(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7492830号
(P7492830)

(45)発行日　令和6年5月30日(2024.5.30)

(24)登録日　令和6年5月22日(2024.5.22)

(51)国際特許分類

		FI		
Ｇ０６Ｆ	*9/308(2018.01)*	Ｇ０６Ｆ	9/308	Ｂ
Ｇ０６Ｆ	*9/345(2018.01)*	Ｇ０６Ｆ	9/345	Ａ

請求項の数　1　（全13頁）

(21)出願番号　特願2019-572555(P2019-572555)
(86)(22)出願日　平成30年6月27日(2018.6.27)
(65)公表番号　特表2020-526825(P2020-526825A)
(43)公表日　令和2年8月31日(2020.8.31)
(86)国際出願番号　PCT/EP2018/067234
(87)国際公開番号　WO2019/011653
(87)国際公開日　平成31年1月17日(2019.1.17)
審査請求日　令和3年6月18日(2021.6.18)
審判番号　不服2023-4398(P2023-4398/J1)
審判請求日　令和5年3月15日(2023.3.15)
(31)優先権主張番号　17386023.0
(32)優先日　平成29年7月10日(2017.7.10)
(33)優先権主張国・地域又は機関
欧州特許庁(EP)

(73)特許権者　500395107
アーム・リミテッド
グレート・ブリテン及び北部アイルランド連合王国、イングランド、 シー・ビー１、９エヌ・ジェイ、ケンブリッジ、フルボーン・ロード １１０
(74)代理人　110000855
弁理士法人浅村特許事務所
(72)発明者　マグクリス、グリゴリオス
英国、ケンブリッジシャー、ケンブリッジ、チェリー ヒントン、フルボーン ロード １１０、エイアールエム リミテッド 気付
(72)発明者　スティーブンス、ナイジェル ジョン
英国、ケンブリッジシャー、ケンブリッ
最終頁に続く

(54)【発明の名称】　ベクトル要素内のビット値のテスト

(57)【特許請求の範囲】

【請求項１】

命令をデコードし、前記命令に応じた制御信号を生成する命令デコード回路と、

前記命令デコード回路によって生成された前記制御信号に応えてデータ処理動作を実行するデータ処理回路と、

を備える装置であって、

前記命令デコード回路は、

ソースベクトルレジスタおよびインデックスを指定するビットテスト命令に応答し、前記データ処理回路が複数の結果ビットを生成するためにソースベクトルレジスタ内に格納された複数の要素に対してビットテスト手順を実行させるようにし、

ここで、前記ビットテスト手順は、前記複数の要素の各要素について、前記インデックスによって示される前記ソースベクトルレジスタの前記要素内のビット位置のテストビットの値に応じて前記複数の結果ビットのそれぞれの結果ビットを設定することを含み、

前記ビットテスト手順では、前記複数の要素のうち、支配述語ビット値のセットのうちの第１所定値を有する述語ビットに対応する要素が前記ビットテスト手順のビットテストを受け、

前記ビットテスト手順では、前記複数の要素のうち、前記支配述語ビット値のセットのうちの第１所定値を有さない述語ビットに対応する要素に対する結果ビットが第２所定値に設定される、

装置。

【発明の詳細な説明】
【技術分野】
【０００１】
本開示は、データ処理装置に関する。より具体的には、データ処理装置によって処理されたベクトル要素内のビット値のテストに関する。
【背景技術】
【０００２】
データ処理装置がデータ処理を実行しているとき、入力値の指定されたビットが設定されているか否かをデータ処理装置がテストすることは有用であり得る。このように、たとえば特定の機能をオンまたはオフに切り替えるように、特定のデータ処理挙動を修正するように、処理のために特定の指定された入力データを包含または除外するようになど、データ処理装置の動作を制御するために、入力値の当該ビットはプログラマによって使用され得る。
【発明の概要】
【０００３】
例示的な一実施形態では、命令をデコードし、命令に応じた制御信号を生成する命令デコード回路と、命令デコード回路によって生成された制御信号に応えてデータ処理動作を実行するデータ処理回路とを備える装置であって、命令デコード回路は、複数の結果ビットを生成するためにソースベクトルレジスタ内に格納された複数の要素に対してデータ処理回路にビットテスト手順を実行させるため、ソースベクトルレジスタおよびインデックスを指定するビットテスト命令に応答し、ビットテスト手順は、複数の要素の各処理済み要素について、インデックスによって示されるソースベクトルレジスタの処理済み要素内のビット位置のテストビットの値に応じて複数の結果ビットのそれぞれの結果ビットを設定することを含む、装置がある。
【０００４】
別の例示的実施形態では、データ処理装置を動作させる方法であって、命令をデコードし、命令に応じた制御信号を生成するステップと、生成された制御信号に応えてデータ処理動作を実行するステップと、ソースベクトルレジスタおよびインデックスを指定するビットテスト命令に応えて、複数の結果ビットを生成するためにソースベクトルレジスタ内に格納された複数の要素に対するビットテスト手順の実行を行わせるステップと、を備え、ビットテスト手順は、複数の要素の各処理済み要素について、インデックスによって示されるソースベクトルレジスタの処理済み要素内のビット位置のテストビットの値に応じて複数の結果ビットのそれぞれの結果ビットを設定することを備える、方法がある。
【０００５】
別の例示的実施形態では、命令をデコードし、命令に応じた制御信号を生成する手段と、生成された制御信号に応えてデータ処理動作を実行する手段と、ソースベクトルレジスタおよびインデックスを指定するビットテスト命令に応えて、複数の結果ビットを生成するためにソースベクトルレジスタ内に格納された複数の要素に対するビットテスト手順の実行を行わせる手段と、を備える装置であって、ビットテスト手順は、複数の要素の各処理済み要素について、インデックスによって示されるソースベクトルレジスタの処理済み要素内のビット位置のテストビットの値に応じて複数の結果ビットのそれぞれの結果ビットを設定することを備える、装置がある。
【０００６】
別の例示的実施形態では、命令をデコードし、命令に応じた制御信号を生成するための命令デコードプログラムロジックと、命令デコードプログラムロジックによって生成された制御信号に応えてデータ処理動作を実行するためのデータ処理プログラムロジックと、を備える命令実行環境を提供するようにホストデータ処理装置を制御するためのコンピュータプログラムであって、命令デコードプログラムロジックは、データ処理プログラムロジックに、複数の結果ビットを生成するためにソースベクトルデータ構造内に格納された複数の要素に対してビットテスト手順を実行させるため、ソースベクトルデータ構造およ

びインデックスを指定するビットテスト命令に応答し、ビットテスト手順は、複数の要素の各処理済み要素について、インデックスによって示されるソースベクトルデータ構造の処理済み要素内のビット位置のテストビットの値に応じて複数の結果ビットのそれぞれの結果ビットを設定することを含む、コンピュータプログラムがある。

【０００７】

本技術は、以下の添付図面に示されるようなその実施形態を参照して、単なる例示によって、さらに説明される。

【図面の簡単な説明】

【０００８】

【図１】一実施形態のデータ処理装置を概略的に示す図である。

【図２】一実施形態のソースベクトルの要素に対してビットテスト手順を実行するデータ処理回路を概略的に示す図である。

【図３Ａ】一実施形態の支配述語値を参照してソースベクトルの要素に対してビットテスト手順を実行するデータ処理回路を概略的に示す図である。

【図３Ｂ】図３Ａに示される実施形態に対応する２つの例示的なビットテスト命令を示す図である。

【図４Ａ】一実施形態のスカラーインデックス値を参照してビットテスト手順を実行するための例示的なビットテスト命令および対応するデータ処理回路を示す図である。

【図４Ｂ】一実施形態のインデックスベクトルによって与えられた各要素について特定のインデックス値を参照してソースベクトルの要素に対してビットテスト手順を実行するための、例示的なビットテスト命令および対応するデータ処理回路を示す図である。

【図５】一実施形態の方法を実行するときに取られる一連のステップを示す図である。

【図６】一実施形態のシミュレータ実装を提供するシステムのコンポーネントを概略的に示す図である。

【発明を実施するための形態】

【０００９】

少なくともいくつかの実施形態は、命令をデコードし、命令に応じた制御信号を生成する命令デコード回路と、命令デコード回路によって生成された制御信号に応えてデータ処理動作を実行するデータ処理回路とを備える装置であって、命令デコード回路は、複数の結果ビットを生成するためにソースベクトルレジスタ内に格納された複数の要素に対してデータ処理回路にビットテスト手順を実行させるため、ソースベクトルレジスタおよびインデックスを指定するビットテスト命令に応答し、ビットテスト手順は、複数の要素の各処理済み要素について、インデックスによって示されるソースベクトルレジスタの処理済み要素内のビット位置のテストビットの値に応じて複数の結果ビットのそれぞれの結果ビットを設定することを含む、装置を提供する。

【００１０】

したがって、ビットテスト命令、およびこれを実行するための対応するサポート回路は、テストすべきそれぞれのビットがソースベクトルのそれぞれの要素内で見つかるアプローチによって、複数のビットに対して平行してビットテストを実行する能力をプログラマに提供する。これを保持するベクトルレジスタを参照して指定されたソースベクトルは、こうしてビットテスト手順への１つの入力を形成し、ビットテスト命令で指定されたインデックスにより、プログラマは、ソースベクトル内の要素のどのビットがビットテストを受けるかを定義することができる。ビットテスト命令に応えて実行されるビットテスト手順の結果は、結果ビットのセットであり（１つの結果ビットは、ビットテスト手順を受けるソースベクトルの複数の要素の各処理済み要素に対応する）、複数の結果ビットの各結果ビットは、対応するテストビットの値に応じて設定される。したがって、ビットテストの文脈では、結果ビットはテストビットと一致するように設定されることが可能であり、または反対に、結果ビットはテストビットの補数（逆数）になるように設定されることが可能であることが理解され、これは本質的に実装の選択である。したがって、このビットテスト命令の提供により、複数のビットテストを実行するために必要とされ、ベクトル化

された形式に適切に定式化され得る、任意のプログラムコードの性能を向上させることができる。基本的に、線形アプローチの代わりに並列化アプローチが提供される。

【００１１】

データ処理回路の能力および構成によっては、ソースベクトルレジスタの要素の全てがビットテスト手順を受けなくてもよい。言い換えると、ソースベクトルレジスタの要素の厳密なサブセットのみがビットテスト手順を受けてもよい。しかしながら、いくつかの実施形態では、ビットテスト手順を受ける複数の要素は、ソースベクトルレジスタの全ての要素を含む。

【００１２】

プログラマには、ソースベクトルレジスタのどの要素がビットテスト手順を受けるかを指定する能力が提供されてもよい。したがって、ビットテスト手順のいくつかの実施形態では、支配述語ビット値のセットのそれぞれの述語ビットが第１所定値を有するとき、複数の要素のうちの１つの要素は処理済み要素としてビットテスト手順を受ける。言い換えると、プログラマは、ソースベクトルの対応する要素に対してビットテスト手順を実行させるために、対応する述語ビットを第１所定値（たとえば１）に設定する。

【００１３】

データ処理回路の構成および能力に応じて、支配述語ビット値のセットは、たとえばソースベクトルレジスタの要素の厳密なサブセットにのみ対応する、ソースベクトルレジスタ内の要素の数に関連して長さが異なる可能性があるが、いくつかの実施形態では、支配述語ビット値のセット内のいくつかの値は、ソースベクトルレジスタ内のいくつかの要素と一致する。

【００１４】

支配述語ビット値のセットは様々な方法で提供され得るが、いくつかの実施形態では、ビットテスト命令は支配述語ビット値のセットを指定し、さらにいくつかの実施形態では、ビットテスト命令は支配述語ビット値のセットを保持するレジスタを指定する。したがって、たとえば指定されたレジスタに支配述語ビット値を設定することによって、プログラマは、ソースベクトルの要素のうちのどれがビットテスト手順を受けるかを決定できる。

【００１５】

ビットテスト手順は、このような支配述語ビット値のセットが提供されるときに様々に構成され得るが、ビットテスト手順のいくつかの実施形態では、支配述語ビット値のセットのそれぞれの述語ビットが第１所定値を有していないとき、複数の結果ビットのそれぞれの結果ビットが第２所定値に設定される。第１所定値および第２所定値の選択が任意の実装選択であることは明確に理解されるが、いくつかの実施形態では、支配述語ビット値が０であるとき、対応するそれぞれの結果ビットもまた０に設定される。したがって、反対に述語ビットが１に設定されるとき、ソースベクトルレジスタの対応する要素はビットテスト手順の一部として処理され、インデックスによって示されるビット位置における処理済み要素のテストビットは、対応する結果ビットを決定する。

【００１６】

データ処理回路は、様々な方法で複数の結果ビットを利用し得るが、いくつかの実施形態では、データ処理回路は、複数の結果ビットを結果レジスタに格納するように配置される。この結果レジスタは、デフォルトで既知であってもよく、またはいくつかの実施形態では、結果レジスタはビットテスト命令において指定される。したがって、プログラマには、複数の結果ビットが格納されるべきレジスタを指定する能力が与えられる。

【００１７】

一般に、複数の結果ビットの数は、ソースベクトルの複数の要素に関連して、たとえばデータ処理回路の能力および構成に応じて異なってもよく、複数の結果ビットの数はソースベクトルレジスタ内の要素の数よりも少なくてもよいが、いくつかの実施形態では、複数の結果ビットのカウントは、ソースベクトルレジスタに格納された複数の要素のカウントと一致する。

【００１８】

複数の結果ビットのそれぞれの結果ビットは、インデックスによって示されるソースベクトルレジスタの処理済み要素内のビット位置におけるテストビットの値に応じて設定され、装置がこの設定を補完的に行うために設定され得ることは、理解されるだろう。言い換えると、ビットテスト手順のいくつかの実施形態では、複数の結果ビットのそれぞれの結果ビットは、テストビットの値と一致するように設定される。反対に、ビットテスト手順の別の実施形態では、複数の結果ビットのそれぞれの結果ビットは、テストビットの値と一致しないように設定される。実際、これら２つの変形は２つの異なるビットテスト命令として提供され得、これらは本明細書において、ＴＳＴＺ（「テストゼロ(test zero)」）およびＴＳＴＮＺ（「テスト非ゼロ(test non-zero)」）と呼ばれる。

【００１９】

また、両方のタイプのビットテスト命令に応答し得る本技術による装置が提供されてもよく、こうしてプログラマに、ゼロ設定ビットでテストするか非ゼロ設定ビットでテストするかの選択肢を与え、したがっていくつかの実施形態では、命令デコード回路はさらなるビットテスト命令に応答し、これによってデータ処理回路は、さらなるテストビットの値と一致しないようにさらなる複数の結果ビットのさらなるそれぞれの結果ビットを設定することを含む、さらなるビットテスト手順を実行することができる。言い換えると、プログラマは、結果ビットがテストビットの値と一致するように設定されるビットテスト命令を使用し、その後、結果ビットがテストビットと一致するように設定されないさらなるビットテスト命令を使用することができる。

【００２０】

反対に、いくつかの実施形態では、命令デコード回路はさらなるビットテスト命令に応答し、これによってデータ処理回路は、さらなるテストビットの値と一致するようにさらなる複数の結果ビットのさらなるそれぞれの結果ビットを設定することを含む、さらなるビットテスト手順を実行する。したがって、プログラマは、結果ビットをテストビットの値と一致しないように設定させるビットテスト命令を使用し、その後、結果ビットをテストビットと一致させるさらなるビットテスト命令を使用することができる。本明細書で論じられる特定の実施形態の言語では、プログラマは、ＴＳＴＺ命令およびＴＳＴＮＺ命令の両方を、どのような定義でも、どちらの順序でも、使用することができる。

【００２１】

インデックスは、様々に定義され得るが、いくつかの実施形態では、インデックスは、ビットテスト命令で指定された即値である。あるいは、ビットテスト命令はスカラーインデックスレジスタを指定してもよく、このような実施形態では、インデックスは、ビットテスト命令で指定されたスカラーインデックスレジスタに格納されたスカラー値である。さらに別の実施形態では、ビットテスト命令は、複数のインデックス値を保持するベクトルインデックスレジスタを指定してもよく、ビットテスト手順において、複数の要素の各処理済み要素では、インデックスは、複数のインデックス値のそれぞれのインデックス値によって与えられる。これは、スカラーの例とは異なり、プログラマが、ソースベクトルの複数の要素の各処理済み要素について異なるインデックス値を使用し、こうして所与の要素のいずれのビットもビットテストできることを意味する。

【００２２】

少なくともいくつかの実施形態は、命令をデコードして命令に応じた制御信号を生成し、生成された制御信号に応えてデータ処理動作を実行し、ソースベクトルレジスタおよびインデックスを指定するビットテスト命令に応えて、複数の結果ビットを生成するためにソースベクトルレジスタ内に格納された複数の要素に対するビットテスト手順の実行を行わせる方法であって、ビットテスト手順は、複数の要素の各処理済み要素について、インデックスによって示されるソースベクトルレジスタの処理済み要素内のビット位置のテストビットの値に応じて複数の結果ビットのそれぞれの結果ビットを設定することを備える、方法を提供する。

【００２３】

少なくともいくつかの実施形態は、命令をデコードし、命令に応じた制御信号を生成す

る手段と、生成された制御信号に応えてデータ処理動作を実行する手段と、ソースベクトルレジスタおよびインデックスを指定するビットテスト命令に応えて、複数の結果ビットを生成するためにソースベクトルレジスタ内に格納された複数の要素に対するビットテスト手順の実行を行わせる手段と、を備える装置であって、ビットテスト手順は、複数の要素の各処理済み要素について、インデックスによって示されるソースベクトルレジスタの処理済み要素内のビット位置のテストビットの値に応じて複数の結果ビットのそれぞれの結果ビットを設定することを備える、装置を提供する。

【００２４】

別の例示的実施形態では、命令をデコードし、命令に応じた制御信号を生成するための命令デコードプログラムロジックと、命令デコードプログラムロジックによって生成された制御信号に応えてデータ処理動作を実行するためのデータ処理プログラムロジックと、を備える命令実行環境を提供するようにホストデータ処理装置を制御するためのコンピュータプログラムであって、命令デコードプログラムロジックは、データ処理プログラムロジックに、複数の結果ビットを生成するためにソースベクトルデータ構造内に格納された複数の要素に対してビットテスト手順を実行させるため、ソースベクトルデータ構造およびインデックスを指定するビットテスト命令に応答し、ビットテスト手順は、複数の要素の各処理済み要素について、インデックスによって示されるソースベクトルデータ構造の処理済み要素内のビット位置のテストビットの値に応じて複数の結果ビットのそれぞれの結果ビットを設定することを含む、コンピュータプログラムがある。

【００２５】

少なくともいくつかの実施形態は、上述の実施形態によるコンピュータプログラムを格納するコンピュータ可読記憶媒体を提供する。プログラムは、非一時的に格納されてもよい。

【００２６】

ここで、いくつかの具体的な実施形態が、図面を参照して説明される。

【００２７】

図１は、一実施形態のデータ処理装置を概略的に示す。データ処理装置１０の一般的な構成は、当業者によく知られており、本技術に特に関連しないコンポーネントの詳細な説明は、簡潔にするために本明細書では省略される。概略的に、図１に示されるように、データ処理装置１０は、関連するシステムレジスタ１４、汎用レジスタ１６、およびベクトルレジスタ１８を有する実行回路１２を備える。実行回路１２は、そのデータ処理動作を実行する際に、これらのレジスタを利用する。実行回路１２が実行するデータ処理動作は、フェッチ回路２２の動作によってメモリ２０から取得された一連の命令によって定義される。メモリ２０から命令を取得する際に、命令は、レベル１命令キャッシュ２４および統合レベル２キャッシュ２６の一方または両方に一時的にキャッシュされてもよい。このようにしてフェッチ回路２２によってフェッチされた命令は、取得された命令およびこれらが担持する特定の命令パラメータに基づいて実行回路１２の制御信号を生成する、デコード回路２８に渡される。実行回路１２は、上述のように、レジスタ１４、１６、および１８に関して、また必要に応じて、メモリ２０からデータ項目を取得し、処理したデータ項目をメモリ２０に書き戻すことによって、データ処理動作を実行する。命令の場合のように、実行回路１２によってメモリ２０内でアクセスされたデータ項目は、キャッシュ階層、すなわちレベル１データキャッシュ３０および統合レベル２キャッシュ２６によって、同様にアクセスされる。実行回路１２のさらなる詳細は、以下の図面を参照して、様々な実施形態に関連してここで与えられる。

【００２８】

図２は、ビットテスト命令によって開始されるビットテスト手順を実行するために提供された一実施形態における実行回路１２の一部の構成を概略的に示す。ビットテスト命令は、２つの変形、すなわちＴＳＴＺおよびＴＳＴＮＺで、図２の上部に示される。図示されるように、これらの命令は、ソースベクトル（レジスタ）およびインデックス値を指定する。この例では、インデックス値は即値である。図示される例では２の即値を有するイ

ンデックス値により、データ処理回路は、ソースベクトルの各要素内のビット位置２におけるそれぞれのビットにアクセスできる。したがって、ソースベクトルが、各々８ビットを含む４つの要素を備える図示の例では、これによってデータ処理回路は、（ビット位置２において）各要素の３つ目のビットにアクセスできる。これらのビットは、コンパレータ４０、４２、４４、および４６に渡される。これらのコンパレータの各々は、ビットテスト命令がＴＳＴＺかＴＳＴＮＺかを示すバイナリ入力である、第２の入力を有する。このバイナリ入力は、ＴＳＴＺ命令では、コンパレータが、ソースベクトルの対応する要素内のインデックスされたビット位置におけるテストビットの補数を生成し、その一方で命令がＴＳＴＮＺであるときにはそれぞれのコンパレータにより、結果ビットがソースベクトルの対応する要素内のインデックスされたビット位置におけるテストビットと一致するように、各コンパレータによって実行されるビットテストを反転させる。図２に示される例示的な値はＴＳＴＺ命令を受け、したがってコンパレータは、結果ビットのセットを作成するために、ソースベクトルからの４つのテストビットのセットを反転させる。

【００２９】

図３Ａは、ここではビットテスト手順が支配述語ビット値のセットの内容にさらにしたがって実行される、さらなる例示的実施形態におけるデータ処理回路の関連コンポーネントを概略的に示す。ＴＳＴＺおよびＴＳＴＮＺとも呼ばれる、ビットテスト命令の対応する対が、図３Ｂに示される。図３Ａの項目６０として示される、支配述語ビット値のセットのそれぞれのビットは、ソースベクトルの特定の要素がビットテスト手順のビットテストを受けるか否かを決定する。図示される例では、支配述語ビット値のセットの対応するビットが１に設定されるソースベクトルの要素のみが、このビットテストを受ける。これを実施するために、ビットテスト制御回路６２は、支配述語ビット値６０のセットを受け取り、ビットテストコンパレータ６４、６６、６８、および７０の動作を制御するオーバーライド信号のセットを生成する。図２の例のように、これらのコンパレータは、ソースベクトルの各要素からのインデックス値によって識別された対応するテスト対象ビットを受け取り、バイナリＴＳＴＺ／ＴＳＴＮＺ入力によっても制御される。結果ビット７２のセットは、こうしてコンパレータによって生成される。したがって、支配述語６０の２つの最下位ビットのみが設定される図３Ａの例では、結果ビットのセットの２つの最上位ビットが自動的に０に設定され、その一方で、結果ビット７２のセットの２つの最下位ビットは、これらがテストとしたそれぞれのビットに応じて、コンパレータ６８および７０によって設定される。これは、図３Ａの例で実行されているＴＳＴＺ命令であり、したがってテストビットは、結果ビット７２のセットの２つの最下位ビットを「０」および「１」として生成するために反転される。図３Ｂに示される例に見られるように、この例のＴＳＴＺおよびＴＳＴＮＺ命令は、図２の例としてソースベクトルレジスタおよびインデックス即値として指定するが、ここでは支配述語レジスタおよび宛先レジスタをさらに指定する。したがって、支配述語ビット値６０のセットは指定された支配述語レジスタから取得され、一方で結果ビット７２のセットは指定された宛先レジスタに書き込まれる。図３Ｂの命令は図２の例に関して宛先レジスタおよび支配述語レジスタの両方を追加で指定するという事実に意味はなく、別の例示的実施形態では、ビットテスト命令は、ソースレジスタおよびインデックスに加えてこれらのうちの１つのみを指定してもよいことは、理解されるだろう。

【００３０】

図４Ａは、ＴＳＴＺ命令が、宛先レジスタ、支配述語レジスタ、ソースベクトルレジスタ、インデックス（スカラー）レジスタを指定する例を示す。したがって、対応するビットテスト手順を実行するデータ処理回路は、指定されたレジスタＰｇから支配述語ビット値８０のセットを、指定されたインデックスレジスタからインデックス値８２を、取得する（この例では、インデックス値は７）。したがって、ソースベクトルＺｓのそれぞれの要素において、支配述語８０の対応するビットが設定されると、各（ｉ＝７）の８番目のビットがアクセスされてテストされる。説明を簡単にするために、簡略化された構成が図４Ａに示されており、ソースベクトルＺｓのテストビットから宛先述語８４のそれぞれの

結果ビットまで矢印が直接延びている。実際には、これらの矢印は、図３Ａの例に示されるもののようなコンパレータのセットを介して実施され、これらは支配述語８０のそれぞれのセットによって、ならびに命令がＴＳＴＺかＴＳＴＮＺかを示すバイナリ値によっても制御される。命令がＴＳＴＺである図４Ａの例では、テストビットは、宛先述語８４の対応するビットを提供するために反転される（これは表記によって図４Ａに示されている：ソースビット（０／１）は結果ビット（１／０）へ）。

【００３１】

図４Ｂを見ると、やはり宛先結果レジスタ、支配述語レジスタ、およびソースベクトルレジスタを指定し、この例ではインデックスベクトルレジスタも指定する、ＴＳＴＮＺ命令の例が示されている。したがって、支配述語８６は、ビットテスト手順を受けるべきソースベクトルＺｓの要素を再び選択する一方で、各テスト済み要素のうちどのビットがテストされるかは、インデックスベクトル８８の対応する要素において与えられたインデックスによって個別に指定される。このように、図４Ｂで与えられる例では、ビット位置４は要素０でテストされ、ビット位置２は要素２でテストされ、ビット位置７は要素４でテストされ、ビット位置０は要素６でテストされることがわかる。支配述語８６の対応するビットは０に設定され、したがって述語９０の宛先の対応するビットも０に設定されるので、要素１、３、５、および７ではビットはテストされない。

【００３２】

図４Ｂに示される例はＴＳＴＮＺ命令の例であり、したがってソースベクトルのテストビットは宛先述語９０の対応するビット内に効率的に複製される（これは表記によって図４Ｂに示されている：ソースビット（０／１）は結果ビット（０／１）へ）。図４Ａの場合のように、図４Ｂの説明は、説明を簡単にするために、ソースベクトルＺｓのそれぞれの要素のテストビットから宛先述語９０の対応するビット位置まで単に矢印が延びている、簡略化された形態で示されている。以前のように、実際には、これらは、テスト対象入力ビットを受け取り、命令がＴＳＴＺかＴＳＴＮＺかを示すバイナリ入力によって制御される、コンパレータのセットを介して実施される。

【００３３】

さらなる変形例は、上記で説明された例示的実施形態のいずれにも適用可能であり、すなわちビットテスト命令が、そのレジスタの内容のサイズ解釈を示すサイズ指定子を用いてその指定されたレジスタのいずれかをさらに修整し得るものである。これは、たとえばＴＳＴＺ＜Ｐｄ＞．＜Ｔ＞，＜Ｐｇ＞．＜Ｔ＞，＜Ｚｓ＞．＜Ｔ＞，インデックスなどのように記述され得る。このサイズ指定子は、たとえば００＝バイト長（Ｂ）；０１＝半語長（Ｈ）；１０＝１倍長（Ｓ）；および１１＝２倍長（Ｄ）などの命令において、コンパクトに符号化され得る。

【００３４】

サイズ指定子＜Ｔ＞の使用により、プログラマは、要素のそれぞれのベクトルの各要素においてデータ値のサイズを指定することができる。したがって、このサイズ指定子がない場合、または実際にサイズ指定子がデフォルトと一致する値を有する場合には、たとえば「Ｓ」、１倍長など、レジスタの要素のデフォルト解釈が使用され得る。しかしながら、＜Ｔ＞の値が命令で指定された場合には、各ベクトル要素の指定されたサイズが使用される。サイズ指定子自体がレジスタによって与えられ、必要なサイズ値を抽出するためにデータ処理回路がその指定されたサイズ指定レジスタにアクセスする、この機能に関するさらなる変形例もまた、上記の例示的実施形態のいずれでも可能である。

【００３５】

図５は、一実施形態の方法にしたがって実行される一連のステップを示す。フローはステップ１００で開始されると見なされ、命令デコード回路が、装置に提供された一連の命令における次の命令をデコードする。この図を簡単にするために、デコードすべき次の命令が常にあると想定される。ステップ１０２において、これがビットテスト命令であるか否かが判断され、そうでない場合には、この命令は、この命令にとって通常の方法でステップ１０４において単に実行され、フローはステップ１００に戻る（ここでは新規なビッ

トテスト命令のみが対象となるため）。ビットテスト命令では、フローはステップ１０２からステップ１０４に進み、そこで支配述語のいずれかのビットが０に設定されているか否かが判断される。これは、結果述語の対応する１つまたは複数のビットもまた０に設定されるステップ１０６を介してフローが進む場合である。次にステップ１０８において、ビットテスト命令がベクトル（定義された即値またはスカラーレジスタの対語として）インデックスを指定したか否かが判断される。そうである場合には、ベクトルインデックスの対応する要素から取得された、ソースベクトルの各処理済み要素に一意のインデックスが使用される、ステップ１１０を介してフローが進む。そうでなければ、ステップ１１２において、インデックスのスカラー値（スカラーレジスタ内で、または命令内の即値として提供される）は、ソースベクトルの全ての処理済み要素に使用される。次にステップ１１４において、これがＴＳＴＺ命令かＴＳＴＮＺ命令かが判断される。ＴＳＴＮＺ命令では、フローはステップ１１６に進み、そこでソースベクトルの各処理済み要素のインデックスビットが、結果述語のそれぞれのビットに複製される。あるいは、これがＴＳＴＺ命令である場合、フローは、ソースベクトルの各処理済み要素の反転インデックスビットが結果述語のそれぞれのビットに複製される、ステップ１１８を介して進む。その後、このビットテスト命令の実行が完了し、フローはステップ１００に戻る。

【００３６】

図６は、使用され得るシミュレータ実装を示す。先に説明された実施形態は、関連する技術をサポートする特定の処理ハードウェアを動作させる装置および方法に関して本発明を実施したが、コンピュータプログラムの使用を通じて実施される本明細書に記載の実施形態による命令実行環境を提供することも、可能である。このようなコンピュータプログラムは、ハードウェアアーキテクチャのソフトウェアベースの実装を提供する限りにおいて、しばしばシミュレータと呼ばれる。様々なシミュレータコンピュータプログラムは、エミュレータ、仮想マシン、モデル、および動的バイナリトランスレータを含むバイナリトランスレータを含む。通常、シミュレータ実装はホストプロセッサ７３０上で実行されてもよく、任意選択的に、ホストオペレーティングシステム７２０を実行し、シミュレータプログラム７１０をサポートする。いくつかの配置では、ハードウェアと提供された命令実行環境との間、および／または同じホストプロセッサ上に提供された複数の異なる命令実行環境の間に、複数のシミュレーションの層があってもよい。歴史的には、合理的な速度で実行するシミュレータ実装を提供するために強力なプロセッサが必要とされてきたが、このようなアプローチは、互換性または再利用の理由で別のプロセッサにネイティブなコードを実行したいときなど、特定の状況において正当化され得る。たとえば、シミュレータ実装は、ホストプロセッサハードウェアによってサポートされない追加機能を有する命令実行環境を提供してもよく、または通常は異なるハードウェアアーキテクチャに関連付けられた命令実行環境を提供してもよい。シミュレーションの概要は、“Ｓｏｍｅ Ｅｆｆｉｃｉｅｎｔ Ａｒｃｈｉｔｅｃｔｕｒｅ Ｓｉｍｕｌａｔｉｏｎ Ｔｅｃｈｎｉｑｕｅｓ”，Ｒｏｂｅｒｔ Ｂｅｄｉｃｈｅｋ，Ｗｉｎｔｅｒ １９９０ ＵＳＥＮＩＸ Ｃｏｎｆｅｒｅｎｃｅ、５３～６３ページに記載されている。

【００３７】

特定のハードウェア構造または機能を参照して実施形態が以前に説明された限りにおいて、シミュレートされた実施形態では、適切なソフトウェア構造または機能によって同等の機能が提供され得る。たとえば、特定の回路が、コンピュータプログラムロジックとしてシミュレートされた実施形態で実装されてもよい。同様に、レジスタまたはキャッシュなどのメモリハードウェアが、ソフトウェアデータ構造としてシミュレートされた実施形態に実装されてもよい。先に記載された実施形態で言及されたハードウェア要素の１つ以上がホストハードウェア（たとえば、ホストプロセッサ７３０）上に存在する配置では、適切であれば、いくつかのシミュレートされた実施形態は、ホストハードウェアを利用してもよい。

【００３８】

シミュレータプログラム７１０は、コンピュータ可読記憶媒体（非一時的媒体であって

もよい）上に格納されてもよく、シミュレータプログラム７１０によってモデリングされているハードウェアアーキテクチャのアプリケーションプログラムインターフェースと同じであるターゲットコード７００にプログラムインターフェース（命令実行環境）を提供する。したがって、上記のビットテスト命令を含む、ターゲットコード７００のプログラム命令は、上記で論じられた装置のハードウェア機能を実際には有していないホストコンピュータ７３０がこれらの機能をエミュレートできるように、シミュレータプログラム７１０を使用して、命令実行環境内から実行され得る。

【００３９】

簡単な全体的要約では、装置および装置を動作させる方法が提供される。装置は、複数の結果ビットを生成するためにソースベクトルレジスタに格納された複数の要素に対してビットテスト手順を実行するため、ソースベクトルレジスタおよびインデックスを指定するビットテスト命令に応答する。ビットテスト手順は、複数の要素の各処理済み要素について、インデックスによって示されるソースベクトルレジスタの処理済み要素内のビット位置のテストビットの値に応じて複数の結果ビットのそれぞれの結果ビットを設定することを含む。したがって、このビットテスト命令により、複数のビットテストを実行するために必要とされ、ベクトル化された形式に適切に定式化され得る、プログラムコードの性能を向上させることができる。

【００４０】

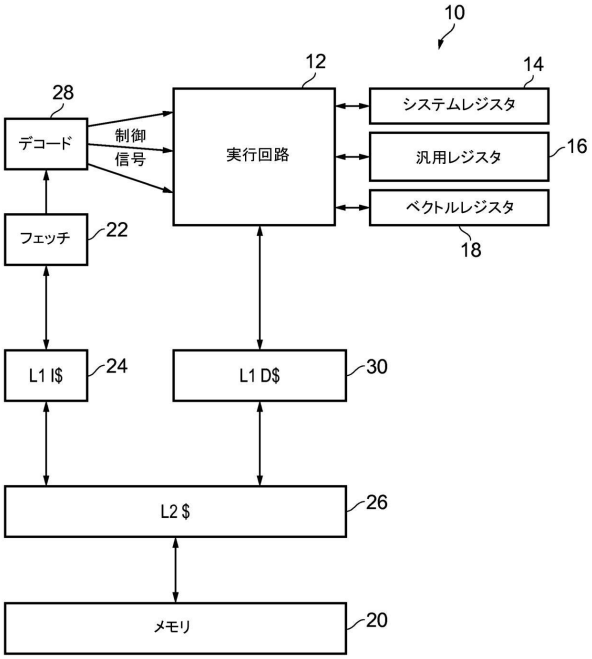
本出願において、「～するように構成された」という用語は、装置の要素が定義された動作を実行できる構成を有することを意味するために使用される。この文脈で、「構成」は、ハードウェアまたはソフトウェアの配置、または相互接続の方法を意味する。たとえば、装置は、所定の動作を提供する専用ハードウェアを有してもよく、あるいはプロセッサまたは他の処理装置が機能を実行するようにプログラムされてもよい。「構成された」は、所定の動作を提供するために装置要素が多少なりとも変更されなければならないことを示唆するものではない。

【００４１】

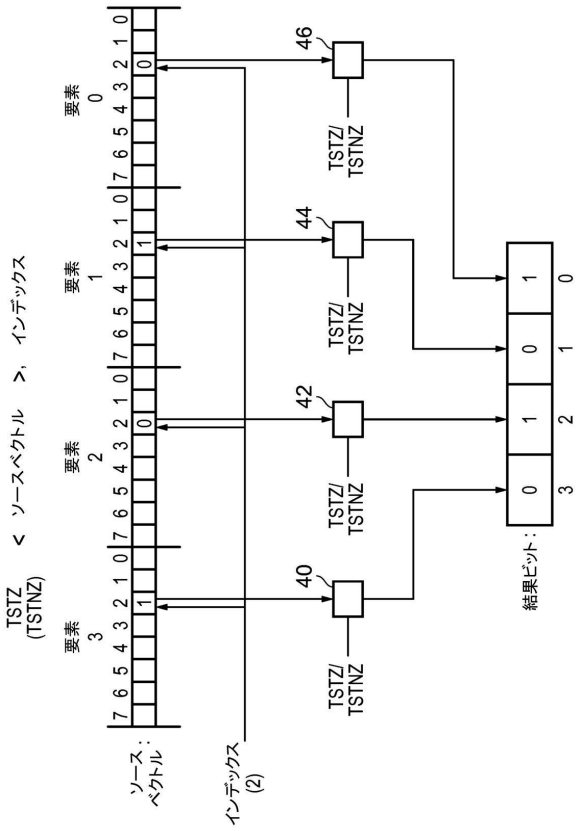
以上、添付図面を参照して例示的な実施形態を本明細書で詳細に説明したが、本発明はこれらの厳密な実施形態に限定されるものではなく、添付の特許請求によって規定される本発明の範囲および趣旨から逸脱することなく、様々な変更、追加、および修正が当業者によって実施され得ることが、理解されるべきである。たとえば、従属請求項の特徴の様々な組み合わせは、本発明の範囲から逸脱することなく、独立請求項の特徴と一緒になされ得る。

【図面】

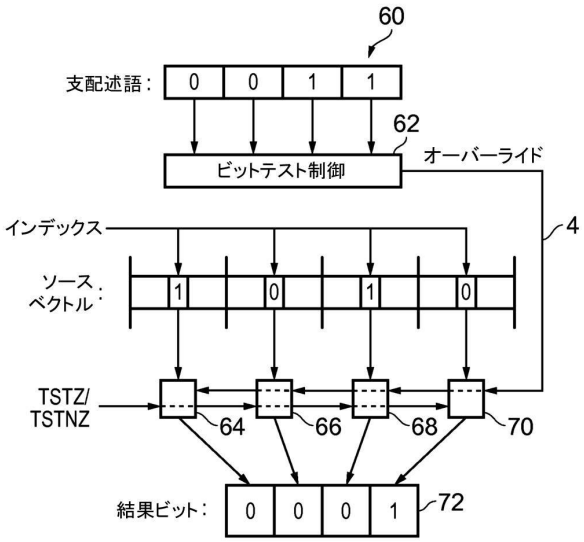
【図１】



【図２】



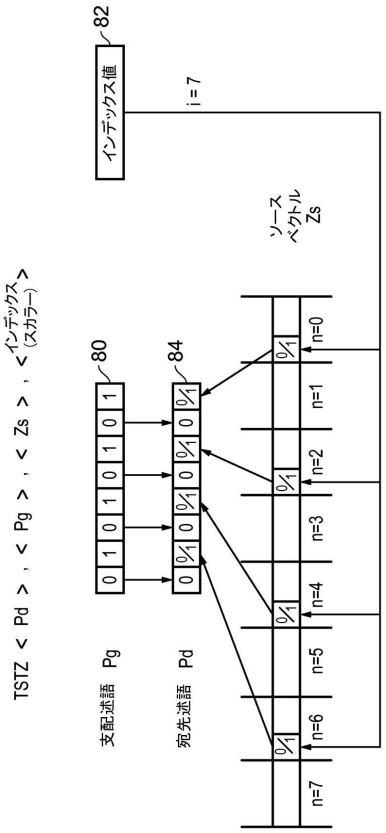
【図３Ａ】



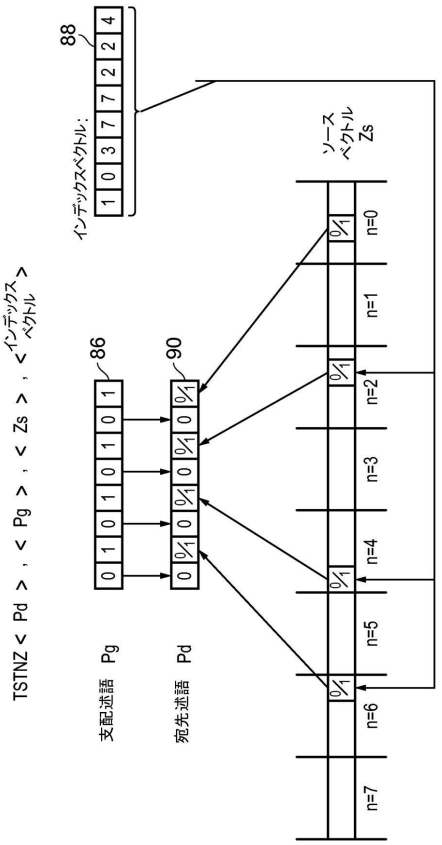
【図３Ｂ】

TSTZ (TSTNZ) < 宛先レジスタ >，< 支配述語レジスタ >，< ソースベクトルレジスタ >，インデックス

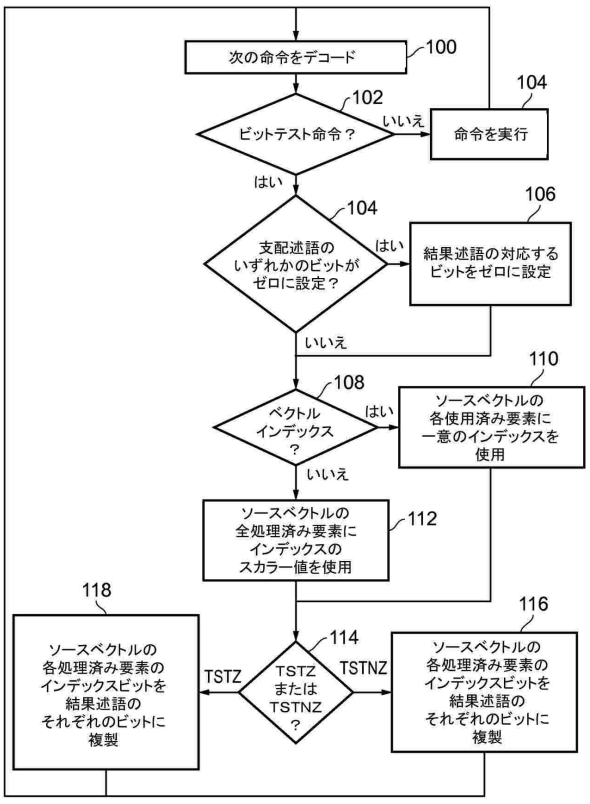
【図４Ａ】


82
インデックス値
i = 7
TSTZ < Pd > , < Pg > , < Zs > , < インデックス (スカラー) >
80
84
ソース ベクトル Zs
n=0
n=1
n=2
n=3
n=4
n=5
n=6
n=7
支配述語 Pg
宛先述語 Pd

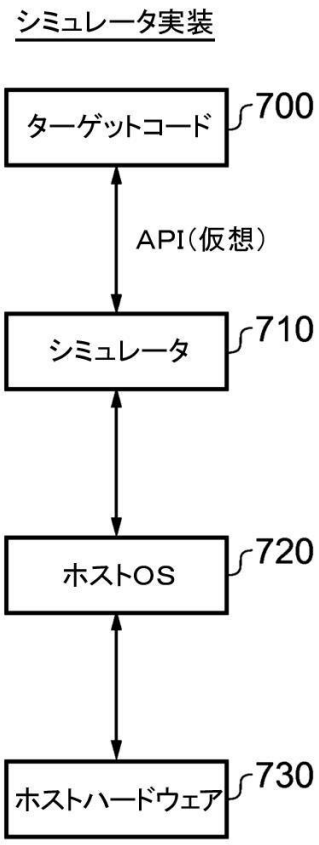
【図４Ｂ】


88
インデックスベクトル:
1 0 3 7 7 2 2 4
TSTNZ < Pd > , < Pg > , < Zs > , < インデックス ベクトル >
86
90
ソース ベクトル Zs
n=0
n=1
n=2
n=3
n=4
n=5
n=6
n=7
支配述語 Pg
宛先述語 Pd

【図５】


次の命令をデコード
100
ビットテスト命令？
102
いいえ
命令を実行
104
はい
支配述語の いずれかのビットが ゼロに設定？
104
はい
結果述語の対応する ビットをゼロに設定
106
いいえ
ベクトル インデックス ？
108
はい
ソースベクトルの 各使用済み要素に 一意のインデックスを 使用
110
いいえ
ソースベクトルの 全処理済み要素に インデックスの スカラー値を使用
112
TSTZ または TSTNZ ？
114
TSTZ
TSTNZ
ソースベクトルの 各処理済み要素の インデックスビットを 結果述語の それぞれのビットに 複製
118
ソースベクトルの 各処理済み要素の インデックスビットを 結果述語の それぞれのビットに 複製
116

【図６】


シミュレータ実装
ターゲットコード
700
API(仮想)
シミュレータ
710
ホストOS
720
ホストハードウェア
730

フロントページの続き

ジ、チェリー ヒントン、フルボーン ロード １１０、エイアールエム リミテッド 気付

合議体
審判長　林 毅
審判官　脇岡 剛
審判官　山崎 慎一

(56)参考文献　欧州特許出願公開第０２８８９７５６号明細書（ＥＰ，Ａ１）
特開昭５３－０５３２３７号公報（ＪＰ，Ａ）
特開２０１４－１８２８０２号公報（ＪＰ，Ａ）
米国特許出願公開第２００８／０２５３６６８号明細書（ＵＳ，Ａ１）
特開２００５－１７４２９８号公報（ＪＰ，Ａ）

(58)調査した分野　(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　９／３０８
Ｇ０６Ｆ　９／３４５