—ノート—

FACOM230－25カード・イニシャル・プログラム・ローダーの試作

日　置　伸　一

Trial Construction of Card·Initial·Program Loader for FACOM230-25

Shinichi　HIOKI

〈概　　要〉

最近（第三世代以降）のコンピュータは，通常オペレーティング・システムがメーカーから提供され，ユーザはオペレーティング・システムの管理下でのみコンピュータを使う事を強いられている．

ユーザーはノー・クラッチの自動車をアクセルとブレーキとハンドルだけで運転していて，万一エンジンの調子がおかしくてもボンネットを開ける事すら許されないでいる様なものである．

筆者は，イニシャル・プログラム・ローダをカード・ベースで作ることによって FACOM230—25のハードウェアの極一部を覗き見ることを試みたのでここに報告します．

ま　え　が　き

当大学に電子計算機が初めて設置されたのは今から10年前（昭和38年）で，記憶容量は2000語で演算時間もmS(10^{-3}秒）オーダ，使われている素子はトランジスタで名前をNEAC2203（以下N—03と略す）と称するものであった．そして10年後，記憶容量32Kバイト(16000語)，演算時間μS（10^{-6}秒）オーダ，ICが素子として使われていて，FACOM230—25（以下 F—25 と略す）という名前がついている．

ところが，この２つの計算機で例えば25元連立一次方程式を解いてみると，計算時間はそれほど差がない．これはN—03ではプログラムを作るのに機械語を使っているが，F—25 では FORTRAN 言語を使用しているという条件が異っているが，F—25のアセンブラ言語（機械語とほぼ1対1に対応する）を使ってプログラミングしてもオペレーティング・システム（以下 OS と略す）の管理下でしか動かせないという制約条件と F—25のハードウェアそのものが OS を意識して設計されている為に，結局 FORTRAN コンパイラがオブジェクト・プログラムの効率を考慮していれば，いずれも OS の下で働くので大差が出ないと思われます．

そこで，OS とは全く関係のないプログラム，即ち機械語プログラムに注目する事になりました．この時第一に問題となるのが機械語プログラムを読み込む為のプログラム，即ちイニシャル・プログラム・ローダが必要であるということ，しかもカード・ベースで行ないたいということでした．このノートはカードによるイニシャル・プログラム・ローダを試作したので，その内容を述べるものです．

0.　F—25のハードウェア

F—25のシステム構成の概略は 図0—1の様になっている．

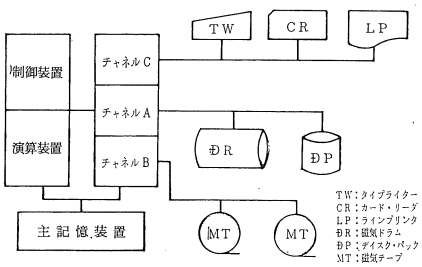


図　0—1　システム構成

〈ハードウェア固定領域〉

主記憶装置にはハードウェア的にあらかじめ占有される領域があって，これをハードウェア領域と呼んでいる．

ハードウェア領域にはハードウェア固定領域とサブチャネル領域があり，これ等の領域を特別の目的に使用す

0～1：CAW（Channel Address wōrd）
I/Ō（入出力装置）を動作させる為に，チャネルに与える指命（CCW）が格納されているアドレスをセットする領域

4～7：CCW 1（Channel Cōmmand Wōrd）
イニシャル・ロードを行なう為のチャネル・コマンド・ワードのための領域

8～B：CCW 2
CCW 1と同様

C～F：CSW（Channel Staus Wōrd）
入出力動作の結果が格納される領域

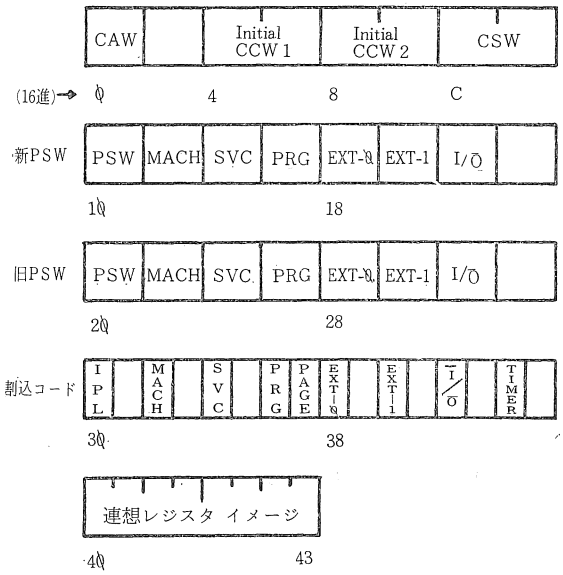


図　0－2
ハードウエア固定領域

10～1F：新PSW（Prōgram Staus Wōrd）
計算機の状態はプログラム・ステイタス・ワード（PSW）と呼ばれるレジスタにまとめられているが割込が起ったとき，この番地に格納されている内容が自動的にアクテイブＰＳＷにセットされる

20～2F：旧PSW
割込みが起きた時，その時のＰＳＷの内容が，この番地へ自動的に格納される

30～3F：割込コード
割込みが起きた時，その詳細な情報を示すコードが格納される

ることによって計算機の制御をスムーズに行なおうとするのがF—25の設計思想のようである．

サブチャネル領域は400～440（16進）番地に領域が確保されていて，ハードウェア固定領域は0～43（16進）番地に確保されている．

ハードウェア固定領域は図0—2に示す通りである．

〈イニシャル・ローディング機能〉

次にF—25のハードウェアがもっているイニシャル・ローディング機能について述べる．

イニシャル・ローディングに際しては，I／O装置のチャネルおよび機番を本体のオペレータ・パネル上にあるロータリー・スイッチによって選択して，次にロード・キィー（LOAD）を押す．この操作が行なわれるとシステムはリセットされロードランプが点灯する．引き続き選択されたI／O装置から読込みを開始し，読込みが正常に完了すると新PSWがアクティブ PSW にロードされ cpu は動作状態に入り，LOAD ランプは消灯する．

選択されたI／O 装置から読込みを開始し，最初に読込まれた36バイトが主記憶装置の0～11（16進）番地に格納される．この際メモリプロテクションは無視される．

4～7（16進）番地に読込まれた4語はCCW として次の入出力操作の為に使用される．このCCW の中でチェイニングが指定されると8（16進）番地のCCWによって操作が実行される．

36バイトの読込みが終了するとCCW1によって次の読込み動作に移る．チェイニング指定があれば，これを行ない全てのチェイニングが終ったところで10（16進）番地の新 PSW をアクティブPSW にロードして計算機は

このPSW の制御下に入り，イニシャル・ローディング機能を終了する．

この動作中，入出力割込みは発生せず入出力操作やPSW のロードが正規に行なわれない場合は計算機は動作状態にならずロード・ランプは消灯しない．

1. カードを1枚読む

計算機にカードを読むプログラムが入っていなくとも，0.で述べたイニシャル・ローディング機能を用いれば，とにかく1枚のカードを読む動作を行なう事は可能である，（但し，最初の36バイトが主記憶に格納されるだけである）

ところが，この様にして主記憶装置に読込まれた情報はパンチカードのホール・パターン（カードにパンチされたホールの位置と個数）を一定の法則で変換されたものがカード読取装置から主記憶装置内へ転送されるだけである．

80欄カードは12のホールポジションが1桁を構成している．一方，主記憶は8bit（1BYTE）で1文字（EBCDICコード）を構成している．このため12のホールポジションを 8bit に変換する場合には圧縮カードコードと呼ばれるコードに変換される．この圧縮カードコードは計算機メーカー特有のものでメーカー毎に違っている．（EBCDICコードに変換するもや，ホール・パターンをそのまま転送するハードウェアも有る）

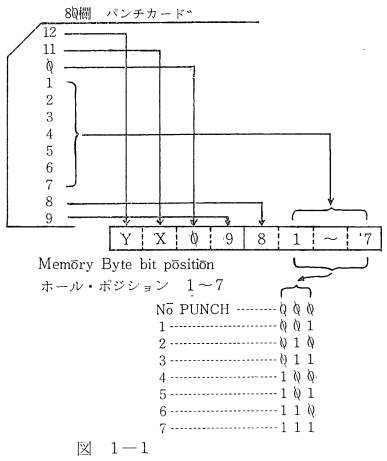


図 1—1

b5～b8 \ b1～b4	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111
0 0000	SP	9	0	Z	(マイナス) —	R		ヤ	&	I	± ア	ケ		ネ		
1 0001	1		/		J				A		ア		タ			
2 0010	2		S		K		ヘ		B		イ		チ			
3 0011	3		T		L		ホ		C		ウ		ツ			
4 0100	4		U		M		マ		D		エ		テ			
5 0101	5		V		N		ミ		E		オ		ト			
6 0110	6		W		O		ム		F		カ		ナ			
7 0111	7		X		P		メ		G		キ		ニ			
8 1000	8		Y		Q		モ		H		ク		ヌ			
9 1001			`										ソ			
A 1010	:				↑		ユ				コ		ソ		レ	
B 1011	#		,		¥				.						ロ	
C 1100	@		%		*		ヨ		<		サ				ワ	
D 1101	▽		(ハイフン) -		)		ラ		(		シ		ハ		ン	
E 1110	=		>		;		リ		+		ス		ヒ		ミ	
F 1111	〃		?		¬		ル		|		セ		フ		。	

表　1—1　圧縮カードコード逆変換表

- ○ $b_1\,b_2\,b_3\,b_4$ | $b_5\,b_6\,b_7\,b_8$
- ○ SPはSpace（ブランク）
- ○ 対応コードが書いてないところは図1—1によりマルチパンチを行なえばよい
- ○ 例　内部bitパターンを〝1000 0001〟としたければ〝A〟をパンチすればよい

ここではF—25で使用している圧縮コードを図1—1，表1—1に示す．

〈圧縮カードコード〉

圧縮カードコードの変換則は下図→図1—1の通りである．

〈カードを1枚読むには〉

上述した圧縮カードコードに変換される事を承知していれば0.で述べたイニシャル・ローディングを行なえばよい．

即ち．カード読取装置のチャネル番号（C)*，機番（0)* を本体のオペレータ・パネルのロータリ・スイッチで指定してロード・キイを押せばイニシャル・ローディングが行なわれる．

（注　*印は当大学計算センターに於るチャネル番号および機番である）

2. 続いてカードを1枚読む

〈BOOT1〉

1.でイニシャル・ローディング機能を使ってカードを1枚読んでも，その場合読まれたカードの36バイトが圧縮カードコードに変換されて主記憶装置の0番地から11（16進）番地へ格納されるだけである．

この場合，主記憶の0～11（16進）番地に格納された情報だけで“次のカードを読む”ことが出来なければならない．

即ち，最初に1枚カードを読む事によって，圧縮カードコードに変換された結果が続いてカードを1枚（図2—1でCIPL…カード4枚を読む為にはカードを1枚読む必要がある）読む命令を構成していなければならない．この為まず機械語でプログラムを作り，1.の圧縮カードコード逆変換表を利用してカードをパンチする．この様にして作ったパンチカードをBOOT1と呼ぶことにする．

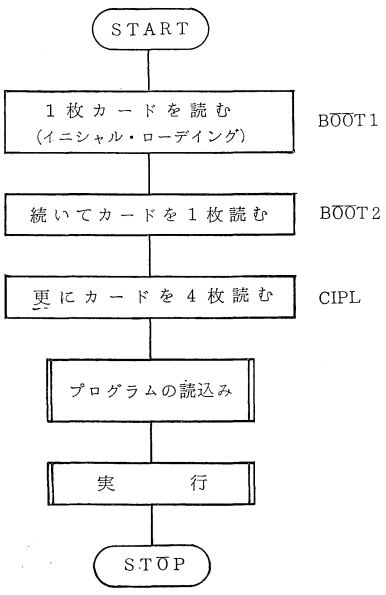


図　2—1　BOOT1, BOOT2, CIPL

〈BOOT2〉

BOOT1がイニシャル・ローディングによって読込まれと，直ちに次の1枚のカードが読込まれる．

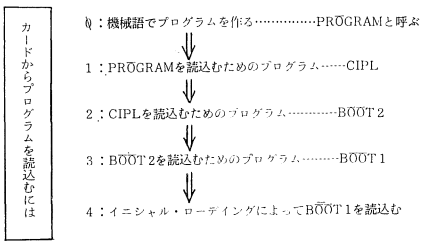
この1枚のカードをBOOT2と呼び，次の機能をもたせる．

(1) チャネルに関してメモリ・プロテクションを解除する．

(2) 次の4枚のカードを読込む．

これで，4枚のカードからなるカード・イニシャル・プログラム・ローダを読込む準備が整ったことになる．

ここまでをまとめてみると，



となる．目的のプログラムを読込むのに4ステップを要するのは，イニシャル・ローディングによって読込ませうるのは36バイト（BOOT1）であり，36バイト（ハードウェア固定領域）の中にはカードを4枚（CIPL）読ませる情報は入り切らないのでBOOT2を使って，中継することによってCIPLを読込む．

BOOT1，BOOT2は図2—2および図2—3に一例を示す．

3. 更にカードを4枚読む

ここで読む4枚のカードが，機械語で作ったプログラムを16進数でカードにパンチしたものを読込み，計算機の中に機械語命令を作り出すプログラム，即ちイニシャル・プログラム・ローダである．このプログラム（カード4枚）をCIPLと呼ぶことにする．

CIPLがもつ機能は次の6種である．

(1) cpuに関してメモリ・プロテクションの解除．

(2) 格納開始番地をパンチしたカードを読込む．
（／S××××：××××が16進表示の格納開始番地）

(3) 格納開始番地の計算．

(4) プログラム・カードを読込み，機械語命令を主記憶内に作り出し格納開始番地から順に格納する．

(5) ／R××××で始まるカードを読込んだ場合には実行開始番地を計算する．

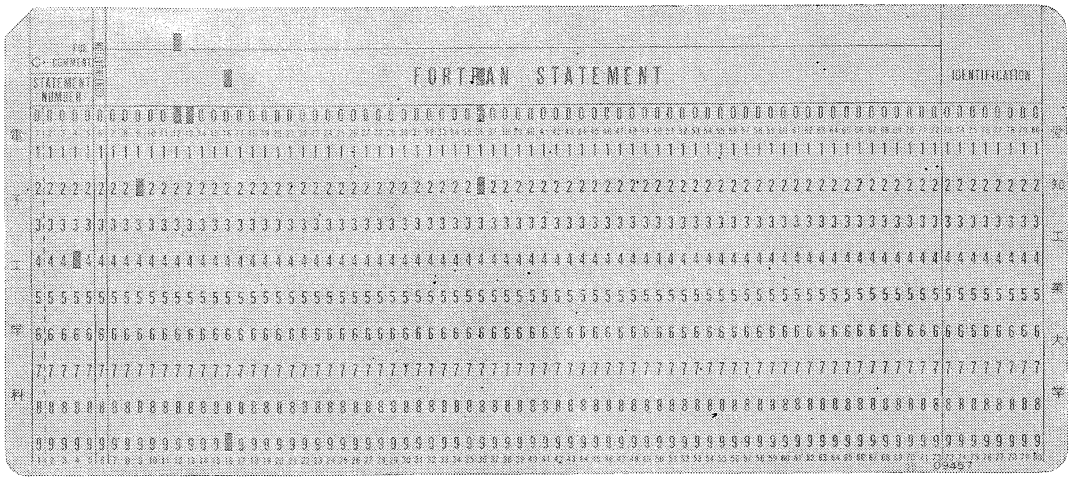


図2—2　BŌŌT　1

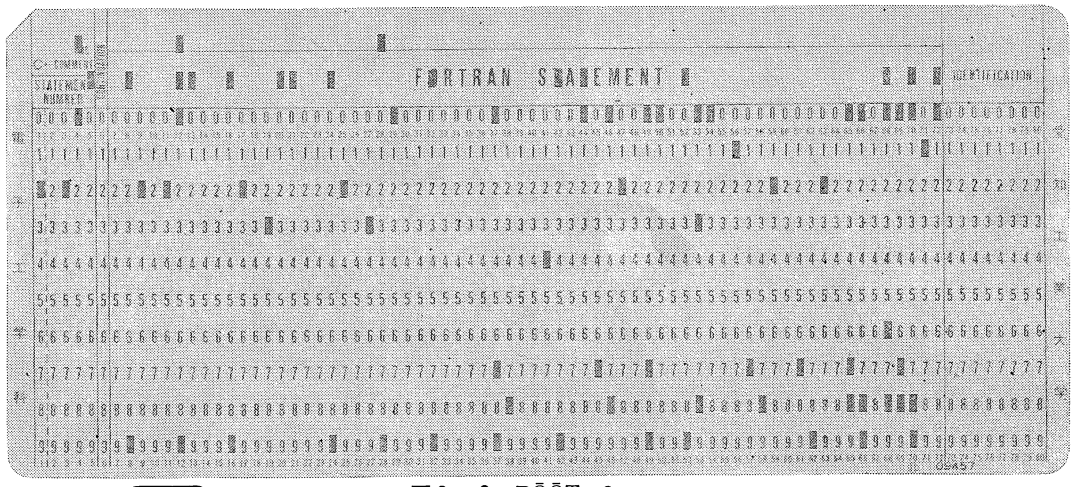


図2—3　BŌŌT　2

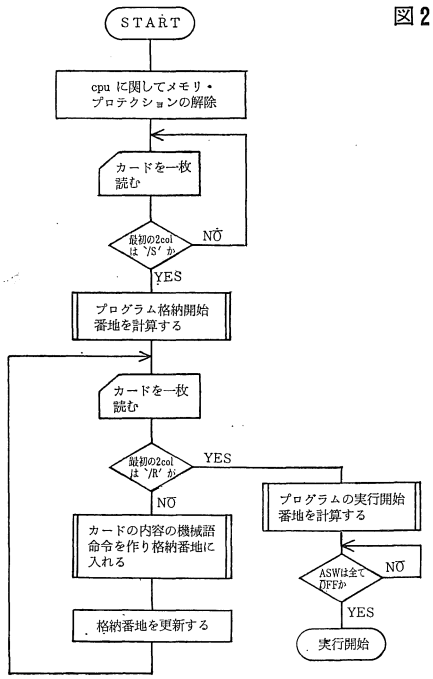
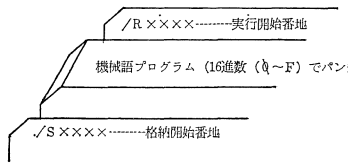
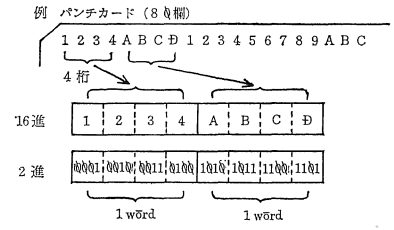


図　3—1　CIPLのフローチャート

(6)　プログラム読込み後直ぐ実行を開始するかどうかASW(オルターネーション・スイッチ)によつて決める．図3—1はCIPLのフローチャートである．CIPLのカードホールパターンを図3—2に示す．





カード上の0～9，A～Fを0000～1001，1010～1111に変換して，カード上の4桁を1word (16bit) に格納する

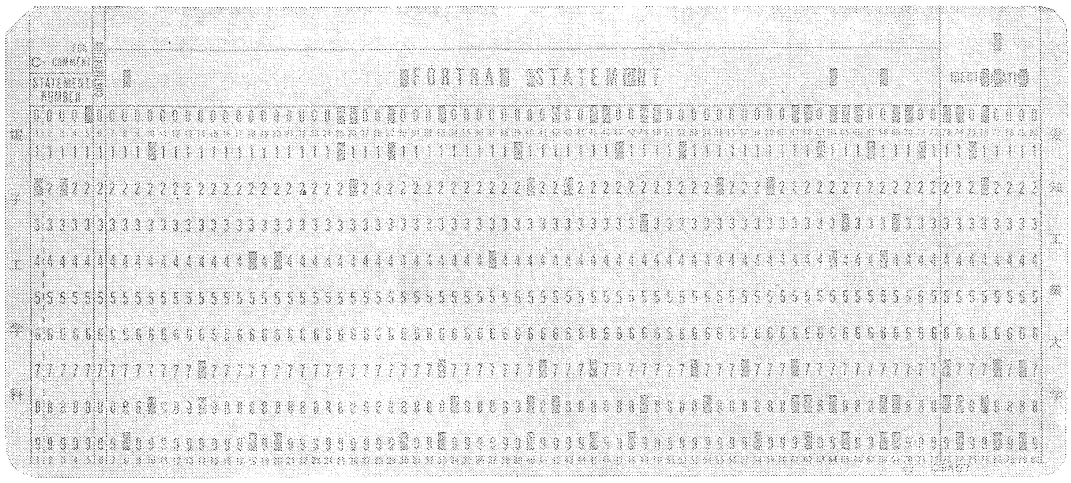


図3—2 CIPL その1

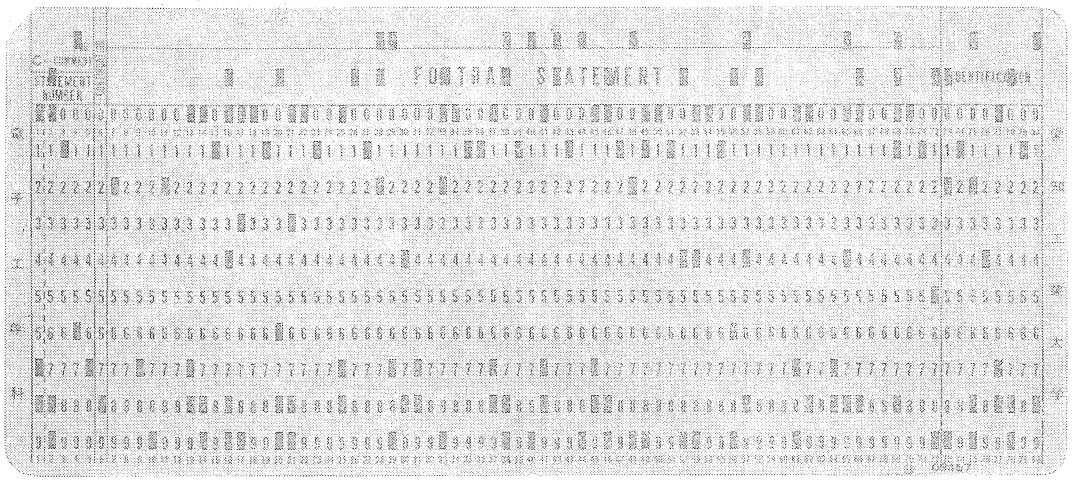


図3—2 CIPL その2

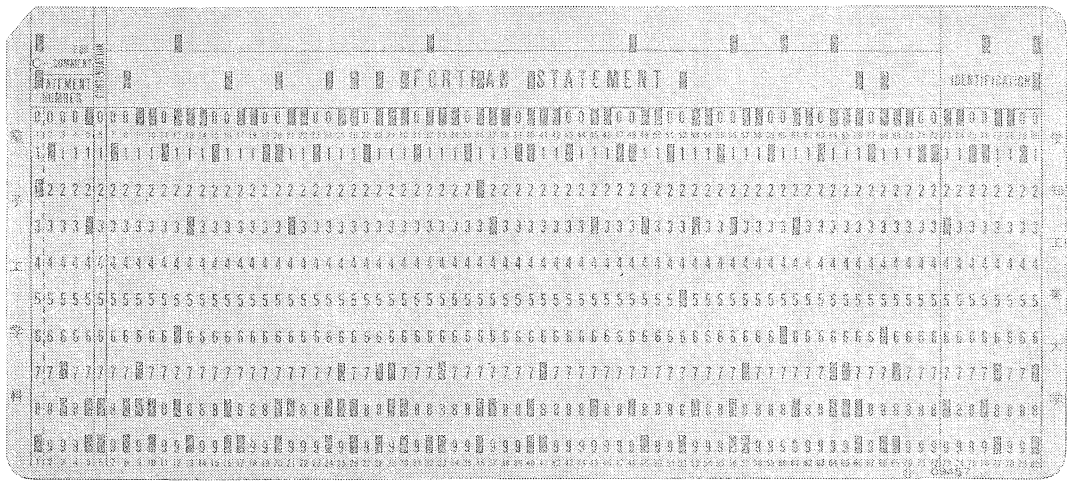


図3—2 CIPL その3

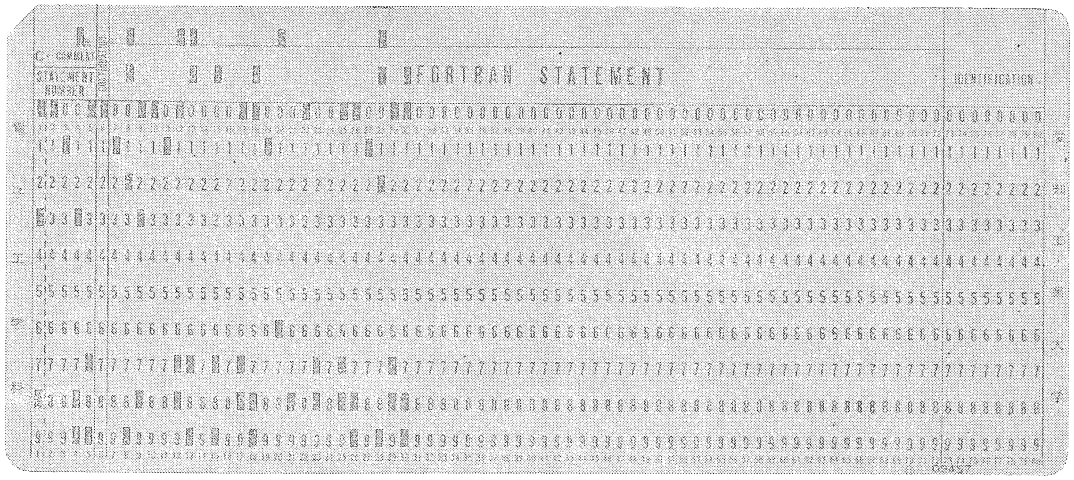


図3—2　CIPL　その4

あ と が き

カードによるイニシャル・プログラム・ローダを試作することによって，F—25のハードウェアの入出力部分の一部，即ち入出力命令と入出力コマンドおよび入出力割込みについての概要を知ることが出来た．

尚このノートのテーマに関して適切な御助言を下さった当大学計算センター皆福講師並びに富士通名古屋営業所CE諸兄に対し深謝する次第です．

参 考 文 献

1. 石上孝雄・成毛吉一共著「計算機の基本動作」竹内書店（1971）
2. FACOMマニュテル「FACOM230—25／35／45 ハードウェア総合解説編Ⅰ」（EX-011-1-3）富士通株式会社（1971）