

制御用電子計算機 HITAC 501, 502 について

Electronic Computers for Controlling Use, HITAC 501 and 502

三 卷 達 夫* 佐 藤 昭 治*
Tatsuo Mitsumaki Shōji Sato

中 田 育 男* 池 田 隆 一*
Ikuo Nakata Ryūichi Ikeda

内 容 梗 概

工業用のデータ処理装置用として設計され、特にオンラインとしての使用に耐えるよう考慮された制御用計算機 HITAC 501、またコアメモリによる演算速度の向上、オーダシステムの完備、インデックスレジスタによるプログラムの能率化、さらに入出力の制御信号など、制御用としての機動性を備えた HITAC 502 について、その装置の概略、およびプログラムの詳細を述べる。

1. 緒 言

最近電子計算機の産業面への応用として、計算機の頭脳としての機能を生かし、プラントの操業の助けとなるデータの測定、計算処理に、またプラントの制御、たとえば化学プラントの運転状態を監視し、最も適した状態にもしくは最大の利益が得られるように制御したり、また発電所の複雑な過程をシーケンス制御することが行なわれ出している。

日立製作所の制御用計算機としては HITAC 501, 502 の 2 機種あるが、これらはいずれも全固体素子化され現場での使用に耐えるよう製作されたものである。501 は発電所のデータロガーとしてすでに数台納入の実績を持っており、現在調製中、または製作中のものを含め十数台を数えている。502 はこれの姉妹機ともいべきもので、磁気コア記憶装置により高速化を図ったものである。命令類も比較的豊富に準備され、たとえば最適制御に必要な複雑な科学計算にも便利に使用できるようになっている。

ここではこの両者を比較しつつその特長について述べる。装置についてはすでに報告されてもいるので⁽¹⁾⁽²⁾概略にとどめ、本文では主としてプログラムの面に重きをおいて説明する。

2. HITAC 501, 502 の概要

第 1, 2 図に HITAC 501 および 502 の外観を示す。また第 1 表に計算機本体の概略仕様を示す。実際にプラントと連結して用いる場合には、これにアナログ量を受け渡しするための入出力部が付属しているが、これらは使用する場合により、入力点数、処理速度、要求精度などが異なる。一般には低速度のものに対してはリレー切り替え、高速度のものに対しては純電子的な切り替えにより種々の構成が可能である。

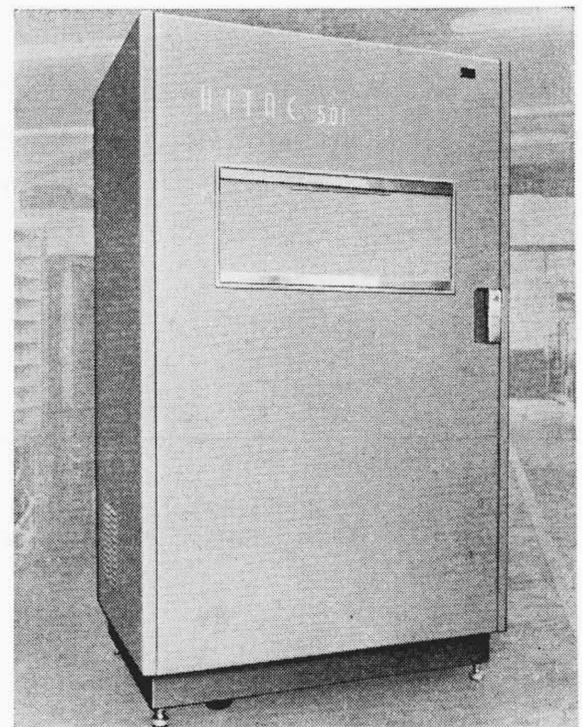
また制御あるいはコンピューティングロガー用として用いる場合には通常の計算機以上に信頼性が問題である。本機は内部 2 進の構成をとり部品点数の減少を図るとともに、各種誤動作のチェック、停電した場合の計算内容の消失を防ぐ手段など種々の機能を備えており、また外部装置と計算機が情報、指令の受け渡しに融通性をもって行なえるような構成をとっている。第 2 表はこれら 2 機種種の HISIP による命令表を示している。

3. HITAC 501, 502 の特長

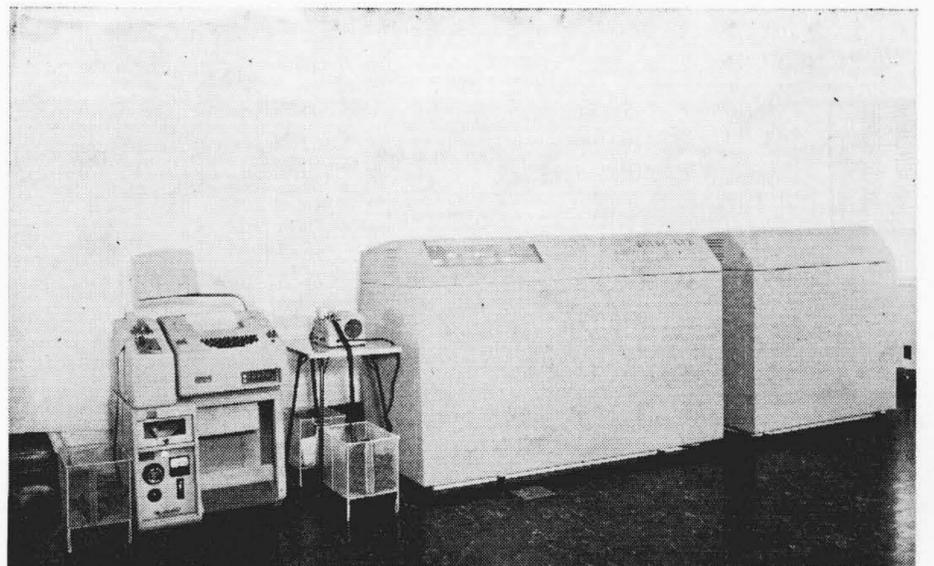
制御用として、オンラインで実時間演算を行なうには計算機の処理能力が所要速度を下まわらないことが必要である。計算速度がまにあわないということは制御の場合には致命的である。

HITAC 501 では価格の低減という点より磁気コアを有せず磁気

* 日立製作所中央研究所



第 1 図 HITAC 501 外 観



第 2 図 HITAC 502 (右端はアナログ信号入出力部)

ドラムのみであるが演算速度を早めるための工夫がなされている。ドラムのみときはアドレス方式も 2-アドレスを採用するか、または命令実行後指定により次の命令を読むようなことがよく行なわれる。ここでは装置を簡単にするため 1-アドレス方式を用い、アドレスが 3 番地飛んで、4 番地ごとにドラム上で割りふられている。これは特殊な命令を除き一般の命令は 3 語時間 (3 word time) で終わるものが多いので、このようにしておくとも命令実行が終わったころ、次の命令が読まれることになり、いっぽうプログラムも従来と全く同じように書け、速度の向上と使用の便が得られる。

HITAC 502 は磁気コア 192 語を有し、ドラムとの情報の転送がトラックごとにブロックとしてできるので、501 よりもさらに高速

な演算処理が可能である。また豊富な命令を有しており、かなりの規模の科学計算もこなせる能力をもっている。この命令の中には、入出力情報、入出力制御信号、外部装置の遅れをカバーする命令などもあり、これらを組み合わせて割り込み演算が可能である。また 2 個のインデックスレジスタを有しているためプログラムが容易であり、また記憶装置の容量 8,000 語とあいまって、種々の関数サブルーチン、倍精度演算、浮動小数点演算のサブルーチンも完備しているので制御用のみならず大学、研究所などにおける研究用としても便利に使用できる。

以下本文ではこれら 2 機種種のプログラムについて述べる。

4. HITAC 501, 502 のプログラム

HITAC 501 は、すでに述べたようにおもにデータ・ロガーとしての実績を持っているが、プログラム面でも特にそれらを考慮に入れた整備ができています。数値の入力ならびに出力プログラム(注1)、プログラムミスの分析に有用なプログラム、すなわちデバッグ・プログラム(注2)(debugging program)がそれである。

ここでは、それらの中から HISIP 501 とデバッグ・プログラムについてその使用法を中心に概略を記述する。

HITAC 502 は制御用として、より高度の演算制御、たとえば最高効率運転などを目標として設計されており、そのため倍精度演算

プログラム、また科学計算用としても小形機として便利に使用できるよう浮動小数点演算プログラムなどが整備されているが、以下これらにつき簡単に述べる。

4.1 HISIP 501 および HISIP 502

HITAC-502 は高速記憶装置(コア・メモリ)を 192 語持っているから、HISIP 502 をその範囲内に収める(HISIP 502 は、ふだんはドラムに格納してあるが、実際に使用するときはコアに取り出して使う)ことにより高速読み込みが可能になるようにした。したがって、短い入力プログラムにするために制御指令は必要最小限のものにとどめている。

HITAC 501 では HITAC-502 のような条件はないが、HISIP 502 と同様制御指令を最小限にとどめた。大きな相違点は、HISIP 501 には記号番地を許していることである。両者とも、命令語は HISIP 語に準拠し、一般に

$$O_p/A$$

と表現する。ここに O_p は演算部を表わし、第 2 表(その 1 およびその 2)で示した命令コードを用いて書く。

A は番地部を表わし、これにはインデックスによる番地部の修飾が可能か、あるいは記号番地が使用できるかにより表現が異なる。演算部と番地部は“/”(slant)を入れて区別し、必ず入れるものと約束する。

HISIP 501 では A (番地部)には

- (1) 絶対番地 (0~4095 の整数)
- (2) 記号番地 (NP**)
- (3) M による修飾
- (4) L による修飾

およびこれらを組み合わせた書き方が可能である。記号番地は NP

第 1 表 制御用計算機概略仕様

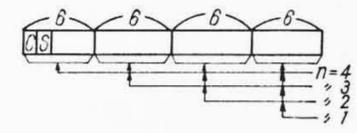
計 算 機 形 名		HITAC-501	HITAC-502
演算方式	回路基本パルス直並列の区分	ダイナミック逐次演算方式 クロック周波数 210 kc 直列同期方式	ダイナミック+スタティック 220 kc 逐次演算方式 直列同期方式
プログラム	内部記憶アドレス方式 命令語の形式 命令の種類	内部記憶 1 A 12+F 8 33	内部記憶 1½ F 6+M 3+A 13 61
数値表現	ビット形式 数値語の表現 小数点形式 文字の表現	純 2 進 {符号+絶対値19けた+チェックビット2けた 固定 6 ビット	純 2 進 {符号+絶対値22けた+チェックビット1けた 固定 6 ビット
演算速度	加 減 算 乗 算 除 算	0.21 ms 4.6 ms 4.6 ms (アクセス時間を含む)	0.36 ms 6 ms 6 ms (コアのアクセス時間を含む)
主記憶装置	容量 待 時 間	磁気ドラム(9,000 rpm) 4,096 語 平均 3.3 ms	磁気ドラム(8,000 rpm) 7,936 語 192 語 平均 3.7 ms 0.11 ms
	インデックス・レジスタ	なし	13 けた 2 ケ
	チェック方式	4n+3 チェック	奇数パリティチェック
紙入出力装置	テープ方式 テープ読取器 テープせん孔機 さん孔タイプライタ	7 単位 PTR 200 字/s	7 単位 MTR500字/m PTR200字/s 500字/m 500字/m
その他拡張可能装置		電動タイプライタ 1台 A-D 変換器 1台 D-A 変換器 1台	電動タイプライタ 1台 A-D 変換器 10台 D-A 変換器 10台
装置の大きさ重量		本 体 1,110×800×1,897 (mm) 約 800 kg 床 面 積 約 7m ²	本 体 1,900×780×1,100 (mm) 約 600 kg 床 面 積 約 11m ²
電 源		本 体 1φ 100V 50/60 cps 600VA 磁気ドラム用 3φ 200V 50/60 cps 400VA	MG 入 力 3φ 200V 50/60 cps 5kVA 出力(1)1φ 100V 50/60 cps (2)3φ 100V 150/180 cps

(注 1) 初期入力プログラム I, チェック付入力プログラム, 16 進コード入力プログラム, HISIP 501, 小数形および整数形印字プログラム, 16 進コード出力プログラムなど。
(注 2) チェック・プログラム, 追跡プログラム(tracer), 飛越追跡プログラム(jump tracer), 番地検索プログラム, 検屍プログラム(post-mortem), 命令読み出しプログラムなど。

第 2 表(その 1) HITAC 501 命令表

種 類	Order	Operation	Note
Add Sub	XA	Clear add	(n)→Acc-U
	A	Add upper	(Acc-U)+(n)→Acc-U
	XB	Clear Sub	-(n)→Acc-U
	B	Sub upper	(Acc-U)-(n)→Acc-U
Mult Div	XMA	Multiply	(MD)•(n)→Acc
	MA	Multiply add	(Acc)+(MD)•(n)→Acc
	XMB	Multiply negative	-(MD)•(n)→Acc
	MB	Multiply sub	(Acc)-(MD)•(n)→Acc
	XAD	Divide	(n)/(MD)→Acc-U
	AD	Add Divide	{(Acc)+(n)}/(MD)→Acc-U
Add Sub address parts	XAA	Clear add address	n•2 ⁻¹¹ →Acc-U
	AA	Add address	n•2 ⁻¹¹ +(Acc-U)→Acc-U
	XBA	Clear Sub address	-n•2 ⁻¹¹ →Acc-U
	BA	Sub address	-n•2 ⁻¹¹ +(Acc-U)→Acc-U
Jump	J	Jump unconditionally	n へとべ
	JP	Jump (Acc) ≥ 0	(Acc) ≥ 0 なら n へとべ
	JM	Jump (Acc) < 0	(Acc) < 0 なら n へとべ
Shift	SR	Shift right	(Acc) n けた右に移動
	SL	Shift left	(Acc) n けた左に移動
Store Load	T	Store upper	(Acc-U)→n
	XT	Clear and store	0→Acc, 0→n
	LMD	Load MD	(n)→MD
Input Output	ON	Type	(Acc-U) n けたタイプ, 1 ≤ n ≤ 5
	OS	Type Code	n コード(指定)をタイプ
	RH	Tape read	テープから n けた Acc-U に読み込む
	XRH	Clear and Tape read	Acc-U を Clear 後, Tape read
Skip	SWS	Skip Switch set	n=1~10, n という入力ハブを set する
	SWR	Skip Switch reset	n のハブを reset する
	ST	Skip test	{n という入力ハブが on あれば次の命令を 1 つ skip する
A-D D-A	TAD	Convert A-D	{A-D 変換が終了しているか否かを調べる
	TDA	Convert D-A	(Acc-U)→D-A 変換器

第2表 (その2) HITAC 502 命令表

種類	Order	Operation	Note	種類	Order	Operation	Note	
Add Sub	XAU	Clear add upper	$(n) \rightarrow \text{Acc-U}$	Index Register	I 1 B	Sub IR 1	$(\text{IR } 1) - n \rightarrow (\text{IR } 1)$	
	XAL	Clear add lower	$(n) \rightarrow \text{Acc-L}$		I 2 B	Sub IR 2	$(\text{IR } 2) - n \rightarrow (\text{IR } 2)$	
	AU	Add upper	$(\text{Acc-U}) + (n) \rightarrow \text{Acc-U}$		J 1 Z	Jump $(\text{IR } 1) = 0$	$(\text{IR } 1) = 0$ なら n へとべ	
	AL	Add lower	$(\text{Acc-L}) + (n) \rightarrow \text{Acc-L}$		J 2 Z	Jump $(\text{IR } 2) = 0$	$(\text{IR } 2) = 0$ なら n へとべ	
	XBU	Clear sub upper	$-(n) \rightarrow \text{Acc-U}$		S I 1	IR 1 Skip	$(\text{IR } 1) = n$ なら次の命令を一つ Skip する	
	XBL	Clear sub lower	$-(n) \rightarrow \text{Acc-L}$		S I 2	IR 2 Skip	$(\text{IR } 2) = n$ なら次の命令を一つ Skip する	
	BU	Sub upper	$(\text{Acc-U}) - (n) \rightarrow \text{Acc-U}$		T I 1	Store IR 1-Order	n に Clear add IR1/0 + $(\text{IR } 1)$ なる命令を store する	
	BL	Sub lower	$(\text{Acc-L}) - (n) \rightarrow \text{Acc-L}$		T I 2	Store IR 2-Order	n に Clear add IR2/0 + $(\text{IR } 2)$ なる命令を store する	
Mult Div	XMA	Clear multi	$(\text{MD}) \cdot (n) \rightarrow \text{Acc}$	Block Transfer	ATD	Core A to Drum	n は Drum の番地の指定のみで、これらの命令で常に n を含むバンドの 64 語の transfer が行なわれる (Core A: 8000~8063 番地) (Core B: 8064~8127 番地) (Core C: 8128~8191 番地)	
	MA	Multi add	$(\text{Acc}) + (\text{MD}) \cdot (n) \rightarrow \text{Acc}$		BTD	Core B to Drum		
	MB	Multi sub	$(\text{Acc}) - (\text{MD}) \cdot (n) \rightarrow \text{Acc}$		CTD	Core C to Drum		
	D	Div	$(n) / (\text{MD}) \rightarrow \text{Acc-U}$		DTA	Drum to Core A		
Jump	J	Jump unconditionally	n へとべ		DTB	Drum to Core B		
	J P	Jump $(\text{Acc}) \geq 0$	$(\text{Acc}) \geq 0$ なら n へとべ		DTC	Drum to Core C		
	J M	Jump $(\text{Acc}) < 0$	$(\text{Acc}) < 0$ なら n へとべ	Input Output	TR	Tape select	$n=1$ で mechanical tape reader を、 $n=1$ 以外で phototape reader を選ぶ	
	J Z	Jump $(\text{Acc-U}) = 0$	$(\text{Acc-U}) = 0$ なら n へとべ		PCH	Punch select	n は punch あるいは type の machine number, $n=1$ の場合のみ有効	
	J L Z	Jump $(\text{Acc}) = 0$	$(\text{Acc}) = 0$ なら n へとべ		TYP	Type select		
	I 1 J	IR 1 set & Jump	IR 1 に $(\text{scc}) + 1$ を set し n へとべ		RH	Read Character	n けた ($n \leq 4$) Acc-U にはいる。そのはり方は 	
	I 2 J	IR 2 set & Jump	IR 2 に $(\text{scc}) + 1$ を set し n へとべ		OH	Out Character	(Acc-U) を n けた ($n \leq 4$) punch および type する。そのとり方は 	
	HJ	Halt jump	$n \rightarrow (\text{scc}), \text{stop}$ する					
Shift	SR	Shift right	$(\text{Acc}) \times 2^{-n} \rightarrow \text{Acc}$ (numerical)					
	SL	Shift left	$(\text{Acc}) \times 2^n \rightarrow \text{Acc}$ (logical)					
	SC	Shift & count	符号が 1 つ変わる手前まで (Acc) を左に shift し shift したけた数を IR ₂ に加える					
			TU	Store upper				$(\text{Acc-U}) \rightarrow n$
Store, Load	TL	Store lower	$(\text{Acc-L}) \rightarrow n$	Skip	SSW	Switch skip	$n=0 \sim 9$ であって、 n という入力ハブが on であれば次の命令を一つ skip する。 $n=0 \sim 3$ manual 可能 入力ハブ, 出力ハブ } $n=0 \sim 9$ 入力ハブ 入力ハブ, 出力ハブ } $n=10 \sim 39$ 出力ハブ	
	TZ	Store zero	$0 \rightarrow n$		L SW	Switch		
	LMD	Load MD	$(n) \rightarrow \text{MD}$		X SW	Switch reset		
	R	Round	$(\text{Acc}) + 2^{-23} \rightarrow (\text{Acc})$		Delay	DLY	Delay	$n \cdot \text{Tms}$ を経て次の命令を家行する。ただし T はドラムの 1 回転に要する時間 約 7.7 ms
	XEA	Extract	(MD) と (n) との対応するけたの and をとり Acc に入れる					
	TCR	Store CR	$(\text{CR}) \rightarrow n$					
	L CR	Load CR	$(n) \rightarrow \text{CR}$					
	FCR	Free CR	CR レジスタを他のレジスタ表示のために開放する					
Index Register	X 1 A	Clear add IR 1	$n \rightarrow (\text{IR } 1)$					
	X 2 A	Clear add IR 2	$n \rightarrow (\text{IR } 2)$					
	X 1 B	Clear sub IR 1	$-n \rightarrow (\text{IR } 1)$					
	X 2 B	Clear sub IR 2	$-n \rightarrow (\text{IR } 2)$					
	I 1 A	Add IR 1	$(\text{IR } 1) + n \rightarrow (\text{IR } 1)$					
	I 2 A	Add IR 2	$(\text{IR } 2) + n \rightarrow (\text{IR } 2)$					

(Notice Point) の次に 2 けたの数字をつけて表わす。M による修飾はいわゆるアドレス・カウンタによる修飾と同じ意味をもっているものであるが、これはテープ読み込みに際して行なわれる。

たとえば命令語

$$O_p/3M$$

が l 番地に格納されるものであればテープ読み込みが終わったあとの結果は

$$O_p/3+l$$

となる。L による修飾はそのテープの読み込み先頭番地 (LOm の m) を加算することである。たとえば

$$O_p/3L$$

は $O_p/3+m$ となる。これらを組み合わせて使用するときは次のような表わし方をするが、特に順序に意味をもっている。例で示すと

$$O_p/25NP01$$

の番地部は記号番地 NP01 を基準にして 25 番地後の番地を示す。

HISIP 502 では A は

$$x_1 + x_2 \text{ または } x_1 + x_2 + x_3$$

という形に書く。

(1) x_1 は 0~7 の整数で、インデックス・レジスタまたは SCC による修飾部を示す。

(2) x_2 は 0~8191 の整数で番地部を示す。

(3) x_3 はテープの読み込みに際して番地部に修飾を加える部分であり、L, P, Q の 3 文字の任意個数の列からなっている。その修飾内容は次のとおりである。

まず、この修飾を施したい命令語の前にそれぞれ

lL

l_1P

l_2Q

をせん孔しておく必要がある。ここに、 l, l_1, l_2 は 0~8191 の整数である。これらのあとにくる各命令語の x_3 部の個々の L, P, Q に対し、 x_2 の部分 (番地部) にそれぞれ l, l_1, l_2 が加えられる。

HISIP 501 および 502 とともに、数値の扱いは小数形と整数形の二通りある。とり扱える数値の範囲は、HITAC 501 と 502 のビットの長さから決められている。

HISIP 501 で、小数形の数値 x ($-1 \leq x < 1$) を読み込ませるためには

$$\square. [x \times 10^5]$$

の形に書く。ここに $[]$ はガウス記号であり、 \square には x が正のときは P (Plus), 負のときは M (Minus) を代入する。

また、整数値 y ($-2^{19} \leq y < 2^{19}$) を 1 語の右端に小数点を置いた形で読み込ませるためには

$$I \square |y|$$

で表現する。 \square は小数形の場合と同様、P または M を入れる。

I は Integer (整数) の頭文字をとったものである。

例で示すと次のようになる。

$$x = 0.12345 \text{ のときは } P.12345$$

$$x = -0.0123 \text{ のときは } M.01230 \text{ または } M.1230$$

$$y = -2345 \text{ のときは } IM 2345$$

次に HISIP 502 で小数形の数值 x ($-1 \leq x < 1$) を読み込ませるためには

$$[x \times 10^6] \text{ sign } x$$

の形に書けばよい。ここに $\text{sign } x$ は x の付号である。

整数値 y ($-2^{22} \leq y < 2^{22}$) を読み込ませるためには

$$|y| I \text{ sign } y$$

の形に書けばよい。たとえば

$$x = -0.01234 \text{ のときは } 12340-$$

$$y = 523 \text{ のときは } 523 I +$$

と書けばよい。

さらに、HISIP 502 では 8 進法表示の数值語および文字語を読むことができる。これらは、それぞれ

$$\bar{O}CT/x_8 \quad (x_8 \text{ は 8 進法表示の数值})$$

$$BCH/* * * * \quad (1 \text{ 語には 4 文字が入る})$$

の形に書けばよい。

最後に HISIP 501, HISIP 502 を用いて命令および数值語を記憶装置の l 番地から格納するためには、それぞれ

$$LOl \quad (\text{HISIP 501 の表現})$$

$$lL \quad (\text{HISIP 502 の表現})$$

をテープの先頭にせん孔し、その後命令あるいは数值語をせん孔する。命令あるいは数值語を読み込んだあとで m 番地に格納されている命令から演算を開始させたい場合にはテープにそれぞれ

$$LOmZ \quad (\text{HISIP 501 の表現})$$

$$mS \quad (\text{HISIP 502 の表現})$$

をせん孔しておく。

4.2 HITAC 501 のデバッグング・プログラム

一般に機械語またはそれに近い形 (HISIP 程度のものまで) でコーディングを行なう場合、その誤りを見いだすために費される時間は決して少なくない。したがってそれに費される時間をできるだけ短くするような考慮が払われねばならないことは当然である。

デバッグングの方法は一般にダイナミック・デバッグング (Dynamic Debugging) と検屍デバッグング (Post-mortem) の二つに分類することができる。ダイナミック・デバッグングはデバッグングの対象となっているプログラムを進行させながら、そのプログラムをチェックしてゆく方法であり、検屍デバッグングはプログラムを実行させて、停止またはハング・アップ (Hang up) したあとに計算機のメモリーをチェックしてゆくものである。

(a) ダイナミック・デバッグング

この方法にはさらにオートモニタ (Automoniter) 方式とチェックポイント方式の二つがある。

チェックポイント方式はコーディングの際、中間の重要な結果の出る点や、危険な点をあらかじめマークしておき、そこに命令語をそう入して (実際にはサブルーチンへのつなぎの命令になることが多い) 自動的に停止させるか、または印字させることである。HITAC 501 におけるデバッグング・プログラムとしては HISIP 501 に付随して使用するチェック・プログラム (アキュムレイタの内容を印刷する) がある。

オートモニタ方式はチェックポイント方式で要求されたようなコーディングの際の特別な考慮は必要としない。これはチェックすべきプログラムで 1 ステップずつ押ボタンを用いて人間が調べてゆくと同じ方法をオートモニタ・プログラムと呼ばれるプログラムで行なうものである。これには追跡プログラムと飛越追跡プログラムの二つが作成されている。

追跡プログラムは印字する範囲を番地で指定し、その範囲にある命令の実行の際は

(i) チェックの対象となる命令の格納番地

(ii) チェックの対象となる命令

(iii) 命令実行後のアキュムレータ (Acc) の内容 (必要な場合は lower Acc も印字)

(iv) MDレジスタの内容

を自動的に印字する。

飛越追跡プログラムはプログラムのループが正しく進行しているか、あるいは筋道が流れ図どおりであるかを、時間的に早く調べるために考えられたもので、プログラムが停止しないか、あるいは非常に長く続く場合などに用いて有効である。これはチェックすべきプログラムの進行中、ジャンプ命令またはスキップ命令に遭遇したときその格納番地と、その命令と次の実行番地を印字するもので、それによってループの繰り返し回数やプログラムの進行過程が一目でわかる。この追跡プログラムはさらにチェックすべきプログラムの中で指定した数值の番地部 (使用の際、任意の数に指定する) をもつ命令に出会ったとき、格納番地とその命令とその命令が実行されたあとのアキュムレイタの内容を印字するような機能を持たせている。この機能を利用して数值の移動の様子や、中間結果を知ることができる。

(b) 検屍デバッグング

追跡プログラムは一般に費用がかかるばかりでなく、必要とする情報のすべてが常に見やすい形で与えられるとは限らない。たとえば、3,000 ステップ位のプログラムで、ある特定の値の番地部をもつ命令がおかしい、あるいはある番地の命令が変わると見当がついたとき、これを追跡プログラムで追いかけることは、時間が極端にかかるばかりでなく、印字された結果は見ずらいものとなる。このようなとき、まず必要な情報はエラーが生じたあとの結果である。番地検索プログラムや検屍プログラムはこのような場合に役立つものである。

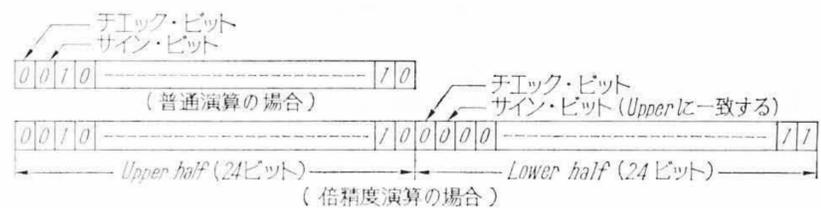
番地検索プログラムは、ある番地以後に格納されている n 個の命令群の中で、指定された値の番地部をもつ命令語のみ拾って、その格納番地とともに、命令の種類を印字するものである。検屍プログラムはデバッグングを要するプログラムが停止、またはハングアップしたあとに、そのプログラムを再度テープ読取器にかけ、記憶装置内にあるプログラムと照合して、変化している点 (命令とその所在番地) を印字するものである。

そのほか、メモリの内容を命令形として印字する命令読み出しプログラムも検屍デバッグング・プログラムとして役立っている。

4.3 HITAC 502 の倍精度、浮動小数点演算プログラム

4.3.1 倍精度演算プログラム

1 数值語を普通演算の 2 数值語すなわち数值部 44 けた、符号部 1 けたからなるものとして計算を進めるプログラムで、乗算、除算、加減算を含む複合演算、平方根、数種の特異関数、倍数けた数值の入力、出力などのプログラムからなっている。



プログラムの使用法について概略を述べると各演算プログラムには普通演算と同趣旨の一定の記号命令 (SIP 語) が与えられており、プログラミングの際これを書いておけばあとは計算機がこの命令を判読して所定の演算プログラムを呼び出し計算を行なうようなくみになっていて、いわゆるインタプリティブ (Interpretive) 方式による処理を行なっている。したがって倍精度演算はとくに意識することなく普通演算とまったく同じ要領でプログラ

第3表(その1) 倍精度演算命令表

記号命令	演算内容	対応する普通演算の命令	演算時間 (アクセス・タイムを含む)
WLMD	$(\bar{n}) \rightarrow \overline{MD}$	LMD	0.1s
WXMA	$(\bar{n}) \times (\overline{MD}) \rightarrow \text{Acc}$	XMA	0.2s
WXMB	$-(\bar{n}) \times (\overline{MD}) \rightarrow \text{Acc}$	XMB	0.2s
WMA	$(\text{Acc}) + (\bar{n}) \times (\overline{MD}) \rightarrow \text{Acc}$	MA	0.2s
WMB	$(\text{Acc}) - (\bar{n}) \times (\overline{MD}) \rightarrow \text{Acc}$	MB	0.2s
WXDA	$(\bar{n}) \div (\overline{MD}) \rightarrow \text{Acc}$	D	0.3s
WXDB	$-(\bar{n}) \div (\overline{MD}) \rightarrow \text{Acc}$	なし	0.3s
WDA	$(\text{Acc}) + (\bar{n}) \div (\overline{MD}) \rightarrow \text{Acc}$	なし	0.3s
WDB	$(\text{Acc}) - (\bar{n}) \div (\overline{MD}) \rightarrow \text{Acc}$	なし	0.3s
WSQ	$\sqrt{(\bar{n})} \rightarrow \text{Acc}$	なし	0.4s
WOM	数値または文字タイプ	なし	
SIN	$\sin x \rightarrow \text{Acc}$	なし	1.6s
COS	$\cos x \rightarrow \text{Acc}$	なし	1.6s
EXP+	$e^x \rightarrow \text{Acc}$	なし	1.7s
EXP-	$e^{-x} \rightarrow \text{Acc}$	なし	1.7s
LOG	$\log_{10} x \rightarrow \text{Acc}$	なし	1.7s
LOG2	$\log_2 x \rightarrow \text{Acc}$	なし	1.7s
LOGE	$\log_e x \rightarrow \text{Acc}$	なし	1.7s
ARCT	$\tan^{-1} x \rightarrow \text{Acc}$	なし	2.3s
ARCS	$\sin^{-1} x \rightarrow \text{Acc}$	なし	1.7s
READ	倍数ケタの数値読込	なし	

- (注) 1. \bar{n} はアドレス $n, n+1$ の2進44けたを続けて1語とみなした場合の1アドレスを示す。たとえば $\overline{10}$ とあれば 10, 11 番地を意味する。
 2. \overline{MD} : 擬似MDレジスタ。普通演算におけるMDレジスタの働きを持つ。
 3. $\sin x$ など関数プログラムの x は Acc 内にあるものとする。

ムを組むことができる。

第3表(その1)は用意された各演算プログラムと記号命令などを列記したもので、表中対応する普通演算の命令がないものは使用ひん度を考慮して倍精度演算プログラム用に特につけ加えたものである。

4.3.2 浮動小数点演算

浮動小数点演算(以下 F_t と略記する)では数値はすべて

$$x = x' \times 10^t, \quad 0.1 \leq |x'| < 1$$

の形で表わす。この502用の F_t プログラムでは指数部 t は

$$-127 \leq t \leq 127$$

の整数、仮数部 x' (小数)の有効けたは10進10けたを保持しているから(注1)かなり広範囲の数値を扱うことができる。1数値語は下図のように構成され倍精度演算の場合に同じく2単一語を要している。



プログラムは入出力、四則演算を基に平方根、特殊関数などの演算プログラムをも含み、通常しばしば当面する計算のほとんどは容易に処理することができる。

プログラムの使用法は簡単に述べるとその計算に必要な情報(注2)を特定のアドレスに格納し、ジャンプテーブル内の一定の番地に飛ばせば所定の計算を終えて主プログラムに戻るようになっている(各演算と飛び先番地などは第3表(その2)に示した)。また計算途上における数値のけたあふれ(overflow)やけた落ち(underflow)、あるいは負の数値について平方根や対数を求めるなど計算処理の仕方に誤りがあると計算機が判断した場合には適宜その旨を外部表示するよう配慮されている。そのほか浮動小数

- (注1) x' は $0.1000000000 \leq x' \leq 0.9999999999$ なる数になる。
 (注2) たとえば加算 $x+y$ を行なうとして x, y などを指す。

第3表(その2) 浮動小数点演算の種類とジャンプ・テーブル

飛び先番地	演算内容	データ格納番地			演算時間
		700	702	704	
2000	$x+y \rightarrow \text{Acc}$	x	y		0.6s
2001	$x-y \rightarrow \text{Acc}$	x	y		0.7s
2002	$y \times z \rightarrow \text{Acc}$		y	z	0.5s
2003	$x+(y \times z) \rightarrow \text{Acc}$	x	y	z	1.1s
2004	$x-(y \times z) \rightarrow \text{Acc}$	x	y	z	1.3s
2005	$y \div z \rightarrow \text{Acc}$		y	z	0.5s
2006	$x+(y \div z) \rightarrow \text{Acc}$	x	y	z	1.2s
2007	$x-(y \div z) \rightarrow \text{Acc}$	x	y	z	1.4s
2008	$+ y \rightarrow \text{Acc}$		y		0.2s
2009	$- y \rightarrow \text{Acc}$		y		0.2s
2010	$\sqrt{y} \rightarrow \text{Acc}$		y		1.2s
2011	ノーマライズ-1				
2012	ノーマライズ-2				
2013	整数, 小数分離				
2014	$-y \rightarrow \text{Acc}$		Acc		
2015	$10^y \rightarrow \text{Acc}$		y		
2016	$e^y \rightarrow \text{Acc}$		y		4.9s
2017	$e^{-y} \rightarrow \text{Acc}$		y		5.2s
2018	$\cosh y \rightarrow \text{Acc}$		y		
2019	$\sinh y \rightarrow \text{Acc}$		y		
2020	$\log_{10} y \rightarrow \text{Acc}$		y		3.4s
2021	$\log_e y \rightarrow \text{Acc}$		y		
2022	$\sin y (\text{rad}) \rightarrow \text{Acc}$		y		4.5s
2023	$\cos y (\text{rad}) \rightarrow \text{Acc}$		y		4.6s
2024	$\sin y (\text{deg}) \rightarrow \text{Acc}$		y		4.5s
2025	$\cos y (\text{deg}) \rightarrow \text{Acc}$		y		4.6s
2026	$\sin \pi/2 y \rightarrow \text{Acc}$		y		4.5s
2027	$\cos \pi/2 y \rightarrow \text{Acc}$		y		4.6s
2028	$\sin \pi/2 y -1 < y < 1 \rightarrow \text{Acc}$		y		4.0s
2030	$\tan^{-1} y \rightarrow \text{Acc}$		y		3.8s
2044	数値読込み	テ	-	ブ	
2047	数値タイプ	x			

点演算に必要な各種定数などをまとめて一定の番地に内蔵するなどプログラム作成の際の便宜を計っているいろいろな処置がとられている。

5. 結 言

制御用計算機 HITAC 501, 502 の装置、プログラムについて述べた。プログラムとしては、ここでは実際の制御用のプログラムより、むしろ使用する場合に基本となるプログラムを、501についてはデバッグ・プログラム、502についてはより高度の演算・制御を行なうという面から開発した倍精度演算プログラム浮動小数点演算プログラムについて詳しく述べた。ロガーおよび経済負荷配分などの実際のプログラムについては別に報告されているゆえ参照せられたい。

HITAC 501 は主としてロガー、または簡単な計算を伴う制御用として適しており、現在すでに各所で実働しているが、最近では無故障時間が2,000時間を越える成績を示している。HITAC 502 はたとえば高効率運転などのように、より高度の演算処理が要求される場合に用いられることになる。また計算機をオンラインで使用することは今後ますます多くなるものと考えられる。

終わりに本計算機は開発途上から、日立製作所戸塚工場と中央研究所が協同行なったものである。また通産省補助金を受けるなど社外、社内の多くの方々のご援助、ご協力をいただいている。

参 考 文 献

- (1) 須藤他: 日立評論 別冊 47, 51 (昭37-5)
 (2) 谷他: 日立評論 別冊 47, 42 (昭37-5)