

(19) 日本国特許庁(JP)

(12) 公表特許公報(A)

(11) 特許出願公表番号

特表2016-537726
(P2016-537726A)

(43) 公表日 平成28年12月1日(2016.12.1)

(51) Int.Cl.	F I	テーマコード (参考)
G06F 17/16 (2006.01)	G06F 17/16 D	5B013
G06F 9/38 (2006.01)	G06F 9/38 310G	5B033
G06F 9/30 (2006.01)	G06F 9/38 370A	5B056
	G06F 9/38 370C	
	G06F 9/30 350A	

審査請求 未請求 予備審査請求 有 (全 111 頁)

(21) 出願番号 特願2016-531030 (P2016-531030)
 (86) (22) 出願日 平成26年11月14日 (2014.11.14)
 (85) 翻訳文提出日 平成28年7月13日 (2016.7.13)
 (86) 国際出願番号 PCT/US2014/065825
 (87) 国際公開番号 W02015/073915
 (87) 国際公開日 平成27年5月21日 (2015.5.21)
 (31) 優先権主張番号 14/082,073
 (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)

(71) 出願人 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】 実行ユニットとベクトルデータメモリとの間のマーキング回路を備えるベクトル処理エンジンおよび関連する方法

(57) 【要約】

ベクトルデータメモリに記憶される出力ベクトルデータのインフライトマーキングを提供するために、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においてマーキング回路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の命令、システム、および方法も開示される。マーキング回路は、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る。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記憶されるために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ベクトル処理動作を実行する結果として、実行ユニットからの出力ベクトルデータサンプルセットをマーキングするように構成される。マーキングされた出力ベクトルデータサンプルセットは、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マーキングされた形式で記憶される。

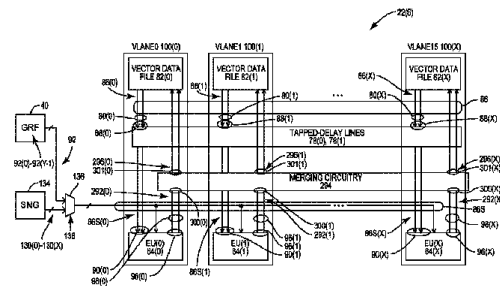


FIG. 31

【選択図】 図 3 1

【特許請求の範囲】

【請求項 1】

ベクトル処理動作を実行する少なくとも 1 つの実行ユニットによって生成された、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インフライトマージ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、

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内にフェッチ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、

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からの少なくとも 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セットを受信することと

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上で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、

前記少なくとも 1 つの出力データフローパス上に、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するために、前記入力ベクトルデータサンプルセットに対して前記ベクトル処理動作を実行することと

を行うように構成された、前記少なくとも 1 つの入力データフローパス内に設けられた少なくとも 1 つの実行ユニットと、

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、

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が前記少なくとも 1 つのベクトルデータファイルに記憶されることなく、少なくとも 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セットを供給するために、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マージすることと、

前記少なくとも 1 つの出力データフローパス上に前記少なくとも 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セットを供給することと

を行うように構成された、少なくとも 1 つのマージング回路と

を備える、 V P E 。

【請求項 2】

前記少なくとも 1 つのベクトルデータファイルは、

前記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内に前記少なくとも 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少なくとも 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 3】

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内の少なくとも 1 つのベクトルデータファイル出力上に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内の少なくとも 1 つのベクトルデータファイル入力上で前記少なくとも 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 つの実行ユニットは、

前記少なくとも 1 つの入力データフローパス内の少なくとも 1 つの実行ユニット入力上で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、

前記少なくとも 1 つの入力データフローパス内の少なくとも 1 つの実行ユニット出力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をコードシーケンスベクトルデータサンプルセットと乗算することと

10

20

30

40

50

を行うように構成され、

前記少なくとも1つのマーキング回路は、

前記少なくとも1つの実行ユニットから、前記少なくとも1つの入力データフローパス内の少なくとも1つのマーキング回路入力上で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、

前記少なくとも1つの出力データフローパス内の少なくとも1つのマーキング回路出力上で前記マーキングされた、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することと

を行うようにさらに構成される、請求項1に記載のVPE。

【請求項4】

10

前記マーキング回路は、前記少なくとも1つのマーキングされた、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するために、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内の少なくとも2つの、結果として生じる出力ベクトルデータサンプルをマーキングするように構成された少なくとも1つの加算器から構成される、

請求項1に記載のVPE。

【請求項5】

前記少なくとも1つの加算器は、加算器ツリー内に設けられた複数の加算器から構成され、前記複数の加算器の各々が、各々が異なるビット幅を有する複数の加算マーキングされた、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するように構成される、

請求項4に記載のVPE。

20

【請求項6】

前記マーキング回路は、前記少なくとも1つのマーキングされた、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するために、より大きいベクトルデータ値を有する、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間の結果として生じる出力ベクトルデータサンプルを最大マーキングするように構成された、少なくとも1つの最大ベクトルデータサンプル選択器から構成される、

請求項1に記載のVPE。

【請求項7】

前記少なくとも1つの最大ベクトルデータサンプル選択器は、各々が異なるビット幅を有する複数の最大マーキングされた、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するように各々が構成された、複数の最大値データサンプル選択器から構成される、

請求項6に記載のVPE。

30

【請求項8】

前記マーキング回路は、前記少なくとも1つのマーキングされた、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するために、より小さいベクトルデータ値を有する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間の、結果として生じる出力ベクトルデータサンプルを最小マーキングするように構成された、少なくとも1つの最小ベクトルデータサンプル選択器から構成される、

請求項1に記載のVPE。

40

【請求項9】

前記少なくとも1つの最小ベクトルデータサンプル選択器は、各々が異なるビット幅を有する複数の最小マーキングされた、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するように各々が構成された、複数の最小値データサンプル選択器から構成される、

請求項8に記載のVPE。

【請求項10】

前記マーキング回路は、前記少なくとも1つのマーキングされた、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うちの1つを選択するように構成されたマーキング選択器をさらに備える、

50

請求項 4 に記載の V P E。

【請求項 1 1】

前記コードシーケンスベクトルデータサンプルセットが、少なくとも 1 つの C D M A チップコードシーケンスから構成される、

請求項 1 に記載の V P E。

【請求項 1 2】

前記少なくとも 1 つのマーキング回路は、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選択的にマージするために、プログラム可能なマージデータパス構成入力に基づいて再構成されるように構成可能である、

請求項 1 に記載の V P E。

10

【請求項 1 3】

前記少なくとも 1 つのマーキング回路は、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べき前記 V P E のクロックサイクルごとに、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選択的にマージするために、前記プログラム可能なマージデータパス構成入力に基づいて再構成されるようにさらに構成される、

請求項 1 2 に記載の V P E。

【請求項 1 4】

前記少なくとも 1 つのマーキング回路は、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べき次のベクトル命令上で、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選択的にマージするために、前記プログラム可能なマージデータパス構成入力に基づいて再構成されるようにさらに構成される、

請求項 1 2 に記載の V P E。

20

【請求項 1 5】

前記少なくとも 1 つのマーキング回路は、複数のラッチをさらに備え、前記少なくとも 1 つのマーキング回路は、前記複数のラッチに前記少なくとも 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セットを記憶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 1 6】

前記少なくとも 1 つのマーキング回路は、前記複数のラッチの中の選択されたラッチに前記少なくとも 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セットを記憶するようにさらに構成される、

請求項 1 5 に記載の V P E。

30

【請求項 1 7】

前記少なくとも 1 つのマーキング回路は、前記複数のラッチに対応する複数の選択器をさらに備え、前記少なくとも 1 つのマーキング回路は、前記複数のラッチの中の前記選択されたラッチに前記少なくとも 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セットを記憶するために、前記複数の選択器の中の選択器を制御するように構成される、

請求項 1 6 に記載の V P E。

【請求項 1 8】

前記少なくとも 1 つのマーキング回路は、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ために、前記少なくとも 1 つの出力データフローパス内に前記少なくとも 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セットを供給する前に、前記複数のラッチに前記少なくとも 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セットを記憶するようにさらに構成される、

請求項 1 7 に記載の V P E。

40

【請求項 1 9】

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ム可能な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からの入力ベクトルデータサンプルの様々なビット幅を処理するように構成可能で

50

ある、

請求項 1 に記載の V P E 。

【請求項 2 0】

ベクトル処理動作を実行する少なくとも 1 つの実行ユニットによって生成された、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インフライトマージ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、

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バス手段内にフェッチ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、

記憶されるべき少なくとも 1 つの出力データフローバス手段からの少なくとも 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セットを受信するための手段と

10

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バス手段上で前記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、

前記少なくとも 1 つの入力データフローバス手段上に、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するために、前記入力ベクトルデータサンプルセットに対して前記ベクトル処理動作を実行するための実行手段と

を備える、前記少なくとも 1 つの入力データフローバス手段内に設けられた少なくとも 1 つの実行ユニット手段と、

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ための手段と、

20

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が前記少なくとも 1 つのベクトルデータファイル手段に記憶されることなく、少なくとも 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セットを供給するために、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前記コードシーケンスベクトルデータサンプルセットとマージするためのマージング手段と、

前記少なくとも 1 つの出力データフローバス手段上に前記少なくとも 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セットを供給するための手段と

を備える、少なくとも 1 つのマージング回路手段と

を備える、ベクトル処理エンジン (V P E) 。

【請求項 2 1】

30

ベクトル処理動作を実行する少なくとも 1 つの実行ユニットによって生成された、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インフライトマージする方法であって、

少なくとも 1 つのベクトルデータファイルから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バス内にフェッチ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、

前記少なくとも 1 つの入力データフローバス内に設けられた少なくとも 1 つの実行ユニット内の前記少なくとも 1 つの入力データフローバス上で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、

前記少なくとも 1 つの入力データフローバス上に、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するために、前記入力ベクトルデータサンプルセットに対して前記ベクトル処理動作を実行することと、

40

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が前記少なくとも 1 つのベクトルデータファイルに記憶されることなく、少なくとも 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セットを供給するために、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マージすることと、

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 つの出力データフローバスからの前記少なくとも 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セットを記憶することと

を備える、方法。

50

【請求項 2 2】

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前記マージすることは、前記少なくとも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セットを供給するために、少なくとも1つの加算器内で、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内のマージサンプルを加算することからさらに構成される、

請求項 2 1 に記載の方法。

【請求項 2 3】

前記少なくとも1つの加算器は、加算器ツリー内に設けられた複数の加算器から構成され、前記複数の加算器の各々は、各々が異なるビット幅を有する複数のマージされた、結果として生じる出力ベクトルデータサンプルセットを供給するように構成される、

10

請求項 2 2 に記載の方法。

【請求項 2 4】

前記少なくとも1つの出力データフローパス内に、前記少なくとも1つ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として供給するために、前記複数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うちの1つを選択することをさらに備える、

請求項 2 3 に記載の方法。

【請求項 2 5】

プログラム可能なマージデータパス構成入力を受信することと、

前記プログラム可能なマージデータパス構成入力に基づいて、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選択的にマージすることと

20

をさらに備える、請求項 2 1 に記載の方法。

【請求項 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べき V P E のクロックサイクルごとに、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選択的にマージすることをさらに備える、

請求項 2 5 に記載の方法。

【請求項 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べき次のベクトル命令のために、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選択的にマージすることをさらに備える、

30

請求項 2 5 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日に提出され、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。

40

【0002】

[0002]本出願は、2013年3月13日に提出され、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】

50

[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 RE-FETCHING 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 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 (VPES) EMPLOYING FORMAT CONVERSION CIRCUITRY IN DATA FLOW PATHS BETWEEN VECTOR DATA MEMORY AND EXECUTION UNITS TO PROVIDE IN-FLIGHT FORMAT-CONVERTING OF INPUT VECTOR DATA TO EXECUTION UNITS FOR VECTOR PROCESSING OPERATIONS, AND RELATED VECTOR PROCESSOR SYSTEMS AND METHODS」、124365と題する、米国特許出願第14/082,088号にも関連する。

【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 (VPES) EMPLOYING REORDERING CIRCUITRY IN DATA FLOW PATHS BETWEEN EXECUTION UNITS AND VECTOR DATA MEMORY TO PROVIDE IN-FLIGHT REORDERING OF OUTPUT VECTOR DATA STORED TO VECTOR DATA MEMORY, AND RELATED VECTOR PROCESSOR SYSTEMS AND METHODS」、124450と題する、米国特許出願第14/082,081号にも関連する。

【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 (VPES) EMPLOYING DESPREADING CIRCUITRY IN DATA FLOW PATHS BETWEEN EXECUTION UNITS AND VECTOR DATA MEMORY TO PROVIDE IN-FLIGHT DESPREADING OF SPREAD-SPECTRUM SEQUENCES, AND RELATED VECTOR PROCESSING INSTRUCTIONS, 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。

【技術分野】

10

20

30

40

50

【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。

【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、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(CDMA)、広帯域CDMA(WCDMA(登録商標))、およびロングタームエボリューション(LTE(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】

[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。

【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(PE)12を含む。この例では、6つの別個のPE12(0)~PE12(5)がベースバンドプロセッサ10内に設けられる。PE12(0)~PE12(5)は各々、共有メモリ16からPE12(0)~PE12(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

10

20

30

40

50

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、PE12(0)~PE12(5)は、高度の並列性を達成するために、PE12(0)~PE12(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PE12(0)~PE12(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(VR)を含む場合がある。

【0013】

[0013]図1のベースバンドプロセッサ10内の各PE12(0)~PE12(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WCDMAのPE12(0)、PE12(1)と、LTEのPE12(4)、PE12(5)とを含むが、これは、WCDMAおよびLTEが異なるタイプの特殊な演算を伴うからである。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。

10

20

【0014】

[0014]いくつかのワイヤレスベースバンド動作は、前の処理動作から決定されたデータサンプルのマージングを必要とする。たとえば、実行ユニットのデータ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ジングを提供するために、様々な実行ユニット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。これらのベクトル処理動作におけるベクトルデータサンプルは、ベクトルデータレーンと交差するデータパスを提供する複合ルーティングを含むことができる。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、複雑度が増大し、ベクトル処理エンジン(VPE)の効率が低減する可能性がある。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ジングを実行する回路を含むこともできる。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、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少利用される原因になる可能性がある。

30

【発明の概要】

【0015】

[0015]本明細書で開示される実施形態は、ベクトルデータメモリに記憶される出力ベクトルデータのインフライトマージングを提供するために、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においてマージング回路を利用するベクトル処理エンジン(VPE)を含む。関連するベクトル処理の命令、システム、および方法も開示される。マージング回路は、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る。マージング回路は、出力ベクトルデータサンプルセットが記憶されるために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ベクトル処理動作を実行する結果として、実行ユニットからの出力ベクトルデータサンプルセットをマージするように構成される。出力データサンプルセットのインフライトマージングは、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セット内の所望のプログラムされた出力ベクトルデータサンプルが、ベ

40

50

クトルデータメモリに記憶される前にマージ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。非限定的な例として、出力ベクトルデータのマージングは、マージされた出力ベクトルデータサンプルセットを供給する出力ベクトルデータサンプルセットと、出力スカラーデータサンプルセットとを加算することを含む場合がある。別の非限定的な例として、出力ベクトルデータサンプルセットのマージングは、実行ユニットからの比較された出力ベクトルデータサンプルセット間の最大出力ベクトルデータおよび/または最小出力ベクトルデータを生成することを含む場合がある。マージされた出力ベクトルデータサンプルセットは、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マージされた形式で記憶される。

10

【0016】

[0016]したがって、VPE内のデータフローパスの効率は、出力ベクトルデータのマージングによって制限されない。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マージされた形式で記憶されるべきとき、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。VPEはまた、実行ユニットのコンピュータ要素の効率に影響を与えることなく、ベクトルデータメモリ内の所望の宛先位置内に、マージされたベクトル内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。

20

【0017】

[0017]この関連で、一実施形態では、ベクトル処理動作を実行する少なくとも1つの実行ユニットによって生成された、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インフライトマージするように構成されたVPEが提供される。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、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内にフェッチ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。ベクトルデータファイルはまた、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からの少なくとも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セットを受信するように構成される。VPEはまた、少なくとも1つの入力データフローパス内に設けられた少なくとも1つの実行ユニットを備える。実行ユニットは、少なくとも1つの入力データフローパス上で、入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。実行ユニットはまた、少なくとも1つの出力データフローパス上に、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するために、入力ベクトルデータサンプルセットに対してベクトル処理動作を実行するように構成される。VPEはまた、少なくとも1つのマージング回路を含む。マージング回路は、結果として生じる出力ベクトルデータサンプルセットを受信するように構成される。マージング回路はまた、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ることなく、少なくとも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セットを供給するために、結果として生じる出力ベクトルデータサンプルセットをマージするように構成される。マージング回路はまた、少なくとも1つの出力データフローパス上に少なくとも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セットを供給するように構成される。

30

40

【0018】

[0018]別の実施形態では、ベクトル処理動作を実行する少なくとも1つの実行ユニットによって生成された、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インフライトマージするように構成されたVPEが提供される。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、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内にフェッチ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を備える。ベクトルデータファイル手段はまた、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の少なくとも1つのマージ

50

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受信するための手段を備える。V P E はまた、少なくとも1つの入力データフローバス手段内に設けられた少なくとも1つの実行ユニット手段を備える。実行ユニット手段は、少なくとも1つの入力データフローバス手段上で、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を備える。実行ユニット手段はまた、少なくとも1つの入力データフローバス手段上に、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するために、入力ベクトルデータサンプルセットに対してベクトル処理動作を実行するための実行手段を備える。

【0019】

【0019】さらに、V P E はまた、少なくとも1つのマーキング回路手段を備える。マーキング回路手段は、少なくとも1つの入力データフローバス手段上で、結果として生じる出力ベクトルデータサンプルセットを受信するための手段を備える。マーキング回路手段はまた、結果として生じる出力ベクトルデータサンプルセットが少なくとも1つのベクトルデータファイル手段に記憶されることなく、少なくとも1つのマーキングされた、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をコードシーケンスベクトルデータサンプルセットとマーキングするためのマーキング手段を備える。マーキング回路手段はまた、少なくとも1つの出力データフローバス手段上に少なくとも1つのマーキングされた、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するための手段を備える。

【0020】

【0020】別の実施形態では、ベクトル処理動作を実行する少なくとも1つの実行ユニットによって生成された、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インフライトマーキングする方法が提供される。方法は、少なくとも1つのベクトルデータファイルから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バス内にフェッチ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を備える。方法はまた、少なくとも1つの入力データフローバス内に設けられた少なくとも1つの実行ユニット内の少なくとも1つの入力データフローバス上で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。方法はまた、少なくとも1つの入力データフローバス上に、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するために、入力ベクトルデータサンプルセットに対してベクトル処理動作を実行することを備える。方法はまた、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ることなく、少なくとも1つのマーキングされた、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をマーキングすることを備える。方法はまた、少なくとも1つのベクトルデータファイル内に少なくとも1つの出力データフローバスからの少なくとも1つのマーキングされた、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記憶することを備える。

【図面の簡単な説明】

【0021】

【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。

【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。

10

20

30

40

50

【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。

【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。

【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。

【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 10】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 11】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 V P E の概略図。

【図 12 A】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 V P E 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 12 B】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 V P E 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 13】図 11 の V P E 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 V P E 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、V P E 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

10

20

30

40

50

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 1 5 A】 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 1 1 の V P E 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。

【図 1 5 B】 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 1 1 の V P E 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 1 5 C】 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 1 1 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。

10

【図 1 5 D】 相関 / 共分散ベクトル処理動作の第 1 4 の処理ステージの一部として、図 1 1 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 1 6】 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 1 1 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 7 A】 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。

20

【図 1 7 B】 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。

【図 1 8 A】 符号付き複素数の 1 6 ビットフォーマットで V P E 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 1 8 B】 符号付き複素数の 8 ビットフォーマットで V P E 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。

30

【図 1 9】 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 V P E の概略図。

【図 2 0】 図 1 9 の V P E 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

40

【図 2 1】 図 1 9 の V P E 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。

【図 2 2】 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 1 9 の V P E 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 2 3】 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

50

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。

【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インターリーブ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。

【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。

【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。

【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

10

20

30

40

50

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一つがプログラム可能なデータバ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バス構成を有し、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バス構成を有する、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バス構成を有する、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バス構成を有する、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】

【0022】

[0073]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、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。

【0023】

[0074]本明細書で開示される実施形態は、ベクトルデータメモリに記憶される出力ベクトルデータのインフライトマージングを提供するために、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においてマージング回路を利用するベクトル処理エンジン(VPE)も含む。関連するベクトル処理の命令、システム、および方法も開示される。マージング回路は、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る。マージング回路は、出力ベクトルデータサンプルセットが記憶されるために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ベクトル処理動作を実行する結果として、実行ユニットからの出力ベクトルデータサンプルセットをマージするように構成される。出力データサンプルセットのインフライトマージングは、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セット内の所望のプログラム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

10

20

30

40

50

る。非限定的な例として、出力ベクトルデータのマーキングは、マージされた出力ベクトルデータサンプルセットを供給する出力ベクトルデータサンプルセットと、出力スカラーデータサンプルセットとを加算することを含む場合がある。別の非限定的な例として、出力ベクトルデータサンプルセットのマーキングは、実行ユニットからの比較された出力ベクトルデータサンプルセット間の最大出力ベクトルデータおよび/または最小出力ベクトルデータを生成することを含む場合がある。マージされた出力ベクトルデータサンプルセットは、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マージされた形式で記憶される。

【0024】

[0075]したがって、VPE内のデータフローパスの効率は、出力ベクトルデータのマーキングによって制限されない。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マージされた形式で記憶されるべきとき、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。VPEはまた、実行ユニットのコンピュータ要素の効率に影響を与えることなく、ベクトルデータメモリ内の所望の宛先位置内に、マージされたベクトル内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。

【0025】

[0076]この関連で、図2は、ベクトル処理エンジン(VPE)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、VPE22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VPE22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VPE22を含む。このようにして、VPE22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VPE22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。

【0026】

[0077]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

10

20

30

40

50

命令実行時間を提供するようにレジスタ書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。

【0027】

[0078] 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 VPE 22 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、VPE 22 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、VPE 22 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、VPE 22 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。

10

【0028】

[0079] 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、VPE 22 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バス 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(DMEM) 56 がベースバンドプロセッサ 20 に含まれる。DMEM 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バス 62 を通ってメインメモリへのアクセスを求め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。

20

30

【0029】

[0080] 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。

40

【0030】

[0081] 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

50

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ート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($x[0], x[1], \dots, x[n]$)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、 $x[0], x[1], \dots, x[n]$)が各々フィルタ係数($h[0] \sim h(Y-1)$)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、 $x[0], x[1], \dots, x[n]$)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のディスクリートFIRフィルタ64は以下のように要約され得る。

10

【0031】

【数1】

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

【0032】

20

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、

$h(1)$ 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 $h(1)$ は複素数であり得る。一態様では、VPE22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。VPE22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。

30

【0033】

[0082] 図3の上記のディスクリートFIRフィルタ64は、以下のように書き直され得る。

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

【0034】

[0083] しかしながら、図3のディスクリート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バスに起因して、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。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バスは、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

40

50

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少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。

【0035】

[0084]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。

【0036】

[0085]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。

【0037】

[0086]「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、

10

20

30

40

50

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。

10

【0038】

[0087]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。

【0039】

[0088]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。

20

30

40

【0040】

[0089]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

50

入力データフローパス 80(0) ~ 80(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、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。

10

【0041】

[0090]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。

20

【0042】

[0091]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 VPE22(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 86S(0) ~ 86S(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。

30

40

【0043】

[0092]引き続き図 4 を参照すると、この実施形態における VPE22(1)は、並列化処理のための(VLANE0 100(0) ~ VLANEX 100(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 100(0) ~ 100(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 100(0)を取り上げると、その中のベ

50

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 86S(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、VPE22(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。

10

20

30

40

50

【0044】

[0093] 必要に応じて、任意の数のベクトルデータレーン 100(0) ~ 100(X)がVPE22(1)内に設けられる場合がある。VPE22(1)内に設けられるベクトルデータレーン 100(0) ~ 100(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 100(0) ~ 100(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 100がVPE22(1)内に設けられる場合があり、各ベクトルデータレーン 100は、VPE22(1)内の512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。

【0045】

[0094] 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 100(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。VPE22(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。

【0046】

[0095] 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。

【0047】

[0096] 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86S(0)~86S(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。

10

【0048】

[0097] 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(プロック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のディスクリフト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。

20

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

30

【0049】

[0098] 図6B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VPE22(1)内の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86(0)~86(X)を示す。この例では、サンプルX0は最も古いサンプルであり、サンプルX63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X63は、時間的にサンプルX0の後に発生する。ベクトルデータファイル82(0)~82(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82(0)~82(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86(0)~86(X)は、図6Bに示されたように、ADDRESS0およびADDRESS1にまたがる。これにより、ベクトルデータファイル82(0)~82(X)が、図4のVPE22(1)の例における実行ユニット84(0)~84(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86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86(0)~86(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、X0~X63)が存在する。同様に、ADDRESS2およびADDRESS3は、ベクトルデータファイル82(0)~82(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86(0)~86(X)を記憶する。図6Bのこの例では、各ベクトルデータファイル82(0)~82(X)の8個のアドレス(ADDRESS0~7)が示され、256個の合計入力ベクトルデータサンプル86(すなわち、X0~X255)を示

40

50

すが、それは限定的でないことに留意されたい。

【 0 0 5 0 】

[0099]フィルタベクトル処理動作 1 0 2 に関与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 1 0 2 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 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) がフィルタベクトル処理動作 1 0 2 に利用され得る。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 に利用され得る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 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。現在の例を説明する目的で、フィルタベクトル処理動作 1 0 2 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 1 0 0 (0) ~ 1 0 0 (X) を要すると想定する。

10

【 0 0 5 1 】

[00100]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バス 8 0 (0) ~ 8 0 (X) に供給される (ブロック 1 0 6) 。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 1 0 2 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 7 8 の目的は、フィルタベクトル処理動作 1 0 2 の次のフィルタタップ動作のために実行ユニット 8 4 (0) ~ 8 4 (X) に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) を供給するために、入力ベクトルデータサンプルセット 8 6 (0) ~ 8 6 (X) のシフトを提供することである。実行ユニット 8 4 (0) ~ 8 4 (X) によって実行されるフィルタベクトル処理動作 1 0 2 の各処理ステージの間、実行ユニット 8 4 (0) ~ 8 4 (X) に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) を供給するために、入力ベクトルデータサンプル 8 6 がプライマリタップ付き遅延線 7 8 (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 8 6 (0) ~ 8 6 (X) は、フィルタベクトル処理動作 1 0 2 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 8 2 (0) ~ 8 2 (X) 内でシフト、および再フェッチされる必要がない。

20

30

40

【 0 0 5 2 】

[00101]オプションのシャドウタップ付き遅延線 7 8 (1) が V P E 2 2 (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 8 6 N (0) ~ 8 6 N (X) も、ベクトルデータファイル 8 2 (0) ~ 8 2 (X) からシャドウタップ付き遅延線 7 8 (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 8 6 N (0) ~ 8 6 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 8 6 S (0) ~ 8 6 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 1 0 2 の間に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。このように、プライマリタップ付き遅延線 7 8 (0) は、フィルタベクトル処理動作 1 0 2 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 8 6 N (0

50

) ~ 86N(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 86S(0) ~ 86S(X) を有することができる。

【0053】

[00102]この関連で、図 7 は、図 4 の VPE22(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 86S(0) ~ 86S(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 86S(0) ~ 86S(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 86S(0) ~ 86S(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。

【0054】

[00103]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、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。

【0055】

[00104]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。

10

20

30

40

50

【0056】

[00105]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。

【0057】

[00106]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86S(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B₃₁およびB₃₀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A₃₁およびA₃₀である。図7に示されたように、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)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'

および A'_{10} 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_{10} 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。

【0058】

[00107]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 $82(0) \sim 82(X)$ から新しい入力ベクトルデータサンプルセット $86(0) \sim 86(X)$ 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 $120(0) \sim 120(4X+3)$ 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 $124(0) \sim 124(4X+3)$ 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 $122(0) \sim 122(4X+3)$ 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 $126(0) \sim 126(4X+3)$ が提供される。入力ベクトルデータサンプル選択器 $124(0) \sim 124(4X+3)$ および入力ベクトルデータサンプル選択器 $126(0) \sim 126(4X+3)$ 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 $124(0) \sim 124(4X+3)$ 、 $126(0) \sim 126(4X+3)$ は、各々、プライマリパイプラインレジスタ $120(0) \sim 120(4X+3)$ およびシャドウパイプラインレジスタ $122(0) \sim 122(4X+3)$ の中にロードまたはシフトされるべき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。

【0059】

[00108]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'_{10} 、 A'_{00} 、 B_{10} 、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。

【0060】

[00109]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 $VPE22(1)$ 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 $82(0) \sim 82(X)$ 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

10

20

30

40

50

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 VPE22(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 VPE22(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。

10

【0061】

[00110]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 VPE22(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) ~ 141(4X+3) および 143(0) ~ 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) ~ 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) ~ 126(4X+3) のために含まれる。

20

30

【0062】

[00111]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

40

50

うに、プライマリパイプラインレジスタ120(4X+2)(すなわち、B₃₀)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、B₀)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、A₃₁)に出力データフローパス141(2X+2)上でシフトされる。

【0063】

[00112]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、B₀)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、A₃₀)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、B₃₀)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、B₁)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、A₃₁)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、B₃₁)に出力データフローパス143(1)上でシフトされる。

【0064】

[00113]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。

【0065】

[00114]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86S(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6B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6B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)内の入力ベクトルデータサンプルセット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。

【0066】

[00115]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

10

20

30

40

50

(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。

10

【0067】

[00116]図9A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6B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ADDRESS0およびADDRESS1に記憶された。ひとえにこの例のための図4のVPE22(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、X0、X1、X32、およびX33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9A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を一括にグループ化して、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。

20

30

【0068】

[00117]図9B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86N(0)~86N(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86N(0)~86N(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86N(0)~86N(X)は、入力ベクトルデータサンプルX64~X127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86N(0)~86N(X)は、図6B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ADDRESS2およびADDRESS3に記憶された。ひとえにこの例のための図4のVPE22(1)

40

50

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、X64、X65、X96、および X97 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 9B 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。

【0069】

[00118] 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 10 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 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。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(4X+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 86N(0) ~ 86N(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) 20 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) ~ 82(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 102 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 102 におけるフィルタタップ 30 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 よりも大きい場合、フィルタベクトル処理動作 102 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。フィルタベクトル処理動作 102 が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) に対して完了した後、タップ付き遅延線 78(0)、78(1) 内に未処理の入力ベクトルデータサンプル 86S が存在する場合、実行ユニット 84(0) ~ 84(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 86S(0) ~ 86S(X) 40 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) ~ 86N(X) を供給され得る。

【0070】

[00119] シャドウタップ付き遅延線 78(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 102 が、ベクトルデータレン 50 ン 100(0) ~ 100(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 86 を要する場合、シャドウタップ付き遅延線 78(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 86(0) ~ 86(X) は、遅延がないフィルタベクトル処理動作 102 の間、実行ユニット 84(0) ~ 84(X) に利用可能である。フィルタベクトル処理動作 102 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 86S(0) ~ 86S(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 78(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 86N(0) ~ 86N(X) が、プライマリタップ付き遅延線 78(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 84(0) ~ 84(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 86N(0) ~ 86N(X) は、遅延なく利用可能である。ベクトルデータファイル 82(0) ~ 82(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 86(0) ~ 86(X) が、フィルタベクトル処理動作 102 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 84(0) ~ 84(X) は、フィルタベクトル処理動作 102 の間、十分に利用され続けることができる。

10

【0071】

[00120] 最初の入力ベクトルデータサンプルセット 86N(0) ~ 86N(X) および次の入力ベクトルデータサンプルセット 86N(0) ~ 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、プライマリタップ付き遅延線 78(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 86(0) ~ 86(X) は、フィルタベクトル処理動作 102 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) ~ 84(X) に供給される (図 5 のブロック 108)。最初の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 86(0) ~ 86(X) は、実行ユニット 84(0) ~ 84(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 86S(0) ~ 86S(X) になるために、プライマリタップ付き遅延線 78(0) 内でシフトされる。図 4 の VPE22(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 86S(0) は実行ユニット 84(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 86S(1) は実行ユニット 84(1) に供給され、以下同様である。

20

【0072】

[00121] 次に、実行ユニット 84(0) ~ 84(X) は、フィルタベクトル処理動作 102 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。

30

40

【0073】

[00122] 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)

50

）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。

10

20

30

40

50

【0074】

[00123]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)~84(X)に供給されたフィルタ係数92は、この例では「h6」である次のフィルタ係数92でもある。

【0075】

[00124]引き続き図5を参照すると、次のフィルタ係数92と乗算される（図5のブロック110）ために、実行ユニット84(0)~84(X)に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86S(0)~86S(X)を供給する（図5のブロック108）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94(0)~94(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94(0)~94(X)と累算される（図5のブロック112）。図9Dは、例示的なフィルタベクトル処理動作102の最後の処理ステージの間、タップ付き遅延線78(0)、78(1)内に存在する入力ベクトルデータサンプル86の状態を示す。図9Dに示されたこの例では、フィルタ係数92「h7」~「h0」（すなわち、92(0)~92(Y-1)）のせいで、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(Y)が存在した。図9Dに示されたように、「h0」はフィルタベクトル処理動作102における最後のフィルタ係数92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86S(0)~86S(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102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X39が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86S(0)に記憶される。

【0076】

[00125]上述されたフィルタベクトル処理動作102の例は、フィルタベクトル処理動

作 102 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 100 (0) ~ 100 (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 102 は、フィルタベクトル処理動作 102 に利用されるべき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。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) の幅よりも小さい場合があり、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。このシナリオでは、図 7 のタップ付き遅延線 78 (0)、78 (1) は、最後のベクトルデータレーン 100 (X) に到達するより前に、ベクトルデータレーン 100 内の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) として、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) をシフトするように、修正される必要があり得る。

10

20

30

40

50

【 0077 】

[00126] 図 10 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 84 (0) ~ 84 (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 94) の概略図である。この例では、各実行ユニット 84 (0) ~ 84 (X) は、ベクトルデータレーン 100 (0) ~ 100 (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 10 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 94 (0) ~ 94 (X) として、ベクトルデータファイル 82 (0) ~ 82 (X) に出力データフローパス 98 (0) ~ 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) ~ 82 (X) から図 2 のベクトルユニットデータメモリ 32 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) 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。

【 0078 】

[00127] フィルタベクトル処理動作 102 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 78 と同じまたは同様のタップ付き遅延線 78 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 86 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ベクトル拡散コード (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。

【 0079 】

[00128] 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少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。

10

【0080】

[00129]この関連で、図11は、図2のVPE22として提供され得る別の例示的なVPE22(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11のVPE22(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、VPE22(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、VPE22(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VPE22(1)に含まれるタップ付き遅延線78も、VPE22(2)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(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。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。

20

30

40

【0081】

[00130]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

50

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。

【0082】

[00131]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。

【0083】

[00132]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) 内の後発入力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、

【0084】

10

20

30

40

【数 2】

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

【0085】

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。

【0086】

[00133]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、

【0087】

【数 3】

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

【0088】

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。

【0089】

[00134]引き続き図11を参照すると、実行ユニット84(0)~84(X)は、各々、実行ユニット84(0)~84(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130(0)~130(X)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84(0)~84(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、VPE22(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82(0)~82(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86(0)、86(1)、...86(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、結果として生じる相関出力ベクトルデータサンプルセット132(0)~132(X)がもたらされる。

10

20

30

40

50

【 0 0 9 0 】

[00135]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関連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。関連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。

10

【 0 0 9 1 】

[00136]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。関連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、関連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、実行ユニット 8 4 (0) ~ 8 4 (X) に局在するタップ付き遅延線 7 8 によって供給される。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。

20

30

【 0 0 9 2 】

[00137]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 7 8 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 8 4 (0) ~ 8 4 (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 8 2 (0) ~ 8 2 (X) に記憶される必要がないからである。実行ユニット 8 4 (0) ~ 8 4 (X) からベクトルデータファイル 8 2 (0) ~ 8 2 (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 8 4 (0) ~ 8 4 (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 8 4 (0) ~ 8 4 (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 8 2 (0) ~ 8 2 (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

40

50

ットのための中間相関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、結果は前の相関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。

【0093】

[00138]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。

【0094】

[00139]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、例示的な相関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相関ベクトル処理動作 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 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相関 / 共分散ベクトル処理動作を示すフローチャートである。

10

【0095】

[00140]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 の相関ベクトル処理動作 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、相関ベクトル命令に従って相関ベクトル処理動作 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、相関ベクトル処理動作 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) と相関されるべき、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 6 B 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 86 (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。

20

30

【0096】

[00141]相関ベクトル処理動作 140 において相関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相関ベクトル処理動作 140 を提供するために、図 1 1 の V P E 2 2 (2) 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) が相関ベクトル処理動作 140 に利用され得る。相関ベクトル処理動作 140 は、相関ベクトル処理動作 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

40

50

るし、ここで、相関ベクトル処理動作 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、相関ベクトル処理動作 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、VPE 22 (2) 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。

【0097】

[00142] 図 12A に戻って参照すると、相関ベクトル処理動作 140 のための第 1 の入力ベクトルデータサンプルセット 86 S (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、相関ベクトル処理動作 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、相関ベクトル処理動作 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 86 N (0) ~ 86 N (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 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、相関ベクトル処理動作 140 の動作の間に、次の相関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 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相関ベクトル処理動作 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 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 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、相関ベクトル処理動作 140 の相関演算ごとに、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。

【0098】

[00143] この関連で、図 14 は、図 11 の VPE 22 (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 86 S (0) ~ 86 S (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 86 S (0) ~ 86

10

20

30

40

50

S (X) を受信し、それらの相関ベクトル処理動作 1 4 0 を実行することによって、十分利用される。

【 0 0 9 9 】

[00144] 相関ベクトル処理動作 1 4 0 のために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) 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 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 の数が、相関ベクトル処理動作 1 4 0 における相関演算の数よりも大きい場合、実行ユニット 8 4 (0) ~ 8 4 (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 8 6 (0) ~ 8 6 (X) がベクトルデータファイル 8 2 (0) ~ 8 2 (X) から再フェッチされることなく、相関ベクトル処理動作 1 4 0 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 1 4 0 における相関演算の数が、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 の数よりも大きい場合、相関ベクトル処理動作 1 4 0 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。

10

20

【 0 1 0 0 】

[00145] この実施形態では、プライマリパイプラインレジスタ 1 2 0 (0) ~ 1 2 0 (4 X + 3) はまとめて、ベクトルデータファイル 8 2 (0) ~ 8 2 (X) の幅である。1 5 に等しい「 X 」を有する幅が 5 1 2 ビットであるベクトルデータファイル 8 2 (0) ~ 8 2 (X) の例では、5 1 2 ビット（すなわち、6 4 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 6 4 個の合計プライマリパイプラインレジスタ 1 2 0 (0) ~ 1 2 0 (6 3) 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 7 8 (0) は、1 つ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 1 2 0 (0) ~ 1 2 0 (4 X + 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 8 6 (0) ~ 8 6 (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、1 6 ビットまたは 3 2 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 8 6 のサイズが相関ベクトル処理動作 1 4 0 に望ましい場合、入力ベクトルデータサンプルセット 8 6 (0) ~ 8 6 (X) は、プライマリパイプラインレジスタ 1 2 0 (0) ~ 1 2 0 (4 X + 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 1 2 0 によってシフトされ得る。

30

50

【 0 1 0 1 】

[00146] 図 1 5 A は、相関ベクトル処理命令 1 4 0 の第 1 のクロックサイクル (C Y C L E 0) の間に、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) 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) を示す。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) が入力ベクトルデータサンプル X 1 ~ X 3 2 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が、6 4 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 1 2 0 (0) ~ 1 2 0 (2 X + 1) (図 1 4 も参照) は、入力ベクトルデータサンプルセット 8 6 (0) ~ 8 6 (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 7 8 (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 7 8 (1) の中にも）、入力ベクトルデータサンプルセット 8 6 (0) ~ 8 6 (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

40

50

例えば、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)。

10

【0102】

[00147]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。

20

【0103】

[00148]図15Bは、関連ベクトル処理命令140の第2のクロックサイクル(CYC LE1)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。関連ベクトル処理動作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)~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)~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)~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)~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)~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)~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)~8

30

40

50

4 (X) へのレジスタ (「 C 」) 内で供給されるものとして示される。

【 0 1 0 4 】

[00149] 図 1 4 に戻って参照すると、 相関ベクトル処理動作 1 4 0 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 8 6 (0) ~ 8 6 (X) 内の入力ベクトルデータサンプル 8 6 がプライマリタップ付き遅延線 7 8 (0) 内でシフトされる時、 シャドウパイプラインレジスタ 1 2 2 に記憶された次の入力ベクトルデータサンプル 8 6 N も、 シャドウタップ付き遅延線 7 8 (1) のシャドウパイプラインレジスタ 1 2 2 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 8 6 (0) ~ 8 6 (X) の入力ベクトルデータサンプル 8 6 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 1 4 のタップ付き遅延線 7 8 (0) と 7 8 (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 7 8 (0) と 7 8 (1) との間で提供されるシフトパターンとは異なる。 図 1 4 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 8 6 O T は、 シャドウタップ付き遅延線 7 8 (1) 内のシャドウパイプラインレジスタ 1 2 2 (0) から、 プライマリタップ付き遅延線 7 8 (0) 内のプライマリパイプラインレジスタ 1 2 0 (2 X + 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 8 6 L は、 シャドウタップ付き遅延線 7 8 (1) 内のシャドウパイプラインレジスタ 1 2 2 (2 X + 2) から、 プライマリタップ付き遅延線 7 8 (0) 内のプライマリパイプラインレジスタ 1 2 0 (4 X + 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 8 6 のシフトが相関ベクトル処理動作 1 4 0 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 8 6 O T および後発入力ベクトルデータサンプル 8 6 O T は、 タップ付き遅延線 7 8 (0) 、 7 8 (1) 内で互いから隔離され続ける。

10

20

【 0 1 0 5 】

[00150] 相関ベクトル処理動作 1 4 0 の処理ステージが実行ユニット 8 4 (0) ~ 8 4 (X) において進行し、 最終的に、 シャドウタップ付き遅延線 7 8 (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 8 6 N (0) ~ 8 6 N (X) 全体は、 処理のために実行ユニット 8 4 (0) ~ 8 4 (X) に供給されるために、 プライマリタップ付き遅延線 7 8 (0) の中に完全にシフトされる。 このようにして、 相関ベクトル処理動作 1 4 0 が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) に対して完了した後、 実行ユニット 8 4 (0) ~ 8 4 (X) は、 次いで、 必要な場合、 遅延なく、 次の相関ベクトル処理動作 1 4 0 のための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、 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。

30

40

【 0 1 0 6 】

[00151] 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) 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、 図 1 5 B に示されたように、 プライマリタップ付き遅延線 7 8 (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 8 6 (0) ~ 8 6 (X) は、 相関ベクトル処理動作 1 4 0 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 1 2 A のブロック 1 4 6) 。 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) によって処理されている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。 図 1 1 の V P E 2 2 (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 8 6 (0) は実行ユニット 8 4 (0) に供給され、 現在の入力ベクトルデータサンプル 8 6 (1) は実行ユニット 8 4 (1) に供給され、 以下同様である。 相関ベクトル処理動作 1 4 0 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 8 6 (0) ~ 8 6 (X) と相関されるべき基準ベクトルデータ入力サンプル 1 3 0 (0) ~ 1 3 0 (X) が実行ユニット 8 4 (0) ~ 8 4 (X) に供給される (図 1 2 A のブロック 1 4 8) 。

【 0 1 0 7 】

50

[00152]次に、実行ユニット84(0)~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)~84(X)は、演算： $R(OT)[n] = y[0] * x[n]$ 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)~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)~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)~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)~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)~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)~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)~132(X)が前の入力ベクトルデータサンプルセット86(0)~86(X)になるにすぎない。

10

20

【0108】

[00153]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)~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、結果として生じる相関出力ベクトルデータサンプルセット132(0)~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)~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)~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)~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)~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プルセット132(0)~132(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132(0)~132(X)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)内で入力ベクトルデータサンプル86をシフトすることは、図14に関して詳細に上記で前述された。

30

【0109】

[00154]図15Cは、次の相関処理動作140、オンタイム入力ベクトルデータサンプル86SOTのための $R(OT)[n] = y[2] * x[2+n]$ および後発入力ベクトルデータサンプル86SL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86S(0)~86S(X)になるために、相関ベクトル処理動作140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)内の入力ベクトルデータサンプルセット86(0)~86(X)が、2つの入力ベクトルデータサンプル86によってシフトされる。たとえば、 $x(2)$ および $x(3)$ の図15Bの入力ベクトルデータサンプル86OT(1)が、次に図15Cの入力ベクトルデータサンプル86S(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86S(0)~86S(X)は、現在の入力ベクトルデータサンプルセット86(0)~86(X)になる。実

40

50

行ユニット 84(0) ~ 84(X) に供給された基準ベクトルデータサンプル 130 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 130 である。

【0110】

[00155]引き続き図 12B を参照すると、次の基準ベクトルデータサンプル 130 と乗算されるために、プライマリタップ付き遅延線 78(0) から (およびシャドウタップ付き遅延線 78(1) の一部分から) 実行ユニット 84(0) ~ 84(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 86S(0) ~ 86S(X) を供給する (図 12A のブロック 150) 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 132(0) ~ 132(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 132(0) ~ 132(X) と累算される (図 12B のブロック 152)。図 15D は、例示的な相関ベクトル処理動作 140 の最後の処理ステージの間に、タップ付き遅延線 78(0)、78(1) 内に存在する入力ベクトルデータサンプル 86 の状態を示す。この例では、図 15D に示されたように、タップ付き遅延線 78 のフルデータ幅は、入力ベクトルデータサンプルセット 86(0) ~ 86(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 86OT と後発入力ベクトルデータサンプル 86L との間で分割されるので、相関ベクトル処理動作 140 のための 16 個の処理ステージが存在した。図 15D 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 140 における最後の基準ベクトルデータサンプル 130(X) であり、それは、図 13 の例では基準ベクトルデータサンプル 130(15) 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 86S(0) ~ 86S(X) は (この例ではベクトルデータレーン 100(0) ~ 100(X) の幅である) 16 回シフトされており、その結果、相関ベクトル処理動作 140 のための最後の 16 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 78(0) 内のシフトされた入力ベクトルデータサンプル 86S(0) に記憶される。

10

20

【0111】

[00156]図 16 は、上記の例における例示的な 16 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 11 の VPE22(2) 内の実行ユニット 84(0) ~ 84(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 132) 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、132(0) ~ 132(X) として示される。この例では、各実行ユニット 84(0) ~ 84(X) は、ベクトルデータレーン 100(0) ~ 100(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 Acc0 ~ Acc3 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) ~ 132(X) として、ベクトルデータファイル 82(0) ~ 82(X) に出力データフローパス 98(0) ~ 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) ~ 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) ~ 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。

30

40

【0112】

[00157]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) ~ 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) ~ 132(X) とを含む、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) ~ 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) ~ 82(X) を提供するために、この例では

50

「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)~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)~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)~158(X)は、この例では158(0)、158(1)、. . .、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)~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)~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。

10

20

【0113】

[00158]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)~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)~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~Y31 158(0)~158(X)、158(31)は、ベクトルデータファイル82(0)~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。

30

【0114】

[00159]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

40

50

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。

10

【0115】

[00160]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(SC16)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(SC8)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。

20

30

【0116】

[00161]この関連で、図19は、図2のVPE22として提供され得る別の例示的なVPE22(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19のVPE22(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、VPE22(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、デインターリーブング)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82(0)~82(X)と実行ユニット84(0)~84(X)との間のベクトルデータレーン100(0)~100(X)の各々に、フォーマット変換回路159(0)~159(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86(0)~86(X)のデインターリーブ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84(0)~84(X)にフォーマット変換された入力ベクトル

40

50

ルデータサンプルセット 86F(0) ~ 86F(X) を供給するために、VPE22(3) 内のフォーマット変換回路 159(0) ~ 159(X) において、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) 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 86F 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 86F(0) ~ 86F(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 86 の処理用に VPE22(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。

【0117】

[00162] このようにして、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 82(0) ~ 82(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 86(0) ~ 86(X) のフォーマット変換は、ベクトルデータファイル 82(0) ~ 82(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 86(0) ~ 86(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 84(0) ~ 84(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、VPE22(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 86(0) ~ 86(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 86F(0) ~ 86F(X) は、実行ユニット 84(0) ~ 84(X) に局在化されるように供給される。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。

【0118】

[00163] 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) が図 19 の VPE22(3) 内に示されるが、図 19 の VPE22(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 19 に示されたように、フォーマット変換回路 159(0) ~ 159(X) は、オプションのプライマリタップ付き遅延線 78(0) に含まれ得る。この配置は、図 19 の VPE22(3) 内の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) 内にフォーマット変換回路 159(0) ~ 159(X) を設ける。プライマリタップ付き遅延線 78(0) 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、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 86F(0) ~ 86F(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、86SF(0) ~ 86SF(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。

【0119】

[00164] 図 19 の VPE22(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 VPE22(2) 内に設けられることに留意されたい。図 19 の VPE22(3) と図 11 の VPE22(2) との間の共通構成要素が、VPE22(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 VPE22(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 VPE22(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。

【0120】

[00165] 図 19 の VPE22(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 86F(0) ~ 8

10

20

30

40

50

6 F (X) を供給するためのタップ付き遅延線 7 8 のさらなる説明が次に記載される。この関連で、図 2 0 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 1 5 9 (0) ~ 1 5 9 (X) を利用する図 1 9 の V P E 2 2 (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 1 6 0 を示すフローチャートである。

【 0 1 2 1 】

[00166] 図 2 0 を参照すると、ベクトル命令に従うベクトル処理動作 1 6 0 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) 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) の中にフェッチされる (ブロック 1 6 2) 。たとえば、ベクトル処理動作 1 6 0 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 8 6 (0) ~ 8 6 (X) が、ベクトルデータファイル 8 2 (0) ~ 8 2 (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 8 6 F (0) ~ 8 6 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 1 6 0 であり得る。ベクトル処理動作 1 6 0 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 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 1 6 0 を提供するために、図 1 9 の V P E 2 2 (3) 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 1 6 0 に利用され得る。ベクトル処理動作 1 6 0 は、ベクトル処理動作 1 6 0 に利用され得る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 1 6 0 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。現在の例を説明する目的で、ベクトル処理動作 1 6 0 のための入力ベクトルデータサンプルセット 8 6 F (0) ~ 8 6 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 8 6 (0) ~ 8 6 (X) が、図 1 9 の V P E 2 2 (3) 内のすべてのベクトルデータレーン 1 0 0 (0) ~ 1 0 0 (X) を要すると想定する。

【 0 1 2 2 】

[00167] 引き続き図 2 0 を参照すると、ベクトル処理動作 1 6 0 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 8 6 (0) ~ 8 6 (X) が、フォーマット変換回路 1 5 9 (0) ~ 1 5 9 (X) への入力データフローパス 8 0 (0) ~ 8 0 (X) の中に供給される (ブロック 1 6 4) 。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 8 6 (0) ~ 8 6 (X) は、場合によっては、ベクトル処理動作 1 6 0 のために実行ユニット 8 4 (0) ~ 8 4 (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 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 (0) ~ 8 6 (X) は、場合によっては、実行ユニット 8 4 (0) ~ 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 1 6 0 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。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) も、ベクトル処理動作 1 6 0 の間にタップ付き遅延線 7 8 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、 $86SF(0) \sim 86SF(X)$ と指定される。

【0123】

[00168]引き続き図20を参照すると、実行ユニット $84(0) \sim 84(X)$ 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 $86F(0) \sim 86F(X)$ 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 $84(0) \sim 84(X)$ 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 $86F(0) \sim 86F(X)$ 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 $86F(0) \sim 86F(X)$ をシフトするために利用される場合、実行ユニット $84(0) \sim 84(X)$ 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 $86SF(0) \sim 86SF(X)$ 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 $86F(0) \sim 86F(X)$ 、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 $86SF(0) \sim 86SF(X)$ 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 $172(0) \sim 172(X)$ が、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、出力データフローバス $98(0) \sim 98(X)$ 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 $172(0) \sim 172(X)$ は、この例では $172(0)$ 、 $172(1)$ 、 \dots 、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。

10

20

【0124】

[00169]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 $86S(0) \sim 86S(X)$ を受信する例示的なフォーマット変換回路 $159(0) \sim 159(X)$ の概略図である。この例では、フォーマット変換回路 $159(0) \sim 159(X)$ は、入力データフローバス $80(0) \sim 80(X)$ 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 $159(0) \sim 159(X)$ が次に記載される。

30

【0125】

[00170]例示的なフォーマット変換回路 $159(0) \sim 159(X)$ 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 $159(1) \sim 159(X)$ 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 $86F(0)$ 、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 $86SF(0)$ 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 $86S(0)$ のデインターリーピングと符号拡張(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 $174(3) \sim 174(0)$ 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 $120(0) \sim 120(2X+3)$ に従って配置される。各マルチプレクサ $174(3) \sim 174(0)$ 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 $86S(0)$ 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 $86S(0)$ の部分のいずれかを選択するように構成される。

40

50

【 0 1 2 6 】

[00171]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86S(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86S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86S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86S(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86S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86S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86S(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86S(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。

10

20

【 0 1 2 7 】

[00172]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86F(0)~86F(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。

30

40

【 0 1 2 8 】

50

[00173]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。

【0129】

[00174]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

20

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。

【0130】

[00175]上記で説明されたように、入力データフローバス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バ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バス80(0)~80(X)に含まれるかどうかを制御するように、VPE22(3)にプログラミングを提供する。

【0131】

[00176]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、S

10

20

30

40

50

C8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VPE22(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VPE22(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。

【0132】

[00177]上記で説明されたように、VPE22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。

【0133】

[00178]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。

【0134】

[00179]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)

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。

【0135】

[00180] 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
 10 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
 20 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
 30 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
 40 リービングまたはデインターリーブを含む場合がある。

【0136】

[00181] 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
 40 (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
 50 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。

【0137】

[00182]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。

10

【0138】

[00183]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。

20

【0139】

[00184]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)~194(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196(0)~196(X)を利用する図23のVPE22(4)において実行され得るベクトル処理動作202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194(0)~194(X)の例示的な並び替えを示すフローチャートである。

30

【0140】

[00185]図23と図24とを参照すると、ベクトル命令に従うベクトル処理動作202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24のブロック204)。たとえば、ベクトル処理動作202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202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202を提供するために、図23のVPE22(4)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202に利用され得

40

50

る。ベクトル処理動作 202 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 202 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。

【0141】

[00186]引き続き図 23 と図 24 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 24 のブロック 206)。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 202 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を実行する (図 24 のブロック 208)。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 86 (0) ~ 86 (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 86 (0) ~ 86 (X) のシフトを伴う実行ユニット 84 (0) ~ 84 (X) によって実行されるベクトル処理動作 202 の各処理ステージの間のベクトル処理動作 202 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 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 86N (0) ~ 86N (X) は、場合によっては、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 202 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 86S (0) ~ 86S (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。

【0142】

[00187]引き続き図 23 と図 24 とを参照すると、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 202 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 86F (0) ~ 86F (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、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。

【0143】

[00188]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

10

20

30

40

50

(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、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。

10

20

【0144】

[00189]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。各出力ベクトルデータサンプル選択器214(3)~214(0)は、割り当てられた実行ユニット出力96(0)(3)~96(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194(0)の部分、または割り当てられた実行ユニット出力96(0)(3)~96(0)(0)に隣接する実行ユニット出力96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194(0)の部分のいずれかを選択するように構成される。

30

40

【0145】

[00190]たとえば、実行ユニット出力96(0)(3)~96(0)(0)が、16ビット符号付き複素数フォーマット、実数[31:24]、実数[23:16]、虚数[15:8]、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194(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[31:24]、虚数[23:16]、実数[15:8]、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214(3)~214(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214(3)は、出力データフローパス98(0)(3)上で供給するために、実行ユニット出力96(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194(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

50

択器 214(2)は、出力データフローパス 98(0)(2)上で供給するために、実行ユニット出力 96(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 194(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 194R(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 194R(0)(3)、194R(0)(2)として、隣接する出力データフローパス 98(0)(3)、98(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 194(0)(すなわち、実数[31:24]、虚数[23:16])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 214(0)は、出力データフローパス 98(0)(0)内で供給するために、実行ユニット出力 96(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 194(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 214(1)は、出力データフローパス 98(0)(1)上で供給するために、実行ユニット出力 96(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 194(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 194R(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 194R(0)(1)、194R(0)(0)として、隣接する出力データフローパス 98(0)(1)、98(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 194(0)(2)、194(0)(0)(すなわち、実数[15:8]、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 216(1)、216(0)は、図 25 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 96(0)(3)~96(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 194(0)(3)~194(0)(0)の間を選択する能力を提供する。

10

20

30

40

50

【0146】

[00191]引き続き図 23 と図 25 とを参照すると、並び替え回路 196(0)~196(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 194(0)~194(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 196(0)~196(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 196(0)~196(X)に直接流れる出力データフローパス 98(0)~98(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 22 に示されたように、出力ソースデータ(たとえば、図 23 の VPE22(4)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(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。

【0147】

[00192]この関連で、図 25 のプログラム可能な並び替えデータバス構成入力 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バス構成入力 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バス構成入力 218(0) ~ 218(X)は、VPE 22(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バス構成入力 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。

【0148】

[00193] 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。

【0149】

[00194] 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、シーケンス番号と相関される。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。

【0150】

[00195] 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。

【0151】

[00196] 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

10

20

30

40

50

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。

【0152】

[00197]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、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は、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。

【0153】

[00198]この関連で、図27は、図2のVPE22として提供され得る別の例示的なVPE22(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VPE22(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、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。

【0154】

[00199]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

10

20

30

40

50

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 VPE 22(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。

10

【0155】

[00200]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、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(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。

20

30

【0156】

[00201]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 VPE 22(5) 内のベクトルデータファイル 82(0) ~ 82(X) に記憶されることを必要とするはずであり、それにより、再フ

40

50

エ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。

【0157】

[00202] 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。

【0158】

[00203] 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。

【0159】

[00204] 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VPE22(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。

【0160】

[00205] 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VPE22(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

10

20

30

40

50

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。

【0161】

[00206]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VPE22(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VPE22(5)の出力データフローパス98(0)~98(X)内に供給される。

【0162】

[00207]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VPE22(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)。

【0163】

[00208]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

10

20

30

40

50

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)。

【0164】

[00209]図29は、図27のVPE22(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。

10

【0165】

[00210]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。

20

30

【0166】

[00211]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。

40

50

【 0 1 6 7 】

[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)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。

10

20

【 0 1 6 8 】

[00213]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) ~ 254($((X+1)/4) - 1$)、254(3)への第2の加算器ツリーレベル248(2)に供給される。この関連で、加算器254(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256(0)を供給するために、逆拡散ベクトルデータサンプル252(0)および252(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254(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256(1)を供給するために、逆拡散ベクトルデータサンプル252(2)および252(3)に対して逆拡散を実行するように構成される。加算器254($((X+1)/4) - 1$)、254(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256($((X+1)/4) - 1$)、256(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252($((X+1)/4) - 2$)、252($((X+1)/4) - 1$)、252(3)に対して逆拡散を実行するように構成される。加算器254(0) ~ 254($((X+1)/4) - 1$)、254(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256(0) ~ 256($((X+1)/4) - 1$)、256(3)は、ラッチ257(0) ~ 257($((X+1)/4) - 1$)、257(3)の中にラッチされる。

30

40

【 0 1 6 9 】

50

[00214]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 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 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 256 (0) ~ 256 ((X + 1) / 4) - 1)、256 (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 229 (0) ~ 229 (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 236 が 8 よりも高い拡散係数（たとえば、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 256 (0) ~ 256 ((X + 1) / 4) - 1)、256 (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 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 256 (0) ~ 256 ((X + 1) / 4) - 1)、256 (3) は、加算器 258 (0) ~ 258 ((X + 1) / 8) - 1)、258 (1) への第 3 の加算器ツリーレベル 248 (3) に供給される。この関連で、加算器 258 (0) は、それらのサンプルのための 16 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 256 (0) および 256 (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 258 (1) は、16 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 260 (0) ~ 260 ((X + 1) / 8) - 1)、260 (1) を供給するために、逆拡散ベクトルデータサンプル 256 (2) および 256 (3) に対して逆拡散を実行するように構成される。加算器 258 (0) ~ 258 ((X + 1) / 8) - 1)、258 (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 260 (0) ~ 260 ((X + 1) / 8) - 1)、260 (1) は、ラッチ 259 (0) ~ 259 ((X + 1) / 8) - 1)、259 (2) の中にラッチされる。

10

20

【0170】

[00215]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 16 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 260 (0) ~ 260 ((X + 1) / 8) - 1)、256 (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 229 (0) ~ 229 (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 236 が 16 よりも高い拡散係数（たとえば、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 260 (0) ~ 260 ((X + 1) / 8) - 1)、260 (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 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 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。

30

40

【0171】

[00216]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

50

カ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。

10

20

30

40

50

【0172】

[00217]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。

【0173】

[00218]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。

10

【0174】

[00219]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。

20

30

【0175】

[00220]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

40

50

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。

10

20

30

40

50

【0176】

[00221]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。

【0177】

[00222]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ジングを必要とする。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ジ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ジ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少利用される原因になる可能性がある。

【 0 1 7 8 】

[00223]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)を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。

10

【 0 1 7 9 】

[00224]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ジング回路を含むVPEを含む。マージング回路は、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ジングは、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に、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ジ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。

20

30

【 0 1 8 0 】

[00225]この関連で、図31は、図2のVPE22として提供され得る別の例示的なVPE22(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31のVPE22(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、VPE22(6)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292(0)~292(X)のインフライトマージ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、結果として生じる出力ベクトルデータサンプル292(0)、...、292(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292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292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292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31のVPE22(6)では、結果として生じる出

40

50

カベクトルデータサンプルセット 292 (0) ~ 292 (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 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。

【0181】

[00226] マージング回路 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 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 のインフライトマージ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 296 (0) ~ 296 (Z) は、出力データフローパス 98 (0) ~ 98 (X) 内のマージング回路 294 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 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) 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 296 (0) ~ 296 (Z) は、この例では 296 (0)、...、および 296 (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 296 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 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) に対して実行されるマージング動作に依存する。図 31 の VPE 22 (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 292 のインフライトマージングは、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 292 (0) ~ 292 (X) 内の、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 296 (0) ~ 296 (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 296 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) として、マージされた形式でベクトルデータファイル 82 (0) ~ 82 (X) に記憶され得る。

【0182】

[00227] このように、出力データフローパス 98 (0) ~ 98 (X) 内に設けられたマージング回路 294 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 292 (0) ~ 292 (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。

【0183】

[00228]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)内にマーキング回路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のマーキ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)においてマーキ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のマーキングは、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VPE22(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/またはマーキ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。

10

20

【0184】

[00229]図4、図11、図19、図23および図27のVPE22(1)~22(5)内に設けられた共通構成要素が、図31のVPE22(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VPE22(6)において示される。VPE22(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VPE22(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。

【0185】

[00230]引き続き図31を参照すると、より具体的には、マーキング回路294は、出力データフローパス98(0)~98(X)上のマーキング回路入力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)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、マーキ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。マーキング回路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)内のマーキング回路出力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。

30

40

【0186】

[00231]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

50

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。

【0187】

[00232] 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 10
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 20
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。

【0188】

[00233] 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。
。

【0189】

[00234] 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ジング回路 294 に供給される。マージング回路 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 のマージングを要求する場合、出力データフロ 40
50

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に、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)。

10

【0190】

[00235]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。

20

【0191】

[00236]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)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)がマーキング回路294内に設けられる。

30

40

【0192】

[00237]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

50

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 3 3 に示されたように、加算器 3 2 0 (0) 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) と 2 9 2 (1) とをマージするように構成される。同じく、加算器 3 2 0 (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 2 9 2 (2) と 2 9 2 (3) とをマージするように構成される。加算器 3 2 0 (((X + 1) / 2) - 1)、3 2 0 (7) は、マージベクトルデータサンプルセット 3 2 2 (0) ~ 3 2 2 (((X + 1) / 2) - 1)、3 2 2 (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 2 9 2 (X - 1) と 2 9 2 (X) とをマージするように構成される。加算器 3 2 0 (((X + 1) / 2) - 1)、3 2 0 (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 3 2 2 (0) ~ 3 2 2 (((X + 1) / 2) - 1)、3 2 2 (7) は、ラッチ 3 2 5 (0) ~ 3 2 5 (((X + 1) / 2) - 1)、3 2 5 (7) の中にラッチされる。

【 0 1 9 3 】

[00238] マージベクトル処理動作 3 0 2 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 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 2 (0) ~ 3 2 2 (((X + 1) / 2) - 1)、3 2 2 (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 2 9 6 (0) ~ 2 9 6 (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 3 0 2 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 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 3 2 2 (0) ~ 3 2 2 (((X + 1) / 2) - 1)、3 2 2 (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 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 2 (0) ~ 3 2 2 (((X + 1) / 2) - 1)、3 2 2 (7) は、加算器 3 2 4 (0) ~ 3 2 4 (((X + 1) / 4) - 1)、3 2 4 (3) への第 2 の加算器ツリーレベル 3 1 8 (2) に供給される。この関連で、加算器 3 2 4 (0) は、結果として生じるマージベクトルデータサンプル 3 2 6 (0) 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 (0) および 3 2 2 (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 4 (1) は、結果として生じるマージベクトルデータサンプル 3 2 6 (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 2 (2) および 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 (((X + 1) / 4) - 1)、3 2 4 (3) は、結果として生じるマージベクトルデータサンプル 3 2 6 (((X + 1) / 4) - 1)、3 2 6 (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 4 】

[00239] 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

10

20

30

40

50

ータサンプルセット 326(0) ~ 326((X+1)/4) - 1)、326(3)は、加算器 328(0) ~ 328((X+1)/8) - 1)、328(1)への第3の加算器ツリーレベル 318(3)に供給される。この関連で、加算器 328(0)は、それらのサンプルのための16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 326(0)および326(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 328(1)は、16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 330(0) ~ 330((X+1)/8) - 1)、330(1)を供給するために、マージベクトルデータサンプル 326(2)および326(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 328(0) ~ 328((X+1)/8) - 1)、328(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 330(0) ~ 330((X+1)/8) - 1)、330(1)は、ラッチ 329(0) ~ 329((X+1)/8) - 1)、329(1)の中にラッチされる。

10

【0195】

[00240]引き続き図 33を参照すると、マージベクトル処理動作 302が16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 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 330(0) ~ 330((X+1)/8) - 1)、330(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 296(0) ~ 296(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 236が16よりも高いマージ係数(たとえば、32、64、128、256)を要求する場合、マージベクトルデータサンプルセット 330(0) ~ 330((X+1)/8) - 1)、330(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 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 330(0) ~ 330((X+1)/8) - 1)、330(1)は、加算器 332への第4の加算器ツリーレベル 318(4)に供給される。この関連で、加算器 332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を供給するために、マージベクトルデータサンプル 330(0)および330(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 332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 334は、ラッチ 336および338の中にラッチされる。

20

30

【0196】

[00241]引き続き図 33を参照すると、マージベクトル処理動作 302が32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 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 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

40

50

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。

【0197】

[00242]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 のマージング回路 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、マージング回路 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 のマージング回路 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。

【0198】

[00243]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。

【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 に対してデータプ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。

【0200】

[00245] 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 348(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 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(4) および 354(12) は、ラッチ 346(4) および 346(12) に加算器 320(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 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(5) および 354(13) は、ラッチ 346(5) および 346(13) に加算器 320(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 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(6) および 354(14) は、ラッチ 346(6) または 346(14) に加算器 320(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 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(7) および 354(15) は、ラッチ 346(7) または 346(15) に加算器 320(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 296 を記憶するように制御され得る。

【0201】

[00246] 図 33 のマージング回路 294 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 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 282 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292(0)を、結果として生じる出力ベクトルデータサンプル292(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318(1)~318(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292(9)との、結果として生じる出力ベクトルデータサンプル292(0)のマージを、加算器ツリーレベル318(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318(4)内の加算器332は、次いで、マージされた出力ベクトルデータサンプル296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292(0)を、結果として生じる出力ベクトルデータサンプル292(9)とマージする可能性がある。

【0202】

[00247]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)内に設けられる可能性がある。たとえば、図33のマージング回路294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33の加算器ツリー318の加算器ツリーレベル318(1)~318(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292(0)~292(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292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292(0)~292(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292が、図30における2つの入力ベクトルデータサンプル290(0)、290(1)であった場合、マージング回路294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294は、ベクトルデータサンプル290(1)を選択するように構成される可能性がある。

【0203】

[00248]この関連で、図34を参照すると、図33の第1の加算器ツリーレベル318(1)内の加算器320(0)~320((X+1)/2-1)、320(7)は、図34に示されたように、最大または最小のマージ選択加算器320'(0)~320'((X+1)/2-1)、320'(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318(2)内の加算器324(0)~324((X+1)/4-1)、324(3)は、図34に示されたように、最大または最小の選択器324'(0)~324'((X+1)/4-1)、324'(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318(3)内の加算器328(0)~328((X+1)/8-1)、328(1)は、図34に示されたように、最大または最小の選択器328'(0)~328'((X+1)/8-1)、328'(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318(4)内の加算器332は、図34に示されたように、最大または最小の選択器332'と交換される可能性がある。第5の加算器ツリーレベル318(5)内の加算器340は、図34に示されたように、最大または最小の選択器340'と交換される可能性がある。図34のマージング回路294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292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292(0)を、結果として生じる出力ベクトルデータサンプル292(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318(1)~318(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292(9)との、結果として生じる出力ベクトルデータサンプル292(0)のマージを、加算器ツリーレベル318(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318(4)内の加

10

20

30

40

50

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 4 】

[00249] 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バス構成も含む。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バス構成に関するより例示的な詳細が次に説明される。

10

【 0 2 0 5 】

[00250] 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バ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バス構成により、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。

20

【 0 2 0 6 】

[00251] 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。

30

【 0 2 0 7 】

[00252] この関連で、図 3 5 をさらに参照すると、V P E 2 2 の M 0 乗算ベクトルパイプラインステージ 4 6 0 (1) が最初に記載される。M 0 乗算ベクトルパイプラインステージ 4 6 0 (1) は、各々がプログラム可能なデータバス構成を有する、任意の所望の数の乗算器ブロック 4 6 2 (A) ~ 4 6 2 (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 4 6 2 (A) ~ 4 6 2 (0) は、実行ユニット 8 4 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 4 6 2 (A) ~ 4 6 2 (0) は、最大 1 2 個の乗算ベクトルデータサンプルセット 3 4 (Y) ~ 3 4 (0) の乗算を提供するために、M 0 乗算ベクトルパイプラインステージ 4 6 0 (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、M 0 乗算ベクトルパイプラインステージ 4 6 0 (1) に 4 つの乗算器ブロック 4 6 2 (3) ~ 4 6 2 (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 3 4 (Y) ~ 3 4 (0) は、実行ユニット 8 4 内の第 1 のベクトルパイプラインステージ 4 6 0 (0) である、入力読取り (R R) 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 4 6 4 (Y) ~ 4 6 4 (0) へのベクトル処理のための

40

50

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。

10

【0208】

[00253]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。

20

【0209】

[00254]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。

30

【0210】

[00255]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ダ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

40

50

回路 4 8 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 8 4 内の M 0 乗算ベクトルパイプラインステージ 4 6 0 (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 4 6 6 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 4 6 2 (3) ~ 4 6 2 (0) は、命令ディスパッチ回路 4 8 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。

【 0 2 1 1 】

[00256] 乗算器ブロック 4 6 2 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 3 5 を参照すると、あるベクトル処理ブロックデータバス構成において、乗算器ブロック 4 6 2 は、2 つの 8 ビットベクトルデータ入力サンプルセット 4 6 6 を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バス構成では、乗算器ブロック 4 6 2 は、2 つの 1 6 ビットベクトルデータ入力サンプルセット 4 6 6 を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 4 6 6 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 4 6 6 の第 1 のペアから形成される。これは図 3 8 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 4 6 2 (3) ~ 4 6 2 (0) 内にプログラム可能なデータバス構成を設けることにより、乗算器ブロック 4 6 2 (3) ~ 4 6 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、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

10

20

【 0 2 1 2 】

[00257] 図 3 5 に戻って参照すると、複数の乗算器ブロック 4 6 2 (3) ~ 4 6 2 (0) は、プログラム可能な出力データバス 4 7 0 (3) ~ 4 7 0 (0) 内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、次のベクトル処理ステージ 4 6 0 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 4 6 8 (3) ~ 4 6 8 (0) は、複数の乗算器ブロック 4 6 2 (3) ~ 4 6 2 (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バス 4 7 0 (3) ~ 4 7 0 (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バス 4 7 0 (3) ~ 4 7 0 (0) 内のベクトル乗算出力サンプルセット 4 6 8 (3) ~ 4 6 8 (0) は、下記で説明されるように、累算のために M 1 累算ベクトルパイプラインステージ 4 6 0 (2) に供給される。実行ユニット 8 4 のこの特定の設計では、複数の乗算器ブロック 4 6 2 (3) ~ 4 6 2 (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 8 4 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バス構成を有する V P E 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。

30

40

【 0 2 1 3 】

[00258] 引き続き図 3 5 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 4 6 8 (3) ~ 4 6 8 (0) は、M 1 累算ベクトル処理ステージ 4 6 0 (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 4 7 2 (3) ~ 4 7 2 (0) に供給される。複数の累算器ブロック 4 7 2 (A) ~ 4 7 2 (0) の中の各累算器ブロックは、2 つの累算器 4 7 2 (X) (1) および 4 7 2 (X) (0) (すなわち、4 7 2 (3) (1)、4 7 2 (3) (0)、4 7 2 (2) (1)、4 7 2 (2) (0)、4

50

72(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。

【0214】

[00259]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))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。

【0215】

[00260]図35に戻って参照すると、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パス478(3)~78(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)は、乗算器ブロック462(3)~462(0)から累算器ブロック472(3)~472(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468(3)~468(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472は、2つの累算器472(X)(1)、472(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478(A)~478(0)は、478(3)(1)、478(3)(0)、478(2)(1)、478(2)(0)、478(1)(1)、478(1)(0)、および478(0)(1)、478(0)(0)として図35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480(3)~480(0)は、480(3)(1)、480(3)(0)、480(2)(1)、480(2)(0)、480(1)(1)、480(1)(0)、480(0)(1)、480(0)(0)として図35に示されている。累算器ブロック472(3)~472(0)内に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)を設けることは、図39および図40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472(3)~472(0)のプログラム

10

20

30

40

50

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) に従って、累算器ブロック 472(3) ~ 472(0) は、累算されたベクトル乗算出力サンプルセット 468(3) ~ 468(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 476(3) ~ 476(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 472(3) ~ 472(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、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ふ柔軟性がもたらされる。

10

【0216】

[00261]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 472 のプログラム可能な入力データパス 478 および / またはプログラム可能な内部データパス 480 は、非限定的な例として、単一の 40 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 472 のプログラム可能な入力データパス 478 および / またはプログラム可能な内部データパス 480 は、非限定的な例として、二重 24 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 472 のプログラム可能な入力データパス 478 および / またはプログラム可能な内部データパス 480 は、16 ビット桁上げ保存加算器、それに

20

続く単一の 24 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 462(3) ~ 462(0) および累算器ブロック 472(3) ~ 472(0) (たとえば、16 ビット累算を用いる 16 ビット虚数乗算、および 16 ビット累算を用いる 32 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 84 によってサポートされ得る。

【0217】

[00262]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) 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ダ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 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、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、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、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) 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。

30

40

【0218】

[00263]図 3 6 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 3 5 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

50

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。

【0219】

[00264]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)と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ット476(A)(1)~476(0)(0)を受信することを含む(ブロック513)。

【0220】

[00265]プログラム可能なデータバ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。

【0221】

[00266]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、特定の入力データバス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、どの入力データバス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バス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バス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。

【0222】

[00267]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、入力データバス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。入力データバス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、BLは入力被乗数入力「B」の下位ビットを表す。入力データバス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バス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。

【0223】

[00268]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バス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、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、桁上げ伝搬バ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。

【0224】

[00269]それらのプログラム可能な内部データバス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C 1 1と和S 1 1とを生成するように構成される。乗算器ブロック4 6 2 (1)は、8ビット乗算のための3 2ビット値として桁上げC 2 0と和S 2 0とを生成し、1 6ビット乗算のための6 4ビット値として桁上げC 2 1と和S 2 1とを生成するように構成される。乗算器ブロック4 6 2 (0)は、8ビット乗算のための3 2ビット値として桁上げC 3 0と和S 3 0とを生成し、1 6ビット乗算のための6 4ビット値として桁上げC 3 1と和S 3 1とを生成するように構成される。

【 0 2 2 5 】

[00270]図 3 7 の乗算器ブロック4 6 2 内に設けられるプログラム可能なデータバス構成のより例示的な詳細を説明するために、図 3 8 が提供される。図 3 8 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4 6 6 と、1 6ビット×1 6ビットベクトルデータ入力サンプルセット4 6 6 とを乗算することが可能な、プログラム可能なデータバス構成を有する、図 3 7 の乗算器ブロック4 6 2 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4 6 2 は、この例では4つの8×8ビット乗算器4 8 4 (3) ~ 4 8 4 (0)を含む。任意の所望の数の乗算器4 8 4 が設けられる可能性がある。第1の乗算器4 8 4 (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4 6 6 A [H]を受信し、ベクトルデータ入力サンプルセット4 6 6 A [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4 6 6 B [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4 6 6 C [I]のいずれかと乗算するように構成される。乗算器4 8 4 (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]のいずれかを選択するように構成された、マルチプレクサ4 8 6 (3)が設けられる。マルチプレクサ4 8 6 (3)は、この実施形態では、被乗数選択器入力4 8 2 内の上位ビットである被乗数選択器入力4 8 2 [3]によって制御される。このようにして、マルチプレクサ4 8 6 (3)および被乗数選択器入力4 8 2 [3]は、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4 6 6 A [H]と乗算されるかを、乗算器4 8 4 (3)が制御するためのプログラム可能な内部データバス4 6 7 [0]構成を提供する。

【 0 2 2 6 】

[00271]引き続き図 3 8 を参照すると、他の乗算器4 8 4 (2) ~ 4 8 4 (0)も、第1の乗算器4 8 4 (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バス4 6 7 [2] ~ 4 6 7 [0]を含む。乗算器4 8 4 (2)は、被乗数入力「A」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 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バス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バス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バス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バス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バス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467[0]を含む。選択は、この実施形態では、乗数選択器入力482内の被乗数選択器ビット入力482[0]に従って、マルチプレクサ486(0)によって制御される。

【0227】

[00272]引き続き図38を参照すると、上記で説明されたように、乗算器484(3)~484(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484(3)~484(0)は、それぞれ、ビット長乗算モード入力488(3)~488(0)を含む。この例では、各乗算器484(3)~484(0)は、それぞれ、プログラム可能なデータパス490(3)~490(0)、491、および492(3)~492(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484(3)~484(0)はまた、それぞれ、プログラム可能なデータパス490(3)~490(0)、491、および492(3)~492(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484(3)~484(0)が、プログラム可能なデータパス490(3)~490(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484(3)~484(0)は、乗算器ブロック462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484(3)~484(0)が、プログラム可能なデータパス491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484(3)~484(0)は、乗算器ブロック462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484(3)~484(0)が、プログラム可能なデータパス492(3)~492(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484(3)~484(0)は、乗算器ブロック462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。

【0228】

[00273]引き続き図38を参照すると、この例における乗算器484(3)~484(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ビットの入力和494(3)、494(2)および入力桁上げ496(3)、496(2)は、それぞれ、各乗算器484(3)、484(2)によって生成される。16ビットの入力和494(1)、494(0)および入力桁上げ496(1)、496(0)は、それぞれ、各乗算器484(1)、484(0)によって生成される。16ビットの入力和494(3)、494(2)および入力桁上げ496(3)、496(2)はまた、一緒に入力和494(3)~494(0)と入力桁上げ496(3)~496(0)とを加算するために、16ビットの和入力494(1)、494(0)および入力桁上げ496(1)、496(0)とともに24ビット4:2圧縮器515に供給される。加算された入力和494(3)~494(0)および入力桁上げ496(3)~496(0)は、プログラム可能なデータパス491がアクティブであり、入力和494(3)~494(0)および入力桁上げ496(3)~496(0)とゲート制御されるとき、16ビット×16ビット乗算モードで単一和498と単一桁上げ500とをもたらす。プログラム可能なデータパス491は、24ビット4:2圧縮器515に供給されるように、16ビットワードとして、組み合わせされた入力和494(3)、494(2)を有する第1のANDベースのゲート502(3)によって、および16ビットワードとして、組み合わせされた入力桁上げ496(3)、496(2)を有する第2のANDベースのゲート502(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491はまた、24ビット4:2圧縮器515に供給されるように、16ビットワードとして、組み合わせされた入力和494(1)、494(0)を有する第3のANDベースのゲート502(1)によって、および16ビットワードとして、組み合わせされた入力桁上げ496(1)、496(0)を有する第4のANDベースのゲート502(0)によってゲート制御される。乗算器

ブロック462が16ビット×16ビット乗算モードまたは24ビット×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470[0]は、圧縮された32ビット和50および32ビット桁上げC0部分積として、ベクトル乗算出力サンプルセット468[0]を供給される。

【0229】

[00274]乗算器ブロック462内の乗算器484(3)~484(0)が8ビット×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470[1]構成は、圧縮なしで、16ビット入力494(3)~494(0)および部分積としての対応する16ビット入力桁上げ496(3)~496(0)として提供される。乗算器ブロック462内の乗算器484(3)~484(0)が8ビット×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470[1]は、圧縮なしで、16ビット入力494(3)~494(0)およびベクトル乗算出力サンプルセット468[1]としての対応する16ビット入力桁上げ496(3)~496(0)として提供される。乗算器ブロック462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468[0]、468[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472(3)~472(0)に供給される。

10

【0230】

[00275]プログラム可能なデータパス構成を有する、図37および図38の乗算器ブロック462(3)~462(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472(3)~472(0)と融合されるべき、実行ユニット84内の乗算器ブロック462(3)~462(0)の特徴が、図39に関して次に概説される。

20

【0231】

[00276]この関連で、図39は、上述された実行ユニット84(0)~84(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38に示されたように、乗算器ブロック462は、被乗数入力466[H]と466[L]とを乗算し、少なくとも1つの入力494と少なくとも1つの入力桁上げ496とを、ベクトル乗算出力サンプルセット468として、プログラム可能な出力データパス470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472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470内のベクトル乗算出力サンプルセット468内の少なくとも1つの入力494および少なくとも1つの入力桁上げ496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。

30

40

【0232】

[00277]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。

50

【 0 2 3 3 】

[00278]引き続き図 3 9 を参照すると、圧縮器 5 0 8 は、冗長な形式での入力 4 9 4 および入力桁上げ 4 9 6 を、それぞれ、前の累算ベクトル出力和 5 1 2 および前のシフトされた累算ベクトル出力桁上げ 5 1 7 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 5 1 7 は、次の受信された入力 4 9 4 および入力桁上げ 4 9 6 の次の累算が圧縮器 5 0 8 によって実行される前に、累算ベクトル出力桁上げ 5 1 4 をシフトすることにより、圧縮器 5 0 8 によって生成された累算ベクトル出力桁上げ 5 1 4 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 5 1 7 は、最終的なシフトされた累算ベクトル出力桁上げ 5 1 7 内で桁上げ累算を伝搬して、最終累算ベクトル出力和 5 1 2 を最終累算器出力サンプルセット 4 7 6 の補数表現に変換するために、累算器ブロック 4 7 2 内に設けられた単一の最終桁上げ伝搬加算器 5 1 9 によって最終累算ベクトル出力和 5 1 2 に加算される。最終累算ベクトル出力和 5 1 2 は、プログラム可能な出力データパス 4 7 4 内で累算器出力サンプルセット 4 7 6 として供給される（図 3 5 参照）。

10

【 0 2 3 4 】

[00279]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 との乗算器ブロック 4 6 2 の融合を示す図 3 9 が記載されたので、累算器ブロック 4 7 2 (3) ~ 4 7 2 (0) に関するより例示的な詳細が、図 4 0 に関してここで概説される。図 4 0 は、図 3 5 の実行ユニット 8 4 内に設けられた累算器ブロック 4 7 2 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 4 7 2 は、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) を用いて構成され、その結果、累算器ブロック 4 7 2 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 4 7 2 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 4 7 2 内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) の具体例が開示される。また、累算器ブロック 4 7 2 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 4 7 2 [0]、4 7 2 [1] を含むように構成される。

20

30

【 0 2 3 5 】

[00280]累算器ブロック 4 7 2 の例示的な内部構成要素が図 4 0 に示される。そこに示されているように、この実施形態における累算器ブロック 4 7 2 は、一緒に累算されるために、第 1 の入力 4 9 4 [0] および第 1 の入力桁上げ 4 9 6 [0] と、第 2 の入力 4 9 4 [1] および第 2 の入力桁上げ 4 9 6 [1] とを乗算器ブロック 4 6 2 から受信するように構成される。図 4 0 に関して、入力 4 9 4 [0]、4 9 4 [1] および入力桁上げ 4 9 6 [0]、4 9 6 [1] は、ベクトル入力 4 9 4 [0]、4 9 4 [1] およびベクトル入力桁上げ 4 9 6 [0]、4 9 6 [1] と呼ばれる。前述され、図 3 9 に示されたように、この実施形態におけるベクトル入力 4 9 4 [0]、4 9 4 [1] およびベクトル入力桁上げ 4 9 6 [0]、4 9 6 [1] は、各々長さが 1 6 ビットである。この例における累算器ブロック 4 7 2 は、2 つの 2 4 ビット桁上げ保存累算器ブロック 4 7 2 [0]、4 7 2 [1] として設けられ、「 [0] 」が桁上げ保存累算器 4 7 2 [0] 用に指定され、「 [1] 」が桁上げ保存累算器 4 7 2 [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 4 7 2 [0]、4 7 2 [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。

40

【 0 2 3 6 】

[00281]図 4 0 の桁上げ保存累算器 4 7 2 [0] を参照すると、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] は、プログラム可能な内部データパス 4 8 0 [0] の一部として設けられたマルチプレクサ 5 0 4 (0) 内の入力である。負のベク

50

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。
【0237】

[00282]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 $476(3) \sim 476(0)$ 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、累算ベクトル出力和 $512(0)$ 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 $508(0)$ に融合されるバレルシフトである。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、累算ベクトル出力和 $512(0)$ 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 $472[0]$ 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。
【0238】

[00283]さらなる後続のベクトル入力 $494[0]$ 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 $496[0]$ は、現在の累算ベクトル出力和 $512(0)$ および現在の累算ベクトル出力桁上げ $517(0)$ と累算され得る。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 $496[0]$ 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 $520(0)$ に従うプログラム可能な内部データパス $480[0]$ の一部として、マルチプレクサ $518(0)$ によって選択される。現在の累算ベクトル出力和 $512(0)$ および現在のシフトされた累算ベクトル出力桁上げ $517(0)$ は、桁上げ保存累算器ブロック $472[0]$ が、更新された累算ベクトル出力和 $512(0)$ と累算ベクトル出力桁上げ $514(0)$ とを供給するために、圧縮器 $508(0)$ 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 $520(0)$ により、累算器ブロック $472[0]$ のプログラム可能な内部データパス $480[0]$ が、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、圧縮器 $508(0)$ にベクトル入力 $494[0]$ とベクトル

10

20

30

40

50

入力桁上げ496[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472[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526(0)に従って累算ベクトル出力和512(0)およびシフトされた累算ベクトル出力桁上げ517(0)の現在の状態を、マルチプレクサ518(0)に保持させるために、この実施形態では保持ゲート522(0)、524(0)も提供される。

【0239】

[00284]引き続き図40を参照すると、桁上げ保存累算器ブロック472[0]の累算ベクトル出力和512(0)およびシフトされた累算ベクトル出力桁上げ517(0)、ならびに桁上げ保存累算器ブロック472[1]の累算ベクトル出力和512(1)およびシフトされた累算ベクトル出力桁上げ517(1)は、それぞれ、制御ゲート534(0)、536(0)および534(1)、536(1)によってゲート制御される。制御ゲート534(0)、536(0)および534(1)、536(1)は、それぞれ、圧縮器508(0)、508(1)に戻される、累算ベクトル出力和512(0)およびシフトされた累算ベクトル出力桁上げ517(0)と、累算ベクトル出力和512(1)およびシフトされた累算ベクトル出力桁上げ517(1)とを制御する。

10

【0240】

[00285]要約すると、図40の累算器ブロック472の累算器ブロック472[0]、472[1]のプログラム可能な入力データパス478[0]、478[1]およびプログラム可能な内部データパス480[0]、480[1]により、累算器ブロック472は様々なモードで構成され得る。累算器ブロック472は、図40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。

20

【0241】

[00286]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VPE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(PDA)、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(DVD)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。

30

【0242】

[00287]この関連で、図41は、プロセッサベースのシステム550の例を示す。この例では、プロセッサベースのシステム550は、各々が1つまたは複数のプロセッサまたはコア554を含む、1つまたは複数の処理ユニット(PU)552を含む。PU552は、非限定的な例として、図2のベースバンドプロセッサ20であり得る。プロセッサ554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20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554は、図2の実行ユニット84を含むが、それに限定されないVPE556を含む場合もある。PU552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554に結合されたキャッシュメモリ558を有する場合がある。PU552は、システムバス560に結合され、プロセッサベースのシステム550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、PU552は、システムバス560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、PU552は、スレーブデバイスの例として、メモリコントローラ562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41には示されていないが、複数のシステムバス560が提供される可能性があり、ここで、各システムバス560は様々なファブリックを構成する。

40

50

【 0 2 4 3 】

[00288]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560に接続され得る。図41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564、1つまたは複数の入力デバイス566、1つまたは複数の出力デバイス568、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570、および1つまたは複数のディスプレイコントローラ572が含まれ得る。メモリシステム564は、メモリコントローラ562によってアクセス可能なメモリ565を含むことができる。入力デバイス566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568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570は、ネットワーク574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574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(LAN)、ワイドローカルエリアネットワーク(WLAN)、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570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。

10

【 0 2 4 4 】

[00289]PU552はまた、1つまたは複数のディスプレイ578に送られる情報を制御するために、システムバス560を介してディスプレイコントローラ572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572は、1つまたは複数のビデオプロセッサ580を介して表示されるべき情報をディスプレイ578に送り、ビデオプロセッサ580は、表示されるべき情報をディスプレイ578に適したフォーマットに処理する。ディスプレイ578は、限定はしないが、陰極線管(CRT)、液晶ディスプレイ(LCD)、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。

20

【 0 2 4 5 】

[00290]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。

30

40

【 0 2 4 6 】

[00291]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

50

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。

【0247】

[00292]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。

10

【0248】

[00293]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。

20

【0249】

[00294]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。

30

【 図 1 】

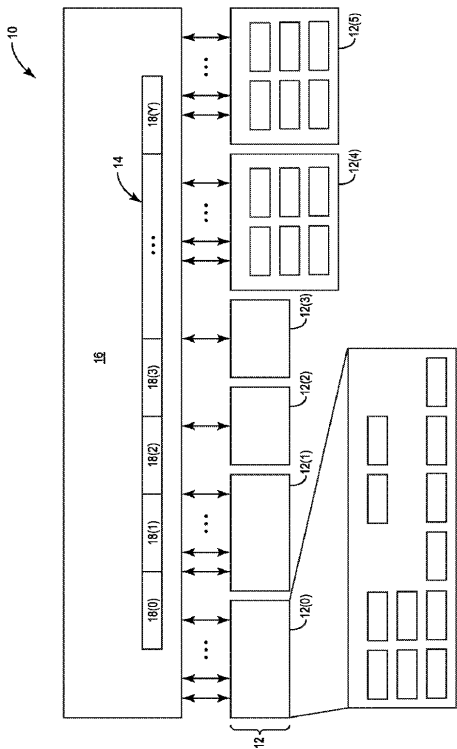


FIG. 1

【 図 2 】

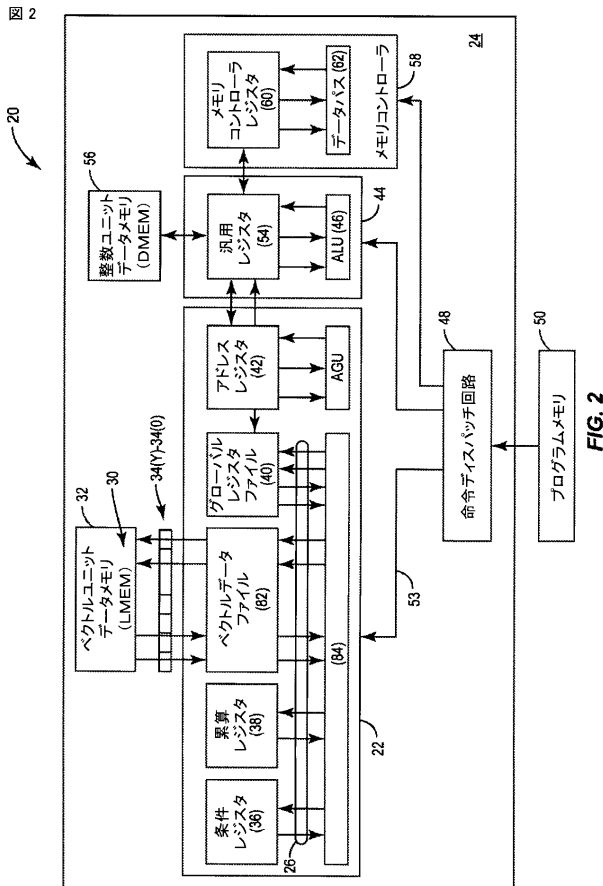


FIG. 2

【 図 3 】

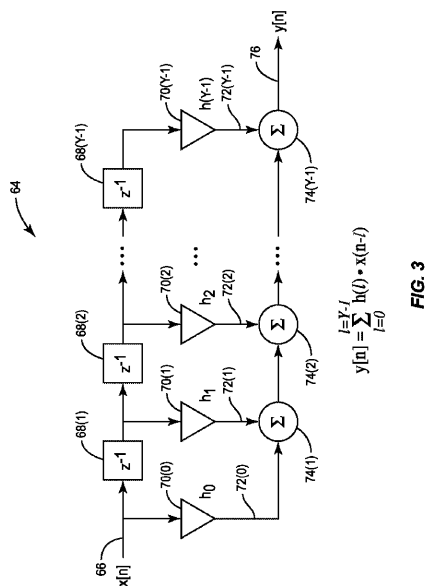


FIG. 3

【 図 4 】

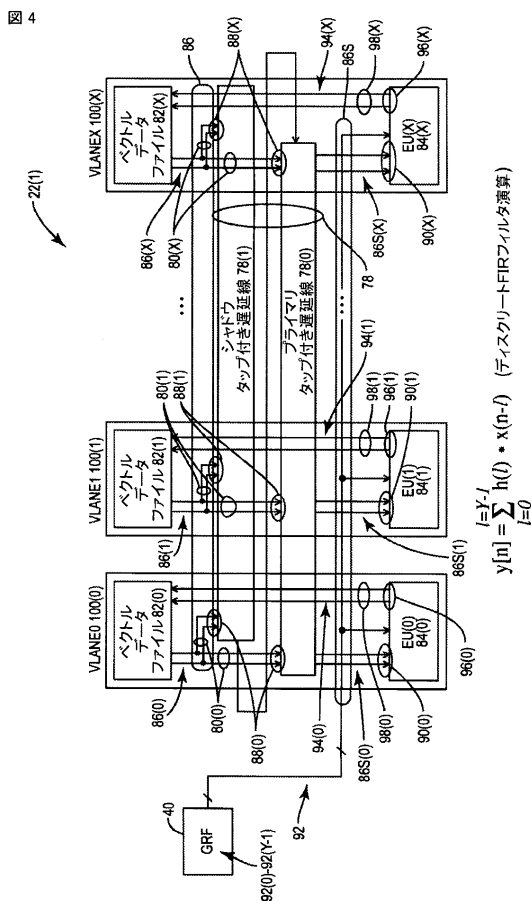


FIG. 4

【 図 5 】

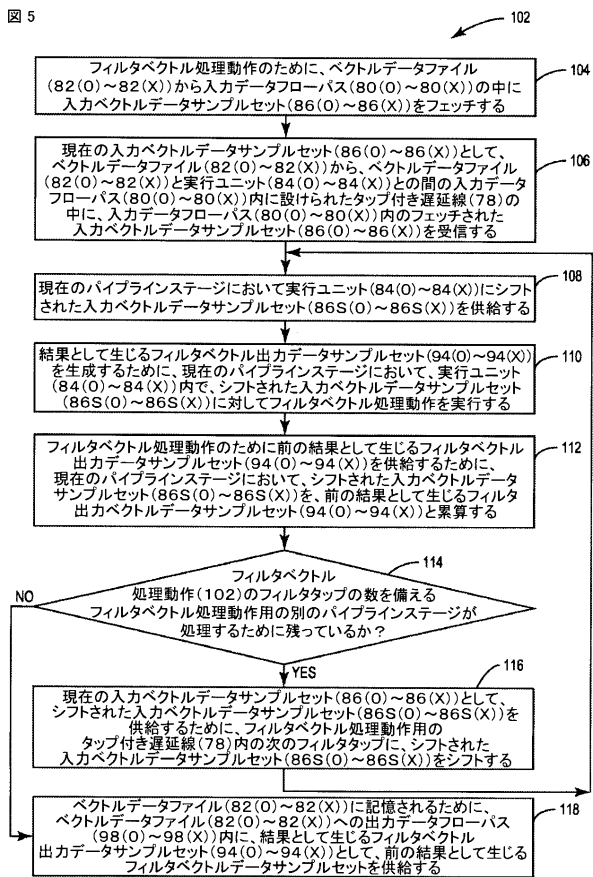


FIG. 5

【 図 6 B 】

図 6B

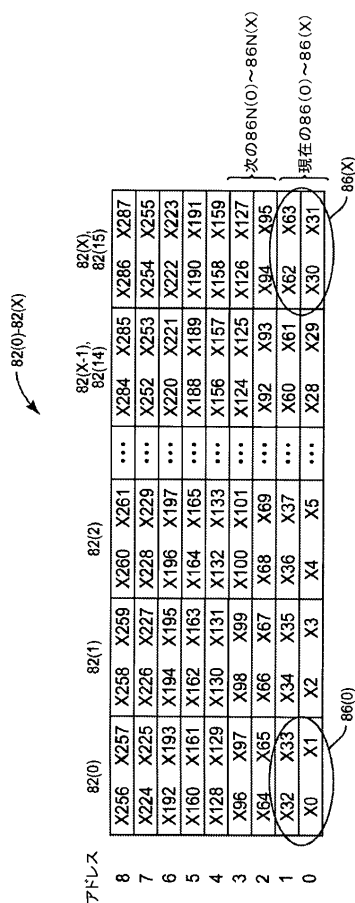


FIG. 6B

【 図 6 A 】

図 6A

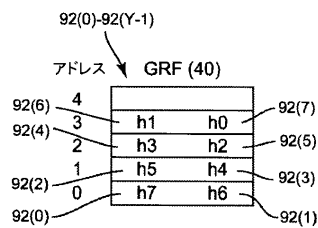


FIG. 6A

【 図 7 】

図 7

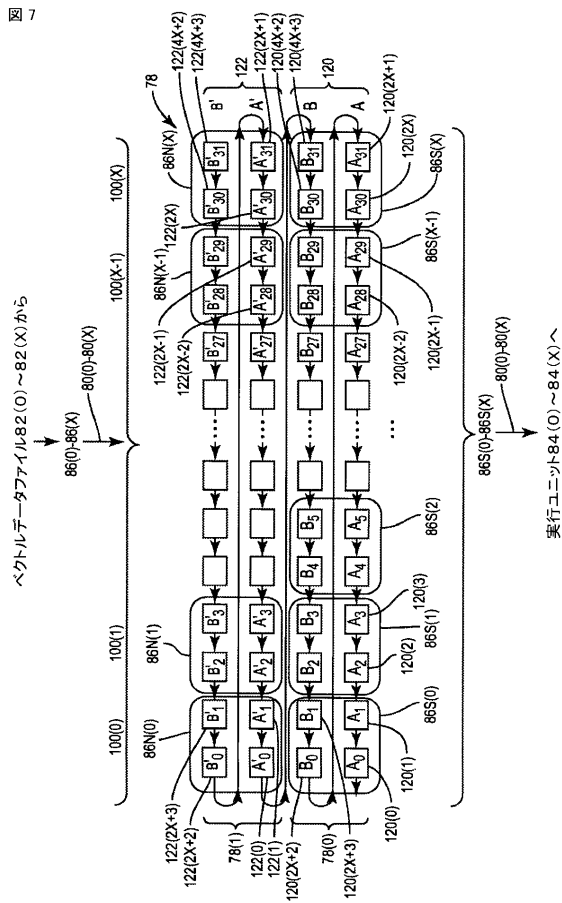


FIG. 7

【 図 8 】

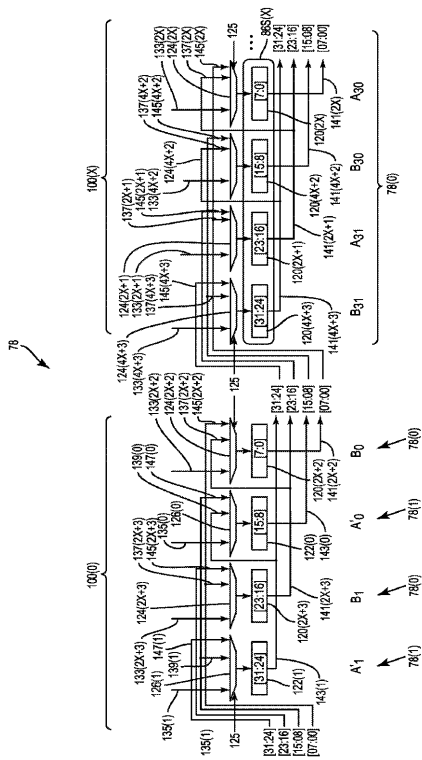


FIG. 8

【 図 9 A 】

図 9A

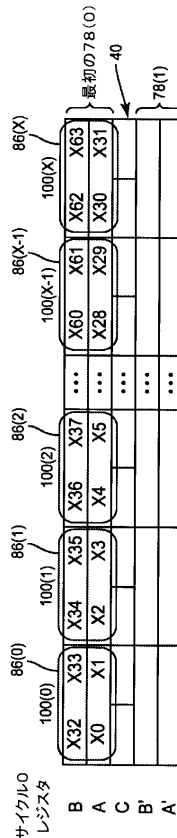


FIG. 9A

【 図 9 B 】

図 9B

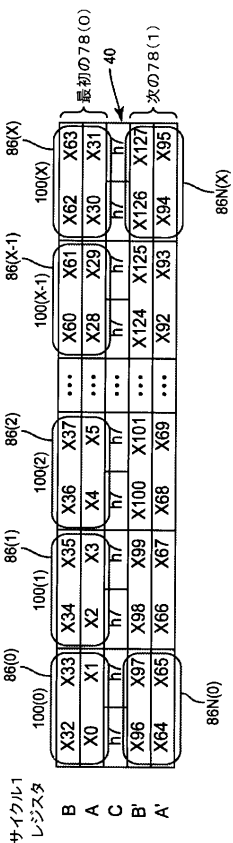


FIG. 9B

【 図 9 C 】

図 9C

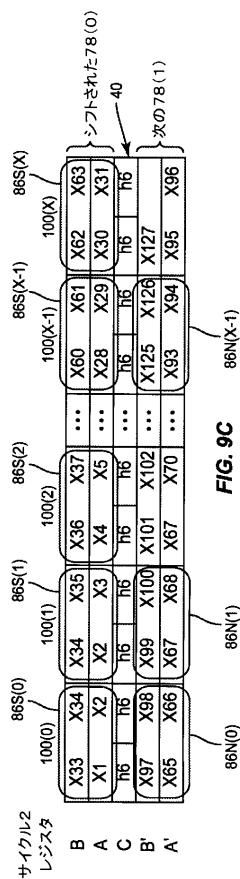


FIG. 9C

【 図 9 D 】

図 9D

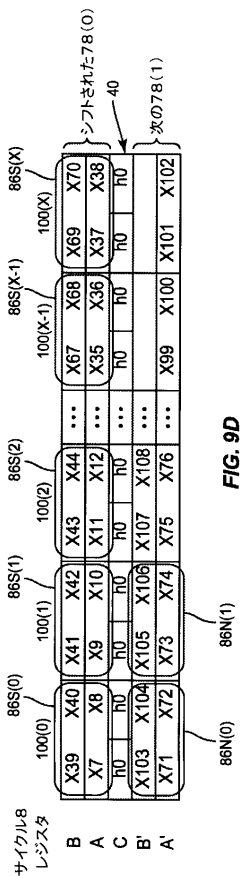


FIG. 9D

【 図 1 0 】

図 10

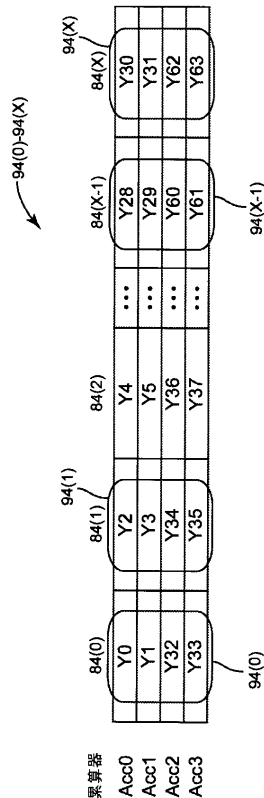


FIG. 10

【 図 1 1 】

図 11

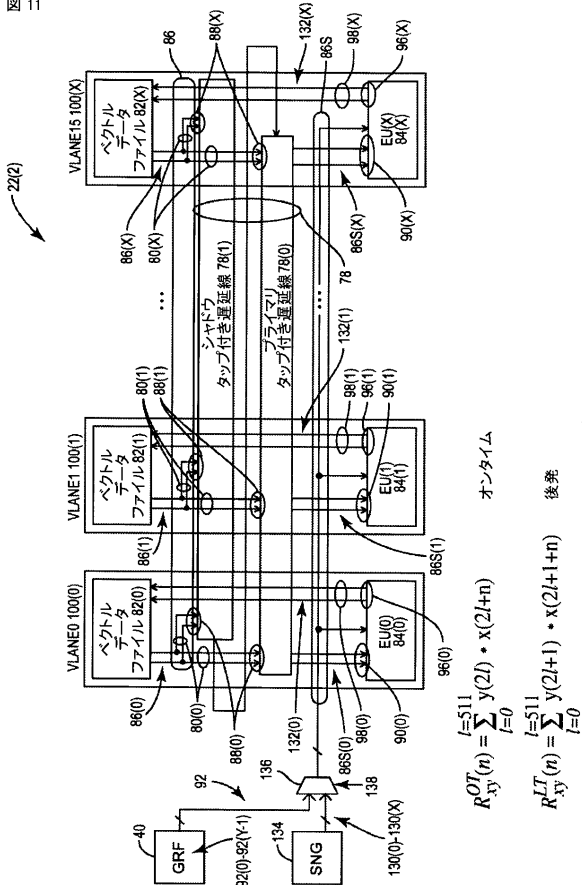


FIG. 11

オンタイム 後乗

$$R_{xy}^{OT}(n) = \sum_{l=0}^{l=511} y(2l) \cdot x(2l+n)$$

$$R_{xy}^{LT}(n) = \sum_{l=0}^{l=511} y(2l+1) \cdot x(2l+1+n)$$

【 図 1 2 A 】

図 12A

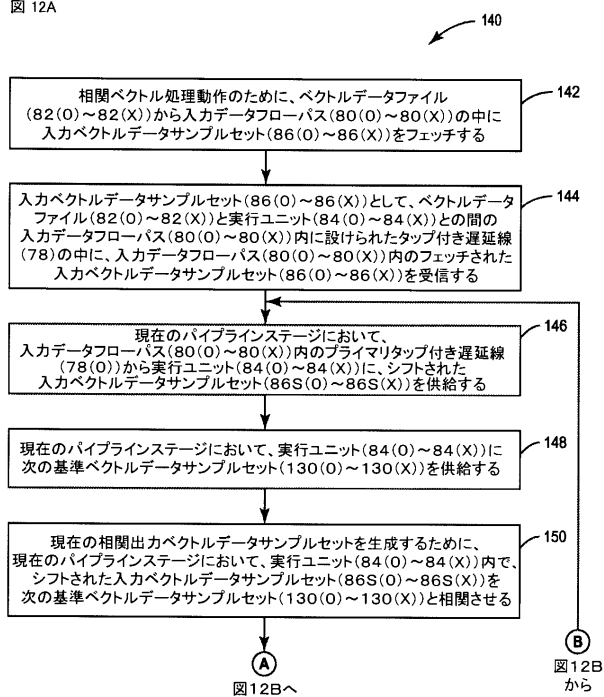


FIG. 12A

図 12B から

【 図 1 2 B 】

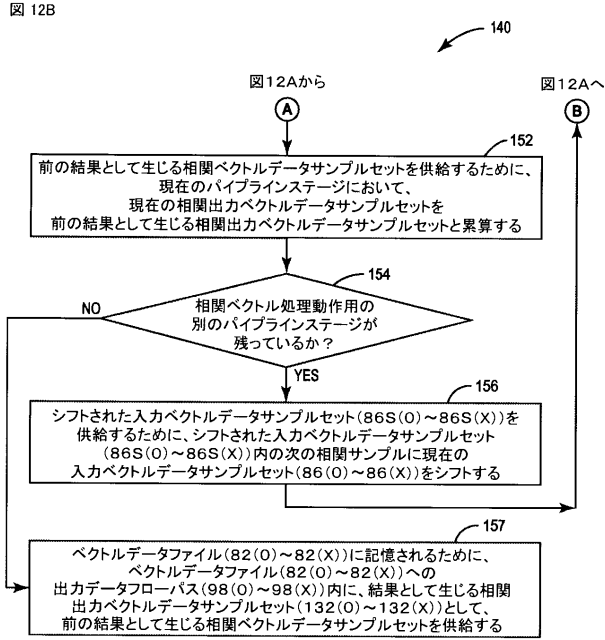


FIG. 12B

【 図 1 3 】

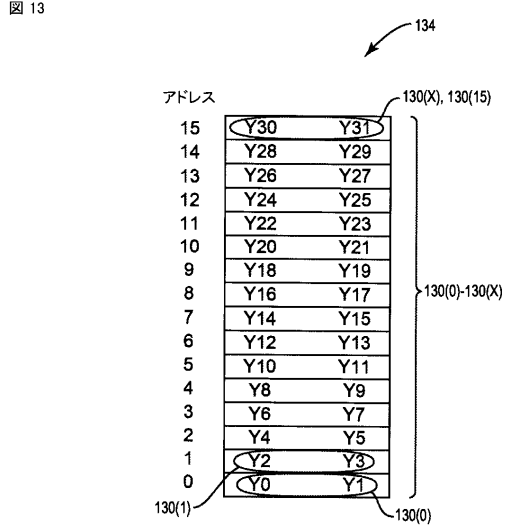


FIG. 13

【 図 1 4 】

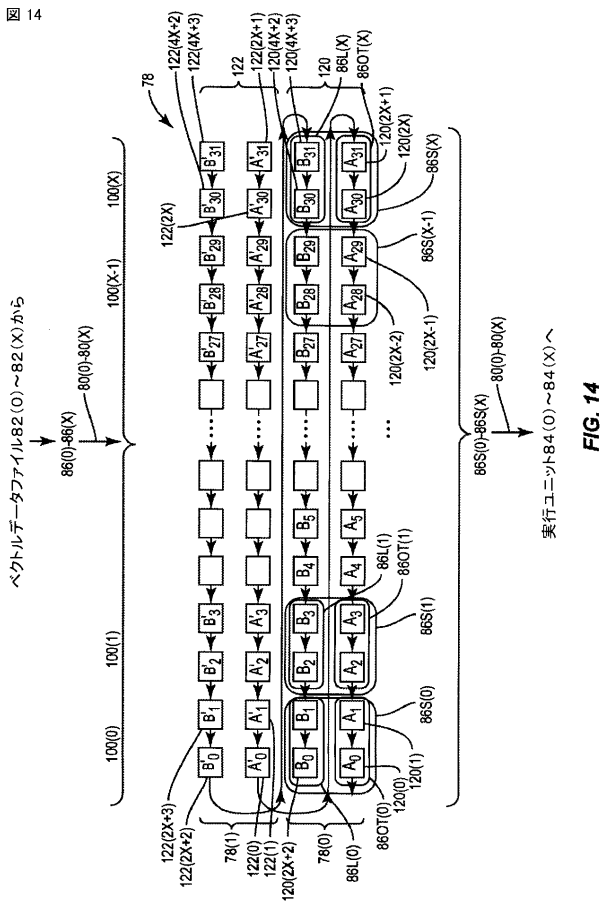


FIG. 14

【 図 1 5 A 】

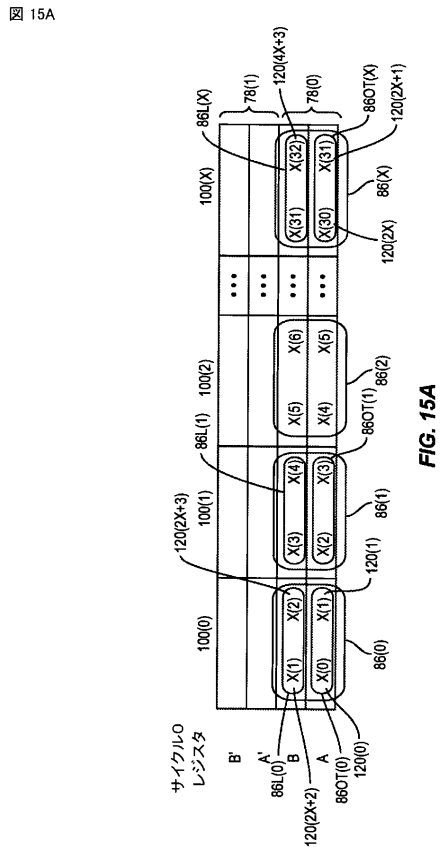


FIG. 15A

【 図 15 B 】

図 15B

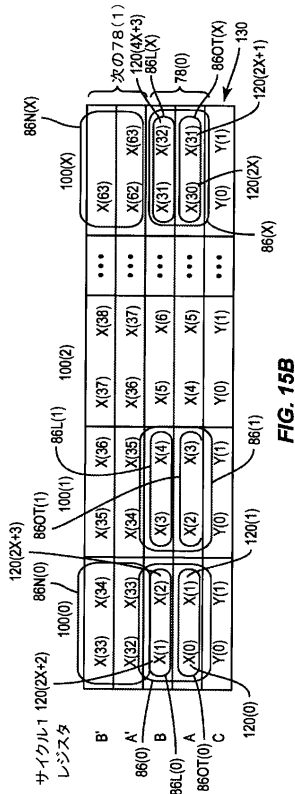


FIG. 15B

【 図 15 C 】

図 15C



FIG. 15C

【 図 15 D 】

図 15D



FIG. 15D

【 図 16 】

図 16

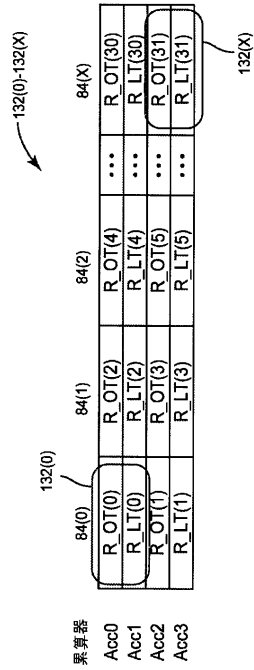


FIG. 16

【 17 A 】

17A

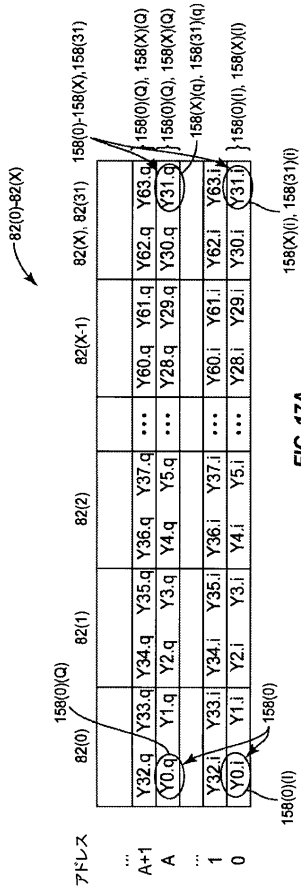


FIG. 17A

【 17 B 】

17B

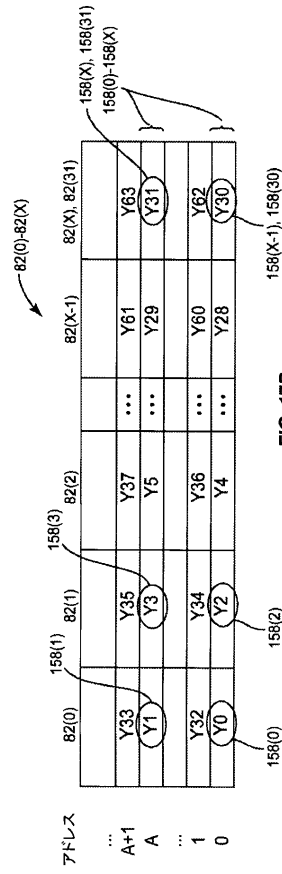


FIG. 17B

【 18 A 】



FIG. 18A

【 18 B 】



FIG. 18B

【図 19】

図 19

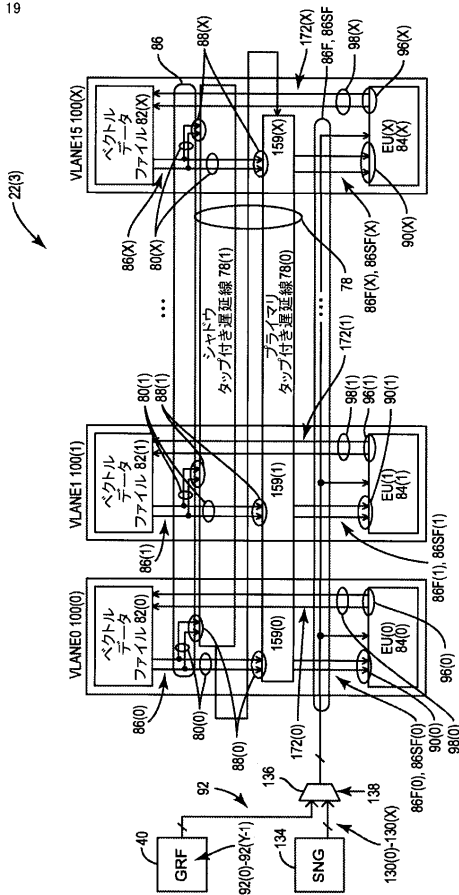


FIG. 19

【図 20】

図 20

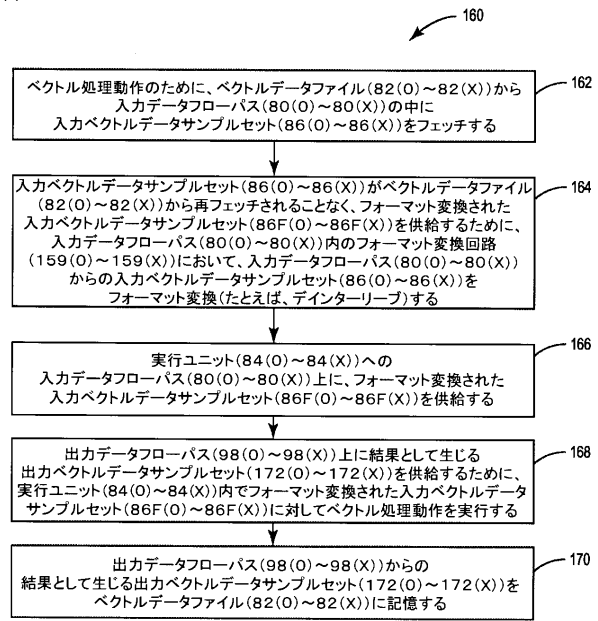


FIG. 20

【図 21】

図 21

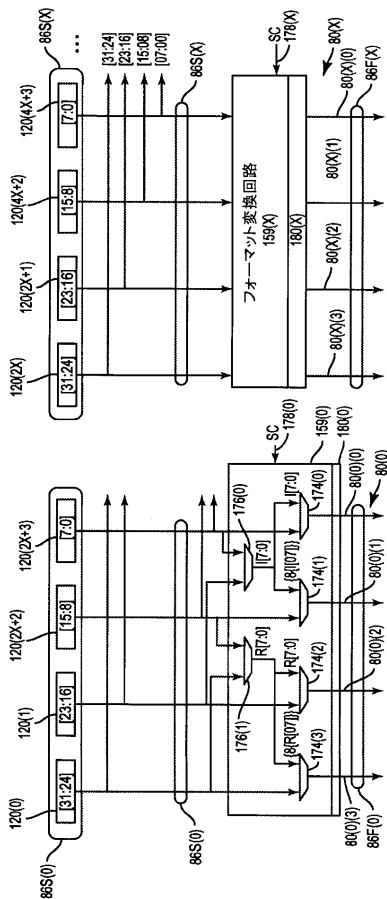


FIG. 21

【図 22】

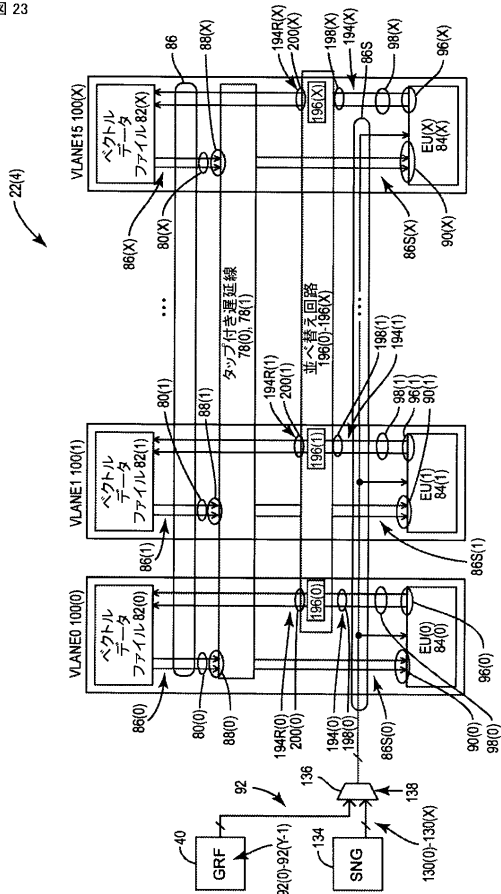
図 22

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
予備	[15:8]	予備フィールド
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フォーマットで記述される 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
予備	[31:20]	予備フィールド

FIG. 22

【図 2 3】

図 23



【図 2 5】

図 25

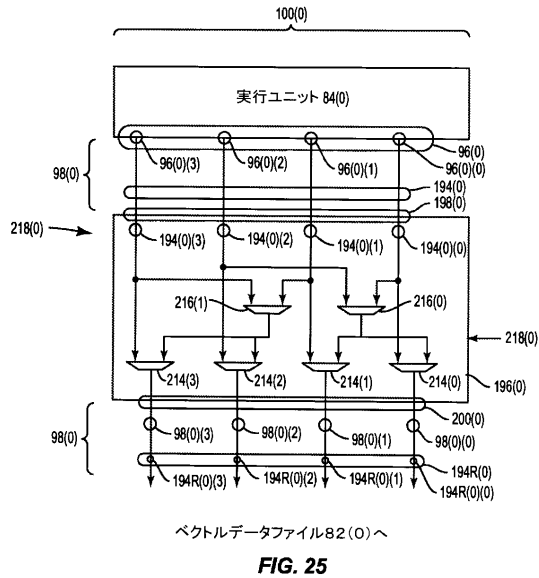


FIG. 25

【図 2 4】

図 24

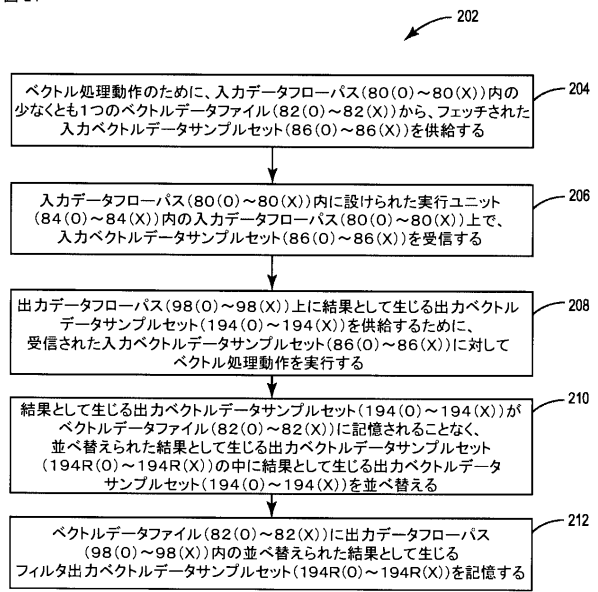
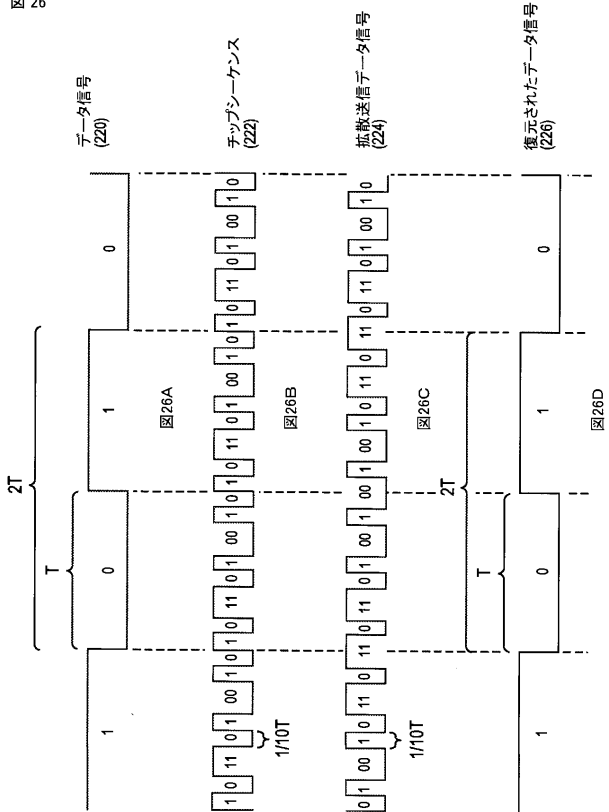


FIG. 24

【図 2 6】

図 26



【 図 27 】

図 27

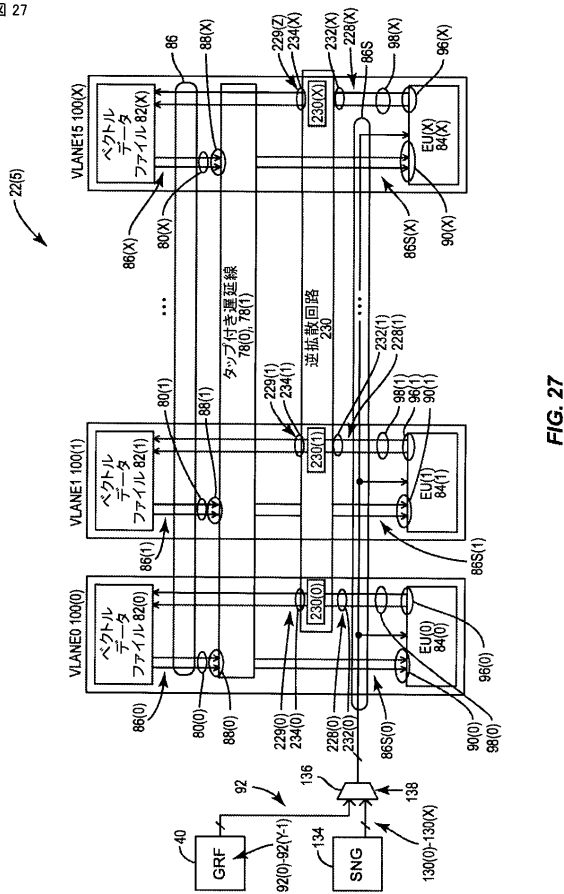


FIG. 27

【 図 28 】

図 28

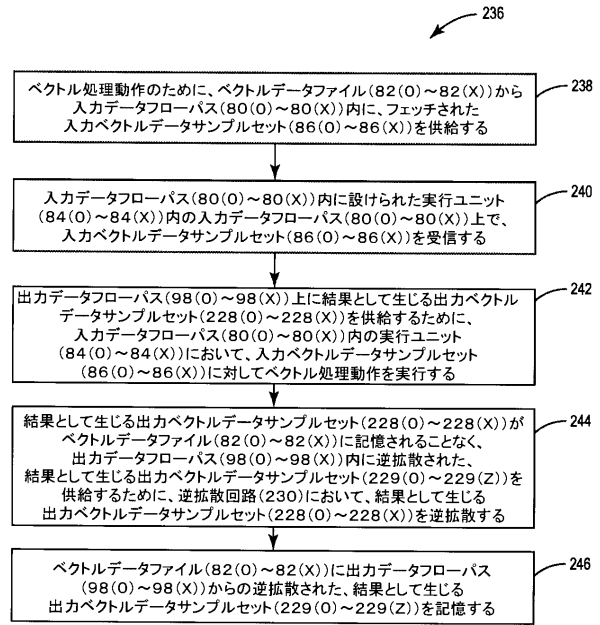


FIG. 28

【 図 29 】

図 29

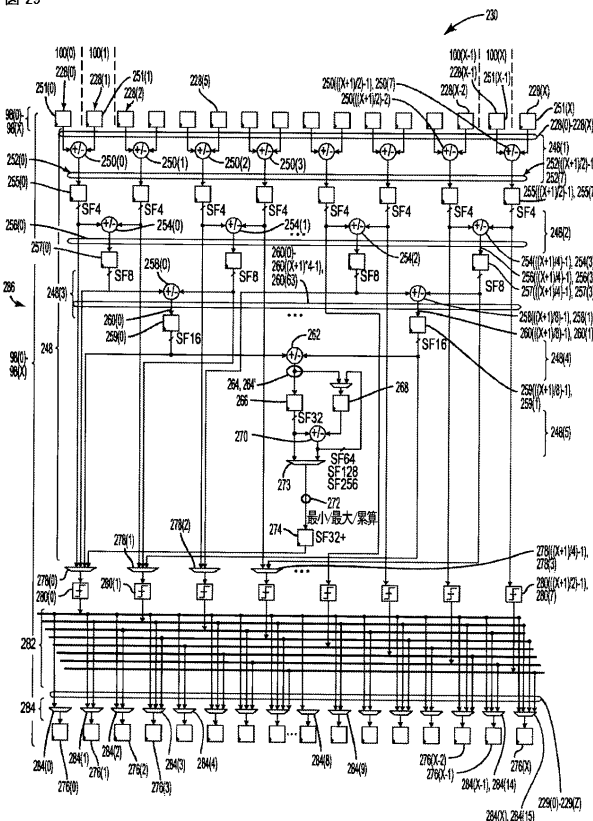


FIG. 29

【 図 30 】

図 30

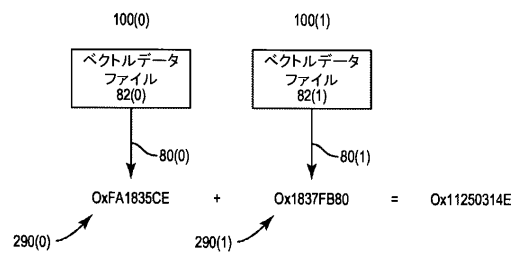


FIG. 30

【 図 3 1 】

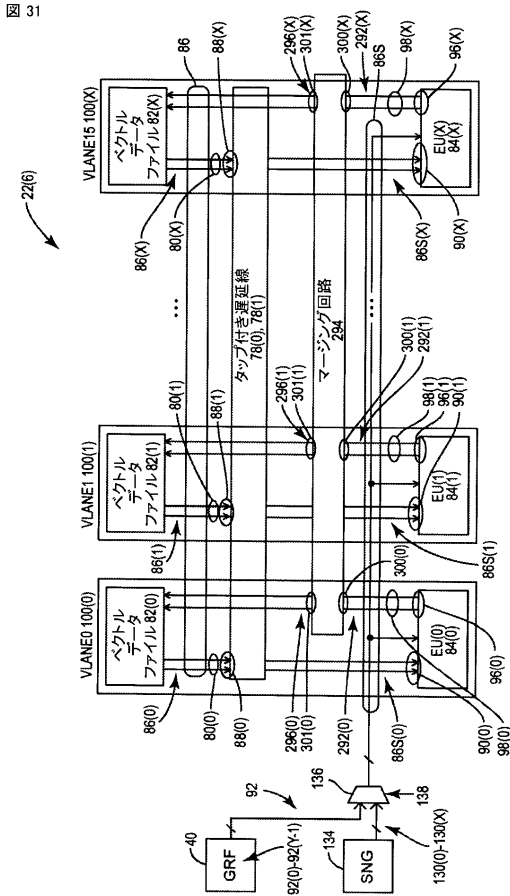


FIG. 31

【 図 3 2 】

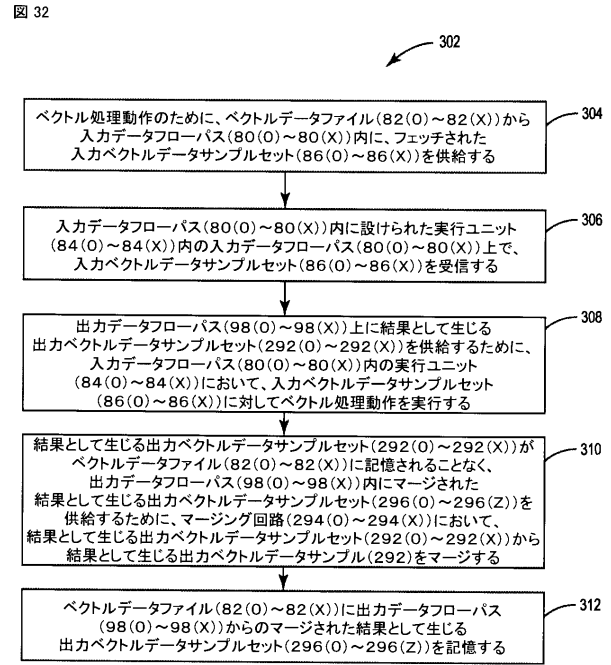


FIG. 32

【 図 3 3 】

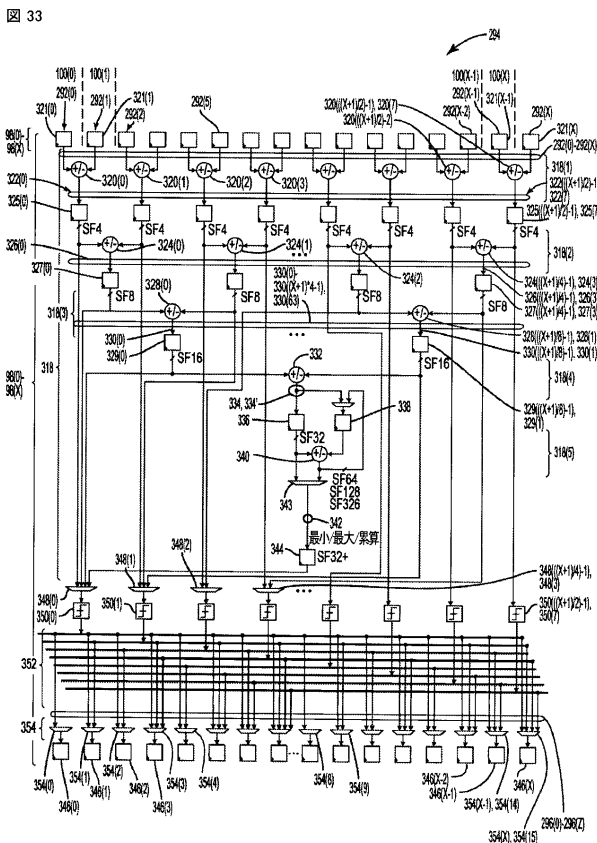


FIG. 33

【 図 3 4 】

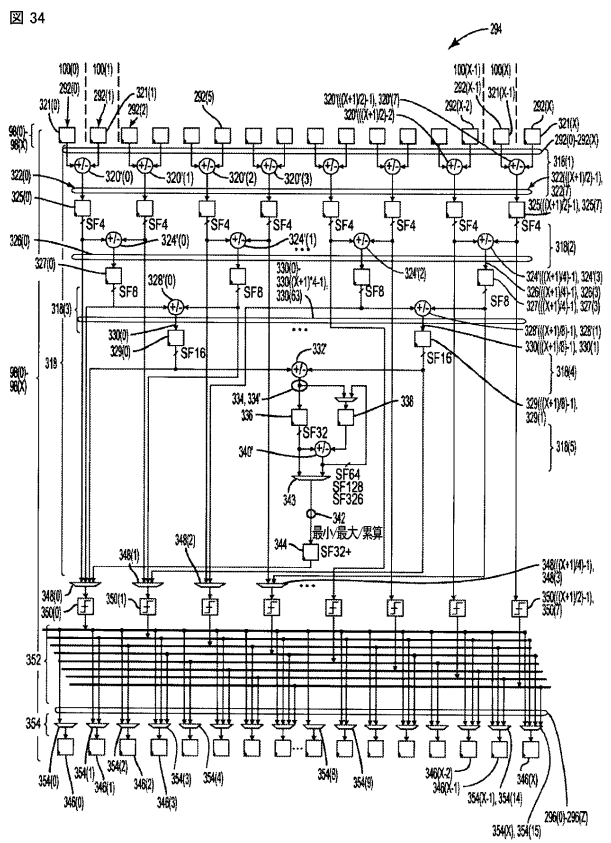


FIG. 34

【 図 3 5 】

図 35

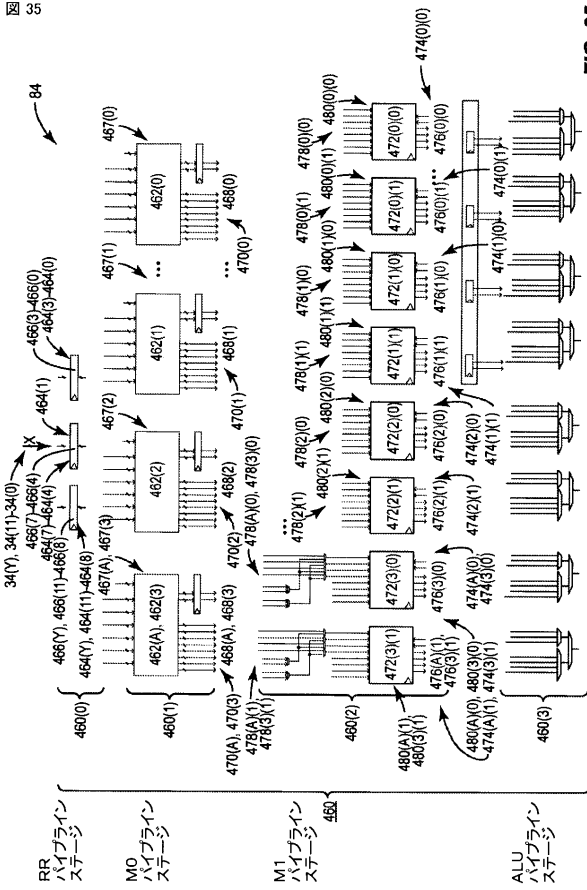


FIG. 35

【 図 3 7 】

図 37

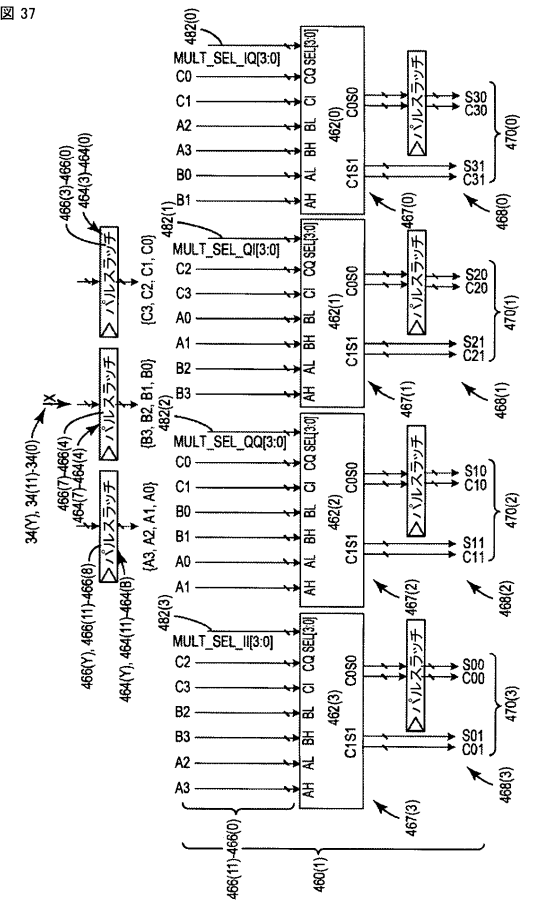


FIG. 37

【 図 3 6 】

図 36

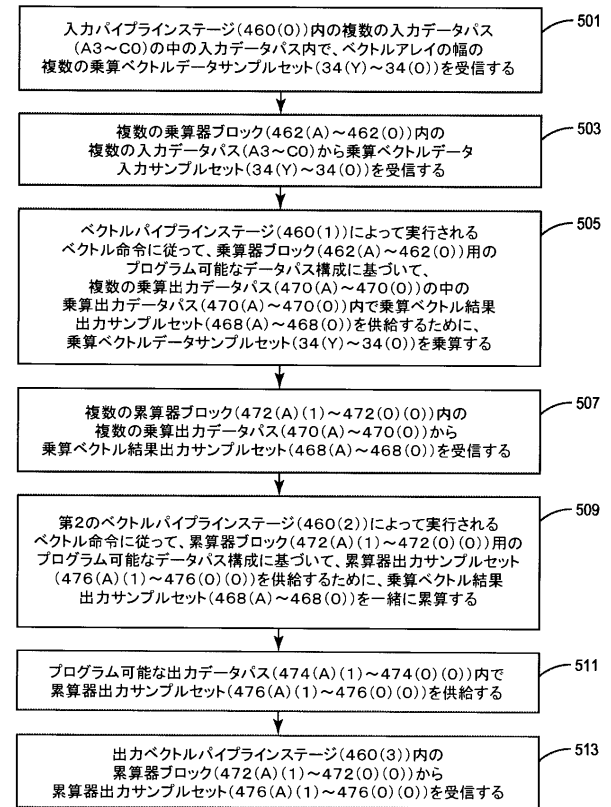


FIG. 36

【 図 3 8 】

図 38

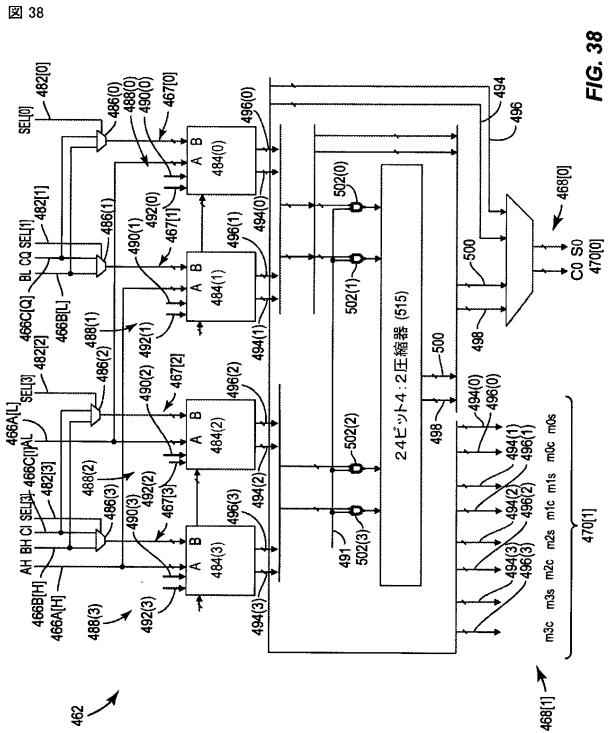


FIG. 38

【 図 3 9 】

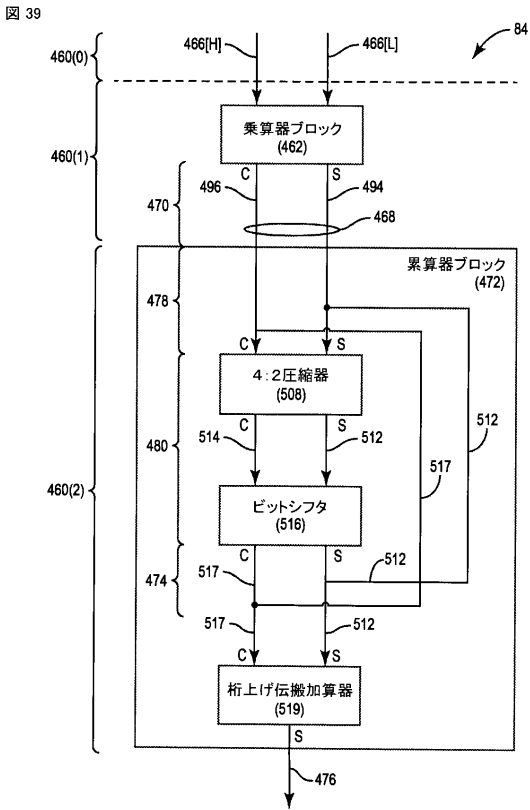


FIG. 39

【 図 4 0 】

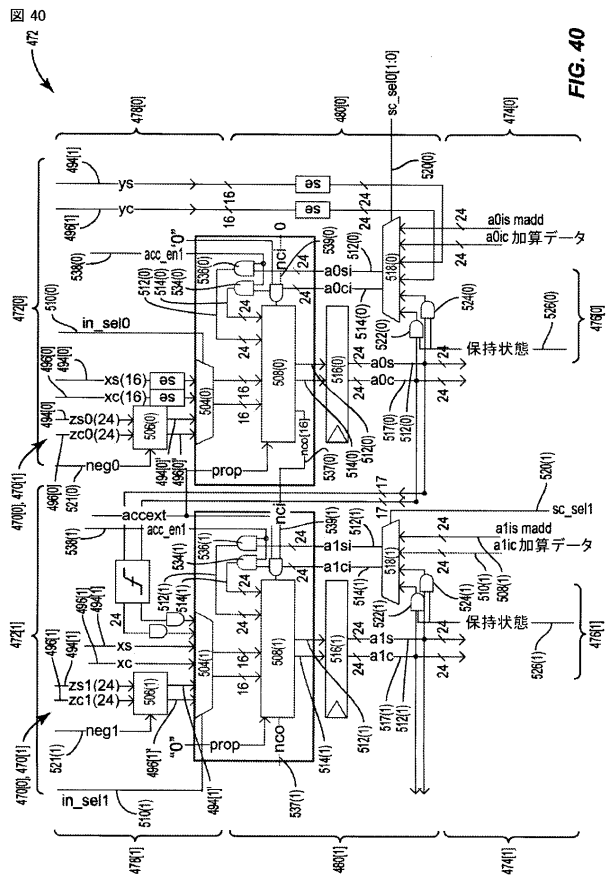


FIG. 40

【 図 4 1 】

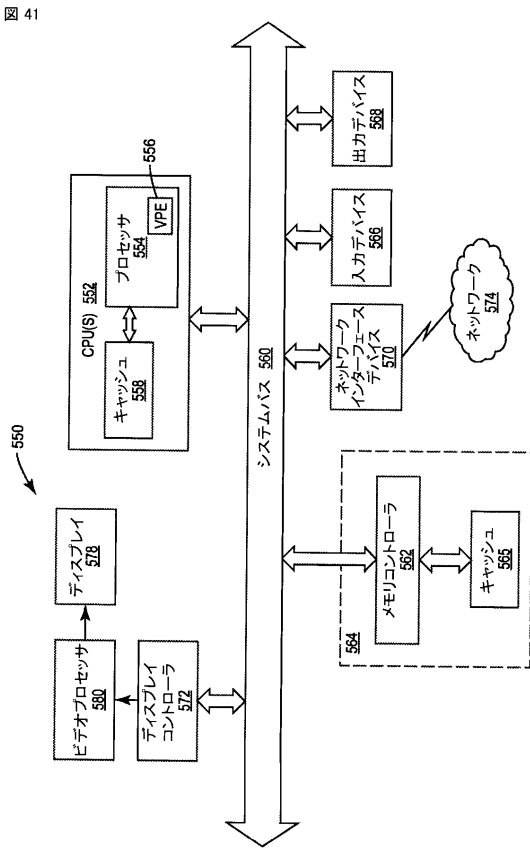


FIG. 41

【 国際調査報告 】

INTERNATIONAL SEARCH REPORT

International application No

PCT/US2014/065825

A. CLASSIFICATION OF SUBJECT MATTER INV. G06F9/38 G06F15/80 ADD.		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) G06F G05F		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) EPO-Internal		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2006/106342 A2 (ICERA INC [GB]; KNOWLES SIMON [GB] ICERA INC [GB]; KNOWLES SIMON [GB];) 12 October 2006 (2006-10-12) the whole document	1,2,4,6,8,10,12,13,15,17-22,24-26
X	US 2007/061550 A1 (BARLOW STEPHEN [GB] ET AL) 15 March 2007 (2007-03-15) paragraph [0041] - paragraph [0061] paragraph [0081] - paragraph [0082]	1,6,8,20,21
X	US 2008/140750 A1 (KERSHAW DANIEL [GB] ET AL) 12 June 2008 (2008-06-12) paragraph [0014] - paragraph [0015] paragraph [0025] paragraph [0073] - paragraph [0074]	1,20,21
	-/--	
<input checked="" type="checkbox"/> Further documents are listed in the continuation of Box C.		<input checked="" type="checkbox"/> See patent family annex.
* Special categories of cited documents :		
"A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed		"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family
Date of the actual completion of the international search		Date of mailing of the international search report
17 February 2015		27/02/2015
Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016		Authorized officer Klocke, Lynn

INTERNATIONAL SEARCH REPORT

International application No PCT/US2014/065825

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 2 455 854 A1 (CEVA D S P LTD [IL]) 23 May 2012 (2012-05-23) paragraph [0038] paragraph [0040] paragraph [0057] -----	1,20,21
A	WO 2009/144683 A1 (NXP BV [NL]; SMRITI MAHIMA [NL]; SMEETS JEAN-PAUL CHARLES FRANC [NL];) 3 December 2009 (2009-12-03) the whole document -----	1,20,21
A	US 2008/077768 A1 (INOUE HIROSHI [JP] ET AL) 27 March 2008 (2008-03-27) the whole document -----	1,20,21
A	US 2005/193185 A1 (TAUNTON MARK [GB] ET AL) 1 September 2005 (2005-09-01) the whole document -----	1,20,21

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2014/065825

Patent document cited in search report	Publication date	Patent family member(s)	Publication date	
WO 2006106342	A2	12-10-2006	BR P10609742 A2	18-10-2011
			CA 2600744 A1	12-10-2006
			CN 101208658 A	25-06-2008
			CN 103744639 A	23-04-2014
			EP 1866744 A2	19-12-2007
			JP 5047944 B2	10-10-2012
			JP 2008535115 A	28-08-2008
			KR 20070118623 A	17-12-2007
			US 2006227966 A1	12-10-2006
			US 2006106342 A2	12-10-2006
			US 2007061550	A1
US 2003154361 A1	14-08-2003			
US 2007061550 A1	15-03-2007			
US 2008140750	A1	12-06-2008	NONE	
EP 2455854	A1	23-05-2012	CA 2758366 A1	18-05-2012
			EP 2455854 A1	23-05-2012
			US 2012131308 A1	24-05-2012
WO 2009144683	A1	03-12-2009	CN 102047219 A	04-05-2011
			EP 2300911 A1	30-03-2011
			US 2011314254 A1	22-12-2011
			WO 2009144683 A1	03-12-2009
US 2008077768	A1	27-03-2008	US 2008077768 A1	27-03-2008
			US 2009222644 A1	03-09-2009
			US 2013042092 A1	14-02-2013
US 2005193185	A1	01-09-2005	NONE	

フロントページの続き

(81)指定国 AP(BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), EA(AM, AZ, BY, KG, KZ, RU, TJ, TM), EP(AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG), AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US

(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

Fターム(参考) 5B013 AA14 DD01 DD03

5B033 BE00

5B056 BB32 FF10 FF14 FF16