数理解析研究所講究録
第687巻 1989年 174-194


# 有限体上の高速逆元演算回路の構成

埼玉大学工学部電子工学科　荒木純道 (Kiyomichi Araki)
埼玉大学工学部電子工学科　藤田育雄 (Ikuo Fujita)
埼玉大学工学部電子工学科　森末道忠 (Mititada Morisue)

## 1. はじめに

いくつかの重要な情報処理を有限体上での演算過程として取り扱おうとする試みがなされている。例えば、誤り訂正符号・復号、暗号理論、スイッチング理論、デジタル信号処理などがそれである[1][2][3]。また、これらはごく自然に多値化が行なえるという利点がある。

これらの演算過程を具体的にハードウェア化し演算回路を構成するに際して様々な試みがなされているが、特に高速化、高モジュール化に大きな精力がそそがれている。

有限体上での演算過程で最も困難なものは逆元もしくは除算である。本稿では（一般化された）ユークリッド互除法の高速性を生かし、しかもモジュラリティの高い逆元演算回路

の構成法を提案することにする。

## 2. 有限体での逆元

有限体 GF($p^n$) では通常の数と同じく4則演算が自由に行えるが、具体的に除算 a／b を行なうためには、（乗法に関する）逆元を求める必要があり、$a \cdot b^{-1}$ で a／b を実行している。この逆元を求める演算過程（もしくは演算回路）は加算や乗算に比べはるかに複雑で、またモジュール化しにくいとされている。

この逆元を高速に求める方法として、次の2つがよく知られている。

### （I）フェルマの小定理による方法

GF（p）上のn次原始既約多項式を f（x）とすると、GF($p^n$) の非零の任意の元 a に対して常に

$$a^{p-1} = 1 \pmod{f(x)} \qquad (1)$$

が成立する。これをフェルマの小定理という。

両辺に $a^{-1}$ を掛ければ

$$a^{p-2} = a^{-1} \pmod{f(x)} \qquad (2)$$

を得る。

これにより乗算の反復的適用だけで逆元が求まることがわかる[4]。

（Ⅱ）ユークリッドの互除法による方法[5]

フェルマの小定理より高速な演算過程として、次の方法がある。

GF（p）上のn次原始既約多項式をf（x）としaのベクトル表現（$a_0$, $a_1$, ・・・, $a_{n-1}$）を多項式に対応させa（x）とする。

$$a(x) = a_0x^0 + a_1x^1 + \cdots + a_{n-1}x^{n-1} \quad (3)$$

このとき、ユークリッド互除法により

$$a(x)b(x) + f(x)c(x) = 1$$

$$deg[b(x)] < deg[f(x)] \quad (4)$$

を満たすb（x）を求める。（当然のことながら、f（x）が既約であるので（4）の解b（x）、c（x）は必ず存在し一意的である。）

本稿では、（Ⅱ）ユークリッド互除法による方法により高速逆元演算回路を構成することにする。

## 3. 高速逆元演算回路の構成

### 3－1 基本アルゴリズム

さて、

$$ab = 1 \pmod{f(x)} \qquad (5)$$

を満足するbがGF($p^n$)におけるaの逆元$a^{-1}$であるが、このbを求めることは、不定方程式

$$ab + fc = 1 \qquad (6)$$

を解くことに他ならない。よって、一般化されたユークリッドの互除法を使ってbを求めることができる。（なお、必要に応じて、a(x)をaと略記している。）

また、bを高速に求めるために（図1）のアルゴリズムを使用する。（pascal風に略記。）

変数s, tによってGCD(a, f)を求める計算をしながら、u, vを求めているが、$av = t \pmod{f}$が常に成り立っているので、t＝1に達した時（この時s＝0）、$av = 1 \pmod{f}$となりb＝vが求められるわけである。

例として、GF($2^4$)（原始既約多項式$f(x) = x^4 + x + 1$とする）における$a = x^3 + x + 1$の逆元を求めると

（図２）のようにそれぞれ変化し、最終的にｖの値として、

$$b = a^{-1} = (x^3 + x + 1)^{-1} = (x^2 + 1) \qquad (7)$$

を得る。

```
begin
    s:=a;   t:=f;   u:=1;   v:=0;
    while s>0 do
        begin
            q:=t div s;
            w:=t-q*s;    t:=s;    s:=w;
            w:=v-q*u;    v:=u;    u:=w;
        end;
    inv:=v;
end
```

（図１）基本アルゴリズム

| s | t | u | v |
|---|---|---|---|
| $x^3 + x + 1$ | $x^4 + x + 1$ | 1 | 0 |
| $x^2 + 1$ | $x^3 + x + 1$ | x | 1 |
| 1 | $x^2 + 1$ | $x^2 + 1$ | x |
| 0 | 1 | $x^4 + x + 1$ | $x^2 + 1$ |

（図２）各変数の変化の様子

このアルゴリズムを回路化するにあたって、多項式の除算回路や多項式の乗算回路が必要となるので、まず、それらから説明する。（蛇足ながら、多項式の除算と有限体上の除算とは別の演算過程であることを注意しておきたい。）

## ３－２　多項式の除算回路

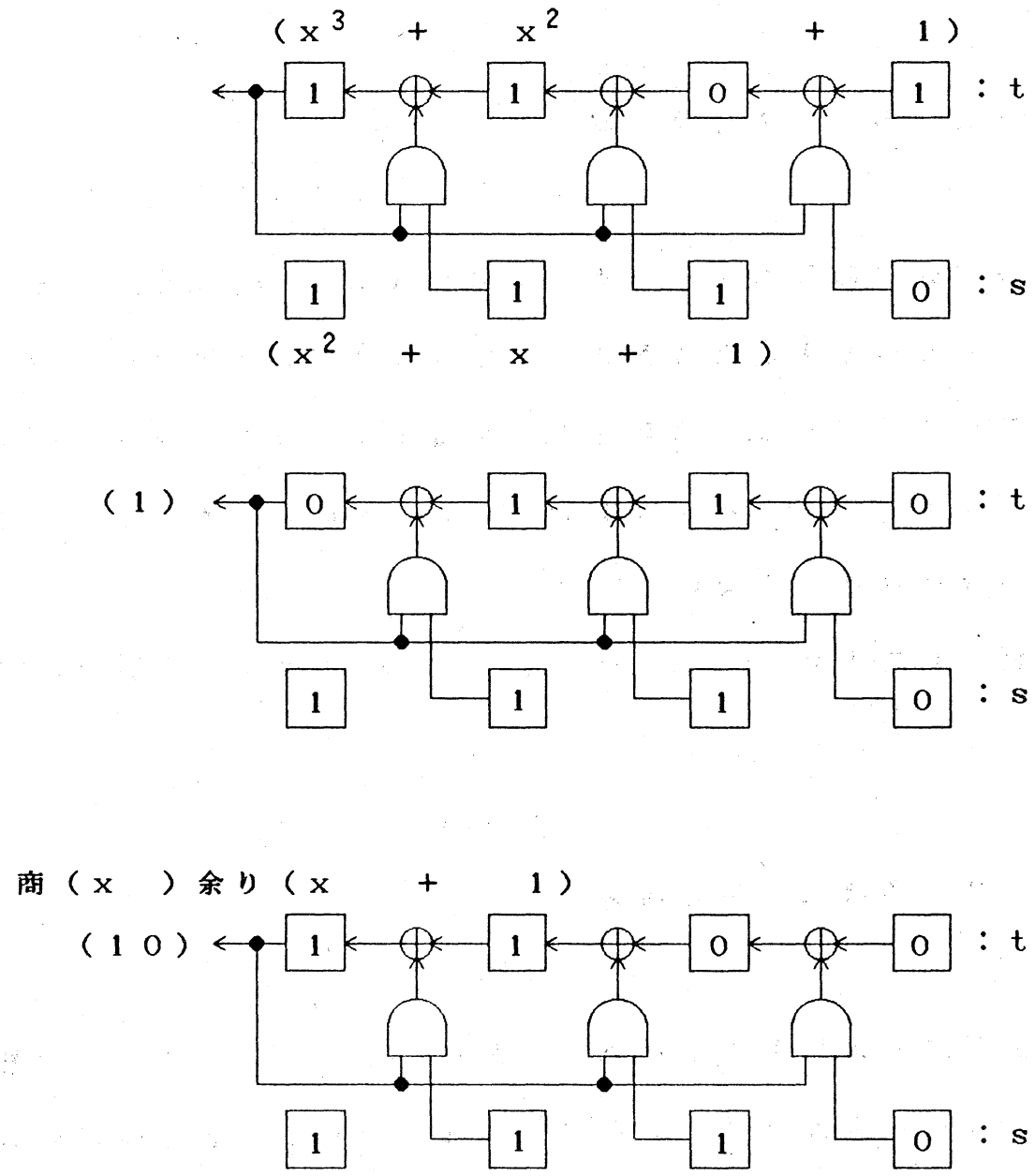
多項式の除算をするための回路を（図３）に示す。商多項式ｔ／ｓの結果が出力され、ｔに余り多項式が残るようになっている。ただし、除算を実行するまえにｓレジスタの最高次ビット$s_n = 1$となるまでｓレジスタの内容をシフトしておかなければならない。

例としてｔ＝$x^3 + x^2 + 1$，ｓ＝$x^2 + X + 1$を取り上げる。

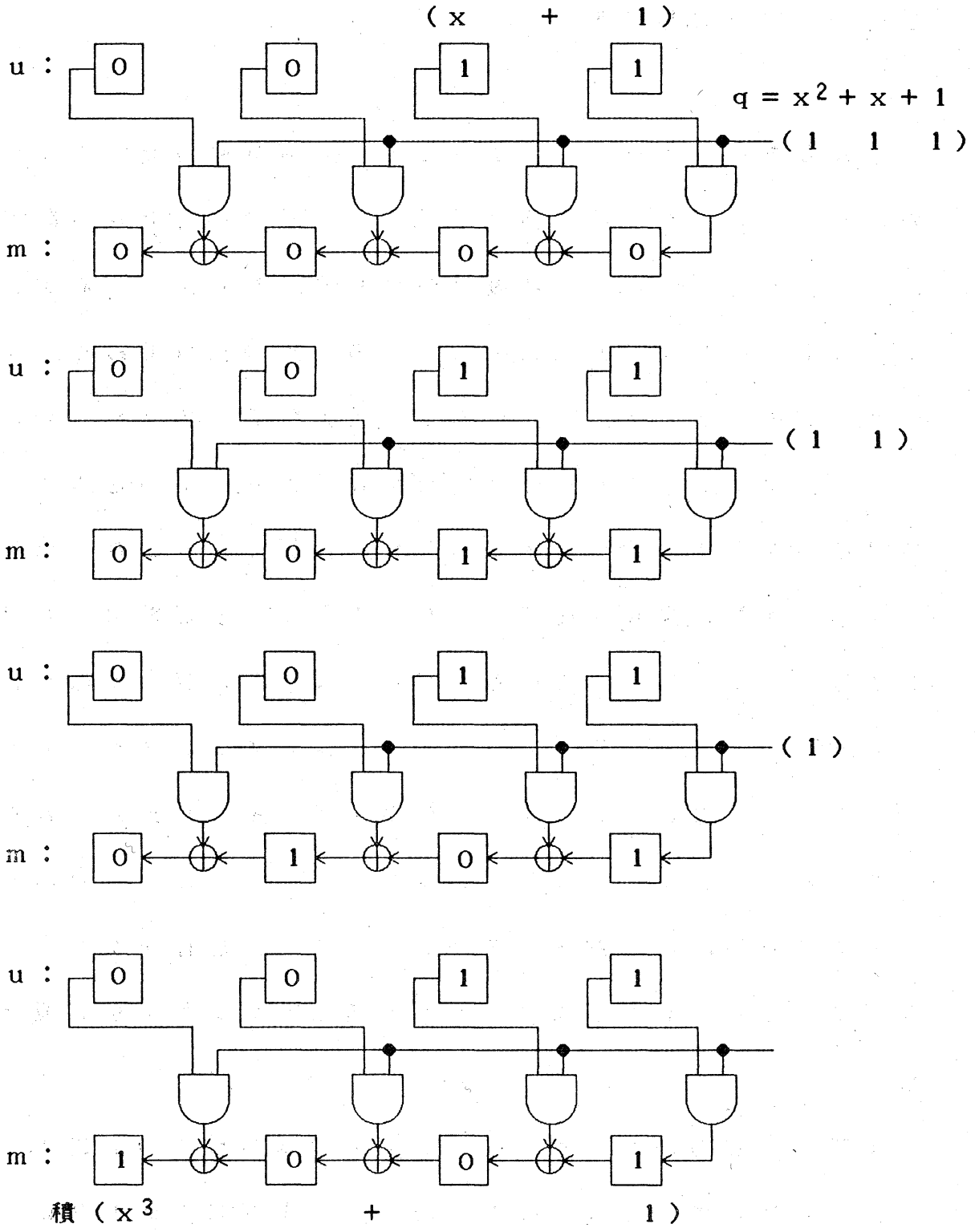
## ３－３　多項式の乗算回路

多項式の乗算をするための回路を（図４）に示す。入力端子からｑの係数を高次から順に入れていくとｕとｑの積の多項式がレジスタｍに得られるような回路となっている。

例としてｕ＝ｘ＋１、ｑ＝$x^2 + x + 1$とする。




( 図 3 ) 除 算 回 路 の 例


( x + 1 )
u :
0 0 1 1
q = x 2 + x + 1
( 1 1 1 )
m :
0 0 0 0
u :
0 0 1 1
( 1 1 )
m :
0 0 1 1
u :
0 0 1 1
( 1 )
m :
0 1 0 1
u :
0 0 1 1
m :
1 0 0 1
積 ( x 3 + 1 )

( 図 4 ) 乗 算 回 路 の 例

## ３－４　逆元演算回路

これらを組合せることによって、逆元演算回路を（図５）のように構成した。図の例はＧＦ（$2^3$）の逆元演算回路である。フェイズ１は、初期設定時の配線であり、$a_i$および$f_i$はそれぞれａ（ｘ）とｆ（ｘ）のｉ次の係数である。同時に、このフェイズで前回の結果を出力する。フェイズ２は、ｓレジスタのシフトを行なう時の配線であり、その他のレジスタは値を保持する。フェイズ３が、除算をするときの配線であり、除算と乗算を同時に行なうことにより無駄な待ち時間をはぶくようになっている。なお、ｓｐおよびｔｐはそれぞれｓとｔの「１の位」を示すポインタ用のレジスタであり、「１の位」に対応するビットだけが１で、それ以外は０となっている。フェイズ４は、ｓレジスタとｔレジスタの内容を交換する時の配線である。各フェイズごとに配線がかなり違うので、これらを電子式に切り替えるような制御部が必要となる。

実際に、制御部も含めた１ビット分の基本セルの回路を、（図６）に示す。このセルを（図７）のように４個直列につなぐことにより ＧＦ（$2^3$）の逆元演算回路を構成できる。また、このセルをｎ＋１個直列につなぐことにより、任意の

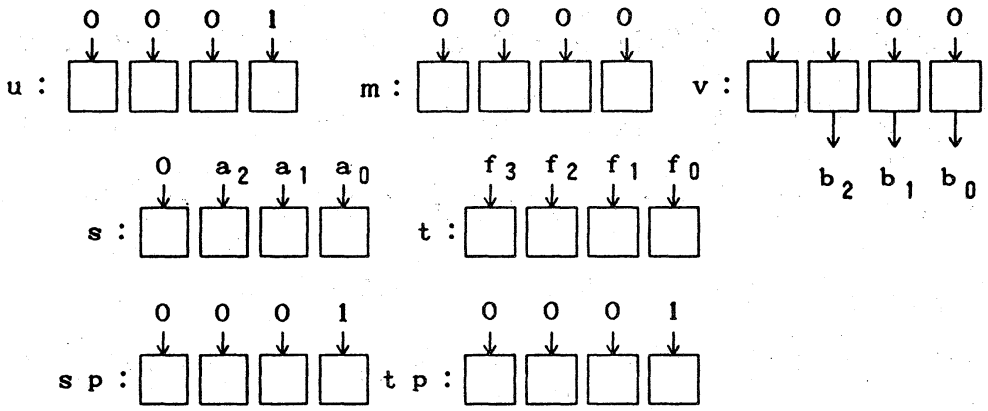
nに対する逆元演算回路を容易に構成することができる。また、レジスタ、基本演算部を、2値からp値に変更すれば、GF($p^n$)における逆元演算回路が直ちに構成できる。

CTL-AからCTL-Dは回路を制御するための制御信号であり、これらの信号により回路の内部結線が制御されている。（図6）の中のA、BそしてCが、それぞれCTL-A、CTL-B、CTL-Cによって制御されるスイッチであり、制御信号が1の時には右からの信号を、制御信号が0の時には上または下からの信号を通すようになっている。CTL-Aはsレジスタのすべてのビットの OR をとったものであり、逆元計算の終了時に0となり、それ以外の時には1となる。CTL-Bはsレジスタの最高次ビット$s_n$であり、除算を実行する前に、これが1となるまでsレジスタがシフトされる。また、CTL-Cは式（8）で表わされるものであり、除算が終了した時に0となる。

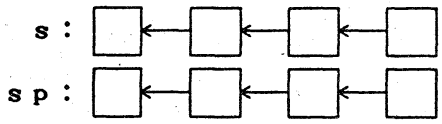
$$CTL\text{-}C = (t_{p_n} \text{ XOR } s_{p_{n-1}}) \text{OR} (t_{p_{n-1}} \text{ XOR } s_{p_{n-2}}) \text{ OR } \cdots \text{ OR} (t_{p_1} \text{ XOR } s_{p_0}) \text{OR} (t_{p_0}) \quad (8)$$

なお、CTL-Dはtレジスタの最高次のビット$t_n$である。

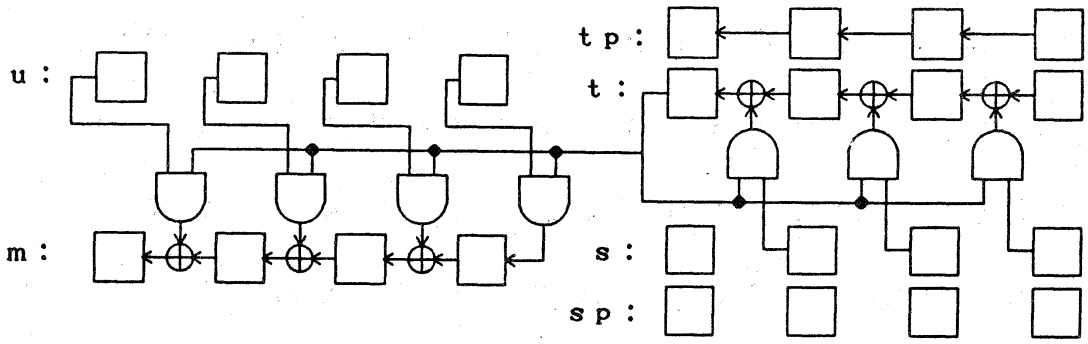
各制御信号と各フェイズの対応を（表1）に示し、各フェイズにおけるデータの流れを（図8-a）から（図8-d）に点線で示す。



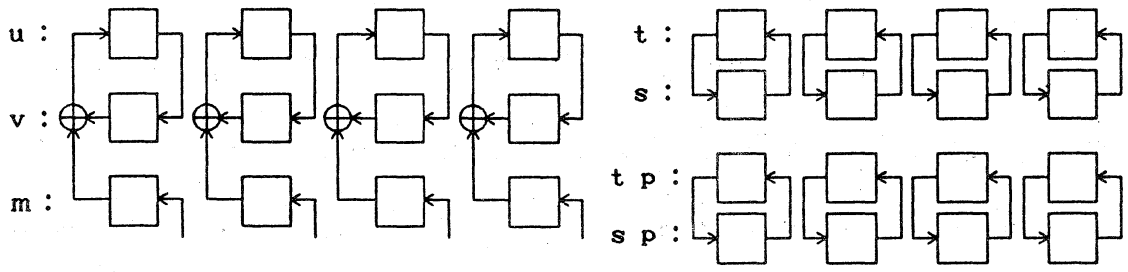

（フェイズ１）　初期設定と結果出力




（フェイズ２）　ｓレジスタのシフト

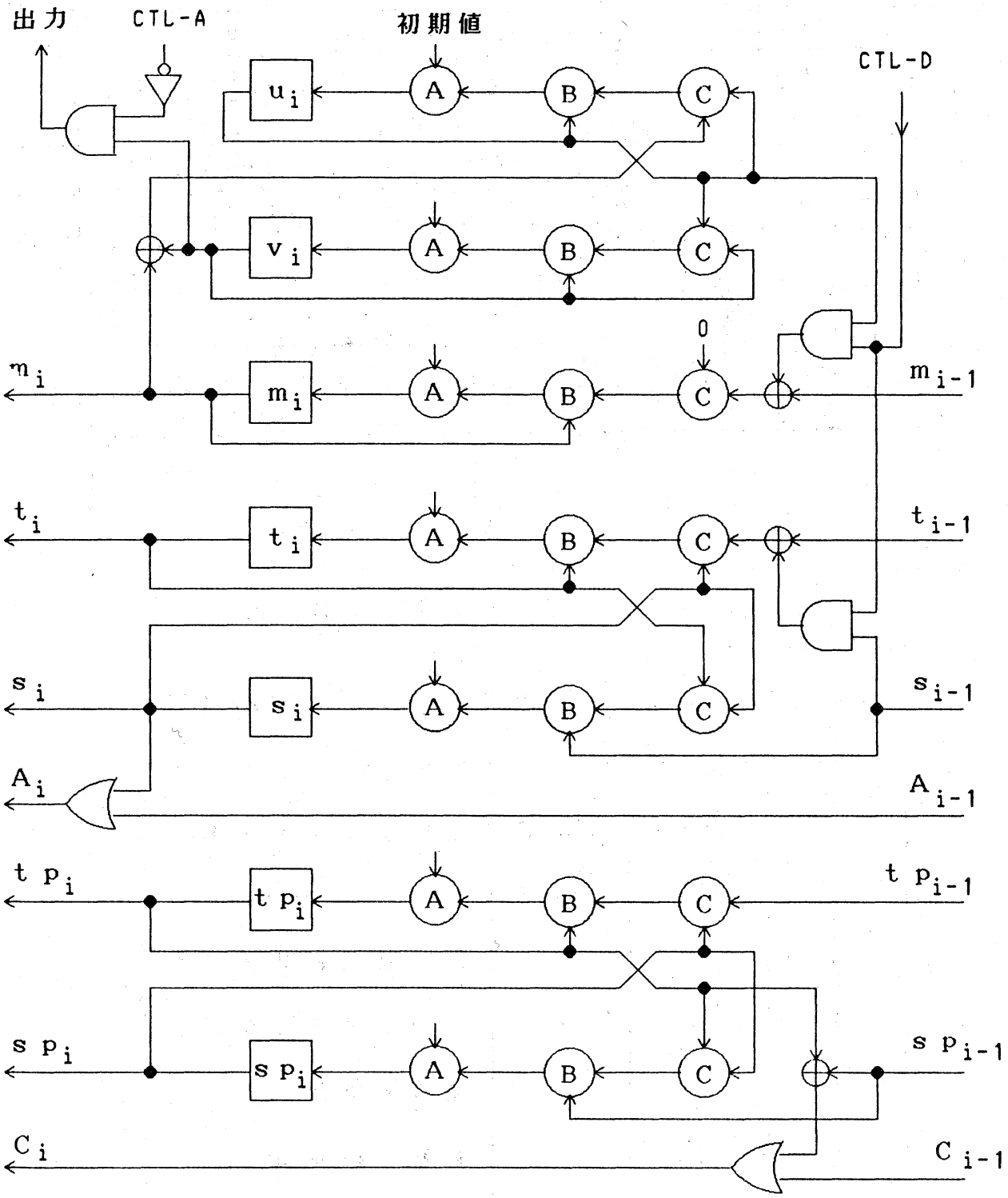



（フェイズ３）　除算の実行

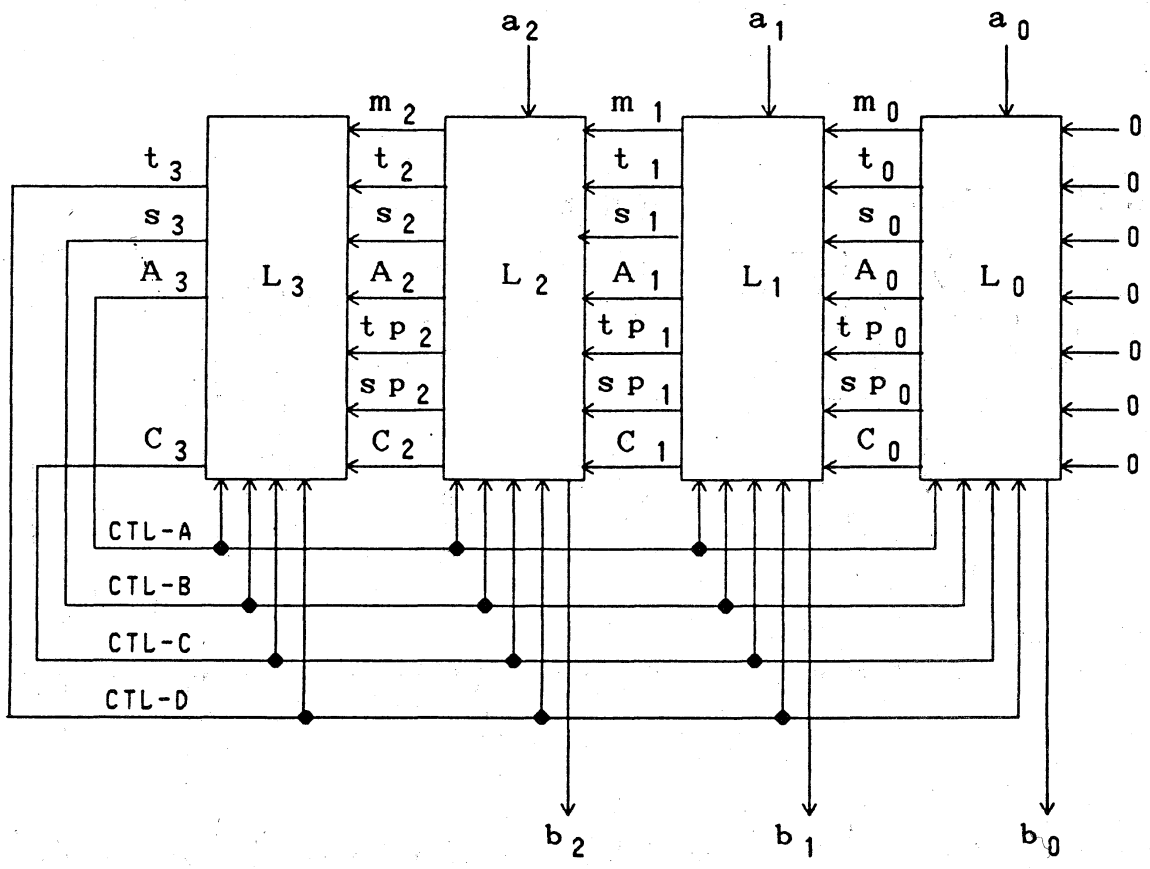



（フェイズ４）　ｓとｔの交換

（図５）各フェイズごとの配線図



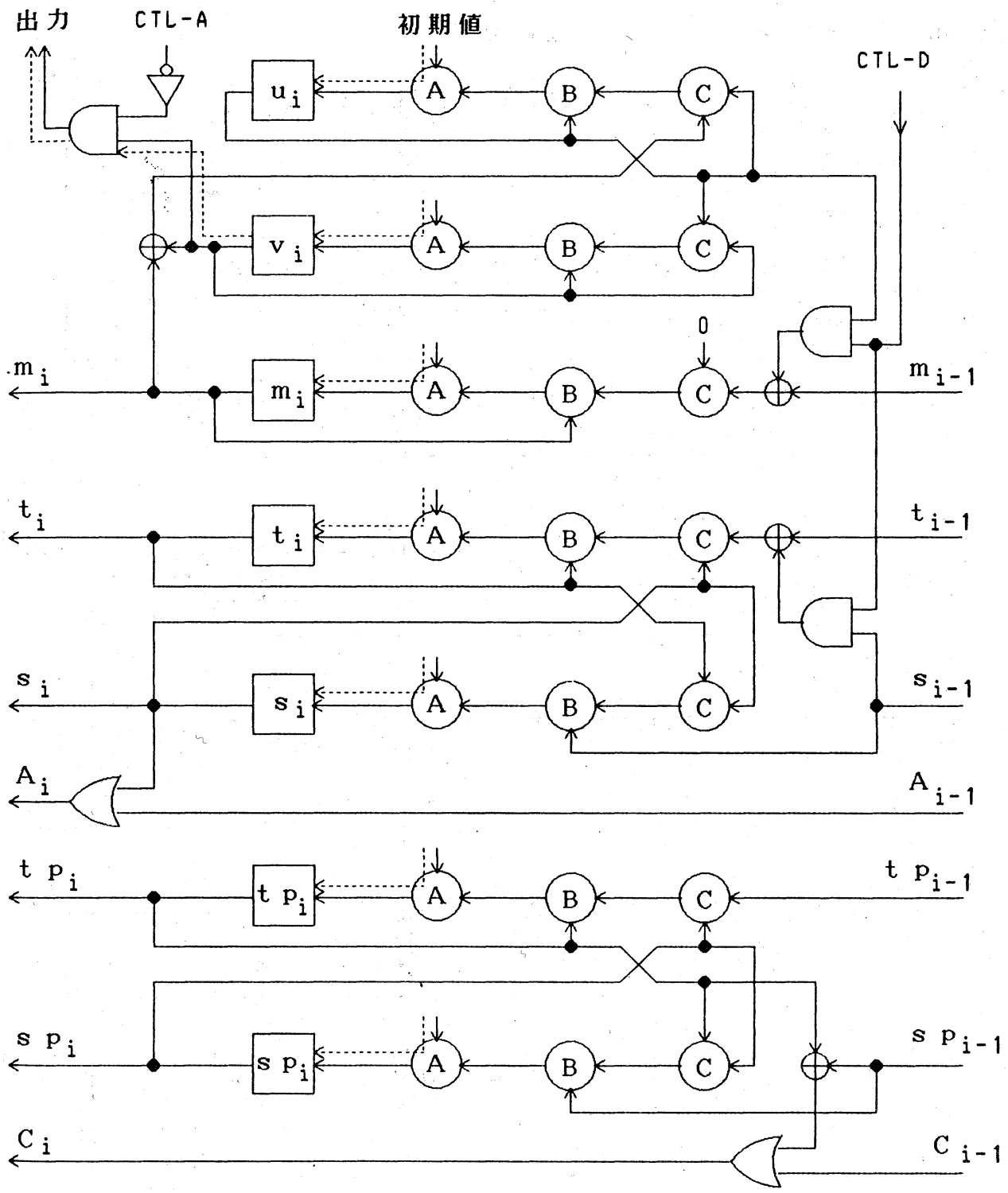

（図６）　１ビット分の基本セルＬｉの回路図



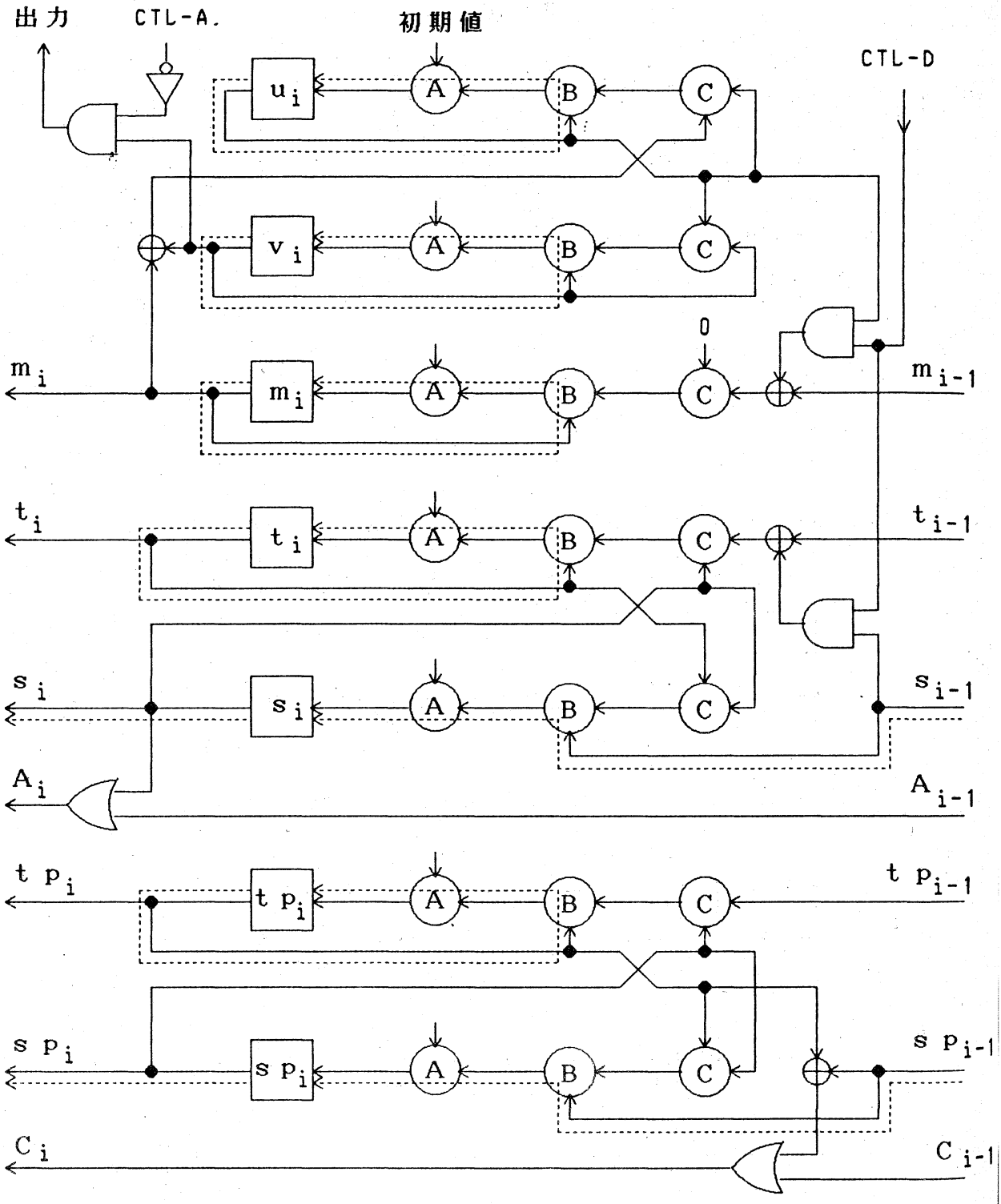

（図７）　ＧＦ（２$^{3}$）の逆元演算回路

（表１）　各制御信号と各フェイズの対応表

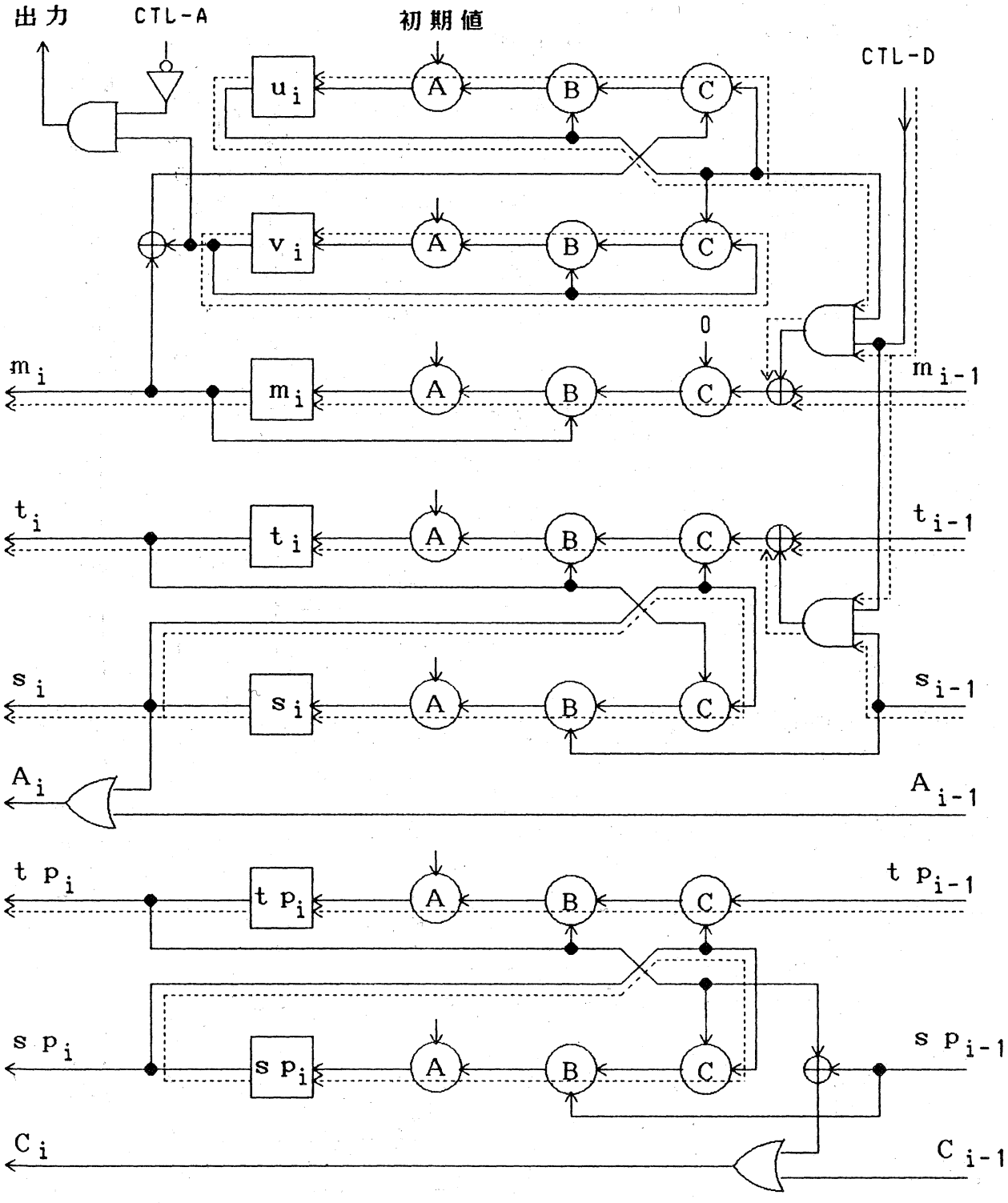
| CTL-A | CTL-B | CTL-C | 対応するフェイズ |
|---|---|---|---|
| 0 | − | − | 初期設定と結果出力 |
| 1 | 0 | − | ｓレジスタのシフト |
| 1 | 1 | 1 | 除算の実行 |
| 1 | 1 | 0 | ｓとｔの交換 |



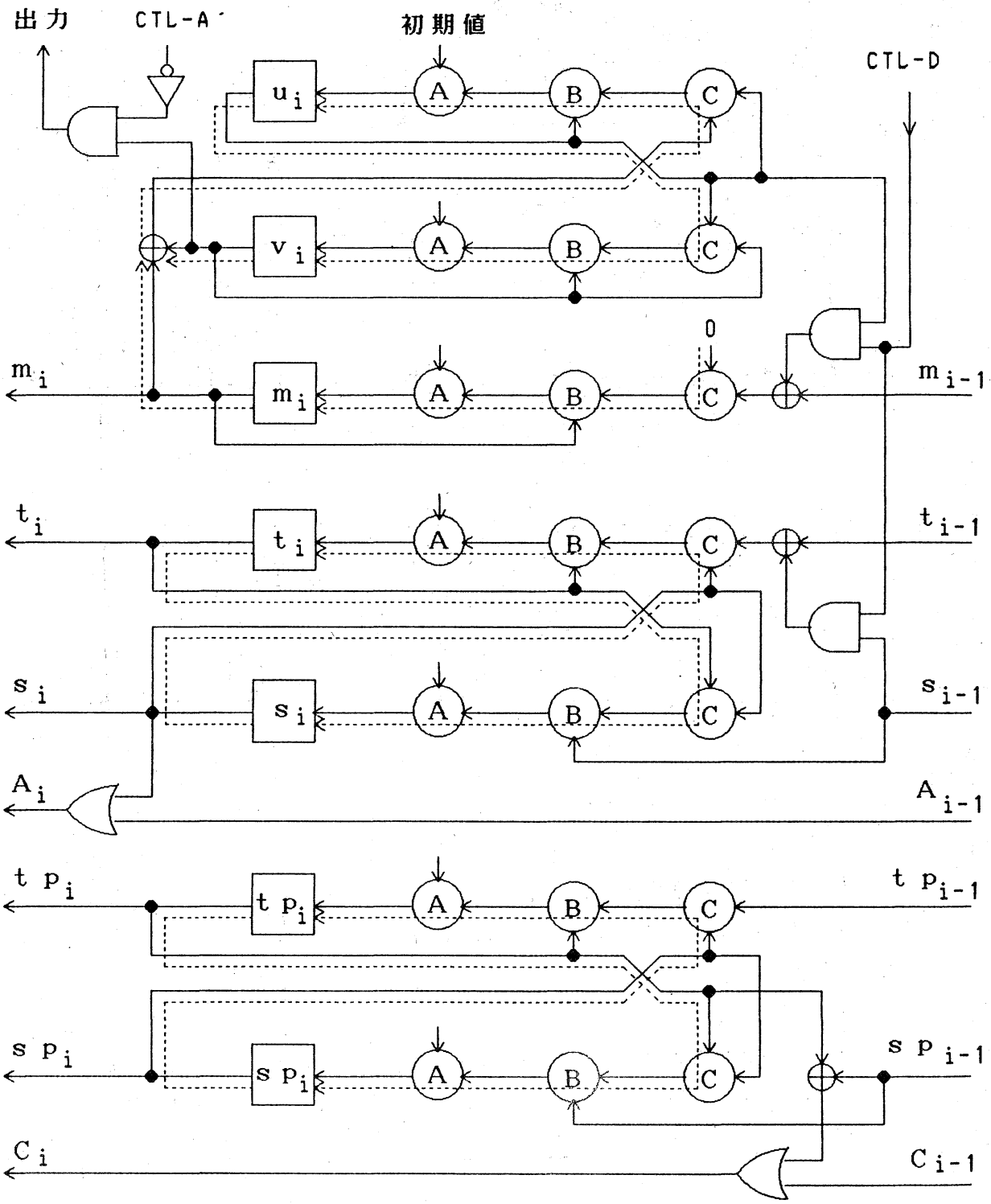

（図８－ａ）フェイズ１のデータの流れ




（図８－ｂ）フェイズ２のデータの流れ




（図８－ｃ）フェイズ３のデータの流れ




（図８－ｄ）フェイズ４のデータの流れ

## 4. 考察

### 4－1　GF（$2^n$）における計算回数とクロック回数

まず、除算を1回やるごとに、s、tは最低でも1ずつ次数が下がるので、除算の回数は最高n回ですむ。

また、1回の除算に何クロックかかるかはわからないが、1クロックごとに必ずsかtのどちらかの次数が1下がるので、s＝0，t＝1となるまで最高でも2n＋1クロックしかかからないことがわかる。これにsとtの交換に最低1クロック、最高でnクロックかかり、初期設定に1クロックかかるのでクロック回数は

$$2n+3 \leqq \text{クロック回数} \leqq 3n+2 \qquad (9)$$

となる。

### 4－2　モジュール化について

この構成法では、1ビット分の基本セル回路をモジュール化することにより、それをn＋1個直列につなぐだけで任意のnに対する逆元演算回路を簡単に構成できる。

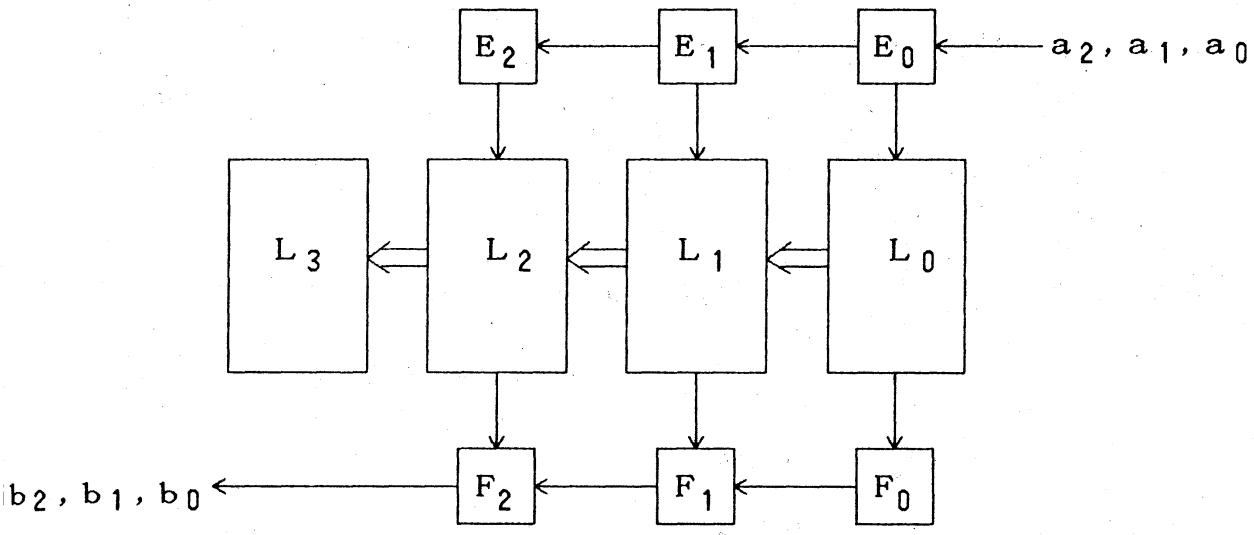
### 4－3　汎用性について

この逆元演算回路は、ＧＦ（$2^n$）での逆元演算回路があれば、$k \leqq n$なる任意のｋに対するＧＦ（$2^k$）の元の逆元を同一の回路を使って求めることが可能である。また、原始既約多項式が何であっても、同じ回路で逆元を計算することができる。

## 5．まとめ

ユークリッドの互除法のアルゴリズムに基づいて、有限体ＧＦ（$2^n$）の元の逆元を求める回路を構成した。この方法により設計すると、１ビット分の基本セルの繰り返しにより逆元演算回路を構成できるので、モジュール性において優れている。また、有限体の大きさや原始既約多項式が変わっても、同じ基本セルを使用して逆元演算回路を構成できることから汎用性が高い。これらにより、回路のＶＬＳＩ化が容易な回路構成法であると言えるであろう。

なお、本稿ではデータの入出力をパラレル型に設計したが、（図９）のように入出力バッファを付け足すことにより、シーケンシャル型のデータ入出力が可能である。




（図９）シーケンシャル型の逆元演算回路

参考文献

[1] C.S.Yeh, et.al " Systolic Multipliers for Finite Fields GF($2^m$)" IEEE Trans.Com. Vol.c-33 No.4 pp.357-360,1984.

[2] 荒木ら "有限体上の論理関数におけるフロベニウスサイクルの性質について",電情通,論文誌（D） Vol.71D No.2 pp.454-455,1988.

[3] K.Araki, et.al " Programmable logical functions over finite field using trace function ". submitted to Electronics Letters.

[4] C.C.Wang, et.al. "VLSI Architectures for Computing Multiplications and Inverters in GF($2^m$)", IEEE Trans.Com. Vol.34 No.8 pp.709-716,1985.

[5] 小山ら,"現代暗号理論",（1章）電子情報通信学会, 1986.