
 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

**B-63. プログラマのためのユティリティー
 ・ファイリング・システム**

M.V. Wilkes; A Programmer's Utility Filing System [The Computer Journal, Vol. 7, No. 3, Oct., 1964. pp. 180~184]

PUFS (Programmer's Utility Filing System) の目的はプログラマの使用するすべてのデータとプログラムを磁気テープ上に容易に記入し、紙テープやカードの量を最小にとどめることである。

PUFS 命令は 16 種あり、これについての説明と例が書いてある。

PUFS では、プログラムおよびデータは紙テープ上にパンチされ、それらはパンチされた時と同形式で、字ごとに記録 (store) される。このシステムで使用する磁気テープは 2 本である。Tape 1 はコンパイラなどを含むシステムテープで、Tape 2 にはユーザの作製したファイルと、そのファイルの目録 (directory) が記録されている。

この目録は、ファイルがその磁気テープ上に記録された場所を示すブロック・ナンバーと、テープ番号を持っている。使用者はこの目録を編集したり、この目録を使用して、ファイルを操作することができる。

PUFS 命令は、このシステムのコントロールの下でインプット・テープから読まれ、次に示す動作の中の一つを PUFS が代行する。

- (1) 紙テープまたは磁気テープから記憶場所 (Memory) にファイルを持って来る (load)。
- (2) メモリーの中にロードされたファイルには、番号が与えられ、磁気テープ上に写される (copy)。
- (3) メモリーの中にロードされたファイルに種々な処理がほどこされる (modification, splitting)。
- (4) ファイルの目録に変更が加えられる。

おもな PUFS 命令は INPUT, MODIFY, EXECUTE 等である。

INPUT 命令はドキュメント (document) を従えており、計算機内で各ドキュメントに番号 (line number) が与えられ、番号順に記録される。各番号には 10 の間隔があるので、誤りや、手ぬかりは、既にパ

ンチされてしまった部分を変えずに修正できる。また、磁気テープ上に記録されているものに対しても、ライン、ナンバーを変更することなく修正、追加ができる。

EXECUTE 命令は、磁気テープ上に記録されているアセンブリランゲージプログラムを実行させる。この命令は実行させるプログラムとデータが入っているファイルのリストを従えていて、実行させるプログラムにコントロールを渡す。

ファイルの目録に関する処理は、高速の記憶装置内に、目録を処理するプログラムと、処理を受けるファイルの目録を置いて行なわれ、全ての処理が完了するとファイルの目録は磁気テープに戻される。この目録に関する変更は DUMP または EXECUTE 命令によりユーザのために Print out される。

その他 PUFS の命令は、一つのファイルを二つに分けたり、二つのファイルを一つに結合したり、ライン・ナンバーを新しい番号にする (Renumbering) などの操作を行なうものである。

最後に、ある会員のメンバーのリストについての修正追加、ファイルの結合、などの例が解説されている。

(藤井 狷介)

**B-64. Two-Level Store をもつ計算機の
 プログラムの自動分割処理について**

F.H. Dearnley and G.B. Newell: Automatic Segmentation of Programs for a Two-Level Store Computer [The Computer Journal, Vol. 7, No. 3, Oct., 1964. pp. 185~187]

この論文は計算機の記憶装置として two-level stores を使用する場合に、プログラマに与える負担を少なくするため、効率の良い自動分割処理 (セグメンテーション) の新しいアルゴリズムを与えるものである。そのためには、このセグメンテーションを使用して得られた目的プログラムと、プログラマが作成するプログラムとの実行時間がほぼ同じであること、および 1-pass assembler の中で使われるセグメンテーションの方法については記憶場所の使用と処理時間が経済的であることが必要となる。そこで、これを解決す

るため、前者に対してはブロックに組込める目的プログラムのループはすべてブロックにまとめ、後者には一度に store される主プログラムは必要とする部分だけに限ることとしている。

セグメンテーションの仕事は jump 命令を除いては磁気テープにおける batching data に関するものと類似なもので、ここでは目的プログラムを optimal にするためと、 assembler や compiler の一部となりうるために、主プログラムに特別記号を含む擬似命令を導入する。

特別記号のうち、プログラムループの切り離しは左右の括弧(かっこ)によって行なわれる。これによってくられた部分が1ブロックの範囲を越えるときのみ、その部分に新しいブロックが与えられる。特別記号にはその他 end of marker や label marker がある。

このセグメンテーションは ICT 1301 の計算機で実行され、特に各種 Compiler や assembler によって利用し得るよう、それぞれの目的に従って分けられた6個の入口を持つ 'black box' パッケージが設計されている。これらはそれぞれ目的に応じた働きをするが、このセグメンテーションの効果のかぎは、擬似命令が与えられたときの働きにかかっている。

この内部操作は次のようである。

まずパッケージに付随したバッファに擬似命令や特別記号を一杯になるまで詰め込む。一このバッファのサイズは目的プログラムのブロックより大きい。

次にこの擬似命令および特別記号を順次調べて、別に設けられた二つのカウンタ (optimistic counter, pessimistic counter) を増減し、これと一つの目的プログラムブロックに使用できる space に対する制限 (blue line) との関連で分割点を決める。

分割点は次の二つの基準の一つが満足されると決定される。

- (1) 左括弧と、それに対応する右括弧内の目的プログラムが1ブロックよりあふれるとき。
- (2) pessimistic counter が blue line を横切り、かつ、この範囲内に戻れないとき。

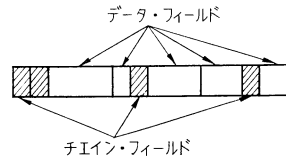
これで分割点が決まると擬似命令を機械命令に変換する。(長谷文子)

B-65. ランダム・アクセス記憶装置のための汎用プログラミング・システム

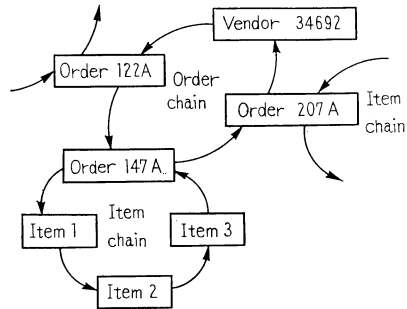
C.W. Bachman and S.B. Williams: A General

Purpose Programming System for Random Access Memories [Proc. FJCC. 1964, pp. 411~422]

大容量のランダム・アクセス・ファイルを用いた情報システムでは、実用しうるファイルの性能に限界があって、ファイル構成やそのアクセスには種々の問題がある。ファイル処理について、プログラマの労力を軽減するための一般的なファイル処理プログラムが必要とされるが、この論文は GE 社で開発した IDS (The Integrated Data Store) について述べたもので、その特殊なものは、GE-215, 225 などのためにつくられたが、一般用としては GE-400, 600 シリーズの COBOL の拡張として、1965年には使用しうるよう進められている。



第1図 レコードの定義



第2図 発注データの構造

このファイルのレコードは第1図のように、データ・フィールドとチェイン・フィールドからなっており、関連するレコードをチェイニングする。これらのレコードはディスク内では、いくつかのレコードのブロックとなっており、ファイルのアクセスはこのブロックごとに行なわれ、処理はコア記憶装置内で行なわれる。たとえば、ある会社の部品発注のデータのファイルでは、COBOL 流のいくつかのレベルに分けたフィールドであらわされ、そのレコードは Vendor No., Order No. Item のデータからなる。これらの各フィールドには、チェイン・フィールドが与えられ、ある特定の Vendor に対する Order No. はチェイニングされる。また、ある Order 中の各 Item のデー

ともチェイニングされる。このように、ファイルのレコードとしては、detail record である。Item records の Master record として、Order record が、Order records の Master record として Vendor record があり、これらはそれぞれ独立にファイルに記憶される。ただし、チェイニングされているレコードは、なるべく同一ブロック内に記憶されるようにするが、その配列順序はどうでもよい。一つのレコードはいくつかのチェインに属し、たとえば Order record は Order chain と Item chain にぞくする(第2図)。ファイル処理のプログラム言語には、Procedural Commands があり、PUT, GET, MODIFY, DELETE の functismal verbs により、レコードの記憶、読み出し、変更、消去を行なう。これらには retrieval rule があり、retrieve するレコードのアドレスをどこからとるか、チェインの中のどのレコードからとるか、などを指定する。

(大野 豊)

B-66. 実験的なオンラインの計算システム (JOSS)

J.C. Shaw: JOSS-A Designer's View of an Experimental On-Line Computing System [Proc. FJCC. 1964, pp. 455~464]

JOSS (JOHNNIAC Open-Shop System)は、Rand社で開発された実験的なオンラインタイムシェア方式の計算システムである。計算機に、電話回線を介して接続されている10個のコンソールから、独立に計算機に指令を送り計算をさせることができる。科学者や技術者が計算機のプログラムやオペレーティングシステムの知識なしに、日常の言語(数式)に近い言語を用いて簡単な計算ができるように開発されたもので、1964年1月から実用されている。

使用された計算機は1950年から53年にRand社に設置されたJOHNNIACである。初期のものであるため、インデックスレジスタ、間接アドレス、浮動小数点演算などはなく、加算速度 $50\mu\text{s}$ 内部記憶容量(磁気コア)4,096語、外部記憶装置(磁気ドラム)12,288語である。コンソールはIBMの電動タイプライタで、電話回線とラインバッファを介してJOHNNIACに接続されている。最大81台のコンソールまで接続できる。

10台のうちの任意の8台のコンソールから同時に計算機に指令を与えることができ、計算機はこれらを

割込み処理により取り扱う。割込みの優先順位はコンソールからの要求が最も高く、次が計算機からコンソールへのアウトプット、最後がその他のものである。通常コンソールからの一つの要求に対して、数分の1秒で答を出すことができる。長いものでも3秒ほどである。

次にJOSSによる計算指令とその解答の簡単な例をあげる。この例よりかなり複雑なこともできる。

U: Type 2+2.

J: 2+2=4

U: Set X=3.

Type X.

J: X=3

U: Type X+2, X-2, 2·X, X/2, X*2.

J: X+2=5

X-2=1

2·X=6

X/2=1.5

X*2=9

U: 使用者のJOSSに対する指令

J: JOSSの解答

なおJOSSのプログラムは約6,000語である。

(関 榮四郎)

C-67. IBM システム/360 の技術

P. Fagg, J.L. Brown, J.A. Hipp, D.T. Doody, J.W. Fairclough and J. Greene: IBM System 360 Engineering [Proc. FJCC. 1964 pp. 205~231]

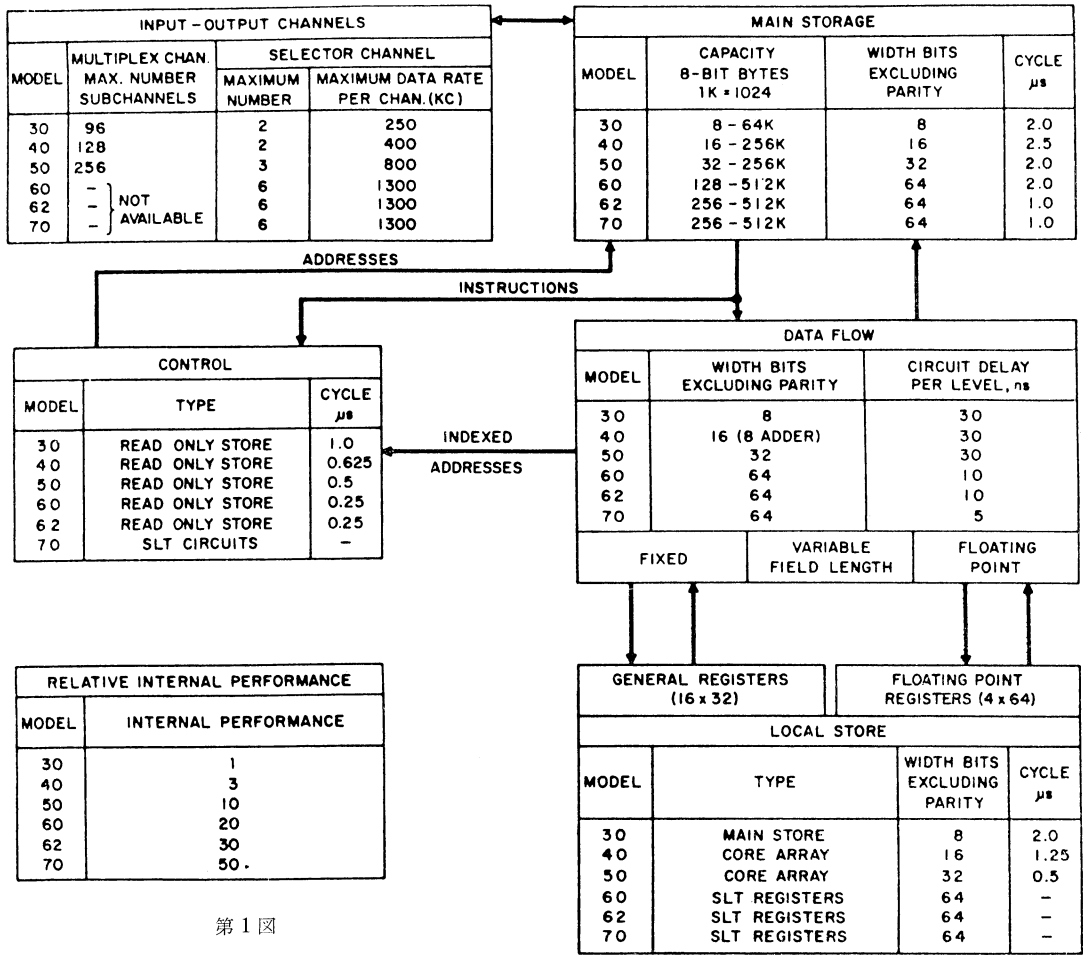
この論文はIBMシステム/360に採用された新技術、設計方針、各モデルの詳細について述べたものである。

システムの設計に先立ち決定した事項は、一つのモデルと次のモデルとの内部性能の比を2.5~3に選び、その結果処理装置の種類を5種にしたこと、上方向および下方向に完全な互換性を持たせたことなどである。

回路は個別的なトランジスタ、抵抗、ダイオードに置き換わるものとしてIBMで開発したマイクロロジック素子、すなわちSLT(Solid Logic Technology)を採用した。

また試作計算機による経験の結果として読出専用記憶装置(ROS)による制御方式を採用したが、その主な理由は次の二つである。

(1) 下方向の互換性に対し、コストの点で有利。



第 1 図

(2) 他の IBM システムとの互換性の実現等に関して融通性がある。

その他、信頼度を高め保守を容易にするための誤りの所在を見つけるシステム (Fault Locating System) の採用があげられる。このシステムは誤りを検出するだけでなく、その位置を5枚の小カード程度に限定する能力を持っている (モデル 50, 60/62, 70)。

モデル 30 は 142 個の演算命令を持ち、IBM 1400 シリーズと互換性を有している。

各モデルのモデル 30 に対する性能の比、使用されている回路の速度、記憶装置の容量ならびにサイクルタイム、ROS のサイクルタイム等が第1図にまとめられている。

なおモデル 70 に関してはサイクルタイム 8 μ s、記憶容量 1,024~8,000 K バイトの大容量記憶装置

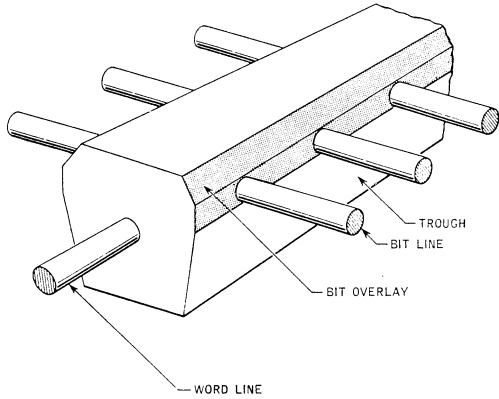
(LCM) の付加が可能となっている。

(幸野真士)

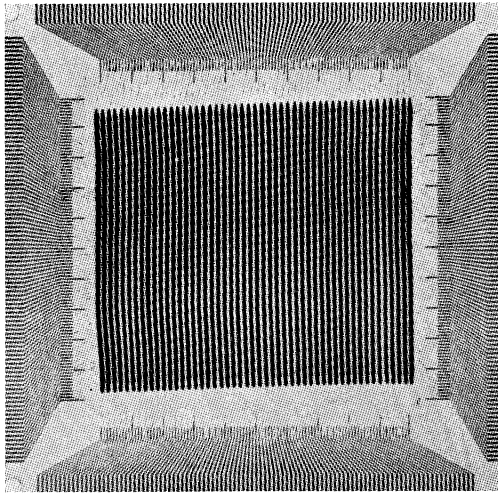
D-68. バッチ構成によるフェライトメモリープレーン

E.A. Bartkus, R.F. Eltant, J.M. Brownlow, K.R. Grebe, W.A. Crape and O.A. Gutwin: An Approach Towards Batch Fabricated Ferrite Memory Planes [IBM Journal of Research & Development, April 1964, pp. 170~176]

フェライトメモリーのバッチ構成の方式については各方面でいろいろ研究されているが、本論文では IBM で開発したフルート状のフェライトメモリープレーンいわゆるフルートメモリーの構造、製造方法並びに電気的特性につきのべる。



第1図 General configuration of the Flute memory element.



第2図 A 50 word x 100 bit Flute memory plane.

形状は第1図に示すように棒状のフェライトの軸方向にワード線が貫通し、多数のビット線が横方向から貫通しているものである。このフェライト棒を70ミル間隔で50本並べてプレインを構成する。これを作るには、上下に対応する溝をもった打ち型を用い、溝にフェライト並びにバインダを詰め、真中に予め格子状に配列したワード線並びにビット線を入れてモールドし、これを焼成することによって作られる。ワード線とビット線との絶縁の確保、並びに焼成によるフェライトの縮小に対しては必要な考慮がなされている。以上の行程によって作られるプレインについて種々の検査が行なわれた。これらの検査はモールドされたプレインの外観並びにX線検査、焼成したのちの外観検査、抗磁力 H_c の測定、ワード、ビット線の導通並び

に絶縁試験および各ビットのパルステストの5段階から成り、その結果が詳細に示されている。

パルステストの結果は IBM-7094 で解析され、その結果、検査基準として“1”の出力 18 mV 以上、“0”の出力 6 mV 以下が適当であると結論され、ワードおよびビット各線に対して 10% のスペアを設け不良ビットを救済するようにした。

以上の実験においてはプレインの動作方式としては両極性ワード電流並びに両極性ビット電流を用いたがフルートメモリーは単極性ワード電流並びに単極性ビット電流の方式でも使うことができる。この場合には出力電圧が若干小さくなる。また高速の非破壊読みだしも行なうことができる。高密度のプレインおよび外径が5ミルのものも試作された。5ミルの素子では 100 mA 以下の駆動電流により 5 mV 以上の出力電圧が得られ、実態密度は1インチ平方当たり 10^4 ビットが可能であると思われる。

動作速度は上述の試験に用いたもので 250 n sec が可能であった。

これらの実験の結果フルートメモリーは新しいバッチ構成フェライトメモリーとして、その可能性が技術的に確認された。さらに小形のエレメントにおいても満足できる結果が得られ、高密度大容量のプレインが小駆動電流で高速に動作することが確認できた。

(伊原博司)

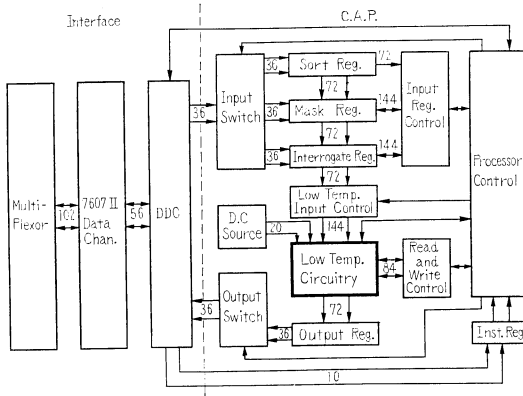
D-69. 超伝導素子による連想記憶装置の構成

J.D. Barnard, F.A. Behnke, A.B. Lindquist and R.R. Seeber: Structure of a Cryogenic Associative Processor [Proc. of IEEE. Vol. 52, No. 10, Oct., 1964, pp. 1182~1190]

最近、超伝導素子が技術的に相当の進展を見せ、これに基づいたアソシイティブメモリーについてもかなり詳しい推測ができるようになったので、5,000語 72 ビット/語の装置を設計し検討した。設計に際しては使用する素子の大きさ並びに特性を仮定し、また装置の機能としては情報の大小比較および orderd retrieval も行なうことのできるものを仮定した。

これを制御する論理回路は第1図の如き構成とし、合計 4,701 個の IBM の SLT を使用することとした。超伝導素子のインダクタンスや抵抗から計算したサイクルタイムが表に示されており、COMPARE AND READ の例をとると約 $10 \mu \text{ sec}$ である。記憶

素子の構成、ブレイン間の接続の方法および論理回路をどこまで超伝導素子で行なうか、などはサイクルタイムに影響を及ぼすが、最小のサイクルタイムが得られるよう考慮した。



この他、語数およびビット数と、サイクルタイム、SLT モジュールの数、冷却器の所要電力、接続点の数、超伝導サブストレートの数などの関係がグラフによって示されている。これによると、サイクルタイムは語数に関係しないが1語当りのビット数には関係し、その他の量は語数、ビット数と共に増加する。

以上の設計の基礎となった回路類は現在まだ十分に解析されていない部分もあるが、さらに詳細な検討並びに素子の改良によりサイクルタイムは短縮されるであろう。

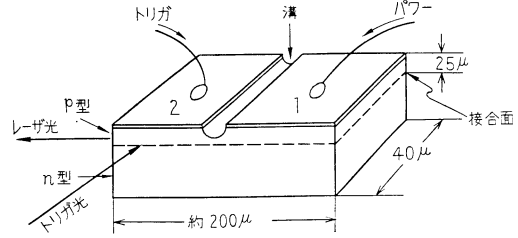
(伊原博司)

D-70. 注入形 GaAs レーザの新制御法とそのスイッチング特性

M.I. Nathan, J.C. Marinace, R.F. Rutz, A.E. Michel and G.J. Lasher: GaAs Injection Laser with Novel Mode Control and Switching Properties [Journal of Applied Physics, Vol. 36, No. 2, Feb., 1965, pp. 473~480]

GaAs などの p-n Junction は、Forward に大電流が流れるときレーザ光(強い単色光)を発生することが知られている。その電流閾値が、接合面全体に均一に流れた時の方が、不均一に流れた時より小さいことを利用して、bistable な装置を製作したという報告である。説明には、3レベルの遷移モデルを考えるとよい。

図のようにP型の部分に溝を etch して二重レーザ



を作る。ただし次の性質を持つものとする。

- (1) クロストーク抵抗は 6Ω 以上あること。
- (2) 片方だけで動作させたときの閾値は、両方向時に動作する時の閾値の1.5倍以上のこと。
- (3) 閾値附近の電流と lasing の関係は鋭いこと。
- (4) 溝はP型の方に作ること。

まず、1の部分に片方だけで lasing する閾値の96%程度のパルス電流(たとえば 77°K で 700 mA)を流しておく("0")(幅は数 μs)。トリガとして2の部分に幅数 10 ns の正の電流パルスを与えると、全体としての閾値が小さくなり lasing が始まる("1")。立上り時間は測定系の立上り時間と同じ 5 ns で、継続時間は 77°K で 800 ns 、 2°K で $4 \mu\text{s}$ だった。

トリガのパルス幅を 1 ns にしても、ピーク電流を数倍にすれば、十分動作することから、応答は相当早いものと思われる。

動作原理としては、2の部分の吸収係数が小さくなること(G.J. Lasherの説)と、2の部分に光電導が起ること(W.P. Dumkeの説)とが考えられる。2の部分の接合面にレーザ光を当ててトリガすることができるから、2の部分は吸収体として振舞っているとも考えられるが、n型の方に溝を作って bistable な装置ができれば、そう結論してよいだろう。

(川合英俊)

D-71. A-D (アナログ-デジタル) 変換器の動特性

W.S. Friauf: Dynamic Characteristics of Analog Digital Converters [Instruments & Control Systems, Vol. 38, Jan., 1965, pp. 111~114]

理想変換器と、誤差係数(サーボ系に実際に使われる)の概念が適用できる線型回路とから成るモデルを用いて A-D 変換器の動特性を論じている。数種類の変換器の係数を導き出すとともに、それらの変換器を種々の入力状態の下で比較検討している。誤差係数を先立ち、まず A-D 変換の誤差の原因を述べる。すな

わち保持回路のない場合の変換中における信号変化によって、出力に生ずる誤差は、入力の変化率と命令信号が与えられてから実際に変換が開始するまでの遅れ時間 T_a (aperture time) に比例する。保持回路のある場合については入力の変化率から得られる誤差 Δe は $\Delta e = T \dot{e}_i$ で、 T もまた aperture time と呼ばれ、この場合サンプル回路を構成する遅延回路の時定数でサンプル回路の周波数特性の 3db 低下の部分から $T = 1/2 \pi f_{3db}$ により得られる。

次にアナログ-周波数型の変換器は、積分型または平均値型とも呼ばれ、その動特性の誤差は平均時間のちょうど中間の入力と比較して得るので、一定入力、ランプ入力に対しては

$$\Delta e = \left[\frac{1}{T_i} \int_t^{t+T_i} e_i(t) dt \right] - e_i \left(t + \frac{1}{2} T_i \right)$$

からゼロである。もし入力波形が放物線 ($e_i = e_0 + kt^2$) であれば誤差は $\Delta e = kT_i^2/12$ となる。

以上に関し線型回路と理想変換器から成る合成モデルを使って動作を評価するが、理想変換器は無限に安定で誤差はなく、動特性の誤差は線型回路だけから生ずるものと仮定する。誤差係数をサーボ誤差のラプラス変換 $E(s)$ から導いて定義している。結果だけを示すと、

$$E(s) = c_0 \mathcal{L}(e_i) + c_1 \mathcal{L}(\dot{e}_i) + c_2 \mathcal{L}(\ddot{e}_i) + \dots$$

となり c_0, c_1, c_2, \dots 等が誤差係数である。

サンプル保持型変換器では $c_0 = 0, c_n = T^n(-1)^{n+1}$ となり、アナログ-周波数型では $c_0 = c_1 = 0, c_2 = a^2$ で、一定入力、ランプ入力に対し誤差はゼロとなる。 $\Delta e = kT_i^2/12$ の場合には $a^2 = T_i^2/24$ が算出される。このうち時間領域での動作を評価するに最適な入力として正弦波を用いた場合の、上記の各変換器についての動特性の誤差を論じている。

また、アナログ-周波数型に関し具体的な数値例をあげて動特性の誤差を論じている。最後に、上記の T_a, T_i の他に変換器は全変換時間についても要求を充たさなくてはならないが、これは主として Shannon のサンプリング理論により決められることになり、このことに関して 2, 3 の必要注意事項を述べている。

(新沢洋一)

E-72. 文字認識論理を設計するためのアルゴリズム

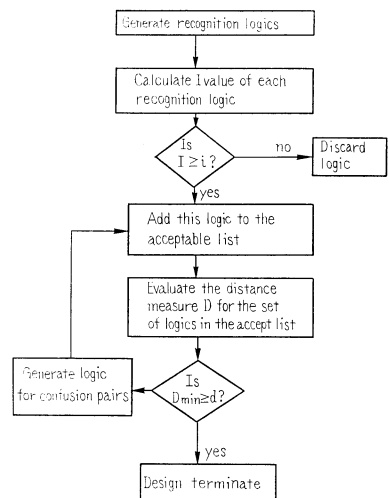
C.N. Liu: A Programmed Algorithm for Designing Multifont Character Recognition Logics

[IEEE Trans. E.C. EC-13 No. 5, Oct., 1964, pp. 586~593]

この論文では、recognition logic の自動設計の algorithm が述べられている。文字を認識する論理回路を設計するには、文字の特徴を抽出する“featuraction”と、文字を幾つかの組の中の一つに属することを見出す“identification”の問題を取扱わねばならない。identification の問題については、多くの理論的研究があり、ここでは、前者、すなわち判定の手続きが与えられた時に、どのような測定を与えられたデータ上に行なうべきか、たとえば文字が 1 と 0 の値をとる要素からなる $N \times M$ マトリックスとして与えられた時に判定を行なう論理回路にその $N \times M$ 個の点のうち、どの点のみを使い、またその点に黒あるいは白のいずれを指定したら効果的であるかをきめる問題を取扱う。この設計の手順は次のようになる。

(1) 文字を特徴づける論理回路の自動生成と評価測定は文字マトリックス上の点の logical And で行ない、平行移動された文字も同じ文字として認識することは、各平行移動された文字の点にたいして行なわれた And の結果の logical Or をとることで行なうようなものを考える。

logic 生成 routine はすべての文字の組から典型的なものの一つずつ読み込み、標準の場所にセットした後にあらかじめ定められた Recognition Logic のよきを示す information measure I の式にしたがって各点の I の値をきめ、その値の大きなものを記録する。これらの high information をもつ点の 5~7 個



の組み合わせをランダムに選び、その各点に指定する黒白の値は、文字の **sample** からきめる。そしてある定められた値より大きな I をもつ組み合わせは **recognition logic** に選ばれうる。

(2) 測定の集合の選択

(1) で作った **logic** の **pool** から与えられた精度で文字を分類できるような **recognition logic** の最少の集合を選び出す。測定の集合をえらぶために特別な **distance measure** を使う。そして2文字間の最少距離を指定し、まず、すべての手に入る **logic** を使って **pairwise** の距離を計算し、最少距離の要求を満たす **logic** を拾い出す。もし、きめられた距離を作るのに十分な **logic** がいないときには、十分な距離をもち得なかった二つの文字を考慮して効果のある **logic** が追加される。この手順を簡単に示すと図のようになる。

(中川圭介)

E-73. Cutpoint を有する論理 Cell の周期的構造

R.C. Minnick: Cutpoint Cellular Logic [IEEE. Trans. EC, EC-13, No. 6, Dec., 1964, pp. 685~698]

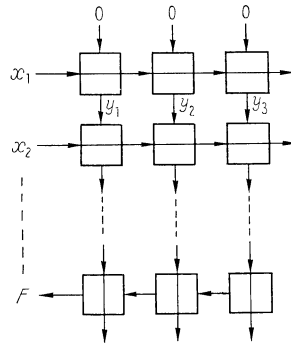
Cutpoint の操作で、論理動作を切換えうる2入力2出力の **cell** を、周期的に配列して論理回路を組む手法について述べている。

可変な **cell** のカスケード結合による論理回路は、既に **Maitra**, **伊吹** も論じているが、この論文は各 **cell** の中に四つの **cutpoint** を置くことで、その構造を簡単にした点に特色がある。

cell は各二とりの場合を持つ6種類の関数を実現できるならば、2入力のすべての関数が作れるので、4ビットの **cutpoint** があれば、その指定に充分なことが示されている。トランジスタは2個用いる。

cell の配列の仕方は第1図のとおりである。各 **cell** は左からの入力 x と、上からの入力 y の関数を出力として下に送り、右の出力へは x をそのまま送る。独立変数 x が入る部分、図で右向き矢印を持つ部分は主部で、左向き部分が収集部である。

このように配置して、論理関数が与えられたときに各 **cell** の論理機能を定める方法は、まず関数を積和形にして、最後で **OR** をとる形を作り、次にそれを簡略化するというもの。簡単化は関数形を吸収律で簡単化する以外に、一般的な手法は与えられていない。レジスタや加算器等の具体例が幾つか与えられているの



第1図

みである。

ハードウェアとしては全く同質な **cell** を周期的に配列した論理回路は、製造上の利点もさることながら、素子のマイクロ化に従い、システムの信頼度の点も大いに期待される。ただこの論文の障害対策は、障害を起した **cell** を含む列をすべて予備の列と置きかえ、それに伴ない幾つかの **cell** の論理機能を変更するというものである。つまり1個の障害のために、1列分の予備が必要となるので、冗長が多くなる。廃棄する **cell** をなるべく少なくし、予備を減らす手法を期待したい。

また障害を起した **cell** の発見は、適当な入出力対応で行なう、とあるが、何行何列目の **cell** が故障したのか厳密には指適できない場合があることには触れていない。これらの点が解決したときに、この技術に期待されるものは大きい。

(伊達 惇)

F-74. PERT/PMD-プロジェクト・モニタリング・デバイス

E.J. Johnson: PERT/PMD-Project Monitoring Device [IEEE. Trans. on E.M. EM-11, No. 2, June 1964, pp. 82~84]

PMD は計算機を用いた計画の立案と修正に関するシステムで、スケジュールの表現は、基本的には **PERT** のネットワーク・システムを用いる。しかし、そのネットワークの作り方が **PERT** と異なる。

「**activity**」という語は仕事の単位を表わすものであるが、**PMD** はスケジュールの基本的な単位として、この **activity** を用いるのである。そして **activity** の配列に重点を置いた、クリティカルパススケジュールの方法を使い、そのスケジュール表に、事象番号をつ

け、それを結び合わせて、スケジュールを表現する。

特徴は次のようなものである。

(1) 仕事にかかる時間の見積りを、一つの時間だけで表現する。

(2) 単一の始点から単一の終点に終る事象の配列をしなくてよい。したがって activity を繋ぐのに、適当な始点と終点とを結ぶ必要がない。

(3) 事象番号は単一のものであり、activity は、その事象番号を共有することができる。

(4) このシステムは仕事の計画において、実際の暦日を算定できる。

(5) PMD は個々の activity の始点と終点の日を算定できる。

この PMD の方法を用いると、従来の PERT にない可能性を、計画の立案において、論理的に引出せる。
(木村幸男)

F-75. 人間-機械系の操作時間とシステムの容量

H. Sackman and J.B. Munson: Investigation of Computer Operating Time and System Capacity for Man-Machine Digital System. [J. ACM. Vol. 11, No. 4, Oct., 1964, pp. 450~464]

一般に実時間方式では負荷が増加すると計算機システム全体の能率が低下する傾向にある。計算機の応答の時間に依りて人間による制御の時間とシステムの構成を考えなければならない。

本論文は人間による制御の時間とシステムの容量の関係についての調査方法に関する方法論的な一般論と経験的な推論が示されている。この研究は防空軍関係の各種設備と数百人に及ぶ軍関係者の協力のもとに行なわれたもので、SAGE システムの入力負荷の関数としての操作時間と、関連したシステムの容量の間の関係について経験的な調査・研究により、実時間処理を行なう大規模な計算機システムの負荷と、これに対する最適システム構成を求めるための一般解を得ようとしている。

この研究は、調査の結果の信頼性、試験の範囲、調査結果の記録、試験の能率、などに十分考慮を払っており試験の統計学的デザインにも苦心している。

試験は SAGE の Frame 処理時間とシステムの能率を測定することにより、これを基として各種の問題を考察している。Frame の処理時間の測定や試験の記録などに用いるために、数種のシミュレーション・

プログラムを作成、使用している。

調査の結果からは SAGE システムにおける各種の分析が行なわれているが、これを一般の計算機のシステムにあてはめることはできない。しかし、この研究で示された考え方、調査のための方法論については一般にも適用できるはずである。
(高村真司)

F-76. 通信網シミュレーションプログラム -UNISIM

J.H. Weber and L.A. Gimpelson: UNISIM—A Simulation Program for Communication Network [AFIPS, FJCC, Vol. 26, 1964, pp. 233~249]

通信網のトラフィックシミュレーションの特殊性はおもに全体の呼の 1% 程度の損失呼を対称として呼損率を推定する点にある。しかもわずか 20 局程度の中継点からなる通信網でも中継回線の束は 190 にもなり、そのおのおのに対し異なった呼量でシミュレーションをしなければならない。それゆえ一つの呼が呼損率に寄与する割合は規模が大きくなるにつれて減少する。したがって 20 局程度のモデルに対しても数百万の呼を加えなければ、十分な信頼度で呼損率を推定することができない。

この論文では特に高速化に重点をおいて IBM 7090 に対して作ったトラフィックシミュレーションプログラム UNISIM の機能と構成について報告する。

UNISIM では即時呼、待時呼の任意の組み合わせのトラフィックに対する迂回中継のシミュレーションが容易に行なえる。また再呼現象のあるトラフィックに対する取扱いもできる。UNISIM でシミュレーションできる上限は、中継点 63、中継回線束 1953、同時接続数約 6,000 までであり、速度は 1 時間に約 500,000 呼ぐらいとなっている。

UNISIM は Traffic Generator, Simulation Program, First Output Processor, Second Output Processor の 4 部に大別できる。

Traffic Generator は簡単なパラメータから Simulation Program への入力となるすべての呼を生成する。同じトラフィックで異なった網構成のシミュレーションができるよう磁気テープに出力する。Simulation Program は通信網構成を記述するパラメータにしたがって、生成されたトラフィックについてシミュレーションを行ない、その結果を呼と回線の状態を二つの磁気テープに出力する。First Output Processor は、シミュレーションの結果を指定された区間ご

に集計し、呼損率、待合せ率など各種のトラフィック統計をとるもので、Second Output Processor は、区間集計された統計から網全体に対する最終的なトラフィック諸統計をとるものである。

(新井克彦)

F-77. 交通制御のアルゴリズム

Michael C. Dunne and Renfrey B. Potts: Algorithm for Traffic Control [ORSA Vol. 12, No. 6, Nov.~Dec., 1964, pp. 870~881]

交叉点の交通制御を線形制御モデルを用いて行なう方法を述べたものである。交叉点に来る車および出て行く車の時間的割合が一定と仮定し、交通量が飽和状態に達しない条件を入れると、この制御モデルは安定であり、しかも短時間で平均待時間が最小な定常状態に収斂する。

(1) 制御モデル 2相の信号機で制御される4又路を考える。互に隣り合う道路を a_{rm1} , a_{rm2} とする。各時刻 t で次の四つの量を検討する。

a_{rm1} の車の待ちの数: $n_1(t)$

a_{rm2} の車の待ちの数: $n_2(t)$

a_{rm1} の制御関数: $f_1(t) = \alpha_1 n_1(t) + \beta_1 - n_2(t)$

a_{rm2} の制御関数: $f_2(t) = \alpha_2 n_2(t) + \beta_2 - n_1(t)$
($\alpha_i > 1$, $\beta_i > 0$)

上のいずれかが零ならば信号を切り換える。ただし最小持続時間までは切り換えをせず、最大持続時間を超えれば自動的に切り換える。

(2) 交叉点に生ずる待ち車の出入は連続とする。

a_{rm1} , a_{rm2} の車の Arrival rate; q_1 , q_2

a_{rm1} , a_{rm2} の車の Departure rate; s_1 , s_2

$y_1 = q_1/s_1$, $y_2 = q_2/s_2$, $Y = y_1 + y_2$

信号1相当りの損失時間: l

j 相の有効な青時間: g_j

で表わす。信号のスタート時は a_{rm1} の側に青であるとして、車の待ちは j に関する差分方程式となる。これを解いて一般解 $n_2(2j-1) = Aa^{j-1} + N_2(\infty)$

ただし a は s_i , q_i , β_i の関数である。 $s_i' < q_i$ に対して $0 < a < 1$ となり、 n_2 は一様に $N_2(\infty)$ に収斂する。すなわちこのモデルは安定である。

(3) リミット・サイクルタイム

a_{rm1} が青の時間: $g_{1\infty} = 2ly_1/(1-Y)$

a_{rm2} が青の時間: $g_{2\infty} = 2ly_2/(1-Y)$

リミット・サイクルタイム:

$$c = g_{1\infty} + g_{2\infty} + 2l = 2l/(1-Y)$$

なおトロント市では感應信号を設置し、信号相互間の調節を UNIVAC 1107 でシミュレートし、最適解を求めて行なっている。(半田 福)

F-78. データ処理システム・シミュレータ

M.I. Youchah, D.D. Rudie and E.J. Jonnson: The Data Processing System Simulator (DPSS) [Proc. FJCC., 1964 pp. 251~276]

DPSS は Strategic Air Command Control System のプロジェクトとして開発され、システムデザインやその開発のために、他のシステムにも応用できる様に一般化された。

システムに対する負荷、そのシステムを設計する場合、何をパラメータにしたらよいか等という、データ処理システムの性能を調べるのに、この DPSS は用いられる。

一般的なシミュレーションと同様、高次のシミュレーション言語を用い、マクロ命令を簡単に結合させることにより、種々のデータ処理システムを表現できるようになっている。

フロチャート、デザイン、コンパイル、チェックなどは必要ない。JOVIAL で書かれた、約1,500の命令で用が足りる。DPSSは、システムへの入力、各入力に対する計算機の処理、システムで処理された結果を表現する。機能をまとめれば次の如きものである。

(1) システムが処理を実行できるかどうかということ調べる。

(2) 計算機を基としたデータ処理システムを模擬する。

(3) 処理原則に基づいた装置とそのシステムに要求される能力が妥当であるかどうかを評価する。

(4) システムの装置をはっきりさせること。

(5) プログラム処方を確立すること。

(6) 詳細にわたる、要求されるハードウェア、ソフトウェアを開発する。

(7) 実際のシステムに対して、初期のパラメータを設定する。

(8) システムがどんな稼働をしなければならないかを決定する。

拡張された DPSS としては、現在モデル-C がある。これは IBM 7090/7094 計算機で使用できる。

(木村幸男)

F-79. コンピュータを用いたレジスタ・トランスファ・ランゲージによるデジタルシステムのデザイン

H. Schorr: Computer-Aided Digital System Design and Analysis Using a Register Transfer Language [IEEE Trans. EC-13, No. 6, Dec., 1964, pp. 730~737]

論理動作を記述する言語, register transfer language を, 論理構造を記述する言語, 論理式にコンピュータによりほん訳し, デジタルシステムの論理設計およびその逆の解析の自動化をおこなう.

register transfer language は, たとえば

$$|t_2| : N \oplus A \rightarrow A; 1 \rightarrow t_3$$

により, 状態 t_2 においては A を N と A の exclusive or とし, 同時に次の状態を t_3 にするという micro-operation を表現し, これらの集合により, システムの行なう動作が記述される.

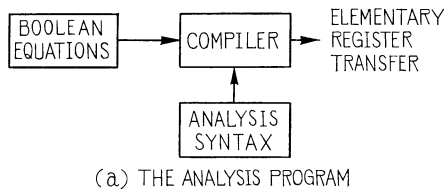
論理式は, たとえば $A[i]$ によりレジスタ A の内の i 番目のフリップ・フロップの状態を表わし,

$$A[i]/1 = N[i] \cdot A[i]' \cdot t_2 + N[i]' \cdot A[i] \cdot t_2$$

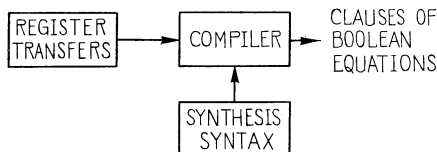
により右辺の論理式の値が 1 の場合は $A[i]$ を 1 とする論理回路をしめし, これらの集合によりシステムの論理構造が記述される.

adder を例にとり, 前者が何段かのステップを経て最終的に後者にほん訳され, 設計がおこなわれることをしめしている.

両言語はいずれも, Buckus normal form により規定される構成をもち, "Syntax directed compiler" の syntax table を本論文中にしめす syntax で置き



(a) THE ANALYSIS PROGRAM



(b) THE SYNTHESIS PROGRAM

The use of a syntax directed compiler in the analysis and synthesis of digital systems.

変えることにより図にしめすフローをおこなうプログラムが得られる.

コンパイラーは syntax table を参照しながら recursive にほん訳を進めるが, その過程は本文中の "Formation diagram" にしめされている.

これは単に設計, 解析の自動化を図っただけでなく, 以後の回路変更とその影響のチェック, 保守, シミュレーション等に有用な record を残すほかシステム評価の手段としても利用できる. この方法の有用性はしめされたが, より有効なものにするためにさらに種々の拡張が必要である. (飯田麒一郎)

F-80. 索引の機械化とその検定

John O'Conner: Mechanized Indexing Methods and Their Testing [JACM. Vol. 11, No. 4, Oct., 1964, pp. 437~449]

この論文は情報検索における索引 (Indexing) 法の機械化, つまり計算機による主題の索引法に関して, これまでに提案されている各種の方法を系統的に要約している.

これまでに提案されている方法はすべて, 完全にまたはほとんど機械化された文書の準備プロセスとそれによって準備された文書に対してある索引規則をあてはめることにつくされているといえる.

この論文では文書の準備プロセスに使われる各種のテクニック, たとえばテキスト全体の入力, 文書にタイトル, アブストラクト等のマークをつける方法, 文章やパラグラフのマーキング, 代名詞のおきかえ, 構文上のマーキングなどについて解説する. このほか辞書式のヘディング, 位置番号, 荷重出現頻度, 重要度のメジャー, 参照情報などの附与等も説明されている. これまでに提案されている方法はいずれもここに述べられた各種テクニックの組み合わせと考えられる.

そのほか索引規則 (Indexing Rule) に関する手法を 3 種に分類し, それぞれ例示する.

そして索引の機械化に対する否定的な意見に対し, 一般的に注釈を加えている.

機械化された索引法の質は, それによって実現される情報検索の効率によって評価できるが, その際の問題点をいくつか述べている.

最後に Indexing duplication に関する研究が, 機械化した索引法の実験的な審査に代わり得るものとして, 提案されている. (新井克彦)

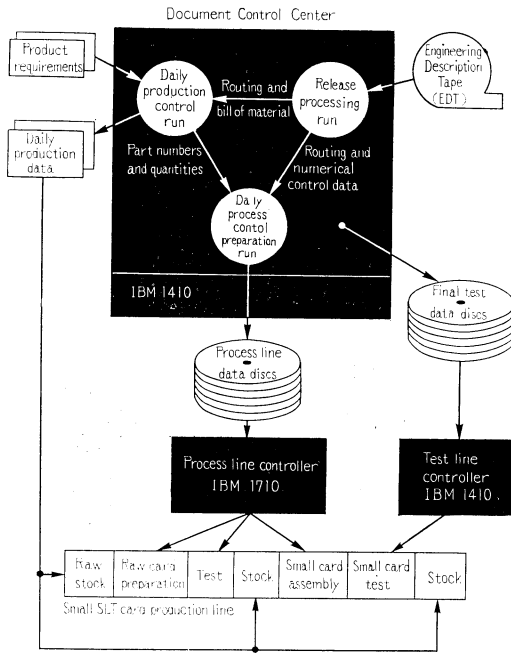
F-81. 製造ラインを制御する多重計算機システム

E.E. Sarafin: Multiple Computer System Controls Manufacturing Line [Control Engineering, Vol. 11, No. 12, Dec., 1964. pp. 83~92]

本文献はIBM/360に使われる論理回路カード (SLTカード) の製造ラインを完全に自動化する計画について、著者が IBM の技術者達とのインタビューからまとめ上げたもので、かなり具体的に述べられている。

計算機は、全システムのモニターおよび制御用として 1710 を、それに必要な種々の情報の集積と判断用として 1410 を、そのバッファ処理機として 1401 を、それぞれ用いている。

図に全システムの概要が示されている。同図で、



A quick look at whole system

EDT と呼ばれる磁気テープには、プリント配線表、穴の配列、構成部品の配置と数、入出力電圧、素子の形状と数、等々のほかに設計およびその変更に関するデータ、図面など、SLT カードの製造および試験に必要な技術的なデータが総て記録されている。このテープが 1410 にかけて、指定部品使用の可否、ドリルの寸法の指定から最適な製造ルートの設定までを含めて、情報の照合、判断、選択などが行なわれる。もしその情報が全く新しい設計に関するものであれば、1401 を使って必要な資料の hard copy が作られる。それと同時に、材料表、NC データ、ルーチン情報が作られるが、NC データにはこの制御システムが故障した際に使われるパンチカードも含まれている。

他方、その日一日の生産計画および試験に必要な情報などを記録している磁気ディスク・ファイルが選択され、部品の在庫調整、発注計画、機械の製造能力等が、これまでの稼働経過にもとづいてチェックされる。このほかに、原価算出、QC 確立、など管理用の各種ファイルも用意されていて、共に 1410 による処理がなされる。

このようにして 1 日分の稼働に必要なデータが揃うと、1710 に引渡され、NC 機械および試験装置との間で、制御信号や比較信号などの授受が行なわれ、SLT カードの製造動作がはじめられる。もし誤りが発生すると、1710 で診断ルーチンが走って、誤った部分のプロセスを孤立させ、症状の原因についての統計的な分析がなされ、製造機械に対して必要な処置をとり、オペレータへは結果を印字して知らせる。

できてきた SLT カードは 1410 を使った試験装置によって最終的に試験される。もし不良が発見されれば診断ルーチンが走り、結果が欠点カードに記され、また将来の分析のために、ファイルに貯えられる。

この自動製造システムは、本年中に稼働に入る予定であるが、このシステムの個々の処理法は、他の似たような製造行程の制御へ広く応用できるだろうと、著者は述べている。(渡辺豊丸)