

I.F.I.P. 論文紹介\*

73. 多項式の根の計算法について

J.L. Howland: On Some Methods for Computing the Roots of Polynomials [III-6, pp. 46~50]

代数方程式の解を計算するのに、多項式の係数からある二つの行列を作って、その行列の固有値として解を求める方法をいくつかの例に適用した。固有値を求めるのに2種類の Crandall の方法を用いた。

これらの方法は多項式

$$P_n(\lambda) = \lambda^n - b_n \lambda^{n-1} - b_{n-1} \lambda^{n-2} \dots - b_2 \lambda - b_1$$

を対称な行列の行列式

$$P_n(\lambda) = -\det(A - \lambda B)$$

とする式に基づいている。ここで

$$A = \begin{pmatrix} -b_1 & 0 & 0 & & \\ 0 & 0 & -1 & & \\ 0 & -1 & -b_3 & & \\ & & & \ddots & \\ & & & & \ddots \end{pmatrix}$$

$$B = \begin{pmatrix} b_2 & -1 & 0 & & \\ -1 & 0 & 0 & & \\ 0 & 0 & b_4 & -1 & \\ & & & & \ddots \\ & & & -1 & & \ddots \end{pmatrix}$$

である。したがって

$$A - \lambda B = \begin{pmatrix} \alpha_1 & \lambda & & & & \\ \lambda & 0 & -1 & & & 0 \\ -1 & & & \alpha_3 & & \\ & & & & \ddots & \\ & & & & & \alpha_{n-1} & \beta_{n-1} \\ 0 & & & & & \beta_{n-1} & \alpha_n \end{pmatrix}$$

となる。ただし  $n$  が偶数ならば  $\alpha_n = -1$ , 奇数ならば  $\alpha_n = -b_n + \lambda$  である。このようにして代数方程式の求解問題は固有値問題

$$Ax = \lambda Bx$$

に移行する。固有値を  $\lambda_i$ , 固有ベクトルを  $x_i$  で表わすことにしよう。

連立一次方程式

$$(A - \lambda B)x = r = \Sigma r^i Bx_i$$

の解は

$$x = \Sigma \{r_i / (\lambda_i - \lambda)\} x_i$$

と書けるから、 $x$  の  $x_i$  方向の成分は、 $\lambda$  がある固有値  $\lambda_i$  に近ければ、他の成分と比較して大きい。

\* 前号に引き続き第2回国際情報処理学会の提出論文(本誌 Vol. 3 No. 4 参照)を Preprint of the Proceedings of the IFIP CONGRESS 62 (Aug. 1962) から紹介します。

Crandall の方法はこの事実に基づいて、適当な  $\lambda^{(0)}$  および  $r^{(0)}$  から始めて、連立一次方程式

$$(A - \lambda^{(j)} B)x^{(j)} = r^{(j)}$$

をとくとき、 $\lambda^{(j+1)}$  を Rayleigh 商で

$$\lambda^{(j+1)} = \frac{(x^{(j)}, Ax^{(j)})}{(x^{(j)}, Bx^{(j)})}$$

と定義し、また  $r^{(j+1)}$  を

$$r^{(j+1)} = Bx^{(j)}$$

と決めることを反復し、Rayleigh 商が停留したら打切る。連立一次方程式をとくのに  $(A - \lambda B)$  の簡単な三対角形がたいへんよい。すなわち

$$Ax = \lambda Bx$$

の固有ベクトル  $x$  の成分  $x^i$  は漸化式で計算できる。

$n = 2m$  のとき

$$x^1 = 1,$$

$$x^{2k+1} = \lambda x^{2k-1} \quad k = 1, 2, \dots, m-1,$$

$$x^n = \lambda x^{n-1},$$

$$x^{2k} = \lambda x^{2k+2} + \alpha_{2k+1} x^{2k+1} \quad k = m-1, m-2, \dots, 0$$

$n = 2m-1$  のとき

$$x^1 = 1,$$

$$x^{2k+1} = \lambda x^{2k-1} \quad k = 1, 2, \dots, m-1$$

$$x^{n-1} = \alpha_n x^n$$

$$x^{2k} = \alpha_{2k+1} x^{2k+1} + \lambda x^{2k+2} \quad k = m-2, m-3, \dots, 0$$

いずれの場合も  $x^0 = P_n(\lambda)$  である。連立一次方程式  $(A - \lambda B)x = r$  の解もこれと同様な漸化式で計算できる。 $r^{(0)}$  としては

$$r^{(0)} = Bv = \Sigma Bx_i$$

を選ぶのがよい。これはすべての  $r^i = 1$  としたこととに相当するから各根が初期ベクトルに対等に表わされている。ベクトル  $v = \Sigma x_i$  の成分  $v^i$  は

$$v^{2j-1} = S_{j-1}, \quad 1 \leq j \leq m$$

$$v^{2j} = S_{n-j} - \sum_{i=1}^{n-2j} b_{2j+i} S_{j+i-1}, \quad n-2j \geq 0$$

から求まる。ここで  $S_j = \Sigma \lambda^j x_i$  は Newton の恒等式によって  $b_i$  から計算できる。根が一つ見つかったならば、再び同じ根に収束する可能性を減らすために  $r^{(0)}$  からその成分を除去して次の反復を行なう。 $\mu$  重根があるときは固有ベクトル  $Y_j^{(k)}$  は

$$y_j^{(0)} = x_j$$

$$(A - \lambda_j B)y_j^{(k)} = B y_j^{(k-1)}, \quad k = 1, 2, \dots, \mu-1$$

から求める。

$$(x_j, B y_j^{(k-1)}) = 0$$

であるから、重根の存在は Rayleigh 商の分母

$$(x^{(j)}, Bx^{(j)})$$

がいくらかでも小さくなることで検出できる。

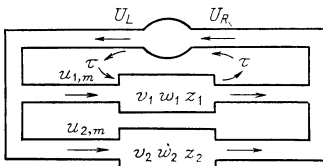
これらの考察に関連して実根の場合の例題がいくつか示されている。(清水留三郎)

### 74. 葉の拡散の数学的模型と微分差分方程式の解

B. Kotkin: A Mathematical Model of Drug Distribution and The Solution of Differential-Difference Equations [IV-4, pp. 67~69]

化学療法では化学薬品を注射してある種の細胞だけを殺し、しかも身体全体は損わないようにしなければならない。ところが薬は注射されると、拡散によって各器管に浸透し、その成分と化合して化合物を作り、それが心臓に戻ってまた各器管へ行く。そこで生理学的にも薬学的にも一般に重要な問題は“ある時間経過した後、血液や各器管における薬や化合物の濃度はどうなるか？”ということである。

2 器管模型に対する化学療法の数学的模型はすでに提案されており、それは血液の循環による時間遅れを含む微分差分方程式になる。毛管状部分を  $n$  個の攪拌器のつながりと考え、そこでの濃度を  $u_{j,m}$  で表わすことから差分が現われる。



時間遅れを含む系の解の計算には常微分方程式では出会わない難しさがある。すなわち時間遅れ  $\tau$  の間の函数値を記憶できるだけの大きな容量の要求と、この種の函数方程式の解の特徴である高次の導函数の不連続性の存在である。

この種の系を解く一つの方法は先ず区間  $[0, \tau]$  について適当な方法で積分し、終点  $t=\tau$  での値を次の区間の初期値に使い、この過程を所望の時間続けることである。

ここでは積分に予測子-修正子法を用いた。2 進の計算機向きに時間遅れ  $\tau=2^{-\alpha}$ 、刻み  $H=2^{-\beta}$  と選び、 $\tau$  を  $M=2^{\beta-\alpha}$  等分し、各循環が  $\tau$  の正確な整数倍で終るようにした。初めの  $3H$  をさらに半分に細分して、修正 Euler 法:

$$y^{(p)}_{i,n+1} = y_{i,n} + hf_{i,n},$$

$$y^{(c)}_{i,n+1} = y_{i,n} + \frac{h}{2}(f_{i,n} + f^{(p)}_{i,n+1})$$

を初めの 3 点に適用して出発値を計算し、次に 4 次の Adams-Moulton 法:

$$y^{(p)}_{i,n+1} = y_{i,n}$$

$$+ \frac{h}{24}(55f_{i,n} - 59f_{i,n-1} + 37f_{i,n-2} - 9f_{i,n-3}),$$

$$y^{(c)}_{i,n+1} = y_{i,n}$$

$$+ \frac{h}{24}(9f^{(p)}_{i,n+1} + 19f_{i,n} - 5f_{i,n-1} + f_{i,n-2})$$

を  $h=H/2$  として次の 3 点に適用し、さらに  $h=H$  として積分を進めた。いずれの場合も修正子は 1 度だけ用いた。

この方法を解析解の知れている微分差分方程式について検査したところ、 $t$  の比較的小さい範囲で十分正確で有効であった。したがって  $t$  の小さいところに興味のあるこのモデルに適用できることが保証された。 $t=30$  分まで計算するのに IBM 7090 で約 2 分要し、計算された定常状態での濃度は物理的考察から計算した値と一致した。

しかし時間遅れ  $\tau$  のために刻み  $h$  を固定しなければならないので、適当な精度を保つためには、積分の初期だけで必要な小さい刻みの値を積分の全過程を通じて使わなければならない。(清水留三郎)

### 75. イギリスにおける銀行のデータ処理

R. Hindle: Data Processing in English Banks [I-1, pp. 3~7]

イギリスでは小切手の流通が非常に多く、ロンドンの小切手決済所では 1955 年以来、委員会を作って小切手その他の伝票整理を自動化することを研究している。それは、伝票を厚くしてカードとし、磁気インクで印刷するものである。符号は、フェランチ社やブル社のものも検討したが、結局アメリカの銀行にならって E 13 B 符号を用いることに決めた。

E 13 B の読み取り機や分類機に適合させるためには、紙質についてカリフォルニアのスタンフォード研究所がいくつかの銀行と協同で研究し、こまかい条件を決めた。

印刷の要領については、カードの下縁から 5/8" を決済専用の欄とし、金額その他必要な情報を数字で印刷した。計算機は 1961 年頃から使われだしているが、入出力には紙テープを主に使っている。小切手カードを直接計算機に読み取ることはまだやっていない。た

だ、記憶装置をもった計算機を使うようになったので分類は計算機の中でするようになって来た。

イギリスの銀行システムは集中的でないが、全国の各支店の計算を一カ所にまとめることは、紙テープを使う限り難かしく、運搬に便利な同一市内のものをまとめるのにとどまっている。パークレイ銀行では、計算機を設置する前から、支店間の通信を開発し、最近では電話線を用いて集中的にデータを処理することを試みている。

現在では、貸借対照表とその統計、コピー、不渡かどうかなどのモニタなどを毎日印刷していて、小切手に対して支払うべきかどうかは直ちにわかるようになった。

なお、計算機の購入や、情報の自動処理、インク、通信などの開発に莫大な金をかけているが、自動銀行システムができると経済的に引き合うことを知っているからである。(川合英俊)

76. 分類まぜあわせ手法の解析

W.C. Carter: Mathematical Analysis of Mergesorting Techniques. [II-1, pp. 13~16]

分類まぜあわせ手法の比較の普遍的尺度として、全データ上のパスの回数 ( $M$ ) をとる。polyphase merge での所要ステージ数、巻きもどし時間は等価的なパス数になおす。これは、分類過程が magnetic tape limited という前提があって有効である。

一般に、分類まぜあわせ手法の Algorithm は、 $j$  番目のテープ上の  $i$  ステージ終了後の連の長さ  $d_{ji}$  に関して

$$d_{ji} = \sum_{v=1}^m a^{(j)}_{vi} d_{v0} \quad j=1(1)m; i=1(1)N;$$

をあたえることである。  $N$  は、所要ステージ数。

上式を、 $i$  を自変数とする差分方程式とみて、

$$d_{ji} = E^i d_{j,0}$$

を左辺に代入し、 $i=1(1)m$  をとりだして、 $(d_{10}, \dots, d_{m0}) \neq 0$  によって、

$$\det(a_{1\alpha} - E^\alpha, a_{2\alpha}, \dots, a_{m\alpha}) = 0$$

$a_{1\alpha} - E^\alpha, a_{2\alpha}, \dots, a_{m\alpha}$  は列ベクトルをうる。この根  $\beta_1, \beta_2, \dots, \beta_m$  によって、

$$d_{ji} = C_{j1}\beta_1^i + C_{j2}\beta_2^i + \dots + C_{jm}\beta_m^i$$

となる。  $C_{j,k}$  は初期条件  $d_{j,0}, \dots, d_{j,m-1}$  からさだめる。

一方、連の個数については、 $d_{ji}$  との双対性によってとめる。以上が、この解析法の基本思想である。たとえば、

(1)  $2n$  way merge では、

$$d_{j,i+1} = nd_{j,i}; \quad d_{ji} = C_j n^i$$

$$m_{j,N-i-1} = hm_{j,N-i}; \quad m_{j,N-i} = C_j n^{i-1}$$

(2)  $(n+1)$  本のテープをつかった cascade merge では、

$$d_{j,i+1} = \sum_{p=1}^{n+1} d_{p,i}; \quad d_{ji} = C_{j1}\beta_1^i$$

$$m_{j,N-i-1} = \sum_{k=1}^{j+1} d_{k,N-i}; \quad m_{j,N-i} = C_{j,\beta_1}\beta_1^{i-1}$$

(3)  $(n+1)$  本のテープをつかった polyphase merge では、 $j=1$  を output tape として、

$$d_{1,i+1} = \sum_{k=1}^n d_{1,i-k+1}; \quad d_{1i} = C_{11}\beta_1^i$$

$$m_{1,N-i-1} = \sum_{k=1}^n m_{1,N-i-k-1}; \quad m_{1,N-i} = C_{11}\beta_1^{i-1}$$

ここで、 $d_{1i}m_{1i} \doteq C_{11}^2\beta_1^{N-1}$  で、略々一定であることに注意する。(2),(3)で、 $\beta_1$  は特性方程式の最大根。

以上、3種および、cascade merge で、各ステージでとられる copy の過程を、逆読みのできるときには rewind でおきかえる方法を  $A$  法として、4者のパス回数を算定する式は表のようになる。

Type of Sort	Number of Passes over all Data ( $M$ )		
	Reading backwards	Reading forward only	Output tapes
Standard Cascade	$N$	$N(1 + 2/nR)$	$1, 2, \dots, n/2$
Polyphase Method A	$N$	$N(1 + 1/R)$	$1, 2, \dots, n-1$ (8)
	$p_1 N_1$	$p_2 N_1(1 + 1/R)$	1
	$N(p_1 + p_2/R)$	$N(1 + 1/R)$	$1, 2$ to $1, 2, \dots, n-1$

この表で、 $R$ は、巻きもどし速度と、読みこみ速度の比、 $P_1$  は1ステージで処理されるデータの割合、 $P_2$  は巻きもどされるデータの割合である。

以上の分析から、使用テープ数が8以下だと、poly phase merge が最良で、 $n=8$  で逆読みのできるばあいには、 $A$ 法がよく、それ以外では、通常の balanced method がよいことを、結論している。(久保宏志)

77. 碁を打つ学習機械のシミュレーション

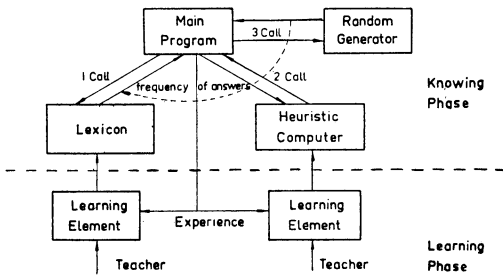
H. Remus: Simulation of a Learning Machine for Playing GO [XII-4, pp. 192~194]

学習をする機械の研究は興味深い。この論文はその一例として IBM 704 に、碁に関する最小の情報を与えるだけで碁を打たせようと試みたものである。もちろんその目的は学習のシミュレーションにあり、したがって碁を打つ能力の高さは問題ではない。碁のルールは簡単で、しかも最適解はまだ与えられていず、またいわゆる順列組み合わせによる手の選び方をしようとするばきわめて場合の数が多くなり計算不能となるという点から、学習の例題としては好適である。

正規の碁のルールとしては 19×19 の盤が使用されることはいうまでもないが、この場合可能な手の数は、 $3^{361} \approx 10^{172}$  程度になるので、本論文では 11×11 の盤を使用し  $3^{121} \approx 10^{57}$  程度の範囲で検討を行なっている。

プログラムの上では 11×11 の 0 と 1 の 2 進マトリクスを 3 組用意し、一つは白石があれば 1 なければ 0 他は黒石、および空白のそれぞれに対応し同様に 1 または 0 で盤の状態を示すことにしている。

ゲーム中はメインプログラムによって (図参照) 碁のルールに従った可能手を決める。第一にあらかじめ作られた辞書の中を見て現在位置に対して前に有効手だったものがあるかどうかをさがす。



碁を打つ機械の構造

一つの点に対して有効手が対応するが、逆に辞書には一つの有効手に対応する幾つかの点が記載されている。もちろんその有効さは過去のものであるが、プログラムの一操作ごとに成功率最大の手を残しながら、メインプログラムによる可能手の数を制限してゆく、

可能手に対して 16 個のパラメータ——たとえば石のグループの逃げ道 (freedom) の数、とられる石の数、敵の最終手からの距離など——を計算し重みをつけた一次式を作って判定する。こうして minimax の方法で 6 手の深さ計算する。(もし最良の手が一つ以上あれば乱数で撰択する。)

各手ごとに計算した値が増える時は良い手であり、前の手よりも急速に減る時は悪い手である。

ゲームが終了と良い手は辞書に登録される、計算機は全ての手についてパラメータを計算し、辞書の中の手がもし良い手であれば、成功率  $a/b$  を  $(a+1)/(b+1)$  とし、そうでなければ  $a/(b+1)$  とする。こうしてこの値が 0.05 以下になったものは辞書からははずす。

辞書に登録されている「局面」の数は 1000 個としておき、これを次のゲームに備えて再分類する。

IBM 704 ではこの Simulation のために SAP 言語で 8,000 個の命令と、常数および入出カサブルー

チンが用意され、15,000 個の点が辞書用に記憶装置内に割り当てられた。minimax 処理には 600 語を要し、学習用に 672 語のテーブルが作られた。辞書は 2 本の磁気テープで分類される。

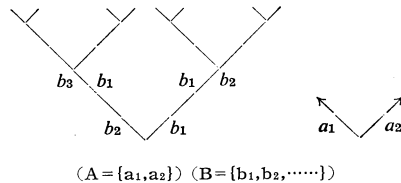
一つの手に対して判別因子を計算するのに 100 ms を要する。また辞書内の局面に対する可能なシフト、回転、反転に対する局面全部の調査は 0.5 秒である。

この論文の発表時点でプログラムはほとんど完成された。(石井康雄)

78. 有限・組み合わせオートマトン, プログラミング・テープ付チューリング機械

J. Bečvar: Finite and Combinatorial Automata. Turing Automata with a Programming Tape. [XI-2, pp. 169~171]

最初空でない任意の集合  $M$  から、その集合の要素を任意に選んで作られる時系列の集合  $M^S$  を定義する。アルファベットの集合を  $A, B$  としたとき、 $A^S$  を  $B^S$  に写像する作用素を  $\Omega[A, B]$  と書き、特に系列の長さを変えないような作用素を IS 作用素と名づけて  $\Phi[A, B]$  と書くものとする。そうすれば  $\Phi[A, B]$  は図のような樹枝状の図表で表わせることになる。



この図の場合、 $A^S$  の要素は各時点で  $a_1, a_2$  いずれかをとるから、それに対応する  $B^S$  のその時点での値は  $b_1, b_2, \dots$  のいずれかをとって、樹枝状図表を順次上方にたどって行くことになる。この樹枝状図表に現われる相異った枝分れの種類の数を  $\Phi$  の重み  $w$  といい、 $w$  が特に有限であるとき、 $\Phi$  は有限的であるという。

同期性決定論的オートマトン  $A[X', Z']$  では、一定の初期状態から出発する入力系列  $\{x(t)\}_{t=1}^{\infty}$  に対して出力系列  $\{z(t)\}_{t=1}^{\infty}$  を生ずるが、 $z(t)$  は  $1 \leq u \leq t$  なる期間中の  $x(u)$  に関して一意的に決定する。したがって  $A$  は IS 作用素  $\Phi_A[X', Z']$  で記述することができる。  $\Phi_A$  が有限的であるとき、 $A$  を有限オートマトンというが、このときの  $\Phi_A$  の重みは  $A$  の内部状態の数に相当することになる。ここで適当な長さの周期をもつ入力系列を与えたとき、初期状態の如何にかかわらず  $A$  の出力が一定周期の後に一意的に決ま

ようになるかどうかという決定性の問題が生ずる。

次に両側に無限に長いテープのついた有限オートマトンを考え、これを組み合わせオートマトンと名付ける。入力テープと結合し、入力系列はテープに順次に書き込まれるが、オートマトンはその内部状態と読み取ったテープ上の記号とによって、次に読みとるべきテープの位置を選択するものとする。入力アルファベット  $X'$ 、テープのアルファベット  $X^*$ 、内部状態の集合  $Y$ 、出力アルファベット  $Z$  はともに有限で、 $X'CX^*$  と仮定する。このような組み合わせオートマトンでは決定性となることもあるが、一般には有限でない。

以上の考察を土台としてプログラミング・テープを持った半万能チューリング機械  $U$  が与えられた。この機械は2本のテープを持つ組み合わせオートマトンで、その一方のテープ（プログラミング・テープ）には真似されるはずの特定のチューリング機械  $T$  の内容が記される。そうすると他のテープから入った入力系列に対して  $U$  は見かけ上  $T$  と全く同じ働きをなすようにすることができる。実例を用いてこのことが示され、その内容が論ぜられている。（飯島泰蔵）

### 79. 音声タイプライタ

T. Sakai, S. Doshita: The Phonetic Typewriter [XIII-3, pp. 204~208]

人間の音声を入力としてカタカナを印字する音声タイプライタを3000個のトランジスタ、5000個のダイオードをもちいて作った。処理手順は、連続した音声の自動分割とこれで得られた音素のボタン認識とにわけられ、とくに前者について詳しく述べてある。

デジタルな論理が多くもちいられ、アナログな要素は入力の初期の段階でデジタル量に変換される。すなわち、時間を10msの時程にきりおのおのをフィルタで周波数にわけ、各時刻における各周波数の強さをデジタル量であらわし、それをさらに統計的に定まる閾値より強いか、弱いかによって1, 0であらわすと、Bool化されたソナグラムが得られる。

隣接した時刻のあいだのHamming距離は母音時程では小さく子音時程では大きい。また一つの周波数を時間軸に沿って走査したとき、連続6個1があったら（または、5個中4個1があったら）その周波数は定常であるとみなす。そして定常な第一ホルマント、第二ホルマントの存在する区間が母音時程である。

このようなHamming距離と定常性とを手掛りとして母音時程をまず定め、これにもとづいて分割する

と音素が得られる。

音素のボタン認識には、音素を構成する周波数の組み合わせを利用する。たとえば第一ホルマントの高さは5段階、第二ホルマントの高さは8段階にわけられ、その組み合わせによって五つの母音が識別される。

（前田陽子）

### 80. 音声合成ルーチン: MUSE

A.W. Slawson: MUSE: A Sound Synthesizer [XIII-4, pp. 209~211]

計算機から人間への通信路ははなはだ貧弱なものしかなく、しかもとくに視覚的表示にかたよっている。聴覚的な方法も、実時間動作には有利と思われるがあまり検討されていない。これらを念頭において、特別な金物によらずに、計算機から完全に制御された音声出力の合成を試みた。

一連の音声の記録を人間が一連のMUSEステートメントに書きなす。一つのステートメントは、ある時刻における基音の高さ、強さ、倍音の高さ、帯域、ノイズなどのパラメータを含む。ステートメントを標本抽出する時間間隔は不定で、だいたい1音素あたり2~3ステートメントになる。たとえば“Sense switch one, down”という音声は29個のステートメントで記述された。

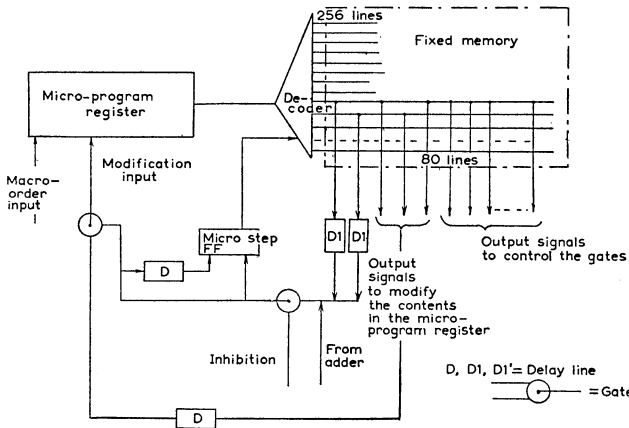
ステートメントを穿孔したカードを計算機に与えると、MUSEプロセッサがステートメントの間を内挿して一定間隔の時点で音圧を算出し、6ビットの数値にまらめ、磁気テープに書く。一連の音声は磁気テープ上の一つの長いブロックとして記録される。磁気テープは計算機から離され、適当なデジタル・アナログ・コンバータで音波に変えられる。

MUSEの意義は当面は研究的な利用に限定されるが、将来オフラインの点をあらためればオペレーティングシステムに組みこんで操作員への通信にも使えるだろう。（西村恕彦）

### 81. KT pilot 電子計算機—ホト・トランジスタ固定記憶装置をもったマイクロプログラム計算機

H. Hagiwara, K. Amo, S. Matsushita, H. Yamauchi: The KT Pilot Computer—A Microprogrammed Computer with a Phototransistor Fixed Memory [XIX-3, pp. 318~321]

電子計算機KT pilotの特長・制御方式・回路に



について述べている。この計算機の特長は、1. マイクロプログラム方式を採用し論理構造を簡単にした、2. 非同同期式である、3. マイクロプログラムを記憶するものとしてホットランジスタ固定記憶装置を用いた、4. 回路素子に超高速メサトランジスタ・メサダイオード・ゴールドボンドダイオードを使用した、などである。

この計算機は9個の演算レジスタおよび1個ずつの入出力バッファレジスタをもっており、加算・排他的OR・転送・補数算・右シフト・左シフトの6個の基本的操作機能がある。他の演算・論理操作は、これら基本的操作の組み合わせの実行によって行なわれる。この組み合わせおよび実行の順序は、固定記憶装置に記憶されたマイクロプログラムによって決められ、かつ制御される。

この制御方法は、まず主記憶装置から読み出されたマクロ命令が8ビットのマイクロプログラムレジスタにおかれ、これが解読されて固定記憶装置の256本の横線の中の1本を選択する(図参照)。この256本の線それぞれが一つずつのマイクロ命令に対応している。すなわち、横線1本を選択によって固定記憶装置の行列にしたがい、80本の縦線の中の何本かが選択される。この縦線のそれぞれは演算および入出力レジスタ間、あるいは6個の基本的操作の回路の間のゲートにつながっており、演算論理操作が実行される。これでマイクロ命令の1ステップが終わったわけである。今選択された80本の縦線のいくつかは、次のステップのマイクロ命令のために、マイクロプログラムレジスタを修飾するのに用いられる。また他のいくつかは、次のステップの実行開始の時間信号を出すのに用いられる。本文ではさらに、固定記憶装置にホットランジスタを

使用した理由、特に記憶内容が容易に変えられる利点について述べ、さらに基本回路や使用トランジスタ・ダイオードについて説明してある。(白井国雄)

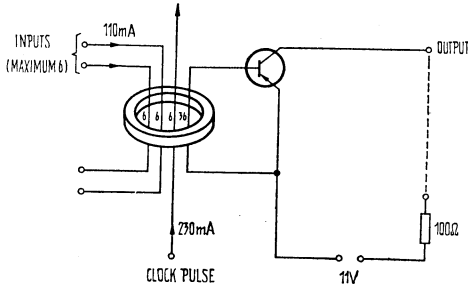
### 82. 内蔵されたプログラムで制御される非常に小形の電子計算機

H. Gumin, F.K. Kroos: A Very Small Electronic Digital Computer with Stored Program Control [XVIII-1, pp. 287~290]

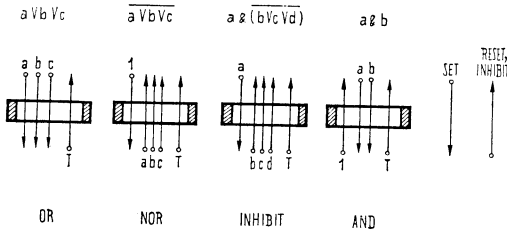
この計算機は、各所で発生したデータを集めて計算機に伝送し、計算を行ない、さらに計算結果を分配するというを目的としたもので、伝送・計算はすべて中央処理装置に貯えられたプログラムで制御される。安価な計算機を作ることを目標としている。

このシステムは、環状にした穿孔紙テープに記録されたプログラムで制御される。この種の応用目的の計算機のプログラムは、経済性を考えれば、この程度で充分である。

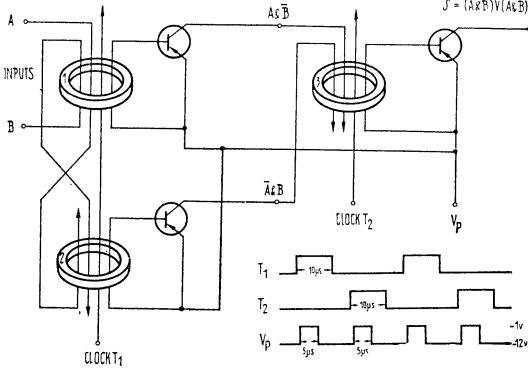
本文では、このシステム全体の情報の流れ・演算制御装置での制御方法・命令体系などについて述べられている。この計算機は、データの伝送・入出力が電信印字機の数値でおさえられるので、速度より安価ということを目標にしており、そのために、ブロック組み立式のコア・トランジスタ論理回路を使用している。基本ブロックは、第1図のごとくで、入力信号の組み合わせで第2図のごとく、OR・NOR・INHIBIT・ANDの論理回路を作ることができる。この計算機では、この回路の電源として、パルス状の電源を使用し、トランジスタやコアの不均一による影響を除いている。電源パルスはトランジスタのエミッタに加えられるが、その前に電源パルスの幅より広くベース側には導通すべきかどうかの条件がかかっている。それゆえ、各出力波形はエミッタに入る電源パルスの波形に対しあまり変形されないで、各ブロックについて出力波形の不均一性が少ない。パルス状電源を使用したことの利点として、さらに次の3点をあげている。a) トランジスタはベース接地型で使用されるので、消費電力が少い。また比較的低速のトランジスタを使用することができる。b) 電源パルスがOFFのときコレクタ遮断電流が流れないから不要な発熱がない。c) セットされていないコアをタイミングパルスで読み出した場合に生



第 1 図



第 2 図



第 3 図

ずるノイズが、おさまったとき電源パルスが来るから、ノイズが出力に出ない。

半加算器の回路図は第 3 図のとおりである。

(白井国雄)

83. 除算のアルゴリズム

A. Svoboda: An Algorithm for Division. [XVIII -5, pp. 307~313]

次のような機能を持つ演算装置に適用可能な除算方式のアルゴリズムを述べている。

- 1) 10 進表示であること。

2)  $(m \times a - b)10^n$  を計算できる装置を持つこと。ただし、 $m$  は 1 桁の 10 進数、 $n$  は整数であり、また  $ab$  は 10 進表示された数値である。

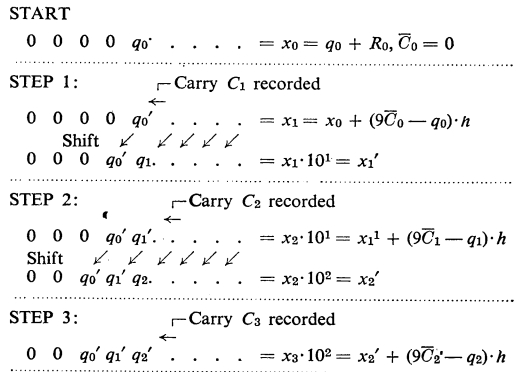
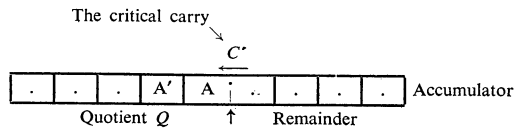
この除算方式は、除数が  $y=1+h$  の形で表わされ、 $h$  が小さい時、商の有効桁が被除数の有効桁より、十分な精度で求められることに基いている。

アルゴリズムは二つの部分、ALG 1 と ALG 2 とに分かれている。ALG 1 は被除数  $X$  と除数  $Y$  とに対して

$$z = x + y = X + Y = Z$$

$$1 \leq y = 1 + h < 1 + 10^{-1}$$

の関係を満たすような  $x, y$  を求めるためのものである。その手法として  $Y$  の MSD の値によって順次に定数  $C_n$  を索引するテーブルを用意し、 $Y_{n+1} = C_n \cdot Y_n$ ,  $X_{n+1} = C_n \cdot X_n$  を行なうことにより  $x, y$  を得る。このステップは最大 4 ステップ、平均 1.982 ステップで終了する。



ALG 2 は、ALG 1 によって得られた  $x, y$  より、商  $z$  を得るためのもので、それを具体的に実施する方法は、第 1 図のごときのものである。すなわち、まず被除数  $x_0 = q_0 + R_0$  を累算器に入れておき、第  $n$  番目のステップが次のように定義される手順を繰り返し実行する。ただし  $q_n$  は  $x_n$  の整数部分である。

ステップ  $n$ ; 累算器に  $(9C_n - q_n)h$  を加算し、その時に整数桁へ出る桁上げを  $C_{n+1}$  とする。さらに累算器の内容を左へ 1 桁シフトする。

このステップを必要な商の桁数回だけくり返すことによって、商  $z$  を累算器に得る。

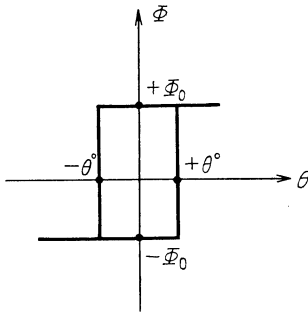
本文中にはアルゴリズム、証明および例が示されている。  
(鶴飼直哉)

84. フェライト磁心のみを用いた論理回路

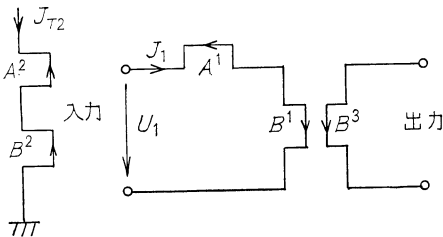
U. Hölken: Ferrite Core Logic in All-Magnetic Technique [XVII-2 pp. 274~278]

フェライト磁心のみを用いた論理回路は、磁心ダイオード論理回路にくらべて信頼度がたかい。ただし情報の流れに方向性をつけるために少なくとも2箇の磁心が必要とする。

第1図に示すように、矩形磁化曲線の残留磁化状態  $+\Phi_0$ ,  $-\Phi_0$  で情報 "0", "1" をあらわす。基本回路は第2図に示すように蓄積磁心Aと出力磁心Bとからなり、少なくとも3ステップの時間によって、蓄積磁心への情報の挿入、蓄積磁心から出力磁心への情報の転送、出力パルスの発生とクリアを行なわせる。"1" に対しては等振幅、逆極性の2個の入力パルスを受けとり、同じ形式の出力パルスを送出する。"0" に対しては入力パルスを受けとらず、また出力パルスを送出しない。

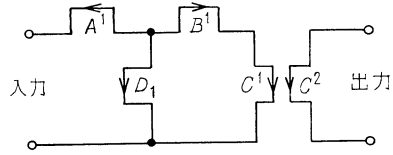


第1図 矩形磁化曲線

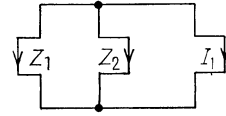


第2図 基本回路 (クリア状態)

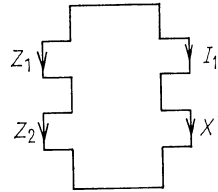
磁心Aをセットするに必要な、入力第1パルスによる電流を  $\theta_e$ 、出力第1パルスによる最大電流を  $\theta_a$  とすると、電流利得  $\theta$  は  $\theta = \theta_a / \theta_e$  であらわされる。第3巻線にバイアス電流を流すことによって電



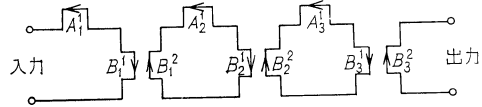
第3図 4磁心回路 (クリア状態)



第4図 AND 回路



第5図 OR 回路



第6図 シフト・レジスタ

流利得を1から1.5の程度にすることができる。さらに大きい電流利得を得るには、第3図に示すように、磁心Dを追加して枝回路をつくらねばよい。枝回路を多くつくるほど電流利得は増すが、情報の転送・送出不に必要な時間ステップ数も増加する。

第4, 5, 6図は基本回路による AND 回路, OR 回路, シフト・レジスタの例である。シフト・レジスタの試作例では 2.5 Mc の繰返しパルスで動作させることができた。これは繰返しサイクル 625 kc に相当する。  
(山中 観)