(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7510932号
(P7510932)

(45)発行日　令和6年7月4日(2024.7.4)

(24)登録日　令和6年6月26日(2024.6.26)

(51)国際特許分類		FI		
G06F 17/10 (2006.01)		G06F	17/10	Z
G06F 9/50 (2006.01)		G06F	9/50	150E
G06F 15/80 (2006.01)		G06F	15/80	
G06F 9/30 (2018.01)		G06F	9/30	350D
G06F 16/245 (2019.01)		G06F	16/245	

請求項の数　14　（全21頁）

(21)出願番号　特願2021-532113(P2021-532113)
(86)(22)出願日　令和1年10月16日(2019.10.16)
(65)公表番号　特表2022-511528(P2022-511528A)
(43)公表日　令和4年1月31日(2022.1.31)
(86)国際出願番号　PCT/US2019/056496
(87)国際公開番号　WO2020/117377
(87)国際公開日　令和2年6月11日(2020.6.11)
審査請求日　令和4年9月14日(2022.9.14)
(31)優先権主張番号　16/212,134
(32)優先日　平成30年12月6日(2018.12.6)
(33)優先権主張国・地域又は機関
米国(US)

(73)特許権者　591025439
ザイリンクス インコーポレイテッド
ＸＩＬＩＮＸ ＩＮＣＯＲＰＯＲＡＴＥＤ
アメリカ合衆国 カリフォルニア州 ９５１２４－３４００ サン ホセ ロジック ドライブ ２１００
(74)代理人　110001195
弁理士法人深見特許事務所
(72)発明者　ベルマ，ヘア・ケイ
アメリカ合衆国、９５１２４ カリフォルニア州、サン・ノゼ、ロジック・ドライブ、２１００
(72)発明者　ティアン，ビング
アメリカ合衆国、９５１２４ カリフォルニア州、サン・ノゼ、ロジック・ドラ
最終頁に続く

(54)【発明の名称】　集積回路、およびデータクエリを加速させる方法

(57)【特許請求の範囲】

【請求項１】

集積回路であって、

並列に配置された複数の処理ユニット（ＰＵ（ｉ））を備え、前記複数のＰＵ（ｉ）の各々は、予め定められたクエリ言語を使用する命令セット群（Ｇ）における予め定められた命令セット（Ｓ（ｉ））がロードされた命令レジスタを備え、前記予め定められた命令セット（Ｓ（ｉ））に従ってデータストリームの選択されたデータブロックを処理するように構成されたハードウェア回路で形成され、前記複数のＰＵ（ｉ）の各々は、前記ＰＵ（ｉ）によって処理される前記データストリームの前記選択された部分に対応する中間出力結果を生成し、前記集積回路はさらに、

前記複数のＰＵ（ｉ）の各々から前記中間出力結果の各々を受信して、集約結果を生成するように結合された連結回路を備え、

前記Ｓ（ｉ）の各々は、ユーザ定義のクエリから抽出された命令の関数および少なくとも１つのパラメータの関数を備え、

前記複数のＰＵ（ｉ）の各々は、（ｉ）変数レジスタに格納された前記データストリームの一部を備える第１のオペランドと、（ｉｉ）定数レジスタに格納された前記抽出されたパラメータのうちの１つを備える第２のオペランドとを使用して実行される演算を実行することによって、前記対応するＳ（ｉ）を実行するように構成される、集積回路。

【請求項２】

前記予め定められた命令セット（Ｓ（ｉ））は、ＳＱＬ命令を備える、請求項１に記載

の集積回路。
【請求項３】
前記予め定められたクエリ言語は、ＳＱＬを備える、請求項１に記載の集積回路。
【請求項４】
前記連結回路は、前記ユーザ定義のクエリに関連付けられた予め定められた関数に従って前記集約結果を生成するように構成される、請求項１に記載の集積回路。
【請求項５】
前記複数の処理ユニットの各々は、ＡＳＩＣにおける固定ハードウェア回路として実現される、請求項１に記載の集積回路。
【請求項６】
前記複数の処理ユニットの各々は、ＦＰＧＡのプログラム可能なファブリックにおける再構成可能なハードウェアとして実現される、請求項１に記載の集積回路。
【請求項７】
前記データストリームを受信するように結合されたスケジューラ回路をさらに備え、前記スケジューラ回路は、前記データブロックの各々を前記複数のＰＵ（ｉ）のうちの１つに選択的に向けるように構成される、請求項１に記載の集積回路。
【請求項８】
前記スケジューラ回路は、ラウンドロビンスケジューラを備える、請求項７に記載の集積回路。
【請求項９】
前記複数のＰＵ（ｉ）の各々は、抽出された命令に基づき、かつ（ｉ）前記第１のオペランドと（ｉｉ）前記第２のオペランドとを使用して、前記演算を実行することによって前記対応するＳ（ｉ）を実行するように適合された演算論理装置ＡＬＵを備える、請求項１に記載の集積回路。
【請求項１０】
前記複数の処理ユニットの各々は、
前記実行された演算の結果を保持するように構成された一時レジスタと、
前記定数レジスタおよび前記変数レジスタから入力を受信するように構成された第１のマルチプレクサと、
前記定数レジスタ、前記変数レジスタおよび前記一時レジスタから入力を受信するように構成された第２のマルチプレクサとをさらに備える、請求項１に記載の集積回路。
【請求項１１】
データクエリを実行するようにファブリックを構成する方法であって、
ユーザからデータクエリを受信するステップと、
前記データクエリを予め定められたクエリ言語のコマンドに変換するステップと、
前記コマンドから、複数の並列処理ユニットＰＵ（ｉ）に格納されるパラメータを抽出するステップと、
前記コマンドから命令を抽出して、前記ＰＵ（ｉ）によって実行される命令セット群Ｇを形成するステップとを備え、前記命令セット群Ｇは、複数の命令セットＳ（ｉ）を備え、前記方法はさらに、
前記複数のＰＵ（ｉ）に前記抽出されたパラメータおよび前記抽出された命令をロードするステップを備え、
前記ＰＵ（ｉ）の各々は、データストリームの予め定められたデータブロック（ｉ）を、その対応するパラメータおよび命令セットＳ（ｉ）を用いて並列に処理するように構成される、方法。
【請求項１２】
前記予め定められたクエリ言語は、ＳＱＬを備える、請求項１１に記載の方法。
【請求項１３】
前記ＰＵ（ｉ）の各々は、同一のＳ（ｉ）で構成される、請求項１１に記載の方法。
【請求項１４】

前記抽出されたパラメータおよび前記抽出された命令をロードする前に全ての前記ＰＵ（ｉ）をクリアするステップをさらに備える、請求項１１に記載の方法。

【発明の詳細な説明】

【技術分野】

【０００１】

技術分野

さまざまな実施形態は、一般に、ハードウェア加速（アクセラレーション）に使用される構成可能なファブリックを有する集積回路に関する。

【背景技術】

【０００２】

背景

ビッグデータは、従来のデータ処理アプリケーションソフトウェアでは処理できないほどに大きくかつ複雑なデータセットの研究および応用を指して用いられる。ビッグデータは、課題を生じさせ、それらの課題は、データ取り込み、データ記憶、データ分析、データ更新、情報プライバシ、およびデータ照会を含むが、これらに限定されるものではない。

【０００３】

データクエリとは、一般に、データベーステーブルまたはテーブルの組み合わせからのデータまたは情報に対する要求のことをいう。このデータは、構造化照会言語（ＳＱＬ）によって返される結果として、または、画像、グラフまたは複雑な結果（たとえば、データマイニングツールからのトレンド分析）として生成されてもよい。ＳＱＬは、データベース内のデータを格納したり操作したり検索したりするための標準的な言語である。

【発明の概要】

【課題を解決するための手段】

【０００４】

概要

ハードウェア加速に関連する集積回路および方法は、データストリームを処理して結果を集約してクエリに応答するようにカスタム適合された独立したプログラム可能な並列処理ユニット（ＰＵ）を含む。例示的な例においては、データベースからのデータストリームは、データブロックに分割されて、対応するＰＵに割り当てられてもよい。各データブロックは、予め定められた命令セットに従って結果を生成するようにＰＵのうちの１つによって処理されてもよい。連結ユニットは、各データブロックの結果を併合および連結して、クエリについての出力結果を生成してもよい。いくつかの実施形態においては、非常に大規模なデータベースＳＱＬクエリは、たとえば、固定ＡＳＩＣまたは再構成可能なＦＰＧＡハードウェア回路内に実装されたハードウェアＰＵ／連結エンジンによって加速されてもよい。

【０００５】

いくつかの実施形態においては、フィールドプログラマブルゲートアレイ（ＦＰＧＡ）は、ファブリック内に形成された電気的に再構成可能なプログラム可能なハードウェア論理回路を提供してもよい。ＦＰＧＡファブリックは、電気構成信号に応答して、クエリに特有のハードウェアリソース配置を提供するように再構成可能であってもよい。本明細書に教示されているように、再構成されたファブリックは、単一のＦＰＧＡが広範にわたるクエリを柔軟に加速させることができるように各クエリについて独自にハードウェア処理回路をカスタマイズすることによって大規模データベースのクエリを効率的に処理するためのハードウェアアクセラレータを生成するように配置されてもよい。

【０００６】

いくつかの実施形態においては、特定用途向け集積回路（ＡＳＩＣ）は、ハードウェア論理回路（たとえば、デジタル、アナログ）の固定配置を提供するように製造されてもよい。ＡＳＩＣは、１つまたは複数のＡＳＩＣが単独でまたは組み合わさって１つまたは複数の予め定められたクエリを加速させることができるように、１つまたは複数の予め定められたクエリのためのカスタマイズされたハードウェア処理回路を用いて大規模データベ

ースのクエリを効率的に処理するためのハードウェアアクセラレータを提供してもよい。
【０００７】
さまざまな実施形態は、１つまたは複数の利点を実現し得る。たとえば、いくつかの実施形態は、たとえば大規模データベース上でクエリを実行する際に、実質的にデータクエリ応答時間を減少させ、および／または、処理スループットを増加させ得る。データベースクエリのハードウェア加速は、各々の特定のクエリについて演算およびパラメータに従って構成されたカスタマイズ可能な（たとえば、固定されていない）ハードウェアブロックで実現されてもよい。いくつかの実装例は、大規模データベースの顧客供給のクエリを実行するように、プログラム可能なファブリックデバイスにおいてプログラムされてもよい。さまざまな実施形態は、高速データストリーム処理を、固定中央処理装置（ＣＰＵ）から、カスタムプログラムされたハードウェア処理チャネルにオフロードしてもよく、このカスタムプログラムされたハードウェア処理チャネルでは、予め定められた命令セットに従ってデータストリームの複数のデータブロックを個々に処理することができる。したがって、相当な計算効率が実現され得て、その結果、大規模データベースのクエリの処理時間が劇的に減少し得る。
【０００８】
たとえば、いくつかの実施形態は、たとえば再構成可能なファブリックデバイス（たとえば、ＦＰＧＡ）を活用して最小限のハードウェアリソースで非常に効率的な並列処理を実行することによって、作製コストを削減して、分散非同期通信を減少させてもよい。たとえば、いくつかの実施形態は、カーネルレベル性能を向上させ、および／または、いくつかのクエリを処理する際にたとえばＣＰＵの１０～２５倍の性能向上を提供してもよい。さまざまな実装例においては、フィールドプログラマブル機能を有するＦＰＧＡは、動的クエリ要件を満たすように１回または複数回エンドユーザによって柔軟にカスタマイズされてもよい。
【０００９】
いくつかの実施形態においては、有利なことに、ＡＳＩＣは、専用の（たとえば、固定された）ハードウェア回路を使用して１つまたは複数の予め定められたクエリ構造にハードウェア加速機能を提供し得る。ＡＳＩＣを組み入れるいくつかの実施形態は、たとえばコンポーネントコスト、ボリュームおよび／または電力要件を減少させたクエリハードウェア加速を提供し得る。
【００１０】
１つの例示的な局面においては、集積回路は、複数の処理ユニット（ＰＵ（ｉ））を含む。上記ＰＵ（ｉ）は、並列に配置される。上記ＰＵ（ｉ）の各々は、予め定められたクエリ言語を使用する命令セット群（Ｇ）における予め定められた命令セット（Ｓ（ｉ））に従ってデータストリームの選択されたデータブロックを処理するように構成されたハードウェア回路で形成される。各ＰＵ（ｉ）は、上記ＰＵ（ｉ）によって処理される上記データストリームの上記選択された部分に対応する中間出力結果を生成する。連結回路は、上記複数のＰＵ（ｉ）の各々から上記中間出力結果の各々を受信して、集約結果を生成するように結合される。上記Ｓ（ｉ）の各々は、ユーザ定義のクエリから抽出された命令の関数を含む。
【００１１】
いくつかの実施形態においては、上記予め定められた命令セット（Ｓ（ｉ））は、ＳＱＬ命令を含んでいてもよい。いくつかの実施形態においては、上記予め定められたクエリ言語は、ＳＱＬを含んでいてもよい。いくつかの実施形態においては、上記連結ユニットは、上記ユーザ定義のクエリに関連付けられた予め定められた関数に従って上記集約結果を生成するように構成されてもよい。いくつかの実施形態においては、上記複数の処理ユニットの各々は、ＡＳＩＣにおける固定ハードウェア回路として実現されてもよい。いくつかの実施形態においては、上記複数の処理ユニットの各々は、ＦＰＧＡのプログラム可能なファブリックにおける再構成可能なハードウェアとして実現されてもよい。
【００１２】

いくつかの実施形態においては、上記集積回路は、上記データストリームを受信するように結合されたスケジューラ回路も含んでいてもよい。上記スケジューラ回路は、上記データブロックの各々を上記複数のＰＵ（ｉ）のうちの１つに選択的に向けるように構成されてもよい。いくつかの実施形態においては、上記スケジューラ回路は、ラウンドロビンスケジューラを含んでいてもよい。いくつかの実施形態においては、上記Ｓ（ｉ）の各々は、上記ユーザ定義のクエリから抽出された少なくとも１つのパラメータの関数も含んでいてもよい。いくつかの実施形態においては、上記複数のＰＵ（ｉ）の各々は、抽出された命令に基づいて演算を実行することによって上記対応するＳ（ｉ）を実行するように適合された演算論理装置ＡＬＵを含んでいてもよい。上記演算は、（ｉ）変数レジスタに格納された上記データストリームの一部を備える第１のオペランドと、（ｉｉ）定数レジスタに格納された上記抽出されたパラメータのうちの１つを備える第２のオペランドとを使用して実行されてもよい。

【００１３】

いくつかの実施形態においては、上記複数の処理ユニットの各々は、上記実行された演算の結果を保持するように構成された一時レジスタと、上記定数レジスタおよび上記変数レジスタから入力を受信するように構成された第１のマルチプレクサと、上記定数レジスタ、上記変数レジスタおよび上記一時レジスタから入力を受信するように構成された第２のマルチプレクサとを含んでいてもよい。

【００１４】

いくつかの実施形態においては、上記複数のＰＵ（ｉ）の各々は、上記ＡＬＵにおいて実行される上記Ｓ（ｉ）を格納するように構成された命令レジスタも含んでいてもよい。いくつかの実施形態においては、上記ＰＵ（ｉ）の各々は、同一のＳ（ｉ）で構成されてもよい。いくつかの実施形態においては、上記ＰＵ（ｉ）の各々は、異なるＳ（ｉ）で構成されてもよい。上記予め定められたクエリ言語の命令セット群Ｇにおける上記複数の命令は、スキャン命令と集約命令とを含んでいてもよい。スキャン命令は、ＡＮＤ論理演算とＯＲ論理演算とを含んでいてもよい。

【００１５】

別の例示的な局面においては、データクエリを実行するようにファブリックを構成する方法は、ユーザからデータクエリを受信するステップと、上記データクエリを予め定められたクエリ言語コマンドに変換するステップとを含む。また、上記方法は、上記コマンドから、複数の並列処理ユニットＰＵ（ｉ）に格納されるパラメータを抽出するステップと、上記コマンドから命令を抽出して、上記ＰＵ（ｉ）によって実行される命令セット群Ｇを形成するステップとを含む。上記命令セット群Ｇは、複数の命令セットＳ（ｉ）を含む。上記方法は、上記複数のＰＵ（ｉ）に上記抽出されたパラメータおよび上記抽出された命令をロードするステップも含む。上記ＰＵ（ｉ）の各々は、データストリームの予め定められたデータブロック（ｉ）を、その対応するパラメータおよび命令セットＳ（ｉ）を用いて並列に処理するように構成される。

【００１６】

いくつかの実施形態においては、上記予め定められたクエリ言語は、ＳＱＬを含んでいてもよい。いくつかの実施形態においては、上記ＰＵ（ｉ）の各々は、同一のＳ（ｉ）で構成されてもよい。上記方法は、上記抽出されたパラメータおよび上記抽出された命令をロードする前に全ての上記ＰＵ（ｉ）をクリアするステップも含んでいてもよい。

【００１７】

さまざまな実施形態の詳細が添付の図面および以下の説明に記載されている。他の特徴および利点は、説明および図面ならびに特許請求の範囲から明らかであろう。

【図面の簡単な説明】

【００１８】

【図１】開示されている回路およびプロセスが実現され得る例示的なプログラム可能な集積回路（ＩＣ）を示す図である。

【図２（Ａ）】例示的なハードウェア加速処理エンジンを有するホストコンピューティン

グシステムを示す図である。
【図２（Ｂ）】例示的なデータクエリを実行する図２（Ａ）のハードウェア加速処理システムのブロック図である。
【図３（Ａ）】図２（Ｂ）のハードウェア加速処理エンジンに含まれる例示的な処理ユニットを示す図である。
【図３（Ｂ）】図２（Ｂ）の例示的な処理ユニットを動作させるためのいくつかの例示的な命令を示す図である。
【図４】データクエリジョブを実行するように構成されたハードウェア加速処理エンジンのための例示的な設計時ファブリック再構成方法のフローチャートである。
【図５】ハードウェア加速処理エンジンの例示的な構造を示す図である。
【図６】プログラム可能なハードウェア加速処理エンジンを用いてデータクエリを実行するための例示的な実行時方法のフローチャートである。
【発明を実施するための形態】
【００１９】
さまざまな図面における同様の参照記号は、同様の要素を示す。
例示的な実施形態の詳細な説明
理解を助けるために、本文献は以下のように構成されている。第一に、図１を参照して、開示されているハードウェア加速処理エンジンおよびプロセスが実現され得る例示的なプログラム可能な集積回路（ＩＣ）が簡単に紹介されている。第二に、図２（Ａ）～図３を参照して、構成されたファブリックの構造およびファブリックを構成する方法を説明する例示的な実施形態について論じられている。次いで、図４（Ａ）～図５を参照して、処理ユニットの例示的な構造およびハードウェア加速処理エンジンの例示的な構造が提示されている。最後に、図６を参照して、実行時にデータクエリを実行するために使用される例示的な方法が提示されている。
【００２０】
図１は、開示されている回路およびプロセスが実現され得る例示的なプログラム可能な集積回路（ＩＣ）を示す図である。プログラム可能なＩＣ１００は、ＦＰＧＡ論理を含む。プログラム可能なＩＣ１００は、さまざまなプログラム可能なリソースで実現されてもよく、システムオンチップ（ＳＯＣ）と称されてもよい。ＦＰＧＡ論理のさまざまな例としては、アレイ状のいくつかのさまざまなタイプのプログラム可能な論理ブロックを挙げることができる。
【００２１】
たとえば、図１は、プログラム可能なＩＣ１００を示し、プログラム可能なＩＣ１００は、マルチギガビットトランシーバ（multi-gigabit transceiver：ＭＧＴ）１０１と、構成可能な論理ブロック（configurable logic block：ＣＬＢ）１０２と、ランダムアクセスメモリのブロック（blocks of random access memory：ＢＲＡＭ）１０３と、入出力ブロック（input/output block：ＩＯＢ）１０４と、構成およびクロック論理（configuration and clocking logic：ＣＯＮＦＩＧ／ＣＬＯＣＫＳ）１０５と、デジタル信号処理ブロック（digital signal processing block：ＤＳＰ）１０６と、専用入出力ブロック（Ｉ／Ｏ）１０７（たとえば、クロックポート）と、他のプログラム可能な論理１０８（たとえば、デジタルクロックマネージャ、アナログ－デジタル変換器、システム監視論理）とを含む多数のさまざまなプログラム可能なタイルを含む。プログラム可能なＩＣ１００は、専用のプロセッサブロック（processor block：ＰＲＯＣ）１１０を含む。プログラム可能なＩＣ１００は、内部再構成ポートおよび外部再構成ポート（図示せず）を含み得る。
【００２２】
さまざまな例においては、シリアライザ／デシリアライザがＭＧＴ１０１を用いて実装され得る。ＭＧＴ１０１は、さまざまなデータシリアライザおよびデシリアライザを含み得る。データシリアライザはさまざまなマルチプレクサ実装例を含み得る。データデシリアライザは、さまざまなデマルチプレクサ実装例を含み得る。

【００２３】
ＦＰＧＡ論理のいくつかの例においては、各々のプログラム可能なタイルは、各々の隣接するタイルにおける対応する相互接続要素への／からの標準化された相互接続１２４を有するプログラム可能な相互接続要素（interconnect element：ＩＮＴ）１１１を含む。したがって、複数のプログラム可能な相互接続要素は、まとめて、図示されるＦＰＧＡ論理のためのプログラム可能な相互接続構造を実現する。プログラム可能な相互接続要素ＩＮＴ１１１は、図１に含まれる例によって示されるように、同じタイル内におけるプログラム可能な論理要素への／からの内部接続１２０を含む。プログラム可能な相互接続要素ＩＮＴ１１１は、図１に含まれる例によって示されるように、同じタイル内にプログラム可能な相互接続要素ＩＮＴ１１１への／からのＩＮＴ間接続１２２を含む。

【００２４】
たとえば、ＣＬＢ１０２は、ユーザ論理を実装するようにプログラムされ得る構成可能な論理要素（configurable logic element：ＣＬＥ）１１２と、単一のプログラム可能な相互接続要素ＩＮＴ１１１とを含み得る。ＢＲＡＭ１０３は、ＢＲＡＭ論理要素（BRAM logic element：ＢＲＬ）１１３および１つまたは複数のプログラム可能な相互接続要素を含み得る。いくつかの例においては、１枚のタイルに含まれる相互接続要素の数は、当該タイルの高さに依存し得る。図示される実装例においては、ＢＲＡＭタイルは、５つのＣＬＢと同じ高さを有するが、他の数（たとえば、４つ）が用いられてもよい。ＤＳＰタイル１０６は、ＤＳＰ論理要素（DSP logic element：ＤＳＰＬ）１１４および１つまたは複数のプログラム可能な相互接続要素を含み得る。ＩＯＢ１０４は、たとえば、入出力論理要素（input/output logic element：ＩＯＬ）１１５の２つのインスタンスと、プログラム可能な相互接続要素ＩＮＴ１１１の１つのインスタンスとを含み得る。たとえば、Ｉ／Ｏ論理要素１１５に接続される実際のＩ／Ｏボンドパッドは、図示されるさまざまな論理ブロックの上に積層される金属を用いて製造されてもよく、入出力論理要素１１５の面積に制限されなくてもよい。

【００２５】
図示される実装例においては、ダイの中心付近の（図１に網掛けして示される）列状区域が、構成、クロック、および他の制御論理に用いられる。列から延びる水平区域１０９は、プログラム可能なＩＣ１００の幅にわたってクロックおよび構成信号を分配する。「列状」および「水平」区域と言及する場合、図面を縦向きで見ることを基準としていることに留意されたい。

【００２６】
図１に示されるアーキテクチャを利用するいくつかのプログラム可能なＩＣは、プログラム可能なＩＣの大部分を構成する規則的な柱状構造を乱す追加の論理ブロックを含み得る。追加の論理ブロックは、プログラム可能なブロックおよび／または専用論理であってもよい。たとえば、図１に示すプロセッサブロックＰＲＯＣ１１０は、ＣＬＢ１０２およびＢＲＡＭ１０３のいくつかの列にわたっている。

【００２７】
図１は、例示的なプログラム可能なＩＣアーキテクチャを示す。列内の論理ブロックの数、列の相対的幅、列の数および順序、列に含まれる論理ブロックの種類、論理ブロックの相対的サイズ、ならびに相互接続／論理実装は純粋に例として提供されているに過ぎない。たとえば、実際のプログラム可能なＩＣにおいては、ＣＬＢ１０２のうち２つ以上の隣接する列が、ユーザ論理の効率的な実装を容易にするために、ＣＬＢ１０２が現れる場所に含められてもよい。

【００２８】
さまざまな分野におけるコンピュータアプリケーションの継続的な拡大に伴って、さまざまなアプリケーションシナリオは、サーバのデータ処理能力にますます要求を突きつけるようになっている。いくつかの特定のシナリオにおいては、サーバがリソースの割り当てのバランスを取ることは非常に難しいであろう。要求される処理速度を実現するためには、より強力な計算能力が必要とされる。データ処理速度が極めて重要であるいくつかの

状況においては、中央処理装置（ＣＰＵ）の作業の一部をハードウェアアクセラレータによって共有して特定のタイプの計算を引き受けるためにＦＰＧＡが使用され得る。

【００２９】

図２（Ａ）は、例示的なハードウェア加速処理エンジンを有するホストコンピューティングシステムを示す図である。ホストコンピューティングシステム２００は、ハードウェア加速処理システム２０５を含む。ハードウェア加速処理システム２０５は、複数の相互接続された回路サブシステムを含み、これらの相互接続された回路サブシステムのうちの１つは、フィールドプログラマブルゲートアレイ（ＦＰＧＡ）２１５に電気的に結合された中央処理装置（ＣＰＵ）２１０である。ＦＰＧＡは、半導体集積回路基板を含み得る。ＦＰＧＡ２１５は、中央処理装置（ＣＰＵ）の作業の一部を、特定の複雑なデータベースクエリ命令を処理するように柔軟に再構成可能なハードウェア加速処理エンジンによって共有するために使用されてもよい。ＦＰＧＡ２１５は、データクエリ応答速度を加速させるための１つまたは複数のハードウェア加速処理エンジン２２５を提供する。さまざまな実装例においては、計算負荷は、ＣＰＵ２１０から、大規模データベースクエリに応答して結果を効率的に生成するためのクエリに特有のハードウェア回路を提供するハードウェア加速処理エンジン２２５に選択的にオフロードされてもよい。

【００３０】

ハードウェア加速処理エンジン２２５は、データ処理を実行するように並列に設置された一組の処理ユニット（ＰＵ）２３０と、処理された結果を連結するように構成された連結ユニット２３５とを含む。処理ユニット２３０の各々は、データストリームに対してたとえば予め定められたフィルタおよび／または集約演算を実行するように独立してプログラムされてもよい。さまざまな例においては、処理ユニットＰＵ２３０の各々に入力されるデータストリームは、演算および／またはレコードデータを含み得る。これらの演算は、たとえばユーザが所望のクエリをユーザインターフェイスを介してハードウェア加速処理システム２０５に入力することによって始まってもよい。レコードデータは、ハードウェア加速処理システム２０５によってデータベースから検索されてもよく、このデータベースは、たとえば、ＣＰＵ２１０と（たとえば、通信ネットワークを介して）作動的にデータ通信する第三者または政府のリモートデータベースであってもよい。いくつかの実装例においては、ＣＰＵ２１０は、大量のデータレコードをデータベースから検索して、ユーザ入力クエリのパラメータを使用して処理してもよい。いくつかの実施形態においては、それらの演算は、ＳＱＬ命令を含み得る。

【００３１】

いくつかの実装例においては、１つまたは複数の処理エンジン２２５は、カスタムＡＳＩＣを単独でまたはＦＰＧＡ２１５と組み合わせて利用することによって形成されてもよい。このような実装例においては、専用のハードウェア回路を有するカスタムＡＳＩＣは、示されている図面に描写されている例示的な処理エンジン機能のうちの１つまたはそれ以上を実行するように構成されてもよい。たとえば、カスタム固定ハードウェア回路を有するＡＳＩＣは、予め定められた一組のクエリ演算の少なくとも一部を効率的に実行することができるＤＤＲリーダ、バッファ、処理ユニットＰＵ２３０および／または連結ユニット２３５のうちの１つまたはそれ以上として機能するように設計されたハードウェア回路として構成されてもよい。ＡＳＩＣは、ハードウェア加速処理システム２０５によって処理される予め定められた一組のクエリ演算を実行するように構成されたハードウェア回路とともに配置されてもよい。いくつかの例においては、ＡＳＩＣに定義されるカスタム固定ハードウェア構成は、たとえば処理エンジン２２５においてＣＰＵ２１０および／またはＦＰＧＡ２１５などの１つまたは複数のＦＰＧＡから計算負荷をオフロードし得るクエリ命令を実行することが可能であってもよい。

【００３２】

図２（Ｂ）は、例示的なデータクエリを実行する図２（Ａ）のハードウェア加速処理システムのブロック図である。この示されている例においては、ハードウェア加速処理システム２０５は、ＣＰＵ２１０と、ＦＰＧＡ２１５と、データベース２２０と、ＦＰＧＡ２

１５内の電気的に再構成可能なハードウェア回路に含まれるクエリ言語処理ユニット２４０とを含む。いくつかの実施形態においては、クエリ言語処理ユニット２４０は、ＳＱＬ処理ユニットエンジンであってもよい。

【００３３】

ＣＰＵ２１０は、ユーザからデータクエリ要求を受信して、データクエリコマンド信号をクエリ言語処理ユニット２４０に送信して、たとえばデータストリームを処理することを含み得るデータベースクエリを実行するようにＦＰＧＡ２１５をプログラムしてもよい。いくつかの実施形態においては、データストリームは、ｉ個の異なるデータブロックを含み得る。いくつかの実施形態においては、ＦＰＧＡ２１５は、たとえばハードウェアリソース、異なる種類のレジスタ、マルチプレクサ、連結ユニットおよび／または加算器を含み得る。

【００３４】

クエリ言語処理ユニット２４０は、データクエリコマンド信号に従ってデータを処理するように適合された予め定められた命令セット群Ｇを提供する。この命令セット群Ｇは、１つまたは複数の命令セットＳ（ｉ）を含む。クエリ言語処理ユニット２４０がデータクエリコマンド信号を受け付けると、ＦＰＧＡ２１５は、予め定められた命令セット群Ｇに従ってデータストリームを並列に処理するための処理ユニットＰＵ（ｉ）を提供するように再構成される。ＰＵ（ｉ）は、図２（Ａ）における処理ユニット２３０である。いくつかの実施形態においては、各処理ユニットＰＵ（ｉ）は、命令セット群Ｇにおける予め定められた命令セットＳ（ｉ）に従ってデータストリームの対応する予め定められたデータブロック（ｉ）を処理するように、対応する一組のハードウェアリソースで構成されてもよい。

【００３５】

この例示的な例においては、ＦＰＧＡ２１５は、（たとえば、１つまたは複数のデータバッファを介して）データベースからデータストリームを受信する。データストリームをデータベースから検索した後、データストリームのデータブロック（ｉ）は、入力データスケジューラによってｉ個の異なる処理ユニットＰＵ（ｉ）に割り当てられてもよい。次いで、それらのｉ個の異なるデータブロックは、ｉ個の並列処理ユニットＰＵ（ｉ）、たとえばＰＵ（１），ＰＵ（２），ＰＵ（３），ＰＵ（４），.....ＰＵ（ｉ）によって処理されてもよい。各ＰＵ（ｉ）は、それ自体の（たとえば、独立した）命令セットＳ（ｉ）に従ってそのデータブロック（ｉ）を処理する。いくつかの実施形態においては、全てのＰＵ（ｉ）は、同一の命令セットＳ（ｉ）を有するように構成されてもよい。いくつかの実施形態においては、いくつかのＰＵ（ｉ）は、同一の命令セットＳ（ｉ）を有していてもよい。いくつかの実施形態においては、各ＰＵ（ｉ）は、異なる命令セットＳ（ｉ）を有していてもよい。

【００３６】

各処理ユニット２３０は、たとえば予め定められたフィルタおよび集約演算を実行するように独立してプログラムされてもよい。いくつかの実施形態においては、それらの演算は、たとえばＳＱＬ命令などの予め規定された一組のクエリ命令を含み得る。次いで、ハードウェア加速処理エンジン２２５は、最終的なクエリ結果をＣＰＵ２１０に転送してもよい。

【００３７】

いくつかの実施形態においては、ハードウェア加速処理エンジン２２５および／またはクエリ言語処理ユニット２４０は、一部が、命令のプログラムを実行するＣＰＵ２１０によって実現されてもよく、これらの命令は、実行されると、演算を実行させて、全面的にハードウェア加速回路の演算を介するのではなく、少なくとも一部がソフトウェア駆動演算を介して、ハードウェア加速クエリ処理結果を生成する。いくつかの実施形態においては、ハードウェア加速処理エンジン２２５および／またはクエリ言語処理ユニット２４０は、全部または一部が、ＡＳＩＣの固定回路に組み込まれてもよい。いくつかの実装例においては、ＰＵ（ｉ）は、たとえばＡＳＩＣに組み込まれた固定ハードウェア回路とＦＰ

ＧＡに組み込まれた再プログラム可能なハードウェア回路との直列および／または並列組み合わせによって実現されてもよい。

【００３８】

図３（Ａ）は、図２（Ｂ）のハードウェア加速処理エンジンに含まれる例示的な処理ユニットを示す図である。示されている図３（Ａ）においては、処理ユニット２３０（図２の処理ユニット２３０の一実施形態であってもよい）は、命令レジスタ３０５と、変数レジスタ３１０と、演算論理装置（ＡＬＵ）３２５とを含む。命令レジスタ３０５は、現在実行またはデコードされている演算および／または論理命令を格納するのに使用される。変数レジスタ３１０は、処理ユニット２３０に割り当てられるデータを受け付ける。演算論理装置（ＡＬＵ）３２５は、コンピュータ命令語の中のオペランドに対して算術および論理演算を実行するのに使用される。命令データのタイプは、ＳＱＬタイプのうちのいずれかであり得る。たとえば、命令データは、小数、整数、データ、ブーリアンであってもよい。この示されている例においては、命令データは、整数およびブーリアンを含む。処理ユニット２３０に追加されるＳＱＬタイプおよび命令は、処理ユニット２３０をたとえばＳＱＬ処理ユニットとして動作させてもよい。

【００３９】

いくつかの実施形態においては、定数データを格納するのに定数レジスタ３１５が使用されてもよい。動作時、たとえば命令の結果（たとえば、定数データ（よりも大きな）変数データ）を評価するために、実行時にデータストリームを処理する前に、ユーザ供給のクエリ基準を表す定数データが入力されてもよい。いくつかの実施形態においては、中間結果を保持するのに一時レジスタ３２０が使用されてもよい。いくつかの実施形態においては、ＡＬＵ３２５によって処理される必要があるデータを選択するのにマルチプレクサ３３０，３３５が使用されてもよい。いくつかの実施形態においては、変数レジスタ３１０は、データストリームのデータブロック（ｉ）をロードされてもよい。ＡＬＵ３２５は、命令レジスタ３０５に格納されたプログラムされた命令を実行することによって、ロードされたデータブロック（ｉ）に対して演算を実行してもよい。

【００４０】

いくつかの実施形態においては、マルチプレクサ３３０は、２：１マルチプレクサであってもよく、マルチプレクサ３３５は、３：１マルチプレクサであってもよい。いくつかの実施形態においては、マルチプレクサ３３０は、定数レジスタ３１５および変数レジスタ３１０から入力を受信してもよく、マルチプレクサ３３５は、定数レジスタ３１５、変数レジスタ３１０および一時レジスタ３２０から入力を受信してもよい。いくつかの実施形態においては、命令レジスタ３０５および定数レジスタ３１５は、所望の機能を実現するように（たとえば、クエリのランタイム実行の前に実行される、ＰＵ（ｉ）としての個々のハードウェア回路の設計時構成に）独立して予めプログラムされてもよい。

【００４１】

図３（Ｂ）は、図２（Ｂ）の例示的な処理ユニットを動作させるためのいくつかの例示的な命令を示す図である。命令データのタイプは、ＳＱＬタイプのうちのいずれかであり得る。たとえば、命令データは、小数、整数、データ、ブーリアンであってもよい。この示されている例においては、命令データは、整数およびブーリアンを含む。処理ユニット２３０に追加されるＳＱＬタイプおよび命令は、処理ユニット２３０をたとえばＳＱＬ処理ユニットとして動作させてもよい。いくつかの実施形態においては、セットの中の命令は、ＳＱＬフィルタおよび集約演算を行うように順番に実行されてもよい。いくつかの実施形態においては、命令は、フィルタ演算と、加算または減算演算と、乗算演算とを含み得る。いくつかの実施形態においては、フィルタ演算は、ＡＮＤ、ＯＲ、ＮＯＴ、ＥＱ、ＮＥＱを含み得る。いくつかの実施形態においては、それらの命令は、予めプログラム可能であって、再構成可能である。いくつかの実施形態においては、処理ユニット２３０の各々は、予めプログラム可能な異なる命令を実行してもよい。

【００４２】

図４は、データクエリジョブを実行するように構成されたハードウェア加速処理エンジ

ンのための例示的な設計時ファブリック再構成方法のフローチャートを示す。この例示的な例においては、プログラム可能な論理回路のブロックで利用可能なハードウェアリソースを実行してハードウェア加速処理エンジン（たとえば、図２（Ａ）におけるハードウェア加速処理エンジン２２５）を形成するためのコマンド信号が生成される。例示的な方法４００においては、４０５において、ＣＰＵ（たとえば、ＣＰＵ２１０）は、ＣＰＵ２１０を介してユーザからデータクエリコマンド信号を受信する。図２Ａを参照して、ＣＰＵ２１０がデータクエリコマンド信号を受信すると、データクエリ応答速度を上げるために、クエリジョブの一部がハードウェア加速処理エンジン２２５にオフロードまたは割り当てられてもよい。４１０において、ＣＰＵ２１０は、データストリームを処理するためのジョブがＦＰＧＡ２１５にオフロードまたは割り当てられ得るかを判断する。ジョブがハードウェア加速処理エンジン２２５にオフロードされるのに適していない場合、プロセスの制御は、ＣＰＵ２１０に移って、４１５において、ハードウェア加速処理エンジン２２５を使用することなくデータクエリを実行するための命令を実行してもよい。

【００４３】

４１０においてジョブがハードウェア加速処理エンジン２２５にオフロードされるのに適している場合、４２０において、ＣＰＵ２１０は、データクエリコマンド信号を、たとえばＳＱＬコマンドなどの予め定められたクエリ言語コマンドに変換する。４２５において、このＳＱＬコマンドを使用して、ＣＰＵ２１０は、パラメータおよび命令を抽出する。示されている例においては、４３０において、ＣＰＵ２１０は、ＦＰＧＡ２１５のプログラム可能な論理に前もってプログラムされたいかなる既存の構成パラメータまたは命令もクリアすることによって、全ての利用可能なＰＵをクリアする。

【００４４】

ユーザからのデータクエリコマンド信号をオフロードするようにＦＰＧＡ２１５を構成する準備をするために、４３５において、ＣＰＵ２１０は、変数ｉ＝１を起動する。４４０において、ＣＰＵは、処理ユニットＰＵ（ｉ）に対応する複数組の抽出されたパラメータおよび命令をロードする。４４５において、いずれの追加の複数組の抽出されたパラメータおよび命令も処理するのにより多くのＰＵ（ｉ）が利用可能である場合、４５０において、ＣＰＵは、変数ｉをインクリメントして、４４０にループバックする。４４５において、いずれの追加の複数組の抽出されたパラメータおよび命令も処理するのにより多くのＰＵ（ｉ）が利用可能でない場合、方法４００は終了する。

【００４５】

例示的な例においては、ＦＰＧＡ２１５は、ジョブをＣＰＵからオフロードするか否かを選択するように構成されてもよい。たとえば、ＦＰＧＡオフロードスイッチを使用して、オフロードを受け付けたり拒否したりしてもよい。ジョブを受け付けるか拒否するかは、ジョブのタイプによって左右され得る。この示されている例においては、ＦＰＧＡ２１５は、ジョブがデータスキャンおよび集約に関連している場合にジョブを受け付けるように構成されてもよい。ＦＰＧＡがオフロードを拒否する場合、ＣＰＵ２１０は、クエリを処理してもよい。データクエリコマンド信号に応答して、クエリ言語処理ユニット２４０は、命令セット群Ｇを生成するように構成されてもよい。この命令セット群Ｇは、１つまたは複数の命令セットＳ（ｉ）を含み得る。並列処理ユニット（たとえば、図２ＢのＰＵ（ｉ））の各々は、予め定められた命令セットＳ（ｉ）を実行するようにプログラムされてもよい。より具体的には、クエリ言語処理ユニット２４０は、クエリをＳＱＬコマンドに移し、ＳＱＬコマンドからパラメータを抽出し、ＦＰＧＡ２１５内の各々の対応する処理ユニットによって実行される命令セットを生成するように構成されてもよい。いくつかの実施形態においては、各処理ユニットは、同一の命令セットを有していてもよい。いくつかの実施形態においては、各処理ユニットは、異なる命令セットを実行してもよい。

【００４６】

次に、命令セット群Ｇおよび抽出されたパラメータをＦＰＧＡ内の並列処理ユニットにロードして、データストリームを処理してもよい。たとえば、ＰＵ（１）は、第１の命令セットＳ（１）によってロードされてもよく、ＰＵ（２）およびＰＵ（３）は、第２の命

令セットＳ（２）によってロードされてもよい。より具体的には、各ＰＵ（ｉ）について、定数レジスタ（たとえば、図３（Ａ）における定数レジスタ３１５）は、１つまたは複数の定数をロードされてもよい。命令レジスタ（たとえば、図３（Ａ）における命令レジスタ３０５）は、予め定められた命令セットＳ（ｉ）をロードされてもよい。新たなパラメータを処理ユニットの各々にロードする前に、処理ユニットの各々はクリアされてもよい。たとえば、定数レジスタおよび命令レジスタがクリアされてもよい。いくつかの実施形態においては、変数レジスタ（たとえば、変数レジスタ３１０）および一時レジスタ（たとえば、一時レジスタ３２０）もクリアされてもよい。

【００４７】

その結果、ハードウェア加速処理エンジンは、データストリームを処理する準備ができているであろう。

【００４８】

設計時プロセスを説明するために、例示的な例についてさらに説明する。たとえば、ユーザは、１９９４年に発生したオンライン小売業者の累積収益を知りたいと思ったとする。商品割引率は５％～７％であり、商品量は２４個未満であった。次いで、ユーザは、クエリをＣＰＵ２１０に送信する。このクエリは、データスキャンおよび集約に関連している。したがって、ＦＰＧＡ２１５は、このクエリを処理することができる能力を有している。次いで、ＣＰＵ２１０は、クエリジョブをＦＰＧＡにオフロードする。クエリ言語処理ユニット２４０は、クエリを受け付けて、クエリをＳＱＬコマンドに移してもよい。

【００４９】

例示的なクエリは、以下のようなものであってもよい。

【００５０】

【数１】

```
SELECT SUM(L_EXTENDEDPRICE*L_DISCOUNT) AS REVENUE
FROM LINEITEM
WHERE L_SHIPDATE >= '1994-01-01' AND L_SHIPDATE < '1995-01-01'
AND L_DISCOUNT BETWEEN .06 - 0.01 AND .06 + 0.01 AND L_QUANTITY < 24
```

【００５１】

次いで、このクエリは、ソフトウェアを使用してＳＱＬコマンドに変換されてもよい。ソフトウェアは、ＳＱＬコマンドからパラメータおよび命令を抽出してもよい。処理ユニットの前の例示的なＳＱＬコマンドは、以下のようなものであってもよい。

【００５２】

【数２】

```
Define constants like -
def l_quantity 0
def l_extendedprice 1
def l_discount 2
def l_shipdate 3
def d1994_01_01 1994/01/01
def c1 1
def d1995_01_01 1995/01/01
def c5 500
LD CONST[0] d1994_01_01
GE VAR[l_shipdate] CONST[0] TEMP[C0]
# save the result of "l_shipdate < date '1994-01-01' + interval '1' year" to TEMP1
LT VAR[l_shipdate] CONST[1] TEMP[1]
# save the result of "l_shipdate >= date '1994-01-01'
# and l_shipdate < date '1994-01-1' + interval '1' year" to TEMP0
AND, TEMP, TEMP0, BOOL, TEMP, TEMP1, BOOL, TEMP, TEMP0, BOOL</V>
# Decompose "l_discount between .06 - 0.01 and .06 + 0.01" as
# a. "l_discount >= .05" and
# b. "l_discount <= .07"
# save the result of "l_discount <.05" to TEMP1
<V>GE, VAR, l_discount, DECIMAL, CONST, exp3_1_con, DECIMAL, TEMP,
TEMP1, BOOL</V>
<V>AND, TEMP, TEMP0, BOOL, TEMP, TEMP1, BOOL, TEMP, TEMP0,
BOOL</V>
<V>LE, VAR, l_discount, DECIMAL, CONST, exp3_2_con, DECIMAL, TEMP,
TEMP1, BOOL</V>
<V>AND, TEMP, TEMP0, BOOL, TEMP, TEMP1, BOOL, TEMP, TEMP0,
BOOL</V>
# save the result of "l_quantity < 24" to TEMP1
<V>LT, VAR, l_quantity, DECIMAL, CONST, exp4_con, DECIMAL, TEMP, TEMP1,
BOOL</V>
# save the result of the whole qualifier to TEMP0
<V>AND, TEMP, TEMP0, BOOL, TEMP, TEMP1, BOOL, TEMP, TEMP0,
BOOL</V>
</E>
```

【００５３】

図４に示されるように、次いで、抽出されたパラメータおよび命令セットＳはそれぞれ、ＦＰＧＡ２１５内の利用可能な処理ユニットにロードされて、クエリ演算を実行してもよい。データが実行時にストリーミングされると、それらのプログラムされた処理ユニットは、クエリ演算の実行を開始してもよく、その一例について図６を参照してさらに詳細に説明する。

【００５４】

図５は、ハードウェア加速処理エンジンの例示的な構造を示す図である。図２（Ａ～Ｂ）を参照して、図５は、ＰＵ（ｉ）２３０への入力を調整するための入力データスケジュ

ーラ５０５と、ＰＵ（ｉ）２３０の出力を処理するための連結ユニット２３５とを含むハードウェア加速処理エンジン２２５の実施形態を示している。

【００５５】

入力データスケジューラ５０５は、入来するデータストリームをいくつかの予め定められたデータブロックに分割するように構成されてもよい。次いで、これらのデータブロックは、そのデータブロックに適切な予め定められた命令セットＳ（ｉ）を実行するように適切に構成されたＰＵ（ｉ）２３０に割り当てられることができる。いくつかの実施形態においては、ＰＵ（ｉ）のうちの少なくとも２つまたはそれ以上がデータブロック上で並列に動作する。示されている例においては、データストリームは、たとえば、データストリームソース（たとえば、データベース）と作動的に通信する高速データ通信チャネル（たとえば、ＰＣｌｅ、ＤＭＡ）を介してハードウェア加速処理エンジン２２５に送られてもよい。入力データスケジューラ５０５によって処理されたストリーミングデータのブロックの各々は、ＰＵ（ｉ）２３０に送られる前にＢＲＡＭのブロックを介してバッファリングされてもよい。

【００５６】

連結ユニット２３５は、ＰＵ（ｉ）２３０の各々によって処理された結果を併合するように構成されてもよい。いくつかの実施形態においては、連結ユニット２３５は、ハードウェアユニットであってもよい。いくつかの実施形態においては、各処理ユニット２３０は、予め定められたデータフィルタおよび集約命令を実行するように構成されてもよい。

【００５７】

ハードウェア加速処理エンジン２２５は、このハードウェア加速処理エンジンにおけるジョブをスケジューリングするためのスケジューラ５０５も含む。スケジューラ５０５は、データストリームにおけるデータブロックを対応する処理ユニットに割り当てる。いくつかの実施形態においては、スケジューラ５０５は、ラウンドロビンスケジューラであってもよい。いくつかの実施形態においては、各処理ユニット２３０は、異なる読み書きデータ幅を有するそれ自体のＨＴＴＰライブストリーミング（ＨＬＳ）ストリームに送り込まれてもよい。いくつかの実施形態においては、読み書きバッファのブロックＲＡＭの使用を最小限に抑えるために、ストライプ読み取りバッファが使用されてもよい。いくつかの実施形態においては、使用される光学ＰＵリソースおよび処理サイクル数で効率を最大化するために、広いＤＤＲ（ダブルデータレート）幅が使用されてもよい。いくつかの実施形態においては、ＤＤＲバースト長は、ＤＤＲ非効率を減少させるのに十分に長いであろう。

【００５８】

図６は、プログラム可能なハードウェア加速処理エンジンを用いてデータクエリを実行するための例示的な実行時方法のフローチャートを示す。方法６００においては、６０５において、プログラム可能なハードウェア加速処理エンジン（たとえば、図５におけるハードウェア加速処理エンジン２２５）は、データベースからデータストリームを検索する。６１０において、スケジューラ（たとえば、図５におけるスケジューラ５０５）は、プログラムされた命令に応答して、データストリームを、プログラム可能なハードウェア加速処理エンジン内のさまざまな独立したプログラム可能な処理ユニット（たとえば、図５における処理ユニット２３０）によって処理されるさまざまなデータブロックにスケジューリングして分割する。次いで、６１５において、処理エンジン内の各々の独立したプログラム可能な処理ユニットは、予め定められたデータフィルタおよび集約演算を実行して、処理結果を生成する。いくつかの実施形態においては、ＳＱＬクエリを使用してデータフィルタおよび集約演算を実行してもよい。６２０において、それらの処理結果は、連結ユニット（たとえば、図５における連結ユニット２３５）によって連結されて、出力結果を形成する。６２５において、データクエリプロセスを終了させるか否かを判断する。データストリームが、処理すべきより多くのレコードを有している場合、方法６００は６０５に戻る。データストリームが、処理すべきより多くのレコードを有していない場合、方法は、クエリの結果を確定して、完了する。いくつかの実施形態においては、次いで、こ

のクエリ結果出力は、たとえばユーザインターフェイス上に表示されてもよい。

【００５９】

例示的な例においては、クエリアルゴリズムは、以下のようなものであってもよい。

【００６０】

【数３】

```
LOAD INSTRUCTION REGISTERS
DO FOR EACH DATA BLOCK {
    LOAD VARIABLE REGISTERS
    EXECUTE INSTRUCTIONS
    MOVE RESULT BACK TO HOST CPU
}
```

【００６１】

設計時プロセスを説明するために、例示的な例についてさらに説明する。たとえば、ＦＰＧＡ２１５は、クエリ演算を実行するのに利用可能な５つの処理ユニットを含み得る。抽出されたパラメータおよび命令は、クエリ演算を実行するために既にＦＰＧＡ２１５にロードされていてもよい。次いで、ＦＰＧＡは、オンライン小売業者の売上記録に関連するデータストリームをデータベースから検索してもよい。この売上記録は、２００ページのデータを含み得る。ＦＰＧＡは、記録データのうちの２０ページをデータベースから検索してもよい。スケジューラ５０５は、１ページ目の検索されたデータを第１の処理ユニットに割り当て、２ページ目のデータを第２の処理ユニットに割り当てる、などしてもよい。割り当てられた結果の一例は、以下の表に示されている。

【００６２】

【表１】

	データを適切なＰＵに割り当てる			
PU1	１ページ目のデータ	６ページ目のデータ	１１ページ目のデータ	１６ページ目のデータ
PU2	２ページ目のデータ	７ページ目のデータ	１２ページ目のデータ	１７ページ目のデータ
PU3	３ページ目のデータ	８ページ目のデータ	１３ページ目のデータ	１８ページ目のデータ
PU4	４ページ目のデータ	９ページ目のデータ	１４ページ目のデータ	１９ページ目のデータ
PU5	５ページ目のデータ	１０ページ目のデータ	１５ページ目のデータ	２０ページ目のデータ

【００６３】

次いで、処理ユニットの各々は、データが変数レジスタにロードされるときにその命令を実行してもよい。一時的な結果は、一時レジスタ３２０に格納されてもよい。１ページ目、６ページ目、１１ページ目、１６ページ目が全て処理ユニットＰＵ１によって照会された後、ＰＵ１は、それらの命令の下で、１ページ目、６ページ目、１１ページ目、１６ページ目の第１の累積収益合計１を出力してもよい。ＰＵ２は、第２の累積収益合計２を出力する、などである。出力連結ユニット２３５は、最終的な計算を実行して、最終結果をユーザに返す。

【００６４】
さまざまな実施形態について図面を参照して説明してきたが、他の実施形態も可能である。たとえば、いくつかの実施形態においては、ＳＱＬクエリは、命令に変換されて、ロードされて、ＦＰＧＡ上で実行されてもよい。いくつかの実施形態においては、ＦＰＧＡ上で再コンパイルすることなく異なるＳＱＬクエリが実行されてもよい。いくつかの実施形態においては、これらの実行されたＳＱＬクエリは、ＯＬＡＰ（オンライン分析処理）に好適であり得る。いくつかの実施形態においては、ＰｏｓｔｇｒｅＳＱＬおよびそのさまざまな拡張機能がデータ分析およびＧＩＳアプリケーションに使用されてもよい。
【００６５】
いくつかの実施形態においては、ユーザは、加速されたＦＰＧＡプラットフォーム上で既存のＰｏｓｔｇｒｅｓＳＱＬクエリを実行することができる。いくつかの実施形態においては、ハードウェア加速処理エンジンは、超並列ＳＱＬ処理ユニットであってもよく、超並列ＳＱＬ処理ユニットのための命令コードは、各々の連続したユーザクエリについて、実行中に生成されてもよい。いくつかの実施形態においては、Ｐｏｓｔｇｒｅｓストレージページは、関係の行をスキャンして、ｗｈｅｒｅ句によって指定された行を選択するように、ＦＰＧＡにおいてネイティブにパースされてもよく、ユーザは、全ての既存のＰｏｓｔｇｒｅｓ特徴を使用してリモートまたはローカルクエリを実行してもよい。
【００６６】
いくつかの実施形態においては、ユーザは、たとえばＦ１実装例のためにＶＵ９Ｐデバイス上で３２　ＳＱＬ　ＰＵを使用してもよい。いくつかの実施形態においては、ＰＵの各々は、ハッシュ、ソート、または顧客に特有の命令に拡張可能である。いくつかの実施形態においては、処理中のデータは、入出力データを保持してＦＰＧＡ加速カーネルが窮乏しないことを確実にするために、複数のバッファを使用してＦＰＧＡからブロックストリーミングされてもよい。
【００６７】
実施形態のいくつかの局面は、コンピュータシステムとして実現されてもよい。たとえば、さまざまな実装例は、デジタルおよび／またはアナログ回路、コンピュータハードウェア、ファームウェア、ソフトウェア、またはそれらの組み合わせを含み得る。装置要素は、プログラム可能なプロセッサによる実行のために、情報担体、たとえば機械読取可能な記憶装置において有形に具体化されるコンピュータプログラム製品に実装されることができ、方法は、入力データ上で動作して出力を生成することによってさまざまな実施形態の機能を実行するように命令のプログラムを実行するプログラム可能なプロセッサによって実行されることができる。有利なことに、いくつかの実施形態は、データストレージシステム、少なくとも１つの入力装置および／または少なくとも１つの出力装置との間でデータおよび命令を送受信するように結合された少なくとも１つのプログラム可能なプロセッサを含むプログラム可能なシステム上で実行可能な１つまたは複数のコンピュータプログラムで実現されてもよい。コンピュータプログラムは、特定のアクティビティを実行するためまたは特定の結果を生じさせるために直接的または間接的にコンピュータで使用することができる一組の命令である。コンピュータプログラムは、コンパイルまたはインタープリタ言語を含む任意の形式のプログラミング言語で書き込むことができ、コンピュータプログラムは、スタンドアロンのプログラムとして、またはコンピューティング環境での使用に適したモジュール、コンポーネント、サブルーチンもしくは他のユニットとして、などの任意の形式で展開することができる。
【００６８】
命令のプログラムの実行に適したプロセッサは、限定としてではなく一例として、汎用マイクロプロセッサおよび特別目的マイクロプロセッサの両方のマイクロプロセッサを含み、これらのマイクロプロセッサは、任意の種類のコンピュータの単一のプロセッサまたは複数のプロセッサのうちの１つを含み得る。一般に、プロセッサは、リードオンリメモリまたはランダムアクセスメモリまたはそれら両方から命令およびデータを受信する。コンピュータの必須の要素は、命令を実行するためのプロセッサ、および、命令およびデー

タを格納するための１つまたは複数のメモリである。コンピュータプログラム命令およびデータを有形に具体化するのに適した記憶装置は、全ての形態の不揮発性メモリを含み、これらの不揮発性メモリは、一例として、ＥＰＲＯＭ、ＥＥＰＲＯＭおよびフラッシュメモリデバイスなどの半導体メモリデバイスを含む。プロセッサおよびメモリは、ＡＳＩＣ（特定用途向け集積回路）によって補完されるか、またはＡＳＩＣに組み入れられることができる。いくつかの実施形態においては、プロセッサおよびメモリは、たとえばＦＰＧＡなどのプログラム可能なハードウェアデバイスによって補完されるか、またはプログラム可能なハードウェアデバイスに組み入れられることができる。

【００６９】

いくつかの実装例においては、各システムは、同一または同様の情報でプログラムされてもよく、および／または、揮発性および／または不揮発性メモリに格納された実質的に同一の情報で初期化されてもよい。たとえば、１つのデータインターフェイスは、デスクトップコンピュータまたはサーバなどの適切なホストデバイスに結合されたときに、自動構成機能、自動ダウンロード機能および／または自動更新機能を実行するように構成されてもよい。

【００７０】

さまざまな実施形態においては、コンピュータシステムは、非一時的なメモリを含み得る。このメモリは、プロセッサによって実行可能なプログラム命令を含むデータおよびコンピュータ読取可能な命令を格納するように構成され得る１つまたは複数のプロセッサに接続されてもよい。これらのデータおよびコンピュータ読取可能な命令は、１つまたは複数のプロセッサがアクセス可能であってもよい。プロセッサによって実行可能なプログラム命令は、１つまたは複数のプロセッサによって実行されると、１つまたは複数のプロセッサにさまざまな動作を実行させてもよい。

【００７１】

さまざまな実施形態においては、コンピュータシステムは、モノのインターネット（ＩｏＴ）デバイスを含み得る。ＩｏＴデバイスは、電子機器、ソフトウェア、センサ、アクチュエータおよびネットワーク接続に組み込まれたオブジェクトを含み得て、これらのオブジェクトがデータを収集してやりとりすることを可能にする。ＩｏＴデバイスは、インターフェイスを介して別のデバイスにデータを送信することによって、有線または無線デバイスと併用されてもよい。ＩｏＴデバイスは、有用なデータを収集して、次いでそのデータを他のデバイス間に自律的に流してもよい。

【００７２】

モジュールのさまざまな例は、さまざまな電子ハードウェアを含む回路を使用して実現されてもよい。限定としてではなく一例として、ハードウェアは、トランジスタ、抵抗器、キャパシタ、スイッチ、集積回路および／または他のモジュールを含み得る。さまざまな例においては、モジュールは、さまざまな集積回路を含むシリコン基板上に作製されたアナログおよび／またはデジタル論理、個別部品、トレースおよび／またはメモリ回路を含み得る。いくつかの実施形態においては、モジュールは、プロセッサによって実行される予めプログラムされた命令および／またはソフトウェアの実行を含み得る。たとえば、さまざまなモジュールは、ハードウェアもソフトウェアも含み得る。

【００７３】

一例においては、集積回路は、並列に配置された複数の処理ユニット（ＰＵ（ｉ））を含み、上記複数のＰＵ（ｉ）の各々は、予め定められたクエリ言語を使用する命令セット群（Ｇ）における予め定められた命令セット（Ｓ（ｉ））に従ってデータストリームの選択されたデータブロックを処理するように構成されたハードウェア回路で形成され、上記複数のＰＵ（ｉ）の各々は、上記ＰＵ（ｉ）によって処理される上記データストリームの上記選択された部分に対応する中間出力結果を生成し、上記集積回路はさらに、上記複数のＰＵ（ｉ）の各々から上記中間出力結果の各々を受信して、集約結果を生成するように結合された連結回路を含み、上記Ｓ（ｉ）の各々は、ユーザ定義のクエリから抽出された命令の関数を備える。

【００７４】
一例においては、上記予め定められた命令セット（Ｓ（ｉ））は、ＳＱＬ命令を備える。一例においては、上記予め定められたクエリ言語は、ＳＱＬを備える。一例においては、上記連結ユニットは、上記ユーザ定義のクエリに関連付けられた予め定められた関数に従って上記集約結果を生成するように構成される。一例においては、上記複数の処理ユニットの各々は、ＡＳＩＣにおける固定ハードウェア回路として実現される。一例においては、上記複数の処理ユニットの各々は、ＦＰＧＡのプログラム可能なファブリックにおける再構成可能なハードウェアとして実現される。一例においては、上記集積回路は、上記データストリームを受信するように結合されたスケジューラ回路を含み、上記スケジューラ回路は、上記データブロックの各々を上記複数のＰＵ（ｉ）のうちの１つに選択的に向けるように構成される。一例においては、上記スケジューラ回路は、ラウンドロビンスケジューラを備える。一例においては、上記Ｓ（ｉ）の各々は、上記ユーザ定義のクエリから抽出された少なくとも１つのパラメータの関数をさらに備える。
【００７５】
一例においては、上記複数のＰＵ（ｉ）の各々は、抽出された命令に基づいて演算を実行することによって上記対応するＳ（ｉ）を実行するように適合された演算論理装置ＡＬＵを備え、上記演算は、（ｉ）変数レジスタに格納された上記データストリームの一部を備える第１のオペランドと、（ｉｉ）定数レジスタに格納された上記抽出されたパラメータのうちの１つを備える第２のオペランドとを使用して実行される。一例においては、上記複数の処理ユニットの各々は、上記実行された演算の結果を保持するように構成された一時レジスタと、上記定数レジスタおよび上記変数レジスタから入力を受信するように構成された第１のマルチプレクサと、上記定数レジスタ、上記変数レジスタおよび上記一時レジスタから入力を受信するように構成された第２のマルチプレクサとをさらに備える。
【００７６】
一例においては、データクエリを実行するようにファブリックを構成する方法は、ユーザからデータクエリを受信するステップと、上記データクエリを予め定められたクエリ言語コマンドに変換するステップと、上記コマンドから、複数の並列処理ユニットＰＵ（ｉ）に格納されるパラメータを抽出するステップと、上記コマンドから命令を抽出して、上記ＰＵ（ｉ）によって実行される命令セット群Ｇを形成するステップとを含み、上記命令セット群Ｇは、複数の命令セットＳ（ｉ）を備え、上記方法はさらに、上記複数のＰＵ（ｉ）に上記抽出されたパラメータおよび上記抽出された命令をロードするステップを含み、上記ＰＵ（ｉ）の各々は、データストリームの予め定められたデータブロック（ｉ）を、その対応するパラメータおよび命令セットＳ（ｉ）を用いて並列に処理するように構成される。
【００７７】
一例においては、上記予め定められたクエリ言語は、ＳＱＬを備える。一例においては、上記ＰＵ（ｉ）の各々は、同一のＳ（ｉ）で構成される。一例においては、上記方法は、上記抽出されたパラメータおよび上記抽出された命令をロードする前に全ての上記ＰＵ（ｉ）をクリアするステップをさらに含む。
【００７８】
多数の実装例について説明してきた。しかし、さまざまな変更がなされてもよいということが理解されるであろう。たとえば、開示されている技術のステップが異なるシーケンスで実行される場合、または、開示されているシステムの構成要素が異なる態様で組み合わせられる場合、または、それらの構成要素が他の構成要素で補完される場合に、有利な結果が実現され得る。したがって、他の実装例は、以下の特許請求の範囲の範囲内である。

【図面】

【図１】

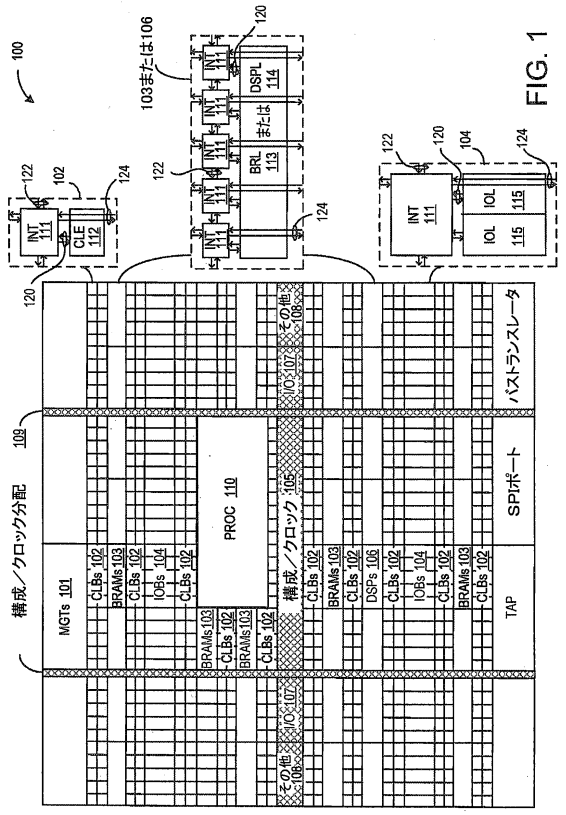

FIG. 1

【図２（Ａ）】

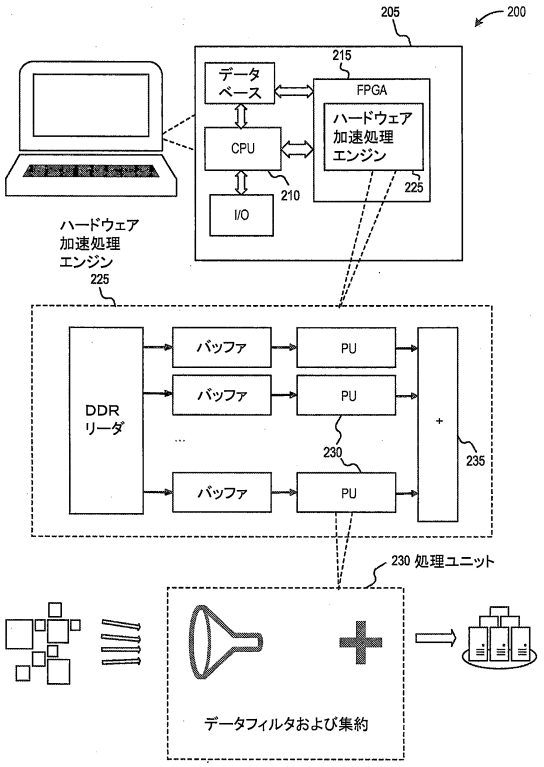

FIG. 2(A)

【図２（Ｂ）】

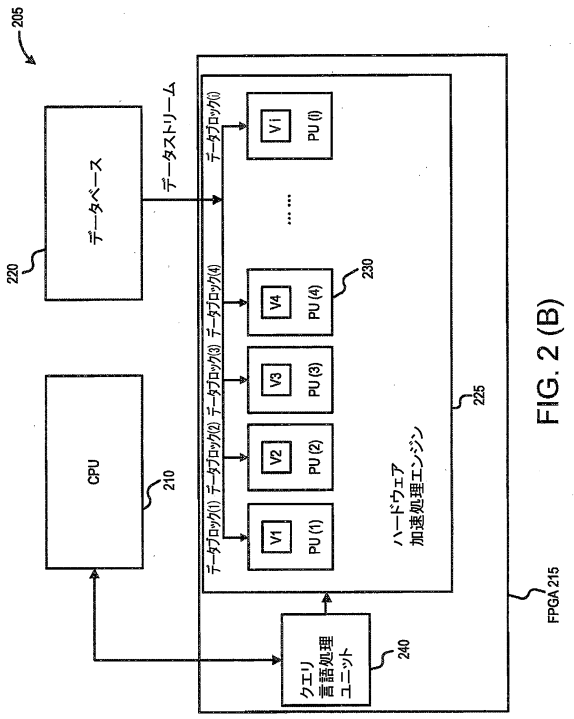

FIG. 2 (B)

【図３（Ａ）】

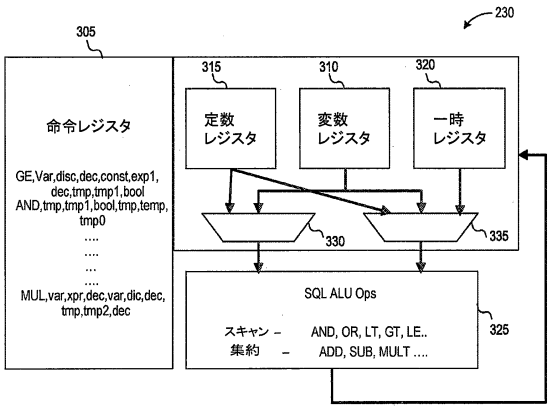

FIG. 3 (A)

【図３（Ｂ）】

命令	左および右オペランド	結果	備考
AND, OR	64 bit reg	1-bit bool	フィルタ演算
NOT, GT, GE, LT, LE, EQ, NEQ	64 bit reg	1-bit bool	フィルタ演算
PLUS, SUB	64 bit reg	64 bit reg	加算／減算
MULT	64 bit reg	64 bit reg	乗算

FIG. 3 (B)

【図４】

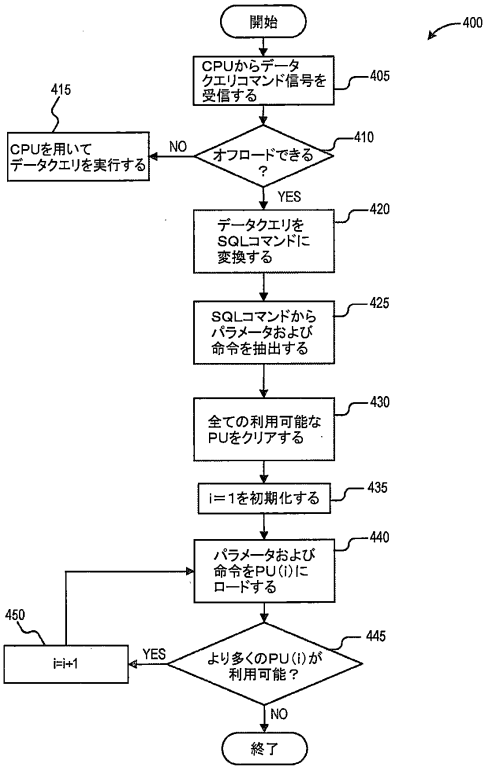

FIG. 4

【図５】

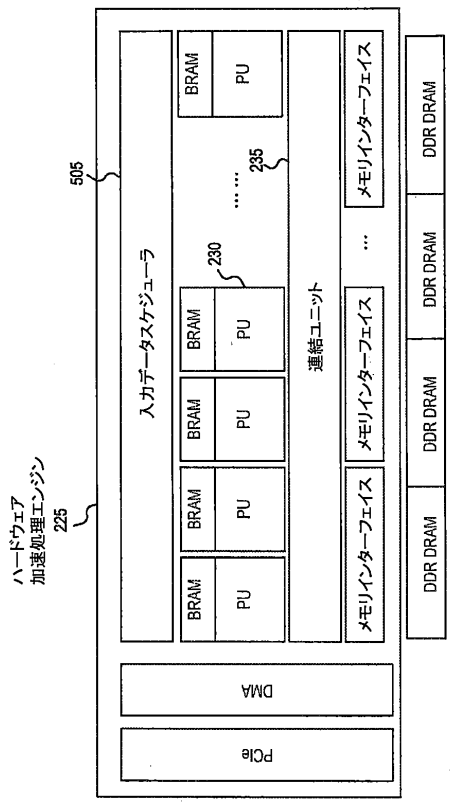

FIG. 5

【図６】

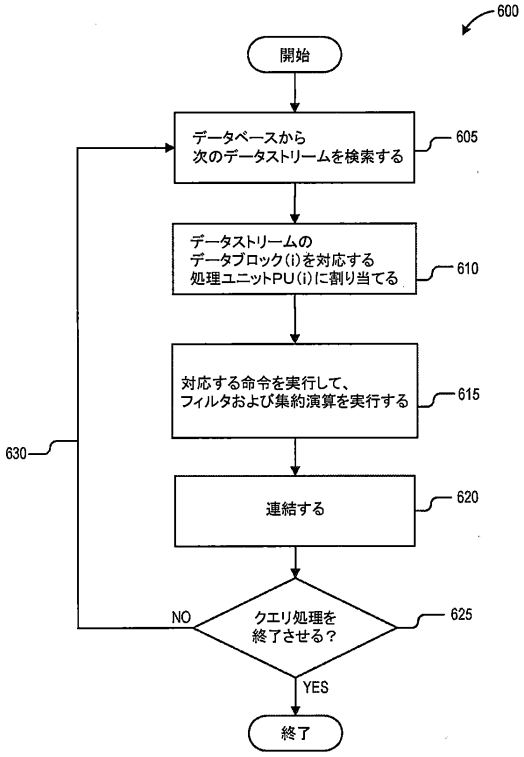

FIG. 6

フロントページの続き

イブ、２１００

審査官　坂東 博司

(56)参考文献
国際公開第２０１６／１８５５４２（WO，Ａ１）
特表２０１５－５３２７４９（ＪＰ，Ａ）
米国特許出願公開第２０１５／０５３２７４９（ＵＳ，Ａ１）
米国特許出願公開第２０１４／０３７９１１３（ＵＳ，Ａ１）
三好 健文 TAKEFUMI MIYOSHI，ストリーム処理エンジン向け動的再構成可能プロセッサアーキテクチャの設計 A Dynamic Reconfigurable Processor Architecture for Stream Processing Engine，情報処理学会論文誌 論文誌トランザクション ２０１１（平成２３）年度▲１▼［ＣＤ－ＲＯＭ］，日本，一般社団法人情報処理学会，2011年10月15日，第4巻第2号，３５～５１，【ISSN】1882-7772
Divya Mahajan et al，In-RDBMS Hardware Acceleration of Advanced Analytics，ARXIV.ORG, Cornell University Library，米国，ARXIV.ORG, Cornell University Library，2018年01月08日，1-15，https://arxiv.org/abs/1801.06027，DOI:10.14778/3236187.3236188

(58)調査した分野 (Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　１７／１０
Ｇ０６Ｆ　９／５０
Ｇ０６Ｆ　１５／８０
Ｇ０６Ｆ　９／３０
Ｇ０６Ｆ　１６／２４５