(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4716911号
(P4716911)
(45)発行日　平成23年7月6日(2011.7.6)　(24)登録日　平成23年4月8日(2011.4.8)

(51)Int.Cl.　F I
A61B 8/00 (2006.01)　A61B 8/00
G06F 15/80 (2006.01)　G06F 15/80

請求項の数 9　(全 25 頁)

(21)出願番号　特願2006-100226 (P2006-100226)
(22)出願日　平成18年3月31日 (2006.3.31)
(65)公開番号　特開2007-268156 (P2007-268156A)
(43)公開日　平成19年10月18日 (2007.10.18)
審査請求日　平成20年12月15日 (2008.12.15)

(73)特許権者　390029791
日立アロカメディカル株式会社
東京都三鷹市牟礼６丁目２２番１号
(74)代理人　100075258
弁理士　吉田　研二
(74)代理人　100096976
弁理士　石田　純
(72)発明者　尾形　太
東京都三鷹市牟礼６丁目２２番１号　アロカ株式会社内

審査官　右▲高▼　孝幸

最終頁に続く

(54)【発明の名称】超音波診断装置用プロセッサ

(57)【特許請求の範囲】
【請求項１】
超音波の送受信により得られた入力データ列を構成する各入力データを並列処理する超音波診断装置用プロセッサにおいて、
前記入力データ列と比較データ列とを比較して、比較結果データ列を生成する第１の演算手段と、
前記比較結果データ列を構成する各比較結果データから代表ビットを抽出し、それらの代表ビットによって代表ビット列を生成する代表ビット列抽出手段と、
前記代表ビット列が表し得るビットパターンに対応した複数の操作データ列を格納した操作テーブルと、
前記複数の操作データ列の中から前記代表ビット列に応じて選択された特定の操作データ列を用いて、前記入力データ列に対するデータ演算を実行し出力データ列を生成する第２の演算手段と、
を有することを特徴とする超音波診断装置用プロセッサ。
【請求項２】
請求項１記載の超音波診断装置用プロセッサにおいて、
前記データ演算は、条件分岐処理に相当する論理演算を含むことを特徴とする超音波診断装置用プロセッサ。
【請求項３】
請求項２記載の超音波診断装置用プロセッサにおいて、

前記データ演算は、更に算術演算を含むことを特徴とする超音波診断装置用プロセッサ。

【請求項４】

請求項１記載の超音波診断装置用プロセッサにおいて、

前記データ演算は、前記入力データ列を構成する各入力データに対して、前記特定の操作データ列を構成する各操作データを作用させて、前記各入力データの保存処理または変更処理を選択的に実行する演算を含むことを特徴とする超音波診断装置用プロセッサ。

【請求項５】

請求項１記載の超音波診断装置用プロセッサにおいて、

前記データ演算は、前記入力データ列を構成する各入力データに対して、前記特定の操作データ列を構成する各操作データに応じて生成された派生操作データを作用させて、前記各入力データの保存処理または変更処理を選択的に実行する演算を含むことを特徴とする超音波診断装置用プロセッサ。

【請求項６】

請求項１記載の超音波診断装置用プロセッサにおいて、

前記代表ビット列を構成する各代表ビットは、正負を表わす符号ビットであることを特徴とする超音波診断装置用プロセッサ。

【請求項７】

請求項１記載の超音波診断装置用プロセッサにおいて、

前記比較データ列を構成する各比較データは、大小判別のための複数の閾値データであることを特徴とする超音波診断装置用プロセッサ。

【請求項８】

超音波の送受信により得られたＮ個（但し、Ｎは２以上の整数）の入力データを並列処理する超音波診断装置用プロセッサにおいて、

前記Ｎ個の入力データとＮ個の比較データとを比較してＮ個の比較結果データを生成する第１の演算手段と、

前記Ｎ個の比較結果データからＮ個の代表ビットを抽出し、それらの代表ビットによって代表ビット列を生成する代表ビット列抽出手段と、

前記代表ビット列が表し得るビットパターンに対応する複数の操作データ列を格納した操作テーブルと、

前記複数の操作データ列の中から前記代表ビット列に応じて特定の操作データ列を選択し、この特定の操作データ列を構成するＮ個の操作データを用いて、前記Ｎ個の入力データに対してデータ演算を実行し、Ｎ個の出力データを生成する第２の演算手段と、

を有することを特徴とする超音波診断装置用プロセッサ。

【請求項９】

超音波の送受信により得られた入力データ列を構成する各入力データを並列処理する超音波診断装置用プロセッサにおいて、

前記超音波診断装置用プロセッサは、演算部と記憶部とを有し、

前記演算部は、

前記入力データ列と比較データ列との算術演算により比較結果データ列を生成し、

前記比較結果データ列を構成する各比較結果データから代表ビットを抽出し、それらの代表ビットによって代表ビット列を生成し、

予め用意された前記複数の操作データ列の中から前記代表ビット列に応じて選択された特定の操作データ列を用いて、前記入力データ列に対するデータ演算を実行して出力データ列を生成し、

前記記憶部には、前記複数の操作データ列が格納され、

前記複数の操作データ列は、前記代表ビット列が表し得るビットパターンに対応することを特徴とする超音波診断装置用プロセッサ。

【発明の詳細な説明】

【技術分野】
【０００１】
本発明は、超音波診断装置用プロセッサに関し、特に、超音波の送受信によって得られたデータを並列処理する超音波診断装置用プロセッサに関する。
【背景技術】
【０００２】
医療用の超音波診断装置においては、超音波探触子が被検者の体表に当接あるいは体内に挿入されて超音波が発信され、超音波の反射波信号に基づいて体内の状態の診断画像等が得られる。一般に、超音波診断装置では装置内部での信号がデジタルデータに変換された上で処理されている。つまり、受信信号としてのアナログ信号はデジタル信号に変換され、その信号に対して多様な演算が何重にも施された結果、診断画像等が形成されている。そのために、超音波診断装置の内部で処理すべきデジタルデータの量は膨大なものとなる。そして、超音波診断装置の画像の応答性やリアルタイム表示性能を改善するためには、大量のデジタルデータを高速に処理し高速に画像化することは必要不可欠となっている。従って、超音波診断装置においてはそれぞれの処理の各段階において複数個のプロセッサが使用されており、膨大な量のデータを処理するために個々のプロセッサには、自ずと高速な処理速度が求められている。
【０００３】
ここで、中央演算処理装置（ＣＰＵ）やデジタルシグナルプロセッサ（ＤＳＰ）に代表されるプロセッサは、データ処理の高速化のための要となる部品である。ＤＳＰとは信号処理に特化した半導体デバイスであり、ＣＰＵと同様に信号処理プログラムを実行することで多様な機能を実現させることができる。これらのプロセッサは技術進歩に伴って、データの処理能力を向上させるための様々な高速化手段を搭載している。その高速化手段の一つとしてＳＩＭＤ(Single Instruction Multiple Data)方式が挙げられる。ＳＩＭＤ方式とは、単一命令で複数個のデータを同時並列的に処理するデータ処理方式を意味する。ＳＩＭＤ方式を採用しているプロセッサにおいて、取り扱われるデータ形式や命令セットは、並列に入力される複数のデータに対して一括の操作を行うことができるようになっている。つまり、プロセッサの最大データバス幅を１個のデータだけで占有するのではなく、そのデータバス幅を機能的に分割することにより、複数のデータが同時に並列して扱えるようになっている。ＳＩＭＤ方式のプロセッサを用いると、データの並列処理が可能となるので、処理速度を高速化できる。
【０００４】
以下、特許文献１にはＳＩＭＤ方式を用いた演算器及びその演算器を用いた演算処理装置に関する技術が記されている。また、特許文献２にはＳＩＭＤ方式のプロセッサを用いた超音波診断システムに関する技術が記されている。
【０００５】
【特許文献１】特開２０００－４７９９８号公報
【特許文献２】特開２０００－１８９４２２号公報
【発明の開示】
【発明が解決しようとする課題】
【０００６】
背景技術として示したように、ＳＩＭＤの機能を搭載したＣＰＵやＤＳＰを使って、データ演算を行うことは、処理速度の高速化のための有力な手段となっている。しかしながら、ＳＩＭＤの機能を用いたとしても、高速な処理速度を維持できない場合がある。それは、ｉｆ文に代表されるような条件分岐処理を実行する場合である。ＳＩＭＤ方式の演算を実行する場合において、演算の途中で演算結果の値を判定して、条件分岐を行うような処理はＳＩＭＤ向きではない。つまり、ある所定の条件に合致するかどうかを判断して、プログラム内のジャンプ先を二者択一で選択するような条件分岐処理は、ＳＩＭＤ方式の処理になじまないのである。なぜなら、ある条件に合致するかどうかの判定を、複数のデータに対して行った場合には、全部のデータが条件に合致するケースや、あるいは逆に全

部のデータが条件に合致しないケースはまれであり、通常の場合には条件に一致するデータと一致しないデータが混在するケースが多いからである。このように判定条件に対する一致と不一致が混在する場合には、判定結果に応じてデータ毎に個別の処理をする必要があり、並進処理の足並みが乱されてしまうことになる。このように、ＳＩＭＤの演算機能を搭載したプロセッサにおいて条件分岐処理をそのまま実行しようとすると、データ列の並進処理が乱されてしまい、プログラムの処理速度を著しく低下させるという問題があった。

【０００７】

なお、前述の特許文献１と特許文献２に記載のＳＩＭＤ方式では、いずれにもこのような問題を解決するための特別な方式については提案されていない。特に、同時並列処理を行う上で操作テーブルを使用する概念は提示されていない。

【０００８】

本発明の目的は、並列処理を維持した状態で高速なデータ演算を実行できる超音波診断装置用プロセッサを提供することにある。

【課題を解決するための手段】

【０００９】

（１）本発明は、超音波の送受信により得られた入力データ列を構成する各入力データを並列処理する超音波診断装置用プロセッサにおいて、前記入力データ列と比較データ列とを比較して、比較結果データ列を生成する第１の演算手段と、前記比較結果データ列を構成する各比較結果データから代表ビットを抽出し、それらの代表ビットによって代表ビット列を生成する代表ビット列抽出手段と、前記代表ビット列が表し得るビットパターンに対応した複数の操作データ列を格納した操作テーブルと、前記複数の操作データ列の中から前記代表ビット列に応じて選択された特定の操作データ列を用いて、前記入力データ列に対するデータ演算を実行し出力データ列を生成する第２の演算手段と、を有することを特徴とする。

【００１０】

上記構成によれば、第１の演算手段において、入力データ列と比較データ列とが並列処理によって比較され比較結果データ列が生成される。そして、その比較結果データ列を構成する各比較結果データから代表ビットが抽出され、それらの代表ビットを並べた代表ビット列が生成される。ここで、代表ビット列によって表し得るビットパターンに対応して、複数の操作データ列が予め準備されており、具体的には、それらの操作データ列は操作テーブルの中に格納されている。そして、代表ビット列が構成するあるビットパターンに応じて、特定の操作データ列が選択される。選択された特定の操作データ列は、第２の演算手段において、入力データ列に対するデータ演算を実行するために用いられる。従って、このような構成においては、第１の演算手段及び第２の演算手段で行われる演算において、演算の並列性を維持したままの状態で、一括のデータ演算を行うことができる。つまり、演算の並列性が乱されることがないので、従来の演算処理と比較して高速な演算処理を行うことができる。また、例えば条件分岐処理を行う場合においても分岐が発生しないので、演算の並列処理を維持しつつ高速な処理速度を実現することができる。ちなみに、前記第１の演算手段と第２の演算手段は、共通の演算部で構成されていてもよい。また、操作テーブルは、プロセッサ内部に備わる内部メモリに構成されていてもよいし、プロセッサの外部に存在する周辺デバイスの記憶部上に構成されてもよい。

【００１１】

望ましくは、前記データ演算は、条件分岐処理に相当する論理演算を含むことを特徴とする。この構成によれば、演算対象であるデータの並列性を担保したままの状態で、論理演算による一括処理で条件分岐処理を行うことができる。すなわち、条件分岐処理を行う場合に、条件分岐命令に代えて、上記のようなテーブルの利用に加えて論理演算が用いられる。

【００１２】

望ましくは、前記データ演算は、更に算術演算を含むことを特徴とする。この構成によ

れば、第２の演算手段において、論理演算と算術演算とを組み合わせることにより、所望の出力データ列を得ることができる。

【００１３】

望ましくは、前記データ演算は、前記入力データ列を構成する各入力データに対して、前記特定の操作データ列を構成する各操作データを作用させて、前記各入力データの保存処理または変更処理を選択的に実行する演算を含むことを特徴とする。この構成によれば操作テーブルに格納されている操作データ列の中から特定の操作データ列として選定された各操作データは、各入力データに対する択一的な処理のための作用を発揮し、演算の同時並列性を維持したまま高速な演算を行うことができる。

【００１４】

望ましくは、前記データ演算は、前記入力データ列を構成する各入力データに対して、前記特定の操作データ列を構成する各操作データに応じて生成された派生操作データを作用させて、前記各入力データの保存処理または変更処理を選択的に実行する演算を含むことを特徴とする。例えば、互いに異なる複数の操作テーブルを使用すると、各々の操作テーブルから特定した操作データ列を相互に作用させることによって、派生操作データ列を生成することができる。派生操作データ列は複数の派生操作データから構成されており、その派生操作データを用いることによって、各入力データに対する択一的な処理のための作用をもって、演算の同時並列性を維持したまま高速な演算を行うことができる。

【００１５】

望ましくは、前記代表ビット列を構成する各代表ビットは、正負を表わす符号ビットであることを特徴とする。この構成によれば、例えば、第１の演算手段における演算操作が差分演算であるような場合に、大小判別のための比較結果が正であるか負であるかを、符号ビット列に基づいて判断することができる。

【００１６】

望ましくは、前記比較データ列を構成する各比較データは、大小判別のための複数の閾値データであることを特徴とする。この構成によれば、第１の演算手段における演算操作は、条件分岐処理の手段としての数値の大小関係を判断する演算手段として用いることができる。

【００１７】

（２）本発明は、超音波の送受信により得られたＮ個（但し、Ｎは２以上の整数）の入力データを並列処理する超音波診断装置用プロセッサにおいて、前記Ｎ個の入力データとＮ個の比較データとを比較してＮ個の比較結果データを生成する第１の演算手段と、前記Ｎ個の比較結果データからＮ個の代表ビットを抽出し、それらの代表ビットによって代表ビット列を生成する代表ビット列抽出手段と、前記代表ビット列が表し得るビットパターンに対応する複数の操作データ列を格納した操作テーブルと、前記複数の操作データ列の中から前記代表ビット列に応じて特定の操作データ列を選択し、この特定の操作データ列を構成するＮ個の操作データを用いて、前記Ｎ個の入力データに対してデータ演算を実行し、Ｎ個の出力データを生成する第２の演算手段と、を有することを特徴とする。Ｎの取り得る値は、２、４、８または１６などの２のべき乗で表される値が好ましいが、それ以外の数値であってもかまわない。ちなみに、Ｎの値は、プロセッサの仕様が許す限り大きな値を選定するのが望ましく、これによって並列処理能力は高くなる。

【００１８】

（３）本発明は、超音波の送受信により得られた入力データ列を構成する各入力データを並列処理する超音波診断装置用プロセッサにおいて、前記超音波診断装置用プロセッサは、演算部と記憶部とを有し、前記演算部は、前記入力データ列と比較データ列との算術演算により比較結果データ列を生成し、前記比較結果データ列を構成する各比較結果データから代表ビットを抽出し、それらの代表ビットによって代表ビット列を生成し、予め用意された前記複数の操作データ列の中から前記代表ビット列に応じて選択された特定の操作データ列を用いて、前記入力データ列に対するデータ演算を実行して出力データ列を生成し、前記記憶部には、前記複数の操作データ列が格納され、前記複数の操作データ列は

、前記代表ビット列が表し得るビットパターンに対応することを特徴とする。上記構成によれば、入力データ列と比較データ列との算術演算により比較結果データ列を生成する機能と、各比較結果データから代表ビットを抽出し代表ビット列を生成する機能と、予め用意された複数の操作データ列の中から選択された特定の操作データ列を用いて、入力データ列に対するデータ演算を実行して出力データ列を生成する機能が、共通の演算部によって達成される。

【発明の効果】

【００１９】

以上説明したように、本発明によれば、演算の並進処理を維持した状態で高速なデータ演算を実行できる。そのために、例えば、条件分岐処理に相当する処理であっても、分岐が発生しないので、入力データ列について処理の流れが乱されることなく、高速な演算処理が可能になる。

【発明を実施するための最良の形態】

【００２０】

以下、本発明の好適な実施の形態を図面に基づいて説明する。

【００２１】

図１は本発明に係るプロセッサが搭載される超音波診断装置の全体的構成を示すブロック図である。図１のシステム制御部３０は、当該超音波診断装置を統括的に制御する役割をもち、システム制御部３０の中には全体制御を行うＣＰＵ３１が搭載されている。また、このシステム制御部３０は図１に示す各ユニットに対して制御タイミングの基本となる信号等を出力している。

【００２２】

送信ビームフォーマー１８は、複数の送信遅延回路を有しており、各々の送信遅延回路の出力信号は、探触子１０の内部にある各々の振動素子に供給される。複数の送信遅延回路は、システム制御部３０から出力された制御タイミングの基本信号に基づいて駆動されており、探触子１０の中に配列された複数の振動素子に対して、遅延時間が制御された送信信号が出力されている。

【００２３】

探触子１０は、多数の振動素子を配列して構成されたアレイ振動子を有している。それぞれの振動素子は、生体に向けてタイミング制御された送信信号により超音波を送信する。生体に向けて送信した超音波は、生体からの反射波として受信される。図１において、この超音波の送受信の状態は１本の超音波ビームＢとして概念的に示されている。この超音波ビームＢを電子的に走査することによりセクタ形状の走査面１４が形成される。この走査面１４はデータ取り込み面であり、振動子からの距離を示すｒと走査角θとを用いてデータの取り込み位置が把握される。本実施形態においてはセクタ走査方式を例示しているが、リニア走査、コンベックス走査、ラジアル走査及びその他の走査方式を採用することもできる。

【００２４】

受信ビームフォーマー１６は、複数のＡ／Ｄ変換回路、複数の受信波位相整合回路、及び加算回路を備える。図示されていない各Ａ／Ｄ変換回路は、各振動素子で受信されたエコー信号をデジタルデータに変換する機能をもつ。また、各受信波位相整合回路は各エコー信号の遅延量を制御して位相を揃える機能をもち、加算回路は位相整合された複数のエコー信号を加算する機能をもつ。探触子１０で得られた複数のエコー信号は、受信ビームフォーマー１６によって整相加算され１つのエコーデータが形成され、それがビーム処理部２０に送られる。

【００２５】

ビーム処理部２０においては、受信ビームフォーマー１６から送られるエコーデータに対して、以下のような信号処理が順次行われる。すなわち、受信した高周波信号の包絡線を検出するための検波処理、信号の輝度を補正するためのダイナミックレンジの圧縮処理などの処理である。このような信号処理を行うために、ビーム処理部には複数のＤＳＰ３

４が搭載されている。各ＤＳＰ３４は、所定のプログラムを実行して信号処理を行う。そして、信号処理が終了したエコーデータは次段の画像処理部２２に送られる。

【００２６】

画像処理部２２においては、エコーデータに対する閾値処理、２値化処理などの変換処理が行われる。画像処理部２２では、当該超音波診断装置の動作モードに応じて様々な処理が行われる場合がある。例えば、Ｂモード画像上でのカラードップラ法による血流画像の表示を行う場合には、画像処理部２２では空間フィルタ処理などが行われる。この画像処理部２２においても複数の画像データ処理用のＤＳＰ３６が使用されている。画像処理部２２から出力されたエコーデータは、次段のデジタルスキャンコンバータ（ＤＳＣ）２４に送られる。

【００２７】

ＤＳＣ２４においては画像データが作成される。ＤＳＣ２４の内部には画像メモリが設けられ、座標変換機能と補間処理機能とを備えている。座標変換処理を行うことによって、送受波座標系で管理されるエコーデータは、表示座標系で管理される画像データに変換される。このような画像データの作製処理にもＤＳＰ３８が使用されている。

【００２８】

ビデオ出力部２６は、ＤＳＣ２４から出力された画像データを、表示部２８で画像表示するためのモニタ信号に変換する。表示部２８としては一般的にＣＲＴやＬＣＤが用いられる。表示部２８には、例えば、Ｂモード表示で、セクタ形状の走査面１４に対応した断層画像が表示される。

【００２９】

操作パネル３２は、キーボードやトラックボール等のデータ入力機器を有している。操作パネルはシステム制御部３０と電気的に接続されており、操作パネル３２から入力されたデータは、システム制御部３０において、そのデータを保存することができる。システム制御部３０に保存されるデータとしては、閾値処理のための閾値や折り返し補正処理のための位相角などがあげられる。

【００３０】

以上、説明したように、超音波診断装置の内部には、本実施形態においてはシステム制御のためのＣＰＵ３１と、エコーデータ処理のための複数のＤＳＰ３４、３６、３８などの複数のプロセッサが使用されている。

【００３１】

図２には、本発明の実施形態である超音波診断装置用プロセッサの概略ブロック図が示されている。プロセッサ４０は、演算器４２と内部メモリ４４を有し、演算器４２は内部メモリ４４を経由してデータバス４６との間でデータを送受する。システム制御部のＣＰＵ３１とＤＳＰ３４，３６，３８は、データバス４６を用いることにより、図示されていない外部メモリを経由して、相互にデータの送受をすることができる。動作の詳細については後述の図５から図２１までの図を用いて説明する。

【００３２】

本発明の実施形態の具体例の説明に入る前に、従来のエコーデータの処理方法を図３を用いて説明する。図３を用いて説明する内容は、前述した従来技術の課題を補足説明するための内容である。

【００３３】

図３に示すＣ言語のプログラム５０は閾値処理のためのプログラムである。これは、演算対象データであるエコーデータと閾値を比較して、閾値以上の値については、エコーデータを無処理のままで保存し、閾値未満の値についてはエコーデータを０にするための処理を記述したものである。比較の対象である閾値ＴＨは一定値である。このプログラム５０の中で用いられるエコーデータは、図４に示すように配列data[i]５４に格納されている。図３のプログラム５０の1行目は、変数ｉを回数カウンタとして、繰り返し処理がＮ回実行されることを示している。２行目は、ｉｆ文の条件分岐処理を示している。演算対象の値data[i]と閾値ＴＨとの大小関係が調べられて、その結果に応じて分岐先が決定さ

れる。もし、あるエコーデータdata[i]の値が閾値ＴＨより大きな値である場合には、プログラム５０の３行目の分岐先に処理が移り、data[i]の値はそのまま配列output[i]に格納される。もし、data[i]の値が閾値ＴＨ以下である場合には、プログラム５０の５行目の分岐先に処理が移り、output[i]には０が格納される。

【００３４】

図３のプログラム５０に示すような閾値処理をＳＩＭＤ方式による並列演算によって行った場合には、上記のような条件分岐先を指定することができないか、あるいは、条件分岐先を指定できたとしても後続の処理が大幅に乱れてしまうという問題が発生する。例えば、４つのエコーデータを並列処理しようとする場合に、１つのケースとして、１つのデータが閾値ＴＨよりも大きく、残りのデータは閾値ＴＨ以下であるような場合は十分に発生しうる。このように、判定条件を満足するデータと満足しないデータが混在してしまうと、分岐先を二者択一で選択する処理を行うことは困難である。判定条件が混在するような事態が発生した場合には、一括で読み込んだ４つのデータの並列性を乱して、１つずつ分解して個別に処理するしかなくなり、結局は、１つずつのデータについて閾値との大小関係をそれぞれ判定し、逐次処理することになる。このようにＳＩＭＤ演算機能を搭載したプロセッサを用いて、並列処理を維持したまま条件分岐処理を実行しようとしても、並列性を維持することができないために高速なデータ処理は望むことはできなかった。そこで、以下に本実施形態を説明する。

【００３５】

図５は、本発明の好適な第１の実施形態の概念図を示したものである。具体的には、エコーデータの大小判断を行うための閾値処理に関する内容を示している。この閾値処理は、図１に記す画像処理部２２の中にあるＤＳＰ３６を用いて実行されるデータ処理である。以下、図５において、閾値処理の説明をデータ処理の流れに沿って詳述する。演算対象の入力データ列７０は、４つの入力データから構成されている。具体的には、図５に示す入力データ列７０は、各１６ビットの４つの入力データ（data[0]、data[1]、data[2]、data[3]）から構成されている。入力データ列７０は、第１演算器７２と第２演算器８４に入力される。ここで、第１演算器７２に入力される比較データ列７４も、４つの比較データから構成されている。第１演算器７２では、入力データ列７０と比較データ列７４とを用いて以下の並列演算が行われる。

【００３６】

図６は、第１演算器７２で行われる演算の詳細を示している。第１演算器７２に入力された入力データ列７０は、比較データ列７４との比較のために減算されて、比較結果データ列８８が得られる。すなわち、ＳＩＭＤ方式の減算操作により、４つの入力データ（data[0]、data[1]、data[2]、data[3]）から、４つの比較データ（TH[0]、TH[1]、TH[2]、TH[3]）を一括で減算することによって、４つの比較結果データ（tmp[0]、tmp[1]、tmp[2]、tmp[3]）が得られる。ここでは減算を用いているが、比較結果データを得るために減算以外の算術演算、論理演算その他の演算を用いてもかまわない。４つの比較結果データは、入力データと比較データとの大小関係を判断するための手段となる。周知のことであるが、２進数表記の数値の最上位ビットは、数値の正負を表す符号ビットとして使用することができる。従って、この比較結果データ列８８についても、各々の比較結果データの最上位ビットである１６ビット目を、各々の符号ビット６８として使用することができる。すなわち、図６において記号Ｓで表す４つの符号ビット（S_0、S_1、S_2、S_3）は、エコーデータと閾値との大小関係に応じて、“１”または“０”のいずれかの値になる（本明細書においては、２進数表記の数値を“　”で括って表記する）。ここでは、具体例として、４つの符号ビットの値が決定し、S_0＝“０”、S_1＝“１”、S_2＝“０”、S_3＝“０”になったとする。この４つの符号ビットを抽出して順に並べると、４ビットの符号ビット列７６が作成される。図６に示す例では、記号ＳＩＧＮで表す符号ビット列７６は、“００１０”となる。本来、符号ビットは、“０”と“１”のいずれかの値を取るので、４ビットのビットパターンで識別できるのは2^4＝１６通りの組み合わせになる。

【００３７】

次に、図５に示した符号ビット列７６と操作テーブルＡ８０について、図７を用いて詳述する。図７においては、符号ビット列７６は、前述した“００１０”のビットパターンとして例示されている。一方、操作テーブルＡ８０は、１６組（１６列）の操作データ列９２から構成されており、それぞれの操作データ列９２は、４つの操作データ９０ａ、９０ｂ、９０ｃ、９０ｄから構成されている。ある符号ビット列７６のビットパターンが決定すると、それに応じて、ある特定の操作データ列８２が決定されることが、図に示す矢印９４によって示されている。

【００３８】

符号ビット列７６を構成する各々の符号ビットの値は、入力データと、閾値の比較データとの大小関係の情報を抽出したものである。閾値処理を実行するためには、符号ビットが“０"であれば、入力データを無変換のまま処理すればよく、逆に、符号ビットが“１"であれば、入力データを０に変換して処理すればよい。これらの変換の必要あるいは不要の選択を行うために、符号ビットの値に応じた操作データとして、0xFFFFあるいは0x0000のいずれかを選択する。0xFFFFと0x0000の閾値処理における選択方法は、以下の２つの方法によって行う。（ｉ）符号ビットが“０”の場合は操作データとして0xFFFFを選択する。（ｉｉ）符号ビットが“１”の場合には操作データとして0x0000を選択する。ちなみに、0xFFFFと0x0000はいずれも１６進数表記の数値である。ここで、例示した符号ビット列７６は、“００１０”であるので、４つの符号ビットS_0＝“０”、S_1＝“１”、S_2＝“０”、S_3＝“０”に応じて選択される４つの操作データを列記すると、0xFFFF、0x0000、0xFFFF、0xFFFFとなる。この４つの操作データは、図７に示した矢印９４で指し示す＃２の位置の操作データ列としてTABLE_A[8]＝0xFFFF、TABLE_A[9]＝0x0000、TABLE_A[10]＝0xFFFF、TABLE_A[11]＝0xFFFFと記されている。この例から分かるように、４つの符号ビットが全て定まれば、それに対応する特定の操作データ列８２の組み合わせも自動的に決定することができる。ここで、符号ビット列７６が取り得るビットパターンは、“００００”から“１１１１”までの１６通りであるので、特定の操作データ列８２は図７に示す＃０から＃１５までの１６組の操作データ列のいずれか一つに該当する。操作テーブルＡ８０は、これらの１６組の操作データ列を集合することで構成されている。更に、操作データ列９２を構成する４つの操作データは、前述の（ｉ）と（ｉｉ）の２つのルールに従って、0xFFFF又は0x0000のいずれかに決定される。符号ビット列７６で取り得る値がどのような値であっても、それに対応する特定の操作データ列８２は操作テーブルＡ８０の中に予め格納してある。なお、操作テーブルＡ８０の実体的な構成は、図２に示すプロセッサ４０内の内部メモリ４４上に構成されるデータ列の集合である。符号ビット列７６のビットパターンに応じて、ある操作データ列９２を特定する際には、プログラム内で管理している相対的なメモリアドレス管理番号を使用してもよいし、メモリに割り当てられているハードウェアのアドレス番号を直接に利用してもよい。図７に示すTABLE_A[i]の記号は、メモリアドレス管理番号を示している。ちなみに、操作テーブルＡ８０は、プロセッサ４０の外部にある周辺デバイスの記憶部内に構成されてもよい。また、不揮発性のメモリに書き込むことによって、消去できない操作テーブルＡ８０を構成することも可能である。

【００３９】

次に、図５に示した特定の操作データ列８２、入力データ列７０及び第２演算器８４の関係を図８に基づいて詳述する。図８は、第２演算器８４で行われる演算の内容を示している。第２演算器８４に入力された入力データ列７０は、特定の操作データ列８２との論理積演算を行うことにより、出力データ列８６に変換される。すなわち、ＳＩＭＤ方式の論理演算により、４つの入力データ（data[0]、data[1]、data[2]、data[3]）に、４つの特定の操作データ（0xFFFF、0x0000、0xFFFF、0xFFFF）を作用させて、一括で論理積を取ることにより、４つの出力データ（out[0]、out[1]、out[2]、out[3]）が得られる。周知のことであるが、論理積演算においては、双方のビット単位の入力値が“１”の場合にだけビット単位の出力値が“１”になる。また、ビット単位の入力値に１つでも“０”がある場合にはビット単位の出力値は“０”となる。本実施形態において閾値処理を行うため

の特定の操作データ列８２を構成する４つの特定の操作データは、0xFFFFまたは0x0000のいずれかの値であり、論理積の演算において、0xFFFFを作用させるということは、演算対象である入力データに対して、全く何の変換処理も行わない操作であることを意味する。つまり、入力データは保存処理されてそのまま出力される。一方、0x0000を作用させるということは、演算対象である入力データを０に置き換える変更処理となる。すなわち、0x0000を作用させることは、ゼロへの置換処理となる。図８に示す具体例では、論理積演算にてdata[1]だけを0x0000と作用させているので、out[1]だけが０になる。その他の３つのデータout[0]、out[2]、out[3]は、それぞれの入力データdata[0]、data[2]、data[3]のそのままの値になる。結果として、第２演算部８４での論理積演算により適正な閾値処理が行われることにより、出力データ列８６が一括で得られる。

【００４０】

以上、図５の概念図に示したように、エコーデータである入力データ列７０が第１演算器７２と第２演算器８４に入力された段階から、第２演算器８４によって出力データ列８６が出力される段階に至るまで、４つのデータは全て並進処理されており、演算の足並みが乱されることはない。このように、データ処理の同時並列性を保つことは、高速な実行速度を保つ上で極めて有効である。

【００４１】

ちなみに、本実施例においては、前述の（ｉ）と（ｉｉ）の２つのルールに従って、符号ビットが“０”の場合は0xFFFFを対応させており、符号ビットが“１”の場合には0x0000を対応させている。しかし、別の適用例としては、符号ビットが“０”の場合は0x0000を対応させて、符号ビットが“１”の場合には0xFFFFを適用させるような他の操作テーブルA_2を閾値処理に使用することも可能である。他の操作テーブルA_2を用いて閾値処理を行う場合には、符号ビット列７６のビットパターンが決定した直後であって、特定の操作データ列８２を決定する前に、そのビットパターンを否定論理演算（ＮＯＴ演算）によって反転させる必要がある。つまり、符号ビット列７６の否定論理演算を追加することにより、他の操作テーブルA_2を用いても正しい閾値処理の結果を得ることができる。ところが、他の操作テーブルA_2を用いると、否定論理演算を実行するための演算の時間が余計に累積されてしまい、処理速度の面で不利な点が生じてしまうことになる。すなわち、演算の高速化を目的として考えると、図示されていない操作テーブルA_2を用いるよりも、図７に示す操作テーブルＡ８０を使用することが望ましい。

【００４２】

次に、参考までに、図９を用いて、図５の概念図に示した閾値処理をプロセッサに実行させる具体的なプログラムの説明を示す。図９に示すプログラム９６は全部で４行の式からなっている。なお、Ｃ言語にはＳＩＭＤ演算を表現する演算子が無いため、データを並列に扱う演算子として暫定的に“ [] ”の記号を使用する。

【００４３】

プログラム９６の１行目に記す式tmp[]＝data[]－TH[]は、ＳＩＭＤ方式による並列の減算処理を示す式であり、第１演算器７２で行う演算に相当する。すなわち、入力データ列data[]を構成する４つの入力データ（data[0]、data[1]、data[2]、data[3]）から、比較データ列TH[]を構成する４つの比較データ（TH[0]、TH[1]、TH[2]、TH[3]）を一括で減算することによって、比較結果データ列tmp[]を構成する４つの比較結果データ（tmp[0]、tmp[1]、tmp[2]、tmp[3]）が得られる。

【００４４】

プログラム９６の２行目に記す式は、比較結果データ列tmp[]から符号ビットＳが抽出されて、符号ビット列ＳＩＧＮが生成される処理を示す式である。本実施形態のＳＩＭＤ処理が可能なプロセッサでは、符号ビットに相当するビットを検出する機能を内蔵している。そのため、２行目の式に示したようなシフト命令や論理積演算等を行うことなく、直接的にＳＩＧＮのビットパターンに相当するデータを導出することができる。従って、符号ビットの抽出処理及び符号ビット列ＳＩＧＮの生成処理において並列演算の足並みを乱すことはない。ちなみに、２行目に記す式は、符号ビットの抽出処理と符号ビット列ＳＩ

ＧＮの生成処理について演算方法を例示したものである。まず（tmp[3]>>12）の部分については、２進数表記のデータとしての比較結果データtmp[3]を右に１２桁シフトさせる。すると、tmp[3]の最上位１６ビット目に位置していた符号ビットの数値は、最下位ビットから数えて４番目のビットに移動される。１２桁右シフトした後の２進数表記データに対して0x8との論理積を行うことにより４桁目のみが“１”か“０”かのビット判定が行われる。他の３つの入力データtmp[2]、tmp[1]、tmp[0]についても同様の処理が行われる。すなわち、tmp[2]の符号ビットは、右に１３桁シフトさせることで３桁目に移動し0x4との論理積を行うことで３桁目のみのビット判定が行われる。tmp[1]の符号ビットは、右に１４桁シフトさせることで２桁目に移動し0x2との論理積を行うことで２桁目のみのビット判定が行われる。tmp[0]の符号ビットは、右に１５桁シフトさせることで１桁目に移動し0x1との論理積を行うことで１桁目のみのビット判定が行われる。４桁目、３桁目、２桁目及び１桁目の各々についてビット判定が行われた４つのビット判定のデータは、４つの論理和を取ることにより１つのデータに集約される。４つの符号ビットを１つのデータに集約することにより、４ビットの符号ビット列ＳＩＧＮのビットパターンが形成される。（実際にはプロセッサ内ではシフト命令や論理積に相当する抽出処理が実行されるだけであり、各データごとの並列性が乱されることはない）なお、正負の判定を行う手段としては、符号ビットが最適であるので利用されるが、抽出されるビットは任意の桁のビットを抽出してもかまわない。例えば、第１演算器７２で算術演算を行った上で、最上位ビット以外のある特定のビットを代表ビットとして抽出し、それらの代表ビットから代表ビット列を集積することも可能である。

【００４５】

プログラム９６の３行目に示す式は、操作データ列が格納された操作テーブルＡ８０の中から１つの操作データ列を特定する処理を示す式である。特定の操作データ列は符号ビット列ＳＩＧＮのビットパターンによって決定される。つまり、符号ビット列ＳＩＧＮのビットパターンに応じて、操作テーブルＡ８０である配列TABLE_A[]の中から１つの操作データ列が特定される。特定された操作データ列は、式の左辺に記す変数としてのＭＡＳＫに代入される。

【００４６】

プログラム９６の４行目に記す式out[]＝data[]＆MASK[]は、ＳＩＭＤ方式による並列の論理積演算を示す式である。MASK[]は特定の操作データ列８２を示しており、３行目の式で決定した変数MASKと同値である。４行目の式によって、入力データ列７０としてのdata[]を構成する４つの入力データ（data[0]、data[1]、data[2]、data[3]）と、特定の操作データ列８２としてのMASK[]との論理積を取ることによって、出力データ列８６としての（out[0]、out[1]、out[2]、out[3]）を得ることができる。

【００４７】

以上のような手順で 閾値処理を行うことにより、４つの入力データを一括して処理することができる。最初の入力データ列（data[0]、data[1]、data[2]、data[3]）の閾値処理が完了した後には、後続の入力データ列である（data[4]、data[5]、data[6]、data[7]）の閾値処理が順次実行されることは言うまでもない。条件分岐命令を使った逐次判定処理を行わないので、判定条件に相当する処理を高速に行うことができ、逐次処理と比較すると数倍の処理速度を期待できる。

【００４８】

次に、本発明の第２の実施形態としてカラードップラ法による血流画像の形成を行う場合における折り返し現象の補正処理について詳述する。カラードップラ法とは、血流で発生する超音波ドップラ効果を利用した血流状態の表示方法である。カラードップラ法は血流を視覚化して表示するための有効な方法であるが、血流速度がサンプリング周期で決定される上限値よりも大きい場合には、血流方向が探触子に近づく方向であるのか、遠ざかる方向であるのか判定できなくなるという側面を有している。これは、偏移周波数がいわゆるナイキスト周波数を越えた場合に発生する折り返し現象として知られており、適切な血流速度の算出を阻害する現象である。そこで、この折り返し速度近傍において、適切な

血流速度の平均値を求めるために、速度データに対しては、折り返し補正処理が行われる。

【００４９】

一般に、カラードップラ法において超音波は生体に対して同一方向に複数回送受信されて、それぞれのエコーデータを基に血流速度の平均値が求められる。この場合に、血流速度の検出上下限が存在するために、血流速度値の中には、折り返したデータが混在する。この血流速度に対して平均化処理などを行う場合、折り返し速度をまたぐデータ同士については、折り返し補正処理をデータ毎に選択的に行う必要がある。以下には、折り返し補正処理に関して、図１０から図１８までを用いて詳述する。

【００５０】

図１０に示す複素平面のグラフ１１４は、超音波ドップラ効果に基づいて得られる血流の速度と血流の方向を図示するための概要図を示したものである。速度情報は複素平面上において位相角θをもつベクトルとして記載することができる。図１０に示す複素平面グラフ１１４上では、２つの速度情報の位相としてθ１（符号１１６）とθ２（符号１１８）が示されている。θ１とθ２のいずれも、探触子に近づく同一方向の血流を観測することで得られた測定値とする。そうすると、θ１もθ２も正の値（例えば、θ１＝１３０°、θ２＝２００°）として測定されるのが適切である。しかしながら、ドップラ法で測定される測定値としては、θ２は負の値（例えばθ２＝－１６０°）として測定されてしまう。これを折り返し現象という。測定値をそのまま用いて、単純な平均値である（θ１＋θ２）／２の計算をすると、θ２が見かけ上、負の値であるので、図１０のグラフ１１４に記すθａのベクトル１２０で示される角度が求められる。しかし、θ２は実際には正の値なので、θａのベクトル１２０の位置は不適切となる。正しい平均値（θ１＋θ２）／２の値は、グラフ１１４に示したθｂのベクトル１２２として表されるのが適正である。

【００５１】

そこで、正しい平均値を求めるために、以下に述べる手順で２段階の処理を行う。まず、第１段階の処理として、演算対象の２つの値の角度差の絶対値を求め、その角度差△θが所定角度１８０°より大きいかどうかを調べる。角度差△θは△θ＝|θ１－θ２|の式で表される。角度差△θが１８０°よりも大きい場合にはベクトルを表す矢印を反転させる必要があると判断し、反対に１８０°以下である場合には反転させないので何の処理も行わない。反転する必要があると判断された場合には、次の第２段階の処理を行う。第２段階の処理においては、ベクトルを表す矢印を反転させるのに、時計周りの回転方向に１８０°に反転するか、それとも反時計方向に１８０°に反転するかを決定する処理を行う。具体的には第２段階の処理においては、単純平均値θavgの計算結果が判断に使用される。単純平均値θavgはθavg＝（θ１＋θ２）／２の式で表される。θavgの値が０以上の値であれば、単純平均した値から１８０°を引く。つまり、時計周りの方向に１８０°回転させることで反転処理を行う。一方、θavgの値が０より小さい値であれば、単純平均した値に１８０°を加える。つまり、反時計方向に１８０°回転させることで反転処理を行う。以上のように、単純平均値θavgと角度差△θを求めて、第１段階と第２段階の処理を併せて行うことによって、折り返し補正処理を行うことができる。この２段階の併合処理自体は公知技術と言えるが、本実施形態ではそれをＳＩＭＤ方式で実現している。

【００５２】

図１１は、カラードップラ法の折り返し補正処理の概念図を示したものである。この処理は図１に記す画像処理部２２の中にあるＤＳＰ３６を用いて実行されるデータ処理である。以下、図１１を用いて、折り返し補正処理の説明をデータ処理の流れに沿って詳述する。演算対象の入力データ列１２４は、４つの入力データave_TH[0]、ave_TH[1]、ave_TH[2]、ave_TH[3]から構成されている。ここで、ave_TH[P]は単純平均値｛θ(2P+1)＋θ(2P+2)｝／２（但し、Ｐは０以上の整数）の値であり、例えばave_TH[0]＝（θ１＋θ２）／２で求められ、ave_TH[1]＝（θ３＋θ４）／２で求められる。更に、θ５及びθ６を用いてave_TH[2]の値が求められ、θ７及びθ８を用いてave_TH[3]の値が求められる。ここでそれぞれのθが取り得る値は単なる例示であり、同じ値が含まれていてもよいし、異な

る値であってもよい。これらの単純平均値ave_TH[P]は、入力データ列１２４に代入される前の段階で予め計算で求められている。入力データ列１２４は、第１演算器１２６と第２演算器１４８に入力される。

【００５３】

図１２を用いて、第１演算器１２６で行われる演算の内容を説明する。第１演算器１２６では、入力データ列１２４から比較データ列１４０を減算することにより比較結果データ列１５２が得られる。折り返し補正処理の実施形態においては、比較データ列１４０の値は全て０であるので、比較結果データ列１５２から抽出した４つの符号ビット（S_0、S_1、S_2、S_3）の各々の値は、各々の単純平均値ave_TH[]の正負を示すことになる。

【００５４】

次に、図１１に示す符号ビット列１２８と操作テーブルＢ１３０との関係について、図１３を用いて説明する。図１３においては、符号ビット列１２８は、“０１０１”を例示のビットパターンとして示している。符号ビット列１２８を構成する各々の符号ビットの値が“０”であれば、単純平均値を示すベクトル１２０は、図１０に示すグラフ１１４の実数軸（横軸）より上方の領域に位置していることを示す。従って、符号ビットが“０”であることは、そのベクトル１２０を反転させようとすると時計回りの方向が適切であることを示している。また、逆に、符号ビットの値が“１”であれば、単純平均値を示すベクトル１２０は、図１０に示すグラフ１１４の実数軸（横軸）より下方の領域に位置していることを示している。従って、符号ビットが“１”であることは、そのベクトル１２０を反転させようとすると反時計回りの方向が適切であることを示している。整理すると、符号ビットが“０”であれば、－１８０°回転の操作データを作用させ、符号ビットが“１”であれば、＋１８０°回転の操作データを作用させればよい。１０進数表記の１８０は１６進数表記で0x00B4であることから、図１３に示す操作デーブルＢ１３０においては、(iii)符号ビットが“０”の場合は0xFF4Cを選択し、（vi）符号ビットが“１”の場合には0x00B4を選択する。操作テーブルＢ１３０に格納される操作データの値は、(iii)と（vi）との２つのルールに従って決定されている。図１３に示すように、例示のビットパターン“０１０１”に対応する特定の操作データ列１３２は、矢印１５４で指し示す＃５の位置の操作データ列１３２であって、TABLE_B[20]＝0xFF4C、TABLE_B[21]＝0x00B4、TABLE_B[22]＝0xFF4C、TABLE_B[23]＝0x00B4となる。特定の操作データ列１３２は、後述の第２演算器１４８での論理演算を行う段階では、OFFSET[]という名称の配列に代入される。符号ビット列１２８のビットパターンがどのような値を取っても、そのビットパターンに対応する特定の操作データ列が操作テーブルＢ１３０の中に予め準備されている。例示した符号ビット列１２８は４ビットであるため、操作テーブルＢ１３０は１６組の操作データ列から構成されている。もし、ＳＩＭＤの演算の並列性が高い場合であって、符号ビット列１２８が８ビットであれば、それに対応して予め準備される操作テーブルＢ１３０は２５６組の操作データ列から構成されることになる。一般的には、代表ビット列がＮ桁である場合には、予め準備される操作テーブルは2^N組の操作データ列から構成される。なお、ここでは角度補正を理解しやすいように±１８０°を整数値で表現したが、実際には精度を確保するため、±１８０°を±１．０に正規化して１６ビット固定小数点で処理する方がよい。

【００５５】

図１１に戻ってこれまでの演算を確認すると、入力データ列１２４、第１演算器１２６、符号ビット列１２８及び操作テーブルＢ１３０に基づいて、特定の操作データ列１３２が決定され、それによって血流の速度を示すベクトル１２０を反転する方向を決定する演算を行ってきた。ところで、ベクトルの回転方向の決定とは別に、反転の必要または不要の判定については別の判定条件を必要とするので、その判定条件を導くための演算を行う必要がある。以下には、図１１に示す参照データ列１５６、第３演算器１３４、符合ビット列１３６及び操作テーブルＣ１３８を用いて決定される反転の必要、不要の判定処理について記す。

【００５６】

演算対象の参照データ列１５６は、４つの入力データdelt_TH[0]、delt_TH[1]、delt_TH[2]、delt_TH[3]から構成されている。ここで、delt_TH[P]は、位相差｛θ(2P+1)－θ(2P+2)｝の絶対値（但し、Ｐは０以上の整数）の値を示しており、例えばdelt_TH[0]＝｜θ１－θ２｜により求められ、delt_TH[1]＝｜θ３－θ４｜で求められる。これらの角度差delt_TH[P]は、参照データ列１５６に代入される前の段階で予め計算で求められている。参照データ列１５６は、第３演算器１３４に入力される。

【００５７】

図１１に示す第３演算器１３４で行われる演算の内容を、図１４を用いて説明する。第３演算器１３４では、参照データ列１５６から比較データ列１５８を減算して比較結果データ列１６０が得られる。反転の要不要の判定が１８０°との大小関係で判定されるので、比較データ列１５８に代入される値は全て１８０となる。つまり、PI[0]=0x00B4、PI[1]=0x00B4、PI[2]=0x00B4、PI[3]=0x00B4となる。比較結果データ列１６０から抽出した４つの符号ビット（S_0、S_1、S_2、S_3）の各々の値は、反転の要不要の判定情報を有している。

【００５８】

次に、符号ビット列１３６と操作テーブルＣ１３８の関係について、図１５を用いて記す。図１５においては、符号ビット列１３６は、“０１１０”を例示のビットパターンとして示している。符号ビット列１３６を構成する各々の符号ビットの値が“０”であれば、角度差の絶対値から１８０°を減算した値が正または０であることを示しており、反転処理が必要であることを意味する。操作テーブルＣ１３８の中から特定される操作データ列１６２は、前述した回転方向を決定するための特定の操作データ列１３２と作用させることを前提としている。従って、反転が必要な場合には、0xFFFFの演算子を選択すればよい。あるいは逆に、符号ビットの値が“１”であれば、角度差の絶対値から１８０°を減算した値が負であることを示しており、反転処理が不要であることを意味する。従って、反転が不要な場合には0x0000の演算子を選択すればよい。図１５に示す操作テーブルＣ１３８においては、(v)符号ビットが“０”の場合は0xFFFFを選択し、（vi）符号ビットが“１”の場合には0x0000を選択する。操作テーブルＣ１３８に格納される全ての操作データの値は、(v)と（vi）との２つのルールに従って決定されている。図１５に示すように、例示のビットパターン“０１１０”に対応する特定の操作データ列１６２は、矢印１６４で指し示す＃６の位置の操作データ列であってTABLE_B[24]＝0xFFFF、TABLE_B[25]＝0x0000、TABLE_B[26]＝0x0000、TABLE_B[27]＝0xFFFFとなる。特定の操作データ列１６２は、後述の第２演算器１４８での論理演算を行う段階では、MASK[]という名称の配列に代入される。ちなみに、反転の要不要を決定する目的である操作テーブルＣ１３８と、閾値処理を行う目的である操作テーブルＡ８０とは、実質的に同じ操作テーブルとなっている。

【００５９】

次に、図１１の第２演算器１４８で行われる演算について記す。第２演算器１４８は、論理演算器１６４と算術演算器１６６の２つの演算器からなっている。図１６には、第２演算器１４８の一部である論理演算器１６４で行われるデータ演算の内容が示されている。論理演算器１６４には、操作テーブルＢ１３０の中から導き出された特定の操作データ列１３２と、操作テーブルＣ１３８の中から導き出された特定の操作データ列１６２と、が入力される。論理演算器１６４では論理積演算が行われて、派生操作データ列１６８が求められる。すなわち論理演算器１６４では、４つの操作テーブルＢ１３０から導出された操作データ（OFFSET[0]、OFFSET[1]、OFFSET[2]、OFFSET[3]）と、４つの特定の操作データ（MASK[0]、MASK[1]、MASK[2]、MASK[3]）との論理積により得られた４つのデータを、再度（OFFSET[0]、OFFSET[1]、OFFSET[2]、OFFSET[3]）に上書きして格納する。特定の操作データ列１３２には、反転の回転方向を指定する＋１８０あるいは－１８０のいずれかの値が格納されており、特定の操作データ列１６２に格納されている0xFFFFあるいは0x0000のいずれかの値と作用して、論理積の演算が行われ、派生操作データ列１６８が得られる。論理積演算が行われることによって、以下に記す３つの処理の中から１つの処理が選択される。つまり、１つは反転が不要な処理であり、もう１つは反転が必要であって反

時計回りに１８０°回転させる処理であり、残りの１つは反転が必要であって時計回りに１８０°回転させる処理である。このように、異なる２つの操作テーブルから導出される２つの操作データ列を相互に作用させることによって、多様な演算を実行することが可能となる。本実施形態においては、反転が不要な処理については、更に２つに細分化する必要がないので３つのオペレーションからなる処理を並列演算で行うことができる。ちなみに、一般的には２つの操作テーブルを用いれば互いに異なる４つの処理を同時並列の処理で行うことができる。得られた演算結果はOFFSET[]に再度格納されて、次段の算術演算器１６６で用いられる。

【００６０】

図１７は、第２演算器１４８の一部である算術演算器１６６で行われるデータ演算の内容を示している。算術演算器１６６には、論理演算器１６４から出力されるOFFSET[]の派生操作データ列１６８と、ave_TH[]の入力データ列１２４とが入力される。算術演算器１６６では入力データ列１２４と、論理演算器１６４の演算結果である派生操作データ列１６８との加算が行われて、出力データ列１７０が得られる。算術演算器１６６で行う操作は、まだ折り返し補正処理が行われていない個々の入力データave_TH[0]、ave_TH[1]、ave_TH[2]、ave_TH[3]を、適切に折り返し補正処理が施されたデータに修正するための操作とみなすことができる。適切なデータに修正するために、前述した３つのオペレーションのいずれか1つの操作が選択的に行われて、出力データ列を構成する各データ（out[0]、out[1]、out[2]、out[3]）が作成される。

【００６１】

以上、図１１の概念図に示すような演算を行うことにより、流速の単純平均値である入力データ列１２４が第１演算器１２６に入力されて符号ビット列１２８が生成され、操作テーブルＢ１３０の中から特定の操作データ列１３２が選択される。また、参照データ列１５６が第３演算器１３４に入力されて符号ビット列１３６が生成され、操作テーブルＣ１３８の中から特定の操作データ列１６２が選択される。そして、第２演算器１４８において、操作データ列１３２と操作データ列１６２との論理演算が行われる。その論理演算で得られた派生操作データ列１６８に対して入力データ列１２４が加算されて、出力データ列１７０が得られる。このように、演算の全ての段階において４つのデータは全て並進して処理が行われており、演算の足並みが乱されることはない。操作テーブルＢ１３０と操作テーブルＣ１３８との２つの操作テーブルを用いることによって、３つの操作から選択的に１つの操作を特定するような処理を行うことができる。

【００６２】

参考までに、次に、図１１の概念図に示した折り返し補正処理を、プロセッサに実行させる具体的なプログラムとして記述する。図１８に示すプログラム１７４が、その折り返し補正処理のためのプログラムであり全部で７行の式からなっている。このプログラム１７４においては、単純平均値のデータ列ave_TH[P]と角度差delt_TH[P]とを演算対象として用いる。ただし、プログラム１７４を実行する前段階において、単純平均値のデータ列ave_TH[P]と角度差delt_TH[P]は既に求められているものとしている。すなわち、単純平均値｛θ(2P+1)＋θ(2P+2)｝／２（但し、Ｐは０以上の整数）の値は、配列ave_TH[］に格納済みであり、角度差の絶対値｜θ(2P+1)－θ(2P+2)｜（但し、Ｐは０以上の整数）の値についても、配列delt_TH[］に格納済みであるとして、それらのデータを用いた演算を開始する部分から説明を記す。

【００６３】

図１８に示すプログラム１７４の１行目に記す式tmp[]＝delt_TH[]－PI[]は、ＳＩＭＤ方式による並列の減算処理を示す式である。図１４に示す第３演算器１３４で行われる比較演算に相当する。すなわち、入力データ列delt_TH[]を構成する４つの入力データ（delt_TH[0]、 delt_TH[1]、 delt_TH[2]、 delt_TH[3]）から、比較データ列PI[]を構成する４つの比較データ（PI[0]、PI[1]、PI[2]、PI[3]）を一括で減算することによって、比較結果データ列tmp[]を構成する４つの比較結果データ（tmp[0]、tmp[1]、tmp[2]、tmp[3]）が得られる。

【００６４】
プログラム１７４の２行目に記す式は、比較結果データ列tmp[]から符号ビットＳが抽出されて、符号ビット列ＳＩＧＮが生成される処理を示す式である。図１４に示す符号ビット列１３６の生成処理に相当する。２行目の式の右辺の処理について記す。まず（tmp[3]>>12）の部分については、比較結果データtmp[3]の２進数表記のデータを右に１２桁シフトさせる。そうすると、tmp[3]の最上位１６ビット目に位置していた符号ビットの数値は、最下位ビットから数えて４番目のビットに移動される。次に１２桁右シフトした後の数値に対して0x8との論理積を行うこと[（tmp[3]>>12）& 0x8]により、４桁目のみが“１”か“０”かのビット判定が行われる。他の３つの入力データtmp[2]、tmp[1]、tmp[0]についても、同様の処理が行われる。すなわち、tmp[2]の符号ビットは、右に１３桁シフトさせることで３桁目に移動し、0x4との論理積を行うことで、３桁目のみが“１”か“０”かのビット判定が行われる。tmp[1]の符号ビットは、右に１４桁シフトさせることで２桁目に移動し、0x2との論理積を行うことで、２桁目のみが“１”か“０”かのビット判定が行われる。tmp[0]の符号ビットは、右に１５桁シフトさせることで１桁目に移動し、0x1との論理積を行うことで、１桁目のみが“１”か“０”かのビット判定が行われる。４桁目、３桁目、２桁目及び１桁目の合計４つのビット判定のデータは、４つの論理和を取ることにより、１つのデータに集約される。２行目に示す式の中の＆記号は論理積を、｜記号は論理和の演算を表す。４つの符号ビットを１つのデータに集約することにより符号ビット列ＳＩＧＮのビットパターンが形成される。

【００６５】
プログラム１７４の３行目に示す式は、操作データ列が格納された操作テーブルＣ１３８の中から、１つの操作データ列を特定する処理を示す式である。図１５に示すような、特定の操作テーブル１６２を特定する処理に相当する。符号ビット列１３６の操作テーブルＣ１３８の符号ビット列ＳＩＧＮのビットパターンに応じて、操作テーブルＣ１３８の配列TABLE_C[]の中から１つの操作データ列が特定され、ＭＡＳＫという特定された操作データ列として決定される。図１５に示すように、符号ビット列１３６がＳＩＧＮ＝“０１１０”であった場合には、TABLE_C[24]=0xFFFF，TABLE_C[25]=0x0000，TABLE_C[26]=0x0000，TABLE_C[27]=0xFFFFの４つのデータを、特定の操作データとして使用する。図１８のプログラムにおいて特定の操作データ列は、ＭＡＳＫという名称で指定される。

【００６６】
プログラム１７４の４行目に記す式は、入力データ列ave_TH[]から符号ビットＳが抽出されて、符号ビット列ＳＩＧＮ２が生成される処理を示す式である。４行目の式の右辺の処理について記すと、まず（ave_TH[3]>>12）の部分については、入力データave_TH[3]の２進数表記のデータとして右に１２桁シフトさせる。そうすると、ave_TH[3]の最上位１６ビット目に位置していた符号ビットの数値は、最下位ビットから数えて４番目のビットに移動される。次に１２桁右シフトした後の数値に対して0x8との論理積を行うこと｛（ave_TH[3]>>12）&0x8｝により、４桁目のみが“１”か“０”かのビット判定が行われる。他の３つの入力データave_TH[2]、ave_TH[1]、ave_TH [0]についても、同様の処理が行われる。すなわち、ave_TH [2]の符号ビットは、右に１３桁シフトさせることで３桁目に移動し、0x4との論理積を行うことにより３桁目のみが“１”か“０”かのビット判定が行われる。ave_TH[1]の符号ビットは、右に１４桁シフトさせることで２桁目に移動し、0x2との論理積を行うことにより２桁目のみが“１”か“０”かのビット判定が行われる。ave_TH [0]の符号ビットは、右に１５桁シフトさせることで１桁目に移動し、0x1との論理積を行うことにより１桁目のみが“１”か“０”かのビット判定が行われる。４桁目、３桁目、２桁目及び１桁目の合計４つのビット判定のデータは、４つの論理和を取ることにより、１つのデータに集約される。４つの符号ビットを１つのデータに集約することにより符号ビット列ＳＩＧＮ２のビットパターンが形成される。

【００６７】
５行目に示す式は、操作データ列が格納された操作テーブルＢ１３０の中から、１つの操作データ列を特定する処理を示す式である。符号ビット列ＳＩＧＮ２のビットパターン

に応じて、操作テーブルＢ１３０の配列TABLE_B[]の中から１つの操作データ列が特定され、OFFSETという名称の特定の操作データ列として決定される。具体的な数値を用いると、例えば、符号ビット列ＳＩＧＮ２が“０１０１”であった場合には、TABLE_B[20]=0xFF4C，TABLE_B[21]=0x00B4，TABLE_B[22]= 0xFF4C，TABLE_B[23]= 0x00B4の４つのデータを、特定の操作データとして使用する。図１８のプログラムにおいて特定の操作データ列１３２は、OFFSETという名称で指定される。

【００６８】

６行目に記す式OFFSET[]＝OFFSET[]＆MASK[]は、ＳＩＭＤ方式による並列の論理積演算を示す式である。MASK[]は特定の操作データ列１６２を示しており、３行目の式で決定した変数MASKと同値である。４行目の式によって、操作データ列１３２を構成する４つの入力データ（OFFSET[0]、OFFSET[1]、OFFSET[2]、OFFSET[3]）と、特定の操作データ列１６２としてのMASK[]との論理積を取ることによって、派生操作データ列としての（OFFSET[0]、OFFSET[1]、OFFSET[2]、OFFSET [3]）を得ることができる。

【００６９】

７行目に記す式out[]＝ave_TH[]＋OFFSET[]は、ＳＩＭＤ方式による並列の加算演算を示す式である。まだ折り返し補正処理がなされていない入力データ列ave_TH[]に対して、派生操作データ列OFFSET[]を加算することによって、各々のデータに対して選択的に適切な折り返し補正処理がなされて、結果がout[]に格納される。

【００７０】

次に、変形例としてエコーデータの２値化処理について詳述する。本明細書に記す２値化処理とは、画像を構成する画素の輝度値が判定基準値より大きければ１に、小さければ０にする処理のことを示す。具体的な適用例としては、心臓の断層面に係るエコーデータを処理する場合に生体組織部である心筋と、心腔との境界を検出するような場合に適用できる。この２値化処理は、図１に記す画像処理部２２の中にあるＤＳＰ３６によって行われるデータ処理である。

【００７１】

以下、図１９を用いて、２値化処理の演算の流れに沿って詳述する。演算対象の入力データ列１７８は、４つの入力データから構成されており、第１演算器１８０に入力される。４つの比較データから構成される比較データ列１８２も、第１演算器１８０に入力される。比較データ列１８２に格納されている値は、例えば、画素の輝度値の明暗を判定するための判定基準値である。第１演算器１８０では、入力データ列１７８から比較データ列１８２を引く減算が行われて、比較結果データ列が生成される。従って、比較結果データ列から抽出した４つの符号ビット（S_0、S_1、S_2、S_3）の各々の値は、判定基準値との大小関係の判定結果の情報を有している。符号ビット列１８４は、４つの符号ビットを集合したものである。

【００７２】

次に、図１９に示した符号ビット列１８４と操作テーブルＤ１８６について、図２０を用いて詳述する。図２０に示す符号ビット列１８４は、“００１０”のビットパターンとして例示されている。一方、図２０に示す操作テーブルＤ１８６は、１６組（１６列）の操作データ列から構成されており、それぞれの操作データ列は４つの操作データから構成されている。符号ビット列１８４のビットパターンが決定すると、それに応じて１６組の中から、ある特定の操作データ列１８８が決定されることが、図２０に示す矢印１９０によって示されている。

【００７３】

符号ビット列１８４を構成する各々の符号ビットの値は、入力データと、閾値の比較データとの大小関係の情報を抽出したものである。従って、２値化処理を実行するためには、符号ビットが“０”であれば、入力データを１に置換処理すればよい。逆に、符号ビットが“１”であれば、入力データを０に置換処理すればよい。これらの置換処理の必要、不要の選択を行うために、符号ビットの値に応じた操作データとして、0x0001あるいは0x0000のいずれかを選択する。0x0001と0x0000の２値化処理における選択方法は、以下の２

つの方法によって行う。（vii）符号ビットが“０”の場合は0x0001を選択する。（viii）符号ビットが“１”の場合には0x0000を選択する。ここで、例示した符号ビット列１８４は、“００１０”であるので、４つの符号ビットS_0＝“０”、S_1＝“１”、S_2＝“０”、S_3＝“０”に応じて選択される４つの操作データを列記すると、0x0001、0x0000、0x0001、0x0001となる。この４つの操作データは、図２０に示した矢印１９０で指し示す＃２の位置の操作データ列としてTABLE_D[8]＝0x0001、TABLE_D[9]＝0x0000、TABLE_D[10]＝0x0001、TABLE_D[11]＝0x0001と記されている。この例から分かるように、４つの符号ビットが全て定まれば、それに対応する特定の操作データ列１８８の組み合わせも自動的に決定することができる。操作テーブルＤ１８６は、この１６組の操作データ列を集合することで構成されている。更に、操作データ列を構成する４つの操作データは、前述の（vii）と（viii）の２つのルールに従って、0x0001又は0x0000のいずれかが決定される。符号ビット列１８４で取り得る値がどのような値であっても、それに対応する特定の操作データ列１８８は操作テーブルＤ１８６の中に予め格納してある。

【００７４】

図１９に示すように、特定された操作データ列１８８はそのまま出力データ列１９２として使用することができる。操作テーブルＤ１８６には、２種類の値を予め格納しておくことができるので、２値化処理の場合には、第２演算器を特には必要としない。

【００７５】

図２１に示すプログラム１９４は、２値化処理のためのプログラムを例示したものである。プログラム１９４の１行目及び２行目で行われる処理は、図９に示す閾値処理のプログラム９６の１行目及び２行目で行う処理と実質的に同じである。すなわち、プログラム１９４の１行目に記す処理によりＳＩＭＤ方式による並列の減算処理が行われる。そして、プログラム１９４の２行目に記す処理によって、ＳＩＧＮで表される４ビットの符号ビット列１８４が作成される。プログラム１９４の３行目に示す式は、複数の操作データ列が格納された操作テーブルＤ１８６の中から、１つの操作データ列を特定する処理を示す式である。つまり、プログラム１９４の２行目に示すＳＩＧＮのビットパターンに応じて、図２０に示す操作テーブルＤ１８６の配列TABLE_D[]の中から１つの操作データ列が特定される。そして、その特定の操作データ列１８８の値は、配列out[]にそのまま代入される。

【００７６】

以上のように、２値化処理を行う場合にも、２値化処理のための操作テーブルＤ１８６を用いることによって、４つの入力データを一括して処理することが可能となる。但し、図１９の概念図に示す２値化処理を行う場合は、図５及び図１１等に示した実施形態とは異なり、第２演算器に相当するものは含まれていない。

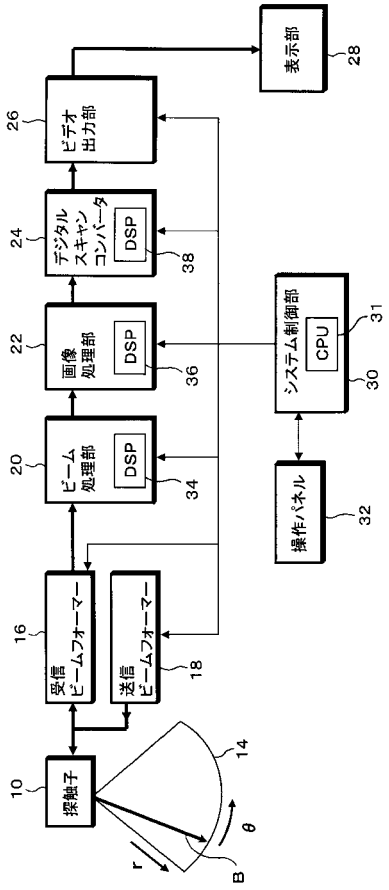
【００７７】

これまで詳述した閾値処理、カラードップラ法による折り返し補正処理及び２値化処理の説明においては、最大のデータバス幅が６４ビットの演算器を用いるものであった。そして、１つのデータ長を１６ビットとし、１６ビット×４データの並列演算が可能なＳＩＭＤ方式の演算器での例を示した。しかし、ＳＩＭＤ方式のプロセッサで行える演算は、１６ビット×４データの場合に限定されるものではない。プロセッサの仕様によっては、３２ビット、１２８ビットもしくは２５６ビットのデータバス幅を有する場合があり、それらのデータバス幅を、２分割、４分割、８分割又は１６分割等のいずれかの分割値の中から１つを選択して並列処理を実行させることも可能である。並列処理を行う場合に、同時に処理できるデータ数はＮ個（但し、Ｎは２以上の整数）となる。その場合には、入力データ列、比較データ列、比較結果データ列、特定の操作データ列、出力データ列は全てＮ個のデータから構成される。なお、代表ビット列の桁数もＮ桁となる。これらのデータ列の個数あるいは桁数に関してＮという数が一致していることは、演算の過程で一貫して並列処理が行われていることを示している。なお、ＳＩＭＤ方式の演算器を複数個備えている場合には、高速化処理の性能改善度は更に大きくなる。

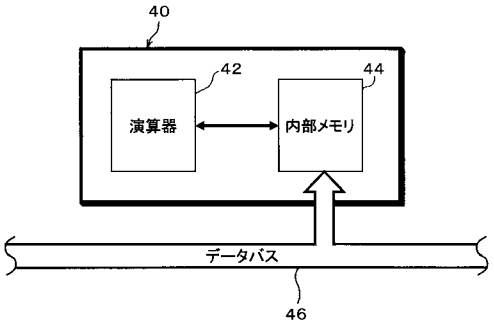
【図面の簡単な説明】

【００７８】
【図１】本発明に係るプロセッサが搭載される超音波診断装置の全体的構成を示すブロック図である。
【図２】本発明の実施形態である超音波診断装置用プロセッサの概略ブロック図である。
【図３】従来技術によるエコーデータの処理方法を示すプログラム例を示す図である。
【図４】演算対象のデータを格納する配列data[i]の概要図である。
【図５】本発明の好適な実施形態の概念図を示した図である。
【図６】第１演算器で行われる演算の概要説明図である。
【図７】符号ビット列と操作テーブルＡの対応関係を例示する図である。
【図８】第２演算器で行われる演算の概要説明図である。
【図９】図５の概念図に示した閾値処理を、プロセッサに実行させるためのプログラム例を示す図である。
【図１０】超音波ドップラ効果に基づいて得られる血液情報の速度と向きを図示する複素平面のグラフを示す図である。
【図１１】カラードップラ法の折り返し補正処理の演算の概念図を示した図である。
【図１２】第１演算器で行われる演算の概要説明図である。
【図１３】符号ビット列と操作テーブルＢの対応関係を例示する図である。
【図１４】第３演算器で行われる演算の概要説明図である。
【図１５】符号ビット列と操作テーブルＣの対応関係を例示する図である。
【図１６】第２演算器の一部である論理演算器で行われるデータ演算の概念説明図である。
【図１７】第２演算器の一部である算術演算器で行われるデータ演算の概要説明図である。
【図１８】図１１の概念図に示した折り返し補正処理を、プロセッサに実行させるためのプログラム例を示す図である。
【図１９】２値化処理の演算の概念図を示した図である。
【図２０】符号ビット列と操作テーブルＤの対応関係を例示する図である。
【図２１】図１９の概念図に示した２値化処理を、プロセッサに実行させるためのプログラム例を示す図である。
【符号の説明】
【００７９】
７０　入力データ列、７２　第１演算器、７４　比較データ列、７６　符号ビット列、８０　操作テーブルＡ、８２　特定の操作データ列、８４　第２演算器、８６　出力データ列。

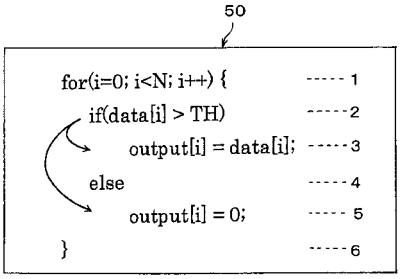
【図１】


表示部
28
ビデオ出力部
26
デジタルスキャンコンバータ
DSP
24
38
画像処理部
DSP
22
36
ビーム処理部
DSP
20
34
システム制御部
CPU
31
30
操作パネル
32
受信ビームフォーマー
送信ビームフォーマー
16
18
探触子
10
14
θ
r
B

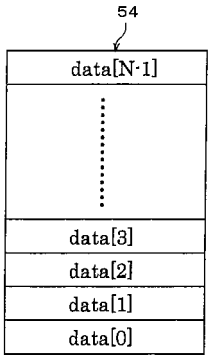
【図２】


40
演算器
42
内部メモリ
44
データバス
46

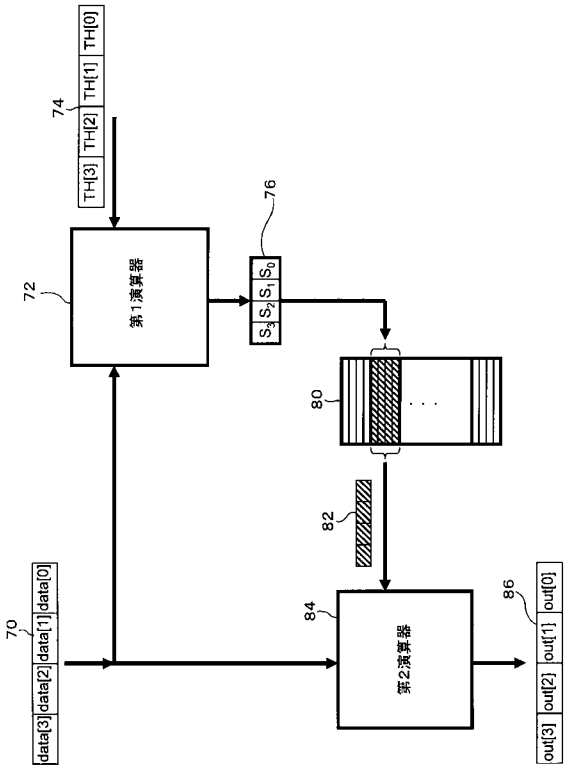
【図３】


50
for(i=0; i<N; i++) {
if(data[i] > TH)
output[i] = data[i];
else
output[i] = 0;
}
1
2
3
4
5
6

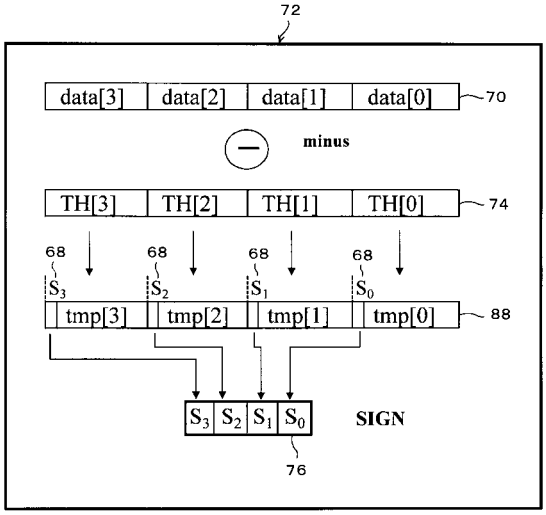
【図４】


54
data[N-1]
data[3]
data[2]
data[1]
data[0]

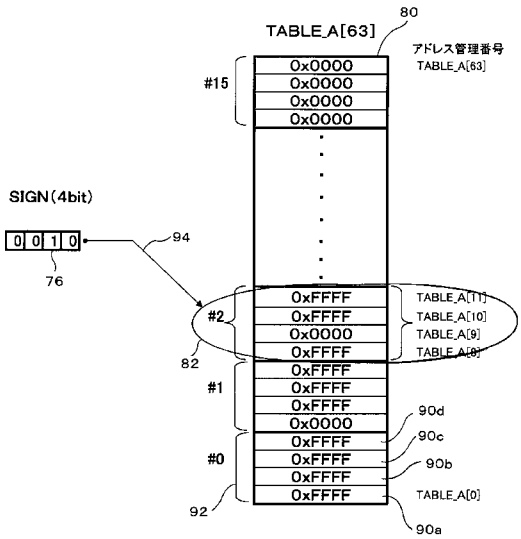
【図５】


74
TH[3] TH[2] TH[1] TH[0]
72
第1演算器
76
S3 S2 S1 S0
80
82
70
data[3] data[2] data[1] data[0]
84
第2演算器
86
out[3] out[2] out[1] out[0]

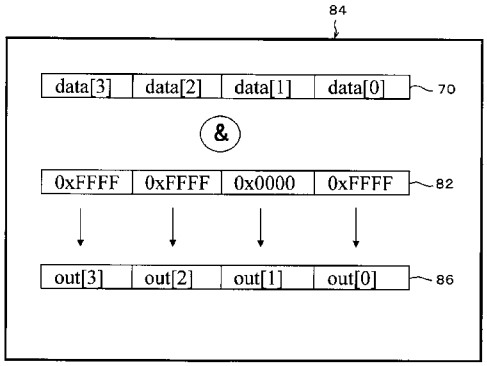
【図６】



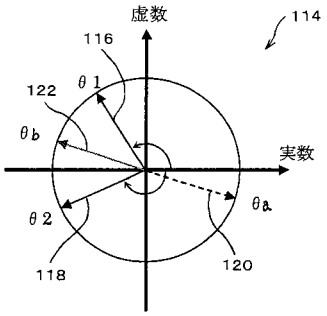
【図７】



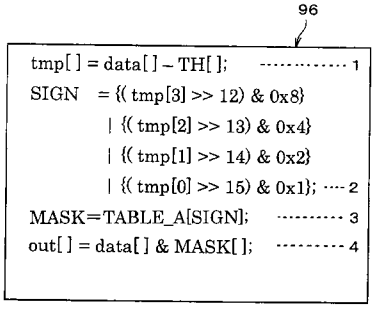
【図８】



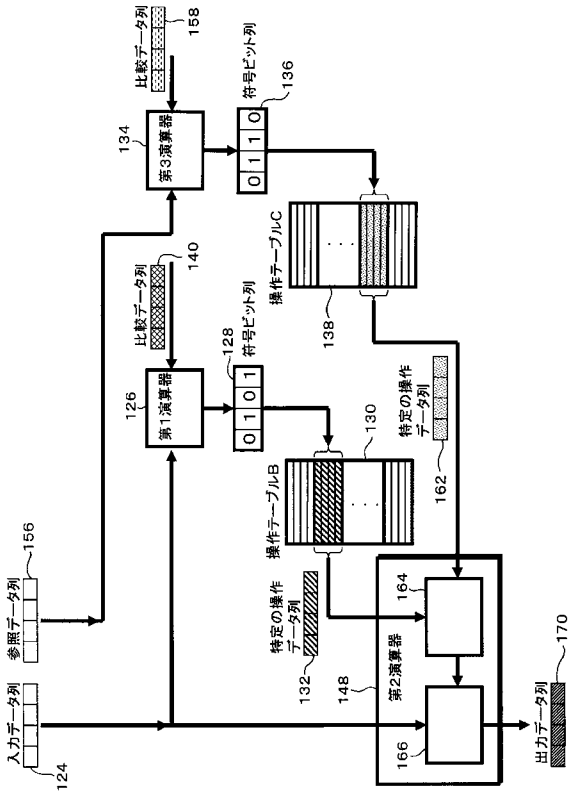
【図１０】



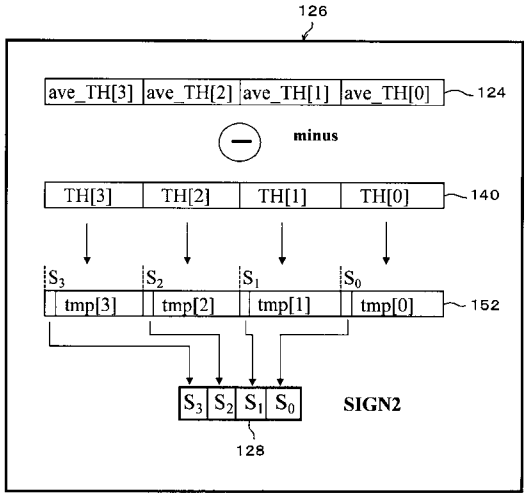
【図９】



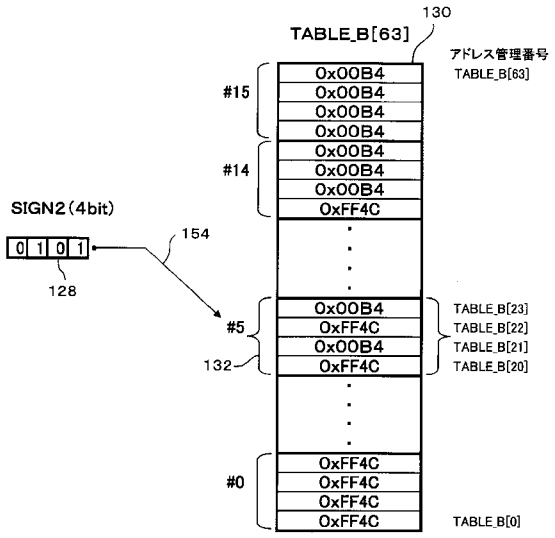
【図１１】



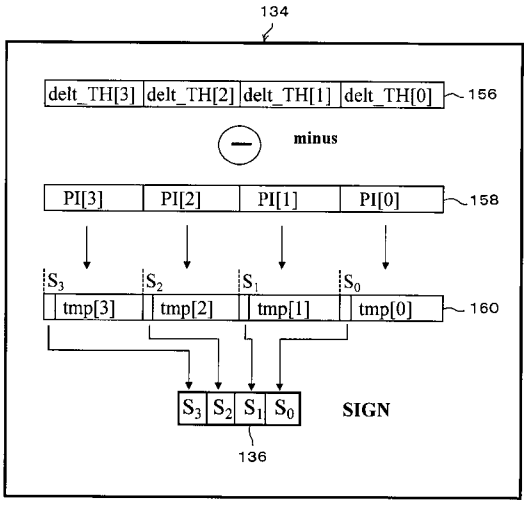
【図１２】



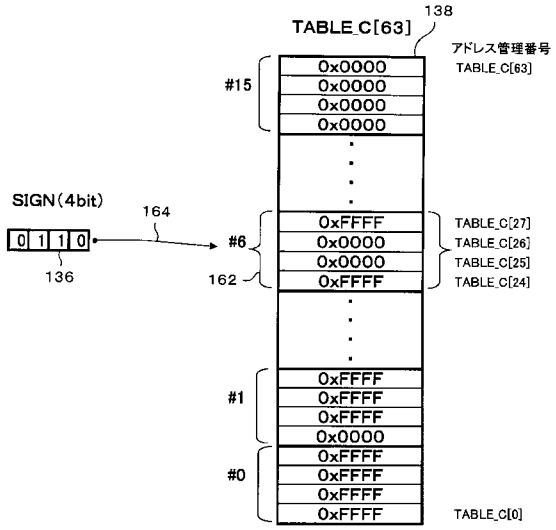
【図１３】



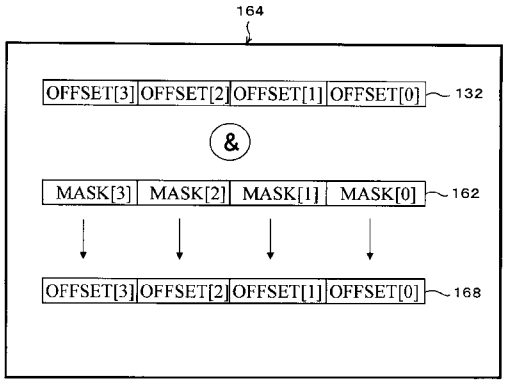
【図１４】



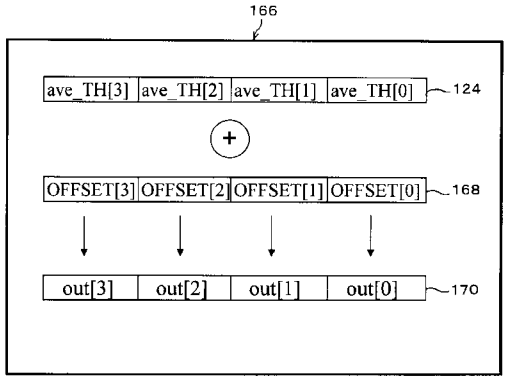
【図１５】



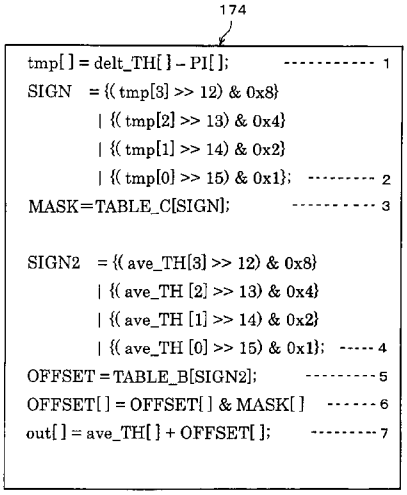
【図１６】



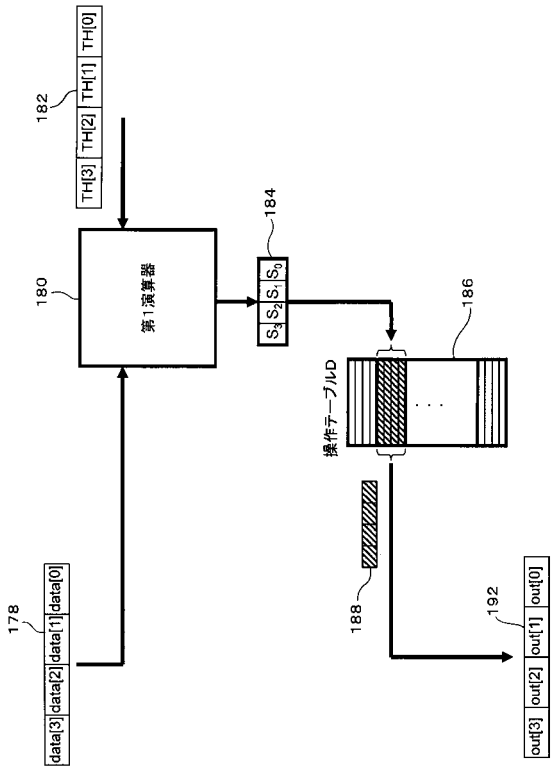
【図１７】



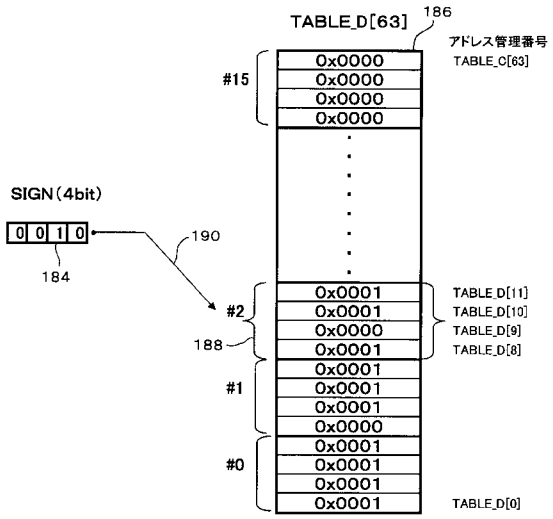
【図１８】



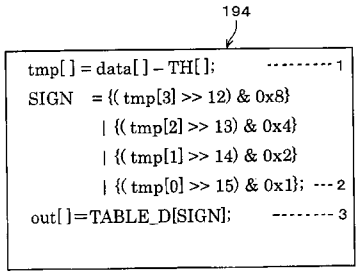
【図１９】



【図２０】


TABLE_D[63]
186
アドレス管理番号
TABLE_C[63]
#15
0x0000
0x0000
0x0000
0x0000
SIGN(4bit)
0 0 1 0
190
184
#2
188
0x0001
0x0001
0x0000
0x0001
TABLE_D[11]
TABLE_D[10]
TABLE_D[9]
TABLE_D[8]
#1
0x0001
0x0001
0x0001
0x0000
#0
0x0001
0x0001
0x0001
0x0001
TABLE_D[0]

【図２１】


194
tmp[ ] = data[ ] – TH[ ]; ---------1
SIGN = {(tmp[3] >> 12) & 0x8}
| {(tmp[2] >> 13) & 0x4}
| {(tmp[1] >> 14) & 0x2}
| {(tmp[0] >> 15) & 0x1}; ---2
out[ ]=TABLE_D[SIGN]; --------3

フロントページの続き

(56)参考文献 特開平5－189585（ＪＰ，Ａ）
特開平11－53189（ＪＰ，Ａ）
特開2000－47998（ＪＰ，Ａ）
特開2000－189422（ＪＰ，Ａ）
特開2001－229135（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ａ６１Ｂ　　８／００
Ｇ０６Ｆ　１５／８０

专利名称(译)	超声波诊断设备的处理器		
公开(公告)号	JP4716911B2	公开(公告)日	2011-07-06
申请号	JP2006100226	申请日	2006-03-31
[标]申请(专利权)人(译)	日立阿洛卡医疗株式会社		
申请(专利权)人(译)	阿洛卡有限公司		
当前申请(专利权)人(译)	日立アロカメディカル株式会社		
[标]发明人	尾形太		
发明人	尾形 太		
IPC分类号	A61B8/00 G06F15/80		
FI分类号	A61B8/00 G06F15/80		
F-TERM分类号	4C601/EE07 4C601/JB60		
代理人(译)	吉田健治 石田 纯		
其他公开文献	JP2007268156A		
外部链接	Espacenet		

摘要(译)

要解决的问题：当使用能够并行处理多个数据的SIMD型算术单元时，提供能够执行高速算术运算而不执行由if语句表示的条件分支处理的超声诊断设备。可能是。 解决方案：用于比较输入数据串70和比较数据序列74的第一算术单元72产生代表性位串76。并且确定与代表性位串76可以表示的位模式相对应的特定操作数据串82。特定操作数据串82对第二计算单元84中的输入数据串70进行操作，并获得输出数据串86。由于在由多个操作数据串构成的操作表A80中预先准备和准备特定操作数据串82，因此可以在保持并行处理的同时高速执行与条件分支对应的计算。 点域5

【图5】