

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7430195号
(P7430195)

(45)発行日 令和6年2月9日(2024.2.9)

(24)登録日 令和6年2月1日(2024.2.1)

(51)国際特許分類		F I	
G 0 6 F	9/50 (2006.01)	G 0 6 F	9/50 1 5 0 Z
G 0 6 F	9/30 (2018.01)	G 0 6 F	9/30 3 5 0 A
G 0 6 F	9/38 (2018.01)	G 0 6 F	9/38 3 7 0 C
G 0 6 F	9/455(2018.01)	G 0 6 F	9/455 1 5 0
請求項の数 14 (全75頁)			
(21)出願番号	特願2021-547172(P2021-547172)	(73)特許権者	390009531
(86)(22)出願日	令和2年2月20日(2020.2.20)		インターナショナル・ビジネス・マシー
(65)公表番号	特表2022-522113(P2022-522113		ンズ・コーポレーション
	A)		INTERNATIONAL BUSI
(43)公表日	令和4年4月14日(2022.4.14)		NESS MACHINES CORPO
(86)国際出願番号	PCT/EP2020/054518		RATION
(87)国際公開番号	WO2020/173808		アメリカ合衆国10504 ニューヨー
(87)国際公開日	令和2年9月3日(2020.9.3)		ク州 アーモンク ニュー オーチャード
審査請求日	令和4年7月25日(2022.7.25)		ロード
(31)優先権主張番号	16/286,990		New Orchard Road, A
(32)優先日	平成31年2月27日(2019.2.27)		rmonk, New York 105
(33)優先権主張国・地域又は機関	米国(US)		04, United States of
			America
		(74)代理人	100112690
			弁理士 太佐 種一
		最終頁に続く	

(54)【発明の名称】 複数のマシン世代にわたる複合体の機能の互換性の維持

(57)【特許請求の範囲】
【請求項1】

第1の世代のマシンおよび第2の世代のマシンを含んでいる複数のマシンを備えているシステムであって、前記複数のマシンの各々がマシン・バージョンを含んでおり、前記第1の世代のマシンが第1の仮想マシンおよび仮想アーキテクチャ・レベルを実行し、前記第2の世代のマシンが第2の仮想マシンおよび前記仮想アーキテクチャ・レベルを実行し、前記仮想アーキテクチャ・レベルが、複合体の割り込み可能命令の互換性レベルを前記第1および第2の仮想マシンに提供し、前記互換性レベルが、前記複数のマシンごとの、前記複合体の割り込み可能命令のためのローカルなパラメータ・ブロック形式を含み、前記互換性レベルが、前記複数のマシン全体の最小公分母のマシン・バージョンのために設計され、前記互換性レベルが、前記最小公分母のマシン・バージョンを識別する最小公分母インジケータを含んでおり、

前記最小公分母のマシン・バージョンは、前記最小公分母のマシン・バージョンを含む前記複数マシンの一つから、前記複数のマシンの残りのマシンに伝搬し、

前記伝搬に基づき、前記複数のマシンの残りのマシンは、前記伝搬された最小公分母のマシン・バージョンに対応するパラメータ・ブロック形式を有するアーキテクチャ固有の命令として、前記複合体の割り込み可能命令を実装し、

アーキテクチャ固有の命令は前記複数のマシンの残りのマシンで実行され、前記アーキテクチャ固有の命令はパラメータ・ブロックにおいて機能コードで指定され、パラメータ・ブロックのサイズは機能により決まっており、前記アーキテクチャ固有の命令の動作は

、照会、生成、圧縮および展開の1つであり、

前記実行の完了に基づき、前記実行から得られた条件コードが決定され、

完了または更なる動作処理の一つが条件コードに基づき実行される、

システム。

【請求項2】

前記最小公分母インジケータが、前記複合体の割り込み可能命令を最初に行うため
に、前記複数のマシンのうちの1つのマシンによって生成される、請求項1に記載のシス
テム。

【請求項3】

前記最小公分母インジケータが、前記複合体の割り込み可能命令を最初に行うため
の前記複数のマシンのうちの1つのマシンから、前記複数のマシンのうちの残りの複数の
マシンに伝搬される、請求項1に記載のシステム。

【請求項4】

前記互換性レベル内の前記最小公分母インジケータが、特定の仮想マシンにおいてどの
機能が使用可能であるかを識別する一連の機能ビットによって制御される、請求項1に記
載のシステム。

【請求項5】

前記複合体の割り込み可能命令がデフレート変換呼び出し命令を含む、請求項1ないし
4のいずれかに記載のシステム。

【請求項6】

前記複合体の割り込み可能命令が、アクセラレータ上で実行されている複合体の命令セ
ットの命令を含む、請求項1に記載のシステム。

【請求項7】

複数のマシンのうちの第1の世代のマシンによって第1の仮想マシンおよび仮想アーキ
テクチャ・レベルを実行することであって、前記複数のマシンが前記第1の世代のマシン
および第2の世代のマシンを含み、前記複数のマシンの各々がマシン・バージョンを含む
、前記実行することと、

前記第2の世代のマシンによって第2の仮想マシンおよび前記仮想アーキテクチャ・レ
ベルを実行することと、

前記仮想アーキテクチャ・レベルによって、複合体の割り込み可能命令の互換性レベル
を前記第1および第2の仮想マシンに提供することを含んでいる方法であって、前記互
換性レベルが、前記複数のマシンごとの、前記複合体の割り込み可能命令のためのローカ
ルなパラメータ・ブロック形式を含み、前記互換性レベルが、前記複数のマシン全体の最
小公分母のマシン・バージョンのために設計され、前記互換性レベルが、前記最小公分母
のマシン・バージョンを識別する最小公分母インジケータを含んでおり、

前記最小公分母のマシン・バージョンは、前記最小公分母のマシン・バージョンを含む
前記複数マシンの一つから、前記複数のマシンの残りのマシンに伝搬し、

前記伝搬に基づき、前記複数のマシンの残りのマシンは、前記伝搬された最小公分母の
マシン・バージョンに対応するパラメータ・ブロック形式を有するアーキテクチャ固有の
命令として、前記複合体の割り込み可能命令を実装し、

アーキテクチャ固有の命令は前記複数のマシンの残りのマシンで実行され、前記アー
キテクチャ固有の命令はパラメータ・ブロックにおいて機能コードで指定され、パラメー
タ・ブロックのサイズは機能により決まっており、前記アーキテクチャ固有の命令の動作
は、照会、生成、圧縮および展開の1つであり、

前記実行の完了に基づき、前記実行から得られた条件コードが決定され、

完了または更なる動作処理の一つが条件コードに基づき実行される、

方法。

【請求項8】

前記最小公分母インジケータが、前記複合体の割り込み可能命令を最初に行うため
に、前記複数のマシンのうちの1つのマシンによって生成される、請求項7に記載の方法。

10

20

30

40

50

【請求項 9】

前記最小公分母インジケータが、前記複合体の割り込み可能命令を最初に実行するための前記複数のマシンのうちの1つのマシンから、前記複数のマシンのうちの残りの複数のマシンに伝搬される、請求項 7 に記載の方法。

【請求項 10】

前記互換性レベル内の前記最小公分母インジケータが、特定の仮想マシンにおいてどの機能が使用可能であるかを識別する一連の機能ビットによって制御される、請求項 7 に記載の方法。

【請求項 11】

前記複合体の割り込み可能命令がデフレート変換呼び出し命令を含む、請求項 7 に記載の方法。

【請求項 12】

前記複合体の割り込み可能命令が、アクセラレータ上で実行されている複合体の命令セットの命令を含む、請求項 7 に記載の方法。

【請求項 13】

コンピュータ・プログラムであって、請求項 7 ないし 12 のいずれか 1 項に記載の方法の各ステップをコンピュータに実行させるための、コンピュータ・プログラム。

【請求項 14】

請求項 13 に記載のコンピュータ・プログラムを記録した、コンピュータ可読ストレージ媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、一般に、複数のマシン世代にわたる複合体の機能の互換性の維持に関連している。

【背景技術】

【0002】

異なるマシン世代のマシン間のワークロードの移行を可能にすることに関して、各マシンは、別のマシンによって開始された作業を続行できる必要がある。例えば、システムが異なるコンピュータ間で移行できる環境内で実行している場合、それらの環境間に共通のアーキテクチャ・レベルが存在する必要がある。アーキテクチャ間で変化するもののある複合体のパラメータ・ブロック形式を伴う命令の場合、マシンは、それらの環境間の共通のアーキテクチャ・レベルを使用して、前の各世代のパラメータ・ブロック形式を作成して利用できる必要がある。しかし、前の各世代のパラメータ・ブロック形式を作成して利用することは、実装および検証の複雑さの原因となる可能性がある。例えば、5つの世代にわたって異なるパラメータ・ブロック定義が現れる複合体の命令の場合、最新の世代のマシンは、5つの異なるパラメータ・ブロックを理解し、生成する必要がある。

【発明の概要】

【0003】

本発明の実施形態に従って、複数のマシンを含むシステムが提供される。これらの複数のマシンは、第1の世代のマシンおよび第2の世代のマシンを含む。複数のマシンの各々は、マシン・バージョンを含む。第1の世代のマシンは、第1の仮想マシンおよび仮想アーキテクチャ・レベルを実行する。第2の世代のマシンは、第2の仮想マシンおよび仮想アーキテクチャ・レベルを実行する。仮想アーキテクチャ・レベルは、複合体の割り込み可能命令のための互換性レベルを第1および第2の仮想マシンに提供する。互換性レベルは、複数のマシン全体の最小公分母のマシン・バージョンのために設計される。互換性レベルは、最小公分母のマシン・バージョンを識別する最小公分母インジケータを含む。本明細書における実施形態の技術的効果および利点は、複合体の命令／機能に関して、各マシン世代が、最大で2つの異なるパラメータ・ブロック（それ自身のパラメータ・ブロックおよび互換性レベル）のみをサポートする場合に、検証および妥当性確認が、パラメー

10

20

30

40

50

タ・ブロックの現在のバージョンおよび互換性バージョンのみに対するテストに減らされることを含むことができる。

【0004】

本発明の1つまたは複数の実施形態または上記のシステムの実施形態に従って、互換性レベルは、複数のマシンごとの、複合体の割り込み可能命令のためのローカルなパラメータ・ブロック形式を含み、ローカルなパラメータ・ブロック形式は、複数のマシンの各々のローカルなマシン・バージョンのために設計される。

【0005】

本発明の1つまたは複数の実施形態または上記のシステムの実施形態のいずれかに従って、最小公分母インジケータは、複合体の割り込み可能命令を最初に実行するために、複数のマシンのうちの1つのマシンによって生成される。

10

【0006】

本発明の1つまたは複数の実施形態または上記のシステムの実施形態のいずれかに従って、最小公分母インジケータは、複合体の割り込み可能命令を最初に実行するための複数のマシンのうちの1つのマシンから、複数のマシンのうちの残りの複数のマシンに伝搬される。本明細書における実施形態の技術的効果および利点は、第1の世代の場合、そのパラメータ・ブロック形式が「発生」形式を表し、1つの形式のみがサポート／テストされる必要があるということを含むことができる。したがって、将来のマシン世代に対する下流の影響を引き起こさずに、その後のすべての世代のマシン固有のパラメータ・ブロックの内容が、そのマシンに対して最適化され得る。

20

【0007】

本発明の1つまたは複数の実施形態または上記のシステムの実施形態のいずれかに従って、互換性レベル内の最小公分母インジケータは、特定の仮想マシンにおいてどの機能が使用可能であるかを識別する一連の機能ビットによって制御される。

【0008】

本発明の1つまたは複数の実施形態または上記のシステムの実施形態のいずれかに従って、複合体の割り込み可能命令は、デフレート変換呼び出し命令を含む。

【0009】

本発明の1つまたは複数の実施形態または上記のシステムの実施形態のいずれかに従って、複合体の割り込み可能命令は、アクセラレータ上で実行される複合体の命令セットの命令を含む。

30

【0010】

本発明の1つまたは複数の実施形態に従って、上記のシステムの実施形態のいずれかは、方法またはコンピュータ・プログラム製品として実装され得る。

【0011】

1つまたは複数の実施形態に従って、第1のマシンによって複合体の割り込み可能命令を実装し、仮想アーキテクチャ・レベルにおいて、第1のマシンのパラメータ・ブロックがこの複合体の割り込み可能命令のためにサポートされることを示す方法が提供される。または、第1のマシンのパラメータ・ブロックは、複体内で実行されている各仮想マシンに伝搬される。本明細書における実施形態の技術的効果および利点は、検証および妥当性確認を、パラメータ・ブロックの現在のバージョンおよび互換性バージョンに対するテストに減らすことを含むことができる。

40

【0012】

本発明の1つまたは複数の実施形態または上記の方法の実施形態に従って、この方法は、新しいマシンがオンラインであることを検出することと、第1のマシンのパラメータ・ブロックを複体のマシン・バージョン全体の最小公分母として識別することと、第1のマシンのパラメータ・ブロックを新しいマシンに伝搬することとを含むことができる。

【0013】

本発明の1つまたは複数の実施形態に従って、上記の方法の実施形態のいずれかは、システムまたはコンピュータ・プログラム製品として実装され得る。

50

【0014】

その他の技術的特徴および利点が、本発明の手法によって実現される。本発明の実施形態および態様は、本明細書において詳細に説明され、請求される対象の一部と見なされる。さらに良く理解するために、詳細な説明および図面を参照すること。

【0015】

本明細書に記載された専有権の詳細は、本明細書の最後にある特許請求の範囲において具体的に指摘され、明確に請求される。本発明の各実施形態の前述およびその他の特徴と長所は、添付の図面と共に行われる以下の詳細な説明から明らかになる。

【図面の簡単な説明】

【0016】

【図1A】本発明の1つまたは複数の態様を組み込んで使用するためのコンピューティング環境の一例を示す図である。

【図1B】本発明の1つまたは複数の態様に従って、図1Aのプロセッサの詳細をさらに示す図である。

【図2】本発明の1つまたは複数の態様を組み込んで使用するためのコンピューティング環境の別の例を示す図である。

【図3A】本発明の態様に従って、デフレート変換呼び出し（D F L T C C : DEFLATE Conversion Call）命令の1つの形式を示す図である。

【図3B】本発明の態様に従って、デフレート変換呼び出し命令によって使用される暗黙のレジスタ（汎用レジスタ0）のフィールドの一例を示す図である。

【図3C】本発明の態様に従って、デフレート変換呼び出し命令の機能コードの一例を示す図である。

【図3D】本発明の態様に従って、デフレート変換呼び出し命令によって使用される暗黙のレジスタ（汎用レジスタ1）のフィールドの一例を示す図である。

【図3E】本発明の態様に従って、デフレート変換呼び出し命令によって指定されるレジスタ（R1）の内容の一例を示す図である。

【図3F】本発明の態様に従って、デフレート変換呼び出し命令によって使用されるレジスタ（R1+1）の内容の一例を示す図である。

【図3G】本発明の態様に従って、デフレート変換呼び出し命令によって指定されるレジスタ（R2）の内容の一例を示す図である。

【図3H】本発明の態様に従って、デフレート変換呼び出し命令によって使用されるレジスタ（R2+1）の内容の一例を示す図である。

【図3I】本発明の態様に従って、デフレート変換呼び出し命令によって指定されるレジスタ（R3）の内容の一例を示す図である。

【図3J】本発明の態様に従って、デフレート変換呼び出し命令のD F L T C C - Q A F（使用可能な機能の照会）機能によって使用されるパラメータ・ブロックの内容の一例を示す図である。

【図3K】本発明の態様に従って、デフレート変換呼び出し命令のD F L T C C - G D H T（動的ハフマン・テーブルの生成）機能によって使用されるパラメータ・ブロックの内容の一例を示す図である。

【図3L】本発明の態様に従って、デフレート変換呼び出し命令のD F L T C C - C M P R（圧縮）機能およびD F L T C C - X P N D（展開）機能によって使用されるパラメータ・ブロックの内容の一例を示す図である。

【図4】本発明の1つまたは複数の態様に従って、サブバイト境界（sub-byte boundary）の一例を示す図である。

【図5A】本発明の態様に従って、D F L T C C - C M P R機能へのサブバイト境界の適用方法を説明する例を示す図である。

【図5B】本発明の態様に従って、D F L T C C - C M P R機能へのサブバイト境界の適用方法を説明する例を示す図である。

【図5C】本発明の態様に従って、D F L T C C - C M P R機能へのサブバイト境界の適

10

20

30

40

50

用方法を説明する例を示す図である。

【図 6】本発明の態様に従って、圧縮されていないデータのブロックの一例を示す図である。

【図 7】本発明の態様に従って、固定ハフマン・テーブル（FHT：fixed-Huffman table）を使用して圧縮されたデータを含むブロックの一例を示す図である。

【図 8】本発明の態様に従って、動的ハフマン・テーブル（DHT：dynamic-Huffman table）を使用して圧縮されたデータを含むブロックの一例を示す図である。

【図 9】本発明の態様に従って、ストレージ内の圧縮データ・セットの一例を示す図である。

【図 10】本発明の態様に従って、圧縮データ・セットの 3 つのブロックにデータを圧縮するプログラムのサンプルの一例を示す図である。

10

【図 11】本発明の態様に従って、セットの第 1 の圧縮データ・ブロックに対して動作する DFLTCC-CMPR 機能のパラメータ・ブロックの内容の一例を示す図である。

【図 12】本発明の態様に従って、セットの第 2 の圧縮データ・ブロックに対して動作する DFLTCC-CMPR 機能のパラメータ・ブロックの内容の一例を示す図である。

【図 13】本発明の態様に従って、圧縮データ・セットからデータを復元するプログラムのサンプルの一例を示す図である。

【図 14 A】本発明の態様に従って、DFLTCC-CMPR を複数回実行する前後の直列履歴バッファ（in-line history buffer）の例を示す図である。

【図 14 B】本発明の態様に従って、DFLTCC-CMPR を複数回実行する前後の直列履歴バッファの例を示す図である。

20

【図 14 C】本発明の態様に従って、DFLTCC-CMPR を複数回実行する前後の直列履歴バッファの例を示す図である。

【図 15 A】本発明の態様に従って、DFLTCC を複数回実行する前後の円形履歴バッファ（circular history buffer）の例を示す図である。

【図 15 B】本発明の態様に従って、DFLTCC を複数回実行する前後の円形履歴バッファの例を示す図である。

【図 15 C】本発明の態様に従って、DFLTCC を複数回実行する前後の円形履歴バッファの例を示す図である。

【図 15 D】本発明の態様に従って、DFLTCC を複数回実行する前後の円形履歴バッファの例を示す図である。

30

【図 15 E】本発明の態様に従って、DFLTCC を複数回実行する前後の円形履歴バッファの例を示す図である。

【図 16 A】本発明の態様に従って、DFLTCC-XPND を複数回実行する前後の直列履歴バッファの例を示す図である。

【図 16 B】本発明の態様に従って、DFLTCC-XPND を複数回実行する前後の直列履歴バッファの例を示す図である。

【図 16 C】本発明の態様に従って、DFLTCC-XPND を複数回実行する前後の直列履歴バッファの例を示す図である。

【図 17】本発明の態様に従って、デフレート変換呼び出し命令の使用の一例を示す図である。

40

【図 18】本発明の態様に従って、円形履歴バッファの使用の一例を示す図である。

【図 19】本発明の実施形態に従って、システムを示す図である。

【図 20】本発明の実施形態に従って、プロセス・フローを示す図である。

【図 21】本発明の実施形態に従って、システムを示す図である。

【図 22】1 つまたは複数の実施形態に従って、処理システムを示す図である。

【発明を実施するための形態】

【0017】

本明細書において示される図は、実例である。本発明の趣旨から逸脱することなく、本明細書に記載された図または動作の多くの変形が存在することが可能である。例えば、動

50

作は異なる順序で実行されることが可能であり、あるいは動作は追加、削除、または変更されることが可能である。また、「結合される」という用語およびその変形は、2つの要素間に通信経路が存在することを表しており、それらの要素間に要素／接続が介在しない要素間の直接的接続を意味していない。これらのすべての変形は、本明細書の一部であると見なされる。

【0018】

添付の図および開示された実施形態の以下の詳細な説明では、図に示されたさまざまな要素が、2桁または3桁の参照番号付きで提供されている。わずかな例外を除いて、各参照番号の左端の数字は、その要素が最初に示された図に対応している。

【0019】

1つまたは複数の実施形態に従って、本発明は、任意の仮想マシンが複合体の割り込み可能命令を利用できるようにする、複合体の割り込み可能命令のための互換性レベルを含む仮想アーキテクチャ・レベルに関連している。これに関して、互換性レベルは、すべての仮想マシン全体の最小公分母のマシン・バージョンのために設計され、最小公分母インジケータがこのマシン・バージョンを識別する。

【0020】

本発明の態様に従って、コンピューティング環境内の処理を容易にするための能力が提供される。1つの例として、データの圧縮または復元 (decompress) (復元 (uncompress) とも呼ばれる) あるいはその両方を実行するための圧縮機能または復元機能などの機能 (動作とも呼ばれる) を実行するために、単一の命令 (例えば、ハードウェア／ソフトウェア・インターフェイスでの単一の設計されたハードウェア・マシン命令) が提供される。この命令は、汎用プロセッサ上でプログラム (例えば、オペレーティング・システムまたはユーザ・プログラム) によってディスパッチされる、汎用プロセッサの命令セット・アーキテクチャ (ISA: instruction set architecture) の一部である。ISA命令を使用して圧縮／復元を実行することによって、圧縮動作／復元動作を実行するためにオペレーティング・システムによるタスク切り替えが不要になり、それによって、実行サイクルを省く。さらに、単一の命令を使用してデータの圧縮または復元あるいはその両方を実行することによって、汎用プロセッサなどのプロセッサ内の実行時間が短縮される。

【0021】

1つの例では、この命令は、デフレート規格と呼ばれる業界標準に準拠する圧縮動作および復元動作を実行し、この命令はデフレート変換呼び出し命令と呼ばれる。デフレート規格は、データの元の形態 (データの圧縮されていない形態) で重複列を表す圧縮データ・シンボルの説明を含む。そのようなシンボルは、処理されているデータの現在の位置との関係において、以前に処理された重複列の位置および長さを表す重複列のポインタおよび長さを含む。データの以前に処理された圧縮されていない形態は、履歴と呼ばれる。1つの例では、履歴は、メモリ内の連続的な複数のバイトであり、例えば32Kバイト程度の大きさであることができる。

【0022】

本発明の1つまたは複数の態様を組み込んで使用するためのコンピューティング環境の一実施形態が、図1Aを参照して説明される。コンピューティング環境100は、例えば、1つまたは複数のバス108またはその他の接続あるいはその両方を介して互いに結合された、例えば、プロセッサ102 (例えば、中央処理装置) と、メモリ104 (例えば、メイン・メモリであり、システム・メモリ、主記憶装置、中央記憶装置、ストレージとも呼ばれる) と、1つまたは複数の入出力 (I/O) デバイスまたはインターフェイス106あるいはその両方を含む。

【0023】

1つの例では、プロセッサ102は、International Business Machines Corporation (ニューヨーク州アーモンク市) によって提供されるz/Architecture (R) ハードウェア・アーキテクチャに基づき、やはりInternational Business Machines Corpor

10

20

30

40

50

ationによって提供されてz/Architectureハードウェア・アーキテクチャを実装するIBM Z (R) サーバなどのサーバに含まれる。z/Architectureハードウェア・アーキテクチャの一実施形態は、“z/Architecture Principles of Operation,” IBM Publication No. SA22-7832-11, 12th edition, September 2017という公開文献に記載されており、この公開文献は本明細書において参照によってその全体が本明細書に組み込まれている。しかし、z/Architectureハードウェア・アーキテクチャは1つの例示的なアーキテクチャにすぎず、他のアーキテクチャまたは他の種類のコンピューティング環境あるいはその両方が、本発明の1つまたは複数の態様を含むか、または使用するか、あるいはその両方であってよい。1つの例では、プロセッサは、やはりInternational Business Machines Corporationによって提供されるz/OS (R) オペレーティング・システムなどのオペレーティング・システムを実行する。

【0024】

プロセッサ102は、命令を実行するために使用される複数の機能コンポーネントを含む。図1Bに示されているように、これらの機能コンポーネントは、例えば、実行される命令をフェッチするための命令フェッチ・コンポーネント120と、フェッチされた命令をデコードするため、およびデコードされた命令のオペランドを取得するための命令デコード・ユニット122と、デコードされた命令を実行するための命令実行コンポーネント124と、必要な場合に、命令を実行するためにメモリにアクセスするためのメモリ・アクセス・コンポーネント126と、実行された命令の結果を提供するための書き戻しコンポーネント130とを含む。これらのコンポーネントのうちの1つまたは複数のは、本発明の1つまたは複数の態様に従って、本明細書において説明されているように、圧縮処理／復元処理（または本発明の1つまたは複数の態様を使用できるその他の処理）で使用される1つまたは複数の他のコンポーネントの少なくとも一部を含むか、またはそれらのコンポーネントにアクセスすることができてよい。1つまたは複数の他のコンポーネントは、例えば、圧縮コンポーネント／復元コンポーネント（またはその他のコンポーネント）136を含む。

【0025】

本発明の1つまたは複数の態様を組み込んで使用するためのコンピューティング環境の別の例が、図2を参照して説明される。1つの例では、コンピューティング環境はz/Architectureハードウェア・アーキテクチャに基づくが、コンピューティング環境は、International Business Machines Corporationまたはその他の企業によって提供される他のアーキテクチャに基づいてよい。

【0026】

図2を参照すると、1つの例では、コンピューティング環境は中央電子回路複合体（CEC：central electronics complex）200を含む。CEC200は、例えば、1つまたは複数のプロセッサ（中央処理装置（CPU：central processing unit）とも呼ばれる）204および入出力サブシステム206に結合されたメモリ202（システム・メモリ、メイン・メモリ、主記憶装置、中央記憶装置、ストレージとも呼ばれる）などの、複数のコンポーネントを含んでいる。

【0027】

メモリ202は、例えば、1つまたは複数の論理パーティション208、論理パーティションを管理するハイパーバイザ210、およびプロセッサ・ファームウェア212を含んでいる。ハイパーバイザ210の1つの例は、International Business Machines Corporation（ニューヨーク州アーモンク市）によって提供されるプロセッサ・リソース／システム管理機構（PR/SM（TM）：Processor Resource/System Manager）ハイパーバイザである。本明細書において使用されるとき、ファームウェアは、例えば、プロセッサのマイクロコードを含む。ファームウェアは、例えば、上位レベルのマシン・コードの実装において使用される、ハードウェア・レベルの命令またはデータ構造あるいはその両方を含む。1つの実施形態では、ファームウェア

ムウェアは、例えば、信頼できるソフトウェアを含んでいるマイクロコード、または基盤になるハードウェアに固有のマイクロコードとして通常は提供される、システムのハードウェアへのオペレーティング・システムのアクセスを制御する独自のコードを含む。

【0028】

各論理パーティション208は、別々のシステムとして機能することができる。すなわち、各論理パーティションは、独立してリセットされて、 π / OSオペレーティング・システムなどのゲスト・オペレーティング・システム220または別のオペレーティング・システムを実行し、異なるプログラム222と共に動作することができる。論理パーティション内で実行されるオペレーティング・システムまたはアプリケーション・プログラムは、完全なシステム全体にアクセスできるように見えるが、実際は、その一部のみが利用可能である。

10

【0029】

メモリ202は、論理パーティションに割り当てることができる物理プロセッサ・リソースであるプロセッサ（例えば、CPU）204に結合される。例えば、論理パーティション208は、1つまたは複数の論理プロセッサを含み、それらの論理プロセッサの各々は、論理パーティションに動的に割り当てることができる物理プロセッサ・リソース204の全部または一部を表す。

【0030】

さらに、メモリ202は、I / Oサブシステム206に結合される。I / Oサブシステム206は、中央電子回路複合体の一部であるか、または中央電子回路複合体から分離してよい。I / Oサブシステム206は、主記憶装置202と、中央電子回路複合体に結合された入出力制御ユニット230および入出力（I / O）デバイス240との間の情報の流れを方向付ける。

20

【0031】

多くの種類のI / Oデバイスが使用されてよい。1つの特定の種類は、データ・ストレージ・デバイス250である。データ・ストレージ・デバイス250は、1つまたは複数のプログラム252、1つまたは複数のコンピュータ可読プログラム命令254、またはデータ、あるいはその組み合わせなどを格納してよい。コンピュータ可読プログラム命令は、本発明の態様の実施形態の機能を実行するように構成されてよい。

【0032】

1つの例として、各プロセッサ204は、1つまたは複数のローカル・キャッシュまたは1つまたは複数の共有キャッシュあるいはその両方などの、複数のレベルのキャッシュを含んでいるキャッシュ階層の少なくとも1つのキャッシュ260（例えば、ローカル・キャッシュ）を含む。さらに、1つの実施形態では、ローカル・キャッシュおよびメモリ202が、データの圧縮または復元あるいはその両方（または本発明の1つまたは複数の態様の他の動作あるいはその両方）のうちの1つまたは複数の実行において使用される圧縮コンポーネント／復元コンポーネント（またはその他のコンポーネント）262に結合される。さまざまな例では、これらのタスクを実行する1つまたは複数のコンポーネントが存在してよい。多くの変形が可能である。

30

【0033】

1つの実施形態では、プロセッサ（例えば、プロセッサ204）が命令（例えば、デフレート変換呼び出し命令）を取得し、命令をデコードし、命令によって使用されるアドレスの変換を含む、命令の設定を実行し、命令のコマンドを、コンポーネント262などのプロセッサに結合されたコンポーネントに送信し、命令によって指定された機能を実行する。コンポーネント262は、指定された機能の実行において、データを読み込んで処理し、処理されたデータを再び格納するように、キャッシュ階層およびメモリにアクセスすることができる。一例として、コンポーネント262は、ハードウェア・コンポーネントである。

40

【0034】

さらに別の実施形態では、コンポーネント262の少なくとも一部が、プロセッサの一

50

部として含まれる。多くの変形が可能である。

【0035】

中央電子回路複合体200は、取り外し可能／取り外し不可、揮発性／不揮発性のコンピュータ・システム・ストレージ媒体を含むか、またはそのようなコンピュータ・システム・ストレージ媒体に結合されるか、あるいはその両方であってよい。例えば、中央電子回路複合体200は、取り外し不可、不揮発性の磁気媒体（通常は、「ハード・ドライブ」と呼ばれる）、取り外し可能、不揮発性の磁気ディスク（例えば、「フロッピー（R）・ディスク」）に対する読み取りと書き込みを行うための磁気ディスク・ドライブ、あるいはCD-ROM、DVD-ROM、またはその他の光媒体などの取り外し可能、不揮発性の光ディスクに対する読み取りと書き込みを行うための光ディスク・ドライブ、あるいはその組み合わせを含むか、またはこれらに結合されるか、あるいはその両方であってよい。その他のハードウェア・コンポーネントまたはソフトウェア・コンポーネントあるいはその両方を、中央電子回路複合体200と併用できるということが理解されるべきである。その例として、マイクロコード、デバイス・ドライバ、冗長プロセッシング・ユニット、外部ディスク・ドライブ・アレイ、RAIDシステム、テープ・ドライブ、およびデータ・アーカイブ・ストレージ・システムなどが挙げられるが、これらに限定されない。

10

【0036】

さらに、中央電子回路複合体200は、他の多数の汎用または専用のコンピューティング・システム環境または構成で運用されてよい。中央電子回路複合体200での使用に適している可能性がある周知のコンピューティング・システム、環境、または構成、あるいはその組み合わせの例としては、パーソナル・コンピュータ（PC：personal computer）システム、サーバ・コンピュータ・システム、シン・クライアント、シック・クライアント、ハンドヘルドまたはラップトップ・デバイス、マイクロプロセッサ・システム、マイクロプロセッサベース・システム、セット・トップ・ボックス、プログラマブル・コンシューマ・エレクトロニクス、ネットワークPC、マイクロコンピュータ・システム、メインフレーム・コンピュータ・システム、およびこれらの任意のシステムまたはデバイスを含む分散クラウド・コンピューティング環境などが挙げられるが、これらに限定されない。

20

【0037】

本明細書ではコンピューティング環境のさまざまな例が説明されるが、本発明の1つまたは複数の態様が、多くの種類の環境と共に使用されてよい。本明細書で提供されるコンピューティング環境は、例にすぎない。

30

【0038】

本発明の態様に従って、コンピューティング環境100または中央電子回路複合体200などのコンピューティング環境は、データを圧縮および復元するためのメカニズムを提供する変換機能を採用する。1つの例では、変換機能は、デフレート圧縮データ形式を使用してデータを圧縮および復元するためのメカニズムを提供するデフレート変換機能である。1つの例では、変換機能は、機能インジケータが例えば1に設定されたときに、システムにインストールされる。z／Architectureハードウェア・アーキテクチャの1つの特定の例として、z／Architectureアーキテクチャ・モードで変換機能がインストールされたときに、機能ビット151が例えば1に設定される。この機能は、例えば、デフレート変換呼び出し命令を含み、デフレート変換呼び出し命令の実施形態が以下で説明される。

40

【0039】

1つの例では、デフレート変換呼び出し命令は、データの元の（圧縮されていない）形態と、IETF（インターネット・エンジニアリング・タスク・フォース）RFC（コメント要求）1951仕様（DEFLATE Compressed Data Format Specification version 1.3 Internet Engineering Task Force, Request for Comments 1951, May 1996に記載されている）などの選択された規格によって規定されたデータの圧縮表現との間でデータの状態を変換することに関連する機能を実行する。

50

【0040】

1つの例では、圧縮されていないデータはバイトのシーケンスであり、データの圧縮表現はシンボルを含む。シンボルは、圧縮されていないデータの個別のバイト（リテラル・バイトと呼ばれる）を表すか、または圧縮されていないデータの繰り返されるバイトのシーケンス（重複列と呼ばれる）を表す。一例として、ハフマン・テーブルは、圧縮データ・シンボルと圧縮されていないデータの間のエンコーディングおよびデコーディングを指定する。2種類のハフマン・テーブルが存在しており、固定ハフマン・テーブル（FHT）は、例えばすべての可能なコーディングを含んでいる既定の仕様であり、動的ハフマン・テーブル（DHT）は、圧縮されるデータに対して具体的に作成されたコーディングのセットであり、すべての可能なコーディングのサブセットであることができる。DHTを使用して生成されたデータの圧縮表現は、FHTを使用して生成された同じデータの圧縮表現よりも、通常は小さい。直前に処理された圧縮されていないデータの一部（履歴と呼ばれる）が、重複列を表す圧縮データ・シンボルをエンコードおよびデコードするために維持される。履歴は、重複列の参照元である。動作中にデータが処理されるときに、履歴が更新される。

10

【0041】

示されているように、1つの例では、デフレート変換呼び出し命令は、RCF 1951, DEFLATE Compressed Data Format Specification version 1.3に記載されているデフレート圧縮データ形式を使用する。デフレート変換呼び出し命令に適用されるデフレート規格の属性は、例えば以下を含む。

20

- ・圧縮データ・セットが一連のブロックを含む。3種類のブロックが存在する。1つの種類は、3ビットのヘッダーならびにその後続く長さの情報および圧縮されていないデータを含み、ブロックの2つの種類は、3ビットのヘッダーおよびその後続く圧縮データ要素を含む。

- ・圧縮データ要素は、動的ハフマン・テーブルの圧縮表現、圧縮データ・シンボル、およびブロック終結（EOB：end-of-block）シンボルを含んでよい。

- ・圧縮データ要素は、さまざまなビット長を有する。

- ・圧縮データ要素は、ストレージ内のバイト境界間で開始または終了してよい。

- ・圧縮データ要素は、例えば右端のビット位置から左端のビット位置まで、順番にバイトに読み込まれる。

30

【0042】

圧縮データ要素が、ストレージ内のあるバイトの全部ではなく一部を占める場合、ストレージ内のそのバイト全体がアクセスされる。ストレージ・オペランドの長さは、アドレス指定可能なバイトの数を指定し、圧縮データが占めるビットよりも多くのビットを指定することができる。

【0043】

以下では、圧縮データ・ブロックに関する詳細がさらに説明される。

【0044】

デフレート変換呼び出し（DEFLATE）命令の一実施形態が、図3A～3Lを参照して説明される。この命令は、1つの例では、汎用プロセッサ（例えば、プロセッサ102または204）を使用して実行される。本明細書における説明では、フィールドの特定の位置、特定のフィールド、またはフィールドの特定のサイズ、あるいはその組み合わせが（例えば、特定のバイトまたはビットあるいはその両方で）示される。しかし、他の位置、フィールド、またはサイズ、あるいはその組み合わせが提供されてよい。さらに、特定の値（例えば、1または0）へのビットの設定が指定されるが、これは単なる例である。ビットは、他の例では、反対の値または別の値などの異なる値に設定されてよい。多くの変形が可能である。

40

【0045】

1つの実施形態では、プログラム（例えば、オペレーティング・システムまたはユーザ・プログラム）が、デフレート変換呼び出し命令を複数回実行して、単一のデータ・スト

50

リームを圧縮または復元してよい。例えば、アプリケーションが大きい（例えば、1 M バイトより大きい）データ・ストリームを圧縮または復元する場合、動作は、データ・ストリームのバッファされた部分を圧縮または復元するための複数の呼び出しを含んでよい。本発明の1つの態様に従って、プログラムは、デフレート変換呼び出し命令の複数の実行にわたる動作中に処理された圧縮されていないデータの履歴を蓄積するために使用されるバッファ（例えば、32 K バイトのバッファ）を宣言する。このバッファは、円形履歴バッファと呼ばれ、本明細書において説明されているように、デフレート変換呼び出し命令を使用して定義される。

【0046】

図3Aを参照すると、1つの例では、デフレート変換呼び出し（D F L T C C）命令300の形式は、拡張されたオペレーション・コード（オペコード）フィールドおよび追加のレジスタ・フィールドを使用してレジスタおよびレジスタの動作を示すR R F形式である。一例として、この命令は、デフレート変換呼び出しの動作を示すオペレーション・コードを含んでいるオペレーション・コード・フィールド302（例えば、ビット0～15）と、汎用レジスタの第1の対を指定する第1のレジスタ・フィールド（R1）304（例えば、ビット24～27）と、汎用レジスタの第2の対を指定する第2のレジスタ・フィールド（R2）306（例えば、ビット28～31）と、第3の汎用レジスタを指定する第3のレジスタ・フィールド（R3）308（例えば、ビット16～19）とを含む。R1フィールド304によって指定されたレジスタの内容は、（ストレージ内の）第1のオペランドの位置を指定し、R2フィールド306によって指定されたレジスタの内容は、（ストレージ内の）第2のオペランドの位置を指定し、R3フィールド308によって指定されたレジスタの内容は、（ストレージ内の）第3のオペランドの位置を指定する。R1+1の内容は第1のオペランドの長さを指定し、R2+1の内容は第2のオペランドの長さを指定する。1つの例では、命令のビット20～23は予備であり、0を含むべきである。そうでない場合、プログラムは将来、互換性を維持して動作しなくなることがある。本明細書において使用されるとき、プログラムは、デフレート変換呼び出し命令の発行元である。プログラムは、ユーザ・プログラム、オペレーティング・システム、または別の種類のプログラムであってよい。

【0047】

1つの実施形態では、命令の実行は1つまたは複数の暗黙の汎用レジスタ（すなわち、命令によって明示的に指定されないレジスタ）の使用を含む。例えば、本明細書において説明されているように、デフレート変換呼び出し命令の実行では汎用レジスタ0および1が使用される。汎用レジスタ0は、1つの例では、実行される機能（および、以下で説明されるように、履歴バッファ・タイプ）を指定するために使用され、汎用レジスタ1は、命令によって使用されるパラメータ・ブロックの位置を提供するために使用される。

【0048】

一例として、図3Bを参照すると、汎用レジスタ0（309）が履歴バッファ・タイプ・フィールド310および機能コード・フィールド312を含んでいる。1つの特定の例では、汎用レジスタ0のビット位置56が履歴バッファ・タイプを含み、汎用レジスタ0のビット位置57～63が機能コードを含むが、他の実施形態では、履歴バッファ・タイプまたは機能コードあるいはその両方を含むために他のビットが使用されてよい。汎用レジスタ0のビット57～63が、割り当てられていないか、またはインストールされていない機能コードを指定している場合、1つの例では、指定例外が認識される。

【0049】

デフレート変換呼び出し命令の例示的な割り当てられた機能コードが、図3Cに示されており、例えば、D F L T C C - Q A F（query available functions：使用可能な機能の照会）機能を示す機能コード0（313）、D F L T C C - G D H T（Generate Dynamic-Huffman Table：動的ハフマン・テーブルの生成）機能を示す機能コード1（315）、D F L T C C - C M P R（圧縮）機能を示す機能コード2（317）、およびD F L T C C - X P N D（展開）機能を示す機能コード4（319）を含んでいる。各コード

はパラメータ・ブロックを使用し、パラメータ・ブロックのサイズは、1つの例では、機能によって決まる。例えば、DFLTCC-QAF機能の場合、パラメータ・ブロックは32バイトであり、DFLTCC-GDHT機能の場合、パラメータ・ブロックは384バイトであり、DFLTCC-CMPR機能およびDFLTCC-XPND機能の場合、パラメータ・ブロックは1536バイトである。他の機能コードは、この例では割り当てられない。例示的な機能および機能コードが説明されているが、他の機能または機能コードあるいはその両方が使用されてよい。

【0050】

指定された機能がDFLTCC-CMPRまたはDFLTCC-XPNDである場合、汎用レジスタ0のビット56が、動作中に使用される履歴バッファ・タイプ（HBT: history buffer type）を指定する。HBTが0である場合、履歴バッファは直列履歴バッファと呼ばれる。直列履歴バッファを使用しているときに、履歴は、DFLTCC-CMPRが指定された場合、例えば第2のオペランドのすぐ左に存在し、DFLTCC-XPNDが指定され場合、例えば第1のオペランドのすぐ左に存在する。HBTが1である場合、履歴バッファは円形履歴バッファと呼ばれる。円形履歴バッファを使用しているときに、履歴は、DFLTCC-CMPRまたはDFLTCC-XPNDのいずれかが指定された場合、第3のオペランドの一部または全部である。DFLTCC-QAF機能またはDFLTCC-GDHT機能が指定された場合、汎用レジスタ0のビット56が無視される。1つの例では、汎用レジスタ0のビット位置0～31は無視される。さらに、1つの例では、汎用レジスタ0のビット位置32～55は予備であり、0を含むべきである。そうでない場合、プログラムは将来、互換性を維持して動作しなくなることがある。

【0051】

デフレート変換呼び出し命令によって使用される別の暗黙のレジスタ（汎用レジスタ1）に関する詳細が、図3Dを参照してさらに説明される。汎用レジスタ1（314）の内容は、例えば、ストレージ内のパラメータ・ブロックの左端のバイトの論理アドレス316を指定する。1つの例では、パラメータ・ブロックは4Kバイト境界上で指定され、そうでない場合、指定例外が認識される。以下では、パラメータ・ブロックに関する詳細がさらに説明される。

【0052】

指定された機能（例えば、DFLTCC-QAF、DFLTCC-GDHT、DFLTCC-CMPR、DFLTCC-XPND）の場合、汎用レジスタ0、1、およびR3の内容は変更されない。さらに、1つの例では、R1フィールド304が、偶数と奇数の汎用レジスタ対を指定する。これは、偶数番号が付けられたレジスタを指定することであり、汎用レジスタ0を指定することではなく、そうでない場合、指定例外が認識される。

【0053】

図3E～3Fに示され、本明細書においてさらに詳細に説明されるように、汎用レジスタR1 318の内容は、第1のオペランド・アドレス320を示し、汎用レジスタR1+1 322の内容は、第1のオペランドの長さ324を決定するために使用される。例えば、指定された機能がDFLTCC-CMPRまたはDFLTCC-XPNDである場合は、汎用レジスタR1 318の内容は、第1のオペランドの左端のバイトの論理アドレスを指定する。指定された機能がDFLTCC-CMPRである場合、汎用レジスタR1+1の内容は、パラメータ・ブロックの新しいタスク（NT: new task）フィールドおよびサブバイト境界（SBB: sub-byte boundary）フィールドの値（以下で説明される）と共に、第1のオペランドの長さを指定する。以下の表は、汎用レジスタR1+1の内容、NTフィールド、およびSBBフィールドに応じてDFLTCC-CMPR機能の第1のオペランドの長さを示す例を提供している。

【0054】

10

20

30

40

50

【表 1】

【表 1】

<u>GR R1+1 の内容(16 進数)</u>	<u>NT</u>	<u>SBB (2 進数)</u>	<u>第 1 のオペランドの長さ</u>
00000000 00000002	0	001	15 ビット
00000000 00000001	1	---	8 ビット
00000000 00000001	0	000	8 ビット
00000000 00000001	0	011	5 ビット
00000000 00000001	0	111	1 ビット
00000000 00000000	-	---	0 ビット

10

【 0 0 5 5 】

指定された機能が D F L T C C - X P N D である場合、汎用レジスタ R 1 + 1 の内容は、第 1 のオペランドの長さを指定する。指定された機能が D F L T C C - C M P R または D F L T C C - X P N D である場合、データの圧縮または復元の結果が、第 1 のオペランドの位置に格納される。D F L T C C - Q A F 機能または D F L T C C - G D H T 機能が指定された場合、汎用レジスタ R 1 および R 1 + 1 の内容が無視される。

20

【 0 0 5 6 】

さらに、指定された機能（例えば、D F L T C C - Q A F、D F L T C C - G D H T、D F L T C C - C M P R、および D F L T C C - X P N D）の場合、1 つの例では、R 2 フィールド 3 0 6 が偶数と奇数の汎用レジスタ対を指定する。これは、偶数番号が付けられたレジスタを指定することであり、汎用レジスタ 0 を指定することではなく、そうでない場合、指定例外が認識される。

【 0 0 5 7 】

図 3 G ~ 3 H に示され、本明細書においてさらに詳細に説明されるように、汎用レジスタ R 2 3 2 6 の内容は、第 2 のオペランド・アドレス 3 2 8 を示し、汎用レジスタ R 2 + 1 3 3 0 の内容は、第 2 のオペランドの長さ 3 3 2 を決定するために使用される。例えば、指定された機能が D F L T C C - G D H T、D F L T C C - C M P R、または D F L T C C - X P N D である場合は、汎用レジスタ R 2 の内容は、第 2 のオペランドの左端のバイトの論理アドレスを指定する。指定された機能が D F L T C C - C M P R または D F L T C C - G D H T である場合、汎用レジスタ R 2 + 1 の内容は、第 2 のオペランドの長さを指定する。指定された機能が D F L T C C - X P N D である場合、汎用レジスタ R 2 + 1 の内容は、パラメータ・ブロックの N T フィールドおよび S B B フィールドの値と共に、第 2 のオペランドの長さを指定する。命令の実行の開始時に、第 2 のオペランドの長さが参照され、0 以外の値である場合、第 2 のオペランドの位置からデータがフェッチされる。命令の実行の開始時に、第 2 のオペランドの長さが参照され、0 の値であり、かつ命令の実行の開始時に、パラメータ・ブロックの継続フラグ（C F : continuation flag）フィールドが 1 である場合、第 2 のオペランドがアクセスされない。

30

40

【 0 0 5 8 】

D F L T C C - Q A F 機能が指定された場合、汎用レジスタ R 2 および R 2 + 1 の内容が無視される。D F L T C C - G D H T 機能が指定され、汎用レジスタ R 2 + 1 の内容が、0 に等しい長さを指定する場合、指定例外が認識され、第 2 のオペランドがアクセスされない。D F L T C C - C M P R 機能または D F L T C C - X P N D 機能が指定され、命令の実行の開始時に、パラメータ・ブロックの継続フラグ（C F）フィールドが 0 であり、汎用レジスタ R 2 + 1 の内容が、0 に等しい長さを指定する場合、指定例外が認識され、第 2 のオペランドがアクセスされない。

50

【0059】

図3Iに示されているように、指定された機能がDFLTCC-CMPRまたはDFLTCC-XPNDであり、履歴バッファ・タイプ(HBT)が円形(例えば、HBT310=1)である場合、汎用レジスタR3 335の内容が、円形履歴バッファのアドレス337を指定する。例えば、第3のオペランドの左端のバイトの論理アドレスが指定される。この指定は、例えば、4Kバイト境界を指定することであり、そうでない場合、指定例外が認識される。1つの例では、円形履歴バッファは、第3のオペランドの位置にある。指定された機能がDFLTCC-CMPRまたはDFLTCC-XPNDであり、かつHBTが0である場合、汎用レジスタR3の内容が無視される。DFLTCC-QAF機能またはDFLTCC-GDHT機能が指定された場合、汎用レジスタR3の内容が無視される。指定された機能(例えば、DFLTCC-QAF、DFLTCC-GDHT、DFLTCC-CMPR、およびDFLTCC-XPND)に関して、R3フィールドは汎用レジスタ0も汎用レジスタ1も指定せず、そうでない場合、1つの例では指定例外が認識される。

10

【0060】

動作の一部として、指定された機能がDFLTCC-CMPRである場合、汎用レジスタR1内のアドレスが、ビット位置0の処理を含んでいる第1のオペランドの処理されたバイト数だけインクリメントされ、汎用レジスタR1+1内の長さが、同じ数だけデクリメントされ、汎用レジスタR2内のアドレスが、第2のオペランドの処理されたバイト数だけインクリメントされ、汎用レジスタR2+1内の長さが、同じ数だけデクリメントされる。ビット位置0の処理を含んでいる第1のオペランドの処理されたバイト数は、例えば、整数除算から得られた整数化した商であり、被除数は、処理された出力ビット数およびSBBの元の値の和であり、除数は8の値である。アドレスおよび長さの形成および更新は、下で説明されているように、アドレス指定モードによって決まる。

20

【0061】

動作の一部として、指定された機能がDFLTCC-XPNDである場合、汎用レジスタR1内のアドレスが、第1のオペランドの処理されたバイト数だけインクリメントされ、汎用レジスタR1+1内の長さが、同じ数だけデクリメントされ、汎用レジスタR2内のアドレスが、ビット位置0の処理を含んでいる第2のオペランドの処理されたバイト数だけインクリメントされ、汎用レジスタR2+1内の長さが、同じ数だけデクリメントされる。ビット位置0の処理を含んでいる第2のオペランドの処理されたバイト数は、整数除算から得られた整数化した商であり、被除数は、処理された入力ビット数およびSBBの元の値の和であり、除数は8の値である。アドレスおよび長さの形成および更新は、下で説明されているように、アドレス指定モードによって決まる。

30

【0062】

1つの実施形態では、24ビット・アドレス指定モードにおいて以下が当てはまる。
・汎用レジスタ1、R1、R2、およびR3のビット位置40~63の内容は、パラメータ・ブロック、第1のオペランド、第2のオペランド、および円形履歴バッファのアドレスをそれぞれ構成し、ビット位置0~39の内容が無視される。
・更新された第1のオペランド・アドレスおよび第2のオペランド・アドレスのビット40~63は、汎用レジスタR1およびR2内の対応するビットをそれぞれ置き換える。更新されたアドレスのビット位置40からの繰り上げが無視され、汎用レジスタR1およびR2のビット位置32~39の内容が0に設定される。汎用レジスタR1およびR2のビット位置0~31の内容は変化しない。命令が部分的完了または正常完了の状態を終了し、更新されたオペランド・アドレスが命令の実行の開始時のオペランド・アドレスに等しい場合、対応する汎用レジスタのビット位置32~39が0に設定される。
・汎用レジスタR1+1およびR2+1のビット位置32~63の内容は、第1および第2のオペランド内のバイト数をそれぞれ指定する、例えば32ビットの符号なし2進整数を形成する。汎用レジスタR1+1およびR2+1のビット位置0~31の内容は無視される。

40

50

・更新された第1のオペランドおよび第2のオペランドの長さのビット32～63は、汎用レジスタR1+1およびR2+1内の対応するビットをそれぞれ置き換える。汎用レジスタR1+1およびR2+1のビット位置0～31の内容は変化しない。

【0063】

1つの実施形態では、31ビット・アドレス指定モードにおいて以下が当てはまる。

・汎用レジスタ1、R1、R2、およびR3のビット位置33～63の内容は、パラメータ・ブロック、第1のオペランド、第2のオペランド、および円形履歴バッファのアドレスをそれぞれ構成し、ビット位置0～32の内容が無視される。

・更新された第1のオペランド・アドレスおよび第2のオペランド・アドレスのビット33～63は、汎用レジスタR1およびR2内の対応するビットをそれぞれ置き換える。更新されたアドレスのビット位置33からの繰り上げが無視され、汎用レジスタR1およびR2のビット位置32の内容が0に設定される。汎用レジスタR1およびR2のビット位置0～31の内容は変化しない。命令が部分的完了または正常完了の状態で終了し、更新されたオペランド・アドレスが命令の実行の開始時のオペランド・アドレスに等しい場合、対応する汎用レジスタのビット位置32が0に設定される。

・汎用レジスタR1+1およびR2+1のビット位置32～63の内容は、第1および第2のオペランド内のバイト数をそれぞれ指定する、32ビットの符号なし2進整数を形成する。汎用レジスタR1+1およびR2+1のビット位置0～31の内容は無視される。

・更新された第1のオペランドおよび第2のオペランドの長さのビット32～63は、汎用レジスタR1+1およびR2+1内の対応するビットをそれぞれ置き換える。汎用レジスタR1+1およびR2+1のビット位置0～31の内容は変化しない。

【0064】

1つの実施形態では、64ビット・アドレス指定モードにおいて以下が当てはまる。

・汎用レジスタ1、R1、R2、およびR3のビット位置0～63の内容は、パラメータ・ブロック、第1のオペランド、第2のオペランド、および円形履歴バッファのアドレスをそれぞれ構成する。

・更新された第1のオペランド・アドレスおよび第2のオペランド・アドレスのビット0～63は、汎用レジスタR1およびR2内の対応するビットをそれぞれ置き換える。更新されたアドレスのビット位置0からの繰り上げは無視される。

・汎用レジスタR1+1およびR2+1のビット位置0～63の内容は、第1および第2のオペランド内のバイト数をそれぞれ指定する、64ビットの符号なし2進整数を形成する。

・更新された第1のオペランドおよび第2のオペランドの長さのビット0～63は、汎用レジスタR1+1およびR2+1内の対応するビットをそれぞれ置き換える。

【0065】

アクセス・レジスタ・モードでは、アクセス・レジスタ1、R1、R2、およびR3が、パラメータ・ブロック、第1のオペランド、第2のオペランド、および円形履歴バッファを含んでいるアドレス空間をそれぞれ指定する。アクセス・レジスタ・モードで、直列履歴バッファを含むDFLTCC-CMPRが指定された場合、アクセス・レジスタR2が、直列履歴を含んでいるアドレス空間を指定する。アクセス・レジスタ・モードで、直列履歴バッファを含むDFLTCC-XPNDが指定された場合、アクセス・レジスタR1が、直列履歴を含んでいるアドレス空間を指定する。

【0066】

以下では、さまざまな機能に関する詳細がさらに説明される。

【0067】

機能コード0：DFLTCC-QAF（使用可能な機能の照会）

【0068】

DFLTCC-QAF（使用可能な機能の照会）機能は、インストール済み機能およびインストール済みパラメータ・ブロック形式の使用可能性を示すためのメカニズムを提供する。DFLTCC-QAF機能のパラメータ・ブロックの1つの例示的な形式が、図3

10

20

30

40

50

Jを参照して説明される。1つの例では、D F L T C C - Q A F機能（例えば、機能コード0）のパラメータ・ブロック3 4 0は、インストール済み機能ベクトル3 4 2およびインストール済みパラメータ・ブロック形式ベクトル3 4 6を含む。1つの特定の例では、これらのベクトルは、パラメータ・ブロックのバイト0～1 5およびバイト2 4～2 5にそれぞれ格納される。これらのベクトルの各々が、以下でさらに説明される。

【0 0 6 9】

一例として、インストール済み機能ベクトル3 4 2のビット0～1 2 7は、デフレート変換呼び出し命令の機能コード0～1 2 7にそれぞれ対応する。あるビットが例えば1である場合、対応する機能がインストールされており、そうでない場合、その機能はインストールされていない。

【0 0 7 0】

さらに、1つの例では、インストール済みパラメータ・ブロック形式ベクトル3 4 6のビット0～1 5が、D F L T C C - G D H T機能、D F L T C C - C M P R機能および、D F L T C C - X P N D機能のパラメータ・ブロック形式0～1 5にそれぞれ対応する。あるビットが例えば1である場合、対応するパラメータ・ブロック形式がインストールされており、そうでない場合、そのパラメータ・ブロック形式はインストールされていない。1つの例では、パラメータ・ブロックの予備のバイト1 6～2 3および2 6～3 1には0が格納される。

【0 0 7 1】

パラメータ・ブロック3 4 0に関して特定のフィールドが説明されているが、他の実施形態では、追加のフィールド、より少ないフィールド、またはその他のフィールド、あるいはその組み合わせが含まれてよい。

【0 0 7 2】

1つの実施形態では、D F L T C C - Q A F機能によって汎用レジスタR 1、R 2、R 3、R 1 + 1、およびR 2 + 1の内容が無視される。

【0 0 7 3】

パラメータ・ブロックに関して、適用可能な場合、P E R（program event recording：プログラム・イベント記録）ストレージ変更イベントが認識される。パラメータ・ブロックに関して、適用可能な場合、P E Rゼロ・アドレス検出イベントが認識される。

【0 0 7 4】

1つの例では、D F L T C C - Q A F機能の実行が完了したときに、条件コード0が設定され、条件コード1、2、および3は、照会機能には適用できない。

【0 0 7 5】

機能コード1：D F L T C C - G D H T（動的ハフマン・テーブルの生成）

【0 0 7 6】

D F L T C C - G D H T機能が指定された場合、デフレート規格によって規定されているように、第2のオペランドは、例えば、動的ハフマン・テーブル（D H T）の圧縮表現を生成するためのソースとして使用される。

【0 0 7 7】

1つの例では、D F L T C C - G D H T機能はパラメータ・ブロックを使用し、その例が図3 Kを参照して説明される。本明細書に記載された例示的なパラメータ・ブロックでは、特定のフィールドのパラメータ・ブロック内の特定の位置、およびフィールドの特定のサイズが（例えば、特定のバイトまたはビットあるいはその両方で）示される。しかし、フィールドのうちの1つまたは複数の他の位置またはサイズあるいはその両方が提供されてよい。さらに、特定の値（例えば、1または0）へのビットの設定が指定されるが、これは単なる例である。ビットは、他の例では、反対の値または別の値などの異なる値に設定されてよい。多くの変形が可能である。

【0 0 7 8】

さらに、1つの例では、パラメータ・ブロックが、1つまたは複数の保存フィールド（preserved fields）および1つまたは複数の予備フィールドを含む。保存フィールドは、

10

20

30

40

50

D F L T C C - G D H T 機能によって変更されない。保存フィールドは、プログラムが単一のストレージ位置を初期化することを可能にする予備フィールドとは区別され、そのストレージ位置を D F L T C C - G D H T 機能のパラメータ・ブロックに使用し、その後、同じストレージ位置を D F L T C C - C M P R 機能のパラメータ・ブロックに使用する。予備フィールドは 0 を含むべきである。そうでない場合、プログラムは将来、互換性を維持して動作しなくなることがある。動作が終了するときに、予備フィールドは、0 として格納されてよく、または変化しなくてよい。

【 0 0 7 9 】

さらに、フィールドの一部が他の機能（例えば、D F L T C C - C M P R または D F L T C C - X P N D）によって使用され、したがって、それらのフィールドの説明と共に、それらの機能に関連する態様を説明することもできる。

【 0 0 8 0 】

1 つの例では、D F L T C C - G D H T 機能の場合、パラメータ・ブロック 3 6 0 は以下のフィールドを含む。

【 0 0 8 1 】

パラメータ・ブロック・バージョン番号（P B V N : Parameter Block Version Number）3 6 2 : パラメータ・ブロックのバイト 0 ~ 1 は、パラメータ・ブロックのバージョンおよびサイズを指定する。P B V N のビット 0 ~ 1 1 は予備であり、0 を含むべきである。そうでない場合、プログラムは将来、互換性を維持して動作しなくなることがある。P B V N のビット 1 2 ~ 1 5 は、パラメータ・ブロックの形式を指定する符号なし 2 進整数を含む。D F L T C C - Q A F 機能は、使用可能なパラメータ・ブロック形式を示すためのメカニズムを提供する。指定されたパラメータ・ブロックの形式がモデルによってサポートされていない場合、一般オペランド・データ例外が認識される。P B V N は、プログラムによって指定され、命令の実行中に変更されない。

【 0 0 8 2 】

モデル・バージョン番号（M V N : Model Version Number）3 6 3 : パラメータ・ブロックのバイト 2 は、命令を実行したモデルを識別する符号なし 2 進整数である。プログラムが M V N を初期化する必要はない。M V N は、命令の実行中に更新される。M V N に格納される値は、モデルに依存する。

【 0 0 8 3 】

動的ハフマン・テーブル（D H T）生成制御（D H T G C : Dynamic-Huffman Table (DHT) Generation Control）3 6 4 : パラメータ・ブロックのバイト 1 7 のビット 2 が、動的ハフマン・テーブル（D H T）の生成に適用される。D H T は、リテラル・バイトを表すシンボルのハフマン・コード、重複列の長さ、ブロック終結（E O B）シンボル、および重複列のポインタ距離を指定する。特定のシンボルのハフマン・コードの値は、データの圧縮されていない形態での、シンボルが表す実体の発生数に応じて変わる。あるシンボルの発生数が 0 である場合、このシンボルのハフマン・コードは D H T に存在しない。D H T G C は、1 つの例では、0 に等しい数が次のように扱われることを指定する。

D H T G C 意味

0 0 に等しいリテラル・バイト、重複列の長さ、およびポインタ距離の数を、1 に等しいとして扱う（汎用 D H T (universal DHT) を生成する）。

1 0 に等しい重複列の長さおよびポインタ距離の数を、1 に等しいとして扱う。

【 0 0 8 4 】

リテラル・バイト、E O B シンボル、重複列の長さ、重複列のポインタ距離のすべての可能な値に対してハフマン・コードを指定する D H T は、汎用 D H T と呼ばれる。データの圧縮されていない形態で発生しないリテラル・バイト、重複列の長さ、または重複列のポインタ距離の値に対してハフマン・コードを指定しない D H T は、非汎用 D H T と呼ばれる。

【 0 0 8 5 】

デフレート規格によって規定されているように、D H T G C のすべての値に関して得ら

10

20

30

40

50

れたDHTは、すべての可能な重複列の長さおよびポインタ距離に対してハフマン・コードを指定する。したがって、下でさらに説明されているように、DHTの得られた圧縮された形態のHLIT（ハフマン・リテラル（Huffman literal））部分要素およびHDIST（ハフマン距離）部分要素は、例えば29の値をそれぞれ含む。

【0086】

DFLTCC-GDHT機能が指定された場合、DHTGCが動作への入力である。DFLTCC-CMPR機能またはDFLTCC-XPND機能が指定された場合、DHTGCは動作に適用されない。1つの実施形態では、DHTGCは、命令の実行中に変更されない。

【0087】

動作終了補助コード（OESC：Operation Ending Supplemental Code）365：パラメータ・ブロックのバイト19は、プログラムに報告されている条件に関する追加情報を提供する符号なし2進整数である。このフィールドは複数の機能によって使用されるため、条件の一部は、他の機能によって使用されるパラメータ・ブロック（例えば、DFLTCC-CMPR機能およびDFLTCC-XPND機能によって使用される図3Lのパラメータ・ブロック）のフィールドを参照する。報告されている条件が一般オペランド・データ例外である場合、パラメータ・ブロックのOESCフィールドが更新されるが、動作は抑制されたと見なされ、その場合、1つの例では、OESCフィールドが次のように定義される。

OESC（16進数） 意味

00 追加情報が提供されない。

01 パラメータ・ブロック・バージョン番号362によって指定されたパラメータ・ブロックの形式が、モデルによってサポートされない。

02 DFLTCC-CMPR機能またはDFLTCC-XPND機能が指定され、履歴長フィールド385（図3L）が例えば32，768より大きく、新しいタスク・フィールド374（図3L）が0である。

11 BTYPE（ブロック・タイプ）が2進数11に等しい圧縮データ・ブロックが発生する。

12 BTYPEが2進数00に等しい圧縮データ・ブロック、およびLEN（長さ）の1の補数に等しくないNLENが発生する。

21 CDHTLフィールド366（図3L）が適用され、例えば、42より小さいか、または2283より大きい。

22 動作中に使用される圧縮されたDHTのHLIT部分要素が、例えば29より大きい（無効なDHT）。

23 動作中に使用される圧縮されたDHTのHDIST部分要素が、例えば29より大きい（無効なDHT）。

24 動作中に使用される圧縮されたDHTが、圧縮されたDHTに対して定義された可能な（例えば、19個の）コード長のビット長を指定するコードのシーケンス内にある、機能するハフマン・ツリーを指定するためにハフマン・アルゴリズムによって必要とされる長さより小さいコードを指定する（無効なDHT）。

26 動作中に使用される圧縮されたDHTが、リテラル・バイト、EOBシンボル、および重複列の長さで構成されている要素のセットの第1のコード長としてコード長16（前のコード長のコピー）を指定する（無効なDHT）。

27 動作中に使用される圧縮されたDHTが、リテラル・バイトのコード長を指定するコードのシーケンス内にあるコードを指定し、このコードが、圧縮されたDHT内で前に指定された、参照されるコード長のセットを表すために決定されたコードのいずれにも一致しない（無効なDHT）。

28 動作中に使用される圧縮されたDHTが、コード長0（CLO）をEOBシンボルに割り当てるコードを指定する。この場合、対応するDHTが、EOBシンボルを表すためのハフマン・コードを指定しない（無効なDHT）。

10

20

30

40

50

2 9 動作中に使用される圧縮された D H T が、重複列の長さおよびポインタ距離のコード長を指定するコードのシーケンス内にあるコードを指定し、このコードが、圧縮された D H T 内で前に指定された、参照されるコード長のセットを表すために決定されたコードのいずれにも一致しない（無効な D H T）。

2 A 動作中に使用される圧縮された D H T が、H L I T フィールド、H D I S T フィールド、および例えば 2 5 8 の値の和によって指定された、D H T 内のハフマン・コードの数より大きいコード長の数を指定する。これは、例えば、コード長 1 6、1 7、および 1 8 の不適切な使用を伴っている可能性がある（無効な D H T）。

2 B 動作中に使用される圧縮された D H T が、リテラル・バイト、E O B シンボル、および重複列の長さのセットに対して、機能するハフマン・ツリーを指定するためにハフマン・アルゴリズムによって必要とされる長さより小さいコード長を指定する（無効な D H T）。

10

2 D 動作中に使用される圧縮された D H T が、重複列のポインタ距離のセットに対して、機能するハフマン・ツリーを指定するためにハフマン・アルゴリズムによって必要とされる長さより小さいコード長を指定する（無効な D H T）。

2 F C D H T L フィールド 3 6 6（図 3 L）が適用され、動作中に使用される C D H T フィールド 3 6 7（図 3 L）内の圧縮された D H T の長さに等しくない。

3 1 動作中に使用される圧縮された D H T が、動作中に処理されるリテラル・バイトまたは重複列の長さに対応するハフマン・コードを指定しないか（不完全な非汎用 D H T）、または D F L T C C - X P N D 機能が指定され、B T Y P E が 2 進数 0 1 に等しい圧縮データ・ブロック内で発生する圧縮データ・シンボルが、重複列の長さに対して無効なコード（2 進数 1 1 0 0 0 1 1 0 または 1 1 0 0 0 1 1 1）を指定する。

20

3 2 動作中に使用される圧縮された D H T が、動作中に処理される重複列のポインタ距離に対応するハフマン・コードを指定しないか（不完全な非汎用 D H T）、または D F L T C C - X P N D 機能が指定され、B T Y P E が 2 進数 0 1 に等しい圧縮データ・ブロック内で発生する圧縮データ・シンボルが、重複列のポインタ距離に対して無効なコード（2 進数 1 1 1 1 0 または 1 1 1 1 1）を指定する。

4 0 重複列のポインタであり、シンボルを処理する時点で使用可能な履歴の長さより大きい距離を指定する圧縮データ・シンボルが発生する。

【 0 0 8 8 】

30

一般オペランド・データ例外を報告せずに動作が終了した場合、0 が O E S C フィールドに格納される。

【 0 0 8 9 】

0 以外の補助コードのサポートは、モデルに依存する。複数の条件が存在する場合、コードがもしあれば、どのコードが O E S C フィールドに報告されるかは、モデルに依存する。

【 0 0 9 0 】

圧縮された動的ハフマン・テーブルの長さ（C D H T L : Compressed Dynamic-Huffman Table Length）3 6 6 : パラメータ・ブロックのバイト 5 6 のビット 4 からバイト 5 7 のビット 7 までの 1 2 ビットが、パラメータ・ブロックの C D H T フィールド（例えば、C D H T 3 6 7）内の圧縮形式の D H T の長さをビット数として指定する符号なし 2 進整数を含む。

40

【 0 0 9 1 】

D F L T C C - G D H T 機能が指定された場合、C D H T L が動作からの出力である。

【 0 0 9 2 】

D F L T C C - C M P R 機能が指定され、ハフマン・テーブル・タイプ（例えば、図 3 L の H T T 3 7 6）が 1 である場合、C D H T L が動作への入力である。C D H T L が C D H T の適切な長さを指定しない場合、一般オペランド・データ例外が認識される。D F L T C C - C M P R 機能が指定された場合、C D H T L は変更されない。

【 0 0 9 3 】

50

D F L T C C - X P N D 機能が指定され、B T Y P E が 2 進数 1 0 であるブロックの一部のみをデコードした後に動作が終了した場合、ブロック内の D H T の圧縮表現の長さがこのフィールドに格納される。D F L T C C - X P N D 機能が指定され、ブロック境界で、または B T Y P E が 2 進数 0 0 または 0 1 であるブロックの一部のみをデコードした後に、動作が終了した場合、0 がこのフィールドに格納される。B T Y P E が 2 進数 1 0 であるブロック内で復元動作が再開された場合（すなわち、下で説明されるように、C F（図 3 L の継続フラグ 3 7 3）が 1 に等しく、I F S（incomplete function status：不完全な機能の状態 3 8 3）が 1 6 進数 C または D に等しい場合）、このフィールドが動作への入力である。

【 0 0 9 4 】

圧縮された動的ハフマン・テーブル（C D H T：Compressed Dynamic-Huffman Table）3 6 7：パラメータ・ブロックのバイト 6 4～3 5 1 は、動的ハフマン・テーブル（D H T）の圧縮形式を含む。

【 0 0 9 5 】

D H T は、要素の 2 つのセットを表すためのハフマン・コード（ビット・シーケンス）を指定する。1 つのセットの要素は、リテラル・バイト、E O B シンボル、および重複列の長さを含む。他のセットの要素は、重複列のポインタ距離を含む。D H T の圧縮表現は、コード長のセットを定義し、各セットの要素ごとにコード長（C L：code length）を指定する。動作中に参照されることが期待される要素のハフマン・コードは、その要素に対して指定された C L、および同じ指定された C L を持つ同じセット内の要素の数から導出される。具体的には、D H T の圧縮表現は、一例として以下を含む。

- ・リテラル・バイト、E O B シンボル、および重複列の長さを表すハフマン・コードの数を指定するための H L I T フィールド。

- ・重複列のポインタ距離を表すハフマン・コードの数を指定するための H D I S T フィールド。

- ・コード長を表すハフマン・コードの数を指定するための H C L E N（ハフマン・コード長）フィールド。

- ・圧縮された D H T に対して定義された例えば 1 9 個のコード長の各々に対してビット長を指定するコードのシーケンス。

- ・リテラル・バイト、E O B シンボル、および重複列の長さで構成されるセットの要素の各々に対してコード長を指定するコードのシーケンス。

- ・重複列のポインタ距離で構成されるセットの要素の各々に対してコード長を指定するコードのシーケンス。

【 0 0 9 6 】

D H T の圧縮表現の詳細が、ブロック・タイプが 2 進数 1 0 である圧縮データ・ブロックの説明を参照して下でさらに説明される。

【 0 0 9 7 】

1 つの例では、D H T の圧縮表現は、C D H T フィールド内で左詰めにされる。すなわち、バイト 6 4 の右端のビットが、D H T の圧縮表現の H L I T 部分要素の最下位ビットを含む。

【 0 0 9 8 】

D F L T C C - G D H T 機能が指定された場合、D H T の圧縮表現が動作からの出力である。

【 0 0 9 9 】

D F L T C C - C M P R 機能が指定され、下で説明される H T T が 1 である場合、D H T の圧縮表現が動作への入力である。C D H T フィールドは、D F L T C C - C M P R 機能によって変更されない。

【 0 1 0 0 】

D F L T C C - X P N D 機能が指定され、B T Y P E が 2 進数 1 0 であるブロックの一部のみをデコードした後に動作が終了した場合、ブロック内の D H T の圧縮表現がこのフ

10

20

30

40

50

フィールドに格納される。D F L T C C - X P N D 機能が指定され、ブロック境界で、または B T Y P E が 2 進数 0 0 または 0 1 であるブロックの一部のみをデコードした後に、動作が終了した場合、0 がこのフィールドに格納される。B T Y P E が 2 進数 1 0 であるブロック内で復元動作が再開された場合（すなわち、C F が 1 に等しく、I F S が 1 6 進数 C または D に等しい場合）、このフィールドが動作への入力である。

【0101】

C D H T が変更される場合、D H T の圧縮表現を表すために使用されないフィールドのビットが 0 として格納される。

【0102】

パラメータ・ブロック 3 6 0 に関してさまざまなフィールドが上で説明されたが、他の実施形態では、追加のフィールド、より少ないフィールド、またはその他のフィールド、あるいはその組み合わせが含まれてよい。

10

【0103】

D H T 生成の態様は、プログラムによって、パラメータ・ブロックの動的ハフマン・テーブル生成制御 (D H T G C) フィールド 3 6 4 を使用して、マシンに対して指定される。ソースが圧縮されていないデータを含み、動作の完了後に、生成された結果が、同じソースを圧縮するための D F L T C C - C M P R 機能と共に指定されるということが、意図される。

【0104】

1 つの実施形態では、現在の動作の処理中に、参照するための前の動作からの履歴が存在しない。

20

【0105】

汎用レジスタ R 2 + 1 の内容が、例えば 3 2 K バイトよりも大きい長さを指定する場合、1 つの例では以下が当てはまる。

- ・第 2 のオペランドの最初の 3 2 K バイトのみが、D H T を生成するために使用される。
- ・第 2 のオペランドの最初の 3 2 K バイトを超える位置に対して、アクセス例外が認識されない。

【0106】

汎用レジスタ R 2 + 1 の内容が、0 に等しい長さを指定する場合、指定例外が認識され、第 2 のオペランドがアクセスされない。

30

【0107】

得られた圧縮された D H T は、ブロック終結 (E O B) シンボルを表すハフマン・コードを含む。

【0108】

生成された圧縮形式の D H T が、パラメータ・ブロックの圧縮された動的ハフマン・テーブル (C D H T) フィールド 3 6 7 に格納される。生成された圧縮形式の D H T の長さが、パラメータ・ブロックの C D H T L フィールド 3 6 6 に格納される。

【0109】

この動作は、モデル識別情報をパラメータ・ブロックのモデル・バージョン番号フィールド 3 6 3 に格納することを含む。

40

【0110】

一般オペランド・データ例外を認識せずに動作が終了した場合、パラメータ・ブロックの動作終了補助コード (O E S C) フィールド 3 6 5 に 0 が格納される。

【0111】

D F L T C C - G D H T 機能の実行が完了したときに、条件コード 0 が設定され、条件コード 1、2、および 3 は、D F L T C C - G D H T 機能に適用できない。

【0112】

この動作によって、汎用レジスタ R 2 および R 2 + 1 は変更されない。

【0113】

D F L T C C - G D H T 機能が指定された場合、汎用レジスタ R 1、R 1 + 1、および

50

R 3 の内容が無視される。

【0114】

第2のオペランドの位置およびパラメータ・ブロックに関して、適用可能な場合、P E R ゼロ・アドレス検出イベントが認識される。

【0115】

機能コード2：D F L T C C - C M P R（圧縮）

【0116】

D F L T C C - C M P R 機能が指定された場合、圧縮動作が実行される。この動作は、第2のオペランドの位置のデータを、第1のオペランドの位置に格納される圧縮データ・シンボルにエンコードすることを含む。

【0117】

1つの例では、D F L T C C - C M P R 機能はパラメータ・ブロックを使用し、その例が図3 Lを参照して説明される。フィールドの一部は、パラメータ・ブロック360に関して上で説明されているため、以下では同じ参照番号で示され、さらに詳細には説明されない。

【0118】

1つの例では、パラメータ・ブロック370は以下を含む。

【0119】

パラメータ・ブロック・バージョン番号（P B V N）362。

【0120】

モデル・バージョン番号（M V N）363。

【0121】

継続フラグ（C F）373：パラメータ・ブロックのビット63は、1である場合、動作が部分的に完了していることを示し、継続状態バッファ（例えば、継続状態バッファ・フィールド392）の内容が、動作を再開するために使用されてよい。プログラムは、継続フラグ（C F）を0に初期化し、動作を再開する目的で命令が再実行される場合にC Fを変更しないようにするべきであり、そうでない場合、結果が予測不可能になる。

【0122】

新しいタスク（N T）374：パラメータ・ブロックのバイト16のビット0は、1である場合、動作が圧縮データ・セットの先頭に適用されることを示す。したがって、前の動作からの履歴およびチェック値が現在の動作に適用されない。動作の開始時にN Tが1であり、部分的完了の後に動作が終了した場合、0がN Tフィールドに格納される。N Tが0である場合、前の動作からの履歴およびチェック値が現在の動作に適用される。

【0123】

チェック値タイプ（C V T：Check Value Type）375：パラメータ・ブロックのバイト16のビット2は、パラメータ・ブロックのチェック値フィールド（例えば、フィールド387）に含まれているチェック値のタイプを指定する。C V Tが0である場合、チェック値タイプは、例えば32ビットの巡回冗長チェック（32-bit cyclic-redundancy-check：C R C - 32）である。C V Tが1である場合、チェック値タイプは、例えば32ビットのアドラー・チェックサム（A d l e r - 32）である。C V Tビットは、命令の実行中に変更されない。

【0124】

ハフマン・テーブル・タイプ（H T T：Huffman Table Type）376：パラメータ・ブロックのバイト16のビット4は、0である場合、デフレート規格によって規定されているように、固定ハフマン・コードを含んでいるテーブル（F H T）を指定し、圧縮動作中に使用される。H T Tが1である場合、パラメータ・ブロックのC D H Tフィールドにおいて指定された、動的ハフマン・コードを含んでいるテーブル（D H T）が、圧縮動作中に使用される。H T Tは、復元動作には適用されない。H T Tビットは、命令の実行中に変更されない。

【0125】

10

20

30

40

50

ブロック継続フラグ（BCF：Block Continuation Flag）377：DFLTCC-CMPR機能が指定された場合、パラメータ・ブロックのバイト16のビット5が適用される。0である場合、3ビットのブロック・ヘッダー、および適用可能な場合、パラメータ・ブロックのCDHTフィールド（例えば、フィールド367）において指定された動的ハフマン・テーブルの圧縮形式が、圧縮データ要素を格納する前に、第1のオペランドの位置に格納される。1である場合、ブロック・ヘッダーも圧縮形式のDHTも、第1のオペランドの位置に格納されない。NTが1である場合、BCFが0に等しいとして扱われる。BCFビットは、命令の実行中に変更されない。

【0126】

ブロック終了制御（BCC：Block Closing Control）378：DFLTCC-CMPR機能が指定された場合、パラメータ・ブロックのバイト16のビット6が適用される。1である場合、すべての圧縮データ・シンボルを格納した後に、ブロック終結（EOB）シンボルが第1のオペランドの位置に格納される。HTTがFHTを使用することを指定する場合、一例として、2進数0000000のハフマン・コード（リテラル・バイト、EOBシンボル、および重複列の長さのコードを指定するテーブル内の256の中間整数表現に対応する）がEOBシンボルに使用される。HTTがDHTを使用することを指定する場合、EOBシンボルのハフマン・コードがDHT内で指定される。BCCビットが0である場合、EOBシンボルが第1のオペランドの位置に格納されない。BCCビットは、命令の実行中に変更されない。

【0127】

ブロック・ヘッダー終端（BHF：Block Header Final）379：DFLTCC-CMPR機能が指定され、BCF377が0であるか、またはNT374が1である場合、パラメータ・ブロックのバイト16のビット7が適用され、そうでない場合、BHFが適用されない。適用可能であり、1である場合、ブロック・ヘッダーを第1のオペランドの位置に格納する前に、ブロック・ヘッダーの最初のビット（BFINAL）が1に設定される。適用可能であり、0である場合、ブロック・ヘッダーを第1のオペランドの位置に格納する前に、ブロック・ヘッダーの最初のビット（BFINAL）が0に設定される。BHFビットは、命令の実行中に変更されない。

【0128】

DHT生成制御（DHTGC）364：DFLTCC-CMPR機能が指定された場合、DHTGCは動作に適用されない。DHTGCは、命令の実行中に変更されない。

【0129】

サブバイト境界（SBB）381：パラメータ・ブロックのバイト18のビット5～7は、圧縮データ・ストリームのバイト内の処理済みのビットと未処理のビットの間の境界を指定する符号なし2進整数を含む。参照されるストリームのバイトは、動作が終了するとき、参照される最後のバイト（つまり、右端のバイト）であり、動作が開始または再開するとき、参照される最初のバイト（つまり、左端のバイト）である。DFLTCC-CMPR機能が指定された場合、SBBは、第1のオペランド・アドレスによって指定されたバイトに適用される。DFLTCC-XPND機能が指定された場合、SBBは、第2のオペランド・アドレスによって指定されたバイトに適用される。SBBは、処理された右端のビットの数を指定する。SBBは、動作への入力および動作の出力である。

【0130】

SBBが2進数011の値を含む場合の圧縮データ・ストリームの一例が、図4に示されている。動作の終了後の処理されたデータが400に示されており、動作の開始前の処理されるデータが402に示されている。

【0131】

さらに、図5A～5Cは、DFLTCC-CMPR機能へのSBBの適用方法を示す例を提供する。例えば、DFLTCC-CMPR機能を実行する前後のSBBの適用方法の一例が、図5Aに示されている。その他の例が、図5B～5Cに示されている。NT374が1である場合、SBB381が2進数000に等しいとして扱われる。

10

20

30

40

50

【0132】

図3Lを参照して、パラメータ・ブロック370の追加のフィールドが説明される。

【0133】

動作終了補助コード（OESC）365。

【0134】

不完全な機能の状態（IFS）383：パラメータ・ブロックのバイト21のビット4～7は、特定の動作が終了するときの状態情報を含む。復元動作が終了するとき、IFSは、1つの例では、次のように第2のオペランドに関する情報を伝達する。

IFS（2進数） 意味

0000 BFINALが1に等しいブロックの最後の要素をデコードした後に、動作が終了した。

1000 BTYPEが2進数00に等しく、BFINALが0に等しいブロックの最後の要素以外の要素をデコードした後に、動作が終了した。

1001 BTYPEが2進数00に等しく、BFINALが1に等しいブロックの最後の要素以外の要素をデコードした後に、動作が終了した。

1010 BTYPEが2進数01に等しく、BFINALが0に等しいブロックの最後の要素以外の要素をデコードした後に、動作が終了した。

1011 BTYPEが2進数01に等しく、BFINALが1に等しいブロックの最後の要素以外の要素をデコードした後に、動作が終了した。

1100 BTYPEが2進数10に等しく、BFINALが0に等しいブロックの最後の要素以外の要素をデコードした後に、動作が終了した。

1101 BTYPEが2進数10に等しく、BFINALが1に等しいブロックの最後の要素以外の要素をデコードした後に、動作が終了した。

1110 動作がブロック境界で終了しており、BFINALが1に等しいブロックの最後の要素がデコードされておらず、その後のブロックのブロック・ヘッダーの最初の要素がまだ処理されていない。

【0135】

1つの実施形態では、復元動作が、2進数0000に等しいIFSを伴って終了し、正常完了を満たさないことがある。そのような場合、条件コード1または3が設定されて動作が終了する。

【0136】

圧縮動作が終了するとき、IFSフィールドは定義されないが、変更されてよい。

【0137】

IFSは動作への入力ではない。

【0138】

不完全な機能の長さ（IFL：Incomplete Function Length）384：パラメータ・ブロックのバイト22～23は、特定の動作が終了するときの長さの情報を含む。復元動作の場合、IFLが第2のオペランドに適用される。BTYPEが2進数00に等しいブロックの全部ではなく一部をデコードした後に、復元動作が終了した場合、IFLは、第2のオペランド内のまだ処理されていないブロックのバイト数を指定する符号なし2進整数を含む。バイト22～23は、BTYPEが2進数00に等しいブロックの、例えばリトルエンディアン（little-endian）・バイト・オーダーであるLENフィールドとは異なり、例えばビッグエンディアン（big-endian）・バイト・オーダーでIFLを含む。

【0139】

BTYPEが2進数00に等しく、BFINALが1に等しい完全なブロックをデコードした後に、復元動作が終了した場合、0がIFLフィールドに格納される。復元動作が、BTYPEが0以外であるブロックの全部ではなく一部をデコードした後に終了するか、またはブロック境界で終了した場合、IFLフィールドは定義されないが、変更されてよい。

【0140】

圧縮動作が終了するときに、I F L フィールドは定義されないが、変更されてよい。

【0141】

I F L は動作への入力ではない。

【0142】

履歴長 (H L : History Length) 3 8 5 : パラメータ・ブロックのバイト 4 4 ~ 4 5 は、動作中に参照できる履歴バッファ内の履歴のバイト数を指定する符号なし 2 進整数を含む。H L は、直列履歴バッファおよび円形履歴バッファに適用される。新しいタスク (N T) が 1 に等しい場合、履歴は動作の開始時に適用されず、動作への入力としての履歴長は、0 として扱われる。

【0143】

履歴長が例えば 3 2 , 7 6 8 より大きく、N T が 0 に等しい場合、一般オペランド・データ例外が認識される。

【0144】

圧縮動作中および復元動作中に、履歴長が変更される。元の H L と、動作中に処理された圧縮されていないデータのバイト数との和が、例えば 3 2 , 7 6 8 以下である場合、更新された H L は、元の H L と、動作中に処理された圧縮されていないデータのバイト数との和に等しく、そうでない場合、更新された H L は 3 2 , 7 6 8 の値に等しい。

【0145】

履歴オフセット (H O : History Offset) 3 8 6 : パラメータ・ブロックのバイト 4 6 のビット 1 からバイト 4 7 のビット 7 までの 1 5 ビットは、履歴バッファ・タイプが円形である場合の第 3 のオペランド内のオフセットを指定する符号なし 2 進整数を含む。R 3 の内容および履歴オフセットの和は、円形履歴バッファ内の履歴の最初のバイトの位置を指定し、このバイトは、バッファ内の圧縮されていないデータのうちの処理されてから最も長い時間が経過したバイトである。履歴バッファ・タイプが円形である場合、履歴オフセットは動作への入力であり、動作の終了時に更新される。元の H L と、動作中に処理された圧縮されていないデータのバイト数との和が、例えば 3 2 , 7 6 8 以下である場合、更新された H O は、元の H L O に等しく、そうでない場合、更新された H O は、元の H O 、元の H L 、および動作中に処理された圧縮されていないデータのバイト数の和を 3 2 , 7 6 8 で割った余りに等しい。

【0146】

履歴バッファ・タイプが直列である場合、パラメータ・ブロックの H O フィールドは定義されないが、変更されてよい。

【0147】

チェック値 3 8 7 : パラメータ・ブロックのバイト 4 8 ~ 5 1 はチェック値を含む。動作の一部として、チェック値が生成される。チェック値は、圧縮されていないデータ・オペランドに適用される。すなわち、チェック値は、D F L T C C - C M P R 機能の場合、第 2 のオペランドに適用され、D F L T C C - X P N D 機能の場合、第 1 のオペランドに適用される。C V T ビット 3 7 5 が 0 である場合、例えば 3 2 ビットの巡回冗長チェックのチェック値 (C R C - 3 2) が生成される。C V T ビットが 1 である場合、例えば 3 2 ビットのアドラー・チェックサムのチェック値 (A d l e r - 3 2) が生成される。

【0148】

チェック値を生成するための入力は、例えば、4 バイトのベース、および動作中に処理される圧縮されていないデータである。ベース入力は、圧縮データ・ブロックの完全なセットを処理するために D F L T C C 命令が実行される回数に関わらず、圧縮データ・ブロックのセットに対して単一の一貫したチェック値を計算するための手段を提供する。N T ビットが 0 である場合、チェック値フィールド内の元の値が、チェック値の生成においてベース入力に使用される。

【0149】

A d l e r - 3 2 チェック値が生成される場合、1 つの例では、以下が適用される。
・ N T ビットが 1 である場合、1 の値が 4 バイトのベース入力に使用される。

10

20

30

40

50

・ Adler-32 チェック値生成において定義された和を 65,521 で割った余りが計算される。

・ その結果が、ビッグエンディアン・バイト・オーダーでチェック値フィールドに格納される。すなわち、チェック値の最上位バイトがバイト 48 に位置し、チェック値の最下位バイトがバイト 51 に位置する。

【0150】

CRC-32 チェック値が生成される場合、1つの例では、以下が適用される。

・ NT ビットが 1 である場合、0 の値が 4 バイトのベース入力に使用される。

・ CRC-32 チェック値の生成において除数として使用される多項式は、 $x^32 + x^26 + x^23 + x^22 + x^16 + x^12 + x^11 + x^10 + x^8 + x^7 + x^5 + x^4 + x^2 + x^1 + x^0$ であり、16 進数 104C11DB7 として表される。この表現では、左端のビットが最上位ビットに対応する。

・ チェック値を生成する最初の段階では、ベース入力の 1 の補数を計算し、最後の段階では、結果を格納する前に、結果の 1 の補数を計算する。

・ その結果が、リトルエンディアン・バイト・オーダーでチェック値フィールドに格納される。すなわち、チェック値の最下位バイトがバイト 48 に位置し、チェック値の最上位バイトがバイト 51 に位置する。

【0151】

1つの例では、条件コード 0 が設定されて動作が終了する場合にのみ、チェック値はプログラムにとって意味を持ち、そうでない場合、チェック値は、単に中間結果であり、動作を再開することのみにするために意味を持つ。DFLTCC-CMPR 機能が指定され、条件コード 1、2、または 3 が設定されて動作が終了した場合、第 2 のオペランド・アドレスによって指定されたバイトの左にある一部のバイトが、得られたチェック値の計算に含まれていないことがある。DFLTCC-XPND 機能が指定され、条件コード 1、2、または 3 が設定されて動作が終了した場合、第 1 のオペランド・アドレスによって指定されたバイトの右にあるまだ格納されていない一部の結果のバイトが、得られたチェック値の計算にすでに含まれていることがある。

【0152】

ブロック終結シンボル (EOBS: End-Of-Block Symbol) 388: パラメータ・ブロックのバイト 52 のビット 0 からバイト 53 のビット 6 までの 15 ビットは、ブロック終結 (EOB) シンボルを含む。パラメータ・ブロックのブロック終結の長さ (EOBL: end-of-block length) フィールド 389 は、EOBS フィールド内の EOB シンボルの長さを指定する。EOB シンボルは、EOBS フィールド内で左詰めにされる。EOB シンボルによって占められない EOBS フィールドのビットは、0 として格納される。どのタイプの Huffman テーブルが適用されるかに関わらず、データを圧縮する場合、EOBS フィールドは動作の出力である。EOBS フィールドは、動作への入力として使用されない。

【0153】

バイト 52 のビット 0 は、EOB シンボルの最上位ビットを含む。EOB シンボルの長さが 7 ビットである場合、バイト 52 のビット 6 が EOB シンボルの最下位ビットを含む。EOB シンボルの長さが 15 ビットである場合、バイト 53 のビット 6 が EOB シンボルの最下位ビットを含む。

【0154】

FHT を使用するブロックの場合、デフレート規格によって規定されているように、EOB シンボルは 2 進数 0000000 である。DHT を使用するブロックの場合、EOB シンボルは DHT によって定義される。EOB シンボルは、ブロックを閉じる能力をプログラムに提供するために伝達される。

【0155】

DFLTCC-XPND 機能が指定された場合、EOBS フィールドは定義されないが、変更されてよい。

10

20

30

40

50

【0156】

ブロック終結の長さ（EOBL）389：パラメータ・ブロックのバイト54のビット0～3は、パラメータ・ブロックのEOBSフィールド388内のブロック終結（EOB）シンボルの長さを指定する符号なし2進整数を含む。この長さは、EOBSフィールド内でEOBシンボルが占めるビット数を指定する。どのタイプのハフマン・テーブルが適用されるかに関わらず、データを圧縮する場合、EOBLフィールドは動作の出力である。EOBLフィールドは、動作への入力として使用されない。

【0157】

DFLTCC-XPND機能が指定された場合、EOBLフィールドは定義されないが、変更されてよい。

【0158】

圧縮された動的ハフマン・テーブルの長さ（CDHTL）366。

【0159】

圧縮された動的ハフマン・テーブル（CDHT）367：DFLTCC-CMPR機能が指定され、HTTが1である場合、DHTの圧縮表現が動作への入力である。CDHTフィールドは、DFLTCC-CMPR機能によって変更されない。

【0160】

継続状態バッファ（CSB：Continuation State Buffer）392：条件が、1の値がCFフィールド373に格納されることを引き起こした場合、内部状態データがパラメータ・ブロックのバイト384～1535に格納され、そうでない場合、パラメータ・ブロックのバイト384～1535が定義されず、変更されてよい。格納される内部状態データは、モデルに依存し、動作を再開するために、その後使用されてよい。プログラムが、例えばすべて0を含むように継続状態バッファを初期化することが期待されるが、これは必須ではない。0以外の条件コードが設定されて命令が終了した後に、動作を再開する目的で命令を再実行する前に、プログラムは継続状態バッファを変更するべきではなく、そうでない場合、結果が予測不可能になる。

【0161】

パラメータ・ブロック370に関してさまざまなフィールドが上で説明されたが、他の実施形態では、追加のフィールド、より少ないフィールド、またはその他のフィールド、あるいはその組み合わせが含まれてよい。

【0162】

以下では、データの圧縮に関して圧縮動作の一例が説明される。

【0163】

第2のオペランド全体が圧縮され、第1のオペランドの位置に格納されたときに、DFLTCC-CMPR機能の正常完了が発生する。正常完了に起因して動作が終了した場合、1つの例では、以下が発生する。

- ・モデルに依存する値が、パラメータ・ブロックのモデル・バージョン番号（MVN）フィールド363に格納される。
- ・パラメータ・ブロックの継続フラグ（CF）フィールド373が0に設定される。
- ・パラメータ・ブロックのサブバイト境界（SBB）フィールド381が更新される。
- ・パラメータ・ブロックのブロック終結の長さ（EOBL）389フィールドおよびブロック終結シンボル（EOBS）388フィールドが更新される。
- ・パラメータ・ブロックの履歴長（HL）フィールド385が更新される。
- ・適用可能な場合、パラメータ・ブロックの履歴オフセット（HO）フィールド386が更新される。
- ・パラメータ・ブロックの動作終了補助コード（OESC）フィールド365が0に設定される。
- ・パラメータ・ブロックのチェック値フィールド387が更新される。
- ・汎用レジスタR1内のアドレスが、ビット0の処理を含んでいる第1のオペランドの処理されたバイト数だけインクリメントされ、汎用レジスタR1+1内の長さが同じ数だけ

10

20

30

40

50

デクリメントされる。ビット0の処理を含んでいる第1のオペランドの処理されたバイト数は、整数除算から得られた整数化した商であり、被除数は、処理された出力ビット数およびSBBの元の値の和であり、除数は8の値である。

- ・汎用レジスタR2内のアドレスが、処理されたソース・バイト数だけインクリメントされ、汎用レジスタR2+1内の長さが同じ数だけデクリメントされる。

- ・条件コード0が設定される。

【0164】

アドレスおよび長さの形成および更新は、アドレス指定モードによって決まる。

【0165】

正常完了が発生した場合、動作の終了後に、パラメータ・ブロックのCSBフィールド392が定義されない。

【0166】

CPUによって決定されたバイト数が処理されたときに動作が終了し、1つの例では、以下が発生する。

- ・パラメータ・ブロック内の継続フラグ(CF)ビット373が1に設定される。
- ・パラメータ・ブロック内の継続状態バッファ(CSB)フィールド392が更新される。
- ・パラメータ・ブロックのサブバイト境界(SBB)フィールド381が更新される。
- ・パラメータ・ブロックの履歴長(HL)フィールド385が更新される。
- ・適用可能な場合、パラメータ・ブロックの履歴オフセット(HO)フィールド386が更新される。
- ・パラメータ・ブロックのチェック値フィールド387が更新される。
- ・モデルに依存する値が、パラメータ・ブロックのモデル・バージョン番号(MVN)フィールド363に格納される。
- ・パラメータ・ブロックのブロック終結の長さ(EOBL)389フィールドおよびブロック終結シンボル(EOS)388フィールドが更新される。
- ・パラメータ・ブロックの動作終了補助コード(OESC)フィールド365が0に設定される。

- ・汎用レジスタR1内のアドレスが、ビット0の処理を含んでいる第1のオペランドの処理されたバイト数だけインクリメントされ、汎用レジスタR1+1内の長さが同じ数だけデクリメントされる。ビット0の処理を含んでいる第1のオペランドの処理されたバイト数は、整数除算から得られた整数化した商であり、被除数は、処理された出力ビット数およびSBBの元の値の和であり、除数は8の値である。

- ・汎用レジスタR2内のアドレスが、処理されたソース・バイト数だけインクリメントされ、汎用レジスタR2+1内の長さが同じ数だけデクリメントされる。

- ・条件コード3が設定される。

【0167】

アドレスおよび長さの形成および更新は、アドレス指定モードによって決まる。

【0168】

CPUによって決定されたバイト数は、モデルによって決まり、命令が実行されるたびに異なる数であってよい。

【0169】

条件コード3が設定されて命令が終了した後に、プログラムが命令に関する入力の仕様も、出力の仕様も変更せずに、分岐して戻り、命令を再実行して動作を再開することが期待される。

【0170】

特定の状況では、条件コード3が設定されて命令を終了したにもかかわらず、パラメータ・ブロックおよび汎用レジスタが更新されない。このような状況は、CPUが静止動作(quiescing operation)を実行するか、またはCPUがデフレート変換呼び出し命令の実行中に再試行するときに、発生することがある。そのような場合、CPUによって決定された処理済みのバイト数は0であり、データが第1のオペランドの位置に格納されてい

10

20

30

40

50

てよく、適用可能な場合、データが第3のオペランドの位置に格納されていてよく、対応する変更のビットが設定されている。

【0171】

1つの例では、次の条件のいずれかが当てはまる場合、第1のオペランドの長さは、動作を完了するのに不十分である。

- ・汎用レジスタR1+1の内容によって指定された第1のオペランドの長さが、命令の実行の開始時に0である。
- ・命令の実行中に第1のオペランドの長さが0に等しくなり、正常完了が発生しない。

【0172】

1つの例では、汎用レジスタR1+1の内容が0である場合、パラメータ・ブロックのNTフィールドおよびSBBフィールド内の値に関わらず、第1のオペランドの長さは0である。

【0173】

命令の実行中に第1のオペランドの長さが0に等しくなった場合、動作が終了し、1つの実施形態では以下が発生する。

- ・パラメータ・ブロック内の継続フラグ(CF)ビット373が1に設定される。
- ・パラメータ・ブロック内の継続状態バッファ(CSB)フィールド392が更新される。
- ・パラメータ・ブロックのサブバイト境界(SBB)フィールド381が更新される。
- ・パラメータ・ブロックの履歴長(HL)フィールド385が更新される。
- ・適用可能な場合、パラメータ・ブロックの履歴オフセット(HO)フィールド386が更新される。
- ・パラメータ・ブロックのチェック値フィールド387が更新される。
- ・モデルに依存する値が、パラメータ・ブロックのモデル・バージョン番号(MVN)フィールド363に格納される。
- ・パラメータ・ブロックのブロック終結の長さ(EOBL)389フィールドおよびブロック終結シンボル(EOS)388フィールドが更新される。
- ・パラメータ・ブロックの動作終了補助コード(OESC)フィールド365が0に設定される。
- ・汎用レジスタR1内のアドレスが、ビット0の処理を含んでいる第1のオペランドの処理されたバイト数だけインクリメントされ、汎用レジスタR1+1内の長さが同じ数だけデクリメントされる。ビット0の処理を含んでいる第1のオペランドの処理されたバイト数は、整数除算から得られた整数化した商であり、被除数は、処理された出力ビット数およびSBBの元の値の和であり、除数は8の値である。
- ・汎用レジスタR2内のアドレスが、処理されたソース・バイト数だけインクリメントされ、汎用レジスタR2+1内の長さが同じ数だけデクリメントされる。
- ・条件コード1が設定される。

【0174】

アドレスおよび長さの形成および更新は、アドレス指定モードによって決まる。

【0175】

命令の実行の開始時に第1のオペランドの長さが0である場合、動作が終了し、1つの実施形態では以下が発生する。

- ・条件コード1が設定される。

【0176】

条件コード1が設定されて命令が終了した後に、プログラムが、第1のオペランドの長さ、第1のオペランド・アドレス、またはその両方を変更し、命令を再実行して動作を再開することが期待される。

【0177】

以下に関して、適用可能な場合、PERストレージ変更イベントが認識される。

- ・下で説明されているように、パラメータ・ブロックに格納する。
- ・第1のオペランドの位置に格納する。

10

20

30

40

50

・第3のオペランドの位置に格納し、この格納が、例えば履歴バッファ・タイプ（HBT）が1（円形）である場合に発生する。

【0178】

パラメータ・ブロック全体がPERストレージ領域指定と重なる場合、パラメータ・ブロックに関して、適用可能な場合、PERストレージ変更イベントが認識される。パラメータ・ブロックの一部のみがPERストレージ領域指定と重なる場合、以下のうちのどれが発生するかは、モデルに依存する。

・パラメータ・ブロックに関して、適用可能な場合、PERストレージ変更イベントが認識される。

・格納されたパラメータ・ブロックの一部に関して、適用可能な場合、PERストレージ変更イベントが認識される。

10

【0179】

HBTが1（円形）である場合、パラメータ・ブロック、第1のオペランドの位置、第2のオペランドの位置、および第3のオペランドの位置に関して、適用可能な場合、PERゼロ・アドレス検出イベントが認識される。

【0180】

条件コード2は、DFLTCC-CMPR機能に適用できない。

【0181】

条件コード1または3が設定されて命令が終了するとき、第2のオペランドの位置から参照された入力データは、完全に処理されているか、または部分的にのみ処理されている。入力データが部分的にのみ処理されている場合、パラメータ・ブロックの第1のオペランドの位置、第1のオペランド・アドレス、第1のオペランドの長さ、およびSBBフィールドにおける結果は、更新された第2のオペランドのアドレスおよび長さとは一致する状態を表さない。そのような場合、部分的に処理されたデータおよび内部状態情報が、パラメータ・ブロックのCSBフィールドに配置されてよい。部分的に処理されたデータの量は、動作の終了時に存在する条件およびモデルによって決まる。一部のデータのみが部分的に処理されることがあるが、更新された第1のオペランド・アドレスによって指定された位置の左に格納された結果は完成しており、動作が再開するときに変更されない。さらに、その後、プログラムが命令を再実行して動作を再開することが期待され、そのとき、動作を再開する前に、CSBフィールドの内容が参照される。条件コード0が設定されて命令が終了した場合、すべてのデータが完全に処理されており、入力データおよび出力データに関連付けられたすべての結果が一貫した状態を表す。

20

30

【0182】

0以外の条件コードが設定されて命令が終了した後に、動作を再開する目的で命令を再実行する前に、プログラムはパラメータ・ブロックのどのフィールドも変更するべきではなく、そうでない場合、結果が予測不可能になる。

【0183】

機能コード4：DFLTCC-XPND（展開）

【0184】

DFLTCC-XPND機能が指定された場合、復元動作が実行される。この動作は、第2のオペランドの位置の圧縮データ・シンボルを、第1のオペランドの位置に格納される圧縮されていないデータにデコードすることを含む。

40

【0185】

1つの例では、DFLTCC-XPND機能はパラメータ・ブロックを使用し、その例が図3K～3Lに関して上で説明されている。

【0186】

以下では、データの復元に関してDFLTCC-XPND動作の一例が説明される。

【0187】

第2のオペランド内のデータ・セットの最後のブロックのすべての要素がデコードされ、すべての圧縮されていないデータが第1のオペランドの位置に格納されたときに、正常

50

完了が発生する。ブロック・ヘッダーの `BFINAL` ビットが 1 である場合、データ・セットの最後のブロックが識別される。正常完了に起因して動作が終了した場合、1 つの実施形態では、以下が発生する。

- ・モデルに依存する値が、パラメータ・ブロックのモデル・バージョン番号 (`MVN`) フィールド 3 6 3 に格納される。
- ・パラメータ・ブロックの継続フラグ (`CF`) フィールド 3 7 3 が 0 に設定される。
- ・パラメータ・ブロックのサブバイト境界 (`SB B`) フィールド 3 8 1 が更新される。
- ・パラメータ・ブロックの履歴長 (`HL`) フィールド 3 8 5 が更新される。
- ・適用可能な場合、パラメータ・ブロックの履歴オフセット (`HO`) フィールド 3 8 6 が更新される。
- ・パラメータ・ブロックの圧縮された動的ハフマン・テーブル (`CDHT`) フィールド 3 6 7 および圧縮された動的ハフマン・テーブルの長さ (`CDHTL`) フィールド 3 6 6 が 0 に設定される。
- ・パラメータ・ブロックの動作終了補助コード (`OESC`) フィールド 3 6 5 が 0 に設定される。
- ・パラメータ・ブロックのチェック値フィールド 3 8 7 が更新される。
- ・汎用レジスタ `R1` 内のアドレスが、第 1 のオペランドの位置に格納されたバイト数だけインクリメントされ、汎用レジスタ `R1 + 1` 内の長さが同じ数だけデクリメントされる。
- ・汎用レジスタ `R2` 内のアドレスが、ビット 0 の処理を含んでいる第 2 のオペランドの処理されたバイト数だけインクリメントされ、汎用レジスタ `R2 + 1` 内の長さが同じ数だけデクリメントされる。ビット 0 の処理を含んでいる第 2 のオペランドの処理されたバイト数は、整数除算から得られた整数化した商であり、被除数は、処理された入力ビット数および `SB B` の元の値の和であり、除数は 8 の値である。
- ・条件コード 0 が設定される。

【0188】

アドレスおよび長さの形成および更新は、アドレス指定モードによって決まる。

【0189】

正常完了が発生した場合、動作の終了後に、パラメータ・ブロックの `CSB` フィールド 3 9 2 が定義されない。

【0190】

`CPU` によって決定されたバイト数が処理されたときに動作が終了し、1 つの実施形態では、以下が発生する。

- ・パラメータ・ブロック内の継続フラグ (`CF`) ビット 3 7 3 が 1 に設定される。
- ・パラメータ・ブロック内の継続状態バッファ (`CSB`) フィールド 3 9 2 が更新される。
- ・パラメータ・ブロックのサブバイト境界 (`SB B`) フィールド 3 8 1 が更新される。
- ・パラメータ・ブロックの圧縮された動的ハフマン・テーブル (`CDHT`) フィールド 3 6 7 および圧縮された動的ハフマン・テーブルの長さ (`CDHTL`) フィールド 3 6 6 が更新される。`BTYPE` 値が 2 進数 1 0 であるブロックを処理しているときに部分的完了が発生した場合、テーブルを表すために必要とされない `CDHT` フィールドのバイトが 0 として格納される。`BTYPE` 値が 2 進数 0 0 または 0 1 であるブロックを処理しているときに部分的完了が発生した場合、`CDHT` フィールドおよび `CDHTL` フィールドに 0 が格納される。
- ・パラメータ・ブロックの履歴長 (`HL`) フィールド 3 8 5 が更新される。
- ・適用可能な場合、パラメータ・ブロックの履歴オフセット (`HO`) フィールド 3 8 6 が更新される。
- ・パラメータ・ブロックのチェック値フィールド 3 8 7 が更新される。
- ・モデルに依存する値が、パラメータ・ブロックのモデル・バージョン番号 (`MVN`) フィールド 3 6 3 に格納される。
- ・パラメータ・ブロックの動作終了補助コード (`OESC`) フィールド 3 6 5 が 0 に設定される。

10

20

30

40

50

- ・パラメータ・ブロックの不完全な機能の状態（I F S）フィールド383が更新される。
- ・適用可能な場合、パラメータ・ブロックの不完全な機能の長さ（I F L）フィールド384が更新される。
- ・汎用レジスタR1内のアドレスが、第1のオペランドの位置に格納されたバイト数だけインクリメントされ、汎用レジスタR1+1内の長さが同じ数だけデクリメントされる。
- ・汎用レジスタR2内のアドレスが、ビット0の処理を含んでいる第2のオペランドの処理されたバイト数だけインクリメントされ、汎用レジスタR2+1内の長さが同じ数だけデクリメントされる。ビット0の処理を含んでいる第2のオペランドの処理されたバイト数は、整数除算から得られた整数化した商であり、被除数は、処理された入力ビット数およびSBBの元の値の和であり、除数は8の値である。
- ・条件コード3が設定される。

10

【0191】

アドレスおよび長さの形成および更新は、アドレス指定モードによって決まる。

【0192】

CPUによって決定されたバイト数は、モデルによって決まり、命令が実行されるたびに異なる数であってよい。

【0193】

条件コード3が設定されて命令が終了した後に、プログラムが命令に関する入力の仕様も、出力の仕様も変更せずに、分岐して戻り、命令を再実行して動作を再開することが期待される。

20

【0194】

特定の状況では、条件コード3が設定されて命令を終了したにもかかわらず、パラメータ・ブロックおよび汎用レジスタが更新されない。このような状況は、CPUが静止動作を実行するか、またはCPUがデフレート変換呼び出し命令の実行中に再試行するときに、発生することがある。そのような場合、CPUによって決定された処理済みのバイト数は0であり、データが第1のオペランドの位置に格納されていてよく、適用可能な場合、データが第3のオペランドの位置に格納されていてよく、対応する変更のビットが設定されている。

【0195】

例えば、以下が当てはまる場合、第2のオペランドの長さは、動作を完了するのに不十分である。

30

- ・動作中に、BFINALが1に等しい圧縮データ・ブロックの最後の要素がデコードされておらず、第2のオペランドの長さおよびSBBによって指定された第2のオペランドのビット数が、次にデコードする要素のビット数よりも少なく、第2のオペランドの位置からのデータのデコーディングのすべての結果が第1のオペランドの位置に配置されている。

【0196】

第2のオペランドの長さが、動作を完了するのに不十分である場合、動作が部分的に完了されており、動作が終了し、1つの実施形態では、以下が発生する。

- ・パラメータ・ブロック内の継続フラグ（CF）ビット373が1に設定される。
- ・パラメータ・ブロック内の継続状態バッファ（CSB）フィールド392が更新される。
- ・パラメータ・ブロックのサブバイト境界（SBB）フィールド381が更新される。
- ・パラメータ・ブロックの圧縮された動的ハフマン・テーブル（CDHT）フィールド367および圧縮された動的ハフマン・テーブルの長さ（CDHTL）フィールド366が更新される。BTYP値が2進数10であるブロックを処理しているときに部分的完了が発生した場合、テーブルを表すために必要とされないCDHTフィールドのバイトが0として格納される。BTYP値が2進数00または01であるブロックを処理しているときに部分的完了が発生した場合、CDHTフィールドおよびCDHTLフィールドに0が格納される。

40

- ・パラメータ・ブロックの履歴長（HL）フィールド385が更新される。

50

- ・適用可能な場合、パラメータ・ブロックの履歴オフセット（H O）フィールド 3 8 6 が更新される。
 - ・パラメータ・ブロックのチェック値フィールド 3 8 7 が更新される。
 - ・モデルに依存する値が、パラメータ・ブロックのモデル・バージョン番号（M V N）フィールド 3 6 3 に格納される。
 - ・パラメータ・ブロックの動作終了補助コード（O E S C）フィールド 3 6 5 が 0 に設定される。
 - ・パラメータ・ブロックの不完全な機能の状態（I F S）フィールド 3 8 3 が更新される。
 - ・適用可能な場合、パラメータ・ブロックの不完全な機能の長さ（I F L）フィールド 3 8 4 が更新される。
 - ・汎用レジスタ R 1 内のアドレスが、第 1 のオペランドの位置に格納されたバイト数だけインクリメントされ、汎用レジスタ R 1 + 1 内の長さが同じ数だけデクリメントされる。
 - ・汎用レジスタ R 2 内のアドレスが、ビット 0 の処理を含んでいる第 2 のオペランドの処理されたバイト数だけインクリメントされ、汎用レジスタ R 2 + 1 内の長さが同じ数だけデクリメントされる。ビット 0 の処理を含んでいる第 2 のオペランドの処理されたバイト数は、整数除算から得られた整数化した商であり、被除数は、処理された入力ビット数および S B B の元の値の和であり、除数は 8 の値である。
 - ・条件コード 2 が設定される。
- 【 0 1 9 7 】
- アドレスおよび長さの形成および更新は、アドレス指定モードによって決まる。
- 【 0 1 9 8 】
- 条件コード 2 が設定されて命令が終了した後に、プログラムが、第 2 のオペランドの長さ、第 2 のオペランド・アドレス、またはその両方を変更し、命令を再実行して動作を再開することが期待される。
- 【 0 1 9 9 】
- 例えば、以下が当てはまる場合、第 1 のオペランドの長さは、動作を完了するのに不十分である。
- ・第 1 のオペランドの長さが 0 に等しいため、第 2 のオペランドの位置からのデータのデコーディングの結果を第 1 のオペランドの位置に配置することができない。
- 【 0 2 0 0 】
- 第 1 のオペランドの長さが、動作を完了するのに不十分である場合、動作が部分的に完了されており、動作が終了し、1 つの実施形態では、以下が発生する。
- ・パラメータ・ブロック内の継続フラグ（C F）ビット 3 7 3 が 1 に設定される。
 - ・パラメータ・ブロック内の継続状態バッファ（C S B）フィールド 3 9 2 が更新される。
 - ・パラメータ・ブロックのサブバイト境界（S B B）フィールド 3 8 1 が更新される。
 - ・パラメータ・ブロックの圧縮された動的ハフマン・テーブル（C D H T）フィールド 3 6 7 および圧縮された動的ハフマン・テーブルの長さ（C D H T L）フィールド 3 6 6 が更新される。B T Y P E 値が 2 進数 1 0 であるブロックを処理しているときに部分的完了が発生した場合、テーブルを表すために必要とされない C D H T フィールドのバイトが 0 として格納される。B T Y P E 値が 2 進数 0 0 または 0 1 であるブロックを処理しているときに部分的完了が発生した場合、C D H T フィールドおよび C D H T L フィールドに 0 が格納される。
 - ・パラメータ・ブロックの履歴長（H L）フィールド 3 8 5 が更新される。
 - ・適用可能な場合、パラメータ・ブロックの履歴オフセット（H O）フィールド 3 8 6 が更新される。
 - ・パラメータ・ブロックのチェック値フィールド 3 8 7 が更新される。
 - ・モデルに依存する値が、パラメータ・ブロックのモデル・バージョン番号（M V N）フィールド 3 6 3 に格納される。
 - ・パラメータ・ブロックの動作終了補助コード（O E S C）フィールド 3 6 5 が 0 に設定される。

10

20

30

40

50

- ・パラメータ・ブロックの不完全な機能の状態（I F S）フィールド383が更新される。
- ・適用可能な場合、パラメータ・ブロックの不完全な機能の長さ（I F L）フィールド384が更新される。
- ・汎用レジスタR1内のアドレスが、第1のオペランドの位置に格納されたバイト数だけインクリメントされ、汎用レジスタR1+1内の長さが同じ数だけデクリメントされる。
- ・汎用レジスタR2内のアドレスが、ビット0の処理を含んでいる第2のオペランドの処理されたバイト数だけインクリメントされ、汎用レジスタR2+1内の長さが同じ数だけデクリメントされる。ビット0の処理を含んでいる第2のオペランドの処理されたバイト数は、整数除算から得られた整数化した商であり、被除数は、処理された入力ビット数およびS B Bの元の値の和であり、除数は8の値である。
- ・条件コード1が設定される。

10

【0201】

アドレスおよび長さの形成および更新は、アドレス指定モードによって決まる。

【0202】

条件コード1が設定されて命令が終了した後に、プログラムが、第1のオペランドの長さ、第1のオペランド・アドレス、またはその両方を変更し、命令を再実行して動作を再開することが期待される。

【0203】

以下に関して、適用可能な場合、P E Rストレージ変更イベントが認識される。

- ・本明細書において説明されているように、パラメータ・ブロックに格納する。
- ・第1のオペランドの位置に格納する。
- ・第3のオペランドの位置に格納し、この格納が、例えば履歴バッファ・タイプ（H B T）が1（円形）である場合に発生する。

20

【0204】

1つの例では、パラメータ・ブロック全体がP E Rストレージ領域指定と重なる場合、パラメータ・ブロックに関して、適用可能な場合、P E Rストレージ変更イベントが認識される。パラメータ・ブロックの一部のみがP E Rストレージ領域指定と重なる場合、1つの実施形態では、以下のうちのどれが発生するかは、モデルに依存する。

- ・パラメータ・ブロックに関して、適用可能な場合、P E Rストレージ変更イベントが認識される。

30

- ・格納されたパラメータ・ブロックの一部に関して、適用可能な場合、P E Rストレージ変更イベントが認識される。

【0205】

H B Tが1（円形）である場合、パラメータ・ブロック、第1のオペランドの位置、第2のオペランドの位置、および第3のオペランドの位置に関して、適用可能な場合、P E Rゼロ・アドレス検出イベントが認識される。

【0206】

条件コード1、2、または3が設定されて命令が終了するとき、第2のオペランドの位置から参照された入力データは、完全に処理されているか、または部分的にのみ処理されていてよい。入力データが部分的にのみ処理されている場合、第1のオペランドの位置、第1のオペランド・アドレス、第1のオペランドの長さ、パラメータ・ブロックのS B Bフィールド、パラメータ・ブロックのチェック値フィールド、パラメータ・ブロックのH Lフィールド、パラメータ・ブロックのI F Sフィールド、ならびに適用可能な場合、第3のオペランドの位置およびパラメータ・ブロックのH Oフィールドにおける結果が、更新された第2のオペランドのアドレスおよび長さとは一致する状態を表さない。そのような場合、部分的に処理されたデータおよび内部状態情報が、パラメータ・ブロックのC S Bフィールドに配置されてよい。部分的に処理されたデータの量は、動作の終了時に存在する条件およびモデルによって決まる。一部のデータのみが部分的に処理されることがあるが、更新された第1のオペランド・アドレスによって指定された位置の左に格納された結果は完成しており、動作が再開するときに変更されない。さらに、その後、プログラムが

40

50

命令を再実行して動作を再開することが期待され、そのとき、動作を再開する前に、CSBフィールドの内容が参照される。条件コード0が設定されて動作が終了した場合、すべてのデータが完全に処理されており、入力データおよび出力データに関連付けられたすべての結果が一貫した状態を表す。

【0207】

0以外の条件コードが設定されて命令が終了した後に、動作を再開する目的で命令を再実行する前に、プログラムはパラメータ・ブロックのどのフィールドも変更するべきではなく、そうでない場合、結果が予測不可能になる。

【0208】

圧縮データ・ブロック

【0209】

1つの例では、ストレージ内の圧縮データ・ブロックのバイトが、例えば左から右に処理される。圧縮データ・ブロックは、バイト境界上で開始または終了してもしなくてもよい。例えば、圧縮データ・ブロックはビット・ストリームである。ブロックの要素は、一度に1ビットずつ、ストレージに読み込まれる。ビット・ストリームは、ストレージの各バイト内では例えば右から左へ読み込まれ、バイト・オーダーでは例えば左から右へ読み込まれる。要素がハフマン・コードである場合、ビットは、例えば要素の最上位ビットから最下位ビットへの順序で格納される。要素がハフマン・コードでない場合、ビットは、例えば要素の最下位ビットから最上位ビットへの順序で格納される。

【0210】

図6は、圧縮データ・シンボルを含んでいない、ブロック・タイプが2進数00であるブロック600の例を示している。1つの実施形態では、以下がこの例に当てはまる。

- ・圧縮データ・ブロック600が、b0として識別されるバイト0のビット4から開始し、b60として識別されるバイト7のビット0で終了するビット・ストリーム602で構成される。

- ・ビット・ストリームに出現する最初の要素が、バイト0のビット4におけるBFINAL（ブロック・ヘッダーの最後のビット）である。

- ・ビット・ストリームに出現する2番目の要素が、バイト0のビット2～3におけるBTYP（ブロック・タイプ）である。この例では、BTYPは2進数00である。

- ・BTYPが2進数00である場合、BTYPの左にあるビット（この例では、バイト0のビット0～1）およびバイト境界の右にあるビットが無視される。

- ・ビット・ストリームに出現する3番目の要素が、LENフィールドの最下位バイト（LSB：least significant byte）であり、その後にLENフィールドの最上位バイト（MSB：most significant byte）が続く。LENフィールドは、リテラル・データを含むブロック内のバイト数を指定する。リテラル・データは、例えば圧縮されていないデータである。リテラル・データを含むバイトは、ビット・ストリーム内のNLENフィールドの後に続く。NLENは、LENの1の補数である。1つの例では、バイト1～2が、リトルエンディアン・バイト・オーダーでLENフィールドを含む。

- ・LENフィールドの後に続いてビット・ストリームに出現する要素が、NLENフィールドの最下位バイトであり、その後にNLENフィールドの最上位バイトが続く。バイト3～4が、リトルエンディアン・バイト・オーダーでNLENフィールドを含む。NLENフィールドは、LENフィールドの1の補数である。

- ・NLENフィールドの後に続いてビット・ストリームに出現する要素が、リテラル・バイトとして識別される圧縮されていないデータである。バイト5～7は、このブロックを生成するために使用されるソース・データから変更されない、圧縮されていないデータを含む。

- ・このブロックに含まれているどの要素も、ハフマン・コードではない。このブロック内のすべての要素は、デフレート規格によって規定されているように、要素の最下位ビットから最上位ビットへの順序で、ビット・ストリームに格納される。LEN、NLEN、およびリテラルの各要素が、バイト境界上でアライメントされた整数個のバイトであるため

10

20

30

40

50

、これらの要素は、バイト単位で処理されてよく、必ずしもビット単位で処理されなくてよい。

【0211】

図7は、固定ハフマン・テーブル（FHT）を使用して生成された圧縮データ・シンボルを含んでいる、ブロック・タイプが2進数01であるブロック700の例を示している。1つの実施形態では、以下がこの例に当てはまる。

- ・圧縮データ・ブロック700が、b0として識別されるバイト0のビット4から開始し、b89として識別されるバイト11のビット3で終了するビット・ストリーム702で構成される。

- ・ビット・ストリームに出現する最初の要素が、バイト0のビット4におけるBFINALである。

- ・ビット・ストリームに出現する2番目の要素が、バイト0のビット2～3におけるBTYP Eである。この例では、BTYP Eは2進数01である。

- ・固定ハフマン・テーブル（FHT）が、ブロックの構成要素ではない。

- ・ビット・ストリームに出現する第3の要素が、バイト0のビット1で開始する最初の圧縮データ・シンボルである。圧縮データ・シンボルは、1つの例では、示されている順序でビット・ストリームに出現する以下の部分要素で構成される。

1. 変数の長さのハフマン・コード。このコードの最上位ビットは、コードの長さを指定する。このコードは、コードの最上位ビットから開始してビット・ストリームに出現し、コードの最下位ビットで終了する。このコードがリテラル値またはブロック終結シンボルを表す場合、このコードは、圧縮データ・シンボルの単なる部分要素である。このコードが履歴バッファへのポインタの長さを表す場合、このコードの後に、圧縮データ・シンボルのその後の部分要素が続く。

2. 適用可能な場合、デフレート規格によって規定されているように、ポインタの長さを表すハフマン・コードの後に、追加の長さビットが続いてよい。追加の長さビットは、追加の長さビットの最下位ビットから開始してビット・ストリームに出現し、追加の長さビットの最上位ビットで終了する。

3. 次にビット・ストリームに出現する部分要素は、履歴バッファへのポインタの5ビットの距離コードである。この距離コードは、例えばコードの最上位ビットから開始してビット・ストリームに出現し、距離コードの最下位ビットで終了する。

4. 適用可能な場合、デフレート規格によって規定されているように、距離コードの後に、追加の距離ビットが続いてよい。追加の距離ビットは、追加の距離ビットの最下位ビットから開始してビット・ストリームに出現し、追加の距離ビットの最上位ビットで終了する。

- ・一例として、バイト0のビット0～1、バイト1～9のすべてのビット、およびバイト10のビット2～7が、圧縮データ・シンボルのビットを含む。

- ・ビット・ストリームに出現する最後の要素が、単一の部分要素を含んでいる圧縮データ・シンボルであり、この部分要素は、ブロック終結（EOB）シンボルを表すハフマン・コードである。BTYP Eが2進数01であるブロックのEOBシンボルは、2進数00000000である。この例では、バイト10のビット1が、EOBシンボルの最上位ビットを含み、バイト11のビット3が、EOBシンボルの最下位ビットを含む。

- ・バイト11のビット3が、圧縮データ・ブロックの最後のビットである、ビット・ストリームの最後のビットを含む。

【0212】

図8は、動的ハフマン・テーブル（DHT）を使用して生成された圧縮データ・シンボルを含んでいる、ブロック・タイプが2進数10であるブロック800の例を示している。1つの実施形態では、以下がこの例に当てはまる。

- ・圧縮データ・ブロック800が、b0として識別されるバイト0のビット4から開始し、b89として識別されるバイト11のビット3で終了するビット・ストリーム802で構成される。

・ビット・ストリームに出現する最初の要素が、バイト 0 のビット 4 における B F I N A L である。

・ビット・ストリームに出現する 2 番目の要素が、バイト 0 のビット 2 ～ 3 における B T Y P E である。この例では、B T Y P E は 2 進数 1 0 である。

・ビット・ストリームに出現する第 3 の要素が、バイト 0 のビット 1 で開始する動的ハフマン・テーブル (D H T) の圧縮表現である。D H T の圧縮表現は、1 つの例では、示されている順序でビット・ストリームに出現する以下の部分要素で構成される。

1. H L I T : 5 ビットの H L I T 部分要素および 2 5 7 の和が、リテラル・バイト、E O B シンボル、および重複列の長さを表すハフマン・コードの数を指定する。H L I T の有効な値の範囲は、例えば 0 ～ 2 9 である。H L I T ビットは、H L I T 部分要素の最下位ビットから開始してビット・ストリームに出現し、H L I T 部分要素の最上位ビットで終了する。この例では、b 3 として識別されるバイト 0 のビット 1 が、H L I T 部分要素の最下位ビットである。

10

2. H D I S T : 5 ビットの H D I S T 部分要素および 1 の和が、重複列のポインタ距離を表すハフマン・コードの数を指定する。H D I S T の有効な値の範囲は、例えば 0 ～ 2 9 である。H D I S T ビットは、H D I S T 部分要素の最下位ビットから開始してビット・ストリームに出現し、H D I S T 部分要素の最上位ビットで終了する。

3. H C L E N : 4 ビットの H C L E N 部分要素および 4 の和が、コード長を表すハフマン・コードの数を指定する。H C L E N の有効な値の範囲は、例えば 0 ～ 1 5 である。H C L E N ビットは、H C L E N 部分要素の最下位ビットから開始してビット・ストリームに出現し、H C L E N 部分要素の最上位ビットで終了する。

20

4. 圧縮された D H T に対して定義されたコード長の各々に対してビット長を指定するコードのシーケンス。コードの数は、H C L E N および 4 の和に等しい。各コードは 3 ビットである。

5. リテラル・バイト、E O B シンボル、および重複列の長さで構成されるセットの要素の各々に対してコード長を指定するコードのシーケンス。指定されるコード長の数は、H L I T および 2 5 7 の和に等しい。

【 0 2 1 3 】

リテラル・バイト、E O B シンボル、および重複列の長さのセットの最後のコード長 (C L) が 1 6、1 7、または 1 8 であり、C L の後に続く追加のビットが、このセットに対して定義されている要素よりも多い要素に対して C L を繰り返すことを指定する場合、このコード長は、重複列のポインタ距離のセットにも適用される。リテラル・バイト、E O B シンボル、および重複列の長さのセットに対してコード長を指定するコードのシーケンス、およびその後続く、重複列のポインタ距離に対してコード長を指定するコードのシーケンスは、両方のセットに関する隣接するシーケンスである。

30

6. 重複列のポインタ距離で構成されるセットの要素の各々に対してコード長を指定するコードのシーケンス。指定されるコード長の数は、H D I S T および 1 の和に等しい。

・ビット・ストリームに出現する第 4 の要素が、最初の圧縮データ・シンボルである。圧縮データ・シンボルは、1 つの実施形態では、示されている順序でビット・ストリームに出現する以下の部分要素で構成される。

40

1. 変数の長さのハフマン・コード。このコードの最上位ビットは、コードの長さを指定する。このコードは、コードの最上位ビットから開始してビット・ストリームに出現し、コードの最下位ビットで終了する。このコードがリテラル値またはブロック終結シンボルを表す場合、このコードは、圧縮データ・シンボルの単なる部分要素である。このコードが履歴バッファへのポインタの長さを表す場合、このコードの後に、圧縮データ・シンボルのその後の部分要素が続く。

2. 適用可能な場合、デフレート規格によって規定されているように、ポインタの長さを表すハフマン・コードの後に、追加の長さビットが続いてよい。追加の長さビットは、例えば追加の長さビットの最下位ビットから開始してビット・ストリームに出現し、追加の長さビットの最上位ビットで終了する。

50

３．次にビット・ストリームに出現する部分要素は、履歴バッファへのポインタの５ビットの距離コードである。この距離コードは、例えばコードの最上位ビットから開始してビット・ストリームに出現し、距離コードの最下位ビットで終了する。

４．適用可能な場合、デフレート規格によって規定されているように、距離コードの後に、追加の距離ビットが続いてよい。追加の距離ビットは、例えば追加の距離ビットの最下位ビットから開始してビット・ストリームに出現し、追加の距離ビットの最上位ビットで終了する。

- ・ビット・ストリームに出現する、例えばバイト１０のビット５までの、ビット５を含んでいるその後のビットは、圧縮データ・シンボルのビットを含む。

- ・ビット・ストリームに出現する最後の要素が、単一の部分要素を含んでいる圧縮データ・シンボルであり、この部分要素は、ブロック終結（EOB）シンボルを表すハフマン・コードである。この例では、バイト１０のビット４が、EOBシンボルの最上位ビットを含み、バイト１１のビット３が、EOBシンボルの最下位ビットを含む。

- ・バイト１１のビット３が、圧縮データ・ブロックの最後のビットである、ビット・ストリームの最後のビットを含む。

【０２１４】

さまざまなブロック・タイプの上記の説明では、特定の一定の値に加えて、特定のビット、バイト、方向などが指定された。これらは例にすぎない。他の実施形態では、その他の一定の値、ビット、バイト、方向などが指定されてよい。

【０２１５】

圧縮データ・セットの処理

【０２１６】

デフレート変換呼び出し命令の使用例を説明し、パラメータ・ブロックのさまざまなフィールドの説明を補強するための圧縮データ・セットの処理の例が提供される。これらの例は、すべての可能なシナリオ、要件、および機能を説明するのではなく、さまざまなシナリオ、要件、または機能、あるいはその組み合わせを説明する。これらの例および説明は、例えばストレージ内の圧縮データ・セットに適用され、その例が図９に示されている。図に示されているように、圧縮データ・セット９００は、複数の圧縮データ・ブロック９０２を含んでおり、データ・セット９００の先頭が、圧縮データ・セット開始アドレス（CDSBA: compressed data set begin address）９０４によって示されている。

【０２１７】

本明細書に記載された例では、圧縮データ・セットを処理するプログラムが、１つの実施形態において以下を考慮することが意図されている。

- ・単一のパラメータ・ブロックが定義され、圧縮データ・セット全体を処理するために、デフレート変換呼び出し命令の複数の使用によって参照されてよい。パラメータ・ブロックのチェック値フィールド３８７およびチェック値タイプ・フィールド３７５が、圧縮データ・セット内の圧縮データ・ブロック（例えば、すべてのブロック）に適用されるべきである。パラメータ・ブロックのサブバイト境界・フィールド３８１が、個別のブロック間の移行に適用されるべきである。履歴長３８５および履歴オフセット３８６が複数のブロックに適用されてよい。パラメータ・ブロックの残りのフィールドは、１つの例では、デフレート変換呼び出し命令の特定の実行によって処理されている個別の圧縮データ・ブロックのみに適用される。

- ・個別のチェック値が、例えば、圧縮データ・セットによって表された圧縮されていないデータのすべてに適用される。

- ・ブロック１には、第１の圧縮データ・シンボルに関して参照する履歴が存在しない。ブロック１内のその後のシンボルは、以前にブロック１に出現したシンボルに対応する履歴を参照してよい。ブロック２内のシンボルは、以前にブロック２および１に出現したシンボルに対応する履歴を参照してよい。ブロック３内のシンボルは、以前にブロック３、２、および１に出現したシンボルに対応する履歴を参照してよい。

【０２１８】

10

20

30

40

50

図10は、図9で説明された圧縮データ・セット900内のデータを圧縮するために使用されるサンプル・プログラム1000の一部の一例を示している。さらに、図11は、図10のIABLK1(1002)というラベルが付けられた命令アドレスにあるDFLTCC命令の実行中に使用されるパラメータ・ブロックの特定のフィールドの値を示している。例えば、図11は、パラメータ・ブロックのさまざまなフィールド1100、圧縮動作の開始時のそれらのフィールドの値1102、条件コード1、2、または3が設定されたときの動作の終了時のそれらのフィールドの値1104、および条件コード0が設定されたときの動作の終了時のそれらのフィールドの値1106を示している。

【0219】

同様に、図12は、図10のIABLK2(1004)というラベルが付けられた命令アドレスにあるDFLTCC命令の実行中に使用されるパラメータ・ブロックの特定のフィールドの値を示している。これらの図は、デフレート変換呼び出し命令を複数回使用して圧縮データ・セット全体を処理することに関連する詳細の一部を示している。

10

【0220】

さらに、図13を参照すると、図9の圧縮データ・セットからデータを復元するために使用されるサンプル・プログラム1300の一部の一例が示されている。

【0221】

データの圧縮

【0222】

データ圧縮のプロセスは、1つまたは複数の圧縮データ・ブロックを生成することを含む。デフレート変換呼び出し命令の圧縮機能は、個別のブロックの部分を構築するために使用される。この部分は、ブロック全体であってよい。この機能は、ブロック・タイプ(BTYPE)が2進数00ではなく2進数01または10であるブロックの部分を生成する。パラメータ・ブロックの新しいタスク(NT)ビットが1である場合、圧縮データの最初のブロックが生成され、以前に実行された圧縮動作からの参照する履歴が存在しない。

20

【0223】

1つの例では、個別のブロックが、以下の要素を示されている順序で含む。

1. 最後のブロックの指示(BFINAL)。
2. ブロック・タイプ(BTYPE)。
3. 適用可能な場合、動的ハフマン・テーブルの圧縮形式。
4. 圧縮データ・シンボル。
5. ブロック終結(EOB)シンボル。

30

【0224】

圧縮動作は、ブロックに対して定義された順序で、指定された要素を生成する。要素は、ストレージ内のバイト境界間で開始または終了してよい。サブバイト境界(SBB)が、第1のオペランドの位置への第1の要素の格納に適用される。圧縮データ・ブロックはビット・ストリームである。ブロックの構成要素は、一度に1ビットずつ、ストレージに読み込まれる。一例として、ビット・ストリームは、ストレージの各バイト内では右から左へ読み込まれ、バイト・オーダーでは左から右へ読み込まれる。

【0225】

SBBが0以外である場合、第1のオペランドの位置での最初のバイトへの参照は、更新参照である。

40

【0226】

第2のオペランドの位置からの圧縮されていないデータが圧縮され、圧縮データ・シンボルとして第1のオペランドの位置に格納される。

【0227】

命令の実行の開始時に第1のオペランドの長さが0であるときに、第1のオペランドはアクセスされず、汎用レジスタR1内の第1のオペランド・アドレスおよび汎用レジスタR1+1内の第1のオペランドの長さは変更されない。これは、命令の実行の開始時にCFフィールド373(図3L)の値が0または1である場合に当てはまる。

50

【0228】

命令の実行の開始時に第2のオペランドの長さが0であるときに、第2のオペランドはアクセスされず、汎用レジスタR2内の第2のオペランド・アドレスおよび汎用レジスタR2+1内の第2のオペランドの長さは変更されない。一例として、以下の場合、命令の実行の開始時に第2のオペランドの長さは0である。

- ・動作を再開するために命令が再実行されており（命令の実行の開始時にパラメータ・ブロックのCFフィールド373が1であり）、第2のオペランドを参照せずに、パラメータ・ブロックのCSBフィールド392を参照して、動作を完了させることができる。

【0229】

1つの実施形態では、プログラムは、以下の動作を実行するためにデフレート変換呼び出し命令を使用しない。

- ・空の圧縮データ・ブロックを生成する。空の圧縮データ・ブロックは、例えばブロック・ヘッダー、適用可能な場合、圧縮形式のDHT、およびEOBシンボルで構成される。
- ・開いている圧縮データ・ブロックを閉じる。すなわち、単にEOBシンボルを圧縮データ・ブロックの末尾に格納する。

【0230】

圧縮アルゴリズムは、第2のオペランドの位置からの現在圧縮されているデータに一致するバイト列に関して、最近圧縮されたデータの最新の履歴を検索することを含む。圧縮動作が開始または再開する前に、1つの実施形態では、以下が当てはまる。

- ・新しいタスク（NT）374が1である場合、参照できる初期履歴が存在しない。
- ・NTが0であり、汎用レジスタ0のビット56（HBT）が0（直列）である場合、参照できる初期履歴が、第2のオペランドの左端のバイトの左に隣接して存在し、初期履歴の長さが、パラメータ・ブロックの履歴長（HL）フィールド385によって指定される。
- ・NTが0であり、汎用レジスタ0のビット56（HBT）が1（円形）である場合、参照できる初期履歴が、パラメータ・ブロックの履歴オフセット（HO）386フィールドおよび履歴長（HL）385フィールドによって指定された第3のオペランドの位置に存在する。

【0231】

圧縮動作中に、動作を実行するために履歴のどのバイトが使用されるかに関わらず、履歴全体へのフェッチタイプの参照が行われてよい。さらに、履歴バッファ・タイプが円形である場合、動作を実行するために履歴のどのバイトが使用されるかに関わらず、32Kバイトの履歴バッファ全体へのフェッチタイプの参照が行われてよい。

【0232】

圧縮動作中に、履歴が更新される。一般オペランド・データ例外条件が発生せずに、ソース・データの1つまたは複数のバイトを圧縮データ・シンボルにエンコードした後に、ソース・バイトが履歴の末尾に連結される。最大で32Kバイトまでの、ソース・データの直前に処理されたバイトが、ソース・データのその後のバイトの処理中に参照できる最新の履歴を構成する。

【0233】

圧縮動作の終了時に、1つの例では、その後、動作を再開するため、または別の動作を開始するために使用できる得られた履歴に、以下が当てはまる。

- ・HBTが直列である場合、履歴が更新されるときに、第2のオペランドの位置に対するストレージの更新が不要である。更新された第2のオペランド・アドレスおよび更新されたHLは、得られた履歴の更新された位置および更新された長さを指定する。
- ・HBTが円形である場合、履歴が更新されるときに、第3のオペランドの位置に対するストレージの更新が実行される。第3のオペランド・アドレス、更新されたHO、および更新されたHLは、得られた履歴の更新された位置および更新された長さを指定する。

【0234】

例として、図14A～14Cは、DFLTCC-CMPR機能が指定され、かつ直列履歴が指定された（例えば、ビット310=0）デフレート変換呼び出し命令の複数の実行

10

20

30

40

50

の前後で、第2のオペランドに関して、各実行が部分的に完了して終了したときの直列履歴バッファの位置を示している。例えば、図14Aは、DFLTCC-CMPRの実行番号1の前の直列履歴を示しており、図14Bは、DFLTCC-CMPRの実行番号2の前、および実行番号1の後の直列履歴を示しており、図14Cは、DFLTCC-CMPRの実行番号2の後の直列履歴を示している。図14Cで提供される説明は、図14Aおよび14Bにも当てはまる。

【0235】

汎用レジスタ0のビット56によって指定されたHBT（履歴バッファ・タイプ）が円形（例えば、ビット310=1）である場合、履歴は、例えば第3のオペランドの位置にある32Kバイトのバッファ内で維持される。バッファ内の履歴の最初のバイトの位置（HB）は、例えば、汎用レジスタR3の内容および履歴オフセット（HO）386（図3L）の和によって指定される。履歴の最初のバイトは、バッファ内の圧縮されていないデータのうちの処理されてから最も長い時間が経過したバイトである。バッファ内の履歴の最後のバイトの位置（HE）は、一例として、以下の方程式によって指定される。

【0236】

$$HE = R3 + \text{modulo } 32K (HO + HL - 1)$$

【0237】

履歴の最後のバイトは、バッファ内の圧縮されていないデータのうちの直前に処理されたバイトである。履歴オフセット（HO）386（図3L）および履歴長（HL）385の和が、第3のオペランドのサイズ（例えば、32Kバイト）を超えた場合、履歴は、第3のオペランドの末尾から第3のオペランドの先頭に戻る。

【0238】

例として、図15A～15Eは、DFLTCC-CMPR機能が指定され、かつ円形履歴バッファが指定された（ビット310=1）デフレート変換呼び出し命令の複数の実行の前後で、各実行が部分的に完了して終了したときの円形履歴バッファ内の履歴の位置を示している。例えば、図15Aは、DFLTCCの実行番号1の前の円形履歴バッファを示しており、図15Bは、DFLTCCの実行番号2の前、および実行番号1の後の円形履歴バッファを示しており、図15Cは、DFLTCCの実行番号3の前、および実行番号2の後の円形履歴バッファを示しており、図15Dは、DFLTCCの実行番号4の前、および実行番号3の後の円形履歴バッファを示しており、図15Eは、DFLTCCの実行番号4の後の円形履歴バッファを示している。図15Eで提供される説明は、図15A～15Dにも当てはまる。

【0239】

HBTが円形であり、第2のオペランドの位置から処理されたバイト数が例えば32,768未満である場合、1つの例では以下が当てはまる。

・第3のオペランドの位置のバイトの範囲への格納が行われる。このバイトの範囲は、例えば次式によって指定された位置から開始し、この位置を含む。

【0240】

$$R3 + \text{modulo } 32K (HOO + HLO)$$

ここで、

【0241】

HOO：命令が実行される前の履歴オフセット。

【0242】

HLO：命令が実行される前の履歴長。

【0243】

このバイトの範囲は、例えば次式によって指定された位置で終了し、この位置を含む。

【0244】

$$R3 + \text{modulo } 32K (HOO + HLO + BP - 1)$$

ここで、

【0245】

10

20

30

40

50

B P：命令の実行中に第2のオペランドの位置から処理されたバイト数。

【0246】

前述したバイトの範囲に対して行われる格納は、一例として、格納タイプ・アクセス例外、P E Rストレージ変更イベント、および設定変更ビットの対象になる。

・ストレージ位置の内容を変更しない、必要ではない格納が、前述した範囲に含まれない第3のオペランドの位置のバイトに対して行われてよい。そのような位置への格納も、格納タイプ・アクセス例外、P E Rストレージ変更イベント、および設定変更ビットの対象になる。

【0247】

H B Tが円形であり、第2のオペランドの位置から処理されたバイト数が例えば32，768以上である場合、格納は、第3のオペランドの位置のすべてのバイトに対して行われ、格納タイプ・アクセス例外、P E Rストレージ変更イベント、および設定変更ビットの対象になる。

【0248】

ブロック継続フラグ(B C F)377が0である場合、B F I N A Lおよびその後続くB T Y P Eを含む3ビットのブロック・ヘッダーが、第1のオペランドの位置に格納される。ブロック・ヘッダーのB F I N A Lビットは、パラメータ・ブロックのブロック・ヘッダーの最後のビット(B H F)379と同じに設定される。ハフマン・テーブル・タイプ(H T T)376が0である場合、ブロック・ヘッダーのB T Y P Eフィールドが例えば2進数01に設定され、H T Tが1である場合、ブロック・ヘッダーのB T Y P Eフィールドが例えば2進数10に設定される。ブロック・ヘッダーが格納されるときに、B F I N A Lビットが、第1のオペランドの最初のバイト内のS B Bによって指定されたビットに格納される。その後、B T Y P Eが第1のオペランドの位置に格納される。B C Fが1である場合、ブロック・ヘッダーは格納されない。

【0249】

ハフマン・テーブル・タイプ(H T T)が1である場合、パラメータ・ブロックで指定された動的ハフマン・テーブル(D H T)367の圧縮形式が、一般オペランド・データ例外条件に関して検査される。D H Tの指定された圧縮形式に関して一般オペランド・データ例外条件が存在する場合、圧縮されたD H Tは無効であると見なされ、データの圧縮に使用されない。以下では、一般オペランド・データ例外条件の例示的な定義がさらに説明される。圧縮形式のD H Tが、コード長に対するビット長、あるいはリテラル・バイト、E O Bシンボル、重複列の長さ、または重複列のポインタ距離に対するコード長を指定し、これらの長さが、適切な機能するハフマン・ツリーを指定するためにハフマン・アルゴリズムによって必要とされる長さよりも大きい場合、機能するD H Tを導出してデータを圧縮するために、圧縮されたD H Tがまだ使用される。ブロック継続フラグ(B C F)が0であり、H T Tが1である場合、パラメータ・ブロックのC D H Tフィールド367で指定された圧縮形式のD H Tが、第1のオペランドの位置に格納される。

【0250】

圧縮動作中に、第2のオペランドの位置からのソース・データが、圧縮データ・シンボルにエンコードされる。このエンコーディングの一部として、ソース・データが履歴と比較される。一致が検出されない場合、ソース・データの間中表現は、ソース・データと同じであるリテラル・バイトである。一致が検出された場合、ソース・データの間中表現は、ソース・データの複製を含んでいる履歴内の位置へのポインタである。ポインタは、長さおよび距離で構成される。この長さは、履歴内の列に一致するソース・データのバイト数である。この距離は、履歴の末尾から、ソース・データに一致する列の先頭までのバイト数である。1つの例では、ソース・データの間中表現を圧縮データ・シンボルにエンコードするために、ハフマン・テーブルからの2つのハフマン・コード・ツリーが使用される。ハフマン・テーブル・タイプ(H T T)が0である場合、デフレート規格に記載されているように、固定ハフマン・テーブル(F H T)が、中間結果をエンコードするために使用される2つのハフマン・コード・ツリーを指定する。H T T376が1である場合、

10

20

30

40

50

パラメータ・ブロックのCDHTフィールド367で指定されたDHTの圧縮表現から導出された動的ハフマン・テーブル(DHT)が、中間結果をエンコードするために使用される2つのハフマン・コード・ツリーを指定する。このエンコーディングは、デフレート規格に記載されている通りに実行される。ソース・データの間接表現をエンコードするために使用されるハフマン・コードを指定しない非汎用DHTが使用される場合、一般オペランド・データ例外が認識される。得られた圧縮データ・シンボルのビットは、結果を第1のオペランドの位置に格納する前に、デフレート規格によって指定された順序で配置される。

【0251】

1つの例では、重複列の長さの範囲は3～258バイトである。

10

【0252】

ソース・データをさらに処理する前に、本明細書において説明されているように、履歴が更新される。

【0253】

このプロセスは、1つの例では、すべてのソース・バイトが処理されるまで繰り返される。

【0254】

ソース・バイト(例えば、すべてのソース・バイト)が処理され、ブロック終了制御(BCC)378が1になった後に、ブロック終結(EOB)シンボルが第1のオペランドの位置に格納される。固定ハフマン・テーブルが使用される場合、2進数00000000のハフマン・コードがEOBシンボルに使用される。動的ハフマン・テーブル(DHT)が使用される場合、EOBシンボルに使用されるハフマン・コードは、DHTによって指定される。EOBシンボルのビットは、EOBシンボルを第1のオペランドの位置に格納する前に、デフレート規格によって指定された順序で配置される。

20

【0255】

動作の最後の圧縮データ・シンボル(EOBシンボルを含む)が、格納する最後のバイトの一部のみを占める場合、この最後のシンボルの一部を含まないビットは、1つの例では0として格納される。

【0256】

最後の圧縮データ・シンボルを処理した後に、1つの実施形態では、以下が発生する。

- ・モデルに依存する値が、パラメータ・ブロックのモデル・バージョン番号(MVN)フィールド363に格納される。
- ・パラメータ・ブロックのサブバイト境界(SBB)フィールド381が更新される。
- ・パラメータ・ブロックのブロック終結の長さ(EOBL)389フィールドおよびブロック終結シンボル(EOS)388フィールドが更新される。
- ・汎用レジスタR1内のアドレスが、ビット0の処理を含んでいる第1のオペランドの処理されたバイト数だけインクリメントされ、汎用レジスタR1+1内の長さが同じ数だけデクリメントされる。ビット0の処理を含んでいる第1のオペランドの処理されたバイト数は、整数除算から得られた整数化した商であり、被除数は、処理された出力ビット数およびSBBの元の値の和であり、除数は8の値である。
- ・汎用レジスタR2内のアドレスが、処理されたソース・バイト数だけインクリメントされ、汎用レジスタR2+1内の長さが同じ数だけデクリメントされる。

30

40

【0257】

アドレスおよび長さの形成および更新は、アドレス指定モードによって決まる。

【0258】

ソース・データを圧縮することと一致して、ソース・データは、前述した32ビットのチェック値を生成するための入力になる。得られたチェック値は、パラメータ・ブロックのチェック値フィールド387に格納される。

【0259】

データの復元

50

【0260】

1つの実施形態では、デフレート変換呼び出し命令の展開機能が、圧縮データ・セットを圧縮されていないデータにデコードするために使用される。第2のオペランドの位置の圧縮データ・セットは、1つまたは複数の連続する圧縮データ・ブロックを含む。データ・セットのブロックは、1つの例では左から右へ処理され、ブロックのバイトは、例えば左から右へ処理される。ブロックは、バイト境界上で開始または終了してもしなくてもよい。各ブロックは、データ・セット内の他のブロックから独立してデコードされる。汎用レジスタR2は、データ・セット内の最初のブロックの左端のバイトの論理アドレスを指定する。データ・セット内の最後のブロックは、処理中に出現する、BFINALビットが1に等しいブロックである。1つの例では、処理する3つのタイプのブロックが存在する。ブロックの内容をデコードする手法は、ブロック・タイプ(BTYPE)に応じて変わる。

10

【0261】

動作が開始するとき(例えば、パラメータ・ブロックの継続フラグ・フィールド373が0であるとき)、汎用レジスタR2、新しいタスク(NT)フィールド374、およびサブバイト境界(SBB)フィールド381によって指定されたビットは、圧縮データ・ブロックの最初のビット(ブロック・ヘッダーのBFINALビット)として解釈される。

【0262】

展開機能は、最近デコードされた圧縮されていないデータの最新の履歴を参照することを含む。復元動作が開始または再開する前に、1つの実施形態では、以下が当てはまる。

- ・新しいタスク(NT)374が1である場合、参照できる初期履歴が存在しない。
- ・NTが0であり、汎用レジスタ0のビット56(HBT)が0(直列)である場合、参照できる初期履歴が、第1のオペランドの左端のバイトの左に隣接して存在し、初期履歴の長さが、パラメータ・ブロックの履歴長(HL)フィールド385によって指定される。
- ・NTが0であり、汎用レジスタ0のビット56(HBT)が1(円形)である場合、参照できる初期履歴が、パラメータ・ブロックの履歴オフセット(HO)386フィールドおよび履歴長(HL)385フィールドによって指定された第3のオペランドの位置に存在する。

20

【0263】

動作中に、動作を実行するために履歴のどのバイトが使用されるかに関わらず、履歴全体へのフェッチタイプの参照が行われてよい。さらに、履歴バッファ・タイプが円形である場合、動作を実行するために履歴のどのバイトが使用されるかに関わらず、(例えば、32Kバイトの)履歴バッファ全体へのフェッチタイプの参照が行われてよい。

30

【0264】

復元動作中に、履歴が更新される。一般オペランド・データ例外条件が発生せずにソース・データをデコードした後に、圧縮されていないデータの得られたバイトが、履歴の末尾に連結される。例えば最大で32Kバイトまでの、圧縮されていないデータの直前にデコードされたバイトが、その後のソース・データの処理中に参照できる最新の履歴を構成する。

【0265】

復元動作の終了時に、1つの例では、その後、動作を再開するため、または別の動作を開始するために使用できる得られた履歴に、以下が当てはまる。

- ・HBTが直列である場合、第1のオペランドの位置に対するストレージの更新が、得られた履歴に対する更新も構成する。更新された第1のオペランド・アドレスおよび更新されたHLは、得られた履歴の更新された位置および更新された長さを指定する。
- ・HBTが円形である場合、履歴が更新されるときに、第3のオペランドの位置に対するストレージの更新が実行される。第3のオペランド・アドレス、更新されたHO、および更新されたHLは、得られた履歴の更新された位置および更新された長さを指定する。

40

【0266】

例として、図16A~16Cは、DFLTCC-XPND機能が指定され、かつ直列履

50

歴が指定されたデフレート変換呼び出し命令の複数の実行の前後で、第1のオペランドに関して、各実行が部分的に完了して終了したときの直列履歴バッファの位置の例を示している。動作中に、履歴長（HL）が385変更される。例えば、図16Aは、DFLTCC-XPNDの実行番号1の前の直列履歴の一例を示しており、図16Bは、DFLTCC-XPNDの実行番号2の前、および実行番号1の後の直列履歴の例を示しており、図16Cは、DFLTCC-XPNDの実行番号2の後の直列履歴の例を示している。図16Cで提供される説明は、図16A～16Bにも当てはまる。

【0267】

汎用レジスタ0のビット56によって指定されたHBTが円形である場合、履歴は、例えば第3のオペランドの位置にある32Kバイトのバッファ内で維持される。バッファ内の履歴の最初のバイトの位置（HB）は、汎用レジスタR3の内容および履歴オフセット（HO）386の和によって指定される。履歴の最初のバイトは、バッファ内の圧縮されていないデータのうちの処理されてから最も長い時間が経過したバイトである。バッファ内の履歴の最後のバイトの位置（HE）は、例えば以下の方程式によって指定される。

【0268】

$$HE = R3 + \text{modulo } 32K (HO + HL - 1)。$$

【0269】

履歴の最後のバイトは、バッファ内の圧縮されていないデータのうちの直前に処理されたバイトである。履歴オフセット（HO）および履歴長（HL）の和が、第3のオペランドのサイズ（例えば、32Kバイト）を超えた場合、履歴は、第3のオペランドの末尾から第3のオペランドの先頭に戻る。本明細書に記載された図15A～15Eは、DFLTCC-XPND機能および円形履歴バッファが指定されたデフレート変換呼び出し命令の複数の実行の前後で、各実行が部分的に完了して終了したときの円形履歴バッファ内の履歴の位置の例を示している。

【0270】

HBTが円形であり、第1のオペランドの位置に格納されたバイト数が例えば32, 768未満である場合、1つの例では以下が当てはまる。

・第3のオペランドの位置のバイトの範囲への格納が行われる。このバイトの範囲は、次式によって指定された位置から開始し、この位置を含む。

【0271】

$$R3 + \text{modulo } 32K (HOO + HLO)$$

ここで、

【0272】

HOO：命令が実行される前の履歴オフセット。

【0273】

HLO：命令が実行される前の履歴長。

【0274】

このバイトの範囲は、例えば次式によって指定された位置で終了し、この位置を含む。

【0275】

$$R3 + \text{modulo } 32K (HOO + HLO + BP - 1)$$

ここで、

【0276】

BP：命令の実行中に第1のオペランドの位置に格納されたバイト数。

【0277】

前述したバイトの範囲に対して行われる格納は、格納タイプ・アクセス例外、PERストレージ変更イベント、および設定変更ビットの対象になる。

・ストレージ位置の内容を変更しない、必要ではない格納が、前述した範囲に含まれない第3のオペランドの位置のバイトに対して行われてよい。そのような位置への格納も、格納タイプ・アクセス例外、PERストレージ変更イベント、および設定変更ビットの対象になる。

10

20

30

40

50

【0278】

HBTが円形であり、第1のオペランドの位置に格納されたバイト数が例えば32, 768以上である場合、格納は、例えば第3のオペランドの位置のすべてのバイトに対して行われ、格納タイプ・アクセス例外、PERストレージ変更イベント、および設定変更ビットの対象になる。

【0279】

BTYPEが2進数00である場合、ブロックは圧縮データを含まない。本明細書に記載された図6は、BTYPEが2進数00に等しいブロックの一例を示している。LENフィールドは、ブロック内のリテラル・バイト数を指定する。LENフィールドのバイト・オーダーはリトルエンディアンである。LENフィールドは、0のリテラル・バイトを指定してよい。ブロックのリテラル・バイトは、第1のオペランドの位置に配置される。前述したように、履歴も、ブロックの各リテラル・バイトで更新される。

【0280】

BTYPEが2進数01である場合、ブロックは、固定ハフマン・テーブル(FHT)を使用して生成された圧縮データ・シンボルを含む。FHTは、デフレート規格によって規定されており、ブロックの一部ではない。本明細書に記載された図7は、BTYPEが2進数01に等しいブロックの一例を示している。ブロック・ヘッダーを解釈した後に、圧縮データ・シンボルが、ブロックに出現する順序でデコードされる。ブロックのバイトは、例えば左から右へ処理され、ブロックの各バイト内のビットは、例えば右から左へ処理される。1つの例では、各シンボルは、ブロック内の次のシンボルを処理する前に、完全に処理される。ブロック終結(EOB)シンボルでない各シンボルは、リテラル値、または履歴バッファ内の以前にデコードされた部分列へのポインタを表す。以前にデコードされた部分列は、重複列とも呼ばれる。1つの例では、重複列の長さの範囲は3~258バイトである。ポインタは、部分列の長さ、および履歴の末尾から部分列の先頭までの距離を表すコードで構成される。シンボルが履歴内の部分列を表している場合、この部分列は履歴バッファから参照される。シンボルのデコーディングから得られた圧縮されていないデータが、第1のオペランドの位置に配置される。

【0281】

ソース・データをさらに処理する前に、前述したように、履歴が更新される。

【0282】

最新の履歴が、ブロックの次のシンボルのデコーディングに適用される。EOBシンボルが出現するときに、ブロックの処理が完了する。

【0283】

BTYPEが2進数10である場合、ブロックは、動的ハフマン・テーブル(DHT)を使用して生成された圧縮データ・シンボルを含む。使用される圧縮形式のDHTは、圧縮データ・ブロックの要素である。本明細書に記載された図8は、BTYPEが2進数10に等しいブロックの一例を示している。ブロック・ヘッダーを解釈した後に、圧縮データ・ブロック内で提供された圧縮形式のDHTが、一般オペランド・データ例外条件に関して検査される。DHTの提供された圧縮形式に関して一般オペランド・データ例外条件が存在する場合、圧縮形式のDHTは無効であると見なされ、データの復元に使用されない。圧縮形式のDHTが、コード長に対するビット長、あるいはリテラル・バイト、EOBシンボル、重複列の長さ、または重複列のポインタ距離に対するコード長を指定し、これらの長さが、適切な機能するハフマン・ツリーを指定するためにハフマン・アルゴリズムによって必要とされる長さよりも大きい場合、機能するDHTを導出してデータを圧縮するために、圧縮されたDHTがまだ使用される。圧縮形式のDHTを検査した後に、圧縮データ・シンボルが、ブロックに出現する順序でデコードされる。ブロックのバイトは、例えば左から右へ処理され、ブロックの各バイト内のビットは、例えば右から左へ処理される。各シンボルは、1つの例では、ブロック内の次のシンボルを処理する前に、完全に処理される。BTYPEが2進数10であるブロック内のシンボルの処理は、BTYPEが01であるブロック内のシンボルの処理に関して前に説明した処理と同じであるが、

10

20

30

40

50

前者がシンボルをデコードするために提供されたDHTを使用し、後者がシンボルをデコードするためにFHTを使用するという点が異なる。圧縮データ・シンボルをデコードするために使用されるハフマン・コードを指定しない非汎用DHTが提供される場合、一般オペランド・データ例外が認識される。

【0284】

第2のオペランドを復元することと一致して、圧縮されていないデータは、チェック値（例えば、32ビットのチェック値）を生成するための入力になる。得られたチェック値は、パラメータ・ブロックのチェック値フィールド387に格納される。

【0285】

データ・セットの最後のブロックを処理した後に、1つの実施形態では、以下が発生する。

- ・モデルに依存する値が、パラメータ・ブロックのモデル・バージョン番号(MVN)フィールド363に格納される。
- ・パラメータ・ブロックのサブバイト境界(SBB)フィールド381が更新される。
- ・汎用レジスタR1内のアドレスが、第1のオペランドの位置に格納されたバイト数だけインクリメントされ、汎用レジスタR1+1内の長さが同じ数だけデクリメントされる。
- ・汎用レジスタR2内のアドレスが、ビット0の処理を含んでいる第2のオペランドの処理されたバイト数だけインクリメントされ、汎用レジスタR2+1内の長さが同じ数だけデクリメントされる。ビット0の処理を含んでいる第2のオペランドの処理されたバイト数は、整数除算から得られた整数化した商であり、被除数は、処理された入力ビット数およびSBBの元の値の和であり、除数は8の値である。

【0286】

アドレスおよび長さの形成および更新は、アドレス指定モードによって決まる。

【0287】

命令の実行の開始時に第1のオペランドの長さが0であるときに、第1のオペランドはアクセスされず、汎用レジスタR1内の第1のオペランド・アドレスおよび汎用レジスタR1+1内の第1のオペランドの長さは変更されない。これは、命令の実行の開始時にCFフィールド373の値が0または1である場合に当てはまる。

【0288】

命令の実行の開始時に第2のオペランドの長さが0であるときに、第2のオペランドはアクセスされず、汎用レジスタR2内の第2のオペランド・アドレスおよび汎用レジスタR2+1内の第2のオペランドの長さは変更されない。1つの実施形態では、以下の場合、命令の実行の開始時に第2のオペランドの長さは0である。

- ・命令が再実行されており（例えば、命令の実行の開始時にパラメータ・ブロックのCFフィールド373が1であり）、命令が前に実行されたときに、第2のオペランド全体が処理された。

【0289】

復元動作は、第2のオペランドの位置からデータが処理されたとしても、結果を第1のオペランドの位置に格納せずに終了してよい。これは、1つの例では、第2のオペランドの位置から処理されたデータが以下の圧縮データ・ブロックの要素のいずれかのみを含んでいる場合に発生する。

- ・ブロック・ヘッダー。
- ・ブロック・タイプが2進数00であるブロックのLENフィールド。
- ・ブロック・タイプが2進数00であるブロックのNLENフィールド。
- ・動的ハフマン・テーブルの圧縮形式。
- ・ブロック終結(EOB)シンボル。

【0290】

1つまたは複数の実施形態では、以下の条件がデフレート変換呼び出し命令の実行に適用される。

【0291】

10

20

30

40

50

1つの例では、DFLTCC-GDHT機能が指定され、以下の条件が発生した場合、一般オペランド・データ例外が認識される。

- ・パラメータ・ブロック・バージョン番号362によって指定されたパラメータ・ブロックの形式が、モデルによってサポートされない。

【0292】

1つの例では、DFLTCC-CMPR機能が指定され、以下の条件のいずれかが発生した場合、一般オペランド・データ例外が認識される。

- ・パラメータ・ブロック・バージョン番号362によって指定されたパラメータ・ブロックの形式が、モデルによってサポートされない。

- ・NT374が0であり、HL385が例えば32，768より大きい。

- ・HTT376が1であり、CDHTL366が、例えば42未満であるか、または例えば2283より大きい。

- ・HTT376が1であり、CDHTL366が、CDHTフィールド367で指定された圧縮形式のDHTの長さに等しくない。

- ・HTT376が1であり、圧縮形式のDHTのHLIT部分要素が例えば29より大きい（無効なDHT）。

- ・HTT376が1であり、圧縮形式のDHTのHDIST部分要素が例えば29より大きい（無効なDHT）。

- ・HTT376が1であり、DHT（CDHTフィールド367の内容）の圧縮形式が、圧縮されたDHTに対して定義された可能な例えば19個のコード長のビット長を指定するコードのシーケンス内にある、機能するハフマン・ツリーを指定するためにハフマン・アルゴリズムによって必要とされる長さより小さいコードを指定する（無効なDHT）。

- ・HTT376が1であり、DHT（CDHTフィールド367の内容）の圧縮形式が、リテラル・バイト、EOBシンボル、および重複列の長さで構成されている要素のセットの第1のコード長として例えばコード長16（前のコード長のコピー）を指定する（無効なDHT）。

- ・HTT376が1であり、DHT（CDHTフィールド367の内容）の圧縮形式が、リテラル・バイトのコード長を指定するコードのシーケンス内にあるコードを指定し、このコードが、圧縮されたDHT内で前に指定された、参照されるコード長のセットを表すために決定されたコードのいずれにも一致しない（無効なDHT）。

- ・HTT376が1であり、DHT（CDHTフィールド367の内容）の圧縮形式が、コード長0（CL0）をEOBシンボルに割り当てるコードを指定する。この場合、対応するDHTが、EOBシンボルを表すためのハフマン・コードを指定しない（無効なDHT）。

- ・HTT376が1であり、DHT（CDHTフィールド367の内容）の圧縮形式が、重複列の長さおよびポインタ距離のコード長を指定するコードのシーケンス内にあるコードを指定し、このコードが、圧縮されたDHT内で前に指定された、参照されるコード長のセットを表すために決定されたコードのいずれにも一致しない（無効なDHT）。

- ・HTT376が1であり、DHT（CDHTフィールド367の内容）の圧縮形式が、HLITフィールド、HDISTフィールド、および例えば258の値の和によって指定された、DHT内のハフマン・コードの数より大きいコード長の数を指定する。これは、例えば、コード長16、17、および18の不適切な使用を伴っている可能性がある（無効なDHT）。

- ・HTT376が1であり、DHT（CDHTフィールド367の内容）の圧縮形式が、リテラル・バイト、EOBシンボル、および重複列の長さのセットに対して、機能するハフマン・ツリーを指定するためにハフマン・アルゴリズムによって必要とされる長さより小さいコード長を指定する（無効なDHT）。

- ・HTT376が1であり、DHT（CDHTフィールド367の内容）の圧縮形式が、重複列のポインタ距離のセットに対して、機能するハフマン・ツリーを指定するためにハフマン・アルゴリズムによって必要とされる長さより小さいコード長を指定する（無効な

10

20

30

40

50

D H T)。

・CPUが、第2のオペランド内のリテラル・バイトを表すために圧縮データ・シンボルを生成しようとし、C D H Tフィールドの内容から導出されたD H Tが非汎用であり、そのリテラル・バイトに対応するハフマン・コードを指定しない。

・CPUが、第2のオペランド内の重複列を表すために圧縮データ・シンボルを生成しようとし、C D H Tフィールドの内容から導出されたD H Tが非汎用であり、その重複列の長さまたはポインタ距離に対応するハフマン・コードを指定しない。

【0293】

例えば、D F L T C C - X P N D機能が指定され、例として以下の条件のいずれかが発生した場合、一般オペランド・データ例外が認識される。

・パラメータ・ブロック・バージョン番号362によって指定されたパラメータ・ブロックの形式が、モデルによってサポートされない。

・N T 3 7 4が0であり、H L 3 8 5が例えば32, 768より大きい。

・B T Y P Eが2進数11に等しい圧縮データ・ブロックが発生する。

・B T Y P Eが2進数00に等しい圧縮データ・ブロック、およびL E Nの1の補数に等しくないN L E Nが発生する。

・圧縮形式のD H T (B T Y P Eが2進数10に等しい圧縮データ・ブロックの内容)が発生し、圧縮されたD H TのH L I T部分要素が例えば29より大きい(無効なD H T)。

・圧縮形式のD H T (B T Y P Eが2進数10に等しい圧縮データ・ブロックの内容)が発生し、圧縮されたD H TのH D I S T部分要素が例えば29より大きい(無効なD H T)。

・圧縮形式のD H T (B T Y P Eが2進数10に等しい圧縮データ・ブロックの内容)が発生し、このD H Tが、圧縮されたD H Tに対して定義された可能な例えば19個のコード長のビット長を指定するコードのシーケンス内にある、機能するハフマン・ツリーを指定するためにハフマン・アルゴリズムによって必要とされる長さより小さいコードを指定する(無効なD H T)。

・圧縮形式のD H T (B T Y P Eが2進数10に等しい圧縮データ・ブロックの内容)が発生し、このD H Tが、リテラル・バイト、E O Bシンボル、および重複列の長さで構成されている要素のセットの第1のコード長として例えばコード長16(前のコード長のコピー)を指定する(無効なD H T)。

・圧縮形式のD H T (B T Y P Eが2進数10に等しい圧縮データ・ブロックの内容)が発生し、このD H Tが、リテラル・バイトのコード長を指定するコードのシーケンス内にあるコードを指定し、このコードが、圧縮されたD H T内で前に指定された、参照されるコード長のセットを表すために決定されたコードのいずれにも一致しない(無効なD H T)。

・圧縮形式のD H T (B T Y P Eが2進数10に等しい圧縮データ・ブロックの内容)が発生し、このD H Tが、コード長0(C L 0)をE O Bシンボルに割り当てるコードを指定する。この場合、対応するD H Tが、E O Bシンボルを表すためのハフマン・コードを指定しない(無効なD H T)。

・圧縮形式のD H T (B T Y P Eが2進数10に等しい圧縮データ・ブロックの内容)が発生し、このD H Tが、重複列の長さおよびポインタ距離のコード長を指定するコードのシーケンス内にあるコードを指定し、このコードが、圧縮されたD H T内で前に指定された、参照されるコード長のセットを表すために決定されたコードのいずれにも一致しない(無効なD H T)。

・圧縮形式のD H T (B T Y P Eが2進数10に等しい圧縮データ・ブロックの内容)が発生し、このD H Tが、H L I Tフィールド、H D I S Tフィールド、および例えば258の値の和によって指定された、D H T内のハフマン・コードの数より大きいコード長の数を指定する。これは、例えば、コード長16、17、および18の不適切な使用を伴っている可能性がある(無効なD H T)。

・圧縮形式のD H T (B T Y P Eが2進数10に等しい圧縮データ・ブロックの内容)が

10

20

30

40

50

発生し、このDHTが、リテラル・バイト、EOBシンボル、および重複列の長さのセットに対して、機能するハフマン・ツリーを指定するためにハフマン・アルゴリズムによって必要とされる長さより小さいコード長を指定する（無効なDHT）。

- ・圧縮形式のDHT（BTYP Eが2進数10に等しい圧縮データ・ブロックの内容）が発生し、このDHTが、重複列のポインタ距離のセットに対して、機能するハフマン・ツリーを指定するためにハフマン・アルゴリズムによって必要とされる長さより小さいコード長を指定する（無効なDHT）。

- ・BTYP Eが2進数10に等しい圧縮データ・ブロックに出現する圧縮データ・シンボルが、同じブロック内の圧縮形式のDHTから導出された汎用DHTによって定義されていないハフマン・コードを指定する。この場合、一般オペランド・データ例外を認識する目的で処理するために使用できる第2のオペランドのビット数は、モデルに依存する。より詳細には、定義されていないコードをデコードしようとするモデルは、より少ないビットを処理した後に例外を決定できたとしても、例外を認識する前に、例えば15ビットを処理してよい。

- ・重複列のポインタであり、シンボルを処理する時点で使用可能な履歴の長さより大きい距離を指定する圧縮データ・シンボルが発生する。

- ・BTYP Eが2進数01に等しい圧縮データ・ブロックに出現する圧縮データ・シンボルが、無効なコード（例えば、重複列の長さの場合は2進数11000110または11000111のコード、あるいは重複列のポインタ距離の場合は2進数11110または11111のコード）を指定する。この場合、一般オペランド・データ例外を認識する目的で処理するために使用できる第2のオペランドのビット数は、モデルに依存する。より詳細には、無効なコードをデコードしようとするモデルは、より少ないビットを処理した後に例外を決定できたとしても、例外を認識する前に、例えば、重複列の長さの場合は8ビット、または重複列のポインタ距離の場合は5ビットを処理してよい。

【0294】

一般オペランド・データ例外が認識された場合、例外に関連する追加情報を提供するよう、パラメータ・ブロックの動作終了補助コード（OESC）フィールド365およびモデル・バージョン番号（MVN）フィールド363が更新されるとしても、動作は抑制されていると見なされる。

【0295】

DFLTCC-CMP R機能またはDFLTCC-XPND機能が実行されており、一般オペランド・データ例外が、第2のオペランドに関して認識されることになっている場合、その結果は、例外が認識されるか、または動作が部分的に完了して終了し、例えば条件コード3が設定されるかのいずれかである。条件コード3が設定された場合、同じオペランドの処理を続行するために命令が再実行され、例外条件がまだ存在するときに、例外が認識される。

【0296】

その他の条件として、例えば以下が挙げられる。

【0297】

命令の実行が割り込み可能である。割り込みが発生した場合、命令が再実行されるときに割り込みの時点で再開するように、汎用レジスタR1およびR2内のアドレス、汎用レジスタR1+1およびR2+1内の長さ、およびパラメータ・ブロックの特定のフィールドが更新される。

【0298】

DFLTCC-CMP R機能またはDFLTCC-XPND機能が実行されており、アクセス例外が、第1または第2のオペランドに関して認識されることになっている場合、その結果は、例外が認識されるか、または動作が部分的に完了して終了し、例えば条件コード3が設定されるかのいずれかである。条件コード3が設定された場合、同じオペランドの処理を続行するために命令が再実行され、例外条件がまだ存在するときに、例外が認識される。

10

20

30

40

50

【0299】

このCPU、他のCPU、およびチャンネル・プログラムによって観察されるとき、パラメータ・ブロック、第1のオペランド、第2のオペランド、および第3のオペランドへの参照は、多重アクセス参照であってよく、これらのストレージ位置へのアクセスは、必ずしもブロック・コンカレントではなく、これらのアクセスまたは参照のシーケンスは定義されない。

【0300】

1つの実施形態では、DFLTCC-CMPR機能またはDFLTCC-XPND機能が指定され、以下のいずれかが当てはまる場合、結果が予測不可能になる。

- ・パラメータ・ブロックが第1のまたは第2のオペランドと重なる。
- ・第1のオペランドが第2のオペランドと重なる。
- ・指定された履歴バッファ・タイプ（HBT）が円形であり、第3のオペランドが、第1のオペランド、第2のオペランド、またはパラメータ・ブロックと重なる。
- ・指定された履歴バッファ・タイプ（HBT）が直列であり、DFLTCC-CMPR機能が指定され、履歴が、第1のオペランドまたはパラメータ・ブロックと重なる。
- ・指定された履歴バッファ・タイプ（HBT）が直列であり、DFLTCC-XPND機能が指定され、履歴が、第2のオペランドまたはパラメータ・ブロックと重なる。

【0301】

特定の状況では、CPUによって決定された処理済みのバイト数が0の状態、デフレート変換呼び出し命令の実行を終了するにもかかわらず、データが第1のオペランドの位置に格納されていてよく、適用可能な場合、データが第3のオペランドの位置に格納されていてよく、適用可能な場合、対応する変更のビットが設定されている。そのような場合、パラメータ・ブロックおよび汎用レジスタの内容は、元の値から変更されていない。このような状況は、CPUが静止動作を実行するか、またはCPUがデフレート変換呼び出し命令の実行中に再試行するときに、発生することがある。

【0302】

デフレート変換呼び出し命令の実行から得られる例示的な条件コードを以下に示す。

【0303】

0 正常完了

【0304】

1 第1のオペランドの長さが、動作を完了するのに不十分である

【0305】

2 第2のオペランドの長さが、動作を完了するのに不十分である（DFLTCC-XPND）

【0306】

3 CPUによって決定された処理済みのデータの量

【0307】

プログラム例外：

- ・アクセス（フェッチ、オペランド2、直列履歴、フェッチおよび格納、パラメータ・ブロック、オペランド1、オペランド3）
- ・DXCOを含むデータ、一般オペランド
- ・演算（デフレート変換機能がインストールされていない場合）
- ・指定
- ・トランザクション制約

【0308】

デフレート変換呼び出し命令の実行の例示的な優先度を以下に示す。

【0309】

1. ～6. 一般的なケースに関する、プログラム割り込み条件の優先度と同じ優先度を持つ例外。

【0310】

7. A 第2の命令のハーフワードに関するアクセス例外。
【0311】
7. B 演算例外。
【0312】
7. C トランザクション制約。
【0313】
8. A 無効な機能コードまたは無効なレジスタ番号に起因する指定例外。
【0314】
8. B 4 Kバイト境界上で指定されないパラメータ・ブロックに起因する指定例外。
【0315】
8. C 4 Kバイト境界上で指定されない円形履歴バッファに起因する指定例外。
【0316】
9. パラメータ・ブロックへのアクセスに関するアクセス例外。
【0317】
10. パラメータ・ブロックの指定された形式がモードによってサポートされていない場合の一般オペランド・データ例外。
【0318】
11. 命令の実行の開始時に第2のオペランドの長さが0に等しく、CFが0に等しいことに起因する指定例外。
【0319】
12. 命令の実行の開始時に第1のオペランドの長さが0に等しく、DFLTCC-CMPRが指定されたことに起因する条件コード1。
【0320】
13. A DFLTCC-CMPRまたはDFLTCC-XPNDが指定された場合に、履歴長フィールドが32, 768より大きく、新しいタスク・フィールドが0であることに起因する一般オペランド・データ例外。
【0321】
13. B 第1のオペランドへのアクセスに関するアクセス例外であり、第1のオペランドの長さが0以外である。
【0322】
13. C 第2のオペランドへのアクセスに関するアクセス例外であり、第2のオペランドの長さが0以外である。
【0323】
13. D 命令の実行の開始時に指定された直列履歴へのアクセスに関するアクセス例外。
【0324】
13. E 第3のオペランドへのアクセスに関するアクセス例外。
【0325】
14. A 上の項目10および13. Aに含まれている条件以外の条件に起因する一般オペランド・データ例外。
【0326】
14. B 上の項目12に含まれている条件以外の条件に起因する条件コード1、2、または3。
【0327】
15. 条件コード0。
【0328】

使用の前に、一般オペランド・データ例外条件の存在に関して、圧縮形式のDHTが検査される。一般オペランド・データ例外条件に起因して圧縮形式のDHTの長さが正確に定義されない場合、解釈される長さは、条件によって決まってよく、モデルに依存してよく、例えば286バイトを超えない。その結果、DFLTCC-XPND機能が指定され、例えば第2のオペランドの右端の286バイトにおいて、一般オペランド・データ例外

10

20

30

40

50

条件を伴う圧縮形式のDHTが発生した場合、例外条件（優先度14．A）または条件コード2（優先度14．B）が認識されるかどうかは、モデルに依存する。

【0329】

例示的なプログラミングに関する注意事項を以下に示す。

1．データを圧縮または復元するときに、デフレート変換呼び出し命令が実行される回数を最小限に抑えて動作が実行された場合に、全体的な効率が高くなることがある。言い換えると、大きいオペランドを使用してDFLTC Cを実行することは、小さいオペランドを使用して複数回DFLTC Cを実行することよりも効率的であることがある。

2．圧縮動作および復元動作で、条件コード3が設定された場合、プログラムが命令に分岐して戻り、動作を続行できるように、命令によって使用される汎用レジスタおよびパラメータ・ブロックが更新されている。

3．1つの実施形態では、命令のパラメータによって指定された処理の、CPUによって決定された下位部分を実行した後に、デフレート変換呼び出し命令が完了してよい。指定されたすべての処理の代わりに、CPUによって決定された量の処理のみを実行した後に命令が完了した場合、命令が条件コード3を設定する。そのような完了時に、命令に分岐して戻り、命令を再実行することによって命令の処理を再開できるように、PSW（program status word：プログラム状態ワード）内の命令アドレスが次の順次命令を指定し、命令のオペランド・パラメータが調整されている。命令は、指定されたすべての処理を実行したときに、3以外の条件コードを設定する。

4．DFLTC C-CMP R機能が指定され、パラメータ・ブロックのサブバイト境界（SBB）フィールド内の0以外の値と共に動作が終了した場合、動作は、得られた第1のオペランド・アドレスによって指定されたバイトに格納することを含んでいた。DFLTC C-XPN D機能が指定され、SBB内の0以外の値と共に動作が終了した場合、動作は、得られた第2のオペランド・アドレスによって指定されたバイトをフェッチすることを含んでいた。

5．0以外の条件コードが設定されて動作が終了した場合、パラメータ・ブロックのCSBフィールド392が、部分的に処理されたデータを含んでよく、プログラムが命令を再実行して動作を再開することが期待される。

6．0以外の条件コードが設定されて動作が終了した後に、動作を再開する目的で命令を再実行する前に、プログラムはパラメータ・ブロックのどのフィールドも変更するべきではなく、そうでない場合、結果が予測不可能になる。

7．DFLTC C-GDHT機能が指定された場合、生成されたDHTの圧縮表現が、ハフマン・アルゴリズムに従って、3つの正しい完全なハフマン・コード・ツリーを表す。すなわち、不完全なハフマン・コード・ツリーは表されない。不完全なハフマン・コード・ツリーは、適切な機能するハフマン・ツリーを指定するためにハフマン・アルゴリズムによって必要とされる長さより大きい要素のコード長を指定するDHTの圧縮表現から導出される。

【0330】

DFLTC C-CMP R機能が指定され、HTTが1であり、DHTの圧縮表現が不完全なハフマン・コード・ツリーの記述を含んでいる場合、DFLTC C-XPN D機能を使用することによって、圧縮データの結果を元の圧縮されていないデータに変換することができるが、デフレート規格を順守する一部のデコーダは、結果を元の圧縮されていないデータに変換できないことがある。これは、例えば、DFLTC C-CMP R機能に対してプログラムによって指定されたDHTの圧縮表現が、DFLTC C-GDHT機能の実行の結果として生成されなかった場合に、発生することがある。

8．条件コード1が設定されてDFLTC C-CMP R機能が終了した場合、パラメータ・ブロックのサブバイト境界（SBB）フィールド381に格納された結果は、2進数000である。このシナリオを認識することは、デフレート変換呼び出し命令で使用するための出力バッファを割り当てるプログラムに関連することがある。

【0331】

本明細書において説明されているように、1つの態様では、汎用プロセッサを使用して圧縮動作または復元動作あるいはその両方を実行するために、単一の命令（例えば、ハードウェア／ソフトウェア・インターフェイスでの単一の設計されたマシン命令（例えば、デフレート変換呼び出し命令））が提供される。例えば、この命令は、命令セット・アーキテクチャ（ISA：Instruction Set Architecture）において定義されたハードウェア命令である。その結果、圧縮動作または復元動作あるいはその両方に関連するプログラムの複雑さが低減される。さらに、動作（したがって、プロセッサ）の性能が改善される。

【0332】

デフレート変換呼び出し命令は、例えばプログラマーによって、I/Oデバイス、I/Oインターフェイスを介して接続されたアプリケーション固有のデバイス、またはその他の種類の専用プロセッサなどの専用プロセッサではなく、有利に、汎用プロセッサ（例えば、本明細書ではプロセッサと呼ばれる中央処理装置）上でディスパッチされる。開示された命令の実行は、ソフトウェアの実装と比較して、同じ動作を実行するために大幅に少ない実行サイクルを必要とする。さらに、開示された命令の実行は、動作をI/Oデバイスにディスパッチすることと比較して、オペレーティング・システムによるI/O操作を必要とせず、動作が完了するのを待っている間に、オペレーティング・システムがタスク切り替えを実行することを引き起こさない。

【0333】

さまざまなフィールドおよびレジスタが説明されたが、本発明の1つまたは複数の態様は、その他の追加の、またはより少ないフィールドまたはレジスタ、あるいはその他のサイズのフィールドおよびレジスタなどを使用してよい。多くの変形が可能である。例えば、命令の明示的に指定されたレジスタまたはフィールドの代わりに、暗黙のレジスタが使用されてよく、または暗黙のレジスタまたはフィールドの代わりに、明示的に指定されたレジスタまたはフィールドが使用されてよく、あるいはその両方であってよい。その他の変形も可能である。

【0334】

デフレート変換呼び出し命令の使用の一実施形態が、図17を参照して説明される。1つの例では、汎用プロセッサなどのプロセッサ上で実行されているプログラムが、ストレージ内のパラメータ・ブロックにおいて、実行される動作の詳細を指定し、パラメータ・ブロックの位置を指定する（ステップ1700）。例えば、実行される機能に応じて、パラメータ・ブロック（例えば、パラメータ・ブロック340、360、または370）のフィールドのうちの1つまたは複数を提供されるか、または設定される。さらに、プログラムは、実行される動作（例えば、照会、生成、圧縮、展開など）を指定する（ステップ1702）。さらに、プログラムは、ストレージ内の位置および入力データの量を指定するか、または更新し（ステップ1704）、ストレージ内の結果バッファの位置およびサイズを指定するか、または更新する（ステップ1706）。

【0335】

その後、プログラムは、デフレート変換呼び出し（DFTCC）命令を実行する（ステップ1708）。1つの例では、汎用プロセッサ上で命令がディスパッチされる。例えば、命令は、汎用プロセッサ上で処理されるか、または汎用プロセッサに結合された、I/Oインターフェイスを使用せずにアクセス可能なハードウェアによって、少なくとも部分的に処理される。

【0336】

命令の終了に基づいて、実行から得られた条件コードが第1の定義された値（例えば、0）に等しいかどうかに関する判定が行われる（照会1710）。条件コードが第1の定義された値に等しい場合、命令の処理が完了する（ステップ1712）。しかし、条件コードが第1の定義された値に等しくない場合、条件コードが第2の定義された値（例えば、3）に等しいかどうかに関する判定がさらに行われる（照会1714）。条件コードが、処理される追加のデータが存在することを示す第2の定義された値に等しい場合、命令が再実行される（ステップ1708）。しかし、条件コードが第2の定義された値に等し

10

20

30

40

50

くない場合、条件コードが第3の定義された値（例えば、1）に設定されているかどうかに関する別の判定が行われる（照会1716）。条件コードが、第1のオペランドの長さが不十分であることを示す第3の定義された値に設定されている場合、処理がステップ1706を続行し、そうでない場合、第2のオペランドの長さが機能のために不十分であり、処理がステップ1704を続行する。

【0337】

示されているように、デフレート変換呼び出し命令は、単一のデータ・ストリームを圧縮または復元するために複数回実行されてよい。したがって、1つの態様では、デフレート変換呼び出し命令が属性を含み、この属性は、デフレート変換呼び出し命令の複数の実行にわたる動作中に処理された圧縮されていないデータの履歴を蓄積するために使用されるバッファ（例えば、32Kバイトのバッファ）を宣言するためのメカニズムをプログラムに提供する。このバッファは、例えば円形履歴バッファである。

10

【0338】

1つの態様では、デフレート変換呼び出し命令は、円形履歴バッファの使用を示すために、暗黙のレジスタ（例えば、GR0.56）内のインジケータ（例えば、ビット）を使用する。円形履歴バッファが示され、デフレート変換呼び出し命令によって実行される指定された機能がデータの圧縮または復元である場合、命令のフィールド（例えば、R3）が、例えば32Kバイトのバッファのメモリ内の位置を指定し、プロセッサが、この位置を使用して、動作の開始時に履歴をフェッチし、動作の終了時に履歴を格納する。円形履歴バッファ内の履歴の長さは、デフレート変換呼び出し命令に関連付けられたパラメータ・ブロックのフィールド（例えば、HLフィールド385）によって指定され、バッファ内の履歴の先頭は、パラメータ・ブロックの別のフィールド（例えば、HOフィールド386）に含まれているオフセットによって指定される。

20

【0339】

円形履歴バッファの使用の詳細が、図18を参照してさらに説明される。1つの例では、汎用プロセッサなどのプロセッサ上で実行されているプログラムが、ストレージ内のパラメータ・ブロックにおいて、実行される動作の詳細を指定し、パラメータ・ブロックの位置を指定する（ステップ1800）。例えば、実行される機能に応じて、パラメータ・ブロック（例えば、パラメータ・ブロック360または370）のフィールドのうちの1つまたは複数が提供されるか、または設定される。さらに、プログラムは、実行される動作（例えば、圧縮、展開など）を指定する。

30

【0340】

さらに、1つの例では、プログラムは、定義済みのサイズ（例えば、32Kバイト）を有する円形バッファを割り当て、そのメモリ内の位置を指定する（ステップ1802）。さらに、プログラムは、圧縮されていないデータ・ストリームの一部をバッファに配置し、このバッファの位置およびサイズをデフレート変換呼び出し命令への入力として指定し（ステップ1804）、ストレージ内の結果バッファの位置およびサイズを指定するか、または更新する（ステップ1806）。

【0341】

次に、デフレート変換呼び出し命令が実行される（ステップ1808）。命令の実行に基づいて、本明細書において説明されているように、プロセッサが、動作への入力として、例えば円形履歴バッファから履歴をフェッチし（ステップ1820）、指定された動作を実行する（ステップ1822）。さらに、プロセッサが、動作の出力として、円形履歴バッファ内の履歴を変更する（ステップ1824）。データ・ストリーム全体が処理されたかどうかに関する判定が行われる（照会1826）。データ・ストリーム全体が処理されていない場合は、処理がステップ1804を続行する。データ・ストリーム全体が処理された場合は、処理が完了する。

40

【0342】

円形履歴バッファを使用することによって、例えば以下が可能になる。

【0343】

50

デフレート変換呼び出し命令の個別の実行で使用するために指定された入力バッファまたは出力バッファのサイズが比較的小さい（例えば、512バイト）場合、バッファされたデータ（例えば、最大32Kバイト）の複数のセグメントにわたる履歴が、少数のバイトを処理するデフレート変換呼び出し命令への入力として使用されてよい。

【0344】

デフレート変換呼び出し命令の個別の実行で使用するために指定された入力バッファまたは出力バッファのサイズが比較的大きい（例えば、128Kバイト）場合、バッファされたデータ（例えば、最大32Kバイト）の以前のセグメントの履歴が、データの最初の32Kバイトを処理しているデフレート変換呼び出し命令への入力として使用されてよい。

【0345】

どちらの場合も、他の方法で利用できるであろう履歴よりも多くの履歴を、データを処理するために使用できる。その結果、重複列の検出の有効性が改善され、全体的な圧縮率の改善が得られる。これによって、コンピューティング環境内の処理を容易にし、性能を改善する。

【0346】

本発明の1つまたは複数の態様は、コンピュータ技術に密接に関係しており、コンピュータ内の処理を容易にし、その性能を改善する。圧縮または復元あるいはその両方を実行するために単一の設計されたマシン命令を使用することによって、コンピューティング環境内の性能を改善する。圧縮／復元されるデータは、データの管理または使用あるいはその両方を行う、コンピュータ処理、医用情報処理、セキュリティ、在庫管理などの多くの技術分野で使用されることがある。圧縮／復元における最適化を実現し、実行時間を削減することによって、これらの技術分野が改善される。

【0347】

本発明の1つまたは複数の態様に関連している1つまたは複数の実施形態の詳細が、図19～22を参照してさらに説明される。

【0348】

次に、本発明の態様に特に関連する技術の概要について説明すると、デフレートは、数ギガバイト（GB）のサイズになることがあるデータの圧縮のためのアルゴリズムであり、このアルゴリズムでは、アプリケーションが一度に小さいバッファのみを有することができ、1メガバイト（MB）以下であることがある比較的小さいブロック内で圧縮が完了する必要がある。デフレートは、通常、ダイレクト・メモリ・アクセス（DMA：direct memory access）に使用されるI/Oインターフェイスに類似する入出力（I/O）インターフェイスを使用してオンチップ・コヒーレンス・サブシステム（例えば、L3キャッシュ）に接続できるアクセラレータまたはNXU上で実行される複合体の命令セットのことを指すことができる。アーキテクチャの観点からは、デフレートは特定の原理原則に従う必要がある。

【0349】

さらに、複合体の割り込み可能命令は、割り込みが使用された後に命令を再開できるようにするために外部に保存する必要がある状態を有する命令（例えば、デフレート、デフレート変換呼び出し命令、アクセラレータ上で実行される複合体の命令セットの命令）である。この状態が、特定の形式設定要件（例えば、パラメータ・ブロック形式）を満たす場合に、再開できるようにするために存在する必要があるすべてのものであり得ることに注意する。特定の形式設定要件のマイクロアーキテクチャ設計は、特に、元のマシン・モデルのために行われ得る。次に、同じ複合体の割り込み可能命令を含む新しいマシン・モデルが導入されるときに、同じ複合体の割り込み可能命令の状態のマイクロアーキテクチャ設計は、特に、この新しいマシン・モデルのために行われる。マイクロアーキテクチャ設計が行われるということは、状態の領域があるマシン・モデルのために作成され、このマシン・モデルによってのみ理解可能であるため、ソフトウェアの観点からは、状態の領域が非透過的であるということの意味する。したがって、新しいマシン・モデルの導入後に、コンピューティング環境は、2つの異なるパラメータ・ブロック形式を有する

10

20

30

40

50

同じ複合体の割り込み可能命令を実行する2つのマシン（例えば、元のマシン・モデルおよび新しいマシン）を含んでいる。

【0350】

複合体の機能およびパラメータ・ブロックを有する命令の場合、本明細書における1つまたは複数の実施形態に従って、新しい仮想アーキテクチャ・レベル（VAL: virtual architecture level）のアプローチが採用される。すなわち、さまざまな複合体のコンピューティング環境内で、VALは、すべての仮想マシンに共通する最低のマシン・レベルに設定された（VAL内の）インジケータを利用することによって、すべての命令が任意の下位のマシン・レベルからのすべての入力をサポートするように、システムの移行を処理するために使用される。したがって、本発明の1つまたは複数の実施形態は、複数のマシン世代にわたって（例えば、コンピューティング環境の元のマシン・モデルと新しいマシンの間で）複合体の機能の互換性を維持することによって、前述した欠点に対処する。さらに、本明細書における1つまたは複数の実施形態は、各マシンが2つの異なるパラメータ・ブロック形式のみを理解することを要求する、世代間の互換性レベルを定義する。これに関して、すべてのマシンは、それ自身のレベルの命令およびパラメータ・ブロックに加えて、最小公分母を実装する。

10

【0351】

図19は、本発明の実施形態に従って、システム1900を示している。システム1900は、パラメータ・ブロック・バージョンn1930を含んでいる互換性レベル1929を有する仮想マシン1920をサポートするマシン・バージョン（n）1910を含んでいる。パラメータ・ブロック・バージョンn1930は、すべてのマシン・バージョンの最小公分母のインジケータ（例えば、最小公分母インジケータ1931）を含んでいる。

20

【0352】

システム1900は、パラメータ・ブロック・バージョンn1930およびパラメータ・ブロック・バージョンn+1 1960を含んでいる互換性レベル1959を有する仮想マシン1950をサポートするマシン・バージョン（n+1）1940も含んでいる。システム1900は、パラメータ・ブロック・バージョンn1930およびパラメータ・ブロック・バージョンn+2 1990を含んでいる互換性レベル1989を有する仮想マシン1980をサポートするマシン・バージョン（n+2）1970も含んでいる。互換性レベル1929、1959、および1989内の最小公分母インジケータ1931は、特定の仮想マシン1920、1950、および1980内でどの機能が使用可能かを識別する一連の機能ビットによって制御される。

30

【0353】

「n」が世代の指定であり、n=0である場合、マシン・バージョン1910が第1の世代のマシンであり、マシン・バージョン1940が第2の世代のマシンであり、マシン・バージョン1970が第3の世代のマシンであるということに注意する。マシン・バージョン1910、1940、および1970は、1つまたは複数の仮想マシン（例えば、仮想マシン1920、1950、および1980）をサポートするハードウェア／ソフトウェアの組み合わせを含んでいる任意のデバイスであることができる。例えば、各マシン・バージョン1910、1940、および1970は、図22のシステム2200の1つのインスタンスと同等であることができる。

40

【0354】

一般に、オペレーティング・システムが仮想マシン1920、1950、および1980間で移行することができるため、すべての仮想マシン1920、1950、および1980は同じVALを有する必要がある。同じVALとは、複合体（例えば、システム1900、図2のCEC200）がマシン・バージョン1910（例えば、第1の世代のマシン）およびマシン・バージョン1940（例えば、第2の世代のマシン）を含んでおり、仮想マシン1950が第1の世代のマシンから第2の世代のマシンに移行されている場合、仮想マシン1950が、移行後に、第1の世代のマシンの命令を実行できるということを意味する。第2の世代のマシンの命令が第1の世代のマシン上で機能しないということ

50

に注意する。第1の世代のマシンの命令および第2の世代のマシンの命令が、第1の世代のマシンおよび第2の世代のマシンの特定の形式設定要件をそれぞれ満たす複合体の割り込み可能命令（例えば、デフレート変換呼び出し命令、またはアクセラレータ上で実行されている複合体の命令セットの命令）であることができるということにも注意する。

【0355】

システム1900は、各仮想マシン1920、1950、および1980のVAL内で、複合体の割り込み可能命令の互換性レベル1929、1959、および1989を維持する。各互換性レベル1929、1959、および1989は、マシン・バージョン1910、1940、および1970全体の最小公分母のマシン・バージョンのため、ならびに複合体の割り込み可能命令のローカルなパラメータ・ブロック形式のために設計される。ローカルなパラメータ・ブロック形式が、マシン・バージョン1910、1940、および1970の各々のローカルなマシン・バージョンのために設計されるということに注意する。これに関して、仮想マシン1920は、マシン・バージョン1910（これが最も古いバージョンである）上で実行できるパラメータ・ブロック形式をサポートし、仮想マシン1940は、マシン・バージョン1910（最小公分母）およびマシン・バージョン1950（ローカル）上で実行できるパラメータ・ブロック形式をサポートし、仮想マシン1970は、マシン・バージョン1910（最小公分母）およびマシン・バージョン1980（ローカル）上で実行できるパラメータ・ブロック形式をサポートする。より詳細には、バージョンnが最小公分母であるため、各仮想マシン1920、1950、および1980上の各互換性レベル1929、1959、および1989内で、パラメータ・ブロック・バージョンn1930が複製される。

【0356】

1つまたは複数の実施形態に従って、対応する複合体の割り込み可能命令を実行するために、マシン・バージョン1910、1940、および1970のマシンによって、互換性レベル1929、1959、および1989が生成され得る。さらに、対応する複合体の割り込み可能命令を最初に実行するために、マシン・バージョン1910、1940、および1970のマシンによって、最小公分母インジケータ1931が生成され得る。VALは、互換性レベル1929を利用して、この最小公分母インジケータ1931を持ち、その後の互換性レベル1959および1989が生成されるときに、パラメータ・ブロック・バージョンn1930が複製されるようにする。最小公分母インジケータ1931は、図に示されているようにパラメータ・ブロック・バージョンn1930に関連付けられるか、またはそれ自身の互換性レベル1929の独立した部分であることができる。1つまたは複数の実施形態に従って、最小公分母インジケータまたは互換性レベルあるいはその両方は、複合体の割り込み可能命令を最初に実行するためのマシン・バージョン1910、1940、および1970のマシンから、残りの複数のマシン・バージョン1910、1940、および1970に伝搬され得る。

【0357】

図20は、本発明の実施形態に従って、プロセス・フロー2000を示している。図21は、本発明の実施形態に従って、システム2100を示している。システム2100は、マシン・バージョンn、n-1、およびn-2（それぞれ、2110、2140、および2170）を有するマシンを最初に関し、これらのマシンの各々は、異なるパラメータ・ブロック形式（それぞれ、パラメータ・ブロック・バージョンn2130、パラメータ・ブロック・バージョンn-1 2160、およびパラメータ・ブロック・バージョンn-2 2190）を実装する。1つまたは複数の実施形態に従って、各パラメータ・ブロック2130、2160、および2190が、マシン・バージョンn、n-1、およびn-2（それぞれ、2110、2140、および2170）内に個別に格納され得る。

【0358】

システム2100によってプロセス・フロー2000が実行され得る。ブロック2010でプロセス・フロー2000が開始し、レベルn-2（2170）でのマシンが、複合体の割り込み可能命令（例えば、デフレート、デフレート変換呼び出し命令、マシン21

70のアクセラレータ上で実行されている複合体の命令セットの命令)を実装するための第1のマシンである。

【0359】

ブロック2020で、システム2100の仮想アーキテクチャ・レベルが、最小公分母インジケータ2191によって、第1のマシンのパラメータ・ブロック(例えば、パラメータ・ブロック・バージョン $n-2$ 2190)が、複合体の割り込み可能命令のために実行/サポートされるということを示す。パラメータ・ブロック・バージョン $n-2$ 2190形式は、互換性レベルまたは発生形式と呼ばれ得る。

【0360】

ブロック2030で、最小公分母インジケータ2191を含むパラメータ・ブロック・バージョン $n-2$ 2190形式が、システム2100内で現在実行されている各仮想マシンに伝搬される。図21に示されているように、矢印Bによって、パラメータ・ブロック・バージョン $n-2$ 2190形式が仮想マシン2120および2150と共有される。ここで、マシン・レベル n および $n-1$ で実行されている仮想マシン2120および2150の両方が、パラメータ・ブロック・バージョン $n-2$ 2190形式を有する複合体の割り込み可能命令を実装することができる。さらに、マシン・レベル n 上で実行されている仮想マシン2120は、パラメータ・ブロック・バージョン n 2130形式に対応する複合体の割り込み可能命令を実装することができ、マシン・レベル $n-1$ 上で実行されている仮想マシン2150は、パラメータ・ブロック・バージョン $n-1$ 2160形式に対応する複合体の割り込み可能命令を実装することができる。

【0361】

ブロック2040で、システム2100が、第1のマシンがオフラインであることを検出する。Xによって図21に示されているように、レベル $n-2$ (2170)でのマシンがオフラインになる。ブロック2050で、システム2100が、新しいマシンがオンラインであることを検出する。例えば、レベル $n+1$ (2191)でのマシンがオンラインになる。ブロック2060で、システム2100が、第1のマシンのパラメータ・ブロックを、マシン・バージョン全体の最小公分母として識別する。

【0362】

ブロック2070で、システム2100が、第1のマシンのパラメータ・ブロックを新しいマシンに伝搬する。図21に示されているように、ここでシステム2100は、パラメータ・ブロック形式(パラメータ・ブロック・バージョン $n+1$ 2193)を実装するマシン・バージョン $n+1$ (2191)を有するマシンを含んでいる。図21に示されているように、矢印Dによって、パラメータ・ブロック・バージョン $n-2$ 2190形式が仮想マシン2192と共有される。したがって、いずれかの仮想マシンが移行されるときに、その仮想マシンは、 $n-2$ レベルの機能を使用する。

【0363】

本明細書における実施形態の技術的効果および利点は、複合体の命令/機能に関して、各マシン世代が、最大で2つの異なるパラメータ・ブロック(それ自身のパラメータ・ブロックおよび互換性レベル)のみをサポートする場合に、検証および妥当性確認が、パラメータ・ブロックの現在のバージョンおよび互換性バージョンに対するテストに減らされることを含む。本明細書における実施形態の技術的効果および利点は、第1の世代の場合、そのパラメータ・ブロック形式が「発生」形式を表し、1つの形式のみがサポート/テストされる必要があるということを含む。したがって、将来のマシン世代に対する下流の影響を引き起こさずに、その後のすべての世代のマシン固有のパラメータ・ブロックの内容が、そのマシンに対して最適化され得る。

【0364】

ここで図22を参照すると、本発明の1つまたは複数の実施形態に従って、本明細書の教示を実装するためのシステム2200が示されている。この実施形態では、システム2200が、1つまたは複数の中央処理装置(CPU)2201a、2201b、2201cなどを含むことができるプロセッサ2201を含んでいる。

【0365】

プロセッサ2201は、処理回路、マイクロプロセッサ、コンピューティング・ユニットとも呼ばれ、システム・バス2202を介してシステム・メモリ2203およびさまざまな他のコンポーネントに結合される。システム・メモリ2203は、読み取り専用メモリ（ROM：read only memory）2204およびランダム・アクセス・メモリ（RAM：random access memory）2205を含む。ROM2204は、システム・バス2202に結合され、システム2200の特定の基本機能を制御する基本入出力システム（BIOS：basic input/output system）を含んでよい。RAMは、プロセッサ2201で使用するためにシステム・バス2202に結合された読み取り書き込みメモリである。

【0366】

図22のシステム2200は、プロセッサ2201による読み取りおよび実行が可能な有形のストレージ媒体の例であるハード・ディスク2207を含んでいる。ハード・ディスク2207は、ソフトウェア2208およびデータ2209を格納する。ソフトウェア2208は、（図21～22のプロセス・フロー2100、2200などのプロセスを実行するために）プロセッサ2201によってシステム2200上で実行される命令として格納される。データ2209は、ソフトウェア2208の動作をサポートし、ソフトウェア2208の動作によって使用されるさまざまなデータ構造に構造化された定性的変数または定量的変数の値のセットを含む。

【0367】

図22のシステム2200は、システム2200のプロセッサ2201、システム・メモリ2203、ハード・ディスク2207、およびその他のコンポーネント（例えば、周辺機器および外部デバイス）の間を相互接続し、これらの間の通信をサポートする1つまたは複数のアダプタ（例えば、ハード・ディスク・コントローラ、ネットワーク・アダプタ、グラフィックス・アダプタなど）を含む。本発明の1つまたは複数の実施形態では、1つまたは複数のアダプタを、中間バス・ブリッジを介してシステム・バス2202に接続された1つまたは複数のI/Oバスに接続することができ、1つまたは複数のI/Oバスが、PCI（Peripheral Component Interconnect）などの一般的なプロトコルを利用することができる。

【0368】

図に示されているように、システム2200は、キーボード2221、マウス2222、スピーカ2223、およびマイクロホン2224をシステム・バス2202に相互接続するインターフェイス・アダプタ2220を含んでいる。システム2200は、システム・バス2202をディスプレイ2231に相互接続するディスプレイ・アダプタ2230を含んでいる。ディスプレイ・アダプタ2230（またはプロセッサ2201あるいはその両方）は、GUI2232の表示および管理などのグラフィックス性能を提供するために、グラフィックス・コントローラを含むことができる。通信アダプタ2241は、システム・バス2202をネットワーク2250と相互接続し、システム2200が、サーバ2251およびデータベース2252などの他のシステム、デバイス、データ、およびソフトウェアと通信できるようにする。本発明の1つまたは複数の実施形態では、ソフトウェア2208およびデータ2209の動作が、サーバ2251およびデータベース2252によってネットワーク2250上に実装され得る。例えば、ネットワーク2250、サーバ2251、およびデータベース2252は、組み合わせさせて、PaaS（Platform as a Service）、SaaS（Software as a Service）、またはIaaS（Infrastructure as a Service）、あるいはその組み合わせとして（例えば、分散システム内のWebアプリケーションとして）、ソフトウェア2208およびデータ2209の内部の反復を提供することができる。

【0369】

したがって、図22で構成されているように、ソフトウェア2208およびデータ2209（例えば、システム2200）の動作は、必然的に、異なるマシン間のワークロードの現在の移行に関する本明細書に記載された欠点を克服し、対処するためのプロセッサ2

10

20

30

40

50

201またはサーバ2251あるいはその両方の計算能力に根差している。これに関して、ソフトウェア2208およびデータ2209は、複数のマシン世代にわたって複合体の機能の互換性を維持することによって、システム2200のプロセッサ2201またはサーバ2251あるいはその両方の計算動作を改善する（それによって、検証および妥当性確認を減らし、下流の影響を引き起こさずに、その後のすべての世代のマシン固有のパラメータ・ブロックの内容を最適化する）。

【0370】

本明細書では、関連する図面を参照して、本発明のさまざまな実施形態が説明される。本発明の範囲を逸脱することなく、本発明の代替の実施形態が考案され得る。以下の説明および図面において、要素間のさまざまな接続および位置関係（例えば、上、下、隣接など）が示される。それらの接続または位置関係あるいはその両方は、特に規定されない限り、直接的または間接的であることができ、本発明はこの点において限定するよう意図されていない。したがって、各実体の結合は、直接的結合または間接的結合を指すことができ、各実体間の位置関係は、直接的位置関係または間接的位置関係であることができる。さらに、本明細書に記載されたさまざまな作業および工程段階は、本明細書に詳細に記載されない追加の段階または機能を含んでいるさらに包括的な手順または工程に組み込まれ得る。

【0371】

以下の定義および略称が、特許請求の範囲および本明細書の解釈に使用される。本明細書において使用されているように、「備える」、「備えている」、「含む」、「含んでいる」、「有する」、「有している」、「含有する」、「含有している」という用語、またはこれらの任意のその他の変形は、非排他的包含をカバーするよう意図されている。例えば、要素のリストを含んでいる組成、混合、工程、方法、製品、または装置は、それらの要素のみに必ずしも限定されず、明示されていないか、またはそのような組成、混合、工程、方法、製品、または装置に固有の、その他の要素を含むことができる。

【0372】

さらに、「例」という用語は、本明細書では「例、事例、または実例としての役割を果たす」ことを意味するために使用される。「例」として本明細書に記載された実施形態または設計は、必ずしも他の実施形態または設計よりも好ましいか、または有利であると解釈されるべきではない。「少なくとも1つ」および「1つまたは複数」という用語は、1以上の任意の整数（すなわち、1、2、3、4など）を含んでいると理解されてよい。「複数」という用語は、2以上の任意の整数（すなわち、2、3、4、5など）を含んでいると理解されてよい。「接続」という用語は、間接的「接続」および直接的「接続」の両方を含んでよい。

【0373】

「約」、「実質的に」、「近似的に」、およびこれらの変形の用語は、本願書の出願時に使用できる機器に基づいて、特定の量の測定に関連付けられた誤差の程度を含むよう意図されている。例えば、「約」は、特定の値の±8%または5%、あるいは2%の範囲を含むことができる。

【0374】

簡潔さの目的で、本発明の態様の作成および使用に関連する従来手法は、本明細書に詳細に記載されることもあれば、記載されないこともある。具体的には、本明細書に記載されたさまざまな技術的特徴を実装するためのコンピューティング・システムおよび特定のコンピュータ・プログラムのさまざまな態様は、よく知られている。したがって、簡略にするために、多くの従来の実装に関する詳細は、本明細書では、既知のシステムまたは工程あるいはその両方の詳細を提供することなく、簡潔にのみ述べられるか、または全体的に省略される。

【0375】

本発明は、任意の可能な統合の技術的詳細レベルで、システム、方法、またはコンピュータ・プログラム製品、あるいはその組み合わせであってよい。コンピュータ・プログラ

10

20

30

40

50

ム製品は、プロセッサに本発明の態様を実行させるためのコンピュータ可読プログラム命令を含んでいるコンピュータ可読ストレージ媒体を含んでよい。

【0376】

コンピュータ可読ストレージ媒体は、命令実行デバイスによって使用するための命令を保持および格納できる有形のデバイスであることができる。コンピュータ可読ストレージ媒体は、例えば、電子ストレージ・デバイス、磁気ストレージ・デバイス、光ストレージ・デバイス、電磁ストレージ・デバイス、半導体ストレージ・デバイス、またはこれらの任意の適切な組み合わせであってよいが、これらに限定されない。コンピュータ可読ストレージ媒体のさらに具体的な例の非網羅的リストは、ポータブル・フロッピー（R）・ディスク、ハード・ディスク、ランダム・アクセス・メモリ（RAM：random access memory）、読み取り専用メモリ（ROM：read-only memory）、消去可能プログラマブル読み取り専用メモリ（EPROM：erasable programmable read-only memoryまたはフラッシュ・メモリ）、スタティック・ランダム・アクセス・メモリ（SRAM：static random access memory）、ポータブル・コンパクト・ディスク読み取り専用メモリ（CD-ROM：compact disc read-only memory）、デジタル多用途ディスク（DVD：digital versatile disk）、メモリ・スティック、フロッピー（R）・ディスク、パンチカードまたは命令が記録されている溝の中の隆起構造などの機械的にエンコードされるデバイス、およびこれらの任意の適切な組み合わせを含む。本明細書において使用されるとき、コンピュータ可読ストレージ媒体は、それ自体が、電波またはその他の自由に伝搬する電磁波、導波管またはその他の送信媒体を伝搬する電磁波（例えば、光ファイバ・ケーブルを通過する光パルス）、あるいはワイヤを介して送信される電気信号などの一過性の信号であると解釈されるべきではない。

【0377】

本明細書に記載されたコンピュータ可読プログラム命令は、コンピュータ可読ストレージ媒体から各コンピューティング・デバイス／処理デバイスへ、またはネットワーク（例えば、インターネット、ローカル・エリア・ネットワーク、広域ネットワーク、または無線ネットワーク、あるいはその組み合わせ）を介して外部コンピュータまたは外部ストレージ・デバイスへダウンロードされ得る。このネットワークは、銅伝送ケーブル、光伝送ファイバ、無線送信、ルータ、ファイアウォール、スイッチ、ゲートウェイ・コンピュータ、またはエッジ・サーバ、あるいはその組み合わせを備えてよい。各コンピューティング・デバイス／処理デバイス内のネットワーク・アダプタ・カードまたはネットワーク・インターフェイスは、コンピュータ可読プログラム命令をネットワークから受信し、それらのコンピュータ可読プログラム命令を各コンピューティング・デバイス／処理デバイス内のコンピュータ可読ストレージ媒体に格納するために転送する。

【0378】

本発明の動作を実行するためのコンピュータ可読プログラム命令は、アセンブラ命令、命令セット・アーキテクチャ（ISA：instruction-set-architecture）命令、マシン命令、マシン依存命令、マイクロコード、ファームウェア命令、状態設定データ、集積回路のための構成データ、あるいは、Smalltalk（R）、C++などのオブジェクト指向プログラミング言語、および「C」プログラミング言語または同様のプログラミング言語などの手続き型プログラミング言語を含む1つまたは複数のプログラミング言語の任意の組み合わせで記述されたソース・コードまたはオブジェクト・コードであってよい。コンピュータ可読プログラム命令は、ユーザのコンピュータ上で全体的に実行すること、ユーザのコンピュータ上でスタンドアロン・ソフトウェア・パッケージとして部分的に実行すること、ユーザのコンピュータ上およびリモート・コンピュータ上でそれぞれ部分的に実行すること、あるいはリモート・コンピュータ上またはサーバ上で全体的に実行することができる。後者のシナリオでは、リモート・コンピュータは、ローカル・エリア・ネットワーク（LAN：local area network）または広域ネットワーク（WAN：wide area network）を含む任意の種類のネットワークを介してユーザのコンピュータに接続されてよく、または接続は、（例えば、インターネット・サービス・プロバイダを使用して

インターネットを介して) 外部コンピュータに対して行われてよい。一部の実施形態では、本発明の態様を実行するために、例えばプログラマブル論理回路、フィールドプログラマブル・ゲート・アレイ (FPGA: field-programmable gate arrays)、またはプログラマブル・ロジック・アレイ (PLA: programmable logic arrays) を含む電子回路は、コンピュータ可読プログラム命令の状態情報を利用することによって、電子回路をカスタマイズするためのコンピュータ可読プログラム命令を実行してよい。

【0379】

本発明の態様は、本明細書において、本発明の実施形態に従って、方法、装置 (システム)、およびコンピュータ・プログラム製品のフローチャート図またはブロック図あるいはその両方を参照して説明される。フローチャート図またはブロック図あるいはその両方の各ブロック、ならびにフローチャート図またはブロック図あるいはその両方に含まれるブロックの組み合わせが、コンピュータ可読プログラム命令によって実装され得るということが理解されるであろう。

10

【0380】

これらのコンピュータ可読プログラム命令は、コンピュータまたはその他のプログラム可能なデータ処理装置のプロセッサを介して実行される命令が、フローチャートまたはブロック図あるいはその両方のブロックに指定される機能/動作を実施する手段を作り出すべく、汎用コンピュータ、専用コンピュータ、または他のプログラム可能なデータ処理装置のプロセッサに提供されてマシンを作り出すものであってよい。これらのコンピュータ可読プログラム命令は、命令が格納されたコンピュータ可読ストレージ媒体がフローチャートまたはブロック図あるいはその両方のブロックに指定される機能/動作の態様を実施する命令を含んでいる製品を備えるように、コンピュータ可読ストレージ媒体に格納され、コンピュータ、プログラム可能なデータ処理装置、または他のデバイス、あるいはその組み合わせに特定の方式で機能するように指示できるものであってもよい。

20

【0381】

コンピュータ可読プログラム命令は、コンピュータ上、その他のプログラム可能な装置上、またはその他のデバイス上で実行される命令が、フローチャートまたはブロック図あるいはその両方のブロックに指定される機能/動作を実施するように、コンピュータ、その他のプログラム可能なデータ処理装置、またはその他のデバイスに読み込まれてもよく、それによって、一連の動作可能なステップを、コンピュータ上、その他のプログラム可能な装置上、またはコンピュータ実装プロセスを生成するその他のデバイス上で実行させる。

30

【0382】

図内のフローチャートおよびブロック図は、本発明のさまざまな実施形態に従って、システム、方法、およびコンピュータ・プログラム製品の可能な実装のアーキテクチャ、機能、および動作を示す。これに関連して、フローチャートまたはブロック図内の各ブロックは、規定された論理機能を実装するための1つまたは複数の実行可能な命令を備える、命令のモジュール、セグメント、または部分を表してよい。一部の代替の実装では、ブロックに示された機能は、図に示された順序とは異なる順序で発生してよい。例えば、連続して示された2つのブロックは、実際には、含まれている機能に応じて、実質的に同時に実行されるか、または場合によっては逆の順序で実行されてよい。ブロック図またはフローチャート図あるいはその両方の各ブロック、ならびにブロック図またはフローチャート図あるいはその両方に含まれるブロックの組み合わせは、規定された機能または動作を実行するか、または専用ハードウェアとコンピュータ命令の組み合わせを実行する専用ハードウェアベースのシステムによって実装され得るということにも注意する。

40

【0383】

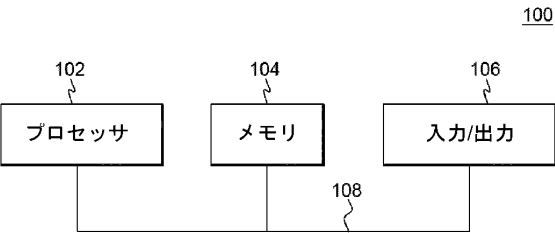
本発明のさまざまな実施形態の説明は、例示の目的で提示されているが、網羅的であることは意図されておらず、開示された実施形態に制限されない。説明された実施形態の趣旨から逸脱することなく多くの変更および変形が可能であることは、当業者にとって明らかであろう。本明細書で使用された用語は、実施形態の原理、実際の適用、または市場で

50

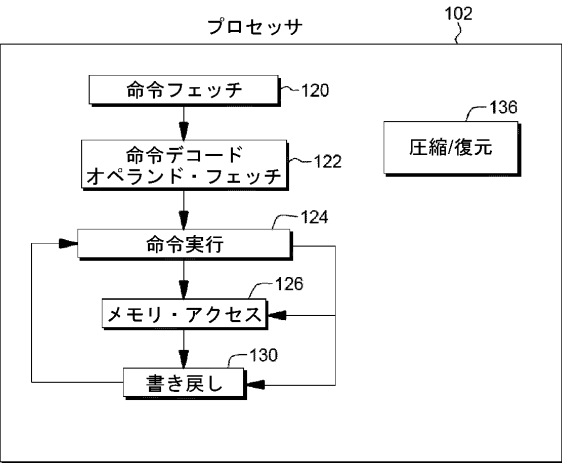
見られる技術を超える技術的改良を最も適切に説明するため、または他の当業者が本明細書に記載された実施形態を理解できるようにするために選択されている。

【図面】

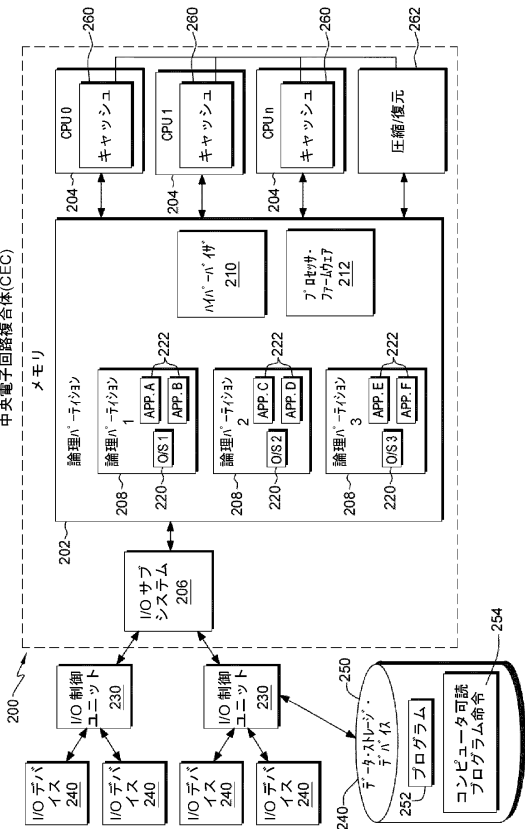
【図 1 A】



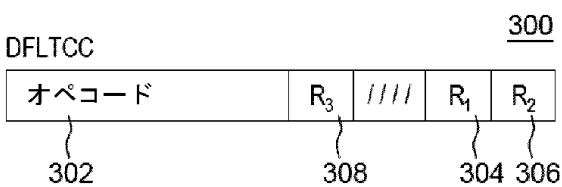
【図 1 B】



【図 2】



【図 3 A】



10

20

30

40

50

【図 3 B】



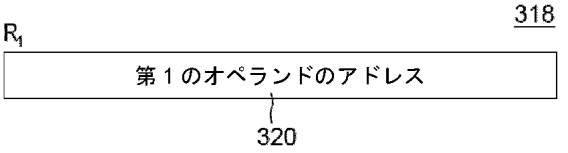
【図 3 C】

	コード	機能	パラメータ・ブロックの サイズ(バイト)
313~	0	DFLTCC - QAF	32
315~	1	DFLTCC - GDHT	384
317~	2	DFLTCC - CMPR	1536
319~	4	DFLTCC - XPND	1536

【図 3 D】



【図 3 E】



【図 3 F】



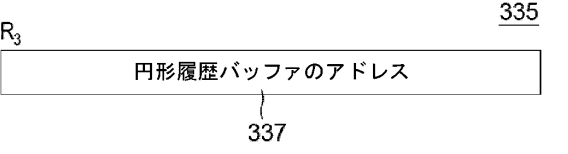
【図 3 G】



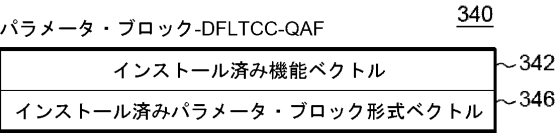
【図 3 H】



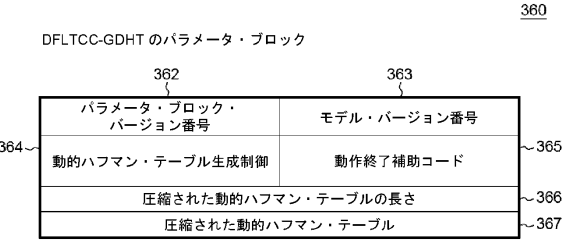
【図 3 I】



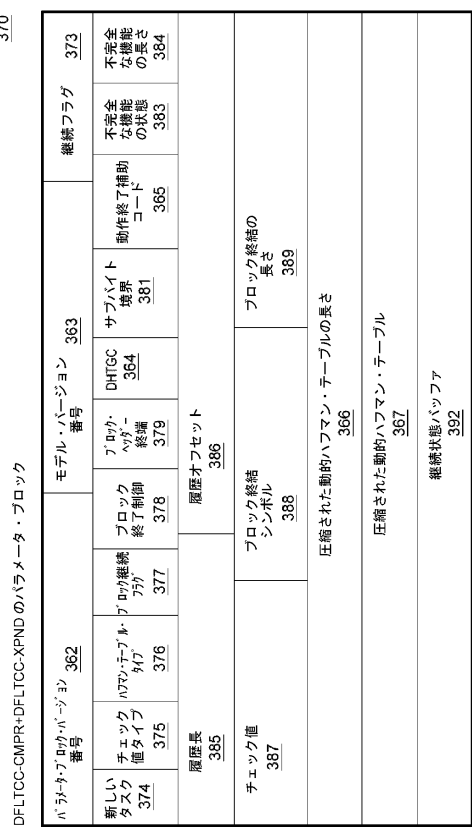
【図 3 J】



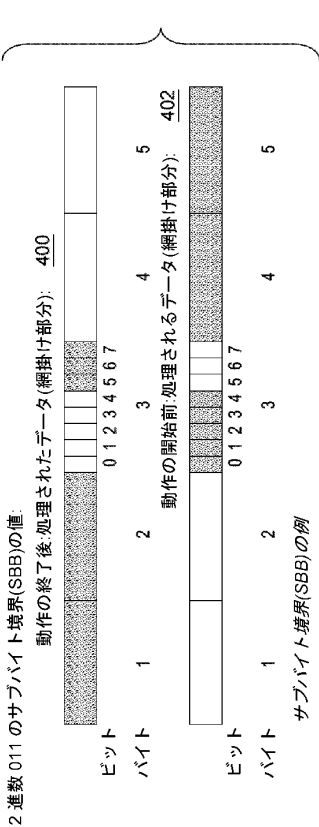
【図 3 K】



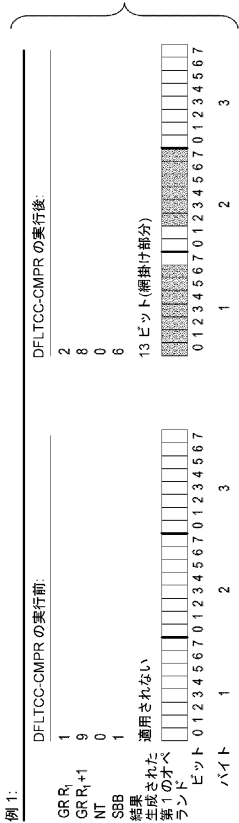
【図 3 L】



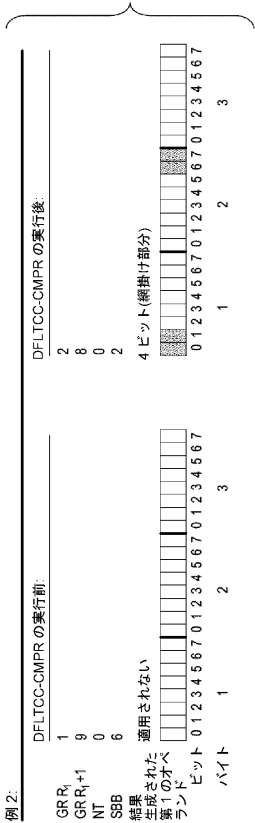
【図 4】



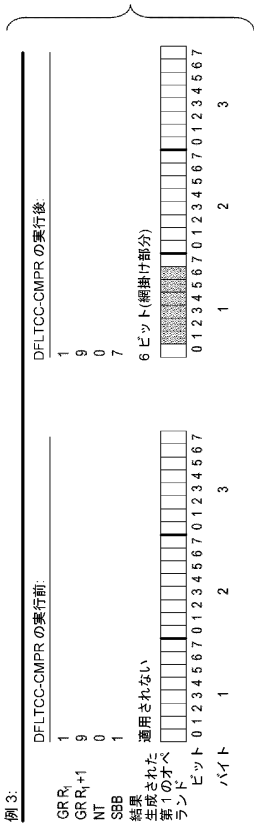
【図 5 A】



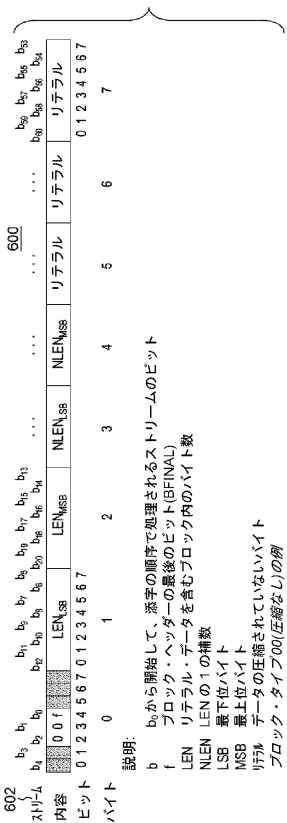
【図 5 B】



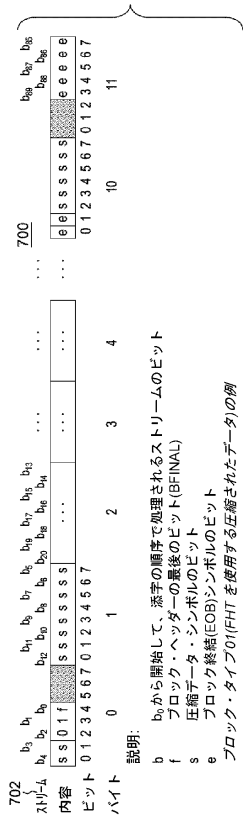
【図 5 C】



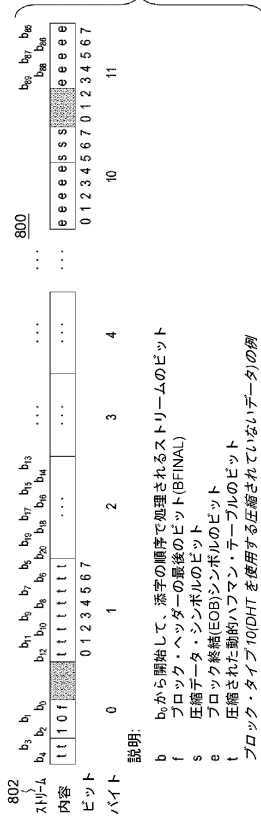
【図 6】



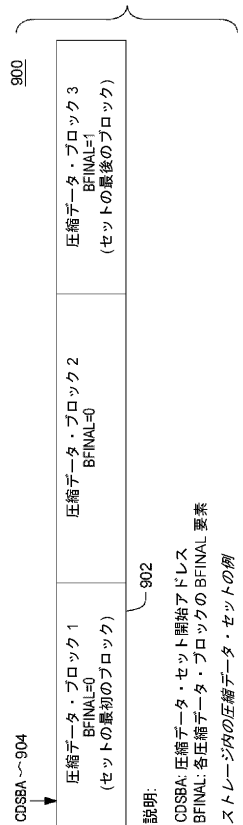
【図 7】



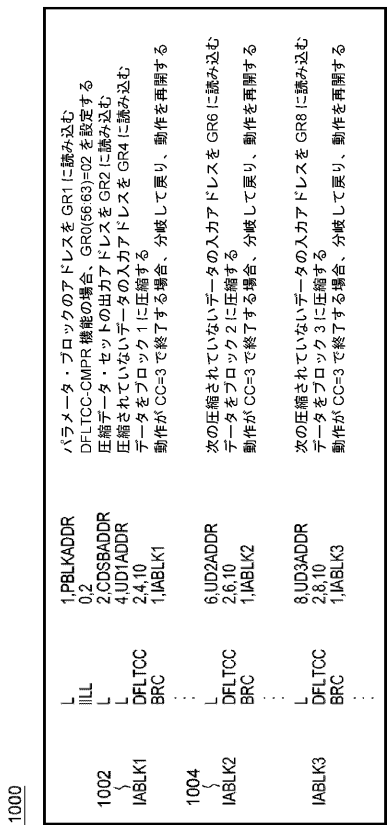
【図 8】



【図 9】



【図 10】



データを 3 つのブロックの圧縮データ・セットに圧縮するプログラムのサンプル

【図 1 3】

L IILL L L DFTCC BRC : IASET	パラメータ・ブロックのアドレスを GR1 に読み込む DFTCC-XPND 機能の場合、GR0(5663)=04 を設定する 圧縮されていないデータの出力アドレスを GR2 に読み込む 圧縮データ・セットの入力アドレスを GR4 に読み込む データ・セットのすべてのブロックを復元する 動作が CC=3 で終了した場合、分岐して戻り、動作を再開する
	1.PBLKADDR 04 2.UDADDR 4.CDSBADDR 24.10 1.IASET

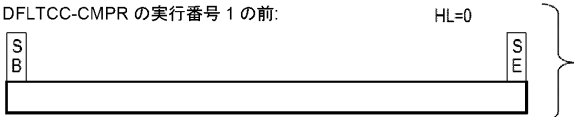
圧縮データ・セットからデータを復元するプログラムのサンプル

【図 1 1】

パラメータ・ブロックのフィールド 1100	動作の開始時のフィールド 値 ~1102	条件コード 1, 2, ~1104 または 3 が設定された場合 の動作の終了時のフィールド 値 (動作の再開の開始時の値 でもある)	条件コード 0 が設定され た場合の動作の終了時の フィールド 値 ~1106
新しいタスク(NT)	1(7 ログラムによって指定される) 0		0
継続フラグ(CF)	プログラムによって 0 に 初期化される	1	0
継続状態バッファ(CSB)	プログラムによってすべて 0 に初期化される	動作の終了時に記録される 内部状態データであり、動 作が後で再開するときには必 要とされる	未定義
チェック値	NT が 1 であるため、 初期値として扱われる	動作の開始以降に処理され たデータの部分から生成さ れる値であり、動作を再開 するための開始値である	動作によって処理された すべてのデータから生成さ れる値であり、その後 の動作のための開始値で ある
サブバイト境界(SBB)	NT が 1 であるため、0 に 等しいとして扱われる	動作によって処理された最 後のバイト、および後で動 作が再開するときの最初の バイトに適用される	動作によって処理された 最後のバイト、およびそ の後の動作のための最初 のバイトに適用される

セットの第 1 の圧縮データ・ブロックに対して動作する DFLTCC-CMPR 機能のパラメータ・ブロックの内容

【図 1 4 A】

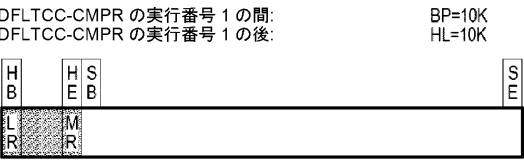


【図 1 2】

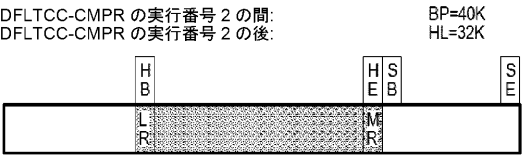
パラメータ・ブロックのフィールド 1100	動作の開始時のフィー ルド値	条件コード 1, 2, または 3 が設定された場合の動作の終 了時のフィールド値(動作の 再開の開始時の値でもある)	条件コード 0 が設定さ れた場合の動作の終了時 のフィールド値
新しいタスク(NT)	0(7 ログラムによって指定される) 0		0
継続フラグ(CF)	プログラムによって 0 に 初期化される	1	0
継続状態バッファ(CSB)	プログラムによってすべて 0 に初期化される	動作の終了時に記録される 内部状態データであり、動 作が後で再開するときには必 要とされる	未定義
チェック値	前の動作から生成され る値(プログラムによ って保存される)	現在の動作の開始以降に処 理されたデータの部分から 生成される値であり、現在 の動作を再開するための開 始値である	動作によって処理され たすべてのデータから生成 される値であり、その後 の動作のための開始値で ある
サブバイト境界(SBB)	前の動作によって処理さ れた最後のバイトに関連 付けられた値(プログラム によって保存される)	現在の動作によって処理さ れた最後のバイト、および 後で現在の動作が再開する ときの最初のバイトに適用 される	動作によって処理された 最後のバイト、およびそ の後の動作のための最初 のバイトに適用される

セットの第 2 の圧縮データ・ブロックに対して動作する DFLTCC-CMPR 機能のパラメータ・ブロックの内容

【図 1 4 B】



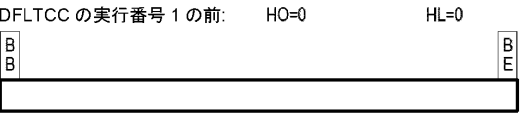
【図 1 4 C】



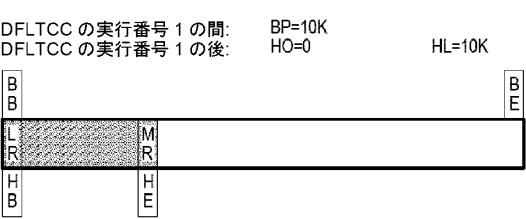
説明:
網掛け部分:履歴のバイト
SB ソース開始アドレス:R₂によって指定された位置
SE ソース終了アドレス:R₂+(R₂+1)-1によって指定された位置
HB 履歴開始アドレス:R₂-HLによって指定された位置
HE 履歴終了アドレス:R₂-1によって指定された位置
LR 履歴内の処理されてから最も長い時間が経過したバイト
MR 履歴内の直前に処理されたバイト
BP 処理された圧縮されていないデータのバイト数
HL 履歴長
DFLTCC-CMPR を複数回実行する前後の直列履歴

10

【図 1 5 A】

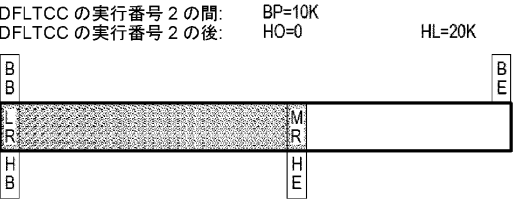


【図 1 5 B】

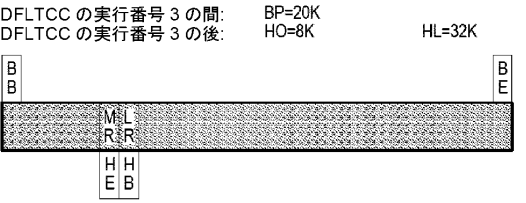


20

【図 1 5 C】



【図 1 5 D】

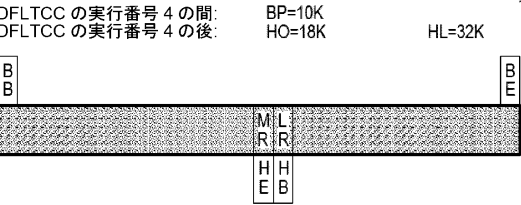


30

40

50

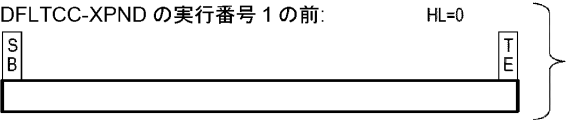
【図 1 5 E】



説明:

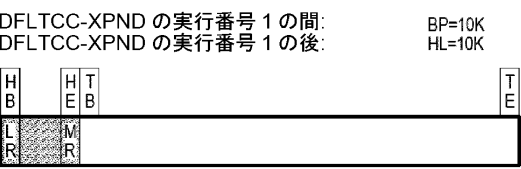
網掛け部分:円形履歴バッファ内の履歴のバイト
BB バッファ開始アドレス:R₃によって指定された位置
BE バッファ終了アドレス:R₃+32K-1によって指定された位置
HB 履歴開始アドレス:R₃+HOによって指定された位置
HE 履歴終了アドレス:R₃+modulo32K(HO+HL-1)によって指定された位置
LR 履歴内の処理されてから最も長い時間が経過したバイト
MR 履歴内の直前に処理されたバイト
BP 処理された圧縮されていないデータのバイト数
HO 履歴オフセット
HL 履歴長
DFLTCC を複数回実行する前後の円形履歴バッファ

【図 1 6 A】

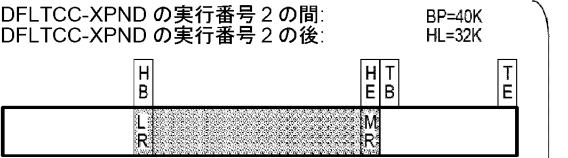


10

【図 1 6 B】



【図 1 6 C】



説明:

網掛け部分:履歴のバイト
TB ターゲット開始アドレス:R_iによって指定された位置
TE ターゲット終了アドレス:R_i+(R_i+1)-1によって指定された位置
HB 履歴開始アドレス:R_i-HLによって指定された位置
HE 履歴終了アドレス:R_i-1によって指定された位置
LR 履歴内の処理されてから最も長い時間が経過したバイト
MR 履歴内の直前に処理されたバイト
BP 処理された圧縮されていないデータのバイト数
HL 履歴長
DFLTCC-XPND を複数回実行する前後の直列履歴

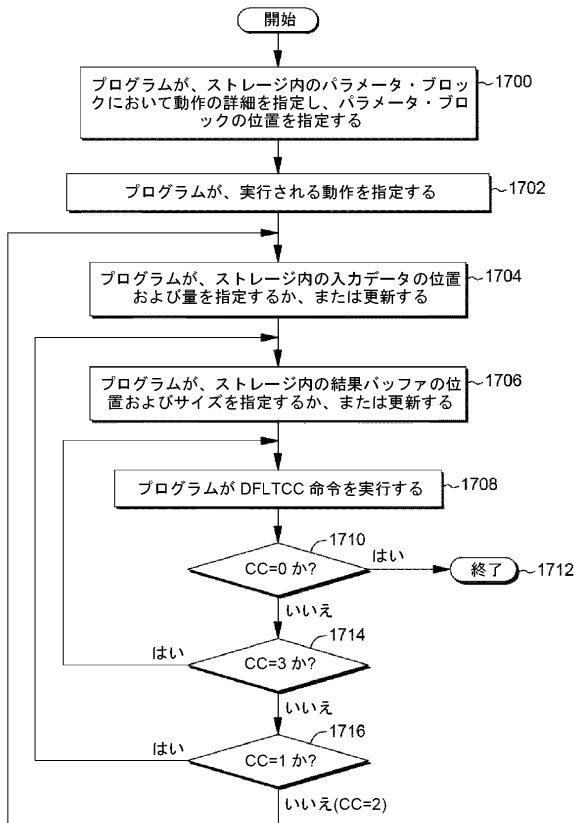
20

30

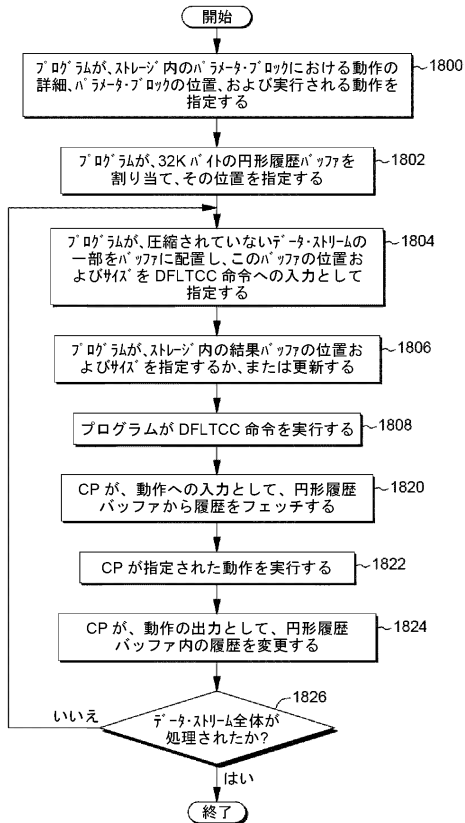
40

50

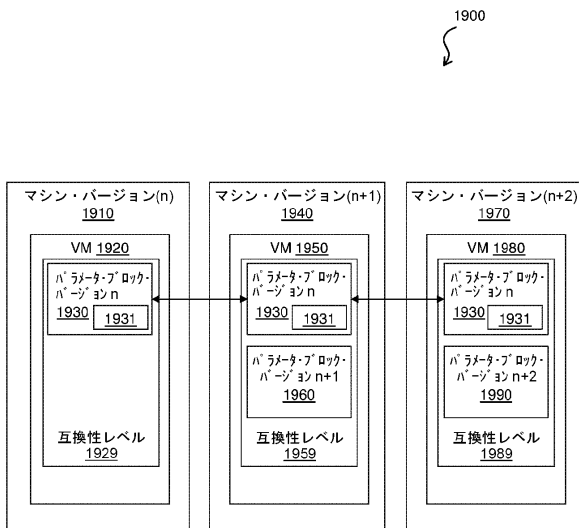
【図 17】



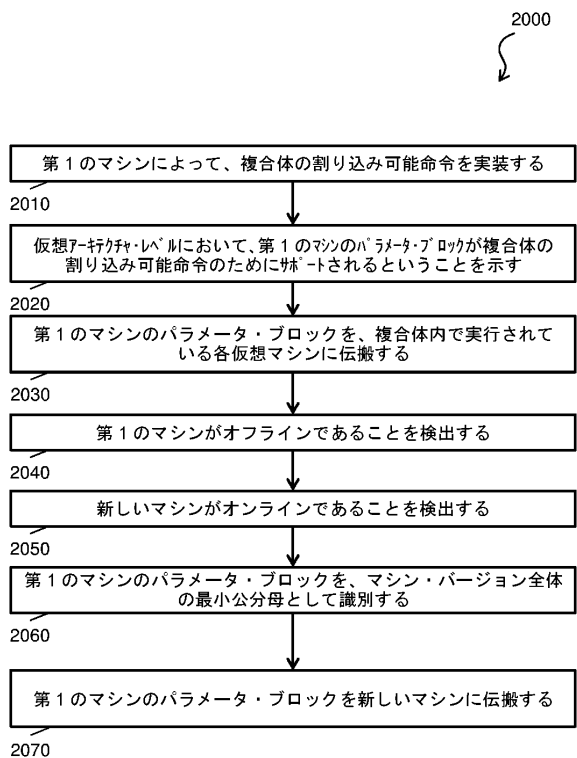
【図 18】



【図 19】



【図 20】



10

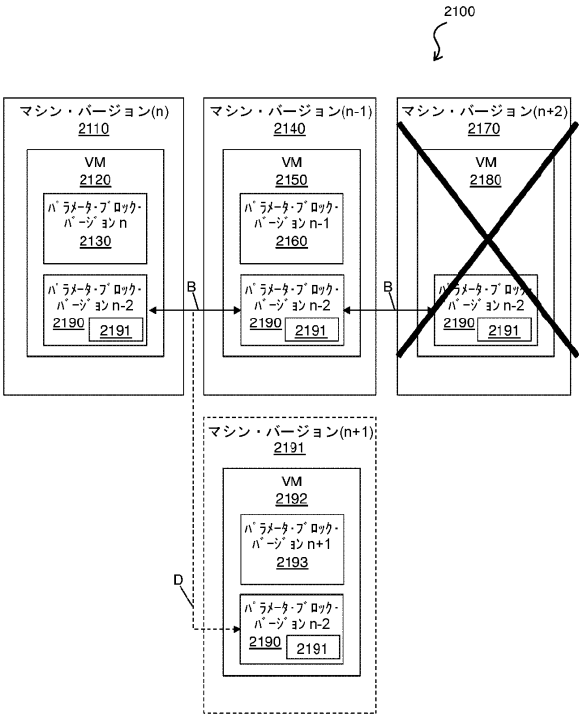
20

30

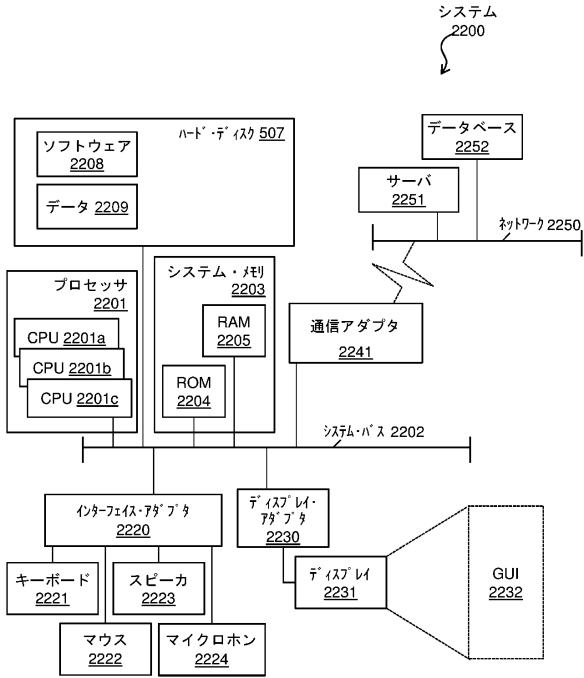
40

50

【図 2 1】



【図 2 2】



10

20

30

40

50

フロントページの続き

- (72)発明者 クライン、マティアス
 アメリカ合衆国12601-5400 ニューヨーク州ボキプシー サウス・ロード2455
- (72)発明者 ギアメイ、ブルース、コンラート
 アメリカ合衆国12601-5400 ニューヨーク州ボキプシー サウス・ロード2455
- (72)発明者 ソフィア、アントニー、トーマス
 アメリカ合衆国12601-5400 ニューヨーク州ボキプシー サウス・ロード2455
- (72)発明者 ファレル、マーク
 アメリカ合衆国12601-5400 ニューヨーク州ボキプシー サウス・ロード2455
- (72)発明者 スワニー、スコット
 アメリカ合衆国12601-5400 ニューヨーク州ボキプシー サウス・ロード2455
- (72)発明者 スリゲル、ティモシー
 アメリカ合衆国12601-5400 ニューヨーク州ボキプシー サウス・ロード2455
- 審査官 漆原 孝治
- (56)参考文献 特表2013-534668 (JP, A)
 特表平08-509830 (JP, A)
 特開2014-182810 (JP, A)
- (58)調査した分野 (Int.Cl., DB名)
 G06F 9/50
 G06F 9/30
 G06F 9/38
 G06F 9/455