



**【特許請求の範囲】****【請求項 1】**

マルチコアマイコンを有するコントローラモデルにおいて、少なくとも一つのパラメータを有する前記コントローラモデルの各コアをシミュレートするステップと、

少なくとも一つのパラメータを有し前記コントローラモデルに制御される装置モデルをシミュレートするステップと、

前記コントローラモデルの各コアおよび装置モデルのパラメータにアクセスするユーザインターフェイスをシミュレートするステップと、

トリガーイベントに応じて前記ユーザインターフェイスを介してコントローラモデルの各コアおよび装置モデルの実行を保留するステップと、

前記コントローラモデルの各コアのパラメータ又は装置モデルのパラメータの少なくとも一つの状態を、前記コントローラモデル及び装置モデルのパラメータを変更することなく、前記ユーザインターフェイスを介して前記保留ステップの間に決定するステップとからなることを特徴とするマルチコアマイコンシステムのシミュレーション方法。

10

**【請求項 2】**

請求項 1 に記載のマルチコアマイコンシステムのシミュレーション方法において、前記トリガーイベントは前記装置モデルの選択されたパラメータ条件からなることを特徴とするマルチコアマイコンシステムのシミュレーション方法。

**【請求項 3】**

請求項 1 に記載のマルチコアマイコンシステムのシミュレーション方法において、前記トリガーイベントは前記コントローラモデルの選択されたコアの選択された出力パラメータ条件からなることを特徴とするマルチコアマイコンシステムのシミュレーション方法。

20

**【請求項 4】**

請求項 1 に記載のマルチコアマイコンシステムのシミュレーション方法において、前記トリガーイベントは前記コントローラモデルの選択された内部パラメータ条件からなることを特徴とするマルチコアマイコンシステムのシミュレーション方法。

**【請求項 5】**

請求項 1 に記載のマルチコアマイコンシステムのシミュレーション方法において、前記トリガーイベントは前記ユーザインターフェイスにセットされた条件からなることを特徴とするマルチコアマイコンシステムのシミュレーション方法。

30

**【請求項 6】**

請求項 1 に記載のマルチコアマイコンシステムのシミュレーション方法において、さらに前記ユーザインターフェイスを介してコントローラモデルおよび装置モデルの各コアの実行速度を制御するステップを有することを特徴とするマルチコアマイコンシステムのシミュレーション方法。

**【請求項 7】**

請求項 1 に記載のマルチコアマイコンシステムのシミュレーション方法において、さらに前記コントローラモデルの各コアのパラメータ又は装置のモデルパラメータの少なくとも一つを表示するステップを有することを特徴とするマルチコアマイコンシステムのシミュレーション方法。

40

**【請求項 8】**

請求項 1 に記載のマルチコアマイコンシステムのシミュレーション方法において、さらに値を計算し、計算された値を表示装置に表示するステップを有することを特徴とするマルチコアマイコンシステムのシミュレーション方法。

**【請求項 9】**

請求項 8 に記載のマルチコアマイコンシステムのシミュレーション方法において、計算された前記の値は少なくとも一つの前記コアの負荷を含むことを特徴とするマルチコアマイコンシステムのシミュレーション方法。

**【請求項 10】**

請求項 8 に記載のマルチコアマイコンシステムのシミュレーション方法において、計算

50

された前記の値は前記コアのソフトウェアコンポーネントの負荷を含むことを特徴とするマルチコアマイコンシステムのシミュレーション方法。

【請求項 1 1】

請求項 8 に記載のマルチコアマイコンシステムのシミュレーション方法において、前記ソフトウェアコンポーネントは、アイドル動作に関するアイドルコンポーネントを含み、前記アイドルコンポーネントの負荷は新しい仕事を指定した場合に比較されることを特徴とするマルチコアマイコンシステムのシミュレーション方法。

【請求項 1 2】

マルチコアマイコンを有するコントローラモデルにおいて、少なくとも一つのパラメータを有する前記コントローラモデルの各コアをシミュレートする手段と、少なくとも一つのパラメータを有し前記コントローラモデルに制御される装置モデルをシミュレートする手段と、前記コントローラモデル及び前記装置モデルのパラメータにアクセスするユーザインターフェイスをシミュレートする手段と、トリガーイベントに応じて前記ユーザインターフェイスを介してコントローラモデルの各コアおよび装置モデルの実行を保留する手段と、前記コントローラモデルの各コアのパラメータ又は装置モデルのパラメータの少なくとも一つの状態を、前記コントローラモデル及び装置モデルのパラメータを変更することなく前記ユーザインターフェイスを介して前記保留ステップの間に決定する手段とを有することを特徴とするマルチコアマイコンシステムのシミュレーション装置。

10

【請求項 1 3】

請求項 12 に記載のマルチコアマイコンシステムのシミュレーション装置において、前記トリガーイベントは前記装置モデルの選択されたパラメータ条件からなることを特徴とするマルチコアマイコンシステムのシミュレーション装置。

20

【請求項 1 4】

請求項 1 2 に記載のマルチコアマイコンシステムのシミュレーション装置において、前記トリガーイベントは前記コントローラモデルの選択された出力パラメータ条件からなることを特徴とするマルチコアマイコンシステムのシミュレーション装置。

【請求項 1 5】

請求項 1 2 に記載のマルチコアマイコンシステムのシミュレーション装置において、前記トリガーイベントは前記コントローラモデルの選択された内部パラメータ条件からなることを特徴とするマルチコアマイコンシステムのシミュレーション装置。

30

【請求項 1 6】

請求項 1 2 に記載のマルチコアマイコンシステムのシミュレーション装置において、さらに前記ユーザインターフェイスにおける前記トリガーイベントをセットする手段を有することを特徴とするマルチコアマイコンシステムのシミュレーション装置。

【請求項 1 7】

請求項 1 2 に記載のマルチコアマイコンシステムのシミュレーション装置において、さらに前記コントローラモデルの各コアのパラメータ又は装置モデルのパラメータの少なくとも一つを表示する手段を有することを特徴とするマルチコアマイコンシステムのシミュレーション装置。

【請求項 1 8】

請求項 1 2 に記載のマルチコアマイコンシステムのシミュレーション装置において、さらに値を計算する手段と、計算された値を表示装置に表示する手段を有することを特徴とするマルチコアマイコンシステムのシミュレーション装置。

40

【請求項 1 9】

請求項 1 8 に記載のマルチコアマイコンシステムのシミュレーション装置において、計算された前記の値は少なくとも一つの前記コアの負荷を含むことを特徴とするマルチコアマイコンシステムのシミュレーション装置。

【発明の詳細な説明】

【技術分野】

【0 0 0 1】

50

本発明はマルチコアマイコンシステムの開発方法および開発装置に関する。

【背景技術】

【0002】

多くのアプリケーションがマイクロコンピュータ（マイコン）を用いたコントローラを有しており、マイコンは装置と呼ばれるデバイスを制御するために用いられる。例えば、自動車産業においてはマイコンはエンジンスロットルの駆動制御に用いられる。この場合、エンジンスロットルが装置を構成する。

【0003】

そのようなマイコン制御システムの開発を容易にするために、従来用いられてきたのがシミュレーションプログラムや開発プログラムであり、これは制御システムの動作をシミュレートする。これらの既知のプログラムは実際のコントローラを実際の装置で用いる。プログラム制御によりコントローラは制御信号を装置に送り、所定の動作を実行させる。

【0004】

マルチコアマイコンシステムの開発の一部として常に必要なのは、コントローラで実行するプログラムのデバッグと微調整である。これは従来コントローラのプログラムコードにブレークポイント（強制実行停止コード）を設けて行ってきた。コントローラ内でブレークポイントに行き当たるとプログラムの実行が停止し、プログラマはコントローラモデルと装置モデルの種々のパラメータをチェックできる。これらパラメータは内部レジスタ値、入出力信号状態、インターラプト状態、パラメータ値等を含む。

【0005】

システムの所望の動作が得られればブレークポイントは一般に除去され、さらにコントローラ及び装置のデバッグと微調整が反復して実施される。例えば、コントローラプログラムの特定部分が微調整及びデバッグされブレークポイントが削除されたとき、全プログラムの所望の装置モデル動作が完成するようデバッグ及び微調整されるまで、ブレークポイントをコントローラプログラムの他の部分に挿入することが必要となる。

【0006】

上記のコントローラ及び装置を開発する従来方法の主な欠点は、少なくとも一つのブレークポイントのコントローラプログラムへの挿入は、コントローラのプログラミングの変更を要するため障害になるということである。すなわち、そのようなコントローラプログラミングの変更は、一方で所望せず予期しない動作変更をコントローラや装置に引き起こすことがある。そのような所望せず予期しない動作変更は、追加のデバッグ及び微調整をマイコン制御システム全体に要求する。

【0007】

従来の装置制御用マイコンでは、一般的に単一のコア（CPU）を備えていた。しかし最近のマイコンは複数のコアを備えている。例えば、デュアルコアマイコンは2つの独立したコアを有し、各々が独立かつ同時に各々のコンピュータプログラムを実行する。マルチコアマイコンはシングルコアマイコンに比べ動作速度が優れており、そのプログラミング及びデバッグは新しい課題を提示している。例えば、特許文献1には、プロセッサにマルチコアを用いたマルチコアモデルのシミュレーション方法が開示され、コアモデル処理とコアモデル間の通信処理を並列に行う例が開示されている。

【0008】

例えば、適切なマルチコアマイコンシステムのプログラミングとデバッグには、所定時間内に実行される命令の数がほぼ同じであることが非常に望ましい。そうでなければ、マルチコアマイコンシステムの全体的な効率が損なわれる。

【0009】

【特許文献1】特開2007-122602号公報

【発明の開示】

【発明が解決しようとする課題】

【0010】

従来知られていたマルチコアマイコンシステムのデバッグを正確に行う開発装置は、マ

10

20

30

40

50

ルチコアマイコンシステムの実行を正確に反映していた。その場合、マルチコアマイコンシステムの開発は、シングルコアマイコン制御システム用の開発システムをマルチコアマイコンの各々のコアに適用していた。しかしながら、これはマイコンの各コア間に存在する相互作用を考慮しておらず、マルチコアマイコンシステムのデバッグに不正確さをもたらした。

【課題を解決するための手段】

【0011】

本発明は、マルチコアマイコンシステムの開発方法及び開発装置であって、上述した従来方法及び装置の欠点を解決するものを提供する。

【0012】

要するに、本発明の方法は、少なくとも一つのパラメータを持つコントローラモデルと少なくとも一つのパラメータを持ち前記コントローラモデルで制御される装置モデルをシミュレートする。これらのパラメータは、コントローラモデルのマルチコアのレジスタ値を含み、コントローラモデル及び装置モデルの入出力状態を含み、各コアの内部パラメータ値とシステムの全状態を示す他のファクタを含む。

【0013】

コントローラモデルと装置モデルのコアパラメータにアクセスするユーザインターフェイスが用いられる。シミュレーションの間、ユーザインターフェイスは、トリガーイベントに応じてコントローラモデルと装置モデルの実行を保留する。トリガーイベントは、装置モデルまたはコントローラモデルの条件、割り当て時間でもよく、ユーザインターフェイス経由で与えられても良い。

【0014】

コントローラモデルと装置モデルの実行を保留する間、ユーザインターフェイスは、コントローラモデルパラメータまたは装置モデルパラメータのいずれも変更することなく、コントローラモデルのマルチコアのパラメータ及び装置モデルパラメータを決定する。

【0015】

コントローラモデルと装置モデルのプログラム実行の保留は目立たずに行われ、言い換えればブレークポイントのコントローラモデルプログラムへの挿入によるコントローラプログラムの変更がないため、そのようなブレークポイントの導入により生じうる所望せず予期しないエラーの導入が回避できる。

【0016】

コントローラモデルのマルチコアおよび装置モデルのための多種パラメータの値は、ユーザインターフェイスによりメモリに記憶され、選択的にディスプレイに表示される。多種パラメータの値に加えて、他の値、例えばマルチコアマイコンシステムの各コアの負荷値も同様に保存され、表示される。

【0017】

このように、マルチコアマイコンの各コアのプログラムは、マイコンの各コアの負荷をバランスさせ、各コアの負荷がほぼ同じになるように実行される。

【発明の効果】

【0018】

本発明は上記のように、一旦トリガーイベントが生じたとき、ブレークポイントをコントローラモデルのコア用プログラムコードへ挿入せずにシミュレーションの保留または停止が発生し、コントローラモデルまたは装置モデルのいかなるパラメータ値への影響も与えない。さらにコントローラモデルのマルチコアのコンピュータコードデバック量を最小にする。

【0019】

さらに、各コアへの種々のソフトウェアの負荷などの種々の計算値もまたプログラマに表示され、プログラマがコントローラの最適効率のためにコアプログラミングを調整することを可能にする。

【発明を実施するための最良の形態】

## 【 0 0 2 0 】

図 1 は、本発明の好適な実施例における、マルチコアマイコンシステムの開発装置 1 0 のブロック図を示す。マルチコアマイコンから構成されるコントローラモデル 1 2 はプログラム制御下で作動し、少なくとも一つの制御ライン 1 6 により装置モデル 1 4 の動作を制御する。装置モデル 1 4 はマイコン制御のどの対象でもよく、自動車産業における装置モデルは電子制御スロットル弁でもよい。その装置モデルにおいて、コントローラモデル 1 2 は動作制御すなわちスロットル弁の開閉制御を行う。

## 【 0 0 2 1 】

図 2 は、コントローラモデル 1 2 のブロック図を示す。コントローラモデル 1 2 は少なくとも二つ乃至 N 個のコア 1 8 を有し、各コア 1 8 は CPU を構成する。各コア 1 8 はさらにバス 2 0 を介して少なくとも一つの周辺モジュール 2 2 と通信する。バス 2 0 はインターラプトコントローラ 2 4 と通信しても良く、入出力ポート 2 6 と通信しても良い。

## 【 0 0 2 2 】

マルチコア 1 8 を有するマルチコアマイコンシステムからなるコントローラモデル 1 2 において、各コア 1 8 は独立に各々のコンピュータプログラムを他のコア 1 8 と同時に実行する。全てのコアは共通データバス 2 0 を共有しており、二つ以上のコア 1 8 が同時にデータバス 2 0 にアクセス要求した場合は、データバス 2 0 が再びアイドル状態のコア 1 8 からデータを受ける自由状態に戻るまで一つまたはそれ以上のコア 1 8 がアイドル状態に入る。

## 【 0 0 2 3 】

図 3 は、電子スロットル制御としての装置モデル 1 4 のブロック図を示す。そのような装置モデル 1 4 は、ライン 4 4 に示すモータ電流とトルクのようなパラメータを持つシミュレートされたモータ 4 0 を有する。装置モデル 1 4 はまたシミュレートされたセンサ 4 6 を有し、スロットルプレートの角度を示すシミュレートされた出力 4 8 を有する。ライン 4 4、4 8 の値は装置モデル 1 4 のパラメータを構成し、これはコントローラモデル 1 2 の命令に応じて変化する。

## 【 0 0 2 4 】

図 1 において、プロセッサを有するユーザインターフェイス 3 0 はコントローラモデル 1 2 と入出力ライン 3 2 を介して通信する。同時に、ユーザインターフェイス 3 0 は入出力ライン 3 4 を介して装置モデル 1 4 と通信する。ユーザインターフェイス 3 0 はまた、ビデオモニタ、プリンタ、データ記憶装置等の出力装置 3 6 と通信し、システムオペレータがコントローラ 1 2 と装置モデル 1 4 の間の全システムシミュレーションの結果を分析することを可能とする。同時に、マウス、キーボード等の入力装置 3 8 もまた、ユーザインターフェイス 3 0 と通信し、システムオペレータに全システムシミュレーションの実行を制御させる。

## 【 0 0 2 5 】

ユーザインターフェイス 3 0 とコントローラ 1 2、装置モデル 1 4 を結ぶ通信ライン 3 2、3 4 は、ユーザインターフェイス 3 0 を介してシステムオペレータにコントローラ 1 2、装置モデル 1 4 の種々のパラメータへのアクセスを可能とする。加えて、通信ライン 3 2、3 4 はユーザインターフェイス 3 0 に、コントローラモデル 1 2、装置モデル 1 4 のどのパラメータ値も変えずに、コントローラモデル 1 2 と装置モデル 1 4 の実行を保留させることができる。

## 【 0 0 2 6 】

さらに、シミュレートされたコントローラモデル 1 2 と装置モデル 1 4 の実行保留の間、システムオペレータは図 2 に示すコントローラモデル 1 2 内の各コア 1 8 の種々のパラメータ、または図 3 に示す装置モデル 1 4 の種々のパラメータを、これらパラメータの値を変えずに取り出すことができる。従って、各コア 1 8 のためのコントローラプログラムシミュレーションの実行再開において、コントローラモデル 1 2 は、コントローラモデル 1 2 と装置モデル 1 4 の実行保留によりシステムのシミュレーションに影響を与えることなく、実行を継続することができる。

10

20

30

40

50

## 【 0 0 2 7 】

図 1 のブロック図において、コントローラモデル 1 2 と装置モデル 1 4 を有するシミュレータおよびユーザインターフェイス 3 0 は、コントローラモデル 1 2 と装置モデル 1 4 およびユーザインターフェイス 3 0 用の個別のデータプロセッサ 1、2、3 を用いて実行される。

## 【 0 0 2 8 】

しかし、コントローラモデル 1 2 と装置モデル 1 4 およびユーザインターフェイス 3 0 はかならずしも個別のプロセッサを用いる必要は無い。代わりに、図 4 に示すように、単一のデータプロセッサのコンピュータシステムが、コントローラモデル 1 2 や装置モデル 1 4、インターフェイス 3 0 の実行のために用いられても良い。

10

## 【 0 0 2 9 】

図 5 A ~ D において、全システムのシミュレーション出力の例が図示され、図 1 に示すビデオディスプレイ等の出力装置 3 6 に表示される。例えば図 5 A においてテーブル 1 0 0 はコントローラモデル 1 2 の各コアのために表示され、ここで各テーブル 1 0 0 はコントローラの特定コアに関連する種々のパラメータと算出値を含む。

## 【 0 0 3 0 】

上記テーブル 1 0 0 は、コア 1 8 の一つで実行される実行コードを示す T A S K、I D L E、O S 等のカテゴリー名を有する。テーブル 1 0 0 はまた、各カテゴリーのコール回数や、命令数、消費されたサイクル数、負荷率を含む。例えばコントローラ 1 2 の第 1 コア ( C P U # 1 ) に示すように、カテゴリー「タスク」は第 1 コア時間の 5 5 % を消費し、「I D L E」には 5 0 0 0 コールがあり第 1 コア時間の 3 5 % を消費する等である。

20

## 【 0 0 3 1 】

図 5 B ~ 5 D は、出力デバイス 3 6 の全マルチコアマイコンシステムのシミュレーションの他の例示出力を示す。例えば、図 5 B はコントローラモデル 1 2 または装置モデル 1 4 のコアの一つにおけるパラメータ値を示すグラフである。図 5 C はコントローラ 1 2 のコアの一つのレジスタ値テーブルを示す。図 5 D はコントローラ 1 2 のコア 1 8 で現在実行中のアセンブリコードテーブルを示す。

## 【 0 0 3 2 】

例えば、図 5 D で、アドレス 0 0 0 0 0 0 はコード 0 F 9 A をストアし、レジスタ R 0、R 1 を図 5 C のテーブル中で A D D する。アドレス 0 0 0 0 0 2 はコード C 0 0 3 をストアし、レジスタ R 1 の値をテーブル中でメモリの R 3 に移動する。アドレス 0 0 0 0 0 4 はコード 2 8 D B をストアし、テーブル中でレジスタ R 2 と R 3 の値を掛ける。アドレス 0 0 0 0 0 6 はコード F 3 5 E をストアし、テーブル中でレジスタの R 4 にジャンプする。

30

## 【 0 0 3 3 】

テーブル 1 0 0 の負荷率は計算値であり、一方テーブル 1 0 0 の他の情報はコントローラモデルの種々のパラメータに関係する。また、他の異なるテーブル 1 0 0 がコントローラ 1 2 の各コアについて生成され、この複数のテーブル 1 0 0 は必要に応じ同時に表示される。他のタイプの表示たとえばグラフもインターフェイス 3 0 で表示可能である。

## 【 0 0 3 4 】

図 6 に示すフローチャートには、コントローラ 1 2 の各コア 1 8 のテーブル 1 0 0 の生成に関する本実施例の動作が記載される。プログラムはステップ 1 1 0 でスタートし、すぐにステップ 1 1 2 に進む。ステップ 1 1 2 ではプログラムは、コアがファンクションコール命令を実行中か否かを決定する。もし実行していない場合はステップ 1 1 2 はステップ 1 1 4 に分岐する。

40

## 【 0 0 3 5 】

もしステップ 1 1 2 でコアがファンクションコール命令を実行している場合は、ステップ 1 1 2 はステップ 1 1 6 に進みファンクションコール I D またはアドレスが決定される。次いでステップ 1 1 6 はステップ 1 1 4 に進む。

## 【 0 0 3 6 】

50

ステップ 1 1 4 において、インターフェイス 3 0 はコントローラモデルの一つのコア 1 8 のサイクル数と命令数を計算し、ステップ 1 1 5 に進む。

【 0 0 3 7 】

ステップ 1 1 5 で、インターフェイス 3 0 は、コントローラモデルのコア 1 8 で処理される各カテゴリーの命令および機能名の負荷率を計算する。ステップ 1 1 5 は次いでステップ 1 1 8 に進み、値が例えば表示装置に表示されまたはメモリに保存される。ステップ 1 1 8 は次いでステップ 1 2 0 に進み、コントローラモデルのコア 1 8 のための次の命令がステップ 1 1 2 に戻って処理され、上記ステップが次の命令のために反復される。

【 0 0 3 8 】

図 6 のフローチャートは、当然ながらコントローラモデル 1 2 の各コア 1 8 につき反復される。この方法で図 5 に示すテーブル 1 0 0 がコントローラモデル 1 2 の各コア 1 8 につき生成され表示装置に表示されても良い。

【 0 0 3 9 】

これはプログラマがコントローラモデル 1 2 の種々のコア 1 8 に対しプログラミングを調整し、これにより全てのコア 1 8 への負荷率をほぼ同一にすることを可能にする。この方法によりコントローラモデル 1 2 の種々のコア 1 8 の計算能力は、コントローラモデルの効率的動作とほぼ同一となる。

【 0 0 4 0 】

図 7 において、図 6 のフローチャートの変形例が示されている。図 7 のフローチャートは、シミュレーションの最後におけるコントローラモデル 1 2 の全てのコア 1 8 の種々のファクタとパラメータの総計を示す。

【 0 0 4 1 】

更に詳細に言えば、プログラムはステップ 1 1 5 の負荷率計算の後でステップ 1 4 0 に進み、命令のカテゴリー名を決定する。カテゴリー名を決定するためにいずれかの公知の手段、例えばルックアップテーブルが用いられても良い。

【 0 0 4 2 】

ステップ 1 4 2 で、プログラムはシミュレーションが終了したか決定する。もし終了していなければ、ステップ 1 4 2 はステップ 1 1 2 に戻り、ステップ 1 1 2 で開始された処理が各コア 1 8 の次の命令のために反復される。

【 0 0 4 3 】

反対に、シミュレーションが終了したと仮定した場合、ステップ 1 4 2 は代わりにステップ 1 4 4 に進み、コントローラモデル 1 2 の各コア 1 8 の負荷率を含む種々の演算カテゴリーの全体の値が決定される。ステップ 1 4 4 は次いでステップ 1 1 8 に進み、それらの値を表示し、ステップ 1 4 6 に進み、シミュレーションが終了する。

【 0 0 4 4 】

動作中に、シミュレーションはユーザインターフェイス 3 0 の制御の下に実行される。シミュレーションの間、シミュレーション速度はユーザインターフェイス 3 0 により例えばクロック信号速度を変更する等で制御される。

【 0 0 4 5 】

シミュレーションはトリガーイベントが検出されるまで継続する。トリガーイベントはユーザによりユーザインターフェイス 3 0 を介して起動され、またはコア 1 8 の一つの特定期スタ値が所定条件になったとき等に、ユーザインターフェイス 3 0 により起動される。

【 0 0 4 6 】

トリガーイベントが発生するいずれのイベントでも、インターフェイス 3 0 はコントローラモデル 1 2 および装置モデル 1 4 へのクロック信号を停止してシステムシミュレーションを保留する。このとき、図 6 及び図 7 に記載されたプログラムが実行され、コントローラモデル 1 2 の種々のコア 1 8 の種々のパラメータ及び装置モデル 1 4 のパラメータが表示装置に表示される。

【 0 0 4 7 】

10

20

30

40

50



言うまでもなくコントローラ及び装置モデルシミュレーション動作の保留に多数の異なるタイプのトリガーイベントを用いることができ、コントローラモデル及び装置モデルの種々のパラメータが表示装置に表示される。

【0048】

例えば、このトリガーイベントは、装置モデルまたはコントローラモデルの少なくとも一つのレジスタの所定値とすることができる。同様に、コントローラモデルのインターラプト状態もトリガーイベントを構成できる。同様に、装置モデルまたはコントローラの種々のパラメータがトリガーイベントを構成できる。

【0049】

本実施例の重要な長所は、一旦トリガーイベントが生じたとき、ブレークポイントをコントローラモデル12のコア18用プログラムコードへ挿入せずにシミュレーションの保留または停止が発生し、コントローラモデルまたは装置モデルのいかなるパラメータ値への影響も与えないことである。これは、同様にコントローラモデル12のコア18のためのコンピュータコードデバック量を最小にする。

【0050】

図8はマルチコアマイコンの各コア(CPU#1、#2、...#N)のテーブル100における表示を示す。マルチコアマイコンの一つのコアを選択して新しいタスクを処理させるには、各カテゴリー(TASK:タスク、ISR:インターラプトルーチン、OS:オペレーションシステム、IDLE:アイドル)の負荷率が各コアにつき計算される。

【0051】

優先度が低く通常優先度のプログラムにインパクトを与えないアイドル動作、たとえばOSスケジューラによりジョブのない時に負荷されたアイドルタスク、の負荷率が各カテゴリーのプロセッサ占有度によって計算される。アイドル動作で最も高い負荷率を持つコアがマルチコアのいずれにも指定されていない新しいタスク処理能力を持つことになる。

【0052】

マルチコアマイコン処理プロセスは、対称マルチ処理(SMP)と非対称マルチ処理(AMP)に分けられる。SMPでは単一のOSが全てのCPUを同時に管理しアプリケーションはどのCPUにもフロート可能である。一方、AMPは個別にOSを各コアで動かし、各アプリケーションは特定コアに固定されている。

【0053】

上記SMPは非決定的システムであり重要なソフトウェア機能を保証応答時間内に実行することが保証されず、保証応答時間を要求する自動車産業のシステムには適当でない。従って本実施例では、各コアの実行情報はAMPプロセスを持つマルチコアを用いて計算される。

【0054】

開発者は、設計段階で最高のパフォーマンスをあげるためにいくつかのタスクを固定プロセスサリソースに指定する必要がある。AMPの使用は各コア情報を用いた計算を必要とするが、レガシーコードの使用が容易であり、ソフトウェアのメモリやインターラプト等のシステムリソースの利用につき完全な制御を保証する。

【0055】

本実施例において各コアでの実行情報、例えばサイクル、命令、コール等の数は、各コアに指示する前に得られる。従って各コアのアイドル負荷率が計算され比較される。これにより、最も高いアイドル負荷率を持つコアに指示された新しいタスクを得ることが可能である。従って、図8においてアイドル負荷率80%を持つ第2コア(CPU#2)が選択される。この選択はユーザのディスプレイによる選択か、あるいは自動的にプログラムが設定しマルチコアの一つに保存したルールによって可能である。

【0056】

上述のように、本発明は、マルチコアマイコンシステムの開発装置と方法を提供し、それは、シミュレーションにおいてマルチコアマイコンコントローラおよび装置の全てのコアの種々のパラメータ状態を目立たずにモニター可能であることが明らかである。

10

20

30

40

50

## 【 0 0 5 7 】

さらに、各コアへの種々のソフトウェアの負荷などの種々の計算値もまたプログラマに表示され、プログラマがコントローラの最適効率のためにコアプログラミングを調整することを可能にする。

## 【 0 0 5 8 】

以上のように本発明を説明したが、当業者に自明な多くの変形例が本発明の主旨に含まれクレーム内に含まれる。

## 【 図面の簡単な説明 】

## 【 0 0 5 9 】

【 図 1 】 シミュレーション装置を示すブロック図である。

10

【 図 2 】 本発明実施例のコントローラモデルの詳細を示すブロック図である。

【 図 3 】 本発明実施例の装置モデルの詳細を示すブロック図である。

【 図 4 】 図 1 のシミュレーション装置の変形例を示すブロック図である。

【 図 5 A 】 本発明実施例の出力ディスプレイの例示画面である。

【 図 5 B 】 本発明実施例の出力ディスプレイの例示画面である。

【 図 5 C 】 本発明実施例の出力ディスプレイの例示画面である。

【 図 5 D 】 本発明実施例の出力ディスプレイの例示画面である。

【 図 6 】 本発明実施例のシミュレーション方法を示すフローチャートである。

【 図 7 】 図 6 の変形例を示すフローチャートである。

【 図 8 】 各コアのパラメータを示すテーブルである。

20

## 【 符号の説明 】

## 【 0 0 6 0 】

1 2 : コントローラモデル

1 4 : 装置モデル

1 8 : コア

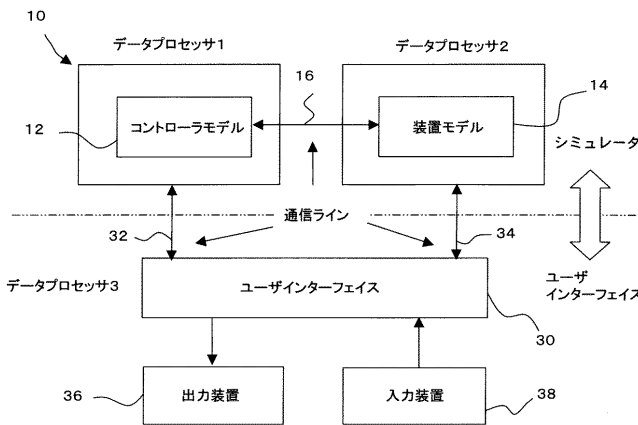
3 0 : ユーザーインターフェイス

3 6 : 出力装置

3 8 : 入力装置

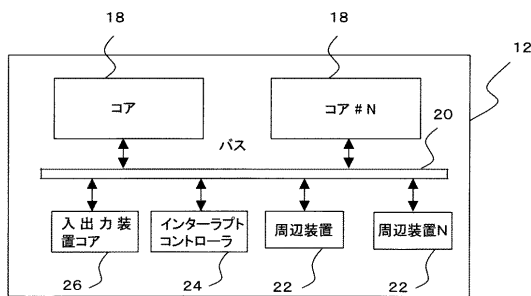
【図 1】

図 1



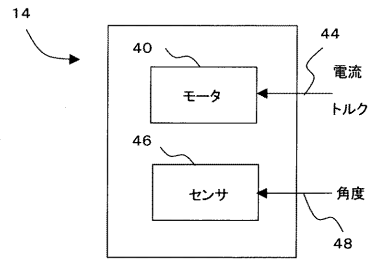
【図 2】

図 2



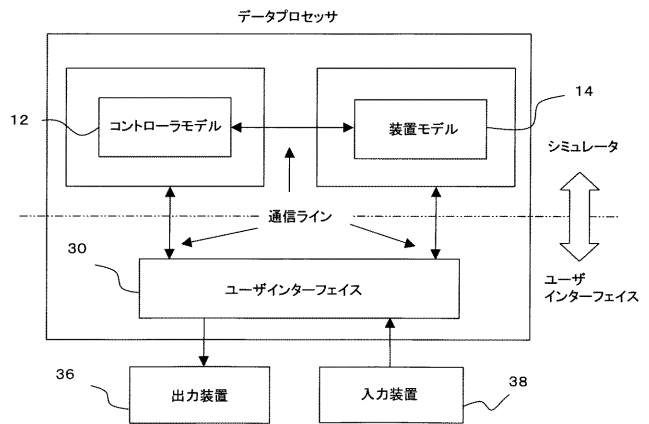
【図 3】

図 3



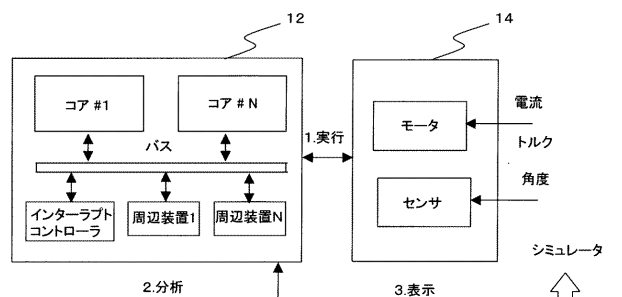
【図 4】

図 4



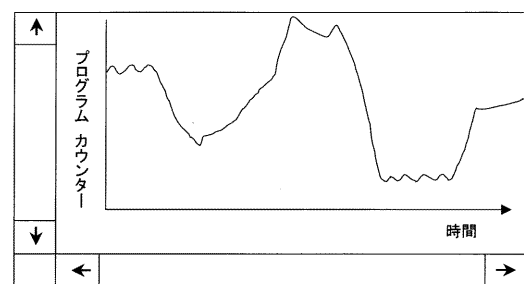
【図 5 A】

図 5A



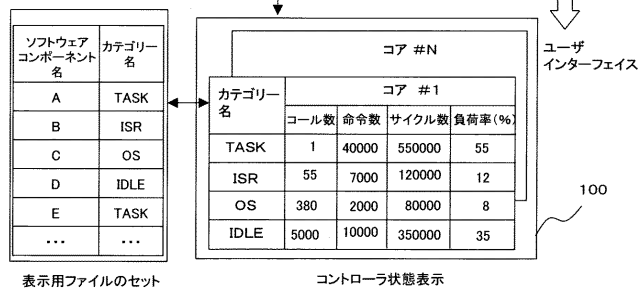
【図 5 B】

図 5B



【図 5 C】

図 5C



R0	3F20
R1	1C2A
R2	001E
・	・
・	・
・	・

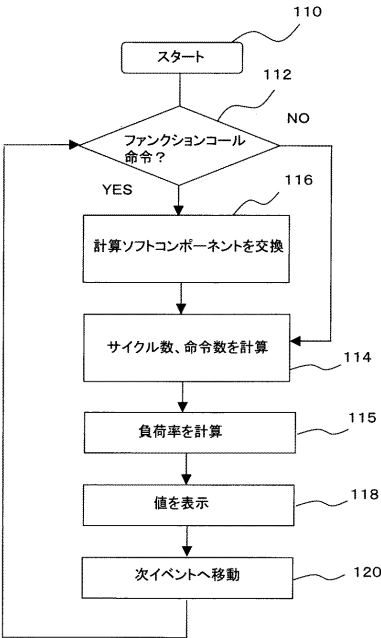
【 図 5 D 】

図 5D

	アドレス	コード	記号
↑	000000	0F9A	ADD R0, R1
	000002	C003	MOV R1, @R3
	000004	28DB	MUL R2, R1
	000006	F35E	JMP @R4
↓			

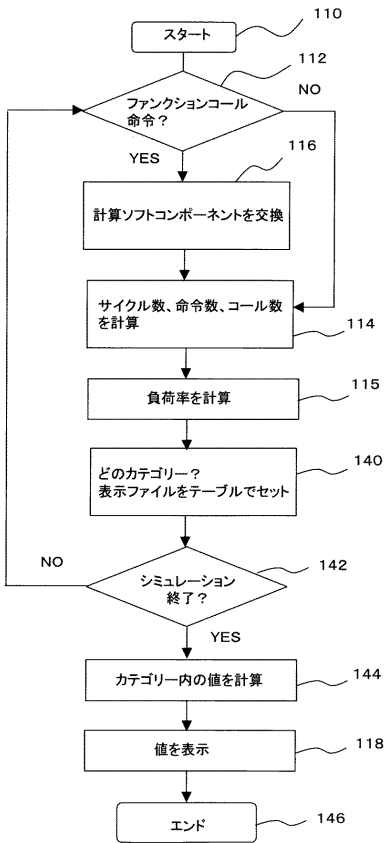
【 図 6 】

図 6



【 図 7 】

図 7



【 図 8 】

図 8

カテゴリ名	コア #1			
	コール数	命令数	サイクル数	負荷率
TASK	1	40000	550000	55
ISR	55	7000	120000	12
OS	380	2000	80000	8
IDLE	5000	100000	350000	35

カテゴリ名	コア #2			
	コール数	命令数	サイクル数	負荷率
TASK	2	4000	70000	7
ISR	55	7000	50000	5
OS	380	2000	80000	8
IDLE	5000	90000	800000	80

・  
・  
・

カテゴリ名	コア #N			
	コール数	命令数	サイクル数	負荷率
TASK	2	90000	800000	80
ISR	55	7000	50000	5
OS	380	2000	80000	8
IDLE	50	1000	70000	7

---

フロントページの続き

(72)発明者 スジット ファタック

アメリカ合衆国 4 8 3 3 5 ミシガン州 ファーミントンヒルズ ウッドリッジドライブ アパ  
ートメント 2 0 3 号 2 4 9 3 1

(72)発明者 ジョージ サイカリス

アメリカ合衆国 4 8 3 2 3 ミシガン州 ウェストブルームフィールド サマービルドライブ  
4 2 5 3

F ターム(参考) 5B042 GA04 GA11 GB05 HH07 HH25 LA05 LA21 NN08

5B046 AA08 HA09 JA04