

ディジタル計算機“HIPAC Mk-1”の論理設計について

Logical Design of Digital Computer “HIPAC Mk-1”

萱 島 興 三* 高 田 昇 平** 島 田 正 三**
Kozo Kayashima Shohei Takada Shozo Shimada

内 容 梗 概

1956 年秋から設計製作の開始されたディジタル計算機 HIPAC Mk-1 は科学技術計画に適する本格的な自動計算機であって、1 年半を経た今日、製作、調整の段階を終り、稼動状態に入っている。演算部にパラメトロン、記憶装置に磁気ドラムを用いているこの計算機は、パラメトロンの変調周波数が 10 kc 程度でかなり低いことと、磁気ドラムの回転数が 5,600 回転/min でかなり低いことのために、演算速度の上からは不満な点が多いが、研究所としてもはじめての試みであり、1,000 語という大容量の記憶装置をもっているために、実用的な計算機として今後の活躍が期待される。大容量の内部記憶装置をもち、ストアード・プログラムの計算を行うこの種の電子計算機は、わが国においてはまだ試作の域を出ない現状である。欧米においてすでに数千台ものディジタル形電子計算機が製作、販売され、動いている今日、これらの計算機の国産化への第一歩として、HIPAC Mk-1 の完成は意義が深いものといえよう。

第 1 節では、HIPAC Mk-1 の紹介を、第 2 節では、演算回路、制御装置について、ブロック・ダイアグラムをもとにして述べる。第 3 節は、この計算機を使う立場よりの設計を、演算指令を中心に述べ、簡単なプログラムの例を示す。

1 箇年という短時日の間に製作が行われたものであるだけに、設計の面からも、実装上の設計の面からも、中途において多くの変更が加えられ、今日の最終的な形になったものであり、幾多の不満足な点が残っている。論理設計上から今後に残された最も重要な問題は、誤動作検出のための回路であって、計算機の信頼度を増すためには、これをさらにつけ加える必要のあることが痛感される。

1. 緒 言

一昨年秋、日立電線株式会社の依頼により設計、製作が開始されたディジタル形電子計算機 HIPAC Mk-1 は、科学技術計算に適する本格的な自動計算機であって、1 年半を経た今日、製作、調整の段階を終り、稼動状態に入っている。設計に際しては、計算機全体の構成、演算指令方式について、イギリス、ケンブリッジ大学の電子計算機 EDSAC を参考にしたが、製作の途中で大変更を加え、アメリカ、イリノイ大学の ILLIAC の Code を一部取り入れ、独自の考案による記憶装置の番地変更方法を付加してできるかぎり使いやすしいものとした。磁気ドラム記憶装置の制御回路は、パラメトロンと組合せて動作させる点においてまったく新しい HIPAC 固有の設計となっている。パラメトロンの演算回路については、演算速度よりも確実に動作することに重点をおいたが、少々パラメトロンの個数がふえても速度の速い並列の制御方式を採用した。演算速度は、パラメトロンの演算回路で実際に費やされる計算の時間よりも、むしろ演算命令乃至はデータを磁気ドラムから読出したり、データを磁気ドラムに書込んだりするのに要する時間によって制約されている。パラメトロン自身の演算時間は、大略下記のごとくである。

加 減 算	40 τ	4 msec
乗 算	80 τ	8 msec

除 算 1,600 τ 160 msec

判断操作 3~10 τ 0.3~1 msec

加減算、乗除算に要する時間の大部分をしめるのは、加算乃至は減算に際し“桁上り”(Carry または borrow) が下の桁から上の桁に順に上ってゆき、落着くまでに要する時間であり、これを最大時間で与えたのが上記の値である。もし、“桁上り”を検出する回路を別に設け、桁上りが落着けば、すぐ次の操作に移れるようにすれば、演算速度は、次のように短縮できる。

加 減 算 平均 1 msec

乗 算 平均 5 msec

除 算 平均 40 msec

つぎに、磁気ドラムから読出したり、磁気ドラムに書き込んだりするのに要する時間——待時間——であるが、HIPAC Mk-1 の磁気ドラムは 5,600 回転/min であって、一回転に約 11 msec を要する。したがって相続く命令をドラムの相隣る位置に記憶させてあれば、加、減、乗算は 1 回転すなわち、11 msec で処理することができる。ただし、被演算数(operand)の書込んである位置が適当でなければ、これらの演算が二回転にまたがり、22 msec かかる場合もあり得ることになる。HIPAC Mk-1 では原則として普通の演算(除算を除く)は、ドラムの一回転の間に行うことになるが、ドラムの回転位置にたいして番地を適当にわりふれば 1 回転のうちに数回の演算を行うことが可能となり、演算速度を上げることができる。

* 日立製作所中央研究所

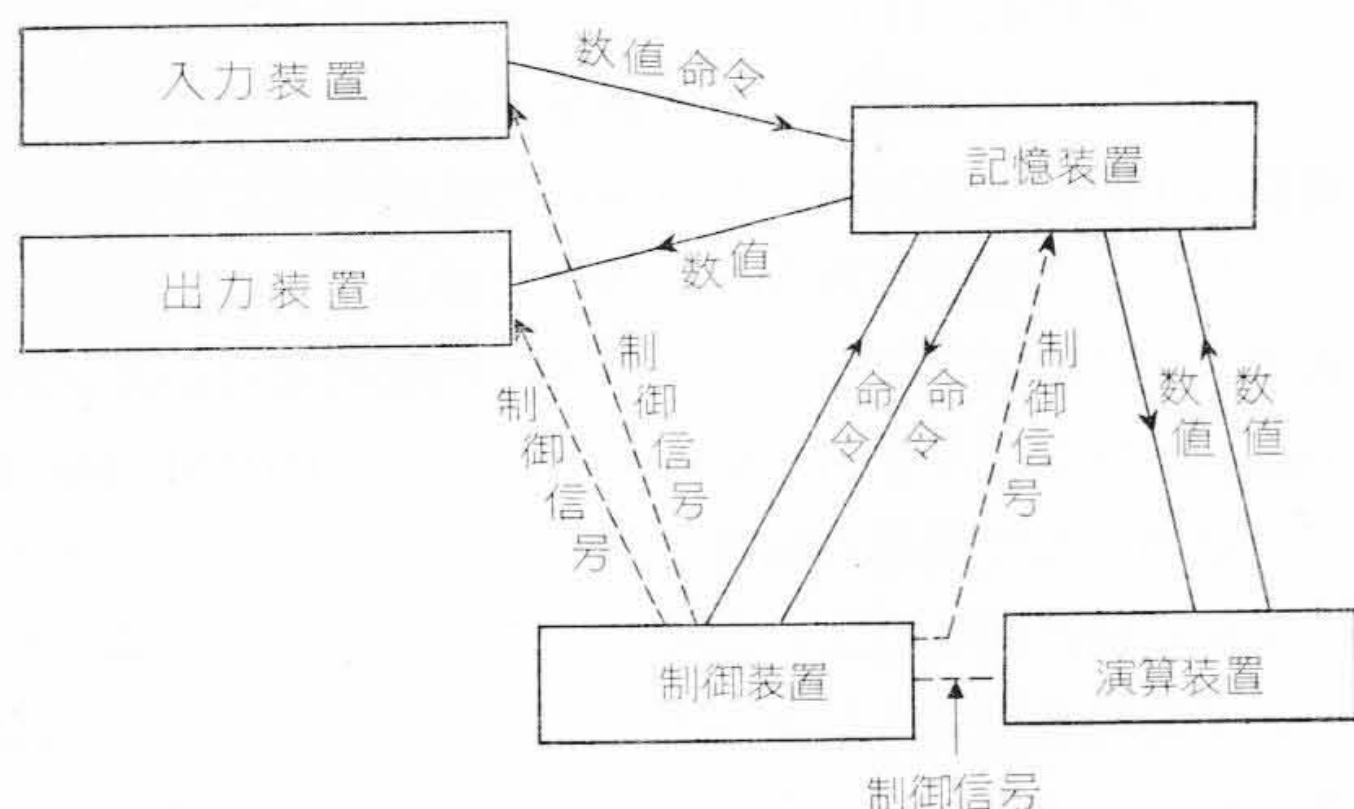
** 日立製作所中央研究所 工博

内部の計算はすべて2進法で行っているので、入出力装置との間でデータを出し入れする際には2進数と10進数との間の変換を行わねばならないが、そのかわり、内部の四則演算は非常に簡単になっている。数値は $[-1, 1]$ の範囲で取り扱うことにし、38桁の2進数字の最上位の桁をサイン・ディジットにあて、このあとに小数点がある固定小数点方式を採用した。したがって計算に際しては、数値の **normalization** を考慮しなければならないが、適当なプログラムをつくることによって自動的に小数点の位置を変える浮動小数点の演算を行うこともできる。

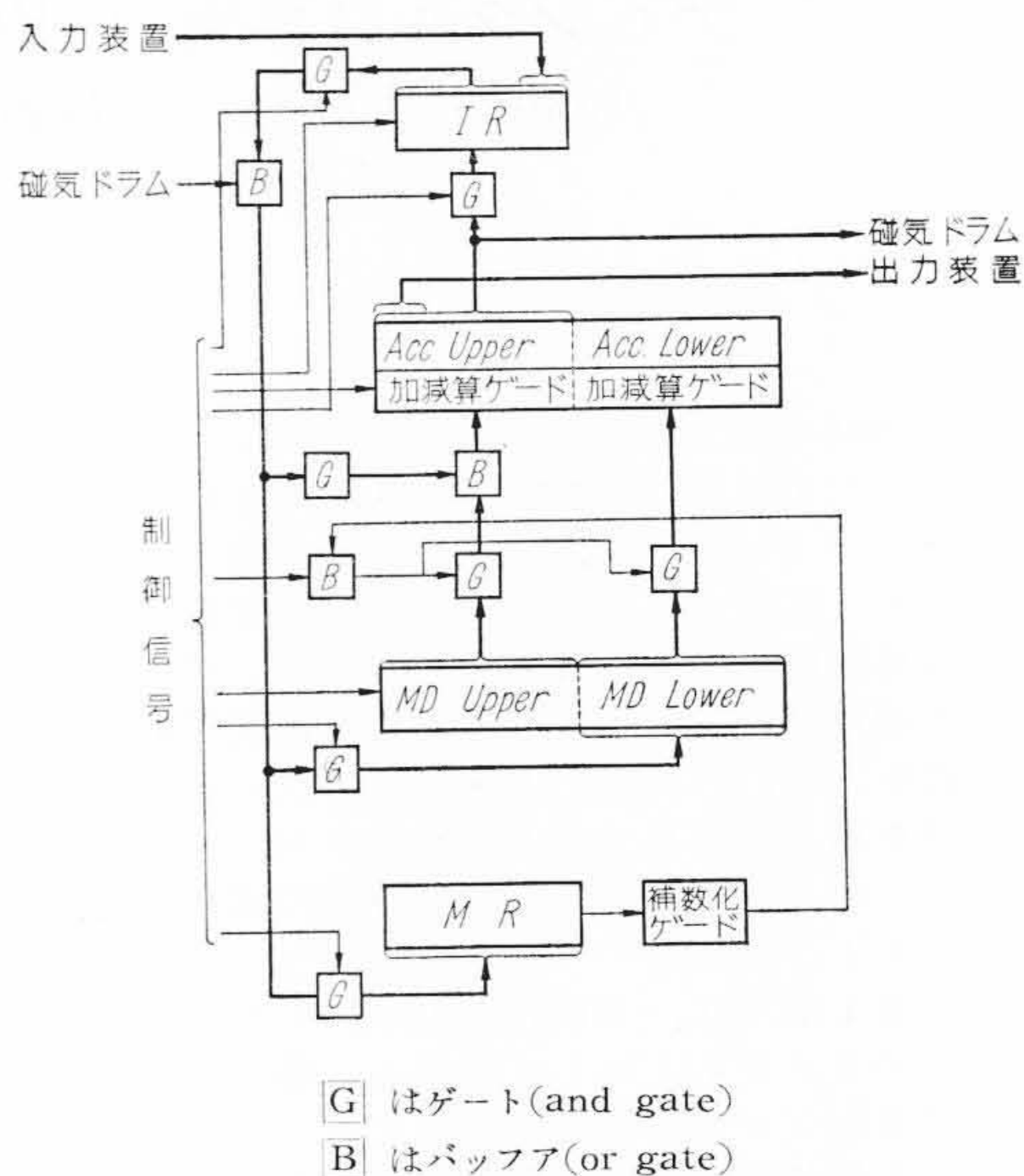
演算指令 (Instruction) としては、最も普通に行われている単一アドレス指令 (One-address Code) 方式を採用した。これは、一つの演算を行うに際してその関係するデータとして記憶装置内の一つのアドレスのみを指定するものであって、演算の順序は記憶装置に入っている演算指令のアドレスの順に行われる。この方式によればやはり制御回路が最も簡単になる。一つの演算指令として、38桁は不要なので、EDSAC の方式に従い、これを二つに分けて 19 桁の短単語とし、短単語ごとに一つの演算指令をつくった。

2. HIPAC Mk-1 の構成

HIPAC Mk-1 は、ほかの自動計算機と同じように演算制御装置、記憶装置、入出力装置の三つの部分よりなっている。この関係を第1図に示す。数値も命令も入力装置から一度記憶装置に書込まれる。記憶装置に入れた命令は、制御装置に読出されて解読され、制御信号が、入出力装置、記憶装置乃至は演算装置に送られて所定の演算を行う。これが終わると、次の命令が記憶装置からふたたび読出され、逐次このような操作をくり返していく。記憶装置の中では数値も命令も全く同等に取り扱われるが、これが命令と解されるのは、制御装置に送られてきた場合だけである。したがって逆に、命令として記憶装置に書込んだデータを演算装置に送り、数値とし



第1図 自動計算機の構成



第2図 演算レジスタ構成ダイアグラム

て取り扱わせて、命令自体を書き換えることができる。また命令が Electronic に記憶され、読出されるために、行う命令の選択が迅速かつ容易であり、何度も同じ計算を繰返し行うような場合には、わずかの命令を記憶させるのみで非常に長い一連の演算操作をやったのけることが容易にできる。

HIPAC Mk-1 においては、入力装置は電信用テープリード、出力装置は同じく電信用タイプライタ、記憶装置は磁気ドラム、演算装置にはパラメトロンを使用している。中央の制御装置はパラメトロンを使用しているが、演算装置以外の各ユニットには、これを制御する真空管の回路が付属しており、パラメトロンの制御信号が直流パルスに変換されて、各ユニットに伝えられる。

演算はパラメトロン回路で行われる。これに用いられるレジスタは、累算器 (Accumulator, Acc), 被乗数・除数レジスタ (Multiplicand-Divisor Register, MD), 乗数レジスタ (Multiplier Register, MR), 入力レジスタ (Input Register, IR) の四つであつて、これらの間の数値の移動の関係を、第2図に示す。数値は2進法38桁よりなるが、Accumulator および Multiplicand-Divisor Register は、それぞれ倍の長さ75桁であつて38桁の Upper と、37桁の Lower とに分かれている。IR レジスタは、入出力に際し2進-10進の変換を行う便宜上設けたものであり、必ずしも必要でないが、磁気ドラム記憶装置の、一つの long address とほぼ同等の機能をもたせているため、保守の上から便利である。図の太線が数値の移動の経路であるが、すべて38桁一度に移動する並列制御方式を採用している。Accumulator

このように、演算の結果はすべて、Acc. Upper および Lower に得られ、入出力装置への移動以外はすべて 38 桁並列に数値が移動するようになっている。

第 3 図は、HIPAC Mk-1 の中央制御回路の構成ダイヤグラムである。パラメトロンゲートと保持回路との組合せであって、図の太線は並列に信号を送る場合の束線を表わしている。フリップ・フロップは、左側から矢印の入力でセットを表わし、右側の丸印の入力でリセットを表わしている。丸印は、今後すべて禁止あるいは否定の入力と約束する。

図において、上半部は、演算の順序およびタイミングを制御する部分、左下は、読出してきた信号を演算指令と数値とに分離するゲート回路、右下は、磁気ドラムのアドレスを制御する部分である。自動演算は、テープ、リーダーから送られてくる左上の演算開始のパルスによって始まる。このパルスは、自動演算のフリップ・フロップを+に倒し、ゲートを通り抜けてアドレス・カウンタの内容をアドレス・レジスタにセットする。これが、最初に行うべき演算指令のアドレスになるのである。セットした後で、アドレス・カウンタの内容を“1”だけ増し、次の演算指令の読出しにそなえる。ついで“命令読出し F.F”を+に倒しておいてから、読出し F.F を+に倒す。これは直流信号に変換され、磁気ドラムへ送られる。磁気ドラムが、アドレス・レジスタの示す所定の位置へ回転してくると、読出しが行われ、読出し P 回路が働いて信号を伝えるとともに、読出し終了の指令を送る。これが“読出し F.F”が+に倒れているため、読出しゲート回路の命令を読出すゲートの方を開いて、演算指令のアドレス部をアドレス・レジスタに、命令を表わす部分を命令読出ゲートに送る。同時に、命令の読出しが完了したのであるから、“読出し F.F”と“命令読出し F.F”とをリセットする。読出し回路のゲートは、演算指令が、数値の半分よりなるため 38 桁の上半か、下半かのいずれかを選び出す機能をもっており、この指定はアドレス・レジスタが行う。命令読出ゲートは、コード化された命令を解釈し、演算の種類を示すパルスを演算制御パルス発生回路に送る。演算制御パルス発生回路は、保持回路、ゲート、カウンタなどの組合せであって、演算が数値の読出しを必要とする場合にはふたたび読出し F.F を+に倒し、ドラムから数値が読出されるのを待って、演算レジスタに必要なパルスを送る。書込みの命令であれば、書込みの F.F を+に倒す。これらの場合のアドレスは、読出された演算指令のアドレス部がアドレス・レジスタにセットされており、これで制御される。読出しも書込みも必要としないときには、ただちに演算レジスタあるいはアドレス指定のカウンタ、レジスタに制御パルスを送る。パラメトロンは、1 個ごとに $\frac{1}{3}\pi$

(π は変調の周期) のおくれを生ずるから、このタイミングを考慮しなければならないのはいうまでもない。かくして命令のパルスが演算制御パルス発生回路を通り抜けて所定の操作が終ると、左上の“演算終了”のバッファへ送られる。このパルスは、自動演算 F.F が+に倒れておれば、ゲートを通り抜けてまたアドレス・レジスタをセットし、命令読出し F.F、読出し F.F をセットして次の演算指令を読出す操作にうつる。かくして一つの演算が終れば、必ず演算終了のパルスが出、演算停止の命令によって自動演算 F.F がリセットされるまでは、ゲートを通り抜けつぎつぎと演算指令を読出して演算を行っていく。制御パルス発生回路は、判断の操作を行うために、前に行った演算の結果としてできた Acc. レジスタ、MD レジスタ、MR レジスタなどの符号の信号を受取っている。

実際の回路は、保守、監視の目的で、種々の補助回路をつけている。便宜上、大部分省略したが、その一つとして、操作卓に Manual Operation のスイッチを設け手動で 1 ステップの演算を行わせてみて回路を点検している。これがバッファを通して命令読出ゲートへいっている束線である。

3. 演 算 指 令

3.1 読込み指令 (control codes)

前に述べたように、テープ読込みはこれを制御するパラメトロン回路を別に設けて、中央制御回路を働かせずに独立に読込みの操作を制御できるようにしている。この制御の信号はテープ自身から読み取られるものであり、control code と名付けている。これによって記憶装置を用いずに、テープ制御回路のチェックに役だつ。

読込み用のテープは、標準 6 単位電信用のものであるが、せん孔機を 2 進のコードで表わし、キーの数をへらすために主として制御の目的に使われる 2 単位は独立に穴をあけ、この時にはテープを送らないようになっている。第 1 表に、せん孔機のコードを示す。

X, Y, Z のキーではテープを送らないようになつていたので、2 進符号で 16 から 63 までの符号は X, Y, Z とほかの 16 のキーとの組合せであけることができる。10 進 1 桁の数を表わすには、A から I までと、スペースのキーで足り、それ以外のキーは命令のコードあるいはコントロール・コードとして使用する。コントロール・コードは、Z および Y せん孔をもつコードで、テープから読取られると、つぎのような制御を行う。

(1) ZA, ZB, ZC

これは、アドレスの指定ないしは修正を行う三つのカウンタ、サイクル・カウンタ #1, サイクル・カウ

第 1 表 セン孔機のコード

第 1 表 セン孔コード																				
キ		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	X	Y	Z	スペース S
—		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O				
セン 孔 コ ー ド	1	○		○		○		○		○		○		○		○				
	2		○	○			○	○			○	○			○	○				
	3				○	○	○	○					○	○	○	○				
	4								○	○	○	○	○	○	○	○				
	5																○		○	
	6																	○	○	

ンタ #2, アドレス・カウンタに, 数をセットする指令で, IR レジスタの最下位 11 桁にある数がそれぞれ上のカウンタにセットされる (第 3 図参照)。回路は IR レジスタからアドレス・レジスタを通してそれぞれのカウンタへいく。

(2) ZI, ZJ, ZK

これは(1)と逆の操作で, 三つのカウンタに入っている数を, アドレス・レジスタにセットする指令である。(1)と(2)とを組合せれば, アドレスをカウンタに一時記憶しておき, 必要に応じて読出してきて使用することができる。

(3) ZD, ZE

これはドラムへ数値を書込む指令であって, このコードが読まれたときアドレス・カウンタの示していたアドレスに, Acc. レジスタにある数が書込まれる。ZE の場合は Acc. の数そのままであるが, ZD の場合は, $C(\text{Acc.}) - 1$ すなわち Acc. の数から 1 を引いたものが書込まれることになる。書込みが終れば, Acc. IR レジスタをリセットするとともに, アドレス・カウンタを進めて次の書込みにそなえる。

(4) ZF

Acc. および IR レジスタ clear の指令である。数値, 命令を書込む前に, 必ず ZF を入れて Acc. および IR を clear しておく。

(5) ZG

テープの読込みをやめて, 自動演算を開始せよ, という指令である。これは第 3 図の, 左上の演算開始を与える信号となる。

(6) ZO

6 単位オール・マークの信号で, ミス・パンチの消去に都合がよいから, この信号はなんにもせず, 無視するようにする。

(7) ZH, ZL, ZM, ZS, ZN

これらのコードは, テープ制御で演算を行わせる便

宜上設けたもので演算指令によって同じようなことを行わせることができるから, 実用上は, たいして重要でない。

ZH は, 後述の “H” に相当するもので, IR レジスタまたはドラムの内容を MD レジスタにセットする。ZL は, “F” に相当するもので, Acc. の内容を IR レジスタまたはドラムに書込む。書込んだあとで Acc. をリセットする。

ZM は, “M” に相当するもので, IR レジスタまたはドラムの内容と, MD レジスタの内容とをかけて, Acc. レジスタに加える。

ZS は “XH” に相当するもので, MD レジスタの内容を Acc. に加える。

ZN は “XI” に相当するもので, アドレス・レジスタの指定するコードと桁数をタイプする。アドレス・レジスタの指定するコードと桁数をタイプするアドレス・レジスタの内容と, タイプの様式との関係は第 2 表(A)(B)のとおりである。

(8) Yx (x=1, 2, ……15, S)

Y パンチは, 演算指令の読込みをコントロールするもので, これが読まれたときの Acc. の内容と, x のうちの 3 ビットと, この次のコードとで命令が構成され, ドラムに書込まれる。このことについては, 次節の演算指令のところで述べる。

以上のコントロールコードを第 2 表にまとめて示す。簡単のため, 次の略号を用いている。

Acc.: Acc. レジスタ, 累算器

MD: MD レジスタ, 被乗数-除数レジスタ

MR: MR レジスタ, 乗数レジスタ

IR: インプット・レジスタ, 入力レジスタ

AC: アドレス・カウンタ

cc #1: サイクル・カウンタ No. 1

cc #2: サイクル・カウンタ No. 2

AR: アドレス・レジスタ

第 2 表(A) タイプの桁数と m との関係

タイプの桁数		1	2	3	4	5	6	7	8	9	10	11	12	13	14
m	Acc.の内容	1024	992	960	928	896	864	832	800	768	736	704	672	640	608
	スペース	1023	996	964	932	900	868	836	804	772	740	708	678	644	612

C(Acc.), C(MD)etc: Acc. の内容, MDの内容など。

Zパンチ, Yパンチのない場合は, 5桁の2進数として取り扱われ, 1字読取られるごとに2進数から10進数へ

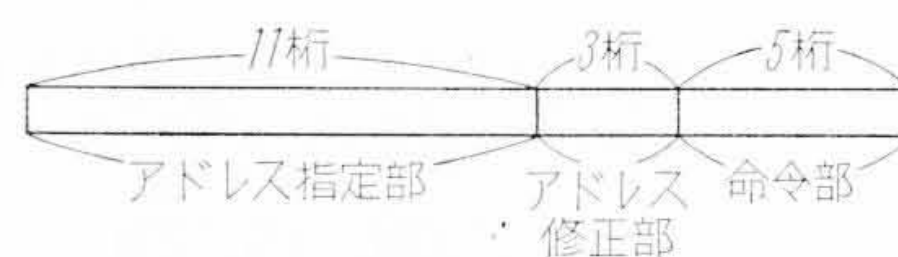
の変換が施される (Yx のつぎにきたときだけ例外で, この場合は演算指令を構成する命令として取り扱われる)。

この変換は, Acc. と IR とを用い, 加算とシフトのくり返しによって行う。すなわち, 1字読取られると, Acc. の内容が10倍され, これにいま読取った数が加えられて, ふたたび Acc. に保存される。したがって, 上の桁から順にテープにパンチすれば, これを整数として2進数に直して Acc. に入れることになる。小数として扱うには, これに適当な常数をかければよい。1で述べたように, HIPAC Mk-1 は, 単一番地指令方式を採用している。1長単語は38桁(2進)であるが, これを19桁の2短単語に分け, これを, 第4図に示すように, アドレス指定部, アドレス修正部, 命令部に分けている。

磁気ドラムは, 19桁の短単語で2,048語の記憶容量があり, この一つ一つにアドレスを付けているので, アドレスの指定に2進11桁を必要とする。これが, 数値として扱ったときの最上位の桁となり, つづく3桁を, この11桁のアドレスを修正する目的に使っている。これは, 演算指令を書き換えることなしにいくつもの目的に使えるように工夫したもので, この3桁の指定によって, AC, cc #1, cc #2 などの内容によって, 実際に演算が行われるときに上の11桁のアドレスに修正を加える。最後の5桁が行う演算の種類, すなわち命令を規定する部分であって, 32種類の命令(実際には38種類ある)を行わせることが可能である。この演

第 2 表(B) タイプライタコードと p, n との関係

セレクトマグネット					コード		p	1桁タイプのときの $m = p + 2^5 \times 32$
1	2	3	4	5	“上”の場合	“下”の場合		
○	○				—	A	24	1048
○			○	○	?	B	19	1043
	○	○	○		:	C	14	1038
○			○		\$	D	18	1042
○					3	E	16	1040
○		○	○		!	F	22	1046
	○		○	○	&	G	11	1035
		○		○	£	H	5	1029
	○	○			8	I	12	1036
○	○		○		,	J	26	1050
○	○	○	○		(	K	30	1054
	○			○	)	L	9	1033
		○	○	○	.	M	7	1031
		○	○		,	N	6	1030
			○	○	9	O	3	1027
	○	○		○	0	P	13	1037
○	○	○		○	1	Q	29	1053
	○		○		4	R	10	1034
○		○			BELL	S	20	1044
				○	5	T	1	1025
○	○	○			7	U	28	1052
	○	○	○	○	;	V	15	1039
○	○			○	2	W	25	1049
○		○	○	○	/	X	23	1047
○		○		○	6	Y	21	1045
○				○	+	Z	17	1041
					(Acc. の内容)		0	1024
			○		Carriage Return		2	1026
	○				Line Feed		8	1032
		○			Space		4	1028
○	○	○	○	○	Letter Shift		31	1055
○	○		○	○	Figure Shift		27	1051



第 4 図 演算指令の構成

第3表 コントロール・コード

ZA	C(IR)→C(cc#1)
ZB	C(IR)→C(cc#2)
ZC	C(IR) C(AC)
ZD	C(Acc)→Memory へ書込み
ZE	C(Acc)→Memory へ書込み
ZF	Acc, IR clear
ZI	C(cc#1)→C(AR)
ZJ	C(cc#2)→C(AR)
ZK	C(AC)→C(AR)
ZL	C(Acc)→C(IR)
ZH	C(IR)→C(MD)
ZM	C(Acc)+C(MD)×C(IR)→C(Acc)
ZS	C(Acc)+C(MD)→C(Acc)
ZN	C(AR) の指定に従つて, タイプ
ZG	プログラムスタート, テープ読取り中止
ZO	無 視
YX	演算指令構成

算指令を $n Yx f$ と書いて表わす。 n, x はそれぞれ, $0 \leq n \leq 2047, 0 \leq x \leq 7$ のような整数を, f は, A, B, C, ..., 0, XA, XB, ..., XO のような Function letter を代表している。これは一般に n, x できる記憶装置のアドレスの数値に f なる演算操作を施す指令と解される。しかし, 38 種類の命令の中には, 指定したアドレスが演算を施すべき数値(Operand)のアドレスではなく, つぎに行うべき演算指令のアドレスである場合や, 全然アドレスとしての意味をもたない場合などがある。したがって, Yx の修正も, アドレスの修正とは異なったパラメータの修正である場合, 修正しても意味をもたない場合が生じてくる。いずれの場合にもこの修正を加えられたアドレスを (n, x) で表わすことに約束する。これらの演算指令を, このような意味で分類して説明すれば, 次のようになる。

(1) (n, x) が演算を施す数値のアドレスを示すもの

- $n Yx A$ $C(n, x)$ を Acc. に加える。
- $n Yx XA$ Acc. をリセットし, $C(n, x)$ を Acc. に加える。
- $n Yx B$ $C(n, x)$ を Acc. から引く。
- $n Yx XB$ Acc. をリセットし, $C(n, x)$ を Acc. から引く。
- $n Yx C$ $C(n, x)$ を Acc. から引き, 結果を $C(MD)$ で割る。商は Acc. にできる。剰余は捨てられる。
- $n Yx XC$ Acc. をリセットし, $C(n, x)$ を Acc. から引き, 結果を $C(MD)$ で割る。商は Acc. 剰余は捨てられる。
- $n Yx D$ $C(n, x)$ を Acc. に加えるほかは, $n Yx C$ に同じ。
- $n Yx XD$ $C(n, x)$ を Acc. に加えるほかは, $n Yx XC$ に同じ。
- $n Yx F$ $C(n, x)$ を (n, x) に書込む。

- $n Yx XF$ Acc. をリセットし, 0 を (n, x) に書込む。
 - $n Yx H$ $C(n, x)$ を MD にセットする。
 - $n Yx M$ $C(n, x)$ と $C(MD)$ とをかけて, 積を Acc. に加える。
 - $n Yx XM$ Acc. をリセットしてから “M” の操作を行う。
 - $n Yx N$ $C(n, x)$ と $C(MD)$ とをかけて, 積を Acc. から引く。
 - $n Yx XN$ Acc. をリセットしてから “N” の操作を行う。
 - $n Yx O$ 絶対値の演算で, x が偶数ならば, $|C(n, x)|$ を Acc. に加える。 x が, 奇数ならば, $|C(n, x)|$ を Acc. から引く。数値はすべて長単語として扱う。
 - $n Yx XO$ Acc. をリセットした後に “O” の演算を行う。やはり, 数値は長単語として扱う。
- (2) (n, x) が次に行う演算指令のアドレスを示すもの。
- $n Yx E$ 条件跳越の指令で, x が偶数のときは, $C(Acc.)$ の符号をしらべ, 正または 0 ならば, つぎに行う演算指令を (n, x) からとる。 x が奇数のときは, $C(MD)$ の符号をしらべて, 正または, 0 ならば, つぎに行う演算指令を (n, x) からとる。
 - $n Yx G$ $C(Acc.)$ または $C(MD)$ の符号をしらべ負ならば “E” と同じように跳越しを行う。
 - $n Yx XE$ 無条件跳越しで, 次に行う演算指令を (n, x) からとる。
 - $n Yx J$ サイクル・カウンタが 0 でなければ, 次に行う演算指令を (n, x) からとる。サイクル・カウンタの内容を, 1 だけ減じておく。 x には No.1 と No.2 のサイクル・カウンタの指定が含まれている。
 - $n Yx XJ$ サイクル・カウンタが 0 ならば, 跳越しを行う。ほかは命令 J の場合と同じ。
- (3) (n, x) が演算のパラメータとなるもの。
- $n Yx I$ 数値 (n, x) をサイクル・カウンタにセットする。 x は, No.1 と No.2 のサイクル・カウンタの指定を含む。
 - $n Yx XI$ 数値 $m=(n, x)$ の指定にしたがってタイプする。タイプの様式と m との関係は第2表, (a), (b) に与えられている。
 - $n Yx K$ Acc. を $m=(n, x)$ 桁左へシフトする。すなわち $C(Acc.)$ に 2^{-m} をかける。
 - $n Yx XK$ “K” の操作を Acc. の代りに MD につ

いて行う。

- $n Yx L$ Acc. を $m=(n, x)$ 桁右へシフトする。
すなわち $C(\text{Acc.})$ に 2^m をかける。
- $n Yx XL$ “L” の操作を Acc. のかわりに MD について行う。
- $n Yx XS$ 演算停止の指令で, Ac に (n, x) をセットした後, x が偶数ならば, テープリードから数値をよむ。 x が奇数ならば, 停止する。

(4) (n, x) が全然意味をもたないもの。

- $n Yx XH$ x が偶数ならば, $C(\text{Acc.})$ に $C(\text{MD})$ を加える。 x が奇数ならば, $C(\text{Acc.})$ から $C(\text{MD})$ を引く。
- $n Yx S$ 四捨五入の指令である。Acc. に, Z^{-38} を加える。
- $n Yx XG$ 何もせず, 無視して次の演算にうつる。

次に, Yx による n の修正について述べる。AR には加算回路があって, AC, $cc \#1$, $cc \#2$, のどの内容でも加えられるようになっている。それで, 演算指令が, 記憶装置から読み出されてきてそのアドレス部が, AR に読込まれたとき, Yx の 3 ビットが解読され, 必要ならば AR へいくゲートを用いて, そのいずれかの内容を AR に加えてやるようにしている。また, 数値を, 短単語 2 進 19 桁として取り扱うこともできるようにしているが, この短単語か, 長単語かの指定も x で行っている。前に述べたように, 命令によって, x の修正の意味が違ってくるので, 前述の分類に準じて修正の方法を整理すれば, 次のようになる。

(1) (n, x) が, 演算を施す数値のアドレスを示すもの。

A, XA, B, XB, C, XC, D, XD, F, XF,
H, M, XM, N, XN, O, XO (O, XO のときは例外)

$x=s$ 数値は長単語, アドレスは n } 何も加えない。
 $x=1$ 数値は短単語, アドレスは n }

$x=2$ 数値は長単語, アドレスは $n+C(\text{AC})$ }
 $x=3$ 数値は短単語, アドレスは $n+C(\text{AC})$ }
AC の内容を加える。

$x=4$ 数値は長単語, アドレスは $n+C(cc\#1)$ }
 $x=5$ 数値は短単語, アドレスは $n+C(cc\#1)$ }
 $cc \#1$ の内容を加える。

$x=6$ 数値は長単語, アドレスは $n+C(cc\#2)$ }
 $x=7$ 数値は短単語, アドレスは $n+C(cc\#2)$ }
 $cc \#2$ の内容を加える。

(2) サイクル・カウンタの指定を要するもの

J, XJ, I

この場合は, x が偶数の場合は $cc \#1$ を, 奇数の

場合は $cc \#2$ を指定することになっている。 n の修正は, (1) の場合とまったく同じで, $x=0, 1$ のときは何も加えず, 2, 3 のときは AC, 4, 5 のときは $cc \#1$, 6, 7 のときは $cc \#2$ の内容を加える。

(3) (1), (2) 以外の演算指令で, アドレスが意味をもつもの。

E, XE, G, XI, K, XK, L, XL, XS

この場合は, 修正の仕方は前と同じであるが, 長単語, 短単語の区別, あるいは, サイクル・カウンタの指定を必要としないから, E, G をのぞいて, 0 と 1, 2 と 3, 4 と 5, 6 と 7 はそれぞれまったく同等に取り扱われる。E, G のときは, 偶数ならば Acc. の内容で, 奇数ならば MD の内容で跳越しを判定する。

(4) アドレス部が全然意味をもたないもの

XG, XH, S

この場合は, x が何であつても結果はすべて同じである。

$Y2, Y3$ の場合は, アドレス・カウンタの内容が加えられるから, その演算指令の入っているアドレスによって実際に演算を施すアドレスが変ってくる。つまり, その演算指令の入っているアドレスを基準とした相対アドレスが n であると考えられ, サブルーチンを用意する場合に便利である。 $Y4, Y5, Y6, Y7$ は Operand のアドレスを変えていきたい場合に, J, XJ, I の演算指令と組合せて同じ演算指令が読み出されるごとに, サイクル・カウンタの数を数えて目的を達することができる。これは, 多くの計算機で用いられている“アドレス記憶”(Store Address) の命令よりもずっと使いやすいと思われる。また, このような修正ができなければ, 演算指令そのものに演算を施して書き換えていかねばならず, 時間もかかり, きわめて不便である。

なお, 上の修正で長単語として取り扱うときは, アドレスが偶数でなければならない。また, F, XF の命令の場合, 短単語とすれば, 偶数番地には, Acc. の上 19 桁が, 奇数番地とすれば, Acc. の下 19 桁が書込まれるようになっている。絶対値の演算を扱う O, XO の演算指令のときは, 短単語として扱うことをやめ, x が偶数ならば加算, 奇数ならば減算を行うようにした。

演算指令を構成する Yx の x は 3 ビットとして (つまり $0 \leq x \leq 7$ として) 説明したが, x は 4 ビットの値つまり 0 から 15 までの値をとる。 x が 8 から 15 までの値の場合は, 下の 3 ビットだけが命令の構成に使われ, $Y8$ と YS , $Y9$ と $Y1$ は同じように 3 ビットがそれぞれ 0, 1 になるが, 読込みのときにアドレス・レジスタの内容が, そのアドレス部に加えられる。最後に簡単なプログラムの一例として平方根を求めるルーチンを示しておく。

第4表 平方根を求めるルーチン $\sqrt{2C(100)} \rightarrow \text{Acc.}$
(その1)

アドレス	演算指令	
	ZF ZC ZF	以下の命令を0番地から書込む Acc. clear
→0	100 YS XA	$x \rightarrow \text{Acc.}$
1	6 Y1 A	$x + \frac{1}{2} \rightarrow \text{Acc.}$ これを初期値とする
2	16 YS F	
→3	16 YS H	$y_n \rightarrow \text{MD}$
4	16 YS XM	$y_n^2 \rightarrow \text{Acc.}$
5	1 YS K	$\frac{1}{2} y_n^2 \rightarrow \text{Acc.}$
6	1024 YS S	四捨五入 (これは、数値 $\frac{1}{2}$ としても扱われる)
7	100 YS D	$(x + \frac{1}{2} y_n^2) / y_n \rightarrow \text{Acc.}$
8	16 YS F	
9	Y1 XH	$y_{n+1} - y_n \rightarrow \text{Acc.}$
10	14 YS A	$y_{n+1} - y_n + \varepsilon \rightarrow \text{Acc.}$
←11	3 YS G	上式が負ならば、まだ収斂していないものと考え3に戻つてくり返す
12	16 YS XA	正になれば、y を Acc. に入れ
←13	Y4 XE	main routine に戻る
14,15	137 ZE	判定のための ε をいれておく
16,17	ZE	y_n をいれておく

5. 結 言

“ディジタル形電子計算機は常に半年後にでき上る”のジンクスを破って、HIPAC Mk-1 は設計開始後一年余にして動いた。この経験をもとに、より高速な、より信頼度の高い(Fast and reliable)ディジタル計算機を目ざして進まねばならない。このためには、より進んだ演算素子の開発や、部品の改良が必要である反面、論理設計が演算速度を上げうるような、そしてプログラミングに際して少ないプログラムステップで複雑な計算の行えるような、融通性に富んだものであることが要請される。以下論理設計の上で問題になる重要な点を列挙する。

(1) 基 数

10進法を用いるか、2進法を用いるかは、それぞれ利害得失があり、計算機の用途によっても異なる。10進法を採用する場合は、入出力装置とのデーターの出し入れに変換が容易であるが、その反面回路が複雑になることは免れない。ビジネス用の計算機では、大量のデーターを扱うだけに10進法を採用するのが得策であり、多くの計算機は10進法を用いている。科学技術計算用の機械では、2進法を採用したものが多い。

(2) 固定小数点と浮動小数点

浮動小数点の計算機は、リレー式以外は割合少ない。これは、演算速度が多少犠牲になることと、回路構成が複雑になるためであるが、数値の normalization を考えないですむ点からプログラミングが非常に楽になる。固定小数点の計算機でも自動的に normalization

を行うようにプログラムで制御することができるが、浮動小数点の演算回路内蔵の機械に比べて速度がおちることは否めない。今後は、当然浮動小数点に移行するものと思われる。

(3) 直列方式と並列方式

HIPAC Mk-1 は、並列の計算機であり、パラメτροンの低いクロック周波数をもってしても、乗算速度をかなり上げることができた。最近の大形電子計算機の乗算速度は数十マイクロセカンド程度まで早くなっているが、これらはすべて並列の演算によって到達し得た速度である。これによって飛躍的に演算素子の数が増すが、乗算速度が1桁以上違うことを思えば、並列方式をめざすべきであろう。

(4) 単語の長さ

最近計画されている計算機では1語の長さを10進12桁、2進法では40桁程度に大きくえらんだものが多い。科学技術計算では、10桁では多少たりないようである。I. B. M 702のように、単語の長さを任意に変えられるような設計もあるが、あまり有利とは考えられない。もちろんこの場合は直列方式となる。

(5) 単一番地符号と多番地符号

単一番地符号の計算機が圧倒的に多い。多番地符号が有利な点は、数値の1語と同じ長さに命令の1語をとると、桁数がかかなり多いのでいくつかの被演算数の番地を指定することができ、演算速度を上げることができる点であった。これにたいして、単一番地符号を用いるときは1語を二つの単語に分け、このそれぞれを命令とするやり方が行われたが、最近では1語の中に二つの単一番地演算指令を含むいわゆるオーダーペアの方式が案出されている。これは、短単語を用いる方法に比べ、それほど得策でもないと思われる。

(6) チェックの問題

一番簡単なチェック法は Parity チェックで、何ビットかをひとまとめとし、これにチェック・ビットを挿入する。これだけでは、error の検出しかできないが、訂正はできないでも一応目的は達することができる。このほか、演算を2回行つて、合致するものをたしかめる方法や、特殊なチェックを含んだコード(Biquinary code や Two out of five code など)を用いる方法もあるが、あまり一般的でない。磁気テープには、パリティ・チェックを設けるのが普通である。

HIPAC Mk-1 の設計に先きだち、これらの点を十分に検討する余裕がなく、確実に動作することを目標として最も無難な方策をえらんだが、一度方針を決定すると容易に変更できないものだけに慎重に考慮する必要がある。

本稿を終るにあたり、ディジタル計算機研究試作のい

とぐちを作って下さり、終始、御援助を賜わった日立電線株式会社斎藤工場長、久本部長、八田主任に心からお礼申し上げる。なお、試作に際して回路の構成、図面の

検討に、討論と有益な助言をいただいた日立製作所中央研究所、安藤、池上両主任、井立田、大西両研究員、日立電線株式会社岩上、沼尻両氏に厚くお礼申し上げます。

付 表: オ ー ダ コ ー ド 一 覧 表

Instructions	Description of the operation		Notes
n Yx A n Yx XA	Addition Addition	$C(\text{Acc.}) + C(n, x) \rightarrow \text{Acc.}$ $C(n, x) \rightarrow \text{Acc.}$	
n Yx B n Yx XB	Subtraction Subtraction	$C(\text{Acc.}) - C(n, x) \rightarrow \text{Acc.}$ $-C(n, x) \rightarrow \text{Acc.}$	
n Yx D n Yx XD n Yx C n Yx XC	Division Division Division Division	$\{C(\text{Acc.}) + C(n, x)\} / C(\text{MD}) \rightarrow \text{Acc.}$ $C(n, x) / C(\text{MD}) \rightarrow \text{Acc.}$ $\{C(\text{Acc.}) - C(n, x)\} / C(\text{MD}) \rightarrow \text{Acc.}$ $-C(n, x) / C(\text{MD}) \rightarrow \text{Acc.}$	
n Yx M n Yx XM n Yx N n Yx XN	Multiplication Multiplication Multiplication Multiplication	$C(\text{Acc.}) + C(n, x) \times C(\text{MD}) \rightarrow \text{Acc.}$ $C(n, x) \times C(\text{MD}) \rightarrow \text{Acc.}$ $C(\text{Acc.}) - C(n, x) \times C(\text{MD}) \rightarrow \text{Acc.}$ $-C(n, x) \times C(\text{MD}) \rightarrow \text{Acc.}$	
n Yx F n Yx XF	Transfer Transfer	$C(\text{Acc.}) \rightarrow C(n, x)$ $O \rightarrow \text{Acc.}, O \rightarrow C(n, x)$	} See note 4
n Yx H	MD Set	$C(n, x) \rightarrow C(\text{MD})$	
n Yx ⁰ E n Yx ⁰ G n Yx ¹ E n Yx ¹ G n Yx J n Yx XJ n Yx XE	Jump Jump Jump Jump Jump Jump Jump	Jump to loc.(n, x) ; if $C(\text{Acc.}) \geq O$ Jump to loc.(n, x) : if $C(\text{Acc.}) < O$ Jump to loc.(n, x) : if $C(\text{MD}) \geq O$ Jump to loc.(n, x) : if $C(\text{MD}) < O$ Jump to loc.(n, x) : if $C(\text{cc}) \neq O$ Jump to loc.(n, x) : if $C(\text{cc}) = O$ Jump to loc.(n, x) : unconditionally	C(cc) decreases by one
Yx I	CC Set	$(n, x) \rightarrow C(\text{cc})$	
n Yx K n Yx L n Yx XK n Yx XL	Shift Shift Shift Shift	$C(\text{Acc.}): (n, x) \text{ Right Shift}$ $C(\text{Acc.}): (n, x) \text{ Left Shift}$ $C(\text{MD}): (n, x) \text{ Right Shift}$ $C(\text{MD}): (n, x) \text{ Left Shift}$	} (n, x) Mod. 64 Maximum Shift 63 places
n Yx XI	Type	Type the number in Acc. or letter	
n Yx ⁰ XH n Yx ¹ XH	Arithmetic on MD Arithmetic on MD	$C(\text{Acc.}) + C(\text{MD}) \rightarrow \text{Acc.}$ $C(\text{Acc.}) - C(\text{MD}) \rightarrow \text{Acc.}$	Independent of (n, x ⁰) Independent of (n, x ¹)
n Yx ⁰ O n Yx ⁰ XO n Yx ¹ O n Yx ¹ XO	Arith. of Abs. value Arith. of Abs. value Arith. of Abs. value Arith. of Abs. value	$C(\text{Acc.}) + \begin{vmatrix} C(n, x) \\ C(n, x) \end{vmatrix} \rightarrow \text{Acc.}$ $C(\text{Acc.}) - \begin{vmatrix} C(n, x) \\ C(n, x) \end{vmatrix} \rightarrow \text{Acc.}$	
n Yx S	Round off	$C(\text{Acc.}) + 2^{-38} \rightarrow \text{Acc.}$	Independent of (n, x)
n Yx ¹ XS n Yx ⁰ XS	Stop Stop	Stop Stop & Start the tape reader	} (n, x) $\rightarrow C(\text{AC})$
n Yx XG	Neglect	Neglect	
			Independent of (n, x)

Notes

1. (N, X) designates location or number n modified by x : Contents of the register assigned by x may be added to n . x stands for an integer between 0 and 7 (We use letter "S" for 0). The manner of modification is as follows.

x	Register (or counter) which content is added to n
S	—— (Nothing is added)
1	—— (Nothing is added)
2	Address counter
3	Address counter
4	Cycle counter No. 1
5	Cycle counter No. 1
6	Cycle counter No. 2
7	Cycle counter No. 2

2. $C(n, x)$ represents the content of storage location (n, x) .
We can handle words in long (38 binary) number or in short (19 binary) number:
When x is even: long words. (In this case, (n, x) must be even)
 x is odd: short words.
In case of orders refer to cycle counter (such as J, XJ, I),
We can choose either of two cycle counters:
When x is even: cycle counter No. 1.
 x is odd: cycle counter No. 2.
3. X^0 represents even number of x , i.e. 0, 2, 4, 6
 X^1 represents odd number of x , i.e. 1, 3, 5, 7
4. In case of short number transfer order $n Yx^1 F$ or $n Yx^1 XF$, 19 digits of upper or lower half of Accumulator are transferred to storage location (n, x) .
When (n, x) is even: upper half of Accumulator.
 (n, x) is odd: lower half of Accumulator.

(第70頁より続く)

日立製作所社員の通信機器に関する社外既発表論文一覧

(その8)

(5) コンピュータ(その2)

番号	題 目	執 筆 者	掲 載 誌	掲 載 号
8	折 線 近 似 函 数 発 生 器 の 改 良	三 浦 武 雄 沼 倉 俊 郎 雨 宮 洋	電 気 学 会 雑 誌	Vol. 76 No. 818
9	低速度型電子管式アナログ計算機の確度についての 討 論	三 浦 武 雄	電 気 学 会 雑 誌	Vol. 76 No. 819
10	サ ー ボ 掛 算 器 の 解 析	沼 倉 俊 郎 三 浦 武 雄 阿 部 善 右 衛 門 鴨 井 章	電 気 学 会 雑 誌	Vol. 77 No. 827
11	低 速 度 型 ア ナ ロ グ 計 算 機 の 積 分 誤 差	三 浦 武 雄 阿 部 善 右 衛 門 永 田 穰	電 気 学 会 雑 誌	Vol. 77 No. 827
12	サ ー ボ 掛 算 器 の 解 析 に つ い て の 討 論	沼 倉 俊 郎 阿 部 善 右 衛 門	電 気 学 会 雑 誌	Vol. 77 No. 831
13	フ ォ ト フ ォ ー マ の 誤 差 に 関 す る 検 討	三 浦 武 雄 阿 部 善 右 衛 門	電 気 学 会 雑 誌	Vol. 78 No. 833
14	低 速 度 型 ア ナ ロ グ 計 算 機 の 総 合 演 算 誤 差	三 浦 武 雄 阿 部 善 右 衛 門 永 田 穰	電 気 学 会 雑 誌	Vol. 78 No. 838
15	ア ナ ロ グ デ ジ タ ル 変 換 器	青 木 正 直	電 気 学 会 雑 誌	Vol. 76 No. 811
16	パイロット発電機に直列自動界磁巻線を施した場合 の 効 果	三 浦 武 雄 藤 木 勝 美 西 政 雄	電 気 三 学 会 連 合 大 会 資 料	昭 29
17	ア ナ ロ グ 電 気 演 算 器 の 試 作 と 応 用	三 浦 武 雄 沼 倉 俊 郎	電 気 三 学 会 連 合 大 会 資 料	昭 29
18	繰 返 型 ア ナ ロ グ コ ン プ ュ ー タ の 使 用 限 界	三 浦 武 雄 沼 倉 俊 郎	電 気 三 学 会 連 合 大 会 資 料	昭 31
19	ヘリカルポテンショメータの周波数特性とアナコン 誤 差 に お よ ぼ す 影 響	三 浦 武 雄 衣 川 武	電 気 三 学 会 連 合 大 会 資 料	昭 31
20	演 算 増 幅 器 の ト リ フ ト 自 動 補 償 回 路 の 解 析	沼 倉 俊 郎 阿 部 善 右 衛 門 三 浦 武 雄 日 比 正 男 沼 倉 比 正 男 日 比 武 雄 三 浦 武 雄	電 気 三 学 会 東 京 支 部 大 会 資 料	昭 30
21	低 速 度 型 ア ナ ロ グ 計 算 機 の 試 作 実 験	沼 倉 俊 郎 日 比 正 男 三 浦 武 雄	電 気 三 学 会 東 京 支 部 大 会 資 料	昭 30
22	電 子 管 式 ア ナ ロ グ コ ン プ ュ ー タ 解 指 示 装 置	三 浦 武 雄	電 気 三 学 会 東 京 支 部 大 会 資 料	昭 30
23	任 意 函 数 発 生 装 置 の 試 作	沼 倉 俊 郎 三 浦 武 雄	電 気 四 学 会 連 合 大 会 資 料	昭 32-4
24	直 流 再 生 交 流 増 幅 器 の 動 作 解 析	三 浦 武 雄 柏 迫 雄	電 気 四 学 会 連 合 大 会 資 料	昭 32-4
25	日 立 ポ ー タ ブ ル ア ナ ロ グ 計 算 機 に つ い て	三 浦 武 雄 安 藤 文 雄 阿 部 善 右 衛 門 長 谷 川 毅	電 気 四 学 会 連 合 大 会 資 料	昭 32-4
26	指 向 性 計 算 装 置	三 浦 武 雄 沼 倉 俊 郎 小 倉 正 美 衣 川 武 洋	電 気 四 学 会 連 合 大 会 資 料	昭 32-4
27	改 良 型 折 線 近 似 函 数 発 生 器 に 関 す る 実 験 的 検 討	雨 宮 武 雄	電 気 四 学 会 連 合 大 会 資 料	昭 32-4
28	低 速 度 型 ア ナ ロ グ 計 算 機 の 総 合 演 算 誤 差	三 浦 武 雄 永 田 穰 阿 部 善 右 衛 門	電 気 四 学 会 連 合 大 会 資 料	昭 33
29	む だ 時 間 要 素 の 試 作 と 応 用	三 浦 武 雄 高 見 雄	電 気 四 学 会 連 合 大 会 資 料	昭 33

(第90頁へ続く)