

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2010-26607

(P2010-26607A)

(43) 公開日 平成22年2月4日(2010.2.4)

(51) Int.Cl.	F I	テーマコード (参考)
G06F 15/173 (2006.01)	G06F 15/173 640A	5B045
G06F 1/18 (2006.01)	G06F 1/00 320F	

審査請求 未請求 請求項の数 10 O L (全 27 頁)

(21) 出願番号 特願2008-184301 (P2008-184301) (22) 出願日 平成20年7月15日 (2008.7.15)	(71) 出願人 504132881 国立大学法人東京農工大学 東京都府中市晴見町3-8-1 (74) 代理人 100122884 弁理士 角田 芳末 (74) 代理人 100133824 弁理士 伊藤 仁恭 (72) 発明者 関根 優年 東京都府中市晴見町3-8-1 国立大学 法人東京農工大学内 (72) 発明者 飯島 浩晃 東京都府中市晴見町3-8-1 国立大学 法人東京農工大学内
特許法第30条第1項適用申請有り 平成20年1月9日 社団法人電子情報通信学会発行の「電子情報通信学会技術研究報告 信学技報 Vol. 107 No. 4 14」に発表	

最終頁に続く

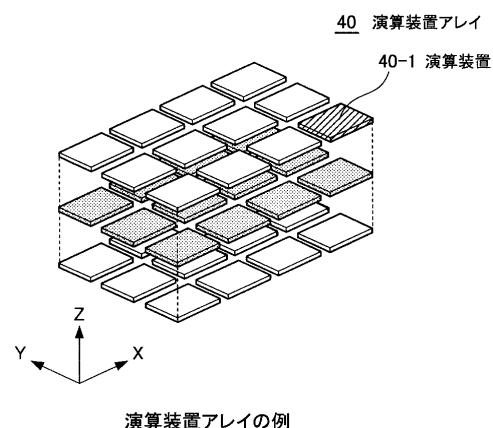
(54) 【発明の名称】 情報処理装置および仮想回路書き込み方法

(57) 【要約】

【課題】個人用に使える安価で高性能な大規模演算用のコンピュータを提供する。

【解決手段】演算対象について所定の演算を行うアプリケーション・プログラムを記憶するメモリと、演算対象の各々の問題領域に対応して隣接する演算装置間で直接データ通信可能に接続されて配置され、またアプリケーション・プログラムの実行に用いられ各々の問題領域に対応した演算を行う演算回路が再構成され、隣接する演算装置間で問題領域についての演算結果データを送受信する複数の演算装置からなる演算装置アレイ40と、アプリケーション・プログラムを実行し、演算装置アレイ40を構成する各演算装置から各問題領域についての演算結果データを取得し、演算対象について演算結果を算出するホストプロセッサと、メモリ、ホストプロセッサおよび演算装置アレイ40との間で、データを通信するバスを備える。

【選択図】図4



【特許請求の範囲】**【請求項 1】**

演算対象について所定の演算を行うアプリケーション・プログラムを記憶するメモリと

、
前記演算対象の各々の問題領域に対応して隣接する演算装置間で直接データ通信可能に接続されて配置され、また前記アプリケーション・プログラムの実行に用いられ前記各々の問題領域に対応した演算を行う演算回路が再構成され、隣接する演算装置間で前記問題領域についての演算結果データを送受信する複数の演算装置からなる演算装置アレイと、

前記アプリケーション・プログラムを実行し、前記演算装置アレイを構成する各演算装置から各問題領域についての演算結果データを取得し、前記演算対象について演算結果を算出するホストプロセッサと、

前記メモリ、前記ホストプロセッサおよび前記演算装置アレイとの間で、データを通信するバスと、を含む

情報処理装置。

【請求項 2】

前記複数の演算装置は、ホストプロセッサによって予め決められた演算結果の転送先に従い、当該演算装置の演算結果データを隣接する所定の演算装置に転送する

請求項 1 に記載の情報処理装置。

【請求項 3】

前記演算装置アレイを構成する前記複数の演算装置に対して、前記ホストプロセッサの指示に基づいて、前記演算回路を書き込むための仮想回路データ書き込み回路を所定の順序で書き込む

請求項 2 に記載の情報処理装置。

【請求項 4】

前記演算装置アレイを構成する前記複数の演算装置において、全ての演算装置について前記仮想回路データ書き込み回路の書き込みが終了後、前記仮想回路データ書き込み回路が最後に書き込まれた演算装置から最初に書き込まれた演算装置まで前記全ての演算装置について前記演算回路を書き込む

請求項 3 に記載の情報処理装置。

【請求項 5】

前記演算装置は、前記演算回路が書き込まれる演算ボードと、前記仮想回路データ書き込み回路が書き込まれる書き込みボードから一対の演算装置が構成される

請求項 4 に記載の情報処理装置。

【請求項 6】

前記演算装置アレイを構成する前記複数の演算装置において、前記演算回路の書き込みとともに、前記ホストプロセッサと送受信するデータを転送するためのデータ入出力回路を書き込む

請求項 5 に記載の情報処理装置。

【請求項 7】

前記演算装置の前記演算ボードは、上下左右前後の隣接する演算ボードと接続するための第 1 のコネクタおよび対をなす前記書き込みボードと接続するための第 2 のコネクタを備え、また前記書き込みボードは、対をなす前記演算ボードと接続するための第 3 のコネクタを備え、

前記演算ボードの演算回路は、第 1 のコネクタを介して隣接する演算ボードの演算回路と前記演算結果データの送受信を行うとともに、第 2 および第 3 のコネクタを介して前記書き込みボードの前記書き込み回路および前記データ入出力回路とデータの送受信を行う

請求項 6 に記載の情報処理装置。

【請求項 8】

前記ホストプロセッサは、前記演算装置アレイを構成する複数の演算ボードのうち所定の演算ボード同士を第 1 のコネクタを介して電氣的に接続するように制御する

10

20

30

40

50

請求項 7 に記載の情報処理装置。

【請求項 9】

前記演算対象が周期境界条件を満たす場合、前記ホストプロセッサは、前記演算装置アレイを構成する複数の演算装置のうち周期境界に該当する演算装置同士を第 1 のコネクタを介して電氣的に接続するように制御する

請求項 7 に記載の情報処理装置。

【請求項 10】

演算対象の各々の問題領域に対応して隣接する演算装置間で直接データ通信可能に接続されて配置された複数の演算装置の各々に対して、ホストプロセッサの指示に基づき演算回路を書き込むための仮想回路データ書き込み回路を所定の順序で書き込むステップと、

10

全ての演算装置について前記仮想回路データ書き込み回路の書き込みが終了後、前記仮想回路データ書き込み回路が最後に書き込まれた演算装置から最初に書き込まれた演算装置まで前記全ての演算装置について前記演算回路を書き込むステップと、

を含む仮想回路書き込み方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、情報処理装置および情報処理方法に関し、特に、書き換え可能なプログラマブル・ロジック・デバイス (PLD: Programmable Logic Device) を用いて膨大な計算量の処理を行うコンピュータに適用して好適な情報処理装置および仮想回路書き込み方法に関する。

20

【背景技術】

【0002】

近年、大規模演算の環境が高度に整備されてきたのを受け、生物・化学・天文・工学などの広い分野において H P C (high-performance computing) による演算手法が利用されている。H P C は複数の計算機システムを結合し、一つのシステムとして演算環境を提供するものである。一つの計算機システムの性能が低くとも、システム全体としては高速に演算が可能となる。

【0003】

例えば、H P C の一例としてスーパーコンピュータがある。スーパーコンピュータは従来、専用のプロセッサを及び専用アーキテクチャで構成されてきたが、図 1 に示すように、バス 2 を介して汎用のマイクロ・プロセッサ 1 を多数接続して並列実行するアーキテクチャで構成されたものが現れている。

30

【0004】

また、H P C には、汎用 P C (Personal Computer) によるグリッド構成のものや、G P G P U (General Purpose Graphics Processing Unit) を利用したグリッド構成など、I C (Integrated Circuit) を並べた構成のものもある。

【0005】

さらに F P G A (Field Programmable Gate Array) 等のリコンフィギャラブル (再構成可能) な半導体集積回路 (LSI: Large Scale Integration) を用いて並列実行するアーキテクチャで構成されたものなど、高性能な演算を行う研究も見られるようになった。特に大規模演算向けに構成されているクラスタ型のは、R H P C (Reconfigurable High Performance Computing) や H P R C s (High-Performance Reconfigurable Computers) などと呼ばれている。このような回路をリコンフィギャラブルなハードウェアは、総称してプログラマブル・ロジック・デバイス (PLD: Programmable Logic Device) と呼ばれており、少数の製品から量産品まで幅広く流通している。

40

【0006】

例えば、本出願人は、ハードウェア・モジュール (hwModule) と呼ばれる P C I 型 F P G A ボードを利用した、ハードウェア / ソフトウェア (hw/sw) 複合体を提案している (例えば、特許文献 1 を参照。)。

50

【 0 0 0 7 】

hw/sw複合体とは、ハードウェア・モジュールのFPGAを仮想回路(hwNet)として利用し、ハードウェア・オブジェクト(hwObject)と呼ばれる仮想回路の詳細な制御を隠蔽するクラスを継承することにより、回路資源を容易に利用できるシステムである。このハードウェア/ソフトウェア複合体の特長として、汎用性、並列分散処理や外部機器との接続などがあげられる。

【 0 0 0 8 】

すなわち、ソフトウェアから直接、FPGA内の仮想回路を制御でき、対象となるアプリケーション対応して作成された適切な仮想回路をFPGAに書き込むことで、最速な仮想回路により、アプリ毎の問題を計算することが可能である。

10

【 0 0 0 9 】

一般に、同一の演算をソフトウェアとハードウェア回路で実行した場合、ハードウェア回路による実行はソフトウェアの場合と比較して30倍から数百倍、高速であることが経験的に知られている。例えば、現行のマイクロ・プロセッサは、3GHzの周波数で動作するが、安価なFPGAでは200MHz動作が普通に得られる最良の速度である。したがって、マイクロ・プロセッサによるソフトウェア実行に対して、FPGAによる仮想回路では、実効的に6GHz(200MHz×30)から20-40GHzで動作するマイクロ・プロセッサに対応する。さらに、FPGA内に複数個の演算回路を入れることで、10数倍の性能向上が可能となり、60GHz~400GHz相当の性能が見込まれる。

【特許文献1】特許第3845021号公報

20

【発明の開示】

【発明が解決しようとする課題】

【 0 0 1 0 】

上述したHPCに共通した特徴は、アプリケーション・プログラムを並列実行可能な形態に分割し、並列実行可能な演算装置に割り当てて実行するというものである。このような構成のHPCは高価であり、かつ、多人数が異なる問題領域に使用するため、HPCは汎用的な構成、すなわち、汎用のマイクロ・プロセッサ間で任意のデータ転送路が実現できるように設計される。概念的には、図1に示したバス2のような共通の通信路を介して、マイクロ・プロセッサ1が接続される形態である。しかし、いずれの演算結果もバス2を介して分散した各マイクロ・プロセッサ2に転送され、そのバスネックが性能のボトルネックとなっている。

30

【 0 0 1 1 】

上述のとおり、図1に示したような形態では、共通の通信路であるバス2のオーバーヘッドが大きくなる。そこで、バス2のオーバーヘッドをなくすために、図2に示すようなスイッチボックス5で高速化を図るものが普通である。スイッチボックス5において、任意のマイクロ・プロセッサからのデータを最適なマイクロ・プロセッサを選択して送るというものである。マイクロ・プロセッサの代わりにFPGAを用いたものもあるが、バス2を用いることやスイッチボックス5等の構成は同じである。

【 0 0 1 2 】

しかし、これらの方式では、消費電力が膨大なものになり、これ以上の高速化を実現することは困難であった。また、通信路での高速性と低消費電力を実現するために広帯域のデータ通信が可能な光通信を使用することが考えられているが高価なものになる。また、高価なスーパーコンピュータとなるので金銭面から多くの台数を用意することは、難しく多人数の利用者が予約して時分割で使用するため、利用できるまでの待ち時間が長くなるという問題も生じている。

40

【 0 0 1 3 】

また特許文献1に記載されたhw/sw複合体のアプリケーション例には、これまでhwModuleのネットワーク対応に関する研究がなされてきた。しかし、大規模並列化を要求する、数値シミュレーションなどのアプリケーションには、計算機間を非常に高速にデータ転送することを必要とするものもあり、その際にバスネックがボトルネックとなって必ずしも

50

H P Cとしての要求性能を満たせなかった。すなわち、内部演算速度の向上に対してデータ転送は、通信路の電気特性に律束されるので、安価で高速の通信路を得ることができなかった。

【 0 0 1 4 】

本発明は、このような状況に鑑みてなされたものであり、個人用に使える安価で高性能な大規模演算用のコンピュータを提供できるようにする。

【課題を解決するための手段】

【 0 0 1 5 】

本発明の情報処理装置は、演算対象の構造に合わせて複数の演算装置を配置し、演算実行時はバスを介在させず演算装置間で直接データ転送し、各演算装置から演算結果を取り出してホストプロセッサへ送出またはホストプロセッサからデータを入力するときのみバスを使用するようにしたものである。

具体的には、本発明の一側面の情報処理装置は、メモリと、演算装置アレイと、ホストプロセッサと、バスを備えるように構成する。

メモリは、演算対象について所定の演算を行うアプリケーション・プログラムを記憶する。

演算装置アレイは、複数の演算装置から構成される。この複数の演算装置は、前記演算対象の各々の問題領域に対応して隣接する演算装置間で直接データ通信可能に接続されて配置され、また前記アプリケーション・プログラムの実行に用いられ前記各々の問題領域に対応した演算を行う演算回路が再構成され、隣接する演算装置間で前記問題領域についての演算結果データを送受信する。

ホストプロセッサは、前記アプリケーション・プログラムを実行し、前記演算装置アレイを構成する各演算装置から各問題領域についての演算結果データを取得し、前記演算対象について演算結果を算出する。

前記バスは、前記メモリ、前記ホストプロセッサおよび前記演算装置アレイとの間で、データを通信する。

【 0 0 1 6 】

本発明の一側面の情報処理装置によれば、演算対象の構造に合わせて複数の演算装置が配置され、演算実行時はバスを介在させず演算装置間で直接データ転送が行われる。そして、各演算装置で演算された演算結果は、取り出してバスを介してホストプロセッサへ送出される。全ての演算装置は隣接演算装置に対してのみデータを転送する処理を実行すればよく、データ転送区間が短くなる。

【 0 0 1 7 】

また、本発明の一側面の仮想回路書き込み方法は、演算対象の各々の問題領域に対応して隣接する演算装置間で直接データ通信可能に接続されて配置された複数の演算装置の各々に対して、ホストプロセッサの指示に基づき演算回路を書き込むための仮想回路データ書き込み回路を所定の順序で書き込むステップを行う。次に、全ての演算装置について前記仮想回路データ書き込み回路の書き込みが終了後、前記仮想回路データ書き込み回路が最後に書き込まれた演算装置から最初に書き込まれた演算装置まで前記全ての演算装置について前記演算回路を書き込むステップを行う。

【 0 0 1 8 】

また、本発明の一側面の仮想回路書き込み方法によれば、演算装置アレイを構成する複数の演算装置に対して、まず演算回路を書き込むための仮想回路データ書き込み回路が書き込まれた後、前記仮想回路データ書き込み回路が最後に書き込まれた演算装置から最初に書き込まれた演算装置まで全ての演算装置について前記演算回路が書き込まれる。そのため、演算回路が途中で書き込めなくなるといことが生じる恐れがない。

【発明の効果】

【 0 0 1 9 】

以上のように、本発明によれば、個人用に使える安価で高性能な大規模演算用のコンピュータを実現することができる。それにより、個人でも用途に応じて手軽に大規模演算可

10

20

30

40

50

能なコンピュータを利用でき、待ち時間も減らすことができる。

【発明を実施するための最良の形態】

【0020】

以下、本発明を実施するための最良の形態の例について、添付図面を参照しながら説明する。説明は下記項目に従って順に行うとする。

1. 本発明の一実施の形態に係る演算装置アレイの概念
2. 隣接する演算装置との接続
3. 情報処理装置の概要
4. 情報処理装置の全体構成
5. ホストプロセッサの構成
6. 演算装置（演算ボード、書き込み・入出力ボード）の構成
7. 演算装置アレイによる演算処理
8. 演算装置への書き込み処理
9. 本発明の他の実施の形態に係る演算装置アレイ
10. 本発明のさらに他の実施の形態に係る演算装置アレイ

10

【0021】

[1. 本発明の一実施の形態に係る演算装置アレイの概念]

本発明に係る情報処理装置は、特許文献1（特開2003-208311号公報）に記載の技術をスーパーコンピュータ等のHPCに適用し、大規模演算をスケーラブルに行える基盤として、演算対象（対象問題）の構造を反映させて多数のPLD同士を相互接続したPLDアレイ（演算装置アレイ）を構成したものである。演算装置アレイは、PLDとして例えば大規模FPGAを搭載し、外部IO（入出力部）を大量に装備した小型のFPGAカード（演算装置）を、格子状に並べた構成を取る。

20

【0022】

すなわち、演算対象が1次元、2次元、3次元の構造である場合、それらの構造に合わせて隣接する演算装置を直接接続して1次元、2次元、3次元に配置し、演算対象において生じる数値あるいはそのデータ変化を隣接した演算装置に直接転送する。したがって、全ての演算装置は隣接演算装置に対してのみデータを転送する処理を実行すればよく、データ転送区間が短くなり、データ転送処理が情報処理装置の処理能力のボトルネックになることを回避する。また電力消費が少なくなる。

30

【0023】

FPGA等のRC-LSI（Reconfigurable LSI）は、演算対象に合わせて回路を構成することができるため、専用回路を用いた高性能な演算手段を比較的安価に手に入れることができる。なお、以下に述べる実施の形態の例では、演算装置にFPGAを利用した例を説明するが、広義の再構成可能なPLDであればこの例に限られるものでなく、CPLD（Complex Programmable Logic Device）等を適用できる。

【0024】

図3は、演算対象の物理的または論理的な構造の一例を示すものである。

演算対象6は、流体モデルや液体モデルなどであり、1次元、2次元および3次元の任意の構造を取り得る。演算対象の具体例としては、例えば天文学や物理学分野において現在進められているALMAプロジェクトや重力波検出、さらには、次世代核融合実証炉等における炉設計、高エネルギー物理学分野においては、次世代加速器の内部反応予測シミュレーション、地球科学分野における様々な可視化シミュレーション（具体的には断層モデルの可視化等）、気象学的には中長期予報のためのシミュレーション分野の拡充等が要望として上がってきている。また、宇宙航空分野においては、人工衛星の機能設計や国際宇宙ステーションにおけるシミュレーション、将来の月への基地建設、有人火星探査におけるリスクシミュレーション等の分野からの要望も生じている。

40

【0025】

例えば、電磁気学や半導体工学などでよく利用される方程式の一つにポアソン方程式があるが、これは楕円型の偏微分方程式である。この方程式を差分法で解く場合、領域内の

50

格子点数だけの連立方程式を解く必要がある。格子幅については任意でよいが、精度に反映されるため、いかに効率よく大型の連立方程式を解くかが問題となる。解きたい問題の領域について、格子点間隔を狭めれば格子点が増加し、それゆえ大規模な演算が必要となる。

【0026】

図3に示した演算対象6は3次元構造の例であり、本発明の情報処理装置を用いて当該演算対象6について演算処理を行うにあたり、この演算対象6を格子状の複数の問題領域6-1に分割する。この例では、演算対象6を、4(X方向)×3(Y方向)×3(Z方向)=36の問題領域に分割している。そして、図4に示すように、演算対象6に対し複数の演算装置40-1を、当該演算対象6の各々の問題領域6-1に対応するように配置して演算装置アレイ40を構成する。例えば斜線で表した個別の問題領域6-1の演算は、同じく斜線で表した対応する位置の演算装置40-1が担当する。したがって、この例では演算装置アレイ40は、問題領域6-1と同じ36個の演算装置40-1を用いて構成される。

10

【0027】

各々の問題領域においては、隣接した問題領域と相互作用が行われ、物理量が隣接の問題領域に伝搬されるので、その物理現象に対応して隣接した演算装置間で対応するデータ転送が行われる。具体的には、一つの問題領域内には演算の対象となる多数のデータが格子点6-1A, 6-1B, 6-1C, ...上にあり、対応する演算装置による演算が領域境界に達した場合にその演算結果が、当該演算装置から隣接する演算装置へ転送される。

20

【0028】

このように、本発明では、分割した問題領域の物理形状や特性に対応して、演算装置を割り当て、隣接する演算装置間で直接データ転送することで、高性能なデータ処理能力を維持している。

【0029】

これに対し、従来の方式では、固定的に物理構成のままであり通信路でのデータ転送がネックとなり、演算対象に対して演算装置の性能を十分に発揮することができない。また、通常の場合には、演算装置間のデータ転送がボトルネックとなり通信路の高速性能を使用できずにいる。それを改善するために光ケーブル等の最高性能の通信路を用意すると、システム全体が高価なものとならざるを得ない。

30

【0030】

本発明では、演算対象に応じて演算装置間を接続接続する通信路がスケーラブルに増大するので、高性能・高価格の通信路を用いる必要がない。また、データ転送先が隣接する演算装置までのため、通信路に掛かる処理負荷も小さく、消費電力も少なくすることができる。

【0031】

[2. 隣接する演算装置との接続]

本発明の情報処理装置においては、上記のとおり隣接する演算装置間で直接データ転送できるように構成している。以下、この隣接する演算装置との接続の形態について、図5~図7を参照して説明する。

40

【0032】

図5、演算装置(演算ボード)のコネクタ配置を示す図である。図6は、演算装置(演算ボード)の相互接続を示す図である。さらに、図7は、演算装置の分解斜視図である。

【0033】

本実施の形態における演算装置40-1は、図5~図7に示すように、上下に配置された1対の演算ボード50と書き込み・入出力ボード60(特許請求の範囲に記載された書き込みボードの一例)から構成される。

演算ボード50は、主に演算処理を担当する演算回路等(特許請求の範囲に記載された演算回路に相当)が書き込まれるFPGA51、主記憶装置として機能するメモリ52、

50

および複数のコネクタ（接続端子）が設置された基板 50A を備える。また、書き込み・入出力ボード 60 は、演算ボード 50 の F P G A に演算回路等を書き込むための書き込み回路およびデータ転送用の入出力回路等が書き込まれる F P G A 61、主記憶装置として機能するメモリ 62、および複数のコネクタ（接続端子）が設置された基板 60A を備える。各コネクタは、演算ボード 50 の基板 50A および書き込み・入出力ボード 60 の基板 60A 上の各辺の所定位置に設けられる。演算ボード 50 と書き込み・入出力ボード 60 の機能等の詳細については後述する。

【0034】

演算ボード 50 は、これらの複数のコネクタにより相互に接続することが可能である。例えば、図 6 に示すように、演算ボード 50 は前接続用のコネクタ 53F を当該演算ボード 50 の前に配置されている演算ボード 50 の後用接続用のコネクタ 53B と直結することによって、前に配置された演算ボード 50 と直接データ転送可能になる。同様にして、後接続用のコネクタ 53B、左接続用のコネクタ 53L、右接続用のコネクタ 53R、上接続用のコネクタ 53U、下接続用のコネクタ 53D により、それぞれ当該演算ボード 50 の後ろ、左、右、上、下に位置する演算ボード 50 と直接データ転送可能に接続できる。

10

【0035】

また、演算ボード 50 は、書き込み・入出力ボード 60 と接続するためのコネクタ 54 を備え、コネクタ 54 を対応する書き込み・入出力ボード 60 のコネクタ 64 と直結することで、2つのボード間を直接データ転送可能にする。

20

【0036】

演算ボード 50 は、前後左右方向の接続についてはコネクタ直結とし、高速転送を可能とする。また隣接接続端子の上下端子すなわちコネクタは、積層に適するよう対応する同一の場所にレイアウトする。ただし、各ボードの廃熱を考えると、上下端子の接続は直結とはせずにケーブルを介して接続するのが好ましい。

【0037】

なお、書き込み・入出力ボード 60 の基板 60A において、対応する演算ボード 50 の下接続用のコネクタ 53D に対応する位置に、コネクタ 53D の形状に合わせて切り欠き 60A1 が形成してある。書き込み・入出力ボード 60 の切り欠き 60A1 が形成された辺は、コネクタ 63B、63F が設けられた辺の長さと比較して短い。この切り欠き 60A1 を通して、演算ボード 50 の下接続用のコネクタ 53D が対応する書き込み・入出力ボード 60 の基板 50A を貫通し、その下に配置された演算ボード 50 と接続することができる。

30

【0038】

書き込み・入出力ボード 60 に切り欠き 60A1 を形成した場合、コネクタ 63B、63F が設けられた辺が短いので基板 60A の面積が小さくなり、材料の節約、コスト削減に繋がる。なお、演算ボード 50 のコネクタ 53D が書き込み・入出力ボード 60 を貫通できればよいので、例えば演算ボード 50 と同面積・同形状の書き込み・入出力ボード 60 に、切り欠き 60A1 に代えて貫通用の孔を設けてもよい。

【0039】

図 7 に示した例では、コネクタ 53F、53R、54 は芯線の接続部が凸であるオス型コネクタとし、コネクタ 53B、53L、64 は芯線の接続部が凹であるメス型コネクタとしているが、この例に限られるものではない。

40

【0040】

[3 . 情報処理装置の概要]

次に、上述のような構成の演算装置アレイを用いた情報処理装置の概要について説明する。

【0041】

既述のとおり、本発明の情報処理装置は、特許文献 1 に記載された半導体回路制御装置を応用して構成する。つまり、本発明においては、F P G A 等の P L D の内部構成には関

50

与しないが、動的に回路（いわゆる「仮想回路」）が書き換え可能な R C - L S I を用いる。そして、複数の R C - L S I（演算装置に相当）を演算対象の構造に合わせて配置し、かつ、隣接する R C - L S I を相互に直接接続して演算装置アレイを構成することにより、R C - L S I 間でバスを用いず直接データ転送可能に構成する。

【 0 0 4 2 】

このように構成することにより、本発明においても、特許文献 1 に記載された半導体回路制御装置と同様に、仮想回路をオブジェクトでラッピングしたハードウェア・オブジェクトモデルは通常のソフトウェア・オブジェクトと外見は同様であり、自由にプログラムの中で使用できる。オブジェクト・ライブラリ中のオブジェクトから派生する方法を踏襲し、ハードウェア・オブジェクト・ライブラリを用意する。並列性が重視される画像認識、音声認識などの処理、常時観測が必要な処理など、専用回路に適した処理はハードウェア・オブジェクトとしてライブラリから R C - L S I に読み出して処理する。ただし、ハードウェア・オブジェクトは回路であるので、所定の同期信号に基づいてタイミングや同期の制御は行わなければならない。また、仮想回路を包み込むオブジェクトを演算対象に合わせて必要数だけ確保できるように、メモリや R C - L S I を演算装置に用意する。

【 0 0 4 3 】

図 8 は、本発明の一実施の形態に係る情報処理装置の概要を示す図である。

本実施の形態における情報処理装置は、例えば、C P U などからなるホストプロセッサ 1 1 と、実際のメモリ空間を構築するメモリ 1 2 と、仮想回路空間を構築する P L D 1 3 とを備える。

【 0 0 4 4 】

アプリケーション（アプリケーション・プログラム）2 1 は、ソフトウェアで実現してあるオブジェクトの他に最適な性能を実現する回路を含んだハードウェア・オブジェクト 2 6 も含まれている。アプリケーション 2 1 を実行する計算機は、メモリ空間を実現するメモリ（メモリ素子）1 2 上にアプリケーションのソフトウェア・プログラム部分を配置し、これとソフトウェアを実行するホストプロセッサ 1 1 をシステムバスで接続している。同時にホストプロセッサ 1 1 には複数の P L D 1 3 を接続した標準バスが接続されている。これらのバスを総称してバス 1 4 と表記する。

【 0 0 4 5 】

この P L D 1 3 はアプリケーションが起動され、動作中にハードウェア・オブジェクト 2 6 が起動されると、その回路部分のハードウェア・ネット 2 9 が書き込まれる実ハードウェア部品である。メモリ素子と同様に多数の L S I で回路空間を構成する。P L D 1 3 は、後述するハードウェア・モジュール 3 0 に搭載される F P G A および演算装置アレイ 4 0 を構成する演算装置 4 0 - 1 に搭載された F P G A（図 7 参照）に相当する。

【 0 0 4 6 】

アプリケーション 2 1 が起動されるとメモリ 1 2 内にアプリケーション 2 1 がまず配置される。プログラムの処理が進みハードウェア・オブジェクト（hwObject）2 6 を生成するコンストラクタ文が実行されると、メモリ 1 2 と P L D 1 3 とに跨って（両方でセットとして）ハードウェア・オブジェクト 2 6 が配置される。ここでは、hwObject - 1 と hwObject - 2 のコンストラクタ文が実行されて二つのハードウェア・オブジェクト 2 6 が生成されている。アプリケーション 2 1 は、hwObject - 1 及び hwObject - 2 を含む。各ハードウェア・オブジェクト 2 6 に対応して、ハードウェア・ドライバ（hwDD）2 7 及びハードウェア・ネット 2 9 が存在する。ハードウェア・ドライバ 2 7 は、メモリ 1 2 内に生成される。一方、ハードウェア・ネット 2 9 は、仮想回路空間である P L D 1 3 内に、直接データ転送可能に生成され且つ消去可能とされる。

【 0 0 4 7 】

図 9 は、ホストプロセッサとハードウェア・ネット間の制御概要の流れを示す図である。

ここでは、ハードウェア・オブジェクト 2 6 に対する読み書きの要求がアプリケーション 2 1 から、順次、要求を伝えてハードウェア・ネット 2 9 にまで伝えて応答を返すまで

10

20

30

40

50

処理が続く。この時、ハードウェア・ネット 29 は並列動作を行う。アプリケーション 21 はハードウェア・オブジェクト 26 に実行を命令して、返答を待たずに、次の処理、例えば、別のハードウェア・オブジェクト 26 の実行を行っていく。ハードウェア・ネット 29 の処理が終了したかどうかを調べる処理を行いハードウェア・オブジェクト 26 のメンバー変数の値を読み取る。このように、アプリケーション 21 のレベルでの並列処理にあった形で、ハードウェア・オブジェクト 26 すなわち演算回路を含む組み込み回路の並列性を適切に利用することができる。

【0048】

以下に、情報処理装置の基本的な動作を説明する。

まず、アプリケーション 21 が起動されると、OS はアプリケーション 21 の実行に必要な領域をアプリケーション 21 が排他的に使用できるように確保して制御をアプリケーション 21 に渡す。アプリケーション 21 の起動・初期化プログラムは、プログラムで使われるイベント管理、メッセージ管理等の OS との通信やアプリケーション 21 内での管理に必要な基盤部分の立ち上げを行う。なお、ハードウェア・オブジェクト 26 の管理に必要な管理制御部分もこの時にアプリケーション部に組み込まれる。

【0049】

アプリケーションの動作中にハードウェア・オブジェクト 26 を生成する文が実行されると、ハードウェア・オブジェクト 26 がメモリ領域に I/O 処理部であるハードウェア・ドライバ 27 を含んで生成される。PLD 13 には同時に回路としてのハードウェア・ネット 29 が書き込まれ、ハードウェア・オブジェクト 26 が排他的に使用できるように設定される。このハードウェア・ネット 29 の回路データは、例えば、このハードウェア・ネット 29 の動作仕様を動作記述言語により記述し、設計自動化ツールである高位・論理合成、配置・配線ツールを使い PLD 13 に書き込まれる回路データを事前に作成して、回路ライブラリに登録しておくことができる。ここで、事前にハードウェア・ネット 29 を書き込んでおき、ハードウェア・オブジェクト 26 の生成時に回路を単に活性化させ、回路の書き込み時間を短縮させることは容易に考えられる。

【0050】

ハードウェア・オブジェクト 26 は一度生成されると、ハードウェア・オブジェクト 26 を消滅させる文が実行されるかアプリケーション 21 が終了するまで存在し続ける。アプリケーション 21 の動作中にハードウェア・オブジェクト 26 に対する読み書きを行う文が実行されると、それがハードウェア・ネット 29 に関係するメンバー変数やメンバー関数に対する場合には、ハードウェア・ドライバ 27 を介してハードウェア・ネット 29 に対する I/O 処理を行う。

【0051】

[4 . 情報処理装置の全体構成]

図 10 は、情報処理装置の全体構成を示す図である。

本実施の形態における情報処理装置は、例えば、ホストプロセッサ 11 と、主記憶装置としてのメモリ 12 と、仮想回路空間であるハードウェア・モジュール 30 と、演算対象の各問題領域について演算を行う演算装置アレイ 40 とを備える。

演算装置アレイ 40 はハードウェア・モジュールの一種であり、図 4 に示した例では 3 次元構造であるが、ここでは説明の便宜のため 2 次元構造で表現してある。また、ホストプロセッサ 11、メモリ 12、およびハードウェア・モジュール 30 は、所定の規格 (PCI 規格等) のバス 14 を通じて高速なデータ転送が可能となっている。

【0052】

図に示すように、ソフトウェア・プログラムの大半はメモリ 12 に置かれ、ホストプロセッサ 11 により実行される。各種制御・処理のための回路は仮想化され、ハードウェア・ネット (hwNet) として、バス 14 上に置かれたハードウェア・モジュール 30 と名付けたハードウェア部品中に設けられ、ホストプロセッサ 11 が必要となった時点に一時的に書き込まれる。そして、ハードウェア・モジュール 30 に書き込まれた仮想回路によって、演算装置アレイ 40 の各演算装置 40 - 1 内の FPGAs に演算回路等の仮想回路が書

き込まれる。ハードウェア・モジュール 30 および各演算装置 40 - 1 に書き込まれた仮想回路は、不要になれば消去または再利用のために初期化される。

【0053】

この生成・消去の手順の詳細は後述するが、ソフトウェアのオブジェクトと同じようにコンストラクタとデストラクタ演算子で行われる。ハードウェア・モジュール 30 および演算装置アレイ 40 はメモリ型のデバイスとして認識され、ハードウェア・オブジェクト 26 への読み書きはメモリ 12 上に作られたオブジェクトと同じように行われる。メモリ空間上に自由にオブジェクトが作れるのと同じように、ハードウェア・オブジェクト 26 が自由に作られる仮想回路空間を実部品として提供するのがハードウェア・モジュール 30 および演算装置アレイ 40 である。したがって、特に演算装置アレイ 40 は演算対象に 10
合わせて大きな仮想回路空間を張るように多数の演算装置 40 - 1 を事前に用意する。ハードウェア・オブジェクト 26 を使った計算モデルは仮想回路をオブジェクトとして扱い、アプリケーション 21 にハードウェア・ネット 29 を埋め込むインターフェースを提供するものである。

【0054】

本実施の形態では、再構成可能システムを構成する一要素として、図 10 に示したようなハードウェアボードである、ハードウェア・モジュール 30 を導入している。ハードウェア・モジュール 30 は、例えば、標準バス・インターフェース (BI) 31、ローカルメモリ (LM) 32、ローカルプロセッサ (LP) 33、FPGA 34、入出力インターフェース 35 をデータ転送可能に接続する構成になっている。 20

【0055】

LP 33 は、FPGA 等の RC - LSI で構成され、ハードウェア・モジュール 30 内の制御を行う。すなわち LP 33 によって LM 32 を利用しつつハードウェア・ネット (仮想回路データ書き込みのための書き込み回路の一例) が JTAG 方式で FPGA 34 に書き込まれる。FPGA 34 は、入出力インターフェース 35 および通信路 36 を介して演算装置アレイ 40 の各演算装置 40 - 1 内の FPGA にハードウェア・ネット 29 (演算回路等の仮想回路) を書き込む。演算装置アレイ 40 の各演算装置 40 - 1 に書き込まれた仮想回路のそれぞれが、ハードウェア・ネット 29 として機能する。

【0056】

BI 31 は、後述するハードウェア・ドライバの制御命令によりホストプロセッサ 11 とハードウェア・ネット 29 (FPGA 34、演算装置アレイ 40) との通信を制御する。ハードウェア・モジュール 30 (演算装置アレイ 40 を含む) はメモリ・デバイスとして認識され、メモリと同じような形態でバス 14 (例えば、PCI) に接続して情報処理装置 (計算機) に組み込まれる。この時、OS によりハードウェア・モジュール・ドライバと呼ぶデバイス・ドライバが演算装置アレイ 40 の各演算装置 40 - 1 に組み込まれ、メモリ領域が割り当てられる。多数のメモリ・チップが一樣且つ平坦なメモリ空間を構成するように多数のハードウェア・モジュール (演算装置 40 - 1) 内の RC - LSI は仮想回路空間を構成する。 30

【0057】

なお、LP 33 および FPGA 34 は、ホストプロセッサ 11 から送信されたデータを演算装置アレイ 40 へ転送し、また、演算装置アレイ 40 から送信されたデータをホストプロセッサ 11 へ転送する入出力回路としての機能も備える。 40

【0058】

図 10 に示すようなハードウェア・モジュール 30 を介してホストプロセッサ 11 からの仮想回路データを演算装置アレイ 40 の各演算装置 40 - 1 の FPGA に転送することで、演算装置アレイ 40 の各演算装置 40 - 1 にそれぞれハードウェア・ネット 29 を配置することができる。詳細は後述するが、例えば演算装置「00」の FPGA 51 に書き込まれた最初の hwNet - 1 の出力を演算装置「01」に転送し、それを演算装置「01」の FPGA 51 に書き込まれた hwNet - 2 が読み込む事で、hwNet - 1 と hwNet - 2 (図 8 参照) とは同時に動作してパイプライン処理を行う事ができる。 50

【 0 0 5 9 】

また、L P 3 3 によりホストプロセッサ 1 1 の負担を減らす事ができる。すなわち、各演算装置 4 0 - 1 の F P G A 5 1 に書き込まれる使用頻度が高いハードウェア・ネット 2 9 の書き込みは、B I 3 1 を通してこの L P 3 3 に書き込み命令を送る事で可能である。

【 0 0 6 0 】

[5 . ホストプロセッサの構成]

図 1 1 に、ホストプロセッサの階層構成図を示す。

ホストプロセッサ 1 1 は、アプリケーション 2 1、オブジェクト・マネジャー 2 2、ハードウェア・モジュール・ドライバ 2 3、O S 2 4、バス 2 5 を備える。この例では、アプリケーション 2 1 を 3 つ含む例を示しているが、適宜の数のアプリケーションを有することができる。各アプリケーション 2 1 は、ひとつ又は複数のハードウェア・オブジェクト 2 6、ハードウェア・ドライバ 2 7、アプリケーション 2 1 とハードウェア・ドライバ 2 7 との入出力を制御するインターフェース 2 8 の組を有する。

【 0 0 6 1 】

各ハードウェア・ドライバ 2 7 は、ハードウェア・ネット 2 9 毎に定義され、(ハードウェア・モジュール 3 0 を介して)ハードウェア・ネット 2 9 の入出力動作を制御する。ハードウェア・ドライバ 2 7 には、例えば、hwNet の端子情報、書き込みや読み込み、イネーブル、アウトプット・イネーブルなどの hwNet の制御情報、hwObject 番号、hwNet 番号、hwModule (演算装置) 番号、割り当て P L D 番号 (演算装置と hwModule が一対一に対応する場合は不要)、hwNet 割り当て端子番号、局所 (ローカル) メモリ割り当てアドレス、局所メモリ割り当て領域サイズ、hwNet 状態、hwNet 命令、主メモリでの hwNet 通信領域アドレス、hwNet 通信領域サイズ、hwNet 通信領域カレント・アドレスなどの通信制御情報が組み込まれている。これらの情報は、ハードウェア・ネットの回路情報と共にハードウェア・ネットライブラリに保存されている。ハードウェア・オブジェクト 2 6 が生成されてハードウェア・ネット 2 9 がロードされる時には、hwModule (演算装置) 番号、hwNet 番号、P L D 番号などを取得してハードウェア・ドライバ 2 7 がハードウェア・オブジェクト 2 6 の一部として生成される。

【 0 0 6 2 】

ハードウェア・モジュール 3 0 および演算装置アレイ 4 0 を情報処理装置 (計算機) に組み込んだ時、図に示すハードウェア・モジュール・ドライバ 2 3 は添付されたデバイス情報をもとに O S 2 4 にデバイス・ドライバとして永続的に登録される。このハードウェア・モジュール・ドライバ 2 3 は、計算機のバスに接続されたハードウェア・モジュール 3 0 に対する通信を制御している。例えば、バス 1 4 として P C I バスを使う時には P C I デバイス情報などが O S から割り当てられる。情報処理装置を立ち上げる時には常にハードウェア・モジュール・ドライバ 2 3 は、O S に事前に組み込まれる。一方、このハードウェア・モジュール・ドライバ 2 3 に反して、ハードウェア・オブジェクト 2 6 のハードウェア・ドライバ 2 7 は、ハードウェア・ネット 2 9 が存在する時だけハードウェア・モジュール・ドライバ 2 3 の中に組み込まれ、O S 2 4 がハードウェア・ドライバ 2 7 を感知することはない。他方、ハードウェア・オブジェクト 2 6 側は、ハードウェア・ドライバ 2 7 を O S に組み込まれたデバイス・ドライバのようにみなしてハードウェア・ネット 2 9 に対するアクセスを行う。このとき、どのハードウェア・モジュール 3 0 および演算装置アレイ 4 0 に対するアクセスを行うかなどは考慮する必要がない。

【 0 0 6 3 】

このように、本発明では、O S 2 4 側からは、ハードウェア・モジュール・ドライバ 2 3 が常時組み込まれるデバイス・ドライバとして安定して制御及び監視を行う事ができるので、システムの安定性を保証する事ができる。他方、アプリケーション 2 1 側からは、ハードウェア・ドライバ 2 7 が必要なときにだけ割り当てられるので、ハードウェア・ネット 2 9 を使う自由度が大幅に増すことになる。

【 0 0 6 4 】

[6 . 演算装置 (演算ボード、書き込み・入出力ボード) の構成]

図 1 2 は、演算装置（演算ボード、書き込み・入出力ボード）の仮想回路書き込み後の機能を示すブロック図である。以下、図 1 2 を参照して、演算ボード 5 0、書き込み・入出力ボード 6 0 の順に説明する。

【 0 0 6 5 】

演算ボード 5 0 は少なくとも、仮想回路空間が形成される F P G A 5 1 と、メモリ空間が形成されるメモリ 5 2 を備える。メモリ 5 2 には、ハードウェア・モジュール 3 0 を介してホストプロセッサ 1 1 から送信された演算対象の問題領域に対応する仮想回路データや演算対象データ、各回路による演算結果データ等が一時的に格納される。また、メモリ 5 2 は、ある問題領域における演算中でない格子点に関する情報など、演算に必要な情報を一時的に格納したりもする。

10

【 0 0 6 6 】

F P G A の内部構造は周知であり、例えば、任意の論理を構成可能な 4 入力程度の組合せ回路と順序回路からなる論理ブロックが、格子状に多数配置され、その間の配線を簡易なスイッチブロックで接続し、スイッチを切り替えることで所望の機能を持つ仮想回路を再構成するようにしている。そして、仮想回路が再構成された論理ブロックにより、入出力ブロック I / O を介してデータの入力および演算結果の出力等が行われる。

【 0 0 6 7 】

本実施の形態における F P G A 5 1 は、メモリ制御回路 5 5、並列演算回路 5 6（演算回路の一例）、演算対象データ入出力回路 5 7、演算結果データ入出力回路 5 8、隣接 F P G A データ転送回路 5 9 X、5 9 Y、5 9 Z を含む回路が書き込まれる。並列演算回路 5 6 を含む全ての回路は、書き込み・入出力ボード 6 0 に搭載された F P G A 6 1 により書き込まれる。なお、以降、演算回路を含むこれらの回路を総称して「演算回路等」という。

20

【 0 0 6 8 】

メモリ制御回路 5 5 は、メモリ 5 2 に記憶された各種データの読み出しおよび書き込みを行い、演算ボード 5 0 上のメモリ 5 2 と各仮想回路とのデータ転送を実現する回路である。

【 0 0 6 9 】

並列演算回路 5 6 は、複数の論理ブロック 5 6 A、5 6 B・・・を備え、演算対象の問題領域に対応した演算を行う回路である。例えば、演算対象が連続系の物理問題のときには、問題領域の各格子点上に定義された微分方程式を離散化した連立方程式を計算する並列回路が再構成される。また、演算対象が多分岐とグラフ構造で示される場合には、ノード分岐を辿りながら計算を行う並列回路が再構成される。

30

【 0 0 7 0 】

問題領域内の格子点（ i, j, k ）における物理量 f は、演算対象が 3 次元の場合、次式で表され、隣接格子上の物理量で与えられる方程式が定義できる。

$$f(i, j, k) = F(i-1, j-1, k-1, i, j, k, i+1, j+1, k+1)$$

【 0 0 7 1 】

今、問題領域の 1 辺の格子点数を N とすると、問題領域内の格子点数は 3 次元で増加するのに対し、境界面では 2 次元で増加するため、格子点での演算量は N の 3 乗に比例（ $\sim N^3$ ）し、境界面では $6 / N^2$ に比例する。よって、演算量に関して格子点 / 境界面（ $\sim N^3 / 6 N^2$ ） $= N / 6$ である。したがって、両者の演算量と演算手段とのバランスをとるには、 N 個の格子点用の並列演算回路に対する境界面用のその数を $N / 6$ 以上とすればよい。例えば、論理ブロック 5 6 A が領域内の格子点に対応し、論理ブロック 5 6 B が境界面に対応するとすれば、論理ブロック 5 6 A の数を論理ブロック 5 6 B の数の $N / 6$ 以上とする。

40

【 0 0 7 2 】

演算対象データ入出力回路 5 7 は、ホストプロセッサ 1 1 の制御下でハードウェア・モジュール 3 0 から送られる演算対象データが書き込み・入出力ボード 6 0 を介して入力される回路である。入力された演算対象データは、メモリ制御回路 5 5 を介してメモリ 5 2

50

に記憶され、並列演算回路 5 6 等による処理に利用される。演算対象データには、例えば演算対象の範囲、問題領域、演算結果の転送先、演算条件等、演算に使用するパラメータが含まれる。

【 0 0 7 3 】

なお、演算対象データのパラメータ（演算条件、演算対象、問題領域等）は、メモリ 5 2 に記憶させている途中で、つまりホストプロセッサ 1 1 から送られる段階で変更になることも考えられる。そのような場合、演算対象データ入出力回路 5 7 は、メモリ 5 2 に途中まで記憶させた演算対象データに、メモリ 5 2 における当該演算対象データが格納されたアドレス情報を付加してホストプロセッサ 1 1 へ送信するようにしてもよい。このようにした場合、ホストプロセッサ 1 1 は、各演算装置の変更後の演算対象データと、各演算装置から戻ってきた演算対象データとを比較し、真に変更された演算対象データについてのみ該当演算装置へ再送すればよいので、通信路のリソースの節約および再送処理に要する時間を短縮できる。

【 0 0 7 4 】

演算結果データ入出力回路 5 8 は、当該演算ボード 5 0 の並列演算回路 5 6 等による演算完了後の演算結果データ、あるいは演算に使用するパラメータ等のデータを、メモリ制御回路 5 5 を介してメモリ 5 2 に記憶したり、対応する書き込み・入出力ボード 6 0 へ送信したりする回路である。この演算結果データには、並列演算回路 5 6 による演算結果に加え、当該演算ボード（演算装置）の番号、問題領域の各格子点情報とそれに対応する演算結果データ等が含まれる。なお、演算結果データは、演算対象データに含まれる「演算結果の転送先」に基づいて隣接する所定の演算ボードへ転送される。

【 0 0 7 5 】

なお、演算中の演算対象データのパラメータ変更やシミュレーション方法の変更などの理由により、演算を途中で停止することがある。このような場合、演算結果データ入出力回路 5 8 は、メモリ 5 2 に途中まで記憶させた演算結果データに、メモリ 5 2 における当該演算結果データが格納されたアドレス情報を付加してホストプロセッサ 1 1 へ送信するようにしてもよい。このようにした場合、ホストプロセッサ 1 1 は、各演算装置による途中までの演算結果を利用して演算対象の解析を行うことができる。さらに、その途中までの演算結果をホストプロセッサ 1 1（メモリ 1 2）から読み込むようにしてもよい。それにより、途中まで実施した演算結果を流用することができるので、新たに演算が必要な部分のみ演算を行えばよく、演算に要する時間を短縮できる。

【 0 0 7 6 】

隣接 F P G A データ転送回路 5 9 X は、各コネクタを介して X 方向（前後）に隣接する演算ボード 5 0 との間でデータの送受信を行い、並列演算回路 5 6 へ転送する回路である。この隣接する演算ボード 5 0 との間でやり取りするデータには、隣接する演算ボード 5 0 による演算結果に加え、領域境界を跨ぐデータの送受信を実現するため、送信元の演算ボード（演算装置）の番号、各々の格子点情報とそれに対応する演算結果データ等が含まれる。同様にして、隣接 F P G A データ転送回路 5 9 Y は Y 方向（左右）に隣接する演算ボード 5 0 と、隣接 F P G A データ転送回路 5 9 Z は Z 方向（上下）に隣接する演算ボード 5 0 とデータの送受信を行う。なお、各隣接 F P G A データ転送回路 5 9 X, 5 9 Y, 5 9 Z から隣接する演算装置 4 0 - 1 へのデータ転送は、ハードウェア・モジュール 3 0 を介してホストプロセッサ 1 1 から送られるクロック信号（C L K）に基づき全演算装置 4 0 - 1 でタイミングを合わせて行われる。

【 0 0 7 7 】

次に、書き込み・入出力ボード 6 0 について説明する。図 6 に示したように隣接する演算ボード 5 0 同士を接続し、書き換え可能な半導体装置に演算回路等を書き込むためには、書き込みを行う書き込み回路が必要である。

【 0 0 7 8 】

書き込み・入出力ボード 6 0 は少なくとも、仮想回路空間が形成される F P G A 6 1 と、メモリ空間が形成されるメモリ 6 2 を備える。メモリ 6 2 には、ハードウェア・モジュ

10

20

30

40

50

ール 30 を介してホストプロセッサ 11 から送信された演算対象の問題領域に対応する仮想回路データや演算対象データ、また演算ボード 50 から送られてくる演算結果データや各回路の演算結果データ等が一時的に格納される。

【0079】

本実施の形態における FPG A 61 は、ハードウェア・モジュール 30 の FPG A 34 の制御の下、メモリ制御回路 65、書き込み回路 66（仮想回路データ書き込み回路の一例）、仮想回路データ入出力回路 67、各種データ入出力回路 68 が書き込まれる。以降において、各種データ入出力回路 68 を「演算回路等」に含める場合もある。

【0080】

メモリ制御回路 65 は、メモリ 62 に記憶された各種データの読み出しおよび書き込みを行い、書き込み・入出力ボード 60 上のメモリ 62 と各仮想回路とのデータ転送を実現する回路である。

10

【0081】

書き込み回路 66 は、ホストプロセッサ 11 の制御下でハードウェア・モジュール 30 から送信される演算対象の問題領域に対応する仮想回路データに基づいて、演算ボード 50 の FPG A 51 に各仮想回路を書き込む回路である。ホストプロセッサ 11 からの指示内容によって、問題領域ごとに各演算ボード 50 の FPG A 51 に書き込む仮想回路を変える場合もあれば、仮想回路の論理構成は同一で演算対象パラメータを変えることで問題領域ごとに異なる演算結果を得るような場合もある。

【0082】

仮想回路データ入出力回路 67 は、ハードウェア・モジュール 30 から送信される仮想回路データの入力、および当該仮想回路データを指定された書き込み・入出力ボード 60 へ転送するべく制御する回路である。

20

【0083】

各種データ入出力回路 68 は、ハードウェア・モジュール 30 から送られてきた仮想回路データや演算対象データを受信し、メモリ制御回路 65 を介してメモリ 62 に記憶したり、対応する演算ボード 50 へ転送したりする回路である。また、対応する演算ボード 50 から送られてくる問題領域の演算結果データや各回路の演算結果データ等を、ハードウェア・モジュール 30 を介してホストプロセッサ 11 へ送信する回路である。

【0084】

なお、書き込み・入出力ボード 60 の書き込み回路 66 および仮想回路データ入出力回路 67 は、対応する演算ボード 50 への演算回路等の書き込みが終了した時点で消去してもよい。このようにした場合、書き込み・入出力ボード 60 の FPG A 61 のリソースを節約することができる。勿論、FPG A 61 へ各種データ入出力回路 68 を書き込んでもなおリソースに余裕があれば、書き込み回路 66 および仮想回路データ入出力回路 67 を残しておいてもよい。

30

【0085】

[7 . 演算装置アレイによる演算処理]

次に、図 10 に示した演算装置アレイ 40 を例に演算処理の概要を説明する。

前提として、演算装置アレイ 40 を構成する各演算装置 40 - 1 の演算ボード 50 および書き込み・入出力ボード 60 に、各々が担当する演算対象の問題領域に合わせて hwNet29 である仮想回路（図 12 参照）が書き込まれた状態であるとする。演算を行う順番は、演算対象の物理現象や特性等に基づいて決定される。ここでは演算装置「00」、演算装置「10」、演算装置「20」から演算を開始して横方向へ順に移動していき、末端の演算装置「03」、演算装置「13」、演算装置「23」で演算を終了する場合を想定する。

40

【0086】

演算装置アレイ 40 を構成する 12 個の演算装置、演算装置「00」～演算装置「23」の各々の演算ボード 50 に、それぞれが担当する演算対象の問題領域ごとの演算対象パラメータの初期値が入力されメモリ 52 に一時記憶される。また、メモリ 52 は、当該演

50

算装置が転送すべき「演算結果の転送先」の情報を記憶している。

【0087】

演算装置「00」,「10」,「20」の演算ボード50における並列演算回路56（図12参照）は、ハードウェア・モジュール30を介して入力されるホストプロセッサ11からの指示を受けて、あるいは所定の信号や条件をトリガとするタイミングで、各々の対応する問題領域について演算を開始する。生成された各演算結果データはそれぞれのメモリ52に記憶する。

【0088】

続いて、演算装置「00」,「10」,「20」はメモリ52に格納した各々の演算結果データを、ホストプロセッサ11（または図示せぬクロック発生部）が発信するクロック信号（CLK）に同期して、隣接FPGAデータ転送回路59X（図12参照）からコネクタ53R（図7参照）を介してそれぞれ「演算結果の転送先」である演算装置「01」,「11」,「21」へ転送する。

【0089】

演算装置「01」,「11」,「21」では、各隣接FPGAデータ転送回路59Xがコネクタ53Lを介して演算装置「00」,「10」,「20」からそれぞれ演算結果データを受信する。そして、受信した演算結果データと演算対象パラメータ値、及び各演算装置に保存された過去の演算時点での演算結果を用いて、演算装置「01」,「11」,「21」の並列演算回路56が新たな演算結果データを生成し、それぞれのメモリ52に記憶する。

【0090】

同様にして、演算装置「01」,「11」,「21」は各々の演算結果データを、「演算結果の転送先」の情報に従って演算装置「02」,「12」,「22」に転送する。演算装置「02」,「12」,「22」では、演算装置「01」,「11」,「21」から受信した各々の演算結果データと演算対象パラメータ値、及び各演算装置に保存された過去の演算時点での演算結果を用いて、並列演算回路56が新たな演算結果データを生成し、それぞれのメモリ52に記憶する。

【0091】

また、演算装置「02」,「12」,「22」は各々の演算結果データを、「演算結果の転送先」の情報に従って演算装置「03」,「13」,「23」に転送する。演算装置「03」,「13」,「23」では、演算装置「02」,「12」,「22」から受信した各々の演算結果データと演算対象パラメータ値、及び各演算装置に保存された過去の演算時点での演算結果を用いて、演算装置「03」,「13」,「23」の並列演算回路56が新たな演算結果データを生成し、それぞれのメモリ52に記憶する。

【0092】

演算装置「00」～演算装置「23」の全ての演算装置で演算処理が終了した後、演算結果データをホストプロセッサ11へ送信する処理を実行する。まず演算装置「00」～演算装置「23」の各々の演算ボード50における並列演算回路56は、担当する問題領域について演算結果データを生成した後、当該演算結果データを演算結果データ入出力回路58からコネクタ54を介して対応する書き込み・入出力ボード60へ転送する。

【0093】

演算装置「00」～演算装置「23」の各々の書き込み・入出力ボード60において、各種データ入出力回路68がコネクタ64を介して、対応する演算ボード50から演算結果データを受信する。そして、各種データ入出力回路68は、受信した演算結果データを、通信路36を通じてハードウェア・モジュール30へクロック信号（CLK）に同期して転送する。

【0094】

ハードウェア・モジュール30においては、演算装置アレイ40の演算装置「00」～演算装置「23」から受信した各演算結果データをFPGA34からローカルプロセッサ33へ送る。ローカルプロセッサ33は、標準バス・インターフェース31からバス14

10

20

30

40

50

を介して、ホストプロセッサ 11 へ演算結果データを転送する。

【0095】

演算装置アレイ 40 を構成する各演算装置 40 - 1 から個々の問題領域に対する演算結果データを取得したホストプロセッサ 11 は、当該演算結果データと対応する演算装置の番号を基に演算対象全体の演算を実行する。そして、図 3 に示したような演算対象全体の物理的構造や論理的構造、特性などを解析し、表示装置（図示せず）に出力する。

【0096】

上記構成の演算ボード 50 と書き込み・入出力ボード 60 を備える演算装置 40 - 1 によれば、任意の演算装置 40 - 1 の演算ボード 50 による演算結果データを隣接する演算装置 40 - 1 の演算ボード 50 へ直接転送することができる。つまり、演算のためのデータ転送にはバスを介さない。

10

【0097】

また、演算装置アレイ 40 の各演算装置 40 - 1 による演算結果データ等を取り出してホストプロセッサ 11 宛てに送信するとき、および、ホストプロセッサ 11 から仮想回路データや演算対象データ等を受信するときのみバス（例えばバス 14）を利用する。

このように構成したことにより、隣接する演算装置 40 - 1 間のデータ転送処理が情報処理装置の処理能力のボトルネックになることを回避することができる。

【0098】

なお、演算装置アレイ 40 の各演算装置からホストプロセッサ 11 への演算結果データの転送は、各演算装置において問題領域の演算が終了後、演算結果データをメモリに記憶する処理と併せて実行するようにするとよい。このようにした場合、演算が終了した演算装置から順に演算結果データがホストプロセッサ 11 へ向けて転送されるので、データの輻輳が抑制され、バス 14 を始めとする各通信路におけるオーバーヘッドを小さくすることができる。

20

【0099】

本出願人の試作した情報処理装置によれば、128 セットの演算装置を用いて演算装置アレイを構成したところ、約 0.7 テラフロップス（TFLOPS）を達成している。

【0100】

[8 . 演算装置への書き込み処理]

次に、図 13 および図 14 を参照して、演算装置アレイを構成する各演算装置に対する演算回路等の書き込み処理を説明する。図 13 は、演算回路等書き込み処理を示すフローチャートである。図 14 は、演算回路等書き込み処理時の状態遷移を示すものである。

30

【0101】

図 10 に示したように、最初の書き込み回路はホストプロセッサ 11 上にあり、ホストプロセッサ 11 から段階的に、演算装置アレイ 40 の各演算装置 40 - 1 に演算回路等の仮想回路を書き込んでいく。しかし、仮想回路の中に書き込み回路が構成されていないと、次段（隣接）の演算装置 40 - 1 に仮想回路を書き込むことができない。他方、演算時には、書き込み回路は不用であるので、仮想回路の書き込みが終了したときには削除するのが望ましい。したがって、書き込みが可能なように、全ての演算装置 40 - 1 に対して、書き込み回路あるいは書き込み用の接続回路を書き込み、その後に末端の演算装置 40 - 1 から演算回路等に置き換えていく必要がある。この処理は、ホストプロセッサ 11 上で動作するソフトウェアが図 13 に示す流れに沿って行う。

40

【0102】

まず、ホストプロセッサ 11 がバス 14 および標準バス・インターフェース 31 を介して、メモリ 12 に記憶してある演算対象に基づく仮想回路書き込みデータをハードウェア・モジュール 30 へ送信する（ステップ S1）。この仮想回路書き込みデータには少なくとも、演算対象の各問題領域に対応する「演算装置の番号」と、それに対応する「仮想回路データ」と、「書き込み順番」の情報が含まれる。現段階では、演算装置 40 - 1 を構成する演算ボード 50 および書き込み・入出力ボード 60 には仮想回路が書き込まれていない初期状態である（図 14 A）。

50

【 0 1 0 3 】

ホストプロセッサ 1 1 は、ハードウェア・モジュール 3 0 内の F P G A からなるローカルプロセッサ 3 3 に、仮想回路書き込みデータを書き込む（ステップ S 2）。

【 0 1 0 4 】

そして仮想回路書き込みデータが書き込まれたローカルプロセッサ 3 3 は、ハードウェア・モジュール 3 0 内の演算装置アレイ制御用の F P G A 3 4 に、仮想回路書き込みデータを書き込む（ステップ S 3）。

【 0 1 0 5 】

仮想回路書き込みデータが書き込まれた F P G A 3 4 は、入出力インターフェース 3 5 および通信路 3 6 を通じて、演算装置アレイ 4 0 の所定の演算装置 4 0 - 1 に仮想回路書き込みデータを送信する（ステップ S 4）。

10

【 0 1 0 6 】

F P G A 3 4 は、演算装置 4 0 - 1 の書き込み・入出力ボード 6 0 の F P G A 6 1 に、仮想回路データに基づく書き込み回路 6 6 および仮想回路データ入出力回路 6 7 を再構成する。さらに、各種データ入出力回路 6 8 を再構成する。そして、書き込み回路 6 6 および仮想回路データ入出力回路 6 7 が書き込まれた書き込み・入出力ボード 6 0 では、入力された仮想回路データを当該仮想回路データに含まれる書き込み順番に従って、隣接する演算装置 4 0 - 1 の書き込み・入出力ボード 6 0 へ転送する（ステップ S 5）。このとき、書き込み・入出力ボード 6 0 に書き込み回路 6 6 および仮想回路データ入出力回路 6 7 が書き込まれた状態となる（図 1 4 B）。

20

【 0 1 0 7 】

図 1 0 を例に説明すると、一例として演算装置「 0 0 」→演算装置「 1 0 」, 「 0 1 」→演算装置「 2 0 」, 「 1 1 」, 「 2 2 」→・・・の順番に書き込み回路 6 6 等を書き込んでいく。なお、図 1 0 に示した例では、演算装置が 2 次元に配置してあるが、演算装置が 3 次元に配置されていれば書き込み処理を 3 次元で展開してもよい。

【 0 1 0 8 】

ホストプロセッサ 1 1 は、書き込み・入出力ボード 6 0 へ書き込み回路 6 6 等を書き込むと、その都度、動作確認のため書き込み・入出力ボード 6 0 から書き込み完了を示す終了信号（フラグ）を取得する。そして、ホストプロセッサ 1 1 は、全ての演算装置 4 0 - 1 の書き込み・入出力ボード 6 0 について、書き込み回路等の再構成が終了したか否かを判定する（ステップ S 6）。

30

【 0 1 0 9 】

全ての書き込み・入出力ボード 6 0 について書き込み回路 6 6 等の書き込みが終了していない場合、ステップ S 5 の処理に戻り、書き込み処理を継続する。

【 0 1 1 0 】

一方、全ての書き込み・入出力ボード 6 0 について書き込みが終了した場合、演算装置アレイ 4 0 の末端の書き込み・入出力ボード 6 0 の書き込み回路 6 6 が、対応する演算ボード 5 0 の F P G A 5 1 に並列演算回路 5 6 等の仮想回路（図 1 2 参照）を書き込んでいく（ステップ S 7）。

【 0 1 1 1 】

そして、全ての書き込み・入出力ボード 6 0 に対応する書き込み回路 6 6 が、入力された仮想回路データに含まれる書き込み順番情報に従って、対応する書き込み演算ボード 5 0 の F P G A 5 1 に並列演算回路 5 6 等の仮想回路を書き込んでいく（ステップ S 8）。

40

【 0 1 1 2 】

例えば、演算装置「 2 3 」→演算装置「 2 2 」, 「 1 3 」→演算装置「 2 1 」, 「 1 2 」, 「 0 3 」→・・・の順番に並列演算回路 5 6 へ置き換えていく。なお、図 1 0 に示した例では、演算装置が 2 次元に配置してあるが、演算装置が 3 次元に配置されていれば書き込み処理も 3 次元で展開してもよい。

【 0 1 1 3 】

ホストプロセッサ 1 1 は、演算ボード 5 0 へ並列演算回路 5 6 等が書き込まれると、そ

50

の都度、演算ボード 50 から書き込み完了を示す終了信号（フラグ）を取得する。ホストプロセッサ 11 は終了信号に基づき、全ての演算装置 40 - 1 の演算ボード 50 について、並列演算回路 56 等の書き込みが終了したか否かを判定する（ステップ S9）。

【0114】

全ての演算ボード 50 について書き込みが終了していない場合、ステップ S8 の処理に戻り、並列演算回路 56 等の書き込み処理を継続する。

【0115】

一方、全ての演算ボード 50 について書き込みが終了した場合、演算装置アレイ 40 の演算装置 40 - 1 に対する仮想回路の書き込みを終了する。このとき、書き込み・入出力ボード 60 には書き込み回路 66 と仮想回路データ入出力回路 67 等が、演算ボード 50 には並列演算回路 56 等が書き込まれた状態となる（図 14C）。

10

【0116】

1 つの演算装置を書き換えるのにおおよそ 40 ms であるので、演算装置アレイ全体でも数秒で書き換えることができる。

【0117】

なお、FPGA はゲートアレイであるからゲート数のリソースには限りがある。また、演算時には、書き込み回路は不用であるので、仮想回路の書き込みが終了したときには削除するのが望ましい。そこで、演算ボード 50 に並列演算回路 56 等を書き込むその一方で、仮想回路データ書き込み回路が最後に書き込まれた演算装置 40 - 1 から最初に書き込まれた演算装置 40 - 1 まで順に、書き込み・入出力ボード 60 の FPGA 61 に書き込んだ書き込み回路 66 と仮想回路データ入出力回路 67 を消去する。このとき、書き込み・入出力ボード 60 には各種データ入出力回路 68 が、また演算ボード 50 には並列演算回路 56 等が書き込まれた状態となる（図 14D）。

20

【0118】

また、仮想回路データ書き込み回路が最後に書き込まれた演算装置 40 - 1 から最初に書き込まれた演算装置 40 - 1 まで順に、書き込み・入出力ボード 60 の FPGA 61 に書き込んだ書き込み回路 66 と仮想回路データ入出力回路 67 を消去するとしたが、この例に限られない。すなわち、ホストプロセッサ 11 が全ての演算装置 40 - 1 について演算回路等の書き込み動作が正常かどうかを確認できればよく、適切な順番で、書き込み・入出力ボード 60 の書き込み回路 66 と仮想回路データ入出力回路 67 を消去していけばよい。

30

【0119】

さらに上述した例では、仮想回路データ入出力回路 67 と各種データ入出力回路 68 を別々に構成したが、データ転送回路などとして一体に構成してもよい。

【0120】

図 15 は、演算回路等の仮想回路を書き込んだ後の演算装置アレイの状態を示したものである。

この例では、第 1 の演算装置の演算ボード 50 - 1 と隣接する第 2 の演算装置の演算ボード 50 - 2 が接続し、第 1 の演算装置の書き込み・入出力ボード 60 - 1 と隣接する第 2 の演算装置の書き込み・入出力ボード 60 - 2 が接続している。第 1 の演算装置はハードウェア・モジュールを介してハードウェア・モジュール 30（ホストプロセッサ 11 側）と接続しているとする。

40

【0121】

第 1 の演算装置の演算ボード 50 - 1 の並列演算回路 56 で生成された演算結果データは、対応する書き込み・入出力ボード 60 - 1 の各種データ入出力回路 68 からハードウェア・モジュール 30 へ送信され、ハードウェア・モジュール 30 からバス 14 を介してホストプロセッサ 11 へ転送される。

【0122】

一方、第 2 の演算装置の演算ボード 50 - 2 の並列演算回路 56 で生成された演算結果データは、対応する書き込み・入出力ボード 60 - 2 の各種データ入出力回路 68 からま

50

ずはホストプロセッサ 11 側に近い第 2 の演算装置の書き込み・入出力ボード 60 - 2 へ送信される。それから、書き込み・入出力ボード 60 - 2 の各種データ入出力回路 68 によって、第 2 の演算装置の当該演算結果データが通信路 36 を介してハードウェア・モジュール 30 へ送信され、ハードウェア・モジュール 30 からバス 14 を介してホストプロセッサ 11 へ転送される。

【0123】

本例では、説明の便宜上、隣接する 2 個の演算装置間の演算結果データの転送処理について説明したが、それ以上の個数の演算装置間、あるいは 3 次元配置された複数の演算装置間においても技術思想の基本は同じである。

【0124】

上述のように構成された本実施の形態によれば、演算装置アレイを構成する全ての演算装置 (FPGA 等) は隣接演算装置に対してのみ演算結果等のデータを転送する処理を実行すればよく、データ転送処理が情報処理装置の処理能力のボトルネックになることを回避することができる。

【0125】

また、従来の演算装置は全てのデータ転送にバスを介して行っていたため消費電力が膨大なものになり、データ転送の高速化を実現する上で障害となっていたが、上記実施の形態では演算装置間のデータ転送にバスを用いずコネクタ直結によるデータ転送を行うようにしたので、バスを用いた場合と比較して消費電力を小さくできる。

【0126】

また、従来は通信路での高速性と低消費電力を実現するために光通信を利用することが行われていたが、本実施の形態では光通信を用いずに処理能力の向上および低消費電力を実現するので、大規模演算用の情報処理装置を安価に提供できる。したがって、個人であっても、予約待ちをすることなく、大規模演算用のコンピュータを利用できるようになる。

【0127】

[9 . 本発明の他の実施の形態に係る演算装置アレイ]

3 次元構造の演算装置アレイ 40 において、図 12 に示したように、一つの演算装置 40 - 1 の演算ボード 50 には、隣接 FPGA データ転送回路が 3 対ある。この隣接 FPGA データ転送回路の接続先を電気的に変えることで、3 次元領域の中に 2 次元領域を畳み込むことができる。すなわち、3 次元に構成した演算装置アレイの各演算装置を電気的に制御することにより、2 次元または 3 次元の演算装置アレイとして使用することが可能になる。

【0128】

例えば、図 16 に示す演算装置アレイ 70 は、XY 領域に配列された 4 つの演算装置アレイ 71, 72, 73, 74 が順に接続された構成である。末端の演算装置、例えば X 端の演算装置において Z 方向の演算装置に接続することで XY 領域は Z 方向に向きを変え、上層の演算装置アレイで逆方向に進むことが可能である。

【0129】

すなわち、演算装置アレイ 71 の X 端に位置する演算装置 71 - 1, 71 - 2, 71 - 3 と、対応する下層の演算装置アレイ 72 の X 端に位置する演算装置 72 - 10, 72 - 11, 72 - 12 をそれぞれケーブル 76 で接続している。一方、演算装置アレイ 72 と演算装置アレイ 73 との関係では、演算装置アレイ 71 と演算装置アレイ 72 の接続部分とは反対側の X 端を接続する。同様にして、演算装置アレイ 73 と演算装置アレイ 74 との関係では、演算装置アレイ 72 と演算装置アレイ 73 の接続部分とは反対側の X 端を接続する。

【0130】

結果として、最下層にある演算装置アレイ 74 の演算装置 74 - 1, 71 - 2, 71 - 3 から、演算装置アレイ 72 および演算装置アレイ 73 を経由して、最上層にある演算装置アレイ 71 の演算装置 71 - 10, 71 - 11, 71 - 12 までは 2 次元に接続され、

10

20

30

40

50

演算装置アレイ 70 を 2 次元領域で構成することができる。

【0131】

これらの接続態様は、ホストプロセッサ 11 が演算対象に合わせて決定する。つまり、演算装置アレイを構成する複数の演算装置のうちいずれの演算装置間を接続するかをホストプロセッサ 11 が決定し、その決定事項を「演算結果の転送先」として各演算装置に送るようにする。

【0132】

このようにして、コネクタを用いて物理的に 3 次元に配置した演算装置アレイであっても、より低次（2 次元、1 次元）の演算対象となる問題領域を畳み込んで演算が行えるよう各演算装置に所望の演算回路等を書き込むことで、隣接する演算装置と物理的な接続を変えることなく電氣的に接続を変更することができる。

10

【0133】

[10. 本発明のさらに他の実施の形態に係る演算装置アレイ]

次に、演算対象が周期境界条件を満たす場合の演算装置アレイについて説明する。

演算対象が周期境界条件を満たす場合、演算装置アレイを構成する複数のハードウェア・ネットすなわち演算装置のうち周期境界に位置する演算装置同士を電氣的に接続し、トラス状（円環）を形成する。

【0134】

例えば、図 17 に示す 2 次元の演算装置アレイ 80 において、周期境界に該当する最終段の演算装置 80 - 10, 80 - 11, 80 - 12 と、一方の周期境界に該当する初段の演算装置 80 - 1, 80 - 2, 80 - 3 をコネクタ 81 を介して電氣的に接続する。この時、各演算装置間で発生する遅延時間が同一になるように各演算装置のコネクタはフレキシブルな配線を用いて適用する事も勿論である。このようにして、演算装置 80 - 10, 80 - 11, 80 - 12 の各々の演算結果データを、最初に演算を行う演算装置 80 - 1, 80 - 2, 80 - 3 に戻すことにより、繰り返し処理が行える。なお、図 17 に示す例では、2 次元的周期境界条件を満たす場合を説明したが、3 次元の周期境界条件を満たす場合にも適用できることは勿論である。

20

【0135】

以上に述べた実施の形態は、本発明を実施するための好適な形態の具体例であるから、技術的に好ましい種々の限定が付されている。ただし、本発明は、以上の実施の形態の説明において特に本発明を限定する旨の記載がない限り、これらの実施の形態に限られるものではない。したがって、例えば、以上の説明で挙げた使用材料とその使用量、処理時間、処理順序および各パラメータの数値的条件等は好適例に過ぎず、また、説明に用いた各図における寸法、形状および配置関係等も実施の形態の一例を示す概略的なものである。したがって、本発明は、上述した実施の形態の例に限定されるものではなく、本発明の要旨を逸脱しない範囲において、種々の変形、変更が可能である。

30

【0136】

例えば、図 12 に示す演算装置 41 - 1 において、演算ボード 50 と書き込み・入出力ボード 60 に、それぞれメモリ 52 とメモリ 62 を備えるようにしたがこれらは一つで代用してもよい。例えば、メモリ 62 は搭載せずメモリ 52 だけを使用するようにしてもよい。また、メモリ 52 とメモリ 62 に代えて大規模なメモリボードを用意したり、ハードディスク等の大容量記録装置を設けるようにしたりしてもよい。

40

【図面の簡単な説明】

【0137】

【図 1】従来のシステムの概要を示す構成図である。

【図 2】スイッチボックスの説明に供する概略図である。

【図 3】演算対象の一例を示す模式図である。

【図 4】本発明の一実施の形態に係る演算装置アレイを示す概略図である。

【図 5】本発明の一実施の形態に係る演算装置（演算ボード）のコネクタ配置を示す図である。

50

【図 6】本発明の一実施の形態に係る演算装置（演算ボード）の相互接続の説明に供する図である。

【図 7】本発明の一実施の形態に係る演算装置の分解斜視図である。

【図 8】本発明の一実施の形態に係る情報処理装置の概要を示す図である。

【図 9】本発明の一実施の形態に係るホストプロセッサとハードウェア・ネット間の制御概要を示す図である。

【図 10】本発明の一実施の形態に係る情報処理装置の全体構成を示す概略図である。

【図 11】本発明の一実施の形態に係るホストプロセッサの階層を示す構成図である。

【図 12】本発明の一実施の形態に係る演算装置（演算ボード、書き込み・入出力ボード）の機能を示すブロック図である。

10

【図 13】本発明の一実施の形態に係る演算回路等書き込み処理を示すフローチャートである。

【図 14】A ~ D は、本発明の一実施の形態に係る演算回路等書き込み時の状態遷移を示す図である。

【図 15】本発明の一実施の形態に係る演算回路等書き込み後の状態を示す図である。

【図 16】本発明の他の実施の形態に係る 3 次元領域に 2 次元領域を畳み込む場合の演算装置アレイを示す図である。

【図 17】本発明のさらに他の実施の形態に係る周期境界条件を満たす場合の演算装置アレイを示す図である。

【符号の説明】

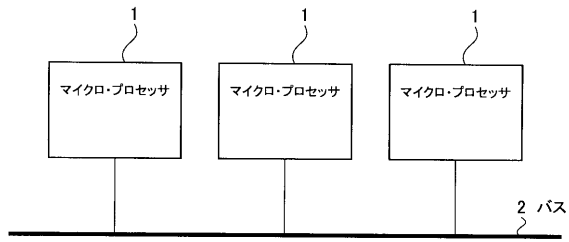
20

【0138】

6 - 1 A ~ 6 - 1 C ... 格子点、1 1 ... ホストプロセッサ、1 2 ... メモリ、1 3 ... P L D、1 4 ... バス、2 1 ... アプリケーション、2 2 ... オブジェクト・マネジャー、2 3 ... ハードウェア・モジュール・ドライバ、2 5 ... バス、2 6 ... ハードウェア・オブジェクト、2 7 ... ハードウェア・ドライバ、2 8 ... インターフェース、2 9 ... ハードウェア・ネット、3 0 ... ハードウェア・モジュール、3 1 ... 標準バス・インターフェース、3 2 ... ローカルメモリ、3 3 ... ローカルプロセッサ、3 4 ... F P G A、3 5 ... 入出力インターフェース、3 6 ... 通信路、4 0 ... 演算装置アレイ、4 0 - 1 ... 演算装置、5 0 ... 演算ボード、5 1 ... F P G A、5 2 ... メモリ、5 3 F , 5 3 B , 5 3 L , 5 3 R , 5 3 U , 5 3 D , 5 4 ... コネクタ、5 5 ... メモリ制御回路、5 6 ... 並列演算回路、5 6 A , 5 6 B ... 論理ブロック、5 7 ... 演算対象データ入出力回路、5 8 ... 演算結果データ入出力回路、5 9 X , 5 9 Y , 5 9 Z ... 隣接 F P G A データ転送回路、6 0 ... 書き込み・入出力ボード、6 0 A 1 ... 切り欠き、6 1 ... F P G A、6 2 ... メモリ、6 3 F , 6 3 B , 6 4 ... コネクタ、6 5 ... メモリ制御回路、6 6 ... 書き込み回路、6 7 ... 仮想回路データ入出力回路、6 8 ... 各種データ入出力回路

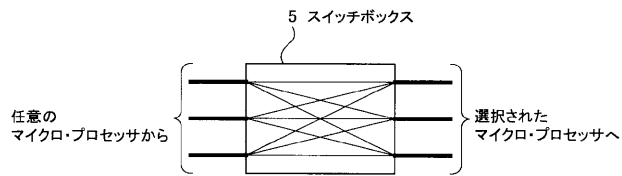
30

【図 1】



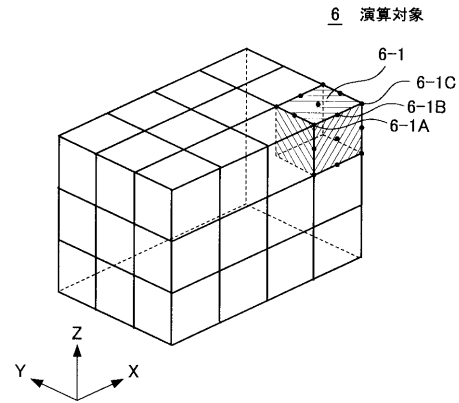
従来のシステムの概要

【図 2】



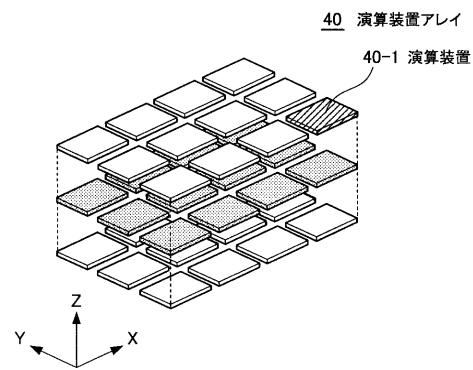
スイッチボックス

【図 3】



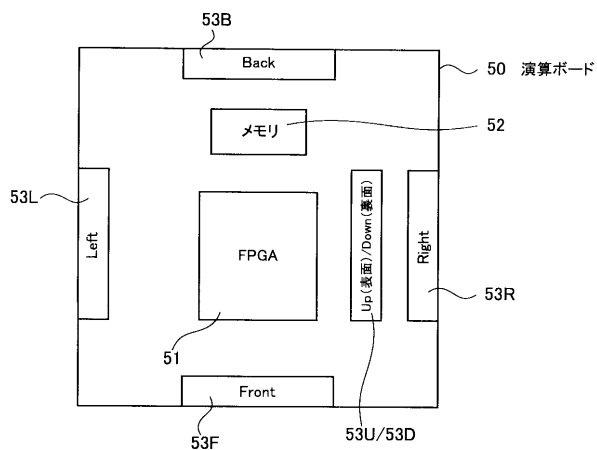
演算対象の一例

【図 4】



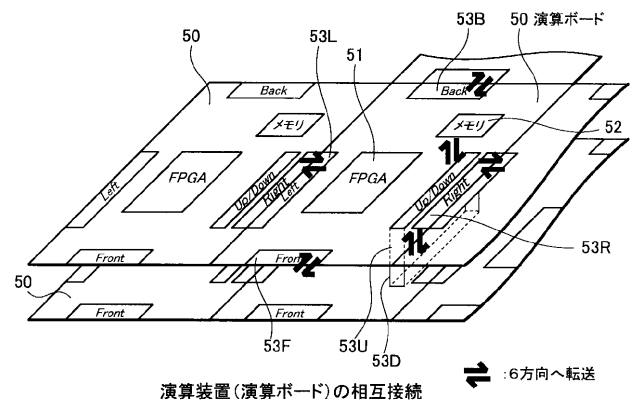
演算装置アレイの例

【図 5】



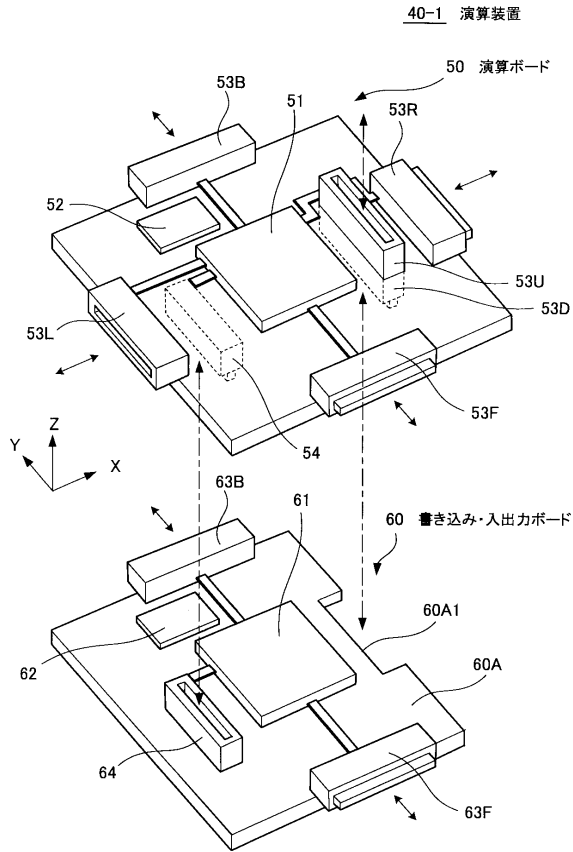
演算装置(演算ボード)のコネクタ配置

【図 6】



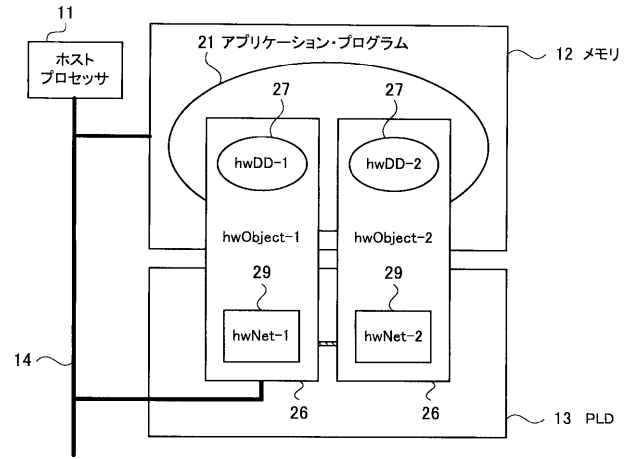
演算装置(演算ボード)の相互接続

【 圖 7 】



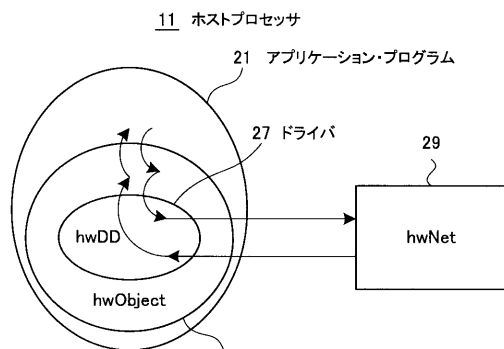
演算装置の分解斜視図

【 図 8 】



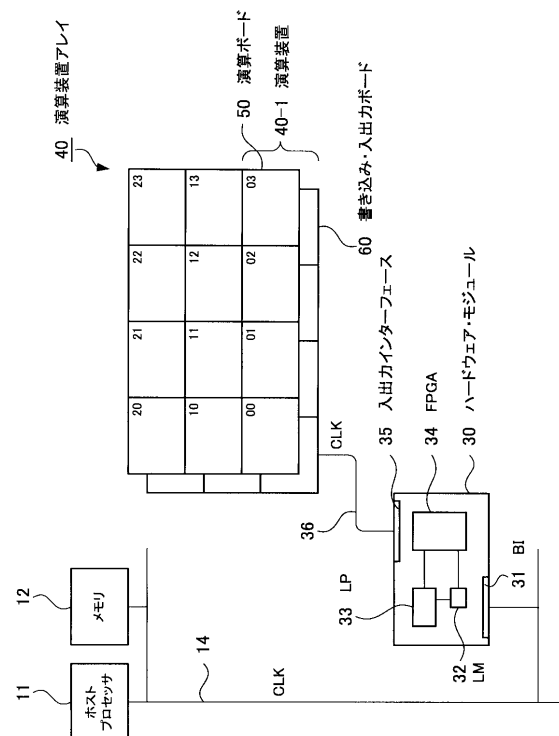
情報処理装置の概要

【 図 9 】



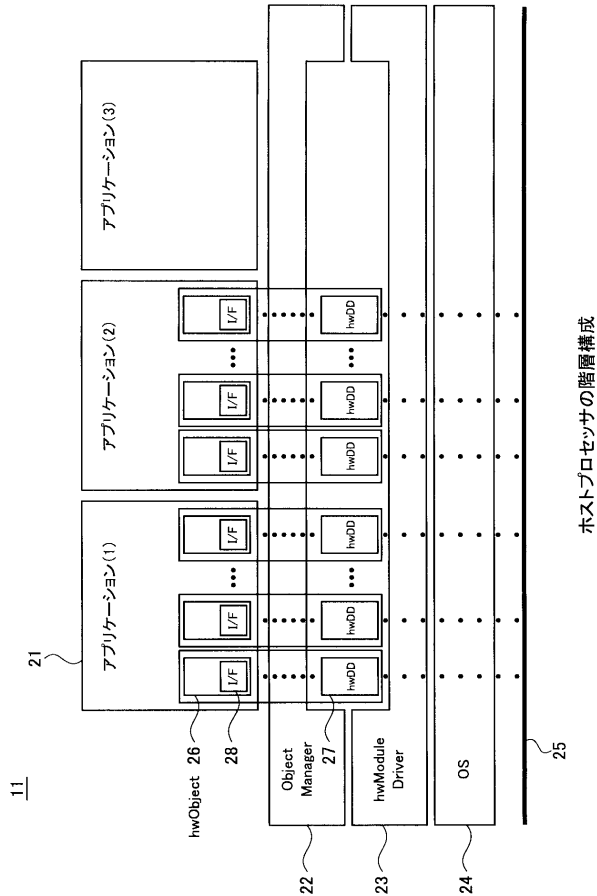
ホストプロセッサとハードウェア・ネット間の制御概要

【 図 1 0 】

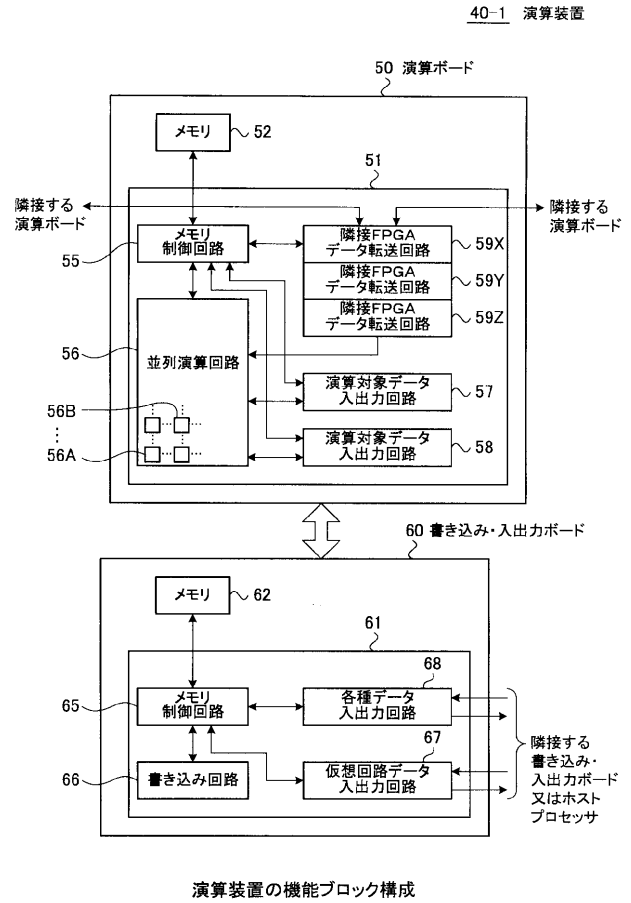


情報処理装置の全体構成

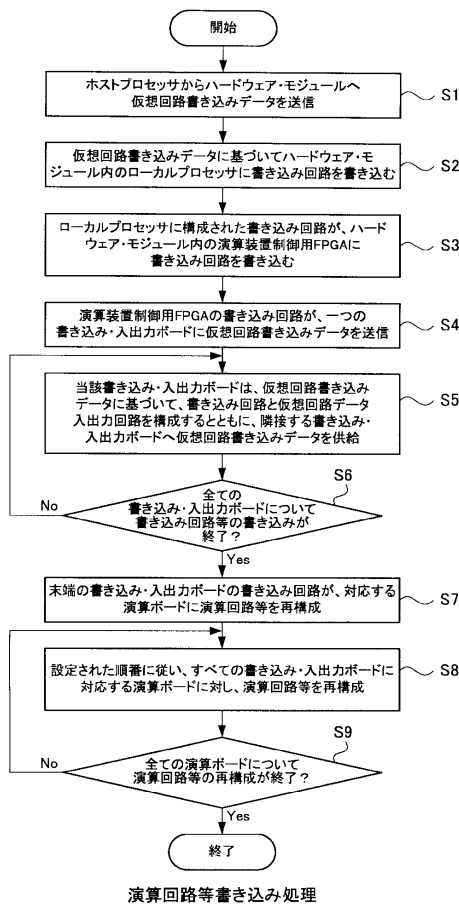
【図 1 1】



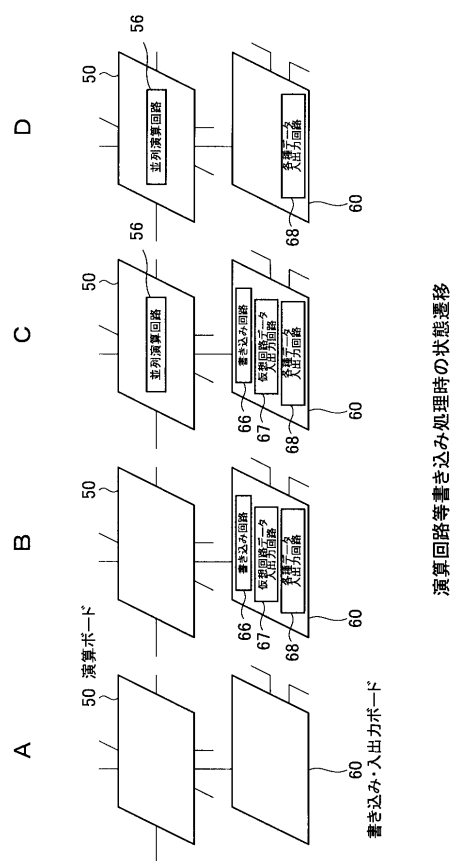
【図 1 2】



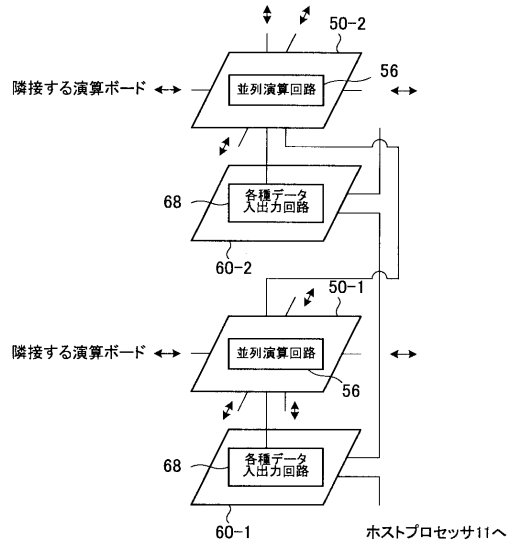
【図 1 3】



【図 1 4】

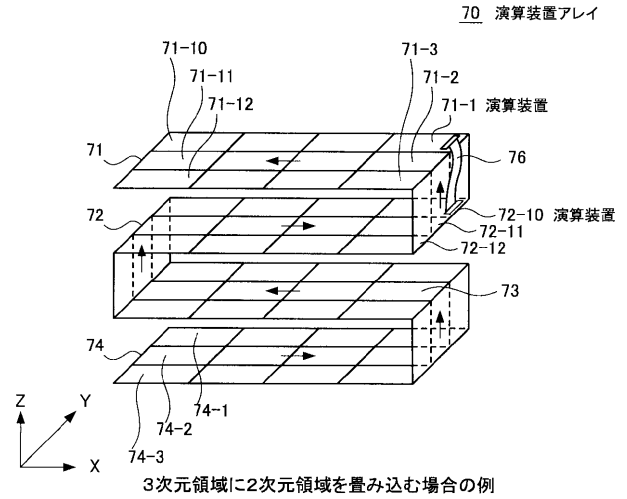


【図 15】

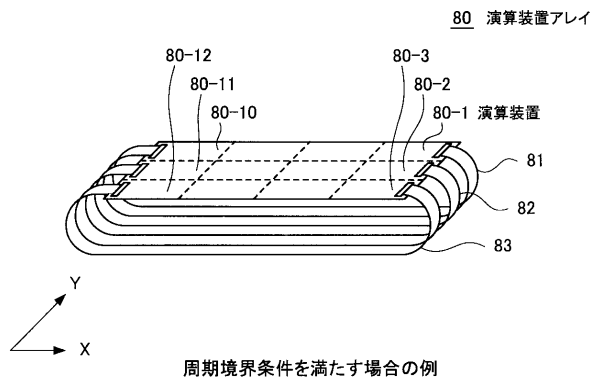


演算回路等書き込み後の状態

【図 16】



【図 17】



フロントページの続き

(72)発明者 佐藤 一輝

東京都府中市晴見町 3 - 8 - 1 国立大学法人東京農工大学内

Fターム(参考) 5B045 BB01 BB12 BB28 BB47 GG12 GG13 KK06