(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第6673202号
(P6673202)
(45)発行日　令和2年3月25日(2020.3.25)　(24)登録日　令和2年3月9日(2020.3.9)

(51)Int.Cl.　　F I
G06F 9/38 (2006.01)　G06F 9/38 310J
G06F 9/48 (2006.01)　G06F 9/48 370
G06T 1/20 (2006.01)　G06T 1/20 C

請求項の数 8　(全 41 頁)

(21)出願番号　特願2016-529022(P2016-529022)
(86)(22)出願日　平成27年6月11日(2015.6.11)
(86)国際出願番号　PCT/JP2015/002924
(87)国際公開番号　WO2015/194133
(87)国際公開日　平成27年12月23日(2015.12.23)
審査請求日　平成30年5月15日(2018.5.15)
(31)優先権主張番号　特願2014-125942(P2014-125942)
(32)優先日　平成26年6月19日(2014.6.19)
(33)優先権主張国・地域又は機関
日本国(JP)

(73)特許権者　000004237
日本電気株式会社
東京都港区芝五丁目７番１号
(74)代理人　100109313
弁理士　机　昌彦
(74)代理人　100124154
弁理士　下坂　直樹
(72)発明者　小堀　友義
東京都港区芝五丁目７番１号
日本電気株式会社内

審査官　清木　泰

最終頁に続く

(54)【発明の名称】演算装置、演算装置の制御方法、及び、演算装置の制御プログラム

(57)【特許請求の範囲】
【請求項１】
データに対する演算処理を実行する複数の演算手段と、
複数の前記演算手段の内、第１の演算手段と、第２の演算手段との間に接続され、前記第１の演算手段から出力されるデータと、前記第２の演算手段に入力されるデータとを保持可能な、少なくとも１以上のデータバッファと、
１以上の前記第１の演算手段と、１以上の前記第２の演算手段との少なくともいずれかである１以上の特定の前記演算手段において実行される、前記データに対する一連の演算処理であるタスクの実行を指定するタスクコマンドを、前記演算手段に対して出力可能に保持するコマンドバッファと、
前記タスクコマンドと、前記タスクが実行される少なくともいずれかの前記演算手段を特定可能なタスク設定情報と、複数の前記演算手段のそれぞれから取得した当該演算手段における演算処理の状態を表す情報と、に基づいて、前記データバッファ及び前記コマンドバッファの少なくともいずれかを制御することにより、１以上の特定の前記演算手段において実行される前記タスクに関する演算処理を制御するタスク制御手段と、を備える
演算装置。
【請求項２】
前記タスク制御手段は、前記１以上の特定の前記演算手段に対する、前記コマンドバッファからの前記タスクコマンドの出力を制御することにより、前記１以上の特定の前記演算手段における前記タスクに関する演算処理の開始タイミングを制御する

請求項１に記載の演算装置。
【請求項３】
前記第１の演算手段は、当該第１の演算手段において実行される前記タスクに関する演算処理が終了した際、当該第１の演算手段の出力側に接続された前記データバッファに対して、その前記タスクに関する前記タスクコマンドを出力し、
当該第１の演算手段の出力側に接続された前記データバッファは、上記出力された前記タスクコマンドを保持し、
前記タスク制御手段は、当該データバッファが入力側に接続された前記第２の演算手段に対する、当該データバッファからの前記タスクコマンドの出力を制御することにより、当該第２の演算手段における前記タスクに関する演算処理の開始タイミングを制御する
請求項１または請求項２に記載の演算装置。
【請求項４】
前記タスク設定情報は、前記タスク毎に、当該タスクに関する最初の演算処理を実行する前記演算手段である開始演算手段と、当該タスクに関する最後の演算処理を実行する前記演算手段である終了演算手段とを特定可能な情報を保持し、
前記タスク制御手段は、当該タスク制御手段により制御される前記１以上の特定の前記演算手段である制御対象演算手段が前記開始演算手段であるか否かを判定した結果に基づいて、複数の前記演算手段の内、当該タスク制御手段により制御される前記１以上の特定の前記演算手段である制御対象演算手段において、特定の前記タスクを実行可能か否か判定し、
前記タスクコマンドには、当該タスクコマンドにより指定される前記タスクと実行順序において依存関係がある前記タスクである依存タスクを表すタスク依存関係情報が含まれ、
前記タスク制御手段は、
前記タスク設定情報に基づいて、前記制御対象演算手段が、前記タスクコマンドにより指定された特定のタスクに対する前記開始演算手段であると判定した場合には、当該特定の前記タスクに関する前記タスクコマンドに含まれる前記タスク依存関係情報に基づいて、当該特定の前記タスクよりも前に実行されるよう設定された前記依存タスクが存在するか否かを確認し、
当該確認の結果、当該依存タスクが存在しない場合には、当該特定の前記タスクに関する前記タスクコマンドよりも前に当該タスク制御手段に出力された他のタスクコマンドにより指定された他の前記タスクに関する演算処理が、当該他の前記タスクに対する前記終了演算手段において完了する前に、当該特定の前記タスクに関する演算処理を実行するよう前記制御対象演算手段を制御し、
前記タスク制御手段は、
前記タスク設定情報に基づいて、当該タスク制御手段が制御する前記制御対象演算手段が特定の前記タスクに対する前記開始演算手段であると判定した場合には、当該特定の前記タスクの実行を指定する前記タスクコマンドに含まれる前記タスク依存関係情報に基づいて、当該特定の前記タスクよりも前に実行されるよう依存関係が設定された依存タスクが存在するか否かを確認し、
当該確認の結果、当該依存タスクが存在する場合には、前記タスク設定情報に基づいて、当該依存タスクに対する前記終了演算手段を特定し、当該特定した前記終了演算手段の演算処理の状態を表す情報に基づいて、当該特定した前記終了演算手段における当該依存タスクに関する演算処理が終了しているか確認し、
当該確認結果に基づいて、前記制御対象演算手段における特定のタスクに関する演算処理を制御することを特徴とする、
請求項１乃至請求項３のいずれかに記載の演算装置。
【請求項５】
前記タスク設定情報は、前記タスク毎に、当該タスクに関する最初の演算処理を実行する前記演算手段である開始演算手段を特定可能な情報と、当該タスクに関する一連の演算

処理に用いられる前記演算手段の数を表す情報である演算手段使用数と、を保持し、
前記タスク制御手段は、
前記開始演算手段を特定可能な情報と、前記演算手段使用数とに基づいて、前記タスク毎に当該タスクに関する最後の演算処理を実行する終了演算手段を特定し、
当該タスク制御手段により制御される前記１以上の特定の前記演算手段である制御対象演算手段が前記開始演算手段であるか否かを判定した結果に基づいて、複数の前記演算手段の内、当該タスク制御手段により制御される前記１以上の特定の前記演算手段である制御対象演算手段において、特定の前記タスクを実行可能か否か判定し、
前記タスクコマンドには、当該タスクコマンドにより指定される前記タスクと実行順序において依存関係がある前記タスクである依存タスクを表すタスク依存関係情報が含まれ、
前記タスク制御手段は、
前記タスク設定情報に基づいて、前記制御対象演算手段が、前記タスクコマンドにより指定された特定のタスクに対する前記開始演算手段であると判定した場合には、当該特定の前記タスクに関する前記タスクコマンドに含まれる前記タスク依存関係情報に基づいて、当該特定の前記タスクよりも前に実行されるよう設定された前記依存タスクが存在するか否かを確認し、
当該確認の結果、当該依存タスクが存在しない場合には、当該特定の前記タスクに関する前記タスクコマンドよりも前に当該タスク制御手段に出力された他のタスクコマンドにより指定された他の前記タスクに関する演算処理が、当該他の前記タスクに対する前記終了演算手段において完了する前に、当該特定の前記タスクに関する演算処理を実行するよう前記制御対象演算手段を制御し、
前記タスク制御手段は、
前記タスク設定情報に基づいて、当該タスク制御手段が制御する前記制御対象演算手段が特定の前記タスクに対する前記開始演算手段であると判定した場合には、当該特定の前記タスクの実行を指定する前記タスクコマンドに含まれる前記タスク依存関係情報に基づいて、当該特定の前記タスクよりも前に実行されるよう依存関係が設定された依存タスクが存在するか否かを確認し、
当該確認の結果、当該依存タスクが存在する場合には、前記タスク設定情報に基づいて、当該依存タスクに対する前記終了演算手段を特定し、当該特定した前記終了演算手段の演算処理の状態を表す情報に基づいて、当該特定した前記終了演算手段における当該依存タスクに関する演算処理が終了しているか確認し、
当該確認結果に基づいて、前記制御対象演算手段における特定のタスクに関する演算処理を制御することを特徴とする、
請求項１乃至請求項３のいずれかに記載の演算装置。
【請求項６】
前記データバッファは、複数の前記演算手段の内、前記第１の演算手段の出力側と、前記第２の演算手段の入力側との間に接続され、
前記データバッファには、前記第１の演算手段の出力側から、当該第１の演算手段における演算処理の結果と、当該データバッファの読み出しを抑制可能な同期コマンドと、が入力され、
当該データバッファは、当該同期コマンドが入力された場合、当該同期コマンドが入力されるより前に当該データバッファに入力されたデータが全て読み出されるまで、当該同期コマンドが入力された後に当該データバッファに入力されたデータの読み出しを禁止する、
請求項１乃至請求項５のいずれかに記載の演算装置。
【請求項７】
データに対する演算処理を実行する演算手段である１以上の第１の演算手段と、１以上の第２の演算手段と、の少なくともいずれかである１以上の特定の前記演算手段において実行される前記データに対する一連の演算処理である、タスクの実行を指定するタスクコ

マンドと、前記タスクが実行される前記１以上の特定の前記演算手段を特定可能な情報と、複数の前記演算手段のそれぞれから取得した当該前記演算手段の処理状態を表す情報と、に基づいて、

前記第１の演算手段と、前記第２の演算手段との間に接続され、前記第１の演算手段から出力されるデータと、前記第２の演算手段に入力されるデータと、を保持可能な少なくとも１以上のデータバッファ、及び、前記タスクコマンドを前記１以上の特定の前記演算手段に対して出力可能に保持するコマンドバッファ、の少なくともいずれかを制御することにより、前記１以上の特定の前記演算手段において実行される前記タスクに関する演算処理を制御する

演算装置の制御方法。

【請求項８】

データに対する演算処理を実行する演算手段である１以上の第１の演算手段と、１以上の第２の演算手段と、の少なくともいずれかである１以上の特定の前記演算手段において実行される前記データに対する一連の演算処理である、タスクの実行を指定するタスクコマンドと、前記タスクが実行される前記１以上の特定の前記演算手段を特定可能な情報と、複数の前記演算手段のそれぞれから取得した当該前記演算手段の処理状態を表す情報と、に基づいて、

前記第１の演算手段と、前記第２の演算手段との間に接続され、前記第１の演算手段から出力されるデータと、前記第２の演算手段に入力されるデータと、を保持可能な少なくとも１以上のデータバッファ、及び、前記タスクコマンドを前記１以上の特定の前記演算手段に対して出力可能に保持するコマンドバッファ、の少なくともいずれかを制御することにより、前記１以上の特定の前記演算手段において実行される前記タスクに関する演算処理を制御する処理を、演算装置に実行させる

演算装置の制御プログラム。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、１以上の演算部における一連の演算処理であるタスクの実行を、演算部毎に制御可能な、演算装置等に関する。

【背景技術】

【０００２】

信号処理等のデータ処理においては、多量のデータ（例えば、特定の信号に関するストリームデータ等）が演算処理の対象として扱われる。そして、それら多量のデータに対して、複数の演算処理（例えば、ＦＦＴ（高速フーリエ変換：Ｆａｓｔ　Ｆｏｕｒｉｅｒ　Ｔｒａｎｓｆｏｒｍ）や、各種フィルタリング等）が、順次実行される場合がある。

【０００３】

このようなデータ処理においては、上記した演算処理は、それぞれ専用ハードウェアや、ＤＳＰ（Ｄｉｇｉｔａｌ　Ｓｉｇｎａｌ　Ｐｒｏｃｅｓｓｏｒ）等のプロセッサ（演算装置）等により実現されてもよい。同様に、上記した演算処理は、コンピュータ等の汎用のハードウェアとソフトウェア（コンピュータ・プログラム）の組合せとして実現されてもよい。以下、上記したような演算処理を実現可能な各種構成を、まとめて「演算ブロック」または、「演算部」と称する場合がある。

【０００４】

上記のようなデータ処理を実行する際、上記演算ブロックを制御可能な上位の制御モジュール（制御装置）が、１以上の上記演算ブロックにおける一連の演算処理をタスクとしてまとめて、各演算ブロックにおける当該タスクの実行を制御する構成が知られている。このようなタスクは、１以上の演算ブロックによる、データに対する演算処理を含む。

【０００５】

例えば、ある演算ブロックの処理結果（出力）が他の演算ブロックの入力となるように、複数の演算ブロックを接続して一連の演算処理である演算フローを構築する場合、図２

０に例示するような構成が用いられる場合がある。この場合、複数の演算ブロック（演算ブロック２００１乃至演算ブロック２００３）がデータバッファ（データバッファ２００４乃至２００５）を介して接続される。そして、それら複数の演算ブロックによる演算処理が、まとめて一つのタスクとして制御される。即ち、複数の演算ブロック及びデータバッファがパイプライン状に接続され、係るパイプラインにより一つのタスクが実行される。上記データバッファは、当該データバッファに接続された演算ブロックから各種データを入出力可能な記憶領域である。

【０００６】

図２０に例示するような構成によって、タスク制御の粒度を粗く（大きく）することが可能であることから、タスクの実行を制御するための処理負荷を削減可能である。また、データバッファを用いることにより、各演算ブロックにおけるメモリアクセスのレイテンシ（遅延）を削減することが可能であることから、演算処理効率の向上が見込まれる。

【０００７】

複数の演算ブロックを用いてデータ処理を実行する技術に関連して、以下の特許文献が開示されている。

【０００８】

特許文献１は、依存関係のある複数のタスクを実行する情報処理装置において、タスク間のデータの受渡しに使用されるバッファに対するアクセスのオーバーヘッドを低減する技術を開示する。特許文献１に開示された技術は、プロセッサ（演算ブロック）内部に設けられた内部メモリを介してデータを受け渡すことにより、バッファアクセスの遅延を低減する。また、特許文献１に開示された技術は、依存関係のある複数のタスクが連続して処理されるように、タスクの実行順序を制御する。

【０００９】

また、特許文献２は、マルチプロセッサにより構成された印刷装置において、複数頁にわたる印刷コマンドを解釈し、各頁の描画及び印刷処理を複数の異なるプロセッサに順次割り当てる技術を開示する。

【００１０】

また、特許文献３は、タスク間の接続状態に応じて、タスクの実行順序とタスク毎の実行時間の割当てを制御することにより、リアルタイム処理を維持しながら演算ブロックの利用効率を高める技術を開示する。

【先行技術文献】

【特許文献】

【００１１】

【特許文献１】国際公開第２０１２／０３９２１６号

【特許文献２】特開２００３－０６７１５６号公報

【特許文献３】特開２００１－０２２５９５号公報

【発明の概要】

【発明が解決しようとする課題】

【００１２】

図２０に例示するような単純な構成を採用した場合、各演算ブロックは、例えば図２１に模式的に例示するように、あるタスクに対する全ての演算処理が終了してから、次のタスクを実行することとなる。より具体的には、図２１に例示されるように、タスク１に関する全ての演算処理が完了するまで、各演算ブロックはタスク２に関する処理を実行しない。

【００１３】

この場合、あるタスクが実行されてから、次のタスクが実行されるまでの間、演算処理が実行されない（アイドル状態となる）演算ブロックが存在する場合がある。より具体的には、あるタスクの終了部分、もしくは、開始部分において、ある演算ブロックがアイドル状態となってしまう場合がある。このような演算ブロックのアイドル状態は、全体の演算処理を高速化する際に、ボトルネックとなる。換言すると、パイプライン状に接続した

演算ブロックにおいて、複数のタスクを実行する際にパイプラインのオーバーヘッドが生じることにより、各演算ブロックの稼働率が低下する。このような演算ブロックの稼働率の低下は、演算処理全体のボトルネックとなる。

【００１４】

このような問題の対策として、例えば、タスクの処理単位を細かく（粒度を小さく）することが考えられる。即ち、タスクの処理単位を小さくすることにより、単位タスク当りの実行時間を短くし、演算ブロックのアイドル状態を短くする方法が考えられる。あるいは、例えば、タスクの処理単位を大きく（粒度を大きく）して、タスクの処理に対するパイプラインのオーバーヘッドの割合を小さくする方法が考えられる。

【００１５】

しかしながら、タスクの処理単位を細かくした場合、タスクの実行を制御するための処理負荷が増大する。タスクの実行を制御する制御系の演算能力が低い場合、係る処理負荷の増大に伴い、タスクのスケジューリングが間に合わなくなる可能性がある。

【００１６】

また、タスクの処理単位を大きくした場合、当該大きくなった処理単位に応じてデータバッファのサイズを大きくする必要があることから、回路規模が大きくなる等の問題が生じる。

【００１７】

上記特許文献１は、依存関係のあるタスクを同一のプロセッサ（演算ブロック）においてまとめて処理する技術を開示するにすぎない。特許文献２は、頁毎に独立した描画処理を複数のプロセッサが独立して実行し、描画結果を並行して印刷処理する技術にすぎない。特に印刷処理は、一定の処理（給紙、印写、排紙）を繰り返すのみであるから、処理の効率向上は比較的容易である。特許文献３は、タスク間の接続関係と、タスク毎のリアルタイム性の要求度に応じて、特定のプロセッサにおけるタスクの実行順序とタスク毎の実行時間の割当てを変更する技術を開示するのみである。これより、上記いずれの特許文献においても、種々の異なるタスクを実行する場合に、パイプライン状に接続した複数の演算ブロックの稼働率を向上させる技術に関しては十分考慮されていない。

【００１８】

本願発明は、上記のような事情を鑑みてなされたものである。即ち、本願発明は、複数の演算ブロック（演算部）がパイプライン状に接続された構成を有する演算装置（あるいは演算システム）において、各演算ブロックが効率的に稼働するようにタスクの実行を制御可能な演算装置等を提供することを主たる目的とする。

【課題を解決するための手段】

【００１９】

上記の目的を達成すべく、本発明の一態様に係る演算装置は、以下の構成を備える。即ち、本発明の一態様に係る演算装置は、データに対する演算処理を実行する複数の演算手段と、複数の上記演算手段の内、第１の演算手段と、第２の演算手段との間に接続され、上記第１の演算手段から出力されるデータと、上記第２の演算手段に入力されるデータとを保持可能な、少なくとも１以上のデータバッファと、１以上の上記第１の演算手段と、１以上の上記第２の演算手段との少なくともいずれかである１以上の特定の上記演算手段において実行される、上記データに対する一連の演算処理であるタスクの実行を指定するタスクコマンドを、上記演算手段に対して出力可能に保持するコマンドバッファと、上記タスクコマンドと、上記タスクが実行される少なくともいずれかの上記演算手段を特定可能なタスク設定情報と、複数の上記演算手段のそれぞれから取得した当該演算手段における演算処理の状態を表す情報と、に基づいて、上記データバッファ及び上記コマンドバッファの少なくともいずれかを制御することにより、１以上の特定の上記演算手段において実行される上記タスクに関する演算処理を制御するタスク制御手段と、を備える。

【００２０】

また、本発明の一態様に係る演算装置の制御方法は、データに対する演算処理を実行する演算手段である１以上の第１の演算手段と、１以上の第２の演算手段と、の少なくとも

いずれかである１以上の特定の上記演算手段において実行される上記データに対する一連の演算処理である、タスクの実行を指定するタスクコマンドと、上記タスクが実行される上記１以上の特定の上記演算手段を特定可能な情報と、複数の上記演算手段のそれぞれから取得した当該上記演算手段の処理状態を表す情報と、に基づいて、上記第１の演算手段と、上記第２の演算手段との間に接続され、上記第１の演算手段から出力されるデータと、上記第２の演算手段に入力されるデータと、を保持可能な少なくとも１以上のデータバッファ、及び、上記タスクコマンドを上記１以上の特定の上記演算手段に対して出力可能に保持するコマンドバッファ、の少なくともいずれかを制御することにより、上記１以上の特定の上記演算手段において実行される上記タスクに関する演算処理を制御する。

【００２１】

また、同目的は、上記構成を有する演算装置の制御方法を、コンピュータによって実現するコンピュータ・プログラム、及び、そのコンピュータ・プログラムが格納（記録）されている、コンピュータ読み取り可能な記憶媒体によっても達成される。

【発明の効果】

【００２２】

本発明によれば、複数の演算ブロックがパイプライン状に接続された構成を有する演算装置において、各演算ブロックが効率的に稼働するようにタスクの実行を制御可能である。

【図面の簡単な説明】

【００２３】

【図１】図１は、本願発明の第１の実施形態に係る演算装置の機能的な構成を例示する図である。

【図２】図２は、本願発明の第１の実施形態に係るタスク設定テーブルの内容を例示する図である。

【図３】図３は、本願発明の第１の実施形態に係るタスクコマンドの内容を例示する図である。

【図４】図４は、本願発明の第１の実施形態に係るタスクコマンドの具体例を示す図である。

【図５】図５は、本願発明の第１の実施形態に係るタスクコマンドの別の具体例を示す図である。

【図６】図６は、本願発明の第１の実施形態に係るタスク設定テーブルの具体例を示す図である。

【図７】図７は、本願発明の第１の実施形態に係るタスク制御部の動作を例示するフローチャート（１／３）である。

【図８】図８は、本願発明の第１の実施形態に係るタスク制御部の動作を例示するフローチャート（２／３）である。

【図９】図９は、本願発明の第１の実施形態に係るタスク制御部の動作を例示するフローチャート（３／３）である。

【図１０】図１０は、本願発明の第２の実施形態に係るタスク設定テーブルの内容を例示する図である。

【図１１】図１１は、本願発明の第３の実施形態に係る演算装置の機能的な構成を例示する図である。

【図１２】図１２は、本願発明の第３の実施形態に係るデータバッファ１１０４に格納されるデータの具体例を示す図である。

【図１３】図１３は、本願発明の第４の実施形態に係る演算装置の機能的な構成を例示する図である。

【図１４】図１４は、本願発明の第４の実施形態に係るタスク制御部の動作を例示するフローチャートである。

【図１５】図１５は、本願発明の第４の実施形態に係る演算部の動作を例示するフローチャートである。

【図１６】図１６は、本願発明の第５の実施形態に係る演算装置の機能的な構成を例示する図である。
【図１７】図１７は、本願発明の第５の実施形態に係る演算装置の別の構成を例示する図である。
【図１８】図１８は、本願発明の第１の実施形態に係る演算装置における、タスクの実行過程を模式的に表す図である。
【図１９】図１９は、本願発明の各実施形態に係る制御部を実現可能なハードウェア構成を例示する図である。
【図２０】図２０は、本願発明に関連する技術である演算装置の機能的な構成を例示する図である。
【図２１】図２１は、本願発明に関連する技術である演算装置における、タスクの実行過程を模式的に表す図である。
【発明を実施するための形態】
【００２４】
次に、本発明を実施する形態について図面を参照して詳細に説明する。以下の実施の形態に記載されている構成は単なる例示であり、本願発明の技術範囲はそれらには限定されない。
【００２５】
＜第１の実施形態＞
以下、本発明の第１の実施形態について説明する。
【００２６】
まず、図１を参照して、本実施形態における演算装置について説明する。図１は本実施形態における演算装置の機能的な構成を例示する、ブロック図である。
【００２７】
図１に例示するように、本実施形態における演算装置１００は、演算制御部１０１と、演算対象であるデータを保持する第１のメモリ部１０２と、演算結果を保持する第２のメモリ部１０３とを備える。また、本実施形態における演算装置１００は、入力データと、上記演算制御部１０１から入力される各種制御信号（例えば、後述するタスクコマンド等）とに基づいて各種演算処理を実行する演算部（演算ブロック）１０４乃至演算部１０６を備える。
【００２８】
本実施形態における演算制御部１０１は、演算部（１０４乃至１０６）に対して、あるタスクの実行を指示する制御信号を入力する。より具体的には、演算制御部１０１は、演算部（１０４乃至１０６）における一連の演算処理をタスクとしてまとめて、当該タスクの実行を指定するコマンド（タスクコマンド）を生成する。そして、演算制御部１０１は、係るタスクコマンドを演算部（１０４乃至１０６）に対して入力する。
【００２９】
本実施形態における演算部１０４乃至演算部１０５は、上記説明した演算ブロックに相当する。係る演算部（１０４乃至１０６）は、特定の処理を実行する専用の演算処理ハードウェア（ＤＳＰ）等であってもよい。演算部（１０４乃至１０６）は、あるいは、汎用のＣＰＵ（Ｃｅｎｔｒａｌ　Ｐｒｏｃｅｓｓｉｎｇ　Ｕｎｉｔ）や、ＭＰＵ（Ｍｉｃｒｏ　Ｐｒｏｃｅｓｓｉｎｇ　Ｕｎｉｔ）であってもよい。係る演算部（１０４乃至１０６）においてどのような演算処理を実行するかは、任意に定めてよい。
【００３０】
また、本実施形態における演算装置１００は、上記演算制御部１０１から上記演算部（１０４乃至１０６）へ入力される制御信号を蓄えるコマンドバッファ（コマンドバッファ１０７、コマンドバッファ１０８、及び、コマンドバッファ１０９）を備える。係るコマンドバッファ（１０７乃至１０９）は、例えば、ＦＩＦＯ（Ｆｉｒｓｔ　ＩＮ　Ｆｉｒｓｔ　ＯＵＴ）構成を実現可能な、任意の一時記憶装置に構成されてもよい。以下、本実施形態においては、係るコマンドバッファ１０７、コマンドバッファ１０８、及び、コマン

ドバッファ１０９を、それぞれ、ＦＩＦＯ１０７、ＦＩＦＯ１０８、及び、ＦＩＦＯ１０９と称する場合がある。本実施形態は上記には限定されず、コマンドバッファ１０７乃至コマンドバッファ１０９は、ＦＩＦＯ構成以外の構成を採用してもよい。

【００３１】

本実施形態におけるコマンドバッファ（１０７乃至１０９）は、例えば、周知の揮発性メモリ（例えばＤＲＡＭ（Ｄｙｎａｍｉｃ　Ｒａｎｄｏｍ　Ａｃｃｅｓｓ　Ｍｅｍｏｒｙ）やＳＲＡＭ（Ｓｔａｔｉｃ　Ｒａｎｄｏｍ　Ａｃｃｅｓｓ　Ｍｅｍｏｒｙ）等）と、当該メモリを制御するコントローラ等とにより実現可能である。当該メモリは、デュアルポートメモリとして構成されてもよい。

【００３２】

また、本実施形態における演算装置１００は、各演算部（１０４乃至１０６）の間に介在するデータバッファ１１０、データバッファ１１１を備える。

【００３３】

データバッファ１１０は、前段の演算部１０４の出力側と、後段の演算部１０５の入力側との間に接続され、演算部１０４における演算結果を保持する。データバッファ１１０は、前段の演算部１０４における演算結果を、後段の演算部１０５に転送可能である。

【００３４】

データバッファ１１１は、前段の演算部１０５の出力側と、後段の演算部１０６の入力側との間に接続され、演算部１０５における演算結果を保持する。データバッファ１１１は、前段の演算部１０５における演算結果を後段の演算部１０６に転送可能である。本実施形態におけるデータバッファ１１０乃至データバッファ１１１は、例えば、周知の揮発性メモリ（例えばＤＲＡＭやＳＲＡＭ等）と、当該メモリを制御するコントローラ等により実現可能である。当該メモリは、デュアルポートメモリとして構成されてもよい。

【００３５】

また、本実施形態における演算装置１００は、ＦＩＦＯ（１０７乃至１０９）、及び、データバッファ（１１０、１１１）からの各種データ等の出力を制御可能な、タスク制御部（タスク制御部１１２、タスク制御部１１３、及び、タスク制御部１１４）を備える。タスク制御部（１１２乃至１１４）の具体的な構成と動作については、後述する。

【００３６】

以下、上記それぞれの構成要素間の接続態様について説明する。

【００３７】

演算制御部１０１はＦＩＦＯ（１０７乃至１０９）、並びに、タスク制御部（１１２乃至１１４）と通信可能に接続される。

【００３８】

演算部１０４は、ＦＩＦＯ１０７と通信可能に接続され、ＦＩＦＯ１０７から各種制御信号（例えば、上記コマンド等）等を取得する（受信する）。

【００３９】

また、演算部１０４は、第１のメモリ部１０２と通信可能に接続される。演算部１０４は、当該第１のメモリ部１０２から、演算処理の対象となるデータを取得してもよい。

【００４０】

また、演算部１０４は、データバッファ１１０と通信可能に接続され、データバッファ１１０の状態を取得可能である。演算部１０４は、データバッファ１１０に対して、演算部１０４における演算処理の結果等を出力する。

【００４１】

また、演算部１０４は、タスク制御部（１１２乃至１１４）のそれぞれに対して通信可能に接続される。演算部１０４は、あるタスクに関する演算処理の実行状態を表すステータス情報を、タスク制御部（１１２乃至１１４）に通知する。この場合、演算部１０４は、例えば、当該演算部１０４と、各タスク制御部（１１２乃至１１４）との間を接続する任意の通信バスにおける信号線を用いて、当該ステータス情報を表す信号を通知してもよい。

【００４２】
演算部１０５は、ＦＩＦＯ１０８と通信可能に接続される。演算部１０５は、ＦＩＦＯ１０８から各種制御信号（例えば、上記コマンド等）を取得する（受信する）。
【００４３】
また、演算部１０５は、データバッファ１１０と通信可能に接続され、当該データバッファ１１０から、演算処理の対象となるデータを取得する。
【００４４】
また、演算部１０５は、データバッファ１１１と通信可能に接続され、データバッファ１１１の状態を取得可能である。演算部１０５は、データバッファ１１１に対して、演算部１０５における演算処理の結果等を出力する。
【００４５】
また、演算部１０５は、タスク制御部（１１２乃至１１４）のそれぞれに対して通信可能に接続される。演算部１０５は、上記ステータス情報を、タスク制御部（１１２乃至１１４）に通知する。この場合、演算部１０５は、例えば、各タスク制御部（１１２乃至１１４）との間を接続する任意の通信バスにおける信号線を用いて、当該ステータス情報を表す信号を通知してもよい。
【００４６】
演算部１０６は、ＦＩＦＯ１０９と通信可能に接続される。演算部１０６は、ＦＩＦＯ１０９から各種制御信号（例えば、上記コマンド等）を取得する（受信する）。
【００４７】
また、演算部１０６は、データバッファ１１１と通信可能に接続され、当該データバッファ１１１から、演算処理の対象となるデータを取得する。
【００４８】
また、演算部１０６は、第２のメモリ部１０３と通信可能に接続され、第２のメモリ部１０３に対して、演算部１０６における演算処理の結果等を出力する。
【００４９】
また、演算部１０６は、タスク制御部（１１２乃至１１４）のそれぞれに対して通信可能に接続される。演算部１０６は、上記ステータス情報を、タスク制御部（１１２乃至１１４）に通知する。この場合、演算部１０６は、例えば、各タスク制御部（１１２乃至１１４）との間を接続する任意の通信バスにおける信号線を用いて、当該ステータス情報を表す信号を通知してもよい。
【００５０】
なお、演算部１０４は、第２のメモリ部１０３に通信可能に接続されていてもよい。演算部１０５は、第１のメモリ部１０２及び第２のメモリ部１０３に通信可能に接続されていてもよい。演算部１０６は、第１のメモリ部１０２に通信可能に接続されていてもよい。
【００５１】
タスク制御部（１１２乃至１１４）は、上記したように、演算制御部１０１と通信可能に接続され、演算制御部１０１から、各種制御信号（例えば、上記コマンド等）を受け付ける。
【００５２】
また、タスク制御部（１１２乃至１１４）は、上記したように各演算部（１０４乃至１０６）とそれぞれ通信可能に接続され、各演算部における演算処理のステータス情報を取得する。この場合、タスク制御部（１１２乃至１１４）は、各演算部（１０４乃至１０６）から上記ステータス情報に関する通知を受け付けてもよく、各演算部（１０４乃至１０６）に対して上記ステータス情報を問い合わせてもよい。
【００５３】
タスク制御部１１２は、ＦＩＦＯ１０７と通信可能に接続され、ＦＩＦＯ１０７からの演算部１０４に対するデータの出力を制御可能である。
【００５４】

タスク制御部１１３は、ＦＩＦＯ１０８と通信可能に接続され、ＦＩＦＯ１０８からの演算部１０５に対するデータの出力を制御可能である。

【００５５】

また、タスク制御部１１３は、データバッファ１１０と通信可能に接続され、データバッファ１１０からの演算部１０５に対するデータの出力を制御可能である。タスク制御部１１３は、データバッファ１１０に対する演算部１０５のアクセスを制御してもよい。

【００５６】

タスク制御部１１４は、ＦＩＦＯ１０９と通信可能に接続され、ＦＩＦＯ１０９からの演算部１０６に対するデータの出力を制御可能である。

【００５７】

タスク制御部１１４は、データバッファ１１１と通信可能に接続され、データバッファ１１１からの演算部１０６に対するデータの出力を制御可能である。タスク制御部１１４は、データバッファ１１１に対する演算部１０６のアクセスを制御してもよい。

【００５８】

図１に例示する具体例においては、説明の便宜上、第１のメモリ部１０２と第２のメモリ部１０３とを、別の機能ブロックとして記述しているが、本実施形態はこれには限定されない。即ち、第１のメモリ部１０２及び第２のメモリ部１０３は、書き込み、読み出しポートが個別に存在するメモリ部として一つに統合されていてもよい。また、第１のメモリ部１０２及び第２のメモリ部１０３は、調停により書き込み、読み出しが独立して実行可能なメモリを用いて一つに統合されていてもよい。

【００５９】

また、本実施形態において、演算装置１００を構成する上記各構成要素の間を接続する通信回線としては、任意の通信バスや、通信ネットワークを適宜採用してよい。係る通信バスや、通信ネットワークについては、周知の技術を採用してよいので、詳細な説明は省略する。

【００６０】

図１は、演算部の数が３個の場合の具体例を例示するが、本実施形態はこれには限定されない。演算部、データバッファ、コマンドバッファ（ＦＩＦＯ）、及びタスク制御部をいくつ設けるかは、演算装置に求められる処理性能や必要とされる演算処理の種類等に応じて、適宜選択されてよい。

【００６１】

次にタスク制御部（１１２乃至１１４）について詳細に説明する。図１に示すようにタスク制御部（１１２乃至１１４）は、それぞれ、タスク実行判定部（タスク実行判定部１１２ａ、タスク実行判定部１１３ａ、及び、タスク実行判定部１１４ａ）を有する。また、タスク制御部（１１２乃至１１４）は、それぞれ、タスク設定テーブル（タスク設定テーブル１１２ｂ、タスク設定テーブル１１３ｂ、及び、タスク設定テーブル１１４ｂ）を有する。

【００６２】

以下、図２を参照して、タスク設定テーブル（１１２ｂ乃至１１４ｂ）について説明する。

【００６３】

タスク設定テーブル（１１２ｂ乃至１１４ｂ）は、あるタスクに関する演算処理が実行される演算部（１０４乃至１０６のいずれか）を特定可能な情報を有する。

【００６４】

より具体的には、タスク設定テーブル（１１２ｂ乃至１１４ｂ）には、実行されるタスク毎に、開始演算ノード番号２０１と、終了演算ノード番号２０２とが設定される。即ち、タスク設定テーブル（１１２ｂ乃至１１４ｂ）には、実行されるタスクの数（タスクの種類の数）に相当する数の開始演算ノード番号２０１と、終了演算ノード番号２０２との組が設定される。タスク設定テーブル（１１２ｂ乃至１１４ｂ）においては、上記開始演算ノード番号２０１と、終了演算ノード番号２０２とが、タスクの種類を識別可能なタス

ク識別符号２０３に関連付けて保持（設定）されてもよい。図２に例示する具体例においては、タスクＡ乃至タスクＮについて、それぞれ開始演算ノード番号２０１と、終了演算ノード番号２０２とが、設定される。

【００６５】

開始演算ノード番号２０１は、あるタスクに関する最初の演算処理を実行する演算部（以下、「開始演算部」と称する場合がある）を特定可能な情報である。また、終了演算ノード番号２０２は、あるタスクに関する最後の演算処理を実行する演算部（以下、「終了演算部」と称する場合がある）を特定可能な情報である。

【００６６】

換言すると、開始演算ノード番号２０１は、あるタスクに関する演算処理が開始される演算部（開始演算部）を示す情報である。また、終了演算ノード番号は、あるタスクに関する演算処理が終了される演算部（終了演算部）を示す情報である。即ち、開始演算ノード番号により指定された開始演算部から、終了演算ノード番号により指定された終了演算部まで、当該タスクに関する演算処理が実行される。

【００６７】

各演算ノード番号と、実際の演算部（１０４乃至１０６）との関係は、例えば、タスク設定テーブル（１１２ｂ乃至１１４ｂ）や、タスク制御部（１１２乃至１１４）に予め設定されていてもよい。

【００６８】

タスク設定テーブル（１１２ｂ乃至１１４ｂ）は、タスク実行判定部（タスク実行判定部１１２ａ、タスク実行判定部１１３ａ、及び、タスク実行判定部１１４ａ）とそれぞれ接続されてもよい。

【００６９】

タスク設定テーブル（１１２ｂ乃至１１４ｂ）は、それぞれ開始演算ノード番号２０１と終了演算ノード番号２０２とを、タスク実行判定部（１１２ａ乃至１１４ａ）に提供する。この場合、タスク実行判定部（１１２ａ乃至１１４ａ）が、それぞれ、タスク設定テーブル（１１２ｂ乃至１１４ｂ）を参照してもよい。

【００７０】

タスク設定テーブル（１１２ｂ乃至１１４ｂ）は、例えば、タスク実行判定部（１１２ａ乃至１１４ａ）から上記タスク識別符号を受け付けてもよい。そして、タスク設定テーブル（１１２ｂ乃至１１４ｂ）は、当該識別符号に該当するタスクに関する開始演算ノード番号２０１と終了演算ノード番号２０２とを、タスク実行判定部（１１２ａ乃至１１４ａ）に提供してもよい。

【００７１】

各タスク実行判定部（１１２ａ乃至１１４ａ）は、当該タスク実行判定部を有するタスク制御部（１１２乃至１１４）が制御対象とする上記演算部（１０４乃至１０６）において、タスクを実行可能であるか否かを判定する。より具体的には、各タスク実行判定部（１１２ａ乃至１１４ａ）は、制御対象である上記演算部が、タスクを実行可能なタイミングを判定する。

【００７２】

本実施形態において、タスク制御部１１２が制御対象とする上記演算部は、演算部１０４である。同様に、タスク制御部１１３が制御対象とする上記演算部は、演算部１０５である。同様に、タスク制御部１１４が制御対象とする上記演算部は、演算部１０６である。

【００７３】

上記各タスク実行判定部（１１２ａ乃至１１４ａ）は、各演算部（演算部１０４乃至１０６）のステータス情報と、上記タスク設定テーブルから提供された開始演算ノード番号及び終了演算ノード番号とに基づいて、各演算部（演算部１０４乃至１０６）がタスクを実行可能なタイミングを判定する。係るタイミングの具体的な判定方法については、後述する。なお、本実施形態における各タスク実行判定部（１１２ａ乃至１１４ａ）は、タス

ク制御部（１１２乃至１１４）と通信可能に接続されている各構成要素と、通信可能である。

【００７４】

次に、上記のように構成された本実施形態における演算装置１００の動作について説明する。以下の説明においては、上記第１のメモリ部１０２に、演算部（１０４乃至１０６）における演算処理の対象となるデータが格納されていることを想定する。

【００７５】

また、以下の説明においては、説明の便宜上、実行されるタスクの数が２つである場合を想定する。２つのタスクのうち、第１のタスク（以下「タスクＡ」と称する場合がある）は、演算部１０４、１０５、及び、１０６において実行されることを想定する。また、２つのタスクのうち、第２のタスク（以下「タスクＢ」と称する場合がある）は、演算部１０５、及び演算部１０６において実行されることを想定する。

【００７６】

また、以下の説明においては、演算制御部１０１から出力される、特定のタスクの実行を指定するコマンドを「タスクコマンド」と称する。演算制御部１０１から出力される、タスクＡの実行を指定する上記コマンドを、タスクコマンドＡと称する。同様に、演算制御部１０１から出力される、タスクＢの実行を指定する上記コマンドを、タスクコマンドＢと称する。

【００７７】

上記タスクコマンドは、図３に例示するように、当該タスクコマンドより実行されるタスク（この場合はタスクＡまたはタスクＢ）を判別可能なタスク識別符号（インデックス等）３０１を有する。

【００７８】

また、上記タスクコマンドは、当該タスクコマンドにより実行が指定されたタスク（この場合はタスクＡまたはタスクＢ）と実行順序において依存関係がある他のタスクを指定するタスク依存関係情報を有する。より具体的には、タスク依存関係情報は、図３に例示する前タスク依存情報３０２と、後タスク依存情報３０３とを含む。

【００７９】

前タスク依存情報３０２は、上記コマンドにおいて指定されたタスクと、係るタスクより前に実行されるタスクとの間に依存関係が存在するか否かを表す情報である。また、後タスク依存情報は、上記コマンドにおいて指定されたタスクと、係るタスクより後に実行されるタスクとの間に依存関係が存在するか否かを表す情報である。

【００８０】

以下、上記タスクコマンドにおいて指定されたタスクに対して、前タスク依存情報が設定されたタスクを「前依存タスク」と称する場合がある。また、上記コマンドにおいて指定されたタスクに対して、後タスク依存情報が設定されたタスクを「後依存タスク」と称する場合がある。上記タスクコマンドには、タスク識別符号３０１、前タスク依存情報３０２、後タスク依存情報３０３以外に、タスクの実行に関する任意のデータが含まれてもよい。

【００８１】

以下の説明においては、図４または図５に例示するタスクコマンドＡ、タスクコマンドＢが、演算制御部１０１より出力されること想定とする。

【００８２】

図４に例示する具体例においては、タスクコマンドＡ（４０１）には、タスク識別符号として「タスクＡ」が設定され、タスクコマンドＢ（４０２）には、タスク識別符号として「タスクＢ」が設定される。そして、図４に例示する具体例においては、これらのタスクコマンドＡ、タスクコマンドＢには、タスク依存関係情報が設定されていない。これより、各演算部（１０４乃至１０６）は、タスクＡに関する演算処理とタスクＢに関する演算処理とを独立して実行可能である。

【００８３】

図５に例示する具体例においては、タスクコマンドＡ（５０１）には、タスク識別符号として「タスクＡ」が設定され、タスクコマンドＢ（５０２）には、タスク識別符号として「タスクＢ」が設定される。そして、タスクコマンドＡには、後タスク依存情報３０３として「タスクＢ」が設定されており、タスクコマンドＢには、前タスク依存情報３０２として「タスクＡ」が設定されている。この場合、各演算部（１０４乃至１０６）は、タスクＡに関する演算処理の実行が完了した後に、タスクＢに関する演算処理を実行する必要がある。

【００８４】

また、上記タスク設定テーブル（１１２ｂ乃至１１４ｂ）には、図６に例示するように、タスクＡに対する開始演算ノード番号として、「＃０」が、終了演算ノード番号として「＃２」が格納されていることを想定する。また、上記タスク設定テーブル（１１２ｂ乃至１１４ｂ）には、図６に例示するように、タスクＢに対する開始演算ノード番号として、「＃１」が、終了演算ノード番号として「＃２」が予め格納されていることを想定する。

【００８５】

演算ノード番号「＃０」、「＃１」、及び、「＃２」は、それぞれ、演算部１０４、演算部１０５、及び、演算部１０６を示す。演算ノード番号と、各演算部との関係は、予めタスク設定テーブル（１１２ｂ乃至１１４ｂ）や、タスク制御部（１１２乃至１１４）に設定されていてもよい。

【００８６】

また、以下においては、開始演算ノード番号に該当する開始演算部から、演算ノード番号が増大する順に、順次終了演算ノード番号に該当する終了演算部まで、タスクに関する演算処理が実行されることを前提とする。即ち、タスクＡについては、演算ノード番号が「＃０」、「＃１」、「＃２」の順に、それぞれの演算ノード番号に該当する演算部における処理が実行される。また、タスクＢについては、演算ノード番号が、「＃１」、「＃２」の順に、それぞれの演算ノード番号に該当する演算部における処理が実行される。

【００８７】

それぞれのタスク（タスクＡ及びタスクＢ）について、各演算部（１０４乃至１０６）において実行される処理の具体的な内容は、適宜定められてよい。

【００８８】

また、以下の説明においては、各演算部（１０４乃至１０６）における演算処理の実行状態を表すステータス情報が各演算部（１０４乃至１０６）から、それぞれタスク制御部（１１２乃至１１４）に入力される。係るステータス情報は、「待機中」、「処理実行中」、及び、「処理終了」を表す情報である。下記説明においては、「待機中」を表すステータス情報に「０」が、「処理実行中」を表すステータス情報に「１」が、「処理終了」を表すステータス情報に「２」が、それぞれ割り当てられることを想定する。なお、本実施形態はこれには限定されず、ステータス情報を表現する形式は、適宜定められてよい。また、本実施形態においては、各演算部（１０４乃至１０６）に関して、「待機中」、「処理実行中」、及び、「処理終了」以外の状態を表すステータス情報が、タスク制御部（１１２乃至１１４）に入力されてもよい。

【００８９】

以下、主に各タスク制御部（１１２乃至１１４）の動作を中心に、本実施形態における演算装置１００の動作について説明する。

【００９０】

まず、演算制御部１０１が、タスクＡとタスクＢとの実行を指定するコマンド（タスクコマンドＡ、タスクコマンドＢ）をそれぞれ生成する。そして、演算制御部１０１は、タスクコマンドＡ、及び、タスクコマンドＢを、ＦＩＦＯ１０７乃至１０９、並びに、タスク制御部（１１２乃至１１４）に入力する。

【００９１】

各ＦＩＦＯ１０７乃至１０９は、入力されたタスクコマンドＡ及びタスクコマンドＢを

保持する。同様に、各タスク制御部（１１２乃至１１４）は、入力されたタスクコマンドＡ及びタスクコマンドＢを保持してもよい。

【００９２】

各タスク制御部（１１２乃至１１４）は、タスクコマンドＡが入力された場合、下記ステップ（図７乃至図９に例示）に従って、各タスク制御部（１１２乃至１１４）が制御対象とする演算部（１０４乃至１０６）において、タスクＡが実行可能であるか否かを判定する。

【００９３】

タスクコマンドＡが入力された場合、各タスク制御部（１１２乃至１１４）は、それぞれが有するタスク設定テーブル（１１２ｂ乃至１１４ｂ）を参照し、タスクＡに関して設定された情報を取得する（ステップＳ７０１）。より具体的には、各タスク制御部（１１２乃至１１４）はそれぞれ、タスク設定テーブル（１１２ｂ乃至１１４ｂ）から、タスクＡに対する開始演算ノード番号と、終了演算ノード番号とを読み出す。この場合、図６に例示するように、タスクＡに対する開始演算ノード番号２０１には「＃０」（演算部１０４に対応）が、終了演算ノード番号には「＃２」（演算部１０６に対応）が設定されている。

【００９４】

次に、各タスク制御部（１１２乃至１１４）は、タスク設定テーブル（１１２ｂ乃至１１４ｂ）の設定に基づいて、各演算部（１０４乃至１０６）が、タスクコマンドにおいて指定されたタスクの処理に含まれるか否かを確認する（ステップＳ７０２）。より具体的には、各タスク制御部（１１２乃至１１４）は、例えば、それぞれの制御対象である各演算部（１０４乃至１０６）が、タスクＡに関する演算処理を実行する演算部として指定されているか否かを確認する。

【００９５】

上記ステップＳ７０２における確認の結果、制御対象である演算部がタスク（例えばタスクＡ）に関する演算処理を実行しない場合（ステップＳ７０３においてＮＯ）、各タスク制御部（１１２乃至１１４）は、ステップＳ７０１に戻って処理を実行する。この場合、例えば、次のタスク（この場合はタスクＢ）についてステップＳ７０１から処理を実行してもよい。また、全てのタスクについて処理を実行した場合は、演算制御部１０１からの次のタスクコマンドの入力を待ってもよい。

【００９６】

タスクＡについては、各演算部（１０４乃至１０６）が、タスクＡ処理に関する演算処理を実行する演算部として指定されている。よって、各タスク制御部（１１２乃至１１４）によるステップＳ７０３における確認結果は、「ＹＥＳ」（「含まれる」）となる。

【００９７】

上記ステップＳ７０３における確認結果がＹＥＳの場合、各タスク制御部（１１２乃至１１４）は、演算部（１０４乃至１０６）のステータス情報を取得する（ステップＳ７０４）。この場合、タスクコマンドＡが入力された時点では、各演算部（１０４乃至１０６）は演算処理を実行していないので、各演算部（１０４乃至１０６）のステータス情報は「待機中（０）」を表す。

【００９８】

次に、各タスク制御部（１１２乃至１１４）は、ステップＳ７０４において取得したステータス情報を用いて、各演算部（１０４乃至１０６）がタスクを実行可能であるか否かを判定する（ステップＳ７０５）。より具体的には、各タスク制御部（１１２乃至１１４）は、それぞれの制御対象である各演算部（１０４乃至１０６）が、タスクＡに関する演算処理を実行可能であるか否かを判定する。

【００９９】

以下、図８に例示するフローチャートを参照してステップＳ７０５における処理について説明する。まず、各タスク制御部（１１２乃至１１４）は、それぞれの制御対象である演算部（１０４乃至１０６）に関するステータス情報が、「処理実行中（１）」であるか

否かを確認する（ステップＳ８０１）。この場合、上記ステップＳ７０４の結果から、各演算部（１０４乃至１０６）に関するステータス情報は「待機中（０）」を表し、ステップＳ８０２における確認結果は「ＮＯ」となる。ステップＳ８０１における確認の結果、ステップＳ８０２において「ＹＥＳ」と判定された場合、当該制御対象である演算部は、何らかのタスクに関する演算処理を実行中である。よって、各タスク制御部（１１２乃至１１４）は、それぞれの制御対象である演算部（１０４乃至１０６）に対するタスクの投入（入力）は不可であると判定する（ステップＳ８０９）。

【０１００】

ステップＳ８０２においてＮＯの場合、タスク制御部（１１２乃至１１４）は、それぞれの制御対象である演算部（１０４乃至１０６）が、タスクコマンドにより指定されたタスクの開始演算部に該当するか否かを確認する（ステップＳ８０３）。この場合、タスク制御部（１１２乃至１１４）は、それぞれの制御対象である演算部（１０４乃至１０６）がタスクＡの開始演算部に該当するか否かを判定する。

【０１０１】

タスク設定テーブル（１１２ｂ乃至１１４ｂ）には、タスクＡの開始演算ノード番号が「＃０」と設定されている。これより、開始演算部は演算部１０４である。よって、ステップＳ８０２における確認の結果、タスク制御部１１２は制御対象である演算部１０４が、タスクＡの開始演算部に該当すると判定する（ステップＳ８０４においてＹＥＳ）。

【０１０２】

また、この場合、他のタスク制御部１１３、及び、タスク制御部１１４は、ステップＳ８０３における確認の結果、制御対象である演算部（演算部１０５、演算部１０６）が、タスクＡの開始演算部に該当しないと判定する（ステップＳ８０４においてＮＯ）。

【０１０３】

この場合、タスク制御部１１３及び１１４は、各演算部（演算部１０５及び１０６）の前段の演算部において、タスクＡに関する処理が完了しているか否かを確認する（ステップＳ８０６）。

【０１０４】

より具体的には、タスク制御部１１３は、例えば、タスク設定テーブル１１３ｂを参照することにより、制御対象である演算部１０５の前段となる演算部（演算部１０４）を特定する。タスク制御部１１３は、演算部１０５の演算ノード番号（「＃１」）を参照し、係る演算ノード番号より一つ前の演算ノード番号（「＃０」）に対応する演算部１０４を特定してもよい。

【０１０５】

そして、タスク制御部１１３は、当該演算部１０４のステータス情報を参照することにより、タスクＡに関する演算処理が終了しているか否かを確認する。この場合、演算部１０４におけるタスクＡに関する演算処理は終了していないので、タスク制御部１１３は、ステップＳ８０７においてＮＯと判定する。

【０１０６】

同様に、タスク制御部１１４は、演算部１０６に対する前段の演算部（演算部１０５）において、タスクＡに関する演算処理が終了していないので、ステップＳ８０７においてＮＯと判定する。

【０１０７】

これにより、タスク制御部１１３、及び、タスク制御部１１４は、制御対象である演算部（演算部１０５及び１０６）に対するタスクＡの投入は不可であると判定する（ステップＳ８０９）。

【０１０８】

図７に戻って、上記ステップＳ８０９の結果、タスク制御部１１３、及び、タスク制御部１１４は、演算部１０５及び１０６におけるタスクＡの処理を抑制する（ステップＳ７１０）。

【０１０９】

再び図８に戻って、次に、タスク制御部１１２は、タスクコマンドより指定されたタスク（この場合はタスクＡ）に関して、依存関係にあるタスクを確認する（ステップＳ８０５）。以下、図９を参照して、ステップＳ８０５における処理について説明する。

【０１１０】

まず、タスク制御部１１２は、演算制御部１０１から入力されたタスクコマンド（この場合タスクコマンドＡ）に設定されたタスク依存関係情報（図３における３０２、３０３）を参照する。そして、タスク制御部１１２は、タスクＡと依存関係があるタスクが存在するか否かを確認する（ステップＳ９０１）。

【０１１１】

この場合、タスクコマンドＡには前タスク依存情報が設定されていないことから、タスク制御部１１２は依存関係にある前依存タスクが無いと判定する（ステップＳ９０２においてＮＯ）。

【０１１２】

そして、タスク制御部１１２は、制御対象である演算部１０４に対してタスク（この場合はタスクＡ）を投入可能であると判定する（ステップＳ９０３）。即ち、タスク制御部１１２は、制御対象である演算部１０４においてタスクＡを実行可能であると判定する。ステップＳ９０２においてＹＥＳの場合の動作については、後述する。

【０１１３】

再び図７に戻り、ステップＳ９０３の判定結果から、タスク制御部１１２は、ステップＳ７０５において、制御対象である演算部１０４に対してタスクＡを投入可能であると判定する（ステップＳ７０６においてＹＥＳ）。

【０１１４】

次に、タスク制御部１１２は、演算部１０４に対してタスク実行制御信号（タスクコマンド）及び、タスク実行に必要なデータを出力する（ステップＳ７０８）。より具体的には、タスク制御部１１２は、ＦＩＦＯ１０７を制御して、ＦＩＦＯ１０７が保持するタスクコマンドＡを演算部１０４に出力する。タスク制御部１１２は、必要に応じて、第１のメモリ部１０２から、タスクＡの実行に必要なデータを読み込むよう、演算部１０４を制御してもよい。

【０１１５】

ステップＳ７０８における処理の結果、演算部１０４において、タスクＡに関する演算処理が実行される。演算部１０４における、タスクＡに関する演算処理の結果は、データバッファ１１０に一旦格納される。データバッファ１１０がほぼ演算部１０４の演算結果で満たされた場合（空き容量が少なくなった場合）は、データバッファ１１０が、特定の信号（「ＡｌｍｏｓｔＦｕｌｌ」信号）を演算部１０４に出力する。この場合、演算部１０４における演算処理はストールする。データバッファ１１０が「ＡｌｍｏｓｔＦｕｌｌ」信号を出力する空き容量のサイズは、適宜設定可能としてよい。

【０１１６】

タスク制御部１１２は、ステップＳ７０８の後、次のタスクの制御へ遷移する（ステップＳ７０９）。より具体的には、タスク制御部１１２は、例えば、演算部１０４のステータス情報を確認し、この場合はタスクＡに関する演算処理の終了を待機する。そして、タスク設定テーブル１１２ｂと、入力されたタスクコマンドとを参照し、ステップＳ７０１から次のタスク（この場合はタスクＢ）に関する制御を開始する。

【０１１７】

演算部１０４において、タスクＡに関する処理が終了すると、演算部１０４のステータス情報が「処理終了（２）」となる。

【０１１８】

上記ステップＳ７１０において制御対象である演算部１０５に対するタスクＡの投入を抑制したタスク制御部１１３は、再びステップＳ７０４において、演算部１０４のステータス情報を取得する。この場合、係るステータス情報は「処理終了（２）」である。

【０１１９】

次に、タスク制御部１１３は、ステップＳ７０５（ステップＳ８０１乃至Ｓ８０６）の処理を実行する。その結果、タスク制御部１１３は、前段の演算部１０４においてタスクＡに関する演算処理が終了していることから、演算部１０５にタスクＡを投入可能であると判定する（ステップＳ８０８）。即ち、この場合、ステップＳ７０６における判定結果がＹＥＳとなる。

【０１２０】

この際、タスク制御部１１４は、再びステップＳ７０４乃至Ｓ７０６を実行する。この場合、演算部１０６に対する前段の演算部１０５において、タスクＡに関する演算処理が終了していない。よってタスク制御部１１４は、演算部１０６に対するタスクＡの投入は不可であると判定する。

【０１２１】

次に、タスク制御部１１３は、制御対象である演算部１０５に対してタスク実行制御信号（タスクコマンド）及び、タスク実行に必要なデータを出力する（ステップＳ７０８）。より具体的には、タスク制御部１１３は、ＦＩＦＯ１０８を制御して、ＦＩＦＯ１０８が保持するタスクコマンドＡを演算部１０５に出力する。また、タスク制御部１１３は、データバッファ１１０を制御して、演算部１０４の出力（演算結果）を、演算部１０５に出力する。タスク制御部１１３は、必要に応じて、第１のメモリ部１０２から、タスクＡの実行に必要なデータを読み込むよう、演算部１０５を制御してもよい。

【０１２２】

ステップＳ７０８における処理の結果、演算部１０５において、タスクＡに関する演算処理が実行される。演算部１０５におけるタスクＡに関する演算処理の結果は、データバッファ１１１に一旦格納される。データバッファ１１１がほぼ演算部１０５の演算結果で満たされた場合（空き容量が少なくなった場合）は、データバッファ１１１が「ＡｌｍｏｓｔＦｕｌｌ」の信号を演算部１０５に出力する。この場合、演算部１０５における演算処理はストールする。データバッファ１１１が「ＡｌｍｏｓｔＦｕｌｌ」信号を出力する空き容量のサイズは、適宜設定可能である。

【０１２３】

タスク制御部１１３は、ステップＳ７０８の後、次のタスクの制御へ遷移する（ステップＳ７０９）。より具体的には、タスク制御部１１３は、例えば、演算部１０５のステータス情報を確認し、この場合はタスクＡに関する演算処理の終了を待機する。そして、タスク設定テーブル１１３ｂと、入力されたタスクコマンドとを参照し、ステップＳ７０１から次のタスク（この場合はタスクＢ）に関する制御を開始する。

【０１２４】

演算部１０４におけるタスクＡに関する演算処理が終了した際、タスク制御部１１２は、ステップＳ７０１から処理を開始する。この場合、タスク制御部１１２は、タスク設定テーブル１１２ｂを参照する（ステップＳ７０１）。タスク制御部１１２は、タスク設定テーブル１１２ｂの内容と、演算制御部１０１から出力されたタスクコマンドとに基づいて、次のタスクは「タスクＢ」であると確認する。

【０１２５】

次に、タスク制御部１１２は、制御対象である演算部１０４が、タスクＢの処理に含まれるか確認する（ステップＳ７０２）。タスク設定テーブル１１２ｂを参照すると、タスクＢの開始演算ノード番号は「＃１」である。演算ノード番号「＃１」は、演算部１０５を示す。このため、タスク制御部１１２は、制御対象である演算部１０４が「タスクＢ」の処理に含まれないと判定する（ステップＳ７０３においてＮＯ）。この場合、タスク制御部１１２と、演算部１０４とは、演算制御部１０１から次のタスクが出力されるまで、待機状態となる。

【０１２６】

演算部１０５においてタスクＡに関する処理が終了すると、演算部１０５のステータス情報が「処理終了（２）」を表すように更新される。

【０１２７】

上記ステップＳ７１０において制御対象である演算部１０６に対するタスクＡの投入を抑制したタスク制御部１１４は、再びステップＳ７０４において、演算部１０５のステータス情報を取得する。この場合、係るステータス情報は「処理終了（２）」を表す。

【０１２８】

次に、タスク制御部１１４は、ステップＳ７０５（ステップＳ８０１乃至Ｓ８０７）の処理を実行する。その結果、タスク制御部１１４は、前段の演算部１０５においてタスクＡに関する演算処理が終了していることから、演算部１０６にタスクＡを投入可能であると判定する（ステップＳ８０８）。

【０１２９】

次に、タスク制御部１１４は、制御対象である演算部１０６に対してタスク実行制御信号（タスクコマンド）及び、タスク実行に必要なデータを出力する（ステップＳ７０８）。より具体的には、タスク制御部１１４は、ＦＩＦＯ１０９を制御して、ＦＩＦＯ１０９が保持するタスクコマンドＡを演算部１０６に出力する。また、タスク制御部１１４は、データバッファ１１１を制御して、演算部１０５の出力（演算結果）を、演算部１０６に出力する。タスク制御部１１４は、必要に応じて、第１のメモリ部１０２から、タスクＡの実行に必要なデータを読み込むよう、演算部１０６を制御してもよい。

【０１３０】

ステップＳ７０８における処理の結果、演算部１０６において、タスクＡに関する演算処理が実行される。演算部１０６におけるタスクＡに関する演算処理の結果は、第２のメモリ部１０３に出力されてもよい。

【０１３１】

タスク制御部１１４は、ステップＳ７０８の後、次のタスクの制御へ遷移する（ステップＳ７０９）。より具体的には、タスク制御部１１４は、例えば、演算部１０６のステータス情報を確認し、この場合はタスクＡに関する演算処理の終了を待機する。そして、タスク設定テーブル１１４ｂと、入力されたタスクコマンドとを参照し、ステップＳ７０１から次のタスク（この場合はタスクＢ）に関する制御を開始する。

【０１３２】

演算部１０５におけるタスクＡに関する演算処理が終了した際、タスク制御部１１３は、ステップＳ７０１から処理を開始する。この場合、タスク制御部１１３は、タスク設定テーブル１１３ｂを参照する（ステップＳ７０１）。タスク制御部１１３は、タスク設定テーブル１１３ｂの内容と、演算制御部１０１から出力されたタスクコマンドとに基づいて、次のタスクは「タスクＢ」であると確認する。

【０１３３】

次に、タスク制御部１１３は、制御対象である演算部１０５が、タスクＢの処理に含まれるか否かを確認する（ステップＳ７０２）。タスク設定テーブル１１３ｂを参照すると、タスクＢの開始演算ノード番号は「＃１」である。演算ノード番号「＃１」は、演算部１０５を示す。このため、タスク制御部１１３は、制御対象である演算部１０５が「タスクＢ」の処理に含まれると判定する（ステップＳ７０３においてＹＥＳ）。

【０１３４】

次に、タスク制御部１１３は、ステップＳ７０４及びＳ７０５を実行する。この際、タスク制御部１１３は、ステップＳ８０３において、制御対象である演算部１０５が、タスクＢの開始演算部であることを確認する（ステップＳ８０４においてＹＥＳ）。

【０１３５】

次に、タスク制御部１１３は、ステップＳ８０５を実行する。この際、タスク制御部１１３は、タスクコマンドＢのタスク依存関係情報（図３における３０２、３０３）を参照し、依存関係があるタスクが存在するか否か確認する（ステップＳ９０１）。

【０１３６】

まず、図４に示すタスクコマンドＡ（４０１）、タスクコマンドＢ（４０２）のように、タスク依存関係情報が設定されていない場合について説明する。この場合、タスクＢとタスクＡは独立して実行可能である。この場合、タスク制御部１１３は依存関係にあるタ

スクが無いと判定する（ステップＳ９０２においてＮＯ）。
【０１３７】
そして、タスク制御部１１３は、制御対象である演算部１０５に対してタスク（この場合はタスクＢ）を投入可能（即ち、演算部１０５がタスクＢを実行可能）であると判定する（ステップＳ９０３）。
【０１３８】
この場合、タスク制御部１１３は、制御対象である演算部１０５に対してタスク実行制御信号（タスクコマンド）及び、タスク実行に必要なデータを出力する（ステップＳ７０８）。より具体的には、タスク制御部１１３は、ＦＩＦＯ１０８を制御して、ＦＩＦＯ１０８が保持するタスクコマンドＢを演算部１０５に出力する。タスク制御部１１３は、必要に応じて、第１のメモリ部１０２から、タスクＢの実行に必要なデータを読み込むよう、演算部１０５を制御してもよい。
【０１３９】
ステップＳ７０８における処理の結果、演算部１０５において、タスクＢに関する演算処理が実行される。以降の処理は、上記説明した演算部１０５におけるタスクＡに関する演算処理と同様である。
【０１４０】
次に、仮に、図５に示すタスクコマンドＡ（５０１）、タスクコマンドＢ（５０２）のように、タスク依存関係情報が設定された場合について説明する。この場合、タスクＢとタスクＡとの間に依存関係が存在することから、タスクＢは、タスクＡの終了後に実行される必要がある。この場合、タスク制御部１１３がステップＳ９０１を実行した結果、ステップＳ９０２における確認結果がＹＥＳとなる。
【０１４１】
タスク制御部１１３は、タスク設定テーブル１１３ｂを参照し、タスクＢと依存関係があるタスク（タスクＡ）の終了演算部を特定する（ステップＳ９０４）。タスクＡの終了演算部は、演算部１０６（終了演算ノード番号が「＃２」）である。
【０１４２】
次に、タスク制御部１１３は、ステップＳ９０４において特定した演算部（演算部１０６）のステータス情報を確認する。そして、当該演算部１０６のステータス情報に基づいて、演算部１０６において、タスクＢと依存関係のあるタスク（タスクＡ）に関する演算処理が終了しているか確認する（ステップＳ９０５）。
【０１４３】
この段階では、上記の通り、演算部１０５においてタスクＡに関する演算処理が終了しているものの、演算部１０６におけるタスクＡに関する処理は終了していない場合がある。演算部１０６におけるタスクＡに関する処理は終了していない場合、ステップＳ９０６における確認結果はＮＯとなる。この場合、タスク制御部１１３は、演算部１０５に対するタスクＢの投入は不可であると判定する（ステップＳ９０７）。この場合、タスク制御部１１３は、タスクＢの実行を抑制する（ステップＳ７１０）。より具体的には、タスク制御部１１３は、例えば、ＦＩＦＯ１０８からのタスクコマンドＢの出力等を抑制してもよい。
【０１４４】
次に、演算部１０６においてタスクＡに関する処理が終了すると、演算部１０６のステータス情報が「処理終了（２）」となる。
【０１４５】
再び、図５に示すタスクコマンドＡ（５０１）、タスクコマンドＢ（５０２）のように、タスク依存関係情報が設定された場合について説明する。
【０１４６】
上記ステップＳ７１０において、制御対象である演算部１０６に対するタスクＢの投入を抑制したタスク制御部１１３は、ステップＳ７０４において、演算部１０６のステータス情報を取得する。この場合、係るステータス情報は「処理終了（２）」である。

【０１４７】
次に、タスク制御部１１３は、ステップＳ７０５（ステップＳ８０１乃至Ｓ８０５）の処理を実行する。その結果、タスク制御部１１３は、タスクＢと依存関係のあるタスクＡに関する演算処理が終了している（ステップＳ９０１乃至ステップＳ９０５）と判定する。これより、タスク制御部１１３は、演算部１０５にタスクＢを投入可能であると判定する（ステップＳ９０６においてＹＥＳ）。即ち、この場合、ステップＳ７０６における判定結果がＹＥＳとなる。
【０１４８】
次に、タスク制御部１１３は、制御対象である演算部１０５に対してタスク実行制御信号（タスクコマンド）及び、タスク実行に必要なデータを出力する（ステップＳ７０８）。より具体的には、タスク制御部１１３は、ＦＩＦＯ１０８を制御して、ＦＩＦＯ１０８が保持するタスクコマンドＢを演算部１０５に出力する。また、タスク制御部１１３は、データバッファ１１０を制御して、データバッファ１１０の内容を、演算部１０５に出力してもよい。タスク制御部１１３は、必要に応じて、第１のメモリ部１０２から、タスクＢの実行に必要なデータを読み込むよう、演算部１０５を制御してもよい。
【０１４９】
ステップＳ７０８における処理の結果、演算部１０５において、タスクＢに関する演算処理が実行される。演算部１０５におけるタスクＢに関する演算処理の結果は、データバッファ１１１に一旦格納される。以下の処理は、上記説明した演算部１０５におけるタスクＡの処理と同様である。
【０１５０】
演算部１０６におけるタスクＡに関する演算処理が終了した際、タスク制御部１１４は、ステップＳ７０１から処理を開始する。この場合、タスク制御部１１４は、タスク設定テーブル１１４ｂを参照する（ステップＳ７０１）。タスク制御部１１４は、タスク設定テーブル１１４ｂの内容と、演算制御部１０１から出力されたタスクコマンドに基づいて、次のタスクは「タスクＢ」であると確認する。
【０１５１】
次に、タスク制御部１１４は、制御対象である演算部１０６が、タスクＢの処理に含まれるか確認する（ステップＳ７０２）。タスク設定テーブル１１２ｂを参照すると、タスクＢの終了演算ノード番号は「＃２」である。このため、タスク制御部１１４は、制御対象である演算部１０６が「タスクＢ」の処理に含まれると判定する（ステップＳ７０３においてＹＥＳ）。
【０１５２】
次に、タスク制御部１１４は、ステップＳ７０４及びＳ７０５を実行する。この際、タスク制御部１１４は、ステップＳ８０３において、制御対象である演算部１０６がタスクＢの開始演算部では無いことを確認する（ステップＳ８０４においてＮＯ）。
【０１５３】
この場合、タスク制御部１１４及は、演算部１０６の前段の演算部１０５において、タスクＢに関する処理が完了しているか確認する（ステップＳ８０６）。この場合、演算部１０５におけるタスクＢに関する演算処理は終了していないので、タスク制御部１１４は、ステップＳ８０７においてＮＯと判定する。これにより、タスク制御部１１４は、制御対象である演算部１０６に対するタスクＢの投入は不可であると判定する（ステップＳ８０９）。
【０１５４】
図７に戻って上記ステップＳ８０９の結果、演算部１０６におけるタスクＢの処理は抑制される（ステップＳ７１０）。
【０１５５】
次に、演算部１０５においてタスクＢに関する処理が終了すると、演算部１０５のステータス情報が「処理終了（２）」となる。
【０１５６】

上記ステップＳ７１０において制御対象である演算部１０６に対するタスクＢの投入を抑制したタスク制御部１１４は、再びステップＳ７０４において、演算部１０５のステータス情報を取得する。この場合、係るステータス情報は「処理終了（２）」である。

【０１５７】

次に、タスク制御部１１４は、ステップＳ７０５（ステップＳ８０１乃至Ｓ８０７）の処理を実行する。その結果、タスク制御部１１４は、前段の演算部１０５においてタスクＢに関する演算処理が終了していることから、演算部１０６にタスクＢを投入可能であると判定する（ステップＳ８０８）。

【０１５８】

次に、タスク制御部１１４は、制御対象である演算部１０６に対してタスク実行制御信号（タスクコマンド）及び、タスク実行に必要なデータを出力する（ステップＳ７０８）。より具体的には、タスク制御部１１４は、ＦＩＦＯ１０９を制御して、ＦＩＦＯ１０９が保持するタスクコマンドＢを演算部１０６に出力する。また、タスク制御部１１４は、データバッファ１１１を制御して、演算部１０５の出力（演算結果）を、演算部１０６に出力する。タスク制御部１１４は、必要に応じて、第１のメモリ部１０２から、タスクＢの実行に必要なデータを読み込むよう、演算部１０６を制御してもよい。

【０１５９】

ステップＳ７０８における処理の結果、演算部１０６において、タスクＢに関する演算処理が実行される。演算部１０６におけるタスクＢに関する演算処理の結果は、第２のメモリ部１０３に出力されてもよい。

【０１６０】

演算部１０６において、タスクＢの処理が終了すると、演算部１０６のステータス「処理終了（２）」となり、演算制御部１０１から投入されたタスクＡ、タスクＢに関する演算処理が終了する。

【０１６１】

以上説明したように、本実施形態によれば、演算装置１００において、各演算部（１０４乃至１０６）が効率的に稼働するようにタスクの実行が制御される。ここで、係る演算装置１００は、複数の演算部（１０４乃至１０６）が、データバッファ（１１０乃至１１１）を介してパイプライン状に接続された構成を有する。即ち、本実施形態における演算装置１００は、複数の演算部（１０４乃至１０６）を用いてパイプライン的に演算処理を実行するタスクを実行する際、各演算部におけるアイドル状態やレイテンシを削減することが可能である。これにより、本実施形態における演算装置１００は、パイプラインのオーバーヘッドを低減可能である。

【０１６２】

より具体的には、本実施形態における演算装置１００は、特定のタスクに関する全ての演算処理の終了を待たずに、一部の演算部において、当該特定のタスクと依存関係の無い他のタスクに関する演算処理を実行可能である。

【０１６３】

上記説明した具体例においては、演算装置１００は、例えば、タスクＡとタスクＢとの間に依存関係が存在しない場合、タスクＡの処理が終了する前に、一部の演算部においてタスクＢに関する演算処理を実行可能である。より具体的には、演算装置１００は、演算部１０６におけるタスクＡに関する演算処理が終了する前に、一部の演算部（例えば、演算部１０５）において、タスクＢに関する演算処理を実行可能である。この場合、演算装置１００は、図１８に例示するように、タスクＡに関する演算処理が完了する前に、タスクＢに関する演算処理を開始可能である。これより、本実施形態における演算装置１００は、全体の処理時間を短縮可能であり、また、各演算部がアイドル状態となる期間を短くすることができる。

【０１６４】

また、本実施形態における各タスク制御部（１１２乃至１１４）は、各演算部（１０４乃至１０６）においてタスクに関する演算処理が実行されるか否かを制御する。即ち、各

タスク制御部が詳細なタスク制御を実行されることから、上位の制御系（例えば、演算制御部１０１等）におけるタスク制御に関する負荷が増大しない。

【０１６５】

以上より、本実施形態における演算装置１００は、タスクに関する演算処理の効率を向上し、演算処理を高速化可能である。その理由は、各演算部（１０４乃至１０６）の実行状態（ステータス）と、各タスクの設定情報（各タスク設定テーブル（１１２ｂ乃至１１４ｂ）、及びタスクコマンド等）とに基づいて、演算部毎に、タスクの実行が制御されるからである。

【０１６６】

＜第２の実施形態＞

次に、本願発明の第２の実施形態について説明する。以下の説明においては、本実施形態に係る特徴的な構成を中心に説明する。上述した第１の実施形態と同様な構成に関する重複する説明は省略される。

【０１６７】

本実施形態は、上記第１の実施形態と、各タスク設定テーブル（１１２ｂ乃至１１４ｂ）に設定される情報の種類のみ異なり、その他は同様である。このため、本実施形態における演算装置１００の構成は、上記第１の実施形態における演算装置１００と同様としてよい。

【０１６８】

上記第１の実施形態においては、図２に例示するように、タスク設定テーブル（１１２ｂ乃至１１４ｂ）には、実行されるタスク毎に、各タスクに対する開始演算ノード番号２０１と、終了演算ノード番号２０２とが設定される。

【０１６９】

これに対して、本実施形態におけるタスク設定テーブル（１１２ｂ乃至１１４ｂ）には、図１０に例示するように、実行されるタスク毎に、各タスクに対する開始演算ノード番号２０１と、各タスクが使用する演算部の個数を表す使用演算ノード数１００２とが設定される。

【０１７０】

各タスク制御部は、特定のタスクに関して、上記開始演算ノード番号２０１と、使用演算ノード数１００２とに基づいて、当該タスクの終了演算部を特定する。この場合、各タスク制御部は、上記開始演算ノード番号２０１と、使用演算ノード数１００２とに基づいて、上記第１の実施形態における終了演算ノード番号を算出してもよい。

【０１７１】

本実施形態における各タスク制御部は、上記説明したステップＳ９０４において、上記開始演算ノード番号２０１と、使用演算ノード数１００２とに基づいて、当該タスクの終了演算部を特定する。その他の動作については、上記第１の実施形態と同様である。

【０１７２】

以上より、本実施形態における演算装置１００は、上記第１の実施形態における演算装置１００と同様の効果を奏する。また、本実施形態における演算装置１００は、終了演算部を直接指定する必要が無く、使用する演算部の個数のみを指定することにより、各演算部においてタスクに関する演算処理の実行を制御可能である。

【０１７３】

＜第３の実施形態＞

次に、本願発明の第３の実施形態について説明する。以下の説明においては、本実施形態に係る特徴的な部分を中心に説明する。上述した各実施形態と同様な構成についての重複する説明は省略される。

【０１７４】

以下、本実施形態について、図１１を参照して説明する。

【０１７５】

本実施形態は、上記説明した第１の実施形態の変形例である。上記第１の実施形態にお

ける演算装置１００と、本実施形態における演算装置１１００との差分は、データバッファ（図１１における１１０４、１１０５）と、タスク制御部（図１１における１１０１、１１０２、及び、１１０３）とが接続されない点である。

【０１７６】

上記第１の実施形態においては、タスク制御部（例えば、タスク制御部１１３）が、データバッファ（例えばデータバッファ１１０）を制御することにより、各演算部（例えば演算部１０５）におけるタスクの実行に必要なデータがデータバッファから出力される。または、タスク制御部（例えば、タスク制御部１１３）が、各演算部におけるタスクの実行に必要なデータが格納されたデータバッファの領域（アドレス等）を各演算部に通知することにより、各演算部が、係る領域からデータを取得する。

【０１７７】

これに対して、本実施形態における演算装置１１００は、タスク制御部（１１０１乃至１１０３）は、データバッファ（１１０４、１１０５）を直接的に制御しない点において、上記第１の実施形態における演算装置１００と異なる。

【０１７８】

本実施形態においては、各データバッファ（１１０４、１１０５）に接続された前段の演算部（１０４、１０５）から、特殊なコマンドがデータバッファに出力される。

【０１７９】

具体的には、データバッファ１１０４の前段の演算部１０４は、あるタスクに関する処理を終了した際に、特殊コマンド「Ｓｙｎｃ」をデータバッファ１１０４に出力（投入）する。同様に、データバッファ１１０５の前段の演算部１０５は、あるタスクに関する処理を終了した際に、特殊コマンド「Ｓｙｎｃ」をデータバッファ１１０５に出力（投入）する。

【０１８０】

各データバッファ（１１０４、１１０５）は、特殊コマンド（「Ｓｙｎｃ」）の投入より前に投入されたデータが全て読み出されるまで、特殊コマンドが投入された後に投入されたデータの読み出しを禁止する。より具体的には、本実施形態においては、図１２に例示するように、演算部１０５は、前段のデータバッファ１１０４から、特殊コマンド（「Ｓｙｎｃ」）が投入されるより前に投入されたデータを取得可能である。図１２には例示していないが、演算部１０６についても同様である。

【０１８１】

上記のように構成された本実施形態における演算装置１１００は、上記説明した第１の実施形態と同様の演算処理を実行可能である。その理由は、各データバッファ（１１０４，１１０５）が、特殊コマンドが投入された後に投入されたデータへのアクセスを禁止することにより、タスク間におけるデータへのアクセスを排他制御することが可能であるからである。

【０１８２】

なお、上記説明した特殊コマンドは「Ｓｙｎｃ」に限定されない。本実施形態における演算装置１１００は、「Ｓｙｎｃ」以外の任意の識別符号をコマンドとして採用してよい。

【０１８３】

＜第４の実施形態＞

次に、本願発明の第４の実施形態について説明する。以下の説明においては、本実施形態に係る特徴的な部分を中心に説明する。上述した各実施形態と同様な構成についての重複する説明は省略される。

【０１８４】

以下、図１３を参照して、本実施形態における演算装置１３００について説明する。

【０１８５】

演算制御部１３０１は、コマンドバッファ１３０２と接続されている。以下においてコマンドバッファ１３０２を、「ＦＩＦＯ１３０２」と称する場合がある。ＦＩＦＯ１３０

２は、演算部１３０７と接続されている。他の演算部（１３０８、１３０９）に対しては、演算制御部１３０１から制御信号（タスクコマンド）が直接的には入力されない。

【０１８６】

演算部（演算部１３０７、演算部１３０８、及び、演算部１３０９）は、あるタスクに関する演算処理を終了する際、当該タスクに関するタスクコマンドを、後段の演算部に受け渡す（転送する）。即ち、本実施形態においては、後段の演算部に対して、前段の演算部から、タスクコマンドがデータバッファを介して転送される。

【０１８７】

より具体的には、演算部１３０８に対しては、前段の演算部１３０７から、データバッファ１３０５を介して、タスクコマンドが受け渡される（転送される）。また、演算部１３０９に対しては、前段の演算部１３０８から、データバッファ１３０６を介して、タスクコマンドが受け渡される。本実施形態における演算部（１３０７乃至１３０９）のその他の構成は、上記各実施形態と同様としてよい。

【０１８８】

本実施形態におけるタスク制御部１３０３は、制御対象である演算部１３０７におけるステータスを確認して、後段の演算部１３０８に対してタスクを投入可能か否か判断する。同様に、本実施形態におけるタスク制御部１３０４は、制御対象である演算部１３０８におけるステータスを確認して、後段の演算部１３０９に対してタスクを投入可能か否か判断する。

【０１８９】

本実施形態においては、開始演算部が演算部１３０７に固定される。このため、各タスク制御部（１３０３、１３０４）が有するタスク設定テーブル（１３０３ｂ、１３０４ｂ）には、終了演算ノード番号、あるいは、使用演算ノード数のみが格納されてもよい。

【０１９０】

本実施形態におけるデータバッファ（１３０５、１３０６）は、上記各実施形態におけるデータバッファと同様としてもよい。

【０１９１】

以下、上記のように構成された本実施形態における演算装置１３００の動作について説明する。

【０１９２】

まず、第１のメモリ部１０２には、演算部（１３０７乃至１３０９）における演算処理の対象となるデータが格納されていることを想定する。

【０１９３】

説明の便宜上、実行されるタスクは２つ（タスクＡ、タスクＢ）とする。係る２つのタスクの内、第１のタスク（タスクＡ）は、演算部１３０７、演算部１３０８、及び、演算部１３０９において実行される。第２のタスク（タスクＢ）は、演算部１３０７、演算部１３０８、及び、演算部１３０９において実行される。

【０１９４】

タスクＡ、及び、タスクＢは、任意の演算処理を実行してよい。タスクＡに関するコマンドをタスクコマンドＡ、タスクＢに関するコマンドをタスクコマンドＢとする。タスクコマンドＡ、タスクコマンドＢは、上記第１の実施形態と同様、タスク識別符号３０１と、前タスク依存情報３０２と、後タスク依存情報３０３とを含む。

【０１９５】

各タスク制御部（１３０３、１３０４）におけるタスク設定テーブル（１３０３ｂ、１３０４ｂ）には、タスクＡ、及び、タスクＢに対する終了演算ノード番号（図２に例示する符号２０２）として「＃２」が格納されている。終了演算ノード番号の「＃２」は、演算部１３０９を表す。

【０１９６】

上記各実施形態と同様、各演算部（１３０７乃至１３０９）のステータスとして、待機中（「０」）、処理実行中（「１」）、処理終了（「２」）のいずれかが、各タスク制御

部（１３０３、１３０４）に入力される。
【０１９７】
以下、各演算部（１３０７乃至１３０９）におけるタスクＡ、及び、タスクＢの具体的な処理について、図１４及び図１５に例示するフローチャートを参照して説明する。
【０１９８】
図１４に例示するフローチャートには、図７に例示するフローチャートに対して、本実施形態におけるタスク制御部（１３０３、１３０４）に特有の処理が追加されている。上記第１の実施形態における演算装置１００と同様の処理については、上記第１の実施形態において説明したフローチャート（図７乃至図９）を適宜参照して説明する。
【０１９９】
まず、演算制御部１３０１が、タスクＡ、及び、タスクＢの実行を指示するコマンド（タスクコマンドＡ、タスクコマンドＢ）を生成する。そして、生成されたタスクコマンドが、ＦＩＦＯ１３０２、タスク制御部１３０３、及び、タスク制御部１３０４に入力される。
【０２００】
ＦＩＦＯ１３０２にタスクコマンドＡ、タスクコマンドＢが入力された場合、ＦＩＦＯ１３０２は、それらのタスクコマンドを蓄積する。
【０２０１】
タスクコマンドＡが入力された場合、タスク制御部（１３０３、１３０４）は、まず、上記第１の実施形態におけるタスク制御部（１１２、１１３、及び、１１４）と同様、投入されたタスク（タスクＡ）の実行可否を判定する。
【０２０２】
タスクコマンドＡが投入されると、タスク制御部（１３０３、１３０４）は、タスク設定テーブル（１３０３ｂ、１３０４ｂ）を参照する（ステップＳ７０１）。より具体的には、タスク制御部（１３０３、１３０４）は、タスク設定テーブル（１３０３ｂ、１３０４ｂ）から、終了演算ノード番号（この場合は「＃２」）を読み出す。本実施形態においては、上記したように、タスクＡ、及び、タスクＢに対する開始演算部は演算部１３０７である（開始演算ノード番号は「＃０」である）。
【０２０３】
そして、タスク制御部（１３０３、１３０４）は、制御対象である演算部がタスク（タスクＡ、タスクＢ）の処理に含まれるか確認する（ステップＳ７０２）。タスク制御部（１３０３、１３０４）は、タスク設定テーブル（１３０３ｂ、１３０４ｂ）を参照することにより、タスクＡ、及び、タスクＢに関する処理を実行する演算部を表す演算ノード番号を確認してもよい。
【０２０４】
この場合、タスクＡ、タスクＢ共に全ての演算部（１３０７乃至１３０９）を使用して処理されることから、ステップＳ７０３における確認結果はＹＥＳとなる。
【０２０５】
次に、タスク制御部（１３０３、１３０４）は、各演算部（１３０７乃至１３０９）のステータス（ステータス信号）を取得する。タスク制御部（１３０３、１３０４）は、取得したステータスを元に、各演算部（１３０７乃至１３０９）における演算処理の状態を確認する（ステップＳ７０４）。タスク制御部（１３０３、１３０４）は、上記確認した演算ノード番号に相当する演算部に対するステータス信号を確認する。
【０２０６】
次に、タスク制御部（１３０３、１３０４）は、取得したステータスに基づいて、制御対象である演算部において、タスク（タスクＡ）に関する演算処理を実行可能か否か判定する（ステップＳ７０５）。
【０２０７】
タスクコマンドＡが投入されたタイミングでは、演算部（１３０７乃至１３０９）のステータスは待機中（「０」）である。よって、これらの演算部は、タスクＡに関する演算

処理を実行可能である。

【０２０８】

以下、この場合における、タスク制御部１３０３の動作について説明する。

【０２０９】

開始演算ノード番号の値が「＃０」であることから、タスク制御部１３０３は、演算部１３０７においてタスクＡに関する演算処理を実行可能であると判定する（ステップＳ７０６においてＹＥＳ）。

【０２１０】

次に、タスク制御部１３０３は、演算部１３０７に対してタスク実行制御信号（タスクコマンド）及び、タスク実行に必要なデータを出力する（ステップＳ７０８）。より具体的には、タスク制御部１３０３は、ＦＩＦＯ１３０２を制御して、ＦＩＦＯ１３０２が保持するタスクコマンドＡを演算部１３０７に出力する。タスク制御部１３０３は、必要に応じて、第１のメモリ部１０２から、タスクＡの実行に必要なデータを読み込むよう、演算部１３０７を制御してもよい。ステップＳ７０８における処理の結果、演算部１３０７において、タスクＡに関する演算処理が実行される。

【０２１１】

演算部１３０７における、タスクＡに関する演算処理の結果は、データバッファ１３０５に保持される。データバッファ１３０５がほぼ演算部１３０７の演算結果で満たされた場合（空き容量が少なくなった場合）は、データバッファ１３０５は、「ＡｌｍｏｓｔＦｕｌｌ」の信号を演算部１３０７に出力する。この場合、演算部１３０７における演算処理はストールする。データバッファ１３０５が「ＡｌｍｏｓｔＦｕｌｌ」信号を出力する空き容量は、適宜設定可能である。

【０２１２】

以下、演算部１３０７における演算処理について図１５を参照して説明する。

【０２１３】

まず、ＦＩＦＯ１３０２からタスクコマンド（タスクコマンドＡ）が演算部１３０７に入力された際、演算部１３０７は、ステータスを実行中（「１」）に変更する（ステップＳ１５０１）。係るステータスの変更は、後述するステップＳ１５０３において実行してもよい。

【０２１４】

次に、演算部１３０７は、タスクコマンドにより指定されたタスク（タスクＡ）の実行に必要なデータを読み出す（ステップＳ１５０２）。この場合、演算部１３０７は、第１のメモリ部１０２から、タスクＡの実行に必要なデータを読み出してもよい。

【０２１５】

次に、演算部１３０７は、タスクコマンド（タスクコマンドＡ）により指定されたタスクに関する演算処理を実行する（ステップＳ１５０３）。

【０２１６】

タスクコマンド（タスクコマンドＡ）により指定されたタスクに関する演算処理が終了する際、演算部１３０７は、データバッファ１３０５に対して、タスクコマンドを発行する（ステップＳ１５０４）。この場合、演算部１３０７は、データバッファ１３０５に対して、タスクコマンドＡを出力してもよい。

【０２１７】

また、タスクに関する演算処理が終了する際、演算部１３０７は、ステータスを処理終了（「２」）に変更する（ステップＳ１５０５）。この場合、演算部１３０７は、係るステータスを、タスク制御部（１３０３、１３０４）に通知してもよい。

【０２１８】

次に、タスク制御部１３０３は、後段の演算部１３０８に対するタスクＡの投入を制御する（ステップＳ１４０１）。

【０２１９】

より具体的には、演算部１３０７のステータスが処理終了（「２」）となると、タスク

制御部１３０３は、後段の演算部１３０８に対して、タスクＡを投入可能であると判定する。

【０２２０】

この場合、タスク制御部１３０３は、タスクコマンドＡを演算部１３０８に出力するよう、データバッファ１３０５を制御する。なお、タスク制御部１３０３は、データバッファ１３０５からタスクコマンドＡ及び演算部１３０７における演算部結果を取得するよう、演算部１３０８を制御してもよい。

【０２２１】

データバッファ１３０５からタスクコマンドＡが演算部１３０８に投入されると、演算部１３０８において、タスクＡに関する演算処理が実行される。演算部１３０８における演算処理の結果は、データバッファ１３０６に格納される。データバッファ１３０６がほぼ演算部１３０８の演算結果で満たされた場合（空き容量が少なくなった場合）は、データバッファ１３０６は、「ＡｌｍｏｓｔＦｕｌｌ」の信号を演算部１３０８に出力する。この場合、演算部１３０８における演算処理はストールする。

【０２２２】

次に、タスク制御部１３０３は、ステップＳ７０９から処理を続行し、ステップＳ７０１乃至Ｓ７０５を実行する。そして、タスク制御部１３０３は、タスクＢが実行可能であるか否かを判定する（ステップＳ７０５）。

【０２２３】

仮に、タスクコマンドＡにおいて、タスクＡに対する後タスク依存情報（タスクＢ）が設定されており、タスクコマンドＢにおいて、タスクＢに対する前タスク依存情報（タスクＡ）が設定されている場合、タスクＡの演算結果がタスクＢにおいて使用される。

【０２２４】

この場合、タスク制御部１３０３は、依存関係があるタスク（タスクＢに対するタスクＡ）が終了していないことから、演算部１３０７に対するタスクＢの投入は不可であると判定する（ステップＳ７０６においてＮＯ）。即ち、この場合、演算部１３０７におけるタスクＢに関する演算処理の実行は、タスクＡの完了まで抑制される（ステップＳ７１０）。

【０２２５】

また、仮に、タスクコマンドＡに後タスク依存情報が設定されておらず、タスクコマンドＢに前タスク依存情報が設定されていない場合、タスクＡとタスクＢとは独立して処理可能である。

【０２２６】

この場合、タスク制御部１３０３は、上記第１の実施形態において説明したように、演算部１３０７に対してタスクＢを投入可能であると判定する（ステップＳ７０６においてＹＥＳ）。タスク制御部１３０３は、ステップＳ７０８から処理を続行し、演算部１３０７に対してタスクＢが投入される。

【０２２７】

タスク制御部１３０３は、ＦＩＦＯ１３０２を制御して、ＦＩＦＯ１３０２が保持するタスクコマンドＢを演算部１３０７に出力する。タスク制御部１３０３は、必要に応じて、第１のメモリ部１０２から、タスクＢの実行に必要なデータを読み込むよう、演算部１３０７を制御してもよい。ステップＳ７０８における処理の結果、演算部１３０７において、タスクＢに関する演算処理が実行される。この場合、演算部１３０７は、ステップＳ１５０１乃至ステップＳ１５０５を実行することにより、タスクＢに関する演算処理を実行する。

【０２２８】

演算部１３０７における演算処理の結果は、データバッファ１３０５に格納される。データバッファ１３０５がほぼ演算部１３０７の演算結果で満たされた場合（空き容量が少なくなった場合）は、データバッファ１３０５は、「ＡｌｍｏｓｔＦｕｌｌ」の信号を演算部１３０７に出力する。この場合、演算部１３０７における演算処理はストールする。

【０２２９】
再度、演算部１３０８におけるタスクＡに関する演算処理に戻って説明する。
【０２３０】
演算部１３０８は、まず、ステータスを実行中（「１」）に変更する（ステップＳ１５０１）。係るステータスの変更は、後述するステップＳ１５０３において実行してもよい。
【０２３１】
次に、演算部１３０８は、データバッファ１３０５から取得したタスクコマンドにより指定されたタスク（タスクＡ）の実行に必要なデータを読み出す（ステップＳ１５０２）。この場合、演算部１３０７は、第１のメモリ部１０２あるいはデータバッファ１３０５から、タスクＡの実行に必要なデータを読み出してもよい。
【０２３２】
次に、演算部１３０８は、タスクコマンド（タスクコマンドＡ）により指定されたタスクに関する演算処理を実行する（ステップＳ１５０３）。
【０２３３】
タスクコマンド（タスクコマンドＡ）により指定されたタスクに関する演算処理が終了する際、演算部１３０８は、データバッファ１３０６に対して、タスクコマンドを発行する（ステップＳ１５０４）。この場合、演算部１３０８は、データバッファ１３０６に対して、タスクコマンドＡを出力してもよい。
【０２３４】
また、タスクに関する演算処理が終了する際、演算部１３０８は、ステータスを処理終了（「２」）に変更する（ステップＳ１５０５）。この場合、演算部１３０８は、係るステータスを、タスク制御部（１３０３、１３０４）に通知してもよい。
【０２３５】
次に、タスク制御部１３０４は、後段の演算部１３０９に対するタスクＡの投入を制御する（ステップＳ１４０１）。
【０２３６】
より具体的には、演算部１３０８のステータスが処理終了（「２」）となると、タスク制御部１３０４は、後段の演算部１３０９に対して、タスクＡを投入可能であると判定する。
【０２３７】
タスク制御部１３０４は、タスクコマンドＡを演算部１３０９に出力するよう、データバッファ１３０６を制御する。なお、この場合、タスク制御部１３０４は、データバッファ１３０６からタスクコマンドＡ及び演算部１３０８における演算部結果を取得するよう、演算部１３０９を制御してもよい。
【０２３８】
データバッファ１３０６からタスクコマンドＡが演算部１３０９に投入されると、演算部１３０９において、タスクＡに関する演算処理が実行される。
【０２３９】
再度、このタイミングにおけるタスク制御部１３０３の処理に戻って説明する。
【０２４０】
上記説明したように、タスクＡとタスクＢとの間に依存関係が無い場合、このタイミングにおいて、演算部１３０７におけるタスクＢの処理が完了している可能性がある。この場合、演算部１３０７は、データバッファ１３０５にタスクコマンドＢを出力する。そして、演算部１３０７は、自身のステータスを処理終了（「２」）に変更する。
【０２４１】
この場合、タスク制御部１３０３は、ステップＳ１４０１において、後段の演算部１３０８に対するタスクＢの投入を制御する（ステップＳ１４０１）。
【０２４２】
より具体的には、演算部１３０７のステータスが処理終了（「２」）となると、タスク

制御部１３０３は、後段の演算部１３０８に対して、タスクＢを投入可能であると判定する。

【０２４３】

タスク制御部１３０３は、タスクコマンドＢを演算部１３０８に出力するよう、データバッファ１３０５を制御する。なお、この場合、タスク制御部１３０３は、データバッファ１３０５からタスクコマンドＢ及び演算部１３０７における演算部結果を取得するよう、演算部１３０８を制御してもよい。

【０２４４】

データバッファ１３０５からタスクコマンドＢが演算部１３０８に投入されると、演算部１３０８において、タスクＢに関する演算処理が実行される。演算部１３０８における演算結果は、データバッファ１３０６に格納される。データバッファ１３０６がほぼ演算部１３０８の演算結果で満たされた場合、データバッファ１３０６は、「ＡｌｍｏｓｔＦｕｌｌ」の信号を演算部１３０８に出力する。この場合、演算部１３０８はストールする。

【０２４５】

演算部１３０９におけるタスクＡの処理に戻って説明する。

【０２４６】

演算部１３０９は、タスクＡに関する演算処理を実行する（ステップＳ１５０１乃至Ｓ１５０５）。そして、演算部１３０９は、自身のステータスを処理終了（「２」）に変更する。

【０２４７】

ここで、タスクＡとタスクＢとの間に依存関係が存在する場合、タスク制御部１３０３は、ステップＳ７１０において、演算部１３０７におけるタスクＢの実行を抑制している。

【０２４８】

演算部１３０９のステータスが処理終了（「２」）になった場合、タスクＢと依存関係にあるタスクＡに関する処理が終了する。よって、この場合、タスク制御部１３０３は、演算部１３０７においてタスクＢに関する演算処理を実行可能と判定する（ステップＳ７０６においてＹＥＳ）。タスク制御部１３０３は、ステップＳ７０８から処理を続行し、演算部１３０７に対してタスクＢが投入される。

【０２４９】

この場合、タスク制御部１３０３は、ＦＩＦＯ１３０２を制御して、ＦＩＦＯ１３０２が保持するタスクコマンドＢを演算部１３０７に出力する。タスク制御部１３０３は、必要に応じて、第１のメモリ部１０２から、タスクＢの実行に必要なデータを読み込むよう、演算部１３０７を制御してもよい。ステップＳ７０８における処理の結果、演算部１３０７において、タスクＢに関する演算処理が実行される。この場合、演算部１３０７は、ステップＳ１５０１乃至ステップＳ１５０５を実行することにより、タスクＢに関する演算処理を実行する。以下、上記説明と同様にして、演算部１３０８、演算部１３０９においてタスクＢに関する演算処理が実行される。

【０２５０】

一方、タスクＡとタスクＢとの間に依存関係が存在しない場合、このタイミングにおいて、演算部１３０８におけるタスクＢの処理が完了している可能性がある。この場合、演算部１３０８は、データバッファ１３０６にタスクコマンドＢを出力する。そして、演算部１３０８は、自身のステータスを処理終了（「２」）に変更する。

【０２５１】

この場合、タスク制御部１３０４は、ステップＳ１４０１において、後段の演算部１３０９に対するタスクＢの投入を制御する（ステップＳ１４０１）。以下、演算部１３０９において、タスクＢの演算処理が実行される。

【０２５２】

以上のように構成された本実施形態における演算装置１３００は、タスクの実行を開始

する先頭の演算部が固定されるものの、より少ない数のタスク制御部（１３０３、１３０４）により、各演算部（１３０７乃至１３０９）を制御可能である。また、本実施形態における演算装置１３００は、コマンドバッファ（ＦＩＦＯ）１３０２の数も削減可能である。よって、このため、本実施形態における演算装置１３００は、よりシンプルな構成により、上記各実施形態における演算装置と同様の効果を奏することが可能である。

【０２５３】

＜第５の実施形態＞

次に、本願発明の第５の実施形態について、図１６、及び、図１７を参照して説明する。

【０２５４】

図１６に例示するように、本実施形態における演算装置１６００は、演算制御部１６０１と、複数の演算部（図１６における１６０２、１６０３、及び、１６０４）と、データバッファ（図１６における１６０８、１６０９）と、タスク制御部（図１６における１６１０、１６１１、及び、１６１２）と、を備える。本実施形態における演算装置１６００は、コマンドバッファ（図１６における１６０５、１６０６、及び、１６０７）を更に備えてもよい。以下、各構成要素について説明する。

【０２５５】

演算部（１６０２乃至１６０４）は、それぞれデータに対する特定の演算処理を実行する、演算処理装置である。演算部（１６０２乃至１６０４）において実行される係る演算処理は任意であり、例えば、ＦＦＴやフィルタリング等、各種信号処理に関する演算でもよい。演算部（１６０２乃至１６０４）は、上記第１の実施形態における演算部（１０４、１０５、及び、１０６）と同様としてもよい。

【０２５６】

データバッファ（１６０８、１６０９）は、複数の上記演算部の内、第１の演算部（前段の演算部）と、第２の演算部（後段の演算部）との間に接続される。データバッファ（１６０８、１６０９）は、上記第１の演算部から出力されるデータと、上記第２の演算部に入力されるデータとを保持可能である。データバッファ（１６０８、１６０９）は、例えば、上記第１の演算部における出力側と、第２の上記演算部における入力側との間に接続されてもよい。

【０２５７】

より具体的には、データバッファ１６０８は、演算部１６０２と、演算部１６０３との間に接続され、演算部１６０２からの出力データと、演算部１６０３に対する入力データとを保持する。この場合、例えば、演算部１６０２が第１の演算部、演算部１６０３が第２の演算部に相当すると考えられる。

【０２５８】

同様に、データバッファ１６０９は、演算部１６０３と、演算部１６０４との間に接続され、演算部１６０３からの出力データと、演算部１６０４に対する入力データとを保持する。この場合、例えば、演算部１６０３が第１の演算部、演算部１６０４が第２の演算部に相当すると考えられる。

【０２５９】

上記したように、あるデータバッファ（例えば、１６０８）に対する第２の演算部（例えば、演算部１６０３）が、他のデータバッファ（例えば、１６０９）に対する第１の演算部であってもよい。

【０２６０】

データバッファ（１６０８、１６０９）は、上記第１の実施形態におけるデータバッファ（１１０、１１１）と同様としてもよい。

【０２６１】

演算制御部１６０１は、１以上の上記第１の演算部、及び、１以上の上記第２の演算部の少なくともいずれかにおいて実行されるタスクの実行を指定する、タスクコマンドを生成する。なお、係るタスクは、上記データに対して実行される一連の演算処理である。

【０２６２】
演算制御部１６０１は、係るタスクコマンドを、後述する各コマンドバッファ（１６０５乃至１６０７）及び、各タスク制御部（１６１０乃至１６１２）に対して入力してもよい。演算制御部１６０１は、上記第１の実施形態における演算制御部１０１と同様としてもよい。
【０２６３】
コマンドバッファ（１６０５乃至１６０７）は、上記演算制御部１６０１から出力される上記タスクコマンドを、上記少なくともいずれかの演算部（１６０２乃至１６０４）に対して出力可能に保持する。
【０２６４】
本実施形態におけるコマンドバッファは、図１６に例示するように、上記各演算部（１６０２乃至１６０４）に対して通信可能に接続されてもよい。また、本実施形態におけるコマンドバッファは、図１７に例示するように、特定の上記演算部（例えば、１６０２）に対して接続されてもよい。
【０２６５】
コマンドバッファ（１６０５乃至１６０７）は、上記第１の実施形態におけるコマンドバッファ（１０７乃至１０９）と同様としてもよい。
【０２６６】
タスク制御部（１６１０乃至１６１２）は、上記タスクコマンドと、タスク設定情報と、各演算部（１６０２乃至１６０４）のステータス情報と、に基づいて、上記少なくともいずれかの演算部において実行される、上記タスクに関する演算処理を制御する。
【０２６７】
タスク設定情報は、あるタスクに関して、当該タスクが実行される上記演算部（１６０２乃至１６０４）を特定可能な情報である。ステータス情報は、各演算部（１６０２乃至１６０４）における演算処理の状態を表す情報である。タスク制御部１６１０は、複数の上記演算部（１６０２乃至１６０４）のそれぞれから、上記ステータス情報を取得する。
【０２６８】
タスク制御部（１６１０乃至１６１２）は、上記第１の実施形態におけるタスク制御部（１１２、１１３、及び、１１４）と同様としてもよい。
【０２６９】
以上のように構成された、本実施形態における演算装置１６００によれば、複数の演算部（１６０２乃至１６０４）が、データバッファ（１６０８乃至１６０９）を介してパイプライン状に接続された構成を有する演算装置１６００において、各演算部（１６０２乃至１６０４）が効率的に稼働するようにタスクの実行が制御される。即ち、演算装置１６００は、複数の演算部（１６０２乃至１６０４）を用いてパイプライン的に処理を実行するようなタスクを実行する場合、各演算部におけるアイドル状態やレイテンシを削減することが可能である。
【０２７０】
より具体的には、演算装置１６００は、あるタスクに関する特定の演算部における演算処理が終了する前に、他の一部の演算部において、他のタスクに関する演算処理を実行可能である。その理由は、各演算部（１６０２乃至１６０４）の実行状態（ステータス）と、各タスクの設定情報に基づいて、演算部毎に、タスクの実行が制御されるからである。
【０２７１】
図１６及び図１７に例示する構成は、本実施形態における演算装置を実現可能な一具体例であり、本実施形態はこれには限定されない。即ち、本実施形態における演算部の数は任意に定めてよい。また、演算部の数に応じて、データバッファ、タスク制御部の数も適宜選択されてよい。
【０２７２】
＜ハードウェア及びソフトウェア・プログラム（コンピュータ・プログラム）の構成＞
以下、上記説明した各実施形態を実現可能なハードウェア構成について説明する。以下

においては、特にタスク制御部（１１２乃至１１４、１１０１乃至１１０３、１３０３及び１３０４、並びに、１６１０乃至１６１２）の構成について説明する。以下、これらをまとめて「タスク制御部」と称する場合がある。

【０２７３】

上記各実施形態において説明した演算装置（１００、１１００、１３００、１６００）におけるタスク制御部は、専用のハードウェア装置により構成してもよい。その場合、上記各図に示した各部は、一部または全部を統合したハードウェア（処理ロジックを実装した集積回路等）として実現してもよい。

【０２７４】

また、上記タスク制御部は、図１９に例示するようなハードウェアと、係るハードウェアによって実行される各種ソフトウェア・プログラム（コンピュータ・プログラム）とによって構成してもよい。

【０２７５】

図１９における演算装置１９０１は、汎用のＣＰＵ（Ｃｅｎｔｒａｌ　Ｐｒｏｃｅｓｓｉｎｇ　Ｕｎｉｔ）やマイクロプロセッサ等の演算処理装置である。演算装置１９０１は、例えば後述する不揮発性記憶装置１９０３に記憶された各種ソフトウェア・プログラムを記憶装置１９０２に読み出し、係るソフトウェア・プログラムに従って処理を実行してもよい。

【０２７６】

記憶装置１９０２は、演算装置１９０１から参照可能な、ＲＡＭ（Ｒａｎｄｏｍ　Ａｃｃｅｓｓ　Ｍｅｍｏｒｙ）等のメモリ装置であり、ソフトウェア・プログラムや各種データ等を記憶する。記憶装置１９０２は、揮発性のメモリ装置であってもよい。

【０２７７】

不揮発性記憶装置１９０３は、例えば半導体記憶装置によるＲＯＭ（Ｒｅａｄ　Ｏｎｌｙ　Ｍｅｍｏｒｙ）や、フラッシュメモリのような、不揮発性の記憶装置であり、各種ソフトウェア・プログラムやデータ等を記録してもよい。

【０２７８】

外部記憶装置１９０４は、例えば、後述する記憶媒体１９０５に対するデータの読み込みや書き込みを処理する装置である。

【０２７９】

記憶媒体１９０５は、例えば光ディスク、光磁気ディスク、半導体フラッシュメモリ等、データを記録可能な任意の記録媒体である。

【０２８０】

入出力インタフェース１９０６は、タスク制御部と、上記各実施形態において説明した演算装置（１００、１１００、１３００、１６００）における他の構成要素との間の入出力を制御する装置である。係る入出力インタフェース１９０６は、各種通信ネットワークに接続可能な、ネットワークインタフェースであってもよい。

【０２８１】

例えば、上記各実施形態においては、タスク制御部と、上記各実施形態において説明した演算装置（１００、１１００、１３００、１６００）における他の構成要素との間は、入出力インタフェース１９０６を介して任意の通信バスにより接続されてもよい。

【０２８２】

上述した各実施形態を例に説明した本発明は、例えば、図１９に例示したハードウェア装置によりタスク制御部を構成し、係るタスク制御部に対して、上記各実施形態において説明した機能を実現可能なソフトウェア・プログラムを供給することにより実現してもよい。この場合、係るタスク制御部に対して供給したソフトウェア・プログラムを、演算装置１９０１が実行することによって、本願発明が達成されてもよい。

【０２８３】

上述した各実施形態において、上記各図に示した各部は、上述したハードウェアにより実行されるソフトウェア・プログラムの機能（処理）単位である、ソフトウェアモジュー

ルとして実現することができる。但し、これらの図面に示した各ソフトウェアモジュールの区分けは、説明の便宜上の構成であり、実装に際しては、様々な構成が想定され得る。

【０２８４】

例えば、図１、図１１、図１３、及び、図１６に例示したタスク制御部をソフトウェアモジュールとして実現する場合、これらのソフトウェアモジュールを不揮発性記憶装置１９０３に記憶しておき、演算装置１９０１がそれぞれの処理を実行する際に、これらのソフトウェアモジュールを記憶装置１９０２に読み出すよう構成してもよい。

【０２８５】

また、これらのソフトウェアモジュール間は、共有メモリやプロセス間通信等の適宜の方法により、相互に各種データを伝達できるように構成してもよい。このような構成により、これらのソフトウェアモジュール間は、相互に通信可能に接続可能である。

【０２８６】

更に、上記各ソフトウェア・プログラムを記憶媒体１９０５に記録しておき、上記演算装置（１００、１１００、１３００、１６００）の出荷段階、あるいは運用段階等において、適宜外部記憶装置１９０４を通じて当該ソフトウェア・プログラムが不揮発性記憶装置１９０３に格納されるよう構成してもよい。

【０２８７】

なお、上記の場合において、タスク制御部への各種ソフトウェア・プログラムの供給方法は、出荷前の製造段階、あるいは出荷後のメンテナンス段階等において、適当な治具を利用して当該演算装置（１００、１１００、１３００、１６００）内にインストールする方法を採用してもよい。また、各種ソフトウェア・プログラムの供給方法は、インターネット等の通信回線を介して外部からダウンロードする方法等のように、現在では一般的な手順を採用してもよい。

【０２８８】

そして、このような場合において、本発明は、係るソフトウェア・プログラムを構成するコード、あるいは係るコードが記録されたところの、コンピュータ読み取り可能な記憶媒体によって構成されると捉えることができる。

【０２８９】

以上、本発明を、上述した模範的な実施形態に適用した例として説明した。しかしながら、本発明の技術的範囲は、上述した各実施形態に記載した範囲には限定されない。当業者には、係る実施形態に対して多様な変更または改良を加えることが可能であることは明らかである。そのような場合、係る変更または改良を加えた新たな実施形態も、本発明の技術的範囲に含まれ得る。そしてこのことは、請求の範囲に記載した事項から明らかである。

【産業上の利用可能性】

【０２９０】

本発明は、例えば、複数の演算部を用いてパイプライン的に演算処理を実行するような演算装置に適用可能である。より具体的には、本発明は、無線信号処理や、画像処理等の信号処理用途に使用する演算装置に適用可能である。

この出願は、２０１４年６月１９日に出願された日本出願特願２０１４－１２５９４２を基礎とする優先権を主張し、その開示の全てをここに取り込む。

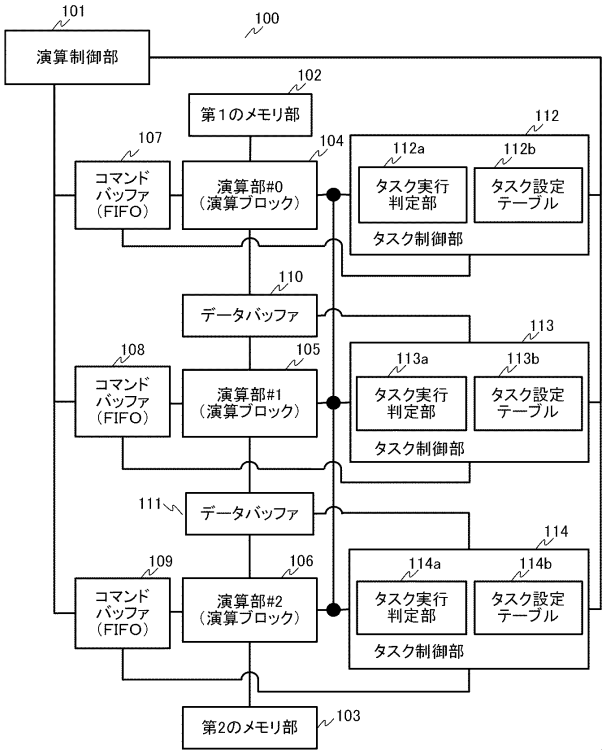
【符号の説明】

【０２９１】

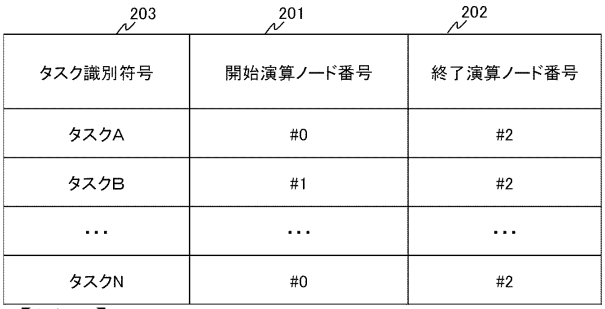
１００　演算装置
１０１　演算制御部
１０２　第１のメモリ部
１０３　第２のメモリ部
１０４、１０５、１０６　演算部
１０７、１０８、１０９　コマンドバッファ
１１０、１１１　データバッファ

１１２、１１３、１１４　タスク制御部
１１００　演算装置
１１０１、１１０２、１１０３　タスク制御部
１３０１　演算制御部
１３０２　コマンドバッファ
１３０３、１３０４　タスク制御部
１３０５、１３０６　データバッファ
１３０７、１３０８、１３０９　演算部
１６００　演算装置
１６０１　演算制御部
１６０２、１６０３、１６０４　演算部
１６０５、１６０６、１６０７　コマンドバッファ
１６０８、１６０９　データバッファ
１６１０、１６１１、１６１２　タスク制御部
１９０１　演算装置
１９０２　記憶装置
１９０３　不揮発性記憶装置
１９０４　外部記憶装置
１９０５　記憶媒体
１９０６　入出力インタフェース

【図１】

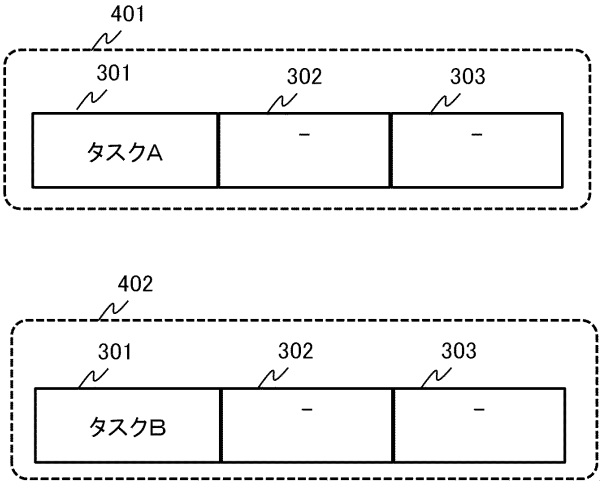


【図２】

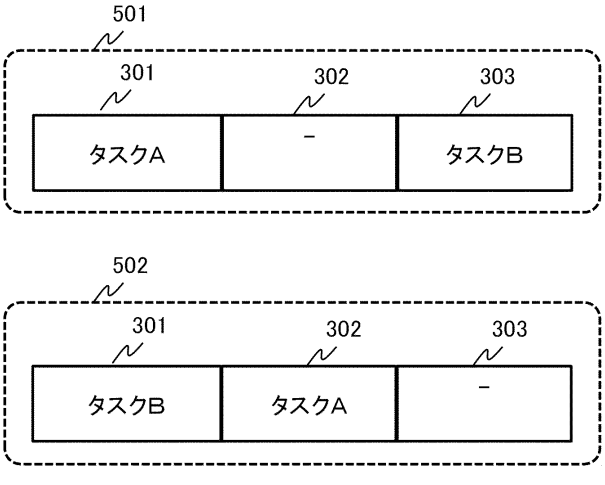


【図３】

【図４】



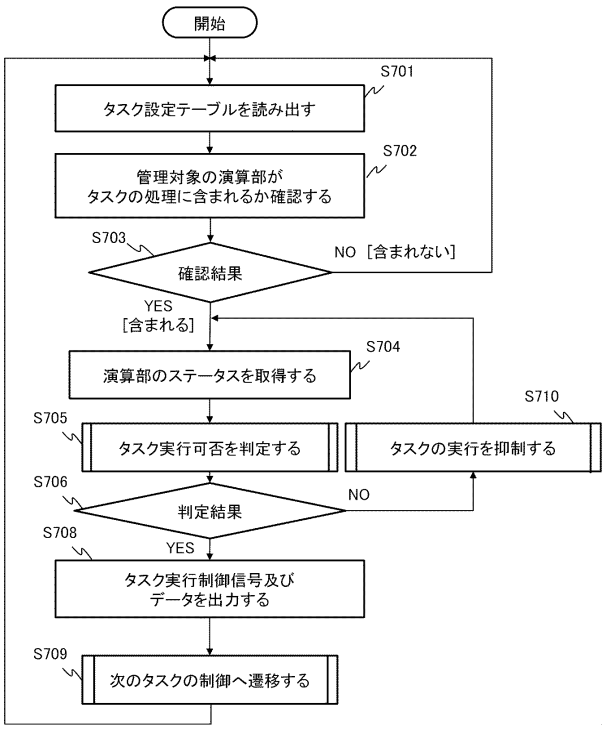
【図５】



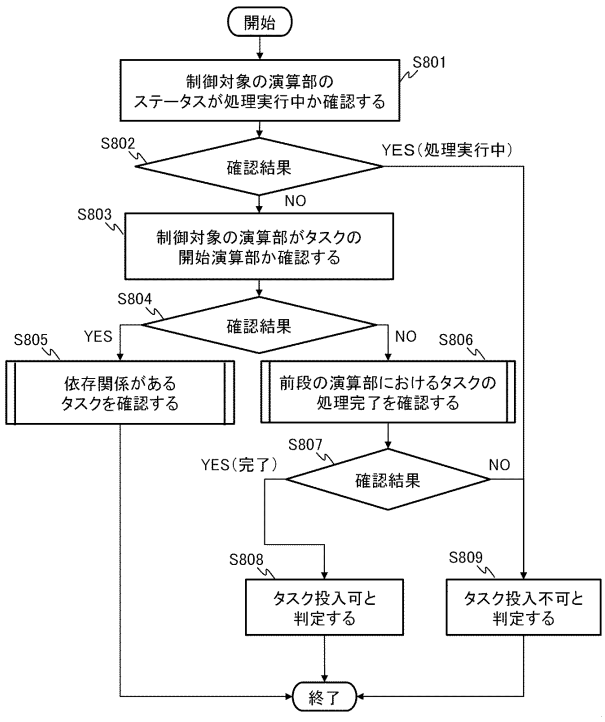
【図６】

203 タスク識別符号	201 開始演算ノード番号	202 終了演算ノード番号
タスクA	#0	#2
タスクB	#1	#2

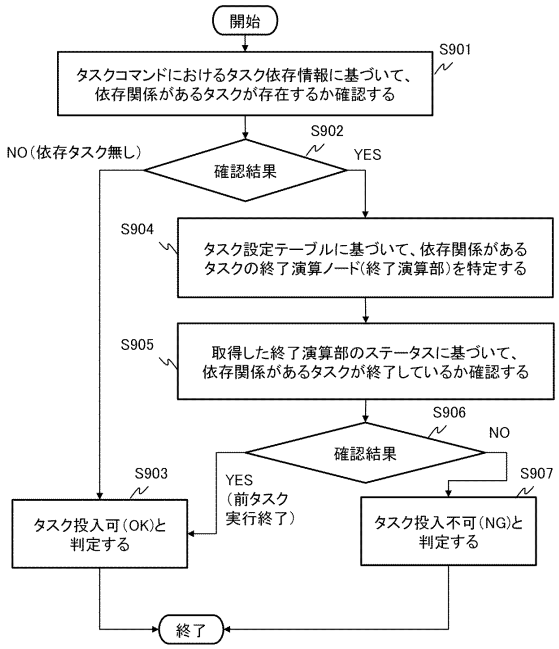
【図７】



【図８】



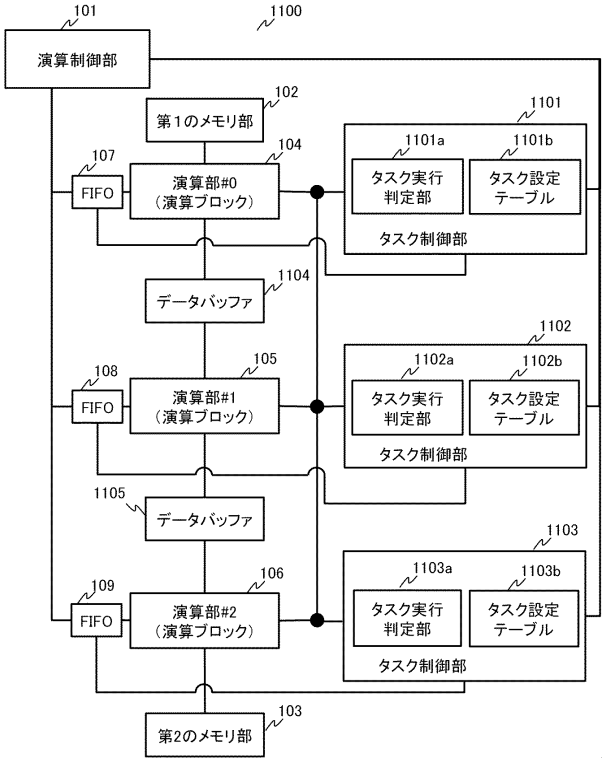
【図９】



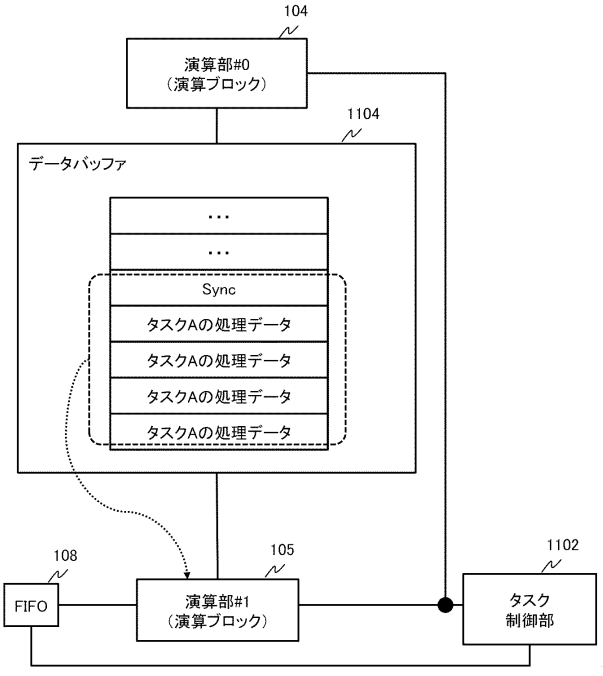
【図１０】

203 タスク識別符号	201 開始演算ノード番号	1002 使用演算ノード数
タスクA	#0	2
タスクB	#1	1
…	…	…
タスクN	#0	2

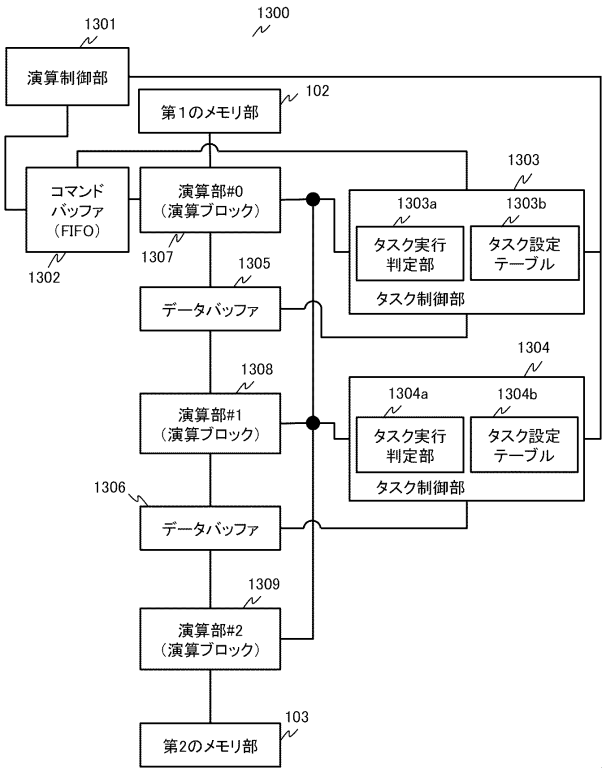
【図１１】



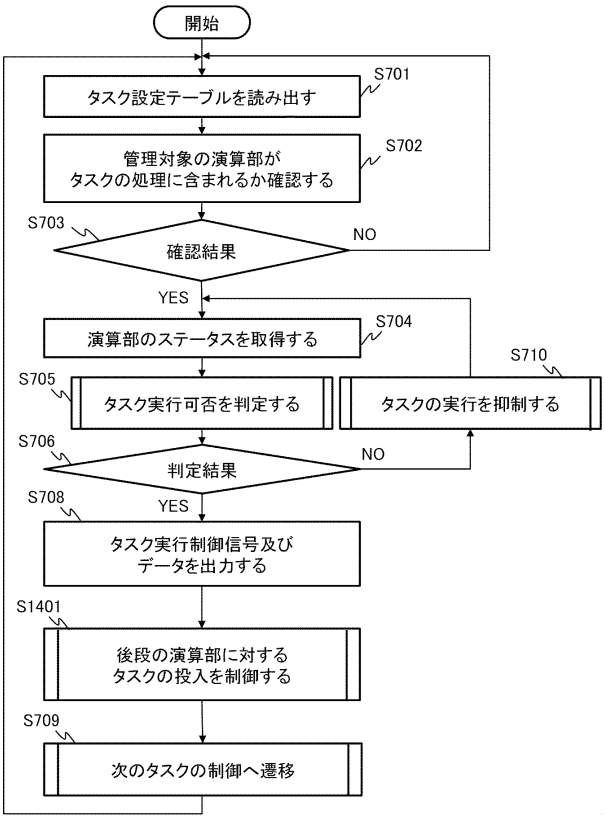
【図１２】



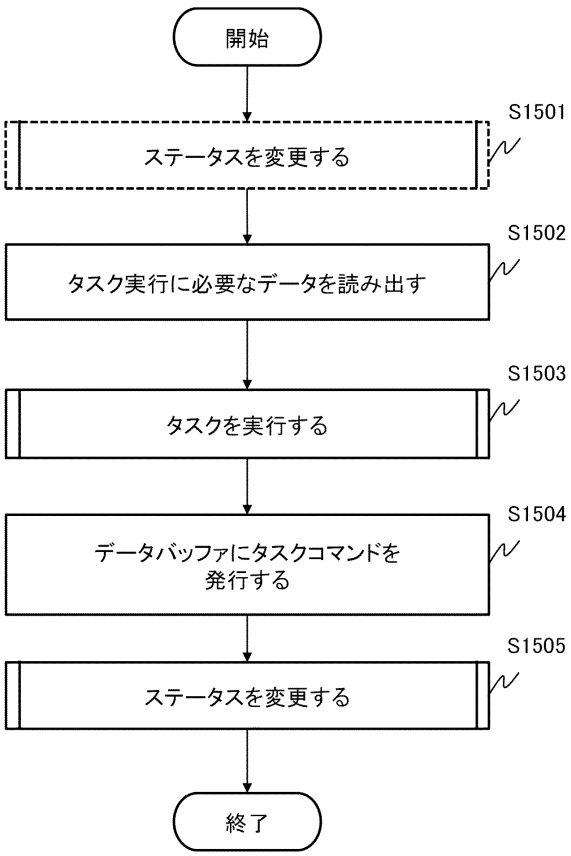
【図１３】



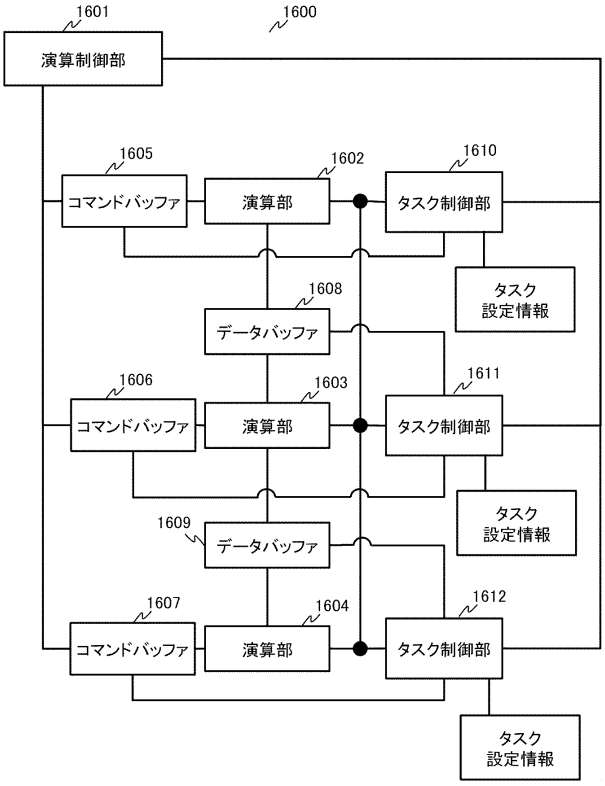
【図１４】



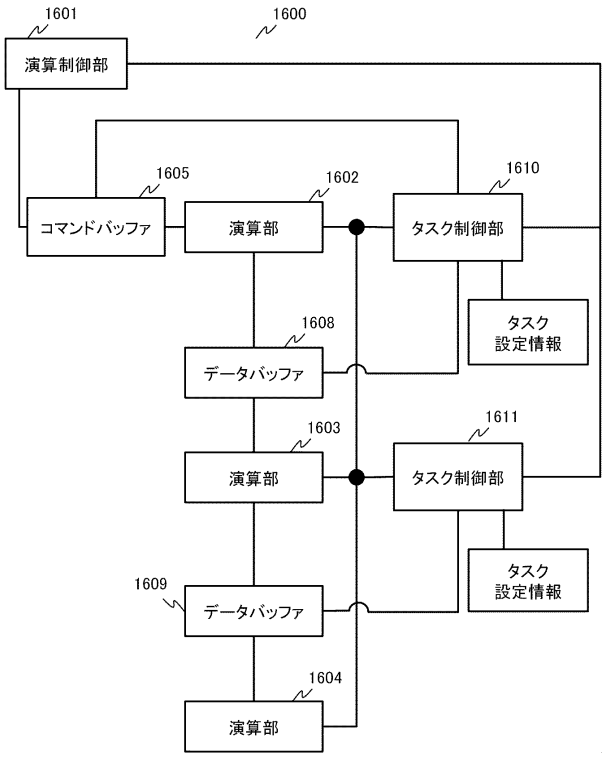
【図１５】



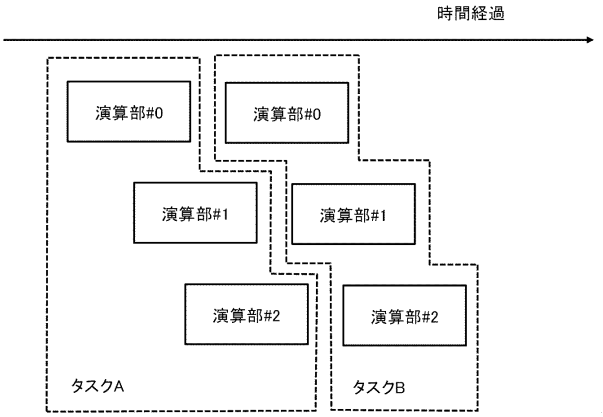
【図１６】



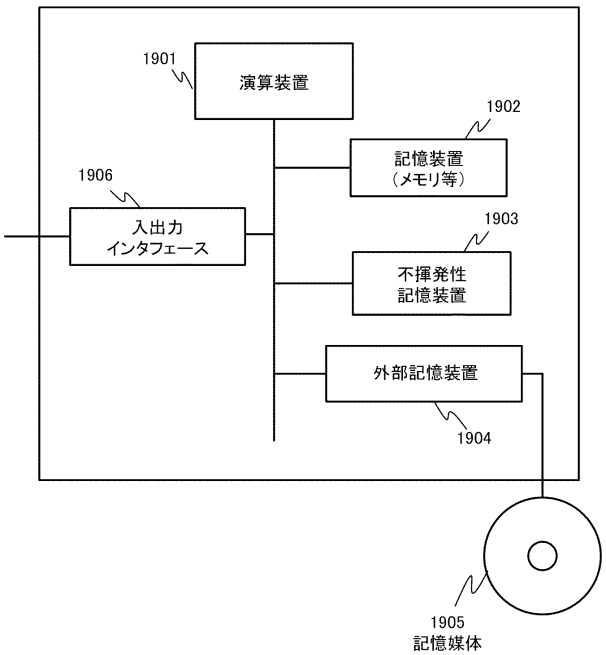
【図１７】



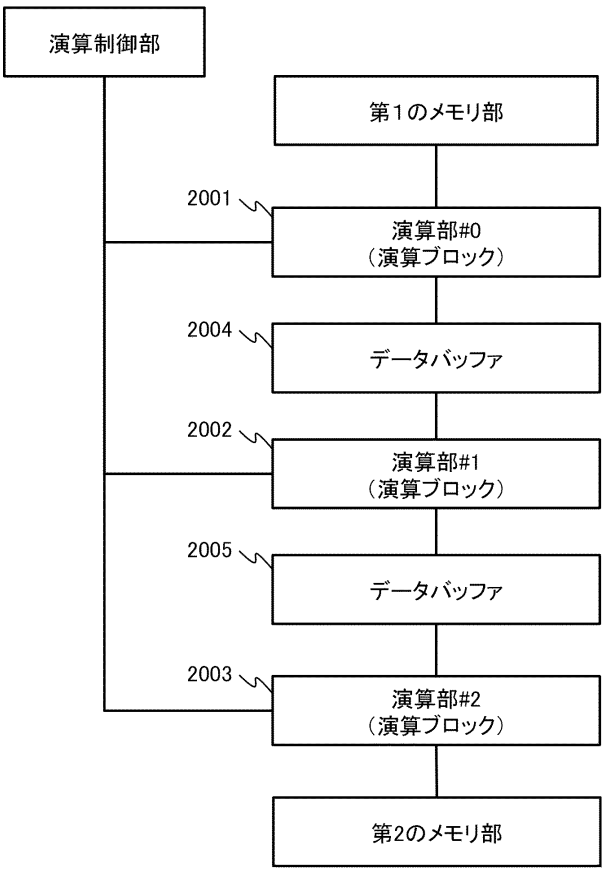
【図１８】



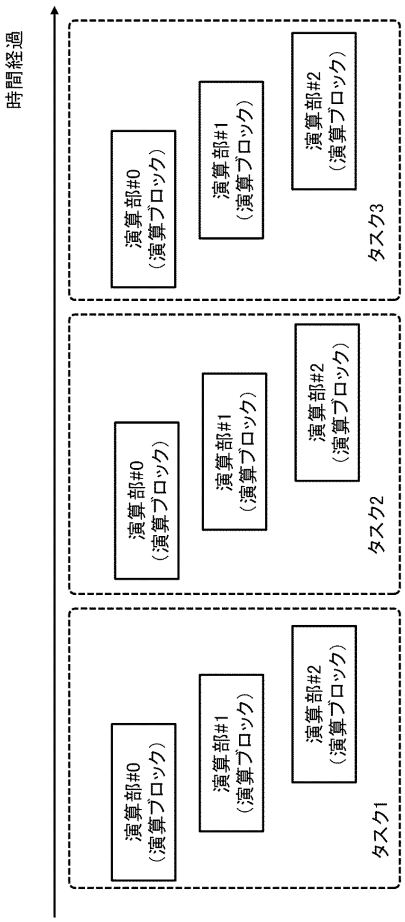
【図１９】



【図２０】



【図２１】


時間経過
演算部#0
（演算ブロック）
演算部#1
（演算ブロック）
演算部#2
（演算ブロック）
タスク1
演算部#0
（演算ブロック）
演算部#1
（演算ブロック）
演算部#2
（演算ブロック）
タスク2
演算部#0
（演算ブロック）
演算部#1
（演算ブロック）
演算部#2
（演算ブロック）
タスク3

フロントページの続き

(56)参考文献 特開２０１４－１０２６９２（ＪＰ，Ａ）
特開平１１－１９１０９６（ＪＰ，Ａ）
国際公開第２０１３／０３０６３０（WO，Ａ１）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ ９／３８
Ｇ０６Ｆ ９／４５５－ ９／５４
Ｇ０６Ｆ１５／１６ －１５／１７７
Ｇ０６Ｆ１５／８０
Ｇ０６Ｔ １／２０