

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7401215号
(P7401215)

(45)発行日 令和5年12月19日(2023.12.19)

(24)登録日 令和5年12月11日(2023.12.11)

(51)国際特許分類

FI

G 0 6 F 12/02 (2006.01)

G 0 6 F 12/00 (2006.01)

G 0 6 F 8/65 (2018.01)

G 0 6 F 12/02 5 3 0 E

G 0 6 F 12/00 5 9 7 U

G 0 6 F 8/65

請求項の数 21 (全22頁)

(21)出願番号	特願2019-133752(P2019-133752)	(73)特許権者	000001007 キャノン株式会社 東京都大田区下丸子3丁目30番2号
(22)出願日	令和1年7月19日(2019.7.19)	(74)代理人	100125254 弁理士 別役 重尚
(65)公開番号	特開2021-18595(P2021-18595A)	(72)発明者	徳元 曜子 東京都大田区下丸子3丁目30番2号 キャノン株式会社内
(43)公開日	令和3年2月15日(2021.2.15)	審査官	松平 英
審査請求日	令和4年7月15日(2022.7.15)		

最終頁に続く

(54)【発明の名称】 情報処理装置、その制御方法およびプログラム

(57)【特許請求の範囲】

【請求項1】

複数の領域を有する不揮発性半導体メモリと、
ユーザ操作を受け付けることができるユーザインタフェースと、
一連の命令を格納するメモリデバイスと、
前記複数の領域のうちの一の領域に対する消去処理を前記不揮発性半導体メモリに実行させるためのトリムコマンドを出力する制御部と、
を有する情報処理装置であって、
前記制御部は、
前記一の領域のマウント設定を取得し、
取得したマウント設定に応じて、前記一の領域を対象とするトリムコマンドを前記不揮発性半導体メモリに出力するか否かを判断し、
前記制御部は、前記メモリデバイスに格納された命令を実行することによって、
前記情報処理装置についての再起動や前記不揮発性半導体メモリへのアップデート処理に使用するプログラムの実行に応じて、前記不揮発性半導体メモリにトリムコマンドを出力するか否かを判断し、
前記不揮発性半導体メモリは、前記情報処理装置の通常起動において実行されるプログラムを記録する第一領域と、前記第一領域をアップデートする際に実行されるプログラムのみを記録する第二領域と、を有し、
前記制御部は、前記メモリデバイスに格納された命令を実行することによって、

前記情報処理装置を通常起動の状態からアップデートのための起動状態へ再起動し、
アップデートのための再起動状態において、前記第一領域のプログラムをアップデートし、

前記第一領域に対する消去処理を前記不揮発性半導体メモリに実行させるためのトリムコマンドを出力する、

情報処理装置。

【請求項 2】

前記制御部は、前記ユーザインタフェースに対するユーザ操作を受け付けられない状態において、前記メモリデバイスに格納された命令を実行することによって、

前記一の領域のマウント設定を取得し、

取得したマウント設定に応じて、前記一の領域を対象とするトリムコマンドを出力するか否かを判断する、

請求項 1 記載の情報処理装置。

【請求項 3】

前記ユーザインタフェースに対するユーザ操作を受け付けられない状態は、前記情報処理装置をシャットダウンするシャットダウン処理を実行させるためのユーザ操作が前記ユーザインタフェースに受け入れられた場合に実現される、

請求項 2 記載の情報処理装置。

【請求項 4】

前記制御部は、前記メモリデバイスに格納された命令を実行することによって、

取得した前記一の領域のマウント設定が書込可能であるか否かを判断し、

取得した前記一の領域のマウント設定が書込可能であると判断した場合に、前記一の領域を対象とするトリムコマンドを出力する、

請求項 1 から 3 のいずれか一項記載の情報処理装置。

【請求項 5】

前記ユーザインタフェースに対するユーザ操作を受け付けられない状態は、前記情報処理装置についての再起動に伴って行われる前記不揮発性半導体メモリへのアップデート処理によって実現される、

請求項 2 記載の情報処理装置。

【請求項 6】

前記制御部は、前記メモリデバイスに格納された命令を実行することによって、

前記情報処理装置をアップデートのための起動状態から通常起動の状態へ再起動し、

再起動後の通常起動の状態において前記第二領域のプログラムをアップデートする、

請求項 1 記載の情報処理装置。

【請求項 7】

前記制御部は、前記メモリデバイスに格納された命令を実行することによって、

前記情報処理装置のアップデート処理での再起動を開始した後にトリムコマンドを生成するために、前記複数の領域のうちの前記一の領域を書込可能な領域としてマウントし、

アップデートデータを前記一の領域にコピーして差し替え、

前記一の領域に対する消去処理を実行させるためのトリムコマンドを出力し、

前記一の領域のアップデートデータによる再起動処理を実行する、

請求項 1 または 6 記載の情報処理装置。

【請求項 8】

前記制御部は、前記メモリデバイスに格納された命令を実行することによって、

前記情報処理装置のアップデート処理での再起動を開始した後にトリムコマンドを生成するために、前記複数の領域のうち前記一の領域とは異なる他の領域を書込可能な領域としてマウントし、

前記他の領域に対するトリムコマンドを生成する、

請求項 7 記載の情報処理装置。

【請求項 9】

10

20

30

40

50

前記情報処理装置は、プリントジョブ、スキャンジョブまたはコピージョブを実行する画像形成装置である、

請求項 1 から 8 のいずれか一項記載の情報処理装置。

【請求項 1 0】

前記制御部は、前記ユーザインタフェースに対するユーザ操作を受け付けない状態において、

前記一の領域のマウント設定を取得し、

取得した前記一の領域のマウント設定に応じて、前記一の領域を対象とするトリムコマンドを出力するか否かを判断する、

請求項 1 記載の情報処理装置。

10

【請求項 1 1】

複数の領域を有する不揮発性半導体メモリと、

ユーザ操作を受け付けることができるユーザインタフェースと、

一連の命令を格納するメモリデバイスと、

前記複数の領域のうちの一の領域に対する消去処理を前記不揮発性半導体メモリに実行させるためのトリムコマンドを出力する制御部と、

を有する情報処理装置であって、

前記制御部は、

前記一の領域のマウント設定を取得し、

取得したマウント設定に応じて、前記一の領域を対象とするトリムコマンドを前記不揮発性半導体メモリに出力するか否かを判断し、

20

前記複数の領域は、通常起動において実行されるプログラムを記録する第一領域と、アップデートのための起動で実行されるプログラムを記録する第二領域と、を有し、

前記制御部は、前記情報処理装置が通常起動での始動状態にある場合、

前記第一領域と前記第二領域のマウント設定を取得し、

マウント設定が書込可能でない前記第一領域を対象とするトリムコマンドを前記不揮発性半導体メモリに出力せず、

マウント設定が書込可能である前記第二領域を対象とするトリムコマンドを前記不揮発性半導体メモリに出力する、

情報処理装置。

30

【請求項 1 2】

前記制御部は、前記情報処理装置がアップデートのための起動での始動状態にある場合、

前記第一領域と前記第二領域のマウント設定を取得し、

マウント設定が書込可能でない前記第二領域を対象とするトリムコマンドを前記不揮発性半導体メモリに出力せず、

マウント設定が書込可能である前記第一領域を対象とするトリムコマンドを前記不揮発性半導体メモリに出力する、

請求項 1 1 記載の情報処理装置。

【請求項 1 3】

前記トリムコマンドは、対象とする領域で、使用済みで不要なデータを削除するためのトリム処理を実行するためのコマンドである、

40

請求項 1 1 記載の情報処理装置。

【請求項 1 4】

前記制御部は、

前記一の領域のマウント設定が書込可能でない場合、前記一の領域を対象とするトリムコマンドを前記不揮発性半導体メモリに出力せず、

前記一の領域のマウント設定が書込可能である場合、前記一の領域を対象とするトリムコマンドを前記不揮発性半導体メモリに出力する、

請求項 1 1 記載の情報処理装置。

【請求項 1 5】

50

前記第一領域に記録されるプログラムには、前記情報処理装置の起動プログラムを含む、請求項 1 または 1 1 記載の情報処理装置。

【請求項 1 6】

前記第一領域に記録されるプログラムには、スキャナまたはプリンタを制御するプログラムを含む

請求項 1 または 1 1 記載の情報処理装置。

【請求項 1 7】

前記第二領域に記録されるプログラムには、f s t r i m 実行プログラムを含む

請求項 1 または 1 1 記載の情報処理装置。

【請求項 1 8】

複数の領域を有する不揮発性半導体メモリと、一連の命令を格納するメモリデバイスと、を用いて処理を実行する情報処理装置の制御方法であって、

前記複数の領域のうちの一の領域に対する消去処理を前記不揮発性半導体メモリに実行させるためのトリムコマンドを前記不揮発性半導体メモリに出力する第 1 出力工程を有し、前記第 1 出力工程は、

前記一の領域のマウント設定を取得する取得工程と、

取得したマウント設定に応じて、前記一の領域を対象とするトリムコマンドを前記不揮発性半導体メモリに出力するか否かを判断する第 1 判断工程と、

前記メモリデバイスに格納された命令を実行することによって、前記情報処理装置についての再起動や前記不揮発性半導体メモリへのアップデート処理に使用するプログラムの実行に応じて、前記不揮発性半導体メモリにトリムコマンドを出力するか否かを判断する第 2 判断工程と、を有し、

前記不揮発性半導体メモリは、前記情報処理装置の通常起動において実行されるプログラムを記録する第一領域と、前記第一領域をアップデートする際に実行されるプログラムのみを記録する第二領域と、を有し、

前記第 1 出力工程は、前記メモリデバイスに格納された命令を実行することによって、

前記情報処理装置を通常起動の状態からアップデートのための起動状態へ再起動する再起動工程と、

アップデートのための再起動状態において、前記第一領域のプログラムをアップデートするアップデート工程と、

前記第一領域に対する消去処理を前記不揮発性半導体メモリに実行させるためのトリムコマンドを出力する第 2 出力工程と、

を有する、情報処理装置の制御方法。

【請求項 1 9】

複数の領域を有する不揮発性半導体メモリと、一連の命令を格納するメモリデバイスと、を用いて処理を実行する情報処理装置の制御方法であって、

前記複数の領域のうちの一の領域に対する消去処理を前記不揮発性半導体メモリに実行させるためのトリムコマンドを前記不揮発性半導体メモリに出力する第 1 出力工程を有し、前記第 1 出力工程は、

前記一の領域のマウント設定を取得する第 1 取得工程と、

取得したマウント設定に応じて、前記一の領域を対象とするトリムコマンドを前記不揮発性半導体メモリに出力するか否かを判断する判断工程とを、有し、

前記複数の領域は、通常起動において実行されるプログラムを記録する第一領域と、アップデートのための起動で実行されるプログラムを記録する第二領域と、を有し、

前記第 1 出力工程は、前記情報処理装置が通常起動での始動状態にある場合、

前記第一領域と前記第二領域のマウント設定を取得する第 2 取得工程と、

マウント設定が書込可能でない前記第一領域を対象とするトリムコマンドを前記不揮発性半導体メモリに出力せず、マウント設定が書込可能である前記第二領域を対象とするトリムコマンドを前記不揮発性半導体メモリに出力する第 2 出力工程と、

を有する、情報処理装置の制御方法。

10

20

30

40

50

【請求項 20】

複数の領域を有する不揮発性半導体メモリと、一連の命令を格納するメモリデバイスと、を用いて処理を実行する情報処理装置の制御方法をコンピュータに実行させるプログラムであって、

前記情報処理装置の制御方法は、

前記複数の領域のうちの一の領域に対する消去処理を前記不揮発性半導体メモリに実行させるためのトリムコマンドを前記不揮発性半導体メモリに出力する第1出力工程を有し、

前記第1出力工程は、

前記一の領域のマウント設定を取得する取得工程と、

取得したマウント設定に応じて、前記一の領域を対象とするトリムコマンドを前記不揮発性半導体メモリに出力するか否かを判断する第1判断工程と、

前記メモリデバイスに格納された命令を実行することによって、前記情報処理装置についての再起動や前記不揮発性半導体メモリへのアップデート処理に使用するプログラムの実行に応じて、前記不揮発性半導体メモリにトリムコマンドを出力するか否かを判断する第2判断工程と、を有し、

前記不揮発性半導体メモリは、前記情報処理装置の通常起動において実行されるプログラムを記録する第一領域と、前記第一領域をアップデートする際に実行されるプログラムのみを記録する第二領域と、を有し、

前記第1出力工程は、前記メモリデバイスに格納された命令を実行することによって、

前記情報処理装置を通常起動の状態からアップデートのための起動状態へ再起動する再起動工程と、

アップデートのための再起動状態において、前記第一領域のプログラムをアップデートするアップデート工程と、

前記第一領域に対する消去処理を前記不揮発性半導体メモリに実行させるためのトリムコマンドを出力する第2出力工程と、

を有する、プログラム。

【請求項 21】

複数の領域を有する不揮発性半導体メモリと、一連の命令を格納するメモリデバイスと、を用いて処理を実行する情報処理装置の制御方法をコンピュータに実行させるプログラムであって、

前記情報処理装置の制御方法は、

前記複数の領域のうちの一の領域に対する消去処理を前記不揮発性半導体メモリに実行させるためのトリムコマンドを前記不揮発性半導体メモリに出力する第1出力工程を有し、

前記第1出力工程は、

前記一の領域のマウント設定を取得する第1取得工程と、

取得したマウント設定に応じて、前記一の領域を対象とするトリムコマンドを前記不揮発性半導体メモリに出力するか否かを判断する判断工程とを、有し、

前記複数の領域は、通常起動において実行されるプログラムを記録する第一領域と、アップデートのための起動で実行されるプログラムを記録する第二領域と、を有し、

前記第1出力工程は、前記情報処理装置が通常起動での始動状態にある場合、

前記第一領域と前記第二領域のマウント設定を取得する第2取得工程と、

マウント設定が書込可能でない前記第一領域を対象とするトリムコマンドを前記不揮発性半導体メモリに出力せず、マウント設定が書込可能である前記第二領域を対象とするトリムコマンドを前記不揮発性半導体メモリに出力する第2出力工程と、

を有する、プログラム。

【発明の詳細な説明】

【技術分野】

【0001】

10

20

30

40

50

本発明は、情報処理装置に関する。

【背景技術】

【0002】

近年、情報処理装置では、そのメモリとして、たとえばSSD（ソリッドステートドライブ）、eMMCといった、フラッシュメモリを用いる不揮発性半導体メモリ装置を使用することが増えている。不揮発性半導体メモリ装置は、アクセス速度、静音性などの点においてHDD（ハードディスクドライブ）などの他のメモリより優れている。その一方で、不揮発性半導体メモリ装置に用いる不揮発性半導体メモリには、書き換え回数に制限がある。また、不揮発性半導体メモリの書き込み済みの領域に新たなデータを書込むためには、書込処理とは別に、領域を書き込み可能にするための消去処理を実行する必要がある。書き換え回数の制限については、たとえば不揮発性半導体メモリ装置にキャッシュメモリを設け、不揮発性半導体メモリへの書き込みをキャッシュメモリからの書き込みとすることにより、外部装置から不揮発性半導体メモリ装置への書き込みの度に不揮発性半導体メモリへの書き込みが生じないようにしたり、不揮発性半導体メモリ装置のメモリコントローラにおいてウェアレベリングと呼ばれる分散書き込みを実行して、たとえばブロックといった単位の領域で書き込み回数の平均化を図ったり、することにより改善できる。書き込み済みのデータの消去処理については、不揮発性半導体メモリ装置のメモリコントローラが、不揮発性半導体メモリへの新たなデータの書き込みの度に、処理対象の領域の不要になったデータを消去するトリム処理を実行したり、情報処理装置のコントローラが、消去処理を指示するトリムコマンドを、不揮発性半導体メモリ装置のメモリコントローラに指示したり、することにより改善できる。

10

20

【先行技術文献】

【特許文献】

【0003】

【文献】特許第6289128号公報

【発明の概要】

【発明が解決しようとする課題】

【0004】

しかしながら、このような対策を講じ得たとしても、不揮発性半導体メモリ装置においてメモリコントローラは、トリムコマンドなどの消去コマンドに基づいて、不揮発性半導体メモリの使用済みの領域についてのデータの消去処理を実行する必要がある。そして、その消去処理の実行中には、不揮発性半導体メモリへアクセスすることができない。このため、情報処理装置の処理は、不揮発性半導体メモリにアクセスできないことに起因して遅延したり、動作速度が低下したりする。このため、特許文献1では、メモリコントローラが、情報処理装置についてのジョブ実行などの使用状況に基づいてジョブの中断状況であると判断できる場合にトリム処理を実行し、ジョブの中断状況でなくなるとトリム処理を中断する。しかしながら、情報処理装置がその処理に不揮発性半導体メモリへのアクセスを必要とする場合は、ジョブを実行する場合に限られるものではない。たとえばユーザが情報処理装置に対して操作をする場合においても、情報処理装置は、その処理のために不揮発性半導体メモリにアクセスすることがある。そして、ジョブの中断状態においてユーザが情報処理装置に対して操作をすると、メモリコントローラがトリム処理の中断を終えるまで、情報処理装置は、その処理のために不揮発性半導体メモリにアクセスすることができない。この場合においても、ユーザは、不揮発性半導体メモリ装置にアクセスできないことに起因する遅延を体感したり、動作速度の低下を体感したり、することになる。

30

40

【0005】

このように情報処理装置では、不揮発性半導体メモリへのアクセスを改善することが求められている。

【課題を解決するための手段】

【0006】

本発明に係る情報処理装置は、複数の領域を有する不揮発性半導体メモリと、ユーザ操

50

作を受け付けることができるユーザインタフェースと、一連の命令を格納するメモリデバイスと、前記複数の領域のうちの一の領域に対する消去処理を前記不揮発性半導体メモリに実行させるためのトリムコマンドを出力する制御部と、を有する情報処理装置であって、前記制御部は、前記一の領域のマウント設定を取得し、取得したマウント設定に応じて、前記一の領域を対象とするトリムコマンドを前記不揮発性半導体メモリに出力するか否かを判断し、前記制御部は、前記メモリデバイスに格納された命令を実行することによって、前記情報処理装置についての再起動や前記不揮発性半導体メモリへのアップデート処理に使用するプログラムの実行に応じて、前記不揮発性半導体メモリにトリムコマンドを出力するか否かを判断し、前記不揮発性半導体メモリは、前記情報処理装置の通常起動において実行されるプログラムを記録する第一領域と、前記第一領域をアップデートする際に実行されるプログラムのみを記録する第二領域と、を有し、前記制御部は、前記メモリデバイスに格納された命令を実行することによって、前記情報処理装置を通常起動の状態からアップデートのための起動状態へ再起動し、アップデートのための再起動状態において、前記第一領域のプログラムをアップデートし、前記第一領域に対する消去処理を前記不揮発性半導体メモリに実行させるためのトリムコマンドを出力する。

10

【発明の効果】

【0007】

本発明では、不揮発性半導体メモリへのアクセスを改善することができる。

【図面の簡単な説明】

【0008】

20

【図1】本発明の実施形態に係る情報処理装置の構成図である。

【図2】図1のeMMCの構成図である。

【図3】図2のeMMCによるシステム領域の一例の説明図である。

【図4】フラッシュメモリの領域のマウント状態の説明図である。

【図5】図1の情報処理装置のファームウェアプログラムのモジュール構成の説明図である。

【図6】図1の情報処理装置についての、トリム処理を含むシャットダウン処理のフローチャートである。

【図7】図1の情報処理装置についての、トリム処理を含むファームウェアプログラムのアップデート処理のフローチャートである。

30

【発明を実施するための形態】

【0009】

以下、本発明の実施形態について図面を参照しながら詳細に説明する。しかしながら、以下の実施形態に記載されている構成はあくまで例示に過ぎず、本発明の範囲は実施形態に記載されている構成によって限定されることはない。

【0010】

図1は、本発明の実施形態に係る情報処理装置100の構成図である。図1の情報処理装置100は、操作部110、プリンタエンジン120、スキャナエンジン130、および、これらが接続される制御部140、電源ユニット160、を有する。図1の情報処理装置100は、プリントジョブ、スキャンジョブおよびコピージョブを実行可能なMFP (Multi Function Peripheral) である。なお、情報処理装置100は、MFPといった画像形成装置に限られない。

40

【0011】

電源ユニット160は、商用電力を変換し、情報処理装置100の各部へ供給する。電源ユニット160では、不図示の電源ユニット160の電源コントローラが、不図示のROMに記録される電源ファームウェアプログラムを実行し、各部への電力供給を制御する。操作部110は、たとえばテンキーデバイス、タッチパネルデバイス、液晶デバイス、を有する。操作部110では、不図示の操作部110の操作コントローラが、不図示のROMに記録される操作ファームウェアプログラムを実行し、操作部110の各部の動作を制御する。これにより、操作部110は、情報処理装置100の処理のためにユーザ操作

50

可能なユーザインタフェースとして機能する。プリンタエンジン 120 では、不図示のプリンタエンジンコントローラが、不図示の ROM に記録されるプリンタファームウェアプログラム 121 を実行し、用紙などの印刷媒体へ画像データを印刷する。スキャナエンジン 130 では、不図示のスキャナエンジンコントローラが、不図示の ROM に記録されるスキャンファームウェアプログラム 131 を実行し、原稿を読み取って画像データを生成する。制御部 140 は、情報処理装置 100 の全体的な動作を制御する。制御部 140 は、CPU 141、ROM 142、Chip set 143、RAM 144、SRAM 145、HDD 146、eMMC 147、Embedded Controller 148、および、これらを接続するシステムバス 149、を有する。システムバス 149 には、この他にも、LED 150、ネットワーク I/F カード (NIC) 151、操作 I/F 152、プリンタ I/F 153、スキャナ I/F 154、が接続される。操作 I/F 152 は、操作部 110 に接続され、操作部 110 との間でデータを入手力する。スキャナ I/F 154 は、スキャナエンジン 130 に接続され、スキャナエンジン 130 との間でデータを入手力する。プリンタ I/F 153 は、プリンタエンジン 120 に接続され、プリンタエンジン 120 との間でデータを入手力する。LED 150 は、情報処理装置 100 の状態を表示する。情報処理装置 100 の状態には、たとえば情報処理装置 100 のソフトウェアの異常状態、ハードウェアの異常状態がある。LED 150 は、たとえば操作 I/F 152 と並べて設けられる。ネットワーク I/F カード 151 は、たとえば LAN 170 に接続される。LAN 170 には、この他にもたとえば外部サーバ 180、パーソナルコンピュータ (PC) 190、が接続される。外部サーバ 180 は、LAN 170 に接続されるインターネットに設けられてもよい。この場合、ネットワークは、LAN 170 およびインターネットにより構成される。ネットワーク I/F カード 151 は、LAN 170 上のたとえば外部サーバ 180 といった外部装置との間でデータを送受する。送受するデータには、たとえばプリントジョブのデータ、情報処理装置 100 で使用するアップデートプログラムのデータ、設定データ、がある。ネットワーク I/F カード 151 は、情報処理装置 100 の通信部として機能する。Chip set 143 は、関連のある複数の集積回路の集合体である。Chip set 143 には、たとえばリアルタイムクロック (Real Time Clock) 155 を有する。リアルタイムクロック 155 は、時間や時刻を計測する専用の集積回路である。リアルタイムクロック 155 は、情報処理装置 100 が休止しているスリープ状態においても不図示の内蔵電池から電源供給を受けて動作する。リアルタイムクロック 155 は、たとえば所定のタイミングにおいて情報処理装置 100 を起動する信号を出力する。ただし、情報処理装置 100 が停止しているシャットダウン状態では、リアルタイムクロック 155 は、停止する。この場合、情報処理装置 100 の電源ユニット 160 の不図示のスイッチを操作することにより、情報処理装置 100 は起動できる。Embedded Controller 148 は、情報処理装置 100 のために開発された集積回路である。Embedded Controller 148 は、サブ CPU 156、サブ RAM 157、を有する。サブ RAM 157 は、ランダムアクセスメモリで、CPU 141 が情報処理装置 100 を制御する際に、プログラムや一時的なデータの格納などに使用される。サブ CPU 156 は、Embedded Controller 148 のソフトウェアプログラムを実行し、情報処理装置 100 の中で一部の制御を行う。

【0012】

SRAM 145 は、情報処理装置 100 の設定値、画像調整値などを記録する。SRAM 145 は、不揮発メモリであり、電源を遮断しても再投入までデータを保持する。HDD 146 は、大容量の記録デバイスである。HDD 146 は、たとえば、画像データ、ユーザデータ、ファームウェアのアップデートプログラム、などを記録する。HDD 146 は、たとえばファームウェアプログラムを記録してよい。eMMC 147 は、フラッシュメモリ 205 を有する不揮発性半導体メモリ装置である。フラッシュメモリ 205 は、不揮発半導体メモリである。eMMC 147 は、ファームウェアプログラム、ワーク領域 304、ユーザデータ領域、などをフラッシュメモリ 205 に記録する。この他にもたとえ

10

20

30

40

50

ば、eMMC 147は、情報処理装置100にHDD 146が設けられない場合などにおいて、HDD 146に記録する画像データ、ユーザデータ、をフラッシュメモリ205に記録してよい。ROM 142は、リードオンリーメモリである。ROM 142は、たとえば情報処理装置100のBIOS、固定設定値を記録する。RAM 144は、CPU 141のメインメモリである。RAM 144は、ワーク領域304などの一時記録領域として用いられてよい。CPU 141は、eMMC 147などに記録されているファームウェアプログラムを読み出してRAM 144に展開する。これにより、CPU 141は、ファームウェアプログラムを実行して、情報処理装置100の全体の動作を制御する。制御部140としてのCPU 141は、たとえば、ユーザインタフェースとしての操作部110におけるユーザ操作に基づいて、読取処理、印刷処理といった処理を実行する。これにより、情報処理装置100は、画像形成装置として、プリントジョブ、スキャンジョブまたはコピージョブを実行できる。

10

【0013】

図2は、図1のeMMC 147の構成図である。図2のeMMC 147は、SATAI/F 201、メモリコントローラ203、キャッシュメモリ204、フラッシュメモリ205、およびこれらが接続されるバスコントローラ202、を有する。図2には、情報処理装置100のSATAI/F 210も図示されている。フラッシュメモリ205は、たとえばNAND型のフラッシュメモリである。フラッシュメモリ205は、たとえばブロックといった単位での領域ごとに、データを削除することができる。また、フラッシュメモリ205は、データが書き込まれている領域には、その領域からデータを削除した後でないと、新たなデータを書き込むことができない。キャッシュメモリ204は、フラッシュメモリ205から読み出されたデータ、フラッシュメモリ205に書き込むデータの一時格納メモリである。SATAI/F 201は、システムバス149により他のSATAI/F 201と接続されるインタフェースである。バスコントローラ202は、SATAI/F 201とキャッシュメモリ204との間でのデータ入出力と、キャッシュメモリ204とフラッシュメモリ205との間でのデータ入出力とを、排他的に制御する。メモリコントローラ203は、eMMC 147を管理して動作を制御する。メモリコントローラ203は、たとえばSATAI/F 201を通じたSATA規格に基づくeMMC 147へのデータ入出力を制御する。メモリコントローラ203は、キャッシュメモリ204とフラッシュメモリ205との間でのデータ入出力を制御する。

20

30

【0014】

図3は、図2のeMMC 147によるシステム領域300の一例の説明図である。図3は、後述するように情報処理装置100が通常起動された場合でのeMMC 147によるシステム領域300である。システム領域300とは、CPU 141が管理してアクセス可能な記憶領域である。eMMC 147によるシステム領域300には、複数のパーティションとして、stdファーム格納領域301、safeファーム格納領域302、情報処理装置100のユーザ数分の複数のユーザデータ格納領域303および複数のワーク領域304、を有する。HDD 146が搭載されない情報処理装置100においては、eMMC 147によるシステム領域300には、さらに後述する、ファームウェアのアップデートプログラムのダウンロード領域571、アップデートプログラムを展開するための一時展開領域572、をマウントしてよい。eMMC 147では、図3のシステム領域300の各パーティションに、フラッシュメモリ205の物理的な記憶領域を割り当てる。stdファーム格納領域301は、情報処理装置100を通常起動する場合においてCPU 141により実行されるファームウェアプログラムを記録する。safeファーム格納領域302は、情報処理装置100をアップデートのための起動する場合においてCPU 141により実行されるファームウェアプログラムを記録する。ユーザデータ格納領域303は、ユーザ毎のデータを記録する。ワーク領域304は、ジョブ実行の際にユーザ毎のデータを記録する。このように図3のシステム領域300のstdファーム格納領域301とsafeファーム格納領域302とには、情報処理装置100のCPU 141が実行可能なファームウェアプログラム、が記録される。

40

50

【 0 0 1 5 】

図 4 は、フラッシュメモリ 2 0 5 の領域のマウント状態の説明図である。図 4 のフラッシュメモリ 2 0 5 の各領域は、図 3 のシステム領域 3 0 0 に対応する。情報処理装置 1 0 0 の起動状態には、図 4 に示すように、通常起動モード 4 0 1 と、アップデートのための起動モード 4 0 2 と、がある。通常起動モード 4 0 1 で起動する場合、e M M C 1 4 7 のメモリコントローラ 2 0 3 および C P U 1 4 1 は、通常起動において実行するプログラムを記録する s t d ファーム格納領域 3 0 1 を、読み込み専用モードに設定する。それ以外の s a f e ファーム格納領域 3 0 2、複数のユーザデータ格納領域 3 0 3 および複数のワーク領域 3 0 4 を、読み書き可能に設定する。アップデートのための起動モード 4 0 2 で起動する場合、メモリコントローラ 2 0 3 および C P U 1 4 1 は、アップデートのための起動において実行するプログラムを記録する s a f e ファーム格納領域 3 0 2 を、読み込み専用モードに設定する。また、アップデートする s t d ファーム格納領域 3 0 1 を、読み書き可能に設定する。複数のユーザデータ格納領域 3 0 3 および複数のワーク領域 3 0 4 は、アップデートの際にはアクセスしない領域であるため、マウントしない。このように、たとえば e M M C 1 4 7 のメモリコントローラ 2 0 3 は、情報処理装置 1 0 0 の起動時に、起動状態に応じて、フラッシュメモリ 2 0 5 の各領域を、図 4 の起動モードの設定に応じた状態でシステム領域 3 0 0 にマウントする。マウント後には、C P U 1 4 1 は、それぞれのマウント状態に基づいてフラッシュメモリ 2 0 5 の各領域にアクセスできる。

10

【 0 0 1 6 】

図 5 は、図 1 の情報処理装置 1 0 0 のファームウェアプログラムのモジュール構成の説明図である。図 5 には、情報処理装置 1 0 0 で実行されるプログラムとして、ブートプログラム 5 1 0、B I O S プログラム 5 2 0、ローダープログラム 5 3 0、ファームウェアプログラム 5 4 0、が図示されている。ブートプログラム 5 1 0 は、E m b e d d e d C o n t r o l l e r 1 4 8 の不図示のメモリに記録される。情報処理装置 1 0 0 の電源ユニット 1 6 0 において不図示のシーソーススイッチがオン操作されると、電源ユニット 1 6 0 から E m b e d d e d C o n t r o l l e r 1 4 8 へ電力が供給される。E m b e d d e d C o n t r o l l e r 1 4 8 が起動すると、サブ C P U 1 5 6 は、ブートプログラム 5 1 0 を実行する。サブ C P U 1 5 6 は、情報処理装置 1 0 0 において起動に関わる処理を実行する。B I O S プログラム 5 2 0 は、たとえば S R A M 1 4 5 に記録される。サブ C P U 1 5 6 がブートプログラム 5 1 0 を実行すると、C P U 1 4 1 は、B I O S プログラム 5 2 0 を R A M 1 4 4 に読み込んで実行する。C P U 1 4 1 は、情報処理装置 1 0 0 において起動に関わる処理を実行する。ローダープログラム 5 3 0 は、たとえば S R A M 1 4 5 に記録される。C P U 1 4 1 は、B I O S プログラム 5 2 0 を実行した後、ローダープログラム 5 3 0 を R A M 1 4 4 に読み込んで実行する。C P U 1 4 1 は、情報処理装置 1 0 0 において起動に関わる処理を実行する。

20

30

【 0 0 1 7 】

ファームウェアプログラム 5 4 0 は、基本的に通常起動モード 4 0 1 において実行されるために、e M M C 1 4 7 のフラッシュメモリ 2 0 5 の s t d ファーム格納領域 3 0 1 に記録される。C P U 1 4 1 は、ローダープログラム 5 3 0 を実行した後、ファームウェアプログラム 5 4 0 を R A M 1 4 4 に読み込んで実行する。C P U 1 4 1 は、情報処理装置 1 0 0 において起動に関わる処理を実行する。s t d ファーム格納領域 3 0 1 に記録されるファームウェアプログラム 5 4 0 は、情報処理装置 1 0 0 の各機能に対応する複数のプログラムで構成される。s t d ファーム格納領域 3 0 1 に記録されるファームウェアプログラム 5 4 0 には、たとえば情報処理装置 1 0 0 の起動プログラム、操作 I / F 1 5 2、スキャナ I / F 1 5 4 またはプリンタ I / F 1 5 3 を制御するプログラム、複数のアプリケーションプログラム 5 4 1、が含まれる。

40

【 0 0 1 8 】

C P U 1 4 1 は、ファームウェアプログラム 5 4 0 を実行する場合、最初に起動プログラムを R A M 1 4 4 に読み込んで実行する。起動プログラムを実行する C P U 1 4 1 は、通常起動中に使用する e M M C 1 4 7 や H D D 1 4 6 の各領域をマウントする起動処理を

50

実行する。また、起動プログラムを実行するCPU141は、その他のプログラムであるアプリケーションプログラム541を、ファームウェアプログラム540から読み込んで実行する。アプリケーションプログラム541には、シャットダウン実行プログラム542が含まれる。一連の実行が完了すると、情報処理装置100は、ファームウェアプログラム540により制御された状態に起動される。情報処理装置100は、たとえば操作部110のユーザ操作に基づいて、プリントジョブなどを実行することができる。また、情報処理装置100には、図5の通信管理部550、UI制御部560が実現される。通信管理部550は、ネットワークI/Fカード151を用いて、外部サーバ180などとデータを送受する。UI制御部560は、操作I/F152を介して操作部110を制御して、操作部110へ操作画面を表示させたり、操作部110に対するユーザ操作を取得したりする。起動状態にある情報処理装置100において電源ユニット160のシーソーススイッチがオフ操作されると、CPU141は、シャットダウン実行プログラム542を実行して、シャットダウン処理を開始する。CPU141は、たとえば各アプリケーションプログラム541に終了通知を行う。CPU141は、各アプリケーションプログラム541の終了を確認すると、電源ユニット160に電源をOFFする要求を出力する。電源ユニット160は、情報処理装置100の各部への給電を終了する。これにより、情報処理装置100は、停止しているシャットダウン状態となる。

【0019】

また、図5に示すように、eMMC147のフラッシュメモリ205のstdファーム格納領域301には、ファームウェアプログラム540の一部として、ファームウェアのダウンロード実行プログラム543、バックグラウンドアップデート実行プログラム544、fs trim実行プログラム545、safeファーム格納領域302のアップデート実行プログラム546、が記録される。CPU141は、情報処理装置100が通常起動した状態において、stdファーム格納領域301に実行可能に記録されているファームウェアのダウンロード実行プログラム543を繰り返し実行する。これにより、CPU141は、外部サーバ180などから、情報処理装置100のファームウェアのアップデートプログラムを取得する。CPU141は、取得したアップデートプログラムを、HDD146のダウンロード領域571に記録する。アップデートプログラムをHDD146のダウンロード領域571に記録すると、CPU141は、バックグラウンドアップデート実行プログラム544を実行する。CPU141は、ダウンロード領域571のアップデートプログラムを、HDD146の一時展開領域572に展開する。HDD146の一時展開領域572には、アップデートされたファームウェアプログラム540が記録される。HDD146の一時展開領域572に展開されているファームウェアプログラムには、プリンタファームウェアプログラム121やスキャンファームウェアプログラム131についてのアップデートプログラムが含まれてよい。情報処理装置100に不図示のフィニッシャ装置などの他の処理装置が連結されている場合、HDD146の一時展開領域572に展開されているファームウェアプログラムには、他の処理装置のファームウェアプログラム540についてのアップデートプログラムが含まれてよい。CPU141は、情報処理装置100が通常起動した状態において、stdファーム格納領域301に実行可能に記録されているsafeファーム格納領域302のアップデート実行プログラム546を実行する。CPU141は、HDD146の一時展開領域572に展開されているアップデートプログラムをeMMC147へ出力し、eMMC147にsafeファーム格納領域302への格納を指示する。eMMC147のメモリコントローラ203は、フラッシュメモリ205のsafeファーム格納領域302を、アップデートプログラムにより更新する。safeファーム格納領域302には、アップデートされたファームウェアプログラム540が記録される。

【0020】

CPU141は、情報処理装置100が通常起動した状態において、アプリケーションプログラム541による任意のタイミングにおいて、stdファーム格納領域301のfs trim実行プログラム545を適宜実行してよい。CPU141は、制御部140と

10

20

30

40

50

して、フラッシュメモリ 205 の記憶領域のトリム処理を指示するトリムコマンドとして `f s t r i m` コマンドを生成し、`e M M C 1 4 7` へ出力する。`e M M C 1 4 7` のフラッシュメモリ 205 は、書込済みの領域に新たなデータを書込むためには、書込処理とは別に、領域ごとの消去処理を実行する必要がある。`e M M C 1 4 7` のメモリコントローラ 203 は、`f s t r i m` コマンドに基づいて、フラッシュメモリ 205 の領域についてのトリム処理を実行する。これにより、フラッシュメモリ 205 からは、情報処理装置 100 において不要となったデータなどが、それを格納している領域から削除される。データが削除された領域は、新たなデータを書込むことができる状態になる。その後、`e M M C 1 4 7` のメモリコントローラ 203 は、データを削除した領域を除いて、ウェアレベリング処理を実行することが可能になる。これにより、`e M M C 1 4 7` のパフォーマンスの低下を抑制できる。

10

【0021】

なお、`C P U 1 4 1` は、`f s t r i m` コマンドの代わりに、`e M M C 1 4 7` のパーティションをマウントする際に、`" d i s c a r d "` と呼ばれるマウントオプションを指定して登録することもできる。この場合、`e M M C 1 4 7` のメモリコントローラ 203 は、フラッシュメモリ 205 の `" d i s c a r d "` とのマウントオプションが設定されている領域にデータを書き込む場合、データを書き込んだ領域について、トリム処理を実行する。これにより、`" d i s c a r d "` とのマウントオプションが設定されている領域は、常に新たなデータを書き込むことができる状態に維持される。この場合、`C P U 1 4 1` は、領域に対するトリム処理を実行させるために、アプリケーションプログラム 541 などの実行中において `f s t r i m` コマンドを生成する必要がない。また、オペレーティングシステムによっては、`f s t r i m` コマンドに対応していないこともある。しかしながら、`" d i s c a r d "` と呼ばれるマウントオプションを指定すると、領域へデータを書込むたびにトリム処理が逐次的に実行される。トリム処理を実行している間、`C P U 1 4 1` は、`e M M C 1 4 7` のフラッシュメモリ 205 にアクセスすることができない。その結果、一回の書き込み処理に実質的に必要となる時間が長くなる。

20

【0022】

また、図 5 において `e M M C 1 4 7` のフラッシュメモリ 205 の `s a f e` ファーム格納領域 302 には、`f s t r i m` 実行プログラム 547、`s t d` ファーム格納領域 301 のアップデート実行プログラム 548、が記録される。`C P U 1 4 1` が実行するファームウェアプログラム 540 は、フラッシュメモリ 205 の `s t d` ファーム格納領域 301 と `s a f e` ファーム格納領域 302 とにパーティションで分けて記録される。そして、`s t d` ファーム格納領域 301 と `s a f e` ファーム格納領域 302 とは、図 4 の起動モードに応じた設定でマウントされる。`C P U 1 4 1` は、通常起動では `s t d` ファーム格納領域 301 のファームウェアプログラムのみを実行する。`C P U 1 4 1` は、アップデートのための起動では `s a f e` ファーム格納領域 302 のファームウェアプログラムのみを実行する。たとえば、`C P U 1 4 1` は、情報処理装置 100 が通常起動した状態において、`s t d` ファーム格納領域 301 に記録されている `s a f e` ファーム格納領域 302 のアップデート実行プログラム 546 を実行する。これにより、`C P U 1 4 1` は、`s a f e` ファーム格納領域 302 のファームウェアプログラムをアップデートする。また、`C P U 1 4 1` は、情報処理装置 100 が通常起動した状態において、`s t d` ファーム格納領域 301 に記録されている `f s t r i m` 実行プログラム 545 を実行する。これにより、`C P U 1 4 1` は、`s a f e` ファーム格納領域 302 についての `f s t r i m` コマンドを生成する。また、`C P U 1 4 1` は、情報処理装置 100 がアップデートのために起動した状態において、`s a f e` ファーム格納領域 302 に記録されている `s t d` ファーム格納領域 301 のアップデート実行プログラム 548 を実行する。これにより、`C P U 1 4 1` は、`s t d` ファーム格納領域 301 のファームウェアプログラムをアップデートする。また、`C P U 1 4 1` は、情報処理装置 100 がアップデートのために起動した状態において、`s a f e` ファーム格納領域 302 に記録されている `f s t r i m` 実行プログラム 547 を実行する。これにより、`C P U 1 4 1` は、`s t d` ファーム格納領域 301 についての `f s t r i m` コマンドを

30

40

50

生成する。

【 0 0 2 3 】

図 6 は、図 1 の情報処理装置 1 0 0 についての、トリム処理を含むシャットダウン処理のフローチャートである。通常起動されている情報処理装置 1 0 0 の C P U 1 4 1 は、通常起動している情報処理装置 1 0 0 を終了する際に e M M C 1 4 7 の s t d ファーム格納領域 3 0 1 に格納されているシャットダウン実行プログラム 5 4 2 を実行し、図 6 のシャットダウン処理を実行する。ステップ S 6 0 1 において、C P U 1 4 1 は、電源ユニット 1 6 0 のシーソーススイッチの O F F によるシャットダウン指示、パーソナルコンピュータ 1 9 0 または外部サーバ 1 8 0 からのネットワーク経由でのシャットダウン指示、操作部 1 1 0 からのシャットダウン指示を取得する。これにより、C P U 1 4 1 は、e M M C 1 4 7 の s t d ファーム格納領域 3 0 1 に含まれるシャットダウン実行プログラム 5 4 2 によるシャットダウン処理を開始する。ステップ S 6 0 2 において、C P U 1 4 1 は、シャットダウン中を示す画面を操作部 1 1 0 に表示する。C P U 1 4 1 は、シャットダウン中の画面表示を操作部 1 1 0 へ指示する。操作部 1 1 0 は、シャットダウン用画面を表示する。また、操作部 1 1 0 は、ユーザの操作を受け付けられない状態になる。情報処理装置 1 0 0 は、情報処理装置 1 0 0 の処理のためにユーザ操作可能なユーザインタフェースに対するユーザ操作を受け付けられない状態となる。ステップ S 6 0 3 において、C P U 1 4 1 は、s t d ファーム格納領域 3 0 1 の複数のアプリケーションプログラム 5 4 1 について、シャットダウン処理を実行する。C P U 1 4 1 は、各アプリケーションプログラム 5 4 1 に対して終了を指示する。これにより、C P U 1 4 1 は、各アプリケーションプログラム 5 4 1 の終了処理を開始する。C P U 1 4 1 は、各アプリケーションプログラム 5 4 1 についての実行中のジョブをキャンセルし、使用していたリソースのデータを保存し、リソースを解放する。C P U 1 4 1 は、アプリケーションプログラム 5 4 1 についての終了処理を終え、終了をシャットダウン実行プログラム 5 4 2 に通知する。このステップ S 6 0 1 からステップ S 6 0 3 の処理により、C P U 1 4 1 は、ユーザ操作を受け付けられない状態となり、かつ、新規の印刷ジョブ、スキャンジョブ、F A X ジョブ等のジョブも受け付けられない状態となる。ステップ S 6 0 4 において、C P U 1 4 1 は、各アプリケーションの終了の通知を待ちながら、e M M C 1 4 7 についてマウントしている各領域のマウント設定を確認する。ステップ S 6 0 5 において、C P U 1 4 1 は、e M M C 1 4 7 の各領域について、書込可 (R W) でマウントされているか否かを判断する。これにより、制御部 1 4 0 としての C P U 1 4 1 は、フラッシュメモリ 2 0 5 に領域ごとの消去処理を指示するトリムコマンドを生成するために、フラッシュメモリ 2 0 5 の領域ごとの設定が書込可能であるか否かを判断する。判断に係る領域が書込可 (R W) でマウントされている場合、C P U 1 4 1 は、処理をステップ S 6 0 6 へ進める。判断に係る領域が書込可 (R W) でマウントされていない場合、C P U 1 4 1 は、処理をステップ S 6 0 7 へ進める。ステップ S 6 0 6 において、C P U 1 4 1 は、書込可 (R W) でマウントされている領域についての、f s t r i m 処理を指示する。C P U 1 4 1 は、書込可能と判断した領域について領域ごとの消去処理を指示する f s t r i m コマンドを生成し、e M M C 1 4 7 のメモリコントローラ 2 0 3 へ出力する。e M M C 1 4 7 のメモリコントローラ 2 0 3 は、フラッシュメモリ 2 0 5 についての指示された領域について、使用済みの不要な書込データを削除するトリム処理を実行する。トリム処理において、メモリコントローラ 2 0 3 は、たとえば、処理対象の領域から不要なデータを消去する。これにより、C P U 1 4 1 は、書込済みの領域に新たなデータを書込むためには書込処理とは別に領域ごとの消去処理を実行する必要があるフラッシュメモリ 2 0 5 についてのトリム処理を実行する。フラッシュメモリ 2 0 5 についての、書込可 (R W) でマウントされている領域は、使用済みのデータを記録している場合トリム処理により使用済みの不要なデータが削除され、新たなデータを書込むことができる状態となる。ステップ S 6 0 7 において、C P U 1 4 1 は、フラッシュメモリ 2 0 5 についてのマウントされているすべての領域についての設定確認が完了したか否かを判断する。マウントされているすべての領域についての設定確認が完了していない場合、C P U 1 4 1 は、処理をステップ S 6 0 4 へ戻す。C P U 1 4 1 は、フラッ

10

20

30

40

50

シュメモリ 205 についてのマウントされているすべての領域についての設定確認が完了するまで、領域ごとにマウント設定を確認して必要に応じて `fs trim` コマンドを生成する。すべての領域についての設定確認が完了すると、CPU 141 は、処理をステップ S608 へ進める。これにより、フラッシュメモリ 205 についての書込可 (RW) でマウントされている領域は、すべて、使用済みのデータが削除されて、新たなデータを書込むことができる状態になる。図 4 の通常起動の場合、複数のすべてのユーザデータ領域、複数のすべてのワーク領域 304 は、すべて、使用済みのデータが削除されて、新たなデータを書込むことができる状態となる。ステップ S608 において、CPU 141 は、すべてのアプリケーションプログラム 541 についてのシャットダウン処理がすべて終了しているか否かを判断する。CPU 141 は、たとえばすべてのアプリケーションプログラム 541 から終了が通知されているか否かを判断する。終了していないアプリケーションプログラム 541 が存在する場合、CPU 141 は、ステップ S608 の判断処理を繰り返し、そのシャットダウン処理の終了を待つ。終了していないアプリケーションプログラム 541 が存在しなくなってすべてのアプリケーションプログラム 541 が終了すると、CPU 141 は、処理をステップ S609 へ進める。ステップ S609 において、CPU 141 は、フラッシュメモリ 205 についてのマウントされている領域についてのすべてのトリム処理が完了しているか否かを判断する。たとえば、CPU 141 は、ステップ S606 を最初に実行する際に、`fs trim` 実行中を示すフラグを立てる。CPU 141 は、`fs trim` 実行プログラム 545 による処理がすべて終了すると、フラグを倒す。この場合、CPU 141 は、フラグが倒れていることに基づいて、すべてのトリム処理が完了していることを判断できる。すべてのトリム処理が完了していない場合、CPU 141 は、ステップ S609 の判断処理を繰り返し、その終了を待つ。すべてのトリム処理が完了している場合、CPU 141 は、処理をステップ S610 へ進める。ステップ S610 において、CPU 141 は、通常起動中にマウントしていたフラッシュメモリ 205 のすべての領域のアンマウント処理を実行する。ステップ S611 において、CPU 141 は、電源 OFF を、電源ユニット 160 に指示する。電源ユニット 160 は、情報処理装置 100 の各部への給電を停止する。

【0024】

このように図 6 の処理では、情報処理装置 100 をシャットダウンする処理において、ユーザインタフェースに対するユーザ操作を受け付けられないようにした後に、領域ごとの消去処理を指示するトリムコマンドを生成する。情報処理装置 100 は、ユーザインタフェースに対するユーザ操作を受け付けられない状態において、領域ごとの消去処理を指示するトリムコマンドを生成する。情報処理装置 100 は、情報処理装置 100 が処理するジョブが生じない状態であって、かつ、ユーザインタフェースに対するユーザ操作を受け付けられない状態において、トリムコマンドを生成することができる。これにより、情報処理装置 100 は、通常起動中のユーザ操作やジョブ実行が発生する可能性のあるタイミングを避けて、`fs trim` トリムコマンドを生成できる。eMMC 147 においてトリム処理が実行されている `fs trim` 処理中に、情報処理装置 100 においてジョブ実行やユーザ操作があると、それに対する eMMC 147 のレスポンスが遅延する。その結果、情報処理装置 100 のパフォーマンスが低下する。本実施形態では、このようなパフォーマンス低下を生じさせることなく、eMMC 147 にトリム処理を実行させることができる。しかも、本実施形態では、フラッシュメモリ 205 の領域ごとの消去処理を指示するトリムコマンドを生成するために、フラッシュメモリ 205 の領域ごとの設定が書込可能であるか否かを判断し、書込可能と判断した領域について領域ごとの消去処理を指示するトリムコマンドを生成する。フラッシュメモリ 205 の全体の領域に対して、トリムコマンドを一律に生成しない。これにより、本実施形態では、シャットダウン処理に追加されるフラッシュメモリ 205 からの不要なデータの消去処理の負荷および時間を軽減できる。フラッシュメモリ 205 の全体の領域に対してトリムコマンドを一律に生成する場合のように、シャットダウン処理が無用に長くなってしまうことはない。

【0025】

図7は、図1の情報処理装置100についての、トリム処理を含むファームウェアプログラム540のアップデート処理のフローチャートである。通常起動されている情報処理装置100のCPU141は、情報処理装置100のファームウェアプログラム540のアップデート処理のために、eMMC147のstdファーム格納領域301に格納されているアップデート用の各種の実行プログラムを実行する。これにより、CPU141は、図7のアップデート処理を実行する。ステップS701において、CPU141は、通常起動の状態において、外部サーバ180もしくはユーザの操作部110からの指示により、ファームウェアプログラム540のアップデート指示を取得する。CPU141は、eMMC147のstdファーム格納領域301に記録されているファームウェアプログラム540ダウンロード実行プログラム543に対してアップデートを指示する。これにより、CPU141は、eMMC147のstdファーム格納領域301に記録されているsafeファーム格納領域302のアップデート実行プログラム546を実行する。CPU141は、アップデート対象の新しいファームウェアプログラムの情報を取得し、eMMC147のファームウェア情報格納領域に保存する。ステップS702において、CPU141は、認証画面を、操作部110に表示する。ステップS703において、CPU141は、ユーザの認証を判断する。ユーザが認証されない場合、CPU141は、図7の処理を終了する。ユーザが認証されると、CPU141は、処理をステップS704へ進める。なお、ファームウェアのアップデートプログラムがユーザ認証を必要としないものである場合、CPU141は、ステップS701の後に、処理をステップS704へ進めてよい。ステップS704において、CPU141は、バックグラウンド処理によりファームウェアプログラム540についての新たなアップデートプログラムを外部サーバ180からダウンロードし、HDD146のダウンロード領域571に保存する。ダウンロードされた新たなアップデートプログラムには、eMMC147のsafeファーム格納領域302に格納される新しいバージョンのファームウェアプログラム540などが含まれる。ダウンロードされたアップデートプログラム、改ざん防止のために署名暗号化されている。ステップS705において、CPU141は、ダウンロード領域571にダウンロードされたアップデートプログラムを復号化し、一時展開領域572に展開する。これにより、最終的にeMMC147に展開して記録されるべきファームウェアプログラム540が、HDD146において予め展開される。このため、後に実施する情報処理装置100の再起動後の処理では、CPU141は、予めHDD146の一時展開領域572に展開されている新たなファームウェアプログラム540を、eMMC147の所定の領域へコピーするだけでよくなる。再起動後の処理において復号化、展開、および適用のすべてを行う場合と比べて、アップデートのためのeMMC147のダウンタイムを短縮できる。ステップS706において、CPU141は、ファームウェアプログラム540の適用時刻を確認する。ファームウェアプログラム540の適用時刻は、たとえば外部サーバ180から取得したり、ユーザの設定を取得したりすることにより、eMMC147に予め記録されている。ファームウェアプログラム540の適用時刻に到達していない場合、CPU141は、ステップS706の処理を繰り返す。ファームウェアプログラム540の適用時刻に到達すると、CPU141は、処理をステップS707へ進める。ステップS707において、CPU141は、認証画面を操作部110に表示する。ステップS708において、CPU141は、ユーザの認証を判断する。ユーザが認証されない場合、CPU141は、図7の処理を終了する。ユーザが認証されると、CPU141は、処理をステップS709へ進める。なお、ファームウェアプログラム540のアップデートについてユーザ認証が不要と設定されている場合、CPU141は、ステップS706の後に、処理をステップS709へ進めてよい。

【0026】

ステップS709において、CPU141は、アップデートのための再起動を指示する。CPU141は、たとえば図6のステップS601からステップS603およびステップS608からステップS610の処理を実行し、その後に情報処理装置100をアップデートのために再起動する。アップデートのための再起動では、CPU141は、safe

10

20

30

40

50

eファーム格納領域302のファームウェアプログラムを実行する。CPU141は、safeファーム格納領域302に記録されているstdファーム格納領域301のアップデート実行プログラム548を実行する。safeファーム格納領域302には、stdファーム格納領域301のアップデート実行プログラム548と、fstrim実行プログラム547と、が含まれ、アプリケーションプログラムなどが含まれない。このため、CPU141は、アップデートのための再起動により、情報処理装置100に対するユーザ操作を受け付けることができない状態となり、かつ、印刷ジョブ、スキャンジョブ、FAXジョブ等のジョブも受け付けることができない状態となる。アップデートのために再起動されているステップS710において、CPU141は、図4に基づいて、stdファーム格納領域301を通常起動とは異なり書込可(RW)の状態のマウントする。CPU141は、図4に基づいて、safeファーム格納領域302を通常起動とは異なり書込不可(RO)の状態のマウントする。ステップS711において、CPU141は、一時展開領域572に展開済の新しいファームウェアプログラムを、書込可(RW)のstdファーム格納領域301にコピーする。CPU141は、展開済の新しいファームウェアプログラムを、HDD146から読み込んでeMMC147へ出力する。eMMC147のメモリコントローラ203は、新しいファームウェアプログラムを、フラッシュメモリ205のstdファーム格納領域301に書き込む。これにより、stdファーム格納領域301のファームウェアプログラムは、新しいものに差し替えられてアップデートされる。ステップS712において、CPU141は、書込可(RW)でマウントされているアップデート後のstdファーム格納領域301について、fstrim処理を指示する。CPU141は、safeファーム格納領域302のfstrim実行プログラム547を実行して、fstrimコマンドを生成し、eMMC147のメモリコントローラ203へ出力する。CPU141は、書込済みの領域に新たなデータを書込むためには書込処理とは別に領域ごとの消去処理を実行する必要があるフラッシュメモリ205についてのトリム処理を指示する。eMMC147のメモリコントローラ203は、フラッシュメモリ205についての指示されたstdファーム格納領域301について、アップデートにより不要となったデータを削除するトリム処理を実行する。トリム処理において、メモリコントローラ203は、たとえば、処理対象の領域から不要なデータを消去する。これにより、フラッシュメモリ205についての、書込可(RW)でマウントされているstdファーム格納領域301は、使用済みのデータが削除され、新たなデータを書込むことができる状態になる。stdファーム格納領域301は、ユーザインタフェースに対するユーザ操作を受け付けない状態において、かつ、ジョブを受け付けることができない状態において、トリムコマンドに基づいて不要なデータが消去される。ファームウェアプログラム540更新による書込が行われた後にトリム処理を行うことで、eMMC147のパフォーマンス劣化の抑止に寄与することができる。fstrim実行プログラム547を実行するCPU141は、トリム処理が終了すると、トリム処理の終了をstdファームウェアのアップデートプログラムに通知する。ステップS713において、CPU141は、stdファーム格納領域301をアンマウントする。また、CPU141は、safeファーム格納領域302をアンマウントする。

【0027】

ステップS714において、CPU141は、情報処理装置100を、通常起動により再起動する。通常起動による再起動では、CPU141は、stdファーム格納領域301のファームウェアプログラムを実行する。通常起動により再起動されているステップS715において、CPU141は、stdファーム格納領域301に記録されているファームウェアプログラム540のアップデート実行プログラムを実行する。ステップS716において、CPU141は、一時展開領域572に展開済の新しいファームウェアプログラムを、書込可(RW)のsafeファーム格納領域302にコピーする。CPU141は、展開済の新しいファームウェアプログラムを、HDD146から読み込んでeMMC147へ出力する。eMMC147のメモリコントローラ203は、新しいファームウェアプログラムを、フラッシュメモリ205のsafeファーム格納領域302に書き込

10

20

30

40

50

む。これにより、`safe`ファーム格納領域302のファームウェアプログラムは、新しいものに差し替えられてアップデートされる。通常起動時は、`safe`ファーム格納領域302に格納されるファームウェアプログラム540はCPU141の実行対象とならない。したがって、情報処理装置100の動作に影響を与えないように、`safe`ファーム格納領域302のファームウェアプログラム540をアップデートできる。なお、CPU141は、ここでさらに、書込可(RW)でマウントされている`safe`ファーム格納領域302について、`trim`処理を指示してよい。CPU141は、`std`ファーム格納領域301の`trim`実行プログラム545を実行して、`trim`コマンドを生成し、eMMC147のメモリコントローラ203へ出力する。eMMC147のメモリコントローラ203は、フラッシュメモリ205についての指示された`safe`ファーム格納領域302について、アップデートにより不要となったデータを削除するトリム処理を実行する。また、図7では、CPU141は、通常起動時にバックグラウンドで実行される`safe`ファーム格納領域302のアップデート実行プログラム546により、HDD146の一時展開領域572に展開済の新しいファームウェアプログラムを、`safe`ファーム格納領域302にコピーしている。この他にもたとえば、CPU141は、`safe`ファーム格納領域302のアップデート実行プログラム546の実行により、ダウンロードされた新たなファームウェアプログラム540を復号し、展開し、`safe`ファーム格納領域302に記録してもよい。

【0028】

このように図7の処理では、CPU141は、情報処理装置100のファームウェアプログラム540をアップデートする処理において、アップデートされる領域についての領域ごとの消去処理を指示するトリムコマンドを生成する。CPU141は、ユーザインタフェースに対するユーザ操作を受け付けられない状態において、アップデートされる領域についての領域ごとの消去処理を指示するトリムコマンドを生成する。CPU141は、情報処理装置100のアップデートのための再起動を開始した後に、フラッシュメモリ205についてのアップデートに係る`std`ファーム格納領域301を書込可能な領域としてマウントしてアップデートを実行する。CPU141は、さらにトリムコマンドを生成してアップデートにより不要となるデータの消去処理を指示する。なお、ステップS701、ステップS703において、外部サーバ180からファームウェアプログラム540を取得する例を説明したが、新しいファームウェアプログラムの取得方法は他の方法であってもかまわない。たとえば、USBメモリやパーソナルコンピュータ190にファームウェアプログラム540を格納し、デバイスに接続することでも同様の処理を実施することができる。また、ファームウェアプログラムのアップデート方法として、本実施形態ではバックグラウンドであらかじめ新ファームウェアプログラム540を通常起動中に一時展開しておく例を説明したが、新ファームウェアプログラム540の展開は`safe`ファーム格納領域302によるアップデート用起動処理中に実施してもよい。

【0029】

そして、本実施形態では、情報処理装置100についての再起動を伴うアップデート処理において、フラッシュメモリ205へのアップデート処理に用いるプログラムのみを実行する状態で情報処理装置100を再起動する処理を実行する。その後、フラッシュメモリ205に領域ごとの消去処理を指示する消去コマンドとしてのトリムコマンドを生成する。このようにアップデート処理での再起動後にトリムコマンドを生成するようにすることで、トリムコマンドは、情報処理装置100が処理するジョブが生じない状態であって、かつ、ユーザインタフェースに対するユーザ操作を受け付けられない状態において生成され得る。しかも、本実施形態では、フラッシュメモリ205に領域ごとの消去処理を指示するトリムコマンドを生成するために、情報処理装置100のアップデート処理での再起動を開始した後に、フラッシュメモリ205についてのアップデートに係る記録領域を書込可能な領域としてマウントし、アップデートデータを記録領域にコピーして差し替え、記録領域について領域ごとの消去処理を指示するトリムコマンドを生成し、記録領域のアップデートデータによる再起動処理を実行する。これにより、アップデートに係る記録領域

10

20

30

40

50

は、トリムコマンドにより不要なデータが削除された状態で、アップデートデータによる再起動に用いられる。本実施形態では、アップデート処理に追加されるフラッシュメモリ 205 からの不要なデータの消去処理の負荷および時間を軽減して、アップデート処理が無用に長くなってしまふことを抑制できる。特に、本実施形態では、上述したシャットダウン時の `fs trim` 処理実行と合わせて、ファームウェアプログラムのアップデート時に `std` ファーム格納領域 301 にも `fs trim` 処理実行を行う。このため、`eMMC 147` のシステム領域 300 には、一通りのトリム処理を実行することができる。また、本実施形態では、ファームウェアプログラムのアップデート時の `std` ファーム格納領域 301 への `fs trim` 処理によって、`eMMC 147` のファームウェアプログラムのアップデート時にできた不要な領域を即座に `eMMC 147` に通知できる。このため、通常起動中に実施するシャットダウン処理中での `fs trim` 処理自体の処理時間の短縮も見込まれる。

10

【0030】

以上のように、本実施形態では、ユーザインタフェースに対するユーザ操作を受け付けない状態において、フラッシュメモリ 205 に領域ごとの消去処理を指示するトリムコマンドを生成する。具体的にはたとえば、情報処理装置 100 が処理するジョブが生じない状態であって且つユーザインタフェースに対するユーザ操作を受け付けない状態において、フラッシュメモリ 205 に領域ごとの消去処理を指示するトリムコマンドを生成する。これにより、情報処理装置 100 は、ユーザがユーザインタフェースに対して操作をする際には、書込済みの領域に新たなデータを書込むためには書込処理とは別に領域ごとの消去処理を実行する必要があるフラッシュメモリ 205 について、領域ごとの消去処理を実行させないようにできる。情報処理装置 100 においてフラッシュメモリ 205 は、情報処理装置 100 に対してユーザが操作する際には領域ごとの消去処理を実行しない。このため、ユーザは、不揮発性半導体メモリ装置にアクセスできないことに起因する遅延を直接的に体感したり、動作速度の低下を直接的に体感したりし難くなる。このように情報処理装置 100 では、書込済みの領域に新たなデータを書込むためには書込処理とは別に領域ごとの消去処理を実行する必要があるフラッシュメモリ 205 を用いつつ、ユーザによる操作性を改善することができる。しかも、本実施形態では、フラッシュメモリ 205 の領域ごとの設定に基づいて書込可能と判断できる領域について、領域ごとの消去処理を指示するトリムコマンドを生成する。よって、フラッシュメモリ 205 についての、書込不可と判断される領域については、領域ごとの消去処理を実行させない。フラッシュメモリ 205 の全体についての領域ごとの消去処理の負荷および時間を、フラッシュメモリ 205 の領域ごとの設定に基づいて削減できる。本実施形態では、情報処理装置 100 において書込済みの領域に新たなデータを書込むためには書込処理とは別に領域ごとの消去処理を実行する必要があるフラッシュメモリ 205 を用いつつ、ユーザによる操作性を改善することができる。

20

30

【0031】

以上、本発明をその好適な実施形態に基づいて詳述してきたが、本発明はこれら特定の実施形態に限られるものではなく、この発明の要旨を逸脱しない範囲の様々な形態も本発明に含まれる。

40

【0032】

本発明は、上述の実施の形態の 1 以上の機能を実現するプログラムを、ネットワークや記憶媒体を介してシステムや装置に供給し、そのシステム又は装置のコンピュータの 1 つ以上のプロセッサがプログラムを読み出して実行する処理でも実現可能である。また、本発明は、1 以上の機能を実現する回路（例えば、ASIC）によっても実現可能である。

【0033】

上記実施形態では、ファームウェアプログラム 540 をアップデートする場合に、CPU 141 は、ファームウェアプログラム 540 を記録する `std` ファーム格納領域 301 についてのみ、トリム処理を実行している。この他にもたとえば、ファームウェアプログラム 540 をアップデートする場合に、CPU 141 は、ファームウェアプログラム 54

50

0 を記録する s t d ファーム格納領域 3 0 1 とは別の他の領域についても、トリム処理を実行してもよい。この場合、C P U 1 4 1 は、フラッシュメモリ 2 0 5 に領域ごとの消去処理を指示するトリムコマンドを生成するために、情報処理装置 1 0 0 のアップデート処理での再起動を開始した後に、メモリについての使用済のデータが書き込まれている他の領域を書込可能な領域としてマウントしてよい。たとえば図 4 のマウント構成において、アップデートのための起動モード 4 0 2 のときに複数のユーザデータ格納領域 3 0 3 および複数のワーク領域 3 0 4 を書込可 (R W) であえてマウントしてよい。そして、C P U 1 4 1 は、これらの領域を領域ごとの設定に基づいて書込可能と判断し、図 7 の S 7 1 2 において領域ごとの消去処理を指示するトリムコマンドを生成する。これにより、フラッシュメモリ 2 0 5 についての、アップデートに係る旧データを記録している s t d ファーム格納領域 3 0 1 以外の領域についても、不使用のデータが書き込まれている領域については、アップデート処理中に不要なデータを削除することができる。フラッシュメモリ 2 0 5 についての、書込可 (R W) でマウントされている領域は、使用済みのデータを記録している場合トリム処理により使用済みの不要なデータが削除され、新たなデータを書込むことができる状態となる。

10

【符号の説明】

【 0 0 3 4 】

- 1 0 0 情報処理装置
- 1 1 0 操作部
- 1 4 0 制御部
- 1 4 1 C P U
- 1 4 7 e M M C
- 2 0 2 バスコントローラ
- 2 0 3 メモリコントローラ
- 2 0 4 キャッシュメモリ
- 2 0 5 フラッシュメモリ
- 3 0 1 s t d ファーム格納領域
- 3 0 2 s a f e ファーム格納領域
- 4 0 1 通常起動モード
- 4 0 2 起動モード
- 5 4 0 ファームウェアプログラム
- 5 4 2 シャットダウン実行プログラム
- 5 4 5 f s t r i m 実行プログラム
- 5 4 6 アップデート実行プログラム
- 5 4 7 f s t r i m 実行プログラム
- 5 4 8 アップデート実行プログラム

20

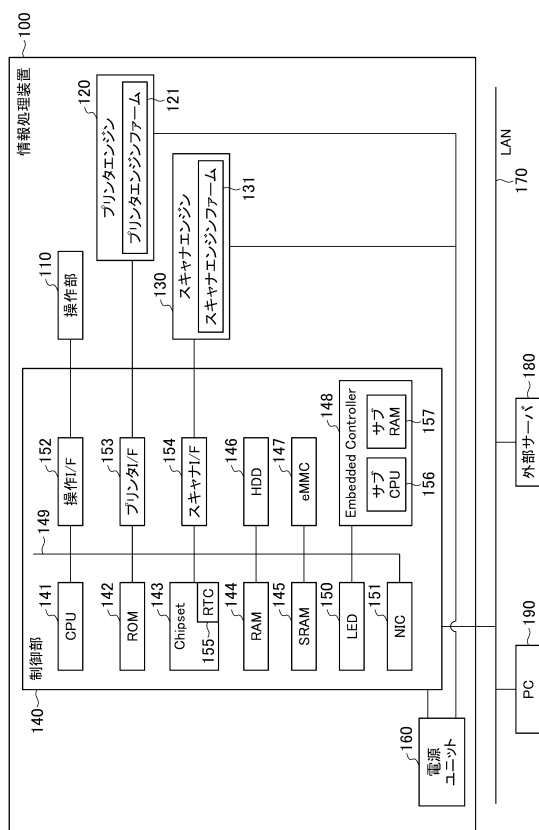
30

40

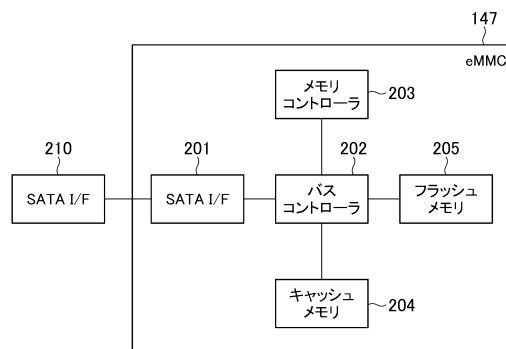
50

【図面】

【圖 1】



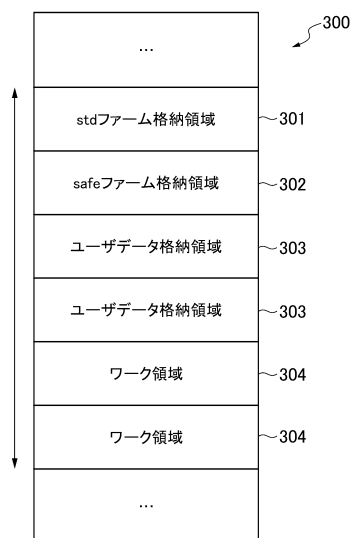
【圖 2】



10

20

【 図 3 】



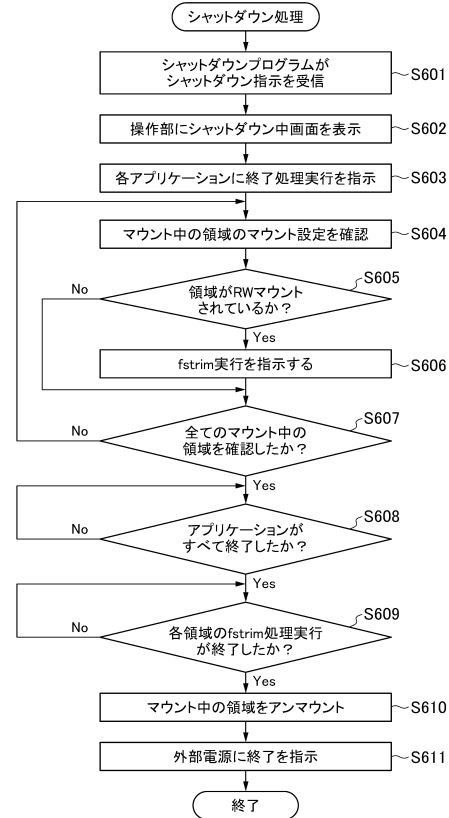
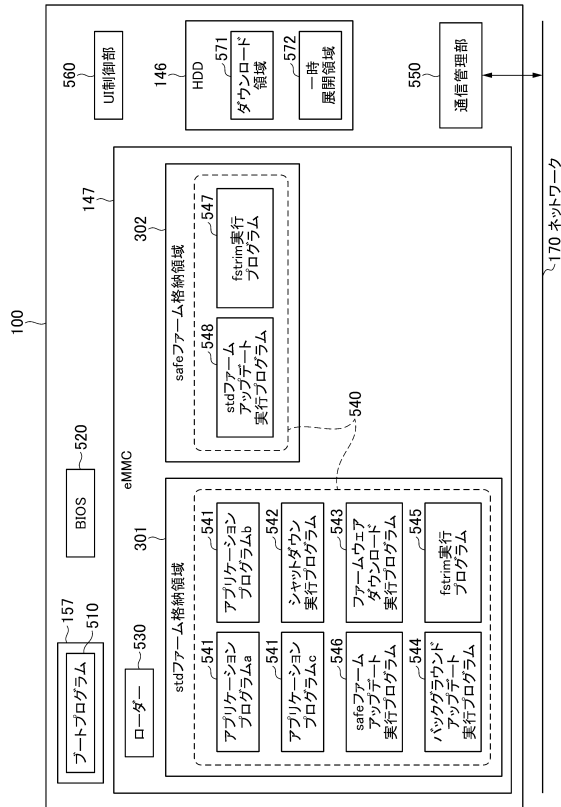
【圖 4】

	401	402
パーティション	通常起動モード	アップデート用起動モード
stdファーム格納領域	読み込みのみ (RO)	読み書き可 (RW)
safeファーム格納領域	読み書き可 (RW)	読み込みのみ (RO)
ユーザデータ格納領域a	読み書き可 (RW)	-
ユーザデータ格納領域b	読み書き可 (RW)	-
ワーク領域a	読み書き可 (RW)	-
ワーク領域b	読み書き可 (RW)	-

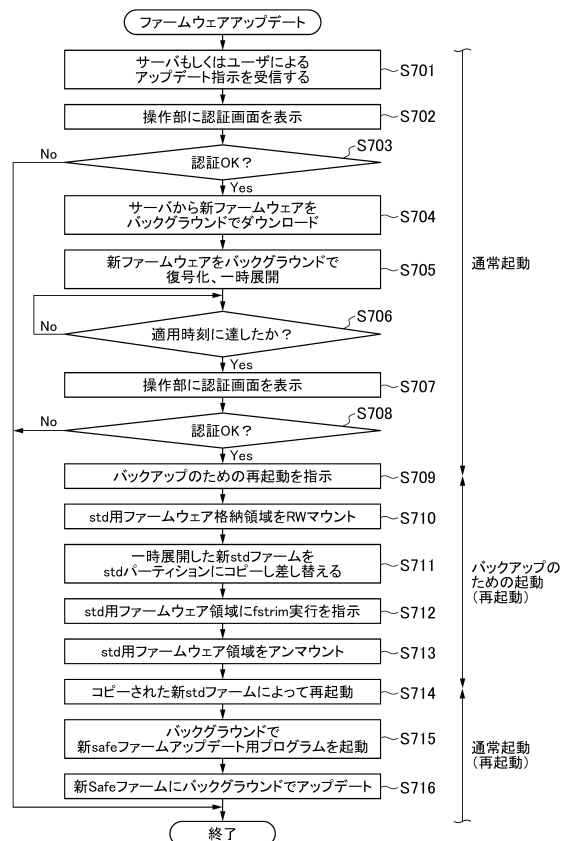
30

40

【 図 6 】



【圖 7】



10

20

30

40

50

フロントページの続き

- (56)参考文献 特開 2 0 0 4 - 2 8 7 5 4 1 (J P , A)
特開 2 0 0 7 - 0 6 9 3 6 8 (J P , A)
特開 2 0 0 9 - 2 2 6 6 9 0 (J P , A)

- (58)調査した分野 (Int.Cl. , D B 名)
G 0 6 F 3 / 0 6 - 3 / 0 8
8 / 0 0 - 8 / 3 8
8 / 6 0 - 8 / 7 7
9 / 4 4 - 9 / 4 4 5
9 / 4 5 1
1 2 / 0 0 - 1 2 / 1 2 8
1 3 / 1 6 - 1 3 / 1 8
G 1 1 C 2 9 / 0 0 - 2 9 / 5 6