(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2012-198786
(P2012-198786A)
(43)公開日　平成24年10月18日(2012.10.18)

| (51)Int.Cl. | F I | テーマコード（参考） |
|---|---|---|
| G06F 12/00 (2006.01) | G06F 12/00 510B | |

審査請求 未請求　請求項の数 8　ＯＬ　（全 33 頁）

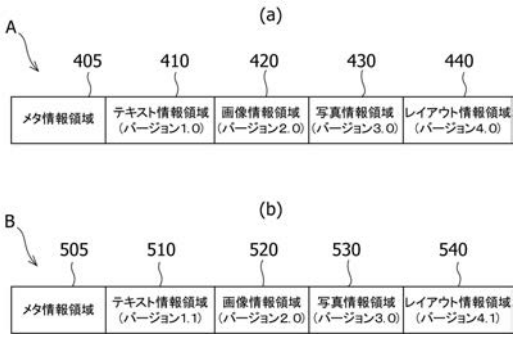
(21)出願番号　特願2011-62832(P2011-62832)
(22)出願日　平成23年3月22日(2011.3.22)

(71)出願人　000005223
富士通株式会社
神奈川県川崎市中原区上小田中４丁目１番１号
(74)代理人　100078330
弁理士　笹島　富二雄
(72)発明者　野口　泰生
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内
(72)発明者　小沢　年弘
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内
(72)発明者　大江　和一
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

最終頁に続く

(54)【発明の名称】ファイル処理プログラム及び装置

(57)【要約】
【課題】冗長性除去機能を有する記憶装置において、冗長性をさらに除去する。
【解決手段】第１の記憶装置から読み出された一つのファイル（Ａ）を、同一の内容を有するファイルを別領域に書き込むことを回避する冗長性除去機能を有する第２の記憶装置に書き込む際に、前記一つのファイルについて、それぞれ別個に変更可能な複数のデータ領域（４１０～４４０）を検出し、前記一つのファイルを前記複数のデータ領域ごとに、別個の複数のファイルに分割し、分割した該複数のファイルを前記第２の記憶装置に書き込む。このような技術によれば、電子文書という単位で冗長性の有無を判断するのではなく、それぞれ別個に内容を変更することができる部品という単位で冗長性の有無を判断することができる。したがって、電子文書という単位で冗長性の有無を判断した場合には除去しきれない冗長性を除去することができる。
【選択図】　図４



【特許請求の範囲】
【請求項１】
第１の記憶装置から読み出された一つのファイルを、同一の内容を有するファイルを別領域に書き込むことを回避する冗長性除去機能を有する第２の記憶装置に書き込む際に、
前記一つのファイルについて、それぞれ別個に変更可能な複数のデータ領域を検出し、
前記一つのファイルを前記複数のデータ領域ごとに、別個の複数のファイルに分割し、
分割した該複数のファイルを前記第２の記憶装置に書き込む、
処理をコンピュータに実行させるファイル処理プログラム。
【請求項２】
前記別個の複数のファイルに分割する処理は、
前記データ領域ごとに対応した分割ファイルを前記第２の記憶装置にそれぞれ作成し、
各データ領域の内容を前記分割ファイルにそれぞれ書き込み、
前記一つのファイルに対応するマスタファイルを前記第２の記憶装置上に作成し、
前記分割ファイルのそれぞれの保存場所を前記マスタファイルに書き込む
処理である、請求項１に記載のファイル処理プログラム。
【請求項３】
前記マスタファイルを読み込み、
前記第１の記憶装置上に復元ファイルを作成し、
前記読み込まれたマスタファイルの内容に基づいて前記分割ファイルを１つ読み込み、
前記読み込まれた分割ファイルの内容を前記復元ファイルに書き込み、
全ての分割ファイルの内容が前記復元ファイルに書き込まれるまで、前記分割ファイルを１つ読み込む処理と読み込まれた分割ファイルの内容を前記復元ファイルに書き込む処理とを繰り返す、
前記一つのファイルを前記復元ファイルとして復元する処理をコンピュータにさらに実行させる請求項２に記載のファイル処理プログラム。
【請求項４】
前記別個の複数のファイルに分割する処理は、
前記第２の記憶装置上に第２のファイルを作成し、前記データ領域のうちの１つのデータ領域の内容を前記第２のファイル内のサブファイルとして書き込み、
前記書き込まれたサブファイルに続けて当該サブファイルの区切りであることを示す情報を書き込み、
前記データ領域の全てが前記第２のファイル内のサブファイルとして書き込まれるまで、前記サブファイルとして書き込む処理と当該サブファイルの区切りであることを示す情報を書き込む処理とを繰り返す
処理である、請求項１に記載のファイル処理プログラム。
【請求項５】
前記第１の記憶装置上に復元ファイルを作成し、
前記第２のファイルの先頭から最初のサブファイルの区切りであることを示す情報まで前記第２のファイルを読み込むことにより、前記第２のファイル内のサブファイルを１つ読み込み、
前記読み込まれたサブファイルの内容を前記復元ファイルに書き込み、
前記第２のファイル内の全てのサブファイルが読み込まれるまで、前記第２のファイル内の次のサブファイルの区切りであることを示す情報まで前記第２のファイルを読み込むことにより前記第２のファイル内の次のサブファイルを読み込む処理と、読み込まれたサブファイルの内容を前記復元ファイルに書き込む処理とを繰り返す、
前記一つのファイルを前記復元ファイルとして復元する処理をコンピュータにさらに実行させる請求項４に記載のファイル処理プログラム。
【請求項６】
第１の記憶装置から読み出された一つのファイルを、同一の内容を有するファイルを別領域に書き込むことを回避する冗長性除去機能を有する第２の記憶装置に書き込む際に、

前記一つのファイルについて、それぞれ別個に変更可能な複数のデータ領域を検出し、
前記一つのファイルを前記複数のデータ領域ごとに、別個の複数のファイルに分割し、
分割した該複数のファイルを前記第２の記憶装置に書き込むファイル処理装置。
【請求項７】
前記別個の複数のファイルに分割する処理は、
前記データ領域ごとに対応した分割ファイルを前記第２の記憶装置にそれぞれ作成し、
各データ領域の内容を前記分割ファイルにそれぞれ書き込み、
前記一つのファイルに対応するマスタファイルを前記第２の記憶装置上に作成し、
前記分割ファイルのそれぞれの保存場所を前記マスタファイルに書き込む
処理である、請求項６に記載のファイル処理装置。
【請求項８】
前記別個の複数のファイルに分割する処理は、
前記第２の記憶装置上に第２のファイルを作成し、前記データ領域のうちの１つのデータ領域の内容を前記第２のファイル内のサブファイルとして書き込み、
前記書き込まれたサブファイルに続けて当該サブファイルの区切りであることを示す情報を書き込み、
前記データ領域の全てが前記第２のファイル内のサブファイルとして書き込まれるまで、前記サブファイルとして書き込む処理と当該サブファイルの区切りであることを示す情報を書き込む処理とを繰り返す
処理である、請求項６に記載のファイル処理装置。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、冗長性を除去する技術に関する。
【背景技術】
【０００２】
情報技術の進歩に伴い、コンピュータを用いて電子文書を編集することが日常的に行われている。このような電子文書は、テキスト、画像、表といった部品を含んでいる。また、ある電子文書をファイルとして記憶装置に記憶させた上で、当該電子文書に変更を加えた電子文書を別のファイルとして記憶装置に記憶させることも頻繁に行われている。
【０００３】
そして、前記２つのファイルを解析することにより、前記２つの電子文書の相違点を差分として検出する文書差分検出装置が知られている。
【先行技術文献】
【特許文献】
【０００４】
【特許文献１】特開２００６－２６８６３８号公報
【発明の概要】
【発明が解決しようとする課題】
【０００５】
前記した電子文書に含まれているテキスト、画像、表といった各部品の内容は、それぞれ別個に変更することができる。例えば、前記した２つの電子文書の双方に同一の画像が含まれている一方で、テキストの内容が両者で異なる場合がある。つまり、電子文書全体として見ると前記２つの電子文書は内容が異なるものの、画像という部品単位で見ると内容が同じである。
【０００６】
このような２つの電子文書を、同一の内容を有するファイルを別領域に書き込むことを回避する冗長性除去機能を有する記憶装置に記憶させる際に、２つの電子文書が前記記憶装置内の別々の領域に書き込まれることがある。この場合、一方の文書を記憶した領域には前記画像の情報を記憶した領域があり、他方の文書を記憶した領域にも同一の画像の情

報を記憶した領域がある。すなわち、冗長性除去機能を有する記憶装置であっても、除去しきれない冗長性が存在する。そこで、本願の開示内容は、冗長性除去機能を有する記憶装置において、冗長性をさらに除去することを目的とする。

【課題を解決するための手段】

【０００７】

本願に開示するファイルを処理する技術では、第１の記憶装置から読み出された一つのファイルを、同一の内容を有するファイルを別領域に書き込むことを回避する冗長性除去機能を有する第２の記憶装置に書き込む際に、前記一つのファイルについて、それぞれ別個に変更可能な複数のデータ領域を検出する。そして、前記一つのファイルを前記複数のデータ領域ごとに、別個の複数のファイルに分割し、分割した該複数のファイルを前記第２の記憶装置に書き込む。

【発明の効果】

【０００８】

本願の開示内容によれば、冗長性除去機能を有する記憶装置において、冗長性をさらに除去することができる。

【図面の簡単な説明】

【０００９】

【図１】冗長性除去ディスクを備えたコンピュータの構成例を示す説明図である。

【図２】コンピュータのハードウェア構成例を示す説明図である。

【図３】（ａ）及び（ｂ）はプレゼンテーションの一例を示す説明図である。

【図４】（ａ）及び（ｂ）はローカルディスクに書き込まれるファイル構造の一例を示す説明図である。

【図５】（ａ）～（ｅ）は全て、冗長性除去ディスクに書き込まれた分割ファイルの構造の一例を示す説明図である。

【図６】マスタファイルの内容の一例を示す説明図である。

【図７】分割ファイルを作成して冗長性除去ディスクに書き込む処理のフローチャートの一例である。

【図８】分割ファイルを作成する処理のフローチャートの一例である。

【図９】分割ファイルから元ファイルを復元する処理のフローチャートの一例である。

【図１０】分割ファイルの内容を復元ファイルに書き込む処理のフローチャートの一例である。

【図１１】（ａ）及び（ｂ）は、同じ内容の２つの分割ファイルが別々のブロックに書き込まれないことを示す説明図である。

【図１２】（ａ）及び（ｂ）は、冗長性除去ディスクにおけるファイル構造の別の例を示す説明図である。

【図１３】冗長性除去ファイルシステムを備えたコンピュータの構成例を示す説明図である。

【図１４】（ａ）及び（ｂ）は、アンカーが挿入されたファイルの構造の一例を示す説明図である。

【図１５】アンカーが挿入されたファイルを作成して、当該ファイルを冗長性除去ファイルシステムに書き込む処理のフローチャートの一例である。

【図１６】アンカーが挿入されたファイルから元ファイルを復元する処理のフローチャートの一例である。

【発明を実施するための形態】

【００１０】

［１．概要］

以下に説明する技術では、電子文書に含まれているテキスト、画像、表といった各部品の内容をそれぞれ別個に変更することができるという点を踏まえて、次のような処理を行う。まず、ある記憶装置に書き込まれている電子文書のファイルを読み込む。そして、当該ファイルの内容を、部品ごとに分けて、冗長性除去機能を有する記憶装置に書き込む。

【００１１】
このような技術によれば、電子文書という単位で冗長性の有無が判断されるのではなく、それぞれ別個に内容を変更することができる部品という単位で冗長性の有無が判断される。したがって、電子文書という単位で冗長性の有無が判断された場合には除去しきれない冗長性を除去することができる。
【００１２】
［２．冗長性除去ディスクを備えた記憶システム］
図１は、第１の記憶装置に相当するローカルディスク２と、第２の記憶装置に相当する冗長性除去ディスク３とを備えたコンピュータ１を示している。コンピュータ１には、ＯＳ（オペレーティング・システム）１１と、アプリケーション１２と、前記冗長性除去ディスク３にファイルを書き込むためのファイル書き込みソフトウェア１３とがインストールされている。アプリケーション１２は、プレゼンテーションと呼ばれる電子文書を編集するためのアプリケーションである。アプリケーション１２としては、ワードプロセッサアプリケーション、ＤＴＰ（デスクトップパブリッシング）アプリケーション、表計算アプリケーションが挙げられる。
【００１３】
このようなアプリケーション１２によってローカルディスク２にファイルが書き込まれると、ファイル書き込みソフトウェア１３は、当該ファイルを読み込んでその内容を冗長性除去ディスク３に書き込む。その後、ファイル書き込みソフトウェア１３は、ローカルディスク２にあったファイルを削除する。つまり、ローカルディスク２はファイルを一時的に記憶させるために用いる。
【００１４】
また、ファイル書き込みソフトウェア１３は、冗長性除去ディスク３上のファイルを読み込んでその内容をローカルディスク２上に復元する処理をも行う。つまり、ファイル書き込みソフトウェア１３は、コンピュータ１を、ローカルディスク２と冗長性除去ディスク３との両者にアクセスしてファイルを処理するファイル処理装置として具現化する。
【００１５】
冗長性除去ディスク３の記憶領域は、一定のサイズの複数のブロックから構成されている。ブロックとは、ＯＳ１１がデータの読み込み及び書き込みを行う際のデータの最小の単位である。一例として、１ブロックあたりの容量は５１２バイトである。
【００１６】
冗長性除去ディスク３は、新たに書き込もうとするブロックと同じ内容のブロックを、冗長性除去ディスク３に既に書き込まれているブロックの中から検索することにより、同じ内容のブロックを別の領域に再び書き込むことはしないディスクである。このようにして、冗長性除去ディスク３は冗長性の除去を実現している。
【００１７】
図２は、コンピュータ１のハードウェア構成例を示している。コンピュータ１は、ＣＰＵ１１１と、インタフェース装置１１２と、表示装置１１３と、入力装置１１４と、ドライブ装置１１７と、メモリ装置１１５と、補助記憶装置１１６を備えており、これらがバス１１９により相互に接続されている。このうち、補助記憶装置１１６は、図１に示したローカルディスク２及び冗長性除去ディスク３に相当する。
【００１８】
コンピュータ１の機能を実現するプログラムは、ＣＤ－ＲＯＭ等の記録媒体１１８によって提供される。プログラムを記録した記録媒体１１８がドライブ装置１１７にセットされると、プログラムが記録媒体１１８からドライブ装置１１７を介して補助記憶装置１１６にインストールされる。あるいは、プログラムのインストールは必ずしも記録媒体１１８により行う必要はなく、インタフェース１１２を介してネットワーク経由で他のコンピュータからダウンロードしてインストールすることもできる。補助記憶装置１１６は、インストールされたプログラムを格納すると共に、必要なファイルやデータ等を格納する。
【００１９】

メモリ装置１１５は、プログラムの起動指示があった場合に、補助記憶装置１１６からプログラムを読み出して格納する。ＣＰＵ１１１は、メモリ装置１１５に格納されたプログラムに従ってコンピュータ１の機能を実現する。表示装置１１３はプログラムによるＧＵＩ（Graphical User Interface）等を表示する。入力装置１１４はキーボード及びマウス等であり、ユーザがアプリケーション１２を用いてプレゼンテーションを編集するなど様々な指示を入力するために用いるものである。

【００２０】

このようなハードウェア構成の元で、入力装置１１４を通してユーザの入力指示を受けたアプリケーション１２により、図３（ａ）に示したようなプレゼンテーション４が作成される。図示しているように、プレゼンテーション４は、テキスト４１と画像４２と写真４３とを含んでいる。これらテキスト４１と画像４２と写真４３とをそれぞれ、プレゼンテーション４に含まれる部品と呼ぶ。そして、これら部品は、それぞれ、プレゼンテーション４内のある位置に配置されている。つまり、プレゼンテーションは、一つまたは複数の部品と、そのプレゼンテーションにおける各部品のレイアウトとを有している。

【００２１】

そして、各部品はそれぞれバージョン番号を有している。このバージョン番号は、次回以降ユーザの指示を受けてアプリケーション１２が各部品を変更する場合に、その変更前後を区別するために便宜的に付した番号である。プレゼンテーション４が作成された時点でのテキスト４１のバージョン番号は「１．０」であり、画像４２のバージョン番号は「２．０」であり、写真４３のバージョン番号は「３．０」である。また、レイアウトにもバージョン番号を付すが、詳細は後述する。

【００２２】

さらに、ユーザの入力指示を受けたアプリケーション１２により、図３（ｂ）に示したようなプレゼンテーション５が作成される。プレゼンテーション５は、プレゼンテーション４に変更を加えたものである。一例として、プレゼンテーション４がある顧客への提案書であり、プレゼンテーション５が別の顧客への提案書である。プレゼンテーション５は、プレゼンテーション４と比較して、テキスト４１からテキスト５１にテキストの内容が変わったために、バージョン番号が「１．０」から「１．１」に変わっている。これと同時に、プレゼンテーション５におけるテキスト５１のレイアウトも変わっている。その一方で、画像４２及び写真４３の内容には変更がないため、それぞれのバージョン番号にも変更はない。

【００２３】

アプリケーション１２は、前記プレゼンテーション４をＯＳ１１を通してローカルディスク２にファイルとして書き込む。このファイルを元ファイルＡと呼ぶ。前記したように元ファイルＡは最終的に削除される。つまり、ローカルディスク２は、元ファイルＡを一時的に記憶させるために用いられる。その後、アプリケーション１２は、前記プレゼンテーション５をＯＳ１１を通してローカルディスク２に元ファイルＢとして書き込む。元ファイルＢも最終的に削除される。

【００２４】

元ファイルＡ及び元ファイルＢの構造を図４（ａ）及び図４（ｂ）にそれぞれ示している。図示しているように、元ファイルＡは、メタ情報領域４０５とテキスト情報領域４１０と画像情報領域４２０と写真情報領域４３０とレイアウト情報領域４４０とを有している。メタ情報領域４０５は、元ファイルＡのメタ情報、すなわち領域４１０～４４０のそれぞれの先端及び終端と元ファイルＡの先端との距離に関する情報が書き込まれた領域である。この距離のことをオフセットともいう。テキスト情報領域４１０は、プレゼンテーション４に含まれる部品であるテキスト４１の情報が書き込まれた領域である。画像情報領域４２０と写真情報領域４３０も同様である。また、レイアウト情報領域４４０は、プレゼンテーション４における各部品のレイアウトに関する情報が書き込まれた領域である。レイアウト情報領域４４０に書き込まれたレイアウト情報もバージョン番号を有しており、その値は「４．０」である。

【００２５】
ドキュメント４内のテキスト４１と、画像４２と、写真４３と、これらのレイアウトとは、それぞれ別個に内容を変更することができる。例えば、アプリケーション１２のユーザは、テキスト４１の内容とテキスト４１のレイアウトとを変更し、画像４２及び写真４３の内容を変更しないとすることができる。つまり、テキスト情報領域４１０と、画像情報領域４２０と、写真情報領域４３０と、レイアウト情報領域４４０とは、それぞれ別個に変更することのできる情報を有する領域である。
【００２６】
そして、元ファイルＡは、テキスト情報領域４１０と画像情報領域４２０と写真情報領域４３０とレイアウト情報領域４４０とが元ファイルＡの先頭から順に隣接した状態で保存されている。同様に、元ファイルＢはメタ情報領域５０５とテキスト情報領域５１０と画像情報領域５２０と写真情報領域５３０とレイアウト情報領域５４０とを有している。
【００２７】
再び図１を参照すると、ファイル書き込みソフトウェア１３は、ファイル分割モジュール１４を有している。このファイル分割モジュール１４は、ローカルディスク２に書き込まれた元ファイルＡを読み込む。続いて、ファイル分割モジュール１４は、読み込んだ元ファイルＡの内容を分割し、分割ファイルＡ１～Ａ５を作成し、ＯＳ１１を通して冗長性除去ディスク３に書き込む。これら分割ファイルＡ１～Ａ５の各々は、それ自体別個のファイルである。図１には、分割ファイルＡ１～Ａ５が書き込まれた冗長性除去ディスク３を示している。なお、ファイル分割モジュール１４は、元ファイルＡの内容を冗長性除去ディスク３の分割ファイルＡ１～Ａ５に書き込んだ後、元ファイルＡを削除する。
【００２８】
その後、ファイル分割モジュール１４は、ローカルディスク２に書き込まれた元ファイルＢを読み込む。続いて、ファイル分割モジュール１４は、読み込んだ元ファイルＡの内容を分割し、分割ファイルＢ１～Ｂ５を作成する。これら分割ファイルＢ１～Ｂ５の各々も、それ自体別個のファイルである。これら分割ファイルは、ＯＳ１１を通して冗長性除去ディスク３に書き込まれる。ただし、分割ファイルＢ１～Ｂ５の各々のブロックの内容が、既に冗長性除去ディスク３に書き込まれている分割ファイルＡ１～Ａ５のいずれかのブロックの内容と同じである場合は、その分割ファイルが別のブロックに再び書き込まれることはない。なお、ファイル分割モジュール１４は、分割ファイルを冗長性除去ディスク３に書き込んだ後、ローカルディスク２にある元ファイルＢを削除する。
【００２９】
分割ファイルＡ１～Ａ５の構造と、これら分割ファイルＡ１～Ａ５が書き込まれている冗長性除去ディスク３上のブロックの並びとを図５（ａ）～図５（ｅ）に示している。このうち、分割ファイルＡ２～Ａ４はプレゼンテーション４の各部品すなわちテキスト、画像、写真の内容をそれぞれ有している。そして、分割ファイルＡ５は、各部品のレイアウト情報を有している。つまり、分割ファイルＡ２～Ａ５は、それぞれ別個に変更することのできる情報を有する領域４１０～４４０のそれぞれに対応している。
【００３０】
また、分割ファイルＡ１は、冗長性除去ディスク３における分割ファイルＡ２～Ａ５の保存場所に関する情報（以下、マスタ情報とも呼ぶ）を有している。このマスタ情報は、ファイル分割モジュール１３が分割ファイルＡ２～Ａ５から元ファイルＡを復元する際に必要となる。復元については後述する。以下、分割ファイルＡ１をマスタファイルＡ１とも呼ぶことにする。マスタファイルは１つの元ファイルに対して１つ作成される。
【００３１】
マスタファイルＡ１は、第０ブロックの先端から第３ブロックにかけて書き込まれている。分割ファイルＡ２は、第１０ブロックの先端から第１６ブロックの途中にかけて書き込まれている。分割ファイルＡ３は、第２０ブロックの先端から第２６ブロックの途中にかけて書き込まれている。分割ファイルＡ４は、第３０ブロックの先端から第３７ブロックの途中にかけて書き込まれている。分割ファイルＡ５は、第４０ブロックの先端から第

４６ブロックの途中にかけて書き込まれている。なお、ブロックに付した番号は、単に複数のブロックをそれぞれ区別することを目的としたものに過ぎない。また、第４ブロックと第１０ブロックとが隣接していてもよく、第１７ブロックと第２０ブロックとが隣接していてもよい。つまり、分割ファイルＡ１～Ａ５がどのブロックに書き込まれているかが問題なのではなく、それぞれが別個のファイルとして冗長性除去ディスク３に書き込まれていればよい。

【００３２】

マスタファイルＡ１の内容を図６に示している。マスタファイルＡ１には、分割ファイルＡ２～Ａ５のそれぞれの保存場所を示すパスが書き込まれている。本実施形態では、分割ファイルＡ２～Ａ５は全て「／ｘｘ／ｙｙ／」という場所に保存されている。

【００３３】

同様に、分割ファイルＢ２～Ｂ４は、プレゼンテーション５の各部品すなわちテキスト、画像、写真の内容をそれぞれ有している。そして、分割ファイルＢ５は、各部品のレイアウト情報を有している。つまり、分割ファイルＢ２～Ｂ５は、それぞれ別個に変更することのできる情報を有する領域５１０～５４０のそれぞれに対応している。また、分割ファイルＢ１はマスタ情報を有している。分割ファイルＢ１（あるいは、マスタファイルＢ１）は１つの元ファイルＢに対して１つ作成される。

【００３４】

以上のような記憶システムの構成及びファイル構造のもとに、アプリケーション１２がローカルディスク２に書き込んだ元ファイルＡをファイル分割モジュール１４が分割ファイルＡ１～Ａ５に分割して冗長性除去ディスク３に書き込む処理手順を図７及び図８に示している。

【００３５】

まず、ステップＳ１１０にて書き込みを開始し、ステップＳ１２０にてファイル分割モジュール１４がローカルディスク２にある元ファイルＡをオープンする。なお、「オープン」という用語は、既にオープンの対象であるファイルが存在している場合には当該ファイルを開くことを意味する。また、オープンの対象であるファイルが存在していない場合には、当該ファイルを新たに作成した上で開く意味で、「オープン」という用語を今後用いる。さらに、このステップＳ１２０にて、ファイル分割モジュール１４は、メタ情報領域４０５を読み込み、元ファイルＡに領域４１０～４４０が含まれていることを検出する。

【００３６】

ステップＳ１３０にてファイル分割モジュール１４が、まだ分割ファイルに書き込まれていない１つ以上の部品あるいはレイアウト情報があるか否かを判断する。この時点では、元ファイルＡ内の領域４１０～４４０にある部品またはレイアウトのいずれの情報も分割ファイルに書き込まれていないため、この判断結果は「正」となる。

【００３７】

そして、ステップＳ１４０に進み、ファイル分割モジュール１４が分割ファイルに書き込まれていないと判断した領域４１０～４４０のうちの１つを選択する。ここでは、テキスト情報領域４１０が選択されたとする。

【００３８】

ステップＳ１５０では、ファイル分割モジュール１４が、ステップＳ１４０にて選択したテキスト情報領域４１０の内容を、冗長性除去ディスク３の「／ｘｘ／ｙｙ／」という場所に作成した分割ファイルＡ２として書き込む。このステップＳ１５０の詳細については図８を参照して後述する。

【００３９】

このようなステップＳ１３０～Ｓ１５０は、部品情報あるいはレイアウト情報の全てが分割ファイルに書き込まれるまで繰り返される。この反復により、図５に示したような分割ファイルＡ３～Ａ５が冗長性除去ディスク３の「／ｘｘ／ｙｙ／」という場所に書き込まれる。そして、最終的にステップＳ１３０の判断結果が「否」となる。

【００４０】
続いてステップＳ１６０に進み、ファイル分割モジュール１４は、冗長性除去ディスク３の「／ｘｘ／ｙｙ／」という場所にマスタファイルＡ１をオープンする。
【００４１】
ステップＳ１７０では、ファイル分割モジュール１４は、冗長性除去ディスク３における分割ファイルＡ２～Ａ５の保存場所を示すパスをマスタファイルＡ１に書き込む。このマスタファイルＡ１の内容は図６に示したとおりである。
【００４２】
ステップＳ１８０では、ファイル分割モジュール１４は、元ファイルＡ及びマスタファイルＡ１をクローズする。ステップＳ１９０にて書き込み処理は終了する。なお、図示していないが、ファイル分割モジュール１４はステップＳ１９０の後にローカルディスク２にある元ファイルＡを削除する。
【００４３】
続いて、図８を参照して前記ステップＳ１５０の詳細を説明する。まずステップＳ１５１にて分割ファイル書き込み処理を開始する。
【００４４】
ファイル分割モジュール１４は、ステップＳ１５２にて分割ファイルの１つをオープンする。ここでは、分割ファイルＡ２がオープンされたとする。ステップＳ１５３にてファイル分割モジュール１４は、分割ファイルＡ２の先頭から書き込む準備をする。
【００４５】
ステップ１５４では、ファイル分割モジュール１４が、元ファイルＡ内のテキスト情報情報領域４１０の内容を分割ファイルＡ２に書き込む。
【００４６】
ステップＳ１５５では、ファイル分割モジュール１４が書き込みを終えた分割ファイルＡ２をクローズする。そして、ステップＳ１５６にて分割ファイル書き込み処理は終了する。なお、分割ファイルＡ３～Ａ５も同様の手順で作成される。
【００４７】
次に、冗長性除去ディスク３に保存されている分割ファイルＡ１～Ａ５を元にしてファイル分割モジュール１４がローカルディスク２において元ファイルＡを復元する処理手順を図９及び図１０を参照して説明する。
【００４８】
まずステップＳ２１０にて復元を開始する。続いてファイル分割モジュール１４は、ステップＳ２２０にてマスタファイルＡ１をオープンし、ステップＳ２３０にてマスタファイルＡ１に書き込まれている分割ファイルＡ２～Ａ５の保存場所を示すパスを読み込む。
【００４９】
ステップＳ２４０では、ファイル分割モジュール１４は、ローカルディスク２において復元ファイルをオープンする。この復元ファイルが最終的に元ファイルＡとなる。ステップＳ２５０にて、ファイル分割モジュール１４は、復元ファイルの先頭から書き込む準備をする。
【００５０】
ステップＳ２６０では、ファイル分割モジュール１４が、まだ読み込まれていない分割ファイルがあるか否かを判断する。この時点では、分割ファイルＡ２～Ａ５のいずれの内容も読み込まれていないため、この判断結果は「正」となる。
【００５１】
そして、ステップＳ２７０に進み、ファイル分割モジュール１４がまだ読み込まれていないと判断した分割ファイルＡ２～Ａ５のうちの１つを選択する。ここでは、分割ファイルＡ２が選択されたとする。
【００５２】
ステップＳ２８０では、ファイル分割モジュール１４が、ステップＳ２７０にて選択した分割ファイルＡ２を読み込む。このステップＳ２８０の詳細については図１０を参照し

て後述する。

【００５３】

このようなステップＳ２６０～Ｓ２８０は、全ての分割ファイルが読み込まれてその内容が復元ファイルに書き込まれるまで繰り返される。この反復により、図４（ａ）に示したような元ファイルＡ内の領域４１０～４４０がローカルディスク２に復元される。そして、最終的にステップＳ２６０の判断結果が「否」となる。

【００５４】

続いてファイル分割モジュール１４は、ステップＳ２９０にてマスタファイルＡ１をクローズし、ステップＳ２９２にてローカルディスク２の復元ファイルの先頭にメタ情報領域４０５を作成した上で当該復元ファイルをクローズする。そして、ステップＳ２９４にて復元処理を終了する。なお、図示していないが、ファイル分割モジュール１４はステップＳ２９４の後に冗長性除去ディスク３にある分割ファイルＡ１～Ａ５を削除する。

【００５５】

続いて、図１０を参照して前記ステップＳ２８０の詳細を説明する。まずステップＳ２８１にて分割ファイル読み込み処理を開始する。

【００５６】

ファイル分割モジュール１４は、ステップＳ２８２にて分割ファイルの１つをオープンする。ここでは、分割ファイルＡ２がオープンされたとする。ステップＳ２８３にてファイル分割モジュール１４は、分割ファイルＡ２を読み込む。

【００５７】

ステップＳ２８４では、ファイル分割モジュール１４が、分割ファイルＡ２の内容すなわちプレゼンテーション４のテキストの内容を復元ファイルに書き込む。

【００５８】

ステップＳ２８５では、ファイル分割モジュール１４が復元ファイルに内容の書き込みを終えた分割ファイルＡ２をクローズする。そして、ステップＳ２８６にて分割ファイル書き込み処理は終了する。なお、分割ファイルＡ３～Ａ５の内容も同様の手順で復元ファイルに書き込まれる。

【００５９】

これまで、プレゼンテーション４に対応する元ファイルＡを分割ファイルＡ１～Ａ５に分割して冗長性除去ディスク３に書き込む流れと、これら分割ファイルＡ１～Ａ５から元ファイルＡを復元する流れとについて説明してきた。同様の流れで、元ファイルＢは分割ファイルＢ１～Ｂ５に分割されて冗長性除去ディスク３に書き込まれる。ただし、分割ファイルＢ１～Ｂ５の各々の内容が、既に冗長性除去ディスク３に書き込まれている分割ファイルＡ１～Ａ５のいずれかの内容と同じである場合は、その分割ファイルが別のブロックに再び書き込まれることはない。そして、冗長性除去ディスク３に書き込まれている分割ファイルから元ファイルＢが復元される。

【００６０】

図５（ｃ）に示した分割ファイルＡ３の構造を図１１（ａ）に示している。さらに、分割ファイルＢ３の構造を図１１（ｂ）に示している。図示しているように、分割ファイルＡ３は、冗長性除去ディスク３上の第２０ブロックの先端から第２６ブロックの途中にかけて書き込まれている。この第２０ブロックの先端から第２６ブロックの途中にかけての画像情報領域は、プレゼンテーション４内の画像４２の情報を含んだ領域であると同時に、プレゼンテーション５内の画像４２の情報を含んだ領域でもある。すなわち、プレゼンテーション５内の画像４２の情報を含んだ分割ファイルＢ３が、冗長性除去ディスク３上の別のブロックに再び書き込まれることはない。

【００６１】

つまり、分割ファイルＡ３が第２０ブロックの先端から第２６ブロックの途中にかけていったん書き込まれると、その後、これらブロックと同じ内容が別のブロックに書き込まれることはない。これは、元ファイルＡの内容が、冗長性除去ディスク３上の単一のファイルに書き込まれるのではなく、複数のファイルに分割されて書き込まれることによるも

のである。そのため、冗長性除去ディスク３上に単一のファイルとして書き込まれた場合には除去しきれない冗長性を除去することができる。

【００６２】

なお、図５（ｄ）に示した冗長性除去ディスク３の第３０ブロックから第３７ブロックの途中にかけての写真情報領域は、プレゼンテーション４内の写真４３の情報を含んだ領域であると同時に、プレゼンテーション５内の写真４３の情報を含んだ領域でもある。つまり、分割ファイルＢ４が別のブロックに再び書き込まれることはない。

【００６３】

前記実施形態との比較として、元ファイルＡ及び元ファイルＢを分割せずにそれぞれ単一のファイルＡ’及びＢ’として冗長性除去ディスク３に書き込む場合を説明する。ファイルＡ’及びＢ’の構造を図１２（ａ）及び図１２（ｂ）に示している。ファイルＡ’は、メタ情報領域６０５とテキスト情報領域６１０と画像情報領域６２０と写真情報領域６３０とレイアウト情報領域６４０とを有しており、メタ情報領域６０５を先頭にしてこれら領域が順に隣り合って配置されている。また、ファイルＡ’の先端からテキスト情報領域６１０の終端までのオフセットはＤ１である。

【００６４】

ファイルＢ’は、メタ情報領域７０５とテキスト情報領域７１０と画像情報領域７２０と写真情報領域７３０とレイアウト情報領域７４０とを有しており、メタ情報領域７０５を先頭にしてこれら領域が順に隣り合って配置されている。また、ファイルＢ’の先端からテキスト情報領域７１０の終端までのオフセットはＤ２である。オフセットＤ２はオフセットＤ１よりも大きい。なお、ファイルＡ’内のメタ情報領域６０５のサイズは１ブロックであり、ファイルＢ’内のメタ情報領域７０５のサイズも１ブロックである。

【００６５】

図示しているように、画像情報領域６２０は第８７ブロックの途中から第９３ブロックの終端にかけて書き込まれている。これに対し、画像情報領域７２０は第１１８ブロックの先端から第１２４ブロックの途中にかけて書き込まれている。つまり、画像情報領域６２０と画像情報領域７２０とは内容が同じであるにも関わらず、両者は異なるブロックに書き込まれている。これは、オフセットＤ１よりもオフセットＤ２が大きいために、画像情報領域６２０の先端と画像情報領域７２０の先端とがずれてしまい、第８７ブロックから第９３ブロックと第１１８ブロックから第１２４ブロックとが、ブロック単位で見ると、異なる内容になっていることによるものである。これと同じ理由により、写真情報領域６３０と写真情報領域７３０とについても、内容が同じであるにも関わらず、両者は別々のブロックに書き込まれている。

【００６６】

つまり、元ファイルＡ及び元ファイルＢを分割せずにそれぞれ単一のファイルＡ’及びＢ’として冗長性除去ディスク３に書き込むと、ファイルＡ’の一部とファイルＢ’の一部とが同じ内容であるにも関わらず、異なるブロックに書き込まれることになるため、冗長性除去ディスク３であっても冗長性を除去しきれない。

【００６７】

これに対し、元ファイルＡ及び元ファイルＢをそれぞれ分割ファイルに分割した上で、冗長性除去ディスク３に書き込んだ場合には、図１１（ａ）及び図１１（ｂ）に示したように、既に書き込まれている分割ファイルとブロックの内容が同一の分割ファイルが別のブロックに再び書き込まれることはない。したがって、単一のファイルとして冗長性除去ディスク３に書き込んだ場合には除去しきれない冗長性を除去することができる。

【００６８】

［３．冗長性除去ファイルシステムを備えた記憶システム］

図１３は、第１の記憶装置に相当するローカルディスク２と、第２の記憶装置に相当する冗長性除去ファイルシステム６とを備えたコンピュータ１を示している。コンピュータ１には、ＯＳ１１と、アプリケーション１２と、前記冗長性除去ファイルシステム６にファイルを書き込むためのファイル書き込みソフトウェア１５とがインストールされている

。
【００６９】
冗長性除去ファイルシステム６は、外部から読み込んだファイルを複数のサブファイルに分割し、分割されたサブファイルのそれぞれと同じ内容のサブファイルが既に冗長性除去ファイルシステム６に書き込まれていれば、当該サブファイルを再度書き込むことはしないファイルシステムである。
【００７０】
サブファイルへの分割について具体的に説明する。冗長性除去ファイルシステム６は、前記外部から読み込んだファイルの内容つまりビット列からアンカーと呼ばれる所定のビット列を検索する。そして、そのアンカーを区切りとして前記外部から読み込んだファイルをサブファイルに分割する。アンカーは、例えばビット列「０１」をｎ個並べてできるようなビット列であり、冗長性除去ファイルシステムごとに事前に定められている。
【００７１】
具体的には、ファイル書き込みソフトウェア１５は、アンカー挿入モジュール１６を有している。アンカー挿入モジュール１６は、アプリケーション１２によってローカルディスク２に書き込まれた元ファイルＡ内のメタ情報領域４０５を読み込み、元ファイルＡに領域４１０～４４０が含まれていることを検出する。そして、アンカー挿入モジュール１６は、テキスト情報領域４１０の内容を、後述するアンカーが挿入されたファイルＰの先頭に書き込み、続けて冗長性除去ファイルシステム６によって定められているアンカーを書き込む。元ファイルＡ及びアンカーが挿入されたファイルＰは、それぞれ第１のファイル及び第２のファイルに相当する。
【００７２】
このようにして、アンカー挿入モジュール１６は、元ファイルＡにおける部品の情報を有する全ての領域がファイルＰにそれぞれ書き込まれるまで、ファイルＰへの各領域の内容の書き込みとアンカーの書き込みとを繰り返す。つまり、このアンカーは元ファイルＡ内のいずれの領域にも含まれていないビット列である。このようにして、アンカーが挿入されたファイルＰが作成される。
【００７３】
そして、アンカー挿入モジュール１６は、アンカーが挿入されたファイルＰを冗長性除去ファイルシステム６に書き込む。このときファイルＰは、アンカー挿入モジュール１６によって挿入されたアンカーにより、部品の情報を有する領域ごとにサブファイルに分割されて書き込まれる。これらサブファイルの各々は、それ自体別個のファイルである。なお、アンカー挿入モジュール１６は、アンカーが挿入されたファイルＰを冗長性除去ファイルシステム６に書き込んだ後、ローカルディスク２にある元ファイルＡを削除する。
【００７４】
同様に、アンカー挿入モジュール１６は、アプリケーション１２によってローカルディスク２に書き込まれた元ファイルＢを読み込み、この元ファイルＢにアンカーを挿入して、アンカーが挿入されたファイルＱを作成する。続いて、アンカー挿入モジュール１６は、アンカーが挿入されたファイルＱを冗長性除去ファイルシステム６に書き込む。このときファイルＱは、アンカー挿入モジュール１６によって挿入されたアンカーにより、サブファイルに分割されて書き込まれる。これらサブファイルの各々も、それ自体別個のファイルである。ただし、ファイルＱに含まれるサブファイルが、既に冗長性除去ファイルシステム６に書き込まれているファイルＰのサブファイルのいずれかと内容が同じである場合には、当該サブファイルが別の領域に再び書き込まれることはない。この後、アンカー挿入モジュール１６は、ローカルディスク２にある元ファイルＢを削除する。
【００７５】
アンカーが挿入されたファイルＰ及びＱの構造を図１４（ａ）及び図１４（ｂ）にそれぞれ示している。ファイルＰは、サブファイル８１０～８４０という４つのサブファイルを有している。サブファイル８１０には、元ファイルＡ内のテキスト情報領域４１０の内容が書き込まれている。サブファイル８２０には、画像情報領域４２０の内容が書き込ま

れている。サブファイル８３０には、写真情報領域４３０の内容が書き込まれている。サブファイル８４０には、レイアウト情報領域４４０の内容が書き込まれている。そして、サブファイル８１０とサブファイル８２０との間にはアンカー８５０が挿入されている。同様に、サブファイルとサブファイルを区切るために、アンカー８６０及び８７０が挿入されている。前記したように、アンカー８５０～８７０は元ファイルＡに含まれていたものではなく、アンカー挿入モジュール１６が挿入したものである。ファイルＱも、ファイルＰと同様の構造である。ここで、サブファイル８２０とサブファイル９２０とは内容が同じである。また、サブファイル８３０とサブファイル９３０とは内容が同じである。

【００７６】

また、ファイル書き込みソフトウェア１５は、冗長性除去ファイルシステム６上のファイルＰ及びＱを読み込み、その内容をローカルディスク２上の元ファイルＡ及びＢとしてそれぞれ復元する処理をも行う。つまり、ファイル書き込みソフトウェア１５は、コンピュータ１を、ローカルディスク２と冗長性除去ファイルシステム６との両者にアクセスしてファイルを処理するファイル処理装置として具現化する。

【００７７】

以上のような記憶システムの構成及びファイル構造のもとに、アプリケーション１２がローカルディスク２に書き込んだ元ファイルＡにアンカーを挿入し、アンカーが挿入されたファイルＰを冗長性除去ファイルシステム６に書き込む処理手順を図１５に示している。

【００７８】

まず、ステップＳ４１０にて書き込みを開始し、ステップＳ４２０にてアンカー挿入モジュール１６が元ファイルＡと、冗長性除去ファイルシステム上の書き込み用ファイルとをオープンする。この書き込み用ファイルは、最終的にはファイルＰとなるファイルである。さらに、このステップＳ４２０にて、アンカー挿入モジュール１６は、メタ情報領域４０５を読み込み、元ファイルＡに領域４１０～４４０が含まれていることを検出する。ステップＳ４３０にて、アンカー挿入モジュール１６が書き込み用ファイルの先頭に書き込む準備をする。

【００７９】

ステップＳ４４０にてアンカー挿入モジュール１６が、まだ書き込み用ファイルに書き込まれていない１つ以上の部品あるいはレイアウト情報があるか否かを判断する。この時点では、元ファイルＡ内のテキスト領域５１と画像領域５２と写真領域５３との各部品領域とレイアウト情報領域５４とのいずれの領域にある情報も分割ファイルに書き込まれていないため、この判断結果は「正」となる。

【００８０】

そして、ステップＳ４５０に進み、アンカー挿入モジュール１６が書き込み用ファイルに書き込まれていないと判断した領域５１～５４のうちの１つを選択する。ここでは、テキスト領域５１が選択されたとする。

【００８１】

ステップＳ４６０では、アンカー挿入モジュール１６が、ステップＳ４５０にて選択したテキスト領域５１の内容を書き込み用ファイルに書き込む。

【００８２】

ステップＳ４７０では、アンカー挿入モジュール１６がステップＳ４６０の結果を受けて現在の書き込み位置を更新する。

【００８３】

ステップＳ４８０では、アンカー挿入モジュール１６が、アンカーを書き込む。ステップＳ４９０では、アンカー挿入モジュール１６がステップＳ４８０の結果を受けて現在の書き込み位置を更新し、ステップＳ４４０に戻る。

【００８４】

このようなステップＳ４４０～Ｓ４９０は、部品あるいはレイアウト情報の全てが書き込み用ファイルに書き込まれるまで繰り返される。この反復により、図１４（ａ）に示し

たようなファイルＰが冗長性除去ファイルシステム６に書き込まれる。そして、最終的にステップＳ４４０の判断結果が「否」となる。

【００８５】

続いてステップＳ４９２に進み、アンカー挿入モジュール１６は、元ファイルＡと、書き込み用ファイルつまりファイルＰをクローズする。そして、ステップＳ４９４にて書き込み処理を終了する。なお、図示していないが、アンカー挿入モジュール１６はステップＳ４９４の後にローカルディスク２にある元ファイルＡを削除する。

【００８６】

次に、冗長性除去ファイルシステム６に書き込まれているファイルＰを元にしてアンカー挿入モジュール１６がローカルディスク２において元ファイルＡを復元する処理手順を図１６を参照して説明する。

【００８７】

まずステップＳ５１０にて復元を開始する。続いてアンカー挿入モジュール１６は、ステップＳ５２０にて、アンカーが挿入されたファイルＰをオープンする。このとき、ファイルＰの読み込み位置はファイルＰの先頭である。

【００８８】

ステップＳ５３０にて、アンカー挿入モジュール１６は、ローカルディスク２において復元ファイルをオープンする。この復元ファイルが最終的には元ファイルＡとなる。ステップＳ５５０にて、アンカー挿入モジュール１６は、アンカーが挿入されたファイルＰの読み込み位置がファイルＰの終端に到達した否かを判断する。ここでの判断結果は、「否」となるため、ステップＳ５５０に進む。

【００８９】

ステップＳ５６０では、アンカー挿入モジュール１６が、ファイルＰの先頭からアンカー８５０までの領域を読み込む。つまり、サブファイル８１０の内容が読み込まれることになる。

【００９０】

ステップＳ５７０では、アンカー挿入モジュール１６が、ステップＳ５６０で読み込んだ内容を前記復元ファイルの現在の書き込み位置に書き込む。そして、ステップＳ５８０にて、アンカー挿入モジュール１６が復元ファイルにおける現在の書き込み位置を更新する。

【００９１】

ステップＳ５９０では、アンカー挿入モジュール１６が、ファイルＰ内のアンカー８５０を読みとばす。そして、ステップＳ５５０に戻る。

【００９２】

このようなステップＳ５５０～Ｓ５９０は、ファイルＰ内のサブファイル８１０～８４０の内容が復元ファイルに書き込まれるまで繰り返される。この反復により、図４（ａ）に示したような元ファイルＡ内の領域４１０～４４０がローカルディスク２に復元される。そして、最終的にステップＳ５５０の判断結果が「正」となる。

【００９３】

続いてアンカー挿入モジュール１６は、ステップＳ５９２にてローカルディスク２上の復元ファイルの先頭にメタ情報領域４０５を作成した上で当該復元ファイルをクローズし、ステップＳ５９４にて冗長性除去ファイルシステム６上のファイルＰをクローズする。そして、ステップＳ５９６にて復元処理を終了する。なお、図示していないが、アンカー挿入モジュール１６はステップＳ５９６の後に冗長性除去ファイルシステム６にあるファイルＰを削除する。

【００９４】

これまで、ローカルディスク２上の元ファイルＡが冗長性除去ファイルシステム６上のアンカーが挿入されたファイルＰとして書き込まれる流れと、このファイルＰから元ファイルＡが復元される流れとについて説明してきた。これと同様に、ローカルディスク２上の元ファイルＢは、アンカーが挿入されたファイルＱとして冗長性除去ファイルシステム

６に書き込まれる。ただし、既に書き込まれているファイルＰ内のサブファイルと同じ内容のサブファイルがファイルＱ内にある場合には、ファイルＱ内の当該サブファイルが冗長性除去ファイルシステム６の別の領域に再び書き込まれることはない。そして、このようにして冗長性除去ファイルシステム６に書き込まれたファイルから元ファイルＢが復元される。

【００９５】

つまり、前記形態によれば、既にサブファイル８２０が冗長性除去ファイルシステム６に書き込まれているため、内容が同じサブファイル９２０が再び別の領域に書き込まれることはない。また、既にサブファイル８３０が冗長性除去ファイルシステム６に書き込まれているため、内容が同じサブファイル９３０が別の領域に再び書き込まれることはない。

【００９６】

これに対し、前記実施形態との比較として、元ファイルＡ及び元ファイルＢに対してアンカーを挿入せずにそのまま冗長性除去ファイルシステム６に書き込んだ場合について説明する。まず、プレゼンテーション４に対応する元ファイルＡがそのままファイルＸとして冗長性除去ファイルシステム６に書き込まれているとする。続いて、プレゼンテーション５に対応する元ファイルＢがそのままファイルＹとして冗長性除去ファイルシステム６に書き込まれたとする。

【００９７】

このとき、ファイルＸとファイルＹとで画像４２に関する情報は同一である。しかし、画像４２に関する情報の中にアンカーが含まれない場合には、画像４２に関する情報のみが書き込まれたサブファイルは作成されない。そのため、冗長性除去ファイルシステム６において、ファイルＸ内の画像４２に関する情報が書き込まれた部分と、ファイルＹ内の画像４２に関する情報が書き込まれた部分と重複して存在することとなる。結果として、除去しきれない冗長性が生まれてしまう。

【００９８】

アンカーによるサブファイルへの分割は、バックアップストリームなどの容量が巨大なデータの分割に適している。しかし、上述したように、アプリケーション１２が作成した比較的小さな容量のデータには、所定のアンカーが含まれていないことがあり、その場合には前記データを部品ごとにサブファイルに分割することは不可能である。すると、同じ部品の情報であっても、冗長性除去ファイルシステム６上の別々のファイルにそれぞれ書き込まれるため、冗長性を除去しきれない。

【００９９】

これに対し、先に述べたような、部品と部品との間にアンカーを新たに挿入したファイルを作る形態によれば、部品ごとにサブファイルが作られる。したがって、冗長性除去ファイルシステム６に既に書き込まれているサブファイルに格納されている部品情報と同一の部品情報を含んだサブファイルが再び別の領域に書き込まれることはない。よって、アプリケーション１２が作成したファイルにアンカーを挿入しない場合には除去しきれない冗長性を、アンカーを新たに挿入する実施形態では除去することができる。

【０１００】

［４．他の形態］

図５及び図６を参照して述べた実施形態では、分割ファイルＡ２～Ａ５がいずれも「／ｘｘ／ｙｙ／」という同じ場所に保存されている。しかし、全ての分割ファイルが同じ場所に保存されている必要はない。例えば、分割ファイルＡ３が「／ｘｘ／ｚｚ／」という場所に保存されていて、その他の分割ファイルＡ１、Ａ２、Ａ４、Ａ５が「／ｙｙ／ｘｘ／」という場所に保存されていてもよい。

【０１０１】

補助記憶装置１１６は、物理的にコンピュータ１の外部に設けることもできる。また、補助記憶装置１１６に相当するローカルディスク２及び冗長性除去ディスク３の両者が物理的に一体化されていてもよい。同じく補助記憶装置１１６に相当するローカルディスク

２及び冗長性除去ファイルシステム６の両者が物理的に一体化されていてもよい。

【０１０２】

また、前記実施形態では、ローカルディスク２が不揮発性の補助記憶装置１１６に相当するとしたが、揮発性のメモリ装置とすることもできる。

【０１０３】

前記実施形態では、メタ情報領域４０５は、元ファイルＡ内の領域４１０～４４０のそれぞれの先端及び終端と元ファイルＡの先端とのオフセットに関する情報が書き込まれた領域である。しかし、これに限られず、メタ情報領域４０５には、元ファイルＡ内の領域４１０～４４０のそれぞれの先端と元ファイルＡの先端とのオフセットに関する情報と各領域のサイズに関する情報とが書き込まれていてもよい。元ファイルＢ内のメタ情報領域５０５についても同様である。

【０１０４】

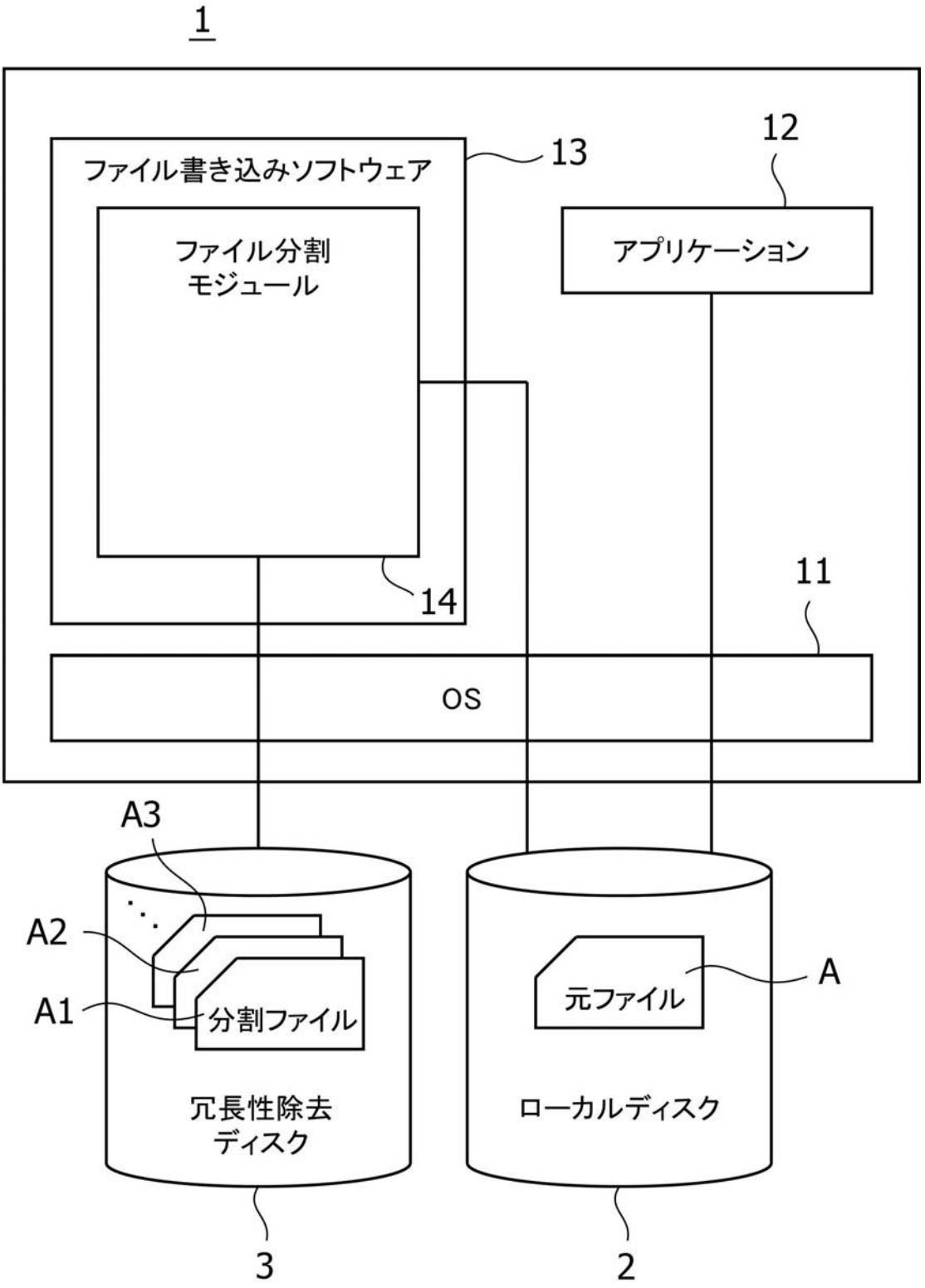
前述した情報処理装置の機能的構成及び物理的構成は、前述の態様に限られるものではなく、例えば、各機能や物理資源を統合して実装したり、逆に、さらに分散して実装したりすることも可能である。

【符号の説明】

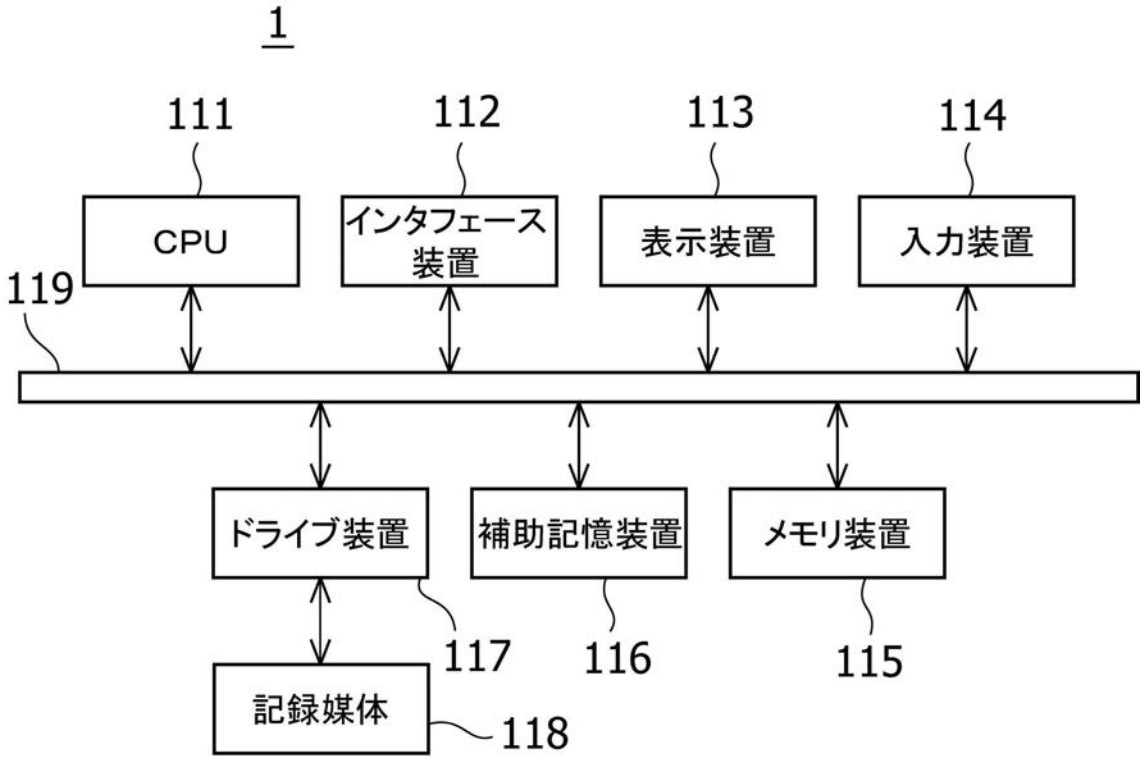
【０１０５】

１　コンピュータ
２　ローカルディスク
３　冗長性除去ディスク
４　プレゼンテーション
５　プレゼンテーション
６　冗長性除去ファイルシステム
１１　ＯＳ
１２　アプリケーション
１３　ファイル書き込みソフトウェア
１４　ファイル分割モジュール
１５　ファイル書き込みソフトウェア
１６　アンカー挿入モジュール
１１１　ＣＰＵ
１１２　インタフェース装置
１１３　表示装置
１１４　入力装置
１１５　メモリ装置
１１６　補助記憶装置
１１７　ドライブ装置
１１８　記録媒体
１１９　バス

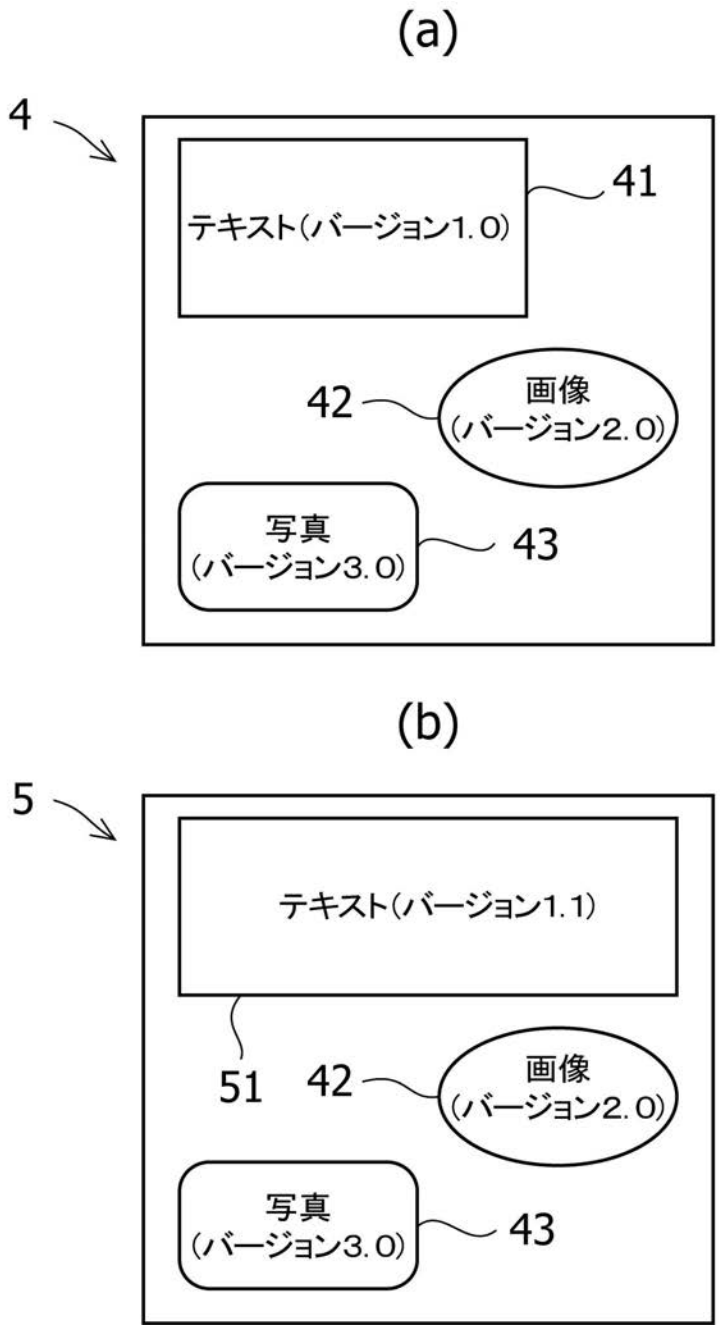
【図１】


1
ファイル書き込みソフトウェア
13
12
ファイル分割
モジュール
アプリケーション
14
11
OS
A3
A2
A1
分割ファイル
元ファイル
A
冗長性除去
ディスク
ローカルディスク
3
2

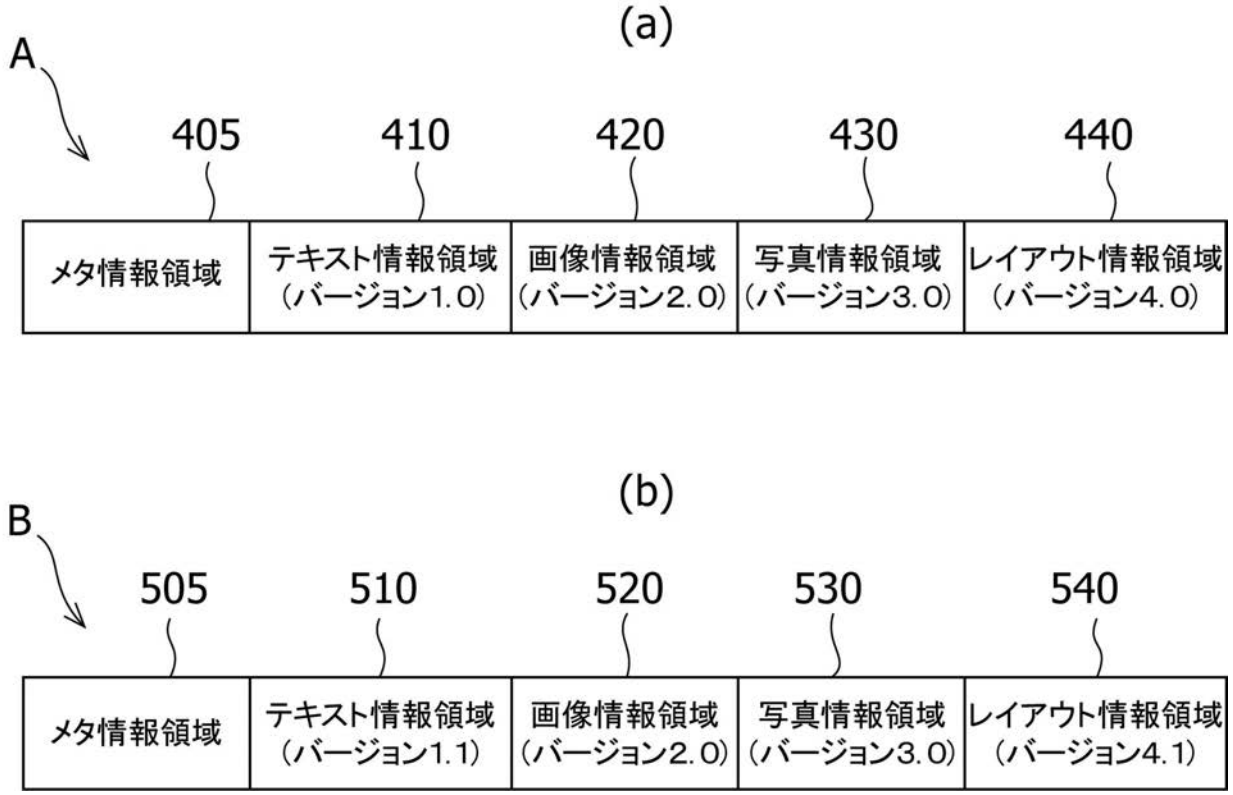
【図２】


1
111
112
113
114
CPU
インタフェース装置
表示装置
入力装置
119
ドライブ装置
補助記憶装置
メモリ装置
117
116
115
記録媒体
118

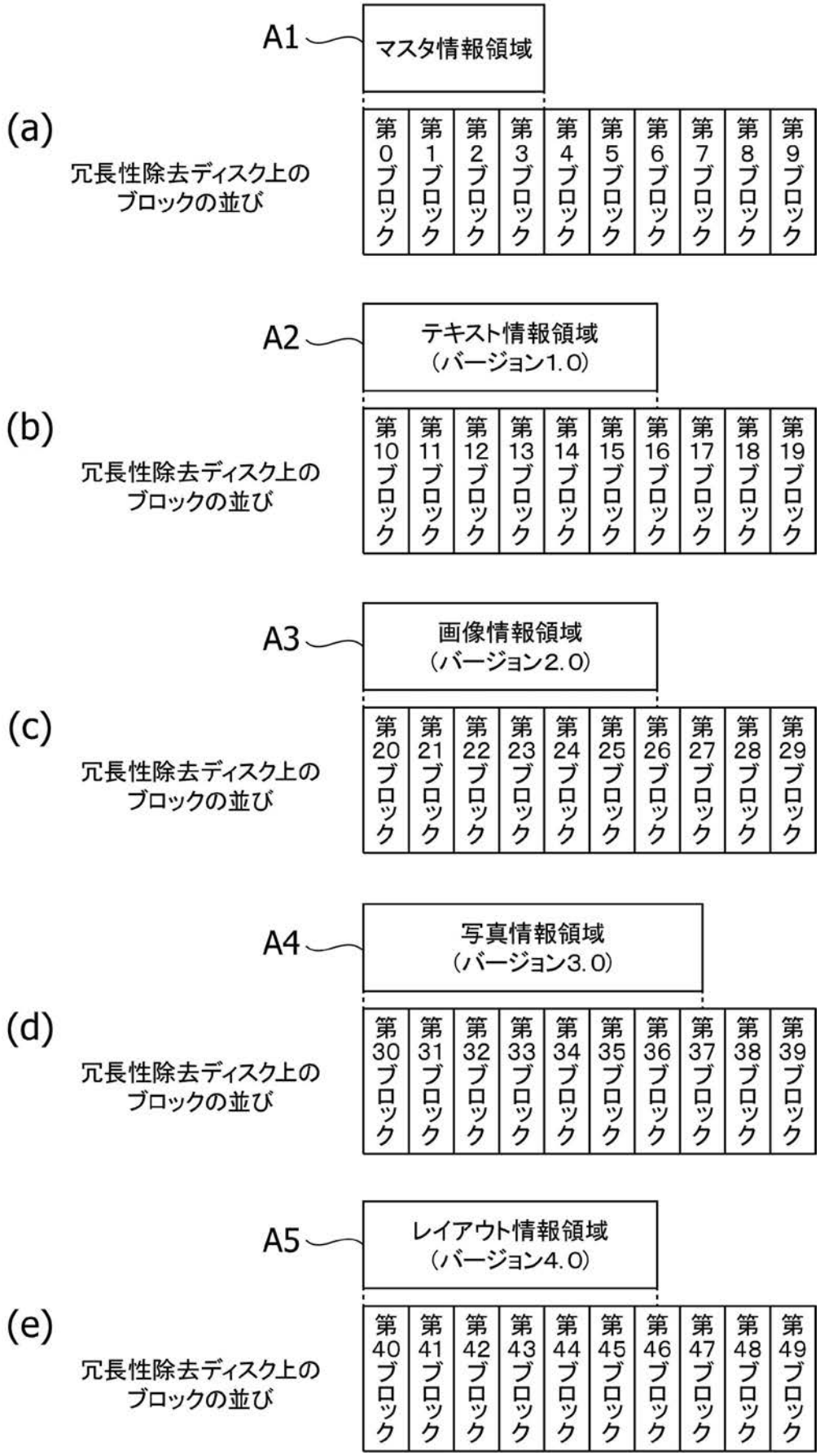
【図３】


(a)
4
テキスト(バージョン1.0)
41
42
画像
(バージョン2.0)
写真
(バージョン3.0)
43
(b)
5
テキスト(バージョン1.1)
51
42
画像
(バージョン2.0)
写真
(バージョン3.0)
43

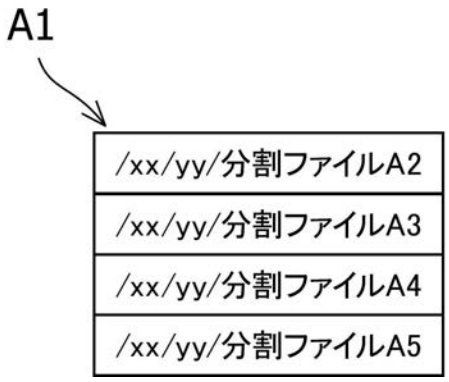
【図４】


(a)
A
405
410
420
430
440
メタ情報領域
テキスト情報領域
（バージョン1.0）
画像情報領域
（バージョン2.0）
写真情報領域
（バージョン3.0）
レイアウト情報領域
（バージョン4.0）
(b)
B
505
510
520
530
540
メタ情報領域
テキスト情報領域
（バージョン1.1）
画像情報領域
（バージョン2.0）
写真情報領域
（バージョン3.0）
レイアウト情報領域
（バージョン4.1）

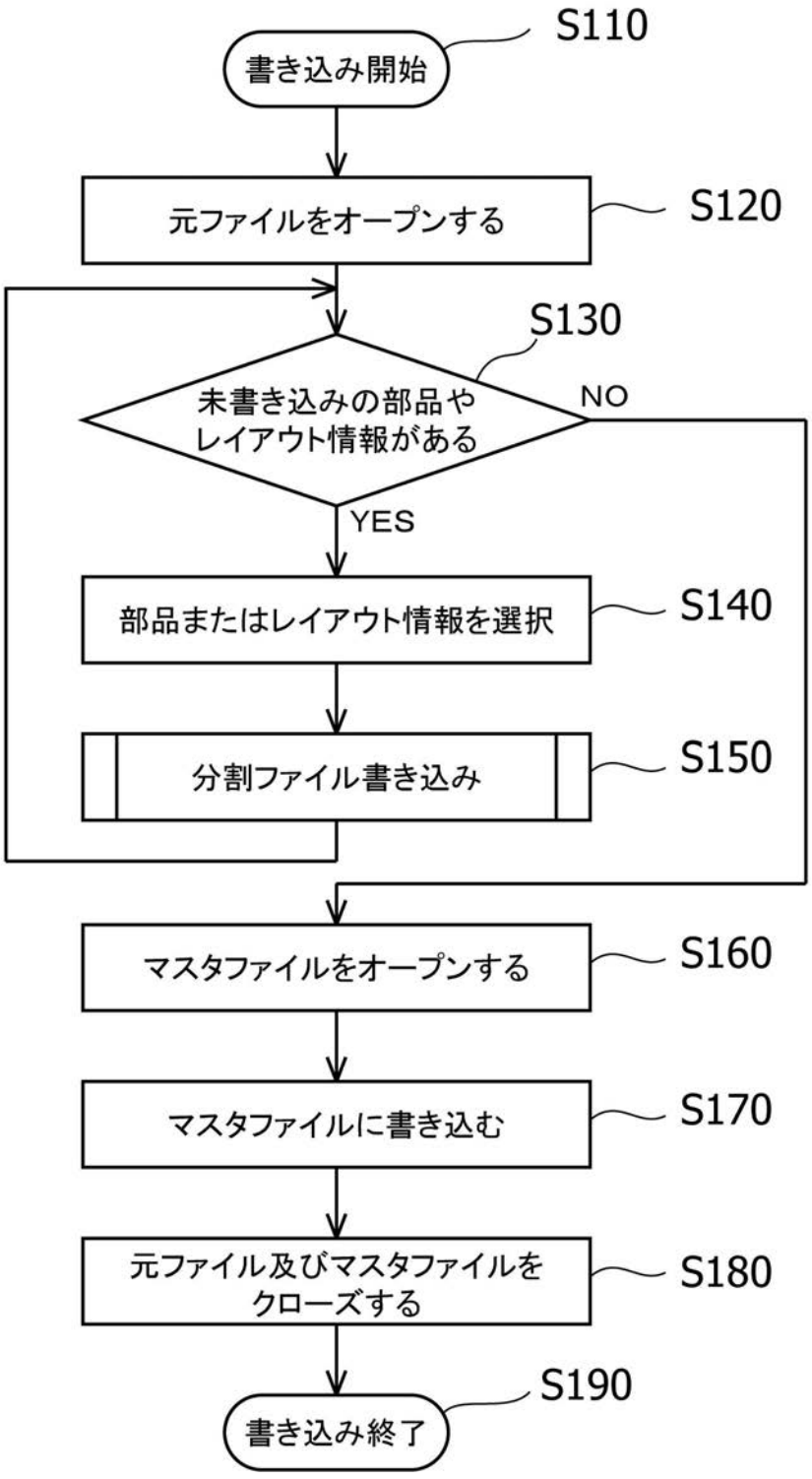
【図５】


A1
マスタ情報領域
(a)
冗長性除去ディスク上のブロックの並び
第0ブロック
第1ブロック
第2ブロック
第3ブロック
第4ブロック
第5ブロック
第6ブロック
第7ブロック
第8ブロック
第9ブロック
A2
テキスト情報領域
（バージョン1.0）
(b)
冗長性除去ディスク上のブロックの並び
第10ブロック
第11ブロック
第12ブロック
第13ブロック
第14ブロック
第15ブロック
第16ブロック
第17ブロック
第18ブロック
第19ブロック
A3
画像情報領域
（バージョン2.0）
(c)
冗長性除去ディスク上のブロックの並び
第20ブロック
第21ブロック
第22ブロック
第23ブロック
第24ブロック
第25ブロック
第26ブロック
第27ブロック
第28ブロック
第29ブロック
A4
写真情報領域
（バージョン3.0）
(d)
冗長性除去ディスク上のブロックの並び
第30ブロック
第31ブロック
第32ブロック
第33ブロック
第34ブロック
第35ブロック
第36ブロック
第37ブロック
第38ブロック
第39ブロック
A5
レイアウト情報領域
（バージョン4.0）
(e)
冗長性除去ディスク上のブロックの並び
第40ブロック
第41ブロック
第42ブロック
第43ブロック
第44ブロック
第45ブロック
第46ブロック
第47ブロック
第48ブロック
第49ブロック

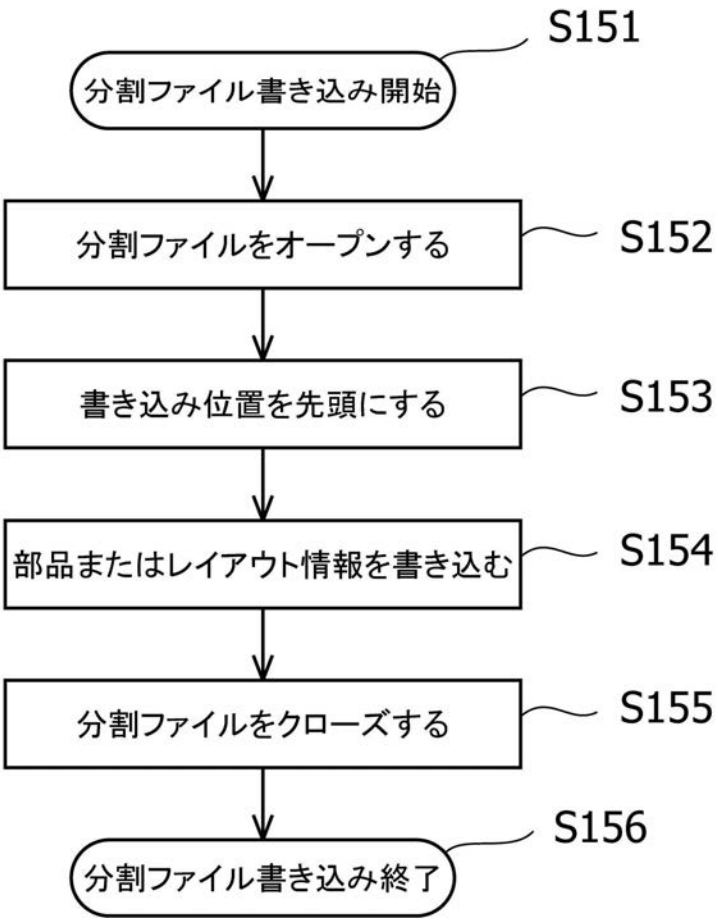
【図６】


A1
/xx/yy/分割ファイルA2
/xx/yy/分割ファイルA3
/xx/yy/分割ファイルA4
/xx/yy/分割ファイルA5

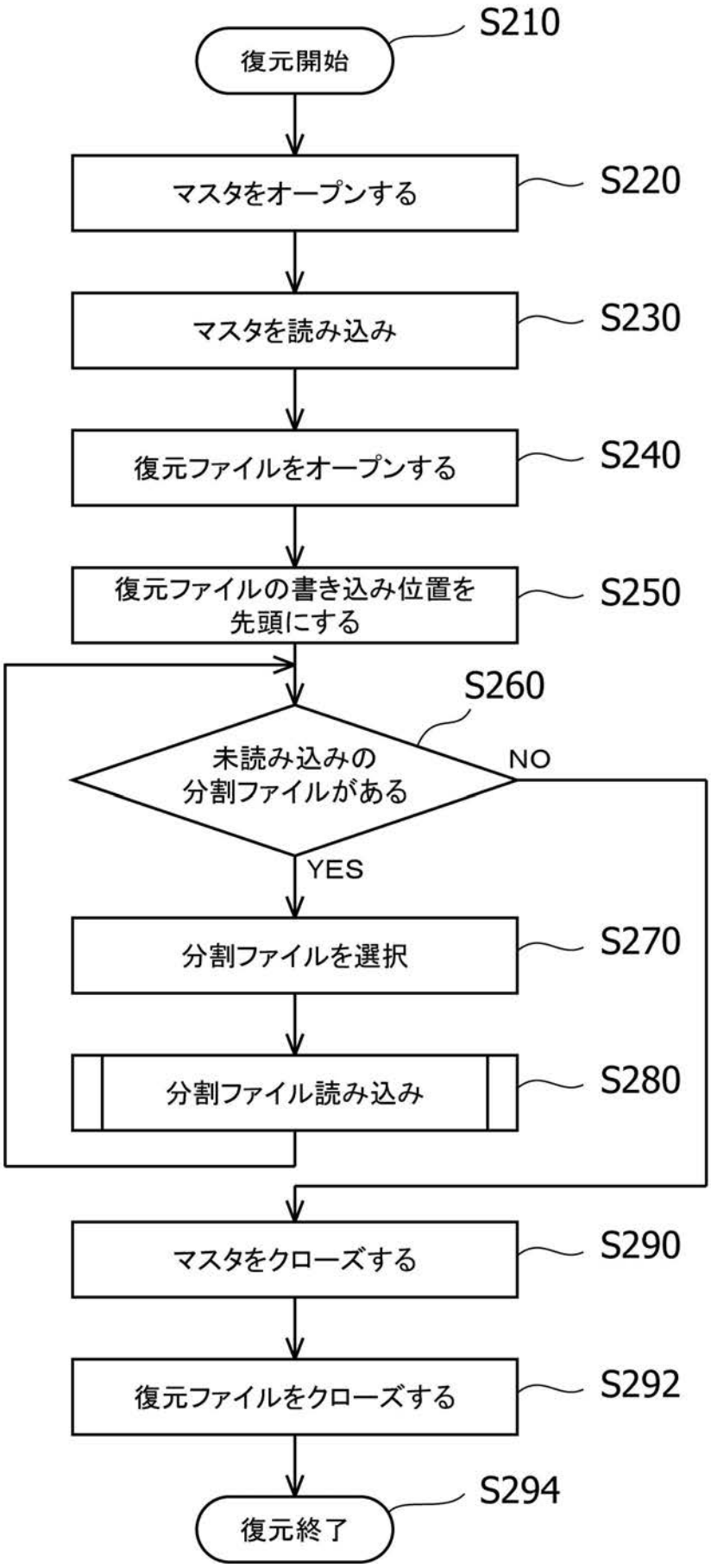
【図７】


S110
書き込み開始
元ファイルをオープンする
S120
S130
未書き込みの部品や
レイアウト情報がある
NO
YES
部品またはレイアウト情報を選択
S140
分割ファイル書き込み
S150
マスタファイルをオープンする
S160
マスタファイルに書き込む
S170
元ファイル及びマスタファイルを
クローズする
S180
S190
書き込み終了

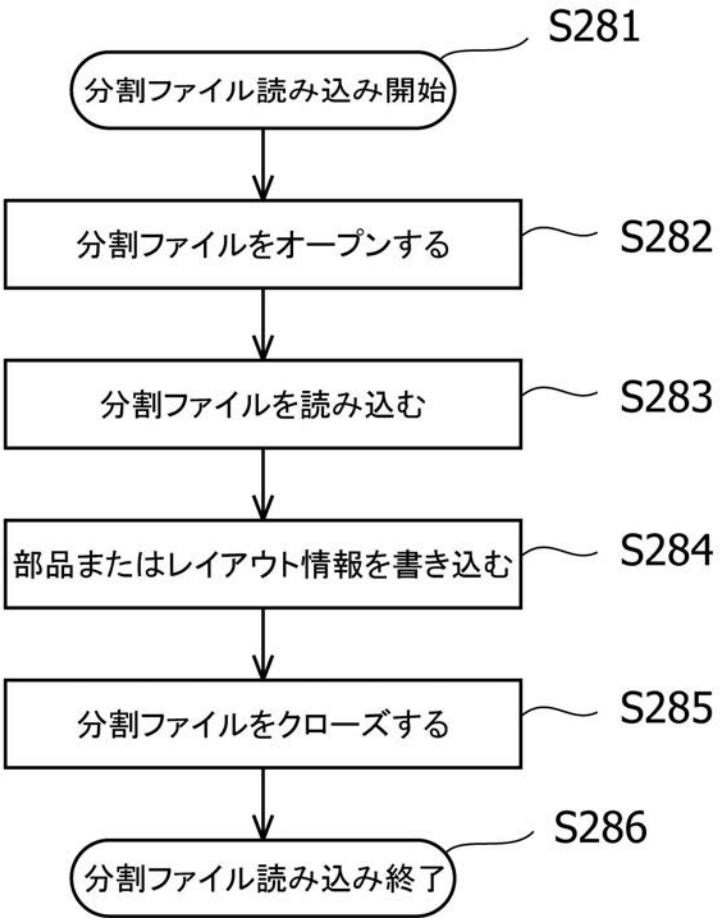
【図８】


S151
分割ファイル書き込み開始
分割ファイルをオープンする
S152
書き込み位置を先頭にする
S153
部品またはレイアウト情報を書き込む
S154
分割ファイルをクローズする
S155
S156
分割ファイル書き込み終了

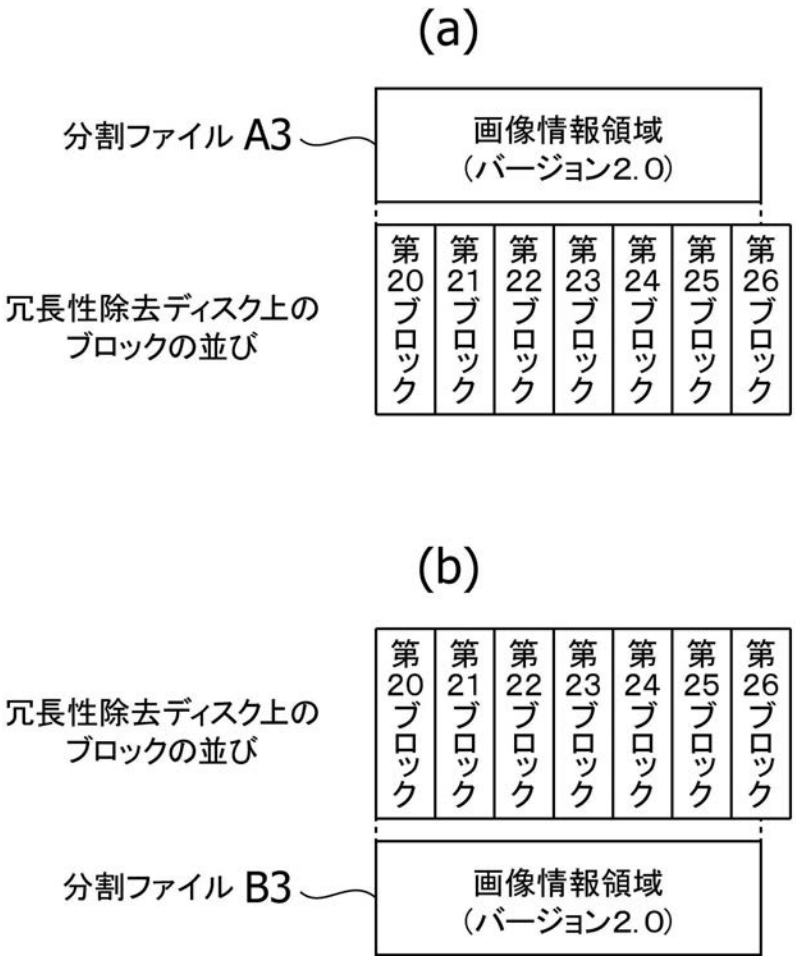
【図９】


S210
復元開始
マスタをオープンする
S220
マスタを読み込み
S230
復元ファイルをオープンする
S240
復元ファイルの書き込み位置を
先頭にする
S250
S260
未読み込みの
分割ファイルがある
NO
YES
分割ファイルを選択
S270
分割ファイル読み込み
S280
マスタをクローズする
S290
復元ファイルをクローズする
S292
S294
復元終了

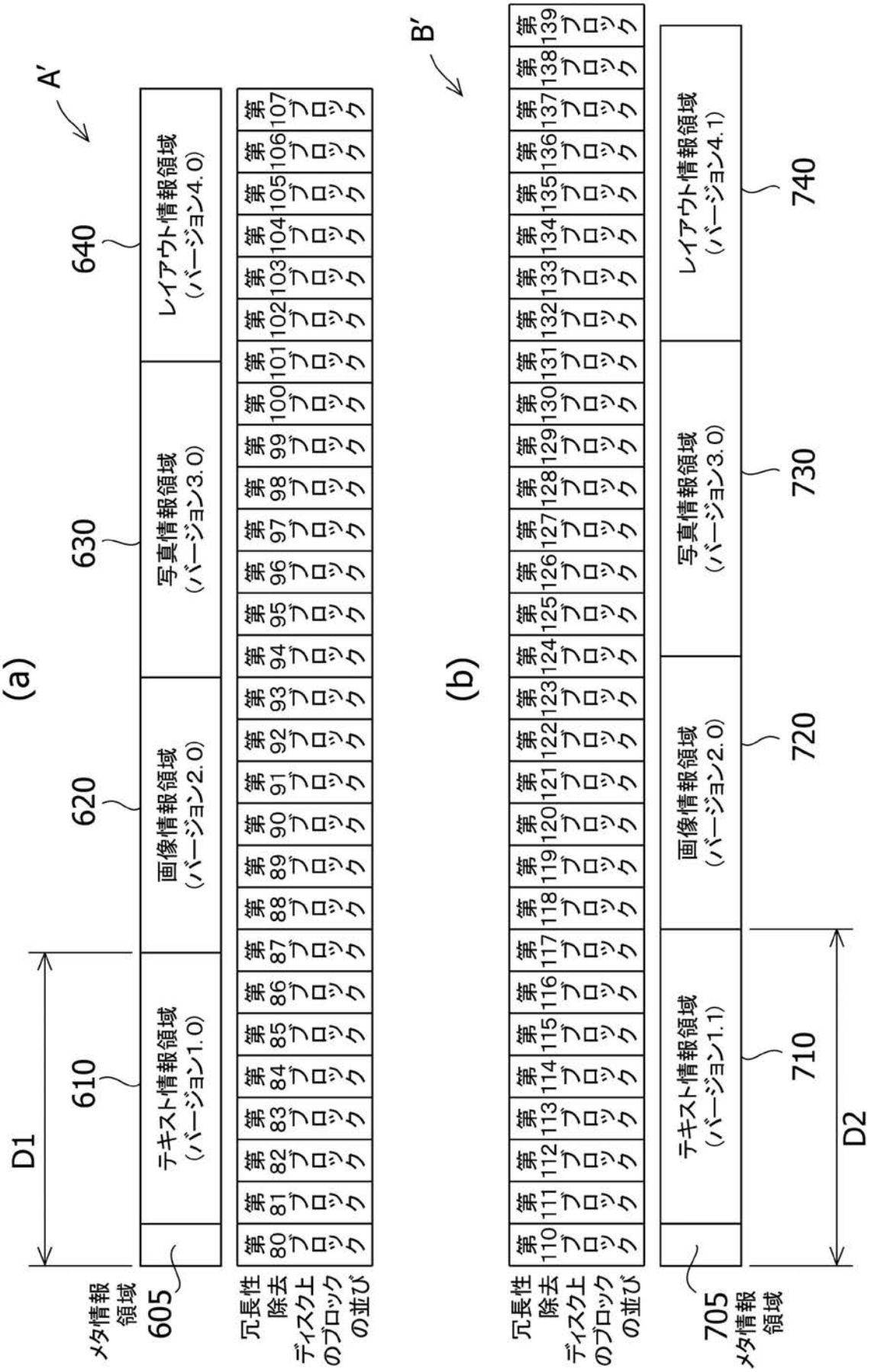
【図１０】


S281
分割ファイル読み込み開始
分割ファイルをオープンする
S282
分割ファイルを読み込む
S283
部品またはレイアウト情報を書き込む
S284
分割ファイルをクローズする
S285
分割ファイル読み込み終了
S286

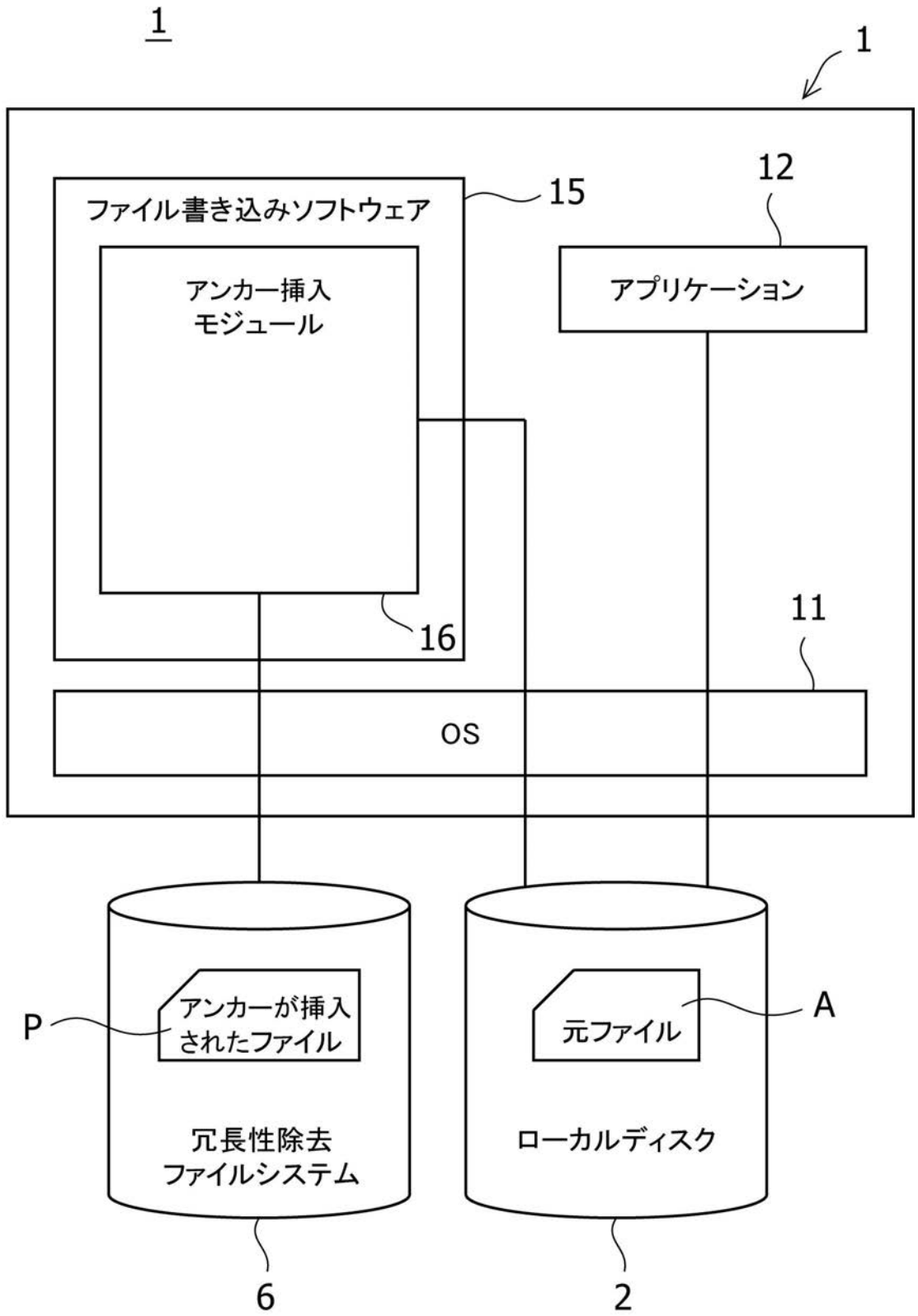
【図１１】


(a)
分割ファイル A3
画像情報領域
（バージョン2.0）
冗長性除去ディスク上の
ブロックの並び
第20ブロック
第21ブロック
第22ブロック
第23ブロック
第24ブロック
第25ブロック
第26ブロック
(b)
冗長性除去ディスク上の
ブロックの並び
第20ブロック
第21ブロック
第22ブロック
第23ブロック
第24ブロック
第25ブロック
第26ブロック
分割ファイル B3
画像情報領域
（バージョン2.0）

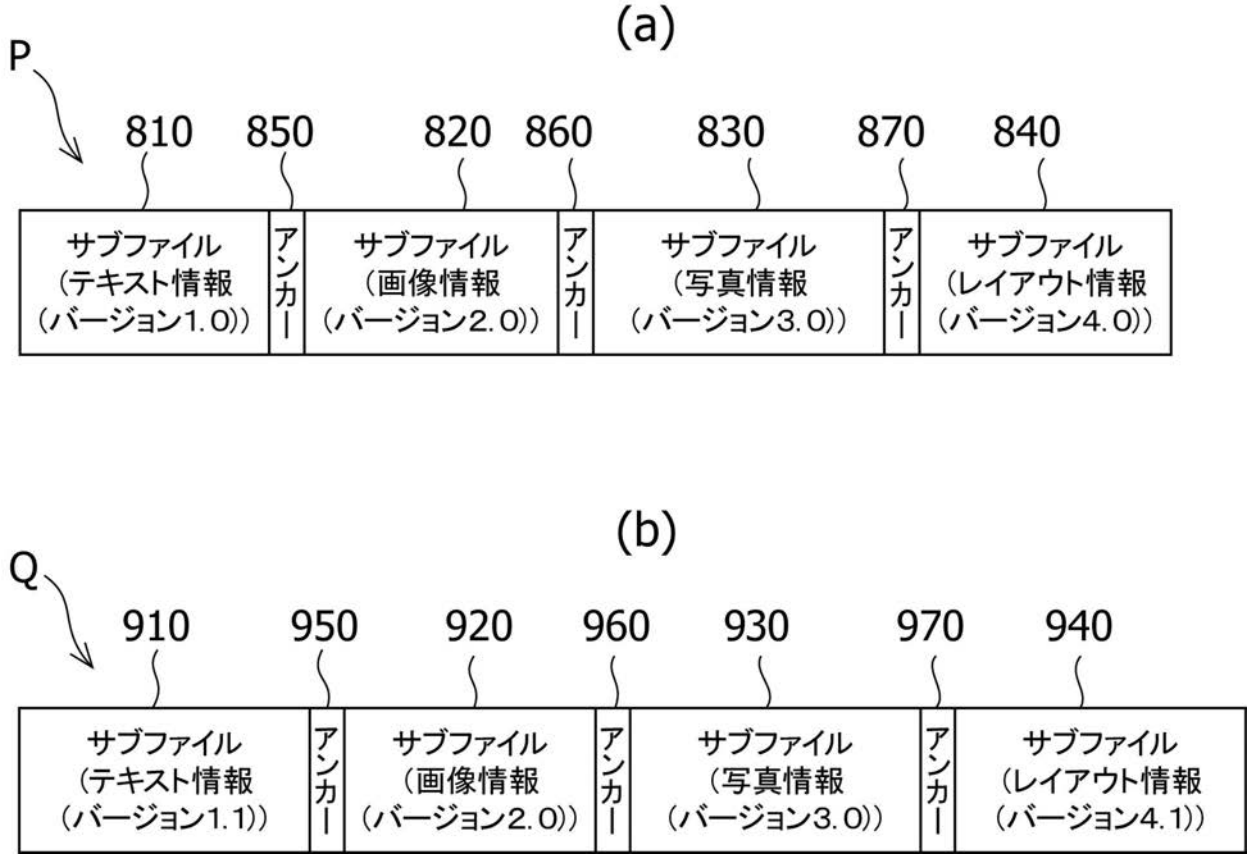
【図１２】


(a)
A′
D1
610
620
630
640
メタ情報領域
605
テキスト情報領域（バージョン1.0）
画像情報領域（バージョン2.0）
写真情報領域（バージョン3.0）
レイアウト情報領域（バージョン4.0）
冗長性除去ディスク上のブロックの並び
第80ブロック
第81ブロック
第82ブロック
第83ブロック
第84ブロック
第85ブロック
第86ブロック
第87ブロック
第88ブロック
第89ブロック
第90ブロック
第91ブロック
第92ブロック
第93ブロック
第94ブロック
第95ブロック
第96ブロック
第97ブロック
第98ブロック
第99ブロック
第100ブロック
第101ブロック
第102ブロック
第103ブロック
第104ブロック
第105ブロック
第106ブロック
第107ブロック
(b)
B′
冗長性除去ディスク上のブロックの並び
第110ブロック
第111ブロック
第112ブロック
第113ブロック
第114ブロック
第115ブロック
第116ブロック
第117ブロック
第118ブロック
第119ブロック
第120ブロック
第121ブロック
第122ブロック
第123ブロック
第124ブロック
第125ブロック
第126ブロック
第127ブロック
第128ブロック
第129ブロック
第130ブロック
第131ブロック
第132ブロック
第133ブロック
第134ブロック
第135ブロック
第136ブロック
第137ブロック
第138ブロック
第139ブロック
705
メタ情報領域
テキスト情報領域（バージョン1.1）
画像情報領域（バージョン2.0）
写真情報領域（バージョン3.0）
レイアウト情報領域（バージョン4.1）
710
720
730
740
D2

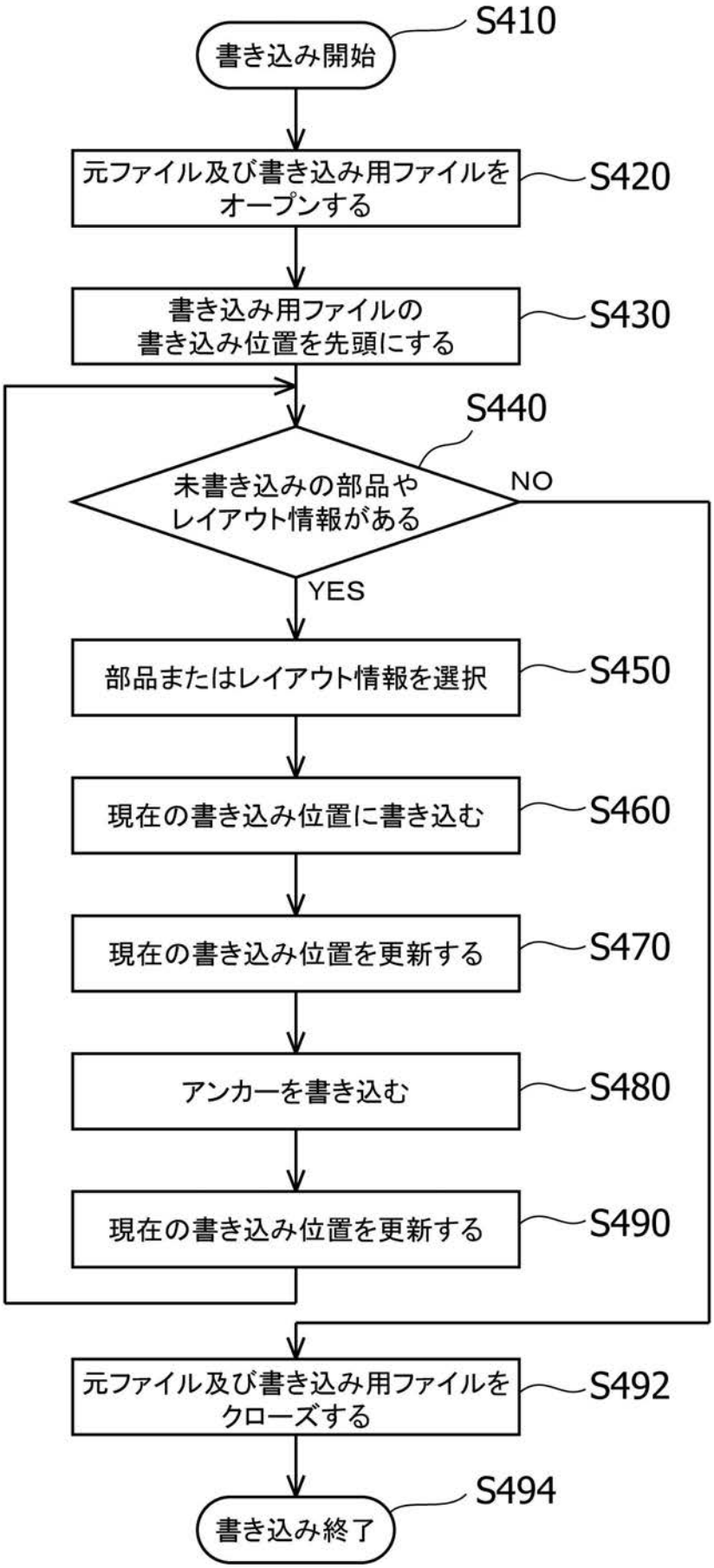
【図１３】


1
1
ファイル書き込みソフトウェア
15
12
アンカー挿入
モジュール
アプリケーション
16
11
OS
P
アンカーが挿入
されたファイル
元ファイル
A
冗長性除去
ファイルシステム
ローカルディスク
6
2

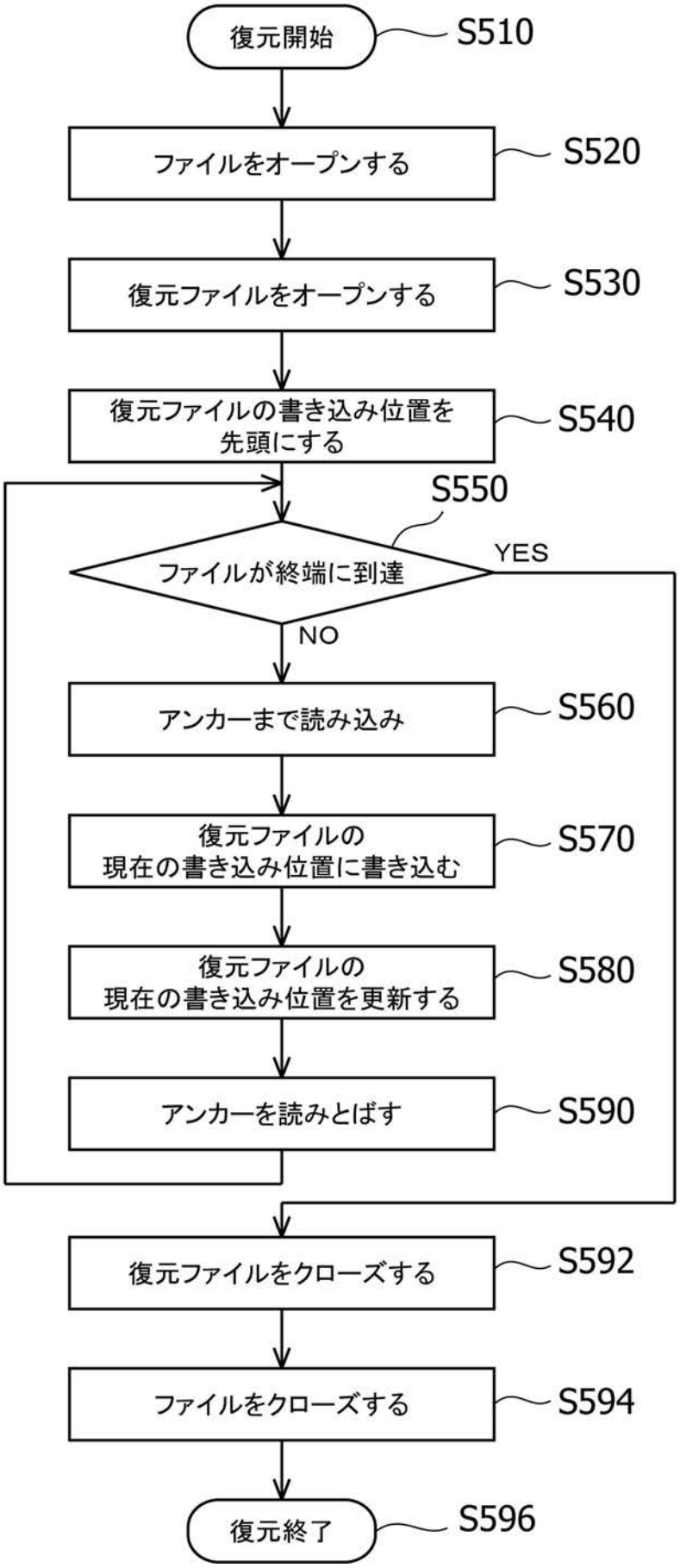
【図１４】


(a)
P
810
850
820
860
830
870
840
サブファイル（テキスト情報（バージョン1.0））
アンカー
サブファイル（画像情報（バージョン2.0））
アンカー
サブファイル（写真情報（バージョン3.0））
アンカー
サブファイル（レイアウト情報（バージョン4.0））
(b)
Q
910
950
920
960
930
970
940
サブファイル（テキスト情報（バージョン1.1））
アンカー
サブファイル（画像情報（バージョン2.0））
アンカー
サブファイル（写真情報（バージョン3.0））
アンカー
サブファイル（レイアウト情報（バージョン4.1））

【図１５】


S410
書き込み開始
元ファイル及び書き込み用ファイルを
オープンする
S420
書き込み用ファイルの
書き込み位置を先頭にする
S430
S440
未書き込みの部品や
レイアウト情報がある
NO
YES
部品またはレイアウト情報を選択
S450
現在の書き込み位置に書き込む
S460
現在の書き込み位置を更新する
S470
アンカーを書き込む
S480
現在の書き込み位置を更新する
S490
元ファイル及び書き込み用ファイルを
クローズする
S492
書き込み終了
S494

【図１６】


復元開始
S510
ファイルをオープンする
S520
復元ファイルをオープンする
S530
復元ファイルの書き込み位置を
先頭にする
S540
S550
ファイルが終端に到達
YES
NO
アンカーまで読み込み
S560
復元ファイルの
現在の書き込み位置に書き込む
S570
復元ファイルの
現在の書き込み位置を更新する
S580
アンカーを読みとばす
S590
復元ファイルをクローズする
S592
ファイルをクローズする
S594
復元終了
S596

フロントページの続き

(72)発明者　土屋　芳浩
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内
(72)発明者　荻原　一隆
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内
(72)発明者　田村　雅寿
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内
(72)発明者　渡辺　高志
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内
(72)発明者　熊野　達夫
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内