

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6373991号
(P6373991)

(45) 発行日 平成30年8月15日 (2018. 8. 15)

(24) 登録日 平成30年7月27日 (2018. 7. 27)

(51) Int. Cl.

F I

G 0 6 F 17/16 (2006. 01)

G 0 6 F 9/38 (2006. 01)

G 0 6 F 9/30 (2018. 01)

G 0 6 F 17/16 D

G 0 6 F 9/38 3 1 O G

G 0 6 F 9/38 3 7 O A

G 0 6 F 9/38 3 7 O C

G 0 6 F 9/30 3 5 O A

請求項の数 30 (全 119 頁)

(21) 出願番号 特願2016-530939 (P2016-530939)
 (86) (22) 出願日 平成26年11月13日 (2014. 11. 13)
 (65) 公表番号 特表2016-537722 (P2016-537722A)
 (43) 公表日 平成28年12月1日 (2016. 12. 1)
 (86) 国際出願番号 PCT/US2014/065557
 (87) 国際公開番号 W02015/073731
 (87) 国際公開日 平成27年5月21日 (2015. 5. 21)
 審査請求日 平成29年10月19日 (2017. 10. 19)
 (31) 優先権主張番号 14/082, 075
 (32) 優先日 平成25年11月15日 (2013. 11. 15)
 (33) 優先権主張国 米国 (US)

早期審査対象出願

(73) 特許権者 595020643
 クアアルコム・インコーポレイテッド
 QUALCOMM INCORPORATED
 アメリカ合衆国、カリフォルニア州 92
 121-1714、サン・ディエゴ、モア
 ハウス・ドライブ 5775
 (74) 代理人 100108855
 弁理士 蔵田 昌俊
 (74) 代理人 100109830
 弁理士 福原 淑弘
 (74) 代理人 100158805
 弁理士 井関 守三
 (74) 代理人 100112807
 弁理士 岡田 貴志

最終頁に続く

(54) 【発明の名称】 フィルタベクトル処理動作のためのタップ付き遅延線を利用するベクトル処理エンジンと、関連するベクトル処理システムおよび方法

(57) 【特許請求の範囲】

【請求項 1】

フィルタベクトル処理動作を供給するように構成されたベクトル処理エンジン (VPE) であって、

フィルタベクトル処理動作のための少なくとも1つの入力データフローパスにおいて入力ベクトルデータサンプルセットを供給することと、

少なくとも1つの出力データフローパスから結果として生じるフィルタ処理された出力ベクトルデータサンプルセットを受信することと、

前記結果として生じるフィルタ処理された出力ベクトルデータサンプルセットを記憶することと

を行うように構成された少なくとも1つのベクトルデータファイルと、

前記少なくとも1つの入力データフローパスにおいて少なくとも1つの実行ユニットと前記少なくとも1つのベクトルデータファイルとの間に設けられた少なくとも1つのタップ付き遅延線、前記少なくとも1つのタップ付き遅延線は、複数の処理ステージのうちの処理ステージごとに、シフトされた入力ベクトルデータサンプルセットを供給するように、前記フィルタベクトル処理動作におけるフィルタタップの数に等しい前記複数の処理ステージのうちの処理ステージごとに、複数のパイプラインレジスタにおいて、ベクトルデータサンプル幅ぶん前記入力ベクトルデータサンプルセットをシフトするように構成される、と、

前記少なくとも1つの入力データフローパス中に設けられた前記少なくとも1つの実行

ユニット、前記少なくとも 1 つの実行ユニットは、

前記複数の処理ステージのうちの処理ステージごとに、前記シフトされた入力ベクトルデータサンプルセットと前記少なくとも 1 つの実行ユニット内のレジスタファイルから受信したフィルタ係数を乗算して、前記フィルタベクトル処理動作のフィルタタップごとに中間出力ベクトルデータサンプルセットを生成するように構成された少なくとも 1 つの乗算器と、

前記複数の処理ステージのうちの処理ステージごとに、少なくとも 1 つの累算器において前記中間出力ベクトルデータサンプルセットを累算するように構成された前記少なくとも 1 つの累算器と

を備える、と

10

を備え、

前記少なくとも 1 つの実行ユニットは、前記少なくとも 1 つの出力データフローパス上に前記結果として生じるフィルタ処理された出力ベクトルデータサンプルセットを供給するように構成される、

V P E。

【請求項 2】

前記少なくとも 1 つの累算器は、前記入力ベクトルデータサンプルセットが前記少なくとも 1 つのベクトルデータファイルから再フェッチされることなく、前記複数の処理ステージのうちの処理ステージごとに前記中間出力ベクトルデータサンプルセットを累算するように構成される、

20

請求項 1 に記載の V P E。

【請求項 3】

前記少なくとも 1 つの累算器は、前記入力ベクトルデータサンプルセット中の入力ベクトルデータサンプルの数が前記フィルタベクトル処理動作におけるフィルタタップの数より小さい場合、前記入力ベクトルデータサンプルセットが前記少なくとも 1 つのベクトルデータファイルから再フェッチされることなく、前記複数の処理ステージのうちの処理ステージごとに前記中間出力ベクトルデータサンプルセットを累算するように構成される、

請求項 1 に記載の V P E。

【請求項 4】

前記少なくとも 1 つのタップ付き遅延線は、前記フィルタベクトル処理動作における前記フィルタタップの数が前記入力ベクトルデータサンプルセット中の入力ベクトルデータサンプルの数より大きい場合、前記少なくとも 1 つのベクトルデータファイルから前記少なくとも 1 つの入力データフローパス上に追加の入力ベクトルデータサンプルセットの別のフェッチを受信するようにさらに構成される、

30

請求項 1 に記載の V P E。

【請求項 5】

前記少なくとも 1 つのベクトルデータファイルは、

前記フィルタベクトル処理動作のための前記少なくとも 1 つの入力データフローパス中に前記少なくとも 1 つのベクトルデータファイルの幅の前記入力ベクトルデータサンプルセットを供給することと、

40

記憶されるべき、前記少なくとも 1 つの出力データフローパスからの前記少なくとも 1 つのベクトルデータファイルの前記幅の前記結果として生じるフィルタ処理された出力ベクトルデータサンプルセットを受信することと

を行うように構成される、請求項 1 に記載の V P E。

【請求項 6】

前記少なくとも 1 つのタップ付き遅延線は、

前記複数のパイプラインレジスタに、前記少なくとも 1 つのベクトルデータファイルから、前記少なくとも 1 つの入力データフローパスにおいて前記入力ベクトルデータサンプルセットを受信することと、

前記複数の処理ステージのうちの処理ステージごとに、前記少なくとも 1 つの実行ユ

50

ニットに、前記少なくとも1つの入力データフローパスにおいて前記シフトされた入力ベクトルデータサンプルセットの各々を供給することと

を行うようにさらに構成され、

前記少なくとも1つの実行ユニットは、前記複数の処理ステージのうちの処理ステージごとに、前記少なくとも1つのタップ付き遅延線から、前記少なくとも1つの入力データフローパスにおいて前記シフトされた入力ベクトルデータサンプルセットを受信するようにさらに構成される、

請求項1に記載のVPE。

【請求項7】

前記少なくとも1つのタップ付き遅延線は、

10

前記複数の処理ステージのうちの第1の処理ステージにおいて、前記少なくとも1つの実行ユニットに、前記少なくとも1つの入力データフローパスにおいて第1の入力ベクトルデータサンプルセットを供給することと、

前記複数の処理ステージのうちの後続の処理ステージにおいて、前記少なくとも1つの実行ユニットに、前記少なくとも1つの入力データフローパスにおいて前記シフトされた入力ベクトルデータサンプルセットを供給することと

を行うように構成され、

前記少なくとも1つの乗算器は、

前記第1の処理ステージにおいて第1の中間出力ベクトルデータサンプルセットを生成するために、前記第1の入力ベクトルデータサンプルセットと第1のフィルタ係数を乗算することと、

20

前記後続の処理ステージにおいて後続の中間出力ベクトルデータサンプルセットを生成するために、前記シフトされた入力ベクトルデータサンプルセットと後続のフィルタ係数を乗算することと

を行うように構成され、

前記少なくとも1つの累算器は、前記結果として生じるフィルタ処理された出力ベクトルデータサンプルセットを供給するために、前記少なくとも1つの累算器中の前記後続の中間出力ベクトルデータサンプルセットと前記第1の中間出力ベクトルデータサンプルセットとを累算するように構成される、

請求項6に記載のVPE。

30

【請求項8】

前記少なくとも1つのベクトルデータファイルは、

前記少なくとも1つの入力データフローパス中の少なくとも1つのタップ付き遅延線入力における前記入力ベクトルデータサンプルセットとして、前記入力ベクトルデータサンプルセットを供給することと、

前記少なくとも1つの出力データフローパス中の少なくとも1つの実行ユニット出力における前記結果として生じるフィルタ処理された出力ベクトルデータサンプルセットを受信することと

を行うようにさらに構成され、

前記少なくとも1つのタップ付き遅延線は、

40

前記少なくとも1つのベクトルデータファイルから、前記少なくとも1つの入力データフローパス中の前記少なくとも1つのタップ付き遅延線入力において前記入力ベクトルデータサンプルセットを受信することと、

前記複数の処理ステージのうちの処理ステージごとに、前記少なくとも1つの実行ユニットに、前記少なくとも1つの入力データフローパス中の少なくとも1つの実行ユニット入力において前記シフトされた入力ベクトルデータサンプルセットの各々を供給することと

を行うようにさらに構成され、

前記少なくとも1つの実行ユニットは、

前記複数の処理ステージのうちの処理ステージごとに、前記少なくとも1つのタップ

50

付き遅延線から、前記少なくとも 1 つの入力データフローパス中の前記少なくとも 1 つの実行ユニット入力において前記シフトされた入力ベクトルデータサンプルセットを受信することと、

前記少なくとも 1 つの実行ユニット出力において前記少なくとも 1 つの出力データフローパス上に前記結果として生じるフィルタ処理された出力ベクトルデータサンプルセットを供給することと

を行うようにさらに構成される、請求項 6 に記載の V P E。

【請求項 9】

前記少なくとも 1 つのタップ付き遅延線は、

複数のシャドウパイプラインレジスタに、前記少なくとも 1 つの入力データフローパス中の前記少なくとも 1 つのベクトルデータファイルから、次の入力ベクトルデータサンプルセットを受信することと、

前記シフトされた入力ベクトルデータサンプルセットを供給するために、プライマリタップ付き遅延線に、処理ステージごとに前記複数のシャドウパイプラインレジスタにおいて前記ベクトルデータサンプル幅ぶん前記次の入力ベクトルデータサンプルセットをシフトすることと

を行うように構成されるシャドウタップ付き遅延線を備え、

前記プライマリタップ付き遅延線は、前記複数の処理ステージ中の処理ステージごとに、前記少なくとも 1 つの実行ユニットに、前記少なくとも 1 つの入力データフローパスにおいて前記シフトされた入力ベクトルデータサンプルセットを供給するように構成される

、

請求項 1 に記載の V P E。

【請求項 10】

複数の入力ベクトルデータサンプル選択器をさらに備え、前記複数の入力ベクトルデータサンプル選択器の各々は、前記少なくとも 1 つのタップ付き遅延線中の前記複数のパイプラインレジスタのうちの 1 つのパイプラインレジスタに割り当てられ、

前記複数の入力ベクトルデータサンプル選択器は各々、前記割り当てられたパイプラインレジスタに、シフトされた入力ベクトルデータサンプルを記憶するために、隣接パイプラインレジスタ中に記憶された入力ベクトルデータサンプルと前記少なくとも 1 つのベクトルデータファイルからの前記入力ベクトルデータサンプルセットからの入力ベクトルデータサンプルとの間で選択するように構成される、

請求項 1 に記載の V P E。

【請求項 11】

前記少なくとも 1 つのタップ付き遅延線は、前記少なくとも 1 つの実行ユニットによって実行されるベクトル命令に従って、前記少なくとも 1 つのタップ付き遅延線のためのプログラマブル入力データパス構成に基づいて、前記少なくとも 1 つの実行ユニットと前記少なくとも 1 つのベクトルデータファイルとの間の前記少なくとも 1 つの入力データフローパス中に選択的に供給されるように構成可能である、

請求項 1 に記載の V P E。

【請求項 12】

前記少なくとも 1 つのタップ付き遅延線は、前記少なくとも 1 つの実行ユニットによって実行される次のベクトル命令に従って、前記少なくとも 1 つのタップ付き遅延線のための前記プログラマブル入力データパス構成に基づいて、前記少なくとも 1 つの入力データフローパス中に選択的に供給されるように再構成されるように構成される、

請求項 11 に記載の V P E。

【請求項 13】

前記少なくとも 1 つのタップ付き遅延線は、前記少なくとも 1 つの実行ユニットによって実行される前記 V P E のクロックサイクルごとに前記少なくとも 1 つの入力データフローパス中に選択的に供給されるように再構成されるように構成される、

請求項 12 に記載の V P E。

10

20

30

40

50

【請求項 14】

前記少なくとも 1 つの実行ユニットは、前記少なくとも 1 つの実行ユニットのためのプログラマブル入力データフローパス構成に基づいて、前記入力ベクトルデータサンプルセットとは異なるビット幅の入力ベクトルデータサンプルを処理するように構成可能である、

請求項 1 に記載の V P E。

【請求項 15】

前記少なくとも 1 つの実行ユニットは、

前記少なくとも 1 つのベクトルデータファイル中に一緒に記憶される前記少なくとも 1 つの出力データフローパス中の前記結果として生じるフィルタ処理された出力ベクトルデータサンプルセットのうちの実数の、結果として生じるフィルタ処理された出力ベクトルデータサンプルを供給することと、

前記実数の、結果として生じるフィルタ処理された出力ベクトルデータサンプルと別個に、前記少なくとも 1 つのベクトルデータファイル中に一緒に記憶される前記少なくとも 1 つの出力データフローパス中の前記結果として生じるフィルタ処理された出力ベクトルデータサンプルセットのうちの虚数の、結果として生じるフィルタ処理された出力ベクトルデータサンプルを供給することと

を行うように構成される、請求項 1 に記載の V P E。

【請求項 16】

前記少なくとも 1 つの実行ユニットは、

前記少なくとも 1 つのベクトルデータファイル中に一緒に記憶される前記少なくとも 1 つの出力データフローパス中の前記結果として生じるフィルタ処理された出力ベクトルデータサンプルセットのうちの偶数の、結果として生じるフィルタ処理された出力ベクトルデータサンプルを供給することと、

前記偶数の、結果として生じるフィルタ処理された出力ベクトルデータサンプルと別個に、前記少なくとも 1 つのベクトルデータファイル中に一緒に記憶される前記少なくとも 1 つの出力データフローパス中の前記結果として生じるフィルタ処理された出力ベクトルデータサンプルセットのうちの奇数の、結果として生じるフィルタ処理された出力ベクトルデータサンプルを供給することと

を行うように構成される、請求項 1 に記載の V P E。

【請求項 17】

前記少なくとも 1 つの実行ユニットのためのフィルタ係数を記憶するように構成されたレジスタファイルをさらに備え、

前記少なくとも 1 つの実行ユニットは、処理ステージごとに前記フィルタベクトル処理動作のために前記レジスタファイルからフィルタ係数を受信するようにさらに構成され、

前記少なくとも 1 つの乗算器は、処理ステージごとにそれぞれの中間出力ベクトルデータサンプルセットを生成するために、処理ステージのための前記受信したフィルタ係数と各受信した入力ベクトルデータサンプルセットを、前記少なくとも 1 つの乗算器中で乗算することによって、各処理ステージにおいて前記シフトされた入力ベクトルデータサンプルセットの各々に前記フィルタ係数を乗算するようにさらに構成される、

請求項 1 に記載の V P E。

【請求項 18】

前記少なくとも 1 つの実行ユニットは、以下の式に従って、ディスクリート時間有限インパルス応答 (F I R) フィルタから成る前記フィルタベクトル処理動作を実行するように構成される、

【数 1】

$$y[n] = \sum_{l=0}^{l=Y-1} h(l) * x[n-l]$$

10

20

30

40

50

ここで、

$x[n]$ は、前記入力ベクトルデータサンプルセットを備え、

$y[n]$ は、前記結果として生じる フィルタ処理された出力ベクトルデータサンプルセット を備え、

$h(1)$ は、フィルタ係数から成り、

Y は、前記フィルタ係数の数である、

請求項 1 に記載の V P E。

【請求項 19】

ベクトル処理エンジン (V P E) においてフィルタベクトル処理動作を実行するための装置であって、

フィルタベクトル処理動作のための少なくとも 1 つの入力データフローパス中に少なくとも 1 つのベクトルデータファイルから入力ベクトルデータサンプルセットをフェッチするための手段と、

前記少なくとも 1 つのベクトルデータファイルと少なくとも 1 つの実行ユニットとの間の前記少なくとも 1 つの入力データフローパスに設けられた少なくとも 1 つのタップ付き遅延線中に前記少なくとも 1 つの入力データフローパスにおける前記フェッチされた入力ベクトルデータサンプルセットを受信するための手段と、

複数の処理ステージの数が前記フィルタベクトル処理動作のフィルタタップの数に等しい、前記少なくとも 1 つの実行ユニット中の前記複数の処理ステージの処理ステージごとに、

前記入力ベクトルデータサンプルセットとして、シフトされた入力ベクトルデータサンプルセットを供給するように、前記入力ベクトルデータサンプルセット中の前記フィルタベクトル処理動作のための次のフィルタタップに前記少なくとも 1 つのタップ付き遅延線内の前記入力ベクトルデータサンプルセットをシフトするための手段、前記少なくとも 1 つのタップ付き遅延線は、前記フィルタベクトル処理動作におけるフィルタタップの数に等しい複数の処理ステージのうちの処理ステージごとに、複数のパイプラインレジスタにおいて、ベクトルデータサンプル幅ぶん前記入力ベクトルデータサンプルセットをシフトするように構成される、と、

現在の処理ステージにおいて、前記少なくとも 1 つの実行ユニットに、前記少なくとも 1 つの入力データフローパスにおいて前記少なくとも 1 つのタップ付き遅延線から前記シフトされた入力ベクトルデータサンプルセットを供給するための手段と、

中間出力ベクトルデータサンプルセットを生成するために、前記現在の処理ステージにおいて前記少なくとも 1 つの実行ユニット中の前記シフトされた入力ベクトルデータサンプルセットに前記少なくとも 1 つの実行ユニット内のレジスタファイルから受信したフィルタ係数を乗算するための手段と、

前記現在の処理ステージにおいて、前の結果として生じる フィルタ処理された出力ベクトルデータサンプルセットと前記中間出力ベクトルデータサンプルセットを累算するための手段と、

前記少なくとも 1 つのベクトルデータファイルに、前記少なくとも 1 つのベクトルデータファイルに記憶されるべき少なくとも 1 つの出力データフローパスにおける結果として生じる フィルタ処理された出力ベクトルデータサンプルセットとして、前記前の結果として生じる フィルタ処理された出力ベクトルデータサンプルセットを供給するための手段と を備える、装置。

【請求項 20】

ベクトル処理エンジン (V P E) においてフィルタベクトル処理動作を実行する方法であって、

前記フィルタベクトル処理動作のための少なくとも 1 つの入力データフローパス中に少なくとも 1 つのベクトルデータファイルから入力ベクトルデータサンプルセットをフェッチすることと、

前記少なくとも 1 つのベクトルデータファイルと少なくとも 1 つの実行ユニットとの間

10

20

30

40

50

の前記少なくとも1つの入力データフローパスに設けられた少なくとも1つのタップ付き遅延線中に前記少なくとも1つの入力データフローパスにおける前記フェッチされた入力ベクトルデータサンプルセットを受信することと、

複数の処理ステージの数が前記フィルタベクトル処理動作のフィルタタップの数に等しい、前記少なくとも1つの実行ユニット中の前記複数の処理ステージの処理ステージごとに、

前記入力ベクトルデータサンプルセットとして、シフトされた入力ベクトルデータサンプルセットを供給するように、前記入力ベクトルデータサンプルセット中の前記フィルタベクトル処理動作のための次のフィルタタップに前記少なくとも1つのタップ付き遅延線内の前記入力ベクトルデータサンプルセットをシフトすること、前記少なくとも1つのタップ付き遅延線は、前記フィルタベクトル処理動作におけるフィルタタップの数に等しい複数の処理ステージのうちの処理ステージごとに、複数のパイプラインレジスタにおいて、ベクトルデータサンプル幅ぶん前記入力ベクトルデータサンプルセットをシフトするように構成される、と、

10

現在の処理ステージにおいて、前記少なくとも1つの実行ユニットに、前記少なくとも1つの入力データフローパスにおいて前記少なくとも1つのタップ付き遅延線から前記シフトされた入力ベクトルデータサンプルセットを供給することと、

中間出力ベクトルデータサンプルセットを生成するために、前記現在の処理ステージにおいて前記少なくとも1つの実行ユニット中の前記シフトされた入力ベクトルデータサンプルセットに前記少なくとも1つの実行ユニット内のレジスタファイルから受信したフ

20

ィルタ係数を乗算することと、
前記現在の処理ステージにおいて、前の結果として生じるフィルタ処理された出力ベクトルデータサンプルセットと前記中間出力ベクトルデータサンプルセットを累算することと、

前記少なくとも1つのベクトルデータファイルに、前記少なくとも1つのベクトルデータファイルに記憶されるべき少なくとも1つの出力データフローパスにおける結果として生じるフィルタ処理された出力ベクトルデータサンプルセットとして、前記前の結果として生じるフィルタ処理された出力ベクトルデータサンプルセットを供給することと

を備える、方法。

【請求項 2 1】

30

前記入力ベクトルデータサンプルセット中の入力ベクトルデータサンプルの数が前記フィルタベクトル処理動作における前記フィルタタップの数より小さい場合、前記フィルタベクトル処理動作の間に前記少なくとも1つのベクトルデータファイルから前記入力ベクトルデータサンプルセットを再フェッチしないことを更に備える、

請求項 2 0 に記載の方法。

【請求項 2 2】

前記入力ベクトルデータサンプルセット中の入力ベクトルデータサンプルの数が前記フィルタベクトル処理動作における前記フィルタタップの数より大きい場合、

前記フィルタベクトル処理動作のための前記少なくとも1つの入力データフローパス中に前記少なくとも1つのベクトルデータファイルから次の入力ベクトルデータサンプルセットをフェッチすることと、

40

前記少なくとも1つのタップ付き遅延線中に、前記少なくとも1つのベクトルデータファイルから、前記少なくとも1つの入力データフローパスにおける前記次のフェッチされた入力ベクトルデータサンプルセットを受信することと

をさらに備える、請求項 2 0 に記載の方法。

【請求項 2 3】

複数のシャドウパイプラインレジスタ中に、前記少なくとも1つの入力データフローパス中の前記少なくとも1つのベクトルデータファイルから、次の入力ベクトルデータサンプルセットを受信することと、

前記シフトされた入力ベクトルデータサンプルセットを供給するために、プライマリタ

50

ップ付き遅延線に、処理ステージごとに前記複数のシャドウパイプラインレジスタにおいてベクトルデータサンプル幅ぶん前記次の入力ベクトルデータサンプルセットをシフトすることと

をさらに備え、

前記プライマリタップ付き遅延線は、前記複数の処理ステージ中の処理ステージごとに、前記少なくとも1つの実行ユニットに、前記少なくとも1つの入力データフローパスにおける前記シフトされた入力ベクトルデータサンプルセットを供給するように構成される、

請求項20に記載の方法。

【請求項24】

10

前記少なくとも1つのタップ付き遅延線中の複数のパイプラインレジスタのうちの隣接パイプラインレジスタに前記シフトされた入力ベクトルデータサンプルセットを記憶するために、前記少なくとも1つのタップ付き遅延線中の前記隣接パイプラインレジスタ中に記憶された入力ベクトルデータサンプルと前記少なくとも1つのベクトルデータファイルからの前記入力ベクトルデータサンプルセットからの入力ベクトルデータサンプルとの間で選択することをさらに備える、

請求項20に記載の方法。

【請求項25】

前記少なくとも1つの実行ユニットによって実行されるベクトル命令に従って、前記少なくとも1つのタップ付き遅延線のためのプログラマブル入力データパス構成に基づいて、前記少なくとも1つの実行ユニットと前記少なくとも1つのベクトルデータファイルとの間の前記少なくとも1つの入力データフローパス中に前記少なくとも1つのタップ付き遅延線を選択的に供給することをさらに備える、

20

請求項20に記載の方法。

【請求項26】

前記少なくとも1つの実行ユニットによって実行される次のベクトル命令に従って、前記少なくとも1つのタップ付き遅延線のための前記プログラマブル入力データパス構成に基づいて、前記少なくとも1つの入力データフローパス中に選択的に供給されるように前記少なくとも1つのタップ付き遅延線を再構成することをさらに備える、

請求項25に記載の方法。

30

【請求項27】

前記少なくとも1つの実行ユニットによって実行される前記VPEのクロックサイクルごとに前記少なくとも1つの入力データフローパス中に選択的に供給されるように前記少なくとも1つのタップ付き遅延線を再構成することを備える、

請求項26に記載の方法。

【請求項28】

前記少なくとも1つの実行ユニットのためのプログラマブル入力データフローパス構成に基づいて、前記少なくとも1つの実行ユニット中の前記入力ベクトルデータサンプルセットとは異なるビット幅の入力ベクトルデータサンプルを処理することをさらに備える、

請求項20に記載の方法。

40

【請求項29】

前記結果として生じるフィルタ処理された出力ベクトルデータサンプルセットとして前記前の結果として生じるフィルタ処理された出力ベクトルデータサンプルセットを供給することは、

前記少なくとも1つのベクトルデータファイル中に一緒に記憶される前記少なくとも1つの出力データフローパス中の前記結果として生じるフィルタ処理された出力ベクトルデータサンプルセットのうちの実数の、結果として生じるフィルタ処理された出力ベクトルデータサンプルを供給することと、

前記実数の、結果として生じるフィルタ処理された出力ベクトルデータサンプルと別個に、前記少なくとも1つのベクトルデータファイル中に一緒に記憶される前記少なくとも

50

1つの出力データフローパス中の前記結果として生じるフィルタ処理された出力ベクトルデータサンプルセットのうちの虚数の、結果として生じるフィルタ処理された出力ベクトルデータサンプルを供給することと

を備える、請求項20に記載の方法。

【請求項30】

前記結果として生じるフィルタ処理された出力ベクトルデータサンプルセットとして前記前の結果として生じるフィルタ処理された出力ベクトルデータサンプルセットを供給することは、

前記少なくとも1つのベクトルデータファイル中に一緒に記憶される前記少なくとも1つの出力データフローパス中の前記結果として生じるフィルタ処理された出力ベクトルデータサンプルセットのうちの偶数の、結果として生じるフィルタ処理された出力ベクトルデータサンプルを供給することと、

前記偶数の、結果として生じるフィルタ処理された出力ベクトルデータサンプルと別個に、前記少なくとも1つのベクトルデータファイル中に一緒に記憶される前記少なくとも1つの出力データフローパス中の前記結果として生じるフィルタ処理された出力ベクトルデータサンプルセットのうちの奇数の、結果として生じるフィルタ処理された出力ベクトルデータサンプルを供給することと

を備える、請求項20に記載の方法。

【発明の詳細な説明】

【関連出願】

【0001】

[0001]本出願は、2013年3月13日に出願され、参照によりその全体が本明細書に組み込まれている、「VECTOR PROCESSING ENGINES HAVING PROGRAMMABLE DATA PATH CONFIGURATIONS FOR PROVIDING MULTI-MODE VECTOR PROCESSING, AND RELATED VECTOR PROCESSORS, SYSTEMS, AND METHODS」、123249と題する、米国特許出願第13/798,641号に関連する。

【0002】

[0002]本出願は、2013年3月13日に出願され、参照によりその全体が本明細書に組み込まれている、「VECTOR PROCESSING CARRY-SAVE ACCUMULATORS EMPLOYING REDUNDANT CARRY-SAVE FORMAT TO REDUCE CARRY PROPAGATION, AND RELATED VECTOR PROCESSORS, SYSTEMS, AND METHODS」、123248と題する、米国特許出願第13/798,618号に関連する。

【0003】

[0003]本出願は、2013年11月15日に出願され、参照によりその全体が本明細書に組み込まれている、「VECTOR PROCESSING ENGINES (VPES) EMPLOYING A TAPPED-DELAY LINE(S) FOR PROVIDING PRECISION FILTER VECTOR PROCESSING OPERATIONS WITH REDUCED SAMPLE REFETCHING AND POWER CONSUMPTION, AND RELATED VECTOR PROCESSOR SYSTEMS AND METHODS」、124362と題する、米国特許出願第14/082,075号にも関連する。

【0004】

[0004]本出願は、2013年11月15日に出願され、参照によりその全体が本明細書に組み込まれている、「VECTOR PROCESSING ENGINES (VPES) EMPLOYING TAPPED-DELAY LINE(S) FOR PROVIDING PRECISION CORRELATION/COVARIANCE

10

20

30

40

50

E VECTOR PROCESSING OPERATIONS WITH REDUCED SAMPLE RE-FETCHING AND POWER CONSUMPTION, AND RELATED VECTOR PROCESSOR SYSTEMS AND METHODS」、124364と題する、米国特許出願第14/082,079号にも関連する。

【0005】

[0005]本出願は、2013年11月15日に出願され、参照によりその全体が本明細書に組み込まれている、「VECTOR PROCESSING ENGINES (VP
Es) EMPLOYING REORDERING CIRCUITRY IN DATA
FLOW PATHS BETWEEN EXECUTION UNITS AND
D VECTOR DATA MEMORY TO PROVIDE IN-FLIGHT
T REORDERING OF OUTPUT VECTOR DATA STORED
D TO VECTOR DATA MEMORY, AND RELATED VEC
TOR PROCESSOR SYSTEMS AND METHODS」、12445
0と題する、米国特許出願第14/082,081号にも関連する。

10

【0006】

[0006]本出願は、2013年11月15日に出願され、参照によりその全体が本明細書に組み込まれている、「VECTOR PROCESSING ENGINES (VP
Es) EMPLOYING MERGING CIRCUITRY IN DATA
FLOW PATHS BETWEEN EXECUTION UNITS AND V
ECTOR DATA MEMORY TO PROVIDE IN-FLIGHT M
ERGING OF OUTPUT VECTOR DATA STORED TO V
ECTOR DATA MEMORY, AND RELATED VECTOR PR
OCESSING INSTRUCTIONS, SYSTEMS, AND METH
ODS」、124363U1と題する、米国特許出願第14/082,073号にも関連する。

20

【0007】

[0007]本出願は、2013年11月15日に出願され、参照によりその全体が本明細書に組み込まれている、「VECTOR PROCESSING ENGINES (VP
Es) EMPLOYING DESPREADING CIRCUITRY IN D
ATA FLOW PATHS BETWEEN EXECUTION UNITS A
ND VECTOR DATA MEMORY TO PROVIDE IN-FLIG
HT DESPREADING OF SPREAD-SPECTRUM SEQUEN
CES, AND RELATED VECTOR PROCESSING INSTR
UCTIONS, SYSTEMS, AND METHODS」、124363U2と
題する、米国特許出願第14/082,067号にも関連する。

30

【技術分野】

【0008】

[0008]本開示の分野は、単一命令多重データ(SIMD)プロセッサと多重命令多重データ(MIMD)プロセッサとを含む、ベクトル演算とスカラー演算とを処理するためのベクトルプロセッサおよび関連システムに関する。

40

【背景技術】

【0009】

[0009]ワイヤレスコンピューティングシステムは、デジタル情報領域において最も普及した技術の1つに急速になりつつある。技術における進歩により、ワイヤレス通信デバイスは、より小型でより強力になった。たとえば、ワイヤレスコンピューティングデバイスには、一般に、小型で軽量の、ユーザが容易に持ち運べるポータブルワイヤレス電話、携帯情報端末(PDA)、およびページングデバイスが含まれる。より具体的には、携帯電話およびインターネットプロトコル(IP)電話などのポータブルワイヤレス電話は、ワイヤレスネットワークを介して音声とデータパケットとを通信することができる。さらに

50

、多くのそのようなワイヤレス通信デバイスには、他のタイプのデバイスが含まれる。たとえば、ワイヤレス電話には、デジタルスチルカメラ、デジタルビデオカメラ、デジタルレコーダ、および/またはオーディオファイルプレーヤが含まれ得る。また、ワイヤレス電話は、インターネットにアクセスするために使用され得るウェブインターフェースを含むことができる。さらに、ワイヤレス通信デバイスは、設計されたワイヤレス通信技術規格（たとえば、符号分割多元接続（C D M A）、広帯域C D M A（W C D M A（登録商標））、およびロングタームエボリューション（L T E（登録商標）））に従って高速ワイヤレス通信データを処理するための複合処理リソースを含む場合がある。そのため、これらのワイヤレス通信デバイスはかなりのコンピューティング能力を含む。

【0010】

10

[0010]ワイヤレスコンピューティングデバイスは、より小型でより強力になるにつれて、ますますリソースの制約を受けるようになる。たとえば、画面サイズ、利用可能なメモリおよびファイルシステム空間の量、ならびに入出力能力の量は、デバイスの小さいサイズによって制限される場合がある。さらに、バッテリーサイズ、バッテリーによって供給される電力の量、およびバッテリーの寿命も制限される。デバイスのバッテリー寿命を増やすための1つの方法は、より少ない電力を消費するプロセッサを設計することである。

【0011】

[0011]この関連で、ベクトルプロセッサを含むベースバンドプロセッサが、ワイヤレス通信デバイスに利用され得る。ベクトルプロセッサは、ベクトル、すなわちデータのアレイに働く高水準の演算を提供するベクトルアーキテクチャを有する。ベクトル処理は、1つのデータセットに対してベクトル命令を実行し、次いで、ベクトル内の後続要素のためにベクトル命令を再フェッチし復号することとは対照的に、ベクトル命令を1度フェッチし、次いで、データ要素のアレイ全体にわたってベクトル命令を複数回実行することを伴う。このプロセスにより、他の要因の中でも、各ベクトル命令はより少ない回数しかフェッチされる必要がないので、プログラムを実行するために必要とされるエネルギーの低減が可能になる。ベクトル命令は、同時に複数のクロック周期にわたって長いベクトルに対して動作するので、簡易な順序ベクトル命令ディスパッチを用いて、高度の並列性が達成可能である。

20

【0012】

[0012]図1は、ワイヤレスコンピュータデバイスなどのコンピューティングデバイス内で利用され得る例示的なベースバンドプロセッサ10を示す。ベースバンドプロセッサ10は、特定のアプリケーションのための関数固有ベクトル処理を提供することに各々が専用化された複数の処理エンジン（P E）12を含む。この例では、6つの別個のP E 12（0）～P E 12（5）がベースバンドプロセッサ10内に設けられる。P E 12（0）～P E 12（5）は各々、共有メモリ16からP E 12（0）～P E 12（5）に供給される固定Xビット幅のベクトルデータ14に対してベクトル処理を提供するように構成される。たとえば、ベクトルデータ14は512ビット幅であり得る。ベクトルデータ14は、Xのより小さい倍数のビット幅のベクトルデータサンプルセット18（0）～18（Y）（たとえば、16ビットおよび32ビットのサンプルセット）内で定義され得る。このようにして、P E 12（0）～P E 12（5）は、高度の並列性を達成するために、P E 12（0）～P E 12（5）に並列に供給される複数のベクトルデータサンプルセット18に対するベクトル処理を提供することが可能である。各P E 12（0）～P E 12（5）は、ベクトルデータ14上で処理されるベクトル命令の結果を記憶するためのベクトルレジスタファイル（V R）を含む場合がある。

30

40

【0013】

[0013]図1のベースバンドプロセッサ10内の各P E 12（0）～P E 12（5）は、特定のタイプの固定演算を効率的に実行するように特に設計された、特定の専用回路とハードウェアとを含む。たとえば、図1のベースバンドプロセッサ10は、別個のW C D M AのP E 12（0）、P E 12（1）と、L T EのP E 12（4）、P E 12（5）とを含むが、これは、W C D M AおよびL T Eが異なるタイプの特殊な演算を伴うからである

50

。したがって、別個のWCDMA固有PE12(0)、PE12(1)とLTE固有PE12(4)、PE12(5)とを設けることによって、PE12(0)、PE12(1)、PE12(4)、PE12(5)の各々は、高効率演算のための、WCDMAおよびLTE用の頻繁に実行される関数に固有の特殊な専用回路を含むように設計され得る。この設計は、効率的でない方式ではあるが、より多数の無関係な演算をサポートするために柔軟であるように設計された、より一般的な回路とハードウェアとを含むスカラー処理エンジンとは対照的である。

【0014】

[0014]特殊ベクトル処理動作の1つのタイプは、フィルタリングである。フィルタ動作は、サンプリングされた入力時間関数の重畳の量子化時間領域表現と、フィルタの重み付け関数の表現とを計算する。時間領域内の重畳は、周波数領域内の乗算に対応する。このように、デジタルフィルタは、間隔が均一なサンプル間隔で実行される乗算および加算の拡張シーケンスにより実現され得る。デジタル化入力信号は、加算器(アダー)、遅延サンプル、および乗算器にクロックされたデータをシフトする構造を通じてデジタル化入力信号サンプルを渡すことによってPEにおいてフィルタされ得る。

たとえば、有限インパルス応答(FIR)フィルタは、フィルタ関数を計算するために、「N」計算フィルタ係数を有する遅延線上の遅延タップの有限数(「N」)を使用して実施され得る。

【0015】

[0015]しかしながら、フィルタリング演算は、ベクトルプロセッサにおいて供給される特殊データフローパスに起因して、ベクトルプロセッサにおいて並列化することは困難であり得る。フィルタリングされるべき入力ベクトルデータサンプルセットが、フィルタ遅延タップの間でシフトされると、入力ベクトルデータサンプルセットは、ベクトルデータファイルから再フェッチされ、したがって電力消費が増大し、スループットが低減される。メモリからの入力ベクトルデータサンプルセットの再フェッチを最小化するために、データフローパスは、効率的な並列化処理のために、遅延タップと同じ数の乗算器を設けるように構成される可能性がある。しかしながら、他のベクトル処理動作は、より少ない乗算器を要求し、その結果、データフローパス内の乗算器の非効率的なスケーリングおよび過少利用がもたらされる。スケラビリティを供給するために、乗算器の数がフィルタ遅延タップの数よりも少なくなるように削減された場合、フィルタ処理の様々なフェーズに対して同じ入力ベクトルデータサンプルセットを取得するために、メモリにより多くの再フェッチが必要とされることによって、並列化が制限される。

【発明の概要】

【0016】

[0016]再フェッチおよび電力消費が低減される精度フィルタベクトル処理動作を供給するためのタップ付き遅延線を利用するベクトル処理エンジン(VPE)が開示される。関連するベクトル処理システムおよび方法がさらに開示される。本願に開示されるVPEは、フィルタ係数を持つ入力ベクトルデータサンプルセットの乗算および結果の累算を含む。電力消費を低減するために、メモリからの入力ベクトルデータサンプルの再フェッチを最小化するために、タップ付き遅延線は、ベクトルデータファイルとVPE中の実行ユニットとの間のデータフローパス中に含まれる。タップ付き遅延線は、フィルタベクトル処理動作を実行するための実行ユニットに入力ベクトルデータサンプルセットを受信し供給するように構成される。タップ付き遅延線は、各フィルタ遅延タップのための入力ベクトルデータサンプルをシフトすることと、実行ユニットにシフトされた入力ベクトルデータサンプルセットを供給することとを行うようにさらに構成され、シフトされた入力ベクトルデータサンプルセットは、フィルタベクトル処理動作の各処理ステージの間のベクトルデータファイルから再フェッチされることはない。この方法では、低減される再フェッチ動作は、電力消費を低減する。さらにVPE中のデータフローパスの効率は、メモリからの入力ベクトルデータサンプルセットの再フェッチ遅延によって制限されない。このシフトされた入力ベクトルデータサンプルセットは、実行ユニットに局在するように設けられ

る。実行ユニット中のベクトル処理は、データフローの制限ではなく、コンピュータリソースのみによって制限される。さらに、フィルタベクトル処理動作は、実行ユニット中の中間フィルタステージのための累算がメモリ中に格納されないため、より正確であり、実行ユニットがメモリから再フェッチされるべきシフトされた入力フィルタサンプルデータを待つ間に、結果として丸められる。

【0017】

[0017]さらに、本願に開示されるタップ付き遅延線を利用するVPE中のフィルタベクトル処理動作は、フィルタ遅延タップと同じ数の乗算器を要求すること無しに完全に並列化されることができる。これは、タップ付き遅延線がベクトルデータファイルからの入力ベクトルデータサンプルセットの再フェッチからの遅延を被ること無しにフィルタ処理のための実行ユニットに、シフトされた入力ベクトルデータサンプルを供給することができるからである。フィルタタップ遅延より少ない乗算器でフィルタベクトル処理動作の並列化を許容することによって、VPEのデータフローパスは、より少ない乗算器を要求するベクトル処理動作の他のタイプのために予約された乗算器の数を簡単にスケールダウンするように再プログラムされることができ、そのため、VPEの乗算器は、効率的な目的のために過少利用されることはない。例えば、VPE中のデータフローパスがベクトル処理動作の他のタイプのために要求されるより、フィルタベクトル処理動作のためにより多くの乗算器の数を利用するように構成される場合、追加の乗算器は、これらの他のベクトル処理動作の実行の間使用され得ず、その結果、VPEの乗算器処理リソースを過少利用することになる。

【0018】

[0018]この関連で、一実施形態では、フィルタベクトル処理動作を供給するように構成されたVPEが供給される。VPEは、少なくとも1つのベクトルデータファイルを備える。ベクトルデータファイルは、フィルタベクトル処理動作のための少なくとも1つの入力データフローパス中の入力ベクトルデータサンプルセットを供給するように構成される。ベクトルデータファイルはまた、少なくとも1つの出力データフローパスからの結果として生じるフィルタ化出力ベクトルデータサンプルセットを受信するように構成される。ベクトルデータファイルはまた、結果として生じるフィルタ化出力ベクトルデータサンプルセットを記憶するようにさらに構成される。VPEはまた、少なくとも1つの入力データフローパス中の少なくとも1つの実行ユニットと少なくとも1つのベクトルデータファイルとの間に供給される少なくとも1つのタップ付き遅延線を備える。タップ付き遅延線は、複数の処理ステージの中の各処理ステージのためのシフトされた入力ベクトルデータサンプルセットを供給するために、フィルタベクトル処理動作中のフィルタタップの数と等しい複数の処理ステージの中の各処理ステージのための複数のパイプラインレジスタ中のベクトルデータサンプルセットで入力ベクトルデータサンプルセットをシフトするように構成される。

【0019】

[0019]さらに、VPEはまた、少なくとも1つの入力データフローパス中に供給される少なくとも1つの実行ユニットを備える。実行ユニットは、フィルタベクトル処理動作の各フィルタタップのためのフィルタタップ出力ベクトルデータサンプルセットを生成するために、複数の処理ステージの中の各処理ステージのためのシフトされた入力ベクトルデータサンプルセット上にフィルタタップ動作を適用するように構成された少なくとも1つの乗算器を備える。実行ユニットはまた、複数の処理ステージの中の各処理ステージのための少なくとも1つの累算器中のフィルタタップ出力ベクトルデータサンプルセットを累算するように構成された少なくとも1つの累算器を備える。実行ユニットは、少なくとも1つの出力データフローパス上の結果として生じるフィルタ化出力ベクトルデータサンプルセットを供給するように構成される。この方法では、非限定的な例として、実行ユニットは、入力ベクトルデータサンプルセットがベクトルデータファイルからの再フェッチされずに、複数の処理ステージの中の各処理ステージのためのフィルタタップ出力ベクトルデータサンプルセットを累算するように構成される。

【 0 0 2 0 】

[0020]別の実施形態では、フィルタベクトル処理動作を供給するように構成されたVPEが供給される。VPEは、少なくとも1つのベクトルデータファイル手段を備える。ベクトルデータファイル手段は、フィルタベクトル処理動作のための少なくとも1つの入力データフローパス手段中に入力ベクトルデータサンプルセットを供給するための手段を備える。ベクトルデータファイル手段はさらに、記憶されるべき、少なくとも1つの出力データフローパス手段から結果として生じるフィルタ化出力ベクトルデータサンプルセットを受信するための手段を備える。VPEはさらに、少なくとも1つのベクトルデータファイル手段と少なくとも1つの入力データフロー手段中の少なくとも1つの実行ユニットとの間に設けられた少なくとも1つのタップ付き遅延線手段を備える。タップ付き遅延線手段は、複数の処理ステージの中の各処理ステージのためのシフトされた入力ベクトルベクトルデータサンプルセットを供給するために、フィルタベクトル処理動作中のフィルタタップの数に等しい、複数の処理ステージの中の各処理ステージのための複数のパイプラインレジスタ中のベクトルデータサンプル幅ぶん、入力ベクトルデータサンプルセットをシフトするための手段を備える。

10

【 0 0 2 1 】

[0021]さらに、VPEはまた、少なくとも1つの入力データフローパス中に供給される少なくとも1つの実行ユニット手段を備える。実行ユニット手段は、フィルタベクトル処理動作の各フィルタタップのためのフィルタタップ出力ベクトルデータサンプルセットを生成するために、複数の処理ステージの中の各処理ステージのためのシフトされた入力ベクトルデータサンプルセット上にフィルタタップ動作を適用するための手段を備える。実行ユニット手段は、複数の処理ステージの中の各処理ステージのための少なくとも1つの累算器においてフィルタタップ出力ベクトルデータサンプルセットを累算するための手段を備える。実行ユニット手段はさらに、フィルタベクトル処理動作のための少なくとも1つの出力データフローパス手段上に結果として生じるフィルタ化出力ベクトルデータサンプルを供給するための手段を備える。

20

【 0 0 2 2 】

[0022]別の実施形態では、VPEにおいてフィルタベクトル処理動作を実行する方法が供給される。方法は、フィルタベクトル処理動作のための少なくとも1つの入力データフローパス中に少なくとも1つのベクトルデータファイルからの入力ベクトルデータサンプルセットをフェッチすることを備える。方法はさらに、少なくとも1つのベクトルデータファイルと少なくとも1つの実行ユニットとの間の少なくとも1つの入力データフローパス中に設けられる少なくとも1つのタップ付き遅延線中に少なくとも1つの出力データフローパス中のフェッチされた入力ベクトルデータサンプルセットを受信することを備える。複数の処理ステージの数がフィルタベクトル処理動作のフィルタタップの数に等しい、少なくとも1つの実行ユニット中の複数の処理ステージの各処理ステージに対して、方法はさらに、現在の処理ステージ中の少なくとも1つの実行ユニットに、少なくとも1つの入力データフローパス中の少なくとも1つのタップ付き遅延線から入力ベクトルデータサンプルセットを供給することと、フィルタタップ出力ベクトルデータサンプルセットを生成するために、現在の処理ステージ中の少なくとも1つの実行ユニット中の入力ベクトルデータサンプルセット上にフィルタタップ動作を適用することと、現在の処理ステージ中の前の結果として生じるフィルタ化出力ベクトルデータサンプルセットとフィルタタップ出力ベクトルデータサンプルセットを累算することと、前の結果として生じるフィルタ化出力ベクトルデータサンプルセットを供給することと、入力ベクトルデータサンプルセットとしてシフトされた入力ベクトルデータサンプルセットを供給するために、入力ベクトルデータサンプルセット中のフィルタベクトル処理動作のための次のフィルタタップに少なくとも1つのタップ付き遅延線内の入力ベクトルデータサンプルセットをシフトすることとを備える。方法はさらに、少なくとも1つのベクトルデータファイル中に記憶されるべき少なくとも1つのベクトルデータファイルに、少なくとも1つの出力データフローパス中の結果として生じるフィルタ化出力ベクトルデータサンプルセットとして前の結果と

30

40

50

して生じるフィルタ化出力ベクトルデータサンプルセットを供給することを備える。

【図面の簡単な説明】

【 0 0 2 3 】

【図 1】特定のアプリケーション向けの関数固有ベクトル処理を提供するために各々が専用化された、複数のベクトル処理エンジン（V P E）を含む例示的なベクトルプロセッサの概略図。

【図 2】V P E 内に設けられた共通の回路およびハードウェアが、別個のV P Eを設ける必要なしに複数のアプリケーションまたは技術のために、特定のタイプのベクトル演算を高効率な方式で実行するために複数のモードでプログラムされ得るように、プログラム可能なデータパス構成を有するV P Eを含む例示的なベースバンドプロセッサの概略図。

10

【図 3】V P Eによってサポートされるフィルタベクトル処理動作において提供され得るディスクリット有限インパルス応答（F I R）フィルタの概略図。

【図 4】再フェッチおよび電力消費が低減される精度フィルタベクトル処理動作を提供するためのフィルタ係数データを用いて処理されるべきシフトされた入力ベクトルデータサンプルセットを受信し、実行ユニットに供給するためにタップ付き遅延線を利用する例示的なV P Eの概略図。

【図 5】例示的なフィルタベクトル命令に従って図 4のV P Eにおいて実行され得る例示的なフィルタベクトル処理動作を示すフローチャート。

【図 6 A】図 4のV P E内のレジスタファイルに記憶されたフィルタタップ係数の概略図。

20

【図 6 B】図 4のV P E内のベクトルデータファイルに記憶された例示的な入力ベクトルデータサンプルセットの概略図。

【図 7】図 4のV P Eにおいて提供され得る例示的なタップ付き遅延線とオプションのシャドウタップ付き遅延線とを示す概略図であって、例示的なタップ付き遅延線が、各々、V P Eによって実行されるフィルタベクトル処理動作の間に、ベクトルデータメモリからの入力ベクトルデータサンプルセットとシフトされた入力ベクトルデータサンプルセットとを受信し、実行ユニットに供給するための複数のパイプラインレジスタを備える、概略図。

【図 8】フィルタベクトル処理動作の間に入力ベクトルデータサンプルセット内の入力ベクトルデータサンプルをシフトするためのパイプラインレジスタの中のレーン内およびレーン間のルーティングを含む、データレーン内のパイプラインレジスタの例示的な詳細を示す、図 7のタップ付き遅延線のより例示的な詳細を示す概略図。

30

【図 9 A】例示的な 8 タップフィルタベクトル処理動作の第 1 のフィルタタップ実行の一部として、図 4のV P E内のプライマリタップ付き遅延線に最初に記憶された入力ベクトルデータサンプルセットの概略図。

【図 9 B】レジスタファイルに記憶されたフィルタタップ係数、および図 9 Aに示された例示的な 8 タップフィルタベクトル処理動作フィルタベクトル処理動作の第 1 のフィルタタップ実行の一部として、図 4のV P E内のシャドウタップ付き遅延線に最初に記憶されたシャドウ入力ベクトルデータサンプルセットの概略図。

【図 9 C】例示的な 8 タップフィルタベクトル処理動作の第 2 のフィルタタップ実行の一部として、図 4のV P E内のプライマリタップ付き遅延線およびシャドウタップ付き遅延線に記憶された、シフトされた入力ベクトルデータサンプルセット、ならびにレジスタファイルに記憶されたフィルタタップ係数の概略図。

40

【図 9 D】例示的な 8 タップフィルタベクトル処理動作の第 8 のフィルタタップ実行の一部として、図 4のV P E内のプライマリタップ付き遅延線およびシャドウタップ付き遅延線に記憶された、シフトされた入力ベクトルデータサンプルセット、ならびにレジスタファイルに記憶されたフィルタタップ係数の概略図。

【図 1 0】例示的な 8 タップフィルタベクトル処理動作が完全に実行された後の図 4のV P E内の実行ユニットの累算器のコンテンツの概略図。

【図 1 1】再フェッチおよび電力消費が低減される精度相関 / 共分散ベクトル処理動作を

50

提供するために、シーケンス番号データを用いて処理されるべきシフトされた入力ベクトルデータサンプルセットを受信し、実行ユニットに供給するためにタップ付き遅延線を利用する例示的なVPEの概略図。

【図12A】例示的な相関/共分散ベクトル処理動作に従って、インターリーブされたオンタイムおよび後発の入力ベクトルデータサンプルセットがフェッチされる、図11のVPE内で並列に実行され得る例示的な相関/共分散ベクトル処理動作を示すフローチャート。

【図12B】例示的な相関/共分散ベクトル処理動作に従って、インターリーブされたオンタイムおよび後発の入力ベクトルデータサンプルセットがフェッチされる、図11のVPE内で並列に実行され得る例示的な相関/共分散ベクトル処理動作を示すフローチャート。

10

【図13】図11のVPE内のレジスタファイルに記憶された相関/共分散入力ベクトルデータサンプルセットの概略図。

【図14】図11のVPE内に設けられ得る例示的なタップ付き遅延線とオプションのシャドウタップ付き遅延線とを示す概略図であって、例示的なタップ付き遅延線が、各々、VPEによって実行される相関/共分散ベクトル処理動作の間に、ベクトルデータメモリからの入力ベクトルデータサンプルセットとシフトされた入力ベクトルデータサンプルセットとを受信し、実行ユニットに供給するための複数のパイプラインレジスタを備える、概略図。

【図15A】相関/共分散ベクトル処理動作の第1の処理ステージの一部として、図11のVPE内のプライマリタップ付き遅延線に最初に供給されたベクトルデータファイルからの入力ベクトルデータサンプルセットの概略図。

20

【図15B】相関/共分散ベクトル処理動作の第1の処理ステージの一部として、図11のVPE内のシャドウタップ付き遅延線に最初に供給されたベクトルデータファイルからのシャドウ入力ベクトルデータサンプルセットの概略図。

【図15C】相関/共分散ベクトル処理動作の第2の処理ステージの一部として、図11のVPE内のプライマリタップ付き遅延線およびシャドウタップ付き遅延線に記憶された、シフトされた入力ベクトルデータサンプルセット、ならびにレジスタファイルに記憶されたシフトされた入力ベクトルデータサンプルセットの概略図。

【図15D】相関/共分散ベクトル処理動作の第14の処理ステージの一部として、図11のVPE内のプライマリタップ付き遅延線およびシャドウタップ付き遅延線に記憶された、シフトされた入力ベクトルデータサンプルセット、ならびにレジスタファイルに記憶されたシフトされた入力ベクトルデータサンプルセットの概略図。

30

【図16】例示的な相関/共分散ベクトル処理動作が完全に実行された後の図11のVPE内の実行ユニットの累算器のコンテンツの概略図。

【図17A】別々に記憶された、結果として生じるフィルタ出力ベクトルデータサンプルの実数成分および虚数成分に記憶された、結果として生じるフィルタ出力ベクトルデータサンプルセットを示す例示的なベクトルデータファイルの図。

【図17B】別々に記憶されたその偶数および奇数の、結果として生じるフィルタ出力ベクトルデータサンプルとともに記憶された、結果として生じるフィルタ出力ベクトルデータサンプルセットを示す例示的なベクトルデータファイルの図。

40

【図18A】符号付き複素数の16ビットフォーマットでVPEのベクトルデータファイルに記憶されたベクトルデータサンプルセットの例示的なインターリーブされたベクトルデータサンプルの図。

【図18B】符号付き複素数の8ビットフォーマットでVPEのベクトルデータファイルに記憶されたベクトルデータサンプルセットの例示的なインターリーブされたベクトルデータサンプルの図。

【図19】ベクトル処理動作を実行するための少なくとも1つの実行ユニットにフォーマット変換された入力ベクトルデータサンプルセットを供給するために、入力ベクトルデータサンプルセットがベクトルデータファイルから再フェッチされる必要なしに、ベクトル

50

データファイルと少なくとも1つの実行ユニットとの間の少なくとも1つの入力データフローパスにおいて、入力ベクトルデータサンプルセットのインフライトフォーマット変換を提供するように構成されたフォーマット変換回路を利用する例示的なVPEの概略図。

【図20】図19のVPEにおいて実行され得る、ベクトルデータファイルと少なくとも1つの実行ユニットとの間の少なくとも1つの入力データフローパスにおける入力ベクトルデータサンプルセットの例示的なインフライトフォーマット変換を示すフローチャート。

【図21】図19のVPE内のタップ付き遅延線と実行ユニットとの間に設けられた例示的なフォーマット変換回路の概略図であって、フォーマット変換回路が実行ユニットへの入力データフローパス内のタップ付き遅延線によって供給される入力ベクトルデータサンプルセットのインフライトフォーマット変換を提供するように構成される、概略図。

10

【図22】実行ユニットにおける受信前に入力データフローパス内で入力ベクトルデータサンプルセットのインフライトフォーマット変換を提供するために、図19のVPEにプログラミングを提供する例示的なベクトル命令データフォーマットを示す図。

【図23】並び替えられた、結果として生じる出力データサンプルセットを供給し記憶するために、結果として生じる出力ベクトルデータサンプルセットが少なくとも1つのベクトルデータファイルに記憶されずに、少なくとも1つの実行ユニットと少なくとも1つのベクトルデータファイルとの間の少なくとも1つの出力データフローパスにおいて、結果として生じる出力ベクトルデータサンプルセットのインフライト並び替えを提供するように構成された並び替え回路を利用する例示的なVPEの概略図。

20

【図24】ベクトルデータファイルに並び替えた形式で記憶される図23のVPE内のベクトルデータファイルと少なくとも1つの実行ユニットとの間の少なくとも1つの出力データフローパスにおける出力ベクトルデータサンプルセットの例示的なインフライトデインターリーピングを示すフローチャート。

【図25】ベクトルデータファイルに記憶された出力ベクトルデータサンプルセットのインフライト並び替えを提供するために、実行ユニットとベクトルデータファイルとの間の出力データフローパス内の並び替え回路を利用する例示的なVPEの概略図。

【図26】図26Aは、通信信号を表す例示的なベクトルデータサンプルシーケンスの図である。図26Bは、例示的な符号分割多元接続(CDMA)チップシーケンスの図である。

図26Cは、図26BのCDMAチップシーケンスで拡散された後の図26Aのベクトルデータサンプルシーケンスの図である。図26Dは、図26Aの元のベクトルデータサンプルシーケンスを復元するために、図26BのCDMAチップシーケンスで図26Cの拡散されたベクトルデータサンプルシーケンスを逆拡散する図である。

30

【図27】逆拡散された、結果として生じる出力ベクトルデータサンプルセットを供給し記憶するために、結果として生じる出力ベクトルデータサンプルセットが少なくとも1つのベクトルデータファイルに記憶されずに、少なくとも1つの実行ユニットと少なくとも1つのベクトルデータファイルとの間の少なくとも1つの出力データフローパスにおいて、結果として生じる出力ベクトルデータサンプルセットの逆拡散を提供するように構成された逆拡散回路を利用する例示的なVPEの概略図。

40

【図28】少なくとも1つのベクトルデータファイル内に逆拡散された、結果として生じる出力ベクトルデータサンプルセットを供給し記憶するために、図27のVPE内の少なくとも1つのベクトルデータファイルと少なくとも1つの実行ユニットとの間の少なくとも1つの出力データフローパスにおける、結果として生じる出力ベクトルデータサンプルセットの例示的な逆拡散を示すフローチャート。

【図29】少なくとも1つのベクトルデータファイル内に逆拡散された、結果として生じる出力ベクトルデータサンプルセットを供給し記憶するために、結果として生じる出力ベクトルデータサンプルセットの逆拡散を提供する、図27のVPE内の少なくとも1つの実行ユニットと少なくとも1つのベクトルデータファイルとの間の出力データフローパス内の例示的な逆拡散回路の概略図。

50

【図30】マージされるべき例示的なベクトルデータサンプルとマージされた、結果として生じるベクトルデータサンプルとを示す図。

【図31】マージされた、結果として生じる出力ベクトルデータサンプルセットを供給し記憶するために、結果として生じる出力ベクトルデータサンプルセットが少なくとも1つのベクトルデータファイルに記憶されず、少なくとも1つの実行ユニットと少なくとも1つのベクトルデータファイルとの間の少なくとも1つの出力データフローパスにおいて、結果として生じる出力ベクトルデータサンプルセットのマージングを提供するように構成されたマージ回路を利用する例示的なVPEの概略図。

【図32】ベクトルデータファイル内に加算マージされた、結果として生じる出力ベクトルデータサンプルセットを供給し記憶するために、図31のVPE内のベクトルデータファイルと少なくとも1つの実行ユニットとの間の少なくとも1つの出力データフローパスにおける、結果として生じる出力ベクトルデータサンプルセットの例示的な加算マージングを示すフローチャート。

【図33】結果として生じる出力ベクトルデータサンプルセットの加算マージングと、ベクトルデータファイル内への加算マージされた、結果として生じる出力ベクトルデータサンプルセットの記憶とを提供する、図31のVPE内の実行ユニットとベクトルデータファイルとの間の出力データフローパス内の例示的なマージ回路の概略図。

【図34】結果として生じる出力ベクトルデータサンプルセットの最大/最小マージングと、ベクトルデータファイル内への最大/最小マージされた、結果として生じる出力ベクトルデータサンプルセットの記憶とを提供する、図31のVPE内の実行ユニットとベクトルデータファイルとの間の出力データフローパス内の例示的なマージ回路の概略図。

【図35】VPE内に設けられ得る例示的なベクトル処理ステージの概略図であって、ベクトル処理ステージのうちのいくつかがプログラム可能なデータパス構成を有する例示的なベクトル処理ブロックを含む、概略図。

【図36】各々がプログラム可能なデータパス構成を有し、図35の例示的なVPE内の様々なベクトル処理ステージ内に設けられる、乗算器ブロックおよび累算器ブロックの例示的なベクトル処理を示すフローチャート。

【図37】図35のVPEのベクトル処理ステージ内に設けられる複数の乗算器ブロックのより詳細な概略図であって、複数の乗算器ブロックが特定の様々なタイプのベクトル乗算演算を実行するために複数のモードでプログラムされ得るように、複数の乗算器ブロックが各々プログラム可能なデータパス構成を有する、概略図。

【図38】8ビット×8ビットの入力ベクトルデータサンプルセットおよび16ビット×16ビットの入力ベクトルデータサンプルセットについての乗算演算を提供するようにプログラムされることが可能なプログラム可能なデータパス構成を有する、図37の複数の乗算器ブロックの中のある乗算器ブロックの内部構成要素の概略図。

【図39】図38のVPE内の乗算器ブロックおよび累算器ブロックの一般化された概略図であって、累算器ブロックが冗長桁上げ伝搬を低減するために冗長桁上げ保存フォーマットを利用する桁上げ保存累算器構造を利用する、概略図。

【図40】図35のVPE内に設けられた図39の累算器ブロックの例示的な内部構成要素の詳細な概略図であって、累算器ブロックが冗長桁上げ保存フォーマットを用いて特定の様々なタイプのベクトル累算演算を実行するために複数のモードでプログラムされ得るように、累算器ブロックがプログラム可能なデータパス構成を有する、概略図。

【図41】本明細書で開示された実施形態による、ベクトル処理回路とベクトル処理動作とを提供するために、本明細書で開示されたVPEを含むことができるベクトルプロセッサを含むことができる、例示的なプロセッサベースシステムのブロック図。

【発明を実施するための形態】

【0024】

[0075]ここで図面を参照すると、本開示のいくつかの例示的な実施形態が記載される。「例示的」という単語は、本明細書において、「例、事例、または例示として働くこと」を意味するために使用される。本明細書で「例示的」と記載されたいかなる実施形態も、

10

20

30

40

50

必ずしも他の実施形態より好ましいか、または有利であると解釈されるべきであるとは限らない。

【 0 0 2 5 】

[0076]再フェッチおよび電力消費が低減される精度フィルタベクトル処理動作を供給するためのタップ付き遅延線を利用するベクトル処理エンジン（V P E）が開示される。関連するベクトル処理システムおよび方法がさらに開示される。本願に開示されるV P Eは、フィルタ係数を持つ入力ベクトルデータサンプルセットの乗算および結果の累算を含む。電力消費を低減するために、メモリからの入力ベクトルデータサンプルの再フェッチを最小化するために、タップ付き遅延線は、ベクトルデータファイルとV P E中の実行ユニットとの間のデータフローパス中に含まれる。タップ付き遅延線は、フィルタベクトル処理動作を実行するための実行ユニットに入力ベクトルデータサンプルセットを受信し供給するように構成される。タップ付き遅延線は、各フィルタ遅延タップのための入力ベクトルデータサンプルをシフトすることと、実行ユニットにシフトされた入力ベクトルデータサンプルセットを供給することとを行うようにさらに構成され、シフトされた入力ベクトルデータサンプルセットは、フィルタベクトル処理動作の各処理ステージの間のベクトルデータファイルから再フェッチされることはない。この方法では、低減される再フェッチ動作は、電力消費を低減する。さらにV P E中のデータフローパスの効率は、メモリからの入力ベクトルデータサンプルセットの再フェッチ遅延によって制限されない。このシフトされた入力ベクトルデータサンプルセットは、実行ユニットに局在するように設けられる。実行ユニット中のベクトル処理は、データフローの制限ではなく、コンピュータリソースのみによって制限される。さらに、フィルタベクトル処理動作は、実行ユニット中の中間フィルタステージのための累算がメモリ中に格納されないため、より正確であり、実行ユニットがメモリから再フェッチされるべきシフトされた入力フィルタサンプルデータを待つ間に、結果として丸められる。

【 0 0 2 6 】

[0077]さらに、本願に開示されるタップ付き遅延線を利用するV P E中のフィルタベクトル処理動作は、フィルタ遅延タップと同じ数の乗算器を要求すること無しに完全に並列化されることができる。これは、タップ付き遅延線がベクトルデータファイルからの入力ベクトルデータサンプルセットの再フェッチからの遅延をこうむること無しにフィルタ処理のための実行ユニットに、シフトされた入力ベクトルデータサンプルを供給することができるからである。フィルタタップ遅延より少ない乗算器でフィルタベクトル処理動作の並列化を許容することによって、V P Eのデータフローパスは、より少ない乗算器を要求するベクトル処理動作の他のタイプのために予約された乗算器の数を簡単にスケールダウンするように再プログラムされることができ、そのため、V P Eの乗算器は、効率的な目的のために過剰利用されることはない。例えば、V P E中のデータフローパスがベクトル処理動作の他のタイプのために要求されるより、フィルタベクトル処理動作のためにより多くの乗算器の数を利用するように構成される場合、追加の乗算器は、これらの他のベクトル処理動作の実行の間使用され得ず、その結果、V P Eの乗算器処理リソースを過剰利用することになる。

【 0 0 2 7 】

[0078]この関連で、図2は、ベクトル処理エンジン（V P E）22とも呼ばれる例示的なベクトル処理ユニット22を含むベースバンドプロセッサ20の概略図である。下記でより詳細に説明されるように、V P E 22は、実行ユニット84と、本明細書で開示される例示的なベクトル処理動作を含むベクトル処理動作を提供する他の特定の例示的な回路および機能とを含む。ベースバンドプロセッサ20およびそのV P E 22は、半導体ダイ24内に設けられ得る。この実施形態では、下記でより詳細に説明されるように、ベースバンドプロセッサ20は、様々なプログラム可能なデータパス構成を提供するためにプログラムされ得るプログラム可能なデータパス26を含む共通のV P E 22を含む。このようにして、V P E 22内の実行ユニット84とベクトルデータファイル82との間のプログラム可能なデータパス26は、ベースバンドプロセッサ20内に別々のV P E 22を設

ける必要なしに、様々な動作モードで様々な特定のタイプのベクトル処理動作を提供するようにプログラムおよび再プログラムされ得る。

【 0 0 2 8 】

[0079]図3で始まる効率的な処理について、この開示におけるVPE22によって提供されるように構成された特定の回路とベクトル処理動作とを説明する前に、図2のベースバンドプロセッサ20の構成要素が最初に記載される。この非限定的な例におけるベースバンドプロセッサ20は、512ビットベクトルプロセッサである。ベースバンドプロセッサ20は、ベースバンドプロセッサ20内のベクトル処理を提供するVPE22をサポートするために、VPE22に加えて構成要素を含む。ベースバンドプロセッサ20は、ベクトルユニットデータメモリ(LMEM)32からベクトルデータ30を受信し記憶するように構成された、ベクトルデータファイル82としても知られる、ベクトルレジスタを含む。たとえば、ベクトルデータ30はXビット幅であり、「X」は設計選択に従って定義される(たとえば、512ビット)。ベクトルデータ30は、ベクトルデータサンプルセット34に分割され得る。非限定的な例として、ベクトルデータ30は256ビット幅であり得るし、より小さいベクトルデータサンプルセット34(Y)~34(0)を備える場合がある。いくつかのベクトルデータサンプルセット34(Y)~34(0)は、例として16ビット幅であり得るし、ベクトルデータサンプルセット34(Y)~34(0)の他は、32ビット幅であり得る。VPE22は、高度の並列性を達成するために、VPE22に並列に供給されるいくつかの選ばれたベクトルデータサンプルセット34(Y)~34(0)に対するベクトル処理を提供することが可能である。ベクトルデータファイル82はまた、VPE22がベクトルデータ30を処理するとき生成される結果を記憶するように構成される。いくつかの実施形態では、VPE22は、より速いベクトル命令実行時間を提供するようにレジスタ書き込みを低減するために、ベクトルデータファイル82内に中間ベクトル処理結果を記憶しないように構成される。この構成は、スカラー処理デジタル信号プロセッサ(DSP)などの、レジスタに中間結果を記憶するスカラー処理エンジンによって実行されるスカラー命令とは反対である。

【 0 0 2 9 】

[0080]図2のベースバンドプロセッサ20は、ベクトル命令の条件付き実行において使用するためにVPE22に条件を与えるように、およびベクトル命令実行の結果として更新された条件を記憶するように構成された条件レジスタ36も含む。ベースバンドプロセッサ20はまた、累算レジスタ38と、グローバルレジスタを含むグローバルレジスタファイル40と、アドレスレジスタ42とを含む。累算レジスタ38は、ベクトルデータ30に対していくつかの特殊な演算を実行する結果として累算された結果を記憶するために、VPE22によって使用されるように構成される。グローバルレジスタファイル40は、VPE22によってサポートされるいくつかのベクトル命令のためのスカラーオペランドを記憶するように構成される。アドレスレジスタ42は、ベクトルユニットデータメモリ32からベクトルデータ30を取り出し、ベクトルユニットデータメモリ32にベクトル処理結果を記憶するために、ベクトルロードによってアドレス指定可能なアドレスを記憶し、VPE22によってサポートされる命令を記憶するように構成される。

【 0 0 3 0 】

[0081]引き続き図2を参照すると、この実施形態におけるベースバンドプロセッサ20は、VPE22によって提供されるベクトル処理に加えて、ベースバンドプロセッサ20においてスカラー処理を提供する(「整数ユニット」とも呼ばれる)スカラープロセッサ44も含む。高効率演算のために実行される命令のタイプに基づいて、ベクトル命令演算とスカラー命令演算の両方をサポートするように構成された中央処理装置(CPU)を設けることが望ましい場合がある。この実施形態では、スカラープロセッサ44は、非限定的な例として、32ビット縮小命令セットコンピューティング(RISC)スカラープロセッサである。スカラープロセッサ44は、この例では、スカラー命令処理をサポートするための算術論理ユニット(ALU)46を含む。ベースバンドプロセッサ20は、プログラムメモリ50から命令をフェッチし、フェッチされた命令を復号し、命令タイプに基

づいて、スカラープロセッサ 44 に、またはベクトルデータパス 53 を通って VPE 22 に、フェッチされた命令を向けるように構成された命令ディスパッチ回路 48 を含む。スカラープロセッサ 44 は、スカラー命令を実行するときにスカラープロセッサ 44 によって使用される汎用レジスタ 54 を含む。スカラー命令実行のためにスカラープロセッサ 44 によるアクセス用に、メインメモリから汎用レジスタ 54 にデータを供給するように、整数ユニットデータメモリ (DMM) 56 がベースバンドプロセッサ 20 に含まれる。DMM 56 は、非限定的な例としてキャッシュメモリであり得る。ベースバンドプロセッサ 20 は、メモリコントローラデータパス 62 を通ってメインメモリへのアクセスを求めるベクトル命令をスカラープロセッサ 44 が実行しているときに汎用レジスタ 54 からメモリアドレスを受信するように構成されたメモリコントローラレジスタ 60 を含むメモリコントローラ 58 も含む。

10

【0031】

[0082] VPE 22 によるベクトル命令処理によってサポートされることが望ましい場合がある特殊ベクトル処理動作の 1 つのタイプは、フィルタリングである。フィルタ動作は、サンプリングされた入力時間関数の重畳の量子化時間領域表現と、フィルタの重み付け関数の表現とを計算する。時間領域内の重畳は、周波数領域内の乗算に対応する。このように、デジタルフィルタは、間隔が均一なサンプル間隔で実行される乗算および加算の拡張シーケンスにより、VPE 22 において実現され得る。たとえば、ディスクリット有限インパルス応答 (FIR) フィルタは、フィルタ関数を計算するために、「Y」計算フィルタ係数を有する遅延線上の遅延タップの有限数 (Y) を使用して実施され得る。

20

【0032】

[0083] この関連で、図 3 は、図 2 の VPE 22 におけるフィルタベクトル処理動作を介してサポートされることが望ましい場合がある、例示的なディスクリット FIR フィルタ 64 の概略図である。デジタル化入力信号 66 ($x[n]$) は、「フィルタ遅延タップ」68 (1) ~ 68 (Y - 1) と呼ばれる遅延構造を通してデジタル化入力信号サンプル ($x[0]$, $x[1]$, ..., $x[n]$) を渡すことによってフィルタリングされ得る。フィルタ遅延タップ 68 (1) ~ 68 (Y - 1) は、フィルタサンプル被乗数 72 (0) ~ 72 (Y - 1) を供給するために、すべてのデジタル化入力信号サンプル (すなわち、 $x[0]$, $x[1]$, ..., $x[n]$) が各々フィルタ係数 ($h[0]$ ~ $h[Y - 1]$) によって乗算される (すなわち、 $h[1] * x[n - 1]$) ために、クロックされたデジタル化入力信号サンプル (すなわち、 $x[0]$, $x[1]$, ..., $x[n]$) を乗算器 70 (0) ~ 70 (Y - 1) の中にシフトする。フィルタサンプル被乗数 72 (0) ~ 72 (Y - 1) は、結果として生じるフィルタ処理された出力信号 76 (すなわち、 $y[n]$) を供給するために、加算器 (すなわち、アダー) 74 (1) ~ 74 (Y - 1) によって一緒に加算される。このように、図 3 のディスクリット FIR フィルタ 64 は以下のように要約され得る。

30

【0033】

【数 1】

$$y[n] = \sum_{l=0}^{Y-1} h[l] * x[n-l]$$

40

【0034】

ここで、

n は入力信号サンプルの数であり、

$x[n]$ はデジタル化入力信号 66 であり、

$y[n]$ は、結果として生じるフィルタ処理された出力信号 76 であり、

$h[l]$ はフィルタ係数であり、

Y はフィルタ係数の数である。

フィルタ係数 $h[l]$ は複素数であり得る。一態様では、VPE 22 は、(たとえば、グローバルレジスタファイル 40 から) フィルタ係数を受信することができる。VPE 22

50

は、FIRフィルタ関数を実行するために受信されたフィルタ係数を直接使用することができ、その場合、上記の式におけるフィルタ係数 $h(1)$ は、受信されたフィルタ係数を表すことができる。代替として、VPE22は、FIRフィルタ関数を実行するためにそれらを使用する前に、受信されたフィルタ係数の複素共役を計算することができ、その場合、上記の式におけるフィルタ係数 $h(1)$ は、受信されたフィルタ係数の共役を表すことができる。

【0035】

[0084]図3の上記のディスクリットFIRフィルタ64は、以下のように書き直され得る。

$$y[n] = x[n] * h_0 + x[n-1] * h_1 + \dots + x[n-7] * h_7$$

10

【0036】

[0085]しかしながら、図3のディスクリットFIRフィルタ64などのフィルタリング演算は、ベクトルプロセッサにおいて提供される特殊データフローパスに起因して、ベクトルプロセッサにおいて並列化することは困難であり得る。フィルタリングされるべき入力ベクトルデータサンプルセット（たとえば、ベクトル化されたデジタル化入力信号66）が、フィルタ遅延タップ（たとえば、68(1)～68(Y-1)）の間にシフトされると、入力ベクトルデータサンプルセットはベクトルデータファイルから再フェッチされ、したがって電力消費が増大し、スループットが低減される。ベクトルデータファイルからの入力ベクトルデータサンプルセットの再フェッチを最小化するために、ベクトルプロセッサ内のデータフローパスは、効率的な並列化処理のために、フィルタ遅延タップ（たとえば、68(1)～68(Y-1)）と同じ数の乗算器（たとえば、70(0)～70(Y-1)）を設けるように構成される可能性がある。しかしながら、他のベクトル処理動作は、より少ない乗算器しか必要としない場合があり、それにより、データフローパス内の乗算器の非効率的なスケーリングおよび過剰利用がもたらされる。スケラビリティを提供するために、乗算器の数がフィルタ遅延タップの数よりも少なくなるように削減された場合、フィルタ処理の様々なフェーズに対して同じ入力ベクトルデータサンプルセットを取得するために、メモリにより多くの再フェッチが必要とされることによって、並列化が制限される。

20

【0037】

[0086]この関連で、図4は、図2のVPE22として提供され得る例示的なVPE22(1)の概略図である。下記でより詳細に記載されるように、図4のVPE22(1)は、ベクトルデータサンプルの再フェッチが除去または低減され、電力消費が低減される、VPE22(1)内の精度フィルタベクトル処理動作を提供する。精度フィルタベクトル処理動作は、ベクトルデータサンプルの再フェッチを必要とし、それにより結果として電力消費が増大する、中間結果の記憶を必要とするフィルタベクトル処理動作と比較して、VPE22(1)において提供され得る。ベクトルデータファイルからの入力ベクトルデータサンプルの再フェッチを除去または最小化して、電力消費を低減し、処理効率を改善するために、VPE22(1)内のベクトルデータファイル82(0)～82(X)と（「EU」とも標記される）実行ユニット84(0)～84(X)との間の入力データフローパス80(0)～80(X)にタップ付き遅延線78が含まれる。「X」+1は、この例におけるベクトルデータサンプルの処理用にVPE22(1)内に設けられる並列入力データレーンの最大数である。タップ付き遅延線78は、ベクトルデータファイル82(0)～82(X)の対応するサブセットまたはすべてから入力ベクトルデータサンプルセット86(0)～86(X)の入力ベクトルデータサンプル86のサブセットまたはすべてとして、タップ付き遅延線入力88(0)～88(X)上で入力ベクトルデータサンプルセット86(0)～86(X)を受信するように構成される。入力ベクトルデータサンプルセット86(0)～86(X)は、この例では86(0)、86(1)、...、および86(X)である、「X+1」個の入力ベクトルデータサンプル86から構成される。

30

40

【0038】

50

[0087]引き続き図4を参照すると、タップ付き遅延線78は、フィルタベクトル処理動作のために実行ユニット84(0)~84(X)によって処理されるべき、ベクトルデータファイル82(0)~82(X)からフェッチされた入力ベクトルデータサンプルセット86(0)~86(X)を記憶する。下記の図6および図7に関して下記でより詳細に説明されるように、タップ付き遅延線78は、実行ユニット84(0)~84(X)にシフトされた入力ベクトルデータサンプルセット86S(0)~86S(X)を供給するために、VPE22(1)によって実行されるべきフィルタベクトル命令に従うフィルタベクトル処理動作のフィルタ遅延タップ(すなわち、フィルタ処理ステージ)ごとに、入力ベクトルデータサンプルセット86(0)~86(X)をシフトするように構成される。シフトされた入力ベクトルデータサンプル86Sのすべては、シフトされた入力ベクトルデータサンプルセット86S(0)~86S(X)を備える。タップ付き遅延線78は、フィルタベクトル処理動作中、実行ユニット84(0)~84(X)の実行ユニット入力90(0)~90(X)にシフトされた入力ベクトルデータサンプル86S(0)~86S(X)を供給する。このようにして、フィルタベクトル処理動作のフィルタタップのためのシフトされた入力ベクトルデータサンプルセット86S(0)~86S(X)に対して実行される動作に基づく中間フィルタ結果は、VPE22(1)によって実行されるフィルタベクトル処理動作の各処理ステージの間、記憶、シフト、およびベクトルデータファイル82(0)~82(X)から再フェッチされる必要がない。このように、タップ付き遅延線78は、VPE22(1)によって実行されるフィルタベクトル処理動作についての電力消費を低減し、処理効率を上げることができる。

【0039】

[0088]「ベクトル処理ステージ」とも呼ばれるVPE22(1)内の処理ステージは、特定のタスクまたは動作を実行するように設計された回路と関連するベクトルデータパスとを備える。ベクトル処理動作は、いくつかの異なる処理ステージにおいて、VPE22(1)によって実行される場合がある。各処理ステージは、VPE22(1)の1つまたは複数のクロックサイクルにわたって実行される場合がある。その結果、VPE22(1)内のベクトル処理動作の実行は、ベクトル処理動作の各処理ステージが各々1つまたは複数のクロックサイクルを消費する可能性があるため、完了するために多くのクロックサイクルを要する可能性がある。たとえば、処理ステージは、図4のVPE22(1)内のタップ付き遅延線78の中に入力ベクトルデータサンプルセット86(0)~86(X)をフェッチすることを含む場合がある。VPE22(1)内のベクトル処理ステージはパイプライン化され得る。

【0040】

[0089]実行ユニット84(0)~84(X)は、フェッチされた入力ベクトルデータサンプルセット86(0)~86(X)を処理する1つまたは複数のパイプラインステージを含む場合がある。たとえば、実行ユニット84(0)~84(X)内の1つのパイプラインステージは、累算演算を実行するように構成された累算器から構成される累算ステージを含む場合がある。別の例として、実行ユニット84(0)~84(X)内の別のパイプラインステージは、乗算演算を実行するように構成された乗算器から構成される乗算ステージを含む場合がある。

【0041】

[0090]引き続き図4を参照すると、実行ユニット84(0)~84(X)は、フィルタベクトル処理動作のための図2のグローバルレジスタファイル40に記憶されたフィルタ係数92(0)~92(Y-1)の中からフィルタ係数92を受信する、ここで、「Y」はフィルタベクトル処理動作のためのフィルタ係数の数に等しい場合がある。実行ユニット84(0)~84(X)は、各々、実行ユニット84(0)~84(X)内に中間フィルタベクトルデータ出力サンプルを供給するために、ベクトルフィルタ処理動作の各処理ステージの間に、受信されたフィルタ係数92(0)、90(1)、...90(Y-1)のうちの1つを、シフトされた入力ベクトルデータサンプルセット86S(0)~86S(X)のシフトされた入力ベクトルデータサンプル86S(0)、86S(1)、...

・ 86S(X)と乗算するように構成される。中間フィルタベクトルデータ出力サンプルセットは、実行ユニット84(0)~84(X)の各々において累算される(すなわち、前に累算されたフィルタ出力ベクトルデータサンプルが現在の累算されたフィルタ出力ベクトルデータサンプルに加算される)。これにより、シフトされた入力ベクトルデータサンプルセット86S(0)~86S(X)内のシフトされた入力ベクトルデータサンプル86S(0)、86S(1)、...86S(X)ごとに、それぞれ、出力データフローパス98(0)~98(X)上の実行ユニット出力96(0)~96(X)上に実行ユニット84(0)~84(X)によって供給される、最終的な、結果として生じるフィルタ出力ベクトルデータサンプルセット94(0)~94(X)がもたらされる。結果として生じるフィルタ出力ベクトルデータサンプルセット94(0)~94(X)は、この例では94(0)、94(1)、...、および94(X)である、「X+1」個の、結果として生じるフィルタ出力ベクトルデータサンプル94から構成される。実行ユニット84(0)~84(X)によって生成された中間フィルタベクトルデータ出力サンプルセットを記憶しシフトする必要なしに、結果として生じるフィルタ出力ベクトルデータサンプルセット94(0)~94(X)は、VPE22(1)によるさらなる使用および/または処理のために、それぞれのベクトルデータファイル82(0)~82(X)に戻されて記憶される。

10

【0042】

[0091]引き続き図4を参照すると、下記でより詳細に説明されるように、タップ付き遅延線78は、処理されているベクトル命令に従って制御されるようにプログラム可能である。フィルタベクトル命令が処理されていない場合、タップ付き遅延線78は、ベクトルデータファイル82(0)~82(X)と実行ユニット84(0)~84(X)との間の入力データフローパス80(0)~80(X)に含まないようにプログラムされ得る。この実施形態では、タップ付き遅延線78は、フィルタベクトル処理動作のフィルタタップごとにシフトされた入力ベクトルデータサンプルセット86S(0)~86S(X)を供給するために、ベクトルデータファイル82(0)~82(X)から受信された入力ベクトルデータサンプルセット86(0)~86(X)をロードしシフトするように構成される。このように、シフトされた入力ベクトルデータサンプルセット86S(0)~86S(X)は、フィルタベクトル処理動作のフィルタタップの実行のために、実行ユニット84(0)~84(X)に供給され得る。タップ付き遅延線78がないと、フィルタベクトル処理動作の次のフィルタタップのために、実行ユニット84(0)~84(X)にシフトされた中間入力ベクトルデータサンプルセットを再び供給するために、別個のシフティングプロセスが実行される必要があるはずであり、それにより、遅延時間が増大し、さらなる電力が消費される。さらに、フィルタベクトル処理動作中、ベクトルデータファイル82(0)~82(X)からのシフトされた入力ベクトルデータサンプルセット86S(0)~86S(X)の再フェッチ遅延によって、VPE22(1)内の入力データフローパス80(0)~80(X)および出力データフローパス98(0)~98(X)の効率が制限される。

20

30

【0043】

[0092]シフトされた入力ベクトルデータサンプルセット86S(0)~86S(X)は、実行ユニット84(0)~84(X)に局在するタップ付き遅延線78によって供給される。実行ユニット84(0)~84(X)におけるベクトル処理は、データフローの制限ではなく、コンピュータリソースのみによって制限される。これは、シフトされた入力ベクトルデータサンプルセット86S(0)~86S(X)がベクトルデータファイル82(0)~82(X)からフェッチされるまで待つ必要なしに、実行ユニット84(0)~84(X)が、ベクトル処理動作を実行するためにシフトされた入力ベクトルデータサンプルセット86S(0)~86S(X)を受信することに、連続して、または実質的に連続して忙しいことを意味する。

40

【0044】

[0093]さらに、図4のVPE22(1)によって実行されるフィルタベクトル処理動作

50

は、タップ付き遅延線 78 を利用することによってより精密になり得るが、これは、実行ユニット 84 (0) ~ 84 (X) 内の中間フィルタ処理ステージのための出力累算がベクトルデータファイル 82 (0) ~ 82 (X) に記憶される必要がないからである。実行ユニット 84 (0) ~ 84 (X) からベクトルデータファイル 82 (0) ~ 82 (X) への中間出力ベクトルデータサンプルセットの記憶は、丸めをもたらす可能性がある。したがって、次の中間出力ベクトルデータサンプルセットがベクトル処理動作のために実行ユニット 84 (0) ~ 84 (X) に供給されるとき、ベクトル処理動作の各乗算フェーズの間に任意の丸め誤差が伝搬および加算される。対照的に、図 4 の V P E 2 2 (1) の例では、実行ユニット 84 (0) ~ 84 (X) によって計算された中間出力ベクトルデータサンプルセットは、ベクトルデータファイル 82 (0) ~ 82 (X) に記憶される必要がない。実行ユニット 84 (0) ~ 84 (X) は、前の中間出力ベクトルデータサンプルセットを次のフィルタ遅延タップのための中間出力ベクトルデータサンプルセットと累算することができるが、これは、タップ付き遅延線 78 が、処理されるべきベクトル処理動作の間に、実行ユニット 84 (0) ~ 84 (X) にシフトされた入力ベクトルデータサンプルセット 86 S (0) ~ 86 S (X) を供給するからであり、結果は前のフィルタ遅延タップのための前のベクトルデータサンプルセットと累算される。

【 0 0 4 5 】

[0094]引き続き図 4 を参照すると、この実施形態における V P E 2 2 (1) は、並列化処理のための (V L A N E 0 1 0 0 (0) ~ V L A N E X 1 0 0 (X) と標記された) 複数のベクトルデータレーンから構成される。各ベクトルデータレーン 1 0 0 (0) ~ 1 0 0 (X) は、この実施形態では、ベクトルデータファイル 82 と実行ユニット 84 とを含んでいる。例としてベクトルデータレーン 1 0 0 (0) を取り上げると、その中のベクトルデータファイル 82 (0) は、フィルタベクトル処理のために実行ユニット 84 (0) によって受信されるように、入力データフローパス 80 (0) 上に入力ベクトルデータサンプル 86 (0) を供給するように構成される。上記で説明されたように、タップ付き遅延線 78 は、フィルタベクトル処理のために、入力ベクトルデータサンプル 86 (0) をシフトし、シフトされた入力ベクトルデータサンプル 86 S (0) を実行ユニット 84 (0) に供給するために、入力データフローパス 80 (0) 内に設けられる。ベクトルデータファイル 82 (0) はまた、 V P E 2 2 (1) によって処理されるべき現在または次のベクトル命令に従って、必要または所望に応じて、次のベクトル処理動作のためにベクトルデータファイル 82 (0) に戻されて記憶されるべき、出力データフローパス 98 (0) からのフィルタベクトル処理の結果として、実行ユニット 84 (0) によって供給される、結果として生じるフィルタ出力ベクトルデータサンプル 94 (0) を受信するように構成される。

【 0 0 4 6 】

[0095]必要に応じて、任意の数のベクトルデータレーン 1 0 0 (0) ~ 1 0 0 (X) が V P E 2 2 (1) 内に設けられる場合がある。 V P E 2 2 (1) 内に設けられるベクトルデータレーン 1 0 0 (0) ~ 1 0 0 (X) の数は、効率目的のための並列化ベクトル処理対さらなるベクトルデータレーン 1 0 0 (0) ~ 1 0 0 (X) を設けることに伴うさらなる回路、空間、および電力消費についてのトレードオフに基づく場合がある。1つの非限定的な例として、16個のベクトルデータレーン 1 0 0 が V P E 2 2 (1) 内に設けられる場合があり、各ベクトルデータレーン 1 0 0 は、 V P E 2 2 (1) 内の 5 1 2 ビットまでのベクトルデータの並列化処理を提供するために、32ビットのデータ幅能力を有する。

【 0 0 4 7 】

[0096]引き続き図 4 を参照すると、すべてのベクトルデータファイル 82 (0) ~ 82 (X) に適用可能であるが、例としてベクトルデータレーン 1 0 0 (0) 内のベクトルデータファイル 82 (0) を使用して、ベクトルデータファイル 82 (0) により、入力ベクトルデータサンプル 86 (0) の1つまたは複数のサンプルがベクトル処理のために記憶されることが可能になる。 V P E 2 2 (1) によって実行されている特定のベクトル命

令に従う入力ベクトルデータサンプル 86 (0) のプログラミングに応じて、入力ベクトルデータサンプル 86 (0) の幅が設けられる。入力データフローパス 80 (0) の幅は、所与のベクトル命令がタップ付き遅延線 78 および実行ユニット 84 (0) に様々な幅の入力ベクトルデータサンプル 86 (0) を供給するために、クロックサイクルごとを含むベクトル命令ごとにプログラム可能および再プログラム可能である。このようにして、ベクトルデータレーン 100 (0) は、実行されているベクトル命令のタイプに応じて、入力ベクトルデータサンプル 86 (0) の様々な幅の処理を提供するように、プログラムおよび再プログラムされ得る。

【0048】

[0097]たとえば、ベクトルデータファイル 82 (0) は、32 ビット幅であり、同様に 32 ビットまでの幅である入力ベクトルデータサンプル 86 を記憶することが可能であり得る。入力ベクトルデータサンプル 86 (0) は、ベクトルデータファイル 82 (0) の幅全体（たとえば、32 ビット）を消費する場合があるか、またはベクトルデータファイル 82 (0) の幅のより小さいサンプルサイズで供給される場合がある。入力ベクトルデータサンプル 86 (0) のサイズは、VPE22 (1) によって実行されているベクトル命令に基づく、入力ベクトルデータサンプル 86 (0) のサイズ向けの入力データフローパス 80 (0) の構成のプログラミングに基づいて構成され得る。たとえば、入力ベクトルデータサンプル 86 (0) は、1つのベクトル命令のための2つの別々の16ビットベクトルデータサンプルを備える場合がある。別の例として、入力ベクトルデータサンプル 86 (0) は、1つの32ビットベクトルデータサンプルとは対照的に、別のベクトル命令のためのベクトルデータファイル 82 (0) 内の4つの8ビットベクトルデータサンプルを備える場合がある。別の例では、入力ベクトルデータサンプル 86 (0) は、1つの32ビットベクトルデータサンプルを備える場合がある。VPE22 (1) はまた、ベクトル命令ごとに、および/または所与のベクトル命令のクロックサイクルごとに、実行ユニット 84 (0) によりベクトルデータファイル 82 (0) に供給される様々なサイズの、結果として生じるフィルタ出力ベクトルデータサンプル 94 (0) を受信するように、ベクトルデータファイル 82 (0) のための出力データフローパス 98 (0) をプログラムおよび再プログラムすることが可能である。

【0049】

[0098]図4のVPE22 (1) のさらなる詳細および特徴、ならびにこの実施形態における入力データフローパス 80 (0) ~ 80 (X) 内の実行ユニット 84 (0) ~ 84 (X) にシフトされた入力ベクトルデータサンプルセット 86 S (0) ~ 86 S (X) を供給するためのタップ付き遅延線 78 のさらなる説明が次に記載される。この関連で、図5は、例示的なフィルタベクトル命令に従って、タップ付き遅延線 78 を利用する図4のVPE22 (1) において実行され得る例示的なフィルタベクトル処理動作 102 を示すフローチャートである。図6A ~ 図10において提供される例を参照して、図5のフィルタベクトル処理動作 102 において実行される例示的なタスクが記載される。

【0050】

[0099]図5を参照すると、フィルタベクトル命令に従ってフィルタベクトル処理動作 102 において処理されるべき入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) は、フィルタベクトル処理動作 102 のために、ベクトルデータファイル 82 (0) ~ 82 (X) から入力データフローパス 80 (0) ~ 80 (X) の中にフェッチされる（ブロック 104）。図4のVPE22 (1) に関して上記で説明されたように、入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) は、実行ユニット 84 (0) ~ 84 (X) 内のグローバルレジスタファイル 40 から受信されたフィルタ係数 92 (0) ~ 92 (Y - 1) と乗算される。たとえば、図6Aは、グローバルレジスタファイル 40 内のフィルタ係数 92 (0) ~ 92 (Y - 1) （すなわち、h7 ~ h0）を示す。この例では、実行されるべきフィルタベクトル処理動作 102 において8個のフィルタタップを提供する、グローバルレジスタファイル 40 に記憶された8個のフィルタ係数 92 が存在する。この例では、上記で説明された図3のディスクリット FIR フィルタ 64 の式からのフィルタ

ベクトル処理動作 102 は、下記の通りであることに留意されたい。

$$y[n] = x[n] * h_0 + x[n-1] * h_1 + \dots + x[n-7] * h_7$$

【0051】

[00100] 図 6 B は、フィルタベクトル処理動作 102 によってフィルタリングされるべき入力信号を表す、図 4 の V P E 2 2 (1) 内のベクトルデータファイル 8 2 (0) ~ 8 2 (X) に記憶された例示的な入力ベクトルデータサンプルセット 8 6 (0) ~ 8 6 (X) を示す。この例では、サンプル X 0 は最も古いサンプルであり、サンプル X 6 3 はつい最近のサンプルである。言い換えれば、この例では、サンプル X 6 3 は、時間的にサンプル X 0 の後に発生する。ベクトルデータファイル 8 2 (0) ~ 8 2 (X) の各アドレスは 16 ビット幅であるので、ベクトルデータファイル 8 2 (0) ~ 8 2 (X) に記憶された最初の入力ベクトルデータサンプルセット 8 6 (0) ~ 8 6 (X) は、図 6 B に示されたように、ADDRESS 0 および ADDRESS 1 にまたがる。これにより、ベクトルデータファイル 8 2 (0) ~ 8 2 (X) が、図 4 の V P E 2 2 (1) の例における実行ユニット 8 4 (0) ~ 8 4 (X) の 32 ビット幅能力をサポートするために、32 ビット幅の入力ベクトルデータサンプル 8 6 を供給することが可能になる。この関連で、最初の入力ベクトルデータサンプルセット 8 6 (0) ~ 8 6 (X) を備える、合計 512 ビットの各々 8 ビットの幅である 64 個の合計入力ベクトルデータサンプルサブセット（すなわち、X 0 ~ X 63）が存在する。同様に、ADDRESS 2 および ADDRESS 3 は、ベクトルデータファイル 8 2 (0) ~ 8 2 (X) に記憶された別の 2 番目の入力ベクトルデータサンプルセット 8 6 (0) ~ 8 6 (X) を記憶する。図 6 B のこの例では、各ベクトルデータファイル 8 2 (0) ~ 8 2 (X) の 8 個のアドレス (ADDRESS 0 ~ 7) が示され、256 個の合計入力ベクトルデータサンプル 8 6（すなわち、X 0 ~ X 255）を示すが、それは限定的でないことに留意されたい。

【0052】

[00101] フィルタベクトル処理動作 102 に関する入力ベクトルデータサンプルセット 8 6 (0) ~ 8 6 (X) の幅に応じて、ベクトル命令のプログラミングに従うフィルタベクトル処理動作 102 を提供するために、図 4 の V P E 2 2 (1) 内のベクトルデータレーン 100 (0) ~ 100 (X) の 1 つ、いくつか、またはすべてが利用され得る。ベクトルデータファイル 8 2 (0) ~ 8 2 (X) の幅全体が必要な場合、すべてのベクトルデータレーン 100 (0) ~ 100 (X) がフィルタベクトル処理動作 102 に利用され得る。フィルタベクトル処理動作 102 は、フィルタベクトル処理動作 102 に利用され得るベクトルデータレーン 100 (0) ~ 100 (X) のサブセットを必要とするにすぎない場合があることに留意されたい。これは、入力ベクトルデータサンプルセット 8 6 (0) ~ 8 6 (X) の幅がすべてのベクトルデータファイル 8 2 (0) ~ 8 2 (X) の幅よりも小さいからであり得るし、ここで、フィルタベクトル処理動作 102 と並列に実行されるべき他のベクトル処理動作にさらなるベクトルデータレーン 100 を利用することが望ましい。現在の例を説明する目的で、フィルタベクトル処理動作 102 において利用される入力ベクトルデータサンプルセット 8 6 (0) ~ 8 6 (X) が、すべてのベクトルデータレーン 100 (0) ~ 100 (X) を要すると想定する。

【0053】

[00102] 図 5 に戻って参照すると、現在の入力ベクトルデータサンプルセット 8 6 (0) ~ 8 6 (X) としてタップ付き遅延線 7 8 にロードされるために、フェッチされた入力ベクトルデータサンプルセット 8 6 (0) ~ 8 6 (X) が、ベクトルデータファイル 8 2 (0) ~ 8 2 (X) から入力データフローパス 8 0 (0) ~ 8 0 (X) に供給される（ブロック 106）。入力ベクトルデータサンプルセット 8 6 (0) ~ 8 6 (X) は、フィルタベクトル処理動作 102 のために実行ユニット 8 4 (0) ~ 8 4 (X) によって処理されるべき入力ベクトルデータサンプルセット 8 6 (0) ~ 8 6 (X) として、プライマリタップ付き遅延線 7 8 (0) の中にロードされる。プライマリタップ付き遅延線 7 8 (0) の中にロードされた入力ベクトルデータサンプルセット 8 6 (0) ~ 8 6 (X) は、フィルタベクトル処理動作 102 の最初のフィルタタップ動作のためにシフトされない。し

かしながら、上記で説明され、図 7 に関して下記でさらに詳細に説明されるように、タップ付き遅延線 78 の目的は、フィルタベクトル処理動作 102 の次のフィルタタップ動作のために実行ユニット 84 (0) ~ 84 (X) にシフトされた入力ベクトルデータサンプルセット 86 S (0) ~ 86 S (X) を供給するために、入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) のシフトを提供することである。実行ユニット 84 (0) ~ 84 (X) によって実行されるフィルタベクトル処理動作 102 の各処理ステージの間、実行ユニット 84 (0) ~ 84 (X) にシフトされた入力ベクトルデータサンプルセット 86 S (0) ~ 86 S (X) を供給するために、入力ベクトルデータサンプル 86 がプライマリタップ付き遅延線 78 (0) 内でシフトされる。このようにして、入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) は、フィルタベクトル処理動作 102 のフィルタタップ動作ごとに、記憶、ベクトルデータファイル 82 (0) ~ 82 (X) 内でシフト、および再フェッチされる必要がない。

10

【 0054 】

[00103] オプションのシャドウタップ付き遅延線 78 (1) が VPE 22 (1) 内に設けられた場合、次の入力ベクトルデータサンプルセット 86 N (0) ~ 86 N (X) も、ベクトルデータファイル 82 (0) ~ 82 (X) からシャドウタップ付き遅延線 78 (1) の中にロードされ得る。図 7 に関して下記でさらに詳細に説明されるように、次の入力ベクトルデータサンプルセット 86 N (0) ~ 86 N (X) は、シフトされた入力ベクトルデータサンプルセット 86 S (0) ~ 86 S (X) の少なくとも一部になるために、フィルタベクトル処理動作 102 の間にプライマリタップ付き遅延線 78 (0) の中にシフトされる。このように、プライマリタップ付き遅延線 78 (0) は、フィルタベクトル処理動作 102 のために実行されるべき次の入力ベクトルデータサンプルセット 86 N (0) ~ 86 N (X) が、ベクトルデータファイル 82 (0) ~ 82 (X) からプライマリタップ付き遅延線 78 (0) の中にフェッチされるまで、実行ユニット 84 (0) ~ 84 (X) が待つ必要があった場合、場合によっては被る遅延をフェッチすることなく、フィルタベクトル処理動作 102 の間に利用可能なシフトされた入力ベクトルデータサンプルセット 86 S (0) ~ 86 S (X) を有することができる。

20

【 0055 】

[00104] この関連で、図 7 は、図 4 の VPE 22 (1) 内に設けられ得る例示的なタップ付き遅延線 78 を示す。この実施形態では、タップ付き遅延線 78 は、シャドウタップ付き遅延線 78 (1) とプライマリタップ付き遅延線 78 (0) とを備える。この例におけるプライマリタップ付き遅延線 78 (0) は、入力ベクトルデータサンプル 86 の解像度が 8 ビット長に落ちることを可能にするために、複数の 8 ビットプライマリパイプラインレジスタ 120 から構成される。実行ユニット 84 (0) ~ 84 (X) によって処理される最初の入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) は、下記の図 9A に関して説明されるように、フィルタベクトル処理動作 102 の最初のフィルタタップのために、この例ではシフトされない。実行ユニット 84 (0) ~ 84 (X) がフィルタベクトル処理動作 102 のために次のフィルタタップを処理するとき、プライマリタップ付き遅延線 78 (0) に記憶された入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) の中の入力ベクトルデータサンプル 86 は、シフトされた入力ベクトルデータサンプルセット 86 S (0) ~ 86 S (X) になるために、図 7 の矢印によって示されたように、プライマリパイプラインレジスタ 120 (0) ~ 120 (4X + 3) 内でシフトされる。このようにして、実行ユニット 84 (0) ~ 84 (X) は、入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) を記憶およびシフトする必要なしに、ならびにベクトルデータファイル 82 (0) ~ 82 (X) からシフトされた入力ベクトルデータサンプルセット 86 S (0) ~ 86 S (X) を再フェッチすることなく、シフトされた入力ベクトルデータサンプルセット 86 S (0) ~ 86 S (X) を受信し、それらのフィルタベクトル処理動作 102 を実行することによって、十分利用される。

30

40

【 0056 】

[00105] この実施形態では、プライマリパイプラインレジスタ 120 (0) ~ 120 (

50

4X+3) はまとめて、図4のベクトルデータファイル82(0)~82(X)の幅である。15に等しい「X」を有する幅が512ビットであるベクトルデータファイル82(0)~82(X)の例では、512ビット(すなわち、64個のレジスタ×各8ビット)の合計幅を提供するために、各々が8ビット幅である64個の合計プライマリパイプラインレジスタ120(0)~120(63)が存在する。したがって、この例では、プライマリタップ付き遅延線78(0)は、1つの入力ベクトルデータサンプルセット86(0)~86(X)の幅全体を記憶することが可能である。この例では、8ビット幅のプライマリパイプラインレジスタ120(0)~120(4X+3)を設けることによって、入力ベクトルデータサンプルセット86(0)~86(X)は、プライマリパイプラインレジスタ120(0)~120(4X+3)において、8ビットフィルタベクトル処理動作のために8ビットのベクトルデータサンプルサイズにシフトダウンされ得る。たとえば、16ビットまたは32ビットのサンプルなどのより大きいサイズの入力ベクトルデータサンプル86のサイズがフィルタベクトル処理動作のために望ましい場合、入力ベクトルデータサンプルセット86(0)~86(X)は、プライマリパイプラインレジスタ120(0)~120(4X+3)において、1度に2つのプライマリパイプラインレジスタ120によってシフトされ得る。

【0057】

[00106]引き続き図7を参照すると、シャドウタップ付き遅延線78(1)もタップ付き遅延線78内に設けられる。シャドウタップ付き遅延線78(1)は、次のベクトル処理動作のためにベクトルデータファイル82(0)~82(X)から次の入力ベクトルデータサンプルセット86N(0)~86N(X)をラッチまたは輸送するために利用され得る。フィルタベクトル処理動作102のための各フィルタタップが実行ユニット84(0)~84(X)によって実行されるとき、次の入力ベクトルデータサンプルセット86N(0)~86N(X)からの次の入力ベクトルデータサンプル86Nは、シャドウタップ付き遅延線78(1)からプライマリタップ付き遅延線78(0)の中にシフトされる。シャドウタップ付き遅延線78(1)はまた、入力ベクトルデータサンプル86の解像度が、プライマリタップ付き遅延線78(0)と同様に8ビット長に落ちることを可能にするために、複数の8ビットシャドウパイプラインレジスタ122から構成される。プライマリパイプラインレジスタ120(0)~120(4X+3)のように、シャドウタップ付き遅延線78(1)内に設けられたシャドウパイプラインレジスタ122(0)~122(4X+3)はまとめて、この例では512ビットである、ベクトルデータファイル82(0)~82(X)の幅である。したがって、シャドウタップ付き遅延線78(1)のシャドウパイプラインレジスタ122(0)~122(4X+3)も、1つの入力ベクトルデータサンプルセット86(0)~86(X)の幅全体を記憶することが可能である。したがって、この実施形態では、プライマリタップ付き遅延線78(0)に含まれるシャドウパイプラインレジスタ122(0)~122(4X+3)の数は、この例では合計16である(すなわち、X=15)ベクトルデータレーン100(0)~100(X)の数の4倍である。したがって、シャドウパイプラインレジスタ122の数も、合計512ビット(すなわち、64個のレジスタ×各8ビット)向けにこの例では合計64である。プライマリタップ付き遅延線78(0)に関して上記で説明されたように、この例では、8ビット幅のシャドウパイプラインレジスタ122(0)~122(4X+3)を設けることによって、次の入力ベクトルデータサンプルセット86N(0)~86N(X)は、8ビットフィルタベクトル処理動作のために8ビットのベクトルデータサンプルサイズにシフトダウンされ得る。

【0058】

[00107]図8は、図7のプライマリタップ付き遅延線78(0)およびシャドウタップ付き遅延線78(1)の中に存在する選択されたプライマリパイプラインレジスタ120とシャドウパイプラインレジスタ122とを示す概略図である。図8は、プライマリパイプラインレジスタ120とシャドウパイプラインレジスタ122との間の入力ベクトルデータサンプル86のシフトの例を説明することを容易にするために提供される。上記で説

10

20

30

40

50

明されたように、入力ベクトルデータサンプル 86 はまた、プライマリタップ付き遅延線 78 (0) およびシャドウタップ付き遅延線 78 (1) 内で、ならびにシャドウタップ付き遅延線 78 (1) からプライマリタップ付き遅延線 78 (0) にシフトされ得る。パイプラインレジスタ 120、122 は、入力ベクトルデータサンプル 86 が必要な場合 8 ビットの解像度でシフトすることを可能にするために、この例では各々 8 ビット幅である。これは下記でより詳細に説明される。プライマリタップ付き遅延線 78 (0) およびシャドウタップ付き遅延線 78 (1) はまた、同様に下記でより詳細に説明されるように、入力ベクトルデータサンプル 86 の解像度の 16 ビットシフトと 32 ビットシフトとを実行することが可能である。

【0059】

[00108]この関連で、図 8 は、図 7 のプライマリタップ付き遅延線 78 (0) 内に入力ベクトルデータサンプル 86 $S(X)$ のための記憶レジスタを形成する、プライマリパイプラインレジスタ 120 ($4X+3$)、120 ($2X+1$)、120 ($4X+2$)、および 120 ($2X$) の中への入力ベクトルデータサンプル 86 のシフトを示す。プライマリパイプラインレジスタ 120 ($4X+3$) および 120 ($4X+2$) は、それぞれ、図 7 のプライマリタップ付き遅延線 78 (0) 内のレジスタ B_{31} および B_{30} である。プライマリパイプラインレジスタ 120 ($2X+1$) および 120 ($2X$) は、それぞれ、図 7 のプライマリタップ付き遅延線 78 (0) 内のレジスタ A_{31} および A_{30} である。図 7 に示されたように、レジスタ B_{31} および B_{30} のためのプライマリパイプラインレジスタ 120 ($4X+3$) および 120 ($4X+2$) は、シャドウタップ付き遅延線 78 (1) 内の隣接するシャドウパイプラインレジスタ 122 からシフトされた入力ベクトルデータサンプル 86 を受信するように構成される。したがって、図 8 の例では、それぞれ、レジスタ A'_0 および A'_1 のためのシャドウパイプラインレジスタ 122 (0) および 122 (1) は、 B_{31} および B_{30} のためのプライマリパイプラインレジスタ 120 ($4X+3$) および 120 ($4X+2$) の中に入力ベクトルデータサンプル 86 をシフトするように構成されるものとして示される。同様に、図 8 の例では、プライマリタップ付き遅延線 78 (0) 内の、それぞれ、レジスタ B_1 および B_0 のためのプライマリパイプラインレジスタ 120 ($2X+3$) および 120 ($2X+2$) は、レジスタ A_{31} および A_{30} のための隣接するプライマリパイプラインレジスタ 120 ($2X+1$) および 120 ($2X$) の中に入力ベクトルデータサンプル 86 をシフトするように構成されるものとして示される。これらのレジスタ間の入力ベクトルデータサンプル 86 の例示的なシフトが次に記載される。

【0060】

[00109]引き続き図 8 を参照すると、図 4 ならびに入力ベクトルデータサンプル 86 のシフトにおいて、ベクトルデータファイル 82 (0) ~ 82 (X) から新しい入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) をロードするように、プライマリパイプラインレジスタ 120 とシャドウパイプラインレジスタ 122 とを構成する柔軟性を提供するために、入力ベクトルデータサンプル選択器がプライマリパイプラインレジスタ 120 およびシャドウパイプラインレジスタ 122 の各々に関連付けられる。この関連で、プライマリタップ付き遅延線 78 (0) において、それぞれ、プライマリパイプラインレジスタ 120 (0) ~ 120 ($4X+3$) の中にロードまたはシフトされるベクトルデータに、入力ベクトルデータサンプル選択器 124 (0) ~ 124 ($4X+3$) が提供される。シャドウタップ付き遅延線 78 (1) において、それぞれ、シャドウパイプラインレジスタ 122 (0) ~ 122 ($4X+3$) の中にロードまたはシフトされるベクトルデータに、入力ベクトルデータサンプル選択器 126 (0) ~ 126 ($4X+3$) が提供される。入力ベクトルデータサンプル選択器 124 (0) ~ 124 ($4X+3$) および入力ベクトルデータサンプル選択器 126 (0) ~ 126 ($4X+3$) は、この例では各々マルチプレクサである。下記でより詳細に説明されるように、入力ベクトルデータサンプル選択器 124 (0) ~ 124 ($4X+3$)、126 (0) ~ 126 ($4X+3$) は、各々、プライマリパイプラインレジスタ 120 (0) ~ 120 ($4X+3$) およびシャドウパイプラインレジスタ 122 (0) ~ 122 ($4X+3$) の中にロードまたはシフトされるべき

10

20

30

40

50

入力ベクトルデータを選択するために、データ幅シフト制御入力 125 によって制御され得る。

【0061】

[00110]図8では、それぞれ、レジスタ B_{31} 、 B_{30} 、 A_{31} 、および A_{30} に対応する、それぞれ、プライマリパイプラインレジスタ 120 ($4X+3$)、120 ($4X+2$)、120 ($2X+1$)、120 ($2X$) のために、入力ベクトルデータサンプル選択器 124 ($4X+3$)、124 ($4X+2$)、124 ($2X+1$)、124 ($2X$) のみが示されていることに留意されたい。図8では、それぞれ、レジスタ A'_{11} 、 A'_{10} 、 B_1 、および B_0 に対応する、それぞれ、パイプラインレジスタ 122 (1)、122 (0)、120 ($2X+3$)、120 ($2X+2$) のために、入力ベクトルデータサンプル選択器 126 (1)、126 (0)、124 ($2X+3$)、124 ($2X+2$) のみが示されている。

【0062】

[00111]引き続き図8を参照すると、ベクトル処理動作のために、新しい入力ベクトルデータがプライマリタップ付き遅延線 78 (0) およびシャドウタップ付き遅延線 78 (1) の中にロードされるべき場合、データ幅シフト制御入力 125 は、入力ベクトルデータサンプル選択器 124 ($4X+3$)、124 ($4X+2$)、124 ($2X+1$)、124 ($2X$) に、ロードデータフローパス 133 ($4X+3$)、133 ($4X+2$)、133 ($2X+1$)、133 ($2X$) を選択させるように、図4の VPE 22 (1) によって構成され得る。ロードデータフローパス 133 ($4X+3$)、133 ($4X+2$)、133 ($2X+1$)、133 ($2X$) を選択すると、ベクトルデータファイル 82 (0) ~ 82 (X) からの入力ベクトルデータがプライマリパイプラインレジスタ 120 ($4X+3$)、120 ($4X+2$)、120 ($2X+1$)、120 ($2X$) に記憶されることが可能になる。ベクトルデータファイル 82 (0) ~ 82 (X) から入力ベクトルデータをロードすることは、例として VPE 22 (1) によって処理されるべき新しいまたは次のベクトル命令上で実行される場合がある。同様に、データ幅シフト制御入力 125 はまた、入力ベクトルデータサンプル選択器 126 (1)、124 ($2X+3$)、126 (0)、124 ($2X+2$) に、入力データフローパス 135 (1)、133 ($2X+3$)、135 (0)、133 ($2X+2$) を選択させるように、図4の VPE 22 (1) によって構成され得る。ロードデータフローパス 135 (1)、133 ($2X+3$)、135 (0)、133 ($2X+2$) を選択すると、ベクトルデータファイル 82 (0) ~ 82 (X) からの入力ベクトルデータがパイプラインレジスタ 122 (1)、120 ($2X+3$)、124 (0)、120 ($2X+2$) に記憶されることが可能になる。

【0063】

[00112]引き続き図8を参照すると、ベクトル処理動作のために、プライマリタップ付き遅延線 78 (0) およびシャドウタップ付き遅延線 78 (1) に記憶されたベクトルデータがシフトされる必要がある場合、データ幅シフト制御入力 125 は、入力ベクトルデータサンプル選択器 124 ($4X+3$)、124 ($4X+2$)、124 ($2X+1$)、124 ($2X$) に、ベクトルデータサンプルのシフトのための入力データフローパス 137 ($4X+3$)、137 ($4X+2$)、137 ($2X+1$)、137 ($2X$) を選択させるように、図4の VPE 22 (1) によって構成され得る。データ幅シフト制御入力 125 はまた、入力ベクトルデータサンプル選択器 126 (1)、124 ($2X+3$)、126 (0)、124 ($2X+2$) に、ベクトルデータサンプルのシフトのための入力データフローパス 139 (1)、137 ($2X+3$)、139 (0)、137 ($2X+2$) を選択させる。そこに示されているように、入力ベクトルデータサンプル選択器 124 ($4X+3$)、124 ($4X+2$)、124 ($2X+1$)、124 ($2X$) および入力ベクトルデータサンプル選択器 126 (1)、124 ($2X+3$)、126 (0)、124 ($2X+2$) は、各々、ベクトルデータが他のレジスタにシフトされることを可能にする、それぞれ、出力データフローパス 141 ($4X+3$)、141 ($4X+2$)、141 ($2X+1$)、141 ($2X$) および 143 (1)、141 ($2X+3$)、143 (0)、124 ($2X+2$) を含む。図8に示された出力データフローパスは、次に全体が示される出力デ

ータフローパス $141(0) \sim 141(4X+3)$ および $143(0) \sim 143(4X+3)$ の一部であるが、それぞれ、プライマリタップ付き遅延線 $78(0)$ 内の入力ベクトルデータサンプル選択器 $124(0) \sim 124(4X+3)$ およびシャドウタップ付き遅延線 $78(1)$ 内の入力ベクトルデータサンプル選択器 $126(0) \sim 126(4X+3)$ のために含まれる。

【0064】

[00113]例として、8ビットベクトルデータのシフト中、入力ベクトルデータサンプル選択器 $124(4X+3)$ 、 $124(4X+2)$ 、 $124(2X+1)$ 、 $124(2X)$ および入力ベクトルデータサンプル選択器 $126(1)$ 、 $124(2X+3)$ 、 $126(0)$ 、 $124(2X+2)$ は、それぞれ、入力データフローパス $137(4X+3)$ 、 $137(4X+2)$ 、 $137(2X+1)$ 、 $137(2X)$ 、 $139(1)$ 、 $137(2X+3)$ 、 $139(0)$ 、 $137(2X+2)$ を選択するように構成される。この関連で、例として、プライマリパイプラインレジスタ $120(2X+1)$ (すなわち、 A_{31}) 内のベクトルデータは、図8に示されたように、プライマリパイプラインレジスタ $120(2X)$ (すなわち、 A_{30}) に出力データフローパス $141(2X+1)$ 上でシフトされる。プライマリパイプラインレジスタ $120(4X+3)$ (すなわち、 B_{31}) 内のベクトルデータは、図8に示されたように、プライマリパイプラインレジスタ $120(4X+2)$ (すなわち、 B_{30}) に出力データフローパス $141(4X+3)$ 上でシフトされる。シャドウパイプラインレジスタ $122(0)$ (すなわち、 A'_0) 内のベクトルデータは、図8に示されたように、プライマリパイプラインレジスタ $120(4X+3)$ (すなわち、 B_{31}) に出力データフローパス $143(0)$ 上でシフトされる。プライマリパイプラインレジスタ $120(2X+3)$ (すなわち、 B_1) 内のベクトルデータは、図8に示されたように、プライマリパイプラインレジスタ $120(4X+2)$ (すなわち、 B_{30}) に出力データフローパス $141(2X+3)$ 上でシフトされる。シャドウパイプラインレジスタ $122(1)$ (すなわち、 A'_1) 内のベクトルデータは、図8に示されたように、シャドウパイプラインレジスタ $122(0)$ (すなわち、 A'_0) に出力データフローパス $143(1)$ 上でシフトされる。プライマリパイプラインレジスタ $120(2X+2)$ (すなわち、 B_0) 内のベクトルデータは、図8に示されたように、プライマリパイプラインレジスタ $120(2X+1)$ (すなわち、 A_{31}) に出力データフローパス $141(2X+2)$ 上でシフトされる。

【0065】

[00114]引き続き図8を参照すると、16ビットベクトルデータのシフト中、入力ベクトルデータサンプル選択器 $124(4X+3)$ 、 $124(4X+2)$ 、 $124(2X+1)$ 、 $124(2X)$ および入力ベクトルデータサンプル選択器 $126(1)$ 、 $124(2X+3)$ 、 $126(0)$ 、 $124(2X+2)$ は、それぞれ、入力データフローパス $145(4X+3)$ 、 $145(4X+2)$ 、 $145(2X+1)$ 、 $145(2X)$ 、 $147(1)$ 、 $145(2X+3)$ 、 $147(0)$ 、 $145(2X+2)$ を選択するように構成される。この関連で、例として、プライマリパイプラインレジスタ $120(2X+2)$ (すなわち、 B_0) 内のベクトルデータは、図8に示されたように、プライマリパイプラインレジスタ $120(2X)$ (すなわち、 A_{30}) に出力データフローパス $141(2X+2)$ 上でシフトされる。シャドウパイプラインレジスタ $122(0)$ (すなわち、 A'_0) 内のベクトルデータは、図8に示されたように、プライマリパイプラインレジスタ $120(4X+2)$ (すなわち、 B_{30}) に出力データフローパス $143(0)$ 上でシフトされる。プライマリパイプラインレジスタ $120(2X+3)$ (すなわち、 B_1) 内のベクトルデータは、図8に示されたように、プライマリパイプラインレジスタ $120(2X+1)$ (すなわち、 A_{31}) に出力データフローパス $141(2X+3)$ 上でシフトされる。シャドウパイプラインレジスタ $122(1)$ (すなわち、 A'_1) 内のベクトルデータは、図8に示されたように、プライマリパイプラインレジスタ $120(4X+3)$ (すなわち、 B_{31}) に出力データフローパス $143(1)$ 上でシフトされる。

【0066】

[00115] プライマリタップ付き遅延線 78 (0) およびシャドウタップ付き遅延線 78 (1) において 32 ビットベクトルデータのシフトが望ましい場合、プライマリパイプラインレジスタ 120 (0) ~ 120 (4X + 3) およびシャドウパイプラインレジスタ 122 (0) ~ 122 (4X + 3) に記憶されたベクトルデータは、必要な場合、2つの 16 ビットベクトルデータのシフト動作においてシフトされ得る。

【0067】

[00116] 図 7 では、レジスタ B₃₁ および B₃₀ のためのプライマリパイプラインレジスタ 120 (4X + 3) および 120 (4X + 2)、ならびにレジスタ A₃₁ および A₃₀ のためのプライマリパイプラインレジスタ 120 (2X + 1) および 120 (2X) は、シフトされた入力ベクトルデータサンプル 86 S (X) に対して互いに論理的に関連付けられるが、図 8 に示されたように、互いに物理的に隣接していないことに留意されたい。この配置は、図 6 B に示されたように、ベクトルデータファイル 82 (0) ~ 82 (X) 内の入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) の記憶パターンに起因して、この例において提供される。同様に図 6 B に示されたように、ベクトルデータファイル 82 (0) ~ 82 (X) に記憶された入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) は、ADDRESS0 および ADDRESS1 をまたぐ。しかしながら、本明細書内の開示は、ベクトルデータファイル 82 (0) ~ 82 (X) 内の入力ベクトルサンプルセット 86 (0) ~ 86 (X) のこの記憶パターンに限定されないことに留意されたい。

【0068】

[00117] さらに、図 8 に関して、タップ付き遅延線 78 (0)、78 (1) は、実行されるべきベクトル命令に従って、タップ付き遅延線 78 (0)、78 (1) のためのプログラム可能な入力データパス構成に基づいて、ベクトルデータファイル 82 (0) ~ 82 (X) と実行ユニット 84 (0) ~ 84 (X) との間に入力データフローパス 80 (0) ~ 80 (X) 内に、選択的に設けられるか、または設けられないように構成可能である。たとえば、ベクトル命令がフィルタベクトル処理命令ではなく、および/または場合によっては入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) をシフトするためにタップ付き遅延線 78 (0)、78 (1) を必要としない場合、タップ付き遅延線 78 (0)、78 (1) は、入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) をラッチしないように構成され得る。入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) は、プライマリタップ付き遅延線 78 (0) とシャドウタップ付き遅延線 78 (1) とをバイパスすることによって、それぞれの実行ユニット 84 (0) ~ 84 (X) にベクトルデータファイル 82 (0) ~ 82 (X) から供給され得る。このプログラム可能なデータパス構成により、さらに、プライマリタップ付き遅延線 78 (0) およびシャドウタップ付き遅延線 78 (1) が入力データフローパス 80 (0) ~ 80 (X) 内に設けられるか、または設けられないことが可能になる。プライマリタップ付き遅延線 78 (0) およびシャドウタップ付き遅延線 78 (1) は、必要に応じて、ベクトル命令ごとに、入力データフローパス 80 (0) ~ 80 (X) 内に設けられるか、または設けられないようにプログラムされ得る。

【0069】

[00118] 図 9 A は、フィルタベクトル処理命令の第 1 のクロックサイクル (CYCLE 0) の間にプライマリタップ付き遅延線 78 (0) の中にベクトルデータファイル 82 (0) ~ 82 (X) からロードされた入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) を示す。プライマリタップ付き遅延線 78 (0) およびシャドウタップ付き遅延線 78 (1) は、図 7 から簡略化された形式で示されている。グローバルレジスタファイル 40 も示されている。最初の入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) が入力ベクトルデータサンプル X0 ~ X63 としてプライマリタップ付き遅延線 78 (0) の中にロードされる。たとえば、プライマリタップ付き遅延線 78 (0) の中に (および下記でより詳細に説明されるように、シャドウタップ付き遅延線 78 (1) の中にも) 最初の入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) をロードするために、特殊ベクトル命令がサポートされる場合がある。この最初の入力ベクトルデータサンプル

セット 86 (0) ~ 86 (X) は、図 6 B に示されたように、ベクトルデータファイル 82 (0) ~ 82 (X) 内の ADDRESS 0 および ADDRESS 1 に記憶された。ひとえにこの例のための図 4 の VPE 22 (1) 内のベクトルデータファイル 82 (0) ~ 82 (X) の記憶パターンのせいで、この例では、X 0、X 1、X 32、および X 33 が最初の入力ベクトルデータサンプル 86 (0) を形成することに留意されたい。他の入力ベクトルデータサンプル 86 は、同様に、図 9 A に示されたように形成される（たとえば、86 (1)、86 (2)、... 86 (X)）。入力ベクトルデータサンプル 86 を一緒にグループ化して、入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) を形成するために、他のパターンが提供される可能性がある。

【 0 0 7 0 】

[00119] 図 9 B は、フィルタベクトル処理命令の第 2 のクロックサイクル (CYCLE 1) の間にシャドウタップ付き遅延線 78 (1) の中にロードされた次の入力ベクトルデータサンプルセット 86 N (0) ~ 86 N (X) を示す。フィルタ処理動作の実行をセットアップするために、ベクトルデータファイル 82 (0) ~ 82 (X) からの最初の入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) がプライマリタップ付き遅延線 78 (0) の中にロードされた後に、次の入力ベクトルデータサンプルセット 86 N (0) ~ 86 N (X) がシャドウタップ付き遅延線 78 (1) の中にロードされる。この次の入力ベクトルデータサンプルセット 86 N (0) ~ 86 N (X) は、入力ベクトルデータサンプル X 64 ~ X 127 としてシャドウタップ付き遅延線 78 (1) の中にロードされる。この次の入力ベクトルデータサンプルセット 86 N (0) ~ 86 N (X) は、図 6 B に示されたように、ベクトルデータファイル 82 (0) ~ 82 (X) 内の ADDRESS 2 および ADDRESS 3 に記憶された。ひとえにこの例のための図 4 の VPE 22 (1) 内のベクトルデータファイル 82 (0) ~ 82 (X) の記憶パターンのせいで、この例では、X 64、X 65、X 96、および X 97 が最初の入力ベクトルデータサンプル 86 (0) を形成することに留意されたい。入力ベクトルデータサンプル 86 を一緒にグループ化して、入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) を形成するために、他のパターンが提供される可能性がある。グローバルレジスタファイル 40 からの最初のフィルタ係数 92 (0) も、フィルタベクトル処理動作 102 において使用するために図 9 B の実行ユニット 84 (0) ~ 84 (X) へのレジスタ (「 C 」) 内に設けられるものとして示される。

【 0 0 7 1 】

[00120] 図 7 に戻って参照すると、フィルタベクトル処理動作 102 の各処理ステージの間に入力ベクトルデータサンプル 86 がプライマリタップ付き遅延線 78 (0) 内でシフトされるとき、シャドウパイプラインレジスタ 122 に記憶された次の入力ベクトルデータサンプル 86 N も、シャドウタップ付き遅延線 78 (1) のシャドウパイプラインレジスタ 122 内でシフトされる。図 7 の最初のシャドウパイプラインレジスタ 122 (0) に記憶された入力ベクトルデータサンプル 86 は、各シフトの間にプライマリタップ付き遅延線 78 (0) の最後のプライマリパイプラインレジスタ 120 (4 X + 3) の中にシフトされる。したがって、このようにして、フィルタベクトル処理動作 102 の処理ステージが実行ユニット 84 (0) ~ 84 (X) において進行するとき、シャドウタップ付き遅延線 78 (1) に最初に記憶された次の入力ベクトルデータサンプルセット 86 N (0) ~ 86 N (X) の少なくとも一部分は、処理のために実行ユニット 84 (0) ~ 84 (X) に供給されるために、プライマリタップ付き遅延線 78 (0) の中にシフトされる。シフトの回数は、この例ではフィルタベクトル処理動作 102 において提供されたフィルタタップの数に依存する。ベクトルデータファイル 82 (0) ~ 82 (X) からプライマリタップ付き遅延線 78 (0) およびシャドウタップ付き遅延線 78 (1) の中にフェッチされた入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) 内の入力ベクトルデータサンプル 86 の数が、フィルタベクトル処理動作 102 におけるフィルタタップの数よりも大きい場合、実行ユニット 84 (0) ~ 84 (X) は、任意のさらなる入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) がベクトルデータファイル 82 (0)

10

20

30

40

50

～ 8 2 (X) から再フェッチされることなく、フィルタベクトル処理動作 1 0 2 を実行することができる。しかしながら、フィルタベクトル処理動作 1 0 2 におけるフィルタタップの数が、ベクトルデータファイル 8 2 (0) ～ 8 2 (X) からプライマリタップ付き遅延線 7 8 (0) およびシャドウタップ付き遅延線 7 8 (1) の中にフェッチされた入力ベクトルデータサンプルセット 8 6 (0) ～ 8 6 (X) 内の入力ベクトルデータサンプル 8 6 よりも大きい場合、フィルタベクトル処理動作 1 0 2 の一部として、さらなる入力ベクトルデータサンプルセット 8 6 (0) ～ 8 6 (X) がベクトルデータファイル 8 2 (0) ～ 8 2 (X) からフェッチされ得る。フィルタベクトル処理動作 1 0 2 がシフトされた入力ベクトルデータサンプルセット 8 6 S (0) ～ 8 6 S (X) に対して完了した後、タップ付き遅延線 7 8 (0)、7 8 (1) 内に未処理の入力ベクトルデータサンプル 8 6 S が存在する場合、実行ユニット 8 4 (0) ～ 8 4 (X) は、次いで、次のフィルタベクトル処理動作のためのシフトされた入力ベクトルデータサンプルセット 8 6 S (0) ～ 8 6 S (X) として、プライマリタップ付き遅延線 7 8 (0) に記憶された前の次の入力ベクトルデータサンプルセット 8 6 N (0) ～ 8 6 N (X) を供給され得る。

【 0 0 7 2 】

[00121] シャドウタップ付き遅延線 7 8 (1) を提供するための別の例示的な論理的根拠は以下の通りである。現在のフィルタベクトル処理動作 1 0 2 が、ベクトルデータレーン 1 0 0 (0) ～ 1 0 0 (X) の幅で提供され得るよりも多くの入力ベクトルデータサンプル 8 6 を要する場合、シャドウタップ付き遅延線 7 8 (1) の中にロードされたさらなる入力ベクトルデータサンプルセット 8 6 (0) ～ 8 6 (X) は、遅延がないフィルタベクトル処理動作 1 0 2 の間、実行ユニット 8 4 (0) ～ 8 4 (X) に利用可能である。フィルタベクトル処理動作 1 0 2 が、実行中シフトされた入力ベクトルデータサンプルセット 8 6 S (0) ～ 8 6 S (X) を通じて進行するとき、上記で説明されたように、シャドウタップ付き遅延線 7 8 (1) の中にロードされたさらなる次の入力ベクトルデータサンプルセット 8 6 N (0) ～ 8 6 N (X) が、プライマリタップ付き遅延線 7 8 (0) の中にシフトされる。したがって、このようにして、実行ユニット 8 4 (0) ～ 8 4 (X) によるベクトル処理において使用するための次の入力ベクトルデータサンプルセット 8 6 N (0) ～ 8 6 N (X) は、遅延なく利用可能である。ベクトルデータファイル 8 2 (0) ～ 8 2 (X) の幅の単一のフェッチされた入力ベクトルデータサンプルセット 8 6 (0) ～ 8 6 (X) が、フィルタベクトル処理動作 1 0 2 全体を実行するのに十分であるかどうかにかかわらず、実行ユニット 8 4 (0) ～ 8 4 (X) は、フィルタベクトル処理動作 1 0 2 の間、十分に利用され続けることができる。

【 0 0 7 3 】

[00122] 最初の入力ベクトルデータサンプルセット 8 6 N (0) ～ 8 6 N (X) および次の入力ベクトルデータサンプルセット 8 6 N (0) ～ 8 6 N (X) が、それぞれ、プライマリタップ付き遅延線 7 8 (0) およびシャドウタップ付き遅延線 7 8 (1) の中にロードされた後、プライマリタップ付き遅延線 7 8 (0) に供給された最初の入力ベクトルデータサンプルセット 8 6 (0) ～ 8 6 (X) は、フィルタベクトル処理動作 1 0 2 の最初の処理ステージにおいて処理されるために、それぞれの実行ユニット 8 4 (0) ～ 8 4 (X) に供給される (図 5 のブロック 1 0 8)。最初の入力ベクトルデータサンプルセット 8 6 (0) ～ 8 6 (X) が実行ユニット 8 4 (0) ～ 8 4 (X) によって処理された後、最初の入力ベクトルデータサンプルセット 8 6 (0) ～ 8 6 (X) は、実行ユニット 8 4 (0) ～ 8 4 (X) によって処理されるべきシフトされた入力ベクトルデータサンプルセット 8 6 S (0) ～ 8 6 S (X) になるために、プライマリタップ付き遅延線 7 8 (0) 内でシフトされる。図 4 の V P E 2 2 (1) において示されたように、シフトされた入力ベクトルデータサンプル 8 6 S (0) は実行ユニット 8 4 (0) に供給され、シフトされた入力ベクトルデータサンプル 8 6 S (1) は実行ユニット 8 4 (1) に供給され、以下同様である。

【 0 0 7 4 】

[00123] 次に、実行ユニット 8 4 (0) ～ 8 4 (X) は、フィルタベクトル処理動作 1

02を実行する(図5のブロック110)。より詳細には、実行ユニット84(0)~84(X)は、この例では演算： $y[n] = x[n-7] * h_7$ に従って、第1の繰返しにおいて最初の入力ベクトルデータサンプルセット86(0)~86(X)を現在のフィルタ係数92(0)と乗算し、ここで、 $x[n-7]$ は、結果として生じるフィルタ出力ベクトルデータサンプルセット94(0)~94(X)を供給する最初の入力ベクトルデータサンプルセット86(0)~86(X)である。フィルタベクトル処理動作102の次の繰返し(図5のブロック110)において、フィルタベクトル処理動作102のための次のシフトされた入力ベクトルデータサンプルセット86S(0)~86S(X)は、現在のフィルタ係数92(1)~92(Y-1)と乗算される。実行ユニット84(0)~84(X)は、新しい前の、結果として生じるフィルタ出力ベクトルデータサンプルセット94(0)~94(X)を供給するために、結果として生じるフィルタ出力ベクトルデータサンプルセット94(0)~94(X)を、実行ユニット84(0)~84(X)によって計算された前の、結果として生じるフィルタ出力ベクトルデータサンプルセット94(0)~94(X)と累算する(図5のブロック112)。フィルタベクトル処理動作102の最初の処理ステージでは、前の、結果として生じるフィルタ出力ベクトルデータサンプルセットは存在しない。

【0075】

[00124]フィルタベクトル処理動作102のすべての処理ステージが完了した場合(図5のブロック114)、ベクトルデータファイル82(0)~82(X)に供給され記憶されるために、出力データフローパス98(0)~98(X)内の、結果として生じるフィルタ出力ベクトルデータサンプルセット94(0)~94(X)として、累算された前の、結果として生じるフィルタ出力ベクトルデータサンプルセット94(0)~94(X)が供給される(図5のブロック116)。フィルタベクトル処理動作102のすべての処理ステージが完了していない場合(図5のブロック114)、フィルタベクトル処理動作102に次のシフトされた入力ベクトルデータサンプルセット86S(0)~86S(X)を供給するために、タップ付き遅延線78(0)および78(1)に記憶されたサンプルがタップ付き遅延線78(0)、78(1)内でシフトされる(図5のブロック118)。シフトされた入力ベクトルデータサンプルセット86S(0)~86S(X)は、フィルタベクトル処理動作102が完了するまで、前の、結果として生じるフィルタ出力ベクトルデータサンプルセットと累算されるために、中間結果として次の、結果として生じるフィルタ出力ベクトルデータサンプルセットを計算するために供給される。タップ付き遅延線78(0)、78(1)内にシフトされた入力ベクトルデータサンプルセット86S(0)~86S(X)を供給するために入力ベクトルデータサンプル86をシフトすることは、図7に関して詳細に上記で前述された。フィルタベクトル処理動作102に実行ユニット84(0)~84(X)によって供給された中間結果の最終的な累算は、図4に示されたように、実行ユニット84(0)~84(X)から、結果として生じるフィルタ出力ベクトルデータサンプルセット94(0)~94(X)として供給される。

【0076】

[00125]図9Cは、次のフィルタ処理動作 $y[n] = x[n-6] * h_6$ のための次のシフトされた入力ベクトルデータサンプルセット86S(0)~86S(X)になるために、フィルタベクトル処理動作102の2番目の処理ステージにおいて、入力ベクトルデータサンプルセット86(0)~86(X)がシフトされたときのタップ付き遅延線78のコンテンツを示す。プライマリタップ付き遅延線78(0)内のシフトされた入力ベクトルデータサンプルセット86S(0)~86S(X)は、実行されているベクトル命令によって規定された入力ベクトルデータサンプルのシフト幅に従って、プライマリパイプラインレジスタ120(0)~120(4X+3)内でシフトされる。たとえば、図9Cに示されたように、サンプルX2はシフトされた入力ベクトルデータサンプル86S(0)内でシフトされる。新しいシフトされた入力ベクトルデータサンプルセット86S(0)~86S(X)は、フィルタベクトル処理動作102の次のフィルタタップのための実行のために、実行ユニット84(0)~84(X)に供給される。実行ユニット84(0

10

20

30

40

50

）～ 8 4（X）に供給されたフィルタ係数 9 2 は、この例では「h 6」である次のフィルタ係数 9 2 でもある。

【 0 0 7 7 】

[00126]引き続き図 5 を参照すると、次のフィルタ係数 9 2 と乗算される（図 5 のブロック 1 1 0）ために、実行ユニット 8 4（0）～ 8 4（X）にプライマリタップ付き遅延線 7 8（0）からシフトされた入力ベクトルデータサンプルセット 8 6 S（0）～ 8 6 S（X）を供給する（図 5 のブロック 1 0 8）ことによって、プロセスは繰り返す。結果として生じるフィルタ出力ベクトルデータサンプルセット 9 4（0）～ 9 4（X）が前の、結果として生じるフィルタ出力ベクトルデータサンプルセット 9 4（0）～ 9 4（X）と累算される（図 5 のブロック 1 1 2）。図 9 D は、例示的なフィルタベクトル処理動作 1 0 2 の最後の処理ステージの間、タップ付き遅延線 7 8（0）、7 8（1）内に存在する入力ベクトルデータサンプル 8 6 の状態を示す。図 9 D に示されたこの例では、フィルタ係数 9 2「h 7」～「h 0」（すなわち、9 2（0）～ 9 2（Y - 1））のせいで、フィルタベクトル処理動作 1 0 2 において 8 個のフィルタタップ（Y）が存在した。図 9 D に示されたように、「h 0」はフィルタベクトル処理動作 1 0 2 における最後のフィルタ係数 9 2 である。シフトされた入力ベクトルデータサンプルセット 8 6 S（0）～ 8 6 S（X）は（フィルタタップの数よりも 1 回少ない）7 回シフトされており、その結果、フィルタベクトル処理動作 1 0 2 のための最後の 8 番目の処理ステージにおいて、入力ベクトルデータサンプル X 3 9 がプライマリタップ付き遅延線 7 8（0）内のシフトされた入力ベクトルデータサンプル 8 6 S（0）に記憶される。

【 0 0 7 8 】

[00127]上述されたフィルタベクトル処理動作 1 0 2 の例は、フィルタベクトル処理動作 1 0 2 を提供するために V P E 2 2（1）内のベクトルデータレーン 1 0 0（0）～ 1 0 0（X）の各々を利用するが、それは必要でないことに留意されたい。フィルタベクトル処理動作 1 0 2 は、フィルタベクトル処理動作 1 0 2 に利用されるべきベクトルデータレーン 1 0 0（0）～ 1 0 0（X）のサブセットを必要とするにすぎない場合がある。たとえば、入力ベクトルデータサンプルセット 8 6（0）～ 8 6（X）の幅がすべてのベクトルデータファイル 8 2（0）～ 8 2（X）の幅よりも小さい場合があり、ここで、フィルタベクトル処理動作 1 0 2 と並列に実行されるべき他のベクトル処理動作にさらなるベクトルデータレーン 1 0 0 を利用することが望ましい。このシナリオでは、図 7 のタップ付き遅延線 7 8（0）、7 8（1）は、最後のベクトルデータレーン 1 0 0（X）に到達するより前に、ベクトルデータレーン 1 0 0 内のシフトされた入力ベクトルデータサンプルセット 8 6 S（0）～ 8 6 S（X）として、シャドウタップ付き遅延線 7 8（1）からプライマリタップ付き遅延線 7 8（0）に次の入力ベクトルデータサンプルセット 8 6 N（0）～ 8 6 N（X）をシフトするように、修正される必要があり得る。

【 0 0 7 9 】

[00128]図 1 0 は、上記の例における例示的な 8 個のタップフィルタベクトル処理ステージが、 $y[n] = x[n] * h_0 + x[n-1] * h_1 + \dots + x[n-7] * h_7$ に従って完全に実行された後の、図 4 の V P E 2 2（1）内の実行ユニット 8 4（0）～ 8 4（X）内の累算器のコンテンツ（すなわち、結果として生じるフィルタ出力ベクトルデータサンプル 9 4）の概略図である。この例では、各実行ユニット 8 4（0）～ 8 4（X）は、ベクトルデータレーン 1 0 0（0）～ 1 0 0（X）ごとに並列に配置された 4 つの累算器を有するので、累算器 A c c 0 ～ A c c 3 が図 1 0 に示されている。累算された、結果として生じる出力ベクトルデータサンプルは、さらなる分析および/または処理のためにそこに記憶されるべき全体の、結果として生じるフィルタ出力ベクトルデータサンプルセット 9 4（0）～ 9 4（X）として、ベクトルデータファイル 8 2（0）～ 8 2（X）に出力データフローパス 9 8（0）～ 9 8（X）上で供給され得る。必要な場合、ベクトルデータファイル 8 2（0）～ 8 2（X）から図 2 のベクトルユニットデータメモリ 3 2 に、結果として生じるフィルタ出力ベクトルデータサンプルセット 9 4（0）～ 9 4（X）の行を移動するために、特殊なベクトル命令が V P E 2 2（1）によってサポート

される場合がある。

【 0 0 8 0 】

[00129]フィルタベクトル処理動作 1 0 2 以外の他のタイプのベクトル処理動作も、上記で説明された図 4 の V P E 2 2 (1) 内に設けられたタップ付き遅延線 7 8 と同じまたは同様のタップ付き遅延線 7 8 の使用による、V P E における処理効率を享受することができる。たとえば、V P E における入力ベクトルデータサンプルセット 8 6 のシフトを伴う別の特殊なベクトル処理動作は、(本明細書では「 相関ベクトル処理動作 」と呼ばれる) 相関 / 共分散ベクトル処理動作である。例として、C D M A システムにおいてユーザ信号と他のユーザの信号との間の良好な分離を提供するために、C D M A システムにおいてユーザ信号を復調するための直接スペクトル拡散コード (D S S C) (すなわち、チップシーケンス) を選ぶために相関演算を提供するようにベクトル処理を利用することが望ましい場合がある。信号の分離は、受信された信号を所望のユーザのローカルに生成されたチップシーケンスと相関させることによって行われる。信号が所望のユーザのチップシーケンスと一致する場合、相関関数は高くなり、C D M A システムはその信号を抽出することができる。所望のユーザのチップシーケンスが信号と共通する部分を少ししか、またはまったく有していない場合、相関は可能な限りゼロに近い (したがって信号を除去する) はずであり、これは相互相関と呼ばれる。チップシーケンスがゼロ以外の任意の時間オフセットで信号と相関される場合、相関は可能な限りゼロに近いはずである。これは自己相関と呼ばれ、マルチパス干渉を拒絶するために使用される。

【 0 0 8 1 】

[00130]しかしながら、相関演算は、ベクトルプロセッサにおいて提供される特殊なデータフローパスに起因して、ベクトルプロセッサにおいて並列化することは困難であり得る。相関されるべき信号を表す入力ベクトルデータサンプルセットが遅延タップ間でシフトされると、入力ベクトルデータサンプルセットはベクトルデータファイルから再フェッチされ、したがって電力消費が増大し、スループットが低減される。メモリからの入力ベクトルデータサンプルセットの再フェッチを最小化するために、データフローパスは、効率的な並列化処理のために、遅延タップと同じ数の乗算器を設けるように構成される可能性がある。しかしながら、他のベクトル処理動作は、より少ない乗算器しか必要としない場合があり、それにより、データフローパス内の乗算器の非効率的なスケーリングおよび過剰利用がもたらされる。スケラビリティを提供するために、乗算器の数が遅延タップの数よりも少なくなるように削減された場合、相関処理の様々なフェーズに対して同じ入力ベクトルデータサンプルセットを取得するために、メモリにより多くの再フェッチが必要とされることによって、並列化が制限される。

【 0 0 8 2 】

[00131]この関連で、図 1 1 は、図 2 の V P E 2 2 として提供され得る別の例示的な V P E 2 2 (2) の概略図である。下記でより詳細に記載されるように、図 1 1 の V P E 2 2 (2) は、ベクトルデータサンプルの再フェッチが除去または低減され、電力消費が低減される、V P E 2 2 (2) 内の精度相関ベクトル処理動作を提供するように構成される。精度相関ベクトル処理動作は、ベクトルデータサンプルの再フェッチを必要とし、それにより結果として電力消費が増大する、中間結果の記憶を必要とする相関ベクトル処理動作と比較して、V P E 2 2 (2) において提供され得る。ベクトルデータファイルからの入力ベクトルデータサンプルの再フェッチを除去または最小化して、電力消費を低減し、処理効率を改善するために、図 4 の V P E 2 2 (1) に含まれるタップ付き遅延線 7 8 も、V P E 2 2 (2) 内のベクトルデータファイル 8 2 (0) ~ 8 2 (X) と (「 E U 」 とも標記される) 実行ユニット 8 4 (0) ~ 8 4 (X) との間の入力データフローパス 8 0 (0) ~ 8 0 (X) に含まれる。「 X 」 + 1 は、この例におけるベクトルデータサンプルの処理用に V P E 2 2 (2) 内に設けられる並列入力データレーンの最大数である。上記で前に説明されたように、タップ付き遅延線 7 8 は、ベクトルデータファイル 8 2 (0) ~ 8 2 (X) の対応するサブセットまたはすべてから入力ベクトルデータサンプルセット 8 6 (0) ~ 8 6 (X) の入力ベクトルデータサンプル 8 6 のサブセットまたはすべてと

して、タップ付き遅延線入力 88(0) ~ 88(X) 上で入力ベクトルデータサンプルセット 86(0) ~ 86(X) を受信するように構成される。すべての入力ベクトルデータサンプル 86 は、入力ベクトルデータサンプルセット 86(0) ~ 86(X) を備える。下記でより詳細に説明されるように、ベクトルデータファイル 82(0) ~ 82(X) からの入力ベクトルデータサンプルセット 86(0) ~ 86(X) は、結果として生じる相関出力ベクトルデータサンプルセット 132(0) ~ 132(X) を供給するために、基準ベクトルデータサンプルセット 130(0) ~ 130(X) と VPE22(2) において相関される。基準ベクトルデータサンプルセット 130(0) ~ 130(X) は、この例では 130(0)、130(1)、...、および 130(X) である、「X + 1」個の基準ベクトルデータサンプル 130 から構成される。結果として生じる相関出力ベクトルデータサンプルセット 132(0) ~ 132(X) は、この例では 132(0)、132(1)、...、および 132(X) である、「X + 1」個の、結果として生じる相関出力ベクトルデータサンプル 132 から構成される。

10

【0083】

[00132]引き続き図 11 を参照すると、タップ付き遅延線 78 は、シフトされた入力ベクトルデータサンプルセット 86S(0) ~ 86S(X) を供給するために、VPE22(2) によって実行されるべき相関ベクトル命令に従う相関ベクトル処理動作の相関遅延タップ（すなわち、相関処理ステージ）ごとに、入力ベクトルデータサンプルセット 86(0) ~ 86(X) をシフトする。シフトされた入力ベクトルデータサンプル 86S のすべては、シフトされた入力ベクトルデータサンプルセット 86S(0) ~ 86S(X) を備える。タップ付き遅延線 78 は、相関ベクトル処理動作中、実行ユニット 84(0) ~ 84(X) の実行ユニット入力 90(0) ~ 90(X) にシフトされた入力ベクトルデータサンプルセット 86S(0) ~ 86S(X) を供給するために、入力ベクトルデータサンプルセット 86(0) ~ 86(X) をシフトする。このようにして、シフトされた入力ベクトルデータサンプルセット 86S(0) ~ 86S(X) に対して実行される動作に基づく中間相関結果は、VPE22(2) によって実行される相関ベクトル処理動作の各処理ステージの間に、記憶、シフト、およびベクトルデータファイル 82(0) ~ 82(X) から再フェッチされる必要がない。このように、タップ付き遅延線 78 は、電力消費を低減し、VPE22(2) によって実行される相関ベクトル処理動作についての処理効率を上げることができる。

20

30

【0084】

[00133]引き続き図 11 を参照すると、実行ユニット 84(0) ~ 84(X) は、相関ベクトル処理動作のためのシーケンス番号発生器 (SNG) 134 に記憶された基準ベクトルデータサンプルセット 130(0) ~ 130(X) の中から基準ベクトルデータサンプル 130 も受信する。実行ユニット 84(0) ~ 84(X) は、相関ベクトル処理動作の一部として、基準ベクトルデータサンプルセット 130(0) ~ 130(X) を入力ベクトルデータサンプルセット 86(0) ~ 86(X) と相関させるように構成される。しかしながら、シーケンス番号発生器 (SNG) 134 はレジスタまたは他のファイルでもあり得ることに留意されたい。この例における相関ベクトル処理動作は CDMA 相関ベクトル命令向けなので、シーケンス番号発生器 134 は、基準ベクトルデータサンプルセット 130(0) ~ 130(X) を供給するために、この実施形態において提供される。基準ベクトルデータサンプルセット 130(0) ~ 130(X) と入力ベクトルデータサンプルセット 86(0) ~ 86(X) との間の相関が高い場合、基準ベクトルデータサンプルセット 130(0) ~ 130(X) は、入力ベクトルデータサンプルセット 86(0) ~ 86(X) からの信号抽出に使用するための生成されたチップシーケンスとして供給される。

40

【0085】

[00134]たとえば、CDMA ベクトル相関命令向けの相関ベクトル処理動作は、入力ベクトルデータサンプルセット 86(0) ~ 86(X) 内のオンタイム入力ベクトルデータサンプル 86 と、入力ベクトルデータサンプルセット 86(0) ~ 86(X) 内の後発入

50

カベクトルデータサンプルとの間の相関を提供する可能性がある。たとえば、入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) 内のオンタイム入力ベクトルデータサンプル 86 は、入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) 内の偶数の入力ベクトルデータサンプル 86 (たとえば、86 (0)、86 (2)、86 (4)、... 86 (X - 1)) であり得る。入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) 内の後発入力ベクトルデータサンプル 86 は、入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) 内の奇数の入力ベクトルデータサンプル 86 (たとえば、86 (1)、86 (3)、86 (5)、... 86 (X)) であり得る。代替として、オンタイム入力ベクトルデータサンプル 86 は奇数の入力ベクトルデータサンプル 86 であり得るし、後発入力ベクトルデータサンプル 86 は偶数の入力ベクトルデータサンプル 86 であり得る。相関ベクトル処理動作の結果、オンタイム入力ベクトルデータサンプル 86 のための、結果として生じる相関出力ベクトルデータサンプルセット 132 (0) ~ 132 (X)、および後発入力ベクトルデータサンプル 86 は、信号抽出に入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) からのオンタイム入力ベクトルデータサンプルを使用すべきか、または後発入力ベクトルデータサンプルを使用すべきかを決定するために使用され得る。たとえば、オンタイム相関ベクトル処理動作は、以下の式に従って供給される場合がある、

【0086】

【数2】

$$R_{xy}^{OT}[n] = \sum_{l=0}^{i=511} y[2l] * x[2l+n]$$

【0087】

ここで、

n は入力信号サンプルの数であり、

x[n] はデジタル化入力信号 66 であり、

y[n] は基準信号であり、

l はサンプル数である。

【0088】

[00135] 後発相関ベクトル処理動作は、以下の式に従って提供される場合がある、

【0089】

【数3】

$$R_{xy}^{LT}[n] = \sum_{l=0}^{l=511} y[2l+1] * x[2l+1+n]$$

【0090】

ここで、

n は入力信号サンプルの数であり、

x[n] はデジタル化入力信号 66 であり、

y[n] は基準信号であり、

l はサンプル数である。

基準信号 y[n] (すなわち、基準ベクトルデータサンプル) は複素数であり得る。一態様では、VPE22 (2) は、(たとえば、シーケンス番号発生器 134 から) 基準信号を受信する場合がある。VPE22 (2) は、オンタイム相関演算と後発相関演算とを実行するために受信された基準信号を直接使用する場合があり、その場合、上記の式における基準信号 y[n] は、受信された基準信号を表す場合がある。代替として、VPE22 (2) は、オンタイム相関演算と後発相関演算とを実行するために基準信号を使用する前に、受信された基準信号の複素共役を計算する場合があり、その場合、上記の式における基準信号 y[n] は、受信された基準信号の共役を表す場合がある。

【 0 0 9 1 】

[00136]引き続き図 1 1 を参照すると、実行ユニット 8 4 (0) ~ 8 4 (X) は、各々、実行ユニット 8 4 (0) ~ 8 4 (X) 内の中間相関出力ベクトルデータサンプルを供給するために、相関ベクトル処理動作の各処理ステージの間に、基準ベクトルデータサンプルセット 1 3 0 (0) ~ 1 3 0 (X) を、シフトされた入力ベクトルデータサンプルセット 8 6 S (0) ~ 8 6 S (X) のシフトされた入力ベクトルデータサンプル 8 6 S (0)、8 6 S (1)、... 8 6 S (X) と乗算するように構成される。中間相関出力ベクトルデータサンプルセットは、実行ユニット 8 4 (0) ~ 8 4 (X) の各々において累算される（すなわち、前に累算された相関出力ベクトルデータサンプルが現在の相関出力ベクトルデータサンプルに加算される）。これにより、実行ユニット 8 4 (0) ~ 8 4 (X) によって生成された中間相関出力ベクトルデータサンプルセットを記憶しシフトする必要なしに、V P E 2 2 (2) によるさらなる使用および / または処理のためにそれぞれのベクトルデータファイル 8 2 (0) ~ 8 2 (X) に戻して記憶されるべき入力ベクトルデータサンプルセット 8 6 (0)、8 6 (1)、... 8 6 (X) ごとに、それぞれ、出力データフローパス 9 8 (0) ~ 9 8 (X) 上の実行ユニット出力 9 6 (0) ~ 9 6 (X) 上に実行ユニット 8 4 (0) ~ 8 4 (X) によって供給される、最終的な、結果として生じる相関出力ベクトルデータサンプルセット 1 3 2 (0) ~ 1 3 2 (X) がもたらされる。

10

【 0 0 9 2 】

[00137]さらに、図 1 1 の V P E 2 2 (2) 内に設けられた同じ構成要素およびアーキテクチャが、図 4 の V P E 2 2 (1) 内に設けられることに留意されたい。シーケンス番号発生器 1 3 4 は、フィルタ係数 9 2 (0) ~ 9 2 (Y - 1) または基準ベクトルデータサンプルセット 1 3 0 (0) ~ 1 3 0 (X) と処理されるべき他のデータを供給することができるグローバルレジスタファイル 4 0 と、マルチプレクサ 1 3 6 によって加算および多重化される。したがって、図 1 1 の V P E 2 2 (2) は、前述のフィルタベクトル処理動作と、マルチプレクサ 1 3 6 の制御による、ここで説明され、下記でさらに詳細に説明される相関ベクトル処理動作の両方を提供することができる。マルチプレクサ 1 3 6 は、V P E 2 2 (2) によって実行されているベクトル命令に基づいて制御される選択器信号 1 3 8 によって制御され得る。フィルタベクトル命令の場合、選択器信号 1 3 8 は、実行ユニット 8 4 (0) ~ 8 4 (X) に供給されるべきグローバルレジスタファイル 4 0 からのフィルタ係数 9 2 (0) ~ 9 2 (Y - 1) を供給するように構成され得る。相関ベクトル命令の場合、選択器信号 1 3 8 は、実行ユニット 8 4 (0) ~ 8 4 (X) に供給されるべきシーケンス番号発生器 1 3 4 からの基準ベクトルデータサンプルセット 1 3 0 (0) ~ 1 3 0 (X) を選択するように構成され得る。

20

30

【 0 0 9 3 】

[00138]引き続き図 1 1 を参照すると、下記でより詳細に説明されるように、タップ付き遅延線 7 8 (0)、7 8 (1) は、処理されているベクトル命令に従って制御されるようにプログラム可能である。相関ベクトル命令またはタップ付き遅延線 7 8 を利用しない他の命令が処理されていない場合、タップ付き遅延線 7 8 は、ベクトルデータファイル 8 2 (0) ~ 8 2 (X) と実行ユニット 8 4 (0) ~ 8 4 (X) との間の入力データフローパス 8 0 (0) ~ 8 0 (X) に含まれないようにプログラムされ得る。この実施形態では、前に説明されたように、2 つのタップ付き遅延線 7 8、プライマリタップ付き遅延線 7 8 (0) およびシャドウタップ付き遅延線 7 8 (1) が設けられ、シャドウタップ付き遅延線 7 8 (1) はこの実施形態ではオプションである。前に説明されたように、タップ付き遅延線 7 8 がないと、実行ユニット 8 4 (0) ~ 8 4 (X) にシフトされた中間入力ベクトルデータサンプルセットを再び供給するために、別個のシフティングプロセスが実行される必要があるはずであり、それにより、遅延時間が増大し、さらなる電力が消費される。さらに、相関ベクトル処理動作中、ベクトルデータファイル 8 2 (0) ~ 8 2 (X) からのシフトされた入力ベクトルデータサンプルセット 8 6 S (0) ~ 8 6 S (X) の再フェッチ遅延によって、V P E 2 2 (2) 内の入力データフローパス 8 0 (0) ~ 8 0 (X) および出力データフローパス 9 8 (0) ~ 9 8 (X) の効率が制限されない。シフト

40

50

された入力ベクトルデータサンプルセット 86 S (0) ~ 86 S (X) は、実行ユニット 84 (0) ~ 84 (X) に局在するタップ付き遅延線 78 によって供給される。実行ユニット 84 (0) ~ 84 (X) におけるベクトル処理は、データフローの制限ではなく、コンピュータリソースのみによって制限される。

【 0 0 9 4 】

[00139]さらに、図 1 1 の V P E 2 2 (2) によって実行される関連ベクトル処理動作は、タップ付き遅延線 78 を利用することによってより精密にされ得るが、これは、実行ユニット 84 (0) ~ 84 (X) 内の中間関連処理ステージのための出力累算がベクトルデータファイル 82 (0) ~ 82 (X) に記憶される必要がないからである。実行ユニット 84 (0) ~ 84 (X) からベクトルデータファイル 82 (0) ~ 82 (X) への中間ベクトルデータサンプルセットの記憶は、丸めをもたらす可能性がある。したがって、次の中間ベクトルデータサンプルセットがベクトル処理動作のために実行ユニット 84 (0) ~ 84 (X) に供給されるとき、ベクトル処理動作の各乗算フェーズの間に任意の丸め誤差が伝搬および加算される。対照的に、図 1 1 の V P E 2 2 (2) の例では、実行ユニット 84 (0) ~ 84 (X) によって計算された中間関連出力ベクトルデータサンプルセットは、ベクトルデータファイル 82 (0) ~ 82 (X) に記憶される必要がない。前の中間関連出力ベクトルデータサンプルセットは、次の関連出力ベクトルデータサンプルセットのための中間関連出力ベクトルデータサンプルセットと累算され得るが、これは、タップ付き遅延線 78 が、処理されるべきベクトル処理動作の間に、実行ユニット 84 (0) ~ 84 (X) にシフトされた入力ベクトルデータサンプルセット 86 S (0) ~ 86 S (X) を供給するからであり、結果は前の関連出力ベクトルデータサンプルセットのための前のベクトルデータサンプルセットと累算される。

【 0 0 9 5 】

[00140]上記図 4 の V P E 2 2 (1) 内に設けられた構成要素の前の説明は、図 1 1 の V P E 2 2 (2) に等しく適用可能であり、したがって再び記載されない。

【 0 0 9 6 】

[00141]図 1 1 の V P E 2 2 (2) のさらなる詳細および特徴、ならびにこの実施形態における入力データフローパス 80 (0) ~ 80 (X) 内の実行ユニット 84 (0) ~ 84 (X) にシフトされた入力ベクトルデータサンプルセット 86 S (0) ~ 86 S (X) を供給するためのタップ付き遅延線 78 のさらなる説明が次に記載される。この関連で、図 1 2 A および図 1 2 B は、例示的な関連ベクトル命令に従って、タップ付き遅延線 78 を利用する図 1 1 の V P E 2 2 (2) において実行され得る例示的な関連ベクトル処理動作 140 を示すフローチャートである。図 1 2 A および図 1 2 B は、例示的な関連 / 共分散ベクトル処理動作に従って、インターリーブされたオンタイムおよび後発の入力ベクトルデータサンプルセットがフェッチされる、図 1 1 の V P E 2 2 (2) において並列に実行され得る例示的な関連 / 共分散ベクトル処理動作を示すフローチャートである。

【 0 0 9 7 】

[00142]図 1 3 ~ 図 1 7 B において提供される例を参照して、図 1 2 A および図 1 2 B の関連ベクトル処理動作 140 において実行される例示的なタスクが記載される。図 1 2 A を参照すると、関連ベクトル命令に従って関連ベクトル処理動作 140 において処理されるべき入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) が、関連ベクトル処理動作 140 のために、ベクトルデータファイル 82 (0) ~ 82 (X) から入力データフローパス 80 (0) ~ 80 (X) の中にフェッチされる (ブロック 142)。図 1 1 の V P E 2 2 (2) に関して上記で説明されたように、入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) は、実行ユニット 84 (0) ~ 84 (X) 内のシーケンス番号発生器 134 から受信された基準ベクトルデータサンプルセット 130 (0) ~ 130 (X) と乗算される。たとえば、図 1 3 は、シーケンス番号発生器 134 内の基準ベクトルデータサンプルセット 130 (0) ~ 130 (X) を示す。この例では、入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) 内の 16 個の入力ベクトルデータサンプル 86 (0)、86 (1)、... 86 (15) と関連されるべき、グローバルレジスタファイ

ル40に記憶された16個の基準ベクトルデータサンプル130(0)、130(1)、
... 130(15)が存在する。上記で前に説明された図6Bは、ベクトルデータファ
イル82(0)~82(X)に記憶された例示的な入力ベクトルデータサンプルセット8
6(0)~86(X)を示したが、それはこの例においても適用可能であり、したがって
、ここでは再び記載されない。

【0098】

[00143] 関連ベクトル処理動作140において関連されるべき入力ベクトルデー
タサンプルセット86(0)~86(X)および基準ベクトルデータサンプルセット130(0)
~130(X)の幅に応じて、ベクトル命令のプログラミングに従う関連ベクトル処理
動作140を提供するために、図11のVPE22(2)内のベクトルデータレーン10
0(0)~100(X)の1つ、いくつか、またはすべてが利用され得る。ベクトルデー
タファイル82(0)~82(X)の幅全体が必要な場合、すべてのベクトルデータレー
ン100(0)~100(X)が関連ベクトル処理動作140に利用され得る。関連ベク
トル処理動作140は、関連ベクトル処理動作140に利用され得るベクトルデータレー
ン100(0)~100(X)のサブセットを必要とするにすぎない場合があることに留
意されたい。これは、入力ベクトルデータサンプルセット86(0)~86(X)の幅が
すべてのベクトルデータファイル82(0)~82(X)の幅よりも小さいからであり得
るし、ここで、関連ベクトル処理動作140と並列に実行されるべき他のベクトル処理動
作にさらなるベクトルデータレーン100を利用することが望ましい。現在の例を説明す
る目的で、関連ベクトル処理動作140において利用される入力ベクトルデータサ
ンプルセット86(0)~86(X)および基準ベクトルデータサンプルセット130(0)~
130(X)が、VPE22(2)内のすべてのベクトルデータレーン100(0)~1
00(X)を要すると想定する。

【0099】

[00144] 図12Aに戻って参照すると、関連ベクトル処理動作140のための第1の入
力ベクトルデータサンプルセット86S(0)~86(X)としてタップ付き遅延線78
にロードされるために、フェッチされた入力ベクトルデータサンプルセット86(0)~
86(X)が、ベクトルデータファイル82(0)~82(X)から入力データフローパス80(0)~80(X)に供給される(ブロック144)。入力ベクトルデータサ
ンプルセット86(0)~86(X)は、関連ベクトル処理動作140のために実行ユニット
84(0)~84(X)によって処理されるべき入力ベクトルデータサンプルセット86
(0)~86(X)として、プライマリタップ付き遅延線78(0)の中にロードされる
。プライマリタップ付き遅延線78(0)の中にロードされた入力ベクトルデータサ
ンプルセット86(0)~86(X)は、関連ベクトル処理動作140の最初の動作のために
シフトされない。次の入力ベクトルデータサンプルセット86N(0)~86N(X)も
、実行ユニット84(1)~84(X)によって処理されるべき次の入力ベクトルデー
タサンプルセット86N(0)~86N(X)として、シャドウタップ付き遅延線78(1)
の中にロードされ得る。上記で前に説明され、下記でさらに詳細に説明されるように、
タップ付き遅延線78の目的は、関連ベクトル処理動作140の動作の間に、次の関連演
算のために実行ユニット84(0)~84(X)にシフトされた入力ベクトルデータサ
ンプルセット86S(0)~86S(X)を供給するために、入力ベクトルデータサ
ンプルセット86(0)~86(X)のシフトを提供することである。実行ユニット84(0)
~84(X)によって実行される関連ベクトル処理動作140の各処理ステージの間に、
実行ユニット84(0)~84(X)にシフトされた入力ベクトルデータサンプルセット
86S(0)~86S(X)を供給するために、入力ベクトルデータサンプル86はプライ
マリタップ付き遅延線78(0)内でシフトされる。このようにして、入力ベクトルデ
ータサンプルセット86(0)~86(X)は、関連ベクトル処理動作140の関連演算
ごとに、記憶、ベクトルデータファイル82(0)~82(X)内でシフト、および再フ
ェッチされる必要がない。

【0100】

10

20

30

40

50

[00145]この関連で、図14は、図11のVPE22(2)内に設けられ得る例示的なタップ付き遅延線78を示す。この実施形態では、タップ付き遅延線78は、シャドウタップ付き遅延線78(1)とプライマリタップ付き遅延線78(0)とを備える。上記で前に説明されたように、この例におけるプライマリタップ付き遅延線78(0)は、入力ベクトルデータサンプル86の解像度が8ビット長に落ちることを可能にするために、複数の8ビットプライマリパイプラインレジスタ120から構成される。実行ユニット84(0)~84(X)によって処理される最初の入力ベクトルデータサンプルセット86(0)~86(X)は、相関ベクトル処理動作140の最初の相関演算のためにこの例ではシフトされない。実行ユニット84(0)~84(X)が相関ベクトル処理動作140のために次の相関演算を処理するとき、プライマリタップ付き遅延線78(0)に記憶された入力ベクトルデータサンプルセット86(0)~86(X)内の入力ベクトルデータサンプル86は、シフトされた入力ベクトルデータサンプルセット86S(0)~86S(X)になるために、図14の矢印によって示されたように、プライマリパイプラインレジスタ120(0)~120(4X+3)内でシフトされる。このようにして、実行ユニット84(0)~84(X)は、ベクトルデータファイル82(0)~82(X)から入力ベクトルデータサンプルセット86(0)~86(X)を記憶、シフト、および再フェッチする必要なしに、シフトされた入力ベクトルデータサンプルセット86S(0)~86S(X)を受信し、それらの相関ベクトル処理動作140を実行することによって、十分利用される。

【0101】

[00146]相関ベクトル処理動作140のためにプライマリタップ付き遅延線78(0)およびシャドウタップ付き遅延線78(1)において実行されるシフトの回数は、相関されるべきサンプルの数に依存する。ベクトルデータファイル82(0)~82(X)からプライマリタップ付き遅延線78(0)およびシャドウタップ付き遅延線78(1)の中にフェッチされた入力ベクトルデータサンプルセット86(0)~86(X)内の入力ベクトルデータサンプル86の数が、相関ベクトル処理動作140における相関演算の数よりも大きい場合、実行ユニット84(0)~84(X)は、任意のさらなる入力ベクトルデータサンプルセット86(0)~86(X)がベクトルデータファイル82(0)~82(X)から再フェッチされることなく、相関ベクトル処理動作140を実行することができる。しかしながら、相関ベクトル処理動作140における相関演算の数が、ベクトルデータファイル82(0)~82(X)からプライマリタップ付き遅延線78(0)およびシャドウタップ付き遅延線78(1)の中にフェッチされた入力ベクトルデータサンプルセット86(0)~86(X)内の入力ベクトルデータサンプル86の数よりも大きい場合、相関ベクトル処理動作140の一部として、さらなる入力ベクトルデータサンプルセット86(0)~86(X)がベクトルデータファイル82(0)~82(X)からフェッチされ得る。

【0102】

[00147]この実施形態では、プライマリパイプラインレジスタ120(0)~120(4X+3)はまとめて、ベクトルデータファイル82(0)~82(X)の幅である。15に等しい「X」を有する幅が512ビットであるベクトルデータファイル82(0)~82(X)の例では、512ビット(すなわち、64個のレジスタ×各8ビット)の合計幅を提供するために、各々が8ビットの幅である64個の合計プライマリパイプラインレジスタ120(0)~120(63)が存在する。したがって、この例では、プライマリタップ付き遅延線78(0)は、1つの入力ベクトルデータサンプルセット86(0)~86(X)の幅全体を記憶することが可能である。この例では、8ビット幅のプライマリパイプラインレジスタ120(0)~120(4X+3)を設けることによって、入力ベクトルデータサンプルセット86(0)~86(X)は、8ビット相関ベクトル処理動作のために8ビットベクトルデータサンプルサイズにシフトダウンされ得る。たとえば、16ビットまたは32ビットのサンプルなどのより大きい入力ベクトルデータサンプル86のサイズが相関ベクトル処理動作140に望ましい場合、入力ベクトルデータサンプルセ

ット86(0)~86(X)は、プライマリパイプラインレジスタ120(0)~120(4X+3)において、1度に2つのプライマリパイプラインレジスタ120によってシフトされ得る。

【0103】

[00148]図15Aは、相関ベクトル処理命令140の第1のクロックサイクル(CYCLE0)の間に、ベクトルデータファイル82(0)~82(X)からプライマリタップ付き遅延線78(0)の中にロードされた入力ベクトルデータサンプルセット86(0)~86(X)を示す。最初の入力ベクトルデータサンプルセット86(0)~86(X)が入力ベクトルデータサンプルX1~X32としてプライマリタップ付き遅延線78(0)の中にロードされるが、64個の入力ベクトルデータサンプルが供給される。プライマリパイプラインレジスタ120(0)~120(2X+1)(図14も参照)は、入力ベクトルデータサンプルセット86(0)~86(X)からオンタイム入力ベクトルデータサンプルおよび後発入力ベクトルデータサンプルをロードされる。たとえば、プライマリタップ付き遅延線78(0)の中に(および下記で後により詳細に説明されるように、シャドウタップ付き遅延線78(1)の中にも)、入力ベクトルデータサンプルセット86(0)~86(X)のオンタイム入力ベクトルデータサンプルと後発入力ベクトルデータサンプルとをロードするために、特殊なベクトル命令がサポートされる場合がある。たとえば、プライマリパイプラインレジスタ122(0)、122(1)、122(2X+2)、および122(2X+3)はまとめて、入力ベクトルデータサンプル86(0)を含んでいる。プライマリパイプラインレジスタ122(0)、122(1)は、X(0)およびX(1)であるオンタイム入力ベクトルデータサンプル86OT(0)を含んでおり、ここで「OT」は「オンタイム」を意味する。プライマリパイプラインレジスタ122(2X+2)、122(2X+3)は、X(1)およびX(2)である後発入力ベクトルデータサンプル86L(0)を含んでおり、ここで「L」は「後発」を意味する。プライマリタップ付き遅延線78(0)内のこの入力ベクトルデータサンプル86の記憶パターンは、他のプライマリパイプラインレジスタ122(2)~122(2X+1)および122(2X+4)~122(4X+3)について繰り返される(図14参照)。

【0104】

[00149]図14に戻って参照すると、シャドウタップ付き遅延線78(1)もタップ付き遅延線78内に設けられる。シャドウタップ付き遅延線78(1)は、次のベクトル処理動作のためにベクトルデータファイル82(0)~82(X)から次の入力ベクトルデータサンプルセット86N(0)~86N(X)をラッチまたは輸送するために利用され得る。シャドウタップ付き遅延線78(1)はまた、入力ベクトルデータサンプルの解像度が、プライマリタップ付き遅延線78(0)と同様に8ビット長に落ちることを可能にするために、複数の8ビットシャドウパイプラインレジスタ122から構成される。シャドウパイプラインレジスタ122はまとめて、この例では512ビットであるベクトルデータファイル82(0)~82(X)の幅であり、その結果、シャドウタップ付き遅延線78(1)も、プライマリタップ付き遅延線78(0)のように、1つの入力ベクトルデータサンプルセット86(0)~86(X)の幅全体を記憶することが可能である。したがって、この実施形態では、プライマリタップ付き遅延線78(0)に含まれるシャドウパイプラインレジスタ122(0)~122(4X+3)の数は、合計16であるベクトルデータレーン100(0)~100(X)の数の4倍であり、この例では各ベクトルデータレーン100(0)~100(X)が各々32ビットをサポートすることが可能である。したがって、プライマリパイプラインレジスタ120の数も、この例では合計512ビット(すなわち、64個のレジスタ×各8ビット)用に合計64である。

【0105】

[00150]図15Bは、相関ベクトル処理命令140の第2のクロックサイクル(CYCLE1)の間に、シャドウタップ付き遅延線78(1)の中にロードされた次の入力ベクトルデータサンプルセット86N(0)~86N(X)を示す。相関ベクトル処理動作140の実行をセットアップするために、ベクトルデータファイル82(0)~82(X)

からの最初の入力ベクトルデータサンプルセット $86(0) \sim 86(X)$ がプライマリタップ付き遅延線 $78(0)$ の中にロードされた後に、次の入力ベクトルデータサンプルセット $86N(0) \sim 86N(1)$ がシャドウタップ付き遅延線 $78(1)$ の中にロードされる。この次の入力ベクトルデータサンプルセット $86N(0) \sim 86N(X)$ が、オンタイム入力ベクトルデータサンプル $86OT$ と後発入力ベクトルデータサンプル $86L$ の両方とともに、入力ベクトルデータサンプル $X(32) \sim X(63)$ としてシャドウタップ付き遅延線 $78(1)$ の中にロードされる。この例では、上記で説明されたプライマリタップ付き遅延線 $78(0)$ において提供される記憶パターンのように、 $X(32)$ および $X(33)$ が入力ベクトルデータサンプル $86(0)$ のオンタイム入力ベクトルデータサンプル $86OT$ を形成し、 $X(33)$ および $X(34)$ が入力ベクトルデータサンプル $86(0)$ の後発入力ベクトルデータサンプル $86L$ を形成することに留意されたい。入力ベクトルデータサンプル 86 を一緒にグループ化して、入力ベクトルデータサンプルセット $86(0) \sim 86(X)$ を形成するために、他のパターンが提供される可能性がある。シーケンス番号発生器 134 からの基準ベクトルデータサンプルセット $130(0) \sim 130(X)$ から、関連ベクトル処理動作 140 の第 1 の処理ステージの間に関連された基準ベクトルデータサンプル 130 (すなわち、 $Y(0)$ および $Y(1)$) はまた、関連ベクトル処理動作 140 において使用するための図 $15B$ の実行ユニット $84(0) \sim 84(X)$ へのレジスタ(「 C 」)内で供給されるものとして示される。

【 0106 】

[00151] 図 14 に戻って参照すると、関連ベクトル処理動作 140 の各処理ステージの間に、入力ベクトルデータサンプルセット $86(0) \sim 86(X)$ 内の入力ベクトルデータサンプル 86 がプライマリタップ付き遅延線 $78(0)$ 内でシフトされるとき、シャドウパイプラインレジスタ 122 に記憶された次の入力ベクトルデータサンプル $86N$ も、シャドウタップ付き遅延線 $78(1)$ のシャドウパイプラインレジスタ 122 内でシフトされる。この例では、入力ベクトルデータサンプルセット $86(0) \sim 86(X)$ の入力ベクトルデータサンプル 86 は、オンタイムバージョンおよび後発バージョンとして記憶されるので、図 14 のタップ付き遅延線 $78(0)$ と $78(1)$ との間で提供されるシフトパターンは、図 7 のタップ付き遅延線 $78(0)$ と $78(1)$ との間で提供されるシフトパターンとは異なる。図 14 に示されたように、オンタイム入力ベクトルデータサンプル $86OT$ は、シャドウタップ付き遅延線 $78(1)$ 内のシャドウパイプラインレジスタ $122(0)$ から、プライマリタップ付き遅延線 $78(0)$ 内のプライマリパイプラインレジスタ $120(2X+1)$ にシフトされる。同じく、後発入力ベクトルデータサンプル $86L$ は、シャドウタップ付き遅延線 $78(1)$ 内のシャドウパイプラインレジスタ $122(2X+2)$ から、プライマリタップ付き遅延線 $78(0)$ 内のプライマリパイプラインレジスタ $120(4X+3)$ にシフトされる。このようにして、入力ベクトルデータサンプル 86 のシフトが関連ベクトル処理動作 140 の間に発生するとき、オンタイム入力ベクトルデータサンプル $86OT$ および後発入力ベクトルデータサンプル $86OT$ は、タップ付き遅延線 $78(0)$ 、 $78(1)$ 内で互いから隔離され続ける。

【 0107 】

[00152] 関連ベクトル処理動作 140 の処理ステージが実行ユニット $84(0) \sim 84(X)$ において進行し、最終的に、シャドウタップ付き遅延線 $78(1)$ に最初に記憶された次の入力ベクトルデータサンプルセット $86N(0) \sim 86N(X)$ 全体は、処理のために実行ユニット $84(0) \sim 84(X)$ に供給されるために、プライマリタップ付き遅延線 $78(0)$ の中に完全にシフトされる。このようにして、関連ベクトル処理動作 140 が現在の入力ベクトルデータサンプルセット $86(0) \sim 86(X)$ に対して完了した後、実行ユニット $84(0) \sim 84(X)$ は、次いで、必要な場合、遅延なく、次の関連ベクトル処理動作 140 のための現在の入力ベクトルデータサンプルセット $86(0) \sim 86(X)$ として、プライマリタップ付き遅延線 $78(0)$ に記憶された前の次の入力ベクトルデータサンプルセット $86N(0) \sim 86N(X)$ を供給され得る。

【 0108 】

[00153] 最初の入力ベクトルデータサンプルセット $86(0) \sim 86(X)$ および次の入力ベクトルデータサンプルセット $86N(0) \sim 86N(X)$ が、それぞれ、プライマリタップ付き遅延線 $78(0)$ およびシャドウタップ付き遅延線 $78(1)$ の中にロードされた後、図 15B に示されたように、プライマリタップ付き遅延線 $78(0)$ 内に供給された最初の入力ベクトルデータサンプルセット $86(0) \sim 86(X)$ は、相関ベクトル処理動作 140 の最初の処理ステージにおいて処理されるために、それぞれの実行ユニット $84(0) \sim 84(X)$ に供給される (図 12A のブロック 146)。最初の入力ベクトルデータサンプルセット $86(0) \sim 86(X)$ は、実行ユニット $84(0) \sim 84(X)$ によって処理されている現在の入力ベクトルデータサンプルセット $86(0) \sim 86(X)$ になる。図 11 の VPE22(2) において示されたように、現在の入力ベクトルデータサンプル $86(0)$ は実行ユニット $84(0)$ に供給され、現在の入力ベクトルデータサンプル $86(1)$ は実行ユニット $84(1)$ に供給され、以下同様である。相関ベクトル処理動作 140 の現在の処理ステージにおいて、入力ベクトルデータサンプルセット $86(0) \sim 86(X)$ と相関されるべき基準ベクトルデータ入力サンプル $130(0) \sim 130(X)$ が実行ユニット $84(0) \sim 84(X)$ に供給される (図 12A のブロック 148)。

【0109】

[00154] 次に、実行ユニット $84(0) \sim 84(X)$ が、相関ベクトル処理動作 140 (図 12A のブロック 150) を実行する。より詳細には、実行ユニット $84(0) \sim 84(X)$ は、演算: $\text{オンタイム入力ベクトルデータサンプル } 86OT \text{ のための } R(OT)[n] = y[0] * x[n]$ および後発入力ベクトルデータサンプル $86L$ のための $R(L)[n] = y[1] * x[1+n]$ に従って、最初の処理ステージの間に現在の入力ベクトルデータサンプルセット $86(0) \sim 86(X)$ を基準ベクトルデータサンプル 130 と乗算し、ここで、 $y[]$ は指定された基準ベクトルデータサンプル 130 であり、 $x[n]$ は現在の入力ベクトルデータサンプルセット $86(0) \sim 86(X)$ である。相関の結果は、現在のオンタイム相関出力ベクトルデータサンプルセット $R(OT)[n]$ および現在の後発相関出力ベクトルデータサンプルセット $R(L)[n]$ である。次いで、実行ユニット $84(0) \sim 84(X)$ が、新しい前の入力ベクトルデータサンプルセット $86(0) \sim 86(X)$ を供給するために、各現在の、結果として生じる相関ベクトルデータサンプルセットを、実行ユニット $84(0) \sim 84(X)$ によって計算された前の、結果として生じる相関ベクトルデータサンプルセットと累算する (図 12B のブロック 152)。相関ベクトル処理動作 140 の最初の処理ステージでは、前の、結果として生じる相関出力ベクトルデータサンプルセット $132(0) \sim 132(X)$ は存在しない。したがって、相関ベクトル処理動作 140 の 2 番目の次の処理ステージのために、最初 / 現在の、結果として生じる相関出力ベクトルデータサンプルセット $132(0) \sim 132(X)$ が前の入力ベクトルデータサンプルセット $86(0) \sim 86(X)$ になるにすぎない。

【0110】

[00155] 相関ベクトル処理動作 140 のすべての処理ステージが完了した場合 (図 12B のブロック 154)、ベクトルデータファイル $82(0) \sim 82(X)$ に供給され記憶されるために、出力データフローパス $98(0) \sim 98(X)$ 内の、結果として生じる相関出力ベクトルデータサンプルセット $132(0) \sim 132(X)$ として、累算された前の、結果として生じる相関出力ベクトルデータサンプルセット $132(0) \sim 132(X)$ が供給される (図 12B のブロック 157)。相関ベクトル処理動作 140 のすべての処理ステージが完了していなかった場合 (図 12A のブロック 154)、シフトされた入力ベクトルデータサンプルセット $86S(0) \sim 86S(X)$ を供給するために、シフトされた入力ベクトルデータサンプルセット $86S(0) \sim 86S(X)$ が、相関ベクトル処理動作 140 のための次の場所にタップ付き遅延線 $78(0)$ 、 $78(1)$ 内でシフトされる (図 12B のブロック 156)。シフトされた入力ベクトルデータサンプルセット $86S(0) \sim 86S(X)$ は、前の、結果として生じる相関出力ベクトルデータサン

ルセット 1 3 2 (0) ~ 1 3 2 (X) と累算されるように、次の、結果として生じる相関出力ベクトルデータサンプルセット 1 3 2 (0) ~ 1 3 2 (X) を計算するために供給される。タップ付き遅延線 7 8 (0)、7 8 (1) 内で入力ベクトルデータサンプル 8 6 をシフトすることは、図 1 4 に関して詳細に上記で前述された。

【 0 1 1 1 】

[00156] 図 1 5 C は、次の相関処理動作 1 4 0、オンタイム入力ベクトルデータサンプル 8 6 S O T のための $R(O T)[n] = y[2] * x[2 + n]$ および後発入力ベクトルデータサンプル 8 6 S L のための $R(L)[n] = y[3] * x[3 + n]$ のための新たなシフトされた入力ベクトルデータサンプルセット 8 6 S (0) ~ 8 6 S (X) になるために、相関ベクトル処理動作 1 4 0 の 2 番目の処理ステージにおいて、入力ベクトルデータサンプルセット 8 6 (0) ~ 8 6 (X) がシフトされたときのタップ付き遅延線 7 8 のコンテンツを示す。プライマリタップ付き遅延線 7 8 (0) 内の入力ベクトルデータサンプルセット 8 6 (0) ~ 8 6 (X) が、2 つの入力ベクトルデータサンプル 8 6 によってシフトされる。たとえば、 $x(2)$ および $x(3)$ の図 1 5 B の入力ベクトルデータサンプル 8 6 O T (1) が、次に図 1 5 C の入力ベクトルデータサンプル 8 6 S (0) の中にシフトされる。シフトされた入力ベクトルデータサンプルセット 8 6 S (0) ~ 8 6 S (X) は、現在の入力ベクトルデータサンプルセット 8 6 (0) ~ 8 6 (X) になる。実行ユニット 8 4 (0) ~ 8 4 (X) に供給された基準ベクトルデータサンプル 1 3 0 はまた、この例では $Y(2)$ および $Y(3)$ である基準ベクトルデータサンプル 1 3 0 である。

【 0 1 1 2 】

[00157] 引き続き図 1 2 B を参照すると、次の基準ベクトルデータサンプル 1 3 0 と乗算されるために、プライマリタップ付き遅延線 7 8 (0) から (およびシャドウタップ付き遅延線 7 8 (1) の一部分から) 実行ユニット 8 4 (0) ~ 8 4 (X) に次のシフトされた入力ベクトルデータサンプルセット 8 6 S (0) ~ 8 6 S (X) を供給する (図 1 2 A のブロック 1 5 0) ことによってプロセスが繰り返し、結果として生じる相関出力ベクトルデータサンプルセット 1 3 2 (0) ~ 1 3 2 (X) が、前の、結果として生じる相関出力ベクトルデータサンプルセット 1 3 2 (0) ~ 1 3 2 (X) と累算される (図 1 2 B のブロック 1 5 2)。図 1 5 D は、例示的な相関ベクトル処理動作 1 4 0 の最後の処理ステージの間に、タップ付き遅延線 7 8 (0)、7 8 (1) 内に存在する入力ベクトルデータサンプル 8 6 の状態を示す。この例では、図 1 5 D に示されたように、タップ付き遅延線 7 8 のフルデータ幅は、入力ベクトルデータサンプルセット 8 6 (0) ~ 8 6 (X) に利用されたが、オンタイム入力ベクトルデータサンプル 8 6 O T と後発入力ベクトルデータサンプル 8 6 L との間で分割されるので、相関ベクトル処理動作 1 4 0 のための 1 6 個の処理ステージが存在した。図 1 5 D に示されたように、 $Y(30)$ および $Y(31)$ は、相関ベクトル処理動作 1 4 0 における最後の基準ベクトルデータサンプル 1 3 0 (X) であり、それは、図 1 3 の例では基準ベクトルデータサンプル 1 3 0 (1 5) である。シフトされた入力ベクトルデータサンプルセット 8 6 S (0) ~ 8 6 S (X) は (この例ではベクトルデータレーン 1 0 0 (0) ~ 1 0 0 (X) の幅である) 1 6 回シフトされており、その結果、相関ベクトル処理動作 1 4 0 のための最後の 1 6 番目の処理ステージにおいて、入力ベクトルデータサンプル $x(30)$ および $x(31)$ がプライマリタップ付き遅延線 7 8 (0) 内のシフトされた入力ベクトルデータサンプル 8 6 S (0) に記憶される。

【 0 1 1 3 】

[00158] 図 1 6 は、上記の例における例示的な 1 6 個の相関ベクトル処理ステージが完全に実行された後の、図 1 1 の V P E 2 2 (2) 内の実行ユニット 8 4 (0) ~ 8 4 (X) 内の累算器のコンテンツ (すなわち、結果として生じる相関出力ベクトルデータサンプル 1 3 2) の概略図である。結果として生じる相関出力ベクトルデータサンプルセットは、1 3 2 (0) ~ 1 3 2 (X) として示される。この例では、各実行ユニット 8 4 (0) ~ 8 4 (X) は、ベクトルデータレーン 1 0 0 (0) ~ 1 0 0 (X) ごとに並列に配置さ

れた4つの累算器を有するので、累算器 $A_{cc0} \sim A_{cc3}$ が図16に示されている。累算された、結果として生じる出力ベクトルデータサンプルは、さらなる分析および/または処理のためにそこに記憶されるべき全体の結果として生じる相関出力ベクトルデータサンプルセット $132(0) \sim 132(X)$ として、ベクトルデータファイル $82(0) \sim 82(X)$ に出力データフローパス $98(0) \sim 98(X)$ 上で供給され得る。必要な場合、ベクトルデータファイル $82(0) \sim 82(X)$ からベクトルユニットデータメモリ 32 (図2参照) に、結果として生じる相関出力ベクトルデータサンプルセット $132(0) \sim 132(X)$ の行を移動するために、特殊なベクトル命令が $VPE22(2)$ によってサポートされる場合がある。

【0114】

[00159] 上述された、結果として生じるフィルタベクトル出力データサンプルセット $94(0) \sim 94(X)$ と、結果として生じる相関出力ベクトルデータサンプルセット $132(0) \sim 132(X)$ とを含む、実行ユニット $84(0) \sim 84(X)$ によって供給される、結果として生じる出力ベクトルデータサンプルセットは、 VPE によって実行されるベクトル命令に応じて、異なるインターリーブされたフォーマットでベクトルデータファイル $82(0) \sim 82(X)$ 、 $82(31)$ に戻されて記憶され得る。各々32ビット幅であるベクトルデータファイル $82(0) \sim 82(X)$ を提供するために、この例では「 X 」は31に等しい。たとえば、図17Aに示されたように、結果として生じる出力ベクトルデータサンプルセット $158(0) \sim 158(X)$ 、 $158(31)$ は、それらの実数(「 q 」)成分および虚数(「 i 」)成分によって分離されたベクトルデータファイル $82(0) \sim 82(X)$ に記憶され得る。結果として生じる出力ベクトルデータサンプルセット $158(0) \sim 158(X)$ は、この例では $158(0)$ 、 $158(1)$ 、 \dots 、および $158(X)$ である、「 $X+1$ 」個の、結果として生じる出力ベクトルデータサンプル 158 から構成される。次のベクトル命令が、入力ベクトルデータサンプルセットとして、結果として生じる出力ベクトルデータサンプルセット $158(0) \sim 158(X)$ 、 $158(31)$ の実数成分および虚数成分に対して演算する場合など、効率目的でそれらの実数(「 q 」)成分および虚数(「 i 」)成分によって分離された、結果として生じる出力ベクトルデータサンプルセット $158(0) \sim 158(X)$ 、 $158(31)$ を記憶することはより効率的であり得る。または、結果として生じる出力ベクトルデータサンプル 158 のその実数成分および虚数成分への分離のように、ベクトルデータファイル 82 内に、結果として生じる出力ベクトルデータサンプル 158 を記憶することが可能ではない場合がある。たとえば、16ビットベクトルデータサンプルが別の16ビットベクトルデータサンプルと乗算される場合、32ビットの結果として生じるベクトルデータサンプルがもたらされる。たとえば、32ビットの結果として生じる出力ベクトルデータサンプル 158 は、図17Aの $Y0$ であり得る。 $Y0$ の虚数成分 $Y0.i158(I)$ はベクトルデータファイル $82(0)$ の $ADDRESS\ '0$ 」に記憶され得るし、 $Y0$ の実数成分 $Y0.q158(Q)$ は $ADDRESS\ 'A$ 」などの別の $ADDRESS$ に記憶され得る。

【0115】

[00160] 図17Aの結果として生じる出力ベクトルデータサンプルセット $158(0) \sim 158(X)$ 、 $158(31)$ は、偶数および奇数の結果として生じる出力ベクトルデータサンプルによってインターリーブされたベクトルデータファイル $82(0) \sim 82(X)$ 、 $82(31)$ に記憶される可能性がある。これは図17Bにおける例によって示される。図17Bに示されたように、結果として生じる出力ベクトルデータサンプル $Y0 \sim Y31$ $158(0) \sim 158(X)$ 、 $158(31)$ は、ベクトルデータファイル $82(0) \sim 82(31)$ 内の $ADDRESS\ '0$ 」および $ADDRESS\ 'A$ 」の中の偶数および奇数のベクトルデータサンプルによってインターリーブされたフォーマットで記憶される。結果として生じる出力ベクトルデータサンプル $Y0$ $158(0)$ は、ベクトルデータファイル $82(0)$ 内の $ADDRESS\ '0$ 」に記憶される。結果として生じる出力ベクトルデータサンプル $Y1$ $158(1)$ は、ベクトルデータファイル $82(1)$ 内

10

20

30

40

50

の ADDRESS「0」に記憶されないが、ベクトルデータファイル 82(0)内の ADDRESS「A」に記憶される。結果として生じる出力ベクトルデータサンプル Y2 158(2)は、ベクトルデータファイル 82(1)内の ADDRESS「0」に記憶され、以下同様である。

【0116】

[00161]いくつかのワイヤレスベースバンド動作は、データサンプルが処理される前にフォーマット変換されることを必要とする。たとえば、図 17A および図 17B においてインターリーブされたフォーマットでベクトルデータファイル 82(0)~82(X)に記憶された、結果として生じる出力ベクトルデータサンプルセット 158(0)~158(X)は、次のベクトル処理動作のためにデインターリーブされる必要があり得る。たとえば、結果として生じる出力ベクトルデータサンプル 158(0)~158(X)が CDMA 信号を表す場合、結果として生じる出力ベクトルデータサンプル 158(0)~158(X)は、信号の偶数フェーズおよび奇数フェーズを分離するためにデインターリーブされる必要があり得る。デインターリーブされた信号は、CDMA システムが信号を抽出することができるかどうかを決定するために、図 11~図 16 に関して上述された例示的な相関ベクトル処理動作などの相関処理動作において、ローカルに生成されたコードまたはシーケンス番号と相関される場合もある。従来のプログラマブルプロセッサは、複数のステップでデータサンプルのフォーマット変換を実施し、それは、ベクトルデータサンプルのフォーマット変換において、サイクルと、電力消費と、データフローの複雑化とを加える。ベクトルプロセッサは、フォーマット変換されたベクトルデータサンプルが実行ユニットに供給される前にフォーマット変換を提供するように、ベクトルデータサンプルを前処理することができる。フォーマット変換されたベクトルデータサンプルは、ベクトルデータメモリに記憶され、実行ユニットによって処理されるべきデータフォーマット変換を必要とするベクトル処理動作の一部として再フェッチされる。しかしながら、ベクトルデータサンプルのこのフォーマット前処理は、実行ユニットによるフォーマット変換されたベクトルデータサンプルの次の処理を遅延させ、実行ユニット内のコンピュータ構成要素が過少利用される原因になる。

【0117】

[00162]本明細書において下記で開示される実施形態は、図 18A および図 18B に示されたベクトルデータサンプルセットなどの、インターリーブされたベクトルデータサンプルセットの変換を提供する。たとえば、図 18A および図 18B は、様々なフォーマットでベクトルデータファイル 82(0)~82(X)に記憶されたベクトルデータサンプルセット D(0)~D(X)を示す。図 18A は、符号付き複素数(SC)の 16 ビットサンプル(SC 16)に記憶され、実数成分および虚数成分によってフォーマットインターリーブされたベクトルデータサンプルセット D(0)~D(X)を示す。32 ビットベクトルデータサンプル D(0)の 16 ビットの実数成分 D(0)(Q)および虚数成分 D(0)(I)は、32 ビットベクトルデータファイル 82(0)に記憶される。ベクトルデータサンプル D(X)の 16 ビットの実数成分 D(X)(Q)および虚数成分 D(X)(I)は、32 ビットベクトルデータファイル 82(X)に記憶される。図 18B は、SC の 8 ビットサンプル(SC 8)に記憶され、実数成分および虚数成分によってフォーマットインターリーブされたベクトルデータサンプルセット D(0)~D(X)を示す。16 ビットベクトルデータサンプル D(0)(1)の 8 ビットの実数成分 D(0)(1)(Q)および虚数成分 D(0)(1)(I)は、ベクトルデータファイル 82(0)に記憶される。16 ビットベクトルデータサンプル D(0)(0)の 8 ビットの実数成分 D(0)(0)(Q)および虚数成分 D(0)(0)(I)も、32 ビットベクトルデータファイル 82(0)に記憶される。同じく、16 ビットベクトルデータサンプル D(X)(1)の 8 ビットの実数成分 D(X)(1)(Q)および虚数成分 D(X)(1)(I)は、32 ビットベクトルデータファイル 82(X)に記憶される。16 ビットベクトルデータサンプル D(X)(0)の 8 ビットの実数成分 D(X)(0)(Q)および虚数成分 D(X)(0)(I)も、32 ビットベクトルデータファイル 82(X)に記憶される。

【 0 1 1 8 】

[00163]この関連で、図 1 9 は、図 2 の V P E 2 2 として提供され得る別の例示的な V P E 2 2 (3) の概略図である。下記でより詳細に記載されるように、図 1 9 の V P E 2 2 (3) は、ベクトルデータサンプルの再フェッチが除去または低減され、電力消費が低減される、V P E 2 2 (3) 内のベクトル処理動作のために実行ユニットに供給される入力ベクトルデータサンプルセットのインフライトフォーマット変換（たとえば、デインターリーピング）を提供するように構成される。入力ベクトルデータサンプルセットのインフライトフォーマット変換は、ベクトルデータメモリから取り出された入力ベクトルデータサンプルセットが、実行のために実行ユニットに供給される前に、ベクトルデータメモリに記憶され、そこから再フェッチされる必要なしに、フォーマット変換されることを意味する。ベクトルデータファイルからの入力ベクトルデータサンプルの再フェッチを除去または最小化して、電力消費を低減し、処理効率を改善するために、ベクトルデータファイル 8 2 (0) ~ 8 2 (X) と実行ユニット 8 4 (0) ~ 8 4 (X) との間のベクトルデータレーン 1 0 0 (0) ~ 1 0 0 (X) の各々に、フォーマット変換回路 1 5 9 (0) ~ 1 5 9 (X) が含まれる。下記でより詳細に説明されるように、入力ベクトルデータサンプルセット 8 6 (0) ~ 8 6 (X) のデインターリーピングを必要とするベクトル処理動作のために、実行ユニット 8 4 (0) ~ 8 4 (X) にフォーマット変換された入力ベクトルデータサンプルセット 8 6 F (0) ~ 8 6 F (X) を供給するために、V P E 2 2 (3) 内のフォーマット変換回路 1 5 9 (0) ~ 1 5 9 (X) において、ベクトルデータファイル 8 2 (0) ~ 8 2 (X) からの入力ベクトルデータサンプルセット 8 6 (0) ~ 8 6 (X) がフォーマット変換（たとえば、デインターリーブ）される。フォーマット変換された入力ベクトルデータサンプル 8 6 F のすべては、この例ではフォーマット変換された入力ベクトルデータサンプルセット 8 6 F (0) ~ 8 6 F (X) を備える。「X」+ 1 は、この例における入力ベクトルデータサンプル 8 6 の処理用に V P E 2 2 (3) 内に設けられる並列入力データレーンの最大数である。

【 0 1 1 9 】

[00164]このようにして、V P E 2 2 (3) における入力ベクトルデータサンプルセット 8 6 (0) ~ 8 6 (X) のフォーマット変換は、前処理、記憶、およびベクトルデータファイル 8 2 (0) ~ 8 2 (X) からの再フェッチを必要とせず、それにより、電力消費が低減される。さらに、入力ベクトルデータサンプルセット 8 6 (0) ~ 8 6 (X) のフォーマット変換は、ベクトルデータファイル 8 2 (0) ~ 8 2 (X) からのフォーマット変換された入力ベクトルデータサンプルセット 8 6 (0) ~ 8 6 (X) の前処理、記憶、および再フェッチを必要としないので、実行ユニット 8 4 (0) ~ 8 4 (X) はベクトル処理動作を実行することから遅延されない。したがって、V P E 2 2 (3) 内のデータフローパスの効率は、入力ベクトルデータサンプルセット 8 6 (0) ~ 8 6 (X) のフォーマット変換前処理の遅延によって制限されない。フォーマット変換（たとえば、デインターリーブ）された入力ベクトルデータサンプルセット 8 6 F (0) ~ 8 6 F (X) は、実行ユニット 8 4 (0) ~ 8 4 (X) に局在化されるように供給される。実行ユニット 8 4 (0) ~ 8 4 (X) におけるベクトル処理は、データフローの制限ではなく、コンピュータリソースのみによって制限される。

【 0 1 2 0 】

[00165]プライマリタップ付き遅延線 7 8 (0) およびシャドウタップ付き遅延線 7 8 (1) が図 1 9 の V P E 2 2 (3) 内に示されるが、図 1 9 の V P E 2 2 (3) 内にタップ付き遅延線を含めることは必要でないことに留意されたい。この例では、図 1 9 に示されたように、フォーマット変換回路 1 5 9 (0) ~ 1 5 9 (X) は、オプションのプライマリタップ付き遅延線 7 8 (0) に含まれ得る。この配置は、図 1 9 の V P E 2 2 (3) 内のベクトルデータファイル 8 2 (0) ~ 8 2 (X) と実行ユニット 8 4 (0) ~ 8 4 (X) との間の入力データフローパス 8 0 (0) ~ 8 0 (X) 内にフォーマット変換回路 1 5 9 (0) ~ 1 5 9 (X) を設ける。プライマリタップ付き遅延線 7 8 (0) の動作は、V P E 2 2 (1) および V P E 2 2 (2) に関して上記で前述された。上記で前に説明さ

れたように、プライマリタップ付き遅延線 78 (0) およびシャドウタップ付き遅延線 78 (1) は、ベクトル処理動作に利用される場合があり、フォーマット変換された入力ベクトルデータサンプルセット 86 F (0) ~ 86 F (X) が実行ユニット 84 (0) ~ 84 (X) に供給されることを必要とし、実行ユニット 84 (0) ~ 84 (X) も、86 S F (0) ~ 86 S F (X) と指定された、フォーマット変換されシフトされた入力ベクトルデータサンプルセットを必要とする。

【0121】

[00166] 図 19 の V P E 22 (3) 内に設けられた同じ構成要素およびアーキテクチャが、図 11 の V P E 22 (2) 内に設けられることに留意されたい。図 19 の V P E 22 (3) と図 11 の V P E 22 (2) との間の共通構成要素が、V P E 22 (2) の図 11 の構成要素と共通の要素番号とともに図 19 に示されている。上記図 11 の V P E 22 (2) のためのこれらの共通構成要素の前の記載および説明は、図 19 の V P E 22 (3) にも適用可能であり、したがってここでは再び記載されない。

【0122】

[00167] 図 19 の V P E 22 (3) のさらなる詳細および特徴、ならびにこの実施形態における入力データフローパス 80 (0) ~ 80 (X) 内の実行ユニット 84 (0) ~ 84 (X) にフォーマット変換された入力ベクトルデータサンプルセット 86 F (0) ~ 86 F (X) を供給するためのタップ付き遅延線 78 のさらなる説明が次に記載される。この関連で、図 20 は、入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) のフォーマット変換を必要とする例示的なベクトル命令に従って、フォーマット変換回路 159 (0) ~ 159 (X) を利用する図 19 の V P E 22 (3) において実行され得る、例示的なデインターリーブングフォーマット変換ベクトル処理動作 160 を示すフローチャートである。

【0123】

[00168] 図 20 を参照すると、ベクトル命令に従うベクトル処理動作 160 のための入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) が、ベクトルデータファイル 82 (0) ~ 82 (X) から入力データフローパス 80 (0) ~ 80 (X) の中にフェッチされる (ブロック 162)。たとえば、ベクトル処理動作 160 のためのフォーマット変換は、入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) が、ベクトルデータファイル 82 (0) ~ 82 (X) 内のそのインターリーブされた状態から、デインターリーブされた入力ベクトルデータサンプルセット 86 F (0) ~ 86 F (X) の中にデインターリーブされる、デインターリーブングベクトル処理動作 160 であり得る。ベクトル処理動作 160 のためにフォーマット変換されるべき入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) の幅に応じて、ベクトル命令のプログラミングに従うベクトル処理動作 160 を提供するために、図 19 の V P E 22 (3) 内のベクトルデータレーン 100 (0) ~ 100 (X) の 1 つ、いくつか、またはすべてが利用され得る。ベクトルデータファイル 82 (0) ~ 82 (X) の幅全体が必要な場合、すべてのベクトルデータレーン 100 (0) ~ 100 (X) がベクトル処理動作 160 に利用され得る。ベクトル処理動作 160 は、ベクトル処理動作 160 に利用され得るベクトルデータレーン 100 (0) ~ 100 (X) のサブセットを必要とするにすぎない場合がある。これは、入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) の幅がすべてのベクトルデータファイル 82 (0) ~ 82 (X) の幅よりも小さいからであり得るし、ここで、ベクトル処理動作 160 と並列に実行されるべき他のベクトル処理動作にさらなるベクトルデータレーン 100 を利用することが望ましい。現在の例を説明する目的で、ベクトル処理動作 160 のための入力ベクトルデータサンプルセット 86 F (0) ~ 86 F (X) にフォーマット変換された入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) が、図 19 の V P E 22 (3) 内のすべてのベクトルデータレーン 100 (0) ~ 100 (X) を要すると想定する。

【0124】

[00169] 引き続き図 20 を参照すると、ベクトル処理動作 160 に従ってフォーマット

変換されるために、フェッチされた入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) が、フォーマット変換回路 159 (0) ~ 159 (X) への入力データフローパス 80 (0) ~ 80 (X) の中に供給される (ブロック 164)。非限定的な例として、現在の入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) は、場合によっては、ベクトル処理動作 160 のために実行ユニット 84 (0) ~ 84 (X) に供給される前にフォーマット変換されるべき入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) として、プライマリタップ付き遅延線 78 (0) の中にロードされる場合がある。前に説明されたように、次の入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) は、場合によっては、実行ユニット 84 (0) ~ 84 (X) によって処理されるべき次の入力ベクトルデータサンプルセット 86 N (0) ~ 86 N (X) として、シャドウタップ付き遅延線 78 (1) の中にロードされる場合もある。上記で前に説明されたように、タップ付き遅延線 78 の目的は、シフトされた入力ベクトルデータサンプル 86 に対して演算するベクトル処理動作 160 の動作の間に、実行ユニット 84 (0) ~ 84 (X) に供給されるべきシフトされた入力ベクトルデータサンプルセット 86 S (0) ~ 86 S (X) に、入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) をシフトすることである。フォーマット変換された入力ベクトルデータサンプルセット 86 F (0) ~ 86 F (X) も、ベクトル処理動作 160 の間にタップ付き遅延線 78 内でシフトされた場合、シフトされフォーマット変換された入力ベクトルデータサンプルセットは、86 S F (0) ~ 86 S F (X) と指定される。

【0125】

[00170]引き続き図 20 を参照すると、実行ユニット 84 (0) ~ 84 (X) は、次に、フォーマット変換された入力ベクトルデータサンプルセット 86 F (0) ~ 86 F (X) を使用して、ベクトル処理動作 160 を実行することができる (ブロック 166)。実行ユニット 84 (0) ~ 84 (X) は、フォーマット変換された入力ベクトルデータサンプルセット 86 F (0) ~ 86 F (X) を使用して、乗算および / または累算を提供するように構成される場合がある。タップ付き遅延線 78 がベクトル処理動作 160 の間にフォーマット変換された入力ベクトルデータサンプルセット 86 F (0) ~ 86 F (X) をシフトするために利用される場合、実行ユニット 84 (0) ~ 84 (X) は、ベクトル処理動作 160 が完了するまで、ベクトル処理動作 160 の各処理ステージの間にシフトされフォーマット変換された入力ベクトルデータサンプルセット 86 S F (0) ~ 86 S F (X) を受信することができる (ブロック 168)。ベクトル処理動作 160 が完了すると、フォーマット変換された入力ベクトルデータサンプルセット 86 F (0) ~ 86 F (X)、またはシフトされフォーマット変換された入力ベクトルデータサンプルセット 86 S F (0) ~ 86 S F (X) を伴うベクトル処理動作に基づく、結果として生じる出力ベクトルデータサンプルセット 172 (0) ~ 172 (X) が、ベクトルデータファイル 82 (0) ~ 82 (X) に供給され記憶されるために、出力データフローパス 98 (0) ~ 98 (X) 内に供給される (ブロック 170)。結果として生じる出力ベクトルデータサンプルセット 172 (0) ~ 172 (X) は、この例では 172 (0)、172 (1)、...、および 172 (X) である、「X + 1」個の、結果として生じる出力ベクトルデータサンプル 172 から構成される。

【0126】

[00171]図 21 は、プライマリタップ付き遅延線 78 (0) からシフトされた入力ベクトルデータサンプルセット 86 S (0) ~ 86 S (X) を受信する例示的なフォーマット変換回路 159 (0) ~ 159 (X) の概略図である。この例では、フォーマット変換回路 159 (0) ~ 159 (X) は、入力データフローパス 80 (0) ~ 80 (X) 内のプライマリタップ付き遅延線 78 (0) の出力上に設けられる。例示的なフォーマット変換回路 159 (0) ~ 159 (X) が次に記載される。

【0127】

[00172]例示的なフォーマット変換回路 159 (0) ~ 159 (X) が次に記載される。フォーマット変換回路 159 (0) の内部構成要素の例示的な詳細が図 21 において提

供されるが、それはフォーマット変換回路 159 (1) ~ 159 (X) にも適用可能である。例として図 21 のフォーマット変換回路 159 (0) を取り上げると、この例におけるフォーマット変換回路 159 (0) は、それぞれ、フォーマット変換された入力ベクトルデータサンプル 86 F (0)、またはシフトされフォーマット変換された入力ベクトルデータサンプル 86 S F (0) を供給するために、ベクトルデータレーン 100 (0) 内のプライマリパイプラインレジスタ 120 (0)、120 (1)、120 (2X+2)、120 (2X+3) からの入力ベクトルデータサンプル 86 (0) またはシフトされた入力ベクトルデータサンプル 86 S (0) のデインターリーブと符号拡張 (sign extension) とを提供するように構成される。この関連で、この例では 4 つのマルチプレクサ 174 (3) ~ 174 (0) が提供され、それらは、それぞれ、割り当てられたプライマリパイプラインレジスタ 120 (0) ~ 120 (2X+3) に従って配置される。各マルチプレクサ 174 (3) ~ 174 (0) は、割り当てられたプライマリパイプラインレジスタ 120 (0)、120 (1)、120 (2X+2)、120 (2X+3) 内のシフトされた入力ベクトルデータサンプル 86 S (0) の部分、または割り当てられたプライマリパイプラインレジスタ 120 (0)、120 (1)、120 (2X+2)、120 (2X+3) に隣接するプライマリパイプラインレジスタ 120 に記憶するシフトされた入力ベクトルデータサンプル 86 S (0) の部分のいずれかを選択するように構成される。

【0128】

[00173]たとえば、プライマリパイプラインレジスタ 120 (0)、120 (1)、120 (2X+2)、120 (2X+3) が、実数 [15:8]、虚数 [15:8]、実数 [7:0]、虚数 [7:0] として、複素数のインターリーブされた形式でインターリーブされシフトされた入力ベクトルデータサンプル 86 S (0) を記憶し、所望のデインターリーブされたフォーマットが、実行されるべきベクトル命令に従う実数 [15:0] および虚数 [15:0] である場合、マルチプレクサ 174 (3) ~ 174 (0) の選択は以下のものであるはずである。マルチプレクサ 174 (3) は、その割り当てられたプライマリパイプラインレジスタ 120 (0) に記憶された、シフトされた入力ベクトルデータサンプル 86 S の部分を選択するはずである。しかしながら、マルチプレクサ 174 (2) は、プライマリパイプラインレジスタ 120 (1) に記憶された、シフトされた入力ベクトルデータサンプル 86 S の部分を選択するはずである。これは、隣接する入力データフローパス 80 (0) (3)、80 (0) (2) 内の入力ベクトルデータサンプル 86 S (0) のデインターリーブされた実数部分 (すなわち、実数 [15:0]) を供給するはずである。同様に、マルチプレクサ 174 (0) は、その割り当てられたプライマリパイプラインレジスタ 120 (2X+3) に記憶された、シフトされた入力ベクトルデータサンプル 86 S の部分を選択するはずである。しかしながら、マルチプレクサ 174 (1) は、プライマリパイプラインレジスタ 120 (2X+2) に記憶された、シフトされた入力ベクトルデータサンプル 86 S の部分を選択するはずである。これは、隣接する入力データフローパス 80 (0) (1)、80 (0) (0) 内のシフトされた入力ベクトルデータサンプル 86 S (0) のデインターリーブされた虚数部分 (すなわち、虚数 [15:0]) を供給するはずである。マルチプレクサ 176 (1)、176 (0) は、図 21 に示されたように、割り当てられていない、隣接しないプライマリパイプラインレジスタ 120 (0)、120 (1)、120 (2X+2)、120 (2X+3) から、シフトされた入力ベクトルデータサンプル 86 S (0) の部分を選択する能力を各マルチプレクサ 174 (3) ~ 174 (0) に提供する。

【0129】

[00174]引き続き図 21 を参照すると、フォーマット変換回路 159 (0) ~ 159 (X) はまた、フォーマット変換された入力ベクトルデータサンプルセット 86 F (0) ~ 86 F (X) を符号拡張する (sign extend) ように構成され得る。たとえば、入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) のフォーマット変換が、小さいビット幅から大きいビット幅に変換された符号付きベクトルデータサンプルを要する場合、フォーマット変換回路 159 (0) ~ 159 (X) は、非負数の場合「0」として、負数の場

10

20

30

40

50

合「F」として最上位ビットを拡張することによって、デインターリーブされたベクトルデータサンプルを符号拡張するように構成され得る。フォーマット変換回路159(0)~159(X)は、フォーマット変換された入力ベクトルデータサンプルセット86F(0)~86F(X)に対して符号拡張が実行されるべきか否かを示すために、実行されているベクトル命令に従って設定される符号拡張(SC)入力178(0)~178(X)を有する場合がある。SC入力178(0)~178(X)は、処理されているベクトル命令に従ってSC入力178(0)~178(X)によって提供されたプログラム可能なデータパス構成に従って符号拡張を実行するために、フォーマット変換回路159(0)~159(X)内に設けられた符号拡張回路180(0)~180(X)に供給され得る。SC入力178(0)~178(X)は、VPE22(3)によるベクトル処理において柔軟性を提供するように、ベクトル命令ごとに構成および再構成され得る。たとえば、フォーマット変換回路159(0)~159(X)内のプログラム可能なデータパスは、必要な場合、実行ユニット84(0)~84(X)を十分に利用して、必要に応じてフォーマット変換を提供するために、ベクトル命令のクロックサイクルごとに、必要な場合クロックサイクルごとに、構成および再構成され得るSC入力178(0)~178(X)によって構成され得る。

【0130】

[00175]しかし、上記で説明されたように、フォーマット変換回路159(0)~159(X)は、プライマリタップ付き遅延線78(0)の一部として設けられる必要がない。プライマリタップ付き遅延線78(0)およびシャドウタップ付き遅延線78(1)はオプションである。フォーマット変換回路159(0)~159(X)は、ベクトルデータファイル82(0)~82(X)から直接入力ベクトルデータサンプルセット86(0)~86(X)を受信する可能性がある。このシナリオでは、例として、図21を参照すると、入力ベクトルデータサンプルセット86(0)~86(X)は、直接ベクトルレジスタファイル82(0)~82(X)からプライマリレジスタ120(0)~120(4X+3)の中にロードされる可能性がある。

【0131】

[00176]さらに、フォーマット変換回路159(0)~159(X)は、フォーマット変換された入力ベクトルデータサンプルセット86(0)~86(X)へのプライマリタップ付き遅延線78(0)の出力上に設けられるが、それは必要でないことに留意されたい。図21のフォーマット変換回路159(0)~159(X)は、プライマリタップ付き遅延線78(0)およびシャドウタップ付き遅延線78(1)の入力側に設けられる可能性があり、その結果、ベクトルデータファイル82(0)~82(X)からフェッチされた入力ベクトルデータサンプルセット86(0)~86(X)は、プライマリタップ付き遅延線78(0)およびシャドウタップ付き遅延線78(1)の中にロードされるより前に、フォーマット変換回路159(0)~159(X)内でフォーマット変換される。この例では、入力ベクトルデータサンプルセット86(0)~86(X)は、プライマリタップ付き遅延線78(0)およびシャドウタップ付き遅延線78(1)において、フォーマット変換された入力ベクトルデータサンプルセット86F(0)~86F(X)（またはシフト後の86SF(0)~86SF(X)）として記憶されるはずである。フォーマット変換された入力ベクトルデータサンプルセット86F(0)~86F(X)（またはシフト後の86SF(0)~86SF(X)）は、次いで、ベクトル処理動作における実行のために、直接プライマリタップ付き遅延線78(0)から直接実行ユニット84(0)~84(X)に供給される可能性がある。

【0132】

[00177]上記で説明されたように、入力データフローパス80(0)~80(X)は、実行されるべきベクトル命令に従ってフォーマット変換回路159(0)~159(X)を利用するように、プログラム可能な入力データパス構成に従ってプログラムされ得る。この関連で、図22は、図19のVPE22(3)における入力ベクトルデータサンプルセット86(0)~86(X)のシフトおよびフォーマット変換のプログラミングを制御

10

20

30

40

50

するベクトル命令のビットの例示的なデータフォーマットを提供するチャート 182 である。チャート 182 内のフィールドに提供されたデータは、それらの機能が処理されるべきベクトル命令に必要とされるかどうかに応じて、フォーマット変換回路 159 (0) ~ 159 (X) および / または タップ付き遅延線 78 が入力データフローパス 80 (0) ~ 80 (X) に含まれるかどうかを制御するように、VPE 22 (3) にプログラミングを提供する。

【0133】

[00178] 図 22 では、たとえば、タップ付き遅延線 78 によって符号付き複素数 16 ビットフォーマット (SC16) を使用するとき、算術命令のためのシフトバイアスが提供されるかどうかを示すために、ベクトル命令またはベクトルプログラミングのビット [7 : 0] にバイアスフィールド 184 (BIAS__SC16) が設けられる。第 1 のソースデータ (すなわち、入力ベクトルデータサンプルセット 86 (0) ~ 86 (X)) が縮小化 (すなわち、デインターリーブ) され、SC8 フォーマットから SC16 フォーマットに変換されるべきか否かを示すために、ベクトル命令またはベクトルプログラミングのビット [16] に第 1 のソースデータフォーマット変換フィールド 186 (DECIMATE__SRC1) が設けられる。第 2 のソースデータ (すなわち、入力ベクトルデータサンプルセット 86 (0) ~ 86 (X)) が縮小化 (すなわち、デインターリーブ) され、SC8 フォーマットから SC16 フォーマットに変換されるべきか否かを示すために、ベクトル命令またはベクトルプログラミングのビット [17] に第 2 のソースデータフォーマット変換フィールド 188 (DECIMATE__SRC2) が設けられる。出力ソースデータ (たとえば、図 19 の VPE 22 (3) 内の、結果として生じる出力ベクトルデータサンプルセット 172 (0) ~ 172 (X)) が、ベクトルデータファイル 82 (0) ~ 82 (X) に記憶されるとき、SC16 フォーマットで記憶されるべきか、または SC16 フォーマットから SC8 フォーマットに変換され並び替えられるべきかを示すために、ビット [18] に出力データフォーマットフィールド 190 (DEST__FMT) が設けられる。上記および図 17B に前述されたように、特に CDMA 固有のベクトル処理動作に有用であり得る、偶数 (たとえば、オンタイム) サンプルおよび奇数 (たとえば、後発) サンプルに沿って、入力ソースデータ (すなわち、入力ベクトルデータサンプルセット 86 (0) ~ 86 (X)) および出力データ (たとえば、図 19 の VPE 22 (3) 内の、結果として生じる出力ベクトルデータサンプルセット 172 (0) ~ 172 (X)) が、縮小化 (すなわち、デインターリーブ) されるべきかどうかを示すために、ビット [19] にフェーズフォーマットフィールド 192 (DECIMATE__PHASE) が設けられる。

【0134】

[00179] 上記で説明されたように、VPE 22 内の実行ユニット 84 (0) ~ 84 (X) が入力ベクトルデータサンプルに対してベクトル処理を実行し、結果として出力データフローパス 98 (0) ~ 98 (X) 上に、結果として生じる出力ベクトルデータサンプルセットを供給した後、次のベクトル処理動作は、結果として生じる出力ベクトルデータサンプルセットに対して実行される必要があり得る。しかしながら、結果として生じる出力ベクトルデータサンプルセットは、次のベクトル処理動作のために並び替えられる必要があり得る。したがって、前の処理動作から得られた、結果として生じる出力ベクトルデータサンプルセットは、ベクトルデータファイル 82 (0) ~ 82 (X) に記憶され、並び替えのためにフェッチされ、ベクトルデータファイル 82 (0) ~ 82 (X) に並び替えられたフォーマットで再記憶されなければならない。たとえば、図 17A および図 17B において上記で説明されたように、次の処理動作は、ベクトルデータファイル 82 (0) ~ 82 (X) に記憶されるときに、前に処理されたベクトルデータサンプルがインターリーブされることを必要とする場合がある。

【0135】

[00180] 別の例として、次の処理動作は、ベクトルデータファイル 82 (0) ~ 82 (X) に記憶されるときに、前に処理されたベクトルデータサンプルがデインターリーブさ

10

20

30

40

50

れることを必要とする場合がある。たとえば、CDMA処理動作では、信号を表すデータサンプルは、信号の偶数（たとえば、オンタイム）フェーズおよび奇数（たとえば、後発）フェーズに従って記憶されインターリーブされる必要があり得る。この問題を解決するために、ベクトルプロセッサは、出力ベクトルデータがベクトルデータメモリに記憶された後に、実行ユニットからの出力ベクトルデータの後処理並び替えを実行する回路を含むことができる。ベクトルデータメモリに記憶された、後処理された出力ベクトルデータサンプルは、ベクトルデータメモリからフェッチされ、並び替えられ、ベクトルデータメモリに戻されて記憶される。この後処理は、実行ユニットによる並び替えられたベクトルデータサンプルの次の処理を遅延させ、実行ユニット内のコンピュータ構成要素が過少利用される原因になる。

10

【0136】

[00181]この関連で、図23は、図2のVPE22として提供され得る別の例示的なVPE22(4)の概略図である。下記でより詳細に記載されるように、図23のVPE22(4)は、ベクトルデータサンプルの再フェッチが除去または低減され、電力消費が低減される、VPE22(4)内のベクトルデータファイル82(0)~82(X)に記憶されるべき、ベクトル処理動作のために実行ユニット84(0)~84(X)によって供給される、結果として生じる出力ベクトルデータサンプルセット194(0)~194(X)のインフライト並び替えを提供するように構成される。結果として生じる出力ベクトルデータサンプルセット194(0)~194(X)は、この例では194(0)、194(1)、...、および194(X)である、「X+1」個の、結果として生じる出力ベクトルデータサンプル194から構成される。たとえば、並び替えは、ベクトルデータファイル82(0)~82(X)に記憶される前の、結果として生じる出力ベクトルデータサンプルセット194(0)~194(X)のインターリーブを含む可能性がある。

20

【0137】

[00182]図23に示され、下記でより詳細に説明されるように、並び替え回路196(0)~196(X)は、ベクトルデータレーン100(0)~100(X)の各々の中の実行ユニット84(0)~84(X)とベクトルデータファイル82(0)~82(X)との間の出力データフローパス98(0)~98(X)内に設けられる。並び替え回路196(0)~196(X)は、出力データフローパス98(0)~98(X)内の並び替えられた、結果として生じる出力ベクトルデータサンプルセット194R(0)~194R(X)として、結果として生じる出力ベクトルデータサンプルセット194(0)~194(X)の並び替えを提供するために、実行されるべきベクトル命令に従うプログラミングに基づいて構成される。図23のVPE22(4)における、結果として生じる出力ベクトルデータサンプルセット194(0)~194(X)のインフライト並び替えは、実行ユニット84(0)~84(X)によって供給された、結果として生じる出力ベクトルデータサンプルセット194(0)~194(X)が、ベクトルデータファイル82(0)~82(X)に記憶される前に、並び替えられた、結果として生じる出力ベクトルデータサンプルセット194R(0)~194R(X)として並び替えられることを意味する。このようにして、結果として生じる出力ベクトルデータサンプルセット194(0)~194(X)は、並び替えられた、結果として生じる出力ベクトルデータサンプルセット194R(0)~194R(X)として、並び替えられたフォーマットでベクトルデータファイル82(0)~82(X)に記憶される。非限定的な例として、結果として生じる出力ベクトルデータサンプルセット194(0)~194(X)の並び替えは、ベクトルデータファイル82(0)~82(X)に並び替えられた、結果として生じる出力ベクトルデータサンプルセット194R(0)~194R(X)として記憶されるべき、結果として生じる出力ベクトルデータサンプルセット194(0)~194(X)のインターリーブまたはデインターリーブを含む場合がある。

30

40

【0138】

[00183]このように、出力データフローパス98(0)~98(X)内に設けられた並

50

び替え回路 196(0) ~ 196(X) により、結果として生じる出力ベクトルデータサンプルセット 194(0) ~ 194(X) は、最初にベクトルデータファイル 82(0) ~ 82(X) に記憶され、次いでベクトルデータファイル 82(0) ~ 82(X) からフェッチされ、並び替えられ、ベクトルデータファイル 82(0) ~ 82(X) に再記憶される必要がない。結果として生じる出力ベクトルデータサンプルセット 194(0) ~ 194(X) は、ベクトルデータファイル 82(0) ~ 82(X) に記憶される前に並び替えられる。このようにして、結果として生じる出力ベクトルデータサンプルセット 194(0) ~ 194(X) は、実行ユニット 84(0) ~ 84(X) 内で実行されるべき次のベクトル処理動作を遅延させる可能性がある、さらなる後処理ステップを必要とせず、ベクトルデータファイル 82(0) ~ 82(X) に並び替えられたフォーマットで記憶される。したがって、VPE22(4) 内のデータフローパスの効率性は、結果として生じる出力ベクトルデータサンプルセット 194(0) ~ 194(X) の並び替えによって制限されない。実行ユニット 84(0) ~ 84(X) における次のベクトル処理は、結果として生じる出力ベクトルデータサンプルセット 194(0) ~ 194(X) がベクトルデータファイル 82(0) ~ 82(X) に並び替えられた、結果として生じる出力ベクトルデータサンプルセット 194R(0) ~ 194R(X) として並び替えられたフォーマットで記憶されるべきときに、データフローの制限ではなく、コンピュータリソースのみによって制限される。

【0139】

[00184] この例では、図 23 に示されたように、並び替え回路 196(0) ~ 196(X) を含む VPE22(4) はまた、プライマリタップ付き遅延線 78(0) および / またはシャドウタップ付き遅延線 78(1) をオプションとして含むことができる。タップ付き遅延線 78(0)、78(1) の動作は、VPE22(1) および VPE22(2) に関して上記で前述された。上記で前に説明されたように、タップ付き遅延線 78(0)、78(1) は、実行ユニット 84(0) ~ 84(X) に供給されるべきシフトされた入力ベクトルデータサンプルセット 86S(0) ~ 86S(X) を必要とするベクトル処理動作に利用される場合がある。同様に、図 4、図 11、および図 19 の VPE22(1) ~ 22(3) 内に設けられた共通構成要素が、図 23 の VPE22(4) 内に設けられることに留意されたい。共通構成要素は、共通要素番号とともに図 23 の VPE22(4) において示される。VPE22(1) ~ 22(3) に関する上記これらの共通構成要素の前の記載および説明は、図 23 の VPE22(4) にも適用可能であり、したがってここでは再び記載されない。

【0140】

[00185] 引き続き図 23 を参照すると、より具体的には、並び替え回路 196(0) ~ 196(X) は、出力データフローパス 98(0) ~ 98(X) 上の並び替え回路入力 198(0) ~ 198(X) 上で、結果として生じる出力ベクトルデータサンプルセット 194(0) ~ 194(X) を受信するように構成される。並び替え回路 196(0) ~ 196(X) は、並び替えられた、結果として生じる出力ベクトルデータサンプルセット 194R(0) ~ 194R(X) を供給するために、結果として生じる出力ベクトルデータサンプルセット 194(0) ~ 194(X) を並び替えるように構成される。並び替え回路 196(0) ~ 196(X) は、記憶用にベクトルデータファイル 82(0) ~ 82(X) に供給されるために、出力データフローパス 98(0) ~ 98(X) 内の並び替え回路出力 200(0) ~ 200(X) 上に並び替えられた、結果として生じる出力ベクトルデータサンプルセット 194R(0) ~ 194R(X) を供給するように構成される。

【0141】

[00186] この実施形態における出力データフローパス 98(0) ~ 98(X) 内のベクトルデータファイル 82(0) ~ 82(X) に並び替えられた、結果として生じる出力ベクトルデータサンプルセット 194R(0) ~ 194R(X) を供給するための図 23 の VPE22(4) のさらなる詳細および特徴のさらなる説明が次に記載される。この関連で、図 24 は、結果として生じる出力ベクトルデータサンプルセット 194(0) ~ 19

4 (X) の並び替えを必要とする例示的なベクトル命令に従って、並び替え回路 1 9 6 (0) ~ 1 9 6 (X) を利用する図 2 3 の V P E 2 2 (4) において実行され得るベクトル処理動作 2 0 2 から得られた、結果として生じる出力ベクトルデータサンプルセット 1 9 4 (0) ~ 1 9 4 (X) の例示的な並び替えを示すフローチャートである。

【 0 1 4 2 】

[00187] 図 2 3 と図 2 4 とを参照すると、ベクトル命令に従うベクトル処理動作 2 0 2 に従って処理されるべき入力ベクトルデータサンプルセット 8 6 (0) ~ 8 6 (X) が、ベクトルデータファイル 8 2 (0) ~ 8 2 (X) からフェッチされ、入力データフローパス 8 0 (0) ~ 8 0 (X) 内に供給される (図 2 4 のブロック 2 0 4) 。たとえば、ベクトル処理動作 2 0 2 は、実行されるべきベクトル命令に従って必要とされる任意のベクトル処理動作を含むことができる。上述のフィルタ、相関、およびフォーマット変換のベクトル処理動作を含む非限定的な例。ベクトル処理動作 2 0 2 のための入力ベクトルデータサンプルセット 8 6 (0) ~ 8 6 (X) の幅に応じて、ベクトル命令のプログラミングに従うベクトル処理動作 2 0 2 を提供するために、図 2 3 の V P E 2 2 (4) 内のベクトルデータレーン 1 0 0 (0) ~ 1 0 0 (X) の 1 つ、いくつか、またはすべてが利用され得る。ベクトルデータファイル 8 2 (0) ~ 8 2 (X) の幅全体が必要な場合、すべてのベクトルデータレーン 1 0 0 (0) ~ 1 0 0 (X) がベクトル処理動作 2 0 2 に利用され得る。ベクトル処理動作 2 0 2 は、ベクトルデータレーン 1 0 0 (0) ~ 1 0 0 (X) のサブセットを必要とするにすぎない場合がある。これは、入力ベクトルデータサンプルセット 8 6 (0) ~ 8 6 (X) の幅がすべてのベクトルデータファイル 8 2 (0) ~ 8 2 (X) の幅よりも小さいからであり得るし、ここで、ベクトル処理動作 2 0 2 と並列に実行されるべき他のベクトル処理動作にさらなるベクトルデータレーン 1 0 0 を利用することが望ましい。

【 0 1 4 3 】

[00188] 引き続き図 2 3 と図 2 4 とを参照すると、フェッチされた入力ベクトルデータサンプルセット 8 6 (0) ~ 8 6 (X) が、実行ユニット 8 4 (0) ~ 8 4 (X) にある入力データフローパス 8 0 (0) ~ 8 0 (X) から受信される (図 2 4 のブロック 2 0 6) 。実行ユニット 8 4 (0) ~ 8 4 (X) が、ベクトル命令に従って提供されたベクトル処理動作 2 0 2 に従って、受信された入力ベクトルデータサンプルセット 8 6 (0) ~ 8 6 (X) に対してベクトル処理を実行する (図 2 4 のブロック 2 0 8) 。非限定的な例として、入力ベクトルデータサンプルセット 8 6 (0) ~ 8 6 (X) は、場合によっては、入力ベクトルデータサンプルセット 8 6 (0) ~ 8 6 (X) のシフトを伴う実行ユニット 8 4 (0) ~ 8 4 (X) によって実行されるベクトル処理動作 2 0 2 の各処理ステージの間のベクトル処理動作 2 0 2 の実行中にシフトされるべき入力ベクトルデータサンプルセット 8 6 (0) ~ 8 6 (X) として、プライマリタップ付き遅延線 7 8 (0) の中にロードされる場合がある。前に説明されたように、次の入力ベクトルデータサンプルセット 8 6 N (0) ~ 8 6 N (X) は、場合によっては、実行ユニット 8 4 (1) ~ 8 4 (X) によって処理されるべき次の入力ベクトルデータサンプルセット 8 6 N (0) ~ 8 6 N (X) として、シャドウタップ付き遅延線 7 8 (1) の中にロードされる場合もある。上記で前に説明されたように、タップ付き遅延線 7 8 の目的は、シフトされた入力ベクトルデータサンプル 8 6 に対して演算するベクトル処理動作 2 0 2 の動作の間に、実行ユニット 8 4 (0) ~ 8 4 (X) に供給されるべきシフトされた入力ベクトルデータサンプルセット 8 6 S (0) ~ 8 6 S (X) に、入力ベクトルデータサンプルセット 8 6 (0) ~ 8 6 (X) をシフトすることである。

【 0 1 4 4 】

[00189] 引き続き図 2 3 と図 2 4 とを参照すると、実行ユニット 8 4 (0) ~ 8 4 (X) は、入力ベクトルデータサンプルセット 8 6 (0) ~ 8 6 (X) を使用して、乗算および/または累算を提供するように構成される場合がある。タップ付き遅延線 7 8 がベクトル処理動作 2 0 2 の間にフォーマット変換された入力ベクトルデータサンプルセット 8 6 F (0) ~ 8 6 F (X) をシフトするために利用される場合、実行ユニット 8 4 (0) ~

84(X)は、例によって前述されたように、ベクトル処理動作202が完了するまで、ベクトル処理動作202の各処理ステージの間にシフトされた入力ベクトルデータサンプルセット86S(0)~86S(X)を受信することができる。ベクトル処理動作202が完了すると、入力ベクトルデータサンプルセット86(0)~86(X)、またはシフトされフォーマット変換された入力ベクトルデータサンプルセット86S(0)~86S(X)のベクトル処理に基づく、結果として生じる出力ベクトルデータサンプルセット194(0)~194(X)が、出力データフローパス98(0)~98(X)内に供給される。

【0145】

[00190]引き続き図23と図24とを参照すると、結果として生じる出力ベクトルデータサンプルセット194(0)~194(X)がベクトルデータファイル82(0)~82(X)に記憶される前に、結果として生じる出力ベクトルデータサンプルセット194(0)~194(X)は、実行ユニット84(0)~84(X)とベクトルデータファイル82(0)~82(X)との間に設けられた出力データフローパス98(0)~98(X)内に設けられた並び替え回路196(0)~196(X)に供給される。並び替え回路196(0)~196(X)は、実行されているベクトル命令に従って、および下記でより詳細に説明されるように、ベクトル命令がベクトルデータファイル82(0)~82(X)に記憶されるべき、結果として生じる出力ベクトルデータサンプルセット194(0)~194(X)の並び替えを要求する場合、出力データフローパス98(0)~98(X)に含まれるようにプログラム可能である。結果として生じる出力ベクトルデータサンプルセット194(0)~194(X)がベクトルデータファイル82(0)~82(X)に記憶されることなく、並び替え回路196(0)~196(X)が、実行されているベクトル命令に従うプログラミングにおいて提供される並び替えに従って、結果として生じる出力ベクトルデータサンプルセット194(0)~194(X)を並び替える(図24のブロック210)。このようにして、結果として生じる出力ベクトルデータサンプルセット194(0)~194(X)は、それにより実行ユニット84(0)~84(X)において遅延をもたらす、最初にベクトルデータファイル82(0)~82(X)に記憶され、再フェッチされ、後処理動作において並び替えられ、ベクトルデータファイル82(0)~82(X)に並び替えられたフォーマットで記憶される必要がない。結果として生じる出力ベクトルデータサンプルセット194(0)~194(X)が、並び替え後処理を必要とせずに、ベクトルデータファイル82(0)~82(X)に並び替えられた、結果として生じる出力ベクトルデータサンプルセット194R(0)~194R(X)として記憶される(図24のブロック212)。たとえば、結果として生じる出力ベクトルデータサンプルセット194(0)~194(X)は、並び替え回路196(0)~196(X)によって並び替えられる前に、図18Aおよび図18Bにおいて提供されたフォーマットのようなフォーマットで現れる場合がある。

【0146】

[00191]次に図25に関して、並び替え回路196(0)~196(X)の一例が記載される。ベクトルデータレーン100(0)イズプロバイデッド内に設けられた並び替え回路196(0)の1つの事例のために、並び替え回路196(0)~196(X)の内部構成要素の例示的な詳細が図25において提供されるが、それは並び替え回路196(1)~196(X)にも適用可能である。例として図25における並び替え回路196(0)を取り上げると、この例における並び替え回路196(0)は、並び替えられた、結果として生じる出力ベクトルデータサンプル194R(0)を供給するために、ベクトルデータレーン100(0)内の出力データフローパス98(0)内で、実行ユニット84(0)によって供給された、結果として生じる出力ベクトルデータサンプル194(0)を並び替えるように構成される。この関連で、この例ではマルチプレクサの形態で設けられた4つの出力ベクトルデータサンプル選択器214(3)~214(0)がこの例において提供され、それらは、各々8ビット幅のこの例では4つの96(0)(3)~96(0)(0)である、実行ユニット出力96(0)のビット幅に従って配置される。各出力

10

20

30

40

50

ベクトルデータサンプル選択器 2 1 4 (3) ~ 2 1 4 (0) は、割り当てられた実行ユニット出力 9 6 (0) (3) ~ 9 6 (0) (0) 内の、結果として生じる出力ベクトルデータサンプル 1 9 4 (0) の部分、または割り当てられた実行ユニット出力 9 6 (0) (3) ~ 9 6 (0) (0) に隣接する実行ユニット出力 9 6 からの、結果として生じるシフト出力ベクトルデータサンプル 1 9 4 (0) の部分のいずれかを選択するように構成される。

【 0 1 4 7 】

[00192]たとえば、実行ユニット出力 9 6 (0) (3) ~ 9 6 (0) (0) が、16ビット符号付き複素数フォーマット、実数 [3 1 : 2 4]、実数 [2 3 : 1 6]、虚数 [1 5 : 8]、虚数 [7 : 0] で、結果として生じる出力ベクトルデータサンプル 1 9 4 (0) を供給し、所望の並び替えられた（たとえば、インターリーブされた）フォーマットが、実行されるべきベクトル命令に従って実数 [3 1 : 2 4]、虚数 [2 3 : 1 6]、実数 [1 5 : 8]、虚数 [7 : 0] である場合、出力ベクトルデータサンプル選択器 2 1 4 (3) ~ 2 1 4 (0) の選択は以下のものであるはずである。出力ベクトルデータサンプル選択器 2 1 4 (3) は、出力データフローパス 9 8 (0) (3) 上で供給するために、実行ユニット出力 9 6 (0) (3) から、結果として生じる出力ベクトルデータサンプル 1 9 4 (0) (3) を選択するはずである。しかしながら、出力ベクトルデータサンプル選択器 2 1 4 (2) は、出力データフローパス 9 8 (0) (2) 上で供給するために、実行ユニット出力 9 6 (0) (1) 上の、結果として生じる出力ベクトルデータサンプル 1 9 4 (0) (1) の部分を選択するはずである。これにより、並び替えられた、結果として生じる出力ベクトルデータサンプル 1 9 4 R (0) の並び替えられた、結果として生じる出力ベクトルデータサンプル 1 9 4 R (0) (3)、1 9 4 R (0) (2) として、隣接する出力データフローパス 9 8 (0) (3)、9 8 (0) (2) 内の、結果として生じるシフト出力ベクトルデータサンプル 1 9 4 (0)（すなわち、実数 [3 1 : 2 4]、虚数 [2 3 : 1 6]）のインターリーブされた実数部分がもたらされるはずである。同様に、出力ベクトルデータサンプル選択器 2 1 4 (0) は、出力データフローパス 9 8 (0) (0) 内で供給するために、実行ユニット出力 9 6 (0) (0) から、結果として生じる出力ベクトルデータサンプル 1 9 4 (0) (0) を選択するはずである。しかしながら、出力ベクトルデータサンプル選択器 2 1 4 (1) は、出力データフローパス 9 8 (0) (1) 上で供給するために、実行ユニット出力 9 6 (0) (2) 上の、結果として生じる出力ベクトルデータサンプル 1 9 4 (0) (2) を選択するはずである。これにより、並び替えられた、結果として生じる出力ベクトルデータサンプル 1 9 4 R (0) の並び替えられた、結果として生じる出力ベクトルデータサンプル 1 9 4 R (0) (1)、1 9 4 R (0) (0) として、隣接する出力データフローパス 9 8 (0) (1)、9 8 (0) (0) 内で並び替えられ、インターリーブされた、結果として生じる出力ベクトルデータサンプル 1 9 4 (0) (2)、1 9 4 (0) (0)（すなわち、実数 [1 5 : 8]、虚数 [7 : 0]）がもたらされるはずである。同様にマルチプレクサの形態で設けられた出力ベクトルデータサンプル選択器 2 1 6 (1)、2 1 6 (0) は、図 2 5 に示されたように、割り当てられていない、隣接しない実行ユニット出力 9 6 (0) (3) ~ 9 6 (0) (0) からの、結果として生じる出力ベクトルデータサンプル 1 9 4 (0) (3) ~ 1 9 4 (0) (0) の間を選択する能力を提供する。

【 0 1 4 8 】

[00193]引き続き図 2 3 と図 2 5 とを参照すると、並び替え回路 1 9 6 (0) ~ 1 9 6 (X) は、実行されるべきベクトル命令に従って、結果として生じる出力ベクトルデータサンプルセット 1 9 4 (0) ~ 1 9 4 (X) を並び替えないように構成または再構成されるようにプログラム可能であるものとして提供される可能性がある。この例では、並び替え回路 1 9 6 (0) ~ 1 9 6 (X) は、形成されたいかなる並び替え動作もなしに、並び替え回路 1 9 6 (0) ~ 1 9 6 (X) に直接流れる出力データフローパス 9 8 (0) ~ 9 8 (X) を提供するようにプログラムされる場合がある。上記で前に説明され、図 2 2 に示されたように、出力ソースデータ（たとえば、図 2 3 の V P E 2 2 (4) 内の、結果と

10

20

30

40

50

して生じる出力ベクトルデータサンプルセット 194(0) ~ 194(X)) が、ベクトルデータファイル 82(0) ~ 82(X) に記憶されるとき、SC16フォーマットで記憶されるべきか、またはSC16フォーマットからSC8フォーマットに変換され並び替えられるべきかを示すために、非限定的な例として、ベクトル命令のビット[18]にチャート182内の出力データフォーマットフィールド190(DEST_FMT)が設けられ得る。

【0149】

[00194]この関連で、図25のプログラム可能な並び替えデータパス構成入力218(0)は、出力データフローパス98(0)内の、結果として生じる出力ベクトルデータサンプル194(0)(3) ~ 194(0)(0)を並び替えるか、または並び替えないように、並び替え回路196(0)をプログラムするために、並び替え回路196(0)に供給され得る。プログラム可能な並び替えデータパス構成入力218(1) ~ 218(X)(図示せず)は、それぞれ、出力データフローパス98(1) ~ 98(X)内の、結果として生じる出力ベクトルデータサンプルセット194(1) ~ 194(X)を並び替えるか、または並び替えないように、並び替え回路196(1) ~ 196(X)をプログラムするために、並び替え回路196(1) ~ 196(X)に同様に供給され得る。このようにして、並び替え回路196(0) ~ 196(X)は、ベクトル命令が実行されるべきそのような処理を提供しない場合、結果として生じる出力ベクトルデータサンプルセット194(0) ~ 194(X)を並び替えないようにプログラムされ得る。プログラム可能な並び替えデータパス構成入力218(0) ~ 218(X)は、VPE22(4)によるベクトル処理において柔軟性を提供するように、ベクトル命令ごとに構成および再構成され得る。たとえば、プログラム可能な並び替えデータパス構成入力218(0) ~ 218(X)は、必要な場合実行ユニット84(0) ~ 84(X)を十分に利用して、必要に応じて並び替えを提供するように、ベクトル命令のクロックサイクルごとに、必要な場合クロックサイクルごとに、構成および再構成され得る。

【0150】

[00195]実行ユニット84(0) ~ 84(X)において実行されるべき次のベクトル処理動作を遅延させる可能性がある、さらなる後処理ステップを必要とせずに、実行ユニット84(0) ~ 84(X)からの、結果として生じる出力ベクトルデータサンプルセットのインフライト処理を伴う、他のベクトル処理動作も提供され得る。たとえば、可変長の拡散信号データシーケンスに従ってチップシーケンスの逆拡散を必要とするCDMAワイヤレスベースバンド動作が、インフライトベクトル処理から恩恵を受ける場合がある。

【0151】

[00196]たとえば、CDMAを使用して変調され得るデータ信号220が図26Aにおいて示される。データ信号220は2Tの周期を有する。図26Aに示されたように、データ信号220は、この例ではデータシーケンス1010を表し、ここで、高信号レベルは論理「1」を表し、低信号レベルは論理「0」を表す。CDMA変調において、データ信号220は、図26Bのチップシーケンス222などのチップシーケンス222によって拡張され、それは擬似ランダムコードであり得る。この例では、チップシーケンス222は、この例ではデータ信号220のサンプルごとに10チップの拡散率または拡散係数を有するチップシーケンス222を提供するために、データ信号220の周期の10分の1の大きさである周期を有する。この例では、データ信号220を拡散するために、データ信号220は、図26Cに示されたように、拡散送信データ信号224を供給するために、チップシーケンス222と排他的論理和(すなわち、XOR)される。拡散送信データ信号224とともに同じ帯域幅で送信される他のユーザ向けの他のデータ信号は、互いに直交する他のチップシーケンスとチップシーケンス222とを用いて拡散される。このようにして、元のデータ信号220が復元されるべきとき、拡散送信データ信号224は、図11 ~ 図16に関して上記で前述されたように、シーケンス番号と関連される。チップシーケンス222の場合のように、シーケンス番号と拡散送信データ信号224との間に高相関が存在する場合、元のデータ信号220は、高相関シーケンス番号に関連するチ

10

20

30

40

50

チップシーケンスを使用して復元され得る。拡散送信データ信号 224 は、図 26D における復元されたデータ信号 226 のように、元のデータ信号 220 を復元するために、この例ではチップシーケンス 222 である高相関チップシーケンスを用いて逆拡散される。

【0152】

[00197] 図 26C における拡散送信データ信号 224 の逆拡散は、高相関チップシーケンスを決定するために、図 11 の VPE 22(2) に関して上述された相関ベクトル処理動作と同様に、拡散送信データ信号 224 と潜在的なチップシーケンスとの間の内積として、逆拡散ベクトル処理動作において実行され得る。拡散送信データ信号 224 は、図 26D における復元されたデータ信号 226 を供給するために、元のデータ信号 220 を CDMA 変調するために使用されていると決定されたチップシーケンス 222 を用いて逆拡散され得る。

10

【0153】

[00198] CDMA 処理動作を含むベクトルプロセッサでは、ベクトルプロセッサは、実行ユニットから出力され、ベクトルデータメモリに記憶された後に拡散信号ベクトルデータシーケンスの逆拡散を実行する回路を含むことができる。この関連で、ベクトルデータメモリに記憶された拡散信号ベクトルデータシーケンスは、後処理動作においてベクトルデータメモリからフェッチされ、元のデータ信号を復元するために相関拡散コードシーケンスまたはチップシーケンスを用いて逆拡散される。拡散前の元のデータサンプルである逆拡散ベクトルデータシーケンスは、ベクトルデータメモリに戻されて記憶される。この後処理動作は、実行ユニットによる次のベクトル動作処理を遅延させる可能性があり、実行ユニット内のコンピュータ構成要素が過少利用される原因になる。さらに、逆拡散されるべき拡散信号ベクトルデータシーケンスは実行ユニットからの異なるデータフローパスと交差するので、拡散コードシーケンスを使用する拡散信号ベクトルシーケンスの逆拡散は、並列化することが困難である。

20

【0154】

[00199] この問題に対処するために、下記で開示される実施形態では、VPE 内の実行ユニットとベクトルデータメモリとの間のデータフローパス内に設けられた逆拡散回路を含む VPE が提供される。逆拡散回路は、出力ベクトルデータサンプルセットが実行ユニットからベクトルデータメモリに出力データフローパスを介して供給されている間のインフライトの実行ユニットからの出力ベクトルデータサンプルを使用して、拡散スペクトルシーケンスを逆拡散するように構成される。出力ベクトルデータサンプルセットのインフライト逆拡散は、実行ユニットによって供給された出力ベクトルデータサンプルセットが、ベクトルデータメモリに記憶される前に逆拡散されることを意味し、その結果、出力ベクトルデータサンプルセットは逆拡散されたフォーマットでベクトルデータメモリに記憶される。逆拡散された拡散スペクトルシーケンス(DSSS)は、実行ユニット内で実行されるべき次のベクトル処理動作を遅延させる可能性がある、さらなる後処理ステップを必要とせずに、ベクトルデータメモリに逆拡散された形式で記憶され得る。したがって、VPE 内のデータフローパスの効率、拡散スペクトルシーケンスの逆拡散によって制限されない場合がある。逆拡散された拡散スペクトルシーケンスがベクトルデータメモリに記憶されるとき、実行ユニット内の次のベクトル処理は、データフローの制限ではなく、コンピュータリソースのみによって制限される。

30

40

【0155】

[00200] この関連で、図 27 は、図 2 の VPE 22 として提供され得る別の例示的な VPE 22(5) の概略図である。下記でより詳細に記載されるように、図 27 の VPE 22(5) は、ベクトルデータサンプルの再フェッチが除去または低減され、電力消費が低減される、VPE 22(5) 内のベクトルデータファイル 82(0) ~ 82(X) に記憶されるべき、ベクトル処理動作のためのコードシーケンスを用いて実行ユニット 84(0) ~ 84(X) によって供給される、結果として生じる出力ベクトルデータサンプルセット 228(0) ~ 228(X) のインフライト逆拡散を提供するように構成される。結果として生じる出力ベクトルデータサンプルセット 228(0) ~ 228(X) は、この例

50

では228(0)、228(1)、...、および228(X)である、「X+1」個の入力の結果として生じる出力ベクトルデータサンプル228から構成される。コードシーケンスは、非限定的な例として、CDMA逆拡散ベクトル処理動作のための拡散スペクトルCDMAチップシーケンスであり得る。図27のVPE22(5)では、結果として生じる出力ベクトルデータサンプルセット228(0)~228(X)は、ベクトルデータファイル82(0)~82(X)に記憶される前にコードシーケンスを用いて逆拡散され得る。

【0156】

[00201]図27に示され、下記でより詳細に説明されるように、逆拡散回路230は、ベクトルデータレーン100(0)~100(X)の各々の中の実行ユニット84(0)~84(X)とベクトルデータファイル82(0)~82(X)との間の出力データフローパス98(0)~98(X)内に設けられる。逆拡散回路230は、相関ベクトル処理動作に関して図11~図16において上記で前述されたように、シーケンス番号発生器134によって生成された基準ベクトルデータサンプルセット130(0)~130(X)として供給されるコードシーケンスを用いて、結果として生じる出力ベクトルデータサンプルセット228(0)~228(X)のインフライト逆拡散を提供するために、実行されるべきベクトル命令に従うプログラミングに基づいて構成される。逆拡散された、結果として生じる出力ベクトルデータサンプルセット229(0)~229(Z)は、出力データフローパス98(0)~98(X)内の逆拡散回路230によって供給される。逆拡散された、結果として生じる出力ベクトルデータサンプルセット229(0)~229(Z)は、この例では229(0)、229(1)、...、および229(Z)である、「Z+1」個の逆拡散された、結果として生じる出力ベクトルデータサンプル229から構成される。図27のVPE22(5)における、結果として生じる出力ベクトルデータサンプルセット228(0)~228(X)のインフライト逆拡散は、実行ユニット84(0)~84(X)によって供給された、結果として生じる出力ベクトルデータサンプルセット228(0)~228(X)が、ベクトルデータファイル82(0)~82(X)に記憶される前に、結果として生じるベクトルデータサンプルセット228(0)~228(X)内でコードシーケンスを用いて逆拡散されることを意味する。このようにして、結果として生じる出力ベクトルデータサンプルセット228(0)~228(X)は、逆拡散された、結果として生じる出力ベクトルデータサンプルセット229(0)~229(X)として逆拡散された形式でベクトルデータファイル82(0)~82(X)に記憶される。

【0157】

[00202]このように、出力データフローパス98(0)~98(X)内に設けられた逆拡散回路230により、結果として生じる出力ベクトルデータサンプルセット228(0)~228(X)は、最初にベクトルデータファイル82(0)~82(X)に記憶され、次いでベクトルデータファイル82(0)~82(X)からフェッチされ、逆拡散され、ベクトルデータファイル82(0)~82(X)に逆拡散された形式で再記憶される必要がない。結果として生じる出力ベクトルデータサンプルセット228(0)~228(X)は、ベクトルデータファイル82(0)~82(X)に記憶される前に逆拡散される。このようにして、逆拡散された、結果として生じる出力ベクトルデータサンプルセット229(0)~229(Z)は、実行ユニット84(0)~84(X)において実行されるべき次のベクトル処理動作を遅延させる可能性がある、さらなる後処理ステップを必要とせずに、ベクトルデータファイル82(0)~82(X)に記憶される。したがって、VPE22(5)内のデータフローパスの効率は、結果として生じる出力ベクトルデータサンプルセット228(0)~228(X)の逆拡散によって制限されない。実行ユニット84(0)~84(X)における次のベクトル処理は、結果として生じる出力ベクトルデータサンプルセット228(0)~228(X)がベクトルデータファイル82(0)~82(X)に逆拡散された、結果として生じる出力ベクトルデータサンプルセット229(0)~229(Z)として逆拡散された形式で記憶されるときに、データフローの制

限ではなく、コンピュータリソースのみによって制限される。

【 0 1 5 8 】

[00203]さらに、実行ユニット84(0)~84(X)とベクトルデータファイル82(0)~82(X)との間の出力データフローパス98(0)~98(X)内に逆拡散回路230を設けることによって、結果として生じる出力ベクトルデータサンプルセット228(0)~228(X)は、ベクトルデータファイル82(0)~82(X)と実行ユニット84(0)~84(X)との間の入力データフローパス80(0)~80(X)内のベクトルデータレーン100と交差する必要がない。異なるベクトルデータレーン100の間の入力ベクトルデータサンプルセット86(0)~86(X)内の入力ベクトルデータサンプル86の逆拡散のためのデータフローパスを設けると、ルーティングの複雑さが増大するはずである。結果として、実行ユニット84(0)~84(X)は、入力データフローパス80(0)~80(X)において逆拡散動作が実行されている間、過少利用される可能性がある。同様に、上記で説明されたように、入力データフローパス80(0)~80(X)における、結果として生じる出力ベクトルデータサンプルセット228(0)~228(X)の逆拡散は、結果として生じる出力ベクトルデータサンプルセット228(0)~228(X)が最初に図27のVPE22(5)内のベクトルデータファイル82(0)~82(X)に記憶されることを必要とするはずであり、それにより、再フェッチおよび逆拡散されるとき電力消費が増大し、および/または逆拡散動作が実行されている間に遅延する可能性がある実行ユニット84(0)~84(X)の過少利用のリスクがある。

【 0 1 5 9 】

[00204]図4、図11、図19、および図23のVPE22(1)~22(4)内に設けられた共通構成要素が、図27のVPE22(5)内に設けられることに留意されたい。共通構成要素は、共通要素番号とともに図27のVPE22(5)において示される。VPE22(1)~22(4)内の上記これらの共通構成要素の前の記載および説明は、図27のVPE22(5)にも適用可能であり、したがってここでは再び記載されない。

【 0 1 6 0 】

[00205]引き続き図27を参照すると、より具体的には、逆拡散回路230は、出力データフローパス98(0)~98(X)上の逆拡散回路入力232(0)~232(X)上で、結果として生じる出力ベクトルデータサンプルセット228(0)~228(X)を受信するように構成される。逆拡散回路230は、逆拡散された、結果として生じる出力ベクトルデータサンプルセット229(0)~229(Z)を供給するために、結果として生じる出力ベクトルデータサンプルセット228(0)~228(X)を逆拡散するように構成される。下記でより詳細に説明されるように、逆拡散された、結果として生じる出力ベクトルデータサンプル229の数は、逆拡散された、結果として生じる出力ベクトルデータサンプルセット229(0)~229(Z)内では「Z+1」である。逆拡散された、結果として生じる出力ベクトルデータサンプルセット229(0)~229(Z)内の逆拡散された、結果として生じる出力ベクトルデータサンプル229の数は、結果として生じる出力ベクトルデータサンプルセット228(0)~228(X)を逆拡散するために使用される拡散係数に依存する。逆拡散回路230は、記憶用にベクトルデータファイル82(0)~82(X)に供給されるために、出力データフローパス98(0)~98(X)内の逆拡散回路出力234(0)~234(X)上に逆拡散された、結果として生じる出力ベクトルデータサンプルセット229(0)~229(Z)を供給するように構成される。

【 0 1 6 1 】

[00206]この実施形態における出力データフローパス98(0)~98(X)内のベクトルデータファイル82(0)~82(X)に逆拡散された、結果として生じる出力ベクトルデータサンプルセット229(0)~229(Z)を供給するための図27のVPE22(5)のさらなる詳細および特徴のさらなる説明が次に記載される。この関連で、図28は、結果として生じる出力ベクトルデータサンプルセット228(0)~228(X)

）の逆拡散を必要とする例示的なベクトル命令に従って、逆拡散回路 230 を利用する図 27 の V P E 22 (5) において実行され得る逆拡散ベクトル処理動作 236 から得られた結果として生じる出力ベクトルデータサンプルセット 228 (0) ~ 228 (X) の例示的な逆拡散を示すフローチャートである。

【 0 1 6 2 】

[00207] 図 27 と図 28 とを参照すると、ベクトル命令に従う逆拡散ベクトル処理動作 236 に従って処理されるべき入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) が、ベクトルデータファイル 82 (0) ~ 82 (X) からフェッチされ、入力データフローパス 80 (0) ~ 80 (X) 内に供給される (図 28 のブロック 238)。結果として生じる逆拡散ベクトル処理動作 236 のための、結果として生じる出力ベクトルデータサンプルセット 228 (0) ~ 228 (X) の幅に応じて、ベクトル命令のプログラミングに従う逆拡散ベクトル処理動作 236 を提供するために、図 27 の V P E 22 (5) 内のベクトルデータレーン 100 (0) ~ 100 (X) の 1 つ、いくつか、またはすべてが利用され得る。逆拡散ベクトル処理動作 236 が、結果として生じる出力ベクトルデータサンプルセット 228 (0) ~ 228 (X) の中のすべての、結果として生じる出力ベクトルデータサンプル 228 の逆拡散を実行することを要する場合、実行ユニット 84 (0) ~ 84 (X) からの出力データフローパス 98 (0) ~ 98 (X) 内のすべてのベクトルデータレーン 100 (0) ~ 100 (X) が逆拡散ベクトル処理動作 236 に利用され得る。代替として、逆拡散ベクトル処理動作 236 は、結果として生じる出力ベクトルデータサンプルセット 228 (0) ~ 228 (X) の中の、結果として生じる出力ベクトルデータサンプル 228 のサブセットを逆拡散することのみを要する場合があります、したがって、結果として生じる出力ベクトルデータサンプル 228 のサブセットに対応する出力データフローパス 98 内のベクトルデータレーン 100 のみを要する。

【 0 1 6 3 】

[00208] 引き続き図 27 と図 28 とを参照すると、逆拡散ベクトル処理動作が図 27 の V P E 22 (5) 内の逆拡散回路 230 によって実行されるより前に、フェッチされた入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) が、実行ユニット 84 (0) ~ 84 (X) にある入力データフローパス 80 (0) ~ 80 (X) から受信される (図 28 のブロック 240)。実行ユニット 84 (0) ~ 84 (X) が、ベクトル命令に従って提供されたベクトル処理動作に従って、受信された入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) に対して 1 つまたは複数のベクトル処理動作を実行する (図 28 のブロック 242)。たとえば、実行ユニット 84 (0) ~ 84 (X) は、結果として生じる出力ベクトルデータサンプルセット 228 (0) ~ 228 (X) を供給するために、ベクトル処理動作を実行するための、入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) と、基準ベクトルデータサンプルセット 130 (0) ~ 130 (X) 内のコードシーケンスとを使用して、乗算および/または累算を提供する。たとえば、結果として生じる出力ベクトルデータサンプルセット 228 (0) ~ 228 (X) は、入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) のベクトル処理に基づく場合があります、基準ベクトルデータサンプルセット 130 (0) ~ 130 (X) は、図 27 の V P E 22 (5) の出力データフローパス 98 (0) ~ 98 (X) 内に供給される。

【 0 1 6 4 】

[00209] 引き続き図 27 と図 28 とを参照すると、結果として生じる出力ベクトルデータサンプルセット 228 (0) ~ 228 (X) を逆拡散することが望ましい場合、逆拡散ベクトル処理動作 236 は、結果として生じる出力ベクトルデータサンプルセット 228 (0) ~ 228 (X) がベクトルデータファイル 82 (0) ~ 82 (X) に記憶される前に実行され得る。この例では、結果として生じる出力ベクトルデータサンプルセット 228 (0) ~ 228 (X) は、図 27 の V P E 22 (5) 内の実行ユニット 84 (0) ~ 84 (X) とベクトルデータファイル 82 (0) ~ 82 (X) との間に設けられた出力データフローパス 98 (0) ~ 98 (X) 内に設けられた逆拡散回路 230 に供給される。逆拡散回路 230 は、実行されているベクトル命令に従って、およびベクトル命令がベクト

ルデータファイル 82(0) ~ 82(X) に記憶されるべき、結果として生じる出力ベクトルデータサンプルセット 228(0) ~ 228(X) の逆拡散を要求する場合、出力データフローパス 98(0) ~ 98(X) 内で、結果として生じる出力ベクトルデータサンプルセット 228(0) ~ 228(X) を選択的に逆拡散するようにプログラム可能である。結果として生じる出力ベクトルデータサンプルセット 228(0) ~ 228(X) がベクトルデータファイル 82(0) ~ 82(X) に記憶されることなく、逆拡散回路 230 が、実行されているベクトル命令に従う逆拡散プログラミングに従って、結果として生じる出力ベクトルデータサンプルセット 228(0) ~ 228(X) を逆拡散する (図 28 のブロック 244)。

【0165】

10

[00210]このようにして、結果として生じる出力ベクトルデータサンプルセット 228(0) ~ 228(X) は、それにより実行ユニット 84(0) ~ 84(X) において遅延をもたらす、最初にベクトルデータファイル 82(0) ~ 82(X) に記憶され、再フェッチされ、後処理動作において逆拡散され、ベクトルデータファイル 82(0) ~ 82(X) に逆拡散されたフォーマットで記憶される必要がない。結果として生じる出力ベクトルデータサンプルセット 228(0) ~ 228(X) が、逆拡散後処理を必要とせずに、ベクトルデータファイル 82(0) ~ 82(X) に逆拡散された、結果として生じる出力ベクトルデータサンプルセット 229(0) ~ 229(Z) として記憶される (図 28 のブロック 246)。

【0166】

20

[00211]図 29 は、図 27 の VPE 22(5) 内の実行ユニット 84(0) ~ 84(X) とベクトルデータファイル 82(0) ~ 82(X) との間の出力データフローパス 98(0) ~ 98(X) 内に設けられ得る、例示的な逆拡散回路 230 の概略図である。逆拡散回路 230 は、基準ベクトルデータサンプルセット 130(0) ~ 130(X) 内の反復コードシーケンスの様々な拡散係数に対して、逆拡散された、結果として生じる出力ベクトルデータサンプルセット 229(0) ~ 229(Z) を供給するために、結果として生じる出力ベクトルデータサンプルセット 228(0) ~ 228(X) の逆拡散を提供するように構成される。結果として生じる出力ベクトルデータサンプルセット 228(0) ~ 228(X) は、図 27 に示されたように、実行ユニット出力 96(0) ~ 96(X) から逆拡散回路 230 に供給される。結果として生じる出力ベクトルデータサンプルセット 228(0) ~ 228(X) の拡散係数は知られていない場合があるので、図 27 のシーケンス番号発生器 134 によって生成された基準ベクトルデータサンプルセット 130(0) ~ 130(X) 内の反復シーケンス番号の様々な拡散係数を用いて、結果として生じる出力ベクトルデータサンプルセット 228(0) ~ 228(X) を逆拡散することが望ましい場合がある。

30

【0167】

[00212]たとえば、結果として生じる出力ベクトルデータサンプルセット 228(0) ~ 228(X) が 32 個のサンプルを含んでいて、結果として生じる出力ベクトルデータサンプルセット 228(0) ~ 228(X) 全体が 4 の拡散係数を想定して逆拡散された場合、結果として生じる出力ベクトルデータサンプルセット 228(0) ~ 228(X) の逆拡散が実行された後、逆拡散された、結果として生じる出力ベクトルデータサンプルセット 229(0) ~ 229(Z) は、8 個の逆拡散サンプル (すなわち、32 サンプル / 4 の拡散係数) を含んでいるはずである。しかしながら、この同じ例において、結果として生じる出力ベクトルデータサンプルセット 228(0) ~ 228(X) 全体が 8 の拡散係数を想定して逆拡散された場合、結果として生じる出力ベクトルデータサンプルセット 228(0) ~ 228(X) の逆拡散が実行された後、逆拡散された、結果として生じる出力ベクトルデータサンプルセット 229(0) ~ 229(Z) は、4 個の逆拡散サンプル (すなわち、32 サンプル / 8 の拡散係数) を含んでいるはずである。

40

【0168】

[00213]このように、引き続き図 29 を参照すると、逆拡散回路 230 は、異なる数の

50

拡散係数に対して、結果として生じる出力ベクトルデータサンプルセット228(0)~228(X)を逆拡散するように構成される。この実施形態における逆拡散回路230は、1つのベクトル処理動作/1つのベクトル命令における様々な拡散係数に対して、逆拡散された、結果として生じる出力ベクトルデータサンプルセット229(0)~229(Z)を供給するように構成される。この関連で、逆拡散回路230は、結果として生じる出力ベクトルデータサンプルセット228(0)~228(X)を受信するために、実行ユニット出力96(0)~96(X)に結合された加算器ツリー248を含んでいる。逆拡散回路230の加算器ツリー248は、それらのそれぞれのベクトルデータレーン100(0)~100(X)内で、結果として生じる出力ベクトルデータサンプルセット228(0)~228(X)の各サンプル228を受信するように構成される。加算器ツリー248内に第1の加算器ツリーレベル248(1)が設けられる。第1の加算器ツリーレベル248(1)は、4の拡散係数によって、結果として生じる出力ベクトルデータサンプルセット228(0)~228(X)内のサンプル228を拡散することができるように、加算器250(0)~250((X+1)*2)-1)、250(7)から構成される。出力データフローパス98(0)~98(X)から、結果として生じる出力ベクトルデータサンプルセット228(0)~228(X)をラッチするために、ラッチ251(0)~251(X)が逆拡散回路230内に設けられる。

【0169】

[00214]たとえば、結果として生じる出力ベクトルデータサンプルセット228(0)~228(X)内の各サンプル228が32ビット幅であり、2つの16ビット複素数のベクトルデータ(すなわち、フォーマットI8Q8に従う第1のベクトルデータおよびフォーマットI8Q8に従う第2のベクトルデータ)から構成される場合、結果として生じる出力ベクトルデータサンプルセット228(0)~228(X)内の2つの、結果として生じる出力ベクトルデータサンプル228の中の4つのベクトルデータサンプルを、1つの逆拡散された、結果として生じる出力ベクトルデータサンプルの中に逆拡散するために、4の拡散係数が適用される可能性がある。たとえば、図29に示されたように、加算器250(0)は、結果として生じる出力ベクトルデータサンプル228(0)と228(1)とを、それらのサンプルのための4の拡散係数によって逆拡散するように構成される。同じく、加算器250(1)は、結果として生じる出力ベクトルデータサンプル228(2)と228(3)とを、それらのサンプルのための4の拡散係数によって逆拡散するように構成される。加算器250((X+1)/2)-1)、250(7)は、4の拡散係数を用いて、逆拡散ベクトルデータサンプルセット252(0)~252((X+1)/2)-1)、252(7)を供給するために、結果として生じる出力ベクトルデータサンプルセット228(X-1)と228(X)とを逆拡散するように構成される。加算器250((X+1)/2)-1)、250(7)によって実行された逆拡散からの逆拡散ベクトルデータサンプルセット252(0)~252((X+1)/2)-1)、252(7)は、ラッチ255(0)~255((X+1)/2)-1)、255(7)の中にラッチされる。

【0170】

[00215]逆拡散ベクトル処理動作236が4の拡散係数による、結果として生じる出力ベクトルデータサンプルセット228(0)~228(X)の逆拡散を必要とする場合、下記でより詳細に記載されるように、逆拡散ベクトルデータサンプルセット252(0)~252((X+1)/2)-1)、252(7)は、逆拡散された、結果として生じる出力ベクトルデータサンプルセット229(0)~229(Z)として供給され得るし、ここで、「Z」は7である。しかしながら、逆拡散ベクトル処理動作236がより高い拡散係数(たとえば、8、16、32、64、128、256)を要求する場合、逆拡散ベクトルデータサンプルセット252(0)~252((X+1)/2)-1)、252(7)は、逆拡散された、結果として生じる出力ベクトルデータサンプルセット229(0)~229(Z)として供給されない。逆拡散ベクトルデータサンプルセット252(0)~252((X+1)/2)-1)、252(7)は、加算器254(0)~2

10

20

30

40

50

5 4 (((X + 1) / 4) - 1)、2 5 4 (3) への第 2 の加算器ツリーレベル 2 4 8 (2) に供給される。この関連で、加算器 2 5 4 (0) は、それらのサンプルのための 8 の拡散係数を有する、結果として生じる逆拡散ベクトルデータサンプル 2 5 6 (0) を供給するために、逆拡散ベクトルデータサンプル 2 5 2 (0) および 2 5 2 (1) に対して逆拡散を実行するように構成される。同じく、加算器 2 5 4 (1) は、それらのサンプルのための 8 の拡散係数を有する、結果として生じる逆拡散ベクトルデータサンプル 2 5 6 (1) を供給するために、逆拡散ベクトルデータサンプル 2 5 2 (2) および 2 5 2 (3) に対して逆拡散を実行するように構成される。加算器 2 5 4 (((X + 1) / 4) - 1)、2 5 4 (3) は、8 の拡散係数を有する、結果として生じる逆拡散ベクトルデータサンプル 2 5 6 (((X + 1) / 4) - 1)、2 5 6 (3) を供給するために、逆拡散ベクトルデータサンプルセット 2 5 2 (((X + 1) / 4) - 2)、2 5 2 (((X + 1) / 4) - 1)、2 5 2 (3) に対して逆拡散を実行するように構成される。加算器 2 5 4 (0) ~ 2 5 4 (((X + 1) / 4) - 1)、2 5 4 (3) によって実行された逆拡散からの、結果として生じる逆拡散ベクトルデータサンプルセット 2 5 6 (0) ~ 2 5 6 (((X + 1) / 4) - 1)、2 5 6 (3) は、ラッチ 2 5 7 (0) ~ 2 5 7 (((X + 1) / 4) - 1)、2 5 7 (3) の中にラッチされる。

10

【 0 1 7 1 】

[00216]引き続き図 2 9 を参照すると、逆拡散ベクトル処理動作 2 3 6 が 8 の拡散係数による、結果として生じる出力ベクトルデータサンプルセット 2 2 8 (0) ~ 2 2 8 (X) の逆拡散を必要とする場合、下記でより詳細に記載されるように、逆拡散ベクトルデータサンプルセット 2 5 6 (0) ~ 2 5 6 (((X + 1) / 4) - 1)、2 5 6 (3) は、逆拡散された、結果として生じる出力ベクトルデータサンプルセット 2 2 9 (0) ~ 2 2 9 (Z) として供給され得るし、ここで、「Z」は 3 である。しかしながら、逆拡散ベクトル処理動作 2 3 6 が 8 よりも高い拡散係数（たとえば、1 6、3 2、6 4、1 2 8、2 5 6）を要求する場合、逆拡散ベクトルデータサンプルセット 2 5 6 (0) ~ 2 5 6 (((X + 1) / 4) - 1)、2 5 6 (3) は、逆拡散された、結果として生じる出力ベクトルデータサンプルセット 2 2 9 (0) ~ 2 2 9 (Z) として供給されない。逆拡散ベクトルデータサンプルセット 2 5 6 (0) ~ 2 5 6 (((X + 1) / 4) - 1)、2 5 6 (3) は、加算器 2 5 8 (0) ~ 2 5 8 (((X + 1) / 8) - 1)、2 5 8 (1) への第 3 の加算器ツリーレベル 2 4 8 (3) に供給される。この関連で、加算器 2 5 8 (0) は、それらのサンプルのための 1 6 の拡散係数を供給するために、逆拡散ベクトルデータサンプル 2 5 6 (0) および 2 5 6 (1) に対して逆拡散を実行するように構成される。同じく、加算器 2 5 8 (1) は、1 6 の拡散係数を有する逆拡散ベクトルデータサンプルセット 2 6 0 (0) ~ 2 6 0 (((X + 1) / 8) - 1)、2 6 0 (1) を供給するために、逆拡散ベクトルデータサンプル 2 5 6 (2) および 2 5 6 (3) に対して逆拡散を実行するように構成される。加算器 2 5 8 (0) ~ 2 5 8 (((X + 1) / 8) - 1)、2 5 8 (1) によって実行された逆拡散からの逆拡散ベクトルデータサンプルセット 2 6 0 (0) ~ 2 6 0 (((X + 1) / 8) - 1)、2 6 0 (1) は、ラッチ 2 5 9 (0) ~ 2 5 9 (((X + 1) / 8) - 1)、2 5 9 (2) の中にラッチされる。

20

30

【 0 1 7 2 】

[00217]引き続き図 2 9 を参照すると、逆拡散ベクトル処理動作 2 3 6 が 1 6 の拡散係数による、結果として生じる出力ベクトルデータサンプルセット 2 2 8 (0) ~ 2 2 8 (X) の逆拡散を必要とする場合、下記でより詳細に記載されるように、逆拡散ベクトルデータサンプルセット 2 6 0 (0) ~ 2 6 0 (((X + 1) / 8) - 1)、2 5 6 (1) は、逆拡散された、結果として生じる出力ベクトルデータサンプルセット 2 2 9 (0) ~ 2 2 9 (Z) として供給され得るし、ここで、「Z」は 1 である。しかしながら、逆拡散ベクトル処理動作 2 3 6 が 1 6 よりも高い拡散係数（たとえば、3 2、6 4、1 2 8、2 5 6）を要求する場合、逆拡散ベクトルデータサンプルセット 2 6 0 (0) ~ 2 6 0 (((X + 1) / 8) - 1)、2 6 0 (1) は、逆拡散された、結果として生じる出力ベクトルデータサンプルセット 2 2 9 (0) ~ 2 2 9 (Z) として供給されない。逆拡散ベクトル

40

50

データサンプルセット 260(0) ~ 260((X+1)/8)-1)、260(1)は、加算器 262 への第 4 の加算器ツリーレベル 248(4)に供給される。この関連で、加算器 262 は、32 の拡散係数を有する逆拡散ベクトルデータサンプル 264 を供給するために、逆拡散ベクトルデータサンプル 260(0)および 260(1)に対して逆拡散を実行するように構成される。加算器 262 によって実行された逆拡散からの逆拡散ベクトルデータサンプル 264 は、ラッチ 266 および 268 の中にラッチされる。

【0173】

[00218]引き続き図 29 を参照すると、逆拡散ベクトル処理動作 236 が 32 の拡散係数による、結果として生じる出力ベクトルデータサンプルセット 228(0) ~ 228(X)の逆拡散を必要とする場合、下記でより詳細に記載されるように、逆拡散ベクトルデータサンプル 264 は、逆拡散された、結果として生じる出力ベクトルデータサンプル 229 として供給され得る。しかしながら、逆拡散ベクトル処理動作 236 が 32 よりも高い拡散係数(たとえば、64、128、256)を要求する場合、逆拡散ベクトルデータサンプル 264 は、逆拡散された、結果として生じる出力ベクトルデータサンプルセット 229 として供給されない。逆拡散ベクトルデータサンプル 264 は、ベクトルデータファイル 82 に記憶される必要なしに、ラッチ 268 の中にラッチされたままである。上述されたように、32 の拡散係数を使用して逆拡散されるために、別の結果として生じる出力ベクトルデータサンプルセット 228(0) ~ 228(X)は、さらなる処理サイクルにわたって、ラッチ 251(0) ~ 251(X)の中にロードされる。結果として生じる逆拡散ベクトルデータサンプル 264' は、64 の拡散係数を有する逆拡散ベクトルデータサンプル 272 を供給するために、第 5 の加算器ツリー 248(5)内の加算器 270 により、前の逆拡散ベクトルデータサンプル 264 に加算される。選択器 273 は、32 の拡散係数を有する逆拡散ベクトルデータサンプル 264、または 64 の拡散係数を有する逆拡散ベクトルデータサンプル 264' のどちらが、ラッチ 274 の中にラッチされる逆拡散ベクトルデータサンプル 272 としてラッチされるかを制御する。さらなる結果として生じる出力ベクトルデータサンプルセット 228(0) ~ 228(X)をラッチし、さらなる結果として生じる出力ベクトルデータサンプルセット 228(0) ~ 228(X)を逆拡散するこの同じプロセスは、必要な場合、64 よりも大きい拡散係数を達成するために実行され得る。逆拡散ベクトルデータサンプル 272 は、最終的に、逆拡散ベクトル処理動作 236 のための所望の拡散係数に従って、所望の逆拡散された、結果として生じる出力ベクトルデータサンプル 229 として、ラッチ 274 の中にラッチされる。

【0174】

[00219]引き続き図 29 を参照すると、逆拡散ベクトル処理動作 236 においてどの拡散係数が要求されても、逆拡散された、結果として生じる出力ベクトルデータサンプルセット 229(0) ~ 229(Z)は、図 27 のベクトルデータファイル 82(0) ~ 82(X)に記憶される必要がある。次に説明されるように、図 29 の逆拡散回路 230 はまた、逆拡散された、結果として生じる出力ベクトルデータサンプルセット 229(0) ~ 229(Z)を形成するために、結果として生じる出力ベクトルデータサンプルセット 228(0) ~ 228(X)に対して逆拡散ベクトル処理動作 236 を実行する結果としてもたらされた逆拡散された、結果として生じる出力ベクトルデータサンプル 229 をラッチ 276(0) ~ 276(X)の中にロードするように構成される。逆拡散された、結果として生じる出力ベクトルデータサンプルセット 229(0) ~ 229(Z)は、記憶されるためにベクトルデータファイル 82(0) ~ 82(X)に供給され得る。このようにして、逆拡散回路 230 によって作成された逆拡散された、結果として生じる出力ベクトルデータサンプルセット 229(0) ~ 229(Z)を記憶するために、ベクトルデータファイル 82(0) ~ 82(X)に対して 1 回の書込みが必要とされるにすぎない。図 29 の逆拡散回路 230 内の加算器ツリー 248(1) ~ 248(5)は、逆拡散ベクトル処理動作 236 においてどの拡散係数が要求されても、拡散係数 4、8、16、および 32 のすべてに対して逆拡散された、結果として生じる出力ベクトルデータサンプル 229 を生成することができる。代替として、所望の拡散係数に従って逆拡散ベクトル処理動作

236を実行する必要がない加算器ツリー内の加算器は、無効にされ得るか、または0を加算するように構成され得る。しかしながら、これらの逆拡散された、結果として生じる出力ベクトルデータサンプル229のどれが記憶されるためにラッチ276(0)~276(X)に供給されるかを決定するために、次に説明されるように、選択器278(0)~278((X+1)/4)-1)、278(3)が設けられる。

【0175】

[00220]この関連で、引き続き図29を参照すると、選択器278(0)は、実行されている逆拡散ベクトル処理動作236に基づいて、それぞれ、加算器250(0)、254(0)、258(0)からの拡散係数4、8、および16、ならびに加算器262、270からの拡散係数32、64、128、256のいずれかに対して、逆拡散された、結果として生じる出力ベクトルデータサンプル229を選択することができる。選択器278(1)は、実行されている逆拡散ベクトル処理動作236に基づいて、それぞれ、加算器250(1)、254(1)、および258(1)からの拡散係数4、8、および16に対して、逆拡散された、結果として生じる出力ベクトルデータサンプル229を選択することができる。選択器278(2)は、実行されている逆拡散ベクトル処理動作236に基づいて、それぞれ、加算器250(2)および254(2)からの拡散係数4および8に対して、逆拡散された、結果として生じる出力ベクトルデータサンプル229を選択することができる。選択器278(3)は、実行されている逆拡散ベクトル処理動作236に基づいて、それぞれ、加算器250(3)および254(3)からの拡散係数4および8に対して、逆拡散された、結果として生じる出力ベクトルデータサンプル229を選択することができる。選択器278(4)は、実行されている逆拡散ベクトル処理動作236に基づいて、それぞれ、加算器ツリー248(1)および248(2)からの拡散係数4および8に対して、逆拡散された、結果として生じる出力ベクトルデータサンプル229を選択することができる。8の拡散係数を供給することが選択器278(0)~278(3)によって完全に満足され得るので、選択器は、加算器250(4)~250(7)から供給され逆拡散された、結果として生じる出力ベクトルデータサンプル229を制御するためには設けられない。

【0176】

[00221]引き続き図29を参照すると、それぞれ、選択器278(0)~278((X+1)/4)-1)、278(3)および加算器250(4)~250((X+1)/2)-1)、250(7)によって選択され逆拡散された、結果として生じる出力ベクトルデータサンプル229を受信するために、一連のデータスライサ280(0)~280((X+1)/2)-1)、280(7)が設けられる。データスライサ280(0)~280((X+1)/2)-1)、280(7)は、その受信され逆拡散された、結果として生じる出力ベクトルデータサンプル229が論理高レベル(たとえば、論理「1」)として特徴付けられるか、論理低レベル(たとえば、論理「0」)として特徴付けられるかを選択するように構成される。逆拡散された、結果として生じる出力ベクトルデータサンプル229は、次いで、クロスバー282への接続を介して、記憶されるためにラッチ276(0)~276(X)の中の所望のラッチ276に転送される。クロスバー282は、様々なラッチ276(0)~276(X)に、逆拡散ベクトル処理動作236に従って逆拡散された、結果として生じる出力ベクトルデータサンプル229を供給する柔軟性を提供する。このようにして、逆拡散された、結果として生じる出力ベクトルデータサンプル229は、ベクトルデータファイル82(0)~82(X)に記憶される前に、逆拡散ベクトル処理動作236の様々な繰返しの中で、ラッチ276(0)~276(X)にスタックされ得る。たとえば、逆拡散された、結果として生じる出力ベクトルデータサンプルセット229(0)~229(Z)は、ベクトルデータファイル82(0)~82(X)に記憶される前に、逆拡散ベクトル処理動作236の様々な繰返しの中で、ラッチ276(0)~276(X)にスタックされ得る。このようにして、逆拡散された、結果として生じる出力ベクトルデータサンプルセット229(0)~229(Z)を記憶するためのベクトルデータファイル82(0)~82(X)へのアクセスは、動作効率の

ために最小化され得る。

【 0 1 7 7 】

[00222]たとえば、図 29 に示されたように、クロスバー 282 に結合された選択器 284 (0) ~ 284 (X) は、ラッチ 276 (0) ~ 276 (X) のいずれかの中にデータスライサ 280 (0) からの逆拡散された、結果として生じる出力ベクトルデータサンプル 229 を記憶するように制御され得る。クロスバー 282 に結合された選択器 284 (1)、284 (3)、284 (5)、284 (7)、284 (9)、284 (11)、284 (13)、284 (15) は、ラッチ 276 (1)、276 (3)、276 (5)、276 (7)、276 (9)、276 (11)、276 (13)、および 276 (15) に記憶されるべきデータスライサ 280 (1) からの逆拡散された、結果として生じる出力ベクトルデータサンプル 229 を記憶するように制御され得る。クロスバー 282 に結合された選択器 284 (2)、284 (6)、284 (10)、284 (14) は、ラッチ 276 (2)、276 (6)、276 (10)、および 276 (14) にデータスライサ 280 (2) からの逆拡散された、結果として生じる出力ベクトルデータサンプル 229 を記憶するように制御され得る。クロスバー 282 に結合された選択器 284 (3)、284 (7)、284 (11)、284 (15) は、ラッチ 276 (3)、276 (7)、276 (11)、および 276 (15) にデータスライサ 280 (3) からの逆拡散された、結果として生じる出力ベクトルデータサンプル 229 を記憶するように制御され得る。クロスバー 282 に結合された選択器 284 (4) および 284 (12) は、ラッチ 276 (4) および 276 (12) にデータスライサ 280 (4) からの逆拡散された、結果として生じる出力ベクトルデータサンプル 229 を記憶するように制御され得る。クロスバー 282 に結合された選択器 284 (5) および 284 (13) は、ラッチ 276 (5) および 276 (13) にデータスライサ 280 (5) からの逆拡散された、結果として生じる出力ベクトルデータサンプル 229 を記憶するように制御され得る。クロスバー 282 に結合された選択器 284 (6) および 284 (14) は、ラッチ 276 (6) または 276 (14) にデータスライサ 280 (6) からの逆拡散された、結果として生じる出力ベクトルデータサンプル 229 を記憶するように制御され得る。クロスバー 282 に結合された選択器 284 (7) および 284 (15) は、ラッチ 276 (7) または 276 (15) にデータスライサ 280 (7) からの逆拡散された、結果として生じる出力ベクトルデータサンプル 229 を記憶するように制御され得る。

【 0 1 7 8 】

[00223]引き続き図 29 を参照すると、逆拡散回路 230 は、実行されるべきベクトル命令に従って、結果として生じる出力ベクトルデータサンプル 228 (0) ~ 228 (X) に対して逆拡散動作を実行するか、または実行しないように構成されるようにプログラムされ得る。この関連で、ベクトルデータファイル 82 (0) ~ 82 (X) に記憶されるために、それぞれ、結果として生じる出力ベクトルデータサンプルセット 228 (0) ~ 228 (X) に対して逆拡散動作を実行するか、またはラッチ 276 (0) ~ 276 (X) に、結果として生じる出力ベクトルデータサンプルセット 228 (0) ~ 228 (X) を単に供給するために、図 29 の逆拡散構成入力 286 が逆拡散回路 230 に提供され得る。このようにして、逆拡散回路 230 は、ベクトル命令が実行されるべきそのような処理を提供しない場合、結果として生じる出力ベクトルデータサンプルセット 228 (0) ~ 228 (X) を逆拡散しないようにプログラムされ得る。逆拡散構成入力 284 は、図 27 の VPE 22 (5) によるベクトル処理において柔軟性を提供するように、ベクトル命令ごとに構成および再構成され得る。たとえば、逆拡散構成入力 284 は、必要な場合実行ユニット 84 (0) ~ 84 (X) を十分に利用して、必要に応じて逆拡散を提供するように、ベクトル命令のクロックサイクルごとに、必要な場合クロックサイクルごとに、構成および再構成され得る。

【 0 1 7 9 】

[00224]いくつかの他のワイヤレスベースバンド動作は、拡散スペクトルデータシーケンスの逆拡散以外の理由で前の処理動作から決定されたデータサンプルのマーキングを必

要とする。たとえば、ベクトルデータレーン100(0)~100(X)によって提供された実行ユニット84(0)~84(X)のためのデータフローパスよりも広い変化幅のベクトルデータサンプルを累算することが望ましい場合がある。別の例として、ベクトル処理動作において出力ベクトルデータのマーキングを提供するために、様々な実行ユニット84(0)~84(X)からの出力ベクトルデータサンプルのドット積乗算を提供することが望ましい場合がある。VPE内のベクトルデータレーン100(0)~100(X)は、マージされたベクトル処理動作を提供するために、ベクトルデータレーン100(0)~100(X)と交差するためのベクトル内データパスを提供する複雑なルーティングを含む可能性がある。しかしながら、様々なベクトルデータレーンと交差してマージされるべき出力ベクトルデータにおける並列化は困難なので、これにより、複雑さが増大し、VPEの効率が低減される可能性がある。ベクトルプロセッサは、実行ユニットからベクトルデータメモリに記憶された出力ベクトルデータの後処理マーキングを実行する回路を含む可能性がある。ベクトルデータメモリに記憶された後処理された出力ベクトルデータサンプルは、ベクトルデータメモリからフェッチされ、必要に応じてマージされ、ベクトルデータメモリに戻されて記憶される。しかしながら、この後処理により、VPEの次のベクトル処理動作が遅延し、実行ユニット内のコンピュータ構成要素が過剰利用される原因になる可能性がある。

【0180】

[00225]たとえば、前述されたVPE内のベクトルデータファイル82(0)、82(1)内に供給された2つの入力ベクトルデータサンプル290(0)、290(1)が図30に示される。これらの2つの入力ベクトルデータサンプル290(0)、290(1)と一緒に加算することが望ましい場合がある。この例では、2つの入力ベクトルデータサンプル290(0)、290(1)の和は「0x11250314E」であり、それはベクトルデータレーン100(0)または100(1)のいずれかよりも大きいデータ幅を有する。実行ユニット84(0)、84(1)が、ベクトルデータレーン100(0)、100(1)をまたぐ2つの実行ユニット84(0)、84(1)の間の桁上げ論理を提供することを含む、2つの入力ベクトルデータサンプル290(0)、290(1)一緒にの和の実行を行うことが可能になるように、ベクトルデータレーン100(0)、100(1)の間のベクトルデータルーティングを提供するために、データフローパスがVPE22内に設けられる可能性がある。マージされたベクトルデータサンプルのスカラー結果を供給するために、すべてのベクトルデータレーン100(0)~100(X)と交差する能力が必要となる場合があり、それにより、データフローパス内の複雑さがさらに増大する場合がある。しかしながら、上記で説明されたように、これにより、データフローパス内の複雑さが加わるはずであり、それにより、複雑さが増大し、場合によっては効率が低減される。

【0181】

[00226]この問題に対処するために、下記で開示される実施形態は、VPE内の実行ユニットとベクトルデータメモリとの間の出力データフローパス内に設けられたマーキング回路を含むVPEを含む。マーキング回路は、出力ベクトルデータサンプルセットが実行ユニットからベクトルデータメモリに出力データフローパスを介して供給されている間に、インフライトの実行ユニットによって供給された出力ベクトルデータサンプルセットからの出力ベクトルデータサンプルをマージするように構成される。出力ベクトルデータサンプルのインフライトマーキングは、実行ユニットによって供給された出力ベクトルデータサンプルが、ベクトルデータメモリに記憶される前にマージされ得ることを意味し、その結果、得られた出力ベクトルデータサンプルセットはマージされたフォーマットでベクトルデータメモリに記憶される。マージされた出力ベクトルデータサンプルは、実行ユニット内で実行されるべき次のベクトル処理動作を遅延させる可能性がある、さらなる後処理ステップを必要とせず、ベクトルデータファイルに記憶され得る。したがって、VPE内のデータフローパスの効率は、ベクトルデータマーキング動作によって制限されない。マージされたベクトルデータサンプルがベクトルデータメモリに記憶されるとき、実行

ユニット内の次のベクトル処理は、データフローの制限ではなく、コンピュータリソースのみによって制限される。

【 0 1 8 2 】

[00227]この関連で、図 3 1 は、図 2 の V P E 2 2 として提供され得る別の例示的な V P E 2 2 (6) の概略図である。下記でより詳細に記載されるように、図 3 1 の V P E 2 2 (6) は、ベクトルデータサンプルの再フェッチが除去または低減され、電力消費が低減される、V P E 2 2 (6) 内のベクトルデータファイル 8 2 (0) ~ 8 2 (X) に記憶されるべき、ベクトル処理動作のためのコードシーケンスを用いて実行ユニット 8 4 (0) ~ 8 4 (X) によって供給される、結果として生じる出力ベクトルデータサンプルセット 2 9 2 (0) ~ 2 9 2 (X) のインフライトマーキングを提供するように構成される。結果として生じる出力ベクトルデータサンプルセット 2 9 2 (0) ~ 2 9 2 (X) は、結果として生じる出力ベクトルデータサンプル 2 9 2 (0)、...、2 9 2 (X) から構成される。非限定的な例として、マージベクトル処理動作は、結果として生じる出力ベクトルデータサンプル 2 9 2 を加算すること、複数の結果として生じる出力ベクトルデータサンプル 2 9 2 の中の最大ベクトルデータサンプル値を決定すること、または複数の結果として生じる出力ベクトルデータサンプル 2 9 2 の中の最小ベクトルデータサンプル値を決定することを含む可能性がある。図 3 1 の V P E 2 2 (6) では、結果として生じる出力ベクトルデータサンプルセット 2 9 2 (0) ~ 2 9 2 (X) の中の、結果として生じる出力ベクトルデータサンプル 2 9 2 は、ベクトルデータファイル 8 2 (0) ~ 8 2 (X) に記憶される前にマージされ得る。

【 0 1 8 3 】

[00228]マーキング回路 2 9 4 は、結果として生じる出力ベクトルデータサンプルセット 2 2 8 (0) ~ 2 2 8 (X) の中の、結果として生じる出力ベクトルデータサンプル 2 2 8 のインフライトマーキングを提供するために、実行されるべきベクトル命令に従うプログラミングに基づいて構成される。マージされた、結果として生じる出力ベクトルデータサンプル 2 9 6 (0) ~ 2 9 6 (Z) は、出力データフローパス 9 8 (0) ~ 9 8 (X) 内のマーキング回路 2 9 4 によって供給される。マージされた結果として生じる出力ベクトルデータサンプル 2 9 6 (0) ~ 2 9 6 (Z) における「Z」は、マージされた、結果として生じる出力ベクトルデータサンプルセット 2 9 6 (0) ~ 2 9 6 (Z) 内のマージされた、結果として生じる出力ベクトルデータサンプル 2 9 6 の数を表す。マージされた、結果として生じる出力ベクトルデータサンプルセット 2 9 6 (0) ~ 2 9 6 (Z) は、この例では 2 9 6 (0)、...、および 2 9 6 (Z) である、結果として生じる出力ベクトルデータサンプル 2 9 6 から構成される。マージされた、結果として生じる出力ベクトルデータサンプルセット 2 9 6 (0) ~ 2 9 6 (Z) 内のマージされた、結果として生じる出力ベクトルデータサンプル 2 9 6 の数は、結果として生じる出力ベクトルデータサンプルセット 2 9 2 (0) ~ 2 9 2 (X) に対して実行されるマーキング動作に依存する。図 3 1 の V P E 2 2 (6) における、結果として生じる出力ベクトルデータサンプル 2 9 2 のインフライトマーキングは、実行ユニット 8 4 (0) ~ 8 4 (X) によって供給された、結果として生じる出力ベクトルデータサンプルセット 2 9 2 (0) ~ 2 9 2 (X) 内の、結果として生じる出力ベクトルデータサンプル 2 9 2 が、ベクトルデータファイル 8 2 (0) ~ 8 2 (X) に記憶される前に、一緒にマージされ得ることを意味する。このようにして、マージされた、結果として生じる出力ベクトルデータサンプルセット 2 9 6 (0) ~ 2 9 6 (Z) のマージされた、結果として生じる出力ベクトルデータサンプル 2 9 6 は、マージされた、結果として生じる出力ベクトルデータサンプルセット 2 9 6 (0) ~ 2 9 6 (Z) として、マージされた形式でベクトルデータファイル 8 2 (0) ~ 8 2 (X) に記憶され得る。

【 0 1 8 4 】

[00229]このように、出力データフローパス 9 8 (0) ~ 9 8 (X) 内に設けられたマーキング回路 2 9 4 により、結果として生じる出力ベクトルデータサンプルセット 2 9 2 (0) ~ 2 9 2 (X) は、最初にベクトルデータファイル 8 2 (0) ~ 8 2 (X) に記憶

され、次いでベクトルデータファイル 82 (0) ~ 82 (X) からフェッチされる必要がない。所望の結果として生じる出力ベクトルデータサンプル 292 はマージされ、結果として生じる出力ベクトルデータサンプル 292 は、ベクトルデータファイル 82 (0) ~ 82 (X) にマージされた形式で再記憶される。結果として生じる出力ベクトルデータサンプルセット 292 (0) ~ 292 (X) からの、結果として生じる出力ベクトルデータサンプル 292 は、ベクトルデータファイル 82 (0) ~ 82 (X) に記憶される前にマージされ得る。このようにして、マージされた、結果として生じる出力ベクトルデータサンプルセット 296 (0) ~ 296 (Z) からのマージされた、結果として生じる出力ベクトルデータサンプル 296 は、実行ユニット 84 (0) ~ 84 (X) において実行されるべき次のベクトル処理動作を遅延させる可能性がある、さらなる後処理ステップを必要とせず、ベクトルデータファイル 82 (0) ~ 82 (X) に記憶される。したがって、VPE 22 (6) 内のデータフローパスの効率は、結果として生じる出力ベクトルデータサンプル 292 のマージングによって制限されない。結果として生じる出力ベクトルデータサンプル 292 がベクトルデータファイル 82 (0) ~ 82 (X) にマージされた形式で記憶されるとき、実行ユニット 84 (0) ~ 84 (X) における次のベクトル処理は、データフローの制限ではなく、コンピュータリソースのみによって制限される。

【0185】

[00230]さらに、実行ユニット 84 (0) ~ 84 (X) とベクトルデータファイル 82 (0) ~ 82 (X) との間の出力データフローパス 98 (0) ~ 98 (X) 内にマージング回路 294 を設けることによって、結果として生じる出力ベクトルデータサンプルセット 292 (0) ~ 292 (X) は、ベクトルデータファイル 82 (0) ~ 82 (X) と実行ユニット 84 (0) ~ 84 (X) との間の入力データフローパス 80 (0) ~ 80 (X) 内のベクトルデータレーン 100 と交差する必要がない。異なるベクトルデータレーン 100 の間の入力ベクトルデータサンプルセット 86 (0) ~ 86 (X) 内の入力ベクトルデータサンプル 86 のマージングのためのデータフローパスを設けると、ルーティングの複雑さが増大するはずである。結果として、入力データフローパス 80 (0) ~ 80 (X) においてマージング動作が実行されている間、実行ユニット 84 (0) ~ 84 (X) は過少利用される可能性がある。同様に、上記で説明されたように、入力データフローパス 80 (0) ~ 80 (X) における、結果として生じる出力ベクトルデータサンプルセット 292 (0) ~ 292 (X) からの、結果として生じる出力ベクトルデータサンプル 292 のマージングは、結果として生じる出力ベクトルデータサンプルセット 292 (0) ~ 292 (X) が最初に図 31 の VPE 22 (6) 内のベクトルデータファイル 82 (0) ~ 82 (X) に記憶されることを必要とするはずであり、それにより、再フェッチおよびマージされるとき電力消費が増大し、および / またはマージング動作が実行されている間に遅延する可能性がある実行ユニット 84 (0) ~ 84 (X) の過少利用のリスクがある。

【0186】

[00231]図 4、図 11、図 19、図 23 および図 27 の VPE 22 (1) ~ 22 (5) 内に設けられた共通構成要素が、図 31 の VPE 22 (6) 内に設けられることに留意されたい。共通構成要素は、共通要素番号とともに図 31 の VPE 22 (6) において示される。VPE 22 (1) ~ 22 (5) 内の上記これらの共通構成要素の前の記載および説明は、図 31 の VPE 22 (6) にも適用可能であり、したがってここでは再び記載されない。

【0187】

[00232]引き続き図 31 を参照すると、より具体的には、マージング回路 294 は、出力データフローパス 98 (0) ~ 98 (X) 上のマージング回路入力 300 (0) ~ 300 (X) 上で、結果として生じる出力ベクトルデータサンプルセット 292 (0) ~ 292 (X) を受信するように構成される。マージング回路 294 は、マージされた、結果として生じる出力ベクトルデータサンプルセット 296 (0) ~ 296 (Z) を供給するために、結果として生じる出力ベクトルデータサンプルセット 292 (0) ~ 292 (X)

からの所望の結果として生じる出力ベクトルデータサンプル292をマージするように構成される。マージされた結果として生じる出力ベクトルデータサンプル296(0)~296(Z)における「Z」は、マージされた、結果として生じる出力ベクトルデータサンプルセット296(0)~296(Z)のビット幅を表す。「Z」は、マージング動作に起因して、「X」によって表される、結果として生じる出力ベクトルデータサンプルセット292(0)~292(X)のビット幅よりも小さい場合がある。下記でより詳細に説明されるように、マージされた、結果として生じる出力ベクトルデータサンプルセット296(0)~296(Z)内のマージされた、結果として生じる出力ベクトルデータサンプル296の数「Z+1」は、一緒にマージされるべき結果として生じる出力ベクトルデータサンプルセット292(0)~292(X)からの、結果として生じる出力ベクトルデータサンプル292に依存する。マージング回路294は、記憶用にベクトルデータファイル82(0)~82(X)に供給されるために、出力データフローパス98(0)~98(X)内のマージング回路出力301(0)~301(X)上にマージされた、結果として生じる出力ベクトルデータサンプルセット296(0)~296(Z)を供給するように構成される。

【0188】

[00233]この実施形態における出力データフローパス98(0)~98(X)内のベクトルデータファイル82(0)~82(X)にマージされた、結果として生じる出力ベクトルデータサンプルセット296(0)~296(Z)を供給するための、図31のVPE22(6)のさらなる詳細および特徴のさらなる説明が次に記載される。この関連で、図32は、結果として生じる出力ベクトルデータサンプル292のマージングを必要とする例示的なベクトル命令に従って、マージング回路294を利用する図31のVPE22(6)において実行され得るベクトル処理動作302から得られた、結果として生じる出力ベクトルデータサンプルセット292(0)~292(X)の、結果として生じる出力ベクトルデータサンプル292の例示的なマージングを示すフローチャートである。

【0189】

[00234]図31と図32とを参照すると、ベクトル命令に従うベクトル処理動作302に従って処理されるべき入力ベクトルデータサンプルセット86(0)~86(X)が、ベクトルデータファイル82(0)~82(X)からフェッチされ、入力データフローパス80(0)~80(X)内に供給される(図32のブロック304)。ベクトル処理動作302のための入力ベクトルデータサンプルセット86(0)~86(X)の幅に応じて、ベクトル命令のプログラミングに従うベクトル処理動作302を提供するために、図31のVPE22(6)内のベクトルデータレーン100(0)~100(X)の1つ、いくつか、またはすべてが利用され得る。ベクトルデータファイル82(0)~82(X)の幅全体が必要な場合、すべてのベクトルデータレーン100(0)~100(X)がベクトル処理動作302に利用され得る。ベクトル処理動作302は、ベクトルデータレーン100(0)~100(X)のサブセットを必要とするにすぎない場合がある。これは、入力ベクトルデータサンプルセット86(0)~86(X)の幅がすべてのベクトルデータファイル82(0)~82(X)の幅よりも小さいからであり得るし、ここで、ベクトル処理動作302と並列に実行されるべき他のベクトル処理動作にさらなるベクトルデータレーン100を利用することが望ましい。

【0190】

[00235]引き続き図31と図32とを参照すると、フェッチされた入力ベクトルデータサンプルセット86(0)~86(X)が、実行ユニット84(0)~84(X)にある入力データフローパス80(0)~80(X)から受信される(図32のブロック306)。実行ユニット84(0)~84(X)が、ベクトル命令に従って提供されたベクトル処理動作302に従って、受信された入力ベクトルデータサンプルセット86(0)~86(X)に対してベクトル処理動作302を実行する(図32のブロック308)。実行ユニット84(0)~84(X)は、ベクトル処理動作302が、結果として生じる出力ベクトルデータサンプルセット292(0)~292(X)を供給するために、入力ベク

トルデータサンプルセット 86(0) ~ 86(X) を使用して、乗算および / または累算を提供することができる。ベクトル処理動作 302 が完了すると、入力ベクトルデータサンプルセット 86(0) ~ 86(X) 上で遂行されたベクトル処理動作 302 に基づく、結果として生じる出力ベクトルデータサンプルセット 292(0) ~ 292(X) は、図 31 の VPE22(6) の出力データフローパス 98(0) ~ 98(X) 内に供給される。

【0191】

[00236] 引き続き図 31 と図 32 とを参照すると、結果として生じる出力ベクトルデータサンプルセット 292(0) ~ 292(X) がベクトルデータファイル 82(0) ~ 82(X) に記憶される前に、結果として生じる出力ベクトルデータサンプルセット 292(0) ~ 292(X) は、実行ユニット 84(0) ~ 84(X) とベクトルデータファイル 82(0) ~ 82(X) との間に設けられた出力データフローパス 98(0) ~ 98(X) 内に設けられたマーキング回路 294 に供給される。マーキング回路 294 は、実行されているベクトル命令に従って、および下記でより詳細に説明されるように、ベクトル命令がベクトルデータファイル 82(0) ~ 82(X) に記憶されるべき、結果として生じる出力ベクトルデータサンプルセット 292(0) ~ 292(X) からの、結果として生じる出力ベクトルデータサンプル 292 のマーキングを要求する場合、出力データフローパス 98(0) ~ 98(X) に含まれるようにプログラム可能である。結果として生じる出力ベクトルデータサンプルセット 292(0) ~ 292(X) がベクトルデータファイル 82(0) ~ 82(X) に記憶されることなく、マーキング回路 294 が、実行されているベクトル命令に従って、結果として生じる出力ベクトルデータサンプルセット 292(0) ~ 292(X) からの、結果として生じる出力ベクトルデータサンプル 292 をマージする (図 32 のブロック 310)。このようにして、結果として生じる出力ベクトルデータサンプルセット 292(0) ~ 292(X) は、それにより実行ユニット 84(0) ~ 84(X) において遅延をもたらす、最初にベクトルデータファイル 82(0) ~ 82(X) に記憶され、再フェッチされ、後処理動作においてマージされ、ベクトルデータファイル 82(0) ~ 82(X) にマージされたフォーマットで記憶される必要がない。結果として生じる出力ベクトルデータサンプルセット 292(0) ~ 292(X) が、マージ後処理を必要とせず、ベクトルデータファイル 82(0) ~ 82(X) にマージされた、結果として生じる出力ベクトルデータサンプルセット 296(0) ~ 296(Z) として記憶される (図 32 のブロック 312)。

【0192】

[00237] 図 33 は、図 31 の VPE22(6) 内の実行ユニット 84(0) ~ 84(X) とベクトルデータファイル 82(0) ~ 82(X) との間の出力データフローパス 98(0) ~ 98(X) 内に設けられ得る例示的なマーキング回路 294 の概略図である。マーキング回路 294 は、マージされた、結果として生じる出力ベクトルデータサンプルセット 296(0) ~ 296(Z) を供給するために、結果として生じる出力ベクトルデータサンプルセット 292(0) ~ 292(X) のマーキングを提供するように構成される。結果として生じる出力ベクトルデータサンプルセット 292(0) ~ 292(X) は、図 31 に示されたように、実行ユニット出力 96(0) ~ 96(X) からマーキング回路 294 に供給される。

【0193】

[00238] 引き続き図 33 を参照すると、マーキング回路 294 は、結果として生じる出力ベクトルデータサンプルセット 292(0) ~ 292(X) をマージするように構成される。この実施形態におけるマーキング回路 294 は、マージされた、結果として生じる出力ベクトルデータサンプルセット 296(0) ~ 296(Z) を供給するように構成される。この関連で、マーキング回路 294 は、結果として生じる出力ベクトルデータサンプルセット 292(0) ~ 292(X) を受信するために、実行ユニット出力 96(0) ~ 96(X) に結合された加算器ツリー 318 を含んでいる。マーキング回路 294 の加算器ツリー 318 は、それらのそれぞれのベクトルデータレーン 100(0) ~ 100(X) に

X) 内で、結果として生じる出力ベクトルデータサンプルセット 292(0) ~ 292(X) の各サンプル 292 を受信するように構成される。加算器ツリー 318 内に第 1 の加算器ツリーレベル 318(1) が設けられる。第 1 の加算器ツリーレベル 318(1) は、結果として生じる出力ベクトルデータサンプルセット 292(0) ~ 292(X) 内の隣接サンプル 292 をマージすることができるように、マージ回路 320(0) ~ 320((X+1)/2 - 1)、320(7) から構成される。出力データフローパス 98(0) ~ 98(X) から、結果として生じる出力ベクトルデータサンプルセット 292(0) ~ 292(X) をラッチするために、ラッチ 321(0) ~ 321(X) がマージング回路 294 内に設けられる。

【0194】

10

[00239]たとえば、結果として生じる出力ベクトルデータサンプルセット 292(0) ~ 292(X) 内の各サンプル 292 が 32 ビット幅であり、2 つの 16 ビット複素数のベクトルデータ (すなわち、フォーマット I8Q8 に従う第 1 のベクトルデータおよびフォーマット I8Q8 に従う第 2 のベクトルデータ) から構成される場合、結果として生じる出力ベクトルデータサンプルセット 292(0) ~ 292(X) 内の 2 つの、結果として生じる出力ベクトルデータサンプル 292 の中の 4 つのベクトルデータサンプルを、1 つのマージされた、結果として生じる出力ベクトルデータサンプル 296 の中にマージするために、マージング動作が適用される可能性がある。たとえば、図 33 に示されたように、加算器 320(0) は、結果として生じる出力ベクトルデータサンプル 292(0) と 292(1) とをマージするように構成される。同じく、加算器 320(1) は、それらのサンプルのための、結果として生じる出力ベクトルデータサンプル 292(2) と 292(3) とをマージするように構成される。加算器 320((X+1)/2 - 1)、320(7) は、マージベクトルデータサンプルセット 322(0) ~ 322((X+1)/2 - 1)、322(7) を供給するために、結果として生じる出力ベクトルデータサンプルセット 292(X-1) と 292(X) とをマージするように構成される。加算器 320((X+1)/2 - 1)、320(7) によって実行されたマージングからのマージベクトルデータサンプルセット 322(0) ~ 322((X+1)/2 - 1)、322(7) は、ラッチ 325(0) ~ 325((X+1)/2 - 1)、325(7) の中にラッチされる。

20

【0195】

30

[00240]マージベクトル処理動作 302 が、結果として生じる出力ベクトルデータサンプルセット 292(0) ~ 292(X) のマージングを必要とする場合、下記でより詳細に記載されるように、マージベクトルデータサンプルセット 322(0) ~ 322((X+1)/2 - 1)、322(7) は、マージ、結果として生じる出力ベクトルデータサンプルセット 296(0) ~ 296(Z) として供給され得るし、ここで、「Z」は 7 である。しかしながら、マージベクトル処理動作 302 が、結果として生じる出力ベクトルデータサンプルセット 292(0) ~ 292(X) 内の隣接しない、結果として生じる出力ベクトルデータサンプル 292 のマージングを要求する場合、マージベクトルデータサンプルセット 322(0) ~ 322((X+1)/2 - 1)、322(7) は、マージ、結果として生じる出力ベクトルデータサンプルセット 296(0) ~ 296(Z) として供給されない。マージベクトルデータサンプルセット 322(0) ~ 322((X+1)/2 - 1)、322(7) は、加算器 324(0) ~ 324((X+1)/4 - 1)、324(3) への第 2 の加算器ツリーレベル 318(2) に供給される。この関連で、加算器 324(0) は、結果として生じるマージベクトルデータサンプル 326(0) を供給するために、マージベクトルデータサンプル 322(0) および 322(1) に対してマージングを実行するように構成される。同じく、加算器 324(1) は、結果として生じるマージベクトルデータサンプル 326(1) を供給するために、マージベクトルデータサンプル 322(2) および 322(3) に対してマージングを実行するように構成される。加算器 324((X+1)/4 - 1)、324(3) は、結果として生じるマージベクトルデータサンプル 326((X+1)/4 - 1)、326(

40

50

3) を供給するために、マージベクトルデータサンプル 3 2 2 (((X + 1) / 4) - 2)、3 2 2 (((X + 1) / 4) - 1)、3 2 2 (3) に対してマージングを実行するように構成される。加算器 3 2 4 (0) ~ 3 2 4 (((X + 1) / 4) - 1)、3 2 4 (3) によって実行されたマージングからの、結果として生じるマージベクトルデータサンプルセット 3 2 6 (0) ~ 3 2 6 (((X + 1) / 4) - 1)、3 2 6 (3) は、ラッチ 3 2 7 (0) ~ 3 2 7 (((X + 1) / 4) - 1)、3 2 7 (3) の中にラッチされる。

【 0 1 9 6 】

[00241]引き続き図 3 3 を参照すると、マージベクトル処理動作 3 0 2 が 8 のマージ係数による、結果として生じる出力ベクトルデータサンプルセット 2 9 2 (0) ~ 2 9 2 (X) のマージングを必要とする場合、下記でより詳細に記載されるように、マージベクトルデータサンプルセット 3 2 6 (0) ~ 3 2 6 (((X + 1) / 4) - 1)、3 2 6 (3) は、マージ、結果として生じる出力ベクトルデータサンプルセット 2 9 6 (0) ~ 2 9 6 (Z) として供給され得るし、ここで、「Z」は 3 である。しかしながら、マージベクトル処理動作 3 0 2 が 8 よりも高いマージ係数 (たとえば、1 6、3 2、6 4、1 2 8、2 5 6) を要求する場合、マージベクトルデータサンプルセット 3 2 6 (0) ~ 3 2 6 (((X + 1) / 4) - 1)、3 2 6 (3) は、マージ、結果として生じる出力ベクトルデータサンプルセット 2 9 6 (0) ~ 2 9 6 (Z) として供給されない。マージベクトルデータサンプルセット 3 2 6 (0) ~ 3 2 6 (((X + 1) / 4) - 1)、3 2 6 (3) は、加算器 3 2 8 (0) ~ 3 2 8 (((X + 1) / 8) - 1)、3 2 8 (1) への第 3 の加算器ツリーレベル 3 1 8 (3) に供給される。この関連で、加算器 3 2 8 (0) は、それらのサンプルのための 1 6 のマージ係数を供給するために、マージベクトルデータサンプル 3 2 6 (0) および 3 2 6 (1) に対してマージングを実行するように構成される。同じく、加算器 3 2 8 (1) は、1 6 のマージ係数を有するマージベクトルデータサンプルセット 3 3 0 (0) ~ 3 3 0 (((X + 1) / 8) - 1)、3 3 0 (1) を供給するために、マージベクトルデータサンプル 3 2 6 (2) および 3 2 6 (3) に対してマージングを実行するように構成される。加算器 3 2 8 (0) ~ 3 2 8 (((X + 1) / 8) - 1)、3 2 8 (1) によって実行されたマージングからのマージベクトルデータサンプルセット 3 3 0 (0) ~ 3 3 0 (((X + 1) / 8) - 1)、3 3 0 (1) は、ラッチ 3 2 9 (0) ~ 3 2 9 (((X + 1) / 8) - 1)、3 2 9 (1) の中にラッチされる。

【 0 1 9 7 】

[00242]引き続き図 3 3 を参照すると、マージベクトル処理動作 3 0 2 が 1 6 のマージ係数による、結果として生じる出力ベクトルデータサンプルセット 2 9 2 (0) ~ 2 9 2 (X) のマージングを必要とする場合、下記でより詳細に記載されるように、マージベクトルデータサンプルセット 3 3 0 (0) ~ 3 3 0 (((X + 1) / 8) - 1)、3 3 0 (1) は、マージ、結果として生じる出力ベクトルデータサンプルセット 2 9 6 (0) ~ 2 9 6 (Z) として供給され得るし、ここで、「Z」は 1 である。しかしながら、マージベクトル処理動作 2 3 6 が 1 6 よりも高いマージ係数 (たとえば、3 2、6 4、1 2 8、2 5 6) を要求する場合、マージベクトルデータサンプルセット 3 3 0 (0) ~ 3 3 0 (((X + 1) / 8) - 1)、3 3 0 (1) は、マージ、結果として生じる出力ベクトルデータサンプルセット 2 9 6 (0) ~ 2 9 6 (Z) として供給されない。マージベクトルデータサンプルセット 3 3 0 (0) ~ 3 3 0 (((X + 1) / 8) - 1)、3 3 0 (1) は、加算器 3 3 2 への第 4 の加算器ツリーレベル 3 1 8 (4) に供給される。この関連で、加算器 3 3 2 は、3 2 のマージ係数を有するマージベクトルデータサンプル 3 3 4 を供給するために、マージベクトルデータサンプル 3 3 0 (0) および 3 3 0 (1) に対してマージングを実行するように構成される。加算器 3 3 2 によって実行されたマージングからのマージベクトルデータサンプル 3 3 4 は、ラッチ 3 3 6 および 3 3 8 の中にラッチされる。

【 0 1 9 8 】

[00243]引き続き図 3 3 を参照すると、マージベクトル処理動作 3 0 2 が 3 2 のマージ係数による、結果として生じる出力ベクトルデータサンプルセット 2 9 2 (0) ~ 2 9 2

(X)のマーキングを必要とする場合、下記でより詳細に記載されるように、マージベクトルデータサンプル334は、マージ、結果として生じる出力ベクトルデータサンプル296として供給され得る。しかしながら、マージベクトル処理動作302が32よりも高いマージ係数(たとえば、64、128、256)を要求する場合、マージベクトルデータサンプル334は、マージ、結果として生じる出力ベクトルデータサンプル296として供給されない。マージベクトルデータサンプル334は、ベクトルデータファイル82に記憶される必要なしに、ラッチ338の中にラッチされたままである。上述されたように、32のマージ係数を使用してマージされるために、別の結果として生じる出力ベクトルデータサンプルセット292(0)~292(X)は、さらなる処理サイクルにわたって、ラッチ321(0)~321(X)の中にロードされる。結果として生じるマージベクトルデータサンプル334'は、64のマージ係数を有するマージベクトルデータサンプル342を供給するために、第5の加算器ツリー318(5)内の加算器340により、前のマージベクトルデータサンプル334に加算される。選択器343は、32のマージ係数を有するマージベクトルデータサンプル334、または64のマージ係数を有するマージベクトルデータサンプル334'のどちらが、マージベクトルデータサンプル342としてラッチ344の中にラッチされるかを制御する。さらなる結果として生じる出力ベクトルデータサンプルセット292(0)~292(X)をラッチし、さらなる結果として生じる出力ベクトルデータサンプルセット292(0)~292(X)をマージするこの同じプロセスは、必要な場合、64よりも大きいマージ係数を達成するために実行され得る。マージベクトルデータサンプル342は、最終的に、マージベクトル処理動作302のための所望のマージ係数に従って、所望のマージ、結果として生じる出力ベクトルデータサンプル296として、ラッチ344の中にラッチされる。

【0199】

[00244]引き続き図33を参照すると、マージベクトル処理動作302においてどのマージ係数が要求されても、マージ、結果として生じる出力ベクトルデータサンプルセット296(0)~296(Z)は、ベクトルデータファイル82(0)~82(X)に記憶される必要がある。次に説明されるように、図33のマーキング回路294はまた、マージ、結果として生じる出力ベクトルデータサンプルセット296(0)~296(Z)を形成するために、結果として生じる出力ベクトルデータサンプルセット292(0)~292(X)に対してマージベクトル処理動作302を実行する結果としてもたらされたマージ、結果として生じる出力ベクトルデータサンプル296をラッチ346(0)~346(X)の中にロードするように構成される。マージ、結果として生じる出力ベクトルデータサンプルセット296(0)~296(Z)は、記憶されるためにベクトルデータファイル82(0)~82(X)に供給され得る。このようにして、マーキング回路294によって作成されたマージ、結果として生じる出力ベクトルデータサンプルセット296(0)~296(Z)を記憶するために、ベクトルデータファイル82(0)~82(X)に対して1回の書込みが必要とされるにすぎない。図33のマーキング回路294内の加算器ツリー318(1)~318(5)は、マージベクトル処理動作302においてどのマージ係数が要求されても、マージ係数4、8、16、および32のすべてに対して、マージ、結果として生じる出力ベクトルデータサンプル296を生成することができる。代替として、所望のマージ係数に従ってマージベクトル処理動作302を実行する必要がある加算器ツリー内の加算器は、無効にされ得るか、または0を加算するように構成され得る。しかしながら、これらのマージ、結果として生じる出力ベクトルデータサンプル296のどれが記憶されるためにラッチ346(0)~346(X)に供給されるかを決定するために、次に説明されるように、選択器348(0)~348((X+1)/4)-1)、348(3)が設けられる。

【0200】

[00245]この関連で、引き続き図33を参照すると、選択器348(0)は、実行されているマージベクトル処理動作302に基づいて、それぞれ、加算器320(0)、324(0)、328(0)からのマージ係数4、8、および16、ならびに加算器332、

340からのマージ係数32、64、128、256のいずれかに対して、マージ、結果として生じる出力ベクトルデータサンプル296を選択することができる。選択器348(1)は、実行されているマージベクトル処理動作302に基づいて、それぞれ、加算器320(1)、324(1)、328(1)からのマージ係数4、8、および16に対して、マージ、結果として生じる出力ベクトルデータサンプル296を選択することができる。選択器348(2)は、実行されているマージベクトル処理動作302に基づいて、それぞれ、加算器320(2)および324(2)からのマージ係数4および8に対して、マージ、結果として生じる出力ベクトルデータサンプル296を選択することができる。選択器348(3)は、実行されているマージベクトル処理動作302に基づいて、それぞれ、加算器320(3)および324(3)からのマージ係数4および8に対して、マージ、結果として生じる出力ベクトルデータサンプル296を選択することができる。8のマージ係数を供給することが選択器348(0)~348(3)によって完全に満足され得るので、選択器は、加算器320(4)~320(7)から供給されたマージ、結果として生じる出力ベクトルデータサンプル296を制御するためには設けられない。

【0201】

[00246]引き続き図33を参照すると、マージベクトル処理動作に設けられたデータスライサ350(0)~350(($(X+1)/2$)-1)、350(7)は、バイパスされるか、または、それぞれ、選択器348(0)~348(($(X+1)/4$)-1)、348(3)および加算器320(4)~320(($(X+1)/2$)-1)、320(7)によって選択された、受信されたマージ、結果として生じる出力ベクトルデータサンプル296に対してデータスライシングを実行しないように構成される可能性がある。マージ、結果として生じる出力ベクトルデータサンプル296は、次いで、クロスバー352への接続を介して、記憶されるためにラッチ346(0)~346(X)の中の所望のラッチ346に転送される。クロスバー352は、様々なラッチ346(0)~346(X)に、マージベクトル処理動作302に従ってマージ、結果として生じる出力ベクトルデータサンプル296を供給する柔軟性を提供する。このようにして、マージ、結果として生じる出力ベクトルデータサンプル296は、ベクトルデータファイル82(0)~82(X)に記憶される前に、マージベクトル処理動作302の様々な繰返しの中で、ラッチ346(0)~346(X)にスタックされ得る。たとえば、マージ、結果として生じる出力ベクトルデータサンプルセット296(0)~296(Z)は、ベクトルデータファイル82(0)~82(X)に記憶される前に、マージベクトル処理動作302の様々な繰返しの中で、ラッチ346(0)~346(X)にスタックされ得る。このようにして、マージ、結果として生じる出力ベクトルデータサンプルセット296(0)~296(Z)を記憶するためのベクトルデータファイル82(0)~82(X)へのアクセスは、動作効率のために最小化され得る。

【0202】

[00247]たとえば、図33に示されたように、クロスバー352に結合された選択器354(0)~354(X)は、ラッチ346(0)~346(X)のいずれかの中に選択器348(0)からのマージ、結果として生じる出力ベクトルデータサンプル296を記憶するように制御され得る。クロスバー352に結合された選択器354(1)、354(3)、354(5)、354(7)、354(9)、354(11)、354(13)、354(15)は、ラッチ346(1)、346(3)、346(5)、346(7)、346(9)、346(11)、346(13)、および346(15)に記憶されるべき選択器348(1)からのマージ、結果として生じる出力ベクトルデータサンプル296を記憶するように制御され得る。クロスバー352に結合された選択器354(2)、354(6)、354(10)、354(14)は、ラッチ346(2)、346(6)、346(10)、および346(14)に選択器348(2)からのマージ、結果として生じる出力ベクトルデータサンプル296を記憶するように制御され得る。クロスバー352に結合された選択器354(3)、354(7)、354(11)、354(15)は、ラッチ346(3)、346(7)、346(11)、および346(15)に

選択器 3 4 8 (3) からのマージ、結果として生じる出力ベクトルデータサンプル 2 9 6 を記憶するように制御され得る。クロスバー 3 5 2 に結合された選択器 3 5 4 (4) および 3 5 4 (1 2) は、ラッチ 3 4 6 (4) および 3 4 6 (1 2) に加算器 3 2 0 (4) からのマージ、結果として生じる出力ベクトルデータサンプル 2 9 6 を記憶するように制御され得る。クロスバー 3 5 2 に結合された選択器 3 5 4 (5) および 3 5 4 (1 3) は、ラッチ 3 4 6 (5) および 3 4 6 (1 3) に加算器 3 2 0 (5) からのマージ、結果として生じる出力ベクトルデータサンプル 2 9 6 を記憶するように制御され得る。クロスバー 3 5 2 に結合された選択器 3 5 4 (6) および 3 5 4 (1 4) は、ラッチ 3 4 6 (6) または 3 4 6 (1 4) に加算器 3 2 0 (6) からのマージ、結果として生じる出力ベクトルデータサンプル 2 9 6 を記憶するように制御され得る。クロスバー 3 5 2 に結合された選択器 3 5 4 (7) および 3 5 4 (1 5) は、ラッチ 3 4 6 (7) または 3 4 6 (1 5) に加算器 3 2 0 (7) からのマージ、結果として生じる出力ベクトルデータサンプル 2 9 6 を記憶するように制御され得る。

10

【 0 2 0 3 】

[00248] 図 3 3 のマージング回路 2 9 4 では、加算器は、結果として生じる出力ベクトルデータサンプルセット 2 9 2 (0) ~ 2 9 2 (X) 内の隣接しない、結果として生じる出力ベクトルデータサンプル 2 8 2 がマージされることを可能にするように構成される可能性があることに留意されたい。たとえば、結果として生じる出力ベクトルデータサンプル 2 9 2 (0) を、結果として生じる出力ベクトルデータサンプル 2 9 2 (9) とマージすることが望ましい場合、加算器ツリーレベル 3 1 8 (1) ~ 3 1 8 (3) 内の加算器は、単に、結果として生じる出力ベクトルデータサンプル 2 9 2 (9) との、結果として生じる出力ベクトルデータサンプル 2 9 2 (0) のマージを、加算器ツリーレベル 3 1 8 (4) に渡すように構成される可能性がある。加算器ツリーレベル 3 1 8 (4) 内の加算器 3 3 2 は、次いで、マージされた出力ベクトルデータサンプル 2 9 6 を供給するために、結果として生じる出力ベクトルデータサンプル 2 9 2 (0) を、結果として生じる出力ベクトルデータサンプル 2 9 2 (9) とマージする可能性がある。

20

【 0 2 0 4 】

[00249] ベクトルおよび / またはスカラーの加算以外の他のタイプのベクトルマージング演算を提供するマージング回路も、実行ユニット 8 4 (0) ~ 8 4 (X) とベクトルデータファイル 8 2 (0) ~ 8 2 (X) との間の出力データフローパス 9 8 (0) ~ 9 8 (X) 内に設けられる可能性がある。たとえば、図 3 3 のマージング回路 2 9 4 は、最大または最小のベクトルおよび / またはスカラーのマージング演算を提供するように構成される可能性がある。たとえば、図 3 3 の加算器ツリー 3 1 8 の加算器ツリーレベル 3 1 8 (1) ~ 3 1 8 (5) 内の加算器は、最大または最小の関数回路と交換される可能性がある。言い換えれば、回路は、結果として生じる出力ベクトルデータサンプルセット 2 9 2 (0) ~ 2 9 2 (X) からの 2 つの、結果として生じる出力ベクトルデータサンプル 2 9 2 のうちの大きい方または小さい方のいずれかを渡すことを選択するはずである。たとえば、結果として生じる出力ベクトルデータサンプルセット 2 9 2 (0) ~ 2 9 2 (X) からの 2 つの、結果として生じる出力ベクトルデータサンプル 2 9 2 が、図 3 0 における 2 つの入力ベクトルデータサンプル 2 9 0 (0)、2 9 0 (1) であった場合、マージング回路 2 9 4 が最大ベクトルデータサンプルを選択するように構成されている場合、マージング回路 2 9 4 は、ベクトルデータサンプル 2 9 0 (1) を選択するように構成される可能性がある。

30

40

【 0 2 0 5 】

[00250] この関連で、図 3 4 を参照すると、図 3 3 の第 1 の加算器ツリーレベル 3 1 8 (1) 内の加算器 3 2 0 (0) ~ 3 2 0 (((X + 1) / 2) - 1)、3 2 0 (7) は、図 3 4 に示されたように、最大または最小のマージ選択加算器 3 2 0 ' (0) ~ 3 2 0 ' (((X + 1) / 2) - 1)、3 2 0 ' (7) と交換される可能性がある。第 2 の加算器ツリーレベル 3 1 8 (2) 内の加算器 3 2 4 (0) ~ 3 2 4 (((X + 1) / 4) - 1)、3 2 4 (3) は、図 3 4 に示されたように、最大または最小の選択器 3 2 4 ' (0) ~

50

3 2 4' (((X + 1) / 4) - 1)、3 2 4' (3) と交換される可能性がある。第3の加算器ツリーレベル3 1 8 (3) 内の加算器3 2 8 (0) ~ 3 2 8 (((X + 1) / 8) - 1)、3 2 8 (1) は、図3 4 に示されたように、最大または最小の選択器3 2 8' (0) ~ 3 2 8' (((X + 1) / 8) - 1)、3 2 8' (1) と交換される可能性がある。第4の加算器ツリーレベル3 1 8 (4) 内の加算器3 3 2 は、図3 4 に示されたように、最大または最小の選択器3 3 2' と交換される可能性がある。第5の加算器ツリーレベル3 1 8 (5) 内の加算器3 4 0 は、図3 4 に示されたように、最大または最小の選択器3 4 0' と交換される可能性がある。図3 4 のマージング回路2 9 4 では、加算器は、マージされるべき、結果として生じる出力ベクトルデータサンプルセット2 9 2 (0) ~ 2 9 2 (X) 内の隣接しない、結果として生じる出力ベクトルデータサンプル2 9 2 の間の、最大または最小の、結果として生じる出力ベクトルデータサンプル2 9 2 を選択するように構成される可能性がある。たとえば、結果として生じる出力ベクトルデータサンプル2 9 2 (0) を、結果として生じる出力ベクトルデータサンプル2 9 2 (9) と最大マージすることが望ましい場合、加算器ツリーレベル3 1 8 (1) ~ 3 1 8 (3) 内の加算器は、単に、結果として生じる出力ベクトルデータサンプル2 9 2 (9) との、結果として生じる出力ベクトルデータサンプル2 9 2 (0) のマージを、加算器ツリーレベル3 1 8 (4) に渡すように構成される可能性がある。加算器ツリーレベル3 1 8 (4) 内の加算器3 3 2' は、次いで、マージされた出力ベクトルデータサンプル2 6 4 を供給するために、結果として生じる出力ベクトルデータサンプル2 9 2 (0) を、結果として生じる出力ベクトルデータサンプル2 9 2 (9) と最大マージする可能性がある。

【 0 2 0 6 】

[00251] 上記で説明されたように、入力ベクトルデータサンプルセット8 6 (0) ~ 8 6 (X) に対してベクトル処理動作を実行するために、V P E 2 2 (1) ~ (6) 内に実行ユニット8 4 (0) ~ 8 4 (X) が設けられる。実行ユニット8 4 (0) ~ 8 4 (X) は、実行ユニット8 4 (0) ~ 8 4 (X) が様々なベクトル処理動作のための共通回路とハードウェアとを用いて複数の動作モードを提供することを可能にする、プログラム可能なデータパス構成も含む。実行ユニット8 4 (0) ~ 8 4 (X) および共通回路とハードウェアとを用いて複数の動作モードを提供するためのそれらのプログラム可能なデータパス構成に関するより例示的な詳細が次に説明される。

【 0 2 0 7 】

[00252] この関連で、図3 5 は、V P E 2 2 (1) ~ (6) 内の実行ユニット8 4 (0) ~ 8 4 (X) の各々のために提供され得る、例示的な実行ユニットの例示的な概略図を示す。図3 5 に示されたように、および図3 6 ~ 図3 9 において以下でより詳細に記載されるように、実行ユニット8 4 は、プログラム可能なデータパス構成を用いて構成され得る例示的なベクトル処理ブロックを有する、複数の例示的なベクトルパイプラインステージ4 6 0 を含む。下記でより詳細に説明されるように、ベクトル処理ブロック内に設けられたプログラム可能なデータパス構成により、特定の回路およびハードウェアが、図2 のベクトルユニットデータメモリ3 2 から受信されたベクトルデータ3 0 に対する異なる特定のベクトル処理動作の実行をサポートするようにプログラムおよび再プログラムされることが可能になる。

【 0 2 0 8 】

[00253] たとえば、いくつかのベクトル処理動作は、通常、ベクトルデータ3 0 の乗算、続いて、乗算されたベクトルデータ結果の累算を必要とする場合がある。そのようなベクトル処理の非限定的な例には、ワイヤレス通信アルゴリズムのための高速フーリエ変換 (F F T) 演算を実行するために通常使用される、フィルタリング演算、相関演算、ならびに基数2 および基数4 のパタフライ演算が含まれ、ここで、一連の並列乗算が、続いて乗算結果の一連の並列累算が提供される。同様に図3 9 および図4 0 に関して下記でより詳細に説明されるように、図3 5 の実行ユニット8 4 は、桁上げ保存累算器において冗長桁上げ保存フォーマットを提供するための桁上げ保存累算器を有する融合乗算器のオプションも有する。桁上げ保存累算器において冗長桁上げ保存フォーマットを提供することに

より、桁上げ伝搬パスと、累算の各ステップの間の桁上げ伝搬加算演算とを提供する必要をなくすることができる。

【0209】

[00254]この関連で、図35をさらに参照すると、VPE22のM0乗算ベクトルパイプラインステージ460(1)が最初に記載される。M0乗算ベクトルパイプラインステージ460(1)は、各々がプログラム可能なデータパス構成を有する、任意の所望の数の乗算器ブロック462(A)~462(0)の形式で複数のベクトル処理ブロックを含んでいる第2のベクトルパイプラインステージである。乗算器ブロック462(A)~462(0)は、実行ユニット84内でベクトル乗算演算を実行するために設けられる。複数の乗算器ブロック462(A)~462(0)は、最大12個の乗算ベクトルデータサンプルセット34(Y)~34(0)の乗算を提供するために、M0乗算ベクトルパイプラインステージ460(1)内で互いと並列に配置される。この実施形態では、「A」は3に等しく、この例では、M0乗算ベクトルパイプラインステージ460(1)に4つの乗算器ブロック462(3)~462(0)が含まれることを意味する。乗算ベクトルデータサンプルセット34(Y)~34(0)は、実行ユニット84内の第1のベクトルパイプラインステージ460(0)である、入力読取り(RR)ベクトルパイプラインステージ内に設けられた複数のラッチ464(Y)~464(0)へのベクトル処理のための実行ユニット84の中にロードされる。この実施形態では実行ユニット84内に12個のラッチ464(11)~464(0)があり、この実施形態では「Y」が11に等しいことを意味する。ラッチ464(11)~464(0)は、ベクトルレジスタ(図2のベクトルデータファイル28参照)から取り出された乗算ベクトルデータサンプルセット34(11)~34(0)を、ベクトルデータ入力サンプルセット466(11)~466(0)としてラッチするように構成される。この例では、各ラッチ464(11)~464(0)は8ビット幅である。ラッチ464(11)~464(0)は、各々、それぞれ乗算ベクトルデータ入力サンプルセット466(11)~466(0)を、総計96ビット幅のベクトルデータ30(すなわち、12ラッチ×各8ビット)を求めてラッチするように構成される。

【0210】

[00255]引き続き図35を参照すると、複数の乗算器ブロック462(3)~462(0)は、ベクトル乗算演算を提供するために、ベクトルデータ入力サンプルセット466(11)~466(0)のいくつかの組合せを受信することができるように構成され、ここで、この例では「Y」は11に等しい。乗算ベクトルデータ入力サンプルセット466(11)~466(0)は、実行ユニット84の設計に従って、複数の入力データパスA3~A0、B3~B0、およびC3~C0内で供給される。ベクトルデータ入力サンプルセット466(3)~466(0)は、図35に示されたように、入力データパスC3~C0に対応する。ベクトルデータ入力サンプルセット466(7)~466(4)は、図35に示されたように、入力データパスB3~B0に対応する。ベクトルデータ入力サンプルセット466(11)~466(8)は、図35に示されたように、入力データパスA3~A0に対応する。複数の乗算器ブロック462(3)~462(0)は、ベクトル乗算演算を提供するために、それぞれ、複数の乗算器ブロック462(3)~462(0)に供給された、入力データパスA3~A0、B3~B0、C3~C0に従って受信されたベクトルデータ入力サンプルセット466(11)~466(0)を処理するように構成される。

【0211】

[00256]図37および図38に関して下記でより詳細に説明されるように、図35の乗算器ブロック462(3)~462(0)内に設けられたプログラム可能な内部データパス467(3)~467(0)は、様々なデータパス構成を有するようにプログラムされ得る。これらの様々なデータパス構成は、各乗算器ブロック462(3)~462(0)に供給された、特定の入力データパスA3~A0、B3~B0、C3~C0に従って乗算器ブロック462(3)~462(0)に供給された、特定の受信されたベクトルデータ

入力サンプルセット466(11)~466(0)の様々な組合せおよび/または様々なビット長の乗算を提供する。この関連で、複数の乗算器ブロック462(3)~462(0)は、ベクトルデータ入力サンプルセット466(11)~466(0)の特定の組合せと一緒に乗算した乗算結果を備えるベクトル結果出力サンプルセットとして、ベクトル乗算出力サンプルセット468(3)~468(0)を供給する。

【0212】

[00257]たとえば、乗算器ブロック462(3)~462(0)のプログラム可能な内部データパス467(3)~467(0)は、図2のベースバンドプロセッサ20の命令ディスパッチ回路48内のベクトル命令デコードから供給される設定値に従ってプログラムされ得る。この実施形態では、乗算器ブロック462(3)~462(0)の4つのプログラム可能な内部データパス467(3)~467(0)がある。ベクトル命令は、実行ユニット84によって実行されるべき特定のタイプの演算を指定する。したがって、実行ユニット84は、高効率な方式で同じ共通回路を用いて様々なタイプのベクトル乗算演算を提供するために、乗算器ブロック462(3)~462(0)のプログラム可能な内部データパス467(3)~467(0)を構成するようにプログラムおよび再プログラムされ得る。たとえば、実行ユニット84は、乗算器ブロック462(3)~462(0)のプログラム可能な内部データパス467(3)~467(0)を、命令ディスパッチ回路48内の命令パイプラインにおけるベクトル命令の復号に従って、実行されるベクトル命令ごとにクロックサイクルごとに構成および再構成するようにプログラムされ得る。したがって、実行ユニット84内のM0乗算ベクトルパイプラインステージ460(1)が、クロックサイクルごとにベクトルデータ入力サンプルセット466を処理するように構成されている場合、結果として、乗算器ブロック462(3)~462(0)は、命令ディスパッチ回路48内の命令パイプラインにおけるベクトル命令の復号に従って、クロックサイクルごとにベクトル乗算演算を実行する。

【0213】

[00258]乗算器ブロック462は、実数乗算と虚数乗算とを実行するようにプログラムされ得る。引き続き図35を参照すると、あるベクトル処理ブロックデータパス構成において、乗算器ブロック462は、2つの8ビットベクトルデータ入力サンプルセット466と一緒に乗算するように構成される場合がある。ある乗算ブロックデータパス構成では、乗算器ブロック462は、2つの16ビットベクトルデータ入力サンプルセット466と一緒に乗算するように構成される場合があり、これらは、8ビットベクトルデータ入力サンプルセット466の第2のペアと乗算された8ビットベクトルデータ入力サンプルセット466の第1のペアから形成される。これは図38に示され、下記でより詳細に説明される。やはり、乗算器ブロック462(3)~462(0)内にプログラム可能なデータパス構成を設けることにより、乗算器ブロック462(3)~462(0)が、実行ユニット84内の面積を削減し、場合によっては、所望のベクトル処理動作を遂行するためにベースバンドプロセッサ20内により少ない実行ユニット84が設けられることを可能にするために、様々なタイプの乗算演算を実行するように構成および再構成され得るという柔軟性がもたらされる。

【0214】

[00259]図35に戻って参照すると、複数の乗算器ブロック462(3)~462(0)は、プログラム可能な出力データパス470(3)~470(0)内のベクトル乗算出力サンプルセット468(3)~468(0)を、次のベクトル処理ステージ460または出力処理ステージのいずれかに供給するように構成される。ベクトル乗算出力サンプルセット468(3)~468(0)は、複数の乗算器ブロック462(3)~462(0)によって実行されているベクトル命令に基づいてプログラムされた構成に従って、プログラム可能な出力データパス470(3)~470(0)内で供給される。この例では、プログラム可能な出力データパス470(3)~470(0)内のベクトル乗算出力サンプルセット468(3)~468(0)は、下記で説明されるように、累算のためにM1累算ベクトルパイプラインステージ460(2)に供給される。実行ユニット84のこの

特定の設計では、複数の乗算器ブロック462(3)~462(0)と、続いて、ベクトルデータ入力の乗算、それに続く乗算結果の累算を要求する特殊なベクトル命令をサポートする累算器とを設けることが望ましい。たとえば、FFT演算を提供するために通常使用される、基数2および基数4のバタフライ演算は、一連の乗算演算、それに続く乗算結果の累算を含む。しかしながら、実行ユニット84内に設けられるベクトル処理ブロックのこれらの組合せは例示的であり、限定的でないことに留意されたい。プログラム可能なデータパス構成を有するVPEは、ベクトル処理ブロックを有する1つまたは他の任意の数のベクトル処理ステージを含むように構成される可能性がある。ベクトル処理ブロックは、設計および実行ユニットによってサポートされるように設計された特定のベクトル命令に従って、任意のタイプの演算を実行するために設けられる可能性がある。

10

【0215】

[00260]引き続き図35を参照すると、この実施形態では、ベクトル乗算出力サンプルセット468(3)~468(0)は、M1累算ベクトル処理ステージ460(2)である次のベクトル処理ステージ内に設けられた複数の累算器ブロック472(3)~472(0)に供給される。複数の累算器ブロック472(A)~472(0)の中の各累算器ブロックは、2つの累算器472(X)(1)および472(X)(0)(すなわち、472(3)(1)、472(3)(0)、472(2)(1)、472(2)(0)、472(1)(1)、472(1)(0)、および472(0)(1)、472(0)(0))を含んでいる。複数の累算器ブロック472(3)~472(0)は、ベクトル乗算出力サンプルセット468(3)~468(0)の結果を累算する。図39および図40 20
に関して下記でより詳細に説明されるように、複数の累算器ブロック472(3)~472(0)は桁上げ保存累算器として設けられ得るし、ここで、桁上げ積は本質的に、累算演算が完了されるまで累算プロセス中保存され、伝搬されない。複数の累算器ブロック472(3)~472(0)は、複数の累算器ブロック472(3)~472(0)において冗長桁上げ保存フォーマットを提供するために、図35および図37の複数の乗算器ブロック462(3)~462(0)と融合されるオプションも有する。複数の累算器ブロック472(3)~472(0)において冗長桁上げ保存フォーマットを提供することにより、複数の累算器ブロック472(3)~472(0)内の累算の各ステップの間に、桁上げ伝搬パスと桁上げ伝搬加算演算とを提供する必要をなくすることができる。M1累算ベクトル処理ステージ460(2)およびその複数の累算器ブロック472(3)~472(0)が、図35を参照して次に紹介される。 30

【0216】

[00261]図35を参照すると、M1累算ベクトル処理ステージ460(2)内の複数の累算器ブロック472(3)~472(0)は、累算器出力サンプルセット476(3)~476(0)(すなわち、476(3)(1)、476(3)(0)、476(2)(1)、476(2)(0)、476(1)(1)、476(1)(0)、および476(0)(1)、476(0)(0))を、次のベクトル処理ステージ460または出力処理ステージのいずれかにおいて供給するために、プログラム可能な出力データパス構成に従って、プログラム可能な出力データパス474(3)~474(0)(すなわち、474(3)(1)、474(3)(0)、474(2)(1)、474(2)(0)、474(1)(1)、474(1)(0)、および474(0)(1)、474(0)(0)) 40
内でベクトル乗算出力サンプルセット468(3)~468(0)を累算するように構成される。この例では、累算器出力サンプルセット476(3)~476(0)は、ALU処理ステージ460(3)である出力処理ステージに供給される。たとえば、下記でより詳細に説明されるように、累算器出力サンプルセット476(3)~476(0)はまた、非限定的な例として、図2のベースバンドプロセッサ20内のスカラープロセッサ44内のALU46に供給され得る。たとえば、ALU46は、より一般的な処理動作において使用されるために、実行ユニット84によって実行される特殊なベクトル命令に従って、累算器出力サンプルセット476(3)~476(0)を取る場合がある。

【0217】

50

[00262] 図 3 5 に戻って参照すると、累算器ブロック 4 7 2 (3) ~ 4 7 2 (0) のプログラム可能な入力データパス 4 7 8 (3) ~ 7 8 (0) および / またはプログラム可能な内部データパス 4 8 0 (3) ~ 4 8 0 (0) は、乗算器ブロック 4 6 2 (3) ~ 4 6 2 (0) から累算器ブロック 4 7 2 (3) ~ 4 7 2 (0) に供給された、様々な組合せおよび / またはビット長のベクトル乗算出力サンプルセット 4 6 8 (3) ~ 4 6 8 (0) を受信するように再構成されるようにプログラムされ得る。各累算器ブロック 4 7 2 は、2 つの累算器 4 7 2 (X) (1)、4 7 2 (X) (0) から構成されるので、プログラム可能な入力データパス 4 7 8 (A) ~ 4 7 8 (0) は、4 7 8 (3) (1)、4 7 8 (3) (0)、4 7 8 (2) (1)、4 7 8 (2) (0)、4 7 8 (1) (1)、4 7 8 (1) (0)、および 4 7 8 (0) (1)、4 7 8 (0) (0) として図 3 5 に示されている。同様に、プログラム可能な内部データパス 4 8 0 (3) ~ 4 8 0 (0) は、4 8 0 (3) (1)、4 8 0 (3) (0)、4 8 0 (2) (1)、4 8 0 (2) (0)、4 8 0 (1) (1)、4 8 0 (1) (0)、4 8 0 (0) (1)、4 8 0 (0) (0) として図 3 5 に示されている。累算器ブロック 4 7 2 (3) ~ 4 7 2 (0) 内にプログラム可能な入力データパス 4 7 8 (3) ~ 4 7 8 (0) および / またはプログラム可能な内部データパス 4 8 0 (3) ~ 4 8 0 (0) を設けることは、図 3 9 および図 4 0 に関して下記でより詳細に説明される。このようにして、累算器ブロック 4 7 2 (3) ~ 4 7 2 (0) のプログラム可能な入力データパス 4 7 8 (3) ~ 4 7 8 (0) および / またはプログラム可能な内部データパス 4 8 0 (3) ~ 4 8 0 (0) に従って、累算器ブロック 4 7 2 (3) ~ 4 7 2 (0) は、累算されたベクトル乗算出力サンプルセット 4 6 8 (3) ~ 4 6 8 (0) のプログラムされた組合せに従って、累算器出力サンプルセット 4 7 6 (3) ~ 4 7 6 (0) を供給することができる。やはり、これにより、累算器ブロック 4 7 2 (3) ~ 4 7 2 (0) が、実行ユニット 8 4 内の面積を削減し、場合によっては、所望のベクトル処理動作を遂行するために、ベースバンドプロセッサ 2 0 内により少ない実行ユニット 8 4 が設けられることを可能にするために、プログラム可能な入力データパス 4 7 8 (3) ~ 4 7 8 (0) および / またはプログラム可能な内部データパス 4 8 0 (3) ~ 4 8 0 (0) のプログラミングに基づいて、様々なタイプの累算演算を実行するように構成および再構成され得るという柔軟性がもたらされる。

【 0 2 1 8 】

[00263] たとえば、ある累算器モード構成では、2 つの累算器ブロック 4 7 2 のプログラム可能な入力データパス 4 7 8 および / またはプログラム可能な内部データパス 4 8 0 は、非限定的な例として、単一の 4 0 ビット累算器を提供するようにプログラムされ得る。別の累算器モード構成では、2 つの累算器ブロック 4 7 2 のプログラム可能な入力データパス 4 7 8 および / またはプログラム可能な内部データパス 4 8 0 は、非限定的な例として、二重 2 4 ビット累算器を提供するようにプログラムされ得る。別の累算器モード構成では、2 つの累算器ブロック 4 7 2 のプログラム可能な入力データパス 4 7 8 および / またはプログラム可能な内部データパス 4 8 0 は、1 6 ビット桁上げ保存加算器、それに続く単一の 2 4 ビット累算器を提供するようにプログラムされ得る。乗算演算と累算演算の特定の様々な組合せも、乗算器ブロック 4 6 2 (3) ~ 4 6 2 (0) および累算器ブロック 4 7 2 (3) ~ 4 7 2 (0) (たとえば、1 6 ビット累算を用いる 1 6 ビット虚数乗算、および 1 6 ビット累算を用いる 3 2 ビット虚数乗算) のプログラミングに従って、実行ユニット 8 4 によってサポートされ得る。

【 0 2 1 9 】

[00264] 累算器ブロック 4 7 2 (3) ~ 4 7 2 (0) のプログラム可能な入力データパス 4 7 8 (3) ~ 4 7 8 (0) および / またはプログラム可能な内部データパス 4 8 0 (3) ~ 4 8 0 (0) は、図 2 のベースバンドプロセッサ 2 0 の命令ディスパッチ回路 4 8 内のベクトル命令デコードから供給される設定値に従ってプログラムされ得る。ベクトル命令は、実行ユニット 8 4 によって実行されるべき特定のタイプの演算を指定する。したがって、実行ユニット 8 4 は、累算器ブロック 4 7 2 (3) ~ 4 7 2 (0) のプログラム可能な入力データパス 4 7 8 (3) ~ 4 7 8 (0) および / またはプログラム可能な内部

データバス 480(3) ~ 480(0) を、命令ディスパッチ回路 48 内の命令パイプラインにおけるベクトル命令の復号に従って実行されるベクトル命令ごとに、再プログラムするように構成され得る。ベクトル命令は、実行ユニット 84 の 1 つまたは複数のクロックサイクルにわたって実行することができる。また、この例では、実行ユニット 84 は、累算器ブロック 472(3) ~ 472(0) のプログラム可能な入力データバス 478(3) ~ 478(0) および / またはプログラム可能な内部データバス 480(3) ~ 480(0) を、クロックサイクルごとにベクトル命令のクロックサイクルごとに、再プログラムするように構成され得る。したがって、たとえば、実行ユニット 84 内の M1 累算ベクトル処理ステージ 460(2) によって実行されるベクトル命令が、クロックサイクルごとにベクトル乗算出力サンプルセット 468(3) ~ 468(0) を処理する場合、結果として、累算器ブロック 472(3) ~ 472(0) のプログラム可能な入力データバス 478(3) ~ 478(0) および / またはプログラム可能な内部データバス 480(3) ~ 480(0) は、ベクトル命令の実行中、クロックサイクルごとに再構成され得る。

10

【0220】

[00265] 図 36 は、例示的なベクトル処理のさらなる説明を提供するために、図 2 および図 35 の実行ユニット 84 内の乗算器ブロック 462(A) ~ 462(0) および累算器ブロック 472(A)(1) ~ 472(0)(0) の例示的なベクトル処理を示すフローチャートである。乗算器ブロック 462(A) ~ 462(0) および累算器ブロック 472(A)(1) ~ 472(0)(0) は、各々プログラム可能なデータバス構成を有し、図 2 および図 35 の例示的な実行ユニット 84 内の様々なベクトル処理ステージ内に設けられる。たとえば、FFT ベクトル演算は、乗算演算と、それに続く累算演算とを伴う。

20

【0221】

[00266] この関連で、図 36 に関して、ベクトル処理は、入力処理ステージ 460(0) 内の複数の入力データバス A3 ~ C0 の中の入力データバス内で、ベクトルアレイの幅の複数の乗算ベクトルデータサンプルセット 34(Y) ~ 34(0) を受信することを伴う (ブロック 501)。ベクトル処理は、次いで、複数の乗算器ブロック 462(A) ~ 462(0) 内の複数の入力データバス A3 ~ C0 から乗算ベクトルデータサンプルセット 34(Y) ~ 34(0) を受信することを含む (ブロック 503)。ベクトル処理は、次いで、ベクトル処理ステージ 460(1) によって実行されるベクトル命令に従って、乗算器ブロック 462(A) ~ 462(0) のためのプログラム可能なデータバス構成に基づいて、複数の乗算出力データバス 470(A) ~ 470(0) の中の乗算出力データバス 470(A) ~ 470(0) 内に乗算ベクトル結果出力サンプルセット 468(A) ~ 468(0) を供給するために、乗算ベクトルデータサンプルセット 34(Y) ~ 34(0) を乗算することを含む (ブロック 505)。ベクトル処理は、次に、複数の累算器ブロック 472(A)(1) ~ 472(0)(0) 内の複数の乗算出力データバス 470(A) ~ 470(0) から乗算ベクトル結果出力サンプルセット 468(A) ~ 468(0) を受信することを含む (ブロック 507)。ベクトル処理は、次に、第 2 のベクトル処理ステージ 460(2) によって実行されるベクトル命令に従って、累算器ブロック 472(A)(1) ~ 472(0)(0) のためのプログラム可能な入力データバス 478(A)(1) ~ 478(0)(0)、プログラム可能な内部データバス 480(A)(1) ~ 480(0)(0)、およびプログラム可能な出力データバス 474(A)(1) ~ 474(0)(0) の構成に基づいて、累算器出力サンプルセット 476(A)(1) ~ 476(0)(0) を供給するために、乗算ベクトル結果出力サンプルセット 468(A) ~ 468(0) を一緒に累算することを含む (ブロック 509)。ベクトル処理は、次いで、プログラム可能な出力データバス 474(A)(1) ~ 474(0)(0) 内に累算器出力サンプルセット 476(A)(1) ~ 476(0)(0) を供給することを含む (ブロック 511)。ベクトル処理は、次いで、出力ベクトル処理ステージ 460(3) 内の累算器ブロック 472(A)(1) ~ 472(0)(0) から累算器出力サンプルセ

30

40

50

ット476(A)(1)~476(0)(0)を受信することを含む(ブロック513)。

【0222】

[00267]プログラム可能なデータパス構成を有するベクトル処理ブロックを利用する、図35の例示的な実行ユニット84および図36のベクトル処理の概要が記載されたので、説明の残りは、図37~図40におけるこれらのベクトル処理ブロックのより例示的な、非限定的な詳細を記載する。

【0223】

[00268]この関連で、図37は、図35の実行ユニット84のM0乗算ベクトル処理ステージ460(1)内の複数の乗算器ブロック462(3)~462(0)のより詳細な概略図である。図38は、図37の乗算器ブロック462の内部構成要素の概略図である。図37に示されたように、特定の入力データパスA3~A0、B3~B0、C3~C0に従って、乗算器ブロック462(3)~462(0)によって受信されるベクトルデータ入力サンプルセット466(11)~466(0)が示されている。図38に関して下記でより詳細に説明されるように、この例における乗算器ブロック462(3)~462(0)の各々は、4つの8ビット×8ビット乗算器を含む。図37に戻って参照すると、この例における乗算器ブロック462(3)~462(0)の各々は、被乗数入力「A」を被乗数入力「B」または被乗数入力「C」のいずれかと乗算するように構成される。乗算器ブロック462において一緒に乗算され得る被乗数入力「A」および「B」または「C」は、図37に示されたように、どの入力データパスA3~A0、B3~B0、C3~C0が乗算器ブロック462(3)~462(0)に接続されるかによって制御される。被乗数選択器入力482(3)~482(0)は、被乗数入力「B」または被乗数入力「C」のどちらが被乗数入力「A」と乗算されるために選択されるかを選択するように、各乗算器ブロック462(3)~462(0)内のプログラム可能な内部データパス467(3)~467(0)を制御するために、各乗算器ブロック462(3)~462(0)に入力として供給される。このようにして、乗算器ブロック462(3)~462(0)は、必要に応じて、それらのプログラム可能な内部データパス467(3)~467(0)が様々な乗算演算を提供するように再プログラムされるための能力を提供される。

【0224】

[00269]引き続き図37を参照すると、一例として乗算器ブロック462(3)を使用すると、入力データパスA3およびA2は、それぞれ入力AHおよびALに接続される。入力AHは被乗数入力Aの上位ビットを表し、ALは入力被乗数入力「A」の下位ビットを意味する。入力データパスB3およびB2は、それぞれ入力BHおよびBLに接続される。入力BHは被乗数入力「B」の上位ビットを表し、ALは入力被乗数入力「B」の下位ビットを表す。入力データパスC3およびC2は、それぞれ入力CIおよびCQに接続される。入力CIは、この例では入力被乗数入力「C」の実数ビット部分を表す。CQは、この例では入力被乗数入力「C」の虚数ビット部分を表す。図38に関して下記でより詳細に説明されるように、被乗数選択器入力482(3)はまた、この例では、乗算器ブロック462(3)のプログラム可能な内部データパス467(3)が、被乗数入力「A」に対する8ビット乗算を被乗数入力「B」または被乗数入力「C」のいずれと実行するように構成されるか、または乗算器ブロック462(3)が、被乗数入力「A」に対する16ビット乗算を被乗数入力「B」または被乗数入力「C」のいずれと実行するように構成されるかを制御する。

【0225】

[00270]引き続き図37を参照すると、乗算器ブロック462(3)~462(0)は、各々、それらのプログラム可能な内部データパス467(3)~467(0)の構成に基づいて、乗算演算の桁上げ「C」および和「S」のベクトル出力サンプルセットとして、ベクトル乗算出力サンプルセット468(3)~468(0)を生成するように構成される。図39および図40に関して下記でより詳細に説明されるように、ベクトル乗算出力サンプルセット468(3)~468(0)の桁上げ「C」および和「S」は融合され

10

20

30

40

50

、桁上げ「C」および和「S」が、複数の累算器ブロック472(3)~472(0)において冗長桁上げ保存フォーマットを提供するために、複数の累算器ブロック472(3)~472(0)に冗長桁上げ保存フォーマットで供給されることを意味する。下記でより詳細に説明されるように、複数の累算器ブロック472(3)~472(0)において冗長桁上げ保存フォーマットを提供することにより、複数の累算器ブロック472(3)~472(0)によって実行される累算演算中に、桁上げ伝搬パスと桁上げ伝搬加算演算とを提供する必要をなくすることができる。

【0226】

[00271]それらのプログラム可能な内部データパス467(3)~467(0)の構成に基づいて、乗算演算の桁上げ「C」および和「S」のベクトル出力サンプルセットとして、ベクトル乗算出力サンプルセット468(3)~468(0)を生成する乗算器ブロック462(3)~462(0)の例が図37に示される。たとえば、乗算器ブロック462(3)は、8ビット乗算のための32ビット値として桁上げC00と和S00とを生成し、16ビット乗算のための64ビット値として桁上げC01と和S01とを生成するように構成される。他の乗算器ブロック462(2)~462(0)は、この例では同じ能力を有する。この関連で、乗算器ブロック462(2)は、8ビット乗算のための32ビット値として桁上げC10と和S10とを生成し、16ビット乗算のための64ビット値として桁上げC11と和S11とを生成するように構成される。乗算器ブロック462(1)は、8ビット乗算のための32ビット値として桁上げC20と和S20とを生成し、16ビット乗算のための64ビット値として桁上げC21と和S21とを生成するように構成される。乗算器ブロック462(0)は、8ビット乗算のための32ビット値として桁上げC30と和S30とを生成し、16ビット乗算のための64ビット値として桁上げC31と和S31とを生成するように構成される。

【0227】

[00272]図37の乗算器ブロック462内に設けられるプログラム可能なデータパス構成のより例示的な詳細を説明するために、図38が提供される。図38は、8ビット×8ビットベクトルデータ入力サンプルセット466と、16ビット×16ビットベクトルデータ入力サンプルセット466とを乗算することが可能な、プログラム可能なデータパス構成を有する、図37の乗算器ブロック462の内部構成要素の概略図である。この関連で、乗算器ブロック462は、この例では4つの8×8ビット乗算器484(3)~484(0)を含む。任意の所望の数の乗算器484が設けられる可能性がある。第1の乗算器484(3)は、(入力被乗数入力「A」の上位ビットである)8ビットベクトルデータ入力サンプルセット466A[H]を受信し、ベクトルデータ入力サンプルセット466A[H]を、(入力被乗数入力「B」の上位ビットである)8ビットベクトルデータ入力サンプルセット466B[H]または(入力被乗数入力「C」の上位ビットである)8ビットベクトルデータ入力サンプルセット466C[I]のいずれかと乗算するように構成される。乗算器484(3)に被乗数として供給している8ビットベクトルデータ入力サンプルセット466B[H]または8ビットベクトルデータ入力サンプルセット466C[I]のいずれかを選択するように構成された、マルチプレクサ486(3)が設けられる。マルチプレクサ486(3)は、この実施形態では、被乗数選択器入力482内の上位ビットである被乗数選択器入力482[3]によって制御される。このようにして、マルチプレクサ486(3)および被乗数選択器入力482[3]は、8ビットベクトルデータ入力サンプルセット466B[H]または8ビットベクトルデータ入力サンプルセット466C[I]のどちらが、受信されたベクトルデータ入力サンプルセット466A[H]と乗算されるかを、乗算器484(3)が制御するためのプログラム可能な内部データパス467[0]構成を提供する。

【0228】

[00273]引き続き図38を参照すると、他の乗算器484(2)~484(0)も、第1の乗算器484(3)用に設けられたものと同様のプログラム可能な内部データパス467[2]~467[0]を含む。乗算器484(2)は、被乗数入力「A」の下位ビッ

10

20

30

40

50

トである 8 ビットベクトルデータ入力サンプルセット 4 6 6 A [L] と乗算されるべき、8 ビットベクトルデータ入力サンプルセット 4 6 6 B [H] または 8 ビットベクトルデータ入力サンプルセット 4 6 6 C [I] のいずれかを、プログラム可能な内部データパス 4 6 7 [1] 内に供給するためのプログラム可能な構成を有する、プログラム可能な内部データパス 4 6 7 [2] を含む。選択は、この実施形態では、被乗数選択器入力 4 8 2 内の被乗数選択器入力 4 8 2 [2] に従って、マルチプレクサ 4 8 6 (2) によって制御される。乗算器 4 8 4 (1) は、8 ビットベクトルデータ入力サンプルセット 4 6 6 A [H] と乗算されるべき、被乗数入力「B」の下位ビットである 8 ビットベクトルデータ入力サンプルセット 4 6 6 B [L]、または被乗数入力「C」の下位ビットである 8 ビットベクトルデータ入力サンプルセット 4 6 6 C [Q] のいずれかを、プログラム可能な内部データパス 4 6 7 [1] 内に供給するようにプログラム可能な、プログラム可能な内部データパス 4 6 7 [1] を含む。選択は、この実施形態では、被乗数選択器入力 4 8 2 内の被乗数選択器入力 4 8 2 [1] に従って、マルチプレクサ 4 8 6 (1) によって制御される。さらに、乗算器 4 8 4 (0) は、8 ビットベクトルデータ入力サンプルセット 4 6 6 A [L] と乗算されるべき、8 ビットベクトルデータ入力サンプルセット 4 6 6 B [L] または 8 ビットベクトルデータ入力サンプルセット 4 6 6 C [Q] のいずれかを、プログラム可能な内部データパス 4 6 7 [0] 内に供給するようにプログラム可能な、プログラム可能な内部データパス 4 6 7 [0] を含む。選択は、この実施形態では、乗数選択器入力 4 8 2 内の被乗数選択器ビット入力 4 8 2 [0] に従って、マルチプレクサ 4 8 6 (0) によって制御される。

【 0 2 2 9 】

[00274]引き続き図 3 8 を参照すると、上記で説明されたように、乗算器 4 8 4 (3) ~ 4 8 4 (0) は、様々なビット長乗算演算を実行するように構成され得る。この関連で、各乗算器 4 8 4 (3) ~ 4 8 4 (0) は、それぞれ、ビット長乗算モード入力 4 8 8 (3) ~ 4 8 8 (0) を含む。この例では、各乗算器 4 8 4 (3) ~ 4 8 4 (0) は、それぞれ、プログラム可能なデータパス 4 9 0 (3) ~ 4 9 0 (0)、4 9 1、および 4 9 2 (3) ~ 4 9 2 (0) の構成を制御する入力に従って、8 ビット×8 ビットモードでプログラムされ得る。各乗算器 4 8 4 (3) ~ 4 8 4 (0) はまた、それぞれ、プログラム可能なデータパス 4 9 0 (3) ~ 4 9 0 (0)、4 9 1、および 4 9 2 (3) ~ 4 9 2 (0) の構成を制御する入力に従って、16 ビット×16 ビットモードと 24 ビット×8 ビットモードとを含む、より大きいビット乗算演算の一部を提供するようにプログラムされ得る。たとえば、各乗算器 4 8 4 (3) ~ 4 8 4 (0) が、プログラム可能なデータパス 4 9 0 (3) ~ 4 9 0 (0) の構成に従って 8 ビット×8 ビット乗算モードで構成される場合、ユニットとしての複数の乗算器 4 8 4 (3) ~ 4 8 4 (0) は、乗算器ブロック 4 6 2 の一部として 2 つの個々の 8 ビット×8 ビット乗算器を備えるように構成され得る。各乗算器 4 8 4 (3) ~ 4 8 4 (0) が、プログラム可能なデータパス 4 9 1 の構成に従って 16 ビット×16 ビット乗算モードで構成される場合、ユニットとしての複数の乗算器 4 8 4 (3) ~ 4 8 4 (0) は、乗算器ブロック 4 6 2 の一部として単一の 16 ビット×16 ビット乗算器を備えるように構成され得る。乗算器 4 8 4 (3) ~ 4 8 4 (0) が、プログラム可能なデータパス 4 9 2 (3) ~ 4 9 2 (0) の構成に従って 24 ビット×8 ビット乗算モードで構成される場合、ユニットとしての複数の乗算器 4 8 4 (3) ~ 4 8 4 (0) は、乗算器ブロック 4 6 2 の一部として 1 つの 16 ビット×24 ビット×8 ビット乗算器を備えるように構成され得る。

【 0 2 3 0 】

[00275]引き続き図 3 8 を参照すると、この例における乗算器 4 8 4 (3) ~ 4 8 4 (0) は、16 ビット×16 ビット乗算モードで構成されるものとして示されている。16 ビットの入力 4 9 4 (3)、4 9 4 (2) および入力桁上げ 4 9 6 (3)、4 9 6 (2) は、それぞれ、各乗算器 4 8 4 (3)、4 8 4 (2) によって生成される。16 ビットの入力 4 9 4 (1)、4 9 4 (0) および入力桁上げ 4 9 6 (1)、4 9 6 (0) は、それぞれ、各乗算器 4 8 4 (1)、4 8 4 (0) によって生成される。16 ビットの入力

和 4 9 4 (3)、4 9 4 (2) および入力桁上げ 4 9 6 (3)、4 9 6 (2) はまた、一緒に入力 和 4 9 4 (3) ~ 4 9 4 (0) と入力桁上げ 4 9 6 (3) ~ 4 9 6 (0) とを加算するために、1 6 ビットの和入力 4 9 4 (1)、4 9 4 (0) および入力桁上げ 4 9 6 (1)、4 9 6 (0) とともに 2 4 ビット 4 : 2 圧縮器 5 1 5 に供給される。加算された入力 和 4 9 4 (3) ~ 4 9 4 (0) および入力桁上げ 4 9 6 (3) ~ 4 9 6 (0) は、プログラム可能なデータパス 4 9 1 がアクティブであり、入力 和 4 9 4 (3) ~ 4 9 4 (0) および入力桁上げ 4 9 6 (3) ~ 4 9 6 (0) とゲート制御されるとき、1 6 ビット \times 1 6 ビット乗算モードで単一和 4 9 8 と単一桁上げ 5 0 0 とをもたらす。プログラム可能なデータパス 4 9 1 は、2 4 ビット 4 : 2 圧縮器 5 1 5 に供給されるように、1 6 ビットワードとして、組み合わされた入力 和 4 9 4 (3)、4 9 4 (2) を有する第 1 の AND ベースのゲート 5 0 2 (3) によって、および 1 6 ビットワードとして、組み合わされた入力桁上げ 4 9 6 (3)、4 9 6 (2) を有する第 2 の AND ベースのゲート 5 0 2 (2) によってゲート制御される。プログラム可能なデータパス 4 9 1 はまた、2 4 ビット 4 : 2 圧縮器 5 1 5 に供給されるように、1 6 ビットワードとして、組み合わされた入力 和 4 9 4 (1)、4 9 4 (0) を有する第 3 の AND ベースのゲート 5 0 2 (1) によって、および 1 6 ビットワードとして、組み合わされた入力桁上げ 4 9 6 (1)、4 9 6 (0) を有する第 4 の AND ベースのゲート 5 0 2 (0) によってゲート制御される。乗算器ブロック 4 6 2 が 1 6 ビット \times 1 6 ビット乗算モードまたは 2 4 ビット \times 8 ビット乗算モードで構成される場合、プログラム可能な出力データパス 4 7 0 [0] は、圧縮された 3 2 ビット和 S 0 および 3 2 ビット桁上げ C 0 部分積として、ベクトル乗算出力サンプルセット 4 6 8 [0] を供給される。

【 0 2 3 1 】

[00276] 乗算器ブロック 4 6 2 内の乗算器 4 8 4 (3) ~ 4 8 4 (0) が 8 ビット \times 8 ビット乗算モードで構成される場合、プログラム可能な出力データパス 4 7 0 [1] 構成は、圧縮なしで、1 6 ビット入力 和 4 9 4 (3) ~ 4 9 4 (0) および部分積としての対応する 1 6 ビット入力桁上げ 4 9 6 (3) ~ 4 9 6 (0) として提供される。乗算器ブロック 4 6 2 内の乗算器 4 8 4 (3) ~ 4 8 4 (0) が 8 ビット \times 8 ビット乗算モードで構成される場合、プログラム可能な出力データパス 4 7 0 [1] は、圧縮なしで、1 6 ビット入力 和 4 9 4 (3) ~ 4 9 4 (0) およびベクトル乗算出力サンプルセット 4 6 8 [1] としての対応する 1 6 ビット入力桁上げ 4 9 6 (3) ~ 4 9 6 (0) として提供される。乗算器ブロック 4 6 2 の乗算モードに依存するベクトル乗算出力サンプルセット 4 6 8 [0]、4 6 8 [1] は、実行されているベクトル命令に従って、和および桁上げ積の累算のために、累算器ブロック 4 7 2 (3) ~ 4 7 2 (0) に供給される。

【 0 2 3 2 】

[00277] プログラム可能なデータパス構成を有する、図 3 7 および図 3 8 の乗算器ブロック 4 6 2 (3) ~ 4 6 2 (0) が記載されたので、冗長桁上げ保存フォーマットで構成された累算器ブロック 4 7 2 (3) ~ 4 7 2 (0) と融合されるべき、実行ユニット 8 4 内の乗算器ブロック 4 6 2 (3) ~ 4 6 2 (0) の特徴が、図 3 9 に関して次に概説される。

【 0 2 3 3 】

[00278] この関連で、図 3 9 は、上述された実行ユニット 8 4 (0) ~ 8 4 (X) 内の乗算器ブロックおよび累算器ブロックの一般化された概略図であり、ここで、累算器ブロックは、桁上げ伝搬を低減するために冗長桁上げ保存フォーマットを利用する桁上げ保存累算器構造を利用する。前に説明され、図 3 8 に示されたように、乗算器ブロック 4 6 2 は、被乗数入力 4 6 6 [H] と 4 6 6 [L] とを乗算し、少なくとも 1 つの入力 和 4 9 4 と少なくとも 1 つの入力桁上げ 4 9 6 とを、ベクトル乗算出力サンプルセット 4 6 8 として、プログラム可能な出力データパス 4 7 0 内に供給するように構成される。累算ステップごとに累算器ブロック 4 7 2 内に桁上げ伝搬バスと桁上げ伝搬加算器とを設ける必要をなくするために、プログラム可能な出力データパス 4 7 0 内のベクトル乗算出力サンプルセット 4 6 8 内の少なくとも 1 つの入力 和 4 9 4 および少なくとも 1 つの入力桁上げ 4 9 6

が、少なくとも1つの累算器ブロック472に冗長桁上げ保存フォーマットで融合される。言い換えれば、ベクトル乗算出力サンプルセット468内の桁上げ496は、累算器ブロック472に桁上げ保存フォーマットでベクトル入力桁上げ496として供給される。このようにして、ベクトル乗算出力サンプルセット468内の入力494および入力桁上げ496は、この実施形態では複合ゲート4:2圧縮器である累算器ブロック472の圧縮器508に供給され得る。圧縮器508は、入力494および入力桁上げ496を、それぞれ、前の累算ベクトル出力和512および前のシフトされた累算ベクトル出力桁上げ517と一緒に累算するように構成される。前のシフトされた累算ベクトル出力桁上げ517は、本質的に、累算演算中の保存された桁上げ累算である。

【0234】

10

[00279]このようにして、受信された入力桁上げ496を累算器ブロック472によって生成された累算の一部として入力494に伝搬するために、単一の最終桁上げ伝搬加算器のみが累算器ブロック472内に設けられることが必要である。累算器ブロック472内の累算の各ステップ中に、桁上げ伝搬加算演算を実行することに関連する電力消費が、この実施形態では低減される。また、累算器ブロック472内の累算の各ステップ中に、桁上げ伝搬加算演算を実行することに関連するゲート遅延も、この実施形態ではなくなる。

【0235】

[00280]引き続き図39を参照すると、圧縮器508は、冗長な形式での入力494および入力桁上げ496を、それぞれ、前の累算ベクトル出力和512および前のシフトされた累算ベクトル出力桁上げ517と累算するように構成される。シフトされた累算ベクトル出力桁上げ517は、次の受信された入力494および入力桁上げ496の次の累算が圧縮器508によって実行される前に、累算ベクトル出力桁上げ514をシフトすることにより、圧縮器508によって生成された累算ベクトル出力桁上げ514によって生成される。最終的なシフトされた累算ベクトル出力桁上げ517は、最終的なシフトされた累算ベクトル出力桁上げ517内で桁上げ累算を伝搬して、最終累算ベクトル出力和512を最終累算器出力サンプルセット476の2の補数表現に変換するために、累算器ブロック472内に設けられた単一の最終桁上げ伝搬加算器519によって最終累算ベクトル出力和512に加算される。最終累算ベクトル出力和512は、プログラム可能な出力データパス474内で累算器出力サンプルセット476として供給される(図35参照)。

20

30

【0236】

[00281]冗長桁上げ保存フォーマットで構成された累算器ブロック472との乗算器ブロック462の融合を示す図39が記載されたので、累算器ブロック472(3)~472(0)に関するより例示的な詳細が、図40に関してここで概説される。図40は、図35の実行ユニット84内に設けられた累算器ブロック472の例示的な内部構成要素の詳細な概略図である。前に説明され、下記でより詳細に説明されるように、累算器ブロック472は、プログラム可能な入力データパス478(3)~478(0)および/またはプログラム可能な内部データパス480(3)~480(0)を用いて構成され、その結果、累算器ブロック472は、特定の異なるタイプのベクトル累算演算を実行するように設計された専用回路として働くようにプログラムされ得る。たとえば、累算器ブロック472は、符号付きおよび符号なしの累算演算を含む、いくつかの様々な累算と加算とを提供するようにプログラムされ得る。様々なタイプの累算演算を提供するように構成されている累算器ブロック472内のプログラム可能な入力データパス478(3)~478(0)および/またはプログラム可能な内部データパス480(3)~480(0)の具体例が開示される。また、累算器ブロック472は、低減された組合せ論理を用いて高速累算演算を提供するために、桁上げ伝搬を回避または低減するように冗長桁上げ算術を提供するために、桁上げ保存累算器472[0]、472[1]を含むように構成される。

40

【0237】

[00282]累算器ブロック472の例示的な内部構成要素が図40に示される。そこに示

50

されているように、この実施形態における累算器ブロック472は、一緒に累算されるために、第1の入力494[0]および第1の入力桁上げ496[0]と、第2の入力494[1]および第2の入力桁上げ496[1]とを乗算器ブロック462から受信するように構成される。図40に関して、入力494[0]、494[1]および入力桁上げ496[0]、496[1]は、ベクトル入力494[0]、494[1]およびベクトル入力桁上げ496[0]、496[1]と呼ばれる。前述され、図39に示されたように、この実施形態におけるベクトル入力494[0]、494[1]およびベクトル入力桁上げ496[0]、496[1]は、各々長さが16ビットである。この例における累算器ブロック472は、2つの24ビット桁上げ保存累算器ブロック472[0]、472[1]として設けられ、「[0]」が桁上げ保存累算器472[0]用に指定され、「[1]」が桁上げ保存累算器472[1]用に指定される、共通要素番号を有する同様の構成要素を各々が含んでいる。桁上げ保存累算器472[0]、472[1]は、同時にベクトル累算演算を実行するように構成され得る。

【0238】

[00283]図40の桁上げ保存累算器472[0]を参照すると、ベクトル入力494[0]およびベクトル入力桁上げ496[0]は、プログラム可能な内部データパス480[0]の一部として設けられたマルチプレクサ504(0)内の入力である。負のベクトル入力494[0]'と負のベクトル入力桁上げ496[0]'とを必要とする累算演算のための、マルチプレクサ504(0)への入力として、入力521(0)に従って負のベクトル入力494[0]'と負のベクトル入力桁上げ496[0]'とを生成する、排他的ORベースのゲートから構成され得る否定回路506(0)も設けられる。マルチプレクサ504(0)は、ベクトル命令復号の結果として生成された、選択器入力510(0)に従って圧縮器508(0)に供給されるべき、ベクトル入力494[0]およびベクトル入力桁上げ496[0]、または負のベクトル入力494[0]'および負のベクトル入力桁上げ496[0]'のいずれかを選択するように構成される。この関連で、選択器入力510(0)により、累算器ブロック472によって実行されるように構成された累算演算に従って、桁上げ保存累算器472[0]のプログラム可能な入力データパス478[0]が、ベクトル入力494[0]およびベクトル入力桁上げ496[0]、または負のベクトル入力494[0]'および負のベクトル入力桁上げ496[0]'のいずれかを圧縮器508(0)に供給するようにプログラム可能になる。

【0239】

[00284]引き続き図40を参照すると、この実施形態における桁上げ保存累算器ブロック472[0]の圧縮器508(0)は、複合ゲート4:2圧縮器である。この関連で、圧縮器508(0)は、冗長桁上げ保存演算において和と桁上げとを累算するように構成される。圧縮器508(0)は、圧縮器508(0)への4つの入力として、現在のベクトル入力494[0]およびベクトル入力桁上げ496[0]、または現在の負のベクトル入力494[0]'および負のベクトル入力桁上げ496[0]'を、前の累算されたベクトル入力494[0]およびベクトル入力桁上げ496[0]、または累算された負のベクトル入力494[0]'および負のベクトル入力桁上げ496[0]'と一緒に累算するように構成される。圧縮器508(0)は、累算器出力サンプルセット476(3)~476(0)を供給するために、プログラム可能な出力データパス474[0](図35参照)内の累算器出力サンプルセット476[0]として、累算ベクトル出力512(0)と累算ベクトル出力桁上げ514(0)とを供給する。累算ベクトル出力桁上げ514(0)は、各累算ステップ中にビット幅成長を制御するために、シフトされた累算ベクトル出力桁上げ517(0)を供給するように、累算演算中にビットシフト516(0)によってシフトされる。たとえば、この実施形態におけるビットシフト516(0)は、冗長桁上げ保存フォーマットで圧縮器508(0)に融合されるパレルシフトである。このようにして、シフトされた累算ベクトル出力桁上げ517(0)は、本質的に、累算器ブロック472[0]によって実行される累算演算中に、累算ベクトル出力512(0)に伝搬される必要なしに保存される。このようにして、累算器ブロック4

10

20

30

40

50

7 2 [0] 内の累算の各ステップ中に桁上げ伝搬加算演算を実行することに関連する電力消費およびゲート遅延が、この実施形態ではなくなる。

【 0 2 4 0 】

[00285]さらなる後続のベクトル入力 4 9 4 [0] およびベクトル入力桁上げ 4 9 6 [0]、または負のベクトル入力 4 9 4 [0] ' および負のベクトル入力桁上げ 4 9 6 [0] ' は、現在の累算ベクトル出力 5 1 2 (0) および現在の累算ベクトル出力桁上げ 5 1 7 (0) と累算され得る。ベクトル入力 4 9 4 [0] およびベクトル入力桁上げ 4 9 6 [0]、または負のベクトル入力 4 9 4 [0] ' および負のベクトル入力桁上げ 4 9 6 [0] ' は、ベクトル命令復号の結果として生成された、和桁上げ選択器 5 2 0 (0) に従うプログラム可能な内部データパス 4 8 0 [0] の一部として、マルチプレクサ 5 1 8 (0) によって選択される。現在の累算ベクトル出力 5 1 2 (0) および現在のシフトされた累算ベクトル出力桁上げ 5 1 7 (0) は、桁上げ保存累算器ブロック 4 7 2 [0] が、更新された累算ベクトル出力 5 1 2 (0) と累算ベクトル出力桁上げ 5 1 4 (0) とを供給するために、圧縮器 5 0 8 (0) への入力として供給され得る。この関連で、和桁上げ選択器 5 2 0 (0) により、累算器ブロック 4 7 2 [0] のプログラム可能な内部データパス 4 8 0 [0] が、累算器ブロック 4 7 2 によって実行されるように構成された累算演算に従って、圧縮器 5 0 8 (0) にベクトル入力 4 9 4 [0] とベクトル入力桁上げ 4 9 6 [0] とを供給するようにプログラム可能になる。桁上げ保存累算器ブロック 4 7 2 [0] における累算の動作タイミングを制御するために、保持状態入力 5 2 6 (0) に従って累算ベクトル出力 5 1 2 (0) およびシフトされた累算ベクトル出力桁上げ 5 1 7 (0) の現在の状態を、マルチプレクサ 5 1 8 (0) に保持させるために、この実施形態では保持ゲート 5 2 2 (0)、5 2 4 (0) も提供される。

【 0 2 4 1 】

[00286]引き続き図 4 0 を参照すると、桁上げ保存累算器ブロック 4 7 2 [0] の累算ベクトル出力 5 1 2 (0) およびシフトされた累算ベクトル出力桁上げ 5 1 7 (0)、ならびに桁上げ保存累算器ブロック 4 7 2 [1] の累算ベクトル出力 5 1 2 (1) およびシフトされた累算ベクトル出力桁上げ 5 1 7 (1) は、それぞれ、制御ゲート 5 3 4 (0)、5 3 6 (0) および 5 3 4 (1)、5 3 6 (1) によってゲート制御される。制御ゲート 5 3 4 (0)、5 3 6 (0) および 5 3 4 (1)、5 3 6 (1) は、それぞれ、圧縮器 5 0 8 (0)、5 0 8 (1) に戻される、累算ベクトル出力 5 1 2 (0) およびシフトされた累算ベクトル出力桁上げ 5 1 7 (0) と、累算ベクトル出力 5 1 2 (1) およびシフトされた累算ベクトル出力桁上げ 5 1 7 (1) とを制御する。

【 0 2 4 2 】

[00287]要約すると、図 4 0 の累算器ブロック 4 7 2 の累算器ブロック 4 7 2 [0]、4 7 2 [1] のプログラム可能な入力データパス 4 7 8 [0]、4 7 8 [1] およびプログラム可能な内部データパス 4 8 0 [0]、4 8 0 [1] により、累算器ブロック 4 7 2 は様々なモードで構成され得る。累算器ブロック 4 7 2 は、図 4 0 に示された共通累算器回路による特定のベクトル処理命令に従って、様々な累算演算を提供するように構成され得る。

【 0 2 4 3 】

[00288]本明細書において説明された概念および実施形態による V P E は、任意のプロセッサベースのデバイス内に設けられるか、または任意のプロセッサベースのデバイスの中に統合される場合がある。限定はしないが、例には、セットトップボックス、エンターテインメントユニット、ナビゲーションデバイス、通信デバイス、固定ロケーションデータユニット、モバイルロケーションデータユニット、モバイルフォン、携帯電話、コンピュータ、ポータブルコンピュータ、デスクトップコンピュータ、携帯情報端末 (P D A)、モニタ、コンピュータモニタ、テレビジョン、チューナ、ラジオ、衛星ラジオ、音楽プレーヤ、デジタル音楽プレーヤ、ポータブル音楽プレーヤ、デジタルビデオプレーヤ、ビデオプレーヤ、デジタルビデオディスク (D V D) プレーヤ、およびポータブルデジタルビデオプレーヤが含まれる。

【 0 2 4 4 】

[00289]この関連で、図 4 1 は、プロセッサベースのシステム 5 5 0 の例を示す。この例では、プロセッサベースのシステム 5 5 0 は、各々が 1 つまたは複数のプロセッサまたはコア 5 5 4 を含む、1 つまたは複数の処理ユニット (P U) 5 5 2 を含む。 P U 5 5 2 は、非限定的な例として、図 2 のベースバンドプロセッサ 2 0 であり得る。プロセッサ 5 5 4 は、非限定的な例として、図 2 に提供されたベースバンドプロセッサ 2 0 のようなベクトルプロセッサであり得る。この関連で、プロセッサ 5 5 4 は、図 2 の実行ユニット 8 4 を含むが、それに限定されない V P E 5 5 6 を含む場合もある。 P U 5 5 2 は、一時的に記憶されたデータへの高速アクセスのための、プロセッサ 5 5 4 に結合されたキャッシュメモリ 5 5 8 を有する場合がある。 P U 5 5 2 は、システムバス 5 6 0 に結合され、プロセッサベースのシステム 5 5 0 に含まれるマスタデバイスとスレーブデバイスとを相互結合することができる。よく知られているように、 P U 5 5 2 は、システムバス 5 6 0 を介してアドレスと、制御と、データ情報とを交換することによって、これらの他のデバイスと通信する。たとえば、 P U 5 5 2 は、スレーブデバイスの例として、メモリコントローラ 5 6 2 にバストランザクション要求を通信することができる。図 4 1 には示されていないが、複数のシステムバス 5 6 0 が提供される可能性があり、ここで、各システムバス 5 6 0 は様々なファブリックを構成する。

10

【 0 2 4 5 】

[00290]他のマスタデバイスおよびスレーブデバイスが、システムバス 5 6 0 に接続され得る。図 4 1 に示されたように、これらのデバイスには、例として、メモリシステム 5 6 4、1 つまたは複数の入力デバイス 5 6 6、1 つまたは複数の出力デバイス 5 6 8、1 つまたは複数のネットワークインターフェースデバイス 5 7 0、および 1 つまたは複数のディスプレイコントローラ 5 7 2 が含まれ得る。メモリシステム 5 6 4 は、メモリコントローラ 5 6 2 によってアクセス可能なメモリ 5 6 5 を含むことができる。入力デバイス 5 6 6 は、限定はしないが、入力キー、スイッチ、音声プロセッサなどを含む、任意のタイプの入力デバイスを含むことができる。出力デバイス 5 6 8 は、限定はしないが、オーディオ、ビデオ、他の視覚的インジケータなどを含む、任意のタイプの出力デバイスを含むことができる。ネットワークインターフェースデバイス 5 7 0 は、ネットワーク 5 7 4 との間のデータ交換を可能にするように構成された任意のデバイスであり得る。ネットワーク 5 7 4 は、限定はしないが、有線またはワイヤレスのネットワーク、専用または公共のネットワーク、ローカルエリアネットワーク (L A N)、ワイドローカルエリアネットワーク (W L A N)、およびインターネットが含まれる、任意のタイプのネットワークであり得る。ネットワークインターフェースデバイス 5 7 0 は、所望の任意のタイプの通信プロトコルをサポートするように構成され得る。

20

30

【 0 2 4 6 】

[00291] P U 5 5 2 はまた、1 つまたは複数のディスプレイ 5 7 8 に送られる情報を制御するために、システムバス 5 6 0 を介してディスプレイコントローラ 5 7 2 にアクセスするように構成される場合がある。ディスプレイコントローラ 5 7 2 は、1 つまたは複数のビデオプロセッサ 5 8 0 を介して表示されるべき情報をディスプレイ 5 7 8 に送り、ビデオプロセッサ 5 8 0 は、表示されるべき情報をディスプレイ 5 7 8 に適したフォーマットに処理する。ディスプレイ 5 7 8 は、限定はしないが、陰極線管 (C R T)、液晶ディスプレイ (L C D)、プラズマディスプレイなどを含む、任意のタイプのディスプレイを含むことができる。

40

【 0 2 4 7 】

[00292]本明細書で開示された二重電圧ドメインメモリバッファの実施形態とともに記載された様々な例示的な論理ブロック、モジュール、回路、およびアルゴリズムは、電子ハードウェアとして、メモリもしくは別のコンピュータ可読媒体に記憶され、プロセッサもしくは他の処理デバイスによって実行される命令として、または両方の組合せとして実装され得ることが、当業者ならさらに諒解されよう。本明細書に記載されたアービタ、マスタデバイス、およびスレーブデバイスは、例として、任意の回路、ハードウェア構成要

50

素、集積回路（ＩＣ）、またはＩＣチップ内で利用される場合がある。本明細書で開示されたメモリは、任意のタイプおよびサイズのメモリであり得るし、所望の任意のタイプの情報を記憶するように構成される場合がある。この互換性を明確に示すために、様々な例示的な構成要素、ブロック、モジュール、回路、およびステップが、概してそれらの機能に関して上述された。そのような機能性がどのように実装されるかは、特定の用途、設計選択、および／または全体的なシステムに課された設計制約に依存する。当業者は、特定の用途ごとに様々な方法で記載された機能を実装することができるが、そのような実装の決定は、本開示の範囲から逸脱する原因になると解釈されるべきではない。

【 0 2 4 8 】

[00293]本明細書で開示された実施形態に関して記載された様々な例示的な論理ブロック、モジュール、および回路は、プロセッサ、DSP、特定用途向け集積回路（ASIC）、FPGAもしくは他のプログラマブル論理デバイス、個別ゲートもしくはトランジスタ論地、個別ハードウェア構成要素、または本明細書に記載された機能を実行するように設計されたそれらの任意の組合せを用いて実装または実行される場合がある。プロセッサはマイクロプロセッサであり得るが、代替として、プロセッサは、任意の従来のプロセッサ、コントローラ、マイクロコントローラ、または状態機械であり得る。プロセッサはまた、コンピューティングデバイスの組合せ、たとえば、DSPとマイクロプロセッサとの組合せ、複数のマイクロプロセッサ、DSPコアと連携する１つもしくは複数のマイクロプロセッサ、または任意の他のそのような構成として実装される場合がある。

【 0 2 4 9 】

[00294]本明細書で開示された実施形態は、ハードウェアにおいて、およびハードウェアに記憶された命令において具現化される場合があり、たとえば、ランダムアクセスメモリ（RAM）、フラッシュメモリ、読取り専用メモリ（ROM）、電気的プログラマブルROM（EPROM）、電気的消去可能プログラマブルROM（EEPROM（登録商標））、レジスタ、ハードディスク、リムーバブルディスク、CD-ROM、または当技術分野で知られている任意の他の形態のコンピュータ可読媒体の中に存在する場合がある。例示的な記憶媒体は、プロセッサが記憶媒体から情報を読み出し、記憶媒体に情報を書き込むことができるように、プロセッサに結合される。代替として、記憶媒体はプロセッサと一体であり得る。プロセッサおよび記憶媒体は、ASICの中に存在する場合がある。ASICはリモート局の中に存在する場合がある。代替として、プロセッサおよび記憶媒体は、個別構成要素としてリモート局、基地局、またはサーバの中に存在する場合がある。

【 0 2 5 0 】

[00295]また、本明細書の例示的な実施形態のいずれかにおいて記載された動作ステップは、例および説明を提供するために記載されたことに留意されたい。記載された動作は、図示されたシーケンス以外の多数の様々なシーケンスにおいて実行される場合がある。さらに、単一の動作ステップにおいて記載された動作は、実際には、いくつかの様々なステップにおいて実行される場合がある。さらに、例示的な実施形態において説明された１つまたは複数の動作ステップは、組み合わされる場合がある。フローチャート図に示された動作ステップは、当業者には容易に明らかになるように、多数の様々な修正を受ける場合があることを理解されたい。情報および信号が様々な異なる技術および技法のいずれかを使用して表され得ることも当業者は理解されよう。たとえば、上記の説明全体を通して参照され得るデータ、命令、コマンド、情報、信号、ビット、シンボル、およびチップは、電圧、電流、電磁波、磁場もしくは磁気粒子、光場もしくは光学粒子、またはそれらの任意の組合せによって表される場合がある。

【 0 2 5 1 】

[00296]本開示の前の説明は、当業者が本開示を製作または使用することを可能にするために提供される。本開示に対する様々な修正は当業者には容易に明らかになり、本明細書で定義された一般原理は、本開示の趣旨または範囲から逸脱することなく、他の変形形態に適用される場合がある。したがって、本開示は、本明細書に記載された例および設計

に限定されるものではなく、本明細書で開示された原理および新規の特徴と一致する最も広い範囲が与えられるべきである。

以下に本願の出願当初の特許請求の範囲に記載された発明を付記する。

[C 1] フィルタベクトル処理動作を供給するように構成されたベクトル処理エンジン (V P E) であって、

フィルタベクトル処理動作のための少なくとも 1 つの入力データフローパス中に入力ベクトルデータサンプルセットを供給することと、

少なくとも 1 つの出力データフローパスから結果として生じるフィルタ化出力ベクトルデータサンプルセットを受信することと、

前記結果として生じるフィルタ化出力ベクトルデータサンプルセットを記憶することと

を行うように構成された少なくとも 1 つのベクトルデータファイルと、

前記少なくとも 1 つの入力データフロー中の少なくとも 1 つの実行ユニットと前記少なくとも 1 つのベクトルデータファイルとの間に設けられた少なくとも 1 つのタップ付き遅延線、前記少なくとも 1 つのタップ付き遅延線は、前記フィルタベクトル処理動作中のフィルタタップの数に等しい複数の処理ステージの中の各処理ステージのための複数のパイプラインレジスタ中のベクトルデータサンプル幅ぶん、前記入力ベクトルデータサンプルセットをシフトすることと、前記複数の処理ステージの中の各処理ステージのためのシフトされた入力ベクトルベクトルデータサンプルを供給することとを行うように構成される、と、

前記少なくとも 1 つの入力データフローパス中に設けられた前記少なくとも 1 つの実行ユニット、前記少なくとも 1 つの実行ユニットは、

前記複数の処理ステージの中の各処理ステージのための前記シフトされた入力ベクトルデータサンプルセット上にフィルタタップ動作を適用することと、前記フィルタベクトル処理動作の各フィルタタップのためのフィルタタップ出力ベクトルデータサンプルセットを生成することとを行うように構成された少なくとも 1 つの乗算器と、

前記複数の処理ステージの中の各処理ステージのための少なくとも 1 つの累算器中で前記フィルタタップ出力ベクトルデータサンプルセットを累算するように構成された前記少なくとも 1 つの累算器と

を備える、と

を備え、

前記少なくとも 1 つの実行ユニットは、前記少なくとも 1 つの出力データフローパス上に前記結果として生じるフィルタ化出力ベクトルデータサンプルを供給するように構成される、

V P E。

[C 2] 前記少なくとも 1 つの累算器は、前記入力ベクトルデータサンプルセットが前記少なくとも 1 つのベクトルデータファイルから再フェッチされることなく、前記複数の処理ステージの中の各処理ステージのための前記フィルタタップ出力ベクトルデータサンプルセットを累算するように構成される、

C 1 に記載の V P E。

[C 3] 前記少なくとも 1 つの累算器は、前記入力ベクトルデータサンプルセット中の入力ベクトルデータサンプルの数が前記フィルタベクトル処理動作中のフィルタタップの数より小さい場合、前記入力ベクトルデータサンプルセットが前記少なくとも 1 つのベクトルデータファイルから再フェッチされることなく、前記複数の処理ステージの中の各処理ステージのための前記フィルタタップ出力ベクトルデータサンプルセットを累算するように構成される、

C 1 に記載の V P E。

[C 4] 前記少なくとも 1 つのタップ付き遅延線は、前記フィルタベクトル処理動作中の前記フィルタタップの数が前記入力ベクトルデータサンプルセット中の入力ベクトルデータサンプルの数より大きい場合、前記少なくとも 1 つのベクトルデータファイルから前記

10

20

30

40

50

少なくとも1つの入力データフローパス上に追加の入力ベクトルデータサンプルセットの別のフェッチを受信するようにさらに構成される、

C 1 に記載の V P E。

[C 5] 前記少なくとも1つのベクトルデータファイルは、

前記フィルタベクトル処理動作のための前記少なくとも1つの入力データフローパス中に前記少なくとも1つのベクトルデータファイルの幅の前記入力ベクトルデータサンプルセットを供給することと、

記憶されるべき前記少なくとも1つの入力データフローパスから、前記少なくとも1つのベクトルデータファイルの前記幅の前記結果として生じるフィルタ化出力ベクトルデータサンプルセットを受信することと

を行うように構成される、C 1 に記載の V P E。

[C 6] 前記少なくとも1つのタップ付き遅延線は、

前記複数のパイプラインレジスタ中に、前記少なくとも1つのベクトルデータファイルからの前記少なくとも1つの入力データフローパス中の前記入力ベクトルデータサンプルセットを受信することと、

前記複数の処理ステージの中の各処理ステージのための前記少なくとも1つの実行ユニットに、前記少なくとも1つの入力データフローパス中の前記シフトされた入力ベクトルデータサンプルセットの各々を供給することと

を行うように構成され、

前記少なくとも1つの実行ユニットは、前記複数の処理ステージの中の各処理ステージのための前記少なくとも1つのタップ付き遅延線から、前記少なくとも1つの入力データフローパス中の前記シフトされた入力ベクトルデータサンプルセットを受信するようにさらに構成される、

C 1 に記載の V P E。

[C 7] 前記少なくとも1つのタップ付き遅延線は、

前記複数の処理ステージの中の第1の処理ステージ中の前記少なくとも1つの実行ユニットに、前記少なくとも1つの入力データフローパス中の第1の入力ベクトルデータサンプルを供給することと、

前記複数の処理ステージの中の後続の処理ステージ中の前記少なくとも1つの実行ユニットに、前記少なくとも1つの入力データフローパス中の前記シフトされた入力ベクトルデータサンプルセットを供給することと

を行うように構成され、

前記少なくとも1つの乗算器は、

前記第1の処理ステージ中に第1のフィルタタップ動作出力ベクトルデータサンプルセットを生成するために、前記第1の入力ベクトルデータサンプルセットと第1のフィルタ係数を乗算することと、

前記後続の処理ステージ中に後続のフィルタ出力ベクトルデータサンプルセットを生成するために、前記シフトされた入力ベクトルデータサンプルセットに後続のフィルタタップ係数を乗算することと

を行うように構成され、

前記少なくとも1つの累算器は、前記結果として生じるフィルタ化出力ベクトルデータサンプルセットを供給するために、前記少なくとも1つの累算器中の前記後続のフィルタタップ出力ベクトルデータサンプルセットと前記第1のフィルタタップ出力ベクトルデータサンプルセットとを累算するように構成される、

C 6 に記載の V P E。

[C 8] 前記少なくとも1つのベクトルデータファイルは、

前記少なくとも1つの入力データフローパス中の前記少なくとも1つのタップ付き遅延線入力上に前記入力ベクトルデータサンプルセットとして前記入力ベクトルデータサンプルセットを供給することと、

前記少なくとも1つの出力データフローパス中の少なくとも1つの実行ユニット出力

10

20

30

40

50

上に前記結果として生じる出力ベクトルデータサンプルセットを受信することと
 を行うようにさらに構成され、
 前記少なくとも1つのタップ付き遅延線は、
 前記少なくとも1つのベクトルデータファイルから前記少なくとも1つの入力データ
 フローパス中の前記少なくとも1つのタップ付き遅延線上に前記入力ベクトルデータサン
 プルセットを受信することと、
 前記複数の処理ステージの中の各処理ステージのための前記少なくとも1つの実行ユ
 ニットに、前記少なくとも1つの入力データフローパス中の少なくとも実行ユニット入力
 上の前記シフトされた入力ベクトルデータサンプルセットの各々を供給することと
 を行うようにさらに構成され、
 前記少なくとも1つの実行ユニットは、
 前記複数の処理ステージの中の各処理ステージのための前記少なくとも1つのタップ
 付き遅延線から前記少なくとも1つの入力データフローパス中の前記少なくとも1つの実
 行ユニット入力上に前記シフトされた入力ベクトルデータサンプルセットを受信すること
 と、
 前記少なくとも1つの実行ユニット出力中の前記少なくとも1つの出力データフロー
 パス上の前記結果として生じるフィルタ化出力ベクトルデータサンプルセットを供給する
 ことと
 を行うようにさらに構成される、C 6 に記載の V P E。

10

[C 9]

20

前記少なくとも1つのタップ付き遅延線は、
 複数のシャドウパイプラインレジスタ中に、前記少なくとも1つの入力データフロー
 パス中の前記少なくとも1つのベクトルデータファイルから次の入力ベクトルデータサン
 プルセットを受信することと、
 前記シフトされた入力ベクトルデータサンプルセットを供給するために、プライマリ
 タップ付き遅延線中に、各処理ステージのための前記複数のシャドウパイプラインレジ
 スタ中の前記ベクトルデータサンプル幅ぶん、前記次の入力ベクトルデータサンプルセッ
 トをシフトすることと
 を行うように構成され、
 前記プライマリタップ付き遅延線は、前記複数の処理ステージ中の各処理ステージのた
 めの前記少なくとも1つの実行ユニットに、前記少なくとも1つの入力データフローパ
 ス中の前記シフトされた入力ベクトルデータサンプルセットを供給するように構成される、
 C 1 に記載の V P E。

30

[C 1 0] 複数の入力ベクトルデータサンプル選択器をさらに備え、前記複数の入力ベク
 トルデータサンプル選択器の各々は、前記少なくとも1つのタップ付き遅延線中の前記複数
 のパイプラインレジスタの中のパイプラインレジスタに割り当てられ、

前記複数の入力ベクトルデータサンプル選択器各々は、前記割り当てられたパイプライン
 レジスタ中のシフトされた入力ベクトルデータサンプルを記憶するために、隣接パイプ
 ラインレジスタ中に記憶された入力ベクトルデータサンプルと前記少なくとも1つのベク
 トルデータファイルからの前記入力ベクトルデータサンプルセットからの入力データサン
 プルとの間で選択するように構成される、

40

C 1 に記載の V P E。

[C 1 1] 前記少なくとも1つのタップ付き遅延線は、前記少なくとも1つの実行ユニッ
 トによって実行されるベクトル命令に従って前記少なくとも1つのタップ付き遅延線のた
 めのプログラマブル入力データパス構成に基づいて、前記少なくとも1つの実行ユニット
 と前記少なくとも1つのベクトルデータファイルとの間の前記少なくとも1つの入力デー
 タフローパス中に選択的に供給されるように構成可能である、

C 1 に記載の V P E。

[C 1 2] 前記少なくとも1つのタップ付き遅延線は、前記少なくとも1つの実行ユニッ
 トによって実行される次のベクトル命令に従って前記少なくとも1つのタップ付き遅延線

50

のための前記プログラマブル入力データパス構成に基づいて、前記少なくとも1つの入力データフローパス中に選択的に供給されるように再構成されるように構成される、

C 1 1 に記載の V P E。

[C 1 3] 前記少なくとも1つのタップ付き遅延線は、前記少なくとも1つの実行ユニットによって実行される前記 V P E の各クロックサイクル上の前記少なくとも1つの入力データフローパス中に選択的に供給されるように再構成されるように構成される、

C 1 2 に記載の V P E。

[C 1 4] 前記少なくとも1つの実行ユニットは、前記少なくとも1つの実行ユニットのためのプログラマブル入力データパス構成に基づいて、前記入力ベクトルデータサンプルセットと異なるビット幅の入力ベクトルデータサンプルを処理するように構成可能である、

10

C 1 に記載の V P E。

[C 1 5] 少なくとも1つの実行ユニットは、

前記少なくとも1つのベクトルデータファイル中に一緒に記憶される前記少なくとも1つの出力データフローパス中の前記結果として生じるフィルタ化出力ベクトルデータサンプルのうちの実数の結果として生じるフィルタ化出力ベクトルデータサンプルを供給することと、

前記実数の結果として生じるフィルタ化出力ベクトルデータサンプルと別個に、前記少なくとも1つのベクトルデータファイル中に一緒に記憶される前記少なくとも1つの出力データフローパス中の前記結果として生じるフィルタ化出力ベクトルデータサンプルセットのうちの虚数の結果として生じるフィルタ化出力ベクトルデータサンプルを供給することと

20

を行うように構成される、C 1 に記載の V P E。

[C 1 6] 前記少なくとも1つの実行ユニットは、

前記少なくとも1つのベクトルデータファイル中に一緒に記憶される前記少なくとも1つの出力データフローパス中の前記結果として生じるフィルタ化出力ベクトルデータサンプルセットのうちの偶数の結果として生じるフィルタ化出力ベクトルデータサンプルを供給することと、

前記偶数の結果として生じるフィルタ化出力ベクトルデータサンプルと別個に、前記少なくとも1つのベクトルデータファイル中に一緒に記憶される前記少なくとも1つの出力データフローパス中の前記結果として生じるフィルタ化出力ベクトルデータサンプルセットのうちの奇数の結果として生じるフィルタ化出力ベクトルデータサンプルを供給することと

30

を行うように構成される、C 1 に記載の V P E。

[C 1 7] 前記少なくとも1つの実行ユニットにフィルタ係数を記憶するように構成されたレジスタファイルをさらに備え、

前記少なくとも1つの実行ユニットは、各処理ステージのための前記フィルタベクトル処理動作のための前記レジスタファイルからフィルタタップ係数を受信するようにさらに構成され、

前記少なくとも1つの乗算器は、各処理ステージのためのフィルタタップ出力ベクトルデータサンプルを生成するために、処理ステージのための前記受信したフィルタタップ係数と各受信した入力ベクトルデータサンプルセットを、前記少なくとも1つの乗算器中で乗算することによって、各処理ステージ中の前記シフトされた入力ベクトルデータサンプルセットの各々上に前記フィルタタップ動作を適用するようにさらに構成される、

40

C 1 に記載の V P E。

[C 1 8] 前記少なくとも1つの実行ユニットは、以下の式に従って、ディスクリート時間有限インパルス応答 (F I R) フィルタから成る前記フィルタベクトル処理動作を実行するように構成される、

【数 4】

$$y[n] = \sum_{l=0}^{l=Y-1} h(l) * x[n-l]$$

ここで、x [n] は、前記入力ベクトルデータサンプルセットを備え、y [n] は、前記結果として生じるフィルタ化出力ベクトルデータサンプルセットを備え、h (l) は、フィルタ係数から成り、Y は、前記フィルタ係数の数である、C 1 に記載の V P E。

10

[C 1 9] フィルタ処理動作を供給するように構成されたベクトル処理エンジン (V P E) であって、フィルタベクトル処理動作のための少なくとも 1 つの入力データフローパス手段中に入力ベクトルデータサンプルセットを供給するための手段と、記憶されるべき少なくとも 1 つの出力データフローパス手段から結果として生じるフィルタ化出力ベクトルデータサンプルセットを受信するための手段とを備える少なくとも 1 つのベクトルデータファイル手段と、前記少なくとも 1 つの入力データフロー手段中の少なくとも 1 つの実行ユニットと前記少なくとも 1 つのベクトルデータファイル手段との間に設けられた少なくとも 1 つのタップ付き遅延線手段、前記少なくとも 1 つのタップ付き遅延線手段は、複数の処理ステージの中の各処理ステージのためのシフトされた入力ベクトルベクトルデータサンプルを供給するために、前記フィルタベクトル処理動作中のフィルタタップの数に等しい前記複数の処理ステージの中の各処理ステージのための複数のパイプラインレジスタにおけるベクトルデータサンプル幅ぶん、前記入力ベクトルデータサンプルセットをシフトするための手段を備える、と、

20

前記少なくとも 1 つの入力データフローパス中に設けられた前記少なくとも 1 つの実行ユニット手段、前記少なくとも 1 つの実行ユニット手段は、前記フィルタベクトル処理動作の各フィルタタップのためのフィルタタップ出力ベクトルデータサンプルセットを生成するために、前記複数の処理ステージの中の各処理ステージのための前記シフトされた入力ベクトルデータサンプルセット上にフィルタタップ動作を適用するための手段と、

30

前記複数の処理ステージの中の各処理ステージのための少なくとも 1 つの累算器において前記フィルタタップ出力ベクトルデータサンプルセットを累算するための手段と、前記フィルタベクトル処理動作のための前記少なくとも 1 つの出力データフローパス手段上に前記結果として生じるフィルタ化出力ベクトルデータサンプルを供給するための手段とを備える、とを備える、 V P E。

40

[C 2 0] ベクトル処理エンジン (V P E) においてフィルタベクトル処理動作を実行する方法であって、フィルタベクトル処理動作のための少なくとも 1 つの入力データフローパス中に少なくとも 1 つのベクトルデータファイルからの入力ベクトルデータサンプルセットをフェッチすることと、前記少なくとも 1 つのベクトルデータファイルと少なくとも 1 つの実行ユニットとの間の前記少なくとも 1 つの入力データフローパス中に設けられる少なくとも 1 つのタップ付き遅延線中に前記少なくとも 1 つの出力データフローパス中の前記フェッチされた入力ベクトルデータサンプルセットを受信することと、複数の処理ステージの数が前記フィルタベクトル処理動作のフィルタタップの数に等し

50

い、前記少なくとも 1 つの実行ユニット中の前記複数の処理ステージの各処理ステージに対して、

現在の処理ステージ中の前記少なくとも 1 つの実行ユニットに、前記少なくとも 1 つの入力データフローパス中の前記少なくとも 1 つのタップ付き遅延線から前記入力ベクトルデータサンプルセットを供給することと、

フィルタタップ出力ベクトルデータサンプルセットを生成するために、前記現在の処理ステージ中の前記少なくとも 1 つの実行ユニット中の前記入力ベクトルデータサンプルセット上にフィルタタップ動作を適用することと、

前記現在の処理ステージ中の前の結果として生じるフィルタ化出力ベクトルデータサンプルセットと前記フィルタタップ出力ベクトルデータサンプルセットを累算することと

10

、
前記前の結果として生じるフィルタ化出力ベクトルデータサンプルセットを供給することと、

前記入力ベクトルデータサンプルセットとしてシフトされた入力ベクトルデータサンプルセットを供給するために、前記入力ベクトルデータサンプルセット中の前記フィルタベクトル処理動作のための次のフィルタタップに前記少なくとも 1 つのタップ付き遅延線内の前記入力ベクトルデータサンプルセットをシフトすることと

前記少なくとも 1 つのベクトルデータファイル中に記憶されるべき前記少なくとも 1 つのベクトルデータファイルに、少なくとも 1 つの出力データフローパス中の結果として生じるフィルタ化出力ベクトルデータサンプルセットとして前記前の結果として生じるフィ

20

ルタ化出力ベクトルデータサンプルセットを供給することと

を備える、方法。

[C 2 1] 前記入力ベクトルデータサンプルセット中の入力ベクトルデータサンプルの数が前記フィルタベクトル処理動作中の前記フィルタタップの数より小さい場合、前記フィルタベクトル処理動作の間に前記少なくとも 1 つのベクトルデータファイルからの前記入力ベクトルデータサンプルセットを再フェッチしない、

C 2 0 に記載の方法。

[C 2 2] 前記入力ベクトルデータサンプルセット中の入力ベクトルデータサンプルの数が前記フィルタベクトル処理動作中の前記フィルタタップの数より大きい場合、

前記フィルタベクトル処理動作のための前記少なくとも 1 つの入力データフローパス中の前記少なくとも 1 つのベクトルデータファイルからの次の入力ベクトルデータサンプルセットをフェッチすることと、

30

前記少なくとも 1 つのタップ付き遅延線中に、前記少なくとも 1 つのベクトルデータファイルから、前記少なくとも 1 つの入力データフローパス中の前記次のフェッチされた入力ベクトルデータサンプルセットを受信することと

をさらに備える、C 2 0 に記載の方法。

[C 2 3] 複数のシャドウパイプラインレジスタ中に、前記少なくとも 1 つの入力データフローパス中の前記少なくとも 1 つのベクトルデータファイルから、次の入力ベクトルデータサンプルセットを受信することと、

前記シフトされた入力ベクトルデータサンプルセットを供給するために、プライマリタップ付き遅延線中に、各処理ステージのための前記複数のシャドウパイプラインレジスタ中のベクトルデータサンプル幅ぶん、前記次の入力ベクトルデータサンプルセットをシフトすることと

40

をさらに備え、

前記プライマリタップ付き遅延線は、前記複数の処理ステージ中の各処理ステージのための前記少なくとも 1 つの実行ユニットに、前記少なくとも 1 つのデータフローパス中の前記シフトされた入力ベクトルデータサンプルセットを供給するように構成される、

C 2 0 に記載の方法。

[C 2 4] 前記少なくとも 1 つのタップ付き遅延線中の前記複数のパイプラインレジスタの中の隣接パイプラインレジスタ中の前記シフトされた入力ベクトルデータサンプルセッ

50

トを記憶するために、前記少なくとも1つのタップ付き遅延線中の前記隣接パイプラインレジスタ中に記憶された入力ベクトルデータサンプルと、前記少なくとも1つのベクトルデータファイルからの前記入力ベクトルデータサンプルセットからの入力ベクトルデータサンプルとの間で選択することをさらに備える、

C 2 0 に記載の方法。

[C 2 5] 前記少なくとも1つの実行ユニットによって実行されるべきベクトル命令に従って、前記少なくとも1つのタップ付き遅延線のためのプログラマブル入力データパス構成に基づいて前記少なくとも1つの実行ユニットと前記少なくとも1つのベクトルデータファイルとの間の前記少なくとも1つの入力データフローパス中の前記少なくとも1つのタップ付き遅延線を選択的に供給することをさらに備える、

10

C 2 0 に記載の方法。

[C 2 6] 前記少なくとも1つの実行ユニットによって実行される次のベクトル命令に従って、前記少なくとも1つのタップ付き遅延線のための前記プログラマブル入力データパス構成に基づいて、前記少なくとも1つの入力データフローパス中に選択的に供給される前記少なくとも1つのタップ付き遅延線を再構成することをさらに備える、

C 2 5 に記載の方法。

[C 2 7] 前記少なくとも1つの実行ユニットによって実行される前記 V P E の各クロックサイクル上の前記少なくとも1つの入力データフローパス中に選択的に供給される前記少なくとも1つのタップ付き遅延線を再構成することを備える、

20

C 2 6 に記載の方法。

[C 2 8] 前記少なくとも1つの実行ユニットのためのプログラマブル入力データフローパス構成に基づいて前記少なくとも1つの実行ユニット中の前記入力ベクトルデータサンプルセットからの入力ベクトルデータサンプルの異なるビット幅を処理することをさらに備える、

C 2 0 に記載の方法。

[C 2 9] 前記結果として生じるフィルタ化出力ベクトルデータサンプルセットとして前記前の結果として生じるフィルタ化出力ベクトルデータサンプルセットを供給することは、

前記少なくとも1つのベクトルデータファイル中に一緒に記憶されている前記少なくとも1つの出力データフローパス中の前記結果として生じるフィルタ化出力ベクトルデータサンプルセットのうちの実数の結果として生じるフィルタ化出力ベクトルデータサンプルセットを供給することと、

30

前記実数の結果として生じるフィルタ化出力ベクトルデータサンプルセットと別個に、前記少なくとも1つのベクトルデータファイル中に一緒に記憶される前記少なくとも1つの出力データフローパス中の前記結果として生じるフィルタ化出力ベクトルデータサンプルセットのうちの虚数の結果として生じるフィルタ化出力ベクトルデータサンプルセットを供給することと

を備える、C 2 0 に記載の方法。

[C 3 0] 前記結果として生じるフィルタ化出力ベクトルデータサンプルセットとして前記前の結果として生じるフィルタ化出力ベクトルデータサンプルセットを供給することは、

40

前記少なくとも1つのベクトルデータファイル中に一緒に記憶されている前記少なくとも1つの出力データフローパス中の前記結果として生じるフィルタ化出力ベクトルデータサンプルセットのうちの偶数の結果として生じるフィルタ化出力ベクトルデータサンプルセットを供給することと、

前記偶数の結果として生じるフィルタ化出力ベクトルデータサンプルセットと別個に、前記少なくとも1つのベクトルデータファイル中に一緒に記憶される前記少なくとも1つの出力データフローパス中の前記結果として生じるフィルタ化出力ベクトルデータサンプルセットのうちの奇数の結果として生じるフィルタ化出力ベクトルデータサンプルセットを供給することと

50

を備える、C 2 0 に記載の方法。

【図 1】

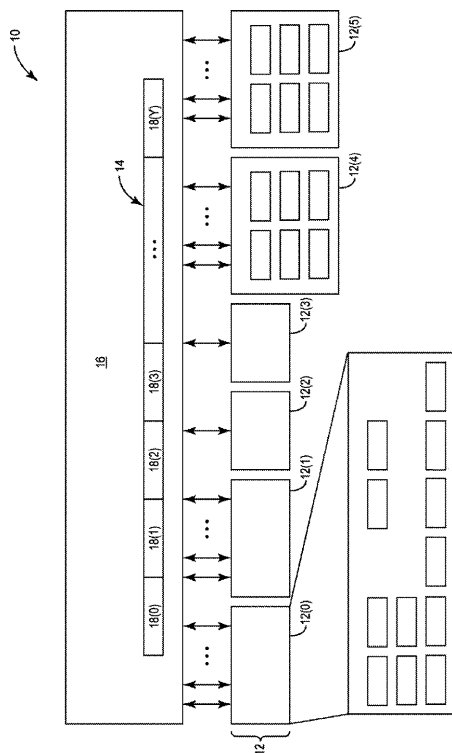


FIG. 1

【図 2】

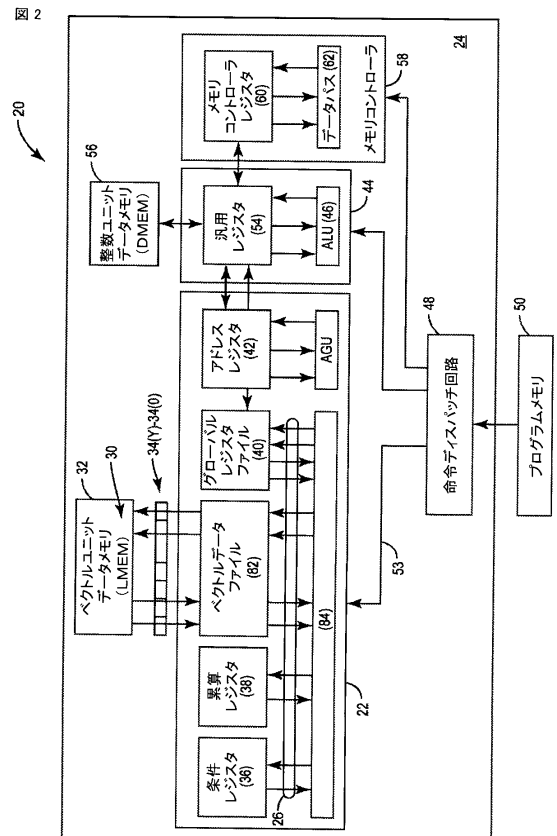
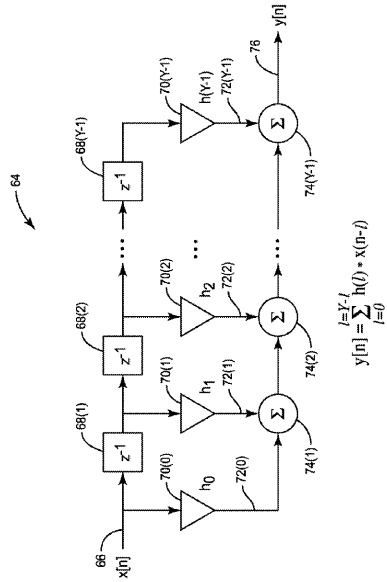


FIG. 2

【図 3】



【図 4】

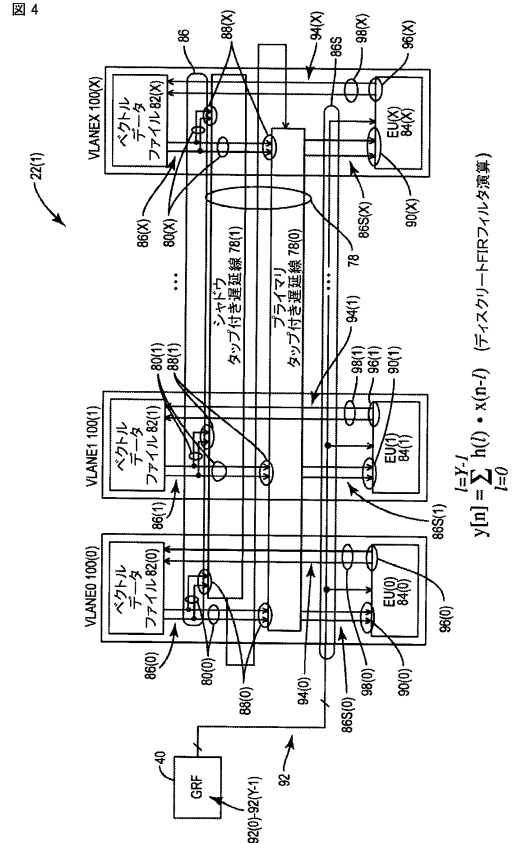
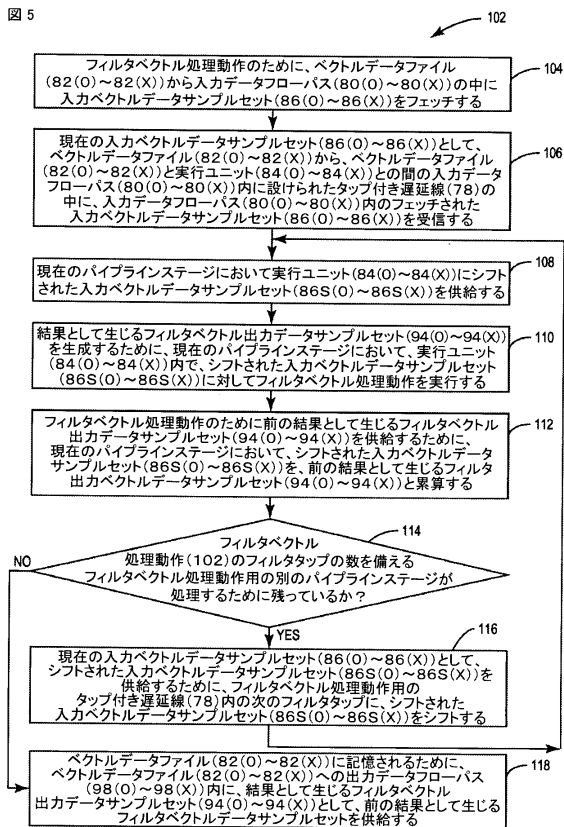


FIG. 4

【図 5】



【図 6 A】

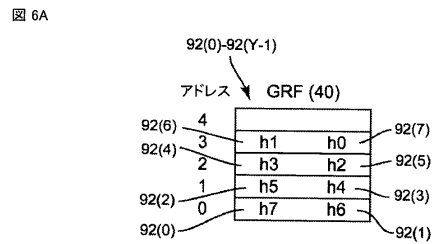
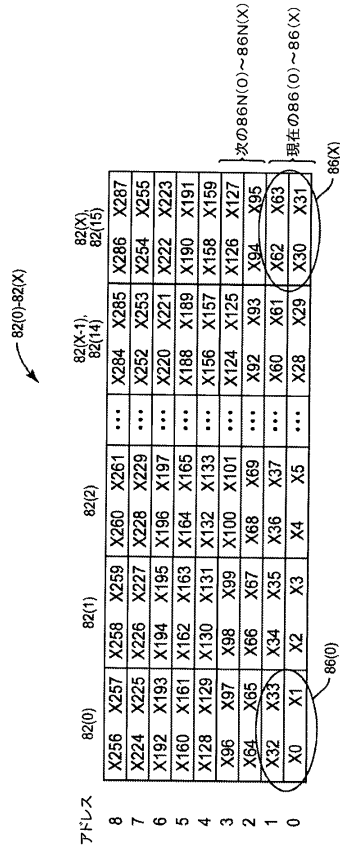


FIG. 6A

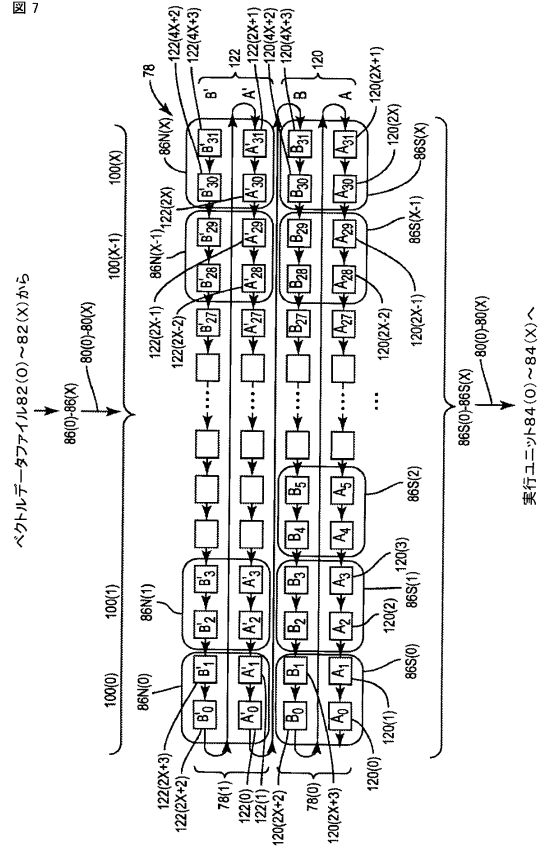
【 図 6 B 】

図 6B



【圖 7】

图 7



【 図 8 】

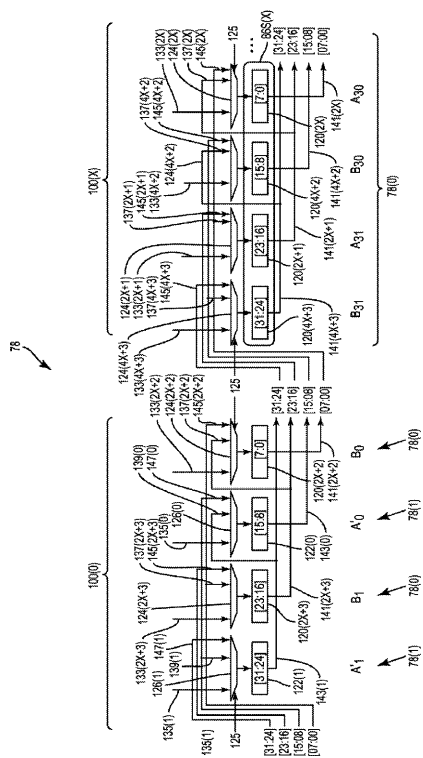


FIG. 8

【 図 9 A 】

図 9A

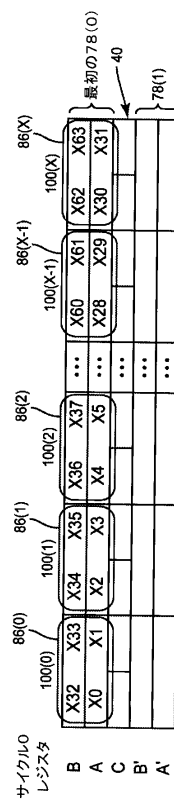
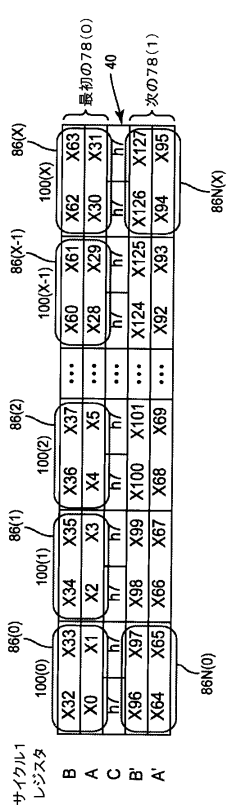


FIG. 9A

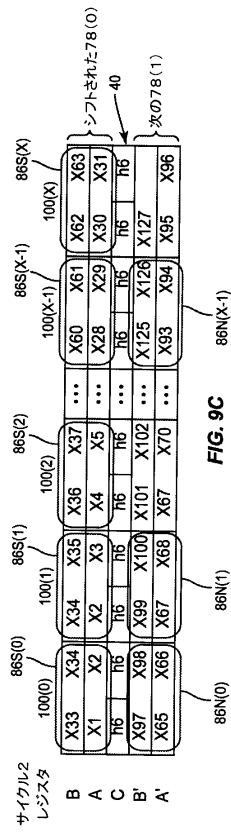
【 図 9 B 】

图 9B



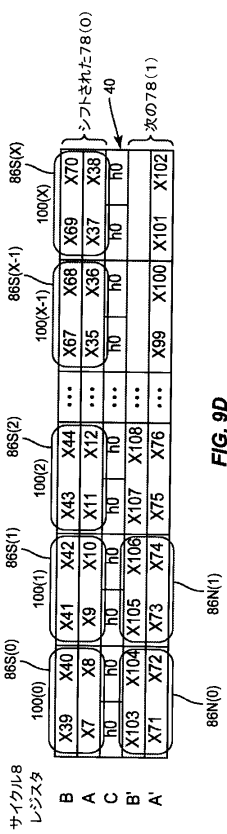
【 図 9 C 】

図 9C



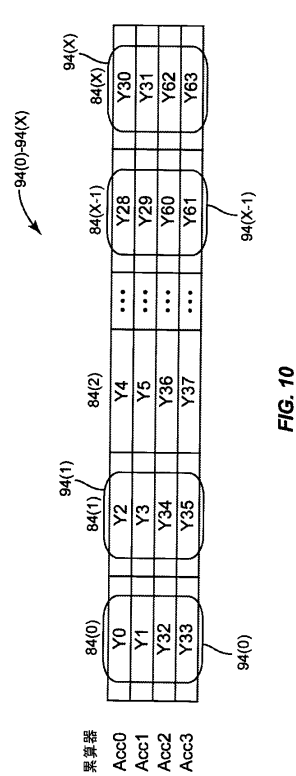
【 図 9 D 】

☒ 9D



【 図 1 0 】

图 10



【 図 1 1 】

图 11

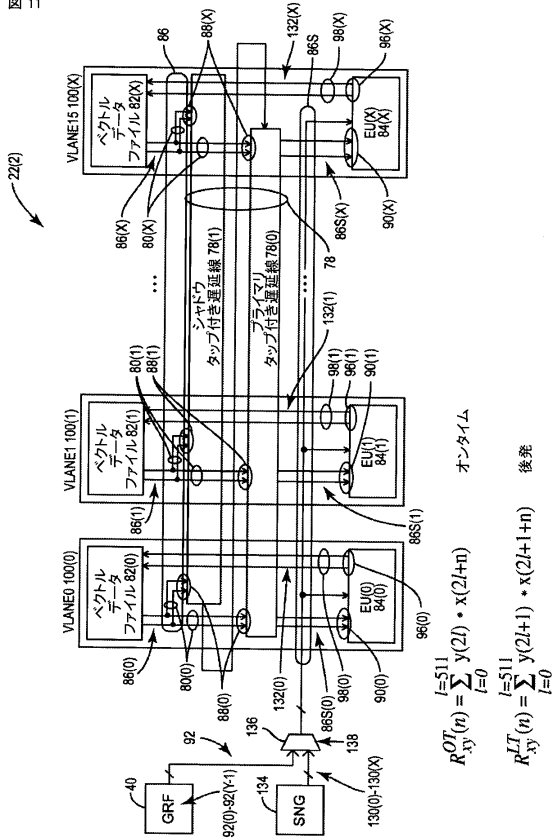


FIG. 11

【 図 1 2 A 】

图 12A

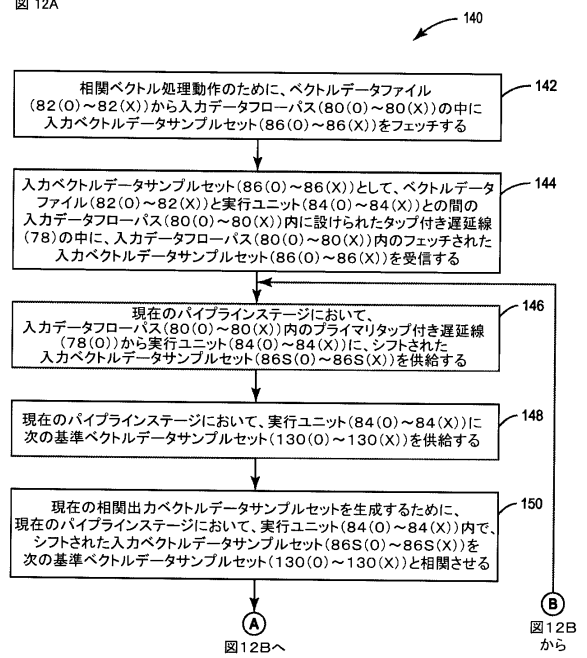


FIG. 12A

【 図 1 2 B 】

图 12B

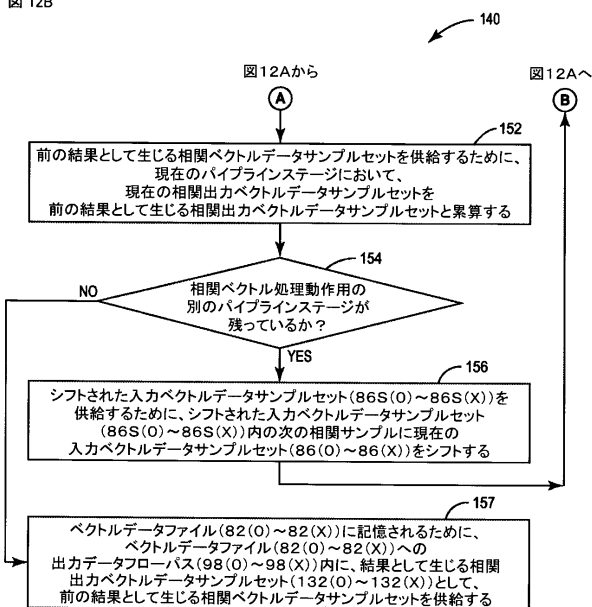


FIG. 12B

【 図 1 3 】

图 13

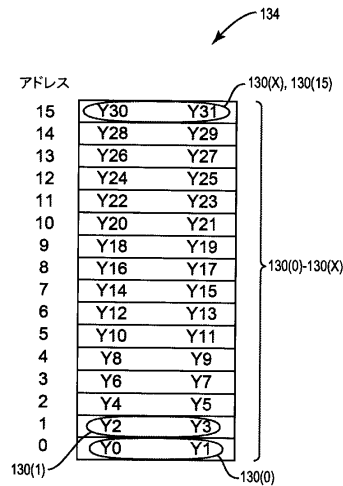


FIG. 13

【図 14】

図 14

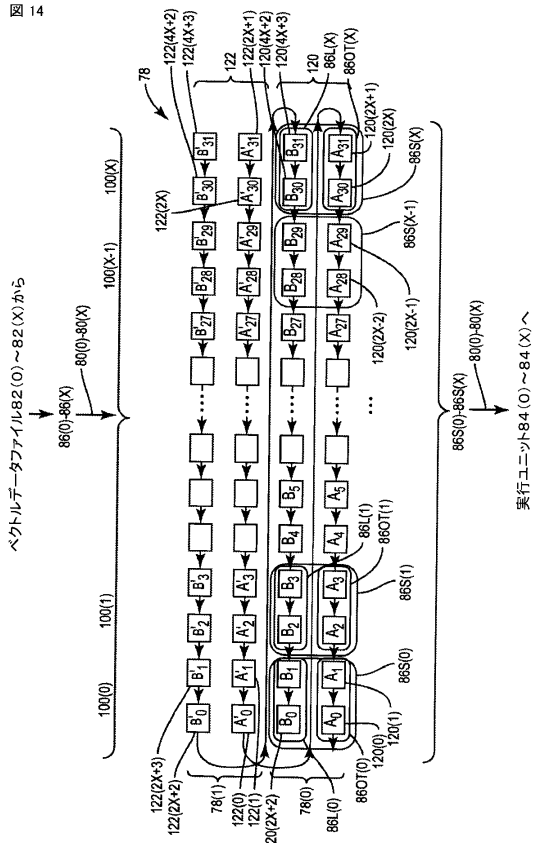
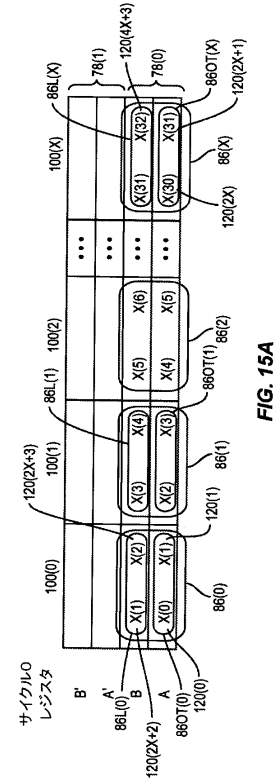


FIG. 14

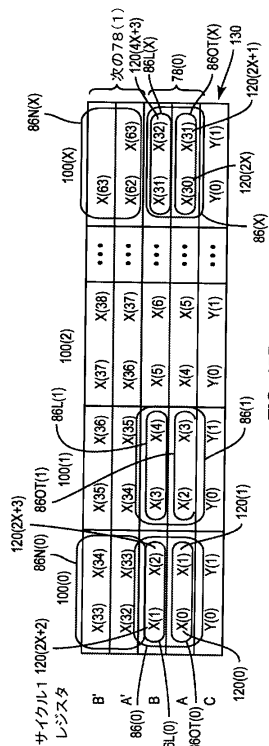
【図 15 A】

図 15A



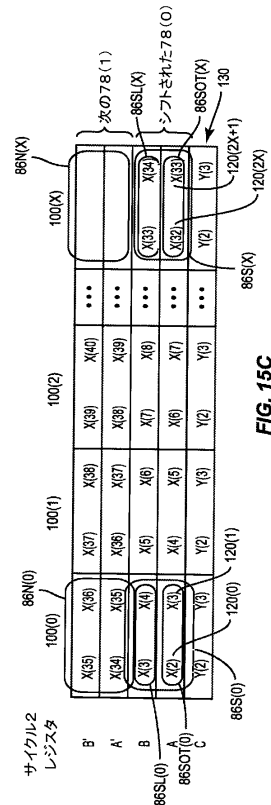
【図 15 B】

図 15B



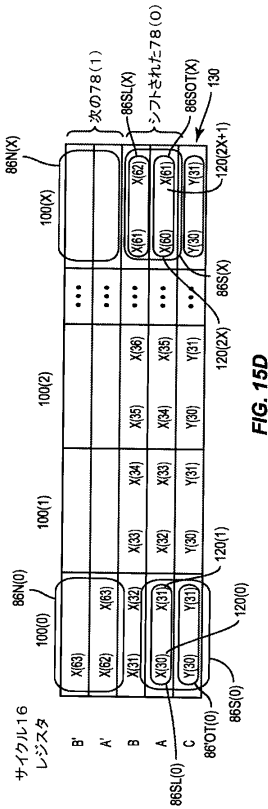
【図 15 C】

図 15C



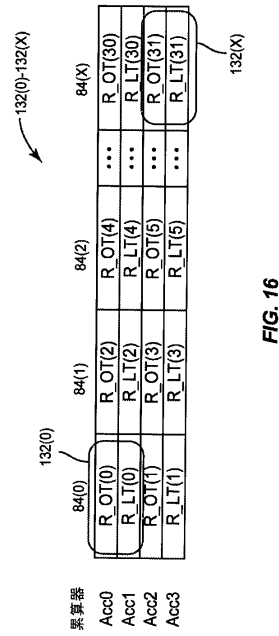
【 図 1 5 D 】

图 15D



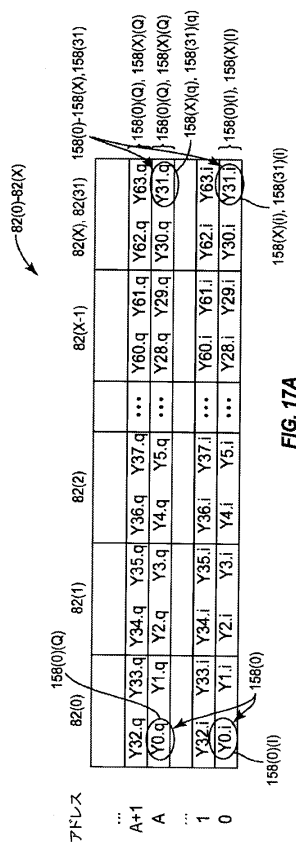
【 図 1 6 】

图 16



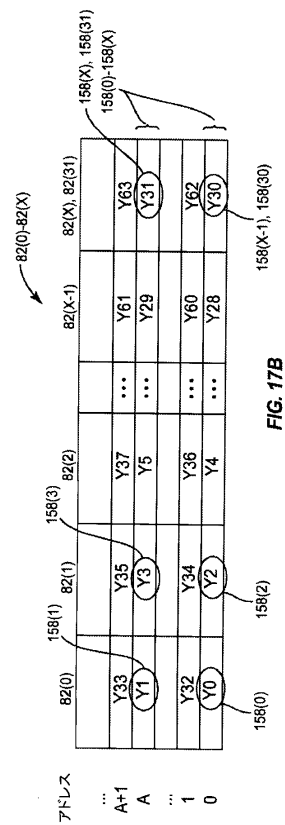
【 図 1 7 A 】

图 17A



【圖 17 B】

图 17B



【図 18 A】



FIG. 18A

【図 18 B】



FIG. 18B

【図 19】

図 19

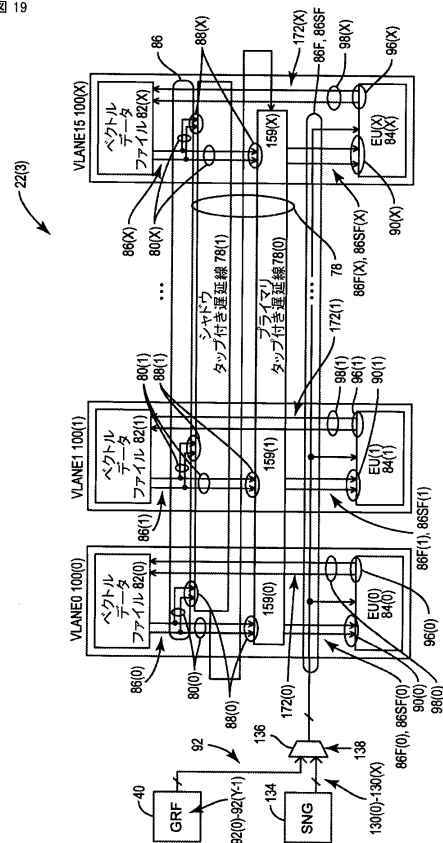


FIG. 19

【図 20】

図 20

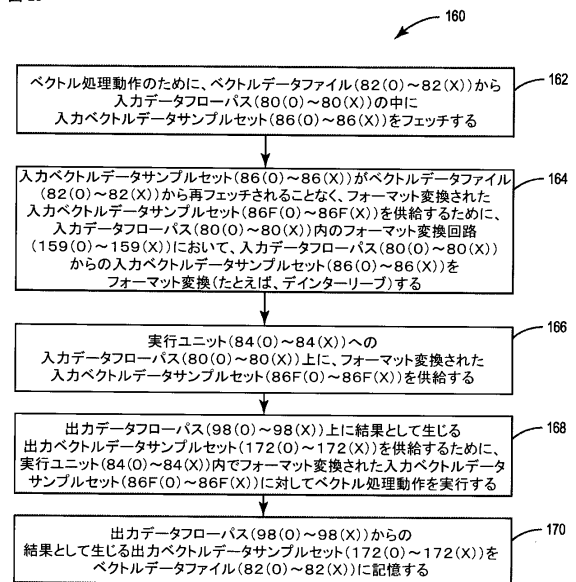


FIG. 20

【図 2 1】

図 21

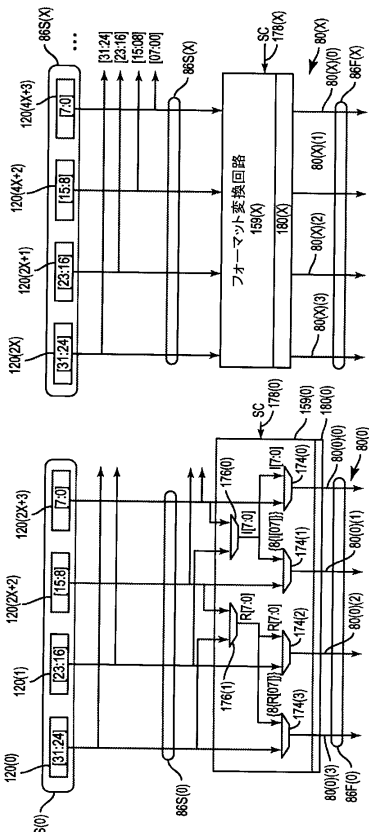


FIG. 21

【図 2 2】

図 22

| フィールド | ビット範囲 | 説明 |
|----------------|---------|---|
| BIAS_SC16 | [7:0] | SC16データフォーマットを使用するときの算術命令用のバイアス。バイアス範囲は-14~14に制限される。正の値は左シフトを示し、負の値は右シフトを示す |
| 準備 | [15:8] | 準備フィールド |
| DECIMATE_SRC1 | [16] | 第1のソースデータ用の縮小化およびフォーマット変換のビット 0 → 縮小化もフォーマット変換もしない 1 → SC8からSC16への縮小化および変換 |
| DECIMATE_SRC2 | [17] | 第2のソースデータ用の縮小化およびフォーマット変換のビット 0 → 縮小化もフォーマット変換もしない 1 → SC8からSC16への縮小化および変換 |
| DEST_FMT | [18] | 出力データのフォーマットを選択する 0 → 宛先はSC16フォーマットで記憶される 1 → SC16からSC8に出力を変換し、DECIMATE_PHASEフィールドにおいて指定されたように、偶数または奇数のいずれかの出力位置に書き込む |
| DECIMATE_PHASE | [19] | ソースと宛先の両方のための縮小化フェーズ 0 → 偶数サンプル(X0, X2など)を取る 1 → 奇数サンプル(X1, X3など)を取る |
| 準備 | [31:20] | 準備フィールド |

FIG. 22

【図 2 3】

図 23

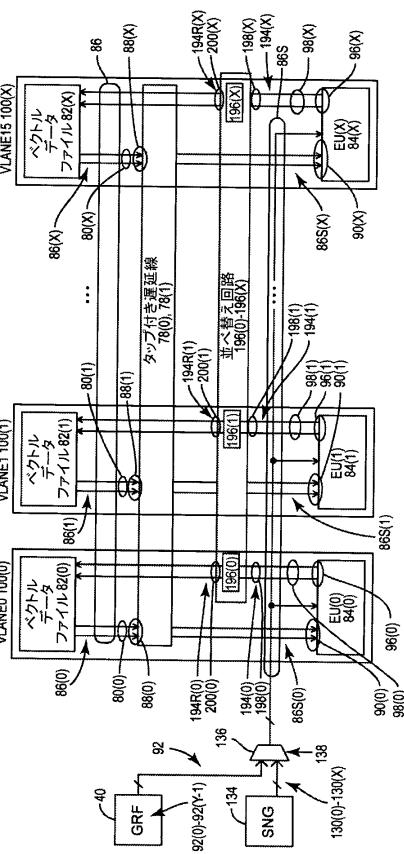


FIG. 23

【図 2 4】

図 24

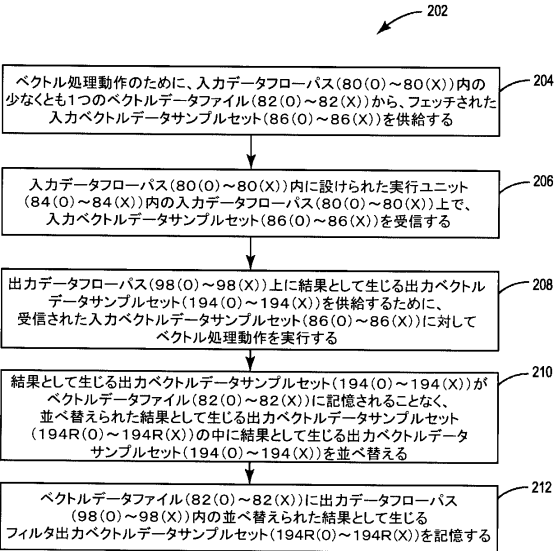
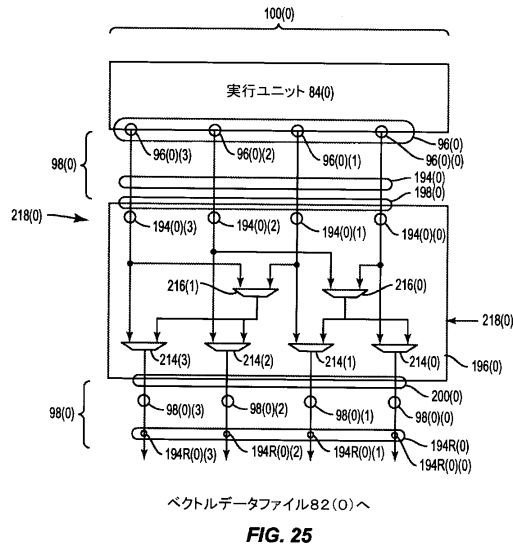


FIG. 24

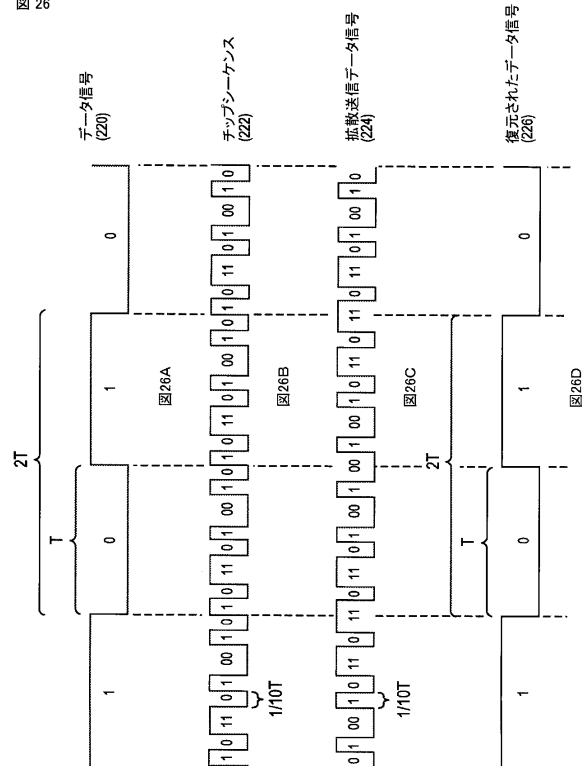
【 図 2 5 】

图 25



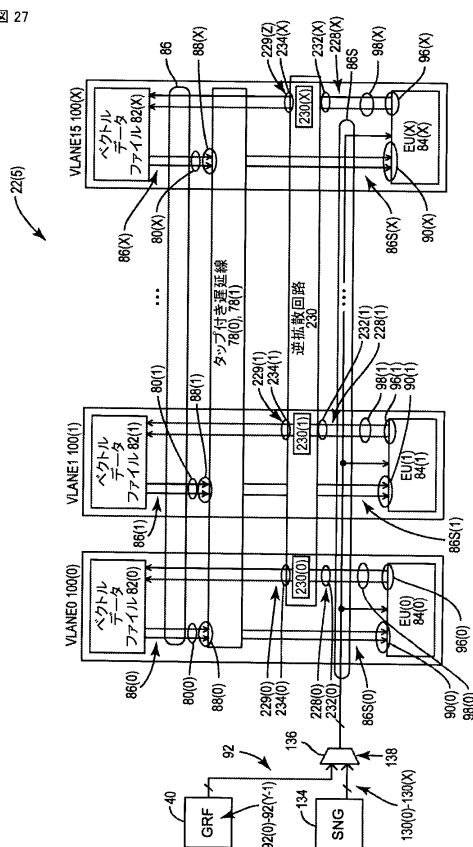
【 図 2 6 】

图 26



【 図 2 7 】

图 27



【 図 2 8 】

图 28

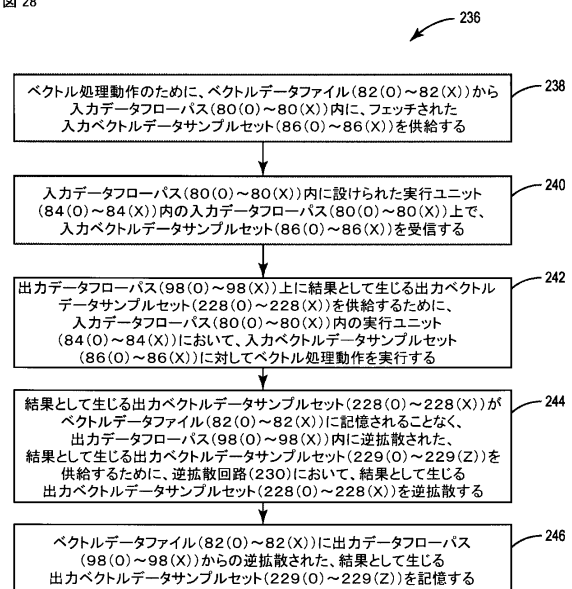


FIG. 28

【図 29】

図 29

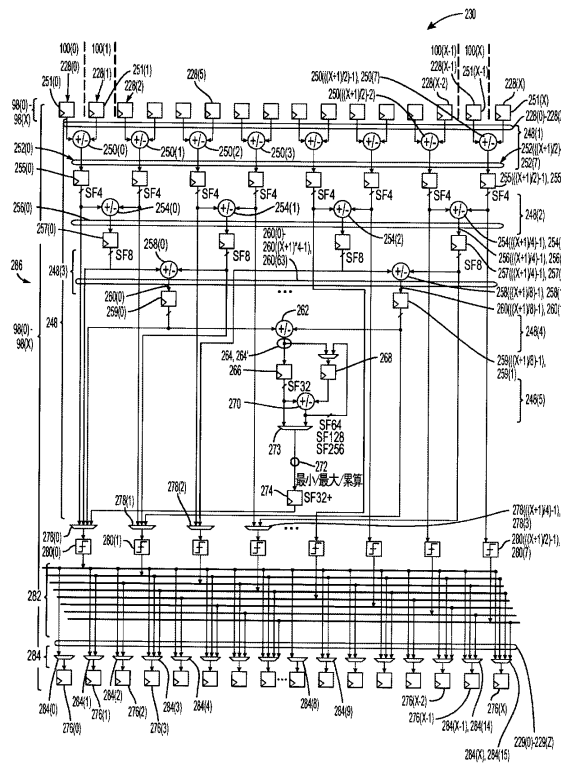


FIG. 29

【図 30】

図 30

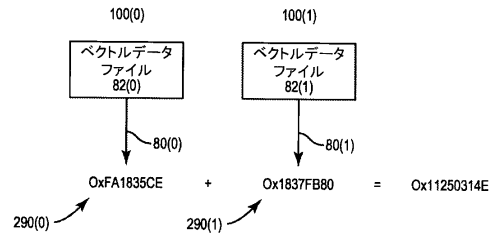


FIG. 30

【図 31】

図 31

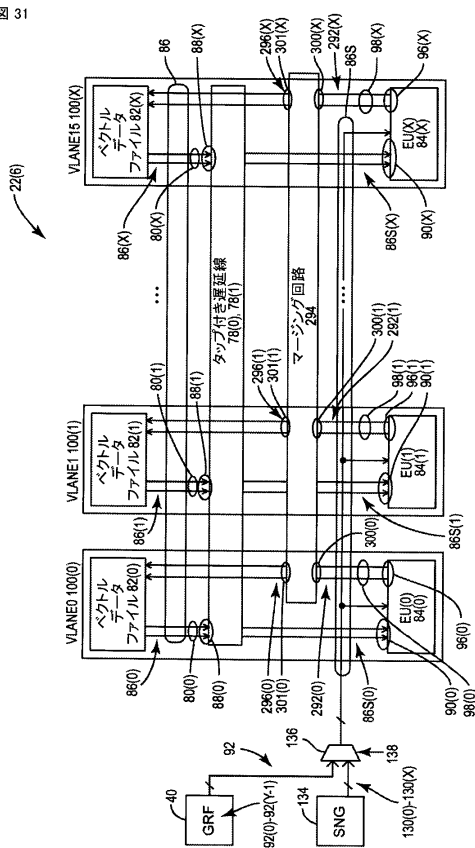


FIG. 31

【図 32】

図 32

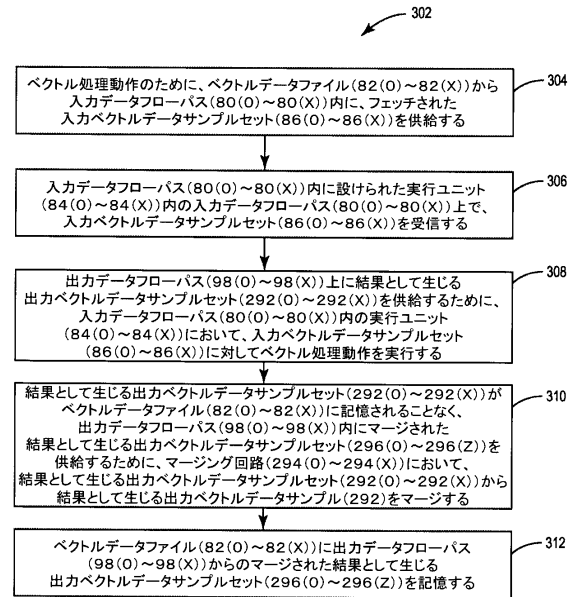


FIG. 32

【図 33】

図 33

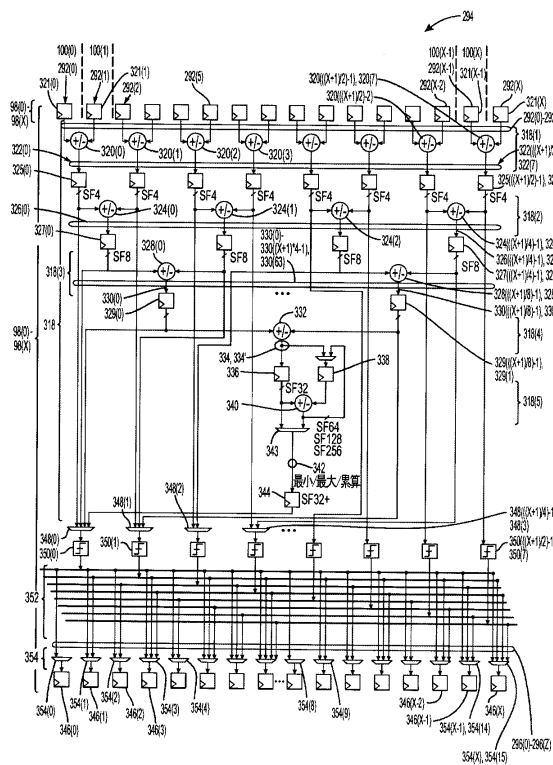
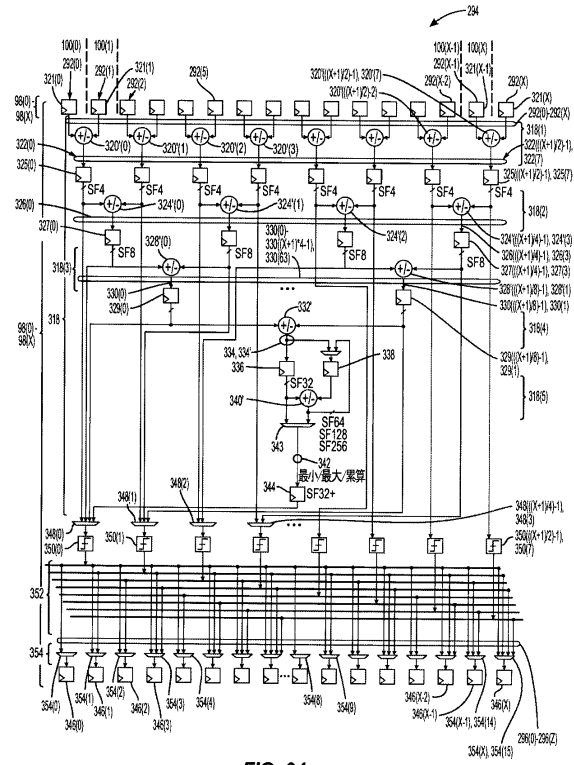


FIG. 33

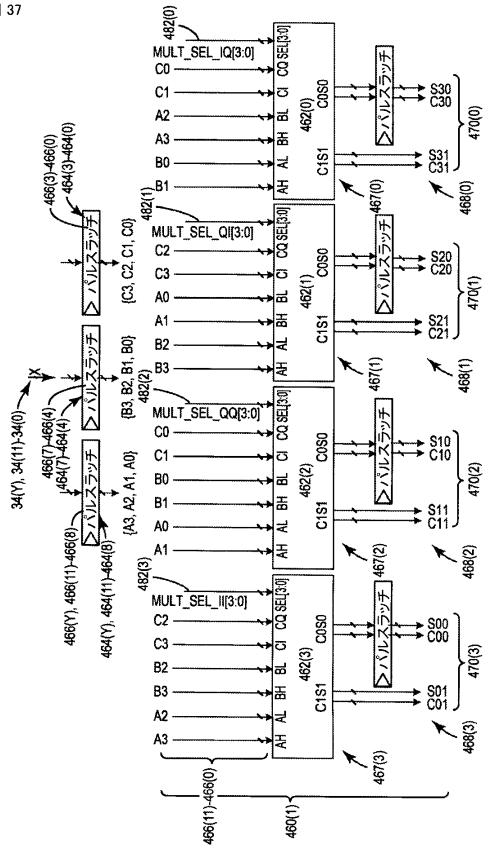
【図 34】

図 34



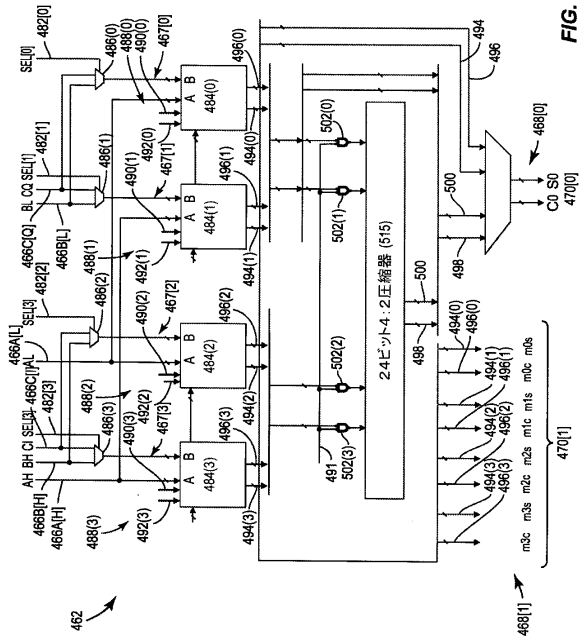
【 図 3 7 】

图 37



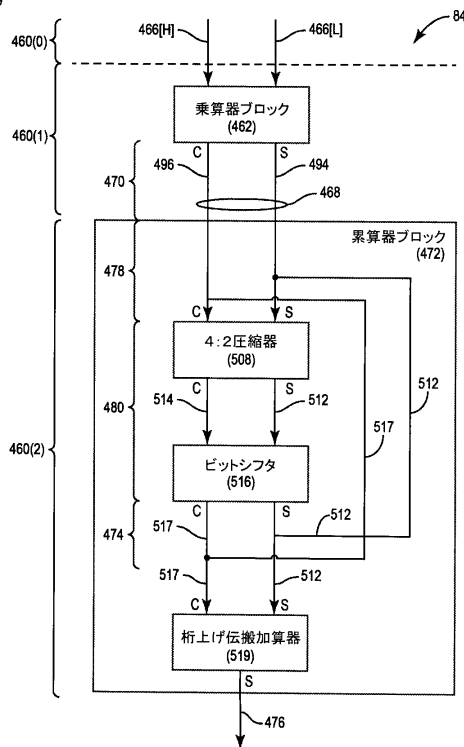
【 図 3 8 】

図 38



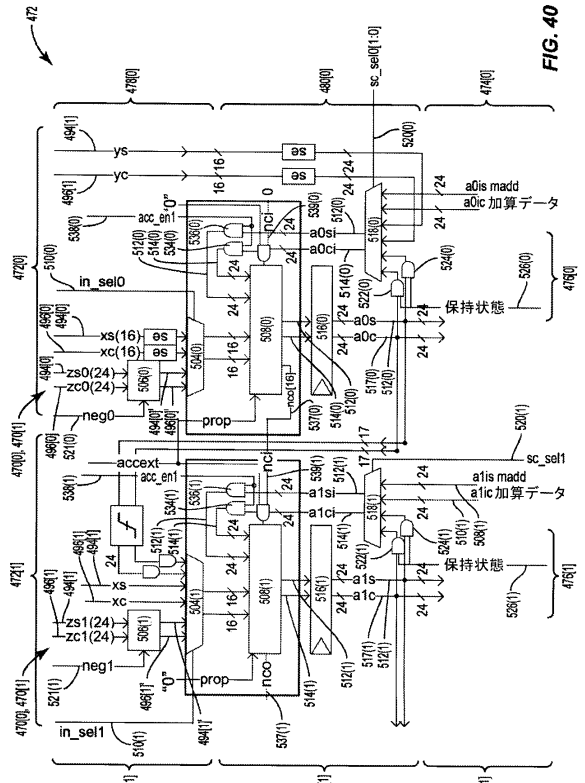
【 図 3 9 】

图 39



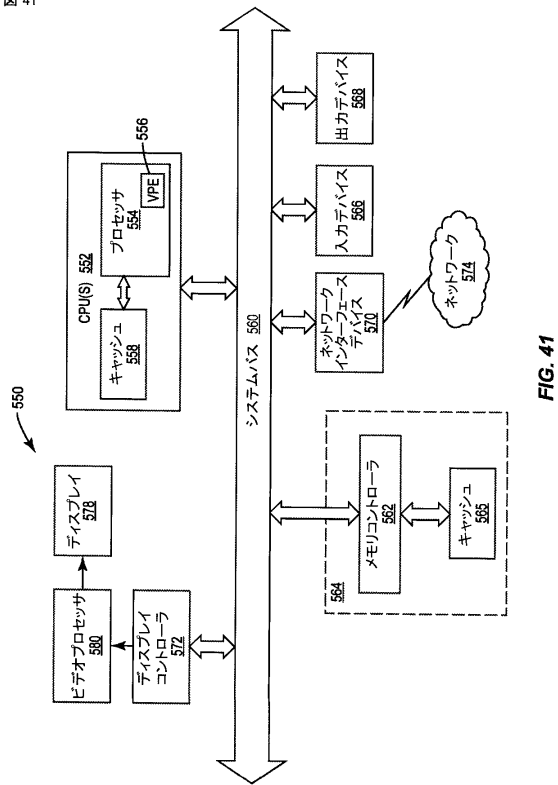
【 図 4 0 】

☒ 40



【図 41】

図 41



フロントページの続き

- (72)発明者 カーン、ラヘル
アメリカ合衆国、カリフォルニア州 9 2 1 2 1 - 1 7 1 4、サン・ディエゴ、モアハウス・ド
ラ イブ 5 7 7 5
- (72)発明者 ムジャヒド、ファハド・アリ
アメリカ合衆国、カリフォルニア州 9 2 1 2 1 - 1 7 1 4、サン・ディエゴ、モアハウス・ド
ラ イブ 5 7 7 5
- (72)発明者 シラビ、アフシン
アメリカ合衆国、カリフォルニア州 9 2 1 2 1 - 1 7 1 4、サン・ディエゴ、モアハウス・ド
ラ イブ 5 7 7 5

審査官 田中 幸雄

- (56)参考文献 特表 2 0 1 6 - 5 1 7 5 7 0 (J P , A)

- (58)調査した分野(Int.Cl. , D B 名)

G 0 6 F 1 7 / 1 6

G 0 6 F 9 / 3 0

G 0 6 F 9 / 3 8