(19)日本国特許庁(JP)　(12)公表特許公報(A)　(11)特許出願公表番号
特表2013-532880
(P2013-532880A)
(43)公表日　平成25年8月19日(2013.8.19)

(51)Int.Cl.		F I			テーマコード（参考）
G06F 12/00	(2006.01)	G06F	12/00	550A	5B005
G06F 12/08	(2006.01)	G06F	12/08	531B	5B060

審査請求 未請求　予備審査請求 未請求　（全 24 頁）

(21)出願番号	特願2013-523301 (P2013-523301)
(86)(22)出願日	平成23年8月3日 (2011.8.3)
(85)翻訳文提出日	平成25年4月1日 (2013.4.1)
(86)国際出願番号	PCT/US2011/046412
(87)国際公開番号	WO2012/018906
(87)国際公開日	平成24年2月9日 (2012.2.9)
(31)優先権主張番号	12/849,724
(32)優先日	平成22年8月3日 (2010.8.3)
(33)優先権主張国	米国 (US)

(71)出願人　591016172
アドバンスト・マイクロ・ディバイシズ・インコーポレイテッド
ＡＤＶＡＮＣＥＤ　ＭＩＣＲＯ　ＤＥＶＩＣＥＳ　ＩＮＣＯＲＰＯＲＡＴＥＤ
アメリカ合衆国、９４０８８－３４５３　カリフォルニア州、サニィベイル、ピィ・オゥ・ボックス・３４５３、ワン・エイ・エム・ディ・プレイス、メイル・ストップ・６８（番地なし）
(74)代理人　100108833
弁理士　早川　裕司
(74)代理人　100111615
弁理士　佐野　良太

最終頁に続く

(54)【発明の名称】メモリ領域を埋めるためのプロセッサ支援

(57)【要約】
プロセッサおよび／または処理要素の間の処理負荷を分散させることに関する技術が開示される。少なくとも第１処理要素および第２処理要素を有するコンピュータシステムは、１以上のメモリ領域を初期化する要求を前記第２処理要素に処理させることができる。初期化は、前記指定された初期化されるべきメモリ領域を有するメモリへ前記第２処理要素を直接アクセスさせることにより実現されることができる。これにより、前記第２処理要素が前記初期化されるべきメモリ領域を処理する一方、前記第１処理要素は解放されて他の計算タスクを行うことができる。前記第１処理要素に関連付けられたキャッシュは、前記第２処理要素が前記初期化を行う結果、初期化により妨げられることがなく、また、データが前記キャッシュから移動されるのを回避することができる。
【選択図】図４

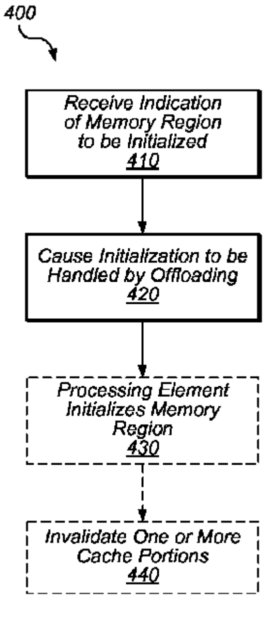


FIG. 4

【特許請求の範囲】
【請求項１】
　コンピュータデバイスにおける初期化されるメモリ領域を示す指示に応答して、該メモリ領域の初期化を該コンピュータデバイスの第２処理要素に処理させるための、少なくとも該コンピュータデバイスの第１処理要素によって実行可能なプログラム命令を記憶する、
　非一時的なコンピュータ読み取り可能な媒体。
【請求項２】
　前記メモリ領域の指示が、第１プログラムから、前記第１処理要素によって実行される制御プログラムにより受付けられる、
　請求項１の非一時的なコンピュータ読み取り可能な媒体。
【請求項３】
　前記制御プログラムが、前記第１プログラムを実行している、
　請求項２の非一時的なコンピュータ読み取り可能な媒体。
【請求項４】
　前記指示は、前記制御プログラムで処理可能な１以上のデータオブジェクトに対応したメモリの１以上のメモリ領域を指定しており、
　前記１以上のメモリ領域のすべての内容を埋める処理がさらに含まれている、
　請求項２の非一時的なコンピュータ読み取り可能な媒体。
【請求項５】
　前記制御プログラムは、ガーベジコレクションプロセスの一部として、初期化される複数のメモリ領域を示す複数の指示を生成する処理と、前記複数のメモリ領域に対する初期化を前記第２処理要素に行わせる処理とをさらに含み、
　前記初期化には、前記複数のメモリ領域のすべての内容を、プログラミング言語の言語仕様にしたがって指定されたデフォルトの内容で埋める処理が含まれる、
　請求項４の非一時的なコンピュータ読み取り可能な媒体。
【請求項６】
　前記制御プログラムは、前記非一時的なコンピュータ読み取り可能な媒体に記憶された１以上のライブラリファイルを有し、かつ、前記指示を受付ける前記制御プログラムは、アプリケーション・プログラミング・インタフェース（ＡＰＩ）を介して前記指示を受付ける前記制御プログラムを有する、
　請求項２の非一時的なコンピュータ読み取り可能な媒体。
【請求項７】
　前記初期化処理には、前記メモリ領域の内容を変更するために前記第２処理要素によって実行可能な１以上の命令からなる命令群の少なくとも一部を動的に生成する処理が含まれている、
　請求項１の非一時的なコンピュータ読み取り可能な媒体。
【請求項８】
　前記メモリの初期化処理を前記第２処理要素に行わせる処理には、前記コンピュータシステムのキャッシュに前記初期化されたメモリ領域の初期化後の内容を記憶させることが含まれず、かつ、
　前記キャッシュは、前記第１処理要素が、前記メモリ領域を有する前記コンピュータシステムのメモリにアクセスするのに応答して、前記メモリ領域の内容を記憶するように構成されている、
　請求項１の非一時的なコンピュータ読み取り可能な媒体。
【請求項９】
　前記第１処理要素および前記第２処理要素の少なくとも一方を生成するために実行可能なプログラム命令をさらに有する、
　請求項１の非一時的なコンピュータ読み取り可能な媒体。
【請求項１０】

第１処理要素と、第２処理要素と、メモリ領域を含むメモリとを備えたコンピュータデバイスにおける初期化される該メモリ領域を示す指示に応答して、前記第１処理要素で実行されている第１プログラムが、該メモリ領域の初期化を前記第２処理要素に処理させる、
方法。
【請求項１１】
前記第１処理要素が前記メモリ領域へ直接アクセスすることなく、前記第２処理要素が、ダイレクトメモリアクセス（ＤＭＡ）を用いて前記メモリ領域を初期化する処理、をさらに含む、
請求項１０の方法。
【請求項１２】
前記指示を生成する前記第１プログラム内においてガーベジコレクションプロセスをさらに含む、
請求項１０の方法。
【請求項１３】
前記第１プログラムは制御プログラムであって、
前記第２処理要素が、前記制御プログラムの１以上のヒューリスティックな規則にしたがって前記メモリ領域を初期化する処理、をさらに含む、
請求項１０の方法。
【請求項１４】
前記コンピュータデバイスが、前記メモリ領域が初期化されるのに応答して、前記コンピュータデバイスのデータキャッシュの１以上の部分であってメモリ領域初期化前の前記メモリ領域の内容に対応する該部分を無効化する処理、をさらに含む、
請求項１０の方法。
【請求項１５】
メインメモリを有するメモリサブシステムと、２次記憶装置と、少なくとも第１および第２処理要素と、を有するコンピュータシステムであって、
前記２次記憶装置には、初期化される前記メインメモリのメモリ領域を示す指示に応答して、該メモリ領域の初期化を前記第２処理要素に処理させるように前記コンピュータシステムを動作させるための、前記第１処理要素によって実行可能なプログラム命令、が記憶されている、
コンピュータシステム。
【請求項１６】
前記第１処理要素と、前記第２処理要素とは異機種である、
請求項１５のコンピュータシステム。
【請求項１７】
前記第１処理要素と関連付けられたキャッシュをさらに有し、
前記キャッシュは、前記第１処理要素による前記メインメモリへのアクセスに応答して該メインメモリの内容を記憶するように構成され、
前記メモリ領域の初期化が行われたとき、前記メモリ領域の初期化後の内容が前記キャッシュに記憶されない、
請求項１５のコンピュータシステム。
【請求項１８】
前記第２処理要素に対して前記メインメモリへの直接アクセスを許可するように構成されたメモリアクセスコントローラをさらに有し、
メモリ領域の初期化には、前記第２処理要素が前記メモリアクセスコントローラを用いて前記メモリ領域にアクセスすることが含まれ、前記第１処理要素が前記メモリ領域にアクセスすることが含まれない、
請求項１５のコンピュータシステム。
【請求項１９】

処理要素であって、
メモリデバイスのメモリ領域の初期化を第２処理要素に処理させるように構成されたメモリ初期化回路を有し、
前記メモリ領域を初期化する指示に応答して、前記初期化を第２処理要素に処理させる処理を行う、
処理要素。
【請求項２０】
コンピュータシステムで実行可能なプログラムによって利用可能なデータ構造であって、該データ構造によって記述された回路を有する集積回路を製造するプロセスの一部を行うためのデータ構造、を有する非一時的なコンピュータ読取り可能な記憶媒体であって、
メモリデバイスのメモリ領域の初期化を、コンピュータデバイスの第１処理要素ではなく該コンピュータデバイスの第２処理要素に行わせるように構成されたメモリ初期化ユニットを有し、
前記初期化を前記第２処理要素に行わせる処理が、前記メモリ領域を初期化する指示に応答して行われる、
非一時的なコンピュータ読取り可能な記憶媒体。
【請求項２１】
ＨＤＬデータ、ＶｅｒｉｌｏｇデータおよびＧＤＳＩＩデータの少なくともいずれかを記憶している、
請求項２０の非一時的なコンピュータ読取り可能な記憶媒体。
【発明の詳細な説明】
【技術分野】
【０００１】
本開示は、コンピュータプロセッサに関し、より具体的には、メモリ領域を埋める要求を受付けるプロセッサに関する。
【背景技術】
【０００２】
コンピュータの動作の間、メモリの領域がある値を用いて初期化される（埋められる）必要が生じることがある。メモリ領域の初期化には、コンピュータ上のあるリソースが用いられる。例えば、初期化を行うプロセッサは、一連のメモリロケーションに値を書き込まなければならないことがあり、これには時間がかかり得る。このような初期化の間、プロセッサは他の演算タスクを行うことができないことがある。
【発明の概要】
【発明が解決しようとする課題】
【０００３】
さらに、メモリの初期化処理は、プロセッサと関連付けられたキャッシュに混乱を生じさせることがある。キャッシュ性能は、キャッシュの内容がメモリの初期化の間にキャッシュから移動されるので、プロセッサによって負の影響を受けることがある。例えば、キャッシュの（メモリ領域の初期化が開始される以前からの）既存の内容の幾つかまたはすべてが、初期化されるメモリ領域の内容によって置き換えられることもありうる。このような置換では、それまでキャッシュ内に存在していたデータを取得するために他のメモリが続けてアクセスされることがあるため、プログラムの実行が遅くなることがある。
【課題を解決するための手段】
【０００４】
コンピュータシステムまたは計算装置が、あるメモリ処理を第１処理要素から第２処理要素へ分散することを可能とする方法および構造の様々な実施形態がここに開示される。
【０００５】
記載された一実施形態において、コンピュータデバイスの初期化されるメモリ領域を示す指示を受付けることと、この受付に対応してメモリ領域の初期化をコンピュータデバイスの第２処理要素に処理させることとを含む処理を、コンピュータデバイスの少なくとも

第１処理要素によって実行可能なプログラム命令を有する、コンピュータ読み取り可能な媒体が開示される（ここで、プログラム命令はコンピュータ読み取り可能な媒体に記憶されている）。さらなる実施形態においては、当該指示は、第１処理要素によって実行される制御プログラムから受付ける。

【０００６】

別の実施形態は、コンピュータデバイスの初期化されるメモリ領域を示す指示を受付ける第１プログラムを含み、当該第１プログラムをコンピュータデバイスの第１処理要素で実行し、第１プログラムが、当該受付に対応して、メモリ領域の初期化をコンピュータデバイスの第２処理要素に処理させる方法を含む。さらなる実施形態においては、第２処理要素は、第１処理要素によるメモリ領域への直接アクセスを伴わずにメモリ領域を初期化するために、ダイレクトメモリアクセス（ＤＭＡ）を利用する。

【０００７】

さらに別の実施形態は、メインメモリを有するメモリサブシステムと、２次記憶装置と、少なくとも第１および第２処理要素と、を有するコンピュータシステムであって、前記２次記憶装置には、初期化されるメモリ領域を示す指示をコンピュータシステムに受付けさせるための、第１処理要素によって実行可能なプログラム命令が記憶され、メモリ領域は前記メインメモリ内に存在し、かつ、前記受付けに応答して、メモリ領域の初期化をコンピュータデバイスの前記第２処理要素に処理させるコンピュータシステムである。さらなる実施形態においては、コンピュータシステムは、前記第１処理要素と関連付けられたキャッシュを有し、前記キャッシュは、前記第１処理要素による前記メインメモリへのアクセスに応答して前記メインメモリの内容を記憶するように構成され、かつ、前記メモリ領域の初期化を前記第２処理要素に処理させることは、前記キャッシュが前記メモリ領域の初期化後の内容を記憶することをもたらさない。

【図面の簡単な説明】

【０００８】

【図１】図１は、メモリの初期化を第１処理要素から第２処理要素へ分散させるように構成されたコンピュータシステムの一実施形態を示すブロック図である。

【図２Ａ】図２Ａは、初期化前後における例示的なメモリ領域を示すブロック図である。

【図２Ｂ】図２Ｂは、初期化前後における例示的なメモリ領域を示すブロック図である。

【図３Ａ】図３Ａは、メモリの初期化を行うように構成された制御プログラムを有するメモリサブシステムの実施形態を説明するブロック図である。

【図３Ｂ】図３Ｂは、メモリの初期化を行うように構成されたオペレーティングシステムを有するメモリサブシステムの実施形態を説明するブロック図である。

【図３Ｃ】図３Ｃは、メモリの初期化を行うように構成されたＪＡＶＡ（登録商標）仮想マシンプログラムを含む実施形態を説明するブロック図である。

【図４】図４は、メモリの初期化を第１処理要素から第２処理要素へ分散する方法の一実施形態を説明するフロー図である。

【図５】図５は、メモリの初期化を第１処理要素から第２処理要素へ分散するコンピュータシステムの別の実施形態を説明するブロック図である。

【発明を実施するための形態】

【０００９】

この明細書は、「一実施形態（ｏｎｅ　ｅｍｂｏｄｉｍｅｎｔ）」または「実施形態（ａｎ　ｅｍｂｏｄｉｍｅｎｔ）」なる記載を含んでいる。「一実施形態において（ｉｎ　ｏｎｅ　ｅｍｂｏｄｉｍｅｎｔ）」または「実施形態において（ｉｎ　ａｎ　ｅｍｂｏｄｉｍｅｎｔ）」というフレーズが出現しても、同一の実施形態を意味するとはかぎらない。ある特徴、構造または特性は、この開示と整合性のある任意の適切な態様において組み合わせられてもよい。

【００１０】

（用語）

この開示（添付の請求項を含む）にみられる用語の定義および／または文脈について、

以下の数段落で述べておく。

【００１１】

「含む・有する（Ｃｏｍｐｒｉｓｉｎｇ）」または「含む・備える（Ｉｎｃｌｕｄｉｎｇ）」について。これらの用語はオープンエンドな用語である。添付した請求項において用いられているとおり、これらの用語は追加的な構造またはステップを排除するものではない。１以上の処理要素を有する装置に言及している請求項を考えてみると、このような請求項は追加的な構成要素（例えば、ネットワークインタフェースユニット、グラフィクス用回路等）を有するということから当該装置を排除するものではない。

【００１２】

「ように構成された（Ｃｏｎｆｉｇｕｒｅｄ　Ｔｏ）」について。様々なユニット、回路またはその他の構成要素が、１つまたは複数のタスクを行う「ように構成されている」というように記載されることがある。このような文脈において、「ように構成された」は、当該ユニット／回路／構成要素が、動作中に当該１つまたは複数のタスクを行う構造（例えば回路）を備えている、ということを示すことによってその構造を特定するために用いられている。そのようなわけで、当該ユニット／回路／構成要素は当該具体的なユニット／回路／構成要素が現在動作中でない（例えば、オンにされていない）場合であっても当該タスクを行うように構成されているということもできる。「ように構成された」表現と合わせて用いられるユニット／回路／構成要素は、ハードウェア（例えば回路や、処理を実施するために実行可能なメモリ記憶プログラム命令等）を含む。ユニット／回路／構成要素が１以上のタスクを行う「ように構成されている」という表現は、当該ユニット／回路／構成要素について米国特許法第１１２条第６パラグラフの適用の問題が生じないようにすることを明示的に意図したものである。追加的に、「ように構成された」は、ソフトウェアおよび／またはファームウェア（例えば、フィールド・プログラマブル・ゲート・アレイ（ＦＰＧＡ）または汎用プロセッサが実行するソフトウェア）によって操作される総称的な構造（例えば総称的回路）を含むことができる。さらに、「ように構成された」は、問題となるタスク（または複数のタスク）を行うことができるような態様にて動作するために１以上のタスクを実装しまたは行うように構成されたデバイス（例えば、集積回路）を製造するための製造工程（例えば、半導体製造施設）を適合させることを含んでもよい。

【００１３】

「処理要素（Ｐｒｏｃｅｓｓｉｎｇ　Ｅｌｅｍｅｎｔ）」について。この用語は、技術分野において通常用いられかつ受容された語義を有し、コンピュータ命令を実行することができるデバイス（例えば回路）またはデバイスの組合せを含む。処理要素は、様々な実施形態において、シングルコアプロセッサ、マルチコアプロセッサのコアの１つ、または、マルチコアプロセッサの２以上のコアからなるグループをさす。

【００１４】

「プロセッサ（Ｐｒｏｃｅｓｓｏｒ）」について。この用語は、技術分野において通常用いられかつ受容された語義を有し、１以上の処理要素を備えたデバイスを含む。プロセッサは、以下のものに限定する趣旨ではないが、中央演算処理装置（ＣＰＵ）、コプロセッサ、演算処理装置、グラフィックス処理装置、デジタルシグナルプロセッサ（ＤＳＰ）等をさすことがある。

【００１５】

「第１（ｆｉｒｓｔ）」、「第２（ｓｅｃｏｎｄ）」等について。ここに用いられているように、これらの用語は当該用語に先行された名詞に付したラベルとして用いられており、どのような形式であれ順序（例えば、空間的、経時的、論理的順序等）を示唆するものではない。例えば、８つの処理要素またはコアを有するプロセッサにおいては、「第１」処理要素および「第２」処理要素という用語は当該８つの処理要素のうちの任意の２つの処理要素を指示するために用いられることができる。換言すれば、「第１」処理要素および「第２」処理要素は論理的な処理要素０および１に限定されるものではない。

【００１６】

「コンピュータ（Ｃｏｍｐｕｔｅｒ）」または「コンピュータシステム（Ｃｏｍｐｕｔｅｒ Ｓｙｓｔｅｍ）」について。この用語は、技術分野において通常用いられかつ受容された語義を有し、一緒に動作する１以上のコンピュータデバイスおよびそれらが有する任意のソフトウェアを含む。コンピュータデバイスは、１以上の処理要素およびメモリサブシステムを含む。メモリサブシステムは、様々なタスクを行うために１以上の処理要素により実行可能なプログラム命令を記憶してもよい。

【００１７】

「コンピュータ読み取り可能な媒体（Ｃｏｍｐｕｔｅｒ－ｒｅａｄａｂｌｅ Ｍｅｄｉｕｍ）」について。ここに用いられているように、この用語は、コンピュータまたはコンピュータシステムによって読み取り可能な（非一時的かつ有形的な）媒体をさし、ハードドライブ、光ディスク、ＤＶＤ、揮発性または不揮発性ＲＡＭデバイス、ホログラフィックストレージ、プログラマブルメモリ等の磁気、光学式および半導体記憶媒体が含まれる。コンピュータ読み取り可能な媒体との関連で用いられる「非一時的な（ｎｏｎ－ｔｒａｎｓｉｔｏｒｙ）」という用語は、もっぱら、一時的な（無形な）媒体（例えば、搬送波）のような、米国特許法第１０１条の規定により適格性を有しないとされる任意の主題を請求の範囲から除外することを意図したものであって、法定の要件を具備する他の任意の主題を除外することを意図するものではない。

【００１８】

「オペレーティングシステム（Ｏｐｅｒａｔｉｎｇ Ｓｙｓｔｅｍ）」について。この用語は、技術分野において通常用いられかつ受容された語義を有し、コンピュータシステムのリソースへのアクセスを（例えばアプリケーションからの要求に応答して）制御するプログラムまたはプログラム群を含む。幾つかの実施形態においては、オペレーティングシステムは通信デバイス、ストレージデバイス等のＩ／Ｏデバイスへのアクセスを制御する。ここに記載したように、オペレーティングシステムは、ある実施形態において、第２処理要素にメモリの初期化を行わせるために実行可能な命令を含む。

【００１９】

「キャッシュ（Ｃａｃｈｅ）」について。この用語は、技術分野において通常用いられかつ受容された語義を有し、メモリまたはデータを保存するその他のストレージを含み、幾つかの他のメモリまたはストレージと比較してより高速なアクセスを提供することでこのようなデータのための将来の要求を改善することができるものである。

【００２０】

「コンピュータシステムに動作ないし演算を行わせる（Ｃａｕｓｉｎｇ ａ Ｃｏｍｐｕｔｅｒ Ｓｙｓｔｅｍ ｔｏ Ｐｅｒｆｏｒｍ Ｏｐｅｒａｔｉｏｎｓ）」について。プログラム命令の実行は、「コンピュータシステムに動作を行わせる」として記載または主張されてもよい。この語句は広く解釈されるべきであって、実行されると対象動作を行う命令や、実行されると当該動作が行われるコードをインストールまたは作成する命令が含まれる。例えば、コンピュータ読み取り可能な媒体は、コンピュータシステムにメモリ領域に対するメモリ初期化をコンピュータシステムの第１処理要素からコンピュータシステムの第２処理要素へ分散させるために実行可能な命令を含んでもよい。

【００２１】

「実行可能な（Ｅｘｅｃｕｔａｂｌｅ）」について。この用語は、技術分野において通常用いられかつ受容された語義を有し、１以上のある処理要素（すなわちあるインストラクション・セット・アーキテクチャ（ＩＳＡ））と関連付けされたフォーマットにおける命令を含むが、また、処理要素のＩＳＡのための命令を生成するために制御プログラム（例えばＪＡＶＡ（登録商標）仮想マシン）によって解釈されることができる中間フォーマット（例えば、ＪＡＶＡ（登録商標）バイトコード）での命令を含む。この定義にしたがい、第１処理要素で「実行される」プログラムは、当該第１処理要素によって実行される命令の少なくとも幾つかを含む（当該プログラムの他の命令は別の処理要素によって実行されてもよいが）。プログラムの実行はまた、プログラムの解釈を含む。

【００２２】

「アプリケーション・プログラミング・インタフェース（ＡＰＩ）」について。この用語は、技術分野において通常用いられかつ受容された語義を有し、ソフトウェアが他のソフトウェアとやりとりをすることを可能にするインタフェースを含む。プログラムは、アプリケーション、ライブラリルーチン、オペレーティングシステム等の機能を利用するためのＡＰＩ呼び出しを作成してもよい。

【００２３】

ここに記載したように、コンピュータプログラムは、コンピュータメモリをあるデータで初期化する（埋める）ことで当該メモリにそれまで記憶されていたデータを消去するという必要性を有することがある。幾つかの実施形態においては、メモリを初期化するという必要性は、（新たな）メモリの割り当てを受付ける要求にしたがって生じることがある。一実施形態においては、（他のＪＡＶＡ（登録商標）プログラムを動作させるために用いられる）ＪＡＶＡ（登録商標）仮想マシン（ＪＶＭ）プログラムは、ＪＡＶＡ（登録商標）プログラムがメモリ領域をブランクの（デフォルト）データで利用を開始することができるように、これらのメモリ領域をゼロで埋める（ｚｅｒｏ　ｏｕｔ）。別の実施形態においては、オペレーティングシステムは、例えば、ユーザプログラムが当該メモリにアクセスすることを許可する前にメモリ領域をすべてゼロで上書きすることもできる（幾つかの実施形態においては、消去されたデータは、パスワード、クレジットカード番号またはオペレーティングシステムがユーザプログラムにアクセスできることを望まないような他のデータを保持していたと考えることもできる）。他のタイプのプログラムによる多くの他の種類のメモリ初期化も想定しうるので、この開示はＪＶＭまたはオペレーティングシステムソフトウェアに限定されるものではない。初期化中にメモリ領域を埋めるデータは、すべてゼロであってもよいが必須ではない点については後述するとおりである。

【００２４】

一実施形態においては、コンピュータシステムは、中央演算処理装置（ＣＰＵ）といった、例えば汎用命令を実行するように構成された第１プロセッサを有している。コンピュータシステムはまた、グラフィックス処理装置（ＧＰＵ）といった、グラフィックス命令等の専用命令を実行するように構成された第２プロセッサを有している。他の実施形態においては、第１プロセッサ（または処理要素）は、ＣＰＵおよびＧＰＵ双方の機能性を単一デバイス、パッケージまたは集積回路内に有してもよい。コンピュータシステムはまた、メモリサブシステムを有している。一実施形態において、コンピュータシステムは、所定の命令シーケンスが前記第２のプロセッサによって行われるように構成されている（すなわち、プログラムされている）。これらの命令シーケンスは、第１プロセッサによって実行される命令によって生成されてもよく、メモリの初期化ルーチンを含むことができる。したがって、第１プロセッサは、第２プロセッサが初期化を行っている間は他のタスクを自由に行うことができる（例えば、初期化されるメモリ領域が直ちに必要とされなければ、第１プロセッサは、第２プロセッサがメモリの初期化を行っている間、プログラムの実行を継続することができる）。第１プロセッサのパフォーマンスを改善させることに加えて、ここに開示した技術は、例えばキャッシュからのデータの移動を避けることにより、第１プロセッサと関連付けられたデータキャッシュのパフォーマンスを改善させることもできる。

【００２５】

次に図１を参照すると、メモリの初期化を第１処理要素から第２処理要素へ分散させるように構成されたコンピュータシステム１０の一実施形態が図示されている。コンピュータシステム１０は、バス２０を介して連結された第１処理要素１００Ａと第２処理要素１００Ｂとを有している。一実施形態においては、バス２０により、処理要素１００Ａおよび処理要素１００Ｂはメモリサブシステム６０内の１以上のメモリ領域６４にアクセスすることができる。メモリサブシステム６０は、様々なプログラム６２を格納することができ、そのようなプログラムの幾つかは、処理要素１００Ｂを用いてメモリを初期化することを要求しまたは初期化をさせるために実行可能なプログラムである。追加的に、図１においては視覚的に区別できる構成要素として図示されているが、メモリサブシステム６０

の一部またはすべては、処理要素１００Ａ、処理要素１００Ｂの回路の一部、または、処理要素１００Ａおよび処理要素１００Ｂの双方を備えた単一のデバイスの一部をなしてもよい。一実施形態においては、キャッシュ３０は、処理要素１００Ａからアクセス可能であって、メモリサブシステム６０に保存されたデータに対応するデータを保存するように構成されている。一実施形態においては、メモリアクセスコントローラ７５は、処理要素１００Ａ、処理要素１００Ｂ、メモリサブシステム６０の任意の組合せに接続される（またはその内部に実装され）ことができ、また、バス２０に接続されてもよい。コンピュータシステム１０は、様々な実施形態において異なる態様にて構成され得る。

【００２６】

処理要素１００Ａおよび処理要素１００Ｂは、任意のタイプのプロセッサ（例えば、中央演算処理装置、演算処理装置、グラフィックス処理装置、デジタル信号処理装置等）に相当して（またはその内部に配置されて）もよい。一実施形態においては、処理要素１００Ａは中央演算処理装置（または１以上のコアからなるグループ）であり、処理要素１００Ｂは異なるタイプの処理装置、例えば、グラフィックス処理装置（１以上のコアを有してもよい）である。幾つかの実施形態においては、処理要素１００Ａおよび処理要素１００Ｂのいずれかまたは双方は、複数のコアを有してもよい。他の実施形態においては、処理要素１００Ａおよび処理要素１００Ｂは、同一チップ上に配置された１以上のプロセッサ・コアからなる異なるグループとされてもよい。処理要素１００Ａおよび処理要素１００Ｂは、幾つかの実施形態においては、様々な処理要素からなるクラスタまたはグループで構成されてもよい（例えば、処理要素１００Ａは２つのクアッドコア・プロセッサからなるグループとされることができる）。

【００２７】

一実施形態においては、処理要素をメモリサブシステム６０に接続するバス２０は、ノースブリッジバス、任意の他のプロセッサバスまたは当業者に公知のプロセッサ相互接続部であってもよい。バス２０は、一実施形態においては、同一チップ上に配置されてもよい（１以上のグループの）プロセッサ・コア間の相互接続部である。バス２０は、単一のバスまたは相互接続部に限定される必要はなく、１以上のバス、（２地点間）相互接続部、またはここに記載した構造体にデータを伝達するのに適したその他の通信経路および通信装置の任意の組合せであってもよい。

【００２８】

メモリサブシステム６０は、１以上のメモリデバイスを備えている。様々な実施形態において、これらのメモリデバイスは、ＲＡＭモジュール、組込みメモリ（例えばｅＤＲＡＭ）、ソリッドステート記憶デバイス、ハードドライブといった２次記憶装置または上記定義のとおりその他の任意のコンピュータ読み取り可能な媒体で構成されてもよい。一実施形態においては、メモリサブシステム６０は、当該メモリサブシステム６０における１以上のメモリデバイス内に１以上のメモリ領域６４を備えている。メモリ領域６４は固定サイズまたは固定ロケーションである必要はなく、これに代えて任意の開始位置（または開始アドレス）および終了位置（または終了アドレス）を含むメモリの１以上の部分をさすものであってもよい。このように、具体的な一実施形態においては、第１メモリ領域はサイズが４０００ＫＢの一連のメモリロケーションであってもよく、他方、第２メモリ領域はサイズが３２ＫＢの一連のメモリロケーションであってもよい。一実施形態においては、メモリ領域６４は、複数のメモリデバイスにわたって（または複数タイプのメモリデバイスにわたって、例えば、単一のメモリ領域がＲＡＭモジュールとハードドライブとの記憶スペースで構成されるように）構成されてもよい。メモリ領域は、物理的または論理的に連続的であってもよく、連続的でなくてもよい。

【００２９】

メモリサブシステム６０とそのメモリ領域は、処理要素１００Ａによってアクセス可能とされている。例えば、処理要素１００Ａは、バス２０を介してメモリサブシステム６０からデータを取得（またはデータをメモリサブシステム６０に保存）してもよい。様々な実施形態において、以下に説明するように、メモリサブシステム６０はまた、処理要素１

００Ｂによってもアクセス可能とされている。様々な実施形態において、メモリサブシステム６０は、１以上のプログラム６２を記憶している。プログラム６２（または複数のプログラム）は、コンピュータシステム１０上で実行可能な任意のプログラム（または複数のプログラム）とされてもよい。このように、様々な実施形態において、プログラム６２は、Ｊａｖａ（登録商標）仮想マシン（ＪＶＭ）、オペレーティングシステム、ＡＰＩライブラリ、ＪＶＭまたはオペレーティングシステム上で動作するユーザプログラム等であってもよい。様々な実施形態において、プログラム６２は、以下にさらに議論するように、メモリの初期化を処理要素１００Ａから処理要素１００Ｂへ分散させる機能を有することができる。

【００３０】

メモリアクセスコントローラ７５は、一実施形態においてはメモリサブシステム６０に接続されており、様々な実施形態においてメモリサブシステム６０の制御、管理、調整および／または処理要素１００からメモリサブシステム６０へのメモリアクセスの許可を行うように構成されている。メモリアクセスコントローラ７５は、一実施形態においてはダイレクトメモリアクセス（ＤＭＡ）コントローラであり、処理要素１００Ａおよび／または処理要素１００Ｂと同一チップ上に配置されてもよい。様々な実施形態において、メモリアクセスコントローラ７５は、処理要素１００Ａによって警告、通知を受けまたは権限を付与された場合を除いて処理要素１００Ｂがメモリ領域６４にアクセスすることを制限してもよい（このような場合には、アクセスコントローラ７５はメモリサブシステム６０の領域の幾つかまたはすべてへのアクセスを許可してもよい）。メモリアクセスコントローラ７５は、一実施形態においてバス２０を使用し（および／またはバス２０に接続され）てもよい。

【００３１】

キャッシュ３０は、処理要素１００Ａによりアクセス可能とされており、メモリサブシステム６０に対応するデータを保持するように構成されたキャッシュを有している。キャッシュ３０は、このようにして、当該データへの処理要素１００Ａの高速アクセスを提供するために、メモリサブシステム６０に保存されたデータのサブセットを保持するように構成されてもよい。様々な実施形態において、キャッシュ３０は、Ｌ１キャッシュ，Ｌ２キャッシュ，Ｌ３キャッシュまたはその他のキャッシュを含む階層型キャッシュシステムを含んでもよい。キャッシュ３０は、様々な実施形態において処理要素１００Ａ内に部分的または完全に配置されてもよく、あるいは、処理要素１００Ａの外部に部分的または完全に配置されてもよい（例えば、一実施形態においては、キャッシュ３０は処理要素１００Ａ内のＬ１キャッシュおよび処理要素１００Ａの外部のＬ２キャッシュとで構成されている）。所定の処理要素と「関連付け」られたキャッシュは、当該処理要素によってアクセスされるように構成されている。

【００３２】

幾つかの例においては、キャッシュ処理によって、キャッシュ３０に既に記憶されていたデータが他のデータと置換される（または他のデータによって置換される）ことがある。幾つかの実施形態においては、処理要素１００Ａがメモリサブシステム６０のメモリ領域に直接アクセスする際には、アクセスされたデータを記憶するためにキャッシュ３０の一部が用いられることになろう。例えば、もし処理要素１００Ａがメモリサブシステム６０に直接アクセスしてメモリ領域６４を初期化するのであれば、キャッシュ３０内の既存データが当該メモリ領域のための新たに初期化されたデータによって移動されることが考えられる。キャッシュから移動されたデータはアクセスに時間がかかることもあり、これにより実行時間が長くなりうる。例えば、以下のＣコードを考えてみるとよい：

```
int C = A + B;
int *Freespace = malloc(8192);
E = C;
```

このコードでは（コンパイルおよび実行されると）、キャッシュされる変数「Ｃ」のためのデータ値が最初にもたらされる。次いで、ｍａｌｌｏｃ（）関数の呼び出しによりメモ

リ８１９２バイト分の初期化が生じると考えられ、キャッシュから「Ｃ」のための値が移動される。次の命令（「Ｃ」の値を変数Ｅに割り当てる命令）が実行されると、キャッシュは「ミス（外れ）」となり、より低レベルのキャッシュまたはより離隔したメモリから変数Ｃの値を取得しなければならなくなり、遅延が生じる。もしＣの値が最初の段階でキャッシュから移動されていなかったとすれば、この遅延は回避し得たのであって、パフォーマンスがスピードアップしていたかもしれない。キャッシュ３０についてのデータの移動／置換は、様々な実施形態において、ＬＲＵ（ｌｅａｓｔ　ｒｅｃｅｎｔｌｙ　ｕｓｅｄ）方式の置換を含む、当業者が考えうる任意の数のハードウェアまたはソフトウェアスキームを含む置換ポリシーによって規律されてもよい。

【００３３】

次に図２Ａを参照すると、初期化前のメモリ領域６４の一例が図示されている。図示されているように、メモリ領域６４は、複数のメモリロケーション（ロケーション２１２－２１６を含む）を備えており、それぞれが個別にアドレス可能とされており様々な実施形態において所定量のデータを記憶するように構成されている。図示されているように、メモリロケーション２１２は、データ２０５を記憶している。メモリロケーション２１２内のデータ２０５は、幾つかの実施形態においては、コンピュータシステム１０によって実行されているプログラムによってあらかじめ書き込まれたものであってもよく、あるいは、単純に任意的（ランダム）であってもよい。

【００３４】

図２Ｂにおいて、初期化後のメモリ領域６４の一例が図示されている。この実施形態においては、メモリロケーション２１２内のデータ２０５は、ゼロの値を有する複数ビットからなるビット列に初期化することにより「ゼロで埋められ（ｚｅｒｏｅｄ　ｏｕｔ）」ている。ここにさらに議論されるように、この初期化は、は幾つかの実施形態においては処理要素１００Ｂによって行われる。「ゼロで埋める処理（Ｚｅｒｏｉｎｇ　ｏｕｔ）」は初期化の一形態にすぎない；その他の初期化には、テストパターンでデータを書き込む（例えば、すべて負の値に対応する値、１６進値０ｘＤＥＡＤＢＥＥＦ等）が含まれる。初期化は、幾つかの実施形態においては、ＪＡＶＡ（登録商標）プログラミング言語仕様といった外部仕様にしたがって行われてもよい。初期化は上述したデータ形式および値に限定されるものではなく、様々な実施形態において、１以上のメモリ領域を埋める任意のデータを含んでもよい。

【００３５】

幾つかの実施形態においては、メモリの初期化は、ある最小サイズのメモリ領域を初期化することに限定されてもよい（最小サイズについては初期化要求に応じる制御プログラムが任意に判断することが考えられる）。例えば、メモリの初期化は、（例えば８ＫＢのページといった、コンピュータシステム１０のオペレーティングシステムによって定義されるような）ページ１単位、キャッシュライン幅または（１０２４バイトといった）所定の固定サイズ等を下回らない領域を初期化することに限定されてもよい。これらの実施形態においては、メモリの初期化のための最小サイズ閾値は、小さいメモリ領域を初期化するために第２処理要素を使用することに伴い生じうる性能ペナルティを回避するために設定されることも考えられるが、これは、第１処理要素ではなく第２処理要素を使用して初期化を行うことで様々な実施形態において不可避的なオーバーヘッド・コストが生じることがあるからである。

【００３６】

次に図３Ａを参照すると、メモリサブシステム６０内にユーザプログラム３０４と制御プログラム３１０とを備えた実施形態を示すブロック図が図示されている。一実施形態においては、プログラム３０４およびプログラム３１０はともに図１に関して前述した各別のプログラム６２である。様々な実施形態において、ユーザプログラム３０４は、メモリに直接アクセスしてメモリ領域（複数のメモリ領域）を初期化する権限を欠いている（またはそのようにプログラムおよび／または設計されていない）こともあるが、他方、制御プログラム３１０は、メモリ領域を初期化するように実行可能とされている（例えば初期

化ルーチン３１３を使用して）。例えばプログラム３０４はＪＡＶＡ（登録商標）プロセスおよび／またはユーザアプリケーションであってもよく、他方、プログラム３１０はＪＶＭまたはオペレーティングシステムであってもよい（下記の図３Ｂないし図３Ｃによる説明を参照）。様々な実施形態において、ユーザプログラム３０４および制御プログラム３１０は、サブシステム６０内の１以上のメモリデバイス内に記憶されている（例えば、制御プログラム３１０はハードドライブに記憶されたうえで実行中にＲＡＭモジュールに（完全または部分的に）ロードされてもよい）。

【００３７】

制御プログラム３１０は、様々な実施形態において、処理要素１００Ａおよび／または処理要素１００Ｂによって実行可能な命令を有している。すなわち、所定の制御プログラム３１０は、処理要素１００Ａ、処理要素１００Ｂ、あるいは処理要素１００Ａおよび処理要素１００Ｂの組合せによって実行可能な命令を有してもよい。例えば、一実施形態においては、制御プログラム３１０は、処理要素１００Ａおよび処理要素１００Ｂの双方によって実行可能な単一のインストラクション・セット・アーキテクチャ（ｉｎｓｔｒｕｃｔｉｏｎ　ｓｅｔ　ａｒｃｈｉｔｅｃｔｕｒｅ；ＩＳＡ）に属する命令を有している。他方、別の実施形態においては、制御プログラム３１０は、処理要素１００Ａにより実行可能な第１のＩＳＡに属する命令を有しており、また、処理要素１００Ｂにより実行可能な第２の異なるＩＳＡに属する命令を有している。メモリの初期化ルーチン３１３は、このようにして、幾つかの実施形態において制御プログラム３１０の他の部分とは異なるＩＳＡに属する命令を有してもよい。

【００３８】

一実施形態においては、制御プログラム３１０は、ユーザプログラム３０４からメモリ要求３０５を受付けるために実行可能な初期化ルーチン３１３を含むプログラム命令群を備えている（別の実施形態においては、制御プログラム３１０は内部でメモリ要求３０５を生成する）。メモリの初期化ルーチン３１３は、処理要素１００Ｂ（処理要素１００Ａではなく）に初期化要求３０５で指定される１以上のメモリ領域６４を初期化させるために実行可能である。メモリの初期化ルーチン３１３は、様々な実施形態において、ＯＰＥＮＣＬ、ＪＡＶＡ（登録商標）、Ｃ＋＋等のプログラミング言語で書かれたコードに対応する命令を含む。ルーチン３１３に対応するコードは初期化ルーチン３１３を様々な実施形態にて行うために解釈および／またはコンパイルされてもよい。

【００３９】

ＯＰＥＮＣＬコードがどのようにして処理要素１００Ｂにより実行可能な命令を生成するために用いることができるかの一例は、２０１０年５月２１日出願の米国特許出願第１２／７８５，０５２号、発明の名称：「コンピューティング・プラットフォームにおける処理負荷の分散（ＤＩＳＴＲＩＢＵＴＩＮＧ　ＷＯＲＫＬＯＡＤＳ　ＩＮ　Ａ　ＣＯＭＰＵＴＩＮＧ　ＰＬＡＴＦＯＲＭ）」に見出すことができるところ、その内容は引用により本明細書に組み込まれる。

【００４０】

メモリの初期化ルーチン３１３は、様々な実施形態において、処理要素１００Ｂにメモリ領域６４を初期化させるために実行されてもよい。一実施形態においては、初期化ルーチン３１３の実行は、初期化要求３０５に応答して開始され、この初期化要求３０５は、ユーザプログラム３０４によって生成されてもよい。初期化要求３０５は、様々な実施形態において様々な形式をとってもよく、初期化すべき１以上のメモリ領域６４を識別または判定するのに利用できる情報を含んでいる。一実施形態においては、要求３０５は、データオブジェクトの名称を指定する。一実施形態においては、初期化要求３０５は、初期化されるべきメモリ空間のメモリベースアドレスおよびオフセット値（長さ）を含んでいる。他の実施形態においては、初期化要求３０５は、初期化されるべきメモリの基底の「開始」アドレスおよびメモリ上限の「停止」アドレスを含んでいる。しかしながら、メモリ要求３０５は、このように限定されたものではなく、初期化されるべき１以上のメモリ領域６４を判定するのに利用可能な任意の情報を含んでもよい。

【００４１】
実行中、制御プログラム３１０は、処理要素１００Ａおよび／または処理要素１００Ｂによって実行されるが、少なくとも一実施形態においては、初期化ルーチン３１３の実行は、初期化要求３０７を用いてもっぱら処理要素１００Ｂによって行われる。処理要素１００Ｂによるルーチン３１３の実行は、様々な実施形態において異なる態様にて進行してもよい。一実施形態においては、制御プログラム３１０の一部は、処理要素１００Ｂによるルーチン３１３の実行を「設定」するために処理要素１００Ａによって実行可能とされてもよい。処理要素１００Ａは、ルーチン３１３への参照を含む処理要素１００Ｂへの制御メッセージ、通知または命令を送信してもよい。このような制御メッセージを受付けると、処理要素１００Ｂはさらにルーチン３１３を実行することが考えられる（例えば、ルーチン３１３の命令が記憶されているメモリおよび／またはキャッシュに直接アクセスすることによって）。別の実施形態においては、初期化ルーチン３１３に対する命令は、単純にバス（バス２０等）上に出力されてもよく、その時点で処理要素１００Ｂは当該命令を認識して実行するであろう。一実施形態においては、処理要素１００Ａは、処理要素１００Ｂがメモリ領域６４に直接アクセスすることをメモリアクセスコントローラ７５に許可させるコンフィギュレーション動作を含む、処理要素１００Ｂのための１以上のコンフィギュレーション動作を行うために、（処理要素１００ＡのＩＳＡに属する）命令を実行してもよい。処理要素１００Ｂに初期化ルーチン３１３を実行させるために、当業者が想到しうる様々な他の技術もまた利用することができる。

【００４２】
初期化ルーチン３１３の命令は、一実施形態においては、初期化すべき１以上のメモリ領域６４を指定する１以上の参照と１以上のメモリ領域を初期化するために処理要素１００Ｂにより実行可能な命令とを含む。初期化されたメモリ領域を埋めるデータはすべてゼロ、すべて負の値、パターン化されたデータまたは前述したようなその他のデータとすることができる。幾つかの実施形態においては、初期化ルーチン３１３の一部分（またはその全体）は、制御プログラム３１０によって動的に生成されてもよい。動的生成は、一実施形態においてはメモリ要求３０５に含まれる情報に応答して行われてもよい。例えば、もしメモリ要求３０５がＲＡＭのうち８ＭＢ相当の部分が初期化すべきであると指定しているならば、少なくとも初期化ルーチン３１３の一部分はこの８ＭＢ値を反映するように動的に変更されてもよい。

【００４３】
初期化ルーチン３１３は、様々なソフトウェアプログラムの一部として行われてもよい。例えば、一実施形態においては、ルーチン３１３は、アプリケーション・プログラミング・インタフェース（ＡＰＩ）の仕様にしたがってメモリ要求３０５が作成されてライブラリルーチンの一部として行われてもよい。別の実施形態においては、ルーチン３１３は、ＪＡＶＡ（登録商標）ガーベジコレクションプロセスの一部として行われてもよい（図３Ｃを参照して後述するとおり）。しかしながら、初期化ルーチン３１３は上述したタイプのプログラムに限定されるものではない。

【００４４】
次に図３Ｂを参照すると、コンピュータシステム１０のオペレーティングシステム３２０がメモリの初期化を第１処理要素から第２処理要素へ分散させるように構成されている実施形態を示したブロック図が図示されている。一実施形態においては、オペレーティングシステム３２０は、制御プログラム３１０について前述した動作のいずれかまたはすべてを完全にまたは部分的に行うようにしてもよい。様々な実施形態において、オペレーティングシステム３２０は、１以上のメモリ要求３０５を受付、生成および／または処理して１以上のメモリ領域６４を初期化してもよい。一実施形態においては、メモリ要求３０５は、オペレーティングシステム３２０内のライブラリ（またはモジュール）により受付けられてもよく、プログラム６２、プログラム３０４といったプログラムまたはオペレーティングシステム３２０それ自体によって呼び出し可能とされてもよい。これらのライブラリは、様々な実施形態において、メモリサブシステム６０内の１以上のファイルとして

記憶されてもよく、要素３２２，３２４といったモジュールのためのＡＰＩインタフェースを有してもよい、これは、Ｃプログラミング言語関数ｍａｌｌｏｃ（）およびｉｎｉｔ（）に対応するものである。例えば、コンピュータシステム１０上で実行されるプログラム６２は、ｍａｌｌｏｃ（）ルーチンを呼び出すことにより（より多くの）メモリの割り当てを受けることを要求してもよい。オペレーティングシステム３２０はしたがって、一実施形態においては、適切な命令（初期化ルーチン３１３等）をロードおよび／または動的に生成したうえでこれらのロードまたは生成された命令を第２処理要素１００Ｂに実行させることにより、当該要求に応じる。このような初期化は、例えば初めて割り当てられたメモリブロックがデータを１つのプログラムから別のプログラムへと漏洩することを回避するなどのセキュリティ上の理由で望ましいであろう。Ｉｎｉｔモジュール３２４は、一実施形態においてはメモリに別のプロセスをロードするために用いられてもよく、そのためメモリ要求３０５を内部で生成してもよいだろう（これに応じて初期化要求３０７が処理要素１００Ｂへ送信されてもよい）。

【００４５】

次に図３Ｃを参照すると、ＪＡＶＡ（登録商標）仮想マシン（ＪＶＭ）３３０がメモリの初期化を第１処理要素から第２処理要素へ分散させるように構成されている実施形態を示したブロック図が図示されている。ＪＶＭ３３０は、制御プログラム３１０について前述した動作のいずれかまたはすべてを完全にまたは部分的に行うようにしてもよく、メモリサブシステム６０（図示せず）に記憶されてもよい。一実施形態においては、ＪＶＭ３３０は、メモリサブシステム６０に記憶された１以上のＪＡＶＡ（登録商標）プログラムのＪＡＶＡ（登録商標）バイトコードを実行するように構成されている（したがって、制御プログラム３１０はこのようにして他のプログラムを実行してもよく、さらに、この観点ではＪＡＶＡ（登録商標）プログラムに限定されるものではない）。ＪＡＶＡ（登録商標）バイトコードの実行は、任意の数のＪＡＶＡ（登録商標）オブジェクト３３１が作成および／または破棄されることを生じさせるかもしれない。ＪＡＶＡ（登録商標）オブジェクトのためのデフォルトの初期値は、ＪＶＭ３３０の様々な実施形態においてすべてゼロに設定されてもよい。このような初期化は、様々な実施形態において、ガーベジコレクションプロセス３３２および／またはコンストラクタルーチン３３４（これは、幾つかの実施形態においては、全体としてまたは部分的に、初期化ルーチン３１３に対応してもよい）によって行われてもよい。ガーベジコレクションプロセス３３２の最後のステップとして、一実施形態においては、１以上のメモリ領域のすべてが、メモリ領域をゼロで埋めることにより将来のオブジェクト割り当てのために利用可能とされてもよい（次のガーベジコレクションが追加的に初期化されたメモリをもたらすまで、これにより既に初期化されたメモリを確保する）。あるいは、ゼロで埋める処理は、様々な実施形態において、新たなオブジェクトがＪＡＶＡ（登録商標）ユーザプログラムによって割り当てを受けるたびにその都ベースで行われることができる。

【００４６】

一実施形態においては、ガーベジコレクションプロセス３３２は、どのＪＡＶＡ（登録商標）オブジェクトがもはや使用されていないかを判定してそれらの使用されていないオブジェクトに対するメモリの割り当てを解除する。このメモリの割り当て解除のプロセスにおいて、ＪＶＭ３３０は、１以上の対応するメモリ領域を初期化してゼロ値を含ませるようにしてもよい。ＪＶＭ３３０はまた、１以上のコンストラクタルーチン３３４（または複数のルーチン）を実行させるようにしてもよい。コンストラクタルーチン３３４（または複数のルーチン）は、デフォルトのルーチンであってもよく、解放されているメモリの割り当て、ＪＶＭ３３０で実行されている１以上のＪＡＶＡ（登録商標）プログラムに対して行うことを必要とするかもしれず、同様にそれらのＪＡＶＡ（登録商標）プログラム（これは様々な実施形態においてユーザプログラム３０４に対応してもよい）の実行中に１以上のメモリ領域６４の初期化を行わせてもよい。ＪＶＭ３３０によるメモリ領域の初期化を最適化するための様々な技術、変形が当業者において想起されると思われる。例えば、ＪＶＭ３３０は、より大きな容量のメモリ（例えば１ＭＢ）を「ゼロ」にして新た

に作成されたＪＡＶＡ（登録商標）オブジェクトの要求を満たすのに必要なだけメモリ分配するように構成されてもよい（クラスが作成されるたびにその都度メモリを初期化するのではなく）。

【００４７】

様々な実施形態において、オペレーティングシステム３２０およびＪＶＭ３３０を除く多くのプログラムが、コンピュータシステム１０にメモリ領域を初期化するタスクの処理要素１００Ａから処理要素１００Ｂへの分散を、コンピュータシステム１０行わせてもよい。コンピュータシステム１０の処理要素によってコンパイルおよび実行（または解釈）されるように設計された異なるプログラミング言語は、メモリの初期化の分散（またはオフロード）能力の利点を活用するように設計されたＡＰＩルーチンを含むライブラリを有してもよい。さらにコンパイラは、高水準ソースコードから実行可能なコードを生成する際に、ここに記載した技術を用いてメモリの初期化の分散を生じさせるように設計されることも考えうる。コンパイラは、一実施形態においては、プログラムが１以上のメモリ埋め処理を第１処理要素から第２要素へ分散することでどの段階で利益を得ることができるかを判定するために、ヒューリスティックスを利用することができる（例えば、このようなヒューリスティックスの基礎をなしうる要素としては、メモリ領域のサイズ（当該領域が十分に大きい場合にはおそらくオフロード／分散）、初期化に続いてどの程度頻繁にかつ直ちにメモリ領域がアクセスされるか、初期化が行われた後所定の期間においてアクセスされた初期化された領域のバイト数、所定のメモリの初期化をオフロード化しないことによるキャッシュ移動の結果として予想されるキャッシュミスの量等が考えられる）。幾つかの実施形態においては、ここに記載したメモリの初期化技術は、ソースコードプログラマに対して幾つかの場合において透過的である。例えば、ソースコードプログラマは、Ｃプログラミング言語によるｍａｌｌｏｃ（）に対する呼び出しを、当該プログラミング言語の言語仕様にしたがって、当該呼び出しを扱うライブラリルーチンがメモリの初期化を第１要素から第２要素へと分散させることを認識することなくプログラムすることができる。

【００４８】

次に図４を参照すると、１以上のメモリ領域の初期化の第１処理要素による第２処理要素へのオフロードに係る方法４００の一実施形態のフロー図が図示されている。方法４００は、全体としてまたは部分的に、コンピュータシステム１０または後述するシステム５００といった他の任意の適切なコンピュータシステムまたはコンピュータデバイスによって行われてもよい。ステップ４１０において、初期化される１以上のメモリ領域の指示が受付けられる。このステップは、一実施形態においては、例えばプログラム３０４からのメモリ要求を受け付けるために制御プログラム３１０を実行する処理要素１００Ａによって行われてもよい。一実施形態においては、ステップ４１０は、ＪＶＭ３３０のプロセス３３２といったガーベジコレクションプロセスによって生成された要求を受け付けることが含まれる。

【００４９】

ステップ４２０において、ステップ４１０の指示の受付に応答して、コンピュータシステム１０は、要求されたメモリ領域の初期化を処理要素１００Ａから処理要素１００Ｂへオフロードする。一実施形態においては、ステップ４２０は、処理要素１００Ａによって行われ、要求されたメモリ領域の初期化を処理要素１００Ｂへオフロードさせる。ステップ４２０はまた、様々な実施形態において、コンフィギュレーション動作を行うかそうでなければ処理要素１００Ｂにメモリ領域６４を初期化させるような態様にて処理要素１００Ｂとやり取りを行う処理要素１００Ａを含んでもよい（例えば初期化ルーチン３１３を実行するように処理要素１００Ｂを設定するなど）。

【００５０】

ステップ４３０において、ステップ４１０の初期化要求がオフロードされた（すなわち分散された）処理要素は、指示された１以上のメモリ領域を初期化する。一実施形態においては、このステップは要求されたメモリ領域を初期化するために（コントローラ７５を

介して）ダイレクトメモリアクセスを利用して処理要素１００Ｂによって行われる。このように、ステップ４３０の様々な実施形態において、初期化は、処理要素１００Ａが初期化されるべきメモリ領域のための値を直接変更することなしに行われる。ある実施形態において、ステップ４３０は、制御プログラム３１０の１以上の所定のルール、ルーチン等にしたがって行われる。これらのルールは、ヒューリスティクス（例えば前述したようなヒューリスティクス）を含むことが考えられる。

【００５１】
ステップ４４０において、コンピュータシステム１０のキャッシュの１以上の部分を無効化してもよい。複数の処理要素（コンピュータシステム１０等）を有するシステムにおいては、メモリ領域６４内のデータのコピーは、幾つかの実施形態においてメモリ階層（キャッシュ３０を含む）に記憶されてもよい。もしメモリ領域６４が方法４００にしたがって初期化されるならば、幾つかの事例においておよび幾つかの実施形態においてはキャッシュ無効化手順を行って、コンピュータシステム１０のキャッシュ（例えば、キャッシュ３０）内に残っている初期化されたメモリ領域６４に対応するデータの古いコピーが存在しないことを確保することが必要となることがある。ステップ４４０は、様々な実施形態において、処理要素１００Ａ、処理要素１００Ｂおよび／またはメモリアクセスコントローラ７５により様々に開始されてもよく、当業者に知られた様々な技術を用いて行ってもよい。

【００５２】
次に図５を参照すると、前述した様々な実施形態を実装することができる例示的なコンピュータシステム５００を示すブロック図が図示されている。コンピュータシステム５００のコンポーネントは、全体的または部分的にコンピュータシステム１０のコンポーネントと同一または類似であってよい。例えば、コンピュータシステム５００は図示したとおり、メモリサブシステム６０、処理要素１００Ａ、処理要素１００Ｂ、キャッシュ３０、メモリアクセスコントローラ７５を備えている。コンピュータシステム５００は、サーバシステム、パーソナルコンピュータシステム、デスクトップコンピュータ、ラップトップノートブックコンピュータ、メインフレームコンピュータシステム、携帯用コンピュータ、ワークステーション、ネットワークコンピュータ、携帯電話、ポケットベルまたは携帯情報端末（ＰＤＡ）といった消費者装置等の任意の様々なタイプのデバイスを含んでもよいが、これらに限定されるものではない。コンピュータシステム５００はまた、ストレージデバイス、スイッチ、モデム、ルータ等の任意のタイプのネットワーク接続周辺デバイスであってもよい。図５には便宜上単一のコンピュータシステム５００が図示されているが、システム５００はまた、一緒に動作する２以上のコンピュータシステムとして実装されてもよい。

【００５３】
コンピュータシステム５００の一実施形態において、メモリサブシステム６０は、２次記憶装置４５５とＲＡＭモジュール４４４，４４６とを備えている。一実施形態においては、２次記憶装置４５５には、プログラム命令が記憶されており、これらのプログラム命令は、コンピュータシステムに初期化されるべきメモリ領域の指示を受付けさせるために、第１処理要素１００Ａにより実行可能とされている。ここで、メモリ領域はコンピュータシステムのメモリ内にあり、前記指示の受付けに応答して、メモリ領域の初期化をコンピュータデバイスの処理要素１００Ｂに処理させる。処理要素１００Ａおよび処理要素１００Ｂは、ある実施形態において、異種（すなわち異なるタイプに属する）のものであってよい。例えば、処理要素１００Ａが中央演算処理装置（ＣＰＵ）であり、処理要素１００Ｂがグラフィックス処理装置（ＧＰＵ）であるような場合であってもよい。さらに、一実施形態においては、キャッシュ３０は、処理要素１００Ａがメモリにアクセスするのに応じてメモリサブシステム６０内の１以上のメモリデバイスの内容を記憶するように構成されてもよく、ここで、メモリ領域の初期化を行うことには、当該メモリ領域の初期化後の内容をキャッシュに記憶させることを含まない。すなわちキャッシュ３０は、初期化されたメモリ領域に対応する新たに初期化されたデータによって他のデータがキャッシュ３

０内から移動されることを、回避することができる。メモリアクセスコントローラ７５は、様々な実施形態において、処理要素１００Ｂに対してメモリサブシステム６０内の１以上のメモリデバイスへのアクセスを提供するように構成されてもよく、ここで、メモリ領域の初期化を行うことには、処理要素１００Ｂがメモリアクセスコントローラ７５を用いてメモリ領域にアクセスすることが含まれ、かつ、初期化を行うことには処理要素１００Ａがメモリ領域にアクセス（すなわちメモリ領域を変更）することは含まれない。

【００５４】

追加的に、一実施形態においては、Ｉ／Ｏ装置４４４は、バス２０を介してメモリサブシステム６０に接続されている。様々な実施形態において、Ｉ／Ｏ装置は、他のストレージデバイス（ハードドライブ、光学ドライブ、リムーバブルフラッシュドライブ、ストレージアレイ、ＳＡＮまたはそれらに関連するコントローラ）、ネットワークインタフェースデバイス（例えばローカルエリアまたはワイドエリアネットワーク）またはその他のデバイス（例えばグラフィックス、ユーザインタフェースデバイス等）を含んでもよい。一実施形態においては、コンピュータシステム５００は、ネットワークインタフェースデバイスを介してネットワークに接続されている。Ｉ／Ｏ装置は、様々なタイプのインタフェースを含んでもよく、これは様々な実施形態にしたがって、他のデバイスおよびそれらのインタフェースに接続され通信を行うように構成されてもよい。一実施形態においては、Ｉ／Ｏインタフェースはフロントサイドバスから１以上のバックサイドバスへのブリッジチップ（例えばサウスブリッジ）である。

【００５５】

メモリサブシステム６０は、処理要素１００Ａおよび／または処理要素１００Ｂにより様々な実施形態において利用可能なメモリを有している。サブシステム６０内のメモリは、ハードディスク・ストレージ、フロッピー（登録商標）ディスク・ストレージ、リムーバブル・ディスク・ストレージ、フラッシュメモリ、ランダムアクセスメモリ（ＲＡＭ—ＳＲＡＭ、ＥＤＯ　ＲＡＭ、ＳＤＲＡＭ、ＤＤＲ　ＳＤＲＡＭ、ＲＡＭＢＵＳ　ＲＡＭ等）、読み出し専用メモリ（ＰＲＯＭ、ＥＥＰＲＯＭ等）といった異なる物理的記憶媒体を用いて実装されてもよい。コンピュータシステム５００内のメモリはＲＡＭ４４４，４４６および２次記憶装置４５５のようなストレージに限定されるものではない。むしろ、コンピュータシステム５００はまた、ここに記載しなかったキャッシュメモリやＩ／Ｏ装置４４４（例えば、ハードドライブ、ストレージアレイ等）上のストレージといった他の形態のストレージを有してもよい。幾つかの実施形態においては、これらの他の形態のストレージはまた、処理要素１００Ａおよび／または処理要素１００Ｂにより実行可能なプログラム命令を記憶してもよい。

【００５６】

前述した技術および方法は、任意の適切なコンピュータ読み取り可能な媒体に記憶されたコンピュータ読み取り可能な命令として実装されてもよい。これらの命令は、コンピュータシステムおよび／またはコンピュータデバイスが前述した態様にて動作することを可能にするソフトウェアであってもよく、メモリサブシステム６０内のコンピュータ読み取り可能な媒体（またはメモリサブシステム６０内にはない別のコンピュータ読み取り可能な媒体）に記憶されてもよい。このように、ライブラリルーチン、ガーベジコレクションプロセス、その他のソフトウェアルーチンおよびオブジェクトならびに任意のもしくはすべてのソフトウェア６２，３０４，３１０，３１３，３２０，３２２，３２４，３３０，３３１，３３２，３３４はこのようなコンピュータ読み取り可能な媒体上に記憶されてもよい（本明細書の段落［００１７］に注記したとおり、このような媒体は非一時的なものとされてもよい）。

【００５７】

さらに、前述した技術および方法は、幾つかの実施形態においてはハードウェアとして実装されてもよい。例えば、一実施形態は、メモリデバイスのメモリ領域の初期化を第２処理要素に処理させるように構成されたメモリ初期化回路を備えた処理要素であり、ここで、初期化を行うことは、メモリ領域が初期化されるべきであるとの指示に応答して行わ

れる。ハードウェア実施形態は、前述したアルゴリズムおよび技術を実装するための回路論理を用いてもよい（例えば、方法４００）。

【００５８】

ハードウェアの実施形態は、ハードウェア生成命令を用いて生成されてもよい。例えば、ハードウェア生成命令は、ＶｅｒｉｌｏｇまたはＶＨＤＬといった高水準設計言語（ＨＤＬ）によるビヘイビア・レベルまたはレジスタ転送レベル（ＲＴＬ）の記述レベルでハードウェア機能を記述した１以上のデータ構造の概要を示すものであってもよい。記述内容は合成ツールにより読み取られてもよく、合成ツールは記述を合成してネットリストを生成するものであってもよい。ネットリストは、ゲート群（例えば合成ライブラリに定義された）を含んでもよく、メモリの初期化の分散化／オフロード化を実現するように構成された処理要素（１００Ａおよび／または１００Ｂ等）の機能性を表現するものである。このようなネットリストを配置・経路処理したうえでマスクに対して適用すべき幾何学形状を記述したデータセットが生成されてもよい。このようなマスクは１以上の処理要素（１００Ａおよび／または１００Ｂ等）に対応する半導体回路または回路系を製造するために様々な半導体製造工程において利用されてもよい。代替的に、データベースは所望によりネットリスト（合成ライブラリ付きまたは合成ライブラリなし）またはデータセットであってもよい。このように、ハードウェア生成命令は、前述した方法および技術を実装するプロセッサおよび／または処理要素を、製造分野における知識を有する当業者に知られた技術にしたがって生成または作成されるために実行されてもよい。追加的に、このようなハードウェア生成命令は、任意の適切なコンピュータ読み取り可能な媒体（これは要素６０のようなメモリサブシステム内に存在してもよく、あるいは他のコンピュータ読み取り可能な媒体に存在していてもよい）に記憶されてもよい。

前述したコンピュータ読み取り可能な記憶媒体は、幾つかの実施形態において、プログラムにより読み出される命令を記憶するために用いられることができ、また、処理要素１００Ａおよび／または１００Ｂを有するハードウェアを製造するために直接的または間接的に用いられることができる。例えば、命令は、ＶｅｒｉｌｏｇまたはＶＨＤＬといった高水準設計言語（ＨＤＬ）によるビヘイビア・レベルまたはレジスタ転送レベル（ＲＴＬ）の記述レベルでハードウェア機能を記述した１以上のデータ構造の概要を示すものであってもよい。記述内容は合成ツールにより読み取られてもよく、合成ツールは記述を合成してネットリストを生成するものであってもよい。ネットリストは、ゲート群（例えば合成ライブラリに定義された）を含んでもよく、処理要素１００、メモリ初期化ユニットおよび／またはメモリ初期化回路の機能性を表現するものである。このようなネットリストを配置・経路処理して、マスクに対して適用すべき幾何学形状を記述したデータセットが生成されてもよい。このようなマスクはハードウェア実施形態に対応する半導体回路または回路系を製造するために様々な半導体製造工程において利用されてもよい。代替的に、データベースは所望によりネットリスト（合成ライブラリ付きまたは合成ライブラリなし）またはデータセットであってもよい。一実施形態はこのようにして（非一時的な）コンピュータシステムで実行可能なプログラムによって利用可能なデータ構造であって該データ構造によって記述された回路を有する集積回路を製造するプロセスの一部を行うためのデータ構造を有する（非一時的な）コンピュータ読取り可能な記憶媒体であって、前記データ構造に記述された前記回路が、メモリデバイスのメモリ領域の初期化をコンピュータデバイスの第１処理要素ではなく該コンピュータデバイスの第２処理要素に行わせるように構成されたメモリ初期化ユニットを有し、かつ、前記初期化を前記第２処理要素に行わせる処理が、前記メモリ領域が初期化されるべきとの指示に応答して行われることを特徴とする非一時的なコンピュータ読取り可能な記憶媒体、となる。

【００５９】

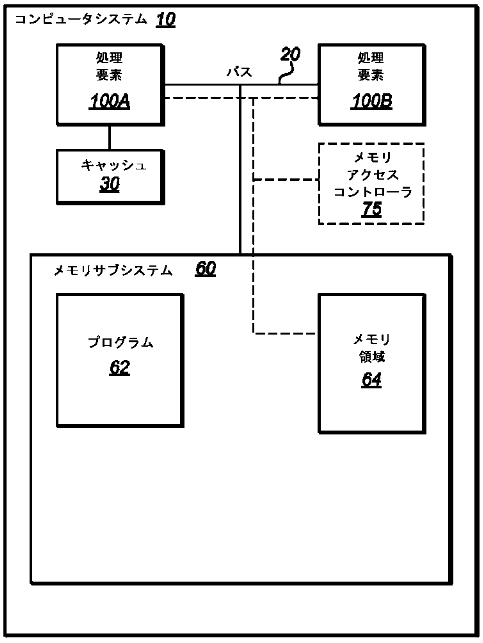
以上のとおり具体的な実施形態について説明したが、これらの実施形態は、本開示の範囲を限定することを意図するものではなく、ある特徴に関して単一の実施形態のみが記載されている場合でも同様である。開示に含まれる特徴の例は説明のためのものであって特に断らないかぎり限定のためのものではない。前述した説明はこの開示の利益を有する当

業者に自明と考えられるように代替、変更、均等物を包含することを意図したものである。

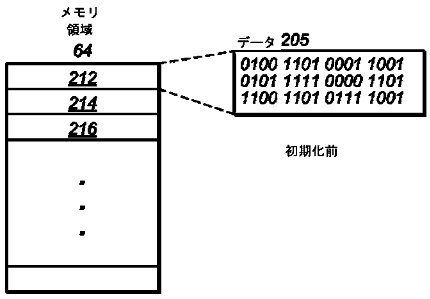
【００６０】

　本開示の範囲は、ここに検討した問題点のいずれかまたはそのすべてを軽減するかどうかにかかわらず、ここに（明示的もしくは暗示的に）開示したあらゆる特徴、それら特徴の組合せおよびそれらの任意の一般化を含むものである。したがって、このような特徴の組合せを対象としてこの出願（またはこの出願が優先権を主張するところの出願）の審査の間に新たな請求項が構成されてもよい。特に、添付の請求項については、従属項に係る特徴は独立項の特徴と組み合わせることができ、各従属項からの特徴は任意の適切な態様にて組み合わせられてもよく、添付の請求項において具体的に適示された具体的な組合せに限定されるものではない。

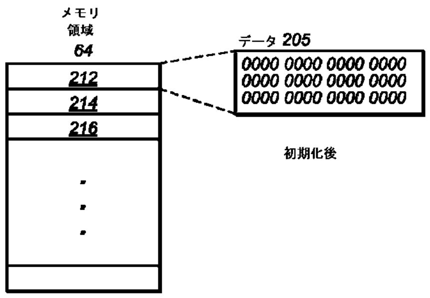
【図１】



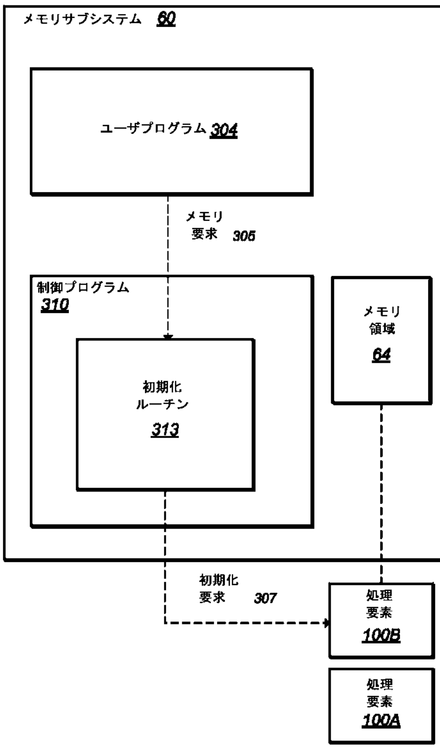
【図２Ａ】



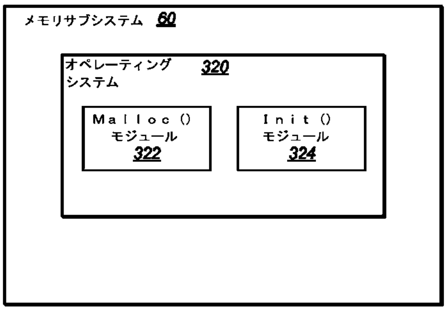
【図２Ｂ】



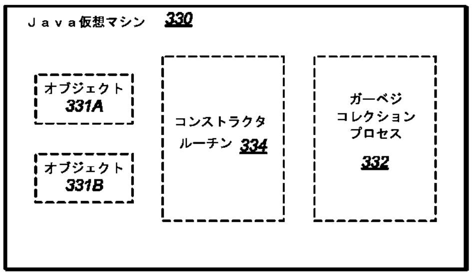
【図３Ａ】



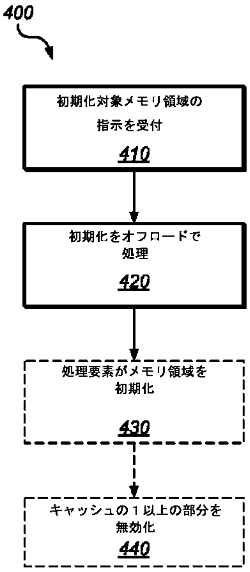
【図３Ｂ】



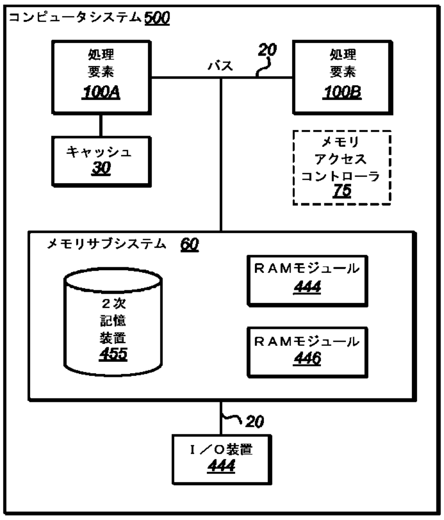
【図３Ｃ】



【図４】



【図５】



【手続補正書】
【提出日】平成25年4月2日(2013.4.2)
【手続補正１】
【補正対象書類名】特許請求の範囲
【補正対象項目名】全文
【補正方法】変更
【補正の内容】
【特許請求の範囲】
【請求項１】
コンピュータデバイスにおける初期化されるメモリ領域を示す指示に応答して、該メモリ領域の初期化を該コンピュータデバイスの第２処理要素に処理させるための、少なくとも該コンピュータデバイスの第１処理要素によって実行可能なプログラム命令を記憶する、
非一時的なコンピュータ読み取り可能な媒体。
【請求項２】
前記メモリ領域の指示が、第１プログラムから、前記第１処理要素によって実行される制御プログラムにより受付けられる、
請求項１の非一時的なコンピュータ読み取り可能な媒体。
【請求項３】
前記制御プログラムが、前記第１プログラムを実行している、
請求項２の非一時的なコンピュータ読み取り可能な媒体。
【請求項４】
前記指示は、前記制御プログラムで処理可能な１以上のデータオブジェクトに対応したメモリの１以上のメモリ領域を指定しており、
前記１以上のメモリ領域のすべての内容を埋める処理がさらに含まれている、
請求項２の非一時的なコンピュータ読み取り可能な媒体。
【請求項５】
前記制御プログラムは、ガーベジコレクションプロセスの一部として、初期化される複数のメモリ領域を示す複数の指示を生成する処理と、前記複数のメモリ領域に対する初期化を前記第２処理要素に行わせる処理とをさらに含み、
前記初期化には、前記複数のメモリ領域のすべての内容を、プログラミング言語の言語仕様にしたがって指定されたデフォルトの内容で埋める処理が含まれる、
請求項４の非一時的なコンピュータ読み取り可能な媒体。
【請求項６】
前記制御プログラムは、前記非一時的なコンピュータ読み取り可能な媒体に記憶された１以上のライブラリファイルを有し、かつ、前記指示を受付ける前記制御プログラムは、アプリケーション・プログラミング・インタフェース（ＡＰＩ）を介して前記指示を受付ける前記制御プログラムを有する、
請求項２の非一時的なコンピュータ読み取り可能な媒体。
【請求項７】
第１処理要素と、第２処理要素と、メモリ領域を含むメモリとを備えたコンピュータデバイスにおける初期化される該メモリ領域を示す指示に応答して、前記第１処理要素で実行されている第１プログラムが、該メモリ領域の初期化を前記第２処理要素に処理させる、
方法。
【請求項８】
前記第１処理要素が前記メモリ領域へ直接アクセスすることなく、前記第２処理要素が、ダイレクトメモリアクセス（ＤＭＡ）を用いて前記メモリ領域を初期化する処理、をさらに含む、
請求項７の方法。

【請求項９】
　前記指示を生成する前記第１プログラム内においてガーベジコレクションプロセスをさらに含む、
　請求項７の方法。
【請求項１０】
　前記第１プログラムは制御プログラムであって、
　前記第２処理要素が、前記制御プログラムの１以上のヒューリスティックな規則にしたがって前記メモリ領域を初期化する処理、をさらに含む、
　請求項７の方法。
【請求項１１】
　前記コンピュータデバイスが、前記メモリ領域が初期化されるのに応答して、前記コンピュータデバイスのデータキャッシュの１以上の部分であってメモリ領域初期化前の前記メモリ領域の内容に対応する該部分を無効化する処理、をさらに含む、
　請求項７の方法。
【請求項１２】
　メインメモリを有するメモリサブシステムと、２次記憶装置と、少なくとも第１および第２処理要素と、を有するコンピュータシステムであって、
　前記２次記憶装置には、初期化される前記メインメモリのメモリ領域を示す指示に応答して、該メモリ領域の初期化を前記第２処理要素に処理させるように前記コンピュータシステムを動作させるための、前記第１処理要素によって実行可能なプログラム命令、が記憶されている、
　コンピュータシステム。
【請求項１３】
　前記第１処理要素と、前記第２処理要素とは異機種である、
　請求項１２のコンピュータシステム。
【請求項１４】
　前記第１処理要素と関連付けられたキャッシュをさらに有し、
　前記キャッシュは、前記第１処理要素による前記メインメモリへのアクセスに応答して該メインメモリの内容を記憶するように構成され、
　前記メモリ領域の初期化が行われたとき、前記メモリ領域の初期化後の内容が前記キャッシュに記憶されない、
　請求項１２のコンピュータシステム。
【請求項１５】
　前記第２処理要素に対して前記メインメモリへの直接アクセスを許可するように構成されたメモリアクセスコントローラをさらに有し、
　メモリ領域の初期化には、前記第２処理要素が前記メモリアクセスコントローラを用いて前記メモリ領域にアクセスすることが含まれ、前記第１処理要素が前記メモリ領域にアクセスすることが含まれない、
　請求項１２のコンピュータシステム。

【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2011/046412

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F9/50
ADD. G06F12/02

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data, COMPENDEX, INSPEC, IBM-TDB

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	"NVIDIA CUDA Compute Unified Device Architecture, programming guide", , 27 November 2007 (2007-11-27), pages I-XIII,1-128, XP008139068, Retrieved from the Internet: URL:http://developer.download.nvidia.com/compute/cuda/1_1/NVIDIA_CUDA_Programming_Guide_1.1.pdf [retrieved on 2007-11-29] in particular appendices D.5 and E.8; the whole document -----	1-21

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
12 October 2011	25/10/2011

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Steinmetz, Christof

Form PCT/ISA/210 (second sheet) (April 2005)

フロントページの続き

(81)指定国 AP(BW,GH,GM,KE,LR,LS,MW,MZ,NA,SD,SL,SZ,TZ,UG,ZM,ZW),EA(AM,AZ,BY,KG,KZ,MD,RU,TJ,TM),EP(AL,AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,HR,HU,IE,IS,IT,LT,LU,LV,MC,MK,MT,NL,NO,PL,PT,RO,RS,SE,SI,SK,SM,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN,GQ,GW,ML,MR,NE,SN,TD,TG),AE,AG,AL,AM,AO,AT,AU,AZ,BA,BB,BG,BH,BR,BW,BY,BZ,CA,CH,CL,CN,CO,CR,CU,CZ,DE,DK,DM,DO,DZ,EC,EE,EG,ES,FI,GB,GD,GE,GH,GM,GT,HN,HR,HU,ID,IL,IN,IS,JP,KE,KG,KM,KN,KP,KR,KZ,LA,LC,LK,LR,LS,LT,LU,LY,MA,MD,ME,MG,MK,MN,MW,MX,MY,MZ,NA,NG,NI,NO,NZ,OM,PE,PG,PH,PL,PT,QA,RO,RS,RU,SC,SD,SE,SG,SK,SL,SM,ST,SV,SY,TH,TJ,TM,TN,TR,TT,TZ,UA,UG,US,UZ,VC,VN,ZA,ZM

(74)代理人 100162156
弁理士 村雨 圭介
(72)発明者 エリック アール. カスポール
アメリカ合衆国 ９４０２５ カリフォルニア州、メンロー パーク、ウェイブレイ ストリート ３１１ ＃2
(72)発明者 ローラン モリケッティ
アメリカ合衆国 ９５１１２ カリフォルニア州、サンノゼ、サウス １４番 ストリート ３７７

Ｆターム(参考) 5B005 KK14 MM01 PP21
5B060 AB30