

(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  
2019年1月17日(17.01.2019)



(10) 国際公開番号

WO 2019/012907 A1

(51) 国際特許分類:  
G06F 11/14 (2006.01) G06F 11/07 (2006.01)

(21) 国際出願番号: PCT/JP2018/022681

(22) 国際出願日: 2018年6月14日(14.06.2018)

(25) 国際出願の言語: 日本語

(26) 国際公開の言語: 日本語

(30) 優先権データ:  
特願 2017-136657 2017年7月12日(12.07.2017) JP

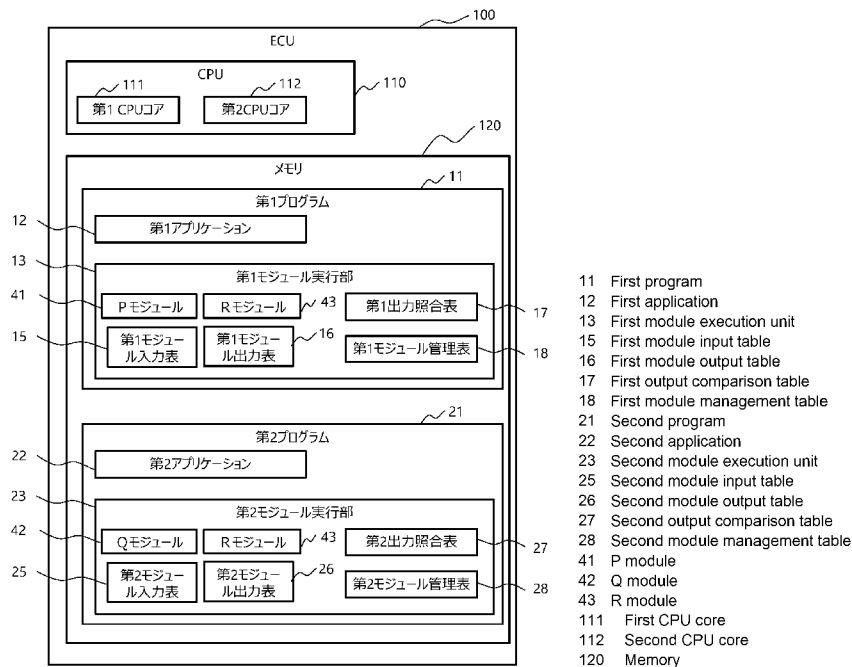
(71) 出願人: 日立オートモティブシステムズ株式会社(HITACHI AUTOMOTIVE SYSTEMS, LTD.) [JP/JP]; 〒3128503 茨城県ひたちなか市高場 2 5 2 0 番地 Ibaraki (JP).

(72) 発明者: 下 沢 拓 (SHIMOSAWA, Taku); 〒1008280 東京都千代田区丸の内一丁目 6 番 6 号 株式会社日立製作所内 Tokyo (JP). 飯室 聡(IIMURO, Satoshi); 〒1008280 東京都千代田区丸の内一丁目 6 番 6 号 株式会社日立製作所内 Tokyo (JP). 小田 裕弘(ODA, Yasuhiro); 〒3128503 茨城県ひたちなか市高場 2 5 2 0 番地 日立オートモティブシステムズ株式会社内 Ibaraki (JP). 成沢 文雄(NARISAWA, Fumio); 〒3128503 茨城県ひたちなか市高場 2 5 2 0 番地 日立オートモティブシステムズ株式会社内 Ibaraki (JP). 林 正人(HAYASHI, Masato); 〒3128503 茨城県ひたちなか市高場 2 5 2 0 番地 日立オートモティブシステムズ株式会社内 Ibaraki (JP).

(54) Title: COMPUTATION DEVICE

(54) 発明の名称: 演算装置

【図1】



(57) Abstract: A computation device comprising: a storage unit storing a first software module, a second software module, and a third software module each having the same function but implemented separately; a computation unit for executing the first software module, the second software module, and the third software module; a detection unit for detecting a failure on the basis of an output from the first software module and an output from the second software module implemented by the computation unit; and a failure identification unit for starting the use of the third software module when a failure is



(74) 代理人: 特許業務法人サンネクスト国際特許事務所(SUNNEXT INTERNATIONAL PATENT OFFICE); 〒1400002 東京都品川区東品川二丁目 3 番 1 2 号 シーフォートスクエア センタービルディング 1 6 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, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, 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, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, 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, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

添付公開書類:

一 国際調査報告(条約第21条(3))

detected by the detection unit, and identifying the software module in which the failure occurred on the basis of an output from the first software module, an output from the second software module, and an output from the third software module.

(57) 要約: 演算装置は、同一の機能を有し異なる実装がなされた第1のソフトウェアモジュール、第2のソフトウェアモジュール、および第3のソフトウェアモジュールが格納される記憶部と、第1のソフトウェアモジュール、第2のソフトウェアモジュール、第3のソフトウェアモジュールを実行する演算部と、演算部による第1のソフトウェアモジュールの出力、および第2のソフトウェアモジュールの出力に基づき障害を検出する検出部と、検出部が障害を検出すると、第3のソフトウェアモジュールの使用を開始し、第1のソフトウェアモジュールの出力と、第2のソフトウェアモジュールの出力と、第3のソフトウェアモジュールの出力と、に基づき障害のあるソフトウェアモジュールを特定する障害特定部と、を備える。

## 明 細 書

**発明の名称 : 演算装置**

**技術分野**

[0001] 本発明は、演算装置に関する。

**背景技術**

[0002] 車載制御装置では、要求される処理内容の高度化が進んでおり、一般のサーバーやワークステーション等で行われている技術やソフトウェアを利用することも増えつつある。このようなソフトウェアは従来の車載制御装置において行われていた開発手法によって作成されておらず、その品質は必ずしも高くない。しかも車載制御装置では、ソフトウェアのバグといった障害が発生したとしても、動作を継続させなければならず、障害の検出のみならず、障害箇所を特定し切り離すことが必要になる。この目的のため、同じ機能を持つ複数の異なる実装を照らし合わせてチェックすることで、車載制御装置全体の最終的な品質を高めることが行われており、多数決を行うことで一つの実装で障害が発生しても動作を継続させることができる。

特許文献1には、予め設定されたアプリケーションによる処理を実行し、実行された処理結果を比較してデータの診断及び修復を行う診断修復装置において、所定のデータを3以上の複数のデータ領域に記憶する記憶手段と、前記所定のデータを前記3以上の複数のデータ領域に異なる形式で書き込み、前記アプリケーションの実行時に前記3以上の複数のデータ領域にそれぞれ書き込んだデータを読み込んで照合し、照合の結果が不一致であった場合に所定の修復を行い、修復後のデータを用いて前記アプリケーションを実行する処理プログラム実行手段と、前記アプリケーションを実行した結果と、少なくとも1つの他の診断修復装置で前記アプリケーションを実行した結果とを照合する照合処理手段が開示されている。

**先行技術文献**

**特許文献**

[0003] 特許文献1：日本国特開2013-109532号公報

## 発明の概要

### 発明が解決しようとする課題

[0004] 特許文献1に記載されている発明では、限られた演算資源を用いて障害の検出および障害の特定を実現することができない。

### 課題を解決するための手段

[0005] 本発明の第1の態様による演算装置は、同一の機能を有し異なる実装がなされた第1のソフトウェアモジュール、第2のソフトウェアモジュール、および第3のソフトウェアモジュールが格納される記憶部と、前記第1のソフトウェアモジュール、前記第2のソフトウェアモジュール、前記第3のソフトウェアモジュールを実行する演算部と、前記演算部による前記第1のソフトウェアモジュールの出力、および前記第2のソフトウェアモジュールの出力に基づき障害を検出する検出部と、前記検出部が障害を検出すると、前記第3のソフトウェアモジュールの使用を開始し、前記第1のソフトウェアモジュールの出力と、前記第2のソフトウェアモジュールの出力と、前記第3のソフトウェアモジュールの出力と、に基づき障害のあるソフトウェアモジュールを特定する障害特定部と、を備える。

### 発明の効果

[0006] 本発明によれば、限られた演算資源を用いて障害の検出および障害の特定を実現することができる。

### 図面の簡単な説明

[0007] [図1]第1の実施の形態におけるECU100の構成を表すブロック図

[図2]第1モジュール入力表15の一例を示す図

[図3]第1モジュール出力表16の一例を示す図

[図4]第1出力照合表17の一例を示す図

[図5]第1モジュール管理表18の一例を示す図

[図6]ECU100の動作例を示す図

[図7]障害検出処理を示すフローチャート

[図8]障害特定処理を示すフローチャート

[図9]第2の実施の形態におけるECU100Aの構成を表すブロック図

[図10]第2の実施の形態における第1モジュール実行部13Bの動作を表すフローチャート

[図11]第3の実施の形態における制御システムSの構成を表すブロック図

[図12]第3の実施の形態における障害特定処理を示すフローチャート

### 発明を実施するための形態

[0008] 一第1の実施の形態—

以下、図1～図8を参照して、制御装置であるECUの第1の実施の形態を説明する。

[0009] (構成)

図1は、ECU100すなわち電子制御装置(Electronic Control Unit)の構成を表すブロック図である。ECU100は、演算装置であるCPU110および、主記憶領域であるメモリ120を有する。メモリ120には後述するように複数のモジュールが格納される。

[0010] モジュールとは、複数のアプリケーションから共通に利用可能なソフトウェアである。モジュールとはたとえば、ライブラリ、関数、関数の一部、または共通に利用される機能に特化したアプリケーションなどである。本実施の形態では、モジュールはアプリケーションから呼びされる複数の関数からなるソフトウェア、たとえば動的にリンクされるライブラリ(Dynamic Link Library)であり、アプリケーションとは独立している。モジュールへの入力、これらの関数に対する引数である。モジュールの出力は、これらの関数の返り値である。また本実施の形態では符号41で表すPモジュール、符号42で表すQモジュール、および符号43で表すRモジュールの3つのモジュールを説明するが、これらをまとめて単に「モジュール」とも呼ぶ。

[0011] Pモジュール41、Qモジュール42、およびRモジュール43はいずれ

も同じ機能を有し、いずれのモジュールを用いる場合であっても、同一の入力に対しては同一の出力が得られることが期待される。ただしこれらのモジュールは実装が異なっており、バイナリー一致ではない。Pモジュール41、Qモジュール42、およびRモジュール43はたとえば、ある規格化されているライブラリの第1の実装、第2の実装、および第3の実装である。Rモジュール43よりもPモジュール41およびQモジュール42の信頼性が高く、換言すると完成度が高い。またたとえばCPU110が32ビットCPU用に作成されたアプリケーションと64ビットCPU用に作成されたアプリケーションの両方が実行可能な場合は、以下の構成でもよい。すなわちPモジュール41が32ビットCPU用の第1の実装、Qモジュール42が64ビットCPU用の第1の実装、Rモジュール43が32ビットCPU用の第2の実装であってもよい。

[0012] CPU110は、汎用的な計算が可能な演算装置であり、並列に動作可能な2つのCPUコア、すなわち第1CPUコア111と第2CPUコア112とを有する。ただしCPU110は、3つ以上のCPUコアを有していてもよいし、1つのCPUが論理的に複数のコアを有するようにふるまう機能、いわゆるハードウェアハイパースレディング機能を有してもよい。さらにCPU110は、半導体チップとして単体のものであってもよいし、以下で述べるメモリ120を一つの半導体チップ上に搭載した、いわゆるSoC (System-on-Chip) であってもよい。

[0013] メモリ120は、揮発性である、たとえばDRAM (Dynamic Random Access Memory) などから構成される主記憶装置である。メモリ120には、CPU110が実行するコードおよび読み書きするデータを含むプログラムが配置される。本実施の形態では、第1CPUコア111の使用するプログラムと第2CPUコア112が実行するプログラムを、それぞれ第1プログラム11と第2プログラム21とする。第1プログラム11および第2プログラム21は同様の構成を有し同様の動作を行う。第1CPUコア111と第2CPUコア112は通信が可能であり、こ

れにより第1プログラム11と第2プログラム21とが情報の授受が可能である。

[0014] 第1プログラム11は、第1アプリケーション12と、第1モジュール実行部13とを有する。第1モジュール実行部13は、Pモジュール41と、Rモジュール43と、第1モジュール入力表15と、第1モジュール出力表16と、第1出力照合表17と、第1モジュール管理表18とを有する。第2プログラム21は、第2アプリケーション22と、第2モジュール実行部23とを有する。第2モジュール実行部23は、Qモジュール42と、Rモジュール43と、第2モジュール入力表25と、第2モジュール出力表26と、第2出力照合表27と、第2モジュール管理表28とを有する。

[0015] 第1アプリケーション12と第2アプリケーション22は同一、いわゆるバイナリー一致の関係にある。第1プログラム11および第2プログラム21は同様の構成を有し同様の動作を行う。仮に第1プログラム11および第2プログラム21に備えられるいずれのモジュールにもバグ等の障害がなく動作する場合には、いずれのアプリケーションの実行結果も同一のものとなる。図示していない照合装置などを用いて、第1プログラム11と第2プログラム21の実行結果を照合して、最終的なECU100の出力とすることによって、安全性と信頼性を高めて制御ソフトウェアを実行することができる。

[0016] 第2プログラム21に含まれるRモジュール43は、第1プログラム11に含まれるRモジュール43と同一、すなわちバイナリー一致の関係にある。第2モジュール入力表25と、第2モジュール出力表26、第2出力照合表27、および第2モジュール管理表28のそれぞれは、第1モジュール入力表15、第1モジュール出力表16、第1出力照合表17、および第1モジュール管理表18のそれぞれと構成が同一である。以下では第1プログラム11の構成および動作を主に説明する。

[0017] 第1プログラム11の第1アプリケーション12は、ECU100における主要な機能を実現するソフトウェアである。なお本実施の形態では、簡単

のために第1アプリケーション12は一つとするが、複数のアプリケーションを備えてもよい。第1アプリケーション12はモジュールの機能を必要とする時に、第1モジュール実行部13を呼び出し、モジュールを実行してその出力を得る。以下では簡単のためにこの処理を「第1アプリケーション12からモジュールを呼び出す」と表現するが、実際には後述するようにモジュールの実行には第1モジュール実行部13が介在する。

[0018] 第1モジュール実行部13は、第1アプリケーション12から呼び出されてモジュールを実行する。第1モジュール実行部13は、第1アプリケーション12から呼び出されると1つのモジュールを選択して実行し、後述する所定の条件を満たす場合に第2モジュール実行部23におけるモジュールの実行結果と比較する。なお後述するように、第2モジュール実行部23は第1モジュール実行部13とは異なるモジュールを使用している。この比較により使用したいずれのモジュールにも障害がないと判断する場合は算出結果を第1アプリケーション12に出力する。しかし、両者が不一致などの理由でいずれかのモジュールに障害が発生していると判断する場合は、第1モジュール実行部13はもう一つのモジュールを使用して障害のあるモジュールを特定する。すなわちECU100は通常は、換言すると障害があると判断するまでは、第1モジュール実行部13と第2モジュール実行部23が異なるモジュールを1つずつ、合計2つのモジュールのみを使用する。そして障害があると判断すると、3つ目のモジュールを使用して障害のあるモジュールを特定する。

[0019] 第1モジュール入力表15には、第1アプリケーション12がモジュールを呼び出して何らかの入力が行われるたびにモジュールへの入力追記される。第1モジュール出力表16には、モジュールが出力を行うたびにその出力が記録される。第1出力照合表17には、モジュールが備えるそれぞれの関数について、その関数の返り値が比較可能なものであるかを示す情報が予め記録される。第1出力照合表17は、モジュールの出力を比較するか否かを判断する際に参照される。第1モジュール管理表18には、モジュールの

障害の有無が記載される。第1モジュール管理表18の記載に基づき、実行するモジュールが選択される。以下、各構成を詳述する。

[0020] (第1モジュール入力表15)

図2は、第1モジュール入力表15の一例を示す図である。ただし図2では、一度に複数の関数が呼び出された場合の例を示しており、1つの関数の1回呼び出しが第1モジュール入力表15の1行に相当する。また第1モジュール入力表15の各行は、呼び出された順に並んでいる。第1モジュール入力表15の最初の列である「No.」には、各行の番号が格納される。なお各行の順番と識別が可能であれば、数字以外の識別子を用いてもよい。第1モジュール入力表15の2番目の列である「呼び出し関数」には、各呼び出しにおいて呼び出された関数を識別する情報、たとえば関数名や関数のアドレス番地が格納される。第1モジュール入力表15の3番目の列である「入力」には、各呼び出しにおけるモジュールへの入力が格納される。なお図2に示す例では全ての呼び出しにおいて何らかの入力がされているが、入力の値が存在せず引数を持たない関数の場合には引数がないことを示す情報、たとえば「なし」や「-」が記録される。第1モジュール実行部13は、第1アプリケーション12がモジュールを呼び出す際に、第1モジュール入力表15に新たな行を追加し、関数の識別子とモジュールへの入力を記録する。

[0021] (第1モジュール出力表16)

図3は、第1モジュール出力表16の一例を示す図である。第1モジュール出力表16の各行は、いずれかのモジュールからの1回の出力に対応する。モジュールの出力は入力に応じて発生するので、第1モジュール出力表16の各行はモジュールへの1回ずつの呼び出し、すなわち第1モジュール入力表15の各行に対応する。第1モジュール出力表16の最初の列である「No.」には、各行の番号が格納される。なお各行の順番と識別が可能であれば、数字以外の識別子を用いてもよい。第1モジュール出力表16の2番目の列である「出力」には、それぞれの呼び出しにおけるモジュールの出力が格納される。なお出力値がない場合は出力値がないことを示す情報、たと

例えば「なし」や「ー」が記録される。第1モジュール実行部13は、モジュールを実行してその出力が得られると、第1モジュール出力表16に新たな行を追加してその出力を追記する。

[0022] ただし第1モジュール出力表16には実行した全てのモジュールの出力が格納される。1つのテーブルに複数のモジュールの出力が格納されてもよいし、モジュールごとに異なるテーブルに出力が格納されてもよい。

[0023] (第1出力照合表17)

図4は、第1出力照合表17の一例を示す図である。第1出力照合表17の各行は、モジュールに備えられる各関数の出力が比較が可能であるか否かを示す情報が格納される。第1出力照合表17の最初の列である「関数」には、関数の識別子が格納される。第1出力照合表17の第二の列である「出力比較区分」には、その関数の出力の比較を行うか否かを示す情報、たとえば、`Compare`、`Ignore`、および`Compare if error`のいずれかである。`Compare`は比較の対象であることを表し、`Ignore`は比較の対象ではないことを表す。また`Compare if error`は、出力がエラーを示す値であった場合にはその内容を比較し、成功した場合には比較しないことを表す。

[0024] `Compare if error`が適用される関数は、たとえば次のような関数である。すなわちその関数は、動的にメモリ領域を確保して成功時には正の整数で確保した領域の先頭アドレスを返し、失敗時には負の整数でエラーの種類を返す。この関数を複数のモジュールでそれぞれ実行した場合に、正常に領域が確保できれば両者は異なる値を出力するので比較する意味がない。しかし失敗した場合にはエラーの種類が異なればモジュールに何らかの問題があることがわかる。このように、特定条件下でのみ出力を比較する関数を規定することにより、比較をしない関数を減らすことができ、単純に`Compare`と`Ignore`の二種類とするよりも、障害の検出できる頻度を増やすことができる。

[0025] (第1モジュール管理表18)

図5は、第1モジュール管理表18の一例を示す図である。第1モジュール管理表18の各行には、各モジュールの動作状況が格納される。第1モジュール管理表18の最初の列である「モジュール」にはモジュールを識別する識別子が格納される。次の列である「動作状況」には、そのモジュールの動作状況が格納される。動作状況は、モジュールが正常であるか何らかの問題があるかを、それぞれ「OK」と「NG」で表している。初期状態では、全てのモジュールは「OK」であり、後で述べるように出力が不一致となり障害があるモジュールが特定され、障害があることが特定されたモジュールは「NG」となる。なお第1モジュール実行部13は、そのモジュールのうち「動作状況」が「OK」のモジュールを使用して出力を得ることで、障害のないモジュールを使用し続けることができる。

[0026] (動作例)

図6は、ECU100の動作例を示す図である。図6(a)～図6(d)はすべて同一の時刻における情報を示しており、図6(a)は第1モジュール入力表15を示す図、図6(b)はPモジュール41の出力である第1モジュール出力表16を示す図、図6(c)はQモジュール42の出力である第2モジュール出力表26を示す図、図6(d)はRモジュール43の出力である第1モジュール出力表16Aの例を示す図である。なお第1出力照合表17は、図4に例示したものを使用して説明する。なおRモジュール43の出力は第1モジュール出力表16に含まれるが、ここでは説明のために第1モジュール出力表16とは独立した第1モジュール出力表16Aに格納されるとして説明する。

[0027] まず、Pモジュール41およびQモジュール42がそれぞれ第1プログラム11と第2プログラム21で使用されて動作した。その結果、図6(a)の第1モジュール入力表15に記載されているように、3回のモジュール呼び出しがあり、Function A、Function B、Function Cがこの順で呼ばれた。この際のPモジュール41の出力は、図6(b)の第1モジュール出力表16に示すように、順に、0、100、-1

であった。その一方でQモジュール42の出力は、図6(c)の第2モジュール出力表26に示すように、順に、0、200、-2であった。

[0028] 図4に示す第1出力照合表17によれば、Function Aは`Compare`なので照合を行うが、Pモジュール41とQモジュール42の出力は一致しており問題はない。次のFunction Bは`Ignore`なので照合は行わず、Pモジュール41とQモジュール42の出力は一致していないが問題はない。最後のFunction Cは`Compare if error`であり、Pモジュール41とQモジュール42の出力はいずれも負数であるのでエラーである。

[0029] したがってPモジュール41とQモジュール42の出力を比較する。両者の出力を比較すると、`-1`と`-2`なので両者は不一致であり、Pモジュール41とQモジュール42のいずれかに障害があると判断される。ここでいずれに障害があるかを判断するために、Rモジュール43に対して同様の入力が行われ、図6(d)の第1モジュール出力表16Aに示す結果となった。Pモジュール41とQモジュール42とでは出力が不一致となったFunction Cの出力は、図6(d)ではその結果は`-2`である。すなわち、Rモジュール43の出力はQモジュール42と一致し、Pモジュール41とは一致しない。多数決により、Qモジュール42およびRモジュール43が正常であり、Pモジュール41に障害があることが特定できる。

[0030] (フローチャート)

図7および図8は、図6を参照して説明した動作例を実現する第1モジュール実行部13の動作を表すフローチャートである。図7は障害検出処理、図8は障害特定処理の詳細を示す。なお第2プログラム21に備えられる第2モジュール実行部23の動作は、第1プログラム11の第1モジュール実行部13の動作とほぼ同様である。以下では両者に共通する点は第1プログラム11の第1モジュール実行部13のみを説明する。

[0031] 第1モジュール実行部13は、第1アプリケーション12がモジュールを

呼び出すと図 7 のフローチャートにより動作が表されるプログラムを実行する。第 1 アプリケーション 1 2 がモジュールを呼び出すと、第 1 モジュール実行部 1 3 にはモジュールの関数の識別子と入力値とが与えられる。第 1 モジュール実行部 1 3 は、この識別子および入力値を第 1 モジュール入力表 1 5 の新たな行に追記する (S 7 0 1)。次に第 1 モジュール実行部 1 3 は、第 1 アプリケーション 1 2 の呼び出しを実行するためのモジュールを選択する (S 7 0 2)。

[0032] モジュールの選択は、第 1 モジュール実行部 1 3 が備えるモジュールのうち、第 1 モジュール管理表 1 8 で `OK` であるモジュールから選択する。ただし第 1 モジュール実行部 1 3 は、第 1 プログラム 1 1 と第 2 プログラム 2 1 で動作するモジュールが異なるようにモジュールを選択する。そのためにたとえば、第 1 モジュール実行部 1 3 はモジュールを昇順に選択し、第 2 モジュール実行部 2 3 はモジュールを降順に選択する。昇順に選択とは、P モジュール 4 1 を最優先で選択し、P モジュール 4 1 が NG の場合は Q モジュール 4 2 を選択し、Q モジュール 4 2 も NG の場合は R モジュール 4 3 を選択する選択方法である。降順に選択とは、昇順とは逆の順番に選択する選択方法である。ただしそれぞれのプログラムに備えられていないモジュールは選択できないので、昇順および降順の場合は上述した順番であってそのプログラムに備えられるモジュールを選択する。

[0033] 第 1 モジュール実行部 1 3 は、選択したモジュールの関数を実行して出力値を得る (S 7 0 3)。次に照合を行うために、第 2 プログラム 2 1 の第 2 モジュール実行部 2 3 における第 2 モジュール出力表 2 6 の内容を取得する (S 7 0 5)。第 1 モジュール実行部 1 3 は、以下に説明する S 7 0 6 ~ S 7 1 0 の処理を第 1 モジュール出力表 1 6 に記載された各行について、処理対象とする行 (以下、「処理対象行」) を 1 行ずつ変化させて繰り返し実行する (S 7 0 5 A)。ただし第 1 アプリケーション 1 2 がモジュールを呼び出すたびに第 1 モジュール実行部 1 3 が図 7 に示す全ての動作を行う場合は、出力表には 1 行しか記載されないので S 7 0 5 A および後述する S 7 1 0

Aは存在しないものとして扱ってよい。

[0034] 第1モジュール実行部13は、処理対象行における呼び出し関数を特定し、第1出力照合表17を参照してその関数の出力比較区分を取得し、その出力比較区分が`ignore`であるか否かを判断する(S706)。第1モジュール実行部13は、その出力比較区分が`ignore`であると判断する場合(S706:Yes)はS710Aに進む。第1モジュール実行部13は、その出力比較区分が`ignore`ではないと判断する場合は(S706:No)、出力比較区分が`Compare if error`であるか否かを判断する(S707)。第1モジュール実行部13は、その出力比較区分が`Compare if error`であると判断する場合(S707:Yes)はS708に進む。第1モジュール実行部13は、その出力比較区分が`Compare if error`ではないと判断する場合(S707:No)はS710に進む。

[0035] 第1モジュール実行部13はS708では、第1プログラム11における処理対象行の出力値、および第2プログラム21における処理対象行の出力値が、両方とも成功を示す値であるか、それとも少なくとも一方は失敗を示す値であるかを判断する。第1モジュール実行部13は、両方とも成功を示す値であると判断する場合は評価が不要なのでS710Aに進み、少なくとも一方は失敗を示す値であると判断する場合はステップS709に進む。第1モジュール実行部13はステップS709では、第1プログラム11における処理対象行の出力値、および第2プログラム21における処理対象行の出力値が、両方ともエラーを示す値であるか、片方のみがエラーを示す値であるかを判断する。

[0036] 第1モジュール実行部13は両方ともエラーを示す値であると判断する場合は比較のためにステップS710に進み、片方のみがエラーを示す値であると判断する場合はステップS720の障害特定処理に進む。ただし障害特定処理の詳細は次の図8を参照して後に説明する。第1モジュール実行部13はステップS710では、第1プログラム11における処理対象行の出力

値、および第2プログラム21における処理対象行の出力値が一致するか否かを判断する。第1モジュール実行部13は、両者が一致すると判断する場合はステップS710Aに進み、両者が一致しないと判断する場合はステップS720の障害特定処理に進む。

[0037] ステップS710Aでは、第1モジュール実行部13は出力表の全ての行を処理対象としたか否かを判断し、処理対象としていない行があると判断する場合はその行を処理対象としてステップS706に戻る。第1モジュール実行部13は出力表の全ての行を処理対象としたと判断する場合はステップS711に進む。第1モジュール実行部13はステップS711では、第1モジュール入力表15および第1モジュール出力表16に格納されている情報を消去する。これにより、次に図7に示す処理が実行された際に同じ照合が実行されることを避けることができ、次の照合時間を短縮できる。次に第1モジュール実行部13は、S703で得た出力値を第1アプリケーション12に渡し図7に示す処理を終了する(S712)。

[0038] 図8は、障害特定処理の詳細を示すフローチャートである。図8のフローチャートで示す処理は、図7のステップS709またはステップS710において否定判定されると実行される。図8ではまず、第1モジュール実行部13は、第1モジュール管理表18から`OK`となっているモジュールであって、自身が使用したモジュールおよびS705によって得た出力を行ったモジュール以外のモジュールを選択する(S801)。S705によって得た出力を行ったモジュールの特定は、第2プログラム21から通信で取得してもよいし、予め知得した第2モジュール実行部23におけるモジュールの選択方法により特定してもよい。S801において選択したモジュールを以下では代替モジュールと呼ぶ。第1モジュール実行部13は、図7に示す処理において出力が不一致となった関数について、第1モジュール入力表15に記載されている入力値を代替モジュールに入力し、その出力値を得る(S802)。

[0039] そして第1モジュール実行部13は、その出力値をこれまでの2つと比較

する（S803）。第1モジュール実行部13は、代替モジュールの出力が図7のS710において比較した2つのいずれかの出力値と一致するか否かを判断する（S804）。第1モジュール実行部13は、いずれかと一致すると判断する場合（S804：Yes）は、一致しない方の出力を行ったモジュールに障害があると特定し、そのモジュールを第1モジュール管理表18で`NG`とする（S805）。そして第1モジュール実行部13は、S711と同様に第1モジュール入力表15と第1モジュール出力表16の内容をクリアする（S806）。さらに第1モジュール実行部13は、多数派の出力値、すなわち、代替モジュールの出力値を第1アプリケーション12に出力して（S807）図8に示す処理を終了する。

[0040] 第1モジュール実行部13は、S804においてモジュール3つの出力が全て一致しないと判断する場合（S804：No）は、全てのモジュールに障害がある可能性を考慮して障害時の処理へと移行する（S811）。障害時の処理とはたとえば、縮退運転など機能を制限して安全を確保する処理である。ただしS811に到達するケースは2つ以上のモジュールが同じ問題、たとえばバグを有しているなどのケースであり、発生確率は極めて低い。

[0041] （動作の説明）

第1プログラム11および第2プログラム21は、同じ動作を行うアプリケーションがそれぞれ動作している。このため、第1プログラム11および第2プログラム21の動作は、モジュールにバグなどの障害がなければ、同じ動作を行う。また図1に示すように、第1プログラム11および第2プログラム21に備えられるモジュールは、それぞれPモジュール41とRモジュール43、Qモジュール42とRモジュール43である。第1プログラム11はPモジュール41を優先し、第2プログラム21はQモジュール42を優先して選択するので、第1プログラム11と第2プログラム21はそれぞれ、同じ入力を異なるモジュールに入力する。したがってこれらの出力は同じ値になることが期待され、これを第1モジュール実行部13が照合する。両者が不一致である場合には、いずれかのモジュールに障害があることが

わかるため、障害を有するモジュールの出力を第1アプリケーション12がそのまま使用することを防ぐことができる。

[0042] さらに両者が不一致の場合には、第1モジュール実行部13が代替モジュールとして前述の例でいえばRモジュール43を選択し、第1モジュール実行部13はRモジュール43の実行によって第3の出力値を得る。第1モジュール実行部13は、Pモジュール41、Qモジュール42、およびRモジュール43の3つの値を比較し多数決により正しい出力を決定する。そして第1モジュール実行部13は、多数決における少数派に属する出力値を出力したモジュールが障害を有するモジュールと判断し、第1モジュール管理表18のそのモジュールの動作状況に“NG”を記録する。

[0043] 第1アプリケーション12は前述の多数決による多数派の値を用いて動作を継続する。そして第1モジュール実行部13は、たとえばPモジュール41に障害があることが特定されたとすると、以降のモジュールの呼び出しではRモジュール43を使用する。これにより、いずれかのモジュールに障害があることが判明した後でも、二つのモジュールによる照合を継続することができ、第1アプリケーション12が誤ったモジュールの出力に基づく動作を防止することができる。すなわち、一つのモジュールが障害を有していても処理を継続できる効果と、2種類のモジュールの出力の照合による安全性の担保の効果の両方を得ることができる。

[0044] さらに、ECU100における計算資源の利用は、機能の実現を行う第1アプリケーション12が大半であり、モジュールによる使用は少ない。上述したようにECU100は、モジュールの出力を比較するので、第1アプリケーション12の出力を比較する場合に比べてCPU110の演算資源およびメモリ120の記憶資源の消費を削減することができる。さらにECU100は、障害特定処理が実行されるまでは2種のモジュールしか実行しないことから、同様に演算資源の利用も削減することができる。

[0045] 上述した第1の実施の形態によれば、次の作用効果が得られる。

(1) ECU100は、同一の機能を有し異なる実装がなされたPモジュール

ル41、Qモジュール42、およびRモジュール43が格納されるメモリ120と、Pモジュール41、Qモジュール42、およびRモジュール43を実行するCPU110と、CPU110によるPモジュール41の出力およびQモジュール42の出力に基づき障害を検出する第1モジュール実行部13および第2モジュール実行部23（図7のS706～S710）と、を備える。ECU100はさらに、図7に示す障害検出処理により障害が検出されるとRモジュール43の使用を開始し、Pモジュール41の出力と、Qモジュール42の出力と、Rモジュール43の出力と、に基づき障害のあるソフトウェアモジュールを特定する第1モジュール実行部13および第2モジュール実行部23（図8のS804）と、を備える。すなわちECU100は、3つのモジュールを備えるが障害が検出されるまではそのうち2つのモジュールしか使用せず、障害が検出されて障害のあるモジュールを特定するために3つ目のモジュールを使用する。そのためECU100は、限られた演算資源を用いて障害の検出および障害の特定を実現することができる。

[0046] (2) ECU100は、障害があることが特定されたソフトウェアモジュールの使用を停止する（図8のS804～S805）。そのためECU100は、障害のないソフトウェアモジュールを用いて動作を継続することができる。

[0047] (3) ECU100は、第1モジュール実行部13がPモジュール41およびQモジュール42のいずれか一方を障害があるソフトウェアモジュールとして特定すると（図8のS804：Yes、S805）、Pモジュール41およびQモジュール42のうち障害があることが特定されていないソフトウェアモジュール、およびRモジュール43の出力に基づき障害を検出する（図7のS702）。

[0048] (4) Pモジュール41、Qモジュール42、およびRモジュール43には、32ビットCPU用のソフトウェアモジュールと、64ビットCPU用のソフトウェアモジュールとが含まれる。幅広い環境での実行を可能とすべく、換言すると複数種類のCPUに対応する目的でライブラリが32ビットC

P U用と64ビットCPU用のそれぞれについて作成されることが多い。それらを利用することにより、独自の実装を行ったライブラリを1つ作成することで上述した第1の実施の形態を実施することができる。

[0049] (5) Pモジュール41およびQモジュール42は、異なるCPUコアで並列に実行される。そのため迅速にPモジュール41およびQモジュール42の出力を得て迅速に障害の有無を判断することができる。

[0050] (6) Pモジュール41、Qモジュール42、およびRモジュール43のそれぞれには複数の関数が含まれる。複数の関数のそれぞれについて障害検出処理のS706およびS707の動作を規定する第1出力照合表17を備える。第1モジュール実行部13は、第1出力照合表17を参照して関数ごとに障害検出の条件を判断する。そのためECU100は、正常な動作が行われても出力値が一定ではない関数、たとえば動的にメモリ領域を確保し、確保した領域の先頭アドレスを返す関数などを含むソフトウェアモジュールにも対応することができる。

[0051] (変形例1)

図7においては、常に出力の照合を行うこととしたが、照合をモジュールの呼び出しのたびに行わないとしてもよい。たとえば、モジュールの呼び出しが一定の回数、たとえば3回ごとに照合を行ってもよい。この場合には、カウンタを利用して呼び出し回数を計数し、3の倍数となった場合にのみ照合を行う。また一定の時間、たとえば10msごとに照合を行ってもよい。この場合には、時間を計測するタイマデバイスなどを利用して、前回の照合から10ms以上経過した場合に照合を行う。照合を行わなかった場合には、そのモジュールの呼び出しの入力と出力は、第1モジュール入力表15と第1モジュール出力表16に追記される。

[0052] この場合はクリア処理(S711)を行わないので、まとめて照合を後で行うことができる。第1アプリケーション12には得たモジュールの出力をそのまま渡せばよい。これにより、以下のような効果が得られる。照合処理は、毎回行うとその照合処理の時間によって、第1アプリケーション12の

制御処理に影響がある可能性がある。そこで何らかの基準にしたがい、まとめて照合処理を行うことによってその影響を低減できる。さらに周期処理の待ち時間など影響の少ないタイミングで照合処理を行うことも可能となる。なおまとめて照合を行う場合には、アプリケーションには間違った値がすでに出力された可能性がある。したがって障害が検出された場合には、障害が検出されたモジュールを利用したアプリケーションは、他方の正常であるアプリケーションの状態を引き継いで実行を継続させるなどの対処が必要である。

[0053] (変形例 2)

図 7 および図 8 に示すフローチャートでは、障害が特定された後でもモジュール選択を行い二重系を維持するものとした。しかし障害が特定された場合は、障害が検出されたモジュールを利用したアプリケーションを停止させ、以降は他方のアプリケーションのみ動作を継続させてもよい。本変形例は、障害が検出されたモジュールを利用したアプリケーションに対して他方の正常であるアプリケーションの状態を引き継いで実行を継続させることが困難な場合に特に有効である。すなわち本変形は、再起動や修復までの一時的な処理として有用である。

[0054] (変形例 3)

上述した実施の形態では、第 1 プログラム 1 1 および第 2 プログラム 2 1 の両方が図 8 のフローチャートで示す処理を実行していた。しかし第 1 プログラム 1 1 と第 2 プログラム 2 1 がそれぞれ代替モジュールとして選択するモジュールが同一、たとえば上述のように R モジュール 4 3 であるときには、片方の処理のみで十分である。したがって、代替モジュールの実行と比較処理 (S 8 0 1 ~ S 8 0 4) は、いずれか片方のプログラムのモジュール実行部のみが行い、その結果を他方のモジュール実行部に伝達してもよい。本変形例によれば、照合処理を行わない他方のプログラムあるいは CPU コアにおいては、その間別の処理を実行することができ、CPU の利用効率を向上させることができる。

## [0055] (変形例4)

ECU100は、Pモジュール41、Qモジュール42、およびRモジュール43の3つのモジュールを有したが、4種類以上のモジュールを有してもよい。たとえば、第1プログラム11はPモジュール41とRモジュール43、第2プログラム21はQモジュール42とZモジュールを有してもよい。この場合は障害特定の処理において、第1プログラム11の第1モジュール実行部13はRモジュール43、第2プログラム21はZモジュールを実行する。

[0056] なおモジュールが偶数個なので2:2になり多数決ができない場合には、奇数個のモジュールを追加して、それぞれのモジュール実行部で複数個のモジュールを実行してもよい。本変形例によれば、より多くの種類のモジュールを利用することで、障害の確率をさらに下げることができる。また同一の機能を有し実装が異なるN種類のモジュールを用いることにより、N-2個のモジュールに障害が発生するまで第1の実施の形態において説明した動作、すなわち障害検出処理および障害特定処理を継続することができる。これによって、動作の継続性や可用性を向上させる効果が得られる。

## [0057] (変形例5)

第1出力照合表17の「出力比較区分」は、図4において`Compare`、`Ignore`、`Compare if error`の3種類のみを示した。しかし「出力比較区分」はこれに限定されない。たとえば、エラーの種類は問わず成功時のみ比較する区分を設けてもよいし、エラーと成功のどちらかであったかのみを比較する区分を設けてもよい。さらに出力値の一部、たとえば上位3ビットや下位1ビットのみ比較する区分を設けてもよい。本変形例によれば、利用するモジュールの仕様にあわせて区分を作成することで、障害判定の精度を向上させ、迅速な障害の検出が行うことができる。

## [0058] (変形例6)

上述した実施の形態では、第1プログラム11および第2プログラム21

は、それぞれ1つのアプリケーションのみを備えた。しかし各プログラムがモジュールを使用するアプリケーションを複数備えてもよい。この場合は、第1モジュール入力表15には、呼び出し元のアプリケーションを識別する列が追加され、呼び出し元のアプリケーションと呼び出された関数との関係を示す情報が格納される。本変形例によれば、モジュールが複数のアプリケーションから共有して利用される場合でも、上述した実施の形態における効果を得ることができる。

[0059] (変形例7)

上述した実施の形態では、ソフトウェアモジュールであるPモジュール41やRモジュール43は、第1アプリケーション12と異なるものとして説明した。しかしソフトウェアモジュールは静的にリンクされるライブラリであり、第1アプリケーション12に予め組み込まれてもよい。この場合は、たとえばPモジュール41への入力およびPモジュール41からの出力は、第1アプリケーション12の内部で実行される。

[0060] (変形例8)

図8のS801において、代替モジュールは以下のように選択された。第1モジュール実行部13は、第1モジュール管理表18において動作状況が“OK”であるモジュールであって、自身が使用したモジュールおよびS705によって得た出力を行ったモジュール以外のモジュールを選択した。すなわちS801では第1モジュール実行部13および第2モジュール実行部23のいずれでも使われなかったモジュールを選択した。しかし自身が実行していない方のモジュール、すなわちS705によって得た出力を行ったモジュールもさらに実行してもよい。

[0061] 本変形例によれば、次の作用効果が得られる。

(7) CPU110は、第1CPUコア111および第2CPUコア112を備える。障害特定部は、第1CPUコア111が演算したPモジュール41の出力と、第2CPUコア112が演算したQモジュール42の出力とが一致しない場合に、第2CPUコア112が演算したPモジュール41の出

力と、第1CPUコア111が演算したQモジュール42の出力とを用いて、第1CPUコア111または第2CPUコア112に障害があることを特定する。

[0062] そのため本変形例ではハードウェア障害にも対応することができる。たとえば第1プログラム11が実行したPモジュール41の出力値を出力値A、第2プログラム21が実行したQモジュール42の出力値を出力値B1、第1プログラム11が実行したQモジュール42の出力値を出力値B2、第1プログラム11が実行した代替モジュールであるRモジュール43の出力値を出力値Cとする。このときECU100は、出力値B1と出力値B2とを比較することにより、出力値Aと出力値B1の不一致が、出力値B1を実行したCPUコア等のハードウェア障害によるものかを判定することができる。

[0063] (変形例9)

図7のS702におけるモジュールの選択は、あらかじめ設定された優先度に基づいて行われてもよい。この場合はたとえば、第1モジュール管理表18のそれぞれの行に優先度が設定される。そして第1モジュール実行部13は、動作状況が`OK`であるモジュールであって優先度が最も高いモジュールを選択する。第2モジュール管理表28および第2モジュール実行部23も同様である。

[0064] (変形例10)

第1CPUコア111および第2CPUコア112は、CPUコア間の通信を行う代わりに、必要な情報が格納されているメモリ120の領域を読み込んでもよい。ただしこの場合は、他方のCPUコアもモジュールの出力を得ていることを確認する必要がある。この確認は、同期的な通信を用いる場合には明らかであるし、第1モジュール出力表16に記録されている行数を検査することで行ってもよい。なお他方のCPUコアがモジュールの出力を得ていない場合には、他方のCPUコアがモジュールの出力を得るまで待機する。

## [0065] (変形例 1 1)

上述した複数のモジュールは、同一のアーキテクチャを対象とした同一の機能を有する異なるバージョンや異なるリビジョンのライブラリであってもよい。バージョンやリビジョンの違いにより、実装が異なる場合があるからである。

さらに上述した複数のモジュールは、全ての機能が同一でなくてもよく、アプリケーションから使用される関数を含んでいれば、その他の関数を含んでいてもよい。たとえばアプリケーションがモジュールに備えられる関数 A、関数 B、関数 C を呼び出す場合に、あるモジュールは関数 A～C のみを備え、他のモジュールは関数 A～C に加えて関数 D をさらに備えてもよい。

## [0066] 一第 2 の実施の形態一

図 9～図 10 を参照して、制御装置である ECU の第 2 の実施の形態を説明する。以下の説明では、第 1 の実施の形態と同じ構成要素には同じ符号を付して相違点を主に説明する。特に説明しない点については、第 1 の実施の形態と同じである。本実施の形態では、主に、CPU コアを 1 つのみ備える点で、第 1 の実施の形態と異なる。

[0067] 図 9 は、第 2 の実施の形態における ECU 100A の構成を表すブロック図である。図 9 において、ECU 100A の CPU 110A は、第 1 CPU コア 111 を有する。第 1 CPU コア 111 は、第 1 プログラム 11 を動作させる。第 1 プログラム 11 の第 1 モジュール実行部 13B は、P モジュール 41、Q モジュール 42、および R モジュール 43 の 3 つのモジュールを動作させる。

[0068] 図 10 は、第 2 の実施の形態における第 1 モジュール実行部 13B の動作を表すフローチャートである。図 10 では第 1 の実施の形態における図 7 と同一の処理には同一のステップ番号を付す。すなわち図 7 との相違点は、S702 の代わりに S702A を実行する点、S703 の代わりに S703A を実行する点、および S705 を実行しない点である。その他のステップは図 7 と同様なので説明を省略する。

[0069] S 7 0 1 を実行すると第 1 モジュール実行部 1 3 B は、第 1 アプリケーション 1 2 の呼び出しを実行するためのモジュールを 2 つ選択する (S 7 0 2 A)。この選択方法は第 1 の実施の形態と同様である。そして第 1 モジュール実行部 1 3 B は、選択したそれぞれのモジュールの関数を実行して出力値を得る (S 7 0 3 A)。これ以後の処理は第 1 の実施の形態と同様なので説明を省略する。

[0070] 本実施の形態では、第 1 CPU コア 1 1 1 上で第 1 アプリケーション 1 2 がモジュールを呼び出すと、第 1 モジュール実行部 1 3 は P モジュール 4 1 と Q モジュール 4 2 とを用いて計算を実行し、その出力値を比較する。すなわち、第 1 の実施形態では第 1 プログラム 1 1 と第 2 プログラム 2 1 とで別々に行っていたモジュールの実行を 1 つのプログラム内で行うものである。したがって、障害検出のために行う処理としては、2 つの異なるモジュールに対して同じ入力を与えてその出力を比較する処理という点では、第 1 の実施の形態と同様である。また障害を特定するために代替モジュールを選択して実行しその出力を得る点も第 1 の実施の形態と同様である。さらに、代替モジュールの実行および比較を行うプログラムが 1 つである点は、第 1 の実施の形態における変形例 3 と同様である。すなわち本実施の形態でも、第 1 の実施形態と同様の障害の検出と特定が可能である。

[0071] 本実施の形態を第 1 の実施の形態と比較すると、モジュール呼び出しのたびに、1 つの CPU コアで 2 つのモジュールを実行するために、障害検出のために必要な 1 つの CPU コアあたりの演算資源は増加する。しかし本実施の形態では複数の CPU コアを必要としないため、ECU 1 0 0 上で必要な CPU コアの数減らすことができる。そのため ECU 1 0 0 のコストの削減効果、または使用しなくなる CPU コアで別の処理を行うことで、効率を増加させる効果が得られる。なお、本実施形態においても、第 1 の実施形態で挙げた変形例は同様に適用可能である。

[0072] 一第 3 の実施の形態一

図 1 1 ~ 図 1 2 を参照して、制御装置である ECU の第 3 の実施の形態を

説明する。以下の説明では、第1の実施の形態と同じ構成要素には同じ符号を付して相違点を主に説明する。特に説明しない点については、第1の実施の形態と同じである。本実施の形態では、主に、障害の特定に別のECUを用いる点で、第1の実施の形態と異なる。

[0073] 図11は、第3の実施の形態における制御システムSの構成を表すブロック図である。制御システムSは、第1ECU1001と、第2ECU1002と、代替実行ECU1003とを備える。第1ECU1001、第2ECU1002、および代替実行ECU1003はそれぞれ、第1ネットワークインタフェース119、第2ネットワークインタフェース219、および第3ネットワークインタフェース1110を備え、ネットワークXを介して相互に通信ができる。ネットワークXの物理的な特性、およびネットワークXにおいて使用される通信プロトコルは特に制限されない。ネットワークXはたとえば、CAN (Car Area Network) またはIEEE 802.3に対応する。

[0074] 第1ECU1001は、CPU110と、メモリ120とを備える。第2ECU1002は、CPU210と、メモリ220とを備える。代替実行ECU1003は、CPU310とメモリ320とを備える。CPU110、CPU210、およびCPU310のハードウェア構成は第1の実施の形態におけるCPU110と同様である。図11に示すように、Pモジュール41は第1ECU1001に格納され、Qモジュール42は第2ECU1002に格納され、Rモジュール43は代替実行ECU1003に格納される。第1モジュール実行部13Cは、第1の実施の形態における第1モジュール実行部13に相当するが、Rモジュール43を備えない点が第1モジュール実行部13と異なる。第2モジュール実行部23Cは、第1の実施の形態における第2モジュール実行部23に相当するが、Rモジュール43を備えない点が第2モジュール実行部23と異なる。

[0075] 第1ECU1001のCPU110に内蔵される第1CPUコア111は、第1の実施の形態における第1CPUコア111と同様の動作を行う。た

だし本実施の形態では、第2CPUコア112との通信はネットワークXを介して行われる点が第1の実施の形態と異なる。また第1ECU1001の第1モジュール実行部13Cは、Rモジュール43を自ら実行せずに代替実行ECU1003から実行結果を取得する。

[0076] 第2ECU1002のCPU210に内蔵される第2CPUコア112は、第1の実施の形態における第2CPUコア112と同様の動作を行う。ただし本実施の形態では、第1CPUコア111との通信はネットワークXを介して行われる点が第1の実施の形態と異なる。また第2ECU1002の第2モジュール実行部23Cは、Rモジュール43を自ら実行せずに代替実行ECU1003から実行結果を取得する。

[0077] 代替実行ECU1003は、障害検出時に障害のあるモジュールを特定するためにRモジュール43を実行する。代替実行ECU1003は、障害の特定を行う代替モジュール実行部1140をメモリ320に備える。代替モジュール実行部1140はCPUコア1151により実行される。代替モジュール実行部1140は、障害特定のためのRモジュール43と、モジュール入力受信部1141と、モジュール出力送信部1142とを備える。モジュール入力受信部1141は、第3ネットワークインタフェース1110を介して、モジュールの関数を実行するための入力を受信する。モジュール出力送信部1142は、第3ネットワークインタフェース1110を介して、モジュールの関数を実行した出力を送信する。

[0078] 代替モジュール実行部1140は、第1ECU1001または第2ECU1002から入力値を受信すると、Rモジュール43にその入力値を入力する。そして代替モジュール実行部1140は得られた演算結果を入力値の送信元へ送信する。

[0079] 図12は、本実施形態における第1モジュール実行部13Cおよび第2モジュール実行部23Cにおける障害特定処理を示すフローチャートである。図12では第1の実施の形態における図8と同一の処理には同一のステップ番号を付す。すなわち図8との相違点は、S801の代わりにS1201を

実行する点、S 8 0 2 の代わりに S 1 2 0 2 を実行する点である。その他のステップは図 8 と同様なので説明を省略する。以下では第 1 モジュール実行部 1 3 C および第 2 モジュール実行部 2 3 C を代表して第 1 モジュール実行部 1 3 C の動作を説明する。

[0080] 第 1 モジュール実行部 1 3 C は、障害特定処理を開始すると、代替実行 ECU 1 1 3 0 に入力値を送信する (S 1 2 0 1)。そして第 1 モジュール実行部 1 3 C は、代替実行 ECU 1 1 3 0 から代替モジュール実行部 1 1 4 0 による R モジュール 4 3 を用いた演算結果を受信する (S 1 2 0 2)。その他のステップは図 8 と同様なので説明を省略する。

[0081] 上述した第 3 の実施の形態によれば、各モジュールを実行する ECU が異なる場合でも、第 1 の実施の形態と同様に障害の検出と特定が可能である。

[0082] (第 3 の実施の形態の変形例 1)

上述した第 3 の実施の形態では、第 1 プログラム 1 1 および第 2 プログラム 2 1 が異なる ECU において実行された。しかし第 1 の実施形態における ECU 1 0 0 と同様に、同一 ECU の異なる CPU コアにおいて実行してもよい。この場合は、障害検出の処理は第 1 の実施例と同じであり、障害特定の処理のみが第 3 の実施の形態で示した処理となる。これによって、必要な ECU の数を削減することができる。

[0083] (第 3 の実施の形態の変形例 2)

第 1 プログラム 1 1 および第 2 プログラム 2 1 が同一 ECU の異なる CPU コアにおいて実行され、さらに同一の ECU が、代替モジュール実行部 1 1 4 0 を実行する第 3 の CPU コアを備えてもよい。この場合は、ネットワークインタフェース 1 1 1 0 および通信ネットワークバス 1 1 2 0 は、同一 ECU 内のコア間通信を行うものとして実現できる。これによって、必要な ECU の数を 1 つとすることができる。

[0084] (第 3 の実施の形態の変形例 3)

上述した第 3 の実施の形態では、第 1 ECU 1 1 0 1 および第 2 ECU 1 1 0 2 を一組のみ示した。しかし同様の処理を行う ECU の組が複数存在し

てもよい。そしてこれらのECUの組において、代替実行ECU1130を共有してもよい。代替実行ECU1130は、障害が検出されたときのみ動作するため、代替実行ECU1130の稼働率は低い。そのため複数組のECUからの要求に応じて処理を実行することで、代替実行ECU1130の稼働率を向上させることができる。

[0085] またこの場合に、複数組のECUはそれぞれ同一のモジュールを使用してもよいし、異なるモジュールを使用してもよい。たとえば機能Aを有するモジュールS、モジュールT、モジュールU、および機能Bを有するモジュールV、モジュールW、モジュールXが存在し、モジュールS、モジュールT、モジュールV、およびモジュールWをそれぞれ異なるECUに格納される場合を想定する。この場合に、代替実行ECU1130がモジュールXおよびモジュールWを備え、モジュールS、モジュールT、モジュールV、およびモジュールWを備える各ECUから入力される入力値に応じてモジュールXまたはモジュールWの算出結果を各ECUに送信すると以下の効果が得られる。すなわち、必要な代替実行ECUの数を削減することができる。

[0086] (第3の実施の形態の変形例4)

上述した第3の実施の形態の変形例3において、代替実行ECU1130は複数あってもよい。さらに、いずれかのプログラムを実行するECUが、代替実行ECU1130としての機能を兼ね備えてもよい。これにより、変形例3の場合と比較して、障害特定ECU一つが故障しても障害特定の機能を他のECUによって継続することができる。

[0087] プログラムを実行するECUが代替実行ECU1130としての機能を兼ね備える場合はさらに、それぞれの代替実行ECU1130は他の代替実行ECU1130における処理負荷の大きさと障害状況を監視する機能を有してもよい。この場合は代替モジュール実行部1140は、CPU310の現在の計算負荷の大きさ、およびRモジュール43を用いた演算処理を行っているか否かを示す情報を周囲の装置に送信する機能を有する。第1モジュール実行部13Cおよび第2モジュール実行部23Cは、それぞれの代替実行

ＥＣＵ１１３０から受信した情報に基づきＲモジュール４３を用いた演算を実行させる代替実行ＥＣＵ１１３０を決定する。具体的には第１モジュール実行部１３Ｃおよび第２モジュール実行部２３Ｃは、Ｒモジュール４３を用いた演算を行っておらず、かつＣＰＵ３１０の現在の計算負荷が最も低い代替実行ＥＣＵ１１３０を特定し、その代替実行ＥＣＵ１１３０にＲモジュール４３を用いた演算を実行させる。

[0088] 本変形例によれば、次の作用効果が得られる。

(８) 第１ＥＣＵ１００１は、他の演算装置と通信を行う第１ネットワークインタフェース１１９と、現在の計算負荷の大きさを通信部を介して他の演算装置に出力する代替モジュール実行部１１４０とを備える。第１モジュール実行部１３Ｃは、受信した現在の計算負荷に基づき第３のソフトウェアモジュールの出力を算出させる代替実行ＥＣＵ１００３を決定する。そのため複数の代替実行ＥＣＵ１００３の負荷を平準化することができる。

[0089] (第３の実施の形態の変形例５)

上述した第３の実施の形態では、障害特定処理は、第１ＥＣＵ１００１の第１モジュール実行部１３Ｃ、または第２ＥＣＵ１００２の第２モジュール実行部２３Ｃにおいて実行された。しかし障害特定処理は、代替モジュール実行部１１４０において実行されてもよい。この場合に代替実行ＥＣＵ１００３は、第１ＥＣＵ１００１および第２ＥＣＵ１００２において実行されたモジュールの出力値も取得する。代替モジュール実行部１１４０における障害特定処理は、図１２におけるＳ８０３、Ｓ８０４と同様である。代替モジュール実行部１１４０は、障害が特定されたモジュールを示す情報を各ＥＣＵに送信する。第１モジュール実行部１３Ｃおよび第２モジュール実行部２３Ｃはその情報を受信し、Ｓ８０５以降の処理を実行する。本変形例によれば、第１ＥＣＵ１００１および第２ＥＣＵ１００２の第１モジュール実行部１３Ｃおよび第２モジュール実行部２３Ｃの処理負荷と代替モジュール実行部１１４０の処理負荷のバランスをとることができる。

[0090] 上述した実施の形態、および変形例では各ＣＰＵコアが１つのプログラム

を動作させるとしたが、それぞれのCPUコアが複数のプログラムを動作していてもよいし、複数のCPUコアが一つのプログラムを共有してもよい。この場合、それぞれのCPUコアが実行するコードの一部または全部や、CPUコアが使用するデータの一部または全部が異なり、異なるコードとデータが必要なだけ格納されている構成であってもよい。また、CPU110が実行するコードや読み込みのみ行うデータなどを不揮発性のメモリに配置し、読み書きするデータを揮発性のメモリに格納する構成であってもよい。

[0091] 上述した各実施の形態および変形例は、それぞれ組み合わせてもよい。上記では、種々の実施の形態および変形例を説明したが、本発明はこれらの内容に限定されるものではない。本発明の技術的思想の範囲内で考えられるその他の態様も本発明の範囲内に含まれる。

[0092] 次の優先権基礎出願の開示内容は引用文としてここに組み込まれる。

日本国特許出願2017-136657（2017年7月12日出願）

## 符号の説明

[0093] 3…モジュール

11…第1プログラム

12…第1アプリケーション

13、13B、13C…第1モジュール実行部

15…第1モジュール入力表

16…第1モジュール出力表

17…第1出力照合表

18…第1モジュール管理表

21…第2プログラム

22…第2アプリケーション

23、23C…第2モジュール実行部

25…第2モジュール入力表

26…第2モジュール出力表

27…第2出力照合表

2 8 …第2 モジュール管理表

4 1 …P モジュール

4 2 …Q モジュール

4 3 …R モジュール

1 1 0 …C P U

1 1 1 …第1 C P U コア

1 1 2 …第2 C P U コア

## 請求の範囲

- [請求項1]           同一の機能を有し異なる実装がなされた第1のソフトウェアモジュール、第2のソフトウェアモジュール、および第3のソフトウェアモジュールが格納される記憶部と、
- 前記第1のソフトウェアモジュール、前記第2のソフトウェアモジュール、前記第3のソフトウェアモジュールを実行する演算部と、
- 前記演算部による前記第1のソフトウェアモジュールの出力、および前記第2のソフトウェアモジュールの出力に基づき障害を検出する検出部と、
- 前記検出部が障害を検出すると、前記第3のソフトウェアモジュールの使用を開始し、前記第1のソフトウェアモジュールの出力と、前記第2のソフトウェアモジュールの出力と、前記第3のソフトウェアモジュールの出力と、に基づき障害のあるソフトウェアモジュールを特定する障害特定部と、を備える演算装置。
- [請求項2]           請求項1に記載の演算装置において、
- 前記障害特定部により障害があることが特定されたソフトウェアモジュールの使用を停止する演算装置。
- [請求項3]           請求項2に記載の演算装置において、
- 前記検出部は、前記障害特定部が前記第1のソフトウェアモジュールおよび前記第2のソフトウェアモジュールのいずれか一方を障害があるソフトウェアモジュールとして特定すると、前記第1のソフトウェアモジュールおよび前記第2のソフトウェアモジュールのうち障害があることが特定されていないソフトウェアモジュール、および前記第3のソフトウェアモジュールの出力に基づき障害を検出する演算装置。
- [請求項4]           請求項1に記載の演算装置において、
- 前記第1のソフトウェアモジュール、前記第2のソフトウェアモジュール、および前記第3のソフトウェアモジュールには、32ビット

CPU用のソフトウェアモジュールと、64ビットCPU用のソフトウェアモジュールとが含まれる演算装置。

[請求項5]           請求項1に記載の演算装置において、  
前記第1のソフトウェアモジュール、および前記第2のソフトウェアモジュールは並列に実行される演算装置。

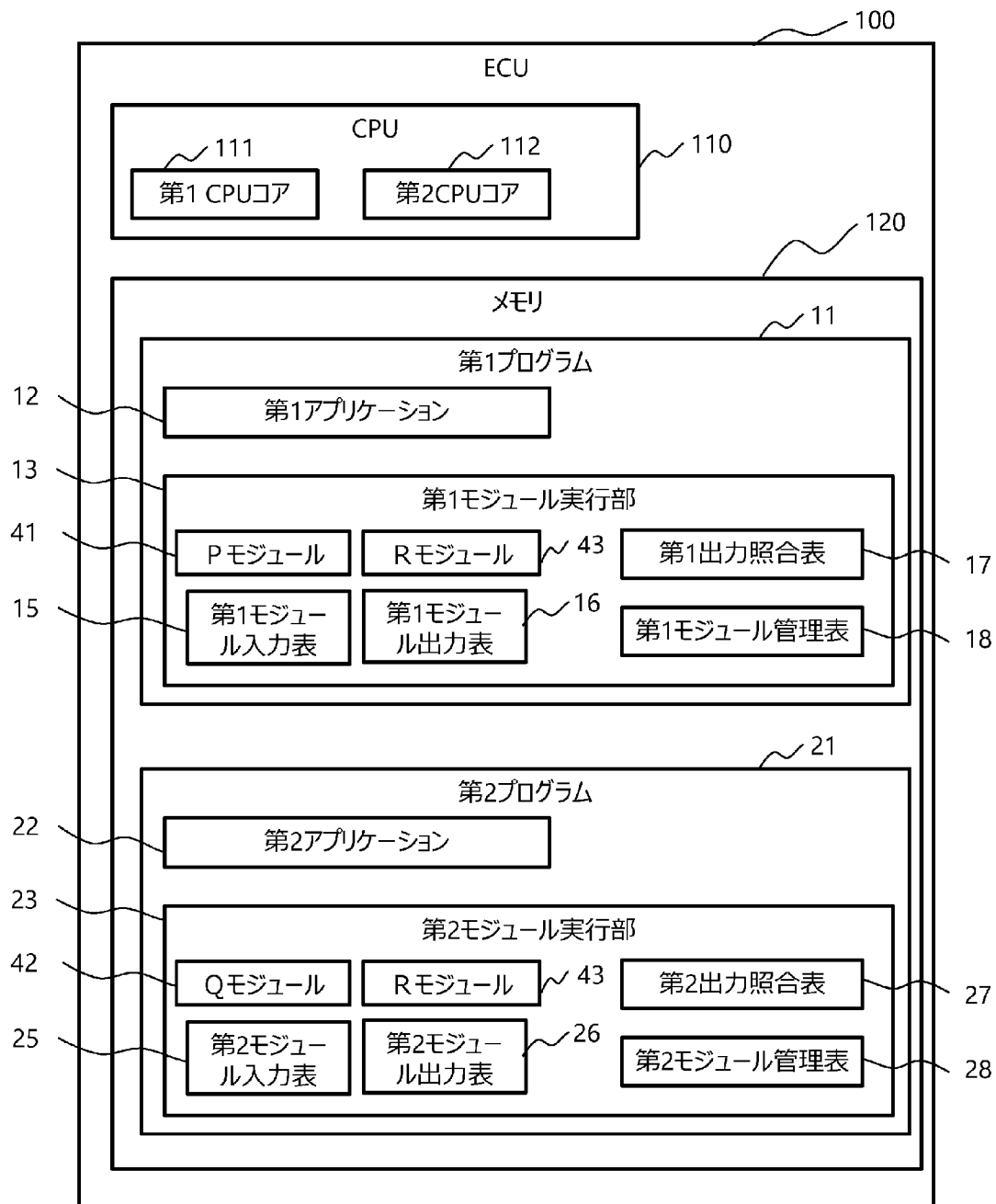
[請求項6]           請求項1に記載の演算装置において、  
前記演算部は第1コアおよび第2コアを備え、  
前記障害特定部は、前記第1コアが演算した前記第1のソフトウェアモジュールの出力と、前記第2コアが演算した前記第2のソフトウェアモジュールの出力とが一致しない場合に、前記第2コアが演算した前記第1のソフトウェアモジュールの出力と、前記第1コアが演算した前記第2のソフトウェアモジュールの出力とを用いて、前記第1コアまたは前記第2コアに障害があることを特定する演算装置。

[請求項7]           請求項1に記載の演算装置において、  
他の前記演算装置と通信を行う通信部と、  
現在の計算負荷の大きさを前記通信部を介して前記他の演算装置に出力する負荷算出部とをさらに備え、  
前記障害特定部は、受信した現在の前記計算負荷に基づき前記第3のソフトウェアモジュールの出力を算出させる前記演算装置を決定する演算指令部とをさらに備える演算装置。

[請求項8]           請求項1に記載の演算装置において、  
前記第1のソフトウェアモジュール、前記第2のソフトウェアモジュール、および前記第3のソフトウェアモジュールのそれぞれには複数の関数が含まれ、  
前記複数の関数のそれぞれについて前記検出部の動作を規定する出力照合表をさらに備え、  
前記検出部は、前記出力照合表を参照して前記関数ごとに障害検出の条件を判断する演算装置。

[図1]

図 1



[図2]

図 2

15

| 第1モジュール入力表 |            |                            |
|------------|------------|----------------------------|
| No.        | 呼び出し関数     | 入力                         |
| 1          | Function A | Param 1: 123, Param 2: 111 |
| 2          | Function B | Param 1: 250               |
| 3          | Function C | Param 1: 100, Param 2: 111 |
| ...        | ...        | ...                        |

[図3]

図 3

| 第1モジュール出力表 |     |
|------------|-----|
| No.        | 出力  |
| 1          | 0   |
| 2          | 100 |
| 3          | -1  |
| ...        | ... |

16

[図4]

図 4

| 第1出力照合表    |                  |
|------------|------------------|
| 関数         | 出力比較区分           |
| Function A | Compare          |
| Function B | Ignore           |
| Function C | Compare if error |
| ...        | ...              |

17

[図5]

図 5

| 第1モジュール管理表 |      |
|------------|------|
| モジュール      | 動作状況 |
| モジュールA     | NG   |
| モジュールB     | OK   |
| モジュールC     | OK   |

18

[図6]

図 6

15

| 第1モジュール入力表 |            |                            |
|------------|------------|----------------------------|
| No.        | 呼び出し関数     | 入力                         |
| 1          | Function A | Param 1: 123, Param 2: 111 |
| 2          | Function B | Param 1: 250               |
| 3          | Function C | Param 1: 100, Param 2: 111 |

(a)

16

| 第1モジュール出力表<br>(Pモジュール) |     |
|------------------------|-----|
| No.                    | 出力  |
| 1                      | 0   |
| 2                      | 100 |
| 3                      | -1  |

(b)

26

| 第2モジュール出力表<br>(Qモジュール) |     |
|------------------------|-----|
| No.                    | 出力  |
| 1                      | 0   |
| 2                      | 200 |
| 3                      | -2  |

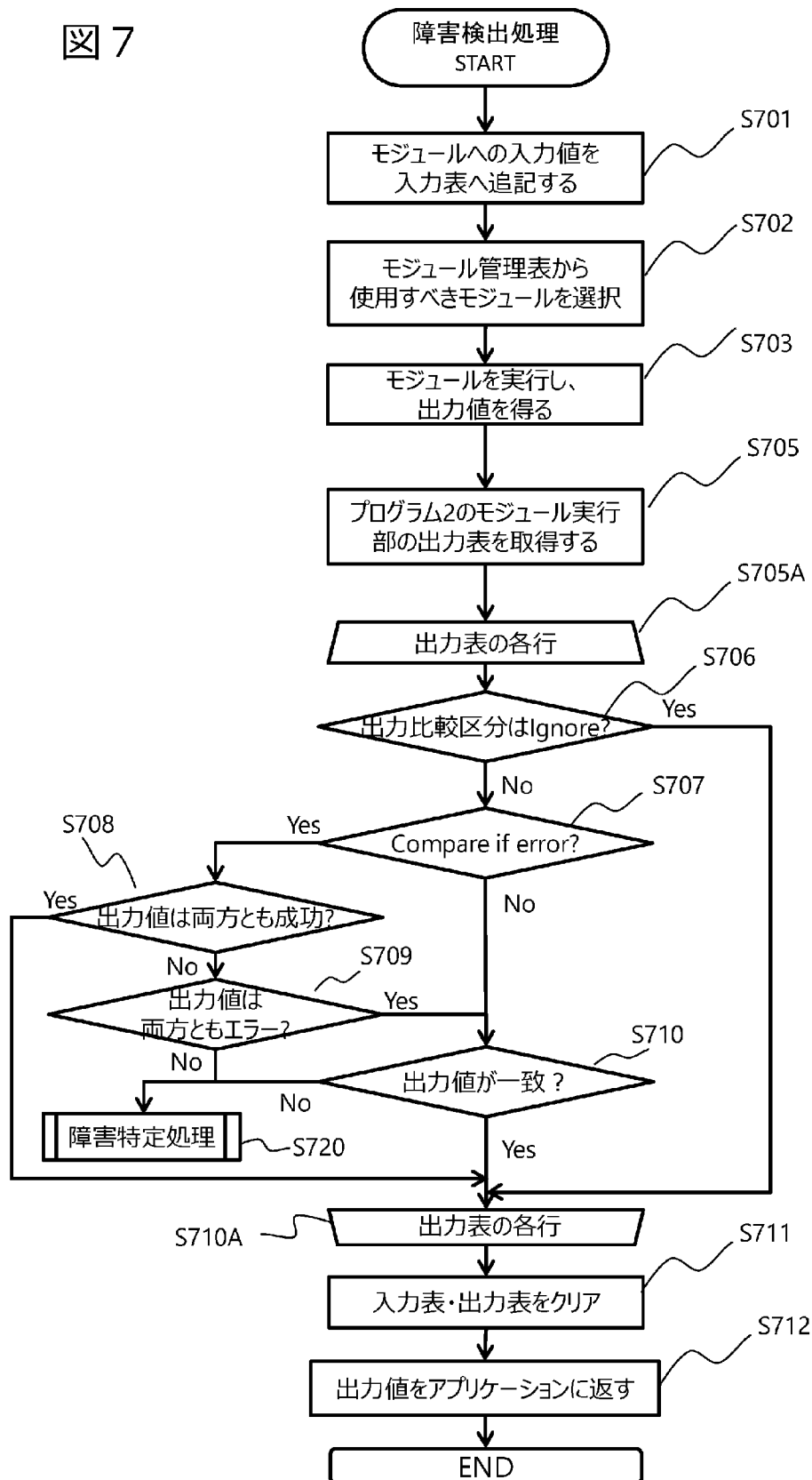
(c)

16A

| 第1モジュール出力表<br>(Rモジュール) |     |
|------------------------|-----|
| No.                    | 出力  |
| 1                      | 0   |
| 2                      | 220 |
| 3                      | -2  |

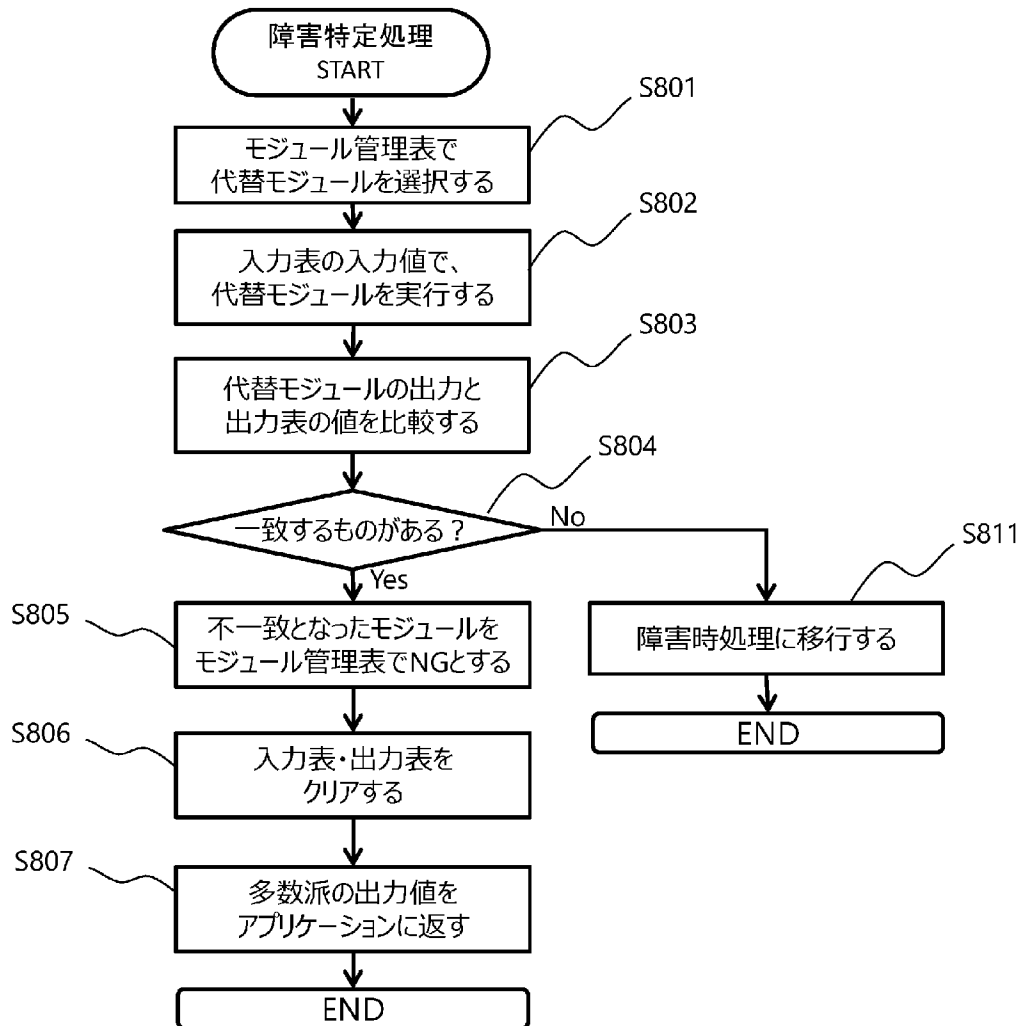
(d)

[図7]



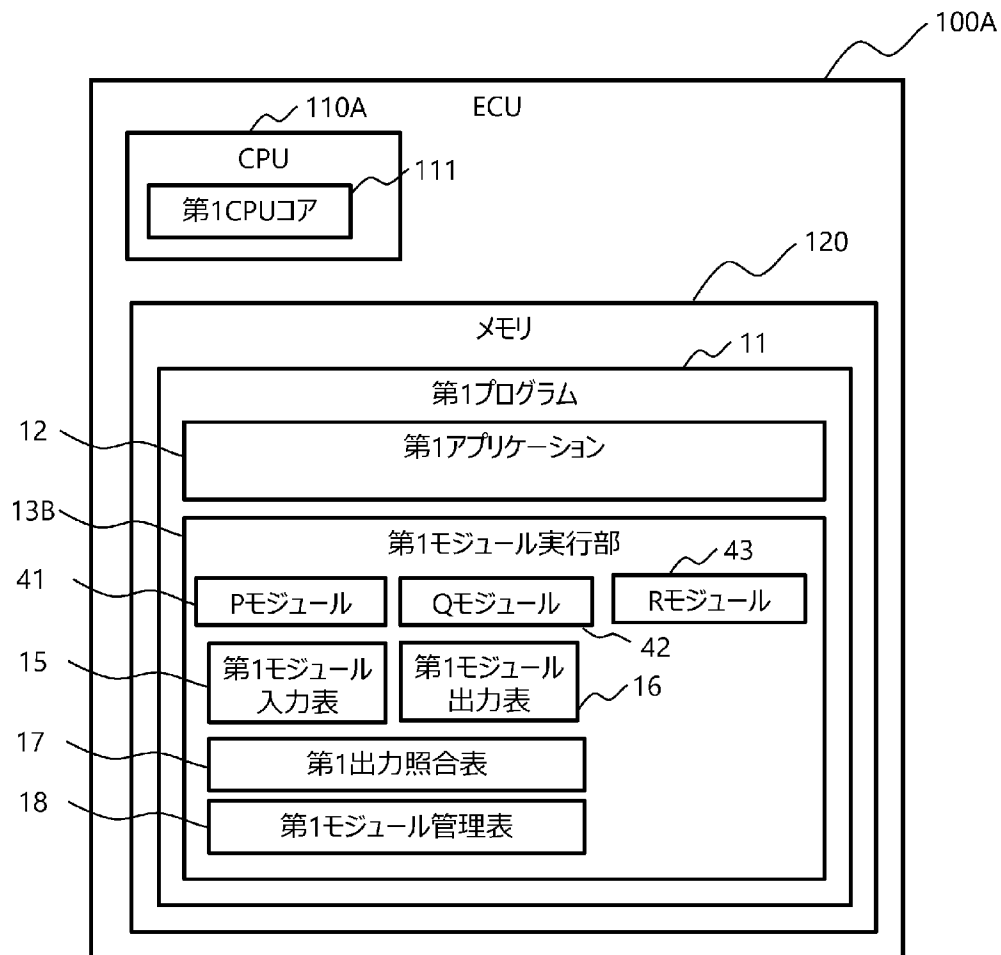
[図8]

図 8



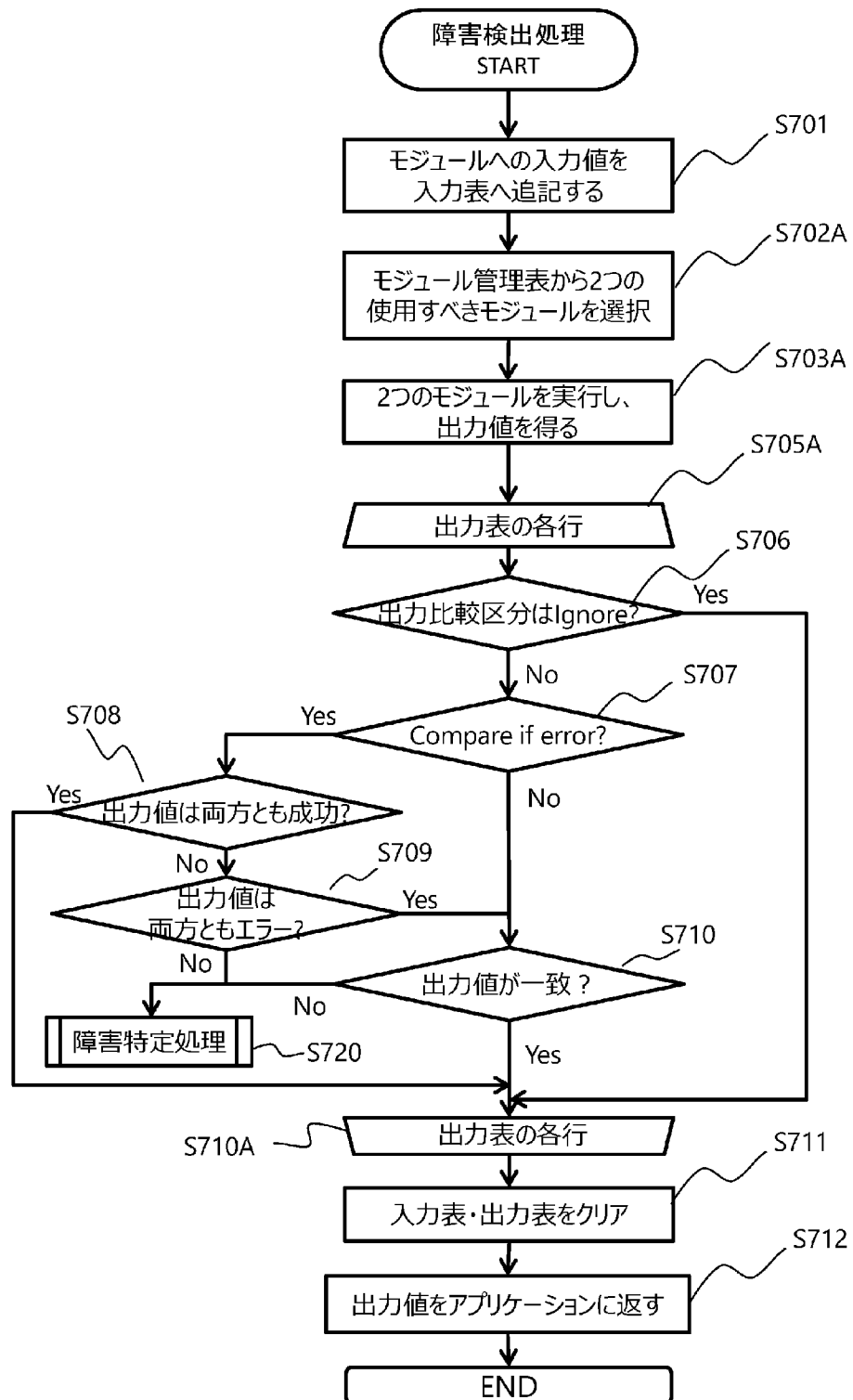
[図9]

図 9

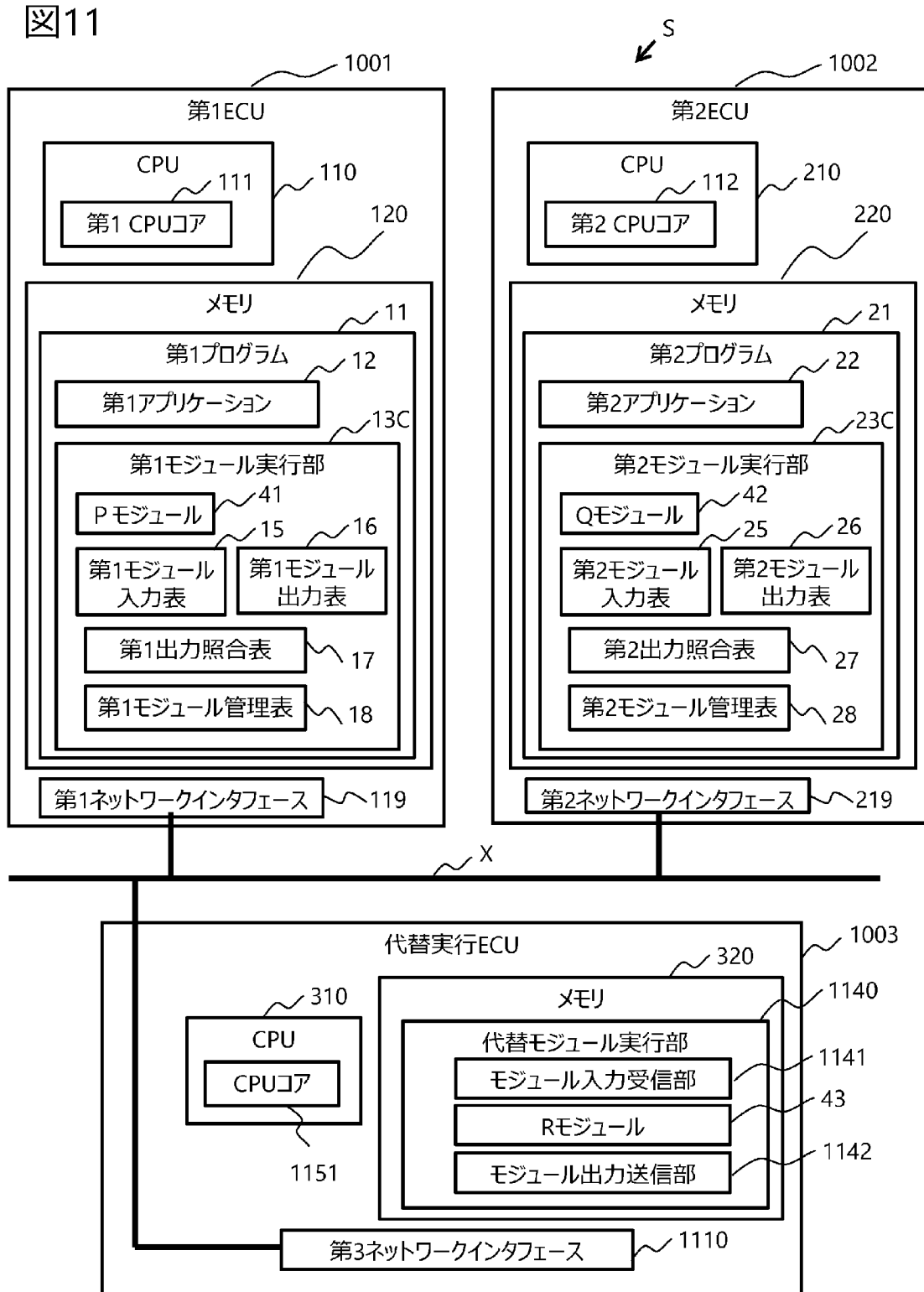


[図10]

図10

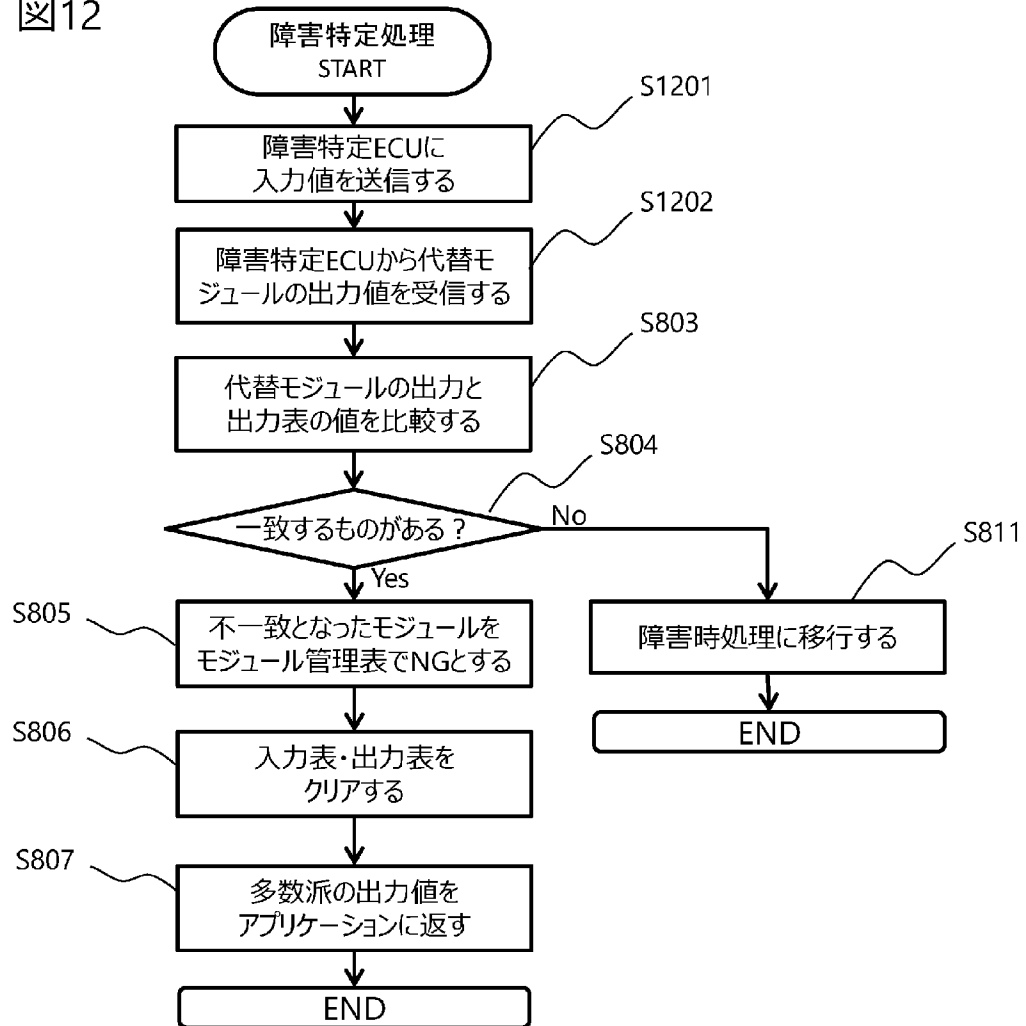


[図11]



[図12]

図12



# INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2018/022681

## A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl. G06F11/14 (2006.01) i, G06F11/07 (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)

Int.Cl. G06F11/14, G06F11/07

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

|                                                          |           |
|----------------------------------------------------------|-----------|
| Published examined utility model applications of Japan   | 1922-1996 |
| Published unexamined utility model applications of Japan | 1971-2018 |
| Registered utility model specifications of Japan         | 1996-2018 |
| Published registered utility model applications of Japan | 1994-2018 |

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. |
|-----------|------------------------------------------------------------------------------------------------------------------------|-----------------------|
| A         | JP 2004-355233 A (NEC CORP.) 16 December 2004, paragraphs [0108]-[0113] & US 2004/0255185 A1, paragraphs [0175]-[0180] | 1-8                   |

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>* Special categories of cited documents:</p> <p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier application or patent but published on or after the international filing date</p> <p>"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)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p> | <p>"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</p> <p>"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</p> <p>"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</p> <p>"&amp;" document member of the same patent family</p> |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Date of the actual completion of the international search  
31 July 2018 (31.07.2018)

Date of mailing of the international search report  
14 August 2018 (14.08.2018)

Name and mailing address of the ISA/  
Japan Patent Office  
3-4-3, Kasumigaseki, Chiyoda-ku,  
Tokyo 100-8915, Japan

Authorized officer  
  
Telephone No.

## A. 発明の属する分野の分類（国際特許分類（IPC））

Int.Cl. G06F11/14(2006.01)i, G06F11/07(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. G06F11/14, G06F11/07

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 | 1971-2018年 |
| 日本国実用新案登録公報 | 1996-2018年 |
| 日本国登録実用新案公報 | 1994-2018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<br>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<br>請求項の番号 |
|-----------------|---------------------------------------------------------------------------------------------------|----------------|
| A               | JP 2004-355233 A（日本電気株式会社）2004.12.16,<br>段落[0108]-[0113]<br>& US 2004/0255185 A1, 段落[0175]-[0180] | 1-8            |

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                                | 「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     |
| 「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                        | 「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                     |
| 「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） | 「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 |
| 「O」 口頭による開示、使用、展示等に言及する文献                                     | 「&」 同一パテントファミリー文献                                                   |
| 「P」 国際出願日前で、かつ優先権の主張の基礎となる出願                                  |                                                                     |

国際調査を完了した日

31.07.2018

国際調査報告の発送日

14.08.2018

国際調査機関の名称及びあて先

日本国特許庁（ISA/J P）

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）

大桃 由紀雄

5 B

5583

電話番号 03-3581-1101 内線 3545