

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6405966号
(P6405966)

(45) 発行日 平成30年10月17日(2018.10.17)

(24) 登録日 平成30年9月28日(2018.9.28)

(51) Int.Cl. F I
G06F 11/14 (2006.01) G O 6 F 11/14 6 3 8
G06F 11/16 (2006.01) G O 6 F 11/16 6 4 1

請求項の数 5 (全 11 頁)

(21) 出願番号	特願2014-249147 (P2014-249147)	(73) 特許権者	000004260 株式会社デンソー
(22) 出願日	平成26年12月9日(2014.12.9)		愛知県刈谷市昭和町1丁目1番地
(65) 公開番号	特開2016-110502 (P2016-110502A)	(74) 代理人	100140486 弁理士 鎌田 徹
(43) 公開日	平成28年6月20日(2016.6.20)	(74) 代理人	100170058 弁理士 津田 拓真
審査請求日	平成29年4月28日(2017.4.28)	(74) 代理人	100139066 弁理士 伊藤 健太郎
		(72) 発明者	田中 亮次 愛知県刈谷市昭和町1丁目1番地 株式会社デンソー内
		審査官	清木 泰

最終頁に続く

(54) 【発明の名称】 電子制御装置

(57) 【特許請求の範囲】

【請求項1】

複数の演算部(110, 120)において同一の演算が並行して実行される電子制御装置(10)であって、

それぞれの前記演算部における演算結果を常に比較し、当該演算結果が互いに同一かどうかを判定する比較判定部(130)と、

情報を一時的に保存するための第1保存部(151)及び第2保存部(152)と、を備えており、

それぞれの前記演算部は、

入力情報(PR)を用いて処理を行うように記述された関数を実行する際、

前記入力情報を前記第1保存部に保存する処理(S101)と、

前記関数の格納場所である関数アドレスを、前記第2保存部に保存する処理(S102)と、

を、前記関数を実行する直前において予め実行するように構成されており、

前記比較判定部により、それぞれの前記演算結果が互いに同一ではないと判定された場合には、

それぞれの前記演算部によって、

前記第1保存部に記憶されている前記入力情報を用いて、前記第2保存部に保存されている前記関数アドレスに対応する前記関数が再度実行され、

前記比較判定部の判定結果を記憶するためのレジスタを更に備えており、

10

20

複数の前記演算部のうちの 하나가、前記レジスタから前記判定結果を読み込むように構成されていることを特徴とする電子制御装置。

【請求項 2】

前記第 1 保存部に記憶されている前記入力情報を用いて、前記第 2 保存部に保存されている前記関数アドレスに対応する前記関数が再度実行された後、

前記比較判定部の判定結果に基づいて、前記演算部に生じた異常が恒久的なものであるか否かの判定が行われることを特徴とする、請求項 1 に記載の電子制御装置。

【請求項 3】

前記第 1 保存部に記憶されている前記入力情報を用いて、前記第 2 保存部に保存されている前記関数アドレスに対応する前記関数が再度実行された後、

前記比較判定部により、それぞれの前記演算結果が互いに同一であると判定された場合には、前記演算部に生じた異常が一時的なものであると判定することを特徴とする、請求項 2 に記載の電子制御装置。

【請求項 4】

前記演算部に生じた異常が恒久的なものであると判定された場合には、安全に動作を停止するためのフェールセーフ処理が実行され、前記演算部に生じた異常が一時的なものであると判定された場合には、前記フェールセーフ処理が実行されないことを特徴とする、請求項 3 に記載の電子制御装置。

【請求項 5】

前記第 1 保存部に記憶されている前記入力情報を用いて、前記第 2 保存部に保存されている前記関数アドレスに対応する前記関数が再度実行されるよりも前に、それぞれの前記演算部がリセットされることを特徴とする、請求項 1 乃至 4 のいずれか 1 項に記載の電子制御装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、複数の演算部において同一の演算が並行して実行される電子制御装置に関する。

【背景技術】

【0002】

自動車用の電子制御装置では、常に正確な制御を行うことが求められる。そこで、制御のための演算を司る CPU を複数備えた構成とした上で、それぞれの CPU において同一の演算を並列して実行させ、全ての演算結果が同一であることを確認しながら制御を行う方式（ロックステップ方式）の電子制御装置が実用化されている（例えば下記特許文献 1 を参照）。

【0003】

尚、ロックステップ方式の電子制御装置としては、上記のように複数設けられた CPU が同一の演算を行う態様他、単一の CPU に設けられた複数のコアが同一の演算を行う態様とされることもある。以下では、同一の演算を行うために複数設けられた CPU 又はコアのことを総じて「演算部」と表記する。

【0004】

下記特許文献 1 に記載の電子制御装置では、一部の演算部における演算結果が他と異なってしまうことが検出された場合、それぞれの演算部において自己診断ソフトを実行させることで、異常が生じた演算部を特定することとしている。一部の演算部における診断結果（自己診断ソフトを実行することにより得られる演算結果）が異常を示すものであった場合には、当該演算部において何らかの異常が生じているということである。そこで、当該演算部からの演算結果の使用を中断し、以降は正常な演算部からの演算結果のみを用いて制御が行われる。

【0005】

また、全ての演算部における診断結果が異常を示すものであった場合には、全ての演算

10

20

30

40

50

部において何らかの異常が生じているということである。そこで、動作が保障されたフェールセーフソフトをそれぞれの演算部において実行させて安全を確保する。

【先行技術文献】

【特許文献】

【0006】

【特許文献1】特開2012-68788号公報

【発明の概要】

【発明が解決しようとする課題】

【0007】

ところで、一部の演算部における演算結果が他と異なってしまったことが検出された後、自己診断ソフトが実行されたにも拘らず、全ての診断結果が正常となってしまう場合がある。このような現象が生じる原因の一つとしては、例えば、演算部において生じた異常はノイズの影響等による一時的なものであり、自己診断ソフトが実行される際には当該異常が解消していた、ということが考えられる。

10

【0008】

もう一つの原因としては、一部又は全ての演算部において恒久的な異常が生じているにも拘らず、当該異常が自己診断ソフトでは検出されなかったということも考えられる。

【0009】

上記特許文献1に記載の電子制御装置では、自己診断ソフトによる診断結果が全ての演算部において正常であった場合、演算結果の違いは演算部の一時的な異常により生じたのか、それとも、(自己診断ソフトでは検出不可な)演算部の恒久的な異常により生じたのか、を区別することができない。

20

【0010】

このため、上記特許文献1に記載の電子制御装置では、全ての診断結果が正常となった場合には、少なくとも一部の演算部に恒久的な異常が生じている可能性に鑑み、フェールセーフソフトを全ての演算部において実行させることとしている。実際には演算部の恒久的な異常は生じておらず、一時的な異常のみが生じたような場合にも、必ずフェールセーフソフトが実行されてしまうため、ソフトに異常がなくてもフェールセーフが働いてしまうという問題がある。

【0011】

30

本発明はこのような課題に鑑みてなされたものであり、その目的は、一部の演算部における演算結果が他と異なってしまったことが検出された際において、演算部の異常が一時的なものであるのか、それとも恒久的なものであるのかを正確に区別し、その後の対応を適切にとることのできる電子制御装置を提供することにある。

【課題を解決するための手段】

【0012】

上記課題を解決するために、本発明に係る電子制御装置では、複数の演算部において同一の演算が並行して実行される電子制御装置であって、それぞれの演算部における演算結果を常に比較し、当該演算結果が互いに同一かどうかを判定する比較判定部と、情報を一時的に保存するための第1保存部及び第2保存部と、を備えている。それぞれの演算部は、入力情報を用いて処理を行うように記述された関数を実行する際、入力情報を第1保存部に保存する処理と、関数の格納場所である関数アドレスを、第2保存部に保存する処理と、を、関数を実行する直前において予め実行するように構成されている。比較判定部により、それぞれの演算結果が互いに同一ではないと判定された場合には、それぞれの演算部によって、第1保存部に記憶されている入力情報を用いて、第2保存部に保存されている関数アドレスに対応する関数が再度実行され、比較判定部の判定結果を記憶するためのレジスタを更に備えており、複数の演算部のうちの 하나가、レジスタから判定結果を読み込むように構成されている。

40

【0013】

上記のような電子制御装置では、複数の演算部におけるそれぞれの演算結果が互いに同

50

一ではないと判定された場合、第1保存部に記憶されている入力情報を用いて、第2保存部に保存されている関数アドレスに対応する関数が再度実行される。すなわち、演算結果が互いに同一ではないと判定される直前に実行されたものと同じの演算が、全ての演算部において、前回と同じの条件で再度実行される。

【0014】

例えば、関数が再度実行された後の演算結果が全ての演算部において同一となった場合には、演算部に生じていたのは一時的な異常であったと推定することができる。また、関数が再度実行された後の演算結果が全ての演算部において同一とならなかった場合には、少なくとも一部の演算部において恒久的な異常が生じていると推定することができる。

【0015】

つまり、本発明では、予め用意された自己診断ソフトによってではなく、演算結果が互いに同一ではないと判定される直前に実行されたものと同じの演算によって、演算部に生じた異常が一時的なものであったかどうか判定される。演算部において恒久的な異常が生じていた場合には確実に演算結果の不一致が生じるため、上記の判定を正確に行うことができ、その後の対応を適切にとることができる。

【発明の効果】

【0016】

本発明によれば、一部の演算部における演算結果が他と異なってしまったことが検出された際において、演算部の異常が一時的なものであるのか、それとも恒久的なものであるのかを正確に区別し、その後の対応を適切にとることのできる電子制御装置を提供することができる。

【図面の簡単な説明】

【0017】

【図1】本発明の実施形態に係る電子制御装置の構成を模式的に示す図である。

【図2】図1に示された電子制御装置によって実行される処理の流れを示すフローチャートである。

【図3】図1に示された電子制御装置によって実行される処理の流れを示すフローチャートである。

【図4】入力情報及び関数について説明するための図である。

【発明を実施するための形態】

【0018】

以下、添付図面を参照しながら本発明の実施形態について説明する。説明の理解を容易にするため、各図面において同一の構成要素に対しては可能な限り同一の符号を付して、重複する説明は省略する。

【0019】

本実施形態に係る電子制御装置10は、不図示の車両に備えられ、当該車両の各種動作（例えば燃料噴射等）を制御するための装置である。尚、本発明の適用範囲はこのような車両用の電子制御装置に限定されるものではなく、機器の制御を行うための電子制御装置全般に適用することができる。特に、利用者の安全のために正確な制御が求められるような機器の電子制御装置に、本発明が適用されることが望ましい。

【0020】

図1に示されるように、本実施形態に係る電子制御装置10は、演算器110と、演算器120と、判定部130と、不揮発性メモリ140と、揮発性メモリ150と、を備えている。

【0021】

演算器110及び演算器120は、互いに同一な構成の中央演算装置（CPU）である。電子制御装置10が制御を行うにあたって実行される演算は、これら演算器110及び演算器120によって実行される。具体的には、演算器110及び演算器120によってそれぞれ異なる演算が行われるのではなく、同一の演算が並行して且つ同期して行われる。このため、演算器110及び演算器120の両方が正常に動作しているのであれば、そ

10

20

30

40

50

それぞれの演算結果は常に一致することとなる。逆に、少なくとも一方の演算器（110，120）に異常が生じた場合には、それぞれの演算結果は通常は一致しない。

【0022】

後に詳しく説明するように、本実施形態においては、演算部110がメインの演算部として演算を行いながら、その演算結果が演算器120の演算結果と一致することの確認を常に行うように構成されている。演算部110は、その演算結果が演算器120の演算結果と異なっている場合には、その原因が演算部110又は演算器120に生じた恒久的な異常なのか、それとも一時的な異常なのかの判定を引き続き行う。

【0023】

判定部130は、演算器110及び演算器120のそれぞれの演算結果を監視し、両者が互いに一致しているかどうかの判定を行う部分である。判定部130は、判定結果を記憶するためのレジスタ（以下、「比較結果レジスタ」と称する）を備えている。演算器110の演算結果（以下、「第1演算結果」とも称する）と演算器120の演算結果（以下、「第2演算結果」とも称する）とが互いに一致している場合、すなわち、演算器110及び演算器120がいずれも正常に動作している場合には、比較結果レジスタには0が書き込まれる。第1演算結果と第2演算結果とが互いに一致していない場合、すなわち、演算器110及び演算器120の少なくとも一方において異常が生じた場合には、比較結果レジスタには1が書き込まれる。

【0024】

不揮発性メモリ140は、所謂ROM（Read Only Memory）であって、演算器110及び演算器120によって実行されるプログラムの一部や、固定値として設定された制御パラメータ等が記憶された記憶装置である。不揮発性メモリ140に記憶された情報は、演算器110及び演算器120の両方から参照される。

【0025】

揮発性メモリ150は、所謂RAM（Random Access Memory）であって、演算器110及び演算器120によって実行されるプログラムの一部や、当該プログラムの処理が行われる際に用いられる情報（変数値等）を一時的に記憶するための記憶装置である。揮発性メモリ150に記憶された情報は、演算器110及び演算器120の両方から参照される。また、揮発性メモリ150に対する情報の書き込みも、演算器110及び演算器120の両方から行われる。

【0026】

揮発性メモリ150は、情報を一時的に保存するための部分として、第1保存部151と、第2保存部152とを有している。尚、揮発性メモリ150はその全体が情報を一時的に保存するためのものなのであるが、後の説明の便宜のために、その一部を特に第1保存部151及び第2保存部152と表記している。これらのそれぞれに保存される情報については、後に説明する。

【0027】

演算器110及び演算器120においてプログラムが実行される際の処理の流れについて説明する。当該プログラムは不揮発性メモリ140に記憶されており、複数の関数を含んだ形で記載されている。ここで、関数とは、例えば揮発性メモリ150に記憶された各種情報（入力情報）を用いて処理を行うように記述された命令群のことである。

【0028】

図4には、C言語によって記載された関数の簡単な例が示されている。関数「example」は、整数型の引数を2つ（int a, int b）渡されて実行されると、これら引数の積（int a * b）を返すものである。この例では、関数「example」が実行されると、引数である変数a及び変数bの値を用いた演算が行われる。変数a及び変数bの値のように、関数内の演算処理に用いられる情報（変数値）を総じて、以下では「入力情報PR」と表記する。尚、入力情報PRには、引数として関数に渡されるような情報のみならず、当該関数の外で定義されたグローバル変数の値が含まれる場合もある。

【0029】

10

20

30

40

50

図2に示されたフローチャートは、プログラムに含まれる関数が実行される際に、電子制御装置10において都度行われる処理の流れを示すものである。具体的には、プログラム中の関数が実行される度に図2に示される処理の一部(S101~S103)が行われるように、当該プログラムが記載されている。図2に示される処理のうち、ステップS101からステップS107までの処理は、演算器110は制御側として処理を行い、演算器120は監視側として処理を行う。図2に示される処理のうち、ステップS104以降の処理は、演算器110のみによって行われる。

【0030】

最初のステップS101では、これから実行される(後のステップS103で実行される)関数に用いられる入力情報PRの値が、全て第1保存部151に記憶(コピー)される。具体的には、例えば、入力情報PRに含まれる変数に対応したメンバ変数を有する構造体を予め用意しておき、構造体の各メンバ変数の値を入力情報PRの値に一致させた後、当該構造体を第1保存部151に記憶すればよい。尚、入力情報PRに含まれる変数の種類(型及びサイズ)や数は、関数によって異なるので、上記の構造体は関数ごとに用意しておけばよい。

10

【0031】

ステップS101に続くステップS102では、これから実行される関数の格納場所(不揮発性メモリ140におけるアドレス)が第2保存部152に保存される。関数の格納場所とは、所謂「関数ポインタ」としてプログラム中に記述される変数に記憶されている情報である。

20

【0032】

ステップS102に続くステップS103では、関数が実行される。ステップS103に続くステップS104では、比較結果レジスタの値が読み込まれる。既に述べたように、比較結果レジスタは、判定部130に設けられたレジスタであって、第1演算結果と第2演算結果とが一致した場合には0が書き込まれ、第1演算結果と第2演算結果とが一致しない場合には1が書き込まれるレジスタである。本実施形態では、演算器110及び演算器120によって一つの命令が実行される毎に、判定部130によって第1演算結果と第2演算結果との比較が行われ、当該比較の結果に基づいて比較結果レジスタへの値の書き込みが行われている。

【0033】

このため、ステップS104において読み込まれる比較結果レジスタの値は、ステップS103における処理(関数の実行)の結果が、演算器110と演算器120とで一致するかどうかを示す値、ということになる。

30

【0034】

ステップS104に続くステップS105では、ステップS104において読み込まれた比較結果レジスタの値が、電子制御装置10が備えるフラッシュメモリ(不図示)に記憶される。

【0035】

ステップS105に続くステップS106では、ステップS104において読み込まれた比較結果レジスタの値が0であるか否か、すなわち、第1演算結果と第2演算結果とが一致しているかが判定される。比較結果レジスタの値が0である場合には、図2に示される一連の処理を終了する。その後は、プログラムにおいて次の関数が実行される際に、再び図2に示される処理が実行される。つまり、ステップS103においては、前回の関数に続く次の関数が実行される。

40

【0036】

ステップS106において、比較結果レジスタの値が1である場合には、ステップS107に移行する。ステップS107では、演算器110及び演算器120の両方が同時にリセット(再起動)される。

【0037】

演算器110及び演算器120がリセットされた直後において行われる処理について、

50

図3を参照しながら説明する。図3に示される一連の処理は、演算器110は制御側として処理を行い、演算器120は監視側として処理を行う。

【0038】

最初のステップS201では、比較結果レジスタの値がフラッシュメモリから読み込まれる。ここで読み込まれる比較結果レジスタの値は、演算器110及び演算器120がリセットされる直前に、図2に示されるステップS105においてフラッシュメモリに記憶されていた値である。

【0039】

ステップS201に続くステップS202では、ステップS201において読み込まれた比較結果レジスタの値が0であるか否か、すなわち、リセット前における第1演算結果と第2演算結果とが一致していたかどうか判定される。比較結果レジスタの値が0である場合には、図3に示される一連の処理を終了する。その後は、図2を参照しながら説明した一連の処理が実行される。

10

【0040】

ステップS202において、比較結果レジスタの値が1である場合には、ステップS203に移行する。ステップS203では、第1保存部151に記憶されていた入力情報PRの値が読み込まれ、関数で用いられる全ての変数にその値が代入される。ここでいう関数とは、リセットが実行される直前において、演算器110及び演算器120で実行された関数のことである。換言すれば、第1演算結果と第2演算結果とが不一致となる直前において実行された関数のことである。

20

【0041】

ステップS203に続くステップS204では、第2保存部152に記憶されていた値(関数ポインタに記憶されていた値)が読み込まれる。

【0042】

ステップS204に続くステップS205では、ステップS204において読み込まれた関数ポインタに対応する関数(関数ポインタに記憶されていたアドレスに格納されている関数)が実行される。ステップS205では、第1演算結果と第2演算結果とが不一致となる直前において実行された関数が、前回(第1演算結果と第2演算結果とが不一致となる直前)と同一条件にて再度実行される。

【0043】

ステップS205に続くステップS206では、図2のステップS104と同様に、比較結果レジスタの値が読み込まれる。

30

【0044】

ステップS206に続くステップS207では、ステップS206において読み込まれた比較結果レジスタの値が0であるか否か、すなわち、第1演算結果と第2演算結果とが一致しているか否かが判定される。比較結果レジスタの値が0である場合には、ステップS208に移行する。

【0045】

ステップS208に移行したということは、前回においては第1演算結果と第2演算結果とが一致しなかった演算を、同一条件において再度実行したところ(ステップS205)、今回は第1演算結果と第2演算結果とが一致した、ということである。このため、前回(リセット前)において第1演算結果と第2演算結果とが一致しなかったのは、演算器110又は演算器120において恒久的な異常が生じていたことに起因するのではなく、ノイズ等の影響により一時的な異常が生じていたことに起因すると推測される。

40

【0046】

このため、ステップS208では、演算器110又は演算器120のいずれかに生じていたのは一時的な異常であって、現時点では演算器110及び演算器120はいずれも正常に機能していると判定される。その後は、図2を参照しながら説明した一連の処理が実行される(ステップS209)。

【0047】

50

ステップS207において、比較結果レジスタの値が1である場合には、ステップS210に移行する。

【0048】

ステップS210に移行したということは、前回においては第1演算結果と第2演算結果とが一致しなかった演算を、同一条件において再度実行したところ、今回もやはり第1演算結果と第2演算結果とが一致しなかった、ということである。このため、前回（リセット前）において第1演算結果と第2演算結果とが一致しなかったのは、演算器110又は演算器120において恒久的な異常が生じていたことに起因すると推測される。

【0049】

このため、ステップS210では、演算器110又は演算器120のいずれかに生じていたのは恒久的な異常であって、現時点においても演算器110又は演算器120（もしくは両方）が正常に機能し得ない状態であると判定される。

【0050】

ステップS210に続くステップS211では、異常に対応する処理が必要に応じて適宜実行される。例えば、本実施形態のように制御対象が車両である場合には、車両を安全に停止させるために必要となる一連の処理、すなわち、フェールセーフ処理が実行される。

【0051】

また、演算器110又は演算器120に生じた異常の具体的な原因を更に詳しく検査するような処理が行われてもよい。当該処理の結果、演算器110又は演算器120に生じた異常が軽微であると判明した場合には、一部の機能を制限しながら引き続き車両の走行を継続するような処理が行われてもよい。

【0052】

以上のように、本実施形態に係る電子制御装置10では、2つの演算器（110，120）におけるそれぞれの演算結果が互いに同一ではないと判定された場合、第1保存部151に記憶されている入力情報PRを用いて、第2保存部152に保存されている関数アドレスに対応する関数が再度実行される（ステップS205）。すなわち、演算結果が互いに同一ではないと判定される直前に実行されたものと同一の演算が、全ての演算器（110，120）において、前回と同一の条件で再度実行される。

【0053】

その後、2つの演算器（110，120）におけるそれぞれの演算結果が再度比較される（ステップS207）。第1演算結果と第2演算結果が互いに一致した場合には、演算器110及び演算器120はいずれも正常であると判定される（ステップS208）。第1演算結果と第2演算結果が再び不一致となった場合には、演算器110又は演算器120の少なくとも一方において恒久的な異常が生じていたと判定され（ステップS210）、必要な対応がとられる（ステップS211）。

【0054】

つまり、本実施形態では、予め用意された自己診断ソフトによってではなく、演算結果が互いに同一ではないと判定される直前に実行されたものと同一の演算によって、演算器（110，120）に生じた異常が一時的なものであったかが判定される。恒久的な異常が生じていた場合には確実に演算結果の不一致が生じるため、上記の判定を正確に行うことができ、その後の対応を適切にとることが可能となっている。

【0055】

また、本実施形態では、前回と同一の条件において同一の関数が再度実行される前に、演算器110及び演算器120の両方が同時にリセット（再起動）される。このため、一時的な異常等によって演算器110等が不安定な状態のまま、関数が再実行されてしまうことが防止される。リセット後の安定した状態において異常が判定されるため、演算器110等に生じた異常が恒久的なものであるかどうかの判定が、正確に行われる。

【0056】

以上、具体例を参照しつつ本発明の実施の形態について説明した。しかし、本発明はこ

10

20

30

40

50

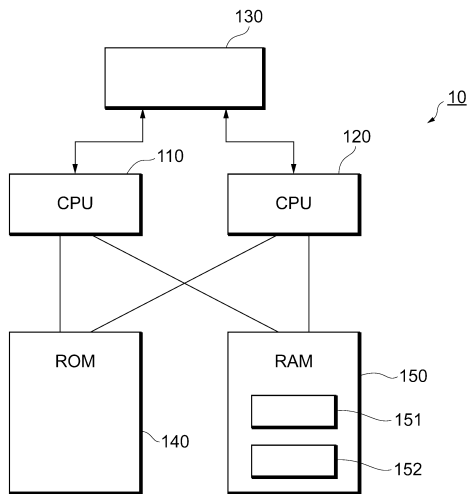
これらの具体例に限定されるものではない。すなわち、これら具体例に、当業者が適宜設計変更を加えたものも、本発明の特徴を備えている限り、本発明の範囲に包含される。例えば、前述した各具体例が備える各要素およびその配置、材料、条件、形状、サイズなどは、例示したものに限定されるわけではなく適宜変更することができる。また、前述した各実施の形態が備える各要素は、技術的に可能な限りにおいて組み合わせることができ、これらを組み合わせたものも本発明の特徴を含む限り本発明の範囲に包含される。

【符号の説明】

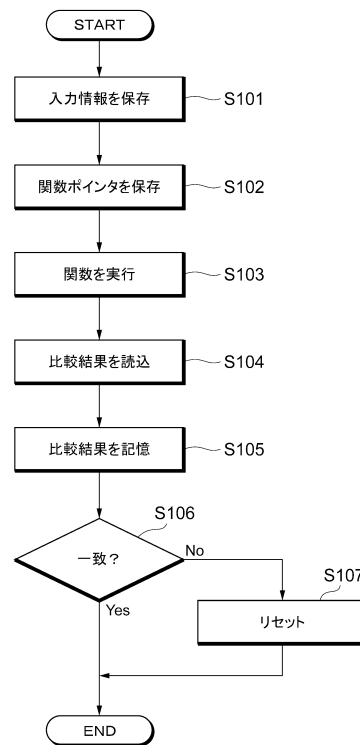
【 0 0 5 7 】

- 1 0 : 電子制御装置
- 1 1 0 : 演算器
- 1 2 0 : 演算器
- 1 3 0 : 統括制御部
- 1 4 0 : 不揮発性メモリ
- 1 5 0 : 揮発性メモリ
- 1 5 1 : 第 1 保存部
- 1 5 2 : 第 2 保存部

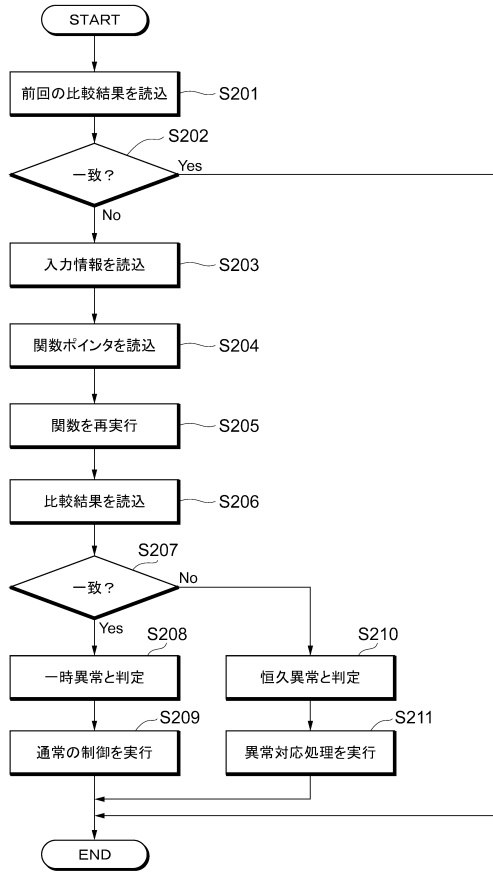
【 図 1 】



【 図 2 】



【図3】



【図4】

```
int example (int a, int b)
{
    PR
    return a * b ;
}
```

フロントページの続き

(56)参考文献 特開2011-095837(JP,A)
特開昭53-032646(JP,A)
特開2001-109635(JP,A)
特開2011-170606(JP,A)
特開2009-295126(JP,A)
特開2008-059402(JP,A)
特開2004-062309(JP,A)
特開平05-189258(JP,A)
特開平05-120047(JP,A)
特開平02-272645(JP,A)
特開昭62-130429(JP,A)
米国特許第06199171(US,B1)

(58)調査した分野(Int.Cl., DB名)

G06F11/00-11/07
G06F11/14-11/36
G06F 9/30- 9/355