

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6017034号
(P6017034)

(45) 発行日 平成28年10月26日 (2016.10.26)

(24) 登録日 平成28年10月7日 (2016.10.7)

(51) Int. Cl.	F I
G06F 17/30 (2006.01)	G06F 17/30 415
G06F 7/00 (2006.01)	G06F 7/00
G06F 9/50 (2006.01)	G06F 9/46 465E

請求項の数 25 (全 33 頁)

(21) 出願番号	特願2015-523124 (P2015-523124)	(73) 特許権者	595168543
(86) (22) 出願日	平成25年7月9日 (2013.7.9)		マイクロン テクノロジー, インク,
(65) 公表番号	特表2015-531111 (P2015-531111A)		アメリカ合衆国, アイダホ州 83716
(43) 公表日	平成27年10月29日 (2015.10.29)		-9632, ボイズ, サウス フェデ
(86) 国際出願番号	PCT/US2013/049744		ラル ウェイ 8000
(87) 国際公開番号	W02014/014709	(74) 代理人	100074099
(87) 国際公開日	平成26年1月23日 (2014.1.23)		弁理士 大宮 義之
審査請求日	平成28年6月20日 (2016.6.20)	(74) 代理人	100106851
(31) 優先権主張番号	13/552,479		弁理士 野村 泰久
(32) 優先日	平成24年7月18日 (2012.7.18)	(72) 発明者	ブラウン, デイビッド アール,
(33) 優先権主張国	米国 (US)		アメリカ合衆国, テキサス州 75002
早期審査対象出願			, ルーカス, マーチモント ドライブ 1
			806
			最終頁に続く

(54) 【発明の名称】 状態機械エンジンが受信したデータを取り扱うための方法およびシステム

(57) 【特許請求の範囲】

【請求項 1】

データ分析システムであって、
 分析されるデータを受信するように構成されるデータバッファと、
 複数の構成可能素子を備え、各々の構成可能素子が、前記データの少なくとも一部分を
 分析し、前記分析の結果を出力するように構成される複数のメモリセルを備える、状態機
 械格子と、
 前記データを前記データバッファから受信し、前記データを前記状態機械格子に提供す
 るように構成される、前記データバッファが、データバーストを介してバッファインター
 フェースにデータを提供するように構成され、各データバーストは、複数の状態機械格子
 のそれぞれ用にデータの所定の一部を含むバッファインターフェースと、
 を備える、データ分析システム。

【請求項 2】

前記データバッファは、第1のデータバッファおよび第2のデータバッファを含み、前
 記第1および第2のデータバッファの各々が、データを受信し、データを前記バッファイ
 ンターフェースに提供するように構成される、請求項1に記載のシステム。

【請求項 3】

前記第1のデータバッファは、前記第2のデータバッファがデータを前記バッファイン
 ターフェースに提供している間にデータを受信するように構成され、前記第2のデータバ
 ッファは、前記第1のデータバッファがデータを前記バッファインターフェースに提供し

10

20

ている間にデータを受信するように構成される、請求項 2 に記載のシステム。

【請求項 4】

前記データバッファは、前記データをデータブロックとして受信するように構成され、各々のデータブロックは、複数の状態機械格子のうちの 1 つに対応する、請求項 1 に記載のシステム。

【請求項 5】

前記データバッファは、1 つ、2 つ、4 つ、または 8 つのデータブロックを受信するように構成される、請求項 4 に記載のシステム。

【請求項 6】

前記データバッファは、前記データをデータブロック群として受信するように構成され、各々のデータブロック群は、前記複数の状態機械格子のうちの各々に対するデータブロックを含む、請求項 1 に記載のシステム。

10

【請求項 7】

前記データバッファは、交互発生するパターンで、データブロック群の各々のデータブロックの所定の部分を、前記データブロック群の各々のデータブロックが記憶されるまで記憶するように構成され、前記交互発生するパターンは、各々のデータブロック間で交互に発生する、請求項 6 に記載のシステム。

【請求項 8】

各々のデータブロックの前記所定の部分は、8 バイト、16 バイト、32 バイト、または 64 バイトのデータを含む、請求項 7 に記載のシステム。

20

【請求項 9】

データの前記所定の部分は、1 バイト、2 バイト、4 バイト、または 8 バイトのデータを含む、請求項 1 に記載のシステム。

【請求項 10】

前記複数の状態機械格子の各々に結合される 1 つ以上の処理バッファを備え、各々の処理バッファは、そのそれぞれの状態機械格子に対応する各々のデータバーストからデータの前記所定の部分を判定するように構成される、請求項 1 に記載のシステム。

【請求項 11】

前記データバッファは、前記分析されるデータに対応する命令を受信するように構成される命令バッファを含む、請求項 1 に記載のシステム。

30

【請求項 12】

データ分析システムであって、

複数の状態機械エンジンであって、各々が複数のデータ分析素子を有する状態機械格子を備え、各々のデータ分析素子が、データの少なくとも一部分を分析し、前記分析の結果を出力するように構成される複数のメモリセルを備える、複数の状態機械エンジンと、

前記複数の状態機械エンジンの各々に結合され、前記データを受信し、前記データを前記複数の状態機械エンジンに提供するように構成される、バッファインターフェースと、

分析されるべき前記データを受信し、前記データを前記バッファインターフェースに提供するように構成され、データバッファは、データブロック群として前記データを受信するように構成され、データブロックの各群は、前記複数の状態機械エンジンのそれぞれ用のデータブロックを含み、前記データブロックの少なくとも 1 つは、バイトパディングと組み合わせられる有効データを含む、データバッファと、

40

を備える、データ分析システム。

【請求項 13】

前記データバッファは、命令バッファを含み、前記命令バッファが、バイトパディングを含むデータブロックの部分を特定する命令を受信するように構成される、請求項 12 に記載のシステム。

【請求項 14】

前記データバッファは、複数の命令群を受信するように構成され、各々の命令群が、データブロック群に関連するデータを含む、請求項 12 に記載のシステム。

50

【請求項 1 5】

各々のデータブロック群は、複数の領域を含み、各々の命令群が、前記複数の領域の各々に関連するデータを含む、請求項 1 4 に記載のシステム。

【請求項 1 6】

前記複数の領域の各々に関連する前記データは、前記それぞれの領域中の総バイト数と、前記それぞれの領域中の前記データブロックの各々に対する総有効データバイト数とを含む、請求項 1 5 に記載のシステム。

【請求項 1 7】

前記複数の状態機械エンジンの各々は、前記総バイト数と前記総有効データバイト数との間の差に対応するデータを無視するように構成される、請求項 1 6 に記載のシステム。

10

【請求項 1 8】

前記複数の状態機械エンジンは、1 つ、2 つ、4 つ、または 8 つの状態機械エンジンを含む、請求項 1 2 に記載のシステム。

【請求項 1 9】

前記データを解析するように構成され、前記複数の状態機械エンジンを含む、単一の論理群を含む、請求項 1 2 に記載のシステム。

【請求項 2 0】

前記データを分析するように構成された複数の論理群を含み、前記複数の論理群の各論理群は、前記複数の状態機械エンジンの少なくとも 1 つを含む、請求項 1 2 に記載のシステム。

20

【請求項 2 1】

前記複数の論理群の各論理群は、前記データを受信し、前記データの所定の一部を分析するように構成されている、請求項 2 0 に記載のシステム。

【請求項 2 2】

前記複数の論理群の各論理群は、前記データを受信し、前記複数の論理群の他の論理群と並列して前記データを分析するように構成され、各論理群によって分析される前記データは同一である、請求項 2 0 に記載のシステム。

【請求項 2 3】

前記複数の状態機械エンジンの各状態機械エンジンは、各状態機械エンジンに割り当てられたデータを分析するように構成される、請求項 1 2 に記載のシステム。

30

【請求項 2 4】

前記複数の状態機械エンジンの各状態機械エンジンは、前記バッファインターフェースに提供されるデータバーストの所定の一部をラッチするように構成される、請求項 1 2 に記載のシステム。

【請求項 2 5】

前記データバッファは、オフセットして前記データを格納し、オフセットなしで、前記バッファインターフェースに前記データを提供するように構成される、請求項 1 2 に記載のシステム。

【発明の詳細な説明】**【技術分野】**

40

【0 0 0 1】

本発明の実施形態は、一般的には、電子デバイスに関し、より具体的には、ある実施形態では、データ分析のための並列デバイスを持つ電子デバイスに関する。

【背景技術】**【0 0 0 2】****[関連技術の説明]**

複雑なデータ分析（例えば、パターン認識）は、従来のフォンノイマン式コンピュータ上で実施するには非効率的でありうる。生物の頭脳、特に人間の頭脳は、しかしながら、複雑なデータ分析は得意である。現在の研究は、人間の頭脳は、新皮質中の一連の階層的に組織化されたニューロン層を用いてデータ分析を実施することを示唆している。階層的

50

より低い層中のニューロンは、例えば感覚器官からの「生の信号」を分析し、より高い層中のニューロンは、より低い水準のニューロンからの信号を分析する。新皮質中のこの階層的システムは、おそらくは頭脳の他のエリアと組み合わされて複雑なデータ分析を遂行し、この分析が、人間が、空間的推論、意識的思考、および複雑な言語などの高い水準の機能を実施することを可能とする。

【 0 0 0 3 】

コンピューティングの分野では、パターン認識タスクは、例えば、ますます意欲をそそるようになっている。さらにより多量のデータがコンピュータ間で伝達され、ユーザが検出することを希望するパターンの数は増している。例えば、スパムまたはマルウェアが、データストリーム、例えば特定のフレーズまたはコードの一部の中のパターンを検索することによって、しばしば検出される。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が、それは、新しい変形を検索するために、新しいパターンが実装され得るからである。これらのパターンの各々を求めてデータストリームを検索することによって、コンピューティング上のボトルネックが形成されかねない。しばしば、データストリームが受信されると、それが、一つずつ、各々のパターンを求めて検索される。システムがデータストリームの次の部分を検索する準備がなるまでの遅延は、パターンの数と共に増加する。したがって、パターン認識は、データの受信を遅速化する。

【 0 0 0 4 】

ハードウェアは、パターンを求めてデータストリームを検索するように設計されてきたが、このハードウェアは、しばしば、所与の時間内で適切な量のデータを処理することが不可能である。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間に分散させることによってそのように実行する。これらの回路は各々が、データストリームがパターンの一部分に適合するかどうかを判定する。しばしば、多数の回路が並列に動作して、各々が、ほぼ同時にデータストリームを検索する。しかしながら、生物の頭脳により匹敵する様式で複雑なデータ分析を実施することを効果的に可能とするシステムは存在しなかった。このようなシステムの開発が望ましい。

【 図面の簡単な説明 】

【 0 0 0 5 】

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

【 図 2 】 本発明の様々な実施形態に関わる、図 1 の状態機械エンジンの有限状態機械 (F S M) 格子の例を示す。

【 図 3 】 本発明の様々な実施形態に関わる、図 2 の F S M 格子のブロックの例を示す。

【 図 4 】 本発明の様々な実施形態に関わる、図 3 のブロックの行の例を示す。

【 図 5 】 本発明の様々な実施形態に関わる、図 4 の行のうちの 2 つずつの群の例を示す。

【 図 6 】 本発明の様々な実施形態に関わる、有限状態機械グラフの例を示す。

【 図 7 】 本発明の様々な実施形態に関わる、 F S M 格子で実装される 2 水準階層の例を示す。

【 図 8 】 本発明の様々な実施形態に関わる、コンパイラがソースコードを図 2 の F S M 格子のプログラミング用の二進法ファイルに変換するための方法の例を示す。

【 図 9 】 本発明の様々な実施形態に関わる、状態機械エンジンを示す。

【 図 1 0 】 本発明の様々な実施形態に関わる、デバイスのランクに配列された複数の物理的状态機械エンジンの例を示す。

【 図 1 1 】 本発明の様々な実施形態に関わる、状態機械エンジンに提供されるようにデータブロックに群分けされたデータセグメントの例を示す。

【 図 1 2 】 本発明の様々な実施形態に関わる、図 1 1 のデータブロックのデータセグメント間に挿入されるデータパディングの例を示す。

【 図 1 3 】 本発明の様々な実施形態に関わる、図 1 2 のデータブロックのデータセグメントの後に挿入されるデータパディングの例を示す。

【 図 1 4 】 本発明の様々な実施形態に関わる、状態機械エンジンのデータバッファシステ

10

20

30

40

50

ムに伝達されるように組織化された図 1 3 のデータブロックの例を示す。

【図 1 5】本発明の様々な実施形態に関わる、状態機械エンジンによって受信されるデータブロックの例を示す。

【図 1 6】本発明の様々な実施形態に関わる、状態機械エンジンのデータバッファシステムに記憶される図 1 5 のデータブロックの例を示す。

【図 1 7】本発明の様々な実施形態に関わる、複数の F S M 格子にデータバッファシステムから提供されているデータの例を示す。

【図 1 8】本発明の様々な実施形態に関わる、複数の論理群中に提供されているデータの例を示す。

【発明を実施するための形態】

10

【0006】

ここで図面に目を転じると、図 1 は、一般に参照番号 1 0 で示されるプロセッサベースのシステムの実施形態を示す。システム 1 0 (例えば、データ分析システム)は、デスクトップコンピュータ、ラップトップコンピュータ、ポケベル、携帯電話、自己管理手帳、携帯オーディオプレイヤー、制御回路、カメラなどの様々なタイプのうちの任意のものであり得る。システム 1 0 はまた、ルーター、サーバ、またはクライアント(例えば、既に述べた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であり得る。システム 1 0 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もしくは記録システム、ケーブルボックス、パーソナルデジタルメディアプレイヤー、ファクトリーオートメーションシステム、自動車コンピュータシステム、または医療デ

20

【0007】

システム 1 0 などの一般的なプロセッサベースのデバイスにおいては、マイクロプロセッサなどのプロセッサ 1 2 は、システム 1 0 中での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 1 2 は、システム制御を共有する複数のプロセッサを備え得る。プロセッサ 1 2 は、システム 1 0 中の素子の各々に直接にまたは間接に結合され得るため、プロセッサ 1 2 は、システム 1 0 の内部またはシステム 1 0 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 1 0 を制御する。

30

【0008】

本明細書に説明する実施形態によれば、システム 1 0 は、プロセッサ 1 2 の制御下で動作し得る状態機械エンジン 1 4 を含む。本明細書で用いられる状態機械エンジン 1 4 は、単一のデバイス(例えば、単一のチップ)のことである。状態機械エンジン 1 4 は、オートマトン理論を用い得る。例えば、状態機械エンジン 1 4 は、これらに限られないが、ミリーアーキテクチャ、ムーアアーキテクチャ、有限状態機械(F S M)、決定性 F S M(D F S M)、ビットパラレル状態機械(B P S M)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とはいえ、説明目的のため、応用分野は F S M のことである。しかしながら、当業者は、説明する技法は、様々な状態機械アーキテクチャのうちの任意の 1 つを用いて採用され得ることを理解するであろう。

40

【0009】

以下にさらに説明するように、状態機械エンジン 1 4 は、いくつかの(例えば、1 以上の)有限状態機械(F S M)格子(例えば、チップのコア)を含み得る。この応用分野の目的上、「格子」という用語は、素子(例えば、ブールセル、カウンタセル、状態機械素子、状態遷移素子)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ックス、ルーティングネットワーク、フレーム)のことである。さらに、「格子」は、任意の適切な形状、構造、または階層的組織(例えば、グリッド、キューブ、球、カスケード)を有し得る。各々の F S M 格子は、各々が同じデータを並列に受信して分析する複数の F S

50

Mを実装し得る。さらに、F S M格子は、群（例えば、クラスタ）に配列され得て、それにより、F S M格子のクラスタは、同じ入力データを並列に分析し得る。さらに、状態機械エンジン14のF S M格子のクラスタは、階層構造のより低い水準の状態機械格子からの出力はより高い水準の状態機械格子に対する入力として用いられ得る階層構造に配列され得る。状態機械エンジン14の並列F S M格子のクラスタを、階層構造を介して直列にカスケードリングすることによって、ますます複雑なパターンが分析され得る（例えば、評価され得る、探索され得るなど）。

【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、高処理速度を利用するシステムにおいて複雑なデータ分析（例えば、パターン認識）のために用いられ得る。例えば、本明細書に説明する実施形態は、1ギガバイト/秒の処理速度を持つ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他の外部デバイスからのデータは迅速に分析され得る。状態機械エンジン14は、ほぼ同時に、例えば、単一のデバイスサイクル中で、データストリームをいくつかの基準（例えば、検索項目）に従って分析し得る。状態機械エンジン14のある水準上のF S Mのクラスタ内のF S M格子の各々は、各々、ほぼ同時にデータストリームから同じ検索項目を受信し、並列F S M格子の各々は、その項目が状態機械エンジン14を処理基準中の次の状態に前進させるかどうかを判定し得る。状態機械エンジン14は、比較的多数の基準、例えば、100を超える、110を超える、または10,000を超える基準に従って項目を分析し得る。それらは並列に動作するため、それらは、基準を比較的高い帯域幅を有するデータストリーム、例えば、1ギガバイト/秒を超えるまたはほぼ等しいデータストリームに、そのデータストリームを遅速化することなく、適用し得る。

【0011】

1つの実施形態では、状態機械エンジン14は、データストリーム中の多数のパターンを認識する（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、ユーザまたは他のエンティティが分析することを希望する様々なタイプのデータストリームのうちの1つ以上の中の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た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た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たデータのストリームを分析するように構成され得る。1つの実施形態では、状態機械エンジン14は、スパムまたはマルウェアに対しデータストリームを分析するように構成され得る。データストリームは、データが、時間的、語彙的、または意味論的に意味のある順序などの意味を有する順序で受信される、直列のデータストリームとして受信され得る。代替的には、データストリームは、並列にまたはバラバラの順序で受信され、次に、例えば、インターネットを介して受信されたパケットを再順序付けすることによって、直列のデータストリームに変換され得る。一部の実施形態では、データストリームは、項目を直列に表すが、項目の各々を表現するビットは並列に受信され得る。データストリームは、システム10の外部のソースから受信される、または、メモリ16などのメモリデバイスに質問して、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。他の例では、状態機械エンジン14は、ある語を綴る一連の文字、遺伝子を指定する一連の遺伝子塩基対、イメージの一部分を形成する写真もしくは動画ファイル内の一連のビット、プログラムの一部を形成する実行可能ファイル中の一連のビット、または歌もしくは話されたフレーズの一部分を形成するオーディオファイル中の一連のビットを認識するように構成され得る。分析されるデータのストリームは、二進法形式または他の形式、例えば、十進法、ASCIIなどの複数のデータビットを含み得る。このストリームは、一桁または複数桁、例えば、いくつかの二進法桁の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を含み得る。メモリ16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（SR

10

20

30

40

50

AM)、同期式DRAM(SDRAM)、ダブルデータレートDRAM(DDR SDRAM)、DDR2 SDRAM、DDR3 SDRAMなどの揮発性メモリを含み得る。メモリ16はまた、リードオンリーメモリ(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、多結晶浮遊ゲートベースメモリ、および/または揮発性メモリと共に用いられる様々なアーキテクチャ(例えば、NANDメモリ、NORメモリなど)の他のタイプのフラッシュメモリなどの不揮発性メモリを含み得る。メモリ16は、状態機械エンジン14によって分析されるデータを提供し得る、DRAMデバイスなどの1つ以上のメモリデバイスを含み得る。本明細書では、「提供する」という用語は、一般的に、方向付ける、入力する、挿入する、送出する、転送する、送信する、生成する、与える、出力する、置く、記述するなどのことであり得る。このようなデバイスは、ソリッドステートドライバ(SSD)、MultimediaMediaCards(MMC)、SecureDigital(SD)カード、CompactFlash(CF)カード、または任意の他の適切なデバイスと呼ばれ得るまたはこれらを含み得る。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)、周辺構成要素相互接続(PCI)、PCI Express(PCI-E)、小型コンピュータシステムインターフェース(SCSI)、IEEE1394(Firewire)、または任意の他の適切なインターフェースなどの任意の適切なインターフェースを介してシステム10に結合し得ることを理解すべきである。フラッシュメモリデバイスなどのメモリ16の動作を容易化するために、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したデバイスであり得るか、または、それはプロセッサ12と統合され得る。加えて、システム10は、磁気記憶デバイスなどの外部記憶装置18を含み得る。外部記憶装置もまた、入力データを状態機械エンジン14に提供し得る。

10

20

【0013】

システム10は、いくつかのさらなる素子を含み得る。例えば、コンパイラ20は、図8に関連してより詳細に説明するように、状態機械エンジン14を構成する(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、ユーザがデータをシステム10に入力することを可能とするために、プロセッサ12に結合され得る。例えば、入力デバイス22は、状態機械エンジン14によって後で分析されるように、データをメモリ16に入力するために用いられ得る。入力デバイス22は、例えば、ボタン、切り替え素子、キーボード、ライトペン、スタイラスペン、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に結合され得る。ディスプレイ24は、例えば、LCD、CRT、LED、および/またはオーディオ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の用途に応じて、多くの他の構成要素を含み得る。

30

【0014】

図2~5は、FSM格子30の例を示す。ある例では、FSM格子30は、ブロック32のアレイを備える。説明されるように、各々のブロック32は、FSM中の複数の状態に対応する複数の選択的に結合可能なハードウェア素子(例えば、構成可能な素子および/または特殊目的の素子)を含み得る。FSM中での状態に類似して、ハードウェア素子は、入力ストリームを分析して、この入力ストリームに基づいて下流のハードウェア素子を起動することが可能である。

40

【0015】

構成可能素子は、多くの異なる機能を実装するように構成する(例えば、プログラムする)ことが可能である。例えば、構成可能素子は、行38(図3および4に示す)およびブロック32(図2および3に示す)に階層的に組織化される状態機械素子(SME)34、36(図5に示す)を含み得る。SMEはまた、状態遷移素子(STE)と考えられ

50

得る。階層的に組織化されたSME34、36の間で信号をルーティングするには、ブロック間切り替え素子40(図2および3に示す)、ブロック内切り替え素子42(図3および4に示す)、行内切り替え素子44(図4に示す)を含む構成可能切り替え素子の階層を用いればよい。

【0016】

以下に説明するように、切り替え素子は、ルーティング用の構造およびバッファを含み得る。SME34、36は、FSM格子30によって実装されたFSMの状態に対応する。SME34、36は、以下に説明するように、構成可能な切り替え素子を用いることによって一緒に結合することが可能である。したがって、FSMは、SME34、36を状態の機能に対応するように構成し、SME34、36をFSM中の状態の間での遷移に対応するように選択的に結合することによって、FSM格子30上に実装することが可能である。

10

【0017】

図2は、FSM格子30の例の全体図を示す。FSM格子30は、構成可能なブロック間切り替え素子40と選択的に一緒に結合することが可能な複数のブロック32を含む。ブロック間切り替え素子40は、導線46(例えば、ワイヤ、トレースなど)と、バッファ48および50とを含み得る。ある例では、バッファ48および50は、ブロック間切り替え素子40へのノからの信号の接続およびタイミングを制御するために含まれる。以下にさらに説明するように、バッファ48はブロック32間で送出されているデータをバッファリングするために提供され得るが、バッファ50はブロック間切り替え素子40間で送出されてい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)を受信して、そのデータをブロック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)に選択的に結合することが可能である。ブロック32はまた、ブロック32からの信号を外部デバイス(例えば、別のFSM格子30)に提供するための出力ブロック54(例えば、出力ポート)に選択的に結合することが可能である。FSM格子30はまた、(例えば、イメージ、プログラムを介して)FSM格子30を構成するために、プログラミングインターフェース56を含むことが可能である。イメージは、SME34、36の状態を構成する(例えば、設定する)ことが可能である。すなわち、イメージは、入力ブロック52での所与の入力に対してある仕方で反応するようにSME34、36を構成することが可能である。例えば、SME34、36は、文字「a」が入力ブロック52で受信されたときに高信号を出力するように設定することが可能である。

20

30

【0018】

ある例では、入力ブロック52、出力ブロック54、およびノまたはプログラミングインターフェース56は、レジスタに対する書き込みまたはそこからの読み出しがそれぞれの素子へのまたはそこからのデータを提供するように、レジスタとして実装され得る。したがって、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ているイメージからのビットを、SME34、36上にロードすることが可能である。図2はブロック32、入力ブロック52、出力ブロック54、およびブロック間切り替え素子40の間のある数の導線(例えば、ワイヤ、トレース)を示しているが、他の例では、より少ないまたはより多い導線が用いられ得ることを理解すべきである。

40

【0019】

図3は、ブロック32の例を示す。ブロック32は、構成可能なブロック内切り替え素子42と選択的に一緒に結合することが可能な複数の行38を含むことが可能である。加えて、行38は、ブロック間切り替え素子40で、別のブロック32内の別の行38に選択的に結合することが可能である。行38は、本明細書では2つずつの群(GOT)60と呼ばれる素子の対に組織化される複数のSME34、36を含む。ある例では、ブロック32は、16(16)の行38を備える。

【0020】

図4は、行38の例を示す。GOT60は、構成可能な行内切り替え素子44によって

50

、行 3 8 内の他の G O T 6 0 および任意の他の素子（例えば、特殊目的素子 5 8）に選択的に結合することが可能である。G O T 6 0 はまた、ブロック内切り替え素子 4 2 によって他の行 3 8 中の他の G O T 6 0 に対して、または、ブロック間切り替え素子 4 0 によって他のブロック 3 2 中の他の G O T 6 0 に対して結合することが可能である。ある例では、G O T 6 0 は、第 1 および第 2 の入力部 6 2、6 4 ならびに出力部 6 6 を有する。図 5 を参照してさらに例示されるように、第 1 の入力部 6 2 は G O T 6 0 の第 1 の S M E 3 4 に結合され、第 2 の入力部 6 4 は G O T 6 0 の第 2 の S M E 3 6 に結合される。

【 0 0 2 1 】

ある例では、行 3 8 は、第 1 および第 2 の複数の行相互接続導線 6 8、7 0 を含む。ある例では、G O T 6 0 の入力部 6 2、6 4 は 1 つ以上の行相互接続導線 6 8、7 0 に結合することが可能であり、出力部 6 6 は 1 つ以上の行相互接続導線 6 8、7 0 に結合することが可能である。ある例では、第 1 の複数の行相互接続導線 6 8 は、行 3 8 内の各々の G O T 6 0 の各々の S M E 3 4、3 6 に結合することが可能である。第 2 の複数の行相互接続導線 7 0 は、行 3 8 内の各々の G O T 6 0 の各々のたった 1 つの S M E 3 4、3 6 に結合することが可能であるが、G O T 6 0 のその他の S M E 3 4、3 6 には結合することは不可能である。ある例では、図 5 に関連してより良好に示されるように、第 2 の複数の行相互接続導線 7 0 のうちの第 1 の半分は、行 3 8 内の S M E 3 4、3 6 のうちの第 1 の半分（各々の G O T 6 0 から 1 つの S M E 3 4）に結合することが可能であり、第 2 の複数の行相互接続導線 7 0 のうちの第 2 の半分は、行 3 8 内の S M E 3 4、3 6 のうちの第 2 の半分（各々の G O T 6 0 からその他の S M E 3 4、3 6）に結合することが可能である。第 2 の複数の行相互接続導線 7 0 と S M E 3 4、3 6 との間の制限された接続は、本明細書では「パリティ」と呼ばれる。ある例では、行 3 8 はまた、カウンタなどの特殊目的素子 5 8、構成可能ブール論理素子、ルックアップテーブル、R A M、フィールド構成可能ゲートアレイ（F P G A）、特定用途向け集積回路（A S I C）、構成可能プロセッサ（例えば、マイクロプロセッサ）、または、特殊目的機能を実施するための他の素子などの特殊目的素子 5 8 を含むことが可能である。

【 0 0 2 2 】

ある例では、特殊目的素子 5 8 は、カウンタ（本明細書ではカウンタ 5 8 とも呼ばれる）を備える。ある例では、カウンタ 5 8 は、1 2 ビットの構成可能ダウンカウンタを備える。1 2 ビット構成可能カウンタ 5 8 は、カウント入力部、リセット入力部、およびゼロカウント出力部を有する。カウント入力部は、アサートされるとき、カウンタ 5 8 の値を 1 だけデクレメントする。リセット入力部は、アサートされるとき、カウンタ 5 8 に、関連付けられたレジスタから初期値をロードさせる。1 2 ビットカウンタ 5 8 の場合、最大で 1 2 ビットの数を、初期値としてロードすることが可能である。カウンタ 5 8 の値がゼロ（0）にデクレメントされるとき、ゼロカウント出力がアサートされる。カウンタ 5 8 はまた、パルスモードとホールドモードとの少なくとも 2 つのモードを有する。カウンタ 5 8 がパルスモードに設定されるとき、ゼロカウント出力は、カウンタ 5 8 がゼロに到達して、クロックが循環するときにアサートされる。このゼロカウント出力は、カウンタ 5 8 の次のクロックサイクルの間にアサートされる。その結果、カウンタ 5 8 は、クロックサイクルから時間的に間に合ってオフセットされる。次のクロックサイクルで、ゼロカウント出力はもはやアサートされない。カウンタ 5 8 がホールドモードに設定されるとき、ゼロカウント出力は、カウンタ 5 8 がゼロにデクレメントするときにクロックサイクル間にアサートされ、カウンタ 5 8 がアサートされているリセット入力によってリセットされるまでアサートされたまま留まる。

【 0 0 2 3 】

別の例では、特殊目的素子 5 8 はブール論理を備える。例えば、ブール論理は、A N D、O R、N A N D、N O R、積和（S o P）、積和のネグート出力（N S o P）、和積のネグート出力（N P o S）、および和積（P o S）の関数などの論理関数を実施するために用いられ得る。このブール論理は、F S M 格子 3 0 中の（以下に後述するように、F S M のターミナルノードに対応する）ターミナル状態の S M E からデータを抽出するために

10

20

30

40

50

用いることが可能である。抽出されたデータは、状態データを他のFSM格子30に提供
するおよび/またはFSM格子30を再構成するために用いられる構成用データを提供す
る、または別のFSM格子30を再構成するために用いることが可能である。

【0024】

図5はGOT60の例を示す。GOT60は、入力部62、64を有し、かつそれらの
出力部72、74がORゲート76および3対1マルチプレクサ78に結合される第1の
SME34および第2のSME36を含む。3対1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に結合させる
ように設定することが可能である。ORゲート76は、GOT60の共有出力66を形成
するように双方の出力部72、74と一緒に結合するために用いることが可能である。あ
る例では、第1および第2のSME34、36は、上述したようにパリティを示すが、こ
の場合、第1のSME34の入力部62は行相互接続導線68のうちの一部に結合するこ
とが可能であり、第2のSME36の入力部64は他の行相互接続導線70に結合するこ
とが可能であり、パリティ問題を克服し得る共通の出力66が、生成され得る。ある例で
は、GOT60内の2つのSME34、36は、切り替え素子79のどちらかまたは双方
を設定することによって、カスケードするおよび/またはそれら自身に対してループバ
ックすることが可能である。SME34、36は、SME34、36の出力部72、74を
他方のSME34、36の入力部62、64に結合することによってカスケードすること
が可能である。SME34、36は、出力部72、74をそれら自身の入力部62、64
に結合することによって、それら自身に対してループバックすることが可能である。し
たがって、第1のSME34の出力部72は、第1のSME34の入力部62と第2のSME
36の入力部64の、どちらにも結合することが不可能であるか、一方または双方に結
合することが可能であるかである。

【0025】

ある例では、状態機械素子34、36は、検出ライン82に並列に結合された、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の複数のメモ
リセル80を備える。このような1つのメモリセル80は、高いまたは低い値(例えば、
1または0)に対応するものなどのデータ状態に設定することが可能なメモリセルを備え
る。メモリセル80の出力部は、検出ライン82に結合され、メモリセル80への入力部
は、データストリームライン84上のデータに基づく信号を受信する。ある例では、入力
ブロック52での入力は、メモリセル80の内の1つ以上を選択するために復号化される
。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、検出ライン82上に出力
として提供する。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
)に提供することが可能であり、デコーダはデータストリームライン84のうちの1つ以
上を選択することが可能である。ある例では、デコーダは、8ビットのASCII文字を
256のデータストリームライン84のうちの対応する1つに変換することが可能である
。

【0026】

メモリセル80は、したがって、メモリセル80が高い値に設定され、データストリー
ムライン84上のデータがメモリセル80を選択するときに、高信号を検出ライン82に
出力する。データストリームライン84上のデータがメモリセル80を選択し、メモリセ
ル80が低い値に設定されるときに、メモリセル80は、低い信号を検出ライン82に出
力する。検出ライン82上のメモリセル80からの出力は、検出セル86によって感知さ
れる。

【0027】

ある例では、入力ライン62、64上の信号は、それぞれの検出セル86をアクティブ
状態または非アクティブ状態に設定する。非アクティブ状態に設定されたとき、検出セル
86は、それぞれの検出ライン82上の信号とは無関係に、それぞれの出力部72、74
上に低い信号を出力する。アクティブ状態に設定されたとき、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つから高い信号が検出されるときに、高

10

20

30

40

50

い信号をそれぞれの出力ライン 72、74 上に出力する。アクティブ状態にあるとき、検出セル 86 は、それぞれの S M E 34、36 のメモリセル 82 の全てからの信号が低いときには、それぞれの出力ライン 72、74 上に低い信号を出力する。

【0028】

ある例では、S M E 34、36 は 256 のメモリセル 80 を含み、各々のメモリセル 80 は異なるデータストリームライン 84 に結合される。したがって、S M E 34、36 は、データストリームライン 84 のうちの選択された 1 つ以上が高い信号をその上に有するときに高い信号を出力するようにプログラムすることが可能である。例えば、S M E 34 は、第 1 のメモリセル 80 (例えば、ビット 0) を高く設定され、全ての他のメモリセル 80 (例えば、ビット 1 ~ 255) を低く設定されることが可能である。それぞれの検出セル 86 がアクティブ状態にあるとき、S M E 34 は、ビット 0 に対応するデータストリームライン 84 がその上に高い信号を有するときに、出力部 72 に高い信号を出力する。他の例では、S M E 34 は、適切なメモリセル 80 を高い値に設定することによって、複数のデータストリームライン 84 のうちの 1 つがその上に高い信号を有するときに、高い信号を出力するように設定することが可能である。

【0029】

ある例では、メモリセル 80 は、関連付けられたレジスタからビットを読み出すことによって高いまたは低い値に設定することが可能である。したがって、S M E 34 は、コンパイラ 20 によって作成されたイメージをレジスタに記憶して、レジスタ中のビットを関連付けられたメモリセル 80 中にロードすることによって構成することが可能である。ある例では、コンパイラ 20 によって作成されたイメージは、高いまたは低い (例えば、1 または 0 の) ビットの二値イメージを含む。このイメージは、S M E 34、36 をカスケードすることによって、F S M を実装するように F S M 格子 30 を構成することが可能である。例えば、第 1 の S M E 34 は、検出セル 86 をアクティブ状態に設定することによってアクティブ状態に設定することが可能である。第 1 の S M E 34 は、ビット 0 に対応するデータストリームライン 84 が高い信号をその上に有するときに高い信号を出力するように設定することが可能である。第 2 の S M E 36 は非アクティブ状態に初期設定することが可能であるが、アクティブであるとき、ビット 1 に対応するデータストリームライン 84 が高い信号をその上に有するときに高い信号を出力するように設定することが可能である。第 1 の S M E 34 および第 2 の S M E 36 は、第 1 の S M E 34 の出力部 72 を第 2 の S M E 36 の入力部 64 に結合するように設定することによって、カスケードすることが可能である。したがって、ビット 0 に対応するデータストリームライン 84 上で高い信号が感知されたとき、第 1 の S M E 34 は出力部 72 に高い信号を出力して、第 2 の S M E 36 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ライン 84 上で高い信号が感知されたとき、第 2 の S M E 36 は、別の S M E 36 を起動するためにまたは F S M 格子 30 から出力されるように、出力部 74 に高い信号を出力する。

【0030】

ある例では、単一の F S M 格子 30 が単一の物理的デバイス上に実装されるが、しかしながら、他の例では、2 つ以上の F S M 格子 30 を、単一の物理的デバイス (例えば、物理的なチップ) 上に実装することが可能である。ある例では、各々の F S M 格子 30 は、区別可能なデータ入力ブロック 52、区別可能な出力ブロック 54、区別可能なプログラミングインターフェース 56、および構成可能素子の区別可能な集合を含むことが可能である。そのうえ、構成可能素子の各々の集合は、それらの対応するデータ入力ブロック 52 でデータに反応する (例えば、高いまたは低い信号を出力する) ことが可能である。例えば、第 1 の F S M 格子 30 に対応する構成可能素子の第 1 の集合は、第 1 の F S M 格子 30 に対応する第 1 のデータ入力ブロック 52 でデータに反応することが可能である。第 2 の F S M 格子 30 に対応する構成可能素子の第 2 の集合は、第 2 の F S M 格子 30 に対応する第 2 のデータ入力ブロック 52 に反応することが可能である。したがって、各々の F S M 格子 30 は構成可能素子の集合を含むが、その場合、異なる集合の構成可能素子は

異なる入力データに反応することが可能である。同様に、各々のFSM格子30と、構成可能素子の各々の対応する集合とは、区別可能な出力を提供することが可能である。一部の例では、第1のFSM格子30からの出力ブロック54を第2のFSM格子30の入力ブロック52に結合することが可能であり、それにより、第2のFSM格子30に対する入力データが、一連のFSM格子30の階層的配列中の第1のFSM格子30からの出力データを含むことが可能となる。

【0031】

ある例では、FSM格子30上にロードされるイメージは、構成可能素子、構成可能切り替え素子、およびFSM格子30内の特殊目的素子を構成するための複数ビットのデータを含む。ある例では、イメージは、ある入力に基づいて所望の出力を提供するようにFSM格子30を構成するために、FSM格子30上にロードすることが可能である。出力ブロック54は、データ入力ブロック52でのデータに対する構成可能素子の反応に基づいて、FSM格子30からの出力を提供することが可能である。出力ブロック54からの出力は、所与のパターンの適合を示す単一ビット、複数のパターンに対する適合および不適合を示す複数ビットを含む語、ならびに所与の瞬間での全てのもしくはある構成可能素子の状態に対応する状態ベクトルを含むことが可能である。説明したように、いくつかのFSM格子30は、パターン認識（例えば、音声認識、イメージ認識など）、信号処理、撮像、コンピュータビジョン、暗号法などのデータ分析を実施するために、状態機械エンジン14などの状態機械エンジン中に含まれ得る。

【0032】

図6は、FSM格子30によって実装することが可能な有限状態機械（FSM）の例としてのモデルを示す。FSM格子30は、FSMの物理的実装物として構成する（例えば、プログラムする）ことが可能である。FSMは、1つ以上のルートノード92を含むダイアグラム90（例えば、有向グラフ、無向グラフ、擬グラフ）として表されうる。ルートノード92に加えて、FSMは、1つ以上のエッジ98を介してルートノード92および他の標準ノード94に接続されたいくつかの標準ノード94およびターミナルノード96から作成することが可能である。ノード92、94、96はFSM中の状態に対応する。エッジ98は、状態の間の遷移に対応する。

【0033】

ノード92、94、96の各々は、アクティブ状態または非アクティブ状態にありうる。非アクティブ状態にあるとき、ノード92、94、96は入力データに反応する（例えば、応答する）ことはない。アクティブ状態にあるとき、ノード92、94、96は入力データに反応することが可能である。上流のノード92、94は、入力データが上流のノード92、94と下流のノード94、96との間のエッジ98によって指定された基準に適合するときに、ノードの下流にあるノード94、96を起動することによって、入力データに反応することが可能である。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たときに、エッジ98によって第1のノード94に接続されている第2のノード94を起動する。本明細書で用いられる「上流」とは1つ以上のノードの間の関係のことであり、この場合、1つ以上の他のノードの上流にある（または、ループもしくは帰還の構成の場合にはそれ自身の上流にある）第1のノードは、第1のノードがその1つ以上の他のノードを起動することが可能である（またはループの場合にはそれ自身を起動することが可能である）状況のことであり、同様に、「下流」とは1つ以上の他のノードの下流にある（または、ループの場合にはそれ自身の下流にある）第1のノードが、その1つ以上の他のノードによって起動することが可能である（または、ループの場合にはそれ自身によって起動することが可能である）関係のことであり、したがって、「上流」および「下流」という用語は、本明細書では、1つ以上のノードの間の関係のことであり、これらの用語は、ノード間でのループまたは他の非線形経路の使用を排除しない。

【0034】

ダイアグラム90では、ルートノード92は、最初に起動され得、入力データがルート

10

20

30

40

50

ノード 9 2 からのエッジ 9 8 と適合するときに、下流のノード 9 4 を起動することが可能である。ノード 9 4 は、入力データがルートノード 9 4 からのエッジ 9 8 と適合するときに、ノード 9 6 を起動することが可能である。ダイアグラム 9 0 全体にわたって、ノード 9 4、9 6 は、入力データが受信されると、この様式で起動することが可能である。ターミナルノード 9 6 は、入力データ中の目的とするシーケンスの適合に対応する。したがって、ターミナルノード 9 6 の起動は、目的と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している F S M 格子 3 0 の文脈では、ターミナルノード 9 6 への到達は、目的とする特定のパターンが入力データ中で検出されたことを示しうる。

す

10

【 0 0 3 5 】

ある例では、各々のルートノード 9 2、標準ノード 9 4、およびターミナルノード 9 6 は、F S M 格子 3 0 中の構成可能素子に対応しうる。各々のエッジ 9 8 は、構成可能素子の間の接続部に対応しうる。したがって、別の標準ノード 9 4 またはターミナルノード 9 6 に遷移する（例えば、これに接続するエッジ 9 8 を有する）標準ノード 9 4 は、別の構成可能素子に遷移する（例えば、これに対して出力を提供する）構成可能素子に対応する。一部の例では、ルートノード 9 2 は、対応する構成可能素子を有しない。

【 0 0 3 6 】

理解されるように、ノード 9 2 をルートノードとして説明し、ノード 9 6 をターミナルノードとして説明したが、必ずしも特定の「出発点」またはルートノードがなくてもよく、必ずしも特定の「終了点」または出力ノードがなくてもよい。言い換えれば、どのノードでも開始点であり得るし、またどのノードでも出力を提供し得る。

20

【 0 0 3 7 】

F S M 格子 3 0 がプログラムされたとき、構成可能素子の各々もまた、アクティブ状態または非アクティブ状態にありうる。所与の構成可能素子は、非アクティブであるとき、対応するデータ入力ブロック 5 2 での、入力データに反応しない。アクティブな構成可能素子はデータ入力ブロック 5 2 での入力データに反応することが可能であり、その入力データが構成可能素子の設定と適合するときに、下流の構成可能素子を起動することが可能である。ある構成可能素子がターミナルノード 9 6 に対応するとき、その構成可能素子は、外部デバイスとの適合の指示を提供するために、出力ブロック 5 4 に結合することが可能である。

30

【 0 0 3 8 】

プログラミングインターフェース 5 6 を介して F S M 格子 3 0 上にロードされたイメージは、構成可能素子および特殊目的素子ならびに構成可能素子と特殊目的素子との間の接続部を構成することが可能であり、それにより、データ入力ブロック 5 2 でのデータに対する反応に基づいたノードの一連の起動によって、所望の F S M が実装される。ある例では、構成可能素子は、1 つのデータサイクル（例えば、1 つの文字、文字の集合、1 つのクロックサイクル）にわたってアクティブのまま留まり、次に、上流の構成可能素子によって再起動されない限り、非アクティブになる。

【 0 0 3 9 】

40

ターミナルノード 9 6 は、過去の事象の圧縮された履歴を記憶するものと考えることが可能である。例えば、ターミナルノード 9 6 に達するために必要とされる入力データの 1 つ以上のパターンは、そのターミナルノード 9 6 の起動によって表すことが可能である。ある例では、ターミナルノード 9 6 によって提供される出力は二進法である、すなわち、その出力は、目的とするパターンが適合したかどうかを示す。ダイアグラム 9 0 中の標準ノード 9 4 に対するターミナルノード 9 6 の比はかなり小さい。言い換えれば、F S M 中には高度の複雑性があり得るとはいえ、F S M の出力は、比較すると小さくあり得る。

【 0 0 4 0 】

ある例では、F S M 格子 3 0 の出力は状態ベクトルを含みうる。状態ベクトルは、F S M 格子 3 0 の構成可能素子の状態（例えば、起動されているか起動されていないか）を含

50

む。別の例では、状態ベクトルは、構成可能素子がターミナルノード 96 に対応するかしないかとは無関係に、構成可能素子の全てまたは部分集合の状態を含むことが可能である。ある例では、状態ベクトルは、ターミナルノード 96 に対応する構成可能素子に対する状態を含む。したがって、出力は、ダイアグラム 90 の全てのターミナルノード 96 によって提供される指示の収集物を含むことが可能である。状態ベクトルは語として表すことが可能であるが、その場合、各々のターミナルノード 96 によって提供される二進法の指示は 1 ビットの語を含む。ターミナルノード 96 のこの符号化は、FSM 格子 30 に対する検出状態（例えば、目的とするシーケンスが、およびどのシーケンスが、検出されたか）の効果的な指示を提供することが可能である。

【0041】

10

上述したように、FSM 格子 30 は、パターン認識機能を実装するようにプログラムすることが可能である。例えば、FSM 格子 30 は、入力データ中の 1 つ以上のデータシーケンス（例えば、署名、パターン）を認識するように構成することが可能である。目的とするデータシーケンスが FSM 格子 30 によって認識されたとき、その認識の指示を、出力ブロック 54 に提供することが可能である。ある例では、パターン認識は、例えば、ネットワークデータ中のマルウェアまたは他のデータを特定するために、記号（例えば、ASCII 文字）のストリングを認識することが可能である。

【0042】

図 7 は、2 つの水準の FSM 格子 30 が直列に結合されて、データを分析するために用いられる階層構造 100 の例を示す。具体的には、図示する実施形態では、階層構造 100 は、直列に配列された第 1 の FSM 格子 30 A と第 2 の FSM 格子 30 B とを含む。各々の FSM 格子 30 は、データ入力を受信するそれぞれのデータ入力ブロック 52、構成用信号を受信するプログラミングインターフェースブロック 56、および出力ブロック 54 を含む。

20

【0043】

第 1 の FSM 格子 30 A は、入力データ、例えば生データをデータ入力ブロックで受信するように構成される。第 1 の FSM 格子 30 A は、上述したように入力データに反応して、出力データを出力ブロックのところで提供する。第 1 の FSM 格子 30 A からの出力は、第 2 の FSM 格子 30 B のデータ入力ブロックに送出される。第 2 の FSM 格子 30 B は、次に、第 1 の FSM 格子 30 A によって提供された出力に基づいて反応して、階層構造 100 の対応する出力信号 102 を提供することが可能である。2 つの FSM 格子 30 A および 30 B をこのように直列に階層的に結合することによって、過去の事象に関するデータを、第 1 の FSM 格子 30 A から第 2 の FSM 格子 30 B に対して圧縮した語で提供する手段が提供される。提供されたデータは、第 1 の FSM 格子 30 A によって記録された複雑な事象（例えば、目的とするシーケンス）の要約と効果的になりうる。

30

【0044】

図 7 に示す FSM 格子 30 A、30 B の 2 水準階層 100 は、2 つの独立したプログラムを、同じデータストリームに基づいて動作することを可能とする。この 2 段階階層は、異なる領域としてモデリングされた生物学的頭脳中での視認に類似しうる。このモデルでは、これらの領域は、各々が類似の計算機能（パターンマッチング）を実施するが、異なるプログラム（署名）を用いる効果的に異なるパターン認識エンジンである。複数の FSM 格子 30 A、30 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増加した知識が獲得され得る。

40

【0045】

（第 1 の FSM 格子 30 A によって実装される）階層の第 1 の水準は、例えば、生データストリームに対して直接的に処理を実施することが可能である。すなわち、生データストリームは、第 1 の FSM 格子 30 A の入力ブロック 52 で受信することが可能であり、第 1 の FSM 格子 30 A の構成可能素子は、この生データストリームに反応することが可能である。階層の（第 2 の FSM 格子 30 B によって実装される）第 2 の水準は、第 1 の水準からの出力を処理することが可能である。すなわち、第 2 の FSM 格子 30 B は、第

50

2のFSM格子30Bの入力ブロック52で第1のFSM格子30Aの出力ブロック54からの出力を受信し、第2のFSM格子30Bの構成可能素子は、第1のFSM格子30Aの出力に反応することが可能である。したがって、この例では、第2のFSM格子30Bは、生データストリームを入力として受信せず、むしろ、第1のFSM格子30Aによって判定された生データストリームと適合する目的とするパターンの指示を受信する。第2のFSM格子30Bは、第1のFSM格子30Aからの出力データストリーム中のパターンを認識するFSMを実装することが可能である。第2のFSM格子30Bは、FSM格子30Aからの出力を受信することに加えて、複数の他のFSM格子からの入力を受信し得ることを理解すべきである。同様に、第2のFSM格子30Bは他のデバイスからの入力を受信し得る。第2のFSM格子30Bは、出力を生成するために、これらの複数の入力を組み合わせ得る。

10

【0046】

図8は、コンパイラが、FSMを実装するために、格子30などのFSM格子を構成するために用いられるイメージにソースコードを変換するための方法110の例を示す。方法110は、ソースコードをシンタクスツリーに解析すること(ブロック112)と、シンタクスツリーをオートマトンに変換すること(ブロック114)と、オートマトンを最適化すること(ブロック116)と、オートマトンをネットリストに変換すること(ブロック118)と、ネットリストをハードウェア上に置くこと(ブロック120)と、ネットリストをルーティングすること(ブロック122)と、結果として得られるイメージを公開すること(ブロック124)とを含む。

20

【0047】

ある例では、コンパイラ20は、ソフトウェア開発者がFSM格子30上にFSMを実装するためにイメージを作成することを可能と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(API)を含む。コンパイラ20は、ソースコード中の正規表現の入力集合を、FSM格子30を構成するために構成されたイメージに変換する方法を提供する。コンパイラ20は、フォンノイマンアーキテクチャを有するコンピュータ用の命令によって実装することが可能である。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の機能を実装させることが可能である。例えば、これら命令は、プロセッサ12によって実行されるとき、プロセッサ12に、プロセッサ12からアクセス可能なソースコードに対して、ブロック112、114、116、118、120、122、および124中で説明したような動作を実施させることが可能である。

30

【0048】

ある例では、ソースコードは、記号の群内の記号のパターンを特定するための検索ストリングを記述する。検索ストリングを記述するには、ソースコードは、複数の正規表現(regex)を含めばよい。regexは、起動検索パターンを記述するためのストリングでありうる。regex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で広く用いられる。ある例では、コンパイラにサポートされる正規表現は、非構造化データの分析のために基準を含む。非構造化データは、自由形態で、データ内の語に適用される索引付けを有しないデータを含みうる。語は、データ内での、印刷可能でありか印刷不可能であるかは問わず、バイトの任意の組み合わせを含みうる。ある例では、コンパイラは、Perl、(例えば、Perlとコンパティブルな正規表現(PCRE))、PHP、Java、および.NET言語を含むregexを実装するための複数の異なるソースコード言語をサポートすることが可能である。

40

【0049】

ブロック112で、コンパイラ20は、ソースコードを解析して、異なるタイプの演算子がソースコードによって実装された異なる機能(例えば、ソースコード中のregexによって実装された異なる機能)に対応する、関係的に接続された演算子の配列を形成することが可能である。ソースコードを解析することで、ソースコードの一般的表現を作成することが可能である。ある例では、この一般的な表現は、シンタクスツリーとして知ら

50

れているツリーグラフという形態でソースコード中の `regex` の符号化された表現を含む。本明細書に説明する例は、他の例でのシンタクスツリー（「抽象シンタクスツリー」としても知られている）としての配列のことであるが、しかしながら、具象シンタクスツリーまたは他の配列を用いることが可能である。

【0050】

上述したように、コンパイラ 20 は複数のソースコード言語をサポートすることが可能であるため、解析することで、ソースコードは、言語とは無関係に、言語に固有ではない表現、例えばシンタクスツリーに変換される。したがって、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とは無関係に、共通の入力構造から作動することが可能である。

10

【0051】

上記のように、シンタクスツリーは、関係的に接続された複数の演算子を含む。シンタクスツリーは、複数の異なるタイプの演算子を含みうる。すなわち、異なる演算子は、ソースコード中の `regex` によって実装される異なる機能に対応しうる。

【0052】

ブロック 114 で、シンタクスツリー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であり、したがって、決定性または非決定性であると分類することが可能である。決定性オートマトンは、所与の時点において単一の実行経路を有し、非決定性オートマトンは、複数の同時実行経路を有する。オートマトンは、複数の状態を含む。シンタクスツリーをオートマトンに変換するために、シンタクスツリー中の演算子と、演算子間の関係とを、状態間の遷移を伴う状態に変換する。ある例では、オートマトンは、FSM 格子 30 のハードウェアに部分的に基づいて変換することが可能である。

20

【0053】

ある例では、オートマトンに対する入力記号は、アルファベット記号、数値 0 ~ 9、および他の印刷可能文字を含む。ある例では、入力記号は、バイト値 0 ~ 255（255 を含む）によって表される。ある例では、オートマトンは、グラフのノードが状態の集合に対応する有向グラフとして表すことが可能である。ある例では、入力記号に関する状態 p から状態 q への遷移、すなわち、 $(p, \text{入力記号})$ は、ノード p からノード q への有向接続によって示される。ある例では、オートマトンの逆は、ある記号に関する各々の遷移 $p \rightarrow q$ が、その記号に関する逆転された $q \rightarrow p$ になる新しいオートマトンを生成する。逆にすると、開始状態は最終状態になり、最終状態は開始状態になる。ある例では、オートマトンによって認識された（例えば、適合した）言葉は、オートマトンに連続的に入力されたときに最終状態に到達する全ての可能な文字ストリングの集合である。オートマトンによって認識された言葉の中の各々のストリングは、開始状態から 1 つ以上の最終状態に至る経路をたどる。

30

【0054】

ブロック 116 で、オートマトンは、構築された後に、とりわけ、その複雑さおよびサイズを減少させるように最適化される。オートマトンは、冗長状態を組み合わせることによって最適化することが可能である。

【0055】

40

ブロック 118 で、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。オートマトンはネットリストに変換することで、オートマトンの各々の状態が、FSM 格子 30 上のハードウェア素子（例えば、SME 34、36、他の素子）にマッピングされ、ハードウェア素子間の接続が決定される。

【0056】

ブロック 120 で、ネットリストは、ネットリストの各々のノードに対応する目標デバイス（例えば、SME 34、36、特殊目的素子 58）の特定のハードウェア素子を選択するように位置付けされる。ある例では、位置付けは、FSM 格子 30 用の一般的な入力および出力の制約に基づいて各々の特定のハードウェア素子を選択する。

【0057】

50

ブロック 1 2 2 で、位置付けされたネットリストは、選択されたハードウェア素子と一緒に結合して、ネットリストによって記述される接続を達成するために、構成可能切り替え素子（例えば、ブロック間切り替え素子 4 0、ブロック内切り替え素子 4 2、および行内切り替え素子 4 4）に対する設定を決定するようにルーティングされる。ある例では、構成可能切り替え素子に対する設定は、選択されたハードウェア素子を接続するために用いられる F S M 格子 3 0 の特定の導線と、構成可能切り替え素子に対する設定とを決定することによって決定される。ルーティングは、ブロック 1 2 0 のその位置付けでのハードウェア素子間の接続のより具体的な制約を考慮することが可能である。したがって、ルーティングは、F S M 格子 3 0 上の導線の実際の制約を考えて適切な接続を作成するためにグローバルな位置付けによって決定されたハードウェア素子の一部の位置を調整し得る。

10

【 0 0 5 8 】

いったんネットリストが位置付けされてルーティングされると、この位置付けされ、ルーティングされたネットリストは、F S M 格子 3 0 を構成するための複数のビットに変換される。これら複数のビットは、本明細書ではイメージ（例えば、二値イメージ）と呼ばれる。

【 0 0 5 9 】

ブロック 1 2 4 で、イメージはコンパイラ 2 0 によって公開される。イメージは、F S M 格子 3 0 の特定のハードウェア素子を構成するための複数のビットを含む。これらのビットは、ソースコードによって記述された機能性を有する F S M をプログラムされた F S M 格子 3 0 が実装するように、S M E 3 4、3 6、特殊目的素子 5 8、および構成可能切り替え素子の状態を構成するために、F S M 格子 3 0 上にロードすることが可能である。位置付け（ブロック 1 2 0）およびルーティング（ブロック 1 2 2）は、F S M 格子 3 0 中の特定の位置にある特定のハードウェア素子をオートマトン中の特定の状態にマッピングすることが可能である。したがって、イメージ中のビットは、所望の機能（複数可）を実装するために、特定のハードウェア素子を構成することが可能である。ある例では、イメージは、機械コードをコンピュータ読み取り可能媒体に保存することによって公開することが可能である。別の例では、イメージは、イメージを表示デバイス上に表示することによって公開することが可能である。さらに別の例では、イメージは、イメージを F S M 格子 3 0 中にロードするための構成用デバイスなどの別のデバイスに送出することによって公開することが可能である。さらに別の例では、イメージは、イメージを F S M 格子（例えば、F S M 格子 3 0）上にロードすることによって公開することが可能である。

20

30

【 0 0 6 0 】

ある例では、イメージは、イメージのビット値を S M E 3 4、3 6 および他のハードウェア素子に直接的にロードしてまたはイメージを 1 つ以上のレジスタ中にロードし、次に、ビット値をレジスタから S M E 3 4、3 6 および他のハードウェア素子に書き込むことによって、F S M 格子 3 0 上にロードすることが可能である。ある例では、F S M 格子 3 0 のハードウェア素子（例えば、S M E 3 4、3 6、特殊目的素子 5 8、構成可能切り替え素子 4 0、4 2、4 4）は、構成用デバイスおよび/またはコンピュータがイメージを、このイメージを 1 つ以上のメモリアドレスに書き込むことによって、F S M 格子 3 0 上にロードすることが可能となるようにマッピングされたメモリである。

40

【 0 0 6 1 】

本明細書に説明する方法の例は、少なくとも部分的に機械またはコンピュータに実装することが可能である。一部の例は、上記の例に説明したように方法を実施するために電子デバイスを構成するように動作可能な命令で符号化されたコンピュータ読み取り可能媒体または機械読み取り可能媒体を含みうる。このような方法の実装例は、マイクロコードなどのコード、アセンブリ言語コード、最高水準言語コードなどを含みうる。このようなコードは、様々な方法を実施するためのコンピュータ読み取り可能命令を含みうる。コードは、コンピュータプログラムプロダクトの部分形成し得る。さらに、コードは、実行中または他の時間で、1 つ以上の揮発性または不揮発性のコンピュータ読み取り可能媒体上に有形に記憶され得る。コンピュータ読み取り可能媒体は、これには限られないが、ハー

50

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리카ードもしくはスティック、ランダムアクセスメモリ（RAM）、リードオンリーメモリ（ROM）などを含み得る。

【0062】

ここで図9を参照すると、状態機械エンジン14（例えば、単一チップ上の単一デバイス）の実施形態が示されている。以前に説明したように、状態機械エンジン14は、データバスを介してメモリ16などのソースからデータを受信するように構成される。図示する実施形態では、データは、ダブルデータレート3（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に送出され得る。DDR3バスインターフェース130は、1ギガバイト/秒以上の速度でデータを交換すること（例えば、提供することおよび受信すること）が可能であり得る。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によって分析される速度を超え得る。理解されるように、分析されるデータのソース次第では、バスインターフェース130は、NANDフラッシュインターフェース、周辺構成要素相互接続（PCI）インターフェース、ギガビット媒体独立インターフェース（GMMI）などの、状態機械エンジン14に対する、データソースへのおよびからのデータを交換するための任意の適切なバスインターフェースであり得る。以前に説明したように、状態機械エンジン14は、データを分析するように構成された1つ以上のFSM格子30を含む。各々のFSM格子30は、2つの半格子に分割され得る。図示する実施形態では、各々の半格子は、24KのSME（例えば、SME34、36）を含み得るため、格子30は48KのSMEを含む。格子30は、図2～5に関連して以前に説明したように配列された任意の所望の数のSMEを備え得る。さらに、たった1つのFSM格子30を図示しているが、状態機械エンジン14は、以前に説明したように複数のFSM格子30を含み得る。

【0063】

分析されるデータは、バスインターフェース130で受信されて、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格子30に提供され得る。図示する実施形態では、データ経路は、データバッファ132、命令バッファ133、処理バッファ134、ならびにランク間（IR）バスおよび処理バッファインターフェース136を含む。データバッファ132は、分析されるデータを受信して、一時的に記憶するように構成される。1つの実施形態では、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2つのデータバッファ132のうちの一方から排出されている間に、他方のデータバッファ132に記憶されて、FSM格子30によって分析される。バスインターフェース130は、分析されるデータを、データバッファ132が一杯になるまでデータバッファ132に提供するように構成され得る。データバッファ132が一杯になった後、バスインターフェース130は、他の目的のために自由に用いられるように（例えば、データバッファ132が分析されるさらなるデータを受信するように利用可能となるまで、データストリームからの他のデータを提供するように）構成され得る。図示する実施形態では、データバッファ132は、各々が32Kバイトであり得る。命令バッファ133は、分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を構成することに対応する命令などの命令を、バスインターフェース130を介してプロセッサ12から受信するように構成される。IRバスおよび処理バッファインターフェース136は、データを処理バッファ134に提供することを容易化し得る。IRバスおよび処理バッファインターフェース136は、データが順番にFSM格子30によって処理されることを保証するために用いることが可能である。IRバスおよび処理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、そのデータが受信されて正確に分析されるように調整し得る。一般的に、IRバスおよび処理バッファインターフェース136は、あるデバイスランクの複数のデバイスの使用を許容する。これらのデバイスランクの複数のデバイスは、これら複数のデバイスの全てが共有されるデータの全てを正しい順序で受信するようにデータを共有する。例えば、複数

10

20

30

40

50

の物理的デバイス（例えば、状態機械エンジン 14、チップ、個別のデバイス）は、ランクに分けて配列し得るし、また、データを I R バスおよび処理バッファインターフェース 136 を介して互いに対して提供し得る。この応用の目的のため、「ランク」という用語は、同じチップ選択物に接続された状態機械エンジン 14 の集合のことである。図示する実施形態では、I R バスおよび処理バッファインターフェース 136 は、8 ビットのデータバスを含み得る。

【0064】

図示する実施形態では、状態機械エンジン 14 はまた、状態機械エンジン 14 におよびこれからデータを提供する際の支援として、デコンプレッサ 138 およびコンプレッサ 140 を含む。理解され得るように、コンプレッサ 140 およびデコンプレッサ 138 は、ソフトウェアおよび/またはハードウェアの設計を簡略化するために同じ圧縮アルゴリズムを用い得るが、しかしながら、コンプレッサ 140 およびデコンプレッサ 138 は、異なるアルゴリズムも用い得る。データを圧縮することによって、バスインターフェース 130（例えば、DDR3 バスインターフェース）を利用する時間が最小化され得る。本実施形態では、コンプレッサ 140 は、状態ベクトルデータ、構成データ（例えば、プログラミングデータ）、および FSM 格子 30 による分析後に得られた適合結果データを圧縮するために用いられ得る。1つの実施形態では、コンプレッサ 140 およびデコンプレッサ 138 は、コンプレッサ 140 およびデコンプレッサ 138 へおよび/から流れるデータが修正されない（例えば、圧縮も解凍もされない）ように、無効化され得る（例えば、オフされ得る）。

【0065】

コンプレッサ 140 およびデコンプレッサ 138 はまた、複数集合のデータを取り扱うように構成することが可能であり、各々の集合のデータは可変長であり得る。圧縮されたデータを「パディングする」して、各々の圧縮された領域がいつ終了するかに関するインジケータを含むことによって、コンプレッサ 140 は、状態機械エンジン 14 を介して全体的な処理速度を改善し得る。

【0066】

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、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 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 格子 30 の複数の状態ベクトルを記憶し、状態ベクトルを状態機械エンジン 14 上にもしくはこれから離れるように移動させ、状態ベクトルを FSM 格子 30 に提供して FSM 格子 30 を、提供された状態ベクトルに対応する状態に復元させるために用いられ得る。例えば、各々の状態ベクトルは状態ベクトルキャッシュメモリ 142 中に一時的に記憶され得る。すなわち、各々の SME 34、36 の各々の状態が記憶され得るが、それにより、新しいデータ集合（例えば、検索項目）の分析のために SME 34、36 を開放しながら、状態が復元され、後でさらなる分析で用いられ得る。一般的なキャッシュのように、状態ベクトルキャッシュメモリ 142 は、状態ベクトルを、例えば、ここでは FSM 格子 30 によって迅速に検索されて用いられるように記憶することを許容する。図示する実施形態では、状態ベクトルキャッシュメモリ 142 は、最大で 512 の状態ベクトルを記憶し得る。各々の状態ベクトルは、FSM 格子 30 の SME 34、36 の状態（例えば、起動された状態または起動されていない状態）およびカウンタ 58 の動的な（例えば、現在の）カウント値を含み得る。

【0067】

理解されるように、状態ベクトルデータは、あるランクの異なる状態機械エンジン 14（例えば、チップ）間で交換され得る。状態ベクトルデータは、FSM 格子 30 の SME 34、36 の状態とカウンタ 58 の動的カウント値を同期化すること、複数の状態機械エンジン 14 全体にわたって同じ機能を実施すること、複数の状態機械エンジン 14 全体にわたって同じ結果を再現すること、複数の状態機械エンジン 14 全体にわたって結果をカスケードすること、複数の状態機械エンジン 14 を介してカスケードされたデータを分析

するために用いられる S M E 3 4、3 6 の状態およびカウンタ 5 8 の動的カウント値の履歴を記憶すること、などの様々な目的のために、異なる状態機械エンジン 1 4 の間で交換され得る。さらにそのうえ、状態機械エンジン 1 4 内で、状態ベクトルデータは状態ベクトルを迅速に復元するために用いられ得ることに留意すべきである。例えば、状態ベクトルデータは、S M E 3 4、3 6 の状態およびカウンタ 5 8 の動的カウント値を（例えば、新しい検索項目を検索するために）初期状態に復元させること、S M E 3 4、3 6 の状態およびカウンタ 5 8 の動的カウント値を（例えば、以前に検索された検索項目を検索するために）先行する状態に復元させること、および S M E 3 4、3 6 の状態およびカウンタ 5 8 の動的カウント値を（例えば、カスケード検索で検索項目を検索するために）カスケード構成用に構成されるように変更すること、のために用いられ得る。ある実施形態では、状態ベクトルデータは、（例えば、状態ベクトルデータの分析、修正を適用するための状態ベクトルデータの再構成、効率を改善するための状態ベクトルデータの再構成などのために）状態ベクトルデータがプロセッサ 1 2 に提供され得るように、バスインターフェース 1 3 0 に提供され得る。

10

【 0 0 6 8 】

例えば、ある実施形態では、状態機械エンジン 1 4 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 1 4 1 によって記憶されたデータ）を F S M 格子 3 0 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、F S M 格子 3 0 を復元させる（例えば、リセットする、初期化する）ために、修正された状態ベクトルデータを状態機械エンジン 1 4 に提供し得る。したがって、外部デバイスは、状態機械エンジン 1 4 が所望次第で状態をスキップする（例えば、ジャンプして回る）ように、状態ベクトルデータを修正し得る。

20

【 0 0 6 9 】

状態ベクトルキャッシュメモリ 1 4 2 は、状態ベクトルデータを任意の適切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 1 4 2 は、状態ベクトルを、F S M 格子 3 0、別の F S M 格子 3 0（例えば、I R バスおよび処理バッファインターフェース 1 3 6 を介して）、デコンプレッサ 1 3 8 などから受信し得る。図示する実施形態では、状態ベクトルキャッシュメモリ 1 4 2 は、状態ベクトルを、他のデバイスから状態ベクトルメモリバッファ 1 4 4 を介して受信し得る。さらにそのうえ、状態ベクトルキャッシュメモリ 1 4 2 は、状態ベクトルデータを任意の適切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 1 4 2 は、状態ベクトルデータを、状態ベクトルメモリバッファ 1 4 4、状態ベクトル中間入力バッファ 1 4 6、および状態ベクトル中間出力バッファ 1 4 8 に提供し得る。

30

【 0 0 7 0 】

状態ベクトルメモリバッファ 1 4 4、状態ベクトル中間入力バッファ 1 4 6、および状態ベクトル中間出力バッファ 1 4 8 などのさらなるバッファを、状態ベクトルキャッシュメモリ 1 4 2 と一緒に用いて、インターリーブされたパケットを持つ別個のデータ集合を状態機械エンジン 1 4 によって処理しながら、状態ベクトルの迅速な取り出しおよび記憶に対応することができる。図示する実施形態では、状態ベクトルメモリバッファ 1 4 4、状態ベクトル中間入力バッファ 1 4 6、および状態ベクトル中間出力バッファ 1 4 8 の各々は、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 1 4 4 は、任意の適切なデバイスから状態ベクトルデータを受信することおよび状態ベクトルデータを任意の適切なデバイスに提供することの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 1 4 4 は、F S M 格子 3 0、別の F S M 格子 3 0（例えば、I R バスおよび処理バッファインターフェース 1 3 6 を介して）、デコンプレッサ 1 3 8、および状態ベクトルキャッシュメモリ 1 4 2 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 1 4 4 は、（例えば、他の F S M 格子 3 0 用の）I R バスおよび処理バッファインターフェース 1 3 6、コンプレッサ 1 4 0、および状態ベクトルキャッシュメモリ 1 4 2 に、状態ベクトルデータを提供するために用いられ得る。

40

50

【 0 0 7 1 】

同様に、状態ベクトル中間入力バッファ 1 4 6 は、状態ベクトルデータを任意の適切なデバイスから受信すること、および状態ベクトルデータを任意の適切なデバイスに提供することの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 1 4 6 は、F S M 格子 3 0 (例えば、I R バスおよび処理バッファインターフェース 1 3 6 を介して)、デコンプレッサ 1 3 8、および状態ベクトルキャッシュメモリ 1 4 2 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 1 4 6 は、状態ベクトルを F S M 格子 3 0 に提供するために用いられ得る。さらにそのうえ、状態ベクトル中間出力バッファ 1 4 8 は、状態ベクトルを任意の適切なデバイスから受信することおよび状態ベクトルを任意の適切なデバイスに提供することの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 1 4 8 は、F S M 格子 3 0 および状態ベクトルキャッシュメモリ 1 4 2 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 1 4 8 は、F S M 格子 3 0 (例えば、I R バスおよび処理バッファインターフェース 1 3 6 を介して)、およびコンプレッサ 1 4 0 に状態ベクトルを提供するために用いられ得る。

10

【 0 0 7 2 】

いったん目的とする結果が F S M 格子 3 0 によって生成されると、適合結果は、適合結果メモリ 1 5 0 に記憶され得る。すなわち、適合 (例えば、目的とするパターンの検出) を示す「適合ベクトル」は、適合結果メモリ 1 5 0 に記憶され得る。適合結果は、次に、例えば、バスインターフェース 1 3 0 を介してプロセッサ 1 2 に伝達されるように、適合バッファ 1 5 2 に送出することが可能である。以前に説明したように、適合結果は圧縮され得る。

20

【 0 0 7 3 】

さらなるレジスタおよびバッファは、状態機械エンジン 1 4 でも提供され得る。例えば、状態機械エンジン 1 4 は、制御およびステータスレジスタ 1 5 4 を含み得る。加えて、復元およびプログラムバッファ 1 5 6 は、初期に F S M 格子 3 0 の S M E 3 4、3 6 を構成し、分析中に F S M 格子 3 0 中の S M E 3 4、3 6 の状態を復元させる際に用いられるように提供され得る。同様に、保存および修復マップバッファ 1 5 8 もまた、セットアップおよび使用目的の保存および修復マップを記憶するために提供され得る。

【 0 0 7 4 】

図 1 0 は、デバイスのランクに配列された例としての複数の物理的状态機械エンジン 1 4 を示す。以前に説明したように、分析されるデータは、バスインターフェース 1 3 0 で受信される。バスインターフェース 1 3 0 は、各々の状態機械エンジン 1 4 (例えば、F 0、F 1、F 2、F 3、F 4、F 5、F 6、F 7) のデータバッファ 1 3 2 および命令バッファ 1 3 3 を含むデータバッファシステム 1 5 9 にデータを方向付ける。データバッファ 1 3 2 は、分析されるデータを受信して一時的に記憶するように構成される。図示する実施形態では、各々の状態機械エンジン 1 4 中に 2 つのデータバッファ 1 3 2 (例えば、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 2 つのデータバッファ 1 3 2 のうち的一方から排出されている間に、他方のデータバッファ 1 3 2 に記憶されて、F S M 格子 3 0 によって分析され得る。以前に説明したように、命令バッファ 1 3 3 は、分析されるデータに対応する命令などの命令を、バスインターフェース 1 3 0 を介してプロセッサ 1 2 から受信するように構成される。データバッファシステム 1 5 9 から、分析されるデータと、このデータに対応する命令とは、I R バスおよび処理バッファインターフェース 1 3 6 を介して、F S M 格子 3 0 (例えば、F a、F b、F c、F d、F e、F f、F g、F h) のうちの 1 つ以上に対して提供される。本実施形態では、物理的 F S M 格子 3 0 は、論理群に配列される。具体的には、F g および F h は論理群 A 1 6 2 に配列され、F e および F f は論理群 B 1 6 4 に配列され、F c および F d は論理群 C 1 6 6 に配列され、F a および F b は論理群 D 1 6 8 に配列される。さらにそのうえ、理解されるように、データは、F S M 格子 3 0 と別のデバイス (例えば、F S M 格子 3 0) との間で、I R バスおよび処理バッファインターフェース 1 3 6 を介して交換され得る。例えば

30

40

50

、 I Rバスおよび処理バッファインターフェース 1 3 6 は、 F S M 格子 3 0 の任意のものの間でデータを交換するために用いられ得る。 8 個の状態機械エンジン 1 4 が図示されているが、デバイスのランクは、任意の適切な数（例えば、 1、 2、 4、 8 など）の状態機械エンジン 1 4 を有し得る。理解されるように、 I Rバスおよび処理バッファインターフェース 1 3 6 は、（例えば、データバッファシステム 1 5 9 および F S M 格子 3 0 から）データを受信するための入力部を含み得る。同様に、 I Rバスおよび処理バッファインターフェース 1 3 6 は、（例えば、 F S M 格子 3 0 に）データを送出するための出力部を含み得る。

【 0 0 7 5 】

バスインターフェース 1 3 0 は、分析されるデータを、そのデータが効率的に使用されるように調整された形式で受信し得る。具体的には、図 1 1 ~ 1 4 は、どのようにしてデータがプロセッサ 1 2 によって、バスインターフェース 1 3 0 を介して状態機械エンジン 1 4 に提供されるデータブロック中に割り当てられる（例えば、グループ分けされる）かの例を示す。さらにそのうえ、図 1 5 ~ 1 7 は、どのようにしてデータブロックが、状態機械エンジン 1 4 のデータバッファシステム 1 5 9 を介して受信され、記憶され、提供され得るかの例を示す。図 1 8 は、どのようにしてデータブロックが論理群 1 6 2、 1 6 4、 1 6 6、 および 1 6 8 によって受信され得るかを示す。

【 0 0 7 6 】

ここで、図 1 1 を参照すると、状態機械エンジン 1 4 に提供されるデータブロック中にプロセッサ 1 2 によって割り当てられたデータセグメント（例えば、データ集合、検索項目）の例が図示されている。本実施形態では、複数のデータセグメントが単一のデータブロック中に割り当てられる。各々のデータブロックは、 F S M 格子 3 0（例えば、状態機械エンジン 1 4 のランクの 1 つ以上の状態機械エンジン 1 4）の単一の論理群によって分析されるように割り当てられる。例えば、データストリーム 1 7 0（例えば、プロセッサ 1 2 によって状態機械エンジン 1 4 に送出される多量のデータ）は、論理群 A 1 6 2 向けと意図されるデータに対応する第 1 のデータブロック 1 7 2 と、論理群 B 1 6 4 向けと意図されるデータに対応する第 2 のデータブロック 1 7 4 と、論理群 C 1 6 6 向けと意図されるデータに対応する第 3 のデータブロック 1 7 6 と、論理群 D 1 6 8 向けと意図されるデータに対応する第 4 のデータブロック 1 7 8 と、の中にプロセッサ 1 2 によって割り当てられる。具体的には、データストリーム 1 7 0 は、プロセッサ 1 2 によって、データセグメント 1 8 0、 1 8 2、 1 8 4、 1 8 6、 1 8 8、 1 9 0、 1 9 2、 1 9 4、 1 9 6、 1 9 8、 および 2 0 0 に分割される。理解されるように、データセグメント 1 8 0、 1 8 2、 1 8 4、 1 8 6、 1 8 8、 1 9 0、 1 9 2、 1 9 4、 1 9 6、 1 9 8、 および 2 0 0 の各々は、 F S M 格子 3 0 によって分析されるように設定されたデータ集合を表し得る。理解されるように、プロセッサ 1 2 は、データセグメント 1 8 0、 1 8 2、 1 8 4、 1 8 6、 1 8 8、 1 9 0、 1 9 2、 1 9 4、 1 9 6、 1 9 8、 および 2 0 0 を、何らかの適切な理由によって、データブロック 1 7 2、 1 7 4、 1 7 6、 および 1 7 8 に割り当て得る。例えば、プロセッサ 1 2 は、データ集合を効率的に処理するために、各々のデータ集合の長さおよび/またはデータ集合が分析される順序に基づいて、データセグメントをあるデータブロックに割り当て得る。

【 0 0 7 7 】

データセグメント 1 8 0、 1 8 2、 1 8 4、 1 8 6、 1 8 8、 1 9 0、 1 9 2、 1 9 4、 1 9 6、 1 9 8、 および 2 0 0 は、なんらかの適切な仕方を用いて、データブロック 1 7 2、 1 7 4、 1 7 6、 および 1 7 8 に割り当てられ得る。例えば、データセグメント 1 8 0、 1 8 2、 1 8 4、 1 8 6、 1 8 8、 1 9 0、 1 9 2、 1 9 4、 1 9 6、 1 9 8、 および 2 0 0 は、データブロック 1 7 2、 1 7 4、 1 7 6、 および 1 7 8 中のバイト数が最小化されるように、データブロック 1 7 2、 1 7 4、 1 7 6、 および 1 7 8 中に割り当てられ得る。別の例として、データセグメント 1 8 0、 1 8 2、 1 8 4、 1 8 6、 1 8 8、 1 9 0、 1 9 2、 1 9 4、 1 9 6、 1 9 8、 および 2 0 0 は、あるデータセグメントと一緒にグループ化されるように、データブロック 1 7 2、 1 7 4、 1 7 6、 および 1 7 8 中

10

20

30

40

50

に割り当てられ得る。

【 0 0 7 8 】

図示するように、第 1 のデータブロック 1 7 2 は、データセグメント A 1 8 0、データセグメント F 1 9 0、およびデータセグメント I 1 9 6 を含む。第 2 のデータブロック 1 7 4 は、データセグメント B 1 8 2 およびデータセグメント K 2 0 0 を含む。さらにそのうえ、第 3 のデータブロック 1 7 6 は、データセグメント C 1 8 4、データセグメント E 1 8 8、およびデータセグメント G 1 9 2 を含む。第 4 のデータブロック 1 7 8 は、データセグメント D 1 8 6、データセグメント H 1 9 4、およびデータセグメント J 1 9 8 を含む。

【 0 0 7 9 】

理解されるように、データブロックを効率的に処理するために、データブロックは全て、等しい量のデータを有し得る。さらにそのうえ、データブロック内のデータセグメントは、処理するデバイスが、データセグメントがいつ開始して停止するかを判定することが可能であるように、データブロック内の所定の間隔（例えば、バイト、語）で開始および／または停止し得る。しかしながら、データセグメントは、所定の間隔で開始および／または停止するための正確な量のデータを有していなくてもよい。したがって、データパディングが、データが所定の間隔で、データブロック内で開始および／または停止するように、あるデータセグメントの間に挿入され得る。加えて、データパディングは、全てのデータブロックが等しい量のデータを有するように、データブロックの最後に追加され得る。

【 0 0 8 0 】

ここで図 1 2 を参照すると、図 1 1 のデータブロック 1 7 2、1 7 4、1 7 6、および 1 7 8 のデータセグメントの間に挿入されるデータパディングの例が図示されている。例えば、第 1 のデータブロック 1 7 2 中で、データパディング 2 0 2 は、データセグメント A 1 8 0 とデータセグメント F 1 9 0 との間に挿入され得る。さらに、データパディング 2 0 4 は、データセグメント F 1 9 0 とデータセグメント I 1 9 6 との間に挿入され得る。別の例として、第 2 のデータブロック 1 7 4 中で、データパディング 2 0 6 は、データセグメント B 1 8 2 とデータセグメント K 2 0 0 との間に挿入され得る。第 3 のデータブロック 1 7 6 中で、データパディング 2 0 8 は、データセグメント C 1 8 4 とデータセグメント E 1 8 8 との間に挿入され得る。同様に、データパディング 2 1 0 は、データセグメント E 1 8 8 とデータセグメント G 1 9 2 との間に挿入され得る。別の例として、第 4 のデータブロック 1 7 8 中で、データパディング 2 1 2 は、データセグメント D 1 8 6 とデータセグメント H 1 9 4 との間に挿入され得る。加えて、データパディング 2 1 4 は、データセグメント H 1 9 4 とデータセグメント J 1 9 8 との間に挿入され得る。

【 0 0 8 1 】

データパディング 2 0 2、2 0 4、2 0 6、2 0 8、2 1 0、2 1 2、および 2 1 4 は、分析されないいずれかの適切なバイト数のデータ（例えば、無効データ、ジャンクデータ、フィルアデータ、ゴミデータなど）を含み得る。1 つの実施形態では、データパディングとして用いられるバイト数は、先行するデータセグメントのバイト数に加算されたときに、全体の語境界に到達するバイト数（すなわち、先行するデータセグメントのバイト数にデータパディングとして用いられるバイト数を加算した数が全体の語境界によって等分されるバイト数）であり得る。例えば、データパディング 2 0 2 のバイト数は、データパディング 2 0 2 とデータセグメント A 1 8 0（すなわち、先行するデータセグメント）との合計バイト数が全体の語境界によって等分されるような（例えば、剰余のない）数であり得る。図示する実施形態では、全体語境界は 8 バイトであり得る。他の実施形態では、全体語境界は、いずれかの適切なバイト数またはビット数であり得る。したがって、図示する実施形態では、データセグメント A 1 8 0 が 6 3 バイトのデータを含むことになっている場合、データパディング 2 0 2 は、（例えば 6 4 は 8 バイトによって等分されるので、データセグメント A 1 8 0 とデータパディング 2 0 2 との間に計で 6 4 バイトのデータを作成するために）1 バイトのデータを含む。別の例として、データセグメント A 1 8 0

が60バイト(例えば、8で等分されない)のデータを含んだ場合、データパディング202は、4バイトのデータを含む。さらなる例として、データセグメントA180が64バイトのデータを含んだ場合、データパディング202は、ゼロバイトのデータを含む、すなわち、言い換えれば、データパディング202は、データセグメントA180とデータセグメントF190の間では必要とされない。理解されるように、各々のデータパディング202、204、206、208、210、212、および214は、類似の仕方

【0082】

ここで図13を参照すると、図12のデータブロック172、174、176、および178のデータセグメントの後に挿入されたデータパディングの例が図示されている。具体的には、データパディングは、各々のデータブロック172、174、176、および178中のバイトの数を等しくするために、必要に応じて各々のデータブロック172、174、176、および178の最後に挿入され得る。さらにそのうえ、各々のデータブロック172、174、176、および178の最後のところのデータパディングは、各々のデータブロック172、174、176、および178が以前に説明したように全体語境界に到達するように用いられ得る。図示する実施形態では、データパディング216はデータセグメントI196の後に挿入され、データパディング218はデータセグメントG192の後に挿入され、データパディング220はデータセグメントJ198の後に挿入される。したがって、データブロック172、174、176、および178の各々は、等しい数のバイトを含み、データブロック172、174、176、および178の各々は、全体語境界に到達する。

【0083】

F S M格子30がデータパディングを有効なデータから区別することは困難であり得る。したがって、有効なデータの分析中に、データパディングがF S M格子30によって特定されて無視されるように、命令はデータブロック172、174、176、および178に付随し得る。このような命令は、バスインターフェース130を介してプロセッサ12によって状態機械エンジン14に送出され得るし、状態機械エンジン14の命令バッファ160によって受信、記憶、および提供され得る。命令を生成するために、プロセッサ12は、データストリーム170を領域222、224、226、228、230、232、234、および236に論理的に分割し得る。領域222、224、226、228、230、232、234、および236の末端境界は、いずれかのデータパディングが終了するときに各々の領域が終了するように形成され得る。例えば、第1の領域222は、データパディング208が終了するときに終了する。別の例として、第5の領域230は、データパディング204が終了するときに終了する。

【0084】

データブロック172、174、176、および178に付随する命令は、各々の領域222、224、226、228、230、232、234、および236のバイト数と、各々の領域内の各々のデータブロック172、174、176、および178の有効なバイト数(例えば、バイトパディングを除いたバイト数)とを含み得る。例えば、命令は、第1の領域222に対応するバイト数238と、第1の領域222内の第1のデータブロック172の有効なバイトに対応するバイト数240と、第1の領域222内の第2のデータブロック174の有効なバイトに対応するバイト数242と、第1の領域222内の第3のデータブロック176の有効なバイトに対応するバイト数244と、第1の領域222内の第4のデータブロック178の有効なバイトに対応するバイト数246と、を含み得る。

【0085】

同様に、命令は、第2の領域224に対応するバイト数248、250、252、254、および256と、第3の領域226に対応するバイト数258、260、262、264、および266と、第4の領域228に対応するバイト数268、270、272、274、および276と、第5の領域230に対応するバイト数278、280、282

、 284、および286と、第6の領域232に対応するバイト数288、290、292、294、および296と、第7の領域234に対応するバイト数298、302、304、および306と、第8の領域236に対応するバイト数308、312、314、および316と、を含み得る。したがって、命令を用いて、FSM格子30は、データセグメントと共に挿入されたデータパディングを特定し得る。1つの具体的なタイプの命令を本明細書に提示したが、データブロック172、174、176、および178の群と共に含まれる命令は、FSM格子30が、有効なデータをデータパディング（すなわち、無効なデータ）から区別することを可能とするいずれかの適切な命令の群であり得ることに留意すべきである。

【0086】

ここで図14を参照すると、状態機械エンジン14のデータバッファシステム159に伝達するようにプロセッサ12によって組織化された図13のデータブロック172、174、176、および178の例が図示されている。データブロック172、174、176、および178の各々は、データの行が全体語長に等しいバイト数318を有するように配列される。図示する実施形態では、全体語長は、状態機械エンジン14（例えば、F0、F1、F2、F3、F4、F5、F6、およびF7）の各々に対して、バイトで表わして8バイトである。データセグメントの各々からの最初のバイトは、各々のデータブロック172、174、176、および178の右側から始まり、各々のデータブロックの左側に向かって増加し、それにより、データセグメントA180の第1のバイトが列F0中にあり、データセグメントA180の第8のバイトが列F7中にあるようにする。理解されるように、列F0は、F0という状態機械エンジン14のデータバッファ132中に最初に記憶されるデータを表し、列F1は、F1という状態機械エンジン14のデータバッファ132中に最初に記憶されるデータを表し、となる。さらにそのうえ、データセグメントは、上から下に向かって行中に置かれる。図示するように、データセグメントとデータパディングとの各々の組み合わせは、列F7で終了する（すなわち、それらは各々が、全体語長にまで延在する）。さらにそのうえ、各々のデータブロック172、174、176、および178は、サイズが等しい。理解されるように、動作中は、データブロック172、174、176、および178は、プロセッサ12から状態機械エンジン14に連続的に提供され得る。

【0087】

データブロック172、174、176、および178からのデータは、論理群162、164、166、および168向けに意図されるデータがデータバッファシステム159中で混ざり合い、それにより各々の論理群162、164、166、および168向けに意図されるデータの一部分が各々の状態機械エンジン14（例えば、F0、F1、F2、F3、F4、F5、F6、およびF7）内で混ざり合うように、配列される。データは、この様式で受信および記憶され、データがバスインターフェース130を介してデータバッファシステム159に迅速に提供されることを可能にし得る。ある実施形態では、データバッファシステム159のデータバッファ132は、バスインターフェース130からのデータを（例えば、所定の間隔で）ラッチするように構成され得る。他の実施形態では、データバッファシステム159のデータバッファ132は、データバッファ132とバスインターフェース130との間の接続に基づいてデータの限られた部分を受信し得る。以下に詳述するように、混ざり合わされたデータは、データがIRバスおよび処理バッファインターフェース136を介してデータバッファシステム159から処理バッファ134に提供されたときに分類される。

【0088】

ここで図15に目を転じると、状態機械エンジン14によって受信されているデータブロック172、174、176、および178の例が図示されている。具体的には、データバッファシステム159は第1のデータブロック172を受信して、その後第2のデータブロック174、第3のデータブロック176、および第4のデータブロック178を受信する。上述したように、データブロック172、174、176、および178の

10

20

30

40

50

各々は、特定の論理群 162、164、166、および 168 によって分析されるように、プロセッサ 12 によって割り当てられ得る。データバッファシステム 159 がデータブロック 172、174、176、および 178 を受信するとき、データバッファシステム 159 は、データブロック 172、174、176、および 178 からのデータを体系的な様式でバッファ中に記憶し、それにより、データがデータバッファシステム 159 から F S M 格子 162、164、166、および 168 に正確に提供されるようにする。

【0089】

したがって、図 16 は、図 15 のデータブロック 172、174、176、および 178 がどのようにして状態機械エンジン 14 のデータバッファシステム 159 中に記憶されるかの例を示す。特に、第 1 のデータブロック 172 からのデータは、第 1 のバッファ行と、その後の 4 つ目毎の行（例えば、行 5、9、13、17 など）とに記憶される。同様に、第 2 のデータブロック 174 からのデータは、第 2 のバッファ行と、その後の 4 つ目毎の行（例えば、行 6、10、14、18 など）とに記憶される。さらに、第 3 のデータブロック 176 からのデータは、第 3 のバッファ行と、その後の 4 つ目毎の行（例えば、行 7、11、15、19 など）とに記憶される。加えて、第 4 のデータブロック 178 からのデータは、第 4 のバッファ行と、その後の 4 つ目毎の行（例えば、行 8、12、16、20 など）とに記憶される。一部の状態機械エンジン 14 は、4 つの F S M 格子 162、164、166、および 168 よりも少ないまたは多い格子を含み得ることに留意すべきである。したがって、他の実施形態では、データバッファシステム 159 は、データブロックからのデータを別の様式で記憶するように構成され得る。例えば、8 個の F S M 格子を持つ状態機械エンジン 14 では、各々の F S M 格子用のデータブロックは、バッファの 1 つの行と、その後の 8 つ目毎の行とに記憶されたデータを有し得る。

【0090】

ここで図 17 を参照すると、データバッファシステム 159 から複数の F S M 格子に提供されるデータの例が図示されている。具体的には、データはデータバッファシステム 159 から取り出されて、I R バスおよび処理バッファインターフェース 136 にデータバーストで提供される。（全体の語中のバイト数 318 が 8 バイトである）1 つの実施形態では、1 つの I R バスサイクルを完了させるのに、8 つのデータバーストを用いる。具体的には、第 1 のデータバースト 320 において、列 F 0 からの 4 バイト（例えば、各々のデータブロック 172、174、176、および 178 からの 1 バイトずつ）を、I R バスおよび処理バッファインターフェース 136 に提供する。同様に、第 2 のデータバースト 322 では、列 F 1 からの 4 バイト（例えば、各々のデータブロック 172、174、176、および 178 からの 1 バイトずつ）を、I R バスおよび処理バッファインターフェース 136 に提供する。第 3 のデータバースト 324 では、列 F 2 からの 4 バイト（例えば、各々のデータブロック 172、174、176、および 178 からの 1 バイトずつ）を、I R バスおよび処理バッファインターフェース 136 に提供する。さらに、第 4 のデータバースト 326 では、列 F 3 からの 4 バイト（例えば、各々のデータブロック 172、174、176、および 178 からの 1 バイトずつ）を、I R バスおよび処理バッファインターフェース 136 に提供する。第 5 のデータバースト 328 では、列 F 4 からの 4 バイト（例えば、各々のデータブロック 172、174、176、および 178 からの 1 バイトずつ）を、I R バスおよび処理バッファインターフェース 136 に提供する。同様に、第 6 のデータバースト 330 では、列 F 5 からの 4 バイト（例えば、各々のデータブロック 172、174、176、および 178 からの 1 バイトずつ）を、I R バスおよび処理バッファインターフェース 136 に提供する。第 7 のデータバースト 332 では、列 F 6 からの 4 バイト（例えば、各々のデータブロック 172、174、176、および 178 からの 1 バイトずつ）を、I R バスおよび処理バッファインターフェース 136 に提供する。さらに、第 8 のデータバースト 334 では、列 F 7 からの 4 バイト（例えば、各々のデータブロック 172、174、176、および 178 からの 1 バイトずつ）を、I R バスおよび処理バッファインターフェース 136 に提供する。したがって、データは、データバッファシステム 159 から取り出されて I R バスおよび処理バッファインター

10

20

30

40

50

フェース 136 に、データバーストを用いて体系的な内容で提供される。

【0091】

図 18 に目を転じると、データバースト 320、322、324、326、328、330、332、および 334 の複数の論理群 162、164、166、および 168 中のデータ提供の例が図示されている。具体的には、図示する実施形態では、論理群 A 162 (例えば、Fg、Fh) の処理バッファ 134 は、IR バスおよび処理バッファインターフェース 136 上に提供された各々のデータバースト 320、322、324、326、328、330、332、および 334 の第 1 のバイトをラッチするように構成され得る。同様に、論理群 B 164 (例えば、Fe、Ff) の処理バッファ 134 は、IR バスおよび処理バッファインターフェース 136 上に提供された各々のデータバースト 320、322、324、326、328、330、332、および 334 の第 2 のバイトをラッチするように構成され得る。加えて、論理群 C 166 (例えば、Fc、Fd) の処理バッファ 134 は、IR バスおよび処理バッファインターフェース 136 上に提供された各々のデータバースト 320、322、324、326、328、330、332、および 334 の第 3 のバイトをラッチするように構成され得る。論理群 D 168 (例えば、Fa、Fb) の処理バッファ 134 は、IR バスおよび処理バッファインターフェース 136 上に提供された各々のデータバースト 320、322、324、326、328、330、332、および 334 の第 4 のバイトをラッチするように構成され得る。

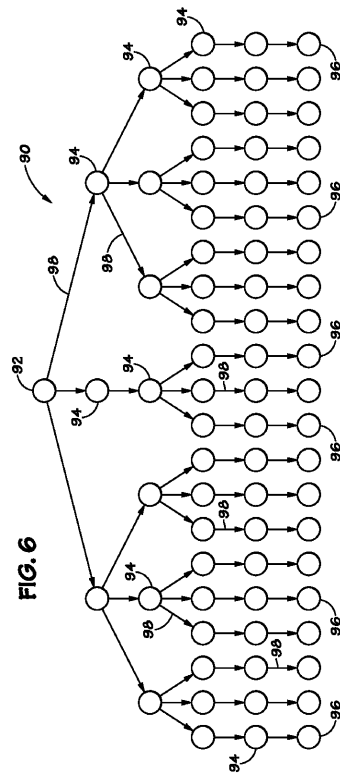
【0092】

理解されるように、論理群 162、164、166、および 168 の処理バッファ 134 は、IR バスおよび処理バッファインターフェース 136 上に提供された任意のバイトまたはバイトの組み合わせをラッチするように構成され得る。さらに、処理バッファ A および処理バッファ B は、同じまたは異なるバイトをラッチするように構成され得る。1つの実施形態では、状態機械エンジン 14 は、2 つよりも少ないまたは多い処理バッファ 134 を含み得る。このような実施形態では、各々の処理バッファ 134 は、IR バスおよび処理バッファインターフェース 136 上に提供された特定のバイト (例えば、バースト) をラッチするように構成され得る。論理群 162、164、166、および 168 の処理バッファ 134 はまた、データバーストに付随する命令をデータバッファシステム 159 から受信するように構成され得る。命令を用いて、論理群 162、164、166、および 168 の処理バッファ 134 は、データ領域中のバイトの総数とそのデータ領域中の有効なバイトの総数との間の差に対応するデータを無視し得る。

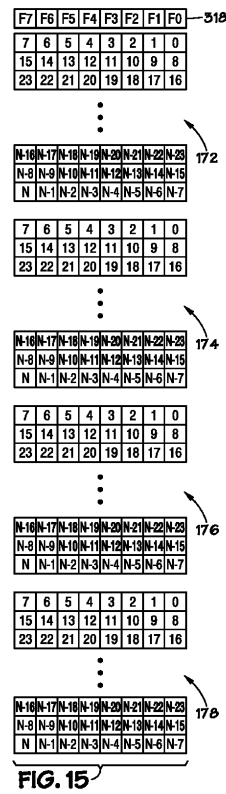
【0093】

本発明は様々な修正および代替の形態の影響を受け得るとはいえ、特定の実施形態を図面中で例として示し、本明細書中で詳細に説明した。しかしながら、本発明は開示する特定の形態に制限されることを意図するものではないことを理解すべきである。むしろ、本発明は、次の添付クレームによって定義される本発明の精神および範囲内の全ての修正例、等価例、および代替例を範囲に収めるものである。

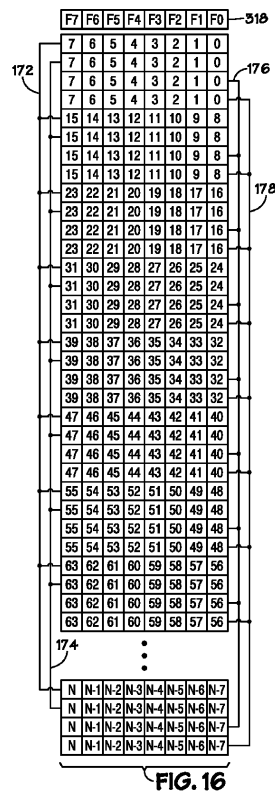
【図 6】



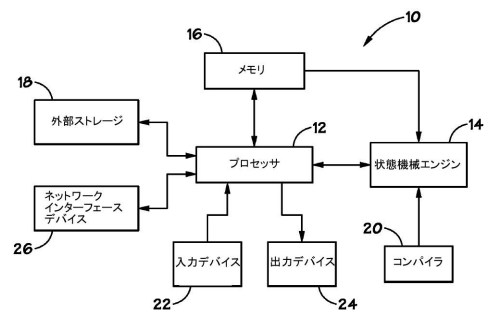
【図 15】



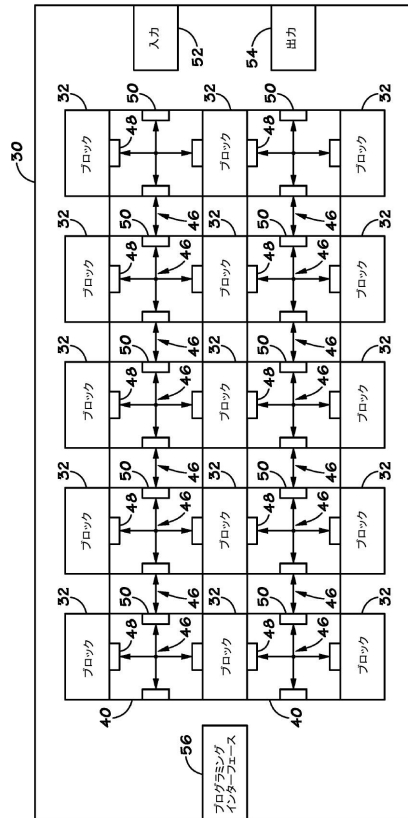
【図 16】



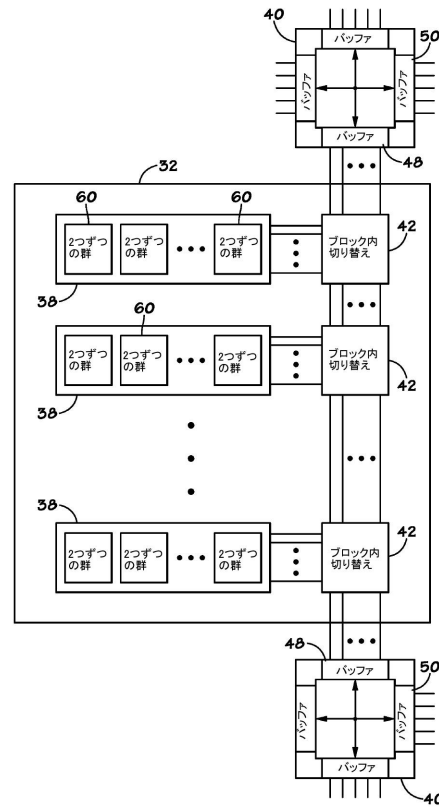
【図 1】



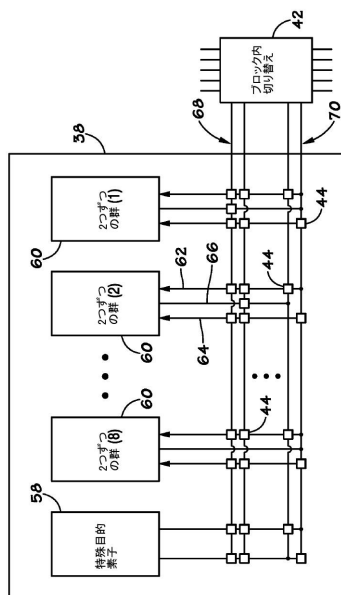
【図 2】



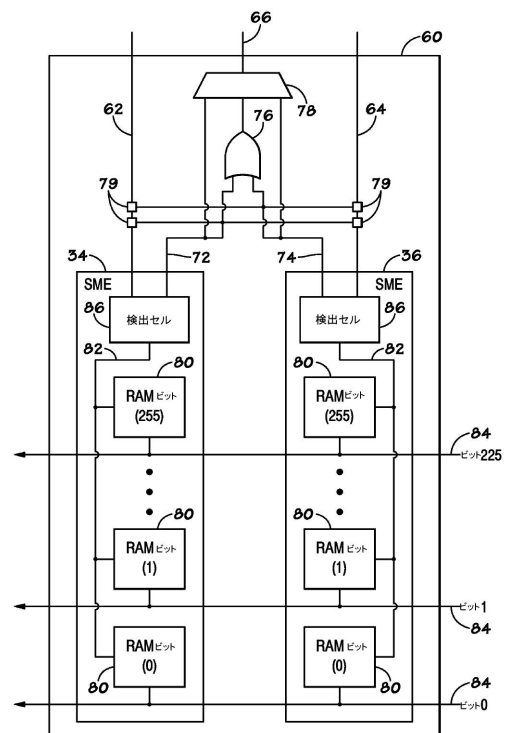
【図 3】



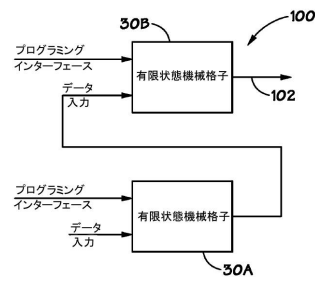
【図 4】



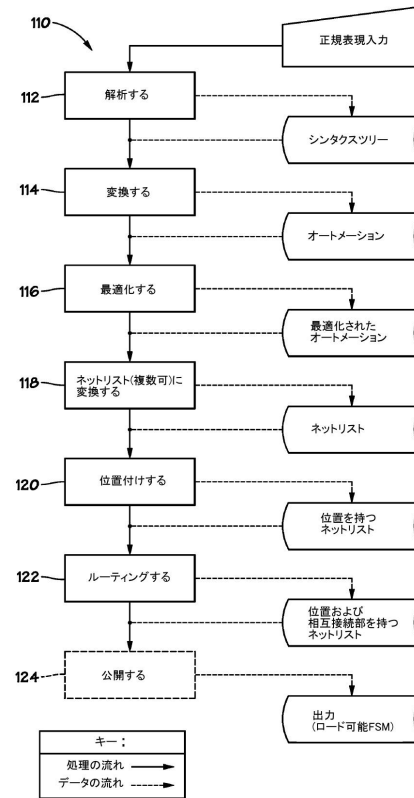
【図 5】



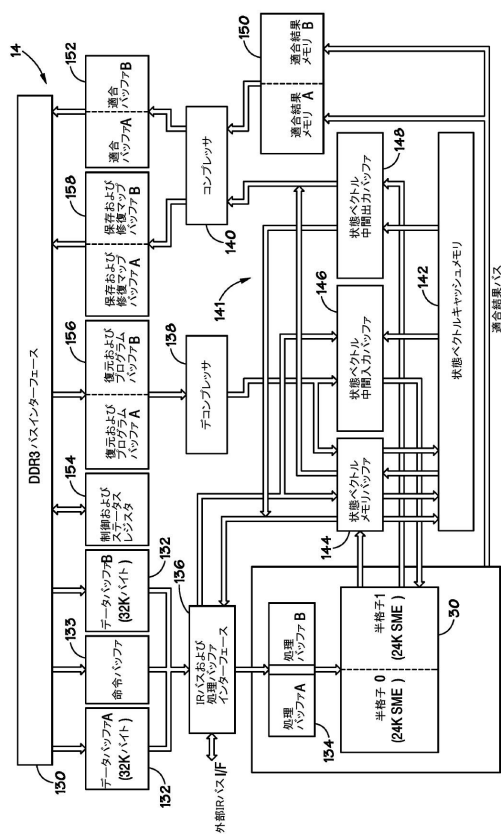
【 図 7 】



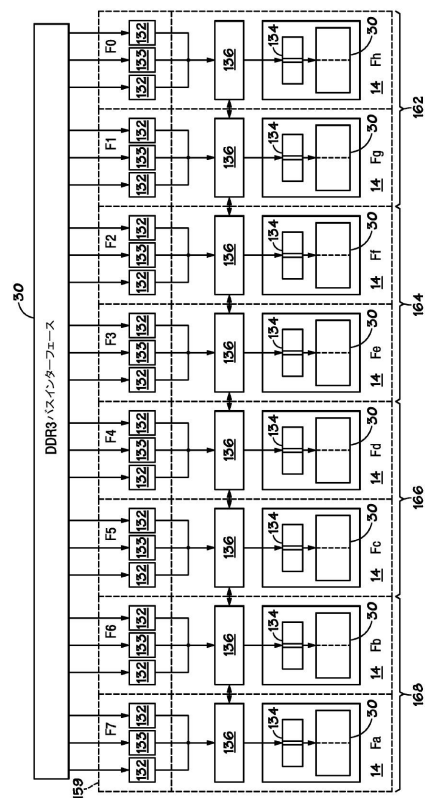
【 図 8 】



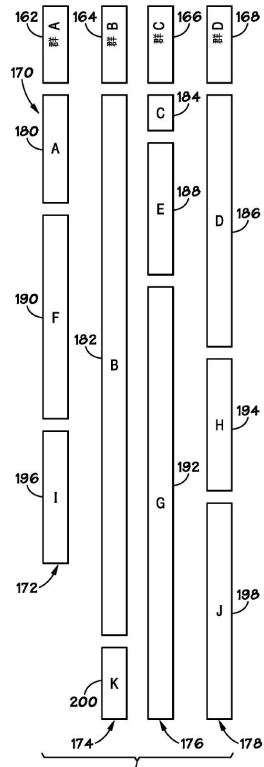
【 図 9 】



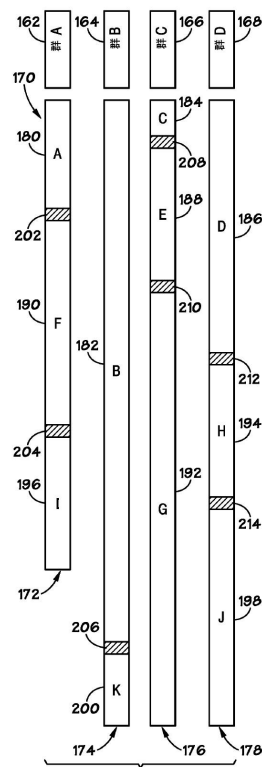
【 図 1 0 】



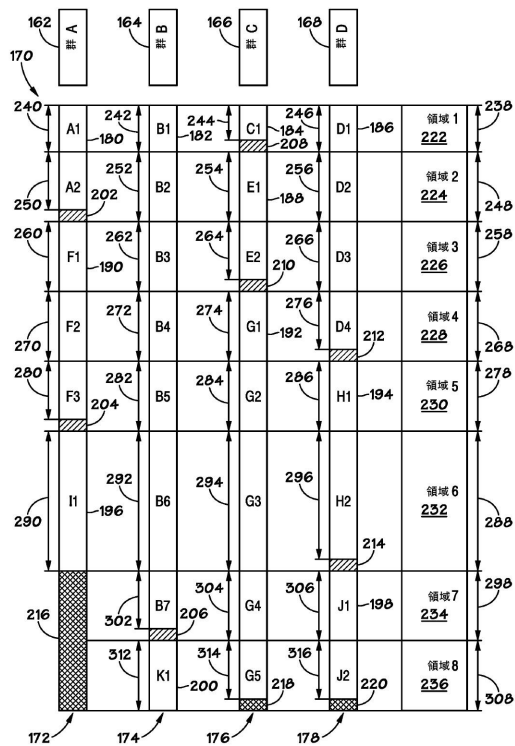
【図 1 1】



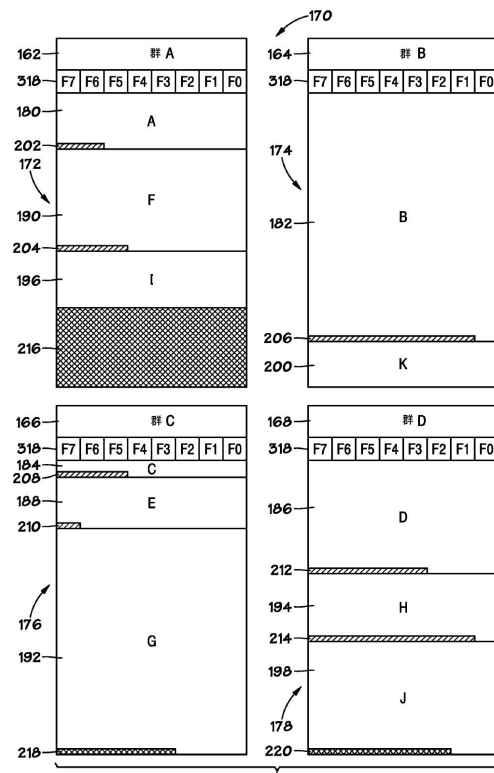
【図 1 2】



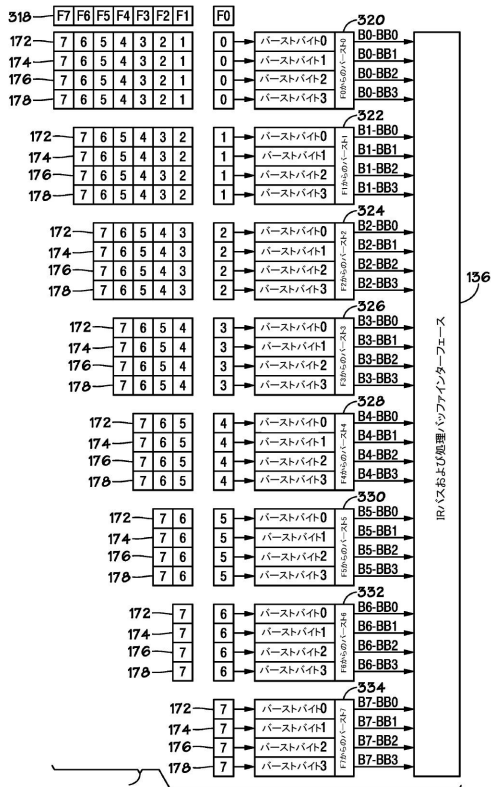
【図 1 3】



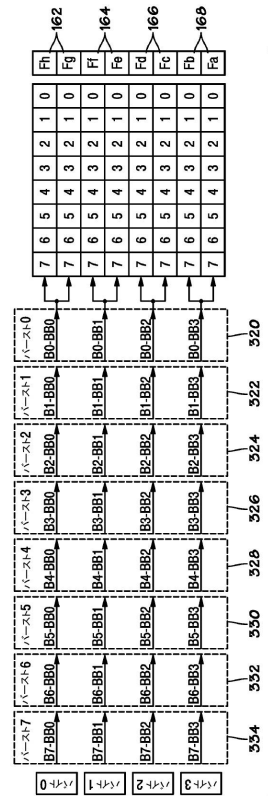
【図 1 4】



【図 17】



【図 18】



フロントページの続き

- (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, ボイズ, ウエスト アルバニー ストリート 108
36
- (72)発明者 ベインズ, インダージット エス .
アメリカ合衆国, アイダホ州 83716, ボイズ, サウス ケルトン プレイス 6291

審査官 樋口 龍弥

- (56)参考文献 特表2005-524142(JP, A)
特開2009-301137(JP, A)
特表2001-503577(JP, A)
特開平10-69459(JP, A)
特開平8-87462(JP, A)

- (58)調査した分野(Int.Cl., DB名)
- | | |
|------|-------|
| G06F | 17/30 |
| G06F | 7/00 |
| G06F | 9/50 |