

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-7. 行列の固有ベクトルの計算について

Д.Ф. Давиженко: К вопросу о вычислении собственных чисел и собственных векторов матриц [ДАН, Доклады Академии Наук СССР, Vol TM-141, No. 5, 1961, pp. 277~280]

$A(\lambda) = \|a_{kl}\|$ は n 次の正方行列で, a_{kl} は有限な区間 $\lambda_0 \leq \lambda \leq \lambda^*$ で λ の関数である. $A(\lambda)$ の固有値は複素数

$P_j(\lambda) = P_{j0}(\lambda) + i \cdot P_{j1}(\lambda)$ で, 区間上のある λ , たとえば

$\lambda = \lambda_0$ で $P_{j0}(\lambda) = P_{j0}^{(0)}$, $P_{j1}(\lambda) = P_{j1}^{(0)}$ がわかっているものとする.

さらに次の二つの仮定を加える.

(1) $a_{kl}(\lambda)$ は区間 $\lambda_0 \leq \lambda \leq \lambda^*$ で定義され連続でかつ連続な導関数を持つ.

(2) 行列 $\|A(\lambda) - (P_{j0} + iP_{j1}) \cdot E\|$ の余因子行列 $C(\lambda, P_{j0}, P_{j1})$ の Spur は点 $(\lambda_0, P_{j0}^{(0)}, P_{j1}^{(0)})$ で 0 でない.

区間上の任意の λ に対して行列 $A(\lambda)$ の固有値 $P(\lambda)$ の近似値を計算するためには次の二つの方程式を上記の初期条件を使って λ_0 から λ まで数値積分すればよい.

$$\frac{dP_0}{d\lambda} = \frac{S_P C_0^* S_P (C_0^* dA(\lambda)/d\lambda) + P_1^2 S_P C_1^* S_P (C_1^* dA(\lambda)/d\lambda)}{(S_P C_0^*)^2 + P_1^2 (S_P C_1^*)^2}$$

$$\frac{dP_1}{d\lambda} = P_1 \frac{S_P C_0^* S_P (C_1^* dA(\lambda)/d\lambda) - S_P C_0^* S_P (C_0^* dA(\lambda)/d\lambda)}{(S_P C_0^*)^2 + P_1^2 (S_P C_1^*)^2}$$

ここで $C_0^* = C_0^*(\lambda, P_0, P_1)$, $C_1^* = C_1^*(\lambda, P_0, P_1)$ は次のようにしてきまる.

$$\|A(\lambda) - (P_0 + iP_1) \cdot E\| = \begin{vmatrix} L(\lambda, P_0, P_1), u(\lambda) \\ v(\lambda), a_{nn}(\lambda) - (P_0 + iP_1) \end{vmatrix}$$

$$\bar{L} = \det\{L(\lambda, P_0, P_1)\}, C(\lambda, P_0, P_1) = \bar{L}(\lambda, P_0, P_1) \cdot C^*(\lambda, P_0, P_1), C^*(\lambda, P_0, P_1) = C_0^*(\lambda, P_0, P_1) + i \cdot P_1 C_1^*(\lambda, P_0, P_1)$$

行列 $C^*(\lambda, P_0, P_1)$ の各列は固有値 $P(\lambda)$ に対する固有ベクトルである. (根田美佐子)

A-8. 一桁毎に演算する多項式の解法

J.E. Meggitt Digit-by-Digit Methods for Polynomials [IBM Journal, Vol. 7 No. 3, July 1963, pp. 237~245]

$f(x)$ を n 次の多項式, $n+1$ 個の $A_r(x_0)$ は既知として

$$f(x) \equiv \sum_{r=0}^n A_r(x_0)(x-x_0)^r$$

とすると

$$f(x_0) \equiv A_0(x_0)$$

$A_r(x_0)$ から $A_r(x_0+a)$ をつくる

$$f(x) \equiv \sum_{r=0}^n A_r(x_0)[(x-x_0-a)+a]^r$$

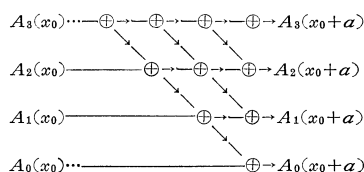
$$\equiv \sum_{r=0}^n \sum_{s=0}^r A_r(x_0) C_s^r a^{r-s} (x-x_0-a)^s$$

$$\equiv \sum_{s=0}^r A_s(x_0+a)(x-x_0-a)^s$$

係数を比較して

$$A_s(x_0+a) = \sum_{r=s}^n C_s^r a^{r-s} A_r(x_0)$$

$A_s(x_0+a)$ は $A_r(x_0)$ から第1図の binomial tree によって簡単に得られる.



第1図 binomial adding tree

$$\begin{matrix} A_3 & A_3 & A_3 & A_3 \\ A_2 + aA_3 & A_2 + 2aA_3 & A_2 + 3aA_3 & A_2 + 3aA_3 \\ & A_1 + aA_2 + a^2A_3 & A_1 + 2aA_2 + 3a^2A_3 & A_1 + 2aA_2 + 3a^2A_3 \\ & & & A_0 + aA_1 + a^2A_2 + a^3A_3 \end{matrix}$$

第2図 adder outputs

第1図の記号 \oplus は word adder を意味し, 斜線は a による乗算すなわち shift を意味する.

このやり方の特徴はすぐ前の計算結果を $A_r(x_0)$ とし, それをもとにして $A_s(x_0+a)$ を計算し, 一桁ずつ精度を高めていくところにある. 即ち $f(x_0) < 0$ $f'(x_0) > 0$ として $x = x_0, x_0+1, x_0+2, \dots$ と $f(x)$

を計算し $f(x_0+n+1) > 0, f(x_0+n) < 0$ なる n に対し $x = x_0+n+0.1, x_0+n+0.2, \dots$ と $f(x)$ を計算することにより $f(x) = 0$ の根を求める。(水田幸夫)

B-9. 小さいフィールドの 10 進-2 進変換

L.D. Yarbrough: Decimal-to-Binary Conversion of Short Fields [Comm. ACM, Vol. 6 No. 2 Feb. 1963, pp. 63~64]

インプット・プログラムを 2 進法の計算機のためにつくるときに、BCD コードであらわされた 10 進整数を 2 進化することが必要となる。普通には BCD コードの 1 バイトずつを 2 進化し 10 (2 進の 1010) をかけて加えてあわせていくのであるが、ここではこの乗算と部分和の演算すべてが乗算命令 1 回で完了する方法をのべる。

この方法は、(1) 二つのレジスタでこれらを結合しても、別々にでもシフトができる、(2) AND 命令をもつ、(3) 乗算が速い、(4) 0 から 9 までをあらわすバイトが 2 進数と同値な 4 ビットとゾーン・ビット 0 からなる計算機に適用される。

BCD コードで

$$A = a_1 \cdot 10^{N-1} + a_2 \cdot 10^{N-2} + \dots + a_N, 0 \leq a_i \leq 9$$

とかかれる整数は上記の条件 (4) から

$$A' = a_1 B^{N-1} + a_2 \cdot B^{N-2} + \dots + a_N,$$

$$B = 2^b, b = 1 \text{ バイト中のビット数}$$

であるような 2 進数をあらわしている。条件 (1) のスプリット、シフトを用い、 M をバイトの長さとして

$$A' = a_1(2^M)^{N-1} + a_2(2^M)^{N-2} + \dots + a_N$$

とあらわせる。ふたたびシフトして

$$(a_1 + a_2(2^M)^{-1} + \dots + a_N(2^M)^{1-N})/16$$

とし、これに定数

$$\{(2^M)^{N-1} + 10(2^M)^{N-2} + \dots + 10^{N-2}(2^M) + 10^{N-1}\} \times 16$$

をかけ、 2^M が共通因数になっている部分と 1 より小さい部分をマスクして除けばよい。

フィールドの大きさを考える。 N 桁の整数を処理するのに必要な語の長さは

$$(4 + (N-1)[\log_2(10^N) + 1])$$

である。ここで [] はガウスの記号である。

(磯田和男)

C-10. IBM 7950 HARVEST

P.S. Herwitz and J.H. Pomerene: The Harvest System [Proc. WJCC May 1960, pp. 23~32]

HARVEST は非算術的データを取扱うためのデータプロセッサで、IBM 7030 STRETCH をその一部として含むようなきわめて大形の組織である。しかし単に大型、高速というだけではなく、データの単位のとりにかた、演算の仕方についてユニークな考慮がはらわれた。

まず命令のオペランドとしてのデータの単位を、一語ではなく、一連の長いつながり (data stream) にとった。このことは記号処理の場合のストリングやデータ処理の場合のファイルを取扱うのに論理的でもあるし、プログラムも容易である。またこのように長い単位を入れるレジスタを用意するのは得ではないので 3 アドレス方式にした。各アドレスはコントロール・ワードリストを介して複雑なデータ・ストリーム・ボタンを指定する。指定されたデータは流れをなしてとりだされる。

演算のうち、算術演算や論理演算のような定型的なものとは金物で組めるが、記号処理の場合ひどく任意な演算がありうるので、それをテーブルでおこなうようにした (on line table lookup)。すなわち二つの入力を引数としてテーブルをひき、値を得る。このときストリームは 8 ビットの要素 (byte) がつながったものと解される。そして上の過程がストリームのつづくあいだ繰返される。

正規の命令語は 192 ビットよりなり、ストリームを解発する働きをもつが、処理中の条件についてマイクロプログラムで割込み、細部の処理をしてからデータの流れを再開できる。プログラムの速さは、内部分類が 3 万件/1.25 秒、クリアが 2^{18} 語/1 ms 程度である。

HARVEST の構成性能は次のとおりである。なお記憶装置の金物の 1 語は 64 ビット + チェック 8 ビットからなっている。

streaming mode section

{	高速記憶	アクセス 0.7 μ s, 1024 語 \times 2
	高速テープ	112 万字/秒
	ストリームユニット	330 万字/秒

arithmetic section (IBM 7030 STRETCH)

{	主記憶	アクセス 2.1 μ s, 16384 語 \times 2
	中央演算装置, memory bus	
	入出力装置, exchange	

この論文の大部分は Buchholz 編: Planning a Computer System 第 17 章に再録された。

(西村恕彦)

C-11. 金物で解釈されるマクロ命令

R.M. Meade: A Discussion of Machine-Interpreted Macroinstructions [Preprints: 16th National Meeting ACM, Sept. 5-8, 1961, p. 6 c-1-1~2]

Neumann 型の計算機では一つの命令で参照処理するのは本質的に一つのデータ語である。データ語はそれ以上分割できない単一の要素である。しかし入力装置での情報のやりとりは、一つの命令で大きいブロックを扱うことが多い。IBM 7950 は一つの命令で要素の長いつながりを処理する機能を一般の内部処理にまで適用した。二つの入力から一つの出力を得る過程があたかもベルトコンベアシステムのように連続的にとられる。命令とデータとを記憶装置から交互にとりだす不能率が完全に避けられ、最大限の速さでデータの流れを処理しつづけられる。

このマクロ命令には FILE-MAINTENANCE, SEARCH, SORT, SELECT, INDIRECTLOAD-STORE, SEQUENTIAL-TABLEREFERENCE などがあつた。これらは 1 次命令としてプログラマが使えるが、機械のほうでは内在サブルーチンで実現される。

(西村怨彦)

C-12 マイクロ命令方式

E.D. Conroy and R.M. Meade: A Microinstruction System [Preprints: 16th National Meeting ACM, Sept. 5-8, 1961, p. 6 c-2-1~2]

計算機の命令には単純なものから複雑なものまでいろいろある。IBM 7950 ではこれを明白に 2 分する。1 次 (マクロ) 命令は非常に複雑な算法をおこなう。比較的単純な処理は 2 次 (マイクロ) 命令とする。これは 1 次命令を実行している過程で発生する特別な条件 (刺激) をずっと見張る働きを担当する。それにぶつかると 1 次命令による処理が中断され、2 次命令のマイクロサブルーチンが割込む。それが終ると 1 次命令に戻り処理をつづける。

割込みが幾つも同時におこったりすると面倒なことになるが、7950 では任意の組合せで入れ子にできるようになっている。なお 2 次命令でできる処理にはカウンタのセット、ステップ、インデクス修正、データ移動、挿入、削除などがある。

(西村怨彦)

C-13. D 825-指令管制用マルチコンピュータシステム

J.P. Anderson, S.A. Hoffman, J. Shifman and R.C. McReynolds: D 825-A Multiple Computer System for Command and Control [AFIPS Conf. Proc. Vol. 22, 1962 FJCC, pp. 86~96]

指令管制用マルチコンピュータシステムであるバロース D 825 の方式について述べたものである。方式設計の基準として Availability (金物の信頼度が高く保守が容易といったこと)、Adaptability (与えられた仕事に無駄なく適応でき、また外部からのアラーム、システム中での事故などの場合金物を自動的に交換できること)、Expansibility (機能の追加などのシステム変更が容易に可能であること)、Programming (システムを変更しても前のプログラムがそのまま使えること) ととりあげる。種々の方式を検討した結果最適なものとして、対等な計算機組織を置き必要なときだけどれか一つを Master とする Totally Modular Scheme が採用された。この場合全てのメモリは全ての計算機に共有され、I/O 制御は計算機には直接結合されない。こうすれば上の基準は全て満足され、システムの柔軟性が増し、あらゆる問題に対処できる。計算機とメモリ間および I/O 制御と I/O 間が自由に結合できる必要があるが、D 825 ではこれを Switching Interlock と Automatic Operating and Scheduling Program (AOSP) により行なう。

Switching Interlock はマトリックス状に配置した電子スイッチで、計算機とメモリ間を結合する。もちろん結合には優先順位もつけられる。AOSP は全システムを制御し、与えられた仕事を実行するもので主な機能は、(1) システムの配置の決定、(2) メモリの割り当て、(3) スケジュール作製、(4) プログラムの準備とあと始末、(5) 報告とロギング、(6) 診断と信頼度チェック、(7) 外部からの割り込みの処理、である。結論として大量の情報を処理するには、単に大きくて速い計算機よりも、このような Modular System の方が自然であることを強調している。なお D 825 の最大実装は、計算機 4、メモリ 4,096 語×16、I/O 交換機 1 または 2、I/O 制御は 1 交換機当たり 10、I/O 装置は 1 交換機当たり 64 である。

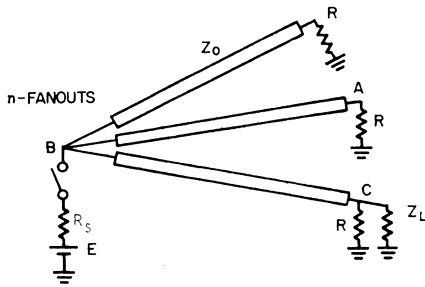
(酒井正昭)

D-14. 高速トランジスタ計算機の信号伝送の解析

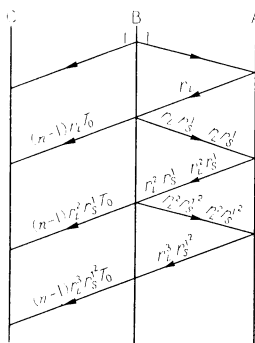
F.C. Yao: Analysis of Signal Transmission in Ultra High Speed Transistorized Digital Computers [IEEE Trans. EC-12, No. 4, 1963, p. 372]

繰り返し周波数が 10 mc を越える高速計算機において、信号の伝送は結合線路上の伝ばん時間が全遅延時間の重要な部分を占めるばかりでなく、分布常数線路が論理回路の一部となって、その動作に影響を与えるのできわめて重要である。

低入力インピーダンスの場合は、出力端子から所望の回路へ第1図のごとく線路を放射状に接続する構造が用いられるべきである。論理回路の入力インピーダンスはゲート作用のために大幅に変化する。もしそのゲートが他の信号によりすでに駆動されているならば、最初負荷電流は0で、時間の経過と共に増大する(負荷排除の条件)。この最初の反射を減ずるために負荷に並列に抵抗Rを接続する。もし、そのゲートが他



第1図



第2図

の信号により駆動されていないときは、負荷の値は低く線路インピーダンス Z_0 に整合するものとする。図のAは負荷排除の場合の受端、Bは送端、Cは整合し

た受端である。この場合の反射の様子は第2図に示される。ラプラス変換を用いて各点の電圧波形の式が求められる。C点の立上りは driver のそれに等しく、立下りは反射のために劣化する。有限な傾斜をもつ入力波形に対してC点の波形の近似式が求められている。立下り時間を指定し、線路長、driver の波形を指定すれば、許容される反射係数が計算される。その具体例と観測波形が紹介されている。

放射状線路は結線がかさばるので、タップ付線路も用いられるが、この場合には、エミッタホローを入力側に用いて高入力インピーダンスにすべきである。エミッタホローの入力インピーダンスは RC 回路で近似される。7個の負荷を線路上にタップをつけて結合する例を示し、送受端波形の表示式を求めた。driver の波形(立上り時間)、負荷相互間の最小距離、線路インピーダンス Z_0 、入力容量を与えたとき、加える波形(1~40 ns)に対する反射の割合を計算機で求め、やや詳細に図示してある。これらの結果は実験でも確かめられ、観測波形が紹介されている。最悪波形は送端に最も近い受端に生ずるという法則が得られる。タップ付線路の最終端は抵抗で終端すべきで、送端も Z_0 に等しくすることが望ましい。エミッタホローを用いるときは望ましくない振動に対して注意しなければならない。漏話は予測しにくい、それが非常に大ならば、ここで得られた波形に重畳しなければならぬ。(畔柳統栄)

D-15. ナノセカンド回路と結線技術

N.R. Crain: Nanosecond Circuits and Interconnection Techniques[Proc. Pacific Computer Conf. March 1963, pp. 131~138]

高速大形計算機の配線および回路に関するもので、種々の原因によるノイズ、比較的長い線路を駆動する回路、高速でかつ信頼性の高いフリップ・フロップ、クロック系統およびその回路などについて具体的に述べている。

まずノイズについて、論理回路間を結ぶ配線上を信号が伝わって行く時間と、信号のスイッチング時間とが近づいてくると、これらの線を従来のように単なる集中容量として考えることはできなくなり、伝送線として、取り扱いねばならない。この場合、負荷を線路に接いだ分岐点や終端におけるミスマッチングにより反射を生じ、これがノイズの一因となる。他のノイズ源として、ゲート回路のトランジスタまたはダイオー

ドの蓄積電荷によるもの、電磁的な結合によるもの、接地面上の電位が場所により異なること、直流的なレベルシフト、線路の特性インピーダンスが一様でないことなどによるものがあり、これらのノイズに対する対策を必要とする。

次々線路を駆動する場合に、通常の電流源を用いた回路では、受端において必要なレベルに達するまで、信号は場合によると線路上を何度か往復しなければならず、大形計算機では線路長はかなり長くなるので、時間的に問題となる。ここでは正帰還回路を用いて立ち上り部分で線路に大きな電圧を加えることにより解決している。

次にフリップフロップの次段にエミッタ接地の飽和形出力増幅器をつなぐことは、反射とか負荷の変動などが、フリップフロップに影響をおよぼすのを防ぐ点では良いが、速度の点で問題がある。ここではフリップフロップの一部として、出力増幅器を組み込むことにより、反転の際の正帰還電流を同時に利用し、ノイズに対する保護作用をもたせ、全遅延時間 12 ns で、200 mA の電流を切り換えることができる。

クロック系統については、主クロック発生器でつくられた信号は、信号の立ち上りの位相だけを決め、各系統に分配された後、クロック増幅器に入り、そこで初めて必要なパルス幅 (8 ns) がきめられる。各系統につき線長を一定にし、0.5 ns ステップの位相補正回路を用い、さらにこの信号を線路に送り込むのに特殊な回路を用いるなどにより、位相およびパルス幅の変動を最小にしている。(向井 誠)

E-16. パターン認識における合成アルゴリズムの実験的考察

R.L. Mattson, O. Firschein and M. Fischler: An Experimental Investigation of a Class of Pattern Recognition Synthesis Algorithms. [IEEE Trans. EC 12, June 1963, pp. 300~306]

ニューロン素子を使った学習では、二元ベクトルをパターンとし、各成分に適当な重みをかけてその和をとったものが、閾値 T を越えるか否かでそのパターンの応答を 1 か 0 とする。この手順の中で、知識の貯蔵としての重みの変え方、閾値のかえ方の規則は今日までに幾つも提案されており、本論文では後述の二通りの方法について、パターンに雑音が入った時の正答率、学習に必要な重みの数等を計算機で計算したものを比較してある。

結論は、学習するパターンの数が少ない時、正答率の高い多数法 (Many Weights Method) が優れ、パターン数が増すにつれて、所望の正答率を得るに必要な重みを計算できる点で、小数法が優ること、および雑音が $x\%$ 含まれるパターンで学習した場合、テスト・パターンの雑音が $x\%$ のとき最も高い正答率を示すこと、したがって雑音の含まれ方が一様でないパターン集合のテストは固定閾値を用いる限り効果に限度があることなどである。

ここで多数法とは漸進的な学習ではなく、全入力を一覧表として眺め、出力 1 となるべきベクトルの第 i 成分の重みを、その成分の 1 の数と 0 の数との差で決め、閾値は 0 となるベクトルの最大値と 1 となるベクトルの最小値の相加平均をとっている。小数法では、ベクトルの値の大小順にパターンをならべたとき、1 となるべきものが閾値の下に落ちていれば、そのベクトルの 1 に相当する成分の重みを増すという操作をくりかえすことで、ベクトル集合を二つに分ける方法である。どちらも分割可能のパターン集合である限り収束する。(伊達 惇)

E-17. 実時間で人工衛星のデータを処理する研究所

R.H. Spittler and B.K. Kersey: A Research Laboratory for Processing and Displaying Satellite Data in Real Time. [AFIPS Conf. Proc. Vol. 23, 1963 SJCC, pp. 117~126]

Lockheed Missiles and Space Company で開発中の軍事用人工衛星組織のためにはリアルタイムのデータ処理組織が必要であるが、実際に世界的な指令制御組織を作る前に追跡センターと遠方監視所の動作についてその各部を正確にシミュレーションする研究所を作ることになった。このために特に新しい装置の開発は行わず当時使えるもので構成され 90 日で稼動に入った。高価な計算機を遊ばしておくわけにいかないで、研究所は研究中のシミュレーションをやる以外に、普通の計算機として計算を行なう。IBM 7090 計算機が使用されこれに、IBM 7281-II データ通信チャンネルがはいっており各種の入出力装置がつながる。入力装置としては Ampex FR 700 磁気テープ (FM で信号が入っており、デジタル信号に直して計算機に入れる)、Ampex FR 100 デジタル磁気テープ、通信シミュレータ、IBM 729 磁気テープを使用する。出力装置としては General Dynamics の表示装置、

IBM の表示装置, 地上局よりの PAM 信号のアナログ表示装置があり, 他に IBM 1401 が付いている. これらの入力出装置は必要に応じて任意の組み合わせをとって各種のシュミレーションを行なう. 計算機本体, 各種入出力装置の詳細がくわしく述べられている. 実際に仕事を行なう前後には診断を行なって全組織が満足に動作しているかどうか, また悪い時にはどの部分が悪いのか, その部分でもどのユニットが悪いのかを

調べる. 診断には静的診断と動的診断とがある. 静的診断では既知のデータを入れて計算機でチェックする. 若しおかしい場合には, 各部分についてそれぞれ調べる. 動的診断では, 結果の解っているシュミレーション・データを入れて調べることができる. また各部が大きな誤りをしていないか常時チェックする機能がある. 現在研究所は以上に述べた線にそって満足に動作しているとのことである. (神田泰典)

書評

Programming and Coding Digital Computers

Phillip M. Sherman (Bell): John Wiley & Sons, New York, 1963, XiV+444 pp., \$ 11.00

計算機とプログラミングの入門書としては, たとえば和訳もた McCracken, Weiss, Lee: Programming Business Computers [John Wiley, 1959] があった. しかしその後の4年間の進歩は, もはや機械語コーディングを非現実的なものとしてしまった. そこから計算機の勉強をはじめた人は, 実際のプログラムにあたってむしろとまどうだろう.

Sherman の著は, 中規模・1アドレス・2進法の計算組織を想定し, アセンブラをもちいてプログラミングを解説し, また関連あるいろいろな話題を広くとりあげた. 現代的な計算組織への一般的な入門書, 教科書としては好適なものであろう.

第1部 問題解決とプログラミング (80頁) では計算機の役割について概説する.

第2部 コーディング (146頁) はアセンブラと入出力モニタとをそなえた架空の計算機 DELTA 63 を想定し, 計算・判断・ループ・サブルーチン・入出力

のコーディングを具体的に説明する. コーディング例は断片的ではあるが豊富である. すべてアセンブラで書かれ, 印刷装置 (IBM 1403?) で印刷されているので実感がでる.

この計算組織は, ハードウェア, ソフトウェアとも IBM 7090 にきわめて似ているが, 中規模のものであるとされている. 中規模の計算組織を設定したのはおそらく意味があるので, つまり大型の計算組織ではコンパイラ, モニタが表面に立つのが現実の姿で, それでは「計算機」の説明ができなくなるおそれがある. しかし1アドレス命令による計算実行の概念は, たとえ実際の仕事ではつかわないとしても, 教養としてだけでも知っておく必要があるというものだ.

だから, 初心者のためには中型の計算組織をとりあげるほうが親切かもしれない. なお磁気テープなどのファイルの概念にはほとんど触れていない.

第3部 コーディング技術と言語 (191頁) はこの本の半分近いページを占め, 算法言語 (FORTRAN, ALGOL), 非数値問題 (リスト), データ処理 (COBOL), プログラマ・マクロ, 通訳ルーチン, 手直しと検査などの章がある. いずれも深くは書いてないが, まんべんなくあつまっているので, 計算機の仕事のどのような分野に入る人でも一応の予備知識をここから得て準備しておくことができるだろう. (西村恕彦)

雑報

ICC の教師養成国際セミナー

ローマの ICC (国際計数センター) は, データ処理部門の教師を養成する国際セミナーを本年末に開くので, その受講者を募集している. その目的は, 受講後,

同センターと連絡を保ちながら, 本国において, 同一主旨のセミナーを開く能力を与えることにある.

プログラムの詳細はまだきまっていないが, 講習期間6カ月 (うち, 4カ月は講義, 2カ月は実習) 講義は英語で行なわれる.

詳細は, 本学会事務局にお問い合わせのこと.