

数字式電子計算機 Fujic

伊 藤 宏*

Fujic は富士写真フィルム株式会社小田原工場で設計並びに製作が行われた計算機で、科学的用途に適した万能型数字式電子計算機として日本最初のものであります。明後年気象庁に入ることになった I. B. M. 704 に較べますと、Fujic は遥かに小規模であり、それだけ機能も単純です。然し数字式電子計算機としての基本的な構造には変わりはないので、ここでは Fujic についての簡単な解説を試みました。将来入る I. B. M. 704 の予備知識として少しでも役に立てば幸いです。

1. 数字式電子計算機の概要

計算機には、数値を数値のまま取り扱えるものと、数値を軸の回転角や線分の長さ等の物理量に転換して計算するものがあります。前者を数字式 (digital) といい、後者をアナログ式 (analogue) といいます。Fujic は前者に属する計算機です。

電子計算機は真空管その他の電子回路を利用した計算機であります。機械式計算機やリレー計算機では機械的な動作に基づいて計算が行われますが、電子計算機の主要部分にはその様に動く所がないので、非常に早く計算が行われるのが第 1 の特徴であります。

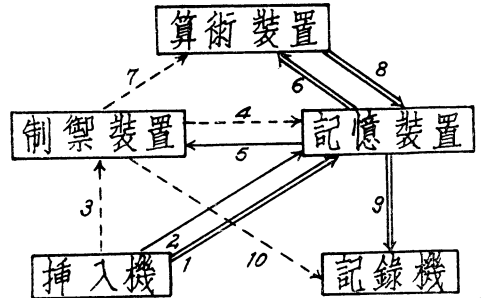
数字式自動計算機は、通常、次の 5 つの部分から出来ています。

- 算術装置 加減乗除の計算を行う他に、新しい命令を作る様な論理的な動きも行う。
- 記憶装置 数値、命令等をしまっておく装置で、必要な時に取り出すことが出来る。位置番号 (address) がつけられており、それにより命令、或いは数値を指定できる。
- 制御装置 制御装置自身をも含めた計算機全体の動作を制御する。
- 挿入機 問題を器械の中に入れる装置。
- 記録機 結果を取り出す装置。

これ等 5 つの装置の機能の相対的な関係が第 1 図に示されています。すなわち、最初、挿入機から計算に必要な幾つかの数値 (矢印 1) や命令 (2) が記憶装置に送りこまれる。次いで挿入機から制御装置へ計算開始の合図 (3) が送られる。制御装置は記憶装置に合図 (4) を送り、記憶装置から取出された最初の命令が、制御装置に伝えられる (5)。

この命令が、例えば $A \times B = C$ の計算を命ずるものであると、制御装置は記憶装置に合図 (4) を送って、A, B の数を取り出しこれが算術装置に送られる (6)。次いで (7) に

より、算術装置中で掛け算が行われ、その結果は (8) を通って記憶装置の中に入れられる。制御装置は直ちに次の命令を取り出し、同様にしてこの命令が行われる。以下同様にして、次々と計算が進められる。



— は数 — は命令 ° …… は信号

第 1 図

数字式自動計算機

結果を記録することを命ずる命令の場合には、記録機に送られた結果 (9) が、合図 (10) により、自動タイプライター、又はその他の方法で記録される。

上に述べた様に、計算の仕方、すなわち命令や計算する数値を記憶することが出来るので、問題を入れると、後は自動的に計算が遂行される。更に制御装置は計算の途中で出た結果によって、計算をどの様に進めるか等の判断をすることが出来る。従ってこの種の器械は人間の頭脳に類似した機能を有しているわけです。

2. Fujic の性能

取扱う数値……他の計算機と同じ様に、Fujic も数値は計算機の中では 2 進法で取扱われる。数値の取り得る絶対値の桁数は 2 進法で 32 桁 (10 進法で 9 桁) である。その他に符号を表示することができる。従って数値 (語) は計算機の中では 33 bits で表示される (符号は 1 bit)。

取扱う数値の範囲は $+16$ と -16 の間である。従って与えられた data がその範囲を越えている時は、あらかじめ適当な常数で割った数値を機械の中に入れる必要がある。記憶装置……水銀柱を用いた遅延回路で、水銀柱は 8 本ある。一本の水銀柱は 32 語を記憶するが、位置番号 '0' は命令中でアキュミュレーターを指定するのに使うので、この装置の容量は

$$8 \times 32 - 1 = 255 \text{ 語}$$

である。前にも書いた様に、記憶装置には位置番号がつけられております。この場合には 1 番から 255 番までの

* 気象研究所 予報研究部

位置番号をもっているわけです。

挿入機及び記録機……数値或いは命令はカードにパンチされ、それを挿入機（読取機）にかけて記憶装置に入れます。読取り速度は毎秒25語位です。従って記憶装置全部に読み込ませる場合でも10秒強で終るわけです。計算結果は電動タイプライターにより印刷されます。その速さは毎秒約10字です。

計算速度……四則演算に要する時間は

加減算	0.0001秒
掛け算	0.0016秒
割り算	0.0021秒

です。一つの命令としての計算時間はこの他に平均約0.002秒加わります。それは命令の中で指定された数を、その入っている記憶装置から算術装置に移し、更に演算の結果を指定された address の記憶装置に入れるに要する時間が加わるからです。この様な時間を待ち時間 (access time) といっています。

一つの演算命令の計算速度はリレー計算機に較べますと数百倍の速さです。格子点が12×14で、time step を1時間にし、中間結果を4時間毎に印刷して、バロトロピックモデルの24時間予報が僅か1時間30分でできます。

さて、一般に電子計算機では、計算機中の計算速度は非常に速いのですが、数値を入れたり、又結果の記録は機械的動作によりますので、演算速度に較べて、挿入又は記録の速度はかなり低速です。一方リレー計算機では読込み速度と演算速度が同程度です。それでリレー計算機では、命令は逐次テープから読込み、それにより演算を行ってゆきます。然しその様な方式では、電子計算機の場合には高速の演算速度が無駄になりますので、電子計算機の場合には命令も記憶装置の中に入れられ、その命令により演算が行われます。

3. Fujic の命令及びプログラミング

命令 (instruction) は操作 (operation) をあらわす部分と記憶の位置番号 (address) をあらわす部分とに分かれています。Fujic の命令は、3つの位置番号を表示します。演算命令ならば、この3つの位置番号は夫々、被演算数、演算数の入っている address 及び答が入る address を表わしています。操作 (例えば寄せ算とか移動) も数であらわします。例えば70は寄せ算操作の一つです。さてこの機械では命令を筆記する時は、0, 1, 2, ……9, J, K, L, M, N, P を数字とした16進法を使います。記憶装置の容量は255箇 (=16×16-1) ですから16進法の数字2字で表わせます。操作をあらわす数も皆2桁にしてありますので、命令記号は8字であらわします。機械の中では2進法ですから丁度32桁になるわけです。

命令記号の一例を次に示します。

位置番号	命令記号
K 1	70 12 2J M1

命令記号の最初の70が操作をあらわし12, 2J, M1 が3つの位置番号です。この命令の意味は位置番号12の内容 (これを記号的に[C]で表わします) と2Jの内容を加え、更にそれにアキュミュレーターの内容 ([C]) を加えた結果をM1に記憶せよということです。これを記号的に

$$[C] + [12] + [2J] \rightarrow M1$$

であらわします。命令の頭につけた K1 はこの命令が記憶される場所です。

命令は8字の数字であらわされていますので、見方をかえれば、この命令は単に70122JM1という8桁の数とみなせます。従って命令と命令を加えたり、引いたりすることが出来ます。従ってその様な操作により、最初与えられた命令を次々と変更してゆくことが可能です。又この事については後で述べることにします。

この小論の末尾に附録1としてFujicの命令の一覧表をのせましたから、それによってどんな言葉、すなわち命令をもっているかを見て下さい。704になりますとこれの何倍もの命令をもっています。

上に示した様な命令記号によって書かれた、機械の遂行する操作を指定した命令系列をプログラムといい、それを作ることをプログラミングと云います。既に書きまされた様に電子計算機では命令は記憶装置の中に入れられるので、プログラミングに際しては命令の格納される位置番号を指定します。又命令を遂行する順番は“飛越し命令”によって特に指示する場合を除き、通常位置番号の順に自動的に行われます。

極く簡単な例についてプログラムの作り方を説明してみましよう。

〔問題〕 $a-b > 0$ の場合 $a-b+c$ を
 $a-b < 0$ の場合 $a-b+d$ を求めよ。

a, b, c, d は夫々位置番号01, 02, 03, 04に入っているとします。又答を05に入れることにします。又命令は10からはじめることにします。

位置番号	命令	内容
10	50, 01, 02, 05	$a-b$ を計算し、その答を一時的に05に入れる。
11	1k 05, 12, 14	$a-b > 0$ ならば、次の命令は12へ、 $a-b < 0$ ならば次の命令は14へ移れ
12	70, 05, 03, 05	$a-b+c$ を計算し、その答を05に入れる。
13	1L 15	この命令は15からはじめよ。
14	70, 05, 04, 05	$a-b+d$ を計算しその答を05に入れる。
15	(これに続く計算の命令)	

上に示した様にプログラムは計算の順序に従って忠実に書き下ろせばよいのです。然し、膨大な計算の場合、計算の各段階毎に順次書き下ろしてゆくと命令が大変な数になります。そしてその命令は記憶装置を占有します

から、計算に必要なデータや常数を全部入れることが出来なくなる恐れがあります。然し前に書きました様に一般に自動計算機では器機の中で命令を造りかえることが出来ます。その事により使用済の命令を次の計算段階の命令に作りかえることが出来ます。従って命令変更の技術を用いれば総べて計算段階を逐一書き下ろさずに済みますので命令の数を少なくすることが出来ます。

一つの命令、例えば (50 01 02 05) は、見方を変えれば一つの数とみなせることは既に前に書きました。それで、今例えば (00 01 01 00) という命令*¹ 或いは数を用意しそれを前の命令に加えますと

```

50 01 02 05
+) 00 01 01 00
-----
50 02 03 05
    
```

となり、前の命令と address part の変わった命令が出来たことになります。上の様な操作をさせることは普通の数値の加算と同じです。

すなわち、命令 (50 01 02 05) が位置番号10に入っており、命令 (00 01 01 00) が位置番号50に入っているとします。又アキュムレーターの内容は 0 であるとし、新しい命令は古い命令をおきかえる場合を考えますと命令変更の命令は (70 10 50 10) となります。従って気象の問題でも見られる様に、ある命令系列が何回も循環して使用され、address part が一定の関係で変化してゆく場合には、基本になる命令系列とその命令を変更するための命令を作ることにより全体の計算をすることが出来ます。address part の変更ばかりでなく操作も変えることが出来ることは容易におわりのことと思います。

(命令変更を用いた簡単な例題が附録Ⅱにあります)

問題が複雑な場合には、命令記号に書き下ろす前に、

〔附 録 Ⅰ〕

取扱う演算等の種類と命令記号

種 類	命 令 記 号	命 令 の 内 容
寄 せ 算	70/LL/MM/NN	$([C]) + [LL] + [MM] \rightarrow NN$
	60/LL/MM/NN	$([C]) - [LL] + [MM] \rightarrow NN$
引 き 算	50/LL/MM/NN	$([C]) + [LL] - [MM] \rightarrow NN$
	40/LL/MM/NN	$([C]) - [LL] - [MM] \rightarrow NN$
掛 け 算	20/LL/MM/NN	$([C]) + [LL] \times [MM] \rightarrow NN$
	15/LL/MM/NN	$([C]) - [LL] \times [MM] \rightarrow NN$
	10/LL/MM/NN	$[C] \times [LL] \times [MM] \rightarrow NN$
	16/LL/MM	$[C] \times [LL] \rightarrow [MM]$
割 り 算	30/LL/MM/NN	$\frac{([C]) + [LL]}{[MM]} \rightarrow NN$
移 動	19/LL/NN/NN	$[LL] \rightarrow NN$
	1L/LL/-/-	無条件：次はLLの命令へ移れ

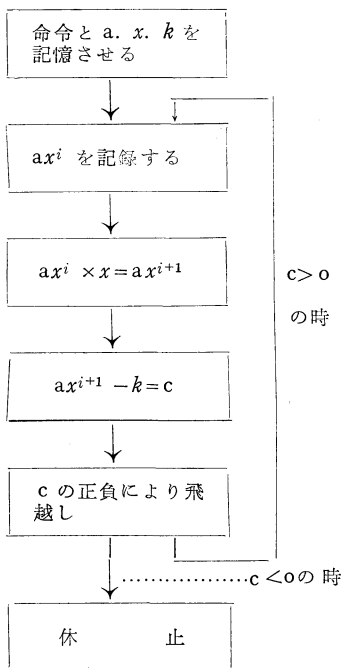
* (00 01 01 00) は動作を止めよという命令に使います

計算の過程を分析し、それを図式に表わします。その様な図の事を flow-diagram といいます。今、簡単な問題について flow-diagram の一例をあげましょう。

〔問題〕 a, x, k ($16 > a > k > 0, 1 > x > 0$) を与えて、 $a, ax, ax^2 \dots$ を k より大きい間、次々に記録せよ。

この問題の flow-diagram の 1 例は第 2 図の様なものです。

第 2 図 flow-diagram の一例



一般に、まず flow-diagram を作りそれにもとづいて命令記号に書き下ろすのがプログラミングの普通の進め方です。

飛越し	1K/LL/MM/NN	条件付：〔LL〕が正ならMM， 負ならNNの命令へ移れ
	1J/LL/MM/-	〔C〕 が16を越えた時はMM， 然らざる時はNNの命令へ移れ
記録	1M/LL/MM/NN	〔LL〕を十進法で記録せよ
	1N/LL/MM/NN	〔LL〕をそのまま記録せよ
挿入	IP/-/-/-	カードを読め
休止	00/-/-/-	動作を止めよ
挿入の際カード のパンチから受 ける指定	LL S _mをそのままLLへ
	LL PB.....を二進法にしてLLへ
	LL HZ.....〔LL〕の命令から始めよ

註：LL MM NN は位置番号 Cはアキュムレータ
〔LL〕は LL の内容

〔附 録 2 〕

命令変更を用いたプログラムの一例。

〔問題〕 2つのn次元ベクトルAとBのスカラー積を求め、答を位置番号Sにしまい。すなわち、2つのベクトルA、Bの成分を夫々(a₁ a₂.....a_n)及び(b₁b₂.....b_n)とすると、

$$A \cdot B = a_1 \times b_1 + a_2 \times b_2 + \dots + a_n \times b_n$$

を計算せよ。

今、a₁, a₂,a_n 及び b₁, b₂.....b_n は夫々位置番号A, A+1,A+(n-1), 及びB, B+1.....B+(n-1)に入っているとします。その他に

	(位置番号)		(命令記号)	
	E ₁	00	01	01 00
	E ₂	20	A+(n-1)	B+(n-2) S
	S	00	00	00 00

を所定の記憶に入れておきます。問題に対するプログラムは

	位置番号	命令記号
	→ I	70 I+2 E ₁ I+2
Start	→ I+1	19 S 00
	I+2	20 A B S
	I+3	50 E ₂ I+2 W
	→ I+4	IK W I I+5
	→ I+5	(次の計算の命令)

となります。

〔説明〕 最初I+1にある命令から出発させたとします。Iにある命令は〔S〕=0をアキュムレータに移します。次の命令は〔C〕+〔A〕×〔B〕→Sですから先づa₁×b₁がSに入ります。この命令が終わった状態ではアキュムレータは0になります。I+3, I+4, の命令

は必要な回数だけ繰返しを行わせるためのものです。

先づI+3にある命令により〔E₂〕-〔I+2〕が計算され、結果がWに入ります。さて〔E₂〕-〔I+2〕は今の段階では

$$\begin{array}{r} 20 \quad A+(n-1) \quad B+(n-2) \quad S \\ -) \quad 20 \quad \quad \quad A \quad \quad \quad B \quad \quad \quad S \\ \hline 00 \quad \quad \quad n-1 \quad \quad \quad n-2 \quad \quad \quad 00^* \end{array}$$

となり、これは正の数です。次の命令は条件付飛越えです。すなわち今の場合は〔W〕が正ですから次にはIにある命令が遂行されます。この命令はI+2にある命令を変更させるためのもので、すなわち

$$\begin{array}{r} [I+2] = 20 \quad A \quad B \quad S \\ +) [E] = 00 \quad 01 \quad 01 \quad 00 \\ \hline 20 \quad A+1 \quad B+1 \quad S \rightarrow I+2 \end{array}$$

が行われます。次はI+1の命令です。今度は〔S〕=a₁×b₁になっておりますから、アキュムレータにはa₁×b₁が入ります。次の命令は変更をうけて(20, A+1, B+1, S) になっていますから、この命令によりa₁×b₁+a₂×b₂がSに入ることとなります。次は前と同様にIに帰る命令です。上にのべた繰返しをn回やればI+2にある命令は(20, A+(n-1) B+(n-1), S)となりSにはa₁b₁+a₂b₂+.....a_n×b_nが入ります。そしてI+3の命令でWの内容は

$$\begin{array}{r} 20 \quad A+(n-1) \quad B+(n-2) \quad S \\ +) 20 \quad A+(n-1) \quad B+(n-1) \quad S \\ \hline - \quad 00 \quad \quad \quad 00 \quad \quad \quad 01 \quad \quad \quad 00 \end{array}$$

となります。すなわち〔W〕<0になりましたからI+4の命令により、その次の命令はI+5に移り、このスカラー積の計算は終わって次の計算の命令に移ります。

* これは n=10とすれば 00 09 08 00 を意味する