

(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2016-514330

(P2016-514330A)

(43) 公表日 平成28年5月19日(2016.5.19)

(51) Int.Cl.	F I	テーマコード (参考)
G06F 17/16 (2006.01)	G06F 17/16	M 5B056
G06F 17/14 (2006.01)	G06F 17/14	Z

審査請求 未請求 予備審査請求 未請求 (全 47 頁)

(21) 出願番号 特願2016-500848 (P2016-500848)
 (86) (22) 出願日 平成26年3月7日 (2014.3.7)
 (85) 翻訳文提出日 平成27年11月2日 (2015.11.2)
 (86) 国際出願番号 PCT/US2014/021782
 (87) 国際公開番号 W02014/164298
 (87) 国際公開日 平成26年10月9日 (2014.10.9)
 (31) 優先権主張番号 13/798,599
 (32) 優先日 平成25年3月13日 (2013.3.13)
 (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 100194814
 弁理士 奥村 元宏

最終頁に続く

(54) 【発明の名称】 マルチモード基数2のX乗のバタフライベクトル処理回路を提供するためのプログラマブルなデータパス構成を有するベクトル処理エンジン、ならびに関連ベクトルプロセッサ、システム、お

(57) 【要約】

マルチモード基数 2^X のバタフライベクトル処理回路を提供するためのプログラマブルなデータパス構成を有するベクトル処理エンジン (VPE)。関連ベクトルプロセッサ、システム、および方法についても開示する。本明細書で開示するVPEは、高速フーリエ変換 (FFT) ベクトル処理演算を効率的に実施するために、基数 2^X のバタフライベクトル演算を実施するためのプログラマブルなデータパス構成を有するベクトル処理ブロックを各々が有する複数のベクトル処理段階を含む。ベクトル処理ブロックのデータパス構成は、異なるタイプの基数 2^X のバタフライベクトル演算ならびに他の算術論理ベクトル演算を提供するようにプログラムされ得る。その結果、より少ないVPEで、ベクトルプロセッサにおいて所望の基数 2^X のバタフライベクトル演算と他のタイプ算術論理ベクトル演算を行うことができ、したがって、ベクトルプロセッサ中の面積を節約するとともに、スカラー処理エンジンに勝る、より少ないレジスタ書込みおよびより速いベクトル命令実行時間というベクトル処理利点を依然として保有する。

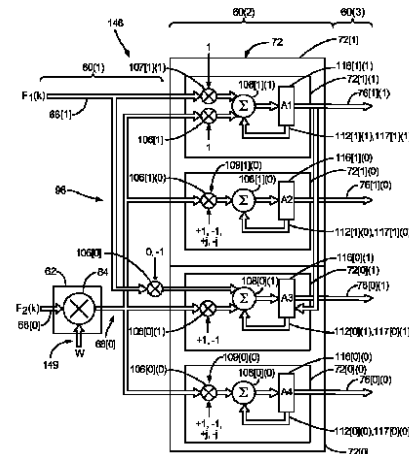


FIG. 11

【特許請求の範囲】**【請求項 1】**

少なくとも 1 つのマルチモード基数 2^x のバタフライベクトル処理回路を装備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、前記ベクトル処理エンジンは下記を備える、

少なくとも 1 つの乗算ベクトル処理段階であって、

複数の入力データパスの中の第 1 の入力データパスから、複数の基数ベクトルデータ入力のサンプルセットからの基数ベクトルデータ入力のサンプルセットを受け取り、

前記少なくとも 1 つの乗算ベクトル処理段階によって実行される基数バタフライベクトル命令に従って、プログラマブルな乗算データパス構成に基づいて複数の乗算出力データパスにおいて基数ベクトル乗算出力のサンプルセットを与えるように、前記基数ベクトルデータ入力のサンプルセットを回転因子成分と乗算する、

10

ように構成された少なくとも 1 つの乗算器ブロックを備える乗算ベクトル処理段階と、

複数の累算器ブロックを備える少なくとも 1 つの累算ベクトル処理段階であって、前記複数の累算器ブロックの中の各累算器ブロックは、

累算器プログラマブルな累算器データパス構成に基づいて、前記複数の乗算出力データパスの中の乗算出力データパスから複数の基数ベクトル乗算出力のサンプルセットを受け取り、

前記累算器プログラマブルな累算器データパス構成に基づいて、基数ベクトル累算結果のサンプルセットを与えるように、前記受け取られた複数の基数ベクトル乗算出力のサンプルセットを累算し、

20

複数の出力データパスの中の出力データパスにおいて前記基数ベクトル累算結果のサンプルセットを与えるように構成される、

前記累算ベクトル処理段階と、

前記複数の累算器ブロックの各々から前記複数の基数ベクトル累算結果のサンプルセットを受け取るように構成された出力処理段階。

【請求項 2】

前記基数バタフライベクトル命令についての基数バタフライ結果を与えるように、前記複数の基数ベクトル累算結果のサンプルセットを加えるように構成された少なくとも 1 つの加算器をさらに備える、請求項 1 に記載の V P E 。

30

【請求項 3】

前記回転因子成分が「e」である、請求項 1 に記載の V P E 。

【請求項 4】

前記複数の累算器ブロックの中の各累算器ブロックは、前記基数バタフライベクトル命令に従って、各累算器ブロック用の前記プログラマブルな累算器データパス構成に基づいて前記受け取られた複数の基数ベクトル乗算出力のサンプルセットに回転因子入力を適用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 5】

前記複数の累算器ブロックの中の各累算器ブロックは、前記基数バタフライベクトル命令に従って、各累算器ブロック用の前記プログラマブルな累算器データパス構成に基づいてマイナス 1 (- 1) の前記回転因子入力を適用するように、前記複数の基数ベクトル乗算出力のサンプルセットの中の受け取られた基数ベクトル乗算出力のサンプルセットを否定するように構成された否定回路をさらに備える、請求項 4 に記載の V P E 。

40

【請求項 6】

前記複数の累算器ブロックの中の各累算器ブロックは、前記基数バタフライベクトル命令に従って、各累算器ブロック用の前記プログラマブルな累算器データパス構成に基づいて、「-j」の前記回転因子入力を適用するように、前記否定された基数ベクトル乗算出力のサンプルセットを、否定された複素基数ベクトル乗算出力のサンプルセットに変換するように構成されたスワップ回路をさらに備える、請求項 5 に記載の V P E 。

【請求項 7】

50

前記複数の累算器ブロックの中の各累算器ブロックは、前記基数バタフライベクトル命令に従って、各累算器ブロック用の前記プログラマブルな累算器データバス構成に基づいて、「j」の前記回転因子入力を適用するように、前記複数の基数ベクトル乗算出力のサンプルセットの中の受け取られた基数ベクトル乗算出力のサンプルセットを複素基数ベクトル乗算出力のサンプルセットに変換するように構成されたスワップ回路をさらに備える、請求項 4 に記載の V P E。

【請求項 8】

前記複数の累算器ブロックの中の各累算器ブロックは、基数 2 のバタフライベクトル命令に従って構成された各累算器ブロック用の前記プログラマブルな累算器データバス構成に基づいて、基数 2 のバタフライ回転因子入力として、前記回転因子入力を前記受け取られた複数の基数ベクトル乗算出力のサンプルセットに適用するように構成される、請求項 4 に記載の V P E。

【請求項 9】

前記複数の累算器ブロックの中の各累算器ブロックは、基数 4 のバタフライベクトル命令に従って構成された各累算器ブロック用の前記プログラマブルな累算器データバス構成に基づいて、基数 4 のバタフライ回転因子入力として、前記回転因子入力を前記受け取られた複数の基数ベクトル乗算出力のサンプルセットに適用するように構成される、請求項 4 に記載の V P E。

【請求項 10】

前記回転因子入力を生成するように構成された回転因子ジェネレータをさらに備える、請求項 4 に記載の V P E。

【請求項 11】

前記少なくとも 1 つの乗算器ブロックは、複数のマルチモード基数 2^x のバタフライベクトル処理回路を装備するための複数の乗算器ブロックからなる、請求項 1 に記載の V P E。

【請求項 12】

前記プログラマブルな乗算データバス構成および前記プログラマブルな累算器データバス構成は各々、前記実行される基数バタフライベクトル命令に基づいて再構成されるように構成される、請求項 1 に記載の V P E。

【請求項 13】

前記プログラマブルな乗算データバス構成および前記プログラマブルな累算器データバス構成は各々、前記 V P E によって実行される各基数のバタフライベクトル命令用に再構成されるように構成される、請求項 12 に記載の V P E。

【請求項 14】

前記プログラマブルな乗算データバス構成および前記プログラマブルな累算器データバス構成は各々、前記 V P E によって実行される前記基数バタフライベクトル命令の各クロック周期において再構成されるように構成される、請求項 13 に記載の V P E。

【請求項 15】

前記複数の累算器ブロックの中の各累算器ブロックは、

前記複数の基数ベクトル乗算出力のサンプルセットを、各々が少なくとも 1 つの基数ベクトル入力値と少なくとも 1 つの基数ベクトル入力値上げとを備えるものとして受け取り、

少なくとも 1 つの以前の累算された基数ベクトル出力値と少なくとも 1 つの以前の累算された基数ベクトル出力値上げとを受け取り、

前記少なくとも 1 つの現在の基数ベクトル累算出力値として、前記少なくとも 1 つの以前の累算された基数ベクトル出力値に累算された前記少なくとも 1 つの基数ベクトル入力値からなる少なくとも 1 つの現在の累算された基数ベクトル出力値を累算し、

前記少なくとも 1 つの現在の累算された基数ベクトル出力値上げとして、前記少なくとも 1 つの以前の累算された基数ベクトル出力値上げに累算された前記少なくとも 1 つの基数ベクトル入力値上げからなる少なくとも 1 つの現在の累算された基数ベクトル出力値

10

20

30

40

50

上げを累算し、

前記少なくとも1つの現在の累算された基数ベクトル出力和と前記少なくとも1つの現在の累算された基数ベクトル出力桁上げとを、前記複数の出力データパスの中の前記出力データパスにおいて、前記基数ベクトル累算結果のサンプルセットの中の少なくとも1つの基数ベクトル累算結果のサンプルセットとして与える、

ように各々が構成された少なくとも1つの桁上げ保存累算器を備える、請求項1に記載のVPE。

【請求項16】

前記少なくとも1つの桁上げ保存累算器は、前記少なくとも1つの以前の累算された基数ベクトル出力桁上げを、前記少なくとも1つの基数ベクトル入力および前記少なくとも1つの基数ベクトル入力桁上げに伝搬しないように構成される、請求項15に記載のVPE。

10

【請求項17】

前記少なくとも1つの桁上げ保存累算器は、前記少なくとも1つの現在の累算された基数ベクトル出力和を第1のベクトル累算データパスにおいて、および前記少なくとも1つの現在の累算された基数ベクトル出力桁上げを、前記第1のベクトル累算データパスとは別個の第2のベクトル累算データパスにおいて維持するようにさらに構成される、請求項15に記載のVPE。

【請求項18】

前記少なくとも1つの桁上げ保存累算器は、
前記少なくとも1つの基数ベクトル入力と前記少なくとも1つの基数ベクトル入力桁上げとを受け取り、

20

前記少なくとも1つの以前の累算された基数ベクトル出力和と前記少なくとも1つの以前の累算された基数ベクトル出力桁上げとを受け取り、

前記少なくとも1つの現在の基数ベクトル累算された出力和として、前記少なくとも1つの以前の累算された基数ベクトル出力和に累算された前記少なくとも1つの基数ベクトル入力からなる前記少なくとも1つの現在の累算された基数ベクトル出力和を生成し、

前記少なくとも1つの現在の累算された基数ベクトル出力桁上げとして、前記少なくとも1つの以前の累算された基数ベクトル出力桁上げに累算された前記少なくとも1つの基数ベクトル入力桁上げからなる前記少なくとも1つの現在の累算された基数ベクトル出力桁上げを生成する、

30

ように構成された少なくとも1つの圧縮器を備える、請求項15に記載のVPE。

【請求項19】

前記少なくとも1つの圧縮器は少なくとも1つの4:2圧縮器からなる、請求項18に記載のVPE。

【請求項20】

前記複数の入力データパスの中の入力データパスにおいて、ベクトルアレイの幅の前記複数の基数ベクトルデータ入力のサンプルセットを与えるように構成された入力読取り段階をさらに備える、請求項1に記載のVPE。

【請求項21】

前記複数の累算器ブロックは、前記受け取られた複数の基数ベクトル乗算出力のサンプルセットをベクトルレジスタに記憶するように構成されない、請求項1に記載のVPE。

40

【請求項22】

前記少なくとも1つのマルチモード基数 2^x のバタフライベクトル処理回路は、高速フーリエ変換(FFT)ベクトル命令を実行するように構成される、請求項1に記載のVPE。

【請求項23】

前記少なくとも1つの基数 2^x のバタフライベクトル処理回路は、補間FFTベクトル命令を実行するように構成される、請求項1に記載のVPE。

【請求項24】

50

少なくとも1つのマルチモード基数 2^x のバタフライベクトル処理回路を装備するように構成されたベクトル処理エンジン（VPE）であって、前記ベクトル処理エンジンは下記を備える、

少なくとも1つの乗算ベクトル処理段階手段であって、

複数の入力データパスの中の第1の入力データパスから、複数の基数ベクトルデータ入力のサンプルセットからの基数ベクトルデータ入力のサンプルセットを受け取り、

前記少なくとも1つの乗算ベクトル処理段階手段によって実行される基数バタフライベクトル命令に従って、プログラマブルな乗算データパス構成に基づいて複数の乗算出力データパスにおいて基数ベクトル乗算出力のサンプルセットを与えるように、前記基数ベクトルデータ入力のサンプルセットを回転因子成分と乗算する、

ように構成された少なくとも1つの乗算器ブロック手段を備える乗算ベクトル処理段階手段と、

複数の累算器ブロック手段を備える少なくとも1つの累算ベクトル処理段階手段であって、前記複数の累算器ブロック手段の中の各累算器ブロック手段は、

プログラマブルな累算器データパス構成に基づいて、前記複数の乗算出力データパスの中の乗算出力データパスから複数の基数ベクトル乗算出力のサンプルセットを受け取り、

前記プログラマブルな累算器データパス構成に基づいて基数ベクトル累算結果のサンプルセットを与えるように、前記受け取られた複数の基数ベクトル乗算出力のサンプルセットを累算し、

複数の出力データパスの中の出力データパスにおいて前記基数ベクトル累算結果のサンプルセットを与える、

ように構成される、前記累算ベクトル処理段階手段と、

前記複数の累算器ブロックの各々から複数の基数ベクトル累算結果のサンプルセットを受け取るように構成された出力処理段階手段。

【請求項25】

ベクトル処理エンジン（VPE）におけるマルチモード基数 2^x のバタフライベクトル処理回路処理において基数 2^x のベクトルデータを処理する方法であって、前記方法は下記を備える、

少なくとも1つの乗算ベクトル処理段階における少なくとも1つの乗算器ブロック中の複数の入力データパスの中の第1の入力データパスから、複数の基数ベクトルデータ入力のサンプルセットからの基数ベクトルデータ入力のサンプルセットを受け取ることと、

前記少なくとも1つの乗算器ブロック中で、前記少なくとも1つの乗算ベクトル処理段階によって実行される基数バタフライベクトル命令に従って、プログラマブルな乗算データパス構成に基づいて複数の乗算出力データパスにおいて基数ベクトル乗算出力のサンプルセットを与えるように、前記基数ベクトルデータ入力のサンプルセットを回転因子成分と乗算することと、

少なくとも1つの累算ベクトル処理段階における複数の累算器ブロックの中の各累算器ブロック中の累算器プログラマブルな累算器データパス構成に基づいて、前記複数の乗算出力データパスの中の乗算出力データパスから複数の基数ベクトル乗算出力のサンプルセットを受け取ることと、

前記累算器プログラマブルな累算器データパス構成に基づいて基数ベクトル累算結果のサンプルセットを与えるように、前記受け取られた複数の基数ベクトル乗算出力のサンプルセットを累算することと、

複数の出力データパスの中の出力データパスにおいて前記基数ベクトル累算結果のサンプルセットを与えることと、

出力処理段階において、前記複数の累算器ブロックの各々から複数の基数ベクトル累算結果のサンプルセットを受け取ること。

【請求項26】

前記基数バタフライベクトル命令についての基数バタフライ結果を与えるように、前記

10

20

30

40

50

複数の基数ベクトル累算結果のサンプルセットを加え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】

前記基数バタフライベクトル命令に従って、各累算器ブロック用の前記プログラマブルな累算器データパス構成に基づいて、前記受け取られた複数の基数ベクトル乗算出力のサンプルセットに回転因子入力を適用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 8】

前記基数バタフライベクトル命令に従って、各累算器ブロック用の前記プログラマブルな累算器データパス構成に基づいてマイナス 1 (- 1) の前記回転因子入力を適用するように、前記複数の基数ベクトル乗算出力のサンプルセットの中の受け取られた基数ベクトル乗算出力のサンプルセットを否定することをさらに備える、請求項 2 7 に記載の方法。

10

【請求項 2 9】

前記基数バタフライベクトル命令に従って、各累算器ブロック用の前記プログラマブルな累算器データパス構成に基づいて「 - j 」の前記回転因子入力を適用するために、前記否定された基数ベクトル乗算出力のサンプルセットを、否定された複素基数ベクトル乗算出力のサンプルセットに変換するように、前記受け取られた基数ベクトル乗算出力のサンプルセットをスワップすることをさらに備える、請求項 2 8 に記載の方法。

【請求項 3 0】

前記基数バタフライベクトル命令に従って、各累算器ブロック用の前記プログラマブルな累算器データパス構成に基づいて「 j 」の前記回転因子入力を適用するために、前記複数の基数ベクトル乗算出力のサンプルセットの中の前記受け取られた基数ベクトル乗算出力のサンプルセットを複素基数ベクトル乗算出力のサンプルセットに変換するように、前記基数ベクトル乗算出力のサンプルセットをスワップすることをさらに備える、請求項 2 7 に記載の方法。

20

【請求項 3 1】

前記受け取られた複数の基数ベクトル乗算出力のサンプルセットをベクトルレジスタに記憶しないことをさらに備える、請求項 2 5 に記載の方法。

【発明の詳細な説明】

【技術分野】

【0 0 0 1】

30

関連出願

[0001]本出願は、2013年3月13日に出版し、参照によって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」と題する米国特許出願第13/798,618号(Qualcomm整理番号123248)にも関連する。

【0 0 0 2】

[0002]本出願は、2013年3月13日に出版し、参照によって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」と題する米国特許出願第13/798,641号(Qualcomm整理番号123249)に関連する。

40

【0 0 0 3】

[0003]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)プロセッサを含む、ベクトルおよびスカラー演算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。

50

【背景技術】

【0004】

[0004]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。技術の進歩により、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を含む。より具体的には、セルラー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およびデータパケット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が、他のタイプのデバイスを含む。たとえば、ワイヤレス電話機は、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を含み得る。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でき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））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み得る。したがって、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。

10

【0005】

[0005]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、小さいデバイスのサイズによって制限されることがある。さらに、電池サイズ、電池によって供給される電力の量、および電池の寿命も制限される。デバイスのバッテリー寿命を増すための1つのやり方は、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。

20

【0006】

[0006]この点において、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用に利用され得る。ベクトルプロセッサは、ベクトル、すなわちデータアレイに働きかけるハイレベル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、ベクトル命令を一度フェッチすることと、次いで、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フェッチし直し、復号するのは反対に、データ要素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のに必要とされるエネルギーが低減される。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。

30

【0007】

[0007]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中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用の関数固有のベクトル処理を提供するのに各々が専用である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）～12（5）がベースバンドプロセッサ10中に設けられる。PE12（0）～12（5）は各々、共有メモリ16からPE12（0）～12（5）に与えられる固定Xビット幅のベクトルデータ14用のベクトル処理を提供するように構成される。たとえば、ベクトルデータ14は512ビット幅であり得る。ベクトルデータ14は、Xのより小さい倍数のビット幅のベクトルデータのサンプルセット18（0）～18（Y）（たとえば、16ビットおよび32ビットサンプルセット）中で定義されてもよい。このようにして、PE12（0）～12（5）は、高度の並列性を達成するように、PE12（0）～12（5）に並列に与えられる複数のベクトルデータのサンプルセットに対するベクトル処理を提

40

50

供することが可能である。各 $PE\ 12(0) \sim 12(5)$ は、ベクトルデータ 14 上で処理されるベクトル命令の結果を記憶するためのベクトルレジスタファイル (VR) を含み得る。

【0008】

[0008] 図 1 のベースバンドプロセッサ 10 中の各 $PE\ 12(0) \sim 12(5)$ は、特定のタイプの固定した演算を効率的に実施するように特に設計された、特定の専用回路要素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、別個の広帯域符号分割多元接続 (WCDMA) $PE\ 12(0)$ 、 $12(1)$ とロングタームエボリューション (LTE) $PE\ 12(4)$ 、 $12(5)$ とを含むが、これは、WCDMA および LTE は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 $PE\ 12(0)$ 、 $12(1)$ と LTE 固有 $PE\ 12(4)$ 、 $12(5)$ とを設けることによって、 $PE\ 12(0)$ 、 $12(1)$ 、 $12(4)$ 、 $12(5)$ の各々は、高効率な演算のための、WCDMA および LTE 用の頻繁に実施される関数に固有である、特殊な専用回路要素を含むように設計され得る。この設計は、より多数の無関係な演算を、比較的非効率的にはあるがサポートするために柔軟であるように設計された、より一般的な回路要素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。

【0009】

[0009] ベクトル累算演算 (accumulation operation) は一般に、PE 中で実施される。この点において、PE は、効率的処理のために特定のベクトル累算演算をサポートするための特殊な回路要素とハードウェアとを各々が有する関数固有の累算器 (accumulator) 構造を含む。ベクトル累算演算を利用する PE によってサポートされる共通ベクトル演算の例には、例として、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 (FFT) ベクトル処理演算を実施するために一般に使われる、フィルタリング演算、相関演算、および基数 2^x (たとえば、基数 2 および基数 4) のバタフライ演算がある。PE 中で関数固有の累算器構造を提供することは、頻繁に実行される、特殊な累算演算にとって有利である。ただし、PE 中で関数固有の累算器構造を設けると、PE 中で設けられる別個の関数固有の累算器構造が各々、特殊な回路要素とメモリとを含むので、ベースバンドプロセッサ用に必要とされる面積および電力が増大する場合がある。

【発明の概要】

【0010】

[0010] 本明細書で開示する実施形態は、マルチモード基数 2^x のバタフライベクトル処理回路を提供するためのプログラマブルな (programmable) データパス (data path) 構成を有するベクトル処理エンジン (VPE) を含む。関連ベクトルプロセッサ、システム、および方法についても開示する。本明細書で開示する VPE は、高速フーリエ変換 (FFT) ベクトル処理を効率的に実施するために、基数 2^x のバタフライベクトル演算を実施するためのプログラマブルなデータパス構成を有するベクトル処理ブロックを各々が有する複数のベクトル処理段階を含む。ベクトル処理ブロックのデータパス構成は、異なるタイプの基数 2^x のバタフライベクトル演算ならびに他の算術論理ベクトル演算を提供するようにプログラムされ得る。その結果、より少ない VPE で、ベクトルプロセッサにおいて所望の基数 2^x のバタフライベクトル演算と他のタイプの算術論理ベクトル演算とを行うことができ、したがって、ベクトルプロセッサ中の面積を節約するとともに、スカラー処理エンジンに勝る、より少ないレジスタ書込みおよびより速いベクトル命令実行時間というベクトル処理利点を依然として保持する。

【0011】

[0011] この点において、本明細書で開示するいくつかの実施形態では、プログラマブルなデータパス構成を有する複数の乗算器ブロックが、VPE 中のベクトル処理段階に設けられる。複数の乗算器ブロックは、VPE 中の別のベクトル処理段階に設けられたプログラマブルなデータパス構成を有する複数の累算器ブロックと融合される。各乗算器ブロックのデータパスは、実行される特定の基数 2^x のバタフライベクトル命令に従って、基数 2^x のバタフライベクトル演算のポイントについての乗算を行うようにプログラムされ得

る。各累算器ブロックのデータパスは、実行される基数 2^x のバタフライベクトル命令に従って、複数の乗算器ブロックの中のある乗算器ブロックからの基数 2^x のバタフライ乗算積を累算するようにプログラムされ得る。累算器ブロックのデータパス構成はまた、他の周期中で、実行される基数 2^x のバタフライベクトル命令に従って、複数の乗算器ブロックの中の他の乗算器ブロックからの基数 2^x のバタフライ乗算積を累算するように再プログラムされ得る。累算器ブロックによって累算された最終的な累算基数 2^x のバタフライ乗算積は、FFTベクトル処理演算のN個のポイントを与えることができる。

【0012】

[0012]マルチモード基数 2^x のバタフライベクトル処理を提供するためのプログラマブルなデータパス構成を有するVPEは、固定した関数を提供するための固定したデータパス構成を含むだけであるVPEとは区別可能である。本明細書で開示するマルチモード基数 2^x のバタフライベクトル処理を提供するためのプログラマブルなデータパス構成を有するVPEは、たとえばデジタル信号プロセッサ(DSP)中に設けられるようなスカラー処理エンジンとも区別可能である。スカラー処理エンジンは、異なるタイプの固定しない関数を実施するために、柔軟な、共通回路と論理を採用するが、同時に、レジスタファイルにベクトル命令実行中に中間結果を書き込む、それによって、追加の電力を消費し、ベクトル命令実行回数を増大させる。

【0013】

[0013]この点において一実施形態では、少なくとも1つのマルチモード基数 2^x のバタフライベクトル処理回路を行うように構成されたVPEが提供される。VPEは、少なくとも1つの乗算器ブロックを備える少なくとも1つの乗算ベクトル処理段階を備える。各乗算器ブロックは、複数の入力データパスの中の第1の入力データパスから、複数の基数ベクトルデータ入力のサンプルセットからの基数ベクトルデータ入力のサンプルセットを受け取るように構成される。各乗算器ブロックはまた、少なくとも1つの乗算ベクトル処理段階によって実行される基数バタフライベクトル命令に従って、プログラマブルな乗算データパス構成に基づいて複数の乗算出力データパスにおいてで基数ベクトル乗算出力のサンプルセットを与えるように、基数ベクトルデータ入力のサンプルセットを回転因子成分と乗算する構成される。VPEは、複数の累算器ブロックを備える少なくとも1つの累算ベクトル処理段階も備える。複数の累算器ブロックの中の各累算器ブロックは、累算器プログラマブルな累算器データパス構成に基づいて、複数の乗算出力データパスの中のある乗算出力データパスから複数の基数ベクトル乗算出力のサンプルセットを受け取るように構成される。複数の累算器ブロックの中の各累算器ブロックはさらに、累算器プログラマブルな累算器データパス構成に基づいて基数ベクトル累算結果のサンプルセットを与えるように、受け取られた複数の基数ベクトル乗算出力のサンプルセットを累算する構成される。複数の累算器ブロックの中の各累算器ブロックはさらに、複数の出力データパスの中のある出力データパスにおいて基数ベクトル累算結果のサンプルセットを与える構成される。VPEは、複数の累算器ブロックの各々から複数の基数ベクトル累算結果のサンプルセットを受け取るように構成された出力処理段階も備える。

【0014】

[0014]別の実施形態では、少なくとも1つのマルチモード基数 2^x のバタフライベクトル処理回路を行うように構成されたVPEが提供される。VPEは、少なくとも1つの乗算器ブロック手段を備える少なくとも1つの乗算ベクトル処理段階手段を備える。各乗算器ブロック手段は、複数の入力データパスの中の第1の入力データパスから、複数の基数ベクトルデータ入力のサンプルセットからの基数ベクトルデータ入力のサンプルセットを受け取るように構成される。各乗算器ブロック手段はまた、少なくとも1つの乗算ベクトル処理段階手段によって実行される基数バタフライベクトル命令に従って、プログラマブルな乗算データパス構成に基づいて複数の乗算出力データパスにおいてで基数ベクトル乗算出力のサンプルセットを与えるように、基数ベクトルデータ入力のサンプルセットを回転因子成分と乗算する構成される。VPEは、複数の累算器ブロック手段を備える少なくとも1つの累算ベクトル処理段階手段も備える。複数の累算器ブロックの手段中の各累算

10

20

30

40

50

器ブロック手段は、プログラマブルな累算器データバス構成に基づいて、複数の乗算出力データバスの中のある乗算出力データバスから複数の基数ベクトル乗算出力のサンプルセットを受け取るように構成される。複数の累算器ブロック手段の中の各累算器ブロック手段はさらに、プログラマブルな累算器データバス構成に基づいて基数ベクトル累算結果のサンプルセットを与えるように、受け取られた複数の基数ベクトル乗算出力のサンプルセットを累算する構成される。複数の累算器ブロック手段の中の各累算器ブロック手段はさらに、複数の出力データバスの中のある出力データバスにおいて基数ベクトル累算結果のサンプルセットを与える構成される。VPEは、複数の累算器ブロックの各々から複数の基数ベクトル累算結果のサンプルセットを受け取るように構成された出力処理段階手段も備える。

10

【0015】

[0015]別の実施形態では、ベクトル処理エンジン(VPE)中のマルチモード基数 2^x のバタフライベクトル処理回路処理において基数 2^x のベクトルデータを処理する方法が提供される。方法は、少なくとも1つの乗算ベクトル処理段階における少なくとも1つの乗算器ブロック中の複数の入力データバスの中の第1の入力データバスから、複数の基数ベクトルデータ入力のサンプルセットからのある基数ベクトルデータ入力のサンプルセットを受け取ることを備える。方法は、少なくとも1つの乗算器ブロック中で、少なくとも1つの乗算ベクトル処理段階によって実行される基数バタフライベクトル命令に従って、プログラマブルな乗算データバス構成に基づいて複数の乗算出力データバスにおいて基数ベクトル乗算出力のサンプルセットを与えるように、基数ベクトルデータ入力のサンプル

セットを回転因子成分と乗算することも備える。方法は、少なくとも1つの累算ベクトル処理段階における複数の累算器ブロックの中のある累算器ブロック中の累算器プログラマブルな累算器データバス構成に基づいて、複数の乗算出力データバスの中のある乗算出力データバスから複数の基数ベクトル乗算出力のサンプルセットを受け取ることも備える。方法は、累算器プログラマブルな累算器データバス構成に基づいて基数ベクトル累算結果のサンプルセットを与えるように、受け取られた複数の基数ベクトル乗算出力のサンプルセットを累算することも備える。方法は、複数の出力データバスにおけるある出力データバスにおいて基数ベクトル累算結果のサンプルセットを与えることも備える。方法は、出力処理段階において、複数の累算器ブロックの各々から複数の基数ベクトル累算結果のサンプルセットを受け取ることも備える。

20

30

【0016】

[0016]別の実施形態では、ベクトルプロセッサが提供される。ベクトルプロセッサは、ベクトルデータメモリから基数ベクトルデータ入力のサンプルセットを受け取るように構成された少なくとも1つのベクトルレジスタを備える。ベクトルプロセッサは、基数ベクトル結果出力のサンプルセットを記憶するように構成された少なくとも1つのベクトル結果レジスタも備える。ベクトルプロセッサは、VPEも備える。VPEは、少なくとも1つの乗算ベクトル処理段階を備える。乗算ベクトル処理段階は、少なくとも1つの乗算ベクトル処理段階における少なくとも1つの乗算器ブロック中の複数の入力データバスの中の第1の入力データバスから、複数の基数ベクトルデータ入力のサンプルセットからのある基数ベクトルデータ入力のサンプルセットを受け取るように構成される。乗算ベクトル

処理段階は、少なくとも1つの乗算器ブロック中で、少なくとも1つの乗算ベクトル処理段階によって実行される基数バタフライベクトル命令に従って、プログラマブルな乗算データバス構成に基づいて複数の乗算出力データバスにおいて基数ベクトル乗算出力のサンプルセットを与えるように、基数ベクトルデータ入力のサンプルセットを回転因子成分と乗算するようにも構成される。VPEは、少なくとも1つの累算ベクトル処理段階も備える。累算ベクトル処理段階は、少なくとも1つの累算ベクトル処理段階における複数の累算器ブロックの中の各累算器ブロック中のプログラマブルな累算器データバス構成に基づいて、複数の乗算出力データバスの中のある乗算出力データバスから複数の基数ベクトル乗算出力のサンプルセットを受け取るように構成される。累算ベクトル処理段階は、少なくとも1つの累算器ブロック中で、プログラマブルな累算器データバス構成に基づいて基

40

50

数ベクトル累算結果のサンプルセットを与えるように、受け取られた複数の基数ベクトル乗算出力のサンプルセットを累算するようにも構成される。累算ベクトル処理段階は、複数の出力データパスの中のある出力データパスにおいて基数ベクトル累算結果のサンプルセットを与えるようにも構成される。VPEは、複数の累算器ブロックの各々から複数の基数ベクトル累算結果のサンプルセットを受け取るように構成された出力処理段階も備える。

【図面の簡単な説明】

【0017】

【図1】[0017]固有のアプリケーション向けの関数固有のベクトル処理を提供するのに各々が専用である、複数のベクトル処理エンジン(VPE)を含む例示的なベクトルプロセッサの概略図。

10

【図2】[0018]VPE中に設けられた共通回路要素およびハードウェアが、別個のVPEを設ける必要なく、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的に実施するための複数のモードでプログラムされ得るように、プログラマブルなデータパス構成を有する共通VPEを含む例示的なベクトルプロセッサの概略図。

【図3】[0019]図2のVPEに設けられる例示的なベクトル処理段階の概略図であって、ベクトル処理段階のうちのいくつかは、プログラマブルなデータパス構成を有する例示的なベクトル処理ブロックを含む。

【図4A】[0020]図2および図3の例示的なベクトルプロセッサ中に含まれるプログラマブルなデータパス構成を有する少なくとも1つのベクトル処理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。

20

【図4B】[0021]プログラマブルなデータパス構成を各々が有するとともに、図2および図3の例示的なベクトルプロセッサにおける異なるベクトル処理段階において提供さ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5】[0022]図2および図3のVPEのベクトル処理段階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は各々、複数の乗算器ブロックが、特定の異なるタイプのベクトル乗算演算を実施するための複数のモードでプログラムされ得るように、プログラマブルなデータパス構成を有する。

【図6】[0023]8ビット×8ビットベクトルデータ入力のサンプルセットおよび16ビット×16ビットベクトルデータ入力のサンプルセットについての乗算演算を行うようにプログラムされることが可能なプログラマブルなデータパス構成を有する、図5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。

30

【図7】[0024]図2および図3のVPE中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、累算器ブロックは、桁上げ伝搬を低減するの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。

【図8】[0025]図2および図3のVPE中に設けられる、図7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で、特定の異なるタイプのベクトル累算演算を実施するための複数のモードでプログラムされ得るように、累算器ブロックはプログラマブルなデータパス構成を有する。

【図9A】[0026]二重24ビット累算器モードを提供するためにプログラムされたデータパス構成を有する図8の累算器ブロックの概略図。

40

【図9B】[0027]単一40ビット累算器モードを提供するためにプログラムされたデータパス構成を有する図8の累算器ブロックの概略図。

【図9C】[0028]16ビット桁上げ保存加算器と、それに続く単一24ビット累算器モードを提供するためにプログラムされたデータパス構成を有する図8の累算器ブロックの概略図。

【図10】[0029]基数4のバタフライベクトル演算の各周期を実施するように構成された基数バタフライ回路を形成するために、図2および図3のVPE中で、冗長桁上げ保存フォーマットで4つの累算器と融合されるようにプログラムされたデータパス構成を有する乗算器の概念的概略図。

50

【図 1 1】[0030] 基数 2 または基数 4 のバタフライベクトル演算のいずれかを実施するように構成された、図 1 0 の基数バタフライ回路の 1 つの周期の概略図。

【図 1 2】[0031] 本明細書で開示する実施形態による、プログラマブルなデータバス構成を有する V P E を含むベクトルプロセッサを含むことができ、したがって V P E 中の共通回路要素およびハードウェアが、別個の V P E を提供する必要なく、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的に実施するように設計された専用回路要素として作用するようにプログラムされ得る、例示的なプロセッサベースのシステムのブロック図。

【発明を実施するための形態】

【 0 0 1 8 】

10

[0032] 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。「例示的な」という言葉は、本明細書中で、「例、事例、または例示として働くこと」を意味するように使用される。本明細書で「例示的な」として記載される任意の実施形態は、必ずしも、他の実施形態よりも好ましいまたは有利であると解釈されるべきでない。

【 0 0 1 9 】

[0033] 本明細書で開示する実施形態は、マルチモード基数 2^x のバタフライベクトル処理を提供するためのプログラマブルなデータバス構成を有するベクトル処理エンジン (V P E) を含む。関連ベクトルプロセッサ、システム、および方法についても開示する。本明細書で開示する V P E は、高速フーリエ変換 (F F T) ベクトル処理演算を効率的に実施するために、基数 2^x のバタフライベクトル演算を実施するためのプログラマブルなデータバス構成を有するベクトル処理ブロックを各々が有する複数のベクトル処理段階を含む。ベクトル処理ブロックのデータバス構成は、異なるタイプの基数 2^x のバタフライベクトル演算ならびに他の算術論理ベクトル演算を提供するようにプログラムされ得る。その結果、より少ない V P E で、ベクトルプロセッサにおいて所望の基数 2^x のバタフライベクトル演算と他のタイプの算術論理ベクトル演算とを行うことができ、したがって、ベクトルプロセッサ中の面積を節約するとともに、スカラー処理エンジンに勝る、より少ないレジスタ書込みおよびより速いベクトル命令実行時間というベクトル処理利点を依然として保持する。

20

【 0 0 2 0 】

[0034] この点において、本明細書で開示するいくつかの実施形態では、プログラマブルなデータバス構成を有する複数の乗算器ブロックが、V P E 中のベクトル処理段階に設けられる。複数の乗算器ブロックは、V P E 中の別のベクトル処理段階に設けられたプログラマブルなデータバス構成を有する複数の累算器ブロックと融合される。各乗算器ブロックのデータバスは、実行される特定の基数 2^x のバタフライベクトル命令に従って、基数 2^x バタフライベクトル演算のポイントについての乗算を行うようにプログラムされ得る。各累算器ブロックのデータバスは、実行される基数 2^x のバタフライベクトル命令に従って、複数の乗算器ブロックの中のある乗算器ブロックからの基数 2^x のバタフライ乗算積を累算するようにプログラムされ得る。累算器ブロックのデータバス構成はまた、他の周期中で、実行される基数 2^x のバタフライベクトル命令に従って、複数の乗算器ブロックの中の他の乗算器ブロックからの基数 2^x のバタフライ乗算積を累算するように再プログラムされ得る。累算器ブロックによって累算された最終的な累算基数 2^x のバタフライ乗算積は、F F T ベクトル処理演算の N 個のポイントを与えることができる。

30

40

【 0 0 2 1 】

[0035] マルチモード基数 2^x のバタフライベクトル処理を提供するためのプログラマブルなデータバス構成を有する V P E は、固定した関数を提供するための固定したデータバス構成を含むだけの V P E とは区別可能である。本明細書で開示するマルチモード基数 2^x のバタフライベクトル処理を提供するためのプログラマブルなデータバス構成を有する V P E は、たとえばデジタル信号プロセッサ (D S P) 中に設けられるようなスカラー処理エンジンとも区別可能である。スカラー処理エンジンは、異なるタイプの固定されない関数を実施するために柔軟な、共通回路と論理を採用するが、同時に、レジスタファイル

50

にベクトル命令実行中の中間結果を書き込む、それにより、追加の電力を消費し、ベクトル命令実行回数を増大させる。

【 0 0 2 2 】

[0036]この点において、図 2 は、ベクトル処理エンジン (V P E) 2 2 と呼ばれる、例示的なベクトル処理ユニット 2 2 を含むベースバンドプロセッサ 2 0 の概略図である。ベースバンドプロセッサ 2 0 およびその V P E 2 2 は、半導体ダイ 2 4 中に設けられ得る。図 3 で始まる、後でより詳しく論じる本実施形態では、ベースバンドプロセッサ 2 0 は、プログラマブルなデータパス構成を有する共通 V P E 2 2 を含む。このように、V P E 2 2 は、ベースバンドプロセッサ 2 0 中に別個の V P E を設ける必要なく、異なる演算モードで異なる特定のタイプのベクトル演算を行うようにプログラムされ、再プログラムされ得る共通回路要素とハードウェアとを含む。V P E 2 2 はまた、ベースバンドプロセッサ 2 0 中に別個の V P E を設ける必要なく、複数のアプリケーションまたは技術のために、一般的な算術演算を高効率的に実施するためのベクトル算術モードでプログラムされ得る。

【 0 0 2 3 】

[0037]図 3 で始まる、ベクトルマルチモード処理のために V P E 2 2 中に設けられるプログラマブルなデータパス構成について論じる前に、図 2 のベースバンドプロセッサ 2 0 の構成要素について、最初に記載する。この非限定的例におけるベースバンドプロセッサ 2 0 は、5 1 2 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 2 0 は、V P E 2 2 がベースバンドプロセッサ 2 0 中でベクトル処理を行うのをサポートするための追加構成要素を、V P E 2 2 に加えて含む。ベースバンドプロセッサ 2 0 は、ベクトルユニットデータメモリ (L M E M) 3 2 からベクトルデータ 3 0 を受け取り、記憶するように構成されたベクトルレジスタ 2 8 を含む。たとえば、ベクトルデータ 3 0 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、5 1 2 ビット)。ベクトルデータ 3 0 は、ベクトルデータのサンプルセット 3 4 に分割され得る。たとえば、ベクトルデータ 3 0 は、2 5 6 ビット幅であってよく、より小さいベクトルデータのサンプルセット 3 4 (Y) ~ 3 4 (0) を備えることができ、ここでベクトルデータのサンプルセット 3 4 (Y) ~ 3 4 (0) の一部は 1 6 ビット幅であり、ベクトルデータのサンプルセット 3 4 (Y) ~ 3 4 (0) のうちのそれ以外は 3 2 ビット幅である。V P E 2 2 は、高度の並列性を達成するように、V P E 2 2 に並列に与えられるいくつかの選ばれた乗算ベクトルデータのサンプルセット 3 4 (Y) ~ 3 4 (0) に対して、ベクトル処理を行うことが可能である。ベクトルレジスタ 2 8 は、V P E 2 2 がベクトルデータ 3 0 を処理するとき生成される結果を記憶するようにも構成される。いくつかの実施形態では、V P E 2 2 は、より速いベクトル命令実行時間を可能にするようにレジスタ書込みを低減するために、中間ベクトル処理結果をベクトルレジスタ 2 8 に記憶しないように構成される。この構成は、スカラー処理 D S P など、中間結果をレジスタに記憶するスカラー処理エンジンによって実行されるスカラー命令とは反対である。

【 0 0 2 4 】

[0038]図 2 のベースバンドプロセッサ 2 0 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、条件を V P E 2 2 に与えるように、およびアップデートされた条件をベクトル命令実行の結果として記憶するように構成された条件レジスタ 3 6 も含む。ベースバンドプロセッサ 2 0 はまた、累算レジスタ 3 8 と、グローバルレジスタ 4 0 と、アドレスレジスタ 4 2 とを含む。累算レジスタ 3 8 は、V P E 2 2 によって、累算された結果を、ベクトルデータ 3 0 に対していくつかの特殊な演算を実行した結果として記憶するのに使われるように構成される。グローバルレジスタ 4 0 は、V P E 2 2 によってサポートされるいくつかのベクトル命令用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 4 2 は、ベクトルユニットデータメモリ 3 2 からベクトルデータ 3 0 を取り出し、ベクトル処理結果をベクトルユニットデータメモリ 3 2 に記憶するために、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、V P E 2 2 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。

【0025】

[0039]引き続き図2を参照すると、本実施形態におけるベースバンドプロセッサ20は、VPE22によって行われるベクトル処理に加え、ベースバンドプロセッサ20中でスカラー処理を行うためのスカラープロセッサ44（「整数ユニット」とも呼ばれる）も含む。高効率な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CPUを設けることが所望され得る。本実施形態では、スカラープロセッサ44は、非限定的例として、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論理演算ユニット（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、フェッチした命令を復号し、フェッチした命令を、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49を通してVPE22に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によって使用するための汎用レジスタ52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によってアクセスするために、メインメモリから汎用レジスタ52にデータを与えるように、整数ユニットデータメモリ（DMEM）54がベースバンドプロセッサ20中に含まれる。DMEM54は、非限定的例として、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59を通る、メインメモリへのアクセスを求めるベクトル命令をスカラープロセッサ44が実行しているとき、汎用レジスタ52からメモリアドレスを受け取るように構成されたメモリコントローラレジスタ58を含むメモリコントローラ56も含む。

【0026】

[0040]図2のベースバンドプロセッサ20の例示的な構成要素について記載したので、共通回路要素およびハードウェアをもつ、基数 2^x のバタフライベクトル処理ならびに他の算術論理ベクトル演算を行うVPE22およびそのプログラマブルなデータバス構成に関するさらなる詳細についてここで論じる。後でより詳しく論じる図3～図8Cは、マルチモード基数 2^x のバタフライベクトル処理ならびに他の算術論理ベクトル演算を行うためのプログラマブルなデータバス構成を有する、図2のVPE22中に含まれる例示的なベクトル処理ブロックを含む。やはり後でより詳しく論じる図9～図11は、各周期の基数 2^x のバタフライ演算を実施するように構成された基数バタフライ回路を形成するために乗算器ブロックを例示的な累算器ブロックと融合するための、例示的なプログラムされたデータバス構成を提供する。

【0027】

[0041]この点において、図3は、図2のVPE22の例示的な概略図を示す。図3に示すように、および図4～図8Cにおいて以下でより詳細に説明するように、VPE22は、プログラマブルなデータバス構成で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処理段階60を含む。後でより詳しく論じるように、ベクトル処理ブロック中に設けられたプログラマブル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32から受け取られたベクトルデータ30に対する異なる特定のベクトル処理演算の実施をサポートするようにプログラムされ、再プログラムされる。たとえば、いくつかのベクトル処理演算は一般に、ベクトルデータ30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し得る。そのようなベクトル処理の非限定的例には、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（FFT）演算を実施するために一般に使われる、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のバタフライ演算があり、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行われる。やはり図7～図9Cに関連して後でより詳しく論じるように、図2のVPE22は、桁上げ保存累算器（carry-save accumulator）において冗長桁上げ（redundant carry-save）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もつ融合乗算器（fusing multiplier）というオプションも有する。桁上げ保存累算器中で冗長桁上げ保存フォーマットを提供することにより、桁上げ伝搬経路と、各累

算ステップ中の桁上げ伝搬加算演算 (carry propagation add operation) とを提供する必要をなくすることができる。

【 0 0 2 8 】

[0042] この点において、さらに図 3 を参照して、V P E 2 2 の M 0 乗算ベクトル処理段階 6 0 (1) について最初に記載する。M 0 乗算ベクトル処理段階 6 0 (1) は、プログラマブルなデータパス構成を各々が有する、任意の所望の数の乗算器ブロック 6 2 (A) ~ 6 2 (0) の形の複数のベクトル処理ブロックを含む第 2 のベクトル処理段階である。乗算器ブロック 6 2 (A) ~ 6 2 (0) は、V P E 2 2 中でベクトル乗算演算を実施するように設けられる。複数の乗算器ブロック 6 2 (A) ~ 6 2 (0) は、最大 1 2 個の乗算ベクトルデータのサンプルセット 3 4 (Y) ~ 3 4 (0) の乗算を行うために、M 0 乗算ベクトル処理段階 6 0 (1) 中に互いと並列に配置される。本実施形態では、「A」は 3 に等しく、この例では、M 0 乗算ベクトル処理段階 6 0 (1) に 4 つの乗算器ブロック 6 2 (3) ~ 6 2 (0) が含まれることを意味する。乗算ベクトルデータのサンプルセット 3 4 (Y) ~ 3 4 (0) は、V P E 2 2 中の第 1 のベクトル処理段階 6 0 (0) である入力読取り (R R) ベクトル処理段階に設けられた複数のラッチ 6 4 (Y) ~ 6 4 (0) へのベクトル処理のために V P E 2 2 中にロードされる。本実施形態では、V P E 2 2 中に 1 2 個のラッチ 6 4 (1 1) ~ 6 4 (0) があり、本実施形態では「Y」が 1 1 に等しいことを意味する。ラッチ 6 4 (1 1) ~ 6 4 (0) は、ベクトルレジスタ 2 8 (図 2 参照) から取り出された乗算ベクトルデータのサンプルセット 3 4 (1 1) ~ 3 4 (0) をベクトルデータ入力のサンプルセット 6 6 (1 1) ~ 6 6 (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 6 4 (1 1) ~ 6 4 (0) は 8 ビット幅である。ラッチ 6 4 (1 1) ~ 6 4 (0) は各々、乗算ベクトルデータ入力のサンプルセット 6 6 (1 1) ~ 6 6 (0) を、総計 9 6 ビット幅のベクトルデータ 3 0 (すなわち、各々、1 2 ラッチ × 8 ビット) を求めてラッチするようにそれぞれ構成される。

【 0 0 2 9 】

[0043] 引き続き図 3 を参照すると、複数の乗算器ブロック 6 2 (3) ~ 6 2 (0) は、ベクトル乗算演算を行うために、ベクトルデータ入力のサンプルセット 6 6 (1 1) ~ 6 6 (0) のいくつかの組合せを受け取ることが可能なように構成され、ここにおいて「Y」は、この例では 1 1 に等しい。乗算ベクトルデータ入力のサンプルセット 6 6 (1 1) ~ 6 6 (0) は、V P E 2 2 の設計に従って、複数の入力データパス A 3 ~ A 0、B 3 ~ B 0、および C 3 ~ C 0 中で与えられる。ベクトルデータ入力のサンプルセット 6 6 (3) ~ 6 6 (0) は、図 3 に示すように入力データパス C 3 ~ C 0 に対応する。ベクトルデータ入力のサンプルセット 6 6 (7) ~ 6 6 (4) は、図 3 に示すように入力データパス B 3 ~ B 0 に対応する。ベクトルデータ入力のサンプルセット 6 6 (1 1) ~ 6 6 (8) は、図 3 に示すように入力データパス A 3 ~ A 0 に対応する。複数の乗算器ブロック 6 2 (3) ~ 6 2 (0) は、ベクトル乗算演算を行うために、それぞれ、受け取られたベクトルデータ入力のサンプルセット 6 6 (1 1) ~ 6 6 (0) を、複数の乗算器ブロック 6 2 (3) ~ 6 2 (0) へと設けられた入力データパス A 3 ~ A 0、B 3 ~ B 0、C 3 ~ C 0 に従って処理するように構成される。

【 0 0 3 0 】

[0044] 図 4 および図 5 に関連して後でより詳しく論じるように、図 3 の乗算器ブロック 6 2 (3) ~ 6 2 (0) 中に設けられたプログラマブルな内部データパス 6 7 (3) ~ 6 7 (0) は、異なるデータパス構成を有するようにプログラムされ得る。これらの異なるデータパス構成は、各乗算器ブロック 6 2 (3) ~ 6 2 (0) へと設けられた特定の入力データパス A 3 ~ A 0、B 3 ~ B 0、C 3 ~ C 0 に従って乗算器ブロック 6 2 (3) ~ 6 2 (0) に与えられる特定の受け取られたベクトルデータ入力のサンプルセット 6 6 (1 1) ~ 6 6 (0) の異なる組合せおよび / または異なるビット長の乗算を可能にする。この点において、複数の乗算器ブロック 6 2 (3) ~ 6 2 (0) は、ベクトル乗算出力のサンプルセット 6 8 (3) ~ 6 8 (0) を、ベクトルデータ入力のサンプルセット 6 6 (1 1) ~ 6 6 (0) の特定の組合せを掛け合わせた乗算結果を備えるベクトル結果出力のサ

ンプルセットとして与える。

【0031】

[0045]たとえば、乗算器ブロック62(3)~62(0)のプログラマブルな内部データパス67(3)~67(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48中のベクトル命令デコーダから与えられる設定に従ってプログラムされ得る。本実施形態では、乗算器ブロック62(3)~62(0)の4つのプログラマブルな内部データパス67(3)~67(0)がある。ベクトル命令は、VPE22によって実施される特定のタイプの演算を指定する。したがって、VPE22は、同じ共通回路要素を用いて高効率で異なるタイプのベクトル乗算演算を行うために、乗算器ブロック62(3)~62(0)のプログラマブルな内部データパス67(3)~67(0)を構成するようにプログラムされ、再プログラムされ得る。たとえば、VPE22は、乗算器ブロック62(3)~62(0)のプログラマブルな内部データパス67(3)~67(0)を、命令ディスパッチ48中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、実行される各ベクトル命令のクロック周期ごとの周期ベースで構成し、再構成するようにプログラムされ得る。したがって、VPE22中のM0乗算ベクトル処理段階60(1)が、ベクトルデータ入力のサンプルセット66を毎クロック周期に処理するように構成されている場合、その結果、乗算器ブロック62(3)~62(0)は、命令ディスパッチ48中の命令パイプライン中のベクトル命令の復号に従って、毎クロック周期でベクトル乗算演算を実施する。

【0032】

[0046]乗算器ブロック62は、実数乗算と複素乗算とを実施するようにプログラムされ得る。引き続き図3を参照すると、あるベクトル処理ブロックデータパス構成において、乗算器ブロック62が、2つの8ビットベクトルデータ入力のサンプルセット66を掛け合わせるように構成され得る。1つの乗算ブロックデータパス構成では、乗算器ブロック62が、2つの16ビットベクトルデータ入力のサンプルセット66を掛け合わせるように構成され得、これらのセットは、8ビットベクトルデータ入力のサンプルセット66の第1のペアに、8ビットベクトルデータ入力のサンプルセット66の第2のペアを乗算して形成される。これについては、図6に示し、以下でより詳細に説明する。やはり、乗算器ブロック62(3)~62(0)中でプログラマブルなデータパス構成を提供することにより、乗算器ブロック62(3)~62(0)が、VPE22における面積を削減するとともに、可能な、所望のベクトル処理演算を実施するために、ベースバンドプロセッサ20中により少ないVPE22が設けられるようにするために、異なるタイプの乗算演算を実施するように構成され、再構成されることができ柔軟性がもたらされる。

【0033】

[0047]図3に参照を戻すと、複数の乗算器ブロック62(3)~62(0)は、プログラマブルな出力データパス70(3)~70(0)中のベクトル乗算出力のサンプルセット68(3)~68(0)を、次のベクトル処理段階60または出力処理段階のいずれかに与えるように構成される。ベクトル乗算出力のサンプルセット68(3)~68(0)は、複数の乗算器ブロック62(3)~62(0)によって実行されるベクトル命令に基づき、プログラムされた構成に従って、プログラマブルな出力データパス70(3)~70(0)中で与えられる。この例では、プログラマブルな出力データパス70(3)~70(0)中のベクトル乗算出力のサンプルセット68(3)~68(0)は、以下で論じるように、累算のためにM1累算ベクトル処理段階60(2)に与えられる。VPE22のこの特定の設計では、複数の乗算器ブロック62(3)~62(0)、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請する特殊なベクトル命令をサポートするための累算器を設けることが所望される。たとえば、FFT演算を行うのに一般に使われる、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。ただし、VPE22中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であり、限定的でないことに留意されたい。プログラマブルなデータパス構成を有するVPEは、ベクトル処理ブロックを有する1つまたは他のどの数のベクトル処理

段階を含むように構成されてもよい。ベクトル処理ブロックは、設計およびVPE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、どのタイプの演算を実施するように設けられてもよい。

【0034】

[0048]引き続き図3を参照すると、本実施形態では、ベクトル乗算出力のサンプルセット68(3)~68(0)は、M1累算ベクトル処理段階60(2)である次のベクトル処理段階中に設けられた複数の累算器ブロック72(3)~72(0)に与えられる。複数の累算器ブロック72(A)~72(0)の中の各累算器ブロックは、2つの累算器72(X)(1)および72(X)(0)(すなわち、72(3)(1)、72(3)(0)、72(2)(1)、72(2)(0)、72(1)(1)、72(1)(0)、および72(0)(1)、72(0)(0))を含む。複数の累算器ブロック72(3)~72(0)は、ベクトル乗算出力のサンプルセット68(3)~68(0)の結果を累算する。図7~図9Cに関連して後でより詳しく論じるように、複数の累算器ブロック72(3)~72(0)は桁上げ保存累算器として設けられることができ、ここにおいて桁上げ積は本質的に、累算演算が完了されるまで、累算プロセス中は保存され、伝搬されない。複数の累算器ブロック72(3)~72(0)は、複数の累算器ブロック72(3)~72(0)中で冗長桁上げ保存フォーマットを提供するために、図5および図6の複数の乗算器ブロック62(3)~62(0)と融合されるという選択肢も有する。複数の累算器ブロック72(3)~72(0)中で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72(3)~72(0)の各累算ステップ中に、桁上げ伝搬経路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累算ベクトル処理段階60(2)およびその複数の累算器ブロック72(3)~72(0)を、ここで図3を参照して提起する。

【0035】

[0049]図3を参照すると、M1累算ベクトル処理段階60(2)中の複数の累算器ブロック72(3)~72(0)は、累算器出力のサンプルセット76(3)~76(0)(すなわち、76(3)(1)、76(3)(0)、76(2)(1)、76(2)(0)、76(1)(1)、76(1)(0)、および76(0)(1)、76(0)(0))を次のベクトル処理段階60または出力処理段階のいずれかにおいて与えるために、プログラマブルな出力データパス74(3)~74(0)(すなわち、74(3)(1)、74(3)(0)、74(2)(1)、74(2)(0)、74(1)(1)、74(1)(0)、および74(0)(1)、74(0)(0))中のベクトル乗算出力のサンプルセット68(3)~68(0)を、プログラマブルな出力データパス構成に従って累算するように構成される。この例では、累算器出力のサンプルセット76(3)~76(0)は、ALU処理段階60(3)である出力処理段階に与えられる。たとえば、後でより詳しく論じるように、累算器出力のサンプルセット76(3)~76(0)は、非限定的例として、図2のベースバンドプロセッサ20中のスカラープロセッサ44中のALU46にも与えられることができる。たとえば、ALU46は、より一般的な処理演算において使われるべき、VPE22によって実行される特殊なベクトル命令に従って、累算器出力のサンプルセット76(3)~76(0)をとることができる。

【0036】

[0050]図3に参照を戻すと、累算器ブロック72(3)~72(0)のプログラマブルな入力データパス78(3)~78(0)および/またはプログラマブルな内部データパス80(3)~80(0)は、乗算器ブロック62(3)~62(0)から累算器ブロック72(3)~72(0)に与えられた、異なる組合せおよび/またはビット長のベクトル乗算出力のサンプルセット68(3)~68(0)を受け取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72は、2つの累算器72(X)(1)、72(X)(0)からなるので、プログラマブルな入力データパス78(A)~78(0)は、78(3)(1)、78(3)(0)、78(2)(1)、78(2)(0)、78(1)(1)、78(1)(0)、および78(0)(1)、78(0)(0)として図3

に示されている。同様に、プログラマブルな内部データバス 80 (3) ~ 80 (A) は、80 (3) (1)、80 (3) (0)、80 (2) (1)、80 (2) (0)、80 (1) (1)、80 (1) (0)、80 (0) (1)、80 (0) (0) として図 3 に示されている。累算器ブロック 72 (3) ~ 72 (0) 中にプログラマブルな入力データバス 78 (3) ~ 78 (0) および / またはプログラマブルな内部データバス 80 (3) ~ 80 (0) を設けることについては、図 8 ~ 図 9 C に関連して後でより詳しく論じる。このようにして、累算器ブロック 72 (3) ~ 72 (0) のプログラマブルな入力データバス 78 (3) ~ 78 (0) および / またはプログラマブルな内部データバス 80 (3) ~ 80 (0) に従って、累算器ブロック 72 (3) ~ 72 (0) は、累算されたベクトル乗算出力のサンプルセット 68 (3) ~ 68 (0) のプログラムされた組合せに従って、累算器出力のサンプルセット 76 (3) ~ 76 (0) を与えることができる。やはり、これにより、累算器ブロック 72 (3) ~ 72 (0) が、VPE 22 中での面積を削減し、可能性としては、所望のベクトル処理演算を実施するためのベースバンドプロセッサ 20 中に設けられる VPE をより少なくするために、プログラマブルな入力データバス 78 (3) ~ 78 (0) および / またはプログラマブルな内部データバス 80 (3) ~ 80 (0) のプログラミングに基づいて異なるタイプの累算演算を実施するように構成され、再構成され得るという柔軟性が与えられる。

【0037】

[0051] 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 72 のプログラマブルな入力データバス 78 および / またはプログラマブルな内部データバス 80 は、非限定的例として、単一 40 ビット累算器を提供するようにプログラムされ得る。これについては、図 9 A に示し、以下でより詳細に説明する。別の累算器モード構成では、2つの累算器ブロック 72 のプログラマブルな入力データバス 78 および / またはプログラマブルな内部データバス 80 は、非限定的例として、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。これについては、図 9 B に示し、以下でより詳細に説明する。別の累算器モード構成では、2つの累算器ブロック 72 のプログラマブルな入力データバス 78 および / またはプログラマブルな内部データバス 80 は、16 ビット桁上げ保存加算器、それに続く単一 24 ビット累算器を提供するようにプログラムされ得る。これについては、図 9 C に示し、以下でより詳細に説明する。乗算演算と累算演算の特定の異なる組合せも、乗算器ブロック 62 (3) ~ 62 (0) および累算器ブロック 72 (3) ~ 72 (0) (たとえば、16 ビット累算を用いる 16 ビット複素乗算、および 16 ビット累算を用いる 32 ビット複素乗算) のプログラミングに従って、VPE 22 によってサポートされ得る。

【0038】

[0052] 累算器ブロック 72 (3) ~ 72 (0) のプログラマブルな入力データバス 78 (3) ~ 78 (0) および / またはプログラマブルな内部データバス 80 (3) ~ 80 (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 48 中のベクトル命令デコードから与えられた設定に従ってプログラムされ得る。ベクトル命令は、VPE 22 によって実施されるべき特定タイプの演算を指定する。したがって、VPE 22 は、累算器ブロック 72 (3) ~ 72 (0) のプログラマブルな入力データバス 78 (3) ~ 78 (0) および / またはプログラマブルな内部データバス 80 (3) ~ 80 (0) を、命令ディスパッチ 48 中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各ベクトル命令用にプログラムし直すように構成され得る。ベクトル命令は、VPE 22 の 1 つまたは複数のクロック周期にわたって実行し得る。また、この例では、VPE 22 は、累算器ブロック 72 (3) ~ 72 (0) のプログラマブルな入力データバス 78 (3) ~ 78 (0) および / またはプログラマブルな内部データバス 80 (3) ~ 80 (0) を、ベクトル命令の各クロック周期ごとに、クロック周期単位でプログラムし直すように構成され得る。したがって、たとえば、VPE 22 中の M1 累算ベクトル処理段階 60 (2) によって実行されるベクトル命令が、ベクトル乗算出力のサンプルセット 68 (3) ~ 68 (0) を毎クロック周期に処理する場合、その結果、累算器ブロック 72 (3) ~ 7

10

20

30

40

50

2 (0) のプログラマブルな入力データパス 7 8 (3) ~ 7 8 (0) および / またはプログラマブルな内部データパス 8 0 (3) ~ 8 0 (0) は、ベクトル命令の実行中、各クロック周期に再構成され得る。図 4 A および図 4 B は、例示的なベクトル処理のさらなる説明を与えるための、図 2 および図 3 の V P E 2 2 中の乗算器ブロック 6 2 (A) ~ 6 2 (0) および累算器ブロック 7 2 (A) (1) ~ 7 2 (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。図 4 A は、図 2 および図 3 の例示的な V P E 中に含まれるプログラマブルなデータパス構成を有する、乗算器ブロック 6 2 (A) ~ 6 2 (0)、累算器ブロック 7 2 (A) (1) ~ 7 2 (0) (0)、または両方のいずれであってもよい一般化されたベクトル処理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である。図 4 B は、各々がプログラマブルなデータパス構成を有するとともに図 2 および図 3 の例示的な V P E 2 2 中の異なるベクトル処理段階中に設けられる、乗算器ブロック 6 2 (A) ~ 6 2 (0) および累算器ブロック 7 2 (A) (1) ~ 7 2 (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。

10

20

30

40

50

【 0 0 3 9 】

[0053] この点において、図 4 A に示すように、V P E 2 2 のプロセスは、入力処理段階 6 0 (0) 中の複数の入力データパスの中のある入力データパス (A 3 ~ C 0) 中で、ベクトルアレイの幅の複数のベクトルデータ入力のサンプルセット 3 4 (Y) ~ 3 4 (0) を受け取ることを含む (ブロック 8 1)。ベクトル処理は次に、ベクトル処理ブロック 6 2 (A) ~ 6 2 (0) および / または 7 2 (A) (1) ~ 7 2 (0) (0) 中の複数の入力データパス A 3 ~ C 0 から、ベクトルデータ入力のサンプルセット 3 4 (Y) ~ 3 4 (0) を受け取ることができる (ブロック 8 3)。ベクトル処理は次に、ベクトル処理段階 6 0 (1)、6 0 (2) によって実行されるベクトル命令に従って、ベクトル処理ブロック 6 2 (A) ~ 6 2 (0)、7 2 (A) (1) ~ 7 2 (0) (0) 用のプログラマブルなデータパス構成 6 7 (A) ~ 6 7 (0)、7 0 (3) ~ 7 0 (0)、7 8 (A) (1) ~ 7 8 (0) (0)、8 0 (A) (1) ~ 8 0 (0) (0)、7 4 (A) (1) ~ 7 4 (0) (0) に基づいて、ベクトル結果出力のサンプルセット 6 8 (A) ~ 6 8 (0)、7 6 (A) (1) ~ 7 6 (0) (0) を与えるように、ベクトルデータ入力のサンプルセット 3 4 (Y) ~ 3 4 (0) を処理することを含む (ブロック 8 5)。ベクトル処理は次に、出力データパス 7 0 (A) ~ 7 0 (0)、7 4 (A) (1) ~ 7 4 (0) (0) 中でベクトル結果出力のサンプルセット 6 8 (A) ~ 6 8 (0)、7 6 (A) (1) ~ 7 6 (0) (0) を与えることを含む (ブロック 8 7)。ベクトル処理は次に、出力処理段階 6 0 (3) において、ベクトル処理段階 6 0 (1)、6 0 (2) からベクトル結果出力のサンプルセット 6 8 (A) ~ 6 8 (0)、7 6 (A) (1) ~ 7 6 (0) (0) を受け取ることを含む (ブロック 8 9)。

【 0 0 4 0 】

[0054] 図 4 A に関連して上述したベクトル処理における各処理段階 6 0 (0) ~ 6 0 (3) は、並列化ベクトル処理のために同時に起こり、ここにおいてベクトル処理ブロック 6 2 (A) ~ 6 2 (0)、7 2 (A) (1) ~ 7 2 (0) (0) のプログラマブルなデータパス構成 6 7 (A) ~ 6 7 (0)、7 0 (3) ~ 7 0 (0)、7 8 (A) (1) ~ 7 8 (0) (0)、8 0 (A) (1) ~ 8 0 (0) (0)、7 4 (A) (1) ~ 7 4 (0) (0) は、各クロック周期と同程度に頻繁に再プログラムされ得ることに留意されたい。上述したように、これにより、ベクトル処理ブロック 6 2 (A) ~ 6 2 (0)、7 2 (A) (1) ~ 7 2 (0) (0) は、異なるベクトル命令用の異なる演算を、効率的に、および共通ベクトル処理ブロック 6 2 (A) ~ 6 2 (0)、7 2 (A) (1) ~ 7 2 (0) (0) の使用により実施することができるようになる。

【 0 0 4 1 】

[0055] 図 4 B は、乗算演算、それに続く累算演算を伴う、ベクトル命令のための、図 3 の V P E 2 2 中の乗算器ブロック 6 2 (A) ~ 6 2 (0) および累算器ブロック 7 2 (A) (1) ~ 7 2 (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。たとえば、F F T ベクトル演算は、乗算演算と、それに続く累算演算を伴う。図 4 B のフロー

チャートは、図 4 A で上述した、V P E 2 2 の例示的な一般化されたベクトル処理のさらなる例示的な詳細を与える。この点において、ベクトル処理は、入力処理段階 6 0 (0) 中の複数の入力データパスの中のある入力データパス (A 3 ~ C 0) 中で、ベクトルアレイの幅の複数のベクトルデータ入力のサンプルセット 3 4 (Y) ~ 3 4 (0) を受け取ることを含む (ブロック 9 3) 。ベクトル処理は次いで、複数の乗算器ブロック 6 2 (A) ~ 6 2 (0) 中の複数の入力データパス A 3 ~ C 0 からベクトルデータ入力のサンプルセット 3 4 (Y) ~ 3 4 (0) を受け取ることを含む (ブロック 9 5) 。ベクトル処理は次いで、ベクトル処理段階 6 0 (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 6 2 (A) ~ 6 2 (0) 用のプログラマブルなデータパス構成 6 7 (A) ~ 6 7 (0) 、 7 0 (3) ~ 7 0 (0) に基づき、複数の乗算出力データパス 7 0 (A) ~ 7 0 (0) 中の乗算出力データパス 7 0 (A) ~ 7 0 (0) 中で乗算ベクトル結果出力のサンプルセット 6 8 (A) ~ 6 8 (0) を与えるように、ベクトルデータ入力のサンプルセット 3 4 (Y) ~ 3 4 (0) を乗算することを含む (ブロック 9 7) 。ベクトル処理は次に、複数の累算器ブロック 7 2 (A) (1) ~ 7 2 (0) (0) 中の複数の乗算出力データパス 7 0 (A) ~ 7 0 (0) から乗算ベクトル結果出力のサンプルセット 6 8 (A) ~ 6 8 (0) を受け取ることを含む (ブロック 9 9) 。ベクトル処理は次に、第 2 のベクトル処理段階 6 0 (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 7 2 (A) (1) ~ 7 2 (0) (0) 用のプログラマブルなデータパス 7 8 (A) (1) ~ 7 8 (0) (0) 、 8 0 (A) (1) ~ 8 0 (0) (0) 、 7 4 (A) (1) ~ 7 4 (0) (0) 構成に基づき、ベクトルの累算された結果のサンプルセット 7 6 (A) (1) ~ 7 6 (0) (0) を与えるように、乗算ベクトル結果出力のサンプルセット 6 8 (A) ~ 6 8 (0) を一緒に累算することを含む (ブロック 1 0 1) 。ベクトル処理は次いで、出力データパス 7 4 (A) (1) ~ 7 4 (0) (0) 中でベクトルの累算された結果のサンプルセット 7 6 (A) (1) ~ 7 6 (0) (0) を与えることを含む (ブロック 1 0 3) 。ベクトル処理は次いで、出力処理段階 6 0 (3) 中の累算器ブロック 7 2 (A) (1) ~ 7 2 (0) (0) からベクトル結果出力のサンプルセット 7 6 (A) (1) ~ 7 6 (0) (0) を受け取ることを含む (ブロック 1 0 5) 。

【 0 0 4 2 】

[0056] プログラマブルなデータパス構成を有するベクトル処理ブロックを利用する、図 3 の例示的な V P E 2 2 および図 4 A および図 4 B のベクトル処理の概観について記載したので、記述の残りでは、図 5 ~ 図 9 C において、これらのベクトル処理ブロックのより例示的な、非限定的詳細について記載する。

【 0 0 4 3 】

[0057] この点において、図 5 は、図 3 の V P E 2 2 の M 0 乗算ベクトル処理段階 6 0 (1) 中の複数の乗算器ブロック 6 2 (3) ~ 6 2 (0) のより詳細な概略図である。図 6 は、図 5 の乗算器ブロック 6 2 の内部構成要素の概略図である。図 5 に示すように、特定の入力データパス A 3 ~ A 0 、 B 3 ~ B 0 、 C 3 ~ C 0 に従って乗算器ブロック 6 2 (3) ~ 6 2 (0) によって受け取られるベクトルデータ入力のサンプルセット 6 6 (1 1) ~ 6 6 (0) が示されている。図 6 に関連して後でより詳しく論じるように、この例における乗算器ブロック 6 2 (3) ~ 6 2 (0) の各々は、4 つの 8 ビット × 8 ビット乗算器を含む。図 5 に参照を戻すと、この例における乗算器ブロック 6 2 (3) ~ 6 2 (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 」のいずれかで乗算するように構成される。乗算器ブロック 6 2 中で掛け合わせられ得る被乗数入力「 A 」、および「 B 」または「 C 」は、図 5 に示すように、どの入力データパス A 3 ~ A 0 、 B 3 ~ B 0 、 C 3 ~ C 0 が乗算器ブロック 6 2 (3) ~ 6 2 (0) に接続されるかによって制御される。被乗数セレクト入力 8 2 (3) ~ 8 2 (0) が、被乗数入力「 B 」それとも被乗数入力「 C 」が被乗数入力「 A 」で乗算されるために選択されるかを選択するために、各乗算器ブロック 6 2 (3) ~ 6 2 (0) 中のプログラマブルな内部データパス 6 7 (3) ~ 6 7 (0) を制御するための入力として、各乗算器ブロック 6 2 (3) ~ 6 2 (0) に与えられる。このようにして、乗算器ブロック 6 2 (3) ~ 6 2 (0) は、それらの

プログラマブルな内部データパス 67(3) ~ 67(0) が、必要に応じて異なる乗算演算を行うように再プログラムされるという能力を備えている。

【0044】

[0058]引き続き図5を参照すると、乗算器ブロック62(3)を一例として使うと、入力データパスA3およびA2が、それぞれ入力AHおよびALに接続される。入力AHは、被乗数入力Aの上位ビットを表し、ALは、入力被乗数入力「A」の下位ビットを意味する。入力データパスB3およびB2は、それぞれ入力BHおよびBLに接続される。入力BHは、被乗数入力「B」の上位ビットを表し、ALは、入力被乗数入力「B」の下位ビットを表す。入力データパスC3およびC2は、それぞれ入力CIおよびCQに接続される。入力CIは、この例では入力被乗数入力「C」の実数ビット部分を表す。CQは、この例では入力被乗数入力「C」の虚数ビット部分を表す。図6に関連して後でより詳しく論じるように、被乗数セクタ入力82(3)は、この例では、乗算器ブロック62(3)のプログラマブルな内部データパス67(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」それとも被乗数入力「C」で実施するように構成されるか、または乗算器ブロック62(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」それとも被乗数入力「C」で実施するように構成されるかも制御する。

10

【0045】

[0059]引き続き図5を参照すると、乗算器ブロック62(3) ~ 62(0)は、それらのプログラマブルな内部データパス67(3) ~ 67(0)の構成に基づいて、乗算演算の桁上げ「C」および和「S」ベクトル出力のサンプルセットとしてベクトル乗算出力のサンプルセット68(3) ~ 68(0)を各々が生成するように構成される。図7 ~ 図9Cに関連して後でより詳しく論じるように、ベクトル乗算出力のサンプルセット68(3) ~ 68(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72(3) ~ 72(0)中で冗長桁上げ保存フォーマットを提供するように、冗長桁上げ保存フォーマットで複数の累算器72(3) ~ 72(0)に与えられることを意味する。後でより詳しく論じるように、複数の累算器72(3) ~ 72(0)中で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72(3) ~ 72(0)によって実施される累算演算中に桁上げ伝搬経路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。

20

【0046】

[0060]ベクトル乗算出力のサンプルセット68(3) ~ 68(0)を、それらのプログラマブルな内部データパス67(3) ~ 67(0)の構成に基づいて乗算演算の桁上げ「C」および和「S」ベクトル出力のサンプルセットとして生成する乗算器ブロック62(3) ~ 62(0)の例を、図5に示す。たとえば、乗算器ブロック62(3)は、桁上げC00と和S00とを8ビット乗算用の32ビット値として、桁上げC01と和S01とを16ビット乗算用の64ビット値として生成するように構成される。他の乗算器ブロック62(2) ~ 62(0)も、この例では同じ能力を有する。この点において、乗算器ブロック62(2)は、桁上げC10と和S10とを8ビット乗算用の32ビット値として、桁上げC11と和S11とを16ビット乗算用の64ビット値として生成するように構成される。乗算器ブロック62(1)は、桁上げC20と和S20とを8ビット乗算用の32ビット値として、桁上げC21と、和S21とを16ビット乗算用の64ビット値として生成するように構成される。乗算器ブロック62(0)は、桁上げC30と和S30とを8ビット乗算用の32ビット値として、桁上げC31と和S31とを16ビット乗算用の64ビット値として生成するように構成される。

30

40

【0047】

[0061]図5の乗算器ブロック62中に設けられるプログラマブルなデータパス構成のより例示的な詳細を説明するために、図6が挙げられる。図6は、8ビット×8ビットベクトルデータ入力のサンプルセット66と、16ビット×16ビットベクトルデータ入力のサンプルセット66とを乗算することが可能なプログラマブルなデータパス構成を有する、図3および図5の乗算器ブロック62の内部構成要素の概略図である。この点において

50

乗算器ブロック 62 は、この例では 4 つの 8×8 ビット乗算器 84 (3) ~ 84 (0) を含む。任意の所望の数の乗算器 84 が設けられてよい。第 1 の乗算器 84 (3) が、8 ビットベクトルデータ入力のサンプルセット 66 A [H] (入力被乗数入力「A」の上位ビットである) を受け取り、ベクトルデータ入力のサンプルセット 66 A [H] を、8 ビットベクトルデータ入力のサンプルセット 66 B [H] (入力被乗数入力「B」の上位ビットである) または 8 ビットベクトルデータ入力のサンプルセット 66 C [I] (入力被乗数入力「C」の上位ビットである) と乗算するように構成される。被乗数として乗算器 84 (3) に与えている 8 ビットベクトルデータ入力のサンプルセット 66 B [H] または 8 ビットベクトルデータ入力のサンプルセット 66 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 86 (3) が設けられる。マルチプレクサ 86 (3) は、本実施形態では、乗数セクタ入力 82 中の上位ビットである被乗数セクタビット入力 82 [3] によって制御される。このようにして、マルチプレクサ 86 (3) および被乗数セクタビット入力 82 [3] は、8 ビットベクトルデータ入力のサンプルセット 66 B [H] それとも 8 ビットベクトルデータ入力のサンプルセット 66 C [I] が、受け取るベクトルデータ入力のサンプルセット 66 A [H] と乗算されるかを乗算器 84 (3) が制御するためのプログラマブルな内部データパス 67 [0] 構成を提供する。

【0048】

[0062] 引き続き図 6 を参照すると、他の乗算器 84 (2) ~ 84 (0) も、第 1 の乗算器 84 (3) 用に設けられるものと同様のプログラマブルな内部データパス 67 [2] ~ 67 [0] を含む。乗算器 84 (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のサンプルセット 66 A [L] と乗算されるべき、8 ビットベクトルデータ入力のサンプルセット 66 B [H] または 8 ビットベクトルデータ入力のサンプルセット 66 C [I] のいずれかをプログラマブルな内部データパス 67 [1] 中で与えるための、プログラマブルな構成を有するプログラマブルな内部データパス 67 [2] を含む。選択は、本実施形態では、マルチプレクサ 86 (2) によって、乗数セクタ入力 82 中の被乗数セクタビット入力 82 [2] に従って制御される。乗算器 84 (1) は、8 ビットベクトルデータ入力のサンプルセット 66 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のサンプルセット 66 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のサンプルセット 66 C [Q] のいずれかをプログラマブルな内部データパス 67 [1] 中で与えるようにプログラマブルな内部データパス 67 [1] を含む。選択は、本実施形態では、マルチプレクサ 86 (1) によって、乗数セクタ入力 82 中の被乗数セクタビット入力 82 [1] に従って制御される。さらに、乗算器 84 (0) は、8 ビットベクトルデータ入力のサンプルセット 66 A [L] と乗算されるべき、8 ビットベクトルデータ入力のサンプルセット 66 B [L] または 8 ビットベクトルデータ入力のサンプルセット 66 C [Q] のいずれかをプログラマブルな内部データパス 67 [0] 中で与えるようにプログラマブルな内部データパス 67 [0] を含む。選択は、本実施形態では、マルチプレクサ 86 (0) によって、乗数セクタ入力 82 中の被乗数セクタビット入力 82 [0] に従って制御される。

【0049】

[0063] 引き続き図 6 を参照すると、上述したように、乗算器 84 (3) ~ 84 (0) は、異なるビット長乗算演算を実施するように構成され得る。この点において、各乗算器 84 (3) ~ 84 (0) は、それぞれビット長乗算モード入力 88 (3) ~ 88 (0) を含む。この例では、各乗算器 84 (3) ~ 84 (0) は、それぞれ、プログラマブルなデータパス 90 (3) ~ 90 (0)、91、および 92 (3) ~ 92 (0) の構成を制御する入力に従って、8 ビット \times 8 ビットモードでプログラムされ得る。各乗算器 84 (3) ~ 84 (0) は、それぞれ、プログラマブルなデータパス 90 (3) ~ 90 (0)、91、および 92 (3) ~ 92 (0) の構成を制御する入力に従って、16 ビット \times 16 ビットモードと 24 ビット \times 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行うようにもプログラムされ得る。たとえば、各乗算器 84 (3) ~ 84 (0) が、プログラ

10

20

30

40

50

マブルなデータパス 90 (3) ~ 90 (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 84 (3) ~ 84 (0) は、2 つの個々の 8 ビット × 8 ビット乗算器を、乗算器ブロック 62 の一部として備えるように構成され得る。各乗算器 84 (3) ~ 84 (0) が、プログラマブルなデータパス 91 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 84 (3) ~ 84 (0) は、単一 16 ビット × 16 ビット乗算器を乗算器ブロック 62 の一部として備えるように構成され得る。乗算器 84 (3) ~ 84 (0) が、プログラマブルなデータパス 92 (3) ~ 92 (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 84 (3) ~ 84 (0) は、1 つの 16 ビット × 24 ビット × 8 ビット乗算器を乗算器ブロック 62 の一部として備えるように構成され得る。

10

【0050】

[0064] 引き続き図 6 を参照すると、この例における乗算器 84 (3) ~ 84 (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 ビット入力 94 (3)、94 (2) および入力桁上げ 96 (3)、96 (2) が、それぞれ、各乗算器 84 (3)、84 (2) によって生成される。16 ビット入力 94 (1)、94 (0) および入力桁上げ 96 (1)、96 (0) が、それぞれ、各乗算器 84 (1)、84 (0) によって生成される。16 ビット入力 94 (3)、94 (2) および入力桁上げ 96 (3)、96 (2) はまた、入力 94 (3) ~ 94 (0) と入力桁上げ 96 (3) ~ 96 (0) を加え合わせるように、16 ビット和の入力 94 (1)、94 (0) および入力桁上げ 96 (1)、96 (0) とともに 24 ビットの 4 : 2 圧縮器 109 に与えられる。加算された入力 94 (3) ~ 94 (0) および入力桁上げ 96 (3) ~ 96 (0) は、プログラマブルなデータパス 91 がアクティブであるとともに入力 94 (3) ~ 94 (0) および入力桁上げ 96 (3) ~ 96 (0) とゲートされるとき、16 ビット × 16 ビット乗算モードで、単一和 98 と単一桁上げ 100 とをもたらす。プログラマブルなデータパス 91 は、24 ビットの 4 : 2 圧縮器 109 に与えられるように、16 ビットワードとして、組み合わせされた入力 94 (3)、94 (2) をもつ第 1 の AND ベースのゲート 102 (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 96 (3)、96 (2) をもつ第 2 の AND ベースのゲート 102 (2) によってゲートされる。プログラマブルなデータパス 91 はまた、24 ビットの 4 : 2 圧縮器 109 に与えられるように、16 ビットワードとして、組み合わせされた入力 94 (1)、94 (0) をもつ第 3 の AND ベースのゲート 102 (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 96 (1)、96 (0) をもつ第 4 の AND ベースのゲート 102 (2) によってゲートされる。乗算器ブロック 62 が 16 ビット × 16 ビットまたは 24 ビット × 8 ビット乗算モードで構成される場合、プログラマブルな出力データパス 70 [0] には、ベクトル乗算出力のサンプルセット 68 [0] が、圧縮された 32 ビット和 S0 および 32 ビット桁上げ C0 部分積として与えられる。

20

30

【0051】

[0065] 乗算器ブロック 62 中の乗算器 84 (3) ~ 84 (0) が 8 ビット × 8 ビット乗算モードで構成される場合、プログラマブルな出力データパス 70 [1] 構成は、圧縮なしで、16 ビット入力 94 (3) ~ 94 (0) および部分積としての対応する 16 ビット入力桁上げ 96 (3) ~ 96 (0) として提供される。乗算器ブロック 62 中の乗算器 84 (3) ~ 84 (0) が 8 ビット × 8 ビット乗算モードで構成される場合、プログラマブルな出力データパス 70 [1] は、圧縮なしで、16 ビット入力 94 (3) ~ 94 (0) およびベクトル乗算出力のサンプルセット 68 [1] としての対応する 16 ビット入力桁上げ 96 (3) ~ 96 (0) として提供される。乗算器ブロック 62 の乗算モードに依存するベクトル乗算出力のサンプルセット 68 [0]、68 [1] は、実行され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 72 (3) ~ 72 (0) に与えられる。

40

【0052】

50

[0066] プログラマブルなデータパス構成を有する、図 5 および図 6 の乗算器ブロック 6 2 (3) ~ 6 2 (0) について記載したので、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 7 2 (3) ~ 7 2 (0) と融合されるべき、V P E 2 2 中の乗算器ブロック 6 2 (3) ~ 6 2 (0) の特徴について、図 7 に関連してここで概説する。

【 0 0 5 3 】

[0067] この点について、図 7 は、図 2 および図 3 の V P E 中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、ここにおいて、累算器ブロックは、桁上げ伝搬を低減するの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。上述し、図 7 に示したように、乗算器ブロック 6 2 は、被乗数入力 6 6 [H] および 6 6 [L] を乗算し、少なくとも 1 つの入力和 9 4 と少なくとも 1 つの入力桁上げ 9 6 を、ベクトル乗算出力のサンプルセット 6 8 としてプログラマブルな出力データパス 7 0 中で与えるように構成される。各累算ステップ用の累算器ブロック 7 2 中に桁上げ伝搬経路と桁上げ伝搬加算器とを設ける必要をなくするために、プログラマブルな出力データパス 7 0 中のベクトル乗算出力のサンプルセット 6 8 中の少なくとも 1 つの入力和 9 4 および少なくとも 1 つの入力桁上げ 9 6 が、少なくとも 1 つの累算器ブロック 7 2 への冗長桁上げ保存フォーマットで融合される。言い換えると、ベクトル乗算出力のサンプルセット 6 8 中の桁上げ 9 6 が、ベクトル入力桁上げ 9 6 として桁上げ保存フォーマットで累算器ブロック 7 2 に与えられる。このようにして、ベクトル乗算出力のサンプルセット 6 8 中の入力和 9 4 および入力桁上げ 9 6 は、本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 7 2 の圧縮器 1 0 8 に与えられ得る。圧縮器 1 0 8 は、入力和 9 4 および入力桁上げ 9 6 を、それぞれ、以前の累算ベクトル出力和 1 1 2 および以前のシフトされた累算ベクトル出力桁上げ 1 1 7 と一緒に累算するように構成される。以前のシフトされた累算ベクトル出力桁上げ 1 1 7 は本質的に、累算演算中の保存された桁上げ累算である。

【 0 0 5 4 】

[0068] このようにして、単一最終桁上げ伝搬加算器のみが、受け取られた入力桁上げ 9 6 を、累算器ブロック 7 2 によって生成された累算の一部として、入力和 9 4 に伝搬するために、累算器ブロック 7 2 中に設けられることを求められるのではない。累算器ブロック 7 2 中の各累算ステップ中に桁上げ伝搬加算演算を実施することに関連した電力消費が、本実施形態では低減される。また、累算器ブロック 7 2 中の各累算ステップ中に桁上げ伝搬加算演算を実施することに関連したゲート遅延も、本実施形態ではなくなる。

【 0 0 5 5 】

[0069] 引き続き図 7 を参照すると、圧縮器 1 0 8 は、冗長な形での入力和 9 4 および入力桁上げ 9 6 を、それぞれ、以前の累算ベクトル出力和 1 1 2 および以前のシフトされた累算ベクトル出力桁上げ 1 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 1 1 7 は、次の受け取られた入力和 9 4 および入力桁上げ 9 6 の次の累算が圧縮器 1 0 8 によって実施される前に、累算ベクトル出力桁上げ 1 1 4 をシフトすることによって、圧縮器 1 0 8 ビットによって生成された累算ベクトル出力桁上げ 1 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 1 1 7 は、累算器ブロック 7 2 に設けられた単一最終桁上げ伝搬加算器 1 1 9 によって最終累算ベクトル出力和 1 1 2 に加算され、最終的なシフトされた累算ベクトル出力桁上げ 1 1 7 を、最終累算ベクトル出力和 1 1 2 を最終累算器出力のサンプルセット 7 6 2 の補数表現に変換するために桁上げ累算を伝搬する。最終累算ベクトル出力和 1 1 2 は、プログラマブルな出力データパス 7 4 中で累算器出力のサンプルセット 7 6 として与えられる (図 3 参照) 。

【 0 0 5 6 】

[0070] 乗算器ブロック 6 2 と、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 7 2 の融合を示す図 7 について説明したので、累算器ブロック 7 2 (3) ~ 7 2 (0) に関する、より例示的な詳細について、図 8 に関連してここで概説する。以下で説明する図 9 A ~ 図 9 C は、共通回路要素およびハードウェアを用いて異なるベクトル累算演算を行うための異なる累算モードにおける、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 7 2 (3) ~ 7 2 (0) のより例示的な詳細を与える。

【 0 0 5 7 】

[0071] 図 8 は、図 3 の V P E 2 2 中に設けられる累算器ブロック 7 2 の例示的な内部構成要素の詳細な概略図である。上述するとともに後でより詳しく論じるように、累算器ブロック 7 2 は、プログラマブルな入力データバス 7 8 (3) ~ 7 8 (0) および / またはプログラマブルな内部データバス 8 0 (3) ~ 8 0 (0) を有して構成され、そうすることによって累算器ブロック 7 2 は、特定の異なるタイプのベクトル累算演算を実施するように設計された専用回路要素として作用するようにプログラムされ得る。たとえば、累算器ブロック 7 2 は、符号付きおよび符号なし累算演算を含む、いくつかの異なる累算および加算を行うようにプログラムされ得る。異なるタイプの累算演算を行うように構成された累算器ブロック 7 2 中のプログラマブルな入力データバス 7 8 (3) ~ 7 8 (0) および / またはプログラマブルな内部データバス 8 0 (3) ~ 8 0 (0) の具体例が、後で論じる図 9 A ~ 図 9 C に示される。また、累算器ブロック 7 2 は、低減された組合せ論理で高速累算演算を行うために、桁上げ伝搬を回避または低減するために、冗長桁上げ算術を行うための桁上げ保存累算器 7 2 [0]、7 2 [1] を含むように構成される。

10

【 0 0 5 8 】

[0072] 累算器ブロック 7 2 の例示的な内部構成要素が、図 8 に示される。そこに示されているように、本実施形態における累算器ブロック 7 2 は、第 1 の入力 9 4 [0] および第 1 の入力桁上げ 9 6 [0] と、第 2 の入力 9 4 [1] および第 2 の入力桁上げ 9 6 [1] とを、一緒に累算するために乗算器ブロック 6 2 から受け取るように構成される。図 8 に関連して、入力 9 4 [0]、9 4 [1] および入力桁上げ 9 6 [0]、9 6 [1] は、ベクトル入力 9 4 [0]、9 4 [1] およびベクトル入力桁上げ 9 6 [0]、9 6 [1] と呼ばれる。上で記載するとともに図 6 に示すように、本実施形態におけるベクトル入力 9 4 [0]、9 4 [1] およびベクトル入力桁上げ 9 6 [0]、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 7 2 は、2 つの 2 4 ビット桁上げ保存累算器 7 2 [0]、7 2 [1] として設けられ、各々が、共通要素番号をもつ同様の構成要素を含み、「 [0] 」は桁上げ保存累算器 7 2 [0] 用に指定され、「 [1] 」は桁上げ保存累算器 7 2 [1] 用に指定される。桁上げ保存累算器 7 2 [0]、7 2 [1] は、ベクトル累算演算を同時に実施するように構成され得る。

20

【 0 0 5 9 】

[0073] 図 8 の桁上げ保存累算器 7 2 [0] を参照すると、ベクトル入力 9 4 [0] およびベクトル入力桁上げ 9 6 [0] は、プログラマブルな内部データバス 8 0 [0] の一部として設けられたマルチプレクサ 1 0 4 (0) 中の入力である。負のベクトル入力 9 4 [0] ' と負のベクトル入力桁上げ 9 6 [0] ' とを必要とする累算演算のための、マルチプレクサ 1 0 4 (0) への入力としての入力 1 0 7 (0) に従って負のベクトル入力 9 4 [0] ' と負のベクトル入力桁上げ 9 6 [0] ' とを生成する、排他的 O R ベースのゲートからなり得る否定回路 1 0 6 (0) も設けられる。マルチプレクサ 1 0 4 (0) は、ベクトル命令復号の結果として生成されたセクタ入力 1 1 0 (0) に従って、圧縮器 1 0 8 (0) に与えられるべきベクトル入力 9 4 [0] およびベクトル入力桁上げ 9 6 [0] または負のベクトル入力 9 4 [0] ' および負のベクトル入力桁上げ 9 6 [0] ' のいずれかを選択するように構成される。この点において、セクタ入力 1 1 0 (0) は、累算器ブロック 7 2 によって実施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 7 2 [0] のプログラマブルな入力データバス 7 8 [0] を、ベクトル入力 9 4 [0] およびベクトル入力桁上げ 9 6 [0]、または負のベクトル入力 9 4 [0] ' および負のベクトル入力桁上げ 9 6 [0] ' のいずれかを、圧縮器 1 0 8 (0) に与えるようにプログラマブルにさせる。

30

40

【 0 0 6 0 】

[0074] 引き続き図 8 を参照すると、本実施形態における桁上げ保存累算器 7 2 [0] の圧縮器 1 0 8 (0) は、複合ゲート 4 : 2 圧縮器である。この点において、圧縮器 1 0 8 (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 1 0 8 (0) は、現在のベクトル入力 9 4 [0] およびベクトル入力桁上げ 9 6 [0]

50

」、または現在の負のベクトル入力 $94[0]$ および負のベクトル入力桁上げ $96[0]$ を、圧縮器 $108(0)$ への4つの入力として、以前の累算されたベクトル入力 $94[0]$ およびベクトル入力桁上げ $96[0]$ 、または累算された負のベクトル入力 $94[0]$ および負のベクトル入力桁上げ $96[0]$ と一緒に累算するように構成される。圧縮器 $108(0)$ は、累算ベクトル出力 $112(0)$ と、累算ベクトル出力桁上げ $114(0)$ とを、累算器出力のサンプルセット $76(3) \sim 76(0)$ を与えるように、プログラマブルな出力データバス $74[0]$ 中の累算器出力のサンプルセット $76[0]$ として与える(図3参照)。累算ベクトル出力桁上げ $114(0)$ は、各累算ステップ中にビット幅成長を制御するための、シフトされた累算ベクトル出力桁上げ $117(0)$ を与えるように、累算演算中にビットシフト $116(0)$ によってシフトされる。たとえば、本実施形態におけるビットシフト $116(0)$ は、冗長桁上げ保存フォーマットでの、圧縮器 $108(0)$ に融合されるパレルシフトである。このようにして、シフトされた累算ベクトル出力桁上げ $117(0)$ は本質的に、累算器 $72[0]$ によって実施される累算演算中に、累算ベクトル出力 $112(0)$ に伝搬される必要なく、保存される。このようにして、累算器 $72[0]$ 中の各累算ステップ中に桁上げ伝搬加算演算を実施することに関連した電力消費およびゲート遅延が、本実施形態ではなくなる。

【0061】

[0075]追加の後続ベクトル入力 $94[0]$ およびベクトル入力桁上げ $96[0]$ 、または負のベクトル入力 $94[0]$ および負のベクトル入力桁上げ $96[0]$ は、現在の累算ベクトル出力 $112(0)$ および現在の累算ベクトル出力桁上げ $117(0)$ と累算され得る。ベクトル入力 $94[0]$ およびベクトル入力桁上げ $96[0]$ 、または負のベクトル入力 $94[0]$ および負のベクトル入力桁上げ $96[0]$ は、ベクトル命令復号の結果として生成された和桁上げセクタ $120(0)$ に従って、プログラマブルな内部データバス $80[0]$ の一部としてのマルチプレクサ $118(0)$ によって選択される。現在の累算ベクトル出力 $112(0)$ および現在のシフトされた累算ベクトル出力桁上げ $117(0)$ は、桁上げ保存累算器 $72[0]$ が、アップデートされた累算ベクトル出力 $112(0)$ と、累算ベクトル出力桁上げ $114(0)$ とを与えるために、入力として圧縮器 $108(0)$ に与えられ得る。この点において、和桁上げセクタ $120(0)$ は、累算器 $72[0]$ のプログラマブルな内部データバス $80[0]$ を、累算器ブロック 72 によって実施されるように構成された累算演算に従って、ベクトル入力 $94[0]$ と、ベクトル入力桁上げ $96[0]$ とを圧縮器 $108(0)$ に与えるようにプログラマブルにさせる。マルチプレクサ $118(0)$ に、桁上げ保存累算器 $72[0]$ における累算の動作タイミングを制御するための保持状態入力 $126(0)$ に従って、累算ベクトル出力 $112(0)$ およびシフトされた累算ベクトル出力桁上げ $117(0)$ の現在の状態を保持させるために、保持ゲート $122(0)$ 、 $124(0)$ も本実施形態では設けられる。

【0062】

[0076]引き続き図8を参照すると、桁上げ保存累算器 $72[0]$ の累算ベクトル出力 $112(0)$ およびシフトされた累算ベクトル出力桁上げ $117(0)$ 、ならびに桁上げ保存累算器 $72[1]$ の累算ベクトル出力 $112(1)$ およびシフトされた累算ベクトル出力桁上げ $117(1)$ 、ならびには、それぞれ、制御ゲート $134(0)$ 、 $136(0)$ および $134(1)$ 、 $136(1)$ によってゲートされる。制御ゲート $134(0)$ 、 $136(0)$ および $134(1)$ 、 $136(1)$ は、それぞれ、圧縮器 $108(0)$ 、 $108(1)$ に戻される、累算ベクトル出力 $112(0)$ およびシフトされた累算ベクトル出力桁上げ $117(0)$ と、累算ベクトル出力 $112(1)$ およびシフトされた累算ベクトル出力桁上げ $117(1)$ とを制御する。

【0063】

[0077]要約すれば、図8の累算器ブロック 72 の累算器 $72[0]$ 、 $72[1]$ のプログラマブルな入力データバス $78[0]$ 、 $78[1]$ およびプログラマブルな内部データバス $80[0]$ 、 $80[1]$ によって、累算器ブロック 72 は異なるモードで構成され得

る。累算器ブロック 72 は、図 8 に示す共通累算器回路要素を用いた特定のベクトル処理命令に従って、異なる累算演算を行うように構成され得る。共通累算器回路要素を用いた特定のベクトル処理命令に従って異なる累算演算を行うように構成される累算器ブロック 72 の例を、以下の図 9 A ~ 図 9 C に示す。

【 0 0 6 4 】

[0078] この点において、図 9 A は、図 8 の累算器ブロック 72 の概略図である。この例では、累算器ブロック 72 は、二重 24 ビット累算器モードを提供するようにプログラムされた、プログラマブルな入力データパス 78 [0]、78 [1] と、プログラマブルな内部データパス 80 [0]、80 [1] とを有する。図 9 A の累算器ブロック 72 中の各桁上げ保存累算器 72 [0]、72 [1] は、24 ビット累算器を提供するように構成される。累算器 72 [0]、72 [1] の 24 ビット累算容量は、圧縮器 108 (0)、108 (1) のビット容量によって与えられる。図 8 に関連した累算器 72 [0]、72 [1] の考察により、図 9 A の累算器 72 [0]、72 [1] によって行われる個々の 24 ビット累算について説明する。累算ブロック 72 において二重 24 ビット累算を行うための累算器 72 [0]、72 [1] によって実施される累算の一般的なデータパスが、それぞれ、図 9 A の、プログラマブルな累算データパス 132 (0) および 132 (1) 中に示されている。

【 0 0 6 5 】

[0079] 引き続き図 9 A を参照して、桁上げ保存累算器 72 [0] の例示的な構成要素が記載されるが、桁上げ保存累算器 72 [1] にも等しく適用可能である。本実施形態では、桁上げ保存累算器 72 [0] の累算ベクトル出力和 112 (0) およびシフトされた累算ベクトル出力桁上げ 117 (0)、ならびに桁上げ保存累算器 72 [1] の累算ベクトル出力和 112 (1) およびシフトされた累算ベクトル出力桁上げ 117 (1)、そして、それぞれ、制御ゲート 134 (0)、136 (0) および 134 (1)、136 (1) によってゲートされる。制御ゲート 134 (0)、136 (0) および 134 (1)、136 (1) は、それぞれ、圧縮器 108 (0)、108 (1) に戻される、累算ベクトル出力和 112 (0) およびシフトされた累算ベクトル出力桁上げ 117 (0) と、累算ベクトル出力和 112 (1) およびシフトされた累算ベクトル出力桁上げ 117 (1) とを制御する。ベクトル命令の復号から、制御ゲート 134 (0)、136 (0) と 134 (1)、136 (1) の両方に与えられた制御入力 138 (0)、138 (1) は、それぞれ、累算ベクトル出力和 112 (0) と、シフトされた累算ベクトル出力桁上げ 117 (0) とを制御し、累算ベクトル出力和 112 (1) およびシフトされた累算ベクトル出力桁上げ 117 (1) は、それぞれ、圧縮器 108 (0)、108 (1) に戻される。制御入力 138 (0)、138 (1) および制御ゲート 134 (0)、136 (0) は、累算器 72 [0]、72 [1] が、図 9 B および図 9 C に関連して後でより詳しく論じるように、別個の累算演算それとも組み合わせられた累算演算を各々が実施するようにプログラムされるかを制御する。したがって、制御入力 138 (0)、138 (1) ならびに制御ゲート 134 (0)、136 (0) および 134 (1)、136 (1) は、本実施形態では、それぞれ、累算器 72 [0]、72 [1] のプログラマブルな内部データパス 80 [0]、80 [1] の一部を形成する。

【 0 0 6 6 】

[0080] 図 8 に参照を戻すと、累算器ブロック 72 のプログラマブルな内部データパス 80 [0]、80 [1] はまた、個々の累算器 72 [0]、72 [1] の 24 ビット容量よりも大きい累算演算を実施するようにプログラムされ、かつ/または再プログラムされ得る。この点において、累算器 72 [0]、72 [1] のプログラマブルな内部データパス 80 [0]、80 [1] は、単一ベクトル累算演算において一緒に利用されるべき両方の累算器 72 [0]、72 [1] を提供するようにプログラムされ得る。累算器 72 [0]、72 [1] は、累算器 72 [0]、72 [1] の個々のビット累算容量よりも大きいビット容量の単一累算演算を行うようにプログラムされ得る。累算器 72 [0]、72 [1] のプログラマブルな内部データパス 80 [0]、80 [1] は、桁上げ保存累算器 72

[0] に、圧縮器 1 0 8 (0) から、次の桁上げ出力 (N C O) 1 3 7 (0) としてオーバーフロー桁上げ出力を伝搬させるように構成され得る。N C O 1 3 7 (0) は、次の桁上げ入力 (N C I) 1 3 9 (1) として、桁上げ保存累算器 7 2 [1] 中の圧縮器 1 0 8 (1) に与えられ得る。図 8 に関連して上で記載したように、本実施形態では、4 0 ビット累算を行うための累算器 7 2 [0]、7 2 [1] に、2 4 ビット桁上げおよび和累算への 2 4 ビットオーバーフロー桁上げ伝搬を行わせるための、累算器 7 2 [0]、7 2 [1] 中でプログラマブルな内部データバス 8 0 [0]、8 0 [1] として提供されるこの桁上げ伝搬構成能力。

【 0 0 6 7 】

[0081] この点において、図 9 B は、図 8 の同じ累算器ブロック 7 2 の概略図である。図 9 B において、累算器ブロック 7 2 が、単一累算演算モードで構成されて示されている。図 9 B において、累算器 7 2 [0]、7 2 [1] は、累算器ブロック 7 2 中で単一累算演算を行うために構成されたプログラマブルな内部データバス 8 0 [0]、8 0 [1] を有する。累算器ブロック 7 2 の単一累算器モードのこの例において、累算器 7 2 [0]、7 2 [1] は、単一 4 0 ビット累算ベクトル出力和 1 1 2 とシフトされた累算ベクトル出力桁上げ 1 1 7 とを累算する。単一累算ベクトル出力和 1 1 2 は、プログラマブルな出力データバス 7 4 [0]、7 4 [1] 中で累算器出力のサンプルセット 7 6 として与えられる累算ベクトル出力和 1 1 2 (0)、1 1 2 (1) からなる (図 3 も参照)。累算ベクトル出力和 1 1 2 (0) は、単一 4 0 ビット累算ベクトル出力和 1 1 2 の最下位ビットを備える。累算ベクトル出力和 1 1 2 (1) は、単一 4 0 ビット累算ベクトル出力和 1 1 2 の最上位ビットを備える。同様に、シフトされた累算ベクトル出力桁上げ 1 1 7 は、シフトされた累算出力桁上げ 1 1 7 (0)、1 1 7 (1) からなる。シフトされた累算ベクトル出力桁上げ 1 1 7 (0) は、単一シフト累算ベクトル出力桁上げ 1 1 7 の最下位ビットを備える。累算ベクトル出力和 1 1 4 (1) は、単一 4 0 ビット累算ベクトル出力桁上げ 1 1 4 の最上位ビットを備える。累算ベクトル出力和 1 1 4 [0] およびシフトされた累算ベクトル出力桁上げ 1 1 7 (0) は、プログラマブルな出力データバス 7 4 [0] 中に設けられる (図 3 参照)。

【 0 0 6 8 】

[0082] 引き続き図 9 B を参照すると、累算器ブロック 7 2 が単一累算モード (たとえば、4 0 ビット累算) で構成されるときに累算器 7 2 [0]、7 2 [1] によって実施される累算演算の一般的なデータバスが、プログラマブルな累算データバス 1 4 6 として示されている。この点において、累算器ブロック 7 2 は、上で記載したように、ベクトル入力 9 4 とベクトル入力桁上げ 9 6 とを受け取る。桁上げ保存累算器 7 2 [0] は、場合によっては、ベクトル入力 9 4 [0] およびベクトル入力桁上げ 9 6 [0]、または負のベクトル入力 9 4 [0] ' および負のベクトル入力桁上げ 9 6 [0] の累算からの、累算ベクトル出力和 1 1 2 (0) と、累算ベクトル出力桁上げ 1 1 4 (0) との最下位ビットを累算する。桁上げ保存累算器 7 2 [1] は、場合によっては、ベクトル入力 9 4 [0] およびベクトル入力桁上げ 9 6 [0]、または負のベクトル入力 9 4 [0] ' および負のベクトル入力桁上げ 9 6 [0] の累算からの、累算ベクトル出力和 1 1 2 (1) と、累算ベクトル出力桁上げ 1 1 4 (1) との最上位ビットを累算する。

【 0 0 6 9 】

[0083] 引き続き図 9 B を参照すると、単一累算ベクトル出力和 1 1 2 と、累算ベクトル出力桁上げ 1 1 4 とを与えるように累算器 7 2 [0]、7 2 [1] をプログラムするために、累算器 7 2 [0]、7 2 [1] のプログラマブルな内部データバス 8 0 [0]、8 0 [1] は、単一累算演算を行うようにプログラムされる。この点において、桁上げ保存累算器 7 2 [0] の圧縮器 1 0 8 (0) の N C O 1 3 7 (0) および桁上げ保存累算器 7 2 [1] の圧縮器 1 0 8 (1) の N C I 1 3 9 (1) は、累算器ブロック 7 2 中に単一累算器 (たとえば、4 0 ビット累算器) を設けるために構成される。桁上げ保存累算器 7 2 [1] の N C I 1 3 9 (1) は、N C I 制御入力 1 4 2 (1) を用いて N C I ゲート 1 4 0 (1) によってゲートされる。このようにして、累算器ブロック 7 2 中の累算器 7 2 [0]

」、72[1]が、桁上げ保存累算器72[0]から桁上げ保存累算器72[1]へのオーバーフロー桁上げ伝搬を利用して単一累算演算を行うことを所望されるとき、NCI制御入力142(1)は、桁上げ保存累算器72[1]のプログラマブルな内部データバス80[1]の一部としてアクティブにされ得る。NCI制御入力142(1)は、NCIゲート140(1)に、圧縮器108(0)から圧縮器108(1)へのオーバーフロー桁上げ伝搬を許可させる。NCI制御入力142(1)は、圧縮器108(0)に、圧縮器108(1)へのNCI139(1)としてNCO137(0)を生成させるように、桁上げ保存累算器72[0]中の圧縮器108(0)の桁上げ伝搬入力144(0)にも結合される。これらの構成により、桁上げ保存累算器72[1]は、単一累算ベクトル出力和112と、累算ベクトル出力桁上げ114とを与え得るように、ベクトル入力94[1]と、ベクトル入力桁上げ96[1]とを累算することができるようになる。

【0070】

[0084]累算器ブロック72中の桁上げ保存累算器72[1]は、図9Bに示すように、NCI139(0)およびNCI制御入力142(0)によってゲートされるNCIゲート140(0)も含むことに留意されたい。ただし、桁上げ保存累算器72[0]が単一累算演算の最下位ビットを累算するので、NCI制御入力142(0)とNCI139(0)の両方は、本実施形態では論理「0」に結び付けられる。桁上げ保存累算器72[1]の圧縮器108(0)は、桁上げ伝搬入力144(1)も含むが、桁上げ伝搬入力144(1)は、本実施形態では、累算器72(1)にNCO12(1)を生成させないために、論理「0」に結び付けられることにも留意されたい。桁上げ保存累算器72[1]は、本実施形態では、次の累算器への桁上げ伝搬を実施する必要はなく、というのは、本実施形態において設けられる桁上げ保存累算器72[1]を超える、累算器ブロック72の別の累算器がないからである。

【0071】

[0085]図9Cは、図8の同じ累算器ブロック72の別の累算器モードの概略図である。この点において、図9Cは、桁上げ保存加算器と、それに続く、累算器ブロック72中で別の累算モードを提供するための単一累算器を提供するための、プログラムされたデータバス構成を有する、図8の累算器ブロック72の概略図である。いくつかの累算演算は、複素算術を実施するための余剰加算器を必要とし得る。図9Cにおいて、累算器72[0]、72[1]は、16ビット桁上げ保存加算器と、それに続く単一24ビット累算器を提供するために構成されたプログラマブルな内部データバス80[0]、80[1]を有する。

【0072】

[0086]図9Cを参照すると、累算器ブロック72が桁上げ保存加算器と、それに続く単一累算器中で構成されるときに累算器72[0]、72[1]によって実施される累算の一般的なデータバスが、プログラマブルなデータバス148によって示されている。この点において、和桁上げセクタ120(0)は、マルチプレクサ118(0)に、ベクトル入力94[1]と、ベクトル入力桁上げ96[0]とを制御ゲート134(0)、136(0)に与えさせるように、ベクトル命令復号の結果として生成される。制御入力138(0)は、制御ゲート134(0)、136(0)に、ベクトル入力94[1]と、ベクトル入力桁上げ96[1]とを圧縮器108(0)へ与えさせるように、桁上げ保存累算器72[0]のプログラマブルな内部データバス80[1]をプログラムするようにアクティブにされる。ベクトル入力94[1]およびベクトル入力桁上げ96[1]は、累算ベクトル出力和112(0)と、累算ベクトル出力桁上げ114(0)とを与えるように、桁上げ保存累算器72[0]の圧縮器108(0)によって、ベクトル入力94[0]およびベクトル入力桁上げ96[1]と累算される。累算ベクトル出力和112(0)およびシフトされた累算ベクトル出力桁上げ117(0)は、桁上げ保存加算器を設けるように、プログラマブルな出力データバス74[0](図3も参照)中で累算器出力のサンプルセット76[0]として与えられる。シフトされた累算ベクトル出力桁上げ117(0)は、24ビット累算器としての、累算ベクトル出力和112(1)と、シ

フトされた累算ベクトル出力桁上げ 1 1 7 (1) とを、プログラマブルな出力データパス 7 4 [1] (図 3 も参照) 中で累算器出力のサンプルセット 7 6 [1] として与えるようにベクトル入力和 9 4 [1] およびベクトル入力桁上げ 9 6 [1] に累算されるために、マルチプレクサ 1 0 4 (1) によって圧縮器 1 0 8 (1) に与えられるように、プログラマブルなデータパス 1 4 8 中に示される桁上げ保存累算器 7 2 [1] にも与えられる。

【 0 0 7 3 】

[0087] 図 5 ~ 図 9 C で上述したように、V P E 2 2 中に設けられる乗算器ブロック 6 2 (0 ~ A) および累算器ブロック 7 2 (0 ~ A) は、基数 2^x のバタフライベクトル演算を行うようにも都合よくプログラムされ得る。基数 2^x のバタフライベクトル演算は、F F T ベクトル処理演算を実施するのに使われ得る。F F T 演算は一般に、例として、線形フィルタリング、相関分析、およびスペクトル分析など、様々なデジタル信号処理アプリケーションにおいて実施される。したがって、F F T ベクトル処理演算を、低減された処理時間で、および電力効率的に実施することができる V P E を設けることが有利であり得る。

10

【 0 0 7 4 】

[0088] 知られているように、離散 F F T (D F T) は、N 個の複素数値数の数列 { X (k) } を、長さ N の別のデータ列 { x (n) } が与えられると、以下の公式に従って計算することによって計算され得る。

【 数 1 】

20

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}, \quad 0 \leq k \leq N-1$$

$$W_N = e^{-j2\pi/N}$$

【 0 0 7 5 】

[0089] 上の X (k) についての公式から、k の各値について、X (k) の直接計算は、N 回の複素乗算 (4 N 回の実乗算) と、N - 1 回の複素加算 (4 N - 2 回の実加算) とを伴うことが観察される。したがって、D F T のすべての N 個の値を計算することは、N² 回の複素乗算と、N² - N 回の複素加算とを必要とする。以下に示す位相因子 W_N の対称性および周期性プロパティを認識することによって、N = 2^x ポイント D F T を計算するのに、分割統治法が利用されることができる。

30

【 数 2 】

$$\text{対称性プロパティ : } W_N^{k+N/2} = -W_N^k$$

$$\text{周期性プロパティ : } W_N^{k+N} = W_N^k$$

40

【 0 0 7 6 】

[0090] N ポイントデータ列は、2 つの N / 2 ポイントデータ列 f 1 (n) および f 2 (n) に分割され、それは x (n) の偶数番号および奇数番号サンプルにそれぞれ対応する、すなわち、

【数 3】

$$f_1(n) = x(2n)$$

$$f_2(n) = x(2n+1), \quad n = 0, 1, \dots, \frac{N}{2}-1$$

【0077】

または

【数 4】

10

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}, \quad k = 0, 1, \dots, N-1$$

$$= \sum_{n \text{ even}} x(n) W_N^{kn} + \sum_{n \text{ odd}} x(n) W_N^{kn}$$

$$= \sum_{m=0}^{\{N/2\}-1} x(2m) W_N^{2mk} + \sum_{m=0}^{\{N/2\}-1} x(2m+1) W_N^{k(2m+1)}$$

20

【0078】

[0091] $W_N^2 = W_{N/2}$ 代入を用いて、上の後者の等式は、以下のように表現され得る。

【数 5】

$$X(k) = \sum_{m=0}^{\{N/2\}-1} f_1(m) W_{N/2}^{km} + W_N^k \sum_{m=0}^{\{N/2\}-1} f_2(m) W_{N/2}^{km}$$

$$= F_1(k) + W_N^k F_2(k), \quad k = 0, 1, \dots, N-1$$

30

【0079】

[0092] $F_1(k)$ および $F_2(k)$ は周期的なので、 $N/2$ の周期で、 $F_1(k + N/2) = F_1(k)$ および $F_2(k + N/2) = F_2(k)$ となる。さらに、回転因子は $W_N^k + N/2 = -W_N^k$ である。したがって、等式は次のように表すことができ、乗算の回数を、 N^2 から $N/2$ の $g_2 N$ 回の乗算に低減させる。

【数 6】

40

$$X(k) = F_1(k) + W_N^k F_2(k), \quad k = 0, 1, \dots, \frac{N}{2}-1$$

$$X(k + \frac{N}{2}) = F_1(k) - W_N^k F_2(k), \quad k = 0, 1, \dots, \frac{N}{2}-1$$

【0080】

[0093] この点において、図 10 ~ 図 12 に関連して以下でより詳細に説明するように、VPE22 は、他の算術論理ベクトル演算を行うのに使われ得ると同じ乗算器ブロック

50

62(0~A)と累算器ブロック72(0~A)とを用いてFFT演算を行うことができる。プログラマブルなデータパス90(3)~90(0)、91、および92(3)~92(0)、乗算器ブロック62(0)~62(A)(図3参照)中の各乗算器84(0)~84(3)(図6参照)のそれぞれは、実行される特定の基数 2^x のバタフライベクトル命令に従って、基数 2^x のバタフライベクトル演算のポイントについての乗算を行うようにプログラムされ得る。累算器ブロック72(0)~72(A)(図3参照)のプログラマブルな入力データパス78(0)~78(A)、内部データパス80(0)~80(A)、および/または出力データパス74(0)~74(A)は、複数の乗算器ブロック62(0)~62(A)の中の乗算器ブロック62からの基数 2^x のバタフライ乗算積を、実行される基数 2^x のバタフライベクトル命令に従って累算するようにプログラムされ得る。累算器ブロック72(0)~72(A)のプログラマブルな入力データパス78(0)~78(A)、内部データパス80(0)~80(A)、および/または出力データパス74(0)~74(A)はまた、複数の乗算器ブロック62(0)~62(A)の中の他の乗算器ブロック62からの基数 2^x のバタフライ乗算積を、実行される基数 2^x のバタフライベクトル命令に従って累算するように、他の周期において再プログラムされ得る。累算器ブロック72(0)~72(A)によって累算された最終的な累算基数 2^x のバタフライ乗算積は、FFTベクトル処理演算のN個のポイントを与えることができる。

10

【0081】

[0094]乗算器ブロック62(0)~62(A)および累算器ブロック72(0)~72(A)は、スケーリングでのFFT、乗算でのFFT、位相ランプ乗算でのFFT、補間FFT、高速畳込み、およびFHT変換など、他の関連FFT演算にも構成され得る。その結果、より少ないVPEで、ベクトルプロセッサにおいて所望の基数 2^x のバタフライベクトル演算と他のタイプの算術論理ベクトル演算とを行うことができ、したがって、ベクトルプロセッサ中の面積を節約するとともに、スカラー処理エンジンに勝る、より少ないレジスタ書込みおよびより速いベクトル命令実行時間というベクトル処理利点を依然として保有する。

20

【0082】

[0095]DFTにおけるデータ点Nの数が4の累乗である(すなわち、 $N = 4^x$)とき、基数2のアルゴリズムが計算に利用され得る。ただし、このケースの場合、基数 2^x のFFTアルゴリズムを利用すると、より計算効率が良い。たとえば、Nポイント入力列を4つの部分列、すなわち $x(4n)$ 、 $x(4n+1)$ 、 $x(4n+2)$ 、 $x(4n+3)$ 、 $n = 0, 1, \dots, N/4 - 1$ に分割またはデシマイト(decimate)することによって、基数4のFFTアルゴリズムが提供され得る。

30

【数 7】

$$X(p, q) = \sum_{l=0}^3 [W_N^{lq} F(l, q)] W_4^{lp}$$

$$F(l, q) = \sum_{m=0}^{(N/4)-1} x(l, m) W_{N/4}^{mq}$$

$$p = 0, 1, 2, 3; \quad l = 0, 1, 2, 3; \quad q = 0, 1, 2, \dots, \frac{N}{4} - 1$$

10

および

$$x(l, m) = x(4m + l)$$

$$X(p, q) = X\left(\frac{N}{4}p + q\right)$$

20

【0083】

[0096]したがって、上記等式から取得された4つのN/4ポイントDFT F(1, q)は、NポイントDFTをもたらしように組み合わせられる。N/4ポイントDFTを組み合わせるための表現は、基数4の時間分割バタフライ演算を定義し、これは、次のように行列の形で表現され得る。

【数 8】

$$\begin{bmatrix} X(0, q) \\ X(1, q) \\ X(2, q) \\ X(3, q) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} W_N^0 F(0, q) \\ W_N^1 F(1, q) \\ W_N^{2q} F(2, q) \\ W_N^{3q} F(3, q) \end{bmatrix}$$

30

【0084】

[0097]上に示す基数4のバタフライ演算行列は、ポイントqに対するNポイントDFTにおける単一ポイントとして与えられたN/4ポイントを再結合するように解くことができる。すべてのポイントqについてのDFT全体を解くために、NポイントDFT中について算出された単一ポイントが累算され得る。

40

【0085】

[0098]この点において、図10は、上に示す基数4のバタフライ演算行列を解くための基数バタフライ回路146を形成するように、累算ベクトル処理段階60(2)(図8参照)中の2つの累算器ブロック72(0)、72(1)に並列に融合された乗算ベクトル処理段階60(1)(図5参照)中の乗算器84をもつ図3のVPE22の概念的概略図である。図10を参照すると、基数バタフライ回路146は、基数2または基数4のバタフライベクトル演算のいずれかを実施するように構成される。VPE22は、乗算器ブロック62が4つの累算器、すなわち累算器ブロック72[1]中の累算器72[1][1]、72(1)(0)および累算器ブロック72[0]中の累算器72(0)(1)、72(0)(0)に融合されるようにプログラムされ、これらは、本実施形態における基数

50

バタフライ回路 146 を設けるように、冗長桁上げ保存フォーマットで構成される。冗長桁上げ保存フォーマットで構成されることが可能な累算器ブロック 72 についての考察は、図 7 ~ 図 9 C に関連して上で前述した。乗算器ブロック 84 は、N ポイント DFT の 1 つのポイントを与えるように、4 つの回転因子で乗算された単一ベクトルデータ入力のサンプルセット 66 (すなわち、N / 4 基数ポイント) の 4 つの乗算積を与え得る。基数バタフライ回路 146 中に含まれるようにプログラムされた乗算器ブロック 62 中の乗算器 84 の数は、実行される特定の基数バタフライベクトル命令に従って基数バタフライベクトル演算を実施する際に使われるように、ビット幅ベクトルデータ入力のサンプルセット 66 に基づく。

【0086】

10

[0099]引き続き図 10 を参照すると、回転因子ジェネレータ 148 (3) ~ 148 (0) が VPE 22 中に設けられる。回転因子ジェネレータ 148 (3) ~ 148 (0) は、それぞれ、バタフライレーン 149 (3) ~ 149 (0) 中で回転因子 W_N^q を、基数バタフライ回路 146 中に含まれるようにプログラムされた乗算器ブロック 62 中の乗算器 84 に与えられるように生成する。回転因子ジェネレータ 148 (3) ~ 148 (0) を設けることによって、基数バタフライベクトル演算用の回転因子 W_N^q が与えられ得る。基数バタフライ回路 146 が基数 2 のバタフライベクトル演算用に構成される場合、単位元 (「1」) である、 W_N^0 の単一回転因子のみが、回転因子ジェネレータ 148 (0) から必要とされる。ただし、基数バタフライ回路 146 が基数 4 のバタフライベクトル演算用に構成される場合、4 つの回転因子 $W_N^3 \sim W_N^0$ が、以下の回転因子行列に従って回転因子ジェネレータ 148 (3) ~ 148 (0) によって生成されることを必要とされる。

20

【数 9】

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

30

【0087】

[00100]本実施形態では、図 10 の同じ乗算器ブロック 62 は、4 回繰り返され、4 つの累算器 72 (1) (1)、72 (1) (0)、72 (0) (1)、72 (0) (0) に融合される。というのは、本実施形態における基数バタフライ回路 146 が構成され得るからであり、ここにおいて単一乗算器ブロック 62 は、N ポイント DFT の 1 つのポイントを与えるように、4 つの回転因子で乗算された単一ベクトルデータ入力のサンプルセット 66 (すなわち、N / 4 基数ポイント) の 4 つの乗算積を与え得る。基数バタフライ回路 146 中に含まれるようにプログラムされた乗算器ブロック 62 中の乗算器 84 の数は、実行される特定の基数バタフライベクトル命令に従って基数バタフライベクトル演算を実施する際に使われるように、ビット幅ベクトルデータ入力のサンプルセット 66 に基づく。また、図 10 の基数バタフライ回路 146 中の累算器ブロック 72 [1] 中の各累算器 72 [1] [1]、72 [1] [0] および累算器ブロック 72 [0] 中の累算器 72 [0] (1)、72 [0] (0) は、基数バタフライ演算のビット幅に依存して、N / 4 基数ポイントを累算するようにプログラムされ、追加累算器ブロック 72 は、基数バタフライ回路 146 中に含まれるようにプログラムされてもよい。

40

【0088】

[00101]また、本実施形態では、図 10 の基数バタフライ回路 146 は、クロック周期ごとに、N ポイントの基数 4 のバタフライ演算用に N / 2 回の $\log_2 N$ 乗算と 4 つの累算とを行うことができる。したがって、N ポイントの基数 4 のバタフライベクトル演算の

50

1つのバタフライ演算を完了するのに、4つのクロック周期だけが必要とされる。これは、基数2のバタフライ演算と同じスループットである。したがって、基数バタフライ回路146は、基数4のバタフライベクトル演算をN回のクロック周期で完了することができる。本明細書における実施形態によるVPE22は、乗算器ブロック62(0)~62(A)および累算器ブロック72(0)~72(A)用のデータバス構成に従って、いくつかの基数バタフライ回路146を設けるように構成され得ることに留意されたい。したがって、VPE22が、クロック周期ごとにM回のNポイントの基数バタフライベクトル演算を実施するように、たとえば、一例として基数16のドラゴンフライベクトル演算を実施するように構成され得る場合、VPE22がNポイント基数4のバタフライトランザクションを実施するための全体的クロック周期カウンタは、 $(N/2 \log_2 N)/M$ まで低減される。さらに、乗算器ブロック62を、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72[91]、72[0]と融合することによって、各累算ステップ中で桁上げ伝搬バスと桁上げ伝搬加算演算とを提供することは、上述したように必要とされず、そうすることによって、追加処理時間と電力消費とを節約する。

【0089】

[00102]さらなる例示を与えるために、図11は、基数2または基数4のバタフライベクトル演算のいずれかを実施するように構成された、図10の基数バタフライ回路146の1つの周期の概略図である。基数2のバタフライベクトル演算を実施するように構成された基数バタフライ回路146の記述について最初に記載し、その後基数4のバタフライベクトル演算が続く。上述した基数2のバタフライベクトル演算に関して、Nポイントの基数2のバタフライ演算を行うために、以下の等式が解かれることになる。

$$X(k) = F_1(k) + W_N^k F_2(k), \quad k = 0, 1, \dots, \frac{N}{2} - 1$$

$$X(k + \frac{N}{2}) = F_1(k) - W_N^k F_2(k), \quad k = 0, 1, \dots, \frac{N}{2} - 1$$

【0090】

[00103]引き続き図11を参照すると、ベクトルデータ入力のサンプルセット66[1]、66[0]が、乗算ベクトル処理段階60(1)において受け取られる。ベクトルデータ入力のサンプルセット66[1]、66[0]は、 $X(k)$ の偶数番号および奇数番号サンプルまたはポイントに対応する、2つの $N/2$ ポイントデータ列 $F_1(k)$ および $F_2(k)$ 中で分割されたDFTポイントからなる。「 k 」は、列中の1つの $N/2$ ポイントを表し、これは各周期について累算される。乗算器84は通常、ベクトル乗算出力のサンプルセット68[0]（すなわち、 $W_N^k F_2(k)$ ）を与えるように、 $F_2(k)$ を、バタフライレーン149上での入力としての回転因子 W の「 e 」成分および累乗成分で乗算するために、2つの乗算を行う必要がある。ただし、基数2のバタフライ演算用の回転因子は単位元なので、乗算器84は、この例では $F_2(k)$ を「 e 」で乗算するのに使われ得るだけである。したがって、乗算器84による中間乗算結果のレジスタ退避が節約される。累算器721は、累算されたベクトル出力和1121および累算されたベクトル出力桁上げ1171として、偶数番号列 $X(k)$ 中で $N/2$ ポイントを与えるように、ベクトル乗算出力のサンプルセット68[0]をデータ点 $F_1(k)$ に加える。累算器72[0](1)は、累算されたベクトル出力和112[1](0)および累算されたベクトル出力桁上げ117[1](0)として、奇数番号列 $X(k + N/2)$ 中で $N/2$ ポイントを与えるように、ベクトル乗算出力のサンプルセット68[0]をデータ点 $F_1(k)$ の否定(negation)に加える。否定回路106[0](1)は、データ点 $F_1(k)$ を、ベクトル乗算出力のサンプルセット68[0]に加算され

る前に否定する。追加ベクトルデータ入力のサンプルセット $66[1]$ 、 $66[0]$ が受け取られ、それぞれ、累算器 721 、 $72[0](1)$ に与えられ、それぞれ、偶数番号列 $X(k)$ および奇数番号列 $X(k + N/2)$ の合計を表す最終累算出力のサンプルセット 761 、 $76[1], (0)$ を与えるようにデータ列が合計されるまで、累算ベクトル出力和 1121 および累算ベクトル出力桁上げ 1171 、ならびに累算ベクトル出力和 $112[0](1)$ および累算ベクトル出力桁上げ $117[0](1)$ と累算される。

【0091】

[00104] 上述したように、基数バタフライ回路 146 は、基数 4 のバタフライベクトル演算を行うようにもプログラムされ得る。この点において、ベクトルフォーマットでの $N/4$ 個の基数ポイントを含むベクトルデータ入力のサンプルセット $66[0]$ が、乗算ベクトル処理段階 $60(1)$ において受け取られる。ベクトルデータ入力のサンプルセット $66[0]$ は、 $X(k)$ のサンプルまたはポイントの $N/4$ ポイントデータ列 $F_2(k)$ 中で分割された DFT ポイントからなる。「 k 」は、列中の 1 つの $N/4$ ポイントを表し、これは各周期について累算される。乗算器 84 は通常、 $F_2(k)$ を「 e 」成分で乗算するために、 2 つの乗算を実施し、ベクトル乗算出力のサンプルセット $68[0]$ (すなわち、 $W_N^k F_2(k)$) を与えるために、回転因子 (すなわち、 W_N^k) の累乗成分を判断する。ただし、基数の 4 バタフライ演算は、 4 つの可能回転因子 1 、 -1 、 j 、 $-j$ を含むだけなので、回転因子の累乗成分の複素乗算は、回転因子 -1 による乗算用の $F_2(k)$ の否定と、回転因子 j による乗算用の $F_2(k)$ のスワッピングと、 $-j$ についての回転因子による乗算用の $F_2(k)$ の否定およびスワッピングとを単に提供することによって回避され得る。 $F_2(k)$ の否定および / またはスワッピングは、レジスタ退避と電力消費とをさらに節約するために、第 1 の周期中にのみ回転因子の「 e 」成分での $F_2(k)$ の乗算を実施するのに乗算器 84 が代わりに使われ得る基数バタフライ回路 146 の位相ランプ演算中にプログラムされ得る。

【0092】

[00105] この点において引き続き図 11 を参照すると、累算器 721 は常に、以下に再現する基数 4 の回転因子行列の行 1 に従って、 1 の回転因子を与える。ただし、累算器 $72[1](0)$ は、以下に再現する基数 4 の回転因子行列の行 2 に従って、基数 4 の回転因子についての回転因子のうちのいずれをも与えることが可能でなければならない。したがって、累算器 $72[1](0)$ は、否定回路 $106[1](0)$ と、それぞれ、各 $N/4$ ポイント列の累算のために、基数 4 の回転因子の各々に従って $+F_2(k)$ 、 $-jF_2(k)$ 、 $-F_2(k)$ 、および $+jF_2(k)$ を累算器ブロック $72[1]$ 中の累算器 $72[1][1]$ 、 $72[1][0]$ および累算器ブロック $72[0]$ 中の累算器 $72[0](1)$ 、 720 に与えるように制御されるスワップ回路 $109[1](0)$ とを含む。累算器 $72[0](1)$ は、以下に再現する基数 4 の回転因子行列の行 3 に従って、基数 4 の回転因子についての回転因子のうちのいずれをも与えることが可能でなければならない。したがって、累算器 $72[0](1)$ は、それぞれ、各 $N/4$ ポイント列の累算のために $+F_2(k)$ を累算器 721 、 $72[0](1)$ に、および $-F_2(k)$ を累算器 $72[1](0)$ 、 720 に与えるのに、否定回路 $106[0](1)$ を必要とするだけである。累算器 720 は、以下に再現する基数 4 の回転因子行列の行 4 に従って、基数 4 の回転因子についての回転因子のうちのいずれをも与えることが可能でなければならない。したがって、累算器 720 は、否定回路 1060 と、それぞれ、各 $N/4$ ポイント列の累算のために、基数 4 の回転因子の各々に従って $+F_2(k)$ 、 $+jF_2(k)$ 、 $-F_2(k)$ 、および $-jF_2(k)$ を累算器ブロック $72[0]$ 中の累算器 $72[1][1]$ 、 $72[1][0]$ および累算器ブロック $72[0]$ 中の累算器 $72[0](1)$ 、 720 に与えるように制御されるスワップ回路 $109[1](0)$ とを含む。

【数 1 1】

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

10

【0 0 9 3】

[00106]したがって要約すると、図 1 1 の基数バタフライ回路 1 4 6 中の累算器 7 2 [1] [1]、7 2 [1] [0]、7 2 [0] (1)、7 2 [0] (0) は、基数 4 のバタフライベクトル演算を実施するようにプログラムされ、以下の算出を、次のように 4 周期にわたって実施する。これらの算出は、4 つの N / 4 累算ポイント列を与えるように累算され、これらの列は次いで、F F T を解決するために一緒に累算され得る。

【表 1】

周期	A0(1161)	A1(116[1](0))	A2(116[0](1))	A3(1160)
周期 0	$F_2(0)*W(0)$	$F_2(0)*W(0)$	$F_2(0)*W(0)$	$F_2(0)*W(0)$
周期 1	$F_2(1)*W(1)$	$-jF_2(1)*W(1)$	$-F_2(1)*W(1)$	$+jF_2(1)*W(1)$
周期 2	$F_2(2)*W(2)$	$-F_2(2)*W(2)$	$F_2(2)*W(2)$	$+F_2(2)*W(2)$
周期 3	$F_2(3)*W(3)$	$+jF_2(1)*W(3)$	$-F_2(1)*W(3)$	$-jF_2(1)*W(3)$

20

30

【0 0 9 4】

[00107]V P E 2 2 は、基数 1 6 のバタフライ演算（「ドラゴンフライ演算」としても知られる）能力を提供するために、図 1 1 の基数バタフライ回路 1 4 6 のような最大 4 つの基数のバタフライ回路を実施するようにもプログラムされ得る。このシナリオにおいて、回転因子は、単に + 1、= 1、+ j、- j よりも複雑になり、そうすることによって、各基数バタフライ回路 1 4 6 中の乗算器 8 4 による複素乗算を求める。

【0 0 9 5】

[00108]図 2 および図 3 の V P E 2 2 を含むが、それに限定されない、マルチモード基数 2^x のバタフライベクトル処理演算を行うためのプログラマブルなデータパス構成を有するとともに、本明細書において論じた概念および実施形態による V P E は、どのプロセッサベースのデバイス中に設けられても、統合されてもよい。例としては、限定はしないが、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、セルラーフォン、コンピュータ、ポータブルコンピュータ、デスクトップコンピュータ、携帯情報端末（P D A）、モニタ、コンピュータモニタ、テレビジョン、チューナー、ラジオ、衛星ラジオ、音楽プレーヤ、デジタル音楽プレーヤ、ポータブル音楽プレーヤ、デジタルビデオプレーヤ、ビデオプレーヤ、デジタルビデオディスク（D V D）プレーヤ、およびポータブルデジタルビデオプレーヤがある。

40

【0 0 9 6】

50

[00109]この点において、図 1 2 は、プロセッサベースのシステム 1 5 0 の例を示す。この例では、プロセッサベースのシステム 1 5 0 は、各々が 1 つまたは複数のプロセッサまたはコア 1 5 4 を含む、1 つまたは複数の処理装置 (P U) 1 5 2 を含む。 P U 1 5 2 は、非限定的例として、図 2 のベースバンドプロセッサ 2 0 であってよい。プロセッサ 1 5 4 は、非限定的例として、図 2 に挙げたベースバンドプロセッサ 2 0 のようなベクトルプロセッサであってよい。この点において、プロセッサ 1 5 4 は、図 2 の V P E 2 2 を含むが、それに限定されない V P E 1 5 6 も含み得る。 P U 1 5 2 は、一時的に記憶されたデータへの高速アクセスのためにプロセッサ 1 5 4 に結合されたキャッシュメモリ 1 5 8 を有し得る。 P U 1 5 2 は、システムバス 1 6 0 に結合され、プロセッサベースのシステム 1 5 0 に含まれるマスタデバイスとスレーブデバイスとを相互結合し得る。よく知られているように、 P U 1 5 2 は、システムバス 1 6 0 を介してアドレス、制御、およびデータ情報を交換することによって、これらの他のデバイスと通信する。たとえば、 P U 1 5 2 は、スレーブデバイスの例として、メモリコントローラ 1 6 2 にバストランザクション要求を通信することができる。図 1 2 には示されていないが、複数のシステムバス 1 6 0 が提供されてよく、ここにおいて、各システムバス 1 6 0 は異なる組織を構成する。

【 0 0 9 7 】

[00110]他のマスタおよびスレーブデバイスが、システムバス 1 6 0 に接続され得る。図 1 2 に示すように、これらのデバイスは、例として、メモリシステム 1 6 4 と、1 つまたは複数の入力デバイス 1 6 6 と、1 つまたは複数の出力デバイス 1 6 8 と、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 1 7 0 と、1 つまたは複数のディスプレイコントローラ 1 7 2 とを含み得る。メモリシステム 1 6 4 は、メモリコントローラ 1 6 2 によってアクセス可能なメモリ 1 6 5 を含み得る。入力デバイス 1 6 6 は、限定はしないが、入力キー、スイッチ、音声プロセッサなどを含む、任意のタイプの入力デバイスを含むことができる。出力デバイス 1 6 8 は、限定はしないが、オーディオ、ビデオ、他の視覚的インジケータなどを含む、任意のタイプの出力デバイスを含むことができる。ネットワークインターフェースデバイス 1 7 0 は、ネットワーク 1 7 4 との間でデータの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 1 7 4 は、限定はしないが、ワイヤードまたはワイヤレスネットワーク、プライベートまたは公共ネットワーク、ローカルエリアネットワーク (L A N)、ワイドローカルエリアネットワーク (W L A N)、およびインターネットを含む、任意のタイプのネットワークであってよい。ネットワークインターフェースデバイス 1 7 0 は、所望の任意のタイプの通信プロトコルをサポートするように構成され得る。

【 0 0 9 8 】

[00111] P U 1 5 2 は、1 つまたは複数のディスプレイ 1 7 8 に送られる情報を制御するため、システムバス 1 6 0 を介してディスプレイコントローラ 1 7 2 にアクセスするようにも構成され得る。ディスプレイコントローラ 1 7 2 は、1 つまたは複数のビデオプロセッサ 1 8 0 を介して表示されるべき情報をディスプレイ 1 7 8 に送り、1 つまたは複数のビデオプロセッサ 1 7 0 は、表示されるべき情報をディスプレイ 1 7 8 に好適なフォーマットに処理する。ディスプレイ 1 7 8 は、限定はしないが、陰極線管 (C R T)、液晶ディスプレイ (L C D)、プラズマディスプレイなどを含む、任意のタイプのディスプレイを含むことができる。

【 0 0 9 9 】

[00112]本明細書で開示する二重電圧ドメインメモリバッファの実施形態とともに説明する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、電子的な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り実行される命令として、またはこれら両方の組合せとして実装され得ることが、当業者にはさらに理解されよう。本明細書に記載されるアービタ、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 (I C)、または I C チップに採用され得る。本明細書で開示するメモリは、どのタイプおよびサイズのメモリでもよく、所望されるどのタイプの情報も記憶する

ように構成され得る。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、およびステップについて、上記では概してそれらの機能性に関して記載された。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、各特定の用途について様々な様式で記載された機能性を実装することができるが、そのような実装決定は、本開示の範囲からの逸脱を引き起こすと解釈されるべきでない。

【0100】

[00113]本明細書に開示された実施形態に関連して記載された、様々な例示の論理ブロック、モジュール、および回路は、本明細書に記載の機能を実施するよう設計された、プロセッサ、DSP、特定用途向け集積回路(ASIC)、FPGAまたは他のプログラマブル論理デバイス、個別ゲートまたはトランジスタロジック、個別ハードウェア構成要素、またはその任意の組合せを用いて実装または実施され得る。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、またはステートマシンであり得る。プロセッサはまた、コンピューティングデバイスの組合せ、たとえば、DSPとマイクロプロセッサとの組合せ、複数のマイクロプロセッサ、DSPコアと連携する1つもしくは複数のマイクロプロセッサ、または任意の他のそのような構成としても実装され得る。

10

【0101】

[00114]本明細書で開示する実施形態は、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実施されてよく、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記憶媒体中に常駐し得る。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。代替形態において、記憶媒体はプロセッサと一体であってもよい。プロセッサおよび記憶媒体はASIC内に存在し得る。ASICは遠隔局中に常駐し得る。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遠隔局、基地局、またはサーバ中に存在してよい。

20

【0102】

[00115]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ものであることにやはり留意されたい。説明された動作は、例示された順序以外の多くの異なる順序で実施されてもよい。さらに、単一の動作ステップで説明される動作は、実際には、多くの異なるステップで実施され得る。加えて、例示的な実施形態において議論された1つまたは複数の動作ステップは組み合わせられ得る。フローチャート図に示した動作ステップは、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得ることを理解されたい。情報および信号は、多種多様な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、当業者には理解されよう。たとえば、上記の説明全体にわたって言及され得るデータ、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、磁界もしくは磁性粒子、光場もしくは光子、またはそれらの任意の組合せによって表され得る。

30

40

【0103】

[00116]本開示の前の記載は、当業者が本開示を製造または使用することを可能にするように提供される。本開示に対する様々な変更形態が、当業者には容易に明らかとなり、本明細書に規定の一般的な原理が、本開示の精神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得る。したがって、本開示は、本明細書に記載される例および設計に限定することを意図しておらず、本明細書に開示される、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。

【 図 1 】

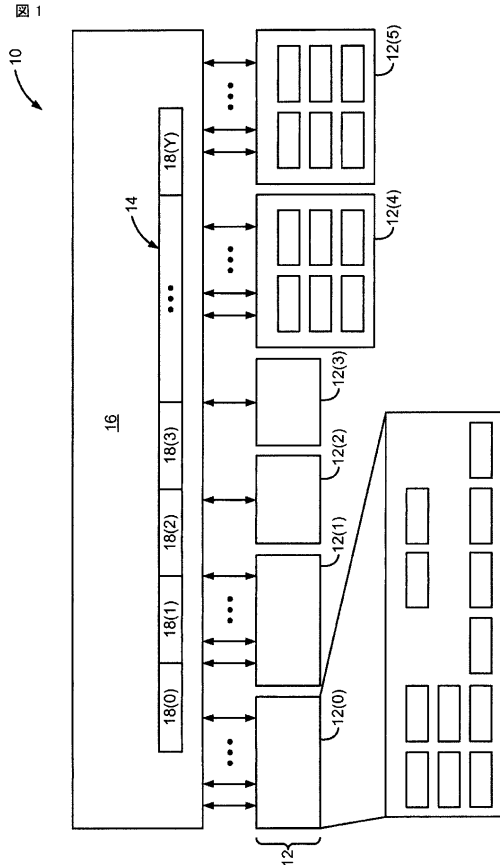


FIG. 1

【 図 2 】

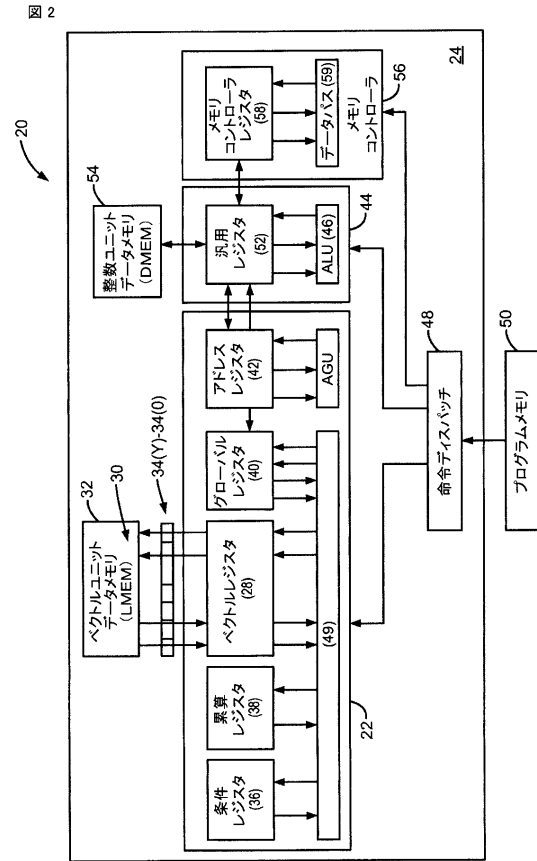


FIG. 2

【 図 3 】

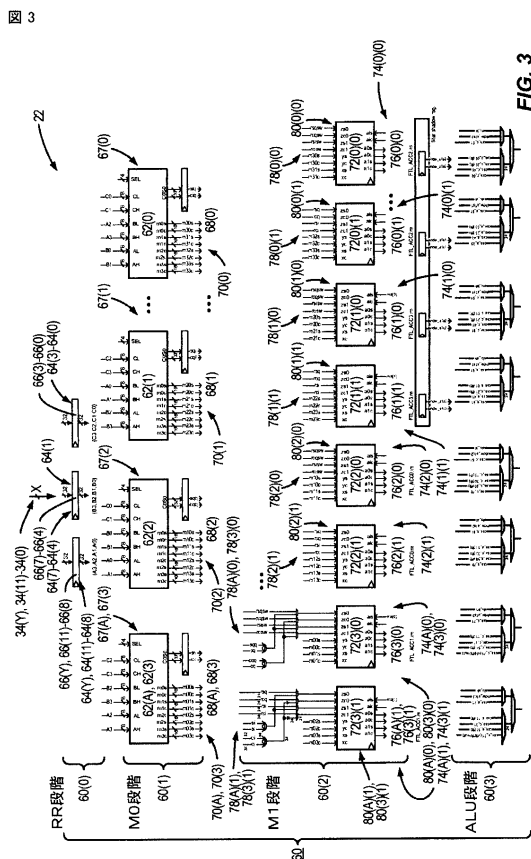


FIG. 3

【 図 4 A 】

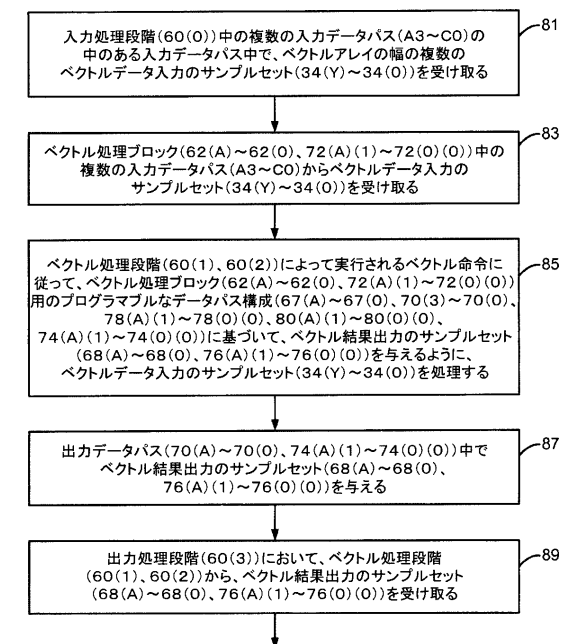


FIG. 4A

【図 4 B】

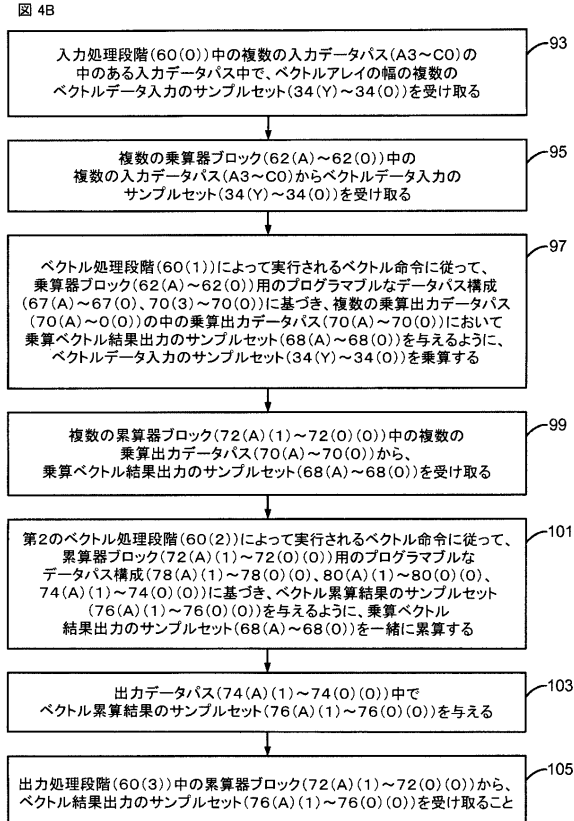


FIG. 4B

【図 6】

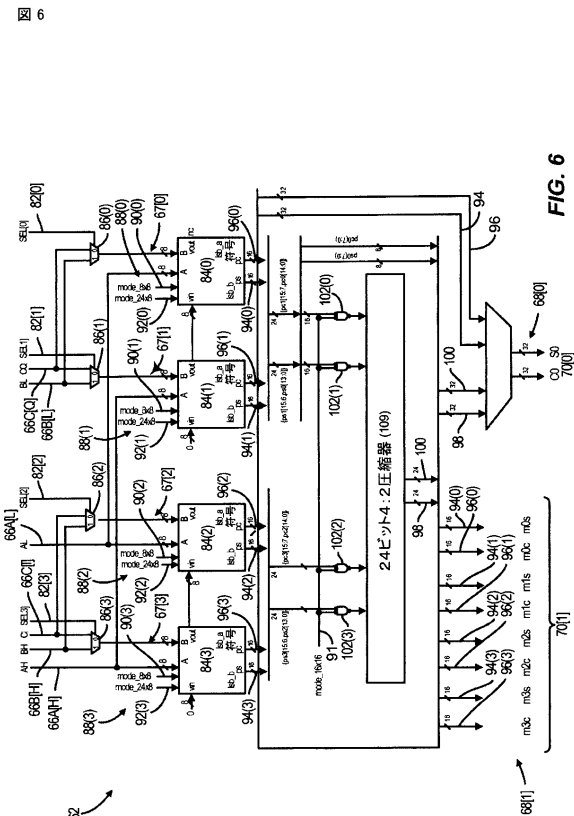


FIG. 6

【図 5】

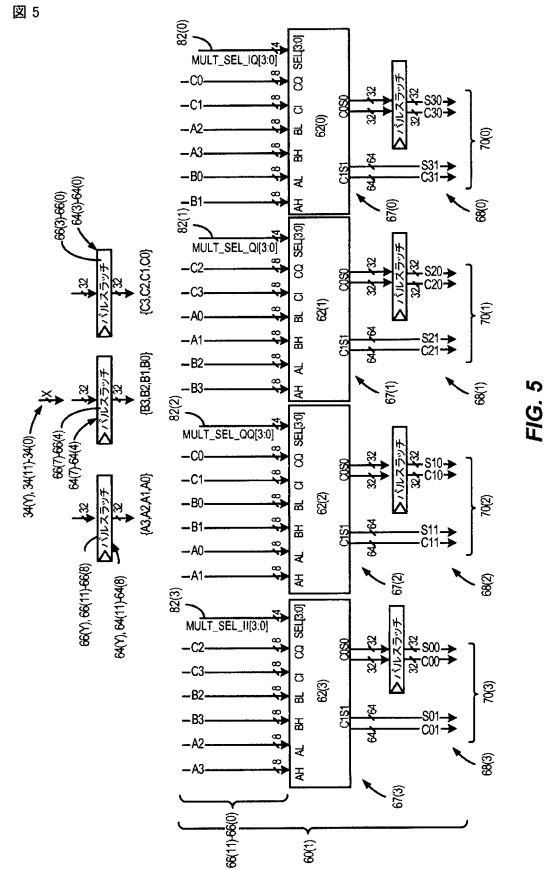


FIG. 5

【図 7】

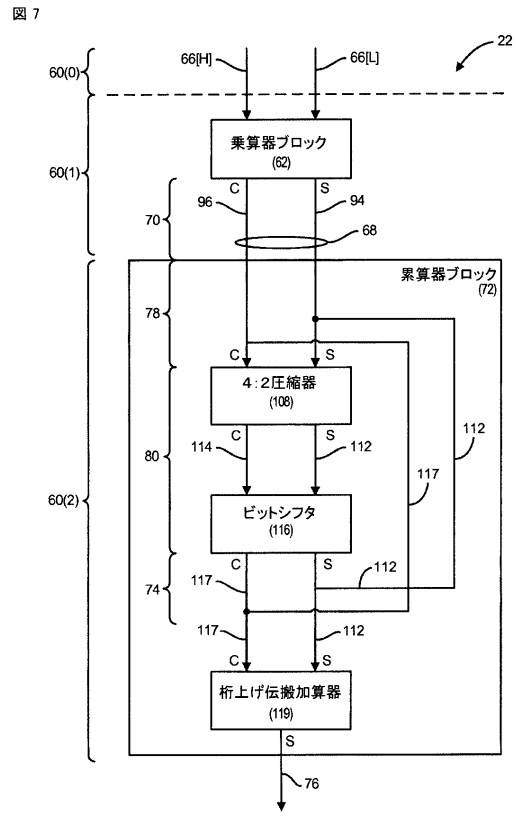
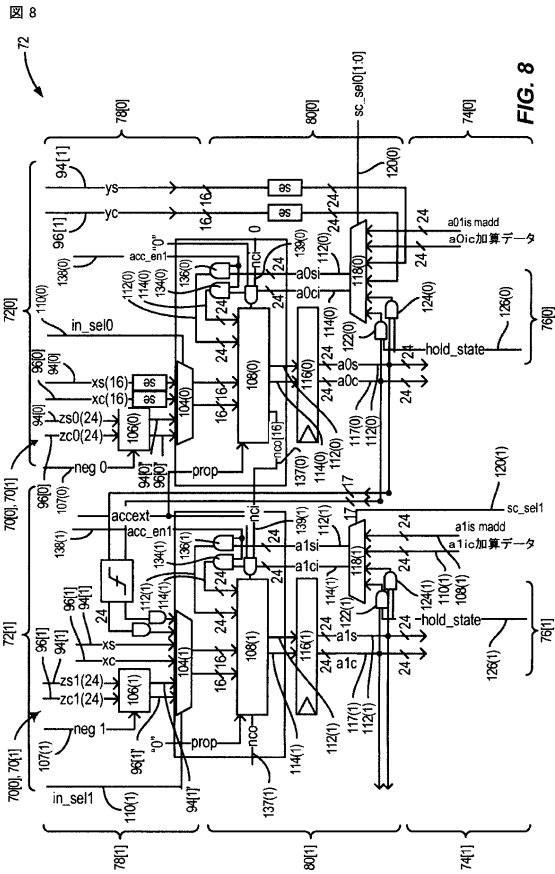
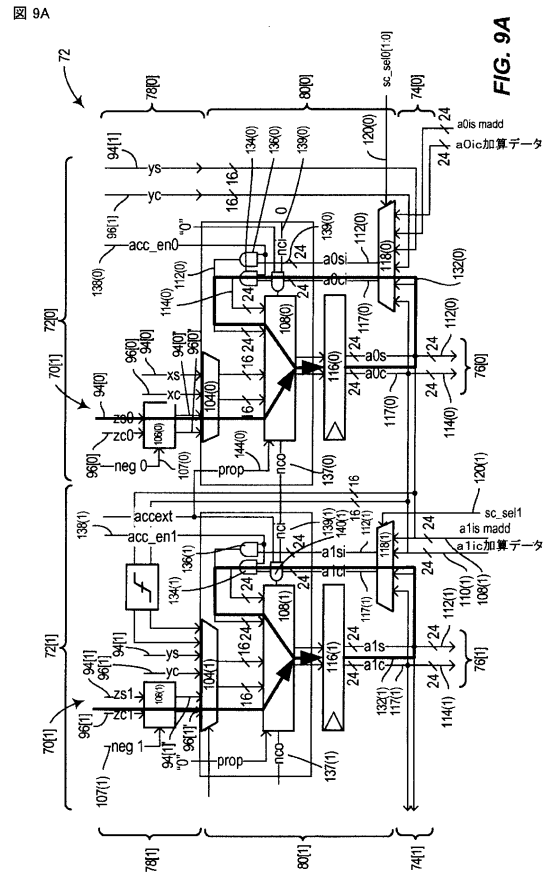


FIG. 7

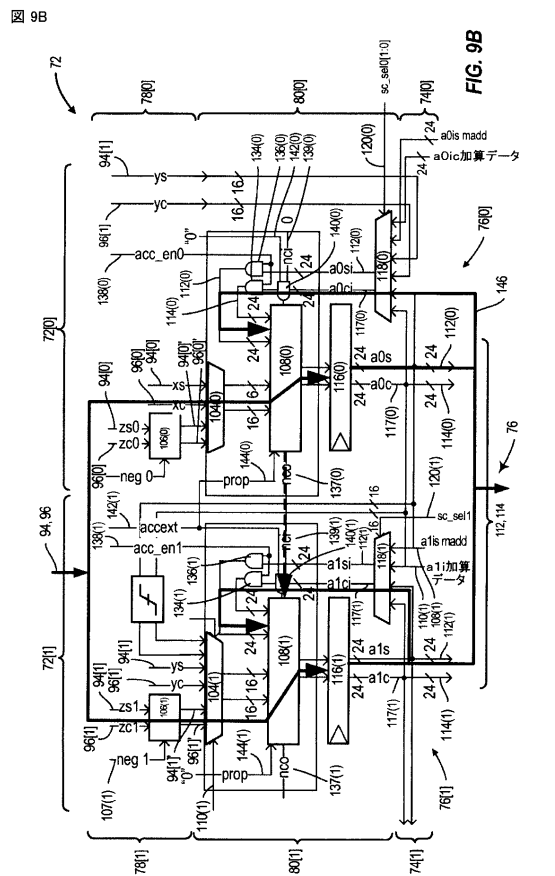
【図 8】



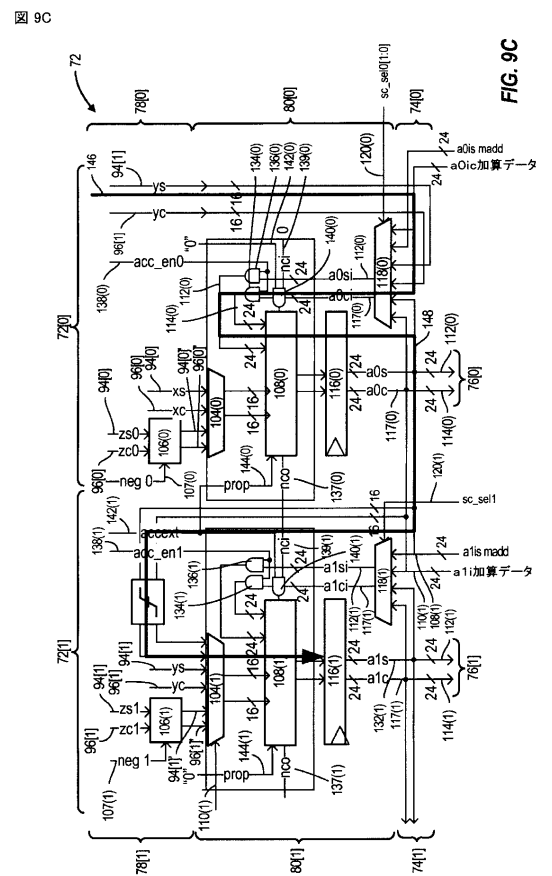
【図 9 A】



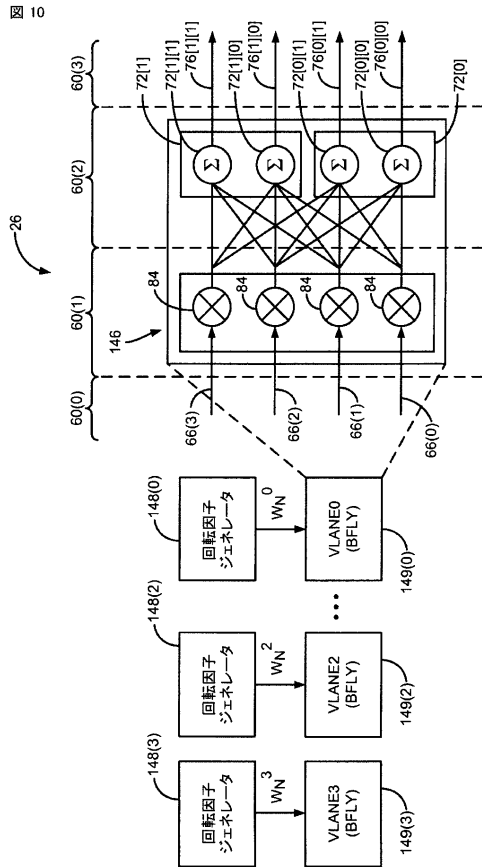
【図 9 B】



【図 9 C】



【 図 1 0 】



【 図 1 1 】

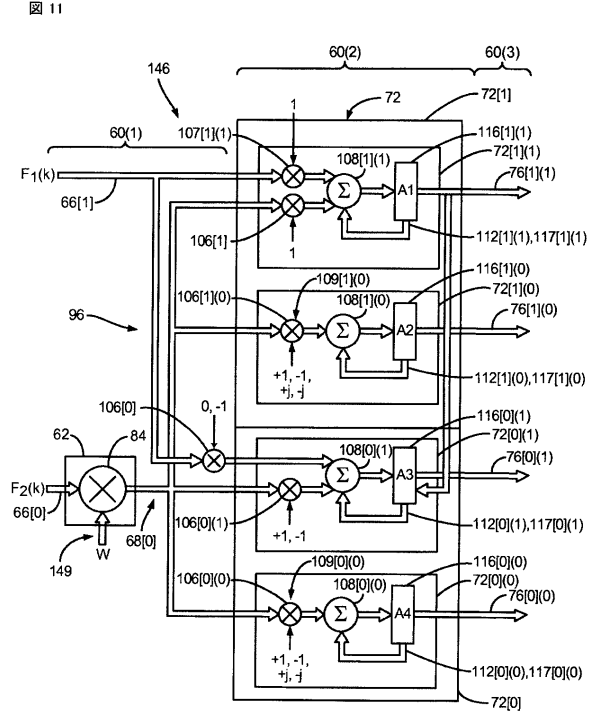
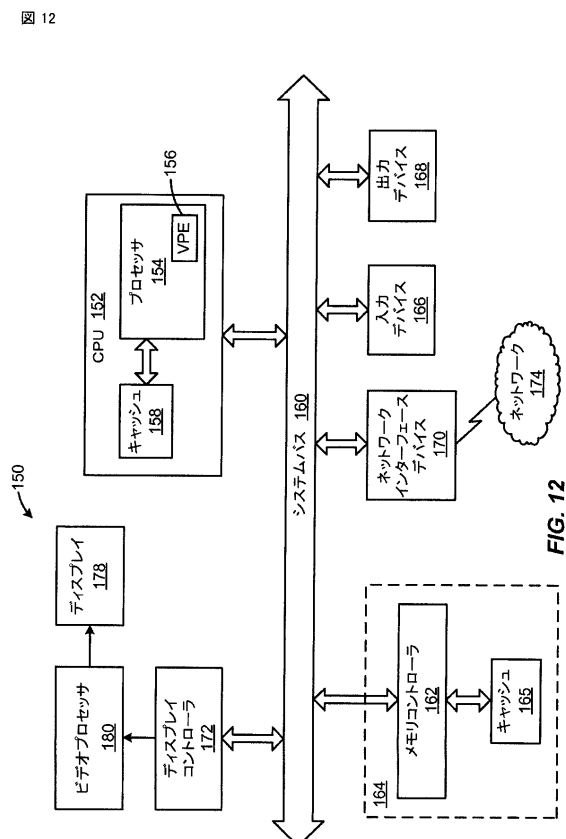


FIG. 11

【 図 1 2 】



【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2014/021782

A. CLASSIFICATION OF SUBJECT MATTER

INV. G06F17/14 G06F9/30
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

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, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	SCHOENES M ET AL: "A novel SIMD DSP architecture for software defined radio", MIDWEST SYMPOSIUM ON CIRCUITS AND SYSTEMS. CAIRO, EGYPT, DEC. 27 - 30, 2003; [MIDWEST SYMPOSIUM ON CIRCUITS AND SYSTEMS], PISCATAWAY, NJ, IEEE, US, vol. 3, 27 December 2003 (2003-12-27), page 1443, XP010866282, DOI: 10.1109/MWSCAS.2003.1562567 ISBN: 978-0-7803-8294-7	1-14, 20-31
Y	section II; figures 2, 3 ----- -/-	15-19

☒ Further documents are listed in the continuation of Box C.

☒ 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

"Z" document member of the same patent family

Date of the actual completion of the international search

11 May 2015

Date of mailing of the international search report

18/05/2015

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

Domingo Vecchioni, M

INTERNATIONAL SEARCH REPORT

International application No

PCT/US2014/021782

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	DERBY J H ET AL: "A high-performance embedded DSP core with novel SIMD features", PROCEEDINGS OF INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING (ICASSP'03) 6-10 APRIL 2003 HONG KONG, CHINA; [IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING (ICASSP)], IEEE, 2003 IEEE INTERNATIONAL CONFE, vol. 2, 6 April 2003 (2003-04-06), pages 301-304, XP010640941, DOI: 10.1109/ICASSP.2003.1202354 ISBN: 978-0-7803-7663-2 sections III and V, figure 3 -----	1-31
Y	WO 99/45462 A1 (SIEMENS AG [DE]; KNUTH ROBERT [DE]) 10 September 1999 (1999-09-10) figure 2 -----	15-19
A	Parhami B: "Computer Arithmetic - Algorithms and Hardware Designs", 2000, Oxford University Press, New York, XP055132227, ISBN: 978-0-19-512583-2 pages 128-133, 203, 204, 468-469, sections 8.2, 8.3, 12.6 figures 8.10, 8.13, 8.14, 12.19 -----	15-19
E	WO 2014/164367 A1 (QUALCOMM INC [US]) 9 October 2014 (2014-10-09) the whole document -----	1-31
E	WO 2014/164931 A2 (QUALCOMM INC [US]) 9 October 2014 (2014-10-09) the whole document -----	1-31

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2014/021782

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
WO 9945462	A1	10-09-1999	NONE	
WO 2014164367	A1	09-10-2014	US 2014281370 A1 WO 2014164367 A1	18-09-2014 09-10-2014
WO 2014164931	A2	09-10-2014	US 2014280407 A1 WO 2014164931 A2	18-09-2014 09-10-2014

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, 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, LT, 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、ラヘール

アメリカ合衆国、カリフォルニア州 9 2 1 2 1、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) 5B056 BB42 FF04

(54)【発明の名称】マルチモード基数2のX乗のバタフライベクトル処理回路を提供するためのプログラマブルなデータパス構成を有するベクトル処理エンジン、ならびに関連ベクトルプロセッサ、システム、および方法