得る。圧縮した状態ベクトルデータをパディングし、各圧縮領域の終点に関する指標を含めることにより、コンプレッサ140は、状態機械エンジン14にわたる処理スピード全体を改善し得る。コンプレッサ140は、FSMラティス30による解析後のマッチ結果データを圧縮するために用いられ得る。コンプレッサ140およびデコンプレッサ138はまた、設定データを圧縮および復元するのに用いられ得る。一実施形態では、コンプレッサ140およびデコンプレッサ138は、機能を無効にされて（例えば、オフにされる）、コンプレッサ140およびデコンプレッサ138との間で行き交うデータが修正されないようにし得る。3040

【0065】

上述のように、FSMラティス30の出力は、状態ベクトルを備え得る。状態ベクトルは、FSMラティス30のSME34、36の状態（例えば、アクティブ化された、またはされていない）と、カウンタ58の動的な（例えば、現時点の）カウントを備える。状態機械エンジン14は、状態ベクトルキャッシュメモリ142と、状態ベクトルメモリバッファ144と、状態ベクトル中間入力バッファ146と、状態ベクトル中間出力バッファ148とを有する状態ベクトルシステム141を含む。状態ベクトルシステム141は、FSMラティス30の複数の状態ベクトルを記憶し、状態ベクトルをFSMラティス30に提供して、FSMラティス30を提供された状態ベクトルに対応する状態に復元する50

のに用いられ得る。例えば、各状態ベクトルは、状態ベクトルキャッシュメモリ142に一時的に記憶され得る。つまり、SME34、36を新しいデータセット（例えば、検索ターム）のさらなる解析のために解放しながら、各SME34、36の状態が記憶されて、状態が復元され、かつ後にさらなる解析において用いられるようにし得る。一般的なキャッシュと同様に、状態ベクトルキャッシュメモリ142は、ここでは、例えば、FSMラティス30により迅速に検索し使用できるような状態ベクトルの記憶を可能にする。図示の実施形態では、状態ベクトルキャッシュメモリ142は、最大512の状態ベクトルまで記憶し得る。

【0066】

理解されるように、状態ベクトルデータは、ランク内の異なる状態機械エンジン14（例えば、チップ）の間で交換され得る。状態ベクトルデータは、様々な目的のために異なる状態機械エンジン14の間で交換され得る。様々な目的とは、状態機械エンジン14のFSMラティス30のSME34、36の状態を同期させるため、複数の状態機械エンジン14にわたって同じ機能を実行するため、複数の状態機械エンジン14にわたって結果を再現するため、複数の状態機械エンジン14にわたって結果をカスケード式に送るため、複数の状態機械エンジン14にわたってカスケード式に送られたデータの解析に用いられるSME34、36の状態の履歴を記憶するためなどである。さらに、状態機械エンジン14内では、状態ベクトルデータが、FSMラティス30のSME34、36を迅速に設定するために用いられ得ることに留意されたい。例えば、状態ベクトルデータが用いられて、SME34、36の状態を初期化された状態（例えば、新しい検索タームを検索するため）に復元したり、SME34、36の状態を先の状態（例えば、先に検索した検索タームを検索するため）に復元したり、SME34、36の状態をカスケード接続する構成に設定されるように変更したりし得る（例えば、カスケード型検索における検索タームを検索するため）。ある実施形態では、状態ベクトルデータは、バスインターフェース130に提供されて、状態ベクトルデータがプロセッサ12に提供され得るようにし得る（例えば、状態ベクトルデータの解析のため、状態ベクトルデータを再設定して、修正を加えるため、状態ベクトルデータを再設定して、SME34、36の効率を高めるためなど）。

【0067】

例えば、ある実施形態では、状態機械エンジン14は、キャッシュされた状態ベクトルデータ（例えば、状態ベクトルシステム141により記憶されたデータ）をFSMラティス30から外部デバイスに提供し得る。外部デバイスは、状態ベクトルデータを受信し、状態ベクトルデータを修正し、FSMラティス30を設定するために修正した状態ベクトルデータを状態機械エンジン14に提供し得る。このように、外部デバイスは、状態ベクトルデータを修正して、状態機械エンジン14が所望どおりに状態をスキップ（例えば、方々に飛び越し）できるようにし得る。

【0068】

状態ベクトルキャッシュメモリ142は、状態ベクトルデータを任意の好適なデバイスから受信し得る。例えば、状態ベクトルキャッシュメモリ142は、FSMラティス30、別のFSMラティス30（例えば、IRバスおよびプロセスバッファインターフェース136を介して）、デコンプレッサ138などから状態ベクトルを受信し得る。図示の実施形態では、状態ベクトルキャッシュメモリ142は、状態ベクトルメモリバッファ144を介して、その他のデバイスから状態ベクトルを受信し得る。さらに、状態ベクトルキャッシュメモリ142は、状態ベクトルデータを任意の好適なデバイスに提供し得る。例えば、状態ベクトルキャッシュメモリ142は、状態ベクトルメモリバッファ144と、状態ベクトル中間入力バッファ146と、状態ベクトル中間出力バッファ148とに、状態ベクトルデータを提供し得る。

【0069】

状態ベクトルメモリバッファ144、状態ベクトル中間入力バッファ146、および状態ベクトル中間出力バッファ148などの追加的なバッファは、状態ベクトルキャッシュ

10

20

30

40

50

メモリ 142と共に利用されて、状態機械エンジン 14 中でパケットを割り込ませた別個のデータセットを処理しながら、状態ベクトルの迅速な検索および記憶を提供し得る。図示の実施形態では、状態ベクトルメモリバッファ 144、状態ベクトル中間入力バッファ 146、および状態ベクトル中間出力バッファ 148 のそれぞれは、1つの状態ベクトルを一時的に記憶するように構成され得る。状態ベクトルメモリバッファ 144 は、状態ベクトルデータを任意の好適なデバイスから受信するために、また状態ベクトルデータを任意の好適なデバイスに提供するために用いられ得る。例えば、状態ベクトルメモリバッファ 144 は、FSM ラティス 30、別の FSM ラティス 30（例えば、IR バスおよびプロセスバッファインターフェース 136 を介して）、デコンプレッサ 138、および状態ベクトルキャッシュメモリ 142 から状態ベクトルを受信するために用いられ得る。別の例として、状態ベクトルメモリバッファ 144 は、IR バスおよびプロセスバッファインターフェース 136（例えば、その他の FSM ラティス 30 のため）、コンプレッサ 140、および状態ベクトルキャッシュメモリ 142 に状態ベクトルデータを提供するために用いられ得る。
10

【0070】

同様に、状態ベクトル中間入力バッファ 146 は、状態ベクトルデータを任意の好適なデバイスから受信するために、また状態ベクトルデータを任意の好適なデバイスに提供するために用いられ得る。例えば、状態ベクトル中間入力バッファ 146 は、FSM ラティス 30（例えば、IR バスおよびプロセスバッファインターフェース 136 を介して）、デコンプレッサ 138、および状態ベクトルキャッシュメモリ 142 から状態ベクトルを受信するために用いられ得る。別の例として、状態ベクトル中間入力バッファ 146 は、FSM ラティス 30 に状態ベクトルを提供するために用いられ得る。さらに、状態ベクトル中間出力バッファ 148 は、状態ベクトルを任意の好適なデバイスから受信するために、また状態ベクトルを任意の好適なデバイスに提供するために用いられ得る。例えば、状態ベクトル中間出力バッファ 148 は、FSM ラティス 30 および状態ベクトルキャッシュメモリ 142 から状態ベクトルを受信するために用いられ得る。別の例として、状態ベクトル中間出力バッファ 148 は、FSM ラティス 30（例えば、IR バスおよびプロセスバッファインターフェース 136 を介して）およびコンプレッサ 140 に状態ベクトルを提供するために用いられ得る。
20

【0071】

着目する結果が FSM ラティス 30 によって生成されると、マッチ結果がマッチ結果メモリ 150 に記憶され得る。つまり、一致（例えば、着目するパターンの検出）を示す「マッチベクトル」が、マッチ結果メモリ 150 に記憶され得る。次いで、マッチング結果は、バスインターフェース 130 を介して、例えばプロセッサ 12 に送信するために、マッチバッファ 152 に送信され得る。先述したように、マッチ結果は、圧縮され得る。
30

【0072】

さらなるレジスタおよびバッファも同様に、状態機械エンジン 14 に提供され得る。例えば、状態機械エンジン 14 は、制御およびステータスレジスタ 154 を含み得る。加えて、プログラムバッファシステム（例えば、修復マップおよびプログラムバッファ 156）が、最初に FSM ラティス 30 をプログラミングするために提供され得る。例えば、初期（例えば、開始）状態ベクトルデータは、プログラムバッファシステムから（例えば、デコンプレッサ 138 を介して）FSM ラティス 30 に提供され得る。デコンプレッサ 138 は、FSM ラティス 30 をプログラムするために提供される設定データ（例えば、状態ベクトルデータ、ルーティングスイッチデータ、SME 34、36 の状態、布尔関数データ、カウンタデータ、マッチ MUX データ）を復元するために用いられ得る。
40

【0073】

同様に、修復マップバッファシステム（例えば、保存および修復マップバッファ 158）はまた、セットアップおよび使用のためのデータ（例えば、保存および修復マップ）の記憶のために提供され得る。修復マップバッファシステムによって記憶されるデータは、どの SME 34、36 が修復されたかを識別するデータなどの修復されたハードウェア要
50

素に対応するデータを含み得る。修復マップバッファシステムは、任意の好適な方法でデータを受信し得る。例えば、データは、最終製造試験中にデバイス上でなされた修復のマッピングを提供する「ヒューズマップ」メモリから、修復マップバッファ 158 に提供され得る。別の例として、修復マップバッファシステムは、標準的なプログラミングファイルを修正（例えば、カスタマイズ）するために用いられるデータを含んで、標準的なプログラミングファイルが、修復されたアーキテクチャを有する FSM ラティス 30 で動作し得るようにし得る（例えば、FSM ラティス 30 における不良の SME 34、36 は、使用されないようにバイパスされる）。コンプレッサ 140 は、ヒューズマップメモリから修復マップバッファ 158 に提供されたデータを圧縮するために用いられ得る。図示のように、バスインターフェース 130 は、データをプログラムバッファ 156 に提供し、修復マップバッファ 158 からのデータを提供するために用いられ得る。理解されるように、プログラムバッファ 156 に提供されるデータおよび／または修復マップバッファ 158 から提供されるデータは、圧縮され得る。一部の実施形態では、データは、状態機械エンジン 14 の外部にあるデバイスを（例えば、プロセッサ 12、メモリ 16、コンパイラ 20 など）介して、バスインターフェース 130 に提供される、および／またはバスインターフェース 130 から受信される。状態機械エンジン 14 の外部にあるデバイスは、修復マップバッファ 158 から提供されるデータを受信し、データを記憶し、データを解析し、データを修正し、および／または新しいもしくは修正したデータをプログラムバッファ 156 に提供するように構成され得る。

【0074】

10

状態機械エンジン 14 は、FSM ラティス 30 をプログラムするのに用いられるラティスプログラミングシステム 159 を含む。図示のように、ラティスプログラミングシステム 159 は、命令キュー 133 からデータ（例えば、設定命令）を受信し得る。さらに、ラティスプログラミングシステム 159 は、プログラムバッファ 156 からデータ（例えば、設定データ）を受信し得る。ラティスプログラミングシステム 159 は、設定命令および設定データを用いて、FSM ラティス 30 を設定し得る（例えば、ルーティングイッチ、SME 34、36、プールセル、カウンタ、マッチMUX を設定し得る）。ラティスプログラミングシステム 159 は、デコンプレッサ 138 を用いて、データを圧縮し、コンプレッサ 140 を用いて、データを圧縮し得る（例えば、プログラムバッファ 156 および修復マップバッファ 158 と交換されたデータのため）。

20

【0075】

30

ここで図 10 を参照すると、状態機械エンジン 14 における FSM ラティス 30 をプログラムする方法 160 を示す流れ図が示されている。本方法 160 は、設定データをプログラムバッファ 156 に提供するステップ（ブロック 162）と、設定命令を命令バッファ 133 に提供するステップ（ブロック 164）と、設定の更新を開始するステップ（ブロック 166）と、命令を命令バッファ 133 から読み出すステップ（ブロック 168）と、命令が設定命令であるかどうかを判定するステップ（ブロック 170）と、設定データをプログラムバッファ 156 から読み出し、圧縮されている場合には、設定データを復元するステップ（ブロック 172）と、設定データを FSM ラティス 30 に提供するステップ（ブロック 174）とを含み得る。

40

【0076】

一実施形態では、ブロック 162において、プロセッサ 12 は、設定データをプログラムバッファシステム（例えば、復元マップおよびプログラムバッファ 156）にバスインターフェース 130 を介して提供し得る。さらに、プロセッサ 12 は、設定命令を命令バッファ 133 にバスインターフェース 130 を介して提供し得る（ブロック 164）。ブロック 166 では、設定の更新が開始され得る（例えば、プロセッサ 12、ラティスプログラミングシステム 159、または状態機械エンジン 14 を介して）。次いで、ブロック 168 では、ラティスプログラミングシステム 159 は、命令バッファ 133 に記憶された命令を読み出し得る。次に、ブロック 170 では、ラティスプログラミングシステム 159 は、命令バッファ 133 から読み出された命令が設定命令かどうかを判定し得る。命

50

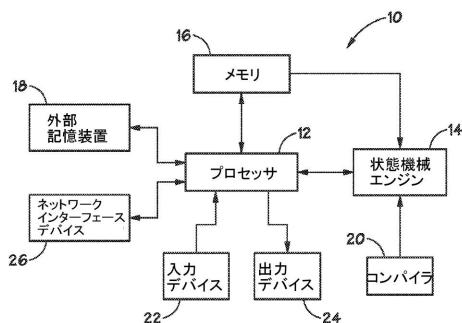
令が設定命令ではない場合、ラティスプログラミングシステム 159 は、ブロック 168 で、命令バッファ 133 に記憶されたさらなる命令を読み出す。しかしながら、命令が設定命令である場合、ラティスプログラミングシステム 159 は、プログラムバッファシステムに記憶された設定データを読み出す（ブロック 172）。一部の実施形態では、設定データは、圧縮され得る。したがって、さらにブロック 172 では、ラティスプログラミングシステム 159 は、設定データを復元し得る（例えば、デコンプレッサ 138 を介して）。次いで、ラティスプログラミングシステム 159 は、設定データを FSM ラティス 30 に提供する（ブロック 174）。

【0077】

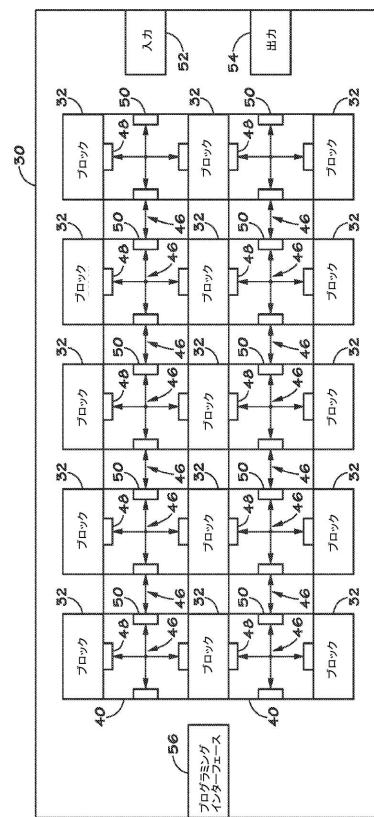
本開示は、様々な修正および代替形態に利用可能であるが、具体的な実施形態が例として図面に示され、本明細書に詳述されてきた。しかしながら、本発明が、開示された特定の形態に限定されることを意図するものではないことを理解されたい。むしろ、本発明は、以下の添付の特許請求の範囲によって定義される、本開示の精神および範囲内に含まれるすべての修正、均等物、および代替策を包含するものである。

10

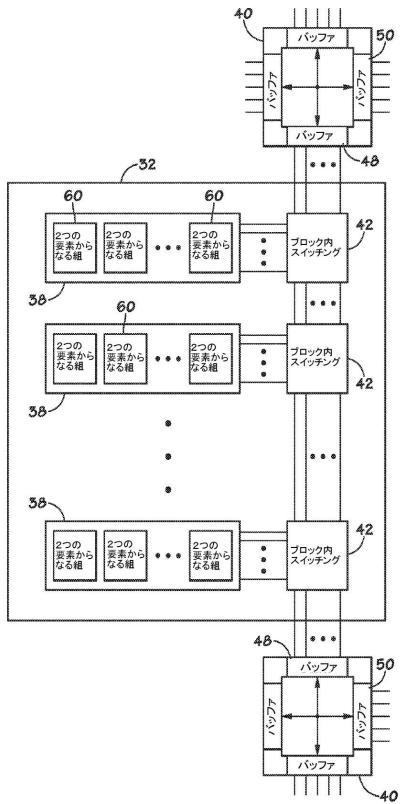
【図1】



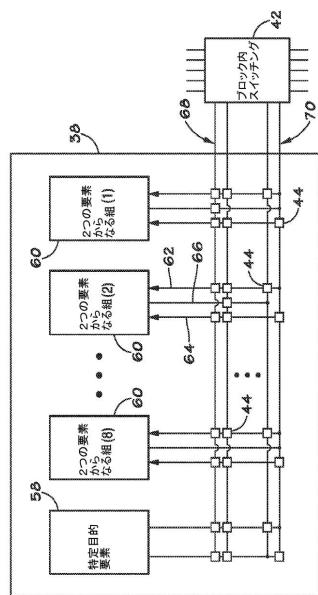
【図2】



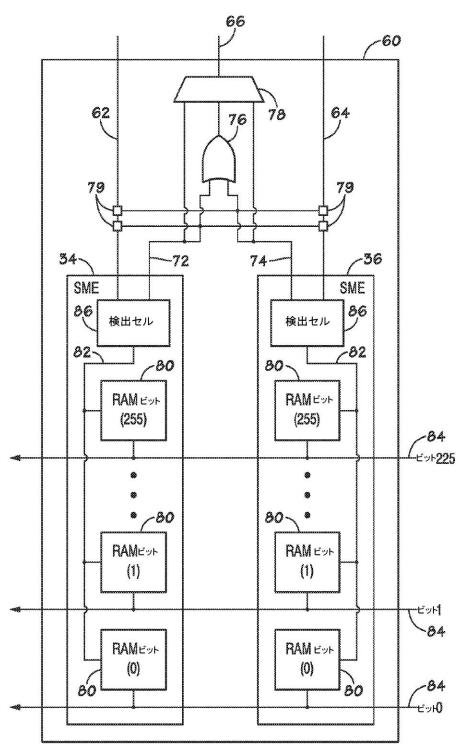
【図3】



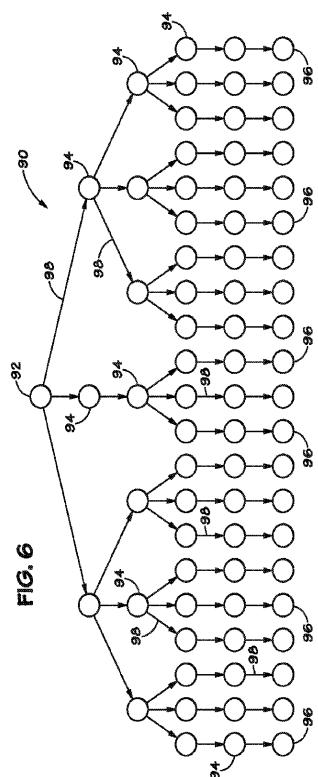
【図4】



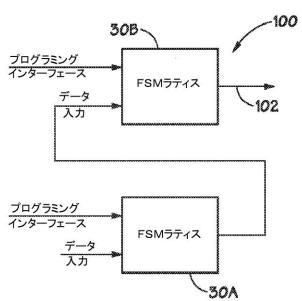
【図5】



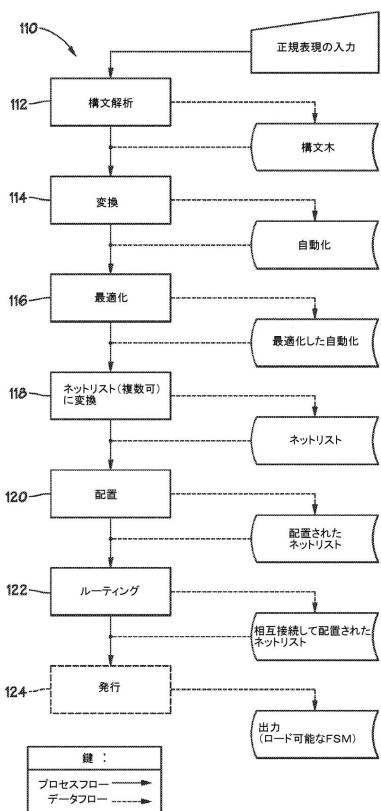
【図6】



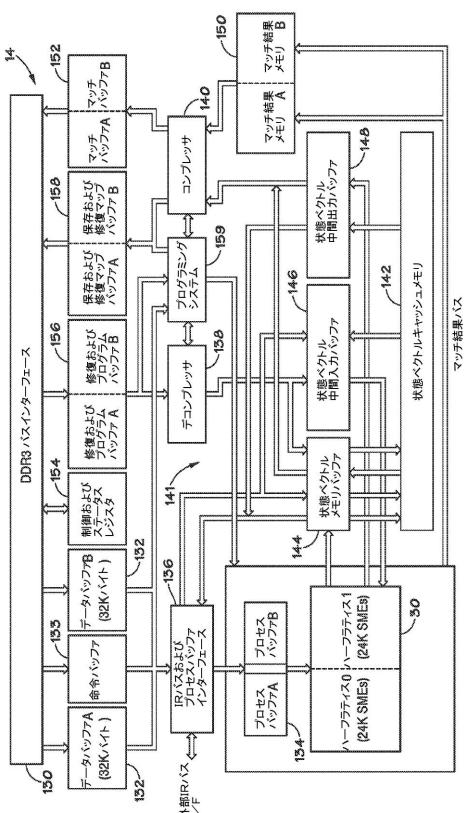
【図7】



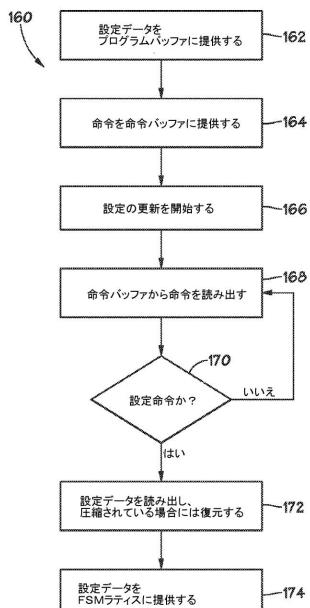
【図8】



【図9】



【図10】



フロントページの続き

(72)発明者 ブラウン , デイビッド アール .

アメリカ合衆国 , テキサス州 75002 , ルーカス , マーチモント ドライブ 1806

審査官 漆原 孝治

(56)参考文献 米国特許出願公開第2011 / 0307433 (US, A1)

阿部 茂樹 , 故障の検出および回復が可能なFPGAアーキテクチャ , 電子情報通信学会技術研究報告 , 日本 , 社団法人電子情報通信学会 , 2000年 4月28日 , 第100巻 , 第30号 , pp.1-8

(58)調査した分野(Int.Cl. , DB名)

G 06 F 15 / 82

G 06 F 9 / 38

G 06 F 15 / 173