(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第4732921号
(P4732921)

(45)発行日　平成23年7月27日(2011.7.27)　(24)登録日　平成23年4月28日(2011.4.28)

(51)Int.Cl.　　F I
G06F 21/22 (2006.01)　　G06F 9/06 660J
G06F 21/24 (2006.01)　　G06F 12/14 560C

請求項の数 5　(全 12 頁)

(21)出願番号　特願2006-48031(P2006-48031)
(22)出願日　平成18年2月24日(2006.2.24)
(65)公開番号　特開2007-226610(P2007-226610A)
(43)公開日　平成19年9月6日(2007.9.6)
審査請求日　平成20年4月17日(2008.4.17)

(73)特許権者　000101732
アルパイン株式会社
東京都品川区西五反田１丁目１番８号
(74)代理人　100081282
弁理士　中尾　俊輔
(74)代理人　100085084
弁理士　伊藤　高英
(74)代理人　100095326
弁理士　畑中　芳実
(74)代理人　100115314
弁理士　大倉　奈緒子
(74)代理人　100117190
弁理士　玉利　房枝
(74)代理人　100120385
弁理士　鈴木　健之

最終頁に続く

(54)【発明の名称】プログラム正当性検証装置

(57)【特許請求の範囲】
【請求項１】
メモリに保持されたプログラムの正当性を検証するプログラム正当性検証装置であって、
前記メモリに保持されたプログラムに対して所定の演算を行うグラフィックアクセラレータを備え、前記メモリに保持されたプログラムが正当な場合における前記所定の演算によって得られる結果を予め用意しておき、前記グラフィックアクセラレータによる前記メモリに保持されたプログラムに対する前記所定の演算の結果が、前記予め用意された結果と一致する場合に、前記メモリに保持されたプログラムが正当であるとみなすように形成され、
前記所定の演算に用いられる設定された初期値を有するデータが格納されたワークメモリを用意するように形成され、
前記グラフィックアクセラレータが、前記メモリに保持されたプログラムにおける１ブロック分のプログラムと、前記ワークメモリに格納された前記データの全部または一部を読み込み、読み込まれた前記１ブロック分のプログラムと前記データの全部または一部を用いて論理演算を行い、この論理演算の結果を前記ワークメモリに格納して前記ワークメモリに格納された前記データを更新するといった一連の処理を、前記メモリに保持されたプログラムにおけるすべてのブロックに至るまで順次繰り返すことによって、前記所定の演算を行うように形成され、
前記一連の処理の繰り返しによって前記ワークメモリに最終的に格納された前記データ

が、前記所定の演算の結果とされていること
を特徴とするプログラム正当性検証装置。
【請求項２】
前記所定の演算が、排他的論理和を含むことを特徴とする請求項１に記載のプログラム正当性検証装置。
【請求項３】
前記所定の演算が、ラスタ演算とされていることを特徴とする請求項１または請求項２に記載のプログラム正当性検証装置。
【請求項４】
前記所定の演算の結果と、前記予め用意された結果とが、それぞれ、複数個の画素データによって形成され、
前記グラフィックアクセラレータが、前記所定の演算の結果と前記予め用意された結果との排他的論理和を、双方の結果における互いに対応する画素データの組ごとに計算するように形成され、
この排他的論理和の計算結果が、すべての画素データの組において０に相当する結果となる場合に、前記所定の演算の結果が前記予め用意された結果と一致すると判定して、前記メモリに保持されたプログラムが正当であるとみなすように形成されていることを特徴とする請求項１乃至請求項３のいずれか１項に記載のプログラム正当性検証装置。
【請求項５】
前記所定の演算の結果と、前記予め用意された結果とが、それぞれ、複数個の画素データによって形成され、
前記予め用意された結果におけるすべての画素データが、０に相当するデータとされ、
前記ワークメモリに格納された前記データの前記初期値が、前記予め用意された結果に対応する値とされ、
前記メモリに保持されたプログラムに対する前記所定の演算の結果におけるすべての画素データが０に相当するデータとなる場合に、前記所定の演算の結果が前記予め用意された結果と一致すると判定して、前記メモリに保持されたプログラムが正当であるとみなすように形成されていることを特徴とする請求項１乃至請求項３のいずれか１項に記載のプログラム正当性検証装置。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、プログラム正当性検証装置に係り、特に、メモリに保持されたプログラムの正当性を検証するのに好適なプログラム正当性検証装置に関する。
【背景技術】
【０００２】
従来から、ＯＳ等のプログラム本体を起動する前には、ＣＰＵが、ブートローダ等のプログラムを実行することによって、FlashROMやＲＡＭ等のメモリに展開されて保持された状態のプログラム本体（すなわちプログラム本体イメージ）が破損しているか否かを検証するプログラム本体イメージの正当性の検証を行っていた。
【０００３】
このプログラム本体イメージの正当性の検証には、いわゆるチェックサムが用いられ、ＣＰＵは、プログラム本体イメージのチェックサムを計算し、この計算結果を予め記憶された正しいチェックサムと照合するようになっていた。
【０００４】
この照合の結果、ＣＰＵは、チェックサムの計算結果と、予め記憶された正しいチェックサムとが一致する場合には、プログラム本体イメージが破損のない正当なものであるとみなし、一致しない場合には、プログラム本体イメージが正当でない（破損している）とみなすようになっていた。
【０００５】

そして、ＣＰＵは、プログラム本体イメージが正当である場合には、プログラム本体を実行し、プログラム本体イメージが正当でない（破損している）場合には、プログラム本体を、２次記憶メモリ等から復元するようになっていた。

【０００６】

【特許文献１】特開平５－３２４３７０号公報

【発明の開示】

【発明が解決しようとする課題】

【０００７】

しかしながら、近年は、プログラムの複雑化にともなうプログラム本体イメージの巨大化により、チェックサムの計算によるプログラム本体イメージの正当性の検証に長い時間がかかり、これによって、プログラム本体の実行が遅延されるといった問題が生じていた。

【０００８】

例えば、従来は、プログラム本体イメージは破損していないにもかかわらず、チェックサムの計算に要する時間によって、チェックサムの計算の開始からプログラム本体の実行までに数秒程度の時間がかかってしまうことがあった。

【０００９】

そこで、本発明は、このような問題点に鑑みなされたものであり、プログラムの正当性の検証を迅速に行うことによって正当なプログラムをすみやかに実行することができ、ひいては、システムの起動時間を短縮することができるプログラム正当性検証装置を提供することを目的とするものである。

【課題を解決するための手段】

【００１０】

前述した目的を達成するため、本発明に係るプログラム正当性検証装置は、メモリに保持されたプログラムの正当性を検証するプログラム正当性検証装置であって、前記メモリに保持されたプログラムに対して所定の演算を行うグラフィックアクセラレータを備え、前記メモリに保持されたプログラムが正当な場合における前記所定の演算によって得られる結果を予め用意しておき、前記グラフィックアクセラレータによる前記メモリに保持されたプログラムに対する前記所定の演算の結果が、前記予め用意された結果と一致する場合に、前記メモリに保持されたプログラムが正当であるとみなすように形成されていることを特徴としている。そして、このような構成によれば、グラフィックアクセラレータにより、メモリに保持されたプログラムに対して、このプログラムを画像として扱った所定の演算を行い、この所定の演算の結果が予め用意された結果（正当なプログラムに対応する結果）と一致するか否かに基づいて、メモリに保持されたプログラムの正当性を迅速に検証することが可能となる。

【００１１】

また、本発明に係るプログラム正当性検証装置は、前記所定の演算が、排他的論理和（ＸＯＲ）を含むことを特徴としている。そして、このような構成によれば、演算が簡便かつプログラムの破損の検出に好適な演算結果を得ることができる排他的論理和を用いることによって、メモリに保持されたプログラムの正当性をより迅速かつ適切に検証することが可能となる。

【００１２】

さらに、本発明に係るプログラム正当性検証装置は、前記所定の演算が、ラスタ演算（raster operation）とされていることを特徴としている。そして、このような構成によれば、メモリに保持されたプログラムの正当性を、ラスタ演算を用いてさらに高速に検証することが可能となる。

【００１３】

さらにまた、本発明に係るプログラム正当性検証装置は、前記所定の演算に用いられる設定された初期値を有するデータが格納されたワークメモリを用意するように形成され、前記グラフィックアクセラレータが、前記メモリに保持されたプログラムにおける１ブロ

ック分のプログラムと、前記ワークメモリに格納された前記データの全部または一部を読み込み、読み込まれた前記１ブロック分のプログラムと前記データの全部または一部を用いて論理演算を行い、この論理演算の結果を前記ワークメモリに格納して前記ワークメモリに格納された前記データを更新するといった一連の処理を、前記メモリに保持されたプログラムにおけるすべてのブロックに至るまで順次繰り返すことによって、前記所定の演算を行うように形成され、前記一連の処理の繰り返しによって前記ワークメモリに最終的に格納された前記データが、前記所定の演算の結果とされていることを特徴としている。そして、このような構成によれば、一般的なグラフィックアクセラレータが備えているソースデータとデスティネーションデータとの論理演算を行う機能を活用して、メモリに保持されたプログラムをソースデータとし、ワークメモリに格納されたデータをデスティネーションデータとした論理演算を含む一連の処理（換言すれば、ワークメモリへの描画）を繰り返すことによって所定の演算を行うことが可能となる。この所定の演算の結果は、ワークメモリに最終的に格納されたデータ（換言すれば、ワークメモリへの最終描画結果）として取得することができる。そして、このようにして取得された所定の演算の結果と、予め用意された結果との一致の有無に基づいて、メモリに保持されたプログラムの正当性をより正確に検証することが可能となる。

【００１４】

また、本発明に係るプログラム正当性検証装置は、前記所定の演算の結果と、前記予め用意された結果とが、それぞれ、複数個の画素データによって形成され、前記グラフィックアクセラレータが、前記所定の演算の結果と前記予め用意された結果との排他的論理和を、双方の結果における互いに対応する画素データの組ごとに計算するように形成され、この排他的論理和の計算結果が、すべての画素データの組において０に相当する結果となる場合に、前記所定の演算の結果が前記予め用意された結果と一致すると判定して、前記メモリに保持されたプログラムが正当であるとみなすように形成されていることを特徴としている。そして、このような構成によれば、グラフィックアクセラレータにより、所定の演算の結果と予め用意された結果との排他的論理和を計算することで、メモリに保持されたプログラムの正当性を容易に検証することが可能となる。

【００１５】

さらに、本発明に係るプログラム正当性検証装置は、前記所定の演算の結果と、前記予め用意された結果とが、それぞれ、複数個の画素データによって形成され、前記予め用意された結果におけるすべての画素データが、０に相当するデータとされ、前記ワークメモリに格納された前記データの前記初期値が、前記予め用意された結果に対応する値とされ、前記メモリに保持されたプログラムに対する前記所定の演算の結果におけるすべての画素データが０に相当するデータとなる場合に、前記所定の演算の結果が前記予め用意された結果と一致すると判定して、前記メモリに保持されたプログラムが正当であるとみなすように形成されていることを特徴としている。そして、このような構成によれば、グラフィックアクセラレータによる所定の演算の結果におけるすべての画素データが０に相当するデータとなるか否かに基づいて、メモリに保持されたプログラムの正当性をさらに迅速に検証することが可能となる。

【発明の効果】

【００１６】

本発明に係るプログラム正当性検証装置によれば、グラフィックアクセラレータの演算機能を用いてメモリに保持されたプログラムの正当性の検証を迅速に行うことによって、正当なプログラムをすみやかに実行することができ、ひいては、システムの起動時間を短縮することができる。

【００１７】

さらに、本発明によれば、グラフィックアクセラレータの演算機能を用いてメモリに保持されたプログラムの正当性の検証を行うことによって、ＣＰＵの負担を軽減してＣＰＵに空き時間を確保することができるので、メモリに保持されたプログラムの正当性の検証が行われている最中においても、既に正当であるとみなされたプログラムをＣＰＵによっ

てすみやかに実行することができる。

【発明を実施するための最良の形態】

【００１８】

以下、本発明に係るプログラム正当性検証装置の実施形態について、図１乃至図３を参照して説明する。

【００１９】

図１は、本実施形態におけるプログラム正当性検証装置１のハードウェア構成を示したものであり、この図１に示すように、本実施形態におけるプログラム正当性検証装置１は、FlashROMやＲＡＭ等のメモリ２と、ＣＰＵ３と、グラフィックアクセラレータ４とを有している。

【００２０】

メモリ２には、ハードディスクドライブ等の図示しない記憶装置から読み込まれたプログラムとしてのＯＳ等のプログラム本体が、プログラム本体イメージとして保持されている。なお、プログラム本体を記憶装置から読み込んでメモリ２に保持する処理は、ブートローダ等のプログラム本体以外のプログラムによって行われるようになっている。

【００２１】

グラフィックアクセラレータ４は、メモリ２に保持されたプログラム本体イメージに対して、このプログラム本体イメージを画像として扱った所定の演算を行い、演算結果（以下、所定演算結果と称する）をメモリ２に格納するようになっている。

【００２２】

また、メモリ２には、予め用意された結果として、メモリ２に保持されたプログラム本体イメージが正当な場合（破損がない場合）における前記所定の演算によって得られる結果（以下、正当結果と称する）が格納されている。

【００２３】

ＣＰＵ３は、所定演算結果が正当結果と一致する場合に、メモリ２に保持されたプログラム本体イメージが正当であるとみなすようになっている。

【００２４】

具体的には、図２に示すように、グラフィックアクセラレータ４は、前記所定の演算の際に、プログラム本体イメージ５を、複数個のブロックに分割された１ブロック分のプログラム本体イメージ５ａ～５ｅの集合として認識するようになっている。なお、図２におけるプログラム本体イメージ５は、第１番目の１ブロック分のプログラム本体イメージ５ａから第５番目の１ブロック分のプログラム本体イメージ５ｅまでの５個の１ブロック分のプログラム本体イメージ５ａ～５ｅによって構成されている。また、図２に示すように、プログラム本体イメージ５は、２進数の場合には、０または１の値によって表現することができる。

【００２５】

グラフィックアクセラレータ４は、このようなプログラム本体イメージ５を前記所定の演算の際に画像として扱い、１ブロック分のプログラム本体イメージ５ａ～５ｅを、複数個の画素データの集合として捉えるようになっている。

【００２６】

例えば、１ブロック分のプログラム本体イメージ５ａ～５ｅを、６４行×６４列（５１２Ｂｙｔｅ）のデータと仮定し、グラフィックアクセラレータ４が、１Ｂｉｔで１画素を表すと仮定した場合には、１ブロック分のプログラム本体イメージ５ａ～５ｅは、４０９６個分の画素データからなる白黒（２値）画像として捉えることができる。

【００２７】

この場合、１ブロック分のプログラム本体イメージ５ａ～５ｅにおける「１」の値をとるデータは、白黒画像における白色の画素データとして捉えられ、１ブロック分のプログラム本体イメージ５ａ～５ｅにおける「０」の値をとるデータは、黒色の画素データとして捉えられる。

【００２８】

また、図２に示すように、メモリ２内には、ＣＰＵ３によってワークメモリ７が用意されており、このワークメモリ７には、前記所定の演算に用いられる所定の初期値が設定されたデータが格納されている。

【００２９】

このワークメモリ７に格納されたデータ（以下、ワークメモリ内データ８と称する）は、複数個の画素データによって形成されている。

【００３０】

本実施形態において、ワークメモリ内データ８の初期値は、ワークメモリ内データ８におけるすべての画素データが、１画素を１Ｂｉｔで表現する場合における「０」に相当する黒色のデータとなるような値（以下、初期値０と称する）とされている。

【００３１】

したがって、図２に示すように、初期値０のワークメモリ内データ８は、全面黒色の画像として表現される。

【００３２】

また、本実施形態において、ワークメモリ内データ８は、１ブロック分のプログラム本体イメージ５ａ～５ｅとデータのサイズが同一とされており、ワークメモリ内データ８の画素データの個数は、１ブロック分のプログラム本体イメージにおける画素データの個数と同行同列の同数とされている。

【００３３】

さらに、図２に示すように、メモリ２内の領域には、前述した正当結果９が予め格納されており、この正当結果９は、ワークメモリ内データ８と同行同列の同数の画素データによって形成されている。

【００３４】

グラフィックアクセラレータ４は、前記所定の演算を開始すると、図２における（１）の番号が付された矢印に示すように、メモリ２から、第１番目の１ブロック分のプログラム本体イメージ５ａを読み込むようになっている。

【００３５】

また、このとき、グラフィックアクセラレータ４は、図２における（２）の番号が付された矢印に示すように、ワークメモリ７から、初期値０が設定されたワークメモリ内データ８を読み込むようになっている。

【００３６】

そして、グラフィックアクセラレータ４は、図２における（３）の番号に示すように、読み込まれた１ブロック分のプログラム本体イメージ５ａと、ワークメモリ内データ８との間で、互いに対応する同行同列に位置する画素データの組ごとに、論理演算としての排他的論理和（ＸＯＲ）を計算するようになっている。

【００３７】

例えば、１ブロック分のプログラム本体イメージ５ａにおける第ｍ行第ｎ列目の画素データは、ワークメモリ内データ８における第ｍ行第ｎ列目の画素データとの間で排他的論理和が計算されるようになっている。

【００３８】

ここで、排他的論理和は、１Ｂｉｔのデータで考えると、演算に係る２つのデータの値が互いに等しい場合（「０」同士または「１」同士の場合）には０となり、互いに異なる場合（一方が「０」、他方が「１」の場合）には１となる。したがって、例えば、１ブロック分のプログラム本体イメージ５ａにおける第ｍ行第ｎ列目の画素データが、「１」に相当する白色のデータで、ワークメモリ内データ８における第ｍ行第ｎ列目の画素データが、「０」に相当する黒色のデータの場合には、排他的論理和は「１」に相当する白色のデータとなる。

【００３９】

なお、この排他的論理和は、１つの画素データが１Ｂｉｔで表現される場合には、ラスタ演算となる。

【００４０】
そして、グラフィックアクセラレータ４は、図２における（４）の番号が付された矢印に示すように、前記画素データの組ごとの排他的論理和の結果を、ワークメモリ７におけるワークメモリ内データ８を読み込んだ位置と同じ位置に格納（上書き）するようになっている。
【００４１】
例えば、第ｍ行第ｎ列目の画素データの組を用いた排他的論理和の結果は、ワークメモリ７における第ｍ行第ｎ列目の画素の位置に格納されることになる。
【００４２】
これにより、図２に示すように、ワークメモリ内データ８が更新され、第１回目の排他的論理和の結果が、ワークメモリ７内に白黒画像として描画されることになる。
【００４３】
次に、グラフィックアクセラレータ４は、図２における（５）の番号が付された矢印に示すように、メモリ２から第２番目の１ブロック分のプログラム本体イメージ５ｂを読み込むようになっている。
【００４４】
そして、グラフィックアクセラレータ４は、この第２番目の１ブロック分のプログラム本体イメージ５ｂに対しても、第１番目のときと同様に、ワークメモリ内データ８との排他的論理和を互いに対応する同行同列に位置する画素データの組ごとに計算し、その結果をワークメモリ７に格納するようになっている。
【００４５】
このような一連の処理が、第５番目（最後）のプログラム本体イメージ５ｅに至るまで順次繰り返し行われることによって、前記所定演算結果として、ワークメモリ７に最終的に格納されたワークメモリ内データ８（最終描画結果）が取得されるようになっている。
【００４６】
そして、ＣＰＵ３は、図２における（６）の番号が付された矢印に示すように、ワークメモリ７に格納された所定演算結果と、正当結果９とを比較し、双方の結果が一致する場合には、メモリ２に格納されたプログラム本体イメージ５が破損のない正当なものであるとみなすようになっている。
【００４７】
このように、本実施形態においては、グラフィックアクセラレータ４により、メモリ２に保持されたプログラム本体イメージ５に対して、このプログラム本体イメージ５を画像として扱った所定の演算を行い、所定演算結果が正当結果９と一致するか否かに基づいて、メモリ２に保持されたプログラム本体イメージの正当性を迅速に検証することが可能となる。
【００４８】
また、前述した排他的論理和は、演算が簡便である上に、所定演算結果が「０」に相当する黒色のデータに偏りやすい論理積（ＡＮＤ）や、所定演算結果が「１」に相当する白色のデータに偏りやすい論理和（ＯＲ）の場合に比べて、黒色のデータと白色のデータとが混在する白黒画像を得やすい演算であるため、プログラム本体イメージ５の破損を検出しやすい。したがって、このような排他的論理和を含む前記所定の演算を行うことによって、メモリ２に保持されたプログラム本体イメージ５の正当性をより迅速かつ適切に検証することが可能となる。
【００４９】
さらに、一般的なグラフィックアクセラレータ４が備えているソースデータとデスティネーションデータとの論理演算を行う機能を活用して、メモリ２に保持されたプログラム本体イメージ５をソースデータとし、ワークメモリ内データ８をデスティネーションデータとした排他的論理和を簡便に実行することが可能となる。
【００５０】
より好ましい実施形態としては、ＣＰＵ３により、グラフィックアクセラレータ４に、

所定演算結果と正当結果９との排他的論理和を互いに対応する同行同列に位置する画素データの組ごとに計算させるようにする。

【００５１】

そして、この排他的論理和の結果が、すべての画素データの組において「０」に相当する黒色のデータとなる場合に、ＣＰＵ３により、所定演算結果が正当結果９と一致すると判定し、メモリ２に格納されたプログラム本体イメージ５が正当であるとみなすようにする。

【００５２】

このようにすれば、メモリ２に格納されたプログラム本体イメージ５の正当性をさらに簡便に検証することが可能となる。

【００５３】

次に、本実施形態の作用について図３を参照して説明する。

【００５４】

なお、初期状態において、メモリ２には、プログラム本体イメージ５がハードディスクドライブ等から読み込まれて保持されているものとする。

【００５５】

そして、初期状態から、メモリ２に保持されたプログラム本体イメージ５の正当性の検証処理を開始すると、ＣＰＵ３は、図３のステップ１（ＳＴ１）において、ブートローダ等を用いて初期値０のワークメモリ内データ８が格納されたワークメモリ７を用意する。そして、メモリ２に格納されたプログラム本体イメージ５をソースデータとし、ワークメモリ内データ８をデスティネーションデータとして、グラフィックアクセラレータ４をセットアップする。

【００５６】

次いで、ステップ２（ＳＴ２）において、グラフィックアクセラレータ４は、メモリ２から読み込む１ブロック分のプログラム本体イメージ５ａ～５ｅのブロック番号ｊを１に設定してステップ３（ＳＴ３）に進む。

【００５７】

次いで、ステップ３（ＳＴ３）において、グラフィックアクセラレータ４は、メモリ２から、第ｊ番目（ＳＴ３の直後においては第１番目）の１ブロック分のプログラム本体イメージ５ａ～５ｅを読み込む。

【００５８】

次いで、ステップ４（ＳＴ４）において、グラフィックアクセラレータ４は、ワークメモリ７からワークメモリ内データ８を読み込む。このステップ４（ＳＴ４）の処理は、ステップ３（ＳＴ３）と同時に行うようにしてもよい。

【００５９】

次いで、ステップ５（ＳＴ５）において、グラフィックアクセラレータ４は、ステップ３（ＳＴ３）において読み込んだ１ブロック分のプログラム本体イメージ５ａ～５ｅと、ステップ４（ＳＴ４）において読み込んだワークメモリ内データ８との排他的論理和（ＸＯＲ）を、互いに対応する画素データの組ごとに計算する。

【００６０】

次いで、ステップ６（ＳＴ６）において、グラフィックアクセラレータ４は、ステップ５（ＳＴ５）における排他的論理和の計算結果をワークメモリ７に格納する。

【００６１】

次いで、ステップ７（ＳＴ７）において、グラフィックアクセラレータ４は、ステップ５（ＳＴ５）における排他的論理和が行われた１ブロック分のプログラム本体イメージ５ａ～５ｅのブロック番号ｊが、最後のブロックの番号（ｊ＝５）であるか否かを判定し、ｊ＝５の場合には、ステップ８（ＳＴ８）に進み、そうでない場合（ｊ＜５の場合）には、ステップ９（ＳＴ９）に進む。

【００６２】

ステップ８（ＳＴ８）に進む場合は、ステップ６までの処理によって前記所定の演算が

完了したことになる。

【００６３】

一方、ステップ９（ＳＴ９）に進む場合には、未だに前記所定の演算が完了していないことになる。この場合は、ステップ９（ＳＴ９）において、ｊの値を１つ増やしてステップ３（ＳＴ３）に戻り、ｊ＝５となるまで、ステップ３（ＳＴ３）～ステップ７（ＳＴ７）およびステップ９（ＳＴ９）の処理を繰り返す。

【００６４】

次いで、ステップ８（ＳＴ８）において、ＣＰＵ３は、ステップ６（ＳＴ６）によって取得された所定演算結果が、メモリ２に保持された正当結果９と一致するか否かを判定し、一致する場合には、ステップ１０（ＳＴ１０）に進み、一致しない場合には、ステップ１１（ＳＴ１１）に進む。

【００６５】

ステップ１０（ＳＴ１０）において、ＣＰＵ３は、メモリ２に保持されたプログラム本体イメージ５が正当であると判定して正当性の検証処理を終了する。

【００６６】

一方、ステップ１１（ＳＴ１１）において、ＣＰＵ３は、メモリ２に保持されたプログラム本体イメージ５が正当でない（破損がある）と判定して正当性の検証処理を終了する。

【００６７】

以上述べたように、本実施形態によれば、グラフィックアクセラレータ４の演算機能を用いてメモリ２に保持されたプログラム本体イメージ５の検証を迅速に行うことができる結果、正当なプログラムをすみやかに実行することができ、ひいては、システムの起動時間を短縮することができる。

【００６８】

さらに、発明によれば、グラフィックアクセラレータ４の演算機能を用いてメモリ２に保持されたプログラム本体イメージ５の正当性の検証を行うことにより、ＣＰＵ３の負担を軽減してＣＰＵ３に空き時間を確保することができるので、メモリ２に保持されたプログラム本体イメージ５の正当性の検証が行われている最中においても、既に正当であるとみなされたプログラム本体イメージ５をＣＰＵ３によってすみやかに実行することができる。

【００６９】

なお、本発明は、前述した実施の形態に限定されるものではなく、必要に応じて種々の変更が可能である。

【００７０】

例えば、前述した実施形態においては、前記所定の演算の際に、初期値０のワークメモリ内データ８を用いたが、本発明は、このような構成に限定されるものではない。例えば、正当結果９におけるすべての画素データを０に相当する黒色のデータとし、ワークメモリ内データ８の初期値を、これに対応する値にしてもよい。

【００７１】

このようにすれば、所定演算結果におけるすべての画素データが０に相当する黒色のデータとなるか否かに基づいて、メモリ２に保持されたプログラム本体イメージ５が正当であるか否かを判定することができるため、前述した実施形態のように所定演算結果と正当結果９との排他的論理和を計算する場合よりも演算回数を削減することができる。この結果、メモリ２に保持されたプログラム本体イメージ５の正当性の検証をさらに迅速に行うことが可能となる。

【００７２】

また、前述した実施形態においては、１ブロック分のプログラム本体イメージ５ａ～５ｅとワークメモリ内データ８とのサイズが同一であるが、ワークメモリ内データ８のサイズを１ブロック分のプログラム本体イメージ５ａ～５ｅのサイズよりも大きくしてもよい。この場合、前述した排他的論理和の計算は、１ブロック分のプログラム本体イメージ５

ａ～５ｅと、ワークメモリ内データ８の一部（１ブロック分のプログラム本体イメージ５ａ～５ｅと同一サイズ分のデータ）との間で行い、その排他的論理和の結果を上書きすることによって、ワークメモリ内データ８の一部を更新するようにすればよい。なお、この場合、ワークメモリ内データ８におけるどの位置にある一部のデータを排他的論理和に用いるのか、また、ワークメモリ内データ８におけるどの位置にあるデータを排他的論理和の結果によって更新するのかについては、コンセプトに応じて種々変更することができる。

【００７３】

このようにすれば、ワークメモリ内データ８のデータ量を多くすることができるので、プログラム本体イメージ５の破損をさらに容易に検出することが可能となる。

【００７４】

さらに、前述した実施形態においては、１画素が１Ｂｉｔで表現されているが、１画素が、８Ｂｉｔや１６Ｂｉｔ等の複数のＢｉｔ数で表現されたものであってもよい。この場合には、前記所定の演算の際に、プログラム本体イメージ５における８Ｂｉｔまたは１６Ｂｉｔ分のデータが、１画素として捉えられることになる。

【００７５】

さらにまた、本発明を車載用ナビゲーションシステムに適用すれば、車載用ナビゲーションシステムをすみやかに起動することができる。

【図面の簡単な説明】

【００７６】

【図１】本発明に係るプログラム正当性検証装置の実施形態ににおいて、ハードウェア構成を示すブロック図

【図２】本発明に係るプログラム正当性検証装置の実施形態において、図１よりも詳細な構成を示すブロック図

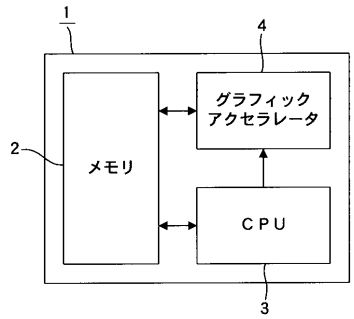
【図３】本発明に係るプログラム正当性検証装置の実施形態を示すフローチャート

【符号の説明】

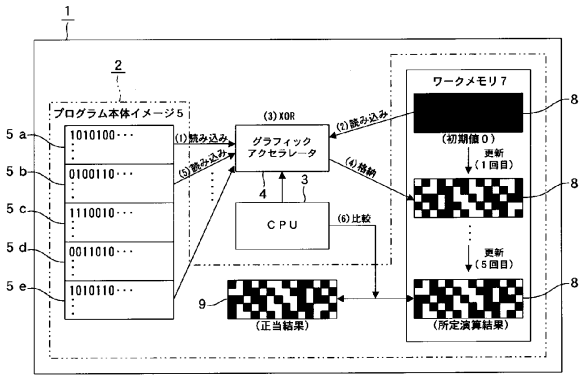
【００７７】

１　プログラム正当性検証装置
２　メモリ
３　ＣＰＵ
４　グラフィックアクセラレータ４
５　プログラム本体イメージ
７　ワークメモリ７
８　ワークメモリ内データ
９　正当結果

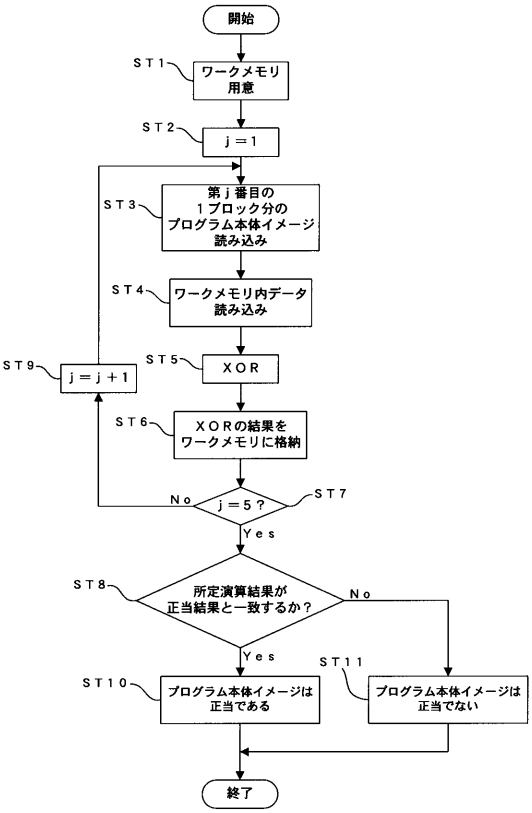
【図１】


1
4
2
メモリ
グラフィック
アクセラレータ
CPU
3

【図２】


1
2
プログラム本体イメージ5
5a
5b
5c
5d
5e
1010100…
0100110…
1110010…
0011010…
1010110…
(1)読み込み
(2)読み込み
(3)XOR
グラフィック
アクセラレータ
4
3
CPU
(6)比較
(7)読み込み
(4)格納
ワークメモリ7
8
(初期値0)
更新
(1回目)
更新
(5回目)
9
(正当結果)
(所定演算結果)

【図３】


開始
ST1
ワークメモリ
用意
ST2
j＝1
ST3
第j番目の
1ブロック分の
プログラム本体イメージ
読み込み
ST4
ワークメモリ内データ
読み込み
ST5
XOR
ST6
XORの結果を
ワークメモリに格納
ST7
j＝5？
No
Yes
ST9
j＝j＋1
ST8
所定演算結果が
正当結果と一致するか？
No
Yes
ST10
プログラム本体イメージは
正当である
ST11
プログラム本体イメージは
正当でない
終了

フロントページの続き

(74)代理人 100123858
弁理士 磯田 志郎
(72)発明者 中村 達喜
東京都品川区西五反田１丁目１番８号 アルパイン株式会社内

審査官 市川 武宜

(56)参考文献 特開昭６２－２４５３７５（ＪＰ，Ａ）
特開２００６－０５３７８７（ＪＰ，Ａ）
特開２００２－２３６６００（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ ２１／２２
Ｇ０６Ｆ ２１／２４