

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6449287号
(P6449287)

(45) 発行日 平成31年1月9日(2019.1.9)

(24) 登録日 平成30年12月14日(2018.12.14)

(51) Int. Cl.			F I		
G06F	12/00	(2006.01)	G06F	12/00	560F
G06F	9/34	(2006.01)	G06F	12/00	550K
G06F	12/06	(2006.01)	G06F	9/34	350Z
G11C	11/401	(2006.01)	G06F	12/06	530A
			G11C	11/401	

請求項の数 15 (全 18 頁)

(21) 出願番号	特願2016-535174 (P2016-535174)	(73) 特許権者	595168543
(86) (22) 出願日	平成26年12月1日(2014.12.1)		マイクロン・テクノロジー・インコーポレ ーテッド
(65) 公表番号	特表2017-504870 (P2017-504870A)		アメリカ合衆国83707, アイダホ州, ボイズ, サウス・フェデラル・ウェイ 8 000番
(43) 公表日	平成29年2月9日(2017.2.9)	(74) 代理人	100074099
(86) 国際出願番号	PCT/US2014/067927		弁理士 大菅 義之
(87) 国際公開番号	W02015/084728	(74) 代理人	100106851
(87) 国際公開日	平成27年6月11日(2015.6.11)		弁理士 野村 泰久
審査請求日	平成28年6月30日(2016.6.30)	(72) 発明者	クレウイツ, ケネス エム.
審判番号	不服2018-1674 (P2018-1674/J1)		アメリカ合衆国, カリフォルニア州 95 682, キャメロン パーク, アッシュレ ンド コート 408
審判請求日	平成30年2月7日(2018.2.7)		
(31) 優先権主張番号	14/094, 273		
(32) 優先日	平成25年12月2日(2013.12.2)		
(33) 優先権主張国	米国 (US)		

最終頁に続く

(54) 【発明の名称】 自律的メモリの方法及びシステム

(57) 【特許請求の範囲】

【請求項1】

命令一式を自律的メモリデバイスにおいて受信することと、
前記命令一式を前記自律的メモリデバイスにおいて実行することと、
前記命令一式と、前記命令一式に応答して前記自律的メモリデバイスにおいて得られた
任意のデータとを併せたパケットを前記自律的メモリデバイスにおいて生成することと、
前記パケットを前記自律的メモリデバイスから送信することと、
を含む方法。

【請求項2】

前記命令一式を前記自律的メモリデバイスにおいて受信することは、前記命令一式を前
記自律的メモリデバイスと接続するネットワークから受信することを含み、前記パケット
を前記自律的メモリデバイスから送信することは、前記パケットを前記ネットワークへ送
信することを含む、請求項1に記載の方法。

【請求項3】

前記命令一式を受信することは、前記命令一式を含むパケットを受信することを含み、
前記方法はさらに、
前記受信した命令一式に対応付けられた初期プログラムカウンタ値をプログラムカウン
タにロードすることと、
前記命令一式を命令メモリにロードすることと、
前記命令一式に対応付けられた初期条件一式をレジスタファイルにロードすることと、

10

20

を含む前記受信したパケットを構文解析することを含む請求項 1 に記載の方法。

【請求項 4】

前記命令一式を実行することは、
前記命令一式の第 1 命令を実行後、新たなプログラムカウンタ値を算出することと、
前記新たなプログラムカウンタ値を前記プログラムカウンタに記憶させることと、
を含む、請求項 3 に記載の方法。

【請求項 5】

前記命令一式を実行することは、第 1 命令を第 1 実行ユニットにおいて、かつ第 2 命令を第 2 実行ユニットにおいて実行することを含み、前記第 1、第 2 命令の前記実行は実質的に並列である、請求項 1 に記載の方法。

10

【請求項 6】

前記自律的メモリデバイスは複数のノードのうちの第 1 ノードであり、前記パケットを前記自律的メモリデバイスから送信することは、前記パケットを前記複数のノードのうちの第 2 ノードへ送信することを含む、請求項 1 に記載の方法。

【請求項 7】

初期条件を前記複数のノードのうちの第 3 ノードから受信することと、
前記初期条件をファイルレジスタに記憶させることと、
をさらに含む、請求項 6 に記載の方法。

【請求項 8】

前記命令一式はフェンスフラグを含み、前記命令一式を記憶させることは、
1 つまたは複数の命令を前記フェンスフラグの前に命令メモリに、かつ 1 つまたは複数の命令を前記フェンスフラグに続いて前記命令メモリに記憶させること、
を含む、請求項 1 に記載の方法。

20

【請求項 9】

前記 1 つまたは複数の命令を前記フェンスフラグの前に第 1 実行ユニットにおいて実行することと、
前記 1 つまたは複数の命令を前記フェンスフラグの後に第 2 実行ユニットにおいて実行することと、
をさらに含む請求項 8 に記載の方法。

【請求項 10】

前記 1 つまたは複数の命令を前記フェンスフラグの前に実行することを、前記 1 つまたは複数の命令を前記フェンスフラグの後に実行することと実質的に同時に行う、請求項 9 に記載の方法。

30

【請求項 11】

前記命令一式を実行することは、
複数のオペランドをプログラムカウンタ実行ユニットに提供することと、
演算子を前記プログラムカウンタ実行ユニットに提供することと、
前記複数のオペランドに対して前記演算子を実行した結果に回答して更新プログラムカウンタ値を生成することと、
を含む、請求項 1 に記載の方法。

40

【請求項 12】

命令一式と開始位置とを含むパケットを受信するように構成されたパケットパーサと、
前記パケットパーサに接続され、前記命令一式を受信するように構成された命令メモリと、
前記命令メモリと前記パケットパーサに接続され、前記パケットパーサから前記開始位置を初めに受信し、前記命令メモリから前記開始位置における命令を取得するように構成されたプログラムカウンタと、

前記命令一式を実行するために、前記命令メモリに接続された複数の実行ユニットと、
前記複数の実行ユニットに接続され、ローカルメモリに対するデータ読み出しおよびデータ書き込みを制御するように構成されたパーサと、

50

前記パーサと前記パケットパーサに接続され、前記パーサと前記パケットパーサからのデータを記憶するように構成されたレジスタファイルと、

前記命令メモリと前記レジスタファイルに接続され、前記命令一式と前記レジスタファイルに記憶されたデータとを含む送信用パケットを生成するように構成されたパケットジェネレータと、

を備える装置。

【請求項 13】

前記複数の実行ユニットはそれぞれ

複数の演算論理装置（ALU）と、

前記複数の演算論理装置のうち少なくとも2つの出力の間に接続された多重化機能と、
を備える、請求項 12 に記載の装置。

10

【請求項 14】

前記複数の ALU は、前記命令一式の各命令に対応付けられた ALU を含む、請求項 13 に記載の装置。

【請求項 15】

前記複数の実行ユニットはそれぞれ if - then - else 文を実施する、請求項 13 に記載の装置。

【発明の詳細な説明】

【技術分野】

【0001】

20

優先出願

本出願は、2013年12月2日に出願された米国特許出願第14/094,273号に対する優先権の利益を主張し、その全体が本明細書に参考により組み込まれる。

【背景技術】

【0002】

一般にメモリデバイスは、コンピュータまたは他の電子デバイスにおける内部半導体集積回路として提供される。ランダムアクセスメモリ（RAM）、読出専用メモリ（ROM）、ダイナミックランダムアクセスメモリ（DRAM）、同期式ダイナミックランダムアクセスメモリ（SDRAM）、及び不揮発性メモリ（例えばフラッシュメモリ）を含む、数多くの異なる種類のメモリが存在する。

30

【0003】

いくつかの不揮発性メモリデバイスを組み合わせて、コンピュータシステムにおいて機械的に作動するハードディスクドライブをエミュレートし得るソリッドステートドライブ（SSD）を作ることが可能である。ソリッドステートドライブは、可動部品がないので、機械的ハードドライブよりもより速く、より信頼性のあるアクセスを提供可能である。

【0004】

コンピュータシステムの高性能化に少なくとも一部起因して、メモリ及びソリッドステートドライブの製造業者には、コンピュータシステムの高性能化に遅れをとらないように自社製品のメモリを高性能化するという継続的な圧力がかけられる。コンピュータシステムに対する任意の動作負荷を軽減するために、メモリに対する読出し及び書込みをより効率的にするという一般的ニーズがある。

40

【図面の簡単な説明】

【0005】

【図1】自律的メモリ処理装置の実施形態の機能的ブロック図を例示する。

【図2】図1の実施形態による、パケットパーサの実施形態のブロック図を例示する。

【図3】図1の実施形態による、プログラムカウンタの実施形態のブロック図を例示する。

【図4】図1の実施形態による、命令メモリの実施形態のブロック図を例示する。

【図5】図1の実施形態による、デコードロジックの実施形態のブロック図を例示する。

【図6】図1の実施形態による、レジスタファイルの実施形態のブロック図を例示する。

50

【図 7 A】図 7 A は、図 1 の実施形態による、実行ユニットの実施形態のブロック図を例示する。

【図 7 B】図 7 B は、図 1 の実施形態による、実行ユニットの実施形態のブロック図を例示する。

【図 8】図 1 の実施形態による、パーサの実施形態のブロック図を例示する。

【図 9】図 1 の実施形態による、パケットジェネレータの実施形態のブロック図を例示する。

【図 10】図 1 の実施形態による、命令実行フォーマットの実施形態の図解を例示する。

【図 11】メモリシステムの実施形態のブロック図を例示する。

【図 12】自律的メモリデバイスにおける自律的メモリ処理装置の作動の実施形態のフローチャートを例示する。

10

【発明を実施するための形態】

【0006】

以下の詳細説明において、本明細書の一部を形成し、例示のために特定の実施形態が示される添付図面を参照する。図面において同じ番号は、いくつかの図を通して実質的に同様の構成要素を説明する。他の実施形態が使用可能であり、そして本開示の範囲から逸脱することなく構造的、論理的、及び電気的変更を行うことが可能である。従って、以下の詳細説明は、限定的な意味で解釈されるべきではない。

【0007】

本開示は、いかなる 1 つの種類メモリにも限定されない。自律的メモリ処理装置は、任意の種類メモリデバイス、メモリデバイス群、または半導体メモリ、光学メモリ、磁気メモリを含むメモリ技術に対応付けることが可能である。例えば、メモリは、不揮発性（例えば NAND フラッシュ、NOR フラッシュ、相変化メモリ（PCM））、または揮発性（例えば DRAM、SRAM）を含み得る。

20

【0008】

本明細書で使用される場合、ノードは、受信パケットを構文解析するパケットパーサと、ノードからネットワークへ送信されるパケットを生成するパケットジェネレータと、任意のネットワークにノードをインターフェイス接続可能なネットワークポートとを備え得る。ノードはその上、ノードの作動を制御する処理要素と、並びにデータを記憶するメモリとを備え得る。別の実施形態において、ノードは、追加機能のための追加ハードウェア及び/またはソフトウェア/ファームウェアを備え得る。自律的処理装置を有する自律的メモリデバイスは、ノードとみなされ得る。

30

【0009】

図 1 は、自律的メモリ処理装置の実施形態の機能的ブロック図を例示する。このような装置は、メモリ 100 に対応付けられ、中央処理装置（CPU）ベースのコンピューティングシステムにおけるボトルネックメモリ帯域幅の渋滞を軽減するのに使用され得る。自律的メモリ処理装置は、自律的メモリデバイス内に配置され得る。

【0010】

自律的メモリ処理装置は、パケットパーサ 101 と、プログラムカウンタ 107 と、命令メモリ 105 と、デコードロジック 103 と、レジスタファイル 109 と、パーサ 115 と、パケットジェネレータ 111 と、1 つまたは複数の実行ユニット（EU）113 と、ページバッファ 117 とを備え得る。図 1 の構成要素及び構成は、他の実施形態が他の構成要素及び他の構成を使用することができるので、例示のみを目的とする。

40

【0011】

図 2 は、パケットパーサ 101 のブロック図を例示する。パケットパーサ 101 はネットワークに接続され、ネットワークから（例えば外部ネットワークからメモリ 100 へ）のデータパケットを受け入れ得る。パケットパーサ 101 がネットワークからのパケットにおいて受信したプログラムカウント（例えば命令メモリ位置）をプログラムカウンタ 107 にロードできるように、パケットパーサ 101 はプログラムカウンタ 107 の入力にも接続され得る。プログラムカウンタ 107 が現在のプログラムカウント（例えば命令メ

50

メモリ位置)をパケットパーサ101にロードできるように、パケットパーサ101はプログラムカウンタ107の出力にも接続され得る。パケットパーサ101はさらに、ネットワークからのパケットにおいて受信したデータ(例えば命令)を命令メモリ105及びレジスタファイル109へロードできるように、命令メモリ105及びレジスタファイル109の入力にも接続され得る。

【0012】

図3は、プログラムカウンタ107のブロック図を例示する。例示のため、プログラムカウンタ107は、32ビットカウンタとして示される。しかしながら、他の実施形態は、他のプログラムカウンタサイズを使用可能である。

【0013】

プログラムカウンタ107は、パケットパーサ101と、1つまたは複数の実行ユニット113から成る部であり得るプログラムカウンタ実行ユニット(PCEU)114とからの入力を有し得る。プログラムカウンタ107は、命令メモリ105に接続された出力を有し得る。

【0014】

プログラムカウンタ107は、プログラム(例えば実行可能命令)を含み得る命令メモリ105における特定の命令位置へアクセスするプログラムカウンタ値(例えば命令メモリ位置)を含み得る。プログラムカウンタ値は、パケットパーサ101により特定されたパケットパーサから受信された、受信パケット内の特定のデータフィールドにより、またはプログラムカウンタ実行ユニット114からの計算値により、設定され得る。そしてプログラムカウンタ107は、プログラムカウンタ値(例えば32ビットレジスタ)を命令メモリ105に出力し得る。

【0015】

図4は、命令メモリ105のブロック図を例示する。命令メモリ105は、プログラム(例えば実行可能命令)を記憶するいくつかのレジスタを備え得る。パケットパーサ101は、命令メモリ105の書込ポートに接続され得る。命令メモリ105はパケットパーサ101により書込可能であり、それによってパケットパーサ101により特定される受信パケット内の受信命令は、当該パケットから命令メモリ105へロードされ得る。

【0016】

命令メモリ105は、命令メモリ105内の特定位置にアクセスするためのアドレスを各自受入可能な2つのアドレスポートを備え得る。1つのアドレスは、プログラムカウンタ107から送られ得る。他方のアドレスは、パケットジェネレータ111から送られ得る。

【0017】

1つの動作中に、命令メモリ105は、プログラムカウンタ107のアドレスが示す位置から、命令(例えばデータポート)を出力し得る。行う演算に関して実行ユニット113に指示するために、この命令は実行ユニット113により解読及び実行され得る。この命令は、実行ユニット113にオペランドを与え、並びに処理のため何のデータを実行ユニット113に出力するかに関してレジスタファイル109に指示するために、レジスタファイル109にインデックスを与え得る。

【0018】

図5は、デコードロジック103のブロック図を例示する。デコードロジック103は、実行ユニットデコードロジック501と、パーサデコードロジック502と、多重分離機能503(例えばデマルチプレクサ)とを備え得る。

【0019】

多重分離機能503への入力は、命令メモリ105の出力からの命令ストリームに接続され得る。命令ストリーム内の1つまたは複数の制御ビットは、命令ストリーム内の特定命令の行先(例えばEUDコードロジック501、パーサデコードロジック502)を選択するのに使用され得る。

【0020】

10

20

30

40

50

命令がEUデコードロジック501へ送られた場合、EUデコードロジック501は、命令を実行ユニット113のうちの1つに送るため命令を処理し得る。命令は、何の種類
の演算を行うのかに関して、並びに命令の実行中に使用するオペランドを実行ユニット1
13のうちの1つに与えるように、実行ユニット113のうちの1つに指示し得る。オペ
ランドは、レジスタファイル109のレジスタに対し指標を付け、実行ユニット113の
うちの1つがデータを処理できるように何のデータを出力するかに関してそのレジスタに
指示し得る。

【0021】

多重分離機能503はまた、命令をパーサ115に接続されたパーサデコードロジック
502へ送信し得る。命令はパーサデコードロジック502を制御し、今度はパーサデ
コードロジック502が、処理のためにページバッファ117の特定セグメントからのデー
タを実行ユニット113のうちの1つへ読み込むために、ページバッファ117のどのセグ
メントにアクセスするかパーサに指示する。

10

【0022】

図6は、レジスタファイル109のブロック図を例示する。レジスタファイル109は
、パケットパーサ101、パケットジェネレータ111、実行ユニット113のうちの1
つまたは複数、及びメモリ読出指示からの入力を備え得る。メモリ読出指示は、パーサ1
15により生成される、メモリ動作が完了したことを示す信号であり得る。レジスタファ
イル109は、パケットジェネレータ111、実行ユニット113、及びパーサ115へ
の出力を備え得る。

20

【0023】

レジスタファイル109は、実行ユニット113による処理が起こっている間に変数を
記憶するメモリ（例えば複数のレジスタ）を備え得る。これらの変数は、1つまたは複数
の命令に回答してメモリから取得されたデータを含み得る。レジスタファイル109は、
レジスタに初期条件を設定するためパケットパーサ101により書込まれ、パケットジェ
ネレータ111により読出され得る。それぞれの実行ユニット113は、多重化機能を介
してレジスタファイル109から引数を受信し得る。パケットジェネレータ111への出
力は、レジスタファイル109のレジスタに記憶されているデータを、ネットワークへ送
信するためのパケットにまとめるのに使用され得る。

30

【0024】

図7Aは、一般の実行ユニット113（例えば実行ユニット(0-N)）の実施形態の
ブロック図を例示する一方、図7Bは、特にプログラムカウンタ実行ユニット114の実
施形態のブロック図を例示する。PC EU 114は、実行ユニット113群の一部とみな
され得るが、他の実行ユニット113とは異なる構成を有し得る。

【0025】

特定の自律的メモリ処理装置に、特定の数の実行ユニット113が含まれ得る必要はな
い。1つの装置は単一の実行ユニット113を有し得る一方、別の装置は複数（例えば1
00個）の実行ユニットを有し得る。

【0026】

図7Aは、実行ユニット113が4つの演算論理装置(ALU)701~704を備え
得ることを例示する。ALU1703及びALU2704の出力は、多重化機能706に
対する入力であり得る。ALU703、704の出力のうちどちらを選ぶかは、Comp
ALU702の出力により決定され得る。CompALU702の出力は、多重化機能7
06に対する選択信号として使用され得る。4番目のALUであるALU Out 701
は、実行ユニット113により行われた演算の結果を記憶する場所をレジスタファイル1
09に指示し得る、レジスタファイル109に対するレジスタアドレスR_dとしての出力
を有し得る。

40

【0027】

下の3つのALU702~704及び多重化機能706は、if-then-else
演算を行い得る。多重化機能706は、CompALU702により決定される条件の「

50

ある条件の時」を提供し得る。従って、条件が真である場合、1つのALU（例えばALU1703）の出力がCompALU702の出力により選択される、それ以外の場合、別のALU（例えばALU2704）の出力がCompALU702の出力により選択される。

【0028】

例えば、ALU1703がオペランド入力のOPERAND1 (R_1)及びOPERAND2 (R_2)とコマンド入力のOPERATOR1を有し、ALU2704がオペランド入力のOPERAND3 (R_3)及びOPERAND4 (R_4)とコマンド入力のOPERATOR2を有すると仮定する場合、if-then-else文は以下のように表され得、

if (条件)
then

Operand1 OPERATOR1 Operand2

else

Operand3 OPERATOR2 Operand4

ここで「Operand1 OPERATOR1 Operand2」はALU1703により提供され、「Operand3 OPERATOR2 Operand4」はALU2704により提供され、そして「if (条件)」はCompALU702及び多重化機能706より提供され得る。

【0029】

図10の命令フォーマットに関連して後述されるように、オペランドと演算子は、命令により提供され得るか、命令がどのレジスタにオペランド値が配置されているかを示し得る。例えば、OPERAND1 (R_1)はレジスタ R_1 に配置され、OPERAND (R_2)はレジスタ R_2 に配置され、OPERAND (R_3)はレジスタ R_3 に配置され、OPERAND (R_4)はレジスタ R_4 に配置され得る。

【0030】

ALU1703及びALU2704は、同一の演算または異なる演算を行い得る。言い換えると、OPERATOR1はOPERATOR2と同じであり得る、またはOPERATOR1はOPERATOR2とは異なり得る。

【0031】

ALUOut701は、オペランド入力 R_5 、及び R_6 （例えばレジスタ R_5 、 R_6 ）とコマンド入力OPERATOR3を有し得る。ALUOut701が R_5 と R_6 の値に対しコマンドOPERATOR3を行うことにより生成される結果 R_d は、実行ユニット113の結果がどこに記憶されるかを決定する。

【0032】

CompALU702は、オペランド入力 R_7 、 R_8 （例えばレジスタ R_7 、 R_8 ）とコマンド入力OPERATOR4を有し得る。前述のように、 R_7 と R_8 の値に対しコマンドOPERATOR4を行うことにより得られる結果が、多重化機能106の選択を決定する。

【0033】

前述のALU701~704におけるコマンド（例えばOPERATOR1、OPERATOR2、OPERATOR3、OPERATOR4）として使用され得る一般演算には、加算、減算、論理AND、論理OR、論理NOT、論理NOR、~と等しい、~以下、~未満、~以外、~以上、または~より大きいが含まれ得る。これらの演算は、他の実施形態が他の演算を使用可能であるので、例示のみを目的とする。

【0034】

図7Bは、プログラムカウンタ実行ユニット(PCEU)114の構成を例示する。この構成は、実行ユニット0~n113と同様であり得るが、ALUOut701を有さない。PCEU114はプログラムカウンタ107に対する新たなアドレスを特定することを専門とし、PCEU114の演算の結果を記憶する場所はプログラムカウンタ107

10

20

30

40

50

であるため、ALU Out 701は含まれない。

【0035】

PCEU 114は、オペランド入力R9、R10とコマンド入力OPERATOR5を有するCompALU710を備え得る。ALU1711は、オペランド入力R11、R12とコマンド入力OPERATOR6を含み得る。ALU2712は、オペランド入力R13、R14とコマンド入力OPERATOR7を含み得る。

【0036】

ALU1711及びALU2712の出力は、多重化機能714に対する入力であり得る。CompALU710の出力は、多重化機能714に対する選択信号を提供し得る。従って、前述の実行ユニット113の場合と同様に、PCEU114は、CompALU710により決定される条件の「ある条件の時」を多重化機能714が提供する、if-then-else文を提供し得る。従って、条件が真である場合、1つのALU（例えばALU1711）の出力がCompALU710の出力により選択される、それ以外の場合、別のALU（例えばALU2712）の出力がCompALU710の出力により選択される。結果は、プログラムカウンタ107へロードされ得る。

【0037】

前述の実行ユニット113の場合と同様に、PCEU114において使用される演算子及びコマンドは、命令メモリの命令からロードされ得るか、命令がどのレジスタが値を含み得るかを示し得る。

【0038】

図8は、パーサ115のブロック図を例示する。パーサ115は、書込むアドレス並びにデータを含むメモリ書込ポートを備え得る。メモリ読出アドレスポートは、読出すアドレスをメモリに提供し、それによって読出されたデータは、メモリ読出データポートに読込まれ得る。パーサ115はまた、メモリ読出動作が完了すると、メモリ読出完了指示信号を出力し得る。パーサ115はさらに、実行ユニット113への出力と、レジスタファイル109からの入力と、及びパーサデコードロジック502からの設定入力とを備え得る。

【0039】

パーサ115は、メモリ100のページバッファ117からの直接読出し、またはページバッファ117への直接書込みができるように、メモリ100への直接アクセスを有し得る。パーサ115は、ページバッファ117の全長に対しアクセス権を有するので、処理をより管理しやすいように、ページバッファ117をより小さいセグメント（例えば規則正しく定義されたセグメント）に細分化し得る。例えば、パーサ115は、ページバッファの最初の100バイトに対し作動し、それから次の100バイトに対し作動し、そしてページバッファ117全体の読出し/書込みが完了するまで、これを続け得る。これを達成するために、ページバッファ117のどのセグメントを読出すかを特定する、パケットパーサ101からのアドレスがパーサ115に与えられ得る。

【0040】

パーサ115は、ページバッファ117の内容をどのように構文解析するかパーサ115に指示し得る設定入力をレジスタファイル109から受信し得る。パーサ115は、新たな内容がレジスタファイル109において利用可能であることを実行プログラムに指示するメモリ読出完了指示信号を生成し得る。

【0041】

図9は、パケットジェネレータ111の実施形態のブロック図を例示する。パケットジェネレータは、命令メモリ105及びレジスタファイル109からの入力と、命令メモリ105及びレジスタファイル109への出力とを備え得る。パケットジェネレータ111はその上、任意の生成パケットを出力するために、ネットワークへの出力を有する。

【0042】

パケットジェネレータ111は、命令メモリ105及びレジスタファイル109からデータを読出すために、命令メモリ105に対するアドレスと、レジスタファイル109に

10

20

30

40

50

に対するアドレスを生成し得る。そしてパケットジェネレータ111は、レジスタファイル109から読出されたデータ（例えば命令メモリ105からの命令及びコンテキスト（例えばデータ、メモリ読出しの結果、行われた演算の結果））を使用して、このデータをまとめ、ネットワーク上に送信するパケットを生成する。

【0043】

図10は、図1の実施形態による、命令実行フォーマットの実施形態を例示する。各命令1001～1003は、実行ユニット113により実行されるために、命令メモリに記憶され得る。

【0044】

命令の例示された実施形態には、4つの命令1000～1003が含まれる。各命令は、実行ユニット113の異なるALUに対応付けられ得る。従って、実行ユニット113が異なる数のALUを含む場合、実行フォーマットは異なる数の命令1000～1003を含み得る。以下の論述においては、図10と図7AのALUの両方を参照する。

【0045】

第1命令1000（例えば命令D）は、実行ユニット113のうちの1つによる演算の結果の行先レジスタ（例えば R_d ）を示し得る。前述のように、ALU Out 701は、実行ユニット113の結果を記憶するための行先レジスタ R_d のアドレスを生成し得る。従って、ALU Out 701は、レジスタ R_d を生成するために、第1命令1000に対応付けられ得る。

【0046】

第2命令1001（例えば命令C）は、実行ユニット113により実行されるif-then-else文の条件を示し得る。例示された実施形態において、条件は比較値 V_c により表される。前述のように、Comp ALU 702は、多重化機能706に対する選択信号として使用される条件を生成し得る。従って、Comp ALU 702は、 V_c が真か否かを比較するために、第2命令1001に対応付けられ得る。

【0047】

第3命令1002（例えば命令T）は、実行ユニット113により実行されるif-then-else文の「then」（真の場合）の結果を示し得る。例示された実施形態において、「then」の結果は V_t 、すなわち真の場合の値により表される。前述のように、ALU 1703は、「then」の結果を生成し得る。従って、ALU 1703は、 V_t である「then」の結果のための第3命令1002に対応付けられ得る。

【0048】

第4命令1003（例えば命令F）は、実行ユニット113により実行されるif-then-else文の「else」（偽の場合）の結果を示し得る。例示された実施形態において、elseの結果は V_f 、すなわち偽の場合の値により表される。前述のように、ALU 2704は、「else」の結果を生成し得る。従って、ALU 2704は、「else」の結果 V_f のため、第4命令1003に対応付けられ得る。

【0049】

条件の V_c 、thenの結果の V_t 、elseの結果の V_f 、及び結果レジスタの R_d を使用すると、if-then-else文は以下のように表され得る。

```
if (  $V_c$  == TRUE )
then
```

$$\text{Reg}[R_d] := V_t$$

```
else
```

$$\text{Reg}[R_d] := V_f$$

【0050】

図11は、図1の自律的メモリ処理装置130を組み込み得るメモリシステムの実施形態のブロック図を例示する。メモリシステムは、ネットワーク1120を介して1つまたは複数のメモリデバイス（例えばSSD）1101、1102と通信し得るコントローラ1100（例えばCPU）を備え得る。ネットワーク1120は、有線バス、または無線

10

20

30

40

50

通信（例えばW i F i ）であり得る。

【 0 0 5 1 】

メモリデバイス1101は、メモリデバイス1101のストレージ部を構成するローカルメモリ100（例えばRAM、DRAM、SRAM、NANDフラッシュ、NORフラッシュ、相変化メモリ（PCM））と、並びに図1の自律的メモリ処理装置130とを備え得る。自律的メモリ処理装置130は、メモリ100に相対的に近い所（例えば同じダイ、同じダイスタック、同じメモリモジュール）に配置され得る。例えば、自律的メモリ処理装置130は、メモリ100のバンクレベルの回路に含まれ得る。各バンクが異なる自律的メモリ処理装置130を含み得、そのために1つのメモリチップは、ほぼ同時に作動する自律的メモリ処理装置130の複数のインスタンスを有し得る。本明細書において使用される場合、ローカルメモリ100は、ネットワークを介さずに自律的メモリ処理装置130に接続されるメモリであり得る。

10

【 0 0 5 2 】

図11のシステムのそれぞれのデバイスは、ノードとみなされ得る。各ノードは他のノードとネットワーク1120を介して通信し得る。それぞれのノードは実質的に同様であり得る、またはノードのうちの1つまたは複数の異なる構成を有し得る。例えば、第1メモリデバイス1101は、プログラムカウンタ実行ユニット114に加えて単一実行ユニット113のみを有し得る一方、第2メモリデバイス1102は、プログラムカウンタ実行ユニット114に加えて複数の実行ユニット113を有し得る。

【 0 0 5 3 】

従って、後述のように、コントローラ1100（例えばソースノード）は、命令及びソースノードの現在の処理状態を含むメッセージ（例えばパケット）を、メモリデバイス1101（例えば目標ノード）へ送信し得る。別の実施形態において、第1メモリデバイス1101がソースノードであり得る一方、第2メモリデバイス1102が目標ノードであり得る。

20

【 0 0 5 4 】

命令は、メモリデバイス1101に対するコマンド（例えば、検索、ソート、比較）を含み得る。メモリデバイス1101は、コントローラの介在なしに、コマンドにより指示されたタスクを行い得る。自律的メモリ処理装置130は、他のノード1100、1102へメッセージの送信、他のノード1100、1102からメッセージの受信、他のノード1100、1102へ処理命令及び処理状態の送信、他のノード1100、1102から処理命令及び処理状態の受信、プログラム状態の復元及び保存、処理命令の実行、ローカルメモリの読み出し及び書き込み、及び/または単一ノードにおける多重処理コンテキストの支援を行い得る。

30

【 0 0 5 5 】

自律的メモリ処理装置130の構成により、動的で滞りがなく柔軟性のある実行ユニット113（例えばALUを含む）の追加及び削除が可能となり、従ってノードは必要に応じて追加の処理能力が与えられる。自律的メモリ処理装置130における実行ユニット113の動的な追加及び削除は、以下の作動実施例において例示され得る。

【 0 0 5 6 】

標準的な先行技術のプログラムは、以下のように生成され得る。

命令1（ADDレジスタ1、レジスタ2、レジスタ3）

命令2（SUBレジスタ2、レジスタ3、レジスタ4）

【 0 0 5 7 】

標準的な先行技術のCPUシステムの場合のように、これらの命令には暗黙の従属性が含まれる。例えば、命令1が実行機会を有する前に、レジスタ2の値が書換えられ得るので、命令2は命令1の前に（または命令1と同じサイクルにおいて）実行不可能であり得る。

40

【 0 0 5 8 】

プログラムを実行するのに必要なサイクル数を削減するために、自律的メモリ処理装置

50

構成において、より複雑な実行ユニット（E U）構成が使用され得る。各E Uは、それぞれが別個のタスクを行ういくつかの異なるA L U（例えば4つのA L U）を含み得る。従って、自律的メモリ処理装置のために書かれたプログラムは、以下のように生成され得る（1つのE Uに加えP C E Uを有する構成と仮定した場合）。

[P C E U命令1] [E U 1命令1]
[P C E U命令2] [E U 1命令2]
【 0 0 5 9 】

各 [E U # 命令 #] は、図 1 0 に例示されるように、以下のように現れる。

[行先命令] [比較命令] [I f - t r u e 命令] [I f - f a l s e 命令]
【 0 0 6 0 】

10

また、自律的メモリ処理装置の構成の一部として、プロセッサはその内部に、異なる数のE Uを組み込み得る。これにより4つのE Uと1つのP C E Uを有する構成は、例えば以下のことが可能になり得る。

[P C E U命令1] [E U 1命令1] [E U 2命令1] [E U 3命令1] [E U 4命令1]
]
[P C E U命令2] [E U 1命令2] [E U 2命令2] [E U 3命令2] [E U 4命令2]
]
【 0 0 6 1 】

当サイクルにおいて実行する追加作業は存在し得ないという事実に基づいて、これらのE U命令のうちいずれかが1つは空であり得る。これは、プログラムの特定の段階において、並列処理がないことに起因し得る。

20

【 0 0 6 2 】

自律的メモリ処理装置の構成により、システムにおける異種の自律的メモリ処理装置エンジン（例えば1つの装置「A」は1つのE Uに加えP C E Uを有し得る一方、同じ相互接続システムにおける別の装置「B」は4つのE Uに加えP C E Uを有し得る）一式内の対話が可能になり得る。当シナリオにおいて、装置Aが装置「B」に対しコンテキストを送る必要があると仮定した場合、プログラムは命令の連続ストリーム内に装入され、装置「B」へ送り出され得る。装置「B」は、そこで、それ自体のハードウェア上にそれら命令を以下同様にスケジュールし得る。

[P C E U命令1] [E U 1命令1] [空] [空] [空]
[P C E U命令2] [E U 1命令2] [空] [空] [空]

30

...

【 0 0 6 3 】

全てのプログラムは最終的に、最も精密な自律的メモリ処理装置の並列処理に臨み得るため、これはシステムにおいて非効率性を生じる無駄な並列処理へと導き得る。

【 0 0 6 4 】

命令間に従属性が何もないことを確かめずに、命令を並列E Uへまとめて渡すことはあり得ない。標準的な先行技術のシステムにおいて、このような種類の比較は計算上コストが高くあり得るため、自律的メモリ処理装置は、「フェンス」フラグという命令の概念を使用し得る。「フェンス」フラグにより、アプリケーションの作者またはアプリケーションの作成者は、命令ストリームがそのストリームにおける前の命令に対する従属性をもう有さないということを示すことが可能になる。当情報により、命令ストリームは、重要処理のオーバーヘッドなしで、異種プロセッサ一式上で回され、スケジュールされることが可能になり得る。

40

【 0 0 6 5 】

例えば、以下の命令ストリーム [P C E U命令] [E U命令1] [E U命令2] [E U命令3] [フェンスマーカ / 命令] [E U命令4] [E U命令5] [E U命令6] [E U命令7] [フェンスフラグ / 命令] は、自律的メモリ処理装置「A」上において [P C E U] [1] [P C E U] [2] [F] [P C E U] [3] [P C E U] [4] [P C E U] [5] [P C E U] [6] [F] [P C E U] [7] のようにスケジュールされ ([F

50

]は「フェンス」マーカを示す)、そして自律的メモリ処理装置「B」において[PCEU][1][2][3][X][F][PCEU][4][5][6][7]としてスケジュールされ得る。

【0066】

「フェンス」命令は、所定の自律的メモリ処理装置(例えば「A」または「B」)の命令メモリへロードされている間に、パケット内論理により処理され得る。「フェンス」フラグの存在は、命令メモリに記憶され得るが、スケジューリングコンテキスト外では意味がない場合がある。しかしながら、これは命令メモリにおいてフラグとして記憶され、そのためパケット送信論理により元のストリームが再構成され得る。

【0067】

自律的メモリ処理装置の作動実施例として(例えばメモリ検索)、パケットは、ネットワーク(例えばメモリネットワーク)からパケットパーサ101により受信され得る。パケットパーサ101は、パケットをセグメントに分解し得る。いくつかのセグメントは、パケットが前のノードから送られた時の前のノードの状態を表すレジスタ内容を含み得るため、コンテキストであり得る。

【0068】

パケットは、プログラムが実行される、命令メモリ105における開始位置を含み得る。この開始点は、プログラムカウンタ107へロードされ得る。パケットはまた、命令メモリ105へロードされる命令一式と、レジスタファイル109へロードされ得る初期条件一式とを含み得る。初期条件は、前のノードからの命令により送られた変数であり得る。初期条件はまた、現行の実行プログラムにより使用される定数でもあり得る。

【0069】

プログラムカウンタ107の値は、どの命令を実行のために命令メモリ105から読出すかを決定する。プログラムカウンタ107の次の値は、前の値より増加した値であり得る、またはプログラムカウンタ実行ユニット114により決定される計算値であり得る。

【0070】

命令は、パーサ115の構成を設定し得る。パーサ115は、命令の実行を通して、変数をページバッファ117から抽出し、最終的にメモリ読出動作を行うように構成され得る。

【0071】

メモリ読出動作が起こると、変数はページバッファ117の内容からリアルタイムに取り出され、実行ユニット113に対し入力として提供され得る。プログラム命令により決定されるように、別の可能入力がレジスタファイルから読出され、処理のため実行ユニット113へ提供され得る。前述のように、「フェンス」により、いくつかの連続命令を並行して実行することが可能になり得る。並列実行不可能な命令は、延期され、次のサイクル中に実行され得る。

【0072】

実行ユニット113は、これらの入力引数を、各セットが並列処理される複数の入力引数セットとして処理し得る。従って、複数の実行ユニット113が、レジスタファイルへ戻され得る、あるいは1つまたは複数のメモリ書込動作のためのデータとして最終的にページバッファ117へ書込まれるようにパーサ115へ移動され得る、出力変数を生成し得る。すなわち出力変数は、ある特定の活動を生み出すためにレジスタファイルへ入り得る。動作は、パケットジェネレータ111によりパケットを生成する、あるいは新たなメモリ読出動作またはメモリ書込動作を開始するためのものであり得る。

【0073】

ページバッファ117の内容(例えば検索コマンドの結果)は、ネットワークを介して要求ノードへ送信されるパケットに含まれるように、パケットジェネレータ111に提供され得る。パケットには、タスク(例えば検索)が完了したこと、及びその結果がパケットに含まれていることを示す要求ノードに対するメッセージが含まれ得る。

【0074】

10

20

30

40

50

作動に関するより広範な実施例として、ネットワークは、それぞれが少なくとも1つの自律的メモリ処理装置を備える複数の自律的メモリデバイスから成る機構を含み得る。データ群はメモリデバイス機構の至る所に記憶され得る。特定のデータリストについてデータ群全体の検索が望まれる時、検索プログラムが1つの自律的メモリデバイスにプッシュされ、当該デバイスにおいて特定のデータリストが検索され得る。その特定の自律的メモリデバイスに記憶されているデータが検索され、リストの全てのデータは存在しないことがプログラムにより判定されると、プログラムは1つまたは複数のパケットにまとめられ、別の自律的メモリデバイスに転送され、この別のデバイスの自律的メモリ処理装置が検索を継続し得る。このプログラムをまとめる動作は、自律的メモリデバイスの機構全体が検索されるまで、またはデータリストが完了するまで、続けられ得る。いくつかの実施形態において、特定の自律的メモリデバイスにおいて見つかったデータもまた、転送するプログラムと一緒にパケット（複数可）にまとめられ得る。

10

【0075】

そのような実施形態が、図12のフローチャートにおいて例示される。例示された方法は、図11のシステムにおいて、自律的メモリデバイス1101内の自律的メモリ処理装置130により実行され得る。

【0076】

メモリデバイス1101は、自律的メモリ処理装置130に提供されるパケットを受信し得る1201。装置130は、前述のように、命令、プログラムカウンタ、データを取り出すために、パケットを構文解析し得る1203。そしてメモリ100に記憶されているデータに対し所望のタスクを行うために、命令が実行され得る1205。そして命令と命令により生成された任意のデータが、ネットワーク上で送信するために1209、パケットにまとめられ得る1207。

20

【0077】

装置は回路、集積回路ダイ、メモリデバイス、メモリアレイ、またはシステムとして定義され得る。

【0078】**結論**

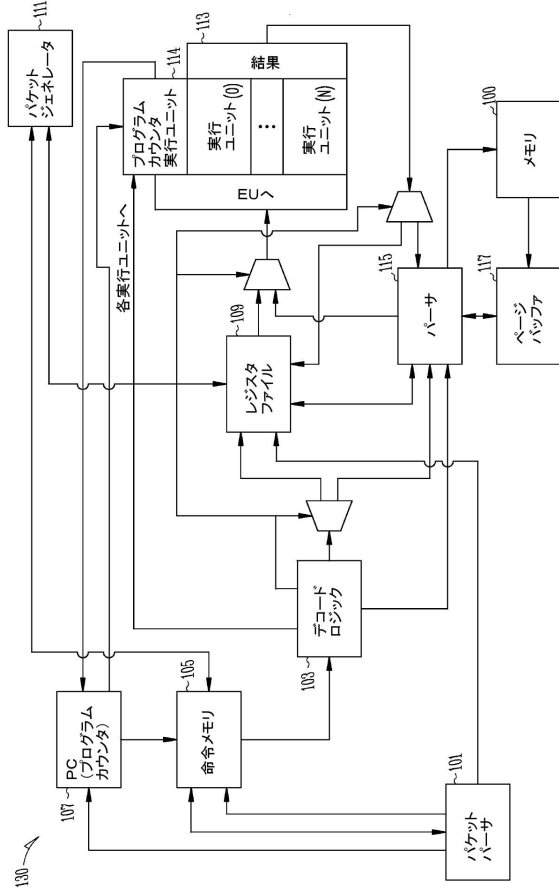
自律的メモリデバイス内の自律的メモリ処理装置の1つまたは複数の実施形態は、従来のCPUベースのコンピューティングシステムにおけるメモリ帯域幅のボトルネックを軽減するために、命令の処理を行い得る。命令（例えばプログラム）一式及び/またはデータを含むパケットはノード間で転送され得、そのためにこれらのノード内のメモリに記憶されたデータは、ソースノードまたはCPUによる制御とは独立した命令により処理され得る。

30

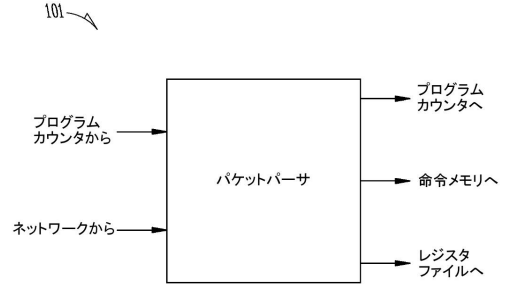
【0079】

特定の実施形態が本明細書において例示され説明されたが、同じ目的を達成する意図の任意の配置が、示された特定の実施形態と置換えられ得ることを、当業者は理解するであろう。当業者にとって、数多くの適応形態が明らかであろう。従って、本出願は、任意の適応形態または変形を含めることが意図される。

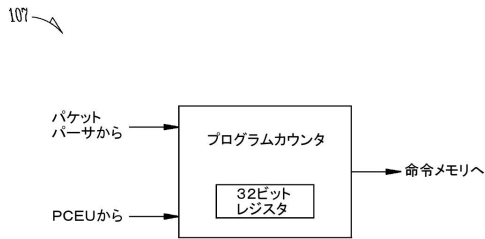
【 図 1 】



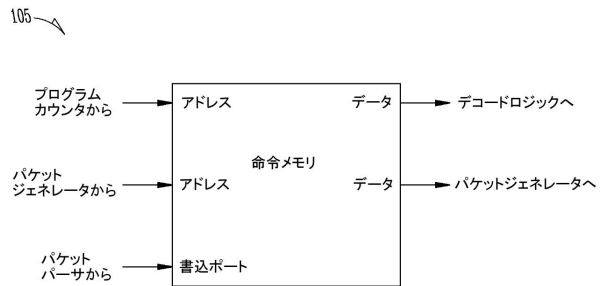
【 図 2 】



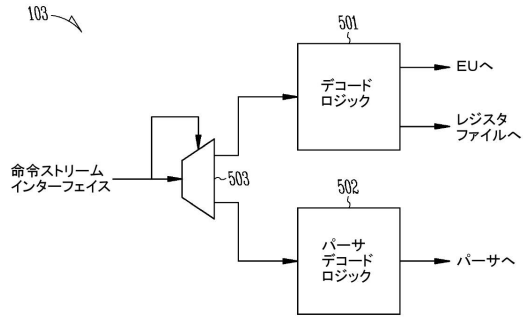
【 図 3 】



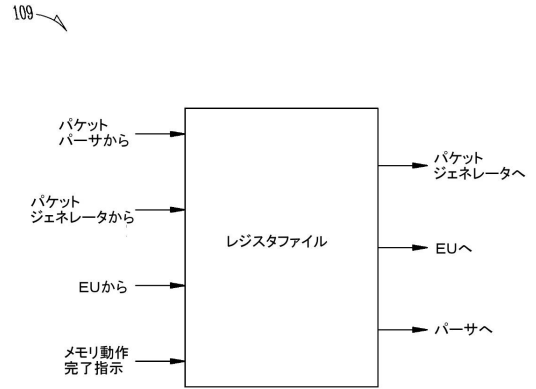
【 図 4 】



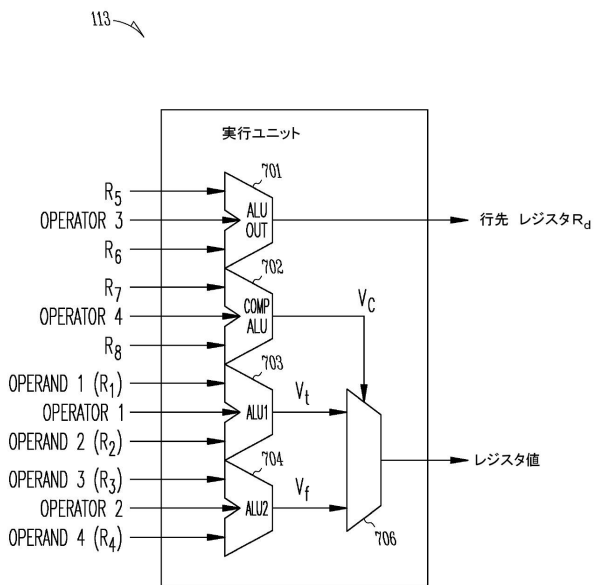
【図5】



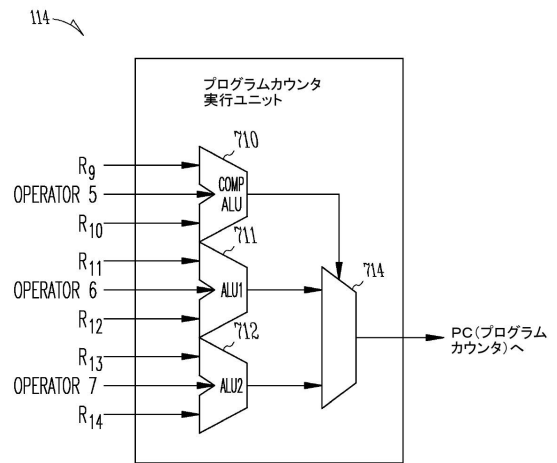
【図6】



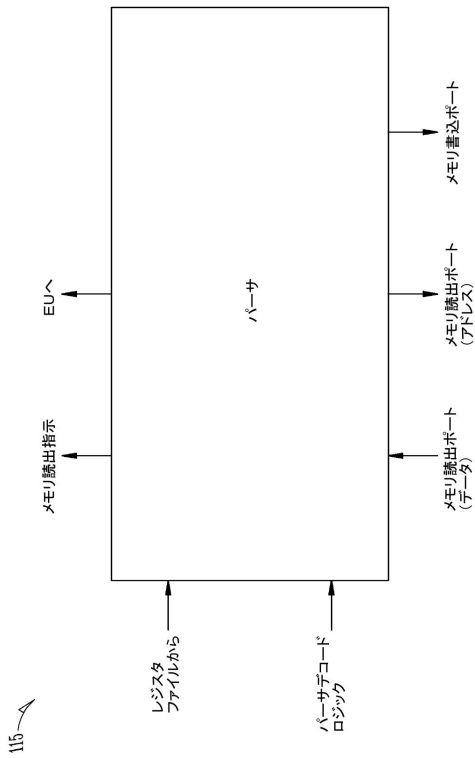
【図7A】



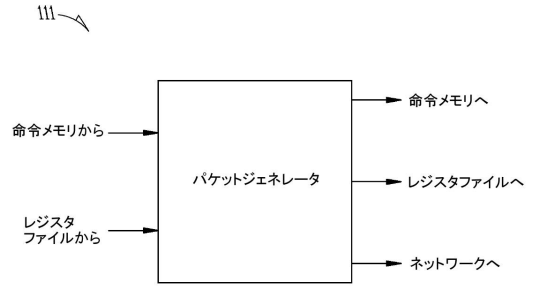
【図7B】



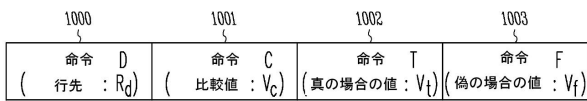
【 図 8 】



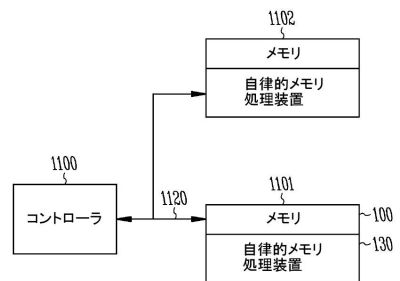
【 図 9 】



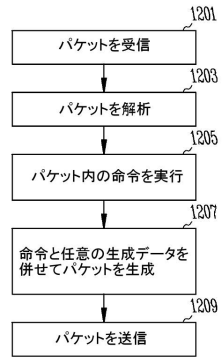
【 図 10 】



【 図 11 】



【図 12】



フロントページの続き

- (72)発明者 エイラート, ショーン エス.
アメリカ合衆国, カリフォルニア州 9 5 6 6 3 , ペンリン, イングリッシュ コロニー ウェイ
1 7 2 1
- (72)発明者 アケル, アミン ディー.
アメリカ合衆国, カリフォルニア州 9 5 7 4 2 , ランチョ コルドヴァ, ボーダーランズ ドラ
イブ 4 2 1 4
- (72)発明者 ワン, ホンギョ
アメリカ合衆国, カリフォルニア州 9 5 6 3 0 , フォルソム, メレディス ウェイ 1 2 2 5

合議体

審判長 辻本 泰隆

審判官 山崎 慎一

審判官 須田 勝巳

- (56)参考文献 米国特許出願公開第2012/0102275 (US, A1)
特開2011-60279 (JP, A)
米国特許第8427952 (US, B1)
特表2007-529833 (JP, A)
米国特許出願公開第2010/0161914 (US, A1)

(58)調査した分野(Int.Cl., DB名)

G06F12/00

G06F12/06

G06F9/34

G11C11/401