

(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局(43) 国際公開日
2011年11月17日(17.11.2011)

PCT



(10) 国際公開番号

WO 2011/141992 A1

(51) 国際特許分類:

G06F 11/22 (2006.01)

(21) 国際出願番号:

PCT/JP2010/057910

(22) 国際出願日: 2010年5月10日(10.05.2010)

(25) 国際出願の言語: 日本語

(26) 国際公開の言語: 日本語

(71) 出願人(米国を除く全ての指定国について): トヨタ自動車株式会社 (TOYOTA JIDOSHA KABUSHIKI KAISHA) [JP/JP]; 〒4718571 愛知県豊田市トヨタ町1番地 Aichi (JP).

(72) 発明者; および

(75) 発明者/出願人(米国についてのみ): 繁原 英一郎 (SHIGEHARA, Eiichiro) [JP/JP]; 〒4718571 愛知県豊田市トヨタ町1番地 トヨタ自動車株式会社内 Aichi (JP).

(74) 代理人: 伊東 忠彦 (ITOH, Tadahiko); 〒1506032 東京都渋谷区恵比寿4丁目20番3号 恵比寿ガーデンプレイスタワー32階 Tokyo (JP).

(81) 指定国(表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) 指定国(表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), ヨーロッパ (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

[続葉有]

(54) Title: FAULT DIAGNOSIS DEVICE AND FAULT DIAGNOSIS METHOD

(54) 発明の名称: 故障診断装置及び故障診断方法

[図1]

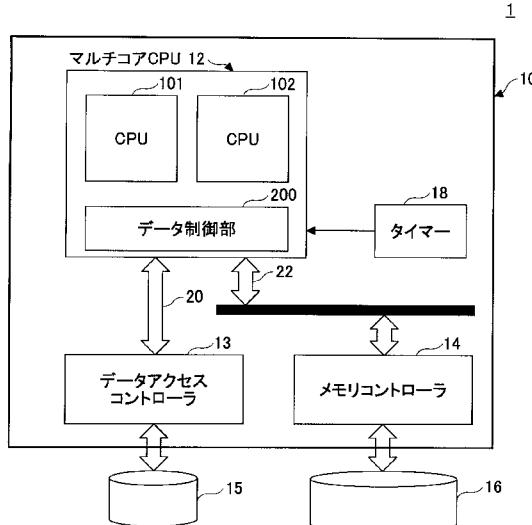


FIG. 1:
 12 MULTI-CORE CPU
 13 DATA ACCESS CONTROLLER
 14 MEMORY CONTROLLER
 18 TIMER
 200 DATA CONTROL UNIT

(57) Abstract: Disclosed is a fault diagnosis device for performing fault diagnostics for a plurality of CPUs, wherein, on the basis of vehicular information related with processing of the plurality of CPUs, the processing load of the plurality of the CPUs as an entirety is predicted or detected, and according to the prediction results or the detection results of the processing load, the execution mode of the fault diagnostics are modified. The plurality of CPUs may be a plurality of CPUs in a multi-core processor. The fault diagnostics may be performed when the processing load of the plurality of the CPUs as an entirety is predicted or detected to be lower than a predetermined criterion.

(57) 要約: 本発明は、複数のCPUの故障診断を行う故障診断装置であって、複数のCPUの処理に関する車両情報に基づいて、複数のCPUの全体としての処理負荷を予測又は検出し、該処理負荷の予測結果又は検出結果に応じて、故障診断の実行態様を変更することを特徴とする。複数のCPUは、マルチコアプロセッサにおける複数のCPUであってよい。複数のCPUの全体としての処理負荷が所定基準よりも低いことが予測又は検出された場合に、故障診断を行うものであってよい。



添付公開書類:

— 国際調査報告（条約第 21 条(3)）

明細書

発明の名称：故障診断装置及び故障診断方法

技術分野

[0001] 本発明は、複数のCPUの故障診断を行う故障診断装置及び故障診断方法に関する。

背景技術

[0002] 従来から、幾つかの作業単位用プロセッサ装置と、それらを総合的に制御する総合制御用プロセッサ装置とが共通バスを介して接続されるマルチプロセッサ装置において、作業単位用プロセッサ装置の故障を調べる診断試験プログラムを、作業単位用プロセッサ装置毎に搭載するマルチプロセッサ装置が知られている（例えば、特許文献1参照）。この構成では、異常を検知した作業単位用プロセッサ装置は、自己に搭載されている診断試験プログラムによって故障の調査を行い、総合制御用プロセッサ装置の手を借りて故障の調査を行わないので、故障の調査をする度に共通バスを占有して他の処理を中断させたり、総合制御用プロセッサ装置に処理負担をかけたりすることがなくなる。

先行技術文献

特許文献

[0003] 特許文献1：特開平7-325730号公報

発明の概要

発明が解決しようとする課題

[0004] ところで、車載マイコン（ECU）のCPUは、通常処理（例えば車両制御用の処理）に加えて故障診断処理を実施するため、故障診断処理を行うことにより通常処理に影響が出る場合がある。

[0005] そこで、本発明は、CPUの通常処理への影響が少なくとも低減される様で故障診断処理を行うことが可能な故障診断装置及び故障診断方法の提供を目的とする。

課題を解決するための手段

[0006] 上記目的を達成するため、本発明の一局面によれば、複数のCPUの故障診断を行う故障診断装置であって、

前記複数のCPUの処理に関連する車両情報に基づいて、前記複数のCPUの全体としての処理負荷を予測又は検出し、該処理負荷の予測結果又は検出結果に応じて、故障診断の実行態様を変更することを特徴とする、故障診断装置が提供される。

発明の効果

[0007] 本発明によれば、CPUの通常処理への影響が少なくとも低減される態様で故障診断処理を行うことが可能な故障診断装置及び故障診断方法が得られる。

図面の簡単な説明

[0008] [図1]本発明の一実施例（実施例1）による故障診断装置1のハードウェア構成を示す図である。

[図2]マルチコアCPU12を内蔵したSOCをハードウェアとしたときのシステムレイヤを示す図である。

[図3]OS内の故障診断テスト制御の一例を示すフローチャートである。

[図4]図1に示したマルチコアCPU12の詳細構成の一例を示す図である。

[図5]CPUコア101の故障診断を開始する直前のマルチプレクサ201, 202, 205の接続状態の概要を示す図である。

[図6]故障診断テスト用のデータと期待値の読み込み状態を示す図である。

[図7]CPUコア101からのテスト結果の出力状態を示す図である。

[図8]CPUコア101の故障診断完了後のマルチプレクサ201, 202, 205の接続状態の概要を示す図である。

[図9]故障診断テスト制御のその他の一例を示すフローチャートである。

[図10]アプリ層に故障診断テスト制御を組み込んだ場合の構成を示す図である。

[図11]三角波比較PWMの概要を示す図である。

[図12]キャリア周波数の変化（車両駆動用電動モータの回転速度の変化）に伴うCPUコアに割当てられる時間の変化を概略的に示す図である。

[図13]車速の変化に伴う周辺監視システムにおける必要な探索範囲の変化を概略的に示す図である。

[図14]本発明のその他の一実施例（実施例2）による故障診断装置2のハードウェア構成を示す図である。

[図15]処理負荷に応じてCPUコアの故障診断の実行部分を可変する故障診断テスト制御の一例を示す。

[図16]処理負荷に応じてタスク調整を行いCPUコアの故障診断の実行可能な機会を増加させる故障診断テスト制御の一例を示す。

[図17]テストモードの間に処理負荷が高くなった場合に故障診断の中止を行う故障診断テスト制御の一例を示す。

発明を実施するための形態

[0009] 以下、図面を参照して、本発明を実施するための最良の形態の説明を行う。

[0010] 図1は、本発明の一実施例（実施例1）による故障診断装置1のハードウェア構成を示す図である。故障診断装置1は、LSI (large-scale integration) 10により構成される。LSI 10は、マルチコアCPU12、データアクセスコントローラ13、メモリコントローラ14及びタイマ18等を含む。

[0011] マルチコアCPU12は、複数のCPUコア（本例では、2つのCPUコア101, 102）と、CPUコア101, 102間のデータ連携を行うデータ制御部200を含む。

[0012] データアクセスコントローラ13は、CPUコア101, 102の故障診断時に必要なデータを格納している記憶媒体15へのアクセスを制御する。記憶媒体15は、例えばメモリやHDDなどにより構成される。データアクセスコントローラ13とマルチコアCPU12との間は、データバス20により接続されている。

- [0013] メモリコントローラ 14 は、通常動作に必要なプログラムやデータを格納している記憶媒体 16 へのアクセスを制御する。記憶媒体 16 は、例えばメモリやHDDなどにより構成される。メモリコントローラ 14 とマルチコアCPU 12との間は、データバス 22 により接続されている。
- [0014] 図2は、マルチコアCPU 12 を内蔵したSOC (System-on-a-chip) をハードウェアとしたときのシステムレイヤを示す図である。OSは、動作させるアプリから今後必要とされるCPU負荷を推定したり、各CPUコア101, 102の動作状態の管理やタスク割り当てを行う。
- [0015] 図3は、OS内の故障診断テスト制御の一例を示すフローチャートである。
- [0016] ステップ300では、CPUコア101, 102の処理負荷に関連する車両情報が入力される。車両情報は、CPUコア101, 102の処理負荷の予測（又は検出）に利用できる情報であれば任意である。車両情報の具体例について後述する。
- [0017] ステップ301では、上記ステップ300で入力された車両情報に基づいて、CPUコア101, 102の処理負荷（マルチコアCPU 12 の処理負荷）が予測される。例えば、現時点から所定期間T（以下、負荷予測期間Tという）内のCPUコア101, 102の処理負荷が予測されてもよい。負荷予測期間Tは、後述の故障診断に要する時間（即ちステップ310からステップ328までの処理に要しうる時間の最大値）に対応してよい。また、CPUコア101, 102の処理負荷は、現時点から負荷予測期間T内のCPUコア101, 102の処理負荷の平均値や最大値として予測されてもよい。尚、ここで予測されるCPUコア101, 102の処理負荷は、CPUコア101, 102が現時点から通常動作を行った場合を想定したときの処理負荷であり、CPUコア101又は102が後述の故障診断を行った場合を想定したときの処理負荷ではない。
- [0018] ステップ302では、上記ステップ301で予測した処理負荷が所定レベルよりも高いか否かが判定される。所定レベルは、1つのCPUコアで処理

可能な処理負荷の最大レベルに対応してよい。この場合、所定レベルは、C P Uコア101, 102の個々の能力（性能）に応じて適合されてよい。尚、C P Uコア101, 102の性能は、一般的に、互いに同一である。上記ステップ301で予測した処理負荷が所定レベルよりも高い場合は、1つのC P Uコアで処理可能で無いと判断して、ステップ304に進み、上記ステップ301で予測した処理負荷が所定レベルよりも低い場合は、1つのC P Uコアで処理可能であると判断して、ステップ310に進む。

- [0019] ステップ304では、C P Uコア101, 102を通常モードのまま動作させると共に、各C P Uコア101, 102への負荷分散を行う。即ち、各C P Uコア101, 102は、SMP (Symmetric Multi-processing) 動作を行う。
- [0020] ステップ306では、OSによりスケジューリングされたタスクに基づいて、各C P Uコア101, 102がそれぞれのタスクを実行する。
- [0021] ステップ308では、各C P Uコア101, 102の通常処理の実行時間（上記ステップ304から計時される実行時間）が負荷予測期間Tに達したか否かが判定される。各C P Uコア101, 102の通常処理の実行時間が負荷予測期間Tに達していない場合は、ステップ306に戻り、各C P Uコア101, 102の通常処理の実行時間が負荷予測期間Tに達した場合は、ステップ300に戻る。
- [0022] ステップ310では、故障診断対象のC P U番号が呼び出される。故障診断対象のC P U番号は、C P Uコア101, 102のいずれかを表す番号であり、所定の順序で交互に呼び出される。
- [0023] ステップ312では、故障診断対象のC P Uコア（C P Uコア101又は102）に、処理が完了していないタスク（処理途中のタスク）が存在するか否かが判定される。処理が完了していないタスクがある場合は、ステップ314に進み、処理が完了していないタスクが無い場合は、ステップ318に進む。
- [0024] ステップ314では、処理が完了しないタスクが完了することを待機する

。

- [0025] ステップ316では、上記ステップ312と同様に、故障診断対象のCPUコア(CPUコア101又は102)に、処理が完了していないタスクが存在するか否かが判定される。処理が完了していないタスクがある場合は、ステップ314に戻り、処理が完了していないタスクが無い場合は、ステップ318に進む。
- [0026] ステップ318では、故障診断対象のCPUコア(CPUコア101又は102)のモードを、通常モードからテストモードへ変更する。このようにして、各CPUコア101, 102の動作は、SMPからAMP(Asymmetric Multiprocessing)に変更される。即ち、故障診断対象のCPUコア101又は102は、テストモードで動作し、他方のCPUコア(故障診断対象でないCPUコア102又は101)は、OSによりスケジューリングされたタスクを実行する。
- [0027] ステップ320では、故障診断対象のCPUコア(CPUコア101又は102)のテストモードが開始され、故障診断処理が開始される。具体的には、データアクセスコントローラ13(図1参照)を通して、故障診断テスト用のデータとそれに対応する期待値の記憶媒体15からの読み出しを開始する。そして、期待値とテスト結果とを照合した結果を記録媒体に書き込む。
- [0028] このテストモードの間、OSは、全てのタスクを他方のCPUコア(故障診断対象でないCPUコア102又は101)に割当てる。かかる場合でも、上記のステップ301, 302での予測・判定が適切である限り、全てのタスクは上述の如く1つのCPUでの処理可能なレベルの処理負荷であるので、全てのタスクを他方のCPUコア単独で実行可能である。従って、この際に他方のCPUコアに割当てられるタスクには、SMP動作時であれば(即ちテストモードに移行していなければ)故障診断対象のCPUコアに割当てられていただろうタスクを含みうる。
- [0029] 本ステップ320における故障診断処理の内容は、自身の故障診断を適切

に行うことができる処理である限りに任意である。例えば、自身のCPUコアのALUの演算テスト等であってよい。故障診断処理の好ましい一例については後述する。

- [0030] ステップ322では、故障診断処理が完了したか否かが判定される。故障診断処理が完了したと判定された場合には、ステップ326に進み、故障診断処理が完了していないと判定された場合は、ステップ324に進む。
- [0031] ステップ324では、故障診断処理が継続される。故障診断テストを継続するために次の故障診断テスト用のデータとそれに対応する期待値をデータアクセスコントローラ13を通して読み出す。そして、期待値とテスト結果とを照合した結果を記録媒体に書き込む。ステップ324の処理は、故障診断処理が完了するまで（全ての故障診断テスト用のデータのテストが終了するまで）継続される。
- [0032] ステップ326では、故障診断対象のCPUコアのモードをテストモードから通常モードに変更する。即ち、故障診断対象のCPUコアを通常モードに復帰させる。
- [0033] ステップ328では、マルチコアCPU12内のデータ制御部200が管理する次回の故障診断対象のCPU番号を更新する。
- [0034] 図4は、図1に示したマルチコアCPU12の詳細構成の一例を示す図である。
- [0035] マルチコアCPU12のデータ制御部200は、図4に示すように、マルチプレクサ（MUX）201, 202, 205と、モード制御部203と、CPU番号レジスタ204と、故障診断ブロック210とを含む。
- [0036] マルチプレクサ201は、CPUコア101の入力／出力データの読み出し／書き込みを、データアクセスコントローラ13（故障診断に必要なデータを格納した記憶媒体15にアクセスするためのデータアクセスコントローラ13）と行うか、若しくは、メモリコントローラ14（通常動作に必要なデータを格納した記憶媒体16にアクセスするためのメモリコントローラ14）と行うかを制御する。

- [0037] マルチプレクサ202は、マルチプレクサ201と同様に、CPUコア102の入力／出力データの読み出し／書き込みを、データアクセスコントローラ13と行うか、若しくは、メモリコントローラ14と行うかを制御する。
- [0038] モード制御部203は、各CPUコア101, 102のモード（通常モード若しくは故障診断用のテストモード）を切り換える。
- [0039] CPU番号レジスタ204は、次の故障診断対象のCPUコア（CPUコア101又は102）の番号（図3のステップ310, 328参照）を格納するレジスタである。
- [0040] マルチプレクサ205は、故障診断対象のCPUコア（CPUコア101又は102）の出力データを選択する。
- [0041] 故障診断ブロック210は、故障診断対象のCPUコア（CPUコア101又は102）の出力結果（処理結果）とテスト結果の期待値とを比較する機能を有する。故障診断ブロック210は、故障診断対象のCPUコア（CPUコア101又は102）の出力結果を記憶する記憶部212と、テスト結果の期待値を記憶する記憶部214と、これらを比較する比較器216とを含む。
- [0042] ここで、図3、図5乃至図8を参照して、CPUコア101が通常動作モードからテストモードに遷移し、CPUコア101のテストモード（故障診断処理）が完了するまでの動作を説明する。
- [0043] 尚、前提条件として、データ制御部200のCPU番号レジスタ204には、値「0」が格納されているものとする。値「0」は、CPUコア101の番号に対応し、値「1」は、CPUコア102に対応するものとする。
- [0044] 先ず、車両情報を入力する（図3のステップ300参照）。次いで、入力された車両情報からCPUコア101, 102の処理負荷（全体としての処理負荷）の推定（予測）を行う（図3のステップ301参照）。推定された処理負荷が1つのCPUコアで処理できないレベルである場合は（図3のステップ302のYES参照）、図3のステップ304乃至308が実行され

、故障診断のテストは実行されない。

- [0045] 次の車両情報更新タイミングで、入力された車両情報からCPUコア101、102の処理負荷の推定を行う（図3のステップ301参照）。推定された処理負荷が1つのCPUコアで処理できるレベルである場合は（図3のステップ302のNO参照）、故障診断用のテストモードに移行する準備として、データ制御部200から故障診断対象のCPUコアの番号を読み出す。このとき、CPU番号レジスタ204には値「0」が格納されているので、CPUコア101が故障診断対象のCPUコアとなる。
- [0046] CPUコア101が故障診断対象のCPUコアとして決定されたときに、CPUコア101が通常処理の実行途中である場合、モードを移行して故障診断処理を開始することはできない（故障診断処理を開始すると、通常動作の処理結果が異常となる）。このため、CPUコア101が実行中のタスクの処理を完了するまで待機される（図3のステップ312乃至ステップ316参照）。
- [0047] CPUコア101で処理されているタスクが無くなると（CPUコア101が実行中のタスクの処理を完了すると）、CPUコア101のモードが、データ制御部200内のモード制御部203により変更される（図3のステップ318参照）。即ち、CPUコア101のモードが、通常モードからテストモードに変更される。これと同時に、モード制御部203は、マルチプレクサ201、202、205に接続するバスをテスト系のバスとする。この状態での各マルチプレクサ201、202、205の内部接続イメージと各CPUコア101、102へのモード設定信号は、図5のように示される。
- [0048] CPUコア101のモードをテストモードに変更して、テスト系のバスに接続すると、テスト用の記憶媒体15からCPUコア101へ故障診断テスト用のデータが入力されると共に、テスト用の記憶媒体15から故障診断ブロック210の記憶部214に、対応する期待値が入力される。この状態が図6に示される。

- [0049] C P Uコア101は、故障診断テスト用のデータが入力されると随時それぞれのデータを処理して、結果を出力する。出力結果は、データ制御部200内の故障診断ブロック210の記憶部212に格納される。この状態が図7に示される。
- [0050] データ制御部200は、C P Uコア101の出力結果と記憶部214内の期待値との比較を随時行う（図3のステップ320乃至ステップ324参照）。
- [0051] 全ての故障診断テスト用のデータを用いた故障診断処理が完了し、各出力結果と各期待値との比較がエラー無く完了すると、C P Uコア101のモードがテストモードから通常モードに変更される（図3のステップ326参照）。これと同時に、C P U番号レジスタ204の値が「1」に変更される（図3のステップ328参照）。この状態が図8に示される。尚、故障診断の結果、故障が検出された場合（各出力結果と各期待値との不一致がある場合）は、外部に故障を知らせる信号を出力することとしてもよい。この信号が出力されると、ユーザ（運転者）にE C Uの故障のためにE C Uを交換又は検査するように促すメッセージが、ユーザ（運転者）に対して音声及び／又は表示により出力されてもよい。
- [0052] 尚、その後、C P Uコア101, 102の処理負荷の予測を行い、故障診断処理に必要な時間（負荷予測期間T）に1つのC P Uコアによる処理が可能なレベルの処理負荷が予測された場合には、同様に、C P Uコア102の故障診断が実行される。
- [0053] 以上説明した本実施例の故障診断装置1によれば、とりわけ、以下のような優れた効果が奏される。
- [0054] 上述の如く、故障診断処理がC P Uコア101, 102の全体としての処理負荷が低い場合に実行されるので、C P Uコア101, 102の通常処理に影響しない態様で、C P Uコア101, 102の故障診断を行うことができる。即ち、C P Uコア101, 102の処理負荷を車両情報から予測して、マルチコアC P U12の動作モードをS M P動作からA M P動作に動的に

切換ながら故障診断を行うことによって、マルチコアCPU12の通常動作の性能に影響が出ない様で、CPUコア101、102の故障診断を実現することができる。また、CPUコア101、102のそれぞれの故障診断が実行され、いずれかのCPUコア101、102に故障が生じた場合には、当該故障を検出することができると共に、どのCPUコア101、102に故障が生じているかを特定することができる。

- [0055] 尚、上述した実施例では、故障診断を行う毎に、記憶媒体15から故障診断テスト用のデータ及び期待値をデータ制御部200に読み込む構成であった。しかしながら、常に同じのテストパターンによりCPUコア101、102の故障診断を行う構成であれば、期待値は一定である。従って、かかる構成であれば、データ制御部200に内蔵のROM若しくはRAMなどの記憶領域を用意しておき、期待値を予め格納しておいてもよい。かかる構成によれば、データアクセスコントローラ13を通したデータアクセスが不要となり、故障診断を行う際にバス能力の低下のリスクを避けることができる。
- [0056] また、上述した実施例では、車両情報更新周期毎（例えば1秒毎）に図3の処理が実行されているが、実行周期は、故障診断の内容等に依存して、任意であってよい。例えば経年劣化に起因した故障発生のチェックは、間隔が短くても1日周期で実行すれば十分である。従って、前回の故障診断を行った時間を記憶しておき、一定時間以上故障診断が行われなかつた場合のみ図3の処理が実行されるようにしてもよい。具体的には、図9に示すように、図3の故障診断テスト制御にステップ900及び902の処理を追加してもよい。即ち、各CPUコア101、102の最終（前回）の故障診断時間と現在の時間を比較して一定時間以上（例えば一日以上）故障診断が行われなかつた場合のみ図3の処理（ステップ300からの処理）が実行される。尚、図9に示す故障診断テスト制御は、CPUコア101、102が車両の走行性能に関連しないシステム（即ち、ユーザの利便性や快適性を高めるためのシステムであり、例えば、ナビゲーションシステム）のECUを構成する場合に好適である。

- [0057] また、上述した実施例では、故障診断テストの制御（図3参照）をOSにより実現しているが、アプリやミドル層で制御しても同様の効果を得ることが可能である。この場合、各アプリや故障診断テストの制御を処理するCPUコアを事前に割り振っておく。例えば、図10に示す例では、故障診断テスト制御をCPUコア101に割当てた例が示される。尚、この場合は、CPUコア101, 102は、AMPモードで動作する。
- [0058] 次に、車両情報からCPUコア101, 102の処理負荷を予測する方法の具体例について幾つか説明する。
- [0059] 第1の具体例は、車両情報として、車両駆動用電動モータの制御に用いるPWM信号のキャリア周波数を用いて、CPUコア101, 102の処理負荷を予測する方法に関する。ここでは、CPUコア101, 102は、車両駆動用電動モータの制御を実行する。即ち、CPUコア101, 102は、ハイブリッドシステムのECUを構成する。
- [0060] ハイブリッド車（及び電気自動車）では、車両駆動用電動モータを駆動して車両走行が制御される。この車両駆動用電動モータの制御には、一般的に、図11に示すような三角波比較PWMが使用される。インバータの制御を滑らかに行うために同期PWM制御（キャリア位相同期）が用いられる。この同期PWM制御では、車両駆動用電動モータの回転速度に応じてキャリア周波数が変化する。即ち、同期PWM制御では、車両駆動用電動モータの回転速度が速くなるにつれて、キャリア周波数が高くなる。キャリア周波数が高くなると、CPUコア101, 102に要求される処理能力が高くなる。即ち、図12に示すように、キャリア周波数が高くなると、サイン波の周期が短くなり、一定処理を行うためにCPUコア101, 102に割当てられた時間が短くなるので、CPUコア101, 102の処理負荷が高くなる。このことから、同期PWM制御にて車両駆動用電動モータの制御を行うシステムでは、キャリア周波数（及びキャリア周波数を制御するカウンタ）を車両情報として用いてCPUコア101, 102の処理負荷を予測することができる。この目的のため、第1の具体例では、タイマ18（図4参照）とし

てキャリア周波数用タイマが使用される。また、図3に示すフローチャートでは、車両情報としてキャリア周波数制御カウンタが入力され（ステップ300）、キャリア周波数からC P Uコア101, 102の処理負荷が予測され（ステップ301）、予測した処理負荷が所定レベルよりも高いか否かが判定される（ステップ302）。尚、図3のステップ301及び302の処理として、キャリア周波数が所定閾値（所定周波数）よりも高いか否かが判定されてもよい。この場合も同様の考え方に基づいて、所定閾値は、1つのC P Uコアで処理可能な処理負荷の最大レベルに適合されてよい。例えば、C P Uコア101, 102の動作周波数を64MHzとしたとき、PWM信号のキャリア周波数が5kHz未満の場合は、1つのC P Uコアで処理可能な処理負荷であると判定して、故障診断を行うこととし、PWM信号のキャリア周波数が5kHz以上の場合は、1つのC P Uコアで処理不能な処理負荷であると判定して、故障診断を行わないようにしてよい。

- [0061] 尚、この第1の具体例において、PWM信号のキャリア周波数は、車両駆動用電動モータの他の制御情報（例えば、車両駆動用電動モータの回転数、トルク、回転方向等）に基づいて算出・推定されてもよい。
- [0062] 第2の具体例は、車両情報として、ナビゲーション（マルチメディア）システムにおけるユーザの操作情報（又はユーザの操作に基づく当該システムの制御状態の情報）を用いて、C P Uコア101, 102の処理負荷を予測する方法に関する。ここでは、C P Uコア101, 102は、ナビゲーションシステムの制御を実行する。即ち、C P Uコア101, 102は、ナビゲーションシステムのE C Uを構成する。
- [0063] ナビゲーションシステムでは、ユーザによる操作（例えばタッチパネルに対する画面操作）により同時に動作させるアプリが分かるため、これに基づいてC P Uコア101, 102の処理負荷を予測することができる。例えば、ナビゲーションシステムにおいて画面操作により同時動作が可能な複数のアプリとして、ナビゲーションのルーティング機能、CDのリッピング、D T V（デジタルテレビ）の受信と音声出力等がある。ここで、同時に動作す

るアプリの数が多くなるにつれて、C P Uコア101, 102の処理負荷が高くなる。このことから、ユーザによる操作により複数のアプリの同時動作が可能なナビゲーションシステムでは、ユーザによる操作情報（及びそれに基づいて同時動作するアプリ数）を車両情報として用いてC P Uコア101, 102の処理負荷を予測することができる。この目的のため、第2の具体例では、マルチコアC P U12には、ナビゲーションの操作情報が入力される。また、図3に示すフローチャートでは、車両情報として操作情報が入力され（ステップ300）、操作情報からC P Uコア101, 102の処理負荷として、同時動作するアプリ数が予測され（ステップ301）、予測した処理負荷が所定レベル（所定アプリ数）よりも高いか否かが判定される（ステップ302）。この場合も同様の考え方に基づいて、所定アプリ数は、1つのC P Uコアで処理可能な処理負荷の最大レベル（最大アプリ数）に適合されてよい。例えば、C P Uコア101, 102の動作周波数を400MHzとしたとき、所定アプリ数は、2個であってよい。この場合、ユーザによる操作により例えばルーティングのみを処理する場合は、1つのC P Uコアで処理可能な処理負荷であると判定して、故障診断を行い、ユーザによる操作により例えばルーティングとリッピングを同時動作させる場合は、1つのC P Uコアで処理不能な処理負荷であると判定して、故障診断を行わない。

- [0064] 第3の具体例は、車両情報として車速を用いて、C P Uコア101, 102の処理負荷を予測する方法に関する。ここでは、C P Uコア101, 102は、車載周辺監視カメラを用いた周辺監視システムの制御を実行する。即ち、C P Uコア101, 102は、周辺監視システムのECUを構成する。周辺監視システムとは、典型的には、車載周辺監視カメラで取得した画像から所定の対象物（例えば白線等の道路標示、道路標識、周辺車両のような障害物等）を画像認識処理するシステムである。尚、画像認識結果は、例えば障害物等の存在等をユーザに報知する警報システムや障害物等と車両の位置関係等に応じて車両制御を行う車両制御システム等で利用されてもよい。
- [0065] 周辺監視システムでは、車速が速くなるにつれて、障害物等の対象物の探

索範囲が広がり、C P Uコア101, 102の処理負荷が高くなる。即ち、車両の速度が高くなるにつれて、所定時間内に移動する車両の距離が長くなるので、その分だけ長い距離の範囲を監視する必要が生ずる。例えば図13に概念的に示すように、低速時の探索範囲は1m先から10m先の画像でよいのに対して、高速時の探索範囲は20m先の画像まで必要となる。このことから、車速に応じて画像処理範囲を変化させる周辺監視システムでは、車速情報を車両情報として用いてC P Uコア101, 102の処理負荷を予測することができる。この目的のため、第3の具体例では、マルチコアC P U12には、車速情報が入力される。車速情報は、車輪速センサからの車速情報であってもよいし、トランスミッションの出力軸の回転数等の車速に関連する情報が使用されてもよい。また、図3に示すフローチャートでは、車両情報として車速情報が入力され（ステップ300）、車速情報からC P Uコア101, 102の処理負荷として、対象物の探索範囲が予測され（ステップ301）、予測した処理負荷が所定レベル（所定探索範囲）よりも高いか否かが判定される（ステップ302）。この場合も同様の考え方に基づいて、所定探索範囲は、1つのC P Uコアで処理可能な処理負荷の最大レベル（最大探索範囲）に適合されてよい。或いは、図3のステップ301及び302の処理として、車速が所定閾値（所定車速）よりも高いか否かが判定されてもよい。この場合も同様の考え方に基づいて、所定車速は、1つのC P Uコアで処理可能な処理負荷の最大レベルに適合されてよい。例えば、車速が所定車速未満の低速領域である場合は、1つのC P Uコアで処理可能な処理負荷であると判定して、故障診断を行うこととし、車速が所定車速以上の中速領域又は高速領域である場合は、1つのC P Uコアで処理不能な処理負荷であると判定して、故障診断を行わないようにしてよい。

[0066] 第4の具体例は、車両情報としてエンジン回転数を用いて、C P Uコア101, 102の処理負荷を予測する方法に関する。ここでは、C P Uコア101, 102は、エンジン制御システムの制御を実行する。即ち、C P Uコア101, 102は、E F I・E C Uを構成する。エンジン制御システムは

、燃料噴射や点火等の各種エンジン制御を行うシステムである。

[0067] エンジン制御システムでは、エンジン回転数が高くなるにつれて、C P Uコア101, 102の処理負荷が高くなる。このことから、エンジン制御システムでは、エンジン回転数を車両情報として用いてC P Uコア101, 102の処理負荷を予測することができる。この目的のため、第4の具体例では、マルチコアC P U12には、エンジン回転数が入力される。また、図3に示すフローチャートでは、車両情報としてエンジン回転数が入力され（ステップ300）、エンジン回転数からC P Uコア101, 102の処理負荷が予測され（ステップ301）、予測した処理負荷が所定レベルよりも高いか否かが判定される（ステップ302）。尚、図3のステップ301及び302の処理として、エンジン回転数が所定閾値（所定回転数）よりも高いか否かが判定されてもよい。即ち、エンジン回転数が所定回転数未満である場合は、1つのC P Uコアで処理可能な処理負荷であると判定して、故障診断を行うこととし、エンジン回転数が所定回転数以上である場合は、1つのC P Uコアで処理不能な処理負荷であると判定して、故障診断を行わないようにしてよい。この場合も同様の考え方に基づいて、所定閾値（所定回転数）は、1つのC P Uコアで処理可能な処理負荷の最大レベルに基づいて適合されてよい。

[0068] 図14は、本発明のその他の一実施例（実施例2）による故障診断装置2のハードウェア構成を示す図である。

[0069] 本実施例の故障診断装置2は、上述した実施例の故障診断装置1に対して、C P Uコアの数が主に異なる。即ち、本実施例の故障診断装置2は、4つのC P Uコア101, 102, 103, 104を有する。このようにマルチコアC P U12においてC P Uコアの数は2以上の任意の数であってもよい。

[0070] 本実施例においても、図3に示すフローチャートと同様の態様で故障診断テスト制御が実行されてよい。この場合、ステップ302の判定においては、所定レベルは、3つのC P Uコアで処理可能な処理負荷の最大レベルに対

応してもよい。即ち、所定レベルは、1つのCPUコアで故障診断を行い他の3つのCPUコアで通常処理を行うことが可能な処理負荷の最大レベルに対応してもよい。従って、所定レベルは、同様に、CPUコア101, 102, 103, 104の個々のCPUコアの能力（性能）に応じて適合されてよい。尚、CPUコア101, 102, 103, 104の能力は、一般的に、それぞれ同一である。ステップ302の判定において、予測された処理負荷が所定レベルよりも高い場合は、ステップ304に進み、同様に、4つのCPUコア101, 102, 103, 104がSMP動作を行う。他方、予測された処理負荷が所定レベルよりも高い場合は、ステップ310に進み、故障診断対象のCPU番号が呼び出される。本実施例では、CPU番号レジスタ204におけるCPU番号の更新は、例えば「0」、「1」、「2」、「3」の順序で周期的に行われてよい。CPU番号「0」、「1」、「2」、「3」は、CPUコア101, 102, 103, 104にそれぞれ対応する。尚、CPU番号の更新の順序は任意であり、例えばCPU番号「1」、「0」、「2」、「3」の順序で周期的に行われてもよい。

[0071] また、本実施例においても、データ制御部200は、図4に示した構成と同様であってよい。但し、図4のCPUコア101, 102に対して設けられるマルチプレクサ201, 202と同様に、CPUコア103, 104に対してそれぞれマルチプレクサが追加される。また、CPU番号レジスタ204は、4つのCPUコア101, 102, 103, 104に対応するため2ビットにされる。また、マルチプレクサ205は、CPUコア101, 102のみからの入力だけでなく、CPUコア103, 104からの入力をも受け付けるように変更される。

[0072] 尚、本実施例2では、4つのCPUコア101, 102, 103, 104が1つずつ故障診断を行うように構成されているが、4つのCPUコア101, 102, 103, 104のうちの1つ、2つ又は3つのCPUコアが選択的に並列で故障診断を行うことが可能な構成であってもよい。例えば、予測された処理負荷が、1つのCPUコアで処理可能なレベルである場合は、

残りの 3 つの C P U コアが並列で故障診断を行い、予測された処理負荷が、1 つの C P U コアで処理不能であるが 2 つの C P U コアで処理可能なレベルである場合は、残りの 2 つの C P U コアが並列で故障診断を行い、予測された処理負荷が、2 つの C P U コアでも処理不能であるが 3 つの C P U コアならば処理可能なレベルである場合は、残りの 1 つの C P U コアが並列で故障診断を行うこととしてもよい。この場合は、故障診断ブロック 210 (図 4 参照) が 3 つ用意されてよい。

- [0073] 以上、本発明の好ましい実施例について詳説したが、本発明は、上述した実施例に制限されることはなく、本発明の範囲を逸脱することなく、上述した実施例に種々の変形及び置換を加えることができる。
- [0074] 例えば、上述した実施例では、車両情報から C P U コアの処理負荷を予測し、予測結果に基づいて故障診断を行うか否かを判定しているが、車両情報から C P U コアの処理負荷を検出し、検出結果に基づいて故障診断を行うか否かを判定してもよい。例えば、故障診断処理に要する時間 (図 3 のステップ 310 からステップ 328 までの処理に要しうる時間の最大値) が、C P U コアの処理負荷の変動が生ずる時間よりも有意に小さい場合には、リアルタイムに検出される C P U コアの処理負荷に応じて故障診断を行うか否かを判定してもよい。尚、マルチコア C P U 12 の処理負荷を検出するために、車両情報として、PWM 信号のキャリア周波数 (現在値) 等の上述した情報の他、マルチコア C P U 12 における現時点から所定時間前までの関数の呼び出し数等が使用されてもよい。
- [0075] また、上述した実施例では、予測したマルチコア C P U 12 の処理負荷が所定レベルより大きいか否かに応じて特定の 1 つの又は 2 つ以上の C P U コアの故障診断を行うか否かを判定しているが、予測したマルチコア C P U 12 の処理負荷に応じて、より多様な態様で C P U コアの故障診断態様を変更してもよい。
- [0076] 例えば、予測したマルチコア C P U 12 の処理負荷に応じて、特定の 1 つの C P U コアの故障診断の実行部分を可変してもよい。例えば、故障診断が

複数個の故障診断テスト用のデータからなる場合、予測したマルチコアCPU 12の処理負荷に応じて、特定の1つのCPUコアの故障診断を実行する際ににおける処理する故障診断テスト用のデータの数を調整してもよい。その場合、次の機会（複数の機会も可）で、特定の1つのCPUコアの故障診断として、特定の1つのCPUコアが残りの故障診断テスト用のデータを処理すればよい。

[0077] 図15は、処理負荷に応じてCPUコアの故障診断の実行部分を可変する故障診断テスト制御の一例を示す。図15に示すフローチャートでは、ステップ1500にて車両情報が入力され（図3のステップ300と同様）、ステップ1501にて車両情報からCPUコア101、102の処理負荷が予測され（図3のステップ301と同様）、ステップ1502にて予測した処理負荷が所定レベルよりも高いか否かが判定される（図3のステップ302と同様）。予測した処理負荷が所定レベルよりも高い場合は、ステップ1504にて通常処理が実行される（図3のステップ304、306、308と同様）。他方、予測した処理負荷が所定レベルよりも小さい場合は、ステップ1506にて、予測した処理負荷と所定レベルとの差が小さいか否かが判定される。即ち、予測した処理負荷が、その後の短時間（負荷予測期間Tよりも短い時間）で、所定レベルよりも高くなりうるほど所定レベルに近接しているか否かが判定される。予測した処理負荷と所定レベルとの差が小さい場合は、その後の負荷予測期間Tよりも短い時間で所定レベルよりも高くなりうると判断し、ステップ1510に進み、予測した処理負荷と所定レベルとの差が大きい場合は、ステップ1508に進む。ステップ1508では、故障診断の全部が実行される。故障診断の全部の実行態様については、図3のステップ310乃至328と同様であってよい。他方、ステップ1510では、負荷予測期間Tよりも短い時間で故障診断が完了するように、故障診断の一部のみが実行される。故障診断の一部の実行態様については、図3のステップ310乃至328と同様であってよい。但し、この場合、ステップ328では、故障診断の残りの部分が実行されるまでCPU番号の更新が実

行されない。故障診断の残りの部分は、その後再び予測処理負荷が所定レベルを下回る機会（複数の機会も可）にて実行されてよい。

[0078] 尚、図15に示した故障診断テスト制御では、実質的に、2つの閾値で予測した処理負荷を評価している。即ち、予測した処理負荷が第1の閾値（所定レベル）よりも高い場合は、故障診断を行わず、予測した処理負荷が第1の閾値（所定レベル）よりも低いが第2の閾値（第2の所定レベル）よりも高い場合は、故障診断の一部のみを実行し、予測した処理負荷が第2の閾値よりも低い場合は、故障診断の全部（1つのCPUコアに対して全部）を実行している。しかしながら、3つ以上の閾値により細分化してもよい。或いは、図3に示した故障診断テスト制御の変形例として、予測した処理負荷が所定レベルよりも高い場合でも、可能な範囲の故障診断（故障診断の一部）のみを実行することとしてもよい。

[0079] また、予測したマルチコアCPU12の処理負荷に応じて、故障診断処理が実行可能となるようにタスク調整を行ってもよい。例えば図3に示す故障診断テスト制御では、テストモード中に、AMP動作に切り替わることでCPUコア101、102間のタスク調整が実質的に実現されている。かかるタスク調整に加えて、予測したマルチコアCPU12の処理負荷が所定レベルよりも高くなった場合に、タスク調整を行って、故障診断処理が実行可能となるようにしてもよい。

[0080] 図16は、処理負荷に応じてタスク調整を行いCPUコアの故障診断の実行可能な機会を増加させる故障診断テスト制御の一例を示す。

[0081] 図16に示すフローチャートでは、ステップ1600にて車両情報が入力され（図3のステップ300と同様）、ステップ1601にて車両情報からCPUコア101、102の処理負荷が予測され（図3のステップ301と同様）、ステップ1602にて予測した処理負荷が所定レベルよりも高いか否かが判定される（図3のステップ302と同様）。予測した処理負荷が所定レベルよりも高い場合は、ステップ1604にてタスク調整が実行される。このタスク調整では、例えば優先度の低いタスクが負荷予測期間Tよりも

後に実行されるようにタスク調整が行われてもよい（即ち、優先度の低いタスクが後回しされる）。ステップ1606では、ステップ1604でのタスク調整を反映させてCPUコア101、102の処理負荷が再度予測され（図3のステップ301と同様）、予測した処理負荷が所定レベルよりも高いか否かが再度判定される（図3のステップ302と同様）。予測した処理負荷が依然として所定レベルよりも高い場合は、ステップ1608にて通常処理が実行される（図3のステップ304、306、308と同様）。但し、この場合は、ステップ1604にて行ったタスク調整をキャンセルすることとしてよい。他方、予測した処理負荷が所定レベルよりも小さい場合は、ステップ1610にて故障診断が実行される（図3のステップ310乃至328と同様）。

[0082] また、予測したマルチコアCPU12の処理負荷に対して、実際のマルチコアCPU12の処理負荷が異なる場合も有りうる。そこで、テストモードの間に、故障診断対象でないCPUコア102又は101により処理できないほど処理負荷が高くなった場合には、故障診断対象のCPUコアのテストモードが中止されてもよい。

[0083] 図17は、テストモードの間に処理負荷が高くなった場合に故障診断の中斷を行う故障診断テスト制御の一例を示す。

[0084] 図17に示すフローチャートは、ステップ1723とステップ1725の処理が図3の故障診断テスト制御に追加された点が主に異なる。ステップ1723では、テストモードの間に、新たにが入力された最新の車両情報に基づいて再度CPUコア101、102の処理負荷が予測／検出され、予測／検出された処理負荷が所定レベルよりも高い場合は、ステップ1725に進み、故障診断対象のCPUコアのテストモードが中止・中断される。この場合、故障診断対象のCPUコアのモードはテストモードから通常モードに変更され、ステップ304の処理に移行する。尚、その後、再びステップ302にて予測される処理負荷が所定レベルよりも低くなった場合には、ステップ320からの処理にて故障診断対象のCPUコアの故障診断が再開（最初

から再開又は途中から再開) される。

[0085] また、上述した実施例は、マルチコア C P U 1 2 に関するものであるが、複数のマイコンからなるマルチプロセッサシステムにも拡張して適用することができる。即ち、複数のマイコンのそれぞれの C P U を、上述の実施例における C P U コア 1 0 1, 1 0 2 (又は C P U コア 1 0 1, 1 0 2, 1 0 3, 1 0 4) と同様に扱ってもよい。この場合、複数のマイコン間には、データ制御部 2 0 0 (図 4 等参照) に相当する構成 (複数のマイコン間のデータ連携を行う部) が設けられる。また、複数のマイコン間では、故障診断時に必要なタスクの調整が実行される。

[0086] 最後に、以上の説明に関して更に以下の付記を開示する。

(付記 1)

マルチコアプロセッサにおける複数の C P U の故障診断を行う故障診断装置であって、

前記複数の C P U の処理に関連する車両情報に基づいて、前記複数の C P U の全体として処理負荷を予測又は検出し、該処理負荷の予測結果又は検出結果に応じて、前記複数の C P U の動作モードを S M P と A M P との間で動的に切換え、A M P に切り換えたときに、前記複数の C P U のうちの特定の C P U が自身の故障診断を行うことを特徴とする、故障診断装置。

(付記 2)

前記予測又は検出した処理負荷が所定基準よりも低い場合に、S M P から A M P に切り換える、付記 1 に記載の故障診断装置。

(付記 3)

A M P の動作モードでは、前記複数の C P U のうちの特定の C P U 以外の C P U は、通常処理を実行する、付記 1 又は 2 に記載の故障診断装置。

(付記 4)

前記予測又は検出した処理負荷が所定基準よりも高い場合は、S M P に動作モードが維持される、付記 1 ~ 3 のうちのいずれかに記載の故障診断装置。

。

(付記 5)

前記予測又は検出した処理負荷に応じて、前記特定のCPUが行うべき故障診断の処理内容（例えば、故障診断テスト用のデータの処理個数や処理範囲）を可変する、付記1～4のうちのいずれかに記載の故障診断装置。

(付記 6)

前記特定のCPUは、1つのCPU又は2つ以上のCPUである、付記1～5のうちのいずれかに記載の故障診断装置。

(付記 7)

複数のCPUの故障診断を行う故障診断装置であって、

前記複数のCPUの処理に関連する車両情報に基づいて、前記複数のCPUの全体としての処理負荷を予測又は検出し、該処理負荷の予測結果又は検出結果に応じて、故障診断の実行態様を変更することを特徴とする、故障診断装置。

(付記 8)

前記複数のCPUは、マルチコアプロセッサにおける複数のCPUである、付記7に記載の故障診断装置。

(付記 9)

前記複数のCPUの全体としての処理負荷が所定基準よりも低いことが予測又は検出された場合に、故障診断を行う、付記7又は8に記載の故障診断装置。

(付記 10)

前記複数のCPUの全体としての処理負荷が所定基準よりも高いことが予測又は検出された場合に、少なくとも一部の故障診断を行わない（即ち、一部のみを実施する又は全部実施しない）、付記7～8のうちのいずれかに記載の故障診断装置。

(付記 11)

故障診断を行っている間に、前記複数のCPUの全体としての処理負荷が所定基準よりも高くなった場合に、故障診断の少なくとも一部を中止する、

付記 1 ~ 1 0 のうちのいずれかに記載の故障診断装置。

(付記 1 2)

前記複数の C P U は、 C P U 每に自身の故障診断を行う、付記 1 ~ 1 1 のうちのいずれかに記載の故障診断装置。

(付記 1 3)

故障診断を行う場合、前記複数の C P U 間でタスクの調整を行う、付記 1 ~ 1 2 のうちのいずれかに記載の故障診断装置。

(付記 1 4)

前記複数の C P U は、駆動モータの制御を行う C P U であり、
前記車両情報は、前記駆動モータの回転数である、付記 1 ~ 1 3 のうちのいずれかに記載の故障診断装置。

(付記 1 5)

前記複数の C P U は、エンジン制御を行う C P U であり、
前記車両情報は、エンジン回転数である、付記 1 ~ 1 4 のうちのいずれかに記載の故障診断装置。

(付記 1 6)

前記車両情報は、車速である、付記 1 ~ 1 5 のうちのいずれかに記載の故障診断装置。

(付記 1 7)

前記複数の C P U は、周辺監視システムの制御を行う C P U であり、
前記車両情報は、車速である、付記 1 ~ 1 5 のうちのいずれかに記載の故障診断装置。

(付記 1 8)

前記複数の C P U は、ナビゲーションシステムの制御を行う C P U であり、
前記車両情報は、ナビゲーションシステムに対するユーザの操作情報である、付記 1 ~ 1 7 のうちのいずれかに記載の故障診断装置。

(付記 1 9)

前記複数のCPUは、複数のマイコンにそれぞれ含まれるCPUから構成される、付記7、9～18のうちのいずれかに記載の故障診断装置。

符号の説明

- [0087] 1, 2 故障診断装置
- 12 マルチコアCPU
- 13 データアクセスコントローラ
- 14 メモリコントローラ
- 15 記憶媒体
- 16 記憶媒体
- 18 タイマ
- 20 データバス
- 22 データバス
- 101, 102, 103, 104 CPUコア
- 200 データ制御部
- 201 マルチプレクサ
- 202 マルチプレクサ
- 203 モード制御部
- 204 CPU番号レジスタ
- 205 マルチプレクサ
- 210 故障診断ブロック
- 212, 214 記憶部
- 216 比較器

請求の範囲

- [請求項1] 複数のCPUの故障診断を行う故障診断装置であって、前記複数のCPUの処理に関する車両情報に基づいて、前記複数のCPUの全体としての処理負荷を予測又は検出し、該処理負荷の予測結果又は検出結果に応じて、故障診断の実行態様を変更することを特徴とする、故障診断装置。
- [請求項2] 前記複数のCPUは、マルチコアプロセッサにおける複数のCPUである、請求項1に記載の故障診断装置。
- [請求項3] 前記複数のCPUの全体としての処理負荷が所定基準よりも低いことが予測又は検出された場合に、故障診断を行う、請求項1に記載の故障診断装置。
- [請求項4] 前記複数のCPUの全体としての処理負荷が所定基準よりも高いことが予測又は検出された場合に、少なくとも一部の故障診断を行わない、請求項1に記載の故障診断装置。
- [請求項5] 故障診断を行っている間に、前記複数のCPUの全体としての処理負荷が所定基準よりも高くなつた場合に、故障診断の少なくとも一部を中止する、請求項1に記載の故障診断装置。
- [請求項6] 前記複数のCPUは、CPU毎に自身の故障診断を行う、請求項1に記載の故障診断装置。
- [請求項7] 故障診断を行う場合、前記複数のCPU間でタスクの調整を行う、請求項1に記載の故障診断装置。
- [請求項8] 前記複数のCPUは、駆動モータの制御を行うCPUであり、前記車両情報は、前記駆動モータの回転数である、請求項1に記載の故障診断装置。
- [請求項9] 前記複数のCPUは、エンジン制御を行うCPUであり、前記車両情報は、エンジン回転数である、請求項1に記載の故障診断装置。
- [請求項10] 前記車両情報は、車速である、請求項1に記載の故障診断装置。

[請求項11] マルチコアプロセッサにおける複数のCPUの故障診断を行う故障診断方法であって、

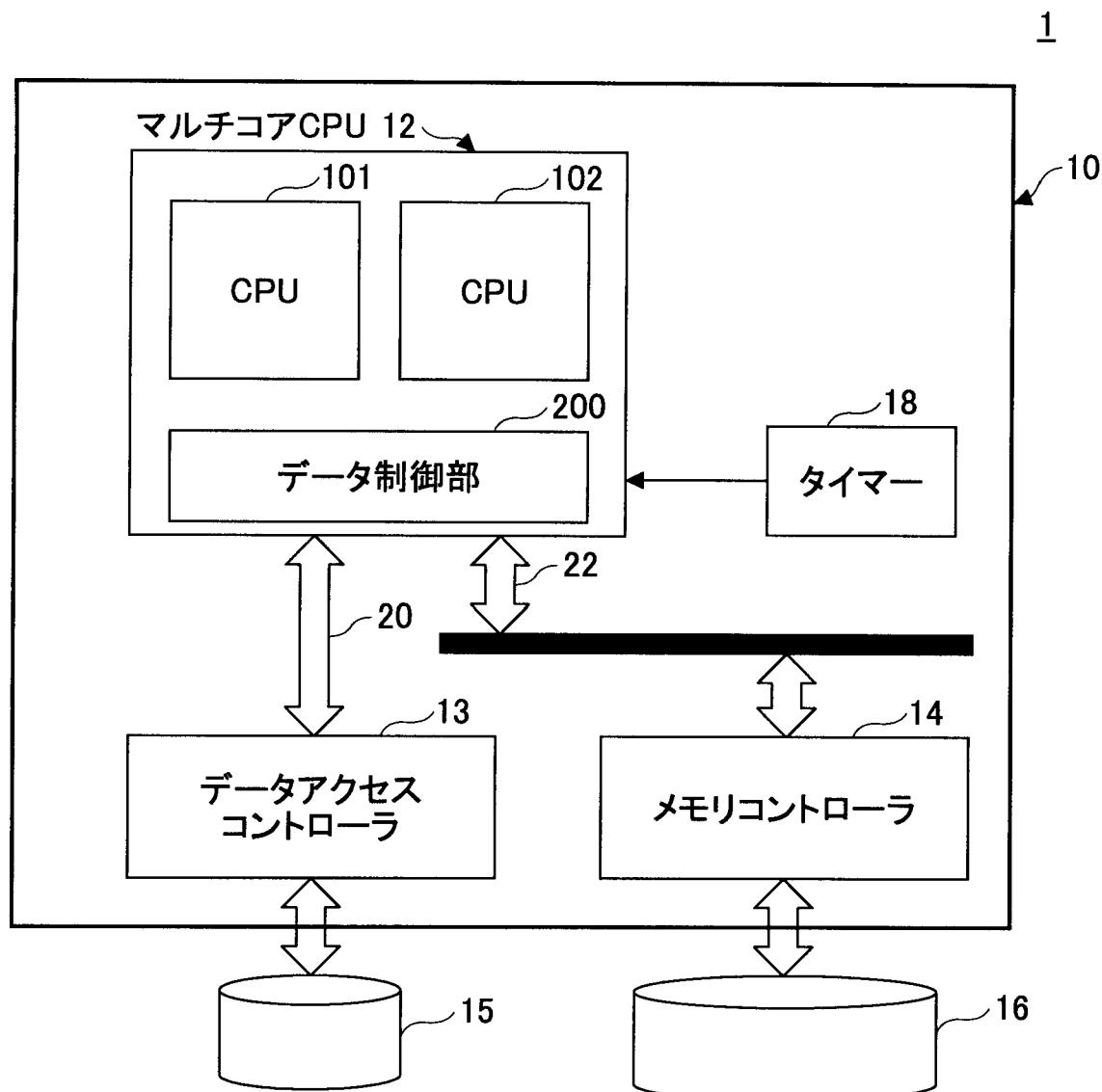
前記複数のCPUの処理に関する車両情報を入力するステップと

、

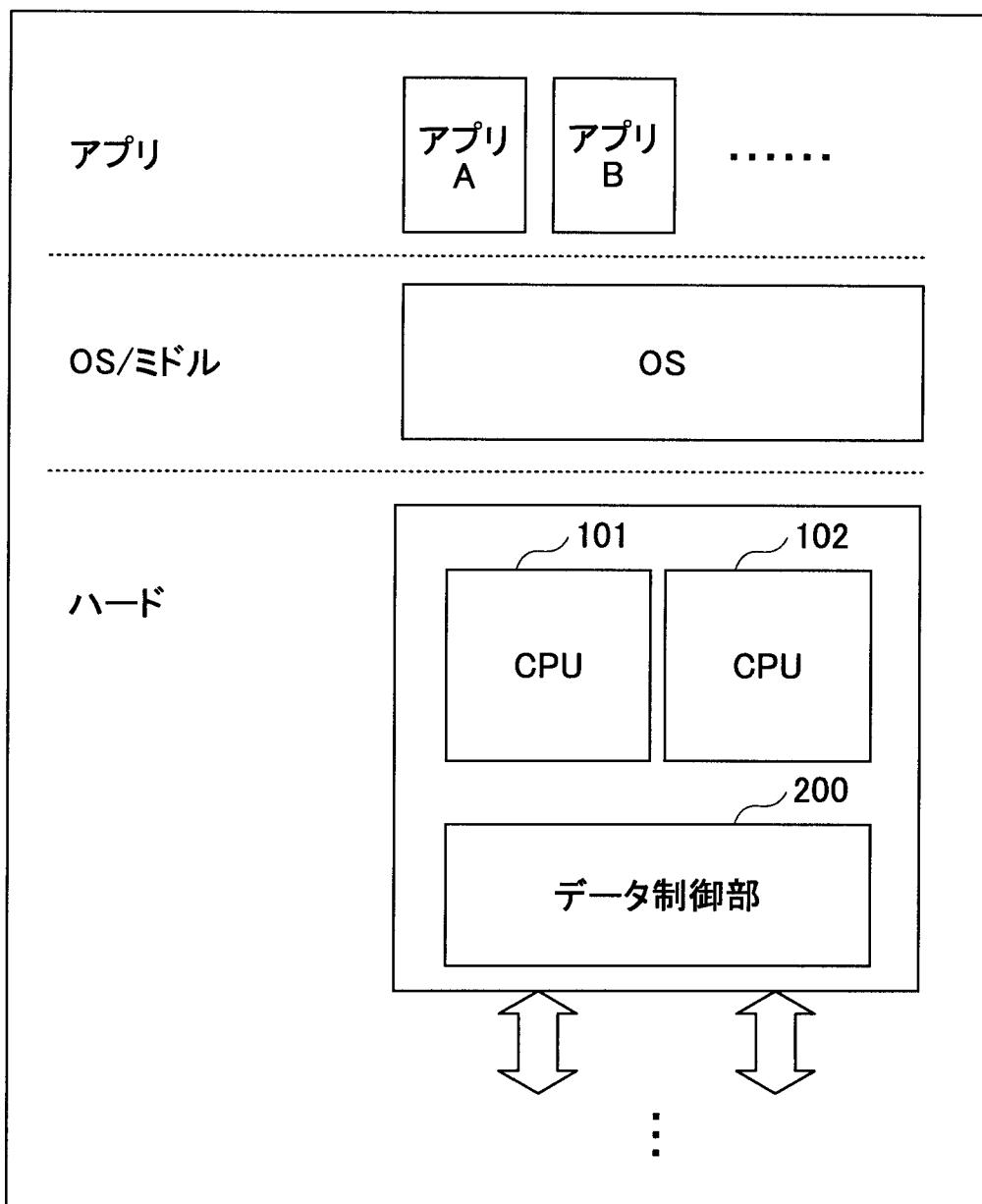
前記車両情報に基づいて、前記複数のCPUの全体としての処理負荷を予測又は検出する予測ステップと、

前記予測ステップにおける処理負荷の予測結果又は検出結果に応じて、故障診断の実行態様を変更するステップとを含むことを特徴とする、故障診断方法。

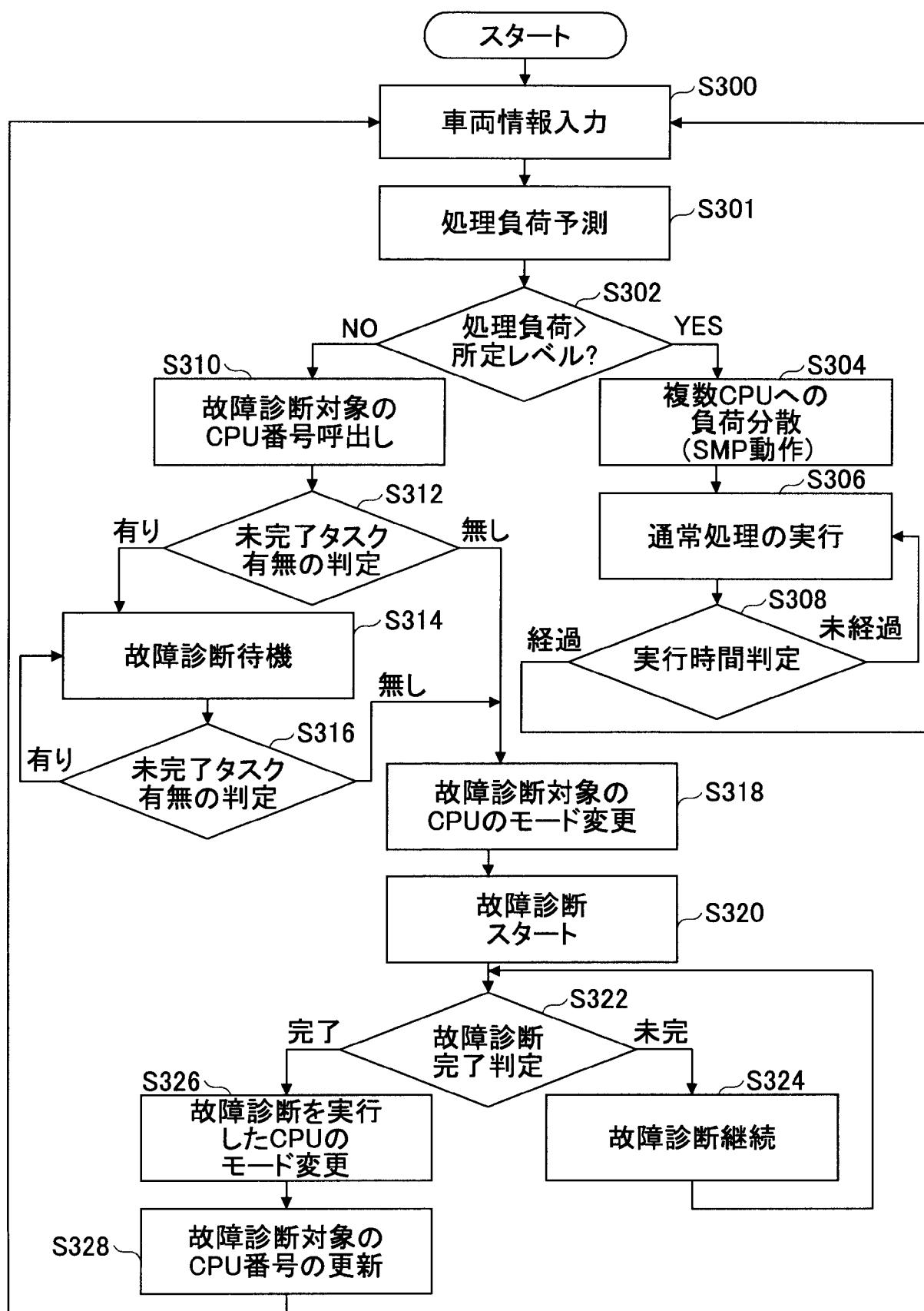
[図1]



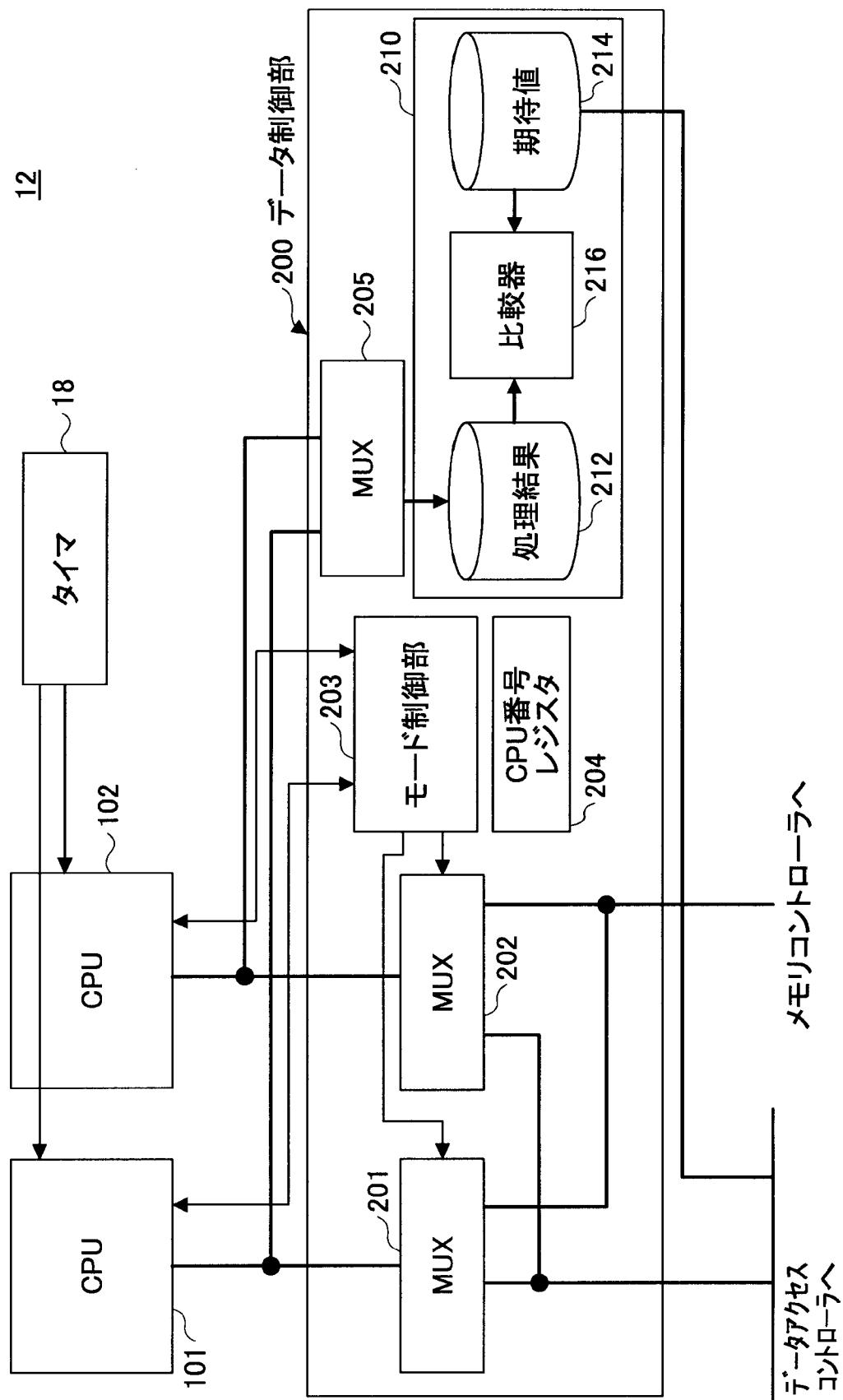
[図2]



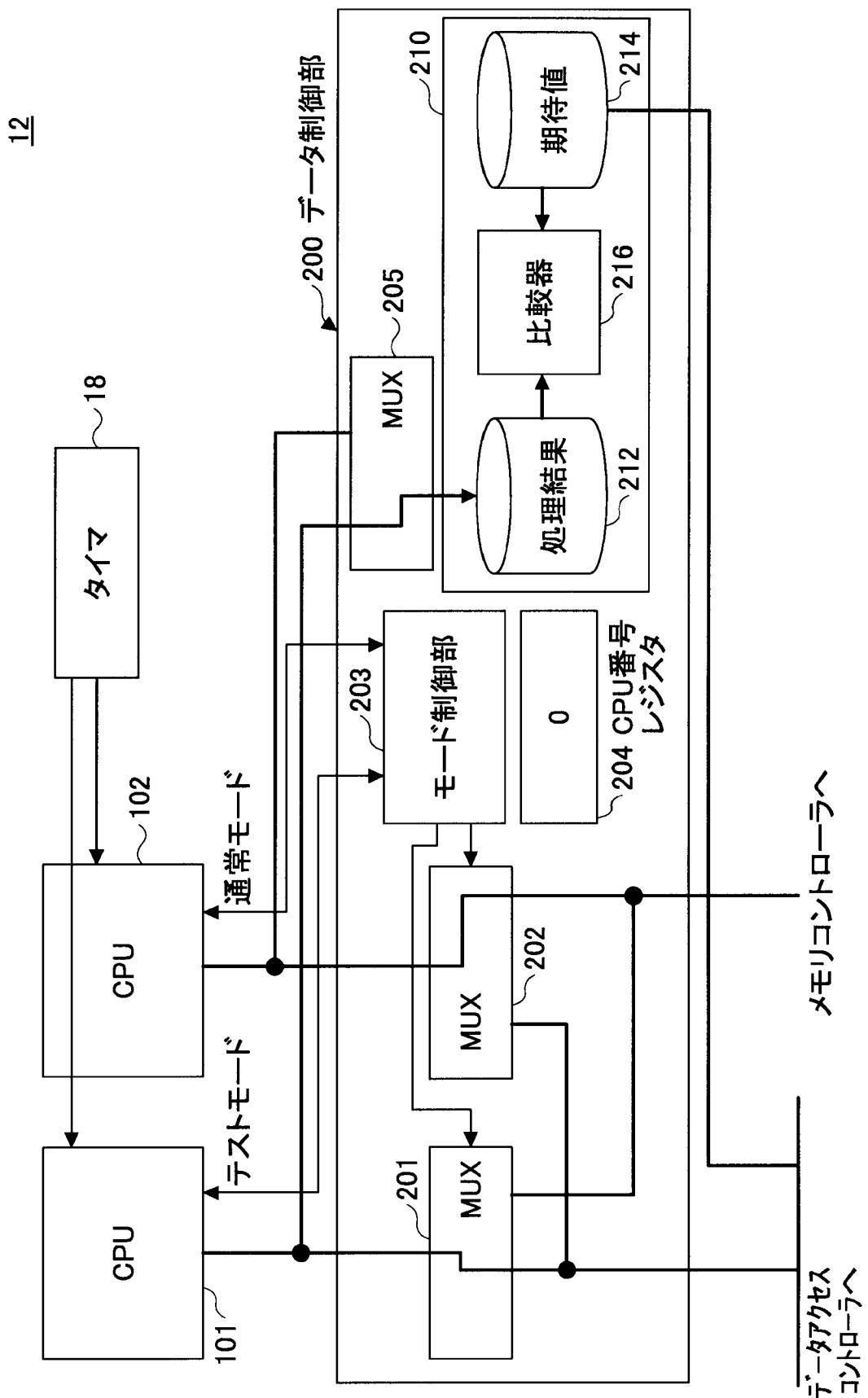
[図3]



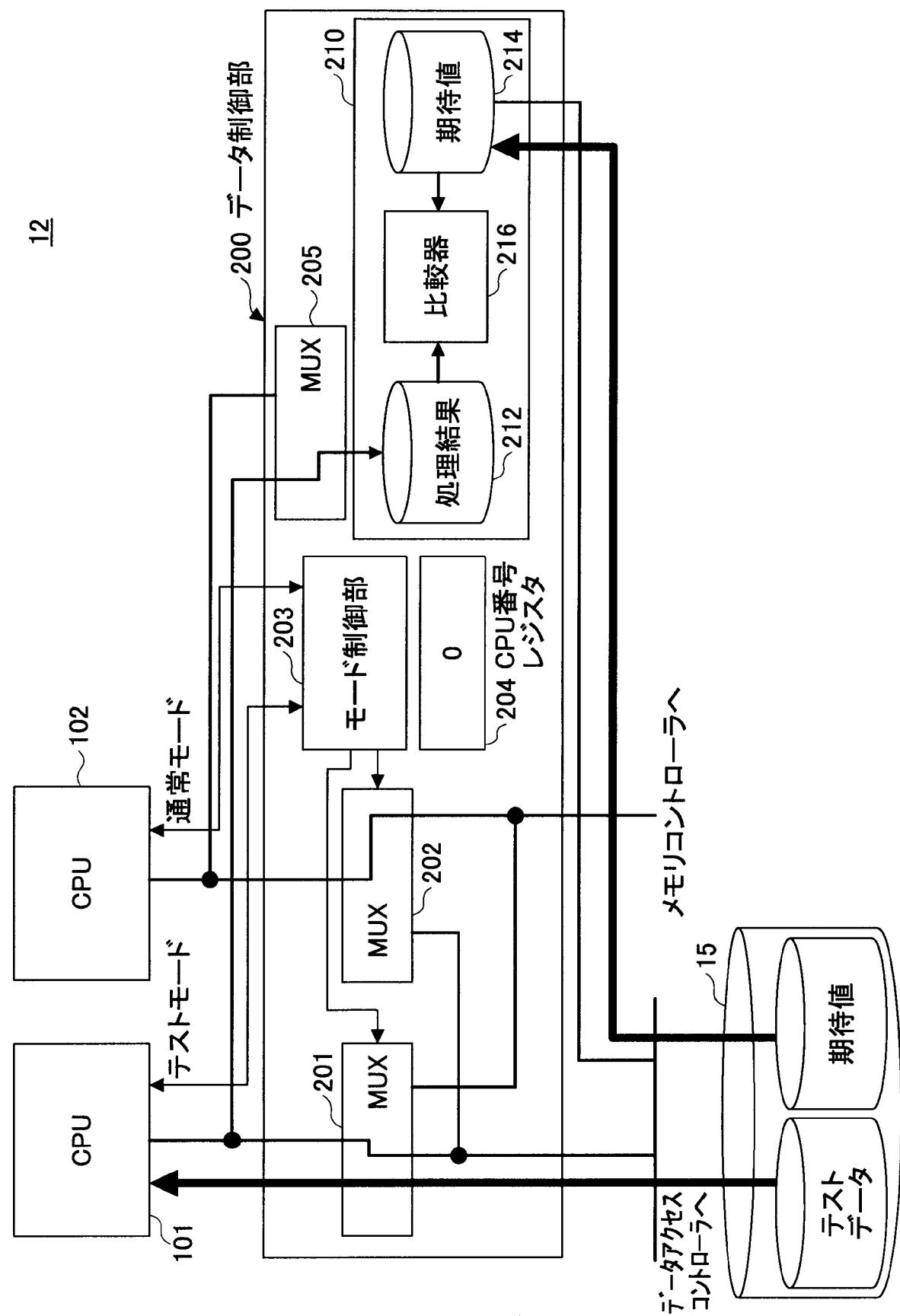
[図4]



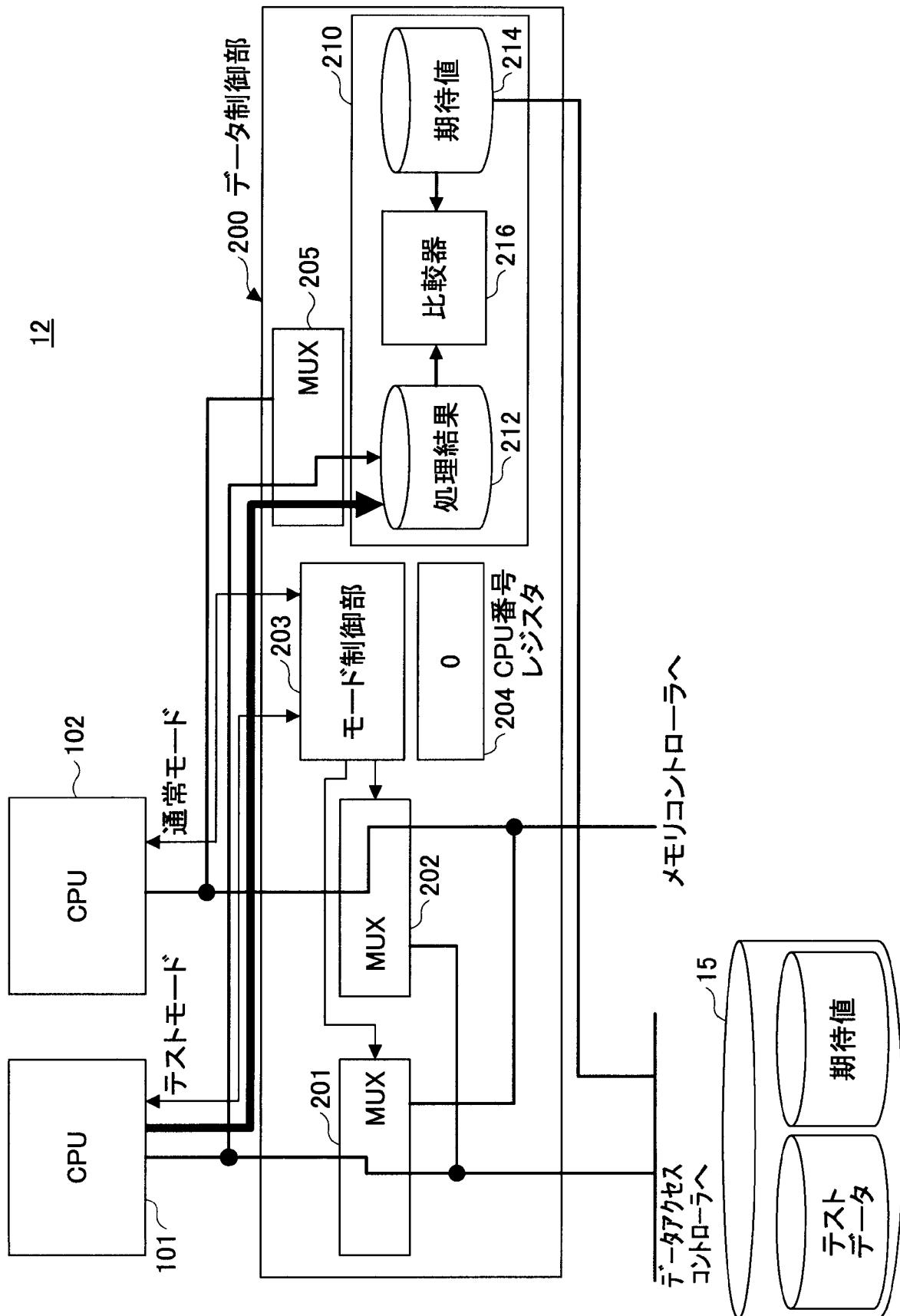
[図5]



[図6]

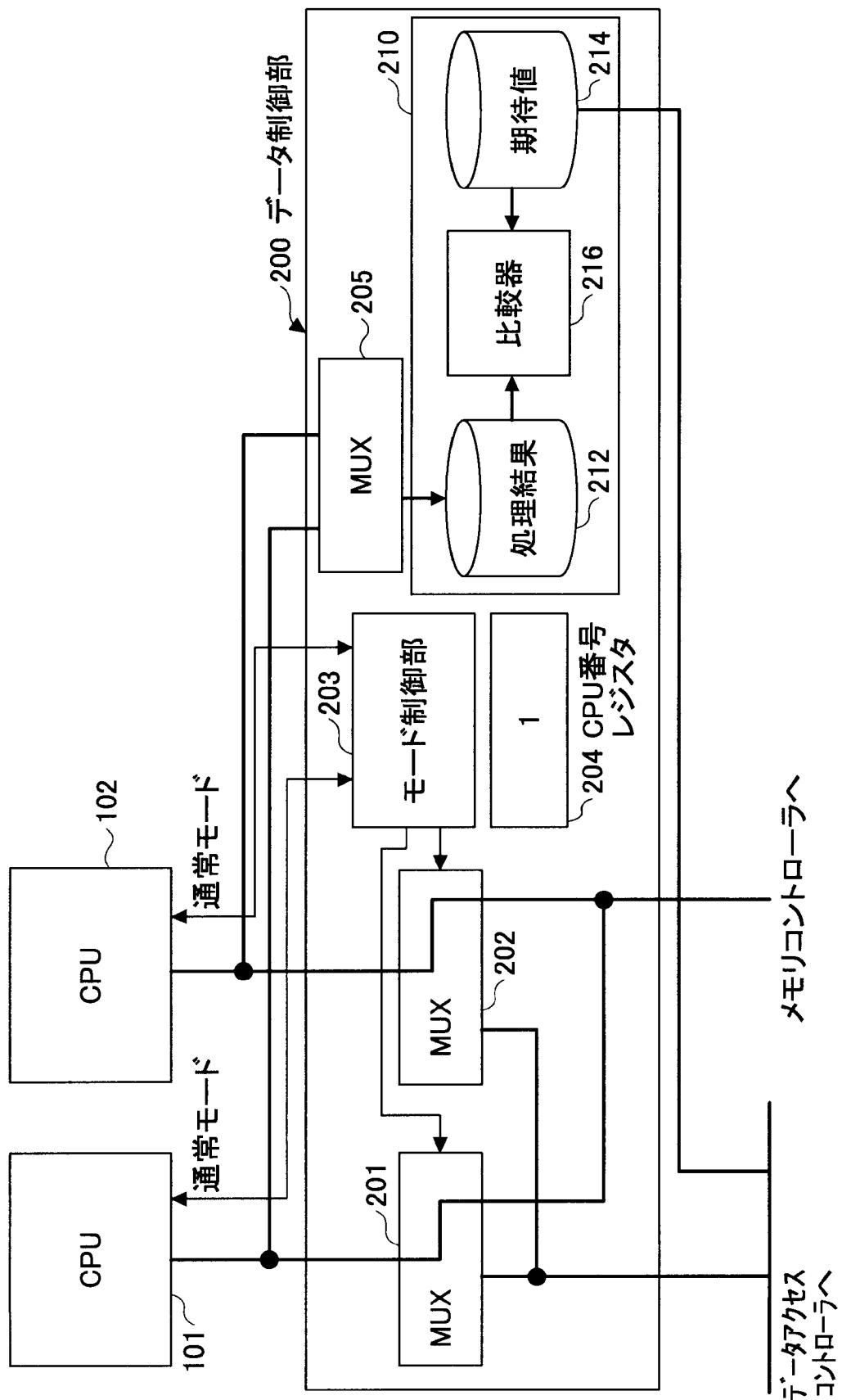


[図7]

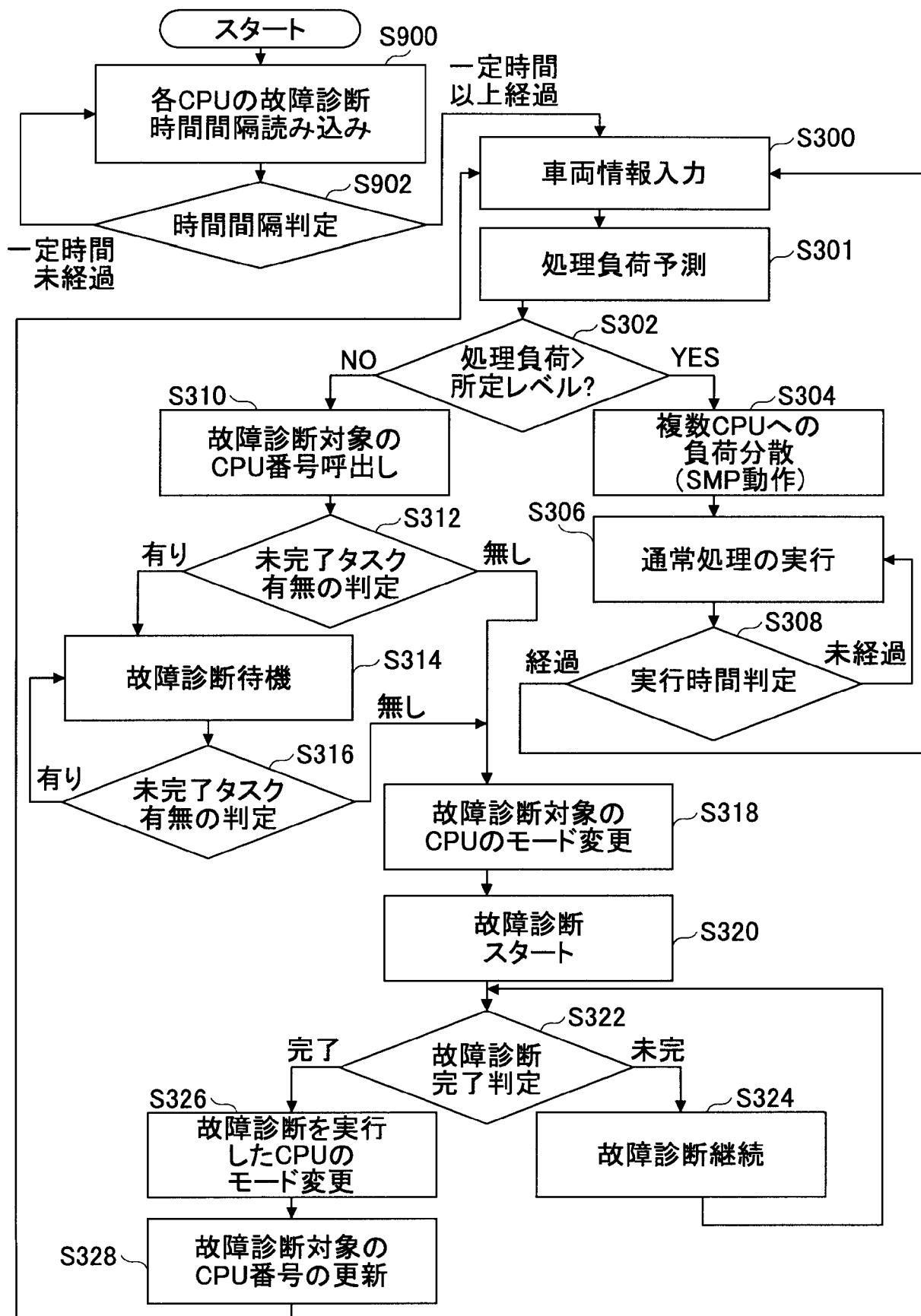


[図8]

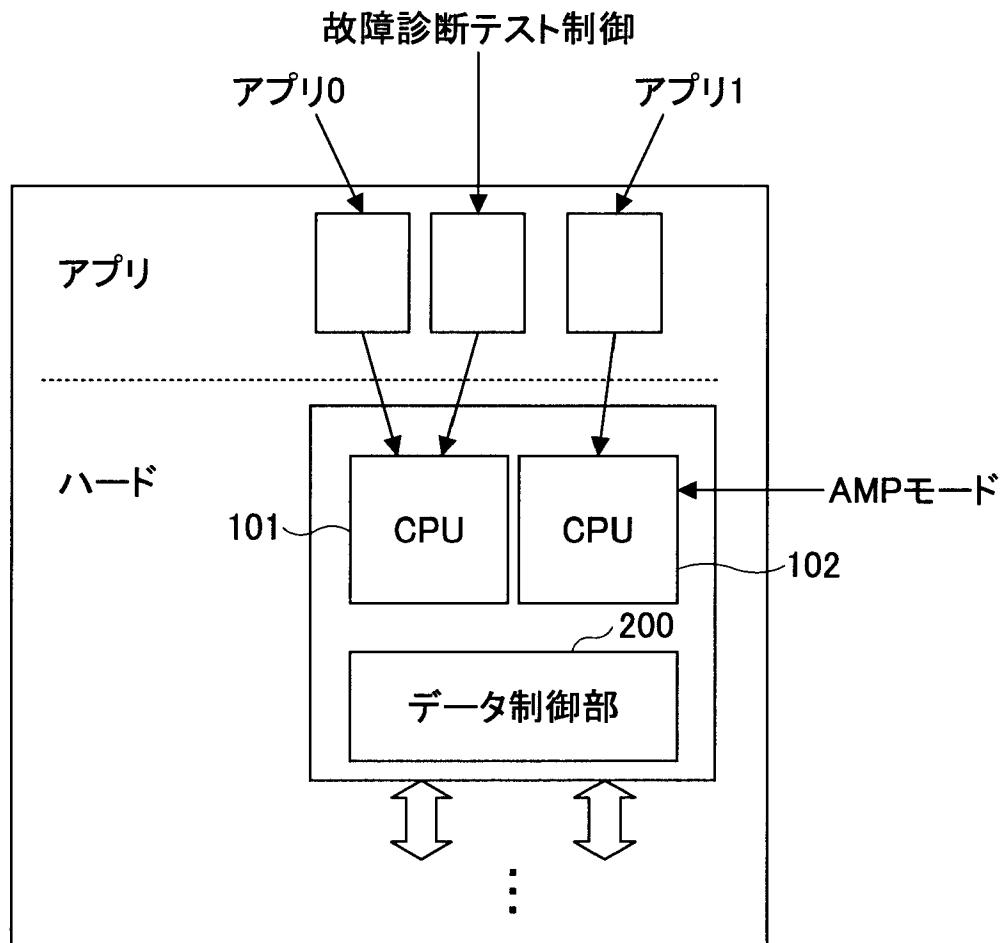
12



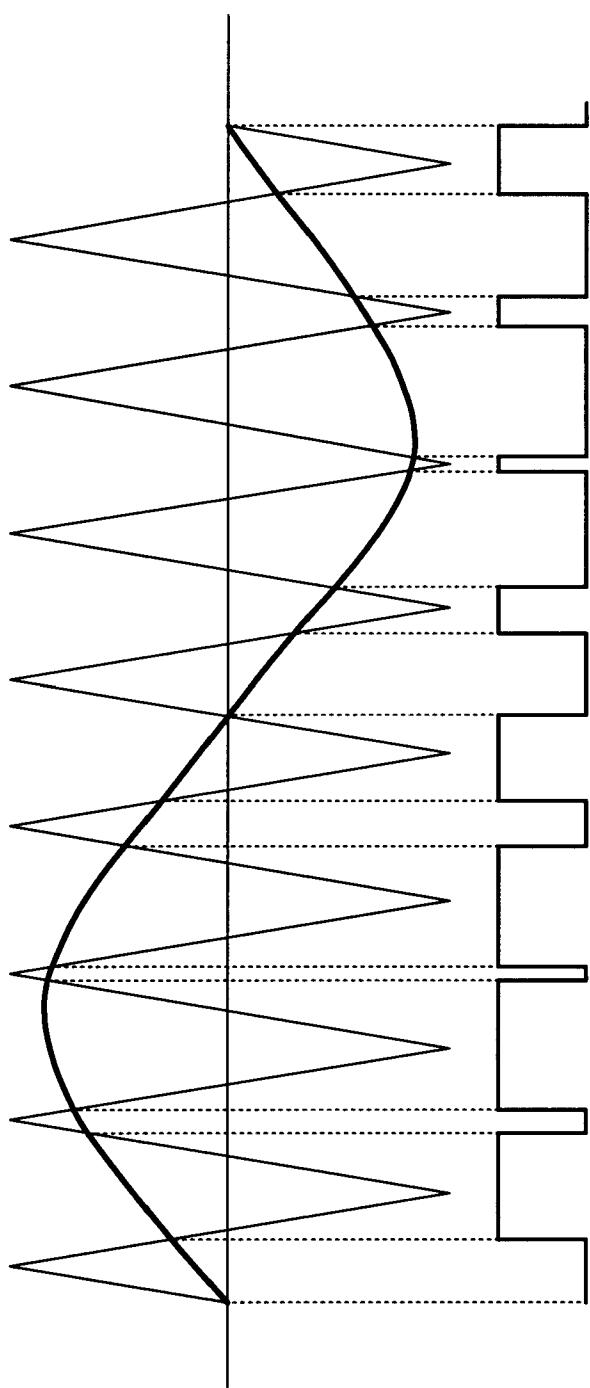
[図9]



[図10]



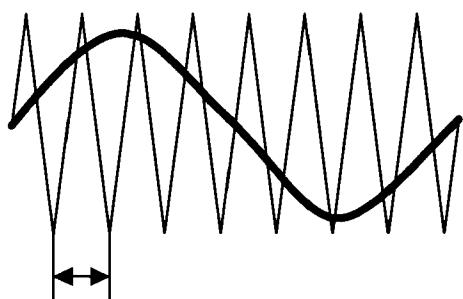
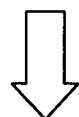
[図11]



[図12]

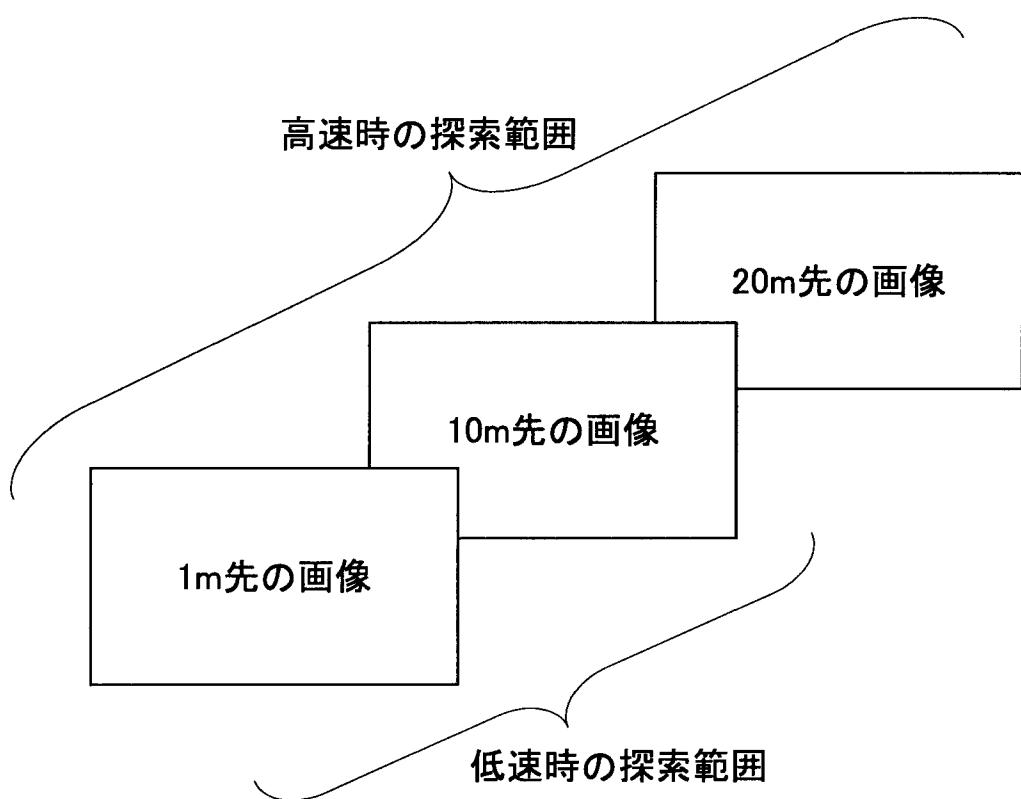


一定処理を行うために
CPUに割り当てられた時間



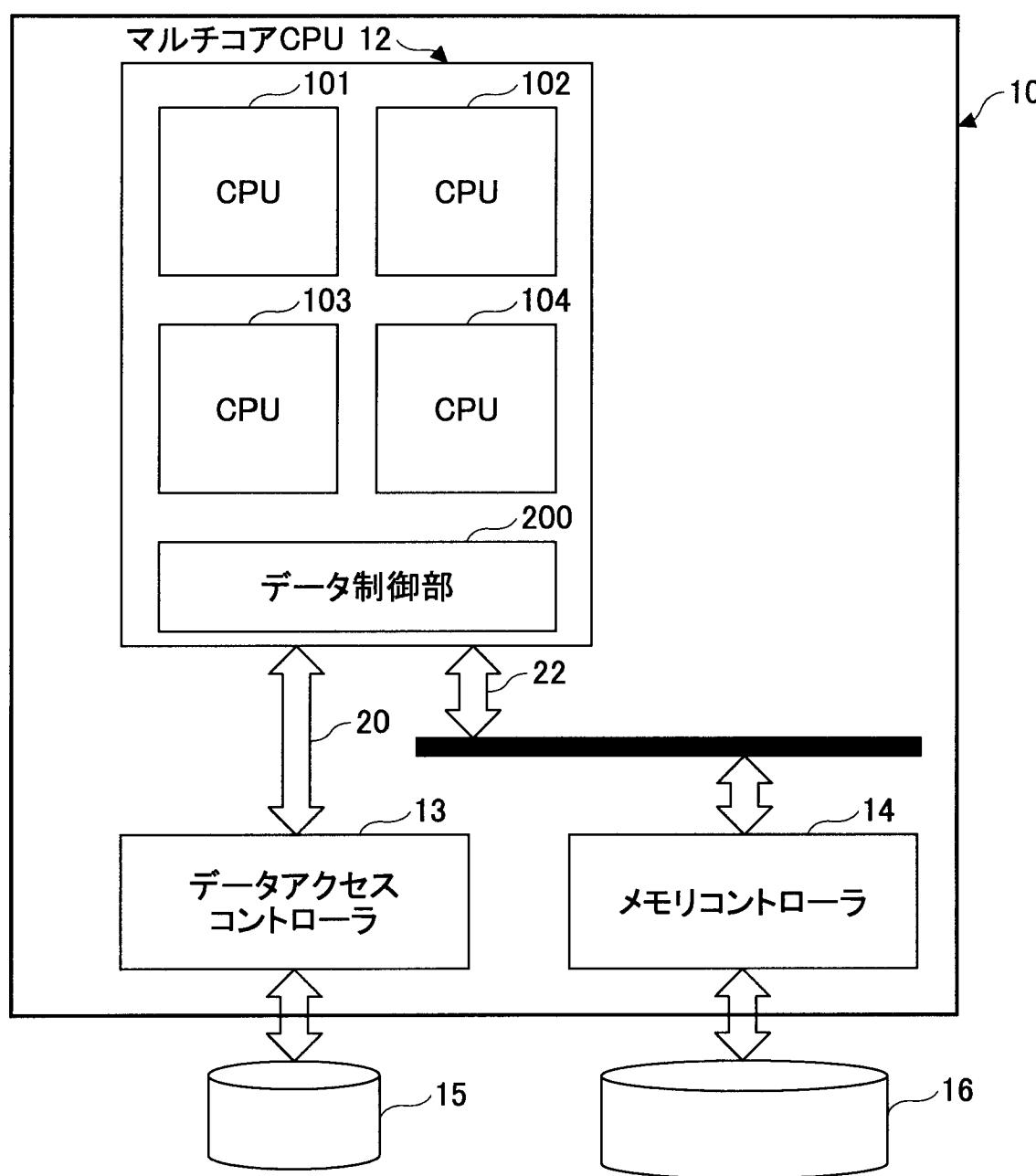
一定処理を行うために
CPUに割り当てられた時間

[図13]

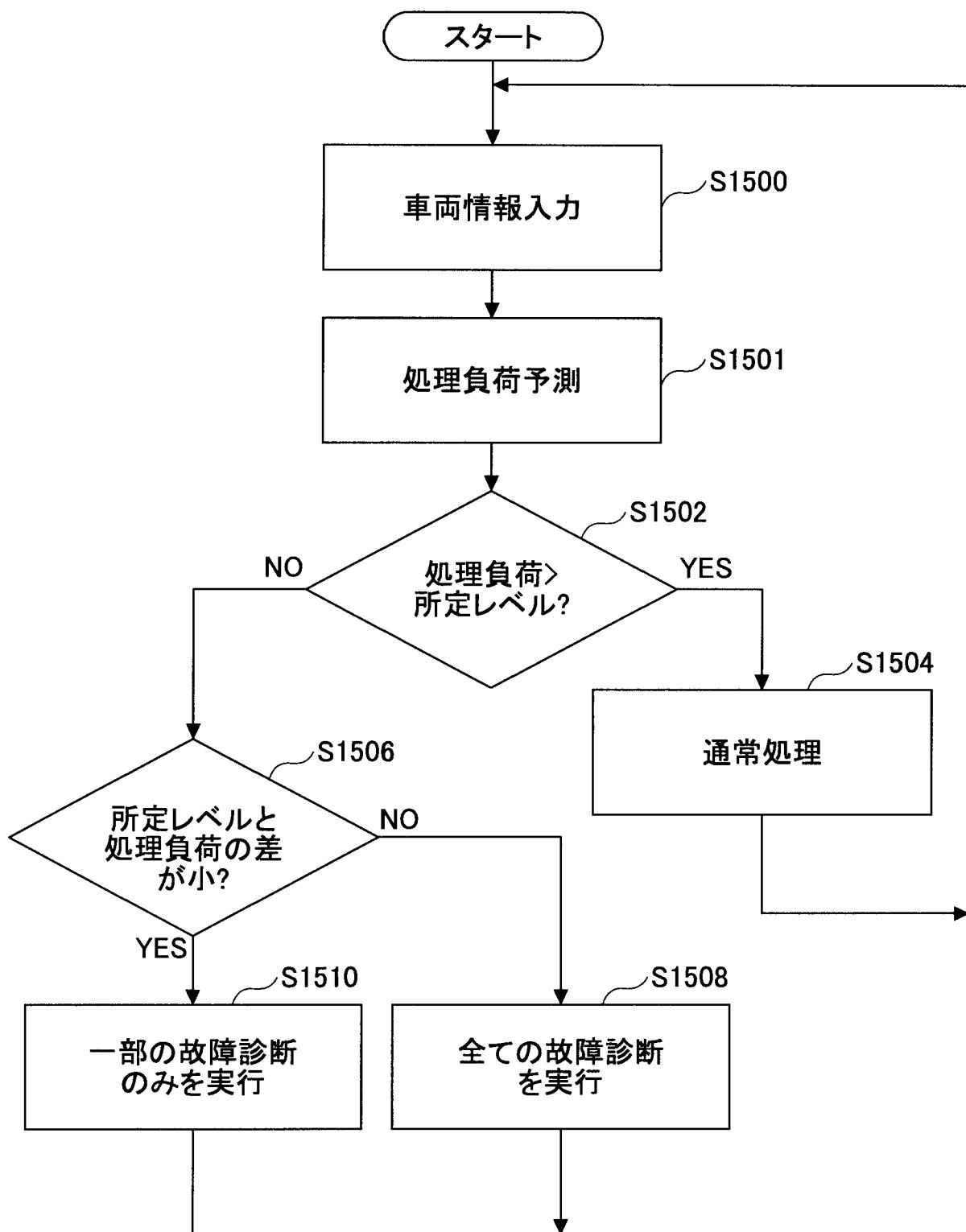


[図14]

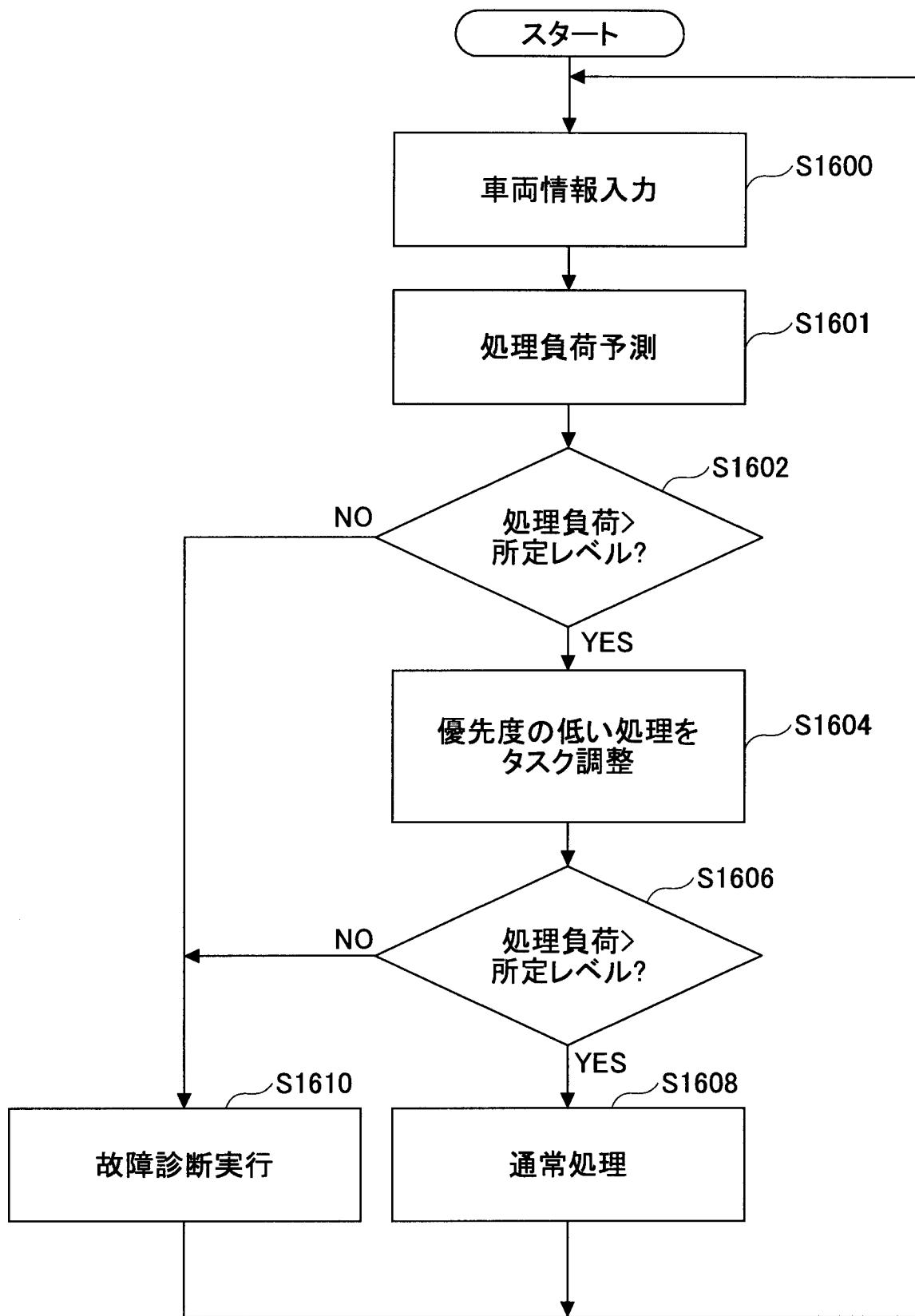
2



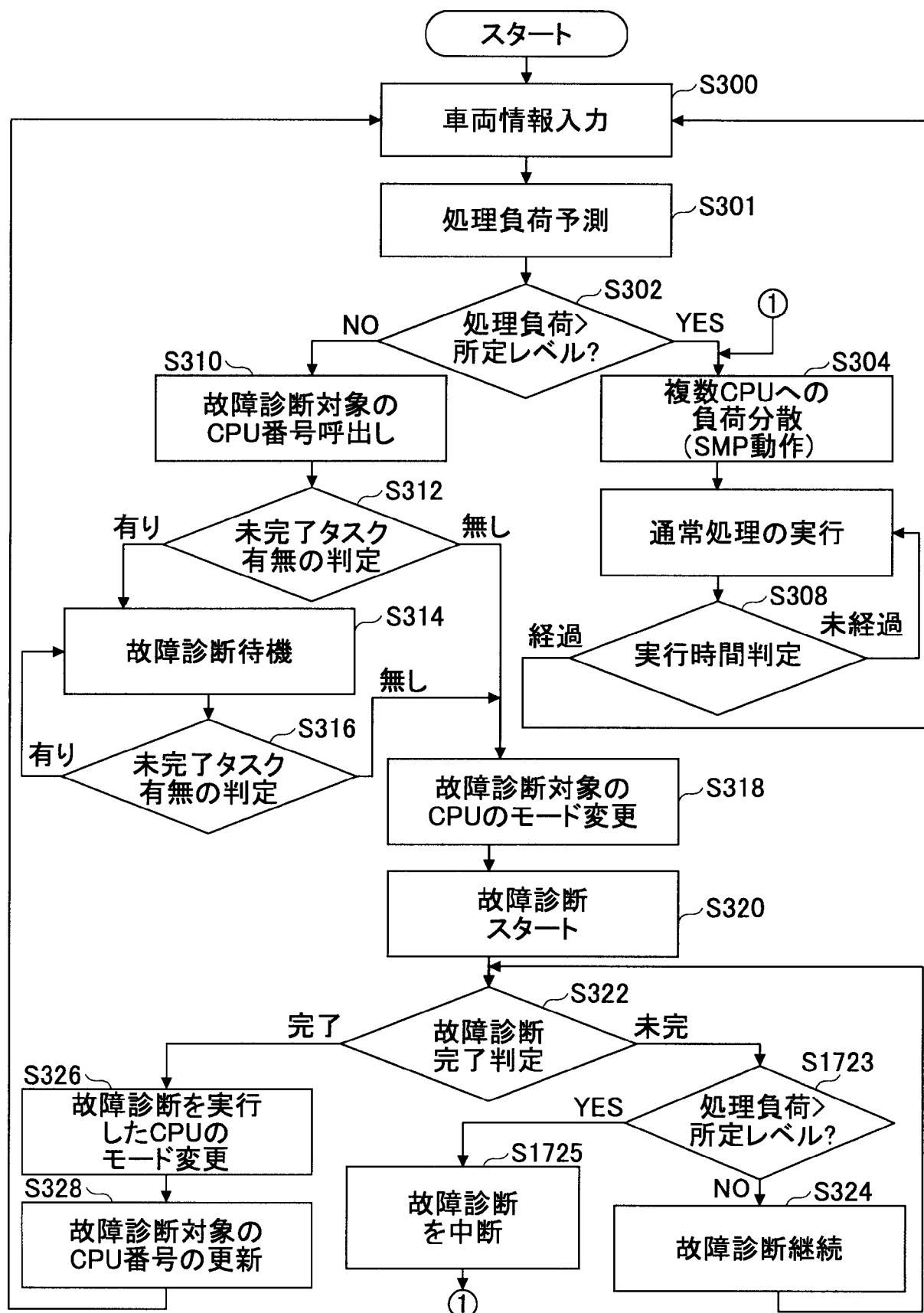
[図15]



[図16]



[図17]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2010/057910

A. CLASSIFICATION OF SUBJECT MATTER

G06F11/22 (2006.01) i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F11/22

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
 Jitsuyo Shinan Koho 1922-1996 Jitsuyo Shinan Toroku Koho 1996-2010
 Kokai Jitsuyo Shinan Koho 1971-2010 Toroku Jitsuyo Shinan Koho 1994-2010

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X Y	JP 11-96044 A (Denso Corp.), 09 April 1999 (09.04.1999), entire text; all drawings & US 6240534 B1	1-6, 9-11 7, 8
Y	WO 2010/010723 A1 (Toyota Motor Corp.), 28 January 2010 (28.01.2010), entire text; all drawings (Family: none)	7, 8
A	JP 2009-251967 A (Toyota Motor Corp.), 29 October 2009 (29.10.2009), entire text; all drawings (Family: none)	1-11

Further documents are listed in the continuation of Box C.

See patent family annex.

* Special categories of cited documents:	
"A"	document defining the general state of the art which is not considered to be of particular relevance
"E"	earlier application or patent but published on or after the international filing date
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O"	document referring to an oral disclosure, use, exhibition or other means
"P"	document published prior to the international filing date but later than the priority date claimed
"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&"	document member of the same patent family

Date of the actual completion of the international search
24 May, 2010 (24.05.10)

Date of mailing of the international search report
01 June, 2010 (01.06.10)

Name and mailing address of the ISA/
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

A. 発明の属する分野の分類（国際特許分類（IPC））

Int.Cl. G06F11/22 (2006.01)i

B. 調査を行った分野

調査を行った最小限資料（国際特許分類（IPC））

Int.Cl. G06F11/22

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	1922-1996年
日本国公開実用新案公報	1971-2010年
日本国実用新案登録公報	1996-2010年
日本国登録実用新案公報	1994-2010年

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求項の番号
X	JP 11-96044 A (株式会社デンソー) 1999.04.09, 全文, 全図	1-6, 9-11
Y	& US 6240534 B1	7, 8
Y	WO 2010/010723 A1 (トヨタ自動車株式会社) 2010.01.28, 全文, 全図 (ファミリーなし)	7, 8
A	JP 2009-251967 A (トヨタ自動車株式会社) 2009.10.29, 全文, 全図 (ファミリーなし)	1-11

□ C欄の続きにも文献が列挙されている。

□ パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー

- 「A」特に関連のある文献ではなく、一般的技術水準を示すもの
 「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの
 「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）
 「O」口頭による開示、使用、展示等に言及する文献
 「P」国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献

- 「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの
 「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの
 「Y」特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの
 「&」同一パテントファミリー文献

国際調査を完了した日 24. 05. 2010	国際調査報告の発送日 01. 06. 2010
国際調査機関の名称及びあて先 日本国特許庁（ISA/JP） 郵便番号100-8915 東京都千代田区霞が関三丁目4番3号	特許庁審査官（権限のある職員） 桑原 雅子 電話番号 03-3581-1101 内線 3545 5B 3981