

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2007-268156

(P2007-268156A)

(43) 公開日 平成19年10月18日(2007.10.18)

(51) Int.Cl.

A61B 8/00 (2006.01)
G06F 15/80 (2006.01)

F I

A61B 8/00
 G06F 15/80

テーマコード (参考)

4C601

審査請求 未請求 請求項の数 9 O L (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)

(71) 出願人 390029791
 アロカ株式会社
 東京都三鷹市牟礼6丁目2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2番1号 アロ
 カ株式会社内
 Fターム(参考) 4C601 EE07 JB60

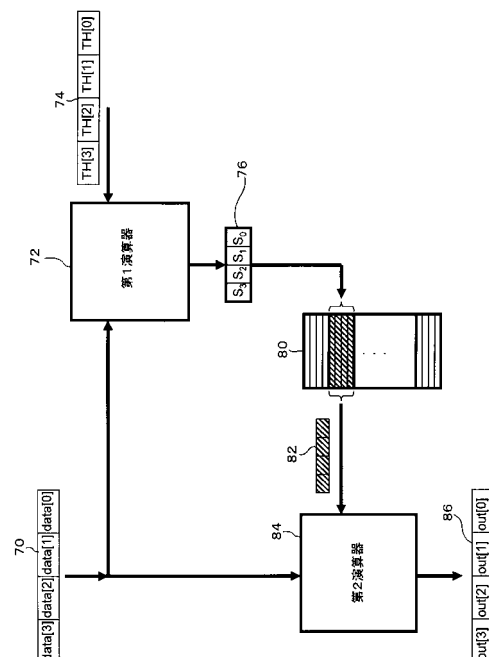
(54) 【発明の名称】 超音波診断装置用プロセッサ

(57) 【要約】

【課題】超音波診断装置において、複数のデータを並列処理することが可能なSIMD方式の演算器を用いる場合に、if文に代表されるような条件分岐処理を実行しないで、高速な演算処理を可能とする。

【解決手段】入力データ列70と比較データ列74とを比較する第1演算器72によって、代表ビット列76を生成する。その代表ビット列76が表し得るビットパターンに対応して、特定の操作データ列82を決定する。特定の操作データ列82は、第2演算器84において入力データ列70と作用し、出力データ列86が得られる。特定の操作データ列82は、複数の操作データ列によって構成される操作テーブルA80に予め格納して準備されているので、並列処理を維持したまま条件分岐に相当する演算を高速に実行できる。

【選択図】 図5



【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、

10

20

30

40

50

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きている。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きている。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

10

20

30

40

50

なるので、処理速度を高速化できる。

【 0 0 0 4 】

以下、特許文献 1 には S I M D 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 S I M D 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。

【 0 0 0 5 】

【特許文献 1】特開 2 0 0 0 - 4 7 9 9 8 号公報

【特許文献 2】特開 2 0 0 0 - 1 8 9 4 2 2 号公報

【発明の開示】

【発明が解決しようとする課題】

10

【 0 0 0 6 】

背景技術として示したように、S I M D の機能を搭載した C P U や D S P 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、S I M D 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、i f 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。S I M D 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 S I M D 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、S I M D 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、S I M D 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。

20

【 0 0 0 7 】

なお、前述の特許文献 1 と特許文献 2 に記載の S I M D 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。

30

【 0 0 0 8 】

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】

【 0 0 0 9 】

(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。

40

【 0 0 1 0 】

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

50

、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、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。

10

【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、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。

20

【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。

30

【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。

40

【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

50

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。 10

【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。 20

【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。 30

【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。 40

【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 50

。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 1 4 が形成される。この走査面 1 4 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。

【 0 0 2 4 】

受信ビームフォーマー 1 6 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 1 0 で得られた複数のエコー信号は、受信ビームフォーマー 1 6 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 2 0 に送られる。

10

【 0 0 2 5 】

ビーム処理部 2 0 においては、受信ビームフォーマー 1 6 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 DSP 3 4 が搭載されている。各 DSP 3 4 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 2 2 に送られる。

20

【 0 0 2 6 】

画像処理部 2 2 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 2 2 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 Doppler 法による血流画像の表示を行う場合には、画像処理部 2 2 では空間フィルタ処理などが行われる。この画像処理部 2 2 においても複数の画像データ処理用の DSP 3 6 が使用されている。画像処理部 2 2 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 (D S C) 2 4 に送られる。

30

【 0 0 2 7 】

D S C 2 4 においては画像データが作成される。D S C 2 4 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 DSP 3 8 が使用されている。

【 0 0 2 8 】

ビデオ出力部 2 6 は、D S C 2 4 から出力された画像データを、表示部 2 8 で画像表示するためのモニタ信号に変換する。表示部 2 8 としては一般的に CRT や LCD が用いられる。表示部 2 8 には、例えば、B モード表示で、セクタ形状の走査面 1 4 に対応した断層画像が表示される。

40

【 0 0 2 9 】

操作パネル 3 2 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 3 0 と電氣的に接続されており、操作パネル 3 2 から入力されたデータは、システム制御部 3 0 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 3 0 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。

【 0 0 3 0 】

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 CPU 3 1 と、エコーデータ処理のための複数の DSP 3 4 、3 6 、3 8 などの複数のプロセッサが使用されている。

50

【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。

10

【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。

20

【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。

30

【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。

40

【0036】

50

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もので、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A 8 0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A 8 0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A 8 0を構成することも可能である。

【0039】

10

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、一括で論理積をとることにより、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。

20

30

【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A₂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A₂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A₂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A₂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A₂を用いるよりも、図7に示す操作テーブルA 8 0を使用することが望ましい。

40

50

【 0 0 4 2 】

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 9 6 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。

【 0 0 4 3 】

プログラム 9 6 の 1 行目に記す式 $\text{tmp}[] = \text{data}[] - \text{TH}[]$ 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 7 2 で行う演算に相当する。すなわち、入力データ列 $\text{data}[]$ を構成する 4 つの入力データ ($\text{data}[0]$ 、 $\text{data}[1]$ 、 $\text{data}[2]$ 、 $\text{data}[3]$) から、比較データ列 $\text{TH}[]$ を構成する 4 つの比較データ ($\text{TH}[0]$ 、 $\text{TH}[1]$ 、 $\text{TH}[2]$ 、 $\text{TH}[3]$) を一括で減算することによって、比較結果データ列 $\text{tmp}[]$ を構成する 4 つの比較結果データ ($\text{tmp}[0]$ 、 $\text{tmp}[1]$ 、 $\text{tmp}[2]$ 、 $\text{tmp}[3]$) が得られる。

10

【 0 0 4 4 】

プログラム 9 6 の 2 行目に記す式は、比較結果データ列 $\text{tmp}[]$ 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 $SIGN$ 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 ($\text{tmp}[3] \gg 12$) 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 $\text{tmp}[3]$ を右に 12 桁シフトさせる。すると、 $\text{tmp}[3]$ 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 $\text{tmp}[2]$ 、 $\text{tmp}[1]$ 、 $\text{tmp}[0]$ についても同様の処理が行われる。すなわち、 $\text{tmp}[2]$ 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。 $\text{tmp}[1]$ 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。 $\text{tmp}[0]$ 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で 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 7 2 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。

20

30

40

【 0 0 4 5 】

プログラム 9 6 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 $\text{TABLE_A}[]$ 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。

【 0 0 4 6 】

プログラム 9 6 の 4 行目に記す式 $\text{out}[] = \text{data}[] \& \text{MASK}[]$ は、SIMD 方式による並列の論理積演算を示す式である。 $\text{MASK}[]$ は特定の操作データ列 8 2 を示しており、3 行目の

50

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。 10

【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。 20

【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 θ_1 (符号116)と θ_2 (符号118)が示されている。 θ_1 と θ_2 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、 θ_1 も θ_2 も正の値(例えば、 $\theta_1 = 130^\circ$ 、 $\theta_2 = 200^\circ$)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、 θ_2 は負の値(例えば $\theta_2 = -160^\circ$)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 $(\theta_1 + \theta_2) / 2$ の計算をすると、 θ_2 が見かけ上、負の値であるので、図10のグラフ114に記す θ_a のベクトル120で示される角度が求められる。しかし、 θ_2 は実際には正の値なので、 θ_a のベクトル120の位置は不適切となる。正しい平均値 $(\theta_1 + \theta_2) / 2$ の値は、グラフ114に示した θ_b のベクトル122として表されるのが適正である。 40

【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 $\theta = |\theta_1 - \theta_2|$ 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 50

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 θ_{avg} の計算結果が判断に使用される。単純平均値 θ_{avg} は $\theta_{avg} = (\theta_1 + \theta_2) / 2$ の式で表される。 θ_{avg} 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、 θ_{avg} 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 θ_{avg} 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。

10

【0052】

図11は、カラードブ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 $\{ \theta(2P+1) + \theta(2P+2) \} / 2$ (但し、 P は0以上の整数) の値であり、例えば $ave_TH[0] = (\theta_1 + \theta_2) / 2$ で求められ、 $ave_TH[1] = (\theta_3 + \theta_4) / 2$ で求められる。更に、 θ_5 及び θ_6 を用いて $ave_TH[2]$ の値が求められ、 θ_7 及び θ_8 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。

20

【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】

30

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き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き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、 $+180^\circ$ 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に示す操作テーブル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

40

50

。符号ビット列 1 2 8 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 B 1 3 0 の中に予め準備されている。例示した符号ビット列 1 2 8 は 4 ビットであるため、操作テーブル B 1 3 0 は 1 6 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 1 2 8 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 B 1 3 0 は 2 5 6 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 $\pm 180^\circ$ を整数値で表現したが、実際には精度を確保するため、 $\pm 180^\circ$ を ± 1.0 に正規化して 1 6 ビット固定小数点で処理する方がよい。

10

【 0 0 5 5 】

図 1 1 に戻ってこれまでの演算を確認すると、入力データ列 1 2 4、第 1 演算器 1 2 6、符号ビット列 1 2 8 及び操作テーブル B 1 3 0 に基づいて、特定の操作データ列 1 3 2 が決定され、それによって血流の速度を示すベクトル 1 2 0 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 1 1 に示す参照データ列 1 5 6、第 3 演算器 1 3 4、符号ビット列 1 3 6 及び操作テーブル C 1 3 8 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。

【 0 0 5 6 】

20

演算対象の参照データ列 1 5 6 は、4 つの入力データ $\text{delt_TH}[0]$ 、 $\text{delt_TH}[1]$ 、 $\text{delt_TH}[2]$ 、 $\text{delt_TH}[3]$ から構成されている。ここで、 $\text{delt_TH}[P]$ は、位相差 $\{ \theta(2P+1) - \theta(2P+2) \}$ 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 $\text{delt_TH}[0] = | \theta_1 - \theta_2 |$ により求められ、 $\text{delt_TH}[1] = | \theta_3 - \theta_4 |$ で求められる。これらの角度差 $\text{delt_TH}[P]$ は、参照データ列 1 5 6 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 1 5 6 は、第 3 演算器 1 3 4 に入力される。

【 0 0 5 7 】

図 1 1 に示す第 3 演算器 1 3 4 で行われる演算の内容を、図 1 4 を用いて説明する。第 3 演算器 1 3 4 では、参照データ列 1 5 6 から比較データ列 1 5 8 を減算して比較結果データ列 1 6 0 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 1 5 8 に代入される値は全て 180 となる。つまり、 $\text{PI}[0]=0x00B4$ 、 $\text{PI}[1]=0x00B4$ 、 $\text{PI}[2]=0x00B4$ 、 $\text{PI}[3]=0x00B4$ となる。比較結果データ列 1 6 0 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。

30

【 0 0 5 8 】

次に、符号ビット列 1 3 6 と操作テーブル C 1 3 8 の関係について、図 1 5 を用いて記す。図 1 5 においては、符号ビット列 1 3 6 は、“0 1 1 0” 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 1 3 6 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 C 1 3 8 の中から特定される操作データ列 1 6 2 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 1 3 2 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 1 5 に示す操作テーブル C 1 3 8 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 C 1 3 8 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 1 5 に示すように、例示のビットパターン“0 1 1 0”に対応する特定の操作データ列 1 6 2 は、矢印 1 6 4 で指し示す # 6 の位置の操作データ列であって $\text{TABLE_B}[24] = 0xFFFF$ 、 $\text{TABLE_B}[25] = 0x$

40

50

0000、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、
10
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
20
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ある。一般的な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。
30

【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
40
50

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 $\{ \theta(2P+1) + \theta(2P+2) \} / 2$ (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 $| \theta(2P+1) - \theta(2P+2) |$ (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。

10

【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。

20

【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 $(tmp[3] \gg 12)$ 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 $[(tmp[3] \gg 12) \& 0x8]$ 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。

30

【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。

40

【0066】

50

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 S I G N 2 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で、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 S I G N 2 のビットパターンが形成される。

10

【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 B 130 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 S I G N 2 のビットパターンに応じて、操作テーブル B 130 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 S I G N 2 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。

20

【0068】

6 行目に記す式 OFFSET[] = OFFSET[] & MASK[] は、S I M D 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 とした MASK[] との論理積を取ることで、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。

30

【0069】

7 行目に記す式 out[] = ave_TH[] + OFFSET[] は、S I M D 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 D S P 36 によって行われるデータ処理である。

40

【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

50

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。

10

【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。

20

30

【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。

40

【0076】

50

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 D 1 8 6 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 1 9 の概念図に示す 2 値化処理を行う場合は、図 5 及び図 1 1 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 $\times$ 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 $\times$ 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ます、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】

10

20

30

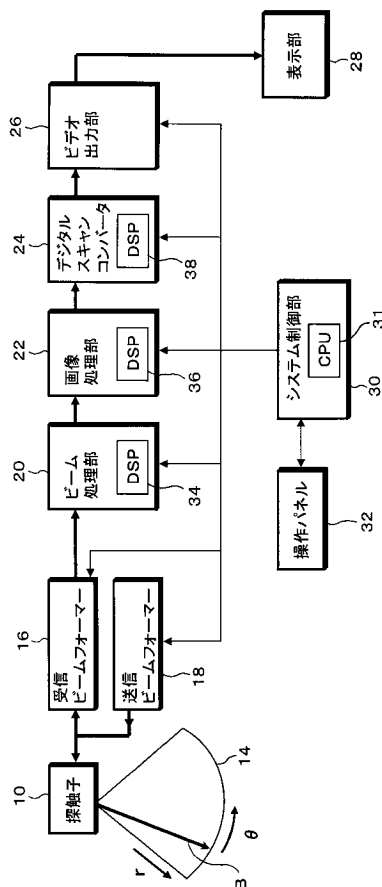
40

50

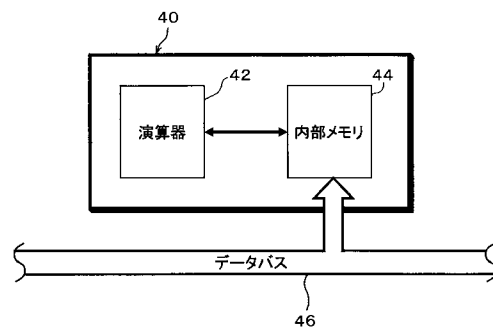
【 0 0 7 9 】

7 0 入力データ列、 7 2 第 1 演算器、 7 4 比較データ列、 7 6 符号ビット列、
8 0 操作テーブル A、 8 2 特定の操作データ列、 8 4 第 2 演算器、 8 6 出力データ列。

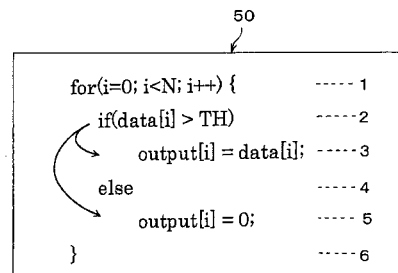
【 図 1 】



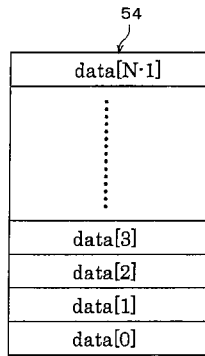
【 図 2 】



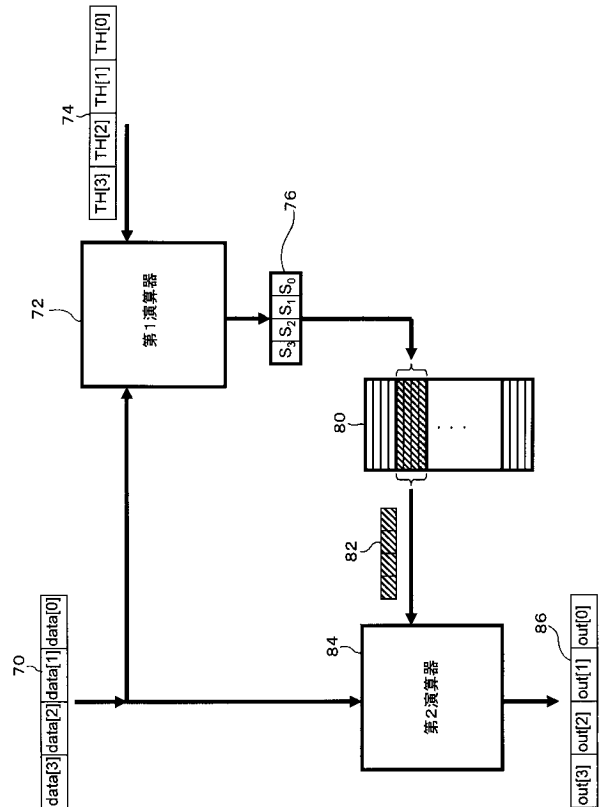
【 図 3 】



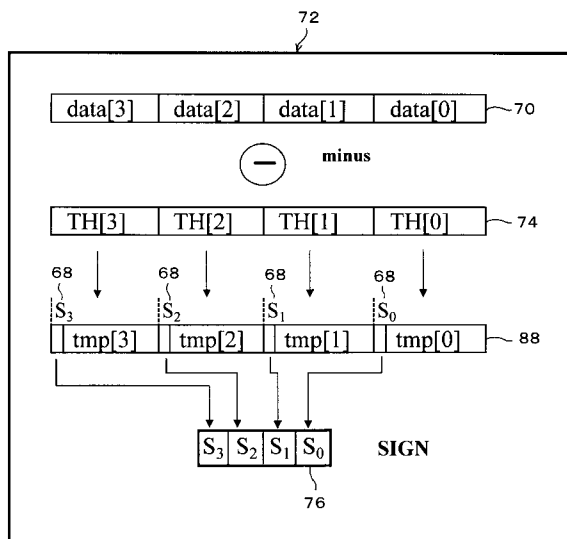
【 図 4 】



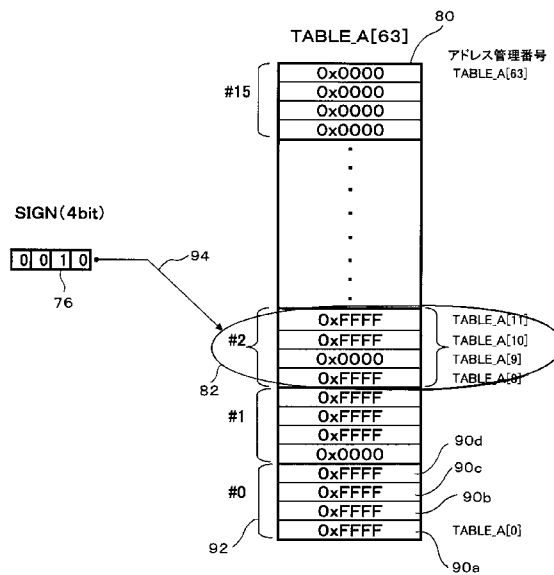
【 図 5 】



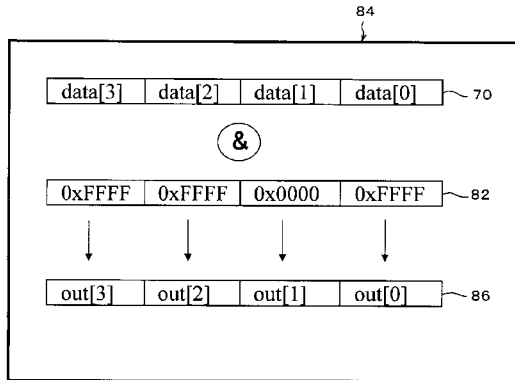
【 図 6 】



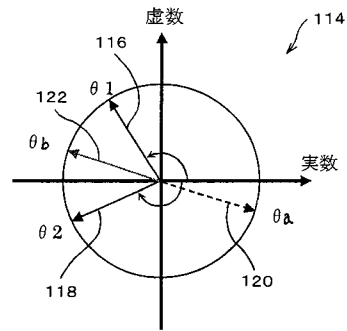
【 図 7 】



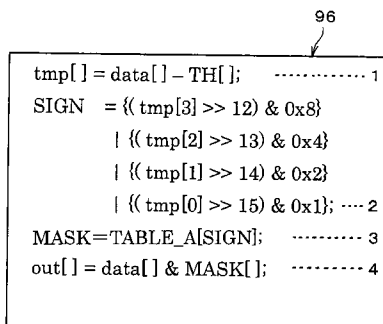
【 図 8 】



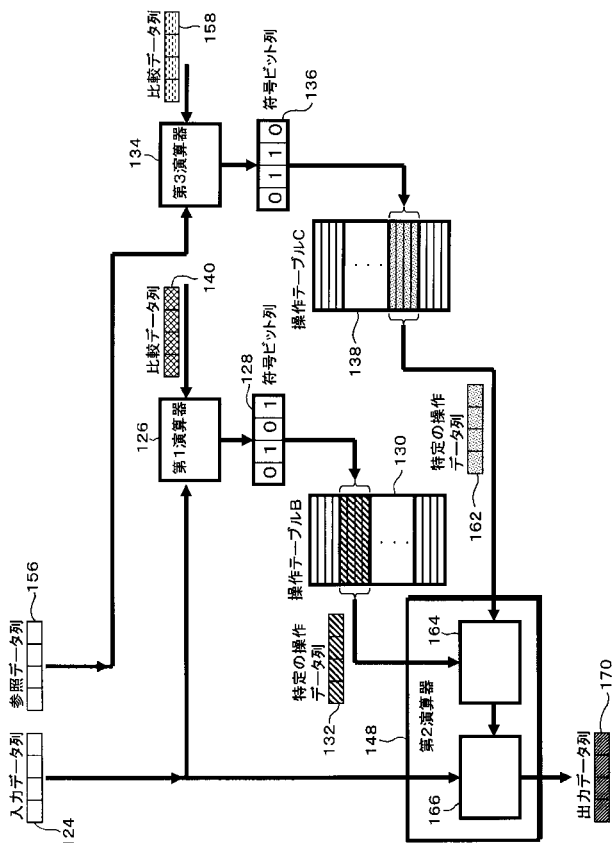
【 図 1 0 】



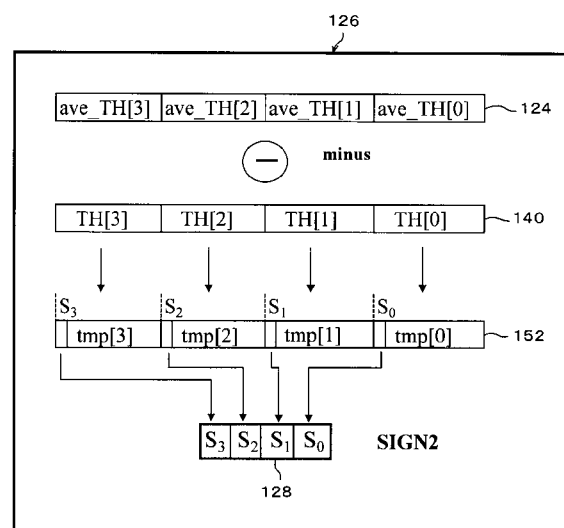
【 圖 9 】



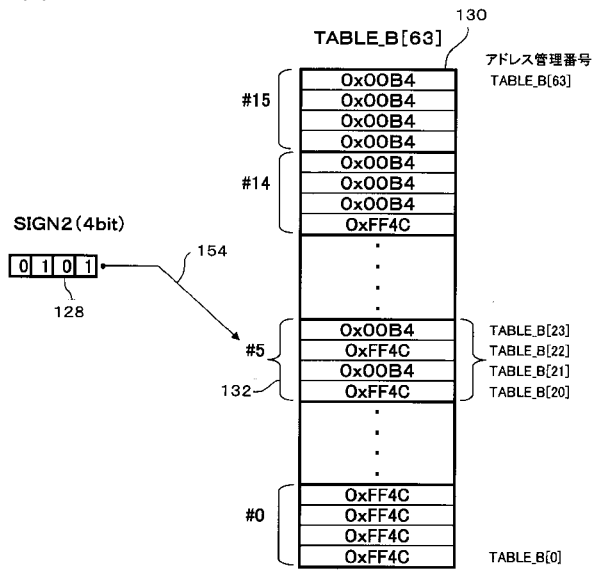
【 図 1 1 】



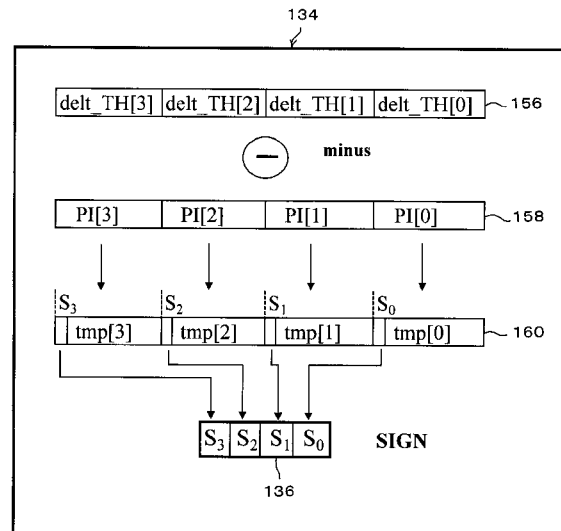
【 図 1 2 】



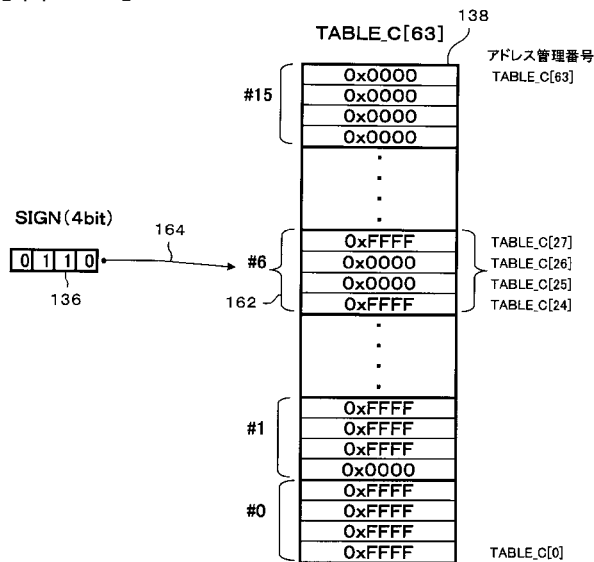
【図 13】



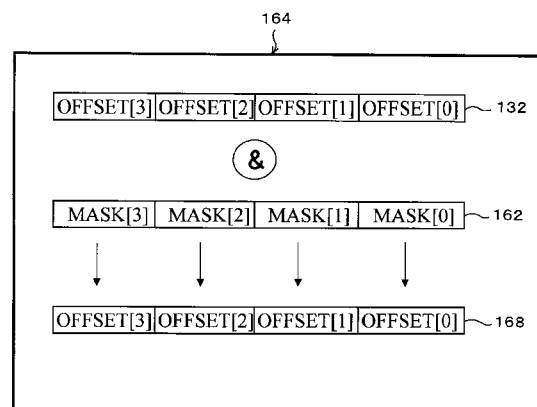
【図 14】



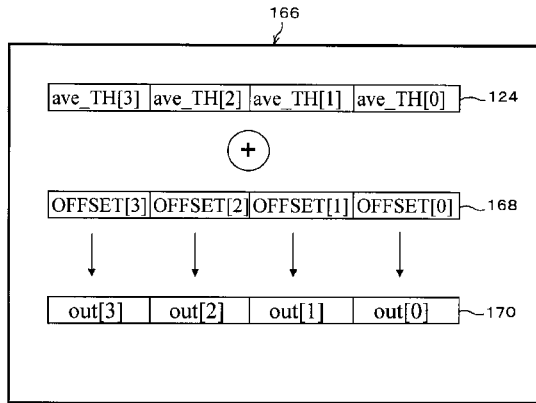
【図 15】



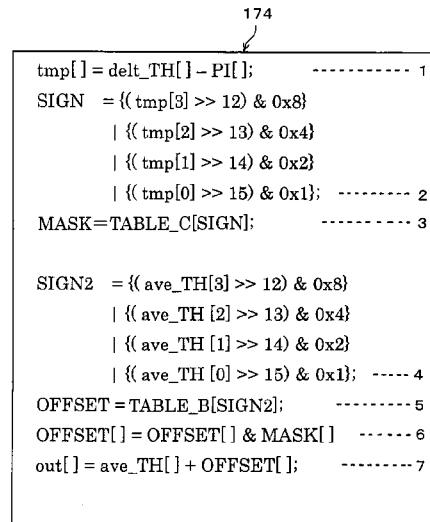
【図 16】



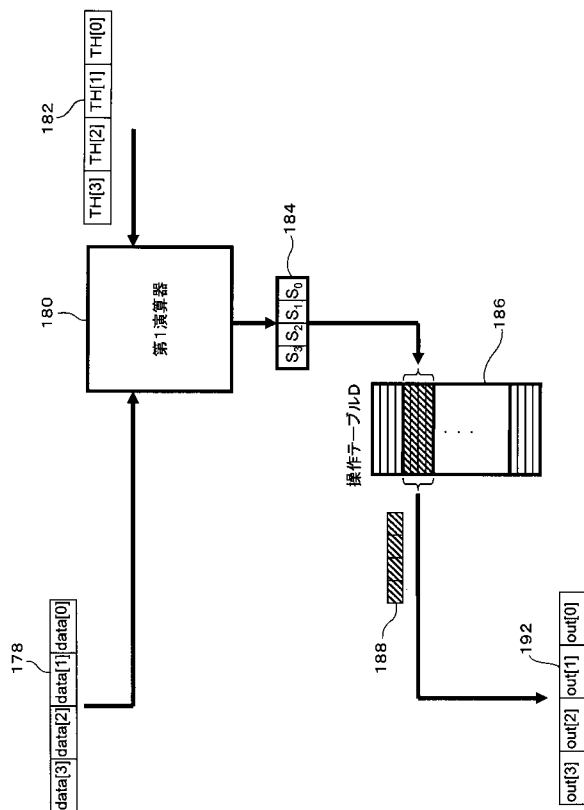
【 図 1 7 】



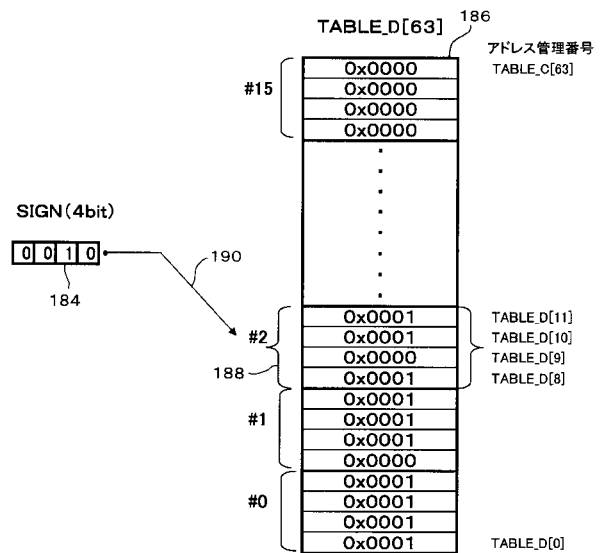
【 図 1 8 】



【 図 1 9 】



【 図 2 0 】



【 図 2 1 】

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
```

专利名称(译)	超声波诊断设备的处理器		
公开(公告)号	JP2007268156A	公开(公告)日	2007-10-18
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		
[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	JP4716911B2		
外部链接	Espacenet		

摘要(译)

解决的问题：在使用能够在超声诊断设备中并行处理多个数据的SIMD型算术单元时，执行高速算术处理而不执行if语句所代表的条件分支处理。有可能。比较输入数据串70和比较数据串74的第一算术单元72生成代表位串76。根据可以由代表位串76表示的位模式来确定特定操作数据串82。特定操作数据串82作用在第二计算器84中的输入数据串70上，并且获得输出数据串86。由于特定操作数据串82被预先存储在由多个操作数据串组成的操作表A80中，因此可以在维持并行处理的同时高速地执行与条件分支相对应的操作。[选择图]图5

