

(19) 日本国特許庁(JP)

## (12) 特許公報(B2)

(11) 特許番号

特許第6177418号  
(P6177418)

(45) 発行日 平成29年8月9日(2017.8.9)

(24) 登録日 平成29年7月21日(2017.7.21)

(51) Int.Cl.

G06F 15/82 (2006.01)

F 1

G06F 15/82 610G  
G06F 15/82 650C

請求項の数 22 (全 46 頁)

(21) 出願番号 特願2016-501291 (P2016-501291)  
 (86) (22) 出願日 平成26年3月11日 (2014.3.11)  
 (65) 公表番号 特表2016-515266 (P2016-515266A)  
 (43) 公表日 平成28年5月26日 (2016.5.26)  
 (86) 國際出願番号 PCT/US2014/023589  
 (87) 國際公開番号 WO2014/150554  
 (87) 國際公開日 平成26年9月25日 (2014.9.25)  
 審査請求日 平成29年1月13日 (2017.1.13)  
 (31) 優先権主張番号 61/788,364  
 (32) 優先日 平成25年3月15日 (2013.3.15)  
 (33) 優先権主張国 米国(US)  
 (31) 優先権主張番号 14/065,168  
 (32) 優先日 平成25年10月28日 (2013.10.28)  
 (33) 優先権主張国 米国(US)

早期審査対象出願

最終頁に続く

(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, ボイズ, サウス フェデ  
 ラル ウェイ 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 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 1  
 806

(54) 【発明の名称】ステートマシンエンジンによって受信されるデータを提供するための方法及び装置

## (57) 【特許請求の範囲】

## 【請求項 1】

第一の複数のデータ線及び第二の複数のデータ線を備えており、前記第一の複数のデータ線が、前記第二の複数のデータ線とパラレルにデータを提供するように構成され、前記第一の複数のデータ線は、前記第二の複数のデータ線と異なる、データバスと、

前記第一の複数のデータ線に結合された第一のステートマシンエンジンと、

前記第二の複数のデータ線に結合された第二のステートマシンエンジンと、を備えており、前記第一のステートマシンエンジンは、前記第二のステートマシンエンジンと異なり、前記第二のステートマシンエンジンが前記第二の複数のデータ線からデータストリームの第二の部分を受信するのとパラレルに、前記第一のステートマシンエンジンが前記第一の複数のデータ線から前記データストリームの第一の部分を受信するように構成されており、前記データストリームの前記第一の部分は、前記データストリームの前記第二の部分と異なり、

前記第二のステートマシンエンジンが、前記データストリームの前記第二の部分を前記第一のステートマシンエンジンに提供するように構成されており、前記第一のステートマシンエンジンが、前記データストリームの前記第一の部分及び前記データストリームの前記第二の部分を解析するように構成された、装置。

## 【請求項 2】

前記第一及び第二のステートマシンエンジンの各々が、複数のコンフィギュラブル素子を有するそれぞれのステートマシンラチスを備えており、前記複数のコンフィギュラブル

素子の各々が、前記データストリームの少なくとも一部を解析し、前記解析の結果を出力するように構成された複数のセルを備えた、請求項 1 の装置。

### 【請求項 3】

前記第一のステートマシンエンジン及び前記第二のステートマシンエンジンに結合されたアドレスバスを備えた、請求項1の装置。

#### 【請求項 4】

前記第一及び第二のステートマシンエンジンが、前記アドレスバスから信号をパラレルに受信するように構成された、請求項3の装置。

## 【請求項 5】

前記第一のステートマシンエンジンに結合された第一の制御線、及び前記第二のステートマシンエンジンに結合された第二の制御線を備えており、前記第二のステートマシンエンジンが前記第二の制御線から第二の組の信号を受信するのとパラレルに、前記第一のステートマシンエンジンが、前記第一の制御線から第一の組の信号を受信するように構成された、請求項1の装置。

### 【請求項 6】

前記データバスが、第三の複数のデータ線、第四の複数のデータ線、第五の複数のデータ線、第六の複数のデータ線、第七の複数のデータ線、及び第八の複数のデータ線を備えており、前記第一の複数のデータ線が、前記第二の複数のデータ線、前記第三の複数のデータ線、前記第四の複数のデータ線、前記第五の複数のデータ線、前記第六の複数のデータ線、前記第七の複数のデータ線、及び前記第八の複数のデータ線とパラレルにデータを提供するように構成された、請求項1の装置。

### 【請求項 7】

前記第三の複数のデータ線に結合された第三のステートマシンエンジンと、  
前記第四の複数のデータ線に結合された第四のステートマシンエンジンと、  
前記第五の複数のデータ線に結合された第五のステートマシンエンジンと、  
前記第六の複数のデータ線に結合された第六のステートマシンエンジンと、  
前記第七の複数のデータ線に結合された第七のステートマシンエンジンと、  
前記第八の複数のデータ線に結合された第八のステートマシンエンジンと、を備えた、  
請求項 6 の装置。

### 【請求項 8】

前記第二のステートマシンエンジンが前記第二の複数のデータ線から前記データストリームの前記第二の部分を受信し、前記第三のステートマシンエンジンが前記第三の複数のデータ線から前記データストリームの第三の部分を受信し、前記第四のステートマシンエンジンが前記第四の複数のデータ線から前記データストリームの第四の部分を受信し、前記第五のステートマシンエンジンが前記第五の複数のデータ線から前記データストリームの第五の部分を受信し、前記第六のステートマシンエンジンが前記第六の複数のデータ線から前記データストリームの第六の部分を受信し、前記第七のステートマシンエンジンが前記第七の複数のデータ線から前記データストリームの第七の部分を受信し、かつ前記第八のステートマシンエンジンが前記第八の複数のデータ線から前記データストリームの第八の部分を受信するのとパラレルに、前記第一のステートマシンエンジンが前記第一の複数のデータ線から前記データストリームの前記第一の部分を受信するように構成された、請求項 7 の装置。

### 【請求項 9】

前記第三のステートマシンエンジンが、前記データストリームの前記第三の部分を前記第一のステートマシンエンジンに提供するように構成されており、前記第四のステートマシンエンジンが、前記データストリームの前記第四の部分を前記第一のステートマシンエンジンに提供するように構成されており、前記第五のステートマシンエンジンが、前記データストリームの前記第五の部分を前記第一のステートマシンエンジンに提供するように構成されており、前記第六のステートマシンエンジンが、前記データストリームの前記第六の部分を前記第一のステートマシンエンジンに提供するように構成されており、前記第

七のステートマシンエンジンが、前記データストリームの前記第七の部分を前記第一のステートマシンエンジンに提供するように構成されており、前記第八のステートマシンエンジンが、前記データストリームの前記第八の部分を前記第一のステートマシンエンジンに提供するように構成されており、そして前記第一のステートマシンエンジンが、前記データストリームの前記第三の部分、前記データストリームの前記第四の部分、前記データストリームの前記第五の部分、前記データストリームの前記第六の部分、前記データストリームの前記第七の部分、及び前記データストリームの前記第八の部分を解析するように構成された、請求項 8 の装置。

【請求項 10】

前記第一のステートマシンエンジンが、前記データストリームの前記第一の部分を格納するように構成された第一のバッファを備えており、前記第二のステートマシンエンジンが、前記データストリームの前記第二の部分を格納するように構成された第二のバッファを備えた、請求項 1 の装置。 10

【請求項 11】

前記第一のステートマシンエンジンが、前記データストリームの前記第一の部分及び前記データストリームの前記第二の部分を格納するように構成された第三のバッファを備えた、請求項 10 の装置。

【請求項 12】

前記第二のステートマシンエンジンから前記データストリームの前記第二の部分を受信し、前記データストリームの前記第二の部分を前記第一のステートマシンエンジンに提供するように構成されたバッファインターフェイスを備えた、請求項 1 の装置。 20

【請求項 13】

前記バッファインターフェイスが、前記第一のステートマシンエンジン及び前記第二のステートマシンエンジンの少なくとも一方に命令を提供するように構成された、請求項 1 2 の装置。

【請求項 14】

前記第一のステートマシンエンジンが、前記バッファインターフェイスに提供された第一の組のデータをラッチするように構成された第一のバッファを備えており、前記第二のステートマシンエンジンが、前記バッファインターフェイスに提供された第二の組のデータをラッチするように構成された第二のバッファを備えた、請求項 1 2 の装置。 30

【請求項 15】

前記第一のバッファが、第一の所定間隔において前記第一の組のデータをラッチするように構成されており、前記第二のバッファが、第二の所定間隔において前記第二の組のデータをラッチするように構成された、請求項 1 4 の装置。

【請求項 16】

第一のチップ及び第二のチップを有し、前記第一のステートマシンエンジンは前記第一のチップに備えられ、前記第二のステートマシンエンジンは前記第二のチップに備えられた、請求項 1 の装置。

【請求項 17】

第一のステートマシンエンジンにおいて、データバスのうちの第一の複数のデータ線からデータストリームの第一の部分を受信することと、 40

第二のステートマシンエンジンにおいて、前記データバスのうちの第二の複数のデータ線から前記データストリームの第二の部分を受信することであって、前記第一のステートマシンエンジンは前記第二のステートマシンエンジンと異なり、前記データストリームの前記第一の部分は、前記データストリームの前記第二の部分と異なり、前記第一の複数のデータ線は、前記第二の複数のデータ線と異なり、前記第一のステートマシンエンジンが前記データストリームの前記第一の部分を受信するのとパラレルに、前記データストリームの前記第二の部分が前記第二のステートマシンエンジンにおいて受信される、ことと、

前記第二のステートマシンエンジンから前記データストリームの前記第二の部分を前記第一のステートマシンエンジンに提供することと、を含む、データ解析方法。 50

## 【請求項 18】

前記第一のステートマシンエンジンを使用して、前記データストリームの前記第一の部分及び前記データストリームの前記第二の部分を解析することを含む、請求項17の方法。

## 【請求項 19】

前記第一のステートマシンエンジンの第一のメモリアドレス、及び前記第二のステートマシンエンジンの第二のメモリアドレスをパラレルに選択するためのアドレス信号を、前記第一及び第二のステートマシンエンジンにおいて受信することを含む、請求項17の方法。

## 【請求項 20】

10

前記第一のメモリアドレスが前記第二のメモリアドレスと等しい、請求項19の方法。

## 【請求項 21】

前記データストリームの前記第一の部分と前記データストリームの前記第二の部分は、データの等しくない量を含み、前記データストリームの前記第一の部分、前記データストリームの前記第二の部分、又は、その両方は、前記データストリームの前記第一の部分と前記データストリームの前記第二の部分が等しいデータ量を含むまで、データパディングを用いて調整される、請求項1に記載の装置。

## 【請求項 22】

前記データパディングは、データが、所定の間隔で、データブロック内から始まる、及び／又は、データブロック内で止まるように、前記データストリームの前記第一の部分、前記データストリームの前記第二の部分、又は、その両方の、あるデータセグメントの間に追加される、請求項21に記載の装置。

20

## 【発明の詳細な説明】

## 【技術分野】

## 【0001】

## [関連出願の参照]

本願は、2013年3月15日に出願された米国特許出願第61/788,364号の非仮出願である。

## [技術分野]

## 【0002】

30

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、ある実施形態においては、データ解析のための並列デバイスを有する電子デバイスに関する。

## 【背景技術】

## 【0003】

複雑なデータ解析（例えば、パターン認識）は、従来のフォンノイマンベースコンピュータ上で実施するために非効率的である場合がある。しかしながら、生物学的脳、特に人の脳は、複雑なデータ解析の実施に熟練している。最新の研究では、人の脳が新皮質においてヒエラルキー的に組織化された一連のニューロン層を利用してデータ解析を実施することが示唆されている。ヒエラルキーのより下層にあるニューロンが、例えば、感覚器官からの「生信号」を解析し、一方で、より上層にあるニューロンが、より低いレベルのニューロンからの信号出力を解析する。新皮質におけるこのヒエラルキーシステムは、ことによると脳の他の区域と組み合わさって、空間推論、意識的思考及び複雑な言語などの高レベル機能を人が実施することを可能にする複雑なデータ解析を達成する。

40

## 【0004】

コンピューティングの分野においては、例えば、パターン認識タスクはますます困難なものになっている。コンピュータ間でますます大量のデータが伝送され、ユーザが検出を望むパターン数は増加している。例えば、スパムまたはマルウェアはしばしば、データストリーム内でパターン、例えば、特定の句またはコードの一部を検索することによって検出される。新規パターンが新規変形を検索するために実装され得るにつれて、パターン数は、スパム及びマルウェアの多様性とともに増加する。これらのパターンの各々に対し

50

てデータストリームを検索することは、コンピューティングボトルネックを形成する場合がある。しばしば、データストリームが受信されると、それは一度に1つずつ各パターンに対して検索される。システムがデータストリームの次の部分を検索するために準備するまでの遅延は、パターン数とともに増大する。したがって、パターン認識は、データの受領を遅延させる場合がある。

【発明の概要】

【発明が解決しようとする課題】

【0005】

ハードウェアは、パターン用のデータストリームを検索するために設計してきたが、このハードウェアは、しばしば、任意の期間において十分なデータ量を処理することができない。データストリームを検索するように構成された幾つかのデバイスは、複数の回路間でデータストリームを分配することによって、十分なデータ量を処理する。回路は、データストリームがパターンの一部にマッチするか否かを各自判定する。しばしば、多数の回路がパラレルに動作して、各自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、生物学的脳以上に効率的に複雑なデータ解析を実施することが可能なシステムは存在しなかった。そのようなシステムの開発が望ましい。

10

【図面の簡単な説明】

【0006】

【図1】本発明の種々の実施形態に従う、ステートマシンエンジンを有するシステムの一実施例を図示する。

20

【図2】本発明の種々の実施形態に従う、図1のステートマシンエンジンの有限ステートマシン(FSM)ラチスの一実施例を図示する。

【図3】本発明の種々の実施形態に従う、図2のFSMラチスのブロックの一実施例を図示する。

【図4】本発明の種々の実施形態に従う、図3のブロックの行の一実施例を図示する。

【図5】本発明の種々の実施形態に従う、図4の行のうちの2のグループの一実施例を図示する。

【図6】本発明の種々の実施形態に従う、有限ステートマシングラフの一実施例を図示する。

【図7】本発明の種々の実施形態に従う、FSMラチスで実現された2レベルヒエラルキーの一実施例を図示する。

30

【図8】本発明の種々の実施形態に従う、コンパイラが図2のFSMラチスのプログラミング用のバイナリファイルへとソースコードを変換するための方法の一実施例を図示する。

【図9】本発明の種々の実施形態に従う、ステートマシンエンジンを図示する。

【図10】本発明の種々の実施形態に従う、デバイスがランクにおいて配列された複数の物理的なステートマシンエンジンの一実施例を図示する。

【図11】本発明の種々の実施形態に従う、ステートマシンエンジンに提供されるデータブロックにグループ化されたデータセグメントの一実施例を図示する。

【図12】本発明の種々の実施形態に従う、図11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一実施例を図示する。

40

【図13】本発明の種々の実施形態に従う、図12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一実施例を図示する。

【図14】本発明の種々の実施形態に従う、ステートマシンエンジンのデータバッファシステムに伝送するように組織化された図13のデータブロックの一実施例を図示する。

【図15】本発明の種々の実施形態に従う、デバイスがランクにおいて配列されており、インターランク(IRD)バス及びプロセスバッファインターフェイスによってともに結合された複数の物理的なステートマシンエンジンの一実施例を図示する。

【図16】本発明の種々の実施形態に従う、複数の物理的なステートマシンエンジンに結合されたIRDバス及びプロセスバッファインターフェイスにおける信号の一実施例を図示

50

する。

【図17】本発明の種々の実施形態に従う、IRバス及びプロセスバッファインターフェイスにおける信号のタイミング図の一実施例を図示する

【図18】本発明の種々の実施形態に従う、複数の論理グループに組織化された複数の物理的なストートマシンエンジンのデータバッファに格納されたデータの一実施例を図示する。

【図19】本発明の種々の実施形態に従う、複数の論理グループに組織化された複数の物理的なストートマシンエンジンのプロセスバッファに格納されたデータの一実施例を図示する。

【図20】本発明の種々の実施形態に従う、1つの論理グループに組織化された複数の物理的なストートマシンエンジンのデータバッファに格納されたデータの一実施例を図示する。 10

【図21】本発明の種々の実施形態に従う、1つの論理グループに組織化された複数の物理的なストートマシンエンジンのプロセスバッファに格納されたデータの一実施例を図示する。

【発明を実施するための形態】

【0007】

ここで図面を参照する。図1は、参照数字10で概して示されるプロセッサベースシステムの実施形態を図示する。システム10（例えば、データ解析システム）は、例えば、デスクトップコンピュータ、ラップトップコンピュータ、ページャ、携帯電話、パーソナルオーガナイザ、ポータブル音声プレイヤ、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ってもよい。システム10は、さらに、ルータ、サーバまたは（例えば、前述のタイプのコンピュータのいずれかの）クライアントなどのネットワークノードであってもよい。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配信もしくは録画システム、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、工場オートメーションシステム、乗用車コンピュータシステム、または医療デバイスなどの、幾つかの他の種類の電子デバイスであってもよい。（本明細書中に使用される多くの他の用語と同様に、これらの種々のシステムの実施例を記述するために使用される用語は、幾つかの指示物を共有し、それゆえ、記載された他の品目があるからという理由で狭義に解釈されるべきではない。） 30

【0008】

システム10などの典型的なプロセッサベース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12が、システム10におけるシステム機能及び要求の処理を制御する。さらに、プロセッサ12は、システム制御を共有する複数のプロセッサを備えてもよい。プロセッサ12がシステム10内部または外部に格納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、プロセッサ12は、システム10内の各々の構成要素に直接的に結合されてもよいし、間接的に結合されてもよい。

【0009】

本明細書に記述する実施形態に従い、システム10は、ストートマシンエンジン14を含み、ストートマシンエンジン14はプロセッサ12の制御下で動作してもよい。本明細書中に使用されるようなストートマシンエンジン14は、単一デバイス（例えば、单一チップ）を指す。ストートマシンエンジン14は、任意のオートマトン理論を利用することができます。例えば、ストートマシンエンジン14は、これらに限定されないが、Mealyアーキテクチャ、Mooreアーキテクチャ、有限ストートマシン（FSM）、決定論的FSM（DFSM）、ビット・パラレルストートマシン（BPSM）などを含む多くのストートマシンアーキテクチャのうちの1つを利用することができる。様々なアーキテクチャを使用できるが、論述する目的のため、本願ではFSMに言及する。しかしながら、様々なストートマシンアーキテクチャのうちの任意の1つを使用しても記述した技術を利用できることが当業者には認識されるであろう。

【0010】

10

20

30

40

50

さらに下記するように、ステートマシンエンジン14は、多数（例えば、1つ以上）の有限ステートマシン（FSM）ラチス（例えば、ステートマシンエンジン14のコア）を含んでもよい。この適用目的のため、「ラチス」という用語は、素子（例えば、ブルセル、カウンタセル、ステートマシン素子、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クス、ルーティングネットワーク、フレーム）を指す。さらにまた、「ラチス」は、任意の適切な形状、構造またはヒエラルキー的組織（例えば、グリッド、立方形、球状、カスケーディング接続）を有してもよい。各FSMラチスは、各々が同一のデータをパラレルに受信して解析する複数のFSMを実現してもよい。さらに、FSMラチスのクラスタが同一の入力データをパラレルに解析できるように、FSMラチスは、グループ（例えば、クラスタ）で配列されてもよい。さらに、ステートマシンエンジン14のFSMラチスのクラスタをヒエラルキー構造に配列することによって、ヒエラルキー構造のより低レベルにおけるステートマシンラチスからの出力を、より高レベルにおけるステートマシンラチスに対する入力として使用してもよい。ヒエラルキー構造を通して直列にステートマシンエンジン14のパラレルFSMラチスのクラスタをカスケード接続することによって、より多くの複雑なパターンを解析（例えば、評価、検索など）できる。10

#### 【0011】

さらに、ステートマシンエンジン14のヒエラルキーパラレル構造に基づいて、ステートマシンエンジン14は、高い処理速度を利用するシステムにおいて複雑なデータ解析（例えば、パターン認識）に使用することができる。例えば、本明細書に記述する実施形態は、1Gバイト/秒の処理速度を持つシステムに組み込むことができる。したがって、ステートマシンエンジン14を利用することによって、高速メモリデバイスもしくは他の外部デバイスからのデータを迅速に解析することができる。ステートマシンエンジン14は、幾つかの基準（例えば、検索語）に従ってデータストリームを、ほぼ同時に、例えば、単一のデバイスサイクル期間に解析することができる。ステートマシンエンジン14のレベル上のFSMのクラスタ内の各FSMは、ほぼ同時にデータストリームから同一の検索語を各々受信して、各パラレルFSMラチスが、当該用語が処理基準における次の状態へとステートマシンエンジン14を進めるか否かを判定してもよい。ステートマシンエンジン14は、比較的多数の基準、例えば、100超、1000超または10000超の基準に従い用語を解析してもよい。これらはパラレルに動作するため、データストリームを遅くすることなく、比較的高いバンド幅を有するデータストリーム、例えば、1Gバイト/秒と同等以上のデータストリームに対して基準を適用してもよい。20

#### 【0012】

一実施形態においては、ステートマシンエンジン14は、データストリームにおける多数のパターンを認識（例えば、検出）するように構成されてもよい。例えば、ステートマシンエンジン14は、ユーザまたは他のエンティティが解析を望み得る様々なタイプのデータストリームのうちの1つ以上におけるパターンを検出するのに利用することができる。例えば、ステートマシンエンジン14は、インターネット上で受信されたパケット、または携帯電話ネットワーク上で受信された音声もしくはデータなどの、ネットワーク上で受信されたデータストリームを解析するように構成できる。一実施例においては、ステートマシンエンジン14は、スパムまたはマルウェア用のデータストリームを解析するように構成できる。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てもよく、データが、時間的、単語的または意味的に重要な順序などの、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、パラレルまたは順序に関係なく受信され、その後、例えば、インターネット上で受信されたパケットを再配列することによってシリアルデータストリームに変換されてもよい。幾つかの実施形態においては、データストリームはシリアルに用語を提示することができるが、各用語を表すビットはパラレルに受信されてもよい。データストリームはシステム10へと外部ソースから受信されてもよいし、メモリ16などのメモリデバイスに問い合わせ、メモリ16内に格納された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てもよい。他の実施例においては、ステート304050

マシンエンジン 14 は、あるワードを綴る文字シーケンス、遺伝子を特定する遺伝的塩基対のシーケンス、画像の一部を形成する画像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なファイルにおけるビットのシーケンス、または歌もしくは発話フレーズの一部を形成する音声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できる。解析されるべきデータストリームは、例えば、ベーステン、A S C I Iなどのバイナリフォーマットまたは他のフォーマットにおける複数のビットのデータを含んでもよい。ストリームは、例えば、幾つかのバイナリデジットなどの単一デジットまたは複数デジットを有するデータをエンコードしてもよい。

#### 【0013】

認識されるように、システム 10 はメモリ 16 を含むことができる。メモリ 16 は、例えば、ダイナミックランダムアクセスメモリ (D R A M)、静态ランダムアクセスメモリ (S R A M)、同期式D R A M (S D R A M)、ダブルデータレートD R A M (D D R S D R A M)、D D R 2 S D R A M、D D R 3 S D R A Mなどの揮発性メモリを含むことができる。メモリ 16 は、揮発性メモリと組み合わせて使用するための、リードオンリーメモリ (R O M)、P C - R A M、シリコン・酸化物・窒化物・酸化物・シリコン (S O N O S) メモリ、金属・酸化物・窒化物・酸化物・シリコン (M O N O S) メモリ、ポリシリコン浮遊ゲートベースメモリ、及び / または種々のアーキテクチャの他のタイプのフラッシュメモリ (例えば、N A N D メモリ、N O R メモリなど)などの不揮発性メモリも含むことができる。メモリ 16 は、ステートマシンエンジン 14 によって解析されるデータを提供できる、D R A M デバイスなどの 1 つ以上のメモリデバイスを含むことができる。本明細書中に使用されるような「提供」という用語は、総称的に、導き、入力、挿入、送信、転送、伝送、生成、付与、出力、配置、書き込みなどを指す。当該デバイスは、ソリッドステートドライブ (S S D)、マルチメディアメディアカード (M M C)、セキュアデジタル (S D) カード、コンパクトフラッシュ (C F) カード、または任意の他の適切なデバイスと称されるか、またはそれらを含んでもよい。さらに、当該デバイスはユニバーサルシリアルバス (U S B)、ペリフェラルコンポーネントインターフェクト (P C I)、P C I エクスプレス (P C I - E)、スマートコンピュータシステムインターフェイス (S C S I)、I E E E 1 3 9 4 (ファイアワイヤ)、または任意の他の適切なインターフェイスなどの任意の適切なインターフェイスを通してシステム 10 に結合できることが認識されるはずである。フラッシュメモリデバイスなどのメモリ 16 の動作を容易にするために、システム 10 は、メモリコントローラ (図示せず) を含むことができる。認識されるように、メモリコントローラは、独立したデバイスであってもよいし、プロセッサ 12 と一体であってもよい。さらに、システム 10 は、磁気記憶デバイスなどの外部ストレージ 18 を含むことができる。外部ストレージは、ステートマシンエンジン 14 に入力データを提供することもできる。

#### 【0014】

システム 10 は、多数のさらなる構成要素を含んでもよい。例えば、コンパイラ 20 を、図 8 に関してより詳細に記述するように、ステートマシンエンジン 14 を構成 (例えば、プログラム) するのに使用してもよい。ユーザがデータをシステム 10 に入力できる入力デバイス 22 も、プロセッサ 12 に結合することができる。例えば、入力デバイス 22 を使用して、ステートマシンエンジン 14 によるその後の解析のためにメモリ 16 にデータを入力することができる。入力デバイス 22 は、例えば、ボタン、スイッチング素子、キーボード、ライトペン、スタイルス、マウス及び / または音声認識システムを含むことができる。ディスプレイなどの出力デバイス 24 も、プロセッサ 12 に結合することができる。ディスプレイ 24 は、例えば、L C D、C R T、L E D 及び / または音声ディスプレイを含むことができる。システムは、インターネットなどのネットワークとインターフェイスを介して接続するために、ネットワークインターフェイスカード (N I C) などのネットワークインターフェイスデバイス 26 も含むことができる。認識されるように、システム 10 は、システム 10 の用途に応じて、多数の他のコンポーネントを含むことができる。

10

20

30

40

50

## 【0015】

図2～図5は、FSMラチス30の一実施例を図示する。一実施例においては、FSMラチス30は、ブロック32のアレイを備える。記述するように、各ブロック32は、FSMにおける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、コンフィギュラブル素子及び／または専用素子）を含むことができる。FSMにおける状態に類似して、ハードウェア素子は、入力ストリームを解析して、入力ストリームに基づいて下流ハードウェア素子をアクティブ化することができる。

## 【0016】

コンフィギュラブル素子は、多くの異なる機能を実施するように構成（例えば、プログラム）することができる。例えば、コンフィギュラブル素子は、（図3及び図4に示すような）行38及び（図2及び図3に示すような）ブロック32にヒエラルキー的に組織化された（図5に示す）ステートマシン素子（SME）34、36を含むことができる。SMEも、状態遷移素子（STE）とみなすことができる。ヒエラルキー的に組織化されたSME34、36の間で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素子40（図2及び図3に示す）、ブロック内スイッチング素子42（図3及び図4に示す）、及び行内スイッチング素子44（図4に示す）を含むプログラマブルスイッチング素子のヒエラルキーを使用することができる。

10

## 【0017】

下記するように、スイッチング素子は、ルーティング構造及びバッファを含むことができる。SME34、36は、FSMラチス30によって実現されるFSMの状態に対応することができる。SME34、36は、下記するようなプログラマブルスイッチング素子を使用することによってともに結合することができる。したがって、状態の機能に対応するようにSME34、36を構成し、かつFSMにおける状態間の遷移に対応するようにSME34、36をともに選択的に結合することによって、FSMをFSMラチス30において実現することができる。

20

## 【0018】

図2は、FSMラチス30の一実施例の全体図を図示する。FSMラチス30は、プログラマブルブロック間スイッチング素子40とともに選択的に結合することができる複数のブロック32を含む。ブロック間スイッチング素子40は、導体46（例えば、ワイヤ、トレースなど）並びにバッファ48及び50を含むことができる。一実施例においては、バッファ48及び50は、ブロック間スイッチング素子40とやり取りする信号の接続及びタイミングを制御するために含まれる。さらに下記するように、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するために提供され、一方で、バッファ50は、ブロック間スイッチング素子40間で送信されるデータをバッファするために提供することができる。さらに、ブロック32は、信号（例えば、データ）を受信し、ブロック32にデータを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できる。ブロック32は、ブロック32からの信号を外部デバイス（例えば、別のFSMラチス30）に提供するための出力ブロック54（例えば、出力ポート）にも選択的に結合することができる。FSMラチス30は、FSMラチス30を構成（例えば、画像を通じたプログラム）するためのプログラミングインターフェイス56も含むことができる。画像は、SME34、36の状態を構成（例えば、設定）することができる。すなわち、画像は、入力ブロック52において与えられた入力に、ある方法で反応するようにSME34、36を構成することができる。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、ハイ信号を出力するように設定できる。

30

## 【0019】

一実施例においては、入力ブロック52、出力ブロック54及び／またはプログラミングインターフェイス56は、レジスタとして実現することができ、レジスタへの書き込みまたはレジスタからの読み出しによってそれぞれの素子にまたはそれらからデータを提供する。したがって、プログラミングインターフェイス56に対応するレジスタに格納され

40

50

た画像からのビットは、SME34、36上でロードすることができる。図2は、ブロック32、入力ブロック52、出力ブロック54及びブロック間スイッチング素子40間にある数の導体（例えば、ワイヤ、トレース）を図示しているが、他の実施例においては、より少ないまたは多い数の導体を使用できることが理解されるはずである。

#### 【0020】

図3は、ブロック32の一実施例を図示する。ブロック32は、プログラマブルブロック内スイッチング素子42とともに選択的に結合することができる複数の行38を含むことができる。さらに、行38は、ブロック間スイッチング素子40によって、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できる。行38は、2のグループ（GOT）60として本明細書に参照される素子の対に組織化された複数のSME34、36を含む。一実施例においては、ブロック32は、16個（16）の行38を含む。

#### 【0021】

図4は、行38の一実施例を図示する。GOT60は、プログラマブル行内スイッチング素子44によって、行38内の他のGOT60及び任意の他の素子（例えば、専用素子58）に選択的に結合することができる。GOT60は、ブロック内スイッチング素子42によって他の行38における他のGOT60、またはブロック間スイッチング素子40によって他のブロック32における他のGOT60にも結合することができる。一実施例においては、GOT60は、第一の入力62、第二の入力64及び出力66を有する。図5を参照してさらに図示するように、第一の入力62は、GOT60の第一のSME34に結合され、第二の入力64は、GOT60の第二のSME36に結合される。

#### 【0022】

一実施例においては、行38は、第一及び第二の複数の行相互接続導体68、70を含む。一実施例においては、GOT60の入力62、64は、1つ以上の行相互接続導体68、70に結合し、出力66も、1つ以上の行相互接続導体68、70に結合することができる。一実施例においては、第一の複数の行相互接続導体68は、行38内の各GOT60の各SME34、36に結合することができる。第二の複数の行相互接続導体70は、行38内の各GOT60のうちの一つのSME34、36に結合することができるが、GOT60の他のSME34、36には結合することができない。一実施例においては、図5によりよく図示するように、第二の複数の行相互接続導体70のうちの第一の半分が、行38内のSME34、36のうちの第一の半分（各GOT60からの一方のSME34）に結合することができ、第二の複数の行相互接続導体70のうちの第二の半分が、行38内のSME34、36のうちの第二の半分（各GOT60からの他方のSME34、36）に結合することができる。第二の複数の行相互接続導体70とSME34、36との間の限定された接続は、本明細書においては、「パリティ」と称される。一実施例においては、行38は、カウンタ、プログラマブルブール型論理素子、ルックアップテーブル、RAM、フィールドプログラマブルゲートアレイ（FPGA）、特定用途向け集積回路（ASIC）、プログラマブルプロセッサ（例えば、マイクロプロセッサ）、または専用機能を実施するための他の素子などの専用素子58も含むことができる。

#### 【0023】

一実施例においては、専用素子58は、（本明細書においては、カウンタ58とも称される）カウンタを含む。一実施例においては、カウンタ58は12ビットプログラマブルダウンカウンタを含む。12ビットプログラマブルカウンタ58は、カウンティング入力、リセット入力及びゼロカウント出力を有する。カウンティング入力は、アサートされると、1ずつカウンタ58の値をデクリメントする。リセット入力は、アサートされると、関連するレジスタからの初期値をカウンタ58にロードさせる。12ビットカウンタ58に対して、12ビットの数までを初期値としてロードすることができる。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つのモードであるパルス及びホールドも有する。カウンタ58がパルスモードに設定されると、カウンタ58がゼロ及びクロックサイクルに達したときに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中に、

ゼロカウント出力がアサートされる。その結果、カウンタ58は、時間内にクロックサイクルからオフセットされる。次のクロックサイクルにおいて、ゼロカウント出力はもはやアサートされない。カウンタ58がホールドモードに設定されると、カウンタ58がゼロにデクリメントするとき、クロックサイクル中にゼロカウント出力がアサートされ、アサートされたリセット入力によってカウンタ58がリセットされるまでアサートされたままである。

#### 【0024】

別の実施例においては、専用素子58は、プール論理を含む。例えば、このプール論理は、AND、OR、NAND、NOR、積和(SOP)、積和の否定出力(NSOP)、和積の否定出力(NPOS)、及び和積(PoS)関数などの論理関数を実行するのに使用してもよい。このプール論理を使用して、FSMラチス30における(本明細書中に後に論じるような、FSMの端末ノードに対応する)端末状態SMEからのデータを抽出できる。抽出されたデータを使用して、状態データを他のFSMラチス30に提供する、及び/またはFSMラチス30の再構成もしくは別のFSMラチス30の再構成に使用される構成データを提供することができる。

#### 【0025】

図5は、GOT60の一実施例を図示する。GOT60は、入力62、64並びにORゲート76及び3:1マルチプレクサ78に結合されたそれらの出力72、74を有する第一のSME34及び第二のSME36を含む。3:1マルチプレクサ78は、第一のSME34、第二のSME36、またはORゲート76のいずれかにGOT60の出力66を結合するように設定することができる。ORゲート76を使用して、GOT60の共通出力66を形成するために双方の出力72、74をともに結合することができる。一実施例においては、第一及び第二のSME34、36が前述のようにパリティを示し、第一のSME34の入力62は行相互接続導体68の幾つかに結合することができ、第二のSME36の入力64が他の行相互接続導体70に結合できるによって、パリティ問題を克服できる共通出力66を生成することができる。一実施例においては、GOT60内の2つのSME34、36は、カスケード接続されるか、かつ/またはスイッチング素子79のいずれかもしくは双方を設定することによってそれ自身にループバックさせることができる。SME34、36の出力72、74を他のSME34、36の入力62、64に結合することによって、SME34、36をカスケード接続することができる。出力72、74をそれ自身の入力62、64に結合することによって、SME34、36をそれ自身にループバックさせることができる。したがって、第一のSME34の出力72は、第一のSME34の入力62及び第二のSME36の入力64のいずれにも結合しないか、またはいずれかもしくは双方に結合させることができる。

#### 【0026】

一実施例においては、ステートマシン素子34、36は、検出ライン82にパラレルに結合された、ダイナミックランダムアクセスメモリ(DRAM)でしばしば使用されるような複数のメモリセル80を備える。当該メモリセル80のうちの1つは、ハイまたはローの値(例えば、1または0)のいずれかに対応するようなデータ状態に設定することができるメモリセルを含む。メモリセル80の出力は、検出ライン82に結合し、メモリセル80の入力は、データストリームライン84上のデータに基づいて信号を受信する。一実施例においては、入力ブロック52における入力は、メモリセル80のうちの1つ以上を選択するようにデコードされる。選択されたメモリセル80は、検出ライン82上の出力としてその格納されたデータ状態を提供する。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)に提供でき、デコーダは、データストリームライン84のうちの1つ以上を選択できる。一実施例においては、デコーダは、256本のデータストリームライン84のうちの対応する1本へと、8ビットACSI文字を変換することができる。

#### 【0027】

メモリセル80は、したがって、メモリセル80がハイ値に設定されると検出ライン8

10

20

30

40

50

2にハイ信号を出力し、そして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。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がロウ値へと設定されるとき、メモリセル80は、検出ライン82にロウ信号を出力する。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によって検知される。

#### 【0028】

一実施例においては、入力ライン62、64上の信号は、アクティブまたは非アクティブ状態のいずれかにそれぞれの検出セル86を設定する。非アクティブ状態に設定されると、検出セル86は、それぞれの検出ライン82上の信号にかかわらず、それぞれの出力72、74にロウ信号を出力する。アクティブ状態に設定されると、検出セル86は、ハイ信号がそれぞれのSME34、36のメモリセル82のうちの1つから検出されたときに、それぞれの出力ライン72、74にハイ信号を出力する。アクティブ状態においては、それぞれのSME34、36のメモリセル82の全てからの信号がロウであるとき、検出セル86は、それぞれの出力ライン72、74上にロウ信号を出力する。

10

#### 【0029】

一実施例においては、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ライン84に結合される。したがって、SME34、36は、データストリームライン84の選択された1つ以上がその上にハイ信号を有するときにハイ信号を出力するようにプログラムすることができる。例えば、SME34は、一番目のメモリセル80（例えば、ビット0）をハイに設定し、全ての他のメモリセル80（例えば、ビット1～255）をロウに設定することができる。それぞれの検出セル86がアクティブ状態にあるとき、ビット0に対応するデータストリームライン84がその上にハイ信号を有すると、SME34は、出力72上にハイ信号を出力する。他の実施例においては、複数のデータストリームライン84のうちの1つが、適切なメモリセル80をハイ値に設定することによってその上にハイ信号を有すると、SME34は、ハイ信号を出力するように設定することができる。

20

#### 【0030】

一実施例においては、メモリセル80は、関連するレジスタからのビットを読み出すことによって、ハイ値またはロウ値に設定することができる。したがって、SME34は、コンパイラ20によって生成された画像をレジスタに格納して、レジスタ内のビットを関連するメモリセル80にロードすることによって構成することができる。一実施例においては、コンパイラ20によって生成された画像は、ハイ及びロウ（例えば、1及び0）ビットのバイナリ画像を含む。画像は、SME34、36をカスケード接続することによって、FSMとして動作させるようにFSMラチス30を構成することができる。例えば、検出セル86をアクティブ状態に設定することによって、第一のSME34をアクティブ状態に設定することができる。第一のSME34は、ビット0に対応するデータストリームライン84がその上にハイ信号を有するとき、ハイ信号を出力するように設定することができる。第二のSME36は、最初に非アクティブ状態に設定することができるが、アクティブ状態時に、ビット1に対応するデータストリームライン84がその上にハイ信号を有するときにハイ信号を出力するように設定することができる。第一のSME34及び第二のSME36は、第一のSME34の出力72を第二のSME36の入力64に結合することによって、カスケード接続することができる。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上にハイ信号が検知されると、第一のSME34は、出力72上にハイ信号を出力して、第二の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ライン84上にハイ信号が検知されると、第二のSME36は、別のSME36をアクティブ化するために、またはFSMラチス30からの出力用に、出力74上にハイ信号を出力する。

30

#### 【0031】

一実施例においては、单一のFSMラチス30は、单一の物理デバイス上に実現されるが、他の実施例においては、2つ以上のFSMラチス30を单一の物理デバイス（例えば

40

50

、物理チップ)上に実現してもよい。一実施例においては、各FSMラチス30は、別個のデータ入力ブロック52、別個の出力ブロック54、別個のプログラミングインターフェイス56、及び別個のコンフィギュラブル素子の組を含むことができる。さらに、コンフィギュラブル素子の各組は、それらの対応するデータ入力ブロック52におけるデータに対して反応(例えば、ハイまたはロウ信号を出力)することができる。例えば、第一のFSMラチス30に対応する第一の組のコンフィギュラブル素子は、第一のFSMラチス30に対応する第一のデータ入力ブロック52におけるデータに対して反応することができる。第二のFSMラチス30に対応する第二の組のコンフィギュラブル素子は、第二のFSMラチス30に対応する第二のデータ入力ブロック52に対して反応することができる。したがって、各FSMラチス30は、一組のコンフィギュラブル素子を含み、異なる組のコンフィギュラブル素子は、異なる入力データに対して反応することができる。同様に、各FSMラチス30及び各対応する組のコンフィギュラブル素子は、個別出力を提供することができる。幾つかの実施例においては、第一のFSMラチス30からの出力ブロック54は、第二のFSMラチス30の入力ブロック52に結合することができ、この構成により第二のFSMラチス30用の入力データが、一連のFSMラチス30のヒエラルキー配列における第一のFSMラチス30からの出力データを含むことができる。10

#### 【0032】

一実施例においては、FSMラチス30上にロードするための画像は、FSMラチス30におけるコンフィギュラブル素子、プログラマブルスイッチング素子及び専用素子の構成用の複数ビットのデータを含む。一実施例においては、画像は、ある入力に基づいて所望の出力を提供するFSMラチス30を構成するように、FSMラチス30上にロードすることができる。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して、コンフィギュラブル素子の反応に基づいて、FSMラチス30からの出力を提供することができる。出力ブロック54からの出力は、任意のパターンとのマッチを示す単一ビット、複数のパターンとのマッチ及び不一致を示す複数ビットを含む語、並びに任意の瞬間ににおける全てまたはあるコンフィギュラブル素子の状態に対応する状態ベクトルを含むことができる。記述するように、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、クリプトグラフィーなどのデータ解析を実施するために、多数のFSMラチス30をステートマシンエンジン14などのステートマシンエンジン内に含ませてもよい。20

#### 【0033】

図6は、FSMラチス30によって実現できる有限ステートマシン(FSM)の一実施例のモデルを図示する。FSMラチス30は、FSMの物理的実装として構成(例えば、プログラム)することができる。FSMは、1つ以上のルートノード92を含有する図形90(例えば、有向グラフ、無向グラフ、擬グラフ)として表すことができる。ルートノード92に加えて、FSMは、幾つかの標準ノード94と、1つ以上の辺98を通してルートノード92及び他の標準ノード94に接続される端末ノード96からなることができる。ノード92、94、96は、FSMにおける状態に対応する。辺98は、状態間の遷移に対応する。30

#### 【0034】

各ノード92、94、96は、アクティブ状態であってもよいし、非アクティブ状態であってもよい。非アクティブ状態にあるとき、ノード92、94、96は入力データに対して反応(例えば、応答)しない。アクティブ状態にあるとき、ノード92、94、96は入力データに対して反応することができる。上流ノード92、94は、上流ノード92、94と下流ノード94、96との間の辺98によって特定される基準に入力データがマッチするときに、そのノードから下流にあるノード94、96をアクティブ化することによって入力データに対して反応することができる。例えば、第一のノード94がアクティブであり、かつ文字「b」が入力データとして受信されるとき、文字「b」を特定する第一のノード94は、辺98によって第一のノード94に接続された第二のノード94をアクティブ化する。本明細書中に使用される「上流」は、1つ以上のノード間の関係を指し40

、他の1つ以上のノードの上流（または、ループもしくはフィードバック構成の場合には、それ自身の上流）である第一のノードとは、その第一のノードが他の1つ以上のノードをアクティブ化できる（または、ループの場合にはそれ自身をアクティブ化できる）状況を指す。同様に、「下流」とは、他の1つ以上のノードの下流（または、ループの場合にはそれ自身の下流）にある第一のノードが他の1つ以上のノードによってアクティブ化できる（または、ループの場合にはそれ自身によってアクティブ化できる）関係を指す。したがって、「上流」及び「下流」という用語は、本明細書中においては、1つ以上のノード間の関係を指すために使用されるが、これらの用語は、ループまたはノード間の他の非線形経路の使用を排除することはない。

## 【0035】

10

図形90においては、ルートノード92は、最初にアクティブ化することができ、入力データがルートノード92からの辺98とマッチするとき、下流ノード94をアクティブ化することができる。ノード94は、ノード94からの辺98に入力データがマッチするとき、ノード96をアクティブ化することができる。図形90中のノード94、96は、入力データが受信されると、この方法でアクティブ化することができる。端末ノード96は、入力データ内の興味あるシーケンスのマッチに対応する。したがって、端末ノード96のアクティブ化は、入力データとして、興味あるシーケンスが受信されたことを示す。パターン認識機能を実現するFSMラチス30の状況においては、端末ノード96への到着が、興味ある特定のパターンが入力データにおいて検出されたことを示すことができる。

20

## 【0036】

一実施例においては、各ルートノード92、標準ノード94及び端末ノード96は、FSMラチス30におけるコンフィギュラブル素子に対応することができる。各辺98は、コンフィギュラブル素子間の接続に対応することができる。したがって、別の標準ノード94または端末ノード96に遷移する（例えば、それらに接続する辺98を有する）標準ノード94は、別のコンフィギュラブル素子に遷移する（例えば、それに出力を提供する）コンフィギュラブル素子に対応する。幾つかの実施例においては、ルートノード92は、対応するコンフィギュラブル素子を有さない。

## 【0037】

30

認識されるように、ノード92をルートノードとして記述し、ノード96を端末ノードとして記述したが、必ずしも特定の「開始」またはルートノードが存在しなくてもよく、必ずしも特定の「終了」または出力ノードが存在しなくてもよい。換言すると、任意のノードを開始点にし、任意のノードが出力を提供してもよい。

## 【0038】

FSMラチス30がプログラムされると、各コンフィギュラブル素子は、アクティブ状態であってもよいし、非アクティブ状態であってもよい。任意のコンフィギュラブル素子は、非アクティブなとき、対応するデータ入力ブロック52において入力データに対して反応しない。アクティブなコンフィギュラブル素子は、データ入力ブロック52において入力データに対して反応することができ、入力データがコンフィギュラブル素子の設定にマッチするとき、下流コンフィギュラブル素子をアクティブ化することができる。コンフィギュラブル素子が端末ノード96に対応するとき、コンフィギュラブル素子は、外部デバイスにマッチの指示を提供するために、出力ブロック54に結合することができる。

40

## 【0039】

プログラミングインターフェイス56を介してFSMラチス30上にロードされた画像は、コンフィギュラブル素子と専用素子との間の接続と同様、コンフィギュラブル素子及び専用素子を構成することができ、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、連続的なノードのアクティブ化を通して所望のFSMが実現される。一実施例においては、コンフィギュラブル素子は、単一のデータサイクル（例えば、単一の文字、一組の文字、単一のクロックサイクル）中にアクティブなままであり、その後、上流コンフィギュラブル素子によって再アクティブ化されるまでは非アクティブとなる。

50

## 【0040】

端末ノード96は、過去のイベントの圧縮されたヒストリを格納すると考慮できる。例えば、端末ノード96に到達することを必要とされる1つ以上のパターンの入力データを、その端末ノード96のアクティブ化によって表すことができる。一実施例においては、端末ノード96によって提供される出力はバイナリであって、つまり、その出力は、興味あるパターンがマッチしたか否かを示す。図形90における標準ノード94に対する端末ノード96の割合は、非常に小さくてもよい。換言すると、FSMには高い複雑性が存在することがあるが、FSMの出力は、比較的に小さく成り得る。

## 【0041】

一実施例においては、FSMラチス30の出力は、状態ベクトルを含むことができる。  
状態ベクトルは、FSMラチス30のコンフィギュラブル素子の状態（例えば、アクティブ化されているか否か）を含む。別の実施例においては、コンフィギュラブル素子が端末ノード96に対応するか否かにかかわらず、状態ベクトルは、コンフィギュラブル素子の全てまたはサブセットの状態を含むことができる。一実施例においては、状態ベクトルは、端末ノード96に対応するコンフィギュラブル素子の状態を含む。したがって、出力は、図形90の全端末ノード96によって提供される指示の集合を含むことができる。状態ベクトルは、単語として表すことができ、各端末ノード96によって提供されるバイナリ指示は一ビットの単語を含む。端末ノード96のこのエンコードは、FSMラチス30に対する検出状態の有効な指示（例えば、興味あるシーケンスが検出されたか否か、検出された興味あるシーケンスは何か）を提供することができる。

10

20

## 【0042】

前述のように、FSMラチス30は、パターン認識機能を実現するようにプログラムすることができる。例えば、FSMラチス30は入力データにおける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できる。興味あるデータシーケンスがFSMラチス30によって認識されると、当該認識の指示を出力ブロック54において提供することができる。一実施例においては、パターン認識は、シンボルのストリング（例えば、ASCII文字）を、例えば、ネットワークデータにおけるマルウェアまたは他のデータを識別するために認識することができる。

## 【0043】

図7は、ヒエラルキー構造100の一実施例を図示し、ここでは、FSMラチス30の二レベルが直列に結合され、データを解析するために使用される。つまり、図示する実施形態においては、ヒエラルキー構造100は、直列に配列された第一のFSMラチス30A及び第二のFSMラチス30Bを含む。各FSMラチス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、構成信号を受信するためのプログラミングインターフェイスブロック56、及び出力ブロック54を含む。

30

## 【0044】

第一のFSMラチス30Aは、データ入力ブロックにおいて、生データなどの入力データを受信するように構成される。第一のFSMラチス30Aは、前述のように入力データに対して反応して、出力ブロックにおいて出力を提供する。第一のFSMラチス30Aからの出力は、第二のFSMラチス30Bのデータ入力ブロックに送信される。第二のFSMラチス30Bは、その後、第一のFSMラチス30Aによって提供された出力に基づいて反応して、ヒエラルキー構造100の対応する出力信号102を提供することができる。直列の2つのFSMラチス30A及び30Bのこのヒエラルキー結合は、第一のFSMラチス30Aから第二のFSMラチス30Bに圧縮された単語における過去のイベントに関するデータを提供するための手段を提供する。提供されたデータは、事実上、第一のFSMラチス30Aによって記録された複雑なイベント（例えば、興味あるシーケンス）の要約であり得る。

40

## 【0045】

図7に示すFSMラチス30A、30Bの2レベルヒエラルキー100は、2つの独立したプログラムを同一のデータストリームに基づいて動作させることができる。二段階ヒ

50

エラルキーは、異なる領域としてモデル化された生物学的脳における視認と同様であり得る。このモデルの下で、領域は、有効に異なるパターン認識エンジンであり、各々が類似する計算機能（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）を利用する。複数のFSMラチス30A、30Bをともに接続することによって、データストリーム入力についてのより多い知識が得られることがある。

#### 【0046】

（第一のFSMラチス30Aによって実現される）ヒエラルキーの第一のレベルは、例えば、生データストリームにおいて直接的に処理を実施することができる。すなわち、生データストリームを第一のFSMラチス30Aの入力ブロック52で受信することができる、第一のFSMラチス30Aのコンフィギュラブル素子が、生データストリームに反応することができる。（第二のFSMラチス30Bによって実現される）ヒエラルキーの第二のレベルは、第一のレベルからの出力を処理することができる。すなわち、第二のFSMラチス30Bは、第一のFSMラチス30Aの出力ブロック54からの出力を、第二のFSMラチス30Bの入力ブロック52で受信し、第二のFSMラチス30Bのコンフィギュラブル素子が、第一のFSMラチス30Aの出力に対して反応することができる。したがって、この実施例においては、第二のFSMラチス30Bは、入力として生データストリームは受信しないが、その代りに、第一のFSMラチス30Aによって判定された、生データストリームとマッチする興味あるパターンの指示を受信する。第二のFSMラチス30Bは、第一のFSMラチス30Aからの出力データストリームにおけるパターンを認識するFSMを実現することができる。第二のFSMラチス30Bが、FSMラチス30Aからの出力を受信することに加えて、複数の他のFSMラチスからの入力も受信できることが認識されるはずである。同じように、第二のFSMラチス30Bは、他のデバイスからの入力を受信できる。第二のFSMラチス30Bは、これらの複数の入力を組み合わせて出力を生成することができる。

10

20

30

#### 【0047】

図8は、コンパイラがFSMを実現するためのラチス30などのFSMラチスを構成するために使用される画像にソースコードを変換するための方法110の一実施例を図示する。方法110は、ソースコードをシンタックスツリーに解析すること（ブロック112）と、シンタッ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、ハードウェア上にネットリストを配置すること（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、結果生じた画像を発行すること（ブロック124）と、を含む。

30

#### 【0048】

一実施例においては、コンパイラ20は、FSMラチス30上のFSMを実現するための、ソフトウェア開発者が画像を生成することを可能にするアプリケーションプログラミングインターフェス（API）を含む。コンパイラ20は、FSMラチス30を構成するように構成された画像にソースコードにおける正規表現の入力セットを変換するための方法を提供する。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現することができる。これらの命令は、コンパイラ20の機能をコンピュータ上のプロセッサ12に実現させることができる。例えば、命令は、プロセッサ12によって実行されると、プロセッサ12に対して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及び124に記述された動作をプロセッサ12に実施させることができる。

40

#### 【0049】

一実施例においては、ソースコードは、シンボルのグループ内のシンボルのパターンを識別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ために、ソースコードは複数の正規表現（regular expression）を含むことができる。Regular expressionは、シンボル検索パターンを記述するためのストリングであってもよい。Regular expression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、種々のコンピュータドメインで

50

広く使用されている。一実施例においては、コンパイラによってサポートされる正規表現は、非構造化データの解析用の基準を含む。非構造化データは、フリーフォームであるデータを含むことができ、データ内の単語に対して適用される索引付けを有さない。単語は、データ内の印刷可能及び印字不能な任意の組み合わせのバイトを含むことができる。一実施例においては、コンパイラは、Perl（例えば、Perl互換正規表現（PCRE））、PHP、Java、及びNET言語を含むregular expressionを実現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できる。

#### 【0050】

ブロック112において、コンパイラ20は、相関的に接続されたオペレータの配列を形成するためにソースコードを解析することができ、異なるタイプのオペレータはソースコードによって実現される異なる機能（例えば、ソースコード内のregular expressionによって実現される異なる機能）に対応する。ソースコードの解析によって、ソースコードのジェネリック表現を生成できる。一実施例においては、ジェネリック表現は、シンタックスツリーとして知られるツリーグラフの形式のソースコードにおけるregular expressionのエンコードされた表現を含む。本明細書に記述する実施例は、（「抽象シンタックスツリー」としても知られる）シンタックスツリーとしての配列を参照する。しかしながら、他の実施例においては、具象シンタックスツリーまたは他の配列を使用することができる。

#### 【0051】

前述のように、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ため、言語にかかわらず、解析によって、非言語の特定の表現、例えば、シンタックスツリーへとソースコードを変換する。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にかかわらず、共通の入力構造から動作することができる。

#### 【0052】

前述のように、シンタックスツリーは、相関的に接続された複数のオペレータを含む。シンタックスツリーは、複数の異なるタイプのオペレータを含むことができる。すなわち、異なるオペレータが、ソースコードにおけるregular expressionによって実現される異なる機能に対応することができる。

#### 【0053】

ブロック114において、シンタッ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を含み、したがって、決定論的または非決定論的として分類することができる。決定論的オートマトンは、所定の時間における単一の実行経路を有し、一方で、非決定論的オートマトンは、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。シンタックスツリーをオートマトンに変換するために、シンタックスツリーにおけるオペレータ及びオペレータ間の関係は、状態間の遷移を伴う状態に変換される。一実施例においては、オートマトンは、FSMラチス30のハードウェアに部分的に基づいて変換することができる。

#### 【0054】

一実施例においては、オートマトン用の入力シンボルは、アルファベット、0～9の数字及び他の印刷可能な文字のシンボルを含む。一実施例においては、入力シンボルは、0～255（その数値も含む）バイト値によって表される。一実施例においては、オートマトンは、グラフのノードが状態の組に対応する有向グラフとして表すことができる。一実施例においては、入力シンボル上の状態pから状態qへの遷移、すなわち、(p、)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。一実施例においては、オートマトンの反転によって、幾つかのシンボルにおける各遷移p→qが、同一シンボルにおいてq→pに反転される新規オートマトンが生成される。反転においては、開始状態が最終状態になり、そして最終状態が開始状態になる。一実施例においては、オートマトンによって認識される（例えば、マッチする）言語は、オートマトンに順次入力されると最終状態に到達する全ての可能性のある文字ストリングの組である。オートマトンによって認識される言語における各ストリングは、開始状態から1つ以上の最終状態への経路を

10

20

30

40

50

追跡する。

**【0055】**

ブロック116においては、オートマトンが構成された後、オートマトンは、とりわけ、その複雑性及びサイズを減少させるために最適化される。オートマトンは、余分な状態を組み合わせることによって最適化することができる。

**【0056】**

ブロック118においては、最適化されたオートマトンが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においては、FSMラチス30上のハードウェア素子（例えば、SME34、36、他の素子）へのオートマトンの各状態をマップして、ハードウェア素子間の接続を判定する。

10

**【0057】**

ブロック120においては、ネットリストが、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（例えば、SME34、36、専用素子58）の特定のハードウェア素子を選択するために配置される。一実施例においては、この配置によって、FSMラチス30の一般的な入力及び出力制約に基づいて、各特定のハードウェア素子が選択される。

**【0058】**

ブロック122においては、配置されたネットリストが、ネットリストによる接続の記述を達成するように選択されたハードウェア素子とともに結合するために、プログラマブルスイッチング素子（例えば、ブロック間スイッチング素子40、ブロック内スイッチング素子42及び行内スイッチング素子44）用の設定を判定するためにルーティングされる。一実施例においては、プログラマブルスイッチング素子の設定は、選択されたハードウェア素子を接続するのに使用されるFSMラチス30の特定の導体、及びプログラマブルスイッチング素子用の設定を判定することによって判定される。ルーティングにおいては、ブロック120において配置されたハードウェア素子間の接続のより特定な制限を考慮することができる。したがって、ルーティングにおいては、FSMラチス30上の導体の実際の制限がある場合でも適切な接続をするために、グローバル配置によって判定されたように、ハードウェア素子の幾つかの位置を調整してもよい。

20

**【0059】**

一旦ネットリストが配置されルーティングされると、配置されてルーティングされたネットリストを、FSMラチス30の構成用の複数ビットに変換することができる。複数ビットは、本明細書においては画像（例えば、バイナリ画像）とも称される。

30

**【0060】**

ブロック124においては、画像がコンパイラ20によって発行される。画像は、FSMラチス30の特定のハードウェア素子の構成用の複数ビットを含む。プログラムされたFSMラチス30がソースコードによって記述された機能を有するFSMを実現するように、ビットは、FSMラチス30上にロードし、SME34、36、専用素子58、及びプログラマブルスイッチング素子の状態を構成することができる。配置（ブロック120）及びルーティング（ブロック122）では、オートマトンにおける特定の状態へとFSMラチス30における特定の位置の特定のハードウェア素子をマップすることができる。したがって、画像におけるビットは、所望の機能（複数可）を実現するために、特定のハードウェア素子を構成することができる。一実施例においては、画像は、コンピュータ可読媒体にマシンコードを保存することによって発行することができる。別の実施例においては、画像は、ディスプレイデバイス上に画像を表示することによって発行することができる。さらに別の実施例においては、画像は、FSMラチス30上に画像をロードするための構成デバイスなどの別のデバイスに画像を送信することによって発行することができる。さらに別の実施例においては、画像はFSMラチス（例えば、FSMラチス30）上に画像をロードすることによって発行することができる。

40

**【0061】**

一実施例においては、画像は、SME34、36及び他のハードウェア素子に画像からのビット値を直接ロードするか、または1つ以上のレジスタに画像をロードして、その後

50

、S M E 3 4 、3 6 及び他のハードウェア素子にレジスタからのビット値を書き込むことのいずれかによって、F S M ラチス 3 0 上にロードすることができる。一実施例においては、構成デバイス及び／またはコンピュータが1つ以上のメモリアドレスに画像を書き込むことによって、F S M ラチス 3 0 上に画像をロードできるように、F S M ラチス 3 0 のハードウェア素子（例えば、S M E 3 4 、3 6 、専用素子 5 8 、プログラマブルスイッチング素子 4 0 、4 2 、4 4 ）は、マップされたメモリである。

#### 【 0 0 6 2 】

本明細書に記述する方法の実施例は、少なくとも部分的にマシンまたはコンピュータで実現することができる。幾つかの実施例は、前述の実施例に記述したような方法を実施するように電子デバイスを構成するように動作可能な命令がエンコードされたコンピュータ可読媒体またはマシン可読媒体を含むことができる。当該方法の実現では、マイクロコード、アセンブリ言語コード、より高いレベルの言語コードなどのコードを含むことができる。当該コードは、種々の方法を実施するためのコンピュータ可読命令を含むことができる。コードは、コンピュータプログラム製品の一部を形成してもよい。さらに、コードは、実行中または他の時間中に、1つ以上の揮発性または不揮発性コンピュータ可読媒体に具体的に格納されてもよい。これらのコンピュータ可読媒体は、これらに限定されないが、ハードディスク、リムーバブル磁気ディスク、リムーバブル光ディスク（例えば、コンパクトディスク及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イック、ランダムアクセスメモリ（R A M ）、リードオンリメモリ（R O M ）などを含むことができる。

10

#### 【 0 0 6 3 】

ここで図9を参照し、ステートマシンエンジン14（例えば、单一チップ上の单一デバイス）の実施形態を図示する。前述のように、ステートマシンエンジン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。図示する実施形態においては、データを、ダブルデータレートスリー（D D R 3 ）バスインターフェイス130などのバスインターフェイスを通して、ステートマシンエンジン14に送信できる。D D R 3 バスインターフェイス130は、1 G バイト／秒以上の速度でデータを交換（例えば、提供及び受信）可能であってもよい。そのようなデータ交換速度は、ステートマシンエンジン14によってデータが解析される速度よりも大きくてよい。認識されるように、解析されるべきデータのソースに応じて、バスインターフェイス130は、N A N D フラッシュインターフェイス、ペリフェラルコンポーネントインターフェイス（P C I ）インターフェイス、ギガビットメディア独立型インターフェイス（G M I I ）などの、ステートマシンエンジン14とデータソースとでデータをやり取りするための任意の適切なバスインターフェイスであってもよい。前述のように、ステートマシンエンジン14は、データを解析するように構成された1つ以上のF S M ラチス 3 0 を含む。各F S M ラチス 3 0 は、2つのハーフラチスに分割されてもよい。図示する実施形態においては、ラチス30が48K S M E を含むように、各ハーフラチスは、24K S M E （例えば、S M E 3 4 、3 6 ）を含んでもよい。ラチス30は、図2～図5に関して前述したように配列された任意の所望の数のS M E を含んでもよい。さらに、唯一のF S M ラチス 3 0 を図示したが、前述のように、ステートマシンエンジン14は複数のF S M ラチス 3 0 を含んでもよい。

20

#### 【 0 0 6 4 】

解析されるべきデータは、バスインターフェイス130で受信され、多数のバッファ及びバッファインターフェイスを通してF S M ラチス 3 0 へと提供することができる。図示する実施形態においては、データ経路は、データバッファ132、命令バッファ133、プロセスバッファ134、並びにインターランク（I R ）バス及びプロセスバッファインターフェイス136を含む。データバッファ132は、解析されるべきデータを受信して一時的に格納するように構成される。一実施形態においては、2つのデータバッファ132（データバッファA及びデータバッファB）が存在する。データは、2つのデータバッファ132のうちのいずれかに格納することができ、一方で、F S M ラチス 3 0 による解

30

40

50

析のために、他方のデータバッファ 132 からデータは除かれている。バスインターフェイス 130 は、データバッファ 132 が満たされるまで、解析されるべきデータをデータバッファ 132 に提供するように構成できる。バスインターフェイス 130 は、データバッファ 132 が満たされた後、他の目的（例えば、データバッファ 132 が解析されるべきさらなるデータの受信に利用可能になるまで、データストリームから他のデータを提供するため）に自由に使用できるように構成することができる。図示する実施形態においては、データバッファ 132 は、それぞれ、32K バイトであってもよいが、他の実施形態においては、データバッファ 132 は、任意の適切なサイズ（例えば、4K バイト、8K バイト、16K バイト、64K バイトなど）であってもよい。命令バッファ 133 は、バスインターフェイス 130 を通じてプロセッサ 12 からの命令、例えば、解析されるべきデータに対応する命令、及びステートマシンエンジン 14 の構成に対応する命令を受信するように構成される。IR バス及びプロセスバッファインターフェイス 136 は、プロセスバッファ 134 へのデータ提供を容易にすることによって、データが FSM ラチス 30 によって順番に処理されることを保証できる。IR バス及びプロセスバッファインターフェイス 136 は、データが正確な順序で受信されて解析されるように、データの交換、タイミングデータ、パッキング命令などを調整することができる。概して、IR バス及びプロセスバッファインターフェイス 136 は、複数のデバイスをランクにおけるデバイスとして使用することができる。ランクにおけるデバイスである複数のデバイスは、複数のデバイスの全てが全ての共有データ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、複数の物理デバイス（例えば、ステートマシンエンジン 14、チップ、個々のデバイス）は、ランクに配列され、IR バス及びプロセスバッファインターフェイス 136 を通じて互いにデータを提供することができる。この用途のための「ランク」という用語は、同一のチップセレクトに接続されたステートマシンエンジン 14 の組を指す。図示する実施形態においては、IR バス及びプロセスバッファインターフェイス 136 は、8 ビットデータバスを含むことができる。10

#### 【0065】

図示する実施形態においては、ステートマシンエンジン 14 は、ステートマシンエンジン 14 を通したデータの提供を援助するためのデコンプレッサ 138 及びコンプレッサ 140 も含む。認識できるように、コンプレッサ 140 及びデコンプレッサ 138 は、ソフトウェア及び／またはハードウェア設計を単純化するための同一の圧縮アルゴリズムを使用することができる。しかしながら、コンプレッサ 140 及びデコンプレッサ 138 は、異なるアルゴリズムを使用することもできる。データを圧縮することによって、バスインターフェイス 130（例えば、DDR3 バスインターフェイス）の利用時間を最小限化することができる。本実施形態においては、コンプレッサ 140 を使用して、状態ベクトルデータ、コンフィギュレーションデータ（例えば、プログラミングデータ）、及び FSM ラチス 30 による解析後に得られるマッチ結果データを圧縮できる。一実施形態においては、コンプレッサ 140 及びデコンプレッサ 138 へとかつ／またはそこから移動するデータが（例えば、圧縮でも解凍でも）変更されないように、コンプレッサ 140 及びデコンプレッサ 138 を無効にする（例えば、停止させる）ことができる。20

#### 【0066】

コンプレッサ 140 及びデコンプレッサ 138 は、複数の組のデータであって、各組のデータの長さが変化し得る、複数組のデータを処理するようにも構成することができる。圧縮データを「パディング」し、各々の圧縮領域が終了する時に該するインジケータを含ませることによって、コンプレッサ 140 は、ステートマシンエンジン 14 を通じた全体的な処理速度を改善することができる。40

#### 【0067】

ステートマシンエンジン 14 は状態ベクトルシステム 141 を含み、これは、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、及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50

システム 141 を使用して、FSM ラチス 30 の複数の状態ベクトルを格納し、状態ベクトルをステートマシンエンジン 14 にまたはそこから離すように移動し、そして状態ベクトルを FSM ラチス 30 に提供して、提供された状態ベクトルに対応する状態に FSM ラチス 30 を復元することができる。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 に一時的に格納することができる。すなわち、状態を復元して後にさらなる解析に使用することができるよう、各 SME 34、36 の状態を格納し、その一方で、新規のデータセット（例えば、検索語）の解析のために、SME 34、36 を使用されていない状態にすることができる。典型的なキャッシュメモリと同様に、状態ベクトルキャッシュメモリ 142 は、例えば、ここでは FSM ラチス 30 によって迅速検索及び使用されるように状態ベクトルを格納することができる。図示する実施形態においては、状態ベクトルキャッシュメモリ 142 は、最大 512 個の状態ベクトルを格納することができる。各状態ベクトルは、FSM ラチス 30 の SME 34、36 の状態（例えば、アクティブ化されているか否か）、及びカウンタ 58 の動的（例えば、最新の）カウントを含む。

#### 【0068】

認識されるように、状態ベクトルデータは、ランク内の異なるステートマシンエンジン 14（例えば、チップ）間でやり取りすることができる。状態ベクトルデータは、種々の目的のために異なるステートマシンエンジン 14 間でやり取りすることができる。この目的は、例えば、FSM ラチス 30 の SME 34、36 の状態とカウンタ 58 の動的カウントとの同期、複数のステートマシンエンジン 14 にわたる同一の機能の実行、複数のステートマシンエンジン 14 にわたる結果の再生成、複数のステートマシンエンジン 14 にわたる結果のカスケード接続、SME 34、36 の状態のヒストリ及び複数のステートマシンエンジン 14 を通じてカスケードされたデータを解析するために使用されるカウンタ 58 の動的カウントの格納などを含む。さらにまた、ステートマシンエンジン 14 内において状態ベクトルデータを使用して状態ベクトルを迅速に復元できることが留意されるべきである。例えば、状態ベクトルデータを使用して、SME 34、36 の状態及びカウンタ 58 の動的カウントを、（例えば、新規の検索語を検索するための）初期化状態に復元し、SME 34、36 の状態及びカウンタ 58 の動的カウントを、（例えば、以前に検索した検索語を検索するために）以前の状態に復元し、かつ SME 34、36 の状態及びカウンタ 58 の動的カウントを、（例えば、カスケード検索において検索語を検索するための）カスケード構成を構成するように変化させることができる。ある実施形態においては、（例えば、状態ベクトルデータの解析、状態ベクトルデータを修正するための再構成、状態ベクトルデータの効率を改善するための再構成などのために）状態ベクトルデータをプロセッサ 12 に提供することができるよう、状態ベクトルデータをバスインターフェイス 130 に提供することができる。

#### 【0069】

例えば、ある実施形態においては、ステートマシンエンジン 14 は、FSM ラチス 30 から外部デバイスに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 によって格納されたデータ）を提供することができる。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、修正した状態ベクトルデータを、FSM ラチス 30 を復元（例えば、リセッティング、初期化）するためにステートマシンエンジン 14 に提供することができる。したがって、外部デバイスは、所望のようにステートマシンエンジン 14 が状態をスキップ（例えば、ジャンプ）できるように、状態ベクトルデータを修正できる。

#### 【0070】

状態ベクトルキャッシュメモリ 142 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できる。例えば、状態ベクトルキャッシュメモリ 142 は、FSM ラチス 30、（例えば、IR バス及びプロセスバッファインターフェイス 136 を通じて）別の FSM ラチス 30、デコンプレッサ 138 などから状態ベクトルを受信することができる。図示する実施形態においては、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 を通じて他のデバイスから状態ベクトルを受信できる。さらにまた、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適切なデバイスに提供することができる。例えば、状態ベクトルキャッシュメモリ 142 は、状態ベクトルデータを、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、及び状態ベクトル中間出力バッファ 148 に提供することができる。

【0071】

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、及び状態ベクトル中間出力バッファ 148 などのさらなるバッファを、高速検索及び状態ベクトルの格納に適応させ、その一方で、ステートマシンエンジン 14 を通じてインターリープされたパケットを有する個々のデータセットを処理するように、状態ベクトルキャッシュメモリ 142 と併せて使用することができる。図示する実施形態においては、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、及び状態ベクトル中間出力バッファ 148 の各々は、1 つの状態ベクトルを一時的に格納するように構成されてもよい。状態ベクトルメモリバッファ 144 を使用して、任意の適切なデバイスから状態ベクトルデータを受信し、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ができる。例えば、状態ベクトルメモリバッファ 144 を使用して、FSM ラチス 30、( 例えれば、IR バス及びプロセスバッファインターフェイス 136 を通じて ) 別の FSM ラチス 30、デコンプレッサ 138、及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ことができる。別の実施例として、状態ベクトルメモリバッファ 144 を使用して、状態ベクトルデータを、( 例えれば、他の FSM ラチス 30 に向けて ) IR バス及びプロセスバッファインターフェイス 136、コンプレッサ 140、及び状態ベクトルキャッシュメモリ 142 に提供することができる。

【0072】

同じように、状態ベクトル中間入力バッファ 146 を使用して、任意の適切なデバイスから状態ベクトルデータを受信し、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ができる。例えば、状態ベクトル中間入力バッファ 146 を使用して、( 例えれば、IR バス及びプロセスバッファインターフェイス 136 を通じて ) FSM ラチス 30、デコンプレッサ 138、及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ことができる。別の実施例として、状態ベクトル中間入力バッファ 146 を使用して、状態ベクトルを FSM ラチス 30 に提供することができる。さらにまた、状態ベクトル中間出力バッファ 148 を使用して、任意の適切なデバイスから状態ベクトルを受信し、状態ベクトルを任意の適切なデバイスに提供することができる。例えれば、状態ベクトル中間出力バッファ 148 を使用して、FSM ラチス 30、及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ことができる。別の実施例として、状態ベクトル中間出力バッファ 148 を使用して、状態ベクトルを、( 例えれば、IR バス及びプロセスバッファインターフェイス 136 を通じて ) FSM ラチス 30、及びコンプレッサ 140 に提供することができる。

【0073】

一旦興味ある結果が FSM ラチス 30 によって生成されると、マッチ結果をマッチ結果メモリ 150 に格納することができる。例えれば、マッチ ( 例えれば、興味あるパターンの検出 ) を示す「マッチベクトル」を、マッチ結果メモリ 150 に格納することができる。マッチ結果を、その後、例えれば、バスインターフェイス 130 を通じてプロセッサ 12 に伝送するためにマッチバッファ 152 に送信することができる。前述のように、マッチ結果は圧縮されてもよい。

【0074】

さらなるレジスタ及びバッファを、同様にステートマシンエンジン 14 に提供してもよい。例えれば、ステートマシンエンジン 14 は、制御及びステータスレジスタ 154 を含んでもよい。さらに、復元及びプログラムバッファ 156 を、最初に FSM ラチス 30 の SME 34、36 を構成し、または解析中に FSM ラチス 30 における SME 34、36 の状態を復元するのに使用するために提供してもよい。同様に、保存及び修復マップバッファ 158 も、セットアップ及び使用のために保存及び修復マップを格納するために提供し

10

20

30

40

50

てもよい。

【0075】

図10は、デバイスがランクにおいて配列された複数の物理的なステートマシンエンジン14の一実施例を図示する。認識できるように、ステートマシンエンジン14とプロセッサ12との間のインターフェイス（例えば、DDR3バスインターフェイス130）によって、ステートマシンエンジン14は、プロセッサ12からの全てのデータ線を使用してプロセッサ12からデータを受信できる。例えば、プロセッサ12が64本のデータ線を有し、各ステートマシンエンジン14が8本のデータ線を有する場合には、プロセッサ12の全てのデータ線からデータを受信するように、8つのステートマシンエンジン14を64本のデータ線に結合することができる。したがって、プロセッサ12は、迅速にかつ標準インターフェイスを使用して、ステートマシンエンジン14にデータを提供することができる。さらに、ステートマシンエンジン14は、ステートマシンエンジン14によって集合的に受信される全てのデータが全てのステートマシンエンジン14に正しい手順で提供されるように、ともに調整するように構成できる。

【0076】

前述のように、解析されるべきデータは、バスインターフェイス130で受信される。バスインターフェイス130は、データバッファ132及び命令バッファ133を含む、各ステートマシンエンジン14（例えば、F0、F1、F2、F3、F4、F5、F6、F7）のデータバッファシステム159にデータを導く。データバッファ132は、解析されるべきデータを受信して一時的に格納するように構成される。図示する実施形態においては、各ステートマシンエンジン14に2つのデータバッファ132（例えば、データバッファA及びデータバッファB）が存在する。データは、2つのデータバッファ132のうちのいずれかに格納することができ、一方で、FSMラチス30による解析のために、他方のデータバッファ132からデータは除かれている。前述のように、命令バッファ133は、バスインターフェイス130を通じてプロセッサ12からの命令、例えば、解析されるべきデータに対応する命令を受信するように構成される。解析されるべきデータ及びデータに対応する命令が、データバッファシステム159からIRバス及びプロセスバッファインターフェイス136を通じて、FSMラチス30の1つ以上に提供される。本実施形態においては、物理的なFSMラチス30は、論理グループにおいて配列される。つまり、ステートマシンエンジン14F0及びF1のFSMラチス30は、論理グループA162に配列され、ステートマシンエンジン14F2及びF3のFSMラチス30は、論理グループB164に配列され、ステートマシンエンジン14F4及びF5のFSMラチス30は、論理グループC166に配列され、そしてステートマシンエンジン14F6及びF7のFSMラチス30は、論理グループD168に配列される。他の実施形態においては、物理的なFSMラチス30は、任意の適切な数（例えば、1、2、3、4、5、6、7、8）の論理グループに配列されてもよい。さらにまた、認識されるように、IRバス及びプロセスバッファインターフェイス136を通じてステートマシンエンジン14間でデータを取り取りすることができる。例えば、IRバス及びプロセスバッファインターフェイス136を使用して、任意のステートマシンエンジン14（例えば、F0、F1、F2、F3、F4、F5、F6、F7）間でデータを取り取りすることができる。8つのステートマシンエンジン14を図示したが、ランクにおけるデバイスは、任意の適切な数のステートマシンエンジン14（例えば、1、2、4、8など）を有することができる。認識されるように、各ステートマシンエンジン14のIRバス及びプロセスバッファインターフェイス136は、（例えば、それ自身のデータバッファシステム159、及び他のステートマシンエンジン14のIRバス及びプロセスバッファインターフェイス136からの）データを受信するための入力を含むことができる。同じように、各ステートマシンエンジン14のIRバス及びプロセスバッファインターフェイス136は、（例えば、FSMラチス30、及び他のステートマシンエンジン14のIRバス及びプロセスバッファインターフェイス136に）データを送信するための出力を含むことができる。

【0077】

10

20

30

40

50

バスインターフェイス 130 は、データを効率的に使用するように調整されたフォーマットにおいて解析されるべきデータを受信することができる。具体的には、図 11 ~ 図 14 は、バスインターフェイス 130 を通じてステートマシンエンジン 14 に提供されたデータブロックに、プロセッサ 12 によってデータを割り当てる（例えば、グループ化する）ことができる方法の実施例を図示する。

#### 【0078】

ここで図 11 を参照し、ステートマシンエンジン 14 に提供されるデータブロックに、プロセッサ 12 によって割り当てられたデータセグメント（例えば、データセット、検索語）の一実施例を図示する。本実施形態においては、複数のデータセグメントが単一のデータブロックに割り当てる。各データブロックは、FSM ラチス 30 の単一の論理グループ（例えば、162、164、166、168）（例えば、ランクにおけるステートマシンエンジン 14 のうちの 1 つ以上のステートマシンエンジン 14）によって解析されるように割り当てる。例えば、データストリーム 170（例えば、プロセッサ 12 によってステートマシンエンジン 14 に送信される大量のデータ）は、プロセッサ 12 によって、論理グループ A 162 用に意図されたデータに対応する第一のデータブロック 172、論理グループ B 164 用に意図されたデータに対応する第二のデータブロック 174、論理グループ C 166 用に意図されたデータに対応する第三のデータブロック 176、及び論理グループ D 168 用に意図されたデータに対応する第四のデータブロック 178 に割り当てる。つまり、データストリーム 170 は、プロセッサ 12 によって、データセグメント 180、182、184、186、188、190、192、194、196、198 及び 200 から組み立てられる。認識されるように、データセグメント 180、182、184、186、188、190、192、194、196、198 及び 200 の各々は、FSM ラチス 30 によって解析されるデータセットを表し得る。認識されるように、プロセッサ 12 は、データセグメント 180、182、184、186、188、190、192、194、196、198 及び 200 を、任意の適切な根拠に従いデータブロック 172、174、176 及び 178 に割り当てることができる。例えば、プロセッサ 12 は、データセグメントを、各データセットの長さ及び／またはデータセットを効率的に処理するための、解析されるデータセットの順序に基づいて、特定のデータブロックに割り当てることができる。

#### 【0079】

データセグメント 180、182、184、186、188、190、192、194、196、198 及び 200 は、任意の適切な様式でデータブロック 172、174、176 及び 178 に割り当てることができる。例えば、データセグメント 180、182、184、186、188、190、192、194、196、198 及び 200 は、データブロック 172、174、176 及び 178 におけるバイト数を最小限化するように、データブロック 172、174、176 及び 178 に割り当てることができる。別の実施例として、データセグメント 180、182、184、186、188、190、192、194、196、198 及び 200 は、特定のデータセグメントがともにグループ化されるように、データブロック 172、174、176 及び 178 に割り当てることができる。

#### 【0080】

図示するように、第一のデータブロック 172 は、データセグメント A 180、データセグメント F 190、及びデータセグメント I 196 を含む。第二のデータブロック 174 は、データセグメント B 182 及びデータセグメント K 200 を含む。さらにまた、第三のデータブロック 176 は、データセグメント C 184、データセグメント E 188、及びデータセグメント G 192 を含む。第四のデータブロック 178 は、データセグメント D 186、データセグメント H 194、及びデータセグメント J 198 を含む。

#### 【0081】

認識されるように、データブロックを効率的に処理するように全てのデータブロックが等しい量のデータを有してもよい。さらにまた、データブロック内のデータセグメントは

10

20

30

40

50

、データセグメントが開始及び終了する時を処理デバイスが判定できるように、データブロック内において所定の間隔（例えば、バイト、単語）で開始及び／または終了してもよい。しかしながら、データセグメントは、所定の間隔で開始及び／または終了するための正確な量のデータを有さないことがある。したがって、データが所定の間隔でデータブロック内において開始及び／または終了するように、データパディングを特定のデータセグメント間に挿入してもよい。さらに、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、データブロックの終端にデータパディングを追加してもよい。

#### 【0082】

ここで図12を参照し、図11のデータブロック172、174、176及び178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一実施例を図示する。例えば、第一のデータブロック172では、データパディング202を、データセグメントA180とデータセグメントF190との間に挿入できる。さらに、データパディング204を、データセグメントF190とデータセグメントI196との間に挿入できる。別の実施例として、第二のデータブロック174では、データパディング206を、データセグメントB182とデータセグメントK200との間に挿入できる。第三のデータブロック176では、データパディング208を、データセグメントC184とデータセグメントE188との間に挿入できる。同じように、データパディング210を、データセグメントE188とデータセグメントG192との間に挿入できる。別の実施例として、第四のデータブロック178では、データパディング212を、データセグメントD186とデータセグメントH194との間に挿入できる。さらに、データパディング214を、データセグメントH194とデータセグメントJ198との間に挿入できる。

#### 【0083】

データパディング202、204、206、208、210、212及び214は、解析されない任意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、フィラーデータ、ガベージデータなど）を含んでもよい。一実施形態においては、データパディングとして使用されるバイト数は、先のデータセグメントのバイト数に加えられた場合に完全なワード境界に達するバイト数であってもよい（すなわち、先のデータセグメントのバイト数とデータパディングとして使用されるバイト数との和は、完全なワード境界によって割り切れる。例えば、データパディング202のバイト数は、データパディング202とデータセグメントA180（すなわち、先のデータセグメント）とのバイトの合計数が、完全なワード境界によって割り切れる（例えば、余りがない）数にできる。図示する実施形態においては、完全なワード境界は、8バイトであってもよい。他の実施形態においては、完全なワード境界は、任意の適切な数のバイトまたはビットであってもよい。それゆえ、図示する実施形態においては、仮にデータセグメントA180が63バイトのデータを含む場合には、データパディング202は、1バイトのデータを含むであろう（例えば、データセグメントA180とデータパディング202とのデータの合計バイトを64にすると、この64は、8バイトで割り切れる）。別の実施例として、データセグメントA180が（例えば、8によって割り切れない）60バイトのデータを含む場合には、データパディング202は、4バイトのデータを含むであろう。さらなる実施例として、データセグメントA180が64バイトのデータを含む場合には、データパディング202は、ゼロバイトのデータを含むであろう。換言すると、データセグメントA180とデータセグメントF190との間にデータパディング202を必要としないであろう。認識されるように、各データパディング202、204、206、208、210、212及び214も、同様の様式で作用することができる。

#### 【0084】

ここで図13を参照し、図12のデータブロック172、174、176及び178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一実施例を図示する。具体的には、各データブロック172、174、176及び178におけるバイト数を等しくするために、必要に応じて、各データブロック172、174、176及び178の終端にデータパディングを挿入してもよい。さらにまた、各データブロック172、174、176

10

20

30

40

50

及び 178 の終端に位置するデータパディングを使用できることによって、各データブロック 172、174、176 及び 178 は前述のような完全なワード境界に達する。図示する実施形態においては、データセグメント I 196 の後ろにデータパディング 216 が挿入され、データセグメント G 192 の後ろにデータパディング 218 が挿入され、そしてデータセグメント J 198 の後ろにデータパディング 220 が挿入される。したがって、データブロック 172、174、176 及び 178 の各々が、等しいバイト数を含み、かつ完全なワード境界に達する。

#### 【0085】

FSM ラチス 30 にとって、データパディングを有効データと区別することが困難である場合がある。したがって、有効データの解析中に FSM ラチス 30 がデータパディングを識別して無視できるように、命令をデータブロック 172、174、176 及び 178 に付随させることができ。そのような命令は、プロセッサ 12 によってバスインターフェイス 130 を通じてステートマシンエンジン 14 に送信され、そしてステートマシンエンジン 14 の命令バッファ 160 によって受信、格納及び提供することができる。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 及び 236 に論理的に分けることができる。領域 222、224、226、228、230、232、234 及び 236 の端部境界は、各領域が任意のデータパディング端部を終端とするように形成できる。例えば、第一の領域 222 は、データパディング 208 端部を終端とする。別の実施例として、第五の領域 230 は、データパディング 204 端部を終端とする。

10

20

#### 【0086】

データブロック 172、174、176 及び 178 に付随する命令は、各領域 222、224、226、228、230、232、234 及び 236 のバイトの合計数、並びに各領域内の各データブロック 172、174、176 及び 178 における有効バイト数（例えば、パディングのバイトを除くバイト数）を含むことができる。例えば、命令は、第一の領域 222 に対応するバイト数 238、第一の領域 222 内の第一のデータブロック 172 における有効バイトに対応するバイト数 240、第一の領域 222 内の第二のデータブロック 174 における有効バイトに対応するバイト数 242、第一の領域 222 内の第三のデータブロック 176 における有効バイトに対応するバイト数 244、及び第一の領域 222 内の第四のデータブロック 178 における有効バイトに対応するバイト数 246 を含むことができる。この実施例においては、データセグメント A1、B1 及び D1 に続くパディングが存在しないため、238、240、242 及び 246 に示すバイト数が等しいことに留意されたい。

30

#### 【0087】

同じように、命令は、第二の領域 224 に対応するバイト数 248、250、252、254 及び 256、第三の領域 226 に対応するバイト数 258、260、262、264 及び 266、第四の領域 228 に対応するバイト数 268、270、272、274 及び 276、第五の領域 230 に対応するバイト数 278、280、282、284 及び 286、第六の領域 232 に対応するバイト数 288、290、292、294 及び 296、第七の領域 234 に対応するバイト数 298、300、302、304 及び 306、並びに、第八の領域 236 に対応するバイト数 308、310、312、314 及び 316 を含むことができる。命令が各領域 222、224、226、228、230、232、234、236 内の各データセグメントにおける有効バイト数を含むことができることが留意されるべきである。したがって、第七の領域 234 では、論理グループ A162 は、有効バイト数 300 がゼロであってもよい。さらにまた、第八の領域 236 では、論理グループ A162 は、有効バイト数 310 がゼロであってもよい。したがって、FSM ラチス 30 は、命令を使用して、データセグメントに挿入されるデータパディングを識別することができる。1 つの特定の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 及び 178 のグループに含まれる命令が、FSM ラチス 30 がデータパディング（すなわち、無効データ）から有効データを区別できる任意の適切なグループ

40

50

の命令であってもよいことが留意されるべきである。

【0088】

ここで図14を参照し、ステートマシンエンジン14のデータバッファシステム159に伝送するために、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及び178の一実施例を図示する。データブロック172、174、176及び178の各々は、全体のワード長と等しいバイト数318を有するデータの行で配列される。図示する実施形態においては、全体のワード長は、各ステートマシンエンジン14（例えば、F0、F1、F2、F3、F4、F5、F6及びF7）が1バイトを示すため、8バイトである。データセグメントの各々からの第一のバイトは各データブロック172、174、176及び178の右側から始まり、データセグメントA180における第一のバイトが列F0に位置し、データセグメントA180における第八のバイトが列F7に位置するように、各データブロックの左側に向かうほど増大する。認識されるように、列F0は、F0ステートマシンエンジン14のデータバッファ132に最初に格納されるデータを表し、列F1は、F1ステートマシンエンジン14のデータバッファ132に最初に格納されるデータを表し、それ以降も同様である。さらにまた、データセグメントは、行内において上部から底部まで配置される。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を終端とする（すなわち、それらの各々が全体のワード長に広がる）。さらにまた、各データブロック172、174、176及び178のサイズは等しい。認識されるように、動作中、データブロック172、174、176及び178は、プロセッサ12からステートマシンエンジン14に連続的に提供することができる。

【0089】

任意のブロック（172、174、176及び178）のデータは、対応する論理グループ（162、164、166または168のそれぞれ）用に意図されたデータが、ランクにおけるステートマシンエンジン14のデータバッファシステム159の全体にわたって散布されるように、ランクにおけるデバイス内の全てのステートマシンエンジン14のデータバッファシステム159にわたって提供され、格納される。データは、バスインターフェイス130を通じてデータバッファシステム159に迅速に提供できるような方法で受信及び格納することができる。ある実施形態においては、データバッファシステム159のデータバッファ132は、バスインターフェイス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できる。他の実施形態においては、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イス130との間の接続に基づいて、データの限られた部分だけを受信することができる。以下に詳細に説明するように、データバッファシステム159に格納されたデータは、データがデータバッファシステム159からIRバス及びプロセスバッファインターフェイス136を通じてプロセスバッファ134に提供されるときに選別される。

【0090】

図15は、デバイスがランクにおいて配列されており、IRバス及びプロセスバッファインターフェイス136によってともに結合された複数の物理的なステートマシンエンジン14の一実施例を図示する。具体的には、図示する実施形態においては、8つのステートマシンエンジン14（F0、F1、F2、F3、F4、F5、F6及びF7）が、单一プリント基板に配列されて、モジュール340を形成する。ある実施形態においては、モジュール340は、デュアルインラインメモリモジュール（DIMM）と同様に配列されてもよい。例えば、单一チップは、ステートマシンエンジン14を1つだけ含んでもよい。さらにまた、1つのステートマシンエンジン14を各々が有する複数のチップを、モジュール340のプリント基板に結合（例えば、電気的に、光学的に、別なように動作可能に結合）してもよい。例えば、プリント基板は、1つのステートマシンエンジン14を各々が含む1個、2個、4個、8個、16個またはそれ以上のチップを含んでもよい。さらに、ある実施形態においては、モジュール340は、モジュール340とコンピュータ、ワークステーションまたはサーバなどのシステムとを結合するための、72個、100個

、144個、168個、184個、200個、240個またはより少ないもしくは多いピンまたは接続素子を含むことができる。例えば、モジュール340は、240個のピンを含み、サーバにおけるスロット内に挿入されるように構成することができる。

#### 【0091】

モジュール340のピンは、DDR3バスインターフェイス130に結合し、プロセッサ12とモジュール340との間のデータ転送を容易にする。さらに、モジュール340のピンは、プリント基板のルーティングラインを使用して、ステートマシンエンジン14(例えば、F0、F1、F2、F3、F4、F5、F6及びF7)を有するチップのピンに結合される。DDR3バスインターフェイス130は、プロセッサ12に、各ステートマシンエンジン14のデータバッファシステム159のメモリアドレスを選択させることができるDDR3アドレスバス342を含む。図示するように、DDR3アドレスバス342は、16個のアドレス線を含む。しかしながら、他の実施形態においては、DDR3アドレスバス342は、16個よりも少ないまたは多いアドレス線を含んでもよい。DDR3アドレスバス342の各アドレス線は、各ステートマシンエンジン14のデータバッファシステム159に結合される。したがって、プロセッサ12は、各ステートマシンエンジン14のデータバッファシステム159のアドレスをともに選択することができる。

#### 【0092】

認識できるように、ステートマシンエンジン14とプロセッサ12との間のDDR3バスインターフェイス130によって、ステートマシンエンジン14が、プロセッサ12からの全てのデータ線を使用して、プロセッサ12からデータを受信することができてもよい。例えば、プロセッサ12が64本のデータ線を有し、各ステートマシンエンジン14が8本のデータ線を有する場合には、プロセッサ12の全てのデータ線からデータを受信するように、8つのステートマシンエンジン14が64本のデータ線に結合することができる。したがって、プロセッサ12は、迅速にかつ標準インターフェイスを使用して、ステートマシンエンジン14にデータを提供することができる。さらに、ステートマシンエンジン14は、ステートマシンエンジン14によって集合的に受信される全てのデータが全てのステートマシンエンジン14に正しい手順で提供されるように、ともに調整するように構成できる。図示する実施形態においては、DDR3バスインターフェイス130は、各ステートマシンエンジン14に結合されたデータ線344、346、348、350、352、354、356及び358の組を有するデータバス343を含む。図示するように、個々のデータ線が各ステートマシンエンジン14に結合される。例えば、ある実施形態においては、データバス343は、64本のデータ線を含み、モジュール340は、8つのステートマシンエンジン14を含む。そのような実施形態においては、8本のデータ線が、8つのステートマシンエンジン14の各々に結合され、かつ/またはそれぞれに専用であってもよい。したがって、データバス343及びDDR3アドレスバス342を使用して、それぞれのデータバイトを、各ステートマシンエンジン14のデータバッファシステム159の選択されたアドレスに同期して提供することができる。ある実施形態においては、8本よりも少ないまたは多いデータ線が、ステートマシンエンジン14とデータバス343とを結合してもよい。さらにまた、DDR3バスインターフェイス130は、64本よりも少ないまたは多いデータ線を含んでもよい。以下の表、表1は、規定の数のデータ線を有するDDR3バスインターフェイス130の構成、及びステートマシンエンジン14に結合されたDDR3バスインターフェイス130におけるデータ線の数の種々の実施例を図示する。

【表1】

ステートマシンエンジン14 入力データバス幅	DDR3バスインターフェイス130 データバス343幅			
	x 8	x 16	x 32	x 64
x 8	1つのステート マシンエンジン14	2つのステート マシンエンジン14	4つのステート マシンエンジン14	8つのステート マシンエンジン14
x 16	該当なし	1つのステート マシンエンジン14	2つのステート マシンエンジン14	4つのステート マシンエンジン14

## 【0093】

表1に図示するように、データバス343が8本のデータ線を含む場合には、その8本のデータ線が、8本の線の入力データバス幅を有する1つのステートマシンエンジン14に結合することができる。別の実施例として、データバス343が16本のデータ線を含む場合には、その16本のデータ線が、8本の線の入力データバス幅を有する2つのステートマシンエンジン14、または16本の線の入力データバス幅を有する1つのステートマシンエンジン14に結合することができる。さらなる実施例として、データバス343が32本のデータ線を含む場合には、その32本のデータ線が、8本の線の入力データバス幅を有する4つのステートマシンエンジン14、または16本の線の入力データバス幅を有する2つのステートマシンエンジン14に結合することができる。さらに、データバス343が64本のデータ線を含む場合には、その64本のデータ線が、8本の線の入力データバス幅を有する8つのステートマシンエンジン14、または16本の線の入力データバス幅を有する4つのステートマシンエンジン14に結合することができる。

## 【0094】

DDR3バスインターフェイス130は、プロセッサ12とモジュール340との間のデータ転送を容易にするための他の制御線360、362、364、366、368、370、372及び374を含む。他の制御線360、362、364、366、368、370、372及び374によって、プロセッサ12と選択されたステートマシンエンジン14との間の個別の通信、及び/またはプロセッサ12と集合的なステートマシンエンジン14との間の通信が可能になる。

## 【0095】

動作中、プロセッサ12は、DDR3バスインターフェイス130にデータを提供できる。例えば、プロセッサ12は、64本のデータ線を有するデータバスを使用して、一度に64ビットのデータを提供することができる。さらにまた、ステートマシンエンジン14は、その各々が、各ステートマシンエンジン14に結合された8本のデータ線から8ビットのデータを受信することができる。したがって、プロセッサ12は、一度に64ビットのデータをモジュール340に提供することができる。前述のように、ステートマシンエンジン14は、他の異なるステートマシンエンジン14によって解析されるべきデータを受信するように構成できる。それゆえ、プロセッサ12は、モジュール340のステートマシンエンジン14の1つ以上によって処理されるように各データブロックが意図されたデータブロックにおいて、モジュール340にデータを提供することができる。換言すると、プロセッサ12は、それが提供するデータブロックをソート及び/またはパックしなくてもよい。例えば、プロセッサ12は、連続的なバイト数のデータの一部がステートマシンエンジン14の各々(F0、F1、F2、F3、F4、F5、F6及びF7)によって受信及び格納される場合でさえも、ステートマシンエンジン14 F0によって解析されるように意図された連続的なバイト数のデータを提供することができる。したがって、プロセッサ12は、単純かつ効率的な様式でデータをモジュール340に提供し、ステートマシンエンジン14に、選択されたステートマシンエンジン14によって処理されるデ

10

20

30

40

50

ータをソートさせることができる。

【0096】

プロセッサ12は、ステートマシンエンジン14に、プロセッサ12から提供される各データブロックを解析するようステートマシンエンジン14に命令するための命令、ステートマシンエンジン14に、一定期間（例えば、所定の期間）中データを解析させるための命令、及びステートマシンエンジン14に、プロセッサ12から提供される各データブロックの長さに関する命令も提供することができる。ある実施形態においては、データバッファシステム159は、プロセッサ12から受信した命令を格納するための特定のメモリロケーションを含むことができる。したがって、プロセッサ12は、DDR3アドレスバス342を使用して、命令の受信に専用のデータバッファシステム159の所定のアドレスを選択することができる。次に、プロセッサ12は、データバス343及びDDR3アドレスバス342を使用して、ステートマシンエンジン14の各々に命令を提供することができる。

【0097】

IRバス及びプロセスバッファインターフェイス136は、モジュール340の一部であり、ステートマシンエンジン14を相互接続する接続（例えば、電気的、光学的または別の動作可能な接続）を含む。図示するように、IRバス及びプロセスバッファインターフェイス136は、インストラクションバスの一部であってもよいIRデータバス376及び他の制御線378を含む。図示する実施形態においては、IRデータバス376は、ステートマシンエンジン14の各々を互いに結合する8本のデータ線を含む。具体的には、IRデータバス376は、各ステートマシンエンジン14のデータバッファシステム159と、各ステートマシンエンジン14のプロセスバッファ134とを結合する。さらに、IRデータバス376を使用して、プロセッサ12から受信したデータをデータの解析のための他のステートマシンエンジン14に提供することができる。他の制御線378を使用して、ステートマシンエンジン14間のデータ転送を同期及び/または制御することができる。

【0098】

ステートマシンエンジン14を有する各チップは、プロセッサ12から受信したデータを処理するために論理グループにグループ化してもよい。認識できるように、モジュール340のステートマシンエンジン14は、1つ以上の論理グループを含むことができる。モジュール340では、全てのステートマシンエンジン14（F0、F1、F2、F3、F4、F5、F6及びF7）を含む1つの論理グループが存在してもよい。さらにまた、2つ、3つ、4つ、5つ、6つ、7つまたは8つの論理グループが存在してもよい。論理グループは、任意の数のステートマシンエンジン14を有することができ、同一サイズでなくてもよい。例えば、一実施形態においては、第一の論理グループは、ステートマシンエンジン14F0及びF1を含んでもよく、第二の論理グループは、ステートマシンエンジン14F2、F3、F4、F5、F6及びF7を含んでもよい。2つ以上のステートマシンエンジン14を含むことができる論理グループを使用することによって、単一のステートマシンエンジン14にプログラムするには大き過ぎる単一のオートマトンプロセッサブロックを、論理グループによって解析することができる。さらにまた、論理グループは、複数のデータブロックを、別個の論理グループによって同時に解析できるようにする。さらに、論理グループは、例えば、同一サイズを有する論理グループを使用して、かつ/またはパラレルプロセッシングによって、ハイスループット速度においてデータを解析できるようにする。したがって、モジュール340のアーキテクチャは、フレキシビリティをもたらし、モジュール340の単一のステートマシンエンジン14に、最大1.0Gb/s以上のレートでデータを処理させることができる。さらにまた、モジュール340のアーキテクチャは、モジュール340に、例えば8つのステートマシンエンジン14を使用したパラレルの8つの論理グループにおいてデータを処理させることができる。それによって、単一のステートマシンエンジン14の最大8倍のレート（例えば、8.0Gb/s以上）のデータ処理レートを達成することができる。

10

20

30

40

50

## 【0099】

各ステートマシンエンジン14が論理グループの一部である場合でさえも、ステートマシンエンジン14は、それらが特定の論理グループの一部であることの何らかの指示、及び／またはその論理グループの一部であるステートマシンエンジン14の数についての何らかの指示を受信しなくてもよい。しかしながら、特定の情報を、命令として及び／または他の制御線360、362、364、366、368、370、372及び374を使用して、ステートマシンエンジン14に提供してもよい。例えば、ステートマシンエンジン14は、DDR3バスインターフェイス130の合計のバス幅、DDR3バスインターフェイス130に結合されたデータ線の数、IRデータバス376のデータ線の数、DDR3バスインターフェイス130におけるデバイス位置、IRデータバス376におけるデバイス位置、ステートマシンエンジン14がマスタデバイスであるか否か、ステートマシンエンジン14がスレーブデバイスであるか否か、実行されるIRデータバス376サイクルの数、受信されるバイト数、及び／または解析される（例えば、有効データの）バイト数についての指示を受信及び／または格納することができる。10

## 【0100】

例えば、モジュール340の初期化中、プロセッサ12は、各ステートマシンエンジン14（例えば、0、1、2、3、4、5、6、7）に数を割り当てるように、各ステートマシンエンジン14にデータを提供することができる。ある実施形態においては、「0」を受信するステートマシンエンジン14が、ステートマシンエンジン14F0かつ「マスター」デバイスであってもよく、全ての他のデバイスが「スレーブ」デバイスであってもよい。他の実施形態においては、「マスター」デバイスに、任意の適切な値を割り当てることができる。ある実施形態においては、「マスター」デバイスは、ステートマシンエンジン14の同期を調整するように構成できる。プロセッサ12は、モジュール340の一部であるステートマシンエンジン14の総数、ステートマシンエンジン14が属する論理グループ、及び／またはモジュール340の一部である論理グループの数を表すデータも各ステートマシンエンジン14に提供することができる。20

## 【0101】

論理グループが1つである一実施形態においては、IRバス及びプロセスバッファインターフェイス136によって、ランクにおける各ステートマシンエンジン14（例えば、F0、F1、F2、F3、F4、F5、F6及びF7）が、プロセッサ12によって提供された全てのデータバイトを解析できる。したがって、そのような実施形態においては、全データストリームを、ステートマシンエンジン14の各々において処理することができる。複数の論理グループを有する別の実施形態においては、IRバス及びプロセスバッファインターフェイス136によって、ランクにおける各ステートマシンエンジン14が、データスライシングのスキームを実施することによって、データストリームのうちの割り当てられた一部をタイムリーに受信することができる。例えば、プロセッサ12は各論理グループ用に意図されたデータを連続的に提供することができるが、ステートマシンエンジン14は、意図されたステートマシンエンジン14にデータを効率的に提供するようIRデータバス376にデータを提供できるように、オフセット様式において、データを各ステートマシンエンジン14のデータバッファシステム159に格納することができる。30

## 【0102】

図16は、複数の物理的なステートマシンエンジン14に結合されたIRバス及びプロセスバッファインターフェイス136の一実施例を図示する。具体的には、図示する実施形態は、2つの論理グループ（すなわち、論理グループ0及び論理グループ1）に分けられた4つのステートマシンエンジン14を含む。さらにまた、4つのステートマシンエンジン14は、32本のデータ線を有するDDR3バスインターフェイス130に結合することができる。前述のように、IRデータバス376がステートマシンエンジン14の各々に結合される。さらにまた、他の制御線378は、ステートマシンエンジン14の各々に結合されたランク同期化線（RSYNC）380、IRバスデータ有効線（IRDV）382、第一のIRバスデータストローブ線（IRDQS）384、及び第二のIRバス40

データストローブ線 (I RDQS #) 386 を含む。ステートマシンエンジン 14 の各々は、RSYNC 380 からの入力を受信し、かつ / または RSYNC 380 の信号を制御するように構成される。例えば、一実施形態においては、「スレーブ」デバイス及び / または「マスタ」デバイスは、RSYNC 380 を論理 0 に引き込むことによって、RSYNC 380 の信号を論理 1 から論理 0 に制御するように構成できる。一方で、「マスタ」デバイスは、RSYNC 380 から受信した入力に基づいて、IRDV 382 の信号を制御するように構成できる。図示する実施形態においては、ステートマシンエンジン 14 F 0 が「マスタ」デバイスとして構成され、一方で、ステートマシンエンジン 14 F 1、F 2 及び F 3 が「スレーブ」デバイスとして構成されている。さらにまた、ステートマシンエンジン 14 F 1、F 2 及び F 3 は、RSYNC 380 の信号を制御するように構成され 10 、一方で、ステートマシンエンジン 14 F 0 は、RSYNC 380 から受信した入力に基づいて、IRDV 382 の信号を制御するように構成される。ステートマシンエンジン 14 F 0 は、ステートマシンエンジン 14 F 1、F 2 及び F 3 の全てがデータの受信に利用可能であることを示す第一の指示 (例えば、論理ハイ) を受信するように構成できる。さらに、ステートマシンエンジン 14 F 0 は、ステートマシンエンジン 14 F 1、F 2 及び F 3 の少なくとも 1 つがデータの受信に利用できないことを示す第二の指示 (例えば、論理ロウ) を受信するように構成できる。例えば、ステートマシンエンジン 14 F 1、F 2 及び F 3 の任意の 1 つは、ステートマシンエンジン 14 F 1、F 2 及び F 3 の少なくとも 1 つがデータの受信に利用できないことを、ステートマシンエンジン 14 F 0 に通知するために、RSYNC 380 の信号を論理ロウに制御 (例えば、論理ロウを出力) することができる。 20

#### 【0103】

さらに、ステートマシンエンジン 14 の各々は、IRDV 382 からの入力を受信し、かつ / または IRDV 382 の信号を制御するように構成される。例えば、一実施形態においては、「マスタ」デバイスは、IRDV 382 の信号を制御するように構成でき、一方で、「スレーブ」デバイスは、IRDV 382 からの入力を受信するように構成できる。図示する実施形態においては、ステートマシンエンジン 14 F 0 は、IRDV 382 の信号を制御するように構成され、一方で、ステートマシンエンジン 14 F 1、F 2 及び F 3 は、IRDV 382 からの入力を受信するように構成される。ステートマシンエンジン 14 F 0 は、ステートマシンエンジン 14 F 1、F 2 及び F 3 に提供されるデータが有効であることを示す第一の指示 (例えば、論理ハイ) に信号を制御するように構成できる。さらに、ステートマシンエンジン 14 F 0 は、ステートマシンエンジン 14 F 1、F 2 及び F 3 に提供されるデータが無効であることを示す第二の指示 (例えば、論理ロウ) に信号を制御するように構成できる。ステートマシンエンジン 14 の各々は、IRDQS 384 及び IRDQS # 386 からの入力を受信し、かつそれらに出力を提供するように構成される。さらに、ステートマシンエンジン 14 の各々は、IRDQS 384 及び / または IRDQS # 386 にデータを出力し、その一方で、IR データバス 376 にもデータを提供するように構成される。 30

#### 【0104】

図 17 は、IR バス及びプロセッサバッファインターフェイス 136 の信号のタイミング図の一実施例を図示する。特定の構成では、RSYNC 380 を使用して、ステートマシンエンジン 14 を同期することができる。例えば、ステートマシンエンジン 14 の各々は、ステートマシンエンジン 14 が RSYNC 380 の信号を論理ロウに駆動する (例えば、RSYNC 380 を論理ロウに「引き込む」) デフォルト状態を含むことができる。プロセッサ 12 は、ステートマシンエンジン 14 の同期を起動するための種々の命令をステートマシンエンジン 14 に提供することができる。例えば、同期を起動するための命令は、M\_BAR、I\_DA.PROCESS、I\_DA.PROCESS\_EoP、I\_DA.PROCESS\_EoD、I\_DB.PROCESS、I\_DB.PROCESS\_EoP、及び I\_DB.PROCESS\_EoD を含んでもよい。ステートマシンエンジン 14 は、これらの命令のうちの 1 つに遭遇すると、RSYNC 380 の信号の論理ロウ 40

への駆動を停止し（例えば、R S Y N C 3 8 0を「手放し」）、動作を中断する（例えば、命令の実行を継続しない）。ステートマシンエンジン1 4は、R S Y N C 3 8 0が論理ハイに遷移する（これは、全てのステートマシンエンジン1 4がR S Y N C 3 8 0の信号の論理ロウへの駆動を停止した後、またはR S Y N C 3 8 0の「手放し」の後に起こる）、かつ／またはI R D V 3 8 2が論理ハイに遷移するまで、動作を中断する。ステートマシンエンジン1 4は、R S Y N C 3 8 0及び／またはI R D V 3 8 2が論理ハイであることを検出すると、同期を起動するための命令の動作を再開する。同期を起動するための命令の終了時に、ステートマシンエンジンは、ステートマシンエンジン1 4がR S Y N C 3 8 0の信号を論理ロウに駆動する（例えば、R S Y N C 3 8 0を論理ロウに「引き込む」）そのデフォルト状態に戻ることができる。

10

#### 【0105】

そのような動作中に交換される信号の一実施形態を図17に図示する。例えば、時点3 8 8に、R S Y N C 3 8 0は、論理ロウから論理ハイに遷移する。そのような遷移は、全てのステートマシンエンジン1 4 F 0、F 1、F 2及びF 3がステートマシンエンジン1 4の同期を起動するための命令を（例えば、プロセッサ1 2から）受信し、全てのステートマシンエンジン1 4 F 0、F 1、F 2及びF 3が動作を中断し、そして全てのステートマシンエンジン1 4 F 0、F 1、F 2及びF 3がR S Y N C 3 8 0の信号の論理ロウへの駆動を停止したことを示す。この実施例においては、ステートマシンエンジン1 4 F 0、F 1、F 2及びF 3は、データバッファ処理命令（例えば、I \_ D A . P R O C E S S、I \_ D A . P R O C E S S \_ E o P、I \_ D A . P R O C E S S \_ E o D、I \_ D B . P R O C E S S、I \_ D B . P R O C E S S \_ E o P、及びI \_ D B . P R O C E S S \_ E o D）に遭遇した。時点3 9 0に、R S Y N C 3 8 0は、論理ハイから論理ロウに遷移する。そのような遷移は、ステートマシンエンジン1 4 F 0、F 1、F 2及びF 3の少なくとも1つがデータバッファ処理命令を完了し、ステートマシンエンジン1 4 F 0、F 1、F 2及びF 3の少なくとも1つがR S Y N C 3 8 0を論理ロウに引き込んだことを示す。

20

#### 【0106】

時点3 9 2に、I R D V 3 8 2は、論理ロウから論理ハイに遷移する。そのような遷移は、「マスタ」ステートマシンエンジン1 4 F 0から全てのステートマシンエンジン1 4 F 0、F 1、F 2及びF 3への有効データが、I R データバス3 7 6に提供され始め、その提供が「マスタ」デバイスから始まり、ランクにおける各ステートマシンエンジン1 4を通して順番に（例えば、F 0、F 1、F 2、F 3）継続していることを示すものである。時点3 9 4に、I R D V 3 8 2は、論理ハイから論理ロウに遷移する。そのような遷移は、「マスタ」ステートマシンエンジン1 4 F 0から全てのステートマシンエンジン1 4 F 0、F 1、F 2及びF 3への有効データがもはやI R データバス3 7 6に提供されていないことを示すものである。

30

#### 【0107】

時点3 9 6に、「マスタ」ステートマシンエンジン1 4 F 0が、データ（例えば、論理グループ0用に意図された第一のデータバイト）をI R データバス3 7 6に提供し、論理ハイをI R D Q S 3 8 4及び／またはI R D Q S # 3 8 6に出力する。その後、時点3 9 7に、「マスタ」ステートマシンエンジン1 4 F 0は、データ（例えば、論理グループ1用に意図された第二のデータバイト）をI R データバス3 7 6に提供し、論理ハイをI R D Q S 3 8 4及び／またはI R D Q S # 3 8 6に出力する。さらに、時点3 9 8に、ステートマシンエンジン1 4 F 1が、データ（例えば、論理グループ0用に意図された第一のデータバイト）をI R データバス3 7 6に提供し、論理ハイをI R D Q S 3 8 4及び／またはI R D Q S # 3 8 6に出力する。その後、時点3 9 9に、ステートマシンエンジン1 4 F 1は、データ（例えば、論理グループ1用に意図された第二のデータバイト）をI R データバス3 7 6に提供し、論理ハイをI R D Q S 3 8 4及び／またはI R D Q S # 3 8 6に出力する。

40

#### 【0108】

さらにまた、時点4 0 0に、ステートマシンエンジン1 4 F 2が、データ（例えば、論

50

理グループ0用に意図された第一のデータバイト)をIRデータバス376に提供し、論理ハイをIRDQS384及び/またはIRDQS#386に出力する。その後、時点401に、ステートマシンエンジン14F2は、データ(例えば、論理グループ1用に意図された第二のデータバイト)をIRデータバス376に提供し、論理ハイをIRDQS384及び/またはIRDQS#386に出力する。さらに、時点402に、ステートマシンエンジン14F3が、データ(例えば、論理グループ0用に意図された第一のデータバイト)をIRデータバス376に提供し、論理ハイをIRDQS384及び/またはIRDQS#386に出力する。その後、時点403に、ステートマシンエンジン14F3は、データ(例えば、論理グループ1用に意図された第二のデータバイト)をIRデータバス376に提供し、論理ハイをIRDQS384及び/またはIRDQS#386に出力する。

10

#### 【0109】

したがって、ステートマシンエンジン14の各々は、2バイトのデータの「バースト長」を有するデータを提供する(例えば、各ステートマシンエンジン14は、ランクにおける次のステートマシンエンジン14がデータを提供する前に2バイトのデータを出力する)。図示する実施形態においては、第一のデータバイトは、第一の論理グループ(論理グループ0)用に意図されたデータであり、第二のデータバイトは、第二の論理グループ(論理グループ1)用に意図されたデータである。認識できるように、「バースト長」は、他の実施形態においては異なってもよい。例えば、「バースト長」は、論理グループの数に基づいてもよいし、論理グループの数と等しくてもよいし、その他であってもよい。データがIRデータバス376に提供されたときに、ステートマシンエンジン14は、データを受信し、ステートマシンエンジン14用に意図されたデータを格納する。例えば、ステートマシンエンジン14F0及びF1は、論理グループ0の一部であるため、ステートマシンエンジン14のF0及びF1各々によって提供されたあらゆる第一のバイトを格納する。さらにまた、ステートマシンエンジン14F2及びF3は、論理グループ1の一部であるため、ステートマシンエンジン14のF2及びF3各々によって提供されたあらゆる第二のバイトを格納する。ある実施形態においては、ステートマシンエンジン14の各々は、IRDQS384及び/またはIRDQS#386の立ち下がりエッジに基づいてデータを格納するように構成される。

20

#### 【0110】

30

図示する実施形態においては、ステートマシンエンジン14の各々は、唯一つのデータバーストを提供した。しかしながら、認識できるように、時点390にRSYNC380が論理ハイから論理ロウに遷移する前に、データバーストを提供する各ステートマシンエンジン14の回転サイクルを任意の回数繰り返してもよい。ある実施形態においては、データバーストを提供する各ステートマシンエンジン14のサイクルは、プロセッサ12からステートマシンエンジン14に提供される命令に基づいて繰り返すことができる。例えば、プロセッサ12は、それぞれのステートマシンエンジン14がIRデータバス376から格納可能なバイト数(CNTC)を示す指示を、ステートマシンエンジン14の各々に提供できる。したがって、データバーストを提供する各ステートマシンエンジン14のサイクルを、CNTC数と等しい回数繰り返すことができる。ある実施形態においては、CNTC数は、論理グループに提供される最長データグループの全長と等しくてもよい。その結果、その意図されたデータを受信するための十分なサイクルを各論理グループに与えることができる。さらに、プロセッサ12は、それぞれのステートマシンエンジン14が解析可能なバイト数(CNTV)を示す指示を、ステートマシンエンジン14の各々に提供できる。ある実施形態においては、CNTC数がCNTV数よりも大きい場合には、ステートマシンエンジン14は、CNTV数を超えてCNTC数に至るまでの受信したバイトを、無効データ(例えば、ジャンクデータ、ガベージデータなど)とみなすことができる。

40

#### 【0111】

時点404に、RSYNC380は、論理ロウから論理ハイに遷移する。そのような遷

50

移は、全てのステートマシンエンジン 14 F 0、F 1、F 2 及び F 3 がステートマシンエンジン 14 の同期を起動するための命令を（例えば、プロセッサ 12 から）受信し、全てのステートマシンエンジン 14 F 0、F 1、F 2 及び F 3 が動作を中断し、そして全てのステートマシンエンジン 14 F 0、F 1、F 2 及び F 3 が RSYNC 380 の信号の論理口ウへの駆動を停止したことを示す。この実施例においては、ステートマシンエンジン 14 F 0、F 1、F 2 及び F 3 は、M\_BAR 命令に遭遇した。M\_BAR 命令は、さらなる命令がステートマシンエンジン 14 によって実行される前に、ステートマシンエンジン 14 を同期するのに使用される。時点 405 に、RSYNC 380 は、論理ハイから論理口ウに遷移する。そのような遷移は、ステートマシンエンジン 14 F 0、F 1、F 2 及び F 3 の少なくとも 1 つが RSYNC 380 を論理口ウに引き込み、ステートマシンエンジン 14 が、同期され、さらなる命令の実行に進むことができる。 10

#### 【0112】

時点 406 に、RSYNC 380 は、論理口ウから論理ハイに遷移する。この実施例においては、ステートマシンエンジン 14 F 0、F 1、F 2 及び F 3 は、別のデータバッファ処理命令（例えば、I\_DA\_PROCESS、I\_DA\_PROCESS\_EoP、I\_DA\_PROCESS\_EoD、I\_DB\_PROCESS、I\_DB\_PROCESS\_EoP、及び I\_DB\_PROCESS\_EoD）に遭遇した。したがって、時点 407 に、IRDV 382 は、論理口ウから論理ハイに遷移する。そのような遷移は、「マスタ」ステートマシンエンジン 14 F 0 から全てのステートマシンエンジン 14 F 0、F 1、F 2 及び F 3 への有效データが、IR データバス 376 に提供され始め、その提供が「マスタ」デバイスから始まり、ランクにおける各ステートマシンエンジン 14 を通して順番に（例えば、F 0、F 1、F 2、F 3）継続していることを示すものである。時点 408 に、「マスタ」ステートマシンエンジン 14 F 0 は、データ（例えば、論理グループ 0 用に意図された第一のデータバイト）を IR データバス 376 に提供し、論理ハイを IRDQS 384 及び / または IRDQS # 386 に出力する。その後、時点 409 に、「マスタ」ステートマシンエンジン 14 F 0 は、データ（例えば、論理グループ 1 用に意図された第二のデータバイト）を IR データバス 376 に提供し、論理ハイを IRDQS 384 及び / または IRDQS # 386 に出力する。他のステートマシンエンジン 14 は、その後、時点 388 ~ 時点 394 に関して前述したように IR データバス 376 にデータを提供することができる。 20

#### 【0113】

ステートマシンエンジン 14 の動作中、RSYNC 380 及び IRDV 382 を使用して全てのステートマシンエンジン 14 がともに同期された同期様式で、データを IR データバス 376 に提供してもよい。さらに、ステートマシンエンジン 14 の各々は、モジュール 340 を構成するランクにおけるデバイス内のそれ自身の位置を示す命令を格納する。それゆえ、ステートマシンエンジン 14 の各々は、データ格納及びデータ出力のタイミングをとって、IR データバス 376 を使用する順序の正しいデータ転送を容易にすることによって、データを適切な時に提供及び格納する。認識できるように、データ転送処理中にはエラーが発生することがある。したがって、「マスタ」ステートマシンエンジン 14 F 0 は、任意の時に IRDV 382 を論理口ウに遷移することによって、エラーが解決するまで、IR データバス 376 へのデータ転送を停止（例えば、ブロック、ストール、遅延など）してもよい。さらにまた、ステートマシンエンジン 14 F 1、F 2 及び F 3 の任意の 1 つは、RSYNC 380 を論理口ウに導くことによって、IR データバス 376 へのデータ転送を停止すべきであるという指示を、「マスタ」ステートマシンエンジン 14 F 0 に提供することができる。 40

#### 【0114】

図 18 は、複数の論理グループに組織化された複数の物理的なステートマシンエンジン 14 のデータバッファ 132 に格納されたデータの一実施例を図示する。具体的には、図 18 は、8 つの論理グループに組織化された 8 つのステートマシンエンジン 14 (F 0、F 1、F 2、F 3、F 4、F 5、F 6 及び F 7) を有するモジュール 340 における DD 50

R 3 インターフェイスビューの表 4 1 0 を図示する。8 つの論理グループを持つそのような構成は、8 つのステートマシンエンジン 1 4 を有するモジュール 3 4 0 において最大データスループットを提供することができる。図示する実施形態においては、ステートマシンエンジン 1 4 は、6 4 本のデータ線を有する DDR3 バスインターフェイス 1 3 0 に結合される。したがって、8 本のデータ線が各ステートマシンエンジン 1 4 に結合される。しかしながら、本明細書に記述するのと同じ概念を、各ステートマシンエンジン 1 4 に結合された任意の適切な数のデータ線を持つ任意の適切な DDR3 バスインターフェイス 1 3 0 に適用することができる。さらにまた、本明細書に記述するのと同じ概念を、任意の適切な数のステートマシンエンジン 1 4 を有するモジュール 3 4 0 に適用することができる。表 4 1 0 は、モジュール 3 4 0 がプロセッサ 1 2 から受信し、ステートマシンエンジン 1 4 の間でオフセット様式において格納するデータを図示する。モジュール 3 4 0 にデータを格納するのに、プロセッサ 1 2 を使用してデータを再配列するのではなく、オフセット様式においてモジュール 3 4 0 を使用することによって、プロセッサ 1 2 を他の機能の実行に自由に使用することができる。

#### 【 0 1 1 5 】

表 4 1 0 は、プロセッサ 1 2 から 6 4 本のデータ線に書き込む数を示す書き込みカウント列 4 1 2 、及びプロセッサ 1 2 からバイトを受信した時に基づく、番号順における個々のバイト数を示すバイト数列 4 1 4 を含む。バイト数列 4 1 4 は、それぞれのステートマシンエンジン 1 4 ( F 0 、 F 1 、 F 2 、 F 3 、 F 4 、 F 5 、 F 6 及び F 7 ) のデータバッファシステム 1 5 9 に格納された具体的なバイトを示す列 F 0 、 F 1 、 F 2 、 F 3 、 F 4 、 F 5 、 F 6 及び F 7 を含む。表 4 1 0 は、バイト数列 4 1 4 に示されるバイトが格納される、データバッファシステム 1 5 9 のデータバッファ A 及び B 1 3 2 の各々内のアドレスを示すデータバッファアドレス列 4 1 6 、及びステートマシンエンジン 1 4 の特定の論理グループによって処理されるデータを示すターゲット列 4 1 8 も含む。例えば、プロセッサ 1 2 から 6 4 本のデータ線への第一の書き込み期間 ( 例えば、書き込みカウント列 4 1 2 における 0 ) に、プロセッサ 1 2 は、 DDR3 アドレスバス 3 4 2 に、データバッファシステム 1 5 9 の第一のアドレスのアドレス選択 ( 例えば、プロセッサ書き込みアドレス 0 ) 、及び対応する 8 バイトのデータを提供する。この結果、第一のデータバイト ( 例えば、 0 ) を、第一のステートマシンエンジン 1 4 ( 例えば、 F 0 ) の第一のデータバッファアドレス 4 1 6 ( 例えば、 0 ) において第一のステートマシンエンジン 1 4 ( 例えば、 F 0 ) に格納し、第二のデータバイト ( 例えば、 1 ) を、第二のステートマシンエンジン 1 4 ( 例えば、 F 1 ) の同一の第一のデータバッファアドレス 4 1 6 ( 例えば、 0 ) において第二のステートマシンエンジン 1 4 ( 例えば、 F 1 ) に格納し、第三のデータバイト ( 例えば、 2 ) を、第三のステートマシンエンジン 1 4 ( 例えば、 F 2 ) の同一の第一のデータバッファアドレス 4 1 6 ( 例えば、 0 ) において第三のステートマシンエンジン 1 4 ( 例えば、 F 2 ) に格納し、第四のデータバイト ( 例えば、 3 ) を、第四のステートマシンエンジン 1 4 ( 例えば、 F 3 ) の同一の第一のデータバッファアドレス 4 1 6 ( 例えば、 0 ) において第四のステートマシンエンジン 1 4 ( 例えば、 F 3 ) に格納し、第五のデータバイト ( 例えば、 4 ) を、第五のステートマシンエンジン 1 4 ( 例えば、 F 4 ) の同一の第一のデータバッファアドレス 4 1 6 ( 例えば、 0 ) において第五のステートマシンエンジン 1 4 ( 例えば、 F 4 ) に格納し、第六のデータバイト ( 例えば、 5 ) を、第六のステートマシンエンジン 1 4 ( 例えば、 F 5 ) の同一の第一のデータバッファアドレス 4 1 6 ( 例えば、 0 ) において第六のステートマシンエンジン 1 4 ( 例えば、 F 5 ) に格納し、第七のデータバイト ( 例えば、 6 ) を、第七のステートマシンエンジン 1 4 ( 例えば、 F 6 ) の同一の第一のデータバッファアドレス 4 1 6 ( 例えば、 0 ) において第七のステートマシンエンジン 1 4 ( 例えば、 F 6 ) に格納し、そして第八のデータバイト ( 例えば、 7 ) を、第八のステートマシンエンジン 1 4 ( 例えば、 F 7 ) の同一の第一のデータバッファアドレス 4 1 6 ( 例えば、 0 ) において第八のステートマシンエンジン 1 4 ( 例えば、 F 7 ) に格納することができる。

#### 【 0 1 1 6 】

10

20

30

40

50

別の実施例として、プロセッサ 12 から 64 本のデータ線への第二の書き込み期間（例えば、書き込みカウント列 412 における 1）に、プロセッサ 12 は、DDR3 アドレスバス 342 に、データバッファシステム 159 の第二のアドレスのアドレス選択（例えば、プロセッサ書き込みアドレス 1）、及び対応する 8 バイトのデータを提供する。この結果、第一のデータバイト（例えば、8）を、第一のステートマシンエンジン 14（例えば、F0）の第二のデータバッファアドレス 416（例えば、8）において第一のステートマシンエンジン 14（例えば、F0）に格納し、第二のデータバイト（例えば、9）を、第二のステートマシンエンジン 14 F1 の同一の第二のデータバッファアドレス 416（例えば、8）において第二のステートマシンエンジン 14（例えば、F1）に格納し、第三のデータバイト（例えば、10）を、第三のステートマシンエンジン 14（例えば、F2）の同一の第二のデータバッファアドレス 416（例えば、8）において第三のステートマシンエンジン 14（例えば、F2）に格納し、第四のデータバイト（例えば、11）を、第四のステートマシンエンジン 14（例えば、F3）の同一の第二のデータバッファアドレス 416（例えば、8）において第四のステートマシンエンジン 14（例えば、F3）に格納し、第五のデータバイト（例えば、12）を、第五のステートマシンエンジン 14（例えば、F4）の同一の第二のデータバッファアドレス 416（例えば、8）において第五のステートマシンエンジン 14（例えば、F4）に格納し、第六のデータバイト（例えば、13）を、第六のステートマシンエンジン 14（例えば、F5）の同一の第二のデータバッファアドレス 416（例えば、8）において第六のステートマシンエンジン 14（例えば、F5）に格納し、第七のデータバイト（例えば、14）を、第七のステートマシンエンジン 14（例えば、F6）の同一の第二のデータバッファアドレス 416（例えば、8）において第七のステートマシンエンジン 14（例えば、F6）に格納し、そして第八のデータバイト（例えば、15）を、第八のステートマシンエンジン 14（例えば、F7）の同一の第二のデータバッファアドレス 416（例えば、8）において第八のステートマシンエンジン 14（例えば、F7）に格納することができる。2つの実施例に図示するように、データバッファアドレス 416 は、プロセッサ 12 からの第一の書き込みとプロセッサ 12 からの第二の書き込みとの間で、8（例えば、モジュール 340 のランクにおける論理グループの数と等しい数）だけ変化した。プロセッサ 12 は、一連のプロセッサ 12 がデータバッファシステム 159 に書き込むものとして、線形アドレスプロックへの書き込みを継続するが、データバッファアドレス 416 は、第一の論理グループ（論理グループ 0）用に意図された全てのデータがモジュール 340 に提供されるまで、8ずつ自動的に増加するように継続する。図示するように、データは、同様の様式で他の論理グループにも提供される。

#### 【0117】

例えば、プロセッサ 12 から 64 本のデータ線への 513 番目の書き込み期間（例えば、書き込みカウント列 412 における 512）に、プロセッサ 12 は、DDR3 アドレスバス 342 に、データバッファシステム 159 の第三のアドレスのアドレス選択（例えば、プロセッサ書き込みアドレス 512）、及び対応する 8 バイトのデータを提供する。この結果、第一のデータバイト（例えば、4096）を、第一のステートマシンエンジン 14（例えば、F0）の第三のデータバッファアドレス 416（例えば、1）において第一のステートマシンエンジン 14（例えば、F0）に格納し、第二のデータバイト（例えば、4097）を、第二のステートマシンエンジン 14（例えば、F1）の同一の第三のデータバッファアドレス 416（例えば、1）において第二のステートマシンエンジン 14（例えば、F1）に格納し、第三のデータバイト（例えば、4098）を、第三のステートマシンエンジン 14（例えば、F2）の同一の第三のデータバッファアドレス 416（例えば、1）において第三のステートマシンエンジン 14（例えば、F2）に格納し、第四のデータバイト（例えば、4099）を、第四のステートマシンエンジン 14（例えば、F3）の同一の第三のデータバッファアドレス 416（例えば、1）において第四のステートマシンエンジン 14（例えば、F3）に格納し、第五のデータバイト（例えば、4100）を、第五のステートマシンエンジン 14（例えば、F4）の同一の第三のデータ

バッファアドレス 416 (例えば、1)において第五のステートマシンエンジン 14 (例えば、F4)に格納し、第六のデータバイト (例えば、4101)を、第六のステートマシンエンジン 14 (例えば、F5)の同一の第三のデータバッファアドレス 416 (例えば、1)において第六のステートマシンエンジン 14 (例えば、F5)に格納し、第七のデータバイト (例えば、4102)を、第七のステートマシンエンジン 14 (例えば、F6)の同一の第三のデータバッファアドレス 416 (例えば、1)において第七のステートマシンエンジン 14 (例えば、F6)に格納し、そして第八のデータバイト (例えば、4103)を、第八のステートマシンエンジン 14 (例えば、F7)の同一の第三のデータバッファアドレス 416 (例えば、1)において第八のステートマシンエンジン 14 (例えば、F7)に格納することができる。

10

## 【0118】

表410が、書き込みカウント列412からの書き込み0～511の全てが第一のステートマシンエンジン 14 (例えば、F0)を含む論理グループ0用に意図されたデータを含むことを示すことが留意されるべきである。さらにまた、表410が、書き込みカウント列412からの書き込み512～1023の全てが第二のステートマシンエンジン 14 (例えば、F1)を含む論理グループ1用に意図されたデータを含み、それ以降も同様であることを示すことが留意されるべきである。

## 【0119】

図19は、複数の論理グループに組織化された複数の物理的なステートマシンエンジン 14 のプロセスバッファ 134 に格納されたデータの一実施例を図示する。具体的には、図19は、8つの論理グループに組織化された8つのステートマシンエンジン 14 (F0、F1、F2、F3、F4、F5、F6 及び F7)を有するモジュール 340 におけるIRデータバス 376 ビューの表421を図示する。表421は、プロセッサ 12 から受信した個々のバイト数を示すバイト数列 414 を含む。バイト数列 414 は、それぞれのステートマシンエンジン 14 (F0、F1、F2、F3、F4、F5、F6 及び F7)のデータバッファシステム 159 に格納された具体的なバイトを示す列 F0、F1、F2、F3、F4、F5、F6 及び F7 を含む。表421は、バイト数列 414 に示されるバイトが格納される、データバッファシステム 159 のデータバッファ A 及び B 132 のいずれか内のアドレスを示すデータバッファアドレス列 416 も含む。さらにまた、表421は、プロセスバッファ 134 に格納された個々のバイト数を示すバイト数列 426 を含む。バイト数列 426 は、それぞれのステートマシンエンジン 14 (F0、F1、F2、F3、F4、F5、F6 及び F7)のプロセスバッファ 134 に格納された具体的なバイトを示す列 F0、F1、F2、F3、F4、F5、F6 及び F7 を含む。例えば、プロセスバッファ A 行 428 は、ステートマシンエンジン 14 の第一のプロセスバッファに格納された具体的なバイトを示し、プロセスバッファ B 行 430 は、ステートマシンエンジン 14 の第二のプロセスバッファに格納された具体的なバイトを示す。

20

## 【0120】

したがって、表421は、データが、オフセット様式でデータバッファシステム 159 に格納され、連続した様式でプロセスバッファ 134 に提供されることを示している。例えば、IRデータバス 376 における第一のデータバースト期間に、ステートマシンエンジン 14 F0 は、プロセッサ 12 から受信したバイト 0、4096、8192、12288、16384、20480、24576 及び 28672 (例えば、論理グループの各々における第一のバイト) を提供することができる。IRデータバス 376 における第二のバースト期間に、ステートマシンエンジン 14 F1 は、プロセッサ 12 から受信したバイト 1、4097、8193、12289、16385、20481、24577 及び 28673 (例えば、論理グループの各々における第二のバイト) を提供することができ、それ以降も同様である。ステートマシンエンジン 14 の各々は、そのステートマシンエンジンの論理グループに対応するデータバーストからのバイトを格納するように構成される。例えば、ステートマシンエンジン 14 F0 は、IRデータバス 376 に提供される各データバーストの第一のバイトを格納するように構成され、ステートマシンエンジン 14 F1

30

40

50

は、IRデータバス376に提供される各データバーストの第二のバイトを格納するように構成され、それ以降も同様である。したがって、データバッファシステム159に格納されるときにオフセット様式で格納されたデータは、各ステートマシンエンジン14がその意図されたデータを解析のための正しい順序で受信できるように、IRデータバス376を通じてステートマシンエンジン14に提供される。

#### 【0121】

図20は、1つの論理グループに組織化された複数の物理的なステートマシンエンジン14のデータバッファ132に格納されたデータの一実施例を図示する。具体的には、図20は、1つの論理グループに組織化された8つのステートマシンエンジン14(F0、F1、F2、F3、F4、F5、F6及びF7)を有するモジュール340におけるDDR3インターフェイスビューの表432を図示する。図示するように、データは、プロセッサ12によって提供され、ステートマシンエンジン14(F0、F1、F2、F3、F4、F5、F6及びF7)に格納される。全てのステートマシンエンジン14は、単一の論理グループ(例えば、論理グループ0)の一部である。図示する実施形態においては、モジュール340によって受信されたデータは、データバッファシステム159に連続して格納される。

#### 【0122】

図21は、1つの論理グループに組織化された複数の物理的なステートマシンエンジン14のプロセスバッファ134に格納されたデータの一実施例を図示する。具体的には、図21は、1つの論理グループに組織化された8つのステートマシンエンジン14(F0、F1、F2、F3、F4、F5、F6及びF7)を有するモジュール340におけるIRデータバス376ビューの表434を図示する。IRデータバス376における第一のデータバースト期間において、データバースト期間に8バイトが提供されるように、ステートマシンエンジン14F0は、プロセッサ12から受信したバイト0(例えば、論理グループにおける第一のバイト)を提供でき、ステートマシンエンジン14F1は、プロセッサ12から受信したバイト1(例えば、論理グループにおける第二のバイト)を提供でき、それ以降も同様である。ステートマシンエンジン14の各々は、そのステートマシンエンジンの論理グループに対応するデータバーストからのバイトを格納するように構成される。例えば、全てのステートマシンエンジン14F0、F1、F2、F3、F4、F5、F6及びF7が、IRデータバス376に提供される各データバーストの全てのバイトを格納するように構成される。したがって、各ステートマシンエンジン14がパラレル解析用の全てのデータを受信できるように、データバッファシステム159に格納される全てのデータが、IRデータバス376を通じてステートマシンエンジン14に提供される。認識できるように、図18～図21は、4Kバイトを有するバッファを図示するが、他の実施形態では、任意の適切なサイズ(例えば、8Kバイト、16Kバイト、32Kバイト、64Kバイトなど)を有するバッファを含んでもよい。

#### 【0123】

本明細書中に使用されるような、装置という用語は、單一モジュールであってもよいし、1つ以上のモジュールを含むシステムであってもよい。本発明は、種々の変更及び代替形態を受け入れることができるが、特定の実施形態を、図面に例証として示し、本明細書に詳細に記述した。しかしながら、本発明が、開示した特定の形態に限定されないことを意図することが理解されるはずである。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ような本発明の精神及び範囲内に収まる全ての変更、均等物及び代替手段を包含することができる。

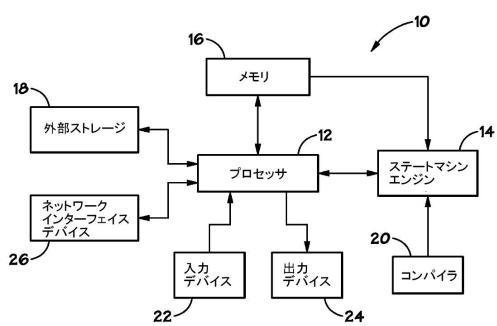
10

20

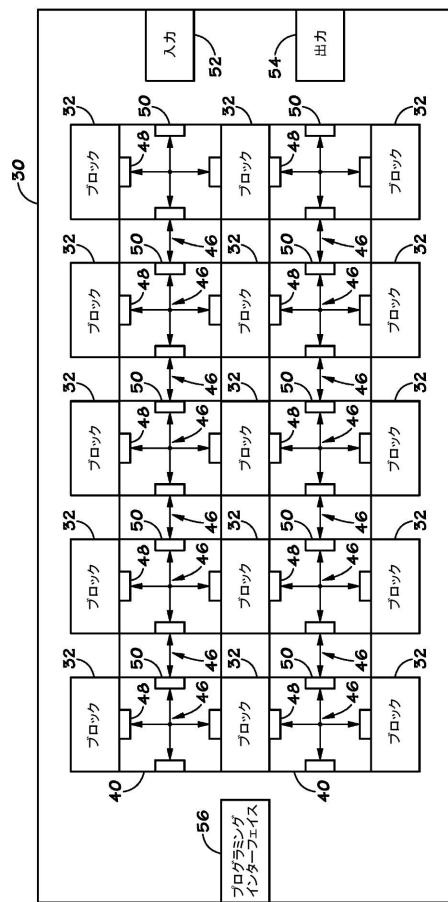
30

40

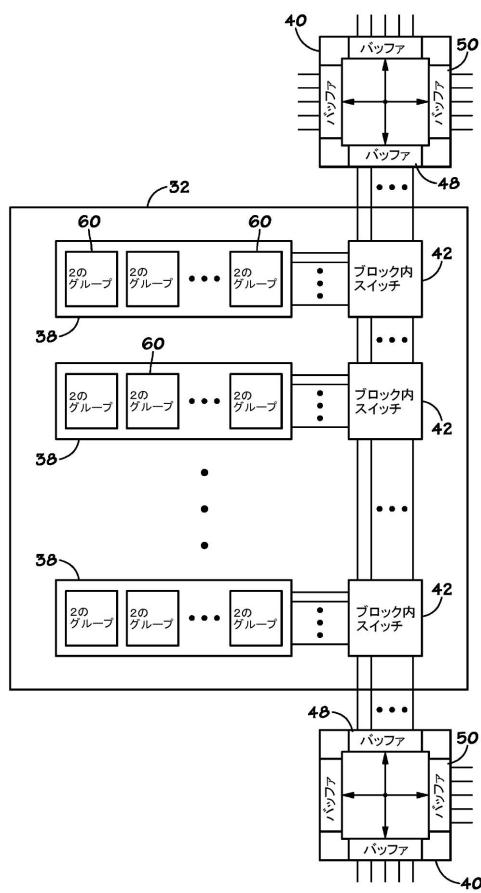
【図1】



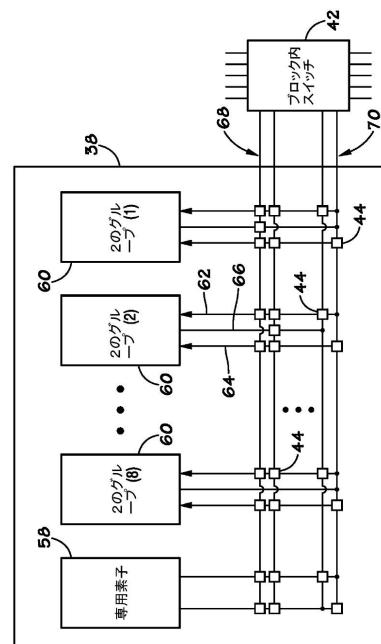
【図2】



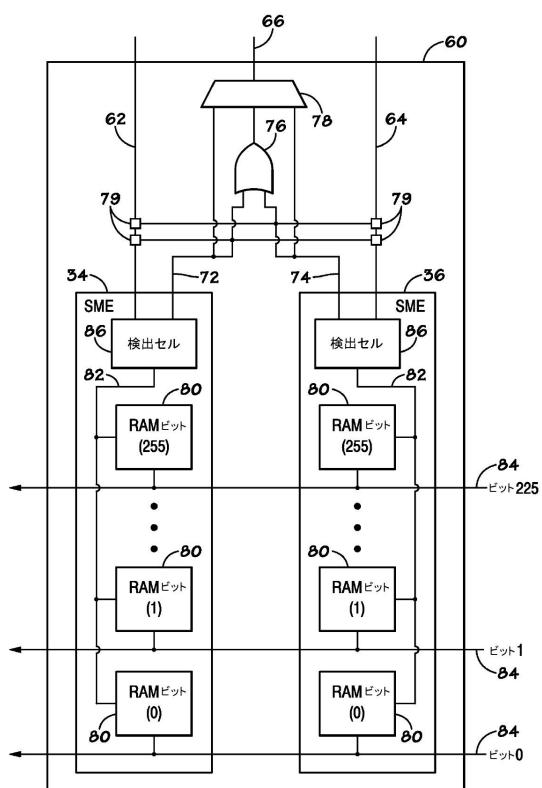
【図3】



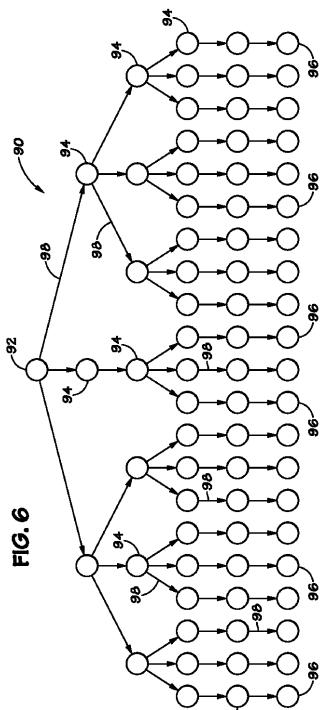
【図4】



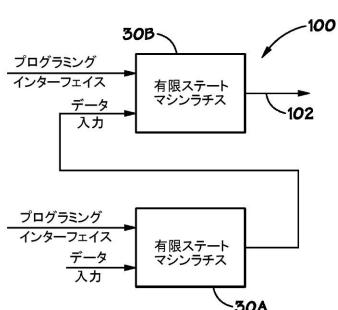
【 図 5 】



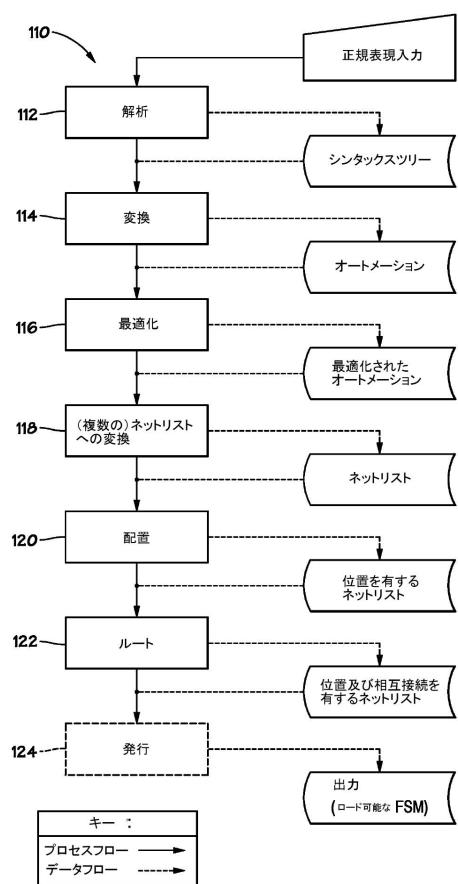
【図6】



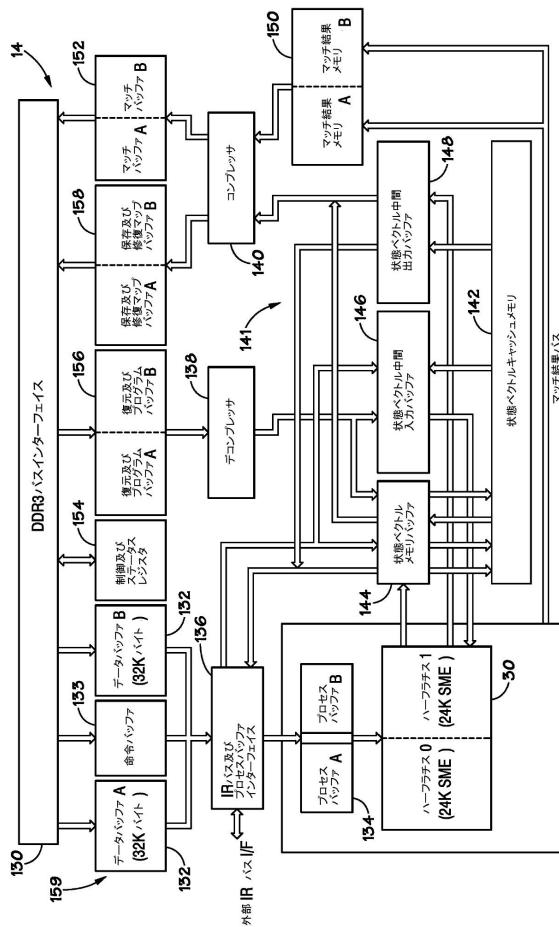
【図7】



【 四 8 】

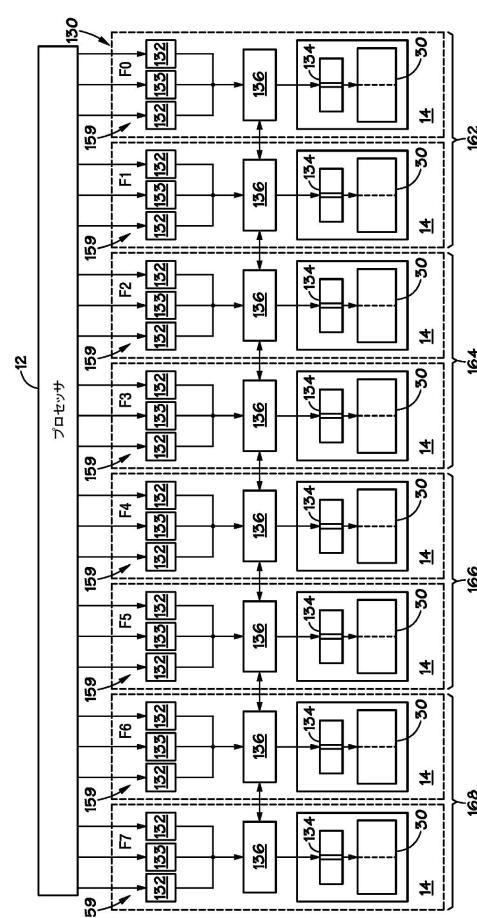


【 四 9 】

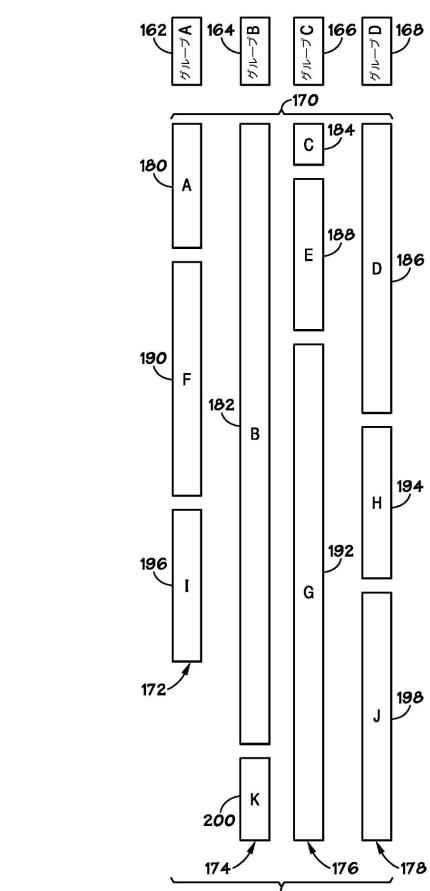


### 【図11】

【図10】



### 【図12】

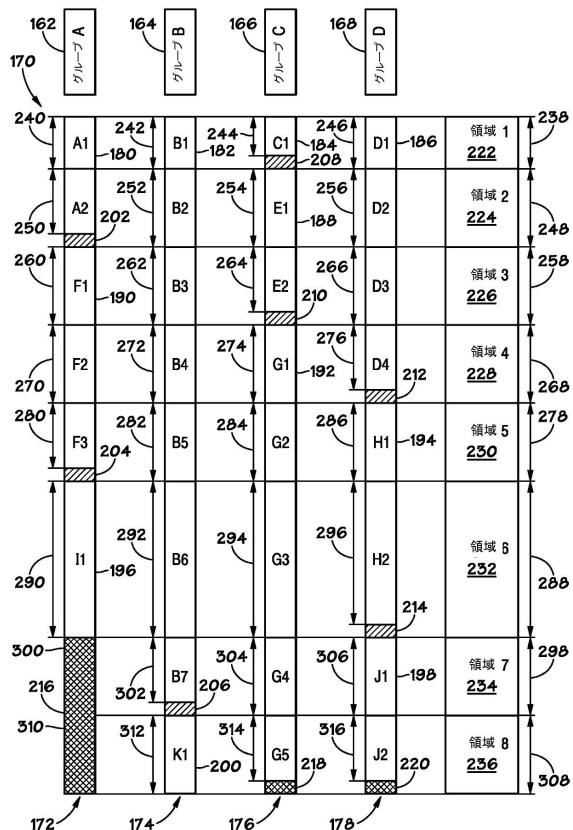


The diagram illustrates a polymer chain structure with the following segments and their molecular weights:

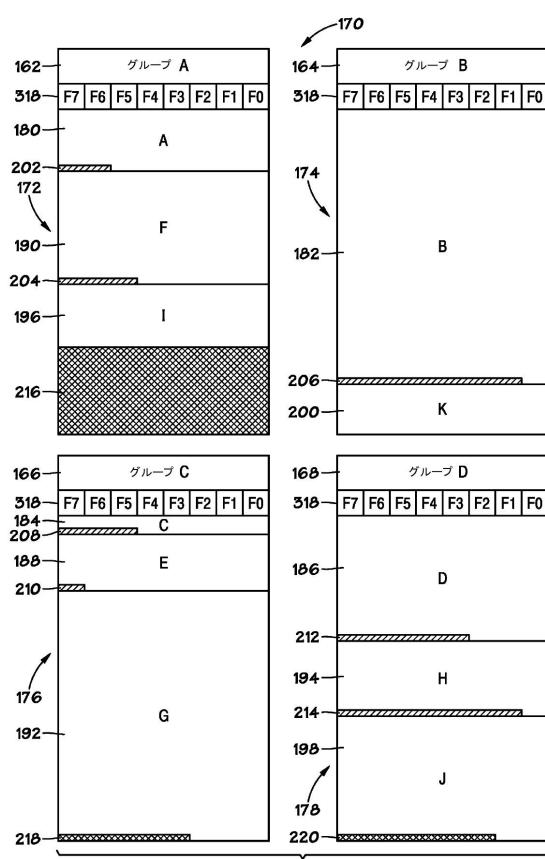
- Segment A:** Molecular weight 180.
- Segment B:** Molecular weight 182.
- Segment C:** Molecular weight 184.
- Segment D:** Molecular weight 186.
- Segment E:** Molecular weight 208.
- Segment F:** Molecular weight 190.
- Segment G:** Molecular weight 192.
- Segment H:** Molecular weight 194.
- Segment I:** Molecular weight 196.
- Segment J:** Molecular weight 198.
- Segment K:** Molecular weight 200.

Arrows at the top indicate the sequence: 162 → A, 164 → B, 166 → C, 168 → D, 170 → E, 172 → F, 174 → G, 176 → H, 178 → I, 180 → J, and 182 → K.

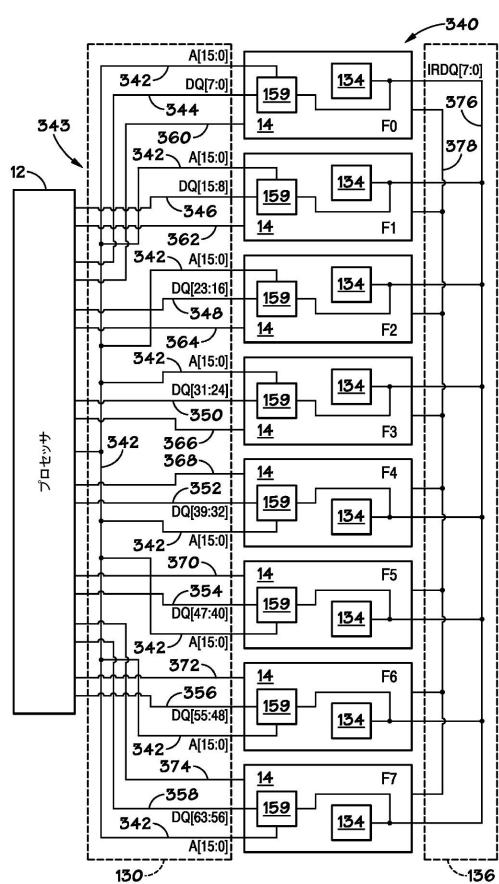
【図13】



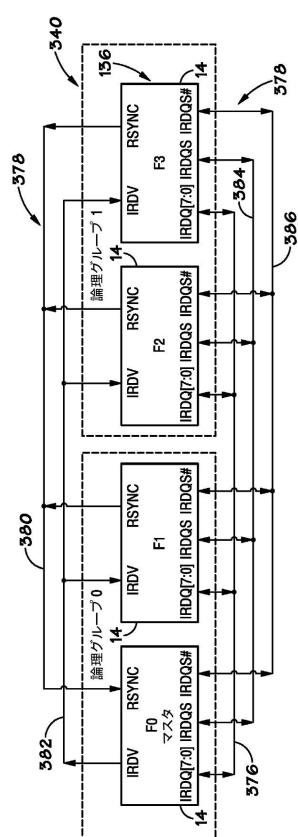
【図14】



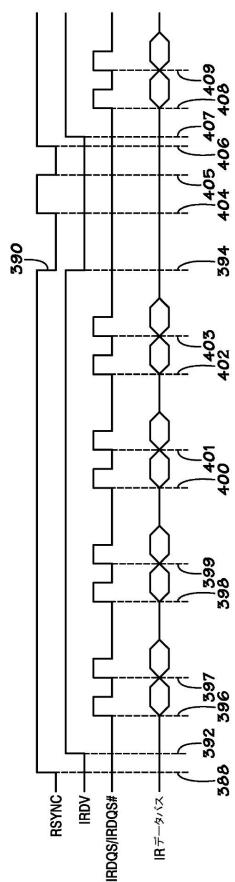
【図15】



【図16】



【図17】



【図18】

DDR3 インターフェイスピューバイト数								416	418
書き込みカウント	F0	F1	F2	F3	F4	F5	F6	データバッファアドレス	ターゲット
0	0	1	2	3	4	5	6	7	0
1	8	9	10	11	12	13	14	15	8
2	17	18	19	20	21	22	23	24	17
3	25	26	27	28	29	30	31	32	24
508	4064	4065	4066	4067	4068	4069	4070	4071	4064
509	4072	4073	4074	4075	4076	4077	4078	4079	4072
510	4080	4081	4082	4083	4084	4085	4086	4087	4080
511	4088	4089	4090	4091	4092	4093	4094	4095	4088
512	4096	4097	4098	4099	4100	4101	4102	4103	1
513	4104	4105	4106	4107	4108	4109	4110	4111	9
514	4112	4113	4114	4115	4116	4117	4118	4119	17
515	4120	4121	4122	4123	4124	4125	4126	4127	25
1020	8160	8161	8162	8163	8164	8165	8166	8167	4065
1021	8168	8169	8170	8170	8172	8173	8174	8175	4073
1022	8176	8177	8178	8179	8180	8181	8182	8183	4073
1023	8184	8185	8186	8187	8188	8189	8190	8191	4089
1024	8192	8193	8194	8195	8196	8197	8198	8199	2
1025	8200	8201	8202	8203	8204	8205	8206	8207	10
1026	8208	8209	8210	8211	8212	8213	8214	8215	18
1027	8216	8217	8218	8219	8220	8221	8222	8223	26
1532	12258	12257	12259	12258	12260	12261	12262	12263	4066
1533	12264	12265	12266	12267	12268	12269	12270	12271	4074
1534	12272	12273	12274	12275	12276	12277	12278	12279	4082
1535	12280	12281	12282	12283	12284	12285	12286	12287	4090
1536	12288	12289	12290	12291	12292	12293	12294	12295	3
1537	12296	12297	12298	12299	12300	12301	12302	12303	11
1538	12304	12305	12306	12307	12308	12309	12310	12311	19
1539	12312	12313	12314	12315	12316	12317	12318	12319	27
2044	16352	16353	16354	16355	16356	16357	16358	16359	4067
2045	16360	16361	16362	16363	16364	16365	16366	16367	4075
2046	16368	16369	16370	16371	16372	16373	16374	16375	4083
2047	16376	16377	16378	16379	16380	16381	16382	16383	4091
2048	16384	16385	16386	16387	16388	16389	16390	16391	2
2049	16392	16393	16394	16395	16396	16397	16398	16399	10
2050	16400	16401	16402	16403	16404	16405	16406	16407	20
2051	16408	16409	16410	16411	16412	16413	16414	16415	18
2556	20448	20449	20450	20451	20452	20453	20454	20455	4068
2557	20456	20457	20458	20459	20460	20461	20462	20463	4076
2558	20474	20475	20476	20477	20478	20479	20478	20479	4082
2559	20481	20482	20483	20484	20485	20486	20487	20488	5
2561	20488	20489	20490	20491	20492	20493	20494	20495	13
2562	20496	20497	20498	20499	20500	20501	20502	20503	21
2563	20504	20505	20506	20507	20508	20509	20510	20511	29
3068	24544	24545	24546	24547	24548	24549	24550	24551	4069
3069	24552	24553	24554	24555	24556	24557	24558	24559	4077
3070	24560	24561	24562	24563	24564	24565	24566	24567	4085
3071	24568	24569	24570	24571	24572	24573	24574	24575	4093
3072	24576	24577	24578	24579	24580	24581	24582	24583	6
3073	24584	24585	24586	24587	24588	24589	24590	24591	13
3074	24592	24593	24594	24595	24596	24597	24598	24599	22
3075	24600	24601	24602	24603	24604	24605	24606	24607	30
3580	26840	26841	26842	26843	26844	26845	26846	26847	4070
3581	26848	26849	26850	26851	26852	26853	26854	26855	4082
3582	26856	26857	26858	26859	26860	26861	26862	26863	4086
3583	26864	26865	26866	26867	26868	26869	26870	26871	4094
3584	26872	26873	26874	26875	26876	26877	26878	26879	15
3585	26880	26881	26882	26883	26884	26885	26886	26887	15
3586	26888	26889	26890	26891	26892	26893	26894	26895	15
3587	26896	26897	26898	26899	26900	26901	26902	26903	31
3588	26904	26905	26906	26907	26908	26909	26910	26911	4077
4083	30736	30737	30738	30739	30740	30741	30742	30743	4071
4084	32752	32753	32754	32755	32756	32757	32758	32759	4068
4095	32760	32761	32762	32763	32764	32765	32766	32767	4095

【 図 1 9 】

426

プロセスバッファビューバイト数								426
	F0	F1	F2	F3	F4	F5	F6	F7
プロセスバッファA	0	4096	8192	12288	16384	20480	24576	28672
	1	4097	8193	12289	16385	20481	24577	28673
	2	4098	8194	12290	16386	20482	24578	28674
	3	4099	8195	12291	16387	20483	24579	28675
	4	4100	8196	12292	16388	20484	24580	28676
	5	4101	8197	12293	16389	20485	24581	28677
	6	4102	8198	12294	16390	20486	24582	28678
プロセスバッファB	7	4103	8199	12295	16391	20487	24583	28679
430	8	4104	8200	12296	16392	20488	24584	28680
	9	4105	8201	12297	16393	20489	24585	28681
	10	4106	8202	12298	16394	20490	24586	28682
	11	4107	8203	12299	16395	20491	24587	28683
	12	4108	8204	12300	16396	20492	24588	28684
	13	4109	8205	12301	16397	20493	24589	28685
	14	4110	8206	12302	16398	20494	24590	28686
	15	4111	8207	12303	16399	20495	24591	28687
F0	F1	F2	F3	F4	F5	F6	F7	データバッファアドレス
0	1	2	3	4	5	6	7	0
4096	4097	4098	4099	4100	4101	4102	4103	1
8192	8193	8194	8195	8196	8197	8198	8199	2
12288	12289	12290	12291	12292	12293	12294	12295	3
16384	16385	16386	16387	16388	16389	16390	16391	4
20480	20481	20482	20483	20484	20485	20486	20487	5
24576	24577	24578	24579	24580	24581	24582	24583	6
28672	28673	28674	28675	28676	28677	28678	28679	7
8	9	10	11	12	13	14	15	8
4104	4105	4106	4107	4108	4109	4110	4111	9
8200	8201	8202	8203	8204	8205	8206	8207	10
12296	12297	12298	12299	12300	12301	12302	12303	11
16392	16393	16394	16395	16396	16397	16398	16399	12
20488	20489	20490	20491	20492	20493	20494	20495	13
24584	24585	24586	24587	24588	24589	24590	24591	14
28680	28681	28682	28683	28684	28685	28686	28687	15

データバッファビューバイト数

414

## 【図20】

DDR3インターフェイスビュー								416	418
書き込みカウント	バイト数							データパックアドレス	
	F0	F1	F2	F3	F4	F5	F6	F7	データアドレス
0	0	1	2	3	4	5	6	7	0
1	0	8	9	10	11	12	13	14	15
2	16	17	18	19	20	21	22	23	24
3	24	25	26	27	28	29	30	31	3
4092	32736	32737	32738	32739	32740	32741	32742	32743	32744
4093	32744	32745	32746	32747	32748	32749	32750	32751	32752
4352	32756	32757	32758	32759	32760	32761	32762	32763	32764

【図21】

426

428

434

450

416

414

プロセスバッファビューバイト数							
F0	F1	F2	F3	F4	F5	F6	F7
0	0	0	0	0	0	0	0
1	2	2	2	1	2	1	2
2	3	3	3	3	3	3	3
3	4	4	4	4	4	4	4
4	5	5	5	5	5	5	5
5	6	6	6	6	6	6	6
6	7	7	7	7	7	7	7

プロセスバッファB							
F0	F1	F2	F3	F4	F5	F6	F7
8	8	8	8	8	8	8	8
9	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
11	11	11	11	11	11	11	11
12	12	12	12	12	12	12	12
13	13	13	13	13	13	13	13
14	14	14	14	14	14	14	14
15	15	15	15	15	15	15	15

データバッファビューバイト数							
F0	F1	F2	F3	F4	F5	F6	F7
データバッファアドレス							
0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71
72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87
88	89	90	91	92	93	94	95
96	97	98	99	100	101	102	103
104	105	106	107	108	109	110	111
112	113	114	115	116	117	118	119
120	121	122	123	124	125	126	127

---

フロントページの続き

(72)発明者 ノイズ, ハロルド ピー.

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 漆原 孝治

(56)参考文献 特開2007-018105 (JP, A)

特開2011-191901 (JP, A)

国際公開第2011/081799 (WO, A1)

(58)調査した分野(Int.Cl., DB名)

G 06 F 15 / 82