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

特許第5609333号
(P5609333)

(45)発行日　平成26年10月22日(2014.10.22)　(24)登録日　平成26年9月12日(2014.9.12)

(51)Int.Cl.			F I		
G06F	9/445	(2006.01)	G06F	9/06	610K
G06F	9/54	(2006.01)	G06F	9/06	640B
G06F	9/50	(2006.01)	G06F	9/06	640H

請求項の数 8　(全 33 頁)

(21)出願番号　特願2010-153190(P2010-153190)
(22)出願日　平成22年7月5日(2010.7.5)
(65)公開番号　特開2012-14637(P2012-14637A)
(43)公開日　平成24年1月19日(2012.1.19)
審査請求日　平成25年5月7日(2013.5.7)

(73)特許権者　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　100092978
弁理士　真田　有
(74)代理人　100112678
弁理士　山本　雅久
(72)発明者　田中　法美
石川県かほく市宇野気ヌ９８番地の２　株式会社ＰＦＵ内

審査官　大塚　俊範

最終頁に続く

(54)【発明の名称】起動処理方法、情報処理装置、起動処理プログラム及び同プログラムを記録したコンピュータ読取可能な記録媒体

(57)【特許請求の範囲】
【請求項１】
プロセッサをそなえた情報処理装置の起動処理方法であって、
該プロセッサにより実行されることにより当該情報処理装置の起動プロセスの一部を実現する複数のモジュールのうち、２以上のモジュールにより共通に用いられる第１共通情報をそなえる第１モジュールをロードするステップと、
該情報処理装置にそなえられた記憶領域に、該第１共通情報へアクセスするためのアクセス情報を格納するステップと、
該アクセス情報にアクセスするためのインタフェース情報をそなえる第２モジュールをロードするステップと、
該第２モジュールにより、該インタフェース情報によって該アクセス情報にアクセスし、該アクセス情報を介して該第１共通情報を取得するステップとをそなえることを特徴とする、起動処理方法。
【請求項２】
該記憶領域に、２以上のモジュールにより共通に用いられる第２共通情報を格納する共通領域を形成するステップと、
該記憶領域に、該共通領域へアクセスするための共通領域アクセス情報を格納するステップと、
該第２モジュールにより、該インタフェース情報によって該アクセス情報にアクセスし、アクセスした当該アクセス情報を用いて該第１共通情報にアクセスし、アクセスした当

該第１共通情報に基づいて該共通領域アクセス情報にアクセスし、アクセスした当該共通領域アクセス情報を用いて、該共通領域に格納された該第２共通情報を取得するステップとをそなえることを特徴とする、請求項１記載の起動処理方法。

【請求項３】

該起動処理が複数のフェーズをそなえ、

該複数のフェーズのうち一のフェーズにおいて、該プロセッサのプロセッサキャッシュを該記憶領域として用い、

該情報処理装置の主記憶装置が使用可能な状態となった後に、該プロセッサキャッシュに格納されている該第２共通情報を該主記憶装置に複写するステップをそなえることを特徴とする、請求項２記載の起動処理方法。

【請求項４】

プロセッサをそなえた情報処理装置であって、

起動処理時に、該プロセッサにより実行されることにより当該情報処理装置の起動プロセスの一部を実現する複数のモジュールのうち、２以上のモジュールにより共通に用いられる第１共通情報をそなえる第１モジュールをロードする第１モジュール設定部と、

起動処理時に、該情報処理装置にそなえられた記憶領域に、該第１共通情報へアクセスするためのアクセス情報を格納するアクセス情報設定部と、

起動処理時に、該アクセス情報にアクセスするためのインタフェース情報をそなえる第２モジュールをロードする第２モジュール設定部と、

該起動処理時に、該第２モジュールにより、該インタフェース情報によって該アクセス情報にアクセスし、該アクセス情報を介して該第１共通情報を取得する第１情報取得部とをそなえることを特徴とする、情報処理装置。

【請求項５】

該起動処理時に、該記憶領域に、２以上のモジュールにより共通に用いられる第２共通情報を格納する共通領域を形成する共通領域設定部と、

該起動処理時に、該記憶領域に、該共通領域へアクセスするための共通領域アクセス情報を格納する共通領域アクセス情報設定部と、

該起動処理時に、該第２モジュールにより、該インタフェース情報によって該アクセス情報にアクセスし、アクセスした当該アクセス情報を用いて該第１共通情報にアクセスし、アクセスした当該第１共通情報に基づいて該共通領域アクセス情報にアクセスし、アクセスした当該共通領域アクセス情報を用いて、該共通領域に格納された該第２共通情報を取得する第２情報取得部とをそなえることを特徴とする、請求項４記載の情報処理装置。

【請求項６】

該起動処理が複数のフェーズをそなえ、

該複数のフェーズのうち一のフェーズにおいて、該プロセッサのプロセッサキャッシュを該記憶領域として用い、

該共通領域設定部が、該情報処理装置の主記憶装置が使用可能な状態となった後に、該プロセッサキャッシュに格納されている該第２共通情報を該主記憶装置に複写することを特徴とする、請求項５記載の情報処理装置。

【請求項７】

起動処理をコンピュータに実行させるための起動処理プログラムであって、

該コンピュータに実行されることにより情報処理装置の起動プロセスの一部を実現する複数のモジュールのうち、２以上のモジュールにより共通に用いられる第１共通情報をそなえる第１モジュールをロードするステップと、

該情報処理装置にそなえられた記憶領域に、該第１共通情報へアクセスするためのアクセス情報を格納するステップと、

該アクセス情報にアクセスするためのインタフェース情報をそなえる第２モジュールをロードするステップと、

該第２モジュールにより、該インタフェース情報によって該アクセス情報にアクセスし、該アクセス情報を介して該第１共通情報を取得するステップとを、該コンピュータに実

行させることを特徴とする、起動処理プログラム。
【請求項８】
起動処理をコンピュータに実行させるための起動処理プログラムを記録したコンピュータ読取可能な記録媒体であって、
該起動処理プログラムが、
該コンピュータに実行されることにより情報処理装置の起動プロセスの一部を実現する複数のモジュールのうち、２以上のモジュールにより共通に用いられる第１共通情報をそなえる第１モジュールをロードするステップと、
該情報処理装置にそなえられた記憶領域に、該第１共通情報へアクセスするためのアクセス情報を格納するステップと、
該アクセス情報にアクセスするためのインタフェース情報をそなえる第２モジュールをロードするステップと、
該第２モジュールにより、該インタフェース情報によって該アクセス情報にアクセスし、該アクセス情報を介して該第１共通情報を取得するステップとを、該コンピュータに実行させることを特徴とする、起動処理プログラムを記録したコンピュータ読取可能な記録媒体。
【発明の詳細な説明】
【技術分野】
【０００１】
本件は、情報処理装置の起動処理を行なう技術に関する。
【背景技術】
【０００２】
近年、ＰＣ（Personal Computer）の起動処理に、ＢＩＯＳ（Basic Input/Output System）に代えて、ＥＦＩ（Extensible Firmware Interface）を用いる手法が知られている（下記特許文献１，非特許文献１）。ＥＦＩは、ＢＩＯＳに代わるプラットフォーム・ファームウェアであり、ＰＣのハードウェアにＯＳ（Operating System）を読み込む準備をさせる作業を行なう。マザーボードにＥＦＩを組み込むことにより、ＰＣの起動時間を短縮でき、ハードウェアの設計やソフトウェアの開発が容易になる。このＥＦＩ　ＢＩＯＳは、Unified EFI Forumによって規格の策定が進められている。
【０００３】
ＥＦＩにおいては、ＯＳが起動するまでに、主に、ＳＥＣ（Security）、ＰＥＩ（Pre-EFI Initialization）、ＤＸＥ（Driver Execution Environment）及びＢＤＳ（Boot Device Selection）の４つのフェーズをそなえ、各フェーズの処理オーダは、ＳＥＣ、ＰＥＩ、ＤＸＥ、ＢＤＳの順となる。
また、各フェーズは、複数のモジュールから構成されており、ＰＥＩフェーズのモジュールをＰＥＩモジュール（ＰＥＩＭ）という。又、ＤＸＥフェーズのモジュールとしては、ＤＸＥドライバやＥＦＩドライバがある。
【０００４】
各モジュールは、ＢＩＯＳ　Ｆｌａｓｈから読み出され、ＣＰＵ（Central Processing Unit）キャッシュやメモリ等の実行領域に展開（ロード）された後、各モジュールが動作できる条件（ＥＦＩではプロトコルという）が整ったタイミングでそれぞれ実行される。なお、ＣＰＵキャッシュにおけるモジュールがロードされる領域のことを、ＣＡＲ（Cache As Ram）領域という。
【０００５】
図２６はＥＦＩの各フェーズを説明するための図であり、各フェーズ（EFI Phase）について、プログラムロード動作領域、ＣＰＵモード及び開発言語を示している。又、図２７は従来のＥＦＩにおけるモジュール構成を説明するための図、図２８はＥＦＩのフェーズ毎のプログラムロード領域を模式的に示す図である。
ＥＦＩにおいては、図２６に示すように、ＥＦＩフェーズでＣＰＵの動作モードが異なり、１６ビット（bit），３２ビット，６４ビット，３２ビットのＳＭＭ（System Manage

ment Mode）及び６４ビットのＳＭＭの各モードがある。又、図２６～図２８に示すように、ＥＦＩにおいては、モジュールのプログラムがロードされ、このプログラムが動作する領域（プログラムロード動作領域）もＥＦＩフェーズで異なる。なお、プログラムロード実行領域としては、図２６に示すように、ＢＩＯＳＦｌａｓｈ、ＣＡＲ及びメモリ（ＳＭＭを含む）がある。

【０００６】

従来のＥＦＩによるＰＣの起動処理においては、先ず、ＢＩＯＳ　Ｆｌａｓｈ上にプログラムがロードされＣＡＲの機能を有効にする。すなわち、ＳＥＣフェーズにおいては、図２７に示すように、ＢＩＯＳ　Ｆｌａｓｈ上にプログラムがロードされる。

そして、ＰＥＩフェーズにおいては、このＣＡＲ領域にプログラムがロードされ、このＰＥＩにおいて、メモリコントローラの初期化が行なわれ、メモリアクセスが有効になった後に、メモリ上にプログラムがロードされる。すなわち、ＰＥＩフェーズにおいては、その初期段階においてＣＡＲ領域にプログラムがロードされた後、メモリ上にプログラムがロードされる。

【０００７】

その後、ＰＥＩの後半において、チップセットの設定等が行なわれ、ＤＸＥが動作できるための必要最小限のチップセット設定等が行なわれる。ＤＸＥフェーズにおいては、メモリ上にプログラムがロードされ、その後、ＢＤＳが読み出される。

また、使用されるプログラム言語もフェーズによって異なり、図２６に示すように、ＳＥＣはアセンブラ言語であり、ＰＥＩ、ＤＸＥ、ＢＤＳはＣ言語がメインとなる。

【０００８】

そして、このような従来のＥＦＩにおいては、画面表示やキーボード入力のような各モジュール間で共通に必要な機能はライブラリ化され、各モジュールにこれらの共通機能ライブラリをそれぞれ付加している。

すなわち、図２７に示すように、従来のＥＦＩにおいては、ＰＥＩ及びＤＸＥ（ＢＤＳ）の各モジュールにそれぞれ共通機能ライブラリが付加されている。

【０００９】

また、従来のＥＦＩにおいては、異なるフェーズ間で、メモリサイズやＣＰＵの種別等の情報を共有する手法として、ＨＯＢ（Hand Of Block）構造体を用いることが知られている。

具体的には、ＰＥＩフェーズにおいて構築したＨＯＢ構造体と呼ばれる情報を、メモリを介してＤＸＥフェーズに受け渡すことにより、ＰＥＩフェーズとＤＸＥフェーズとの間での情報の共有を実現している。

【先行技術文献】

【特許文献】

【００１０】

【特許文献１】特開２００８－１０２９０６号公報

【非特許文献】

【００１１】

【非特許文献１】Vincent Zimmer著　「BIOSに代わるプラットフォーム・ファームウェアをあらゆるインテルシリコンに（日本語参考訳）」（Technology@IntelMagazine）２００４年１月　http://download.intel.com/jp/developer/jpdoc/it01043_j.pdf

【発明の概要】

【発明が解決しようとする課題】

【００１２】

しかしながら、このような従来のＥＦＩにおいては、図２７に示したように、各モジュールにそれぞれ共通機能ライブラリがそなえられているので、共通機能ライブラリのサイズが大きくなると、モジュール自身のサイズも大きくなる。これにより、ＥＦＩが使用するリソースサイズの肥大化が顕著となる。例えば、サーバや組込み系システムでは、必要とされるＲＡＳ（Reliability, Availability and Serviceability；信頼性、可用性、保

守性）機能が多く、共通機能数が多くなるので、共通機能ライブラリのサイズが大きくなる。従って、サーバや組込み系システムではリソースサイズの肥大化が生じ易い。

【００１３】

すなわち、従来のＥＦＩにおいては、各モジュールにそれぞれ共通機能ライブラリがそなえられているので、モジュールのサイズが大きくなり、ＥＦＩを格納するＢＩＯＳ　Ｆｌａｓｈの領域やＥＦＩが使用するプログラムロード動作領域のサイズも肥大化するという課題がある。

また、従来のＥＦＩにおけるフェーズ間での情報共有手法に関し、ＰＥＩからＤＸＥへのフェーズの切り替え時においては、３２ビットから６４ビットへのＣＰＵモードの切り替え等、ＤＸＥフェーズが動作するためのモジュール（ＤＸＥコアという）が整うまではＨＯＢ構造体の参照を行なうことはできないという課題がある。

【００１４】

例えば、ハードエラーが検出され、メモリ構成等を参照する必要がある場合等においては、ＰＥＩからＤＸＥへのフェーズの切り替え時に、これらのフェーズ間での情報共有が必要になるケースが発生する。しかしながら、上述の如く、ＤＸＥコアの準備が完了するまで情報共有を行なうことができず、処理の遅滞をまねくおそれがある。

また、各モジュールにおいて、共通機能ライブラリが展開されるアドレスはモジュール毎に異なり、これらのモジュール間で共通機能ライブラリの情報を共有することもできない。従って、従来のＥＦＩにおいては、たとえ共通機能ライブラリに共有域を設けたとしても、この共有域の情報をモジュール間で共有することができず、又、フェーズ切り替え時における情報参照も行なうことができない。

【００１５】

本件の目的の一つは、このような課題に鑑み創案されたもので、情報処理の起動処理にかかるプログラムサイズを小さくするとともに、起動処理過程におけるフェーズ間での情報を共有できるようにすることである。

なお、前記目的に限らず、後述する発明を実施するための形態に示す各構成により導かれる作用効果であって、従来の技術によっては得られない作用効果を奏することも本発明の他の目的の1つとして位置付けることができる。

【課題を解決するための手段】

【００１６】

このため、この起動処理方法は、プロセッサをそなえた情報処理装置の起動処理方法であって、該プロセッサにより実行されることにより当該情報処理装置の起動プロセスの一部を実現する複数のモジュールのうち、２以上のモジュールにより共通に用いられる第１共通情報をそなえる第１モジュールをロードするステップと、

該情報処理装置にそなえられた記憶領域に、該第１共通情報へアクセスするためのアクセス情報を格納するステップと、該アクセス情報にアクセスするためのインタフェース情報をそなえる第２モジュールをロードするステップと、該第２モジュールにより、該インタフェース情報によって該アクセス情報にアクセスし、該アクセス情報を介して該第１共通情報を取得するステップとをそなえる。

【００１７】

また、この情報処理装置は、プロセッサをそなえた情報処理装置であって、起動処理時に、該プロセッサにより実行されることにより情報処理装置の起動プロセスの一部を実現する複数のモジュールのうち、２以上のモジュールにより共通に用いられる第１共通情報をそなえる第１モジュールをロードする第１モジュール設定部と、起動処理時に、該情報処理装置にそなえられた記憶領域に、該第１共通情報へアクセスするためのアクセス情報を格納するアクセス情報設定部と、起動処理時に、該アクセス情報にアクセスするためのインタフェース情報をそなえる第２モジュールをロードする第２モジュール設定部と、該起動処理時に、該第２モジュールにより、該インタフェース情報によって該アクセス情報にアクセスし、該アクセス情報を介して該第１共通情報を取得する第１情報取得部とをそなえる。

【００１８】
さらに、この起動処理プログラムは、起動処理をコンピュータに実行させるための起動処理プログラムであって、該コンピュータに実行されることにより情報処理装置の起動プロセスの一部を実現する複数のモジュールのうち、２以上のモジュールにより共通に用いられる第１共通情報をそなえる第１モジュールをロードするステップと、該情報処理装置にそなえられた記憶領域に、該第１共通情報へアクセスするためのアクセス情報を格納するステップと、該アクセス情報にアクセスするためのインタフェース情報をそなえる第２モジュールをロードするステップと、該第２モジュールにより、該インタフェース情報によって該アクセス情報にアクセスし、該アクセス情報を介して該第１共通情報を取得するステップとを、該コンピュータに実行させる。又、このコンピュータ読取可能な記録媒体は、上述した起動処理プログラムを記録したものである。
【発明の効果】
【００１９】
開示の技術によれば、情報処理装置の起動処理において、第１モジュールにそなえられた第１共通情報を第２モジュールが取得して使用することができるので、複数のモジュール間で第１共通情報を共有することができる。これにより、情報処理装置の起動処理にかかるモジュールやプログラムのサイズを小さくすることができ、ロードされるメモリサイズやＢＩＯＳ　Ｆｌａｓｈの容量を小さく構成することができる。従って、処理速度の向上や製造コストを低減することができる。
【図面の簡単な説明】
【００２０】
【図１】実施形態の一例としての情報処理装置のハードウェア構成を模式的に示す図である。
【図２】実施形態の一例としての情報処理装置のＢＩＯＳ　Ｆｌａｓｈに格納されたＥＦＩのプログラムデータＰを模式的に示す図である。
【図３】実施形態の一例としての情報処理装置におけるＥＦＩのフェーズ毎のモジュールや各種情報のロード領域を模式的に示す図である。
【図４】実施形態の一例としての情報処理装置におけるデータ参照手法を模式的に示す図である。
【図５】実施形態の一例としての情報処理装置の起動時におけるＥＦＩのフェーズの処理を説明するためのフローチャートである。
【図６】実施形態の一例としての情報処理装置におけるＳＥＣフェーズの処理を説明するためのフローチャートである。
【図７】実施形態の一例としての情報処理装置におけるＰＥＩフェーズの処理を説明するためのフローチャートである。
【図８】実施形態の一例としての情報処理装置におけるＤＸＥ（ＢＤＳ）フェーズの処理を説明するためのフローチャートである。
【図９】実施形態の一例としての情報処理装置におけるＰＥＩフェーズ（ＣＡＲ領域）での共通ライブラリＳＬの読み出し手法を説明するためのフローチャートである。
【図１０】実施形態の一例としての情報処理装置におけるＰＥＩフェーズ（ＣＡＲ領域）での共通ライブラリＳＬへの参照経路を説明する図である。
【図１１】実施形態の一例としての情報処理装置におけるＰＥＩフェーズ（ＣＡＲ領域）での共通領域に格納された共通領域情報の読み出し手法を説明するためのフローチャートである。
【図１２】実施形態の一例としての情報処理装置におけるＰＥＩフェーズ（ＣＡＲ領域）での共通領域への参照経路を説明する図である。
【図１３】実施形態の一例としての情報処理装置におけるＰＥＩフェーズ（メモリ領域）での共通ライブラリＳＬの読み出し手法を説明するためのフローチャートである。
【図１４】実施形態の一例としての情報処理装置におけるＰＥＩフェーズ（メモリ領域）での共通ライブラリＳＬへの参照経路を説明する図である。

【図１５】実施形態の一例としての情報処理装置におけるＰＥＩフェーズ（メモリ領域）での共通領域に格納された共通領域情報の読み出し手法を説明するためのフローチャートである。
【図１６】実施形態の一例としての情報処理装置におけるＰＥＩフェーズ（メモリ領域）での共通領域への参照経路を説明する図である。
【図１７】実施形態の一例としての情報処理装置におけるＤＸＥフェーズ（メモリ領域）での共通ライブラリＳＬの読み出し手法を説明するためのフローチャートである。
【図１８】実施形態の一例としての情報処理装置におけるＤＸＥフェーズ（メモリ領域）での共通ライブラリＳＬへの参照経路を説明する図である。
【図１９】実施形態の一例としての情報処理装置におけるＤＸＥフェーズ（メモリ領域）での共通領域に格納された共通領域情報の読み出し手法を説明するためのフローチャートである。
【図２０】実施形態の一例としての情報処理装置におけるＤＸＥフェーズ（メモリ領域）での共通領域への参照経路を説明する図である。
【図２１】実施形態の一例としての情報処理装置におけるＳＭＭフェーズ（ＰＥＩ，ＤＸＥ）での共通ライブラリＳＬの読み出し手法を説明するためのフローチャートである。
【図２２】実施形態の一例としての情報処理装置におけるＳＭＭフェーズ（ＰＥＩ）及びＳＭＭフェーズ（ＤＸＥ）での共通ライブラリＳＬへの参照経路を説明する図である。
【図２３】実施形態の一例としての情報処理装置におけるＳＭＭフェーズ（ＰＥＩ，ＤＸＥ）での共通領域に格納された共通領域情報の読み出し手法を説明するためのフローチャートである。
【図２４】実施形態の一例としての情報処理装置におけるＳＭＭフェーズ（ＰＥＩ）での共通領域への参照経路を説明する図である。
【図２５】実施形態の一例としての情報処理装置におけるＳＭＭフェーズ（ＤＸＥ）での共通領域への参照経路を説明する図である。
【図２６】ＥＦＩの各フェーズを説明するための図である。
【図２７】従来のＥＦＩにおけるモジュール構成を説明するための図である。
【図２８】ＥＦＩのフェーズ毎のプログラムロード領域を模式的に示す図である。
【発明を実施するための形態】
【００２１】
以下、図面を参照して本起動処理方法，情報処理装置及び起動プログラムに係る実施の形態を説明する。
図１は実施形態の一例としての情報処理装置のハードウェア構成を模式的に示す図である。図２は実施形態の一例としての情報処理装置１００のＢＩＯＳ　Ｆｌａｓｈ２２に格納されたＥＦＩのプログラムデータＰを模式的に示す図である。図３は実施形態の一例としての情報処理装置におけるＥＦＩのフェーズ毎のモジュールや各種情報のロード領域を模式的に示す図である。
【００２２】
本情報処理装置１００は、図１に示すように、ＣＰＵ１０，ＲＡＭ２０，ＲＯＭ２１，ＢＩＯＳ　Ｆｌａｓｈ２２，ストレージ２３，ディスプレイ２４，キーボード２５及びマウス２６をそなえたコンピュータである。又、キーボード２５及びマウス２６は入力装置であり、オペレータはこれらのキーボード２５やマウス２６を操作して、本情報処理装置１００の再起動指示を含む種々の指示や情報の入力操作を行なう。又、この情報処理装置１００は、図示しない電源スイッチをそなえ、オペレータがこの電源スイッチを操作することにより電力投入が行なわれる。
【００２３】
ストレージ２３は、ハードディスクドライブ（Hard disk drive：ＨＤＤ）、ＳＳＤ（Solid State Drive）等の記憶装置であって、ＯＳや種々のプログラムやデータを格納するものである。
ディスプレイ２４は、種々のデータやオペレータに対するメッセージ等の情報を表示す

る表示装置である。

【００２４】

ＲＯＭ２１は、ＣＰＵ１０が実行するプログラムや種々のデータを格納する記憶装置である。

ＲＡＭ２０は、種々のデータやプログラムを一時的に格納する主記憶装置であって、ＣＰＵ１０がプログラムを実行する際に、データやプログラムを一時的に格納・展開して用いる。又、ＲＡＭ２０は、本情報処理装置１００の起動過程において、その所定の記憶領域が固有のメモリ領域２０１（図３参照）やＳＭＭ固定のメモリ領域２１１（図３参照）として機能する。固有のメモリ領域２０１には、管理域２００１と共通領域２０２とが含まれ、管理域２００１には、後述する共通ライブラリへのアドレス情報２０１と、同じく後述する共通領域へのアドレス情報２０３とが格納される。又、共通領域２０２の詳細についても後述する。

【００２５】

ＢＩＯＳ　Ｆｌａｓｈ２２は、ＥＦＩのプログラムデータＰを格納するメモリであり、例えば、フラッシュメモリ（フラッシュＲＯＭ）である。

プログラムデータＰは、図２に示すように、ｎ個（ｎは自然数：図２に示す例ではｎ＝３）のモジュールプログラムＰｎ（図２に示す例ではＰ１～Ｐ３）や、起動処理を制御する制御プログラムＰ０をそなえる。モジュールプログラムＰｎは、後述するＥＦＩの個々のモジュールを実現するプログラムであり、各モジュール毎にそなえられている。各モジュールプログラムＰｎは、それぞれヘッダ部Ｈと実行可能イメージＰＩをそなえている。

【００２６】

制御プログラムＰ０は、後述するＣＰＵ１０に実行されることにより、モジュールプログラムＰｎを所定の順序でロードさせ、ＥＦＩの起動処理を実現させるプログラムである。本情報処理装置１００の起動時において、ＣＰＵ１０がこの制御プログラムＰ０に従い、ＢＩＯＳ　Ｆｌａｓｈ２２からモジュールプログラムＰｎの実行可能イメージＰＩを順次読み出し、情報処理装置１００における所定のロード領域に展開（ロード）して実行することにより、起動処理が行なわれる。

【００２７】

このように、情報処理装置１００における所定の領域にロードされた各モジュールの実行可能イメージＰＩは、各モジュールとしての機能を実現する。

なお、図２に示す例においては、モジュールプログラムＰｎの例としてＰＥＩモジュールにかかるモジュールプログラムＰ１，Ｐ２とＤＸＥドライバにかかるモジュールプログラムＰ３とを示しているが、これに限定されるものではなく、ＳＥＣモジュールや他の機能を実現するプログラムをモジュールプログラムに含めることを否定するものではなく、種々変形して実施することができる。

【００２８】

また、本情報処理装置においては、ＢＩＯＳ　Ｆｌａｓｈ２２に格納された各モジュールプログラムＰ１の実行可能イメージＰに、共有ライブラリＳＬや共有インタフェースライブラリＳＩが含まれている。

共通ライブラリＳＬは、例えば、ディスプレイ２４への画面表示やキーボード２５やマウス２６の入力制御のような、複数のモジュールがそれぞれ使用しうる共通の情報（第１共通情報）である。この共通ライブラリＳＬは、これらの複数のモジュール間で共通に必要な機能をライブラリ化することにより、複数のモジュールが共通に使用できるようにしたものである。この共通ライブラリＳＬは、当該共通ライブラリＳＬをそなえるモジュールが使用できるとともに、後述する共通インタフェースライブラリＳＩをそなえる他のモジュールからも使用される。

【００２９】

また、共通ライブラリＳＬは、後述するＣＡＲ領域１０１やＲＡＭ２０における所定の領域へアクセスするためのアクセス関数もそなえている。後述するＣＰＵ１０が、このアクセス関数を実行することにより、例えば、ＣＡＲ領域１０１やＲＡＭ２０における所定

の領域へアクセスするためのアドレス演算が行なわれる。
これにより、例えば、ＰＥＩモジュールＭ１１の共通ライブラリＳＬからＣＡＲ領域１０１の共通領域へのアドレス情報１０３にアクセスすることができる。同様に、ＰＥＩモジュールＭ２１やＰＥＩモジュールＭ２１の共通ライブラリＳＬから固定のメモリ領域２０１の共通領域へのアドレス情報２０３へアクセスすることができる。又、ＤＸＥドライバＤ１やＤＸＥドライバＤ３の共通ライブラリＳＬから固定のメモリ領域２０１の共通領域へのアドレス情報２０３へアクセスすることができる。
【００３０】
共通ライブラリＳＬは、本情報処理装置１００の起動プロセスの一部を実現する複数のモジュールのうち、２以上のモジュールにより共通に用いられる第１共通情報として機能するのである。
なお、ＰＥＩ，ＤＸＥ，ＳＭＭ（ＤＸＥ）の各フェーズにおいてロード領域が一致するグループ毎において、同一の共通ライブラリＳＬが重複してそなえられることがないようにすることが望ましい。つまり、共通ライブラリＳＬは、上記グループ毎に１つ構築される。これにより、共通ライブラリＳＬの格納に要する容量を削減し、各モジュールのサイズを小さくすることができる。すなわち、ＢＩＯＳ　Ｆｌａｓｈ２２やこれらのモジュールをロードするプログラムロード動作領域を小さくすることができる。望ましくは、共通ライブラリＳＬは、上記フェーズ毎に１つ構築される。
【００３１】
図３に示す例においては、ＰＥＩフェーズの初期段階においてＣＡＲ領域１０１に展開される２つのＰＥＩモジュールＭ１１，Ｍ１２からなる第１のグループにおいては、ＰＥＩモジュールＭ１１に共有ライブラリＳＬがそなえられている。なお、ＰＥＩモジュールＭ１１が、自身にそなえた共通ライブラリＳＬを使用することは言うまでもない。以下、ＰＥＩフェーズにおいて、モジュールのロード場所がＣＡＲ領域１０１の状態をＰＥＩフェーズ（ＣＡＲ領域）と表す場合がある。
【００３２】
また、ＰＥＩフェーズの途中からは、ＲＡＭ２０が使用可となり、このＲＡＭ２０（メモリ領域）にモジュールが展開される。図３に示す例において、メモリ領域に展開される２つのＰＥＩモジュールＭ２１，Ｍ２２からなる第２のグループにおいては、ＰＥＩモジュールＭ２１に共有ライブラリＳＬがそなえられている。なお、ＰＥＩモジュールＭ２１が、自身にそなえた共通ライブラリＳＬを使用することは言うまでもない。以下、ＰＥＩフェーズにおいてモジュールのロード場所がＲＡＭ２０の状態をＰＥＩフェーズ（メモリ領域）と表す場合がある。
【００３３】
さらに、図３に示す例において、ＰＥＩフェーズにおいてＳＭＭ領域３０１に展開される２つのＰＥＩモジュールＭ２３，Ｍ２４からなる第３のグループにおいては、ＰＥＩモジュールＭ２３に共有ライブラリＳＬがそなえられている。なお、ＰＥＩモジュールＭ２３が、自身にそなえた共通ライブラリＳＬを使用することは言うまでもない。そして、以下、ＰＥＩフェーズにおいてモジュールのロード場所がＳＭＭの状態をＳＭＭフェーズ（ＰＥＩ）と表す場合がある。
【００３４】
同様に、図３に示す例において、ＤＸＥフェーズにおいてメモリ領域に展開される２つのＤＸＥドライバＤ１，Ｄ２からなる第４のグループにおいては、ＤＸＥドライバＤ１に共有ライブラリＳＬがそなえられている。なお、ＤＸＥドライバＤ１，Ｄ３が、自身にそなえた共通ライブラリＳＬを使用することは言うまでもない。又、ＤＸＥフェーズにおいてＳＭＭ領域３０２に展開される２つのＤＸＥドライバＤ３，Ｄ４からなる第５のグループにおいては、ＤＸＥドライバＤ３に共有ライブラリＳＬがそなえられている。そして、以下、ＤＸＥフェーズにおいてモジュールのロード場所がＳＭＭの状態をＳＭＭフェーズ（ＤＸＥ）と表す場合がある。
【００３５】

共通インタフェースライブラリＳＩは、他のモジュールに含まれる共通ライブラリＳＬや後述する共通領域１０２，２０２へアクセスするためのインタフェースであり、例えば、管理域１００１，２００１，２１０１における所定のアドレスへのポインタをそなえる。

各モジュールは、この共通インタフェースライブラリＳＩを介することで、後述する管理域１００１，２００１，２１０１を経由し、共通ライブラリＳＬや共通領域１０２，２０２にアクセスすることができる。すなわち、共通インタフェースライブラリＳＩは、共通ライブラリＳＬや共通領域１０２，２０２に格納された共通領域情報へアクセスするためのインタフェース情報として機能する。

【００３６】

また、ＢＩＯＳ　Ｆｌａｓｈ２２には、後述する共通ライブラリへのアドレス情報１０４，２０４，２１２や共通領域へのアドレス情報１０３，２０３の他、共通領域１０２，２０２に格納する共通情報が格納されている。

ＣＰＵ（プロセッサ）１０は、種々の制御や演算を行なう処理装置であり、ストレージ２３等に格納されたプログラムを実行することにより、種々の機能を実現する。又、ＣＰＵ１０は、図示しないキャッシュメモリ（プロセッサキャッシュ）をそなえている。このキャッシュメモリは、データや命令などの情報を一時的に格納する記憶領域であり、ＣＰＵ１０がアクセスしたいデータやそのアドレス、状態、設定など属性情報をコピーし保持する。なお、このキャッシュメモリは、ＰＥＩフェーズ（ＣＡＲ領域）においては、このＰＥＩモジュールが展開（ロード）され、ＰＥＩジュールの実行領域となる。

【００３７】

そして、このＣＰＵ１０は、本情報処理装置１００の起動時において、ＢＩＯＳ　Ｆｌａｓｈ２２に格納されたＥＦＩのプログラムＰ（制御プログラムＰ０）を実行することにより、本情報処理部１００の起動処理を行なう起動処理部１１として機能する。以下、本情報処理装置１００における起動処理について説明する。なお、この起動処理は、例えば、本情報処理装置１００の電源投入時や再起動時に行なわれる。

【００３８】

この起動処理部１１としての機能を実現するためのプログラム（起動処理プログラム）は、例えばフレキシブルディスク，ＣＤ（ＣＤ－ＲＯＭ，ＣＤ－Ｒ，ＣＤ－ＲＷ等），ＤＶＤ（ＤＶＤ－ＲＯＭ，ＤＶＤ－ＲＡＭ，ＤＶＤ－Ｒ，ＤＶＤ＋Ｒ，ＤＶＤ－ＲＷ，ＤＶＤ＋ＲＷ，ＨＤ　ＤＶＤ等），ブルーレイディスク，磁気ディスク，光ディスク，光磁気ディスク等の、コンピュータ読取可能な記録媒体に記録された形態で提供される。そして、コンピュータはその記録媒体からプログラムを読み取って内部記憶装置または外部記憶装置に転送し格納して用いる。又、そのプログラムを、例えば磁気ディスク，光ディスク，光磁気ディスク等の記憶装置（記録媒体）に記録しておき、その記憶装置から通信経路を介してコンピュータに提供するようにしてもよい。

【００３９】

起動処理部１１としての機能を実現する際には、内部記憶装置（本実施形態ではＲＡＭ２０やＲＯＭ２１）に格納されたプログラムがコンピュータのマイクロプロセッサ（本実施形態ではＣＰＵ１０）によって実行される。このとき、記録媒体に記録されたプログラムをコンピュータが読み取って実行するようにしてもよい。

なお、本実施形態において、コンピュータとは、ハードウェアとオペレーティングシステムとを含む概念であり、オペレーティングシステムの制御の下で動作するハードウェアを意味している。又、オペレーティングシステムが不要でアプリケーションプログラム単独でハードウェアを動作させるような場合には、そのハードウェア自体がコンピュータに相当する。ハードウェアは、少なくとも、ＣＰＵ等のマイクロプロセッサと、記録媒体に記録されたコンピュータプログラムを読み取るための手段とをそなえており、本実施形態においては、情報処理装置１００がコンピュータとしての機能を有しているのである。

【００４０】

起動処理部１１は、本情報処理部１００の起動処理をＥＦＩの仕様に従って行なうもの

であり、本情報処理装置１００の起動（電源投入）からＯＳへの制御の移行に至るまでのプラットフォームの初期化にかかる処理を実施する。

この起動処理部１１は、ＳＥＣ，ＰＥＩ，ＤＸＥ及びＤＸＥの４つのフェーズを順次行なうことにより、ＥＦＩの仕様に沿った起動処理を実現する。ＥＦＩにおいては、各フェーズで利用可能なインフラストラクチャは中心となるフレームワークによって提供される。そして、プラットフォーム固有の機能は相互通信可能なモジュールを使って実装される。以下、ＰＥＩフェーズのモジュールをＰＥＩモジュールという。又、ＤＸＥフェーズのモジュールとしては、ＤＸＥドライバやＥＦＩドライバがある。

【００４１】

なお、図３に示す例においては、ＰＥＩモジュールに符号Ｍ１１，Ｍ１２，Ｍ２１～Ｍ２４を付して表し、又、ＤＸＥドライバに符号Ｄ１～Ｄ４を付して表す。又、以下、ＰＥＩモジュールを示す符号としては、複数のＰＥＩモジュールのうち１つを特定する必要があるときには符号Ｍ１１，Ｍ１２，Ｍ２１～Ｍ２４を用いるが、任意のＰＥＩモジュールを指すときには符号Ｍを用いる。同様に、以下、ＤＸＥドライバを示す符号としては、複数のＤＸＥドライバのうち１つを特定する必要があるときには符号Ｄ１～Ｄ４を用いるが、任意のＤＸＥドライバを指すときには符号Ｄを用いる。

【００４２】

起動処理部１１は、図１に示すように、モジュール設定部１１１，アクセス情報設定部１１２，共通領域アクセス情報設定部１１３，共通領域設定部１１４及び情報取得部１１５をそなえ、これらの各部が機能することにより起動処理を実現する。

モジュール設定部１１１は、ＢＩＯＳ　Ｆｌａｓｈ２２から、各モジュールに対応するモジュールプログラムＰｎの実行可能イメージＰＩを読み出し、それぞれの所定の実行領域（プログラムロード動作領域，ロード領域）にロード（設定，格納）するローダとして機能する。

【００４３】

以下、モジュールに対応するモジュールプログラムＰｎの実行可能イメージＰＩをロード領域にロードすることを、便宜上、単にモジュールをロードすると表現する。

モジュール設定部１１１は、ＳＥＣフェーズにおいて、ＢＩＯＳ　Ｆｌａｓｈ２２にＳＥＣモジュール（図示省略）をロードする。そして、ＣＰＵ１０がＢＩＯＳ　Ｆｌａｓｈ２２上のＳＥＣモジュールを実行することにより、ＣＰＵ１０のキャッシュが設定されＣＡＲの機能が有効になる。

【００４４】

また、ＰＥＩフェーズにおいては、ＣＡＲ領域にＰＥＩモジュールＭがロードされ、このＰＥＩにおいて、メモリコントローラの初期化が行なわれ、ＲＡＭ２０に対するメモリアクセスが有効になった後に、メモリ上に各モジュールの実行可能イメージＰ１がロードされるのである。

モジュール設定部１１１は、ＥＦＩの各フェーズを実行するためのＰＥＩモジュールＭやＤＸＥドライバＤのデータをＢＩＯＳ　Ｆｌａｓｈ２２から読み出し、キャッシュメモリ１２やＲＡＭ２０等のそれぞれの所定の実行領域にロードする。

【００４５】

具体的には、モジュール設定部１１１は、ＰＥＩフェーズの初期段階においては、ＢＩＯＳ　Ｆｌａｓｈ２２から読み出したＰＥＩモジュールＭをＣＰＵ１０のキャッシュメモリにおけるＣＡＲ領域１０１に展開する。又、モジュール設定部１１１は、ＰＥＩフェーズにおいて、ＣＰＵ１０がＰＥＩモジュールＭを実行することによりＲＡＭ２０の初期設定が完了し、このＲＡＭ２０が有効となり使用可能となった後においては、このＲＡＭ２０上の固定のメモリ領域２０１もしくはＳＭＭ領域３０１にＰＥＩモジュールＭをロードする。

【００４６】

また、モジュール設定部１１１は、ＤＸＥフェーズにおいては、ＢＩＯＳ　Ｆｌａｓｈ２２から読み出したＤＸＥドライバＤを、ＲＡＭ２０もしくはＲＡＭ２０におけるＳＭＭ

領域３１１上にロードする。
そして、このモジュール設定部１１１によってロードされるモジュールには、共通ライブラリＳＬや共通インタフェースライブラリＳＩをそなえるものも含まれる。すなわち、モジュール設定部１１１は、起動処理時に、２以上のモジュールにより共通に用いられる共通ライブラリＳＬをそなえるモジュールをロードする第１モジュール設定部として機能する。更に、モジュール設定部１１１は、起動処理時に、共通インタフェースライブラリＳＩをそなえるモジュールをロードする第２モジュール設定部としても機能する。
【００４７】
共通領域設定部１１４は、ＣＰＵ１０のキャッシュメモリにおけるＣＡＲ領域１０１やＲＡＭ２０における固有のメモリ領域２０１に、共通領域情報（第２共通情報）を格納する共通領域１０２，２０２を設定する。
ここで、共通領域情報は、例えば、メモリサイズやＣＰＵ１０の種別等、ＥＦＩの複数のフェーズ間で共通に使用される情報である。このような共通領域情報は、例えば、ＰＥＩフェーズのＰＥＩモジュールＭとＤＸＥフェーズのＤＸＥドライバＤとのように、異なるフェーズのモジュールによって共通して使用される（共有される）。
【００４８】
共通領域設定部１１４は、ＰＥＩフェーズの初期段階において、ＣＡＲ領域１０１における所定位置に共通領域１０２を確保し、ＢＩＯＳ　Ｆｌａｓｈ２２から共通領域情報を読み出して、この読み出した共通領域情報を共通領域１０２に格納する。ここで、共通領域１０２においては、同一の共通領域情報が重複することがないように格納することが望ましい。
【００４９】
また、共通領域設定部１１４は、ＰＥＩフェーズにおいてＲＡＭ２０が有効となった後においては、ＲＡＭ２０の固定のメモリ領域２０１における所定位置に共通領域２０２を確保する。そして、共通領域設定部１１４は、ＣＡＲ領域１０１の共通領域１０２に格納されている共通領域情報を読み出して、この読み出した共通領域情報を共通領域２０２に格納する。すなわち、共通領域設定部１１４は、ＣＡＲ領域１０１の共通領域１０２の共通領域情報を固有のメモリ領域２０１の共通領域２０２にコピーする。
【００５０】
これにより、ＰＥＩフェーズ（ＣＡＲ領域）において使用されていた共通領域情報を、ＰＥＩフェーズ（メモリ領域）やＤＸＥフェーズやＳＭＭフェーズ（ＰＥＩ），ＳＭＭフェーズ（ＤＸＥ）においても使用（共用）することができる。
アクセス情報設定部１１２は、ＣＰＵ１０のキャッシュメモリにおけるＣＡＲ領域１０１や、ＲＡＭ２０における固有のメモリ領域２０１，ＳＭＭ固定のメモリ領域２１１に、共通ライブラリへのアドレス情報１０４，２０４，２１２を格納（設定）する。
【００５１】
具体的には、アクセス情報設定部１１２は、ＰＥＩフェーズの初期段階において、ＢＩＯＳ　Ｆｌａｓｈ２２からＰＥＩ用３２ビットの共通ライブラリへのアドレス情報を読み出して、この読み出した共通ライブラリへのアドレス情報を、ＣＰＵ１０のキャッシュメモリにおけるＣＡＲ領域１０１の所定の領域に格納する。
ここで、共通ライブラリへのアドレス情報１０４とは、モジュールにそなえられた共通ライブラリＳＬへアクセスするためのアドレス情報であり、例えば、共通ライブラリＳＬへのポインタを含む。又、この共通ライブラリＳＬへのポインタとしては、単なるアドレスを示すポインタ（共有情報ポインタ）の他、関数ポインタが用いられる。
【００５２】
また、アクセス情報設定部１１２は、ＲＡＭ２０が有効となった後に、ＢＩＯＳ　Ｆｌａｓｈ２２からＰＥＩ用３２ビットの共通ライブラリへのアドレス情報２０４を読み出し、この読み出したＰＥＩ用３２ビットの共通ライブラリへのアドレス情報２０４をＲＡＭ２０上の固定のメモリ領域２０１に格納する。
さらに、アクセス情報設定部１１２は、ＲＡＭ２０におけるＳＭＭ（System Managemen

t Mode）領域に対しても、その特定の領域であるＳＭＭ固定のメモリ領域２１１に共通ライブラリにアドレス情報２１２の設定を行なう。

【００５３】

アクセス情報設定部１１２は、ＰＥＩフェーズにおいて、ＢＩＯＳ　Ｆｌａｓｈ２２からＤＸＥ用６４ビットの共通ライブラリへのアドレス情報２１２を読み出し、この読み出したＤＸＥ用６４ビットの共通ライブラリへのアドレス情報２１２をＳＭＭ固定のメモリ領域２１１に格納する。

なお、共有領域情報は、ＰＥＩ用３２ビットとＤＸＥ用６４ビットとでそのまま共用可能である。

【００５４】

共通領域アクセス情報設定部１１３は、ＣＰＵ１０のキャッシュメモリにおけるＣＡＲ領域１０１やＲＡＭ２０における固有のメモリ領域２０１に、共通領域へのアドレス情報１０３，２０３を格納する。

具体的には、共通領域アクセス情報設定部１１３は、ＰＥＩフェーズの初期段階において、ＣＰＵ１０のキャッシュメモリにおけるＣＡＲ領域１０１の所定の領域に、共通領域へのアドレス情報１０３を格納する。

【００５５】

ここで、共通領域へのアドレス情報１０３は、共通領域１０２へアクセスするための情報（共有領域アクセス情報）であり、例えば、共通領域１０２における共通領域情報の格納位置を表すポインタ（共有情報ポインタ）を含む。これにより、共通ライブラリＳＬ１０３にアクセスするモジュールは、共通領域１０２に格納された共通領域情報にアクセスすることが可能となる。

【００５６】

同様に、共通領域アクセス情報設定部１１３は、ＲＡＭ２０が使用可能となった後において、このＲＡＭ２０の所定の領域に、共通領域へのアドレス情報２０３を格納する。

ここで、共通領域へのアドレス情報２０３は、共通領域２０２へアクセスするための情報である共有領域アクセス情報であり、例えば、共通領域２０２における共通領域情報の格納位置を表すポインタ（共有情報ポインタ）を含む。これにより、共通ライブラリＳＬ２０３にアクセスするモジュールは、共通領域２０２に格納された共通領域情報にアクセスすることが可能となる。

【００５７】

なお、ＣＡＲ領域１０１における、共通ライブラリへのアドレス情報１０４及び共通領域へのアドレス情報１０３が格納される領域を管理領域１００１という場合がある。同様に、ＲＡＭ２０（固定のメモリ領域２０１）における、共通ライブラリへのアドレス情報及び共通領域へのアドレス情報が格納される領域を管理域２００１という場合がある。又、ＲＡＭ２０のＳＭＭ（固定のメモリ領域２１１）における、共通ライブラリへのアドレス情報が格納される領域を管理域２１０１という場合がある。

【００５８】

情報取得部１１５は、本情報処理装置１００の起動処理時において、ＰＥＩモジュールＭやＤＸＥドライバＤ等のモジュールを実行する際に、共通ライブラリＳＬや共通領域情報を参照・取得する。

この情報取得部１１５は、以下に詳述するように、共通インタフェースライブラリＳＩや共通ライブラリへのアドレス情報１０４，２０４，２１２を用いて共通ライブラリＳＬを取得する。すなわち、第１情報取得部として機能する。又、情報取得部１１５は、以下に詳述するように、共通インタフェースライブラリＳＩや共通ライブラリへのアドレス情報１０４，２０４，２１２、共通ライブラリＳＬ及び共通領域へのアドレス情報１０３，２０３を用いて、共通領域１０２，２０２の共通領域情報を取得する。すなわち、第２情報取得部として機能する。

【００５９】

図４は実施形態の一例としての情報処理装置１００におけるデータ参照手法を模式的に

示す図である。
起動処理部１１は、共通インタフェースライブラリＳＩをそなえるモジュールの実行に際して、各共通インタフェースライブラリＳＩを介して、管理域１００１，２００１，２１０１にアクセスする。そして、これらの管理域１００１，２００１，２１０１に格納された共通ライブラリへのアドレス情報１０４，２０４，２１２や共通領域へのアドレス情報１０３，２０３にアクセスする。起動処理部１１は、これらの情報を用いて共通ライブラリＳＬや共通領域情報にアクセスし、これらの情報を用いて本情報処理装置１００の起動処理を実行する。
【００６０】
上述の如く構成された、実施形態の一例としての情報処理装置１００の起動時におけるＥＦＩのフェーズの処理を、図５に示すフローチャート（ステップＡ１０～Ａ３０）に従って説明する。
本実施形態の一例としての情報処理装置１００に電源投入（ＰＯＮ：Power ON）が行なわれると、先ず、ＳＥＣフェーズが行なわれた後（ステップＡ１０）、ＰＥＩフェーズが行なわれ（ステップＡ２０）、更に、ＤＸＥ（ＢＤＳ）フェーズ（ステップＡ３０）が行なわれる。これらのＥＦＩにかかる各フェーズの処理によりシステムの初期化が完了し、その後、ＯＳローダによるＯＳの起動処理が開始される。
【００６１】
次に、本実施形態の一例としての情報処理装置１００におけるＳＥＣフェーズの処理を、図６に示すフローチャート（ステップＡ１０１～Ａ１０５）に従って説明する。
ＳＥＣフェーズにおいては、先ず、ＣＰＵ１０の初期化を行なった後（ステップＡ１０１）、ＣＡＲ領域１０１（ＣＰＵキャッシュ）の設定を行なう(ステップＡ１０２)。
そして、ＣＡＲ領域１０１のスタックポインタを予め規定された所定量ずらすことにより、共通ライブラリへのアドレス情報１０４や共通領域へのアドレス情報１０３及び共通領域１０２の領域を確保する（ステップＡ１０３）。なお、スタックポインタをずらす量は、共通ライブラリへのアドレス情報１０４，共通領域へのアドレス情報１０３及び共通領域１０２のサイズに対応する。
【００６２】
その後、ＣＡＲ領域１０１の設定を行ない（ステップＡ１０４）、ＢＩＯＳ　Ｆｌａｓｈ２２におけるＰＥＩの開始アドレスを読み出し（ステップＡ１０５）、ＳＥＣフェーズを修了する。
次に、本実施形態の一例としての情報処理装置１００におけるＰＥＩフェーズの処理を、図７に示すフローチャート（ステップＡ２０１～Ａ２０９）に従って説明する。
【００６３】
ＰＥＩフェーズにおいては、先ず、アクセス情報設定部１１２及び共通領域アクセス情報設定部１１３が、ＣＡＲのスタック領域にＰＥＩ用３２ビットの共通ライブラリへのアドレス情報１０４及び共通領域へのアドレス情報１０３を設定する（ステップＡ２０１）。
次に、モジュール設定部１１１が、ＰＥＩモジュールをＢＩＯＳ　Ｆｌａｓｈ２２から読み出してＣＡＲ領域１０１へロードし、これらのＰＥＩモジュールをそれぞれ実行する（ステップＡ２０２）。これらのＰＥＩモジュールによりＲＡＭ２０の初期化が実行され（ステップＡ２０３）、ＲＡＭ２０が使用可能になる。
【００６４】
アクセス情報設定部１１２及び共通領域アクセス情報設定部１１３は、固定のメモリ領域２０１に、ＰＥＩ用３２ビットの共通ライブラリへのアドレス情報２０４及びＰＥＩ用３２ビットの共通領域へのアドレス情報２０３を設定する（ステップＡ２０４）。
また、共通領域設定部１１４が、ＣＡＲ領域１０１の共通領域１０２に格納されている共通領域情報を固定のメモリ領域２０１の共通領域２０２にコピーする（ステップＡ２０５）。ここで、キャッシュが無効化される。
【００６５】

モジュール設定部１１１は、ＰＥＩモジュールＭをＲＡＭ２０（メモリ領域）にロードし、これらのＰＥＩモジュールＭの各処理を実行させる（ステップＡ２０６）。
次に、アクセス情報設定部１１２及び共通領域アクセス情報設定部１１３は、固定のメモリ領域２０１に、ＰＥＩ用３２ビットの共通ライブラリへのアドレス情報２０４及びＰＥＩ用３２ビットの共通領域へのアドレス情報２０３を設定する（ステップＡ２０７）。
【００６６】
その後、ＲＡＭ２０にＤＸＥのメイン部分をロードし（ステップＡ２０８）、このＲＡＭ２０上におけるＤＸＥコアの開始アドレスを読み出して（ステップＡ２０９）処理を修了する。
また、本実施形態の一例としての情報処理装置１００におけるＤＸＥ（ＢＤＳ）フェーズの処理を、図８に示すフローチャート（ステップＡ３０１～Ａ３０２）に従って説明する。
【００６７】
ＤＸＥ（ＢＤＳ）フェーズにおいては、先ず、アクセス情報設定部１１２及び共通領域アクセス情報設定部１１３が、固定のメモリ領域２０１にＤＸＥ用６４ビットの共通ライブラリへのアドレス情報１０４及び共通領域へのアドレス情報１０３を設定する（ステップＡ３０１）。
次に、モジュール設定部１１１が、ＤＸＥドライバＤをＢＩＯＳ　Ｆｌａｓｈ２２から読み出してＲＡＭ２０上にロードし、これらの各ＤＸＥドライバをそれぞれ実行することにより（ステップＡ３０２）、ＤＸＥ（ＢＤＳ）フェーズを完了させる。
【００６８】
なお、図７に示す例においては、ＰＥＩフェーズの途中（ステップＡ２０７参照）において、固定のメモリ領域２０１に、ＰＥＩ用３２ビットの共通ライブラリへのアドレス情報２０４及びＰＥＩ用３２ビットの共通領域へのアドレス情報２０３を設定しているが、これに限定されるものではない。
例えば、ＤＸＥ（ＢＤＳ）フェーズのステップＡ３０２においてＤＸＥドライバをＲＡＭ２０上にロードし、これらの各ＤＸＥドライバをそれぞれ実行した後に行なってもよく、適宜変更して実施することができる。
【００６９】
次に、本実施形態の一例としての情報処理装置１００におけるＰＥＩフェーズ（ＣＡＲ領域）での共通ライブラリＳＬの読み出し手法を、図１０を参照しながら、図９に示すフローチャート（ステップＢ１０～Ｂ３０に従って説明する）。なお、図１０は本情報処理装置１００におけるＰＥＩフェーズ（ＣＡＲ領域）での共通ライブラリＳＬへの参照経路を説明する図である。
【００７０】
ＰＥＩフェーズ（ＣＡＲ領域）において、ＰＥＩモジュールＭ１２（第２モジュール）がＰＥＩモジュールＭ１１（第１モジュール）の共通ライブラリＳＬの関数を用いる例について説明する。
ＣＡＲ領域において、ＰＥＩモジュールＭ１２は共通インタフェースライブラリＳＩを呼び出し（ステップＢ１０：図１０の符号Ｐ０１参照）、ＣＡＲ領域１０１の共通ライブラリへのアドレス情報１０４を取得する（ステップＢ２０：図１０の符号Ｐ０２参照）。ＰＥＩモジュールＭ１２は、この共通ライブラリへのアドレス情報１０４を用いて、ＰＥＩモジュールＭ１１の共通ライブラリＳＬの関数を呼び出す（ステップＢ３０：図１０の符号Ｐ０３参照）。
【００７１】
このように、ＰＥＩフェーズ（ＣＡＲ領域）においては、ＰＥＩモジュールＭ１２は、共通インタフェースライブラリＳＩを用いて、ＣＡＲ領域１０１の共通ライブラリへのアドレス情報１０４を介してＰＥＩモジュールＭ１１の共通ライブラリＳＬを読み出すことができる。又、ＰＥＩモジュールＭ１１も自身がそなえる共通ライブラリＳＬを用いることができる。

【００７２】
従って、ＰＥＩモジュールＭ１１の共通ライブラリＳＬを、ＰＥＩモジュールＭ１１とＰＥＩモジュールＭ１２とによって共用することが可能となる。
次に、本実施形態の一例としての情報処理装置１００におけるＰＥＩフェーズ（ＣＡＲ領域）での共通領域に格納された共通領域情報の読み出し手法を、図１２を参照しながら、図１１に示すフローチャート（ステップＣ１０～Ｃ５０に従って説明する）。なお、図１２は本情報処理装置１００におけるＰＥＩフェーズ（ＣＡＲ領域）での共通領域１０２への参照経路を説明する図である。
【００７３】
ＣＡＲ領域において、ＰＥＩモジュールＭ１２（第２モジュール）は共通インタフェースライブラリＳＩを呼び出し（ステップＣ１０：図１２の符号Ｐ１１参照）、ＣＡＲ領域１０１の共通ライブラリへのアドレス情報１０４を取得する（ステップＣ２０：図１２の符号Ｐ１２参照）。ＰＥＩモジュールＭ１２は、この共通ライブラリへのアドレス情報１０４を用いて、ＰＥＩモジュールＭ１１（第１モジュール）の共通ライブラリＳＬの関数を呼び出す（ステップＣ３０：図１２の符号Ｐ１３参照）。
【００７４】
ＰＥＩモジュールＭ１２は、この共通ライブラリＳＬから、ＣＡＲ領域１０１の共通領域へのアドレス情報１０３へアクセスするための情報（例えば、ポインタ）を取得し、この情報に基づいてＣＡＲ領域１０１の共通領域へのアドレス情報１０３を取得する（ステップＣ４０：図１２の符号Ｐ１４参照）。ＰＥＩモジュールＭ１２は、この共通領域へのアドレス情報１０３を用いて、ＣＡＲ領域１０１の共通領域１０２にアクセスし（ステップＣ５０：図１２の符号Ｐ１５参照）、この共通領域１０２に格納されている共通領域情報を取得する。
【００７５】
このように、ＰＥＩフェーズ（ＣＡＲ領域）においては、ＰＥＩモジュールＭ１２は、共通インタフェースライブラリＳＩを用いて、ＣＡＲ領域１０１の共通ライブラリへのアドレス情報１０４を介してＰＥＩモジュールＭ１１の共通ライブラリＳＬを読み出す。更に、この共通ライブラリＳＬを用いてＣＡＲ領域１０１の共通領域へのアドレス情報１０３を取得し、この共通領域へのアドレス情報１０３を用いて共通領域情報を取得することができる。
【００７６】
また、ＰＥＩモジュールＭ１１も自身がそなえる共通ライブラリＳＬを用いて、ＣＡＲ領域１０１の共通領域へのアドレス情報１０３を取得し、この共通領域へのアドレス情報１０３を用いて共通領域情報を取得することができる。
従って、ＣＡＲ領域１０１の共通領域１０２の共通領域情報を、ＰＥＩモジュールＭ１１とＰＥＩモジュールＭ１２とによって共用することが可能となる。
【００７７】
次に、本実施形態の一例としての情報処理装置１００におけるＰＥＩフェーズ（メモリ領域）での共通ライブラリＳＬの読み出し手法を、図１４を参照しながら、図１３に示すフローチャート（ステップＤ１０～Ｄ３０に従って説明する）。なお、図１４は本情報処理装置１００におけるＰＥＩフェーズ（メモリ領域）での共通ライブラリＳＬへの参照経路を説明する図である。
【００７８】
ＰＥＩフェーズ（メモリ領域）において、ＰＥＩモジュールＭ２２（第２モジュール）がＰＥＩモジュールＭ２１（第１モジュール）の共通ライブラリＳＬの関数を用いる例について説明する。
メモリ領域において、ＰＥＩモジュールＭ２２は共通インタフェースライブラリＳＩを呼び出し（ステップＤ１０：図１４の符号Ｐ３１参照）、固有のメモリ領域２０１の共通ライブラリへのアドレス情報２０４を取得する（ステップＤ２０：図１４の符号Ｐ３２参照）。ＰＥＩモジュールＭ２２は、この共通ライブラリへのアドレス情報２０４を用いて

、ＰＥＩモジュールＭ２１の共通ライブラリＳＬの関数を呼び出す（ステップＤ３０：図１４の符号Ｐ３３参照）。

【００７９】

このように、ＰＥＩフェーズ（メモリ領域）においても、ＰＥＩモジュールＭ２２は、共通インタフェースライブラリＳＩを用いて、固有のメモリ領域２０１の共通ライブラリへのアドレス情報２０４を介してＰＥＩモジュールＭ２１の共通ライブラリＳＬを読み出すことができる。又、ＰＥＩモジュールＭ２１も自身がそなえる共通ライブラリＳＬを用いることができる。

従って、ＰＥＩモジュールＭ２１の共通ライブラリＳＬを、ＰＥＩモジュールＭ２１とＰＥＩモジュールＭ２２とによって共用することが可能となる。

【００８０】

次に、本実施形態の一例としての情報処理装置１００におけるＰＥＩフェーズ（メモリ領域）での共通領域２０２に格納された共通領域情報の読み出し手法を、図１６を参照しながら、図１５に示すフローチャート（ステップＥ１０～Ｅ５０に従って説明する）。なお、図１６は本情報処理装置１００におけるＰＥＩフェーズ（メモリ領域）での共通領域２０２への参照経路を説明する図である。

【００８１】

メモリ領域において、ＰＥＩモジュールＭ２２（第２モジュール）は共通インタフェースライブラリＳＩを呼び出し（ステップＥ１０：図１６の符号Ｐ４１参照）、固有のメモリ領域２０１の共通ライブラリへのアドレス情報２０４を取得する（ステップＥ２０：図１６の符号Ｐ４２参照）。ＰＥＩモジュールＭ２２は、この共通ライブラリへのアドレス情報２０４を用いて、ＰＥＩモジュールＭ２１（第１モジュール）の共通ライブラリＳＬを呼び出す（ステップＥ３０：図１６の符号Ｐ４３参照）。

【００８２】

ＰＥＩモジュールＭ２２は、この共通ライブラリＳＬから、固有のメモリ領域２０１の共通領域へのアドレス情報２０３へアクセスするための情報（例えば、ポインタ）を取得し、この情報に基づいて固定のメモリ領域２０１の共通領域へのアドレス情報２０３を取得する（ステップＥ４０：図１６の符号Ｐ４４参照）。ＰＥＩモジュールＭ２２は、この共通領域へのアドレス情報２０３を用いて、固有のメモリ領域２０１の共通領域２０２にアクセスし（ステップＥ５０：図１６の符号Ｐ４５参照）、この共通領域２０２に格納されている共通領域情報を取得する。

【００８３】

このように、ＰＥＩフェーズ（メモリ領域）においても、ＰＥＩモジュールＭ２２は、共通インタフェースライブラリＳＩを用いて、固有のメモリ領域２０１の共通ライブラリへのアドレス情報２０４を介してＰＥＩモジュールＭ２１の共通ライブラリＳＬを読み出す。更に、この共通ライブラリＳＬを用いて固有のメモリ領域２０１の共通領域へのアドレス情報２０３を取得し、この共通領域へのアドレス情報２０３を用いて共通領域情報を取得することができる。

【００８４】

また、ＰＥＩモジュールＭ２１も自身がそなえる共通ライブラリＳＬを用いて、固有のメモリ領域２０１の共通領域へのアドレス情報２０３を取得し、この共通領域へのアドレス情報２０３を用いて共通領域情報を取得することができる。

従って、固有のメモリ領域２０１の共通領域２０２の共通領域情報を、ＰＥＩモジュールＭ２１とＰＥＩモジュールＭ２２とによって共用することが可能となる。

【００８５】

次に、本実施形態の一例としての情報処理装置１００におけるＤＸＥフェーズ（メモリ領域）での共通ライブラリＳＬの読み出し手法を、図１８を参照しながら、図１７に示すフローチャート（ステップＦ１０～Ｆ３０に従って説明する）。なお、図１８は本情報処理装置１００におけるＤＸＥフェーズ（メモリ領域）での共通ライブラリＳＬへの参照経路を説明する図である。

【００８６】
ＤＸＥフェーズ（メモリ領域）において、ＤＸＥドライバＤ２（第２モジュール）がＤＸＥドライバＤ１（第１モジュール）の共通ライブラリＳＬの関数を用いる例について説明する。
メモリ領域において、ＤＸＥドライバＤ２は共通インタフェースライブラリＳＩを呼び出し（ステップＦ１０：図１８の符号Ｐ５１参照）、固有のメモリ領域２０１の共通ライブラリへのアドレス情報２０４を取得する（ステップＦ２０：図１８の符号Ｐ５２参照）。ＤＸＥドライバＤ２は、この共通ライブラリへのアドレス情報２０４を用いて、ＤＸＥドライバＤ１の共通ライブラリＳＬの関数を呼び出す（ステップＦ３０：図１８の符号Ｐ５３参照）。
【００８７】
このように、ＤＸＥフェーズ（メモリ領域）においても、ＤＸＥドライバＤ２は、共通インタフェースライブラリＳＩを用いて、固有のメモリ領域２０１の共通ライブラリへのアドレス情報２０４を介してＤＸＥドライバＤ１の共通ライブラリＳＬを読み出すことができる。又、ＤＸＥドライバＤ１も自身がそなえる共通ライブラリＳＬを用いることができる。
従って、ＤＸＥドライバＤ１の共通ライブラリＳＬを、ＤＸＥドライバＤ１とＤＸＥドライバＤ２とによって共用することが可能となる。
【００８８】
次に、本実施形態の一例としての情報処理装置１００におけるＤＸＥフェーズ（メモリ領域）での共通領域２０２に格納された共通領域情報の読み出し手法を、図２０を参照しながら、図１９に示すフローチャート（ステップＧ１０～Ｇ５０に従って説明する）。なお、図２０は本情報処理装置１００におけるＤＸＥフェーズ（メモリ領域）での共通領域２０２への参照経路を説明する図である。
【００８９】
メモリ領域において、ＤＸＥドライバＤ２（第２モジュール）は共通インタフェースライブラリＳＩを呼び出し（ステップＧ１０：図２０の符号Ｐ６１参照）、固有のメモリ領域２０１の共通ライブラリへのアドレス情報２０４を取得する（ステップＧ２０：図２０の符号Ｐ６２参照）。ＤＸＥドライバＤ２は、この共通ライブラリへのアドレス情報２０４を用いて、ＤＸＥドライバＤ１（第１モジュール）の共通ライブラリＳＬを呼び出す（ステップＧ３０：図２０の符号Ｐ６３参照）。
【００９０】
ＤＸＥドライバＤ２は、この共通ライブラリＳＬから、固有のメモリ領域２０１の共通領域へのアドレス情報２０３へアクセスするための情報（例えば、ポインタ）を取得し、この情報に基づいて固定のメモリ領域２０１の共通領域へのアドレス情報２０３を取得する（ステップＧ４０：図２０の符号Ｐ６４参照）。ＤＸＥドライバＤ２は、この共通領域へのアドレス情報２０３を用いて、固有のメモリ領域２０１の共通領域２０２にアクセスし（ステップＧ５０：図２０の符号Ｐ６５参照）、この共通領域２０２に格納されている共通領域情報を取得する。
【００９１】
このように、ＤＸＥフェーズ（メモリ領域）においても、ＤＸＥドライバＤ２は、共通インタフェースライブラリＳＩを用いて、固有のメモリ領域２０１の共通ライブラリへのアドレス情報２０４を介してＤＸＥドライバＤ１の共通ライブラリＳＬを読み出す。更に、この共通ライブラリＳＬを用いて固有のメモリ領域２０１の共通領域へのアドレス情報２０３を取得し、この共通領域へのアドレス情報２０３を用いて共通領域情報を取得することができる。
【００９２】
また、ＤＸＥドライバＤ１も自身がそなえる共通ライブラリＳＬを用いて、固有のメモリ領域２０１の共通領域へのアドレス情報２０３を取得し、この共通領域へのアドレス情報２０３を用いて共通領域情報を取得することができる。

従って、固有のメモリ領域２０１の共通領域２０２の共通領域情報を、ＤＸＥドライバＤ１とＤＸＥドライバＤ２とによって共用することが可能となる。

【００９３】

次に、本実施形態の一例としての情報処理装置１００におけるＳＭＭフェーズ（ＰＥＩ，ＤＸＥ）での共通ライブラリＳＬの読み出し手法を、図２２を参照しながら、図２１に示すフローチャート（ステップＨ１０～Ｈ３０に従って説明する）。なお、図２２は本情報処理装置１００におけるＳＭＭフェーズ（ＰＥＩ）及びＳＭＭフェーズ（ＤＸＥ）での共通ライブラリＳＬへの参照経路を説明する図である。

【００９４】

ここでは、便宜上、ＳＭＭフェーズ（ＰＥＩ）において、ＰＥＩモジュールＭ２４（第２モジュール）がＰＥＩモジュールＭ２３（第１モジュール）の共通ライブラリＳＬの関数を用いる例と、ＳＭＭフェーズ（ＤＸＥ）において、ＤＸＥドライバＤ４（第２モジュール）がＤＸＥドライバＤ３（第１モジュール）の共通ライブラリＳＬの関数を用いる例とをまとめて説明する。

【００９５】

ＳＭＭ領域において、ＰＥＩモジュールＭ２４は共通インタフェースライブラリＳＩを呼び出し（ステップＨ１０：図２２の符号Ｐ８１参照）、ＳＭＭ固定のメモリ領域２１１の共通ライブラリへのアドレス情報２１２を取得する（ステップＨ２０：図２２の符号Ｐ８２参照）。ＰＥＩモジュールＭ２４は、この共通ライブラリへのアドレス情報２１２を用いて、ＰＥＩモジュールＭ２３の共通ライブラリＳＬの関数を呼び出す（ステップＨ３０：図２２の符号Ｐ８３参照）。

【００９６】

同様に、ＳＭＭ領域において、ＤＸＥドライバＤ４は共通インタフェースライブラリＳＩを呼び出し（ステップＨ１０：図２２の符号Ｐ７１参照）、ＳＭＭ固定のメモリ領域２１１の共通ライブラリへのアドレス情報２１２を取得する（ステップＨ２０：図２２の符号Ｐ７２参照）。ＤＸＥドライバＤ４は、この共通ライブラリへのアドレス情報２１２を用いて、ＤＸＥドライバＤ３の共通ライブラリＳＬの関数を呼び出す（ステップＨ３０：図２２の符号Ｐ７３参照）。

【００９７】

このように、ＳＭＭフェーズ（ＰＥＩ）及びＳＭＭフェーズ（ＤＸＥ）のいずれにおいても、ＰＥＩモジュールＭ２４やＤＸＥドライバＤ４は、共通インタフェースライブラリＳＩを用いて、固有のメモリ領域２１１の共通ライブラリへのアドレス情報２１２を介してＰＥＩモジュールＭ２３やＤＸＥドライバＤ３の共通ライブラリＳＬを読み出すことができる。

【００９８】

また、ＰＥＩモジュールＭ２３やＤＸＥドライバＤ３も自身がそなえる共通ライブラリＳＬを用いることができる。

従って、ＰＥＩモジュールＭ２３の共通ライブラリＳＬを、ＰＥＩモジュールＭ２３とＰＥＩモジュールＭ２４とによって共用することが可能となり、又、ＤＸＥドライバＤ３の共通ライブラリＳＬを、ＤＸＥドライバＤ３とＤＸＥドライバＤ４とによって共用することが可能となる。

【００９９】

次に、本実施形態の一例としての情報処理装置１００におけるＳＭＭフェーズ（ＰＥＩ，ＤＸＥ）での共通領域２０２に格納された共通領域情報の読み出し手法を、図２４及び図２５を参照しながら、図２３に示すフローチャート（ステップＪ１０～Ｊ５０に従って説明する）。なお、図２４は本情報処理装置１００におけるＳＭＭフェーズ（ＰＥＩ）での共通領域２０２への参照経路を説明する図、図２５は本情報処理装置１００におけるＳＭＭフェーズ（ＤＸＥ）での共通領域２０２への参照経路を説明する図である。

【０１００】

ＳＭＭ領域において、ＰＥＩモジュールＭ２４（第２モジュール）やＤＸＥドライバＤ

４（第２モジュール）はそれぞれ共通インタフェースライブラリＳＩを呼び出し（ステップＪ１０：図２４の符号Ｐ９１及び図２５の符号Ｐ１０１参照）、ＳＭＭ固有のメモリ領域２１１の共通ライブラリへのアドレス情報２１２を取得する（ステップＪ２０：図２４の符号Ｐ９２及び図２５の符号Ｐ１０２参照）。

【０１０１】

ＰＥＩモジュールＭ２４やＤＸＥドライバＤ４は、この共通ライブラリへのアドレス情報２１２を用いて、ＰＥＩモジュールＭ２３（第１モジュール）やＤＸＥドライバＤ３（第１モジュール）の共通ライブラリＳＬを呼び出す（ステップＪ３０：図２４の符号Ｐ９３及び図２５の符号Ｐ１０３参照）。

ＰＥＩモジュールＭ２４やＤＸＥドライバＤ４は、この共通ライブラリＳＬから、固有のメモリ領域２０１の共通領域へのアドレス情報２０３へアクセスするための情報（例えば、ポインタ）を取得し、この情報に基づいて固定のメモリ領域２０１の共通領域へのアドレス情報２０３を取得する（ステップＪ４０：図２４の符号Ｐ９４及び図２５の符号Ｐ１０４参照）。ＰＥＩモジュールＭ２４やＤＸＥドライバＤ４は、この共通領域へのアドレス情報２０３を用いて、固有のメモリ領域２０１の共通領域２０２にアクセスし（ステップＪ５０：図２４の符号Ｐ９５及び図２５の符号Ｐ１０５参照）、この共通領域２０２に格納されている共通領域情報を取得する。

【０１０２】

このように、ＳＭＭフェーズ（ＰＥＩ）やＳＭＭフェーズ（ＤＸＥ）においても、ＰＥＩモジュールＭ２４やＤＸＥドライバＤ４は、共通インタフェースライブラリＳＩを用いて、固有のメモリ領域２０１の共通ライブラリへのアドレス情報２０４を介してＰＥＩモジュールＭ２３やＤＸＥドライバＤ３の共通ライブラリＳＬを読み出す。更に、この共通ライブラリＳＬを用いて固有のメモリ領域２０１の共通領域へのアドレス情報２０３を取得し、この共通領域へのアドレス情報１０３を用いて共通領域情報を取得することができる。

【０１０３】

また、ＰＥＩモジュールＭ２３やＤＸＥドライバＤ３も、各自がそなえる共通ライブラリＳＬを用いて、固有のメモリ領域２０１の共通領域へのアドレス情報２０３を取得し、この共通領域へのアドレス情報２０３を用いて、それぞれ共通領域情報を取得することができる。

従って、固有のメモリ領域２０１の共通領域２０２の共通領域情報を、ＰＥＩモジュールＭ２３とＰＥＩモジュールＭ２４とや、ＤＸＥドライバＤ３とＤＸＥドライバＤ４とで共有することが可能となる。

【０１０４】

なお、本実施形態においては、ＰＥＩモジュールＭ１２，Ｍ２２，Ｍ２４やＤＸＥドライバＤ２，Ｄ４が、共通インタフェースライブラリＳＩを用いて、共通ライブラリＳＬや共通領域情報を取得している。すなわち、これらのＰＥＩモジュールＭ１２，Ｍ２２，Ｍ２４やＤＸＥドライバＤ２，Ｄ４を実行することにより、前述した情報取得部１１５としての機能が実現されるといえる。

【０１０５】

このように、実施形態の一例としての情報処理装置によれば、ＰＥＩモジュールＭ１１が自身がそなえる共通ライブラリＳＬを用いることができる。その一方で、ＰＥＩモジュールＭ１２が、共通インタフェースライブラリＳＩ及び共通ライブラリへのアドレス情報１０４を介して、ＰＥＩモジュールＭ１１の共通ライブラリＳＬを用いることができる。

同様にして、ＰＥＩモジュールＭ２１，Ｍ２３やＤＸＥドライバＤ１，Ｄ３が、それぞれがそなえる共通ライブラリＳＬを使用できる。その一方で、ＰＥＩモジュールＭ２２がＰＥＩモジュールＭ２１の共通ライブラリＳＬを、又、ＰＥＩモジュールＭ２４がＰＥＩモジュールＭ２３の共通ライブラリＳＬを用いることができる。更に、ＤＸＥドライバＤ２がＤＸＥドライバＤ３の共通ライブラリＳＬを、又、ＤＸＥドライバＤ４がＤＸＥドライバＤ３の共通ライブラリＳＬをそれぞれ用いることができる。

【０１０６】
すなわち、共通インタフェースライブラリＳＩをそなえるモジュールが、他のモジュールにそなえられた共通ライブラリＳＬを使用することができ、共通ライブラリＳＬを共用することができる。これにより、各モジュールやプログラムデータＰのサイズを小さくすることができ、モジュールがロードされるプロセッサキャッシュやＲＡＭ２０のサイズや、プログラムデータＰを格納するＢＩＯＳ　Ｆｌａｓｈ２２の容量を小さく構成することができる。従って、処理速度の向上や製造コストを低減することができる。
【０１０７】
また、ＰＥＩモジュールＭ１１が自身がそなえる共通ライブラリＳＬを用いて共通領域１０２の共通領域情報を用いることができる。その一方で、ＰＥＩモジュールＭ１２が、共通インタフェースライブラリＳＩ，共通ライブラリＳＬ及び共通領域へのアドレス情報１０３を用いて共通領域１０２の共通領域情報を用いることができる。
同様にして、ＰＥＩモジュールＭ２１，Ｍ２３やＤＸＥドライバＤ１，Ｄ３が、それぞれがそなえる共通ライブラリＳＬを用いて、固定のメモリ領域２０１の共通領域へのアドレス情報２０３を介して共通領域２０２の共通領域情報を使用できる。その一方で、ＰＥＩモジュールＭ２２がＰＥＩモジュールＭ２１の共通ライブラリＳＬを、又、ＰＥＩモジュールＭ２４がＰＥＩモジュールＭ２３の共通ライブラリＳＬを用いて、固定のメモリ領域２０１の共通領域へのアドレス情報２０３を介して共通領域２０２の共通領域情報を使用できる。更に、ＤＸＥドライバＤ２がＤＸＥドライバｄ３の共通ライブラリＳＬを、又、ＤＸＥドライバＤ４がＤＸＥドライバＤ３の共通ライブラリＳＬをそれぞれ用いて、固定のメモリ領域２０１の共通領域へのアドレス情報２０３を介して共通領域２０２の共通領域情報を使用できる。
【０１０８】
従って、複数のモジュールが共通領域１０２や共通領域２０２の共通領域情報を使用することができ、これらの共通領域情報を共用することができる。これによっても、各モジュールやプログラムデータＰのサイズを小さくすることができ、モジュールがロードされるプロセッサキャッシュやＲＡＭ２０のサイズや、プログラムデータＰを格納するＢＩＯＳ　Ｆｌａｓｈ２２の容量を小さく構成することができる。従って、処理速度の向上や製造コストを低減することができる。
【０１０９】
また、固定のメモリ領域２０１に共通領域情報を格納する共通領域２０２をそなえ、ＰＥＩモジュールＭ２１～Ｍ２４やＤＸＥドライバＤ１～Ｄ４が、それぞれ、この共通領域２０２の共通領域情報を使用できる。これにより、ＰＥＩフェーズとＤＸＥフェーズという異なるフェーズのモジュールで共通領域情報を共用することができ、処理速度の向上や製造コストを低減することができる他、利便性が向上する。
【０１１０】
また、共通領域設定部１１４が、ＣＡＲ領域１０１の共通領域１０２に格納されている共通領域情報を固有のメモリ領域２０１の共通領域２０２にコピーすることにより、ＰＥＩフェーズ（ＣＡＲ領域）において使用されていた共通領域情報を、ＰＥＩフェーズ（メモリ領域）やＤＸＥフェーズやＳＭＭフェーズ（ＰＥＩ），ＳＭＭフェーズ（ＤＸＥ）においても使用（共用）することができ、利便性が向上する。
【０１１１】
そして、開示の技術は上述した実施形態に限定されるものではなく、本実施形態の趣旨を逸脱しない範囲で種々変形して実施することができる。
例えば、上述した実施形態においては、ＰＥＩモジュールＭ１１，Ｍ２１，Ｍ２３やＤＸＥドライバＤ１，Ｄ３に共通ライブラリＳＬをそなえ、ＰＥＩモジュールＭ１２，Ｍ２２，Ｍ２４やＤＸＥドライバＤ２，Ｄ４に共通インタフェースライブラリＳＩをそなえているが、これに限定されるものではない。すなわち、複数のモジュールのうち、どのモジュールに共通ライブラリＳＬをそなえてもよく、又、同様に、どのモジュールに共通インタフェースライブラリＳＩをそなえてもよい。又、同一のモジュールに共通ライブラリＳ

Ｌと他のモジュールの共通ライブラリＳＬを参照するための共通インタフェースライブラリＳＩとの両方をそなえてもよく、種々変形して実施することができる。

【０１１２】

また、上述した実施形態においては、ＰＥＩフェーズ（ＣＡＲ領域），ＰＥＩフェーズ（メモリ領域），ＳＭＭフェーズ（ＰＥＩ）及びＳＭＭフェーズ（ＤＸＥ）の各フェーズにおいて、それぞれ２つのモジュールで共通ライブラリＳＬや共通領域情報を共有する例を示しているが、これに限定されるものではない。すなわち、３以上のモジュールで共通ライブラリＳＬや共通領域情報を共有してもよく、各モジュールの数は適宜変更して実施することができる。

【０１１３】

さらに、上述した実施形態においては、プロセッサとしてＣＰＵ１０をそなえ、このＣＰＵ１０において起動処理部１１としての各種機能を実現しているが、これに限定されるものではなく、例えば、ＭＰＵ（Micro-Processing Unit）等、他のプロセッサを用いてもよい。

また、上述した開示により本実施形態を当業者によって実施・製造することが可能である。

【０１１４】

以上の実施形態に関し、更に以下の付記を開示する。

（付記１）

プロセッサをそなえた情報処理装置の起動処理方法であって、

該プロセッサにより実行されることにより当該情報処理装置の起動プロセスの一部を実現する複数のモジュールのうち、２以上のモジュールにより共通に用いられる第１共通情報をそなえる第１モジュールをロードするステップと、

該情報処理装置にそなえられた記憶領域に、該第１共通情報へアクセスするためのアクセス情報を格納するステップと、

該アクセス情報にアクセスするためのインタフェース情報をそなえる第２モジュールをロードするステップと、

該第２モジュールにより、該インタフェース情報によって該アクセス情報にアクセスし、該アクセス情報を介して該第１共通情報を取得するステップとをそなえることを特徴とする、起動処理方法。

【０１１５】

（付記２）

該記憶領域に、２以上のモジュールにより共通に用いられる第２共通情報を格納する共通領域を形成するステップと、

該記憶領域に、該共通領域へアクセスするための共通領域アクセス情報を格納するステップと、

該第２モジュールにより、該インタフェース情報によって該アクセス情報にアクセスし、アクセスした当該アクセス情報を用いて該第１共通情報にアクセスし、アクセスした当該第１共通情報に基づいて該共通領域アクセス情報にアクセスし、アクセスした当該共通領域アクセス情報を用いて、該共通領域に格納された該第２共通情報を取得するステップとをそなえることを特徴とする、付記１記載の起動処理方法。

【０１１６】

（付記３）

該起動処理が複数のフェーズをそなえ、

該複数のフェーズのうち一のフェーズにおいて、該プロセッサのプロセッサキャッシュを該記憶領域として用い、

該情報処理装置の主記憶装置が使用可能な状態となった後に、該プロセッサキャッシュに格納されている該第２共通情報を該主記憶装置に複写するステップをそなえることを特徴とする、付記２記載の起動処理方法。

【０１１７】

（付記４）
プロセッサをそなえた情報処理装置であって、
起動処理時に、該プロセッサにより実行されることにより当該情報処理装置の起動プロセスの一部を実現する複数のモジュールのうち、２以上のモジュールにより共通に用いられる第１共通情報をそなえる第１モジュールをロードする第１モジュール設定部と、
起動処理時に、該情報処理装置にそなえられた記憶領域に、該第１共通情報へアクセスするためのアクセス情報を格納するアクセス情報設定部と、
起動処理時に、該アクセス情報にアクセスするためのインタフェース情報をそなえる第２モジュールをロードする第２モジュール設定部と、
該起動処理時に、該第２モジュールにより、該インタフェース情報によって該アクセス情報にアクセスし、該アクセス情報を介して該第１共通情報を取得する第１情報取得部とをそなえることを特徴とする、情報処理装置。

【０１１８】
（付記５）
該起動処理時に、該記憶領域に、２以上のモジュールにより共通に用いられる第２共通情報を格納する共通領域を形成する共通領域設定部と、
該起動処理時に、該記憶領域に、該共通領域へアクセスするための共通領域アクセス情報を格納する共通領域アクセス情報設定部と、
該起動処理時に、該第２モジュールにより、該インタフェース情報によって該アクセス情報にアクセスし、アクセスした当該アクセス情報を用いて該第１共通情報にアクセスし、アクセスした当該第１共通情報に基づいて該共通領域アクセス情報にアクセスし、アクセスした当該共通領域アクセス情報を用いて、該共通領域に格納された該第２共通情報を取得する第２情報取得部とをそなえることを特徴とする、付記４記載の情報処理装置。

【０１１９】
（付記６）
該起動処理が複数のフェーズをそなえ、
該複数のフェーズのうち一のフェーズにおいて、該プロセッサのプロセッサキャッシュを該記憶領域として用い、
該共通領域設定部が、該情報処理装置の主記憶装置が使用可能な状態となった後に、該プロセッサキャッシュに格納されている該第２共通情報を該主記憶装置に複写することを特徴とする、付記５記載の情報処理装置。

【０１２０】
（付記７）
起動処理をコンピュータに実行させるための起動処理プログラムであって、
該コンピュータに実行されることにより情報処理装置の起動プロセスの一部を実現する複数のモジュールのうち、２以上のモジュールにより共通に用いられる第１共通情報をそなえる第１モジュールをロードするステップと、
該情報処理装置にそなえられた記憶領域に、該第１共通情報へアクセスするためのアクセス情報を格納するステップと、
該アクセス情報にアクセスするためのインタフェース情報をそなえる第２モジュールをロードするステップと、
該第２モジュールにより、該インタフェース情報によって該アクセス情報にアクセスし、該アクセス情報を介して該第１共通情報を取得するステップとを、該コンピュータに実行させることを特徴とする、起動処理プログラム。

【０１２１】
（付記８）
該記憶領域に、２以上のモジュールにより共通に用いられる第２共通情報を格納する共通領域を形成するステップと、
該記憶領域に、該共通領域へアクセスするための共通領域アクセス情報を格納するステップと、

該第２モジュールにより、該インタフェース情報によって該アクセス情報にアクセスし、アクセスした当該アクセス情報を用いて該第１共通情報にアクセスし、アクセスした当該第１共通情報に基づいて該共通領域アクセス情報にアクセスし、アクセスした当該共通領域アクセス情報を用いて、該共通領域に格納された該第２共通情報を取得するステップとを、該コンピュータに実行させることを特徴とする、付記７記載の起動処理プログラム。

【０１２２】

（付記９）

該起動処理が複数のフェーズをそなえ、

該複数のフェーズのうち一のフェーズにおいて、該プロセッサのプロセッサキャッシュを該記憶領域として用い、

該情報処理装置の主記憶装置が使用可能な状態となった後に、該プロセッサキャッシュに格納されている該第２共通情報を該主記憶装置に複写するステップを該コンピュータに実行させることを特徴とする、付記８記載の起動処理プログラム。

【０１２３】

（付記１０）

起動処理をコンピュータに実行させるための起動処理プログラムを記録したコンピュータ読取可能な記録媒体であって、

該起動処理プログラムが、

該コンピュータに実行されることにより情報処理装置の起動プロセスの一部を実現する複数のモジュールのうち、２以上のモジュールにより共通に用いられる第１共通情報をそなえる第１モジュールをロードするステップと、

該情報処理装置にそなえられた記憶領域に、該第１共通情報へアクセスするためのアクセス情報を格納するステップと、

該アクセス情報にアクセスするためのインタフェース情報をそなえる第２モジュールをロードするステップと、

該第２モジュールにより、該インタフェース情報によって該アクセス情報にアクセスし、該アクセス情報を介して該第１共通情報を取得するステップとを、該コンピュータに実行させることを特徴とする、起動処理プログラムを記録したコンピュータ読取可能な記録媒体。

【０１２４】

（付記１１）

該起動処理プログラムが、

該記憶領域に、２以上のモジュールにより共通に用いられる第２共通情報を格納する共通領域を形成するステップと、

該記憶領域に、該共通領域へアクセスするための共通領域アクセス情報を格納するステップと、

該第２モジュールにより、該インタフェース情報によって該アクセス情報にアクセスし、アクセスした当該アクセス情報を用いて該第１共通情報にアクセスし、アクセスした当該第１共通情報に基づいて該共通領域アクセス情報にアクセスし、アクセスした当該共通領域アクセス情報を用いて、該共通領域に格納された該第２共通情報を取得するステップとを、該コンピュータに実行させることを特徴とする、付記１０記載の起動処理プログラムを記録したコンピュータ読取可能な記録媒体。

【０１２５】

（付記１２）

該起動処理が複数のフェーズをそなえ、

該起動処理プログラムが、

該複数のフェーズのうち一のフェーズにおいて、プロセッサキャッシュを該記憶領域として用い、

該情報処理装置の主記憶装置が使用可能な状態となった後に、該プロセッサキャッシュ

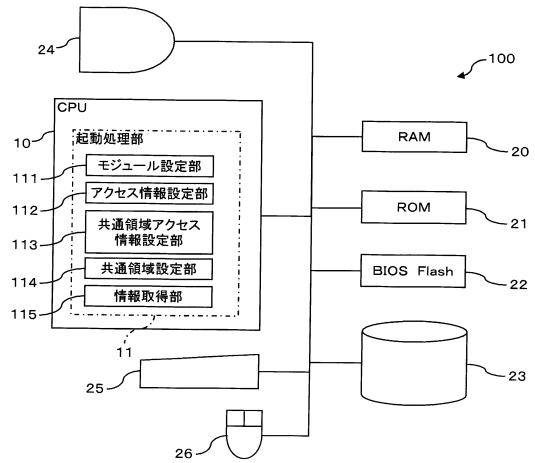
に格納されている該第２共通情報を該主記憶装置に複写するステップを該コンピュータに実行させることを特徴とする、付記１１記載の起動処理プログラムを記録したコンピュータ読取可能な記録媒体。

【符号の説明】

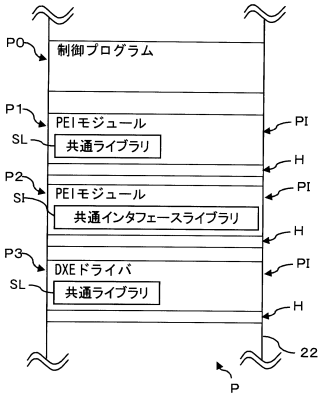
【０１２６】

１０　ＣＰＵ（プロセッサ）
１１　起動処理部
２０　ＲＡＭ
２１　ＲＯＭ
２２　ＢＩＯＳ　Ｆｌａｓｈ
２３　ストレージ
２４　ディスプレイ
１００　情報処理装置
１０１　ＣＡＲ領域（記憶領域）
１０２　共通領域
１０３，２０３　共通領域へのアドレス情報（共通領域アクセス情報）
１０４，２０４，２１２　共通ライブラリへのアドレス情報（アクセス情報）
１１１　モジュール設定部（第１モジュール設定部，第２モジュール設定部）
１１２　アクセス情報設定部
１１３　共通領域アクセス情報設定部
１１４　共通領域設定部
１１５　情報取得部（第１情報取得部，第２情報取得部）
２０１　固定のメモリ領域（記憶領域）
２１１　ＳＭＭ固定のメモリ領域
３０１，３１１　ＳＭＭ領域
１００１，２００１，２１０１　管理域
Ｄ，Ｄ１～Ｄ４　ＤＥＸドライバ（モジュール）
Ｈ　ヘッダ
Ｍ，Ｍ１１，Ｍ１２，Ｍ２１～Ｍ２４　ＰＥＩモジュール（モジュール）
Ｐ　プログラムデータ
Ｐ１～Ｐ３　モジュールプログラム
ＰＩ　実行可能イメージ
ＳＬ　共通ライブラリ（第１共通情報）
ＳＩ　共通インタフェースライブラリ（インタフェース情報）

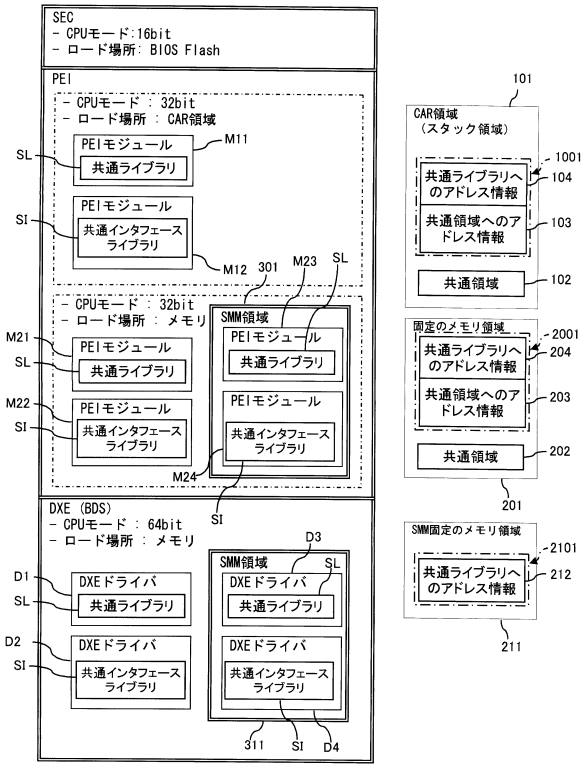
【図１】


24
100
CPU
10
起動処理部
111
モジュール設定部
112
アクセス情報設定部
113
共通領域アクセス情報設定部
114
共通領域設定部
115
情報取得部
11
RAM
20
ROM
21
BIOS Flash
22
23
25
26

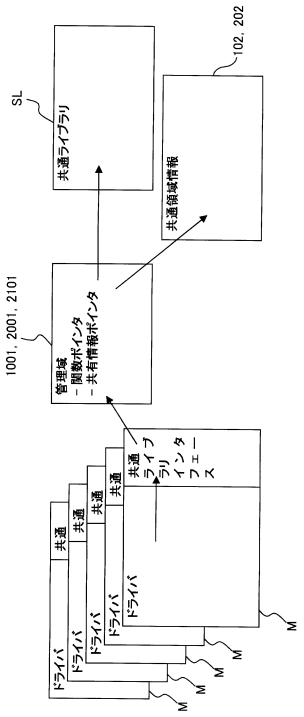
【図２】


P0
制御プログラム
P1
PEIモジュール
SL
共通ライブラリ
PI
H
P2
SI
PEIモジュール
共通インタフェースライブラリ
P3
DXEドライバ
22
P

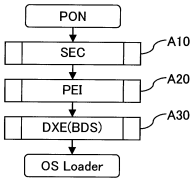
【図３】


SEC
- CPUモード:16bit
- ロード場所: BIOS Flash
PEI
- CPUモード : 32bit
- ロード場所 : CAR領域
PEIモジュール
共通ライブラリ
M11
SL
SI
共通インタフェースライブラリ
M12
301
M23
- CPUモード : 32bit
- ロード場所 : メモリ
SMM領域
M21
M22
M24
DXE (BDS)
- CPUモード : 64bit
- ロード場所 : メモリ
D1
D2
D3
D4
DXEドライバ
311
101
CAR領域
（スタック領域）
1001
104
共通ライブラリへのアドレス情報
103
共通領域へのアドレス情報
102
共通領域
固定のメモリ領域
2001
204
203
202
201
SMM固定のメモリ領域
2101
212
211

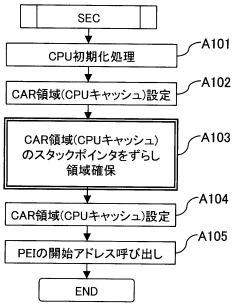
【図４】


SL
共通ライブラリ
102, 202
共通領域情報
1001, 2001, 2101
管理域
- 関数ポインタ
- 共有情報ポインタ
共通ライブラリインターフェース
共通
ドライバ
M

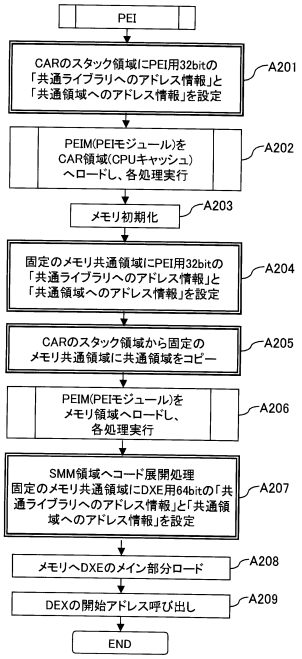
【図５】



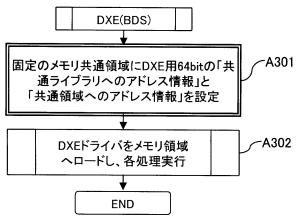
【図６】



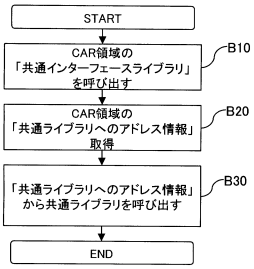
【図７】



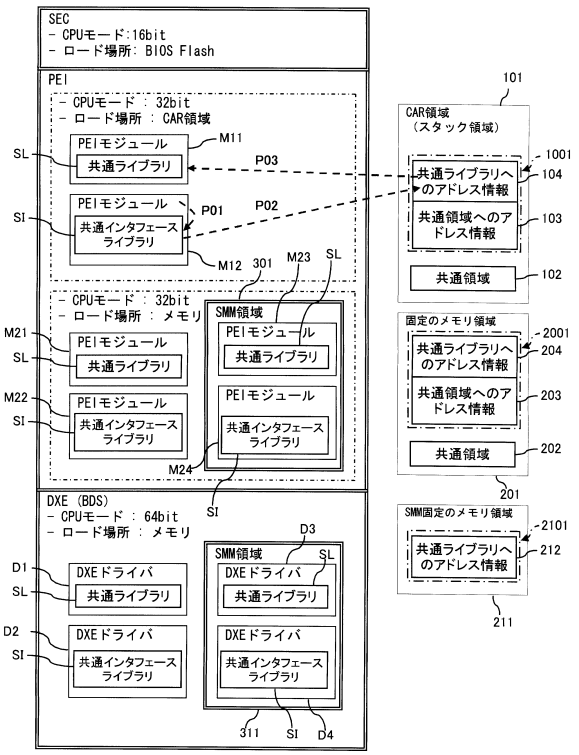
【図８】



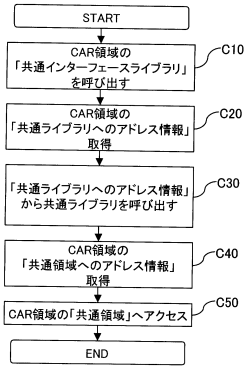
【図９】



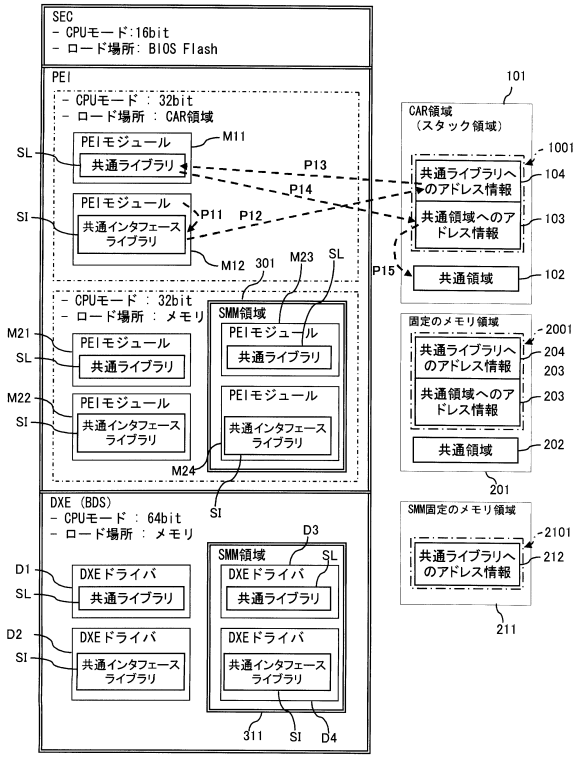
【図１０】



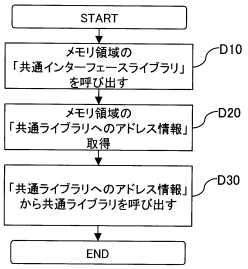
【図１１】



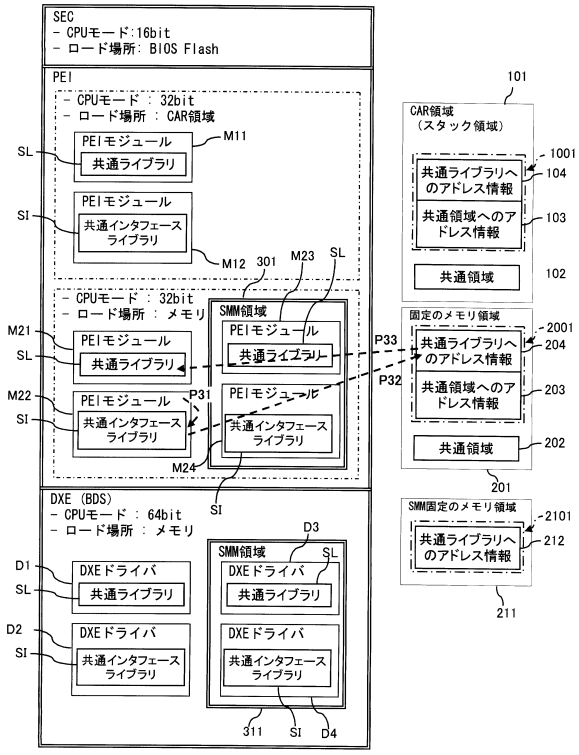
【図１２】



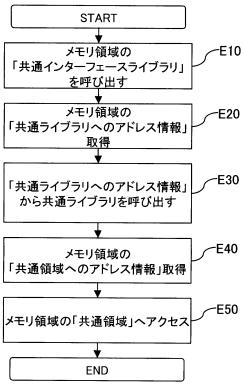
【図１３】



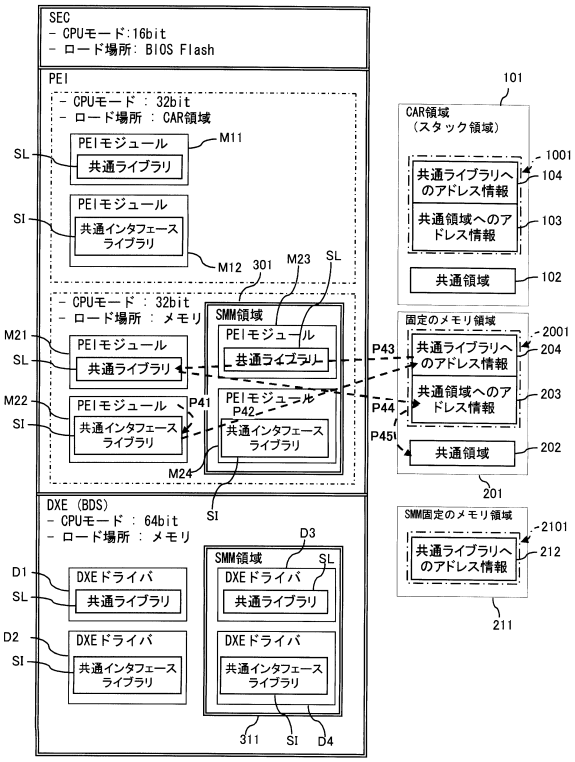
【図１４】



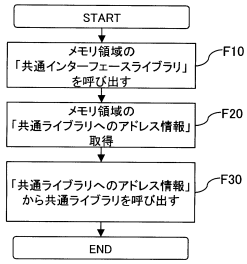
【図１５】



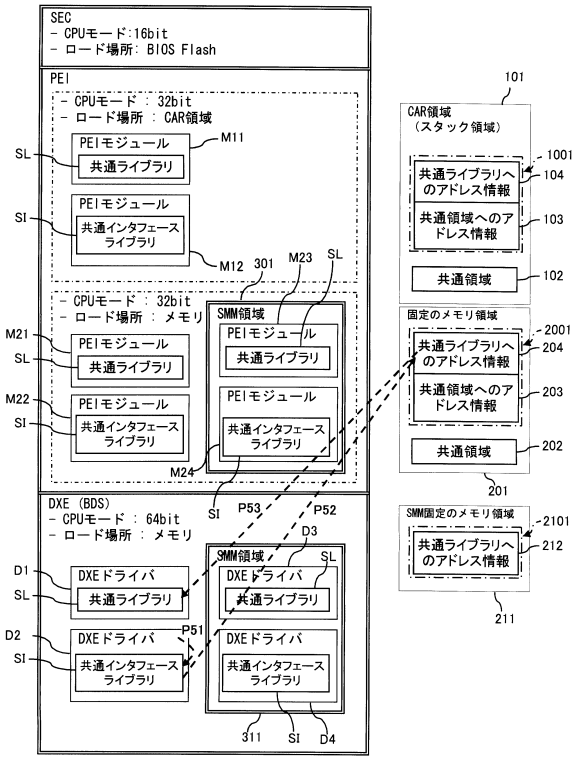
【図１６】



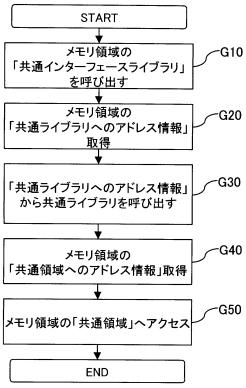
【図１７】



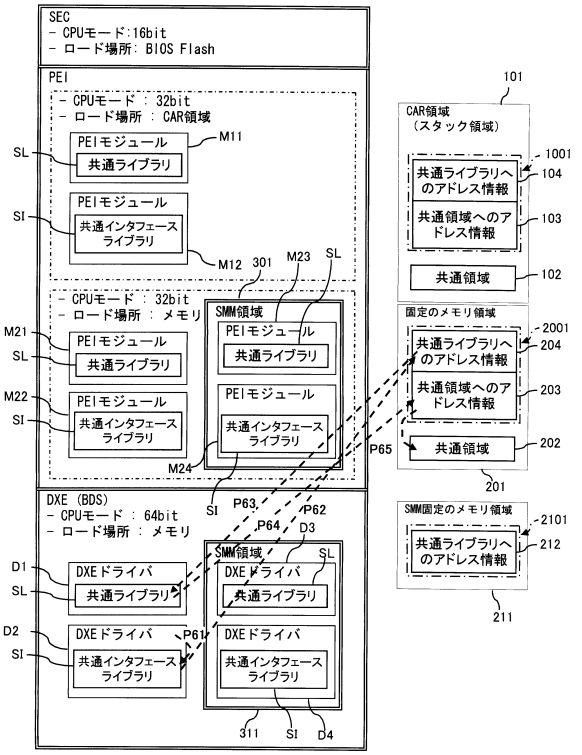
【図１８】



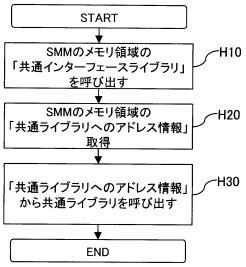
【図１９】



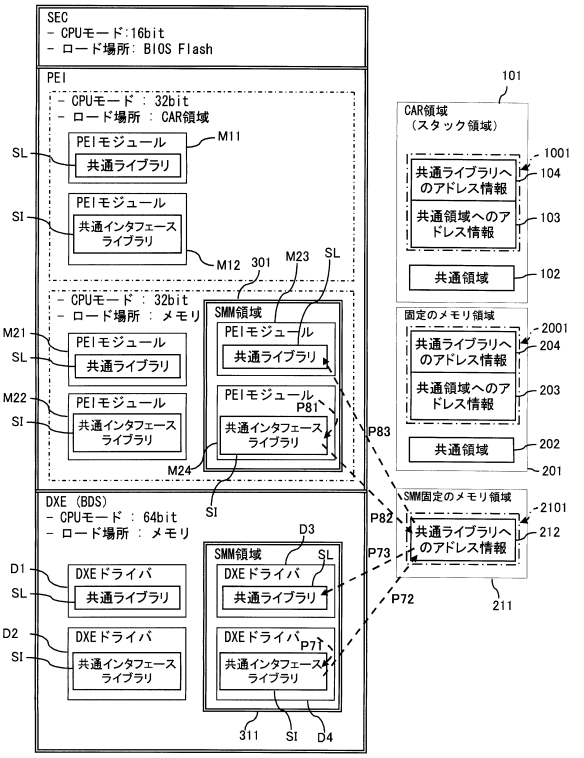
【図２０】



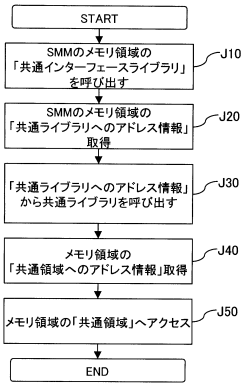
【図２１】



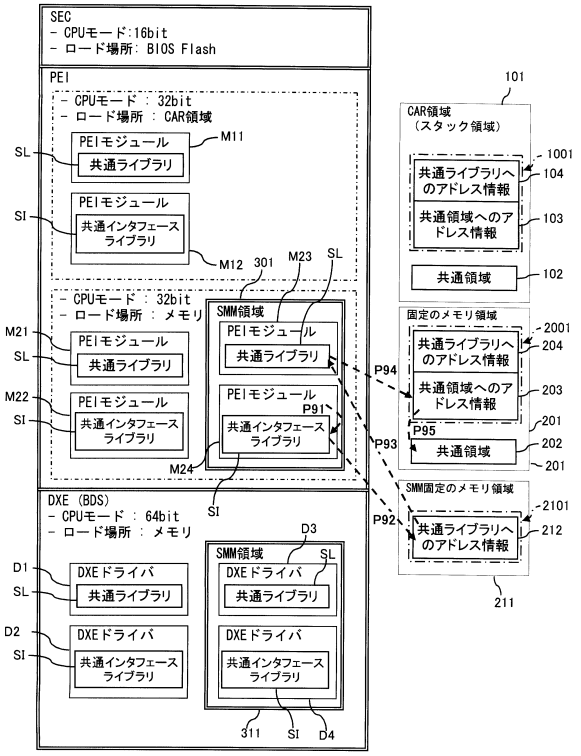
【図２２】



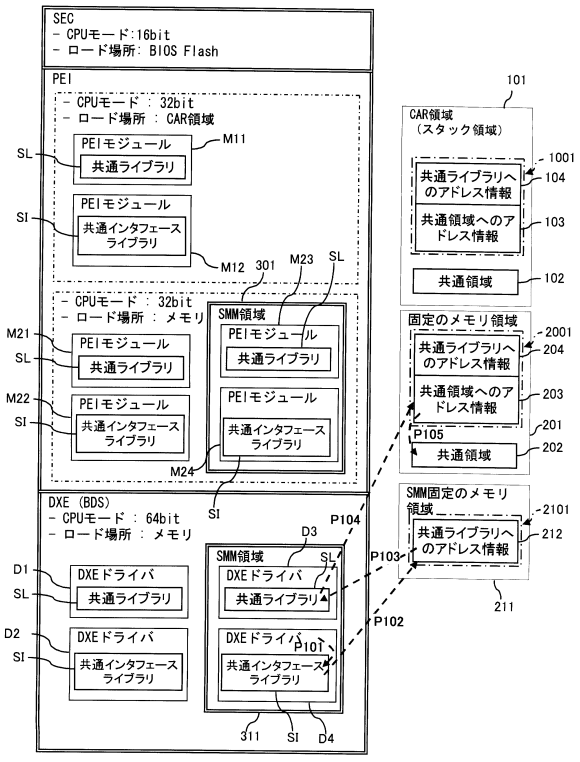
【図２３】



【図２４】



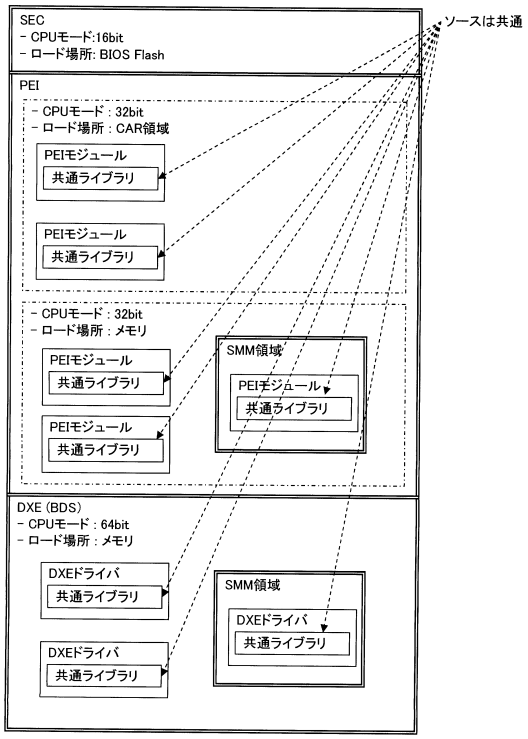
【図２５】



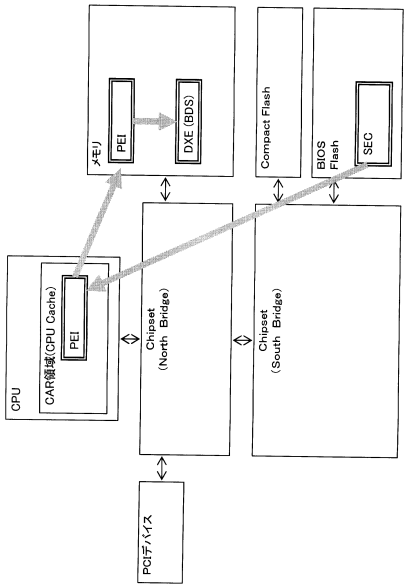
【図２６】

EFI Phase	プログラムロード動作領域	CPUモード	開発言語
SEC	BIOS Flash	16bit	アセンブラ
PEI	CAR領域(CPU Cache)	32bit	アセンブラ
	メモリ	32bit	C言語
	メモリ(SMM領域)	SMM 32bit	C言語
DXE (BDS)	メモリ	64bit	C言語
	メモリ(SMM領域)	SMM 64bit	C言語

【図２７】


SEC
- CPUモード:16bit
- ロード場所: BIOS Flash
ソースは共通
PEI
- CPUモード : 32bit
- ロード場所 : CAR領域
PEIモジュール
共通ライブラリ
PEIモジュール
共通ライブラリ
- CPUモード : 32bit
- ロード場所 : メモリ
PEIモジュール
共通ライブラリ
PEIモジュール
共通ライブラリ
SMM領域
PEIモジュール
共通ライブラリ
DXE (BDS)
- CPUモード : 64bit
- ロード場所 : メモリ
DXEドライバ
共通ライブラリ
DXEドライバ
共通ライブラリ
SMM領域
DXEドライバ
共通ライブラリ

【図２８】


メモリ
PEI
DXE (BDS)
Compact Flash
BIOS Flash
SEC
CPU
CAR領域(CPU Cache)
PEI
Chipset (North Bridge)
Chipset (South Bridge)
PCIデバイス

フロントページの続き

(56)参考文献 特開２００５－１９６２８６（ＪＰ，Ａ）
特開平０６－３３２６７５（ＪＰ，Ａ）
特開２００６－１３９７９５（ＪＰ，Ａ）
米国特許第０７５１２７１９（ＵＳ，Ｂ１）
米国特許出願公開第２０１１／０２８３０９８（ＵＳ，Ａ１）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　　９／４４５
Ｇ０６Ｆ　　９／５０
Ｇ０６Ｆ　　９／５４