(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2014-35740
(P2014-35740A)
(43)公開日　平成26年2月24日(2014.2.24)

(51)Int.Cl.		F I		テーマコード（参考）
G06F 11/30	(2006.01)	G06F 11/30	305D	5B042

審査請求 未請求　請求項の数 6　ＯＬ　（全 8 頁）

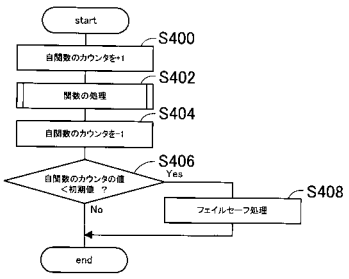
(21)出願番号　特願2012-178168 (P2012-178168)
(22)出願日　平成24年8月10日 (2012.8.10)

(71)出願人　000004260
株式会社デンソー
愛知県刈谷市昭和町１丁目１番地
(74)代理人　110000578
名古屋国際特許業務法人
(72)発明者　安田　政和
愛知県刈谷市昭和町１丁目１番地　株式会社デンソー内

Ｆターム(参考)　5B042 JJ15　JJ29　KK02

(54)【発明の名称】電子制御装置

(57)【要約】
【課題】プログラムカウンタの異常による不正アクセスを検出する電子制御装置を提供する。
【解決手段】電子制御装置は、関数が正常に呼び出されるときの開始処理でカウンタを＋１してから（Ｓ４００）、関数本体の処理を実行し（Ｓ４０２）、カウンタを－１する（Ｓ４０４）。関数が正常にＳ４００から呼び出されてＳ４０４の処理が実行されれば、Ｓ４０４の処理によりカウンタの値はＳ４００で＋１される前の初期値に戻る筈である。電子制御装置は、カウンタの値がＳ４００で＋１する前の初期値よりも小さい場合（Ｓ４０６：Ｙｅｓ）、プログラムカウンタが異常になり、関数が不正なアクセスによりＳ４００の開始処理からではなく途中から実行されたと判断する。電子制御装置は、関数が途中から実行されると関数の処理が適切に実行されないので、フェイルセーフ処理として、例えば自装置をリセットする（Ｓ４０８）。
【選択図】図２



【特許請求の範囲】
【請求項１】
プログラムカウンタ（２２）が示すアドレスの命令を読み出して複数の処理単位をそれぞれ実行する電子制御装置（１０）であって、
複数の前記処理単位のうち自身に対するアクセス状態を表わすアクセス情報が設けられている少なくとも一つの対象処理単位において、前記対象処理単位の実行が正常に開始されるときの開始処理で前記アクセス情報に第１演算を実行する第１演算手段（２０、Ｓ４００、Ｓ４１０）と、
前記第１演算手段により前記第１演算が実行される前の値に前記アクセス情報を戻す第２演算を前記対象処理単位の終了処理で実行する第２演算手段（２０、Ｓ４０４、Ｓ４１４）と、
前記第２演算手段が前記第２演算を実行した後の前記アクセス情報の値と、前記第１演算手段が前記第１演算を実行する前の前記アクセス情報の値とを前記終了処理時に比較して一致しない場合、前記プログラムカウンタの異常による前記対象処理単位への不正アクセスであると判定する判定手段（２０、Ｓ４０６、Ｓ４１６）と、
を備えることを特徴とする電子制御装置。
【請求項２】
前記第１演算手段（Ｓ４００）は、前記第１演算として前記アクセス情報に対して加算または減算の一方を実行し、前記第２演算手段（Ｓ４０４）は、前記第２演算として前記アクセス情報に対して加算または減算の他方を実行す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】
前記対象処理単位として不正なアクセスを許可しない前記処理単位が選択されていることを特徴とする請求項１または２に記載の電子制御装置。
【請求項４】
前記プログラムカウンタの異常であると前記判定手段が判定すると、フェイルセーフ処理を実行するフェイルセーフ手段（２０、Ｓ４０８、Ｓ４１８）を備えることを特徴とする請求項１から３のいずれか一項に記載の電子制御装置。
【請求項５】
前記フェイルセーフ手段は前記フェイルセーフ処理として自装置をリセットすることを特徴とする請求項４に記載の電子制御装置。
【請求項６】
前記電子制御装置は車両に搭載される装置であることを特徴とする請求項１から５のいずれか一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、プログラムカウンタの異常による不正アクセスを検出する電子制御装置に関する。
【背景技術】
【０００２】
マイクロコンピュータ（以下、「マイコン」とも言う。）は、ＣＰＵがメモリに記憶されているプログラムの命令を読み出して実行することにより処理を行う。命令の読み出しアドレスはプログラムカウンタに示されており、処理の流れに応じてプログラムカウンタは更新される。
【０００３】
しかし、宇宙線、電磁ノイズ等の影響によりプログラムカウンタが異常な値に更新されると、実行中のプログラムの処理流れからはアクセスされない他のプログラムの命令をＣＰＵが読み出して実行するおそれがある。
【０００４】

この問題を解決するため、例えば特許文献１には、命令と命令との境界アドレスとプログラムカウンタが示すアドレスとを比較することにより、プログラムカウンタのアドレスが命令の境界アドレスよりも小さい値、境界アドレスに一致した値、境界アドレスよりも大きい値と順番に変化するか否かを判定し、境界アドレスに一致せずに更新されるとプログラムカウンタの異常と判定する技術が開示されている。

【先行技術文献】

【特許文献】

【０００５】

【特許文献１】特開２００１－１８８６８８号公報

【発明の概要】

【発明が解決しようとする課題】

【０００６】

しかしながら、特許文献１の技術では、プログラムカウンタが命令の境界アドレスと一致してから異常値に更新されると、プログラムカウンタの異常のために実行中の処理単位から他の処理単位を不正にアクセスしても、不正アクセスであることを検出できないおそれがある。

【０００７】

本発明は上記課題を解決するためになされたものであり、プログラムカウンタの異常による不正アクセスを検出する電子制御装置を提供することを目的とする。

【課題を解決するための手段】

【０００８】

本発明の電子制御装置によると、複数の処理単位のうち自身に対するアクセス状態を示すアクセス情報が設けられている少なくとも一つの対象処理単位において、対象処理単位の実行が正常に開始されるときの開始処理でアクセス情報に対して第１演算を実行し、第１演算が実行される前のアクセス情報の値に戻す第２演算を対象処理単位の終了処理で実行する。

【０００９】

したがって、対象処理単位が開始処理からではなく不正なアクセスにより途中から実行されると、開始処理が実行されないためにアクセス情報に第１演算が実行されず、第２演算だけが実行されることになる。この場合、対象処理単位が正常に開始されるときの開始処理でアクセス情報に対して第１演算が実行される前の初期値にアクセス情報は戻らない。

【００１０】

そこで、第２演算を実行した後のアクセス情報の値と、第１演算を実行する前のアクセス情報の初期値とを対象処理単位の終了処理時に比較することにより、一致していない場合に、プログラムカウンタの異常により該当する対象処理単位が不正にアクセスされたと判定できる。

【００１１】

また、プログラムカウンタが実行中の処理単位の命令の境界アドレスと一致してから異常値に更新されて対象処理単位を不正にアクセスしても、プログラムカウンタの異常による不正アクセスであるとことを検出できる。

【００１２】

尚、本発明に備わる複数の手段の各機能は、構成自体で機能が特定されるハードウェア資源、プログラムにより機能が特定されるハードウェア資源、またはそれらの組合せにより実現される。また、これら複数の手段の各機能は、各々が物理的に互いに独立したハードウェア資源で実現されるものに限定されない。

【図面の簡単な説明】

【００１３】

【図１】本実施形態による電子制御装置を示すブロック図。

【図２】カウンタを用いたアクセスチェック処理を示すフローチャート。

【図３】関数毎のカウンタの変化を示す説明図。
【図４】不正アクセスにおけるカウンタの変化を示す説明図。
【図５】特定の関数に設けたカウンタの変化を示す説明図。
【図６】関数毎に設けたフラグを示す説明図。
【図７】フラグを用いたアクセスチェック処理を示すフローチャート。
【発明を実施するための形態】
【００１４】
以下、本発明の実施形態を図に基づいて説明する。
（第１実施形態）
図１に示す電子制御装置（Electronic Control Unit：ＥＣＵ）１０は、例えば車両に搭載されており、ＣＰＵ２０、ＲＯＭ３０、ＲＡＭ４０、図示しない入出力インタフェース等からなるマイクロコンピュータにより主に構成されている。ＣＰＵ２０は、プログラムカウンタ２２が示すアドレスの命令を読み出してＲＯＭ３０等に記憶されたプログラムを実行することにより、各種処理を実行する。
【００１５】
ＲＯＭ３０の記憶領域は、複数の重要プログラムを記憶している重要プログラム領域３２と、複数の通常プログラムを記憶している通常プログラム領域３４とに分割されている。ＲＡＭ４０の記憶領域は、重要データを記憶する重要データ領域４２と、通常データを記憶する通常データ領域４４とに分割されている。
【００１６】
ＣＰＵ２０は、プログラムが他のプログラムにアクセスするとき、両方のプログラムのアドレスをチェックすることにより、重要プログラム領域３２から重要プログラム領域３２または通常プログラム領域３４へのアクセス、ならびに通常プログラム領域３４から通常プログラム領域３４へのアクセスであれば許可し、通常プログラム領域３４から重要プログラム領域３２へのアクセスであれば禁止する。
【００１７】
また、プログラムがデータにアクセスするとき、プログラムアドレスとデータアドレスとをチェックすることにより、重要プログラム領域３２から重要データ領域４２または通常データ領域４４へのアクセス、ならびに通常プログラム領域３４から通常データ領域４４へのアクセスであれば許可し、通常プログラム領域３４から重要データ領域４２へのアクセスであれば禁止する。
【００１８】
ここで、プログラムが他のプログラムにアクセスするとき、重要プログラム領域３２から重要プログラム領域３２または通常プログラム領域３４へのアクセス、ならびに通常プログラム領域３４から通常プログラム領域３４へのアクセスは、前述したように領域レベルでは許可されている。
【００１９】
しかし、宇宙線、電磁ノイズ等によりプログラムカウンタの値が異常値に更新されると、プログラムが正常な開始処理からではなく不正なアクセスにより途中から実行されるおそれがある。領域レベルではアクセスが許可されているプログラムであっても、プログラムを正常な開始処理からではなく途中から実行すると、適正な処理を実行できない。
【００２０】
そこで、第１実施形態では、プログラムを構成する関数を処理単位とし、関数に対するアクセス状態を示すアクセス情報としてカウンタをＲＡＭ４０に設けている。重要プログラムを構成する関数であればカウンタを重要データ領域４２または通常データ領域４４のどちらに設けてもよく、通常プログラムを構成する関数であれば通常データ領域４４にアクセス情報を設ける。尚、関数に代えてタスクを処理単位としてよい。
【００２１】
（アクセスチェック処理）
アクセス情報としてカウンタを用いたアクセスチェック処理を図２に示す。図２におい

て、「Ｓ」はステップを表わしている。

【００２２】

ＣＰＵ２０は、関数が正常に呼び出されるときの開始処理で第１演算としてカウンタを＋１してから（Ｓ４００）、関数本体の処理を実行する（Ｓ４０２）。図３に示すように、関数Ａ～Ｃに対応してカウンタＡ～Ｃがそれぞれ設けられており、関数が正常に呼び出されると関数の開始処理でカウンタが＋１される。各カウンタの初期値は０である。関数本体の処理を終了すると、ＣＰＵ２０は第２演算としてカウンタを－１する（Ｓ４０４）。

【００２３】

関数が正常に開始処理から呼び出され、Ｓ４００の処理が実行されてからＳ４０４の処理が実行されれば、カウンタの値はＳ４００で＋１される前の初期値である０に戻る筈である。

【００２４】

これに対し、図４の関数Ｂのように関数Ｃから不正にアクセスされて途中から実行され、Ｓ４００の開始処理が実行されずにＳ４０４でカウンタを－１すると、カウンタの値は、Ｓ４００で＋１される前の初期値である０よりも小さい－１になる。

【００２５】

そこで、ＣＰＵ２０は、カウンタの値がＳ４００で＋１する前の初期値よりも小さいか否かを判定する（Ｓ４０６）。カウンタの値が初期値以上の場合（Ｓ４０６：Ｙｅｓ）、プログラムカウンタは正常であり関数は開始処理から正常に呼び出されたと判断し、ＣＰＵ２０は本処理を終了する。

【００２６】

領域レベルではアクセスが許可されている関数であっても、カウンタの値が初期値よりも小さい場合（Ｓ４０６：Ｙｅｓ）、ＣＰＵ２０は、プログラムカウンタが異常になり関数が不正なアクセスにより途中から実行されたと判断する。関数が途中から実行されると関数の処理が適切に実行されないので、ＣＰＵ２０はフェイルセーフ処理として、例えば自ＥＣＵをリセットする（Ｓ４０８）。

【００２７】

車両に搭載されるＥＣＵの場合には、ガソリンエンジンの場合にはスロットル開度を制限して吸気量を低減したり、ディーゼルエンジンの場合にはインジェクタからの噴射量を低減したり、あるいは変速段を固定にしたりして退避走行を実現するフェイルセーフ処理を実行してもよい。

【００２８】

また、複数の関数のすべてを、アクセス情報であるカウンタを設けてアクセス状態をチェックする対象となる対象処理単位としてもよいし、途中から不正アクセスされると重大な障害を引き起こすので不正アクセスを許可しない重要な関数だけを対象処理単位としてもよい。図５では、関数Ａ～Ｃのうち関数Ｂだけにカウンタを設けている。

【００２９】

アクセス情報を設ける重要な関数として、車両に搭載されるＥＣＵの場合には、ガソリンエンジンであれば吸気量制御、ディーゼルエンジンであれば噴射量制御、さらに過給制御、ブレーキ油圧制御、パワーステアリング制御等を実行する関数が考えられる。

【００３０】

このように、すべての関数ではなく重要な関数だけにアクセス情報を設けることにより、アクセス情報を設けない関数について、ＲＯＭ、ＲＡＭの記憶容量および処理負荷を低減できる。

【００３１】

第１実施形態では、アクセス情報として、専用のレジスタではなくＲＡＭ４０に設けたカウンタに対してＣＰＵ２０が増減処理を行ってプログラムカウンタ２２の異常を判定するので、プログラムカウンタ２２の異常を判定するために使用するＥＣＵ１０の資源を極力低減できる。

【００３２】
（第２実施形態）
本発明の第２実施形態を図６および図７に示す。第２実施形態では、第１実施形態のカウンタに代えて、２値の状態を表わすフラグをアクセス情報として設けている。図６は、関数１～ｎに対応して１ビットのフラグをそれぞれ設けた例を示している。関数１～ｎのアクセス状態を表わすフラグはＲＡＭ４０に設けられている。
【００３３】
アクセス情報としてフラグを設けた場合、図７のＳ４１０の開始処理においてＣＰＵ２０は、第１演算として、実行中の自関数のフラグを初期値から反転する。つまり、フラグの初期値が０であれば１に反転し、１であれば０に反転する。
【００３４】
関数本体の処理の実行後（Ｓ４１２）、終了処理においてＣＰＵ２０は、第２演算としてフラグの値を反転し（Ｓ４１４）、フラグの値がＳ４１０で反転する前の初期値であるか否かを判定する（Ｓ４１６）。
【００３５】
フラグの値が初期値の場合（Ｓ４１６：Ｙｅｓ）、プログラムカウンタは正常であり、関数は開始処理から正常に呼び出されたと判断し、ＣＰＵ２０は本処理を終了する。
フラグの値が初期値ではない場合（Ｓ４１６：Ｎｏ）、ＣＰＵ２０は、プログラムカウンタが異常になり関数が不正なアクセスにより途中から実行されたと判断し、フェイルセーフ処理を実行する（Ｓ４１８）。
【００３６】
第２実施形態の場合にも第１実施形態と同様に、途中から不正アクセスされると重大な障害を引き起こすので不正アクセスを許可しない重要な関数だけを対象処理単位とし、アクセス情報としてフラグを設けてもよい。
【００３７】
以上説明した上記実施形態では、対象処理単位である関数にアクセス情報としてカウンタまたはフラグを設け、関数が正常に呼び出された場合に関数の開始処理および終了処理でカウンタまたはフラグに予め設定された所定の演算を実行し、終了処理で演算を実行した後のカウンタまたはフラグの値に基づいて関数のアクセス状態を判定した。
【００３８】
これにより、上記実施形態で示した重要プログラム領域３２から重要プログラム領域３２または通常プログラム領域３４のように、記憶されている領域レベルでは関数へのアクセスが許可されている場合であっても、プログラムカウンタの異常により正常な開始処理からではなく途中から関数が実行されたことをアクセス情報に基づいて検出することにより、プログラムカウンタの異常による不正アクセスであることを検出できる。
【００３９】
また、プログラムカウンタが実行中の関数の命令の境界アドレスと一致してから異常値に更新されて他の関数を途中から実行しても、プログラムカウンタの異常による不正アクセスであることを検出できる。
【００４０】
［他の実施形態］
上記実施形態で示した関数の開始処理で実行する第１演算と、関数の終了処理で実行する第２演算との組合せは、第１演算を実行する前のアクセス情報の値に第２演算が戻すのであれば、どのような組合せでもよい。例えば、アクセス情報が数値であれば乗算と除算との組合せでもよい。
【００４１】
本発明の電子制御装置は、車両に搭載される装置に限らず、処理単位のアクセス状態に基づいてプログラムカウンタの異常による不正アクセスを検出することを目的とするのであれば、どのような用途に使用される電子制御装置に適用してもよい。
【００４２】

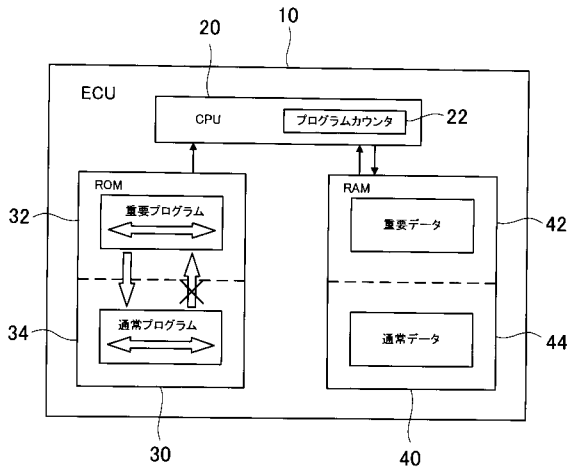
このように、本発明は、上記実施形態に限定されるものではなく、その要旨を逸脱しない範囲で種々の実施形態に適用可能である。

【符号の説明】

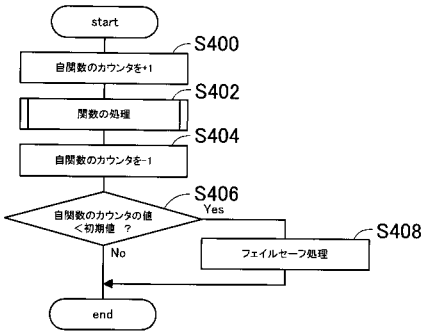
【００４３】

１０：ＥＣＵ（電子制御装置、第１演算手段、第２演算手段、判定手段、フェイルセーフ手段）、２２：プログラムカウンタ

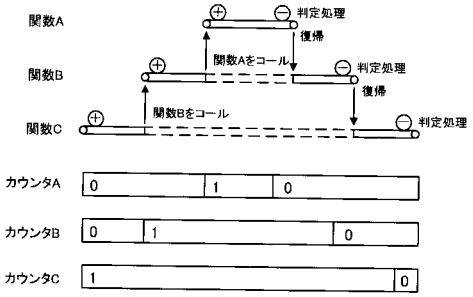
【図１】



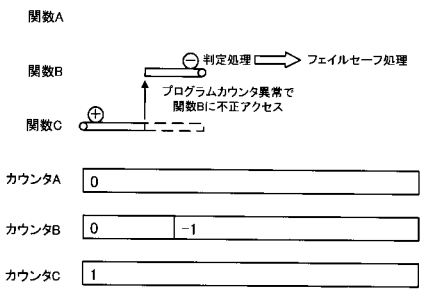
【図２】



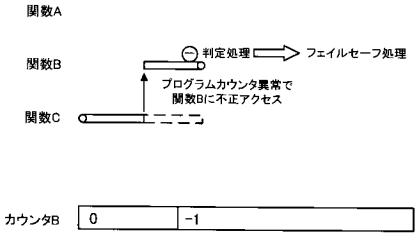
【図３】



【図４】



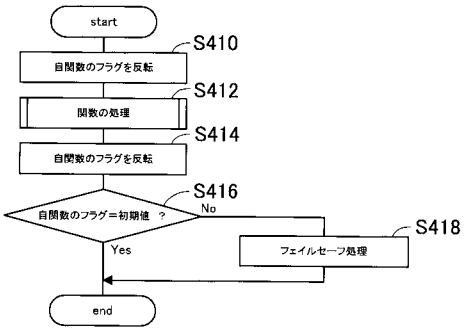
【図５】


関数A
関数B
判定処理
フェイルセーフ処理
プログラムカウンタ異常で
関数Bに不正アクセス
関数C
カウンタB
0
-1

【図６】


フラグ1
フラグ2
フラグn

【図７】


start
S410
自関数のフラグを反転
S412
関数の処理
S414
自関数のフラグを反転
S416
自関数のフラグ＝初期値 ?
No
Yes
S418
フェイルセーフ処理
end