

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6974588号
(P6974588)

(45) 発行日 令和3年12月1日 (2021. 12. 1)

(24) 登録日 令和3年11月8日 (2021. 11. 8)

(51) Int. Cl.	F I
G06F 9/48 (2006.01)	G06F 9/48 300Z
H03K 19/177 (2020.01)	H03K 19/177

請求項の数 20 (全 25 頁)

(21) 出願番号	特願2020-507062 (P2020-507062)	(73) 特許権者	591016172
(86) (22) 出願日	平成30年7月13日 (2018. 7. 13)		アドバンスト・マイクロ・ディバイス・
(65) 公表番号	特表2021-503109 (P2021-503109A)		インコーポレイテッド
(43) 公表日	令和3年2月4日 (2021. 2. 4)		ADVANCED MICRO DEVI
(86) 国際出願番号	PCT/US2018/042172		CES INCORPORATED
(87) 国際公開番号	W02019/099067		アメリカ合衆国 95054 カリフォル
(87) 国際公開日	令和1年5月23日 (2019. 5. 23)		ニア州、 サンタ クララ、 オーガスティ
審査請求日	令和3年7月13日 (2021. 7. 13)		ン ドライブ 2485
(31) 優先権主張番号	15/812, 411	(74) 代理人	100108833
(32) 優先日	平成29年11月14日 (2017. 11. 14)		弁理士 早川 裕司
(33) 優先権主張国・地域又は機関	米国 (US)	(74) 代理人	100111615
			弁理士 佐野 良太
		(74) 代理人	100162156
			弁理士 村雨 圭介
早期審査対象出願			

最終頁に続く

(54) 【発明の名称】 仮想FPGAの管理及び最適化システム

(57) 【特許請求の範囲】

【請求項1】

フィールドプログラマブルゲートアレイ (FPGA) デバイスのセットの複数のマクロコンポーネントを列挙するデータベースを更新するように構成されたリソース追跡モジュールと、

設計のための1つ以上の指定されたマクロコンポーネントを示す第1設計定義を第1クライアントデバイスから受信するように構成された通信インタフェースと、

前記第1設計定義に示された前記1つ以上の指定されたマクロコンポーネント毎に前記複数のマクロコンポーネントのうち1つを割り当てることによって、前記設計のためのマクロコンポーネントの第1セットを割り当てるように構成されたリソース割り当てロジックと、

割り当てられたマクロコンポーネントの前記第1セットを前記第1設計定義に従って構成することによって、前記FPGAデバイスのセットに前記設計を実装するように構成された構成ロジックと、を備える、

マクロスケジューラ。

【請求項2】

前記第1設計定義は、前記1つ以上の指定されたマクロコンポーネント間の接続を定義し、前記1つ以上の指定されたマクロコンポーネントの各々のタイプを示すマクログラフを含む、

請求項1のマクロスケジューラ。

10

20

【請求項 3】

前記 1 つ以上の指定されたマクロコンポーネントは、1 つ以上の指定されたタイルと、1 つ以上の指定された固定関数ユニットと、1 つ以上の指定されたレジスタと、を含み、前記第 1 設計定義は、前記 1 つ以上の指定されたタイルの各々のビットファイルを示し、

前記構成ロジックは、前記 1 つ以上の指定されたタイル毎に、マクロコンポーネントの前記第 1 セット内の割り当てられたタイルを、前記指定されたタイルのビットファイルに基づいてプログラムするように構成されている、

請求項 2 のマクロスケジューラ。

【請求項 4】

前記リソース追跡モジュールは、マクロコンポーネントの前記第 1 セットの各々を、前記第 1 クライアントデバイスのネットワークアドレスと、前記第 1 クライアントデバイスによって要求された第 1 タスクと関連付けるようにさらに構成されている、

請求項 1 のマクロスケジューラ。

【請求項 5】

前記リソース追跡モジュールは、前記複数のマクロコンポーネント毎に、FPGA デバイスのセットにおける前記マクロコンポーネントの位置及び前記マクロコンポーネントの利用可能性を前記データベースに記録するようにさらに構成されている、

請求項 1 のマクロスケジューラ。

【請求項 6】

前記 FPGA デバイスのセットの初期構成におけるタスクの実行中に、前記設計のための最適化された構成を生成するように構成された合成ロジックをさらに備え、

前記初期構成は前記第 1 設計定義によって示され、

前記構成ロジックは、前記 FPGA デバイスのセットの最適化された構成でタスクの実行を再開する前に前記初期構成を前記最適化された構成に置き換えるように、前記 FPGA デバイスのセットを再プログラムするように構成されており、

前記合成ロジックは、前記初期構成における初期ポートの位置を、前記最適化された構成における最適化されたポートの位置と相関させる再マッピング通知を生成するように構成されている、

請求項 1 のマクロスケジューラ。

【請求項 7】

前記第 1 設計定義は、アクセラレータの定義を含み、

前記リソース割り当てロジックは、

第 1 期間中に前記第 1 クライアントデバイスからの第 1 タスクのアクセラレータにおける実行をスケジューリングし、

第 2 期間中に第 2 クライアントデバイスからの第 2 タスクのアクセラレータにおける実行をスケジューリングする、ように構成されている、

請求項 1 のマクロスケジューラ。

【請求項 8】

前記リソース割り当てロジックは、前記複数のマクロコンポーネントの第 2 セットを、第 2 クライアントデバイスから受信した第 2 設計定義によって定義された第 2 設計に対して割り当てるように構成されており、マクロコンポーネントの前記第 1 セット内のマクロコンポーネントの第 1 サブセットは、マクロコンポーネントの前記第 2 セット内のマクロコンポーネントの第 2 サブセットとして、前記 FPGA デバイスのセットのうち同じ FPGA デバイスに配置される、

請求項 1 のマクロスケジューラ。

【請求項 9】

マクロコンポーネントの前記第 1 セットは、前記複数の FPGA デバイスのうち少なくとも 2 つの異なる FPGA デバイスのマクロコンポーネントを含む、

請求項 1 のマクロスケジューラ。

10

20

30

40

50

【請求項 1 0】

フィールドプログラマブルゲートアレイ（FPGA）デバイスのセットの複数のマクロコンポーネントを列挙するデータベースを更新することと、

設計のための1つ以上の指定されたマクロコンポーネントを示す第1設計定義を第1クライアントデバイスから受信することと、

前記第1設計定義に示された前記1つ以上の指定されたマクロコンポーネント毎に前記複数のマクロコンポーネントのうち1つを割り当てることによって、前記設計のためのマクロコンポーネントの第1セットを割り当てることと、

割り当てられたマクロコンポーネントの前記第1セットを前記第1設計定義に従って構成することによって、前記FPGAデバイスのセットに前記設計を実装することと、を含む、

10

方法。

【請求項 1 1】

前記1つ以上の指定されたマクロコンポーネントは、1つ以上の指定されたタイルを含み、

前記第1設計定義は、前記1つ以上の指定されたタイルの各々のビットファイルをさらに示し、

前記方法は、

前記1つ以上の指定されたタイル毎に、マクロコンポーネントの前記第1セット内の割り当てられたタイルを、前記指定されたタイルの前記ビットファイルに基づいてプログラムすることを含む、

20

請求項10の方法。

【請求項 1 2】

マクロコンポーネントの前記第1セットの各々を、前記第1クライアントデバイスのネットワークアドレスと、前記第1クライアントデバイスによって要求された第1タスクと関連付けることをさらに含む、

請求項10の方法。

【請求項 1 3】

前記複数のマクロコンポーネント毎に、前記FPGAデバイスのセットにおける前記マクロコンポーネントの位置及び前記マクロコンポーネントの利用可能性を前記データベースに記録することをさらに含む、

30

請求項10の方法。

【請求項 1 4】

前記FPGAデバイスのセットの初期構成であって、前記第1設計定義によって示される初期構成におけるタスクの実行中に、前記設計のための最適化された構成を生成することと、

前記FPGAデバイスのセットの前記最適化された構成でタスクの実行を再開する前に前記初期構成を前記最適化された構成に置き換えるように、前記FPGAデバイスのセットを再プログラムすることと、

前記初期構成における初期ポートの位置を、前記最適化された構成における最適化されたポートの位置と相関させる再マッピング通知を生成することと、をさらに含む、

40

請求項10の方法。

【請求項 1 5】

前記第1設計定義は、アクセラレータの定義を含み、

前記方法は、

第1期間中に前記第1クライアントデバイスからの第1タスクのアクセラレータにおける実行をスケジューリングすることと、

第2期間中に第2クライアントデバイスからの第2タスクのアクセラレータにおける実行をスケジューリングすることと、をさらに含む、

請求項10の方法。

50

【請求項 16】

前記複数のマクロコンポーネントの第2セットを、第2クライアントデバイスから受信した第2設計定義によって定義された第2設計に対して割り当てることであって、マクロコンポーネントの前記第1セット内のマクロコンポーネントの第1サブセットは、マクロコンポーネントの前記第2セット内のマクロコンポーネントの第2サブセットとして、前記FPGAデバイスのセットのうち同じFPGAデバイスに配置される、ことをさらに含む、

請求項10の方法。

【請求項 17】

複数のマクロコンポーネントを含むフィールドプログラマブルゲートアレイ（FPGA）デバイスのセットと、

前記複数のマクロコンポーネントを列挙するように構成されたデータベースと、
スタンドアロンマクロスケジューラと、を備え、

前記スタンドアロンマクロスケジューラは、

前記データベースを更新するように構成されたリソース追跡モジュールと、

設計のための1つ以上の指定されたマクロコンポーネントを示す第1設計定義を第1クライアントデバイスから受信するように構成された通信インタフェースと、

前記第1設計定義に示された前記1つ以上の指定されたマクロコンポーネント毎に前記複数のマクロコンポーネントのうち1つを割り当てることによって、前記設計のためのマクロコンポーネントの第1セットを割り当てるように構成されたリソース割り当てロジックと、

割り当てられたマクロコンポーネントの前記第1セットを前記第1設計定義に従って構成することによって、前記FPGAデバイスのセットに前記設計を実装するように構成された構成ロジックと、を備える、

システム。

【請求項 18】

第1クライアントデバイスをさらに備え、

前記スタンドアロンマクロスケジューラは、

前記第1クライアントデバイスからのコマンドに応じて、前記リソース割り当てロジックによる割り当て、及び、前記構成ロジックによる構成を開始するように構成されたアプリケーションプログラミングインタフェース（API）をさらに備える、

請求項17のシステム。

【請求項 19】

前記FPGAデバイスのセットのうち1つ以上のFPGAデバイスの各々は、前記1つ以上のFPGAデバイス内の前記割り当てられたマクロコンポーネントのうち少なくとも1つの配置を決定するように構成されたローカルマクロスケジューラを備える、

請求項17のシステム。

【請求項 20】

前記FPGAデバイスのセットを前記スタンドアロンマクロスケジューラに接続する複数のネットワークチャンネルをさらに備える、

請求項17のシステム。

【発明の詳細な説明】

【技術分野】

【0001】

（政府の権利）

本発明は、DOEから与えられたLawrence Livermore National Security（主契約番号DE-AC52-07NA27344、サブ契約番号B620717）のPathForward Projectの下で政府の支援を受けてなされたものである。政府は、本発明において一定の権利を有する。

【背景技術】

【0002】

フィールドプログラマブルゲートアレイ（FPGA）は、製造後にエンドユーザが構成可能に設計された集積回路デバイスである。一般に、FPGAは、プログラマブル相互接続ファブリックを構成する相互接続リソースによって互いに接続される、ロジックブロックと呼ばれる非コミットの回路要素のアレイを含む。相互接続リソースは、例えば、異なる構成で相互配線可能なロジックゲートを含むことができる。FPGAのロジックブロックは、複雑な組み合わせ関数、又は、ANDやXOR等のロジック関数を実行するように構成することができる。FPGAのロジックブロックは、単純なフリップフロップ又はより完全なメモリブロックとして実装することができるメモリ素子を含むこともできる。

【0003】

FPGAでは、ロジックブロックは、ルックアップテーブル（LUT）や、小さなスタティックランダムアクセスメモリ（SRAM）又はレジスタに値を挿入することによってプログラムされる他の固定関数と等の要素を含むことができる。プログラミングは、ロジックブロックを使用する前に行う必要がある。プログラミング後、LUTへの関数入力各組み合わせによって、所定の出力が得られ、任意のロジック関数の実装が可能になる。典型的なFPGAは、外部メモリデバイスと、構成用の単一の比較的低速なシリアルプログラミングポートと、に依存する。FPGAは、通常、「構成モード」と、これとは別の「ユーザモード」とを有し、チップ全体がプログラミングされるのを待機しているアイドル状態、又は、プログラムされたチップが動作中の状態の何れかにある。

【0004】

本開示は、添付の図面において、限定ではなく一例として示される。

【図面の簡単な説明】

【0005】

【図1】フィールドプログラマブルゲートアレイ（FPGA）デバイスのセットにおいてハードウェアリソースの仮想化をサポートするコンピューティングシステムの実施形態を示す図である。

【図2】一実施形態による、FPGAデバイス内のマクロコンポーネントを示す図である。

【図3】一実施形態による、FPGA内のプログラマブルタイルを示す図である。

【図4】マクロスケジューラの実施形態を示す図である。

【図5】一実施形態による、FPGAデバイスを構成し、マクロスケジューリング機能を行うプロセスを示すフロー図である。

【発明を実施するための形態】

【0006】

以下の説明では、実施形態の十分な理解をもたらすために、特定のシステム、コンポーネント、方法等の例等の多数の具体的な詳細を説明する。しかしながら、当業者には明らかであるように、少なくともいくつかの実施形態は、これらの具体的な詳細なしに実施されてもよい。他の例では、実施形態を不要に曖昧にするのを避けるために、周知のコンポーネント又は方法は、詳細に説明されないか、単純なブロック図形式で提示される。したがって、説明される具体的な詳細は、単なる例示である。特定の实装は、これらの例示的な詳細と異なってもよく、それでも実施形態の範囲内であると考えられる。

【0007】

最近のデータセンタは、クライアントワークロードを実行するための固定リソース又はユーザ構成可能リソースとして使用可能なフィールドプログラマブルゲートアレイ（FPGA）デバイスを含む様々なコンピューティングリソースをクライアントに提供する。いくつかのFPGAデバイスは、このようなデータセンタで用いられるために特別に設計され得る。しかしながら、FPGAデバイスを用いる既存のデータセンタは、特に複数の独立クライアントがリソースの使用を要求している場合、FPGAのハードウェアリソースを非効率的に割り当てる可能性がある。したがって、このようなデータセンタは、FPGAの仮想化を可能にするインターフェースから利益を受けることができ、これにより、リソー

ス共有、時分割多重化及び他の最適化を介して複数のクライアントにサービスを提供する際に、FPGAのハードウェアリソースをより効率的に割り当てることができる。

【0008】

一実施形態では、データセンタは、そのFPGAデバイスの仮想化を、FPGAのハードウェアリソースをマクロコンポーネントと呼ばれるロジックユニットに編成することによってサポートし、アクセラレータ設計は、マクロコンポーネント間の接続を定義するマクログラフとして指定することができる。マクロコンポーネントの使用をスケジューリングするための1つ以上のFPGAマクロスケジューラは、FPGAデバイス自体に組み込まれている、及び／又は、ネットワーク若しくはシステム相互接続を介してFPGAデバイスに接続されたスタンドアロンユニットとして動作している。

10

【0009】

一実施形態では、FPGAデバイスは、入出力(I/O)ポート、構成可能ロジックブロック(CLB)領域、強化ブロック又は回路コンポーネントの他の組み合わせ等のハードウェアリソースを表す、いくつかの異なるタイプのマクロコンポーネントを含む。これらは、特定用途向けハードウェアアクセラレータの機能を実装するために、FPGAのハードウェアリソースの大まかな割り当て、配置及びルーティングに使用することができる。したがって、マクロコンポーネントは、データセンタ内の共有FPGAデバイスのロジック配置の単位として機能する。マクロコンポーネントのスケジューリングは、要求に応じてFPGAで完全な合成、配置及びルーティングプロセスを行うよりも遥かに簡単である(例えば、最初のフリーなマクロコンポーネントを要求に割り当てることができる)。より複雑なプリコンパイル回路を含むマクロブロックに対して、合成、配置及びルーティングをオフラインで行うことができる。

20

【0010】

一実施形態では、FPGAデバイスの各々は、ローカルFPGAマクロスケジューラを含み、これが組み込まれたFPGAデバイスに対する仮想化リソースの割り当て、スケジューリング及びハードウェアの最適化を行う。また、データセンタは、システム又はメモリ相互接続上の他のコンポーネントにネットワーク接続されたスタンドアロンのFPGAマクロスケジューラユニットを含む。スタンドアロンのFPGAマクロスケジューラは、ローカルマクロスケジューラを有しないネットワーク内のFPGAのマクロスケジューリング及び管理タスク専用のより強力なコンピューティングリソースを提供する。スタンドアロンのFPGAマクロスケジューラは、ローカルマクロスケジューラが計算能力において制限されるか、より広範なクライアントによって要求されるタスクのより制限されたビューを有する場合があるので、ローカルマクロスケジューラを既に有するFPGAデバイスに対してさらなる最適化及び管理タスクを実行することができる。

30

【0011】

一実施形態では、ローカル及び／又はスタンドアロンマクロスケジューラは、1つ以上のFPGAデバイスに実装されるアクセラレータの設計定義に従ってマクロコンポーネントを割り当て及びスケジューリングするアプリケーションプログラミングインタフェース(API)を提供する。設計定義は、マクロコンポーネント間の接続及びマクロコンポーネントの構成によってアクセラレータが定義される、マクログラフを指定するコードを含む。このように、ユーザは、APIを使用することにより、ハードウェアアクセラレータに実装される関数を生成することができ、さらに、関数の入力データを割り当て、仮想化された方法で結果の出力を読み出すことができる。

40

【0012】

図1は、一実施形態による、タスクをスケジューリングし、複数のFPGAデバイス121~123のリソースを割り当てするためのマクロスケジューラ130~133を含むコンピューティングシステム100を示す図である。コンピューティングシステム100のマクロスケジューラ130~133は、FPGAデバイス121~123に実装されたFPGAアクセラレータの仮想化を有効にする。一実施形態では、コンピューティングシステム100は、例えばクライアント110、111等の複数のクライアントデバイスにコ

50

ンピューティングリソースを提供するデータセンタの一部を表す。コンピューティングシステム100のコンポーネントは、システム、メモリ相互接続、又は、ローカル若しくは広域ネットワーク等の様々なチャネルを介して互いに通信することができ、図示したもの以外のトポロジで接続することができる。

【0013】

コンピューティングシステム100では、FPGAデバイス121～123の各々は、FPGAに実装される同時アクセラレータ (concurrent accelerators) を配置 (すなわち、物理的にマッピング) 及び最適化するためのローカルマクロスケジューラ131～133を有する。例えば、マクロスケジューラ131は、アクセラレータの集合をFPGA121に同時に実装するためにFPGA121上の何れの物理的なハードウェアリソースを用いるかを決定する。また、コンピューティングシステム100は、グローバル (すなわち、クロスFPGA) 割り当て及びスケジューリングを行うことが可能なスタンドアロンマクロスケジューラ130を含む。スタンドアロンマクロスケジューラ130は、複数のクライアント (例えば、110, 111) からの要求にアクセスし、複数のFPGAデバイス (例えば、121～123) のリソースを追跡するので、より多くの最適な配置及びスケジュールを識別することができる。さらに、スタンドアロンマクロスケジューラ130は、FPGAデバイス121～123とは別の場所で動作するので、FPGAデバイス121～123の熱的制約に制限されることなく動作することができる。したがって、マクロスケジューラ130は、より高速でよりコンパクトな実装のために、同時にスケジューリング (co-scheduled) されたレジスタ転送レベル (RTL) の設計を合成することの可能な、より強力なコンピュータとなり得る。

【0014】

一実施形態では、クライアント110, 111は、ネットワーク接続を介してコンピューティングシステム100の他のコンポーネントに接続されたリモートクライアントである。クライアント110, 111は、特定のハードウェアアクセラレータを実装するようにFPGAデバイス121～123の構成を要求することができ、ハードウェアアクセラレータで実行されるタスクを定義することもできる。クライアント110, 111は、設計定義、タスク定義及び他の情報 (例えば、構成ビットファイル) を、スタンドアロンマクロスケジューラ130において、又は、ローカルマクロスケジューラ131～133において、APIに提供する。マクロスケジューラ130～133は、クライアントの要求に応じて、FPGA121～123のハードウェアリソースを割り当て、タスクの実行をスケジューリングする。一実施形態では、アクセスキー、暗号化通信、及び／又は、他のセキュリティ手段を用いて、悪意のあるクライアント又はプロセスが他のタスクに影響を及ぼさないようにする。

【0015】

FPGA121～123は、プログラム可能な相互接続要素によって接続することができる構成可能ロジックブロック (CLB) と呼ばれる非コミットの回路要素を含む。各FPGA内のCLBは、ルックアップテーブルと、FPGA内の特定のスタティックランダムアクセスメモリ (SRAM) 位置又はレジスタに値を挿入することによって構成可能な他の固定関数と、を含む。したがって、CLBを、クライアントデバイスによって指定された任意のロジック関数を行うように構成することができる。

【0016】

コンピューティングシステム100内のマクロスケジューラ130～133は、要求されたアクセラレータ構成を実装するためにFPGA121～123のハードウェアリソースを割り当て、要求されたタスクをアクセラレータ内で実行するためにスケジューリングする機能を実行する。また、マクロスケジューラ130～133は、タスクと構成との間の切替え (すなわち、ビットストリームのスワッピング) を可能にするコンテキストスイッチを実行する。例えば、コンテキストスイッチは、構成された領域 (例えば、構成されたマクロコンポーネントのセットを含む) のレジスタ及びメモリ状態を保存することと、以前に保存された状態を同じ領域に復元することと、異なるタスクを実行するために領域

を再構成することと、を伴うことができる。以前に保存した状態を復元して、後で元のタスクの実行を再開することができる。一実施形態では、十分なコンピューティングリソースを有するマクロスケジューラは、合成並びに／又は配置及びルーティングプロセスを実行してより最適な構成を生成した後に実行を一時停止して、元の構成を新しい最適な構成に置き換える第2パス最適化を行う。

【0017】

コンピューティングシステム100は、ローカルマクロスケジューラ131～133と、スタンドアロンマクロスケジューラ130と、の両方を含む。代替の実施形態は、より少ない又はより多いローカルマクロスケジューラ及び／若しくはスタンドアロンマクロスケジューラを含むことができる。ローカルマクロスケジューラ131～133は、それらが存在するFPGAデバイスに対して上述した機能を行う。例えば、マクロスケジューラ131～133は、各々のFPGAデバイス121～123における割り当てられたマクロコンポーネントの配置を決定する。スタンドアロンマクロスケジューラ130は、ネットワークチャネル（例えば、システム又はメモリ相互接続）を介して複数のFPGAデバイス121～123に接続されており、接続されたFPGAデバイス121～123のうち何れかに対して割り当て、スケジューリング及び最適化機能を実行することができる。一実施形態では、スタンドアロンマクロスケジューラ130は、独自のローカルマクロスケジューラを有しないFPGAデバイスに対してこれらの機能を実行することができる。スタンドアロンマクロスケジューラ130は、クライアント110、111からの関数呼び出しを受信し、当該関数呼び出しに基づいて、要求されたタスクを実行するためにマクロコンポーネントを割り当て、クライアント110、111から受信した設計の合成、配置及びルーティングを実行する。また、マクロスケジューラ130は、パーミッションを管理し、クライアントに通知を発行する。

【0018】

図2は、複数のマクロコンポーネント210～219の形態のハードウェアリソースを含むFPGAデバイス121を示す図である。マクロコンポーネントは、割り当て及びスケジューリングの目的で単一のユニットとして扱われる小さなハードウェアコンポーネントの集合を含む。マクロコンポーネントには、高度に構成可能なものもあれば、単一の固定関数を実行するものもある。FPGA121内のマクロコンポーネント210～219は、プログラマブルタイル210と、入力レジスタ211と、出力レジスタ212と、固定関数ブロック213～216と、物理入出力（I/O）ブロック217～219と、を含む。

【0019】

タイル210の各々は、CLB、スイッチ、固定関数ブロック、及び、他のタイルから独立してプログラムすることができる相互接続を含むプログラミング領域である。タイルは、ビットファイルに関連付けることができる。ビットファイルは、タイルの合成、配置、ルーティングされた設計を含むファイルである。関数のサイズ又は複雑さに応じて、マクロスケジューラ130又は131は、関数を実装するために複数の物理タイルを割り当てることができる。

【0020】

また、FPGA121は、入力レジスタ211及び出力レジスタ212のセットを含む。入力レジスタ211は、FPGA121に実装されている関数に入力データを提供し、出力レジスタ212は、当該関数から結果として得られる出力データを提供する。一実施形態では、入力レジスタ211及び出力レジスタ212は、より高度なパフォーマンスのためにキュー（例えば、先入れ先出し（FIFO）バッファ）として実装される。FPGA121は、入力レジスタ211及び出力レジスタ212が物理入出力（I/O）ブロック（例えば、PHYブロック217～219）を介してFPGAの外部ピン（例えば、ピン227～229）に接続されることによって、外部デバイス及びプロセスにアクセス可能に構成することができる。PHYブロック217～219は、トランシーバ等のI/Oリソースへのアクセスを提供し、FPGAのピン227～229に接続される。

【0021】

FPGA121は、異なるタイプの固定関数（FF）ブロック213～216を含む。FFブロック213は乗算器ブロックであり、FFブロック214は除算器ブロックであり、FFブロック215はメモリブロックであり、FFブロック216は中央処理装置ブロックである。FPGA121は、固定関数ブロックの各タイプの複数のインスタンスを含むことができる。FPGA121の代替の実施形態は、異なる関数を実行する他のタイプのFFブロックを含むことができる。

【0022】

図3は、一実施形態による、FPGA121内のタイル210の内部コンポーネントを示す図である。タイル210は、構成可能ロジックブロック（すなわち、ブロック301～312）のアレイと、ロジックブロック301～312を選択的に接続するのに用いられる相互接続リソース320～322と、を含むプログラミング領域である。

10

【0023】

FPGA121では、タイル210の境界が破線で示されている。タイル331はロジックブロック301～304を含み、タイル332はロジックブロック305～308を含み、タイル333はロジックブロック309、310を含み、タイル334はロジックブロック311、312を含む。タイル331～334は、FPGA121に含まれるプログラマブルロジックを別々の構成ドメインに分割する。別々の構成ドメインは、互いに独立して動作及びプログラムすることができ、又は、より大きなプログラマブル領域として動作するように組み合わせることができる。1つ以上の命令を実行するために異なるタイルが使用されている間に、1つのタイル内のハードウェアの再構成を実行することができる。

20

【0024】

図3に示すように、タイル331、332の各々は、4つの構成可能ロジックブロックを含み、タイル333、334の各々は、2つの構成可能ロジックブロックを含む。したがって、所定のタイルは、同じプログラマブルデバイス内の別のタイルと同数の構成可能ロジックブロックを有する必要がある。代替の実施形態では、タイルは、図示した数よりも少ない数又は多い数の構成可能ロジックブロックを含むことができる。

【0025】

構成可能ロジックブロック301～312の各々は、ルックアップテーブル（LUT）と、固定関数ブロックと、を含む。プログラミング後、LUTへの関数入力の各組み合わせは、所定の出力をもたらす、任意のロジック関数の実装を可能にする。構成可能ロジックブロックの各々は、適切な値を、LUT及びロジックブロック内のコンポーネント（例えば、LUT、固定関数ブロック等）間の接続を制御する構成メモリ領域又はレジスタに書き込むことによって、様々な機能を実行するようにプログラムすることができる。

30

【0026】

構成可能ロジックブロック301～312の各々は、プログラマブル相互接続ファブリック320に接続されており、プログラマブル相互接続ファブリック320は、ブロック301～312が互いに選択的に接続されることを可能とする導電トレース及びスイッチを含む。図3には、構成可能ロジック301をプログラマブル相互接続ファブリック320に接続する導電トレースが示されている。他のブロック302～312も、導電トレースを介して相互接続ファブリック320に接続される。しかしながら、これらのブロック302～312を相互接続ファブリック320に接続する導電トレースは、明瞭にするために図から省略されている。

40

【0027】

代替の実施形態では、プログラマブル相互接続ファブリック320は、図3に示すよりも均一でなくてもよい。例えば、相互接続ファブリックは、異なる長さを有する導電トレースを含んでもよいし、FPGA121の異なる領域において異なる数の並列導電トレースを有してもよい。相互接続ファブリック320の導電トレースはスイッチによって互いに接続され、スイッチでは導電トレースが交差している。例えば、複数のトレースが交差

50

する領域 3 2 1（点線で示される）は、2つの導電トレース間の各交差部間にスイッチを含む。代替として、スイッチは、領域 3 2 1 内の交差部の一部のみに存在してもよい。

【0028】

また、プログラマブル相互接続ファブリックは、スイッチ 3 2 2 等の境界スイッチを含み、これを用いて、一方のタイル内のコンポーネントを他方のタイルから選択的に分離することができる。境界スイッチは、FPGA 1 2 1 内のタイル 3 3 1～3 3 4 の境界に位置する。例えば、境界スイッチ 3 2 2 は、プログラマブル相互接続ファブリック 3 2 0 の導電トレースがタイル 3 3 1 の境界と交差する場所に位置する。

【0029】

したがって、境界スイッチ 3 2 2 を使用して、構成可能ロジックブロック 3 0 1～3 0 4 及びタイル 3 3 1 の他のコンポーネントを他のタイル 3 3 2～3 3 4 から遮断することができる。また、タイルを組み合わせるために、境界スイッチを選択的に閉じたり開いたりすることができる。例えば、タイル 3 3 1, 3 3 2 間の境界スイッチ 3 2 2 は、タイル 3 3 1, 3 3 2 を互いに組み合わせるために閉じることができ、一方、他の境界スイッチは、組み合わせられたタイル 3 3 1, 3 3 2 を他のタイル 3 3 3, 3 3 4 から隔離するために開いたままとすることができる。

【0030】

図 4 は、一実施形態による、スタンドアロンマクロスケジューラ 1 3 0 の機能ブロック図である。マクロスケジューラ 1 3 0 に含まれる機能ブロックは、マクロスケジューラ 1 3 0 で実行可能な動作を表し、回路モジュール等のハードウェアを用いて、又は、ハードウェアとソフトウェアとの組み合わせを用いて実装される。マクロスケジューラ 1 3 0 は、システム及び／又はメモリ相互接続等のネットワークチャネルを介して、FPGA デバイス 1 2 1～1 2 3 及びクライアントデバイス 1 1 0, 1 1 1 等のコンピューティングシステム 1 0 0 内の他のコンポーネントに接続される。

【0031】

マクロスケジューラ 1 3 0 は、通信インタフェース 4 1 0 を介してネットワークチャネルに接続される。アプリケーションプログラミングインタフェース (API) モジュール 4 1 1 は、クライアント 1 1 0, 1 1 1 から通信インタフェース 4 1 0 を介して設計定義、関数呼び出し及び他のコマンドを受信し、リソース割り当てロジック 4 1 3 によるマクロコンポーネントの割り当て、構成ロジック 4 1 4 によるタイルの構成等のマクロスケジューラ 1 3 0 の要求された関数を呼び出すことによって、これらの入力に応答する。

【0032】

クライアントデバイス 1 1 0～1 1 1 によって要求された特定のタスクに対するハードウェア設計は、一般に、任意の単一の FPGA 上の全てのハードウェアリソースを消費するわけではない。したがって、FPGA の余ったハードウェアリソースは、同じプロセス、同じクライアントからの他のプロセス、又は、他のクライアントからのプロセスによって要求された他のアクセラレーションされた関数に割り当てることができる。複数の設計が複数の同時設計のために共有 FPGA デバイス上で迅速にスケジューリングされる（すなわち、「配置される」）のを可能にするために、設計定義は、ハードウェア構成を、レジスタ転送レベル (RTL) よりも高い抽象化レベルで記述する。

【0033】

一実施形態では、API モジュール 4 1 1 が受信した設計定義は、設計に関するマクログラフにおいて、設計のための 1 つ以上の指定されたマクロコンポーネントを示す。マクログラフは、使用されているマクロコンポーネントのタイプと、指定されたマクロコンポーネントの構成及び接続方法と、を示す。マクログラフは、入出力レジスタ、CLB（例えば、タイル）の集合、ルックアップテーブル、ルーティングリソース、固定関数ブロック（例えば、乗算器、除算器、メモリ、CPU 等）等のマクロコンポーネントを含むことができる。

【0034】

また、マクログラフは、ビットファイルのファイル名を、マクログラフで指定された各

10

20

30

40

50

タイルに関連付ける。一実施形態では、ビットファイルは、クライアントデバイス110～111のうち1つによって設計に対して実行された合成プロセスによって生成される。ビットファイルは、タイルが設計の少なくとも一部（例えば、1つ以上のハードウェアアクセラレータ）を実装するように構成するために、構成レジスタに記憶され得る構成データを含む。

【0035】

以下の擬似コードは、一実施形態による、マクログラフを含む設計定義の例である。

【数1】

```
myAccelerator = new MGCreateGraph();

inA = new MGInReg();
inB = new MGInReg();
outC = new MGOutReg();
myLogic = new MGTile( "tile_file.bit" );
myLogic.setInputs( inA, inB );
myLogic.setOutputs( outC );
myAccelerator.addComponent( myLogic );
```

10

上記コードにおいて、「myAccelerator」は、4つのマクロコンポーネント（すなわち、入力レジスタ「inA」及び「inB」、出力レジスタ「outC」、並びに、タイル「myLogic」）を含むマクログラフである。タイル「myLogic」は、タイルをプログラミングするための構成データを含むビットファイル「tile_file.bit」を示すコンストラクタ「MGTile（）」を用いてインスタンス化される。「myLogic.setInputs（）」関数は、タイル「myLogic」を入力レジスタ「inA」及び「inB」に接続する。「myLogic.setOutput（）」関数は、「myLogic」を出力レジスタ「outC」に接続する。最後に、「myAccelerator.addComponent（）」関数を用いて、タイル「myLogic」を、アクセラレータのマクログラフ「myAccelerator」に加える。

20

【0036】

設計定義は、クライアントデバイス110～111のうち1つから受信したタスク定義によって参照することができる。タスク定義は、APIモジュール411によって受信され、設計定義で指定されたマクログラフを用いて実行されるタスクを定義する。例示的なタスク定義を以下に示す。

30

【数2】

```
myFPGA = getFreeFPGAaccelerator();
myFPGA.program( myAccelerator );
inA.writeDataSequence( 1, 2, 3, 4, 5 );
inB.writeDataSequence( 6, 7, 8, 9, 10 );
myFPGA.execute();
myResults = outC.read();
```

40

「getFreeFPGAaccelerator（）」関数は、未使用のFPGAを「myFPGA」変数に返す。次の関数呼び出し「myFPGA.program（myAccelerator）」は、例示的な設計定義で先に定義されているように、返されたFPGAのプログラミングを「myAccelerator」マクログラフに基づいて呼び出す。次の関数「inA.writeDataSequence（1, 2, 3, 4, 5）」及び「inB.writeDataSequence（6, 7, 8, 9, 10）」は、入力バッファ「inA」及び「inB」に対する入力データを指定する。代替の実施

50

形態では、（例えば、FPGA上の）メモリへのポインタを、値のアレイではなく入力バッファに記憶することができる。関数呼び出し「myFPGA.execute()」は、構成されたアクセラレータにおいてタスクの実行を開始し、「myResults=outC.read()」は、出力レジスタ「outC」から実行結果を読み出す。したがって、タスク定義は、タスクに使用されるマクログラフを指定し、入力データを指定し、実行を要求し、結果として得られる出力データの読み出しを要求する。

【0037】

マクロスケジューラ130は、リソース追跡モジュール412によって更新され、FPGAデバイス121～123内のマクロコンポーネントを列挙するデータベース420を含む。一実施形態では、データベース420は、マクロコンポーネントの一意の識別子を「マクロ」フィールド421に記憶し、マクロコンポーネントの位置を「位置」フィールド422に記憶する、マクロコンポーネント毎の行を含む。一実施形態では、「位置」フィールドは、マクロコンポーネントが存在するFPGAの識別子及び／又はマクロコンポーネントの構成メモリのアドレスを記憶する。クライアントデバイス110、111の一方が特定のマクロコンポーネントを使用する設計の実装を要求する場合、リソース追跡モジュール412は、クライアントのネットワークアドレスを「クライアントIP」フィールド423に記憶し、PIDを「PID」フィールド424に記憶することによって、クライアントのネットワークアドレス（例えば、インターネットプロトコル（IP）アドレス）及びオペレーティングシステム（OS）プロセス識別子（PID）をマクロコンポーネントに関連付けるようにデータベース420を更新する。タスクの識別子を「タスク」フィールド425に書き込むことによって、マクロを使用して実行されているタスクをマクロに関連付けることもできる。一実施形態では、データベース420は、スケジュール情報を記録する「スケジュール」フィールド426を含む。例えば、「スケジュール」フィールド426は、マクロコンポーネントがフィールド423～425のクライアント、PID及びタスクの各々に割り当てられる期間を示すことができる。また、データベース420は、優先順位の指標を記憶するのに用いられる「優先順位」フィールド427を含む。一実施形態では、優先順位の指標は、ユーザが選択可能であり、タスクの優先順位に応じたスケジューリングに影響を与える方法をユーザに提供する。例えば、ユーザは、タスクに対してより高い優先順位を選択することができ、そのタスクは、既に進行中のより低い優先順位の他のタスクを中断することができる。

【0038】

一実施形態では、複数のマクロコンポーネントを、同じクライアント及び／又は同じタスクに関連付けることができる。代替の実施形態では、データベース420は、使用コスト、コンテキスト識別子、又は、マクロコンポーネントの割り当て、スケジューリング若しくはプログラミングに関連する他の値を記録する追加フィールドを含む。これらの追加フィールドも、リソース追跡モジュール412によって更新することができる。

【0039】

マクロスケジューラ130は、API411で受信したクライアント要求に応じてマクロコンポーネントを割り当てるリソース割り当てモジュール413を含む。リソース割り当てモジュール413は、設計定義に応じて、設計定義において指定されたマクロコンポーネント毎に、同じタイプの利用可能なマクロコンポーネントをデータベース420にクエリし、設計で使用するのに利用可能なマクロコンポーネントを割り当てることによって、設計のためのマクロコンポーネントのセットを割り当てる。先の例を続けると、設計定義は、入力レジスタ「inA」及び「inB」、出力レジスタ「outC」、並びに、タイル「myLogic」を指定する。したがって、リソース割り当てロジック413は、データベース420にクエリして、2つの入力レジスタ、出力レジスタ及びタイルを見つけ、「myAccelerator」を実装するために、これらのマクロコンポーネントを割り当てる。

【0040】

一実施形態では、マクロコンポーネントは、タスク定義に従って「myFPGA.pr

10

20

30

40

50

ogram (myAccelerator)」を実行するときに設計に割り当てられる。一実施形態では、マクロコンポーネントの経時的な利用可能性が、データベース420内の「スケジュール」フィールド426によって示される。要求された設計を実施するのに十分なマクロコンポーネントが利用可能でない場合、「スケジュール」フィールド426に基づいて決定されるように、十分なマクロコンポーネントが利用可能になった後の時点でタスクをスケジューリングすることができる。

【0041】

したがって、マクロスケジューラ130は、API411を介してクライアントデバイスから受信した設計及びタスク定義に応じて、FPGAリソースのオンデマンド高速割り当てを実行する。一実施形態では、マクロスケジューラ130内のリソース割り当てモジュール413は、割り当てられたマクロコンポーネントを構成するタイミング、及び、構成されたマクロコンポーネントにおいて要求されたタスクを実行するタイミングを制御することによって、スケジューリング機能を実行する。一実施形態では、リソース割り当てモジュール413は、コンテキストスイッチ及び管理機能も実行する。したがって、リソース割り当てモジュール413は、タスクを一時停止し、タスクによって使用されている任意のアクセラレータのレジスタ及びメモリ状態を、それらの構成データ（例えば、ビットファイル又はビットストリーム）と共に保存して、タスクの実行を後の時点で再開することができる。このコンテキストスイッチにより、ハードウェアリソースの時分割多重化が可能になる（例えば、実行中のタスクを、優先順位の高いタスクによって一時的に中断することが可能になる）。

【0042】

マクロスケジューラ130は、複数のクライアント110～111及び複数のFPGAデバイス121～123のマクロコンポーネントからのタスクに関する情報にアクセスできるので、相互に通信又は協働しないクライアントデバイスに対しては識別されない最適化の機会を識別することができる。一実施形態では、マクロスケジューラ130は、同時に実行されるタスクに関するリソースの使用を最適化する。

【0043】

一実施形態では、マクロスケジューラ130は、単一の構成されたアクセラレータを、同じクライアントデバイス又は異なるクライアントデバイスから開始された2つの異なるタスク間で共有することを可能にする。例えば、第1クライアントデバイス110は、第1タスクを実行するための第1設計定義を送信することができ、この設計定義は、FPGAデバイス121～123内のマクロコンポーネントから実装されるアクセラレータの定義を含む。クライアント111からの第2タスクを実行するための第2設計定義が同じアクセラレータ設計の使用を要求する場合、リソース割り当てロジック413は、第1タスク及び第2タスクの両方によって使用される単一のアクセラレータを構成することができる。リソース割り当てロジック413は、第1期間中にアクセラレータにおける第1タスクの実行をスケジューリングし、第1期間と重複しない第2期間中に第2タスクの実行をスケジューリングすることによって、アクセラレータの使用を経時的に多重化する。

【0044】

一実施形態では、マクロスケジューラ130は、単一のFPGAデバイスのハードウェアリソース（例えば、マクロスケジューラ）を、同じ又は異なるクライアントデバイスからの2つの異なる設計間で共有することを可能にする。特に、リソース割り当てロジック413は、第1クライアント（例えば、クライアント110）が要求する第1設計に対してFPGAデバイスからのマクロコンポーネントを割り当て、第2クライアント（例えば、クライアント111）が要求する第2設計に対して同じFPGAデバイスからのマクロコンポーネントを割り当てる。したがって、単一のFPGAデバイスの異なるマクロコンポーネントを複数の設計間で共有することができる。一実施形態では、リソース割り当てロジック413は、単一のマクロコンポーネントの一部を異なる設計間で割り当てることもできる。例えば、2つの異なる設計の各々は、メモリマクロコンポーネントの半分未満を使用してもよい。これにより、リソース割り当てロジック413は、第1設計がメモリ

の上位部分を利用し、第2設計がメモリの下位部分を利用して、2つの設計間で共有される単一のメモリマクロコンポーネントを割り当てることができる。

【0045】

一実施形態では、マクロスケジューラ130は、単一のアクセラレータで使用される複数のFPGAデバイスからのマクロコンポーネントを割り当てることができる。すなわち、アクセラレータに割り当てられたマクロコンポーネントのセットは、2つ以上のFPGAデバイスからのマクロコンポーネントを含むことができる。したがって、アクセラレータは、単一のFPGAが、アクセラレータ設計を収容するのに十分に利用可能なマクロコンポーネントを持たない場合でも、複数のFPGAデバイスに実装可能である。

【0046】

マクロコンポーネントがリソース割り当てモジュール413によって割り当てられると、構成ロジック414は、割り当てられたマクロコンポーネントのセットを設計定義にしたがって構成することによって、FPGAデバイスにおける設計を実装する。構成ロジック414は、設計定義で指定されたタイル毎に、指定されたタイルの設計定義で指定されたビットファイルを用いて、対応する割り当てられたタイルをプログラムする。一実施形態では、構成ロジック414は、構成ポート415を介してFPGAデバイスをプログラムする。或いは、FPGAデバイスのプログラミングは、通信インタフェース410を介して呼び出されてもよい。一実施形態では、構成データ（例えば、ビットファイル又はビットストリーム）は、通信インタフェース410を介して、プログラムされるFPGAデバイス121～123のローカルマクロスケジューラ（例えば、131～133）に送信されてもよく、マクロコンポーネントのプログラミングは、各FPGAデバイスの各々のローカルマクロスケジューラによって実行される。

【0047】

マクロスケジューラ130は、クライアント110、111等のクライアントデバイスから受信した設計に対して合成、配置及びルーティング、又は、他のプロセスを行うために使用される合成ロジック416を含む。一実施形態では、クライアントデバイス内ではなく、マクロスケジューラ130の合成ロジック416内でRTL設計に対して合成を実行することができる。

【0048】

また、合成ロジック416は、設計が既に構成され、FPGAデバイス121～123のうち1つ以上で動作している間に、マクロスケジューラ130が設計のための第2パスの最適化を実行することを可能にする。この場合、設計の初期構成は、設計定義で（例えば、1つ以上のビットファイルとして）示され、FPGAデバイス121～123内の割り当てられたマクロコンポーネントを構成するために使用される。初期構成内の指定されたタスクの実行中、合成ロジック416は、設計に対して合成、配置及びルーティングの計算を実行することによって、設計に対して最適化された構成を自動的に生成する。初期構成においてタスクの実行が一時停止され、アーキテクチャ状態（例えば、レジスタ、メモリ等）が保存され、構成ロジック414は、FPGAデバイスを再プログラムして、初期構成を最適化された構成に置き換える。その後、状態が復元され、最適化された構成でタスクの実行が再開される。

【0049】

第2パスの最適化によって設計における入力及び／又は出力ポートの位置（例えば、制御、データ及び結果ポート等）に変化が生じた場合、合成ロジック416は、初期構成における初期ポートの位置を、最適化された構成における最適化されたポートの位置と相関させる再マッピング通知も生成するので、これらのポートに依存する任意のプロセスが、これらのポートを依然として配置することができる。代替の実施形態では、ポートのハードウェア再マッピングは、FPGAによって実行される。

【0050】

スタンドアロンマクロスケジューラ130及びローカルマクロスケジューラ131～133の両方を含む一実施形態では、計算集約型作業（例えば、合成、配置及びルーティン

10

20

30

40

50

グ) がスタンドアロンマクロスケジューラ 130 上で実行され、計算集約型でない作業がローカルマクロスケジューラ 131～133 上で実行される。特に、ローカルマクロスケジューラ 131～133 は、マクロコンポーネントの割り当てや、個々の FPG A デバイス 121～123 内のアクセラレータの配置等の機能を実行することができる。

【0051】

マクロスケジューラ 130 の機能は、異なる回路モジュールによって実行されるものとして上述した実施形態で説明されているが、マクロスケジューラ 130 内のモジュールは、代替の実施形態において、マイクロコード又はソフトウェアモジュールを用いて実装することができる。いくつかの実施形態では、マクロスケジューラ 130 の機能は、部分的にソフトウェアで、部分的にハードウェアで実装することができる。いくつかの実施形態では、マクロスケジューラ 130 の機能は、1 つ以上のスタンドアロンマクロスケジューラ（例えば、130）、1 つ以上のローカルマクロスケジューラ（例えば、131～133）、又は、クライアントデバイス（例えば、110～111）の間で、多くの異なる方法で分割することができる。例えば、タイルレベルの合成並びに／又は配置及びルーティングのプロセスは、要求するクライアント、ローカルマクロスケジューラ又はスタンドアロンマクロスケジューラデバイスによって実行され得る。

10

【0052】

一実施形態では、マクロスケジューラ 130 は、クライアントデバイスの代わりに、初期合成、配置及びルーティングプロセス等の計算量を必要とするタスクを実行するので、クライアントデバイスは、よりパワーの小さいコンピュータ（例えば、モバイルデバイス）とすることができる。代替として、クライアントデバイスが十分にパワフルであり、計算サイクルに余裕がある場合、クライアントデバイスは、マクロスケジューリング機能をサポートすることができる。一実施形態では、クライアントデバイスは、これらの機能を実行するためのハードウェアアクセラレータも含むことができる。

20

【0053】

図 5 は、クライアントからの設計定義に基づいて 1 つ以上の FPG A デバイスを構成し、第 2 パスの最適化及びコンテキストスイッチ等のマクロスケジューラの他の機能を実行するプロセス 500 を示すフロー図である。プロセス 500 は、FPG A デバイス 121～123 で実行される複数のタスクの各々に対して実行され、これらのプロセス 500 は、時間的に重なってもよい。一実施形態では、プロセス 500 における動作は、マクロスケジューラ 130 によって実行される。或いは、プロセス 500 における動作は、ローカルマクロスケジューラ 131～133 及び／若しくはクライアントデバイス 110～111 において、又は、スタンドアロンマクロスケジューラ 130 と、ローカルマクロスケジューラ 131～133 と、クライアントデバイス 110～111 との組み合わせにおいて実行されてもよい。

30

【0054】

ブロック 501 において、データベース 420 は、FPG A デバイス 121～123 に関するマクロコンポーネントの位置及び利用可能性を示す情報を記憶する。データベース 420 は、各マクロコンポーネントの一意の識別子を記憶する「マクロ」フィールド 421 を含むことによって、FPG A デバイス 121～123 内のマクロコンポーネントを列挙する。マクロコンポーネントの位置は、データベース 420 の「位置」フィールド 422 に記憶される。一実施形態では、「クライアント IP」フィールド 423、「PID」フィールド 424、「タスク」フィールド 425、及び／又は、「スケジュール」フィールド 426 は、マクロコンポーネントの利用可能性（すなわち、マクロコンポーネントが割り当てられていないかどうか）を示す。プロセス 500 は、ブロック 501 からブロック 503 に進む。

40

【0055】

ブロック 503 において、マクロスケジューラ 130 の API モジュール 411 は、第 1 クライアントデバイス 110 から設計定義を受信する。受信した設計定義は、設計で用いられる 1 つ以上のマクロコンポーネントを示す。設計定義は、マクロコンポーネントが

50

マクログラフ（例えば、先の例における「myAccelerator」）においてどのように接続されるかも示す。プロセス500は、ブロック503からブロック505に進む。

【0056】

ブロック505において、APIモジュール411は、クライアント110からタスク定義を受信する。タスク定義は、先に受信した設計定義におけるマクログラフを用いて実行されるアクションに関してタスクを定義する。例えば、タスク定義は、マクログラフによって定義されたアクセラレータに対する入力値を指定し、実行を要求し、結果として得られる出力値を読み出すことができる。プロセス500は、ブロック505からブロック507に進む。

10

【0057】

ブロック507において、リソース割り当てロジック413は、データベース420に列挙されているマクロコンポーネントのうち、設計定義で指定されたマクロコンポーネント毎に利用可能であると示された1つのマクロコンポーネントを割り当てることによって、設計に関するマクロコンポーネントを割り当てる。設計に割り当てられたマクロコンポーネントは、同じFPGAデバイスからのものであってもよいし、異なるFPGAデバイスからのものであってもよい。同じFPGAからのマクロコンポーネントを異なる設計に割り当てることができる。例えば、APIモジュール411は、第1設計定義によって定義された第1設計及び第2設計定義によって定義された第2設計を、同じ又は異なるクライアントデバイスから受信することができる。リソース割り当てロジック413は、第1設計に関するマクロコンポーネントの第1セットと、第2設計に関するマクロコンポーネントの第2セットとを割り当て、マクロコンポーネントの第1セットは、マクロコンポーネントの第2セットと同じFPGAデバイスに配置される。

20

【0058】

一実施形態では、マクロコンポーネントは、特定の期間に割り当てられる（すなわち、スケジューリングされる）。例えば、マクロコンポーネントは、或る期間中には第1設計に割り当てられ、異なる期間中には第2設計に割り当てられてもよい。複数のマクロコンポーネントを有するマクログラフによって定義されるアクセラレータ設計の場合、リソース割り当てロジック413は、設計で指定された全てのマクロコンポーネントを利用可能であり、スケジューリングすることができる期間を識別することができる。一実施形態では、リソース割り当てロジック413は、同じクライアント又は複数の異なるクライアントからの複数の設計のスケジューリングを最適化して、マクロコンポーネントの使用を経時的に最大化する。

30

【0059】

また、リソース割り当てロジック413は、アクセラレータ設計を実行するように構成された割り当てられたマクロコンポーネントにおいてタスクを実行することができる1つ以上の期間を識別することによって、タスク定義によって定義されたタスクの実行をスケジューリングする。一実施形態では、同じアクセラレータ設計を用いる複数のタスクを定義することができ、すなわち、リソース割り当てロジック413は、これらのタスクを、異なる期間中にアクセラレータを用いるようにスケジューリングする。リソース割り当てロジック413によるスケジューリングプロセス507は、さらなる最適化ステップ509A、509Bを含む。

40

【0060】

ブロック509Aにおいて、リソース割り当てロジック413は、他の設計によるアクセラレータと時分割多重化することができる、設計によるアクセラレータを識別する。例えば、2つの異なる（同じ又は異なるクライアントからの）設計の両方が同じアクセラレータ（例えば、同じタイプのマクロコンポーネントが同じように接続されている）を用いる場合、アクセラレータは、一度構成されると、両方の設計で 사용할ことができる。リソース割り当てロジック413は、第1期間中にアクセラレータ内の第1タスクの実行をスケジューリングし、第1期間と重複しない第2期間中に同じアクセラレータ内の第2タ

50

スクの実行をスケジューリングする。

【0061】

ブロック509Bにおいて、リソース割り当てロジック413は、複数の設計間で共有可能なマクロコンポーネントを識別する。例えば、第1設計及び第2設計の各々は、8ビットのメモリを要求する場合がある。したがって、リソース割り当てロジック413は、両方の設計間で共有される単一の16ビットのメモリマクロコンポーネントを割り当てることができる。これにより、マクロコンポーネントの未使用のビットを別の設計に割り当てることができる。入力ポート及び出力ポートのマクロコンポーネントを、設計間で同様に共有することができる。

【0062】

ブロック511において、リソース追跡モジュール412は、データベース420を更新して、クライアント110のIPアドレスをデータベース420のマクロコンポーネントに関連する行に記憶することによって、割り当てられたマクロコンポーネントの各々を、設計が発生したクライアント110のネットワークアドレス（例えば、IPアドレス）に関連付ける。また、リソース追跡モジュール412は、割り当てられたマクロコンポーネントの各々に関連する行内のタスクの識別子をデータベース420に記憶することによって、割り当てられたマクロコンポーネントの各々をタスクに関連付ける。プロセス500は、ブロック511から、プロセス500はブロック513に進む。

【0063】

マクロコンポーネントが、受信した設計に割り当てられると、プロセス500は、ブロック513において、（タスク定義で指定されているように）設計において実行されるタスクの実行時期であるかどうかを、ブロック507において予め判別されたタスクのスケジュールに従って判別する。タスクが実行時期でない場合、プロセス500は、ブロック515に進む。プロセス500は、タスクが時期になるまで、ブロック515、513の間を循環することによって、タスクが時期に達するのを待機する。この待機期間中、合成ロジック416は、設計に関する合成、配置及びルーティングの手順を実行することによって、第2パスの最適化を実行することができる。一実施形態では、第2パスの最適化は、クライアントから受信したRTLレベルの設計を用いて実行され、設計を実行するために実際に割り当てられたマクロコンポーネント、及び／又は、割り当てられた同じマクロコンポーネントの一部若しくは全てを共有する、若しくは、同時にスケジューリングされる他の設計に基づいている。

【0064】

ブロック513において、タスクが実行時期にある場合、プロセス500は、ブロック517に進む。ブロック517において、リソース追跡モジュール412は、データベース420を更新して、割り当てられたマクロコンポーネントをタスクに関連付ける。リソース追跡ロジック412は、タスクが実行されるアクセラレータの割り当てられたマクロコンポーネントの各々に対応するタスクの識別子を、「タスク」フィールド425に書き込む。プロセス500は、ブロック517からブロック519に進む。

【0065】

ブロック519において、構成ロジック414は、設計定義において指定された初期構成又は第2パスの最適化プロセス515によって生成された最適化された構成に従って、割り当てられたマクロコンポーネントのセットを構成することによって、FPGAデバイスのセットにおける設計を実行する。一実施形態では、設計定義は、1つ以上のタイルをマクログラフの一部として指定し、さらに、指定されたタイルの各々を構成するビットファイルを示す。したがって、構成ロジック414は、指定されたタイル毎に、指定されたタイルについて示されたビットファイルを用いて、対応する割り当てられたタイルを、割り当てられたマクロコンポーネントのセットから構成する。一実施形態では、割り当てられたタイルは、ビットファイルの内容をタイルの構成レジスタに書き込むことによって構成される。設計に関して第2パスの最適化が既に実行された場合（例えば、ブロック515において実行を待機している間に）、設計定義で最初に指定された初期構成の代わりに

10

20

30

40

50

、第2パスの最適化から生成された最適化された構成が使用される。プロセス500は、ブロック519からブロック521に進む。

【0066】

ブロック521では、構成ロジック414は、構成が完了し、構成されたマクロコンポーネントにおけるタスクの実行が開始することを示す。ブロック523において、第2パスの最適化がブロック515において未だ実行されていない場合、タスクが初期構成において実行されている間に第2パスの最適化が実行される。したがって、合成ロジック416は、タスクが実行を開始するのを待機する間（すなわち、ブロック515において）又はタスクが実行されている間（すなわち、ブロック523において）、設計に関する最適化された構成を生成する。一実施形態では、第2パスの最適化は、ブロック515で部分的に実行され、ブロック523で完了してもよい。第2パスの最適化は、構成データ（例えば、ビットファイル、ビットストリーム）と、最適化された構成において初期構成に対して移動する入力又は出力ポートの再マッピング情報と、を生成する。プロセス500は、ブロック523からブロック525に進む。

10

【0067】

ブロック525において、タスクの実行が一時停止される。一実施形態では、タスクは、タスクが実行されているマクロコンポーネントに供給されるクロック信号を停止することによって、一時停止される。タスクが一時停止された状態では、タスクのメモリ及びレジスタ状態が保存される。プロセス500は、ブロック525からブロック527に進む。

20

【0068】

ブロック527において、構成ロジック414は、FPGAデバイス内のマクロコンポーネントを再プログラムして、初期構成を、ブロック523で生成された最適化された構成と置き換える。また、再マッピングの通知は、初期構成内の初期ポートの位置を最適化された構成内の最適化されたポートの位置と相関させる再マッピング情報から生成される。再マッピングの通知によって、他のプロセスは、新たな入出力ポートの位置を発見することが可能になる。或いは、ポートの位置を、FPGAハードウェア内で再マッピングしてもよい。タスク用に先に保存されたレジスタ及びメモリ状態は、再構成されたマクロコンポーネントに復元される。このコンポーネントは、最適化された構成でレジスタ及びメモリの位置が変更された場合に、再マッピングのレジスタ及びメモリの位置を含むことができる。最適化された設計に切り替える場合、先に保存されたレジスタ及びメモリ状態の一部が冗長になる可能性もある。一実施形態では、最適化プロセスは、最適化された構成でもはや使われていない状態ビットを決定し、識別し、次に、最適化された構成を用いてタスクの実行を再開する前に、タスクの保存されたコンテキストを、新たな最適化された構成に適合するように変更する（例えば、もはや使われていないと識別された信号の全ての状態ビットを削除することによって）。プロセス500は、ブロック527からブロック529に進む。

30

【0069】

ブロック529において、タスクの実行は、FPGAデバイス内のマクロコンポーネントの最適化された構成において再開される。一実施形態では、マクロコンポーネントに供給されるクロック信号を再開することによって、実行が再開される。

40

【0070】

ブロック531～539は、既に使用されているマクロコンポーネントにおいてより高い優先順位のタスクを実行することができるように、コンテキストスイッチを実行する動作を含む。ブロック531において、元のタスクがこのようより高い優先順位のタスクに置き換えられない場合、プロセスはブロック541に進む。ブロック541において、元のタスクが完了していない場合、プロセス500はブロック529に戻り、元のタスクの実行が継続する。したがって、ブロック529、531、541は、元のタスクが置き換えられるか完了するまで繰り返される。ブロック531において、元のタスクが、元のタスクが実行されているマクロコンポーネントの使用を要求するより高い優先順位のタス

50

クに置き換えられる場合、プロセス500はブロック533に進む。一実施形態では、ブロック531において元のタスクが置き換えられることは、ブロック513において、より高い優先順位のタスクに対して実行されるプロセス500の別のインスタンスにおいて、より高い優先順位のタスクが到来することに対応する。

【0071】

ブロック533において、リソース割り当てロジック413は、より高い優先順位のタスクが実行されるのを可能にするために、コンテキストスイッチを実行する。元のタスクのレジスタ及びメモリ状態が保存され、マクロコンポーネントは、より高い優先順位のタスクを実行するために別の設計に再構成される。より高い優先順位のタスクの実行は、ブロック535において開始され、ブロック537でタスクが完了するまで継続する。元のタスクに対して実行されるブロック535、537は、より高い優先順位のタスクに対して実行されるプロセス500の他のインスタンスにおけるブロック517～543に対応する。より高い優先順位のタスクが完了すると、プロセス500は、ブロック537からブロック539に進む。ブロック539において、リソース割り当てロジック413は、元のタスクの実行を再開するためにコンテキストスイッチを実行する。保存されたレジスタ及びメモリ状態が復元され、マクロコンポーネントは、元のタスクを実行するための設計を実装するように再構成される。

【0072】

元のタスクの実行は、タスクが完了するまで継続する。ブロック541において、タスクが完了した場合、プロセス500はブロック543に進む。ブロック543において、タスクが終了し、タスクによって用いられたマクロコンポーネントは、他の設計で用いられていない場合には、割り当てが解除され、新たな設計に利用可能になる。割り当て解除されたマクロコンポーネントのデータベースの行は、リソース追跡ロジック412によってデータベース420内で更新され、これにより、「タスク」425及び「クライアントIP」423フィールドがクリアされる。次にスケジューリングされているタスクが開始され（すなわち、それ自体のブロック513から）、新たに割り当て解除されたマクロコンポーネントが使用され得る。

【0073】

プロセス500の動作によれば、ローカルマクロスケジューラ131～133及び／又はスタンドアロンマクロスケジューラ130は、設計定義のマクログラフに従ってマクロコンポーネントを割り当て、スケジューリングするメカニズムを提供する。マクログラフは、FPGAデバイス121～123に実装されるアクセラレータを定義する。API411を使用することにより、ユーザは、FPGAデバイス121～123においてハードウェアアクセラレータとして実装される関数を生成することができ、また、当該関数の入力データを割り当て、結果として得られる出力を仮想的に読み出すことができる。リソース共有、時分割多重化及び他の最適化アプローチにより、FPGAリソースの仮想化は、FPGAのハードウェアリソースを複数のクライアントに提供する際のより効率的な割り当て及びスケジューリングを可能にする。

【0074】

マクロスケジューラは、フィールドプログラマブルゲートアレイ（FPGA）デバイスのセットの複数のマクロコンポーネントを列挙するデータベースを更新するように構成されたリソース追跡モジュールと、設計のための1つ以上の指定されたマクロコンポーネントを示す第1設計定義を第1クライアントデバイスから受信するように構成された通信インタフェースと、第1設計定義に示された1つ以上の指定されたマクロコンポーネント毎に複数のマクロコンポーネントのうち1つを割り当てることによって、設計のためのマクロコンポーネントの第1セットを割り当てるように構成されたリソース割り当てロジックと、割り当てられたマクロコンポーネントの第1セットを第1設計定義に従って構成することによって、FPGAデバイスのセットに設計を実装するように構成された構成ロジックと、を含む。

【0075】

マクロスケジューラにおいて、第1設計定義は、1つ以上の指定されたマクロコンポーネント間の接続を定義し、1つ以上の指定されたマクロコンポーネントの各々のタイプを示すマクログラフを含む。

【0076】

マクロスケジューラにおいて、1つ以上の指定されたマクロコンポーネントは、1つ以上の指定されたタイルと、1つ以上の指定された固定関数ユニットと、1つ以上の指定されたレジスタと、を含む。また、第1設計定義は、1つ以上の指定されたタイルの各々のビットファイルを示す。構成ロジックは、1つ以上の指定されたタイル毎に、マクロコンポーネントの第1セット内の割り当てられたタイルを、指定されたタイルのビットファイルに基づいてプログラムするように構成されている。

10

【0077】

マクロスケジューラにおいて、リソース追跡モジュールは、マクロコンポーネントの第1セットの各々を、第1クライアントデバイスのネットワークアドレスと、第1クライアントデバイスによって要求された第1タスクと関連付けるようにさらに構成されている。

【0078】

マクロスケジューラにおいて、リソース追跡モジュールは、複数のマクロコンポーネント毎に、FPGAデバイスのセットにおけるマクロコンポーネントの位置及びマクロコンポーネントの利用可能性をデータベースに記録するようにさらに構成されている。

【0079】

また、マクロスケジューラは、FPGAデバイスのセットの初期構成におけるタスクの実行中に、設計のための最適化された構成を生成し、初期構成における初期ポートの位置を、最適化された構成における最適化されたポートの位置と関連させる再マッピング通知を生成するように構成された合成ロジックを含む。初期構成は、第1設計定義によって示され、構成ロジックは、FPGAデバイスのセットの最適化された構成でタスクの実行を再開する前に初期構成を最適化された構成に置き換えるように、FPGAデバイスのセットを再プログラムするように構成されている。

20

【0080】

マクロスケジューラにおいて、第1設計定義は、アクセラレータの定義を含む。リソース割り当てロジックは、第1期間中に第1クライアントデバイスからの第1タスクのアクセラレータにおける実行をスケジューリングし、第2期間中に第2クライアントデバイスからの第2タスクのアクセラレータにおける実行をスケジューリングするようにさらに構成されている。

30

【0081】

マクロスケジューラにおいて、リソース割り当てロジックは、複数のマクロコンポーネントの第2セットを、第2クライアントデバイスから受信した第2設計定義によって定義された第2設計に対して割り当てるようにさらに構成されている。マクロコンポーネントの第1セット内のマクロコンポーネントの第1サブセットは、マクロコンポーネントの第2セット内のマクロコンポーネントの第2サブセットとして、FPGAデバイスのセットのうち同じFPGAデバイスに配置される。

【0082】

マクロスケジューラにおいて、マクロコンポーネントの第1セットは、複数のFPGAデバイスのうち少なくとも2つの異なるFPGAデバイスのマクロコンポーネントを含む。

40

【0083】

マクロスケジューラの動作方法は、フィールドプログラマブルゲートアレイ（FPGA）デバイスのセットの複数のマクロコンポーネントを列挙するデータベースを更新することと、設計のための1つ以上の指定されたマクロコンポーネントを示す第1設計定義を第1クライアントデバイスから受信することと、第1設計定義に示された1つ以上の指定されたマクロコンポーネント毎に複数のマクロコンポーネントのうち1つを割り当てることとによって、設計のためのマクロコンポーネントの第1セットを割り当てることと、割り当

50

てられたマクロコンポーネントの第1セットを第1設計定義に従って構成することによって、FPGAデバイスのセットに設計を実装することと、を含む。

【0084】

方法において、1つ以上の指定されたマクロコンポーネントは、1つ以上の指定されたタイルを含み、第1設計定義は、1つ以上の指定されたタイルの各々のビットファイルを示す。また、方法は、1つ以上の指定されたタイル毎に、マクロコンポーネントの第1セット内の割り当てられたタイルを、指定されたタイルのビットファイルに基づいてプログラムすることを含む。

【0085】

また、方法は、マクロコンポーネントの第1セットの各々を、第1クライアントデバイスのネットワークアドレスと、第1クライアントデバイスによって要求された第1タスクと関連付けることを含む。

【0086】

また、方法は、複数のマクロコンポーネント毎に、FPGAデバイスのセットにおけるマクロコンポーネントの位置及びマクロコンポーネントの利用可能性をデータベースに記録することを含む。

【0087】

また、方法は、FPGAデバイスのセットの初期構成におけるタスクの実行中に、設計のための最適化された構成を生成することと、FPGAデバイスのセットの最適化された構成でタスクの実行を再開する前に初期構成を最適化された構成に置き換えるように、FPGAデバイスのセットを再プログラムすることと、初期構成における初期ポートの位置を、最適化された構成における最適化されたポートの位置と相関させる再マッピング通知を生成することと、を含む。初期構成は、第1設計定義によって示される。

【0088】

方法において、第1設計定義は、アクセラレータの定義を含む。また、方法は、第1期間中に第1クライアントデバイスからの第1タスクのアクセラレータにおける実行をスケジューリングすることと、第2期間中に第2クライアントデバイスからの第2タスクのアクセラレータにおける実行をスケジューリングすることと、を含む。

【0089】

また、方法は、複数のマクロコンポーネントの第2セットを、第2クライアントデバイスから受信した第2設計定義によって定義された第2設計に対して割り当てることを含む。マクロコンポーネントの第1セット内のマクロコンポーネントの第1サブセットは、マクロコンポーネントの第2セット内のマクロコンポーネントの第2サブセットとして、FPGAデバイスのセットのうち同じFPGAデバイスに配置される。

【0090】

コンピューティングシステムは、複数のマクロコンポーネントを含むフィールドプログラマブルゲートアレイ（FPGA）デバイスのセットと、複数のマクロコンポーネントを列挙するように構成されたデータベースと、スタンドアロンマクロスケジューラと、を含む。スタンドアロンマクロスケジューラは、データベースを更新するように構成されたリソース追跡モジュールと、設計のための1つ以上の指定されたマクロコンポーネントを示す第1設計定義を第1クライアントデバイスから受信するように構成された通信インタフェースと、第1設計定義に示された1つ以上の指定されたマクロコンポーネント毎に複数のマクロコンポーネントのうち1つを割り当てることによって、設計のためのマクロコンポーネントの第1セットを割り当てるように構成されたリソース割り当てロジックと、割り当てられたマクロコンポーネントの第1セットを第1設計定義に従って構成することによって、FPGAデバイスのセットに設計を実装するように構成された構成ロジックと、を含む。

【0091】

また、システムは、第1クライアントデバイスを含む。また、スタンドアロンマクロスケジューラは、第1クライアントデバイスからのコマンドに応じて、リソース割り当てロ

10

20

30

40

50

ジックによる割り当て、及び、構成ロジックによる構成を開始するように構成されたアプリケーションプログラミングインタフェース（API）を含む。

【0092】

システムにおいて、FPGAデバイスのセットのうち1つ以上のFPGAデバイスの各々は、1つ以上のFPGAデバイス内の割り当てられたマクロコンポーネントのうち少なくとも1つの配置を決定するように構成されたローカルマクロスケジューラを含む。

【0093】

また、システムは、FPGAデバイスのセットをスタンドアロンマクロスケジューラに接続する複数のネットワークチャネルを含む。

【0094】

本明細書で使用される場合、「に接続された」という用語は、1つ以上の介在するコンポーネントを介して直接的又は間接的に接続されることを意味し得る。本明細書に記載された様々なバスを介して提供される何れの信号も、他の信号と時分割多重化されてもよく、1つ以上の共通バスを介して提供されてもよい。さらに、回路コンポーネント又はブロック間の相互接続は、バス又は単一の信号線として示されてもよい。各バスは、代替的に、1つ以上の単一の信号線であってもよく、単一の信号線の各々は、代替的に、バスであってもよい。

【0095】

特定の実施形態は、非一時的なコンピュータ可読媒体に記憶された命令を含むことができるコンピュータプログラム製品として実施することができる。これらの命令を用いて、汎用又は特殊目的のプロセッサをプログラムして、説明した動作を実行することができる。コンピュータ可読媒体は、機械（例えば、コンピュータ）によって読み出し可能な形式（例えば、ソフトウェア、処理アプリケーション）で情報を記憶又は送信する任意のメカニズムを含む。非一時的なコンピュータ可読記憶媒体は、磁気記憶媒体（例えば、フロッピー（登録商標）ディスク）、光記憶媒体（例えば、CD-ROM）、光磁気記憶媒体、読み出し専用メモリ（ROM）、ランダムアクセスメモリ（RAM）、消去可能なプログラマブルメモリ（例えば、EPROM、EEPROM）、フラッシュメモリ、又は、電子命令を記憶するのに適した別のタイプの媒体を含むことができるが、これらに限定されない。

【0096】

さらに、いくつかの実施形態は、コンピュータ可読媒体が複数のコンピュータシステムに記憶され及び／又は複数のコンピュータシステムによって実行される分散コンピューティング環境において実施されてもよい。さらに、コンピュータシステム間で転送される情報は、コンピュータシステムを接続する伝送媒体を介してプル（pulled）されてもよいし、プッシュ（pushed）されてもよい。

【0097】

概して、コンピューティングシステム100、及び／又は、コンピュータ可読記憶媒体上に担持されたコンピューティングシステム100の一部を表すデータ構造は、プログラムによって読み出され、コンピューティングシステム100を含むハードウェアを製造するために直接的又は間接的に使用され得る、データベース又は他のデータ構造であってもよい。例えば、データ構造は、Verilog又はVHDL等の高レベル設計言語（HDL）におけるハードウェア機能の動作レベル記述であってもよいし、レジスタ転送レベル（RTL）記述であってもよい。記述は、ゲートのリストを含むネットリストを合成ライブラリから生成するために記述を合成することができる合成ツールによって読み出されてもよい。ネットリストは、コンピューティングシステム100を含むハードウェアの機能を表すゲートのセットを含む。次に、ネットリストを配置及びルーティングして、マスクに適用される幾何学的形状を記述するデータセットを生成することができる。次に、マスクを様々な半導体製造ステップで使用して、コンピューティングシステム100に対応する半導体回路を生成することができる。或いは、コンピュータ可読記憶媒体上のデータベースは、ネットリスト（合成ライブラリを含む若しくは含まない）、又は、必要に応じて

10

20

30

40

50

データセット若しくはグラフィックデータシステム（GDS）IIデータであってもよい。

【0098】

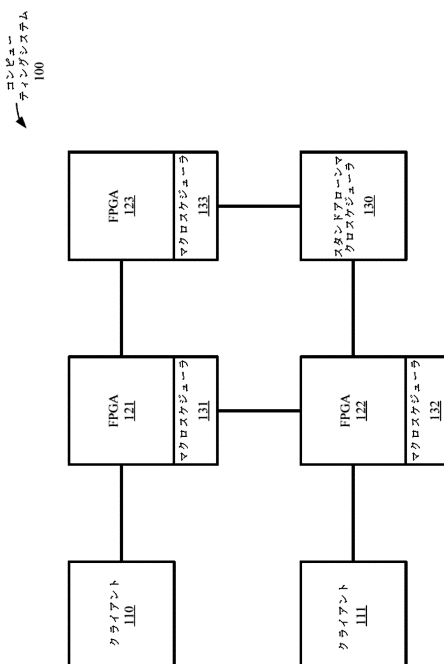
本明細書における方法の動作は、特定の順序で示され、説明されているが、各方法の動作の順序は、特定の動作が逆の順序で実行され得るように、又は、特定の動作が少なくとも部分的に他の動作と同時に実行され得るように、変更されてもよい。別の実施形態では、個別の動作の命令又はサブ動作は、間欠的及び／又は交互であってもよい。

【0099】

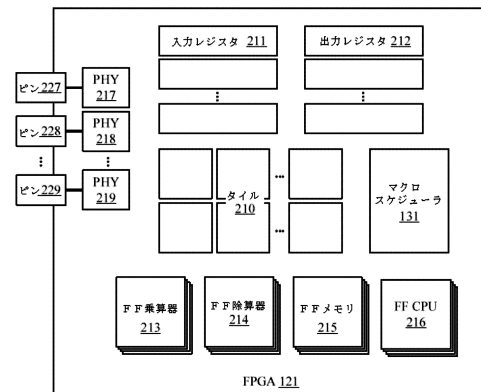
上述した明細書では、実施形態を、その特定の例示的な実施形態を参照して説明した。しかしながら、添付の特許請求の範囲に記載されている実施形態のより広い範囲から逸脱することなく、様々な変更及び修正を行うことができることは明らかである。したがって、明細書及び図面は、限定的な意味ではなく、例示的な意味でみなされる。

10

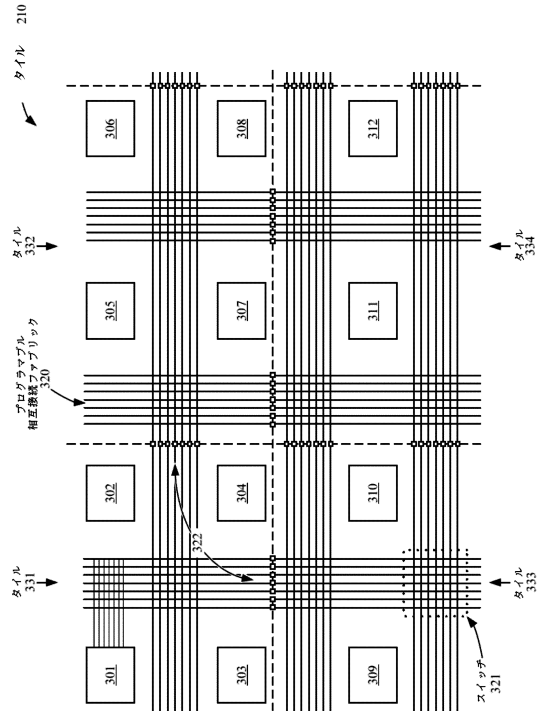
【図1】



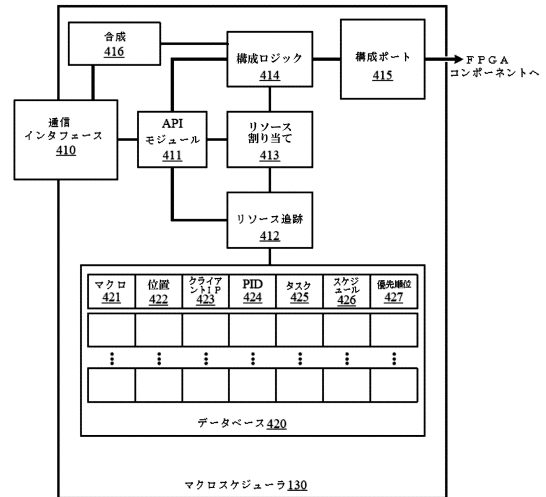
【図2】



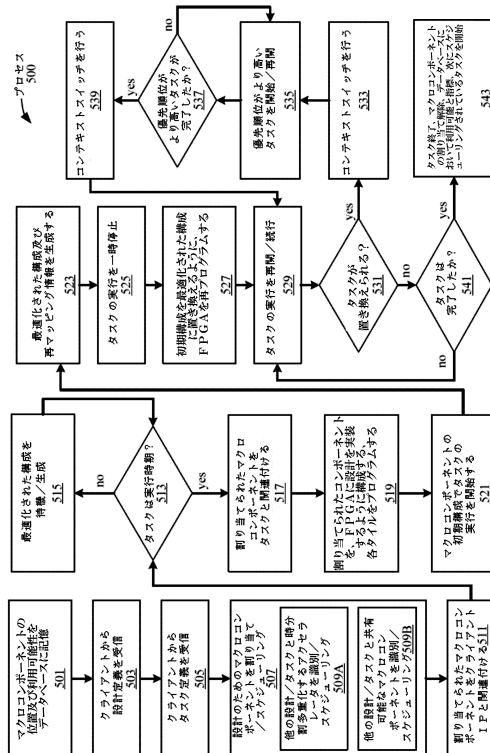
【図 3】



【図 4】



【図 5】



フロントページの続き

- (72)発明者 デイビッド エイ. ロバーツ
アメリカ合衆国 01719 マサチューセッツ州、ボックスボロー、セントラル ストリート
90、1、2 & 3 フロア
- (72)発明者 アンドリュー ジー. ケーゲル
アメリカ合衆国 98007 ワシントン州、ベルビュー、2002 156番 アベニュー ノ
ースイースト、スイート 300
- (72)発明者 エリオット エイチ. メドニック
アメリカ合衆国 01719 マサチューセッツ州、ボックスボロー、セントラル ストリート
90、1、2 & 3 フロア

審査官 漆原 孝治

- (56)参考文献 特表2003-518666 (JP, A)
特開2017-151794 (JP, A)

- (58)調査した分野(Int.Cl., DB名)
G06F 9/48
H03K 19/177