(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2011-198224
(P2011-198224A)
(43)公開日　平成23年10月6日(2011.10.6)

(51)Int.Cl.		F I			テーマコード（参考）
G06F 12/16	(2006.01)	G06F	12/16	310J	5B018
		G06F	12/16	320B	

審査請求 未請求　請求項の数 5　ＯＬ　（全 15 頁）

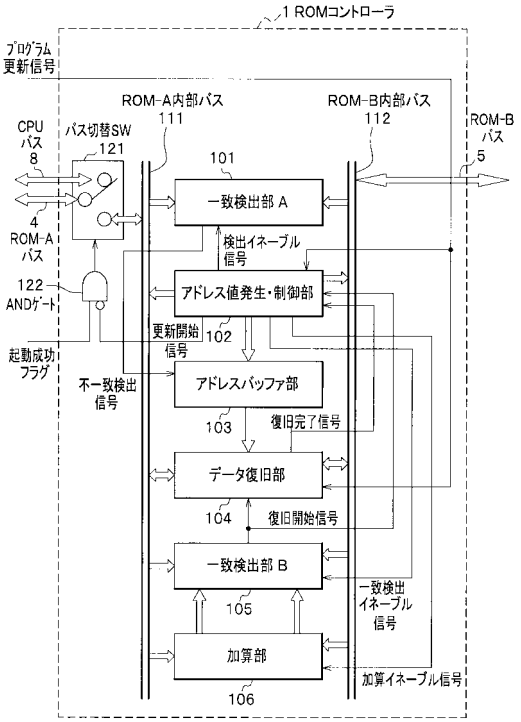
(21)出願番号　特願2010-66176(P2010-66176)
(22)出願日　平成22年3月23日(2010.3.23)

(71)出願人　000005108
株式会社日立製作所
東京都千代田区丸の内一丁目６番６号
(74)代理人　100064414
弁理士　磯野　道造
(74)代理人　100111545
弁理士　多田　悦夫
(72)発明者　宮越　正敏
茨城県ひたちなか市市毛１０７０番地　株式会社日立製作所都市開発システム社内

Ｆターム(参考)　5B018 GA02　HA04　HA13　KA22　MA23
NA06　QA11

(54)【発明の名称】情報処理装置および監視機器システム

(57)【要約】
【課題】データ復旧の信頼性を保持しつつ、データ復旧時間の短縮を目的とする。
【解決手段】それぞれのＲＯＭに格納されているデータが一致しているか否かをアドレス毎に判定する一致検出部Ａ１０１と、それぞれのＲＯＭに格納されているデータの値を加算して、第１チェックサム値を算出する加算部１０６と、第１チェックサム値と、予めＲＯＭに格納されている第２チェックサム値とを、それぞれのＲＯＭにおいて比較する一致検出部Ｂ１０５と、一致検出部Ａ１０１でデータの不一致が検出されたアドレスを記憶するアドレスバッファ部１０３と、第２の一致検出部において、チェックサム値が不一致であると判定された場合、チェックサム値が一致しているＲＯＭから、チェックサム値が一致していないＲＯＭへ、アドレスバッファ部１０３に記憶されているアドレスに該当するデータをコピーするデータ復旧部１０４と、ことを特徴とする。
【選択図】図３



【特許請求の範囲】
【請求項１】
同じデータを格納している、少なくとも２つの不揮発性記憶装置の内容をチェックする情報処理装置であって、
それぞれの前記不揮発性記憶装置に格納されているデータが一致しているか否かを、アドレス毎に判定する第１の一致検出部と、
前記第１の一致検出部において、データの不一致が検出された場合、前記不一致が検出されたアドレスを記憶するアドレス記憶処理部と、
前記それぞれの不揮発性記憶装置に格納されているデータの値を加算することにより、第１のチェックサム値を算出する加算部と、
前記第１のチェックサム値と、予め前記不揮発性記憶装置に格納されている第２のチェックサム値とを、前記それぞれの不揮発性記憶装置において比較する第２の一致検出部と、
前記第２の一致検出部において、一方の不揮発性記憶装置の前記第１のチェックサム値と、前記第２のチェックサム値と、が不一致であると判定された場合、前記第１のチェックサム値と、前記第２のチェックサム値と、が一致している不揮発性記憶装置から、前記第１のチェックサム値と、前記第２のチェックサム値と、が一致していない不揮発性記憶装置へ、前記アドレス記憶処理部に記憶されているアドレスに該当するデータをコピーするデータ復旧部と、
を有することを特徴とする情報処理装置。
【請求項２】
前記情報処理装置は、前記不揮発性記憶装置と、ＣＰＵとの間に接続される
ことを特徴とする請求項１に記載の情報処理装置。
【請求項３】
ＣＰＵによる前記不揮発性記憶装置へのアクセスがない状態となった後、前記情報処理装置自身の処理が開始される
ことを特徴とする請求項１に記載の情報処理装置。
【請求項４】
前記情報処理装置は、
一方の前記不揮発性記憶装置のデータの更新が検知されると、前記情報処理装置自身の処理を停止した後、
前記データ復旧部が、
データの更新が行われた不揮発性記憶装置のデータを、他方の前記不揮発性記憶装置へコピーする
ことを特徴とする請求項１に記載の情報処理装置。
【請求項５】
監視カメラまたはレコーダに、請求項１から請求項４のいずれか一項に記載の情報処理装置が搭載されている
ことを特徴とする監視機器システム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、データ復旧を行う情報処理装置および監視機器システムの技術に関する。
【背景技術】
【０００２】
監視機器が接続されている監視機器システムにおいて、監視機器システムが起動できない状態になると、復旧に手間がかかる。これは、頻繁に人が行かない場所、または人が行くのに困難を伴う場所に監視機器システム設置されることが多いためである。さらに、一度、監視機器システムが稼動を開始すると、２４時間を長期間に渡って稼動し続けることが多い。このため、監視機器システムは、長期間に渡って安定稼動できる状態を保つこと

が重要である。このような監視機器システムにおけるシステム動作用プログラムのデータ（プログラムデータ）は、フラッシュＲＯＭ（Read Only Memory）などの不揮発性ＲＯＭに格納される。通常、プログラムデータは、システム起動時に不揮発性ＲＯＭからＲＡＭ（Random Access Memory）へコピーされるため、システム起動後の不揮発性ＲＯＭ内のプログラムデータは、実行中のシステム動作には影響しない。

【０００３】

このような監視機器システムにおいて、不慮の停電が発生し、停電復旧後に再起動し、システム動作を再開すれば問題ないが、不揮発性ＲＯＭ内のプログラムデータの破壊（データ破壊）がある場合は再起動できないこととなる。つまり、不揮発性ＲＯＭのプログラムデータが破壊されていると、予期せぬ停電が生じた際に監視動作が停止してしまうこととなる。そして、管理者へ監視機器の異常状態の通知がされないと、監視動作が停止したまま放置されてしまうことになる。異常状態の通知がされたとしても、管理者が設置場所へ出向くことが困難な場合など、監視機器の動作停止は様々な問題を生じる。

【０００４】

このように、長期間に渡るシステム運用の結果、不揮発性ＲＯＭに格納していたシステム動作用プログラムデータの一部が破壊されてしまうことがしばしばある。これは、不揮発性ＲＯＭ固有の部品単体の問題であり、完全に防ぐことはできない。

【０００５】

このような不揮発性ＲＯＭに格納していたプログラムデータの一部破壊に関する従来技術として、特許文献１がある。特許文献１では、以下の情報処理システムおよび情報処理方法が記載されている。すなわち、特許文献１に記載の情報処理システムでは、システムを起動するための第１のプログラムおよび第１のプログラムの複製を記憶した不揮発性の第１の記憶装置と、複製の第１のプログラムが転送される揮発性の第２の記憶装置と、第１のプログラムを実行するための第２のプログラムを記憶する不揮発性の第３の記憶装置と、第１のプログラムを実行するＣＰＵ（Central Procesing Unit）とを備える。第２のプログラムには、第１の記憶装置から第２の記憶装置に複数の第１のプログラムを転送し、第２の記憶装置に転送された複数の第１のプログラムの内容を比較し、複数の第１のプログラムの内容に不一致があれば、複数の第１のプログラムの中から多数決によって正常プログラムを判定する命令が含まれる。そして、ＣＰＵは、正常プログラムと判定された第１のプログラムを実行してシステムの初期起動を行う技術が記載されている。

【０００６】

特許文献２には、起動用プログラムが格納される２つのフラッシュメモリを有し、フラッシュメモリに格納されたプログラムのチェックサムを定期的に求め、一方が不一致の場合、他方の正常なフラッシュメモリのプログラムを一方のフラッシュメモリに再度書き込むことで復旧させる情報処理装置、情報処理装置におけるプログラムデータ保護法方が記載されている。

【０００７】

そして、特許文献３には、ＲＡＭ上のファームウェアとＥＥＰＲＯＭ（Electrically Erasable and Programmable Read Only Memory）のファームウェアを定期的に比較し、もし不一致ならば、外部ファイル装置からファームウェアファイルをＥＥＰＲＯＭおよびＲＡＭに再度書き込みを行う記憶装置正常性確認装置および情報処理装置が記載されている。

【先行技術文献】

【特許文献】

【０００８】

【特許文献１】特開２００９－１８７１９９号公報

【特許文献２】特開２００７－２２００９２号公報

【特許文献３】特開平１１－１６１５５４号公報

【発明の概要】

【発明が解決しようとする課題】

【０００９】
通常ＣＰＵが、リセット解除後に読み出すアドレス値は固定値であり、このアドレス値を変更することはできない。
その為、システム内のメモリ配置構成は、ＣＰＵがリセット解除後に読み出すアドレス値として不揮発性ＲＯＭを配置する構成となる。ＣＰＵは、リセット解除後、この不揮発性ＲＯＭ内に格納されたプログラムコードを逐次実行していく。しかし、不揮発性ＲＯＭでは読み出し速度が遅いので、プログラムデータを不揮発性ＲＯＭから読み出し速度が速いＲＡＭへコピーし、ＲＡＭ上でプログラムを実行するように切り替えることで、プログラム実行速度を上げている。ＲＡＭ上にプログラムデータをコピーする際、システム内にある複数個の不揮発性ＲＯＭ、または不揮発性ＲＯＭ内の複数ブロック内からプログラムデータを読み出し、この読み出したプログラムデータにおけるデータ破壊がないことを確認してから、プログラムデータがＲＡＭへコピーされる処理が行われる。
【００１０】
不揮発性ＲＯＭ内のデータ破壊はどの部分で発生するか不定である。ＣＰＵはリセット解除直後に実行されるデータ破壊がないことを確認しながら不揮発性ＲＯＭからＲＡＭへコピーするコピープログラムを実行する。しかしながら、このコピープログラム自体が破壊されていると、システム起動ができないこととなる。
【００１１】
一方、プログラムデータの破壊に備えて不揮発性ＲＯＭを２つ設けるシステムの公知例も多数ある。しかしながら、起動時にプログラムデータが破壊された不揮発性ＲＯＭがシステム内に存在するとどの不揮発性ＲＯＭを使用するかを切り替える回路が必要となる。さらには、不揮発性ＲＯＭを読み出す度にプログラムデータのチェックをする必要があり、システム起動が遅くなってしまうという課題がある。
【００１２】
また、特許文献１に記載の技術では、ＲＯＭとＲＡＭをそれぞれ複数用意する必要があるため、システムの規模が大きくなってしまうという課題がある。
【００１３】
そして、特許文献２に記載の技術では、２つあるフラッシュメモリのどちらでデータ破壊が発生したかは分かるものの、フラッシュメモリ上の、どのアドレスでデータ破壊が発生したかは分からない。データ破壊が発生したアドレスが不明なため、プログラムデータの復旧はフラッシュメモリに格納されているすべてのプログラムデータを再書き込みする必要があり、プログラムデータの復旧に時間がかかってしまう。プログラムデータの復旧に時間がかかってしまうと、この復旧中に停電等電源断になる恐れがあり、システム起動の信頼性向上が図れない。
【００１４】
特許文献３に記載の技術では、外部にファイル装置を備えるシステムでしか適用できないという課題がある。また、ＣＰＵがプログラムの実行動作を行うＲＡＭとは別に確認用のＲＡＭを用意する必要があり必要ＲＡＭサイズが増大してしまうという課題もある。さらに、プログラム実行用ＲＡＭとプログラムデータ確認用のＲＡＭを兼用すると、ＲＡＭ上のデータ破壊が発生した場合、その後ＣＰＵは正常動作できず、外部ファイル装置からファームウェアファイルを読み出し、ＥＥＰＲＯＭおよびＲＡＭに再度書き込みを行うことができなくなるという課題がある。
【００１５】
このような背景に鑑みて本発明がなされたのであり、本発明は、データ復旧の信頼性を保持しつつ、データ復旧時間の短縮を目的とする。
【課題を解決するための手段】
【００１６】
前記課題を解決するため、本発明は、同じデータを格納している、少なくとも２つの不揮発性記憶装置の内容をチェックする情報処理装置であって、それぞれの前記不揮発性記憶装置に格納されているデータが一致しているか否かを、アドレス毎に判定する第１の一

致検出部と、前記第１の一致検出部において、データの不一致が検出された場合、前記不一致が検出されたアドレスを記憶するアドレス記憶処理部と、前記それぞれの不揮発性記憶装置に格納されているデータの値を加算することにより、第１のチェックサム値を算出する加算部と、前記第１のチェックサム値と、予め前記不揮発性記憶装置に格納されている第２のチェックサム値とを、前記それぞれの不揮発性記憶装置において比較する第２の一致検出部と、前記第２の一致検出部において、一方の不揮発性記憶装置の前記第１のチェックサム値と、前記第２のチェックサム値と、が不一致であると判定された場合、前記第１のチェックサム値と、前記第２のチェックサム値と、が一致している不揮発性記憶装置から、前記第１のチェックサム値と、前記第２のチェックサム値と、が一致していない不揮発性記憶装置へ、前記アドレス記憶処理部に記憶されているアドレスに該当するデータをコピーするデータ復旧部と、を有することを特徴とする。

その他の解決手段は、実施形態中にて記載する。

【発明の効果】

【００１７】

本発明によれば、データ復旧の信頼性を保持しつつ、データ復旧時間の短縮が可能となる。

【図面の簡単な説明】

【００１８】

【図１】本実施形態に係る情報処理システムの構成例を示す図である。

【図２】本実施形態に係るＲＯＭアドレスマップのイメージを示す図である。

【図３】ＲＯＭコントローラの機能ブロックの構成例を示す図である。

【図４】本実施形態に係る情報処理システムの処理の流れを示すフローチャートである。

【図５Ａ】本実施形態に係るＲＯＭチェック処理の流れを示すフローチャートである（その１）。

【図５Ｂ】本実施形態に係るＲＯＭチェック処理の流れを示すフローチャートである（その１）。

【図６】本実施形態に係るプログラム更新処理の流れを示すフローチャートである。

【図７】プログラムデータが複数のブロックに分けられてＲＯＭに格納されている場合におけるＲＯＭアドレスマップのイメージを示す図である。

【図８】本実施形態に係る情報処理システムを搭載した監視機器システムの構成例を示す図である。

【発明を実施するための形態】

【００１９】

次に、本発明を実施するための形態（「実施形態」という）について、適宜図面を参照しながら詳細に説明する。なお、本実施形態では、プログラムデータのデータ破壊のチェックおよび復旧を想定しているが、プログラムデータ以外の数値データなどに適用してもよい。

【００２０】

（情報処理システムの構成）

図１は、本実施形態に係る情報処理システムの構成例を示す図である。

情報処理システム１０は、ＣＰＵ３、ＲＡＭ２、ＲＯＭコントローラ１（情報処理装置）、２つの不揮発性記憶装置（ＲＯＭ９）であるＲＯＭ－Ａ６、ＲＯＭ－Ｂ７を有している。

ＣＰＵ３は、起動時にＲＯＭ－Ａ６に格納されているプログラムデータをＲＡＭ２上にコピーする。そして、実際のＣＰＵ３のプログラム実行動作はＲＡＭ２上において行われる。

ＲＯＭ－Ａ６は、プログラムデータを格納しており、ＲＯＭ－Ｂ７は、ＲＯＭ－Ａ６に格納されているプログラムデータのコピーがバックアップのために格納されている。

ＲＯＭコントローラ１は、ＲＯＭ－Ａ６と、ＲＯＭ－Ｂ７のプログラムデータが一致しているか否かを判定し、一致していない場合（データ破壊が生じている場合）にはデータ

破壊の生じていないＲＯＭ９からデータ破壊が生じているＲＯＭ９へプログラムデータをコピーする。
なお、ＲＯＭ－Ａ６とＲＯＭ－Ｂ７のうち、いずれのＲＯＭ９にデータ破壊が生じているかの判定は後記する。
【００２１】
ＣＰＵ３と、ＲＡＭ２と、ＲＯＭコントローラ１とは、ＣＰＵバス８で接続されており、ＲＯＭコントローラ１とＲＯＭ－Ａ６とはＲＯＭ－Ａバス４で接続されており、ＲＯＭコントローラ１とＲＯＭ－Ｂ７とはＲＯＭ－Ｂバス５で接続されている。
また、ＲＯＭコントローラ１にはプログラムの起動成功を示す起動成功フラグがＣＰＵ３から入力される。
【００２２】
（ＲＯＭのアドレスマップ）
図２は、本実施形態に係るＲＯＭアドレスマップのイメージを示す図である。
プログラムデータは、ＲＯＭ９の先頭アドレスであるアドレス「０」から最終アドレスの１つ手前（アドレス値「ｎ－１」）まで格納されている。そして、最終アドレス（アドレス値「ｎ」）にはチェックサム値が格納されている。チェックサム値は、先頭アドレスのプログラムデータ値から、順次アドレス「ｎ－１」までのプログラムデータ値を加算した値の下ｎ桁が格納されている。ここで、「プログラムデータ値」とは、１６進法や、８進法などで表現される数値としてのプログラムデータである。ｎ桁の値は、ＲＯＭ－Ａ６、ＲＯＭ－Ｂ７のデータビット幅により決定され、例えば、８ビット幅では２桁であり、１６ビット幅では４桁となる。例えば、１６ビット幅であれば、先頭アドレス「０」からアドレス「ｎ－１」まで、ＤＦ＋５Ｈ＋・・・＋ＢＢといったようにプログラムデータ値の加算が行われると、その加算値は当然２桁を超える値となるため、下２桁をチェックサム値として保存することになる。
【００２３】
（ＲＯＭコントローラの構成）
図３は、ＲＯＭコントローラの機能ブロックの構成例を示す図である。
ＲＯＭコントローラ１は、一致検出部Ａ（第１の一致検出部）１０１、アドレス値発生・制御部１０２、アドレスバッファ部（アドレス記憶処理部）１０３、データ復旧部１０４、一致検出部Ｂ（第２の一致検出部）１０５、加算部１０６を有している。各回路１０１～１０６は、ＲＯＭ－Ａ内部バス１１１、ＲＯＭ－Ｂ内部バス１１２を介して接続されている。
バス切替ＳＷ（Switch）１２１は、ＣＰＵバス８、ＲＯＭ－Ａバス４、ＲＯＭ－Ａ内部バス１１１に接続されており、ＡＮＤゲート１２２を介して入力される起動成功フラグと、更新開始信号との論理積によってＲＯＭ－Ａバス４とＲＯＭ－Ａ内部バス１１１とを切り替える。
なお、図３における各信号は図５Ａおよび図５Ｂにおいて後記して説明する。
【００２４】
一致検出部Ａ１０１は、ＲＯＭ－Ａ６およびＲＯＭ－Ｂ７におけるアドレス毎のプログラムデータの一致・不一致を検出する機能を有する。
アドレス値発生・制御部１０２は、先頭アドレスから順に最終アドレスまでのアドレス値を発生させると共に、各種信号を出力し、各部１０１，１０３～１０６を制御する機能を有する。
アドレスバッファ部１０３は、プログラムデータの不一致が生じているアドレス値を保存する機能を有する。
データ復旧部１０４は、プログラムデータの不一致が生じている場合（データ破壊が生じている場合）、データ破壊が生じていないＲＯＭ９からデータ破壊が生じているＲＯＭ９へプログラムデータをコピーする機能を有する。
【００２５】
一致検出部Ｂ１０５は、一致検出部Ａ１０１で不一致が検出された場合、各ＲＯＭ９の

チェックサム値を比較することによって、どちらのＲＯＭ９にデータ破壊が生じているかを判定する機能を有する。
加算部１０６は、アドレス毎のプログラムデータ値を逐次加算する機能を有する。
【００２６】
バス切替ＳＷ１２１は、ＲＯＭ－Ａバス４の接続先をＣＰＵバス８、およびＲＯＭ－Ａ内部バス１１１のどちらかに切り替えるバスＳＷである。バス切替ＳＷ１２１は、「０」が入力されると、ＣＰＵバス８とＲＯＭ－Ａバス４とを接続し、「１」が入力されるとＲＯＭ－Ａバス４と、ＲＯＭ－Ａ内部バス１１１とを接続する。バス切替ＳＷ１２１には、ＣＰＵ３から送られる起動成功フラグと、アドレス値発生・制御部１０２から送られる更新開始信号の否定信号が入力される。ここで、更新開始信号とは、ＲＯＭ９に格納されているプログラムデータの更新が開始される際にアドレス値発生・制御部１０２から出力される信号である。
すなわち、バス切替ＳＷ１２１は、起動成功フラグが入力されると、ＲＯＭ－Ａバス４と、ＲＯＭ－Ａ内部バス１１１とを接続し、アドレス値発生・制御部１０２が更新開始信号を出力すると、ＣＰＵバス８とＲＯＭ－Ａバス４とを接続する。
【００２７】
各部１０１～１０５は、ＲＯＭコントローラ１中の図示しないＲＯＭに格納されているプログラムを、ＲＯＭコントローラ中の図示しないＣＰＵが起動することによって具現化されてもよいし、各部１０１～１０５のそれぞれを専用のＩＣ（Integrated Circuit）チップで具現化してもよい。
【００２８】
（情報処理システムの処理）
次に、図１などを参照しつつ、図４に沿って本実施形態に係る情報処理システム１０の処理を説明する。
図４は、本実施形態に係る情報処理システムの処理の流れを示すフローチャートである。
一般的にＣＰＵ３は、電源の投入後、リセット信号が解除されると自発的にＲＯＭ９に格納されたプログラムデータの読み込みを開始する（Ｓ１０１）。
そして、ＲＯＭコントローラ１は、ＣＰＵバス８と、ＲＯＭ－Ａバス４との接続を行う（Ｓ１０２）。これによりＣＰＵ３は電源投入時において、２つあるＲＯＭ９のうち、常にＲＯＭ－Ａ６のプログラムデータをＲＡＭ２にコピーすることができる。
その後　ＣＰＵ３は、ＲＯＭ－Ａ６に格納されたプログラムデータをＲＡＭ２上にコピーし（Ｓ１０３）、ＣＰＵ３はＲＡＭ２上にてプログラム実行動作を行う。
ＣＰＵ３がＲＯＭ－Ａ６からＲＡＭ２へプログラムデータのコピー作業を完了すると、プログラムの起動完了となり、その後は、ＲＯＭ－Ａ６からプログラムデータを読み出す必要がなくなる。つまり、これ以降、ＣＰＵによるＲＯＭ９へのアクセスがない状態となる。
【００２９】
このとき、ＣＰＵ３から起動成功フラグがＲＯＭコントローラ１に入力され（Ｓ１０４）、ＲＯＭコントローラ１はＣＰＵバス８からＲＯＭ－Ａバス４を切り離す（Ｓ１０５）。
以降、ＲＯＭ－Ａバス４とＲＯＭ－Ｂバス５に対するバスマスタはＲＯＭコントローラ１となり、ＣＰＵバス８の動作とは無関係にＲＯＭコントローラ１はＲＯＭ－Ａ６およびＲＯＭ－Ｂ７へのアクセスが可能となる。
【００３０】
前記したように、予めＲＯＭ－Ａ６とＲＯＭ－Ｂ７へは同じプログラムデータが格納されている。ステップＳ１０４でＣＰＵ３からＲＯＭコントローラ１に起動成功フラグが入力され、ステップＳ１０５の切り離しが実行されると、ＲＯＭコントローラ１は、図５Ａおよび図５Ｂで後記するＲＯＭチェック処理を行う（Ｓ１０６）。
【００３１】

（ＲＯＭチェック処理）
次に、図１および図３を参照しつつ、図５Ａおよび図５Ｂに沿って図４のステップＳ１０６におけるＲＯＭチェック処理の説明を行う。
図５Ａおよび図５Ｂは、本実施形態に係るＲＯＭチェック処理の流れを示すフローチャートである。
情報処理システム１０の電源が投入されると、バス切替ＳＷ１２１はＣＰＵバス８とＲＯＭ－Ａバス４とを接続する（図５Ａのステップ２０１：図４のＳ１０２に相当）。これによりＣＰＵ３は、ＲＯＭ－Ａ６からプログラムデータを読み出しシステム起動が可能となる。
ＣＰＵ３がＲＯＭ－Ａ６からＲＡＭ２へのプログラムデータ読出が完了すると（Ｓ２０２：図４のＳ１０３に相当）、起動成功フラグが、ＣＰＵ３からＡＮＤゲート１２２を介してバス切替ＳＷ１２１に入力される（Ｓ２０３：図４のＳ１０４に相当）。すると、バス切替ＳＷ１２１は、ＣＰＵバス８を、ＲＯＭ－Ａバス４から切り離し、ＲＯＭコントローラ１内部のＲＯＭ－Ａ内部バス１１１と、ＲＯＭ－Ａバス４とを接続する（Ｓ２０４）。以降、ＲＯＭコントローラ１はＲＯＭ－Ａ６へ自由にアクセス可能でとなる。なお、図３に示すように、ＲＯＭ－Ｂバス５は、ＲＯＭコントローラ１内部のＲＯＭ－Ｂ内部バス１１２に常時接続されている。
【００３２】
なお、ステップＳ２０３において、ＣＰＵ３から入力された起動成功フラグは、バス切替ＳＷ１２１への入力と同時にアドレス値発生・制御部１０２へも入力される（図３に図示せず）。
起動成功フラグが入力されたアドレス値発生・制御部１０２は、アドレス値をアドレス値「０」から最終アドレス値「ｎ」まで順次発生させていく（Ｓ２０５）。このとき、アドレス値発生・制御部１０２は、検出イネーブル信号を一致検出部Ａ１０１へ出力し、加算イネーブル信号を加算部１０６へ出力する。ただし、アドレス値が最終アドレス値である「ｎ」である場合、アドレス値発生・制御部１０２は加算イネーブルを出力しない。
ＲＯＭコントローラ１は、アドレス値発生・制御部１０２から出力されたアドレス値に対応するプログラムデータを、ＲＯＭ－Ａ６から、ＲＯＭ－Ａバス４およびＲＯＭ－Ａ内部バス１１１を経由して取得すると（Ｓ２０６）、このプログラムデータを一致検出部Ａ１０１、加算部１０６へ入力する。
同時に、ＲＯＭコントローラ１は、アドレス値発生・制御部１０２が発生したアドレス値に該当するＲＯＭ－Ｂ７のプログラムデータも同様に、ＲＯＭ－Ｂバス５およびＲＯＭ－Ｂ内部バス１１２を介して取得する（Ｓ２０７）と、ＲＯＭ－Ｂ内部バス１１２を経由して一致検出部Ａ１０１、加算部１０６へ入力する。なお、この時点における加算部１０６の加算値は初期値としての「０」である。
【００３３】
次に、一致検出部Ａ１０１は、ＲＯＭ－Ａ内部バス１１１、ＲＯＭ－Ｂ内部バス１１２
それぞれから入力されたプログラムデータ（アドレス値発生・制御部１０２が発生したアドレス値に該当するプログラムデータ）が一致しているかを、アドレス値発生・制御部１０２から入力される検出イネーブル信号のタイミングで判定する（Ｓ２０８）。
ステップＳ２０８の結果、プログラムデータ値が一致した場合（Ｓ２０８→Ｙｅｓ）、ＲＯＭコントローラ１はステップＳ２１１へ処理を進める。
ステップＳ２０８の結果、プログラムデータ値の不一致を検出した場合（Ｓ２０８→Ｎｏ）、一致検出部Ａ１０１は、不一致検出信号をアドレスバッファ部１０３へ出力する。
不一致検出信号が入力されたアドレスバッファ部１０３は、不一致検出信号が入力された時においてアドレス値発生・制御部１０２が発生しているアドレス値をアドレスバッファ部１０３自身に保存する（Ｓ２０９）。この保存されたアドレス値はプログラムデータの復旧の際に用いられる。
【００３４】
また、前記したように、ＲＯＭ－Ａ６のプログラムデータ値とＲＯＭ－Ｂ７のプログラムデータ値は、加算部１０６へも入力される。各プログラムデータ値を入力された加算部

１０６は、自身が保持している加算値と、ＲＯＭ－Ａ６およびＲＯＭ－Ｂ７から入力されたそれぞれのプログラムデータ値とを、アドレス値発生・制御部１０２から入力される加算イネーブル信号のタイミングで加算し（Ｓ２１０）、それぞれの加算値を自身に保存する。

【００３５】

次に、アドレス値発生・制御部１０２がアドレス値発生・制御部１０２で発生されたアドレス値が「ｎ」であるか否かを判定する（Ｓ２１１）。

ステップＳ２１１の結果、アドレス値が「ｎ」でない場合（Ｓ２１１→Ｎｏ）、ＲＯＭコントローラ１は、ステップＳ２０５へ処理を戻す。

ステップＳ２１１の結果、アドレス値が「ｎ」である場合（Ｓ２１１→Ｙｅｓ）、アドレス値発生・制御部１０２から一致検出イネーブル信号が一致検出部Ｂ１０５へ出力される。すると、この一致検出イネーブル信号を受信した一致検出部Ｂ１０５は、加算部１０６からＲＯＭ－Ａ６およびＲＯＭ－Ｂ７に関するチェックサム値（第１チェックサム値と称する）を取得し、ＲＯＭ－Ａバス４、ＲＯＭ－Ａ内部バス１１１を介してＲＯＭ－Ａ６に格納されているチェックサム値（第２チェックサム値と称する）を取得する。さらに、一致検出部Ｂ１０５は、ＲＯＭ－Ｂバス５、ＲＯＭ－Ｂ内部バス１１２を介してＲＯＭ－Ｂ７に格納されているチェックサム値（第２チェックサム値）も取得する（Ｓ２１２）。なお、第１チェックサム値は、第２チェックサム値と同じ下ｎ桁を取得する。

【００３６】

そして、一致検出部Ｂ１０５は、取得したＲＯＭ－Ａ６およびＲＯＭ－Ｂ７に関するそれぞれの第１チェックサム値および第２チェックサム値を比較する（Ｓ２１３）。

【００３７】

すなわち、ＲＯＭコントローラ１内で先頭アドレスから順次加算してきた値（第１チェックサム値）と最終アドレスから読み出した値（第２チェックサム値）が一致していれば、ＲＯＭ９でデータ破壊は発生していないと判断できる。２つのＲＯＭ９それぞれにおいて第１チェックサム値と、第２チェックサム値が一致していれば、同時に２つのＲＯＭ９に対してデータ破壊が発生する可能性は、ほぼ「０」であるため、この場合はどちらのＲＯＭ９にもデータ破壊が発生していないと判定しても問題ない。

【００３８】

プログラムデータを復旧する際、ＲＯＭ－Ａ６またはＲＯＭ－Ｂ７のどちらが破壊されていたかによって、復旧元（コピー元）と復旧先（コピー席）が決定される。ＲＯＭ－Ａ６でデータ破壊を検出した場合、ＲＯＭコントローラ１はアドレスバッファ部１０３に記憶していたアドレス値に該当するプログラムデータをＲＯＭ－Ｂ７から読み出し、ＲＯＭ－Ａ６へ上書きすることで破壊されたプログラムデータの復旧動作を行う。つまり、ＲＯＭ－Ａ６の第１チェックサム値と第２チェックサム値が不一致であり、ＲＯＭ－Ｂ７の第１チェックサム値が一致していれば、ＲＯＭコントローラ１は、ＲＯＭ－Ａ６のプログラムデータが破壊していると判定する。ＲＯＭ－Ｂ７でデータ破壊を検出した場合、ＲＯＭコントローラ１はＲＯＭ－Ａ６からプログラムデータを読み出し、ＲＯＭ－Ｂ７へ上書きすることで破壊されたプログラムデータの復旧動作を行う。

【００３９】

そして、一致検出部Ｂ１０５は、ステップＳ２１３の比較の結果、どちらか一方のＲＯＭ９でチェックサム値の不一致を検出したか否かを判定する（Ｓ２１４）。

ステップＳ２１４の結果、チェックサム値の不一致を検出しなかった場合（Ｓ２１４→Ｎｏ）、２つのＲＯＭ９において、データ破壊が生じている確率は非常に低いため、ＲＯＭ－Ａ６およびＲＯＭ－Ｂ７に格納されているプログラムデータは正常であると判定し、ＲＯＭコントローラ１はステップＳ２０５へ処理を戻し、アドレス値「０」からチェック処理を再開する。

ステップＳ２１４の結果、チェックサム値の不一致を検出した場合（Ｓ２１４→Ｙｅｓ）、一致検出部Ｂ１０５は、ステップＳ２１３の比較結果を含む復旧開始信号をデータ復旧部１０４へ出力する。この復旧開始信号には、どちらのＲＯＭ９が正常で、どちらのＲ

ＯＭ９のプログラムデータが破壊されているかの情報が含まれている。
　このとき、アドレス値が「０」に戻り、ステップＳ２０５の処理をアドレス値発生・制御部１０２が行わないように、すなわち、プログラムデータの復旧の間、アドレス値発生・制御部１０２を停止させるため、復旧開始信号はアドレス値発生・制御部１０２へも入力される。
【００４０】
　復旧開始信号が入力されたデータ復旧部１０４は、不一致が検出されたアドレス値を保持しているアドレスバッファ部１０３から当該アドレス値を取得する（Ｓ２１５）。
　そして、データ復旧部１０４は、復旧開始信号に含まれるコピー元のＲＯＭ９からステップＳ２１５で取得したアドレス値で示されるプログラムデータを取得すると、復旧信号に含まれるコピー先のＲＯＭ９におけるステップＳ２１５で取得したアドレス値へプログラムデータをコピーする（Ｓ２１６）ことにより、プログラムデータの復旧を行う。
【００４１】
　そして、データ復旧部１０４は、アドレスバッファ部１０３に保持されていたすべてのアドレス値に関してコピーが完了したか否かを判定する（Ｓ２１７）。
　ステップＳ２１７の結果、コピー完了していない場合（Ｓ２１７→Ｎｏ）、ステップＳ２１６へ処理を戻す。
　ステップＳ２１７の結果、コピーが完了している場合（Ｓ２１７→Ｙｅｓ）、データ復旧部１０４は、復旧完了信号をアドレス値発生・制御部１０２へ出力する。
　プログラムデータの復旧の為に、一時停止していたアドレス値発生・制御部１０２は、復旧完了信号を受けると、図５Ａのステップ２０５へ処理を戻し、アドレス値を「０」から順次発生・出力していく処理を再開する。
【００４２】
　以上の動作を行うことにより、ＲＯＭ９に格納されているプログラムデータのデータ破壊検出動作と自己復旧動作を行なう。これにより常にＲＯＭ－Ａ６には、正常なプログラムデータが格納されシステム起動の信頼性の保持と、システム起動時間の短縮が可能となる。また、アドレスバッファに保持されているアドレス値に該当するプログラムデータのみをコピーするため、復旧時間の短縮が可能となる。
【００４３】
（プログラムデータ更新時）
　情報処理システム１０は、工場から出荷された後、性能向上の為、ＲＯＭ９のプログラムデータの更新を行うことがある。更新プログラムデータは、外部から情報処理システム１０に入力されるのでＣＰＵ３を用いて更新する必要がある。しかし、本実施形態に係る情報処理システム１０では、プログラムの起動後において、ＣＰＵ３は、ＲＯＭ－Ａ６へのアクセスは可能であるが、ＲＯＭ－Ｂ７へのアクセスが不可能である。これでは、プログラム起動時において、本実施形態の目的であるＲＯＭ－Ａ６とＲＯＭ－Ｂ７とに同じプログラムデータを格納することによって、データ破壊を検出し、自己復旧する処理を実現することができない。つまり、図５Ａのステップ２０８でＲＯＭ－Ａ６およびＲＯＭ－Ｂ７における個々のアドレス値に該当するプログラムデータ値が不一致と判定されても、図５Ｂのステップ２１３のチェックサム値の比較の段階で、ＲＯＭ－Ａ６およびＲＯＭ－Ｂ７の双方の第２チェックサム値が、第１チェックサム値と一致してしまうため、一致検出部Ｂ１０５は、双方のプログラムデータにおいてデータ破壊はなしと判断してしまうことになる。
　このような状態を避けるため、ＲＯＭコントローラ１は以下に説明する処理を行う。
【００４４】
　図６は、本実施形態に係るプログラム更新処理の流れを示すフローチャートである。
　ＣＰＵ３は、プログラムデータを更新する際、プログラム更新信号をアドレス値発生・制御部１０２とデータ復旧部１０４のそれぞれへ入力する（Ｓ３０１）。
　プログラム更新信号が入力されたアドレス値発生・制御部１０２とデータ復旧部１０４は、現在行っている処理（すなわち、図５Ａのステップ２０５および図５Ｂのステップ

Ｓ２１６）を停止させる（Ｓ３０２）。
次に、アドレス値発生・制御部１０２は、現在行っている動作を停止させたら、更新開始信号をバス切替ＳＷ１２１へ出力する。更新開始信号の否定（「０」）が入力されたバス切替ＳＷ１２１は、ＲＯＭ－Ａバス４をＣＰＵバス８に接続するよう切り替える（Ｓ３０３）。これによりＣＰＵ３は、ＲＯＭ－Ａ６へアクセスできプログラムデータの更新が可能となる。

【００４５】
次に、ＣＰＵ３は、ＲＯＭ－Ａ６へのプログラムデータ更新を行い（Ｓ３０４）、プログラムデータ更新が完了すると、ＣＰＵ３はプログラム更新信号を更新前の状態（「０」）に戻す）。プログラム更新信号が更新前に戻ったことを検知したアドレス値発生・制御部１０２は、アドレス値を「０」へリセットする（Ｓ３０５）。
そして、アドレス値発生・制御部１０２は、プログラム更新信号が更新前の状態に戻ったのに合わせ、更新開始信号を更新前の状態に戻す。更新前の状態に戻った更新開始信号の否定（「１」）が入力されたバス切替ＳＷ１２１は、ＲＯＭ－Ａバス４と、ＲＯＭ－Ａ内部バス１１１との接続に切り替える（Ｓ３０６）。
次に、アドレス値発生・制御部１０２が「０」から「ｎ」までのアドレス値を順次発生し、データ復旧部１０４は、アドレス値発生・制御部１０２が発生したアドレス値に相当するプログラムデータをＲＯＭ－Ａ６からＲＯＭ－Ｂ７へコピーする（Ｓ３０７）。なお、最終アドレスに格納されるチェックサム値（第２チェックサム値）は予め計算され更新プログラムデータに格納されている。
コピーが完了すると、データ復旧部１０４は、復旧完了信号をアドレス値発生・制御部１０２に出力する。
このとき、アドレス値発生・制御部１０２のアドレス値は、「０」に戻っているので、ＲＯＭコントローラ１は図５Ａ、図５ＢのＲＯＭチェック処理を再開する（Ｓ３０８）。

【００４６】
図６の処理により、システム運用中のプログラム更新へも対応可能なプログラムデータのデータ破壊検出処理と自己復旧処理が実現できる。更に２つのプログラム格納用ＲＯＭ９を搭載していてもＣＰＵ３から見るとＲＯＭ９２つのうちの１つ分のプログラム更新時間で済み、ＣＰＵ３に対するプログラム更新のシステム負荷を軽減できる。また。ユーザから個々のＲＯＭ－Ａ６またはＲＯＭ－Ｂ７に対してプログラム更新の指示入力が不要になる。

【００４７】
（ＲＯＭのアドレスマップの別の例）
図７は、プログラムデータが複数のブロックに分けられてＲＯＭに格納されている場合におけるＲＯＭアドレスマップのイメージを示す図である。
図７に示すように、プログラムデータが「プログラムデータＡ」がアドレス値「０」、「１」～「ｋ－１」に、「プログラムデータＢ」がアドレス値「ｋ＋１」、「ｋ＋２」～「ｍ－１」に、「プログラムデータＣ」がアドレス値「ｍ＋１」、「ｍ＋２」～「ｎ－１」の各ブロックに格納されていた場合、各プログラムデータの一番最後に各プログラムデータのチェックサム値が格納される。
すなわち、アドレス値「ｋ」には、「プログラムデータＡ」のチェックサム値が、アドレス値「ｍ」には、「プログラムデータＢ」のチェックサム値が、アドレス値「ｎ」には、「プログラムデータＣ」のチェックサム値が格納される。そして、ＲＯＭ９の最終アドレス値「ｎ＋１」には全体のチェックサム値が格納される。

【００４８】
このようにすることで、ＲＯＭコントローラ１は、各プログラムデータ（ブロック）毎にデータ破壊検出処理と自己復旧処理を行うことができる。
この場合、図５の処理はプログラムデータ毎に行われることになる。

【００４９】
（監視機器システム）

図８は、本実施形態に係る情報処理システムを搭載した監視機器システムの構成例を示す図である。

監視機器システム３０は、アナログビデオカメラ（監視カメラ）１９、ネットワークカメラ（監視カメラ）２３、レコーダ２１、モニタテレビ２２、外部ＰＣ（Personal Computer）２５を有してなる。

アナログビデオカメラ１９は、アナログ映像信号を出力する機能を有し、アナログカメラの映像信号をレコーダ２１へ伝送する同軸ケーブル２０でレコーダ２１に接続されている。

ネットワークカメラ２３は、映像信号をネットワーク回線２４を介してレコーダ２１へ出力する機能を有している。

レコーダ２１は、アナログビデオカメラ１９やネットワークカメラ２３から入力された映像信号を記録する機能を有する。

モニタテレビ２２は、レコーダ２１に入力された映像信号や、レコーダ２１で再生された映像信号を表示する機能を有する。

外部ＰＣ２５は、遠隔にて監視機器システム３０を制御する機能を有する。

【００５０】

なお、アナログビデオカメラ１９やネットワークカメラ２３は天井など一度設置されたらその後は簡単に操作できない場所に複数台をレコーダ２１に接続することが一般的である。

そして、図８の構成では、アナログビデオカメラ１９や、ネットワークカメラ２３や、レコーダ２１に本実施形態の情報処理装置が搭載されている。

【００５１】

本実施形態のＲＯＭ９は、プログラム起動後、ＣＰＵ３の読み込みが行われないものであればよく、例えば、フラッシュＲＯＭなどを使用することも可能である。

なお、本実施形態では、一致検出部Ａ１０１におけるアドレス毎のプログラムの一致・不一致検出と並行して、加算部１０６における第１チェックサム値の算出が行われているが、一致検出回路Ａ１０１で不一致が検出された後、第１チェックサム値を算出してもよい。

また、本実施形態ではＲＯＭ９を２つとしているが、３つ以上でもよい。この場合、ＲＯＭ－Ａ６以外のすべてのＲＯＭ９がＲＯＭコントローラ１に常時接続されていてもよいし、切替ＳＷによってＲＯＭ－Ａ６以外のＲＯＭ９でＲＯＭコントローラ１に常時接続されるＲＯＭ９が切り替えられてもよい。なお、ＲＯＭ９が３つ以上である場合、一致検出部Ａ１０１や、一致検出部Ｂ１０５や、加算部１０６はすべてのＲＯＭ９に対して、本実施形態と同様の処理を行ってもよい。

【００５２】

（まとめ）

本実施形態によれば、情報処理システム１０の運用中にＲＯＭ９内のプログラムデータが破壊してもプログラムデータを自己復旧することができ、また、自己復旧によるシステム起動が遅延を防止しながら確実なシステム起動を期待できる。

つまり、チェックサム値の比較によって、データ復旧の信頼性を保持しつつ、不一致が検出されたアドレスに該当するデータのみを復旧することで、データ復旧時間を短縮することができる。

また、本実施形態によれば、これまで通り、外部ファイル装置を必要とせず、またＲＡＭ２を１つ搭載すればよい。

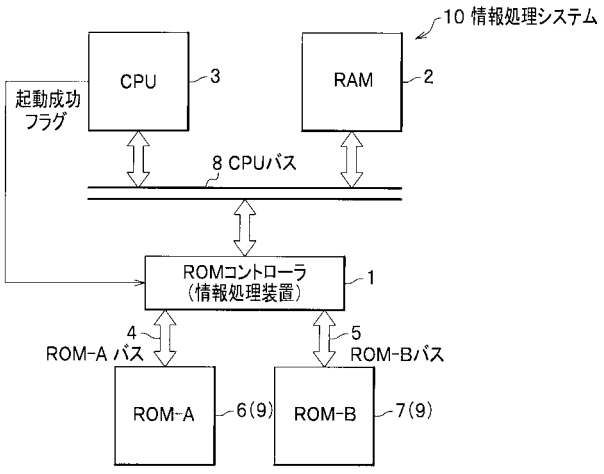
【符号の説明】

【００５３】

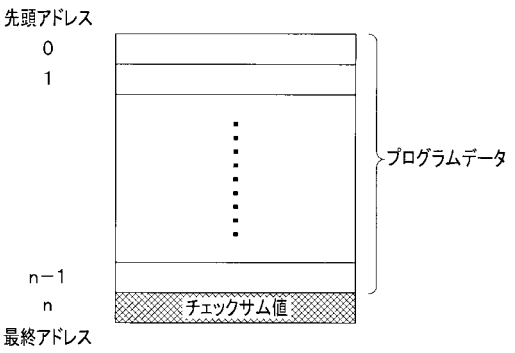
１　ＲＯＭコントローラ（情報処理装置）
２　ＲＡＭ
３　ＣＰＵ
４　ＲＯＭ－Ａバス

５　　ＲＯＭ－Ｂバス
６　　ＲＯＭ－Ａ（不揮発性記憶装置）
７　　ＲＯＭ－Ｂ（不揮発性記憶装置）
８　　ＣＰＵバス
９　　ＲＯＭ
１０　情報処理システム
１９　アナログビデオカメラ（監視カメラ）
２１　レコーダ
２３　ネットワークカメラ（監視カメラ）
３０　監視機器システム
１０１　一致検出部Ａ（第１の一致検出部）
１０２　アドレス値発生・制御部
１０３　アドレスバッファ部（アドレス記憶処理部）
１０４　データ復旧部
１０５　一致検出部Ｂ（第２の一致検出部）
１０６　加算部
１１１　ＲＯＭ－Ａ内部バス
１１２　ＲＯＭ－Ｂ内部バス
１２１　バス切替ＳＷ
１２２　ＡＮＤゲート

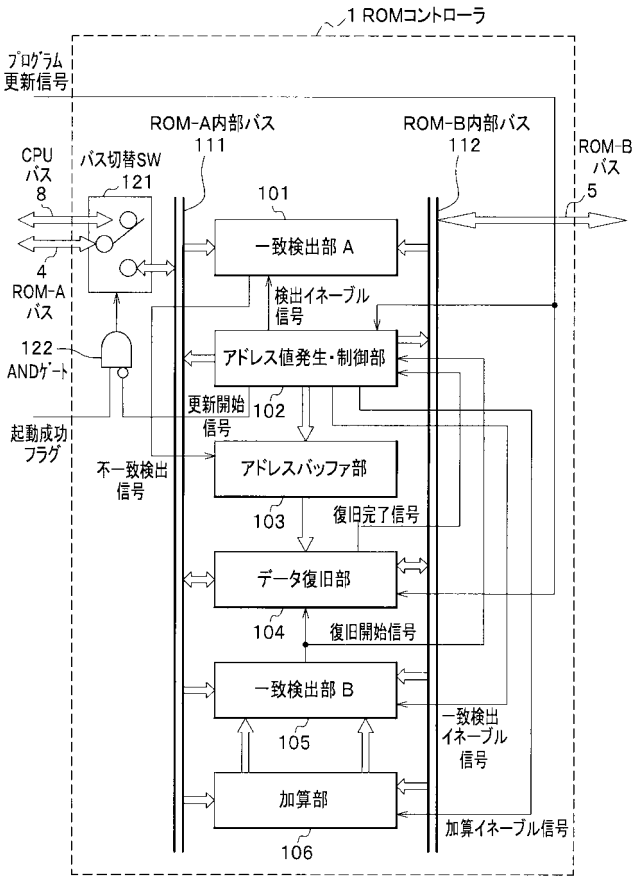
【図１】



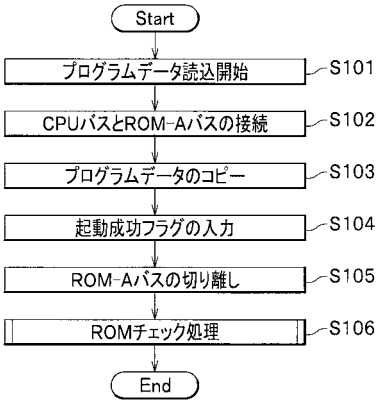
【図２】



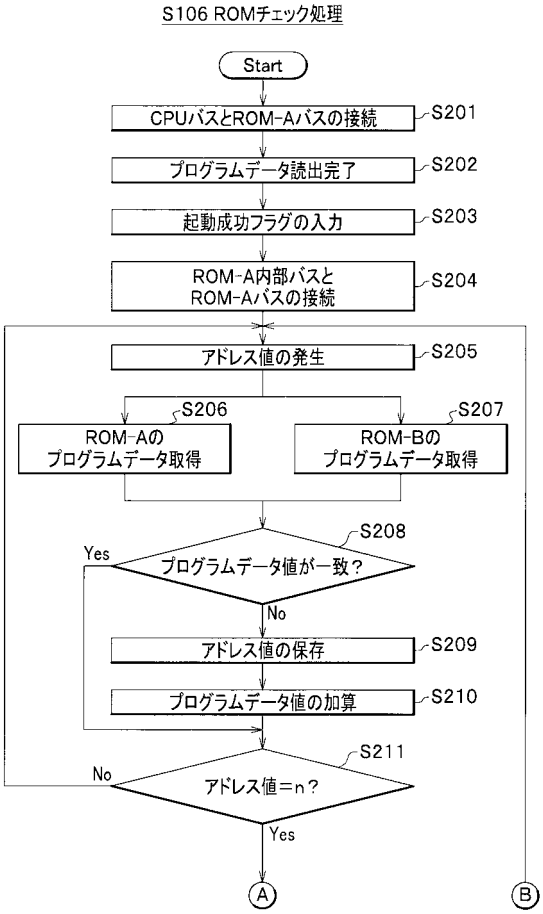
【図３】



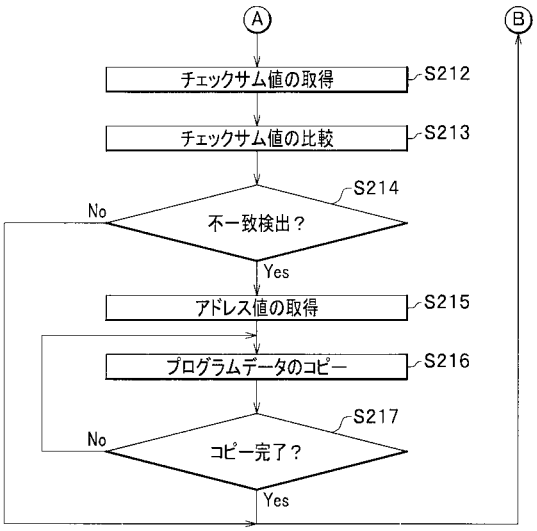
【図４】


Start
プログラムデータ読込開始
S101
CPUバスとROM-Aバスの接続
S102
プログラムデータのコピー
S103
起動成功フラグの入力
S104
ROM-Aバスの切り離し
S105
ROMチェック処理
S106
End

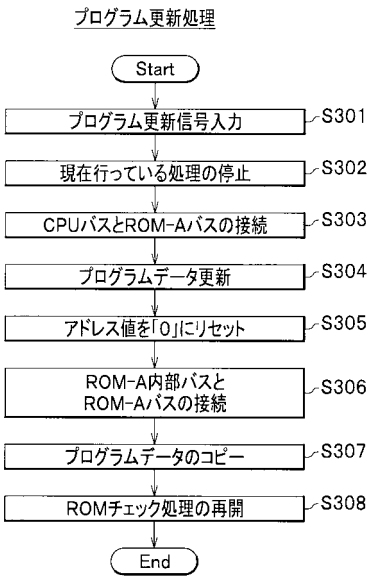
【図５Ａ】


S106 ROMチェック処理
Start
CPUバスとROM-Aバスの接続
S201
プログラムデータ読出完了
S202
起動成功フラグの入力
S203
ROM-A内部バスと
ROM-Aバスの接続
S204
アドレス値の発生
S205
S206
ROM-Aの
プログラムデータ取得
S207
ROM-Bの
プログラムデータ取得
S208
Yes
プログラムデータ値が一致？
No
アドレス値の保存
S209
プログラムデータ値の加算
S210
S211
No
アドレス値＝n？
Yes
A
B

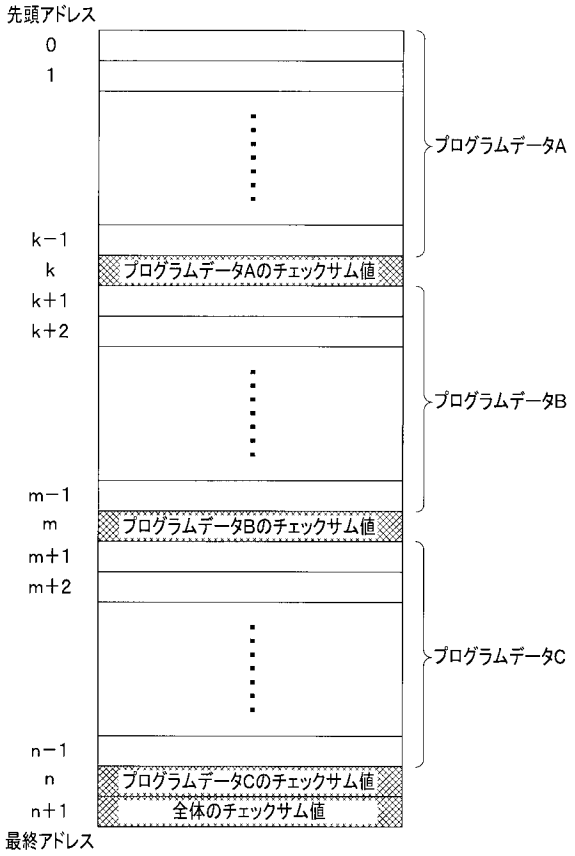
【図５Ｂ】


A
B
チェックサム値の取得
S212
チェックサム値の比較
S213
S214
No
不一致検出？
Yes
アドレス値の取得
S215
プログラムデータのコピー
S216
S217
No
コピー完了？
Yes

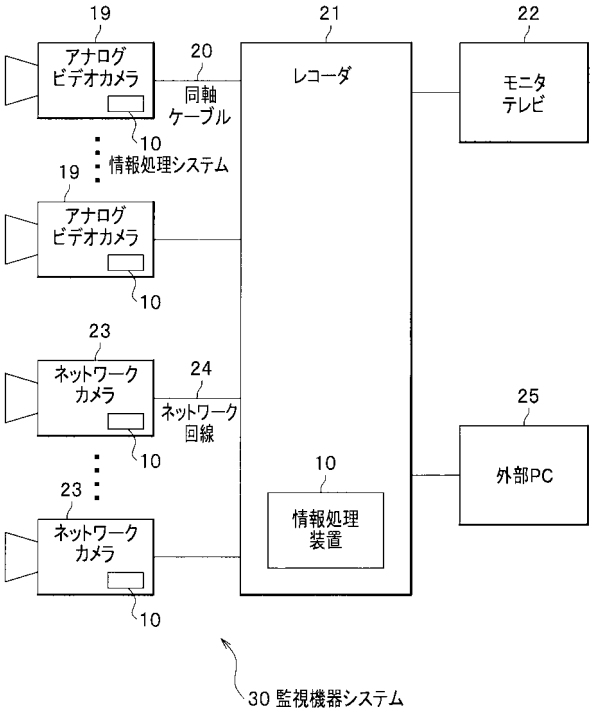
【図６】


プログラム更新処理
Start
プログラム更新信号入力
S301
現在行っている処理の停止
S302
CPUバスとROM-Aバスの接続
S303
プログラムデータ更新
S304
アドレス値を「0」にリセット
S305
ROM-A内部バスと
ROM-Aバスの接続
S306
プログラムデータのコピー
S307
ROMチェック処理の再開
S308
End

【図７】


先頭アドレス
0
1
k−1
k
k+1
k+2
m−1
m
m+1
m+2
n−1
n
n+1
最終アドレス
プログラムデータAのチェックサム値
プログラムデータBのチェックサム値
プログラムデータCのチェックサム値
全体のチェックサム値
プログラムデータA
プログラムデータB
プログラムデータC

【図８】


19
アナログ
ビデオカメラ
20
同軸
ケーブル
21
レコーダ
22
モニタ
テレビ
10
情報処理システム
19
アナログ
ビデオカメラ
10
23
ネットワーク
カメラ
24
ネットワーク
回線
25
外部PC
10
情報処理
装置
23
ネットワーク
カメラ
10
30 監視機器システム