

(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2010-15534
(P2010-15534A)

(43) 公開日 平成22年1月21日(2010.1.21)

(51) Int.Cl.

G06F 11/28 (2006.01)
G06F 17/50 (2006.01)

F 1

G06F 11/28 340C
G06F 17/50 664A
G06F 17/50 664J

テーマコード(参考)

5B042
5B046

審査請求 未請求 請求項の数 19 O L (全 13 頁)

(21) 出願番号 特願2008-332187 (P2008-332187)
 (22) 出願日 平成20年12月26日 (2008.12.26)
 (31) 優先権主張番号 12/164,178
 (32) 優先日 平成20年6月30日 (2008.6.30)
 (33) 優先権主張国 米国(US)

(71) 出願人 000005108
 株式会社日立製作所
 東京都千代田区丸の内一丁目6番6号
 (74) 代理人 110000350
 ポレール特許業務法人
 (72) 発明者 勝 康夫
 アメリカ合衆国 48322 ミシガン州
 ウエストブルームフィールド ケンネック
 トスクエア 7763
 (72) 発明者 ドナルド ジェイ マッキューン
 アメリカ合衆国 48336 ミシガン州
 ファーミントンヒルズ トゥレーンアベ
 ニュー アパートメント103号 213
 69

最終頁に続く

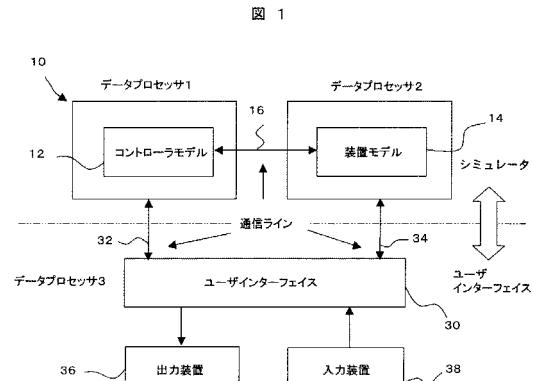
(54) 【発明の名称】マルチコアマイコンシステムのシミュレーション方法及びシミュレーション装置

(57) 【要約】

【課題】マルチコアマイコンシステムの開発方法及び開発装置において、少なくとも一つのパラメータを有するマルチコアを有するコントローラモデルがシミュレートされ、同時に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より制御される装置モデルがシミュレートされる。

【解決手段】ユーザインターフェイスがコントローラモデルおよび装置モデルのパラメータにアクセスし、トリガーアイベントに応じて選択的にコントローラモデルおよび装置モデルの実行を保留する。ユーザインターフェイスは、トリガー時にコントローラモデルおよび装置モデルのパラメータを変更することなく、コントローラモデルの各コアの種々のパラメータ及び装置モデルのパラメータの状態を決定し、決定された両コアのパラメータは表示装置に表示される。

【選択図】図1



【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。

【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。

【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。

【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。

【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

10

20

30

40

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。

【請求項 1 3】

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。

【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。

【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。

【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】

10

20

30

40

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。

10

【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。

20

【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。

20

【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ムの変更は、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。

30

【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。

40

【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

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】

10

20

30

40

50

【0020】

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10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12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16により装置モデル14の動作を制御する。装置モデル14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。その他の装置モデルにおいて、コントローラモデル12は動作制御すなわちスロットル弁の開閉制御を行う。

【0021】

図2は、コントローラモデル12のブロック図を示す。コントローラモデル12は少なくとも二つ乃至N個のコア18を有し、各コア18はCPUを構成する。各コア18はさらにバス20を介して少なくとも一つの周辺モジュール22と交信する。バス20はインターラプトコントローラ24と交信しても良く、入出力ポート26と交信しても良い。

10

【0022】

マルチコア18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12において、各コア18は独立に各自のコンピュータプログラムを他のコア18と同時に実行する。全てのコアは共通データバス20を共有しており、二つ以上のコア18が同時にデータバス20にアクセス要求した場合は、データバス20が再びアイドル状態のコア18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18がアイドル状態に入る。

20

【0023】

図3は、電子スロットル制御としての装置モデル14のブロック図を示す。そのような装置モデル14は、ライン44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40を有する。装置モデル14はまたシミュレートされたセンサ46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48を有する。ライン44、48の値は装置モデル14のパラメータを構成し、これはコントローラモデル12の命令に応じて変化する。

30

【0024】

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30はコントローラモデル12と入出力ライン32を介して通信する。同時に、ユーザインターフェイス30は入出力ライン34を介して装置モデル14と通信する。ユーザインターフェイス30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36と通信し、システムオペレータがコントローラ12と装置モデル14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38もまた、ユーザインターフェイス30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。

30

【0025】

ユーザインターフェイス30とコントローラ12、装置モデル14を結ぶ通信ライン32、34は、ユーザインターフェイス30を介してシステムオペレータにコントローラ12、装置モデル14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32、34はユーザインターフェイス30に、コントローラモデル12、装置モデル14のどのパラメータ値も変えずに、コントローラモデル12と装置モデル14の実行を保留させることができる。

40

【0026】

さらに、シミュレートされたコントローラモデル12と装置モデル14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12内の各コア18の種々のパラメータ、または図3に示す装置モデル14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18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12は、コントローラモデル12と装置モデル14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。

50

【0027】

図1のブロック図において、コントローラモデル12と装置モデル14を有するシミュレータおよびユーザインターフェイス30は、コントローラモデル12と装置モデル14およびユーザインターフェイス30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。

【0028】

しかし、コントローラモデル12と装置モデル14およびユーザインターフェイス30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12や装置モデル14、インターフェイス30の実行のために用いられても良い。

10

【0029】

図5A～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36に表示される。例えば図5Aにおいてテーブル100はコントローラモデル12の各コアのために表示され、ここで各テーブル100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。

20

【0030】

上記テーブル100は、コア18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100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12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55%を消費し、「I D L E」には5000コールがあり第1コア時間の35%を消費する等である。

20

【0031】

図5B～5Dは、出力デバイス36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5Bはコントローラモデル12または装置モデル14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5Cはコントローラ12のコアの一つのレジスタ値テーブルを示す。図5Dはコントローラ12のコア18で現在実行中のアセンブリコードテーブルを示す。

【0032】

例えば、図5Dで、アドレス000000はコード0F9Aをストアし、レジスタR0、R1を図5Cのテーブル中でADDする。アドレス000002はコードC003をストアし、レジスタR1の値をテーブル中でメモリのR3に移動する。アドレス000004はコード28DBをストアし、テーブル中でレジスタR2とR3の値を掛ける。アドレス000006はコードF35Eをストアし、テーブル中でレジスタのR4にジャンプする。

30

【0033】

テーブル100の負荷率は計算値であり、一方テーブル100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100がコントローラ12の各コアについて生成され、この複数のテーブル100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30で表示可能である。

40

【0034】

図6に示すフローチャートには、コントローラ12の各コア18のテーブル100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110でスタートし、すぐにステップ112に進む。ステップ112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112はステップ114に分岐する。

【0035】

もしステップ112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112はステップ116に進みファンクションコールIDまたはアドレスが決定される。次いでステップ116はステップ114に進む。

50

【0036】

ステップ 114において、インターフェイス 30はコントローラモデルの一つのコア 18のサイクル数と命令数を計算し、ステップ 115に進む。

【0037】

ステップ 115で、インターフェイス 30は、コントローラモデルのコア 18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 115は次いでステップ 118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 118は次いでステップ 120に進み、コントローラモデルのコア 18のための次の命令がステップ 112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。

【0038】

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 12 の各コア 18 につき反復される。この方法で図 5 に示すテーブル 100 がコントローラモデル 12 の各コア 18 につき生成され表示装置に表示されても良い。

10

【0039】

これはプログラマがコントローラモデル 12 の種々のコア 18 に対しプログラミングを調整し、これにより全てのコア 18 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 12 の種々のコア 18 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。

【0040】

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 12 の全てのコア 18 の種々のファクタとパラメータの総計を示す。

20

【0041】

更に詳細に言えば、プログラムはステップ 115 の負荷率計算の後でステップ 140 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。

【0042】

ステップ 142 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 142 はステップ 112 に戻り、ステップ 112 で開始された処理が各コア 18 の次の命令のために反復される。

30

【0043】

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 142 は代わりにステップ 144 に進み、コントローラモデル 12 の各コア 18 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 144 は次いでステップ 118 に進み、それらの値を表示し、ステップ 146 に進み、シミュレーションが終了する。

【0044】

動作中に、シミュレーションはユーザインターフェイス 30 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 30 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。

【0045】

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 30 を介して起動され、またはコア 18 の一つの特定レジ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 30 により起動される。

40

【0046】

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 30 はコントローラモデル 12 および装置モデル 14 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 12 の種々のコア 18 の種々のパラメータ及び装置モデル 14 のパラメータが表示装置に表示される。

【0047】

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ます。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。

10

【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。

20

【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。

30

【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ッ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。

40

【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。

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5A】本発明実施例の出力ディスプレイの例示画面である。

【図5B】本発明実施例の出力ディスプレイの例示画面である。

【図5C】本発明実施例の出力ディスプレイの例示画面である。

【図5D】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。

20

【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。

【 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

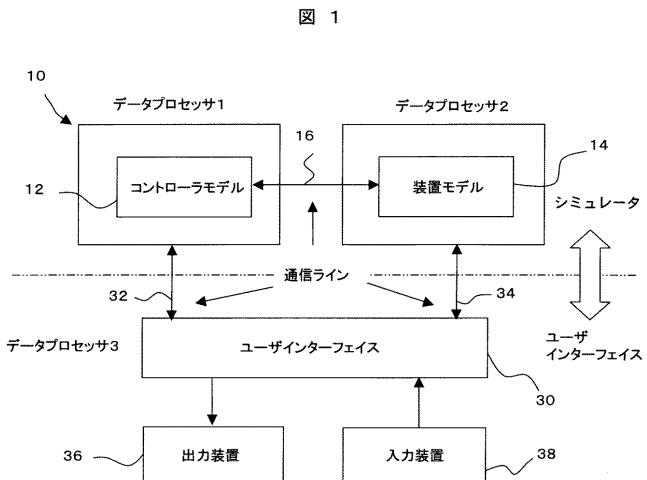
1 8 : コア

3 0 : ユーザインターフェイス

3 6 : 出力装置

3 8 : 入力装置

【 図 1 】



【図2】

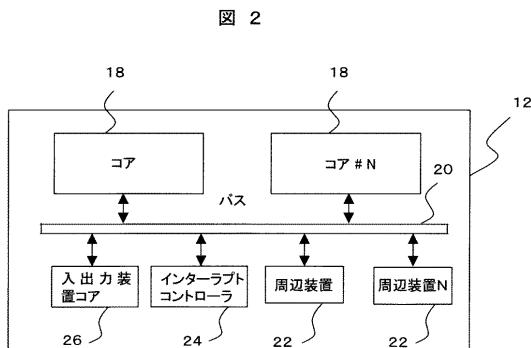
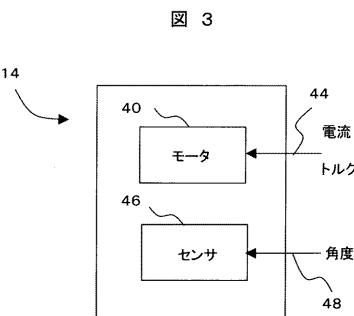


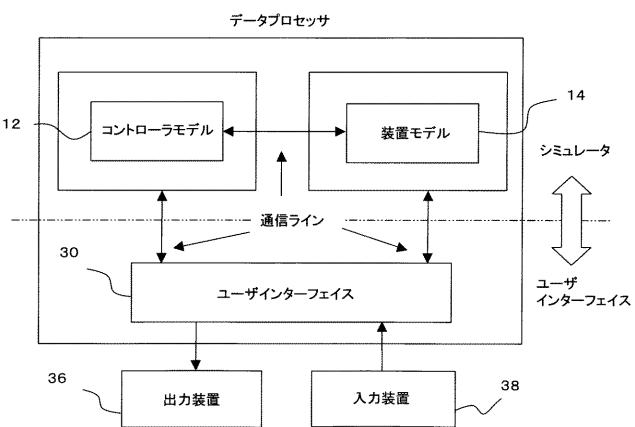
図 2

【 図 3 】



〔 図 4 〕

4

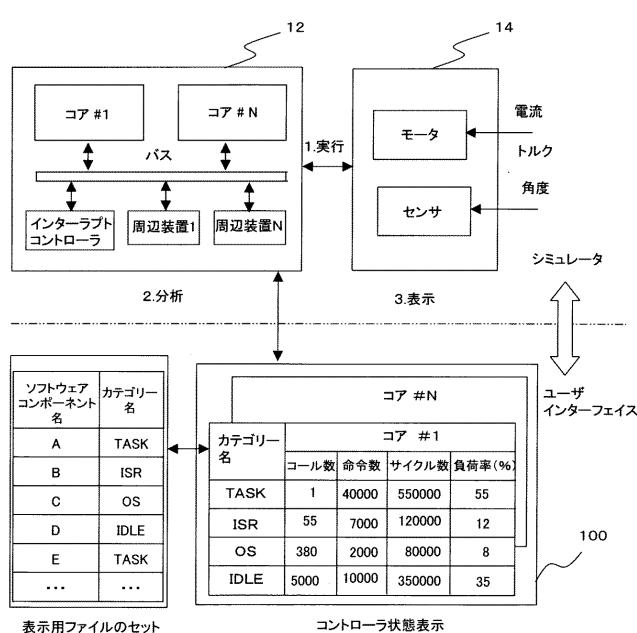


【图 5-1】

四 五

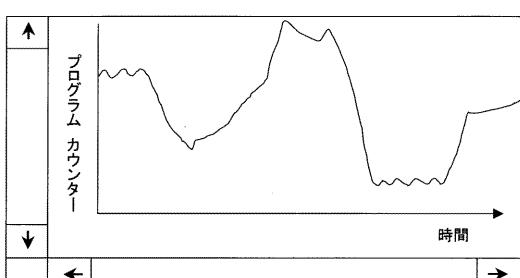
四 58

〔 図 5 A 〕



表示用ファイルのセット

コントローラ状態表示



【 5 C 】

5C

R0	3F20
R1	1C2A
R2	001E
-	-
-	-
-	-

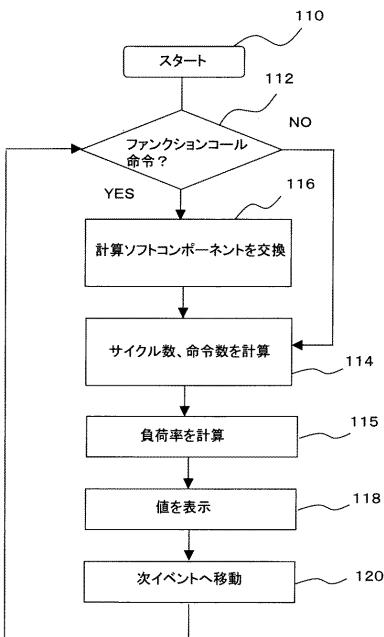
【図 5D】

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
↓			

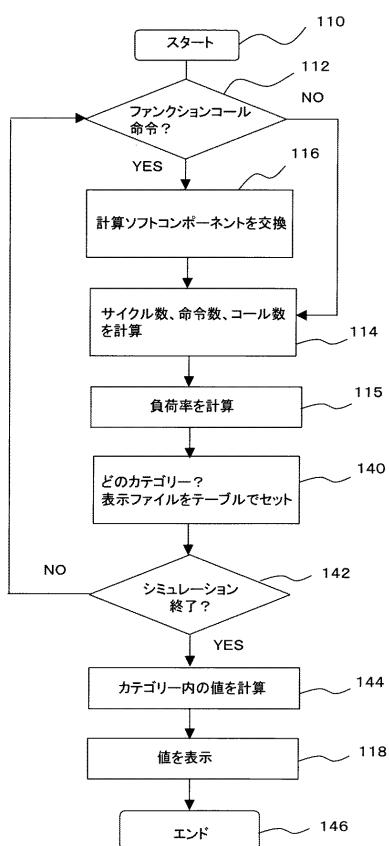
【図 6】

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 203号 24931

(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 ウエストブルームフィールド サマービルドライブ 4253

F ターム(参考) 5B042 GA04 GA11 GB05 HH07 HH25 LA05 LA21 NN08
5B046 AA08 HA09 JA04