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

特許第6157637号
(P6157637)

(45)発行日　平成29年7月5日(2017.7.5)　(24)登録日　平成29年6月16日(2017.6.16)

(51)Int.Cl.		F I		
G06F 12/16	(2006.01)	G06F	12/16	310R
G06F 12/00	(2006.01)	G06F	12/00	597U
G06F 12/06	(2006.01)	G06F	12/06	520F

請求項の数 68　(全 33 頁)

(21)出願番号　特願2015-549390 (P2015-549390)
(86)(22)出願日　平成25年11月7日 (2013.11.7)
(65)公表番号　特表2016-501417 (P2016-501417A)
(43)公表日　平成28年1月18日 (2016.1.18)
(86)国際出願番号　PCT/US2013/068939
(87)国際公開番号　WO2014/099169
(87)国際公開日　平成26年6月26日 (2014.6.26)
審査請求日　平成28年11月4日 (2016.11.4)
(31)優先権主張番号　13/720,532
(32)優先日　平成24年12月19日 (2012.12.19)
(33)優先権主張国　米国(US)

早期審査対象出願

(73)特許権者　507364838
クアルコム，インコーポレイテッド
アメリカ合衆国　カリフォルニア　９２１２１　サン　ディエゴ　モアハウス　ドライブ　５７７５
(74)代理人　100108453
弁理士　村山　靖彦
(74)代理人　100163522
弁理士　黒田　晋平
(72)発明者　ダミン・パッカー・アリ
アメリカ合衆国・カリフォルニア・９２１２１・サン・ディエゴ・モアハウス・ドライヴ・５７７５

最終頁に続く

(54)【発明の名称】リードライトメモリデバイスのデータイメージ中の仮想境界コード

(57)【特許請求の範囲】
【請求項１】
リードライトメモリデバイス上でデータイメージを記憶する方法であって、
第1のリードライトメモリデバイスおよび第2のリードライトメモリデバイスの両方によって指定された、一連の仮想ブロックに共通している仮想ブロックサイズを設定するステップであって、前記仮想ブロックサイズが、前記第1のリードライトメモリデバイスの一連の実ブロックの各々の第1のサイズおよび前記第2のリードライトメモリデバイスの一連の実ブロックの各々の第2のサイズに基づいており、前記第1のサイズと前記第2のサイズとが異なり、前記第1のリードライトメモリデバイスの前記一連の実ブロックの各々および前記第2のリードライトメモリデバイスの前記一連の実ブロックの各々が複数のページを含み、各仮想ブロックが、前記第1のリードライトメモリデバイスの前記実ブロックの前記複数のページまたは前記第2のリードライトメモリデバイスの前記実ブロックの前記複数のページのうちの複数のページを含む、ステップと、
前記仮想ブロックサイズに基づいてブートローダを第1のブートローダ部分および第2のブートローダ部分に分割するステップであって、前記第1のブートローダ部分と前記第2のブートローダ部分とが異なり、前記ブートローダのサイズが前記仮想ブロックサイズより大きく、前記ブートローダの前記サイズが前記第1のサイズまたは前記第2のサイズのうちの1つまたは両方より大きい、ステップと、
前記第1のブートローダ部分に仮想境界コードを付加するステップであって、前記仮想境界コードが、前記第1のブートローダ部分を位置特定するためのマーカを表す、ステッ

プと、
前記仮想境界コードが付加された前記第1のブートローダ部分を、前記第1のリードライトメモリデバイス上の、前記一連の仮想ブロックのうちのある仮想ブロック中に記憶するステップとを含む、方法。
【請求項２】
前記仮想境界コードが、前記第1のブートローダ部分の始端において付加される、請求項1に記載の方法。
【請求項３】
前記第1のサイズおよび前記第2のサイズの両方が、剰余なしで、前記仮想ブロックサイズで割り切れる、請求項1に記載の方法。
【請求項４】
前記第1のリードライトメモリデバイス上の前記一連の仮想ブロックのうちの別の仮想ブロック中に前記第2のブートローダ部分を記憶するステップをさらに含む、請求項1に記載の方法。
【請求項５】
前記第1のリードライトメモリデバイス内の不良ブロックが、前記第1のブートローダ部分と前記第2のブートローダ部分との間で位置特定される、請求項4に記載の方法。
【請求項６】
前記第1のブートローダ部分および前記第2のブートローダ部分が、前記第1のリードライトメモリデバイス上の前記一連の実ブロックのうちの共有実ブロック中に記憶される、請求項4に記載の方法。
【請求項７】
各仮想ブロックが、同じ仮想境界コードを含む、請求項1に記載の方法。
【請求項８】
前記第2のブートローダ部分が、前記第1のリードライトメモリデバイス上の前記仮想境界コードの位置、および前記ブートローダの前記サイズのうちの少なくとも1つに関する情報を有するイメージヘッダを含む、請求項4に記載の方法。
【請求項９】
第3のブートローダ部分に前記仮想境界コードを付加するステップと、
前記仮想境界コードが付加された前記第3のブートローダ部分を、前記一連の仮想ブロックのうちの第3の仮想ブロック中に記憶するステップとをさらに含む、請求項4に記載の方法。
【請求項１０】
前記仮想境界コードが付加された前記第1のブートローダ部分を、前記第2のリードライトメモリデバイス上の仮想ブロック中に記憶するステップであって、ページサイズおよびデバイス固有の不良ブロックアルゴリズム特性のうちの少なくとも1つが、前記第1のリードライトメモリデバイスと前記第2のリードライトメモリデバイスとの間で異なる、ステップをさらに含む、請求項1に記載の方法。
【請求項１１】
リードライトメモリデバイス上にデータイメージを記憶するためのコンピューティングデバイスであって、
メモリと、
前記メモリに結合され、動作を実施するためのプロセッサ実行可能命令によって構成されたプロセッサとを備え、前記動作が、
第1のリードライトメモリデバイスおよび第2のリードライトメモリデバイスの両方によって指定された、一連の仮想ブロックに共通している仮想ブロックサイズを設定することであって、前記仮想ブロックサイズが、前記第1のリードライトメモリデバイスの一連の実ブロックの各々の第1のサイズおよび前記第2のリードライトメモリデバイスの一連の実ブロックの各々の第2のサイズに基づいており、前記第1のサイズと前記第2のサイズとが異なり、前記第1のリードライトメモリデバイスの前記一連の実ブロックの各々および前

記第2のリードライトメモリデバイスの前記一連の実ブロックの各々が複数のページを含み、各仮想ブロックが、前記第1のリードライトメモリデバイスの前記実ブロックの前記複数のページまたは前記第2のリードライトメモリデバイスの前記実ブロックの前記複数のページのうちの複数のページを含む、ことと、

前記仮想ブロックサイズに基づいてブートローダを第1のブートローダ部分および第2のブートローダ部分に分割することであって、前記第1のブートローダ部分と前記第2のブートローダ部分とが異なり、前記ブートローダのサイズが前記仮想ブロックサイズより大きく、前記ブートローダの前記サイズが前記第1のサイズまたは前記第2のサイズのうちの1つまたは両方より大きい、ことと、

前記第1のブートローダ部分に仮想境界コードを付加することであって、前記仮想境界コードが、前記第1のブートローダ部分を位置特定するためのマーカを表す、ことと、

前記仮想境界コードが付加された前記第1のブートローダ部分を、前記第1のリードライトメモリデバイス上の、前記一連の仮想ブロックのうちのある仮想ブロック中に記憶することとを含む、コンピューティングデバイス。

【請求項１２】

前記プロセッサが、前記仮想境界コードが前記第1のブートローダ部分の始端において付加されるような動作を実施するためのプロセッサ実行可能命令によって構成される、請求項11に記載のコンピューティングデバイス。

【請求項１３】

前記第1のサイズおよび前記第2のサイズの両方が、剰余なしで、前記仮想ブロックサイズで割り切れるような動作を実施するためのプロセッサ実行可能命令によって構成される、請求項11に記載のコンピューティングデバイス。

【請求項１４】

前記プロセッサが、

前記第1のリードライトメモリデバイス上の前記一連の仮想ブロックのうちの別の仮想ブロック中に前記第2のブートローダ部分を記憶することをさらに含む動作を実施するためのプロセッサ実行可能命令によって構成される、請求項11に記載のコンピューティングデバイス。

【請求項１５】

前記プロセッサが、前記第1のリードライトメモリデバイス内の不良ブロックが前記第1のブートローダ部分と前記第2のブートローダ部分との間に位置特定されるような動作を実施するためのプロセッサ実行可能命令によって構成される、請求項14に記載のコンピューティングデバイス。

【請求項１６】

前記プロセッサが、前記第1のブートローダ部分および前記第2のブートローダ部分が、前記第1のリードライトメモリデバイス上の前記一連の実ブロックのうちの共有実ブロック中に記憶されるような動作を実施するためのプロセッサ実行可能命令によって構成される、請求項14に記載のコンピューティングデバイス。

【請求項１７】

各仮想ブロックが、同じ仮想境界コードを含む、請求項11に記載のコンピューティングデバイス。

【請求項１８】

前記プロセッサが、前記第2のブートローダ部分が前記第1のリードライトメモリデバイス上の前記仮想境界コードの位置、および前記ブートローダの前記サイズのうちの少なくとも1つに関する情報を有するイメージヘッダを含むような動作を実施するためのプロセッサ実行可能命令によって構成される、請求項14に記載のコンピューティングデバイス。

【請求項１９】

前記プロセッサが、

第3のブートローダ部分に前記仮想境界コードを付加することと、

前記仮想境界コードが付加された前記第3のブートローダ部分を、前記一連の仮想ブロ

ックのうちの第3の仮想ブロック中に記憶することとをさらに含む動作を実施するためのプロセッサ実行可能命令によって構成される、請求項14に記載のコンピューティングデバイス。

【請求項２０】

前記プロセッサが、

前記仮想境界コードが付加された前記第1のブートローダ部分を、前記第2のリードライトメモリデバイス上の仮想ブロック中に記憶することであって、ページサイズおよびデバイス固有の不良ブロックアルゴリズム特性のうちの少なくとも1つが、前記第1のリードライトメモリデバイスと前記第2のリードライトメモリデバイスとの間で異なる、ことをさらに含む動作を実施するためのプロセッサ実行可能命令によって構成される、請求項11に記載のコンピューティングデバイス。

【請求項２１】

リードライトメモリデバイス上にデータイメージを記憶するためのコンピューティングデバイスであって、

第1のリードライトメモリデバイスおよび第2のリードライトメモリデバイスの両方によって指定された、一連の仮想ブロックに共通している仮想ブロックサイズを設定するための手段であって、前記仮想ブロックサイズが、前記第1のリードライトメモリデバイスの一連の実ブロックの各々の第1のサイズおよび前記第2のリードライトメモリデバイスの一連の実ブロックの各々の第2のサイズに基づいており、前記第1のサイズと前記第2のサイズとが異なり、前記第1のリードライトメモリデバイスの前記一連の実ブロックの各々および前記第2のリードライトメモリデバイスの前記一連の実ブロックの各々が複数のページを含み、各仮想ブロックが、前記第1のリードライトメモリデバイスの前記実ブロックの前記複数のページまたは前記第2のリードライトメモリデバイスの前記実ブロックの前記複数のページのうちの複数のページを含む、手段と、

前記仮想ブロックサイズに基づいてブートローダを第1のブートローダ部分および第2のブートローダ部分に分割するための手段であって、前記第1のブートローダ部分と前記第2のブートローダ部分とが異なり、前記ブートローダのサイズが前記仮想ブロックサイズより大きく、前記ブートローダの前記サイズが前記第1のサイズまたは前記第2のサイズのうちの1つまたは両方より大きい、手段と、

前記第1のブートローダ部分に仮想境界コードを付加するための手段であって、前記仮想境界コードが、前記第1のブートローダ部分を位置特定するためのマーカを表す、手段と、

前記仮想境界コードが付加された前記第1のブートローダ部分を、前記第1のリードライトメモリデバイス上の、前記一連の仮想ブロックのうちのある仮想ブロック中に記憶するための手段とを備えるコンピューティングデバイス。

【請求項２２】

前記仮想境界コードが、前記第1のブートローダ部分の始端において付加される、請求項21に記載のコンピューティングデバイス。

【請求項２３】

前記第1のサイズおよび前記第2のサイズの両方が、剰余なしで、前記仮想ブロックサイズで割り切れる、請求項21に記載のコンピューティングデバイス。

【請求項２４】

前記第1のリードライトメモリデバイス上の前記一連の仮想ブロックのうちの別の仮想ブロック中に前記第2のブートローダ部分を記憶するための手段をさらに備える、請求項21に記載のコンピューティングデバイス。

【請求項２５】

前記第1のリードライトメモリデバイス内の不良ブロックが、前記第1のブートローダ部分と前記第2のブートローダ部分との間で位置特定される、請求項24に記載のコンピューティングデバイス。

【請求項２６】

前記第1のブートローダ部分および前記第2のブートローダ部分が、前記第1のリードライトメモリデバイス上の前記一連の実ブロックのうちの共有実ブロック中に記憶される、請求項24に記載のコンピューティングデバイス。

【請求項２７】

各仮想ブロックが、同じ仮想境界コードを含む、請求項21に記載のコンピューティングデバイス。

【請求項２８】

前記第2のブートローダ部分が、前記第1のリードライトメモリデバイス上の前記仮想境界コードの位置、および前記ブートローダの前記サイズのうちの少なくとも1つに関する情報を有するイメージヘッダを含む、請求項24に記載のコンピューティングデバイス。

【請求項２９】

第3のブートローダ部分に前記仮想境界コードを付加するための手段と、

前記仮想境界コードが付加された前記第3のブートローダ部分を、前記一連の仮想ブロックのうちの第3の仮想ブロック中に記憶するための手段とをさらに備える、請求項24に記載のコンピューティングデバイス。

【請求項３０】

前記仮想境界コードが付加された前記第1のブートローダ部分を、前記第2のリードライトメモリデバイス上の仮想ブロック中に記憶するための手段であって、ページサイズおよびデバイス固有の不良ブロックアルゴリズム特性のうちの少なくとも1つが、前記第1のリードライトメモリデバイスと前記第2のリードライトメモリデバイスとの間で異なる、手段をさらに備える、請求項21に記載のコンピューティングデバイス。

【請求項３１】

メモリを有するコンピューティングデバイス上のメモリを管理するための動作をプロセッサに実施させるように構成されたプロセッサ実行可能ソフトウェア命令を記憶した非一時的コンピュータ可読記憶媒体であって、前記動作が、

第1のリードライトメモリデバイスおよび第2のリードライトメモリデバイスの両方によって指定された、一連の仮想ブロックに共通している仮想ブロックサイズを設定することであって、前記仮想ブロックサイズが、前記第1のリードライトメモリデバイスの一連の実ブロックの各々の第1のサイズおよび前記第2のリードライトメモリデバイスの一連の実ブロックの各々の第2のサイズに基づいており、前記第1のサイズと前記第2のサイズとが異なり、前記第1のリードライトメモリデバイスの前記一連の実ブロックの各々および前記第2のリードライトメモリデバイスの前記一連の実ブロックの各々が複数のページを含み、各仮想ブロックが、前記第1のリードライトメモリデバイスの前記実ブロックの前記複数のページまたは前記第2のリードライトメモリデバイスの前記実ブロックの前記複数のページのうちの複数のページを含む、ことと、

前記仮想ブロックサイズに基づいてブートローダを第1のブートローダ部分および第2のブートローダ部分に分割することであって、前記第1のブートローダ部分と前記第2のブートローダ部分とが異なり、前記ブートローダのサイズが前記仮想ブロックサイズより大きく、前記ブートローダの前記サイズが前記第1のサイズまたは前記第2のサイズのうちの1つまたは両方より大きい、ことと、

前記第1のブートローダ部分に仮想境界コードを付加することであって、前記仮想境界コードが、前記第1のブートローダ部分を位置特定するためのマーカを表す、ことと、

前記仮想境界コードが付加された前記第1のブートローダ部分を、前記第1のリードライトメモリデバイス上の、前記一連の仮想ブロックのうちのある仮想ブロック中に記憶することとを含む、非一時的コンピュータ可読記憶媒体。

【請求項３２】

前記記憶されたプロセッサ実行可能命令が、前記仮想境界コードが、前記第1のブートローダ部分の始端において付加されるような動作をプロセッサに実施させるように構成される、請求項31に記載の非一時的コンピュータ可読記憶媒体。

【請求項３３】

前記記憶されたプロセッサ実行可能命令が、前記第1のサイズおよび前記第2のサイズの両方が、剰余なしで、前記仮想ブロックサイズで割り切れるような動作をプロセッサに実施させるように構成される、請求項31に記載の非一時的コンピュータ可読記憶媒体。

【請求項３４】

前記記憶されたプロセッサ実行可能ソフトウェア命令が、

前記第1のリードライトメモリデバイス上の前記一連の仮想ブロックのうちの別の仮想ブロック中に前記第2のブートローダ部分を記憶することをさらに含む動作をプロセッサに実施させるように構成される、請求項31に記載の非一時的コンピュータ可読記憶媒体。

【請求項３５】

前記記憶されたプロセッサ実行可能命令が、前記第1のリードライトメモリデバイス内の不良ブロックが前記第1のブートローダ部分と前記第2のブートローダ部分との間に位置特定されるような動作をプロセッサに実施させるように構成される、請求項34に記載の非一時的コンピュータ可読記憶媒体。

【請求項３６】

前記記憶されたプロセッサ実行可能命令が、前記第1のブートローダ部分および前記第2のブートローダ部分が、前記第1のリードライトメモリデバイス上の前記一連の実ブロックのうちの共有実ブロック中に記憶されるような動作をプロセッサに実施させるように構成される、請求項34に記載の非一時的コンピュータ可読記憶媒体。

【請求項３７】

各仮想ブロックが、同じ仮想境界コードを含む、請求項31に記載の非一時的コンピュータ可読記憶媒体。

【請求項３８】

前記記憶されたプロセッサ実行可能命令が、前記第2のブートローダ部分が、前記第1のリードライトメモリデバイス上の前記仮想境界コードの位置、および前記ブートローダの前記サイズのうちの少なくとも1つに関する情報を有するイメージヘッダを含むような動作をプロセッサに実施させるように構成される、請求項34に記載の非一時的コンピュータ可読記憶媒体。

【請求項３９】

前記記憶されたプロセッサ実行可能ソフトウェア命令が、

第3のブートローダ部分に前記仮想境界コードを付加することと、

前記仮想境界コードが付加された前記第3のブートローダ部分を、前記一連の仮想ブロックのうちの第3の仮想ブロック中に記憶することとをさらに含む動作をプロセッサに実施させるように構成される、請求項34に記載の非一時的コンピュータ可読記憶媒体。

【請求項４０】

前記記憶されたプロセッサ実行可能ソフトウェア命令が、

前記仮想境界コードが付加された前記第1のブートローダ部分を、前記第2のリードライトメモリデバイス上の仮想ブロック中に記憶することであって、ページサイズおよびデバイス固有の不良ブロックアルゴリズム特性のうちの少なくとも1つが、前記第1のリードライトメモリデバイスと前記第2のリードライトメモリデバイスとの間で異なる、ことをさらに含む動作をプロセッサに実施させるように構成される、請求項31に記載の非一時的コンピュータ可読記憶媒体。

【請求項４１】

リードライトメモリデバイスからデータイメージを読み取る方法であって、

第1のリードライトメモリデバイスの第1の仮想ブロック内に記憶されている第1のブートローダの第1の部分にアクセスするステップであって、前記第1のリードライトメモリデバイスの実ブロックの第1の実ブロックサイズおよび第2のリードライトメモリデバイスの実ブロックの第2の実ブロックサイズの両方が、剰余なしで、仮想ブロックサイズで割り切れ、前記第1の実ブロックサイズと前記第2の実ブロックサイズとが異なり、前記第1のリードライトメモリデバイスの前記実ブロックの各々および前記第2のリードライトメモリデバイスの前記実ブロックの各々が複数のページを含む、ステップと、

前記第1の仮想ブロックに続く第2の仮想ブロックの第2の仮想ブロック境界を指定する仮想境界コードを求めて、前記第1のリードライトメモリデバイスを走査するステップであって、前記第1の仮想ブロックおよび前記第2の仮想ブロックの各々が、前記第1のリードライトメモリデバイスの前記実ブロックの前記複数のページまたは前記第2のリードライトメモリデバイスの前記実ブロックの前記複数のページのうちの複数のページを含む、ステップと、

前記第2の仮想ブロック中の前記仮想境界コードを認識したことに応答して、前記第2の仮想ブロック内に記憶されている前記第1のブートローダの第2の部分にアクセスするステップであって、前記第1のブートローダの前記第1の部分が前記第1のブートローダの前記第2の部分とは異なり、前記第1のブートローダのサイズが前記仮想ブロックサイズより大きく、前記ブートローダの前記サイズが前記第1の実ブロックサイズまたは前記第2の実ブロックサイズのうちの1つまたは両方より大きい、ステップと、

前記第1のブートローダの前記第1の部分および前記第1のブートローダの前記第2の部分に基づいて、前記第1のブートローダを読み取るステップとを含む、方法。

【請求項４２】

前記第1のブートローダの前記第1の部分および前記第1のブートローダの前記第2の部分が前記第1のブートローダの完全なイメージを含まないと判断したことに応答して、前記第2の仮想ブロックに続く第3の仮想ブロックの第3の仮想ブロック境界を指定する前記仮想境界コードを求めて、前記第1のリードライトメモリデバイスを走査するステップと、

前記第3の仮想ブロック中の前記仮想境界コードを認識したことに応答して、前記第3の仮想ブロック内の前記第1のブートローダの第3の部分にアクセスするステップと、

前記第1のブートローダの前記第3の部分にさらに基づいて、前記第1のブートローダを読み取るステップとをさらに含む、請求項41に記載の方法。

【請求項４３】

前記第1のブートローダを実行するステップをさらに含む、請求項41に記載の方法。

【請求項４４】

前記仮想境界コードが前記実ブロック中で検出されないことに応答して、前記第1の仮想ブロックに続く、前記第1のリードライトメモリデバイスの前記実ブロックのうちのある実ブロックをスキップするステップをさらに含む、請求項41に記載の方法。

【請求項４５】

前記仮想境界コードが中間距離において検出されないことに応答して、前記第1の仮想ブロックのサイズに対応する、前記第1のリードライトメモリデバイス上の前記中間距離をスキップするステップをさらに含む、請求項41に記載の方法。

【請求項４６】

前記第1の仮想ブロック内の前記第1のブートローダの前記第1の部分中に含まれるヘッダ情報を走査するステップと、

前記ヘッダ情報に基づいて、前記第1のブートローダのサイズおよび前記仮想ブロックのサイズのうち少なくとも1つを判断するステップとをさらに含む、請求項45に記載の方法。

【請求項４７】

第2のリードライトメモリデバイスの第3の仮想ブロック内の第2のブートローダの第1の部分にアクセスするステップであって、ページサイズおよびデバイス固有の不良ブロックアルゴリズム特性のうちの少なくとも1つが、前記第1のリードライトメモリデバイスと前記第2のリードライトメモリデバイスとの間で異なる、ステップと、

前記仮想境界コードを求めて、前記第2のリードライトメモリデバイスを走査するステップと、

前記仮想境界コードを認識したことに応答して、前記第2のリードライトメモリデバイスの第4の仮想ブロック内の前記第2のブートローダの第2の部分にアクセスするステップと、

前記第2のブートローダの前記第1の部分および前記第2のブートローダの前記第2の部分

に基づいて、前記第2のブートローダを読み取るステップとをさらに含む、請求項41に記載の方法。

【請求項４８】

コンピューティングデバイスであって、

メモリと、

前記メモリに結合され、動作を実施するためのプロセッサ実行可能命令によって構成されたプロセッサとを備え、前記動作が、

第1のリードライトメモリデバイスの第1の仮想ブロック内に記憶されている第1のブートローダの第1の部分にアクセスすることであって、前記第1のリードライトメモリデバイスの実ブロックの第1の実ブロックサイズおよび第2のリードライトメモリデバイスの実ブロックの第2の実ブロックサイズの両方が、剰余なしで、仮想ブロックサイズで割り切れ、前記第1の実ブロックサイズと前記第2の実ブロックサイズとが異なり、前記第1のリードライトメモリデバイスの前記実ブロックの各々および前記第2のリードライトメモリデバイスの前記実ブロックの各々が複数のページを含む、ことと、

前記第1の仮想ブロックに続く第2の仮想ブロックの第2の仮想ブロック境界を指定する仮想境界コードを求めて、前記第1のリードライトメモリデバイスを走査することであって、前記第1の仮想ブロックおよび前記第2の仮想ブロックの各々が、前記第1のリードライトメモリデバイスの前記実ブロックの前記複数のページまたは前記第2のリードライトメモリデバイスの前記実ブロックの前記複数のページのうちの複数のページを含む、ことと、

前記第2の仮想ブロック中の前記仮想境界コードを認識したことに応答して、前記第2の仮想ブロック内に記憶されている前記第1のブートローダの第2の部分にアクセスすることであって、前記第1のブートローダの前記第1の部分が前記第1のブートローダの前記第2の部分とは異なり、前記第1のブートローダのサイズが前記仮想ブロックサイズより大きく、前記ブートローダの前記サイズが前記第1の実ブロックサイズまたは前記第2の実ブロックサイズのうちの1つまたは両方より大きい、ことと、

前記第1のブートローダの前記第1の部分および前記第1のブートローダの前記第2の部分に基づいて、前記第1のブートローダを読み取ることとを含む、コンピューティングデバイス。

【請求項４９】

前記プロセッサが、

前記第1のブートローダの前記第1の部分および前記第1のブートローダの前記第2の部分が前記第1のブートローダの完全なイメージを含まないと判断したことに応答して、前記第2の仮想ブロックに続く第3の仮想ブロックの第3の仮想ブロック境界を指定する前記仮想境界コードを求めて、前記第1のリードライトメモリデバイスを走査することと、

前記第3の仮想ブロック中の前記仮想境界コードを認識したことに応答して、前記第3の仮想ブロック内の前記第1のブートローダの第3の部分にアクセスすることと、

前記第1のブートローダの前記第3の部分にさらに基づいて、前記第1のブートローダを読み取ることとをさらに含む動作を実施するためのプロセッサ実行可能命令によって構成される、請求項48に記載のコンピューティングデバイス。

【請求項５０】

前記プロセッサが、前記第1のブートローダを実行することをさらに含む動作を実施するためのプロセッサ実行可能命令によって構成される、請求項48に記載のコンピューティングデバイス。

【請求項５１】

前記プロセッサが、

前記仮想境界コードが前記実ブロック中で検出されないことに応答して、前記第1の仮想ブロックに続く、前記第1のリードライトメモリデバイスの前記実ブロックのうちのある実ブロックをスキップすることをさらに含む動作を実施するためのプロセッサ実行可能命令によって構成される、請求項48に記載のコンピューティングデバイス。

【請求項５２】
前記プロセッサが、
前記仮想境界コードが中間距離において検出されないことに応答して、前記第1の仮想ブロックのサイズに対応する、前記第1のリードライトメモリデバイス上の前記中間距離をスキップすることをさらに含む動作を実施するためのプロセッサ実行可能命令によって構成される、請求項48に記載のコンピューティングデバイス。
【請求項５３】
前記プロセッサが、
前記第1の仮想ブロック内の前記第1のブートローダの前記第1の部分中に含まれるヘッダ情報を走査することと、
前記ヘッダ情報に基づいて、前記第1のブートローダのサイズおよび前記仮想ブロックのサイズのうち少なくとも1つを判断することとをさらに含む動作を実施するためのプロセッサ実行可能命令によって構成される、請求項52に記載のコンピューティングデバイス。
【請求項５４】
前記プロセッサが、
第2のリードライトメモリデバイスの第3の仮想ブロック内の第2のブートローダの第1の部分にアクセスすることであって、ページサイズおよびデバイス固有の不良ブロックアルゴリズム特性のうちの少なくとも1つが、前記第1のリードライトメモリデバイスと前記第2のリードライトメモリデバイスとの間で異なる、ことと、
前記仮想境界コードを求めて、前記第2のリードライトメモリデバイスを走査することと、
前記仮想境界コードを認識したことに応答して、前記第2のリードライトメモリデバイスの第4の仮想ブロック内の前記第2のブートローダの第2の部分にアクセスすることと、
前記第2のブートローダの前記第1の部分および前記第2のブートローダの前記第2の部分に基づいて、前記第2のブートローダを読み取ることとをさらに含む動作を実施するためのプロセッサ実行可能命令によって構成される、請求項48に記載のコンピューティングデバイス。
【請求項５５】
リードライトメモリデバイスからデータイメージを読み取るためのコンピューティングデバイスであって、
第1のリードライトメモリデバイスの第1の仮想ブロック内に記憶されている第1のブートローダの第1の部分にアクセスするための手段であって、前記第1のリードライトメモリデバイスの実ブロックの第1の実ブロックサイズおよび第2のリードライトメモリデバイスの実ブロックの第2の実ブロックサイズの両方が、剰余なしで、仮想ブロックサイズで割り切れ、前記第1の実ブロックサイズと前記第2の実ブロックサイズとが異なり、前記第1のリードライトメモリデバイスの前記実ブロックの各々および前記第2のリードライトメモリデバイスの前記実ブロックの各々が複数のページを含む、手段と、
前記第1の仮想ブロックに続く第2の仮想ブロックの第2の仮想ブロック境界を指定する仮想境界コードを求めて、前記第1のリードライトメモリデバイスを走査するための手段であって、前記第1の仮想ブロックおよび前記第2の仮想ブロックの各々が、前記第1のリードライトメモリデバイスの前記実ブロックの前記複数のページまたは前記第2のリードライトメモリデバイスの前記実ブロックの前記複数のページのうちの複数のページを含む、手段と、
前記第2の仮想ブロック中の前記仮想境界コードを認識したことに応答して、前記第2の仮想ブロック内に記憶されている前記第1のブートローダの第2の部分にアクセスするための手段であって、前記第1のブートローダの前記第1の部分が前記第1のブートローダの前記第2の部分とは異なり、前記第1のブートローダのサイズが前記仮想ブロックサイズより大きく、前記ブートローダの前記サイズが前記第1の実ブロックサイズまたは前記第2の実ブロックサイズのうちの1つまたは両方より大きい、手段と、

前記第1のブートローダの前記第1の部分および前記第1のブートローダの前記第2の部分に基づいて、前記第1のブートローダを読み取るための手段とを備える、コンピューティングデバイス。

【請求項５６】

前記第1のブートローダの前記第1の部分および前記第1のブートローダの前記第2の部分が前記第1のブートローダの完全なイメージを含まないと判断したことに応答して、前記第2の仮想ブロックに続く第3の仮想ブロックの第3の仮想ブロック境界を指定する前記仮想境界コードを求めて、前記第1のリードライトメモリデバイスを走査するための手段と、

前記第3の仮想ブロック中の前記仮想境界コードを認識したことに応答して、前記第3の仮想ブロック内の前記第1のブートローダの第3の部分にアクセスするための手段と、

前記第1のブートローダの前記第3の部分にさらに基づいて、前記第1のブートローダを読み取るための手段とをさらに備える、請求項55に記載のコンピューティングデバイス。

【請求項５７】

前記第1のブートローダを実行するための手段をさらに備える、請求項55に記載のコンピューティングデバイス。

【請求項５８】

前記仮想境界コードが前記実ブロック中で検出されないことに応答して、前記第1の仮想ブロックに続く、前記第1のリードライトメモリデバイスの前記実ブロックのうちのある実ブロックをスキップするための手段をさらに備える、請求項55に記載のコンピューティングデバイス。

【請求項５９】

前記仮想境界コードが中間距離において検出されないことに応答して、前記第1の仮想ブロックのサイズに対応する、前記第1のリードライトメモリデバイス上の前記中間距離をスキップするための手段をさらに備える、請求項55に記載のコンピューティングデバイス。

【請求項６０】

前記第1の仮想ブロック内の前記第1のブートローダの前記第1の部分中に含まれるヘッダ情報を走査するための手段と、

前記ヘッダ情報に基づいて、前記第1のブートローダのサイズおよび前記仮想ブロックのサイズのうち少なくとも1つを判断するための手段とをさらに備える、請求項59に記載のコンピューティングデバイス。

【請求項６１】

第2のリードライトメモリデバイスの第3の仮想ブロック内の第2のブートローダの第1の部分にアクセスするための手段であって、ページサイズおよびデバイス固有の不良ブロックアルゴリズム特性のうちの少なくとも1つが、前記第1のリードライトメモリデバイスと前記第2のリードライトメモリデバイスとの間で異なる、手段と、

前記仮想境界コードを求めて、前記第2のリードライトメモリデバイスを走査するための手段と、

前記仮想境界コードを認識したことに応答して、前記第2のリードライトメモリデバイスの第4の仮想ブロック内の前記第2のブートローダの第2の部分にアクセスするための手段と、

前記第2のブートローダの前記第1の部分および前記第2のブートローダの前記第2の部分に基づいて、前記第2のブートローダを読み取るための手段とをさらに備える、請求項55に記載のコンピューティングデバイス。

【請求項６２】

メモリを有するコンピューティングデバイス上のメモリを読み取るための動作をプロセッサに実施させるように構成されたプロセッサ実行可能ソフトウェア命令を記憶した非一時的コンピュータ可読記憶媒体であって、前記動作が、

第1のリードライトメモリデバイスの第1の仮想ブロック内に記憶されている第1のブー

トローダの第1の部分にアクセスすることであって、前記第1のリードライトメモリデバイスの実ブロックの第1の実ブロックサイズおよび第2のリードライトメモリデバイスの実ブロックの第2の実ブロックサイズの両方が、剰余なしで、仮想ブロックサイズで割り切れ、前記第1の実ブロックサイズと前記第2の実ブロックサイズとが異なり、前記第1のリードライトメモリデバイスの前記実ブロックの各々および前記第2のリードライトメモリデバイスの前記実ブロックの各々が複数のページを含む、ことと、

前記第1の仮想ブロックに続く第2の仮想ブロックの第2の仮想ブロック境界を指定する仮想境界コードを求めて、前記第1のリードライトメモリデバイスを走査することであって、前記第1の仮想ブロックおよび前記第2の仮想ブロックの各々が、前記第1のリードライトメモリデバイスの前記実ブロックの前記複数のページまたは前記第2のリードライトメモリデバイスの前記実ブロックの前記複数のページのうちの複数のページを含む、ことと、

前記第2の仮想ブロック中の前記仮想境界コードを認識したことに応答して、前記第2の仮想ブロック内に記憶されている前記第1のブートローダの第2の部分にアクセスすることであって、前記第1のブートローダの前記第1の部分が前記第1のブートローダの前記第2の部分とは異なり、前記第1のブートローダのサイズが前記仮想ブロックサイズより大きく、前記ブートローダの前記サイズが前記第1の実ブロックサイズまたは前記第2の実ブロックサイズのうちの1つまたは両方より大きい、ことと、

前記第1のブートローダの前記第1の部分および前記第1のブートローダの前記第2の部分に基づいて、前記第1のブートローダを読み取ることとを含む、非一時的コンピュータ可読記憶媒体。

【請求項６３】

前記記憶されたプロセッサ実行可能ソフトウェア命令が、

前記第1のブートローダの前記第1の部分および前記第1のブートローダの前記第2の部分が前記第1のブートローダの完全なイメージを含まないと判断したことに応答して、前記第2の仮想ブロックに続く第3の仮想ブロックの第3の仮想ブロック境界を指定する前記仮想境界コードを求めて、前記第1のリードライトメモリデバイスを走査することと、

前記第3の仮想ブロック中の前記仮想境界コードを認識したことに応答して、前記第3の仮想ブロック内の前記第1のブートローダの第3の部分にアクセスすることと、

前記第1のブートローダの前記第3の部分にさらに基づいて、前記第1のブートローダを読み取ることとをさらに含む動作をプロセッサに実施させるように構成される、請求項62に記載の非一時的コンピュータ可読記憶媒体。

【請求項６４】

前記記憶されたプロセッサ実行可能ソフトウェア命令が、前記第1のブートローダを実行することをさらに含む動作をプロセッサに実施させるように構成される、請求項62に記載の非一時的コンピュータ可読記憶媒体。

【請求項６５】

前記記憶されたプロセッサ実行可能ソフトウェア命令が、

前記仮想境界コードが前記実ブロック中で検出されないことに応答して、前記第1の仮想ブロックに続く、前記第1のリードライトメモリデバイスの前記実ブロックのうちのある実ブロックをスキップすることをさらに含む動作をプロセッサに実施させるように構成される、請求項62に記載の非一時的コンピュータ可読記憶媒体。

【請求項６６】

前記記憶されたプロセッサ実行可能ソフトウェア命令が、

前記仮想境界コードが中間距離において検出されないことに応答して、前記第1の仮想ブロックのサイズに対応する、前記第1のリードライトメモリデバイス上の前記中間距離をスキップすることをさらに含む動作をプロセッサに実施させるように構成される、請求項62に記載の非一時的コンピュータ可読記憶媒体。

【請求項６７】

前記記憶されたプロセッサ実行可能ソフトウェア命令が、

前記第1の仮想ブロック内の前記第1のブートローダの前記第1の部分中に含まれるヘッダ情報を走査することと、
前記ヘッダ情報に基づいて、前記第1のブートローダのサイズおよび前記仮想ブロックのサイズのうち少なくとも1つを判断することとをさらに含む動作をプロセッサに実施させるように構成される、請求項66に記載の非一時的コンピュータ可読記憶媒体。
【請求項６８】
前記記憶されたプロセッサ実行可能ソフトウェア命令が、
第2のリードライトメモリデバイスの第3の仮想ブロック内の第2のブートローダの第1の部分にアクセスすることであって、ページサイズおよびデバイス固有の不良ブロックアルゴリズム特性のうちの少なくとも1つが、前記第1のリードライトメモリデバイスと前記第2のリードライトメモリデバイスとの間で異なる、ことと、
前記仮想境界コードを求めて、前記第2のリードライトメモリデバイスを走査することと、
前記仮想境界コードを認識したことに応答して、前記第2のリードライトメモリデバイスの第4の仮想ブロック内の前記第2のブートローダの第2の部分にアクセスすることと、
前記第2のブートローダの前記第1の部分および前記第2のブートローダの前記第2の部分に基づいて、前記第2のブートローダを読み取ることとをさらに含む動作をプロセッサに実施させるように構成される、請求項62に記載の非一時的コンピュータ可読記憶媒体。
【発明の詳細な説明】
【技術分野】
【０００１】
リードライトメモリデバイスのデータイメージ中の仮想境界コードに関する。
【背景技術】
【０００２】
フラッシュメモリとは、電気的に消去し、プログラムし直すことができる不揮発性コンピュータ記憶チップである。NANDフラッシュメモリ(単にNANDメモリとも呼ばれる)とは、ブロックまたはページでプログラムされ、読み取られ得る高密度タイプのリードライトメモリである。NANDメモリは、メモリカード、USBフラッシュドライブ、ソリッドステートドライブ、および同様の製品において、データの全体的記憶および転送に使われ、ならびに、多数のデジタルデバイス中に構成データを記憶するのに使われる。
【０００３】
読取り専用メモリ(ROM)とは、コンピュータおよび他の電子デバイス内で使われるクラスの記憶媒体である。従来のROMは、読取り専用メモリであり、修正することができないか、修正できるとしても、遅速でしかなく、または問題を起こすことがあるので、主に、ファームウェアを配布するのに使われる。ファームウェアとは、特定のハードウェア用に特に設計されたソフトウェアを指す。デバイスのファームウェアの変更は、デバイスの経済的耐用年数の間には、滅多にまたは一度も行われない場合がある。ファームウェアを更新する一般的な理由には、バグを修正すること、またはハードウェアが使われるための特徴をハードウェアに追加することが含まれる。ファームウェアを更新する1つの方法として、特別な手順で、ROMとともに使われるフラッシュメモリを再プログラムすることがある。ROMとは異なり、NANDメモリは、複数回消去され、再プログラムされ得る。
【０００４】
NANDメモリは一般に、いくつかのブロックで編成され、各ブロックはいくつかのページからなる。NANDメモリの実ページの長さまたはブロックの長さは、製造元によって設定されると、その長さに永久に固定される。「ブロック」は、コンピュータメモリ、および特にNANDメモリに関する限り、公称長さ(nominal length)を有する、一連の記憶バイトまたはビットを備える。その長さは、ブロックサイズと呼ばれる。ブロック中にデータを記憶するプロセスは通常1つのページとして一度に遂行されるが、データの消去は、ブロック単位で行われる。
【０００５】

各ブロックはいくつかのページに編成することもでき、これらのページは一般に、2,048バイト(約2KB)または4,096バイト(約4KB)のサイズである。各ブロックの少なくとも1つのページに関連付けられるのは、誤り訂正コード(ECC)チェックサムの記憶に使うことができる数バイト(一般に、データサイズの1/32)である。典型的なブロックサイズは、以下を含む。

・64ページであって、各々が2,048バイト(約2KB)と64バイトの予備領域とを有し、これは128KB(ページデータ用)および4KB(たとえば、誤り訂正コード、不良ブロック情報、メタデータなどの予備データ用)のブロックサイズを意味する。

・64ページであって、各々が、4,096バイト(約4KB)と128バイトの予備領域とを有し、これは、ページデータ用の256KBのブロックサイズを意味する。

・128ページであって、各々が2,048バイト(約2KB)と128バイトの予備領域とを有し、これは、ページデータ用の256KBのブロックサイズを意味する。

【０００６】

NANDメモリの製作では、製造上の欠陥がしばしば起こり、これらの欠陥によりいくつかのメモリセルが機能しなくなり、および使用できなくなる。1つまたは複数の欠陥メモリセルを含むブロックは、「不良ブロック」と呼ばれる。製造プロセスにより、多くのNANDデバイスはいくつかの不良ブロックを有したまま、工場から出荷される。たとえば、いくつかの不良ブロックを許容することによって、製造元はすべてのブロックが良好であると検証されなければならないという場合よりはるかに高い生産を達成する。こうすることにより、NANDフラッシュのコストが大幅に削減され、部品の記憶容量の減少はほんのわずかとなる。それにもかかわらず、NANDメモリの製造は概して、初期ブロック(ブロック0)が良好であることは保証するが、残りのデータブロックは保証しない。

【０００７】

いくつかの不良ブロック管理方式では、チップ上の不良ブロックを識別する情報をNANDメモリ上に置く。そのような不良ブロック情報は、アクセスされるべきイメージを記憶する良好ブロックまたはスキップされるべき不良ブロックのいずれかを示すリストを含み得る。ただし、NANDメモリ自体の特性における不確定要素、ならびに再販業者および/または製造元固有の不良ブロック検出方式/フォーマットは、カスタマイズされたNANDメモリ操作およびプログラミングをあまりにも頻繁に要求する。

【０００８】

さらに、コンピューティングデバイスにおけるROMファームウェア設計は、ソフトウェアと比較して、ある程度制約されるが、それは、これらのルーチンが開発サイクルにおいて実際には後で書き換えることができず、コンピューティングデバイスが製造されると、概して永久的に設定されるからである。したがって、フラッシュメモリ上のデータイメージの全部または一部が、書き換えられ、したがって更新され得るので、フラッシュメモリは、ブートローダルーチンを稼働するのに使われる。本明細書では、「データイメージ」という用語は、NANDメモリデバイスなどのリードライトメモリデバイスの1つまたは複数のメモリセル中に記憶された情報を指す。現在のブートローダ用の、フラッシュメモリから離れて稼働されるデータイメージは、上記のように再販業者および製造元の間で大きく変わるNANDデバイス固有の特性を収容しようとするので、複雑になりがちである。

【発明の概要】

【課題を解決するための手段】

【０００９】

様々な実施形態は、不良ブロックを収容するとともに、コンピューティングデバイスが、メモリ製造元にかかわらず単一のアルゴリズムを使って、メモリから読取りを行うとき、不良ブロックをスキップすることを可能にする、フラッシュメモリなどのリードライトメモリ上にデータイメージを記憶することを含む。この方法は、データイメージを、良好データブロックの境界にあるコードまたは特殊値とともにリードライトメモリ中に記憶されるように構成するステップを含む。そのようなコードまたは特殊値は、本明細書では「仮想境界コード」と呼ばれ、このコードは、コンピューティングデバイスによって認識さ

れると、識別されたブロックが有効なデータを含むことを示す。データイメージは、メモリ中で、仮想ブロックの最初または最後を示す仮想ブロック境界に対応する仮想ブロックサイズを有する1つまたは複数の仮想ブロック中に記憶することができる。各仮想ブロックのサイズは、NANDデバイスについての実ブロックサイズ以下であり得る。仮想ブロックサイズは、製品によってサポートされることが期待される最小共通実ブロックサイズとして設定され得る。NANDメモリ上にデータイメージを記憶するためのシステムは、不良ブロックを識別するための、メモリ製造元が定義したアルゴリズムを使い、次いで、データイメージを良好ブロック中に記憶するとともに、不良ブロックをスキップする。仮想境界コードは、データイメージを記憶するのに使われる各良好ブロックの仮想ブロック境界に含まれる。データイメージはしたがって、コンピューティングデバイスがメモリチップ中の良好ブロックまたは不良ブロックを識別するために、再販業者固有パラメータに従う必要なく、リードライトメモリにアクセスするとき、読み取られるべきデータブロックを識別すると期待し得る仮想境界コードを含む。

【００１０】

別の態様は、コンピューティングデバイス内で実装され得る、リードライトメモリからデータイメージを読み取る方法を含む。この方法は、メモリの第1の仮想ブロック内のデータにアクセスするステップと、仮想境界コードを求めてメモリデバイスを走査するステップとを含み得る。仮想境界コードは、第1の仮想ブロックに続く第2の仮想ブロックの境界を指定する。方法は、仮想境界コードを認識したことに応答して、第2の仮想ブロック内のデータにアクセスするステップも含み得る。さらに、方法は、少なくとも、第1の仮想ブロック中でアクセスされたデータおよび第2の仮想ブロック中でアクセスされたデータに基づいて、データイメージを読み取るステップを含み得る。方法は、追加仮想ブロック中のデータに、仮想境界コードがそのブロック中で認識されたことに応答して、アクセスするステップをさらに含み得る。

【００１１】

さらなる態様は、上記で説明した方法に対応する様々な動作を実施するように、プロセッサ実行可能命令で構成されたプロセッサを有するコンピューティングデバイスを含む。

【００１２】

さらなる態様は、上記で説明した方法動作に対応する機能を実施するための様々な手段を有するコンピューティングデバイスを含む。

【００１３】

さらなる態様は、上記で説明した方法動作に対応する様々な動作をプロセッサに実施させるように構成されたプロセッサ実行可能命令を記憶した、非一時的プロセッサ可読記憶媒体を含む。

【００１４】

本明細書に組み込まれ、本明細書の一部をなす添付の図面は、本発明の例示的な実施形態を示し、上記の概略的な説明および下記の詳細な説明とともに、本発明の特徴を説明するのに役立つ。

【図面の簡単な説明】

【００１５】

【図１】ある実施形態によるリードライトメモリデバイスのパッケージングの図である。

【図２】図1のリードライトメモリデバイスを構成するためのプロセスステップを示す図である。

【図３】ある実施形態による、5つのリードライトメモリデバイスのパッケージングシナリオを示す図である。

【図４】ある実施形態による、2つの追加リードライトメモリデバイスのパッケージングシナリオを示す図である。

【図５】リードライトメモリ上にデータイメージを記憶する態様方法を示すプロセスフロー図である。

【図６】リードライトメモリデバイスからブートローダを読み取り、実行する態様方法の

プロセスフロー図である。
【図７】リードライトメモリデバイス上のイメージを書き換えるためのプロセスステップを示す図である。
【図８】リードライトメモリデバイス上のイメージを書き換える態様方法のプロセスフロー図である。
【図９】様々な実施形態との使用に適したコンピューティングデバイスの構成要素図である。
【図１０】様々な実施形態との使用に適した別のコンピューティングデバイスの構成要素図である。
【図１１】様々な実施形態との使用に適した別のコンピューティングデバイスの構成要素図である。
【発明を実施するための形態】
【００１６】
様々な実施形態が添付の図面を参照して詳細に説明される。可能な場合には必ず、同じ参照番号は、図面全体にわたって同じまたは同様の部分を指すために使用される。特定の例および実装形態へと行われる言及は、説明を目的とし、本発明の範囲または特許請求の範囲を限定するものではない。本開示の範囲から逸脱することなく、代替実施形態が考案され得る。さらに、本開示に関連する詳細を不明瞭にしないように、本開示のよく知られている要素は詳細に説明されない、または省略される。
【００１７】
「例示的な」という言葉は、「例、事例、または例示として機能すること」を意味するように本明細書で使用される。本明細書に「例示的な」と記載されるいかなる実装形態も、他の実装形態よりも好ましいまたは有利であると必ずしも解釈されるべきではない。さらに、「第1の」、「第2の」、「第3の」、「一次」、「二次」または「三次」という単語または同様の言い回しの使用は、本明細書では、様々な記載要素を区別するための明快さを目的としており、要素の特定の順序または階層に本発明を限定することは意図していない。本明細書で使用する「アクセス」または「アクセスする」という用語は、電子記憶デバイス、または指定される場合はデバイスの特定の部分と、デバイスまたは特定の部分の上のデータまたは情報を走査し、または読み取る目的で相互作用する働きを指す。また、本明細書で使用する「走査」または「走査する」という用語は、特に電子記憶デバイス中のデータまたは情報を調べ、解釈し、または吟味することを意味する。対照的に、本明細書で使用する「読み取る」または「読取り」という用語は、特に電子記憶デバイス中のデータまたは情報を取得し、抽出し、コピーし、または取り出すことを意味する。
【００１８】
ここで開示する様々な態様は、カスタマイズされた、および/またはデバイス固有の不良ブロック検出/管理ルーチンを不要にする、フラッシュメモリなどのリードライトメモリにデータイメージを記憶し、そのようなメモリからデータイメージを読み取るための機構を提供する。開示する技術によるある態様は、フラッシュメモリデバイスなどのリードライトメモリデバイスにデータイメージを記憶する方法を含む。この方法は、リードライトメモリデバイス用に指定された、一連の仮想ブロックのうちの仮想ブロックサイズに基づいて、データイメージの配置を判断するステップを含む。仮想ブロックは本明細書の態様に従って作成され、実ブロックはフラッシュメモリのような、リードライトメモリの固定されたパラメータである。リードライトメモリデバイスの実ブロックサイズは、剰余なしで、仮想ブロックサイズで割り切れる。このように、1つまたは複数の仮想ブロックは、実ブロックと正確に等しくなる。データイメージが仮想ブロックサイズよりも大きい場合、データイメージが記憶されることになるリードライトメモリのブロックサイズを収容するための部分に分割される。データイメージ部分のうちの少なくとも1つに、仮想境界コードが付加される。仮想境界コードは、データイメージ部分の始端に付加され、仮想ブロックの初期境界をマーキングすることができる。データイメージは、ブートプロセスにおいてコンピューティングデバイスによって使われるデータおよびソフトウェアの一部で

あり得る。データイメージは、仮想ブロックサイズおよびデータイメージ全体のサイズに応じて、必要とされるだけの数の部分に分割され得る。データイメージの全部または一部は、NANDメモリのブロック0に依然として記憶されてよい。方法は、一連の仮想ブロックのうちのある仮想ブロック中に、データイメージ部分を記憶するステップをさらに含む。データイメージ部分を記憶する際、いかなる介在不良ブロックもスキップされる。このように、1つまたは複数の不良ブロックは、別個の仮想ブロック中に記憶された記憶データイメージ部分の間に配列され得る。データイメージの第1および第2の部分は、リードライトメモリデバイスの共有実ブロック中に配列され得る。データイメージの後続部分は各々、付加された仮想境界コードを有し得る。このコードは、良好/不良ブロックリストに依拠する必要なく、読み取られるべきメモリの仮想ブロックを位置特定することをコンピューティングデバイスが期待し得る仮想境界コードとともに、データイメージをリードライトメモリ上に記憶する。こうすることにより、コンピューティングデバイスを、大きく変わるような様々な良好ブロック/不良ブロックスキームを収容するようにプログラムする必要がなくなる。このように、仮想境界コードが付加されたものを含む同じデータイメージ部分は、追加リードライトデバイス上に記憶され得る。それらの追加リードライトデバイスは、互いとは異なるページ、ブロックおよび/またはデバイス固有の不良ブロックアルゴリズム特性さえも有し得る。

【００１９】

コンピューティングデバイスが、上述した方法を使って、作成されたデータイメージを使用するリードライトメモリからデータイメージを読み取る態様方法を実装することができる。この方法は、リードライトメモリがブートローダイメージを記憶するのに使われるとき、コンピューティングデバイス向けに実行されるブートローダルーチンにおいて実装され得る。そのブートローダイメージは、コンピューティングデバイス用に実行される排他的ブートローダである必要はない。この方法は、リードライトメモリの仮想ブロック内のデータにアクセスするステップを含む。リードライトメモリデバイスの実ブロックサイズは、剰余なしで、仮想ブロックサイズで割り切れ、これは実ブロックサイズが仮想ブロックサイズに等しいというシナリオを含む。アクセスされている完全なデータイメージが、その仮想ブロック中に含まれていない場合、方法は別の仮想ブロックの境界(たとえば、始端)を指定する仮想境界コードを探して、リードライトメモリ中を前方走査し得る。仮想境界コードが認識されると、ブートローダルーチンなどのコンピューティングデバイスデータロードルーチンは、その他の仮想ブロック内のデータにアクセスすることができる。上述した方法によってメモリがロードされたことにより、仮想境界コードは、仮想境界コードがその中で識別された仮想ブロックが良好であるとともに、読み取ることができるデータイメージの一部分を含むことを示す。この追加データブロック中での読取りの後、完全なデータイメージがまだアクセスされていない場合、コンピューティングデバイスデータロードルーチンはデータイメージ全体が読み取られるまで、さらなる仮想境界コードおよびデータブロックを求めてメモリを走査し続ける。データイメージの量/サイズは、イメージヘッダ情報から判断され得る。このプロセスは、データイメージ全体がロードされたことをデータロードルーチンが認識すると終了することができる。仮想境界コードを求めてメモリを走査することによって、どのブロックが不良または良好であるかをコンピューティングデバイスが前もって判断する必要なく、不良ブロックは(仮想境界コードを含まないので)自動的にスキップされ、良好ブロックからのみデータが読み取られることになる。別の態様によると、方法は、第1のブロック中(たとえば、データイメージのプリアンブル部分中)に記憶された情報に基づいて、データイメージのサイズまたは仮想ブロックサイズを外挿することができる。リードライトメモリのページサイズは、データイメージの第1のブロックまたはプリアンブル部分から推論することができ、これは、仮想ブロックごとのページの数を判断するのに使われ得る。

【００２０】

さらなる態様は、メモリ中に記憶された既存のデータイメージが更新され、または置き換えられたときなどに、更新または新規データイメージでフラッシュメモリを書き換える

ための方法を含む。この方法は、初期データイメージを、リードライトメモリの少なくとも1つの初期仮想ブロック中に含む。また、方法は、リードライトメモリ中で利用可能な少なくとも1つの後続仮想ブロックを求める、少なくとも1つの初期仮想ブロックの前方走査を含む。新規データイメージの第1の部分は、付加された仮想境界コードを含む後続仮想ブロック中に記憶することができる。方法は、新規データイメージ全体が記憶された後、少なくとも1つの初期仮想ブロックから初期データイメージを消去するステップをさらに含み得る。新規データイメージは、その少なくとも1つの初期仮想ブロック中に再び書き込みすることができる。実際、新規データの再書込みおよび古いデータの消去は、1つのアクションとして実施することができる。ただし、新規データイメージは、先頭の仮想ブロック中に書き込まれる必要はなく、後に続く1つまたは複数の仮想ブロックに書き込めばよい。新規データイメージ全体が、その少なくとも1つの初期仮想ブロックまたは先頭の仮想ブロックに続く1つもしくは複数の仮想ブロック中に書き込まれると、方法は、新規データイメージが一時的に書き込まれた後続仮想ブロックから新規データイメージを消去するステップも含み得る。

【００２１】

コンピューティングデバイスのプロセッサは初めて電源投入されたとき、概して、オペレーティングシステムがROMまたはRAMにロードされていない。したがって、プロセッサは最初に、システムを起動するプロセスを始めるブートローダと呼ばれるROMファームウェア中に記憶されたプログラムコードを実行する。ブートローダのジョブは、他のデータおよびプログラムをランダムアクセスメモリ(RAM)中にロードすることであり、プログラムは次いでプロセッサによって実行される。一次ブートローダは概して、ROM中に記憶され、電源投入時またはデバイスがリセットされたときに実行される。その後、CPUに動作可能に結合されたNANDデバイス上に記憶された1つまたは複数の二次ブートローダが、新規および/または追加アプリケーションコードをロードさせ、実行させ得る。従来、複雑さを増大させるいくつかのプログラムが、連鎖ローディングのプロセス中に次々とロードを行う多段階ブートローダが使われている。ただし、メモリの異なるページおよび/またはブロックサイズならびに不良ブロック特性にわたってスケーラブルなNANDフラッシュメモリ中に、単一の可変サイズのブートローダを有することが有利な場合がある。そのようなNANDメモリブートローダは依然として二次ブートローダであり、かつ/または他のブートローダプログラムと協調することができる。

【００２２】

NANDメモリ製造元および再販業者は、メモリデバイス中で不良ブロックを識別するための統一標準フォーマットを使用しない。それにもかかわらず、コンピューティングデバイスは、メモリ中の不良ブロックの扱い方に対応しなければならず、そうでないとデバイスは適切に作用することも一貫して作用することもない。したがって、不良ブロックソフトウェアルーチンは一般に、不良ブロックを収容するとともに、コンピューティングデバイスと一緒に使われることが期待され得る、各製造元/再販業者のNANDメモリデバイスを操作する方法を提供するように構成された、二次ブートローダを含むデータローダルーチン内に含まれる。ただし、NANDメモリチップは、いくつかの製造元または再販業者のうちのどこからも購入される可能性があり、再販業者は、その不良ブロック検出情報/方法を変える場合がある。現在の不良ブロックソフトウェアルーチンは、コンピューティングデバイス内にプログラムされたブートローダに、いくつかの異なるルーチンを含めることによって、多様に異なる不良ブロック検出方式を扱う。これは、再販業者の多様な要求および仕様を扱う目的で行われる。そのような多様性を不良ブロックルーチン中に収容すると、ブートローダROMファームウェアを複雑にする可能性があり、再販業者の変更を受け入れるためにそのようなルーチンを変えることは、概して望ましくない。また第1のブートローダは、良好と保証される、NANDメモリの最も小さい典型的なブロック0内に全体が存在するように設計され得るが、これによりシステムの全体としての起動が複雑になり、しばしば、ドライバとブートローダが重なってしまう。本明細書の態様は、そのような複雑化および/または二重化を縮小し、ROMが、第1の良好ブロック以外にも繋がっているブート

ローダをロードすることを可能にする。

【００２３】

開示する実施形態は、メモリチップ上のデータイメージをロードするプロセスのための、NANDメモリ再販業者によって実装される多様な不良ブロック検出方式の必要性を制限し、かつ/またはなくすための方法を含む。コンピューティングデバイスが、不良ブロックをルックアップするのにリソースを使う必要がなく、ロードされるべきデータブロックを識別することを可能にするための仮想境界コードを追加することによって。処理される(すなわち、データイメージとともにロードされる)リードライトメモリ向けの汎用の不良ブロック検出/操作ルーチンの実装は、機械セットアップの一部として実装され得る。

【００２４】

様々な態様において、リードライトメモリ上で記憶されるべきデータイメージは、仮想ブロックに対応するデータイメージ中の位置に各々が位置特定される、1つまたは複数の仮想境界コードを有して構成される。仮想ブロックは、コンピューティングデバイス内で実装され得るリードライトメモリのブロックサイズに準拠するようにサイズ指定される。仮想ブロックサイズは、メモリの実ブロックサイズと一致し得る。ただし、異なるメモリ構成(すなわち、異なるサイズの実ブロックをもつメモリチップ)を収容するために、仮想ブロックサイズは、コンピューティングデバイス内で実装され得るすべてのメモリの実ブロックの最小公倍数サイズに設定されてよい。こうすることにより、態様方法は各タイプのメモリについて別個のデータイメージを要求することなく、異なるページサイズ(たとえば、2Kおよび4K)のNANDデバイスをサポートすることが可能になる。(少なくとも初期ブロックの後の)仮想ブロック内の正規位置に仮想境界コードを付加することによって、同じデータイメージは実装され得る任意のメモリの良好ブロック中に記憶することができ、いかなる介在不良ブロックもスキップされ得る。そのような仮想境界コードは、始端など、仮想ブロック中のほぼどこにおいても付加され得る。

【００２５】

データイメージがリードライトメモリ、たとえばNANDメモリ上に記憶されると、再販業者固有の不良ブロック検出方法が、不良ブロックを避けるとともにメモリ中の良好ブロックのみを使うために使われる。ただし従来のシステムは、再販業者固有の不良ブロック検出方法を使うためにも、そのデータイメージを引き続き読み取るルーチンを要求する。対照的に、開示する実施形態の態様は、データイメージを読み取るとき、そのような再販業者固有方法を使う必要性を最小限にするか、またはなくし、ほぼどのリードライトメモリデバイスにも適用可能な、より汎用な方法を提供する。本明細書で開示する態様は、データイメージを、リードライトメモリ上の1つまたは複数の仮想ブロック中に記憶する。データイメージは、リードライトメモリの1つまたは複数の仮想ブロック中に記憶することができる。データイメージ全体を収容するため複数の仮想ブロックが必要とされる場合、次の利用可能仮想ブロック(必要なら複数のブロック)が、完全データイメージが記憶されるまでデータイメージの残りの部分を記憶するのに使われてよく、この場合、いかなる介在不良ブロックもスキップされる。仮想ブロックは、各仮想ブロック中に置かれたデータイメージの部分に付加された仮想境界コードによって定義および画定される。一実施形態では仮想境界コードを、各仮想ブロックの最初に置く。また、特に、デバイス上の第1のブロックは概して良好であることが保証されるので、先頭の仮想ブロックは仮想境界コードを必要としない場合がある。また仮想境界コードが、次および/または後続仮想ブロックの最初に置かれた場合、データイメージの部分が記憶されるロケーションを識別する。仮想境界コードが付加されたデータイメージは、メモリの良好な実ブロック中にのみ記憶され、したがって仮想境界コードは、メモリの良好な実ブロック内の良好な仮想ブロックに対応する。その結果、ブートローダは、本明細書の態様による単一のイメージ読取りプロセスを使って、製造元または再販業者の存在を考慮することなく、ほぼどのリードライトメモリも収容する方法で仮想境界コードを認識し、読み取るべきメモリのブロックを判断することができる。態様方法は、上述したようにプログラムされたNANDメモリから、計算能力をもつスマートフォンまたは他の電子機器などのデバイスを起動するために使うこ

とができる。また、態様方法は、二次ブートローダのフェールセーフな更新の操作に適用することができる。

【００２６】

態様方法は、フラッシュメモリ中に記憶されているとともに、スマートフォンまたは他のモバイルコンピューティングデバイスなどのコンピューティングデバイスによって、電源投入時またはリスタート時にロードされる初期ソフトウェアとして使われる初期ブートソフトウェアイメージに特に適用可能であり得る。一般に、そのようなブートソフトウェアは、オペレーティングシステム、プロセッサまたはデバイス製造元によって厳密に制御される。ブートソフトウェアを制御する会社は一般に、一次ブートロードイメージをROMメモリに「焼き」、コンピューティングデバイスは次いで、そのブートルーチンの一部として、NANDメモリなどのリードライトメモリにアクセスして、初期ソフトウェアイメージをロードする。

【００２７】

態様方法、システムおよびデバイスは、不良ブロック操作を提供し、ほぼどのNANDデバイスからの単一の二次ブートローダ(本明細書では、ブートローダとも呼ばれ、「BL」と略される)アーキテクチャも可能にする。現在のシステムでは、二次ブートローダのサイズは、内部SRAMメモリのサイズで規定され得る。したがって、たとえば、NANDデバイス内でSRAMサイズが256KBである場合、概してブロック0に置かれたプリアンブル用のキロバイトを減算し、証明書(Certs)やパディング(PAD)など、その他の追加データ用の追加キロバイトを減算すると、その初期実ブロック中には約240KBが残り得る。その240KBサイズは、多くの個々のブートローダの最大サイズに対応し、現在リードライトメモリ中で概して実装される128KBの最小NANDブロックサイズよりも大きい。本明細書の態様は、仮想ブロックサイズを128KBという現在の最小NANDブロックサイズに対応するように設定する。したがって、そのSRAMから240KB BLを実行するために、複数の仮想ブロックが、一次ブートローダによって読み取られる必要がある。いくつかの現在のフラッシュメモリデバイスは、128KBという最小値よりも大きいブロックサイズを有し、一部は256KBを有し、より大きいメモリが開発されているが、そのような比較的大きい実ブロックは128KBの、2つ以上の仮想ブロックを含み得る。また、本明細書における方法の態様は、32KBなど、より小さいブロックサイズにも拡張することができる。

【００２８】

128KBのセグメントサイズは、本明細書において、説明が目的の例としてのみ使われ、これらのセグメントは、検討される特定のNANDデバイスに応じて、より大きくても、より小さくてもよいことを理解されたい。たとえば、128KBの仮想ブロックサイズが使われる場合、このサイズは、ブロックごとに64ページ未満またはブロックごとに128KB未満であるNANDデバイスをサポートすることができない。ただし、仮想ブロックサイズがますます小さくなると、必要とされる仮想ブロックの数が増し得るので、より小さいブロックサイズは、現実的な制限を有する。より多くの仮想ブロックは、有効なデータイメージを見つけるためにより多くの走査が必要とされることを意味し、これは起動時間の増大につながり得る。

【００２９】

態様方法は、NANDデバイス上に置かれたBLイメージを、128KBの仮想ブロックサイズ範囲内に置くことができるセグメントに分割する。128KBの仮想ブロックサイズは、NANDデバイス向けの現在の最小値に対応するときに選ばれるが、所望されるときは、異なる仮想ブロックサイズが使われてもよい。仮想ブロックは、少なくとも第2の128KBセグメント内にマーカを付加することによって作成される。第1の128KBセグメントは概して、BLプリアンブルおよび他の可読コードを含み、したがって、仮想境界コードを含む必要はない。仮想境界コードによってマーキングされた各セグメントは、仮想ブロックおよびその境界(仮想ブロックの2つの端であって、本明細書では「仮想ブロック境界」とも呼ばれる)を定義する。BLイメージの残りの部分は分割され、イメージ全体を収容するのに必要とされるだけの数の仮想ブロックに配置される。不完全な仮想データブロックを埋めるために、パ

ッキングデータが追加され得る。

【００３０】

様々な態様により、ROMファームウェアの開発者はもはや、NANDメモリ再販業者が定義した不良ブロック検出方式を収容する必要がなくなる。様々な態様は、不良/良好ブロックテーブルを保持する事前プログラムされたページに依拠しない。開示する技法は、異なるNANDデバイス用に別個のツール/ビルドをあらかじめ要求している、フラッシュツールおよびビルド管理の設計を簡素化する。さらに、開示するデバイス、システムおよび方法は、ブートローダおよびその操作のフェールセーフな更新を可能にする。さらに、このNAND設計は、異なるページ/サイズNANDデバイスをサポートするようにスケーラブルである。

【００３１】

図1は、様々な態様による、スタンドアローンブートローダ15をもつNANDデバイス100のパッケージングを示す。本実施形態では、128KBの仮想ブロックサイズが使われ、このサイズは、上記のように、2KBまたは4KB NANDデバイスのいずれかにとって有効である。したがって、128KB仮想ブロックを確立するために、仮想ブロック境界が、仮想境界コード14によって指定され得る。そのような仮想境界コード14は、12バイトのデータ程度に小さくてよい、一意であり認識可能なデータパターンである。一例として、仮想境界コード14は、844bdc56 73531014 d48b54c6という16進数であってよい。これよりも大きいか、または小さい仮想境界コード14が使われてもよいが、仮想ブロック境界を画定するために少量のデータのみが必要とされるので、大きいものである必要はない。また、一次ブートローダスタック用に予約されたSRAMのさらなる部分、すなわち共有エリアおよびブロック0の始端にある他のデータが考慮されてよい。したがって、利用可能SRAMは、実際のSRAMサイズよりも小さくてよい。

【００３２】

図1は、プリアンブルメモリコードと呼ばれ得る1つまたは複数のメモリコード10を含むプリアンブル5も示す。いくつかのプリアンブルメモリコード10を含めても、プリアンブルは、10KBを超えないと想定され得る。したがって、BL15は、NANDデバイス100のブロック0の最初の10KBの後に始まって、書込みを受け得る。図1に示す例では、BL15の最初の80バイトはイメージヘッダ12を含み、イメージヘッダ12はブートローダの第1の部分15aの冒頭を指定し、仮想ブロックに対して設定されたサイズなどの付加情報を含み得る。その後、仮想ブロック境界は、128KB境界における仮想境界コード14を付加することによってマーキングすることができ、その後には、次の128KB中のブートローダの第2の部分15bが続く。仮想境界コード14は、それ以外の場合、典型的な二次ブートローダの中央を占め、したがってBL15は複数の部分15a、15bに分割される。プリアンブルサイズ、イメージヘッダサイズおよび仮想ブロックのサイズを知ることによって、第1のブートローダ部分15aのサイズについての判断が行われ得る。ブートローダの残りは次いで、ブートローダの第2の部分15bまたは必要な場合は追加仮想ブロック中のさらなる部分に含められてよい。したがって、いくつかの既知変数または想定変数が、分割されたBL15の各部分がどの程度大きくなり得るかの判断に含められてよい。たとえば、以下のパラメータを検討する。

・SRAMサイズ=256KB
・予約済み/使われるSRAM=約16KB
・仮想ブロックサイズ=128KB
・プリアンブル(5)サイズ(プリアンブルメモリコード10を含む)=10KB
・ブートローダイメージヘッダ(12)サイズ=80B
・署名および証明書(16)にパディング(18)を加えたもの=6KB
・仮想境界コード(VBC)サイズ=12B

使われる上記バイトの合計を、SRAMサイズから減算すると、2つの128KB仮想ブロック中に記憶することができる、NANDメモリ中のブートローダ用のほぼ223KBが残る。

【００３３】

図2は、ブートローダ15の間隔および配置の別の検討法を示す。最初に20で、仮想ブロ

ックサイズが128KBで確立されてよく、したがって2つのそのような仮想ブロックが、256KB SRAM中に含まれることになる。22で、仮想ブロックのうちの少なくとも2つにわたり得るBL15のサイズが検討され得る。次いで24で、イメージヘッダ12および証明書16が、BL15イメージの全長に追加される。また26で、プリアンブルメモリコード10を含むプリアンブル5、およびパディング18が、イメージの全長に追加される。さらに28で、仮想境界コードが、BL15の2つの部分15a、15bの間に付加され、このコードは、本実施形態では、第1の128KB仮想境界に対応する。これらのパラメータを考慮に入れて、実デバイスのページサイズにより、何個の実ブロックが使われるか、ならびに何個の仮想ブロックが使われるかが決まる。たとえば、サイズが4Kであるページをもつ NANDデバイスを検討する。そのようなデバイスは、256KBの実ブロックを有し、ブロック0には、ブートローダ用におよそ223KBが残る。したがって、4K NANDデバイスは、このBLを保持するのに、1つの良好ブロックを必要とするだけである。128KBの仮想ブロックサイズが使われる場合、その1つの良好な実ブロックが2つの仮想ブロックを含む。ここで、サイズがわずか2kであるページをもつNANDデバイスを検討する。このデバイスは、2つの良好な実ブロックを要求する。これらの実ブロックのサイズは、128KBという仮想ブロックサイズと一致する。したがって2K NANDのブロック0は、第1のBL部分15a用に、ほぼ118KBの空間を残す(128KBから10KBを引いたプリアンブル5、80Bを引いたイメージヘッダ12)。残りの第2のBL部分15bは次いで、仮想境界コード14によって指定された次の良好ブロック中に置かれ得る。

【００３４】

図3は様々なシナリオにおいて、例示的ブートローダの配置がどのように展開し得るかを示す。図示される5つのシナリオはすべて、128KBの仮想ブロックサイズ(5つのシナリオの上に一連の仮想ブロックとして示す)を使う。シナリオのうちの3つは2KB NANDデバイスについて検討し、他の2つのシナリオは4KB NANDデバイスについて検討する。2KBシナリオでは、仮想ブロックサイズは128KBの実ブロックサイズと一致する。4KBシナリオでは、仮想ブロックは実ブロックのサイズの半分である。

【００３５】

2KBシナリオ1において、ブートローダの分割および配置は単純であり、仮想境界コード14がブロック1の始端に置かれ、デバイス上でマーキングされた第1の仮想ブロック境界と一致する。仮想ブロックは、2KBシナリオにおける実ブロックと一致することに留意されたい。上述したNANDメモリのコーディングと同様に、ブロック0はプリアンブルメモリコード10、ブートローダイメージヘッダ12および第1のブートローダ部分15aをもつプリアンブル5も含む。第2のブートローダ部分15bが仮想境界コード14に続き、コード14は本実施形態において、部分15bに付加され、仮想ブロックの始端をマーキングする。この2KBシナリオ1は、ブロック1が良好であり、したがって第1および第2の仮想ブロックが、連続した良好ブロックであるNANDデバイスに該当する。ただし製造元は概して、ブロック0が良好であると保証するだけであり、したがって第2および第3のシナリオを2K NANDデバイス用に検討する。2KBシナリオ2において、ブロック1は不良であるがブロック2は良好である。したがってブロック1が不良であるので、仮想境界コード14をそのブロック中に書き込むことができない。一方、次の良好ブロック、すなわちブロック2が、仮想境界コード14が次いで第2のブートローダ部分15bとともに書き込まれる場所である。2KBシナリオ3は、ブロック0に続く2つの初期不良ブロック(ブロック1およびブロック2)を有する。したがってこの場合、仮想境界コード14は、それらの2つの不良ブロックのうちの、次の良好ブロック(ブロック3)の始端である、最後に続く境界に書き込まれる。同様に、そのシナリオにおいて、BLの残り(BLの第2の部分15b)がブロック3中に書き込まれる。

【００３６】

4KBシナリオにおいて、ブートローダは概して、2つの仮想ブロックを含むのに十分大きいブロック0内に全体が収容され得るので、ブロック1が良好であるか不良であるかはほとんど問題にはならない。したがって、4KBシナリオ1は、ブロック1が良好でありながら、BL第1部分15aおよびBL第2部分15bが両方ともそのままブロック0中に書き込まれ得るので、必要とされない状況を示す。また、4KBシナリオ2は、BL15a、15bの両方の部分が、全体が

依然としてそのブロック0中にどのように書き込まれ得るかを示し、そのシナリオにおけるブロック1が不良であることを不適切にする。

【００３７】

図4は、ブートローダの配置のもう2つのシナリオを示す。図4における第1のシナリオは4KBシナリオ3であり、このシナリオは、3つの部分15a、15b、15cに分割される必要がある、より大きいBLを含む。第2の実ブロック、すなわちブロック1が良好なので、データイメージは、連続した仮想/実ブロックに広がる。ただし、ブロック1が不良だった場合、第2の仮想境界コード14およびBLの第3の部分15cは、ブロック2中に記憶されてよい。図4における第2のシナリオは、2KBシナリオ4である。繰返しになるが、3つの部分15a、15b、15cに分割される必要があるより大きいBLが使われるが、ここで第3のブロック、すなわちブロック2は不良である。したがって、第4のブロック、すなわちブロック3は、第2の仮想境界コード14およびBLの第3の部分15cを記憶するのに使われる。

【００３８】

図5は、リードライトメモリデバイス上にデータイメージを記憶する態様方法500を示す。データイメージは、本明細書の態様によるブートローダであってよい。ブロック510で、NANDデバイスなどのリードライトデバイスが、デバイス上でのデータ記憶のためにロードされ、準備される。ブロック520で、リードライトメモリデバイス用に指定された一連の仮想ブロックのサイズに基づいて、データイメージをパッケージングするために、データイメージの配置が判断される。この判断の一部として、データイメージは、データイメージ全体が単一の仮想ブロックにとって大きすぎる場合、少なくともデータイメージの第1の部分およびデータイメージの第2の部分に分割される必要があり得る。また仮想境界コードが、データイメージの第1の部分およびデータイメージの第2の部分のうちの少なくとも1つに付加される。本実施形態では、仮想境界コードはデータイメージの第2の部分の始端に付加され、したがってブロック530においてデータイメージの第1の部分は、一連の仮想ブロックの第1の仮想ブロック中に記憶される。ブロック540で仮想境界コード(VBC)が記憶され、次の仮想ブロックをマーキングし(したがって、一連の仮想ブロックのうちの第2のものを指定し)、このブロックは当然ながら、良好な実ブロックと一致する。VBCを記憶し仮想ブロックを指定する際、第1の仮想ブロックと第2の仮想ブロックとの間のいかなる介在不良ブロックも、リードライトメモリ内でスキップされる。データイメージを物理的に書き込むのに使われるプログラミングツールは、再販業者固有の生産工場がマーキングした不良ブロックについてのチェックを実施し、必要に応じて不良ブロックをスキップすることができる。プログラミングツールの目的は、データイメージ内にVBCを記憶し、かつ/または組み込むことであり得る。したがって、VBCは再販業者不良ブロック管理方式に依存せずに読み取られ得るパターンを作成するため、データイメージの一部として作られるので、プログラミングツールはVBCを認識する必要がなく、扱う必要さえない。そのようなプログラミングツールはNANDドライバも有し得るが、ROM中のNANDドライバとは異なり、プログラミングツールのNANDドライバはサイズについても複雑さについても、制限される必要はない。これにより、プログラミングツールは、製品に対する改訂または更新に従ってバグを修正し、新規NAND再販業者固有の不良ブロックチェック方式を扱うことも可能になる。このようにして、そのようなプログラミングツールにおいて、ROM中のNANDドライバと比較して、機能性が追加、削除、変更または強化され得る。ブロック550で、データイメージの第2の(次の)部分がVBCと同じ仮想ブロック中に記憶されてよく、このブロックは、一連の仮想ブロックのうちの第2の仮想ブロックである。判断ブロック560は、完全なデータイメージが記憶されたかどうか判断する。方法が、次の利用可能仮想ブロックに別のVBCを記憶するためにブロック540に戻らない場合、再度ブロック550で、データイメージの次の部分を記憶する。このループは、データイメージのすべての部分が記憶されるまで進む。判断ブロック560で、完全なデータイメージ(たとえば、ブートローダイメージ)が記憶されたという肯定的判断が行われると、570で、ブロック510の始端においてロード/準備されたリードライトデバイスの一連のプロセスは完結する。プロセスはしたがって順番に、または追加リードライトデバイスと並行して繰り返され得る。

【００３９】
仮想ブロックの始端にVBCを位置決めすることは、良好ブロックの迅速な検出を助け、ブートローディングプロセスを最適化することができる。仮想ブロックの単一パス読取りによりVBCを検出し、同時に、そのブロック中に含まれるブートローダデータイメージを読み取ることができる。代替として、仮想境界コードは仮想ブロックの始端に置かれる必要はない。たとえば、VBCは各仮想ブロック(ブロック0を含む)の端に、または所望されるほぼどの位置に置かれてもよい。1つまたは複数のVBCに関する位置決め情報は、上で論じたイメージヘッダ中に含まれ得る。
【００４０】
NANDデバイスに1つまたは複数の仮想境界コードが書き込まれると、一次ブートローダがデバイスを解析して、それらの仮想境界コードを見つけ、指標のような仮想境界マーカを使って、完全な二次ブートローダを読み取り、認証し、稼働することができる。図6は、リードライトデバイスからデータイメージを読み取る態様方法を示す。
【００４１】
図6は、リードライトメモリからデータイメージを読み取る態様方法600を示す。たとえば、一次ブートローダ中のNANDパーサが方法600を実装して、二次ブートローダを読み取り、ロードし、認証することができる。ブロック605で、リードライトデバイスは、ブロック0中のプリアンブルから開始して読み取られ、これはデバイスのページ0とも一致する。プリアンブルは、データロードルーチンによって使うことができるデバイス幅およびページサイズ情報を含み得る。ブロック610で、誤り訂正コーディング(ECC)検出が、好ましく現行の方法に対して有効にされているので、ECCが実施され得る。この動作は、4ビットBCH誤り訂正コーディングまたは他のコーディングにデフォルト設定され得る。プログラムされたページの読取りが、ECCが検出されないように不成功の場合、ブロック612では追加ECC構成がチェックされ得る。たとえば、8ビットBCH ECCが使用され、またはそれがデフォルトでなかった場合は4ビットBCH ECCが使われ得る。したがって、1つまたは複数の追加ECC構成がチェックされてよく、そうすることによってブロック614でどのECCも検出されなかった場合、プロセスはブロック616で終了し、これはどのブートローダ(BL)も見つけられなかったことを意味する。代替として、方法はAUTO_DETECTルーチンおよび重み付きアルゴリズムチェックを使って、有効なプリアンブルをチェックすることができる。別の方法では、ブロック610またはブロック614のいずれかでECCが検出された場合、方法はブロック620に進み、ここでページサイズ検出が実施される。ブロック620で、実NANDページサイズを算出するために、連続したページが読み取られて、仮想境界コードサイズおよび/または位置を検証することができる。決定ブロック620で、のページサイズも検出されなかった場合、いずれかの仮想境界コード(VBC)が検出されたかどうかに関するチェックが、さらなる決定ブロック622で行われる。622でVBCが検出された場合、そのオフセット(位置およびサイズ)を明記するVBCが、624で読み取られ、次のページに移り、ページサイズ検出決定ブロック620に戻る。622で、VBCが検出されなかった場合、プロセスはブロック616で終了し、これはどのブートローダ(BL)も見つけられなかったことを意味する。決定ブロック620でページサイズが検出された場合、仮想ブロックサイズをページサイズで除算することによって、仮想ブロックごとのページの数に関する判断が行われ得る。ROMはこの判断を使って、次の仮想ブロックに進む前に、どのくらいの量の固定されたページが読み取られる必要があるかを知ることができる。ページサイズが検出されると、第1の仮想ブロック中のデータイメージがブロック630でアクセスされ、コピーもされ得る。したがって、残りのページがその第1の仮想ブロックから読み出されてよく、結果として、データイメージの第1の仮想ブロック内のデータすべてがアクセスされる。プロセスは次いで、VBCを求めて、次の仮想ブロックの走査に進み得る。640でVBCが検出されなかった場合、次の仮想ブロックは不良および/または破損であり得る。したがって、ブロック642で、プロセスは仮想境界コードを求めて、次の仮想ブロックを走査する。プロレスが無限ループに陥らないようにするために、何個の仮想ブロックがチェックされればよいかについて、閾値(タイムアウト機能とも呼ばれる)が使われ得る。したがって、15個の仮想ブ

ロックなどの閾値が使われてよく、644で、その閾値に達した場合、このことは、BLが見つからなかったことを意味し得るので、プロセスは616で終了する。仮想ブロック閾値に達していない場合、サイクルはブロック640で仮想境界コードが検出されるまで、ブロック640に続くか、あるいはブロック640に戻る。640でVBCが検出されると、次の仮想ブロック内のすべてのデータが、650でアクセスされ得る。その次の仮想ブロックは、VBCが見つかりアクセスされ得る仮想ブロックに対応する。660で、完全なデータイメージ(たとえば、完全なBL)が読み取られたかどうかに関して、判断が行われ得る。660で完全なBLが読み取られていない場合、プロセスはブロック642で、上述したように、そこからさらに進むとき、仮想境界コードを求めて次の仮想ブロックを走査する。660で完全なBLが読み取られている場合、670でBLは認証され、実行されてよく、したがって、第1および第2の仮想ブロック中のデータイメージによって表されるデータからBLをロードする。ロードされたBLは、マーカとしてのみ意図されているとともにロードされる必要がない仮想境界コードバイトをスキップしてよい(含める必要はない)。

【００４２】

図6に示す態様方法は、第1および第2の仮想ブロック、または後続仮想ブロックの間に不良ブロックが配列されているとき、データロードルーチン(たとえば、一次ブートローダ)が、メモリ中の良好ブロックまたは不良ブロックを最初に識別することなく、データイメージ全体(たとえば、二次ブートローダ)を読み取ることができるように、不良ブロックがどのようにスキップされるかを示す。したがって、不良仮想ブロック中で仮想境界コードが検出されないので、データロードルーチンは第1の仮想ブロックに続く不良ブロックをスキップする。また、第1の仮想境界コードは、それらのバイトがマーカとしてのみ必要とされ得るので、二次ブートローダの実行から除外されてよい。

【００４３】

追加態様は、たとえばワイヤレスデバイス向けのOver The Air(OTA)ダウンロードを通した、フェールセーフな更新プロビジョニングを含む。この点において、フラッシュメモリを含むスマートフォンなどリードライトメモリを含む多くのデバイスは、しばしば、ファームウェアに対するアップグレードまたは更新を要求する。開示する不良ブロック管理設計はそのようなアップグレードを、上述した技法の拡張を使う、信頼できる/フェールセーフなやり方で可能にする。最初に、システムは現行のブートローダが存在するブロックを検出する。次いで、システムはより新しい/置換えブートローダが書き込まれるのに利用可能な、1つまたは複数の追加良好ブロックに進行する。その後、新規良好ブロックロケーションが、アップグレード/更新プロセスに何かが割り込んだ場合のバックアップとして使われ、新規ブートローダがプログラムされ得る。次いで、たとえばブロック0において開始する古いブートローダが初期ロケーションにおいて消去され、新規二次ブートローダがブロック0において、および必要な場合は後続良好ブロックにおいてプログラムし直され得る。

【００４４】

図7は、様々な態様による、NANDデバイス上でイメージを改訂する方法の例を示す。このアップグレード方法は、上述した仮想境界コードを使う仮想ブロック指定に十分に準拠する。初期状態70において、システムは現行のブートローダが存在する第1のブロック内のデータにアクセスする。それらのブロックは、プリアンブルメモリコード10をもつプリアンブル5、およびイメージヘッダ12を含み得る。図示した例において、ブートローダは2つの初期ブロック中に配列された2つの部分15a、15bに分割され、これらのブロックはNANDデバイスに依存して、仮想ブロックまたは実ブロックに対応してよく、本明細書の態様による仮想境界コード14を含み得る。代替として、オリジナルブートローダは従来技術による単一の連続ブートローダであってよく、これは本明細書の方法に従って置き換えられ得る。この段階で、システムは新規データイメージ用に何個のブロックが必要とされるかをさらに判断するために、任意のブートローダ部分15a'、15b'、新規プリアンブルメモリコード10'をもつ新規プリアンブル5'、新規イメージヘッダ12'および任意の必要な新規VBC14'を含む実ブロック(または、存在する場合は仮想ブロック)のサイズを判断することが

できる。システムは、次の利用可能良好ブロックを見つけるために、フラッシュメモリデバイス上を前方走査し得る。新規ブートローダ用に複数の良好ブロックが必要とされる場合、システムは新規データイメージを収容するのに十分な利用可能良好ブロックが見つかるまで、フラッシュメモリデバイス上を前方走査し続けてよい。72に示す状態において、システムは次いで、識別された利用可能良好ブロック中に、ブートローダ部分15a'、15b'、新規プリアンブルメモリコード10'をもつ新規プリアンブル5'、新規イメージヘッダ12'および新規VBC14'を含む新規データイメージを記憶する。本発明の態様によると、利用可能良好ブロックは、まさに次の連続して利用可能な良好ブロックである必要はない。図示した例において、この方法の一部として変更される古いブロックと新規ブロックとの間のバッファを提供するために、少なくとも2つのブロックがスキップされる。少なくとも2～3個のブロックを残しておくことにより、方法は、初期ブートローダのサイズの今後の増大を考慮に入れることができ、そうすることにより、現在のシステムにとって必要とされる1つまたは2つの初期ブロックよりも多くの初期ブロックに対処することができる。また、ある程度の空間を残すことにより、デバイスのそれらの初期ブロック中の不良ブロックを避けることもできる。古い方のブートローダを消去する前に、新規データイメージに対して検証チェックが実施され得る。72に示す段階で、システムは依然として、新規データイメージが正しくロードしなかった場合のフェールセーフが破損され、あるいは使用できないとき、古い方のブートローダ部分15a、15bからブートする。新規データイメージがロードされる(かつ、任意選択で認証される)と、74に示す状態において、古いブートローダ、または少なくともその初期部分が消去され得る。具体的には、ブートローダ15aの古いプリアンブル5、古いイメージヘッダ12または初期部分のうちの少なくとも1つが消去される場合、古いブートローダは、それらの要素を探している一次ブートローダによってスキップされる。74に示す第2のブロックが、消去されるものとして示されていない場合、そのブロックは、次のステップに進む前に消去されてよい。その後、76に示す状態において、システムは次いで、段階72であらかじめ記憶された新規データイメージとほとんど同一である新規データイメージをもつ第1のブロックを記憶する。したがって、段階76で、新規データイメージは第1のブロックをマーキングする際にエラーがあった場合に、バックアップとして働くNANDデバイス上の2つの場所で見つけられ得る。初期ブロック中のデータイメージが認証されると、改訂する方法は完結してよい。より新しいデータイメージの第2のコピーは、バックアップとしてNANDデバイス上に残され得る。代替として、方法は次いで、さらに78に示す状態において、新規データイメージの第2のコピーを、72に示す状態においてマーキングされた良好ブロックから消去してよい。

【００４５】

図7において、対象データイメージに関連しない様々なブロックは、「消去済み」と示される。ただし、それらのブロックは、ブランクである必要も、消去済みである必要もない。本明細書の態様によると、方法はプリアンブルおよび仮想境界コードを使って、ブートローダのサイズおよびロケーションを判断することができ、したがって「消去済み」と明記されたブロックは、ブートローダイメージに関係しないデータブロックを示すことを意図している。

【００４６】

図8は、リードライトメモリ上のデータイメージを書き換える態様方法800を示す。ブロック810で、リードライトメモリの少なくとも1つの第1の仮想ブロック中で、初期データイメージがアクセスされる。少なくとも1つの第1の仮想ブロックは、複数の第1の仮想ブロックを含み得る。ブロック820で、方法は、リードライトメモリ中で利用可能な少なくとも1つの第2の仮想ブロックを求めて、少なくとも1つの第1の仮想ブロックを前方走査する。したがって、少なくとも1つの第1の仮想ブロックが複数の仮想ブロックを含んでいた場合、方法800で定義される「第2の仮想ブロック」は、それらの先行する第1の仮想ブロックに続くことになる。ブロック830で、新規データイメージの第1の部分が、第2の仮想ブロック中に記憶される。ブロック840で、方法は、第2の仮想ブロックから第3の仮想ブロックまで前方走査する。第3の仮想ブロックが位置特定されると、850で、新規データイ

メージの第2の部分が第3の仮想ブロック中に記憶される。ブロック860は、少なくとも1つの第1の仮想ブロックの少なくとも一部分内の初期データイメージを消去することを含む。初期データイメージ全体またはその一部分が、ブロック860で消去され得る。たとえば、ブロック860で、初期データイメージが、複数の第1の仮想ブロックから消去され得る。870で、VBCを含む新規データイメージの第1の部分および第2の部分が、少なくとも1つの第1の仮想ブロック中に記憶される。ブロック880で、新規データイメージの第1の部分は、第2の仮想ブロックから消去される。ブロック890で、新規データイメージの第2の部分は、第3の仮想ブロックから消去される。

【００４７】

プリアンブルブロックが、様々なデバイス特性を導出するために使われ得る。有効なプリアンブルブロックが最初に見つからない場合、プロセスは有効なものを見つけようとして、ブロックを横断し続けてよい。プリアンブルブロックは、フラッシュ読出しコマンドの一部として読み取られたブロックの初期バイト(たとえば、最初の12バイト)中で、仮想境界コードなど、特定のコードをチェックすることによって検出され得る。開示する態様によると、汎用NANDデバイス幅操作技法が望ましい場合がある。そのようなものとして指定されてきた検出仮想ブロックをカウントするためのアルゴリズムが使われ得る。この技法を使って、仮想ブロックカウントが取得されない場合、プロセスは二次ブートローダが見つからなかったと結論づけて終了してよい。または、このように、仮想境界コードは仮想ブロックおよびNANDデバイス自体のサイズを判断するのに使われ得る。また、ECC検出およびページサイズ検出は、本明細書で開示した方法の一部として実装され得る。たとえば、自動ページサイズ検出アルゴリズムは、どのくらいの量のページがデバイス上にあるかを判断する、したがってそのページサイズを判断するために、特定の仮想境界コードでマーキングされたページの数を読み取ることができる。

【００４８】

さらに、様々な実施形態は、様々なモバイルコンピューティングデバイスにおいて、および/またはそれらのいずれかとともに実装することができ、その例を、携帯電話の形で図9に示す。典型的なモバイルコンピューティングデバイス900は、図9に示された構成要素を共通に有する。たとえば、モバイルコンピューティングデバイス900は、内部メモリ902および、たとえば抵抗検知タッチスクリーン904、容量検知タッチスクリーン、赤外線検知タッチスクリーン、音響/圧電性検知タッチスクリーンなどのタッチ面入力デバイス/ディスプレイ903に結合されたプロセッサ901を含み得る。モバイルコンピューティングデバイス900は、プロセッサ901に結合されたワイヤレスデータリンクおよび/または携帯電話送受信機920に接続された、電磁放射を送受信するための無線/アンテナ906を有する場合がある。モバイルコンピューティングデバイス900は、デバイスのロケーションを判断するための、プロセッサ901に結合されたGPS受信機も含み得る。モバイルコンピューティングデバイス900はまた、ユーザ入力を受信するための物理ボタン908を含むことができる。

【００４９】

様々な実施形態は、タブレットコンピュータなど、様々なコンピューティングデバイスにおいて、および/またはそれらのうちのいずれかとともに実装することができ、その例を図10に示す。たとえば、ワイヤレスデバイス1000は、内部メモリ1004および1006に結合されたプロセッサ1002を含み得る。内部メモリ1004および1006は、揮発性メモリまたは不揮発性メモリであってもよく、また、セキュアメモリおよび/もしくは暗号化メモリ、または非セキュアメモリおよび/もしくは非暗号化メモリ、またはそれらの任意の組合せであってもよい。プロセッサ1002は、タッチスクリーンディスプレイ1016(たとえば、抵抗検知タッチスクリーン、容量検知タッチスクリーン、赤外線検知タッチスクリーンなど)、または従来のボタン(たとえば、1012aおよび1012b)ならびに非タッチスクリーンディスプレイなどのユーザインターフェースにも結合され得る。さらに、ワイヤレスデバイス1000は、プロセッサ1002が1つまたは複数の有線ネットワークまたはワイヤレスネットワークを介して他のコンピューティングデバイスと通信することを可能にするように構成され

た1つまたは複数のネットワークトランシーバを含み得る。具体例として、ワイヤレスデバイス1000のネットワークトランシーバは、プロセッサ1002に結合された1つまたは複数のワイヤレスデータリンクトランシーバおよび/または携帯電話トランシーバ1010に接続され得る、電磁放射を送受信するための1つまたは複数のアンテナ1018を含み得る。ワイヤレスデバイス1000はまた、ユーザ入力を受信するための物理ボタン1012aおよび1012bを含み得る。

【００５０】

上記で説明した様々な実施形態はまた、図11に示すラップトップコンピュータ1100などの様々なパーソナルコンピューティングデバイス内に、および/またはそれらとともに実装され得る。多くのラップトップコンピュータは、コンピュータのポインティングデバイスとして働くタッチパッドのタッチ面1107を含み、したがって、タッチスクリーンディスプレイを備える上述のモバイルコンピューティングデバイスに実装されるものと同様のドラッグジェスチャ、スクロールジェスチャ、およびフリックジェスチャを受信することができる。ラップトップコンピュータ1100は通常、揮発性メモリと、フラッシュメモリデバイス1102のような大容量の不揮発性メモリとに結合されたプロセッサ1101を含む。ラップトップコンピュータ1100はまた、プロセッサ1101に結合されたフロッピー(登録商標)ディスクドライブおよびコンパクトディスク(CD)ドライブを含み得る。ラップトップコンピュータ1100は、プロセッサ1101が、1つまたは複数の有線ネットワークまたはワイヤレスネットワークを介して他のコンピューティングデバイスと通信することを可能にするように構成された、プロセッサ1101に結合されたいくつかのネットワークトランシーバまたはネットワークコネクタポートも含み得る。具体例として、ラップトップコンピュータ1100のネットワークトランシーバは、イーサネット(登録商標)、USBまたはファイアワイヤ(登録商標)コネクタソケット/トランシーバ、電磁放射を送受信するための1つまたは複数のアンテナに結合されたWi-Fiおよび/またはセルラーデータネットワークトランシーバなどの1つまたは複数のワイヤレスモデムトランシーバを含み得る。ラップトップコンピュータ1100は、プロセッサ1101を、将来開発され得るネットワークに結合するための他のタイプのネットワーク接続回路も含み得る。ノートブック構成では、コンピュータのハウジングはタッチパッドタッチ面1107、キーボード1108、およびディスプレイ1109を含み、すべてはプロセッサ1101に結合される。コンピューティングデバイスの他の構成はよく知られているように、(たとえば、USB入力を介して)プロセッサに結合されたコンピュータマウスまたはトラックボールを含んでよく、それらはまた、様々な実施形態とともに使用され得る。

【００５１】

本明細書で説明する様々な実施形態におけるプロセッサは、上記で説明した様々な実施形態の機能を含む、様々な機能を実施するためのソフトウェア命令(アプリケーション)によって構成され得る、任意のプログラマブルマイクロプロセッサ、マイクロコンピュータ、または1つもしくは複数の多重プロセッサチップであり得る。いくつかのデバイスでは、1つのプロセッサをワイヤレス通信機能専用とし、1つのプロセッサを他のアプリケーションの稼働専用とするなど、複数のプロセッサが設けられてもよい。典型的には、ソフトウェアアプリケーションは、アクセスされプロセッサにロードされる前に、内部メモリに記憶され得る。プロセッサは、アプリケーションソフトウェア命令を記憶するのに十分な内部メモリを含み得る。多くのデバイスでは、内部メモリは、揮発性メモリ、またはフラッシュメモリなどの不揮発性メモリ、または両方の組合せであり得る。本明細書では、メモリへの一般的な言及は、内部メモリまたはデバイスに差し込まれるリムーバブルメモリと、プロセッサ自体の内部のメモリとを含む、プロセッサによってアクセス可能なメモリを指す。

【００５２】

上記の方法の説明およびプロセスフロー図は、単に説明のための例として提供され、様々な実施形態のブロックが提示された順序で実施されなければならないことを要求または暗示するものではない。当業者によって諒解されるように、上記の実施形態におけるブロ

ックの順序は、任意の順序で実施することができる。

【００５３】

「その後」、「次いで」、「次に」などの用語は、ブロックの順序を限定するものではなく、これらの用語は、単に方法の説明を通して読者を案内するために使用されているにすぎない。さらに、たとえば、冠詞「a」、「an」または「the」を使用する単数形での請求要素へのいかなる言及も、その要素を単数形に限定するものとして解釈されるべきではない。

【００５４】

本明細書で開示する実施形態に関して説明する様々な例示的論理ブロックおよびプロセスフロー図は、電子ハードウェア、コンピュータソフトウェア、またはその両方の組合せとして実装され得る。ハードウェアとソフトウェアのこの互換性を明確に示すために、様々な例示的な構成要素、ブロック、モジュール、回路、およびブロックを、上記では概してそれらの機能性に関して説明した。そのような機能性をハードウェアとして実装するか、またはソフトウェアとして実装するかは、特定の適用例および全体的なシステムに課される設計の制約に依存する。当業者は、説明した機能性を特定の適用例ごとに様々な方法で実装し得るが、そのような実装の決定は、本発明の範囲からの逸脱を生じるものと解釈すべきではない。

【００５５】

本明細書で開示した実施形態に関して説明した様々な例示的論理、論理ブロック、モジュール、および回路を実装するために使用されるハードウェアは、汎用プロセッサ、デジタル信号プロセッサ(DSP)、特定用途向け集積回路(ASIC)、フィールドプログラマブルゲートアレイ(FPGA)もしくは他のプログラマブル論理デバイス、個別ゲートもしくはトランジスタ論理、個別ハードウェア構成要素、または本明細書で説明した機能を実施するように設計されたそれらの任意の組合せを用いて実装または実施され得る。汎用プロセッサはマイクロプロセッサであり得るが、代替として、プロセッサは任意の従来のプロセッサ、コントローラ、マイクロコントローラ、または状態機械であり得る。プロセッサはまた、コンピューティングデバイスの組合せ、たとえば、DSPおよびマイクロプロセッサの組合せ、複数のマイクロプロセッサ、DSPコアと連携する1つもしくは複数のマイクロプロセッサ、または任意の他のそのような構成として実装され得る。代替的に、いくつかのブロックまたは方法が、所与の機能に固有の回路によって実施され得る。

【００５６】

1つまたは複数の例示的な態様では、記載された機能は、ハードウェア、ソフトウェア、ファームウェア、またはそれらの任意の組合せに実装することができる。その機能はソフトウェアで実装される場合、1つまたは複数の命令またはコードとして、非一時的コンピュータ可読記憶媒体または非一時的プロセッサ可読記憶媒体に記憶され得る。本明細書で開示された方法またはアルゴリズムのステップは、非一時的コンピュータ可読記憶媒体またはプロセッサ可読記憶媒体上に常駐し得るプロセッサ実行可能ソフトウェアモジュール内で具現化され得る。非一時的コンピュータ可読記憶媒体またはプロセッサ可読記憶媒体は、コンピュータまたはプロセッサによってアクセスされ得る任意の記憶媒体であってよい。限定ではなく例として、そのような非一時的コンピュータ可読媒体またはプロセッサ可読媒体は、RAM、ROM、EEPROM、フラッシュメモリ、CD-ROMもしくは他の光ディスク記憶装置、磁気ディスク記憶装置もしくは他の磁気記憶デバイス、または命令もしくはデータ構造の形式で所望のプログラムコードを記憶するために使用され、コンピュータによってアクセスされ得る任意の他の媒体を含み得る。本明細書で使用する場合、ディスク(disk)およびディスク(disc)は、コンパクトディスク(CD)、レーザーディスク(登録商標)、光ディスク、デジタル多用途ディスク(DVD)、フロッピー(登録商標)ディスク、およびブルーレイディスクを含み、ディスク(disk)は通常、磁気的にデータを再生し、ディスク(disc)はレーザーで光学的にデータを再生する。上記の組合せも非一時的コンピュータ可読媒体およびプロセッサ可読媒体の範囲内に含まれる。加えて、方法またはアルゴリズムの動作は、コンピュータプログラム製品に組み込まれ得る、非一時的プロセッサ可読媒体およ

び/またはコンピュータ可読媒体上のコードおよび/または命令の、1つまたは任意の組合せ、またはそのセットとして存在し得る。

【００５７】

同じ基本的根底をなす機構および方法を依然として使用する一方、開示される実施形態の態様の多くの可能な変更および組合せが使用できることを、当業者は認識されよう。上記の記述は説明の目的で、特定の実施形態を参照しながら記述されてきた。しかしながら、上で示した論述はすべてを網羅するものでも、あるいは本開示を開示された厳密な形態に限定するものでもない。多くの修正および変形が、上記の教示に鑑みて可能である。本開示の原理およびその実際の適用について説明するために、また企図される特定の用途に合わせて様々な修正を加え、本開示および様々な実施形態を他の当業者が最善の形で利用できるように、実施形態が選択され、説明されている。したがって本開示は、本明細書に示し記載した、開示した技術の実施形態および個々の態様に限定されるものではなく、以下の特許請求の範囲ならびに本明細書で開示した原理および新規の特徴に一致する最大の範囲を与えられるものである。

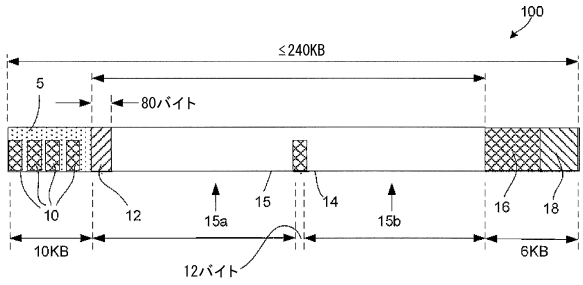
【符号の説明】

【００５８】

5 プリアンブル
5' 新規プリアンブル
10 メモリコード
10' 新規プリアンブルメモリコード
12 イメージヘッダ
12' 新規イメージヘッダ
14 仮想境界コード
14' 新規VBC
15 ブートローダ、BL
15a ブートローダの第1の部分、第1のブートローダ部分
15a' ブートローダ部分
15b ブートローダの第2の部分、第2のブートローダ部分
15b' ブートローダ部分
15c BLの第3の部分
16 証明書
18 パディング
100 NANDデバイス
900 モバイルコンピューティングデバイス
901 プロセッサ
902 内部メモリ
903 ディスプレイ
904 抵抗検知タッチスクリーン
906 無線/アンテナ
908 物理ボタン
920 ワイヤレスデータリンク、携帯電話送受信機
1000 ワイヤレスデバイス
1002 プロセッサ
1004 内部メモリ
1006 内部メモリ
1016 タッチスクリーンディスプレイ
1018 アンテナ
1012a 物理ボタン
1012b 物理ボタン
1100 ラップトップコンピュータ

1101　プロセッサ
1102　フラッシュメモリデバイス
1107　タッチ面
1108　キーボード
1109　ディスプレイ

【図１】



【図２】

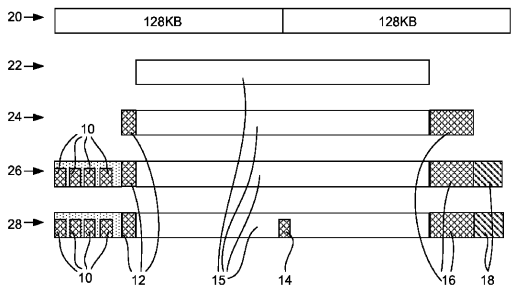
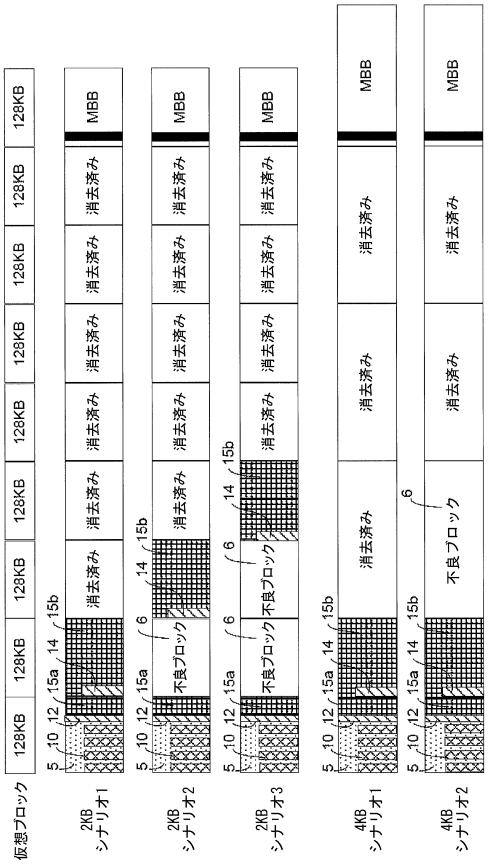
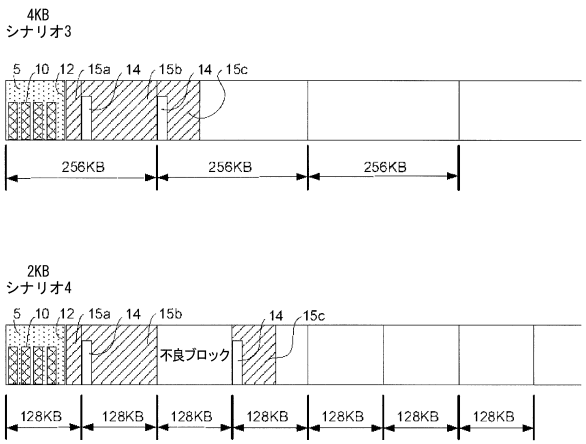


FIG. 2

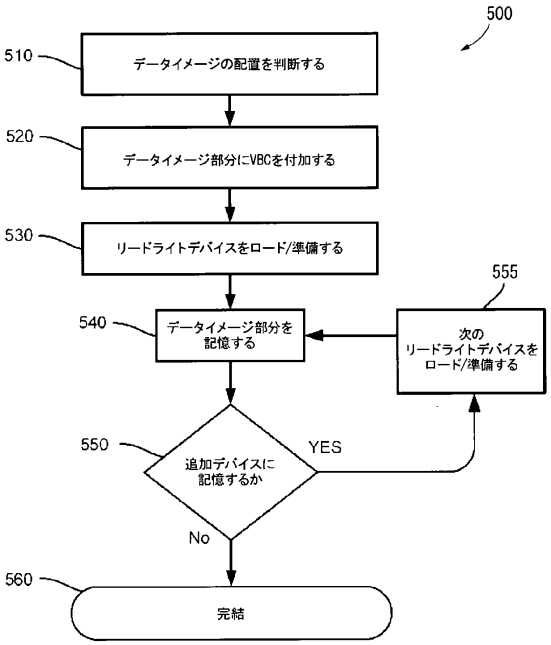
【図３】



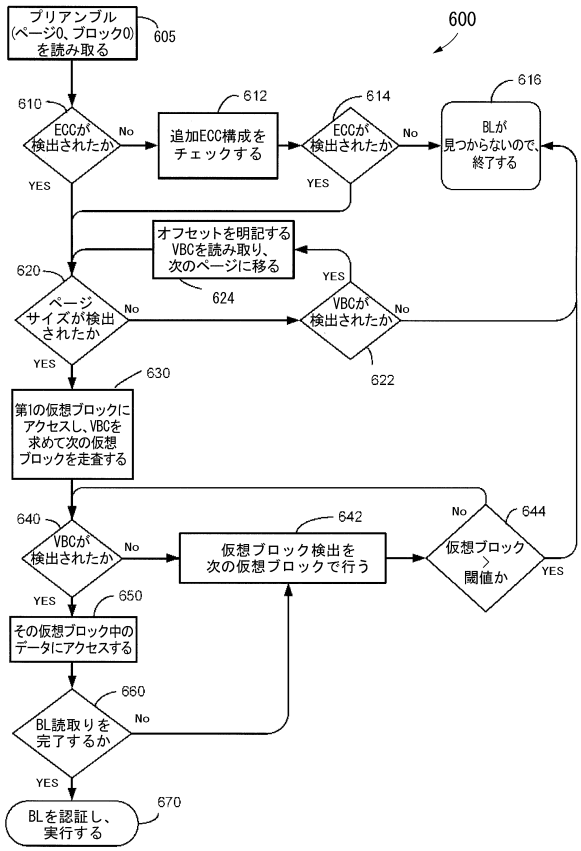
【図４】


4KB
シナリオ3
5
10
12
15a
14
15b
14
15c
256KB
256KB
256KB
2KB
シナリオ4
5
10
12
15a
14
15b
14
15c
不良ブロック
128KB
128KB
128KB
128KB
128KB
128KB
128KB

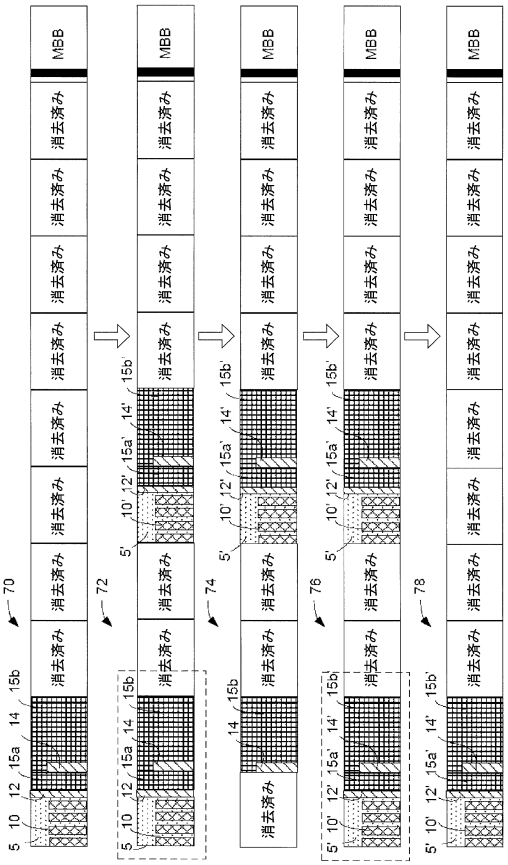
【図５】


500
510
データタイメージの配置を判断する
520
データタイメージ部分にVBCを付加する
530
リードライトデバイスをロード/準備する
555
次の
リードライトデバイスを
ロード/準備する
540
データタイメージ部分を
記憶する
550
追加デバイスに
記憶するか
YES
No
560
完結

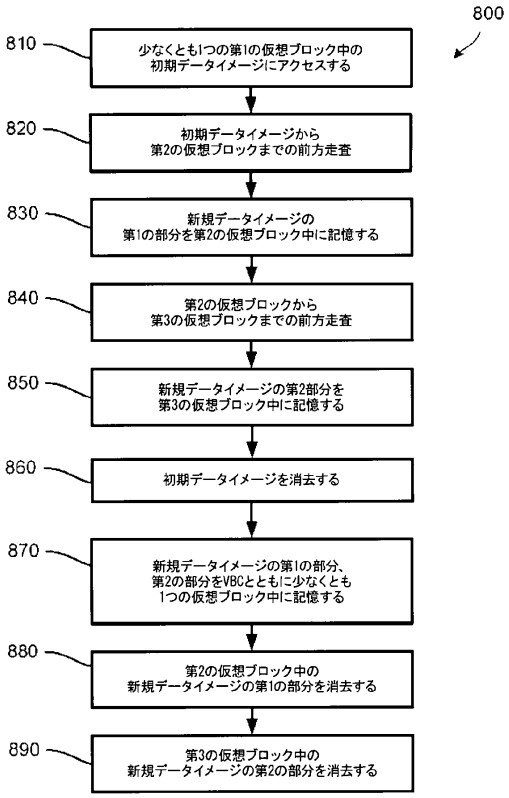
【図６】


600
プリアンブル
(ページ0、ブロック0)
を読み取る
605
610
ECCが
検出されたか
No
YES
612
追加ECC構成を
チェックする
614
ECCが
検出されたか
No
YES
616
BLが
見つからないので、
終了する
オフセットを明記する
VBCを読み取り、
次のページに移る
624
620
ページ
サイズが検出
されたか
No
YES
VBCが
検出されたか
No
YES
622
630
第1の仮想ブロックに
アクセスし、VBCを
求めて次の仮想
ブロックを走査する
640
VBCが
検出されたか
No
YES
642
仮想ブロック検出を
次の仮想ブロックで行う
No
644
仮想ブロック
閾値か
YES
650
その仮想ブロック中の
データにアクセスする
660
BL読取りを
完了するか
No
YES
670
BLを認証し、
実行する

【図７】


MBB
消去済み
70
72
74
76
78
5
10
12
15a
14
15b
5'
10'
12'
15a'
14'
15b'

【図８】



【図９】

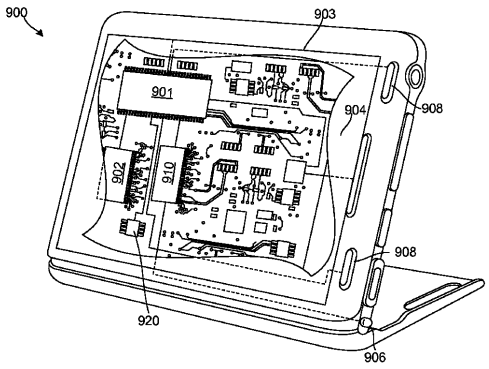


FIG. 9

【図１０】

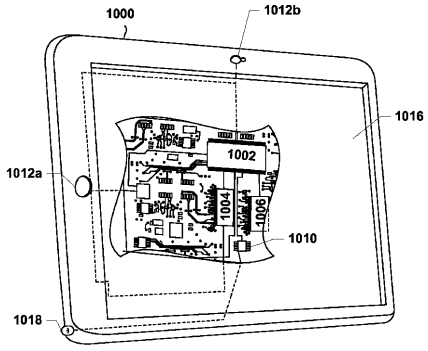


FIG. 10

【図１１】

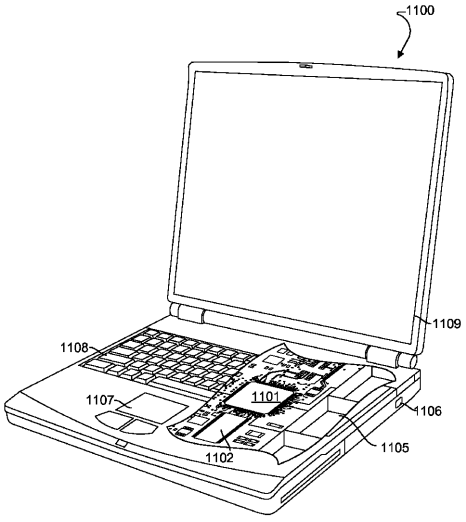


FIG. 11

フロントページの続き

(72)発明者　ベニシュ・バブ
アメリカ合衆国・カリフォルニア・９２１２１・サン・ディエゴ・モアハウス・ドライヴ・５７７５
(72)発明者　ターラ・ナンダキショア・エラーラ
アメリカ合衆国・カリフォルニア・９２１２１・サン・ディエゴ・モアハウス・ドライヴ・５７７５
(72)発明者　アシュワニ・クマール
アメリカ合衆国・カリフォルニア・９２１２１・サン・ディエゴ・モアハウス・ドライヴ・５７７５

審査官　後藤　彰

(56)参考文献　国際公開第２００８／０２６４６６（WO，Ａ１）
特開２００５－３３９４３８（ＪＰ，Ａ）
特開２０１２－１７３７７８（ＪＰ，Ａ）
特表２０１０－５１７１３１（ＪＰ，Ａ）
特開２００７－２９９２４９（ＪＰ，Ａ）
特開２００６－０３９７７２（ＪＰ，Ａ）
特開２００４－１１８４０７（ＪＰ，Ａ）
特開２００４－０９４６２８（ＪＰ，Ａ）
特開平９－２６５４２７（ＪＰ，Ａ）
米国特許出願公開第２０１２／００３０６９１（ＵＳ，Ａ１）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　１２／１６
Ｇ０６Ｆ　１２／００
Ｇ０６Ｆ　１２／０６