

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5373817号

(P5373817)

(45) 発行日 平成25年12月18日(2013.12.18)

(24) 登録日 平成25年9月27日(2013.9.27)

(51) Int.Cl.

F I

G O 6 F 12/14 (2006.01)

G O 6 F 12/10 (2006.01)

G O 6 F 12/14 5 1 O E

G O 6 F 12/10 5 5 1 C

G O 6 F 12/10 5 0 5 Z

G O 6 F 12/10 5 4 1

請求項の数 12 (全 58 頁)

(21) 出願番号 特願2010-541763 (P2010-541763)  
 (86) (22) 出願日 平成21年1月5日(2009.1.5)  
 (65) 公表番号 特表2011-509470 (P2011-509470A)  
 (43) 公表日 平成23年3月24日(2011.3.24)  
 (86) 国際出願番号 PCT/EP2009/050050  
 (87) 国際公開番号 W02009/087133  
 (87) 国際公開日 平成21年7月16日(2009.7.16)  
 審査請求日 平成23年10月17日(2011.10.17)  
 (31) 優先権主張番号 11/972,688  
 (32) 優先日 平成20年1月11日(2008.1.11)  
 (33) 優先権主張国 米国 (US)

(73) 特許権者 390009531  
 インターナショナル・ビジネス・マシー  
 ズ・コーポレーション  
 I N T E R N A T I O N A L B U S I N  
 E S S M A C H I N E S C O R P O R  
 A T I O N  
 アメリカ合衆国10504 ニューヨーク  
 州 アーモンク ニュー オーチャード  
 ロード  
 (74) 代理人 100108501  
 弁理士 上野 剛史  
 (74) 代理人 100112690  
 弁理士 太佐 種一  
 (74) 代理人 100091568  
 弁理士 市位 嘉宏

最終頁に続く

(54) 【発明の名称】 フェッチ保護を有する動的アドレス変換

(57) 【特許請求の範囲】

【請求項 1】

仮想アドレスを主ストレージ内のデータ・ブロックの変換アドレスに変換するために用いられる変換テーブル階層構造を有するコンピュータ・システムにおけるデータを保護するための方法であって、前記変換アドレスへの変換を行う動的アドレス変換部が、

変換すべき前記仮想アドレスを取得するステップと、

前記変換テーブル階層構造の変換テーブルの初期基点アドレスを取得するステップであって、前記変換テーブル階層構造がセグメント・テーブルを含むステップと、

前記取得された初期基点に基づいて、前記セグメント・テーブルからセグメント・テーブル・エントリを取得するステップであって、前記セグメント・テーブル・エントリがフォーマット制御フィールドを含むように構成されているステップと、

強化型 D A T ファシリティがイネーブルであることに応答して、前記セグメント・テーブル・エントリ内の前記フォーマット制御フィールドがイネーブルであるかどうかを判定するステップと、

前記フォーマット制御フィールドがイネーブルであることに応答して、

前記セグメント・テーブル・エントリ内のアクセス妥当性フィールドがイネーブルであるかどうかを判定するステップであって、前記セグメント・テーブル・エントリが、前記アクセス妥当性フィールド、セグメント・アクセス制御フィールド、セグメント・フェッチ保護フィールド、及び主ストレージ内の大データ・ブロックのセグメント・フレームの絶対アドレスを含むステップ

10

20

を実施するステップと、

前記アクセス妥当性フィールドがイネーブルであることに応答して、

所望のメモリ・ブロックへのストアを許可するステップであって、前記ストアは前記仮想アドレスに関連付けられており、前記ストアが、前記セグメント・アクセス制御フィールドが、プログラムによって与えられるプログラム・ステータス・ワード又は実行中のプログラム命令のオペランドのうちのいずれか1つのプログラム・アクセス・キーに一致したことに応答してのみ許可されるステップと、

前記所望のデータ・ブロックからのフェッチを許可するステップであって、前記フェッチは前記仮想アドレスに関連付けられており、a) 前記セグメント・フェッチ保護フィールドがディスエーブルであること、又はb) 前記セグメント・フェッチ保護フィールドがイネーブルであり、且つ前記仮想アドレスに関連付けられた前記プログラム・アクセス・キーが前記セグメント・アクセス制御フィールドと等しいこと、のいずれかに応答してのみ許可されるステップを実施するステップと

を実施するステップと、

強化型DATファシリティがイネーブルでないことに応答して、

前記セグメント・テーブル・エントリからページ・テーブルの基点を取得するステップであって、前記仮想アドレスのページ・インデックス部分を用いて前記ページ・テーブル内のページ・テーブル・エントリを参照するステップと、

前記ページ・テーブル・エントリから、主ストレージ内の小データ・ブロックのページ・フレーム実アドレスを取得するステップであって、前記小データ・ブロックは前記大データ・ブロックよりもサイズが小さく、且つ、ストレージ・キーに関連付けられた4キロバイトのデータ・ブロックから成り、前記ストレージ・キーは、ストレージ・アクセス制御フィールドとストレージ・フェッチ保護フィールドとを含み、主ストレージ内の所望の実データ・ブロックの変換アドレスは、前記ページ・フレーム実アドレスと前記仮想アドレスのバイト・インデックス部分との組合せを含む、ステップと、

を実施するステップと

を含む方法。

#### 【請求項2】

前記変換アドレスが、実アドレス又は絶対アドレスのうちのいずれか1つであり、前記実アドレスに対してプレフィックス付加を行うことによって前記絶対アドレスが形成され、前記変換テーブル階層構造が、領域第1テーブル、領域第2テーブル、領域第3テーブル、及び前記セグメント・テーブルのうちの1つ又は複数から成り、前記セグメント・テーブルから前記セグメント・テーブル・エントリを取得する前記ステップが、A、B、C、Dのうちのいずれか1つを含み、

A) 前記領域第1テーブルの前記取得された初期基点アドレスに基づいて、前記仮想アドレスの第1インデックス部分を用いて前記領域第2テーブルの基点アドレスを含む前記領域第1テーブル内の領域第1テーブル・エントリを参照し、

前記領域第2テーブルの前記基点アドレスに基づいて、前記仮想アドレスの第2インデックス部分を用いて前記領域第3テーブルの基点アドレスを含む前記領域第2テーブル内の領域第2テーブル・エントリを参照し、前記領域第3テーブルの前記基点アドレスに基づいて、前記仮想アドレスの第3インデックス部分を用いて前記セグメント・テーブルの基点アドレスを含む前記領域第3テーブル内の領域第3テーブル・エントリを参照し、前記セグメント・テーブルの前記基点アドレスに基づいて、前記仮想アドレスのセグメント・インデックス部分を用いて前記セグメント・テーブル内のセグメント・テーブル・エントリを参照するステップ、

B) 前記領域第2テーブルの前記取得された初期基点アドレスに基づいて、前記仮想アドレスの前記第2インデックス部分を用いて前記領域第3テーブルの基点アドレスを含む前記領域第2テーブル内の前記領域第2テーブル・エントリを参照し、前記領域第3テーブルの前記基点アドレスに基づいて、前記仮想アドレスの前記第3インデックス部分を用いて前記セグメント・テーブルの前記基点アドレスを含む前記領域第3テーブル内の前記

10

20

30

40

50

領域第3テーブル・エントリを参照し、前記セグメント・テーブルの前記基点アドレスに基づいて、前記仮想アドレスの前記セグメント・インデックス部分を用いて前記セグメント・テーブル内の前記セグメント・テーブル・エントリを参照するステップ、

C) 前記領域第3テーブルの前記取得された初期基点アドレスに基づいて、前記仮想アドレスの前記第3インデックス部分を用いて前記セグメント・テーブルの前記基点アドレスを含む前記領域第3テーブル内の前記領域第3テーブル・エントリを参照し、前記セグメント・テーブルの前記基点アドレスに基づいて、前記仮想アドレスの前記セグメント・インデックス部分を用いて前記セグメント・テーブル内の前記セグメント・テーブル・エントリを参照するステップ、及び

D) 前記セグメント・テーブルの前記取得された初期基点アドレスに基づいて、前記仮想アドレスの前記セグメント・インデックス部分を用いて前記セグメント・テーブル・エントリを参照するステップ

である、請求項1に記載の方法。

【請求項3】

前記強化型DATファシリティがイネーブルであり、且つ前記フォーマット制御フィールドがイネーブルでないことに応答して、

前記セグメント・テーブル・エントリからページ・テーブルの基点を取得するステップであって、前記仮想アドレスのページ・インデックス部分を用いて前記ページ・テーブル内のページ・テーブル・エントリを参照するステップと、

前記ページ・テーブル・エントリから、主ストレージ内の小データ・ブロックのページ・フレーム実アドレスを取得するステップであって、前記小データ・ブロックは前記大データ・ブロックよりもサイズが小さく、且つ4キロバイトのデータ・ブロックから成り、前記小データ・ブロックは、ストレージ・アクセス制御フィールドとストレージ・フェッチ保護フィールドとを含むストレージ・キーに関連付けられており、主ストレージ内の所望の実データ・ブロックの変換アドレスは、前記ページ・フレーム実アドレスと前記仮想アドレスのバイト・インデックス部分との組合せを含む、ステップと、

前記プログラムのアクセス制御フィールドが前記ストレージ・アクセス制御フィールドに一致したことに応答してのみ、前記所望の実メモリ・ブロックに対するストアを許可するステップであって、前記ストアは前記仮想アドレスに関連付けられており、前記プログラムのアクセス制御フィールドは、プログラム・ステータス・ワード又は実行中のプログラム命令のオペランドのうちのいずれか1つを含むプログラム・エンティティによって与えられる、ステップと、

a) 前記ストレージ・フェッチ保護フィールドがディスエーブルであること、又はb) 前記ストレージ・フェッチ保護フィールドがイネーブルであり、且つ前記仮想アドレスに関連付けられたプログラム・アクセス・キーが前記ストレージ・アクセス制御フィールドと等しいこと、のいずれかにのみ応答して、前記所望のデータ・ブロックからのフェッチを許可するステップであって、前記フェッチは前記仮想アドレスに関連付けられており、前記プログラム・アクセス・キーは、プログラム・ステータス・ワード又は実行中のプログラム命令のオペランドのうちのいずれか1つによって与えられる、ステップと  
を実施する、請求項2に記載の方法。

【請求項4】

前記強化型DATファシリティがイネーブルでないことに応答して、

前記セグメント・テーブル・エントリからページ・テーブルの基点を取得し、前記仮想アドレスのページ・インデックス部分を用いて前記ページ・テーブル内のページ・テーブル・エントリを参照するステップと、

前記ページ・テーブル・エントリから、主ストレージ内の小データ・ブロックのページ・フレーム実アドレスを取得するステップであって、前記小データ・ブロックは前記大データ・ブロックよりもサイズが小さく、且つ、ストレージ・キーに関連付けられた4キロバイトのデータ・ブロックから成り、前記ストレージ・キーは、ストレージ・アクセス制御フィールド及びストレージ・フェッチ保護フィールドを含み、主ストレージ内の所望の

10

20

30

40

50

実データ・ブロックの変換アドレスは、前記ページ・フレーム実アドレスと前記仮想アドレスのバイト・インデックス部分との組合せを含む、ステップと、

前記プログラムのアクセス制御フィールドが前記ストレージ・アクセス制御フィールドに一致したことに応答してのみ、前記所望の実メモリ・ブロックに対するストアを許可するステップであって、前記ストアが前記仮想アドレスに関連付けられる、ステップと、

a) 前記ストレージ・フェッチ保護フィールドがディスエーブルであること、又はb) 前記ストレージ・フェッチ保護フィールドがイネーブルであり、且つ前記仮想アドレスに関連付けられたプログラム・アクセス・キーが前記ストレージ・アクセス制御フィールドに等しいこと、のいずれかにのみ応答して、前記所望のデータ・ブロックからのフェッチを許可するステップであって、前記フェッチは前記仮想アドレスに関連付けられており、前記プログラム・アクセス・キーは、プログラム・ステータス・ワード又は実行中のプログラム命令のオペランドのうちのいずれか1つによって与えられる、ステップとを実施する、請求項2に記載の方法。

10

【請求項5】

前記プログラム・アクセス・キーが0であることに応答して、前記所望のメモリ・ブロックへのストア及び前記所望のメモリ・ブロックからのフェッチを許可し、前記ストア及びフェッチが前記仮想アドレスに関連付けられる、請求項1に記載の方法。

【請求項6】

前記アクセス妥当性フィールド及び前記フェッチ保護フィールドがイネーブルであることに応答して、保護例外が、前記所望のメモリ・ブロックからのフェッチ動作が試行されたこと、且つ前記セグメント・アクセス制御フィールドが前記プログラムのアクセス制御フィールドに一致しないことに応答して示され、前記フェッチ動作が前記仮想アドレスに関連付けられている、請求項1に記載の方法。

20

【請求項7】

前記アクセス妥当性フィールドがイネーブルであるかどうかを判定する前記ステップが、a) 制御レジスタのストレージ保護オーバーライド・フィールドがディスエーブルかどうか、又はb) 前記ストレージ保護オーバーライド・フィールドがイネーブルであり、且つ前記セグメント・アクセス制御フィールドがイネーブルでないかどうか、のうちのいずれか1つを判定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8】

前記仮想アドレスの前記変換に用いられた情報を、少なくとも1つの変換ルックアサイド・バッファ内にストアするステップと、

その後の仮想アドレスの、主ストレージ内の前記データ・ブロックの前記絶対アドレスへのそれ以降の変換を、前記変換テーブル階層構造ではなく前記変換ルックアサイド・バッファ由来の前記ストアされた情報を用いて実施するステップとをさらに含む、請求項1に記載の方法。

30

【請求項9】

前記変換が前記システムのアーキテクチャに対してネイティブではないことに応答して、

前記変換をエミュレートするための所定のソフトウェア・ルーチンを識別するステップであって、前記所定のソフトウェア・ルーチンが複数の命令を含むステップと、

前記所定のソフトウェア・ルーチンを実行するステップとをさらに含む、請求項1に記載の方法。

40

【請求項10】

前記大データ・ブロックが、少なくとも1メガバイトのサイズのブロックを含む、請求項1に記載の方法。

【請求項11】

方法に係る請求項1から請求項10のいずれか1項に記載の方法の全てのステップを実行するように適合された手段を備える、システム。

【請求項12】

50

コンピュータ・プログラムであって、前記コンピュータ・プログラムがコンピュータ・システム上で実行されるときに、請求項 1 から請求項 10 のいずれか 1 項に記載の方法の全てのステップをコンピュータに実行させる、コンピュータ・プログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、一般に、コンピュータ・システムにおいて仮想アドレスを変換するためのシステム及び方法に関し、より具体的には、仮想アドレス変換が変換テーブル階層構造を経て生じる動的アドレス・ファシリティを有するコンピュータ・システムにおいて、仮想アドレスをデータ・ブロックの実アドレス又は絶対アドレスに変換するシステム及び方法に関する。

10

【背景技術】

【0002】

動的アドレス変換は、任意の瞬間にプログラムの実行への割り込みを行い、そのプログラム及びそのプログラムのデータを直接アクセス・ストレージ・デバイスなどの補助ストレージに記録し、後の時点で、プログラム及びデータを異なる主ストレージ位置に戻して実行を再開する能力を提供する。主ストレージと補助ストレージとの間のプログラム及びデータの転送は断片的に行うことができ、主ストレージへの情報のリターンは、その情報が実行のために必要になった時点でCPUによるアクセスの試行に回答して行われるものとして行うことができる。これらの機能は、プログラム及びそのデータを変更又は検査することなく実施することができ、再配置されたプログラムにおけるいかなる明示的なプログラミング規則も必要とせず、時間的遅延が伴うことを除いて、プログラムの実行を妨げない。

20

【0003】

オペレーティング・システムによる適切なサポートがあれば、動的アドレス変換ファシリティを用いて、ユーザに、ストレージがその構成において利用可能な主ストレージより大きく見えるようなシステムを提供することができる。この見かけの主ストレージは、しばしば仮想ストレージと呼ばれ、仮想ストレージ内の位置を指定するために用いられるアドレスは、しばしば仮想アドレスと呼ばれる。ユーザの仮想ストレージは、その構成において利用可能な主ストレージのサイズを遥かに超過することができ、通常は補助ストレージ内で維持される。仮想ストレージは、一般にページと呼ばれる（セグメント及び領域とも呼ばれる）データのブロックで構成されるものと見なされる。仮想ストレージのうち、直前に参照されたページのみが、物理的主ストレージのブロックを占有するように割り当てられる。ユーザが、主ストレージに現れていない仮想ストレージのページを参照する際には、それらのページを取り込んで、必要とされる可能性の低い主ストレージ内のページを置き換える。幾つかの場合には、仮想ストレージは、ストレージが参照されるかどうかに関わりなく、長期間にわたって（又は永続的に）主ストレージに割り当てられる。ストレージのページのスワッピングは、オペレーティング・システムによって、ユーザが知ることなく行われる。

30

【0004】

40

プログラムは、アドレス（又は仮想アドレス）を用いて仮想ストレージにアクセスする。プログラムは仮想アドレスを用いて、仮想ストレージから命令をフェッチすることができる、又は仮想ストレージからデータをロードするか若しくはデータをストアすることができる。ある範囲の仮想ストレージに関連付けられた仮想アドレスが、アドレス空間を定める。オペレーティング・システムによる適切なサポートがあれば、動的アドレス変換ファシリティを用いて、多数のアドレス空間を提供することができる。これらのアドレス空間を用いて、ユーザ間に様々な度合いの分離を提供することができる。このようなサポートは、ユーザごとに完全に異なるアドレス空間から成ることによって完全な分離を提供することもでき、又は各アドレス空間の一部を単一の共通ストレージ領域にマッピングすることによって共有エリアを提供することもできる。また、半特権的プログラムが1つより多

50

くのこのようなアドレス空間にアクセスすることを許可する命令も、提供される。

【0005】

動的アドレス変換は、多数の異なるアドレス空間からの仮想アドレスの変換を規定する。これらのアドレス空間は、一次アドレス空間、二次アドレス空間、及びアクセス・レジスタ指定アドレス空間と呼ばれる。特権的プログラムは、アクセスすべきホーム・アドレス空間を生じさせることもできる。動的アドレス変換は、CPUによって生成された命令及びデータ・アドレスに対して指定することができる。

【0006】

当該技術分野において今まで知られていない更なる機能、能力、及び保護を提供する、強化された動的アドレス変換ファシリティが必要とされている。

10

【先行技術文献】

【特許文献】

【0007】

【特許文献1】米国特許第5,551,013号

【特許文献2】米国特許第5,574,873号

【特許文献3】米国特許第5,790,825号

【特許文献4】米国特許第6,009,261号

【特許文献5】米国特許第6,308,255号

【特許文献6】米国特許第6,463,582号

【非特許文献】

20

【0008】

【非特許文献1】「z/Architecture (登録商標) Principles of Operation」、IBM (登録商標) 刊行番号SA22-7832-05、第6版、2007年4月

【非特許文献2】「IBM (登録商標) System/370 Extended Architecture」、IBM (登録商標) 刊行番号SA22-7095、1985年

【発明の概要】

【発明が解決しようとする課題】

【0009】

仮想アドレスを、変換のために用いられる変換テーブル階層構造を備えたマシン・アーキテクチャを有するコンピュータ・システムの主ストレージ内の所望のデータ・ブロックの実アドレス又は絶対アドレスに変換する強化型動的アドレス変換ファシリティのためのシステム、方法、及びコンピュータ・プログラム製品が提供される。実アドレスにプリフィックス付加を行って、絶対アドレスを形成することができる。

30

【課題を解決するための手段】

【0010】

1つの例示的な実施形態において、変換されるべき仮想アドレス、及び変換テーブル階層構造の変換テーブルの初期基点アドレスが取得される。変換テーブルは、領域第1テーブル、領域第2テーブル、領域第3テーブル、又はセグメント・テーブルから成る。取得された初期基点アドレスに基づいて、セグメント・テーブル・エントリがセグメント・テーブルから取得される。セグメント・テーブル・エントリは、フォーマット制御フィールドを含むように構成される。強化型DATファシリティ及びフォーマット制御フィールドがイネーブルであれば、セグメント・テーブル・エントリ内のアクセス妥当性フィールドがイネーブルであるかどうか判定される。アクセス妥当性フィールドがイネーブルであれば、セグメント・テーブル・エントリは、アクセス制御フィールド、保護フィールド、及び主ストレージ内の大データ・ブロックのセグメント・フレーム絶対アドレスをさらに含む。大データ・ブロックは少なくとも1メガバイトのサイズのブロックである。アクセス制御フィールドがイネーブルであれば、所望のデータ・ブロックへのストア動作は、アクセス制御フィールドがプログラム・ステータス・ワード又は実行中のプログラム命令のオペランドのいずれかによって与えられるプログラム・アクセス・キーに一致した場合に

40

50

のみ、許可される。所望のデータ・ブロックからのフェッチ動作は、仮想アドレスに関連付けられたプログラム・アクセス・キーがアクセス制御フィールドに等しい場合にのみ、許可される。

#### 【 0 0 1 1 】

さらに別の実施形態において、アクセス妥当性フィールドがイネーブルであれば、フェッチ動作又はストア動作が所望のメモリ・ブロックにおいて試行され、且つセグメント・アクセス制御フィールドがプログラム・アクセス制御フィールドと一致しないことに応答して、保護例外が示される。

#### 【 0 0 1 2 】

本発明は、次に、特定の例示的な実施形態との関連で説明される。当業者であれば本発明の精神及び範囲から逸脱することなく種々の変更及び修正を行うことが可能であることを理解されたい。

#### 【 0 0 1 3 】

添付の図面において、同一又は機能的に類似の要素には別々の図の全体を通じて同様の参照符号が付されており、これらの図面は以下の詳細な説明と共に本明細書に組み入れられ、その一部を形成するものであり、種々の実施形態をさらに例証し、且つ、全て本発明に従う種々の原理及び利点を説明する役割を果たす。

#### 【 図面の簡単な説明 】

#### 【 0 0 1 4 】

【 図 1 】 強化型動的アドレス変換を実施することができるホスト・コンピュータ・システムの一実施形態を示す。

【 図 2 】 ホスト・アーキテクチャのホスト・コンピュータ・システムをエミュレートする、エミュレートされたホスト・コンピュータ・システムの一例を示す。

【 図 3 】 プログラム・ステータス・ワードを用いて仮想アドレスの動的アドレス変換のための実効 A S C E を判定する手法の一実施形態を示す。

【 図 4 】 図 3 において判定された実効 A S C E を用いて、仮想アドレスの変換に用いられる変換テーブル階層構造の中で最も高位の変換テーブルを判定する一実施形態を示す。

【 図 5 】 変換テーブル階層構造をセグメント・テーブル・レベルまで用いて仮想アドレスを動的アドレス変換するプロセスの一実施形態を示す。

【 図 6 】 セグメント・テーブル・エントリ ( S T E ) フォーマット制御 ( F C ) が 0 である場合の、図 5 の動的アドレス変換の続きを示す。

【 図 7 】 セグメント・テーブル・エントリ ( S T E ) フォーマット制御 ( F C ) が 1 である場合の、図 5 の動的アドレス変換の続きを示す。

【 図 8 】 セグメント・テーブル・エントリ内のフォーマット制御フィールドを取得するための強化型 ( enhanced ) 動的アドレス変換 ( e D A T ) の一実施形態の流れ図を示す。

【 図 9 】 図 8 のノード 6 1 4 からの流れ図の続きを示す。

【 図 1 0 】 図 8 のノード 6 1 6 からの流れ図の続きを示す。

【 図 1 1 】 変換された仮想アドレスによってアドレス指定された所望のデータ・ブロックに適用すべき D A T 保護のレベルを判定する一実施形態の流れ図を示す。

【 図 1 2 】 変換された仮想アドレスによってアドレス指定された所望のデータ・ブロックに適用すべきアクセス制御保護のレベルを判定する、図 1 0 のノード 8 2 2 からの流れ図の続きを示す。

【 図 1 3 】 変換された仮想アドレスによってアドレス指定された所望のデータ・ブロックに適用すべきフェッチ保護のレベルを判定する、図 1 0 のノード 8 2 2 からの流れ図の続きを示す。

【 図 1 4 】 変更記録オーバーライド・フィールドがセグメント・テーブル・エントリから取得される、図 1 0 のノード 8 2 2 からの流れ図の続きを示す。

#### 【 発明を実施するための形態 】

#### 【 0 0 1 5 】

本出願の明細書における記述は、特許請求される種々の発明いずれをも必ずしも限定す

10

20

30

40

50

るものではないことを理解されたい。さらに、幾つかの記述は、幾つかの発明の特徴に当てはまることもあるが、他の発明の特徴には当てはまらないこともある。特に断りのない限り、単数形の要素は、一般性を失うことなく、複数形の場合もあり、その逆もまた真である。

#### 【 0 0 1 6 】

当業者であれば、コンピュータ環境内のストレージに対してアドレス指定すること、及び、様々な状態及びそれらの状態上での動作を示すためにレジスタ又はアドレス・フィールド内のビットを使用することに容易に精通するはずである。さらに、当業者であれば、コンピュータ・プログラム分野の知識を有し、かつ、コンピュータ・システムのコンポーネント間の働き及び相互関係についての知識を有しているはずである。

10

#### 【 0 0 1 7 】

##### 概要

強化型動的アドレス変換 ( D A T ) ファシリティの例示的な実施形態が提供される。強化型 D A T ファシリティがインストールされており、且つイネーブルである場合、 D A T 変換は、ページ・フレーム実アドレス、又はセグメント・テーブル・エントリ内のセグメント・テーブル・エントリ ( S T E ) フォーマット制御によって決定されるセグメント・フレーム絶対アドレスを生成することができる。本明細書において用いられる「強化型 D A T を適用する」という用語は、以下の全てが真であることを意味する。即ち、 1 ) E D A T ファシリティがインストールされており、 2 ) E D A T ファシリティが制御レジスタ 0 ( C R 0 ) ビット 4 0 を通じてイネーブルになっており、且つ 3 ) アドレスは D A T テーブル・エントリによって変換される。

20

#### 【 0 0 1 8 】

強化型 D A T を適用する場合、 D A T プロセスにおいて以下の追加機能が利用可能である。即ち、

D A T 保護ビットが領域テーブル・エントリに追加され、セグメント・テーブル・エントリ及びページ・テーブル・エントリ内の D A T 保護ビットと同様の機能を与える。

S T E フォーマット制御がセグメント・テーブル・エントリに追加される。 S T E フォーマット制御が 0 である場合、 D A T は、そのページに対する変更ビットの設定がバイパスされるかどうかをページ・テーブル・エントリ内の変更記録オーバーライドが示すこと以外は、現在定められている通りに進行する。

30

#### 【 0 0 1 9 】

S T E フォーマット制御が 1 である場合、セグメント・テーブル・エントリは、さらに以下を含む。即ち、

1 メガバイト・ブロックの絶対ストレージ位置を指定するセグメント・フレーム絶対アドレス ( ページ・テーブル基点ではなく ) 。

セグメントの個々のストレージ・キーの中に対応するビットの代わりに随意的に用いることができる、アクセス制御ビット及びフェッチ保護ビット。

セグメント・テーブル・エントリ内のアクセス制御ビット及びフェッチ保護ビットの妥当性を決定するビット。

セグメントの個々のストレージ・キーにおいて変更ビットの設定をバイパスすることができるかどうかを示す、変更記録オーバーライド。

40

#### 【 0 0 2 0 】

##### ホスト・コンピュータ・システム

図 1 を参照すると、ホスト・コンピュータ・システム 1 0 0 の代表的なコンポーネントが描かれている。他のコンポーネント構成をコンピュータ・システムにおいて用いることもでき、これは当該技術分野において周知である。

#### 【 0 0 2 1 】

ホスト・コンピュータ環境は、ニューヨーク州 A r m o n k 所在の I n t e r n a t i o n a l B u s i n e s s M a c h i n e s C o r p o r a t i o n ( I B M ( 登録商標 ) ) により提供される、 z / A r c h i t e c t u r e ( 登録商標 ) に基づくもの

50



であることが好ましい。z / A r c h i t e c t u r e (登録商標)は、引用によりその全体をここに組み入れる非特許文献1に、より詳細に記載されている。z / A r c h i t e c t u r e (登録商標)に基づくコンピュータ環境は、例えば、共にI B M (登録商標)によるe S e r v e r 及びz S e r i e s (登録商標)を含む。

#### 【0022】

代表的なホスト・コンピュータ100は、1つ又は複数のC P U 101を含み、これは、主ストア(コンピュータ・メモリ102)、並びに、ストレージ・デバイス111に対するI / Oインターフェース、及び他のコンピュータ又はストレージ・エリア・ネットワーク(S A N)などとの通信のためのネットワーク110に対するI / Oインターフェースと通信する。C P Uは、プログラム・アドレス(仮想アドレス)をメモリの実アドレスに変換するための動的アドレス変換(D A T)ファシリティ(機能又はユニット)103を含む。D A Tファシリティは、典型的には、後でコンピュータ・メモリ102のブロックにアクセスしたときにアドレス変換による遅延を必要とせずに済むように変換をキャッシュに入れるための、変換ルックアサイド・バッファ107を含む。典型的には、キャッシュ109は、コンピュータ・メモリ102とプロセッサ101との間で用いられる。キャッシュ109は、1つより多くのC P Uが利用できる大型キャッシュと、大型キャッシュと各C P Uとの間のより小型でより高速な(下位レベルの)キャッシュとを有する階層構造とすることができる。幾つかの実装において、下位レベルのキャッシュは、命令のフェッチ及びデータ・アクセスのための個別の低レベルキャッシュを提供するように分割される。一実施形態において、命令は、命令フェッチ・ユニット104によりメモリ102からキャッシュ109を経由してフェッチされる。命令は、命令デコード・ユニット(106)内でデコードされ、(幾つかの実施形態においては他の命令と共に)命令実行ユニット108へディスパッチされる。典型的にはいくつかの実行ユニット108、例えば、算術演算実行ユニット、浮動小数点実行ユニット、及び分岐命令実行ユニットが用いられる。命令は、実行ユニットにより、必要に応じて命令が指定するレジスタ又はメモリからのオペランドにアクセスしながら実行される。オペランドがメモリ102からアクセスされる(ロードされる又はストアされる)場合には、典型的には、ロード・ストア・ユニット105が、実行されている命令の制御下でアクセスを取り扱う。

#### 【0023】

一実施形態において、本発明は、ソフトウェア(ライセンス内部コード(L I C)、ファームウェア、マイクロ・コード、ミリ・コード、ピコ・コードなどと呼ばれる場合もあるが、そのいずれも本発明と整合性がある)により実施することができる。本発明を具体化するソフトウェア・プログラム・コードは、典型的には、コンピュータ・システム100のC P U(中央演算処理装置)101としても知られるプロセッサにより、C D R O Mドライブ、テープ・ドライブ又はハード・ドライブといった長期ストレージ媒体111からアクセスされる。ソフトウェア・プログラム・コードは、ディスク、ハード・ドライブ又はC D R O Mといった、データ処理システムと共に使用するための種々の公知の媒体のいずれかの上で具体化することができる。コードは、そのような媒体上で配布することができ、又は1つのコンピュータ・システムのコンピュータ・メモリ102若しくはストレージから、他のシステムのユーザが使用するために、ネットワーク110を介してそのような他のコンピュータ・システムに配布することができる。

#### 【0024】

あるいは、プログラム・コードをメモリ102内で具体化し、プロセッサ・バスを用いてプロセッサ101によってアクセスすることができる。このようなプログラム・コードは、種々のコンピュータ・コンポーネント及び1つ又は複数のアプリケーション・プログラムの機能及び相互作用を制御するオペレーティング・システムを含む。プログラム・コードは、通常、高密度ストレージ媒体111から高速メモリ102にページングされ、そこでプロセッサ101による処理のために利用可能になる。ソフトウェア・プログラム・コードをメモリ内、物理媒体上で具体化し、及び/又はネットワークを介してソフトウェア・コードを配布する技術及び方法は周知であり、ここではこれ以上論じない。プログラ

ム・コードは、作成され、有形の媒体（電子メモリ・モジュール（ＲＡＭ）、フラッシュメモリ、コンパクトディスク（ＣＤ）、ＤＶＤ、磁気テープなどを含むがこれらに限定されない）に格納された場合には、しばしば「コンピュータ・プログラム製品」と呼ばれる。コンピュータ・プログラム製品媒体は、典型的には、処理回路によって実行されるために、好ましくはコンピュータ・システム内にある処理回路によって読み出し可能である。

#### 【００２５】

図２に、ホスト・アーキテクチャのホスト・コンピュータ・システム１００をエミュレートする、エミュレートされたホスト・コンピュータ・システム２０１の一例を示す。エミュレートされたホスト・コンピュータ・システム２０１において、ホスト・プロセッサ（ＣＰＵ）２０８は、エミュレートされたホスト・プロセッサ（又は仮想ホスト・プロセッサ）であり、ホスト・コンピュータ１００のプロセッサ１０１によって用いられるものとは異なるネイティブ命令セット・アーキテクチャを有するエミュレーション・プロセッサ２０７を備える。エミュレートされたホスト・コンピュータ・システム２０１は、エミュレーション・プロセッサ２０７によってアクセス可能なメモリ２０２を有する。例示的な実施形態において、メモリ２０２は、ホスト・コンピュータ・メモリ１０２部分と、エミュレーション・ルーチン２０３部分（エミュレーションを提供するルーチンはホスト・メモリの一部とすることができる）とのパーティションに分けられている。ホスト・コンピュータ・メモリ１０２は、ホスト・コンピュータ・アーキテクチャに従い、エミュレートされたホスト・コンピュータ・システム２０１のプログラムによって利用可能である。エミュレーション・プロセッサ２０７は、エミュレートされたプロセッサ２０８のアーキテクチャ以外のアーキテクチャのアーキテクチャ化された（architected）命令セットのネイティブ命令を実行し、このネイティブ命令はエミュレーション・ルーチン・メモリ２０３から取得されたものであり、且つ、エミュレーション・プロセッサ２０７は、シーケンス及びアクセス／デコード・ルーチンにおいて取得される１つ又は複数の命令を用いることによって、ホスト・コンピュータ・メモリ１０２の中のプログラム由来の実行のためのホスト命令にアクセスすることができ、このシーケンス及びアクセス／デコード・ルーチンは、アクセスされたホスト命令をデコードして、アクセスされたホスト命令の機能をエミュレートするためのネイティブ命令実行ルーチンを判定することができる。

#### 【００２６】

ホスト・コンピュータ・システム１００アーキテクチャのために定義された、例えば、汎用レジスタ、制御レジスタ、動的アドレス変換、並びにＩ／Ｏサブシステム・サポート及びプロセッサ・キャッシュといったファシリティを含む他のファシリティを、アーキテクチャ化ファシリティ・ルーチンによってエミュレートすることができる。エミュレーション・ルーチンは、エミュレーション・ルーチンの性能を高めるために、エミュレーション・プロセッサ２０７において利用可能な（汎用レジスタ、及び仮想アドレスの動的変換といった）ファシリティを利用することもできる。ホスト・コンピュータ１００の機能をエミュレートする際にプロセッサ２０７を補助するために、特殊ハードウェア及びオフ・ロード・エンジンを設けることもできる。

#### 【００２７】

コンピュータ・プロセッサ及びレジスタ

一実施形態において、ＣＰＵのプログラム命令機能は、通信バスを介して複数のレジスタと通信する。通信バスは、ＣＰＵの内部にあっても、又は外部にあってもよい。いくつかのレジスタは読み出し専用とすることができる。他のハードウェア及び／又はソフトウェアも、ＣＰＵによってアクセス可能な１つ又は複数のレジスタを読み出す／書き込むことができる。命令動作コード（オペコード）が、いずれかの特定のマシン命令動作において、どのタイプのレジスタが用いられるかを決定する。

#### 【００２８】

汎用レジスタ

命令は、１６個の汎用レジスタのうちの１つ又は複数の中の情報を指示することができる。汎用レジスタは、アドレス算術演算におけるベース・アドレス・レジスタ及びインデ

10

20

30

40

50

ックス・レジスタとして用いることができ、且つ、一般算術演算及び論理演算におけるアキュムレータとして用いることができる。各レジスタは、64個のビット位置を含む。汎用レジスタは、番号0 - 15によって識別され、命令内の4ビットのRフィールドによって指示される。ある命令は、数個のRフィールドを有することによって、複数の汎用レジスタをアドレス指定することを規定する。命令によっては、特定の汎用レジスタの使用は、命令のRフィールドによって明示的に指示されるのではなく暗黙的であることもある。

#### 【0029】

演算によっては、2つの隣接する汎用レジスタのビット32 - 63又は0 - 63のいずれかを結合して、それぞれ、64ビット又は128ビット形式を提供する場合もある。これらの演算においては、プログラムは、最左端（高位）の32ビット又は64ビットを含む偶数番号のレジスタを指示しなければならない。次に高位の番号のレジスタは最右端（低位）の32ビット又は64ビットを含む。一般算術演算及び論理演算におけるアキュムレータとしての使用に加えて、16個の汎用レジスタのうちの15個は、アドレス生成におけるベース・アドレス及びインデックス・レジスタとして用いることもできる。これらの場合において、レジスタは、命令内の4ビットのBフィールド又はXフィールドによって指示される。B又はXフィールドにおける0の値は、適用されるべきベース又はインデックスが存在しないことを指定し、従って、汎用レジスタ0を、ベース・アドレス又はインデックスを含むものとして指示することはできない。

#### 【0030】

#### 制御レジスタ

制御レジスタは、プログラム・ステータス・ワードの外部での制御情報の保持及び操作を規定する。CPUは、各々が64のビット位置を有する16個の制御レジスタを有する。レジスタ内のビット位置は、プログラム・イベント記録のようなシステム内の特定のファシリティに割り当てられ、ある演算を行うことができることを指定すること、又はそのファシリティが要求する特別な情報を供給することのどちらかのために用いられる。制御レジスタは、番号0 - 15によって識別され、LOAD CONTROL命令及びSTORE CONTROL命令内の4ビットのRフィールドによって指示される。複数の制御レジスタを、これらの命令によってアドレス指定することが可能である。

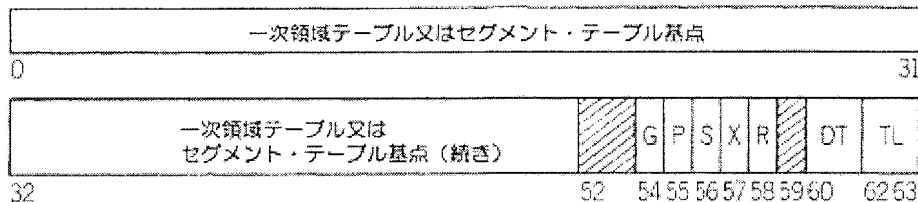
#### 【0031】

#### 制御レジスタ1

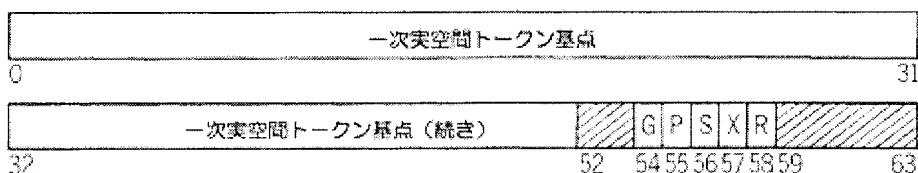
制御レジスタ1は、一次アドレス空間制御要素（PASCЕ）を含む。一実施形態において、制御レジスタ1は、レジスタ内の実空間制御ビット（R）に応じて、以下の2つのフォーマットのうちの1つを有する。

#### 【表1】

一次領域テーブル又は  
セグメント・テーブル指示（R=0）



一次実空間指示（R=1）



#### 制御レジスタ1のフォーマット

## 【 0 0 3 2 】

一次アドレス空間制御要素 ( P A S C E ) 内の選択されたフィールドは、以下のように割り振られる。

一次領域テーブル又はセグメント・テーブル基点：制御レジスタ 1 の中の一次領域テーブル又はセグメント・テーブル指示のビット 0 - 5 1 は、右に 1 2 個の 0 が付加されると、一次領域テーブル又はセグメント・テーブルの開始位置を指示する 6 4 ビット・アドレスを形成する。このアドレスが実であるか絶対であるかは予測不可能である。このテーブルは、一次アドレス空間における仮想アドレスを変換するために用いられるので、一次領域テーブル又はセグメント・テーブルと呼ばれる。

## 【 0 0 3 3 】

一次実空間制御 ( R )：制御レジスタ 1 のビット 5 8 が 0 であれば、レジスタは領域テーブル又はセグメント・テーブル指示を含む。ビット 5 8 が 1 であれば、レジスタは実空間指示を含む。ビット 5 8 が 1 である場合、セグメント・テーブル・エントリの変換ルックアサイド・バッファ表示内の共通セグメント・ビットの値 1 は、制御レジスタ 1 の中のトークン基点と変換ルックアサイド・バッファ・エントリの中のテーブル基点との間に一致が存在したとしても、一次アドレス空間への参照を変換するときにエントリ及びそれが指示する変換ルックアサイド・バッファ・ページ・テーブル・コピーが用いられることを防止する。

## 【 0 0 3 4 】

一次指示タイプ制御 ( D T )：R が 0 である場合、制御レジスタ 1 の中のテーブル指示タイプは、レジスタ内のビット 6 0 及び 6 1 によって以下のように指定される。

## 【表 2】

| ビット 6 0<br>及び 6 1 | 指示タイプ      |
|-------------------|------------|
| 1 1               | 領域第 1 テーブル |
| 1 0               | 領域第 2 テーブル |
| 0 1               | 領域第 3 テーブル |
| 0 0               | セグメント・テーブル |

一次指示タイプ(D T)制御ビット

## 【 0 0 3 5 】

R が 0 である場合、最左端の 1 ビットがアドレスのビット位置 0 - 1 0 にある仮想アドレスを変換するために P A S C E の使用が試行される場合には、ビット 6 0 及び 6 1 は二進数字 1 1 でなければならない。同様に、最左端の 1 ビットがアドレスのビット位置 1 1 - 2 1 にある場合には、ビット 6 0 及び 6 1 は二進数字 1 1 又は 1 0 でなければならない、最左端の 1 ビットがアドレスのビット位置 2 2 - 3 2 にある場合には、二進数字 1 1、1 0、又は 0 1 でなければならない。それ以外の場合には、A S C E タイプ例外が認識される。

## 【 0 0 3 6 】

一次領域テーブル又はセグメント・テーブル長 ( T L )：制御レジスタ 1 の中の一次領域テーブル指示又はセグメント・テーブル指示のビット 6 2 及び 6 3 は、一次領域テーブル又はセグメント・テーブルの長さを 4 , 0 9 6 バイトの単位で指定し、従って、領域テーブル又はセグメント・テーブルの長さを 5 1 2 エントリの倍数として可変にする。一次領域テーブル又はセグメント・テーブルの長さは、4 , 0 9 6 バイト単位で、T L の値より 1 多い。長さフィールドの内容は、そのテーブルによって変換されるべき仮想アドレスの一部 ( R F X、R S X、R T X、又は S X ) が、そのテーブルの範囲内にあるエントリを指示しているかどうかを確立するために用いられる。

## 【 0 0 3 7 】

一次実空間トークン基点：制御レジスタ 1 の中の一次実空間のビット 0 - 5 1 は、右に 1 2 個の 0 が付加されると、6 4 ビットのアドレスを形成し、このアドレスは、一次アドレス空間の参照のための仮想と実が等しい変換 (virtual equals real translation) を提供する変換ルックアサイド・バッファ・エントリの形成及び使用の際に用いることができる。このアドレスはトークンとしてのみ用いられるものであり、ストレージ参照を実施するために用いられるものではないが、それでもなお、有効なアドレスでなければならない。そうでなければ、制御レジスタ 1 の内容が用いられるときに、正しくない変換ルックアサイド・バッファ・エントリが用いられることがある。

## 【 0 0 3 8 】

10

制御レジスタ 1 の以下のビット、即ち、レジスタが領域テーブル指示又はセグメント・テーブル指示を含む場合のビット 5 2、5 3 及び 5 9、レジスタが実空間指示を含む場合のビット 5 2、5 3 及び 5 9 - 6 3 は、割り当てられず、無視される。

## 【 0 0 3 9 】

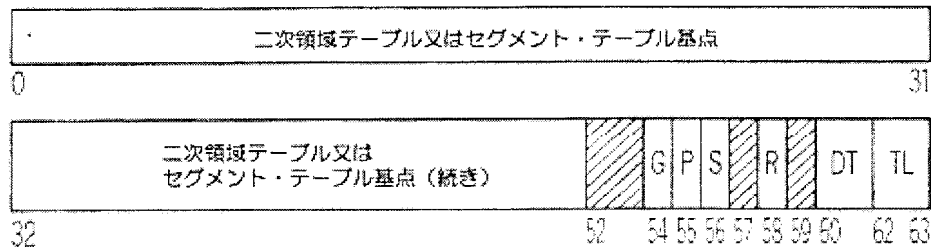
制御レジスタ 7

制御レジスタ 7 は、二次アドレス空間制御要素 (S A S C E) を含む。一実施形態において、制御レジスタ 7 は、レジスタ内の実空間制御ビット (R) に応じて、以下の 2 つのフォーマットのうちの 1 つを有する。即ち、

## 【表 3】

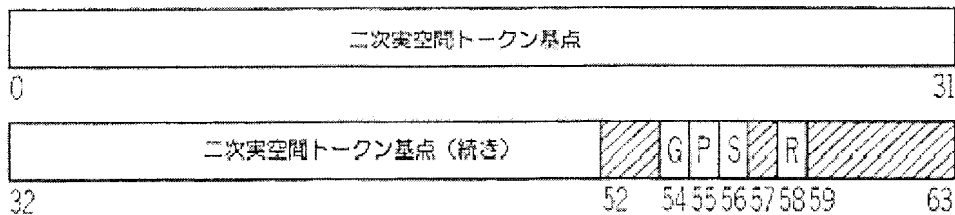
二次領域テーブル又は  
セグメント・テーブル指示 (R=0)

20



二次実空間指示 (R=1)

30



制御レジスタ 7 のフォーマット

## 【 0 0 4 0 】

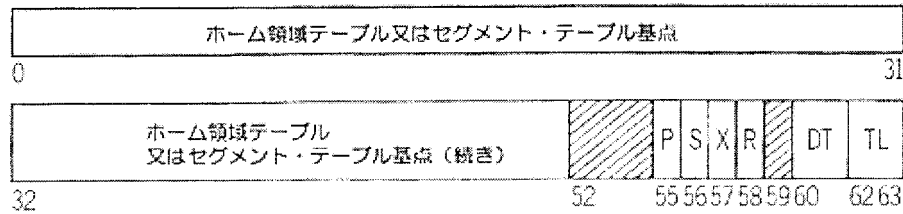
制御レジスタ 1 3

制御レジスタ 1 3 は、ホーム・アドレス空間制御要素 (H A S C E) を含む。一実施形態において、制御レジスタ 1 3 は、レジスタ内の実空間制御ビット (R) に応じて、以下の 2 つのフォーマットのうちの 1 つを有する。即ち、

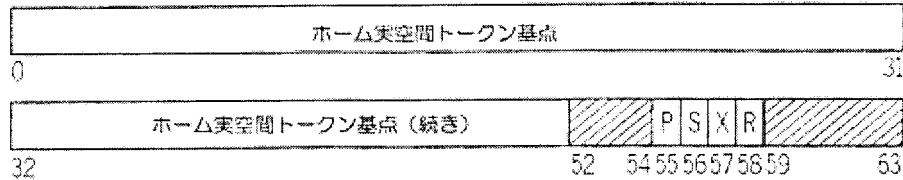
40

【表 4】

ホーム領域テーブル又は  
セグメント・テーブル指示 (R=0)



ホーム実空間指示 (R=1)



### 制御レジスタ 13 のフォーマット

#### 【0041】

#### アクセス・レジスタ

CPUは、0 - 15の番号を与えられた16個のアクセス・レジスタを有する。アクセス・レジスタは、ASCEの間接的な指定を含む32個のビット位置から成る。ASCEは、動的アドレス変換(DAT)機構によって、対応するアドレス空間に対する参照を変換するために用いられるパラメータである。CPUがアクセス・レジスタ・モードと呼ばれるモード(プログラム・ステータス・ワード内のビットによって制御される)にある場合、ストレージ・オペランド参照のための論理アドレスを指定するために用いられる命令Bフィールドがアクセス・レジスタを指示し、そのアクセス・レジスタによって指定されるASCEが、DATによって、行われている参照のために用いられる。命令によっては、Bフィールドの代わりにRフィールドが用いられることもある。アクセス・レジスタの内容をロード及びストアするための命令、並びに1つのアクセス・レジスタの内容を別のアクセス・レジスタに移動するための命令が与えられる。

#### 【0042】

アクセス・レジスタ1 - 15の各々は、現在の命令空間(一次アドレス空間)を含めた任意のアドレス空間を指示することができる。アクセス・レジスタ0は、一次アドレス空間を指示する。アクセス・レジスタ1 - 15のうちの1つがアドレス空間を指示するために用いられている場合、CPUは、そのアクセス・レジスタの内容を変換することによって、どのアドレス空間が指示されているかを判定する。アクセス・レジスタ0がアドレス空間を指示するために用いられる場合、CPUは、そのアクセス・レジスタが一次命令空間を指示するものとして扱い、アクセス・レジスタの実際の内容は検査しない。従って、16個のアクセス・レジスタは、任意の一時点に、一次命令空間と、最大で15個の他の空間とを指示することができる。

#### 【0043】

#### プログラム・ステータス・ワード (PSW)

プログラム・ステータス・ワードは、命令アドレスと、条件コードと、命令の順序付けを制御し、CPUの状態を決定するために用いられる他の情報とを含む。アクティブな又は制御中のプログラム・ステータス・ワードは、カレント・プログラム・ステータス・ワードと呼ばれる。これは、現在実行されているプログラムを支配する。

#### 【0044】

CPUは割り込み能力を有し、これにより、CPUを例外条件及び外部刺激に応答して別のプログラムに迅速に切り替えることが可能になる。割り込みが発生すると、CPUは、カレント・プログラム・ステータス・ワードを、古いプログラム・ステータス・ワード

10

20

30

40

50

位置と呼ばれる、特定の割り込みのクラス毎に割り当てられたストレージ位置に置く。CPUは、第2の割り当てられたストレージ位置から新たなプログラム・ステータス・ワードをフェッチする。この新たなプログラム・ステータス・ワードが、次に実行されるプログラムを決定する。割り込みの処理が終了すると、割り込みを扱うプログラムは、割り込まれたプログラムが実行できるように、古いプログラム・ステータス・ワードをリロードして、それを再びカレント・プログラム・ステータス・ワードとすることができる。

【0045】

割り込みには、外部、I/O、マシン・チェック、プログラム、再起動、及びスーパーバイザー・コールの6つのクラスがある。各クラスは、実ストレージに永続的に割り当てられた、古いプログラム・ステータス・ワード位置と新しいプログラム・ステータス・ワード位置との別個のペアを有する。

10

【0046】

カレント・プログラム・ステータス・ワード

CPU内のカレント・プログラム・ステータス・ワードは、現在アクティブなプログラムの実行に必要な情報を含む。プログラム・ステータス・ワードは、128ビット長であり、命令アドレス、条件コード、及び他の制御フィールドを含む。一般に、プログラム・ステータス・ワードは、命令の順序付けを制御し、CPUの状態のうちの多くを、現在実行中のプログラムとの関係で保持及び表示するために用いられる。付加的な制御及び状態情報は、制御レジスタ及び永続的に割り当てられたストレージ位置に含まれる。CPUの状態は、新たなプログラム・ステータス・ワード又はプログラム・ステータス・ワードの一部をロードすることによって変更することができる。

20

【0047】

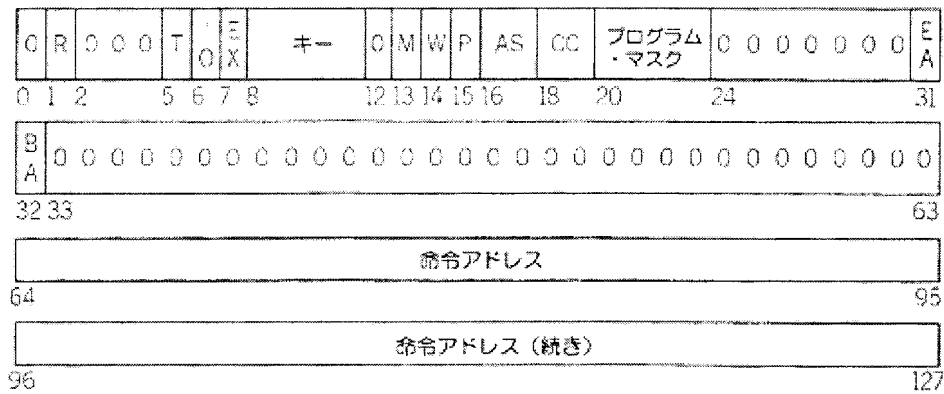
CPUの割り込み中に、CPUの状態を保存するためにカレント・プログラム・ステータス・ワードをストアし、次いで新たなプログラム・ステータス・ワードをロードすることによって、制御が切り替えられる。LOAD PSW又はLOAD PSW EXTENDEDの実行、又は初期プログラム・ロード・シーケンスの終結の成功により、新たなプログラム・ステータス・ワードが導入される。命令アドレスは、逐次的な命令実行によって更新され、成功した分岐によって置換される。プログラム・ステータス・ワードの一部の上で動作する他の命令が与えられる。

【0048】

30

プログラム・ステータス・ワードを変更する命令の割り込み又は実行が完了すると、新たな又は修正されたプログラム・ステータス・ワードがアクティブになる（即ち、カレント・プログラム・ステータス・ワードに導入された情報がCPUに対する制御を担う）。プログラム・ステータス・ワードを変更する命令に関連するプログラム・イベント記録（PER）のための割り込みは、その動作の開始時に有効なPERマスクの制御下で生じる。プログラム・ステータス・ワードのビット0 - 7は集合的にシステム・マスクと呼ばれる。一実施形態において、プログラム・ステータス・ワードは以下のフォーマットを有する。

【表 5】



## プログラム・ステータス・ワードのフォーマット

## 【 0 0 4 9 】

以下は選択されたプログラム・ステータス・ワードフィールドの機能の概要である。

## 【 0 0 5 0 】

D A Tモード ( T ) : ビット 5 は、ストレージにアクセスするために用いられる論理アドレス及び命令アドレスの暗黙的な動的アドレス変換が行われるかどうかを制御する。ビット 5 が 0 である場合には、D A Tはオフであり、論理アドレス及び命令アドレスは実アドレスとして扱われる。ビット 5 が 1 である場合には、D A Tはオンであり、動的アドレス変換機構が起動される。

## 【 0 0 5 1 】

P S Wキー : ビット 8 - 1 1 は、C P Uによるストレージ参照のためのアクセス・キーを形成する。参照がキー制御保護を受ける場合は、情報がストアされる時又は情報がフェッチに対して保護されている位置からフェッチされるときに、P S Wキーは、ストレージ・キーと照合される。しかし、M O V E T O P R I M A R Y、M O V E T O S E C O N D A R Y、M O V E W I T H K E Y、M O V E W I T H S O U R C E K E Y、及びM O V E W I T H D E S T I N A T I O N K E Yの各々のオペランドのうちの1つについては、オペランドとして指定されるアクセス・キーがP S Wキーの代わりに用いられる。

## 【 0 0 5 2 】

アドレス空間制御 ( A S ) : ビット 1 6 及び 1 7 は、プログラム・ステータス・ワードのビット 5 と共に、変換モードを制御する。

## 【 0 0 5 3 】

条件コード ( C C ) : ビット 1 8 及び 1 9 は 2 ビットの条件コードである。条件コードは、特定の命令の実行で得られた結果に応じて、0、1、2、又は 3 に設定される。ほとんどの算術演算及び論理演算、並びに幾つかの他の演算は、条件コードを設定する。B R A N C H O N C O N D I T I O N命令は、分岐のための基準として、条件コード値のいずれかの選択を指定することができる。

## 【 0 0 5 4 】

命令アドレス : プログラム・ステータス・ワードのビット 6 4 - 1 2 7 は、命令アドレスである。このアドレスは、C P Uが待機状態 ( プログラム・ステータス・ワードのビット 1 4 が 1 ) でない限り、次に実行される命令の最左端のバイトの位置を指示する。

## 【 0 0 5 5 】

アドレスのタイプ及びフォーマット

主ストレージをアドレス指定するために、絶対、実、及び仮想の 3 つの基本的なアドレスのタイプが認識される。アドレスは、ストレージ・アクセスの間にアドレスに適用される変形に基づいて区別される。アドレス変換は、仮想アドレスを実アドレスに変換する。プリフィックス付加は、実アドレスを絶対アドレスに変換する。3 つの基本的なアドレス



・タイプに加えて、命令及びカレント・モードに応じて、3つの基本的なタイプのうちの1つ又は他の1つとして扱われる付加的なタイプが定義される。

【0056】

絶対アドレス

絶対アドレスは、主ストレージ位置に割り当てられたアドレスである。絶対アドレスは、いかなる変形も行われることなく、ストレージ・アクセスのために用いられる。構成内のチャネル・サブシステム及び全てのCPUは、同じ絶対アドレスを用いることによって、共有された主ストレージ位置を参照する。利用可能な主ストレージには、通常、0から始まる連続した絶対アドレスが割り当てられ、このアドレスは整数の境界上の完全な4キロバイトのブロックで割り当てられる。物理位置に割り当てられていないブロック内の絶対アドレスを用いようとする試行が行われた場合、例外が認識される。幾つかのモデルでは、オペレータが絶対アドレスと物理位置との間の対応を変更することを許可する、ストレージ再構成制御が提供されることがある。しかし、どの時点であっても、1つの物理位置に1つより多くの絶対アドレスが割り当てられることはない。その絶対アドレスに従って配列されたバイト位置から成るストレージを、絶対ストレージと呼ぶ。

10

【0057】

実アドレス

実アドレスは、実ストレージにおける位置を特定する。主ストレージへのアクセスのために実アドレスが用いられる場合には、実アドレスをプリフィックス付加によって変換して、絶対アドレスが形成される。任意の一時において、構成内のCPUごとに、実アドレスから絶対アドレスへのマッピングが1つ存在する。実アドレスが、CPUによって主ストレージにアクセスするために用いられる場合には、実アドレスをプリフィックス付加によって絶対アドレスに変換することができる。具体的な変形は、そのCPUのためのプリフィックス・レジスタ内の値によって定められる。その実アドレスに従って配列されたバイト位置から成るストレージを、実ストレージと呼ぶ。

20

【0058】

仮想アドレス

仮想アドレスは、仮想ストレージにおける位置を特定する。主ストレージへのアクセスのために仮想アドレスが用いられる場合には、仮想アドレスは、動的アドレス変換によって、プリフィックス付加を行うと絶対アドレスを形成することができる実アドレスに変換されるか、又は直接、絶対アドレスに変換されるかのいずれかである。

30

【0059】

一次仮想アドレス

一次仮想アドレスは、一次アドレス空間制御要素(PASCE)によって変換されるべき仮想アドレスである。論理アドレスは、一次空間モードにある場合に一次仮想アドレスとして扱われる。命令アドレスは、一次空間モード、二次空間モード、又はアクセス・レジスタ・モードにある場合に一次仮想アドレスとして扱われる。MOVE TO PRIMARYの第1オペランド・アドレス、及びMOVE TO SECONDARYの第2オペランド・アドレスは、一次仮想アドレスとして扱われる。

40

【0060】

二次仮想アドレス

二次仮想アドレスは、二次アドレス空間制御要素(SASCE)によって変換されるべき仮想アドレスである。論理アドレスは、二次空間モードにある場合に二次仮想アドレスとして扱われる。MOVE TO PRIMARYの第2オペランド・アドレス、及びMOVE TO SECONDARYの第1オペランド・アドレスは、二次仮想アドレスとして扱われる。

【0061】

AR指定仮想アドレス

AR指定仮想アドレスは、アクセス・レジスタ指定アドレス空間制御要素によって変換されるべき仮想アドレスである。論理アドレスは、アクセス・レジスタ・モードにある場

50

合に A R 指定仮想アドレスとして扱われる。

【 0 0 6 2 】

## ホーム仮想アドレス

ホーム指定仮想アドレスは、ホーム・アドレス空間制御要素（H A S C E）によって変換されるべき仮想アドレスである。論理アドレス及び命令アドレスは、ホーム空間モードにある場合にホーム仮想アドレスとして扱われる。

【 0 0 6 3 】

## 命令アドレス

ストレージから命令をフェッチするために用いられるアドレスは、命令アドレスと呼ばれる。命令アドレスは、実モードにあるときには実アドレスとして扱われ、一次空間モード、二次空間モード、又はアクセス・レジスタ・モードにあるときには一次仮想アドレスとして扱われ、ホーム空間モードにあるときにはホーム仮想アドレスとして扱われる。カレント・プログラム・ステータス・ワード内の命令アドレス、及びEXECUTEのターゲット・アドレスは、命令アドレスである。

【 0 0 6 4 】

実効アドレス

ある状況においては、「実効アドレス」という用語を用いることが便利である。実効アドレスは、動的アドレス変換又はプリフィックス付加によるいずれかの変形が行われる前に存在するアドレスである。実効アドレスは、レジスタ内において直接指定されることもあり、又はアドレス算術演算の結果であることもある。アドレス算術演算は、ベースと変位との加算、又はベースとインデックスと変位との加算である。

【 0 0 6 5 】

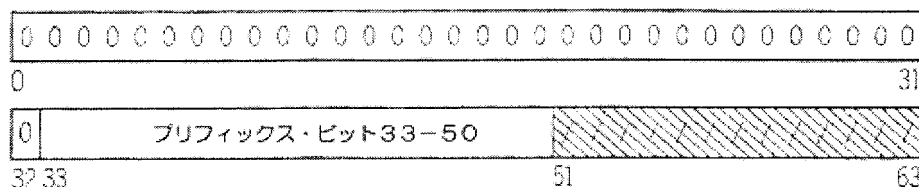
プリフィックス付加

プリフィックス付加は、実アドレスの 0 - 8 1 9 1 の範囲を、C P U ごとに絶対ストレージ内の異なるブロックに割り当て、それにより、特に割り込みの処理の際に、主ストレージを共有する 1 つより多くの C P U が最小限の干渉で同時に動作することを可能にする能力を提供する。プリフィックス付加により、0 - 8 1 9 1 の範囲の実アドレスを、その C P U のためのプリフィックス・レジスタのビット位置 0 - 5 0 における値によって識別される 8 K バイトの絶対アドレス・ブロック（プリフィックス領域）に 1 対 1 で対応させ、且つ、プリフィックス・レジスタ内のその値によって識別される実アドレス・ブロックを、絶対アドレス 0 - 8 1 9 1 に 1 対 1 で対応させる。残りの実アドレスは、対応する絶対アドレスと同一である。この変形により、各 C P U は、最初の 8 K バイトと、他の C P U のプリフィックス・レジスタによって指示される位置とを含む、主ストレージの全てにアクセスすることが可能になる。

【 0 0 6 6 】

プリフィックスは、プリフィックス・レジスタのビット位置 0 - 50 に含まれる 51 ビット量である。一実施形態において、プリフィックス・レジスタは、以下のフォーマットを有する。

## 【表 6】



## プリフィックス・レジスタのフォーマット

【 0 0 6 7 】

プリフィックス付加が適用された場合、実アドレスは、実アドレスのビット 0 - 50 に  
応じて、以下の規則のうちの 1 つを用いることによって絶対アドレスに変形される。即ち

アドレスのビット 0 - 50 は、全てが 0 であれば、プリフィックスのビット 0 - 50 で置き換えられる。

アドレスのビット 0 - 50 は、プリフィックスのビット 0 - 50 と等しければ、0 で置き換えられる。

アドレスのビット 0 - 50 は、0 でないものがあり、且つプリフィックスのビット 0 - 50 と等しくなければ、変更されないままである。

【 0 0 6 8 】

ストレージに提示されたアドレスのみがプリフィックス付加によって変換される。アドレスのソースの内容は変更されないままである。

【 0 0 6 9 】

実アドレスと絶対アドレスとは、プリフィックス・レジスタが全て 0 を含み、従って実アドレスと対応する絶対アドレスとが同一である場合でさえ、区別される。

【 0 0 7 0 】

図 15 は、実アドレスと絶対アドレスとの間の関係を示す。

【 0 0 7 1 】

アドレス空間は、整数番号の連続シーケンス（仮想アドレス）であり、各番号をストレージ内のバイト位置に関連付けることを可能にする特定の变形パラメータを伴う。シーケンスは、0 から開始して左から右へと進む。

【 0 0 7 2 】

仮想アドレスは、CPU によって主ストレージにアクセスするために用いられる場合、最初に、動的アドレス変換（DAT）によって実アドレス又は絶対アドレスに変換される。実アドレスに対してさらにプリフィックス付加を行って、絶対アドレスを形成することができる。DAT は、变形パラメータとして、領域第 1 テーブル、領域第 2 テーブル、領域第 3 テーブル、セグメント・テーブル、及びページ・テーブルを用いることができる。特定のアドレス空間のための最も高位のレベルのテーブルの指示（基点及び長さ）はアドレス空間制御要素（ASCE）と呼ばれ、これは、DAT による使用のために制御レジスタ内に見出されるか、又はアクセス・レジスタによって指定された通りのものである。あるいは、アドレス空間のための ASCE が実空間指示の場合もあり、これは、DAT が、いかなるテーブルも用いることなく単に仮想アドレスを実アドレスとして扱うことで仮想アドレスを変換するということを示す。

【 0 0 7 3 】

DAT は、異なる時点で、異なる制御レジスタ内の ASCE が、又はアクセス・レジスタによって指定された ASCE を用いる。選択は、カレント・プログラム・ステータス・ワードにおいて指定される変換モードによって決定される。一次空間モード、二次空間モード、アクセス・レジスタ・モード、及びホーム空間モードの 4 つの変換モードが利用可能である。変換モードに応じて、異なるアドレス空間をアドレス指定することが可能である。

【 0 0 7 4 】

CPU が一次空間モード又は二次空間モードにある場合にはいつでも、CPU は、2 つのアドレス空間、即ち一次アドレス空間及び二次アドレス空間に属する仮想アドレスを変換することができる。CPU がアクセス・レジスタ・モードにある場合にはいつでも、CPU は、最大で 16 個のアドレス空間、即ち、一次アドレス空間及び最大で 15 個の AR 指定アドレス空間の仮想アドレスを変換することができる。CPU がホーム空間モードにある場合にはいつでも、CPU は、ホーム・アドレス空間の仮想アドレスを変換することができる。

【 0 0 7 5 】

一次アドレス空間は、一次アドレス空間制御要素（PASCE）によって変換される一次仮想アドレスから成るので、それ自体として識別される。同様に、二次アドレス空間は、二次アドレス空間制御要素（SASCE）によって変換される二次仮想アドレスから成る。AR 指定アドレス空間は、アクセス・レジスタ指定アドレス空間制御要素（AR 指定

10

20

30

40

50

A S C E ) によって変換される A R 指定仮想アドレスから成り、ホーム・アドレス空間は、ホーム・アドレス空間制御要素 ( H A S C E ) によって変換されるホーム仮想アドレスから成る。一次 A S C E 及び二次 A S C E は、それぞれ制御レジスタ 1 及び 7 の中にある。A R 指定 A S C E は、制御レジスタ 1 及び 7 の中にある場合もあり、又は、A S N 第 2 テーブル・エントリと呼ばれるテーブル・エントリの中にある場合もある。H A S C E は、制御レジスタ 1 3 の中にある。

#### 【 0 0 7 6 】

##### 動的アドレス変換

動的アドレス変換は、仮想アドレスを (例えば、ストレージ参照中に) 対応するメイン・メモリ・アドレス (本実施形態においては実アドレス又は絶対アドレス) に変換するプロセスである。仮想アドレスは、一次仮想アドレス、二次仮想アドレス、アクセス・レジスタ指定仮想アドレス、又はホーム仮想アドレスとすることができる。これらのアドレスは、それぞれ、P A S C E、S A S C E、A R 指定 A S C E、又は H A S C E によって変換される。適切な A S C E を選択した後の変換プロセスは、4 つのタイプの仮想アドレスの全てについて同一である。

#### 【 0 0 7 7 】

##### アドレス指定変換モード

実効アドレスは、動的アドレス変換又はプリフィックス付加によるいずれかの変形が行われる前に存在するアドレス (仮想アドレス) である。動的アドレス変換を制御するプログラム・ステータス・ワード内の 3 つのビットは、D A T モード・ビットであるビット 5 と、アドレス空間制御ビットであるビット 1 6 及び 1 7 である。D A T モード・ビットが 0 である場合、D A T はオフであり、C P U は実モードにある。D A T モード・ビットが 1 の場合、D A T はオンであり、C P U はアドレス空間制御ビットによって指示される変換モードにある。即ち、二進数字 0 0 は一次空間モードを指示し、二進数字 0 1 はアクセス・レジスタ・モードを指示し、二進数字 1 0 は二次空間モードを指示し、二進数字 1 1 はホーム空間モードを指示する。種々のモードを、各モードにおけるアドレスの取り扱いと共に以下に示す。

#### 【表 7】

| P S W ビット |     |     | D A T | モード           | アドレスの取り扱い |          |
|-----------|-----|-----|-------|---------------|-----------|----------|
| 5         | 1 6 | 1 7 |       |               | 命令アドレス    | 論理アドレス   |
| 0         | 0   | 0   | オフ    | 実モード          | 実         | 実        |
| 0         | 0   | 1   | オフ    | 実モード          | 実         | 実        |
| 0         | 1   | 0   | オフ    | 実モード          | 実         | 実        |
| 0         | 1   | 1   | オフ    | 実モード          | 実         | 実        |
| 1         | 0   | 0   | オン    | 一次空間モード       | 一次仮想      | 一次仮想     |
| 1         | 0   | 1   | オン    | アクセス・レジスタ・モード | 一次仮想      | A R 指定仮想 |
| 1         | 1   | 0   | オン    | 二次空間モード       | 一次仮想      | 二次仮想     |
| 1         | 1   | 1   | オン    | ホーム空間モード      | ホーム仮想     | ホーム仮想    |

##### 変換モード

#### 【 0 0 7 8 】

プログラム・ステータス・ワードは 1 2 8 ビット・ワードであり、これは、一部が、アドレス指定モードを示す 2 ビットを与える。一実施形態において、ビット 3 1 は拡張アドレス指定モード ( E A ) ビットであり、ビット 3 2 はベース・アドレス指定モード ( B A ) ビットである。これらの 2 つのビットは、アドレスのサイズを示す。これらの 2 つのビットの各々の状態は、二進数 ( 1 又は 0 ) である。E A ビットが 0 であり、且つ B A ビットが 0 であるならば、2 4 ビット・アドレス指定が示される。2 4 ビット・アドレス指定

が示される場合、64ビット・ワード(64ビット・エンティティは一般に倍長語と呼ばれる)のビット40-63が、アドレスが配置される位置である。命令アドレスが128ビット・エンティティ(4倍長語)の2番目の64ビットを占める場合、プログラム・ステータス・ワードにおけるビット位置は以下の通りである。24ビット・モードでは、命令アドレスは、プログラム・ステータス・ワードのビット104-127に置かれる。31ビット・モードでは、命令アドレスは、プログラム・ステータス・ワードのビット97-127に置かれる。64ビット・モードでは、命令アドレスは、プログラム・ステータス・ワードのビット64-127に置かれる。EAビットが0であり、且つBAビットが1であるならば、31ビット・アドレス指定が示される。適切な64ビット・ワードは、ビット位置33-63に配置された31ビット・アドレスを含む。EAビットが1であり、且つBAビットが1であるならば、64ビット・ワードのビット0-63、即ち64ビット全体がアドレスを含む。そうでない場合には、例外条件が示される。一旦、アドレス指定モードが取得されると、ASCEを決定する必要がある。

#### 【0079】

アドレス空間制御要素(ASCE)

ここで、プログラム・ステータス・ワードを用いて仮想アドレスの動的アドレス変換のための実効アドレス空間制御要素(ASCE)を決定する方法の一実施形態を示す、図3を参照する。ASCEは、例えば、2ギガバイト(ギガ=2<sup>30</sup>)のアドレス空間を指定することができる。又は、ASCEは、例えば、4テラバイト(テラ=2<sup>40</sup>)、8ペタバイト(ペタ=2<sup>50</sup>)、又は16エクサバイト(エクサ=2<sup>60</sup>)のアドレス空間を指定することができる。又は、ASCEは、実空間指示を指定することができる。実空間指示は、仮想アドレスを、1つ又は複数のアドレス変換テーブルを参照することなく、ストレージ内で実アドレスとして扱うようにさせる。

#### 【0080】

プログラム・ステータス・ワード300は、変換(T)ビット302及びアドレス空間(AS)ビット304を含む。306において、変換(T)ビットが0であれば、アドレスは実アドレス326である。308において、アドレス空間(AS)が0に等しければ(二進数字00)、この仮想アドレスのための実効ASCEは、一次アドレス空間制御要素(PASCE)310である。312において、アドレス空間(AS)が1に等しければ(二進数字01)、実効ASCEは、アクセス・レジスタ指定アドレス空間制御要素314である。316において、アドレス空間(AS)が2に等しければ(二進数字10)、実効ASCEは、二次アドレス空間制御要素(SASCE)318である。それ以外の場合には、アドレス空間(AS)は3に等しく(二進数字11)、実効ASCEはホーム・アドレス空間制御要素(HASCE)322である。

#### 【0081】

実効ASCEを選択した後の動的アドレス変換プロセスは、好ましくは、4つのタイプの仮想アドレスの全てについて同一である。

#### 【0082】

セグメント・テーブル指示又は領域テーブル指示は、オペレーティング・システムによって実ストレージ又は絶対ストレージ内に確立されたテーブルによる変換を実行させる。実空間指示は、ストレージ内のテーブルを用いることなく、仮想アドレスを単に実アドレスとして扱わせる。

#### 【0083】

セグメント・テーブル指示又は領域テーブル指示を用いる場合の変換プロセスにおいて、3つのタイプの情報単位、即ち、領域、セグメント、及びページが認識される。領域は、2ギガバイトにわたり、且つ2ギガバイト境界から始まる逐次的な仮想アドレス・ブロックである。セグメントは、1ギガバイトにわたり、且つ1ギガバイト境界から始まる逐次的な仮想アドレス・ブロックである。ページは、4キロバイトにわたり、且つ4キロバイト境界から始まる逐次的な仮想アドレス・ブロックである。

#### 【0084】

10

20

30

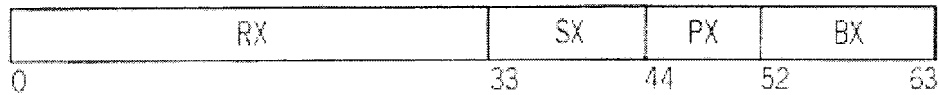
40

50

## 仮想アドレスのフォーマット

仮想アドレスの変換は、実アドレス又は絶対アドレスを取得するために、変換テーブル階層構造の複数の変換テーブルを参照することを伴うことができる。実アドレスに対してさらにプリフィックス付加を行って、絶対アドレスを形成することができる。仮想アドレスは、変換テーブル階層構造の変換テーブル内のエントリに対するインデックスを含む。仮想アドレスは、従って、4つの主要フィールドに分割される。ビット0 - 32は領域インデックス(RX)と呼ばれ、ビット33 - 43はセグメント・インデックス(SX)と呼ばれ、ビット44 - 51はページ・インデックスと呼ばれ、ビット52 - 63はバイト・インデックス(BX)と呼ばれる。一実施形態において、仮想アドレスは以下のフォーマットを有する。

【表 8】

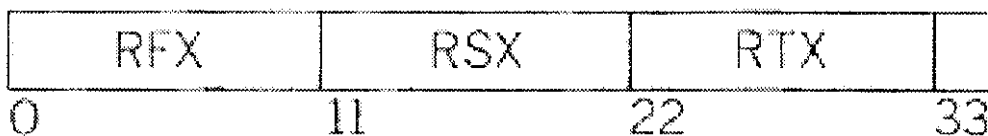


## 仮想アドレスのフォーマット

【 0 0 8 5 】

仮想アドレス空間は、その A S C E によって決定される通りに、1つの領域から成る2ギガバイトの空間とすることができ、又は最大8ギガバイト領域から成る最大で16エクサバイトまでの空間とすることができる。2ギガバイトのアドレス空間に適用される仮想アドレスの R X 部分は、全て0でなければならない。そうでない場合には、例外が認識される。仮想アドレスの R X 部分は、それ自体が3つのフィールドに分割される。ビット0 - 10は領域第1インデックス(R F X)と呼ばれ、ビット11 - 21は領域第2インデックス(R S X)と呼ばれ、ビット22 - 32は領域第3インデックス(R T X)と呼ばれる。一実施形態において、仮想アドレスのビット0 - 32は、以下のフォーマットを有する。

【表 9】



## 仮想アドレスのRXのフォーマット

【 0 0 8 6 】

R T X が最左端の有効部分 (significant part) となっている仮想アドレス (42 ビット・アドレス) は、4 テラバイト (2048 領域) をアドレス指定することができ、R S X が最左端の有効部分となっている仮想アドレス (53 ビット・アドレス) は、8 ペタバイト (4, 193, 044 領域) をアドレス指定することができ、R F X が最左端の有効部分となっている仮想アドレス (64 ビット・アドレス) は、16 エクサバイト (8, 589, 934, 592 領域) をアドレス指定することができる。

【 0 0 8 7 】

R X が 0 である仮想アドレスは、2 つの変換テーブル、即ちセグメント・テーブル及びページ・テーブルによって実アドレスに変換することができる。E D A T ファシリティがイネーブルであれば、変換はセグメント・テーブルのみで完了させることができる。R F X は非 0 であってもよく、その場合は、領域第 1 テーブル、領域第 2 テーブル、及び領域第 3 テーブルが必要とされる。R F X が 0 である場合でも、R S X が非 0 であってもよく、その場合、領域第 2 テーブル及び領域第 3 テーブルが必要とされる。R F X 及び R S X が 0 である場合でも、R T X は非 0 であってもよく、その場合、領域第 3 テーブルが必要とされる。

## 【 0 0 8 8 】

あるアドレス空間のための A S C E が、そのアドレス空間に対する参照を変換するために必要とされる最も高レベルのテーブル（領域第 1 テーブルから始まり、下位に向けてセグメント・テーブルまで続く）を指示していない場合、例外が認識される。

## 【 0 0 8 9 】

仮想アドレスの動的変換

ここで、図 3 において決定された実効 A S C E を用いて、仮想アドレスの変換に用いられる変換テーブル階層構造内の最初の変換テーブルを決定する一実施形態を示す、図 4 を参照する。

## 【 0 0 9 0 】

一実施形態において、制御レジスタ 1（C R 1）は P A S C E を含む。制御レジスタ 7（C R 7）は S A S C E を含む。制御レジスタ 13（C R 13）は H A S C E を含み、アクセス・レジスタ変換（A R T）プロセスによって導出されるアドレス空間第 2 テーブル・エントリ（A S T E）はアクセス・レジスタ指定アドレス空間制御要素を含む。実効 A S C E 4 0 0 は、これらの位置のうちの 1 つから選択される。

## 【 0 0 9 1 】

実効 A S C E 4 0 0 の第 1 の部分は、領域第 1 テーブル、領域第 2 テーブル、領域第 3 テーブル、又はセグメント・テーブルのうちのいずれかを指示する基点アドレスを含んだテーブル基点 4 0 2 を含む。テーブル基点（ビット 0 . . . 5 1）に対して 1 2 個の二進数字 0 が付加されると、仮想アドレスの変換に用いられる変換テーブル階層構造の中で最も高位の変換テーブルの 6 4 ビットの基点アドレスが形成される。実効 A S C E 4 0 0 はさらに、実空間制御（R）ビット 4 0 4 及び D T ビット 4 0 6 も含む。実空間制御（R）ビットが 0 の場合、どの特定の基点アドレスがテーブル基点 4 0 2 であるのかを判定するために D T がセレクタ 4 0 8 によってデコードされる。D T ビットが 3（二進数字 1 1）に等しい場合、テーブル基点 4 0 2 は領域第 1 テーブル 4 1 0 を指示する。D T ビットが 2（二進数字 1 0）に等しい場合、テーブル基点 4 0 2 は領域第 2 テーブル 4 1 2 を指示する。D T ビットが 1（二進数字 0 1）に等しい場合、テーブル基点 4 0 2 は領域第 3 テーブル 4 1 4 を指示する。そうではなくて、D T ビットが 0（二進数字 0 0）に等しい場合、テーブル基点 4 0 2 はセグメント・テーブル 4 1 6 を指示する。

## 【 0 0 9 2 】

領域第 1 テーブル、領域第 2 テーブル、又は領域第 3 テーブルは、単に領域テーブルと呼ばれることもある。同様に、領域第 1 テーブル指示、領域第 2 テーブル指示、又は領域第 3 テーブル指示は、領域テーブル指示と呼ばれることもある。領域テーブル、セグメント・テーブル、及びページ・テーブルは、実ストレージの現在の割り当てを反映する。ページは、仮想ストレージの割り当てに対して用いられる用語である。実ストレージは固定ブロック単位で割り振られる。ページは、逐次的な仮想アドレスの組に対して割り当てられた場合でさえも、実ストレージ内で隣接している必要はない。

## 【 0 0 9 3 】

変換に用いられる A S C E が領域第 1 テーブル指示である場合には、変換プロセスは、例えば、領域第 1 テーブル、領域第 2 テーブル、領域第 3 テーブル、セグメント・テーブル、及び随意にページ・テーブルを用いたマルチ・レベル・ルックアップから成る。これらのテーブルは、実ストレージ又は絶対ストレージ内に存在する。A S C E が、領域第 2 テーブル指示、領域第 3 テーブル指示、又はセグメント・テーブル指示である場合、指示されたレベルより高位のレベルのテーブルにおけるルックアップは省かれ、より高位レベルのテーブル自体が省かれる。

## 【 0 0 9 4 】

ここで、変換テーブル階層構造を用いた仮想アドレスの動的アドレス変換の一実施形態を示す、図 5 を参照する。

## 【 0 0 9 5 】

図 4 の実効 A S C E 4 0 0 は、指示タイプ（D T）ビット 4 0 6 を含む。A S C E の実

10

20

30

40

50

空間制御 ( R ) ビット 4 0 4 が 0 の場合、テーブル基点 4 0 2 がどの基点アドレスを指示するのかを判定するために、D T ビットはセクタ 4 0 8 によってデコードされる。実空間制御 ( R ) ビットが 1 の場合、動的アドレス変換は、図 6 のノード D 5 6 4 において示されるようにして行われる。

#### 【 0 0 9 6 】

セクタ 4 0 8 において、D T ビットが 3 ( 二進数字 1 1 ) に等しい場合、変換テーブル階層構造において指示される最初のテーブルは、領域第 1 テーブルである。テーブル基点 4 0 2 に対して、5 0 2 において、仮想アドレスの領域第 1 インデックス ( R F X ) 5 0 8 部分が算術的に加算され、領域第 1 テーブル内の領域第 1 テーブル・エントリ 5 0 6 が参照される。テーブル基点 ( 右側に 1 2 個の 0 が付加されたもの、又は 4 0 9 6 を乗じたもの ) が、インデックスに 8 を乗じた積 ( 又は右側に 3 つの 0 が付加されたインデックス ) に対して加算される。領域第 1 テーブル・エントリは、変換に用いられる変換テーブル階層構造の中で次に下位のテーブルに対する領域第 2 テーブル基点 5 0 4 を含む。領域第 1 テーブルの次に下位のテーブルは、領域第 2 テーブルである。領域第 1 テーブル・エントリの無効 ( I ) ビットが 1 に等しい場合には、その領域第 1 テーブル・エントリは無効であり、変換に用いることはできない。例外条件が示される。

#### 【 0 0 9 7 】

セクタ 4 0 8 において、D T ビットが 2 ( 二進数字 1 0 ) に等しい場合、変換テーブル階層構造において指示される最初のテーブルは、領域第 2 テーブルである。テーブル基点 4 0 2 に対して、5 1 0 において、仮想アドレスの領域第 2 インデックス ( R S X ) 5 1 6 部分が算術的に加算され、領域第 2 テーブル内の領域第 2 テーブル・エントリ 5 1 4 が参照される。テーブル基点 ( 右側に 1 2 個の 0 が付加されたもの、又は 4 0 9 6 を乗じたもの ) が、インデックスに 8 を乗じた積 ( 又は右側に 3 つの 0 が付加されたインデックス ) に対して加算される。領域第 2 テーブル・エントリは、変換に用いられる変換テーブル階層構造の中で次に下位のテーブルに対する領域第 3 テーブル基点 5 1 2 を含む。領域第 2 テーブルの次に下位のテーブルは、領域第 3 テーブルである。領域第 2 テーブル・エントリの無効 ( I ) ビットが 1 に等しい場合には、その領域第 2 テーブル・エントリは無効であり、例外条件が示される。

#### 【 0 0 9 8 】

セクタ 4 0 8 において、D T ビットが 1 ( 二進数字 ( 0 1 ) に等しい場合、変換テーブル階層構造において指示される最初のテーブルは、領域第 3 テーブルである。テーブル基点 4 0 2 に対して、5 1 8 において、仮想アドレスの領域第 3 インデックス ( R T X ) 5 2 4 部分が算術的に加算され、領域第 3 テーブル内の領域第 3 テーブル・エントリ 5 2 2 が参照される。テーブル基点 ( 右側に 1 2 個の 0 が付加されたもの、又は 4 0 9 6 を乗じたもの ) が、インデックスに 8 を乗じた積 ( 又は、右側に 3 つの 0 が付加されたインデックス ) に対して加算される。領域第 3 テーブル・エントリは、変換に用いられる変換テーブル階層構造の中で次に下位のテーブルに対するセグメント・テーブル基点 5 2 0 を含む。領域第 3 テーブルの次に下位のテーブルは、セグメント・テーブルである。領域第 3 テーブル・エントリの無効 ( I ) ビットが 1 に等しい場合には、その領域第 3 テーブル・エントリは無効であり、例外条件が示される。

#### 【 0 0 9 9 】

セクタ 4 0 8 において、D T ビットが 0 ( 二進数字 ( 0 0 ) に等しい場合、変換テーブル階層構造において指示される最初のテーブルは、セグメント・テーブルである。テーブル基点 4 0 2 に対して、5 2 6 において、仮想アドレスのセグメント・インデックス ( S X ) 5 3 2 部分が算術演算的に加算され、セグメント・テーブル内のセグメント・テーブル・エントリ 5 3 0 が参照される。テーブル基点 ( 右側に 1 2 個の 0 が付加されたもの、又は 4 0 9 6 を乗じたもの ) が、インデックスに 8 を乗じた積 ( 又は右側に 3 つの 0 が付加されたインデックス ) に対して加算される。セグメント・テーブル・エントリは、どちらも 5 2 8 で示される、ページ・テーブルに対する基点アドレスか、又はセグメント・フレーム絶対アドレス ( S F A A ) を含む。セグメント・テーブル・エントリの無効 ( I

10

20

30

40

50



ビットが1に等しい場合には、そのセグメント・テーブル・エントリは無効であり、例外条件が示される。

【0100】

538において、セグメント・テーブルのSTEフォーマット制御(FC)ビットが検査される。STEフォーマット制御が1である場合、セグメント・テーブル・エントリ530はセグメント・フレーム絶対アドレス(SFAA)552を含み、動的アドレス変換は図7のノード562を参照して続行する。そうでない場合には、セグメント・テーブルから取得されたセグメント・テーブル・エントリはページ・テーブル基点アドレスを含み、動的アドレス変換は図6のノード560を参照して続行する。

【0101】

ここで、図6を参照する。セグメント・テーブル・エントリ内のSTEフォーマット制御が0である場合、セグメント・テーブルから取得されたセグメント・テーブル・エントリは、変換テーブル階層構造の中で次に下位のテーブルに対する基点アドレスを含む。セグメント・テーブルの次に下位のテーブルは、ページ・テーブルである。図5のセグメント・テーブル・エントリ530から取得されたページ・テーブル基点528に対して、538において、仮想アドレスのページ・インデックス(PX)534部分が算術的に加算され、ページ・テーブル内のページ・テーブル・エントリ542が参照される。ページ・テーブル・エントリは、ページ・フレーム実アドレス(PFRA)546を含む。ページ・フレーム実アドレスの最左端のビットが、548において、仮想アドレスのバイト・インデックス(BX)536部分に連結されると、64ビットの実アドレス550が得られる。64ビット実アドレスに対してさらにプリフィックス付加操作を行って、絶対アドレスを形成することができる。変換された仮想アドレスは、主ストレージ又はメモリ内の所望の4キロバイト(4096バイト)のデータ・ブロックを参照する。

【0102】

好ましくは、仮想アドレスからメモリ・アドレスへの動的アドレス変換に用いられた情報は、変換ルックアサイド・バッファ・エントリ・タグ内に、その仮想アドレスと関連付けられたメモリのブロックのアドレスと共にストアされる。その後のストレージ・アクセスは、ASC E情報及び仮想アドレス情報を変換ルックアサイド・バッファ・タグと比較することによって、仮想アドレスを迅速に変換することができる。あるタグがその仮想アドレスのものであることが分かれば、関与する各変換テーブルへの時間のかかる逐次的なアクセスを実行する代わりに、そのメモリ・ブロックの変換ルックアサイド・バッファ・アドレスを用いることができる。一実施形態において、ページ・フレーム実アドレス(PFRA)は、例えば、ASC Eと仮想アドレスのRX、SX及びPX部分とから成るタグと共に、変換ルックアサイド・バッファ544のエントリ内にストアされる。この仮想アドレスのその後の変換は、これ以降、変換ルックアサイド・バッファ内にストアされた情報から導出される。

【0103】

ここで、図5Cを参照する。セグメント・テーブル・エントリ530内のSTEフォーマット制御が1である場合、セグメント・テーブル・エントリは、セグメント・フレーム絶対アドレス(SFAA)552を含む。セグメント・フレーム絶対アドレスの最左端のビットが、554において、仮想アドレスのページ・インデックス534部分及びバイト・インデックス536部分に連結されると、64ビットの絶対アドレス556が得られる。変換された仮想アドレスは、主ストレージ又はメモリ内の所望の大データ・ブロックを参照する。大データ・ブロックは少なくとも1メガバイト(1,048,576バイト)のサイズである。

【0104】

一実施形態において、セグメント・フレーム絶対アドレス(SFAA)は、仮想アドレスのRX及びSX部分と共に、変換ルックアサイド・バッファ544内にストアされる。この仮想アドレスのその後の変換は、これ以降、変換ルックアサイド・バッファ内にストアされた情報から導出される。

10

20

30

40

50

## 【 0 1 0 5 】

## 変換テーブル・エントリのフォーマット

変換に用いられる変換テーブル階層構造における種々の変換テーブル・エントリの実施形態は以下の通りである。

## 【 0 1 0 6 】

## 領域テーブル・エントリ

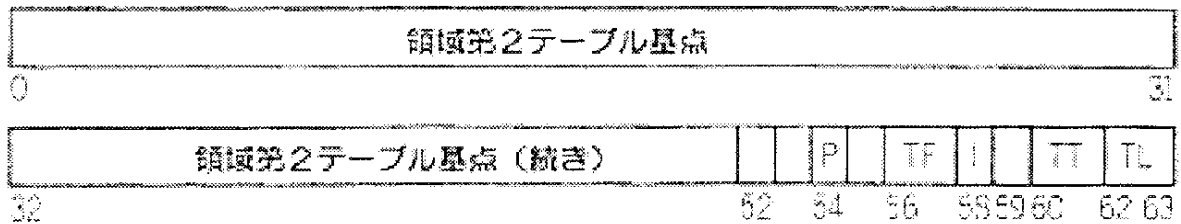
「領域テーブル・エントリ」という用語は、領域第1テーブル・エントリ、領域第2テーブル・エントリ、又は領域第3テーブル・エントリを意味する。領域第1テーブル、領域第2テーブル、及び領域第3テーブルからフェッチされたエントリは、以下のフォーマットを有する。あるエントリを含むテーブルのレベル（第1、第2、又は第3）は、エントリ内のテーブル・タイプ（TT）ビットによって識別される。

## 【 0 1 0 7 】

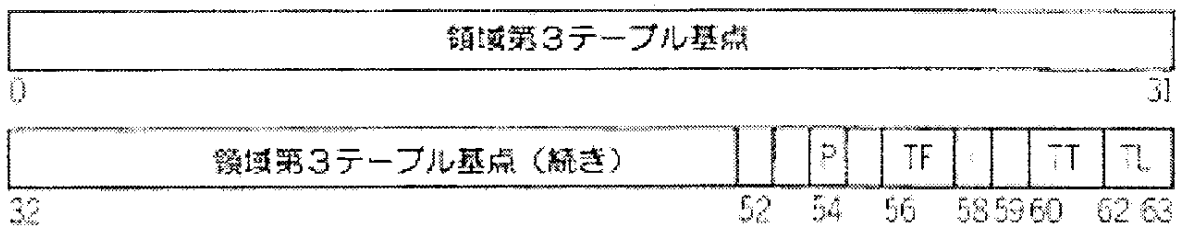
一実施形態において、領域第1テーブル・エントリ、領域第2テーブル・エントリ、及び領域第3テーブル・エントリのフォーマットは以下の通りである。

## 【表 1 0】

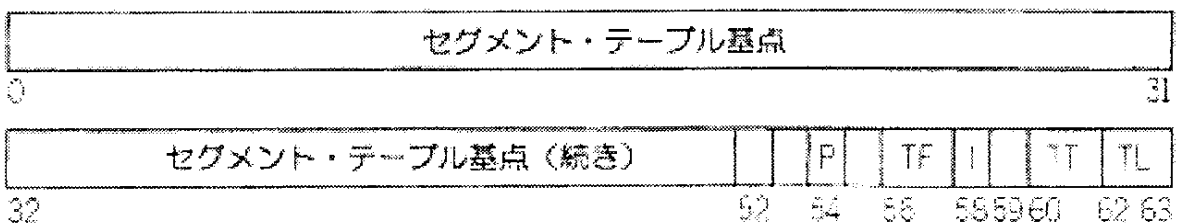
## 領域第1テーブル・エントリ（TT=11）



## 領域第2テーブル・エントリ（TT=10）



## 領域第3テーブル・エントリ（TT=01）



## 領域テーブル・エントリのフォーマット

## 【 0 1 0 8 】

領域第2テーブル基点、領域第3テーブル基点、及びセグメント・テーブル基点：領域第1テーブル・エントリは、領域第2テーブル基点を含む。領域第2テーブル・エントリは、領域第3テーブル基点を含む。領域第3テーブル・エントリは、セグメント・テーブル基点を含む。以下の説明は3つのテーブル基点の各々に対して適用される。エントリのビット0 - 51は、右に12個の0を付加すると、次に下位レベルのテーブルの開始を指

示する 64 ビットのアドレスを形成する。

【0109】

DAT 保護ビット (P) : 強化型 DAT が適用される場合には、ビット 54 は、変換に用いられる後続の各領域テーブル・エントリ、セグメント・テーブル・エントリ、及び、適用可能な場合にはページ・テーブル・エントリの中の DAT 保護ビットと OR (論理和) 演算されたものとして扱われる。それゆえ、そのビットが 1 であるときには、DAT 保護は、領域全体又は領域テーブル・エントリによって指定される領域に対して適用される。強化型 DAT ファシリティがインストールされていない場合、又はファシリティはインストールされているが強化型 DAT イネーブルメント制御が 0 である場合には、領域テーブル・エントリのビット 54 は無視される。

10

【0110】

領域第 2 テーブル・オフセット、領域第 3 テーブル・オフセット、及びセグメント・テーブル・オフセット (TF) : 領域第 1 テーブル・エントリは、領域第 2 テーブル・オフセットを含む。領域第 2 テーブル・エントリは、領域第 3 テーブル・オフセットを含む。領域第 3 テーブル・エントリは、セグメント・テーブル・オフセットを含む。以下の説明は、3 つのテーブル・オフセットの各々に対して適用される。エントリのビット 56 及び 57 は、次に下位レベルのテーブルの中で、テーブルの始めのところで欠落している部分の長さを指定するものであり、即ち、このビットは、次に下位レベルのテーブル内で実際に存在する最初のエントリの位置を指定する。このビットは、欠落部分の長さを 4,096 バイト単位で指定するので、従って、欠落部分の長さは 512 エントリの倍数として可変になる。4,096 バイト単位で表された欠落部分の長さは、TF の値に等しい。オフセット・フィールドの内容を、長さフィールドであるビット 62 及び 63 と組み合わせて用いて、次に下位レベルのテーブルによって変換される仮想アドレスの部分 (RSX、RTX、又は SX) がテーブル内に実際に存在するエントリを指示するのかが確立される。

20

【0111】

領域無効ビット (I) : 領域第 1 テーブル・エントリ又は領域第 2 テーブル・エントリ内のビット 58 は、そのエントリに関連付けられた領域の組が利用可能かどうかを制御する。領域第 3 テーブル・エントリ内のビット 58 は、エントリに関連付けられた単一の領域が利用可能であるかどうかを制御する。ビット 58 が 0 である場合には、アドレス変換は領域テーブル・エントリを用いることによって進行する。ビットが 1 である場合には、そのエントリを変換に用いることはできない。

30

【0112】

テーブル・タイプ・ビット (TT) : 領域第 1 テーブル・エントリ、領域第 2 テーブル・エントリ、又は領域第 3 テーブル・エントリのビット 60 及び 61 は、そのエントリを含むテーブルのレベルを以下のように識別する。ビット 60 及び 61 は、変換に用いられる ASCE であるテーブル指示のタイプ、及び、それまでに用いられたテーブル・レベルの数を考慮して、正しいテーブル・レベルを識別しなければならない。そうでない場合には、変換指定例外 (translation specification exception) が認識される。以下のテーブルはテーブル・タイプ・ビットを示す。

40

【表 1 1】

| ビット 6 0<br>及び 6 1 | 領域テーブル<br>・レベル |
|-------------------|----------------|
| 1 1               | 第 1            |
| 1 0               | 第 2            |
| 0 1               | 第 3            |

### 領域テーブル・エントリのための テーブル・タイプ・ビット

10

## 【0 1 1 3】

領域第 2 テーブル長、領域第 3 テーブル長、及びセグメント・テーブル長 (TL) : 領域第 1 テーブル・エントリは、領域第 2 テーブル長を含む。領域第 2 テーブル・エントリは、領域第 3 テーブル長を含む。領域第 3 テーブル・エントリは、セグメント・テーブル長を含む。以下の説明は、3 つのテーブル長の各々に対して適用される。エントリのビット 6 2 及び 6 2 は、次に下位レベルのテーブルの長さを 4 , 0 9 6 バイト単位で指定するので、従って、テーブルの長さは 5 1 2 エントリの倍数として可変になる。4 , 0 9 6 バイト単位で表された次に下位レベルのテーブルの長さは、TL の値より 1 大きい。長さフィールドの内容を、オフセット・フィールドであるビット 5 6 及び 5 7 と組み合わせて用いて、次に下位レベルのテーブルによって変換される仮想アドレスの部分 (RSX、RTX、又は SX) がテーブル内に実際に存在するエントリを指示するかどうかを確立する。領域テーブル・エントリの他の全てのビット位置は、可能性のある将来的な拡張のために未使用とされ、0 を含むべきである。そうしなければ、プログラムが将来、互換的に動作できなくなるおそれがある。強化型 DAT が適用される場合、領域テーブル・エントリ内の未使用のビット位置は、そのテーブル・エントリがたとえ無効だとしても、0 を含むべきである。

20

## 【0 1 1 4】

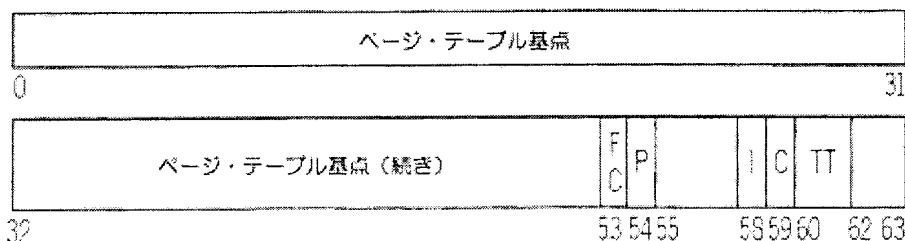
## セグメント・テーブル・エントリ

強化型 DAT が適用されない場合、又は強化型 DAT は適用されるが、STE フォーマット制御、即ちセグメント・テーブル・エントリのビット 5 3 が 0 である場合には、セグメント・テーブルからフェッチされるエントリは、一実施形態において、以下のフォーマットを有する。

30

【表 1 2】

セグメント・テーブル・エントリ (T=00、FC=0)



40

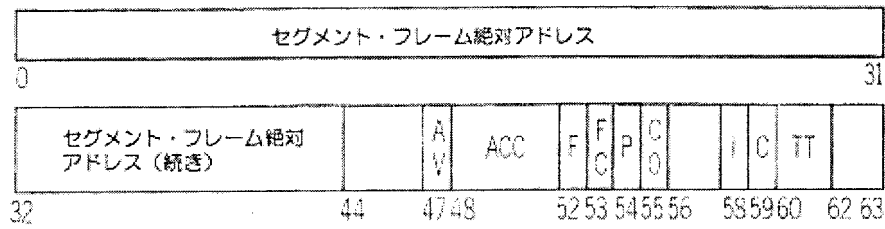
## セグメント・テーブル・エントリのフォーマット I

## 【0 1 1 5】

強化型 DAT が適用され、且つ STE フォーマット制御が 1 である場合には、セグメント・テーブルからフェッチされるエントリは、一実施形態において、以下のフォーマットを有する。

【表 1 3】

セグメント・テーブル・エントリ (T=00, FC=1)



セグメント・テーブル・エントリのフォーマット I I

10

## 【 0 1 1 6】

セグメント・テーブル・エントリ内の選択されたフィールドは、以下のように割り振られる。

ページ・テーブル基点：強化型 D A T が適用されない場合、又は強化型 D A T は適用されるが、S T E フォーマット制御、即ちセグメント・テーブル・エントリのビット 5 3 が 0 である場合には、ビット 0 - 5 2 は、右に 1 1 個の 0 を付加すると、ページ・テーブルの開始を示す 6 4 ビットのアドレスを形成する。アドレスが実であるか絶対であるかは予測不可能である。

## 【 0 1 1 7】

セグメント・フレーム絶対アドレス (S F A A)：強化型 D A T が適用され、且つ S T E フォーマット制御が 1 である場合には、エントリのビット 0 - 4 3 は、右に 2 0 個の 0 を付加すると、セグメントの 6 4 ビット絶対アドレスを形成する。

20

## 【 0 1 1 8】

A C C F 妥当性制御 (A V)：強化型 D A T が適用され、且つ S T E フォーマット制御が 1 である場合には、ビット 4 7 はアクセス制御ビット及びフェッチ保護ビット (A C C F) 妥当性制御である。A V 制御が 0 である場合、セグメント・テーブル・エントリのビット 4 8 - 5 2 は無視される。A V 制御が 1 である場合、ビット 4 8 - 5 2 は以下で説明されるように用いられる。

## 【 0 1 1 9】

アクセス制御ビット (A C C)：強化型 D A T が適用され、S T E フォーマット制御が 1 であり、且つ、A V 制御が 1 である場合には、セグメント・テーブル・エントリのビット 4 8 - 5 1 は、そのアドレスに適用されるいずれのキー制御アクセス・チェックにも用いることができるアクセス制御ビットを含む。

30

## 【 0 1 2 0】

フェッチ保護ビット (F)：強化型 D A T が適用され、S T E フォーマット制御が 1 であり、且つ、A V 制御が 1 である場合には、セグメント・テーブル・エントリのビット 5 2 は、そのアドレスに適用されるいずれのキー制御アクセス・チェックにも用いることができるフェッチ保護ビットを含む。

## 【 0 1 2 1】

S T E フォーマット制御 (F C)：強化型 D A T が適用される場合には、ビット 5 3 は以下のように、セグメント・テーブル・エントリのためのフォーマット制御である。即ち、

40

## 【 0 1 2 2】

F C ビットが 0 である場合、エントリのビット 0 - 5 2 はページ・テーブル基点を形成し、ビット 5 5 は未使用とされる。

## 【 0 1 2 3】

F C ビットが 1 である場合、エントリのビット 0 - 4 3 はセグメント・フレーム絶対アドレスを形成し、ビット 4 7 は A C C F 妥当性制御であり、ビット 4 8 - 5 1 はアクセス制御ビットであり、ビット 5 2 はフェッチ保護ビットであり、ビット 5 5 は変更記録オーバーライドである。強化型 D A T が適用されない場合には、ビット 5 3 は無視される。

50

## 【 0 1 2 4 】

D A T保護ビット ( P ) : ビット 5 4 は、 1 である場合には、 D A T 保護がセグメント全体に適用されることを示す。

## 【 0 1 2 5 】

強化型 D A T が適用されない場合には、ビット 5 4 は、変換に用いられるページ・テーブル・エントリの中の D A T 保護ビットと論理和 ( O R ) 演算されたものとして扱われる。

## 【 0 1 2 6 】

強化型 D A T が適用される場合には、変換に用いられるいずれかの及び全ての領域テーブル・エントリ内の D A T 保護ビットは、セグメント・テーブル・エントリ内の D A T 保護ビットと論理和演算されたものとして扱われ、 S T E フォーマット制御が 0 である場合には、 S T E の中の D A T 保護ビットはさらに、ページ・テーブル・エントリ内の D A T 保護ビットと論理和演算されたものとして扱われる。

10

## 【 0 1 2 7 】

変更記録オーバーライド ( C O ) : 強化型 D A T が適用され、且つ、 S T E フォーマット制御が 1 である場合、セグメント・テーブル・エントリのビット 5 5 は、そのセグメントについての変更記録オーバーライドである。強化型 D A T が適用されない場合、又は、強化型 D A T は適用されるが S T E フォーマット制御が 0 である場合には、セグメント・テーブル・エントリのビット 5 5 は無視される。

## 【 0 1 2 8 】

セグメント無効ビット ( I ) : ビット 5 8 は、そのセグメント・テーブル・エントリに関連付けられたセグメントが利用可能であるかどうかを制御する。

20

## 【 0 1 2 9 】

このビットが 0 である場合、アドレス変換はそのセグメント・テーブル・エントリを用いることによって進行する。

## 【 0 1 3 0 】

このビットが 1 である場合、そのセグメント・テーブル・エントリを変換に用いることができない。

## 【 0 1 3 1 】

共通セグメント・ビット ( C ) : ビット 5 9 は、セグメント・テーブル・エントリの変換ルックアサイド・バッファ・コピーの使用を制御する。強化型 D A T が適用されない場合、又は、強化型 D A T は適用されるがフォーマット制御が 0 である場合には、ビット 5 9 は、そのセグメント・テーブル・エントリによって指示されるページ・テーブルの変換ルックアサイド・バッファ・コピーの使用も制御する。

30

## 【 0 1 3 2 】

0 はプライベート・セグメントを識別し、この場合、セグメント・テーブル・エントリ及びそれが指示するいずれのページ・テーブルも、そのセグメント・テーブル・エントリが存在するセグメント・テーブルを指示するセグメント・テーブル基点との関連でのみ利用することができる。

## 【 0 1 3 3 】

1 は共通セグメントを識別し、この場合、セグメント・テーブル・エントリ及びそれが指示するいずれのページ・テーブルも、異なるセグメント・テーブルが指定されたとしても、引き続き、そのセグメント・インデックスに対応するアドレスを変換するために用いることができる。

40

## 【 0 1 3 4 】

しかしながら、共通セグメントのための、セグメント・テーブル・エントリ及びいずれかのページ・テーブルの変換ルックアサイド・バッファ・コピーは、変換に用いられる A S C E の中のプライベート空間制御であるビット 5 5 が 1 である場合、又は、その A S C E が実空間指示である場合には、使用可能ではない。用いられている A S C E の中のプライベート空間制御が 1 である場合に、セグメント・テーブル・エントリが変換中にストレ

50

ージからフェッチされたのであれば、その共通セグメント・ビットは0でなければならない。そうでない場合には、変換指定例外が認識される。

【0135】

テーブル・タイプ・ビット(TT)：セグメント・テーブル・エントリのビット60及び61は、そのエントリを含むテーブルのレベルを識別するための二進数字00である。領域テーブル・エントリ又はセグメント・テーブル・エントリの中のビット60及び61の可能な全ての値の意味は、以下の通りである。

【表14】

| ビット60<br>及び61 | テーブル・レベル |
|---------------|----------|
| 11            | 領域第1     |
| 10            | 領域第2     |
| 01            | 領域第3     |
| 00            | セグメント    |

テーブル・タイプ・ビット60、61

【0136】

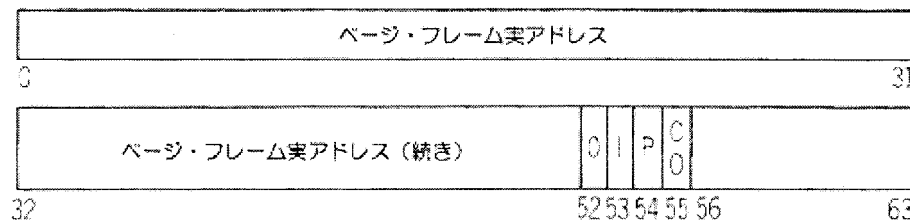
ビット60及び61は、変換に用いられるASCEであるテーブル指示のタイプ、及び、それまでに用いられたテーブル・レベルの数を考慮して、正しいテーブル・レベルを識別するものでなければならない。そうでない場合には、変換指定例外が認識される。セグメント・テーブル・エントリの他の全てのビット位置は、可能性のある将来的な拡張のために未使用とされ、0を含むべきである。そうしなければ、プログラムが将来、互換的に動作できなくなるおそれがある。強化型DATが適用される場合、セグメント・テーブル・エントリ内の未使用のビット位置は、そのテーブル・エントリがたとえ無効だとしても、0を含むべきである。

【0137】

ページ・テーブル・エントリ

ー実施形態において、ページ・テーブルからフェッチされるエントリは、以下のフォーマットを有する。

【表15】



ページ・テーブル・エントリのフォーマット

【0138】

ページ・テーブル・エントリの選択されたフィールドは、以下のように割り振られる。即ち、

ページ・フレーム実アドレス(PFRA)：ビット0-51は、実ストレージのアドレスの最左端のビットを与える。これらのビットの右に仮想アドレスの12ビット・バイトのインデックス・フィールドを連結すると、64ビットの実アドレスが得られる。

【0139】

ページ無効ビット(I)：ビット53は、ページ・テーブル・エントリに関連付けられたページが利用可能であるかどうかを制御する。ビットが0である場合、アドレス変換はそのページ・テーブル・エントリを用いることによって進行する。ビットが1である場合

10

20

30

40

50

、そのページ・テーブル・エントリを変換に用いることはできない。

【 0 1 4 0 】

D A T 保護ビット ( P ) : ビット 5 4 は、そのページでストア・アクセスを行うことができるかどうかを制御する。この保護機構は、キー制御保護及び低いアドレス保護機構に加えてのものである。このビットは、フェッチ・アクセスに対して影響を与えない。このビットが 0 である場合、以下の追加的制約に従って、そのページへのストアが許可される。即ち、

変換に用いられるセグメント・テーブル・エントリ内の D A T 保護ビットが 0 である。

強化型 D A T が適用される場合には、変換に用いられる全ての領域テーブル・エントリ内の D A T 保護ビットが 0 である。

10

【 0 1 4 1 】

他の保護機構

ビットが 1 である場合、ストアは許可されない。より優先順位の高い例外条件が存在しない場合には、D A T 保護ビットが 1 であるときにストアしようとする試行は、保護例外の認識を発生させる。セグメント・テーブル・エントリ内の D A T 保護ビットは、そのページに対して D A T 保護が適用されるかどうかを判定するときに、ビット 5 4 と論理和演算されたものとして扱われる。強化型 D A T が適用される場合、変換に用いられるいずれの領域テーブル・エントリ内の D A T 保護ビットもまた、D A T 保護が適用されるかどうかを判定するときに、ビット 5 4 と論理和演算されたものとして扱われる。

20

【 0 1 4 2 】

変更記録オーバーライド ( C O ) : 強化型 D A T が適用されない場合、ページ・テーブル・エントリのビット 5 5 は、0 を含んでいなければならない。そうしなければ、アドレス変換のためにそのエントリを用いる命令の実行の一部として、変換指定例外が認識される。強化型 D A T が適用され、且つ、S T E フォーマット制御が 0 である場合、ページ・テーブル・エントリのビット 5 5 は、そのページについての変更記録オーバーライドである。

【 0 1 4 3 】

エントリのビット位置 5 2 は、0 を含んでいなければならない。そうしなければ、アドレス変換のためにそのエントリを用いる命令の実行の一部として、変換指定例外が認識される。ビット 5 6 - 6 3 は割り当てられず、無視される。

30

【 0 1 4 4 】

動的変換の別の実施形態

このセクションは、仮想アドレスを用いて主ストレージにアクセスする前に暗黙的に実施されるときの変換プロセスを説明する。

【 0 1 4 5 】

仮想アドレスの変換は、プログラム・ステータス・ワード内の D A T モード・ビット及びアドレス空間制御ビットにより制御され、且つ、制御レジスタ 1、7、及び 1 3 内の A S C E により制御され、そして、アクセス・レジスタによって指定されるように制御される。変換に用いられる A S C E が領域第 1 テーブル指示である場合には、変換は領域第 1 テーブル、領域第 2 テーブル、領域第 3 テーブル、セグメント・テーブル、及びページ・テーブルによって実施され、その全てが実ストレージ又は絶対ストレージ内に存在する。A S C E がより下位レベルのタイプのテーブル指示 ( 領域第 2 テーブル指示、領域第 3 テーブル指示、又はセグメント・テーブル指示 ) である場合には、変換は、指示されたレベルから始まるテーブル・レベルのみを用いて実施され、非 0 であればより高位の 1 つ又は複数のレベルのテーブルの使用を要求するはずの仮想アドレスのビットは、全て 0 でなければならないそうでない場合には、A S C E タイプ例外が認識される。A S C E が実空間指示である場合には、仮想アドレスは実アドレスとして扱われ、実ストレージ又は絶対ストレージの内のテーブル・エントリは用いられない。

40

【 0 1 4 6 】

特定のアドレス変換に用いられる A S C E を、実効 A S C E と呼ぶ。従って、一次仮想

50



アドレスが変換される場合には、制御レジスタ 1 の内容が実効 A S C E として用いられる。同様に、二次仮想アドレスの場合には制御レジスタ 7 の内容が用いられ、A R 指定仮想アドレスの場合にはアクセス・レジスタによって指定された A S C E が用いられ、ホーム仮想アドレスの場合には制御レジスタ 1 3 の内容が用いられる。

【 0 1 4 7 】

実効 A S C E の中の実空間制御が 0 である場合、A S C E の中の指示タイプはテーブル指示タイプ、即ち、領域第 1 テーブル、領域第 2 テーブル、領域第 3 テーブル、又はセグメント・テーブルを指定する。仮想アドレスの対応する部分（領域第 1 インデックス、領域第 2 インデックス、領域第 3 インデックス、又はセグメント・インデックス）が、指示の中のテーブル長フィールドに対してチェックされ、これを指示内の基点に加算して、指示されたテーブルの中のエントリを選択する。選択されたエントリが、指示内のテーブル長フィールドによって判定してテーブルの外部にある場合、又は選択されたエントリ内の I ビットが 1 である場合は、指示によって指定されたテーブル・レベルに応じて、領域第 1 変換、領域第 2 変換、領域第 3 変換、又はセグメント変換の例外が認識される。選択されたエントリ内のテーブル・タイプ・ビットが予期されたテーブル・レベルを示していない場合には、変換指定例外が認識される。

【 0 1 4 8 】

実効 A S C E によって選択されたテーブル・エントリは、用いられるべき次に下位レベルのテーブルを指示する。現在のテーブルが領域第 1 テーブル、領域第 2 テーブル、又は領域第 3 テーブルであれば、仮想アドレスの次の部分（それぞれ、領域第 2 インデックス、領域第 3 インデックス、又はセグメント・インデックス）が現在のテーブル・エントリ内のテーブル・オフセット及びテーブル長フィールドに対してチェックされ、これをエントリ内の基点に加算して、次に下位レベルのテーブルの中のエントリを選択する。次のテーブルの中の選択されたエントリが、現在のテーブル・エントリ内のテーブル・オフセット及びテーブル長フィールドによって判定して、そのテーブルの外部にある場合、又は、選択されたエントリ内の I ビットが 1 である場合は、次のテーブルのレベルに応じて、領域第 2 変換、領域第 3 変換、又はセグメント変換の例外が認識される。選択されたエントリ内のテーブル・タイプ・ビットが予期されたテーブル・レベルを示していない場合には、変換指定例外が認識される。

【 0 1 4 9 】

逐次的なテーブル・レベルによる仮想アドレスの部分の処理は、セグメント・テーブル・エントリが選択されるまで続く。セグメント・テーブル・エントリは、指定されたセグメントの中の全てのページに適用されるページ保護ビットを含む。

【 0 1 5 0 】

仮想アドレスのページ・インデックス部分を、セグメント・テーブル・エントリ内のページ・テーブル基点に加算して、ページ・テーブル内のエントリを選択する。ページ・テーブル・エントリ内の I ビットが 1 である場合には、ページ変換例外が認識される。ページ・テーブル・エントリは、仮想アドレスの変換を示す実アドレスの最左端のビットを含み、且つ、そのページ・テーブル・エントリによって指定されるページのみに適用されるページ保護ビットを含む。

【 0 1 5 1 】

仮想アドレスのバイト・インデックス・フィールドは、実アドレスの最右端のビット位置として、変更せずに用いられる。

【 0 1 5 2 】

実ストレージ又は絶対ストレージ内の変換テーブルを参照することに関連する遅延を排除するために、テーブルからフェッチされた情報は通常、特別なバッファである変換ルックアサイド・バッファ内にも置かれ、同じテーブル・エントリが関与するその後の変換は、変換ルックアサイド・バッファ内に記録された情報を用いて行うことができる。変換ルックアサイド・バッファは、実空間指示に関係した、仮想と実が等しい (virtual equals real) 変換を記録することもできる。

## 【 0 1 5 3 】

領域テーブル、セグメント・テーブル、又はページ・テーブルからエントリをフェッチする目的で、アドレス変換プロセス中に実ストレージ又は絶対ストレージにアクセスする場合であればいつでも、キー制御保護は適用されない。

## 【 0 1 5 4 】

A S C Eによって指示されるテーブルにおけるルックアップ

実効 A S C E の D T 制御であるビット 6 0 - 6 1 は、A S C E のテーブル指示タイプ、及び、指示されたテーブルによって変換される仮想アドレスの部分の両方を以下のように指定する。

## 【表 1 6】

10

| ビット 6 0<br>及び 6 1 | 指示タイプ      | テーブルによって変換される<br>仮想アドレス部分       |
|-------------------|------------|---------------------------------|
| 1 1               | 領域第 1 テーブル | 領域第 1 インデックス<br>(ビット 0 - 1 0)   |
| 1 0               | 領域第 2 テーブル | 領域第 2 インデックス<br>(ビット 1 1 - 2 1) |
| 0 1               | 領域第 3 テーブル | 領域第 3 インデックス<br>(ビット 2 2 - 3 2) |
| 0 0               | セグメント・テーブル | 領域第 4 インデックス<br>(ビット 3 3 - 4 3) |

20

## 指示されたテーブルによる変換

## 【 0 1 5 5 】

ビット 6 0 及び 6 1 が二進数字 1 1 の値を有する場合、仮想アドレスの領域第 1 インデックス部分を、A S C E に含まれる領域第 1 テーブル基点と組み合わせて用いて、領域第 1 テーブルからエントリを選択する。実ストレージ又は絶対ストレージの中の領域第 1 テーブル・エントリの 6 4 ビット・アドレスは、領域第 1 テーブル指示のビット 0 - 5 1 の右側に 1 2 個の 0 を付加すること、及び、最右端に 3 つ、最左端に 5 0 個の 0 が付加された領域第 1 インデックスを加算することによって得られる。領域第 1 テーブル・ルックアップ・プロセスの一部として、アドレス指定されたエントリが領域第 1 テーブル内にあるかどうかを確立するために、仮想アドレスのビット 0 及び 1 (領域第 1 インデックスのビット 0 及び 1 である) が、領域第 1 テーブル指示のテーブル長、即ちビット 6 2 及び 6 3 に対して比較される。テーブル長フィールドにおける値が仮想アドレスの対応するビット位置における値より小さい場合には、領域第 1 変換例外が認識される。変換ルックアサイド・バッファ内の領域第 1 テーブル・エントリの等価物が変換に用いられる場合には、テーブル長に対する比較を省略することができる。領域第 1 テーブルからフェッチされたエントリは、対応する領域第 2 テーブルの開始位置を示し、そのオフセット及び長さを指定する。

30

40

## 【 0 1 5 6 】

A S C E のビット 6 0 及び 6 1 が二進数字 1 0 の値を有する場合、仮想アドレスの領域第 2 インデックス部分を、A S C E に含まれる領域第 2 テーブル基点と組み合わせて用いて、領域第 2 テーブルからエントリを選択する。仮想アドレスのビット 1 1 及び 1 2 (領域第 2 インデックスのビット 0 及び 1 である) が、A S C E の中のテーブル長に対して比較される。テーブル長フィールドにおける値が、仮想アドレスの対応するビット位置における値より小さい場合には、領域第 2 変換例外が認識される。変換ルックアサイド・バッファ内の領域第 2 テーブル・エントリの等価物が変換に用いられる場合には、テーブル長

50

に対する比較を省略することができる。その他の点では、領域第2テーブル・ルックアップ・プロセスは、領域第1テーブル・ルックアップ・プロセスと同じであり、領域第2テーブルからフェッチされたエントリは、対応する領域第3テーブルの開始位置を示し、そのオフセット及び長さを指定する。

#### 【0157】

A S C Eのビット60及び61が二進数字01の値を有する場合、仮想アドレスの領域第3インデックス部分を、A S C Eに含まれる領域第3テーブル基点と組み合わせて用いて、領域第3テーブルからエントリを選択する。仮想アドレスのビット22及び23（領域第3インデックスのビット0及び1である）が、A S C Eの中のテーブル長に対して比較される。テーブル長フィールドにおける値が、仮想アドレスの対応するビット位置における値より小さい場合には、領域第3変換例外が認識される。その他の点では、領域第3テーブル・ルックアップ・プロセスは領域第1テーブル・ルックアップ・プロセスと同じであり、領域第3テーブル・エントリ内のテーブル・タイプ・ビットをチェックすることを含む。領域第3テーブルからフェッチされたエントリは、対応するセグメント・テーブルの開始位置を示し、そのオフセット及び長さを指定する。

#### 【0158】

A S C Eのビット60及び61が二進数字00の値を有する場合、仮想アドレスのセグメント・インデックス部分を、A S C Eに含まれるセグメント・テーブル基点と組み合わせて用いて、セグメント・テーブルからエントリを選択する。仮想アドレスのビット33及び34（セグメント・インデックスのビット0及び1である）が、A S C Eの中のテーブル長に対して比較される。テーブル長フィールドにおける値が、仮想アドレスの対応するビット位置における値より小さい場合には、セグメント変換例外が認識される。変換ルックアサイド・バッファ内のセグメント・テーブル・エントリの等価物が変換に用いられる場合には、テーブル長に対する比較を省略することができる。その他の点では、セグメント・テーブル・ルックアップ・プロセスは領域第1テーブル・ルックアップ・プロセスと同じであり、セグメント・テーブル・エントリ内のテーブル・タイプ・ビットをチェックすることを含む。処理は以下の通りである。

#### 【0159】

強化型D A Tが適用されない場合、又は、強化型D A Tは適用されるが、S T Eフォーマット制御が0である場合には、セグメント・テーブルからフェッチされたエントリは、対応するページ・テーブルの開始位置を示し、処理は以下の「ページ・テーブル・ルックアップ」において説明されるように続行する。

#### 【0160】

強化型D A Tが適用され、且つ、S T Eフォーマット制御が1である場合には、セグメント・テーブルからフェッチされたエントリは、セグメント・フレーム絶対アドレスの最左端のビットを含む。変換に用いられるいずれかの領域テーブル・エントリ、又はセグメント・テーブル・エントリのどちらかにおいてD A T保護ビットが1であり、且つ、変換が実施されているストレージ参照がストアであれば、保護例外が認識される。

#### 【0161】

領域テーブル・エントリによって指示されるテーブルにおけるルックアップ

実効A S C Eが領域テーブル指示である場合には、前掲のセクションにおいて説明されたように領域テーブル・エントリが選択される。次いで、選択されたエントリの内容と、仮想アドレスの次のインデックス部分とを用いて、次に下位レベルのテーブルにおけるエントリが選択され、このテーブルは、別の領域テーブルの場合もあり、セグメント・テーブルの場合もある。A S C Eによって選択されたテーブル・エントリが領域第1テーブル・エントリである場合には、仮想アドレスの領域第2インデックス部分を領域第1テーブル・エントリに含まれる領域第2テーブル基点と組み合わせて用いて、領域第2テーブルからエントリを選択する。実ストレージ又は絶対ストレージの中の領域第2テーブル・エントリの64ビット・アドレスは、領域第1テーブル・エントリのビット0 - 51の右側に12個の0を付加すること、及び、最右端に3つ、最左端に50個の0が付加された領

10

20

30

40

50

域第 2 インデックスを加算することによって得られる。

【 0 1 6 2 】

領域第 2 テーブル・エントリ、領域第 3 テーブル・エントリ、又はセグメント・テーブル・エントリのアドレスを形成する際に、プリフィックス付加が行われる場合には、それは、より高位レベルのテーブル・エントリに含まれるそれぞれのテーブル基点に対してテーブル・インデックスの値を加算する前に適用されるのか、又はテーブル基点とテーブル・インデックス値との加算によって形成されたテーブル・エントリ・アドレスに対して適用されるのかは、予測不可能である。

【 0 1 6 3 】

領域第 2 テーブル・ルックアップ・プロセスの一部として、仮想アドレスのビット 1 1 及び 1 2 ( 領域第 2 インデックスのビット 0 及び 1 である ) は、領域第 1 テーブル・エントリのテーブル・オフセットであるビット 5 6 及び 5 7 に対して、及び領域第 1 テーブル・エントリのテーブル長であるビット 6 2 及び 6 3 に対して比較され、アドレス指定されたエントリが領域第 2 テーブルの範囲内にあるどうかを確立する。テーブル・オフセット・フィールドにおける値が仮想アドレスの対応するビット位置における値より大きい場合、又はテーブル長フィールドにおける値が仮想アドレスの対応するビット位置における値より小さい場合には、領域第 2 変換例外が認識される。

【 0 1 6 4 】

領域第 2 テーブルは、対応する領域第 3 テーブルの開始位置を示し、そのオフセット及び長さを指定する。

【 0 1 6 5 】

A S C E によって選択されたテーブル・エントリが領域第 2 テーブル・エントリである場合、又は領域第 2 テーブル・エントリが領域第 1 テーブル・エントリの内容によって選択されている場合には、仮想アドレスの領域第 3 インデックス部分を、領域第 2 テーブル・エントリに含まれる領域第 3 テーブル基点と組み合わせて用いて、領域第 3 テーブルからエントリを選択する。仮想アドレスのビット 2 2 及び 2 3 ( 領域第 3 インデックスのビット 0 及び 1 である ) が、領域第 2 テーブル・エントリ内のテーブル・オフセット及びテーブル長に対して比較される。テーブル・オフセットがビット 2 2 及び 2 3 より大きい場合、又はテーブル長がビット 2 2 及び 2 3 より小さい場合には、領域第 3 変換例外が認識される。その他の点では、領域第 3 テーブル・ルックアップ・プロセスは、領域第 2 テーブル・ルックアップ・プロセスと同一である。領域第 3 テーブルからフェッチされたエントリは、対応するセグメント・テーブルの開始位置を示し、そのオフセット及び長さを指定する。

【 0 1 6 6 】

A S C E によって選択されたテーブル・エントリが領域第 3 テーブル・エントリである場合、又は領域第 3 テーブル・エントリが領域第 2 テーブル・エントリの内容によって選択されている場合には、仮想アドレスのセグメント・インデックス部分を、領域第 3 テーブル・エントリに含まれるセグメント・テーブル基点と組み合わせて用いて、セグメント・テーブルからエントリを選択する。仮想アドレスのビット 3 3 及び 3 4 ( セグメント・インデックスのビット 0 及び 1 である ) が、領域第 3 テーブル・エントリ内のテーブル・オフセット及びテーブル長に対して比較される。テーブル・オフセットがビット 3 3 及び 3 4 より大きい場合、又はテーブル長がビット 3 3 及び 3 4 より小さい場合には、セグメント変換例外が認識される。変換指定例外は、( 1 ) A S C E 中のプライベート空間制御であるビット 5 5 が 1 であり、且つ ( 2 ) セグメント・テーブルからフェッチされたエントリ内の共通セグメント・ビットであるビット 5 9 が 1 である場合に、認識される。その他の点では、セグメント・テーブル・ルックアップ・プロセスは、領域第 2 テーブル・ルックアップ・プロセスと同一である。処理は以下の通りである。

【 0 1 6 7 】

強化型 D A T が適用されない場合、又は、強化型 D A T は適用されるが S T E フォーマット制御が 0 である場合には、セグメント・テーブルからフェッチされたエントリは、対

10

20

30

40

50

応するページ・テーブルの開始位置を示し、処理は後述の「ページ・テーブル・ルックアップ」において説明されるように続行する。

【 0 1 6 8 】

強化型 D A T が適用され、且つ、S T E フォーマット制御が 1 である場合には、セグメント・テーブルからフェッチされたエントリは、セグメント・フレーム絶対アドレスの最左端ビットを含む。変換に用いられるいずれかの領域テーブル・エントリ、又はセグメント・テーブル・エントリのどちらかにおいて、D A T 保護ビットが 1 であり、且つ、変換が実施されている対象のストレージ参照がストアであれば、保護例外が認識される。

【 0 1 6 9 】

ページ・テーブルのルックアップ

10

強化型 D A T が適用されない場合、又は強化型 D A T は適用されるが S T E フォーマット制御が 0 である場合には、仮想アドレスのページ・インデックス部分を、セグメント・テーブル・エントリに含まれるページ・テーブル基点と組み合わせて用いて、ページ・テーブルからエントリを選択する。

【 0 1 7 0 】

実ストレージ又は絶対ストレージ内のページ・テーブル・エントリの 6 4 ビット・アドレスは、ページ・テーブル基点の右に 1 1 個の 0 を付加すること、及び、最右端に 3 つ、最左端に 5 3 個の 0 が付加されたページ・インデックスを加算することによって得られる。ビット位置 0 のキャリーアウトは、生じることができない。

【 0 1 7 1 】

20

ページ・テーブルからフェッチされたエントリは、そのページの可用性を示し、ページ・フレーム絶対アドレスの最左端ビットを含む。対応するページが利用可能であるかどうかを確立するために、ページ無効ビットであるビット 5 3 が検査される。このビットが 1 である場合、ページ変換例外が認識される。ビット位置 5 2 が 1 を含んでいる場合、変換指定例外が認識される。強化型 D A T が適用されない場合、又は強化型 D A T は適用されるが S T E フォーマット制御が 0 である場合には、ビット位置 5 5 が 1 を含む場合にも変換指定例外が認識される。変換に用いられるセグメント・テーブル・エントリ、ページ・テーブル・エントリ、又は強化型 D A T が適用される場合に変換中に用いられるいずれかの領域テーブル・エントリのうちのいずれかにおいて、D A T 保護ビットが 1 であり、且つ、変換が実施されている対象のストレージ参照がストアであれば、保護例外が認識される。

30

【 0 1 7 2 】

実アドレス及び絶対アドレスの形成

実効 A S C E が実空間指示である場合には、仮想アドレスのビット 0 - 6 3 は直接、実ストレージ・アドレスとして用いられる。実アドレスに対してさらにプリフィックス付加を行って、絶対アドレスを形成することができる。実効 A S C E が実空間指示ではなく、変換プロセスにおいて例外が発生しない場合には、以下の条件が適用される。

【 0 1 7 3 】

強化型 D A T が適用されない場合、又は強化型 D A T は適用されるが S T E フォーマット制御が 0 である場合には、ページ・フレーム実アドレスは、ページ・テーブル・エントリから得られる。ページ・フレーム実アドレスと仮想アドレスのバイト・インデックス部分とを、ページ・フレーム実アドレスが最左端部分を形成するようにして連結させる。その結果が、その仮想アドレスに対応する実ストレージ・アドレスである。実アドレスに対してさらにプリフィックス付加を行って、絶対アドレスを形成することができる。

40

【 0 1 7 4 】

強化型 D A T が適用され、且つ、S T E フォーマット制御が 1 である場合には、セグメント・フレーム絶対アドレスと仮想アドレスのページ・インデックス部分及びバイト・インデックス部分とを左から右に連結して、仮想アドレスに対応する絶対アドレスを形成する。

【 0 1 7 5 】

50

## 変換中の例外の認識

無効アドレス及び無効フォーマットは、変換プロセス中に認識される例外を発生することがある。テーブル・エントリに含まれる情報が変換のために用いられ、それが正しくない正しくないことが見いだされた場合に、例外が認識される。

### 【 0 1 7 6 】

ここで、動的アドレス変換の一実施形態の流れ図をセグメント・テーブル・エントリからフォーマット制御フィールドを取得する時点まで示す、図 8 を参照する。

### 【 0 1 7 7 】

6 0 2 において、変換する仮想アドレスが取得される。6 0 4 において、仮想アドレスの変換に用いられる最も高位の変換テーブルの基点アドレスが取得される。変換に用いられる最初の変換テーブルの基点アドレスは、A D C E 及び D T ビットに依存する。6 0 6 において、仮想アドレスの一部を用いて、変換テーブル内の適切なテーブル・エントリを参照する。6 0 8 において、変換テーブルからフェッチされたエントリがセグメント・テーブル・エントリでない場合には、変換テーブル階層構造の中のセグメント・テーブルはまだ参照されていない。この場合、6 1 0 において、変換テーブル階層構造内の次に低位の変換テーブルの基点アドレスが、テーブル・エントリから取得される。仮想アドレスの適切な部分を用いて、変換に用いられる次に低位のテーブル内の対応するテーブル・エントリを参照する。

### 【 0 1 7 8 】

例えば、変換に用いられる最初の変換テーブルの基点アドレスが領域第 1 テーブルであれば、仮想アドレスの R F X 部分を用いて、領域第 1 テーブル内の領域第 1 テーブル・エントリを参照する。テーブル基点アドレスが領域第 2 テーブルに対するものであれば、仮想アドレスの R S X 部分を用いて、領域第 2 テーブル内の領域第 2 テーブル・エントリを参照する。テーブル基点アドレスが領域第 3 テーブルに対するものであれば、仮想アドレスの R T X 部分を用いて、領域第 3 テーブル内の領域第 3 テーブル・エントリを参照する。テーブル基点アドレスがセグメント・テーブルに対するものであれば、仮想アドレスの S X 部分を用いて、セグメント・テーブル内のセグメント・テーブル・エントリを参照するセグメント・テーブル・エントリがフェッチされるまで、逐次的なテーブルが参照される。

### 【 0 1 7 9 】

一旦セグメント・テーブル・エントリがフェッチされると、6 1 2 において、セグメント・テーブル・エントリ ( S T E ) フォーマット制御ビットが検査され、この特定の仮想アドレスに対してフォーマット制御がイネーブルであるかどうかを判定する。S T E フォーマット制御が 0 であれば、動的アドレス変換はノード 6 1 4 に関して行われる。S T E フォーマット制御が 1 であれば、動的アドレス変換はノード 6 1 6 に関して行われる。

### 【 0 1 8 0 】

動的アドレス変換 ( S T E フォーマット制御が 0 )

ここで、S T E フォーマット制御が 0 である場合の図 8 のノード 6 1 4 からの流れ図の続きを示す、図 9 を参照する。

### 【 0 1 8 1 】

7 1 0 において、セグメント・テーブル・エントリから、ページ・テーブルの基点アドレスが取得される。7 1 2 において、仮想アドレスの P X 部分を用いて、ページ・テーブル内のページ・エントリを参照する。7 1 4 において、ページ・テーブル・エントリから、ページ・フレーム実アドレス ( P F R A ) が取得される。無効 ( I ) ビットがページ・テーブル・エントリから取得される。7 1 6 において、無効 ( I ) ビットが 1 である場合、このページ・テーブル・エントリは無効であるものとしてマークされているので、7 1 8 において、このページ・テーブル・エントリを用いた仮想アドレスの変換を続けることはできない。このページ・テーブル・エントリを用いた仮想アドレスのそれ以上の変換は、停止する ( 7 2 2 )。7 1 6 において、無効 ( I ) ビットが 0 である場合、7 2 0 において、ページ・フレーム実アドレス ( P F R A ) を仮想アドレスの B X 部分と組み合わせ

て、実アドレスを生成する。実アドレスに対してさらにプリフィックス付加操作を行って、絶対アドレスを形成することができる。724において、実アドレスを用いて、変換された仮想アドレスによってアドレス指定されたデータ・ブロックにアクセスする。

#### 【0182】

動的アドレス変換（STEフォーマット制御が1）

ここで、図8のノード616からの流れ図の続きを示す、図10を参照する。

#### 【0183】

810において、セグメント・フレーム絶対アドレス（SFAA）が、セグメント・テーブル・エントリの一部から取得される。無効（I）ビットがセグメント・テーブル・エントリから取得される。812において、無効（I）ビットが1である場合、セグメント・テーブル・エントリは無効であるものとしてマークされているので、814において、このセグメント・テーブル・エントリを用いた仮想アドレスのそれ以上の変換を続けることはできない。一実施形態において、変換を要求するプログラム・エンティティに対して、例外コードが戻される。このセグメント・テーブル・エントリを用いた仮想アドレスのそれ以上の変換は、停止する（818）。

10

#### 【0184】

812において、無効（I）ビットが0である場合、816において、セグメント・フレーム絶対アドレス（SFAA）を仮想アドレスのPX部分及びBX部分と組み合わせて、主ストレージ又はメモリ内の所望の大データ・ブロックに対する絶対アドレスを生成する。820において、変換された仮想アドレスによってアドレス指定される所望の大データ・ブロックにアクセスする。

20

#### 【0185】

各テーブル・エントリにおいて、取得されたテーブル・エントリの妥当性を判定するために、無効ビットが検査される。変換された仮想アドレスによってアドレス指定されるデータ・ブロックを保護する他の変換保護機構について、本明細書においてさらに検討する。

#### 【0186】

別の実施形態において、制限情報が変換テーブル・エントリから取得される。制限情報は、アクセスを仮想アドレス範囲の制限部分に対するものに限定するために用いられる。これ以降は、変換されたアドレスによってアドレス指定される所望の大データ・ブロックに対するアクセスは、アドレス範囲の制限部分に対してのみ許可される。制限情報は、テーブル・オフセット又はテーブル長のどれか1つである。

30

#### 【0187】

さらに別の実施形態において、仮想アドレスの変換に用いられる情報は、少なくとも1つの変換ルックアサイド・バッファ内にストアされる。その後の仮想アドレスの、主ストレージ内のデータ・ブロックの絶対アドレスへの変換は、それ以降、変換テーブル階層構造ではなく変換ルックアサイド・バッファ由来のストアされた情報を用いて実施される。

#### 【0188】

さらに別の実施形態において、変換がマシン・アーキテクチャに対してネイティブでない場合には、変換をエミュレートするための所定のソフトウェア・ルーチンが識別される。所定のソフトウェア・ルーチンは、複数の命令を含む。所定のソフトウェア・ルーチンが実行される。

40

#### 【0189】

アドレス指定されたデータ・ブロックの保護

本明細書で説明されるように、仮想アドレスが強化型DATファシリティを用いて変換されると、変換された仮想アドレスによってアドレス指定される主ストレージ又はメモリ内の所望のデータ・ブロックに対して、付加的な保護機構を適用することができる。

#### 【0190】

DAT保護

DAT保護機能は、仮想ストレージへのアクセスを、各ページ・テーブル・エントリ及

50

びセグメント・テーブル・エントリ内のDAT保護ビット、及び、強化型DATファシリティがインストールされている場合には各領域テーブル・エントリ内のDAT保護ビットを用いることによって、制御する。これは不正なストアに対する保護を与える。

【0191】

ページ・テーブル・エントリのDAT保護ビット(ビット54)は、対応する4キロバイトのページへのストアが許可されるかどうかを制御する。ビットが0である場合にはフェッチ及びストアの両方が許可され、ビットが1である場合にはフェッチのみが許可される。保護されたページへのストアが試行された場合、ページの内容は変更されず、操作単位(unit of operation)又は命令の実行が抑制され、保護のためのプログラム割り込みが発生する。

10

【0192】

セグメント・テーブル・エントリのDAT保護ビット(ビット54)は、対応する1メガバイトのセグメントへのストアが許可されるかどうかを以下のように制御する。

【0193】

強化型DATが適用されない場合、又は強化型DATが適用され、且つSTEフォーマット制御が0である場合には、セグメント・テーブル・エントリのDAT保護ビットは、そのセグメント・テーブル・エントリによって指示されるページ・テーブル内の各エントリのDAT保護ビット位置内へと論理和演算されたものとして扱われる。よって、セグメント・テーブル・エントリのDAT保護ビットが1である場合、その効果は、あたかも指示されたページ・テーブルの中の各エントリ内のDAT保護ビットが1であるかのようなものである。

20

【0194】

強化型DATが適用され、且つSTEフォーマット制御が1である場合には、セグメント・テーブル・エントリのDAT保護ビットは、対応する1メガバイトのセグメントへのストアが許可されるかどうかを制御する。ビットが0である場合にはフェッチ及びストアの両方が許可され、ビットが1である場合にはフェッチのみが許可される。保護されたセグメントへストアが試行された場合、セグメントの内容は変更されず、操作単位又は命令の実行が抑制され、保護のためのプログラム割り込みが発生する。

【0195】

強化型DATが適用される場合、領域テーブル・エントリのDAT保護ビットは、対応する領域へのストアが許可されるかどうかを制御する。領域テーブル・エントリ内のDAT保護ビットは、変換に用いられるいずれかの後続の領域テーブル・エントリ及びセグメント・テーブル・エントリのDAT保護ビット位置内へと論理和演算されたものとして扱われる。STEフォーマット制御が0である場合、DAT保護ビットは、さらにページ・テーブル・エントリまで伝搬される。

30

【0196】

DAT保護は、仮想アドレスを用いる全てのストア・タイプ参照に適用される。

【0197】

ここで、変換に用いられるセグメント・テーブル・エントリからDAT保護ビットを得る一実施形態の流れ図を示す、図11を参照する。

40

【0198】

902において、変換される仮想アドレスが取得される。904において、仮想アドレスの変換に用いられる最初の変換テーブルの基点が取得される。906において、前述された手法で、仮想アドレスの一部を用いて変換テーブル内のエントリを参照する。908において、実アドレス又は絶対アドレスが、取得されたセグメント・フレーム絶対アドレス(SFAA)又はページ・フレーム実アドレス(PFRA)から生成される。

【0199】

910において、DAT保護(P)ビットが、仮想アドレスの変換に用いられた変換テーブルの各エントリから取得される。DAT保護の実効レベルは、変換に用いられた各テーブル・エントリから取得される複数のPビットの論理和であることを理解すべきである

50



。 9 1 2 において、 D A T 保護ビットが検査され、変換された仮想アドレスによってアドレス指定されたデータ・ブロックが D A T 保護されるかどうかを判定する。 9 1 2 において、 D A T 保護 ( P ) ビットが 0 である場合、 9 1 4 において、変換された仮想アドレスによってアドレス指定されるデータ・ブロックへのストア動作が許可される。 9 1 2 において、 D A T 保護 ( P ) ビットが 1 である場合、 9 1 6 において、変換された仮想アドレスによってアドレス指定されたデータ・ブロックへのストア動作が防止される。あるいは、 D A T 保護フィールドによって示される保護レベルが、その仮想アドレスを用いる全てのストア・タイプ参照に対して適用される。 D A T 保護フィールドがイネーブルである場合、変換された仮想アドレスによってアドレス指定された所望のデータ・ブロックへのストアが試行されると、保護例外条件が示される。

10

#### 【 0 2 0 0 】

##### キー制御保護

キー制御保護がストレージ・アクセスに適用される場合、ストアは、ストレージ・キーがストレージ・アクセス要求に関連付けられたアクセス・キーと一致する場合にのみ許可され、フェッチは、キーが一致する場合、又はストレージ・キーのフェッチ保護ビットが 0 である場合に許可される。ストレージ・キーの 4 つのアクセス制御ビットがアクセス・キーに等しい場合、又はアクセス・キーが 0 である場合に、キーが一致したと言う。保護アクションは、以下のように要約される。

#### 【 表 1 7 】

20

| 条件   |       | ストレージへのアクセスは許可されているか |     |
|--|-------|----------------------|-----|
| ストレージ・キーのフェッチ保護ビット   | キーの関係 | フェッチ                 | ストア |
| 0  | 一致    | はい                   | はい  |
| 0  | 不一致   | はい                   | いいえ |
| 1  | 一致    | はい                   | はい  |
| 1  | 不一致   | いいえ                  | いいえ |
| 説明：<br>一致 ストレージ・キーの 4 つのアクセス制御ビットがアクセス・キーに等しい、又はアクセス・キーが 0 である。<br>はい アクセスが許可される。<br>いいえ アクセスは許可されない。<br>フェッチの場合、情報はプログラムに対して利用可能にされない。<br>ストアの場合、ストレージ位置の内容は変更されない。 |       |                      |     |

30

#### 保護アクションの要約

#### 【 0 2 0 1 】

40

ストレージへのアクセスが C P U によって開始され、且つキー制御保護が適用される場合には、 P S W キーがアクセス・キーであるが、但し、 M O V E T O S E C O N D A R Y 及び M O V E W I T H D E S T I N A T I O N K E Y の第 1 オペランド、 M O V E T O P R I M A R Y 、 M O V E W I T H K E Y 、 及び M O V E W I T H S O U R C E K E Y の第 2 オペランド、並びに第 M O V E P A G E の 1 オペランド又は第 2 オペランドのいずれかについては、アクセス・キーは汎用レジスタ内で指定される。 P S W キーは、カレント・プログラム・ステータス・ワードの 8 - 1 1 のビット位置を占める。

#### 【 0 2 0 2 】

キー制御保護が原因で C P U アクセスが禁止されるとき、命令の実行は停止し、保護例

50

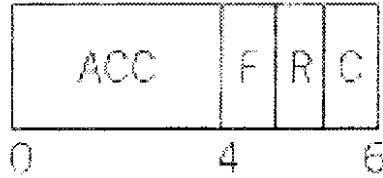
外のためのプログラム割り込みが発生する。しかしながら、操作単位又は命令の実行は抑制されることもある。

#### 【 0 2 0 3 】

##### ストレージ・キー

ストレージ・キーは、構成内で利用可能なストレージの各々の 4 キロバイト・ブロックに関連付けられる。ストレージ・キーはアドレス指定可能なストレージの一部ではない。一実施形態において、ストレージ・キーは以下のフォーマットを有する。

【表 1 8】



10

## ストレージ・キーのフォーマット

#### 【 0 2 0 4 】

ストレージ・キーの中のビット位置は以下のように割り振られる。

アクセス制御ビット (ACC) : 参照がキー制御保護を受けているならば、情報がストアされる場合、及びフェッチに対して保護されている位置から情報がフェッチされる場合に、4つのアクセス制御ビットが、4ビットのアクセス・キーと照合される。

20

#### 【 0 2 0 5 】

フェッチ保護ビット (F) : 参照がキー制御保護を受けているならば、フェッチ保護ビットは、キー制御保護がフェッチ・タイプ参照に適用されるかどうかを制御し、0は、ストア・タイプ参照のみが監視され、どのようなアクセス・キーを有するフェッチも許可されることを示し、1は、キー制御保護がフェッチ及びストアの両方に適用されることを示す。

#### 【 0 2 0 6 】

参照ビット (R) : 参照ビットは通常、対応するストレージ・ブロック内の位置が情報のストア又はフェッチのいずれかのために参照される都度、1に設定される。

30

#### 【 0 2 0 7 】

変更ビット (C) : 変更ビットは、対応するストレージ・ブロック内の位置に情報がストアされる都度、1に設定される。

#### 【 0 2 0 8 】

強化型 DAT が適用される場合、以下の追加条件が効力をもつ。

#### 【 0 2 0 9 】

STE フォーマット制御 (変換中に用いられるセグメント・テーブル・エントリのビット 53 である FC) が 0 である場合、変換中に用いられるページ・テーブル・エントリのビット 55 は、そのページのための変更記録オーバーライド (CO) である。ページ・テーブル・エントリ内の CO ビットが 1 である場合、そのページに対するいずれのストア動作についても変更記録は予測不可能である。

40

#### 【 0 2 1 0 】

セグメント・テーブル・エントリの中のフォーマット制御ビット (CO) が 1 である場合、以下が適用される。

#### 【 0 2 1 1 】

セグメント・テーブル・エントリのビット位置 47 は、ACCF 妥当性制御を含む。ACCF 妥当性制御は、STE の中のアクセス制御ビット及びフェッチ保護ビットの妥当性を決定する。ACCF 妥当性制御が 0 である場合、キー制御保護は、そのアドレスに対応する 4 K バイト・ブロックのためのストレージ・キーの中のアクセス制御ビット及びフェッチ保護ビットを用いる。

50

## 【 0 2 1 2 】

A C C F 妥当性制御が 1 である場合、セグメント・テーブル・エントリのビット位置 4 8 - 5 2 は、そのセグメントについてのアクセス制御ビット及びフェッチ保護ビットを含む。ストレージ・オペランドに対するアクセス可能性を判定するときに、S T E のビット 4 8 - 5 2、又は、セグメントを構成する 4 K バイト・ブロックのための個々のストレージ・キーのビット 0 - 4 のどちらが検査されるかは、予測不可能である。

## 【 0 2 1 3 】

セグメント・テーブル・エントリのビット 5 5 は、そのセグメントについての変更記録オーバーライド (C O) である。セグメント・テーブル・エントリ内の C O ビットが 1 である場合、そのセグメントに対するいずれかのストア動作について変更記録が設定されるかどうかは予測不可能である。

10

## 【 0 2 1 4 】

ストレージ・キー・アクセス

ストレージ・キーへの参照は、以下のように取り扱われる。

## 【 0 2 1 5 】

ストレージの参照が行われ、且つキー制御保護が参照に適用される場合には常に、そのストレージ位置に関連付けられた 4 つのアクセス制御ビット及びフェッチ保護ビットが同時に、且つストレージ位置に対する参照と同時に、検査される。( 1 ) 強化型 D A T が適用されない場合、( 2 ) 強化型 D A T は適用されるが、S T E フォーマット制御が 0 であるセグメント・テーブル・エントリによってストレージがアクセスされる場合、又は( 3 ) 強化型 D A T が適用され、S T E フォーマット制御が 1 であるが、A C C F 妥当性制御が 0 であるセグメント・テーブル・エントリによってストレージがアクセスされる場合、アクセス制御ビット及びフェッチ保護ビットは、4 K バイト・ブロックのためのストレージ・キーのビット 0 - 4 に存在する。強化型 D A T が適用され、S T E フォーマット制御及び A C C F 妥当性制御の両方が 1 であるセグメント・テーブル・エントリによってストレージがアクセスされる場合、ストレージ・キーのビット 0 - 4 又はセグメント・テーブル・エントリのビット 4 8 - 5 2 のどちらがアクセス制御ビット及びフェッチ保護ビットを与えるかは予測不可能である。さらに、セグメント・テーブル・エントリがアクセス制御ビット及びフェッチ保護ビットを与える場合には、変換ルックアサイド・バッファ由来のバッファされたコピーを用いることができる。

20

30

## 【 0 2 1 6 】

強化型 D A T が適用され、且つ、( a ) S T E フォーマット制御が 0 であり、且つ、D A T によって用いられるページ・テーブル・エントリ内の変更記録オーバーライドが 1 であるか、又は( b ) S T E フォーマット制御が 1 であり、且つ、D A T によって用いられるセグメント・テーブル・エントリ内の変更記録オーバーライドが 1 であるかのいずれかの場合、C P U がストア動作を実施するときに変更ビットを設定するかどうかは予測不可能である。変更記録オーバーライドは、P T E 又は S T E の変換ルックアサイド・バッファ・コピーの中にバッファリングすることができる。

## 【 0 2 1 7 】

条件付き S S K E 特徴がインストールされていない場合、S E T S T O R A G E K E Y E X T E N D E D 命令によって、ストレージ・キーの中の 7 つのビットの全てが同時に設定される。条件付き S S K E がインストールされている場合、S E T S T O R A G E K E Y E X T E N D E D 命令を用いて、プログラム指定基準に基づいてストレージ・キーの全て又は一部を設定することができる。

40

## 【 0 2 1 8 】

I N S E R T S T O R A G E K E Y E X T E N D E D 命令は、4 K バイト・ブロックのためのストレージ・キーのビット 0 - 6 の一貫したイメージを提供する。同様に、I N S E R T V I R T U A L S T O R A G E K E Y 命令及び T E S T P R O T E C T I O N 命令は、アクセス制御ビット及びフェッチ保護ビットの一貫したイメージを提供する。

50

## 【0219】

RESET REFERENCE BIT EXTENDED命令は、参照ビットのみを修正する。ストレージ・キーの他の全てのビットは変更されないままである。条件コードを設定するために、参照ビット及び変更ビットが同時に検査される。

## 【0220】

参照ビットによって提供される参照の記録は、必ずしも正確でない。しかし、大部分の状況において、参照記録は、関連するストレージ参照とほぼ一致する。変更ビットを、ストアが発生しなかった場合に設定することができる。

## 【0221】

他のCPUによって観察される際に、ストレージ・キーを明示的に操作する命令 (INSERT STORAGE KEY EXTENDED、INSERT VIRTUAL STORAGE KEY、RESET REFERENCE BIT EXTENDED、及びSET STORAGE KEY EXTENDED) に起因するストレージ・キー・フェッチ及びストアは、あたかもストレージ・キー・アクセス自体がそれぞれストレージ・オペランド・フェッチ及びストアであるかのように、それ自体の中で、及びストレージ・オペランド参照の中で順序付けられる。

10

## 【0222】

アクセス制御及びフェッチ保護 (ACC/F)

主ストレージ又はメモリ内のデータ・ブロックは、セグメント・テーブル・エントリ内に与えられたアクセス制御ビット及びフェッチ保護ビットによって、さらに保護することができる。

20

## 【0223】

アクセス妥当性 (AV) 及びアクセス制御 (ACC)

ここで、変換された仮想アドレスによってアドレス指定される所望のデータ・ブロックに適用されるアクセス制御保護を判定する流れ図の一実施形態を示す、図12を参照する。

## 【0224】

動的アドレス変換は、図8においてノード616のところまで説明されたように、変換に用いられるセグメント・テーブルからセグメント・テーブル・エントリがフェッチされた時点まで進行する。アドレス変換は、図10においてノード822のところまで説明されたように、ストレージ又はメモリ内の所望のデータ・ブロックに対する絶対アドレスが取得されたところまで続行する。図12のフローはノード822から開始する。

30

## 【0225】

1002において、アクセス妥当性 (AV) フィールドがセグメント・テーブル・エントリから取得される。1004において、セグメント・テーブル・エントリ内のアクセス妥当性がイネーブルである (AV = 1) かどうかの判定が行われる。アクセス妥当性フィールドが0であれば、セグメント・テーブル・エントリ内のアクセス制御 (ACC) フィールドは有効ではない。1006において、アクセス制御 (ACC) フィールドのビットは無視される。アクセス妥当性フィールドがイネーブルであれば、アクセス制御フィールドは有効である。1008において、アクセス制御 (ACC) フィールドのビットがセグメント・テーブル・エントリから取得される。1010において、セグメント・テーブル・エントリ内のACCフィールドによって示されるアクセス制御保護レベルが、変換された仮想アドレスによってアドレス指定される所望のデータ・ブロックに適用される。

40

## 【0226】

別の実施形態において、所望のデータ・ブロックへのストア動作は、アクセス制御フィールドが、プログラム・ステータス・ワード又は実行中のプログラム命令のオペランドといったプログラム・エンティティによって与えられるアクセス・キーに一致した場合のみ、許可される。所望のデータ・ブロックにおいてストアが試行され、且つアクセス制御フィールドがアクセス・キーに一致しなければ、保護例外が示されることが好ましい。

## 【0227】

50

さらに別の実施形態において、アクセス妥当性フィールドがイネーブルである場合、所望のメモリ・ブロックにおいてフェッチ動作又はストア動作が試行され、且つセグメント・アクセス制御フィールドがプログラム・アクセス制御フィールドに一致しなければ、保護例外が示される。

#### 【0228】

アクセス妥当性 (AV) 及びフェッチ保護 (F)

ここで、変換された仮想アドレスによってアドレス指定される所望のデータ・ブロックに適用されるフェッチ保護を判定する流れ図の一実施形態を示す、図13を参照する。

#### 【0229】

動的アドレス変換は、図8においてノード616のところまで説明されたように、変換に用いられるセグメント・テーブルからセグメント・テーブル・エントリがフェッチされた時点まで進行する。アドレス変換は、図10においてノード822のところまで説明されたように、ストレージ又はメモリ内の所望のデータ・ブロックに対する絶対アドレスが取得されたところまで続行する。図13のフローはノード822から開始する。

#### 【0230】

1102において、アクセス妥当性 (AV) フィールドがセグメント・テーブル・エントリから取得される。1104において、セグメント・テーブル・エントリ内のアクセス妥当性がイネーブルである (AV = 1) かどうかの判定が行われる。1104において、アクセス妥当性フィールドが0であれば、1106において、フェッチ保護ビットは無視される。1104において、アクセス妥当性フィールドがイネーブルであれば、1108において、フェッチ保護ビットがセグメント・テーブル・エントリから取得される。1110において、セグメント・テーブル・エントリ内のFフィールドによって示されるフェッチ保護レベルが、変換された仮想アドレスによってアドレス指定される所望のデータ・ブロックに適用される。変換された仮想アドレスによってアドレス指定されるデータ・ブロックからのフェッチ動作は、フェッチ保護フィールドがアクセス・キーに一致した場合にのみ、許可される。アクセス・キーは、プログラム・ステータス・ワード又は実行中のプログラム命令のオペランドによって与えられる。所望のデータ・ブロックからのフェッチ動作は、フェッチ保護フィールドがディスエーブルである場合、又はフェッチ保護フィールドがイネーブルであり、且つ仮想アドレスに関連付けられたアクセス制御フィールドがアクセス・キーに等しい場合にのみ、許可される。アクセス・キーが0であれば、所望のデータ・ブロックからのフェッチが許可される。

#### 【0231】

さらに別の実施形態において、プログラム・アクセス・キーが0であれば、所望のメモリ・ブロックへのストア動作及び所望のメモリ・ブロックからのフェッチ動作が許可される。

#### 【0232】

さらに別の実施形態において、アクセス妥当性フィールド及びフェッチ保護フィールドがイネーブルである場合、所望のメモリ・ブロックからのフェッチ動作が試行され、且つセグメント・アクセス制御フィールドがプログラム・アクセス制御フィールドに一致しないことに応答して、保護例外が示される。

#### 【0233】

SET STORAGE KEY EXTENDED (SSKE)

ストレージ・キーは、SET STORAGE KEY EXTENDED (SSKE) 命令によって設定することができる。一実施形態において、SSKE命令は、以下のフォーマットを有する。

【表 19】

| SSKE |    | $R_1, R_2 \text{ [M}_2\text{]}$ |      | [RRF]          |                |
|------|----|---------------------------------|------|----------------|----------------|
| B22B |    | M <sub>3</sub>                  | //// | R <sub>1</sub> | R <sub>2</sub> |
| 0    | 16 | 20                              | 24   | 28             | 31             |

## SSKE命令のフォーマット

10

## 【0234】

1つ又は複数の4 Kバイト・ブロックのためのストレージ・キーが、第1オペランド・レジスタ内の値で置換される。条件付きSSKEファシリティがインストールされている場合には、キー設定動作の特定の機能をバイパスすることができる。条件付きSSKEファシリティがインストールされていない場合、又は条件付きSSKEファシリティがインストールされ、且つM3フィールドのMRビット及びMCビットの両方が0である場合には、汎用レジスタR2の内容によってアドレス指定される4 Kバイト・ブロックのためのストレージ・キーが、汎用レジスタR1からのビットで置換される。命令は、条件コードを変更することなく完了する。

## 【0235】

20

条件付きSSKEファシリティがインストールされ、且つMRビット及びMCビットの一方又は両方が1である場合には、汎用レジスタR2の内容によってアドレス指定されるストレージ・キーのアクセス制御ビット、フェッチ保護ビット、及び随意に参照ビット及び変更ビットが、汎用レジスタR1内の対応するビットと比較される。比較されたビットが等しければ、キーに対する変更は行われない。そうでない場合には、キーの選択されたビットが、汎用レジスタR1内の対応するビットによって置換される。いずれかの変更の前のストレージ・キーは、汎用レジスタR1の中に挿入され、結果は条件コードによって示される。

## 【0236】

強化型DATファシリティがインストールされている場合、後述の複数ブロック制御の制御の下で、同一の1 MBブロック内の複数の4 Kバイト・ブロックのストレージ・キーに対して上記の動作を繰り返すことができる。一実施形態において、M3フィールドは、以下のフォーマットを有する。

30

## 【表 20】

|  |   |   |   |   |
|--|---|---|---|---|
|  | / | M | M | M |
|  |   | R | C | B |
|  | 0 | 1 | 2 | 3 |

40

## M3フィールドのフォーマット

M3フィールドのビットは、以下のように定められる。

## 【0237】

未使用：ビット0は未使用とされる。

## 【0238】

参照ビット更新マスク(MR)：MRビット、即ちM3フィールドのビット1は、後述のように、ストレージ・キーの中の参照ビットに対する更新をバイパスすることができるかどうかを制御する。

50

## 【 0 2 3 9 】

変更ビット更新マスク (MC) : MCビット、即ちM3フィールドのビット2は、後述のように、ストレージ・キーの中の変更ビットに対する更新をバイパスすることができるかどうかを制御する。

## 【 0 2 4 0 】

複数ブロック制御 (MB) : MBビット、即ちM3フィールドのビット3は、ストレージの複数の4Kバイト・ブロックのためのストレージ・キーを、複数の4Kバイト・ブロックにおけるストレージ・キーの設定の項で説明されるように設定してもよいかどうかを制御する。

## 【 0 2 4 1 】

強化型DATファシリティがインストールされていない場合、M3フィールドのビット位置3は未使用とされる。条件付きSSEファシリティがインストールされている場合、処理は以下の通りである。

## 【 0 2 4 2 】

MRビット及びMCビット、即ちM3フィールドのビット1及び2の両方が0である場合、命令は、あたかも条件付きSSEファシリティがインストールされていないかのように完了する。汎用レジスタR2の内容によってアドレス指定される4Kバイト・ブロックのためのストレージ・キーは、汎用レジスタR1からのビットで置換され、命令は条件コードを変更することなく完了する。

## 【 0 2 4 3 】

MRビット及びMCビットの一方又は両方が1である場合、処理は以下の通りである。

## 【 0 2 4 4 】

いずれかの変更在先立ち、汎用レジスタR2の内容によってアドレス指定される4Kバイト・ブロックのためのストレージ・キーの内容が汎用レジスタR1のビット位置48 - 54に置かれ、汎用レジスタR1のビット55は0に設定される。レジスタのビット0 - 47及び56 - 63は変更されない。ストレージ・キーをフェッチするときに無効なチェック・ブロック・コード(CBC)が検出された場合には、(a)4Kバイト・ブロックのためのストレージ・キー全体が汎用レジスタR1のビット56 - 62によって置換され、(b)汎用レジスタR1のビット位置48 - 55の内容は予測不可能であり、且つ(c)命令は条件コード3を設定することによって完了する。

## 【 0 2 4 5 】

指示された4Kバイト・ブロックのためのストレージ・キーのアクセス制御ビット及びフェッチ保護ビットが、汎用レジスタR1のビット56 - 60における対応するフィールドと比較される。それぞれのフィールドが等しくない場合には、4Kバイト・ブロックのためのストレージ・キー全体が汎用レジスタR1からのビットによって置換され、命令は条件コード1を設定することによって完了する。ストレージ・キーの中のアクセス制御ビット及びフェッチ保護ビットが汎用レジスタR1の中のそれぞれのビットに等しい場合には、処理は以下で説明されるように続行する。

## 【 0 2 4 6 】

MRビット及びMCビットの両方が1である場合、命令は条件コード0を設定することによって完了する。この場合は、ストレージ・キーは変更されないままである。

## 【 0 2 4 7 】

MRビットが0であり、且つMCビットが1である場合には、指示された4Kバイト・ブロックのためのストレージ・キーの参照ビットが、汎用レジスタR1のビット61と比較される。ビットが等しければ、命令は条件コード0を設定することによって完了する。この場合は、ストレージ・キーは変更されないままである。ビットが等しくない場合は、(a)指示された4Kバイト・ブロックのためのストレージ・キー全体が汎用レジスタR1の中のビットによって置換され、命令は条件コード1を設定することによって完了するか、又は(b)ストレージ・キーのための参照ビットが汎用レジスタR1のビット61によって置換され、そのキーのための変更ビットは予測不可能であり、命令は条件コード2

10

20

30

40

50

を設定することによって完了するかの、いずれかである。条件コード 1 又は 2 のどちらが設定されるかは、予測不可能である。

【 0 2 4 8 】

MC ビットが 0 であり、且つ MR ビットが 1 である場合には、指示された 4 K バイト・ブロックのためのストレージ・キーの変更ビットが、汎用レジスタ R 1 のビット 6 2 と比較される。ビットが等しければ、命令は条件コード 0 を設定することによって完了する。この場合は、ストレージ・キーは変更されないままである。ビットが等しくない場合は、( a ) 指示された 4 K バイト・ブロックのためのストレージ・キー全体が汎用レジスタ R 1 の中のビットによって置換され、命令は条件コード 1 を設定することによって完了するか、又は ( b ) ストレージ・キーのための参照ビットが汎用レジスタ R 1 のビット 6 2 によって置換され、そのキーのための変更ビットは予測不可能であり、命令は条件コード 2 を設定することによって完了するかの、いずれかである。条件コード 1 又は 2 のどちらが設定されるかは、予測不可能である。

【 0 2 4 9 】

強化型 DAT ファシリティがインストールされていない場合、又はファシリティはインストールされているが複数ブロック制御が 0 である場合には、汎用レジスタ R 2 は実アドレスを含む。強化型 DAT ファシリティがインストールされており、且つ複数ブロック制御が 1 である場合には、汎用レジスタ R 2 は絶対アドレスを含む。24 ビット・アドレス指定モードにおいては、汎用レジスタ R 2 のビット 4 0 - 5 1 が実ストレージ又は絶対ストレージ内の 4 K バイト・ブロックを指示し、レジスタのビット 0 - 3 9 及び 5 2 - 6 3 は無視される。31 ビット・アドレス指定モードにおいては、汎用レジスタ R 2 のビット 3 3 - 5 1 が実ストレージ又は絶対ストレージ内の 4 K バイト・ブロックを指示し、レジスタのビット 0 - 3 2 及び 5 2 - 6 3 は無視される。64 ビット・アドレス指定モードにおいては、汎用レジスタ R 2 のビット 0 - 5 1 が実ストレージ又は絶対ストレージ内の 4 K バイト・ブロックを指示し、レジスタのビット 5 2 - 6 3 は無視される。ストレージ・ブロックを指示するアドレスは、実アドレス又は絶対アドレスであるため、動的アドレス変換の対象ではない。ストレージ・キーに対する参照は、保護例外の対象ではない。

【 0 2 5 0 】

新たな 7 ビットのストレージ・キー値、又はその選択されたビットが、汎用レジスタ R 1 のビット位置 5 6 - 6 2 から取得される。レジスタのビット位置 0 - 5 5 及び 6 3 の内容は無視される。条件付き SSE ファシリティがインストールされており、且つ MR ビット及び MC ビット的一方又は両方が 1 である場合には、ビット位置 6 3 は 0 を含むべきである。さもなければ、プログラムは、将来的に互換的に動作しなくなる可能性がある。

【 0 2 5 1 】

シリアル化及びチェックポイント同期化機能が、動作の開始前に実施され、動作の完了後にも再度、実施されるが、但し、条件付き SSE ファシリティがインストールされており、且つ結果として得られた条件コードが 0 である場合には、動作完了後にシリアル化及びチェックポイント同期化機能が実施されるかどうかは予測不可能である。いずれかの CPU 又はチャネル・プログラムによる、指示された 4 K バイト・ブロックに対する、この命令によるキーの設定の前又は後のどちらかに完了する、どのストア・アクセスについても、そのブロックのためのストレージ・キーにおける関連付けられた参照ビット及び変更ビットの 1 への設定もまた、この命令の実行のそれぞれ前又は後に完了する。

【 0 2 5 2 】

複数の 4 K バイト・ブロックにおけるストレージ・キーの設定

強化型 DAT ファシリティがインストールされていない場合、又はファシリティはインストールされているが複数ブロック制御が 0 である場合には、上述のように、単一の 4 K バイト・ブロックのためのストレージ・キーが設定される。強化型 DAT ファシリティがインストールされており、且つ複数ブロック制御が 1 である場合には、1 メガバイト・ブロック内の複数 4 K バイト・ブロックのためのストレージ・キーを、第 2 オペランド・アドレスによって指定されるブロックから開始して、右方向に各々の逐次的なブロックに続



き、次の1メガバイト境界まで設定することができる。この場合、SET STORAGE KEY EXTENDEDは割り込み可能であり、処理は以下の通りである。

【0253】

割り込みが発生したとき（停止に続くもの以外の）、汎用レジスタR2は更新されているので、命令は、再実行されたときに割り込み時点で再開する。MRビット又はMCビットの一方又は両方が1であれば、条件コードは予測不可能である。そうでない場合には、条件コードは変更されない。

【0254】

命令が割り込みなしに完了したとき、汎用レジスタR2は次の1メガバイト境界まで更新されている。MRビット又はMCビットの一方又は両方が1であれば、条件コード3が設定される。そうでない場合には、条件コードは変更されない。

【0255】

上記の2つの場合のいずれかにおいて、MRビット又はMCビットの一方又は両方が1であれば、汎用レジスタR1のビット48 - 55は予測不可能である。

【0256】

複数ブロック処理が発生し、R1フィールド及びR2フィールドが同じレジスタを指示する場合、第2オペランド・アドレスがレジスタ内に置かれる。複数ブロック処理が24ビット又は31ビット・アドレス指定モードで発生した場合、汎用レジスタR2のビット位置32 - 63におけるアドレスの一部ではない最左端ビットは0に設定され、レジスタのビット0 - 31は変更されない。

【0257】

得られる条件コード：

条件付きSSEファシリティがインストールされていない場合、又はM3フィールドのMRビット及びMCビットの両方が0である場合には、条件コードは変更されないままである。条件付きSSEがインストールされており、且つMRビット及びMCビットの一方又は両方が1である場合には、条件コードは以下のように設定される。

0 - ストレージ・キーは設定されない

1 - ストレージ・キー全体が設定される

2 - ストレージ・キーの一部が設定される

3 - ストレージ・キー全体が設定され、汎用レジスタR1のビット48 - 55は予測不可能である。

【0258】

プログラム例外：

アドレス指定（汎用レジスタR2によって指定されるアドレス）

特権操作

【0259】

変更記録

変更記録は、主ストレージにおいてページが置換されるときにどのページを補助ストレージ内に保存しなければならないかについての情報を提供する。変更記録は、ストレージ・キーの変更ビット（ビット6）を用いる。変更ビットは、ストア・アクセスが対応するストレージ・ブロックの内容を変更させ、且つ（a）強化型DATが適用されない場合、又は（b）強化型DATが適用される場合のどちらかであって、且つ以下のうちのどちらかが真であるときに、その都度、1に設定される。即ち、

DATによって用いられるセグメント・テーブル・エントリ内のSTEフォーマット制御が0であり、且つDATによって用いられるページ・テーブル・エントリ内の変更記録オーバーライド（CO）が0である。

DATによって用いられるセグメント・テーブル・エントリ内のSTEフォーマット制御が1であり、且つDATによって用いられるセグメント・テーブル・エントリ内の変更記録オーバーライド（CO）が0である。

【0260】

10

20

30

40

50

ストレージの内容を変更しないストア・アクセスは、変更ビットを1に設定する場合も設定しない場合もある。アクセスが禁止されている場合にストアする試行については、変更ビットは1に設定されない。具体的には、

CPUについて、ストア・アクセスは、そのアクセスに対してアクセス例外が存在する場合には常に、又はそのアクセスに対するアクセス例外の優先順位より高い優先順位を有する例外が存在する場合には常に、禁止される。

チャンネル・サブシステムについて、ストア・アクセスは、そのアクセスに対してキー制御保護違反が存在する場合には常に、禁止される。

#### 【0261】

変更記録は、常にアクティブであり、いずれかのCPU（但し、本明細書で説明される、変更記録オーバーライドによって抑制される場合を除く）、いずれかのオペレータ・ファシリティ、又はチャンネル・サブシステムによって行われるストア・アクセスを含む、ストレージへの全てのストア・アクセスに対して発生する。それは、マシンによって行われる暗黙的な参照、例えば割り込みの一部である参照に対して発生する。

#### 【0262】

以下の命令のオペランドは、ストレージ位置を修正することなくストレージ・キーを直接修正するので、これらの命令のオペランドに対しては、変更記録は発生しない。

RESET REFERENCE BIT EXTENDED

SET STORAGE KEY EXTENDED（変更ビットは指定された値に設定される）

#### 【0263】

0から1に変更された変更ビットは、CPUの再試行の際に必ずしも0に復旧される必要はない。

#### 【0264】

変更記録オーバーライド（CO）

ストレージ・キーの変更ビットは、ストア・アクセスが対応するストレージ・ブロックの内容を変更させる都度、1に設定される。ストレージの内容を変更しないストア・アクセスは、変更ビットを1に設定する場合も、しない場合もある。アクセスが禁止されている場合にストアする試行については、変更ビットは1に設定されない。変更記録オーバーライドは、ストレージ・キーの変更ビットの設定をバイパスすることを可能にする。

#### 【0265】

強化型DATが適用され、且つDATテーブル・エントリによって仮想アドレスが変換される場合、セグメント・テーブル・エントリ及びページ・テーブル・エントリの両方のビット55において変更記録オーバーライド（CO）が与えられる。セグメント・テーブル・エントリのSTEフォーマット制御（FC）ビット53が0である場合には、ページ・テーブル・エントリ内の変更記録オーバーライドが適用される。PTE内の変更記録オーバーライドが0である場合には、4Kバイト・ブロックへのストア動作に対して、変更記録が発生する。変更記録オーバーライドが1である場合には、4Kバイト・ブロックへのストア動作に対して変更記録が発生するかどうかは予測不可能である。STEフォーマット制御が1である場合には、STE内の変更記録オーバーライドが適用される。STE内の変更記録オーバーライドが0である場合には、セグメントの256個の4Kバイト・ブロックのうちのいずれかへのストア動作に対して、変更記録が発生する。STE内の変更記録オーバーライドが1である場合には、セグメントの256個の4Kバイト・ブロックのうちのいずれかに対して変更記録が発生するかどうかは予測不可能である。

#### 【0266】

変更記録オーバーライドは、実アドレス若しくは絶対アドレスに対しては適用されず、又は、実空間指示によって変換される仮想アドレスに対しても適用されない。

#### 【0267】

ここで、仮想アドレスの動的アドレス変換において用いられるセグメント・テーブル・エントリから変更記録オーバーライド・フィールドを取得する一実施形態の流れ図を示す

10

20

30

40

50

、図 14 を参照する。

【 0 2 6 8 】

動的アドレス変換は、図 8 においてノード 6 1 6 のところまで説明されたように、変換に用いられるセグメント・テーブルからセグメント・テーブル・エントリがフェッチされた時点まで進行する。アドレス変換は、図 10 においてノード 8 2 2 のところまで説明されたように、ストレージ又はメモリ内の所望のデータ・ブロックに対する絶対アドレスが取得されたところまで続行する。図 14 のフローはノード 8 2 2 から開始する。

【 0 2 6 9 】

1 2 1 0 において、変更記録オーバーライド (C O) フィールドが、変換に用いられるセグメント・テーブル・エントリから取得される。変更記録オーバーライドは、ストレージ・キーの変更ビットを上書きすることを可能にする。1 2 1 2 において、変換された仮想アドレスによってアドレス指定される所望のデータ・ブロックに対して行われるアクセスがストア・タイプ又はフェッチ・タイプのどちらであるかについての判定が行われる。アクセスがフェッチ・タイプの場合、1 2 1 3 において、変更ビットの設定は発生せず、変更記録オーバーライドは適用されない。アクセスがストア・タイプ動作であった場合、1 2 1 4 において、セグメント・テーブル・エントリ内の変更記録オーバーライドがイネーブルであるかどうかについての判定が行われる。1 2 1 4 において、変更記録オーバーライドがイネーブルではない場合、1 2 1 8 において、所望のブロックに関連付けられるストレージ・キーの変更ビットは 1 に設定される。1 2 1 4 において、変更記録オーバーライドがイネーブルである場合、1 2 1 6 において、関連付けられたストレージ・キーの変更ビットは設定されない。1 2 2 0 において、所望のデータ・ブロックが修正されたことが示される。

【 0 2 7 0 】

別の実施形態において、強化型 D A T が適用され、且つ S T E 内のフォーマット制御がイネーブルでない場合、ページ・テーブルの基点がセグメント・テーブル・エントリから取得される。仮想アドレスのページ・インデックス部分を用いて、ページ・テーブル内のページ・テーブル・エントリを参照する。主ストレージ内の小データ・ブロックのページ・フレーム実アドレス、及びページ変更オーバーライド・フィールドが、ページ・テーブル・エントリから取得される。小データ・ブロックの各々は、変更ビットを含む関連付けられたストレージ・キーを有する。小データ・ブロックは、大データ・ブロックよりサイズが小さく、4 キロバイト (4 0 9 6 バイト) のデータ・ブロックから成る。主ストレージ内の所望の実データ・ブロックの変換されたアドレスは、ページ・フレーム実アドレスと仮想アドレスのバイト・インデックス部分との組み合わせを含む。所望のデータ・ブロックに対してストア・タイプ動作又はフェッチ・タイプ動作のどちらが行われたかについて判定が行われる。アクセス動作がフェッチ・タイプ動作であった場合、変更記録オーバーライドは適用されない。アクセスがストア・タイプ動作であった場合、仮想アドレスに関連付けられた変更記録オーバーライド・フィールドがディセーブルであれば、所望の 4 K ブロックに関連付けられたストレージ・キーの変更ビットは 1 に設定される。所望の 4 K ブロックが修正されたことが示される。

【 0 2 7 1 】

別の実施形態において、変更記録オーバーライド・フィールドがイネーブルであれば、所望の 4 K ブロックに関連付けられたストレージ・キーの変更ビットは、予測不可能である。

【 0 2 7 2 】

商業的な実装

本明細書においては I B M (登録商標) による z / A r h i t e c t u r e について述べているが、本発明の 1 つ又は複数の態様は、ページング可能なエンティティ又は同様の構成を使用する他のマシン・アーキテクチャ及び / 又はコンピュータ環境に対しても同等に適用可能である。

【 0 2 7 3 】

本明細書において開示される e D A T ファシリティ並びに他のフォーマット、命令及び属性の商業的な実施形態は、ハードウェア内に、又はオペレーティング・システム・プログラマのようなプログラマによって、例えば、アセンブリ言語で記述された状態で実装することができる。このようなプログラミング命令は、I B M (登録商標) S y s t e m z s e r v e r のようなコンピュータ環境において、又は代替的に他のアーキテクチャを実行するマシンにおいて、ネイティブに実行されることが意図されるストレージ媒体上にストアすることができる。命令は、既存の又は将来の I B M (登録商標) サーバにおいて、及び他のマシン又はメインフレーム上で、エミュレートすることができる。それらは、一般に実行がエミュレーション・モードであるマシンにおいて実行することができる。

【 0 2 7 4 】

10

本発明の 1 つ又は複数の態様は、例えば、仮想機械エミュレーションに対しても同等に適用可能であり、そこでは、1 つ又は複数のページング可能なエンティティ (例えば、ゲスト) を 1 つ又は複数のプロセッサ上で実行する。一例として、ページング可能なゲストは、その全体を引用によりここに組み入れる非特許文献 2 に記載される S t a r t I n t e r p r e t i v e E x e c u t i o n ( S I E ) アーキテクチャによって定義される。

【 0 2 7 5 】

エミュレーション・モードにおいては、本明細書の記載を理解した後の当業者の技術範囲内にあるように、エミュレートされる特定の命令がデコードされ、サブルーチン又はドライバの場合のように個々の命令を実装するためにサブルーチンが実行され、又は特定のハードウェアのためのドライバを提供するために他の何らかの技術が用いられる。種々のソフトウェア及びハードウェア・エミュレーション技術は、その各々が引用によりここに組み入れられる特許文献 1、特許文献 2、特許文献 3、特許文献 4、特許文献 5、及び特許文献 6 を含む、多数の米国特許に記載されている。他の多くの教示が、ターゲット・マシンのためのアーキテクチャ化された命令セットのエミュレーションを達成するための種々の手法をさらに説明している。

20

【 0 2 7 6 】

他の変形及びアーキテクチャ

本明細書において説明された種々の実施形態は、単なる例である。これらの実施形態には、本発明の精神から逸脱することなく、多数の変形が存在し得る。本発明の 1 つ又は複数の能力は、ソフトウェア、ファームウェア、ハードウェア、又はその幾つかの組み合わせで実装することができる。本発明の態様は、複数のゾーンを有する他の環境、及びパーティション化されていない環境を含む、多くの種類の環境にとって有利である。さらに、中央処理装置の複合体は存在しないかもしれないが、それでも互いに結合された複数のプロセッサが存在する場合もある。本明細書における種々の態様は、単一プロセッサ環境に対して適用可能である。

30

【 0 2 7 7 】

本明細書においては特定の環境が説明されているが、ここでもまた、本発明の精神から逸脱することなく、これらの環境の多数の変形を実装することができる。例えば、環境が論理パーティション化されている場合、より多数又はより少数の論理パーティションを環境に含めることができる。さらに、互いに結合された複数の中央処理装置複合体が存在し得る。これらは、本発明の精神から逸脱することなく行うことができる変形のごく一部である。付加的に、他の変形も可能である。

40

【 0 2 7 8 】

「ページ」という用語は、固定されたサイズ又は所定サイズのストレージ・エリアを指すために用いられているが、ページのサイズは様々であり得る。同様に、ブロックのサイズは様々であり得る。異なるサイズのブロック及び / 又はページが存在してもよい。ページはブロックと均等であり得る。他の構造を代替的に用いることができ、又は、ソフトウェア及び / 又はハードウェアを通じて、他の形で実装することができる。さらに、本明細書において説明された例において、異なるサイズのワード又はアドレス、異なるビット数

50

、異なる順序のビット、より多数の、より少数の、又は異なるビット、より多数の、より少数の、又は異なるフィールド、異なる順序のフィールド、異なるサイズのフィールドなどを含むがこれらに限定されない多数の変形が存在し得る。ここでもまた、それらは例としてのみ提供される。多数の変形が可能である。

【 0 2 7 9 】

処理ユニットは、ゲスト、ホスト、他のプロセッサ、エミュレータ、仮想マシン、及び/又は他の同様の構造のようなページング可能なエンティティを含む。バッファは、ストレージ及び/又はメモリのエリア、並びに、アレイ又はページング可能なエンティティを含むがこれらに限定されない異なる種類のデータ構造を含む。テーブルも同様に他のデータ構造を含むことができる。命令は、他のレジスタを参照することができる。さらに、ページ、セグメント、及び/又は領域は、本明細書に記載されたものとは異なる種々のサイズとすることができる。

10

【 0 2 8 0 】

本発明の1つ又は複数の態様は、例えば、コンピュータ使用可能又は機械可読媒体を有する製品（例えば、1つ又は複数のコンピュータ・プログラム製品）内に組み入れることができる。媒体は、例えば、本発明の能力を提供し、促進するためのコンピュータ可読プログラム・コード手段又は論理（例えば、命令、コード、コマンドなど）をその中に具体化する。製品は、コンピュータ・システムの一部として組み入れることもでき、又は単独で販売することもできる。さらに、機械によって読み出し可能であり、且つ、本発明の能力を実施するための、その機械によって実行可能な命令の少なくとも1つのプログラムを具体化する、少なくとも1つのプログラム・ストレージ・デバイスを提供することができる。

20

【 0 2 8 1 】

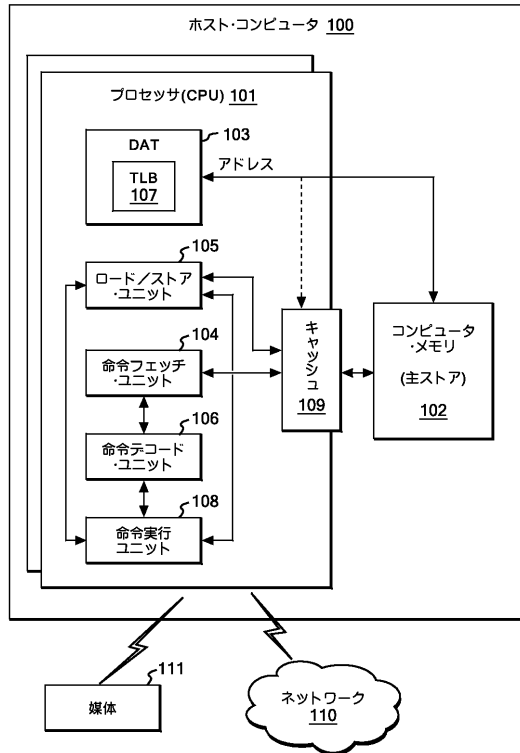
本明細書において描かれた流れ図は例示的なものである。記載された図面又はステップ若しくは動作には、本発明の精神から逸脱することなく、多数の変形が存在し得る。例えば、ステップは違う順序で実施することができ、又はステップを追加、削除若しくは修正することができる。これらの変形の全ては、特許請求される本発明の一部であると見なされる。

【 0 2 8 2 】

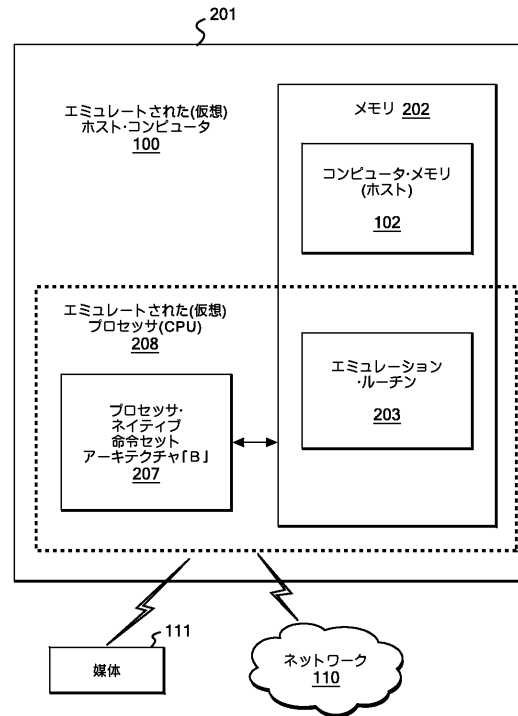
その実施形態を、本明細書において詳細に描写し、説明してきたが、本発明の精神から逸脱することなく、種々の修正、付加、置換などを行うことができること、従って、それらは添付の特許請求の範囲において定められる本発明の範囲内にあると見なされることが当業者には明らかである。

30

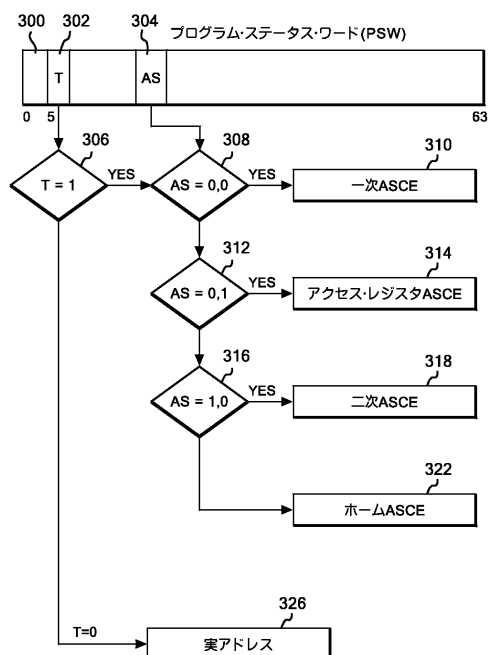
【図 1】



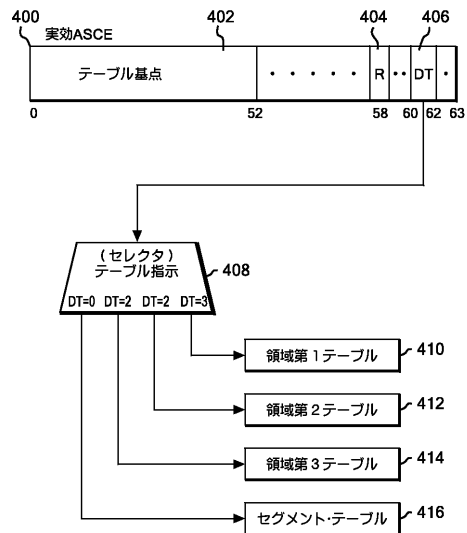
【図 2】



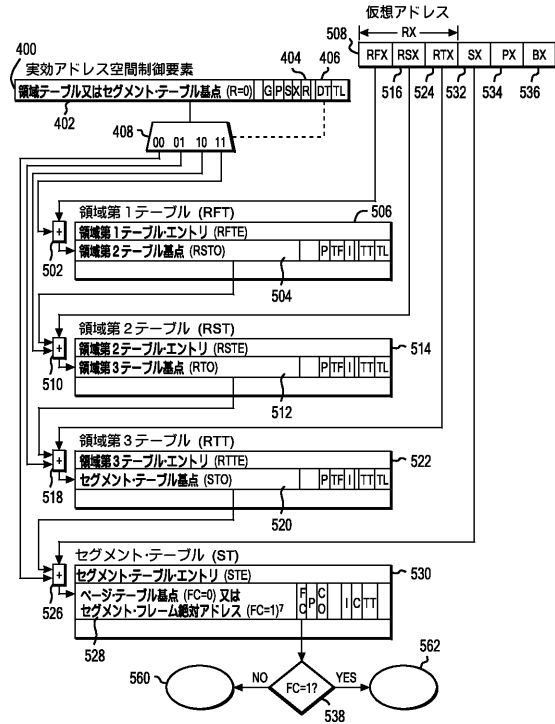
【図 3】



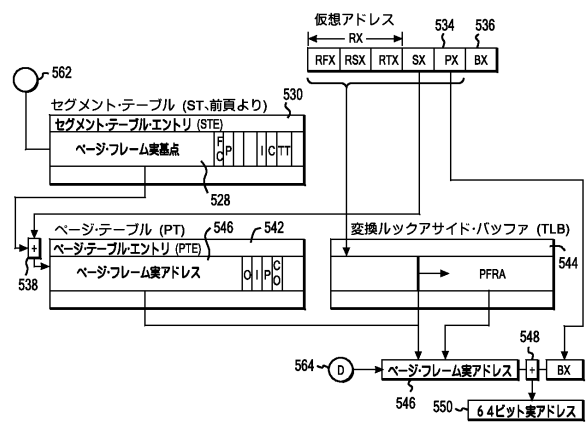
【図 4】



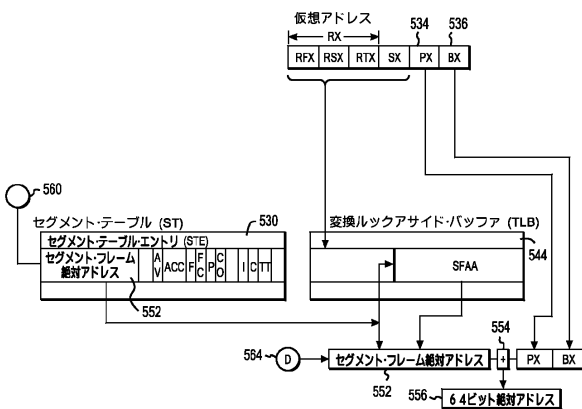
【図 5】



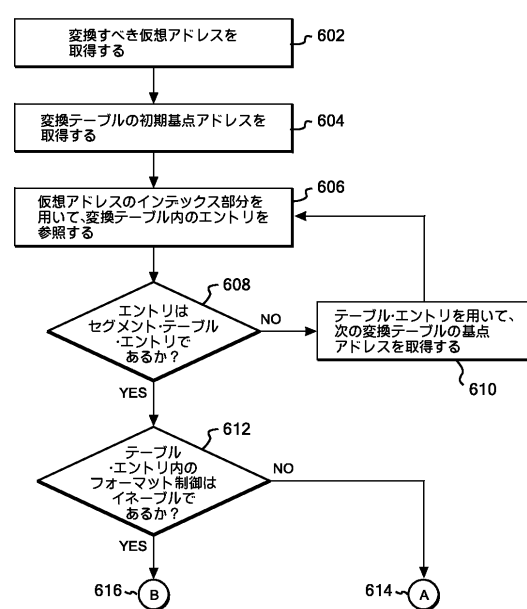
【図 6】



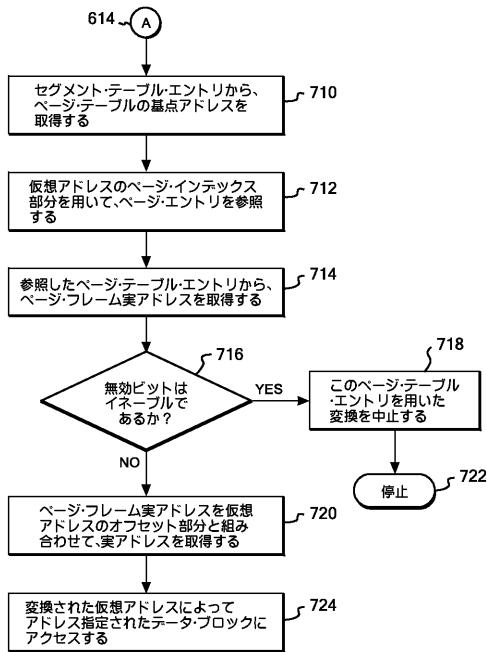
【図 7】



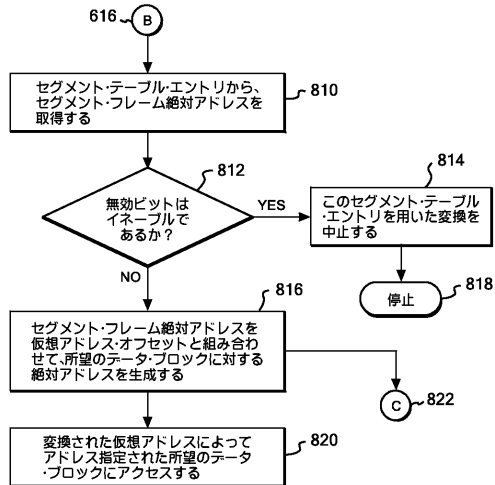
【図 8】



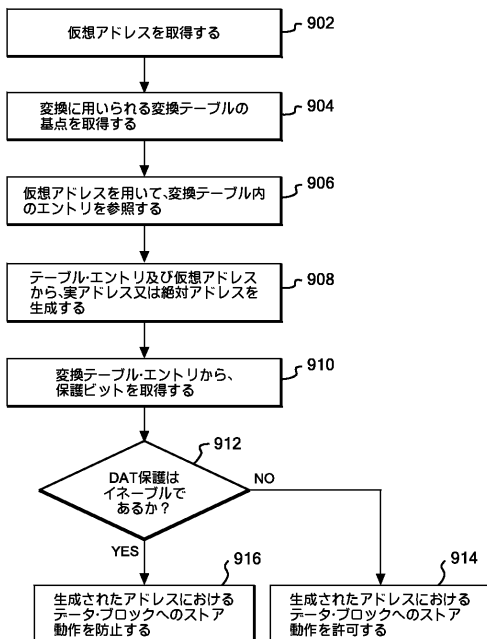
【図 9】



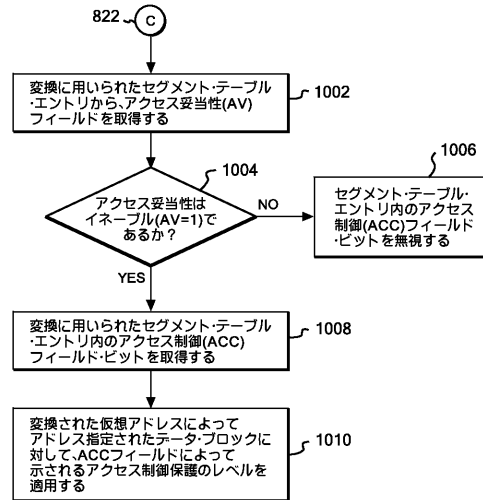
【図 10】



【図 11】

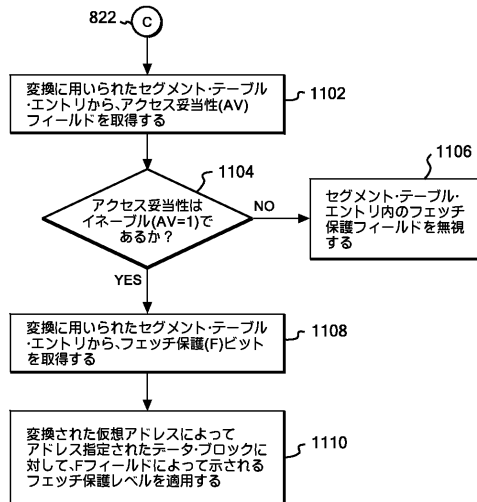


【図 12】

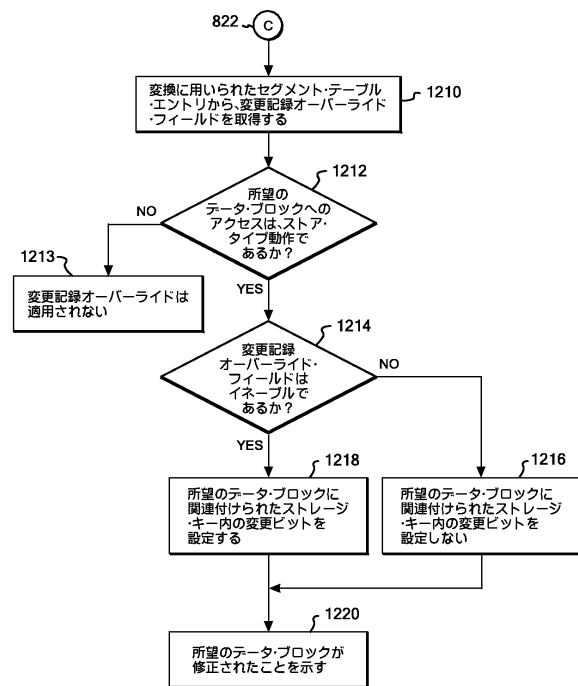




【図 13】



【図 14】



## フロントページの続き

- (72)発明者 グライナー、ダン  
アメリカ合衆国 9 5 1 2 5 - 5 0 2 1 カリフォルニア州 サンノゼ ティファニー・ウェイ  
1 6 1 5
- (72)発明者 ゲイニー、チャールズ、ジュニア  
アメリカ合衆国 1 2 6 0 3 - 1 5 2 4 ニューヨーク州 ポキプシー ジェームズ・ストリート  
2 1
- (72)発明者 ヘラー、リサ  
アメリカ合衆国 1 2 5 7 2 ニューヨーク州 ラインベック ノルウッド・ロード 2 1 5
- (72)発明者 オシセック、ダミアン  
アメリカ合衆国 1 3 8 5 0 - 3 8 5 4 ニューヨーク州 ベスタル セーレム・ドライブ 4 6  
0 5
- (72)発明者 プフェッファー、エルビン  
ドイツ連邦共和国 9 3 4 6 2 ババリア ラム ペーターミハイヴェーク 1
- (72)発明者 スレゲル、ティモシー  
アメリカ合衆国 1 2 5 8 0 ニューヨーク州 シュタッツバーグ コネリー・ドライブ 1 9
- (72)発明者 ウェブ、チャールズ  
アメリカ合衆国 1 2 5 9 0 ニューヨーク州 ワッピンガーズ・フォールズ スミス・クロッシ  
ング・ロード 4 0

審査官 岸野 徹

- (56)参考文献 特開昭 6 2 - 2 0 8 1 4 7 ( J P , A )  
特開平 1 0 - 0 9 1 5 9 7 ( J P , A )  
特開 2 0 0 4 - 3 2 6 7 7 1 ( J P , A )  
Z/ARCHITECTURE PRINCIPLES OF OPERATION, CHAPTER 3, [ONLINE] NO. SA22-7832-05, 2 0 0 7  
年 4 月, U R L , <http://publibz.boulder.ibm.com/epubs/pdf/a2278325.pdf>  
Z/ARCHITECTURE PRINCIPLES OF OPERATION, CHAPTER 3, [ONLINE] NO. SA22-7832-06, 2 0 0 8  
年 2 月, U R L , <http://publibz.boulder.ibm.com/epubs/pdf/dz9zr006.pdf>

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