

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6350111号
(P6350111)

(45) 発行日 平成30年7月4日(2018.7.4)

(24) 登録日 平成30年6月15日(2018.6.15)

(51) Int.Cl.

F I

G 0 6 F 7 / 5 3 3 (2 0 0 6 . 0 1)

G 0 6 F 7 / 5 3 3 6 2 0

請求項の数 7 (全 25 頁)

<p>(21) 出願番号 特願2014-169142 (P2014-169142) (22) 出願日 平成26年8月22日 (2014. 8. 22) (65) 公開番号 特開2016-45685 (P2016-45685A) (43) 公開日 平成28年4月4日 (2016. 4. 4) 審査請求日 平成29年5月11日 (2017. 5. 11)</p>	<p>(73) 特許権者 000005223 富士通株式会社 神奈川県川崎市中原区上小田中4丁目1番 1号 (74) 代理人 100094525 弁理士 土井 健二 (74) 代理人 100094514 弁理士 林 恒徳 (72) 発明者 北村 健一 神奈川県川崎市中原区上小田中4丁目1番 1号 富士通株式会社内 審査官 田川 泰宏</p>
--	--

最終頁に続く

(54) 【発明の名称】 乗算回路及びその乗算方法

(57) 【特許請求の範囲】

【請求項1】

乗数の組合せをデコードするブースデコーダと、デコード結果に応じて被乗数と前記乗数の部分積を生成するブースセレクトとを有する部分積生成回路と、

複数の前記部分積を並列に加算するキャリー保存加算器をツリー状に配置し、所定段の前記キャリー保存加算器が出力する加算データとキャリーデータを後段の前記キャリー保存加算器が加算する部分積加算回路と、

複数のデータを並列に乗算する並列モードで、上位側の並列データのデコード結果に応じて補正加算すべき補正ホットビットを生成する補正ホットビット生成部とを有し、

前記部分積加算回路は、前記並列モードで、下位側の並列データを入力し第1の加算データ及び第1のキャリーデータを生成する第1のキャリー保存加算器と、上位側の並列データを入力し第2の加算データ及び第2のキャリーデータを生成する第2のキャリー保存加算器と、前記第1の加算データ及び第1のキャリーデータと前記第2の加算データ及び第2のキャリーデータとを加算する第3のキャリー保存加算器と、前記並列モードで、前記上位側の並列データに前記補正ホットビットを加算する補正ホットビット加算回路を有する乗算回路。

【請求項2】

さらに、

前記並列モードで、前記上位側の要素データの下位側の桁を0にして前記部分積生成回路に入力される被乗数の上位側の要素データを生成し、前記下位側の要素データの上位側

10

20

の桁を0または符号ビットにして前記部分積生成回路に入力される被乗数の下位側の要素データを生成する分割回路を有し、

前記ブースセクタは、前記デコード結果が負の部分積を選択する場合、セレクトされるデータをビット反転して前記負の部分積を生成し、

前記並列モード及び単一のデータを乗算する通常モードのいずれの場合も、前記負の部分積の最下位ビットにホットビットが加算される請求項1に記載の乗算回路。

【請求項3】

さらに、

前記並列モードで、前記第3のキャリー保存加算器に入力される前記第1の加算データ及び第1のキャリーデータの上位側の桁を0に変更し、前記第2の加算データ及び第2のキャリーデータの下位側の桁を0に変更するゼロマスク回路を有する請求項1または2に記載の乗算回路。

10

【請求項4】

前記第3のキャリー保存加算器は、第1の前記補正ホットビット加算回路を有し、

前記第1の補正ホットビット加算回路は、前記並列モードで、前記第2の加算データまたは前記第2のキャリーデータに前記補正ホットビットを加算する請求項1または3に記載の乗算回路。

【請求項5】

前記部分積加算回路は、前記第3のキャリー保存加算器が出力する第3の加算データ及び第3のキャリーデータを加算する全加算器を有し、

20

前記全加算器は、第2の前記補正ホットビット加算回路を有し、

前記第2の補正ホットビット加算回路は、前記並列モードで、前記第3の加算データ及び前記第3のキャリーデータに前記補正ホットビットを加算する請求項4に記載の乗算回路。

【請求項6】

前記キャリー保存加算器は、4つの入力データを演算して加算データとキャリーデータを有する2つの出力データを出力し、

前記部分積加算回路は、複数段のキャリー保存加算器を有し、

前記補正ホットビット生成部は、前記並列モードで、前記上位側の要素データについて、前記乗数の複数の組合せのデコード結果が全て正の部分積(Z)の場合は前記補正ホットビットを0に、全て負の部分積(F)の場合は前記補正ホットビットを2に、一部に負の部分積が含まれる場合は前記補正ホットビットを1にする請求項1に記載の乗算回路。

30

【請求項7】

乗数の組合せをデコードするブースデコーダと、デコード結果に応じて被乗数と前記乗数の部分積を生成するブースセクタとを有する部分積生成回路と、

複数の前記部分積を並列に加算するキャリー保存加算器をツリー状に配置し、所定段の前記キャリー保存加算器が出力する加算データとキャリーデータを後段の前記キャリー保存加算器が加算する部分積加算回路とを有し、

前記部分積加算回路は、前記並列モードで、下位側の並列データを入力し第1の加算データ及び第1のキャリーデータを生成する第1のキャリー保存加算器と、上位側の並列データを入力し第2の加算データ及び第2のキャリーデータを生成する第2のキャリー保存加算器と、前記第1の加算データ及び第1のキャリーデータと前記第2の加算データ及び第2のキャリーデータとを加算する第3のキャリー保存加算器とを有する乗算回路の乗算方法において、

40

複数のデータを並列に乗算する並列モードで、上位側の並列データのデコード結果に応じて補正加算すべき補正ホットビットを生成し、

前記部分積加算回路は、前記並列モードで、前記上位側の並列データに前記補正ホットビットを加算する乗算方法。

【発明の詳細な説明】

【技術分野】

50

【0001】

本発明は、乗算回路及びその乗算方法に関する。

【背景技術】

【0002】

乗算器は、一般的には、被乗数を乗数の各桁と乗算して複数の部分積を生成し、複数の部分積を加算して乗算値を出力する。部分積の数を減らす方法としてブースアルゴリズム(Booth Algorithm)が知られている。ブースアルゴリズムによれば、例えば2次のブースアルゴリズムであれば、被乗数を複数桁の乗数に応じて1倍の正数もしくは負数、または2倍の正数もしくは負数を部分積として生成する。そして、2次のブースアルゴリズムでは乗数を2ビット単位で処理するので、部分積の数を1/2にすることができる。

10

【0003】

また、複数の部分積を短時間で加算する方法として、ワレスツリー(Wallace tree)が知られている。ワレスツリーは、桁上げ保存加算器(Carry Saved Adder: CSA)をツリー状に配置した構成を有し、ツリーの各段では複数のCSAを並列に配置して並列に演算する。CSAはキャリデータである桁上げビットを下位桁から上位桁に伝搬させることなく保持するので、演算結果が出力されるまでの論理段数を短くできる。

【0004】

上記のように、ブースアルゴリズムによる部分積生成回路と、ワレスツリーによる部分積加算回路とを組み合わせることで、乗算結果が出力されるまでの時間を短くする。

【0005】

このようなブースアルゴリズムとワレスツリーを組み合わせた乗算器は、ビット数を増大させることでn倍精度の被乗数と乗数を乗算する。そして、n倍精度の乗算器は、n/m倍精度でm並列の被乗数と乗数を演算する並列モードで動作することが望まれる。例えば、単精度が32ビットの場合、倍精度は64ビットである。その場合、2つの32ビットデータを並列に乗算することで、乗算効率を高めることができる。

20

【先行技術文献】

【特許文献】

【0006】

【特許文献1】特開平7-121354号公報

【発明の概要】

【発明が解決しようとする課題】

【0007】

ブースアルゴリズムによる部分積生成回路は、乗数Yの複数桁の組合せをデコードするブースデコードと、デコード結果に応じて被乗数Xの、例えば2次であれば、1倍、2倍、マイナス1倍、マイナス2倍(被乗数Xの $\times 1$ 、 $\times 2$ 、 $-\times 1$ 、 $-\times 2$)のいずれかのデータを選択するブースセレクタとを有する。2倍は被乗数Xを左シフトすることで簡単に求めることができる。

30

【0008】

しかしながら、デコード結果がマイナス1倍、マイナス2倍の場合、負数を2の補数にするために、セレクトしたデータのビット反転とその最下位桁に1を加える処理を行う必要がある。この最下位桁に加えられる1を、ホットビット(Hotbit)と称する。したがって、ブースアルゴリズムの部分積生成回路は、ブースデコード結果に応じてホットビットを加算する回路が必要になる。

40

【0009】

このことは、単一のデータを乗算する通常モードでは、最下位ビットにだけホットビットを加算すれば良いが、複数のデータを並列に乗算する並列モードでは、複数の並列データそれぞれの最下位ビットにホットビットを加算することが必要になることを意味する。2の補数にするためのホットビットは、通常、ワレスツリーの入力段で加算される。そのため、通常モードか並列モードかによって、並列データの最下位ビットにホットビットを加算する回路を有効にするか否かを切り替える制御が必要になる。

50

【 0 0 1 0 】

そこで、実施の形態の第 1 の側面の目的は、簡単な回路で並列モードでの乗算を行うことにより、乗算回路の物理量および消費電力を削減できる乗算器及び乗算器の乗算方法を提供することにある。

【課題を解決するための手段】

【 0 0 1 1 】

本実施の形態の第 1 の側面は、乗数の組合せをデコードするブースデコーダと、デコード結果に応じて被乗数と前記乗数の部分積を生成するブースセクタとを有する部分積生成回路と、

複数の前記部分積を並列に加算するキャリー保存加算器をツリー状に配置し、所定段の前記キャリー保存加算器が出力する加算データとキャリーデータを後段の前記キャリー保存加算器が加算する部分積加算回路と、

複数のデータを並列に乗算する並列モードで、上位側の並列データのデコード結果に応じて補正加算すべき補正ホットビットを生成する補正ホットビット生成部とを有し、

前記部分積加算回路は、前記並列モードで、下位側の並列データを入力し第 1 の加算データ及び第 1 のキャリーデータを生成する第 1 のキャリー保存加算器と、上位側の並列データを入力し第 2 の加算データ及び第 2 のキャリーデータを生成する第 2 のキャリー保存加算器と、前記第 1 の加算データ及び第 1 のキャリーデータと前記第 2 の加算データ及び第 2 のキャリーデータとを加算する第 3 のキャリー保存加算器と、前記並列モードで、前記上位側の並列データに前記補正ホットビットを加算する補正ホットビット加算回路を有する乗算回路である。

【発明の効果】

【 0 0 1 2 】

第 1 の側面によれば、簡単な回路構成で並列モードの乗算を行うことができ、乗算回路の物理量および消費電力を削減できる。

【図面の簡単な説明】

【 0 0 1 3 】

【図 1】ブースアルゴリズムとワレスツリーを組み合わせた乗算回路の第 1 の例を示す図である。

【図 2】ブースアルゴリズムとワレスツリーを組み合わせた乗算回路の第 2 の例を示す図である。

【図 3】ブースアルゴリズムとワレスツリーを組み合わせた乗算回路の第 3 の例を示す図である。

【図 4】図 3 の乗算回路を実現する場合の問題点を示す図である。

【図 5】通常モードと並列モードの違いを説明する図である。

【図 6】本実施の形態における乗算回路の概略構成図である。

【図 7】ブースセクタ 1 2 の構成を示す図である。

【図 8】ワレスツリー加算器 20_1 、 20_0 と、ワレスツリー加算器が内蔵する $4\text{ to }2\text{ CSA}$ との回路構成を示す図である。

【図 9】 $4\text{ to }2\text{ CSA}$ の 4 入力の全パターンに対する出力 S 、 C とキャリーの関係を示す図である。

【図 10】図 9 に示した 5 つのパターンの発生したキャリー CRY 、発生すべきキャリー CRY 、それらの差分をまとめた図である。

【図 11】ワレスツリー加算器により伝搬するキャリーと不足するホットビットの数の一例を示す図である。

【図 12】3 段のワレスツリー加算器の各段の $4\text{ to }2\text{ CSA}$ の入力パターンを示す図である。

【図 13】入力パターンの組合せ例に対する 1 段目での差分（不足数）、2、3 段目で発生したキャリー CRY 、最終的に残った数（補正值）を示す図である。

【図 14】入力パターンの組合せ例に対する 1 段目での差分（不足数）、2、3 段目で発

10

20

30

40

50

生したキャリー C R Y , 最終的に残った数 (補正值) を示す図である。

【図 1 5】補正ホットビット生成部 5 0 の回路図である。

【図 1 6】本実施の形態における乗算回路の具体的な構成を示す図である。

【図 1 7】本実施の形態における乗算回路の具体的な構成を示す図である。

【図 1 8】分割回路の一例を示す図である。

【図 1 9】ブースデコーダ 1 1 とブースセクタ 1 2 _ 1 , 1 2 _ 0 の構成を示す図である。

【図 2 0】ワレスツリー加算器の構成を示す図である。

【図 2 1】3 次のブースアルゴリズムの場合のブースデコード表である。

【図 2 2】3 次のブースアルゴリズムを使用した場合の補正ホットビット生成ユニットを示す図である。

10

【発明を実施するための形態】

【0014】

図 1 は、ブースアルゴリズムとワレスツリーを組み合わせた乗算回路の第 1 の例を示す図である。図 1 (A) には、1 つの入力データに対して乗算を行う通常モードの場合の乗算回路が、図 1 (B) には、2 つの入力データに対して乗算を行う並列モードの場合の乗算回路が示されている。

【0015】

乗算回路は、入力データが単精度と倍精度 (または n 倍精度) のいずれでも演算可能に構成されている。図 1 の例では 2 倍精度 ($n = 2$) の演算が可能である。さらに、乗算回路は、 n 倍精度のデータに代えて m 個のデータを並列に演算する n / m 倍精度の m 並列の演算も可能である。 $m = 2$ とすると、単精度 ($2 / 2 = 1$) の 2 つのデータを 2 並列に乗算する。

20

【0016】

図 1 に示された乗算回路は、乗数 Y の複数ビットの組合せをデコードするブースデコーダ 1 1 と、デコード結果に応じて被乗数 X と乗数 Y の部分積 $P M$ を複数生成するブースセクタ 1 2 とを含む部分積生成回路 1 0 を有する。さらに、乗算回路は、複数の部分積 $P M$ を加算して乗算値 $P M$ を出力するワレスツリー構成の部分積加算回路 2 0 を有する。

【0017】

例えば、2 次のブースアルゴリズムの場合、ブースデコーダ 1 1 は乗数 Y の 3 ビットをデコードし、ブースセクタ 1 2 は、デコード結果に応じて 0 , X , $2 X$, $- 2 X$, $- X$, 0 のいずれかを選択して部分積 $P M$ を出力する。そして、乗数 Y は 2 ビットずつシフトしてデコードされるので、部分積 $P M$ の数は乗数 Y のビット数の $1 / 2$ に減らすことができる。部分積 $P M$ の上位側には符号ビット S が拡張して格納されている。

30

【0018】

ブースセクタ 1 2 は、デコード結果に応じて $- 2 X$, $- X$ を選択する場合、被乗数 X を左シフトして $2 X$ を生成しまたは左シフトせずに X を生成し、負数を表すために $2 X$, X のビットを反転し、1 を加算 ($+ 1$) するためのホットビットを生成する。そして、生成されたホットビットはワレスツリーの部分積加算回路 2 0 に入力され加算される。図 1 には、このホットビット H が便宜的に部分積 $P M$ の最下位ビットに表記されている。

40

【0019】

ワレスツリーの部分積加算回路 2 0 は、複数の部分積 $P M$ を並列に加算するキャリー保存加算器 (以下 C S A : carry saved adder, 図示せず) をツリー状に配置し、所定段のキャリー保存加算器が出力する加算データとキャリーデータを後段のキャリー保存加算器が加算する。4 入力に対して加算データとキャリーデータの 2 データを出力する 4 to 2 C S A の場合は、各段の C S A を通過するたびに加算すべきデータ数が $1 / 2$ に減っていく。各 C S A はキャリーを伝搬することなく保存するので、各段の C S A の演算時間は短い。そして、ツリー状に配置することで、加算すべき部分積 $P M$ の入力から最終加算結果が出力されるまでの段数も少なくなり、短時間で結果を出力する。

【0020】

50

以上の基本的な説明に基づいて、ブースアルゴリズムとワレスツリーによる乗算器の問題点について説明する。図1(A)の通常モードの場合、ワレスツリーの加算器20には単一データの部分積PMが入力されるので、ワレスツリー加算器20が出力する乗算値PMは正しい結果になる。

【0021】

一方、図1(B)の並列モードの場合、 $2/2 (= n/m)$ 倍精度のデータを $2 (= m)$ 要素並べた並列データをそのまま演算する例である。この場合、2倍精度の被乗数には上位側の要素の被乗数 X_{-1} と下位側の要素の被乗数 X_{-0} とが並んでいて、この並列データからブースセクタ12が部分積PMを生成する。この場合、部分積PMも並列データであり、上位側の被乗数 X_{-1} の最下位ビットにもホットビットHが加算される必要がある。そして、ワレスツリー加算器20には、並列データの部分積PMが入力されると、部分積PMは上位側にビットシフトしているため、上位側と下位側の部分積PMがワレスツリー加算器20内のCSA加算器で混ざり合い、ワレスツリー加算器20が出力する乗算値PMは間違った結果になる。

10

【0022】

図2は、ブースアルゴリズムとワレスツリーを組み合わせた乗算回路の第2の例を示す図である。この乗算回路では、ブースセクタ12が上位側の部分積 PM_{-1} と下位側の部分積 PM_{-0} とを別々に生成し、2つのワレスツリー加算器 20_{-1} 、 20_{-2} が2つの部分積 PM_{-1} 、 PM_{-0} をそれぞれ加算して、それぞれの乗算値 MP_{-1} 、 MP_{-0} を出力する。最後に、2つの乗算値 MP_{-1} 、 MP_{-0} を合成して正しい並列データの乗算値MPを得る。

20

【0023】

図2の例は、正しい乗算値MPを得ることができるが、並列データの数(2個)に対応した数(2個)のワレスツリー加算器が必要になり、回路規模が大幅に増大する。

【0024】

図3は、ブースアルゴリズムとワレスツリーを組み合わせた乗算回路の第3の例を示す図である。この乗算回路では、ブースセクタ20が、下位側の要素の部分積 PM_{-0} の上位側を0でマスクし上位側の要素の部分積 PM_{-1} の下位側を0でマスクする0マスク機能と、上位側の要素にホットビットを加算するホットビット加算回路を有する。このような下位側の要素の部分積 PM_{-0} と上位側の要素の部分積 PM_{-1} とをワレスツリー加算器20が加算演算しても、两部分積のデータが混ざり合うことはなく、ワレスツリー加算器20が出力する乗算結果は正しい並列データの乗算結果になる。

30

【0025】

図4は、図3の乗算回路を実現する場合の問題点を示す図である。図3で説明したホットビット加算回路は、例えば、部分積PMを入力するワレスツリー加算器20の入力段に追加される。そして、乗算回路が、単一の要素を乗算する通常モードと、複数の要素の並列データを乗算する並列モードとを切り替え制御される必要がある。

【0026】

図5は、通常モードと並列モードの違いを説明する図である。図5(A)の通常モードでは、ブースセクタ12がデコード結果に応じて単一の被乗数の0倍、正の1倍、正の2倍、負の2倍、負の1倍のいずれかを選択して部分積PMを出力する。そして、負の場合は、ビット反転した部分積PMの最下位にホットビットH(+1)を加算すればよい。

40

【0027】

一方、図5(B)の並列モードでは、ブースセクタ12が上位側の要素1と下位側の要素0それぞれをデコード結果に応じて上記と同様に選択する。部分積が負の場合は、各部分積の要素 PM_{-1} 、 PM_{-0} の最下位ビットにホットビットH(+1)を加算する必要がある。したがって、通常モードでは上位側の要素 PM_{-1} の最下位の位置へのホットビットHの加算を行われず、並列モードではその位置へのホットビットHの加算を行う必要がある。図5(B)内に示されたホットビットHである。

【0028】

50

図4に戻り、図5のように通常モードでは上位側の要素1にホットビットの加算は行われず、並列モードでは加算が行われるので、図中に示すとおり、ホットビットセレクト信号Selをワレスツリー20の入力段に供給して、全ての並列モードに対応するビットの位置に設けたホットビット加算回路をイネーブルまたはディセーブルにする制御が必要になる。このような回路構成では、ホットビットセレクト信号Selの伝搬が乗算器全体の動作速度を律則することになり、セレクト信号Selの伝搬路がクリティカルパスになる。また、全ての並列モードに対応するビットの位置にホットビット加算回路を設けることは、回路規模の増大になる。

【0029】

[本実施の形態の乗算回路]

図6は、本実施の形態における乗算回路の概略構成図である。図6の乗算回路は、図1と同様に、乗数Yの組合せをデコードするブースデコーダ11と、デコード結果に応じて被乗数Xと乗数Yの部分積PMを生成するブースセクタ12とを有する部分積生成回路10と、複数の部分積PMを並列に加算するキャリー保存加算器(CSA)をツリー状に配置し、所定段のキャリー保存加算器が出力する加算データとキャリーデータを後段のキャリー保存加算器が加算する部分積加算回路(ワレスツリー加算回路)20, 21を有する。また、ワレスツリー加算回路の最終段のCSA21の後に、加算データSUMとキャリーデータCRYとを加算する全加算器30を有する。

【0030】

図6の乗算回路は、さらに、複数のデータを並列に乗算する並列モードで、上位側の並列データのブースデコード結果に応じて補正加算すべき補正ホットビットを生成する補正ホットビット生成部50を有する。この補正ホットビット生成部50は、ブースデコーダ11のデコード結果に応じて補正キャリーの判定を行って補正キャリー信号を出力する補正キャリー判定回路50_1と、補正キャリー信号に基づいて補正ホットビット信号を生成する補正ホットビット生成回路50_2とを有する。

【0031】

この乗算回路では、第1に、入力される被乗数Xを上位側の並列データ(要素1)と下位側の並列データ(要素0)の並列データに分割し、要素1の下位側を全て0にし、要素0の上位側を全て0にする分割回路30を設け、第2に、ブースセクタ12が生成する要素0の部分積PM_0の最下位ビットと、要素1の部分積PM_1の最下位ビットにホットビットHを加算し、第3に、ワレスツリー加算回路21に入力する上位側の要素1の下位側をゼロにマスクし、下位側の要素0の上位側をゼロにマスクするゼロマスク回路40を設けることで、ワレスツリー加算回路21での上位側と下位側の要素1, 0が混ざってしまい誤った加算結果が出力されることを防止する。

【0032】

このゼロマスク回路40により上位側の要素1の下位側をゼロにマスクすることで、部分積PM_1に加算したホットビットHの一部が消失してしまう。そこで、乗算回路は、この消失するホットビットHを補正するための補正ホットビットをブースデコーダ11のデコード結果に基づいて予め生成する補正ホットビット生成部50_1, 50_2を有する。そして、補正ホットビットCHが上位側の要素1の最下位ビットに加算される。図6の例では、最終段の4to2CSA21と、全加算器22とで加算されている。補正ホットビットCHの加算は、ワレスツリー加算回路20で加算するようにしても良い。但し、補正ホットビットCHは、要素1の32ビットの最下位ビットに加算する必要がある。その結果、ゼロマスク回路40で要素1の下位側がゼロに変換されても加算した補正ホットビットCHが消失することはない。

【0033】

図6の乗算回路について、さらに具体的に説明する。乗算回路は、単精度と倍精度のデータについて乗算することができる。そして、乗算回路は、倍精度のデータを乗算できることを利用して、1組の被乗数Xと乗数Yを乗算する通常モードと、2組の被乗数X_1, X_0と乗数Y_1, Y_0をそれぞれ乗算する並列モードとを、切替可能に構成され

10

20

30

40

50

る。これを一般化すると、乗算回路は、単精度から n 倍精度のデータについて乗算することができ、 n 倍精度のデータを乗算できることを利用して、1組の被乗数 X と乗数 Y を乗算する通常モードと、 m 組の被乗数 X_{-1} 、 X_{-0} と乗数 Y_{-1} 、 Y_{-0} をそれぞれ乗算する並列モードとを、切替可能に構成される。並列モードでのデータは n/m 倍精度である。以下の実施の形態では、入力データは単精度 32 ビットと倍精度 64 ビットのいずれかであり、並列モードでは 32 ビットの 2 組の被乗数 X_{-1} 、 X_{-0} とが入力される例である。

【0034】

図6の乗算回路では、分割回路30が、上位側に要素1の被乗数 X_{-1} が下位側に要素0の被乗数 X_{-0} が格納された入力データを、上位側の要素1の下位側を全て0にし、下位側の要素0の上位側を全て0（または符号ビットS）にして、上位側の要素1のデータと下位側の要素0のデータとに分割する。そして、要素0のデータと要素1のデータがそれぞれブースセクタ12に入力される。

10

【0035】

好ましくは、ブースセクタ12は、要素0のデータを入力し要素0の部分積 PM_{-0} を生成する要素0のブースセクタ12₋₀と、要素1のデータを入力し要素1の部分積 PM_{-1} を生成する要素1のブースセクタ12₋₁とを有する。そして、ブースデコーダ11が、乗数 Y_{-1} 、 Y_{-0} をそれぞれデコードしたデコード結果を出力し、両ブースセクタ12₋₀、12₋₁がデコード結果に応じて要素0の部分積 PM_{-0} 、要素1の部分積 PM_{-1} を同時に出力する。

20

【0036】

図7は、ブースセクタ12の構成を示す図である。この例は、2次のブースアルゴリズムによる構成である。また、図7には、32ビットのデータのうち k ビット目のブースセクタのみが示されている。ブースデコーダ11は、論理値表に示すように、被乗数 Y の3ビットの組合せ $n+1$ 、 n 、 $n-1$ をデコードして、乗数 X の0倍、1倍 ($\times 1$)、2倍 ($\times 2$)、負の2倍 ($-\times 2$)、負の1倍 ($-\times 1$)、0倍のいずれかをデコード結果として出力する。ブースセクタ12は、デコード結果である1倍 ($\times 1$)、2倍 ($\times 2$)、負の2倍 ($-\times 2$)、負の1倍 ($-\times 1$) に対応して、被乗数 $X[k]$ をビットシフトせず出力する ($\times 1$)、左シフトして出力する ($\times 2$)、ビットシフトせずビット反転して出力する ($-\times 1$)、左シフトしてビット反転して出力する ($-\times 2$) のいずれかを選択して、部分積 $PM[k]$ として出力するセクタ121を有する。

30

【0037】

ビット反転された部分積 PM には、最下位ビットにホットビットHとして1が加算される。具体的には、部分積 PM をワレスツリー加算器20の初段の最下位ビットのCSAに加算するホットビット加算回路が設けられる。ビット反転して+1加算することで、負の部分積が2の補数に変換される。

【0038】

図6に戻り、ブースセクタ12₋₀は、要素0の部分積 PM_{-0} を生成し、ブースセクタ12₋₁は、要素1の部分積 PM_{-1} を生成する。乗数 Y_{-0} 、 Y_{-1} がそれぞれ32ビットとすると、2次のブースアルゴリズムにより部分積の数は $32/2 = 16$ 個になる。ここで注意すべき点は、要素0の部分積 PM_{-0} も要素1の部分積 PM_{-1} も、最下位ビットにホットビットH (= +1) が加算されることである。特に、分割回路20が上位側の要素1の下位側を全て0にマスクしたため、ブースセクタ12₋₁が負の部分積をセレクトした場合、要素1の下位側が全て1にビット反転される。その結果、要素1のデータについても64ビットの最下位ビットにホットビットを加算することで、上位側にある要素1の32ビットのデータにホットビットが伝搬する。つまり、要素0の部分積 PM_{-0} も要素1の部分積 PM_{-1} も同様に、64ビットの最下位ビットにホットビットを加算する構成をワレスツリー加算器20の入力部に設ければよく、通常モードと並列モードとで部分積 PM に対するホットビットの加算回路を同じ構成にできる。

40

【0039】

50

次に、乗算回路は、2組のワレスツリー加算器 20__1, 20__0 を有する。このワレスツリー加算器 20__1, 20__0 は、それぞれ3段の4 to 2 CSAで構成される。したがって、上位側のワレスツリー加算器 20__1 は、16個の要素1の部分積 PM__1 を入力し2個の部分積 PM3__1 を出力する。下位側のワレスツリー加算器 20__0 も同様である。

【0040】

図8は、ワレスツリー加算器 20__1, 20__0 と、ワレスツリー加算器が内蔵する4 to 2 CSA との回路構成を示す図である。図8の上側に示した4 to 2 CSA は、3つの入力ビットからキャリーアウトビットCO, キャリービットCと、加算ビットSを出力するキャリー保存加算器 CSA3 を2個有する。そして、一方の CSA3 は入力ビット A2, A3, A4 からキャリーアウトビットCOと加算ビットSを出力し、もう一方の CSA3 は下位ビットからのキャリーインビットCIと入力ビットA1と CSA3 からの加算ビットを入力し、下段へのキャリービットCと加算ビットSとを出力する。各キャリーアウトビットCOとキャリービットCと加算ビットSの論理式が図8に示されている。

10

【0041】

4 to 2 CSA は、入力データのビット数だけ横方向に配列され、下位ビットからのキャリーアウトCOが上位ビットにキャリーインCIとして入力されるが、最下位ビットからのキャリーが最上位ビットまで伝搬することはない。これが CSA の演算時間が短い理由である。

【0042】

20

更に、図8にはワレスツリー加算器 20__1, 20__0 の内部構成が示される。初段の4つの4 to 2 CSA には、16個の部分積が4グループに分けて入力される。つまり、初段の4つの4 to 2 CSA がそれぞれ4つの部分積を有するグループA, B, C, Dを入力する。更に、初段の4つの4 to 2 CSA がそれぞれ出力する4組のキャリーデータCと加算データSとを、次段の2つの4 to 2 CSA がそれぞれ入力し、キャリーデータCと加算データSとをそれぞれ出力する。そして、3段目の1つの4 to 2 CSA は2組のキャリーデータCと加算データSを入力し、キャリーデータCと加算データSとを出力する。

【0043】

図6に戻り、上記の通り、4 to 2 CSA は、4入力A1 - A4 に対して加算データSとキャリーデータCを出力するので、1段の4 to 2 CSA でデータの数半減する。したがって、3段の4 to 2 CSA によりデータ数が $1/2^3 = 1/8$ に減ることになる。よって、ワレスツリー加算器 20__1 が、16個の部分積 PM__1 から2個の部分積 PM3__1, つまり加算データSUMとキャリーデータCRYを生成する。同様に、下位側のワレスツリー加算器 20__0 は、16個の要素0の部分積 PM__0 を入力し2個の部分積 PM3__0, つまり加算データSUMとキャリーデータCRYを出力する。

30

【0044】

2組のワレスツリー加算器 20__1, 20__0 が、上位側の要素1の16個の部分積 PM__1 と下位側の要素0の16個の部分積 PM__0 をそれぞれ加算するので、それぞれのワレスツリー加算器 20__1, 20__0 では、要素1の部分積 PM__1 と要素0の部分積 PM__0 とが混ざり合うことはない。

40

【0045】

さらに、キャリー保存加算器 CSA は、各桁で発生したキャリーを全て伝搬せずに保存して加算データとキャリーデータを出力し、次の段のキャリー保存加算器 CSA に入力する。したがって、ワレスツリー加算器 20__1, 20__0 が出力する部分積 PM3__1, PM3__0 内には加算したホットビットHの伝搬によるキャリーが含まれている。

【0046】

そして、2つの部分積 PM3__1 と2つの部分積 PM3__0 とを合わせて計4つのデータが、最終段の4 to 2 CSA 21 に入力されると、上位側の要素1と下位側の要素0とが混じり合うことになる。そこで、乗算回路は、上位側の要素1の下位ビットを0に置き

50

かえ，下位側の要素0の上位ビットを0に置きかえるゼロマスク回路40を有する。このゼロマスク回路によるゼロへの置きかえにより，上位側の要素1の下位ビットに含まれていたホットビットの伝搬によるキャリーが消失される。

【0047】

そこで，ゼロマスク回路40により消失されたホットビットを，上位側の要素1のデータに加算する必要がある。この追加すべきホットビットを補正ホットビットと称する。図6の乗算回路は，ブースデコーダ11による上位側の要素1の乗数 Y_{-1} のデコード結果に応じて，補正すべきキャリーを判定する補正キャリー判定回路50_1と，補正キャリー信号に基づいて補正ホットビットを生成する補正ホットビット生成回路50_2とを有する。補正キャリー判定回路50_1と補正ホットビット生成回路50_2により，補正ホットビットCHを生成する補正ホットビット生成部が構成される。

10

【0048】

上記の補正ホットビットCHを上位側の要素1の最下位ビットに加算することで，ゼロマスク回路40により消失されたホットビットを補うことができる。補正ホットビットCHを加算する回路は，図6の例では，最終段の4to2CSAの入力部または全加算器22の入力部に設けられる。後述するとおり，3段の4to2CSAの例では，補正ホットビットCHは2，1，0のいずれかである。そこで，図6の例では，補正ホットビットCHが最大値2の場合は，最終段の4to2CSAの入力部と全加算器22の入力部で+1ずつ加算され，補正ホットビットCHが1の場合は最終段の4to2CSAの入力部または全加算器22の入力部のいずれかで加算される。

20

【0049】

図6の乗算回路は，最終段の4to2CSA21が出力する部分積PM5の加算データSUMとキャリーデータCRYを，全加算器22が加算して乗算値データMPを出力する。部分積PM5は，上位側に要素1のデータを，下位側に要素0のデータをそれぞれ有する。そして，全加算器22が，部分積PM5の加算データSUMとキャリーデータCRYとを加算して乗算値データMPを出力する。全加算器22はキャリーを全て伝搬させて入力データを加算するので，乗算値データMPにはブースセクタ12の出力に加算したホットビットHが全て反映される。

【0050】

[補正ホットビットの生成アルゴリズム]

30

次に，図6の乗算回路内の補正キャリー判定回路50_1と補正ホットビット生成回路50_2による補正ホットビットの生成アルゴリズムについて説明する。

【0051】

図6の乗算回路では，分割回路30が要素1の下位側を全て0にし，要素0の上位側を全て0または符号ビットSにする。したがって，要素1の下位側を全て0にしてブースセクタ12_1に入力した結果，ブースセクタ12_1によりセレクトされる要素0の下位側は，正の部分積がセレクトされると全て0になり，負の部分積がセレクトされると全て1になる。この結果，ワレスツリー加算器20_1の入力と出力との間にはある規則性が生じる。

【0052】

40

図9は，4to2CSAの4入力の全パターンに対する出力S，Cとキャリーの関係を示す図である。入力される要素1の下位側は全て0か全て1かである。したがって，4to2CSAの4入力についての組合せは16通りとなる。4to2CSAは組み合わせ回路であるので，4つの入力と2つの出力の組合せは一意に決まり，4つの入力が0000のときは出力S，Cは00，4つの入力が1111のときは出力S，Cは11，それ以外のときは出力S，Cは10または01である。

【0053】

また，ワレスツリー加算器では，4to2CSAが，出力C，S以外にキャリーアウトCOを出力し上位ビットに伝搬する。要素1の下位側のビットにおけるキャリーCとキャリーアウトCOは，負の部分積の場合に加算したホットビットHの伝搬そのものである。

50

したがって、キャリアCとキャリアアウトCOが発生した数は、ホットビットHが伝搬した数になる。以上を前提にして、4 to 2 CSAの入力と出力の組合せ、キャリアC、COが発生した数、ホットビット伝搬のために発生すべきキャリアの数について説明する。

【0054】

なお、図9では、便宜上、要素1の下位側の32ビットのうち任意の3ビットだけを示している。また、図9では、1段の4 to 2 CSAにおいて発生したキャリアの数と発生すべきキャリアの数とそれらの差分とを示している。1段の4 to 2 CSAの法則性が理解できれば、3段またはN段（Nは複数）の4 to 2 CSAによるワレスツリー加算器において発生したキャリアの数と発生すべきキャリアの数とそれらの差分とを知ることができる。

10

【0055】

(1) パターンZは、4入力A1 - A4が全て0の例である。図8の論理式に示したとおり、入力A2、A3、A4からキャリアアウトCOが生成され、キャリアアウトCOは上位ビットでキャリアインCIとなる。そして、入力A2、A3、A4から生成された加算ビットと入力A1とキャリアインCI (= CO) からキャリアビットCと加算ビットSとが生成される。4入力A1 - A4が全て0の場合は、出力S、Cは00となる。また、生成されたキャリアCは0、キャリアアウトCOは0であるので、発生したキャリアCRYも0である。そして、発生すべきキャリアCRYの数は、入力A1 - A4の1の数に等しいので、0である。つまり、入力A1 - A4の下位側が全て1の場合は負の部分積が選択されてホットビットHが加算されているからである。上記から、発生すべきキャリアCRYの数から発生したキャリアCRYの数を減算した差分は、0 - 0 = 0である。つまり、入力A1 - A4が全て0の場合は、ゼロマスク回路40により失われるホットビットの数は、上記の差分の0になることが理解できる。

20

【0056】

(2) パターンX-1は、4入力A1 - A4のうち1つが1で残り3つが0の例である。この場合の出力S、Cは10であり、発生したキャリアCRYの数(CO + C)は0、発生すべきキャリアCRYの数(入力の1の数)は1となる。したがって差分は1になる。4種類の入力の組合せのいずれも同じ結果になる。

【0057】

(3) パターンX-2は、4入力A1 - A4のうち2つが1で残り2つが0の例である。この場合の出力S、Cは10または01であり、発生したキャリアCRYの数(CO + C)は1、発生すべきキャリアCRYの数(入力の1の数)は2となる。したがって差分は1になる。

30

【0058】

(4) パターンX-3は、4入力A1 - A4のうち3つが1で残り1つが0の例である。この場合の出力S、Cは01であり、発生したキャリアCRYの数(CO + C)は2、発生すべきキャリアCRYの数(入力の1の数)は3となる。したがって差分は1になる。

【0059】

(5) パターンFは、4入力A1 - A4がすべて1の例である。この場合の出力S、Cは11であり、発生したキャリアCRYの数(CO + C)は2、発生すべきキャリアCRYの数(入力の1の数)は4となる。したがって差分は2になる。

40

【0060】

上記の4入力パターンに対する出力S、Cには、加算したホットビットは反映されていない。

【0061】

図10は、図9に示した5つのパターンの発生したキャリアCRY、発生すべきキャリアCRY、それらの差分をまとめた図である。図10には、図9と同じように、5つのパターンZ、X-1、X-2、X-3、Fに対する出力S、Cの組合せが示され、図10内の表には、5つのパターンに対する発生したキャリアCRY、発生すべきキャリアCRY

50

、それらの差分が示されている。

【 0 0 6 2 】

そこで、図 1 0 にまとめた入力パターンと、出力 S 、 C と、発生したキャリー CRY と、発生すべきキャリー CRY と、それらの差分（消失で不足するホットビットの数）に基づいて、3 段の $4 \text{ to } 2 \text{ CSA}$ のワレスツリー加算器における不足するホットビットの数について、以下で検討する。

【 0 0 6 3 】

図 1 1 は、ワレスツリー加算器により伝搬するキャリーと不足するホットビットの数の一例を示す図である。図 1 1 は、一例として、ワレスツリー加算器に入力する 4 組の 4 入力 A 、 B 、 C 、 D が、グループ A ($X - 3$)、グループ B ($X - 1$)、グループ C ($X - 2$)、グループ D (F) の場合において、3 段の $4 \text{ to } 2 \text{ CSA}$ で伝搬するキャリーを示している。グループ $A - D$ と図 1 0 のパターン Z 、 $X - 1$ 、 $X - 2$ 、 $X - 3$ 、 F との関係は、括弧内に示したとおりである。

【 0 0 6 4 】

図 1 1 において、1 段目の 4 つの $4 \text{ to } 2 \text{ CSA}$ で発生したキャリー CRY 、発生すべきキャリー CRY 、それらの差分は、図示されるとおりである。したがって、1 段目で発生したキャリー CRY 、発生すべきキャリー CRY 、差分は、4 つの $4 \text{ to } 2 \text{ CSA}$ のキャリー CRY 、発生すべきキャリー CRY 、差分の数を合計した数になり、図示されるとおり、5、10、5 である。

【 0 0 6 5 】

次に、2 段目の 2 つの $4 \text{ to } 2 \text{ CSA}$ には、グループ A 、 B それぞれの出力 S 、 C からなる 4 つのデータ ($X - 2$) と、グループ C 、 D それぞれの出力 S 、 C からなる 4 つのデータ ($X - 3$) とが入力される。図 1 0 によれば、それぞれの $4 \text{ to } 2 \text{ CSA}$ で発生するキャリー CRY の数は 1、2 であるので、2 段目で発生したキャリー CRY の合計は 3、1 段目で残っていたキャリー（差分）が 5 だったため、2 段目での残りのキャリー CRY の数を示す差分は、 $5 - 3 = 2$ になる。

【 0 0 6 6 】

そして、3 段目の 1 つの $4 \text{ to } 2 \text{ CSA}$ には、グループ A 、 B の出力 S 、 C とグループ C 、 D の出力 S 、 C からなる 4 つのデータ ($X - 2$) が入力される。図 1 0 によれば、 $4 \text{ to } 2 \text{ CSA}$ で発生したキャリー CRY の数は 1 であるので、2 段目で残っているキャリー（差分）の数 2 から 3 段目で発生したキャリー CRY の数 1 を減じると、3 段目での残りのキャリー CRY の数を示す差分は、 $2 - 1 = 1$ になる。

【 0 0 6 7 】

図 1 1 によれば、要素 1 の入力の組合せがパターン $X - 3$ 、 $X - 1$ 、 $X - 2$ 、 F の場合は、3 段のワレスツリー加算器の出力 S 、 C に残っているキャリー CRY の数は 1 になるので、補正ホットビット CH は 1 になる。

【 0 0 6 8 】

ワレスツリー加算器 1 2 は、 $4 \text{ to } 2 \text{ CSA}$ が 3 段積まれていて、各グループの入力データは $2^4 = 16$ 種類あるので、図 1 1 により予測される補正ホットビットの種類は膨大な数になる。しかしながら、 $4 \text{ to } 2 \text{ CSA}$ の出力が入力パターンによって一意に決まること、その出力が次段の $4 \text{ to } 2 \text{ CSA}$ の入力となる。したがって、図 1 1 に示した 5 つのパターン Z 、 $X - 1$ 、 $X - 2$ 、 $X - 3$ 、 F に対する出力 S 、 C に基づいて、3 段のワレスツリー加算器において発生する補正ホットビットの規則性は以下の通りとなる。

【 0 0 6 9 】

図 1 2 は、3 段のワレスツリー加算器の各段の $4 \text{ to } 2 \text{ CSA}$ の入力パターンを示す図である。まず、 $4 \text{ to } 2 \text{ CSA}$ の入出力の組合せの規則に基づいて、3 段のワレスツリー加算器の各段の $4 \text{ to } 2 \text{ CSA}$ の入力パターンを検討する。

【 0 0 7 0 】

(1) CSA の入力がパターン Z であれば出力 $SC = 00$ であるので、4 グループの入力 $A - D$ が全てパターン Z の場合は、2 段目の入力パターンも全て Z になり、3 段目の入

10

20

30

40

50

カパターンもZになる。つまり，入力が全てパターンZの場合は3段目の入力パターンはZになり，その出力SC = 00になる。

【0071】

(2) CSAの入力がパターンFであれば出力SC = 11であるので，4グループの入力A - Dが全てパターンFの場合は，2段目の入力パターンも全てFになり，3段目の入力パターンもFになる。つまり，入力が全てパターンFの場合は3段目の入力パターンはFになり，その出力SC = 11になる。

【0072】

(3) CSAの入力がパターンXであれば出力SC = 01または10であるので，ワレスツリーのどこかでパターンXの入力が発生すると，その先の入力はパターンXに収束する。そして，その場合3段目の出力SC = 01または10になる。

10

【0073】

図13，図14は，入力パターンの組合せ例に対する1段目での差分（不足数），2，3段目で発生したキャリーCRY，最終的に残った数（補正值）を示す図である。図10の表を参照して，図13，図14の5つの例について説明する。

【0074】

(1) 4グループの入力がパターンXのみの例である。この場合，1段目の4つのCSAでのキャリーの不足数（差分）は合計4である。2段目以降の入力はパターンXに収束し，発生するキャリー数は合計3となる。したがって，1段目のキャリーの不足数4に対して2，3段目のキャリー発生数3であるので，補正すべきキャリー数は1になる。つまり，補正ホットビットは1になる。

20

【0075】

(2) 4グループの入力にパターンZが含まれる例である。この場合，パターンZが入力の場合のキャリーの不足数（差分）は0であるので，3つのパターンXにより，1段目の4つのCSAでのキャリーの不足数（差分）は合計3である。2段目では，パターンZとXの組合せではパターンX - 1となりキャリーが発生せず，パターンXとXの組合せではパターンX - 2となりキャリーが1発生し，発生するキャリー数は合計1となる。さらに，3段目で発生するキャリー数は1である。したがって，1段目のキャリーの不足数3に対して2，3段目のキャリー発生数2であるので，(1)と同様に，補正すべきキャリー数は1になる。つまり，補正ホットビットは1になる。

30

【0076】

(3) 4グループの入力にパターンFが含まれる例である。この場合，パターンFが入力の場合のキャリーの不足数（差分）は2であるので，3つのパターンXによる不足分の3を加えて，1段目の4つのCSAでのキャリーの不足数（差分）は合計5である。2段目では，パターンXとFの組合せではパターンX - 3となりキャリーが2発生し，パターンXとXの組合せではパターンX - 2となりキャリーが1発生し，発生するキャリー数は合計3となる。さらに，3段目で発生するキャリー数は1である。したがって，1段目のキャリーの不足数5に対して2，3段目のキャリー発生数が4であるので，(1)(2)と同様に，補正すべきキャリー数は1になる。つまり，補正ホットビットは1になる。

【0077】

(4) 4グループの入力が全てパターンZの例である。この場合は，1段目の不足数は0，2，3段目のキャリー発生数は0，その結果補正すべきキャリー数も0になる。

40

【0078】

(5) 4グループの入力が全てパターンFの例である。この場合は，ツリーの入力は全てパターンFになる。よって，1段目で不足する数（差分）は $2 \times 4 = 8$ ，2段目で発生するキャリー数は $2 \times 2 = 4$ ，3段目で発生するキャリー数は2である。したがって，補正すべきキャリー数は2となる。

【0079】

上記の法則は，ワレスツリー加算器が3段構成に限らず，2段または4段以上の構成であっても適用される。したがって，入力データである部分積の数にかかわらず上記の法則

50

は適用できる。

【 0 0 8 0 】

上記の 5 つの例をまとめると、入力にパターン X が含まれる場合は、補正值（不足するキャリー数、差分）は 1、入力が全てパターン Z の場合は 0、入力が全てパターン F の場合は 2 になる。入力のパターン X、Z、F は 1 の数に基づいており、入力が 1 になるのはブースデコーダのデコード値が負の部分積を選択しビット反転した場合、つまりデコード値が $-x_1$ 、 $-x_2$ の場合である。

【 0 0 8 1 】

したがって、この法則を利用すれば、補正すべきキャリー数、つまり補正ホットビットの数は、ブースデコーダのデコード値が $-x_1$ または $-x_2$ になる数に基づいて判定することができる。すなわち、入力にパターン X が含まれるか否かは、16 個のブースデコード値のうち負を示す $-x_1$ 、 $-x_2$ がひとつでもあるかどうかで判定する。入力が全てパターン F か否かは、16 個の全てのブースデコード値が $-x_1$ 、 $-x_2$ のどちらかであるかどうかで判定する。入力が全てパターン Z になるか否かは、16 個のブースデコード値が $-x_1$ 、 $-x_2$ のいずれにもならないかどうかで判定する。

10

【 0 0 8 2 】

図 15 は、補正ホットビット生成部 50 の回路図である。補正ホットビット CH [1 : 0] は、以下のようにして生成される。まず、補正キャリー判定回路 50__1 は、32 ビットの乗数 Y__1 の 3 ビットの組合せをデコードする 16 個のブースデコーダ 11__1 ~ 11__16 それぞれのブースデコード値が $-x_1$ 、 $-x_2$ のいずれかであることを検出する OR ゲート 51__1 ~ 51__16 と、これら 16 個の OR ゲート 51 の出力を入力して第 1 の補正キャリー CRY__1 を出力する OR ゲート 52 と、同じ 16 個の OR ゲート 51 の出力を入力して第 2 の補正キャリー CRY__2 を出力する AND ゲート 53 とを有する。

20

【 0 0 8 3 】

OR ゲート 52 が出力する第 1 の補正キャリー CRY__1 は、入力にパターン X が一つでも含まれるか否かを示す。したがって、第 1 の補正キャリー CRY__1 = 1 であれば、補正ホットビットは 1 になる。また、第 2 の補正キャリー CRY__2 は、入力全てがパターン F であるか否かを示す。したがって、第 2 の補正キャリー CRY__2 = 1 であれば、補正ホットビットは 2 になる。

30

【 0 0 8 4 】

補正ホットビット生成回路 50__2 は、インバータ 54 と AND ゲート 55 とを有する。上記したとおり、2 ビットの補正ホットビット CH [1 : 0] は、次の通りである。

$$CH [0] = CRY_1 * (not CRY_2)$$

$$CH [1] = CRY_2$$

これにより、補正ホットビット CH [1 : 0] は、00、01、10 のいずれか、つまり補正ホットビット数 0、1、2 いずれかになる。

【 0 0 8 5 】

図 6 に示したとおり、補正ホットビット生成ユニット 50 は、ワレスツリー加算器 21 の動作と並行して行うことができる。そして、補正ホットビットは、ワレスツリー加算器 21 以降のどこかで要素 1 の最下位ビットに加算するようにすればよい。この結果、補正ホットビットを加算することによる、本来のブースアルゴリズムによる部分積生成回路とワレスツリー加算器の動作に遅延の影響を与えることはない。

40

【 0 0 8 6 】

[乗算器の具体的構成]

図 16、図 17 は、本実施の形態における乗算回路の具体的な構成を示す図である。図 6 の乗算回路の具体例である。この乗算回路は、単精度 32 ビットまたは倍精度 64 ビットの乗算を行う通常モードと、2 つの 32 ビットの並列データを並列に乗算する並列モードとを有する。図 16、17 には、並列モードでの並列データが示され、図 6 と同様に並列データが上位側の要素 1 と下位側の要素 0 で構成される。図 6 と同じ構成には同じ引用

50

番号を付与している。

【 0 0 8 7 】

図 1 6 において、被乗数として要素 1 の被乗数 X_{1} と要素 0 の被乗数 X_{0} とが入力される。また、乗数として要素 1 の乗数 Y_{1} と要素 0 の乗数 Y_{0} とが入力される。要素 1, 0 の乗数、被乗数はいずれも 3 2 ビット構成である。

【 0 0 8 8 】

乗算回路は、要素 1 と要素 0 を分割する分割回路 3 0 を有する。分割回路 3 0 は、要素 1, 0 の並列データを、上位側を要素 1 の 3 2 ビットデータに下位側を全て 0 にした要素 1 の被乗数データ X_{1} と、上位側を全て 0 (または符号ビット S) に下位側を要素 0 の 3 2 ビットデータにした要素 0 の被乗数データ X_{0} とに分割する。

10

【 0 0 8 9 】

図 1 8 は、分割回路の一例を示す図である。分割回路 3 0 は、並列モードで並列モード信号 $MODE$ が 1 に制御される場合に、入力される被乗数 X のうち要素 1 のデータを上位ビット [6 3 : 3 2] に下位ビット [3 1 : 0] を 0 にし、要素 0 のデータを下位ビット [3 1 : 0] に上位ビット [6 3 : 3 2] を 0 にして、分割後の要素 1 のデータ X_{1} 、要素 0 のデータ X_{0} を出力する。

【 0 0 9 0 】

図 1 6 に戻り、ブースセクタ 1 2 $_{1}$ は、分割後の要素 1 のデータ X_{1} を入力し、ブースデコーダ 1 1 による要素 1 の乗数 Y_{1} のデコード値に応じて、部分積 PM_{1} を出力する。同様に、ブースセクタ 1 2 $_{0}$ は、分割後の要素 0 のデータ X_{0} を入力し、ブースデコーダ 1 1 による要素 0 の乗数 Y_{0} のデコード値に応じて、部分積 PM_{0} を出力する。好ましくは、ブースデコーダ 1 1 は、要素 1, 0 の乗数 Y_{1} 、 Y_{0} それぞれの 1 6 通りの 3 ビットを同時にデコードし、それぞれ 1 6 のデコード値を出力する。そして、好ましくは、ブースセクタ 1 2 $_{1}$ 、1 2 $_{0}$ も、要素 1, 0 それぞれの 1 6 のデコード値に応じて、それぞれ 1 6 個の部分積 PM_{1} 、 PM_{0} を同時に出力する。

20

【 0 0 9 1 】

図 1 9 は、ブースデコーダ 1 1 とブースセクタ 1 2 $_{1}$ 、1 2 $_{0}$ の構成を示す図である。図 1 9 には、ブースデコーダ 1 1 の 1 組のデコード値 ($\times 1$ 、 $\times 2$ 、 $-\times 2$ 、 $-\times 1$) に対するブースセクタ回路が示されている。ブースセクタ 1 2 $_{\#}$ は、6 4 ビットの入力データからデコード値に基づいて選択した 6 4 ビットのデータを出力する。そして、図 1 9 内の 1 ビット分のブースセクタ回路は、図 7 に示した回路と同じである。

30

【 0 0 9 2 】

前述したとおり、好ましい例では、ブースデコーダ 1 1 が要素 1, 0 の乗数 Y_{1} 、 Y_{0} に対してそれぞれ 1 6 個のデコード値を出力する。したがって、図 1 6 の好ましいブースデコーダ 1 1 とブースセクタ 1 2 は、図 1 9 に示した回路を 1 6 個 $\times 2 = 3 2$ 個有する。

【 0 0 9 3 】

ブースセクタ 1 2 $_{\#}$ は、デコード値が $-\times 1$ 、 $-\times 2$ のいずれかの場合に、被乗数 X_{1} 、 X_{0} のビットを反転して出力する。その場合は、反転されたデータにホットビットとして 1 を加算して 2 の補数を生成する必要がある。そのために、図 1 9 の回路は、ブースセクタ 1 2 $_{\#}$ が出力する部分積 $PM_{\#}$ にホットビットを加算する回路 6 0 を有する。このホットビット加算回路 6 0 は、デコード値が $-\times 1$ 、 $-\times 2$ のいずれかの場合に 1 を出力する OR 回路である。そして、部分積 $PM_{\#}$ とホットビット加算回路 6 0 が出力するホットビット H とが、ワレスツリー加算器 2 0 に入力され、ホットビットが加算される。つまり、ホットビット加算回路 6 0 は、ホットビット H をワレスツリー加算器に入力し加算させる。

40

【 0 0 9 4 】

図 1 6 に戻り、ブースセクタ 1 2 $_{1}$ 、1 2 $_{0}$ が要素 1, 0 の部分積 PM_{1} 、 PM_{0} をそれぞれ出力する。部分積 PM_{1} 、 PM_{0} は、それぞれ 1 6 個の部分積を有する。好ましくは、ブースセクタ 1 2 $_{1}$ 、1 2 $_{0}$ が要素 1, 0 それぞれの 1 6 個の

50

部分積を同時に出力する。そして、要素1, 0それぞれの16個の部分積 $PM_{1,0}$ は、ワレスツリー加算器20に入力され、図示しないホットビットと共に加算される。ワレスツリー加算器20は、要素1, 0それぞれの16個の部分積 $PM_{1,0}$ を、最初に異なるワレスツリー20_1, 20_0で加算してから、その加算結果を共通のワレスツリー21のCSAで加算する。図16には、ワレスツリー加算器20の最終段の全加算器22が示されている。

【0095】

図17には、図16のワレスツリー加算器20の構成が示されている。ワレスツリー加算器20は、要素1の部分積 PM_1 を入力して加算する第1のワレスツリー加算器20_1と、要素0の部分積 PM_0 を入力して加算する第2のワレスツリー加算器20_0と、ゼロマスク回路40_1, 40_0と、共通の第3のワレスツリー加算器21と、全加算器22_1, 22_0とを有する。

10

【0096】

第1, 第2のワレスツリー加算器20_1, 20_0は、それぞれ3段の4to2CSAを有し、それぞれ16個の部分積 $PM_{1,0}$ を加算して、それぞれ加算データSとキャリデータCを有する部分積 PM_3 を出力する。つまり、16入力から2出力が生成される。したがって、第1, 第2のワレスツリー加算器20_1, 20_0では、要素1と要素0の部分積 PM_1, PM_0 が混ざり合うことはない。

【0097】

ワレスツリー加算器では、更に、それぞれ加算データSとキャリデータCを有する部分積 PM_3 の4つのデータを一緒に加算する第3のワレスツリー加算器21を有する。但し、この第3のワレスツリー加算器21では、要素1, 0の部分積 PM_3 が混ざり合う。そこで、マスク回路40_1が要素1の部分積 PM_3 の下位側を0に変換し、マスク回路40_0が要素0の部分積 PM_3 の上位側を0に変換し、ワレスツリー加算器21がそれぞれゼロマスクされた要素1, 0の部分積 PM_4 を加算する。ワレスツリー加算器21は1段の4to2CSAを有し、要素1, 0のそれぞれ加算データSとキャリデータCとを有する部分積 PM_4 を加算して、加算データSとキャリデータCを有する部分積 PM_5 を出力する。

20

【0098】

ワレスツリー加算器21では、要素1の部分積 PM_4 と要素0の部分積 PM_4 が4to2SAにより加算されるが、対応するビットでは要素1または要素0のデータしかないので、両要素1, 0のデータが混ざり合うことはない。

30

【0099】

しかし、ゼロマスク回路40-1で下位側がゼロにマスクされた要素1のデータからは残っているキャリが消失する。そこで、補正ホットビット $CH[1:0]$ が、ワレスツリー加算器21と全加算器22_1に入力され加算される。例えば、補正ホットビット $CH[1:0] = 01$ であれば、例えばワレスツリー加算器21にのみ1が加算され、 $CH[1:0] = 10$ であれば、ワレスツリー加算器21と全加算器22_1にそれぞれ1が加算される。

40

【0100】

そして、全加算器22が出力する乗算値PMは要素1の乗算データと要素0の乗算データを含み、要素1の乗算データはホットビットが補正されている。

【0101】

図20は、ワレスツリー加算器の構成を示す図である。ワレスツリー加算器は、要素1, 0のそれぞれ16個の部分積 $PM_{1,0}$ を別々に加算する第1, 第2のワレスツリー加算器20_1, 20_0と、要素1, 0の部分積を合わせて加算する第3のワレスツリー加算器21と、全加算器22とを有する。そして、第1, 第2のワレスツリー加算器20_1, 20_0と第3のワレスツリー加算器21との間に、要素1, 0の下位側と上位側をそれぞれゼロマスクするゼロマスク回路40_1, 40_0を有する。さらに

50

、第3のワレスツリー加算器21の入力と、全加算器22の入力に補正ホットビットを入力する補正ホットビット加算回路62を有する。

【0102】

第1のワレスツリー加算器20__1は、前述のとおり、3段の4to2CSAを有し、初段は4グループA~Dの入力をそれぞれ加算する4組の4to2CSAを有し、2段目はグループA、Bそれぞれの加算データとキャリーデータを加算し、グループC、Dそれぞれの加算データとキャリーデータを加算する2組の4to2CSAを有し、3段目はグループABとCDそれぞれの加算データとキャリーデータを加算する1組の4to2CSAを有する。

【0103】

マスク回路40__1、40__0は、並列モード信号MODE=1の場合にANDゲートにより要素1の下位側のビットを0に変換し、要素0の上位側のビットを0に変換する。

【0104】

そして、第3のワレスツリー加算器21は、要素1の加算データSとキャリーデータC及び要素2の加算データSとキャリーデータCを有し、それぞれゼロマスクされた部分積PM4__1、PM4__0を加算して、加算データとキャリーデータを有する部分積PM5出力する。この部分積PM5の加算データとキャリーデータは、要素1、0のデータを上位側と下位側に有する。最後に、全加算器22が部分積PM5の加算データとキャリーデータを全加算して、乗算データMPを出力する。

【0105】

補正ホットビット加算回路62は、並列モード信号MODE=1で補正ホットビットCH[1:0]を入力するマルチプレクサMUXを有する。補正ホットビット加算回路62は、CH[1:0]=01、10の場合に、ORゲートの出力「1」を第2のワレスツリー加算器21の要素1の最下位の4to2CSAに入力して+1加算し、CH[1:0]=10の場合のみ全加算器22の要素1の最下位の加算器に「1」を入力して+1加算する。したがって、補正ホットビット加算回路62は、実際には補正ホットビットを第3のワレスツリー加算器21と全加算器22の入力に供給し、加算させている。図19で説明したホットビット加算回路60が、第1のワレスツリー加算器20__0の入力にホットビットを供給しているのと同様である。

【0106】

[n=4, m=2の乗算回路]

キャリー補正值である補正ホットビットの予測は、4to2CSAの入出力組合せで一意に決まるので、倍精度の数nと並列数mを変えても同様に予測することができる。したがって、本実施の形態は、たとえば、n=4, m=2とし、2次のブースアルゴリズムを利用した乗算回路に適用することができる。この場合は、データ幅は128ビット、要素の幅は64ビットになる。その結果、ブースセクタが出力する部分積の数は、要素1、0それぞれに32個になる。したがって、第1、第2のワレスツリー加算器20__1、20__0は、それぞれ、入力数が8グループになり、4段構成になる。そして、図15の補正ホットビット生成部の補正キャリー判定回路は、32個のブースデコードの出力を入力する構成になる。

【0107】

この場合でも、入力にパターンXが含まれる場合は補正值(差分)は1、全てパターンZの場合は補正值は0、全てパターンFの場合は補正值は2になる。

【0108】

[3次のブースアルゴリズム]

3次以上のブースアルゴリズムを利用した乗算回路にも、本実施の形態を適用することができる。再び、n=2, m=2の例で説明する。

【0109】

図21は、3次のブースアルゴリズムの場合のブースデコード表である。3次のブースデコードは、乗数の4ビットの組合せをデコードして0, x1, x2, x3, x4, -x

10

20

30

40

50

4, $- \times 3$, $- \times 2$, $- \times 1$ のいずれかをデコード値として出力する。そして、デコード値が $- \times 4$, $- \times 3$, $- \times 2$, $- \times 1$ の場合は、ビット反転をしてホットビットを加算する。したがって、図7, 19のブースセクタは、デコード値が $\times 3$, $\times 4$, $- \times 4$, $- \times 3$ の場合のビットシフト回路や加算回路が必要になる。

【0110】

図22は、3次のブースアルゴリズムを使用した場合の補正ホットビット生成ユニットを示す図である。図示されるとおり、補正キャリア判定回路50_1のORゲート51_1~51_16は、4つのデコード値 $- \times 4$, $- \times 3$, $- \times 2$, $- \times 1$ の論理和を出力する。それ以外の構成は、図15と同じである。

【0111】

以上説明したとおり、本実施の形態の乗算回路によれば、ブースアルゴリズムの部分積生成回路とワレスツリー加算器に、並列モードの場合に上位側の要素1にホットビットを加算する回路を設ける必要がなく構成が簡単になる。また、ワレスツリー加算器により消失するホットビットを簡単な回路で補正することができる。よって、回路規模を大幅に増大することなく乗算回路の演算速度が向上する。

【0112】

以上の実施の形態をまとめると、次の付記のとおりである。

【0113】

(付記1)

乗数の組合せをデコードするブースデコーダと、デコード結果に応じて被乗数と前記乗数の部分積を生成するブースセクタとを有する部分積生成回路と、

複数の前記部分積を並列に加算するキャリア保存加算器をツリー状に配置し、所定段の前記キャリア保存加算器が出力する加算データとキャリアデータを後段の前記キャリア保存加算器が加算する部分積加算回路と、

複数のデータを並列に乗算する並列モードで、上位側の並列データのデコード結果に応じて補正加算すべき補正ホットビットを生成する補正ホットビット生成部とを有し、

前記部分積加算回路は、前記並列モードで、下位側の並列データを入力し第1の加算データ及び第1のキャリアデータを生成する第1のキャリア保存加算器と、上位側の並列データを入力し第2の加算データ及び第2のキャリアデータを生成する第2のキャリア保存加算器と、前記第1の加算データ及び第1のキャリアデータと前記第2の加算データ及び第2のキャリアデータとを加算する第3のキャリア保存加算器と、前記並列モードで、前記上位側の並列データに前記補正ホットビットを加算する補正ホットビット加算回路を有する乗算回路。

【0114】

(付記2)

さらに、

前記並列モードで、前記上位側の要素データの下位側の桁を0にして前記部分積生成回路に入力される被乗数の上位側の要素データを生成し、前記下位側の要素データの上位側の桁を0または符号ビットにして前記部分積生成回路に入力される被乗数の下位側の要素データを生成する分割回路を有し、

前記ブースセクタは、前記デコード結果が負の部分積を選択する場合、セレクトされるデータをビット反転して前記負の部分積を生成し、

前記並列モード及び単一のデータを乗算する通常モードのいずれの場合も、前記負の部分積の最下位ビットにホットビットが加算される付記1に記載の乗算回路。

【0115】

(付記3)

前記部分積加算回路は、前記キャリア保存加算器に前記ホットビットを入力して加算させる付記2に記載の乗算回路。

【0116】

(付記4)

10

20

30

40

50

さらに、

前記並列モードで、前記第3のキャリー保存加算器に入力される前記第1の加算データ及び第1のキャリーデータの上位側の桁を0に変更し、前記第2の加算データ及び第2のキャリーデータの下位側の桁を0に変更するゼロマスク回路を有する付記1または2に記載の乗算回路。

【0117】

(付記5)

前記第3のキャリー保存加算器は、第1の前記補正ホットビット加算回路を有し、前記第1の補正ホットビット加算回路は、前記並列モードで、前記第2の加算データまたは前記第2のキャリーデータに前記補正ホットビットを加算する付記1または4に記載の乗算回路。

10

【0118】

(付記6)

前記部分積加算回路は、前記第3のキャリー保存加算器が出力する第3の加算データ及び第3のキャリーデータを加算する全加算器を有し、

前記全加算器は、第2の前記補正ホットビット加算回路を有し、

前記第2の補正ホットビット加算回路は、前記並列モードで、前記第3の加算データ及び前記第3のキャリーデータに前記補正ホットビットを加算する付記5に記載の乗算回路。

【0119】

20

(付記7)

前記キャリー保存加算器は、4つの入力データを演算して加算データとキャリーデータを有する2つの出力データを出力し、

前記部分積加算回路は、複数段のキャリー保存加算器を有し、

前記補正ホットビット生成部は、前記並列モードで、前記上位側の要素データについて、前記乗数の複数の組合せのデコード結果が全て正の部分積(Z)の場合は前記補正ホットビットを0に、全て負の部分積(F)の場合は前記補正ホットビットを2に、一部に負の部分積が含まれる場合は前記補正ホットビットを1にする付記1に記載の乗算回路。

【0120】

(付記8)

30

前記補正ホットビット生成部は、前記第1、第2のキャリー保存加算器と並列に前記補正ホットビットを生成し、前記補正ホットビット加算回路に前記補正ホットビットを出力する付記1に記載の乗算回路。

【0121】

(付記9)

乗数の組合せをデコードするブースデコーダと、デコード結果に応じて被乗数と前記乗数の部分積を生成するブースセクタとを有する部分積生成回路と、

複数の前記部分積を並列に加算するキャリー保存加算器をツリー状に配置し、所定段の前記キャリー保存加算器が出力する加算データとキャリーデータを後段の前記キャリー保存加算器が加算する部分積加算回路とを有し、

40

前記部分積加算回路は、前記並列モードで、下位側の並列データを入力し第1の加算データ及び第1のキャリーデータを生成する第1のキャリー保存加算器と、上位側の並列データを入力し第2の加算データ及び第2のキャリーデータを生成する第2のキャリー保存加算器と、前記第1の加算データ及び第1のキャリーデータと前記第2の加算データ及び第2のキャリーデータとを加算する第3のキャリー保存加算器とを有する乗算回路の乗算方法において、

複数のデータを並列に乗算する並列モードで、上位側の並列データのデコード結果に応じて補正加算すべき補正ホットビットを生成し、

前記部分積加算回路は、前記並列モードで、前記上位側の並列データに前記補正ホットビットを加算する乗算方法。

50

【符号の説明】

【 0 1 2 2 】

1 1 : ブースデコーダ

1 2 : ブースセクタ

1 0 : 部分積生成回路

2 0 , 2 1 , 2 2 : ワレスツリー加算器 , 部分積加算回路

C S A : キャリー保存加算器

2 2 : 全加算器

H , H B : ホットビット , 補正ホットビット

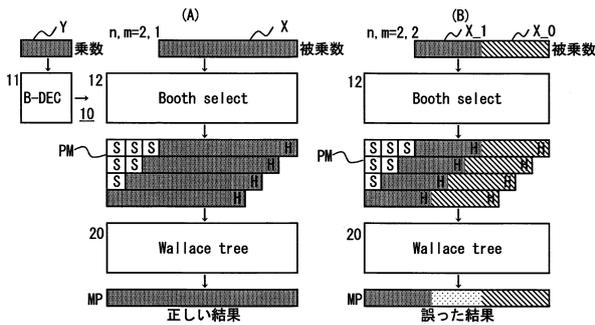
3 0 : 分割回路

4 0 : ゼロマスク回路

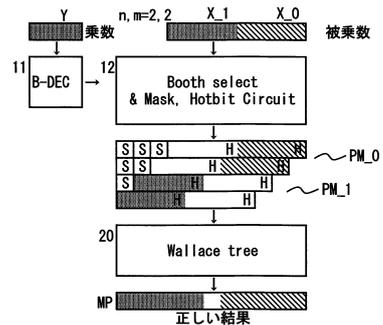
6 0 : ホットビット加算回路

6 2 : 補正ホットビット加算回路

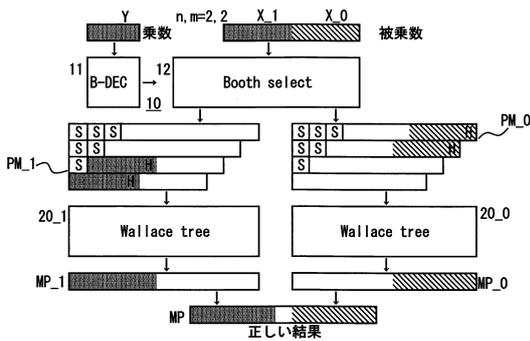
【 図 1 】



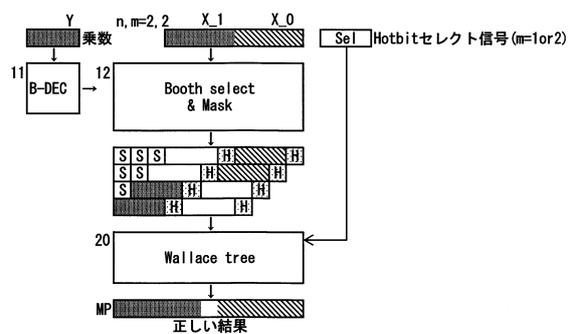
【 図 3 】



【 図 2 】



【 図 4 】



【図10】

ある3-bitに着目

パターンZ	A4	0	0	0
	A3	0	0	0
	A2	0	0	0
	A1	0	0	0
	C0	0	0	0
	S	0		
	C	0		

パターンX-1	A4	1	1	1	0	0	0	0	0	0	0
	A3	0	0	0	1	1	1	0	0	0	0
	A2	0	0	0	0	0	0	1	1	1	0
	A1	0	0	0	0	0	0	0	0	1	1
	C0	0	0	0	0	0	0	0	0	0	0
	S	1			1			1			
	C	0			0			0			

パターンX-2	A4	1	1	1	1	1	1	0	0	0	0	0	0
	A3	1	1	1	0	0	0	0	1	1	1	1	0
	A2	0	0	0	1	1	0	0	1	1	0	0	1
	A1	0	0	0	0	0	1	1	0	0	1	1	1
	C0	1	1	1	1	1	0	0	1	1	0	0	0
	S	1			1			0			0		
	C	0			0			1			1		

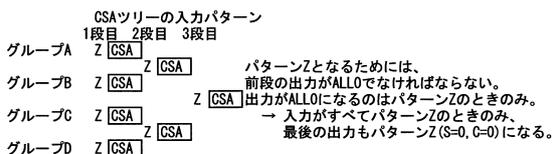
パターンX-3	A4	1	1	1	1	1	1	0	0	0	0
	A3	1	1	1	1	0	0	1	1	1	1
	A2	1	1	1	0	0	1	1	1	1	1
	A1	0	0	1	1	1	1	1	1	1	1
	C0	1	1	1	1	1	1	1	1	1	1
	S	0			0			0			
	C	1			1			1			

パターンF	A4	1	1	1	1
	A3	1	1	1	1
	A2	1	1	1	1
	A1	1	1	1	1
	C0	1	1	1	1
	S	1			
	C	1			

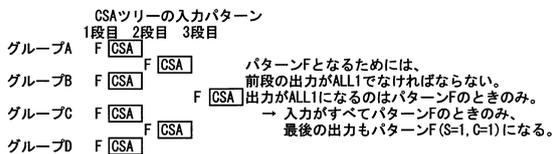
	発生したCRY	発生すべきCRY	差分(不足数)
パターンZ	0	0	0
パターンX-1	0	1	1 * 入力1の数が1
パターンX-2	1	2	1 * 入力1の数が2
パターンX-3	2	3	1 * 入力1の数が3
パターンF	2	4	2

【図12】

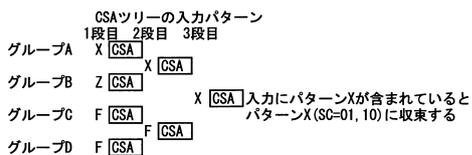
(1) 入力すべてパターンZの場合



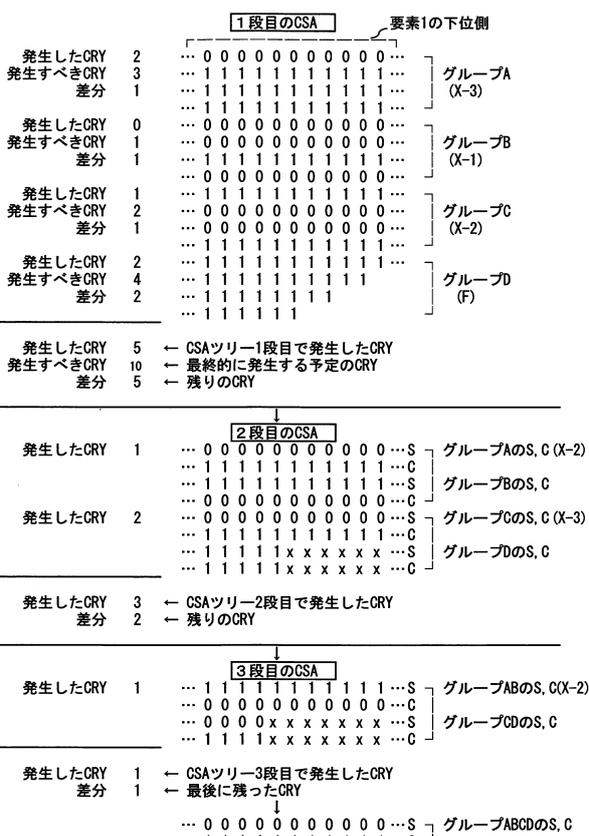
(2) 入力すべてパターンFの場合



(3) 上記以外の場合

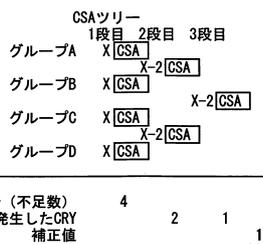


【図11】

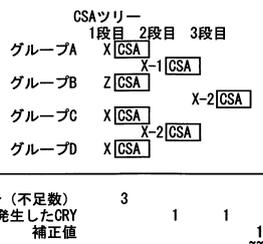


【図13】

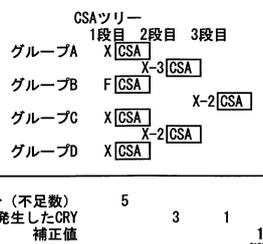
(1) 入力すべてパターンXの場合



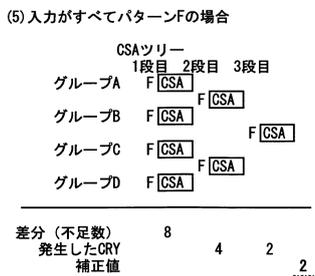
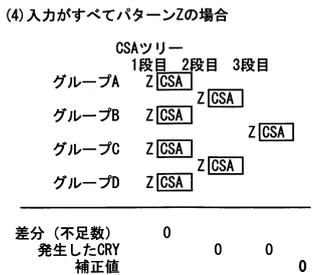
(2) 入力にパターンZを含む場合



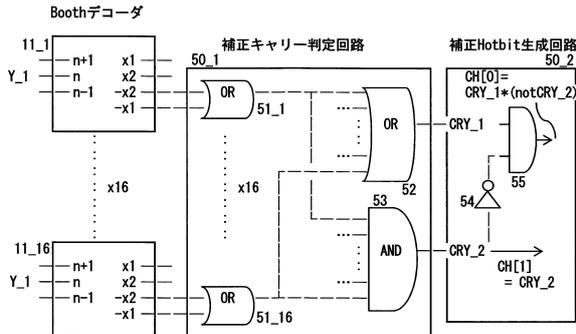
(3) 入力にパターンFを含む場合



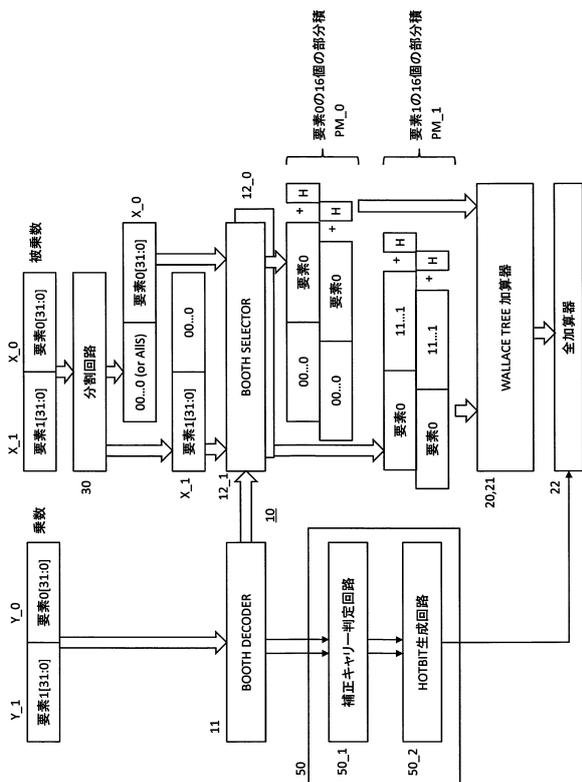
【図14】



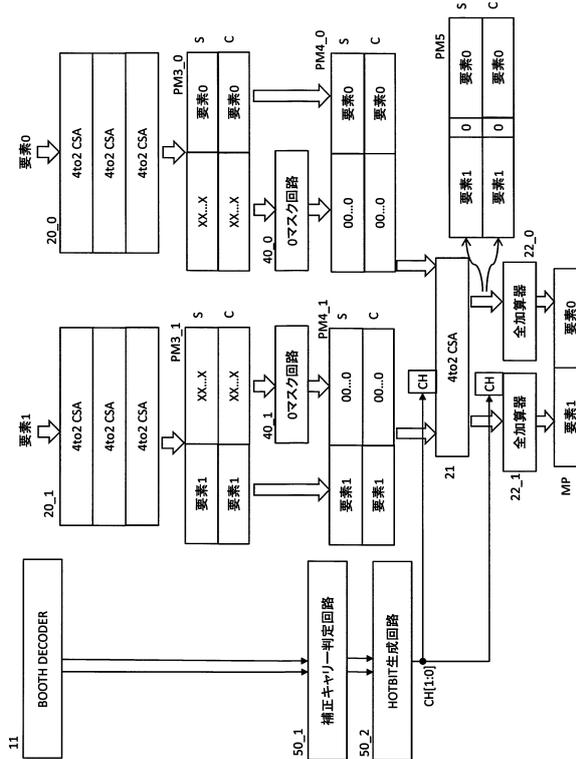
【図15】



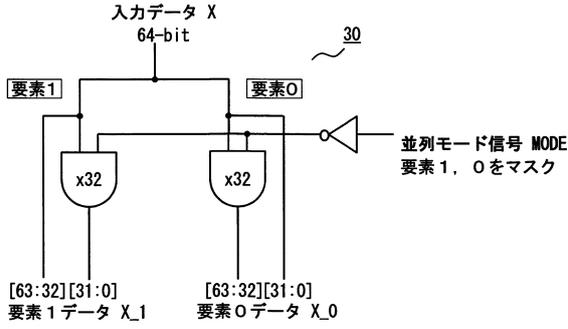
【図16】



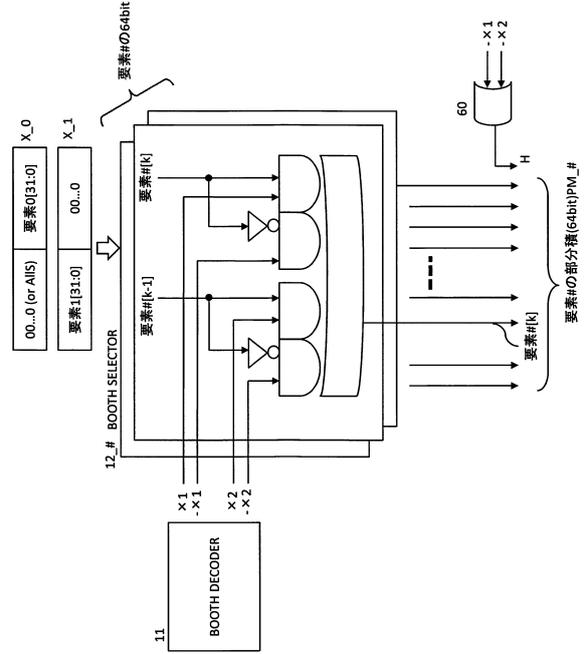
【図17】



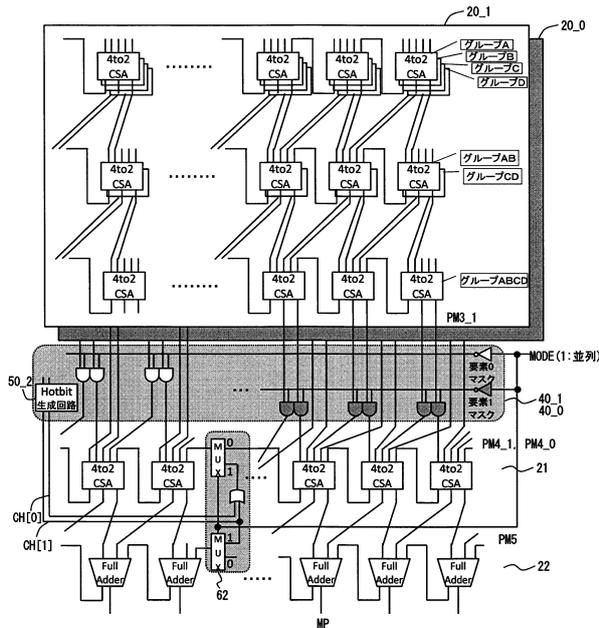
【図18】



【図19】



【図20】

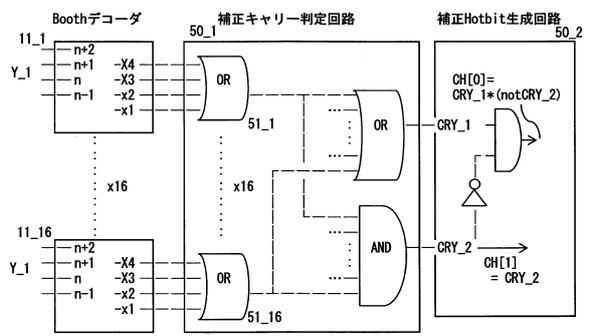


【図21】

n+2	n+1	n	n-1	出力
0	0	0	0	0
0	0	0	1	x1
0	0	1	0	x1
0	0	1	1	x2
0	1	0	0	x2
0	1	0	1	x3
0	1	1	0	x3
0	1	1	1	x4
1	0	0	0	-x4
1	0	0	1	-x3
1	0	1	0	-x3
1	0	1	1	-x2
1	1	0	0	-x2
1	1	0	1	-x1
1	1	1	0	-x1
1	1	1	1	0

加算数が負の場合に 4to2CSAの入力が1になる

【図22】



フロントページの続き

(56)参考文献 米国特許第8037119 (US, B1)
特開2000-215028 (JP, A)
特開平10-149277 (JP, A)
特開平8-44540 (JP, A)
特開平6-4271 (JP, A)

(58)調査した分野(Int.Cl., DB名)
G06F 7/533