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

特許第5019210号
(P5019210)

(45)発行日　平成24年9月5日(2012.9.5)　(24)登録日　平成24年6月22日(2012.6.22)

(51)Int.Cl.		F I		
G06K 19/073	(2006.01)	G06K	19/00	P
G06F 12/16	(2006.01)	G06F	12/16	310A
G06F 21/02	(2006.01)	G06F	21/02	175
		G06F	21/02	177

請求項の数 12　(全 23 頁)

(21)出願番号　特願2007-157870 (P2007-157870)
(22)出願日　平成19年6月14日 (2007.6.14)
(65)公開番号　特開2008-310595 (P2008-310595A)
(43)公開日　平成20年12月25日 (2008.12.25)
審査請求日　平成22年3月16日 (2010.3.16)

(73)特許権者　000003078
株式会社東芝
東京都港区芝浦一丁目１番１号
(74)代理人　100088683
弁理士　中村　誠
(74)代理人　100108855
弁理士　蔵田　昌俊
(74)代理人　100075672
弁理士　峰　隆司
(74)代理人　100109830
弁理士　福原　淑弘
(74)代理人　100084618
弁理士　村松　貞男

最終頁に続く

(54)【発明の名称】携帯可能電子装置、ＩＣカード、および携帯可能電子装置の制御方法

(57)【特許請求の範囲】
【請求項１】
外部装置から供給されるコマンドに応じて動作する携帯可能電子装置において、
複数ビット長の第１のデータを格納する第１の記憶手段と、
当該携帯可能電子装置の状態を示す情報として、前記第１の記憶手段に記憶されている前記第１のデータ、あるいは、前記第１のデータとは異なる第２のデータの何れかを格納する第２の記憶手段と、
前記第２の記憶手段に記憶されているデータが前記第１のデータあるいは前記第２のデータの何れかと一致するか否かを判断する判断手段と、
この判断手段により前記第２の記憶手段に記憶されているデータが前記第１のデータあるいは前記第２のデータの何れかと一致すると判断した場合、前記第２の記憶手段に記憶されている内容に応じて外部装置から供給されるコマンドに対応する本処理を実行する実行手段と、
を有することを特徴とする携帯可能電子装置。
【請求項２】
さらに、乱数を生成する乱数生成手段を有し、
前記第１の記憶手段は、前記乱数生成手段により生成された乱数に基づくデータを第１のデータとして格納する、
ことを特徴とする前記請求項１に記載の携帯可能電子装置。
【請求項３】

前記乱数生成手段は、互いに異なる２つの乱数を生成し、
前記第１の記憶手段は、前記乱数生成手段により生成された一方の乱数に基づく第１のデータと前記乱数生成手段により生成された他方の乱数に基づく第２のデータとを記憶する、
ことを特徴とする前記請求項２に記載の携帯可能電子装置。
【請求項４】
さらに、前記判断手段により前記第２の記憶手段に記憶されているデータが前記第１のデータあるいは前記第２のデータの何れかとも一致しないと判断した場合、当該携帯可能電子装置の動作を停止する停止手段を有することを特徴とする前記請求項１又は３に記載の携帯可能電子装置。
【請求項５】
さらに、前記判断手段により前記第２の記憶手段に記憶されているデータが前記第１のデータあるいは前記第２のデータの何れかとも一致しないと判断した場合、前記第２の記憶手段に異常が発生していることを示す情報を前記外部装置へ送信するエラー処理手段を有することを特徴とする前記請求項１又は３に記載の携帯可能電子装置。
【請求項６】
さらに、前記外部装置から受信したコマンドに対応する処理を実行するための実行条件をチェックし、そのチェック結果を前記第２の記憶手段に記憶する処理を行う前処理手段を有する、
ことを特徴とする前記請求項１乃至５に記載の携帯可能電子装置。
【請求項７】
さらに、各種のデータを記憶する第３の記憶手段と、
前記第３の記憶手段に記憶されている各種のデータに対するアクセス権を示す情報を前記第２の記憶手段に格納する処理を行う前処理手段と、を有する、
ことを特徴とする前記請求項１乃至５に記載の携帯可能電子装置。
【請求項８】
さらに、使用者を認証する認証手段を有し、
前記前処理手段は、前記認証手段により認証が成功した使用者に対する各種のデータに対するアクセス権を示す情報を前記第２の記憶手段に格納する処理を行う、
ことを特徴とする前記請求項７に記載の携帯可能電子装置。
【請求項９】
前記判断手段は、前記第２の記憶手段に記憶されているデータが前記第１のデータと一致する場合に第１の状態であることを示す信号を出力し、前記第２の記憶手段に記憶されているデータが前記第２のデータと一致する場合には第２の状態であることを示す信号を出力し、前記第２の記憶手段に記憶されているデータが前記第１のデータあるいは前記第２のデータの何れとも一致しない場合にはエラー状態であることを示す信号を出力する比較回路である、
ことを特徴とする前記請求項１乃至８に記載の携帯可能電子装置。
【請求項１０】
前記第２の記憶手段は、第１のデータあるいは第２のデータを格納するための複数のフラグを有し、
前記第１の記憶手段は、前記第２の記憶手段の各フラグに対応する種々の第１のデータを格納する、
ことを特徴する前記請求項１乃至９に記載の携帯可能電子装置。
【請求項１１】
外部装置から供給されるコマンドに応じて動作する携帯可能電子装置に用いられる制御方法であって、
複数ビット長の第１のデータを第１の記憶手段に格納しておき、
当該携帯可能電子装置の状態を示す情報として、前記第１の記憶手段に記憶されている前記第１のデータ、あるいは、前記第１のデータとは異なる第２のデータの何れかを前記

第２の記憶手段に格納し、
前記第２の記憶手段に記憶されているデータが前記第１のデータあるいは前記第２のデータの何れかと一致するか否かを判断し、
この判断により前記第２の記憶手段に記憶されているデータが前記第１のデータあるいは前記第２のデータの何れかと一致すると判断した場合、前記第２の記憶手段に記憶されている内容に応じて外部装置から供給されるコマンドに対応する本処理を実行する、
ことを特徴とする携帯可能電子装置の制御方法。
【請求項１２】
外部装置から供給されるコマンドに応じて動作するＩＣカードにおいて、
複数ビット長の第１のデータを格納する第１の記憶手段と、当該携帯可能電子装置の状態を示す情報として、前記第１の記憶手段に記憶されている前記第１のデータ、あるいは、前記第１のデータとは異なる第２のデータの何れかを格納する第２の記憶手段と、前記第２の記憶手段に記憶されているデータが前記第１のデータあるいは前記第２のデータの何れかと一致するか否かを判断する判断手段と、この判断手段により前記第２の記憶手段に記憶されているデータが前記第１のデータあるいは前記第２のデータの何れかと一致すると判断した場合、前記第２の記憶手段に記憶されている内容に応じて外部装置から供給されるコマンドに対応する本処理を実行する実行手段とを具備するモジュールと、
前記モジュールを具備する本体と、
を有することを特徴とするＩＣカード。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、例えば、個人情報あるいは取引情報などが記憶されているＩＣチップが内蔵されているＩＣカードあるいはＩＣタグなどの携帯可能電子装置、および、上記携帯可能電子装置の制御方法などに関する。
【背景技術】
【０００２】
近年、ＩＣカードなどの携帯可能電子装置は、様々な用途に利用されている。特に、ＩＣカードには、個人情報や金銭的な取引情報などが記憶されることが多い。このようなＩＣカードあるいはＩＣカードを用いたシステムでは、高いセキュリティ性が求められており、不正なアクセスを確実に防止する技術が求められている。一方、近年、ＩＣカードに対して電源にノイズを印加したり、電磁波、光波あるいは温度などのストレスを与えたりして、ＩＣカードを誤動作させることにより、ＩＣカード内の内部情報あるいは処理手順などを解析しようとする脅威が高まっている。
【０００３】
従来、たとえば、特開昭６０－２０７９５７号公報（特許文献１）、あるいは、特開平１１－２８２９９１号公報（特許文献２）には、ＩＣカードに対するコマンドの種類と実行順とに基づいて不正なアクセスを検出する技術が記載されている。これらの技術では、ＩＣカード内のメモリに格納するフラグ（識別情報）に基づいて処理の実行順などを監視する。しかしながら、当該ＩＣカード内のメモリの情報は、上述のような外的な要因により変化してしまう可能性がある。このような場合、ＩＣカード内のメモリに格納されるフラグなどの情報も変化してしまう可能性がある。すなわち、外的な要因によりＩＣカード内のメモリの情報が変化されると、上記したような技術を適用してもＩＣカードが誤動作する可能性があるという問題点がある。
【特許文献１】特開昭６０－２０７９５７号公報
【特許文献２】特開平１１－２８２９９１号公報
【発明の開示】
【発明が解決しようとする課題】
【０００４】
この発明の一形態は、セキュリティ性が高い携帯可能電子装置および携帯可能電子装置

の制御方法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】
【０００５】
この発明の一形態としての携帯可能電子装置は、外部装置から供給されるコマンドに応じて動作するものにおいて、複数ビット長の第１のデータを格納する第１の記憶手段と、当該携帯可能電子装置の状態を示す情報として、前記第１の記憶手段に記憶されている前記第１のデータ、あるいは、前記第１のデータとは異なる第２のデータの何れかを格納する第２の記憶手段と、前記第２の記憶手段に記憶されているデータが前記第１のデータあるいは前記第２のデータの何れかと一致するか否かを判断する判断手段と、この判断手段により前記第２の記憶手段に記憶されているデータが前記第１のデータあるいは前記第２のデータの何れかと一致すると判断した場合、前記第２の記憶手段に記憶されている内容に応じて外部装置から供給されるコマンドに対応する本処理を実行する実行手段とを有する。
【０００６】
この発明の一形態としての携帯可能電子装置の制御方法は、外部装置から供給されるコマンドに応じて動作する携帯可能電子装置に用いられる方法であって、複数ビット長の第１のデータを第１の記憶手段に格納しておき、当該携帯可能電子装置の状態を示す情報として、前記第１の記憶手段に記憶されている前記第１のデータ、あるいは、前記第１のデータとは異なる第２のデータの何れかを前記第２の記憶手段に格納し、前記第２の記憶手段に記憶されているデータが前記第１のデータあるいは前記第２のデータの何れかと一致するか否かを判断し、この判断により前記第２の記憶手段に記憶されているデータが前記第１のデータあるいは前記第２のデータの何れかと一致すると判断した場合、前記第２の記憶手段に記憶されている内容に応じて外部装置から供給されるコマンドに対応する本処理を実行する。
【発明の効果】
【０００７】
この発明の一形態によれば、セキュリティ性が高い携帯可能電子装置および携帯可能電子装置の制御方法を提供することができる。
【発明を実施するための最良の形態】
【０００８】
以下、この発明に係る実施の形態について図面を参照しつつ説明する。
図１は、第１の実施の形態に係る携帯可能電子装置としてのＩＣカード１およびＩＣカード１を含むＩＣカードシステムの構成例を示すブロック図である。
上記ＩＣカード１は、外部装置としてのＩＣカード処理装置２からの電源供給により動作可能な状態となる。動作可能となったＩＣカード１は、上記ＩＣカード処理装置２からのコマンドに応じて種々の処理を行う。上記ＩＣカード処理装置２は、ＩＣカード１を動作させるための電源を供給するとともに、当該ＩＣカード１に対して種々の処理を要求するコマンドを供給する。上記ＩＣカード処理装置２がＩＣカード１に対して供給するコマンドは、用途あるいは運用形態などに応じた処理を要求するものである。
【０００９】
また、上記ＩＣカード１は、アンテナあるいは無線通信部等により上記ＩＣカード処理装置２と非接触の状態で無線通信を行う非接触式の携帯可能電子装置（非接触式ＩＣカード）であっても良し、上記ＩＣカード処理装置２と物理的に接触して通信を行う接触式の携帯可能電子装置（接触式ＩＣカード）であっても良い。さらには、上記ＩＣカード１は、非接触式ＩＣカードとしての通信機能と接触式ＩＣカードとしての通信機能とを有する複合型のＩＣカード（デュアルインターフェースＩＣカード）であっても良い。なお、第１および第２の実施の形態では、主に、非接触式ＩＣカードを想定して説明する。非接触式ＩＣカードと接触式ＩＣカードとはＩＣカード処理装置２との通信方式等が異なるだけである。このため、以下に説明する実施の形態は、接触式ＩＣカードにも同様に適用できる。

【００１０】
次に、上記ＩＣカード１の構成例について説明する。
図１に示すように、上記ＩＣカード１は、ＣＰＵ１０、プログラムメモリ１１、ワーキングメモリ１２、データメモリ１３、乱数生成部１４、通信制御部１５、電源部１６、および、インターフェース１７などにより構成される。
また、上記ＩＣカード１は、カード状の本体により構成される。上記ＩＣカード１を形成するカード状の本体には、１つ（あるいは複数）のＩＣチップ１ａとアンテナ１７とが埋設される。上記ＩＣチップ１ａは、ＣＰＵ１０、プログラムメモリ１１、ワーキングメモリ１２、データメモリ１３、通信制御部１５および電源部１６などにより構成される。上記ＩＣチップ１ａは、上記インターフェース１７としてのアンテナに接続された状態でモジュール化され、当該ＩＣカード１を形成するカード状の本体内に埋設される。たとえば、図２は、非接触式ＩＣカード全体の構成例を示す図である。図２に示す非接触式ＩＣカードは、カード状の本体１ｃを有している。この本体１ｃ内には、図２に点線で示すように、１つ（あるいは複数）のＩＣチップ１ａとアンテナ１７とを有するモジュール１ｂが埋め込まれている。
【００１１】
上記ＣＰＵ１０は、ＩＣカード１全体の制御を司るものである。上記ＣＰＵ１０は、データを記憶するための内部メモリとしてのレジスタ１０ａを有している。上記ＣＰＵ１０は、上記プログラムメモリ１１あるいはデータメモリ１３に記憶された制御プログラムおよび制御データなどに基づいて動作する。上記ＣＰＵ１０は、基本的な動作を司る制御プログラムを実行することにより、外部装置から与えられるコマンドに応じた処理を実行する。たとえば、外部装置から上記データメモリ１３へのデータの書込みを要求するコマンドが与えられれば、上記ＣＰＵ１０は、上記データメモリ１３へのデータの書き込み処理を実行する。また、外部装置から上記データメモリ１３に記憶されているデータの読み出しを要求するコマンドが与えられれば、上記ＣＰＵ１０は、上記データメモリ１３からのデータの読み出し処理を実行する。さらに、上記ＣＰＵ１０は、当該ＩＣカード１の用途などに応じてインストールされる処理プログラムを実行することにより、用途に応じた処理を実現するようになっている。
【００１２】
上記プログラムメモリ１１は、読み出し専用のメモリ（ＲＯＭ：リードオンリーメモリ）により構成される。上記プログラムメモリ１１には、予め基本動作を司る制御プログラムおよび制御データなどが記憶されている。上記プログラムメモリ１１には、予め当該ＩＣカード１の仕様に応じた制御プログラム及び制御データが記憶される。たとえば、上記ＣＰＵ１０は、上記プログラムメモリ１１に記憶される制御プログラムにより外部から与えられるコマンドに応じた処理を実現する。また、上記プログラムメモリ１１には、データメモリ１３におけるメモリ領域の属性などを指定するための情報なども記憶されている。
【００１３】
上記ワーキングメモリ１２は、揮発性のメモリ（ＲＡＭ；ランダムアクセスメモリ）により構成される。上記ワーキングメモリ１２は、データを一時保管するバッファメモリとして機能する。例えば、上記ワーキングメモリ１２には、ＩＣカード処理装置（外部装置）２との通信処理において、送受信されるデータが一時的に保管される。また、上記ワーキングメモリ１２には、種々の書込みデータなどを一時的に保持するメモリとしても利用される。さらに、上記ワーキングメモリ１２には、当該ＩＣカード１における処理状況あるいは設定情報などを識別するための情報を格納するステータスフラグ（以下、単にフラグとも称する）が設定されるフラグ領域１２ａが設けられる。
【００１４】
上記データメモリ（不揮発性メモリ）１３は、データの書き込みが可能な不揮発性のメモリである。上記データメモリ１３は、例えば、ＥＥＰＲＯＭあるいはフラッシュメモリなどにより構成される。上記データメモリ１３には、当該ＩＣカード１の使用目的に応じ

た種々の情報が記憶される。上記データメモリ１３には、種々の設定情報などを記憶するためのデータテーブルなども設けられる。たとえば、当該ＩＣカードを複数の使用者が使用することが想定される場合、上記データメモリ１３には、各使用者の識別情報（ＩＤ）、パスワード、および、各使用者ごとの各種のデータに対するアクセス権を示す情報などが格納されるデータテーブル１３ａが設けられる。

【００１５】

また、当該ＩＣカードの使用目的に応じたアプリケーション（処理プログラムおよび運用データなど）は、上記データメモリ１３に記憶される。また、当該ＩＣカード１が複数の使用目的に使用される場合、上記データメモリ１３には、各使用目的に応じた複数のアプリケーションが記憶される。なお、当該ＩＣカード１の使用目的に応じたアプリケーションは、上記データメモリ１３上に定義された使用目的ごとのプログラムファイルおよびデータファイルなどの各ファイルに記憶される。このようなファイル構造は、たとえば、ＩＳＯ／ＩＥＣ７８１６－４に基づくものである。つまり、上記ＩＣカード１のデータメモリ１３には、種々のアプリケーションおよび種々の運用データが記憶可能である。

【００１６】

上記乱数生成部１４は、任意のデータとしての乱数を生成するものである。上記乱数生成部１４は、ＩＣチップなどにより構成しても良いし、上記ＣＰＵなどの制御素子がプログラムを実行することにより実現するようにしても良い。

上記通信制御部１５は、上記インターフェース１７を介して外部装置（たとえば、ＩＣカード処理装置２）とのデータ通信を制御するものである。外部装置からデータを受信する場合、上記通信制御部１５は、上記インターフェース１７により受信した電波としての送信データを復調し、復調した信号を上記ＣＰＵ１０に供給する。また、外部装置へデータを送信する場合、上記通信制御部１５は、上記ＣＰＵ１０から与えられるデータを変調し、変調したデータを上記インターフェース１７としてのアンテナにより電波として発信する。なお、接触式ＩＣカードの場合、上記インターフェース１７は、外部装置のコンタクト部と物理的に接触する端子などにより構成されるものとなる。

【００１７】

上記電源部１６は、上記インターフェース１７により受信した電波から当該ＩＣカード１の各部を動作させるための電源およびクロックパルスを生成する。上記電源部１６は、上記アンテナ１７により受信した電波から生成した電源電圧およびクロックパルスを各部に供給するようになっている。また、上記電源部１６からの電源供給により起動した場合、上記ＣＰＵ１０は、当該ＩＣカード１の処理状態をリセットする処理を行うようになっている。なお、接触式ＩＣカードの場合、インターフェース１７を介して外部装置から直接的に供給される電源およびクロックパルスにより各部が動作するようになっている。

【００１８】

次に、上記ＩＣカード処理装置２について説明する。

上記ＩＣカード処理装置２は、図１に示すように、制御装置２１およびカードリーダライタ２２を有している。上記制御装置２１は、パーソナルコンピュータ（ＰＣ）などにより構成される。上記制御装置２１は、ＣＰＵなどの演算処理部、ＲＡＭ、ＲＯＭ、不揮発性メモリおよびハードディスクドライブなどの各種メモリ、通信インターフェースなどの各種インターフェースなどにより構成される。上記制御装置２１では、上記演算処理部がメモリに記憶されている各種の制御プログラムを実行することにより各種の処理を実現している。また、上記制御装置２１は、ＩＣカード１とのデータ通信を行う上記カードリーダライタ２２とのデータの入出力を行うようになっている。

【００１９】

たとえば、上記制御装置２１には、上記ＩＣカード１を用いた各種の処理に応じた制御プログラムが予め記憶されている。上記制御装置２１では、上記のような制御プログラムを実行することにより上記ＩＣカード１を用いた各種の処理を実行する。たとえば、上記ＩＣカード１を用いた各種の処理において、上記制御装置２１は、所定のコマンドを所定の手順で供給する。上記制御装置２１では、上記のような各コマンドに対するＩＣカード

１からの各レスポンス（コマンドに対する処理結果等を示す情報）に基づいて各種の処理を行うようになっている。

【００２０】

上記カードリーダライタ２２は、上記ＩＣカード１とのデータ通信を行う通信手段として機能する。上記カードリーダライタ２２は、上記ＩＣカード１の通信方式に応じた通信方式によるデータ通信を行うためのものである。つまり、上記カードリーダライタ２２を介して制御装置２１は、上記ＩＣカード１とのデータ通信を実現している。

【００２１】

上記ＩＣカード１が非接触型のＩＣカードである場合、上記カードリーダライタ２２は、上記ＩＣカード１との無線によるデータ通信を行うためのアンテナおよび通信制御部（変復調回路等）などにより構成される。非接触型のＩＣカード１へデータを送信する場合、上記カードリーダライタ２２では、上記制御装置２１から与えられる送信データを変調し、変調した信号を電波としてアンテナにより発信する。また、非接触型のＩＣカード１からデータを受信する場合、上記カードリーダライタ２２では、アンテナにより受信した電波としての信号を通信制御部により復調し、復調したデータを受信データとして上記制御装置２１へ供給する。また、上記カードリーダライタ２２では、上記のようなデータの送受信とともに、上記ＩＣカード１を動作させるための電源およびクロックパルスとなる電波をアンテナにより発信するようになっている。

【００２２】

また、上記ＩＣカード１が接触型のＩＣカードである場合、上記カードリーダライタ２２は、ＩＣカード１と物理的に接触してデータ通信を行うためのコンタクト部および通信制御部などにより構成される。接触型のＩＣカードとのデータの送受信を行う場合、上記カードリーダライタ２２では、上記コンタクト部がＩＣカード１側に設けられているコンタクト部と物理的に接触して各種のデータ通信を行う。また、上記カードリーダライタ２２では、ＩＣカード１に物理的に接触しているコンタクト部を介して当該ＩＣカード１に対して電源およびクロックパルスを供給するようになっている。

【００２３】

次に、上記第１の実施の形態に係るＩＣカード１の動作について概略的に説明する。

上記ＩＣカード処理装置２からコマンドを受信した場合、上記ＩＣカード１は、コマンドの種類によっては、当該コマンドを実行するための実行条件のチェック処理を前処理として実行する。また、上記データメモリ１３などに記憶されている各種のデータに対するアクセス権を設定されている場合、当該ＩＣカード１では、外部からのコマンド処理を行う前の処理（前処理）として、各種のデータに対するアクセス権を設定する処理を実行することもある。ここでは、これらのような外部装置からのコマンドに対応する所定の処理を実行する前に実行される処理を前処理と総称するものとする。

【００２４】

上記のような前処理の結果として得られる情報は、上記フラグ領域１２ａに設けられる各フラグに格納され、ＣＰＵ１０により適宜参照される。たとえば、外部装置からのコマンドに対応する処理を実行するための実行条件のチェック結果は、当該コマンドに対応する所定の処理を実行する前に各実行条件に対応づけたフラグに格納される。また、各種のデータに対するアクセス権を示す情報は、各種のデータへのアクセスを要求するコマンドを実行する前に各種のデータに対応づけたフラグに格納される。

【００２５】

また、上記のような各フラグは、処理結果あるいは設定情報などを示す２値情報が格納されるものである。各フラグに格納される２値情報は、当該フラグが示す状態が第１の状態（たとえば、オン状態）であるか第２の状態（たとえば、オフ状態）であるかを示すものである。また、本実施の形態において、各フラグに格納される２値情報は、複数ビット長の第１のデータ、あるいは、第１のデータとは値が異なる複数ビット長の第２のデータの何れかである。このため、各フラグとしては、第１のデータおよび第２のデータを構成する複数ビットのデータが格納可能なデータ領域が設定される。

【００２６】
まず、各フラグに格納される２値情報としての第１のデータおよび第２のデータについて説明する。
上記のように、各フラグには、処理結果あるいは設定内容などに応じて、複数ビット長の第１のデータあるいは複数ビット長の第２のデータが格納される。第１のデータは、当該フラグが第１の状態（オン状態）であることを示し、第２のデータは、当該フラグが第２の状態（オフ状態）であることを示すものである。各フラグに格納される第１のデータおよび第２のデータは、ＩＣカード１内のＣＰＵ１０が識別できるようになっている必要がある。このため、各フラグに格納される第１のデータおよび第２のデータを示す情報は、たとえば、ＣＰＵ１０内のレジスタ１０ａなどの当該ＩＣカード１内のメモリに格納される。
【００２７】
また、上記各フラグに格納される第１のデータおよび第２のデータは、任意の値が設定可能である。ここでは、第１のデータおよび第２のデータは、主に、上記乱数生成部１４により生成される乱数に基づいて決定されるランダムな値であるものとする。ただし、第１のデータおよび第２のデータは、予め設定されている値（固定値）であっても良い。この場合、第１のデータおよび第２のデータとしての固定値は、たとえば、当該ＩＣカード１を利用可能な状態とする発行処理時に設定されるようにすれば良い。また、第１のデータおよび第２のデータは、予め設定されている固定値と上記乱数生成部１４により生成される乱数とに基づいて決定される値であっても良い。この場合、第１のデータおよび第２のデータは、たとえば、予め設定されている固定値と上記乱数生成部１４により生成される乱数との排他的論理和などの演算処理により生成することが可能である。
【００２８】
また、第１のデータおよび第２のデータは、それぞれ関連性のない異なる値を設定するようにしても良いし、何れか一方の値から他方の値を算出できるようにしても良い。前者の場合、第１のデータおよび第２のデータがそれぞれ上記ＩＣカード１内のメモリ（例えば、レジスタ１０ａ）に記憶される。また、後者の場合、たとえば、第１のデータをＩＣカード１内のメモリ（例えば、レジスタ１０ａ）に記憶しておき、その第１のデータの各ビットの値を反転させた反転値を第２のデータとすることにより実現可能である。なお、以下の説明では、主に、第１のデータを基準値としてＣＰＵ１０内のレジスタ１０ａに格納し、第１のデータ（基準値）の反転値を第２のデータとする場合を想定する。
【００２９】
また、第１のデータおよび第２のデータは、任意のタイミングで設定することが可能である。たとえば、ＩＣカード内で発生させる乱数などに基づいて第１のデータおよび第２のデータを決定する場合、第１のデータおよび第２のデータは、当該ＩＣカードのリセット処理を行う際に決定するようにしても良いし、コマンドを受信するごとに決定するようにしても良いし、各フラグを初期化する際に決定するようにしても良い。
なお、以下の説明では、ＩＣカード１は、リセット処理において、上記乱数生成部１４が生成する乱数に基づいて第１のデータとしての基準値Ｒを決定し、第１のデータの反転値を第２のデータとするものとする。また、第１のデータとしての基準値Ｒは、上記内部レジスタ１０ａに格納されるものとする。
図３（ａ）は、基準値Ｒの設定例を示す図である。図３（ａ）に示す例では、基準値Ｒとして、８ビットのデータ「１０１０００１１」がＣＰＵ１０の内部メモリとしてのレジスタ１０ａに格納されている。すなわち、図３（ａ）に示す例では、外部装置からの電力供給を受けてリセットされたＩＣカード１のＣＰＵ１０は、上記乱数生成部１４により乱数を生成させ、生成された乱数に基づく基準値（第１のデータ）Ｒとして、８ビットのデータ「１０１０００１１」をレジスタ１０ａに格納する。これにより、当該ＩＣカード１のＣＰＵ１０は、次に当該ＩＣカード１がリセットされるまで、レジスタ１０ａに格納した基準値Ｒと基準値Ｒの反転値Ｒ´とを識別する。
【００３０】

次に、上記各フラグの設定について説明する。
各フラグは、たとえば、ワーキングメモリ１２などのＩＣカード１内のメモリ上に設けられる。ここでは、図１に示すように、各フラグは、ワーキングメモリ１２上のフラグ領域１２ａに設定されるものとする。
図３（ｂ）、（ｃ）、（ｄ）、（ｅ）は、フラグ領域１２ａに設定された３つのフラグＡ、Ｂ、Ｃの設定例を示す図である。図３（ｂ）は、３つのフラグが初期化された状態を示している。各フラグＡ、Ｂ、Ｃは、第１のデータおよび第２のデータのデータ長に応じたワーキングメモリ１２上の領域が設定される。図３（ｂ）に示す例では、図３（ａ）に示す基準値Ｒ（第１のデータ）および反転値Ｒ´（第２のデータ）が８ビットのデータであるため、各フラグは、８ビットのデータが格納可能な領域として設定されている。なお、図３（ｂ）に示す例では、初期化された状態の各フラグには、初期値として「００００００００」が格納されている。
【００３１】
図３（ｂ）に示すように初期化された各フラグＡ、Ｂ、Ｃには、図３（ｃ）、（ｄ）、（ｅ）に示すように、処理結果あるいは設定情報などに応じて基準値Ｒあるいは反転値Ｒ´が格納される。たとえば、図３（ｃ）に示す例では、フラグＡに基準値Ｒ「１０１０００１１」が格納された状態を示している。また、図３（ｄ）に示す例では、フラグＢに基準値Ｒ「１０１０００１１」が格納された状態を示している。また、図３（ｅ）に示す例では、フラグＣに反転値Ｒ´「０１０１１１００」が格納された状態を示している。
【００３２】
このように、各フラグには、８ビットの第１のデータとしての基準値Ｒ、あるいは、８ビットの第２のデータとしての反転値Ｒ´が格納される。このようなフラグでは、全てのビットの値が反転させられなければ、状態が変化しない。つまり、上記のようなフラグは、外的な要因により何れかのビットの値が反転させられても、フラグが示す状態が変化しない（ビット異常のエラーとなる）。また、電源にノイズを付加したり、電磁波、光波あるいは温度などのストレスを与えたりしても８ビットのデータを全て反転させるのは、困難であると考えられる。したがって、上記のように、フラグに８ビットのデータを格納するようにすることにより、当該ＩＣカードのセキュリティ性を向上させることができる。
【００３３】
さらに、上記のように、ＩＣカード１内で発生させる乱数に基づいて基準値Ｒを決定するようにすることにより、基準値Ｒを推定されにくくすることが可能である。なお、第１のデータとしての基準値Ｒだけでなく、第２のデータも乱数に基づく値とするようにしても良い。この場合、上記ＩＣカード１では、２つの乱数を発生させ、各乱数に基づいて第１のデータと第２のデータとを設定するようにすれば良い。
【００３４】
次に、上記第１の実施の形態に係る第１の処理例について説明する。
この第１の処理例は、コマンドに対応する処理を実行するための種々の実行条件が設定されている場合の処理例である。すなわち、第１の処理例において、上記ＩＣカード１では、外部装置から与えられたコマンドに対応する処理を実行するための種々の実行条件を順次チェックし、それらのチェック結果を順次各フラグに格納する（前処理）。この場合、各実行条件について、実行条件を満たしていれば、対応するフラグに基準値Ｒ（第１のデータ）が格納され、実行条件を満たしていなければ、対応するフラグに反転値Ｒ´（第２のデータ）が格納されるようになっている。なお、第１の実施の形態として説明する第１の処理例および第２の処理例は、上記ＣＰＵ１０がプログラムメモリ１１あるいはデータメモリ１３に記憶されているプログラムを実行することにより実現されるものとする。
【００３５】
図４は、第１の処理例を説明するためのフローチャートである。
【００３６】
上記ＩＣカード１は、上記ＩＣカード処理装置２からの電力の供給を受けて起動するようになっている。上記ＩＣカード処理装置２から電力の供給を受けて起動すると、上記Ｉ

ＣカードｌのＣＰＵｌ０は、リセット処理を行う（ステップＳｌｌ）。上記リセット処理が完了すると、上記ＩＣカードｌのＣＰＵｌ０は、上記乱数生成部ｌ４により乱数を生成する（ステップＳｌ２）。上記乱数生成部ｌ４により乱数が生成されると、上記ＩＣカードｌのＣＰＵｌ０は、生成された乱数に基づいて所定の桁数（ビット数）の基準値（第ｌのデータ）Ｒを決定し、決定した基準値Ｒを内部レジスタｌ０ａに格納する（ステップＳｌ３）。たとえば、図３（ａ）に示す例では、乱数生成部ｌ４により発生させた乱数に基づく基準値Ｒとして「ｌ０ｌ０００ｌｌ」が決定され、その基準値Ｒが内部レジスタｌ０ａに格納される

内部レジスタｌ０ａに基準値Ｒを格納すると、上記ＣＰＵｌ０は、ＩＣカード処理装置２からのコマンドに対する処理が実行可能な状態となる。この状態においてＩＣカード処理装置２からコマンドを受信すると、上記ＣＰＵｌ０は、当該コマンドの実行準備として、当該コマンドを実行するための実行条件に対応づけたフラグの初期化を行う（ステップＳｌ４）。ここで、上記ＩＣカード処理装置２から受信したコマンドには、３つの実行条件が設定されているものとする。このような場合、上記ＣＰＵｌ０は、たとえば、図３（ｂ）に示すように、当該コマンドに対応する処理の３つの実行条件に対応する３つのフラグＡ、Ｂ、Ｃを初期化する。

【００３７】

このようなコマンドの実行準備としての各フラグＡ、Ｂ、Ｃの初期化が完了すると、上記ＣＰＵｌ０は、当該コマンドに対応する処理を実行するための各実行条件（第ｌ、第２、第３の実行条件）をチェックする処理（前処理手段）を行う。すなわち、各フラグＡ、Ｂ、Ｃの初期化が完了すると、上記ＣＰＵｌ０は、第ｌの実行条件をチェックする（ステップＳｌ５）。この結果として第ｌの実行条件を満たすと判定した場合、上記ＣＰＵｌ０は、図３（ｃ）に示すように、フラグＡに上記内部レジスタｌ０ａに格納している基準値Ｒを格納する（ステップＳｌ６）。なお、第ｌの実行条件を満たしていないと判定した場合、上記ＣＰＵｌ０は、内部レジスタに格納されている基準値Ｒの各ビットを反転させた値（以下、単に反転値と称する）（第２のデータ）Ｒ´をフラグＡにセットする（ステップＳｌ６）。

【００３８】

上記第ｌの実行条件のチェックが完了すると、上記ＣＰＵｌ０は、第２の実行条件をチェックする（ステップＳｌ７）。上記第２の実行条件を満たすと判定した場合、上記ＣＰＵｌ０は、図３（ｄ）に示すように、内部レジスタｌ０ａに格納している基準値Ｒをフラグ Ｂに格納する（ステップＳｌ８）。また、上記第２の実行条件を満たさないと判定した場合、上記ＣＰＵｌ０は、上記反転値Ｒ´をフラグＢにセットする（ステップＳｌ８）。

【００３９】

同様に、上記第２の実行条件のチェックが完了すると、上記ＣＰＵｌ０は、第３の実行条件をチェックする（ステップＳｌ９）。上記第３の実行条件を満たすと判定した場合、上記ＣＰＵｌ０は、フラグＢに内部レジスタに格納した基準値Ｒを格納する（ステップＳ２０）。また、上記第３の実行条件を満たさないと判定した場合、上記ＣＰＵｌ０は、図３（ｅ）に示すように、上記反転値Ｒ´をフラグＣにセットする（ステップＳ２０）。

【００４０】

全ての実行条件のチェックが完了すると、上記ＣＰＵｌ０は、各フラグＡ、Ｂ、Ｃの値が基準値Ｒまたは反転値Ｒ´であるか否かを判定する（ステップＳ２ｌ）。これは、各フラグＡ、Ｂ、Ｃに格納されている値が正常な値であるか否かを判定する処理である。上記のような処理が正常に実行され、かつ、外的な要因などで値が変化させられていなければ、各フラグＡ、Ｂ、Ｃには、基準値Ｒまたは反転値Ｒ´が格納されているはずである。したがって、何れかのフラグＡ、Ｂ、Ｃに基準値Ｒまたは反転値Ｒ´以外の値が格納されている場合、何かの異常がＩＣカード内で発生したものと判断できる。すなわち、外的な要因によりフラグＡ、Ｂ、Ｃの値が変更された場合、フラグＡ、Ｂ、Ｃの値は、基準値Ｒまたは反転値Ｒ´以外に値になることが予測される。言い換えると、各フラグの値が基準値Ｒまたは反転値Ｒ´であれば、各実行条件のチェックが正常に実行されたものと判定でき

る。

【００４１】

上記判定により基準値Ｒまたは反転値Ｒ´以外の値がフラグに格納されていると判定した場合（ステップＳ２１、ＮＯ）、上記ＣＰＵ１０は、フラグ内のビット異常が発生したものと判定し、動作を停止するなどのエラー処理を行う（ステップＳ２２）。この場合、外的な要因により異常となったことが考えられる。このため、上記ＣＰＵ１０は、セキュリティ確保のために、エラー処理として、強制的に動作を停止するようにする。

また、上記判定により各フラグが基準値Ｒまたは反転値Ｒ´であると判定した場合（ステップＳ２１、ＹＥＳ）、上記ＣＰＵ１０は、各フラグの値が全て基準値Ｒであるか否かを判定する（ステップＳ２３）。これは、コマンドに応じた処理の全ての実行条件を満たしているか否かを判定する処理である。

【００４２】

すなわち、上記判定により何れかのフラグが基準値Ｒでないと判定した場合（ステップＳ２３、ＮＯ）、上記ＣＰＵ１０は、ＩＣカード処理装置２に実行条件が満たされていない旨のレスポンスを出力するなどのエラー処理を行う（ステップＳ２４）。この場合、当該ＩＣカード１内の動作異常ではないと考えられる。このため、上記ＣＰＵ１０は、通常のエラー処理を行う。

また、上記判定により各フラグＡ、Ｂ、Ｃの値が全て基準値Ｒであると判定した場合（ステップＳ２３、ＹＥＳ）、上記ＣＰＵ１０は、ＩＣカード処理装置２から受信したコマンドに応じた本処理を実行する（ステップＳ２５）。

【００４３】

上記のような第１の処理例では、コマンドを実行するための種々の実行条件に対応づけた複数ビットからなるフラグを設定し、各フラグには乱数に基づいて決定した複数ビットの基準値あるいは上記基準値に対する反転値の何れかを各実行条件のチェック結果としてセットする。これにより、各フラグに記憶されている情報に基づいて、実行の可否を判定できる。また、各フラグが複数ビットからなっているため、外的なエネルギーでフラグとして利用するメモリ領域の一部のビットの値を反転されても、フラグ全体が示す状態が反転することがなく、ビット異常としてエラー処理することが可能となる。

【００４４】

次に、上記第１の実施の形態としての第２の処理例について説明する。

この第２の処理例では、設定情報をフラグに格納し、フラグに格納されている情報に基づいて各種の処理を実行する場合の処理例である。ここでは、各種のデータのアクセス権を示す情報をフラグに格納する処理（前処理）を実行し、各フラグに格納しているアクセス権に従ってデータの出力処理を行う処理例について説明する。また、アクセス可能なデータに対応するフラグには基準値Ｒ（第１のデータ）が格納され、アクセスが禁止されているデータに対応するフラグには反転値Ｒ´（第２のデータ）が格納されるものとする。

【００４５】

次に、上記ＩＣカード１における複数の使用者に対する各種のデータへのアクセス権の設定例について説明する。ここでは、当該ＩＣカード１には、複数の使用者ごとに各種のデータに対するアクセス権が設定されているものとする。

図５は、複数の使用者ごとの各種のデータに対するアクセス権を示す情報を記憶するデータテーブル１３ａの例を示す図である。図５に示す例では、各使用者ごとに、各使用者の識別情報としてのＩＤ、各使用者のパスワード、各データＸ、Ｙ、Ｚに対するアクセスの可否を示す情報が格納されている。上記ＩＤは、各使用者を識別するための識別情報であり、上記パスワードは、各使用者に与えられている認証情報である。これにより、上記ＣＰＵ１０は、上記データテーブル１３ａを参照して、上記ＩＤおよび上記パスワードにより使用者が特定（認証）できる。また、上記のようなデータテーブル１３ａを参照して、上記ＣＰＵ１０は、認証が成功した使用者に対する各データＸ、Ｙ、Ｚへのアクセス権が判別できるようになっている。

【００４６】

また、図６は、フラグ領域１２ａに設けたフラグによるアクセス権の設定例を示す図である。図６（ａ）は、図３（ａ）と同様に、基準値Ｒの設定例を示す図である。

【００４７】

図６（ｂ）、（ｃ）は、フラグ領域１２ａに設定された３つのフラグＸ、Ｙ、Ｚの設定例を示す図である。図６（ｂ）は、３つのデータＸＹＺに対応づけた３つのフラグＸ、Ｙ、Ｚが初期化された状態を示している。ここで、各フラグＸ、Ｙ、Ｚは、それぞれデータＸ、Ｙ、Ｚに対するアクセス権を示すものとする。各フラグＸ、Ｙ、Ｚは、基準値（第１のデータ）および反転値（第２のデータ）のデータ長に応じてワーキングメモリ１２上のフラグ領域１２ａに設定される。図６（ｂ）に示す例では、図６（ａ）に示す基準値Ｒ（第１のデータ）および反転値Ｒ´（第２のデータ）が８ビットのデータであるため、各フラグＸ、Ｙ、Ｚは、８ビットのデータが格納可能な領域として設定されている。なお、図６（ｂ）に示す例では、初期化された状態の各フラグには、初期値として「００００００００」が格納されている。

【００４８】

図６（ｂ）に示すように初期化された各フラグＡ、Ｂ、Ｃには、図３（ｃ）に示すように、各データへのアクセス権に応じて基準値Ｒあるいは反転値Ｒ´が格納される。たとえば、図６（ｃ）に示す例では、図５に示す使用者ＩＤ「１０００２」の使用者に対する各データへのアクセス権を反映したものとなっている。すなわち、図５に示す例では、ＩＤ「１０００２」の使用者は、データＸへのアクセスが許可され、データＹ、Ｚへのアクセスが禁止されている。このため、図６（ｃ）に示す例では、フラグＸに基準値Ｒ「１０１００００１」が格納され、フラグＹおよびフラグＺに反転値Ｒ´「０１０１１１００」が格納されている。つまり、図６（ｃ）に示す例では、基準値Ｒがセットされたフラグ Ｘに対応するデータＸへのアクセスが許可され、反転値Ｒ´がセットされたフラグＹ、Ｚに対応するデータＹ、Ｚへのアクセスが禁止されていることを示している。

【００４９】

次に、上記第２の処理例の流れについて説明する。

図７は、第２の処理例の流れを説明するためのフローチャートである。

すなわち、上記ＩＣカード処理装置２から電力供給を受けて起動すると、上記ＩＣカード１のＣＰＵ１０は、リセット処理を行う（ステップＳ３１）。上記リセット処理が完了すると、上記ＩＣカード１のＣＰＵ１０は、上記乱数生成部１４により乱数を生成する（ステップＳ３２）。上記乱数生成部１４により乱数が生成されると、上記ＩＣカード１のＣＰＵ１０は、生成された乱数に基づいて所定の桁数（ビット数）の基準値（第１のデータ）Ｒを決定し、決定した基準値Ｒを内部レジスタ１０ａに格納する（ステップＳ３３）。

【００５０】

上記内部レジスタ１０ａに基準値Ｒを格納すると、上記ＣＰＵ１０は、各コマンドの実行準備として、各種のデータへのアクセス権を示すフラグの初期化を行う（ステップＳ３４）。ここでは、例として、使用者ごとに設定されている３つのデータＸ、Ｙ、Ｚに対するアクセス権をフラグＸ、Ｙ、Ｚで示すものとする。このような場合、上記ＣＰＵ１０は、３つのデータＸ、Ｙ、Ｚへのアクセス権を示す３つのフラグＸ、Ｙ、Ｚを初期化する。

【００５１】

各フラグＸ、Ｙ、Ｚの初期化が完了すると、上記ＣＰＵ１０は、まず、使用者を特定するための使用者の認証処理を行う（ステップＳ３５）。ここでは、使用者が指定するＩＤおよびパスワードにより使用者認証を行うものとする。たとえば、ＩＤおよびパスワードは、使用者がＩＣカード処理装置２に接続された操作部により入力し、認証コマンドとともにＩＣカードへ供給されるものとする。ＩＤおよびパスワードを取得すると、当該ＩＣカード１のＣＰＵ１０は、取得したＩＤおよびパスワードと上記データテーブル１３ａに記憶されている各使用者のＩＤおよびパスワードと照合する。このような照合処理により使用者が特定される。なお、使用者が特定できない場合、つまり、使用者認証が失敗した場合（ステップＳ３６、ＮＯ）、上記ＣＰＵ１０は、使用者認証の失敗に伴うエラー処理

を実行する（ステップＳ３７）。

【００５２】

上記使用者認証により使用者が特定されると（ステップＳ３６、ＹＥＳ）、上記ＣＰＵ１０は、アクセス権の設定処理（前処理手段）として、当該使用者の各データに対するアクセス権を上記データテーブル１３ａにより照会し（ステップＳ３８）、上記データテーブル１３ａにおける設定内容に従って各フラグに基準値Ｒまたは反転値Ｒ´をセットする（ステップＳ３９）。つまり、各データへのアクセス権を示す各フラグには、当該使用者のアクセス権に応じた値がセットされる。たとえば、アクセス可能なデータに対応するフラグには、基準値Ｒがセットされ、アクセスが禁止されているデータに対応するフラグには、反転値Ｒ´がセットされる。

【００５３】

上記のような各フラグにアクセス権を示す値をセットした状態において、上記ＣＰＵ１０は、データのアクセスを要求するコマンドを受信した場合（ステップＳ４０、ＹＥＳ）、各データへのアクセスをフラグの状態に応じて制御する。すなわち、データへのアクセス要求を受けた場合、上記ＣＰＵ１０は、まず、各フラグＸ、Ｙ、Ｚが正常な値であるか否かを判定する（ステップＳ４２）。これは、上記ＣＰＵ１０は、各フラグＸ、Ｙ、Ｚの値が基準値Ｒまたは反転値Ｒ´であるか否かを判定する処理である。すなわち、外的な要因などでフラグの値が変化させられていなければ、各フラグＸ、Ｙ、Ｚには、基準値Ｒまたは反転値Ｒ´が格納されているはずである。したがって、何れかのフラグＸ、Ｙ、Ｚに基準値Ｒまたは反転値Ｒ´以外の値が格納されている場合、何かの異常がＩＣカード内で発生したものと判断できる。

【００５４】

上記判定により基準値Ｒまたは反転値Ｒ´以外の値がフラグに格納されていると判定した場合（ステップＳ４２、ＮＯ）、上記ＣＰＵ１０は、フラグ内のビット異常が発生したものと判定し、エラー処理を行う（ステップＳ４３）。この場合、外的な要因によりフラグ内のビット異常が発生した可能性が考えられる。このため、上記ＣＰＵ１０は、たとえば、セキュリティ確保のために、エラー処理として、強制的に動作を停止するようにする。また、上記ＣＰＵ１０は、エラー処理として、データへのアクセスを行わずに、ビット異常を示す応答を送信する処理のみを行うようにしても良い。

【００５５】

また、上記判定により各フラグが基準値Ｒまたは反転値Ｒ´であると判定した場合（ステップＳ４２、ＹＥＳ）、上記ＣＰＵ１０は、アクセスが要求されているデータがアクセス可能なデータであるか否かを判定する（ステップＳ４４）。これは、アクセスが要求されているデータに対応するフラグが基準値Ｒ（アクセス可能な状態であることを示す第１のデータ）であるか否かを判定する処理である。

【００５６】

上記判定によりアクセスが要求されたデータがアクセス可能なデータでないと判定した場合、つまり、当該データに対応するフラグの値が反転値Ｒ´である場合（ステップＳ４４、ＮＯ）、上記ＣＰＵ１０は、アクセスが要求されたデータに対するアクセス権がない旨のレスポンスを出力するなどのエラー処理を行う（ステップＳ４４）。この場合、当該ＩＣカード１内の動作異常ではないと考えられる。このため、上記ＣＰＵ１０は、通常のエラー処理を行う。

また、上記判定によりアクセスが要求されたデータがアクセス可能なデータであると判定した場合、つまり、当該データに対応するフラグの値が基準値Ｒである場合（ステップＳ４４、ＹＥＳ）、上記ＣＰＵ１０は、ＩＣカード処理装置２から受信したコマンドに応じた本処理を実行する（ステップＳ４６）。

【００５７】

上記のような第２の処理例では、アクセス権が設定されている各種のデータに対応づけて複数ビットからなるフラグを設定し、上記各フラグには使用者認証により特定された使用者のアクセス権を示す情報として、乱数に基づいて決定した複数ビットの基準値あるい

は上記基準値に対する反転値の何れかをセットする。これにより、各フラグに記憶されている情報に基づいて、各データへのアクセス権を判定できる。また、各フラグが複数ビットからなっているため、外的な要因でフラグとして利用するメモリ領域の一部のビットの値を反転されても、フラグ全体が示す状態が反転することがなく、ビット異常としてエラー処理することが可能となる。

【００５８】

次に、第２の実施の形態について説明する。

本第２の実施の形態では、上記ＣＰＵ内のハードウエア構成によって、上記第１の実施の形態で説明したような処理におけるフラグによる制御を実現するものである。すなわち、第２の実施の形態は、たとえば、上記第１の実施の形態で説明したステップＳ２１およびＳ２３の処理、あるいは、ステップＳ１２およびＳ１３の処理などをＣＰＵ内のハードウエア構成によって実現するものである。

【００５９】

図８は、第２の実施の形態に係る携帯可能電子装置としてのＩＣカード１０１およびＩＣカード１０１を含むＩＣカードシステムの構成例を示すブロック図である。

図８に示すＩＣカード１０１およびＩＣカードシステムは、図１に示すＩＣカード１およびＩＣカードシステムとほぼ同様な機能を有している。このため、図８に示すＩＣカード１０１およびＩＣカードシステムにおいて、図１に示すＩＣカード１と同様な構成については、同一箇所に同一符号を付して詳細な説明を省略するものとする。なお、上記ＩＣカード１０１は、ＩＣチップ１０１ａを有するモジュール１０１ｂが埋設された本体１０１ｃにより構成されている。

【００６０】

図８に示すＩＣカード１０１と図１に示すＩＣカード１とでは、主に、ＣＰＵ１１０内の構成が異なっている。つまり、第２の実施の形態に係るＩＣカード１０１では、上記第１の実施の形態で説明したようなフラグによる制御（たとえば、図４のステップＳ２１、Ｓ２３の処理、あるいは、図７のステップＳ４２、Ｓ４４の処理）を実現するための構成を主としてＣＰＵ１１０内のハードウエア構成によって実現している。

【００６１】

図８に示すＣＰＵ１１０は、レジスタ１１０ａおよびロジック回路１１０ｂなどを有している。上記レジスタ１１０ａは、各種のステータスフラグ、あるいは、各ステータスフラグにセットすべき基準値などが記憶されるようになっている。なお、上記基準値は、上記第１の実施の形態で説明したような複数ビット長のデータ（第１のデータ）である。また、本第２の実施の形態では、上記第１の実施の形態と同様に、上記基準値の各ビットを反転させた反転値を第２のデータとするものとする。ただし、第２のデータを反転値以外の値とする場合、第２のデータも上記レジスタ１１０ａに格納される。

【００６２】

上記ロジック回路１１０ｂは、上記レジスタ１１０ａに設けられた各種のステータスフラグにセットされているデータを処理するためのハードウエアにより構成される。上記ロジック回路１１０ｂは、たとえば、各ステータスフラグの値と基準値および反転値を比較する比較回路などを有している。なお、上記レジスタ１１０ａおよび上記ロジック回路１１０ｂについては、後で詳細に説明するものとする。

【００６３】

次に、上記ＣＰＵ１１０内の第１の構成例について説明する。

【００６４】

図９は、上記ＣＰＵ１１０内の第１の構成例を示す図である。

図９に示す構成例では、上記ＣＰＵ１１０内のレジスタ１１０ａに設定データ領域１３１とステータスフラグ領域１３２とが設けられている。また、図９に示す構成例では、各種のステータスフラグとして、ゼロフラグ１３２ａ、キャリーフラグ１３２ｂ、×××フラグ１３２ｃ、…が上記ステータスフラグ領域１３２に設けられている。また、図９に示す構成例では、上記設定データ領域１３１には、予め決定されている１つの基準値Ｒが記

憶されているものとする。従って、各ステータスフラグ１３２ａ、１３２ｂ、１３２ｃ、…には、上記基準値Ｒあるいはその反転値Ｒ´の何れかの値が格納されるようになっている。

【００６５】

上記ＣＰＵ１１０内のロジック回路１１０ｂには、各ステータスフラグ１３２ａ、１３２ｂ、１３２ｃ、…に対応する複数の比較回路１４１ａ、１４１ｂ、１４１ｃ、…が設けられている。上記比較回路１４１ａ、１４１ｂ、１４１ｃ、…は、各ステータスフラグの値に対して、基準値Ｒおよび反転値Ｒ´を比較し、その比較結果を出力する。

【００６６】

たとえば、上記比較回路１４１ａ（１４１ｂ、１４１ｃ、…）は、ステータスフラグ１３２ａ（１３２ｂ、１３２ｃ、…）の値が基準値Ｒと一致する場合、ステータスフラグ１３２ａ（１３２ｂ、１３２ｃ、…）が第１の状態（オンの状態）であることを示すオン信号を出力する。また、上記比較回路１４１ａ（１４１ｂ、１４１ｃ、…）は、ステータスフラグ１３２ａ（１３２ｂ、１３２ｃ、…）の値と反転値Ｒ´とが一致する場合、ステータスフラグ１３２ａ（１３２ｂ、１３２ｃ、…）が第２の状態（オフの状態）であることを示すオフ信号を出力する。また、上記比較回路１４１ａ（１４１ｂ、１４１ｃ、…）は、ステータスフラグ１３２ａ（１３２ｂ、１３２ｃ、…）の値が基準値Ｒ及び反転値Ｒ´の何れとも一致しない場合、ステータスフラグ１３２ａ（１３２ｂ、１３２ｃ、…）がエラー状態であることを示すエラー信号を出力するようになっている。

【００６７】

また、上記エラー信号は、各ステータスフラグにおけるビット異常が検出された場合に出力される信号である。このため、上記エラー信号は、各ステータスフラグにおけるビット異常に伴う動作を指示する信号であっても良い。たとえば、フラグのビット異常が検出された場合にＣＰＵ１１０の動作を強制的に停止させる形態では、エラー信号として、ＣＰＵ１１０の動作を停止させるための信号（停止信号）を出力する。これにより、ステータスフラグにおけるビット異常が検出された場合、強制的にＣＰＵ１１０の動作を停止させることができる。この結果として、外的な要因による不正な動作を防止できる。

【００６８】

また、ステータスフラグにおけるビット異常が検出された場合にＣＰＵ１１０が優先的（強制的）にステータスフラグにおけるビット異常に伴う所定の処理（エラー処理）を実行する形態では、エラー信号として、ＣＰＵ１１０の演算部にステータスフラグにおけるビット異常に伴う所定のエラー処理を優先的に実行させるための割り込み信号を出力する。これにより、上記ＣＰＵ１１０は、ステータスフラグにおけるビット異常が検出された場合に、優先的にＣＰＵ１１０がビット異常の伴う所定のエラー処理を実行するようにできる。この結果として、外的な要因による不正な動作を防止できる。

【００６９】

上記のように、第１の構成例のＣＰＵ１１０には、各ステータスフラグ１３２ａ、１３２ｂ、１３２ｃ、…に対応づけた比較回路１４１ａ、１４１ｂ、１４１ｃ、…が設けられ、各比較回路１４１ａ、１４１ｂ、１４１ｃ、…が各ステータスフラグ１３２ａ、１３２ｂ、１３２ｃ、…の状態（オン、オフ、エラーの何れかであるか）を示す信号を出力する。このような処理は、上記第１の実施の形態で説明した図４のステップＳ２１およびＳ２３の処理あるいは図７のステップＳ４２およびＳ４４の処理に相当する処理である。言い換えると、第１の構成例では、ＣＰＵ１１０内のハードウエア構成により、複数ビットからなる基準値が格納されるフラグのチェック処理を実現している。これにより、第１の構成例では、ステータスフラグを参照すべき処理において、複数ビット化された各ステータスフラグのチェック処理などを高速化することができる。また、上記第１の構成例では、ＣＰＵ内のハードウエア構成によってステータスフラグの状態（オン、オフ、エラー）を機械的に出力することができるため、ＣＰＵ内の演算部（図示しない）によるソフトウエア処理に負荷をかけることなく、ステータスフラグによる制御を効率的に実現できる。

【００７０】

次に、第２の実施の形態に係る上記ＣＰＵ１１０内の第２の構成例について説明する。

【００７１】

図１０は、上記ＣＰＵ１１０内の第２の構成例を示す図である。

図１０に示す構成例では、上記ＣＰＵ１１０内のレジスタ１１０ａの設定データ領域１３１には、各種のステータスフラグに対する複数の基準値を格納するための複数の基準値格納領域１３１ａ、１３１ｂ、１３１ｃ、…が設けられている。また、上記ＣＰＵ１１０内のレジスタ１１０ａのステータスフラグ領域１３２には、図９に示す第１の構成例と同様に、各種のステータスフラグとして、ゼロフラグ１３２ａ、キャリーフラグ１３２ｂ、×××フラグ１３２ｃ、…が設けられている。つまり、図１０に示す第２の構成例では、各ステータスフラグごとに基準値が設定されている。

【００７２】

たとえば、図１０に示す構成例では、各基準値格納領域１３１ａ、１３１ｂ、１３１ｃ、…は、それぞれゼロフラグ１３２ａ、キャリーフラグ１３２ｂ、×××フラグ１３２ｃ、…に対応づけられ、ゼロフラグ１３２ａに基準値としてセットすべき値、キャリーフラグ１３２ｂに基準値としてセットすべき値、×××フラグ１３２ｃに基準値としてセットすべき値が格納されている。つまり、各ステータスフラグ１３２ａ、１３２ｂ、１３２ｃには、それぞれ対応する基準値格納領域１３１ａ、１３１ｂ、１３１ｃに格納されている基準値Ｒａ、Ｒｂ、Ｒｃあるいはその反転値Ｒａ´、Ｒｂ´、Ｒｃ´の何れかの値が格納されるようになっている。

【００７３】

すなわち、図９に示す第１の構成例では、各ステータスフラグに格納すべき基準値を共通の値としたのに対して、図１０に示す第２の構成例では、各ステータスフラグごとに基準値を設定するようにしたものである。なお、図１０に示す構成例において、一部の基準値を共通化するようにしても良い。上記のような第２の構成例によれば、特定のステータスフラグに対応する基準値が第３者に推定された場合であっても、全てのステータスフラグに対する基準値が推定されることがなく、安全性を高めることが可能となる。

【００７４】

また、図１０に示す第２の構成例において、上記ＣＰＵ１１０内のロジック回路１１０ｂには、図９に示す第１の構成例と同様に、各ステータスフラグ１３２ａ、１３２ｂ、１３２ｃ、…に対応する複数の比較回路１４１ａ、１４１ｂ、１４１ｃ、…が設けられている。上記比較回路１４１ａ、１４１ｂ、１４１ｃ、…は、各ステータスフラグの値に対して、対応する各基準値Ｒａ、Ｒｂ、Ｒｃおよび各基準値Ｒａ、Ｒｂ、Ｒｃの反転値Ｒａ´、Ｒｂ´、Ｒｃ´を比較し、その比較結果を出力する。

【００７５】

たとえば、上記比較回路１４１ａ（１４１ｂ、１４１ｃ、…）は、ステータスフラグ１３２ａ（１３２ｂ、１３２ｃ、…）の値が基準値格納領域１３１ａ（１３１ｂ、１３１ｃ、…）に格納されている基準値Ｒａ（Ｒｂ、Ｒｃ、…）と一致する場合、ステータスフラグ１３２ａ（１３２ｂ、１３２ｃ、…）が第１の状態（オンの状態）であることを示すオン信号を出力する。また、上記比較回路１４１ａ（１４１ｂ、１４１ｃ、…）は、ステータスフラグ１３２ａ（１３２ｂ、１３２ｃ、…）の値と基準値格納領域１３１ａ（１３１ｂ、１３１ｃ、…）に格納されている基準値の反転値Ｒａ´（Ｒｂ´、Ｒｃ´、…）とが一致する場合、ステータスフラグ１３２ａ（１３２ｂ、１３２ｃ、…）が第２の状態（オフの状態）であることを示すオフ信号を出力する。また、上記比較回路１４１ａ（１４１ｂ、１４１ｃ、…）は、ステータスフラグ１３２ａ（１３２ｂ、１３２ｃ、…）の値が基準値Ｒａ（Ｒｂ、Ｒｃ、…）及び反転値Ｒａ´（Ｒｂ´、Ｒｃ´、…）の何れとも一致しない場合、ステータスフラグ１３２ａ（１３２ｂ、１３２ｃ、…）がエラー状態であることを示すエラー信号を出力するようになっている。

【００７６】

なお、ステータスフラグにおけるビット異常が検出された場合のエラー信号としては、上述した第１の構成例と同様に、ＣＰＵ１１０を強制的に停止させるための停止信号を出

力するようにしても良いし、ステータスフラグにおけるビット異常に伴う所定のエラー処理を優先的に実行させるための割り込み信号を出力するようにしても良い。これにより、上記ＣＰＵ１１０は、ステータスフラグにおけるビット異常が検出された場合に、強制的（優先的）に停止したり、ビット異常の伴う所定のエラー処理を実行したりすることができる。この結果として、外的な要因による不正な動作を防止できる。

【００７７】

上記のように、第２の構成例のＣＰＵ１１０には、各ステータスフラグ１３２ａ、１３２ｂ、１３２ｃ、…に対応づけた基準値格納領域１３１ａ、１３ｂ、１３１ｃをレジスタに設け、さらに、各ステータススラグと基準値とに対応づけた比較回路１４１ａ、１４１ｂ、１４１ｃ、…をロジック回路に設けられている。上記ＣＰＵ１１０では、各比較回路１４１ａ、１４１ｂ、１４１ｃ、…がそれぞれ基準値格納領域１３１ａ、１３ｂ、１３１ｃの値と各ステータスフラグ１３２ａ、１３２ｂ、１３２ｃ、…の値とを比較することにより、各ステータスフラグ１３２ａ、１３２ｂ、１３２ｃ、…の状態（オン、オフ、エラーの何れかであるか）を示す信号を出力する。上記のような処理は、上記第１の実施の形態で説明した図４のステップＳ２１およびＳ２３の処理あるいは図７のステップＳ４２およびＳ４４の処理に適用可能である。

【００７８】

言い換えると、上記第２の構成例では、ＣＰＵ内のハードウエア構成により各フラグごとに設定されている複数ビットからなる基準値に基づくフラグの制御を実現している。これにより、上記第２の構成例では、複数ビット化された基準値が各ステータスフラグごとに設定されている場合であっても、ソフトウエア制御による負荷をかけることなく、各ステータスフラグのチェック処理などを高速に実行することができる。

【００７９】

次に、上記ＣＰＵ１１０内の第３の構成例について説明する。

【００８０】

図１１は、上記ＣＰＵ１１０内の第３の構成例を示す図である。

図１１に示す第３の構成例では、図９に示す第１の構成例に加えて、上記ＣＰＵ１１０内のロジック回路１１０ｂに乱数生成回路１４２が追加されている。上記乱数生成回路１４２は、リセット信号を受けて乱数を生成し、生成した乱数を基準値Ｒとしてレジスタ１１０ａ内の設定データ領域１３１に格納する。上記設定データ領域１３１に基準値Ｒが設定された状態では、上述した図９に示す第１の構成例と同様な動作を行う。

【００８１】

すなわち、図１１に示す第３の構成例では、当該ＩＣカード１０１がリセットされるごとに、ロジック回路１１０ｂ内の乱数生成回路１４２が乱数を生成し、生成した乱数に基づく基準値Ｒが設定データ領域１３１に格納されるようになっている。なお、乱数生成回路１４２が乱数を生成し、生成した乱数に基づく基準値を設定データ領域１３１に格納する処理は、リセット時に実行することに限定されるものではなく、たとえば、コマンドを受信するごとなどの任意のタイミングで実現することも可能である。なお、第３の構成例による乱数に基づく基準値の設定処理は、たとえば、第１の実施の形態で説明した図４のステップＳ１１～Ｓ１３の処理あるいは図７のステップＳ３１～Ｓ３３の処理に適用可能である。

【００８２】

上記のように、第３の構成例では、設定データ領域１３１に格納される基準値がリセットされるごとに任意のデータに変更されるため、基準値が第３者に特定されるリスクを軽減し、第３者が種々の解析手法で基準値を推定したとしても、その基準値をリセット時に無効とすることが可能となる。この結果として、上記第３の構成例によれば、ＩＣカードのセキュリティ性を向上させることができる。さらに、上記第３の構成例では、乱数により基準値を設定する処理、および、ステータスフラグのチェック処理などをハードウエアにより実行するため、ＩＣカードにおける処理の高速化および効率化が期待できる。

【００８３】

次に、上記ＣＰＵ１１０内の第４の構成例について説明する。
図１２は、上記ＣＰＵ１１０内の第４の構成例を示す図である。
図１２に示す第４の構成例では、図１０に示す第２の構成例に加えて、上記ＣＰＵ１１０内のロジック回路１１０ｂに乱数生成回路１４２が追加されている。上記乱数生成回路１４２は、リセット信号を受けて各基準値格納領域１３１ａ、１３１ｂ、１３１ｃ、…ごとに乱数を生成し、生成した各乱数を基準値Ｒａ、Ｒｂ、Ｒｃとしてレジスタ１１０ａ内の各基準値格納領域１３１ａ、１３１ｂ、１３１ｃ、…に格納する。各基準値格納領域１３１ａ、１３１ｂ、１３１ｃ、…に基準値Ｒａ、Ｒｂ、Ｒｃが設定された状態において、上記第４の構成例では、上述した第２の構成例と同様な動作を行う。

【００８４】

すなわち、図１２に示す第４の構成例では、当該ＩＣカード１０１がリセットされるごとに、ロジック回路１１０ｂ内の乱数生成回路１４２が各基準値となる乱数をそれぞれ生成し、生成した乱数に基づく各基準値Ｒａ、Ｒｂ、Ｒｃ、…が各基準値格納領域１３１ａ、１３１ｂ、１３１ｃ、…に格納されるようになっている。なお、乱数生成回路１４２が各基準値用の乱数を生成し、生成した乱数に基づいて各基準値を各基準値格納領域に格納する処理は、リセット時に実行することに限定されるものではなく、たとえば、コマンドを受信するごとなどの任意のタイミングで実現することも可能である。また、第４の構成例による乱数に基づく基準値の設定処理は、たとえば、第１の実施の形態で説明した図４のステップＳ１１～Ｓ１３の処理あるいは図７のステップＳ３１～Ｓ３３の処理に適用可能である。

【００８５】

上記のように、第４の構成例では、各基準値格納領域１３１ａ、１３１ｂ、１３１ｃ、…に格納される基準値Ｒａ、Ｒｂ、Ｒｃ、…がリセットされるごとに任意のデータに変更される。このため、第４の構成例では、各ステータスフラグごとに基準値を設定することにより何れかの基準値が第３者に解析された場合のリスクを軽減するだけでなく、基準値が第３者に特定されるリスクを軽減し、第３者が種々の解析手法で基準値を推定したとしても、その基準値をリセット時に無効とすることが可能となる。この結果として、上記第４の構成例によれば、ＩＣカードのセキュリティ性を向上させることができる。さらに、上記第４の構成例では、乱数により各基準値を設定する処理、および、ステータスフラグのチェック処理などをハードウエアにより実行するため、ＩＣカードにおける処理の高速化および効率化が期待できる。

【図面の簡単な説明】

【００８６】

【図１】第１の実施の形態に係るＩＣカードおよびＩＣカードを含むシステムの構成例を示すブロック図。
【図２】非接触式ＩＣカードの全体の構成例を示す図。
【図３】実行条件のチェック結果を示す各フラグの設定例を示す図。
【図４】第１の実施の形態としての第１の処理例の流れを説明するためのフローチャート。
【図５】各使用者に対する各種データへのアクセス権を示すデータテーブルの構成例を示す図。
【図６】各種データへのアクセス権を示す各フラグの設定例を示す図。
【図７】第１の実施の形態としての第１の処理例の流れを説明するためのフローチャート。
【図８】第２の実施の形態に係るＩＣカードおよびＩＣカードを含むシステムの構成例を示すブロック図。
【図９】第２の実施の形態に係るＩＣカードのＣＰＵにおける第１の構成例を示す図。
【図１０】第２の実施の形態に係るＩＣカードのＣＰＵにおける第２の構成例を示す図。
【図１１】第２の実施の形態に係るＩＣカードのＣＰＵにおける第３の構成例を示す図。
【図１２】第２の実施の形態に係るＩＣカードのＣＰＵにおける第４の構成例を示す図。

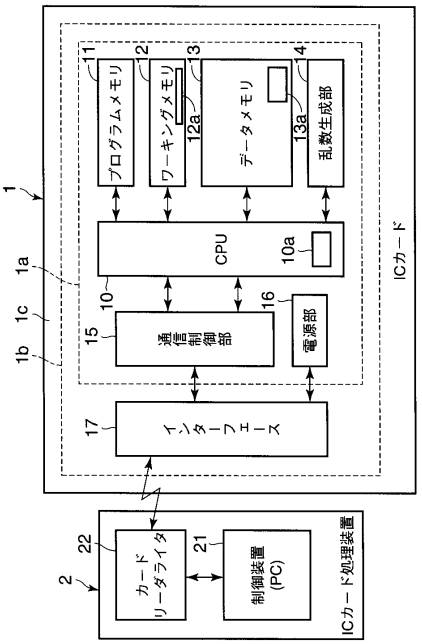
【符号の説明】

【００８７】

Ｒ、Ｒａ、Ｒｂ、Ｒｃ…基準値（第１のデータ）、Ｒ´、Ｒａ´、Ｒｂ´、Ｒｃ´…反転値（第２のデータ）、１、１０１…ＩＣカード（携帯可能電子装置）、１ａ、１０１ａ…ＩＣチップ、１ｂ、１０１ｂ…モジュール、１ｃ、１０１ｃ…本体、２…ＩＣカード処理装置、１０、１１０…ＣＰＵ（判断手段、実行手段、停止手段、エラー処理手段）、１０ａ、１１０ａ…レジスタ（第１の記憶手段）、１１…プログラムメモリ、１２…ワーキングメモリ、１２ａ…フラグ領域（第２の記憶手段）、１３…データメモリ（第３の記憶手段）、１３ａ…データテーブル、１４…乱数生成部（乱数生成手段）、１５…通信制御部、１６…電源部、１７…インターフェース、１１０ｂ…ロジック回路、１３１…設定データ領域（第１の記憶手段）、１３１ａ、１３１ｂ、１３１ｃ…基準値格納領域（第１の記憶手段）、１３２…ステータスフラグ領域（第２の記憶手段）、１３２ａ、１３２ｂ、１３２ｃ…ステータスフラグ（ゼロフラグ、キャリーフラグ、×××フラグ）（第２の記憶手段）、１４１ａ、１４１ｂ、１４１ｃ…比較回路（判断手段）、１４２…乱数生成回路（乱数生成手段）

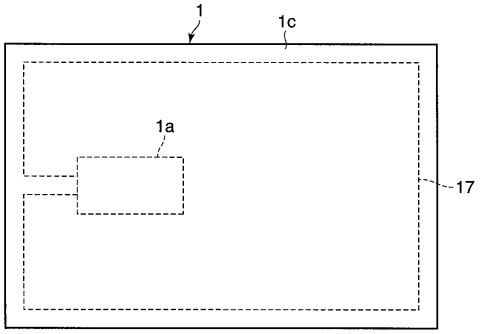
【図１】

図1



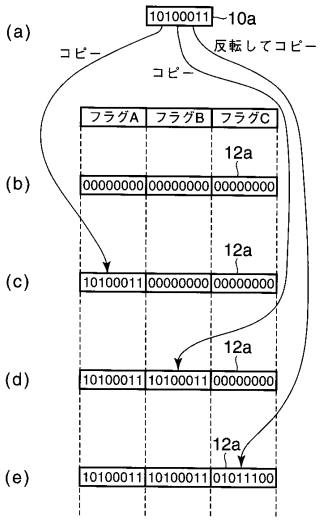
【図２】

図2



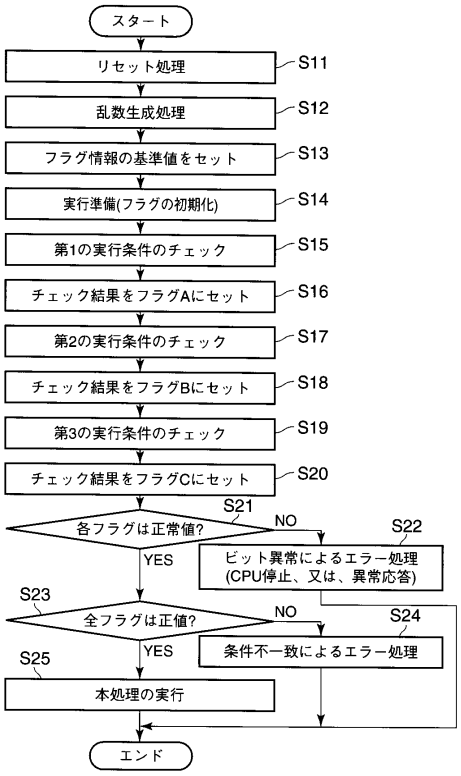
【図３】

図３



【図４】

図４



【図５】

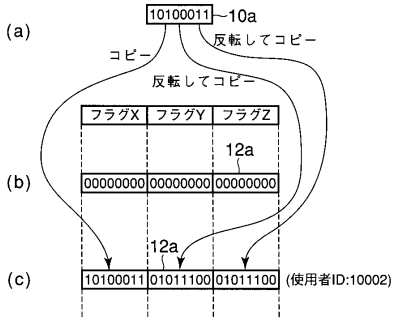
図５

13a

使用者ID	パスワード	アクセス可否		
		データX	データY	データZ
10001	*****	○	×	×
10002	*****	○	×	×
10003	*****	○	×	×
20001	*****	○	○	×
30001	*****	○	○	○

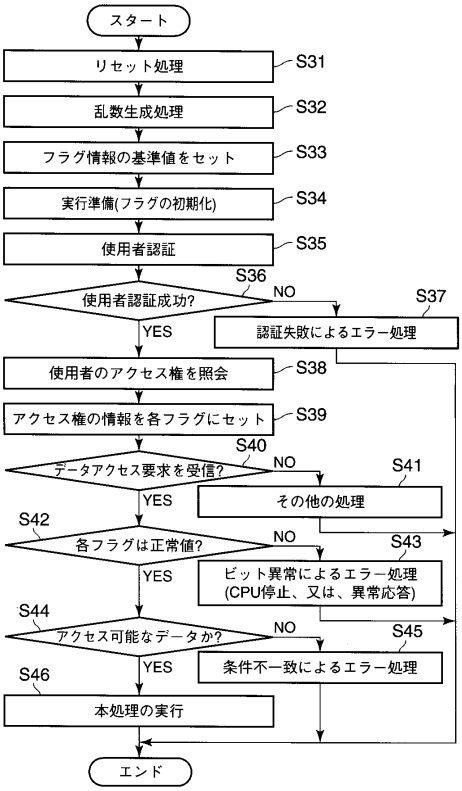
【図６】

図６



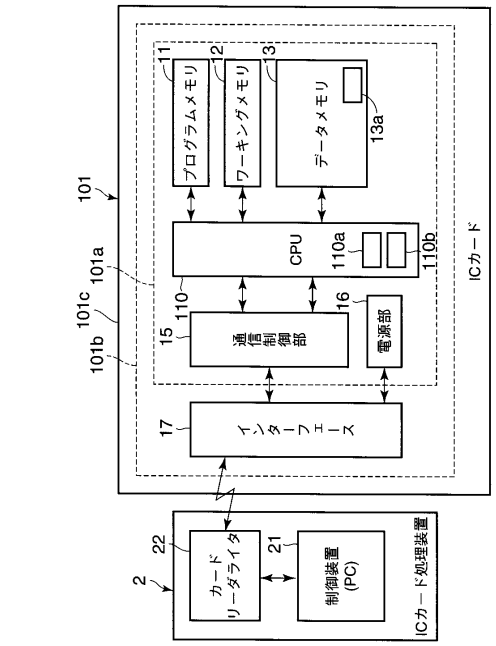
【図７】

図７



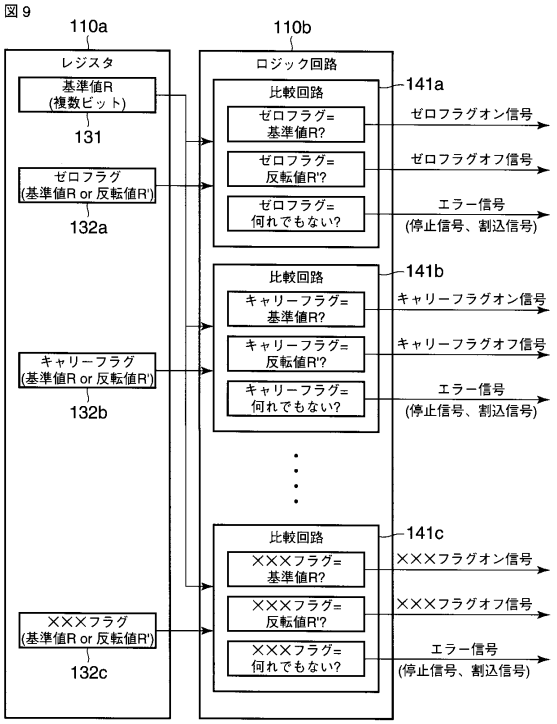
【図８】

図8



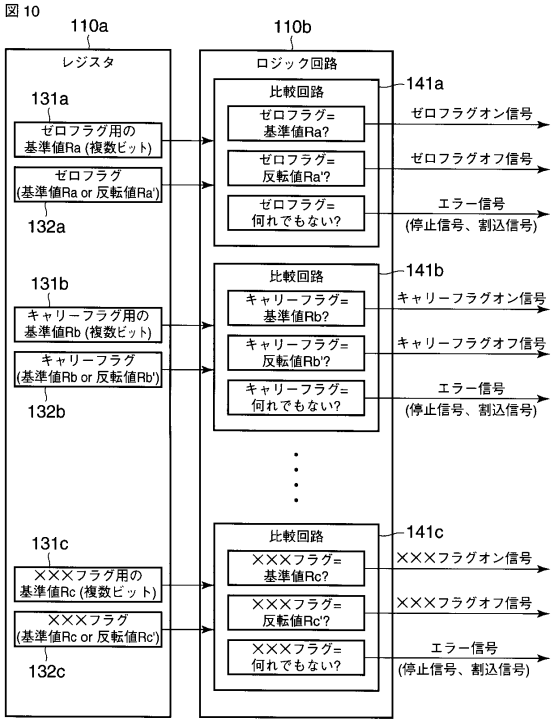
【図９】

図9



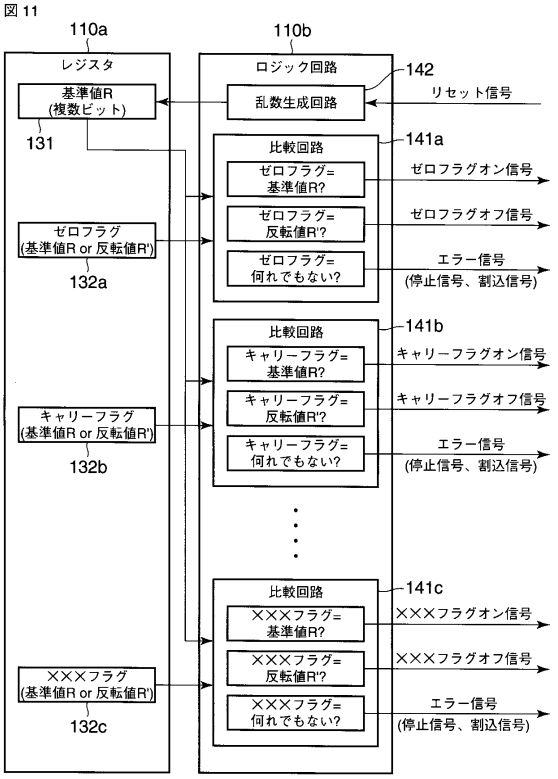
【図１０】

図10



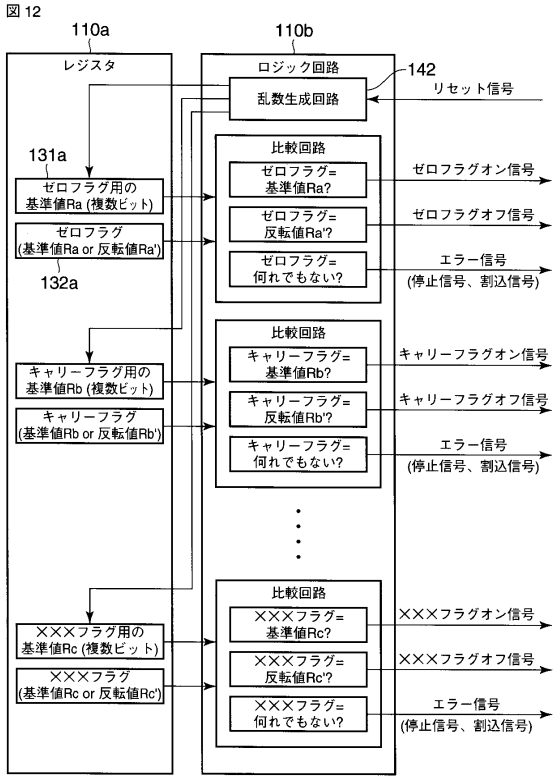
【図１１】

図11



【図１２】

図12


110a
110b
レジスタ
ロジック回路
142
乱数生成回路
リセット信号
131a
ゼロフラグ用の
基準値Ra (複数ビット)
ゼロフラグ
(基準値Ra or 反転値Ra')
132a
比較回路
ゼロフラグ=
基準値Ra?
ゼロフラグ=
反転値Ra'?
ゼロフラグ=
何れでもない?
ゼロフラグオン信号
ゼロフラグオフ信号
エラー信号
(停止信号、割込信号)
キャリーフラグ用の
基準値Rb (複数ビット)
キャリーフラグ
(基準値Rb or 反転値Rb')
比較回路
キャリーフラグ=
基準値Rb?
キャリーフラグ=
反転値Rb'?
キャリーフラグ=
何れでもない?
キャリーフラグオン信号
キャリーフラグオフ信号
エラー信号
(停止信号、割込信号)
×××フラグ用の
基準値Rc (複数ビット)
×××フラグ
(基準値Rc or 反転値Rc')
比較回路
×××フラグ=
基準値Rc?
×××フラグ=
反転値Rc'?
×××フラグ=
何れでもない?
×××フラグオン信号
×××フラグオフ信号
エラー信号
(停止信号、割込信号)

フロントページの続き

(72)発明者　関谷　哲
神奈川県川崎市幸区小向東芝町１番地　東芝ソシオシステムズ株式会社内

審査官　和田　財太

(56)参考文献　国際公開第２００５／０２７４０３（WO，Ａ１）
特開２００５－１４９４３８（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｋ　１９／０７３
Ｇ０６Ｆ　１２／１６
Ｇ０６Ｆ　２１／０２