(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第4042215号
(P4042215)

(45)発行日　平成20年2月6日(2008.2.6)　(24)登録日　平成19年11月22日(2007.11.22)

(51)Int.Cl.			F I		
G06F	7/535	(2006.01)	G06F	7/535	A
G06F	7/62	(2006.01)	G06F	7/62	C

請求項の数 9　(全 19 頁)

(21)出願番号　特願平10-167418
(22)出願日　平成10年6月15日(1998.6.15)
(65)公開番号　特開2000-3267(P2000-3267A)
(43)公開日　平成12年1月7日(2000.1.7)
審査請求日　平成17年3月8日(2005.3.8)

(73)特許権者　000002185
ソニー株式会社
東京都港区港南１丁目７番１号
(74)代理人　100094053
弁理士　佐藤　隆久
(72)発明者　小沼　弘一
東京都品川区北品川６丁目７番３５号　ソニー株式会社内

審査官　石井　茂和

(56)参考文献　特開平５８－０８６６３７（ＪＰ，Ａ）
特開平５７－１５９３４６（ＪＰ，Ａ）
特開昭６３－０７１７２９（ＪＰ，Ａ）

最終頁に続く

(54)【発明の名称】演算処理装置およびその方法

(57)【特許請求の範囲】

【請求項１】

自然数ｉおよびｊを０≦ｉ，ｊ≦ｎ－１とし、整数ｎを２≦ｎとした場合に、ｎビットのデータＡを構成するビットデータA_0，A_1，．．，A_i，．．，A_{n-2}，A_{n-1}と、ｎビットのデータＢを構成するビットデータB_0，B_1，．．，B_i，．．，B_{n-2}，B_{n-1}と、ｎビットのデータＣを構成するビットデータC_0，C_1，．．，C_j，．．，C_{n-2}，C_{n-1}とを用いて、演算「（Ａ－Ｂ）×Ｃ」を行う演算処理装置において、

ビットデータA_i，B_iおよびC_jを入力し、ビットデータC_jが論理値「１」のときにビットデータA_iを出力し、ビットデータC_jが論理値「０」のときにビットデータB_iを出力し、前記自然数ｉとｊとの全ての組み合わせのそれぞれに対応して設けられた複数のビットデータ選択手段と、

前記ビットデータ選択手段から出力されたビットデータを（ｉ＋ｊ）ビット目に下位のビットからの桁上げであるキャリーデータを含めて加算するように、各ビット毎に、前記ビットデータ選択手段から出力されたビットデータと、前記データＢの２の補数データを最上位ビットに向けてｎビットだけシフトしたデータと、前記データＢとを加算する加算手段と

を有する演算処理装置。

【請求項２】

前記ビットデータB_0，B_1，．．，B_i，．．，B_{n-2}，B_{n-1}を反転したビットデータ$\bar{B}_0$，$\bar{B}_1$，．．，$\bar{B}_i$，．．，$\bar{B}_{n-2}$，$\bar{B}_{n-1}$を求める反転値生成手段

をさらに有し、
前記加算手段は、
ｎ＋１，．．，（ｎ＋ｉ），．．，（２ｎ－１）ビット目に、前記反転値生成手段で求めた前記ビットデータB_1^-，．．，B_i^-，．．，B_{n-1}^-をそれぞれ加算し、
ｎビット目に、前記ビットデータB_0^-と、論理値「１」とを加算する
請求項１に記載の演算処理装置。
【請求項３】
前記加算手段は、
０，１，．．，ｉ，．．，（ｎ－２），（ｎ－１）ビット目に、前記データＢのビットデータB_0，B_1，．．，B_i，．．，B_{n-2}，B_{n-1}をそれぞれ加算する
請求項１に記載の演算処理装置。
【請求項４】
前記加算手段は、前記加算の加算結果として、ビットデータS_0，S_1，．．，S_{2n-1}，S_{2n}からなる（２ｎ＋１）ビットのデータＳを出力し、
前記ビットデータS_{2n}は、符号値を示す
請求項１に記載の演算処理装置。
【請求項５】
前記ビットデータ選択手段は、入力したビットデータC_iが論理値「１」のときに導通状態となる第１のトランスミッションゲートと、前記入力したビットデータC_iが論理値「０」のときに導通状態となる第２のトランスミッションゲートとを有する
請求項１に記載の演算処理装置。
【請求項６】
自然数ｉおよびｊを０≦ｉ，ｊ≦ｎ－１とし、整数ｎを２≦ｎとした場合に、ｎビットのデータＡを構成するビットデータA_0，A_1，．．，A_i，．．，A_{n-2}，A_{n-1}と、ｎビットのデータＢを構成するビットデータB_0，B_1，．．，B_i，．．，B_{n-2}，B_{n-1}と、ｎビットのデータＣを構成するビットデータC_0，C_1，．．，C_j，．．，C_{n-2}，C_{n-1}とを用いて、演算「（Ａ－Ｂ）×Ｃ」を行う演算処理方法において、
前記自然数ｉとｊとの全ての組み合わせのそれぞれについて、ビットデータA_i，B_iおよびC_jを入力し、ビットデータC_jが論理値「１」のときにビットデータA_iを選択し、ビットデータC_jが論理値「０」のときにビットデータB_iを選択する処理を行い、
前記選択されたビットデータを（ｉ＋ｊ）ビット目に下位のビットからの桁上げであるキャリーデータを含めて加算するように、各ビット毎に、前記選択されたビットデータと、前記データＢの２の補数データを最上位ビットに向けてｎビットだけシフトしたデータと、前記データＢとを加算する
演算処理方法。
【請求項７】
前記ビットデータB_0，B_1，．．，B_i，．．，B_{n-2}，B_{n-1}を反転したビットデータB_0^-，B_1^-，．．，B_i^-，．．，B_{n-2}^-，B_{n-1}^-を求め、
ｎ＋１，．．，（ｎ＋ｉ），．．，（２ｎ－１）ビット目に、前記ビットデータB_1^-，．．，B_i^-，．．，B_{n-1}^-をそれぞれ加算し、
ｎビット目に、前記ビットデータB_0^-と、論理値「１」とを加算する
請求項６に記載の演算処理方法。
【請求項８】
０，１，．．，ｉ，．．，（ｎ－２），（ｎ－１）ビット目に、前記データＢのビットデータB_0，B_1，．．，B_i，．．，B_{n-2}，B_{n-1}をそれぞれ加算する
請求項６に記載の演算処理方法。
【請求項９】
前記加算の加算結果として、ビットデータS_0，S_1，．．，S_{2n-1}，S_{2n}からなる（２ｎ＋１）ビットのデータＳを出力し、
前記ビットデータS_{2n}は、符号値を示す

請求項６に記載の演算処理方法。
【発明の詳細な説明】
【０００１】
【発明が属する技術分野】
本発明は、演算処理装置およびその方法に関する。
【０００２】
【従来の技術】
正の２進数データＡ，ＢおよびＣを入力して、演算「（Ａ－Ｂ）×Ｃ」を行う演算処理装置がある。
以下、演算「（Ａ－Ｂ）×Ｃ」を行う従来の演算処理装置について説明する。
図６は、演算「（Ａ－Ｂ）×Ｃ」を行う従来の演算処理装置１の構成図である。
図６に示すように、演算処理装置１は、減算器２および乗算器３を有し、それぞれ４ビットのデータＡ，ＢおよびＣを用いて、演算「（Ａ－Ｂ）×Ｃ」を行う。
演算処理装置１では、例えば、４ビットのデータＡと４ビットのデータＢとの減算を減算器２で行い、その符号付きの５ビットの減算結果Ｙと、符号拡張によりＭＳＢ(Most Significant Bit)が論理値「０」の５ビットのデータＣとの乗算を乗算器３で行う。そして、乗算器３の乗算結果が、演算「（Ａ－Ｂ）×Ｃ」の演算結果となる。
【０００３】
減算器２としては、例えば、図７に示すように、全加算器（ＦＡ: Full Adder）10_0，10_1，10_2，10_3を直列に接続したリップルキャリー(Ripple Carry)型の加算器が用いられる。
この減算器２では、ＬＳＢ(Least Significant Bit) に対応する演算を行う全加算器10_1のＣｉ(Carry in)端子に２の補数を求めるための「１」が入力される。また、全加算器10_0～10_3には、それぞれデータＡのビットデータA_0～A_3が入力されると共に、インバータ11_0～11_3を介して、データＢのビットデータB_0～B_3が入力される。そして、全加算器10_0～10_3のｓ端子から、４ビットの減算結果ＹのビットデータY_0～Y_3がそれぞれ出力されると共に、全加算器10_3のＣＯ(Carry Out) 端子から減算結果Ｙの符号を示すビットデータY_4が出力される。
【０００４】
なお、全加算器10_1～10_3としては、図８に示すように、ＡＮＤ回路15_1，15_2、ＯＲ回路16_1，16_2およびＸＯＲ(exclusive OR)回路17_1，17_2を組み合わせて構成される一般的な全加算器が用いられる。全加算器10_1～10_3では、in_1端子、in_2端子およびＣｉ(Carry in)端子から入力されたビットデータが加算され、当該加算結果のキャリーがＣＯ(Carry Out) 端子から出力され、和データがＳ端子から出力される。
【０００５】
次に、図６に示す乗算器３の構成について説明する。
図９は、乗算器３が採用するバウウーリ(Baugh Wooly) 方式による２の補数乗算を説明するための図である。
図１０は、図９に示す２の補数乗算を行う乗算器３の構成図である。
図１０に示すように、乗算器３は、部分積加算回路２０および最終段加算回路３０を有する。
部分積加算回路２０は、Wallace-tree方式を採用しており、ＡＮＤ回路21_0～21_{24}、全加算器22_1～22_{13}、半加算器23_1～23_3、インバータ回路24_1～24_{11}を有する。
また、最終段加算回路３０は、リップルキャリー方式を採用しており、全加算器22_{14}～22_{19}および半加算器23_4，23_5を有する。
ここで、全加算器22_{14}～22_{19}は、前述した図８に示す構成をしている。また、半加算器23_1～23_3としては、図１１に示すように、ＡＮＤ回路15_3およびＸＯＲ回路17_3を有し、in_1端子およびin_2端子から入力したデータを加算し、当該加算結果の

キャリーをＣＯ(Carry Out) 端子から出力し、和データをＳ端子から出力する。
【０００６】
乗算器３では、部分積加算回路２０のＡＮＤ回路２１_{0}～２１_{24}において、減算器２からの減算結果Ｙのビットデータ$Ｙ_{0}$，$Ｙ_{1}$，$Ｙ_{2}$，$Ｙ_{3}$，$Ｙ_{4}$と、符号拡張によりＭＳＢが論理値「０」のデータＣのビットデータ$Ｃ_{0}$，$Ｃ_{1}$，$Ｃ_{2}$，$Ｃ_{3}$，０とを用いて、図９に示す部分積がそれぞれ演算される。そして、当該演算された部分積が、各桁毎に、下の桁からの桁上げを含めて、部分積加算回路２０および最終段加算回路３０の全加算器２２_{1}～２２_{19}および半加算器２３_{1}～２３_{5}で加算される。これにより、ＡＤＮ回路２１_{0}の出力端子、半加算器２３_{1}，２３_{4}、全加算器２２_{14}，２２_{15}，２２_{16}，２２_{17}，２２_{18}，２２_{19}および半加算器２３_{5}のｓ端子から出力される和データが、１０ビットの演算結果Ｓのそれぞれビットデータ$Ｓ_{0}$，$Ｓ_{1}$，$Ｓ_{2}$，$Ｓ_{3}$，$Ｓ_{4}$，$Ｓ_{5}$，$Ｓ_{6}$，$Ｓ_{7}$，$Ｓ_{8}$，$Ｓ_{9}$になる。
【０００７】
【発明が解決しようとする課題】
ところで、上述した従来の演算処理装置１では、図７および図１０に示すように、全加算器１０_{0}～１０_{3}、ＡＮＤ回路２１_{0}～２１_{24}、全加算器２２_{1}～２２_{19}、半加算器２３_{1}～２３_{5}およびインバータ回路２４_{1}～２４_{11}を有し、回路が大規模化するという問題がある。
すなわち、演算処理装置１では、図１に示すように、減算器２で減算を行うため、４ビットのデータＡ，Ｂを用いた場合には、その減算結果は符号ビットを含めて５ビットになる。その結果、乗算器３では、５ビットの乗算を行う必要があり、回路が大規模化する。
また、上述した従来の演算処理装置１では、演算のクリティカルパスは、全加算器１０_{0}～１０_{3}、半加算器２３_{4}、全加算器２２_{14}，２２_{15}，２２_{16}，２２_{17}，２２_{18}，２２_{19}および半加算器２３_{5}となり、演算時間が長いという問題がある。
【０００８】
本発明は上述した従来技術の問題点に鑑みてなされ、演算「（Ａ－Ｂ）×Ｃ」を行う回路の規模を縮小できる演算処理装置を提供することを目的とする。
また、本発明は、演算「（Ａ－Ｂ）×Ｃ」の演算時間を短縮化できる演算処理装置を提供することを目的とする。
【０００９】
【課題を解決するための手段】
上述した従来技術の問題点を解決し、上述した目的を達成するために、本発明の演算処理装置は、自然数ｉおよびｊを$０ \leqq ｉ，ｊ \leqq ｎ－１$とし、整数ｎを$２ \leqq ｎ$とした場合に、ｎビットのデータＡを構成するビットデータ$Ａ_{0}$，$Ａ_{1}$，．．，$Ａ_{i}$，．．，$Ａ_{n-2}$，$Ａ_{n-1}$と、ｎビットのデータＢを構成するビットデータ$Ｂ_{0}$，$Ｂ_{1}$，．．，$Ｂ_{i}$，．．，$Ｂ_{n-2}$，$Ｂ_{n-1}$と、ｎビットのデータＣを構成するビットデータ$Ｃ_{0}$，$Ｃ_{1}$，．．，$Ｃ_{j}$，．．，$Ｃ_{n-2}$，$Ｃ_{n-1}$とを用いて、演算「（Ａ－Ｂ）×Ｃ」を行う演算処理装置であって、ビットデータ$Ａ_{i}$，$Ｂ_{i}$および$Ｃ_{j}$を入力し、$Ｃ_{j}$が論理値「１」のときにビットデータ$Ａ_{i}$を出力し、$Ｃ_{j}$が論理値「０」のときにビットデータ$Ｂ_{i}$を出力し、前記自然数ｉとｊとの全ての組み合わせのそれぞれに対応して設けられた複数のビットデータ選択手段と、前記ビットデータ選択手段から出力されたビットデータを（ｉ＋ｊ）ビット目に下位のビットからの桁上げであるキャリーデータを含めて加算するように、各ビット毎に、前記ビットデータ選択手段から出力されたビットデータと、前記データＢの２の補数データを最上位ビットに向けてｎビットだけシフトしたデータと、前記データＢとを加算する加算手段とを有する。
【００１０】
本発明の演算処理装置では、下記式（１）に基づいて、演算「（Ａ－Ｂ）×Ｃ」を行う。
【００１１】
【数１】

$$S=\left(\sum_{j=0}^{j=n-1}\sum_{i=0}^{i=n-1}2^{i+j}\cdot(A_i\cdot C_j \mid B_i\cdot \overline{C_j})\right)-2^n\times B+B \quad \cdots (1)$$

【００１２】
すなわち、本発明の演算処理装置では、複数のビットデータ選択手段の各々において、入力されたビットデータA_i，B_iのうち、入力したC_jが論理値「１」のときにビットデータA_iが出力され、C_jが論理値「０」のときにビットデータB_iが出力される。
次に、加算手段において、前記ビットデータ選択手段から出力されたビットデータが（ｉ＋ｊ）ビット目に下位のビットからの桁上げであるキャリーデータを含めて加算されるように、各ビット毎に、前記ビットデータ選択手段から出力されたビットデータと、前記データＢの２の補数データを最上位ビットに向けてｎビットだけシフトしたデータと、前記データＢとが加算される。
【００１３】
また、本発明の演算処理方法は、同様に、前記式（１）に基づいて、演算「（Ａ－Ｂ）×Ｃ」を行う。
すなわち、本発明の演算処理方法は、自然数ｉおよびｊを０≦ｉ，ｊ≦ｎ－１とし、整数ｎを２≦ｎとした場合に、ｎビットのデータＡを構成するビットデータA_0，A_1，．．，A_i，．．，A_{n-2}，A_{n-1}と、ｎビットのデータＢを構成するビットデータB_0，B_1，．．，B_i，．．，B_{n-2}，B_{n-1}と、ｎビットのデータＣを構成するビットデータC_0，C_1，．．，C_j，．．，C_{n-2}，C_{n-1}とを用いて、演算「（Ａ－Ｂ）×Ｃ」を行う演算処理方法において、
前記自然数ｉとｊとの全ての組み合わせのそれぞれについて、ビットデータA_i，B_iおよびC_jを入力し、C_jが論理値「１」のときにビットデータA_iを選択し、C_jが論理値「０」のときにビットデータB_iを選択する処理を行い、前記選択されたビットデータを（ｉ＋ｊ）ビット目に下位のビットからの桁上げであるキャリーデータを含めて加算するように、各ビット毎に、前記選択されたビットデータと、前記データＢの２の補数データを最上位ビットに向けてｎビットだけシフトしたデータと、前記データＢとを加算する。
【００１４】
【発明の実施の形態】
以下、本発明の実施形態に係わる演算処理装置およびその方法について説明する。
図１は、４ビットのデータＡ，Ｂ，Ｃを用いて、演算「Ｓ＝（Ａ－Ｂ）×Ｃ」を行う演算処理装置４０の構成図である。
図１に示すように、演算処理装置４０は、部分積加算回路４１および最終段加算回路４２を有する。
部分積加算回路４１は、本発明のビットデータ選択手段としてのマルチプレクサ50_0～50_{15}、全加算器（ＦＡ）52_1～52_{10}、半加算器（ＨＡ）53_1～53_3、インバータ回路54_1～54_4、入力部５５，５６および出力部５７を有する。
最終段加算回路４２は、全加算器52_{11}～52_{16}および半加算器53_4を有する。
【００１５】
入力部５５は、それぞれビットデータA_0，A_1，A_2，A_3，B_0，B_1，B_2，B_3を入力するA_0，A_1，A_2，A_3，B_0，B_1，B_2，B_3端子を有する。
ここで、ビットデータA_0，A_1，A_2，A_3によって４ビットのデータＡが構成され、ビットデータB_0，B_1，B_2，B_3によって４ビットのデータＢが構成される。
【００１６】
入力部５６は、それぞれビットデータC_0，C_1，C_2，C_3を入力するC_0，C_1，C_2，C_3端子を有する。
ここで、ビットデータC_0，C_1，C_2，C_3によって４ビットのデータＣが構成される

。
【００１７】
出力部５７は、それぞれビットデータS_0，S_1，S_2，S_3，S_4，S_5，S_6，S_7，S_8を出力するS_0，S_1，S_2，S_3，S_4，S_5，S_6，S_7，S_8端子を有する。
ここで、ビットデータS_0，S_1，S_2，S_3，S_4，S_5，S_6，S_7，S_8によって９ビットのデータＳが構成される。
【００１８】
マルチプレクサ50_0～50_{15}は、それぞれ下記式（２）で定義される部分積$P_{0,0}$，$P_{1,0}$，$P_{0,1}$，$P_{2,0}$，$P_{1,1}$，$P_{0,2}$，$P_{3,0}$，$P_{2,1}$，$P_{1,2}$，$P_{0,3}$，$P_{3,1}$，$P_{2,2}$，$P_{1,3}$，$P_{3,2}$，$P_{2,3}$，$P_{3,3}$を求める。
【００１９】
【数２】

$$P_{ij} = A_i \cdot C_j \mid B_i \cdot \overline{C_j} \quad (i = 0 \sim 3,\ j = 0 \sim 3) \qquad \cdots (2)$$

【００２０】
図２は、マルチプレクサ50_0～50_{15}の構成図である。
図２に示すように、マルチプレクサ50_0～50_{15}は、インバータ70_1～70_4、トランスミッションゲート70_1，70_2、ｓ端子、ａ端子、ｂ端子およびｏ端子を有する。マルチプレクサ50_0～50_{15}では、ｓ端子が論理値「１」のときに、トランスミッションゲート70_1をオフにし、トランスミッションゲート70_2をオンにして、ａ端子のレベルをｏ端子のレベルとして出力する。一方、マルチプレクサ50_0～50_{15}では、ｓ端子が論理値「０」のときに、トランスミッションゲート70_1をオンにし、トランスミッションゲート70_2をオフにして、ｂ端子のレベルをｏ端子のレベルとして出力する。
【００２１】
すなわち、マルチプレクサ50_0～50_{15}では、ビットデータC_jが論理値「１」のときにａ端子に入力されたビットデータA_iをｏ端子から出力し、ビットデータC_jが論理値「０」のときにａ端子に入力されたビットデータB_iをｏ端子から出力する。
【００２２】
マルチプレクサ50_0～50_{15}で算出された、部分積$P_{0,0}$，$P_{1,0}$，$P_{0,1}$，$P_{2,0}$，$P_{1,1}$，$P_{0,2}$，$P_{3,0}$，$P_{2,1}$，$P_{1,2}$，$P_{0,3}$，$P_{3,1}$，$P_{2,2}$，$P_{1,3}$，$P_{3,2}$，$P_{2,3}$，$P_{3,3}$は、それぞれ（ｉ＋ｊ）ビット目に加算されるように、全加算器52_1～52_{16}および半加算器53_1～53_4において、各ビット毎に、下位ビットからのキャリーを含めて加算される。これにより、下記式（３）に示す「Ａ×Ｃ＋Ｂ×Ｃ¯」が演算される。
【００２３】
【数３】

$$A \times C + B \times \overline{C} = \sum_{j=0}^{j=3} \sum_{i=0}^{i=3} 2^{i+j} \cdot P_{ij} \qquad \cdots (3)$$

【００２４】
ここで、全加算器52_1～52_{16}としては、前述した図８に示す構成のものが用いられる。また、半加算器53_1～53_4としては、前述した図１１に示すものが用いられる。
【００２５】
インバータ回路54_1，51_2，51_3～54_4は、それぞれビットデータB_0，B_1，

B_2，B_3を入力し、それを反転したビットデータB_0^-，B_1^-，B_2^-，B_3^-を出力する。
インバータ回路54_1～54_4における演算は、「$-2^4 \times B$」の演算を行うための反転値演算に対応する。また、全加算器52_6のＣｉ(Carry in)端子に「１」を入力して行われる加算によって、「$-2^4 \times B$」の演算を行うための「１の加算」が行われる。
【００２６】
また、ビットデータB_0～B_3が、それぞれ半加算器53_1、全加算器52_1、半加算器53_2および全加算器52_4のｉｎ$_2$端子に出力される。半加算器53_1、全加算器52_1、半加算器53_2および全加算器52_4において、「＋Ｂ」の演算が行われる。
【００２７】
上述したように、全加算器52_1～52_{16}および半加算器53_1～53_4によって、「Ａ×Ｃ＋Ｂ×Ｃ$^-$」の演算結果と、「$-2^4 \times B$」と、「Ｂ」とを加算することで、下記式（４）の演算が行われる。
【００２８】
【数４】

$$S=\left(\sum_{j=0}^{j=3}\sum_{i=0}^{i=3}2^{i+j}\cdot P_{ij}\right)-2^4\times B+B \qquad \cdots (4)$$

【００２９】
なお、上記式（４）の演算は、図３に示すように表される。
なお、図３において、「（　　）」は、図１に示す演算処理装置４０の構成要素の符号を示しており、隣接して示される演算が当該構成要素で求められること、あるいは、隣接して示されるデータが当該構成要素に入力されることを示している。
ここで、上記式（４）は、演算式「Ｓ＝（Ａ－Ｂ）×Ｃ」と等価である。
【００３０】
以下、上記式（４）が演算式「Ｓ＝（Ａ－Ｂ）×Ｃ」と等価であることを証明する。
演算式「Ｓ＝（Ａ－Ｂ）×Ｃ」は、下記式（５）のように変形できる。
【００３１】
【数５】

$$\begin{aligned}S&=(A-B)\times C\\&=A\times C-B\times C\\&=A\times C+B\times(-C)\end{aligned} \qquad \cdots (5)$$

【００３２】
ところで、２進数ｘの１の補数ｘ$^-$は、下記式（６）で示される。
【数６】

$$\overline{X}=1-X$$

【００３３】
従って、上記式（６）を前記式（５）の「－Ｃ」に適用すると、下記式（７）が成り立つ。
【数７】

$$-C=\overline{C}-\sum_{i=0}^{i=3}2^i \qquad \cdots (7)$$

【００３４】

また、上記式（７）を用いて、上記式（５）を書換え、さらに変形を行うと下記式（８）のようになる。

【００３５】

【数８】

$$
\begin{aligned}
S &= A \times C + B \times (\overline{C} - \sum_{i=0}^{i=3} 2^i) \\
&= A \times C + B \times \overline{C} - B \times \sum_{i=0}^{i=3} 2^i \\
&= A \times C + B \times \overline{C} - B \times (2^4 - 1) \\
&= A \times C + B \times \overline{C} - 2^4 \times B + B \qquad \cdots (8)
\end{aligned}
$$

【００３６】

ここで、上記式（８）および式（３）から、演算式「Ｓ＝（Ａ－Ｂ）×Ｃ」は、前記式（４）と等価であることが分かる。

【００３７】

以下、図１に示す演算処理装置４０の各構成要素の接続形態について説明する。

マルチプレクサ50_0，50_2，50_5，50_9は、ａ端子が入力部５５のA_0端子に接続され、ｂ端子が入力部５５のB_0端子に接続されている。

マルチプレクサ50_1，50_4，50_8，50_{12}は、ａ端子が入力部５５のA_1端子に接続され、ｂ端子が入力部５５のB_1端子に接続されている。

マルチプレクサ50_3，50_7，50_{11}，50_{14}は、ａ端子が入力部５５のA_2端子に接続され、ｂ端子が入力部５５のB_2端子に接続されている。

マルチプレクサ50_6，50_{10}，50_{13}，50_{15}は、ａ端子が入力部５５のA_3端子に接続され、ｂ端子が入力部５５のB_3端子に接続されている。

【００３８】

マルチプレクサ50_0，50_1，50_3，50_6のｓ端子は、入力部５６のC_0端子に接続されている。

マルチプレクサ50_2，50_4，50_7，50_{10}のｓ端子は、入力部５６のC_1端子に接続されている。

マルチプレクサ50_5，50_8，50_{11}，50_{13}のｓ端子は、入力部５６のC_2端子に接続されている。

マルチプレクサ50_9，50_{12}，50_{14}，50_{15}のｓ端子は、入力部５６のC_3端子に接続されている。

【００３９】

半加算器53_1は、in_1端子がマルチプレクサ50_0のｏ端子に接続され、in_2端子が入力部５５のB_0端子に接続されている。

また、半加算器53_1は、ｓ端子が出力部５７のS_0端子に接続され、ＣＯ端子が半加算器53_4のin_2端子に接続されている。

全加算器52_1は、in_1端子がマルチプレクサ50_2のｏ端子に接続され、in_2端子がマルチプレクサ50_1のｏ端子に接続され、Ｃｉ端子が入力部５５のB_1端子に接続されている。

また、全加算器52_1は、ｓ端子が半加算器53_4のin_1端子に接続され、ＣＯ端子が全加算器52_{11}のin_2端子に接続されている。

【００４０】

半加算器53_2は、in_1端子がマルチプレクサ50_3のｏ端子に接続され、in_2端子が入力部５５のB_2端子に接続されている。

また、半加算器５３$_2$は、ｓ端子が全加算器５２$_2$のＣｉ端子に接続され、ＣＯ端子が全加算器５２$_4$のＣｉ端子に接続されている。
半加算器５３$_3$は、ｉｎ$_1$端子がマルチプレクサ５０$_9$のｏ端子に接続され、ｉｎ$_2$端子がマルチプレクサ５０$_8$のｏ端子に接続されている。
また、半加算器５３$_3$は、ｓ端子が全加算器５２$_3$のｉｎ$_1$端子に接続され、ＣＯ端子が全加算器５２$_7$のｉｎ$_2$端子に接続されている。

【００４１】

全加算器５２$_5$は、ｉｎ$_1$端子がマルチプレクサ５０$_{12}$のｏ端子に接続され、ｉｎ$_2$端子がマルチプレクサ５０$_{11}$のｏ端子に接続され、Ｃｉ端子がマルチプレクサ５０$_{10}$のｏ端子に接続されている。
また、全加算器５２$_5$は、ｓ端子が全加算器５２$_6$のｉｎ$_1$端子に接続され、ＣＯ端子が全加算器５２$_9$のｉｎ$_2$端子に接続されている。
全加算器５２$_8$は、ｉｎ$_1$端子がマルチプレクサ５０$_{14}$のｏ端子に接続され、ｉｎ$_2$端子がマルチプレクサ５０$_{13}$のｏ端子に接続され、Ｃｉ端子がインバータ５４$_2$の出力端子に接続されている。
また、全加算器５２$_8$は、ｓ端子が全加算器５２$_9$のｉｎ$_1$端子に接続され、ＣＯ端子が全加算器５２$_{10}$のＣｉ端子に接続されている。

【００４２】

全加算器５２$_2$は、ｉｎ$_1$端子がマルチプレクサ５０$_5$のｏ端子に接続され、ｉｎ$_2$端子がマルチプレクサ５０$_4$に接続されている。
また、全加算器５２$_2$は、ｓ端子が全加算器５２$_{11}$のｉｎ$_1$端子に接続され、ＣＯ端子が全加算器５２$_{12}$のｉｎ$_2$端子に接続されている。
全加算器５２$_3$は、ｉｎ$_2$端子がマルチプレクサ５０$_7$のｏ端子に接続され、Ｃｉ端子がマルチプレクサ５０$_6$のｏ端子に接続されている。
また、全加算器５２$_3$は、ｓ端子が全加算器５２$_4$のｉｎ$_1$端子に接続され、ＣＯ端子が全加算器５２$_7$のＣｉ端子に接続されている。

【００４３】

全加算器５２$_6$は、ｉｎ$_2$端子がインバータ５４$_1$の出力端子に接続され、Ｃｉ端子に、論理値「１」が入力されている。
また、全加算器５２$_6$は、ｓ端子が全加算器５２$_7$のｉｎ$_1$端子に接続され、ＣＯ端子が全加算器５２$_9$のＣｉ端子に接続されている。
全加算器５２$_{10}$は、ｉｎ$_2$端子がインバータ５４$_3$の出力端子に接続されている。
また、全加算器５２$_{10}$は、ｓ端子が全加算器５２$_{15}$のｉｎ$_1$端子に接続され、ＣＯ端子が全加算器５２$_{16}$のｉｎ$_2$端子に接続されている。

【００４４】

全加算器５２$_4$は、ｉｎ$_2$端子が入力部５５のＢ$_3$端子に接続されている。
また、全加算器５２$_2$は、ｓ端子が全加算器５２$_{12}$のｉｎ$_1$端子に接続され、ＣＯ端子が全加算器５２$_{13}$のｉｎ$_2$端子に接続されている。
全加算器５２$_7$は、ｓ端子が全加算器５２$_{13}$のｉｎ$_1$端子に接続され、ＣＯ端子が全加算器５２$_{14}$のｉｎ$_2$端子に接続されている。
全加算器５２$_9$は、ｓ端子が全加算器５２$_{14}$のｉｎ$_1$端子に接続され、ＣＯ端子が全加算器５２$_{15}$のｉｎ$_2$端子に接続されている。

【００４５】

半加算器５３$_4$は、ｓ端子が出力部５７のＳ$_1$端子に接続され、ＣＯ端子が全加算器５２$_{11}$のＣｉ端子に接続されている。
全加算器５２$_{11}$は、ｓ端子が出力部５７のＳ$_2$端子に接続され、ＣＯ端子が全加算器５２$_{12}$のＣｉ端子に接続されている。
全加算器５２$_{12}$は、ｓ端子が出力部５７のＳ$_3$端子に接続され、ＣＯ端子が全加算器５２$_{13}$のＣｉ端子に接続されている。
全加算器５２$_{13}$は、ｓ端子が出力部５７のＳ$_4$端子に接続され、ＣＯ端子が全加算器５２

$_{14}$のＣｉ端子に接続されている。
全加算器５２$_{14}$は、ｓ端子が出力部５７のＳ$_5$ 端子に接続され、ＣＯ端子が全加算器５２$_{15}$のＣｉ端子に接続されている。
全加算器５２$_{15}$は、ｓ端子が出力部５７のＳ$_6$ 端子に接続され、ＣＯ端子が全加算器５２$_{16}$のＣｉ端子に接続されている。
【００４６】
全加算器５２$_{16}$は、ｉｎ$_1$ 端子がインバータ５４$_4$ の出力端子に接続され、ｓ端子が出力部５７のＳ$_7$ 端子に接続され、ＣＯ端子がインバータ５４$_5$ を介して出力部５７のＳ$_8$ 端子に接続されている。
【００４７】
以下、図１に示す演算処理装置４０の動作について説明する。
図４は、演算処理装置４０における演算方法を説明するためのフローチャートである。
〔ステップＳ１〕
ビットデータＡ$_0$，Ｂ$_0$，Ａ$_1$，Ｂ$_1$，Ａ$_2$，Ｂ$_2$，Ａ$_3$，Ｂ$_3$ が、入力部５５のＡ$_0$，Ｂ$_0$，Ａ$_1$，Ｂ$_1$，Ａ$_2$，Ｂ$_2$，Ａ$_3$，Ｂ$_3$ 端子に入力される。また、ビットデータＣ$_0$，Ｃ$_1$，Ｃ$_2$，Ｃ$_3$ が、入力部５６のＣ$_0$，Ｃ$_1$，Ｃ$_2$，Ｃ$_3$ 端子に入力される。
【００４８】
そして、以下に示すマルチプレクサ５０$_0$ ～５０$_{15}$におけるビットデータＡ$_0$ ～Ａ$_3$，Ｂ$_0$ ～Ｂ$_3$ の選択が同時に行われ、それぞれ選択されたビットデータが対応する半加算器および全加算器に出力される。
【００４９】
具体的には、マルチプレクサ５０$_0$ において、ビットデータＣ$_0$ が論理値「１」のときに、ビットデータＡ$_0$ がｏ端子から半加算器５３$_1$ のｉｎ$_1$ 端子に出力され、ビットデータＣ$_0$ が論理値「０」のときに、ビットデータＢ$_0$ がｏ端子から半加算器５３$_1$ のｉｎ$_1$ 端子に出力される。
マルチプレクサ５０$_1$ において、ビットデータＣ$_0$ が論理値「１」のときに、ビットデータＡ$_1$ がｏ端子から全加算器５２$_1$ のｉｎ$_2$ 端子に出力され、ビットデータＣ$_0$ が論理値「０」のときに、ビットデータＢ$_1$ がｏ端子から全加算器５２$_1$ のｉｎ$_2$ 端子に出力される。
【００５０】
マルチプレクサ５０$_3$ において、ビットデータＣ$_0$ が論理値「１」のときに、ビットデータＡ$_2$ がｏ端子から半加算器５３$_2$ のｉｎ$_1$ 端子に出力され、ビットデータＣ$_0$ が論理値「０」のときに、ビットデータＢ$_2$ がｏ端子から半加算器５３$_2$ のｉｎ$_1$ 端子に出力される。
マルチプレクサ５０$_6$ において、ビットデータＣ$_0$ が論理値「１」のときに、ビットデータＡ$_3$ がｏ端子から全加算器５２$_3$ のＣｉ端子に出力され、ビットデータＣ$_0$ が論理値「０」のときに、ビットデータＢ$_3$ がｏ端子から全加算器５２$_3$ のＣｉ端子に出力される。
【００５１】
マルチプレクサ５０$_2$ において、ビットデータＣ$_1$ が論理値「１」のときに、ビットデータＡ$_0$ がｏ端子から全加算器５２$_1$ のｉｎ$_1$ 端子に出力され、ビットデータＣ$_1$ が論理値「０」のときに、ビットデータＢ$_0$ がｏ端子から全加算器５２$_1$ のｉｎ$_1$ 端子に出力される。
マルチプレクサ５０$_4$ において、ビットデータＣ$_1$ が論理値「１」のときに、ビットデータＡ$_1$ がｏ端子から全加算器５２$_2$ のｉｎ$_2$ 端子に出力され、ビットデータＣ$_1$ が論理値「０」のときに、ビットデータＢ$_1$ がｏ端子から全加算器５２$_2$ のｉｎ$_2$ 端子に出力される。
【００５２】
マルチプレクサ５０$_7$ において、ビットデータＣ$_1$ が論理値「１」のときに、ビットデータＡ$_2$ がｏ端子から全加算器５２$_3$ のｉｎ$_2$ 端子に出力され、ビットデータＣ$_1$ が論理値「０」のときに、ビットデータＢ$_2$ がｏ端子から全加算器５２$_3$ のｉｎ$_2$ 端子に出力され

る。
マルチプレクサ５０_{10}において、ビットデータ$Ｃ_{1}$が論理値「１」のときに、ビットデータ$Ａ_{3}$がｏ端子から全加算器５２_{5}の$Ｃｉ_{1}$端子に出力され、ビットデータ$Ｃ_{1}$が論理値「０」のときに、ビットデータ$Ｂ_{3}$がｏ端子から全加算器５２_{5}のＣｉ端子に出力される。
【００５３】
マルチプレクサ５０_{5}において、ビットデータ$Ｃ_{2}$が論理値「１」のときに、ビットデータ$Ａ_{0}$がｏ端子から全加算器５２_{2}の$ｉｎ_{1}$端子に出力され、ビットデータ$Ｃ_{2}$が論理値「０」のときに、ビットデータ$Ｂ_{0}$がｏ端子から全加算器５２_{2}の$ｉｎ_{1}$端子に出力される。
マルチプレクサ５０_{8}において、ビットデータ$Ｃ_{2}$が論理値「１」のときに、ビットデータ$Ａ_{1}$がｏ端子から半加算器５３_{3}の$ｉｎ_{2}$端子に出力され、ビットデータ$Ｃ_{2}$が論理値「０」のときに、ビットデータ$Ｂ_{1}$がｏ端子から半加算器５３_{3}の$ｉｎ_{2}$端子に出力される。
【００５４】
マルチプレクサ５０_{11}において、ビットデータ$Ｃ_{2}$が論理値「１」のときに、ビットデータ$Ａ_{2}$がｏ端子から全加算器５２_{5}の$ｉｎ_{2}$端子に出力され、ビットデータ$Ｃ_{2}$が論理値「０」のときに、ビットデータ$Ｂ_{2}$がｏ端子から全加算器５２_{5}の$ｉｎ_{2}$端子に出力される。
マルチプレクサ５０_{13}において、ビットデータ$Ｃ_{2}$が論理値「１」のときに、ビットデータ$Ａ_{3}$がｏ端子から全加算器５２_{8}の$ｉｎ_{2}$端子に出力され、ビットデータ$Ｃ_{2}$が論理値「０」のときに、ビットデータ$Ｂ_{3}$がｏ端子から全加算器５２_{8}の$ｉｎ_{2}$端子に出力される。
【００５５】
マルチプレクサ５０_{9}において、ビットデータ$Ｃ_{3}$が論理値「１」のときに、ビットデータ$Ａ_{0}$がｏ端子から半加算器５３_{3}の$ｉｎ_{2}$端子に出力され、ビットデータ$Ｃ_{3}$が論理値「０」のときに、ビットデータ$Ｂ_{0}$がｏ端子から半加算器５３_{3}の$ｉｎ_{2}$端子に出力される。
マルチプレクサ５０_{12}において、ビットデータ$Ｃ_{3}$が論理値「１」のときに、ビットデータ$Ａ_{1}$がｏ端子から全加算器５２_{5}の$ｉｎ_{1}$端子に出力され、ビットデータ$Ｃ_{3}$が論理値「０」のときに、ビットデータ$Ｂ_{1}$がｏ端子から全加算器５２_{5}の$ｉｎ_{1}$端子に出力される。
【００５６】
マルチプレクサ５０_{14}において、ビットデータ$Ｃ_{3}$が論理値「１」のときに、ビットデータ$Ａ_{2}$がｏ端子から全加算器５２_{8}の$ｉｎ_{1}$端子に出力され、ビットデータ$Ｃ_{3}$が論理値「０」のときに、ビットデータ$Ｂ_{2}$がｏ端子から全加算器５２_{8}の$ｉｎ_{1}$端子に出力される。
マルチプレクサ５０_{15}において、ビットデータ$Ｃ_{3}$が論理値「１」のときに、ビットデータ$Ａ_{3}$がｏ端子から全加算器５２_{10}の$ｉｎ_{1}$端子に出力され、ビットデータ$Ｃ_{3}$が論理値「０」のときに、ビットデータ$Ｂ_{3}$がｏ端子から全加算器５２_{10}の$ｉｎ_{1}$端子に出力される。
【００５７】
また、入力部５５の$Ｂ_{0}$端子からのビットデータ$Ｂ_{0}$が、半加算器５３_{1}の$ｉｎ_{2}$端子に出力される。
入力部５５の$Ｂ_{1}$端子からのビットデータ$Ｂ_{1}$が、全加算器５２_{1}のＣｉ端子に出力される。
入力部５５の$Ｂ_{2}$端子からのビットデータ$Ｂ_{2}$が、半加算器５３_{2}の$ｉｎ_{2}$端子に出力される。
入力部５５の$Ｂ_{3}$端子からのビットデータ$Ｂ_{3}$が、全加算器５２_{4}の$ｉｎ_{2}$端子に出力される。

【００５８】
また、入力部５５のB_0端子からのビットデータB_0が、インバータ回路54_1で反転された後に、全加算器52_6のin_2端子に出力される。
入力部５５のB_1端子からのビットデータB_1が、インバータ回路54_2で反転された後に、全加算器52_8のＣｉ端子に出力される。
入力部５５のB_2端子からのビットデータB_2が、インバータ回路54_3で反転された後に、全加算器52_{10}のin_2端子に出力される。
入力部５５のB_3端子からのビットデータB_3が、インバータ回路54_4で反転された後に、最終段加算回路４２の全加算器52_{16}のin_1端子に出力される。
【００５９】
〔ステップＳ２〕
半加算器53_1において、ビットデータB_0と、マルチプレクサ50_0からのビットデータとの加算が行われ、当該加算結果の和データがｓ端子からS_0端子に出力され、当該加算結果のキャリーデータがＣｏ端子から最終段加算回路４２の半加算器53_4のin_2端子に出力される。
全加算器52_1において、ビットデータB_1と、マルチプレクサ50_1からのビットデータと、マルチプレクサ50_2からのビットデータとの加算が行われ、当該加算結果の和データがｓ端子から最終段加算回路４２の半加算器53_4のin_1端子に出力され、当該加算結果のキャリーデータがＣｏ端子から最終段加算回路４２の全加算器52_{11}のin_2端子に出力される。
【００６０】
半加算器53_2において、ビットデータB_2と、マルチプレクサ50_3からのビットデータとの加算が行われ、当該加算結果の和データがｓ端子から全加算器52_2のＣｉ端子に出力され、当該加算結果のキャリーデータがＣｏ端子から全加算器52_4のＣｉ端子に出力される。
半加算器53_3において、マルチプレクサ50_8からのビットデータと、マルチプレクサ50_9からのビットデータとの加算が行われ、当該加算結果の和データがｓ端子から全加算器52_3のin_1端子に出力され、当該加算結果のキャリーデータがＣｏ端子から全加算器52_7のin_2端子に出力される。
【００６１】
全加算器52_5において、マルチプレクサ50_{10}からのビットデータと、マルチプレクサ50_{11}からのビットデータと、マルチプレクサ50_{12}からのビットデータとの加算が行われ、当該加算結果の和データがｓ端子から全加算器52_6のin_1端子に出力され、当該加算結果のキャリーデータがＣｏ端子から全加算器52_9のin_2端子に出力される。
全加算器52_8において、インバータ回路54_2からのビットデータB_1^-と、マルチプレクサ50_{13}からのビットデータと、マルチプレクサ50_{14}からのビットデータとのデータとが加算され、当該加算結果の和データがｓ端子から全加算器52_9のin_1端子に出力され、当該加算結果のキャリーデータがＣｏ端子から全加算器52_{10}のＣｉ端子に出力される。
【００６２】
また、全加算器52_2において、半加算器53_2からの和データと、マルチプレクサ50_4からのビットデータと、マルチプレクサ50_5からのビットデータとが加算され、当該加算結果の和データがｓ端子から最終段加算回路４２の全加算器52_{11}のin_1端子に出力され、当該加算結果のキャリーデータがＣｏ端子から最終段加算回路４２の全加算器52_{12}のin_2端子に出力される。
全加算器52_3において、マルチプレクサ50_6からのビットデータと、マルチプレクサ50_7からのビットデータと、半加算器53_3からの和データとが加算され、当該加算結果の和データがｓ端子から全加算器52_4のin_1端子に出力され、当該加算結果のキャリーデータがＣｏ端子から全加算器52_7のＣｉ端子に出力される。
【００６３】

全加算器５２$_6$において、Ｃｉ端子に入力された倫理値「１」と、インバータ回路５４$_1$からのビットデータＢ$_0$ ¯と、全加算器５２$_5$のｓ端子からの和データとの加算が行われ、当該加算結果の和データがｓ端子から全加算器５２$_7$のｉｎ$_1$端子に出力され、当該加算結果のキャリーデータがＣｏ端子から全加算器５２$_9$のＣｉ端子に出力される。
全加算器５２$_{10}$において、全加算器５２$_8$からのキャリーデータと、インバータ回路５４$_3$からのビットデータＢ$_2$ ¯と、マルチプレクサ５０$_{15}$からのビットデータとが加算され、当該加算結果の和データがｓ端子から最終段加算回路４２の全加算器５２$_{15}$のｉｎ$_1$端子に出力され、当該加算結果のキャリーデータがＣｏ端子から全加算器５２$_{16}$のｉｎ$_2$端子に出力される。

【００６４】
また、全加算器５２$_4$において、半加算器５３$_2$からのキャリーデータと、ビットデータＢ$_3$と、全加算器５２$_3$からの和データとの加算が行われ、当該加算結果の和データがｓ端子から最終段加算回路４２の全加算器５２$_{12}$のｉｎ$_1$端子に出力され、当該加算結果のキャリーデータがＣｏ端子から全加算器５２$_{13}$のｉｎ$_2$端子に出力される。
全加算器５２$_7$において、全加算器５２$_3$からのキャリーデータと、半加算器５３$_3$からのキャリーデータと、全加算器５２$_6$からの和データとが加算され、当該加算結果の和データがｓ端子から最終段加算回路４２の全加算器５２$_{13}$のｉｎ$_1$端子に出力され、当該加算結果のキャリーデータがＣｏ端子から最終段加算回路４２の全加算器５２$_{14}$のｉｎ$_2$端子に出力される。

【００６５】
全加算器５２$_9$において、全加算器５２$_6$からのキャリーデータと、全加算器５２$_5$からのキャリーデータと、全加算器５２$_8$からの和データとが加算され、当該加算結果の和データがｓ端子から最終段加算回路４２の全加算器５２$_{14}$のｉｎ$_1$端子に出力され、当該加算結果のキャリーデータがＣｏ端子から最終段加算回路４２の全加算器５２$_{15}$のｉｎ$_2$端子に出力される。

【００６６】
〔ステップＳ３〕
最終段加算回路４２において、以下に示す処理が行われる。
先ず、半加算器５３$_4$において、半加算器５３$_1$からのキャリーデータと、全加算器５２$_1$からの和データとの加算が行われ、その和データが出力部５７のＳ$_1$端子からビットデータＳ$_1$として出力され、そのキャリーデータが全加算器５２$_{11}$のＣｉ端子に出力される。
次に、全加算器５２$_{11}$において、半加算器５３$_4$からのキャリーデータと、全加算器５２$_1$からのキャリーデータと、全加算器５２$_2$からの和データとが加算され、その和データが出力部５７のＳ$_2$端子からビットデータＳ$_2$として出力され、そのキャリーデータが全加算器５２$_{12}$のＣｉ端子に出力される。

【００６７】
次に、全加算器５２$_{12}$において、全加算器５２$_{11}$からのキャリーデータと、全加算器５２$_2$からのキャリーデータと、全加算器５２$_4$からの和データとが加算され、その和データが出力部５７のＳ$_3$端子からビットデータＳ$_3$として出力され、そのキャリーデータが全加算器５２$_{13}$のＣｉ端子に出力される。
次に、全加算器５２$_{13}$において、全加算器５２$_{12}$からのキャリーデータと、全加算器５２$_4$からのキャリーデータと、全加算器５２$_7$からの和データとが加算され、その和データが出力部５７のＳ$_4$端子からビットデータＳ$_4$として出力され、そのキャリーデータが全加算器５２$_{14}$のＣｉ端子に出力される。

【００６８】
次に、全加算器５２$_{14}$において、全加算器５２$_{13}$からのキャリーデータと、全加算器５２$_7$からのキャリーデータと、全加算器５２$_9$からの和データとが加算され、その和データが出力部５７のＳ$_5$端子からビットデータＳ$_5$として出力され、そのキャリーデータが全加算器５２$_{15}$のＣｉ端子に出力される。

次に、全加算器５２$_{15}$において、全加算器５２$_{14}$からのキャリーデータと、全加算器５２$_{9}$からのキャリーデータと、全加算器５２$_{10}$からの和データとが加算され、その和データが出力部５７のＳ$_{6}$端子からビットデータＳ$_{6}$として出力され、そのキャリーデータが全加算器５２$_{16}$のＣｉ端子に出力される。

【００６９】

次に、全加算器５２$_{16}$において、全加算器５２$_{15}$からのキャリーデータと、全加算器５２$_{10}$からのキャリーデータと、インバータ５４$_{4}$からのビットデータB_3^-とが加算され、その和データが出力部５７のＳ$_{7}$端子からビットデータＳ$_{7}$として出力され、そのキャリーデータがインバータ５４$_{5}$を介して、出力部５７のＳ$_{8}$端子からビットデータＳ$_{8}$として出力される。

ここで、ビットデータＳ$_{8}$は符号を示しており、ビットデータＳ$_{8}$が論理値「１」のときに、演算「（Ａ－Ｂ）×Ｃ」の演算結果であるデータＳが負であることを示し、ビットデータＳ$_{8}$が論理値「０」のときに、データＳが正であることを示している。

なお、ビットデータＳ$_{8}$が論理値「１」のときに、ビットデータＳ$_{0}$～Ｓ$_{7}$は、演算「（Ａ－Ｂ）×Ｃ」の演算結果の２の補数値を示している。

【００７０】

図１に示す演算処理装置４０では、例えば、図５に示すように、データＡ（Ａ$_{0}$，Ａ$_{1}$，Ａ$_{2}$，Ａ$_{3}$）＝（０，０，０，１）、Ｂ（Ｂ$_{0}$，Ｂ$_{1}$，Ｂ$_{2}$，Ｂ$_{3}$）＝（０，０，１，０）およびＣ（Ｃ$_{0}$，Ｃ$_{1}$，Ｃ$_{2}$，Ｃ$_{3}$）＝（１，１，１，１）を入力した場合には、データＳ（Ｓ$_{0}$，Ｓ$_{1}$，Ｓ$_{2}$，Ｓ$_{3}$，Ｓ$_{4}$，Ｓ$_{5}$，Ｓ$_{6}$，Ｓ$_{7}$，Ｓ$_{8}$）＝（０，１，０，１，１，０，１，０，０）を出力する。

【００７１】

以上説明したように、演算処理装置４０によれば、図１に示すように、マルチプレクサ５０$_{0}$～５０$_{15}$、全加算器（ＦＡ）５２$_{1}$～５２$_{16}$、半加算器（ＨＡ）５３$_{1}$～５３$_{4}$、インバータ回路５４$_{1}$～５４$_{4}$、入力部５５，５６および出力部５７によって、演算「（Ａ－Ｂ）×Ｃ」を行うことができ、前述した図６，図７および図１０に示す従来の演算処理装置１のように５ビットの乗算器を用いる必要がなく、回路規模を大幅に縮小できる。

【００７２】

また、演算処理装置４０によれば、演算のクリティカルパスは、マルチプレクサ５０$_{0}$、半加算器５３$_{1}$，５３$_{4}$、全加算器５２$_{11}$，５２$_{12}$，５２$_{13}$，５２$_{14}$，５２$_{15}$，５２$_{16}$およびインバータ回路５４$_{5}$となり、前述した従来の演算処理装置１に比べて、クリティカルパスを短縮でき、演算時間を短縮できる。

【００７３】

本発明の実施形態は、上述したものには限定されない。

例えば、図１では、４ビットのデータＡ，Ｂ，Ｃを用いて演算「（Ａ－Ｂ）×Ｃ」を行う場合を例示したが、本発明は、２以上の全ての整数ｎについて、ｎビットのデータＡ，Ｂ，Ｃを用いて演算「（Ａ－Ｂ）×Ｃ」を行う場合にも適用できる。

この場合には、下記式（９）に基づいて演算が行われる。

【００７４】

【数９】

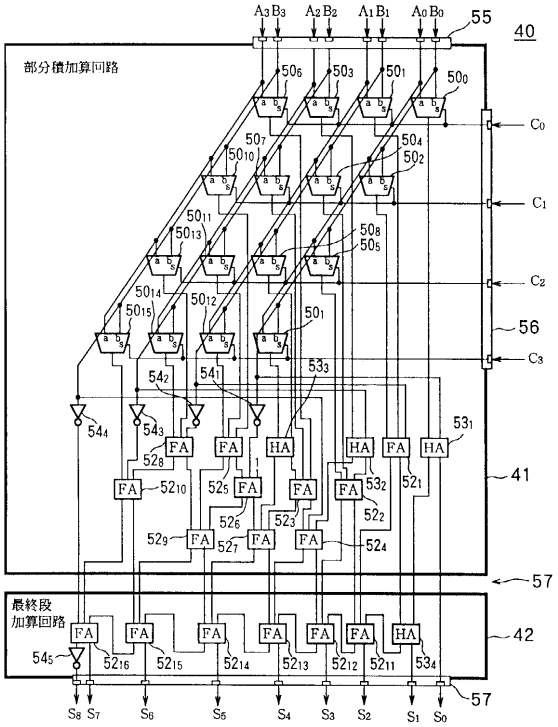
$$S = \left(\sum_{j=0}^{j=n-1} \sum_{i=0}^{i=n-1} 2^{i+j} \cdot P_{ij} \right) - 2^n \times B + B \qquad \cdots (9)$$

【００７５】

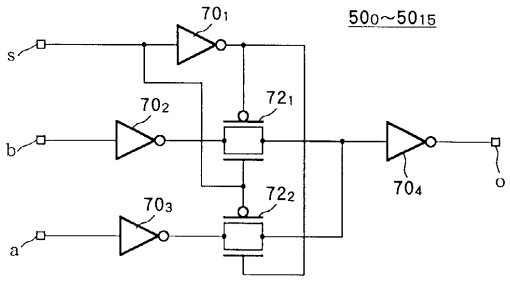
上記式（９）において、第１項の演算は、本発明のビットデータ選択手段としてのマルチプレクサを用いてビットデータＣ$_{j}$が論理値「１」のときにビットデータＡ$_{i}$を選択し、ビットデータＣ$_{j}$が論理値「０」のときにビットデータＢ$_{i}$を選択し、当該選択したデータを（ｉ＋ｊ）ビット目に加算するように、各ビット毎に、下位のビットからのキャリーデータを含めて加算して行われる。

【００７６】
また、上記式（９）において、「－２^{n} ×Ｂ」の演算は、データＢの２補数を求めるために、データＢのレベルを反転し、このレベル反転したデータのＬＳＢに「１」を加算し、当該加算結果をＭＳＢに向けてｎビットだけシフトして行われる。当該シフトは、例えば、前記加算結果の各ビットデータB_0 ～B_{n-1} を、それぞれ演算結果Ｓのｎ～２ｎ－１ビット目に対応する加算を行う加算器に入力して実現される。
そして、上記式（９）の第１項の演算結果と、「－２^{n} ×Ｂ」の演算結果と、データＢとを加算することで、上記式（９）の演算が行われる。
【００７７】
【発明の効果】
以上説明したように、本発明の演算処理装置およびその方法によれば、演算「（Ａ－Ｂ）×Ｃ」の演算時間を短縮化できる。
また、本発明の演算処理装置によれば、演算「（Ａ－Ｂ）×Ｃ」を行う装置の規模を縮小できる。
【図面の簡単な説明】
【図１】図１は、本実施形態の演算処理装置の構成図である。
【図２】図２は、図１に示すマルチプレクサの構成図である。
【図３】図３は、図１に示す演算処理装置が採用する演算方法を説明するための図である。
【図４】図４は、図１に示す演算処理装置の動作を説明するためのフローチャートである。
【図５】図５は、図１に示す演算処理装置における具体的な動作を説明するための図である。
【図６】図６は、演算「（Ａ－Ｂ）×Ｃ」を行う従来の演算処理装置の構成図である。
【図７】図７は、図６に示す減算器の構成図である。
【図８】図８は、図７に示す全加算器（ＦＡ）の構成図である。
【図９】図９は、図２に示す乗算器が採用するバウウーリ(Baugh Wooly) 方式による２の補数乗算を説明するための図である。
【図１０】図１０は、図９に示す２の補数乗算を行う乗算器の構成図である。
【図１１】図１１は、図１０に示す半加算器の構成図である。
【符号の説明】
４０…演算処理装置、５０_{0} ～５０_{15}…マルチプレクサ、５２_{1} ～５２_{16}…全加算器（ＦＡ）、５３_{1} ～５３_{4} …半加算器（ＨＡ）、５４_{1} ～５４_{4} …インバータ回路、５５，５６…入力部、５７…出力部

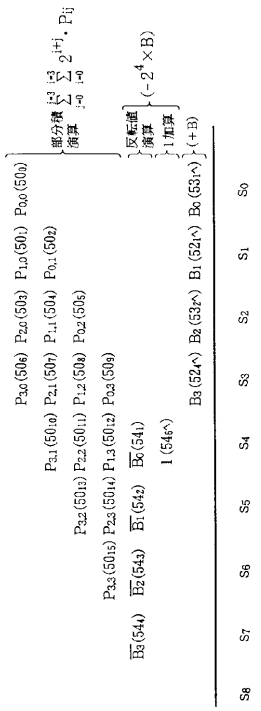
【図１】


部分積加算回路
最終段
加算回路
FA
HA
40
41
42
55
56
57

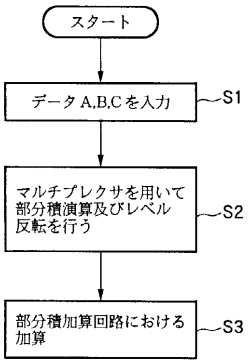
【図２】


s
b
a
o
50₀～50₁₅

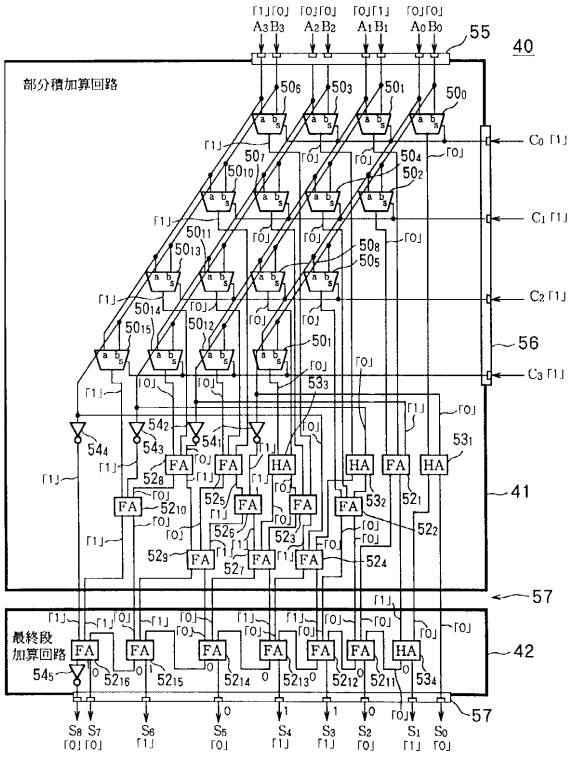
【図３】



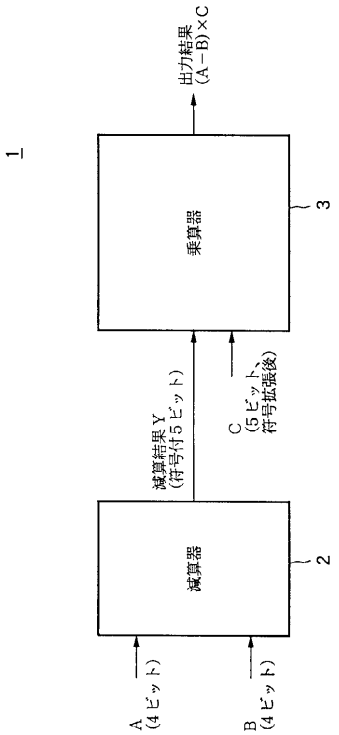
【図４】


スタート
データA,B,Cを入力
S1
マルチプレクサを用いて
部分積演算及びレベル
反転を行う
S2
部分積加算回路における
加算
S3

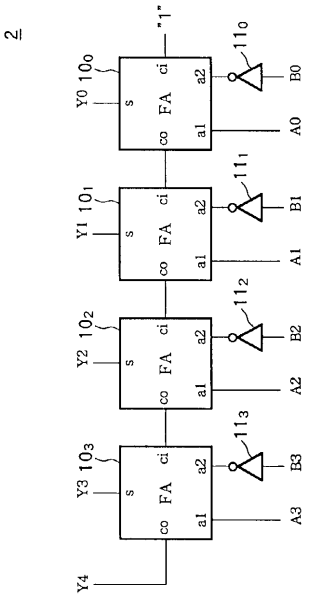
【図５】


部分積加算回路
最終段
加算回路
FA
HA
40
41
42
55
56
57

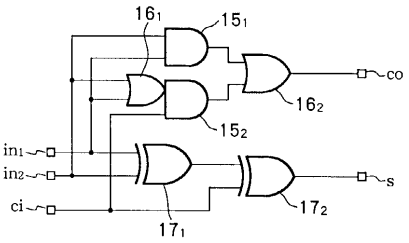
【図６】


1
出力結果
(A−B)×C
乗算器
3
減算結果Y
(符号付5ビット)
C
(5ビット、
符号拡張後)
減算器
2
A
(4ビット)
B
(4ビット)

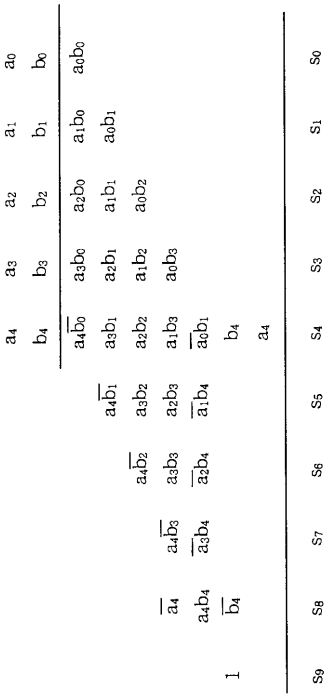
【図７】


2
"1"
FA
ci
s
co
a2
a1

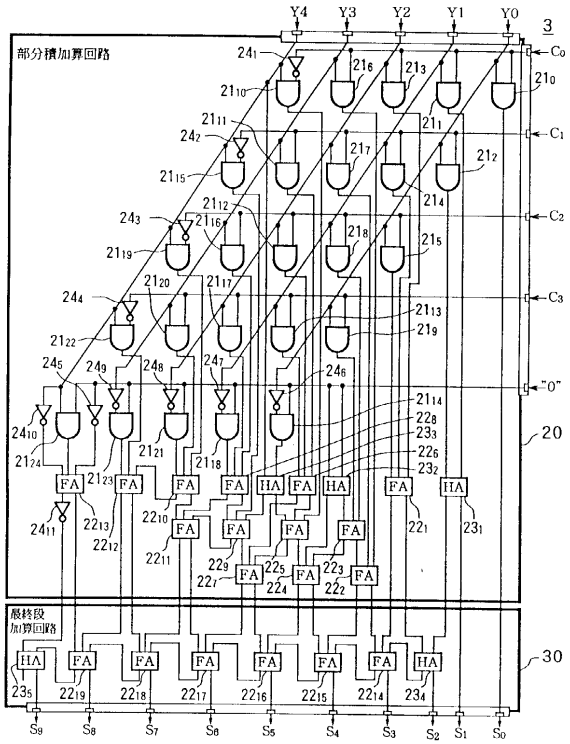
【図８】


in1
in2
ci
co
s

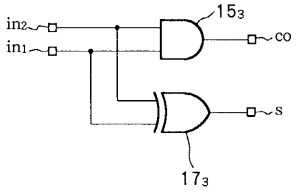
【図９】


a0 b0 a0b0
a1 b1 a1b0 a0b1
a2 b2 a2b0 a1b1 a0b2
a3 b3 a3b0 a2b1 a1b2 a0b3
a4 b4 a4b0 a3b1 a2b2 a1b3 a0b1 b4 a4
a4b1 a3b2 a2b3 a1b4
a4b2 a3b3 a2b4
a4b3 a3b4
a4 a4b4 b4
1
S0 S1 S2 S3 S4 S5 S6 S7 S8 S9

【図１０】


部分積加算回路
最終段
加算回路
Y4 Y3 Y2 Y1 Y0
3
C0 C1 C2 C3
"0"
20
30
FA
HA
S9 S8 S7 S6 S5 S4 S3 S2 S1 S0

【図１１】


in2
in1
15
co
s
17

フロントページの続き

(58)調査した分野(Int.Cl., ＤＢ名)

G06F 7/535

G06F 7/62

WPI(DIALOG)