文　献　紹　介

70-28　システムデッドロックの防止法

A. N. Habermann: Prevention of System Deadlocks [CACM 12-7 (1969), pp. 373-385] key: Multiprogramming, TSS, Resoure allocation, deodlock, interlock

OS 設計のリソースわりあての際に問題となることのひとつにデッドロックがある．たとえば，プロセス i, j がリソース A, B をそれぞれ占有 (seize) したあとで，プロセス i がリソース B に，j が A に要求をだすとすると，どちらのプロセスも相手のプロセスがリソースの解除 (release) をするのを待って動きがとれなくなる．このようなデッドロックを，リソースわりあてのときに防止する方法が述べられている．

プロセスとリソースに，それぞれ $1\cdots n$ および $1\cdots m$ と番号をふる．リソース i の個数を a_i とし，ベクトル $\vec{a}=\begin{pmatrix}a_1\\ \vdots\\ a_m\end{pmatrix}$ でわりつけられるクリースを表わす．プロセス k が一時にリソース i を必要とする最大個数を b_{ik} とし，ベクトル $\vec{b_k}=\begin{pmatrix}b_{1k}\\ \vdots\\ b_{mk}\end{pmatrix}$，行列 $\boldsymbol{B}=(\vec{b_1}\cdots\vec{b_n})$ によってリソースへの要求を表わす．プロセス k にわりあてられた（そのプロセスにわりあてられたリソースを通じて，間接的にわりあてられたとみなされるものも含む）リソース i の個数を c_{ik} とし，ベクトル $\vec{c_k}=\begin{pmatrix}c_{1k}\\ \vdots\\ c_{mk}\end{pmatrix}$，行列 $\boldsymbol{C}=(\vec{c_1}\cdots\vec{c_n})$ でわりあて状態を表わす．$\vec{b_k}$ と $\vec{c_k}$ の組がプロセス k のわりあて状態を記述している．

ベクトルや行列の $<$ や $\leqq$ は，すべての要素にその不等号がなりたつときのみ用いることにすると，つぎの3つの式が成立する．

$$① \forall k: \vec{b_k}\leqq\vec{a},\quad ② \boldsymbol{C}\leqq\boldsymbol{B},\quad ③ \sum_{k=1}^{m}\vec{c_k}\leqq\vec{a}$$

時刻 t での $(\vec{a}-\sum_{k=1}^{n}\vec{c_k})$ を $\vec{r}(t)$ とおく．(③は $\vec{r}(t)\geqq\vec{0}$ となる)

デッドロックに対して安全な状態は次式で表わされる．　④ $\forall k\in S: \vec{b_k}\geqq r(t)+\sum_{s(e)\leqq s(k)}\vec{c_e}(t)$

S は n 個のプロセスの系列で，$s(k)$ はその系列中での k の順番を示す．④は少なくともひとつの系列があって，残っているリソースの範囲内で，それぞれの仕事を完成できることを保証している．

$(\vec{a}\boldsymbol{BC}(t))$ である状態が安全かどうかは，④を満たすかどうか調べればよい．これは $n!$ の順列を試みることになるが，実はすべてのプロセスの系列 S の代わりに，④を満たす部分系列をみつけるだけでよいのである．

また，$(\vec{a}\boldsymbol{BC}(t))$ で，プロセス k が新たにリソースを要求したときに，これを与えるかどうかの決定は，与えた場合での状態が安全かどうかをみてきめればよい．このアルゴリズムは“THE”マルチプログラミングシステムで実用になっている．　（有沢　誠）

70-29　アルゴルを基礎にした連想言語

An Algol-Based Associative Language

J. A. Feldman & P. D. Rovner: [CACM 12-8 (1969), pp. 439-449] key: ALGOL, associative, programming language, data structure

複雑に関連しあったデータ構造をとりあつかうための高級プログラミング言語 (LEAP) が設計され，作られた．もとになるデータ構造は，hash-coding technique を用いてある．LEAP は，リフトウェアによって associative processing を実現するための言語であり，データ構造の部分をのぞいて，トランスレータ・ライティング・システム VITAL を用いて作られている．

LEAP syntax には3つの構成要素がある．それらは，1) reserved word (例: set)，2) nonterminal symbol (例: ⟨set expression⟩)，3) identifier (例: fifth) である．ALGOL の拡張として，つぎの言葉を新たに定義する．それらは，item, set, itemvar, local, association, form, component, bound, datum である．

3つのエレメントが1つの associative cell を構成し，それを (a, o, v) あるいは $a\cdot o\equiv v$ と表わす．この各エレメントはいずれも item 全体の集合からとられ，この 3-tuple を association という．association を構成する item は，その component である．たとえば

father・john doe≡don doe

end・line≡point

form は，a, o, b のうち1つまたは2つを変数としたものをいう．たとえば $a・o≡x$, $a・x≡v$, $x・o≡z$ などである．association と form の表現および操作が LEAP およびデータ構造の発展のおもな動機となっている．

新たにつけ加えられたデータ・タイプとして，item, itemvar, local, set がある．association は，make statement により item から作られるが，実際には，dictionary plrase ですべての item は，internal name に変換され，association はすべて internal name を使って作られる．

最も特徴的なステートメントは foreach ステートメントである．これは，前に述べた form の操作である．たとえば

foreach x in sons do make father・x≡john

association の内部表現は，この form の操作が簡単にできるように考えられた．たとえば (son, don, x) に対しては，son, don の internal name を 17,453, 21,411 とすると，son の上2けたでページを示し，ページ 17 上では，453 と 21,411 を hash して，たとえば，4530+2140=6670 というアドレスを得る．このページの1つの cell は，つぎの形をしている．

O	type	conflict list
A-use		value list

このページは，son というアトリビュートを持つすべての association が，A-use list でつながっている．ゆえに，このページを A-page という．システムは，この A-page の他に O-page, V-page を持つ．

LEAP を使った代表的な応用例としては，集積回路の設計とそのマスクの layout を求めるインタラクティブなプログラムとか，グラフィック・ディスプレイ・システムでの，2次元プログラミング・ランゲージ Ambit/G などがある．（古川康一）

70-30 簡単な曖昧な自由言語について

H. A. Mauer: A Direct Proof of Inhevent Ambiguity of A Simple Context-Free Language [JACM, April, 1969] key: inherent ambiguity, unambiguous grammar, contextfree language, production system, reduced grammar

自由言語 (context-free language) $L=\{a^ib^jc^k|i=j \text{ or } j=k\}$ が曖昧な言語 (inherently ambiguous language) であることは，[GS] (The Mathematical Theory of Context-Free Languages) などにも述べられているが，その証明はきわめて複雑なものである．本論文は，線形集合 (linear set)，有界言語 (bounded language) などの持つ特質をつかうことなく，その一証明を与えたものである．

普通に定義されている自由文法 $G=(V_N, V_T, R, A)$ が既約 (reduced)，かつ，無曖昧 (unambiguous) であれば，長さ1以上の生成過程 (derivation) に対して，$A\overset{*}{\Rightarrow}A$ とはなりえぬことを最初の補題として示す．

ここで，$P(G)=\{B\in V_N|B\overset{*}{\Rightarrow}xB_y, xy\neq\epsilon\}$ とし，自由文法 G が，(i)，(ii)，(iii) の条件を満たすとき，G は概周期 (almost looping) と定義する．(i) G は既約である．(ii) $V_N-A\in P(G)$, (iii) $A\in P(G)$ であるが，またはすべての $x\in L$ に対して，その生成過程に A が一度しかあらわれない．

つぎの補題は，任意の無曖昧な既約文法 G' に対して，$L(G)=L(G')$ である無曖昧な概周期文法 G が存在することを述べるものである．これは条件を満たす具体的な文法列 $G'=G_0, G_1, G_2, \cdots, G_K=G$〔$K=\sharp(V_N)-1$〕が存在することにより証明される．

また，概周期文法 $G=(V_N, V_T, R, A)$ が $L=\{a^ib^jc^k|i=j \text{ or } j=k\}$ を生成すれば，G の変数が L の語と関連してある種の特質を持つことがわかる．そして，最後に定理として，L が曖昧な自由言語であることが論じられる．ここで，L を生成する概周期文法が $a^{2^p}b^{2^p}c^{2^p}\in L$ に対して，2とおり以上の生成過程を持つことが示され，証明が終わる次第である．

周知のとおり，L に対するこの種の直接的証明は [GS] の open problem であり，著者は概周期という概念を設定することによって，それに答えたわけである．（山下 元）

70-31 ファン・インを限定された NAND ネット・ワークの設計自動化

Donald L. Dietmeter & Yueh-Hsung Su: Logic Design Automation of Fan-In Limited NAND Networks [IEEE Trans. on Computers, Jan. 1969, Vol. C-18, No. 1 pp. 11-22] key: factoring fan-in limit, logic design automation, NAND networks, synthesis algorithms

factoring テクニックを用いて，ファン・インを限定された NAND スイッチング・ネットワークを構成

するアルゴリズムを提案している．もちろん，論理設計自動化の一環として，計算機向きに開発されたものであるが，小さな問題には人手でも十分に実行可能である．このアルゴリズムにより構成された樹状ネット・ワークでは，ゲートの数，論理のレベルが少なくなっている．

ファン・イン問題を解決するために，I 本の入力端子を持つ NAND ゲートで，ξ 本のファン・インを持つ NAND ゲートを実現するとき必要になる NAND ゲートの数 $g(\xi)$ は

$$g(\xi)=\begin{cases}0 & \xi=1 \text{ のとき}\\ 1 & 1<\xi\leq I \text{ のとき}\\ 1+2\left\lfloor\dfrac{\xi-I}{I-1}\right\rfloor & \xi>I \text{ のとき}\end{cases}$$

であると定義している．ここで，$\lfloor x\rfloor$ は x に等しいか，あるいは大きい最小の非負の整数を表わしている．これを用いて factoring によって必要になるゲートの数を見積り，それが最小になるように3つの基本構成パターンのうちの1つを選択するようになっている．

このアルゴリズムへの入力論理は，無冗長主項の論理和形式であることを前提にして議論がなされている．この制限により，factoring によって節約されるゲートの数の度合を示す figure-of-merit の性質が定量化できるので，これを最大にする factor を発見するアルゴリズムを確立し，これをもとにして単出力論理関数を構成するという手法をとっている．

Appendix には，このアルゴリズムを FORTRAN で書いたプログラムを掲げている．また，このアルゴリズムをより精巧なものにするためのテクニックについても言及している．　(宇都宮公訓)

70-32 キノフォーム——新しい Wavefront Reconstruction Device——

L. B. Lesem, P. M. Hirsch & J. A. Jordan: The kinoform: A new Wavefront Reconstruction Device [IBM J. Res. and Dev. Vol. 13, No. 2, 1969, pp. 150–155] key: Holography, Three dimensional display Read only storage

Kinoform は計算機による新しい Wavefront reconstruction device であり，hologram と同様に3次元の像を再生することができる．Hologram と異なる点は，1次回折像だけを再生し，再生の際の illuminate 光は，すべてこの1つの像を結ぶのに作用するために，充分明るい像が得られるなどがあげられる．また，Kinoform の作成に対しては，digital hologram の場合に比較して，reference beam や image separation のための計算が不必要になるため，time scale が短くてすむ．

physical basis (A) Amplitude hologram: 一般に Scattered wavefront は

$$W(x,y,z)=A(x,y,z)\exp\{i\phi(x,y,z)\} \qquad (1)$$

の形で表わされる．この W と Reference beam W_0 の干渉により，

$$I=|W+W_0|^2$$
$$=|W|^2+|W_0|^2+W^*W_0+WW_0^*$$

に比例した Pattern が hologram として記録される．したがって，これに Reference beam を照射すると，W 以外にその conjugate wavefront W^*，および Central beam $|W|^2+|W_0|^2$ が同時に Rewnstruct されてしまうことになる．また，image の明るさをあげるために bleaching を行ない，Amplitude hologram を Phase hologram に変換している．

(B) Kinoform: Kinoform は，もっぱら計算機と Bleaching 技術を用いて作られ，hologram の場合と異なり完全に光学的技術だけでは作成が困難である．Kinoform は入射光の位相成分だけに作用する．すなわち，Scattered wavefront の位相情報だけが，Scattering object の像を作るのに使われ，Wavefront の amplitude は一定であると仮定している．その際の phase-shaping の方法として，次式で示される transmission grating (一次元的に示してある) を最終的に Kinoform の表面に作り，入射光の phase modulation を行なっている．

$$d(x)=d_0+ax(\mathrm{mod}\ d_{\max}) \qquad (2)$$

ただし，$d_{\max}$ は入射光の phase を 2π だけおくれさせるための medium の厚さである．

そこで，まず

$$T_{ijk}=t_{ijk}\exp(i\alpha_{ijk}) \qquad (3)$$

を Object 上の $(i,j,k,\cdots x,y,z$ に対応) 番目の点の reflectance とする．t_{ijk} は object point (i,j,k) の明るさに相当し，α_{ijk} は object point (i,j,k) の diffuse scattering による位相を表わし，この場合には a random に変化するものとする．いま，1次元的に考えると，Object の一点 j から出た光の Wavefront に Kirchhoff diffraction theory により，近軸光の場合には Kinoform 面上で

$$W_j(x)=T_j\exp\left\{\frac{i\pi(x-\alpha_j)^2}{\lambda z_0}\right\} \qquad (4)$$

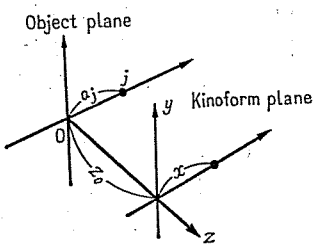

図 1

となる (図1参照).

したがって, Resultant Wavefront は $k=\pi/\lambda z_0$ とおくと

$$W(x)=\sum_j W_j(x)$$

$$=\exp(ikx^2)\sum_j T_j\exp(ika_j^2)$$

$$\times\exp\frac{-2\pi ixa_j}{\lambda z_0} \qquad (5)$$

そこで, $U_j\equiv T_j\exp(ika_j^2)$ とおくと

$$W(x)=\overline{U}(x)\exp(ikx^2) \qquad (6)$$

となり, $\overline{U}(x)$ は U_j の Fourier 変換となる.

このようにして計算された $W(x)$ を Modulo 2π で変換し, plotter を用いて図化する. これを本論文では 1/100 程度に Photoreduction をしてから, Bleaching を行なっている. 再生に際しては phase missmatching の問題がある. これは phase $\phi=0$ の点に入射した光が, $\phi=2\pi$ の点に入射した光と相対的に1波長おくれなくてはならないのであるが, 種々の条件により, これがくずれるために生ずる trouble である. Phase missmatching が生ずると再生象に real 以外にその conjugate image が重なったり, 中央に bright spot が生じたりするので, kinoform 本来の特徴が失なわれることになる. (保原 信)

70-33 ランダム・アクセス・プログラム内蔵式機械, プログラム言語の一方法

C. C. Elgot & A. Robinson : Random-Access Stored-Program Machine, an Approach to Programming Languages [J. ACM, Vol. 11, No. 4 (October, 1964), pp. 365-399] key: programming language, semantics

ディジタル計算機に対するプログラミングのために, 多くのプログラム言語が開発された. それらは大ざっぱにいうと, 問題向き言語と機械語の2つに分類される. この論文の目的は, それらの言語, 言語の機能, 言語の間の関係, および言語の拡張と改良の方法に関する理論的考察のための, 基礎を与えることである.

そのために, 現在のディジタル計算機の数学的モデルとして, ランダム・アクセス・プログラム (RASP) を導入する. RASP はプログラム内蔵式機械とデータを記憶装置に内蔵し, 記憶装置のいかなる場所にも, 直ちにアクセスできるという, ディジタル計算機の中央処理装置の特徴を持っている.

RASP の "状態" は記憶装置 (語の無限集合) の configulation k と制御が置かれている番地 a との対 $\langle k, a\rangle$ である. $\langle k, a\rangle$ を他の状態 $\langle k', a'\rangle$ に変える写像を RASP の命令と呼ぶ. 状態の変換を k, a および有限個の番地と, その内容によって決定する命令だけを持つ RASP (有限決定性 RASP) について, つぎの2つの性質が証明されている.

1) 単純な3つの命令を持つ RASP によって, すべての部分帰納的関数を計算できる.

2) 自己変更可能なプログラム (たとえば, index register などによって) は, そうでないプログラムよりも, より多くの sequential function を計算できる.

RASP の記憶装置内に語として記憶されたプログラム言語の構成要素の意味は, その語が RASP の命令として実行されたときに, 状態をどのように変換するかということである.

機械語から問題向き言語への拡張は, より多くの命令を持つ RASP を作ることに対応する.

コンパイラの作成は, 拡張された RASP のプログラムを同じ仕事をするもとの RASP プログラムに翻訳するアルゴリズムの作成であるということができる. (二村良彦)

70-34 プログラム言語の意味とコンパイラの理論について

E. K. Blum : Towards a theory of semantics and compilers for programming languages. [J. of Computer and System Sciences, Vol. 3, No. 3 (Aug. 1969), pp. 248-275] key: compiler, programming language, semantics

プログラム・システムをつぎのような対 (L, Φ) と定義する. ただし, L は有限アルファベット上の, プログラムと呼ばれる語の帰納的集合, そして Φ は L か

ら部分的に計算可能な関数の集合εへの写像とする．Lをプログラム言語，そしてΦを意味演算子(semantic operator)と呼ぶ．

$p=(L,\Phi)$ と $p'=(L',\Phi')$ を2つのプログラム・システムとする．このとき，LからL'の中への写像Γをコンパイラと呼ぶ．すべての $p\in L$ に対して

$$\Phi'[\Gamma(p)]=\Phi(p)$$

が成立するならば，Γは“正しい”という．ΦとΦ'を帰納的に定義すれば，Γの正しさの証明が可能となる．

上の議論を曖昧さなく行なうために，この論文ではLを整合論理式(well-formed formulas)の集合，そしてΦをその誘導規則(derivation rule)とする形式的体系(formal system)にプログラムシステム(L,Φ)を埋蔵する．

その概念を明確にするためにこの論文の大部分は，Gödel-Herbrand-Kleeneによる帰納的関数の形式的体系およびTuring機械の体系に埋蔵された2つのプログラム・システムを，おのおの p_R および p_T とし，p_R およびその部分体系の構造，p_R から p_T へのコンパイラおよびその正しさについて述べている．

(二村良彦)

70-35　マクロプリプロセッサ SYMPLE システム

J. E. Vander Mey, R. C. Varney & R. E. Patchen: SYMPLE——A general syntax directed macro preprocessor [FJCC 1969, pp. 157–167] key: macro, extensible languages, metalangnages, contextfree languages

SYMPLE マクロプリプロセッサは，高級レベル言語にうめこむマクロを処理するシステムである．マクロがうめこまれる言語の文法と，マクロがはいる文脈の指定とを，BNFを拡張したメタ言語で記述する．このメタ言語にはつぎのような特徴がある．

① かっこを用いて，いくつかの概念をまとめて記述することにより，本質的でないノンターミナル記号を減らすことができる．

② 記号系列や文字の数の上限と下限の指定ができる．

③ 存在禁止の記号 (¬) を用いて，簡略化をはかっている．

④ 文法を読みながら対比するときに行なう走査(scan)の際のポインタの移動指定ができる．

このメタ言語によって文法が記述される言語の中には，CFLでない $\{0^n1^nO^n|n\geqq1\}$ のようなθが含まれている．

マクロの定義は，マクロがつめこまれるときの形式の指定をするテンプレイト指定と，マクロを用いていく手順の指定をするセマンティクス指定とからなる．

マクロのうめこみの認識は，もとの言語の中の正しい文脈かどうか，正しい形式でマクロ指定がされているかの2層にわかれる．

セマンティクス指定の大部分は文字系列処理であって，各種のそう入命令を中心とする実行命令が並んでいる．とくに，この中にマクロ定義自身を変える命令が含まれていることが注目される．

システム全体の機能の流れは図2のようになる．

(有沢　誠)

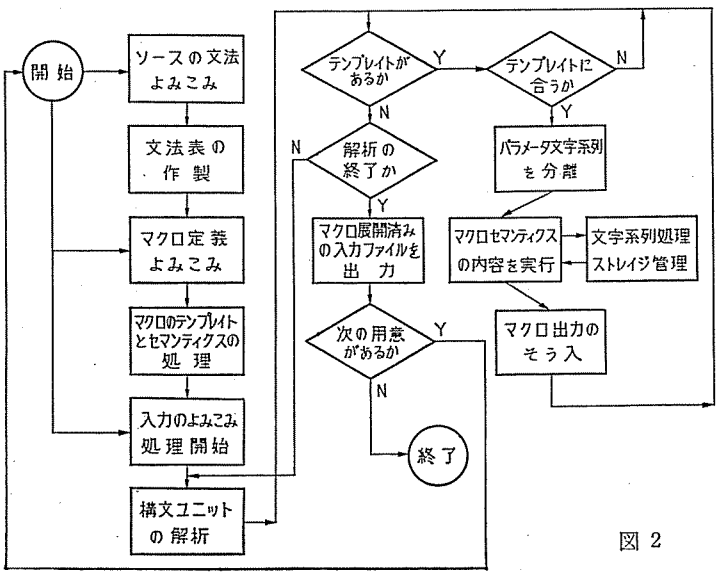

図 2

ニュース

図形処理に関する理研シンポジウム

「電子計算機による図形処理」をテーマとするシンポジウムが，去る1月28日，理化学研究所（埼玉県大和町）で，同所情報科学研究室の主催のもとに開かれた．このシンポジウムでは，高精度ブラウン管の開発（理研・東大），泡箱写真の解析（東大・東北大・名大），図形処理のデータ構造とリスト処理言語（東大）などに関する10件の研究発表があり，また統計力学における秩序無秩序現象のシミュレーションを，ディスプレイ（NHKのIBM 2250）によって視覚化したカラー映画（理研）も上映された．

当日は，情報科学研究室で開発された二重偏向方式の高分解能・高精度ブラウン管，移動レンズ方式の高分解能ブラウン管，光点移動制御用のボール・コントロールなども公開された．このうち，二重偏向管（第1次試作品）は，5インチの電磁偏向管で，スポット径は40ミクロン，位置ぎめ精度は $\pm 2^{-12}$ (2,000×2,000点に相当）のものが得られている．また，移動レンズ管の試作品は，2インチのCRTで，スポット径は800ミクロン，走査線500本の解像力があり，実際にテレビ画像を2インチの画面上に出したとき，各走査線が識別できる．

ボール・コントロールは，金属のボールを手の平でまわすことにより，CRTの光点を二次元的に移動させるための装置で，光点の位置がきまったところでボタンを押すと，その X, Y 座標が紙テープにパンチされるようになっており，放電箱写真の原子核反応の解析への応用が予定されている．

国立大学に情報関係の新設学科

1970年度政府予算で認められたとおり，時代の要請にこたえて，五つの国立大学は下記のような情報関係学科が新設され，新人生を迎えて，各学科ともこの4月から発足する．

電気通信大学電子計算機学科（定員60名）
山梨大学計算機科学科（60名）
東京工業大学情報科学科（40名）
京都大学情報工学科（40名）
大阪大学情報工学科（40名）

これら諸学科のうち，定員60名のところは専門5講座（教授5，助教授10），また，40名のところは4講座（教授4，助教授8）となり，これら教官要員は，これから約3年間にわたって募集される．

これらの学科では，いずれも電子計算機（ハードウェア，ソフトウェア）の教育・研究を中心として，将来の情報化社会の中核となる人材の育成を目的としている．しかし，技能的な色彩のこい分野だけに，単なる技術者の養成をこえて，たとえば，現在職人芸的なプログラム技術を，「電子計算機学」といった学問として確立させるのに，こうした学科がどの程度貢献できるかは大いに注目される．

また，学生の実習・演習・実験に不可欠な教育用電子計算機の導入は，明年度から始まるが，これを契機として，教育用計算機の開発もうながされよう．各大学では，BASICのような教育用コンパイラー言語，多くの端末がついた教育用準TSS，マークセンス形入力装置などの導入が検討されている．

LSIメモリ実用化へ

アメリカの半導体メーカINTEL社（国内ではパネトロン社扱い）は，このほどPチャネルMOS（金属酸化膜）形LISを使った大容量ICメモリ（1101形，11011形）を発売した．これは大きさ4 mm×21 mm×8 mmの16ピン・デュアル・インライン・パッーケージに読み書きのできる256ビットの非破壊記憶素子を一挙におさめたもので，そのアクセス・タイムは1 μs，ビットあたりの消費電力はアクセス時で2 mW，スタンバイ時で50 μWである．各パッケージは，入出力バッファ，選択駆動回路，センスアンプを含めて，各語1ビットで16×16語の構成（コアの電流一致形に対応）となっているが，出力がORで接続できるので，たとえば，256個のパッケージを使えば，16ビット，4K語のメモリが作れることになる．価格は1ビットあたり約10セントである．

また，同時にINTEL社から発売されたPチャネルMOS形の高速ICメモリ（3101形）は，1101形と同様な構造で，容量は64ビット（各語4ビットで16語の語選択構成）であるが，アクセス・タイムは60 nsと速い．価格は1ビットあたり約40セントで

ある.

一方，同じくアメリカの半導体メーカ Motrola 社（国内ではモトローラ・ジャパン扱い）でも，6個のマルチ・パッケージよりなる8,192ビットの IC メモリ・ユニットを発表した．これもPチャネル MOS 形 IC を使ったもので，アクセス・タイムは 100 ns，消費電力は 6W であり，今年末から発売される.

こうした大容量 IC メモリの開発・発売によって，メーカ筋では，単にスクラッチパッド・メモリとしてではなく，電子計算機の主記憶装置として，磁心の代わりに IC メモリの使われる時代が近づいたとしている.

新世代を開く文字読取装置：ASPET/70 を開発

——空間回路網技術に基づく高性能OCR——

通産省工業技術院では，昭和41年以来6箇年の計画で，超高性能電子計算機の研究開発をめざす大形プロジェクトの推進をはかってきたが，このほど，その一環として進められていた光学文字読取装置（OCR）の開発に成功した.

この装置は，工業技術院電気試験所でかねて研究中であった独自の空間回路網技術に基づいて，同所と東芝との間で共同開発された，全く新しい方式の高性能 OCR であり，Analog Spatial Rrocessor の意味で ASPET/70 と名付けられた.

従来の OCR とは異なって，認識操作の全過程が原理上完全なアナログ的・並列的な処理となっている点がこの装置の大きな特長であり，毎秒2,000字という高速認識性能を持つばかりでなく，かすれた文字や目つぶれした文字など，印字品質のかなり悪い文字を正確に判読する能力を持っている.

大形プロジェクトでは，ページ式リーダを最終的に開発する予定になっているが，この装置はその実規模パイロット・モデルとして試作されたものである．このため，とくに認識部に重点を置いた設計がなされている.

対象とされた文字は，IRO OCR-B フォント（サイズI）の英大文字・数字・記号など約 50 種であるが，同一サイズの類似フォントに対する認識性能もきわめて高く，また，字種の増設も容易である．とくに従来問題視されてきたOと0などの，間違いやすい類似文字に対する弁別性能の高いことが実証された.

ASPET/70 では，文字の位置および線幅に対する自動正規化方式が採用されているが，標本化過程を通じて，ボケの操作が積極的に取り入れられていることや，複合類似度法と呼ぶ独特な識別方式が採用されていることなど，画期的な手法の導入によって高性能化がはかられ，従来の技術の障壁を大きく打ち破ることに成功をおさめたものである.

国際会議案内

1970年5月4-6日

Second Annual ACM Symposium on Theory of Computing, Northampton, Mass. Sponsor: ACM Special Interest Committee on Automata and Computability Theory (SICACT). Contact: Prot. Richard M. Karp, Dept. of Computer Science, University of California, Berkeley, Calif.

1970年5月12-14日

1970 Spring Joint Computer Conference, Atlantic City, N. J. Sponsor: AFIPS. Chm: H. Cook, RCA Laboratories, Box 432, Princeton, NJ 08540; ACM Liaison: A. Tonik, Univac, P. O. Box 8100, Philadelpha, PA 19101.

1970年5月21-22日

International Computing Symposium 1970, Bonn, Germany. Spon: Belgian, British, French, German, and Italian Chapters of the ACM in cooperation with Gesellschaft für Mathematik und Datenverarbeitung, Bonn. Coord: Horst Hunke, ACM Conference, c/o Gesellschaft für Mathematik und Datenverarbeitung, 5201 Birlinghoven (Schloss), West Germany.

1970年6月16-18日

IEEE 1970 Computer Conference: The Challenge of the 70's—Memories, Peripherals and Terminals: Trends and their Meaning, Washington Hilton Hotel, Washington, D. C. Conf. Chm: Bob O. Evans, IBM FSD, 18100, Frederick Pike, Gaithersburg, MD 20760; Tech. Chm: T. C. Foote, P. O. Box 1727, Rockville, MD 20850.

1970年6月21-25日

Seventh Annual Design Automation Workshop, Sheraton Palace, San Francisco. Sponsors: ACM SHARE, IEEE. Contact: Ralph Preiss, IBM Corp., P. O. Box 390, Poughkeepsie, NY 12603.

1970年8月24-28日

IFIP World Conference on Computer Education, Amsterdam. Sponsor: IFIP Technical Committee for Education and Administrative Data Proces-

sing Group. Chm: A. A. M. Veenhuis, Sec. Gen., IFIP Conf. Cptr. Educ. 1970, 6 Stadhouderskade, Amsterdam 13, Netherlands. US Reps: W. F. Atchison, Mrs. S. Charp, D. Teichroew.

1970年9月1-4日

●**ACM NATIONAL CONFERENCE,** New York Hilton, New York City. Conf. Chmn: Sam Matsa, IBM Corp., 410 East 62 St., New York, NY 10021. Prog. Chmn: Robert E. Bemer. General Electric Co., 13430 North Black Canyon Highway, Phoenix, AR 85029.

1970年9月7-11日

Sixth International Congress on Cybernetics, Namur, Belgium. Contact: Secretariat, Association Internationale de Cybernetique, Palais des Expositions, Place Andre Rijckmans, Namur, Belgium.

1970年10月4-9日

American Society for Information Science, 33rd Annual Meeting, Bellevue Stratford Hotel, Philadelphia. Sponsor: ASIS. Contact: Kanneth H. Zubriskie, Jr., Biosciences Information Services of Biological Abstract, 2100 Arch St., Philadelphia, PA 19103.

1971年8月3-5日

●**ACM NATIONAL CONFERENCE,** Chicago. Chm: Thomas G. Patterson, Continental Illinois National Bank and Trust Company of Chicago, 209 W. Jackson Blvd., Chicago, Ill.

1971年8月23-28日

IFIP Congress 71, Ljubljana, Yugoslavia. US Comm. Chm: Herbert Freeman (NYU), P. O. Box 4197, Grand Central Post Office, New York, NY 10017.

——— 人工知能特集号論文募集 ———

電子計算機の普及とともに，その高度利用のための背景として，パターン認識，学習，自然言語処理，バイオニックス等々の研究が，ますます盛んに行なわれ，人工知能に関する国際会議も開かれるようになりました．本会会員の中には，既に研究を進めている方，あるいはこの方面に高い関心をもつ方も多数あることと思われます．

この際本会では，第11巻，第11号（昭和45年11月号）を人工知能特集号として，解説のほかに，広く一般からの論文その他の寄稿を歓迎いたします．

下記により奮ってご応募ください．

記

寄稿〆切　6月末日

寄稿の体裁　第11巻，第1号（45年1月号）62ページ記載の「寄稿案内」により学会所定の原稿用紙（注）を使用のこと．

以 上

（注）ご寄稿される方は，学会事務局へご連絡いただければ無料送付いたします．

お詫び （イ） 第11巻の第2号の目次中，「記憶のない通信路の容量の計算法（有本卓）」77頁は79頁の誤りでした． （ロ） 同号63，64頁のノンブルは，同巻第1号と重複していました．深くお詫び申上げます．