

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7428787号
(P7428787)

(45)発行日 令和6年2月6日(2024.2.6)

(24)登録日 令和6年1月29日(2024.1.29)

(51)国際特許分類

F I

G 0 6 F 7/57 (2006.01) G 0 6 F 7/57 2 0 4

H 0 3 K 19/17704(2020.01) H 0 3 K 19/17704

請求項の数 33 (全121頁)

(21)出願番号	特願2022-516115(P2022-516115)	(73)特許権者	522096178
(86)(22)出願日	令和2年9月10日(2020.9.10)		コーナミ、インコーポレーテッド
(65)公表番号	特表2022-548046(P2022-548046 A)		アメリカ合衆国 9 5 0 0 8 カリフォルニア、キャンベル、オーチャード シテ
(43)公表日	令和4年11月16日(2022.11.16)		イー ドライブ 3 0 0 , スイート 1 3 1
(86)国際出願番号	PCT/US2020/050069	(74)代理人	100094112
(87)国際公開番号	WO2021/050643		弁理士 岡部 譲
(87)国際公開日	令和3年3月18日(2021.3.18)	(74)代理人	100106183
審査請求日	令和5年9月11日(2023.9.11)		弁理士 吉澤 弘司
(31)優先権主張番号	62/898,452	(74)代理人	100114915
(32)優先日	令和1年9月10日(2019.9.10)		弁理士 三村 治彦
(33)優先権主張国・地域又は機関	米国(US)	(74)代理人	100125139
(31)優先権主張番号	62/899,025		弁理士 岡部 洋
(32)優先日	令和1年9月11日(2019.9.11)	(74)代理人	100209808
	最終頁に続く		弁理士 三宅 高志
			最終頁に続く

(54)【発明の名称】 再構成可能プロセッサ回路アーキテクチャ

(57)【特許請求の範囲】

【請求項1】

第1の相互接続ネットワークと、
第2の相互接続ネットワークと、
前記第1の相互接続ネットワークに結合されたプロセッサと、
アレイ状に配置された複数の計算コアとを備える再構成可能プロセッサ回路であって、
前記複数の計算コアは、前記第1の相互接続ネットワークおよび前記第2の相互接続ネットワークに結合され、前記第2の相互接続ネットワークは前記複数の計算コアの隣接する計算コアに直接結合し、
各計算コアは、メモリ回路と、再構成可能演算回路とを備え、
前記再構成可能演算回路は、少なくとも1つの入力並べ替えキューと、前記少なくとも1つの入力並べ替えキューに結合された乗算器シフタコンバイナネットワークであって、シフタ回路と、前記シフタ回路に結合された複数の直列結合された加算器回路とを備え、乗算器積をシフトして浮動小数点積を基数32の指数を有する積に変換するように構成され、複数の単一命令多重データ(SIMD)積を合計してSIMDドット積を形成するように構成される前記乗算器シフタコンバイナネットワークと、アキュムレータ回路と、前記乗算器シフタコンバイナネットワークおよび前記アキュムレータ回路に結合された少なくとも1つの制御論理回路とを備える、再構成可能プロセッサ回路。

【請求項2】

前記再構成可能演算回路は、複数の動作モードを有する構成可能乗算器をさらに備え、

前記構成可能乗算器は、前記少なくとも1つの入力並べ替えキューと、前記乗算器シフトコンバイナネットワークとに結合され、

前記複数の動作モードは、固定小数点動作モードおよび浮動小数点動作モードを備え、

前記構成可能乗算器は、符号あり入力を処理するようにさらに構成可能な $2^7 \times 2^7$ 符号なし乗算器のネイティブ動作モードを有する、請求項1に記載の再構成可能プロセッサ回路。

【請求項3】

前記乗算器シフトコンバイナネットワークを1つまたは複数の隣接する再構成可能な演算回路に選択的に結合して、単一サイクルの 32×32 および 54×54 乗算、単精度 24×24 乗算、および単一命令多重データ(SIMD)ドット積を実行するように適合された第3の相互接続ネットワークをさらに含む、請求項2に記載の再構成可能プロセッサ回路。

10

【請求項4】

前記構成可能乗算器はさらに、4つの 8×8 乗算器、2つの 16×16 単一命令多重データ(SIMD)乗算器、1つの 32×32 乗算器、および1つの 54×54 乗算器になるように構成可能である、請求項2に記載の再構成可能演算回路。

【請求項5】

前記複数の計算コアの各計算コアは、

前記再構成可能演算回路、前記第1の相互接続ネットワークおよび前記第2の相互接続ネットワークに結合された複数の入力マルチプレクサと、

20

複数の入力レジスタであって、各入力レジスタは、前記複数の入力マルチプレクサの対応する入力マルチプレクサに結合される、複数の入力レジスタと、

前記再構成可能演算回路に結合された複数の出力マルチプレクサであって、各出力マルチプレクサは前記複数の入力レジスタの対応する入力レジスタに結合される、複数の出力マルチプレクサと、

複数の出力レジスタであって、各出力レジスタは、前記複数の出力マルチプレクサの対応する出力マルチプレクサと、前記第1の相互接続ネットワークと、前記第2の相互接続ネットワークとに結合されている、複数の出力レジスタとをさらに含む、請求項1に記載の再構成可能プロセッサ回路。

【請求項6】

30

複数のゼロ圧縮解除回路であって、各ゼロ圧縮解除回路は、前記複数の入力マルチプレクサの対応する入力マルチプレクサに結合される、複数のゼロ圧縮解除回路と、

複数のゼロ圧縮回路であって、各ゼロ圧縮回路は、前記複数の出力マルチプレクサの対応する出力マルチプレクサに結合される、複数のゼロ圧縮回路とをさらに含む、請求項5に記載の再構成可能プロセッサ回路

【請求項7】

データペイロードにすべてゼロを有するいくつかのデータパケットが、非ゼロのデータペイロードを有する次のデータパケットのサフィックスとして符号化される、請求項6に記載の再構成可能プロセッサ回路。

【請求項8】

40

前記第1の相互接続ネットワークは、FATツリー構成を有する階層型ネットワークであり、複数のデータルーティング回路を備える請求項1に記載の再構成可能プロセッサ回路。

【請求項9】

前記再構成可能プロセッサ回路は、前記プロセッサおよび前記複数の計算コアを使用して、任意およびすべてのRISC-Vプロセッサ命令を実行するように適合される、請求項1に記載の再構成可能プロセッサ回路。

【請求項10】

前記少なくとも1つの入力並べ替えキューは、複数の入力を格納するように適合され、

前記少なくとも1つの入力並べ替えキューは、前記複数の入力のシーケンスを並べ替え

50

、負および絶対値関数のための符号ビットを調整し、IおよびQデータ入力と奇数および偶数データ入力とをデインタリーブするように適合された入力並べ替え論理回路をさらに備える、請求項1に記載の再構成可能プロセッサ回路。

【請求項11】

前記再構成可能演算回路は、複数の再構成可能演算回路からの出力を受信し並び替えるために結合された少なくとも1つの出力並べ替えキューをさらに備える、請求項1に記載の再構成可能プロセッサ回路。

【請求項12】

前記再構成可能演算回路は複数の入力を有し、

前記複数の入力は第1のX入力、第2のY入力および第3のZ入力を含み、

前記少なくとも1つの制御論理回路が、比較回路、ブール論理回路、Z入力シフタ、指数論理回路、加算、飽和、丸め回路、およびこれらの組合せからなる群から選択された1つまたは複数の回路を含む、請求項1に記載の再構成可能プロセッサ回路。

【請求項13】

前記Z入力シフタは、浮動小数点Z入力値を基数32の指数値にシフトし、乗算器合計出力のスケールリングに一致するように32ビットの倍数だけシフトするように適合され、前記Z入力シフタが64、32、 2×16 および 4×8 ビットシフトまたは回転モードを含む複数の整数モードについてさらに適合される、請求項12に記載の再構成可能プロセッサ回路。

【請求項14】

前記ブール論理回路は、32ビット整数入力に対してAND、NAND、OR、NOR、XOR、XNOR、およびセクタ演算を実行するように適合されたAND-OR-INVERT論理ユニットを含む、請求項12に記載の再構成可能プロセッサ回路。

【請求項15】

前記比較回路は、入力データストリームから最小または最大のデータ値を抽出し、前記入力データストリームからインデックスを抽出し、2つの入力データストリームを比較し、2つの入力データストリームを交換し、前記2つの入力データストリームの前記最小値を第1の出力に置き、前記2つの入力データストリームの前記最大値を第2の出力に置き、データステアリングを実行し、アドレスシーケンスを生成し、等しいか、より大きいか、より小さいかの複数の比較フラグを生成するように適合される、請求項12に記載の再構成可能プロセッサ回路。

【請求項16】

単一の再構成可能演算回路が、

サイクルあたりの1つのIEEE単一または整数 27×27 乗算と、

2つの並列IEEE半精度、16ビットブレイン浮動小数点（「BFLOAT」）（BLOAT16）、または符号付きおよび符号なし16ビット整数値の16ビット整数（INT16）のサイクルあたりの乗算と、

4つの並列IEEE1/4精度、または符号付きおよび符号なし8ビット整数値（INT8）のサイクルあたりの乗算と、

2つの並列IEEE半精度の合計、BLOAT16またはINT16のサイクルあたりの乗算と、

4つの並列IEEE1/4精度の合計、または符号付きおよび符号なし8ビット整数値（INT）のサイクルあたりの乗算と、

1つの1/4精度、またはINT8のサイクルあたりの複素乗算と、融合加算と、累積と、

任意のビット数による64、32、 2×16 、または 4×8 ビットシフトと、

任意のビット数による64、32、 2×16 、または 4×8 ビット回転と、

32ビット単位のブール論理と、

データストリームの比較、最小値、または最大値と、

2つの演算対象のソートと、

これらの組合せからなる群から選択された少なくとも2つの数学的計算または関数を実行するように適合される、

請求項1に記載の再構成可能プロセッサ回路。

【請求項17】

対構成を有する2つの隣接するリンクされた再構成可能演算回路が、

符号付きおよび符号なし32ビット整数値 (INT32) についての1つの32ビット整数のサイクルあたりの乗算と、

符号付きおよび符号なし64ビット整数値 (INT64) についての、4つの32×32の部分積を加算する前記アキュムレータ回路を使用する、1つの64ビット整数の4サイクルシーケンスでの乗算と、

符号付きおよび符号なし24ビット整数値 (INT24) についてのIEEE単精度または2つの24ビット整数の合計の、サイクルあたりの乗算と、

4つの並列IEEE半精度の合計、16ビットブレイン浮動小数点 (「BFloat16」) (BFloat16)、または符号付きおよび符号なし16ビット整数値の16ビット整数 (INT16) のサイクルあたりの乗算と、

IEEEの8つの並列1/4精度の合計、または符号付きおよび符号なし8ビット整数値 (INT8) のサイクルあたりの乗算と

1つの半精度、またはINT16のサイクルあたりの複素乗算と、4つの乗算と、2つの加算と、融合加算と、累積と、これらの組合せからなる群から選択された少なくとも2つの数学的計算または関数を実行するように適合される、請求項1に記載の再構成可能プロセッサ回路。

【請求項18】

クアッド構成を有する4つのリンクされた再構成可能演算回路は、

符号付きおよび符号なし64ビット整数値 (INT64) についての2つの64ビット整数の4サイクルでの乗算と、

符号付きおよび符号なし32ビット整数値 (INT32) についての2つの32ビット整数のサイクルあたりの乗算と、

サイクル当たり2つのINT32乗算の合計と、

符号付きおよび符号なし24ビット整数値 (INT24) についての4つのIEEE単精度または24ビット整数の合計と、

8つの並列IEEE半精度の合計、16ビットブレイン浮動小数点 (「BFloat16」) (BFloat16)、または符号付きおよび符号なし16ビット整数値の16ビット整数 (INT16) のサイクルあたりの乗算と、

16つの並列IEEEの1/4精度の合計、または符号付きおよび符号なし8ビット整数値についての8ビット整数 (INT8) のサイクルあたりの乗算と、

1つの単精度または符号付きおよび符号なし24ビット整数値 (INT24) についてのサイクルあたりの複素乗算と、融合加算と、累積と、これらの組合せからなる群から選択される少なくとも2つの数学的計算または関数を実行するように適合される、請求項1に記載の再構成可能プロセッサ回路。

【請求項19】

第1の相互接続ネットワークと、

第2の相互接続ネットワークと、

前記第1の相互接続ネットワークに結合されたプロセッサと、

アレイ状に配置された複数の計算コアとを備える再構成可能プロセッサ回路であって、

前記複数の計算コアは、前記第1の相互接続ネットワークおよび前記第2の相互接続ネットワークに結合され、前記第2の相互接続ネットワークは前記複数の計算コアの隣接する計算コアに直接結合し、

各計算コアは、メモリ回路と、再構成可能演算回路とを備え、

前記再構成可能演算回路は、

複数の入力を格納するように適合された少なくとも1つの入力並べ替えキューであって

10

20

30

40

50

、前記再構成可能演算回路の前記複数の入力シーケンスと、前記複数の計算コアの隣接する再構成可能演算回路とを並べ替えるように適合された入力並べ替え論理回路をさらに含む、入力並べ替えキューと、

複数の動作モードを有する構成可能乗算器であって、前記構成可能乗算器は、前記少なくとも1つの入力並び替えキューに結合され、前記複数の動作モードは、固定小数点動作モードおよび浮動小数点動作モードを含み、前記構成可能乗算器は、符号付き入力を処理するようにさらに構成可能な 27×27 符号なし乗算器のネイティブ動作モードを有し、前記構成可能乗算器は、4つの 8×8 乗算器、2つの 16×16 単一命令マルチデータ(SIMD)乗算器、1つの 32×32 乗算器および1つの 54×54 乗算器になるようにさらに構成可能である、構成可能乗算器と、

10

前記構成可能乗算器に結合された乗算器シフトコンバイナネットワークであって、前記乗算器シフトコンバイナネットワークは、シフト回路と、前記シフト回路に結合された複数の直列結合加算器回路とを備える、乗算器シフトコンバイナネットワークと、

アキュムレータ回路と、

前記乗算器シフトコンバイナネットワークおよび前記アキュムレータ回路に結合された少なくとも1つの制御論理回路と、

前記複数の計算コアの前記再構成可能演算回路および前記隣接する再構成可能演算回路からの複数の出力を受信し並び替えるために結合された少なくとも1つの出力並び替えキューとを備える、再構成可能プロセッサ回路。

【請求項20】

20

前記乗算器シフトコンバイナネットワークに結合され、複数の再構成可能演算回路をマージし、倍精度乗算-加算、単精度単一サイクル複素乗算、FFTバタフライ、指数分解、乗算-累積、および論理演算を実行するように適合された第3の構成可能相互接続ネットワークをさらに含む、請求項19に記載の再構成可能プロセッサ回路。

【請求項21】

前記第3の構成可能相互接続ネットワークは、再構成可能演算回路の対構成としておよび再構成可能演算回路のクアド構成として、前記複数の計算コアの隣接する複数の再構成可能演算回路にリンクする複数の直接接続を含む、請求項20に記載の再構成可能プロセッサ回路。

【請求項22】

30

単一の再構成可能演算回路が、

サイクルあたり1つのIEEE単一または整数 27×27 乗算と、

2つの並列IEEE半精度、16ビットブレイン浮動小数点(「BFLOAT」)(BLOAT16)、または符号付きおよび符号なし16ビット整数値の16ビット整数(INT16)のサイクルあたりの乗算と、

4つのIEEE並列 $1/4$ 精度、または符号付きおよび符号なし8ビット整数値(INT8)のサイクルあたりの乗算と、

2つの並列IEEE半精度の合計、BLOAT16またはINT16のサイクルあたりの乗算と、

4つの並列IEEE $1/4$ 精度または符号付きおよび符号なし8ビット整数値(INT8)のサイクルあたりの乗算と、

40

1つの $1/4$ 精度、またはINT8のサイクルあたりの複素乗算と、融合加算と、累積と、

任意のビット数による64、32、 2×16 、または 4×8 ビットシフトと、

任意のビット数による64、32、 2×16 、または 4×8 ビット回転と、

32ビット単位のブール論理と、

データストリームの比較、最小値、または最大値と、

2つの演算対象のソートと、

これらの組合せからなる群から選択された少なくとも2つの数学的計算または関数を実行するように適合される、請求項21に記載の再構成可能プロセッサ回路。

50

【請求項 23】

前記対構成を有する 2 つの隣接するリンクされた再構成可能演算回路が、

符号付きおよび符号なし 32 ビット整数値 (INT 32) についての 1 つの 32 ビット整数のサイクルあたりの乗算と、

符号付きおよび符号なし 64 ビット整数値 (INT 64) についての、4 つの 32×32 の部分積を加算する前記アキュムレータ回路を使用する、1 つの 64 ビット整数の 4 サイクルシーケンスでの乗算と、

符号付きおよび符号なし 24 ビット整数値 (INT 24) についての IEEE 単精度または 2 つの 24 ビット整数の合計の、サイクルあたりの乗算と、

4 つの並列 IEEE 半精度の合計、16 ビットブレイン浮動小数点 (「BFloat16」) (BFloat16)、または符号付きおよび符号なし 16 ビット整数値の 16 ビット整数 (INT 16) のサイクルあたりの乗算と、

IEEE の 8 つの並列 $1/4$ 精度の合計、または符号付きおよび符号なし 8 ビット整数値 (INT 8) のサイクルあたりの乗算と

1 つの半精度、または INT 16 のサイクルあたりの複素乗算と、融合加算と、累積と、これらの組合せからなる群から選択された少なくとも 2 つの数学的計算または関数を実行するように適合される、請求項 21 に記載の再構成可能プロセッサ回路。

【請求項 24】

クアッド構成を有する 4 つのリンクされた再構成可能演算回路は、

符号付きおよび符号なし 64 ビット整数値 (INT 64) についての 2 つの 64 ビット整数の 4 サイクルでの乗算と、

符号付きおよび符号なし 32 ビット整数値 (INT 32) についての 2 つの 32 ビット整数のサイクルあたりの乗算と、

サイクル当たり 2 つの INT 32 乗算の合計と、

符号付きおよび符号なし 24 ビット整数値 (INT 24) についての 4 つの IEEE 単精度または 24 ビット整数の合計と、

8 つの並列 IEEE 半精度の合計、16 ビットブレイン浮動小数点 (「BFloat16」) (「BFloat16」)、または符号付きおよび符号なし 16 ビット整数値の 16 ビット整数 (INT 16) のサイクルあたりの乗算と、

16 つの並列 IEEE $1/4$ 精度の合計、または符号付きおよび符号なし 8 ビット整数値についての 8 ビット整数 (INT 8) のサイクルあたりの乗算と、

1 つの単精度または符号付きおよび符号なし 24 ビット整数値 (INT 24) についてのサイクルあたりの複素乗算と、融合加算と、累積と、これらの組合せからなる群から選択される少なくとも 2 つの数学的計算または関数を実行するように適合される、請求項 21 に記載の再構成可能プロセッサ回路。

【請求項 25】

前記乗算器シフタコンバイナネットワークは、乗算器積をシフトさせて、浮動小数点積を、基数 32 指数を有する積に変換するように、および複数の単一命令多重データ (SIMD) 積を合計して SIMD ドット積を形成するように適合される、請求項 19 に記載の再構成可能プロセッサ回路。

【請求項 26】

前記乗算器シフタコンバイナネットワークに結合される第 3 の構成可能相互接続ネットワークをさらに備え、

前記第 3 の構成可能相互接続ネットワークは、前記乗算器シフタコンバイナネットワークを前記複数の計算コアの隣接する再構成可能演算回路の 1 つまたは複数の乗算器シフタコンバイナネットワークに結合直接接続を備え、単一サイクル 32×32 および 54×54 乗算、単精度 24×24 乗算、および単一命令多重データ (SIMD) ドット積を実行する、請求項 19 に記載の再構成可能プロセッサ回路。

【請求項 27】

前記入力並べ替え論理回路は、否定および絶対値関数についての符号ビットを調整し、

10

20

30

40

50

I および Q データ入力および奇数および偶数データ入力をデインタリーブするようにさらに適合される、請求項 19 に記載の再構成可能プロセッサ回路。

【請求項 28】

前記複数の再構成可能演算回路のそれぞれの再構成可能演算回路は、複数の入力を有し、前記複数の入力、第 1 の X 入力、第 2 の Y 入力および第 3 の Z 入力を含み、

前記少なくとも 1 つの制御論理回路が、比較回路、ブール論理回路、Z 入力シフタ、指数論理回路、加算、飽和、丸め回路、およびこれらの組合せからなる群から選択された 1 つまたは複数の回路を含む、請求項 19 に記載の再構成可能プロセッサ回路。

【請求項 29】

前記 Z 入力シフタは、浮動小数点 Z 入力値を基数 32 の指数値にシフトし、乗算器合計出力のスケールリングに一致するように 32 ビットの倍数だけシフトするように適合され、前記 Z 入力シフタが 64、32、 2×16 および 4×8 ビットシフトまたは回転モードを含む複数の整数モードについてさらに適合される、請求項 28 に記載の再構成可能プロセッサ回路。

【請求項 30】

前記ブール論理回路は、32 ビット整数入力に対して AND、NAND、OR、NOR、XOR、XNOR、およびセクタ演算を実行するように適合された AND-OR-INVERT 論理ユニットを含む、請求項 28 に記載の再構成可能プロセッサ回路。

【請求項 31】

前記比較回路は、入力データストリームから最小または最大のデータ値を抽出し、前記入力データストリームからインデックスを抽出するように適合され、および、2 つの入力データストリームを比較し、前記 2 つの入力データストリームを交換し、前記 2 つの入力データストリームの前記最小値を第 1 の出力に置き、前記 2 つの入力データストリームの前記最大値を第 2 の出力に置き、等しいか、より大きいのか、より小さいかの複数の比較フラグを生成するように構成される、請求項 28 に記載の再構成可能プロセッサ回路。

【請求項 32】

前記乗算器シフタコンバイナネットワークは、複数の再構成可能演算回路の対構成における別の再構成可能演算回路からの積を加算し、前記複数の再構成可能演算回路のクアッド構成における別の半分の再構成可能演算回路からの積の合計を生成し、積を合計するために、クアッド構成における他の再構成可能演算回路からの Z 入力および Z 入力のスケールリングに一致するように 32 ビットの倍数だけシフトするように適合される、請求項 19 に記載の再構成可能なプロセッサ回路。

【請求項 33】

第 1 の相互接続ネットワークと、

第 2 の相互接続ネットワークと、

前記第 1 の相互接続ネットワークに結合されたプロセッサと、

アレイ状に配置された複数の計算コアとを備える再構成可能プロセッサであって、

前記複数の計算コアは、前記第 1 の相互接続ネットワークおよび前記第 2 の相互接続ネットワークに結合され、前記第 2 の相互接続ネットワークは前記複数の計算コアの隣接する計算コアに直接結合し、

各計算コアは、

前記第 1 の相互接続ネットワークおよび前記第 2 の相互接続ネットワークに結合された複数の入力マルチプレクサと、

複数の入力レジスタであって、各入力レジスタが前記複数の入力マルチプレクサの対応する入力マルチプレクサに結合される、複数の入力レジスタと、

複数の出力マルチプレクサであって、各出力マルチプレクサは、前記複数の入力レジスタの対応する入力レジスタに結合される、複数の出力マルチプレクサと、

複数の出力レジスタであって、各出力レジスタは、前記複数の出力マルチプレクサの対応する出力マルチプレクサと、前記第 1 の相互接続ネットワークと、前記第 2 の相互接続ネットワークとに結合される、複数の出力レジスタと、

10

20

30

40

50

複数のゼロ圧縮解除回路であって、各ゼロ圧縮解除回路は、前記複数の入力マルチプレクサの対応する入力マルチプレクサに結合される、複数のゼロ圧縮解除回路と、

複数のゼロ圧縮回路であって、各ゼロ圧縮回路は、前記複数の出力マルチプレクサの対応する出力マルチプレクサに結合される、複数のゼロ圧縮回路と、

メモリ回路と、

前記メモリ回路と、前記複数の入力レジスタと、前記複数の出力マルチプレクサに結合された再構成可能演算回路とを備え、

前記再構成可能演算回路は、

複数の入力を格納するように適合された少なくとも1つの入力並べ替えキューであって、前記再構成可能演算回路の前記複数の入力のシーケンスおよび前記複数の計算コアの隣接する再構成可能演算回路を並べ替えるように適合された入力並べ替え論理回路をさらに備える、少なくとも1つの入力並べ替えキューと、

複数の動作モードを有する構成可能乗算器であって、前記構成可能乗算器は、前記入力並び替えキューに結合され、前記複数の動作モードは、固定小数点動作モードおよび浮動小数点動作を含み、前記構成可能乗算器は、符号付き入力処理するようにさらに構成可能な 27×27 符号なし乗算器のネイティブ動作モードを有し、前記構成可能乗算器はさらに、4つの 8×8 乗算器、2つの 16×16 単一命令多重データ(SIMD)乗算器、1つの 32×32 乗算器、および1つの 54×54 乗算器になるように構成可能である、構成可能乗算器と、

前記構成可能乗算器に結合された乗算器シフタコンバイナネットワークとを備え、

前記乗算器シフタコンバイナネットワークは、

シフタ回路と

前記シフタ回路に結合された複数の直列結合加算器回路と、

アキュムレータ回路と、

乗算器シフタコンバイナネットワークおよびアキュムレータ回路に結合された少なくとも1つの制御論理回路と、

前記複数の計算コアの再構成可能演算回路及び前記隣接する再構成可能演算回路からの複数の出力を受信し並び替えるように結合された少なくとも1つの出力並び替えキューと、

前記乗算器シフタコンバイナネットワークを1つまたは複数の隣接する再構成可能な演算回路に選択的に結合して、単一サイクル 32×32 および 54×54 乗算、単精度 24×24 乗算、および単一命令多重データ(SIMD)ドット積を実行するように結合された第3の相互接続ネットワークとを含む、再構成可能プロセッサ回路。

【発明の詳細な説明】

【技術分野】

【0001】

[関連出願への相互参照]

【0002】

本出願は、「Reconfigurable Arithmetic Engine」という名称で2019年9月10日に出願された米国仮特許出願第62/898,452号の非仮出願であり、その利益と優先権を主張するものであり、「Reconfigurable Processor Circuit Architecture with an Array of Fractal Cores」という名称で2019年9月11日に出願された米国仮特許出願第62/899,025号の利益と優先権を主張するものであり、さらに、本出願と共通して譲渡される、2020年9月9日に出願された米国特許出願第17/015,950号および2020年9月9日に出願された米国特許出願第17/015,973号の優先権を主張するものであり、これらのすべては、本明細書にその全体が記載されているかのように、参照により本明細書に組み込まれる。

【背景技術】

【0003】

本発明は、一般に、構成可能および再構成可能計算回路に関し、より具体的には、算術

10

20

30

40

50

および論理計算のための電子回路を有する構成可能および再構成可能演算エンジンに関する。

【0004】

多くの既存のコンピューティングシステムは、ニューラルネットワーク計算、デジタル通貨、ブロックチェーンなどを含む数学的に集中的なアプリケーションのための計算速度が不十分であるなど、計算処理能力の大きな限界に達している。さらに、多くの既存のコンピューティングシステムは、過剰にエネルギー（または電力）を消費し、それに伴って放熱する。例えば、既存のコンピューティングソリューションは、人工知能、ニューラルネットワーク、暗号化、復号化、およびその他の重要なコンピューティングアプリケーションに対応するための高度なコンピューティングテクノロジーのニーズが増大するにつれて、ますます不適切になってきている。

10

【0005】

したがって、人工知能、ニューラルネットワーク計算、デジタル通貨、ブロックチェーン、暗号化、復号化、高速フーリエ変換（FFT）の計算、および機械学習を含むがこれらに限定されないような、数学的に集約的なアプリケーションのための高性能およびエネルギー効率の高いソリューションを提供することができるコンピューティングアーキテクチャに対する継続的なニーズがある。

【0006】

さらに、これらの様々なアプリケーションのいずれに対しても構成可能な、構成可能かつ再構成可能コンピューティングアーキテクチャに対する継続的なニーズがある。このような構成および再構成可能なコンピューティングアーキテクチャは、数百万または複数の処理コアなど、容易にスケーラブルであるべきであり、低レイテンシであるべきであり、計算的およびエネルギー効率的であるべきであり、リアルタイムでストリーミングデータを処理することが可能であるべきであり、選択されたアプリケーションのためにコンピューティングハードウェアを最適化するように再構成可能であるべきであり、大規模並列処理が可能であるべきである。

20

【0007】

本発明の多数の他の利点および特徴は、本発明およびその実施形態の以下の詳細な説明、請求の範囲および添付の図面から容易に明らかになるであろう。

【発明の概要】

30

【発明が解決しようとする課題】

【0008】

以下により詳細に説明するように、代表的な装置、システム、および方法は、人工知能、ニューラルネットワーク計算、デジタル通貨、暗号化、復号化、ブロックチェーン、高速フーリエ変換（FFT）の計算、および機械学習などを含むがこれらに限定されない、数学的に集約的なアプリケーションのための高性能およびエネルギー効率の高いソリューションを提供することができるコンピューティングアーキテクチャを提供する。

【0009】

さらに、本明細書に開示された再構成可能プロセッサは、装置およびシステムとして、これらの様々なアプリケーションのいずれに対しても構成することができ、いくつかのそのような例を以下により詳細に説明する。このような再構成可能プロセッサは、数百万の計算コアなどに容易に拡張可能であり、低レイテンシであり、計算的でありエネルギー効率がよく、リアルタイムでストリーミングデータを処理することができ、選択されたアプリケーションのために計算ハードウェアを最適化するように再構成可能であり、大規模並列処理が可能である。例えば、単一のチップ上で、再構成可能な複数のプロセッサを、相互接続ネットワークを使用してアレイ化して接続し、チップ当たり数百から数千の計算コアを提供することもできる。次に、このようなチップを複数配列して回路基板上に接続すると、基板当たり数千から数百万の計算コアが得られる。任意の選択された数の計算コアを再構成可能プロセッサ内に実装することができ、任意の数の再構成可能プロセッサを単一の集積回路上に実装することができ、任意の数のそのような集積回路を回路基板上に実

40

50

装することができる。このように、計算コアのアレイを有する再構成可能プロセッサは、任意の選択された程度（経路および熱放散などの他の制約の対象となるが、これに限定されるものではない）にスケラブルである。

【課題を解決するための手段】

【0010】

代表的な実施形態では、再構成可能演算回路は、複数の入力並べ替えキューと、前記複数の入力並べ替えキューに結合された乗算器シフタコンバイナネットワークと、アキュムレータ回路と、前記乗算器シフタコンバイナネットワークと前記アキュムレータ回路に結合された少なくとも1つの制御論理回路とを含む。

【0011】

代表的な実施形態では、そのような再構成可能演算回路は、複数の動作モードを有する構成可能な乗算器をさらに含むことができ、前記複数の入力並べ替えキューおよび前記乗算器シフタコンバイナネットワークに結合され、前記複数の動作モードは、固定小数点動作モードおよび浮動小数点動作を含み、前記構成可能な乗算器は、符号付き入力を処理するようにさらに構成可能な 27×27 符号なし乗算器のネイティブ動作モードを有する。前記構成可能な乗算器はさらに、4つの 8×8 乗算器、2つの 16×16 単一命令多重データ（SIMD）乗算器、1つの 32×32 乗算器、および1つの 54×54 乗算器になるように構成可能であり得る。たとえば、構成可能な乗数は、1つまたは複数の部分積を再割り当てして 32×32 乗数にするようにさらに構成可能にすることができる。

【0012】

代表的な実施形態では、乗算器シフタコンバイナおよび結合器ネットワークは、シフタ回路と、前記シフタ回路に結合された複数の直列結合された加算器回路とを含むことができる。そして、前記乗算器シフタコンバイナネットワークは、乗算器積をシフトさせて、浮動小数点積を、基数32の指数を有する積に変換するように、適合され得る。代表的な実施形態では、乗算器シフタコンバイナネットワークは、複数の単一命令多重データ（SIMD）積を合計してSIMDドット積を形成するように適合され得る。

【0013】

代表的な実施形態では、そのような再構成可能演算回路は、前記乗算器シフタコンバイナネットワークを1つまたは複数の隣接する再構成可能演算回路に選択的に結合して、単一サイクル 32×32 および 54×54 乗算、単精度 24×24 乗算、および単一命令多重データ（SIMD）ドット積を実行する構成可能相互接続ネットワークをさらに含むことができる。

【0014】

代表的な実施形態では、前記複数の入力並べ替えキューは、複数の入力を格納するように適合され、前記入力並べ替えキューが、前記複数の入力のシーケンスを並べ替え、ネゲートおよび絶対値関数のための符号ビットを調整するように構成された入力並べ替え論理回路をさらに含む。代表的な実施形態では、入力並べ替え論理回路は、IおよびQデータ入力と奇数および偶数データ入力とをデインタリーブするようにさらに適合されてもよい。

【0015】

代表的な実施形態では、そのような再構成可能演算回路は、複数の再構成可能演算回路からの出力を受信し、再構成するために結合された出力並べ替えキューをさらに含むことができる。代表的な実施形態では、アキュムレータ回路は、128ビットの桁上げ保存形式を有する単クロックサイクル固定浮動小数点アキュムレータであってもよい。

【0016】

代表的な実施形態では、再構成可能演算回路は複数の入力を有し、複前記複数の入力は第1のX入力、第2のY入力および第3のZ入力を含み、前記少なくとも1つの制御論理回路が、比較回路、ブール論理回路、Z入力シフタ、指数論理回路、加算、飽和、丸め回路、およびこれらの組合せからなる群から選択された1つまたは複数の回路を含む

【0017】

代表的な実施形態では、Z入力シフタは、浮動小数点Z入力値を基数32指数値にシフ

10

20

30

40

50

トし、乗算器和出力のスケールリングに一致するように32ビットの倍数だけシフトするように適合されてもよく、Z入力シフタが64、32、 2×16 および 4×8 ビットシフトまたは回転モードを有するシフタまたは回転子として使用される複数の整数モードを有する。

【0018】

前記ブール論理回路は、32ビット整数入力に対してAND、NAND、OR、NOR、XOR、XNOR、およびセクタ演算を実行するように適合されたAND-OR-INVERT論理ユニットを含むことができる。

【0019】

代表的な実施形態では、前記比較回路は、入力データストリームから最小または最大のデータ値を抽出し、前記入力データストリームからインデックスを抽出するように構成されてもよく、さらに、2つの入力データストリームを比較するように構成されてもよい。代表的な実施形態では、前記比較回路は、2つの入力データストリームを交換し、前記2つの入力データストリームの前記最小値を第1の出力に置き、前記2つの入力データストリームの前記最大値を第2の出力に置くように構成されてもよい。代表的な実施形態では、前記比較回路は、データステアリングを実行し、アドレスシーケンスを生成し、等しいか、より大きいのか、より小さいかの複数の比較フラグを生成するように構成されてもよい。

【0020】

アレイ内に配置された複数の再構成可能演算回路もまた開示されており、複数の再構成可能演算回路は、複数の入力を格納するように適合された入力並べ替えキューと、再構成可能な演算回路の複数の入力のシーケンスを並べ替えるように適合された入力並べ替え論理回路と、複数の再構成可能な演算回路の隣接する再構成可能演算回路とをさらに含む。また、入力並べ替えキューに結合された乗算器シフタコンバイナネットワークと、アキュムレータ回路と、前記乗算器シフタコンバイナネットワークならびに前記アキュムレータ回路に結合された少なくとも1つの制御論理回路とをさらに含む。さらに、複数の再構成可能演算回路の再構成可能演算回路および隣接する再構成可能演算回路からの出力を受信および並べ替えるために結合された出力並べ替えキューを含む。

【0021】

代表的な実施形態では、そのような再構成可能演算回路のアレイは、前記乗算器シフタコンバイナネットワークに結合され、複数の再構成可能演算回路をマージし、倍精度乗算-加算、単精度単一サイクル複素乗算、FFTバタフライ、指数分解、乗算-累積、および論理演算を実行する構成可能相互接続ネットワークをさらに含むことができる。例えば、構成可能相互接続ネットワークは、再構成可能演算回路の対構成としておよび複数の再構成可能演算回路のクワッド構成として、前記複数の計算コアの隣接する複数の再構成可能演算回路にリンクする複数の直接接続を含むことができる。

【0022】

代表的な実施形態では、そのような再構成可能演算回路のアレイにおいて、単一の再構成可能演算回路が、サイクルあたり1つのIEEE単一または整数 27×27 乗算と、2つの並列IEEE半(half)精度、16ビットブレイン浮動小数点(「BFLOAT」)(BLOAT16)、または符号付きおよび符号なし16ビット整数値の16ビット整数(INT16)のサイクルあたりの乗算と、IEEEの4つの並列 $1/4$ (quarter)精度、または符号付きおよび符号なし8ビット整数値(INT8)のサイクルあたりの乗算と、2つの並列IEEE半精度の合計、BLOAT16またはINT16のサイクルあたりの複素乗算と、1つの $1/4$ 精度、またはINT8のサイクルあたりの複素乗算と、融合加算と、累積と、任意のビット数による64、32、 2×16 、または 4×8 ビットシフトと、任意のビット数による64、32、 2×16 、または 4×8 ビット回転と、32ビットビット単位のブール論理と、データストリームの比較、最小値、または最大値と、2つの演算対象のソートと、これらの組合せからなる群から選択された少なくとも2つの数学的計算または関数を実行するように適合されてもよい。

【0023】

代表的な実施形態では、そのような再構成可能演算回路のアレイにおいて、対構成を有する2つの隣接するリンクされた再構成可能演算回路が、符号付きおよび符号なし32ビット整数値（INT32）についての1つの32ビット整数のサイクルあたりの乗算と、符号付きおよび符号なし64ビット整数値（INT64）についての、4つの32×32の部分積を加算する前記累積回路を使用する、1つの64ビット整数の4サイクルシーケンスでの乗算と、符号付きおよび符号なし24ビット整数値（INT24）についてのIEEE単精度または2つの24ビット整数の合計の、サイクルあたりの乗算と、4つの並列IEEE半精度の合計、16ビットブレイン浮動小数点（「BFLOAT」）（BLOAT16）、または符号付きおよび符号なし16ビット整数値の16ビット整数（INT16）のサイクルあたりの乗算と、8つのIEEE並列1/4精度の合計、または符号付きおよび符号なし8ビット整数値（INT8）のサイクルあたりの乗算と1つの1/4精度、またはINT16のサイクルあたりの複素乗算と、融合加算と、累積と、これらの組合せからなる群から選択された少なくとも2つの数学的計算または関数を実行するように適合され得る。

【0024】

代表的な実施形態では、このような再構成可能演算回路のアレイにおいて、クアド構成を有する4つのリンクされた再構成可能演算回路は、符号付きおよび符号なし64ビット整数値（INT64）についての2つの64ビット整数の4サイクルでの乗算と、符号付きおよび符号なし32ビット整数値（INT32）についての2つの32ビット整数のサイクルあたりの乗算と、サイクル当たり2つのINT32乗算の合計と、符号付きおよび符号なし24ビット整数値（INT24）についての4つのIEEE単精度または24ビット整数の合計と、8つの並列IEEE半精度の合計、16ビットbrain浮動小数点（「BFLOAT」）（「BFLOAT」）、または符号付きおよび符号なし16ビット整数値の16ビット整数（INT16）のサイクルあたりの乗算と、16つの並列IEEEの1/4精度の合計、または符号付きおよび符号なし8ビット整数値についての8ビット整数（INT8）のサイクルあたりの乗算と、1つの単精度または符号付きおよび符号なし24ビット整数値（INT24）についてのサイクルあたりの複素乗算と、融合加算と、累積と、これらの組合せからなる群から選択される少なくとも2つの数学的計算または関数を実行するように適合され得る。

【0025】

代表的な実施形態では、そのような再構成可能演算回路のアレイにおいて、複数の再構成可能演算回路の各再構成可能演算回路は、さらに、複数の動作モードを有する構成可能乗算器であって、前記構成可能乗算器は、前記複数の入力並べ替えキューに結合され、前記乗算器シフタコンバイナネットワークに結合され、前記複数の動作モードは、固定小数点動作モードおよび浮動小数点動作を含み、前記構成可能乗算器は、符号付き入力を処理するようにさらに構成可能な27×27符号なし乗算器のネイティブ動作モードを有し得る。たとえば、4つの8×8乗算器、2つの16×16単一命令多重データ（SIMD）乗算器、1つの32×32乗算器、および1つの54×54乗算器になるように構成可能であり得る。たとえば、構成可能な乗数は、1つまたは複数の部分積を再割り当てして32×32の乗数にするようにさらに構成可能にすることができる。

【0026】

代表的な実施形態では、そのような再構成可能演算回路のアレイにおいて、乗算器シフタコンバイナネットワークは、シフタ回路と、シフタ回路に結合された複数の直列結合された加算器回路とを含むことができる。例えば、乗算器シフタコンバイナネットワークは、乗算器積をシフトして、浮動小数点積を基数32指数を有する積に変換し、複数の単一命令多重データ（SIMD）積を合計してSIMDドット積を形成するように適合されてもよい。代表的な実施形態では、そのような再構成可能演算回路のアレイにおいて、乗算器シフタコンバイナネットワークは、さらに、前記乗算器シフタコンバイナネットワークを前記複数の計算コアの隣接する再構成可能演算回路の1つまたは複数の乗算器シフタおよびコンバイナネットワークに結合して単一サイクル32×32および54×54乗算、

単精度 24×24 乗算、および単一命令多重データ (SIMD) ドット積を実行する複数の直接接続を含むことができる。

【0027】

代表的な実施形態では、そのような再構成可能演算回路のアレイにおいて、前記乗算器シフタコンバイナネットワークは、再構成可能演算回路の対構成における別の再構成可能演算回路からの積を加算し、再構成可能演算回路のクアッド構成における別の半分の再構成可能な演算回路からの積の合計を生成するように構成されてもよい。例えば、乗算器シフタコンバイナネットワークは、積を合計するために、クアッド構成における他の再構成可能演算回路からの Z 入力および Z 入力のスケーリングに一致するように 32 ビットの倍数だけシフトするように構成されてもよい。

【0028】

代表的な実施形態では、再構成可能演算回路は、複数のデータ入力であって、第 1 の X データ入力を含む複数のデータ入力と、第 2 の Y データ入力および第 3 の Z データ入力と、複数のデータ出力と、前記複数のデータ出力と接続され、出力データを受信して並び替える複数の出力並べ替えキューと、前記複数のデータ入力と接続され、入力データを格納するように適合された複数の入力並べ替えキューであって、前記入力データのシーケンスを並び替えるように適合された入力並べ替え論理回路をさらに含む入力並べ替えキューと、前記複数の入力並べ替えキューと接続される構成可能乗算器であって、複数の動作モードを有する構成可能乗算器とを含み、前記複数の動作モードは、固定小数点動作モードおよび浮動小数点動作モードを含み、前記構成可能な乗算器は、符号付き入力を処理するようにさらに構成可能な 27×27 符号なし乗算器のネイティブ動作モードを有し、前記構成可能乗算器はさらに、4 つの 8×8 乗算器、2 つの 16×16 単一命令多重データ (SIMD) 乗算器、1 つの 32×32 乗算器、および 1 つの 54×54 乗算器になるように構成可能であり、前記再構成可能演算回路は、前記構成可能乗算器に接続される乗算器シフタコンバイナネットワークであって、シフタ回路と、前記シフタ回路に接続された複数の直列結合された加算器回路と、前記乗算器シフタコンバイナネットワークを 1 つまたは複数の隣接する再構成可能演算回路に接続して、単一サイクル 32×32 および 54×54 乗算、単精度 24×24 乗算、および単一命令多重データ (SIMD) ドット積を実行する複数の直接接続を含む乗算器シフタコンバイナネットワークと、1 クロックサイクルの固定および浮動小数点桁上げ保存アキュムレータ回路と、乗算器シフタコンバイナネットワークおよびアキュムレータ回路に接続された複数の制御論理回路とをさらに含み、前記複数の制御論理回路が、入力データストリームから最小または最大データ値を、入力データストリームからインデックスを抽出するように適合された比較回路を含み、前記複数の制御論理回路は、2 つの入力データストリームを比較し、2 つの入力データストリームを交換して、2 つの入力データストリームの最小値を最初の出力に配置し、2 つの入力データストリームの最大値を 2 番目の出力に配置し、データステアリングを実行し、アドレスシーケンスを生成し、等しい、より大きい、より小さい、の比較フラグを生成するようにさらに適合され、前記ブール論理回路は、32 ビット整数入力に対して AND、NAND、OR、NOR、XOR、XNOR、およびセクタ演算を実行するように適合された AND-OR-INVERT 論理ユニットを含み、Z 入力シフタは、浮動小数点 Z 入力値を基数 32 の指数値にシフトし、乗算器和出力のスケーリングに一致するように 32 ビットの倍数だけシフトするように構成され、前記 Z 入力シフタが 64、32、 2×16 および 4×8 ビットシフトまたは回転モードを有するシフタまたは回転子として使用される複数の整数モードと、指数論理回路と、加算、飽和、および丸め回路とを有する。

【0029】

再構成可能プロセッサ回路も開示され、代表的な実施形態は、第 1 の相互接続ネットワークと、前記第 1 の相互接続ネットワークに結合されたプロセッサと、アレイ状に配置された複数の計算コアであって、前記複数の計算コアは、前記第 1 の相互接続ネットワークと、前記複数の計算コアの隣接する計算コアと直接結合する第 2 の相互接続ネットワークとに結合され、各計算コアは、メモリ回路と再構成可能演算回路とを含む再構成可能プロ

10

20

30

40

50

セッサ回路であって、前記再構成可能前記演算回路は、入力並べ替えキューと、前記入力並べ替えキューに接続された乗算器シフタコンバイナネットワークと、前記乗算器シフタコンバイナネットワークおよび前記アキュムレータ回路に結合された少なくとも1つの制御論理回路を含む。

【0030】

代表的な実施形態では、再構成可能な演算回路は、複数の動作モードを有する構成可能乗算器であって、前記構成可能乗算器は、前記複数の入力並べ替えキューに結合され、前記乗算器シフタコンバイナネットワークに結合され、前記複数の動作モードは、固定小数点動作モードおよび浮動小数点動作モードを含み、前記構成可能乗算器は、符号付き入力を処理するようにさらに構成可能な 27×27 符号なし乗算器のネイティブ動作モードを有する。

10

【0031】

代表的な実施形態では、再構成可能プロセッサ回路は、さらに、乗算器シフタコンバイナネットワークを1つまたは複数の隣接する再構成可能演算回路に選択的に結合して、単一サイクルの 32×32 および 54×54 乗算、単精度 24×24 乗算、および単一命令多重データ(SIMD)ドット積を実行する第3の相互接続ネットワークをさらに含んでもよい。

【0032】

代表的な実施形態では、構成可能乗算器はさらに、4つの 8×8 乗算器、2つの 16×16 単一命令多重データ(SIMD)乗算器、1つの 32×32 乗算器、および1つの 54×54 乗算器になるように構成可能である。

20

【0033】

代表的な実施形態では、複数の計算コアの各計算コアは、前記再構成可能演算回路、前記第1の相互接続ネットワークおよび第2の相互接続ネットワークに結合された複数の入力マルチプレクサと、複数の入力レジスタであって、各入力レジスタは、前記複数の入力マルチプレクサの対応する入力マルチプレクサに結合される、複数の入力レジスタと、前記再構成可能演算回路に結合された複数の出力マルチプレクサであって、各出力マルチプレクサは前記複数の入力レジスタの対応する入力レジスタに結合される、複数の出力マルチプレクサと、複数の出力レジスタであって、各出力レジスタは、前記複数の出力マルチプレクサの対応する出力マルチプレクサと、前記第1の相互接続ネットワークと、前記第2の相互接続ネットワークとに結合されている、複数の出力レジスタとをさらに含むことができる。

30

【0034】

代表的な実施形態では、複数の計算コアの各計算コアは、複数のゼロ圧縮解除回路であって、各ゼロ圧縮解除回路は、前記複数の入力マルチプレクサの対応する入力マルチプレクサに結合される、複数のゼロ圧縮解除回路と、複数のゼロ圧縮回路であって、各ゼロ圧縮回路は、前記複数の出力マルチプレクサの対応する出力マルチプレクサに結合される、複数のゼロ圧縮回路とをさらに含むことができる。

【0035】

代表的な実施形態では、データペイロードにすべてゼロを有する多数のデータパケットが、非ゼロのデータペイロードを有する次のデータパケットのサフィックスとして符号化される。

40

【0036】

代表的な実施形態では、前記第1の相互接続ネットワークは、FATツリー構成を有する階層型ネットワークであってもよく、複数のデータルーティング回路を備えてもよい。

【0037】

代表的な実施形態では、前記再構成可能プロセッサ回路は、前記プロセッサおよび前記複数の計算コアを使用して、任意およびすべてのRISC-Vプロセッサ命令を実行するように適合される。

【0038】

50

別の代表的な実施形態では、第1の相互接続ネットワークと、前記第1の相互接続ネットワークに結合されたプロセッサと、アレイ状に配置された複数の計算コアであって、前記複数の計算コアは、前記第1の相互接続ネットワークと、前記複数の計算コアの隣接する計算コアと直接結合する第2の相互接続ネットワークとに結合され、各計算コアは、メモリ回路と再構成可能演算回路とを含む再構成可能プロセッサ回路であって、前記再構成可能前記演算回路は、複数の入力を格納するように結合された複数の入力並べ替えキューであって、前記再構成可能演算回路の前記複数の入力のシーケンスと、前記複数の計算コアの隣接する再構成可能演算回路とを並べ替えるように適合された入力並べ替え論理回路をさらに含む、入力並べ替えキューと、複数の動作モードを有する構成可能乗算器であって、前記構成可能乗算器は、前記複数の入力並び替えキューに結合され、前記複数の動作モードは、固定小数点動作モードおよび浮動小数点動作を含み、前記構成可能乗算器は、符号付き入力を処理するようにさらに構成可能な 27×27 符号なし乗算器の固有動作モードを有し、前記構成可能乗算器は、4つの 8×8 乗算器、2つの 16×16 単一命令マルチデータ(SIMD)乗算器、1つの 32×32 乗算器および1つの 54×54 乗算器になるようにさらに構成可能である、構成可能乗算器と、構成可能乗算器に結合された乗算器シフトコンバイナネットワークとを含み、前記乗算器シフトコンバイナネットワークは、シフト回路と、前記シフト回路に結合された複数の直列結合加算器回路と、アキュムレータ回路と、前記乗算器シフトコンバイナネットワークおよび前記アキュムレータ回路に結合された少なくとも1つの制御論理回路と、前記複数の計算コアの前記再構成可能演算回路及び前記隣接する再構成可能演算回路からの出力を受信し並び替えるために結合された複数の出力並び替えキューとを含むことができる。

【0039】

別の代表的な実施形態では、再構成可能プロセッサ回路は、第1の相互接続ネットワークと、前記第1の相互接続ネットワークに結合されたプロセッサと、アレイ状に配置された複数の計算コアであって、前記複数の計算コアは、前記第1の相互接続ネットワークと、前記複数の計算コアの隣接する計算コアと直接結合する第2の相互接続ネットワークとに結合され、各計算コアは、メモリ回路と再構成可能演算回路とを含む再構成可能プロセッサ回路であって、前記再構成可能演算回路は、複数の入力を格納するように結合された複数の入力並べ替えキューであって、前記再構成可能演算回路の前記複数の入力のシーケンスと、前記複数の計算コアの隣接する再構成可能演算回路とを並べ替えるように適合された入力並べ替え論理回路をさらに含む、入力並べ替えキューと、複数の動作モードを有する構成可能乗算器であって、前記構成可能乗算器は、前記複数の入力並び替えキューに結合され、前記複数の動作モードは、固定小数点動作モードおよび浮動小数点動作を含み、前記構成可能乗算器は、符号付き入力を処理するようにさらに構成可能な 27×27 符号なし乗算器の固有動作モードを有し、前記構成可能乗算器は、4つの 8×8 乗算器、2つの 16×16 単一命令マルチデータ(SIMD)乗算器、1つの 32×32 乗算器および1つの 54×54 乗算器になるようにさらに構成可能である、構成可能乗算器と、構成可能乗算器に結合された乗算器シフトコンバイナネットワークとを含み、前記乗算器シフトコンバイナネットワークは、シフト回路と、前記シフト回路に結合された複数の直列結合加算器回路と、アキュムレータ回路と、前記乗算器シフトコンバイナネットワークおよび前記アキュムレータ回路に結合された少なくとも1つの制御論理回路と、前記複数の計算コアの前記再構成可能演算回路および前記隣接する再構成可能演算回路からの出力を受信し並び替えるために結合された複数の出力並び替えキューと、前記乗算器シフトコンバイナネットワークを1つまたは複数の隣接する再構成可能演算回路に選択的に接続させ、単一サイクルの 32×32 および 54×54 乗算、単精度 24×24 乗算、および単一命令多重データ(SIMD)ドット積を実行する第3の相互接続ネットワークとを含む。

【0040】

本発明の多数の他の利点および特徴は、本発明およびその実施形態の以下の詳細な説明、請求の範囲および添付の図面から容易に明らかになるであろう。

【0041】

本発明の目的、特徴および利点は、添付の図面と併せて考慮すると、以下の開示を参照することによって、より容易に理解されるであろう。ここで、種々の図において同一の構成要素を識別するために同様の参照番号が使用され、アルファベット文字を有する参照番号は、種々の図において選択された構成要素の実施形態の追加のタイプ、インスタンス化または変形を識別するために使用される。

【図面の簡単な説明】

【0042】

【図1】フラクタルコアのアレイを有する再構成可能プロセッサのブロック図である。

【図2】フラクタルコアとRAE回路の高レベルブロック図である。

【図3】隣接するフラクタルコア間の複数の直接接続を示すフラクタルコアのアレイのブロック図である。

【図4A】フラクタルコアの詳細なブロック図である。

【図4B】フラクタルコアの詳細なブロック図である。

【図5】再構成可能演算エンジン（「RAE」）回路の例示的または代表的な第1の実施形態を示すブロック図である。

【図5A】RAE回路の例示的または代表的な第2の実施形態を示すハイレベルのブロック図である。

【図6】専用接続を有する複数の例示的または代表的なRAE回路を示す高レベルブロック図である。

【図7】 24×24 として接続されたネイティブ 27×27 構成のRAE乗数を示す。

【図8】 32×32 乗算器を2つの 16×16 SIMD乗算器として使用する方法を示す。

【図9】2つの 16×16 SIMD乗算器に対する 27×27 乗算器の修正を示す。

【図10】可動部分積を有する乗算器の構造を示す。

【図11】4つの 8×8 SIMD乗算器としてプルーニングされた 32×32 乗算器を使用することを示す。

【図12】2つのRAE乗算器のシフトされた和によって形成される 32×32 乗算を示し、そのうちの1つは修正される。

【図13】4つの修正されていないRAE乗算器のシフトされた和によって形成される 54×54 の乗算を示す。

【図14】符号なし乗算器を使用する符号付き乗算のための符号付き補正回路を示す。

【図15】各符号付き整数モードに対する入力、符号補正、および出力のアラインメントを示す。

【図16】選択的否定も行うことができる符号付き補正回路を示す。

【図17】縮小された32ビットSIMD拡張命令を持つ 27×27 乗算器のドット図である。

【図18】第1レイヤダダ（Dadda）加算器を有する再構成されたドット図である。

【図19】フレキシブル乗算器のための第2レイヤダダツリーのドット図である。

【図20】フレキシブル乗算器のための第3レイヤダダツリーのドット図である。

【図21】フレキシブル乗算器のためのダダツリーの複数の残りの層のドット図である。

【図22】4:2圧縮器とレイヤ5のセグメントとの等価回路を示し、レイヤ5とレイヤ6とを1段階で実行するための4:2圧縮器の使用を示す。

【図23】乗算後乗算器シフトコンバイナネットワーク310の論理およびブロック図である。

【図24】アキュムレータに対する乗数積のアラインメントをモードごとに示す。

【図25】乗算後のシフトをモードごとに示す図である。

【図26】レーンシフトおよび第1のコンプレッサ回路の詳細を示すブロック図である。

【図27】レーン桁上げブロッキングのための4:2コンプレッサへの追加論理を示すブロック図である。

【図28】乗算器シフトコンバイナネットワークの第1の代替実施形態を示すブロック図

10

20

30

40

50

である。

【図 29】乗算器シフトコンバイナネットワークの第 2 の代替実施形態を示すブロック図である。

【図 30】Z-入力回転／シフト論理データ経路を示すブロック図である。

【図 31】モードごとの Z-入力シフト構成を示す図である。

【図 32】シフトネットワークの構成を示すブロック図である。

【図 33】Z-入力論理のモードによるビットアラインメントを示す。

【図 34】IEEE 単精度から基数 32 への浮動小数点フォーマット変換を示すブロック図である。

【図 35】、指数論理回路を示すブロック図である。

10

【図 36】Z-シフトおよび乗算器シフトコンバイナネットワークのための超過シフト論理を示すブロック図である。

【図 37】2つの入力×ファンアウト-2 (2×2) ゲートを複数有する 3 ビットインデックス対 8 ビットバー回路を示すブロック図であり、1 はバッファである。

【図 38】完全加算器を使用してバーの XOR 差をインデックスに変換するタリー回路を示すブロック図である。

【図 39】2つの入力×ファンアウト-2 (2×2) ゲートを複数有する 8 ビットバー対 3 ビットインデックス回路を示すブロック図である。

【図 40】乗算器およびシフト／コンバイナ指数論理を示すブロック図である。

【図 41】Z-入力指数論理を示すブロック図である。

20

【図 42】アキュムレータを示すブロック図である。

【図 43】アキュムレータを示す回路図である。

【図 44】先行符号と $N \times 32$ シフト回路を示す回路図である。

【図 45】SIMD 分割のために論理が追加された深さ $\log_2(n)$ を有するバー回路構造へのタリーを示す回路図である。

【図 46】プール論理段階を示す回路図である。

【図 47】最小／最大ソートおよび比較回路を示す高レベル回路およびブロック図である。

【図 48】最小／最大ソートおよび比較回路の比較器を示す詳細な回路およびブロック図である。

【図 49】デコーダ回路を示す回路図である。

30

【図 50】比較回路を使用したインデックス適用によるストリーミング最小／最大を示す詳細な回路およびブロック図である。

【図 51】比較回路を使用した 2 入力ソートアプリケーションを示す詳細な回路およびブロック図である。

【図 52】比較回路を使用したデータ置換アプリケーションを示す詳細な回路およびブロック図である。

【図 53】比較回路を用いたヒステリシスアプリケーションによる閾値を示す詳細な回路およびブロック図である。

【図 54】比較回路を使用したフラグトリガイイベントアプリケーションを示す詳細な回路およびブロック図である。

40

【図 55】比較回路を使用した閾値トリガ事象適用を示す詳細な回路およびブロック図である。

【図 56】比較回路を使用したデータステアリングアプリケーションを示す詳細な回路およびブロック図である。

【図 57】比較回路を使用するモジュロ N 計数アプリケーションを示す詳細な回路およびブロック図である。

【図 58】コーナーターンアドレスの導出を示すブロック図である。

【図 59】FFT ビット逆コーナーターンアドレスの導出を示すブロック図である。

【図 60】RAE 入力並べ替えキューの論理回路構造を示す詳細な回路およびブロック図である。

50

【図 6 1】入出力並べ替えキューのためのシーケンサ論理回路構造を示す詳細な回路およびブロック図である。

【図 6 2】混合基数アルゴリズムを使用した F F T の組み合わせを示す。

【図 6 3】基数 2 F F T カーネルの実行のための R A E 対を示す詳細な回路およびブロック図である。

【図 6 4】完全な回転子のために構成された R A E 3 0 0 の対を示す詳細な回路およびブロック図である。

【図 6 5】基数 4 F F T (バタフライ) カーネルを実行するための R A E 回路クアドを示す詳細な回路およびブロック図である。

【図 6 6】F F T カーネルを実行するための複数の R A E 3 0 0 カスケード対を示す詳細な回路およびブロック図である。

10

【図 6 7】文字列マッチングのユースケースを示す図である。

【図 6 8】再構成可能プロセッサと共に使用される代表的なデータパケットの図である。

【図 6 9】データパケットで利用される代表的なデータペイロードタイプの図である。

【図 7 0】ルーティングコントローラのブロック図および回路図である。

【図 7 1】サフィックス制御回路のブロック図である。

【図 7 2】ゼロ圧縮回路のブロックおよび回路図である。

【図 7 3】代表的なゼロ圧縮データパケットシーケンスの図である。

【図 7 4】ゼロ解凍回路のブロック図および回路図である。

20

【発明を実施するための形態】

【0043】

本発明は、多くの異なる形の実施形態の影響を受けやすいが、図面に示されており、本明細書では、本開示が本発明の原理の例示と考えられるべきであり、本発明を図示された特定の実施形態に限定することを意図していないことを理解した上で、その具体的な例示の実施形態について詳細に説明する。この点に関して、本発明と一致する少なくとも 1 つの実施形態を詳細に説明する前に、本発明は、その適用において、構造の詳細および上および下に記載された構成要素の配置、図面に示された構成要素の配置、または実施例に記載された構成要素の配置に限定されないことを理解されたい。本発明に合致する方法および装置は、他の実施形態が可能であり、様々な方法で実施および実施されることが可能である。また、本明細書で使用される用語および用語、ならびに以下に含まれる要約は、説明のためのものであり、限定するものとみなされるべきではないことを理解されたい。

30

【0044】

1. 再構成可能プロセッサ 100

図 1 は、フラクタルコア 200 のアレイを有する再構成可能プロセッサ 100 のブロック図である。図 2 は、フラクタルコア 200 および R A E 回路 300 の高レベルブロック図である。図 3 は、隣接する計算コア 200 間に複数の直接接続を形成する第 2 の相互接続ネットワーク 220 を示すフラクタルコア 200 のアレイのブロック図である。図 4 は、フラクタルコア 200 の詳細なブロック図である。図 1 ~ 4 を参照すると、集積回路またはチップなどの再構成可能プロセッサ 100 は、プロセッサ回路 130 と、複数の計算（または「フラクタル」）コア 200（すなわち、図中では「F C」と略記することができ、本明細書では「フラクタルコア」200 とも呼ぶ計算コア回路）とを備える。複数の計算（フラクタル）コア 200 は、典型的にはアレイ状に配列され、複数のルーティング（スイッチング）回路 225 および典型的には階層的な（例えば、F A T ツリーとして実装されているようなものであり、そのような相互接続ネットワークはすべて同等であり、開示の範囲内にあると考えられる）他の接続を有する第 1 の相互接続ネットワーク 120 を介してプロセッサ回路 130 に結合される。第 2 の相互接続ネットワーク 220 も図示されており、これは、第 2 の相互接続ネットワーク 220 の北（N）、南（S）、東（E）及び西（W）接続として図示されているように、隣接する（例えば最近接）計算コア 200 の入力と出力との間に複数の直接、経路指定されていないまたはスイッチされていない接続を提供する。第 1 の相互接続ネットワーク 120 は、以下により詳細に説明する通

40

50

り、計算コア200のアレイ間のデータパケットルーティングのために複数のレベルを有し、第2の相互接続ネットワーク220は、様々な計算コア200間に複数の直接接続を提供する。サフィックスのためのビットの位置を含む種々のデータパケットのフォーマットは、図68および69を参照して以下に説明される。

【0045】

プロセッサ回路130は、汎用プロセッサ（例えば、RISC-Vプロセッサ）として実装または具体化されてもよく、またはより限定されてもよく、Cコードを処理するための様々な計算論理および複数の状態機械などの制御論理回路を含んでもよい。例えば、プロセッサ回路130は、計算論理および1つまたは複数の状態機械（例えば、乗算器および／または除算器などの構成要素が省略または除去されている、高度に「軽量化された」RISC-Vプロセッサ）として実施されてもよい。プロセッサ回路130は、典型的には、プログラムカウンタ（「PC」）160と、命令デコーダ165と、再帰的Cコードのような複数のフラクタルコア200によって処理されていないCコードを処理するための種々の状態機械および他の制御論理回路とを含む。複数の計算コア200は、自己相似であるので「フラクタル」コアと呼ばれ、再構成可能プロセッサ100は、全体的に再構成可能プロセッサとしてだけでなく、大規模並列再構成可能アクセラレータ集積回路としても機能する複数のフラクタル計算コア200に「砕かれて（fractured）」いる。再構成可能プロセッサ100はまた、オフチップおよび他の複数のネットワーク通信のための入出力インターフェース140、オプションの演算論理ユニット（「ALU」）135、オプションのメモリコントローラ170、およびオプションのメモリ（および／またはレジスタ）155を含む。オプションのメモリコントローラ170および／またはオプションのメモリ（および／またはレジスタ）155は、メモリサブシステム175（図3に示す）として提供することもできる。例えば、再構成可能プロセッサ100は、RISC-Vプロセッサ全体を置き換えることができるだけでなく、大規模な計算高速化を提供することもできる。

【0046】

再構成可能プロセッサ100は、人工知能、ニューラルネットワーク計算、デジタル通貨、暗号化、復号化、ブロックチェーン、高速フーリエ変換（FFT）の計算、および機械学習を含むような数学的に集中的なアプリケーションのための高性能およびエネルギー効率の良いソリューションを提供するが、これらに限定されるものではない。

【0047】

さらに、再構成可能プロセッサ100は、これらの様々なアプリケーションのいずれに対しても構成することができ、いくつかのそのような例を以下により詳細に説明する。このような再構成可能プロセッサ100は、数百万の計算コア200などに容易に拡張可能であり、低レイテンシを有し、計算的でありエネルギー効率がよく、リアルタイムでストリーミングデータを処理することができ、選択されたアプリケーションのために計算ハードウェアを最適化するように再構成可能であり、大規模並列処理が可能である。例えば、単一のチップ上で、複数の再構成可能プロセッサ100を配列し、相互接続ネットワーク120を用いて接続して、チップ当たり数百から数千の計算コア200を提供することもできる。次に、このようなチップを複数配列して回路基板上に接続し、基板当たり数千から数百万の計算コア200を得ることができる。任意の選択された数の計算コア200を再構成可能プロセッサ100内に実装することができ、任意の数の再構成可能プロセッサ100を単一の集積回路上に実装することができ、任意の数のそのような集積回路を回路基板上に実装することができる。このように、計算コア200のアレイを有する再構成可能プロセッサ100は、任意の選択された程度（degree）にスケラブルである（経路および熱放散などの他の制約の対象となるが、これに限定されるものではない）。図3に示すような代表的な実施形態では、16個の計算コア200が、各再構成可能プロセッサ100に対してプロセッサ回路130ごとに実装され、再構成可能プロセッサ100は、例えば、限定されるものではないが、IC全体にタイリングされる。

【0048】

2. 計算コア 200

図2を参照すると、計算コア200は、再構成可能演算エンジン300または再構成可能演算エンジン回路300（本明細書では「RAE」300またはRAE回路300と呼ぶ）と呼ばれる再構成可能演算回路300を含む。RAE回路300は、（概してRAE入力マルチプレクサ105を介して）第1の複数の入力マルチプレクサ205に結合され、および複数の出力マルチプレクサ110に結合される。第1の複数の入力マルチプレクサ205および複数の出力マルチプレクサ110は、第1の相互接続ネットワーク120および第2の相互接続ネットワーク220に結合される。RAE回路300はまた、メモリ150と、オプションの制御インターフェース回路250とに結合される。制御インターフェース回路250は、例えば、RAE回路300の構成および／または命令を記憶するためのオプションとしての構成記憶装置またはメモリ180を含むことができ、オプションのレジスタ、オプションのプログラムカウンタ、およびオプションの命令デコーダも含むことができ、別のオプションとして、Cコードを処理するための種々の状態機械および他の制御論理回路を含むことができるがこれに限定されない。メモリ150は、演算対象のデータ（operand data）を記憶することができ、オプションとして、RAE回路300による実行のために計算コア200についての構成または他の命令を記憶することもできる。代替的に、RAE 300による実行のための計算コア200についての構成または他の命令は、制御インターフェース回路250に含まれ得るレジスタ（例えば、構成記憶装置またはメモリ180）に格納され得る。代表的な実施形態では、オプションとして、後述するように、構成記憶装置またはメモリ180がRAE回路300内に分散される。計算コア200は、図4を参照して説明した他の構成要素も含むことができる。代表的な実施形態では、メモリ150は、例えば、SRAMを使用して実装されるが、これに限定されない。入力マルチプレクサ205および出力マルチプレクサ110を介してデータ選択を提供するために使用されるルーティングコントローラ820の代表的な実施形態が、図70を参照して以下に示され、説明される。

【0049】

RAE回路300は、SIMDモードを含む種々の整数および浮動小数点モードを含む浮動小数点または整数演算およびブール論理を用いてストリーミングデータを処理するように主に設計されたデータフローアーキテクチャであり、関連データの受信時に実行される。これには、例外フラグを設定したり、データフローをゲートで制御するために使用したり、比較結果に基づいてデータを置換したりできる比較論理が追加されている。さらに、以下により詳細に説明するように、RAE回路300は、対で、4つ（例えば、これに限定されないが、図6に示される2行、2列）のグループで、および4つ以上のRAE回路300のグループでグループ化され得る。これには、（乗算器シフタコンバイナネットワーク310の）乗算器加算器ツリーを接続する特別な専用の選択可能な有線接続（複数のバス）360、445が含まれ、グループ内のRAE回路300を、例えば、これに限定されないが、より大きな加算器、複素乗算器、および一般積和使用のために組み合わせることを可能にする。本明細書で言及される様々なバスのすべては、特定のバスまたは相互接続ネットワーク120、220、295に必要なまたは望ましい（データパケットまたは制御ワードの対応する可変ビット数を送信するための）「N」ビット幅であると考えることができる。

【0050】

複数のRAE回路300間のこれらの専用の選択可能な有線接続（複数のバス）360、445は、構成可能な第3の相互接続ネットワーク295を形成して、複数のRAE回路300を複数のRAE回路対400および複数のRAE回路クアッド450にマージして、例えば、4つのリンクされたRAE回路300を単精度単一サイクル複素乗算として使用したり、複数のFFTバタフライ演算を実行したり、指数分解を実行したりするような、倍精度の乗算－加算、乗算－累積および後述する他の複数のアプリケーションにも適用可能な論理演算を実行するが、これに限定されない。RAE回路300は、図2、図5および図6を参照して以下により詳細に説明される。

【0051】

図4を参照すると、計算コア200は、(図4に簡略化して示す)RAE回路300およびメモリ150、ならびに一般に入力マルチプレクサ205(および/または複数のRAE入力マルチプレクサ105)、出力マルチプレクサ110、および、オプションとして、(構成記憶装置またはメモリ180および上述の他のコンポーネント(命令デコーダなど)を有することができる)制御インターフェース250を含む。図4には、(図5および図5Aに示す)RAE回路300のサフィックス(および/または条件付きフラグ)入力380およびサフィックス(および/または条件付きフラグ)出力405が別々に示されていない。代表的な実施形態では、複数の入力マルチプレクサ205は、第1の入力マルチプレクサ205A、第2の入力マルチプレクサ205B、および第3の入力マルチプレクサ205Cを含み、これらは、第1の相互接続ネットワーク120および第2の相互接続ネットワーク220から入力を受け取るように結合される。したがって、計算コア200のそれぞれの、(第2の相互接続ネットワーク220の直接接続を介して)隣接するそれぞれの計算コア200から、および(第1の相互接続ネットワーク120を介して)隣接しない複数の計算コア200、プロセッサ回路130、および他のコンポーネントからデータを受信するように結合され、これらのすべては、それぞれ、第1、第2、および第3の入力マルチプレクサ205A、205B、205Cのそれぞれに供給される。第1、第2、および第3の入力マルチプレクサ205A、205B、205Cのそれぞれに対する動的選択制御(個別には図示せず)は、構成記憶装置またはメモリ180に格納され得る構成などの複数の構成および/または複数の命令から提供され得る。第1、第2、および第3の入力マルチプレクサ205A、205B、205Cのそれぞれからの出力は、図4に示されるように、それぞれ(入力レジスタ230A、230B、230Cとして示される)対応する入力レジスタ230を使用するなどして、メモリ150、RAE回路300のための第2の複数の(RAE)入力マルチプレクサ105、または出力マルチプレクサ110などの他の構成要素に供給される前に、レジスタ段階にあってもよい。代表的な実施形態では、複数の入力レジスタ230は、典型的にはダブルバッファされる。

【0052】

代表的な実施形態では、計算コア200は、第1の出力マルチプレクサ110A、第2の出力マルチプレクサ110B、および第3の出力マルチプレクサ110Cも含み、これらは、RAE回路300、メモリ150、および第1、第2、および第3の入力マルチプレクサ205A、205B、205Cから入力を受け取り、出力を第1の相互接続ネットワーク120および第2の相互接続ネットワーク220に提供するように結合される。したがって、各計算コア200は、(第2の相互接続ネットワーク220の直接接続を介して)隣接する計算コア200のそれぞれと、(第1の相互接続ネットワーク120を介して)隣接しない複数の計算コア200およびプロセッサ回路130にデータを提供するように結合され、これらのすべては、第1、第2および第3の出力マルチプレクサ110A、110B、110Cのそれぞれから入力を受け取る。第1、第2、および第3の出力マルチプレクサ110A、110B、110Cのそれぞれについての動的選択制御(個別には図示せず)は、構成記憶装置またはメモリ180に格納され得る構成などの複数の構成および/または複数の命令から提供され得る。第1、第2、および第3の出力マルチプレクサ110A、110B、110Cのそれぞれからの出力は、図4に示されるように、それぞれ(出力レジスタ242A、242B、242Cとして示される)対応する出力レジスタ242を使用するなど、これらの他の構成要素に提供される前に、レジスタ段階にあってもよい。代表的な実施形態では、複数の出力レジスタ242は、典型的にはダブルバッファされる。

【0053】

代表的な実施形態では、図4に示すように、計算コア200には、複数の別々に独立したデータ経路があり、(バス241に沿って延在する)第1のデータ経路240(「左分配器」)、(バス243に沿って延在する)第2のデータ経路245(「右分配器」)、および(バス247に沿って延在する)第3のデータ経路255(「アグリゲータ(a

10

20

30

40

50

gregator) 」) として示され、それぞれが、構成記憶装置またはメモリ 180 に記憶され得る複数の構成または複数の命令を介するなど、互いに別々で独立した制御を有する。また、各計算コア 200 内では、これらのデータ経路のそれぞれは、一方の側の第 1、第 2、および第 3 の入力マルチプレクサ 205 A、205 B、205 C のそれぞれと、他方の側の第 1、第 2、および第 3 の出力マルチプレクサ 110 A、110 B、110 C のそれぞれとの間に、それぞれ 1 つ以上の経路スルー接続を含むことに留意されたい。例えば、第 1 の入力マルチプレクサ 205 A からのデータは、第 1 の出力マルチプレクサ 110 A、第 2 の出力マルチプレクサ 110 B、および第 3 の出力マルチプレクサ 110 C のいずれかまたはすべてにそれぞれ直接渡されてもよく、第 2 の入力マルチプレクサ 205 B からのデータは、第 1 の出力マルチプレクサ 110 A、第 2 の出力マルチプレクサ 110 B、および第 3 の出力マルチプレクサ 110 C のいずれかまたはすべてにそれぞれ直接渡されてもよく、第 3 の入力マルチプレクサ 205 C からのデータは、第 1 の出力マルチプレクサ 110 A、第 2 の出力マルチプレクサ 110 B、第 3 の出力マルチプレクサ 110 C のいずれかまたはすべてに直接渡されてもよい。また、RAE 回路 300 内にはいくつかの独立したデータ経路があり、メモリ 150 内には別の独立したデータ経路がある。これらの様々なデータ経路の全ては、異なる複数の計算を同時に、および別々に、かつ互いに独立して実行するために並列に使用することができる。代表的な実施形態では、第 3 のデータ経路 255 (「アグリゲータ」) は、複数のソースからのデータを集約するために利用され、例えば、出力データパケットのストリームを生成したり、複数のソースからのデータをインターリーブしたりするが、これらに限定されない。また、第 1 のデータ経路 240、第 2 のデータ経路 245 および第 3 のデータ経路 255 のそれぞれからのまたはそれらの中のデータを (RAE 入力マルチプレクサ 105 A、105 B、および 105 C を介して) RAE 回路 300 およびメモリ 150 のそれぞれに提供し、RAE 回路 300 およびメモリ 150 のそれぞれからのデータを第 1 のデータ経路 240、第 2 のデータ経路 245 および第 3 のデータ経路 255 のそれぞれに提供してもよい。

【0054】

複数の構成および複数のプログラム (例えば、複数の構成、複数の命令、複数の命令シーケンス) はまた、より集中化されたプログラムまたは構成記憶装置 (構成記憶装置やメモリ 180 など) を利用するのではなく、メモリ 150 (例えば SRAM プログラム 262) 内を含む、計算コア 200 内でローカルに (且つ別々に且つ独立して) 提供されてもよい。例えば、プログラム記憶装置 (またはメモリ) 264 および 266 が、第 1 のデータ経路 240 および第 2 のデータ経路 245 (およびオプションとして第 3 のデータ経路 255 (個別に図示せず)) の各々に設けられ、RAE 回路 300、メモリ 150、および RAE 入力マルチプレクサ 105 の入力選択に対して 2 つの別々のプログラムを提供する。また、例えば、いくつかの出力プログラム記憶装置 (またはメモリ) 272、274、および 276 が、第 1 のデータ経路 240、第 2 のデータ経路 245、および第 3 のデータ経路 255 のそれぞれに対する複数の構成または複数の命令の一部として、第 1、第 2、および第 3 の出力マルチプレクサ 110 A、110 B、110 C のそれぞれにそれぞれ提供される。第 1 のデータ経路 240、第 2 のデータ経路 245、および第 3 のデータ経路 255 は、「ゼロ圧縮」を実施することもでき、この場合、データストリーム内の比較的長いゼロのストリングは、直接送信されるのではなく、送信のために符号化される (そしてそれにより圧縮される)。

【0055】

メモリ 150、第 1 のデータ経路 240、第 2 のデータ経路 245、および第 3 のデータ経路 255 のいずれかからの入力データは、RAE 入力マルチプレクサ 105 A、105 B、および 105 C を介して RAE 回路 300 に供給され、より具体的には、それぞれ RAE 回路 300 の第 1 (「X」) 入力 365、第 2 (「Y」) 入力 370、および第 3 (「Z」) 入力 375 に供給される。RAE 回路 300 からの出力結果は、第 1 (「X」) 出力 420、第 2 (「Y」) 出力 415、第 3 (「Z」) 出力 410 を介して、(バス 303 を介して) メモリ 150、第 1 のデータ経路 240、第 2 のデータ経路 245、お

よび第3のデータ経路255に供給され、(バス303を介して)出力マルチプレクサ110に供給され、また、バス303を介して、およびRAE入力マルチプレクサ105A、105B、および105Cを介して、様々なRAE入力365、370、375のいずれかにフィードバックされる。RAE回路300、第1のデータ経路240、第2のデータ経路245、および第3のデータ経路255のいずれかからの入力データは、メモリ150への記憶のために、メモリ書き込みマルチプレクサ268(メモリ入力マルチプレクサ268Aおよび268Bとして図示)および(RAEメモリシステム152の)RAM書き込みインターフェース290を介して提供され、RAE回路300、第1のデータ経路240、第2のデータ経路245、および第3のデータ経路255のいずれかによってメモリ150から読み取られるデータは、図示のように、メモリ読み取り(またはロード)マルチプレクサ287を使用して選択され(RAEメモリシステム152の)RAM読み取りインターフェース297を使用して(バス301上で)提供されてもよい。また、RAEメモリシステム152は、オプションとして、トラッキングカウンタ292、ライトポインタストア294、およびリードポインタストア296を含んでもよい。

【0056】

3. RAE回路300、RAE回路対400、およびRAE回路クアッド450

図2、5、および6を参照すると、RAE回路300は、入力並べ替えキュー350と、乗算器シフトコンバイナネットワーク310(乗算器シフトコンバイナネットワーク310またはシフトコンバイナネットワーク310とも呼ばれる)と、アキュムレータ315と、1つまたは複数の制御論理回路275とを備える。入力並べ替えキュー350は、データ入力(第1(「X」)入力365、第2(「Y」)入力370、第3(「Z」)入力375)に(およびRAE回路対400の隣接するRAE回路300のデータ入力に)結合され、入力並べ替えキュー350は、乗算器シフトコンバイナネットワーク310および一つまたは複数の制御論理回路275に結合され、これらは、アキュムレータ315にも結合される。乗算器シフトコンバイナネットワーク310は、シフト回路425と、複数の直列結合加算器(桁上げ保存構成)430、435、440を含む加算器ツリーとを備え、従って、シフト及び加算に加えて、乗算の複数のモードを実行することが可能である(また、単一命令多重データ(「SIMD」)ドット積を計算するためにも利用される)。図5に示される代表的な実施形態では、追加の(構成可能な)乗算器305は、別々の構成要素として、またはシフトコンバイナネットワーク310の一部として、RAE回路300に含まれてもよく、全体として含まれてもよい。乗算器シフトコンバイナネットワーク310はまた、第3の相互接続ネットワーク295の専用通信線360、445(例えば、配線)を介して、(RAE回路対400およびRAE回路クアッド450の)隣接するRAE回路300の乗算器シフトコンバイナネットワーク310に結合される。

【0057】

複数の別々の独立したデータ経路280、281、282、283がRAE回路300内で利用され、複数の入力並べ替えキュー350から乗算器シフトコンバイナネットワーク310を通る第1のデータ経路280と、複数の入力並べ替えキュー350から複数の制御論理回路275を通る第2および第3のデータ経路281、282と、複数の制御論理回路275から乗算器シフトコンバイナネットワーク310およびアキュムレータ315を通る第4のデータ経路283と、第1のRAE回路300の複数の乗算器シフトコンバイナネットワーク310、RAE回路クアッド450のその(第1の)RAE回路対400の第2のRAE回路300、およびRAE回路クアッド450の別の(第2の)RAE回路対400の第3のRAE回路300のいずれか2つの間およびこれらの任意の群の間の第3の相互接続ネットワーク295(通信線360、445)を通る第5の双方向データ経路284と、RAE回路対400の隣接する複数のRAE回路300間の複数の入力並べ替えキュー350、および/またはRAE回路対400の隣接するRAE回路300から受信される追加の入力通信線395(図5に示す)を共有することによって作成されるオプションの第6のデータ経路285と、(図6に示す)RAE回路対400の隣接する複数のRAE回路300間で出力並べ替えキュー355を共有することによって生成

される第7のデータ経路286とを有する。この複数の別々であり、独立した複数のデータ経路を使用して、RAE回路300の様々な構成要素は、それぞれ、別々にかつ独立して、異なる計算、ビット操作、論理機能などを実行することができ、RAE回路300内のあらゆるリソースの重要な利用を提供する。また、入力並べ替えキュー350からの任意の出力(582、584、586)は、図5に示されたものに加えて、または図5に示されたものの代わりに、RAE回路300の任意の様々な構成要素に提供され得ることに留意されたい。

【0058】

図5は、RAE回路300の例示的または代表的な第一の実施形態を示すブロック図である。図5Aは、RAE回路300Aの例示的または代表的な第2の実施形態のソフトウェアプログラミング図を示す簡略化された高レベルのブロック図であり、入力並べ替えキュー350および出力並べ替えキュー355の追加の位置または回路構造を示し、サフィックス(および/または条件付きフラグ)入力380および/またはサフィックス制御回路390からのサフィックス情報の分布を示すために使用される。なお、図5Aには示されていないが、RAE回路300Aにおいては、RAE回路300の構成要素は除去または削除されていない。RAE回路300Aは、RAE回路300の変形例であり、上述したように入力並べ替えキュー350と出力並べ替えキュー355との配置(回路配置)のみがRAE回路300と異なっており、それ以外は、RAE回路300と全く同様に構成され、機能も同一である。したがって、別段の指定がない限り、RAE回路300へのいかなる参照も、RAE回路300Aを意味し、これを含み、RAE回路300Aは、本明細書においては、別の方法で個別には扱われない。

【0059】

図5および5Aを参照すると、RAE回路300は、複数の入力並べ替えキュー350と、(乗算器305を含むことができる)乗算器シフトコンバイナネットワーク310と、(乗算器シフターコンバイナネットワーク310の一部として含まれない場合に)構成可能(またはフレキシブルな)乗算器305(であって複数の構成可能な動作モードを有する乗算器305)と、アキュムレータ315と、オプションとしての出力並べ替えキュー355とを含み、制御論理回路275は、(1)min/maxソートおよび比較回路320(より全体的には「比較回路」320とも呼ばれる)(本明細書では「min」を「最小」と呼び、「max」を「最大」と呼ぶ)と、(2)ブール論理回路325と、(3)Z入力シフト回路330と、(4)指数論理回路335と、(5)(オプションとしてRAE回路300に含まれる場合)サフィックス制御回路390と、(6)最後に追加する飽和および丸め回路340と、(7)オプションとしてのビット反転回路345との内の1つまたは複数を含む。オプションとして、RAE回路300は、さらに、第1のデータ選択(ステアリング)マルチプレクサ387、第2のデータ選択(ステアリング)マルチプレクサ392、および第3のデータ選択(ステアリング)マルチプレクサ397などの様々なデータステアリング構成要素を含んでもよい。第1のマルチプレクサ387、第2のマルチプレクサ392および第3のマルチプレクサ397は、RAE回路300内のデータルーティング選択のための複数の構成および/または複数の命令を使用して構成される。また、図4に示すように、RAE回路300は、一般に、第1(「X」)入力365、第2(「Y」)入力370、第3(「Z」)入力375、サフィックス(および/または条件付きフラグ)入力380、および第1(「X」)出力420、第2(「Y」)出力415、第3(「Z」)出力410、およびサフィックス(および/または条件付きフラグ)出力405を含む。乗算器シフトコンバイナネットワーク310は、シフト回路425(単一命令多重データ(「SIMD」)ドット積を実行することもできる)と、直列に結合されて加算器「ツリー」を形成する複数の加算器430、435および440とを含む。乗算器シフトコンバイナネットワーク310はまた、図6を参照して後述するように、通信ライン360が、一对のRAE回路300内の別の隣接するRAE回路300に選択的に結合可能または接続可能であり、通信ライン445が、RAE回路クワッド450内の別の隣接するRAE回路300に選択的に結合可能または接続可能である、専用

の通信ライン 360 および 445 を含む。

【0060】

また、制御論理回路 275 を含む回路 320、325、330、335、340 および 345 のうちの 1 つまたは複数の、異なる方法で組み合わせまたは実装することができ、すべてが制御論理回路 275 に含まれる必要はないことに留意されたい。例えば、限定されるものではないが、比較回路 320 のソートは、入力並べ替えキュー 350 またはブール論理回路 325 内で実行することもでき、ビット反転回路 345 のビット反転は、比較回路 320、ブール論理回路 325、または複数の入力並べ替えキュー 350 によって実行することもできる。

【0061】

代表的な実施形態のオプションとして、RAE 回路 300 は、種々の構成要素が複数の構成、複数の命令、および／または複数の他の制御ワードまたはビットを受信するために、上述した制御インターフェース回路 250 または他の構成および／または命令ストアにも結合され、明確にするために、これらの別個の接続は、図 5 に別々には示されておらず、乗算器 305、乗算器シフトコンバイナネットワーク 310、アキュムレータ 315、最小／最大ソートおよび比較回路 320、ブール論理回路 325、Z シフト回路 330、指数論理回路 335、サフィックス制御回路 390、最終的な加算、飽和および丸め回路 340、入力並べ替えキュー 350、出力並べ替えキュー 355、および任意に、ビット反転回路 345、第 1 のマルチプレクサ 387、第 2 のマルチプレクサ 392、第 3 のマルチプレクサ 397、ならびに他のマルチプレクサおよび構成可能な構成要素、これらはすべて、制御インターフェース回路 330 または他の構成および／または複数の命令記憶装置に結合されて、様々な動作モードを選択し、RAE 回路 335 内および RAE 回路 390 間でデータをルーティングするなど、このような構成、命令および／または他の制御を受け取る。

【0062】

代表的な実施形態では、乗算器 305 は、桁上げ保存加算器（例えば、乗算器 305 について別々に示されていないが、乗算器シフトコンバイナネットワーク 310 のシフト 425 および加算器回路 430、435、440 と同様にまたは同一に実施されるシフトレジスタおよび加算器回路を含む）として実装されるが、固定小数点モードと浮動小数点モードの両方を有するように構成可能である。種々の異なる構成は、種々の部分積の移動／再配置によって達成され、説明され、次いで、それらは、当技術分野において公知であるか、または公知となった任意の種類の桁上げ保存加算器を使用して一緒に加算され、それらのいずれかおよびすべてが同等であり、開示の範囲内にあると考えられる。乗算器 305 は、符号付き入力を処理するための複数の拡張（追加回路）を備えた 27×27 符号なし乗算器としての「ネイティブモード」を有する。乗算器 305 は、4 つの 8×8 または 2 つの 16×16 SIMD 乗算器になるように構成および再構成可能である。これは、部分積の一部を再割り当てして、非対角部分積を除去した 32×32 の刈り込まれた乗算器として乗算器を配置することによって達成される。乗算器 305 の第 3 の構成は、再構成された乗算器 305 がネイティブモード乗算器 305 と対になって 2 つの RAE 回路 300 を用いて 32×32 の乗算器を形成することができるように部分積を再構成する。

【0063】

乗算器 305 の後には、（乗算器 305 から出力される）積をシフトして浮動小数点積を（シフト 425 を使用して）基数 32 指数を有するシステムに変換する乗算器シフトコンバイナネットワーク 310 が続く。乗算器シフトコンバイナネットワーク 310 は、SIMD 積を合計してドット積を形成することもできる。この乗算器シフトコンバイナネットワーク 310 はまた、スケーリングされた第 3（Z）入力を製品に追加する。さらに、他の RAE 回路 300 からの積を対で、および加算器 430、435、440 を使用して RAE 回路クアド 450 の 1 対のまたは残りの半分の隣接する RAE 回路 300 からの積の合計で追加することができる。必要に応じて積を合計するために、RAE 回路クアド 450 内の Z 入力および他の RAE 回路 300 からの入力のスケーリングを整合させる

ために、シフタ425によって32ビットの倍数による追加のシフトが行われる。加算ネットワーク（または「加算器ツリー」）としての複数の加算器430、435および440は、隣接するRAE回路300を結合して、単一サイクル32×32および54×54乗算、ならびに最大4項（SIMDモードの場合は8項または16項）の単精度24×24およびSIMDドット積を実行することを可能にする。

【0064】

Z入力シフタ330は、浮動小数点Z入力値をシフトして基数32指数を有するシステムに変換し、また必要に応じて32ビットの倍数だけシフトして、乗算器和出力（乗算器シフタコンパイナネットワーク310の）のスケールに整合する。整数モードの場合、Z入力シフタ回路330は、64、32、2×16および4×8ビットシフト／回転モードを有するシフタまたは回転子として使用される。

【0065】

アキュムレータ315は、単クロックサイクル浮動小数点アキュムレータとして実現される。アキュムレータ315は、単一レーンおよび2つの浮動小数点および1、2または4レーンINTEGER引数に対して、固定および浮動小数点乗算累算をサポートする。アキュムレータ315ハードウェアは、128ビットおよび64ビットセグメントのための追加の浮動小数点指数制御（4レーンSIMD浮動小数点はアキュムレータ315において整数として扱われる）を備えた桁上げ保存形式の128ビット加算器である（そしてアキュムレータ315ハードウェアは、公知の桁上げ保存アキュムレータとして具体化され得る）。

【0066】

ブール論理回路325は、Z入力のシフト／回転後に32ビットINT入力に対してAND、NAND、OR、NOR、XOR、XNOR、およびセレクト操作を実行するためにZ入力の浮動小数点アラインメントシフトネットワークに結合するAND-OR-INVERT論理ユニットを含む。

【0067】

最小／最大ソートおよび比較回路320は、入力ストリームからインデックスとともに最小値または最大値を抽出するか、または2つの入力ストリームを比較して、対の最小値を一方の出力に、最大値を他方の出力に常に置くようにストリームを交換するように設計されている（2つの引数ソート）。最小／最大ソートおよび比較回路320は、等しいか、より大きいのか、より小さいかの比較フラグを生成することもできる。この最小／最大ソートおよび比較回路320は、複数のレーンを独立してソートしたり、ストリームから最小または最大を抽出したりすることができるように、SIMD演算をサポートする。

【0068】

複数の入力並べ替えキュー（input reorder queue）350を使用する入力並べ替えは、最大4つの入力の履歴を再シーケンスし、IおよびQまたは奇数／偶数サンプルをデインタリーブするために、XおよびZ入力（365、375）間で交換することを可能にするが、これに限定されない。追加の論理は、X、Y、およびZ入力のデータソース（365、370：375）と、X、Y、およびZ出力（420、415、410）のデータシンクを選択する。また、回路内のキー位置に論理を追加して、否定および絶対値関数の符号ビットを任意に調整し、Y入力の入力セレクトに論理を追加して、Xの符号に基づく条件付き乗算をサポートする。入力並べ替えキュー350は、図5に示すように、入力（365、370 375、380）とRAE回路300の他の構成要素との間に配置されてもよいし、（図5Aに示すように）入力マルチプレクサ105と入力（365、370 375、380）との間に配置されてもよい。出力並べ替えキュー355は、以下により詳細に説明するように、入力並べ替えキュー350と同様に実装される。出力並べ替えキュー355は、図5に示すように、出力マルチプレクサ110と出力（405、410 415、420）との間に配置されてもよいし、出力（405、410 415、420）とRAE回路300の他の構成要素（図5Aに示すように）との間に配置されてもよい。入力並べ替えキュー350および出力並べ替えキュー355は、通常、

10

20

30

40

50

R A E回路対400の隣接する複数のR A E回路300の間で共有される。別の代表的な実施形態では、出力並べ替えキュー355は複数のX出力420のみを共有するものとして図示されているが、X、Y、およびZ出力420、415、および410のすべては、出力並べ替えキュー355を使用して共有され得る。

【0069】

サフィックス制御回路390は、(1)アプリケーションにおけるプログラミングおよび制御(条件、分岐など)と、(2)ロスレスゼロ圧縮および圧縮解除とに利用される。一部のアルゴリズムでは、多数の0データ値が生成される。これらは、ニューラルネットワークにおけるReLU演算によるものか、スパース行列演算によるものである可能性がある。数値に0を乗算したり、累積値に0を加算したりする操作は、本質的に意味がない。同様に、計算コア200の処理要素間でゼロ値を送ることは、帯域幅および電力を浪費する。オプションとして、代表的な実施形態(計算コア200および/またはR A E回路300)は、サフィックスビットを全体的に利用する種々のフラグ(条件フラグなど)または他の条件を処理することに加えて、データ経路のゼロを除去することによってデータ転送を圧縮および圧縮解除する能力を有するサフィックス制御回路390を含んでもよい。サフィックス制御回路390については、以下により詳細に説明する。

【0070】

さらに、以下により詳細に説明するように、サフィックス制御回路390は、例えば、限定されるものではないが、出力マルチプレクサ110を介したデータの出力の一部としてゼロ圧縮(ゼロ圧縮回路800を使用)を含み、入力マルチプレクサ205を介したデータの入力の一部としてゼロ圧縮解除(ゼロ圧縮解除回路805を使用)を含むように、計算コア200内で分散的に実施することもできる。

【0071】

図6は、第3の相互接続ネットワーク295の専用接続360、445を備え、第1のR A E回路対400Aおよび第2のR A E回路対400Bとして示されるR A E回路「クアッド」450および2つのR A E回路300対400を形成する複数の例示的または代表的なR A E回路300を示す高レベルブロック図である(これらは共にR A E回路クアッド450を形成する)。隣接する4つのR A E回路300(2行2列)は、第3の相互接続ネットワーク295の専用配線360、445によって接続され、R A E回路クアッド450としてマージされて、倍精度の乗算-加算、乗算-加算および論理演算を実行することができる。マージはまた、リンクされた4つのR A E回路300を単精度単サイクル複素乗算またはFFTバタフライとして使用するための接続、指数分解のためのワイヤ、および隣接するR A E回路300の符号に基づく否定のためのワイヤを含む。R A E回路対400およびR A E回路クアッド450の構成は、第3の相互接続ネットワーク295の専用配線360、445間の選択によって、例えば、選択(「SEL」)信号381を有する1つまたは複数のマルチプレクサ383および/またはイネーブル(「EN」)信号389を有する図示のANDゲート385を使用することによって達成されるが、これらに限定されるものではない。

【0072】

代表的な実施形態では、一例として、R A E回路300は、3つのデータ経路入力と、3つのデータ経路出力と、R A E 300機能を設定するために使用される制御インターフェース250とを有する。また、R A E回路クアッド450を構成する4つのR A E回路300間には、第3の相互接続ネットワーク295の専用の補助データ経路および制御接続360、445がある。各データ入力および出力は、32ビットのデータワード、単一ビットの有効なデータ(A X I - 4 s t r e a m t v a l i d)、およびセットの最初または最後のサンプル(アキュムレータの初期化、セットの完全な合計または最大値など)をマークするために使用される単一ビットマーカーを含む。オプションとして、r e a d y信号は、入力インターフェース上でフロー制御として出力することができ、出力インターフェース上で動作を停止するための入力です。

【0073】

代表的な実施形態では、XおよびZ入力（365、375）はそれぞれ、バイパスすることができ、最大4つの32ビットの定数値を介して配列することができ、または入力データを再配列するようにプログラムすることができる一組の定数レジスタのために使用される、4ディープの並べ替えキューを有する。XおよびZ入力並べ替えキュー350は、逆の入力を選択することもできる選択多重を含む（XおよびZはそれぞれ、Zからの4つのレジスタとXからの4つのレジスタとの間で選択するか、バイパスする）。入力並べ替えキュー350は、代表的な実施形態において、最大4サンプルのIQインターリーブ／デインターリーブ、FFTおよび複素多重並べ替えを可能にする。代表的な実施形態では、オプションとして、Y入力370は、別の入力へのクロスコネクタを有していない。定数は、使用前に入力370を介してロードされる。Y入力バイパス選択は、条件付き乗算のX入力の符号によって制御でき、Y入力と定数値（または最大4つのシーケンス定数値）の間で選択できます。各RAE回路300の出力はまた、対になった2つのRAE回路300のいずれかのZ出力から4つの遅延タップのいずれかを選択して配列することができる4ディープの出力並べ替えキュー355を有する。この並べ替えキューは、以下により詳細に説明するように、各RAE回路300への入力におけるものと同様に実施される。

【0074】

RAE回路300は、例えば、限定されるものではないが、ALUおよび多くの追加機能として動作するようないくつかのモードで動作する。これらには、いくつかの浮動小数点および複数の整数演算モード、複数のロジカル操作モード（ブール論理とシフト／回転）、複数の条件付き操作、および形式変換が含まれる。各RAE回路クアッド450は、独立して使用することができるか、または専用リソースを使用して互いにリンクすることができる4つのALUを実質的に含む。単一ALUはパイプライン融合多重集積を有する。基本乗数は 27×27 である。部分積のいくつかは、ゲート入力および／または合計出力を、リダクションツリー内の2つの重みにルーティングして、乗算器305を、（1） 24×24 各入力の上位3ビットをゼロにすることで乗算し、（2） 24×24 の乗算器として設定された第2の乗算器と対にされた（加算された）場合に 32×32 の乗算器を完成させるためのL字型の部分積であり、2つの乗算器305の出力を合計して、2つの乗算器から構成される合成 32×32 乗算器を達成し、（3）SIMD乗算を行うために、対角上に2つの 16×16 の乗算器を形成する複数の部分のみが存在する「ブルーニングされた（pruned） 32×32 」として再構成する。

【0075】

単一のRAE回路300は、

1. IEEE単一または整数上で、（パイプライン化された）サイクル当たり最大 27×27 の乗算を実行可能であり、
2. 2つの並列IEEE半精度、BFloat16、または（パイプライン化された）サイクル当たりINT16整数の乗算を実行可能であり、
3. （パイプライン化された）サイクル当たり4つの並列IEEE1／4精度またはINT8を実行可能であり、
4. 単一のRAE回路300は、2つの並列IEEEハーフ、すなわちBFloat16またはINT16の和を（パイプライン化された）サイクル当たり乗算可能であり、
5. 4つの並列IEEEクォーターの和または（パイプライン化された）サイクル当たりINT8乗算を実行可能であり、
6. 1／4精度または（パイプライン化された）サイクル当たりINT8複素乗算－2つの複素入力を1つの複素出力として実行可能であり、
7. 上記の機能のいずれかを有する融合加算を実行可能であり、
8. 4サイクルで1つのIEEE倍を実行可能であり、
9. 上記のいずれかの累積を実行可能であり、
10. 任意のビット数だけ64、32、 2×16 または 4×8 ビットシフトを実行可能であり、
11. 任意のビット数だけ64、32、 2×16 または 4×8 ビット回転を実行可能であ

り、

12. 32ビットのビットごとのブール論理を実行可能であり、および、

13. ストリーム、2つの演算対象のソートにおいて最小または最大を比較することが可能である。

【0076】

隣接する二つのリンクされたRAE回路300は、さらに、

1. サイクル当たり一つのINT32乗算を実行可能であり、

2. 4サイクルシーケンスで1つの（アキュムレータを使用して432 × 32部分を加算する）INT64乗算を実行可能であり、

3. 2つのIEEEシングルの和またはサイクルあたり2つのINT24乗算を実行可能であり、

4. 4つの並列IEEE半精度、BFLOAT16またはINT16の和を（パイプライン化された）サイクル当たり乗算することが可能であり、

5. 8つの並列IEEE1/4精度の和または（パイプライン化された）サイクル当たりINT8乗算を実行可能であり、

6. 1つの半精度または（パイプライン化された）サイクル当たりINT16複素乗算—2つの複素入力、1つの複素出力、4つの乗算、および2つの加算を実行可能であり、

7. 上記の機能のいずれかを有する融合加算を実行可能であり、および、

8. 上記のいずれかの累積を実行可能である。

【0077】

RAE回路300は、内部的に全てのモードについて乗算器出力幅以上の精度で演算を行い、「融合乗算加算」または「融合乗算累積演算」機能を実現する。丸めは、アキュムレータ／加算器の最終段階で行われ、IEEE754（2008）の「最も近い偶数に丸める」丸めモードに従うものとする。RAE回路300は、浮動小数点例外を生成も処理もしない。比較論理を使用して、入力データ内の浮動小数点例外（InfinitesおよびNaN）を検出してフラグを立てることができ、フラグを使用して、これらの条件が重要である場合に、追加のRAE回路300を犠牲にして例外を処理することができる。RAE回路300には、例外を処理するための専用論理は存在しない。非正規性は0として扱われる。0指数が入力に存在する場合、仮数の値に関係なく、その入力は0として解釈される。

【0078】

最終的な加算、飽和および丸め回路340は、整数オーバーフローを検出し、オーバーフローがなかった場合の出力と同じ符号の最大値で出力を置き換える。オーバーフロー検出では、内部データ経路がエラーなしでオーバーフローに対応できる十分な幅である必要があり、出力段階でオーバーフローを検出して出力を飽和値に置き換える必要がある。

【0079】

サフィックス（および／または条件付きフラグ）出力405は、構成によって設定された内部ソースからのセレクタを備える。条件ソースは、例えば、整数オーバーフロー、指数オーバーフロー、指数アンダーフロー非ゼロ、比較ブロックフラグ、乗算器積符号、Zシフト出力符号、およびアキュムレータ符号を含むことができる。選択された条件フラグは、関連付けられた出力のパイプライン遅延に一致する適切な複数のパイプライン遅延レジスタを持つ。条件フラグ出力は、同じクアッド内の他のRAE、同じクアッド内のFCシーケンサ、および隣接するクアッド内のシーケンサに配線される。別のRAE回路300または別のシステム構成要素から来ることができるサフィックス（および／または条件付きフラグ）入力380を、データ入力に加えて設けることができる。サフィックス（および／または条件付きフラグ）入力380は、例えば、乗算器出力の否定またはゼロ化、Z入力シフト論理出力の否定またはゼロ化、カウンタリロード／リセットの比較、およびアキュムレータ初期化のいずれかを制御することができる。

【0080】

再び図6を参照すると、RAE回路300は、RAE回路対400にグループ化するこ

10

20

30

40

50

とができ、これはさらに、RAE回路クアッド450と呼ばれる対の対にグループ化される。一对のRAE回路300は、 32×32 乗算または積の単一サイクル和として使用するよう設計されている（半複素乗算器にも有用である）。RAE回路クアッド450は、倍精度浮動小数点をサポートするために、4つの 27×27 乗算器を 54×54 単一サイクル乗算器に結合するよう設計される。密結合は、4つのRAE回路300間で積のプリアキュムレータ和および指数情報を共有するための専用配線360、445を含む。それはまた、4つのRAE回路300の間で入力を共有し、CORDICおよびNewton-Raphsonのような条件付きアルゴリズムをサポートするための条件フラグを共有するための追加の専用入力配線（個別には図示せず）を含む。さらに、代表的な実施形態では、RAE回路対400の隣接するRAE回路300も入力並べ替えキュー350および出力並べ替えキュー355を共有することができ、選択可能な構成によって制御される入力および出力のそれぞれの共有、交換、および置換を可能にする。

【0081】

4. 乗算器305

乗算器305は、符号付き固定小数点および符号なし固定小数点のための1レーン、2レーン、および4レーンSIMD演算、ならびに二重、単一、半、および $1/4$ のIEEE精度の浮動小数点、ならびにBFLOATフォーマットをサポートするよう具体的に定式化される。これらのフォーマットは、様々なサイズの乗算器と、SIMDモードをサポートするために乗算器305を破壊する能力と、高次モードのためにそれを結合する能力とを必要とする。モードと関連する乗数サイズを表1にまとめる。さらに、乗算器305は、RAE回路クアッド450内の乗算器の積の和を乗算器ツリー内で実行することを可能にし、同じRAE回路クアッド450の乗算器305を結合して、最も一般的に使用される単精度ユースケースのためにALUサイズを最小化するために、使用頻度の低い倍精度およびINT32乗算器を生成することを可能にする。

【0082】

【表1】

表1: 乗算器モード

モード	要求される乗算器	モード	実施	クアッド当たりのオペレーション
IEEE double	53×53	Sign-mag	54×54 4つのRAEs	クロックあたり1
IEEE single	24×24	Sign-mag	27×27	クロックあたり4
IEEE 半精度	2つの 11×11	Sign-mag	2つの 16×16	クロックあたり8
IEEE $1/4$ 精度	4つの 4×4	Sign-mag	4つの 8×8	クロックあたり16
BFLOAT16	2つの 8×8	Sign-mag	2つの 16×16	クロックあたり8
INT64	64×64	Signed/unsigned	32×32 およびシフト付きアキュムレータを使用した4サイクル	4クロックあたり2
INT32	32×32	Signed/unsigned	32×32 2つのRAEs	クロックあたり2
INT24	24×24	Signed/unsigned	24×24 1つのRAE	クロックあたり4
INT16	16×16	Signed/unsigned	2つの 18×18	クロックあたり8
INT8	8×8	Signed/unsigned	4つの 9×9	クロックあたり16

【0083】

乗算器305の複数のデータ入力は、入力並べ替えキュー350を介してRAE回路300のXおよびY入力365、370から得られる。入力並べ替えキュー350は、定数、最大4つの定数のシーケンス、または最大4つのサンプルを乗算器ブロックのX入力に再配列する入力データをアサートする機能を有する論理を含む。Y入力符号ビット（ビット31）は、X入力を制御して、Y符号の値に基づいてX入力を定数に選択的に置き換えることができる。XおよびY入力は、モードに応じてさまざまな形式で解釈される32ビ

ット信号である。浮動小数点形式の場合、符号ビットは常に各SIMDレーンの左端のビットで、指数は次の最上位ビットになる。隠しビット (hidden bit) は指数の最下位ビットと同じビットであり、非正規性は0として解釈されるため、浮動小数点形式の乗算器アレイ入力では常に「1」に強制される。残りの指数ビットと符号はマスクされ、各浮動小数点モードの乗算器アレイ入力で「0」に強制される。符号なし整数 (uINT) モードでは、INT24ビットモードを除き、すべてのビットが乗算器アレイに渡される。INT24ビットモードでは、入力の最上位バイトがマスクされ、乗算器アレイの最上位3ビットを「0」に強制する。

【0084】

符号付き整数は符号ビットをマスクし、これらのビットを乗算器アレイに強制的に「0」にし、入力符号ビット値を乗算器出力論理に渡して符号補正を適用する。形式を表2にまとめる。各入力からの27を超えるビットは、 27×27 乗算器アレイに接続されない。double、INT32、INT64モードは、複数の乗数を使用する特殊なケースである。このような場合、入力は関連する複数のRAEに伝播される必要がある。これらの特殊な場合の符号付き整数モードでは、適切な結果を得るために、最上位の入力セグメントは符号付きとして扱われ、その他のセグメントは符号なしとして扱われる。64ビット符号付き整数の場合、乗算は4つの32ビット乗算のシーケンスである。この場合の入力の符号は、各入力の最も重要な半分だけが符号付けられるようにシーケンス処理する必要がある。両方の入力の指数ビットと符号ビットも指数論理に接続され、指数論理は否定を制御する。

【0085】

10

20

30

40

50

【表 2】

表2: モード別乗算器の入力フォーマット

モード	符号ビット	指数	hidden	仮数	ノート
IEEE double ¹	H 31 L n/a	H: 30:20 L: n/a	H: 20 L: n/a	H=19:0, L=31:27 L=26:0	1
IEEE single	31	30:23	23	22:0	
IEEE 半精度	31,15	30:26, 14:10	26, 10	25:16, 9:0	
IEEE 1/4精度	31,23,15,7	30:27, 22:19, 14:11, 6:3	27, 19, 10, 3	26:24, 18:16, 10:8, 2:0	
BFLOAT16	31,15	30:23, 14:7	23, 7	22:16, 6:0	
uINT64 ²	n/a	n/a	n/a		2
uINT32 ³	n/a	n/a	n/a	31:0	3
uINT24	n/a	n/a	n/a	23:0	
uINT16	n/a	n/a	n/a	31:16, 15:0	
uINT8	n/a	n/a	n/a	31:24, 23:16 15:8, 7:0	
INT64	*	n/a	n/a	*	2
INT32	31	n/a	n/a	30:0	3
INT24	23	n/a	n/a	22:0	
INT16	31,15	n/a	n/a	30:16, 14:0	
INT8	31,23,15,7	n/a	n/a	30:24, 22:16, 14:8, 6:0	

(注) 1. ダブル(Double)は、クアッドの4つの乗算器をすべて使用する。
各64ビット入力の上半分は、クアッド内のRAEの2x2アレイの1行または1列に適用される。
内部の専用クワッド配線は、必要なビットを行／列内の他のRAEに分配する。
仮数の27ビットの上半分と下半分の各組み合わせは、複数の乗算器の内の1つで乗算される。
したがって仮数の27ビットの上半分と下半分は分配されるべきである。
2. Int64は、1対のint32乗算器と1つのアキュムレータを使用して、4クロック当たり4つの
32x32部分積を順次実行する。符号付きINT64の場合、論理は、最上位半分の入力が符号付きで、
最下位半分が符号なしになるように、入力の符号コントロールを順序付けする必要がある。
3. INT32は、同じRAE対で2つの乗算を使用する。上半分は特殊モードを使用
両方の32ビット入力を対の両方のRAEに分配する必要がある。

【0086】

各乗算器（XおよびY）入力には、入力を符号付きまたは符号なしとして個別に指定する符号付き制御がある。浮動小数点モードの場合、符号付き入力を符号なしとして指定するには、符号付き入力を「0」にする必要がある。この入力は、符号付きINT64モードをサポートするように順序付けできる必要がある。符号付き制御は、すべてのSIMDレーンに同じように適用される。乗算器はまた、アサートされたときに乗算器出力を否定する否定入力を有する。この制御は、シーケンス処理が可能である必要がある。

【0087】

コンフィギュレーション入力は、（適切な桁上げブロックビットを設定し、符号補正と否定の加算のフォーマットを設定する）乗算器SIMDモードを設定し、入力マスクを選択します。構成には、上記の符号付きまたは符号なし、否定、絶対値、および0の設定も含めることができる（これらは構成の別の場所で設定できる）。

【0088】

乗算器の積は、桁上げ保存形式で64ビットベクトルの対として表される64ビット2の補数積である。これら2つのベクトルの合計は、積の2の補数値である。この積は、モードに応じて、4つの16ビットレーン、2つの32ビットレーン、または単一の64ビットレーンに分けられる。符号付きSIMDモードは、各レーンに個別に符号補正を適用し、隣接するレーンへのオーバーフローを防ぐために桁上げブロッカーを備えている。2つの乗算器入力のそれぞれには、data_valid_in_bitが付随する。これらのビットはANDで結合され、乗算器のパイプライン遅延に合わせて遅延され、乗算器

から `data_valid_out` になる。

【0089】

乗算器の設計は、 27×27 の乗算を実現するために積項を再配置するように修正された 27×27 の符号なし乗算器、SIMD 使用のために対角線上に2つの 16×16 の部分積のみを有するブルーニングされた 32×32 の乗算器、および2つの乗算器を合計して 32×32 の乗算を行うことを可能にする「L」字形の拡張である。さらに、符号なし乗算器は、符号付き2の補数入力での使用をサポートするために、出力に補正を加える。また、出力補正論理は、符号絶対値 (`sign-magnitude`) の積を2の補数に変換し、乗数の出力を否定する制御を提供する。

【0090】

A. ネイティブ 27×27 モードの RAE 乗算器 305

図7は、ネイティブ 27×27 モードにおける乗算器305を示す。ブロック402は、仮想 8×8 個の部分積を表し、ブロック402内の数字404は、部分積を合計するときにその部分積に適用される8ビット左シフトの数を表す。図7の上側および右側の網掛けブロックは、24ビットモードでは強制的にゼロになる。下部に沿ったX入力 は列内のすべてのセルにブロードキャストされ、左側に沿ったY入力 は行内のすべてのセルにブロードキャストされる。この表現は説明のためだけのものであることに注意すべきである。24ビットモードは、単精度浮動小数点および24ビット整数モードに使用されます。24ビットモードでは、27ビットの乗算器が使用され、各入力の上位3ビットが強制的に0に設定される。あるいは、ゼロ化された入力ビットに対応する個々のビット積を強制的に0にしてもよい。浮動小数点形式には、ビット31の符号ビットと、ビット22:0の仮数ビットが含まれる。指数部がすべてゼロでない限り「1」である隠しビットもあり、その場合は「0」である。乗算器への入力ゲーティングは、XおよびY入力上の入力ビット42:0を乗算器アレイのビット32:0に割り当て、浮動小数点が入力のビット23を「1」に強制し、0が 27×27 乗算器の各入力上の上位3ビットをパディングする。

【0091】

B. SIMDモードにおけるRAE乗算器305

図8は、2つの 16×16 SIMD乗算器として 32×32 乗算器を使用することを示す。部分積配列内の対角線上にない部分積はゼロ化されるため、残りのゼロ化されていない部分積はすべて入力の独立したサブセットを使用し、ネイティブ配列内のセルの重み付けによって、配列の出力で積が重複しないようにする。したがって、図8に示すように、対角線から部分積403をゼロにすることによって、他の点では修正されていない符号なし乗算器を、任意のサイズのより小さい独立乗算器のグループとして使用することができる。図示のケースでは、 32×32 符号なし乗算器を使用して、2つの 16×16 乗算を実行する。乗算が符号なしである限り、16ビット入力と32ビット積は重複しない。同様の方法を使用して、 32×32 符号なし乗算器で4つの 8×8 符号なし乗算を実行する。ゼロ化された部分積のためのハードウェアは、SIMD演算のために存在する必要がないことに留意されたい。これらが存在する場合は、結果として得られる部分積が目的の部分積と結合するのを防ぐために、それらは無効にされる。

【0092】

図9は、2つの 16×16 SIMD乗算器に対する 27×27 乗算器の修正を示す。乗算器305は 27×27 ビットのみであるので、図8に示される右上の部分積は、修正されていない 27×27 乗算器を使用して、 16×16 (点線は 27×27 乗算器サイズを示す) の代わりに 11×11 に切り捨てられる。上位SIMD積を増加させるために乗算器に拡張を加える代わりに、SIMDモードに適したブルーニングされた 32×32 乗算器を達成するために既存の無効部分積を再割当できる。部分積は、慎重に選択した場合、元の乗算器と同じように、拡張内で互いに同じ重みを有する。これにより、移植された部分積の下移動部分積および桁上げ保存ツリーを、出力の1レベル上の桁上げ保存加算器のメイン乗算器ツリーに追加される完全なままの桁上げ保存ツリーとして移動させることができる。可動部分積の選択は、図9に、網掛けブロック (27×27 の位置のブロック

10

20

30

40

50

406、SIMD位置のブロック408)によって示される。点線の内側にある 27×27 乗算器の網掛けされていない部分は、SIMDモードでは無効になる。図示されているように、乗算器の 8×8 セグメントへの分割は説明のためのものであり、実際の乗算器は、その元の位置に対して24ビットの左シフトで再割り当てされる部分積の 5×16 ビットおよび 11×5 ビットの合計を有する。これらの部分積への入力は、入力ビットを再割り当てするセレクタを必要とすることもある。

【0093】

あるいは、 27×27 の乗算器は、SIMD拡張子を常に使用し、 27×27 の使用では無効にすることで構築できる。これは、桁上げ保存ツリーにおけるスイッチングを排除し、ビット積入力における選択を減少させて、伝搬遅延をわずかに減少させるが、追加されたビット部分およびその下の桁上げ保存ツリーを考慮してゲート数を増加させる。乗算器305の設計のための別のオプションは、ベースとして 24×24 乗算器を使用して、 8×8 ブロックを切り替えることである。これにより、より多くの1ビット積が移動するが、両方の入力でビット27:24がゼロでないため、切り替えの複雑さが増すことによって、桁上げ保存ツリーの追加のコストが相殺される場合がある。

【0094】

図10は、可動部分積を有する乗算器305の構成を示す。図10に示すように、ブロック402内の数字は、各入力からの1ビットの論理積によって生成される1ビット部分積の配列を表す。これらの1ビット積は、入力のビット位置の和に対応する重みの2乗で加算ツリー306に加算される。図の左側412に示すように、 27×27 乗算器305は、可動部分に対応する1ビット部分積を除去するようにブルーニングされる。除去された部分積は、右側414で別々に合計され、桁上げ保存ツリーの最後のレイヤで、削減された 27×27 乗算器桁上げ保存出力に追加される。 $27 / 27$ の使用のために、追加されたツリー306は元の重みを保持する。

【0095】

SIMDブルーニングされた 32×32 を使用する場合、パーティション化されたツリー出力には、元の位置の左側に24ビットの有線シフトが追加される。分割ツリーへの入力は、 27×27 モードか 32×32 モードかによって、2つの可能なビットの間で選択される。図10に示す加算器の出力は、その合計が積(桁上げ保存形式)となる一対のベクトルである。桁上げ保存形式を使用すると、ALUからの出力まで、コストのかかる桁上げの伝播が延期される。図10は、部分積を移動させる概念を示す。わかりやすくするために、図10は、1ビット部分積の 8×8 ブロックを示す。実際の設計では、部分的な製品の 27×27 アレイがあり、そのうち 5×16 と 11×5 の領域が削除され、別々に合計される。SIMDモードの乗算器は符号なしのみである。以下に説明するように、符号付き動作が選択されると、各SIMDレーンの乗算器305出力に補正が加えられる。

【0096】

図11は、4つの 8×8 SIMD乗算器として、ブルーニングされた 32×32 乗算器を使用することを示す。 8×8 SIMDモードの4つのレーンは、2つのレーン 16×16 ブルーニング 32×32 乗算器305のサブセットである。上述の乗算器モードは、図11に示すように、ブルーニングされた 32×32 の対角線上に配置された4つの 8×8 乗算器307に到達するために、 8×8 ビット部分積308の4つのブロックを無効にする追加の制御を有する。 16 ビットSIMDと同様に、1ビット部分積は、それらが積に追加されないように、それらを無効にするようにゲートされる必要がある。SIMDモードの乗算器305は符号なしのみである。符号付き動作が選択されると、各SIMDレーンの乗算器305出力に符号補正が加えられる。

【0097】

C. RAE乗算器305 (32×32 モード)

図12は、2つのRAE乗算器305のシフトされた和によって形成される 32×32 乗算を示し、そのうちの1つは修正される。2つの隣接する乗算器305は、INT32およびINT64モードで使用するための単一サイクルパイプライン化された 32×32

10

20

30

40

50

乗算器を構成するために結合される。結果として得られる乗算器 305 は符号付きと符号なしの両方のモードで動作し、整数演算に使用される。乗算器 305 は、 24×24 （強制的に 0 にされたまたは無効にされた部分積両方の上位 3 ビット）用に構成された RAE 回路対 400 の 2 つの乗算器 305 を使用して構成される。下位乗算器は、ネイティブの 24×24 であり、X 入力 365 および Y 入力 370 のそれぞれの下位 24 ビットを受け入れる。第 2 の乗算器 305 は、図 12 に示すように、上側および右側 416 で乗算器アレイを 8 ビット拡張するために「L」形状の部分積を形成するように修正される。

【0098】

L 脚の両端は、中央の行の最初の左の列と下の行の中央の 8×8 ブロックから取られる。これらは両方とも、それらが再配置されたときの重みとして、上側の残りの部分に対して 28 の相対的な重みを有するので、部分積のこれら 2 つの 8×8 ブロックの相対的な重みを変更するために、加算器ツリーに修正を加える必要はない。 8×8 部分積乗数のこれら 2 つのブロックへの入力、物理的な列と行への入力と一致しなくなったため、これらの入力を切り替えて、仮想形状を実現するために正しい入力を接続する必要がある。残りの 2 つの部分積は未使用であり、したがって、各ビット部分積への入力の少なくとも 1 つをゼロにすることによって、または、積和への不要な加算を防止するためにこれらの部分積を無効にすることによって無効にされる。上側の「L」字形部分積は、下側の乗算器の積に対して 16 ビットだけ左にシフトされる。このシフトは、2 つの RAE からの積が合計されるポスト乗算シフト／コンバイナ論理における下側積の右シフトによって達成される。下側製品の Z 入力 375 は、Z 入力 375 を 32×32 積と合計するように適切に位置合わせされる。両方の 32 ビット入力は、対の両方の RAE 乗算器 305 に分配されることに留意されたい。RAE 回路クアッド 450 内の専用配線は、2 つの RAE 入力を相互接続し、出力を合計するための配線を提供することに注意を払う。 32×32 は、最初の 27×27 を 24×24 として使用し（各入力の上位 3 ビットを 0 にする）、2 番目の 24×24 の 8×8 部分をシフトして 32×32 ビット L を形成することによって、より効率的にレンダリングされる。これは、レーン部分間のゼロなしと同じ部分サブセットを使用し、拡張機能で 16 ビットシフトを使用して下位の 24×24 に追加する。16 ビットシフトは、16 対シフト入力によるシフトを排除して、既存のモードシフトで達成することができる。

【0099】

D. RAE 乗算器 305 (54×54 モード)

図 13 は、4 つの修正されていない RAE 乗算器 305 のシフトされた和によって形成される 54×54 の乗算を示す。IEEE 倍精度モードには、 53×53 ビットの符号なし乗算器 305 が必要である。この場合、RAE 回路クアッド 450 内の 4 つの 27×27 乗算器 305 は、単に、合計の適切なシフトおよび入力の再割り当てとともに合計される。加算は、 $3 \times 9 = 27$ ビット左シフトの右乗算器を左乗算器に加算し、次に、図 13 に示すように、その上の和を下との和にさらに 27 ビットシフトして加算し、 54×54 積に到達する。なお、左側の乗算器への同じ Y 入力ビットは、右側の乗算器にも分配されるべきであり、乗算器の下の方に適用される同じ X 入力は、上側の行にも適用されるべきである。RAE 回路クアッド 450 内の配線は、これを可能にするための専用ルーティングを含む。

【0100】

E. 64 ビット逐次乗算および符号補正

128 ビットアキュムレータと結合されたシフトコンバイナ設計は、32 ビット乗算累算として 2 つの RAE 回路 300 を使用して 64 ビット乗算の 4 サイクル逐次計算を可能にする。入力とシフトコンバイナの第 2 のレイヤは、4 つの 32×32 の部分積を計算し、128 ビットの結果に対してそれらを合計するために適切な量だけそれらをシフトするようにシーケンスされる。下*下部分積はシフトされず、上*下および下*上部分積は 32 ビットだけ左シフトされ、上*上部分積は 64 ビットだけ左シフトされてから、累積合計に加算される。シフトは、後述する乗算後乗算器シフトコンバイナネットワーク 310

で行われる。出力には、128ビット製品の32ビットまたは64ビットセグメントを取り込み、選択し、出力するためのセクタおよび／またはシフトレジスタが必要である。符号付き演算では、各64ビット入力の上半分のみが符号付きであるため、乗数への各入力、上半分は符号付き、下半分は符号なしとして順序付けされる必要がある。部分積シーケンスは、各入力の下半分の積で始まり、各入力の上半分の積で終わる。符号付きコントロールは、各入力の上半分のみを符号付きとして処理するシーケンスでシーケンス処理される。

【0101】

追加の考慮事項は、すべての整数モードで符号付き乗算と符号なし乗算の両方を処理する必要がある。これは、符号の大きさの整数表現もサポートする制御論理への単純な拡張である。データは符号絶対値形式で表されるため、浮動小数点モードではすべて符号なし乗算器が使用される。浮動小数点のユースケースでは、符号は排他的にORされ、乗数の周りで渡され、乗数自体は符号なしである。整数形式は、整数モードごとに符号付きと符号なしの両方の整数入力をサポートする。

10

【0102】

符号なし乗算器は、符号付き演算の $-2n-1$ に対して符号なしデータの $-2n-1$ である各被乗数の最上位ビットの重みを考慮しないため、負の2の補数被乗数を正しく処理しない。部分積を合計する場合、負の部分積は、正しい答えを得るために、製品の幅に符号拡張され、負として扱われる必要がある。符号付き乗算器を実現するために拡張符号に対する部分積項の生成を避けるための符号化トリックを含むBoothやBah-Woolleyのような乗算器最適化がある。

20

【0103】

代表的な実施形態では、SIMD演算をサポートする必要性と、全てのモードをサポートするためにハードウェアを削減する目的で乗算器を部分積に分離する設計選択は、これらの符号付き乗算器方式を著しく複雑にするか、または完全に破壊する。1つの方法は、符号を保持する各入力の2の補数の絶対値を実行して、2の補数の符号付き入力を符号の大きさの形式に変換し、それをさらなる操作なしに乗算器305に供給することであり、これにより、符号なし乗算を使用することが可能になる。2の補数変換は、入力のすべてのビット（1の補数）を反転し、それに1を加えて2の補数を生成することによって行われる。1の付加による桁上げ伝搬は、高速桁上げ方式に対してさえ顕著なゲート遅延を付加し、高速方式に対しても大きな面積ペナルティを生じる。

30

【0104】

代表的な実施形態は、符号なし乗算器305が符号付き値を乗算するために使用される場合に生じるエラーを理解し、次いで、そのエラーに対して補正を適用することを含む別のアプローチを使用する。発生したエラーは次のように分析される。被乗数Aが負で、Bが負でない乗算を考えてみる。

【0105】

【数1】

$$A \times_s B = -((-A) \times_u B)$$

$$= 2^{2n} - ((-A) \times_u B)$$

$$= 2^{2n} - ((2^n - A) \times_u B)$$

$$= 2^{2n} - 2^n \times B + A \times_u B$$

40

$2n$ 項および $2 \times 2n$ 項は、それぞれ n ビットおよび $1n$ ビットワードに対する2's complement補数表記への変換であり、これらはそのビット数に対するモジュロ範囲外であるが、2's complementの正しいビット表現を得るために必要である。解析の最終行の $2 \times 2n$ 項は、mod $2 \times 2n$ の結果の範囲外であるため無視される。積 $2n \times B$ は、正しい符号付き積を得るために

50

減算する必要がある符号なし積の誤差項で、単純にBを積の上半分にシフトする。同様に、Bが負の場合、 $-2n \cdot A$ 項が積から欠落している。同様の分析では、両方の負の入力が、これらの同じA補正とB補正の合計である補正で終わることが示される（これは、乗算と分布特性の直線性からも導かれる）。

【0106】

減算は、補正の2の補数を加算することによって実行される。補正は加算であり、乗算器の出力は加算器のツリーであるため、出力ではなく乗算器の加算器ツリーの任意のポイントで補正を適用できる。補正論理は、乗算器モードおよび乗算器への入力にのみ依存するので、補正值は、符号なし乗算器部分積に平行な経路で計算することができ、桁上げ保存形式での最終的な補正は、加算器ツリーの任意の都合のよい点で加えることができる。

10

【0107】

図14は、符号なし乗算器305を用いた符号付き乗算のための符号付き補正回路428を示し、上述のように符号補正を実行するための論理を示す。2つの補数は、すべての入力ビットを反転して1を加算することによって、各入力に対して計算される(418)。2つの補数は、それぞれ、反対側の入力の符号ビットによってゲートされ(422)、ゲートされた補正が合計され(424)、そして、有線シフトを介して上側半積に加算される(426)。図の2つの補数値のゲートには、符号付きと符号なしの間でモードを切り替えることができるように、高いアサート時に補正(強制0)の生成を無効にする符号付き `int` コントロールが含まれる。実際には、補正計算加算器は、最適化されたツリーとして一緒に移動され、ゲートは、補正論理への入力で反転と共に適用される。

20

【0108】

SIMD演算では、各レーンには、そのレーンの入力フィールドと符号のみを使用する独自の符号補正回路があり、補正が次のレーンに桁上げを伝播することは許可されない。これは、各SIMDレーン間に少なくとも1つのバッファビットまたは桁上げブロッキング論理が必要であることを意味している。この設計では、レーンを8ビットの倍数で維持することを選択する。そのため、各レーンのガードビットに必要な入力の余分な切り替えを回避するために、レーン境界で桁上げブロッキングゲート(`carry-blocking gating`)が使用される。ガードビットがない場合、桁上げブロッキングは、補正が適用される段階および後続のすべての段階で適用される。このため、この設計では、乗算器の桁上げツリーの出力まで補正を延期する。

30

【0109】

図15は、各符号付き整数モードに対する入力、符号補正、および出力のアライメント、すなわち、各符号付き整数モードに対する出力に対する入力および符号補正のアライメントを示す。各レーンの補正は、他のレーンから独立して計算される。32ビット符号付き乗算器の場合、補正は乗算器の一方にのみ適用される(示されているアラインメントは下位乗算器に適用される補正用である)。それ以外の場合、補正は2つの異なるアラインメントで2回適用される。

【0110】

図16は、選択的否定も行うことができる符号付き補正回路432を示す。浮動小数点の乗算と加算を使用する場合、加算を実行するには、通常、数値形式を2つの補数形式に変換する必要がある。乗算器と加算器の間に別個の2つの補数段階を追加するのではなく、乗算器の出力の前に積に対して2つの補数を実行することによって、浮動小数点符号と否定制御の関数として積を選択的に否定することができる。2つの補数は、積のすべてのビットを反転してから1を加算する(418)か、積から1を減算してからすべてのビットを反転することによって行われる。後者は、乗算器ツリー内の積に負の1を選択的に加算し、次いで、ツリー内の伝搬またはゲートカウントに影響を与えずに否定を達成するために、完成した積を反転するために、上述の符号補正論理を増強することができることを示唆する。ただし、この場合、乗数は桁上げ保存出力を持つため、2つの出力ベクトルの合計は積になる。

40

【0111】

50

積を否定するために、 $-(C+S) = -C-S = \sim(C-1) + \sim(S-1)$ の合計出力と桁上げ出力の両方が否定される。したがって、出力を反転する前に、修正された積は $C+S-2$ であるため、出力で2つのベクトルを反転する場合は、ツリーに -2 を追加する必要がある。図16は、積の選択的否定を含む符号補正論理の修正を示す。浮動小数点または符号の大きさの場合は、符号ビットを否定制御と共に評価して、乗数の出力が否定されているかどうかを判断する必要がある。この場合、乗数の出力は常に2つの補数の桁上げ保存ベクトルである。積の上半分に符号補正が追加され、下半分の補正は0になる。否定の場合、下半分は -2 であり、上半分は符号拡張+符号補正(2の補数符号付き乗算器の場合)であるため、乗算器符号補正にすべて「1」ビットが追加される。否定コントロールは、積の絶対値制御を含むように拡張される。これは、絶対値が選択されている場合に、負の制御を入力した符号の排他的ORにするだけである。

10

【0112】

乗算器305が 36×36 または 54×54 モードのために結合される場合、2つの補数は各RAE回路300の部分積に対して実行されるので、負の制御が関係するすべての乗算器について同じであることを確認する以外に、RAE積を合計するための調整は必要ない。

【0113】

浮動小数点演算が選択されている場合、指数は32ビット入力の一部である。浮動小数点モードの場合、指数部に対応する乗算器305への入力は、指数部の最下位ビットを除いて「0」にマスクされ、指数部の最下位ビットは隠しビットとして「1」に強制される。非正規性は0と解釈され、検出されると、乗数の後にデータ経路が強制的に0になる。乗算器指数処理(ゼロ検出、加算、アラインメントシフト)は、指数論理335において行われる。シフトコンバイナ指数論理335の詳細は、以下で説明する。指数ビットのマスクされていない値は、変更されることなく、乗算器シフトコンバイナの指数論理335に渡される。

20

【0114】

乗算器入力は、いくつかの部分積を無効にし、32ビット拡張およびSIMD拡張モードのための入力を再割り当てするために、いくつかのスイッチングを必要とし、入力並べ替えキュー350または他のスイッチング回路において実行することができる。入力論理は、浮動小数点モードの隠しビットもアサートする。

30

【0115】

F. 乗算器305モード

基本的な乗算器の設定には 27×27 乗算、SIMD用の 32×32 の削減、32ビット拡張の3つがある。また、 24×24 は 27×27 のサブセットであり、各入力の3msbsが強制的に0になり、4レーンSIMDは2レーンSIMDのサブセットであり、 8×8 ビット部分積の4ブロックが無効になり、0に強制される。浮動小数点モードでは、入力にマスクが適用され、符号と指数に関連付けられたビットが0になる。SIMD構成は、上述したように加算器ツリーの一部を移動することによって達成される、いくつかの部分積の重み変更を必要とする。他のすべての構成は、入力を部分的なプロダクトのグループに切り替えることによって厳密に処理される。

40

【0116】

乗算器305アレイのネイティブモードは、乗算器305である。このモードでは、乗算器への各27ビット入力は、XおよびY入力の最下位27ビットから取得されるか、または倍精度乗算の場合、入力は、下位積については最下位27ビットから、上位積についてはビット53:27から取得される。隠しビットの場合、ビット53は強制的に0になり、ビット52は強制的に「1」になる。ダブルスの切り替えは、乗算器305の前に行われる。

【0117】

24×24 モードの入力は 27×27 と同じであるが、XとYの入力の最上位3ビットが強制的に0になり、有効な乗算器が符号なし 24×24 の乗算器になる点が異なる。表

50

3に、乗算器の各 8×8 サブセットへの入力割り当てを示す。各ブロックは、下位24ビットの 8×8 部分積への入力を表す。このモードでは、各27ビットの乗算器入力に対する上位3ビットの入力が強制的に0になります。

【0118】

表3：24×24乗算器入力

【表3】

表3: 24x24乗算器入力

X(7:0) 000	X(15:8) 000	X(23:16) 000	000 000
X(7:0) Y(23:16)	X(15:8) Y(23:16)	X(23:16) Y(23:16)	000 Y(23:16)
X(7:0) Y(15:8)	X(15:8) Y(15:8)	X(23:16) Y(15:8)	000 Y(15:8)
X(7:0) Y(7:0)	X(15:8) Y(7:0)	X(23:16) Y(7:0)	000 Y(7:0)

10

【0119】

32ビット拡張は、二つの乗算器を組み合わせて 32×32 の乗算を実現する場合の上位乗算器である。下側の乗数は、前述の 24×24 モードに設定される。上の乗算器は、2つの仮想 8×8 部分積への入力を切り替え、さらに2つをゼロにして、乗算器を、 32×32 乗算器の8つの最上位行および列に対応するL字型部分積にする。この拡張では、上記で説明した 24×24 配列への入力としてXおよびY入力の最上位24ビットを使用するが、両方の入力の下位16ビットの部分入力を置き換えて、下側の対角線を0にし、他の2つの製品を最上位の行にするようにする。また、最下位の8ビットの場合も同様である。表4は、 24×24 アレイを構成する各仮想 8×8 部分積の入力を示している。27×27乗算器への各入力の最上位3ビットは強制的に0になる。表4の太字の凡例は、このモードの通常の 24×24 とは異なる入力を示す。

20

【0120】

表4：32×32乗算器の拡張

【表4】

表4: 32x32乗算器の拡張

X(7:0) 000	X(15:8) 000	X(23:16) 000	000 000
X(15:8) Y(31:24)	X(23:16) Y(31:24)	X(31:24) Y(31:24)	000 Y(23:16)
X(7:0) Y(31:24)	0	X(31:24) Y(23:16)	000 Y(15:8)
0	X(31:24) Y(7:0)	X(31:24) Y(15:8)	000 Y(7:0)

40

【0121】

SIMDモードは、24ビットの共通オフセットによって、 8×8 部分積ブロックのうちの3つの重み付けを変更する。両方の入力の16個の最下位ビットに対応する仮想部分積に対する入力は、両方の入力の8個の最上位ビットに対応する8個の 8×8 部分積と同様に、 24×24 乗算器305に対する入力と同じである。3つの可動部分積は入力の再割り当てを必要とし、赤色の凡例は、 32×32 拡張モードで使用される入力と同じである。太字の凡例は、SIMDモード固有の入力アサインを示している。 $4 \times 8 \times 8$ SIMDモードでは、グレーの網掛けされたセルを強制的に0にする必要がある。8ビットSIM

50

Dのためにこれらの仮想部分積をゼロにするために、これらの部分積だけについて、XのY入力のいずれかを強制的にゼロにすることができる。

【0122】

表5：乗算器SIMDモード入力

【表5】

表5：乗算器SIMDモード入力

X(7:0) 000	X(15:8) 000	X(23:16) Y(26:24)	X(26:24) Y(26:24)
X(23:16) Y(31:27)&000 / 0	X(31:24) Y(31:27)&000	X(23:16) Y(23:16)	X(26:24) Y(23:16)
X(7:0) / 0 Y(15:8)	X(15:8) Y(15:8)	X(31:27)&000 0 & Y(26:24)	000 Y(15:8)
X(7:0) Y(7:0)	X(15:8) Y(7:0) / 0	X(31:27)&000 / 0 Y(23:16)	000 Y(7:0)

10

【0123】

上記のモードは、各乗算器入力ビットに対して最大3入力マルチプレクサを必要とする。各モードのチャートは、表6および7の要約チャートにまとめられている。表6の各ブロックは、 24×24 の乗算器305の仮想 8×8 の部分積を表す。乗算器入力の各々への最上位3ビットは、27ビット整数及び倍精度浮動小数点モードを除いて、部分積の最上位の行及び列を効果的に遮断するために「0」に強制される。各部分積セルの一番上の行は、 24×24 モードのXおよびY入力に対応し、中央の行は32ビット拡張モードに対応し、各セルの一番下の行はSIMDモードの入力に対応する。8ビットSIMDの場合、太字のセルは強制的に0になる。／0は、SIMD8モードで強制的に0になる入力を示す。「&0」は右側のパディングが0であることを示し、「0&」は左側のパディングが0であることを示す。

20

【0124】

表6：乗数入力の要約

【表6】

表6：乗数入力の要約

X(7:0) X(7:0) X(7:0)	Y(26:24)/0 000 000	X(15:8) X(15:8) X(31:24)	Y(26:24)/0 000 000	X(23:16) X(23:16) X(23:16)	Y(26:24)/0 000 Y(26:24)	X(26:24)/ 0 000 X(26:24)	Y(26:24)/ 0 000 Y(26:24)
X(7:0) X(15:8) X(23:16)	Y(23:16) Y(31:24) Y(31:27)&0/ 0	X(15:8) X(23:16) X(31:24)	Y(23:16) Y(31:24) Y(31:27)& 0	X(23:16) X(31:24) X(23:16)	Y(23:16) Y(31:24) Y(23:16)	X(26:24)/ 0 000 X(26:24)	Y(23:16) Y(23:16) Y(23:16)
X(7:0) X(7:0) X(7:0)	Y(15:8) Y(31:24) Y(15:8) / 0	X(15:8) 0 X(15:8)	Y(15:8) Y(15:8)	X(23:16) X(31:24) X(31:27)&0/ 0	Y(15:8) Y(23:16) 0&Y(26:24))	X(26:24)/ 0 000 000	Y(15:8) Y(15:8) Y(15:8)
X(7:0) 0 X(7:0)	Y(7:0) Y(7:0)	X(15:8) X(31:24) X(15:8)	Y(7:0) Y(7:0) Y(7:0)/0	X(23:16) X(31:24) X(31:27)&0/ 0	Y(7:0) Y(15:8) Y(23:16)	X(26:24)/ 0 000 000	Y(7:0) Y(7:0) Y(7:0)

30

40

【0125】

50

【表 7】

表7: 8x8セルのみの乗算器入力の変約

X(7:0)	Y(26:24)/0	X(15:8)	Y(26:24)/0	X(23:16)	Y(26:24)/0	X(26:24)/0	Y(26:24)/0
X(7:0)	000	X(15:8)	000	X(23:16)	000	000	000
X(7:0)	000	X(31:24)	000	X(23:16)	000	000	000
X(7:0)	Y(23:16)	X(15:8)	Y(23:16)	X(23:16)	Y(23:16)	X(26:24)/0	Y(23:16)
X(15:8)	Y(31:24)	X(23:16)	Y(31:24)	X(31:24)	Y(31:24)	000	Y(23:16)
X(23:16)	Y(31:24)/0	X(31:24)	Y(31:24)	X(23:16)	Y(23:16)	000	Y(23:16)
X(7:0)	Y(15:8)	X(15:8)	Y(15:8)	X(23:16)	Y(15:8)	X(26:24)/0	Y(15:8)
X(7:0)	Y(31:24)	0		X(31:24)	Y(23:16)	000	Y(15:8)
X(7:0)/0	Y(15:8)	X(15:8)	Y(15:8)	X(31:24)	0	000	Y(15:8)
X(7:0)	Y(7:0)	X(15:8)	Y(7:0)	X(23:16)	Y(7:0)	X(26:24)/0	Y(7:0)
0		X(31:24)	Y(7:0)	X(31:24)	Y(15:8)	000	Y(7:0)
X(7:0)	Y(7:0)	X(15:8)	Y(7:0)/0	X(31:24)	Y(23:16)/0	000	Y(7:0)

10

【0126】

浮動小数点入力は、IEEE 隠しビットを再表示する。非正規性はALUによって0として解釈されるので、対応する浮動小数点モードが選択されたときに、隠しビットを常にアサートすることができる。いずれかの入力で指数が0になると、データは強制的に0になる。浮動小数点に適した隠しビットは、対応するモードが選択されたときに、乗算器のX入力とY入力の両方で「1」にアサートされる必要がある。それ以外の場合、入力は上記の乗数設定に従う。隠しビットを表8に示す。指定されたモードが選択されている場合、隠しビットは強制的に1になる。選択されていない場合、入力は上記の入力変約で表化された入力をトラッキングする。

20

【0127】

【表 8】

表8: モード別浮動小数点隠しビット位置

モード	隠しビット
double	52
single	23
half	26,10
quarter	27,19,11,3
bfloat	23,7

30

【0128】

条件付き被乗数操作は、入力並べ替えキュー350の入力並べ替え論理内で処理され、これは、Y入力符号の関数としてX入力を定数で置き換える。条件付き被乗数はSIMD演算をサポートしていない。

【0129】

G. 乗算器305桁上げ保存加算器ツリー

図17は、ブルーニングされた32ビットSIMD拡張を有する27×27乗算器のドット図である。乗算は、小学校で教えられている長尺乗算と同様の方法で行われるが、10進基数ではなく2進基数を使用して行われる。長尺乗算は、実際には2つの異なる操作、すなわち、部分積の生成と、それらの部分積の累積である。被乗数と乗数の各桁の対ごとに部分積が生成される。結果の部分積は、その部分積に使用される桁の重みの合計によって重み付けされる。被乗数の桁の重みは数系基数の累乗であるので、重みは位置シフトによって達成される。これらのシフトは、部分積の右側に適切な数のゼロを挿入することによって表される。各部分積のシフトpは、被乗数と乗数の位置シフトmとnの合計に等しくなる。2進乗数では、1ビットの部分積は論理積と同じである（1×1の組み合わせのみが0による乗算ではない）。Nビット×Nビット乗算器に対しては、N2個の1ビット部分積が存在し、各ビットは、被乗数および乗数からのビットの一意的な対の結果であ

40

50

る。図17は、上述の一对の 16×16 SIMD乗算器を収容するように増強された 27×27 乗算器を示す。この図の各点は、 1×1 の部分積を表す。同じ積の重みのドットが縦に並ぶように配置されます。このレイアウトは、長尺乗算における部分積を思い起こさせる。ここでは、各行は乗算器の1桁に乗算器の各桁を掛けたものである（この場合、各積は0または1であるため、次の桁への桁移動はない）。

【0130】

整数加算のプロパティは、任意の便利な順序で追加を行うことができる。単純な乗数の構成では、図のシフトを持つ行を追加するだけで、正しい答えが得られる。ただし、この方法は、前の合計が追加されるときに各行に桁上げ伝播が含まれるため、速度や領域に最適ではない。代わりに、列を合計し、最後の加算まで行の桁上げを延期する。同じ重みを持つビットのグループは、フル加算とハーフ加算を使用して合計される（そして、ほとんどのライブラリには、性能を向上させ、面積を減少させることができる、一般に圧縮器と呼ばれる、技術に依存する高次のタリー加算器がある）。

【0131】

全加算器は、同じ重みの3入力ビットを和と桁上げ出力に低減する。桁上げ出力の重みは、合計出力よりも1だけ大きくなる。列の追加の利点は、桁上げが行全体ではなく出力に向かって伝播されることだが、積はベクトルの対（桁上げへの合計に関係するもの、したがって桁上げ保存という名前）に縮小されるだけである。これらのベクトルの合計が乗数積になる。この設計では、積は、性能を維持し、面積を最小限に抑えるために、アキュムレータの後まで桁上げ保存の形で残される。

【0132】

最適ツリーを生成するアルゴリズムはいくつかある。最小の伝搬遅延とゲート数で知られているツリーの1つはダダ（Dadda）ツリーである。伝統的に、ダダツリーは、完全加算器と半加算器のみから構成されるが、技術ライブラリが、離散的な完全加算器と半加算器を使用する場合と比較して、面積を減少させかつ／または伝搬遅延を改善するような圧縮器を含む場合には、高次の圧縮器を使用するように修正することができる。ほとんどの場合、より高次のコンプレッサを使用する利点がある。考察と説明のために、この仕様では従来のダダのツリー構造を使用しています。回路設計者は、柔軟な乗算器のフットプリントを低減するために、高次の圧縮器を自由に使用し、ツリー構造を変更することができる。

【0133】

図17のドット図は、 27×27 モードとSIMDモードとの間の部分積重シフトの設計を説明するために強調される。右側の大きなひし形436は、 27×27 の乗算である。これは、前に説明したブルーニングされた 32×32 SIMD乗算に対応するために、左および下（最上位ビット）に拡張されている。拡張子438は、これらの部分文字が他の場所から借用されていることを示す。借用ドット434はSIMD用である。これらは2つの領域を占有し、1つは高さが5行で、図の下中央付近の行あたり16ビットであり、もう一つはアレイの左上付近の行あたり5ビットである11行である。縦方向に一緒に移動したこれら2つは、延長ドット438と同じ形状を形成する。黒で塗りつぶされたドットは、SIMDモードと 27×27 モードの両方に共通する部分的な積を表し、塗りつぶされていないドットは、SIMDモードで無効になっている（強制的に0になる）ものである。

【0134】

図18は、第1レイヤ層ダダ加算器を有する再構成されたドット図である。図18において、図17のドット図は、 27×27 の乗算器の列を上揃えすることによって再配置される。借用された部分積（442）を表すドットは、対応する列の下部に移動され、青色の塗りつぶしで示されるSIMD拡張に対応するドットは、アレイの残りの部分から垂直方向に分離されるが、重み付けのための正しい列内にある。ここでは、部分積（442）借用領域と同じレイアウトに拡張領域のレイアウトを残し、借用領域と拡張領域との類似性を強調する。ダダツリー縮小では、部分積（442）ドットは、可動部分積が別のツリ

ーに結合されるように、そこにはないものとして扱われる。ダダアルゴリズムは、反復またはレイヤごとに行数を特定の行数に減らすことによって、レイヤを最小化する。3：2加算器の行ターゲットのシーケンスは、19、13、9、6、4、3、2である反復のたびに、列内のエントリ数を減らすために、列に完全加算子と半加算子が配置される。各加算器は、その加算器が入っている列に1つの出力を持ち、次のレベルの左の列に1つの出力を持つ。行数をカウントする場合、前の列には各加算器の行があり、現在の列には各加算器の行があり、現在の列には非表示のドットの行がある。図18のドットはダダ層への入力を表し、3ドットを覆うボックス437は完全加算器であり、2ドットを覆うボックス439は半加算器である。加算子は、このレイヤの加算子を表す。覆われていないドットは、そのまま次のレイヤに移動する。拡張ツリーはすでにカラムあたり10ドット以下であることに注意すべきである。そのため、この最初の反復では、単に拡張全体を変更せずに渡す（すなわち拡張は後のレイヤで始まる）。

【0135】

図19は、可撓性乗算器305のための第2レイヤダダツリーのドット図である。次の層は、各列内に、前のレイヤ内の覆われていない各ドットに対するドット、列から右への各桁上げに対するドット（前のレイヤの右の列内の各加算器に対する桁上げ）、および前のレイヤ内のその列内の各加算器の合計出力に対するドットを有する。図19は、第1のレイヤによって列当たり最大13ドットに縮小された行を示す。拡張配列は前のレイヤと同じですが、縮小プロセスを明確にするために列が上揃えになっている。この反復により、出力が9行に減少する。

【0136】

図20は、フレキシブル乗算器のための第3レイヤダダツリーのドット図である。ツリーの次のレイヤには、行が9行に縮小されて表示され、このレイヤのターゲットは6行です。この第3のレイヤを図20に示す。

【0137】

図21は、フレキシブル乗算器305のためのダダツリーの残りのレイヤのドット図である。図21は、各出力で2つの行に到達する残りのレイヤを示す。2つの行447は、移動体部分積を含まない 27×27 乗算器305からの出力であり、ドット444は、SIMD拡張のためのアラインメントに示された、移動体積のためのツリーの出力であり、ドット446は、 27×27 乗算のための移動体積のアラインメントを示す。4：2圧縮器の1つの追加のレイヤを使用して、静止ツリーと移動ツリーを結合する。移動ツリーは、どちらの接続を行うかを選択するゲートを通じて両方の場所に配線されます。モバイル製品用の加算器は、拡張ツリーの2つの画像が占める列に4：2の圧縮器のみを必要とし、さらに、各グループの最も左の列のすぐ左にある列に全加算器を必要とし、4：2の圧縮器とそれらの左にある列に半加算器を必要とし、出力の灰色で網掛けされた領域によって示される。左側の全加算器の理由は、4：2圧縮器に追加の桁上げアウトと桁上げインがあるためです。追加された緑の点は、4：2圧縮器チェーンからの実行用です。加算器ツリーは、符号補正と否定ベクトルを組み合わせるための選択可能な反転のための加算器（4：2加算器）とXORの追加レイヤを示さない。RAE回路300は、実際の設計において、他のツリー構造アルゴリズムを置換し、より高次の圧縮器を使用する自由度を有する。従来のダダツリーは、移動ツリーと静止ツリーの部分を結合する加算器を含む7レイヤツリーとなる。移動ツリーの深さは出力加算器を含めてわずか6レイヤであるため、移動部分積（移動ツリーの経路にある）を移動するために必要な入力選択ゲートと出力ステアリングゲートは、7レイヤ固定ツリーを通る最長伝搬経路に寄与しない。

【0138】

従来のダダを使用した単純な 27×27 の乗数も7レイヤであるため、モバイルツリーのパフォーマンスの低下はない。ベンダーのライブラリによっては、Daddaツリーは上位のコンプレッサを使用することで改善できる。たとえば、図21のツリーの5番目と6番目のレイヤは、4：2の圧縮器を使用して簡単に1つのレイヤに置き換えることができる。4：2圧縮器には、5つの入力、4つの均等重み付け入力、同じレベルの別のレベ

10

20

30

40

50

ルからの桁上げイン、同じレベルの別のレベルへの桁上げ出力、および次のレベルへの合計と桁上げ出力がある。

【0139】

図22は、レイヤ5およびレイヤ6(447)を1段で実行するために4:2圧縮器を使用することを示す、等価回路448から4:2圧縮器およびレイヤ5のセグメントを示す。桁上げ出力と桁上げの組み合わせは、次のカラムへのジョグであり、その後、他の入力と同様の遅延で次のレイヤに伝播します。図22は、完全加算器448(種々の図では「FA」と省略)を使用する4:2圧縮器のための等価回路を示す。これは、完全加算器の2つのレイヤよりも利点があるようには見えないが、大部分のライブラリにおけるカスケードされた完全加算器よりも高速でありかつより小さい4:2圧縮器の回路レベルの実装がある。

10

【0140】

乗算器305はまた、可動拡張ではなく固定拡張として、 16×16 SIMD拡張で拡張された 27×27 乗算器を含む、プルーニングされた 32×32 から構成されてもよい。移動拡張は桁上げ保存加算器アレイの約14%を構成するので、好ましい実装であると思われる。しかし、レイアウト上の問題により、ゲート数が多くても、追加入力とツリースイッチングを考慮すると、固定プルーニングされた 32×32 が占めるシリコン面積が小さくなる場合がある。回路設計者は、最小面積に対する乗算器構造を選択する際に自由裁量を認められている。さらに、ブース符号化などの製品用語の最適化は、この仕様では考慮されていないが、このような最適化は、全体的な機能が維持され、最適化の結果、物理的なフットプリントが小さくなる場合に許容される。

20

【0141】

5. 乗算器シフトコンバイナネットワーク310

図23は、乗算後乗算器シフトコンバイナネットワーク310の論理ブロック図である。乗算器305の後には、乗算器シフトコンバイナネットワーク310が続き、これは、基数32指数を有するシステムに変換するために、仮数を最大32ビット左シフトするように主に設計されている。基数32の指数を使用すると、臨界アキュムレータループ内での動的アラインメントのシフトが簡単になる。この乗算器シフトコンバイナネットワーク310はまた、SIMDモードのうちの1つである場合に、SIMDレーン309(4つのSIMDレーン309A、309B、309C、309Dとして図示される)を広げて、蓄積および位置合わせシフトのための空間を可能にするように働く。乗算器シフトコンバイナネットワーク310はまた、RAE回路300内のSIMD積の単一クロック和のためにSIMDレーンを別々に整列させ、それらを合計し、またRAE回路クアッド450内の四つのRAE300からの積の合計を合計する能力を含む。

30

【0142】

乗算器シフトコンバイナネットワーク310はまた、32ビット整列された積を32ビットの倍数だけ右シフトさせて、融合乗算加算のための加算Z入力との和を容易にすると共に、積のより大きな和のための同じRAE回路クアッド450内の他のRAE300からの積和と積を整列させる。乗算後ツリーの構造を図23に示す。図示のように、乗算器シフトコンバイナネットワーク310は、複数のマルチプレクサ、シフタ、及び圧縮器(追加の乗算器シフトコンバイナネットワーク310の実施形態310A、310Bと共に、以下により詳細に説明および議論される)を含む。

40

【0143】

要約すると、乗算器シフターコンバイナネットワーク310は、

1. 浮動小数点の積を基数32の指数に変換する(5lsbsの指数の値によって左シフトする)機能と、
2. 被乗数が非正規化右シフトまたは過剰右シフトの場合に、乗数積を強制的に0にする機能と、
3. 0ビット、32ビット、64ビット、または96ビットの右シフトを追加して、他の加数に対する浮動小数点アラインメントを完了する機能と、

50

4. アキュムレータ 315 に一致する 128 ビット（4 つの 32 ビットレーン、2 つの 64 ビットまたは 1 つの 128 ビット）レーン 309 へ SIMD レーンを拡張する機能と、
5. 標識 2 の補完積をレーンの幅に拡張する機能と、
6. SIMD 内積モードの SIMD レーンを合計する機能と
7. 内積モード用に内の他の RAE からの積の合計を追加する機能と、
8. RAE 回路 300 の単一サイクル 32×32 および倍精度乗算の部分積を結合する機能と、
9. INT 64 の乗算をサポートするために、64 ビット製品を 128 ビットの上位、中間、または下位に配置する機能と
を実行する。

【0144】

浮動小数点積の場合、乗算器シフタコンバイナネットワーク 310 は、指数の最下位の 5 つのビットに等しいビット数だけ仮数を左シフトすることによって、データを基数 32 指数フォーマットに変換する。これが行われると、最下位の 5 つの指数ビットは破棄され、残りの指数ビットのみが下流で使用される。

【0145】

乗算器シフタコンバイナネットワーク 310 は、アキュムレータの前に積および Z 入力加算器を加算する。内積モードで動作する場合、すべての SIMD レーンからの積、および同じ RAE 回路クアド 450 内の他の RAE 300 からの合計も合計に加算される。浮動小数点演算では、加算はすべての加数と同じアラインメントを持つ必要がある。つまり、すべての指数が同じである必要がある。指数論理 335 は、すべての加数のうち最大基数 32 の指数を決定し、その指数と最大指数との差によって示される 32 ビットの倍数だけ各加数を右シフトする。指数論理 335 はまた、積と Z 入力とを互いに、および隣接する RAE ユニットからの積が積の和に寄与する場合にそれらの積を整列させるために必要な過剰な右シフトを決定することを担当する。

【0146】

1/4 精度と半精度の指数はそれぞれ 4 ビットと 5 ビットであるため、基数 32 の指数事前シフトは、これらの場合の仮数を完全に整列し、アキュムレータ 315 の前に積を合計するために、さらなるアラインメントシフトは必要ない。この場合、アキュムレータ 315 への仮数は整数として扱われ、アキュムレータは整数モードで動作する。最後の加算／丸め／飽和論理は、整数として残すことも、浮動小数点形式に戻すこともできる。

【0147】

Z 入力 375 は、積の共通和および独立 Z 入力の同時和または差を必要とする関数をサポートするために、積の和と結合される。FFT バタフライは、 $Z \pm (A * \cos + B * \sin)$ を使用した例である。積の和と Z 入力加算器との間には、否定およびゼロ制御が加えられ、これを可能にし、また、積が隣接する RAE 300 内の積の和を供給するために使用されるときに、アキュムレータ 315 と共に Z 入力 375 を使用することを可能にする。5 を超える指数ビット（倍精度および単精度 IEEE および FLOAT16）を持つモードでは、指数を均等化するために追加のシフトが必要である。追加のシフトが必要な場合、指数の小さい加数はそれぞれ右シフトされる（および、IEEE 標準に従って最も近い値に丸められる）。右シフト距離は、32 ビット右シフトの数として、超過指数（基数 32 指数に変換した後に残った指数）を使用して計算される。右シフトの数が 3 より大きい場合、加数は 0 になる。さらに、出力基数点を出力形式の正しい位置に揃えるために必要なモード依存の固定バイアスがある。これら 3 つのシフト量（変換左シフト、アラインメント右シフト、バイアスシフト）を合計して、各加数に対するネットシフト量を決定し、そのネットシフト量をシフターコントロールに記録する。各加数に対する正味シフト決定が計算され、指数論理ブロック 335 内のシフト制御に変換される。単精度の場合は最大 8 個（4 つの RAE（それぞれ Z 入力と製品を含む））、FLOAT16 の場合は最大 12 個の加算器があり、RAE 300 ごとに 2 つの積と 1 つの整数の加算器がある）。

【0148】

図24は、アキュムレータ315の128ビットに対する各モードの乗算器積のアライメントを示す。乗算器積幅は、セル451が乗算器入力幅に対応する複数の網掛けセルによって示される。セル441は、融合していないアキュムレータ内のスティッキービットの位置を示す。この設計では、アキュムレータと加算器は128ビット幅以下のスティッキービットを持っている。セル間の線453は、32ビット境界に位置合わせされたときの浮動小数点の基数点の位置を示す。「X」とラベル付けされたセルは、浮動小数点の符号を示す。整数部の「Y」と表示されたセルは、桁上げブロッカーの位置を示す。

【0149】

ALUとしてRAE回路300内でドット積和がより大きなSIMDレーンに累積され、シフトネットワークによって32ビット境界に整列される。半精度および1/4精度浮動小数点の場合、最大で5ビットの指数があるため、32ビット境界へのシフトは表現可能な範囲全体を表す。したがって、これらのモードでは、指数のさらなるシフトまたは処理は必要ない。Bfloat16は8ビットの指数を持つため、内積モードで積の合計の積を整列するために追加のシフトが必要である。指数論理は、モードに応じて、そのレーンの上位3指数ビットとRAE回路300、RAE回路対400またはRAE回路クワッド450内の最大指数上位3ビットとの間の差に基づいて、BFloat16ドット積モードのうちの1つを選択する（名前の数字対は、それぞれ、上位および下位レーンの追加右シフトを示す）。

【0150】

これらの追加のBFloat16モードは、一方または両方のレーンを0または32ビットだけ右にシフトさせるか、またはより大きな差がレーンをゼロにする。各RAE回路300内の最大指数(3msbs)は、シフトを解決するために、RAE回路クワッド450内の他のRAE300と共有されるべきである。解像度は積のシフトに間に合うように乗算と並行して起こる。単精度ドット積は、加算前に位置合わせするために32ビットシフトが必要かどうかを決定するために、同様のRAE間300指数処理を必要とする。このため、single>>32、single>>64、およびsingle>>96の規定が、シフトコンバイナネットワークの機能テーブルに追加された。BFloatドット積と同様に、単精度ドット積も、追加のシフトを事前に決定するために、RAEALU間の上位指数ビットの分解能を必要とする。

【0151】

IEEE標準の丸めに合わせるために、シフトコンバイナは、128アキュムレータのビット幅に加えて、シングルおよびbfloatドットモードの丸め、ガード、およびスティッキービットも組み込む必要がある（これらはアキュムレータの丸めビットにも渡される）。他のドットモードでは、完全に展開された浮動小数点の整数として扱われるため、double、single、およびbfloatの各ドットモードを除き、加算器ツリーの右側からシフトするビットはない。

【0152】

アキュムレータ内に浮動小数点(double、single、およびbfloat)を保持する浮動小数点モードは、各レーン内で3ビットまたは4ビットずつシフトして、プリアキュムレータ内での成長を可能にする必要がある。アキュムレータ315は、適切な3ビット位置合わせを維持するように適宜調整される。また、プリアドは、シフト時に「1」ビットを右端から押し出すときにフロートのスティッキービットとしてレーンのlsbに設定される。

【0153】

表9は、柔軟な乗算器305の出力構成を、乗算器シフトコンバイナネットワーク310と共に説明する。乗算器シフトコンバイナネットワーク310の論理は、その1次入力データを、プルーニングされた32ビット/27ビットのフレキシブル乗算器305から受け取る。入力は桁上げ保存形式の64ビット（合計が乗算器積となる2つの64ビットベクトル）である。データは、2つの32ビットレーンまたは1つの64ビットレーンを

作成するために必要に応じて組み合わされる4つの16ビットレーン309に分離される。合成された各レーンは、出力符号を決定するために、被乗数記号および否定制御を使用して、負の積に対して否定または符号拡張される。乗算器出力の内容は、表9に示すように、乗算器モードに依存する。

【0154】

【表9】

表9:フレキシブルな乗算器の出力構成

モード	説明
27x27	乗算器出力はビット53:0で54ビット 2つの補数出力は、符号を64ビット幅に拡張したものである。 倍精度は、27の部分的な下位部分積を54ビットずつ右シフトし、中央の部分積をそれぞれ27ビットずつ右シフトして、4つの27x27の部分積を合計して54x54の積を作成することによって、4つの27x27の乗算器を結合する。 積は桁上げ保存形式であるため、54ビットベクトルが2つある。
24 x 24	両方の入力の入力ビット26:24が0であることを除き、27x27と同じ。 出力はビット47:0(桁上げ保存形式の2つの48ビットベクトル) 符号付き演算符号が64ビットの出力幅に拡張
2つの 16x16	乗算器は、2つの16x16乗算器として配置される。 下位乗算器の入力はXビットとYビット15:0、出力はビット31:0である。 上位乗算器の入力はXおよびYビット31:15、出力はビット63:32である。 浮動小数点は各入力の0個の上位Nビットを書式設定するため、積の上位2 * Nビットも0になる。符号付き演算符号は、各サブ乗算器幅の幅まで拡張される。
4つの 8x8	乗算器は、4つの8x8乗算器として配置される。 レーン0の入力はXおよびYビット7:0、出力はビット15:0である。 レーン1の入力はXおよびYビット15:8、出力はビット31:16である。 レーン2の入力はXおよびYビット23:16、出力はビット47:32である。 レーン3の入力はXおよびYビット31:24、出力はビット63:48である。 浮動小数点は各入力の0個の上位Nビットを書式設定するため、積の上位2 * Nビットも0になる。符号付き演算符号は、各サブ乗算器幅の幅まで拡張される。
32 ビット拡張	乗算器は、除去されたX(23:0)*Y(23:0)と関連する24x24の部分積を乗算した、ブルーニングされた32x32乗算器として配置される。 16ビット左にシフトして、24x24の符号なし乗算器で合計すると、この組み合わせは32x32の乗算器になる。

【0155】

この設定にはSIMD制御が含まれており、各レーンの桁上げブロッキングを適切に設定する。他の複数のRAEからの入力に対する静的シフト／0制御も、制御入力に含まれる。(a) 1ビット符号拡張制御(4つのレーンごとに1つ)と、(b) 第1レイヤ4:2圧縮器の2ビット乗算器SIMDサイズ選択制御と、(c) 2ビットアキュムレータSIMDサイズ選択制御(残りの圧縮器レイヤ用)と、(d) 対入力の他のRAE300の2ビットシフト／0選択と、(e) 2ビットシフト／クアッドの残り半分からの入力に対する0選択と、(f) Z入力加算器への入力における積和の2ビットネゲート／0制御との静的構成コントロールが含まれるか、またはデコードされる。否定およびゼロ制御は、サフィックス(および／または条件付きフラグ)入力380論理を用いて動的に制御可能である。

【0156】

各シフトセレクトに対するシフト制御は、指数論理335ブロックによって設定され、指数値および構成を用いて、図23のブロック図に示される各シフトセレクト選択に対す

る適切なシフト設定を計算する。各レーンのゼロ化制御はまた、指数値および構成を使用してレーンがゼロ化されるかどうかおよびいつゼロ化されるかを決定する指数論理 3 3 5 から得られる（非正規化またはゼロ乗算または 1 2 8 ビットアキュムレータを超える右シフトのためゼロ化される）。ここには、次のシフト制御が含まれる。

1. 5 ビット 0 : 3 1 ビットの左シフト制御で、4 レーン 3 0 9 毎に設定される。レーンは、5 ビット符号なしバイナリ制御で指定されたビット数だけ左にシフトされる。
2. 1 ビット 0 / 3 2 ビットの右シフト制御で、4 レーンに 1 つずつ。「1」を指定するとシフトが発生し、「0」を指定するとデータは変更されずに渡される。
3. 4 つのレーンごとに 1 つずつ、1 ビットのゼロサイズ制御。「1」はレーンを 0 にし、「0」はデータを変更せずに渡す。
4. 1 ビット 0 / 3 2 ビット右シフト制御（2 つのダブルレーンごとに 1 つずつ）。「1」を指定するとシフトが発生し、「0」を指定するとデータは変更されずに渡される。
5. 最上位 3 2 ビットレーンの 1 ビット 0 / 6 4 右シフト制御。「1」を指定するとシフトが発生し、「0」を指定するとデータは変更されずに渡される。
6. 最下位 3 2 ビットレーンの 1 ビット 0 / 6 4 左シフト制御。「1」を指定するとシフトが発生し、「0」を指定するとデータは変更されずに渡される。

【0 1 5 7】

Z 入力シフタ 3 3 0 ブロックの算術出力からの Z 軸加算器は、1 2 8 ビットの和ベクトルと、Z 入力の 4 つの 3 2 ビットレーンの各々に対する 2 の補数増分からなる 4 ビットの桁上げベクトルとである。

桁上げベクトルビットは、Z 入力を加える 3 : 2 圧縮器 4 5 6 に先行する積データ経路上の 4 : 2 圧縮器段階 4 5 4 からの前のビット位置からブロックされる各アクティブレーンの L S B への桁上げ入力を置き換える。

レーンが結合されると、その桁上げベクトル L S B は、次の下位レーンの桁上げ出力から取得される。

Z 入力 3 7 5 は、Z 入力シフタ 3 3 0 ブロックの算術出力をオフにすることによって無効にされ、それによって出力は強制的にゼロになる。

【0 1 5 8】

リンクされた対の R A E 3 0 0 内の他の R A E 3 0 0 からの入力およびその同じ R A E 3 0 0 への出力は、両方とも桁上げ保存形式の 1 2 8 ビット幅信号（2 つの 1 2 8 ビットベクトル）である。入力には、シフトされていないデータと 2 7 ビットのデータで右シフトされたデータの間で選択するセレクタがある。このセレクタはまた、他の R A E 3 0 0 からのリンクが望まれない場合に入力をゼロにする選択を有する。入力シフト m u x は、デコードされた構成ワードによって制御される。Z 入力加算器からの和は、対の他方の R A E への出力である。Z 入力加算器からのその信号も、他の R A E 3 0 0 からの入力と加算される。内積モードは S I M D レーンを他の R A E 3 0 0 への出力の前に 1 つの値に予め結合するので、これらの接続は常に単一レーンである。

【0 1 5 9】

R A E 回路クワッド 4 5 0 の他の半分からの入力および他の半分への出力も、桁上げ保存形式の 1 2 8 ビット幅信号（2 つの 1 2 8 ビットベクトル）である。入力はシフトされていないか 2 7 ビットのデータだけ右シフトされたかを選択するシフト多重を持っている。この m u x には、クアッドの残り半分からのリンクが望ましくない場合に、入力を 0 にする選択もある。入力シフト m u x は、デコードされた構成ワードによって制御される。R A E 回路 3 0 0 の対加算器からの和は、対内の他の R A E 3 0 0 への出力となる。R A E 対加算器からのその信号はまた、四辺形の他の半分からの入力と加算され、アキュムレータ出力を形成する。内積モードは S I M D レーンを他の R A E 3 0 0 への出力の前に 1 つの値に予め結合するので、これらの接続は常に単一レーンである。

【0 1 6 0】

アキュムレータ 3 1 5 への出力も、桁上げ保存形式の 1 2 8 ビット出力（2 つの 1 2 8 ビットベクトル）である。出力は、S I M D 演算のために 2 つの 6 4 ビットレーンまたは

10

20

30

40

50

4つの32ビットレーンに分離される。1つまたは2つのレーンアキュムレータ出力は浮動小数点値であり得るので、指数論理335からの2つのレーンに対するアキュムレータ指数出力が付随する。上側レーンの指数は、倍精度を含む単一レーンの指数としても機能するため、6ビットの基数32の指数である。下のレーンは、B F L O A T 1 6 S I M Dモードでのみ使用され、3ビットの基数32の指数を持つ。アキュムレータ315の出力はまた、アキュムレータへのデータ有効出力を有し、これは、被乗数データ有効値とZ入力加数データ有効値とのANDの遅延コピーである。無効になっている入力のデータの有効値は、有効な入力がない場合を除き無視される。

【0161】

乗算後シフトコンバイナ回路動作および動的制御入力は、指数論理ブロック335によって部分的に供給される構成ワードによって制御される。この構成は、レーン桁上げブロッカーを設定し、静的シフト/0は、他のR A E 3 0 0からの入力を選択する。動的制御は、レーンの組み合わせとアラインメントのプロセスで、各レーンおよびシフトのシフト距離を設定する。乗算器からの64ビット入力は128ビットに拡張され、浮動小数点モードの場合、シフトオフセットも導入されて、128ビットワード内の浮動小数点値を適切に整列させ、指数シフト左は仮数をさらに基数32指数に変換する。S I M Dモードでは、各レーンの幅が乗数出力の2倍になる。各入力レーンのシフトはモードに依存して変化し、そのシフト距離は指数基数シフトを加算するベースとなる。適切なレーンおよび指数=0のアラインメントを得るためのレーン拡張を表10に示す。

【0162】

シフトに入る各16ビットレーン間の16ビット間隔の乗算器からの出力。より広い複数のレーンのために複数のレーンが組み合わせられると、複数のレーンはギャップを埋めるように移動する。S I M Dドット積の場合、全てのレーンは出力の最下位レーンの位置にシフトされ、合計される。点4については、各レーンは合計前に符号が拡張されている。点2の場合、各入力レーンは合計前に32ビットに符号拡張される。

【0163】

【表10】

表10: シフトコンバイナ入力から出力へのモード別アラインメント

出力ビット 呼び出し	入力 アライン	出力						
		4レーン	2レーン	シングル	Int32	ダブル	ドット4	ドット 2
127:112								
111:96	63:48	63:48				63:48		
95:80			63:48	47:32		47:32		
79:64	47:32	47:32	47:32	31:16		31:16		* *
63:48				15:0	63:48	15:0		
47:32	31:16	31:16			47:32			
31:16			31:16		31:16			
15:0	15:0	15:0	15:0		15:0		* * * *	* *

【0164】

レーンシフトは、モード依存シフトバイアスでレーン指数4または5 l s b sを合計することによって計算される。その合計の下位5ビットは、最初の0:31シフトを直接制御する。バイアスされた指数の上位ビットは、レーン1および3に対する超過シフト論理からの超過シフト距離に加算され、次いで、その和が、コンバイナ段階におけるマルチプレクサに対するシフト制御に記録される。図25は、各レーンおよび各第2レベルシフトのモード毎のシフト距離を示す。図25のいくつかのモードは、モードに対する追加の改良を有する。浮動小数点ドット4の合計は、そのサイズの最も左の出力レーンで1/4精

度または半精度のアラインメントにマップできる。これには、異なるシフトバイアスが必要である。BFLOATドットモードでは、基数32の指数を等化するために必要な余分なアラインメントシフトが考慮される。2つの数値は、指数論理によって決定された上レーンと下レーンの相対的な右シフトを示します。

【0165】

シフト量は、指数に応じて32ビットレーンごとに異なる。同様に、シングルおよびダブルは、可能な4つの32ビットシフト距離に対して異なるシフト設定を有し、Z入力または他のRAEに加算するときにゼロでない右シフトデータをもたらす。これらもまた、最大指数からの指数差によって制御されるシフト設定の変調を必要とし、したがって、テーブル内に別個の行エントリを有する。INT64には3つのシフト設定があり、部分積のシフト-追加累積の4サイクルをサポートする。64×64乗算シーケンサ (multiply sequencer) は、乗数シーケンスと連携してこれらのシフト設定をシーケンス処理する必要がある。

【0166】

図23のブロック図は、乗算器シフタコンバイナネットワーク310のアーキテクチャを示す。この回路は、各レーン309のシフタを個別に制御して、シフタに示されたレーンシフトを行う。レーン0 (309A) および2 (309C) は、64ビットバスに対して-16ビットから47ビットの左シフト範囲を有する。レーン1 (309B) およびレーン3 (309D) の左シフト範囲は-32ビットから31ビットである。各レーンシフタは、各データがシフトされていないか、または前の段階よりも1大きい2の累乗だけ左にシフトされるように配置された6つの2:1マス (図26の316) のシリーズを含む。最後の段階は、偶数レーンの最後の段階が右または左に16ビットシフトし、奇数レーンの最後の段階が32ビットだけ右にシフトするか、またはシフトしないように、バイアスが組み込まれている。最終段階の出力は、0制御がアサートされると、レーン出力を強制的に0にするようゲート制御される。最初の段階は16ビット幅で、乗算器レーンからの16ビットに最上位ビットを加え、符号なしと重複のためにゲートされる。

【0167】

後続の段階はビットシフトの量だけ広くなり、入力最上位ビットが出力幅を埋めるように複製される (その入力ビットは入力からのゲート拡張符号である)。段階の幅は、それぞれ18、20、24、32、48、および64ビットである。これらのレーンシフトネットワークの出力は、1つは2つの低次レーン用であり、もう1つは2つの高次レーン用である一対の64ビット2:1圧縮器 (マルチプレクサ311) に供給される。SIMDドットモードに使用される場合、2つのレーンのシフト距離は、レーンがオーバーラップし、4:2圧縮器313内で合計されるように設定される (データはすべてシフトコンバイナを通して桁上げ保存形式である)。この場合、符号は64ビット幅に拡張されます。他のモードでは、適切な符号拡張を含むシフトされたデータは重複しないので、和は、2つのレーンを結合するためにORゲートが使用された場合と同じである。第1圧縮器段階は、バス幅がドットモードの最大合計の2倍に拡張されるので、オーバーフローを処理するためにビットを必要としない。

【0168】

図26は、レーンシフトおよび第1圧縮器回路の詳細を示すブロック図である。図27は、レーン桁上げブロッキングのための4:2圧縮器への追加論理を示すブロック図である。指数論理によっても制御される第2のシフタ452の層は、32ビットの倍数だけ2つの結合されたレーンを別々にシフトする。下位レーンのシフト範囲は、-32から64ビット左シフトである。上位レーンのシフト範囲は、-96から0ビット左シフトである。これらのシフトは、図24に示すように、2つの縦続接続された2:1マルチプレクサ449を用いて行われる。データ経路幅は、第1のシフタへの入力において、上側および下側レーンの両方に対する符号ビットのゲート付き複製を用いて96ビットに符号拡張され、第1のシフタは、データを変更せずに通過させるか、または両側レーン上で32ビットだけ左シフトさせる。複数のレーンは独立して制御される。入力は第2レベルシフタへ

の入力で128ビットに拡張される。下のレーンは符号が128ビットに拡張され、上のレーンは0が128ビットに拡張される。また、ドットの組み合わせによってレーン幅が2倍になるため、オーバーフロービットも不要となる。シフトされたレーンの出力は、128ビット4:2圧縮器461を用いて合計される。その圧縮器461の出力は次に、シフトされたZ=入力を有し、次に、RAE回路対400内の他のRAE300からの合計を有し、次に、一連の4:2圧縮器454によって、RAE回路クアッド450の他の半分の合計が順にそれに加えられる。Z入力は、その桁上げベクトルがちょうど4つのレーン桁上げビットであるので、3:2圧縮器456を使用する。シフトコンバイナの各圧縮器は、SIMDレーンが有効であるときに活性化されるレーン境界を横切る桁上げをブロックするゲートを有する。第2レベル圧縮器の桁上げゲートは、SIMDが内部桁上げをブロックするとき、Z入力からの桁上げに接続する。図27は、レーン桁上げブロッキングのための追加論理を示す。加算器を結合するために使用される圧縮器は、ガード、ラウンドおよびスティッキービットを実装するために、最下位端に3つの追加ビットを必要とし、融合演算のIEEE丸めに準拠する。

【0169】

図28は、乗算器シフトコンバイナネットワーク310Aの代替実施形態を示すブロック図である。可能な代替構成は、乗算器305の出力に桁上げ伝播加算器702を追加して（各レーンの桁上げブロックのゲーティングと共に）、乗算器ツリーの桁上げ保存形式を非冗長2の補数（単一の64ビットベクトル）表現に解決する。これにより、桁上げ保存出力に必要な最初のレイヤでのシフトツリーの重複が排除され、多くの2:1マルチプレクサインスタンスが節約される。2つの単一ベクトルシフトされたレーンは共に桁上げ保存対を表すので、レーン対を合計するために使用される第1段階4:2圧縮器も除去される。これにより、多くの4:2圧縮器インスタンスが削除される。付加された桁上げ伝播加算器アーキテクチャがより少ない論理を占有し、かつタイミングを満たす限り、これはかなりの量のシリコン面積を節約するはずである。最初のレベルの後の論理は変わらない。また、レーンゼロ化は、データ経路がより狭い乗算器出力に押し戻すことができる。図29は、乗算器シフトコンバイナネットワーク310Bの別の代替実施形態を示すブロック図である。

【0170】

6. Z入力シフト330

図30は、Z入力回転/シフト論理330データ経路を示すブロック図である。RAE300は、アキュムレータ/加算器への加算入力、並びにブール論理回路325及び比較回路320への入力として機能する第3の32ビットZ入力375を含む。このZ入力は、隣接するRAE300（対の他のRAE300）のZ入力375またはこのRAE300のY入力370のいずれかをZ入力375に連結することによって、64ビットに拡張可能である。拡張ソースの選択は、RAE300モード設定によって制御される。Z入力375は、乗算器積アラインメントに一致するようにアラインメントシフトを実行するように設計されたシフトネットワークによって調整される。このシフトネットワークは、ワードサイズが8ビット、16ビット、32ビット、または64ビットの場合に必要なブールシフト/回転関数によっても使用される。ブール論理回路325の入力はまた、min-max比較およびソートのための比較ブロック入力の1つに接続される。Z入力375およびZ入力回転/シフト論理330は、算術演算および論理演算の両方のため1レーン、2レーンおよび4レーンSIMD演算をサポートする。Z入力375に関連するZ入力回転/シフト論理330は、

1. 各SIMDレーンの浮動小数点入力の隠しビットを回復する関数と、
2. 浮動小数点モードで非正規（指数=0）入力が提示された場合の出力データを0にする関数と、
3. 加算器に合わせてSIMDレーンを128ビット（4つの32ビットレーン、2つの64ビットまたは1つの128ビット）に拡張する関数と、
4. 浮動小数点の符号の大きさを2の補数仮数に変換する関数と、

5. 2の補数出力をレーンの幅に符号拡張する関数と、
6. 8、16、32、および64ビットレーンでの整数論理シフトおよび回転操作する関数と、
7. 64ビット入力を128ビットの32ビット境界の上、中央、または下の位置に配置することが可能な関数と、
8. レーンごとに0、反転、または否定の演算出力を選択する関数と、
9. レーンごとに0または反転論理出力を選択する関数と、
10. 絶対値関数と、
11. 浮動小数点数を基数32の指数に変換する(5 l s b sの指数で左シフト)関数と
12. 0ビット、32ビット、64ビット、または96ビットの右シフトを追加して、乗数に対する浮動小数点の調整を完了する関数と、
13. レーンの並べ替え(8ビットレーン)関数とをなす。

【0171】

典型的なZ入力375は、単精度浮動小数点、SIMD-2半精度浮動小数点、SIMD-4 1/4精度浮動小数点、SIMD-2 B F L O A T-16、32ビット整数、SIMD-2 16ビット整数、またはSIMD-48ビット整数のいずれかの形式の32ビットである。整数は符号付きでも符号なしでもよい。論理シフト/回転演算およびブール論理回路325への入力および比較320ブロックは、符号なし整数に対してのみ有効である。他の形式は通過できるが、論理モードの場合は、このブロックで符号なし整数として扱う必要がある。

【0172】

Z入力375は、拡張が64ビット入力の最上位32ビットになるように、このRAEのY入力370または他のRAE300のZ入力375のいずれかをこの入力の左側のRAE対に連結することによって、64ビットに拡張することができる。64の入力は、符号付き整数、符号なし整数、またはIEEE倍精度浮動小数点である。64ビットとして使用する場合、拡張機能の最上位12ビットは倍精度浮動小数点の符号と指数である。拡張は、論理演算および出力では整数として扱われる。拡張入力はまた、4つの主レーンに対するシフトバイアスに64のオフセットを加え、レーン4に対するシフトバイアスに-64のオフセットを加えることによって、拡張が最下位32ビットになるように連結されてもよい。隠しビットと指数マスクビットも、上下に入れ替える必要があります。二重指数論理は高いx*高いyRAEに存在するため、Z入力分割がxまたはy入力分割と一致するようにするには、この拡張交換が必要となる。

【0173】

コンフィギュレーション入力は、各レーンのシフトまたはシフトバイアスの設定、マスクの選択、動作モードの設定、論理出力の論理シフト、回転または0および反転の選択、符号付きまたは符号なし、ネゲート、絶対値、および算術出力の0の設定を行う。否定および0は、内部条件フラグまたはZ記号によって制御することもできる。コンフィギュレーション入力は、各レーンのシフトまたはシフトバイアスを設定し、マスクを選択し、動作モードを設定し、論理出力用に論理シフト、回転または0および反転を選択し、算術出力用に符号付きまたは符号なし、否定および0を設定する。

【0174】

整数フォーマットのシフト距離は、構成内のビットによって、またはRAEのX入力365を介して適用される外部シフト制御によって制御される。X入力を選択されると、32ビットX値は、各8ビット入力レーンに対応する複数の48ビットシフト制御に分離される。各制御レーンの下位7ビットはシフト距離に対応し、8番目のビットはレーン無効(lane disable)である。SIMDモードが4に設定されている場合、Xコントロールの4つのレーンは、Z入力の4つのレーンに1:1に対応する。SIMDモードが2または1に設定されている場合、シフトシェアリングは、SIMDレーンの最上位バイトに関連付けられた制御レーンを使用して有効になる(制御ビット15:8は下位1

10

20

30

40

50

6ビットレーンを制御し、ビット31:24は上位16ビットレーンを制御し、ビット31:24は単一レーンSIMDのすべてのレーンを制御する)。同様に、制御ワード固定シフト設定も同様に分割された32ビットワードである。浮動小数点演算の場合、Xシフトは浮動小数点指数に加算し、指数を増減する手段を提供する。シフト値が0(レーンのシフト制御とシフトバイアスの合計)の場合、付随するレーンのデータは、128ビットの内部ワードのビット0の最下位ビットに配置される。非ゼロシフト値は、入力のLSBがシフト制御とシフトバイアスの合計に等しい128ビット内部ワードのビットになるようにデータを左シフトする。動的制御は、指数論理335および構成によって生成される。

【0175】

算術出力は、乗算後シフトコンバイナ310論理へのZ入力である。この出力は、桁上げ保存形式の128ビット幅の2の補数(2つの128ビットベクトル)である。出力はアキュムレータ入力と一致するようにフォーマットされる(ドットおよび複素積モードの場合は、乗数SIMDモードとは異なる)。出力は、単一の128ビットレーン、2つの64ビットレーン、または4つの32ビットレーンにすることができる。4レーンの出力は、厳密に符号付きまたは符号なし整数である(1/4精度と半精度のIEEE floatは、指数の5lsbsだけ左シフトして整数に完全に展開される)。2レーンSIMDは、BFLOAT16 SIMDの乗算-加算-累積の場合は2つの独立したBFLOAT16であり、それ以外の場合は符号付き整数である。BFLOAT16の出力は、アキュムレータにおいて指数の最下位5ビットを0にシフトしたままである。単一の128ビットレーンは、基数32の指数に左シフトされた単精度または倍精度のfloatであり、他の加数に合わせて必要に応じて32ビットの倍数だけ右シフトされる。または、単一の128ビットレーンは128ビットの符号付き整数である。演算出力128ビット合計ベクトルは、シフトされ、マスクされ、渡され、反転またはゼロ化されたZ入力である。桁上げベクトル出力には、各レーンの最下位ビット位置にある+1の2の補数インクリメントのみが含まれる。その他のビットは暗黙的に0であるため、桁上げベクトルには0以外のビットが4つしかない。桁上げベクトルビットは、シフト-コンバイナネットワークにおけるZ入力加算器の前の乗算器シフト段階からの常に0の最下位ビット出力を置き換える。

【0176】

Z入力回転/シフト論理330の論理出力は、ブール論理回路325および比較論理320に接続される32(拡張されると64ビット)論理シフト出力である。これは、シフトされマスクされたZ入力であり、レーン幅だけシフトされたシフトされた入力の第2の画像を有し、回転を達成するためにZ入力シフトされたデータと任意選択的にORされる。この出力は反転(シフトのみ、回転は不可)またはゼロにすることができる。論理出力には2の補数インクリメントはない。これは、インクリメントのための高価な桁上げ伝播加算器を回避するためである。論理出力は、符号付きまたは符号なし整数入力専用である。符号付き整数入力は、回転またはレーンの再割り当て操作に使用すべきでない。符号拡張は、これらの機能の適切な操作を妨げるからである。符号拡張シフトデータにおけるシフト結果に対する符号付き入力。符号なし入力では、シフトの符号は拡張されない。シフトされていないデータは、そのまま(シフト距離は入力車線の幅によってバイアスされるため、0シフト入力は車線の幅によって右シフトになり、結果として0または拡張符号が生成される)入力から論理出力に渡される。X、Y、Z入力データ有効フラグは、Z入力論理では内部的に使用されない。

【0177】

Z入力動作および動的入力のソースは、指数論理335ブロックによって部分的に供給される構成ワードによって制御される。この構成では、浮動小数点、符号拡張、シフト距離またはシフト距離のソースの入力マスク、レーンマスク、反転/ネゲート制御またはそのソース、回転/シフト関数の選択、レーンの無効化、算術および論理出力の有効化(無効化された出力は強制的に0になります)、論理ウィンドウの選択、各レーンの指数ソースの選択、指数入力マスク、およびシフトバイアスを選択する。図31は、Z入力375

のモード別構成を示す。

【0178】

また、次のようなデータ経路制御装置をRAE回路300に設けることもできる。

1. 入力マスク：入力マスクはマスクされたビットをゼロにして指数をフィルタリングし、浮動小数点モードのシフトデータからビットを符号化する。ゼロ指数またはレーン無効化（レーンをゼロにすることができるデータ経路にはいくつかのポイントがあるが、エントリポイントは設計者の裁量に任されている）のためのゼロレーンの手段として使用できる。入力マスクは、SIMDモードと入力フォーマットによって設定される。

2. 隠しビット：浮動小数点フォーマットが選択されている場合、隠しビットは浮動小数点隠しビットの位置に「1」ビットを強制する。隠しビットは、レーンのゼロ化によって上書きされる。隠しビットはSIMDモードと入力フォーマットに依存する。

3. 符号拡張：8ビット入力レーンごとに符号拡張制御がある。設定されている場合、シフタネットワークは、128ビットシフタの幅までそのレーン内を符号拡張する。レーンを組み合わせて16、32、または64ビットの入力レーンを作成する場合は、最も重要なレーンのみに符号拡張を設定する必要がある。符号拡張は符号付き整数モードにのみ使用し、レーン内に符号拡張を含めるためにSIMDモードに適したレーンマスクと組み合わせて使用する必要がある。符号拡張制御はSIMDモードで設定され、format = signedと入力する。

4. シフト距離ソース：シフト距離ソースコントロールは、シフト距離のソースがコンフィギュレーションビットとRAE X入力のどちらであるかを選択する。

5. 構成シフト距離：シフト距離ソースが環境設定に設定されている場合に、各レーンの固定シフト距離を設定する。シフト距離は、レーンのシフト距離に到達するために、各レーンのシフトバイアスおよび指数シフトに加算される。シフト距離ソースがX入力に設定されている場合、コンフィギュレーションのシフト距離は無視される。

6. シフトバイアス：各レーンの7ビットのシフトバイアスの値がシフト距離と指数シフトに追加され、8ビットレーンごとの合計シフト設定に到達する。

7. 浮動小数点整列：浮動小数点整列は、double、single、およびbfloatモードのアラインメントのための追加の32ビット右シフトの倍数のソースを示す。これは浮動小数点形式で設定される。

8. シフト共有：シフト共有は、8ビットレーンごとにシフトを制御するレーンを選択する。これにより、シフト制御を複数の8ビットレーンで共有できるため、各レーンでシフト設定を重複させる必要がない。

9. レーンマスク選択：レーンマスクは、各レーンのシフタの出力に適用される32ビットの固定レーンマスクの中から選択する。オフ、2レーン、4レーンを選択できる。

これはSIMDモードで設定され、レーンの再割り当て機能で上書きされる。

10. Negate/Invert（否定/反転）：ソース選択によって否定/反転のソースが選択され、各レーンのビットによって32ビットの出力レーンが制御される。ソースはすべてのレーンに共通であり、Xの構成ワード、否定フラグ、およびレーンのmsbの設定が含まれる。レーンソースが「1」のとき、レーンは反転され、算術出力が有効であれば、レーン桁上げビットが演算出力にアサートされる。レーン反転コントロールは、ソース車線のシフト共有の選択に従う。複数のレーンがSIMDモードごとに結合される場合、桁上げは最下位レーンに対してのみアサートされる。

11. 回転/シフトMode：1つの構成ビットで、回転イメージをシフトイメージと組み合わせて論理出力するかどうかを設定する。「1」は回転モードを示す。論理出力が無効になっている場合、回転設定は関係ない。

12. 回転幅：回転レーンの幅は、SIMDおよび64ビット拡張命令で選択したレーンの幅に設定される。

13. 演算および論理を有効にする：2ビットを個別に使用して、算術および論理出力を有効にする。「1」の値は出力を有効にする。出力が無効の場合、出力は強制的に0になり、対応する有効なデータ出力は0に保持される。

10

20

30

40

50

14. 論理ウィンドウ：論理ウィンドウの設定では、論理シフト／回転出力に出力するビットを選択する4つのウィンドウのいずれかを選択する。設定は、論理出力SIMDから決定される。

15. 指数マスクー指数マスクは、複数の指数幅が適用される各レーンのZまたはY入力から指数論理に渡されるビット数を選択する。

【0179】

マスクは浮動小数点形式で選択される。フォーマット、SIMD、機能に加え、回転選択、出力イネーブル、およびシフト距離を最小制御入力にカプセル化した14のモードがある。倍精度モードでは、各被乗数の指数ビットにアクセスするために、上位*上位部分積を処理するRAEの指数論理が必要である。これには、64ビットの下半分に64ビット拡張を追加し、64ビットZ入力の上半分に4つのローカルレーンを追加する必要がある。これにより、指数マスクも適切な位置に配置される。これを実現するために、シフト距離が変更される。ダブルHL反転モードはこれを反映している。

【0180】

図30を参照すると、Z入力回転／シフト論理330は、5つの入力レーンの各々について、可変0-127ビット左シフト458を含む。4つのレーンは、4つのレーンSIMDに対応する8ビット入力を有する。レーンは2レーンと1レーンの組み合わせである。5番目のレーンは、64ビット操作の拡張入力に接続された32ビット幅のレーンである。シフトの前には、モードに応じて変化するマスク462があり、入力時の浮動小数点の指数ビットおよび符号ビットをマスクすることを意図している。シフトされたデータレーンは、論理ORゲート464と組み合わせられ、否定又は符号の大きさを負の2の補数出力に変換するために反転される。反転は、32ビット算術出力レーンごとに行われる。32ビットの算術レーンは、論理出力用に8ビットのレーンにマップされる。

【0181】

選択的に反転された出力は、演算への桁上げ保存フォーマットされた出力の合計部分を形成する。その出力の桁上げベクトルには、必要に応じて各レーンの2の補数補完の増分が含まれる。桁上げ出力と保存出力の合計は、各レーンのシフトされたZ入力の2の補数表現であり、否定制御が設定されている場合は否定される。0制御を設定すると、出力は強制的に0になる。0は、指数が0に等しい浮動小数点入力の場合、または各レーンのゼロ化構成コントロールに回答してアサートされる。Z入力論理パイプライン待ち時間は、入力からZ入力へのシフト／コンバイナへの乗算器および乗算後シフト／コンバイナネットワーク待ち時間と一致する。シフトが0の場合、レーン入力のLSBがすべてのレーンの128ビット出力のビット0にマップされるように、Zシフトは0ベースにされる。入力を0にすると、シフトとOR結合器の間に別のマスクを追加してレーンを右にシフトすることができ、また、任意の並べ替えとレーンの結合（例えば、2-4の精度と半精度のレーン）、または仮数と指数の分割処理も可能になる。

【0182】

各レーンは、浮動小数点符号ビットおよび指数ビットをゼロに置き換え、シフトへの入力で隠しビットをアサートするマスク462を有する。レーンおよびモードごとのこれらのマスク値は、0に強制される入力ビットに対応する「1」ビットとともに表11に詳細に示されている。マスクビットのセンスは、実装された設計において便宜上反転されてもよい。また、この表には、マスクをすべて「1」にし、レーンデータを0にするゼロサイズ制御も示されていない。ゼロ化は、レーンが選択解除された場合、およびモードとレーンに適した浮動小数点指数が0の場合にアサートされる。マスク生成は、Z入力指数および制御論理の一部である。

【0183】

Z入力SIMDモードは、必ずしも乗数SIMDモードと同じではない。SIMDドット積および複素乗算モードの場合、乗算器コンバイナは、レーンの数を減らし、Z入力およびアキュムレータにおいて、乗算器のモードとは異なるモードをもたらす。浮動小数点モードでは、IEEEおよびBfloat形式の一部である隠しビットのアサーションが

10

20

30

40

50

必要である。隠しビットは、浮動小数点モードが選択されたときに、関連するレーンでアサートされる。隠しビットは、0の指数によって0に強制される場合、または0の構成制御に対してレーンを強制する場合を除き、常に「1」である。隠れビットベクトルは、表12のモード別に表化されている。定型入力論理はマスクされて $\leq$ (隠しOR (Z - i n A N D N O T m a s k))、ゼロ化されない。

【0184】

【表11】

表11: Z入力シフト論理入力モード別マスク

Z-モード	レーン4のマスク	レーン3のマスク	レーン2マスク	レーン1のマスク	レーン0マスク
ダブル	0xFFF00000	0x00	0x00	0x00	0x00
ダブルHL反転	0x00000000	0xFF	0xF0	0x00	0x00
シングル	0xFFFFFFFF	0xFF	0x00	0x00	0x00
半精度	0xFFFFFFFF	0xFC	0x00	0xFC	0x00
1/4精度	0xFFFFFFFF	0xF8	0xF8	0xF8	0xF8
Bfloat16	0xFFFFFFFF	0xFF	0x00	0xFF	0x00
INT64	0x00000000	0x00	0x00	0x00	0x00
INT32	0xFFFFFFFF	0x00	0x00	0x00	0x00
INT27	0xFFFFFFFF	0xF8	0x00	0x00	0x00
INT24	0xFFFFFFFF	0xFF	0x00	0x00	0x00
INT16	0xFFFFFFFF	0x00	0x00	0x00	0x00
INT8	0xFFFFFFFF	0x00	0x00	0x00	0x00

【0185】

【表12】

表12: Z入力シフト論理入力の隠しビットバイモード

Z-モード	レーン4のマスク	レーン3のマスク	レーン2マスク	レーン1のマスク	レーン0マスク
ダブルHL	0x00000000	0x00	0x10	0x00	0x00
シングル	0x00000000	0x00	0x80	0x00	0x00
半精度	0x00000000	0x04	0x00	0x04	0x00
1/4精度	0x00000000	0x08	0x08	0x08	0x08
Bfloat16	0x00000000	0x00	0x80	0x00	0x80
すべてのINTモード	0x00000000	0x00	0x00	0x00	0x00

【0186】

図32は、シフトネットワーク458の構成を示すブロック図である。各レーンのシフトネットワークは、レーンを128ビットに拡張し、Z入力指数論理で発生する各レーンの7ビット制御に基づいて、レーンデータを0から127ビット位置に左シフトする。ビットが左にシフトされるのと同じように、0ビットが右にシフトされて埋められる。各レーンのシフタは7レイヤの2:1マスであり、それぞれが前のシフタよりも1大きい2の累乗だけシフトする。シフトネットワークの幅を最小にするために、最小のシフトが最初に行われる。

【0187】

シフトネットワーク458の構成を図32に示す。1段階目の幅は入力幅+1、2段階目の幅は入力幅+1+2、3段階目は入力幅+1+2+4となる。すべての段階の幅は1

28ビットに制限されているため、128ビットを超えるビットは失われる。レーン0～3の入力はそれぞれ9ビット（8ビット＋符号拡張用拡張）で、レーン4には32ビット＋符号拡張入力（64ビットのみに使用）がある。シフト制御はモードの関数であり、浮動小数点モードの場合は各レーンの指数、Z軸シフト制御の場合は各レーンの指数である。シフト制御は、Z入力指数論理から取得される。符号付き整数モードの場合、シフトネットワーク458は、入力をレーンの幅に符号拡張する。これを行うために、シフトの各段の幅が1ビット増加され、その拡張されたビットが、シフトされていない入力上の2k最上位ビットで次のmuxに複製される。最初のレイヤmuxは、符号付き整数モードの最上位ビットと重複するように追加ビットを設定し、それ以外の場合は0に設定する。2つのレーンと1つのレーンの符号付き整数の場合、レーンの最も重要なレーンのみが結合され、幅の広いレーンの符号が拡張される。下位8ビットのサブレーンの拡張入力ビットは「0」に設定される。

【0188】

各シフトネットワーク458レーンの出力は、出力をレーン内に維持するためにレーンマスクを必要とし、さもなければ、符号拡張は次の最上位レーンに伝播し、浮動小数点整列のための右シフトは、より低いレーンにアンダーフローする。SIMDモードの各レーンのデータを通ずるビット範囲は、以下の表13に示されている。マスクが有効になっている場合、表示されているビット範囲外の範囲は強制的に0になる。演算モードではレーンマスクが必要である。論理モードの場合は無効にして、レーンの順序を変更できる。演算シフトに符号付き整数を使用する場合は、論理モードに適用する必要がある。レーンの並べ替えは、符号なし入力でのみ有効である。これは、符号拡張では、符号拡張を制約するためにレーンマスクが必要だが、レーンの並べ替えによってレーンデータがレーンマスクの外側に移動されるためである。

【0189】

【表13】

表13: 演算レーンブロックのレーンマスク

モード	レーン0	レーン1	レーン2	レーン3	レーン4
SIMD 2	63:0	63:0	127:64	127:64	なし
SIMD 4	31:0	63:32	95:64	127:96	なし
その他すべて	127:0	127:0	127:0	127:0	127:0

【0190】

入力レーンの左端のビットは、浮動小数点と符号付き整数の両方の符号ビットとして解釈され、整数のデータビットとして解釈される（このビットは浮動小数点形式のシフトにマスクされるが、否定と2の補数変換を制御するために指数論理ブロックによって使用される）。

【0191】

各レーンのシフトネットワークからの128ビット出力は、（464からの）単一の128ビット合成出力を形成するために、他のすべてのレーンシフトの128ビット出力と論理ORされる。すべてのレーンに対してシフトを0に設定すると、各レーンのデータは出力のビット0に最下位ビットで配置される。各レーン上のシフト値は、指数論理ブロックにおいてバイアスされ、各レーンに異なるシフトを与え、算術演算のためのレーンの重複を防止する。OR論理はまた、関連する出力レーンのすべてのビットを反転する（すなわちレーンデータに対して1の補数を実行する）、出力の各32ビットレーンに対する選択可能な反転を含む。反転はモードと符号ビットの関数として指数ブロック内からも制御される。演算で使用する場合、反転は2の補数演算の一部を形成し、符号拡張と算術出力での1の加算によって完了する。レーン結合および反転論理の出力は、論理出力のための演算出力および回転画像論理に送られる。各レーンの反転と2の補数論理は入力と同じレ

ーンによって制御されるため、算術出力ではレーンの並べ替えは許可されない。図 3 3 は、モードによるビット整列を示す。

【0192】

Z 入力回転／シフト論理回路 3 3 0 の演算出力は、乗算後乗算器シフタコンバイナネットワーク 3 1 0 に接続され、そこで、同じ R A E 3 0 0 内で計算された積または積の和に加算される。演算出力は、桁上げ保存形式で表される各レーンの 2 の補数符号付きデータであり、各レーンの合計が乗数からの積の合計に加算される 2 の補数値である 1 2 8 ビットベクトルの対として表される。レーンの数はアキュムレータ S I M D モードに依存し、これは乗算器 S I M D モードと同じであっても異なってもよい。レーンは、S I M D - 4 の場合は 3 2 ビット、S I M D - 2 の場合は 6 4 ビット、単一操作の場合は 1 2 8 ビットである。和ベクトルは、各レーンの反転付きシフト Z 入力（反転が選択されている場合）である。桁上げベクトルは、各レーンで 2 の補数を完了するための + 1 を含む。桁上げベクトルの他のすべてのビットは常に 0 であるため、これらのビットは物理的に実装されていないため、ポスト乗算シフト／コンバイナネットワークへの接続はシングルエンドになり、各レーンで 1 ビットずつ増加する。インクリメントは桁上げベクトルで行われ、Z 入力論理の算術出力で桁上げを伝播する必要がなくなる。

【0193】

回転画像 4 6 6 O R は、シフトされたデータのコピーであり、データが二つの連結されたコピーであるように、入力 S I M D レーン幅だけシフトされたデータに左シフトされる。論理出力はウィンドウ化され（4 6 8）、連結されたデータがウィンドウを通してシフトされると、ウィンドウ内のデータが回転したように見える。図 3 3 のビット位置は、0 シフトについて示されている。シフトは、左シフトが追加されると、ウィンドウの上部からシフトされたビットがウィンドウの下部に再び入るように見えるように、レーン入力をシフトされたビット数だけ上に移動させる。回転ではなくシフトが選択されると、オフセットシフトされた画像の代わりに赤色の凡例データが 0 に設定されるので、0 シフトはウィンドウ内にデータを生じず、シフト距離が増加するとウィンドウの下端に上位ビットが表示され、最大シフトでは L S B のみがウィンドウの上部に残る。回転モードでは、符号拡張子との O R が符号拡張子によってマスクされるため、回転されたイメージに符号なし入力が必要である。

【0194】

論理出力は、モードに応じてレーンウィンドウ内のビットのみを選択することによって、S I M D レーンを元のサイズのレーンに再マージする。4 × 8 S I M D、2 × 1 6 S I M D、3 2 ビットまたは 6 4 ビット出力用のウィンドウを選択する 4 : 1 m u x を介して選択する。上位 3 2 ビットは、6 4 ビットモードでない場合、0 に無効化される。

【0195】

図 3 4 は、I E E E 単精度から基数 3 2 への浮動小数点フォーマット変換を示すブロック図である。浮動小数点モードでは、指数値に従って仮数データ経路を操作するだけでなく、浮動小数点指数を処理する必要がある。2 つの浮動小数点値の積の指数は、被乗数の指数の合計である。浮動小数点加算では、加算されるためには各加数の指数が同じでなければならない。加数指数が等しくない場合、指数の小さい方の加数は、指数の差で表されるビット数だけ右にシフトされる。右シフト後、両方の加数はより大きな指数を共有し、合計することができる。合計する加数が 2 つを超える場合は、加数の中の最大指数を決定し、指数の差に等しいビット数だけ、より小さい指数を持つ加数を右にシフトする必要がある。

【0196】

アキュムレータ 3 1 5 ループ内のシフト量を最小にするために、R A E 3 0 0 は、浮動小数点数が基数 3 2 指数に変換された左に 3 1 ビットまで事前にシフトされる、従来とは異なる内部形式を使用する。左シフトに対抗するための指数の調整では、指数の最下位 5 ビットがゼロになり、その後破棄され、すべての下流アラインメントシフトは 3 2 ビットの倍数になる。この修正は、より広いアキュムレータを犠牲にして、指数比較の複雑さと

10

20

30

40

50

臨界アキュムレータループ内のシフト論理を著しく低減する。さらに、アキュムレータ 315 の前に和が発生することは、アキュムレータ 315 の前に追加のシフト論理を必要とし、2 つ以上の加算器があるので、各パスはそれ自身のシフトを必要とする。基数 32 指数への変換は、図 34 に示されており、指数論理回路 335 によって実行される。

【0197】

7. 指数論理回路 335

図 35 は、指数論理回路 335 を示すブロック図であり、これは、Z 入力指数論理 465、過剰シフト論理 470、および乗算器指数論理 475 を含み、これらの各々について、以下により詳細に説明する。代表的な実施形態では、アキュムレータ 315 は、それ自体の指数論理を含む。

【0198】

この RAE 回路 300 はまた、4 つもの RAE 300 からの複数の積および関連する付加的入力を合計する内積モードをサポートする。これらの積を合計するには、すべての加数から最大指数を求め、各加数の仮数に対する右シフト距離を決定するために、各指数とその最大値との差を計算する必要がある。IEEE の半精度と 1/4 精度の値は基数 32 の指数変換により整数に変換されるため、各レーンで結果を適切に揃えるための変換シフトとシフトバイアス以外の指数処理は必要ない。アキュムレータ 315 は、冗長符号ビットを推定し、オーバーフローなしで可能な場合は 32 ビットの倍数だけ左シフトし、アキュムレータの指数はそれに応じてデクリメントされる（アキュムレータ指数は、5 lsb を除去した後の最左ビットである）。累算後最終加算器、丸めおよび正規化段階（340）は、累積された合計を再正規化し、正規化シフト距離をアキュムレータの指数値に加算して、正規化の一部として IEEE または Bfloat の全指数を再構成する。

【0199】

指数論理回路 335 は、アキュムレータ 315 の前での指数処理である。指数論理回路 335 の機能は、乗算器指数を合計して指数バイアスを補償すること、左シフトによるすべての浮動小数点入力を基数 32 の指数へ変換すること、RAE 回路クアッド 450 内の他の RAE 300 からの指数を含む、全ての加算器の中の最大超過指数（超過は基数 32 指数）を見つけること、各加数のシフト距離を、最大値からの差の 32 倍として計算すること、正しい出力アラインメントのためにモード依存シフトバイアスを追加すること、Z 入力シフト 330 および乗算器シフトコンバイナネットワーク 310 のシフト設定を生成すること、および、指数 0 を検出すると、仮数と指数を強制的に 0 に戻す（非正規性を 0 に変換する）こと、として要約することができる。

【0200】

超過シフト論理 470 は、RAE 回路クアッド 450 内の他の RAE 300 へのリンクを含む最大指数を求める論理である。乗算器指数論理 475 は、乗算器指数の加算と、シフトコンバイナのシフトに対するシフトネットワーク制御の導出とを含む。Z 入力指数論理 465 は、Z 入力シフト 330 のシフト制御を計算する。

【0201】

非正規化入力は 0 に置き換えられる（指数が 0 の場合、データ経路は強制的に 0 になる）。まれに非正規化数が必要とされる場合には、RAE 300 内の比較ブロック 320 および他の論理を使用して、非正規化を検出し、IEEE 標準に従って、整数乗算および加算を使用してプロセス非正規化への代替処理パスを導くことができる。

【0202】

32 ビットの X、Y 及び Z 入力 365、370、375 は、浮動小数点指数及び符号にアクセスするために指数論理回路 335 に入力される。これらには、SIMD モードの指数の複数のセットが含まれる。

【0203】

各 RAE 300 は、そのローカル最大 3 ビット基数 32 指数を 7 線バーコード信号（7 wire bar code signal）として出力する。3 ビット指数を変換して、3 ビットコードで示される連続配線数をオンにする。これら 7 本の配線は、RAE 回路ク

10

20

30

40

50

アッド 4 5 0（個別には図示せず）内の他の 3 つの R A E 3 0 0 のそれぞれに接続される。これらは、各 R A E 3 0 0 の最大指数を共有し、またはダブルモードの場合には、分解された過剰シフトを R A E 回路クアッド 4 5 0 内のすべての R A E 3 0 0 に送信する。

【 0 2 0 4 】

乗算器シフタコンバイナネットワーク 3 1 0 の各レーンに対する 6 レイヤシフタおよびゼロゲートに対するシフト制御は、指数論理回路 3 3 5 によって生成される。第 2 レベル乗算器シフタのための 4 つのシフト制御も指数論理回路 3 3 5 によって生成される。7 層シフタのシフト制御および Z 入力シフタ 3 3 0 の各レーンのゼロゲートは、指数論理回路 3 3 5 によって生成される。第 2 レベル乗算器シフタのための 4 つのシフト制御も指数論理回路 3 3 5 によって生成される。

10

【 0 2 0 5 】

指数論理回路 3 3 5 の構成は、一般に、Z 整数シフト／回転距離の制御と、Z シフトソース制御（X 入力、設定、指数）と、数値モード（S I M D レーン、F l o a t 形式）制御と、指数マスク（数値モードと S I M D で設定）制御と、レーンおよびネイバー R A E 3 0 0 入力の制御をイネーブルにする制御を含む。

【 0 2 0 6 】

超過シフト論理 4 7 0 は、各加算器に必要な 3 2 ビット右シフトの数を決定する。Z 入力と積は融合多重加算演算の一般的な場合に対して合計される。この加算は、アキュムレータ 3 1 5 の前、および隣接する R A E 3 0 0 積ツリーの出力が結合される前に行われる。明確にするために、過剰な論理は、Z 入力指数論理 4 6 5 および積加算器の両方について議論される。

20

【 0 2 0 7 】

余分なシフトは、合計を完成させるために、加数を揃えるために必要な右の 3 2 ビットシフトを指す。5 ビット以上の指数を持つモードでは、各加数をどれだけ右シフトする必要があるかを決定するために、過剰シフト論理が必要である。超過シフト論理 4 7 0 は、まず、全ての加数の中から最大指数を決定しなければならない。次に、各加数について、その指数を最大値から減算して、適用する必要がある右シフトの量を決定する。

【 0 2 0 8 】

8 ビットの指数を持つ最大 8 つの加数を組み合わせる（単精度内積それぞれが浮動小数点 Z の加数を持つ、4 つの融合された乗算加算の積）必要がある。Z 入力は、隣接する R A E 3 0 0 積の後に合計に加算されるが、合計されたすべての積は同じ重み付けにシフトされるべきであり、各積の Z 入力は同様に重み付けされる必要がある。したがって、指数論理回路 3 3 5 は、隣接する R A E Z 入力が局所和に寄与しない場合でも、すべてのアクティブ Z 入力を調べる。これは、最大指数を決定するために、R A E 回路クアッド 4 5 0 内の 4 つの R A E 3 0 0 間の通信を必要とする。

30

【 0 2 0 9 】

B f l o a t 1 6 モードには、F P M A C ごとに独立した 8 ビットの指数を持つ 2 つの浮動小数点値がある。B F L O A T ドット積の場合、各 F P M A C からの 2 つの積と 1 つの Z 入力加数が合計され、指数を持つ最大 1 2 の加数が必要になる。ドット積のない B F L O A T は、R A E 間指数接続が 2 つの指数をサポートしないため、隣接 R A E の融合和を許容しない。最後に、倍精度浮動小数点では、4 つの R A E コアを結合した単一の融合多重加算が可能である。過剰な指数シフトを用いるモードを表 1 4 にまとめる。

40

【 0 2 1 0 】

50

【表 1 4】

表14:

モード	指数 msbs	その他の RAE 指数	Z レーン	
Double	6	Yes*	1	Z拡張の最大値、HH mult、 シフトをすべてのRAEに分散HH
Single	3	no	1	Multの最大値、Zローカルのみ
Bfloat16 SIMD	3	no	2	2レーンローカルのみ最大Z ／レーン当たりのmult
Half SIMD	0	no	2	32バウンドの整数
QTR SIMD	0	no	4	16バウンドの整数
Single Dot	3	yes	1	RAEあたり最大4つのRAE、 Zに加えて1つの積
Bfloat dot	3	yes	1 half	RAEあたり最大4つのRAE、 Z+2つのレーンbfloat bfloatから拡張された1レーンZ全幅
Half dot	0	No	1 half	単一のレーンZから拡張された整数
Qtr dot	0	No	1 qtr	単一のレーン1／4Zから展開された整数

10

20

【0 2 1 1】

図 3 6 は、Z シフタ 3 3 0 および乗算器シフタコンバイナネットワーク 3 1 0 のための超過シフト論理 4 7 0 を示すブロック図である。図 3 7 は、2つの入力 x ファン出力 2 (2 × 2) ゲートを有する 3 ビット・インデックス対 8 ビットバー回路 4 8 0 を示すブロック図であり、図示された「1」はバッファである。図 3 8 は、フル加算器を用いて複数のバーの X O R 差をインデックスに変換するタリー回路 4 8 5 を示すブロック図である。図 3 9 は、2つの入力 x ファン出力 2 (2 × 2) ゲートを有する 8 ビットバー対 3 ビットインデックス回路 4 9 0 を示すブロック図である。

30

【0 2 1 2】

超過シフト論理 4 7 0 は、その中心部分として 1 2 加数プロセスを含み、論理は、倍精度のより広い指数および固有分布要件によって要求される特殊な処理のために追加される。SIMD モードの 2 番目の B F L O A T 1 6 レーンには、大きな加数プロセスの余分なコピーが使用される。超過シフト論理 4 7 0 は、各 3 ビット指数を 8 ビットバー表現（バーの表現はワンホットデコードに似ているが、ワンホットビットに加えて、デコードされたビットより下位のすべてのビットもオンになる）に変換することによって、1 2 個までの 3 ビット超過指数を比較するタスクを単純化する。これは 1 ホットよりも複雑ではないデコードであり、この設計には利点がある。最大指数を求めるために、各バーの上位 7 ビットはビットごとに O R される（バーの最下位ビットは常に「1」であるため、破棄される）。最も高いバーは短いバーより優先され、最大指数に対応する。O R ツリーは、R A E 3 0 0 内から 1 つまたは 2 つの積指数と Z 入力指数とを予め結合するために、8 ビット O R によってローカル 3 入力に分割される。O R への各入力を選択された入力を遮断するためのゲートを有するように、O R は A N D - O R - I N V E R T ゲート 4 7 1 から構成される。7 ラインのローカル最大値は、R A E 3 0 0 間の専用の 7 ビット接続を介して、R A E 回路クアッド 4 5 0 内の他の 3 つの R A E のそれぞれに送信される。

40

【0 2 1 3】

各 R A E の受信機は、他の R A E 3 0 0 からの 3 つの 7 ビット入力と、それ自体からの内部 7 ビット入力とを有する。これらは、各 R A E 3 0 0 内で O R 結合され、各 R A E は、最大バーの複製を保持する。R A E 結合構造も A N D - O R - I N V E R T ゲートであ

50

るため、他の R A E からの入力をこの R A E への入力でブロックすることができる（これにより、R A E を独立した和または独立した和と対で使用することができる）。最大指数バーは、各ローカル B A R ソースと個別にビットごとの排他的論理和がとられる。排他的 O R 出力は、最大値と異なる複数のバービット上にのみ「1」ビットを有する。「1」ビットのカウントは、その加数に対する指数の差を示す。この差は、タリー回路 4 8 5 を用いて 1 ビット（これらは連続しているが、7 ビットフィールドの任意の場所に配置できる）をカウントアップするために 7 ビットのタリー加算を用いてバイナリインデックスとして回復される。

【0 2 1 4】

最大の実際のシフトは、3 3 2 ビットシフトまたは 9 6 ビットである。それを超えると、シフトは 1 2 8 ビット以上であるため、加数はゼロになる。タリー加算器の 2 つの最下位ビットの値は、0、1、2、または 3 個の 3 2 ビットシフトのシフトに対応する。残りのタリー加算器ビットは、「1」が加数をゼロにする。この差は最大値からローカル指数を引いた値であるため、常に負ではないシフトになる。

【0 2 1 5】

差 $\text{max} - Z$ は Z 入力の超過シフトであり、同様に差 $\text{max} - \text{Product}$ は積の超過シフトである。これらの余分なシフトは、追加された右シフトを制御するために、符号化ブロックを介して関連するシフトネットワークに適用される。最大バーは、最大指数を表すインデックスにもデコードされる。これは、アキュムレータ入力の指数としてアキュムレータによって使用される。タリー加算器 4 8 5 を使用してアキュムレータ指数を復号化することもできるが、出力は常にバーであるため、復号化は単純であり、タリー加算器の全加算器を必要としない。2 番目の b f l o a t レーンには、レーン 1 Z の入力と積の最大値を見つけるために、同じ余分な論理のローカルのためのバージョンがある。この 2 番目の最大値の R A E の外側にはリンクがないため、2 つの加数の超過指数のローカル O R にすぎない。最大値はアキュムレータの 2 進指数に変換され、余分なシフト距離はプライマリと同じように計算される。

【0 2 1 6】

倍精度の論理は異なる。これは、指数の超過が 3 ビットではなく 6 ビットであり、加数が 2 つしかないためである（Z 入力と積、ダブルス用のドットモードはない）。倍精度は、乗算器が 4 つの R A E に分配され、その結果、積超過シフトを R A E 回路クアッド 4 5 0 内の他の R A E 3 0 0 に分配する必要があるため、ユニークである。倍精度超過論理は、最上位の X ビットと Y ビットの積を含む同じ R A E 3 0 0 内に存在し、両方の指数が求められる。さらに、Z 入力論理は入力拡張機能を使用しますが、Z 入力論理が上位の入力と同じ R A E に存在するためには、最下位の半分である必要がある。X と Y の入力は両方とも被乗数を使用するため、Z 拡張入力は隣接する R A E の Z 入力から取得される。Z シフトの最下位ビットにある拡張のバイアスは、この特別な場合（二重反転 H L）に対して Z 入力で修正される。

【0 2 1 7】

二重超過論理は 1 2 ビット最終加算を供給する桁上げ保存加算器を使用して $X + Y - Z$ を実行し、積指数と加数指数の差を求める。これは、高速 $(X + Y) - Z$ を得るために、2 層の先読み加算器の遅延と追加領域を持つのではなく、個別の加算器で行われる。1 2 ビットの差には、どちらが大きいかを識別するための符号ビットが付加される。1 2 ビット合計は、1 2 ビットバイナリを、超過積シフトおよび超過 Z 入力シフトに対する飽和 5 ビット B A R 値の対に直接再エンコードするデコーダに供給される。デコーダの真理値表を表 1 5 に示す。各バー値は、8 ビットバーの左側に「0」パディングされ、最下位ビットは、1 次超過回路で最大に使用されるものと同様の 7 ビットバーに到達するように計算されない。積バーは、他の R A E 3 0 0 への分配へのパスを有するように、ローカル A N D - O R - i n v e r t 最大値に配線される。

【0 2 1 8】

ダブルモードでは、A N D - O R - I N V E R T への他の入力が無効になり、積の超過

10

20

30

40

50

シフトはそのまま出力される。各 R A E の受信側では、二重超過を計算する R A E からの入力だけがイネーブルになる。ダブルモードは、レーン 3 超過シフト論理への他方の入力をオフにして、積超過シフトが正しいシフトにデコードされるようにする。演算二重超過差論理を含む R A E では、Z 超過バーは同じ R A E 内に配線され、マルチプレクサを介して Z 超過シフトタリー加算器に直接接続され、Z 超過シフトタリー加算器は Z シフト距離に直接変換される。

【0219】

差分加算器出力の M S B は、アキュムレータ指数として使用される多重指数の和 $X_P L U S_Y$ または Z 指数の 6 つの最上位ビットのいずれかを選択するために使用される。二重モードによって制御される M U X は、その最大二重指数と復号化された最大指数との間で、上述した 1 次超過論理から選択する（指数は、アキュムレータにおいて、倍精度と単精度の両方に対して同じ指数論理を使用するために、左に 3 ビット拡張された 0 である）。

【0220】

【表 15】

表 15: 超過シフトダブルシフトデコーダーの真理値表

ビット11:8	ビット7:5	X バー	Z バー	アクション
0 1 x x	x x x	0	F	Z+3<X Zを0に強制する
0 x 1 x	x x x	0	F	
0 x x 1	x x x	0	F	
0 x x x	1 x x	0	F	
0 0 0 0	0 1 1	0	7	Z+3=X Zを右96ビットシフトする
0 0 0 0	0 1 0	0	3	Z+2=X Zを右64ビットシフトする
0 0 0 0	0 0 1	0	1	Z+1=X Zを右32ビットシフトする
0 0 0 0	0 0 0	0	0	Z=Xシフトなし
1 1 1 1	1 1 1	1	0	Z=X+1 Xを右32ビットシフトする
1 1 1 1	1 1 0	3	0	Z=X+2 Xを右64ビットシフトする
1 1 1 1	1 0 1	7	0	Z=X+3 Xを右96ビットシフトする
1 1 1 1	1 0 0	F	0	Z>X+3 Xを0に強制する
1 x x x	0 x x	F	0	
1 x x 0	x x x	F	0	
1 x 0 x	x x x	F	0	
1 0 x x	x x x	F	0	

【0221】

図 40 は、乗算器（およびシフトコンバイナ）指数論理 475 を示すブロック図である。S I M D レーンごとに個別の積指数を維持する必要がある。積指数は、そのレーンの乗数指数の合計である。さらに、各乗数指数が 0 に等しいかどうかを確認する必要がある。

また、レーン内のいずれかの1つが0の場合は、非正規性がゼロとして扱われるため、そのレーンの積がゼロになる。各レーンの指数加算器476は、指数オーバーフローを検出するために、1ビットだけ拡張されてもよい。実行される場合、指数オーバーフローは、追加のリソースを使用して無限大を生成する際に使用するために、RAEのサフィックス（および／または条件付きフラグ）出力405に送ることができる。

【0222】

各レーンの最下位積指数ビットは、除去されたモード固有バイアスに加えられ、指数プレシフトを制御するためにシフトコンバイナ論理に提供される。複数の積および／またはZ入力加数が合計されるモード（ドットおよび複素乗算）の場合、最大積指数を選択し、各積をその指数と最大指数との差だけ右に移動して仮数を揃える必要がある。

10

【0223】

最大指数の計算については、前のセクションの過剰シフトの計算で説明した。仮数を揃えるためのシフトは、32ビットの倍数の右シフトである。128ビット以上のシフトは加算器幅をアンダーフローするため、余分なシフトが3×32ビットを超えると、加算器または積は0に置き換えられる。最大基数32の指数は、積と加数の合計の指数としてアキュムレータに渡される。図40は、乗算器指数論理475を示し、これについては、以下でさらに詳細に説明する。

【0224】

すべてのSIMDモードに対応するために、各RAE300に対して4つの別個の指数が維持される。4つのレーンモードでは4つすべての指数が使用され、2つのレーンモードではそのうちの2つ（レーン1と3）が使用され、残りのモードでは1つ（レーン3）の指数のみが使用される。倍精度浮動小数点形式には11ビットの指数があり、lane-3指数論理のみが使用される。単精度浮動小数点には8ビットの指数があり、8ビットをアクティブにしたまま、0にマスクすることで3つのLSBを無効にしたレーン3の指数論理のみを使用する。Bfloat16は、8ビット指数を持つ2レーンSIMDである。上のレーンでは3lsbsがマスクされ、下のレーンでは8ビットのレーン1指数が使用される。

20

【0225】

半精度も2つのSIMDレーンであり、レーン3とレーン1を使用するが、半精度の指数は5ビットであるため、これらのレーン指数の最上位の5ビットを除くすべてのビットは半精度のためにマスクされ、余分なシフトは発生しない。1/4精度は4ビット指数を持つ4レーンSIMDである。レーン0とレーン2の指数は、1/4精度でのみ使用されるため、これらのレーンの指数論理は4ビットに固定されている。レーン1とレーン3の指数は、1/4の精度のために、これらの各レーンで4つのMSBのみを使用するようにマスクされる。半精度IEEEフォーマットと同様に、指数は6ビット未満であるため、余分なシフトはない。

30

【0226】

また、現在のモード指数幅に一致するようにマスクされた、各被乗数指数に対するゼロ検出論理も存在する。レーンの被乗数指数のいずれかが0であるか、または超過シフトが3×32より大きい場合、強制レーン0信号が生成される。マルチプレクサのセットは、SIMDモードによって各レーンに適用する適切な力0信号を選択する。

40

【0227】

マルチプレクサの第2のセットは、各レーンにおける基数32シフトを制御するために、適切な積指数から適切な5つのLSBを選択する。乗数積はアキュムレータの前に繰り込まれないので、積の指数を調整する必要もない。仮数データ経路幅は、基数点の左側の余分なビットと、積を合計するときの成長を考慮する。

【0228】

レーンシフトは、モード依存シフトバイアスでレーン指数5lsbsを合計することによって計算される。その合計の下位4ビットは、最初の0:15シフトを直接制御する。バイアスされた指数の上位ビットは、レーン1および3に対する過剰シフト論理からの過

50

剰シフト距離に加算され、次いで、その和が、コンバイナ段階におけるマルチプレクサに対するシフト制御に記録される。

【0229】

ポスト乗算シフトコンバイナのためのシフト距離は、図25において表にされる（ここで、E4とE5は、それぞれ製品指数の4と5のLSBを表し、負のシフトは右シフト、正のシフトは左シフトであり、「0」は、レーンが強制的にすべて「0」になることを示し、「0」は0シフトを示す）。シフト距離はモード、指数LSBに影響され、いくつかのモードではシフトに関連する指数と合計値の最大指数との差に影響されることに注意すべきである。乗算後シフターコンバイナのシフト距離

【0230】

図41は、Z入力指数論理465を示すブロック図である。Zシフタ330は、積とZ成分とを合計するために、Z入力を各レーン内の積のシフトされた乗算和と位置合わせすべきである。乗算器305の積は、最下位の5指数ビットを除去するために、32の倍数である指数に予めシフトされる。同様に、Z入力指数論理465は、シフトを行うために5つの指数LSBを使用して5つの最下位指数ビットを除去するためにZ入力データをプリシフトする。各レーンのシフトにモード依存シフトバイアスを加えて、適切なシフトを行い、仮数を適切にレーンに配置する。指数論理には、構成またはX入力から指定された整数シフトの入力もある。

【0231】

5指数ビットを超えるモード（倍精度および単精度IEEEおよびBFLOAT16）では、Z入力を乗算器に揃えるために、プリシフトを超える追加のシフトが必要である。追加のシフトは、指数超過シフトのセクションで既に説明した超過シフト論理によって決定される。超過シフトに-32を乗算した値がバイアスされたシフトに加算され、各レーンの0:127ビットシフト距離になる。モード別に集計されたレーン別シフトバイアスは、図29Aに詳細に示されている。シフト合計は8ビットで、0:127ビットシフト範囲を超えるシフトを検出するために余分なmsbが使用される。このシフトオーバーフローは、127ビットを超えてシフトされたときに、レーンを強制的に0にするために使用される。

【0232】

指数論理はまた、0指数を検出するために、各入力レーンの指数に対する0との比較回路を有する。レーン1および3に対して、指数LSBは、モードに応じてマスクされ、11、8、5または5ビットの指数を選択する。レーン0および2は、指数なしまたは4ビット指数のみである。各レーンについての指数が0に等しい検出は、モードセレクトによって選択されて、0レーン論理を生成し、その出力は、レーン（図示せず）についてのシフトオーバーフローと組み合わせられて、各レーンについての強制レーン0信号を生成する。

【0233】

8. アキュムレータ315

図42は、アキュムレータ315を示すブロック図である。アキュムレータ315は、乗算器シフターコンバイナネットワーク310によって提供される積の合計を蓄積するように設計された融合128ビットアキュムレータである。アキュムレータ315は、様々な浮動小数点および固定小数点フォーマットを提供するマルチモードであり、1レーン、2レーンおよび4レーンSIMD演算をサポートする。

【0234】

アキュムレータは、一方の入力がその前の出力によって供給され、他方の入力が乗算器シフトコンバイナネットワーク310論理（Z入力と、このRAE300乗算器305および付加されたRAE300乗算器積からの積の和である）によって、連続する入力を合計するように配置された登録加算器である。アキュムレータ315の入力、出力および内部データ経路は、桁上げ保存形式である。アキュムレータ315は、1つの128ビット、2つの64ビットもしくは4つの32ビット整数算術レーン、または1つの128ビット（IEEEダブルもしくはシングル）もしくは2つの64ビット（Bfloat16）

10

20

30

40

50

浮動小数点レーンをサポートする。アキュムレータ315は、アキュムレータ指数演算器513と、基数32指数（32ビットの倍数によるシフト）をサポートするシフタ511を含む。シフタ511は、加算のために基数点を整列させるために、乗算器入力（Z入力）またはアキュムレータフィードバックの小さい方に、32ビットの左シフトおよび32ビットの倍数による右シフトを再正規化する役割を果たす。

【0235】

要約すると、アキュムレータ315は、以下のように動作する。

1. 選択した形式で連続する入力を合計する。
2. 128ビットレーン×1、64ビットレーン×2、32ビットレーン×4の整数累積をサポートする。
3. IEEEの倍精度および単精度浮動小数点累積（1レーン）をサポートする。
4. 2レーンのBFLT16浮動小数点累積をサポートする。
5. 加算に備えて基数点を整列させるために、浮動小数点の前の累積和またはZ入力を32ビットの倍数だけ右シフト（指数の小さい方をシフト）する。
6. 2レーンモードまたは4レーンモードで走行しているときに、必要に応じて桁上げブロッカーおよびシフトブロッカーを使用してレーンを分離する。
7. 各浮動小数点レーンのアキュムレータ指数を計算して維持する。
指数は基数32の指数で、6ビットの場合のdoubleを除き、3ビットである。
8. 基数32の値を再正規化するために、32ビットの倍数でアキュムレータ出力を左シフトする（そしてそれに応じてアキュムレータの指数を調整する）。
9. 1GHzタイミングのアキュムレータループ周りの単一レジスタ遅延。
10. 浮動小数点の右シフトでは、偶数への丸めをサポートするために、各レーンに補助丸めビットを設定する必要がある。
11. 左側の余分なビットは、加算器のオーバーフローを維持し、浮動小数点のオーバーフローを修正するために右シフトと指数インクリメントを行うように検知されるべきである。
12. 整数オーバーフローを検出して符号でラッチし、整数飽和論理の最終加算器論理に渡す必要がある。
13. 右シフトがレーン幅を超える場合、Z入力またはアキュムレータフィードバックのいずれかをゼロにする。
14. Z入力の第1の有効な入力と同時にフィードバック入力を0に強制することによってアキュムレータを初期化する。
15. すべての入力に対して初期化条件を保持することで、融合された乗算－加算（アキュムレータなし）を可能にする。
16. 先頭符号予測器出力を最終加算器に供給する。
先行符号予測器は、nまたはn－1の符号ビットの繰り返しを示す（内部シフトを32ビットの倍数で制御し、最終加算器でより細かいシフトを制御するために使用される）。
17. アキュムレータZの入出力を登録する。
18. 整数をfloat、1、2、または4レーンに変換する（最終的な加算論理にアタッチされている場合）。

【0236】

図42を参照すると、固定小数点演算の場合、アキュムレータのシフトはすべて強制的に0シフトにされ、指数レジスタは無視されるか、または使用されない。初期化フラグが「1」の場合、アキュムレータフィードバックのシフト右論理はフィードバックアキュムレータ値を「0」に強制し、アキュムレータがZin入力に存在する値を2の補数桁上げ保存形式でコピーできるようにする。初期化が「0」の場合、アキュムレータ出力レジスタの現在値は4:2圧縮器509の右入力に供給され、シフトされずに、シフトされていないZin入力データに（桁上げ保存形式で）加算され、桁上げ保存合計がアキュムレータレジスタに戻される。追加によって余分なMSBへのオーバーフローが発生した場合、問題のレーンのオーバーフローフラグが設定され、アキュムレータが再度初期化されるま

で残る。固定ポイント動作では、デフォルトの128ビットシングルレーンモードに加えて、2レーン（各64ビット）と4レーン（各32ビット）のSIMDモードが許可される。複数レーンの場合、4:2圧縮器509は、各レーン間に搬送阻止ゲートを有し、各レーンには、オーバーフローを検出するための補助MSBが取り付けられている。入力シフタ511Aは、常に0シフトに設定されて固定点累積を行う。出力シフタ511Bは、タイミングおよび機能要件が、論理ゲート数を低減するために満たされ得る場合、出力のための4つの32ビットフィールドのうちの1つを選択するために、設計者の裁量で使用され得る。

【0237】

アキュムレータは、基数32指数を使用して、クリティカルアキュムレータループ内のシフト論理を簡略化する。入力データは、下位5指数ビットがゼロになるように、31ビットまで左に予めシフトされる。これらのゼロ化された指数ビットはドロップされ、指数ビットのみが左側に残る。アキュムレータ内のすべてのアライメントおよび正規化シフトは、32ビットの倍数で行われる。この実装は、アキュムレータ内のシフタの層を減少させ、またアキュムレータの指数論理をかなり減少させ、タイミングを閉じるのを助ける。IEEEの1/2および1/4精度形式（それぞれ5および4の指数ビットを持つ）は、シフトコンバイナおよびZシフト論理で行われる基数32指数変換によって、実質的に整数に変換される。これら2つの形式の場合、アキュムレータは適切な整数モードで動作する。アキュムレータの設計では、SIMD Floatは省略される。このモードが、かなりの追加ハードウェアを必要とする特殊なケースなためである。アキュムレータ浮動小数点モードは常に単精度レーンで、単精度が選択されると最終加算器で単精度に丸められる。整数モードの場合、アキュムレータは、1つの128ビットレーン、2つのSIMD64ビットレーン、または4つのSIMD32ビットレーンとして動作することができる。浮動小数点加算では、両方の加数の基数点が同じである必要がある。これは、両方の指数が一致するまで、加数の一方を他方に対して相対的にシフトする必要があることを意味する。この設計では、右シフト用の指数部が小さい方の加算器（図では4:2圧縮器）を、アラインメントに必要な32ビットシフト数だけ選択する。各32ビット右シフトは、その入力に関連付けられた指数に1を追加することに対応する。指数論理は、必要なシフトの方向と距離を計算し、シフト論理が32ビットの正しい倍数だけ小さい方の入力を右シフトさせる。128ビット以上のシフトでは、小さい方の入力は加算器の128ビット幅からシフトされるため、大きい方のシフトは入力をシフトする代わりに0にシフトする。シフトすると、128ビットアキュムレータのlsb端に追加されたIEEEラウンド、ガード、およびスティッキービットも設定され、最も近い偶数モードへのIEEEラウンドがサポートされる。浮動小数点2レーンSIMD（BFLOAT16でのみ使用）では、丸め用の3ビットが両方のレーンのLSBに追加される。2つの加数の符号が互いに反対の場合、アキュムレータの結果は、入力のどちらよりも先頭の符号ビットが多い可能性がある。先頭符号ビットの数が、符号を失うことなく左シフトを可能にするのに十分大きい場合、出力シフタは、32ビットの倍数だけデータを左シフトして、余分な先頭符号ビットを除去し、それによって基数32指数システムでくりこみを行う。アキュムレータ指数は、左シフトの指数を調整するために、32ビットシフトの数だけデクリメントされる。指数論理は8ビット幅、すなわち、IEEEの二重指数に対応する6ビット（11～5ビット）、および指数のオーバーフローとアンダーフローを検出するための追加の2ビットである。

【0238】

アキュムレータ315は、オーバーフローを吸収するのに十分な3つの追加ビットを左側に有する（追加ビットは、多重シフトコンバイナ論理チェーンの後段階にも存在する）。これらのビットへのオーバーフローが発生すると、アキュムレータ出力シフトは32ビットの右シフトとそれに伴うアキュムレータ指数の増分を実行してオーバーフローを修正する。アキュムレータ315はSIMD浮動小数点をサポートしておらず、半および1/4精度IEEEは基数32指数変換によって整数に変換される。発明者らは、アキュム

10

20

30

40

50

レータの複雑さを実質的に低減するために、アキュムレータによる `B f l o a t 1 6 S I M D` をサポートしないことを選択した。浮動小数点 `S I M D - 2 ( B F L O A T 1 6` のみ) の場合、余分な `M S B` は両方のレーンに追加される。浮動小数点 `S I M D - 2` モードでは、レーン 0 がレーン 1 の合計に影響を与えないように、4 : 2 コンプレッサのビット 64 のレーンブロックがアクティブになり、すべてのシフタは、下位レーンから上位レーンにシフトされたデータを 0 で置き換え、上位レーンから下位レーンにデータを拡張符号で置き換えるための追加ゲートを必要とする。

【0239】

アキュムレータ 315 は、乗算器シフトコンバイナネットワーク 310 から有効なデータと共に、仮数、1 次および 2 次の指数 (`S I M D - 2` をサポートするため) を符号化している。また、すべての `S I M D` レーンに共通の設定、アキュムレータフラグの初期化、リセットおよびクロック入力も備えている。ブロック出力は、累積データ、1 次および 2 次指数、推定先行符号ビット数、および累積データ有効フラグを含む。

10

【0240】

`Z i n` 符号付き仮数入力 502 部分は、桁上げ保存形式 (実際には、各 `S I M D - 2` レーンの `l s b` で 3 ビットを拡張し、各 `S I M D 2` レーンの `m s b` で `T B D` ビット、各 `S I M D 4` レーンの `m s b` で `T B D` ビットを拡張した。) 128 で提示される。入力は、アキュムレータ論理へのエントリで登録され、そのレジスタは、データ有効入力信号によってクロック可能にされる。`Z i n` 仮数は、1 つの 128 ビットレーン、2 つの 64 ビットレーン、または 4 つの 32 ビットレーンとすることができ、`l s b s` での `I E E E` 丸めのための補助拡張、および各レーンの `m s b s` でのオーバーフロー検出／補正のための拡張符号を備える。データは、桁上げ保存形式で表される符号付き 2 - s 補数である。

20

【0241】

6 ビットの `Z i n` 1 次指数入力 504 は、11 ビットの `I E E E` 二重指数に対応する基数 32 指数である。また、`I E E E` シングルおよび `S I M D - 2 B F L O A T 1 6` の上位レーン (レーン 1) の 3 ビット基数 32 指数としても使用される。指数は、`B F L O A T` および `I E E E` シングルの場合は基数 32 に変換され、`I E E E` 倍の場合は剰余 1023 に変換され、基数 32 にも変換される。指数は、指数のアンダーフローおよびオーバーフローの検出および処理を支援するために、1 ビットだけ拡張されてもよい。

【0242】

3 ビットの `Z i n` 2 次指数入力 506 は、浮動小数点 `S I M D - 2` モードが選択された場合、レーン 0 に対応する 8 ビットの `B F L O A T` 指数に対応する基数 32 指数である。2 次指数入力は他の全てのモードに対して無視されるが、設計者は、アキュムレータの臨界タイミングループ内の論理を単純化するために、他のモードに対する 2 次入力に 1 次指数を複製することを要求することができる。指数は、指数のアンダーフローおよびオーバーフローの検出および処理を支援するために、1 ビットだけ拡張されてもよい。

30

【0243】

アキュムレータ論理は、有効な `Z i n` データ 508 がアサートされるときを除いて、その現在の状態を保持する。「1」条件は、`Z i n` 仮数上の入力データを示し、指数は選択されたモードに対して有効である。初期化フラグが `Z i n` データが有効であると同時に「1」である場合、`Z i n` の値は、何も追加せずにアキュムレータレジスタにコピーされる (オーバーフロー (右シフト) がある場合、または正規化シフトを可能にするシフトコンバイナにおける減算から生じる十分な先行符号ビットがある場合、正規化左シフトは -32、0、32、64、または 96 ビットになることがある)。初期化フラグが「0」であり、かつ `Z i n D a t a V a l i d = 「1」` である場合、`Z i n` 入力のデータが (適切なアラインメントシフトで) アキュムレータ出力の現在値に追加される。

40

【0244】

代表的な実施形態では、`t l a s t` フラグ 510 を使用して、累積合計を出力させ、次の有効な入力でアキュムレータを再初期化する。`T l a s t` は、累積された一連のサンプルの最後の有効なサンプルに対して「1」に設定される。アキュムレータは、最後の入力

50

サンプルが追加された合計を出力するときに有効な出力データをアサートし、次の有効なデータ入力で再初期化する（再初期化とは、ZINデータに何も追加せずに読み込むことを意味する）。Tlastが、「1」も有効なデータなしで「1」になると、アキュムレータは、有効なデータを出力せずに、次の有効なデータで再初期化する。

【0245】

別の実施形態では、初期化フラグ510は、加算器へのアキュムレータフィードバックを強制的にゼロにして、Zin上の値がアキュムレータにコピーされるようにする。コピーされた値は、入力のデータにオーバーフローがある場合、または先頭のゼロが31個を超えている場合に、モードが浮動小数点の場合に再正規化される。また、初期化フラグは、アキュムレータが新しい初期データで書き込まれるのと同じクロックでのみアサートされるように、有効なアキュムレータデータをゲートする。このゲートは、すべての有効な入力に対して有効なアキュムレータデータが存在するように、累積構成ビットによってオーバーライドされる。

【0246】

出力の128ビット仮数部512は、桁上げ保存形式（2つの128ビットベクトル）で表される。出力は、128ビットレーン1つ、64ビットレーン2つ、または32ビットレーン4つである。データは、桁上げ保存形式で表される符号付き2-s補数である。データは、AccumulatorDataValidフラグが付いている場合にのみ有効である。データ出力は、有効なデータの1クロック後にアサートされ、データoutは、累積された合計を新しい初期データでアキュムレータレジスタに置き換える前の累積値である。

【0247】

推定先頭符号ビット出力514は、内部32ビットおよび64ビット再正規化左シフトまたは32ビットオーバーフロー補正右シフト前のアキュムレータにおける先頭符号ビットの数を示す。これは、アキュムレータ出力で繰り返される符号ビットの数を示す符号化出力である。この推定値は、桁上げベクトルと保存ベクトルとの間のビットの分布に応じて、n個またはn-1個の繰り返し符号ビットを示す1ビットの誤差を有することができる。符号化された先頭符号ビットは、データと指数をIEEE形式に再正規化するために最終加算器論理によって使用される。最後の加算器論理（340）は、データを復号して、くりこみを完了するために追加のシフトが必要かどうかを判定する。

【0248】

1次アキュムレータ指数出力516は、名目上、IEEEシングルおよびfloatについては8ビット超過127であり、IEEEダブルについては11ビット超過1023である。これらはすべての浮動小数点に対して12ビットの超過2047コードに変更され、浮動小数点オーバーフローとアンダーフローを簡単に検出して例外フラグを作成できるようになる。指数の5lsbsから仮数を0に左シフトし、これらのゼロ化されたビットをドロップすることによって、シフト結合器およびZ入力シフト回路によって、12ビット指数が7ビット基数32指数に変換される。アキュムレータの構成が浮動小数点モードではない場合、アキュムレータの指数出力は未定義である。

【0249】

2次アキュムレータ出力518は、浮動小数点SIMD-2（BFLOAT16のみ）モードにのみ使用される8ビット超-127指数の最上位3ビットである。この出力は他のモードでは定義されませんが、設計者は、アキュムレータまたは最終加算器の論理を単純化する場合に、これらを使用してプライマリ指数出力のlsbsを複製する必要がある。指数アンダーフローおよびオーバーフローの検出および処理を支援するために、指数を1ビット拡張することができる。アキュムレータが固定小数点モードのいずれかで動作している場合も、指数出力は未定義である。

【0250】

オプションとして、アキュムレータデータ有効出力は、それが「1」である場合、先行符号、仮数、および指数を含むアキュムレータ出力上の有効なデータを示す（これらのフ

10

20

30

40

50

ィールドのいくつかは、いくつかのモードでは未定義である）。それ以外の場合、データは無効と見なされる。有効なアキュムレータデータは、構成に応じて有効なデータの1または2クロック後の「1」であり、初期化フラグおよび累積構成によってゲートされる。

【0251】

設定には、たとえば、

1. SIMD設定の制御が含まれる場合があり、固定小数点操作00=1レーン、10=2レーン、11=4レーンのレーン数を設定する制御と、
2. 浮動小数点選択の固定小数点または浮動小数点の制御であって、浮動小数点の場合、SIMDは内部的に“00”に強制される制御と、
3. `init=1`を保持するのと同等の累積ビットなしの制御（サイクルごとに入力を入力に渡す）と、
4. 入力合計に追加されるたびに有効なデータを出力する、累積合計のCumsumビットの制御と、
5. Cumsum制御ビットの制御と、
6. フォーマットビット（3）セット数値フォーマット、固定／フロート、レーンの数の制御と、
7. `init=1`を保持するのと同様の（外部アキュムレータの処理を実行可能な）アキュムレータなしビットの制御と、
8. `cumsum`と組み合わせることができる、データ有効遅延ビットの制御とを含めることができる。

【0252】

累積構成ビットがセットされると、アキュムレータ315の有効なデータoutは、すべてのZinDataValidに対応する「1」になる。これにより、カウンタおよび積分に使用されるような累積合計を生成できる。`cumsum=「0」`の場合、有効なデータは、初期化フラグ=「1」と同時に到着した新しい有効なデータでアキュムレータレジスタが更新される前のクロックサイクルでのみ有効である。Cumsumは、最新の入力を加えた後に出力が累計されるように、有効なデータを1クロック遅延させる必要もある。浮動小数点および固定小数点のフォーマットとレーン数のフォーマット設定の例を表16に示す。

【0253】

【表16】

表16:

コード	固定	レーン	フォーマット
000	0	1	IEEE single
001	0	2	2x Bfloat16
010	0	1	IEEE double
011	0	4	illegal
100	1	1	INT 128
101	1	2	2x INT64
110	1	1	illegal
111	1	4	4x INT32

【0254】

非累算構成ビットは、「1」に設定されたときにアキュムレータ315のフィードバックを常に0にする。これにより、アキュムレータは、浮動小数点の正規化パススルーと、固定小数点の単純なパススルーになる。内部的には、これは強制的に`initializ`フラグを常に「1」にすることと同じである。この構成ビットは、（条件フラグ論理内で）初期化フラグを「1」に強制する外部手段がある場合には削除することができる。

【0255】

図43は、アキュムレータ315を示す回路図であり、2番目のレーン（Bfloatのみ）の指数がグレー表示され、クリティカルパスが強調されている。RAEアキュムレータ315は、アキュムレータ出力レジスタからアキュムレータを通してレジスタに戻る臨界フィードバック経路内の論理レベルを最小にするために、基数32シフトを使用する。
【0256】

図44は、アキュムレータ315に用いられるN*32シフト回路517を示す回路図である。図45は、同様にアキュムレータ315で使用される、SIMDスプリット用に論理が追加された深さ $\log_2(n)$ を有するタリットゥバー回路519の構造を示す回路図である。
【0257】

9. ブール論理回路325

図46は、ブール論理回路325のブール論理段階520を示す回路図であり、ブール論理回路325に入力される各ビットに対してブール論理段階520を使用して、32ビットを独立して構成することができる。ブール論理回路325は、RAE300 ALU関数を2つの入力ブール関数のフルセットで補完するために使用される構成可能なビットごとのブール論理回路325を提供する。32ビットデータ経路上の各ビットは、同じビット位置にある入力ビットの任意の2つの入力ブール関数が可能な、独立にプログラムされたブール論理段階520を有する。ブール論理回路325は、X入力365によって制御されるビットごとの2:1選択を提供する方法も提供する。32ビット出力は、32ビット出力の32入力NAND522によって補足され、条件フラグ出力論理に送出される。
【0258】

Z入力は、比較ブロック320内のインデックスカウンタからZローテータを介して接続され、そのカウンタのアドレス生成器としての使用を可能にする。このパスには、混合基数アルゴリズムから構築されたFFTで使用するZ入力シフタの前に、選択可能な有線ビット反転も存在する。
【0259】

ブール論理回路325は、一般的な使用に加えて、入力カウンタのビット反転、マスクおよび回転（Zシフト論理の組み合わせ）順列を含む複雑なアドレスシーケンスを生成するためのカウント順列を可能にするように特に設計されており、これは比較ブロック320内のインデックスカウンタによってRAE300内で生成することもできる。ブール論理はまた、1つのソースの回転と、回転されたソースと第2の固定ソースとの間のビットごとの選択とを含む単純なフィールドマージを可能にするように設計される。
【0260】

Z入力は、ブール論理回路325への1次32ビット入力の1つである。それは、セレクタ397（第3のデータ選択（ステアリング）マルチプレクサ397として示されている）を介して、Z入力シフタ330出力または比較ブロック320 Z出力（カウント出力を兼ねる）のいずれかに接続されている。このブール論理回路325は入力セレクタを含むので、ブール論理回路325には別個のZシフト入力とZ比較入力がある。Z入力のビットは、論理モードの各ビットで2つのブール変数の1つとして、または選択モードの選択変数として機能する。Zが「0」のとき、出力は、Y入力370によって選択されるように、2つの下位レジスタビットのうちの1つである。Zが「1」の場合、出力は、論理モードのY入力または選択モードのX入力365によって選択された2つの上位レジスタビットのうちの1つである。
【0261】

Y入力370は、ブール論理回路325への1次32ビット入力の1つである。これは、比較ブロック320のY出力（Y入力を通過させることができる）に接続される。Y入力370は、「0」のとき偶数レジスタビットを選択し、「1」のとき奇数レジスタビットを選択する。上位2ビット（Z＝「1」のときに選択される）は、論理モードのときはYで、選択モードのときはXでアドレス指定される。比較ブロック320は、YまたはZRAE入力をY出力に接続するようにプログラムすることができ、Zおよびシフトされた

10

20

30

40

50

Zを用いてビットごとの演算を行う方法を提供する。

【0262】

ブール論理回路325へのX入力、選択モードが設定されている場合にのみ使用される補助入力である。上位2つのレジスタビットで定義されるXの1ビット関数は、Zが「1」であり、選択モードが設定されている場合に選択される。それ以外の場合、X入力は無視される。

【0263】

コンフィギュレーションインタフェース524は、コンフィギュレーションレジスタ526ビットにアクセスする役割を果たす。ブール論理回路325の32ビットの各々に対して4つの構成ビットがあり、各ビット位置に対してブール関数を独立に設定する。モードを通常または選択モードにグローバルに設定し、ZシフトまたはZ用比較器のいずれかからの入力選択入力を設定するために、二つの付加的な構成ビット(534)がある。

【0264】

ブール論理回路325は、32ビットのQ出力528を有する。各出力ビットは、構成レジスタにプログラムされたそのビットのブール論理関数の結果です。選択モードでは、選択論理入力の一部のYをXに置き換えるために論理関数を変更される。フラグ出力532は、X、Y及びZ入力ビットのいずれか又は全ての関数である1ビット出力を生成する手段を提供する。フラグ出力は、32ビットQ出力の32ビットNAND関数です。

【0265】

ブール論理回路325の構成は、各ビットスライスに対して4ビット論理構成を保持する4×32レジスタファイル526と、ブロック全体に対して論理(0)または選択モード(1)を選択する1ビットと、Z入力(0=論理比較、1=Zシフタ)のソースを選択する1ビットを有する2ビットグローバルレジスタ534とを含む。各ビットスライスの4ビット構成では、論理モードのときに、YおよびZビット入力の4つの可能な組み合わせの出力値がそのビットスライスに設定される。選択モードでは、出力するレジスタ内容を選択する際、Z入力が「1」の場合、X入力の値がY入力の値に置き換えられます。レジスタコードとモードによるビット関数を以下の表17に示す。

【0266】

論理モードの場合、各ビットスライスの論理は、スライスへのZおよびYビット入力によってアドレス指定される4入力セクタである。2入力論理関数には、4つの入力の組み合わせがある。レジスタビットの選択のために、Z入力は重み2を有し、Y入力は重み1を有する。4つの構成レジスタビットを適切に設定することにより、モードが論理モードに設定されたときに、2つの入力の任意のブール関数をプログラムすることができる。選択モードでは、ZはXとYの1ビット論理関数を選択する。選択モードでは、第1層上位セクタの選択入力がY入力からX入力に変更され、Z入力は、Z=「0」のときにレジスタ0と1によって設定されるY(0、~Y、Y、または1)の1つの入力関数、またはZ=「1」のときにレジスタ2と3によって設定されるXの1つの入力関数を選択する。

【0267】

10

20

30

40

50

【表 17】

表17:

登録内容	論理モード関数	モードの 選択関数
0000	0	0
0001	Y NOR Z	Z ? 0 : ~Y
0010	Y AND ~Z	Z ? 0 : Y
0011	~Z	~Z
0100	Z AND ~Y	Z ? ~X : 0
0101	~Y	Z ? ~X : ~Y
0110	Y XOR Z	Z ? ~X : Y
0111	Y NAND Z	Z ? ~X : 1
1000	Y AND Z	Z ? X : 0
1001	Y XNOR Z	Z ? X : ~Y
1010	Y	Z ? X : Y
1011	Y OR ~Z	Z ? X : 1
1100	Z	Z
1101	Z OR ~Y	Z ? 1 : ~Y
1110	Y OR Z	Z ? 1 : Y
1111	1	1

【0268】

Z 入力は Z シフト後に行われ、RAE Z 入力または比較論理の Z 出力（比較論理のインデックスカウント論理に接続可能）のいずれかを入力するオプションがある。また、シフタの入力側の Z 接続は、32 ビット入力の有線ビット反転のための接続を有する。この構成は、32 ビット範囲内の任意の場所でアドレスフィールドをシフトまたは回転させることができ、0 または 1 でビットを選択的にマスクすることができ、またはカウントビットを反転させることができる、非常に柔軟なアドレス生成能力を提供する。Z シフトの前の有線ビット反転はまた、混合基数構築高速フーリエ変換に必要な回転ビット反転シーケンスの生成を可能にする。プール論理回路 325 の出力は、デコード又はデータ依存条件フラグのような使用のための単一ビット出力 532 を提供するためにビットを集約する 32 入力 NAND ゲート 522 も有する。これを拡張して、各 SIMD レーン内の 8 ビットにそれぞれ関係する 4 ビット出力フラグと、SIMD サイズに関係なくレーンごとに 1 ビットを提供する結合ネットワークとを提供することができる。

【0269】

10. 回路 320 の比較

図 47 は、min/max ソートおよび比較回路 320（より一般的には比較回路 320 とも呼ばれる）を示すハイレベル回路およびブロック図である。図 48 は、比較器 556a の min/max ソートおよび比較回路 320 を示す詳細な回路およびブロック図である。比較回路 320 は、入力、内部インデックスカウントおよび内部定数の比較結果に基づいてデータフローおよびシーケンスを変更する RAE 回路 300 の能力を追加する。比較回路 320 は、各レーンについて別々の比較結果を使用して 2 つまたは 4 つの SIMD レーンをサポートするように分離可能である。

【0270】

この比較回路 320 は、

1. 最小値の最初の出現のインデックスを使用してストリームの最小値を累積する機能と、
2. 最大値の最初の出現のインデックスを持つストリームで最大値を累積する機能と、

3. 2つの入力ソート（1レーン、2レーン、4レーンのSIMD）し、大きい場合は交換する機能と、
 4. アキュムレータ315と同時に実行し、同時にリセットすることができるサンプルをカウントする機能と、
 5. 現在のインデックスカウントがインデックス入力と一致する場合を除いて、すべてのサンプルをゼロにし、他の入力を出力する機能と、
 6. `threshold positive`（閾値を超えるサンプルは合格し、小さいサンプルは定数に置き換えられ、閾値を超えるサンプルをカウントする）機能と、
 7. `threshold negative`（閾値を超えるサンプルは、定数に置き換えられ、閾値を下回るサンプルをカウントする）機能と、
 8. 入力を変更せずに渡す（ブール論理回路325への入力のためのフロースルー）機能と、
 9. 比較条件を満たす入力のみを渡す（ゲートデータは有効）機能と、
 10. トリガ条件の前または後にのみ入力を渡す機能と、
 11. 等値フラグと小なりフラグの出力（1レーン、2レーン、4レーンのSIMD）機能と、
 12. (`rotator`と`Boolean`を使用して、ビット反転、回転、マスクを行うことができる) アドレスカウントを生成する（上記のすべてのモードは、サポートされているすべての浮動小数点、符号付き整数、符号なし整数、および32ビットデータの1、2、または4SIMDレーンに適用され、SIMDモードでは、すべてのレーンが同じ構成とデコーダマッピングを共有する必要がある、各レーンはこのブロックで個別に扱われ、SIMDモードが選択されると、32ビットのインデックスカウントも同様の数のSIMDレーンに分割される）機能と
- を果たす。

【0271】

Y入力542は、Y`_valid`（546）を伴う。どちらかの有効値が「0」の場合、バイパスされない限り（一般に、定数またはフィードバックが入力として選択されている場合）、比較は処理されない。32ビットのZ入力544は、レーン交換ならびに融合比較演算の一部としてシフトまたは回転のためにシフタを使用することができるように、Zシフタ論理出力によって供給される。符号付き入力に対して符号なし比較器を適切に使用するために符号付き入力を選択されると、入力の最上位ビットが反転される。フラグ入力（546）は、比較結果に対する追加の検証であり、これを使用して、最小または最大のストリーミングの終了、サンプルのタグ付け（パススルー）、カウンタまたはトリガイベントのリセット、およびその他の制御イベントを行うことができる。フラグ入力は、FC200シーケンスカウンタまたは隣接するRAE300のフラグ出力から供給される。リセット入力（546）は、「1」がアサートされたときに有効なYおよびZに関係なく、カウンタおよびデータ出力レジスタをリセットする。

【0272】

Y出力548は1次データ出力である。YまたはZ入力またはYまたはZ定数レジスタから選択されたデータである。YまたはZソースの選択は、比較結果と結果デコードのプログラミングに依存する。ライブデータまたは定数の選択は、構成設定によってYとZに対して個別に設定される。Y出力`valid`（552）は、Y出力がダウンストリームブロックに対して有効であることを示す。Y出力有効信号は、比較条件、入力フラグ、およびYおよびZデータ入力有効信号のプログラマブル関数である。プログラム可能なデコードは、Y出力レジスタのクロックイネーブルおよびリセットも制御し、出力がデータをキャプチャして保持したり、比較条件に応じてカウントしたりできるようにする。

【0273】

Z出力554は、有効なZ出力（552）も有する。Z出力は、構成設定に応じて、出力がY出力選択（Y出力がZinの場合はYin、その逆の場合はYin）またはインデックス数の反対になる2次データ出力である。Z出力有効信号は、比較条件、入力フラグ

10

20

30

40

50

、およびYおよびZデータ入力有効信号のプログラマブル関数である。プログラム可能なデコードは、Z出力レジスタのクロックイネーブルおよびリセットも制御し、出力がデータをキャプチャして保持したり、比較条件に応じてカウントしたりできるようにする。条件フラグ出力（552）は、比較結果、フラグイン、YおよびZデータ有効入力、およびリセットのプログラマブル論理機能である補助制御信号である。これは、条件フラグ制御として下流のRAE300によって、およびシーケンシングに影響を与えるためにフラクタルコア200シーケンサによって使用することができる。フラグを使用してFC200シーケンサを制御する場合は、パイプライン遅延を含めるように注意する必要がある。

【0274】

比較回路320は、接続を選択する一組の構成ビットによって定義される多くの動作モードを有する。構成は、3つの定数レジスタの設定も含む。コンフィギュレーションは、7つのデータ経路セレクトの設定、SIMDモードの選択（2ビット）、入力符号タイプ（2ビット）、制御マッピングへの比較デコード（49ビット）の定義、および3つの32ビット定数レジスタの設定で構成されます。コンフィギュレーションは、構成と定数に分けられる。コンフィギュレーションには、7つのデータ経路選択ビットと2つのSIMDモード選択ビットが含まれる。

【0275】

この比較回路320は、SIMDマグニチュード比較器556と、データステアリングセクタ564と、レジスタと、加算器／カウンタ562（カウンタ572付き）と、プログラム可能なデコーダ558とを含み、データとカウンタパスとレジスタとを制御する。32ビット比較器556は、1つの32ビットレーン、2つの16ビットレーンまたは4つの8ビットレーンのためのモードを有する。各レーンに対して「Equal-to」および「Less-Than」出力を生成する。デコーダ558は、各レーンの比較条件をデコードし、それを有効およびフラグ入力でゲートして、各レーンのデータステアリング制御を生成し、カウンタ572、レジスタフリップフロップクロックは、データ出力レジスタおよびカウンタレジスタのそれぞれについて、各レーンについてイネーブルおよびリセットする。比較器556への入力は、ブロックのYおよびZ入力、YおよびZ定数レジスタ、またはZ入力用のY出力レジスタ、またはY入力用のカウンタレジスタから来ることができる。これにより、比較条件に基づいてカウントを生成する柔軟性、最小または最大を累積する機能、カウントの発生、およびその他の用途が提供される。

【0276】

図48を参照すると、マグニチュード比較器556は、その2つの入力の相対的な大きさを決定し、結果信号（等しいおよびより小さい）を使用して、ブロックの残りの部分におけるデータのステアリングを制御する。マグニチュード比較器556は、基本的な形式では、2つの32ビットワードを比較し、比較器568から二つの出力、AはBに等しい、AはBより小さい、を生成する。比較器556は、32ビットワード内の1つ、2つまたは4つのSIMDレーンに対応する1つの32ビット、2つの16ビットまたは4つの8ビット算術比較器として動作することを可能にする分割ゲートを含む。8つの3:1マックス566のセットは、SIMDモードに応じて各レーンに適切な比較ツリー出力を選択する。4つ未満のSIMDレーンの場合、信号出力は複製され、SIMDレーン内の8ビットスライスごとに別々に出力される。比較器およびカウンタ572の論理の外側では、ブロックの残りの部分は、すべてのデータを4つの独立した8ビットレーンとして扱う。8ビットの比較をツリー構造の継続と置き換えて、入力での1ビットの比較にすることで、追加の最適化を利用できる。入力レイヤは、1つの入力反転された半加算器である。これにより、ツリーの深さが1ゲートレイヤだけ減少し、SIMD2の場合と同様のSIMD4モードのためのブロッキングゲートの追加が必要となる。最適化ツリーは、符号補正とSIMD出力選択を除いて、計数バッファなしの258個の2入力ゲートと12個のゲート遅延を持つ。最適なバイナリ比較の遅延およびリソース（2つの入力ゲート）のメトリックは、 $T(n) = 2(\log(n) + 1)$ および $C(n) = n \log(n) + 3n - 1$ である。

【0277】

比較器556はまた、符号付き2の補数および符号マグニチュード入力に対する補正を有する。比較器556は、両方の入力と同じ数体系を有すると仮定する。2の補数の場合、比較では負の値が正の値より大きいと誤って比較される。これは、数値システムが（符号に関係なく）2の補数である場合は常に、符号ビットを反転することによって修正される。符号マグニチュードの場合、符号を反転すると、負の数は正の数よりも小さいものとして正しくテストされるが、2つの負の入力は、負の数が大きくなると負の数が大きくなるため、予想される比較結果の逆になる。これは、符号ビットを常に反転し、符号が負の場合にのみ残りのビットを反転することによって、符号の大きさについて正しい比較結果を提供するように修正される。これにより、負の数の1の補数が実行され、 -0 が -1 、 -1 が -2 などにマップされる。数値表現が変更されても、変更によって正しい比較結果が得られる。負の数は1だけ減らされ、一意の -0 を格納できるようになる。各ビットの符号補正は、符号付きモードの関数として、各SIMDモードについて表18に示される。正規化浮動小数点値は、上記の補正を使用して符号の大きさの整数として解釈される場合、正しい比較結果を生成する。非正規化浮動小数点値と無限浮動小数点値も、符号とマグニチュードの補正を使用して正しく比較される。

【0278】

【表18】

表18:モード別の比較器入力符号補正

ビット数	SIMD1	SIMD2	SIMD4	全体
31	符号付き xor 31	符号付き xor 31	符号付き xor 31	符号付き xor 31
30:24	sm*31 xor (30:24)	sm*31 xor (30:24)	sm*31 xor (30:24)	(31*sm) xor (30:24)
23	sm*31 xor (23)	sm*31 xor (23)	符号付き xor 23	(SIMD4? 符号付き:sm*31) xor 23
22:16	sm*31 xor (22:16)	sm*31 xor (22:16)	sm*23 xor (22:16)	((SIMD4? 23 : 31)*sm) xor 22:16
15	sm*31 xor (15)	符号付き xor 15	符号付き xor 15	(SIMD1 ? sm*31 : 符号付き) xor 15
14:8	sm*31 xor (14:8)	sm*15 xor (14:8)	sm*15 xor (14:8)	((SIMD1 ? 31 : 15)*sm) xor (14:8)
7	sm*31 xor (7)	sm*15 xor (7)	符号付き xor 7	(SIMD1*31*sm + SIMD2*15*sm + SIMD4*7* 符号付き) xor (7)
6:0	sm*31 xor (6:0)	sm*15 xor (6:0)	sm*7 xor (6:0)	((SIMD1*31 + SIMD2*15 + SIMD4*7)*sm) xor (6:0)

【0279】

比較器の後のSIMD選択は、4つの8ビット比較結果対、2つの16ビット結果の2つのコピー（8ビットレーンごとに1つの結果、2つの上位レーンと2つの下位レーンに同じコントロールがある。）、または単一の32ビット比較信号対の4つのコピーのいずれかを選択する。これは、エンコードされたSIMD設定を選択として使用して、8ビットレーンごとに1対の3:1マルチプレクサ566として実装される。比較およびカウンタ572の外側では、すべてのデータは4レーンSIMDとして扱われ、SIMD-4より小さい場合には、レーンは重複制御を受ける。

【0280】

図49は、デコーダ回路558（27制御、49構成ビット）を示す回路図である。比較回路320の柔軟性は、制御信号を生成するデコーダ558のプログラム可能性に依存する。デコーダ558への入力は、各8ビットレーンに対する2ビット比較結果と、プロ

ック入力からの単一ビットリセット、フラグ、YおよびZデータ有効信号である。生成される制御には、レーンごとのデータステアリングマックス、レーンごとの `count_bypassmux`、レーンごとのY、Z、およびカウントレジスタクロックイネーブル、レーンごとのカウントレジスタリセット（Z-muxがカウンタ572を選択し、YおよびZデータ出力がリセットを使用しない場合、Zレジスタはカウントレジスタリセットを使用する）、およびフラグ、およびYとZのデータ有効出力が含まれる。各制御出力の設定ビットはすべてのレーンに適用されるが、各レーンに個別のゲートがあるため、各レーンの比較結果が各レーンの出力に個別に影響を与えることができる。内部で生成されるINIT信号は、データの有効性に関係なくリセット入力のアサートされたときに設定され、フラグ入力が0のときにクリアされ、オプションで入力データの有効性によって検証される。INITは、フラグがアサートされたときに、新しいデータブロックの先頭でレジスタの初期化として次の有効なサンプルで使用できるように、オプションで再び設定される。ほとんどのフラグ付き操作では、フラグがアサートされてデータブロックの終わりをマークし、それによってインデックスカウンタ572をリセットし、および／または（例えば）累積min/maxを再開する。比較器は、各レーンについて「より小さい」および「等しい」出力を生成する。レーンを結合すると、8ビットレーン出力ごとに合成レーンの出力が複製されます。デコーダ558は、追加の「より大きい」信号を生成し、3つの比較結果のうちのどれをデコーダの残りの部分に転送するかを選択するグローバル比較選択回路を有する。各制御デコーダは、選択された比較結果またはその逆または「1」または「0」を選択して、各制御を別々にゲートすることができる。初期化およびフラグ入力は、これらの信号が完全に復号化されたローカル比較信号と論理的にORされ、次にYまたはZまたは（YおよびZ）または「1」であり得る有効な複合データとANDされて各ローカル制御を生成することを可能にするイネーブル制御を有する。各ローカルコントロールの設定は4つすべてのレーンで共有されるが、そのレーンの比較入力を使用する。一部の制御関数はすべての入力信号を使用しないため、これらの未使用の入力は定数に関連付けられ、設計外で最適化する。

【0281】

表19に、各コントロールの構成入力を要約する。デコーダ558には、27個の制御を生成するために49個の構成ビットが関連付けられている（そのうちのいくつかは、4レーンに対して4個のコピーを有する）。入力符号モードとSIMDモードを設定するための追加設定ビットがある。

【0282】

10

20

30

40

50

【表 19】

表19:デコーダ制御別の設定ビット

	比較セル	Init en	Flag en	Y ignore	Z ignore	Num cfg bits	Num ctls
ステアリングレーン	Cfg* 2	0	0	1	1	2	4
Cnt バイパスレーン	Cfg* 2	Cfg	0	1	1	3	4
Yreg ceレーン	Cfg* 2	Cfg	Cfg	Cfg	cfg	6	4
全てのYreg rst	0	0	0	1	1	0	1
Zreg ceレーン	Cfg* 2	Cfg	Cfg	Cfg	cfg	6	4
Zreg rstレーン	zmux=cntの場合はcntrst、それ以外の場合はnoneを使用する。						
Cnt ceレーン	Cfg* 2	Cfg	Cfg	Cfg	cfg	6	4
Cnt rstレーン	cfg* 2	cfg	cfg	1	1	4	4
フラグ出力	Cfg* 2	cfg	cfg	Cfg	cfg	5	1
Y有効	Cfg* 2	cfg	Cfg	cfg	Cfg	6	1
Z有効	Cfg* 2	cfg	Cfg	cfg	Cfg	6	1
コモン	Cfg* 2		cfg	cfg	cfg	5	

【0283】

比較回路320は、内部および外部使用のためのインデックスカウンタを生成するための32ビット加算器／カウンタ562を含む。また、閾値動作にヒステリシスを導入するために閾値を変調する加算器として設定することもできる。カウンタの加算器の出力は、別々のクロックイネーブルを持つ2つの同一のレジスタにルーティングされる。これらのうちの1つは、カウントレジスタであり、その出力はカウンタ加算器および内部使用のための比較器論理ブロックの1つの入力にフィードバックされる。2番目のレジスタ（Z出力レジスタ）は、個別にクロックが有効になっており、ストリーミングデータの最小値または最大値のインデックスを保持するためのカウンタを条件付きでキャプチャすることを目的としている。加算器562の一方の入力は、Y入力またはKz定数またはカウンタ572フィードバックのいずれかから選択され、他方の入力は、ブロック構成からの増分定数である。SIMDモードに応じてレーン間の桁上げをゲートでオフにすることによってSIMD演算が選択されると、インデックスカウンタの加算器はSIMDレーンに分割される。増分定数は、2レーンおよび4レーンSIMDモードの各レーンの増分を含むように調整する必要がある。インデックスカウンティングについては、カウンタ572は、典型的には1だけインクリメントされ、カウントレジスタは、有効なサンプルについてクロック可能である。カウンタ572は、フィードバックmuxをYinに、カウントバイパスmuxをクロックイネーブルに設定した加算器をバイパスするステアリング論理によりリセットされる。これにより、カウンタレジスタにYinの値がロードされる（Yinの正しいリセット値に依存するのではなく、追加の定数レジスタとmuxを同等に使用できる）。カウンタ572の増分を1以外に変更して、lsbから上にシフトされたカウンタと負のカウントをサポートできる。ヒステリシスを有する閾値化は、カウンタ572のフィードバックをYinに設定し、ヒステリシス不感帯幅の半分の閾値をYに入力し、不感帯幅を増分値定数にロードする。カウンタ572の出力は、比較器ブロックの一方の入力に供給され、Zinポート上のインデックスと比較される。カウンタレジスタmuxは、前の比較の結果に基づいて、変更された閾値に対してYinまたはYin+インクリメントのいずれかを選択する。カウンタ572のYinは、構成によって設定されるカウンタkmuxを使用して定数Kzに置き換えることができる。カウンタ572の出力は、Z出

力を介して行われ、Z出力は、選択可能な有線ビット反転および比較バイパスを介してZ入力シフタに接続され、回転、ビット反転およびブール論理によってカウントを並べ替えることによって、かなり複雑なアドレス生成を可能にする。

【0284】

ステアリング論理は、比較器入力のためのセレクタ（マルチプレクサ）564と、データ経路内の交換／代替マルチプレクサと、クロックイネーブル付きの出力レジスタと、カウンタ572の出力と第2のデータ経路の出力とを切り替える出力選択とを含む。それはまた、ステアリングおよびカウンタ572論理のためのステアリング制御を生成するための関数および比較結果デコーダを含む。データ経路交換／代替マルチプレクサは、ソートユースケースでZレーンとYレーンを交換するために、または逆ブーリングおよび閾値処理のストリーミングユースケースでデータの定数を置き換えるために使用される。ストリーミング最小／最大使用ケースでは、左側のレジスタのクロックイネーブルを使用して、新しい最大値または最小値（使用に応じて）が発生したときにレジスタを更新し、そのクロックイネーブルのコピーを使用して、capレジスタが現在のインデックスカウントをキャプチャできるようにする。ストリーミングモードでは、有効な出力データは最後のサンプルフラグ（AXIS-4 TLASTまたは同等のフラグ）で修飾する必要がある。クロックフィードバックとレジスタ入力のためのマルチプレクサ制御、左右の選択マルチプレクサ（定マルチプレクサではない）、およびレジスタクロックイネーブルは、比較結果によって部分的に制御される。残りのマルチプレクサ制御は、ブロック構成によってのみ影響を受ける。比較器及びカウンタ572 SIMD制御のための別個の構成制御があり、インデックスが必要でない場合には、比較器及びステアリングとは独立したカウンタ572の使用を可能にする。表示されていない他のユースケースが考えられる。左側のデータ経路のマルチプレクサは、定数レジスタKxおよびKyからの入力のための選択を含む。これは、設定モードと比較の結果に基づいて、条件付きでデータを定数値（Kx、Kyの既定値は0である）に置き換えるために使用される。同様に、インデックスカウント論理のインクリメント値Dも定数レジスタからのものであり、既定値は1である。定数は、任意の数のメカニズム（例えば、設定ワードの一部として、何らかの種類のシリアル定数負荷インターフェースを介して、またはクロックにより、例えばX、Y、Z入力を介して有効になるが、これに限定されない）を介してロードされる。

【0285】

使用例としては、インデックスを使用したストリーミングの最小値または最大値、2つの入力ソート（2つの入力の同時最小最大）、逆最大ブーリング（内部的に生成されたインデックス数がストリーミングインデックスと一致する場合を除き、ストリーミングデータは0に置き換えられる。）、サンプル・カウント・インデックス生成、データステアリング、データ置換、アドレス生成、比較フラグ出力、および、閾値サンプルのカウントまたはヒステリシスのいずれかによる閾値処理、などがあげられる。

【0286】

図50は、比較回路320を用いたインデックス適用によるストリーミング最小／最大を示す詳細な回路およびブロック図である。比較回路320はまた、現在のY入力をY出力レジスタに保持された前のminまたはmaxと比較して、ストリーミングminまたはmaxを実行する。Yレジスタは比較器Z入力にフィードバックされ、Y入力に入力された新しいデータは比較器Y入力に入力され、新しい入力Yレジスタの前の極値と比較される。カウントレジスタは、有効なY入力（Y_valid_in = 「1」）ごとに使用可能にされるので、カウントは、リセット後の有効なサンプル（プラス初期値）の数になる。ステアリングバイパスマックスとカウントバイパスマックスは、このアプリケーションでは（初期化用を除いて）静的であるため、Yデータと次のカウントは、それぞれYおよびZレジスタ入力に常に存在する。新しい値が現在のレジスタ値を超えると、クロックイネーブルからYおよびZ出力レジスタがオンになり、これらのレジスタが新しい極値および対応するインデックスカウントで更新される。最後のサンプルには、入力セットの終了を示すフラグinput = 「1」を付ける必要がある。Y_in_validと一致す

10

20

30

40

50

るフラグにより、`Yout__valid`は「1」に移動して累積最大値または最小値、および出力時のサンプルのインデックスを検証する。表20は、ストリーミング最小およびストリーミング最大の両方の使用ケースの設定をまとめたものである。

【0287】

最小値または最大値を検出するには、比較結果にかかわらず、セットの最初のサンプルのデータレジスタにデータを送信する必要がある、カウンタ572は（通常は0だが、オフセットすることもできる）初期インデックス値に設定される。データレジスタの初期値は、デコーダの初期化信号を使用して強制的に設定され、リセット入力または有効化フラグ入力の後に設定される。これにより、比較結果に関係なく、最初のY値が初期極値として受け入れられる。カウント値は、2つの方法のいずれかで初期化できる。0に初期化されると、`init`信号は単にZレジスタと`count`レジスタを0にリセットする。ゼロでない場合、`init`がアサートされたときにカウンタ572をKzでロードするためにカウントバイパスをアサートすることを含む代替方法が使用される。ゼロ以外の初期インデックスの代替設定は、初期インデックス=0の場合と異なる設定について、表22の右側の列に示されている。図50のKzに続くルートは、定数からのカウンタ572の初期化を示す。レーンリセットは、カウントレジスタをリセットすることによって0に初期化する代わりに使用される。ストリーミング最小／最大アプリケーションもSIMDデータと連携し、各レーンは独自の極値とインデックスを維持する。有効なデータはすべてのSIMDレーンに適用されるため、SIMD処理では、すべてのレーンを同時に有効にする必要がある。また、レーン間のデータセットはすべて、すべてのレーンの同じサンプルで開始および終了する。

【0288】

10

20

30

40

50

【表 2 0】

表20: ストリーミング最小値とストリーミング最大値の回路320セットアップ比較

コントロール	最小	最大	ゼロ以外の インデックス init
ファンクション			
	Y=init以降の最小値 Z=init以降の カウント	Y=init以降の最大値 Z=init以降の カウント	
コンフィギュレーション			
SIMD	任意	任意	
Zin mux を比較	Yout reg	Yout reg	
Yin mux を比較	Yin	Yin	
データステアリング Kz mux	気にしない	気にしない	
データステアリング Ky mux	Yin	Yin	
カウンタ K mux	気にしない	気にしない	Kz
カウンタフィードバック mux	カウント reg	カウント reg	
Z出力 mux	カウンタ	カウンタ	
Ky	気にしない	気にしない	気にしない
Kz	気にしない	気にしない	初期カウント値
Kinc	1	1	1
デコーダ			
データステアリング制御	常に Y	常に Y	
カウントバイパス制御	0	0	Ydv & init
Yレジスタリセット	常に「0」	常に「0」	
YレジスタCE	(init Y<Z) & Ydv	(init Y>Z) & Ydv	
Zレジスタリセット	Ydv&init	Ydv&init	0
ZレジスタCE	(init Y<Z) & Ydv	(init Y<Z) & Ydv	
カウントリセット	常に「0」	常に「0」	0
カウントCE	Ydv	Ydv	
Y有効	フラグ&Ydv	フラグ&Ydv	
Z有効	フラグ&Ydv	フラグ&Ydv	
フラグ	「0」	「0」	

【0 2 8 9】

図 5 1 は、比較回路を用いた 2 入力ソートアプリケーションを示す詳細な回路およびブロック図である。2 つの入力ソートでは、Y 入力と Z 入力の値が比較される。Z > Y の場合、Z は Z 出力に、Y は Y 出力に配置される。それ以外の場合は、データを反対の出力に交換する。その結果、対の最小値は Y 軸上にあり、最大値は Z 軸上にある。このアプリケーションは、SIMD の各レーンが Z でそのレーンの最大値を示し、Y で最小値を示す場合に SIMD 演算をサポートする。SIMD では、すべてのレーンと一緒にデータを表示および生成する必要がある。カウンタ 572 は、その出力がこのモードでは見えないので使用できない。表 2 1 に、この 2 つの入力ソートの設定を示す。

【0 2 9 0】

10

20

30

40

50

【表 2 1】

表21: 比較回路320の2つの入力ソートの設定

制御関数	2 入力ソート	データパススルー
	Y=最小 Z= 最大	Y= Y Z= Z
構成		
SIMD	任意	任意
Zin mux を比較	Zin	気にしない
Yin mux を比較	Yin	気にしない
データステアリングKz mux	Zin	Zin
データステアリングKy mux	Yin	Yin
カウンタ K mux	気にしない	気にしない
カウンタフィードバックmux	気にしない	気にしない
Z出力mux	データ	データ
Ky	気にしない	気にしない
Kz	気にしない	気にしない
Kinc	気にしない	気にしない
デコーダ		
データステアリング制御	$Z \leq Y$	「0」
カウントバイパス制御	気にしない	気にしない
Yレジスタリセット	常に「0」	常に「0」
YレジスタCE	常に「1」	常に「1」
Zレジスタリセット	常に「0」	常に「0」
ZレジスタCE	常に「1」	常に「1」
カウントリセット	常に「1」	気にしない
カウントCE	常に「0」	気にしない
Y有効	Yvld&Zvld	Yvld
Z有効	Yvld&Zvld	Zvld
フラグ	「0」	気にしない

【0 2 9 1】

デコード論理は、2つの入力ソートの特殊なケースとして、単にYおよびZデータをYおよびZ出力に渡す（または交換する）ように設定できる。これに必要なのは、muxを固定値に設定し、データの有効値をそれぞれの出力にコピーすることだけである。このモードは、ブール論理ブロックにYまたはZデータを接続する場合に必要なことがある。Zレジスタmuxがカウンタに設定されている場合、データ経路スルーはY出力のために残り、Z出力はカウンタ572論理によって供給される。比較論理は、パススルーには使用されないので、このモードでは、フラグへの出力との比較のために、または、ステアリングMuxがパススルーに設定されているときにインデックスカウンタ572との使用のために、依然として使用可能である。

【0 2 9 2】

図52は、比較回路320を用いたデータ置換アプリケーションを示す詳細な回路およびブロック図である。比較回路320は、定数、第2の入力、またはインデックスカウンタに対する入力ストリームの比較を使用して、1次入力に対する定数または第2の入力の

置換を制御するデータ置換機能を有する。比較結果では、データを定数または他のストリームからの入力に置き換えることができる。インデックス数は出力としても使用できる。比較回路320の設定オプションには、

ケース1a: Zinをインデックスと比較して、YinまたはKyをZストリームに代入する、

ケース1b: Zinをインデックスと比較して、ZinまたはKzをYストリームに代入する、

ケース2a: Zinを定数(Ky)と比較して、Yinまたは同じ定数(Ky)をZストリームに代入する、

ケース2b: Zinを定数(Ky)と比較して、Zinまたは定数(Kz)をYストリームに代入する、

ケース3a: ZinとYinを比較して、YinまたはKyをZストリームに代入する、

ケース3b: ZinとYinを比較して、ZinまたはKzをYストリームに代入する、

ケース4a: Yinを定数(Kz)と比較して、YinまたはKyをZストリームに代入する、および

ケース4b: Yinを定数(Kz)と比較して、ZinまたはKzをYストリームに代入する、

がある。

【0293】

各ケースは入力muxとステアリング制御の極性の異なる置換である。それぞれのサブケースは、ステアリングマックスの解釈を除いて同じセットアップを有する。カウンタ572は、レジスタリセットを用いて初期化してもよいし、上述のようにカウントバイパスmuxおよびKz（またはYin）を用いて0以外に初期化することによって初期化してもよい。図52では、比較Yinおよび比較Zinmuxおよびステアリング定数muxは、異なるモードを説明するために構成された接続なしで示されている。上記の4つのケースの設定を表22に示す。

【0294】

逆プールは、ローカルインデックスカウントを維持しながら、同期されたインデックスストリームとデータストリームを受け入れる。インデックスストリームがインデックスカウントと等しい場合、データ値が渡される。それ以外の場合、出力データを0にする。この関数は、データ置換、Kz定数を0に設定したケース1、Zにインデックス入力、Yにデータ入力を行うことで実現される。セットアップは、表22の最後から2番目の列に含まれる。逆プーリングはまた、Y出力にYを出力するようにステアリングマルチプレクサを固定し、Zinがインデックスカウントに等しくないときに、比較結果を使用してYレジスタリセットをアサートすることによっても達成され得る。最大プーリングのこの代替構成では、電力消費がわずかに減少する可能性があるが、データ置換方法では、等しくないデータを0以外に設定できる。代替セットアップは、表22の最後の列に含まれている。

【0295】

10

20

30

40

50

【表 2 2】

表22: 比較回路320のデータ置換および逆プールの設定

制御						
ファンクション	ケース 1	ケース 2	ケース 3	ケース 4	Inv pool	Inv pool alt
	Z ? index	Z ? Ky	Z ? Y	Y ? Kz		
構成						
SIMD	任意	任意	任意	任意	任意	任意
Zin mux を比較	Zin	Zin	Zin	Kz	Zin	Zin
Yin mux を比較	カウンタ	Ky	Yin	Yin	カウンタ	カウンタ
データステアリング Kz mux	*	*	*	*	Zin	Zin
データステアリング Ky mux	*	*	*	*	Ky	Ky
カウンタ K mux	気にしない	気にしない	気にしない	気にしない	気にしない	気にしない
カウンタ フィードバック mux	カウンタ	カウンタ	カウンタ	カウンタ	カウンタ	カウンタ
Z出力 mux	カウンタ	カウンタ	カウンタ	カウンタ	カウンタ	カウンタ
Ky	Ky	Y 比較	*	*	0	気にしない
Kz	気にしない	*	*	Z 比較	気にしない	気にしない
Kinc	1	1	1	1	1	1
デコーダ						
データステアリング 制御	cmpr	cmpr	cmpr	cmpr	cmpr	常に Y=>Y
カウンタ バイパス制御	add	add	add	add	add	add
Yレジスタ リセット	常に「0」	常に「0」	常に「0」	常に「0」	常に「0」	Z /= idx
YレジスタCE	常に「1」	常に「1」	常に「1」	常に「1」	常に「1」	常に「1」
Zレジスタ リセット	リセット	常に「0」	常に「0」	常に「0」	常に「0」	init
ZレジスタCE	常に「1」	常に「1」	常に「1」	常に「1」	常に「1」	常に「1」
カウンタリセット	init	init	init	init	init	init
カウンタCE	Yvld&Zvld	Zvld	Yvld&Zvld	Yvld&Zvld	Yvld&Zvld	Yvld&Zvld
Y 有効	Yvld&Zvld	Zvld	Yvld&Zvld	Yvld&Zvld	Yvld&Zvld	Yvld&Zvld
Z 有効	Yvld&Zvld	Zvld	Yvld&Zvld	Yvld&Zvld	Yvld&Zvld	Yvld&Zvld
フラグ	「0」	「0」	「0」	「0」	「0」	「0」

【0 2 9 6】

データ置換はまた、閾値未満のデータが定数（典型的には閾値値または0）で置換され、閾値を超えるデータが渡されるように、データを閾値処理するために使用されてもよい。あるいは、閾値未満のデータを通過させることができ、閾値を超えるデータを定数（飽和など）で置き換えることができる。ケース2および4は表22にあり、一方の入力は比較のためにYまたはZ入力から来ており、他方は閾値定数に設定されている。ステアリングマルチプレクサ論理において閾値定数を設定することは、表23にリストされている様々な閾値を提供する。閾値は、データに使用されない入力を介して提供することもできる。インデックスカウンタ572は、（閾値が定数レジスタからのものである場合）閾値より上または下のサンプルをカウントするために、有効なサンプルをカウントするために、

またはフラグ入力または未使用データ入力上で有効なデータを使用する独立したイベントカウンタ572として使用することができる。

【0297】

【表23】

表23: 閾値モードによる回路320の構成設定の比較

入力	Z	Z	Y	Y
代替	下	上	下	上
ケース	2	2	4	4
Zin mux	Zin	Zin	Kz	Kz
Yin mux	Ky	Ky	Yin	Yin
ステアリングZ	Zin	Zin	Kz	Kz
ステアリングY	Ky	Ky	Yin	Yin
比較	$Z > Y$	$Z < Y$	$Z < Y$	$Z > Y$

10

【0298】

図53は、比較回路320を用いたヒステリシス適用による閾値を示す詳細な回路ブロック図である。ヒステリシス付き閾値は、現在のデータ値から離れる方向に、閾値を公称閾値から一定の距離だけシフトする。この目的は、信号ノイズが閾値を超える可能性を減らすことである。これは、カウンタの加算器とバイパスmuxを使用して、比較結果に応じて閾値にオフセットを追加するかどうかを指定することで実現される。カウンタ572の出力は比較器Yの入力に送られ、データはZ入力を介して入力される。比較結果は、カウントバイパスmuxとステアリングmuxを選択する。プログラムされる値は、閾値と、上限閾値と下限閾値の間の距離であるデルタである。デルタは正または負にすることができ、閾値は上限または下限のいずれかになる。

20

【0299】

30

40

50

【表 2 4】

表24: 回路320の閾値設定とヒステリシスの比較

制御	
関数	ヒステリシス閾値
構成	
SIMD	任意
Zin mux を比較	Zin
Yin mux を比較	カウント
データステアリングKz mux	Zin
データステアリングKy mux	Ky
カウンタ K mux	Kz
カウンタフィードバックmux	k-mux
Z出力 mux	気にしない
Ky	Constant out 値
Kz	低閾値
Kinc	Δ 閾値
デコーダ	
データステアリング制御	cmpr
カウントバイパス制御	cmpr
Yレジスタリセット	常に「0」
YレジスタCE	Zvld
Zレジスタリセット	常に「1」
ZレジスタCE	常に「0」
カウントリセット	「0」
カウントCE	Zvld
Y有効	Zvld
Z有効	Zvld
フラグ	「0」

【0300】

図54は、比較回路320を用いたフラグトリガ事象適用を示す詳細な回路およびブロック図である。比較回路320は、トリガ事象のためにある状態で待機し、次に状態を切り替え、リセット入力まで第2の状態のままであるように設定することができる。一般的なケースはフラグ入力でトリガされ、リセット入力で初期状態にリセットされる。これはフラグとリセットに依存するため、すべてのレーンが同じトリガでトリガおよびリセットされた場合にのみ有効である。初期状態では、cntレジスタは0にリセットされ、リセット入力のアサートされると強制される。カウントインクリメント定数は0に設定され、Kmuxはカウントレジスタへの入力が常にKzになるようにKzに接続する。Kzは0以外の任意の定数である。カウントレジスタのクロックイネーブルは、デコーダを介して入力されるフラグに接続され、フラグのアサートされると、レジスタがKzでロードされる。このようにして、トリガイベントまでカウントは0になり、その時点でKzになる。リセットがアサートされ、カウントレジスタが0にリセットされるまで、Kzのままである。比較器は、カウント値をKzと比較し、比較出力をデコードして、ステアリングマッ

クスを制御し、および／または、YおよびZデータ有効又はフラグ出力をゲートする。このようにして、トリガイメント時にデータを開始、停止、置換、交換、またはリダイレクトできる。Z出力をカウントレジスタに接続して、状態（0またはKz）への直接接続、またはデータステアリングおよび交換アプリケーションのためのデータmuxへの直接接続を提供することができる。データステアリングでは、0以外のKzが使用されている場合は、Kzである限り定数を使用できる。

【0301】

図55は、比較回路を用いた閾値トリガ事象適用を示す詳細な回路およびブロック図である。一般的な場合は、比較ブロックの外部でフラグを生成する必要がある、比較に基づくトリガの場合は、フラグを提供するために第2のRAE300を必要とする。閾値交差がイベントをトリガする特別な場合は、単一のRAE300比較回路320で実施することができる。このためのラッチ論理は、「ヒステリシス付き閾値」ユースケースと同様に設定されたカウンタ572を利用するが、トリガイメントによって閾値が表現可能な最小値または最大値のいずれかに移動するように閾値が設定され、比較関数がそれぞれ閾値未満または閾値を超えるように設定されるため、比較条件が満たされる（閾値が超過する）と、それが満たされなくなることはあり得ない。比較回路320は、フラグまたはリセット入力を用いてカウントバイパス制御を強制的にバイパスさせることにより、元の状態にリセットされる。比較条件を使用すると、定数をデータに置き換えたり（Kzは、閾値がYinを介して外部から供給されない限り、閾値に使用されるため、使用できないことに注意する）、データ出力を交換したり、データをデータの有効期間で操作したりできる。また、ZストリームとYストリームの両方のデータ有効を使用して、データマスを静止させ、データフローを中断または開始することもできる。トリガされたイベントのセットアップは、Kinc値が最小表現可能値－合計が最小表現可能値になるように閾値に具体的に設定されることを除いて、ヒステリシスを持つ閾値と同様である。もしトリガが閾値を超えた信号に反応して起こるならば、KincとKzで設定された閾値の和は数値モードで表現できる最小値（符号付き、符号なし、または符号絶対値／浮動小数点）でなければならない、比較は $Z \geq cnt$ でなければならないので、信号はシフトされた閾値より小さくなることはない。シグナルが閾値を下回ったときにトリガが発生する場合は、Kincと閾値の合計を表現可能な最大値にする必要がある。比較条件は、シフトされた閾値を超えないように、代わりに $Z < cnt$ にする必要がある。閾値トリガはSIMDモードでも動作するが、すべてのレーンが同時にリセットされることに注意する。

【0302】

10

20

30

40

50

【表 2 5】

表25: 比較回路320トリガイイベントの設定

制御			
ファンクション	フラグトリガ	閾値超過	閾値未満
構成			
SIMD	1	任意	任意
Zin mux を比較	Kz	Zin	Zin
Yin mux を比較	カウンタ	カウンタ	カウンタ
データステアリングKz mux	*	*	*
データステアリングKy mux	*	*	*
カウンタ K mux	Kz	Kz	Kz
カウンタフィードバック mux	k-mux	k-mux	k-mux
Z出力 mux	*	気にしない	気にしない
Ky	Constant out 値	Constant out 値	Constant out 値
Kz	低閾値	閾値	閾値
Kinc	Δ 閾値	MinRV - 閾値	MaxRV - 閾値
デコーダ			
データステアリング制御	cmpr	Zin ≥ カウンタ	Zin ≤ カウンタ
カウンタバイパス制御	cmpr	Zin ≥ カウンタ	Zin ≤ カウンタ
Yレジスタリセット	常に「0」	常に「0」	常に「0」
YレジスタCE	Zvld	Zvld	Zvld
Zレジスタリセット	常に「1」	常に「1」	常に「1」
ZレジスタCE	常に「0」	常に「0」	常に「0」
カウントリセット	「0」	「0」	「0」
カウンタCE	Zvld&フラグ	Zvld	Zvld
Y有効	Zvld	Zvld	Zvld
Z有効	Zvld	Zvld	Zvld
フラグ	「0」	「0」	「0」

【0303】

図56は、比較回路を用いたデータステアリング用途を示す詳細な回路およびブロック図である。制御デコードにより、YoutおよびZoutに対して有効なデータを、比較結果によって個別に制御することができる。比較がtrueの場合に一方が有効になり、比較がfalseの場合にもう一方が有効になるようにデータの有効性を設定することで、データの伝搬は1つの出力のみから異なる接続に送られる。さまざまな比較オプションを設定する構成を設定するには、いくつかの方法がある。

ケース1：Zinとインデックス数を比較してYをYoutまたはZoutにステアリングする。

ケース2：Zinを定数と比較し、YをYoutまたはZoutにステアリングする。

ケース3：ZinとYinを比較してYをYoutまたはZoutにステアリングする。

ケース4：Yinを定数と比較し、YをYoutまたはZoutにステアリングする。

ケース5：インデックスを定数と比較して、YをYoutまたはZoutにステアリングする。

ケース6：フラグ入力を使用してYをYoutまたはZoutにステアリングする。

それぞれの場合の設定を表26に示す。データステアリングを通るZパスは、接続されている出力Zが無効であるため注意が必要である。電源を考慮して、Z入力をKzに接続し、レジスタクロックイネーブルをY有効に接続して（表では、有効値はYvおよびZv

と省略されている)、無効な出力が選択解除されたときに伝搬しないようにすることが可能である。カウンタは、リセット入力を使用してリセットすることができ、これは、必要に応じて、データ有効入力で検証することができる。Kyが比較論理内で使用されない場合、Kyを使用して、ストリーミングmin/maxに関して説明したように、カウンタ572を初期化することができる。データステアリングでは、データ有効信号を使用してデータを方向付けるため、非SIMD(32ビットレーン)演算でのみ使用できる。

【0304】

【表26】

表26: データステアリングのための回路320の設定の比較

制御						
関数	ケース1	ケース2	ケース3	ケース4	ケース5	ケース6
	Z ? idx	Z ? Ky	Z ? Y	Kz ? Y	Kz ? idx	フラグ
構成						
SIMD	1	1	1	1	1	1
Zin mux を比較	Zin	Zin	Zin	Kz	Kz	Kz
Yin mux を比較	カウンタ	Ky	Yin	Yin	カウンタ	Ky
データ ステアリング Kz mux	気にしない	気にしない	気にしない	気にしない	気にしない	気にしない
データ ステアリング Ky mux	Yin	Yin	Yin	Yin	Yin	Yin
カウンタ K mux	気にしない	気にしない	気にしない	気にしない	気にしない	気にしない
カウンタ フィードバック mux	カウンタ	気にしない	気にしない	気にしない	気にしない	気にしない
Z出力 mux	カウンタ	気にしない	気にしない	気にしない	気にしない	気にしない
Ky	気にしない	Cmpr 値	Cmpr 値	気にしない	気にしない	気にしない
Kz	気にしない	気にしない	気にしない	Cmpr 値	Cmpr 値	Cmpr 値
Kinc	1	1	1	1	1	1
デコーダ						
データ ステアリング 制御	cmpr	cmpr	cmpr	cmpr	cmpr	フラグ
カウンタ バイパス 制御	追加する	気にしない	気にしない	気にしない	追加する	追加する
Yレジスタ リセット	常に「0」	常に「0」	常に「0」	常に「0」	常に「0」	常に「0」
Yレジスタ CE	常に「1」	常に「1」	常に「1」	常に「1」	常に「1」	常に「1」
Zレジスタ リセット	常に「0」	常に「0」	常に「0」	常に「0」	常に「0」	常に「0」
Zレジスタ CE	常に「1」	常に「1」	常に「1」	常に「1」	常に「1」	常に「1」
カウンタ リセット	Init&dv	常に「1」	常に「1」	常に「1」	Init&dv	常に「1」
カウンタCE	Zv	常に「0」	常に「0」	常に「0」	Zv	常に「0」
Y有効	Zv&Yv&cmpr	Zv&Yv&cmpr	Zv&Yv&cmpr	Zv&Yv&cmpr	Zv&Yv&cmpr	Yv&フラグ
Z有効	Zv&Yv&~cmpr	Zv&Yv&~cmpr	Zv&Yv&~cmpr	Zv&Yv&~cmpr	Zv&Yv&~cmpr	Yv&~フラグ
フラグ	「0」	「0」	「0」	「0」	「0」	「0」

【0305】

図57は、比較回路320を用いたモジュロN計数アプリケーションを示す詳細な回路

およびブロック図である。比較ブロックのインデックスカウンタ572は、サンプルインデックスのサンプルをカウントするように設計されている。アドレスカウントにも使用できる。カウンタの増分は増分定数によって設定され、任意の32ビット値に設定できる。レジスタ入力は、カウント加算器または入力のバイパスを選択するセレクトからのものであり、カウントをゼロ以外の値に初期化する手段を提供する。カウンタレジスタと桁上げは、2つおよび4つのSIMDレーンを処理するために8ビットセグメントに分割される。比較回路320は、ローテータおよびブール論理の前に配置され、コーナータンおよびFFTビット反転およびコーナータンアドレッシングを含むより複雑なシーケンスのアドレスカウントをこれらのブロックが操作できるようにする。カウント論理のさまざまな使用ケースのセットアップを表27に示す。

10

【0306】

基本的な用途は、Kinc定数レジスタによって設定された増分値を持つ単純なカウンタ（線形カウント）である。単純カウントの場合、Kincレジスタは1に設定される。これを任意の32ビット値に設定して、増分を変更することもできる。カウンタのクロックイネーブルは、カウントceのデコード条件が満たされたときにカウントをインクリメントする。CEは、有効なデータ、比較条件、フラグ入力、またはそれらの組み合わせでインクリメントするために使用できる。単純に使用すると、レジスタリセットを使用してカウンタを0にクリアする。レジスタリセットは、プログラム可能な有効、フラグ、およびリセット入力と比較結果の論理関数である。カウンタを別の値に初期化する場合がある場合、リセット条件がデコードされ、カウンタCEがアサートされると、カウンタバイパスmuxを切り替えて、カウンタのKmux（KzまたはYinput）によって選択された値でカウンタレジスタをロードする。バイパスmuxおよびKzを使用したリセットは、Kzの比較器またはステアリングmuxの使用に干渉する可能性があり、その場合、代わりにYinによって定数が供給され得ることに留意されたい。比較およびデータステアリングは、線形カウントには使用されない。比較ブロックのこれらの部分は、使用されるカウント論理に干渉しない比較アプリケーションに使用できる。

20

【0307】

比較回路320を使用してカウントを制限し、制限に達するとカウントを凍結することができる。カウントを制限するには、比較をカウント制限に設定し、比較結果によってクロックイネーブルがゲートされる。これにより、カウントが端末カウントに達すると、リセットされるまでカウントの増分がディセーブルになる。制限カウンタ接続は、モジュロカウントの場合と同様に、カウンタCEがリセットではなく制限カウント用にゲートされる点を除いて同じである。フラグ出力は、比較結果によって駆動されて、制限カウンタの外部指示を提供することができる。比較結果を使用して、未使用の入力および定数からデータフローを選択またはゲートすることもできる（制限値にはKzが使用される）。カウント制限は、定数KzではなくZinputで指定されている場合は変数にすることもできる。

30

【0308】

比較回路320を使用して、端子カウントに達したときに次のクロック可能クロックのカウントを同期的にリセットすることもでき、その結果、端子カウントがN-1に設定され、カウンタのレジスタリセットを使用して比較結果によってリセットが行われる場合には、モジュロNカウントとなる。図57に、modulo-Nカウンタの接続を示す。N-1がKz定数ではなくZinputを介して適用される場合、モジュロは変数になる。

40

【0309】

50

【表 27】

表27: 比較回路320のカウント設定

制御					
ファンクション	単純カウント	制限カウント	モジュロカウント	非ゼロリセット	コーナーターン
構成					
SIMD	1	1	1	1	1
Zin mux を比較	気にしない	ZinまたはKz(N-1)	ZinまたはKz(N-1)	気にしない	Kz
Yin mux を比較	気にしない	カウンタ	カウンタ	気にしない	カウンタ
データステアリングKz mux	気にしない	気にしない	気にしない	気にしない	気にしない
データステアリングKy mux	気にしない	気にしない	気にしない	気にしない	気にしない
カウンタK mux	気にしない	気にしない	気にしない	気にしない	気にしない
カウンタフィードバックmux	カウンタ	気にしない	気にしない	カウンタ	気にしない
Z出力 mux	カウンタ	気にしない	気にしない	カウンタ	気にしない
Ky	気にしない	気にしない	気にしない	気にしない	気にしない
Kz	気にしない	N-1	N-1	リセット値	$(N-1) \cdot (2^{\text{bits}+1})$
Kinc	1	1	1	1	$2^{\text{bits}+1}$
デコーダ					
データステアリング制御	気にしない	気にしない	気にしない	気にしない	気にしない
カウンタバイパス制御	追加する	気にしない	気にしない	リセット	気にしない
Yレジスタリセット	気にしない	気にしない	気にしない	気にしない	気にしない
YレジスタCE	気にしない	気にしない	気にしない	気にしない	気にしない
Zレジスタリセット	リセット*	リセット*	Cmpr&vld	「0」	Cmpr&vld
ZレジスタCE	vld	Cmpr&vld	vld	vld	vld
カウンタリセット	リセット*	Cmpr&vld	Cmpr&vld	「0」	Cmpr&vld
カウンタCE	*	Cmpr&vld	vld	*	vld
Y有効	*	気にしない	気にしない	*	気にしない
Z有効	*	vld	vld	*	vld
フラグ	「0」	Cmpr*	Cmpr*	「0」	Cmpr*

【0310】

図58は、コーナーターンアドレスの導出を示すブロック図である。コーナーで曲げられたアドレッシングは、メモリに格納されている行列のアドレスを列メジャー順序でアクセスするために行メジャーで並べ替える。2次元の電力を持つ配列の場合、 $\log_2$ 副軸次元（回転フィールドがアドレスと同じサイズである場合）によって左のアドレスビットを回転させることによって、コーナー方向に向いたアドレッシングを生成できる。これは、行に対応するビットフィールドの位置を列に対応するビットフィールドの位置と入れ替えるのと同じである。8ビットおよび16ビットの場合、Z入力シフト回転を直接使用して回転アドレスを取得できる。しかし、一般的なケースでは、回転を生成するためのシフト距離が8ビット、16ビット、32ビット、または64ビットに固定されているため、組み込みの回転は不十分である。ローテーションを行うには、アドレスカウンタのコピーを実際のカウンタのすぐ隣に配置する必要がある。アドレスカウンタは、アドレス幅をオーバーフローさせない限り、隣接するビットフィールドに同じカウンタを2つ生成するように変更できる。モジュロカウントまたは制限カウントの設定は、オーバーフローが発生しないことを保証するために使用できる。2つの隣接カウンタは、増分を各フィールドのlsbに「1」を有するように変更することによって生成することができ、したがって、 $9 = \text{ビット} \cdot \text{アドレス} \cdot \text{カウンタ}$ の場合、カウンタ増分は、図58の上部に示すように、 0×0201 になる。これにより、2つの隣接カウンタが生成される。1つはビット0のlsbとビット8のmsb、もう1つはビット9のlsb（図58に「Zシフトに入力されるインデックスカウンタ」として示される）です。下のカウンタが上のカウンタのlsbにオーバーフローするのを防ぐために、カウンタがすべての「1」を超えないようにする必要があります。モジュロカウントの設定を使用し、最大数を保持する2つのカウンタフィールドと等しい制限を設定すると、適切なモジュロカウントになる。次に、この結合されたカウンタの対は、Zシフト論理によってシフトされて、シフトウィンドウの最下位ビット（32ビット入力のシフトウィンドウはビット63:32）にビットの正しい回転を入

れ、次に、シフトウィンドウの外側のビットを廃棄するために、ブール論理でその結果をマスクする。ブールマスクを使用して、メモリ内のページを選択するための上位ビットを設定することができる。

【0311】

図59は、FFTビット逆コーナーターンアドレスの導出を示すブロック図である。この設計は、比較ブロックと、高速フーリエ変換をサポートするビット反転アドレッシングのビット順序を反転するために使用されるZシフト論理との間の選択可能な有線32ビット反転を含む。高速フーリエ変換は、混合基数アルゴリズムを使用して結合される基数2と基数4の段階で構成される。各段階の間にコーナーターンアドレスがあり、段階の入力と出力はビット反転アドレス指定される（lsbはmsbになり、逆も同様である）。データが自然な順序でメモリに格納される場合、読み取りおよび書き込みアドレス指定は、各段階でビット反転アドレスの1つのコーナーターンである（ただし、各段階で異なるサイズ）。アドレス指定は、ビット反転時のビットの再配置を考慮して、上記のコーナーターンアドレス指定を変更したものである。カウント論理のセットアップはコーナー回転セットアップと同一であるが、32ビットカウント出力は、Zシフタの前に最初にビット反転され、これにより、関連ビットがワードの最上位端に置かれる。これを考慮してシフト距離が調整され、残りのステップはコーナー回転の場合と同じになる。図59は、導出のためのビットアライメントを示す。

【0312】

11. 入力並べ替えキュー350および出力並べ替えキュー355

図60は、RAE入力並べ替えキュー350およびRAE出力並べ替えキュー355の論理回路構造を示す詳細な回路およびブロック図である。これらの複数のRAE入力並べ替えキュー350は、次の3つの主な目的を果たす。（1）それらは、時間的にデータの再配列を可能にし、X入力とZ入力との間でデータを再分配する。（2）それらは、パイプラインレイテンシーバランシングを補助するための調整可能な遅延を提供し、（3）それらは、RAE入力365、370、375に適用される多数の定数を記憶し、配列することを可能にする。並べ替え機能を使用すると、I/Q対として表示する必要がある複素データを含む特定のアルゴリズムのデータ処理が大幅に簡略化されますが、処理効率を高めるために、一度に2つのサンプルをIとQでインターリーブ処理する必要がある。この例は、バタフライと呼ばれる高速フーリエ変換（FFT）要素演算である。バタフライの2つの標本は、ほとんど同じ処理（符号の変化だけが異なる）に従うので、効率のために、両方の標本のI成分は、両方の標本のQ成分と交互に処理される。その他の複素演算アルゴリズムでも、IおよびQコンポーネントをインターリーブし、同時に2つのサンプルを処理することでメリットが得られる。このアプリケーションでは、深さが4サンプルの並べ替えキューを使用する。

【0313】

同じRAE回路クアッド450の二つのRAE回路300によって共有され得るRAE出力並べ替えキュー355は、図60に示されるように、Yデータ経路を有さないことを除いて、RAE入力並べ替えキュー350と同一の構造を有し、また、条件付き多重論理を必要としない。RAE出力並べ替えキュー355への入力、RAE回路クアッド450の同じ半分にある二つのRAE300のRAE300X出力420（例えば、アキュムレータ315出力）から来る。出力並べ替えキュー355は、二つのRAE300間の交換を可能にして、FFTのためにI及びQを再インターリーブし、半複素乗算を行い、また、サンプルシーケンスを直近の四つのサンプルまで短距離で並べ替える。これらのRAE出力並べ替えキュー355回路をRAE300に含めることにより、IQまたは偶数奇数インターリーブおよび同様の演算（例えば、高速フーリエ変換）を必要とするアルゴリズムのアルゴリズム設計およびレイアウトが簡素化される。代表的な実施形態では、RAE300の対ごとに、RAE出力並べ替えキュー355の一組が存在する。

【0314】

調整可能なパイプライン遅延関数は、並べ替えキューによって提供される関数のサブセ

ットである。1つまたは複数のの定数入力を乗数に必要とするアプリケーションがある。並べ替えキューレジスタ580は、並べ替えキューシーケンサを使用してシーケンスすることができる定数値を保持するように再利用することができる。定負荷機構は、再配列待ち行列内の32ビットレジスタをデイジーチェーン（個別には図示せず）にリンクして、これらの定数がクロック当たり32ビット入力され、12クロックの終わりに再配列待ち行列が12個の定数値で満たされるようにチェーンを下方に伝搬する。データの有効期限がゲートされている限り、値は動作中にレジスタに残る。チェーン内の最後のレジスタの出力は、比較ブロック内のレジスタおよびブール論理回路325内のブール関数テーブルを含む、RAE300内の他の定数レジスタのチェーン内でリンクされ、定数レジスタのチェーン全体の順次ロードを可能にする。明確にするために、定負荷チェーンおよび書き込み論理は、図60には示されていない。荷重データはY入力を介して入力される。最後のYレジスタは、Xinにおけるmux585を介してXチェーンに接続され、Xチェーンの出力は、別の32ビットmux590を介してZチェーンに接続される。

【0315】

並べ替えキューは、RAE300入力365、370、375に対応する3つの32ビットX、Y及びZデータ入力を有する。X、YおよびZ入力365、370、375の各々は、それぞれ有効なデータ595、596、597に関連付けられる。データは、同じ（対応する）入力に対して有効なデータが「1」の場合にのみ有効と見なされる。データは、対応する有効データが「1」であるとき、クロックの各立ち上がりエッジで、RAE入力並べ替えキュー350にシフトされる。有効データが「0」の場合、対応する入力のデータは並べ替えキューレジスタ580に転送されない。複数のデータ有効595、596、597は、RAE入力並べ替えキュー350へのデータのシフトを可能にする。

【0316】

RAE入力並べ替えキュー350は、それぞれ3つの32ビットX、Y及びZデータ出力582、584、586を有する。出力のデータは、現在アドレス指定されているシーケンサメモリの状態に応じて、ディレイキューレジスタまたは対応する入力のいずれかから選択される。データ出力には、データの有効性を示すデータ有効出力フラグが付随する。有効なデータの出力は、有効なデータの遅延バージョンであり、並べ替えによる意図された並び替えられたデータ遅延に対応する最大4クロックのプログラム可能な遅延がある。これは、特定の数のサンプルを収集し、ステートマシンを使用してグループで出力する必要がある場合がある。出力データ有効およびシーケンスカウンタは、入力サンプルと同期させる必要がある。キューが定数を保持している場合でも、有効なoutデータは存在しますが、構成ビットを使用してオフにすることができる。リセットフラグは、シーケンスカウンタ575を00状態にリセットする。シーケンス・カウントは、シーケンス命令または条件フラグ・論理（個別には図示せず）で使えるように、出力インターフェースで提供される。

【0317】

RAE入力並べ替えキュー350からのデータ出力選択は、シーケンスカウンタ575によってアドレス指定される4つのレジスタ602の内容によって決定される。これらのレジスタ602のそれぞれは、XおよびZ出力用に選択する3ビットマルチプレクサ604、608と、Y出力用に選択する2ビットマルチプレクサ606と、X、Y、およびZバイパスセクタ用にそれぞれ1ビットと、シーケンサ用に割り当てられる2ビットの次の状態とを含む。シーケンサレジスタのためのプログラミングのソースは、定数ロード機構の一部としてロードすることができ、その場合、それは、二つのシーケンサ状態のための13ビットをそれぞれ含む2つの32ビットワードを構成する。42ビットの順次ロードチェーンを保持するために、各作業における6つの未使用ビットもレジスタを有し、予備ビットは、RAE300の他の場所でシーケンサ出力として使用するためにシーケンサの出力セクタを介してブロックピンに持ち出され得る。追加の構成には、各データ出力に対する条件付き多重化選択、およびおそらく有効な出力データ有効化、および有効なデータとフラグ再生成の制御が含まれる。

【0318】

R A E X、Y、およびZ入力は、全ての3つの入力上で独立した短距離データの並べ替え、およびフーリエ変換、複素乗算、I/Qインターリーブおよびデインターリーブ、ならびに同様の操作のためのシーケンス修正をサポートするためのXおよびZ入力間のシーケンス交換を可能にするように設計された、小さな4つのサンプル並べ替えキュー610（レジスタ580を用いて図示）を含む。また、並べ替えキュー610内のレジスタ580に定数をロードし、次に保持して、X、Y、Z R A E入力365、370、375のそれぞれに対する最大4つの入力定数の循環を可能にすることができ、これは、定数（フィルタ、相関器などに使用されます。）を有するドット積に有用である。入力並べ替え論理は、Yが負でない場合にX入力がX_{in}であり、負である場合に定数レジスタの1つに格納される定数である条件付き被乗数操作を容易にするために、Yの符号が定数レジスタを選択またはバイパスするためのパスを含む。XおよびZ出力のための出力セクタは、それぞれ8:1セクタ605、590であり、その後2:1バイパス選択（それぞれ604、608マス）が続く。各8:1セクタは、同じ入力の4つの遅延サンプルまたは反対側の入力の4つの遅延サンプルから選択する。XおよびY選択は、シーケンスカウンタ575によって選択された4つの構成レジスタのうちの1つからの2つの4ビット値によって制御される。Y入力は、どのキュータップが各シーケンサ状態の出力に向けられるかを選択するための4:1マルチプレクサ607を備えた同様の4ディープシフトレジスタキューを有する。これには、シーケンスカウンタ575によって制御される2:1キューバイパスマルチプレクス606も続く。

【0319】

図61は、入出力並べ替えキュー350、355に対するシーケンサ575の論理回路構造を示す詳細な回路およびブロック図である。シーケンスカウンタ575は、次の状態が構成状態レジスタ内の2つのビットによって決定される非分岐ステートマシンである。各コンフィギュレーションレジスタは16ビットで、32ビットのコンフィギュレーションワード書き込みごとに2つ書き込まれる。コンフィギュレーションビットのうち2つは、レジスタの次の状態である。選択された構成ビットのうち11ビットが、図60に示すように、データ経路内のデータ選択マルチプレクサを駆動する。残りの3ビットは現在使用されていないが、構成データをチェーンの下に順番に伝播できるようにするためにレジスタは残っている。

【0320】

12. データパケットとルーティング制御

図68は、再構成可能プロセッサと共に使用される代表的なデータパケット850の図である。代表的な実施形態では、データパケット850は、計算コア20036ビットペイロード852および制御ビット853を含む37ビットである。ペイロード852は、32ビットのデータペイロード854と、4ビットのサフィックス856としてのサフィックスビットとを含む。32ビットデータ854は、図69に示す整数、浮動小数点、およびSIMD表現を含む様々な潜在的なデータフォーマットをサポートする。上述したように、サフィックス856は、アレイによるゼロ圧縮のために主に使用されるが、種々のフラグ、条件付き演算、演算インデックスの搬送などの他の機能をサポートすることもできる。制御（データタイプ）ビット853は、相互接続120によってのみ使用され、計算コア200内では搬送されない。

【0321】

図69は、データパケット850で利用される代表的なデータペイロード854タイプの図であり、IEEE754倍精度浮動小数点（FP64）用の64ビット浮動小数点（2つのパケット858内）、IEEE754単精度浮動小数点用の32ビット浮動小数点（FP32）（862）、IEEE754の半精度浮動小数点（FP16）の16ビット浮動小数点（864）、B F L O A T 16用16ビットブレイン浮動小数点（B F 16）（866）、IEEE754の1/4精度浮動小数点（FP8）（868）の場合は8ビット浮動小数点、符号付きおよび符号なしの32ビット整数値（I n t 32）（872）

の場合は32ビット整数、符号付きおよび符号なしの16ビット整数値（Int16）（874）の場合は16ビット整数、符号付きおよび符号なしの8ビット整数値（Int8）（876）の場合は8ビット整数を含むが、これに限定されない。

【0322】

図70は、計算コア200に出入りするデータのためにそれぞれ入力マルチプレクサ205および出力マルチプレクサ110と共に使用されるルーティングコントローラ820のブロックおよび回路図であり、オプションとして、プログラムシーケンサ825を含むこともできる。ルーティングコントローラ820は、動的出力選択832または静的またはプログラムされた出力選択834のいずれかの出力選択マルチプレクサ830から制御出力836を選択する動的出力選択によって制御される出力選択マルチプレクサ830を含む。次に、制御出力836を利用して、対応する入力マルチプレクサ205または出力マルチプレクサ110に利用可能な入力から、対応する入力マルチプレクサ205または出力マルチプレクサ110の出力を（相互接続ネットワーク120、220から）選択する。

10

【0323】

上述のように、再構成可能プロセッサ100は、データフローを利用する。その一部として、データ生成者は、送信すべきデータがあることを示すデータ送信要求信号（「REQ」）をアサートし（842）、データ消費者は、データを受け入れる余地があることを示すデータ送信許可信号（「GNT」）をアサートする（844）。データ転送コーディネータ回路840は、データ送信要求信号（REQ）を受信し、動的出力選択852によって制御され、データ送信要求信号（REQ）をそれぞれ対応する入力または出力レジスタ230、242に渡す第1のデータ転送マルチプレクサ846を含む、このようなデータ送信許可信号（GNT）を送信するために利用される。第2のデータ転送マルチプレクサ848は、データ送信許可信号（GNT）を受信し、動的出力選択852によって制御されて、対応する入力または出力レジスタ230、242からのデータ送信許可信号（GNT）をそれぞれ要求データ送信機に戻す。共有プログラムメモリ856（出力プログラム272、274、276のような）にアクセス可能なプログラム854の制御下で、第1および第2のデータ転送マルチプレクサ846、848にそれぞれ入力を提供する任意のプログラムシーケンサ825を含むことができる。

20

【0324】

また、この要求および許可メカニズムは、データパケットが複数の宛先に送信されるときにデータ順序を維持するなど、さまざまな状況下でデータフローを制御するためにも利用される。たとえば、データが複数の場所に分岐される場合、データ送信機は、データを送信する前に、各データ受信機からデータ送信許可信号（GNT）を受信する必要がある。また、例えば、データが複数のソースから単一の宛先にマージされる場合、データ受信装置は、複数のデータ送信要求信号（REQ）を受信し、各データ送信装置に向けて結合されたデータ送信許可信号（GNT）を発行すべきであり、各データ受信装置は、データを送信する前に、データ送信許可信号（GNT）を受信すべきである。また、例えば、複数の送信元から1つの送信先にデータを切り替える場合、データ受信装置は、複数のデータ送信要求信号（REQ）を受信し、各個別データ送信装置に対して個別データ送信許可信号（GNT）を発行し、各データ受信装置は、データを送信する前に、対応するデータ送信許可信号（GNT）を受信する必要がある。また、例えば、データが選択可能な位置にステアされる場合、データ送信機は、データを送信する前に、選択されたデータ受信機からデータ送信許可信号（GNT）を受信すべきである。

30

40

【0325】

13. サフィックス制御回路およびゼロ圧縮／圧縮解除

図71は、サフィックス制御回路390の代表的な実施形態のブロック図である。図72は、ゼロ圧縮回路800のブロック及び回路図である。図73は、代表的なゼロ圧縮データパケット列の図である。図74は、ゼロ圧縮解除回路805のブロック及び回路図である。

50

【0326】

図71-74を参照すると、代表的な実施形態では、サフィックス制御回路390は、ゼロ圧縮を実行するゼロ圧縮回路800と、ゼロ圧縮解除を実行するゼロ圧縮解除回路805と、任意選択的に、条件論理、分岐、条件フラグ処理および生成など、ユーザが決定することができるサフィックス856に関する他の活動を実行するために、制御論理およびステートマシン回路810を含む。別の代表的な実施形態では、例えば、ゼロ圧縮回路800およびゼロ圧縮解除回路805が計算コア200の他の部分（図4に示すような）で実施される場合、サフィックス制御回路390は、典型的には、制御論理回路およびステートマシン回路810を含む。ゼロ圧縮回路800は、ゼロカウンタ802と、パケット生成部804とを備える。ゼロ圧縮を有効にすると、ゼロ圧縮によって有効な0の値が伝播されなくなる。代わりに、各有効なゼロ値は、ゼロカウンタ802を使用してゼロカウントで追跡される。値が0のストリームの間、ゼロカウントは最大値の15に達するまで増加する。この場合、次の有効なデータ転送は、値に関係なく、パケット生成器804（すなわち元のデータパケットを伝播するが、そのサフィックスビットをカウント0に置き換える）からのデータパケットとしてDATA__OUTに伝播され、ゼロカウントはサフィックス856に含まれ、ゼロカウントはロールオーバーされるか、ゼロにリセットされる。ゼロ以外の転送の文字列では、データは0のサフィックス856とともに正常に伝播する。これは、前に0の値がなかったことを示す。

10

【0327】

図73は、ゼロ圧縮がイネーブルされたデータシーケンスの例を示す。最初の2つの転送806は正常に伝搬する。値91および92はデータパケット（814）としてDATA__OUTに送信され、サフィックス856のビットはゼロに設定され、ゼロの前にゼロがあることを示す。入力ランザクション3から5（808）は伝播されないが、有効な入力ランザクションごとにゼロカウントが増加する。入力ランザクション6（812）では、ゼロ以外の値93が到着し、サフィックス856を介して累積ゼロカウント3でDATA__OUT（816）に伝播される。つまり、値93の前には30個の値があった。ゼロカウントは、ゼロ以外の値でもリセットされる。図示の例では、24の入力転送がゼロ圧縮回路に入り、5つの出力転送のみが生じ、情報の損失がないという利点がある。

20

【0328】

図74を参照すると、ゼロ解凍回路805は、サフィックスカウンタ818とパケット生成器804とを含む。データペイロード854を有し、サフィックス856にゼロでないカウントを有するデータパケットが到着すると、サフィックスカウンタ818はサフィックス856のゼロカウントを決定し、パケットに到着した実際のデータペイロード（そのサフィックス856にゼロでない値を有する）を送信する前に、ペイロード854にゼロを有するその数のデータパケットを（DATA__OUT 819上に）発行するようにパケット・ジェネレータ804に信号を送る。

30

【0329】

上述のように、サフィックス制御回路390内に配置される代わりに、ゼロ圧縮回路800およびゼロ圧縮解除回路805は、入力マルチプレクサ205からデータを受信し、任意のゼロ圧縮を解除するためのゼロ圧縮解除回路805を含むように、また、出力マルチプレクサ110の前にゼロ圧縮回路を含むように、様々な相互接続ネットワーク120、220上の出力データパケットの選択および送信の前にゼロ圧縮を実行するためのゼロ圧縮回路800を含むように、計算コア200全体に分散されてもよい。

40

【0330】

14. 代表的なアプリケーション

RAE回路300は、例えば、2つの乗算器305（乗算のため）および2つの乗算器シフトコンバイナネットワーク310（加算のため）を使用し、メモリ150を使用して、補間LUT（ルックアップテーブル）を生成するために使用することができる。メモリから直接係数を得るブルートフォース法は、メモリ利用の観点から禁止される。幸いなことに、一連の係数を表す関数は滑らかな関数（ほぼsinc関数）であるため、リアルタ

50

イムでの係数の圧縮と生成は、リソースの使用率の点で魅力的である。係数セットは、近似的に、 $1/(P \times F)$ の間隔でサンプリングされた sinc 関数であり、ここで、 $P \times F$ はフィルタの長さであり、 P は多相分岐長であり、 F は FFT サイズである。係数は多相分岐に渡って分配され、1つの分岐上で連続するタップが $C(k)$ 、 $C(k+F)$ 、 $C(k+2F)$ 、... に関連付けられる。特定の乗算器 305 に与えられる係数は連続的な係数であり、したがって、乗算器 m において、係数は $C(T \times F = m)$ であり、ここで T はタップ数であり、 F は FFT 長である。これは、任意の乗算器 305 における係数が sinc 関数の連続セグメントであることを意味する。これにより、補間方式を使用して、係数を格納するためのメモリ要件を減らすことができる。

【0331】

設計に用いた補間法は、係数発生器当たり単一ブロック RAM で実装した小メモリ (512×72) に記憶した 2 次係数から生成した 2 次スプラインである。係数を記憶する代わりに、係数インデックスの最上位ビットによって表される近傍に適合した曲線に対する 2 次係数を記憶する。次に、インデックスの最下位ビットは、最上位インデックスビットによって示される係数位置からのオフセットとして 2 次式に適用される。上位 9 ビットは、曲線の 3 つの 2 次係数を含む 512×72 ビットのメモリをアドレス指定し、(32768 ポイントについての) 下位 6 ビットを使用して、補間値 $y = Ax^2 + Bx + C$ が計算される。メモリの内容は、スケーリングされた A 、 B 、および C 係数であり、これは、 MathLabPolyfit 関数を使用して、フィルタのインパルス応答のセグメントに 2 次式を適合させることで見つけることができる。

【0332】

図 62 は、混合基数アルゴリズムを用いた FFT の組み合わせを示す。図 63 は、基数 2 の FFT カーネルを実行するための RAE300 対を示す詳細な回路およびブロック図である。図 64 は、完全な回転子のために構成された RAE300 対を示す詳細な回路およびブロック図である。図 65 は、基数 4 の FFT (バタフライ) カーネルを実行するための RAE 回路クアッド 450 を示す詳細な回路およびブロック図である。図 66 は、 FFT カーネルおよび他のアプリケーションを実行するための複数の RAE300 カスケード対を示す詳細な回路およびブロック図である。図 62 を参照すると、行の数は第 1 の変換のサイズと等しく、列の数は第 2 の変換のサイズであり、行 $\times$ 列は複合変換のサイズである。4 つの (基数 - 4) 複素 FFT カーネル + 双回転子の半分は、図 63、64、および 65 に示すように、単一の RAE 回路クワッド 450 内に構築することができる。 RAE 回路クアッド 450 は、クロックごとに 1 つの FFT ポイントを処理する (たとえば、4 ポイント変換は 4 クロックで完了し、クロックごとに 1 つのサンプルを受け入れて出力する)。この実装は、 FFT カーネルをカスケードしてより大きな FFT を構築するとき位相回転器として使用される入力に複素乗算を含む。カスケード戦略は、より長い変換長に対してより小さい FFT を結合するための混合基数アルゴリズムに従う。複数の RAE300 の各対は、専用の対ルーティングを使用して、偶数ポイントと奇数ポイントの両方の虚数部を同時に実行しながら、インターリーブされた一方のクロック内の偶数ポイントと奇数ポイントの両方の実部を同時に実行する。 $I = I_e + jI_o$ 。基数 4 の変換はビルディングブロックであり、例えば図 66 に示すように、さらにカスケードされてもよい。

【0333】

関連付けられたメモリを使用して、最大 256 ポイントの変換長の IQ データの 2 ページ、および最大 512 ポイントの変換長のカーネル入力の回転子のコサインとサインのツイストを格納できる。基数 4 のカーネルは、より大きな変換長のための構成要素である。より大きなフーリエ変換は、「混合基数」アルゴリズムを使用して、任意のサイズの小さな変換「カーネル」から構築することができる。このアルゴリズムは、本質的に、データを $k \times n$ 行列に入力する。ここで、 k および n は、 kn 点変換に対する構成要素変換のサイズである。最初にデータが行に沿って入力され、次に最初の変換が列に適用されます。中間結果要素は、行列内のインデックスに従ってフェーズ回転され、変換の 2 番目のセットが行列の各行に適用され、最後にデータが列方向に読み取られたときに出力が自然に順

序付けられる。このシーケンスを図 6 2 に示す。

【0334】

混合基数アルゴリズムを繰り返し適用して、図 6 5 に示すような漸進的により大きな変換を構築することができる。たとえば、1 K ポイント変換は、混合基数アルゴリズムを使用して 4 ポイントカーネルからそれぞれ 2 つの 1 6 ポイント変換を作成し、1 つの 1 6 ポイントと残りの 4 ポイントを組み合わせて 6 4 ポイント変換を作成し、その 6 4 ポイントと残りの 1 6 ポイントを組み合わせて 1 0 2 4 ポイント変換を作成することによって構築される。同様に、実装は、混合基数アルゴリズムを使用して 8 点および 1 6 点の変換から構築され、最初に 1 2 8 点の変換を生成し、次にアルゴリズムの別の適用によって 1 2 8 点および 1 6 点から 2 k の変換を生成することができる。8 ポイントと 1 6 ポイントの変換は、混合基数アルゴリズムを使用して 2 ポイントと 4 ポイントの変換から作成することもできる。入力と出力はビット反転順であってもよい。

10

【0335】

図 6 7 は、限定されるものではないが、例えば、1 つ以上の R A E 回路クアッド 4 5 0 を含む 1 つ以上の R A E 回路 3 0 0 の比較器 3 2 0 およびブール論理回路 3 2 5 を使用して実装することもできる、ストリングマッチングのユースケースを示す図である。文字列ごとに 1 つの S I M D 4 レーンを使用し、S I M D レーンごとに 1 つの入力を使用する。入力ストリームを 4 つのレーンで複製して、4 つの異なるストリングを同時に検索できる。入力遅延キューを使用して、A N D 1 クロックによる遅延を比較による遅延よりも短くすることができる。

20

【0336】

15. 結論

【0337】

再構成可能プロセッサ 1 0 0 は、人工知能、ニューラルネットワーク計算、デジタル通貨、暗号化、復号化、ブロックチェーン、高速フーリエ変換 (F F T) の計算、および機械学習を含むような数学的に集中的なアプリケーションのための高性能およびエネルギー効率の良いソリューションを提供するが、これらに限定されるものではない。

【0338】

さらに、再構成可能プロセッサ 1 0 0 は、これらの様々なアプリケーションのいずれに対しても構成することができ、いくつかのそのような例を以下により詳細に説明する。このような再構成可能プロセッサ 1 0 0 は、数百万の計算コア 2 0 0 などに容易に拡張可能であり、低レイテンシを有し、計算的にエネルギー効率がよく、リアルタイムでストリーミングデータを処理することができ、選択されたアプリケーションのために計算ハードウェアを最適化するように再構成可能であり、大規模並列処理することができる。例えば、単一のチップ上で、複数の再構成可能プロセッサ 1 0 0 を配列し、相互接続ネットワーク 1 2 0 を用いて接続して、チップ当たり数百から数千の計算コア 2 0 0 を提供することもできる。次に、このようなチップを複数配列して回路基板上に接続し、基板当たり数千から数百万の計算コア 2 0 0 を得ることができる。任意の選択された数の計算コア 2 0 0 を再構成可能プロセッサ 1 0 0 内に実装することができ、任意の数の再構成可能プロセッサ 1 0 0 を単一の集積回路上に実装することができ、任意の数のそのような集積回路を回路基板上に実装することができる。このように、計算コア 2 0 0 のアレイを有する再構成可能プロセッサ 1 0 0 は、任意の選択された程度にスケラブルである（ただし、経路および熱放散などの他の制約の対象となるが、これに限定されるものではない）。

30

40

【0339】

16. 一般事項

プロセッサ回路 1 3 0 は、任意のタイプのプロセッサとすることができ、本明細書で説明する機能を実行するように構成され、設計され、プログラムされ、または他の方法で適合された、一つ以上の R I S C - V または他のプロセッサとして具体化することができる。ここでプロセッサ回路 1 3 0 という用語が使用されているように、プロセッサ回路 1 3 0 は、単一の集積回路 (「I C」) の使用を含んでもよく、または、複数の集積回路また

50

は一緒に接続され、配列され、またはグループ化された他の構成要素、例えば、コントローラ、マイクロプロセッサ、デジタル信号プロセッサ、並列プロセッサ、マルチコアプロセッサ、カスタムIC、特定用途向け集積回路、フィールドプログラマブルゲートアレイ（「FPGA」）、適応コンピューティングIC、関連メモリ（RAM、DRAM、ROMなど）、およびアナログかデジタルかを問わない他のICおよび構成要素の使用を含んでもよい。結果として、本明細書で使用される場合、プロセッサ回路130という用語は、マイクロプロセッサメモリまたは追加のRAM、DRAM、SDRAM、SRAM、MRAM、ROM、FLASH、EPROMまたはE²PROMのような関連するメモリと共に、単一のIC、またはカスタムIC、ASIC、プロセッサ、マイクロプロセッサ、マイクロプロセッサ、コントローラ、FPGA、適応コンピューティングICの配置、または以下で説明する機能を実行する集積回路の他のグループを同等に意味し、含むものと理解されるべきである。プロセッサ回路130は、関連するメモリと共に、上述したように、本発明の方法論を実行するように適合または構成されてもよい（プログラミング、FPGA相互接続、またはハード配線による）。例えば、方法論は、プロセッサ回路130が動作可能である（すなわち電源が入っており、機能している）場合に、その後の実行のために、一組のプログラム命令または他のコード（または同等の構成または他のプログラム）として、関連するメモリ（および／またはメモリ）および他の同等の構成要素とともに、プロセッサ回路130内にプログラムされ、格納され得る。同様に、プロセッサ回路130が全体的にまたは部分的にFPGA、カスタムICおよび／またはASICとして実装され得る場合、FPGA、カスタムIC又はASICもまた、本発明の方法論を実装するように設計され、構成され、および／または配線され得る。例えば、プロセッサ回路130は、「コントローラ」と総称されるアナログおよび／またはデジタル回路、コントローラ、マイクロプロセッサ、DSP、および／またはASICの配列として実装することができ、これらはそれぞれ、ハードワイヤード、プログラム、設計、適応または構成されて、場合によってはメモリと組み合わせて、本発明の方法を実装する。

【0340】

データリポジトリ（またはデータベース）を含むことができるメモリ150、155は、任意のコンピュータまたは他の機械可読データ記憶媒体、情報の記憶または通信のためのメモリ装置または他の記憶または通信装置であって、現在知られているもの、または将来利用可能になるものを含むがこれらに限定されない、任意の数の形態で具体化することができ、メモリ集積回路（「IC」）または集積回路のメモリ部分（プロセッサ内の常駐メモリなど）、揮発性であるか不揮発性であるか、リムーバブルであるか非リムーバブルであるかを問わず、RAM、FLASH、DRAM、SDRAM、SRAM、MRAM、FeRAM、ROM、EPROMまたはE²PROMを含むがこれらに限定されない、または磁気ハードドライブ、光学ドライブ、磁気ディスクまたはテープドライブ、ハードディスクドライブ、フロッピーディスク、CDROM、CD-RWなどの他の機械可読記憶またはメモリ媒体などの任意の他の形態のメモリ装置を含む、デジタル多用途ディスク（DVD）または他の光学メモリ、または他のタイプのメモリ、記憶媒体、またはデータ記憶装置または回路であって、選択された実施形態に応じて既知または既知になるものを含んでもよい。メモリ150、155は、種々のルックアップテーブル、パラメータ、係数、他の情報およびデータ、（本発明のソフトウェアの）プログラムまたは命令、ならびにデータベーステーブルなどの他のタイプのテーブルを記憶するように適合されてもよい。

【0341】

上述のように、プロセッサ回路130は、例えば、本発明のソフトウェアおよびデータ構造を使用して、本発明の方法を実行するために、ハードワイヤードまたはプログラムされる。結果として、本発明のシステムおよび方法は、上述したような一時的でないコンピュータ可読媒体内に具体化された命令および／またはメタデータのセットのようなプログラミングまたは他の命令を提供するソフトウェアとして具現化することができる。さらに、メタデータを利用して、ルックアップテーブルまたはデータベースの様々なデータ構造を定義することもできる。このようなソフトウェアは、ソースコードまたはオブジェクト

コードの形式であってもよいが、これに限定されるものではない。ソースコードはさらに、何らかの形式の命令またはオブジェクトコード（アセンブリ言語命令または構成情報を含む）にコンパイルされ得る。本発明のソフトウェア、ソースコードまたはメタデータは、任意のタイプのコード、例えば、C、C++、SystemC、LISA、XML、Java、Brew、SQLおよびその変形（例SQL99またはSQLのプロプライエタリなバージョン）、DB2、Oracle、または種々のハードウェア定義またはハードウェアモデリング言語（例Verilog、VHDL、RTL）および結果として得られるデータベースファイル（例GDSII）を含む、本明細書に記載の機能性を実行する任意の他のタイプのプログラミング言語として実施することができる。結果として、本明細書において同等に使用される「構築物」、「プログラム構築物」、「ソフトウェア構築物」または「ソフトウェア」は、任意の構文または署名を有する任意の種類の任意のプログラミング言語を意味し、言及し、これは、特定された関連する機能または方法論（例えば、プロセッサ回路130を含むプロセッサ回路130またはコンピュータにインスタンス化またはロードされて実行される場合）を提供するか、または提供するように解釈することができる。

【0342】

本発明のソフトウェア、メタデータ、または他のソースコード、および結果として生じるビットファイル（オブジェクトコード、データベース、またはルックアップテーブル）は、コンピュータまたは他の機械可読データ記憶媒体のいずれかのような任意の有形の一次的でない記憶媒体内に、コンピュータ可読命令、データ構造、プログラムモジュール、または上述のようなメモリ140、例えば、フロッピーディスク、CDROM、CD-RW、DVD、磁気ハードドライブ、光学ドライブ、または他のタイプのデータ記憶装置または媒体に関して上述したような他のデータとして具体化することができる。

【0343】

本開示は、本発明の原理の例示と考えるべきであり、本発明を例示された特定の実施形態に限定することを意図していない。この点に関して、本発明は、その適用において、図面に図示された、または実施例に記載された、上および下に記載された構成要素の構成および配置に限定されないことが理解されるべきである。本発明と一致するシステム、方法および装置は、他の実施形態が可能であり、様々な方法で実施および実施されることが可能であり、それらのすべては同等であり、開示の範囲内であると考えられる。

【0344】

本発明は、その特定の実施形態に関して説明されてきたが、これらの実施形態は、本発明の単なる例示であり、限定的ではない。本明細書の説明では、本発明の実施形態を完全に理解するために、電子部品、電子のおよび構造的接続、材料、および構造的変形の例など、多数の特定の詳細が提供される。しかしながら、当業者は、本発明の実施形態が、1つ以上の特定の詳細なしで、または他の装置、システム、アセンブリ、構成部品、材料、部品などを用いて実施され得ることを認識するであろう。他の例では、本発明の実施形態の態様を不明瞭にすることを避けるために、周知の構造、材料、または動作は、具体的に示されず、または詳細に説明されない。さらに、様々な図は縮尺通りに描かれておらず、限定的なもののみならずべきではない。

【0345】

本明細書を通して、「1つの実施形態」、「1つの実施形態」、または特定の「実施形態」への参照は、実施形態に関連して説明される特定の特徵、構造、または特性が、本発明の少なくとも1つの実施形態に含まれ、必ずしもすべての実施形態に含まれるわけではなく、さらに、必ずしも同じ実施形態を参照しているわけではないことを意味する。さらに、本発明の任意の特定の実施形態の特定の特徵、構造、または特徴は、任意の適切な方法で、および他の特徴の対応する使用を伴わない選択された特徴の使用を含む、1つ以上の他の実施形態と任意の適切な組み合わせで組み合わせることができる。さらに、特定の用途、状況または材料を本発明の本質的な範囲および精神に適合させるために、多くの修正がなされ得る。本明細書に記載され、図示された本発明の実施形態の他の変形および変

10

20

30

40

50

更は、本明細書の教示に照らして可能であり、本発明の精神および範囲の一部と考えられることが理解されるべきである。

【0346】

ここでの数値範囲の列举では、同じ程度の精度でその間に介在する各数値が明示的に考えられる。例えば、6-9の範囲については、6および9に加えて数字7および8が考えられ、また、6.0-7.0の範囲については、数字6.0、6.1、6.2、6.3、6.4、6.5、6.6、6.7、6.8、6.9及び7.0が明確に考えられる。さらに、範囲内のすべての介在するサブレンジは、任意の組み合わせで企図され、開示の範囲内である。例えば、5から10の範囲については、サブ範囲5から6、5から7、5から8、5から9、6から7、6から8、6から9、6から10、7から8、7から9、7から10、8から9、8から10、および9から10が、開示された範囲内で考えられる。

【0347】

また、図に示されている1つ以上の要素は、特定の用途に応じて有用であり得るように、より別個のまたは統合された方法で実装され得るか、または特定の場合には除去され得るか、または動作不能にされ得ることも理解されるであろう。一体的に形成された構成要素の組み合わせもまた、本発明の範囲内にあり、特に、個別の構成要素の分離または組み合わせが不明確であるか、識別できない実施形態の場合にはそうである。さらに、本明細書における「結合された」という用語の使用は、「結合」または「結合可能な」などの様々な形態を含めて、他の構成要素を介してまたはそれを介して結合される一体的に形成された構成要素および構成要素を含めて、直接的または間接的な電氣的、構造的または磁氣的結合、接続または取り付け、またはそのような直接的または間接的な電氣的、構造的または磁氣的結合、接続または取り付けのための適応または能力を意味し、かつ含む。

【0348】

さらに、図面／図中のいずれの信号矢印も、特に明記しない限り、例示的なものにすぎず、限定するものではないと考えるべきである。工程の成分の組み合わせも、特に分離または組み合わせの能力が不明確であるか予測可能である場合に、本発明の範囲内で考慮される。本明細書および以下の請求の範囲全体で使用される場合、分離用語「または」は、特に指示されない限り、一般に、接続的意味および分離的意味の両方を有する（そして「排他的または」意味に限定されない）「および／または」を意味することを意図する。本明細書および以下の請求の範囲全体で使用されるように、「a」、「an」、および「the」は、文脈が明確に他を指示しない限り、複数の参照を含む。また、本明細書の説明および以下の請求の範囲全体において使用されるように、「in」の意味は、文脈が明確に他を指示しない限り、「in」および「on」を含む。

【0349】

要約または要約に記載されたものを含む、本発明の例示された実施形態の前述の説明は、包括的であることも、本明細書に開示された正確な形態に本発明を限定することも意図していない。以上から、本発明の新規な概念の精神および範囲から逸脱することなく、多数の変形、改変および置換が意図され、実施され得ることが分かるであろう。本明細書に例示されている特定の装置および装置に関する制限は意図されておらず、または推論されるべきではないことを理解されたい。当然のことながら、特許請求の範囲の範囲に含まれる全てのそのような変更を添付の請求の範囲によってカバーすることが意図されている。

10

20

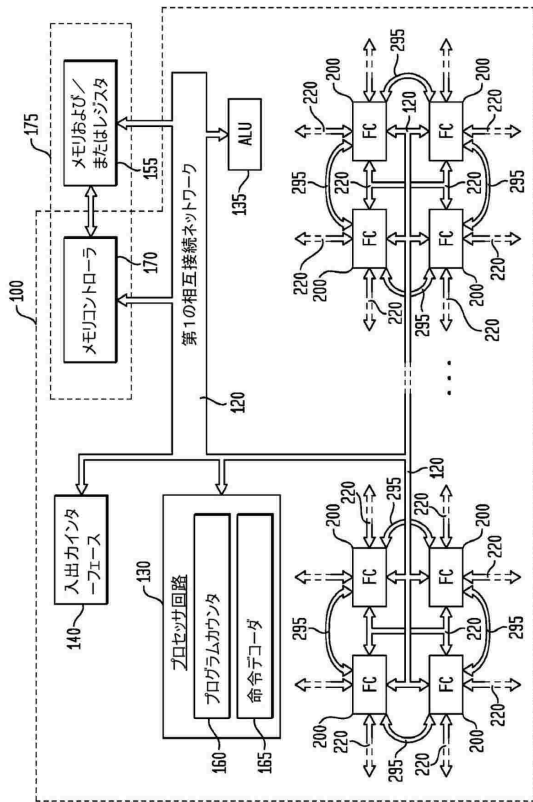
30

40

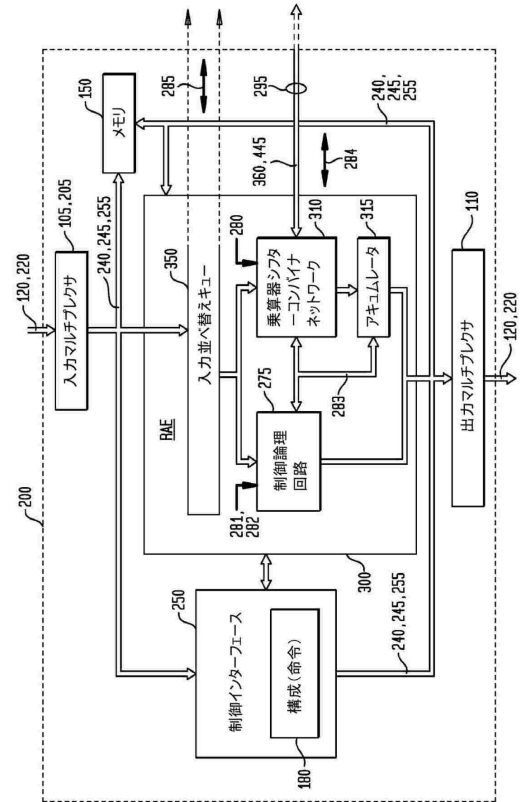
50

【図面】

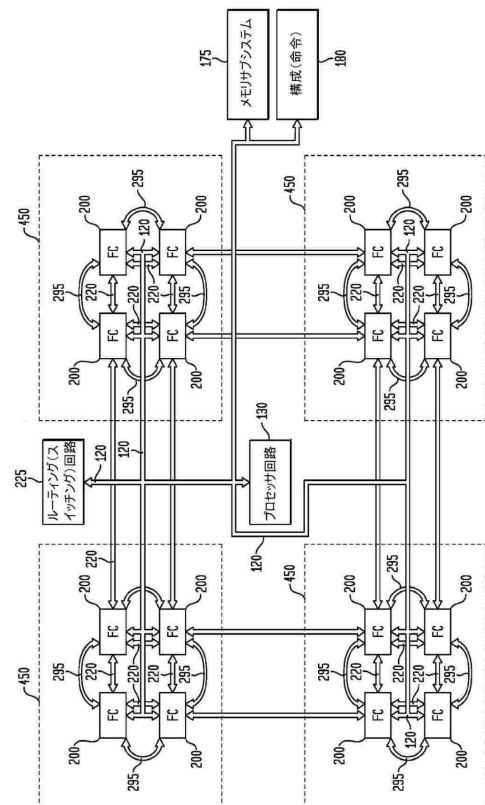
【図 1】



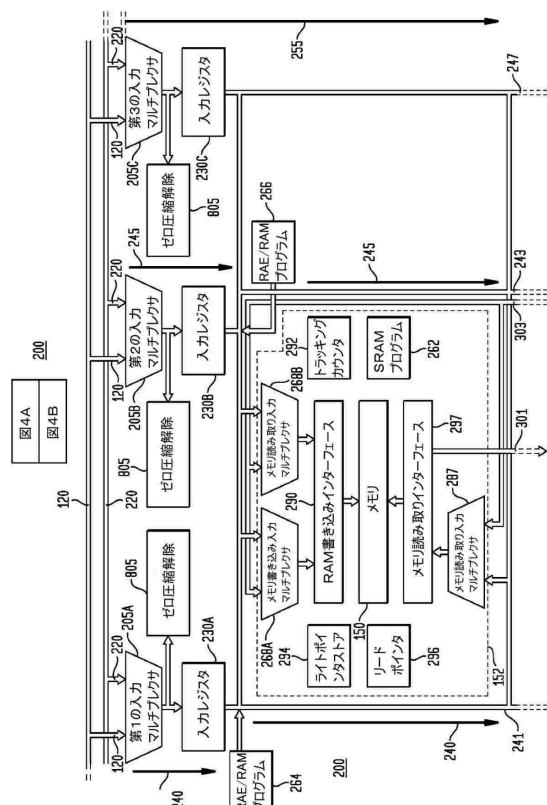
【図 2】



【図 3】



【図 4 A】



10

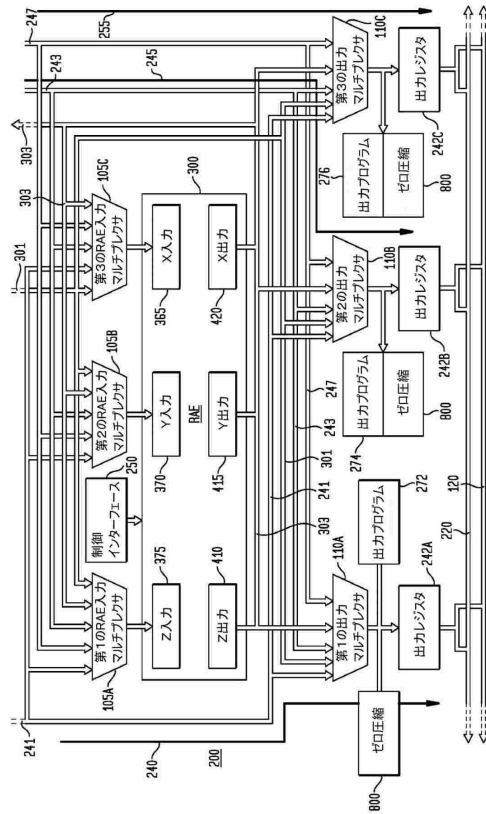
20

30

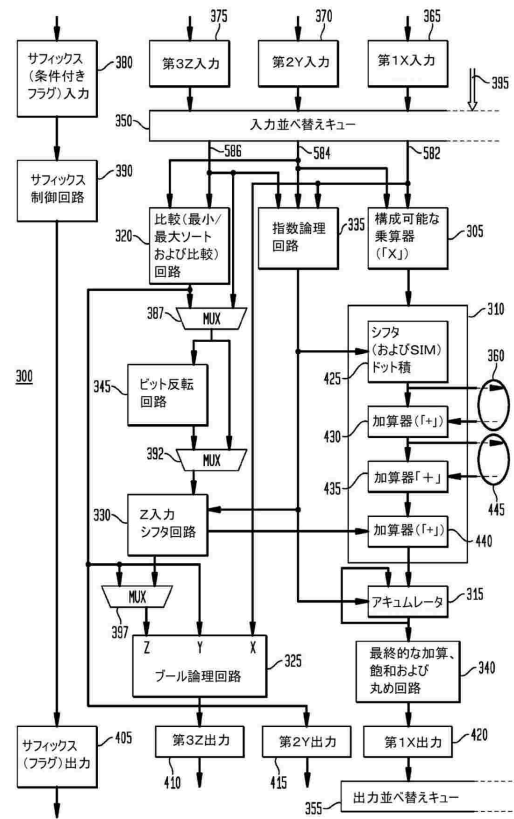
40

50

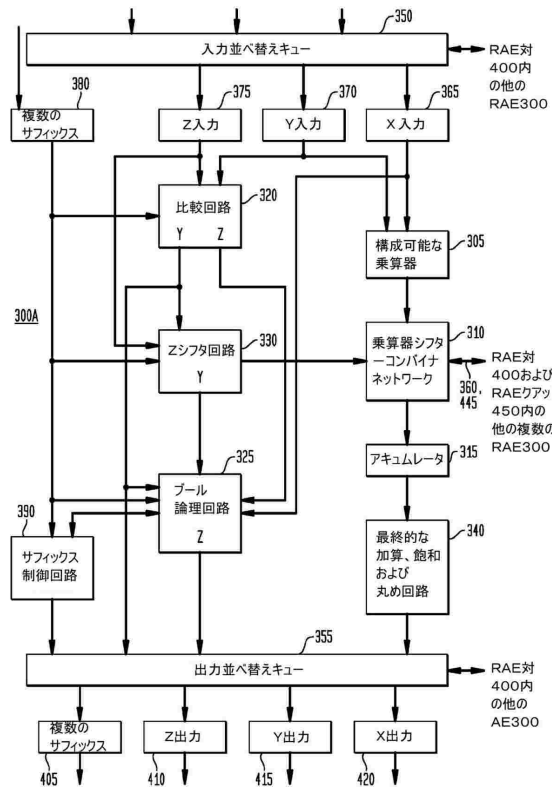
【図 4 B】



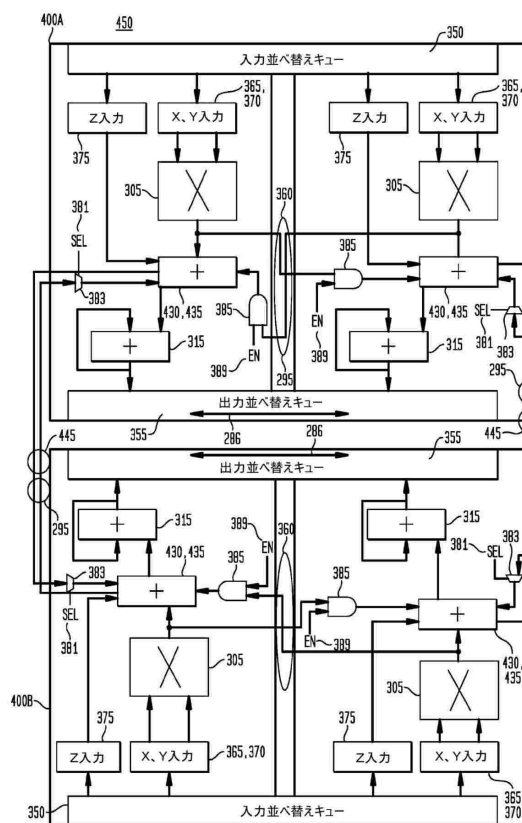
【図 5】



【図 5 A】



【図 6】



10

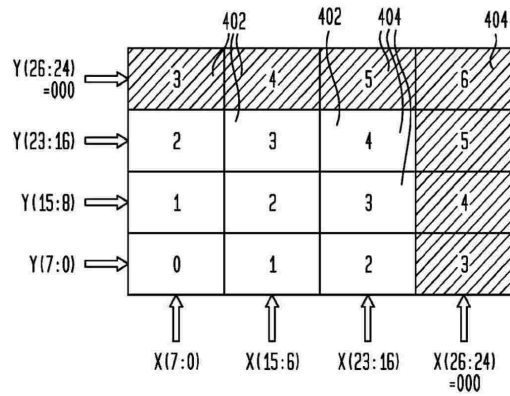
20

30

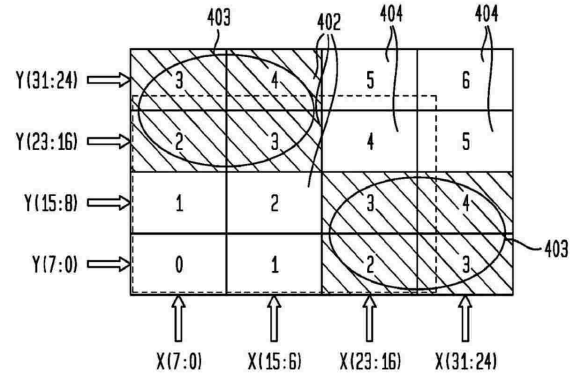
40

50

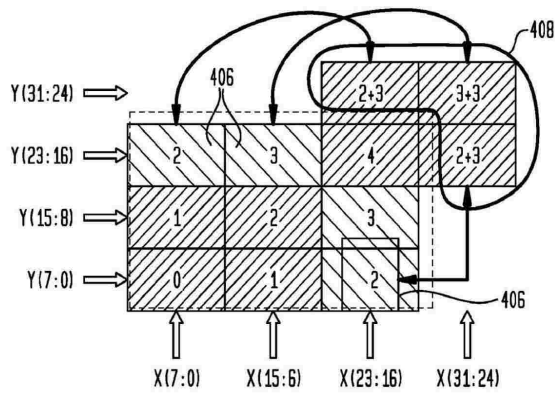
【圖 7】



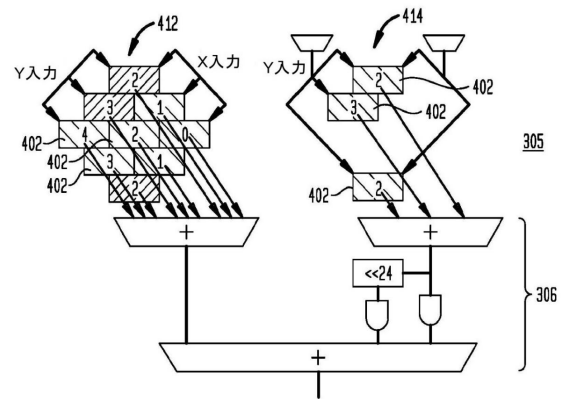
【図 8】



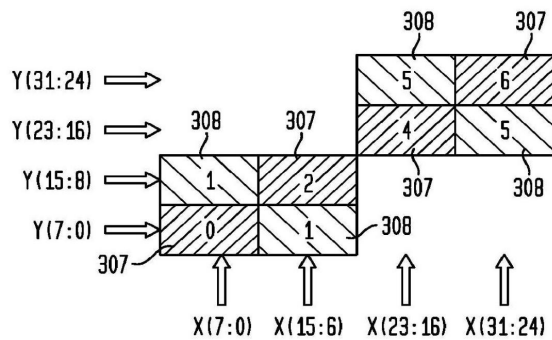
【図 9】



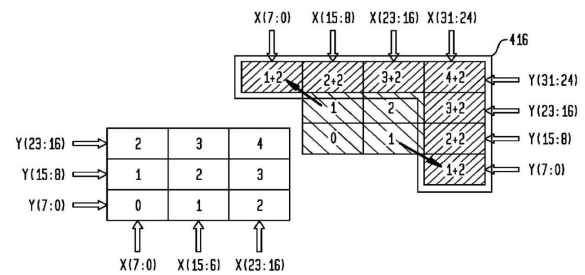
【図 10】



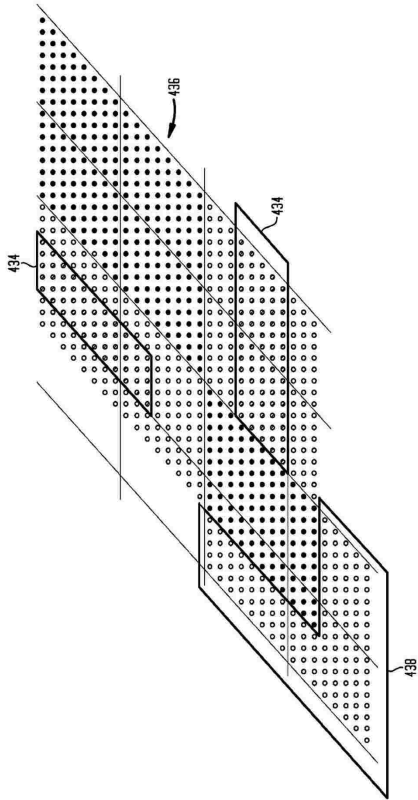
【図 11】



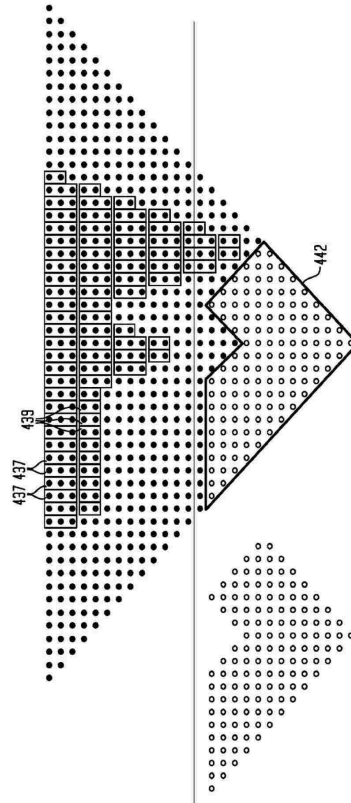
【図 12】



【図 17】



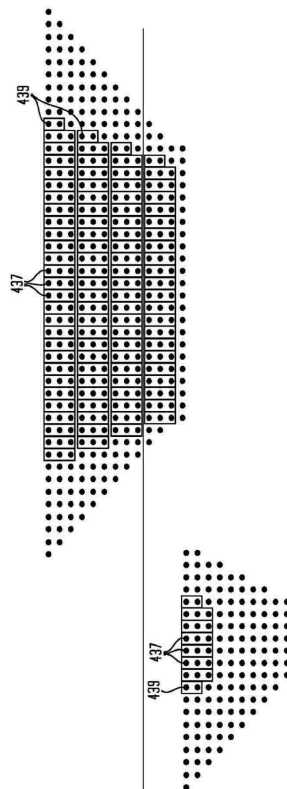
【図 18】



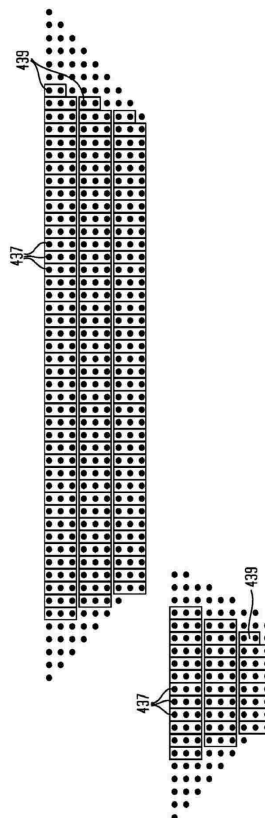
10

20

【図 19】



【図 20】

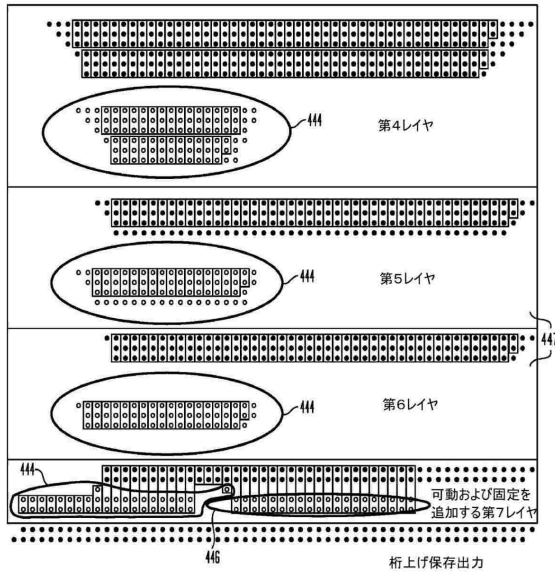


30

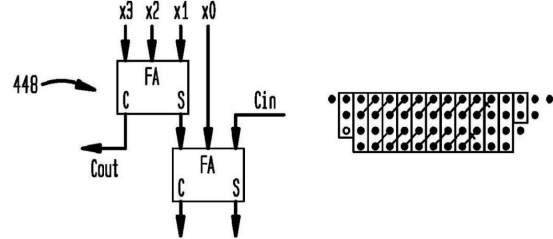
40

50

【図 2 1】

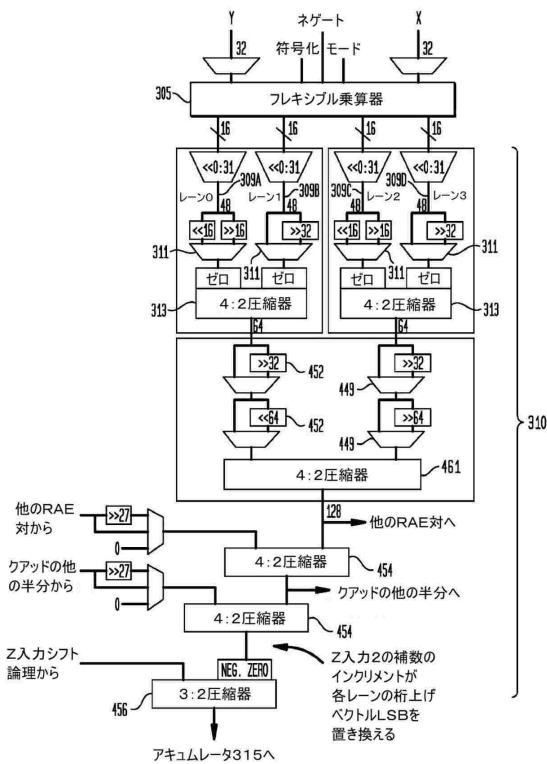


【図 2 2】

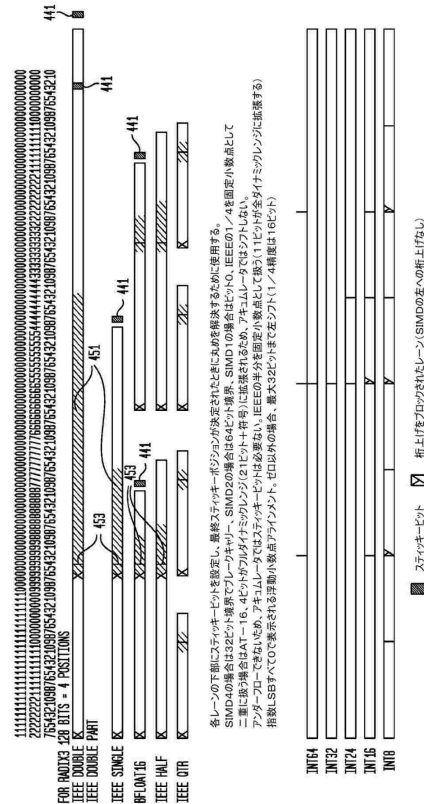


10

【図 2 3】



【図 2 4】



20

30

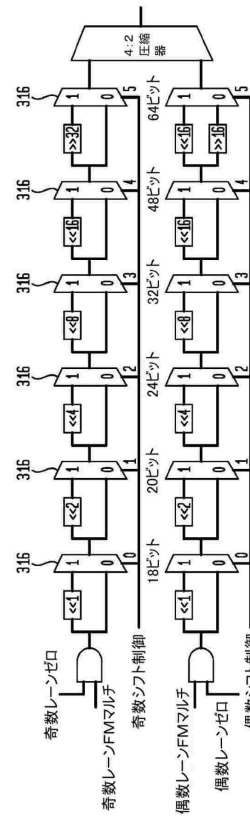
40

50

【図 25】

	第1レイヤ				第2レイヤ		結果 ビット範囲
	レーン3	レーン2	レーン1	レーン0	レーン3-2	レーン1-0	
DOT4 INT	-32	0	-32	0	-64	0	33-0
DOT4 FP QTR TO QTR	E4-8	E4+24	E4-8	E4+24	0	+64	97-80
DOT4 FP QTR TO HALF	E4-16	E4+16	E4-16	E4+16	0	+64	97-80
DOT2 INT	-16	0	-16	0	-64	0	32-0
DOT2 FP HALF	E5-6	E5+10	E5-6	E5+10	0	+64	97-80
SIMD4 INT	0	0	0	0	0	0	
SIMD4 FP QTR	E4+8	E4+8	E4+8	E4+8	0	0	
SIMD2 INT	-16	0	-16	0	0	0	
SIMD2 FP HALF	E5-6	E5+10	E5-6	E5+10	0	0	
SIMD2 BFLOAT	ZERO	E5+16	SIGN	E5+16	0	0	95-80, 31-16
SIMD2 BFLT DOT 0,0	ZERO	E5+16	SIGN	E5+16	-64	0	31-16
SIMD2 BFLT DOT 32,0	ZERO	E5+16	SIGN	E5+16	-64	0	31-16
SIMD2 BFLT DOT 0,32	ZERO	E5+16	SIGN	E5+16	-64	0	31-16
SIMD2 BFLT DOT 32,32	ZERO	E5+16	SIGN	E5+16	-64	0	31-16
SIMD2 BFLT DOT 64,0	ZERO	ZERO	SIGN	E5+16	-64	0	31-16
SIMD2 BFLT DOT 64,32	ZERO	ZERO	SIGN	E5+16	-64	0	31-16
SIMD2 BFLT DOT 64,64	ZERO	ZERO	SIGN	ZERO	-64	0	31-16
SIMD2 BFLT DOT 32,64	ZERO	E5+16	SIGN	ZERO	-64	0	31-16
SIMD2 BFLT DOT 0,64	ZERO	E5+16	SIGN	ZERO	-64	0	31-16
SINGLE	ZERO	E5+16	<0	E5+16	0	+32	95-48
SINGLE >>32	ZERO	E5+16	<0	E5+16	-32	0	63-16
SINGLE >>64	ZERO	E5+16	E5-32	ZERO	-64	0	31-0
SINGLE >>96	ZERO	E5+16	ZERO	ZERO	-64	0	
INT	-16	0	-16	0	-32	0	63-0
INT 32 EXT	0	+16	0	+16	-32	0	63-0
CPLX 4 INT	-32	0	-32	0	0	0	
CPLX 4 FP QTR TO QTR	-32	0	-32	0	0	0	
CPLX 4 FP QTR TO HALF	-32	0	-32	0	0	0	
DOUBLE	E5-4	0	E5-4	0	0	+32	97-44
DOUBLE >>32	E5-4	0	E5-4	0	-32	0	65-12
DOUBLE >>64	E5-4	0	E5-4	ZERO	-64	-32	65-12
DOUBLE >>96	E5-4	ZERO	ZERO	ZERO	-96	0	65-12
INT64 HI	+16	+32	-16	0	0	+64	127-64
INT64 MID	-16	0	-16	0	0	+32	95-32
INT64 LO	-16	0	-16	0	-32	0	63-0
SHIFT RANGE	-32-27	-16-41	-32-27	-16-41			

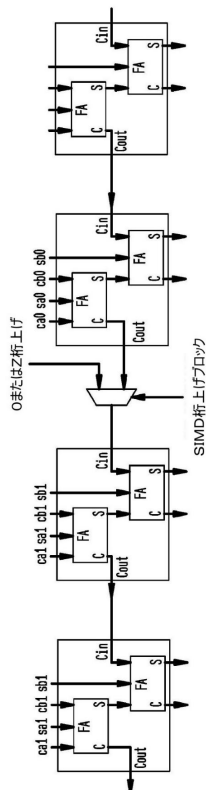
【図 26】



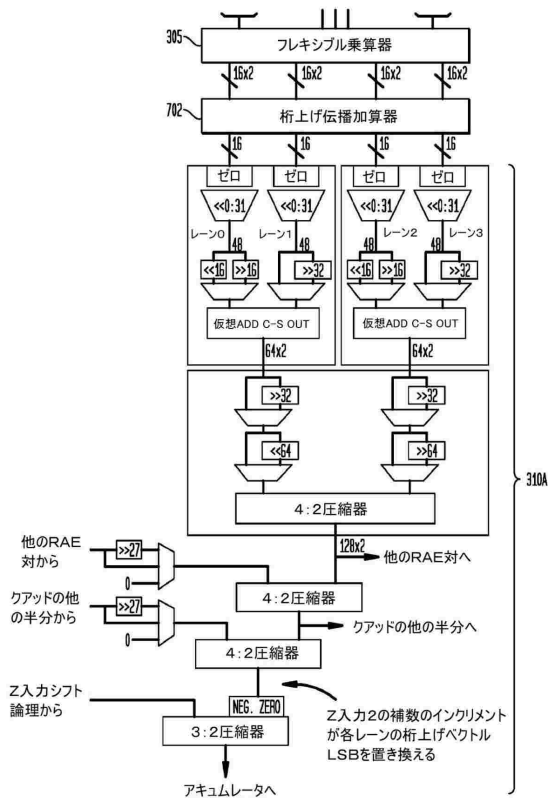
10

20

【図 27】



【図 28】

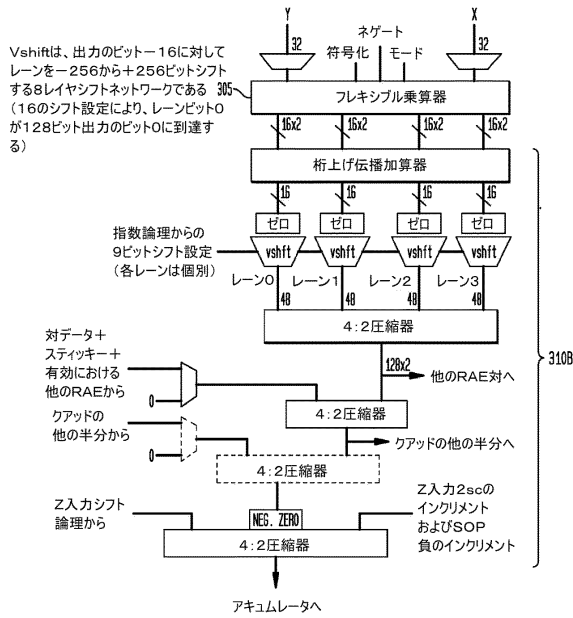


30

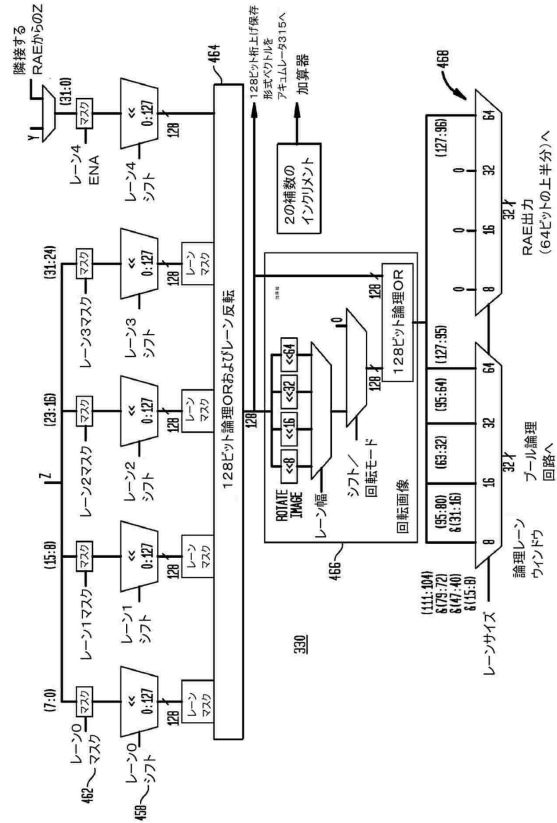
40

50

【図 29】



【図 30】



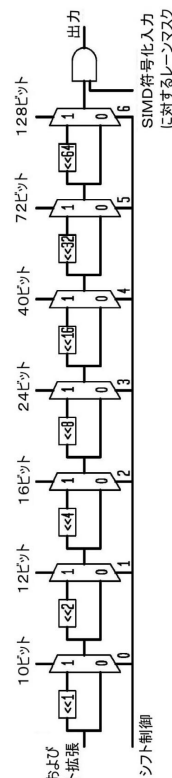
10

20

【図 31】

モード	拡張 ENA	入力マスク/ 隠れビット	符号 拡張	シフト SRC	シフト 距離	シフト バイアス	浮動 小数点 アライン	シフト 共有	レーン マスク /反転	RAE /反転	回転 モード	算術/ 論理	レーン ウィンドウ 機能	レーン1 機能
ダブル	1	0000 0000 FFF0 0000 0000 0000 0010 0000	0000	EXP	EXP	LANE 43:32, LANE 44:3	ダブル	3333	OFF	SHIFT	x	算術	x	11
シングル	0	FFF0 FFF0 0000 0000 0000 0000 0010 0000	0000	EXP	EXP	LANE 44:72	シングル	3333	OFF	SHIFT	x	算術	x	8
ハーフ	0	FFF0 FFF0 FFF0 0000 0000 0000 0010 0000	0000	EXP	EXP	LANE 44:72, LANE 72:44+21	NO	x3111	2	SHIFT	x	算術	x	5
クォーター	0	FFF0 FFF0 FFF0 FFF0 0000 0000 0010 0000	0000	EXP	EXP	LANE 44:72, LANE 72:44+21	NO	x210	4	SHIFT	x	算術	x	4
BE10AT	0	FFF0 FFF0 FFF0 FFF0 0000 0000 0010 0000	0000	EXP	EXP	LANE 44:72, LANE 72:44+21	DOUBLE, SINGLE	x3111	2	SHIFT	x	算術	x	8
符号なし 入力64	1	0000 0000 0000 0000 0000 0000 0000 0000	0000	C/X	C/X	LANE 48	NO	x3333	OFF	CFG	64	x	64	x
符号なし 入力32	0	FFF0 FFF0 0000 0000 0000 0000 0000 0000	0000	C/X	C/X	LANE 48	NO	x3333	OFF	CFG	32	x	32	x
符号なし 入力16	0	FFF0 FFF0 0000 0000 0000 0000 0000 0000	0000	C/X	C/X	LANE 48	NO	x3111	2	CFG	16	x	16	x
符号なし 入力8	0	FFF0 FFF0 0000 0000 0000 0000 0000 0000	0000	C/X	C/X	LANE 48	NO	x210	4	CFG	8	x	8	x
符号あり 入力64	1	0000 0000 0000 0000 0000 0000 0000 0000	1000	C/X	C/X	LANE 48	NO	x3333	OFF	CFG	64	x	64	x
符号あり 入力32	0	FFF0 FFF0 0000 0000 0000 0000 0000 0000	0000	C/X	C/X	LANE 48	NO	x3333	OFF	CFG	32	x	32	x
符号あり 入力16	0	FFF0 FFF0 0000 0000 0000 0000 0000 0000	0010	C/X	C/X	LANE 48	NO	x3111	2	CFG	16	x	16	x
符号あり 入力8	0	FFF0 FFF0 0000 0000 0000 0000 0000 0000	0111	C/X	C/X	LANE 48	NO	x210	4	CFG	8	x	8	x
レーン ウィンドウ	0	0000 0000 0000 0000 0000 0000 0000 0000	0000	C/X	C/X	0	NO	x210	OFF	SHIFT	x	論理	8	x

【図 32】



30

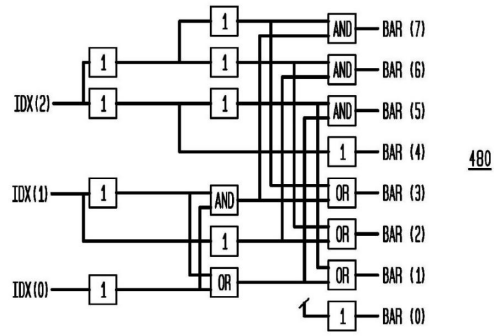
40

50

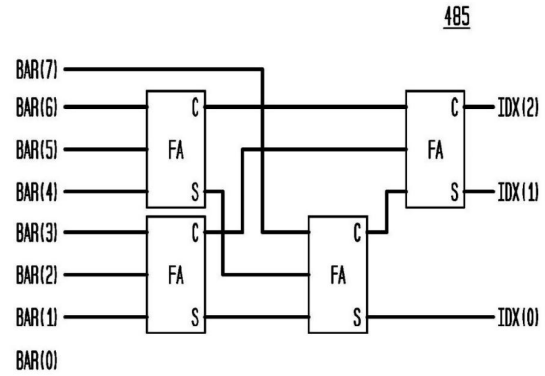
【図 3 3】

OUTPUT	SHIFT	EXP	SM4	SM2	32 BIT	64 BIT	128 BIT	256 BIT	512 BIT	1024 BIT	2048 BIT	4096 BIT	8192 BIT	16384 BIT	32768 BIT	65536 BIT	131072 BIT	262144 BIT	524288 BIT	1048576 BIT	2097152 BIT	4194304 BIT	8388608 BIT	16777216 BIT	33554432 BIT	67108864 BIT	134217728 BIT	268435456 BIT	536870912 BIT	1073741824 BIT	2147483648 BIT	4294967296 BIT	8589934592 BIT	17179869184 BIT	34359738368 BIT	68719476736 BIT	137438953472 BIT	274877906944 BIT	549755813888 BIT	1099511627776 BIT	2199023255552 BIT	4398046511104 BIT	8796093022208 BIT	17592186044416 BIT	35184372088832 BIT	70368744177664 BIT	140737488355328 BIT	281474976710656 BIT	562949953421312 BIT	1125899906842624 BIT	2251799813685248 BIT	4503599627370496 BIT	9007199254740992 BIT	18014398509481984 BIT	36028797018963968 BIT	72057594037927936 BIT	144115188075855872 BIT	288230376151711744 BIT	576460752303423488 BIT	1152921504606846976 BIT	2305843009213693952 BIT	4611686018427387904 BIT	9223372036854775808 BIT	18446744073709551616 BIT	36893488147419103232 BIT	73786976294838206464 BIT	147573952589676412928 BIT	295147905179352825856 BIT	590295810358705651712 BIT	1180591620717411303424 BIT	2361183241434822606848 BIT	4722366482869645213696 BIT	9444732965739290427392 BIT	18889465931478580854784 BIT	37778931862957161709568 BIT	75557863725914323419136 BIT	151115727451828646838272 BIT	302231454903657293676544 BIT	604462909807314587353088 BIT	1208925819614629174706176 BIT	2417851639229258349412352 BIT	4835703278458516698824704 BIT	9671406556917033397649408 BIT	19342813113834066795298816 BIT	38685626227668133590597632 BIT	77371252455336267181195264 BIT	154742504910672534362390528 BIT	309485009821345068724781056 BIT	618970019642690137449562112 BIT	1237940039285380274899124224 BIT	2475880078570760549798248448 BIT	4951760157141521099596496896 BIT	9903520314283042199192993792 BIT	19807040628566084398385987584 BIT	39614081257132168796771975168 BIT	79228162514264337593543950336 BIT	158456325028528675187087900672 BIT	316912650057057350374175801344 BIT	633825300114114700748351602688 BIT	1267650600228229401496703205376 BIT	2535301200456458802993406410752 BIT	5070602400912917605986812821504 BIT	10141204801825835211973625643008 BIT	20282409603651670423947251286016 BIT	40564819207303340847894502572032 BIT	81129638414606681695789005144064 BIT	162259276829213363391578010288128 BIT	324518553658426726783156020576256 BIT	649037107316853453566312041152512 BIT	1298074214633706907132624082305024 BIT	2596148429267413814265248164610048 BIT	5192296858534827628530496329220096 BIT	10384593717069655257060992658440192 BIT	20769187434139310514121985316880384 BIT	41538374868278621028243970633760768 BIT	83076749736557242056487941267521536 BIT	166153499473114484112975882535043072 BIT	332306998946228968225951765070086144 BIT	664613997892457936451903530140172288 BIT	1329227995784915872903807060280344576 BIT	2658455991569831745807614120560689152 BIT	5316911983139663491615228241121378304 BIT	10633823966279326983230456482242756608 BIT	21267647932558653966460912964485513216 BIT	42535295865117307932921825928971026432 BIT	85070591730234615865843651857942052864 BIT	170141183460469231731687303715884105728 BIT	340282366920938463463374607431768211456 BIT	680564733841876926926749214863536422912 BIT	1361129467683753853853498429727072845824 BIT	2722258935367507707706996859454145691648 BIT	5444517870735015415413993718908291383296 BIT	10889035741470030830827987437816582766592 BIT	21778071482940061661655974875633165533184 BIT	43556142965880123323311949751266331066368 BIT	87112285931760246646623899502532662132736 BIT	174224571863520493293247799005065324265472 BIT	348449143727040986586495598010130648530944 BIT	696898287454081973172991196020261297061888 BIT	1393796574908163946345982392040522594123776 BIT	2787593149816327892691964784081045188247552 BIT	5575186299632655785383929568162090376495104 BIT	11150372599265311570767859136324180752990208 BIT	22300745198530623141535718272648361505980416 BIT	44601490397061246283071436545296723011960832 BIT	89202980794122492566142873090593446023921664 BIT	178405961588244985132285746181186892047843328 BIT	356811923176489970264571492362373784095686656 BIT	713623846352979940529142984724747568191373312 BIT	1427247692705959881058285969449495136382746624 BIT	2854495385411919762116571938898990272765493248 BIT	5708990770823839524233143877797980545530986496 BIT	11417981541647679048466287755595961091061972992 BIT	22835963083295358096932575511191922182123945984 BIT	45671926166590716193865151022383844364247891968 BIT	91343852333181432387730302044767688728495783936 BIT	182687704666362864775460604089535377456991567872 BIT	365375409332725729550921208179070754913983135744 BIT	730750818665451459101842416358141509827966271488 BIT	1461501637330902918203684832716283019655932542976 BIT	2923003274661805836407369665432566039311865085952 BIT	5846006549323611672814739330865132078623730171904 BIT	11692013098647223345629478661730264157247460343808 BIT	23384026197294446691258957323460528314494920687616 BIT	46768052394588893382517914646921056628989841375232 BIT	93536104789177786765035829293842113257979682750464 BIT	187072209578355573530071658587684226515959365500928 BIT	374144419156711147060143317175368453031918731001856 BIT	748288838313422294120286634350736906063837462003712 BIT	1496577676626844588240573268701473812127674924007424 BIT	2993155353253689176481146537402947624255349848014848 BIT	5986310706507378352962293074805895248510699696029696 BIT	11972621413014756705924586149611790497021399392059392 BIT	23945242826029513411849172299223580994042798784118784 BIT	47890485652059026823698344598447161988085597568237568 BIT	95780971304118053647396689196894323976171195136475136 BIT	191561942608236107294793378393788647952342390272950272 BIT	383123885216472214589586756787577295904684780545900544 BIT	766247770432944429179173513575154591809369561091801088 BIT	1532495540865888858358347027150309183618739122183602176 BIT	306499108173177771671669405430061836723747824436720432 BIT	612998216346355543343338810860123673447495648873440864 BIT	1225996432692711086686677621720247346894991297746881728 BIT	2451992865385422173373355243440494693789982595493763456 BIT	4903985730770844346746710486880989387579965190987526912 BIT	9807971461541688693493420973761978775159930381975053824 BIT	19615942923083377386986841947523957550319860763950107648 BIT	3923188584616675477397368389504791510063972152790021536 BIT	7846377169233350954794736779009583020127944305580043072 BIT	15692754338466701909589473558019166040255888611160086144 BIT	31385508676933403819178947116038332080511777222320172288 BIT	62771017353866807638357894232076664161023554444640344576 BIT	125542034707733615276715788464153328322047108889280689152 BIT	251084069415467230553431576928306656644094217778561378304 BIT	502168138830934461106863153856613313288188435557122756608 BIT	1004336277661868922213726307713226626576376871114245513216 BIT	200867255532373784442745261542645325315275374222849102432 BIT	401734511064747568885490523085290650630550748445698204864 BIT	803469022129495137770981046170581301261101496891396409728 BIT	1606938044258990275541962092341162602522202993782792819456 BIT	3213876088517980551083924184682325205044405987565585638912 BIT	6427752177035961102167848369364650410088811975131171277824 BIT	12855504354071922204335696738729300820177623950262342547008 BIT	25711008708143844408671393477458601640355247900524685094016 BIT	51422017416287688817342786954917203280710495801049370188032 BIT	102844034832575377634685573909834406561420991602098740376064 BIT	205688069665150755269371147819668813122841983204197480752128 BIT	411376139330301510538742295639337626245683966408394961504256 BIT	822752278660603021077484591278675252491367932816789923008512 BIT	1645504557321206042154969182557350504982735865633579846017024 BIT	3291009114642412084309938365114701009965471731267159692034048 BIT	6582018229284824168619876730229402019930943462534319384068096 BIT	13164036458569648337239753460458804039861886925068638768136192 BIT	26328072917139296674479506920917608079723773850137277536272384 BIT	52656145834278593348959013841835216159447547700274555072544768 BIT	105312291668557186697918027683670432318895095400549110145089536 BIT	210624583337114373395836055367340864637790190801098220290179072 BIT	421249166674228746791672110734681729275580381602196440580358144 BIT	842498333348457493583344221469363458551160763204392881160716288 BIT	1684996666696914987166688442938726917102321526408785762321432576 BIT	3369993333393829974333376885877453834204643052817571524642865152 BIT	6739986666787659948666753771754907668409286105635143049295730304 BIT	13479973333575319897333507543509815336818572211270286098591460608 BIT	26959946667150639794667015087019630673637144422540572197182921216 BIT	53919893334301279589334030174039261347274288845081144394365842432 BIT	107839786668602559178668060348078522694548577690162288788731684864 BIT	215679573337205118357336120696157045389097155380324577577463369728 BIT	431359146674410236714672241392314090778194310760649155154926739456 BIT	862718293348820473429344482784628181556388621521298310309853478912 BIT	1725436586697640946858688965569256363112777243042596620619706957824 BIT	3450873173395281893717377931138512726225554486085193241239413915648 BIT	6901746346790563787434755862277025452451108972170386482478827831296 BIT	13803492693581127574869511724554050904902217944340772964957655662592 BIT	27606985387162255149739023449108101809804435888681545929915311325184 BIT	55213970774324510299478046898216203619608871777363091859830622650368 BIT	110427941548649020598956093796432407239217743554726183719661245300736 BIT	220855883097298041197912187592864814478435487109452367439322490601472 BIT	441711766194596082395824375185729628956870974218904734878644981202944 BIT	883423532389192164791648750371459257913741948437809469757289962405888 BIT	1766847064778384329583297500742918515827483896875618939514579924811776 BIT	3533694129556768659166595001485837031654967793751237879029159849623552 BIT	7067388259113537318333190002971674063309935587502475758058319699247104 BIT	14134776518227074636666380005943348126619871175004951516116639398494208 BIT	28269553036454149273332760011886696253239742350009903032233278796988416 BIT	56539106072908298546665520023773392506479484700019806064466557593976832 BIT	113078212145816597093331040047546785012958969400039612128933115187953664 BIT	226156424291633194186662080095093570025917938800079224257866230375907328 BIT	452312848583266388373324160190187140051835877600158448515732460751814656 BIT	904625697166532776746648320380374280103671755200316897031464921503629312 BIT	1809251394333065553493296640760748560207343510400633794062929843007258624 BIT	3618502788666131106986593281521497120414687020801267588125859686014517248 BIT	7237005577332262213973186563042994240829374041602535176251719372029034496 BIT	14474011154664524427946373126085988481658748083205070352503438744058068992 BIT	28948022309329048855892746252171976963317496166410140705006877488116137984 BIT	57896044618658097711785492504343953926634992332820281410013754976232275968 BIT	115792089237316195423570985008687907853269984665640562820027509952464551936 BIT	231584178474632390847141970017375815706539969331281125640055019904929103872 BIT	463168356949264781694283940034751631413079938662562251280110039809858207744 BIT	926336713898529563388567880069503262826159877325124502560220079619716415488 BIT	1852673427797059126777135760139006525652319754650249005120440159239432830976 BIT	3705346855594118253554271520278013051304639509300498010240880318478865661952 BIT	7410693711188236507108543040556026102609279018600996020481760636957731323904 BIT	14821387422376473014217086081112052205218558037201992040963521273914662647808 BIT	29642774844752946028434172162224104410437116074403984081927042547829325295616 BIT	59285549689505892056868344324448208820874232148807968163854085095658650591232 BIT	118571099379011784113736688648896417641748464297615936327708170191317301182464 BIT	23714219875802356822747337729779283528
--------	-------	-----	-----	-----	--------	--------	---------	---------	---------	----------	----------	----------	----------	-----------	-----------	-----------	------------	------------	------------	-------------	-------------	-------------	-------------	--------------	--------------	--------------	---------------	---------------	---------------	----------------	----------------	----------------	----------------	-----------------	-----------------	-----------------	------------------	------------------	------------------	-------------------	-------------------	-------------------	-------------------	--------------------	--------------------	--------------------	---------------------	---------------------	---------------------	----------------------	----------------------	----------------------	----------------------	-----------------------	-----------------------	-----------------------	------------------------	------------------------	------------------------	-------------------------	-------------------------	-------------------------	-------------------------	--------------------------	--------------------------	--------------------------	---------------------------	---------------------------	---------------------------	----------------------------	----------------------------	----------------------------	----------------------------	-----------------------------	-----------------------------	-----------------------------	------------------------------	------------------------------	------------------------------	-------------------------------	-------------------------------	-------------------------------	-------------------------------	--------------------------------	--------------------------------	--------------------------------	---------------------------------	---------------------------------	---------------------------------	----------------------------------	----------------------------------	----------------------------------	----------------------------------	-----------------------------------	-----------------------------------	-----------------------------------	------------------------------------	------------------------------------	------------------------------------	-------------------------------------	-------------------------------------	-------------------------------------	--------------------------------------	--------------------------------------	--------------------------------------	--------------------------------------	---------------------------------------	---------------------------------------	---------------------------------------	--	--	--	---	---	---	---	--	--	--	---	---	---	--	--	--	--	---	---	---	--	--	--	---	---	---	---	--	--	--	---	---	---	--	--	--	--	---	---	---	--	--	--	---	---	---	---	--	--	--	---	---	---	--	--	--	--	---	---	---	--	--	--	---	---	---	---	--	--	--	---	--	--	---	---	---	---	--	---	---	--	--	--	---	---	---	--	---	---	---	--	--	--	---	---	---	--	--	--	--	---	---	---	--	--	--	---	---	---	---	--	--	--	---	---	---	--	--	--	--	---	---	---	--	--	--	---	---	---	---	--	--	--	---	---	---	--	--	--	--	---	---	---	--	--	--	---	---	---	---	--	--	--	---	---	---	--	--

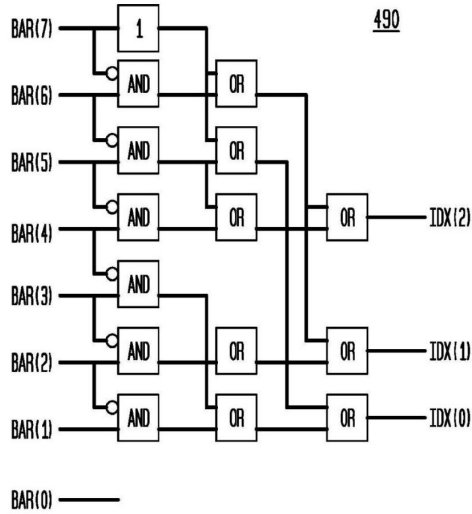
【図 37】



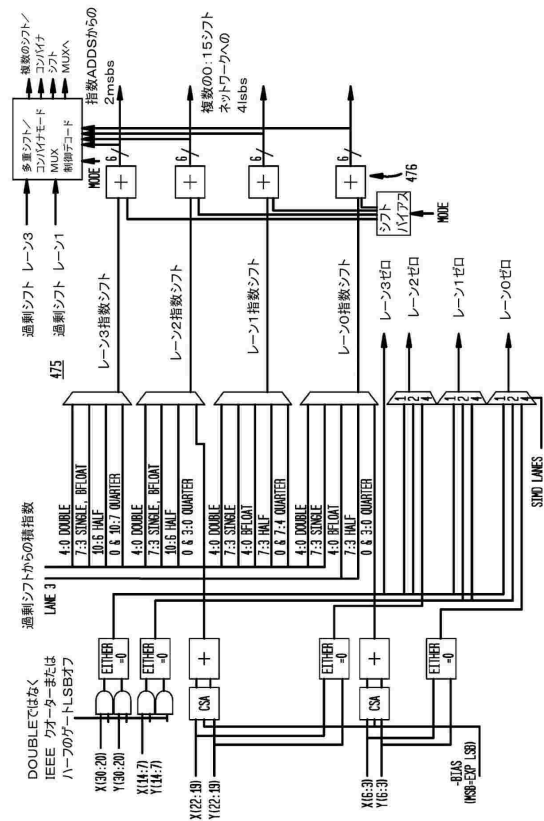
【図 38】



【図 39】



【図 40】



10

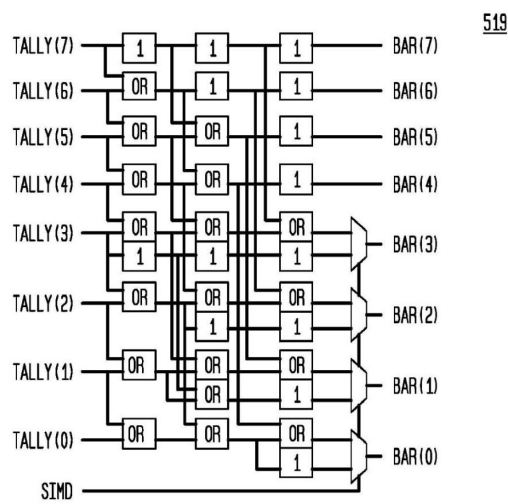
20

30

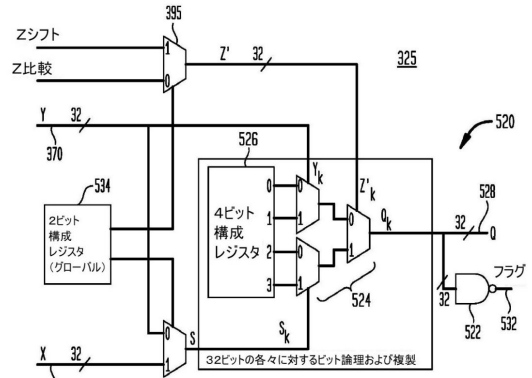
40

50

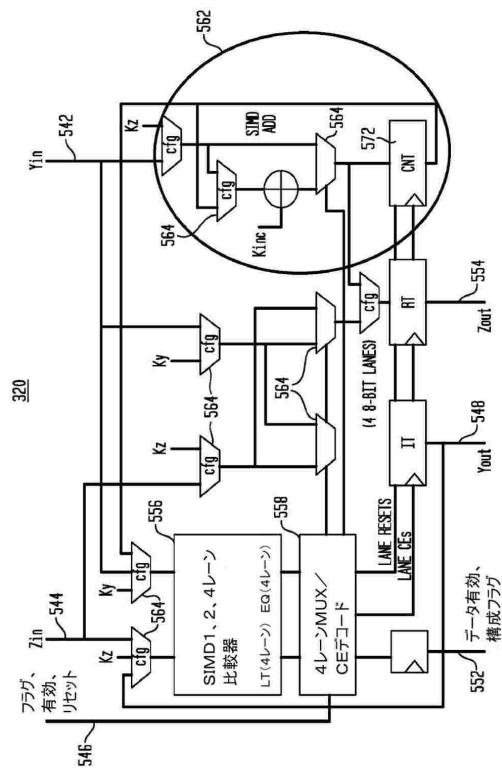
【図 4 5】



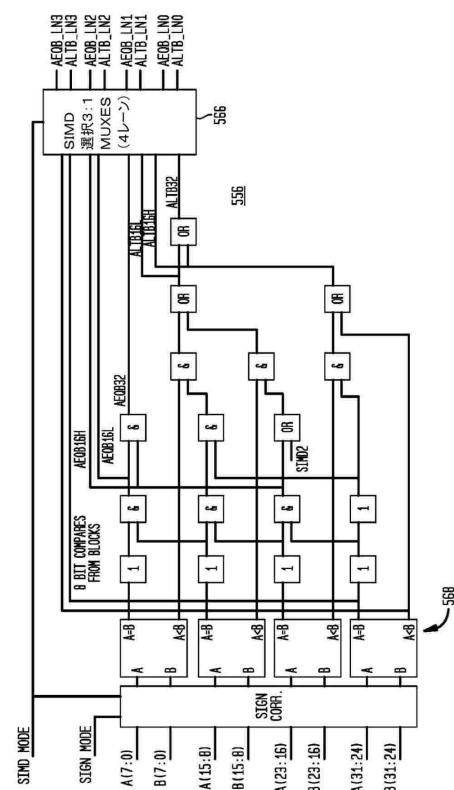
【図 4 6】



【図 4 7】



【図 4 8】



10

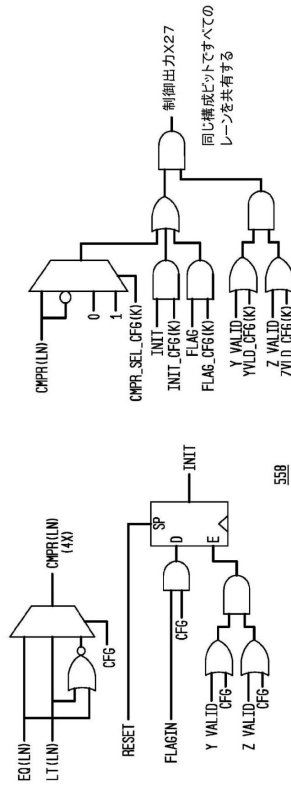
20

30

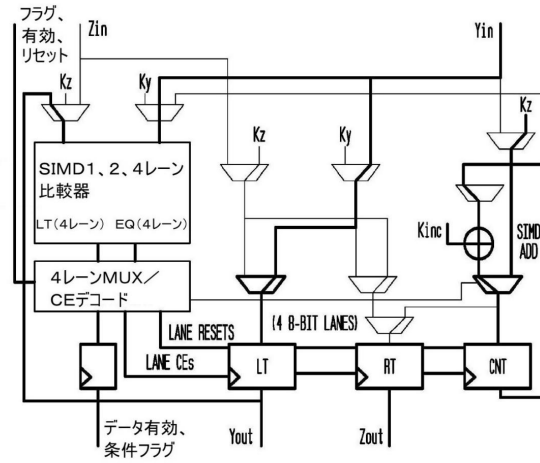
40

50

【図 49】



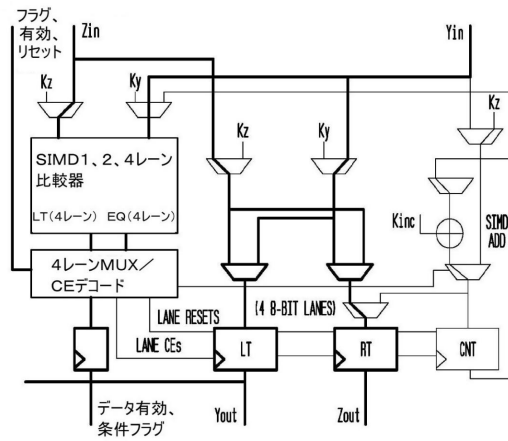
【図 50】



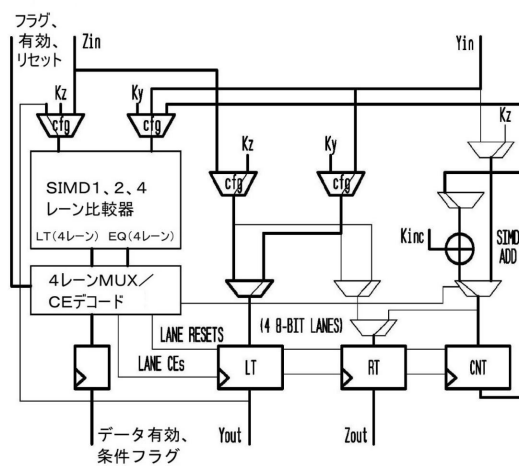
10

20

【図 51】



【図 52】

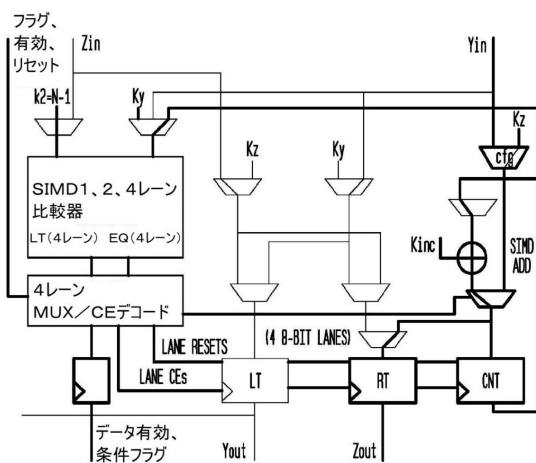


30

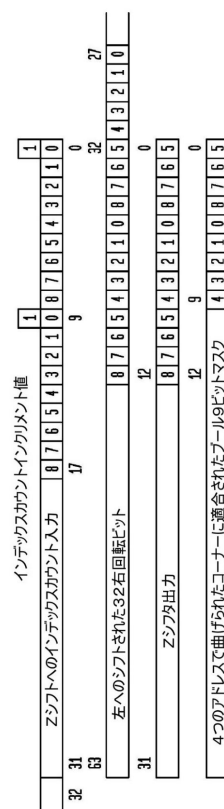
40

50

【図 5 7】



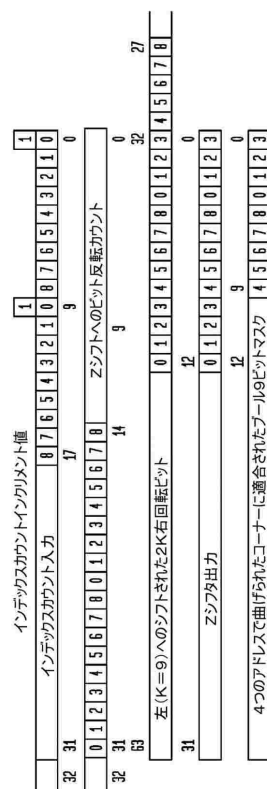
【圖 5 8】



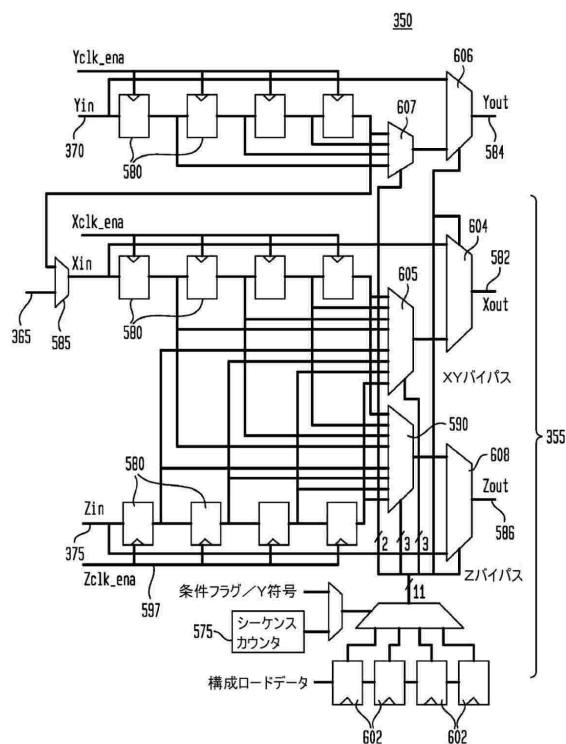
10

20

【図 5 9】



【図 60】

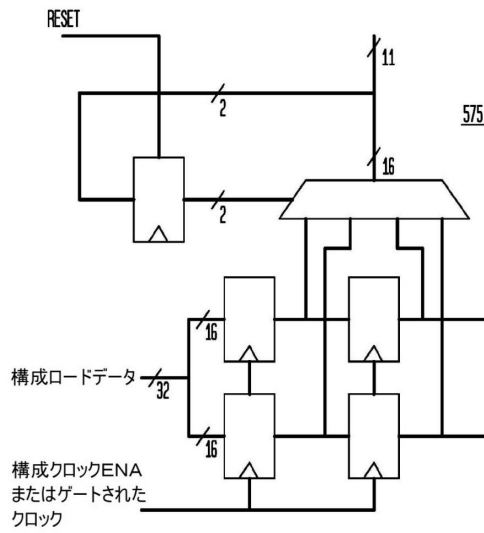


30

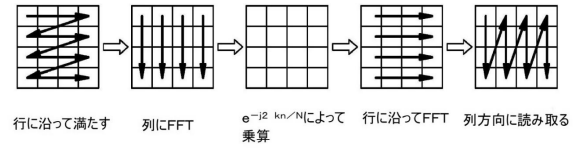
40

50

【図 6 1】

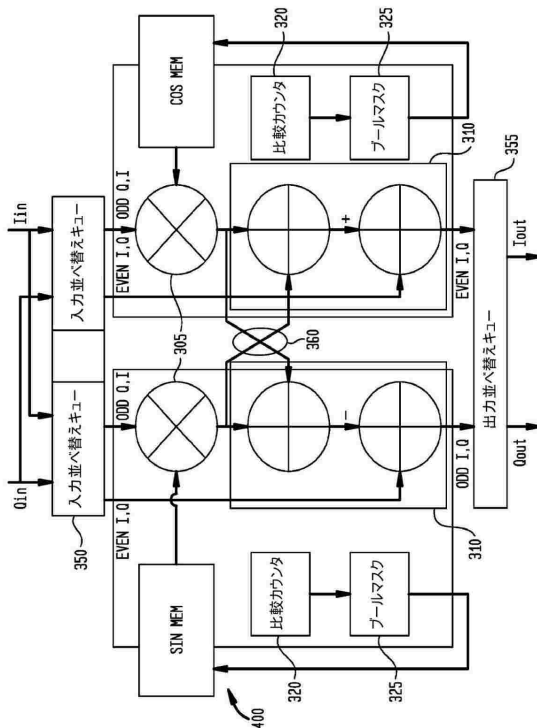


【図 6 2】

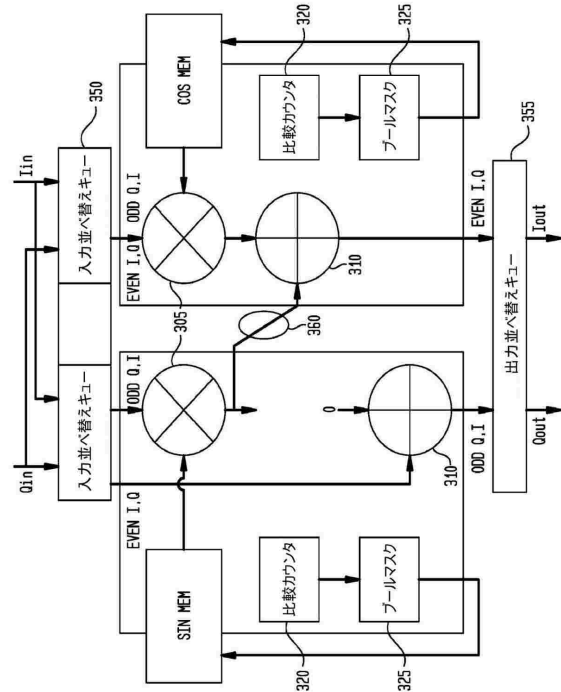


10

【図 6 3】



【図 6 4】



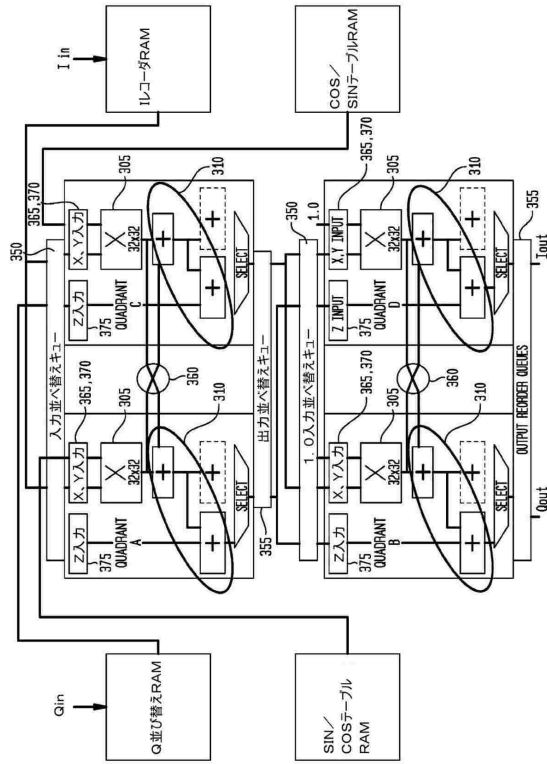
20

30

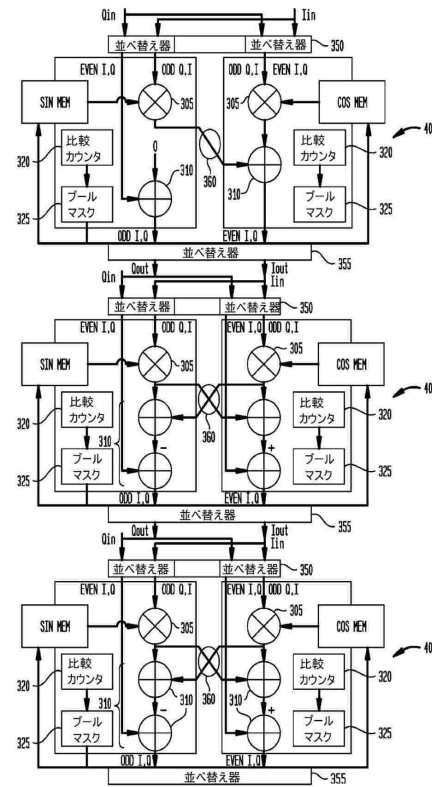
40

50

【図 65】



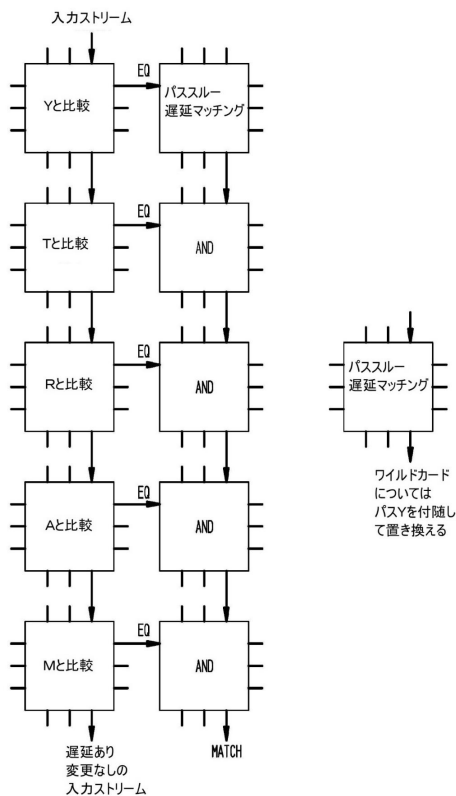
【図 66】



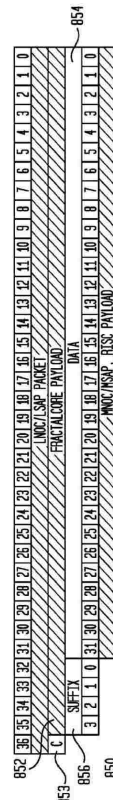
10

20

【図 67】



【図 68】

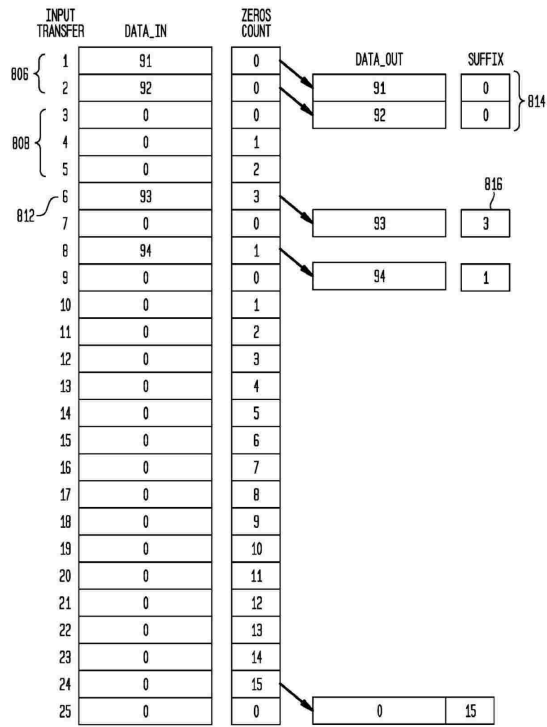


30

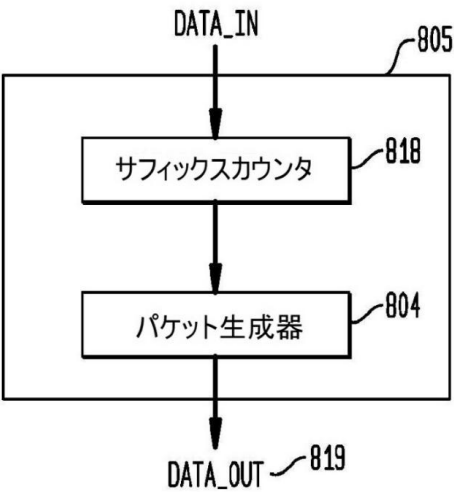
40

50

【図 7 3】



【図 7 4】



10

20

30

40

50

フロントページの続き

(33)優先権主張国・地域又は機関

米国(US)

(31)優先権主張番号 17/015,973

(32)優先日 令和2年9月9日(2020.9.9)

(33)優先権主張国・地域又は機関

米国(US)

(31)優先権主張番号 17/015,950

(32)優先日 令和2年9月9日(2020.9.9)

(33)優先権主張国・地域又は機関

米国(US)

早期審査対象出願

(72)発明者 マスター, ポール, エル.

アメリカ合衆国 9 4 0 8 7 カリフォルニア, サニーベール, ダートシャー ウエイ 7 9 6

(72)発明者 クナップ, スティーブン, ケー.

アメリカ合衆国 9 5 0 7 3 カリフォルニア, ソケル, オリーブ スプリング ロード 1 7 0 1

(72)発明者 アンドレカ, レーモンド, ジェー.

アメリカ合衆国 0 2 8 5 2 ロード アイランド, ノース キングス タウン, トーリー レーン 2 0 2

(72)発明者 フルテク, フレデリク, カーチス

アメリカ合衆国 9 4 0 2 5 カリフォルニア, メンロー パーク, シャロン オークス ドライヴ
2 4 7 0

(72)発明者 ベリヤーエフ, アレクセイ

アメリカ合衆国 9 5 0 0 8 カリフォルニア, キャンベル, ユニオン アヴェニュー 1 8 5, ナ
ンバー 2 8

(72)発明者 フランツ, マーティン, エー.

アメリカ合衆国 9 4 0 8 6 カリフォルニア, サニーベール, フローレンス ストリート 1 1 9

(72)発明者 ミーセン, レネ

アメリカ合衆国 9 4 1 1 6 カリフォルニア, サン フランシスコ, フィフティーンズ アヴェニ
ュー 2 2 1 2

審査官 佐賀野 秀一

(56)参考文献 特表2005-531172 (JP, A)

米国特許出願公開第2017/0286117 (US, A1)

米国特許出願公開第2009/0083518 (US, A1)

(58)調査した分野 (Int.Cl., DB名)

G 0 6 F 7 / 0 0 - 7 / 3 6

G 0 6 F 7 / 5 7

H 0 3 K 1 9 / 1 7 7 0 4