

HITAC 3010 の テーブル 演算

Table Translation by the HITAC 3010 Computer

浦 城 恒 雄*
Tsuneo Uraki

内 容 梗 概

HITAC 3010 は字単位のデータ処理に主力をおいた中形の事務用電子計算機である。いくつかの特徴の一つに、金物的にも紙物的にもテーブル演算技術を取りいれていることがあげられる。本文では金物的なものとして、コード変換命令，加減算命令，ラインプリンタ制御方式を，紙物的なものとして，カードコード変換サブルーチン，乗算サブルーチンを取りあげて解説する。

1. 緒 言

HITAC 3010 は RCA 社よりの技術導入によって，電子計算機市場に日立製作所の主力機として登場した中形の事務用計算機である。字単位のデータ処理が容易であること，各種の入出力装置がビルディングブロックで接続可能であることなどを特長とするすぐれた金物的性能をもつとともに，アッセンブリシステムを中心に完備した紙物システムをもっている。

内外に HITAC 3010 と同類の字単位の計算機としては，IBM 1400 シリーズ，NCR 315，B 250 シリーズ，TOSBAC-4200 などあげられる。それらと比較してみると 3010 は比較的素直な方式を採用しているが，いくつかの方式的な特徴をもっている。その中で興味をひく特徴の一つとして金物的にも紙物的にもテーブル演算技術を取りいれていることがあげられる。

以下，金物的なものとして，コード変換命令，加減算命令，ラインプリンタ制御方式，紙物的なものとして，カードコード変換サブルーチン，乗算サブルーチンを取りあげ，どのようにテーブル演算技術を取りいれているかを解説してみよう。

2. コード変換命令

コード変換は通常の計算機ではいくつかの命令を組み合わせて実行される。HITAC 3010 ではこれを 1 命令で実行する。この命令は TRA (TRANSLATE BY TABLE の略) と呼ばれ，1 命令で 45 字以下のフィールドの情報を変換テーブルにしたがって 1 字ずつ変換することができる。命令の形式は下のとおりである。

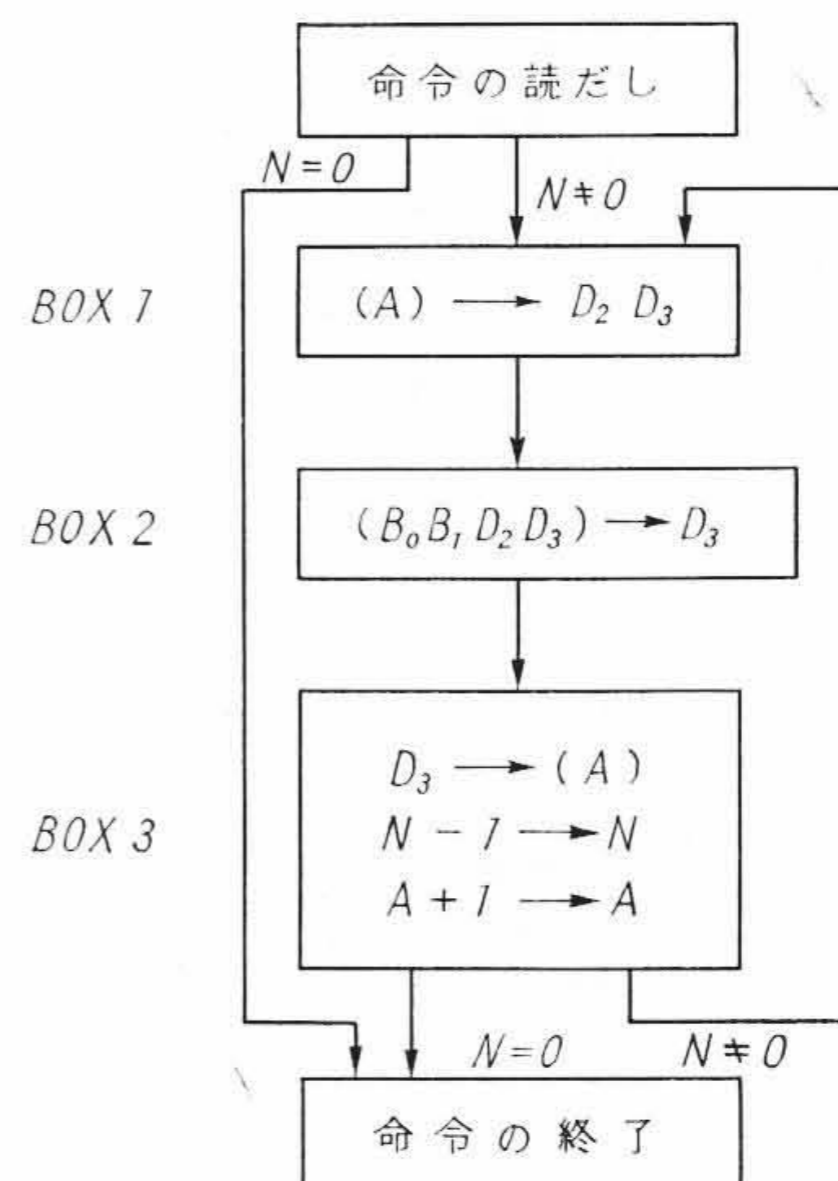
- OP (1 字) A TRA のコード
- N (1 字) 0~45 変換される字数
- A (4 字) $A_0A_1A_2A_3$ 変換されるフィールドの最左端アドレス
- B (4 字) $B_0B_1B_2B_3$ 変換テーブルの先頭アドレス，ただし下 2 けたの B_2, B_3 はともに 0

第 1 図に実行過程の流れ図を示す。A アドレスで示される記憶場所の 1 字を読みだす。それを二進表示で $(a_5 a_4 a_3 a_2 a_1 a_0)$ とすると $D_2 (000 a_5 a_4 a_3)$ ， $D_3 (000 a_2 a_1 a_0)$ で 4 けたのアドレス $B_0 B_1 D_2 D_3$ を作り，そのアドレスで示される記憶場所の 1 字を A アドレスで示す記憶場所へ書きこむ。これが基本的なサイクルである。

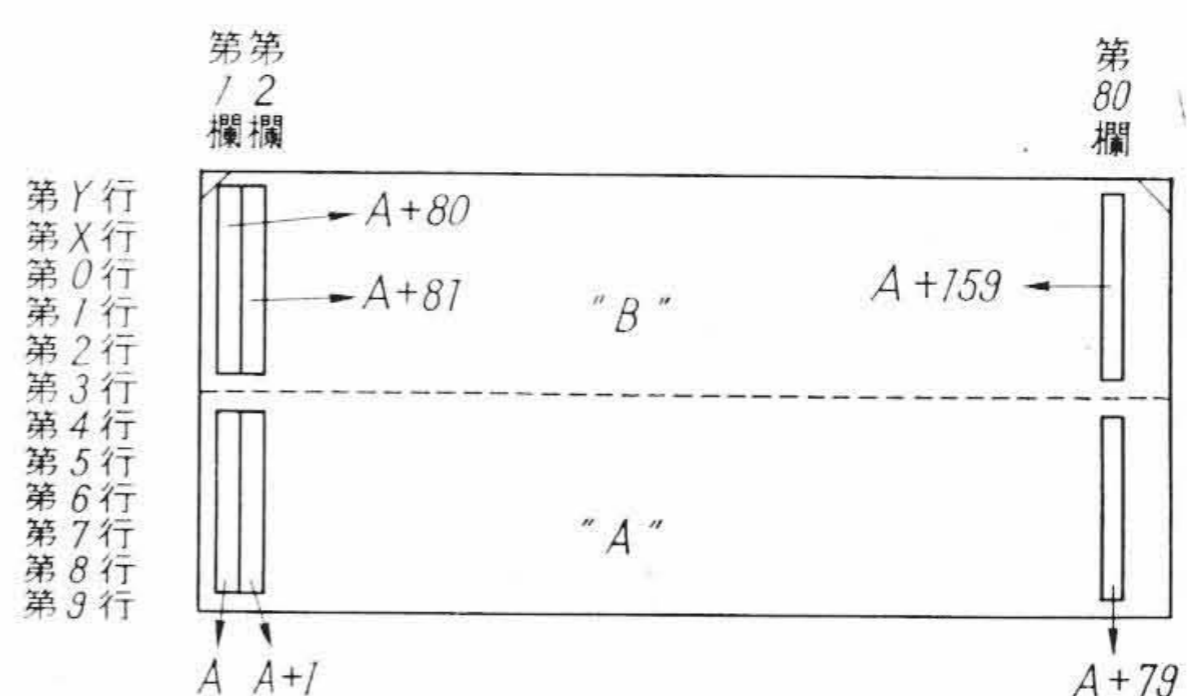
例を示す。TRA 1 1000 2000 に対し，1000 番地の情報を A (2 進表示で 010001) とする。命令が実行されれば 2021 番地の情報が 1000 番地へ書きこまれる。

この命令は通常の意味での単純なコード変換で使用されるのみならず，種々の用途に用いられ，3010 の特長的な命令といってよい。つぎには紙物システムでこの命令を有効に利用した例をカードコー

* 日立製作所神奈川工場



第 1 図 コード変換命令の流れ図



第 2 図 カードの二進読みとり

ド変換サブルーチンにみてみよう。

3. カードコード交換サブルーチン

カードでは 1 字の情報は 1 欄につき 3 個以下のせん孔で表現される。つまり 12 ビットのコードで表現されているといえる。一方計算機内部では通常 6 ビットのコードで表現されるから，ここにコード変換が必要になる。3010 では標準カードコードを標準内部コードに変換する装置を金物的にもっているものとそうでないものがある。またいずれの場合も 12 ビットの情報をすべて読みこむ機能 (二進読みとり) をもっている。A 番地から先へ読みこめという命令に対し，第 2 図に示すように第 1 欄の第 9~4 行が A 番地へ，以下第 80 欄の第 9~4 行が A+79 番地へ，第 1 欄の第 3~Y 行が A+80 番地へ，第 80 欄の第 3~Y 行が A+159 番地にそれぞれ 6 ビットコードとしてしまわれる。

ここでは標準コードのカードを二進読みとりしてプログラムで標準内部コードに変換するサブルーチンの流れ図を第 3 図で，コードおよびテーブルを第 1 表で示す。例を示すと，カードでは C という

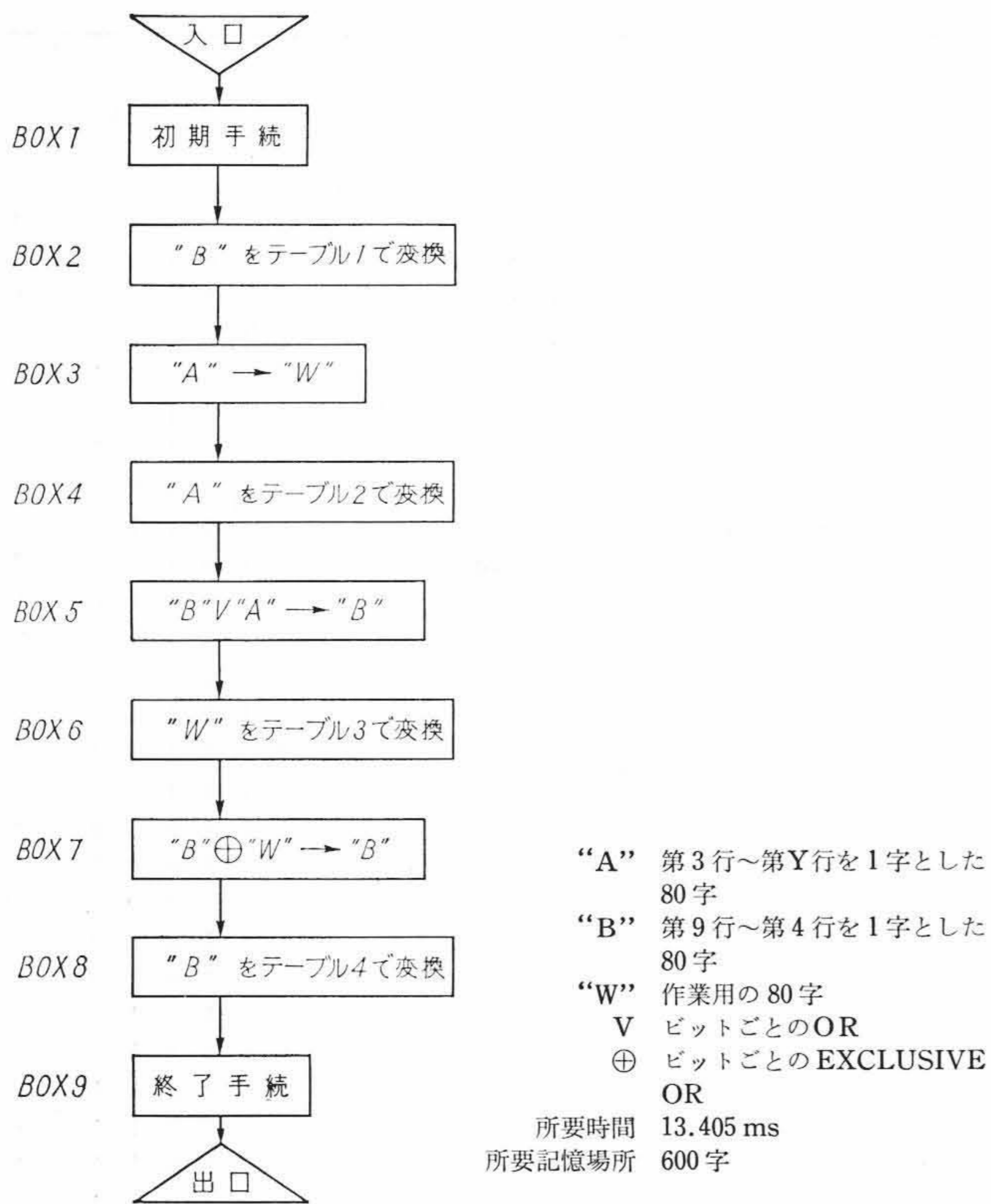
第1表 カ ー ド コ ー ド 変 換 表

	カードイメージ	字	カードイメージ		テーブル1	テーブル2	テーブル3	テーブル4	字
			B	A					
			9 8 7 6 5 4	3 2 1 0 X Y					
0	0 0 0 0 0 0	0	0 0 0 0 0 0	0 0 0 1 0 0	1 1 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 1 0 0	4
1	0 0 0 0 0 1	1	0 0 0 0 0 0	0 0 1 0 0 0	0 0 0 0 0 0	0 0 0 0 0 1	0 0 0 0 0 0	0 1 0 1 0 0	D
2	0 0 0 0 1 0	2	0 0 0 0 0 0	0 1 0 0 0 0	0 0 0 1 0 0	0 0 0 0 1 0	0 0 0 0 0 0	1 0 0 1 0 0	M
3	0 0 0 0 1 1	3	0 0 0 0 0 0	1 0 0 0 0 0				1 1 0 1 0 0	U
4	0 0 0 1 0 0	4	0 0 0 0 0 1	0 0 0 0 0 0	0 0 1 0 0 0	0 0 0 0 1 1	0 0 0 0 0 0	0 0 0 1 0 1	5
5	0 0 0 1 0 1	5	0 0 0 0 1 0	0 0 0 0 0 0		0 0 1 1 0 0	1 0 0 0 0 0	0 1 0 1 0 1	E
6	0 0 0 1 1 0	6	0 0 0 1 0 0	0 0 0 0 0 0		0 0 1 0 1 1	0 0 0 0 0 0	1 0 0 1 0 1	N
7	0 0 0 1 1 1	7	0 0 1 0 0 0	0 0 0 0 0 0				1 1 0 1 0 1	V
10	0 0 1 0 0 0	8	0 1 0 0 0 0	0 0 0 0 0 0	0 0 1 1 0 0	0 0 1 1 0 0	0 0 0 0 0 0	0 0 0 1 1 0	6
11	0 0 1 0 0 1	9	1 0 0 0 0 0	0 0 0 0 0 0		0 0 1 1 0 1	0 0 0 0 0 0	0 1 0 1 1 0	F
12	0 0 1 0 1 0	-	0 0 0 0 0 0	0 0 0 0 0 0		0 0 1 1 1 0	0 0 0 0 0 0	1 0 0 1 1 0	O
13	0 0 1 0 1 1	#	0 1 0 0 0 0	1 0 0 0 0 0				1 1 0 1 1 0	W
14	0 0 1 1 0 0	@	0 1 0 0 0 1	0 0 0 0 0 0		0 1 1 1 1 1	1 0 0 0 0 0	0 0 0 1 1 1	7
15	0 0 1 1 0 1	(0 1 0 0 1 0	0 0 0 0 0 0				0 1 0 1 1 1	G
16	0 0 1 1 1 0)	0 1 0 1 0 0	0 0 0 0 0 0				1 0 0 1 1 1	P
17	0 0 1 1 1 1	&	0 0 0 0 0 0	0 0 0 1 0 1				1 1 0 1 1 1	X
20	0 1 0 0 0 0	&	0 0 0 0 0 0	0 0 0 0 0 1	0 1 0 0 0 0	0 0 1 1 1 1	0 0 0 0 0 0	0 0 1 0 0 0	8
21	0 1 0 0 0 1	A	0 0 0 0 0 0	0 0 1 0 0 1	1 0 0 0 0 0	0 1 0 1 0 0	0 0 0 0 0 0	0 1 1 0 0 0	H
22	0 1 0 0 1 0	B	0 0 0 0 0 0	0 1 0 0 0 1	1 0 0 1 0 0	0 1 0 1 0 1	0 0 0 0 0 0	1 0 1 0 0 0	Q
23	0 1 0 0 1 1	C	0 0 0 0 0 0	1 0 0 0 0 1				1 1 1 0 0 0	Y
24	0 1 0 1 0 0	D	0 0 0 0 0 1	0 0 0 0 0 1	1 0 1 0 0 0	0 1 0 1 1 0	0 0 0 0 0 0	0 1 1 0 1 0	+
25	0 1 0 1 0 1	E	0 0 0 0 1 0	0 0 0 0 0 1				1 0 1 0 1 0	EI
26	0 1 0 1 1 0	F	0 0 0 1 0 0	0 0 0 0 0 1				1 1 1 0 1 0	EB
27	0 1 0 1 1 1	G	0 0 1 0 0 0	0 0 0 0 0 1				0 0 1 0 1 1	#
30	0 1 1 0 0 0	H	0 1 0 0 0 0	0 0 0 0 0 1				0 1 1 0 1 1	•
31	0 1 1 0 0 1	I	1 0 0 0 0 0	0 0 0 0 0 1				1 0 1 0 1 1	¥
32	0 1 1 0 1 0	+	0 1 0 0 0 0	0 1 0 0 0 1				1 1 1 0 1 1	,
33	0 1 1 0 1 1	.	0 1 0 0 0 0	1 0 0 0 0 0					
34	0 1 1 1 0 0	;	0 1 0 0 0 1	0 0 0 0 0 0				0 1 1 1 1 1	古
35	0 1 1 1 0 1	:	0 1 0 0 1 0	0 0 0 0 0 0					
36	0 1 1 1 1 0	,	0 1 0 1 0 0	0 0 0 0 0 0					
37	0 1 1 1 1 1	-						1 1 0 0 0 1	/
40	1 0 0 0 0 0	-	0 0 0 0 0 0	0 0 0 0 1 0	1 0 1 1 0 0	0 1 0 1 1 1	0 0 0 0 0 0	0 0 1 1 0 0	@
41	1 0 0 0 0 1	J	0 0 0 0 0 0	0 0 1 0 1 0		0 1 1 0 0 0	0 0 0 0 0 0	0 1 1 1 0 0	;
42	1 0 0 0 1 0	K	0 0 0 0 0 0	0 1 0 0 1 0		0 1 1 0 0 1	0 0 0 0 0 0	1 0 1 1 0 0	*
43	1 0 0 0 1 1	L	0 0 0 0 0 0	1 0 0 0 1 0				1 1 1 1 0 0	%
44	1 0 0 1 0 0	M	0 0 0 0 0 1	0 0 0 0 1 0		0 1 1 0 1 0	0 0 0 0 0 0	0 0 1 1 0 1	(
45	1 0 0 1 0 1	N	0 0 0 0 1 0	0 0 0 0 1 0				0 1 1 1 0 1	:
46	1 0 0 1 1 0	O	0 0 0 1 0 0	0 0 0 0 1 0				1 0 1 1 0 1	ED
47	1 0 0 1 1 1	P	0 0 1 0 0 0	0 0 0 0 1 0				1 1 1 1 0 1	●
50	1 0 1 0 0 0	Q	0 1 0 0 0 0	0 0 0 0 1 0				0 0 1 1 1 0)
51	1 0 1 0 0 1	R	1 0 0 0 0 0	0 0 0 0 1 0				0 1 1 1 1 0	,
52	1 0 1 0 1 0	EI	0 1 0 0 0 0	0 1 0 0 1 0				1 0 1 1 1 0	EF
53	1 0 1 0 1 1	¥	0 1 0 0 0 0	1 0 0 0 1 0				1 1 1 1 1 0	=
54	1 0 1 1 0 0	*	0 1 0 0 0 1	0 0 0 0 1 0				0 0 1 0 0 1	9
55	1 0 1 1 0 1	ED	0 1 0 0 1 0	0 0 0 0 1 0				0 1 1 0 0 1	I
56	1 0 1 1 1 0	EF	0 1 0 1 0 0	0 0 0 0 1 0				1 0 1 0 0 1	R
57	1 0 1 1 1 1							1 1 1 0 0 1	Z
60	1 1 0 0 0 0	"	0 0 0 0 0 0	0 0 0 1 1 0				0 0 1 0 1 0	-
61	1 1 0 0 0 1	/	0 0 0 0 0 0	0 0 1 1 0 0				0 1 0 0 0 0	&
62	1 1 0 0 1 0	S	0 0 0 0 0 0	0 1 0 1 0 0				1 0 0 0 0 0	-
63	1 1 0 0 1 1	T	0 0 0 0 0 0	1 0 0 1 0 0				0 0 0 0 0 0	0
64	1 1 0 1 0 0	U	0 0 0 0 0 1	0 0 0 1 0 0				0 1 0 0 1 0	B
65	1 1 0 1 0 1	V	0 0 0 0 1 0	0 0 0 1 0 0				1 0 0 0 1 0	K
66	1 1 0 1 1 0	W	0 0 0 1 0 0	0 0 0 1 0 0				1 1 0 0 1 0	S
67	1 1 0 1 1 1	X	0 0 1 0 0 0	0 0 0 1 0 0				0 0 0 0 1 1	3
70	1 1 1 0 0 0	Y	0 1 0 0 0 0	0 0 0 1 0 0				0 1 0 0 1 1	C
71	1 1 1 0 0 1	Z	1 0 0 0 0 0	0 0 0 1 0 0				1 0 0 0 1 1	L
72	1 1 1 0 1 0	EB	0 1 0 0 0 0	0 1 0 1 0 0				1 1 0 0 1 1	T
73	1 1 1 0 1 1	,	0 1 0 0 0 0	1 0 0 1 0 0				1 1 0 0 0 0	"
74	1 1 1 1 0 0	%	0 1 0 0 0 1	0 0 0 1 0 0				0 0 0 0 0 1	1
75	1 1 1 1 0 1	。	0 1 0 0 1 0	0 0 0 1 0 0				0 1 0 0 0 1	A
76	1 1 1 1 1 0	=	0 1 0 1 0 0	0 0 0 1 0 0				1 0 0 0 0 1	J
77	1 1 1 1 1 1							0 0 0 0 1 0	2

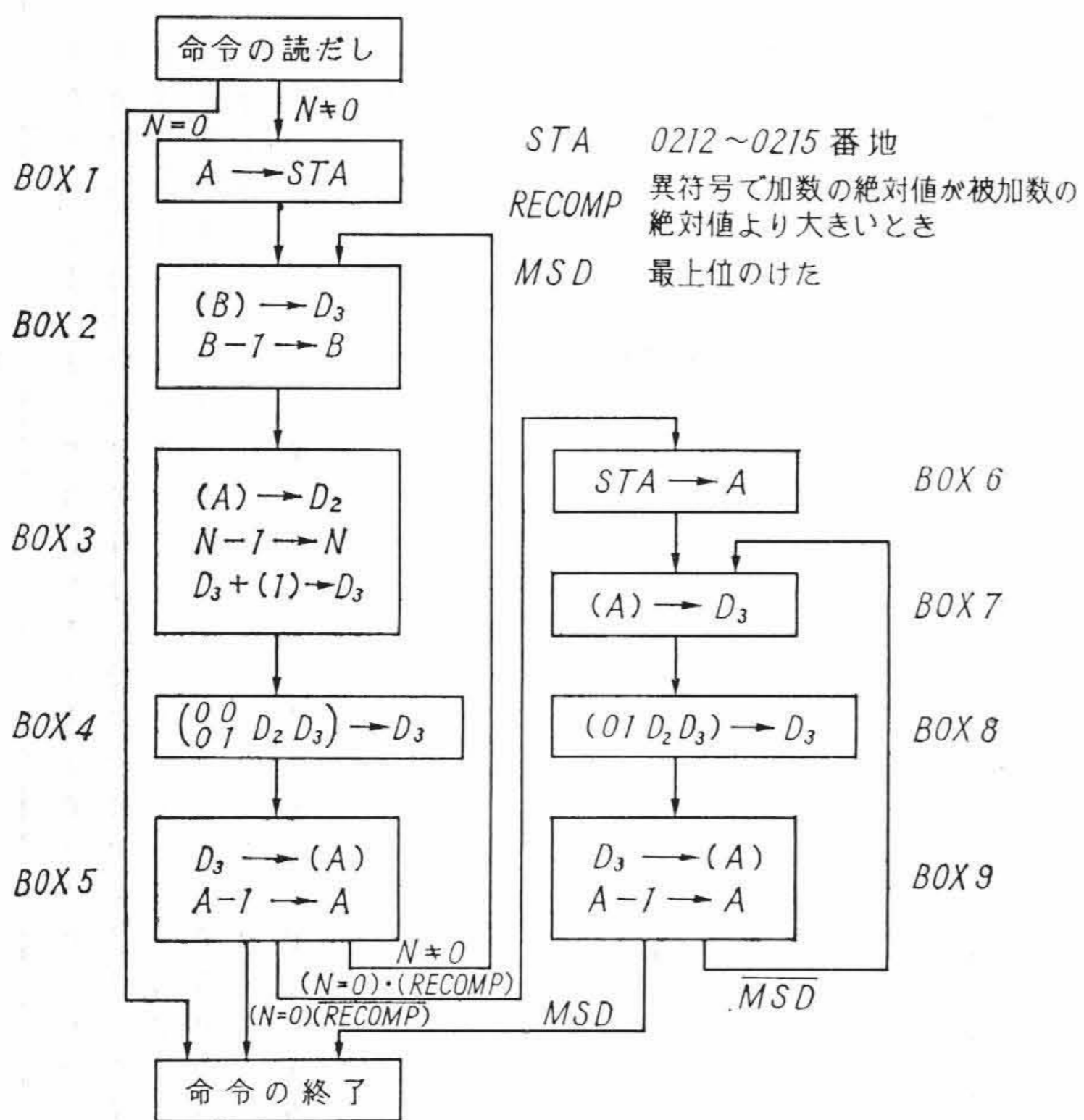
字は3~Yの行にせん孔があるから、下のようにして (010011) という内部コードに変換される。

- 000000 “B”の字
- (a) 110000 テーブル1の(0)₈で変換 (BOX2)
- 100001 “A”の字
- (b) 011000 テーブル2の(1)₈で変換 (BOX4)

- (c) 111000 (a) OR (b) (BOX5)
- (d) 000000 テーブル3の(1)₈で変換 (BOX6)
- 111000 (c) EXOR (d) (BOX7)
- 010011 テーブル4の(70)₈で変換 (BOX8)



第 3 図 カードコード変換サブルーチンの流れ



第 4 図 加算の流れ図

4. 加減算命令

3010では字単位の加減算命令に対して金物ではいわゆる加減算回路をもたず、0~199番地を加減算テーブルとして、テーブル技術を用いて加減算を実行している。加算を例にとって説明すると命令の形式は下のおりである。

- OP ADD のコード
- N 0~45 フィールドの字数
- A 被加数および和のフィールドの最右端アドレス
- B 加数のフィールドの最右端アドレス

この命令が実行される過程の流れ図を第 4 図に示す。

ADD 2 1001 2001

という命令を例にとる。記憶場所の内容が下のおりであったとすると

番 地	1000	1001	2000	2001
記憶場所の内容	2	8	3	5

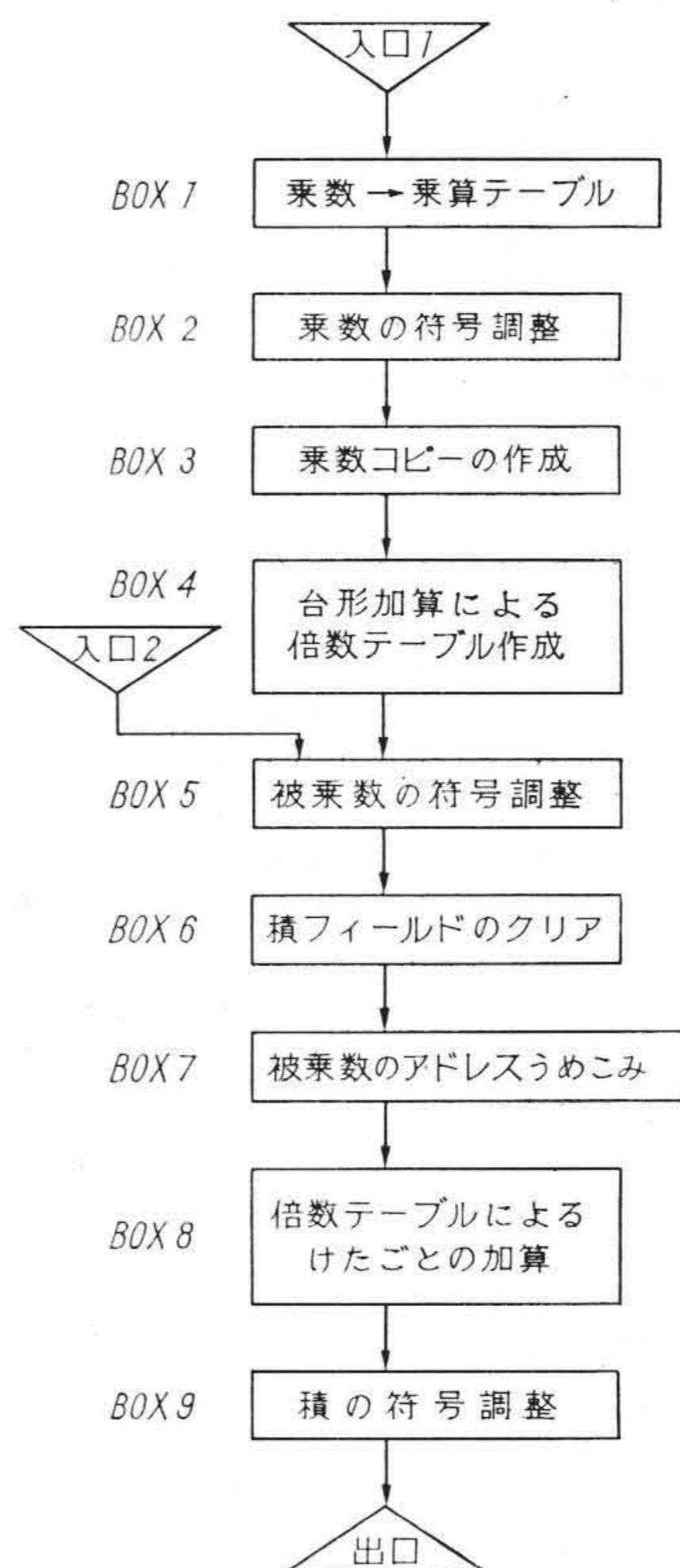
上の命令は 28+35 を実行することになる。Aレジスタを退避したあと、2001番地の内容5を読みだして補助レジスタ D₃ へ入れる。次に1001番地の内容8を読みだして補助レジスタ D₂ へ入れる。次に同符号の加算では和テーブル(0~99番地)の中で85番地の内容(010011)を読みだし、数字部分である3を D₃ へ入れる。2⁴ビットの1はけた上げを示している。次に D₃ の内容3を1001番地へ書きこむ。次に2000番地の内容3を読み D₃ へ入れる。そして1000番地の内容2を読んで D₂ へ入れると同時に前のけたでけた上げがあったから D₃ を1ふやすので D₃ は4となる。次は24番地の内容(000110)を読みだして再び1000番地へ入れる。こうして1000, 1001番地に63がしまわれることになる。

異符号の加算では、差テーブル(100~199番地)が使用される。またこの場合は修正の必要な場合もあり、流れ図ではこの場合も示してある。減算もほとんど同じ方法で実行される。

5. 乗算サブルーチン

3010の処理装置303, 304は金物では乗算命令をもっていない。したがって加算の反復によってサブルーチンで乗算を実行する。しかし単に乗数の反復加算ではある程度以上のけたのものではあまり能率がよくないので、3010では乗数の0~9倍のテーブルを作り、被乗数の各けたごとにテーブルをひいて1けたずつずらせながら加算を行なう方法を採用している。第5図はサブルーチンの流れ図を示す。サブルーチンの入口は2箇所あって、入口1は新たな乗数を用いる場合、入口2は前に用いた乗数を用いる場合である。乗数、被乗数、積はいずれも特定の記憶場所を用いる。

乗数テーブルの作成にいわゆるはしご形加算を用いている。乗数テーブルの例を第6図に示す。まず乗数を X1 にうつしたあと、X1



第 5 図 乗算サブルーチンの流れ図

X 0	0 0 0 0 0 0	BOX 3
	0 0 0 0 0 0	BOX 4
X 1	0 1 9 6 3 8	BOX 3
	0 1 9 6 3 8	BOX 4
X 2	0 1 9 6 3 8	BOX 3
	0 3 9 2 7 6	BOX 4
X 3	0 1 9 6 3 8	BOX 3
	0 5 8 9 1 4	BOX 4
X 4	0 1 9 6 3 8	BOX 3
	0 7 8 5 5 2	BOX 4
X 5	0 1 9 6 3 8	BOX 3
	0 9 8 1 9 0	BOX 4
X 6	0 1 9 6 3 8	BOX 3
	1 1 7 8 2 8	BOX 4
X 7	0 1 9 6 3 8	BOX 3
	1 3 7 4 6 6	BOX 4
X 8	0 1 9 6 3 8	BOX 3
	1 5 7 1 0 4	BOX 4
X 9	0 1 9 6 3 8	BOX 3
	1 7 6 7 4 2	BOX 4

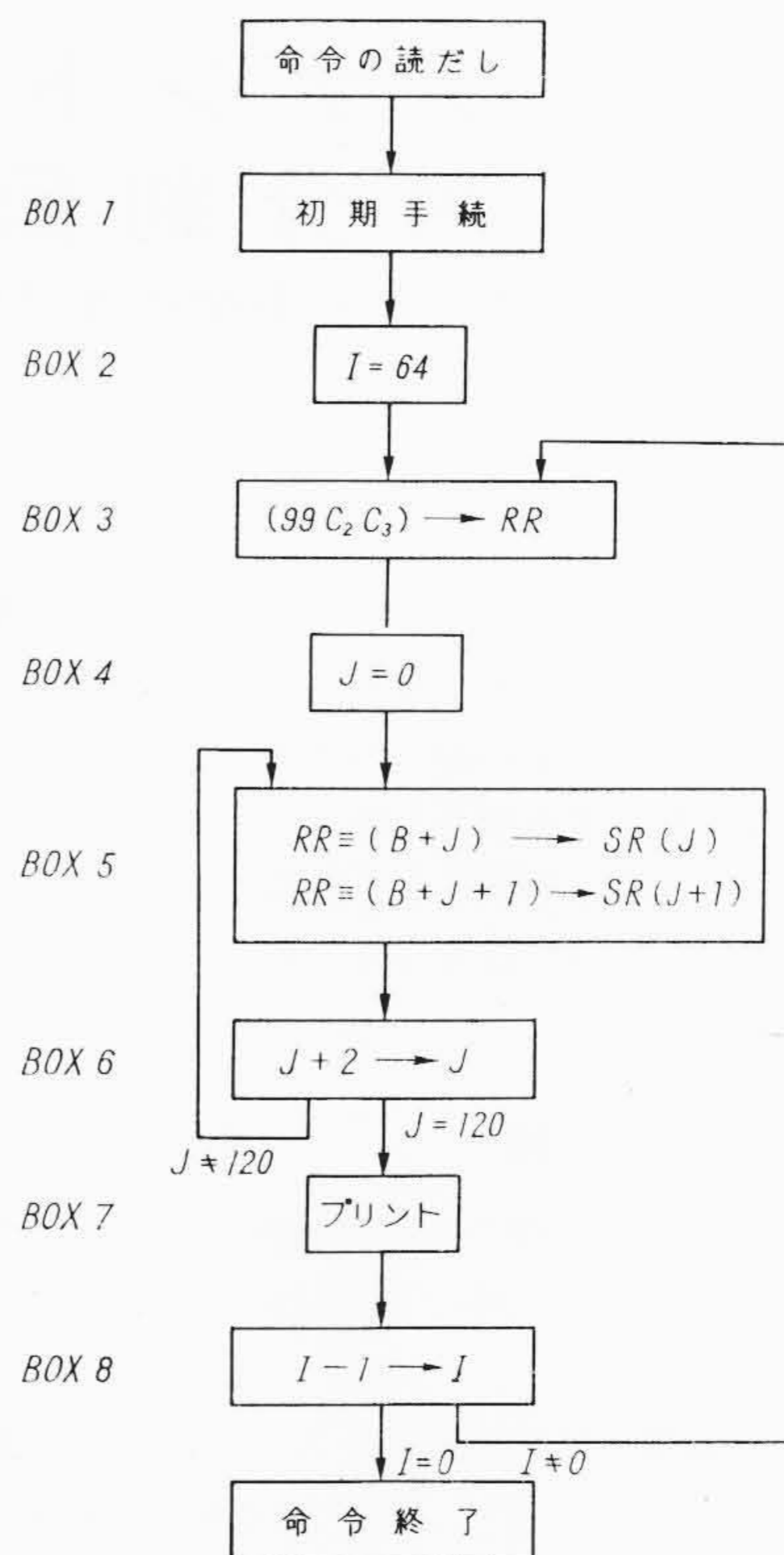
第6図 乗数の倍数テーブル

と同じものを X2~X9 へ移す。次に X1+X2→X2, X2+X3→X3, …… を行なえば X1~X9の倍数テーブルができる。演算所要時間は下のおりである。

乗数, 被乗数のけた数	入口 1 (ms)	入口 2 (ms)
5	4.2	1.7
8	6.7	3.2
10	8.6	4.5
16	15.8	9.6

6. ラインプリンタ制御方式

3010にはドラム式のプリンタが接続されている。プリントドラムには一周 64 (スペース 1 を含む) の活字がある。次に印字可能なプリントドラムの位置を示すための位置カウンタ (6 ビット) C_2C_3 があり、通常はプリントすべきフィールドにおいてこの位置カウンタに示されるコードと一致のとれる字の位置をしらべ、その位置に印字する方法をとっている。しかし 3010 では他の機種との関連から、ラインプリンタでのコードと計算機内部のコードが異なっており、そのためにラインプリンタの制御に際しては自動的に特定番地 (1 万字の記憶容量のときは 9900 番地以降, 2 万字のときは 19900 番地以降) から先のプリントテーブルでコード変換を行なっている。この制御方式の流れ図を第 7 図で示す。次に印字すべき字を示す位置カウンタが (000100) とすると 9904 番地の内容を読みだすと標準の場合 (000001) が読みだされ、参照レジスタ (RR) へいれられる。この場合は内部コードで 1 という字を印字することになる。そのため B アドレスで指定した記憶場所から先 120 字について RR との一致を調べ、対応するシフトレジスタ (SR) を一致がとれた場合セットしていく。セットされた SR に対応したハンマはプリンタのタイミングで駆動され、プリントフィールドのうち 1 という字の位



第7図 ラインプリンタ制御の流れ図

置に 1 が印字される。この動作を 64 種の字について実行する。

スペースに対応するテーブルのコードは (001111) であるが、これを他のコードに対応するテーブルに書きこむとそのコードも印字されなくてスペースとなり、編集のうえでいくらか妙味もっている。

7. 結 言

以上、五つの点について 3010 でテーブル技術を使用している例を示し解説した。金物的なテーブル演算技術は字単位の計算機では比較的容易に構成できること、3010 で採用している基本回路の論理特性からは 10 進の加減算回路を比較的構成しにくいことなどを説明しなければどうしてテーブル技術を採用したかを説明したことにならないが、ここでは割愛した。紙物での応用例もソーティングをはじめさらに興味深いものも多いが、ここではもっとも基本的なものの紹介にとどめた。われわれはテーブル技術の種々の適用例を通していろいろ検討した結果、これらは 3010 に盛られた技術的に成功をおさめた成果と判断しており、これらの技術を吸収してわれわれのものにしたいと考えている。

参 考 文 献

- (1) HITAC 3010 命令語の説明 (マニュアル)
- (2) HITAC 3010 状態流れ図 (マニュアル)
- (3) HITAC 3010 サービスルーチン (マニュアル)
- (4) HITAC 3010 コントロールロジック (マニュアル)