文　献　紹　介

A：数値解析　B：プログラミング　C：計算機方式
D：回路および機器　E：オートマトン　F：応用その他

A-19. FORMAC の算法を設計した経験

R.G. Tobey: Experience with FORMAC Algorithm Design [CACM, Vol. 9, No. 8, Aug., 1966, pp. 589～597]

記号計算のプログラム FORMAC を作成し実験をおこなったところ，いろいろ改善の必要をみとめた．

まず括弧をはずして式を展開するのに素朴なやりかたでは途中の式を保存するのに多くの記憶容量を必要とする．

$$(a+b)(a-b)$$

展開　$\longrightarrow a^2-ab+ba-b^2$　4項

単純化　$\longrightarrow a^2-b^2$

そこで展開を1項ずつおこない，そのたびに単純化すると，時間はかかるが容量が節約できる．

$$(a+b)(a-b)$$

一部展開　$\longrightarrow a^2-ab+ab$　3項

単純化　$\longrightarrow a^2$

一部展開　$\longrightarrow a^2-b^2$

また $(\Sigma a_i x^i)^n$ のような多項式を展開するにはもっと巧妙なやりかたが必要である．

次に記号微分については公式を単純に適用するだけでは，

$$\frac{d}{dx}x^2 \rightarrow \left(x^{2-1}\cdot\frac{d}{dx}x\cdot 2\right)+\left(\log x\cdot\frac{d}{dx}2\cdot x^2\right)$$

のような中間結果があらわれるので，微分にさきだって式の部分を走査し微分する変数を含まない部分は単純に空とし，時間と容量とを節約した．

ある式を FORMAC で微分した結果を，よそが微分した結果と照合しようとした．ところが結果は60字×30行にもおよぶのでひとめでは，正しいかどうかわからない．FORMAC の照合ルーチンはすべての括弧をはずして展開してしまうので，記憶容量がまったく不足した．そこで内側の因数や項を順次に助変数でおきかえて，展開しないままで照合するルーチンをつけくわえ，この場合はうまくいった．しかし，たとえば，

$(x+y)(x-y)$ は $s\cdot t$ となり

x^2-y^2 は $u-v$ となるから

いつでも成功するとはかぎらない．

式を表示するときにうまい助変数を用いるとか，二次元的な表示方法，たとえば

$$\frac{1}{2}\cdot\frac{a\cdot x+b\cdot y}{\sqrt{a}\cdot\sqrt{b}}$$

の形式にするとかしないと，実務家からは，敬遠される．　（西村恕彦）

B-20. 生物学における図形認識

R.S. Ledley 他：Pattern Recognition Studies in the Biomedical Sciences [Proc. SJCC, 1966, 1966, pp. 411～430]

生物学の研究においては，データが写真などの図形として与えられることが多い．これらの図形から必要な情報を取出す作業を計算機に行なわせるため開発されたシステムの報告である．入力図形はフィルムで与えられ，飛点走査管による走査によって計算機（IBM-7094）の記憶装置に読み込まれる（1枚当たり 700×500 のメッシュ，7水準の白黒値または，1,000×800，2水準）．続いて，記憶装置内での「内部的な走査」によって図形の境界線が追跡され，セグメントに分解され，リストにとられる．以上は「FIDACSYS」なるシステムプログラムによって自動的に行なわれるが，ほかに点走査のシミュレーション言語「BAGSYS」が用意されていて，任意のプログラムを書くことができる．走査点は，位置とそこでの白黒値で特徴づけられ，複数個定義でき，おのおの名称がつけられる．命令には，走査点を動かす「MOVE」，白黒値を閾値と比較する「TEST」，白黒値を変える「CHANGE」，二つの走査点を結ぶ直線に垂直な方向へ別の走査点を進める「NPROVE」などがある．この最後の命令は，対象物の境界線（近接した2点は線分の表現となる）から内部へ入って行き，他の側の境界線を探る場合などに使用される．「CHANGE」は，たとえば，一度走査した点を特別な値に変え，他と区別するのに使われる．応用例として，1）神経細胞の樹枝状突起，2）シュ

リーレン写真，3) 兎の脳細胞，4) 染色体地図の解析が紹介されている．1) では，最初に走査点が対象物を探し出し，周囲を回って樹枝状突起を識別し，その分枝点，先端点の位置を確定する．4) では，有糸分裂の際の染色（分）体の先端点および動原体の位置を見出し，長さ，面積を測定し，分類作業を行なう．

このような，プログラムによる走査では，制御が任意にできるほか，走査点を複数個同時的に動かすことができるので，相当複雑な操作が行なえる利点がある．しかし，点走査であるかぎり，雑音，雑物などの影響が大きな問題となると思われるが，この報告ではふれられていない．（葛城純夫）

B-21. Atlas Autocode の主な特徴

R.A, Brooker. J.S. Rohl and S.R. Clark: The Main Features of Atlas Autocode [Computer J., Vol. 8, No. 4, Jan., 1966, pp. 303～310]

本論文はイギリスマンチェスタ大学で開発された大形高速計算機 Atlas 用に作製された ALGOL 型コンパイラのおもな特徴について述べたものである．その言語は Atlas Autocode と呼ばれ ALGOL の各項について比較説明が与えられている．

ブロックは ALGOL ブロック構造が使われ，一つの完全なプログラムは **begin** で始まり **end of program** で終る．ラベルは simple numerical labels と switch label の2種でそれぞれ N: と A(N): と記述される．したがって飛越し命令は ⟶N または ⟶A(I) という型をとる．その他 expression, type, assignment および標準関数についてそれぞれ説明が与えられているがほとんど ALGOL と変わりがなく，むしろ単純化されている．一つのプログラムに付属する routine および function について，ALGOL では本質的にパラメータをもった named block であるが，ここでは routine の頭に次の型のリストを置くことで処理される．

$$\left\{\begin{matrix}\text{ROUTINE}\\ \text{TYPE}\end{matrix}\right\}\{\text{NAME}\}(\{\text{FORMAL PARAMETER LIST}\})$$

routine type は3種類あり，それに対応して exit 命令が決まる．また routine name はそれが参照される前に declare されていなければならない．そのための specification および call statement の型が与えられている．

次に入出力 function に関しては，大部分 PERM routine で行なわれる．作製された routine の中で最も有用なもののリストが示されている．

また目的プログラムの誤りを検出する方法が数種与えられている．これらの誤りは supervisor プログラムやコンパイラなどで検出され，trappable または，untrappable に分けられる．ここには supervisor で検出される誤りのリストが与えられている．しかし，ある trappable な誤りに関しては user の与える情報によりプログラムを続行できる．たとえば次のような命令

fault 1, 5 ⟶ 3, 2, 4 ⟶ 1

を使用する．この意味は，1, 5 型の誤りの場合は，statenent No. 3 へ飛び越し，2, 4 型の誤りの場合は同じく1へ飛び越す．コンパイルの終了時には，種々の情報が印刷される．たとえば全プログラムに使われたメモリ数など．

最近の開発について arithmetic type では複素数と新しい expression の型の導入 structure においては list processing でない木構造を定義し組立て解析するための方法を導入している．

最後に付録として faulty programme の例とそれに関する monitoring が与えられている．

（長谷文子）

B-22. アセンブルかコンパイルか

C.J. Shaw: Assemble or Compile? [Datamation Vol. 12, No. 9, Sept., 1966, pp. 59～62]

アセンブラは語彙は大きいが文法が簡単だから，教育訓練の手間はコンパイラとかわらない．プログラム作製や保守の労力はソース・プログラムのステップ数によるのでコンパイラのほうが有利である．とくにデータの型や構造の変更にともなうプログラムの変更についてそうである．コンパイラのほうがあきらかに読みやすく，文書化もしやすいので，通信や機種変更に適する．

最良のプログラマならばアセンブラによるほうが効率の高いプログラムを作れるだろうが，数年間保守されてきたコンパイラならばその差はたかだか 10～15 %であろう．中程度のプログラマならば，アセンブラでもコンパイラでも同じぐらいの効率の機械語プログラムとなる．それに記憶容量や実行時間が費用に関係しないこともある．

SDC で作ったいくつかのプログラムの統計を示す．

	アセンブラ	JOVIAL	
コーディング	322	555	機械語/人月
機械時間	24	12	手直し時間/1000 機械語
ドキュメンテーション	88	40	ページ数/1000 機械語

（西村恕彦）

B-23. 微分のための形式的体系

E.K. Blum: A Formal System for Differentiation [JACM, Vol. 13, No. 4, Oct., 1966, pp. 495～504]

数式の微分をするいくつかの計算機プログラムシステムが作成されたが，それらは与えられた唯一の数式を微分するか，または与えられた方程式によって定義された複雑な関数を微分する手順を記述するためのものであった．本文で述べるのは，方程式の集合およびそれによって定義された関数を与えて求めるべき微係数を指定すると，人間が計算の手順を示さなくても，自動的に手順を決定してその関数の微分または偏微分を計算するためのプログラムシステムである．

論理学において well formed formulas, 公理および誘導法則を用いて形式的体系を記述するのと同じ仕方で，方程式，微分公式および方程式の集合から他の方程式を導く規則を用いてプログラムシステムを形式的に記述し，最後に，このシステムにおける数式微分のための一般的手順を述べている．

例　問題とプログラム

問

Let $x=yx_0$, where

$y=t-(\theta-\sin\theta)x_0$,

$t=\theta-x_0\sin\theta$.

Consider $x=x(t, x_0)$ as a function of the independent varinbles t and x_0, and compute $\partial x/\partial x_0$.

プログラム

```
X=Y*X 0,
Y=T-(THETA-FSIN(THETA))*X 0,
DX 0(THETA)=FSIN(THETA)/(1-X 0
  * FCOS(THETA)),
OUTPUT: DX 0(X)
```

上の問に対して下のようなプログラムを与えるとシステムは

```
DX 0(X)=-(THETA-FSIN(THETA))-
  (FSIN(THETA)/(1. -X 0*FCOS
  (THETA))-FCOS(THETA)*FSIN
  (THETA)/(1. -X 0*FCOS(THETA)))
  *X 0+Y
```

を output する．このシステムでは数式の単純化を行なわず，また実際には数式を前置記法で入出力する．

（二村良彦）

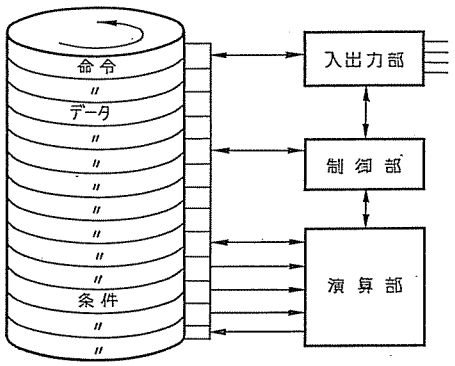
C-24. リストむき計算機

Joseph E. Wirsching: A List-Oriented Computer [Datamation, Vol. 12, No. 12, Dec., 1966, pp. 41～43]

本論文ではリスト形式のデータを処理する目的にはドラム，ディスクなどの回転形記憶装置を使用して，通常のランダム・アクセスメモリを使用したものとほとんど等しい速度で，かつ安価な計算機が作れることを指適，提案している．

たとえば $c_i-a_i+b_i$ を大きな i について，計算する場合，普通インデックスを使用しても，一組の計算に4命令程度を要し，命令フェッチの時間が全所要時間に対し，大きい比重を占めてしまう．そこで命令を1回しかとって来ないですます，一命令多オペランドの思想が生じる．これは各データのリストをドラムの1トラックにそれぞれ記憶しておき，これらを同時に読み出して演算し，別のトラックに書いていくようにすれば実現できる．命令フェッチの時間がないから，ドラムからの読み出し速度で演算できる（当然のことながら乗除算ではドラムの転送速度で演算できないから，適当なバッファと先回り制御が必要である）．

この種のリスト形データの処理は多数の変数についてメッシュに別けて計算を行なう場合（典型的な水力学の問題では 10～20 の変数のリストがある）に有効



第1図　NOVA 計算機のブロックダイアグラム

である．

第1図に NOVA の構造を示す．回転形メモリは各トラックごとにヘッドを有する容量 100～200 万語のもので，同時に2個のオペランドと1個の制御情報，1個の命令を読み，1個の演算結果と制御情報を書きこめるものを使用する．価格はディスクを使用するとして，ディスクに5万ドル，演算制御部に5万ドルで，約10万ドル程度である．

一命令多オペランドという考え方の一適用例であるが，かなり限られた用途にしか使用できないであろうし，速度についても，ほとんど等しいというだけで，定量的検討はなされていない．また値段も，最近安い汎用計算機が出回ってきており，10万ドルは必ずしも，著者のいうほど非常に安価とはいいがたい．

（飯塚　肇）

D-25. 大容量磁性薄膜記憶装置のセンス記号の伝搬

F.C. Yao: Propagation of Sense Signals in Large-Scale Magnetic Thin Film Memories [IEEE Trans. EC, EC-15, No. 4, Aug., 1966, pp. 468～474]

磁性薄膜記憶装置で，センス線に誘起された信号電圧はセンス線と直列になっており，正の半分と負の半分が互に逆の方向に伝ぱんする．この信号はセンス線の伝ぱん定数にしたがって減衰し，位相歪を生ずる．それと同時に語線との結合容量による波形の劣化も考慮せねばならない．大容量記憶装置ではセンス線は幾本かに縦続に接続される．もしその間を接続している線路のインピーダンスがセンス線のインピーダンスと異なれば，そこで反射が起り波形が劣化する．これらを考慮した場合に，センス増幅器の入力端に現われる信号波形がどのように表わされるかを求めている．平衡形ストリップ線を考え，センス端は整合しており他端は短絡である場合センス増幅器の入力波形はラプラス変換法を用いると次式のようになる．

$$V_{(p)}=\frac{Vs_{(p)}}{2}e^{-rd}\{1+e^{-r2(D-d)}\}$$

r としては，線路の G を無視し表皮効果を考慮すると

$$r=p\sqrt{LC}+\frac{1}{2}\sqrt{\frac{C}{L}}Rdc+\frac{1}{4}\sqrt{\frac{C}{L}}(Rdc_1\eta_1+Rdc_2\eta_2)$$
$$=p/u+\alpha+\Delta\beta$$

語線との結合1段当たりの透過係数は

$$T=a/(p+a):\ a=2/z_0c_0$$

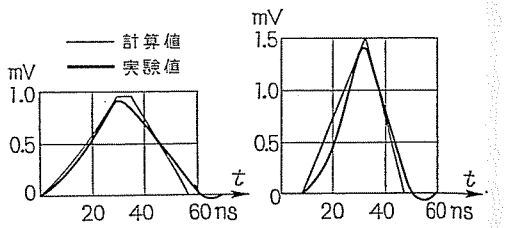
センス線間を接続している線路 z_1 の両端での透過係数

$$T'=8z_0z_1(z_1^2+z_0^2)/(z_1+z_0)^4$$

最終的な式は

$$V_{(p)}=\frac{Vs_{(p)}}{2}[e^{-\frac{pd}{u}}\cdot e^{-\alpha d}\cdot e^{-\Delta\beta d}\cdot T^n\cdot T'^m$$
$$\{1+e^{-\frac{p}{u}2(D-d)}\cdot e^{-2\alpha(D-d)}\cdot e^{-2\Delta\beta(D-d)}$$
$$\cdot T^{2(N-n)}\cdot T'^{2(M-m)}\}]$$

となる．n, N, m, M は距離 d 中の不連続数および総不連続数である．



（a）近端からの信号　　（b）遠端からの信号

$Vs_{(p)}$ として三角波を入れた場合の計算値と実験値とを示す．語数 512，センス線の長さ 40 インチ．

（大表良一）

D-26. 連想記憶装置研究の概観

A. G. Hanlon: Content Addressable and Associative Memory System, A Survey [IEEE Trans. EC, EC-15, No. 4, Aug., 1966, pp. 509～521]

序論において本論文の目的は non-technical に，すなわち個々のシステムの金物の詳細などに立入らずに研究の動向を概観し，将来何に重点を置くべきかを見極める基礎を与えることであると述べている．文末には125項目に及ぶ文献表があり，文中でその多くを引用して連想記憶装置に関する種々の問題の歴史的発展過程，現状，技術的問題点，将来における希望的観測，批判的な観点などをまとめてある．また連想記憶装置の有用性，応用可能分野に関してもかなりの紙数を費して概観を試みている．

序論では Catalog memory, Associative memory, Content addressed memory, Data addressed memory 等々の言葉があいまいな定義のままに乱用されているが，Content addressable memory が妥当であろうと述べている．

第2節は連想記憶装置の形態に関して述べており，

普通の記憶装置の延長として用いる場合，かなり大容量で primary store として用いる場合，および処理装置自身が associative な機能を有する場合の3段階があるとし，それぞれ例を上げ短かい文章で内容にふれている．

第3節は関連する金物に関する概観のあとで文献を引用して素子の金物の特性のうち，何が重要かについて述べている．各ビット自身を全 IC 化する傾向にはふれていない．

第4節は連想記憶読み出しの基本的な技術にふれ，また現在までに連想記憶装置の機能として考慮されて来た種々の論理機能を挙げている．

第5節は速度，値段および大きさに関する概観であり，速度に関する種々の観点と現状，値段の具体的な例，大きさに伴う種々の問題，実例などを述べている．ソフトウェアによる連想記憶のシミュレーション並びに大容量の場合のクライオトロンの有利な点などの記述は興味深い．

第6節はソフトウェアとの関連で book keeping が不要になるので，たとえば番地割りつけを記録しておく必要がなくなるなどの点を指摘．

第7節では応用の可能性のある分野を数十項目列記している．

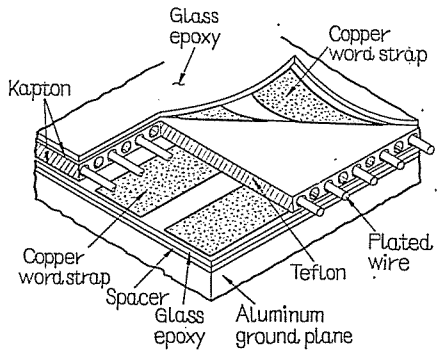
第8節では連想記憶装置の利点と欠点に関して述べているが，次のごとく文を結んでいる．連想記憶装置の利点はデータ処理が速くなり，ソフトウェアが簡単になり，また batch fabrication に向いたものになりつつあるということであるが，文献で積極的に不利な点に触れたものはない．（佐々木彬夫）

D-27. 500 ns 磁性線記憶装置

J.P. McCallister and C.F. Chong: A 500-nanosecond Main Computer Memory Utilizing Plated-Wire Elements [Proc. FJCC, 1966, pp. 305～314]

記憶素子として電着磁性線を用いた磁性線記憶装置は，高速無作為接近形電子計算機用記憶装置として有望視されているが，ここでは UNIVAC 計算機の主記憶装置として，開発したサイクル時間 500 nsec. 容量 16K 語（1語9ビット）の磁性線記憶装置の設計ならびに試験結果について報告している．

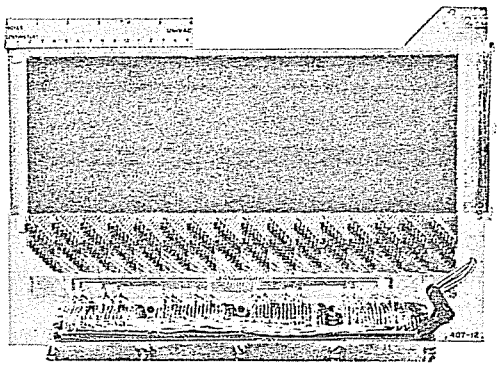
第1図が磁性線記憶平面の構成図であり，まずテフロンが塗布された2枚の Kapton (du Pont Co. の製品) の間に直径 0.2 mm のパイロット線をおいて加熱圧縮してケーブルを製作し，ついでアルミニウム



第 1 図

基板，絶縁板，2枚の駆動線板をラミネートし，つづいてパイロット線を引抜いて代わりに直径 0.13 mm の電着磁性線を挿入すれば記憶平面ができあがる．駆動線の寸法は 1.02 mm×38 μ であり，厚さ 0.28 mm のエポキシガラス板に蒸着でつくられている．磁性線としては厚さ 10,000 Å の鉄ニッケル合金が電着された円周方向容易軸磁性線が用いられ，立上り時間 40 ns，振幅 800 mA の駆動電流に対するスイッチング時間は約 80 nsec，出力電圧は 5～10 mV，また情報電流は約 40 mA である．

第2図の記憶平面実装図に示すように，駆動選択用



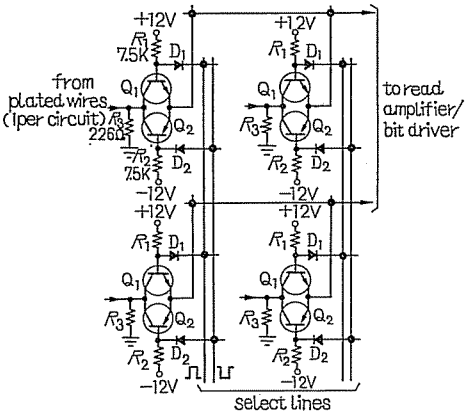
第 2 図

のダイオードを含めて約 15 cm×10 cm の記憶平面に 36K ビット（9×16 ビット，256 語）を収容することができる．

1本の駆動線を駆動選択することによって16語が，並列に読み出されるため読取信号を増幅整形したのちに 1/16 の番地選択を行なっているが，非破壊読み出し（NDRO）動作を行なわせているため，その都度再書き込みを行なう必要がなく装置の価格はほとんど上

昇しない．

読み取線あるいは情報線の選択回路図を第3図に示



第　3　図

す．ビット間相互の干渉を防ぐため情報電流として両極性パルスを用いて，ビット密度をあげている．

論理回路としては DTL 形の NAND 回路を採用し，最悪条件における遅延時間は $T_{on}=10$ ns，$T_{off}=14$ ns である．

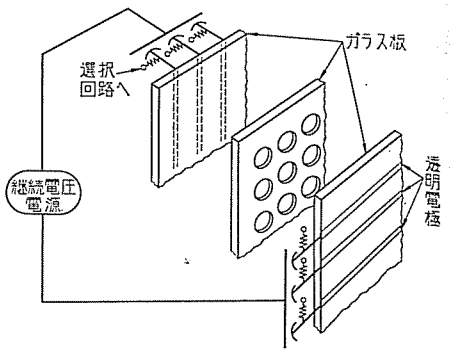
アクセス時間 300 ns，サイクル時間 500 ns で動作試験を行なった結果，安定に動作し，また小容量電子計算機の主記憶装置として数ヵ月間良好に動作している．　（伊藤陽之助）

D-28. プラズマ表示パネル-記憶作用を有しディジタルに座標指定ができる放電型図形表示盤

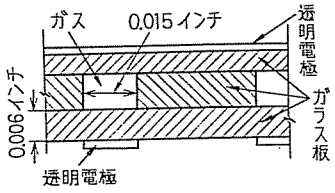
D.L. Bitzer and H.G. Slottow: The Plasma Display Panel-A Digitally Addressable Display with Inherant Memory. 〔Proc. FJCC, 1966, pp. 541～547〕

計算機システムにおいて図形データの入出力装置として，ブラウン管が広く用いられているが，その欠点として，表示面自身に記憶作用がない，座標指定にアナログな量をもちいる，高電圧が必要である，形が大きいなどが考えられる．これらを補うものとして，新しく提案されたプラズマ表示パネルは，その像を保持する作用があり，計算機からのディジタル信号で直接座標の指定ができる．また分解能もブラウン管と同程度のものが可能であり，さらにライトペンで同形を直接かくことができ，それを計算機によみこませることができる．

構造および標準的な寸法を第1図および第2図に示す．中央のガラス板にはマトリックス状に小さな穴が



第1図　プラズマ表示パネル



第2図　ガス放電セル

あり，両側のガラス板には，それぞれ外側に，互に直交するように，透明な電極が帯状に蒸着され，その交点がちょうど中央のガラス板の穴の位置に一致するようになっている．これら三枚のガラス板を密着し，間隙から空気をぬいて，適当なガスをつめてある．中央ガラス板の穴がそれぞれ放電セルを形成している．

両電極間に交流電圧を加え，セルの両端電圧が，放電開始電圧を越えると，放電が起こりすぐにグロー放電に成長する．同時にガラス壁面への電荷の移動がはじまり，セルの両端の電圧は下がる．電極間の電圧は次第に低くなるので，グローは滅り，放電が停止する．この放電の過程は測定の結果 50～75～10^9 sec の間に起っている．次の半サイクルでは，電圧が逆転するので，壁電荷による電圧が足し合わさり，より低い電圧でも放電が起こることになる．つまり適当な電圧を加えることによってこのセルは，on-off 二つの安定状態をもった α 値素子となる．記憶作用はもちろん壁電荷によるものである．セルの放電状態を保接するには，このように交流電圧ばかりではなく直流成分のないパルス電圧でもよい．

あるセルの状態を変化させるためには，交叉している電極に，適当なパルス電圧を加えればよい．他のセ

ルには半分の影響しかなく状態は変化しない．この制御電圧を同時にいくつかのセルに加えることにより，矩形状の領域を同時に on-off することができる．また状態を変えるのに，早いパルスばかりでなく，ゆっくり変化する信号でも行なうことができる．これは高インピーダンスの制御回路が使えるという利点がある．

セルの状態をしらべるには，指定したセルだけに電圧を加え，発生する光をパネル全体をみわたす検出器で検出する．壁電荷はミリsec程度とどまっているので，次のサイクルで全体に維持電圧を与えれば他のセルが消えることはない．維持電圧にパルスを用いる場合には，間の時に読み出しをおこなうこともできる．

もし放電開始電圧以上にしても，放電をおさえることができれば，ライトペンの光で，壁からの光電子を発生させ，放電を開始さすことができるはずである．これによって図形を直接かくことができるわけであるが，セル間の光学的なしゃへいの問題などいろいろむずかしい点もある．一つのセルについての実験はうまくいっている．

このプラズマ表示パネルの開発の動機は，教育用計算機システムへの応用である．

以下，現在 PLATO システムにもちいられているブラウン管による表示装置に比較して，スピード，分解能，経済性について述べている．プラズマ表示パネルの本質的な利点は記憶作用があることである．

（相馬 嵩）

F-29. 汎用データ処理システム（ADAM）

T.L. Connors: ADAM–A Generalized Data Management System [Proc. SJCC, 1966, pp. 193~203]

ADAM は，データ処理システムの設計と評価を行なうための補助的手段として作られたプログラムシステムである．ADAM は，システムをシミュレートするために用いられ，大きなシステムの設計やプログラムの完成，比較，評価するためにも用いられる．

データ処理システムを特徴づける機能を一般化して次のようにまとめてあり，モデルの特性はデータとしてファイルされる．

（1） ファイル作成と保守

（2） 捜索とファイル処理メッセージの解釈および処理

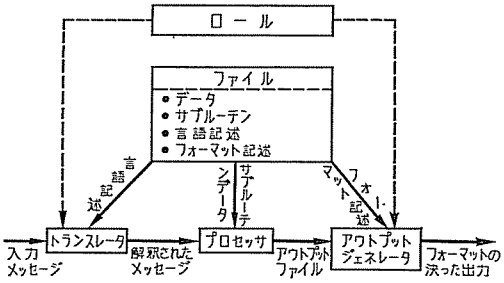
（3） オンラインおよびオフラインの I/O

（4） 報告作成と形式整正

（5） サブルーチンの編集と実行

（6） 計算機の各機器のダイナミックアロケーション

ADAM は本来ファイル処理システムである．入力メッセージを解釈して必要な処理を行ない．適当な出力を出す．これに全てファイルが関係するのである．メッセージ処理の仕方は第1図に示されている．おも



第1図 基本実行サイクル

なメッセージ処理のプログラムは，トランスレータ，プロセッサ，アウトプットジェネレータである．メッセージが入ってくると，トランスレータはそれを解釈して処理表にまとめる．プロッセッサは処理表に規定されているステップを実行する．アウトプットはファイルの形に作られる．

ファイルは一連のエントリーとして構成され，エントリーは特性値によって特徴づけられている．実際のデータは特性値として与えられる．

ロールは内部名称と外部名称を関係づけるエレメントの集りとして構成され，付属的な値がエレメントにつけられている．

対称のシステムを，モデル化しようとするとき，ADAM 言語ファイル中の言語を用いても，新しく言語を作って用いてもよい．オンラインオペレーション中にインプット言語の意味を，ストリングサブルーチンで変えることができる．

このシステムは IBM 7030 にプログラムされているが，その機器構成，システムコントロール，コアアロケーション，ファイルアロケーションなどについて言及している．（木村幸男）

F-30. 計算機組織における多重性

I. Flores: Multiplicity in Computer Systems [Computers and Automation, Vol. 15, No. 7, July,

1966, pp. 19～23]

マルチプロセシング，マルチプログラミング，マルチコンピューティング，マルチユーザーズ等々の最近のトピックスの平易な解説である．

マルチプログラミングについては，入出力動作の同時動作のような初等的段階から，多くの段階を経て，マルチプライオリティ機能をもった，もっとも一般的なマルチプログラミングにまで，発展してゆくことが説明されている．

マルチ・オペレータ・システム（いわゆる MAC）については，phylosophy と称して，次の四つのものを解説している．

- to-Completion philosophy
- round-robin philosophy
- use-priority philosophy
- need-priority philosophy

この他，マルチプロセシングおよびマルチコンピューターズについても簡単に解説している．

（村田賢一）

F-31. ソフトウェア・デバッグ作業改善のための論理設計——一つの提案

N. Chapin: Logical Design to Improve Software Debugging—a Proposal. [Computers and Automation, Vol. 15, No. 2, Feb., 1966, pp. 22～24]

本論文は，ソフトウェアのデバッグ作業を改善することを目的とした，プログラマの立場からの，ハードウェア設計者への提案であって，2種類の新しい命令がその内容である．

まず最初に，AMR (Auto-MonitoR command) は，2個のアドレスを有し，第1のアドレスは記憶装置のある番地，第2のアドレスはドラムディスクなどの番地を示す．この命令による作用は，後続の命令に関して，次のごときデータを，第1のアドレス指示された場所へ格納することである．そのデータの内容は，(1) 命令の有効アドレス，(2) 有効命令 (Execnte Remote Instruction のごとき場合)，(3) 命令実行後のオペランドの値（その1部分でもよい），(4) その命令で変更された語，文字，数字などの数，(5) インジケータ類の命令実行形の状態．

AMR の第2のアドレスは，記憶装置の，AMR 用バッファがいっぱいになった時に，ドラム，ディスクなどへダンプするのに使用される．

なお，実際には，AMR は，すぐ次の命令に作用するよりも，AMR が解読されて以後，ある一定の条件が発生した時に，そのとき実行された命令に対して作用するようにした方がよい．

次の SSD (Snap Shot Dump Command) は，AMR の変形で，AMR よりアドレスが一つ多い．

この第3のアドレスは，ダンプを取るべき場所（複数）を直接示し，この点が AMR と異なる．

以上のごときハードウェア機能があれば，システム・プログラマは，プログラマからデバッグを要求するリスト（シンボリック・ネームおよびデータを含む）を受け取り，コンパイラで作り出されたシンボル，テーブルの助けを借りて，たとえばプログラマが指定したプログラムの部分が，どのように働いて，その結果になったかをリストすることができよう．

（村田賢一）

F-32. 1-正規システムの決定問題の可解性

Stephen A, Cook: The Solvability of the Derivability Problem for One-Normal Systems [J ACM, Vol. 13, No. 2, Apr., 1966, pp. 223～225]

この論文ではタッグシステムのある拡張である1-正規システムの導出問題が可解であることが示されている．まず Post によって定義されたタッグシステムというのは組み合わせシステムの一種で，有限個の文字の集合 $\{S_1, \cdots, S_a\}$，プロダクションの集合 $\{T_i: S_i \to E_i\}$，正整数 β からなっている．ただし各 E_i は空または有限の長さをもつ語とする．ここで各プロダクション T_i は，長さ β 以上で文字 S_i からはじまる語 W にのみ適用される．たとえば $W=S_iS_{i_2}, \cdots\cdots S_{i\beta}W_1$ とするとプロダクション T_i は W を W_1E_i に変換する．

一方，導出問題とは，任意の語 P, Q に対し与えられたシステムのプロダクションを有限回用いることによって Q が P から導びかれるか否かを決定する問題である．タッグシステムについては，$\beta=2$ の場合に導出問題は非可解であることが Cocke, Minski および Wang によって示され，また $\beta=1$ の場合には可解であることが Wang によって証明されている．

この論文であつかわれる1-正規システムは，$\beta=1$ のタッグシステムの拡張（もちろん正規システムの特別な場合である）となっている．$\beta=1$ のタッグシステムではプロダクションは各文字 S_i に対し高々一つの E_i が定まっているものであったのに対し，1-正規システムでは，S_i に対しある正整数 n_i 個のプロダ

クション $T_{ij}: S_i \to E_{ij}$ が存在する．

ここで語 W の長さを $|W|$ で表わす．いま，$P_0 \to P_1 \to \cdots\cdots \to P_m$ を一つの導出の系列とする時，もし $m=|P_0|$ ならばこの系列は1ラウンドをなすと呼ぶ．すると，もし Q が P から導出されるならば，P から Q にいたる導出の系列で $(\alpha+1)^n+\sigma$ ラウンド以下のものが存在する $(n=|Q|)$ ことが証明され，したがって次の定理が成り立つ．

定理　各 1-正規システムに対する導出問題は可解である．　　（小野寛晰）

F-33. 自然言語処理

R.F. Simons: Natural Language Processing [Datamation, Vol. 12, No. 6, June, 1966, pp. 61～72]

自然言語処理の終極の目標は，高度に知識化された質疑応答系の発展である．これらは自然言語の質問を受け入れて解析し，図書から最も適した参考書を捜し出し参考にしながら，質問に答える文章を生成する．本論文はこのような研究が各所でいかなる立場で行なわれどのような問題に遭遇し，その解決にどんな手法がとられ得るか，についてまとめ上げたものである．

Vannevar Bush の夢といわれた Memex (1945) を始祖として，今日では多くの試みがなされている．著者の Protosynthex は百科辞典を整備している．'くもの巣はどれだけ強いか' ときれば，百科辞典にもとづいて長い答えが出て来て，それと共にその答えがどれだけ質問と関係があるかを，点数にして出す．Raphael (M.I.T) のは推論ができる．情報を提供する文を与えておいて質問する．すると簡単な推論を行なって答えを作り出す．たとえば，'人は手が2本ある．手には指が5本ある．男の子は人である．ハリーは男の子である'．と与えて，'ハリーは指が何本か' ときけば '答えは10である'．とでる．Dan Boblows (M.I.T) のは高校程度の算術がとける．Oettinger と Kuno のは質問文解析に重点がおかれている．Klein (Cornegie Institute of Technology) は自動解析機を用いて本を解析し文章製作プログラムへの入力として文を生成する．その他，機械翻訳，幾何の作図等々の試みがある．

自然言語処理の研究には自然言語の研究が密接な関係を持っている．言語研究は，構造に関するものと意味に関するものに大別され，構造の方面では最も有益なものに最近の変換文法がある．それは語が結合されて句になり，句がある形から他のものへと変換される機構を，だんだんと明らかにしつつある．しかしながら，意味の方面は最も低いレベルにおいてすら少しも知られていない．最近になって Forder & Katz が意味解析を取り扱い，また Quillian, Karen Sparch-Jones などが実験的にも寄与している．言語に関する問題はそのまま言語処理に関する問題でもある．具体的にいえば，意味の異なる度合を定義し，その度合によって意味を構造化し，それにもとづいて辞書を作るという辞書の問題，また文の理解のためには解析する必要があり構造上の問題，すなわち多義語の選択，綴字の修正，あいまい性などの問題がある．

今日，自然言語処理はいくつかの問題をかかえているが，言語に関する研究の進展と相まって，将来急速に進むであろうと思われる．　　（五十嵐　実子）