(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3544821号
(P3544821)
(45)発行日　平成16年7月21日(2004.7.21)　(24)登録日　平成16年4月16日(2004.4.16)

(51)Int.Cl.[7]　F I
G06F 7/52　G06F 7/52　320A
G06F 7/38　G06F 7/38　A

請求項の数 12　(全 15 頁)

(21)出願番号　特願平9-88187
(22)出願日　平成9年4月7日(1997.4.7)
(65)公開番号　特開平10-283164
(43)公開日　平成10年10月23日(1998.10.23)
審査請求日　平成13年6月4日(2001.6.4)

(73)特許権者　000003078
株式会社東芝
東京都港区芝浦一丁目１番１号
(74)代理人　100083806
弁理士　三好　秀和
(74)代理人　100068342
弁理士　三好　保男
(74)代理人　100100712
弁理士　岩▲崎▼　幸邦
(74)代理人　100100929
弁理士　川又　澄雄
(72)発明者　矢野　直佳
神奈川県川崎市幸区堀川町５８０番１号
株式会社東芝　半導体システム技術センター内

最終頁に続く

(54)【発明の名称】除算回路

(57)【特許請求の範囲】
【請求項１】
符号付き除算を行う除算回路であって、
加算又は減算を行う加減算器と、
加減算の結果得られる仮の部分剰余が０であるか否かを判別する判別器と、
被除数の符号と除数の符号を比較する第１の比較器と、
前記加減算器の演算結果の符号と被除数の符号、及び前記判別器の判別結果にしたがって、商デジットを決定し、新しい部分剰余を決定する制御回路を具備し、
前記加減算器は、前記第１の比較器の比較結果にしたがって部分剰余の所定ビットデータと除数に加算又は減算を施し、
前記制御回路は、前記加減算器の演算結果の符号と被除数の符号及び前記判別器による判別結果に基づいて商を決定し、商デジットの値にしたがって部分剰余の所定ビットのデータ又は前記加減算器の演算結果から得られる仮の部分剰余の所定ビットのデータを選択して新しい部分剰余の所定ビットのデータとする動作を、所定の回数反復して行うことにより符号付き除算を行う
ことを特徴とする除算回路。
【請求項２】
前記加減算器は、前記第１の比較器の比較結果として、被除数の符号と除数の符号が同一の場合には、部分剰余の所定ビットのデータから除数を減算し、
前記第１の比較器の比較結果として、被除数の符号と除数の符号が異なる場合には、部分

剰余のうちの所定ビットのデータと除数を加算する
ことを特徴とする請求項１記載の除算回路。
【請求項３】
被除数と除数の符号が異なる場合には、商を得るために、得られた各ビットの商デジットの補数をとる
ことを特徴とする請求項１又は２記載の除算回路。
【請求項４】
符号付き除算を行う除算回路であって、
加算又は減算を行う加減算器と、
加減算の結果得られる仮の部分剰余が０であるか否かを判別する判別器と、
被除数の符号と除数の符号を比較する第１の比較器と、　前記加減算器の演算結果の符号と被除数の符号、及び前記判別器の判別結果にしたがって、商デジットを決定し、次サイクル時の加算又は減算を前記加減算器に指示する制御回路を具備し、
前記加減算器は、前記第１の比較器の比較結果にしたがって、符号拡張された被除数の所定ビットのデータと除数に加算又は減算を施し、
前記制御回路は、前記加減算器の演算結果の符号と被除数の符号及び前記判別器による判別結果に基づいて商デジットを決定した後、前サイクルで得られた仮の部分剰余のうちの所定ビットのデータと除数を、前サイクルで得られた商に基づく前記制御回路の指示にしたがって前記加減算器により加算又は減算する動作と、その加算又は減算の演算結果にしたがって、前記加減算器の演算結果の符号と被除数の符号及び前記判別器による判別結果に基づいて商を決定する動作を、所定の回数反復して行うことにより符号付き除算を行うことを特徴とする除算回路。
【請求項５】
前記加減算器は、前記第１の比較器の比較結果として、被除数の符号と除数の符号が同一の場合には、前記符号拡張された被除数のうち所定ビットのデータから除数を減算し、
前記第１の比較器の比較結果として、被除数の符号と除数の符号が異なる場合には、前記符号拡張された被除数のうちの所定ビットのデータと除数を加算することを特徴とする請求項４記載の除算回路。
【請求項６】
最終剰余を得るために、演算の最後のサイクルで得られた商に応じて剰余の補正を行い、
被除数と除数の符号が異なる場合には、商を得るために、得られた各ビットの商デジットの補数をとる
ことを特徴とする請求項４又は５記載の除算回路。
【請求項７】
前記制御回路は、
前記加減算器の出力の符号と被除数の符号が同一の場合は、商デジットとして１を得て、
前記加減算器の出力の符号が被除数の符号と異なり、かつ前記判別器の判別結果として前記仮の部分剰余が０である場合には、商デジットとして１を得て、
前記加減算器の出力の符号が被除数の符号と異なり、かつ前記判別器の判別結果として前記仮の部分剰余が０でない場合には、商デジットして０を得ることを特徴とする請求項１，２，４及び５のいずれか１項に記載の除算回路。
【請求項８】
被除数を格納する被除数レジスタと、
被除数の所定ビットのデータ及び仮の部分剰余の所定ビットのデータを入力し、前記制御回路における商デジットにしたがって、何れかを選択して出力するセレクタと、
前記セレクタの出力を格納する剰余レジスタとを具備し、
前記制御回路は、前記剰余レジスタに格納された内容の所定ビットに前記被除数レジスタに格納された内容の上位ビットを連結した連結値を、新たな被除数の所定ビットデータとして前記加減算器及び前記セレクタに渡しつつ、前記加減算器に加算又は減算を所定の回数反復して行わせる

ことを特徴とする請求項１，２，４及び５のいずれか１項に記載の除算回路。
【請求項９】
前記制御回路は、
前記加減算器に加算又は減算を所定の回数反復して行わせる際に、前記被除数レジスタに格納された値の所定ビットを順に所定ビットずつ上位側にシフトし、商デジットを前記被除数レジスタの下位側から順次格納する
ことを特徴とする請求項８記載の除算回路。
【請求項１０】
被除数を格納するレジスタと商を格納するレジスタを兼用する兼用レジスタを具備し、
前記兼用レジスタは、除算実行中には格納された値の所定ビットを順に所定ビットずつ上位側にシフトし、商デジットは兼用レジスタの下位側から順次格納される
ことを特徴とする請求項１，２，４，５及び９のいずれか１項に記載の除算回路。
【請求項１１】
符号なし除算を行う場合には、被除数ならびに除数の上位側に０を拡張し、符号付き除算を行う場合には、被除数ならびに除数にそれぞれの符号ビットを符号拡張し、符号付き除算及び符号なし除算を同一の除算回路で行う
ことを特徴とする請求項１，２，４，５及び９のいずれか１項に記載の除算回路。
【請求項１２】
引き放し法を用いて除算を行う除算回路であって、
最初は、被除数と除数が同じ符号の場合には減算を、被除数と除数が異なる符号の場合には加算を行い、それ以降は、被除数と除数が同じ符号で、商デジットが１の場合には、次のステップで減算を、商デジットが０の場合には次のステップでは加算を行い、被除数と除数が異なる符号で、商デジットが１の場合には、次のステップでは加算を、商デジットが０の場合には次のステップでは減算を行うという動作を繰り返すことによって除算を行い、被除数の符号と部分剰余の符号との比較結果、ならびに部分剰余が０であるか否かの判別結果に基づいて商の各ビットの商デジットを決定することを特徴とする除算回路。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】
本発明は、除算回路、特に符号付除算を行なう除算回路に関する。
【０００２】
【従来の技術】
従来、２の補数表示された符号付きの被除数及び除数の除算を行う場合に、正確な商及び剰余を求めようとすると、負の被除数及び／又は除数を正に変換して除算を行なっていた。この代表的な除算アルゴリズムとして、回復型（引き戻し法）の除算と非回復型（引き放し法）の除算の２例を挙げて説明する。
【０００３】
まずはじめに、従来の回復型除算のアルゴリズムを、図６に示す除算回路の構成を参照して説明する。
【０００４】
図６において、ｎビットの被除数をａ、ｎビットの除数をｂとすると、除算回路は、除算実行中に部分剰余の上位側を格納し除算終了時に剰余を格納するｎビットの剰余レジスタ１０１と、除算開始時に被除数を格納し除算実行中に部分剰余の下位側と商を順次格納するｎビットの被除数＆商レジスタ１０２と、除数を格納するｎビットの除数レジスタ１０３と、ｎビットの減算器１０４と、剰余レジスタ１０１と被除数＆商レジスタ１０２の内容を連結して上位側（図面左方向）に１ビットシフトし最上位ビットをシフトアウトした上位ｎビットのデータＰ’又は減算器１０４の出力を選択して剰余レジスタ１０１に与えるセレクタ１０５と、減算器１０４の減算結果を受けて商を決定し、セレクタ１０５を選択制御する制御回路１０６を備えて構成されている。
【０００５】

次に、このような構成の除算回路を用いた除算動作を動作サイクルにしたがって説明する。

【０００６】

まず、被除数ａ，除数ｂの双方とも負の場合には補数をとり正の数に変換し、負でない場合にはそのままとし、この値をａ’、ｂ’とする（第１サイクル、第２サイクル、作用１）。

【０００７】

次に、被除数ａ’を被除数＆商レジスタ１０２に、除数ｂ’を除数レジスタ１０３にロードする。剰余レジスタ１０１には、全ビット０をロードする（第３サイクル、作用２）。

【０００８】

次に、剰余レジスタ１０１、被除数＆商レジスタ１０２をそれぞれ１ビット左にシフトし、シフト後の両レジスタの連結した内容の上位ｎビットのデータをＰ’とする（第４サイクル、作用３－１）。続いて、減算器１０４によってデータＰ’から除数レジスタ１０３の出力を減じる（第４サイクル、作用３－２）。続いて、被除数＆商レジスタ１０２は出力を左に１ビットシフトして格納する。減算器１０４の出力が負の時は被除数＆商レジスタ１０２の最下位に商として０を書き込み、減算器１０４の出力が負でなければ被除数＆商レジスタ１０２の最下位に商として１を書き込む（第４サイクル、作用３－３）。続いて、減算器１０４の出力が負でなければ減算器１０４の出力を剰余レジスタ１０１に格納し、減算器１０４の出力が負の時は剰余レジスタ１０１にデータＰ’を格納する（引き戻し）（第４サイクル、作用３－４）。

【０００９】

次に、上記第４サイクルの作用３－１～３－４をさらに（ｎ－１）回繰り返して行う（第５サイクル～第（ｎ＋３）サイクル）。

【００１０】

次に、被除数と除数の符号が同一の場合は、被除数＆商レジスタ１０２の出力は商として、そのまま被除数＆商レジスタ１０２に保存される。一方、被除数と除数の符号が異なる場合には、被除数＆商レジスタ１０２の出力の補数をとり、被除数＆商レジスタ１０２に書き戻される（第（ｎ＋４）サイクル、作用５）。

【００１１】

次に、被除数が負でない場合は、剰余レジスタ１０１の出力は剰余としてそのまま保存される。一方、被除数が負の場合には、剰余レジスタ１０１の出力の補数をとり剰余レジスタ１０１に書き戻される（第（ｎ＋５）サイクル、作用６）。

【００１２】

最後に、被除数＆商レジスタ１０２には商が格納され、剰余レジスタ１０１には剰余が格納され、符号付の除算が終了する（第（ｎ＋６）サイクル、作用７）。

【００１３】

このようにして除算処理が行われ、例えば被除数ａ＝１００１（２進数）（１０進数で－７を示す）と除数ｂ＝００１１（２進数）（１０進数で３を示す）の除算例を図７に示す。なお、図７において、剰余レジスタ１０１の内容をＰと表記し、被除数＆商レジスタ１０２の内容をＡと表記し、除数レジスタ１０３の内容をＢと表記している。

【００１４】

次に、従来の非回復型除算のアルゴリズムを、図８に示す除算回路の構成を参照して説明する。

【００１５】

上述した従来の回復型除算では、部分剰余が負になった場合は、（剰余＋除数）×２－除数という演算を行っているのと同じであった。そこで、（剰余＋除数）×２－除数は剰余×２＋除数であるので、部分剰余が負になった場合には、そのまま部分剰余を格納して左に１ビットシフトし、部分剰余と除数の減算に代えて加算を行ってもよい。この手法では回復型除算に比べてサイクル当たりの演算時間を短縮することができる。このような手法を用いたのが非回復型除算のアルゴリズムである。

【００１６】
図８において、ｎビットの被除数をａ、ｎビットの除数をｂとすると、除算回路は、除算実行中に部分剰余の上位側を格納し除算終了時に剰余を格納するｎビットの剰余レジスタ１１１と、除算開始時に被除数を格納し除算実行中に部分剰余の下位側と商を順次格納するｎビットの被除数＆商レジスタ１１２と、除数を格納するｎビットの除数レジスタ１１３と、ｎビットの加減算器１１４と、加減算器１１４の演算結果を受けて商を決定し、加減算器１１４の演算結果にしたがって加減算器１１４に加算又は減算を指示する制御回路１１５を備えて構成されている。
【００１７】
次に、このような構成の除算回路を用いた除算動作を動作サイクルにしたがって説明する。
【００１８】
まず、被除数ａ，除数ｂの双方とも負の場合には補数をとり正の数に変換し、負でない場合にはそのままとし、この値をａ’、ｂ’とする（第１サイクル、第２サイクル、作用１）。
【００１９】
次に、被除数ａ’を被除数＆商レジスタ１１２に、除数ｂ’を除数レジスタ１１３にロードする。剰余レジスタ１１１には全ビット０をロードする（第３サイクル、作用２）。
【００２０】
次に、剰余レジスタ１１１、被除数＆商レジスタ１１２をそれぞれ１ビット左にシフトし、シフト後の両レジスタの連結した内容の上位ｎビットのデータをＰ’とする（第４サイクル、作用３－１）。続いて、加減算器１１４によってデータＰ’から除数レジスタ１１３の出力を減じる（第４サイクル、作用３－２）。続いて、被除数＆商レジスタ１１２は、出力を左に１ビットシフトして格納する。被除数＆商レジスタ１１２の最下位ビットに格納する値は、加減算器１１４の出力が負の時は被除数＆商レジスタ１１２の最下位に商として０を書き込み、加減算器１１４の出力が負でなければ被除数＆商レジスタ１０２の最下位に商として１を書き込む（第４サイクル、作用３－３）。続いて、加減算器１１４の出力を剰余レジスタ１１１に格納する（第４サイクル、作用３－４）。
【００２１】
次のサイクルにおいて、剰余レジスタ１１１の出力が負の場合は、剰余レジスタ１１１，被除数＆商レジスタ１１２をそれぞれ１ビット左にシフトする。シフト後の両レジスタの連結した内容の上位ｎビットのデータＰ’と除数レジスタ１１３の出力を加減算器１１４により加算する。一方、剰余レジスタ１１１の出力が負でない場合には、剰余レジスタ１１１ならびに被除数＆商レジスタ１１２をそれぞれ１ビット左にシフトし、シフト後の両レジスタの連結した内容の上位ｎビットのデータＰ’から除数レジスタ１１３の出力を加減算器１１４により減じる（第５サイクル、作用４－１）。続いて、被除数＆商レジスタ１１２は出力を左に１ビットシフトして格納する。被除数＆商レジスタ１１２の最下位ビットには、加減算器１１４の出力が負の時は商として０を書き込み、加減算器１１４の出力が負でなければ商として１を書き込む（第５サイクル、作用４－２）。続いて、加減算器１１４の出力を剰余レジスタ１１１に格納する（第５サイクル、作用４－３）。
【００２２】
次に、上述した第５サイクルでの作用４－１～４－３をさらに（ｎ－２）回繰り返して行う（第６サイクル～第（ｎ＋３）サイクル）。
【００２３】
次に、剰余レジスタ１１１の出力が負でなければ、剰余レジスタ１１１はそのままの値を保持する。一方、剰余レジスタ１１１の出力が負であれば、加減算器１１４により剰余レジスタ１１１の出力と除数レジスタ１１３の出力を加算して剰余レジスタ１１１に格納する（第（ｎ＋４）サイクル、作用６）。
【００２４】
次に、被除数と除数の符号が同一の場合は、被除数＆商レジスタ１１２の出力はそのまま

被除数＆商レジスタ１１２に保存される。被除数と除数の符号が異なる場合には、被除数＆商レジスタ１１２の出力の補数をとり、被除数＆商レジスタ１１２に書き戻される（第（ｎ＋５）サイクル、作用７）。

【００２５】

次に、被除数が負でない場合は、剰余レジスタ１１１の出力はそのまま保持される。被除数が負の場合には、剰余レジスタ１１１の出力の補数をとり、剰余レジスタ１１１に書き戻される（第（ｎ＋６）サイクル、作用８）。

【００２６】

最後に、被除数＆商レジスタ１１２には商が格納され、剰余レジスタ１１１には剰余が格納され、符号付の除算が終了する（第（ｎ＋７）サイクル、作用９）。

【００２７】

このようにして除算処理が行われ、例えば前述した従来例と同様に被除数ａ＝１００１（２進数）（１０進数で－７を示す）と除数ｂ＝００１１（２進数）（１０進数で３を示す）の除算例を図９に示す。なお、図９において、剰余レジスタ１１１の内容をＰと表記し、被除数＆商レジスタ１１２の内容をＡと表記し、除数レジスタ１１３の内容をＢと表記している。

【００２８】

【発明が解決しようとする課題】

以上説明したように、符号付の被除数及び除数の除算において、商及び剰余を正確に求めようとすると、被除数及び／又は除数が負の値の場合には、負の値の被除数及び／又は除数の２の補数をとり負の値を正の値に変換し、正の値の被除数及び除数で除算を行っていた。このため、被除数ならびに除数の双方が負の値の場合には、両者とも２の補数をとらなければならないため、補数をとるための時間が除算時間に加わり、除算時間が長くなるという不具合を招いていた。また、被除数が負の値の場合には、剰余も２の補数をとり補正しなければならず、さらに除算時間が長くなっていた。

【００２９】

そこで、この発明は、上記に鑑みてなされたものであり、その目的とするところは、被除数及び／又は除数の負の値を正の値に変換することなく符号付の除算を行い、正確な商と剰余を求めることができる除算回路を提供することにある。

【００３０】

【課題を解決するための手段】

上記目的を達成するために、課題を解決する第１の手段は、符号付き除算を行う除算回路であって、加算又は減算を行う加減算器と、加減算の結果得られる仮の部分剰余が０であるか否かを判別する判別器と、被除数の符号と除数の符号を比較する第１の比較器と、前記加減算器の演算結果の符号と被除数の符号、及び前記判別器の判別結果にしたがって、商デジットを決定し、新しい部分剰余を決定する制御回路を具備し、前記加減算器は、前記第１の比較器の比較結果にしたがって部分剰余の所定ビットデータと除数に加算又は減算を施し、前記制御回路は、前記加減算器の演算結果の符号と被除数の符号及び前記判別器による判別結果に基づいて商を決定し、商デジットの値にしたがって部分剰余の所定ビットのデータ又は前記加減算器の演算結果から得られる仮の部分剰余の所定ビットのデータを選択して新しい部分剰余の所定ビットのデータとする動作を、所定の回数反復して行うことにより符号付き除算を行うことを特徴とする。

【００３１】

第２の手段は、符号付き除算を行う除算回路であって、加算又は減算を行う加減算器と、加減算の結果得られる仮の部分剰余が０であるか否かを判別する判別器と、被除数の符号と除数の符号を比較する第１の比較器と、前記加減算器の演算結果の符号と被除数の符号、及び前記判別器の判別結果にしたがって、商デジットを決定し、次サイクル時の加算又は減算を前記加減算器に指示する制御回路を具備し、前記加減算器は、前記第１の比較器の比較結果にしたがって、符号拡張された被除数の所定ビットのデータと除数に加算又は減算を施し、前記制御回路は、前記加減算器の演算結果の符号と被除数の符号及び前記判

別器による判別結果に基づいて商デジットを決定した後、前サイクルで得られた仮の部分剰余のうちの所定ビットのデータと除数を、前サイクルで得られた商に基づく前記制御回路の指示にしたがって前記加減算器により加算又は減算する動作と、その加算又は減算の演算結果にしたがって、前記商デジットの決定と同様に商を決定する動作を、所定の回数反復して行うことにより符号付き除算を行うことを特徴とする。

【００３３】

第４の手段は、引き放し法を用いて除算を行う除算回路であって、最初は、被除数と除数が同じ符号の場合には減算を、被除数と除数が異なる符号の場合には加算を行い、それ以降は、被除数と除数が同じ符号で、商デジットが１の場合には、次のステップで減算を、商デジットが０の場合には次のステップでは加算を行い、被除数と除数が異なる符号で、商デジットが１の場合には、次のステップでは加算を、商デジットが０の場合には次のステップでは減算を行うという動作を繰り返すことによって除算を行い、被除数の符号と部分剰余の符号と部分剰余が０であるか否かに基づいて商の各ビットの商デジットを決定することを特徴とする。

【００３４】

【発明の実施の形態】

以下、図面を用いてこの発明の実施形態を説明する。

【００３５】

図１はこの発明の一実施形態に係る除算回路の構成を示す図である。

【００３６】

この実施形態は、前述した従来の回復型の除算アルゴリズムを基本にして符号付の被除数ならびに除数により除算を行うものであり、前述した回復型の除算アルゴリズムと大きく異なるところは、商の判定手法である。

【００３７】

商として下位からｋビット目に１が立つか否かの判定は、その時点での部分剰余の一部と除数×2^{k-1}　との絶対値の比較による。部分剰余の一部の絶対値が、除数×2^{k-1}　の絶対値よりも小さい場合には商は０となり、それ以外は商は１となる。絶対値の比較は、部分剰余と除数とが同符号の場合は、部分剰余の一部から除数の減算を行い、異符号の場合には加算を行えばよい。その結果が、部分剰余と同符号であれば部分剰余の一部の絶対値が大きいことになり、商として１が得られる。一方、演算結果が部分剰余と異符号になれば商は０になる。これに対して、加減算結果の符号の比較で決定できないのは、加減算結果の絶対値が０の場合である。この場合には、演算結果の符号によらず部分剰余の一部と除数の絶対値は等しいことになり商は１となる。この実施形態の除算回路では、この状態を検出するための従来にはない本発明に特徴的な構成を備えている。以下、この実施形態の除算回路の具体的な構成ならびに作用を説明する。

【００３８】

図１において、この実施形態の除算回路は、２の補数表現された符号付きｎビットの被除数をａ、ｍビットの除数をｂとすると、符号付の除数を格納するｍビットの除数レジスタ１と、除算開始時に符号付の被除数を格納し除算実行中に部分剰余の下位側を格納してシフトするｎビットの被除数レジスタ２と、除算実行中に部分剰余の上位側を格納してシフトし除算終了時に剰余を格納するｍビットの剰余レジスタ３と、商を順次格納するｎビットの商レジスタ４と、加算又は減算を行なうｍビットの加減算器５と、剰余レジスタ３に格納された内容の最上位ビットを除く値の下位に被除数レジスタ２に格納された内容の上位所定ビットを連結した連結値又は加減算器５の演算結果を選択して剰余レジスタ３に出力するセレクタ６と、被除数レジスタ２の出力の下位（ｎ－１）ビットがすべて０であるか否かを判別する判別器（ａｌｌ　ｚｅｒｏ　ｄｅｔｅｃｔｏｒ）７と、加減算器５の出力のうち下位（ｍ－１）ビットがすべて０であるか否かを判別する判別器（ａｌｌ　ｚｅｒｏ　ｄｅｔｅｃｔｏｒ）８と、被除数と除数の符号を比較する第１の比較器９と、加減算器５で行った演算結果の符号が部分剰余の符号と同符号か否かを比較する第２の比較器１０と、加減算器５の演算結果、判別器７、８の判別結果ならびに第２の比較器１０の比

較結果を受けて、商を決定し、セレクタ６の選択を制御する制御回路１１を備えている。

【００３９】

判別器７、８は例えば図２に示すように構成される。図２において判別器７、８は、クロック信号（ＣＬＫ）に同期したプリチャージ型の回路であり、クロック信号を受けるＰチャネルのＦＥＴ（電界効果トランジスタ）２１と、０か否かを判別しようとするデータ、ここでは被除数レジスタ２の出力の下位（ｎ－１）ビット又は加減算器５の出力のうち下位（ｍ－１）ビットを受けて並列接続されたＮチャネルのＦＥＴ２２と、クロック信号を受けるＮチャネルのＦＥＴ２３が高位電源と低位電源との間に接続され、クロック信号がハイレベル時にＦＥＴ２２に与えられたデータのすべてが０の場合はバッファ２４、２５を介して判別結果の１が出力され、ＦＥＴ２２に与えられたデータの内ひとつでも０でない場合にはバッファ２４、２５を介して判別結果の０が出力されて構成される。

【００４０】

次に、このような構成の除算回路を用いて符号付の除算動作を動作サイクルにしたがって説明する。

【００４１】

まず、被除数及び除数の符号によらず被除数ａを被除数レジスタ２に、除数ｂを除数レジスタ１にロードする。また、剰余レジスタ３には被除数ａの符号、すなわちここでは被除数の最上位ビットのデータを全ビットにロードし、商レジスタ４には全ビットに０をロードする（第１サイクル、作用１）。

【００４２】

次に、剰余レジスタ３と被除数レジスタ２をそれぞれ１ビット左（上位）にシフトし、シフト後の両レジスタの連結した内容の上位ｍビットのデータ（連結値）をＰ’とする（第２サイクル、作用２－１）。続いて、被除数ａと除数ｂの符号が同一の場合は、加減算器５によりデータＰ’から除数レジスタ１の出力を減じる。一方、被除数ａと除数ｂの符号が異なる場合には、加減算器５によりデータＰ’と除数レジスタ１の出力を加算する（第２サイクル、作用２－２）。続いて、被除数レジスタ２は出力を左に１ビットシフトして格納する。被除数レジスタ２の最下位ビットには０を格納する（第２サイクル、作用２－３）。

【００４３】

続いて、商レジスタ４は出力を左に１ビットシフトして格納する。商レジスタ４の最下位ビットに格納するデータは次のようにして決定する。ただし、判別器７、８の出力がともに１の場合は、この時点での部分剰余と除数×2^{k-1}　との絶対値が等しいことを意味する。（ｎ－ｋ＋１）は加減算の回数に相当する。また、基本的には被除数ａの符号と部分剰余の符号は同じになるので、ここでは被除数ａの符号を使用して商を決定する。加減算器５の出力の符号と被除数ａの符号が同一の場合は、商レジスタ４の最下位に商として１を書き込む。加減算器５の出力の符号が被除数ａの符号と異なり、かつ判別器７及び判別器８の出力がともに１である場合は、商レジスタ４の最下位に商として１を書き込み、判別器７又は判別器８の出力のいずれか一方が０の場合には、商レジスタ４の最下位に商として０を書き込む（第２サイクル、作用２－４）。続いて、前作用（２－４）で決定した商の値が１ならば減算器５の出力を剰余レジスタ３に格納し、商の値が０ならば剰余レジスタ３には、データＰ’の値を書き込む（第２サイクル、作用２－５）。

【００４４】

次に、前サイクルの作用２－１～２－５をさらに（ｎ－１）回繰り返す（第３サイクル～第（ｎ＋１）サイクル）。

【００４５】

次に、被除数と除数の符号が同一の場合は、商レジスタ４の出力はそのまま商レジスタ４に保存される。一方、被除数と除数の符号が異なる場合には、商レジスタ４の出力の補数をとり商レジスタ４に書き戻される（第（ｎ＋２）サイクル、作用４）。

【００４６】

最後に、上述した動作の結果として、商レジスタ４には商が格納され、剰余レジスタ３に

は剰余が格納され、符号付の除算が終了する（第（ｎ＋３）サイクル、作用５）。

【００４７】

このようにして除算処理が行われ、例えば前述した従来例と同様に被除数ａ＝１００１（２進数）（１０進数で－７を示す）と除数ｂ＝００１１（２進数）（１０進数で３を示す）の除算例を図３に示す。なお、図３において、除数レジスタ１の内容をＢと表記し、剰余レジスタ３の内容をＰと表記し、被除数レジスタ２の内容をＡと表記し、商レジスタ４の内容をＱと表記している。

【００４８】

この実施形態では、被除数レジスタ２と商レジスタ４を別のレジスタに分けたが、図６に示す従来構成のように一つのレジスタを共用するようにすればレジスタの本数を削減することができ、面積の削減が可能になる。

【００４９】

上記実施形態にあっては、符号付きの被除数及び／又は除数をそのまま扱うことにより、図６に示す従来例と比較して、除数、被除数の補数の算出、及び剰余の補正を行う必要がなくなり、除算時間の短縮を実現することができる。

【００５０】

図４はこの発明の他の実施形態に係る除算回路の構成を示す図である。

【００５１】

この実施形態は、前述した従来の非回復型の除算アルゴリズムを基本にして符号付の被除数ならびに除数により除算を行うものであり、前述した非回復型の除算アルゴリズムと大きく異なるところは、商の判定手法である。商が０の場合は、正確な部分剰余の代わりに仮の部分剰余として加減算器の出力を仮の部分剰余として一旦レジスタに格納しておき、除数と被除数が同符号ならば、前のサイクルでは絶対値比較として減算を行なっている。正確な部分剰余を求めるためには、（仮の部分剰余＋除数）を行わなければならない。したがって、次のサイクルで、仮の部分剰余と除数の加算を行なえばよい（（仮剰余＋除数）×２－除数＝仮剰余×２＋除数による）。一方、除数と被除数が異符号ならば、前のサイクルでは絶対値比較として加算を行なっている。正確な部分剰余を求めるためには、（剰余－除数）を行なわなければならない。したがって、次のサイクルで、仮の部分剰余と除数との減算を行なえばよい（（仮剰余－除数）×２＋除数＝仮剰余×２－除数による）。これにより、上記実施形態の除算と同じ結果が得られる。

【００５２】

図４において、この実施形態の除算回路は、２の補数表現された符号付きｎビットの被除数をａ、ｍビットの除数をｂとすると、図１に示す構成に比べて、図１に示すセレクタ６を削除して剰余レジスタ３に加減算器５の出力を入力し、図１に示す制御回路１１に代えて、第１及び第２の比較器９，１０と判別器７、８の判別結果を受けて商を決定するとともに加減算器５に加算又は減算を指示する制御回路３１を設け、他の構成は図１に示すものと同様であり、図１と同符号をもって示す。

【００５３】

次に、このような構成の除算回路を用いて符号付の除算動作を動作サイクルにしたがって説明する。

【００５４】

まず、被除数及び除数の符号によらず被除数ａを被除数レジスタ２に、除数ｂを除数レジスタ２にロードする。剰余レジスタ３には被除数ａの符号、ここでは被除数ａの最上位ビットのデータを全ビットにロードし、商レジスタ４には全ビットに０をロードする（第１サイクル、作用１）。

【００５５】

次に、剰余レジスタ３及び被除数レジスタ２をそれぞれ１ビット左にシフトし、シフト後の両レジスタの連結した内容の上位ｍビットのデータ（連結値）をＰ’とする（第２サイクル、作用２－１）。続いて、被除数ａと除数ｂの符号が同一の場合は、加減算器５によりデータＰ’から除数レジスタ１の出力を減じ、被除数ａと除数ｂの符号が異なる場合に

は、加減算器５によりデータＰ’と除数レジスタ１の出力を加算する（第２サイクル、作用２－２）。続いて、被除数レジスタ２は出力を左に１ビットシフトして格納し、被除数レジスタ２の最下位ビットには０を格納する（第２サイクル、作用２－３）。

【００５６】

続いて、商レジスタ４は出力を左に１ビットシフトして格納する。商レジスタ４の最下位ビットに格納するデータは次のようにして決定する。ただし、判別器７、８の出力がともに１の場合は、この時点での部分剰余と除数×2^{k-1}　との絶対値が等しいことを意味する。（ｎ－ｋ＋１）は加減算の回数に相当する。また、基本的には被除数ａの符号と、部分剰余の符号は同じになるので、ここでは被除数ａの符号を使用して商を決定する。加減算器５の出力の符号と被除数ａの符号が同一の場合は、商レジスタ４の最下位に商として１を書き込む。一方、加減算器５の出力の符号が被除数ａの符号と異なり、かつ判別器７、８の出力がともに１である場合には商レジスタ４の最下位に商として１を書き込み、判別器７、８の出力のいずれか一方でも０の場合には、商レジスタ４の最下位に商として０を書き込む（第２サイクル、作用２－４）。続いて、加減算器５の出力を被除数レジスタ３に格納する（第２サイクル、作用２－５）。

【００５７】

次のサイクルでは、前サイクルの作用２－４で算出した商が１の場合は、剰余レジスタ３、被除数レジスタ２をそれぞれ１ビット左にシフトする。被除数ａと除数ｂの符号が同一の場合には、加減算器５によりシフト後の両レジスタの連結した上位ｍビットのデータＰ’から除数レジスタ１の出力を減じ、被除数ａと除数ｂの符号が異なる場合には加減算器５によりデータＰ’と除数レジスタ１の出力を加算する。一方、前サイクルの作用２－４で算出した商が０の場合には、剰余レジスタ３、被除数レジスタ２をそれぞれ１ビット左にシフトする。被除数ａと除数ｂの符号が同一の場合は、加減算器５によりデータＰ’と除数レジスタ１の出力とを加算し、被除数ａと除数ｂの符号が異なる場合には、加減算器５によりデータＰ’から剰余レジスタ１の出力を減じる（第３サイクル、作用３－１）。続いて、被除数レジスタ２は出力を左に１ビットシフトして格納し、被除数レジスタ２の最下位ビットには０を格納する（第３サイクル、作用３－２）。続いて、商レジスタ４は出力を左に１ビットシフトして格納する。商レジスタ４の最下位ビットに格納するデータを（第２サイクル、作用２－４）で行ったのと同様の方法で決定し商レジスタ４に格納する（第３サイクル、作用３－３）。続いて、加減算器５の出力を剰余レジスタ３に格納する（第３サイクル、作用３－４）。

【００５８】

次に、第３サイクルの作用３－１～３－４をさらに（ｎ－２）回繰り返す（第４サイクル～第（ｎ＋１）サイクル）。

【００５９】

次に、直前のサイクルで算出した商が１の場合は、剰余レジスタ３はそのままの値を保持し、算出した商が０でかつ被除数ａと除数ｂの符号が同一の場合は、加減算器５により剰余レジスタ３の出力と除数レジスタ１の出力とを加算し、算出した商が０でかつ被除数ａと除数ｂの符号が異なる場合には、加減算器５により剰余レジスタ３の出力から除数レジスタ１の出力を減じ、その結果を剰余レジスタ３に格納する（第（ｎ＋２）サイクル、作用５）。

【００６０】

次に、被除数と除数の符号が同一の場合は、被除数レジスタ２の出力はそのまま被除数レジスタ２に保存され、被除数と除数の符号が異なる場合には、被除数レジスタ２の出力の２の補数をとり被除数レジスタ２に書き戻される（第（ｎ＋３）サイクル、作用６）。

【００６１】

最後に、商レジスタ４には商が格納され、剰余レジスタ３には剰余が格納されて符号付の除算が終了する（第（ｎ＋４）サイクル、作用７）。

【００６２】

このようにして除算処理が行われ、例えば前述した従来例と同様に被除数ａ＝１００１（

２進数）（１０進数で－７を示す）と除数ｂ＝００１１（２進数）（１０進数で３を示す）の除算例を図５に示す。なお、図５において、除数レジスタ１の内容をＢと表記し、剰余レジスタ３の内容をＰと表記し、被除数レジスタ２の内容をＡと表記し、商レジスタ４の内容をＱと表記している。

【００６３】

この実施形態では、被除数レジスタ２と商レジスタ４を別のレジスタに分けたが、図８に示す従来構成のように一つのレジスタを共用するようにすればレジスタの本数を削減することができ、面積の削減が可能になる。

【００６４】

上記実施形態にあっては、符号付きの被除数及び／又は除数をそのまま扱うことにより、図８に示す従来例と比較して、除数、被除数の２の補数の算出、及び剰余の補正を行う必要がなくなり、除算時間の短縮を実現することができる。また、図１に示す実施形態に比べて、最後に剰余の補正のための加減算の回数が一回多くなるが、引き戻し分の時間を削減できるのでサイクル当たりの演算にかかる時間を短縮することができる。

【００６５】

また、符号なし除算を行う場合は被除数ならびに除数の上位側に０を拡張し、符号付の除算を行う場合には被除数ならびに除数にそれぞれの符号ビットを符号拡張するようにすれば、少ないハードウェア量の増加で符号付除算及び符号なし除算を同一の除算回路で行うことができる。

【００６６】

【発明の効果】

以上説明したように、この発明によれば、符号が付いた状態での回復型又は非回復型の除算過程において、部分剰余の一部と除数との絶対値の比較を行ない、部分剰余の一部が大きい場合又は双方が等しい場合はその桁の商を１とし、小さい場合には商を０とするようにしたので、除数、被除数及び剰余の補数をとる必要がなくなり、除算時間を短縮することが可能となる。

【図面の簡単な説明】

【図１】この発明の一実施形態に係る除算回路の構成を示す図である。

【図２】図１に示す判別器の構成を示す図である。

【図３】図１に示す構成における具体的な除算例を示す図である。

【図４】この発明の他の実施形態に係る除算回路の構成を示す図である。

【図５】図４に示す構成における具体的な除算例を示す図である。

【図６】従来の除算回路の一構成例を示す図である。

【図７】図６に示す構成における具体的な除算例を示す図である。

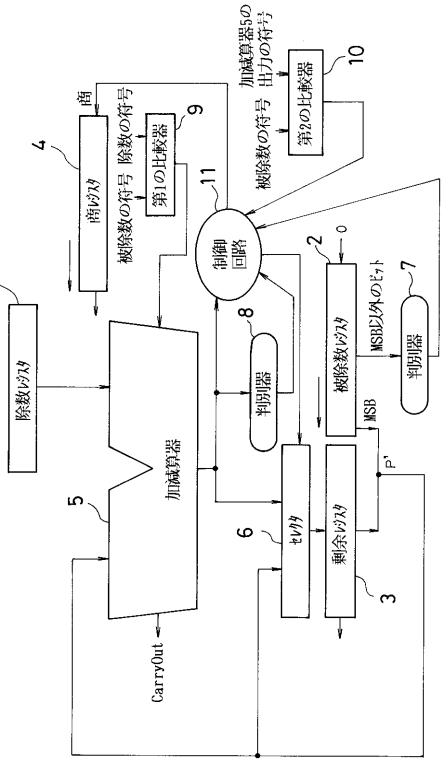
【図８】従来の除算回路の他の構成例を示す図である。

【図９】図８に示す構成における具体的な除算例を示す図である。

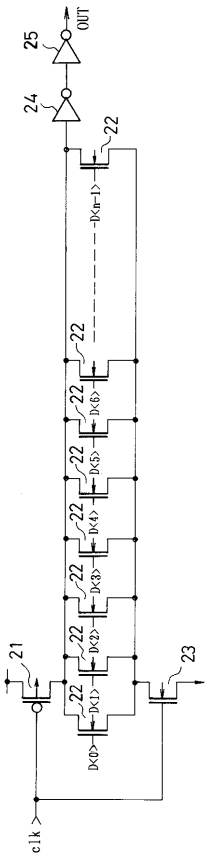
【符号の説明】

１　除数レジスタ
２　被除数レジスタ
３　剰余レジスタ
４　商レジスタ
５　加減算器
６　セレクタ
７，８　判別器
９，１０比較器
１１，３１　制御回路
２１，２２，２３　ＦＥＴ
２４，２５　バッファ

【図１】



【図２】

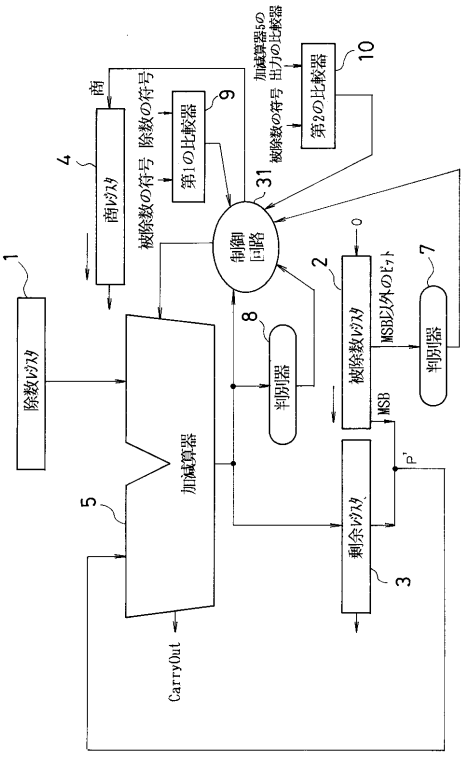


【図３】

ｻｲｸﾙ	作用	動作	ﾚｼﾞｽﾀ10出力	ﾚｼﾞｽﾀ9出力	ﾚｼﾞｽﾀ7出力	ﾚｼﾞｽﾀ20出力	ﾚｼﾞｽﾀ40出力
1	1	ﾃﾞｰﾀをﾛｰﾄﾞ					
2	2-1	P' =1111	0011	1111	1001	0000	
	2-2	被除数と除数の符号が異なるから，P' +B=0010	0011	1111	1001	0000	
	2-3	Aを左へ1ﾋｯﾄｼﾌﾄ	0011	1111	1001	0000	
	2-4	判別器7，8ともに0_c(P' +B)は正で，被除数の符号と異なるので，商は0	0011	1111	1001	0000	
	2-5	商は0なので，ﾚｼﾞｽﾀ3にP' の値を書き込む	0011	1111	1001	0000	
3	2-1	P' =1110	0011	1111	0010	0000	
	2-2	被除数と除数の符号が異なるから，P' +B=0010	0011	1111	0010	0000	
	2-3	Aを左へ1ﾋｯﾄｼﾌﾄ	0011	1111	0010	0000	
	2-4	判別器7，8ともに0_c(P' +B)は正で，被除数の符号と異なるので，商は0	0011	1111	0010	0000	
	2-5	商は0なので，ﾚｼﾞｽﾀ3にP' の値を書き込む	0011	1111	0010	0000	
4	2-1	P' =1100	0011	1110	0100	0000	
	2-2	被除数と除数の符号が異なるから，P' +B=1111	0011	1110	0100	0000	
	2-3	Aを左へ1ﾋｯﾄｼﾌﾄ	0011	1110	0100	0000	
	2-4	判別器7，8ともに0_c(P' +B)は負で，被除数の符号と同じなので，商は1	0011	1110	0100	0000	
	2-5	商は1なので，ﾚｼﾞｽﾀ3に(P' +B)の値を書き込む	0011	1110	0100	0000	
5	2-1	P' =1111	0011	1111	1000	0001	
	2-2	被除数と除数の符号が異なるから，P' +B=0010	0011	1111	1000	0000	
	2-3	Aを左へ1ﾋｯﾄｼﾌﾄ	0011	1111	1000	0000	
	2-4	判別器7，8-0_c(P' +B)は正で，被除数の符号と異なるので，商は0	0011	1111	1000	0000	
	2-5	商は0なので，ﾚｼﾞｽﾀ3にP' の値を書き込む	0011	1111	1000	0000	
6	4	被除数と除数の符号が異なるので，Qの2の補数をとる	0011	1111	0000	0010	
7	5	除算終了	0011	1111	0000	1110	

a = 1001, b = 0011, a/bを計算する場合

【図４】

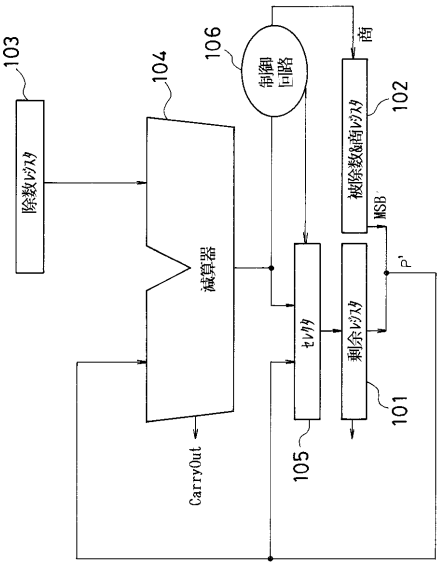


【図５】

ｻｲｸﾙ	作用	動作	ﾚｼﾞｽﾀ10出力	ﾚｼﾞｽﾀ30出力	ﾚｼﾞｽﾀ20出力	ﾚｼﾞｽﾀ40出力
1	1	ﾃﾞｰﾀをﾛｰﾄﾞ				
2	2-1	P'=1111	0011	1111	1001	0000
	2-2	被除数と除数の符号が異なるから，P'+B=0010	0011	1111	1001	0000
	2-3	Aを左へ1ビットシフト	0011	1111	1001	0000
	2-4	判別器7，8ともに0_2(P'+B)は正で，被除数の符号と異なるので，商は0	0011	1111	1001	0000
	2-5	ﾚｼﾞｽﾀ3に(P'+B)の値を書き込む	0011	1111	1001	0000
3	3-1	P'=0100 被除数と除数の符号が異なり，前ｻｲｸﾙの商は0だから，P'−B=0001	0011	0010	0010	0000
	3-2	Aを左へ1ビットシフト	0011	1111	0010	0000
	3-3	判別器7，8ともに0_2(P'−B)は正で，被除数の符号と異なるので，商は0	0011	1111	0010	0000
	3-4	ﾚｼﾞｽﾀ3に(P'−B)の値を書き込む	0011	1111	0010	0000
4	3-1	P'=0010 被除数と除数の符号が異なり，前ｻｲｸﾙの商は0だから，P'−B=1111	0011	0001	0100	0000
	3-2	Aを左へ1ビットシフト	0011	0001	0100	0000
	3-3	判別器7，8ともに0_2(P'−B)は負で，被除数の符号と同じなので，商は1	0011	0001	0100	0000
	3-4	ﾚｼﾞｽﾀ3に(P'−B)の値を書き込む	0011	0001	0100	0000
5	3-1	P'=1111 被除数と除数の符号が異なり，前ｻｲｸﾙの商が1だからP'+B=0010	0011	1111	1000	0001
	3-2	Aを左へ1ビットシフト	0011	1111	1000	0001
	3-3	判別器7，8-0_2(P'+B)は正で，被除数の符号と異なるので，商は0	0011	1111	1000	0001
	3-4	ﾚｼﾞｽﾀ3に(P'+B)の値を書き込む	0011	1111	1000	0001
6	5	前ｻｲｸﾙの商から0，被除数と除数の符号が異なるから(P'−B)をﾚｼﾞｽﾀ3に書き込む	0011	0010	0000	0010
7	6	被除数と除数の符号が異なるので，Qの2の補数をとる	0011	1111	0000	0010
8	7	除算終了	0011	1111	0000	1110

a = 1001, b = 0011, a/bを計算する場合

【図６】

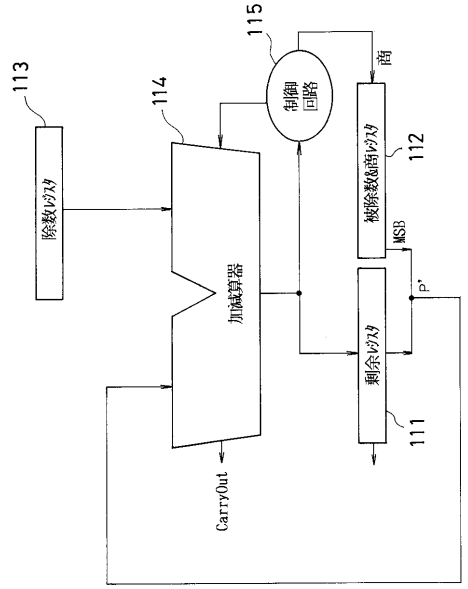


【図７】

ｻｲｸﾙ	作用	動作	ﾚｼﾞｽﾀ30出力	ﾚｼﾞｽﾀ10出力	ﾚｼﾞｽﾀ20出力
1, 2	1	a'=0111, b'=0011に変換			
3	2	ﾃﾞｰﾀをﾛｰﾄﾞ			
4	3-1	P'=0000	0011	0000	0111
	3-2	P'−B=1101	0011	0000	0111
	3-3	減算器の出力は負だから，商は0	0011	0000	0111
	3-4	ﾚｼﾞｽﾀ01にP'を書き込む	0011	0000	0111
5	3-1	P'=0001	0011	0000	1110
	3-2	P'−B=1110	0011	0000	1110
	3-3	減算器の出力は負だから，商は0	0011	0000	1110
	3-4	ﾚｼﾞｽﾀ01にP'を書き込む	0011	0000	1110
6	3-1	P'=0011	0011	0001	1100
	3-2	P'−B=0000	0011	0001	1100
	3-3	減算器の出力は負でない，商は1	0011	0001	1100
	3-4	ﾚｼﾞｽﾀ01に(P'−B)の値を書き込む	0011	0001	1100
7	3-1	P'=0001	0011	0000	1001
	3-2	P'−B=1110	0011	0000	1001
	3-3	減算器の出力は負だから，商は0	0011	0000	1001
	3-4	ﾚｼﾞｽﾀ01にP'を書き込む	0011	0000	1001
8	5	被除数と除数の符号が異なるので，Aの2の補数をとる。(1110)	0011	0001	0010
9	6	被除数は負なので，Pの2の補数をとる。(1110)	0011	0001	1110
10	7	除算終了	0011	1111	1110

a=0111, b=0011, a/bを計算する場合

【図８】



【図９】

サイクル	作用	動作	レジスタ1030出力	レジスタ1010出力	レジスタ1020出力
1, 2	1	a'=0111, b'=0011に変換			
3	2	データをロード			
4	3-1	P'=0000	0011	0000	0111
	3-2	P'=B=1101	0011	0000	0111
	3-3	加減算器の出力は負だから，商は0	0011	0000	0111
	3-4	レジスタ101に(P'-B)を書き込む	0011	0000	0111
5	4-1	P'=1011，レジスタ101は負だから，P'+B=1110	0011	1101	1110
	4-2	加減算器の出力は負だから，商は0	0011	1101	1110
	4-3	レジスタ101に(P'+B)の値を書き込む	0011	1101	1110
6	4-1	P'=1101，レジスタ101は負だから，P'+B=0000	0011	1110	1100
	4-2	加減算器の出力は負でないから，商は1	0011	1100	1100
	4-3	レジスタ101に(P'+B)の値を書き込む	0011	1100	1100
7	4-1	P'=0001，レジスタ101は負でないから，P'-B=1110	0011	0000	1001
	4-2	加減算器の出力は負だから，商は0	0011	0000	1001
	4-3	レジスタ101に(P'-B)を書き込む	0011	0000	1001
8	6	レジスタ101の出力は負なので，レジスタ101に(P'+B)の値を書き込む(0001)	0011	1110	0010
9	7	被除数と除数の符号が異なるので，Aの2の補数をとる。(1110)	0011	0001	0010
10	8	被除数は負なので、レジスタ101の2の補数をとる。(1111)	0011	0001	1110
11	9	除算終了	0011	1111	1110

a=0111, b=0011, a/bを計算する場合

フロントページの続き

審査官　田中　友章

(56)参考文献　特開昭６２－０３２５３６（ＪＰ，Ａ）
特開昭６１－０２６１３７（ＪＰ，Ａ）
特開平０８－２６３２７１（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.[7]，ＤＢ名)
G06F 7/52 320
G06F 7/38