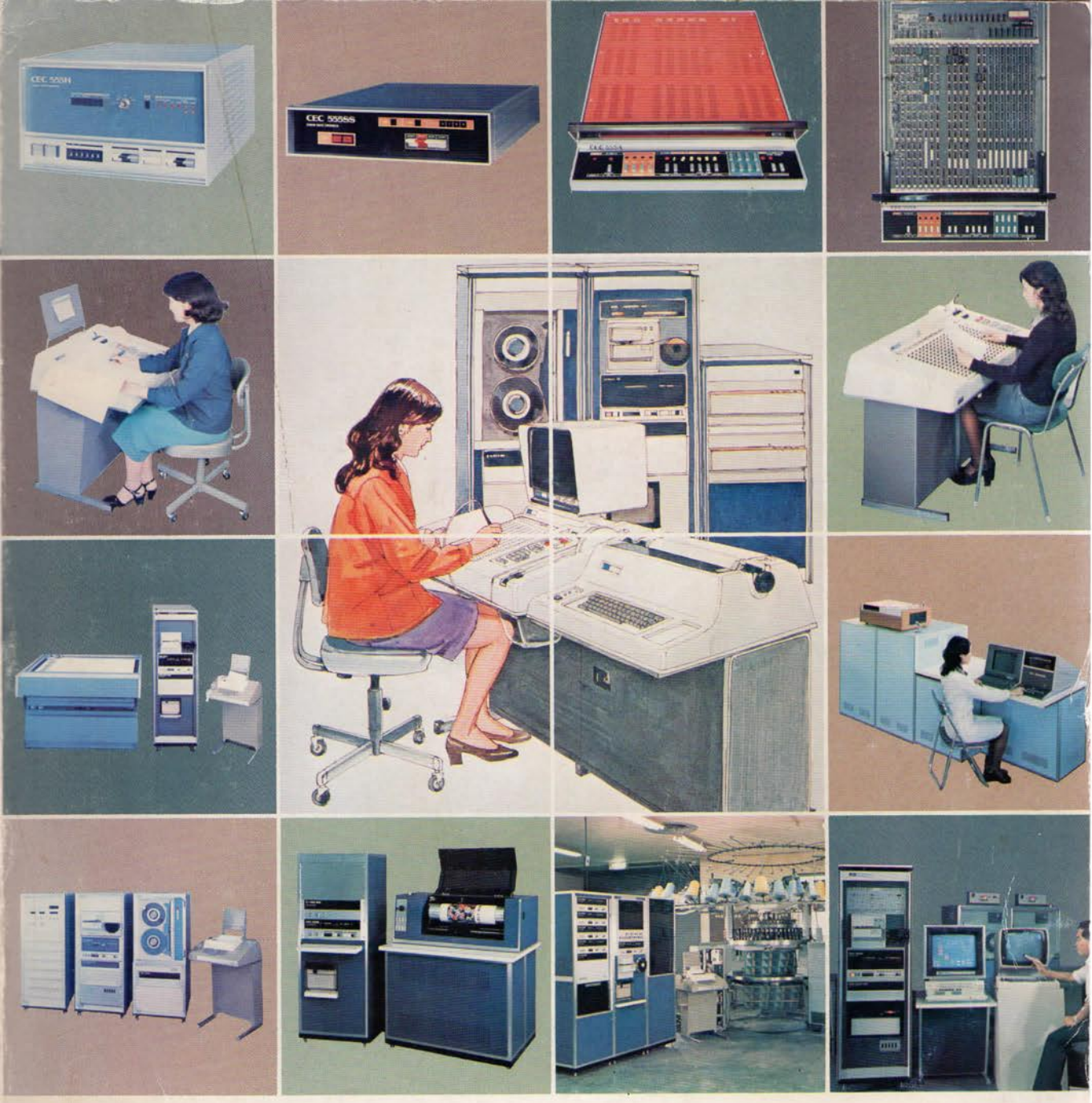


CEC 555
MINI COMPUTER HANDBOOK
ミニコンピュータ ハンドブック

ミニコンピュータ
ハンドブック

CEC 555

1975年版　中央電子株式会社

序にかえて

個別の科学技術は，一部の出遅れたものを除いて，戦後30年の間に一通り確立したようだ．これからは，個別の科学とエレクトロニクス，とりわけ電子計算機技術を結びつける総合的な方法が未来性があって，しかも，それが確立すると利用効果が大きいために重要な課題とみられる．

CEC 555ミニコンピュータシリーズとCEC 555 SSマイクロコンピュータシリーズは，いままでに広範囲にわたって確立した個別の科学と結びつけて，新しい大きな価値を創造しようとしている．

CECは，現在までCECのユーザー，指導者，ならびに各分野の諸先輩の御支援によって歩を進めることができた．

本書は，これらの方々への中間報告と併せて，それぞれの分野にあってCEC 555ミニコンピュータシリーズとCEC 555 SSマイクロコンピュータシリーズを駆使して，明日を築こうとする方々への御参考として編集した．

第1版のため御期待を充分に満足するものではないことは必定であるが，御寛容願いたい．

CECは，情報処理文明を利用して素晴らしい明日を建設しようとする方々とともに前進したいと念じている．願わくば，本書が志を同じくする方々の御発展のための一助になれば幸甚である．

1975年7月

中央電子株式会社

CEC 555

ミニコンピュータハンドブック　目　次

序にかえて……1
CEC 555シリーズの評価……5

ハードウェア編

① 555H電子計算機ハードウェア……11
② 555A超小型電子計算機ハードウェア……131
③ 555SSマイクロコンピュータ(555SS－I，555SS－II)……143
（CEC 4000 ROMプログラマ)……165
④ 555S電子計算機ハードウェア……171

周辺装置編

① CKシリーズコードキーヤ……229
② CKシリーズマルチターミナル……238
③ DRISK－2002カートリッジ式ディスク装置……242
④ CFD－300ミニディスク装置……254
⑤ CFP－500フロッピィディスク装置……256
⑥ CFD－130磁気ドラム記憶装置……261
⑦ CMT－8磁気テープ装置……264
⑧ CMT－5磁気テープ装置……267
⑨ CMT－100カセット磁気テープ……270
⑩ CS－1200入出力タイプライタ……274
⑪ CMD8，CMD4キャラクタディスプレイ……277
⑫ CS－408高速紙テープリーダ……281
⑬ CS－420高速紙テープパンチャ……283
⑭ CS－100入出力タイプライタ……285
⑮ CS－1000入出力タイプライタ……288
⑯ CLP－4100ラインプリンタ……291
⑰ CCD－5Sカラーキャラクタディスプレイ……294
⑱ CCD－5Gカラーグラフィックディスプレイ……296
⑲ CCD－5GH高速カラーグラフィックディスプレイ……298
⑳ C－1525デジタルX－Yプロッター……300
㉑ CS－60カードリーダ……302
㉒ CS－50，CS－80キーボード入力装置……303

ソフトウェア編

① ソフトウェア概要……307
② 紙テープシステム……315
§1　ローダ……315

§2　エディタ……335
§3　アセンブラ……343
§4　オクタルデバッガ……395
§5　ユーティリティ……407
§6　基本保守診断……411
§7　その他……413
③　マークIIモニタシステム〈その1〉……415
④　マークIIモニタシステム〈そのII〉……463
⑤　演算インタープリータ……465
⑥　ココル（対話計算）……473
⑦　ベーシック……509
⑧　RPG（Report Program Generator）……515
⑨　4Kフォートラン……547
⑩　マークIIフォートラン……579
⑪　コボル……627
⑫　リアルタイムモニタ……677

応用編

①　工場自動群制御システム……702
②　トータシステム……706
③　CEC EDPシステム……710
④　医療情報システム……740
⑤　繊維産業システム……763
⑥　計測制御システム……769
⑦　自動車エレクトロニクス……778
⑧　文教システム……781

マネージメント編

1．CEC 555電子計算機システムについて……785
2．CEC 医療情報システムについて……787
3．CEC EDPシステムについて……788
4．システム障害対策，および保守・サービス体制……790
5．販売代理店……794
6．リースおよびローンについて……796
7．CEC 中央電子株式会社の概要……799
8．関連会社について……818
付録A．売買契約書……820
付録B．CEC 装置保守，サービス基本契約書……825

CEC555シリーズの評価

本章は，CEC555システムについて，長年にわたりご指導いただいた方々，および直接ご使用いただいている方々から寄せられたCEC555システムについての評価とご意見を収録したものである．(順不同·敬称略)

日本トーター株式会社

昭和45年にCEC555ミニコンピュータ利用のトータシステム第1号を桐生競艇場に納入させていただき，以来，津，三国，蒲郡，浜名湖，芦屋，下関の各競艇場，足利，金沢の各競馬場にも同種システムをご採用いただいた．

競艇，競輪，競馬投票業務機械化の推進役としてCEC555ミニコンピュータが果たしてきた役割は実に大きく，当社が今日公営競技界に確固たる実績と信頼を得ている所以もここにある．すなわち，トータシステムはその業務処理の特質からシステム障害の発生，マシンダウンはもちろんのこと，システム内の些細な支障でも絶対に起こってはならない必須条件があり，この点でCEC555の信頼性は高く評価されてよいものと思う．

日本アイシーエス株式会社
取締役営業部長 **兼盛 玲二**

従来のコンピュータ利用があまりにもコストパフォーマンスに重点を置きすぎ，大型化，高速化へと方向づけされ，その結果として「要員の確保難」「オペレーションの複雑さ」「ハンドリングタイムの増加」といった問題が発生し，逆にコストパフォーマンスが下るといった矛盾が表面化して来たのであります．

中央電子㈱においては，いずれ起るこの問題点を察知し，その創生期からユニットコンピュータシステムのための技術的考察を加えられたことはその洞察力に感服いたす次第でございます．

弊社およびＩＣＳ関連企業にはすでに前記の主旨のコンピュータシステムが23セット稼動いたしておりますがその効率の優位性は同業他社の追従を許さず，コンピュータ利用の大衆化への道を開くものであると確信いたしております．

株式会社さくらカラー大阪
業務部経理課長 **井上 克磨**

当社社内の経理計算業務は今まで外注していたために，その月のデータは月末にしか得られなかったが，CECEDPシステム30を導入してからは，毎日の売上げ額が品種別ごとに，その日中につかむことができるうえ，月締切日後の請求書の発行も迅速に処理でき，そのあとの入金状況も一目で把握できている．

このシステムによって今まで得られなかった正確な販売，生産量を知ることができるので，効率の良い営業活動ができるため，今後かなりの利益が期待できるものと信じている．

全国アイシーエス協議会
専務理事 **阿部 久夫**

全国アイシーエス協議会は，全国的に組織化されたコンピュータ会計のネットワークである．現在，北は北海道札幌から南は九州鹿児島にいたるまで，23個所のセンタシステムが稼動中であり，加盟いただいている会計事務所は900を上回る成長をみている．いずれもCEC CKシリーズコードキーヤという誰でも操作できる便利な入力端末装置を使用し，CEC555ミニコンピュータシステムを設置したセンターとの密接な連係プレイで，会計業務の大幅な省力化を推進している．

今後業務処理の多様化に伴い，ミニコンならではの小回りのきくサービスを実現していくために，CEC555システムに期待すること大である．

株式会社岩崎

専務取締役 **古口　等士**

測量計算の場合に要求されるコンピュータの機能としては，多種のプログラムが用意されていること，プロッタや座標読取装置などの周辺装置が随意に接続できることなどが主としてあげられるが，何より大切なことはマシンの信頼性と故障知らずのタフネスさにつきると考える．この点CEC 555電子計算機はこれらの要求にピッタリ応えてくれており，これが発売以来100システムにも及ぶ販売実績を得，ユーザ各位に喜んでいただいている大きなポイントであろうと考えている．

東京大学医学部教授

医学博士 **大島　正光**

このたび中央電子株式会社よりCEC健康管理システムが研究開発された．これはコストが安く，狭い空間に展開することができること，省力化がなせられていること，集団検診向きであることなどの特徴をもったものである．わが国においても人口の老齢化にともない成人病予防に大きな関心がはらわれているが，このシステムは健康管理システムの従来のものに対して意欲的に世の中のニーズに応えたということができるものであろう．したがってこれを外からながめると普及型といってもよいものである．その点からも「治療よりも予防へ」という最近の医療の方向づけを実現させるためにも，このような普及型ができたことは新しい意義があると考えられ，世界中に推せんするとともに，今後このような方向にさらに前進して健康維持増進の実のあがることを期待する次第である．

石塚病院

院長 **石塚　民幸**

当病院は，かねてより，病院施設の近代化を計画しており，X線テレビ，血液自動分析装置など，各種医療機器の高度化導入を行い，患者の便宜を計っていたが，昭和48年10月中央電子株式会社(CEC555)のコンピュータシステムを採用することにした．コンパクトながら，その機能は大型汎用機に相当し，CK-5コードキーヤの特殊性を生かして，TTY，高速ラインプリンタ等がオンラインで接続されたシステムで，窓口会計業務，レセプト作成，未収金処理，各種病院管理資料の作成等の広範にわたる事務処理を短時間のうちに行い，将来は事務処理だけに終らず，EKG，EEGなどの自動診断および，地区センターとして，他病医院のレセプト作成にも役立ちたいと考えている．ミニコンシステムとはいいながら，極めて良い成果をあげ，患者からも窓口での待時間が非常に短くなったと感謝されている．本システムの運用にしても，従来のように，要員に対する特別教育を要せず，当病院の事務職全員が，パネルシート企画作成等にも直接参加し，短期間に，そのオペレート等も男女を問わずマスターしている．このシステムの特性である「正確にしてオペレーションの安易」を最大に発揮して，便利に運営しており，導入前のコンピュータアレルギーは全然ない．

これからは，このようなシステムこそ，わずらわしい事務処理の近代化，省力化に役立つシステムとして推薦したいともと思います．

萱場工業株式会社

東京工場制御機器部門

部長 **中島　正也**

清水　邦宏

繊維情報システムのミニコンピュータとしてCEC555シリーズをすでに3年間にわたって全面的に利用してきた．555シリーズは筐体，電源部，パッケージ，周辺装置にいたるまで同社の一貫生産であり，システムビルドする場合に大変便利である．今後とも協力を深くしてこの成果をより大きなものに発展させていきたい．

日産自動車株式会社
宇宙航空部実験課長 **和地　英麿**

実験所内でのデータ処理に電子計算機を導入する場合，データの流れをとどこおらすことなく，スムーズに処理できることが人と機械の結合で充分検討されなければならない問題である．そのひとつの目的として何んといっても故障が絶対に起らないことが第一の条件となると思う．私共とCEC555ミニコンピュータとの出会いは4年前ある種のアナログデータ処理に555M型を導入したのに始まるが，現在まで故障知らずで稼動をつづけており，組織内でのオンライン処理には価格，性能はもとより「信頼性」の点で最適のマシンであると思う．

大阪大学基礎工学部
助教授 **豊田　順一**

CEC製ミニコンピュータを使用して約3年を経過したが，ちょっとした故障はともかく大きなトラブルもなく現在に至っている．

これらの計算機はコンパイラの開発，パターン認識等の研究に用いられている．特にCEC555Hは，過去の経験がかなり生かされているようであり，若干の機構部制御バスの引き出し方に不満があるが，まず平均的なミニコンピュータといえよう．保守サービス，周辺機器の接続について積極的に対処してくれているようである．

立命館大学工学部電気工学科
教授・工学博士 **苅屋　公明**

CEC-1051の教育用シュミレータは，発売当時の7年ほども以前より，学生の基礎実験として使用しています．昨年，さらにこれのグレードアップとして556Aのミニコンピュータを導入し実習しています．1051は，主にハードウェア面での基礎の実験であり，556Aはアッセンブラによるロジックの流れや，コンピュータの考え方を会得し，さらにコンパイラ言語による科学技術計算の学習も，自主的に学生にやらせています．

とくに，556Aはハードの学習としてロジックの流れ，タイミング等が理解でき，ソフトウェアとしては豊富なソフトウェアがあり，学生はベーシックな対話的コンパイラから，高級な計算を行えるソフトウェアまで適宜に選んで使用しています．

また，卒業研究の学生や，大学院の学生が研究，実験データの整理や計算にたいへん便利に使用しており，研究速度の向上に役立てており，高学年の学生にとって不可欠な教材であると思います．

神戸大学第二外科教室
講師 **津嶋　昭平**

CEC－555電子計算機を導入し，約半年になるが，その間，心音，心電，心臓カテーテルおよび，インピーダンスカルディオグラフ等の解析に利用している．

使用期間が短期間のため，概略的なことしか述べることができないが，検査機器（心電計，心音計，インピーダンスカルディオグラフ等）を，ON-LINEにて接続し，データの収集と解析に使用している．

CEC－555電子計算機は，特にデータの解析と整理に，その威力を発揮し，使いやすく信頼性の面でも，別に問題はないように思われる．

現在5名のグループにて運営しているが，今まではデータの解析に，多くの労力と時間が費やされていた．特に，ＩＣＵ等でデータを正確に，かつ解析をいそぐ場合に，現在では計算機の導入により，処理の正確，かつ迅速化，省力化がなされ非常に喜んでいる．

都馬県成人病研究所
所長 **渡辺　孝**

本邦における死因の第1位を占める脳血管疾患や第3位を占める心臓病は予防可能という点で理論的には見通しが明かるい．つまり何らの自覚症もない時期に積極的に集団検診を行ない，発見された芽を未然につんでしまえば発病からまぬがれることができる．

脳血管疾患についでは心電図所見に異常を示す場合に4倍も本症発生の危険があり，心筋梗塞や狭心症の基盤である心筋虚血は心電図だけが知っている病態である．

ここに心電図検査の重要性があるが，循環器集団検診の隘路は莫大な数の心電図を限られた時間内に正しく判読する専門医が容易に得られない点にある．

CEC552心電図自動診断装置，CEC552D心電・心音自動診断システムは，この点を解決した画期的な開発であり，集団検診の実施を容易にしたと同時に，人間ドックやメジカルセンターにおけるスクリーニングにも有用であるものと期待される．

ハードウェア編

ハードウェア編目次

① 555H電子計算機ハードウエア…… 11
② 555A超小型電子計算機ハードウエア…… 131
③ 555SS マイクロコンピュータ(555SS-I,555SS-II)…… 143
(CEC4000ROMプログラマ) …… 165
④ 555S電子計算機ハードウエア…… 171

❶ 555H電子計算機ハードウェア

1．概要

CEC555H電子計算機は，小型，汎用の電子計算機で，情報処理システムの中央処理装置または各種の制御システムのコントロールエレメントとして設計されている．

特に仕様は5年間にわたる実績経験をもとに，44年から製作販売されてきたCEC555シリーズの集大成をめざしている．したがって従来の命令やソフトウェアの変更をすることなく互換性を保ち，かつ最新のMSI，LSIを採用して高速化をはかったことおよび情報処理応用システムへの実際利用に際して便利で信頼性の高い入出力接続制御系に独特の設計をしていることによって，今後5年～8年の長期にわたりユーザの利益を保証している．

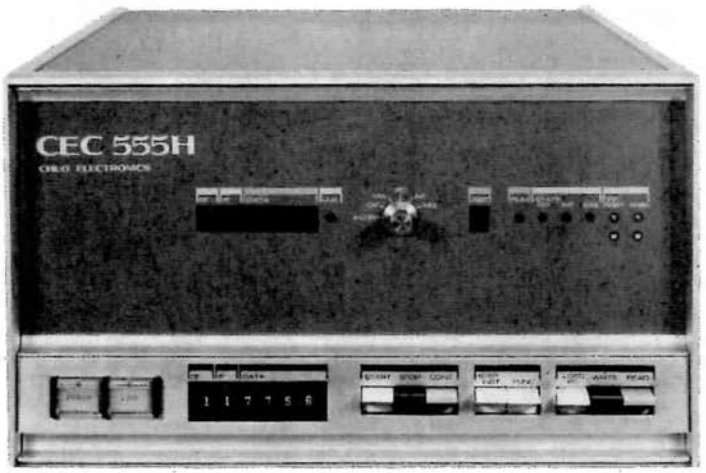
555H電子計算機は，1語12ビット，1フィールド4096語のメモリをもち，4K1フィールドごと32K語まで拡張することができる．

演算処理装置（PU），記憶装置（MEMORY），チャネルマルチプレクサ機構から構成され，従来用いられているPUを中心とした電子計算機組織と異り，記憶装置が中心となり，PUは他の周辺装置（例えばドラム，ディスク，磁気テープなどの各装置）と同等に扱われているところに特色がある．

メモリとの情報のやりとりはすべてチャネルマルチプレクサ機構（6ch内蔵）を通してDSA方式で行われ，マルチプロセッシングシステムを構成することができる．また割込みレベルを2レベル設けオンライン処理を容易にしている．

2．仕様

方　　式：プログラム記憶，並列2進，単アドレス
素　　子：磁気コア
語　　長：12ビット
　　PU　　12ビット
　　メモリ　14ビット { 1プロテクションビット（オプション）／1パリティビット }
サイクルタイム：0.8μs
記 憶 容 量：4～32K語
演 算 方 式：並列2進，2の補数表示
演 算 速 度：加算1.8μs
アドレス方式：1アドレス方式，直接，間接，オート，インデックス
割込みレベル：2レベル
入出力制御方式：(イ)プログラムコントロールモード
　　IOバスライン方式
　　(ロ)ダイレクトストレージアクセス方式（DSA）
　　DSAチャネルを6チャネル内蔵し，各チャネルには優先順位がつけられている．この内1chはPUに他はオプションにより他のPUまたは各周辺機器に割当てられる．
外 形 寸 法：W430×D600×H249（20K内蔵），199（12K内蔵），これ以上は外部メモリユニ



単　体



ラック実装

ットに装備

※　また他のCEC555シリーズ，C555，555，S555T，555Mとプログラム，インタフェースがコンパチブルなので555シリーズ用の周辺装置と容易に接続することができる．

ソフトウェア：アセンブラ，フォートラン（4,8K），ココル，コボルなど555シリーズとコンパチブル．

3．構成

CEC555H電子計算機は，図のように，処理装置，チャネルマルチプレクサ機構，記憶装置から構成されている．各装置はすべて独立しており，すべての算術演算，論理演算，システムの制御指令は処理装置で行われる．処理装置と周辺装置のインタフェースはバスライン方式で行われ，プログラムによる個々のデータの転送はアキュームレータ（AC）経由で行われる

① 処理装置：仕様は2項参照

プロセッサユニットは4台の12ビットレジスタ（MSIスクラッチパッドメモリ）（PC，AC，MQ，MA）とリンク（フリップフロップ），アリスメティックロジック，命令レジスタおよび制御回路，入出力機構から成立っている．

② 記憶装置：4KW～32KW（4KWを1FIELDとし0～7FIELDまで増設可）

③ チャネルマルチプレクサ機構

チャネルマルチプレクサ機構は，周辺装置および処理装置が記憶装置との情報の交換を速かに，効率よく行うための機構である．

記憶装置とのインタフェースは，ダイレクトストレージアクセス方式（DSA）で行われる．

記憶装置は，記憶素子として17ミルの磁気コアを用い，0.8μsのサイクルタイムでランダムアクセスすることができる．

またメモリサイクルとプロセッササイクルは非同期方式で接続されているので，コアメモリの代りにRAM，ROM（REPROGRAMMABLE）などのLSI MEMORYを接続することもできる(オプション)．

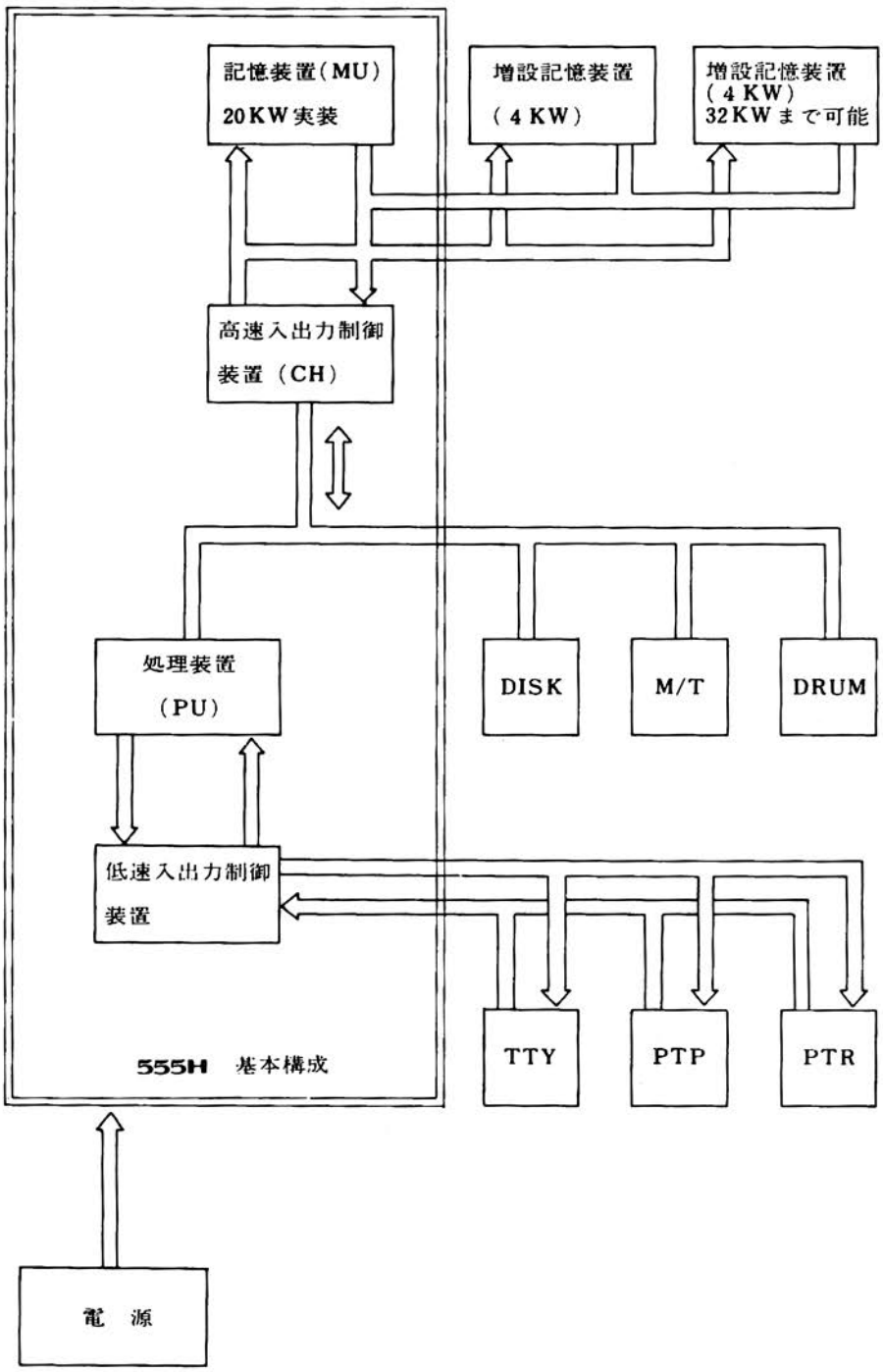
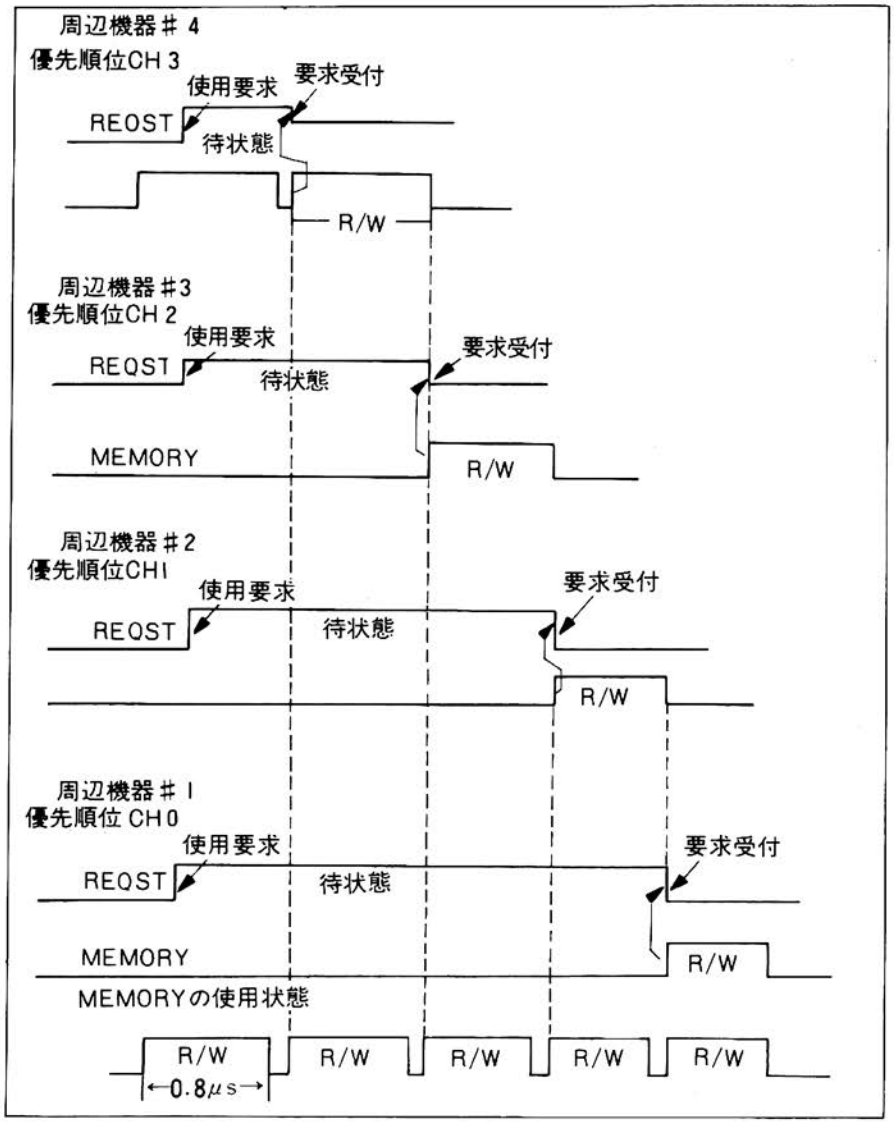


図 3 － 1　555 H　基本構成図



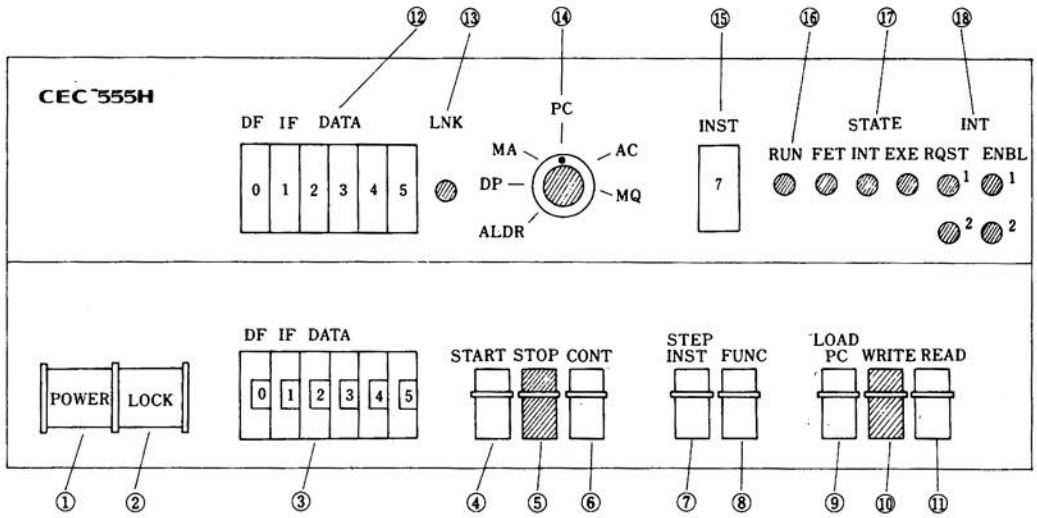
3-2図　周辺装置，記憶装置の情報交換図

4. 基本構成，性能

構　　成	性　　能
記憶装置（ME）	
素　　子	磁気コア（17ミル）
語　　長	12ビット＋1パリティ＋1ビット　計14ビット
サイクルタイム	800nS
容　　量	4 KW～32KW
処理装置(PU)	
演算方式	並列2進法　2の補数表示　固定小数点
命令数	基本命令8種類
演算速度	加減算2 μs　アドレス修飾3μs
アドレス方式	直接，間接
インデックスレジスタ	8個（記憶装置上）
割　　込	2レベル
入出力制御	
低速入出力制御	2種類（Hモード，Lモード）
（I/O）	接続台数　64台
	転送速度　Hモード　700KW/S
	Lモード　250KW/S
高速入出力制御	3種類（1サイクル，2サイクル，3サイクル）
（CH）	接続台数　6台
	転送速度　1サイクル　1000KW/S　2サイクル 500KW/S
	3サイクル　300KW/S
設置条件	
周囲温度	0～40℃
湿　　度	23～85%
電　　源	100V±10V

5．各部の説明

5—1コンソールパネル



番号	名前	内容
①	POWER SWITCH	電源オン／オフスイッチ．オールタネイト型押ボタンスイッチで一度押してオン（ランプ点燈）もう一度押してオフ（ランプ消燈）となる．
②	LOCK SWITCH	オールタネイト型押ボタンスイッチで一度押してオン（ランプ点燈）もう一度押してオフ（ランプ消燈）となる．オンの時コンソールスイッチの機能は電源スイッチおよびＰＰＣ機能（INST STEPキーとFUNC STEPキーを同時に押す）を除いてすべてロックされる． ただし③DATA SWITCHは禁止されない．
③	DATA SWITCH	0～7の8進ディジスイッチでDF,IFはLOAD PCキーと共にデータフィールド，インストラクションフィールドの設定に用いられる． DATAはLOAD PCキーと共にプログラムカウンタの設定としてまたWRITEキーと共にコアメモリへのライトデータとして用いられる． OSR命令によりDATA SWITCHをアキュムレータに読み込むことができる．
④	START KEY	モーメンタリ型のピアノスイッチでSTARTキーを押すとRUNフリップフロップがセットされ続いてKR信号が発生し状態フリップフロップ、アキュムレータ，リンク等をクリヤし計算機はプログラムカウンタで示されるPC番地からプログラムの実行を開始する．
⑤	STOP KEY	モーメンタリ型のピアノスイッチでSTOPキーを押すとその時の実行サイクル（EXECUTEサイクル）の終了時にRUNフリップフロップをクリヤする．
⑥	CONT KEY	モーメンタリ型のピアノスイッチでCONTキーを押すとRUN

フリップフロップがセットされその時のＰＣ番地からプログラムの実行を開始する．
アキュムレータ，リンク，状態フリップフロップなどはクリヤされない（ＫＲパルスは発生しない）．

⑦ INST STEP KEY
オールタネイト型のピアノスイッチで一度押してオン，もう一度押してオフとなる．
INST STEPキーオンのとき計算機はEXECUTEサイクルの終了時にSTOPする．CONTキー押してFETCH→EXECUTEと一命令ずつ実行させることができる．

⑧ FUNC STEP KEY
オールタネイト型のピアノスイッチで一度押してもう一度押してオフとなる．

⑨ LOAD PC KEY
FUNC STEPキーオンの時，計算機はFETCHサイクルまたはEXECUTEサイクルを実行しSTOPする．CONTキーによりFETCH EXECUTEの各サイクルを一つずつ実行させることができる．

⑨ LOAD PC KEY
モーメンタリ型のピアノスイッチで，DATAスイッチで設定されたデータをLOAD PCキーを押してプログラムカウンタに書き込む．

⑩ WRITE KEY
モーメンタリ型のピアノスイッチでDATAスイッチで設定されたデータをWRITEキーを押してPCで示されるコア番地に書き込む．書き込みが終了するとPCは＋１される．

⑪ READ KEY
モーメンタリ型のピアノスイッチでREADキーを押すとPCで示されるコア番地の内容が読み出され，ＤＰに書き込まれる．⑭SELECTスイッチをDPに選んでおけば⑫データ部にDPの内容が表示される．
読み出しが終了するとＰＣは＋１される．

⑫ DISPLAY
DFはデータフィールド，IFはインストラクションフィールドを表示する．DATAは⑭SELECTスイッチで選ばれたDP，MA，PC，AC，MQのいずれかを表示する．

⑬ LNK
リンクを表示する．

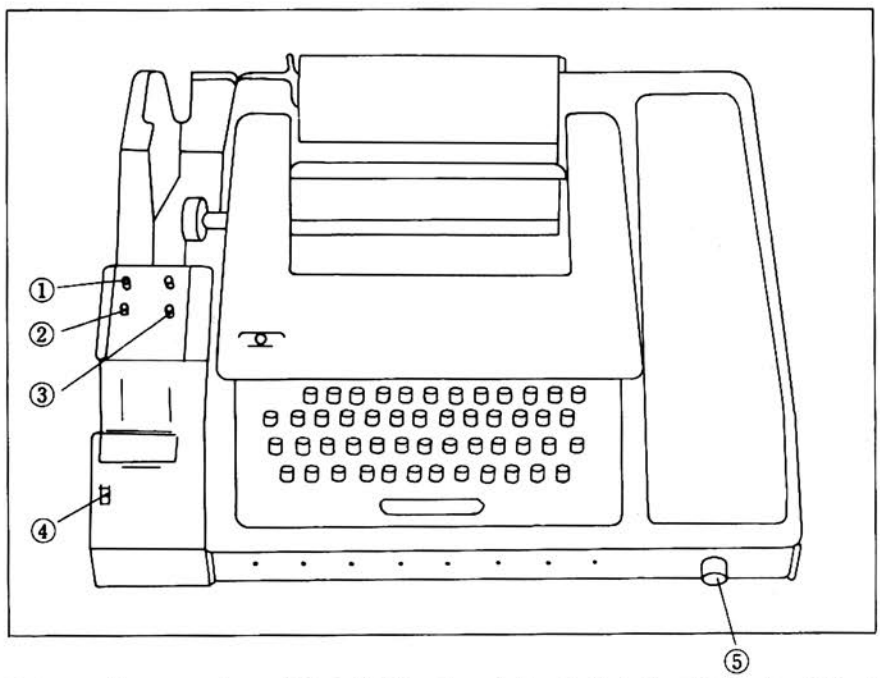
⑭ SELECT SWITCH
DATA表示の内容をセレクトする．
DP：DPレジスタが選ばれる．
MA：PCBレジスタが選ばれる．
PC：PCレジスタが選ばれる．
AC：ACレジスタが選ばれる．
MQ：MQレジスタが選ばれる．
＊ALDRはAUTO LOADERの略で，SELECT SWITCHをALDRに選んでSTARTキーを押すとRIM LOADERがコアに自動的に書き込まれる．
（AUTO LOADER機能はOPTION）

⑮ INST
INSTRUCTIONレジスタを表示する．
0：AND，1：TAD，2：ISZ，3：DCA，4：JMS，5：JMP，6：IOT，7：OPR

⑯ RUN	RUNフリップフロップを表示する.
⑰ STATE	FET：FETCHサイクルを表示する. INT ： INTERRUPTフリップフロップを表示する. EXE：EXECUTEサイクルを表示する.
⑱ INT	INTERRUPT REQUESTおよびINTERRUPT ENABLEを表示する.

	INT RQST	ENBL
INTRQST の表示→❶		❶←FION の表示
INTRQST 1 の表示→❷		❷←FION 1 の表示

5－2 入出力タイプライタ制御パネル（CS—100以下TTY）



このほかにシステムコンソールとしてCMD—8，4キャラクタディスプレイ，CS—1200入出力タイプライタが用意されている.

番号 名前	内容
① REL. プッシュボタン	テープを取りはずしたり取りつけたりするためにパンチからテープを解放する.
② B. SP. プッシュボタン	今パンチしたばかりのキャラクタを手動で修正または抹消するためにパンチ上のテープを1スペースだけバックスペースする.
③ OFF and ON プッシュボタン	テレタイプキーボードプリンタを連動して，テープパンチを動作させる.
④ START/STOP /FREE	FREEの位置にセットするとリーダは切り離され，テープをロードしたりとりはずしたりできる．中間のSTOPの位置にセットするとリーダ機構は動作しないがテープを束縛する.

STARTの位置にセットするとリーダはテープを束縛しプログラムコントロール下で動作する.

キーボード　パンチONプッシュボタンを押したとき，タイプライタおよびパンチングテープとして使用し用紙上に印刷する．そしてLINE/OFF/LOCALスイッチをLINEの位置にすればコンピュータに入力データを供給する用法となる.

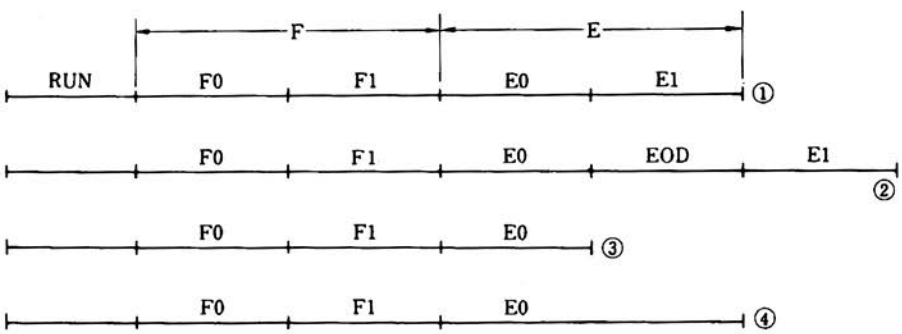
⑤ LINE/OFF/LOCAL　テレタイプ基本電源を供給し，また中央処理機構とのデータの接続を制御する．LINEの位置にセットするとテレタイプはコンピュータのI/O機器として接続される．OFFの位置にセットするとテレタイプの電源が切れる．LOCALの位置にセットするとテレタイプは，オフライン機器として動作し，中央処理機構に対する信号接続は断たれる.

5—3　PU部

PU部内の各ブロックのタイミング信号はすべてROM（READ ONLY MEMORY）で作られている．ADDRESSを選択するだけで各ブロックのタイミングを指定することができる．仕事の割当時間はRUN，FETCH，EXECUTの３つのサイクルで定まる．RUNはMANUAL操作のとき，この時間帯ですべて処理する.

FETCHはF０，F１のサブサイクルに分れる．F０はADDRESS DATAをメモリに転送．F１はメモリからREAD DATAをIRにセットするまでの時間.

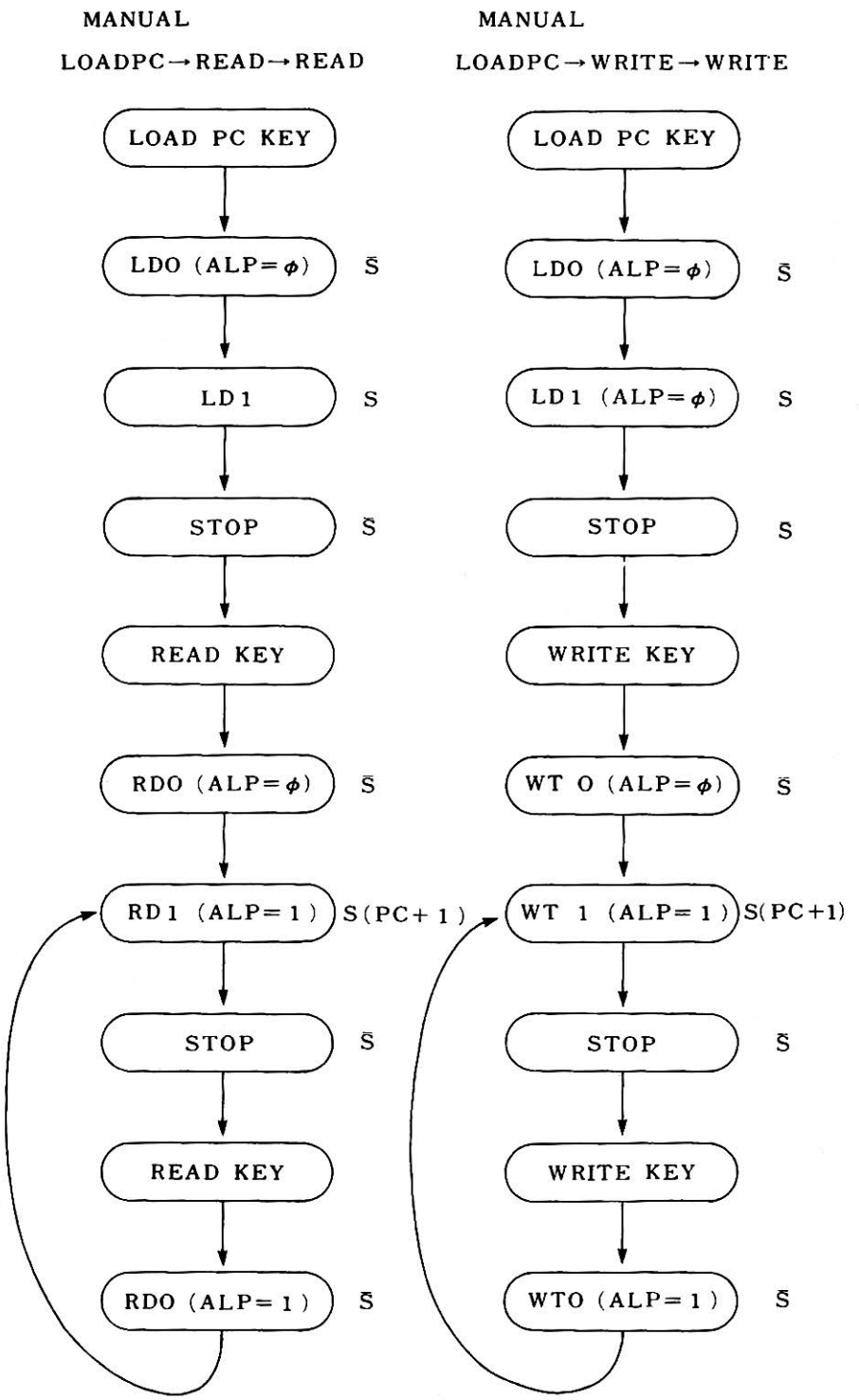
EXECUTはE０，EOD，E１サイクルに分れる．各命令の実行時間に使用される.



仕事の内容によって上記の４種類に分けられる.

① MRI（メモリーリファレンス・インストラクション）命令で直接アドレスの場合
② MRI命令で間接アドレスの場合
③ OP１，OP２，IOT命令の場合
④ OP３，JAP，IOT命令の場合

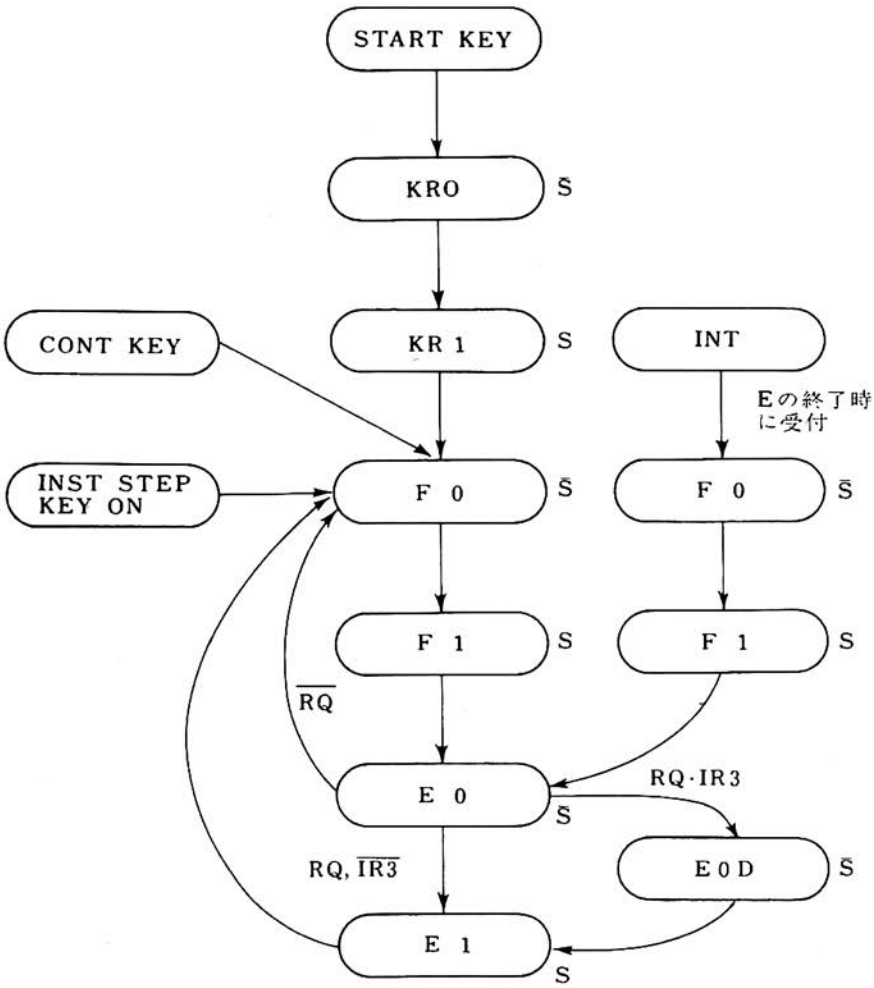
STATUS			ROMA	ROMB					備考
				0	1	2	INT	S	
RUN	LOADPC KEY		0	0	0	0		0	
								1	(SR)→PC(SR)→(SM)→(AL)→(SH)→(DP)→(FP)
	STOP KEY		0	0	0	1		0	前面パネルSWによって各レジスタの内容をチェックのためPRO=1 PRO=1 PRG=0にしておく。
								1	
	WRITE KEY		0	0	1	0		0	(PC)→PC(LOAD PC→READ) RQ=1 (PC+1)→PC(READ→READ)
								1	(SR)→(SM)→(AL)→(DB)→(CH) (DP)←(SH)
	READ KEY		0	0	1	1		0	(PC)→PC(LOAD PC→WRITE) RQ=1 (PC+1)→PC(READ→READ)
								1	(CH)→(SM)→(AL)→(SM)→(DP)
				1	0	0			未使用
RUN	START KEY	KRO	0	1	0	1	0	0	0→AC
		KRI	0	1	0	1	0	1	0→MQ
				1	0	0			未使用
	F	FO	0	1	1	1	0	0	(PC)→PCB(SKP) RQ=1 (PC)+1→PCB(SKP)→(FP)→(AL)→(DB)→(CH) (PC)がアドレスとなる(IIP)←(SH)
		F1	0	1	1	1	0	1	(PCB)+1→PC ◯→IR
	F (INT)	FO	0	1	1	1	1	0	(PC)→PCB(SKPは無視) RQ=0
		F1	0	1	1	1	1	1	(PCB)+1→PC JMSO →IR
	E	EO	1	1	1	1		0	
		EOD	1	1	1	1		0	
		E1	1	1	1	1		1	



ALP＝0の場合　PC＋1を禁止

ALP＝1の場合　PC＋1用信号

RUN

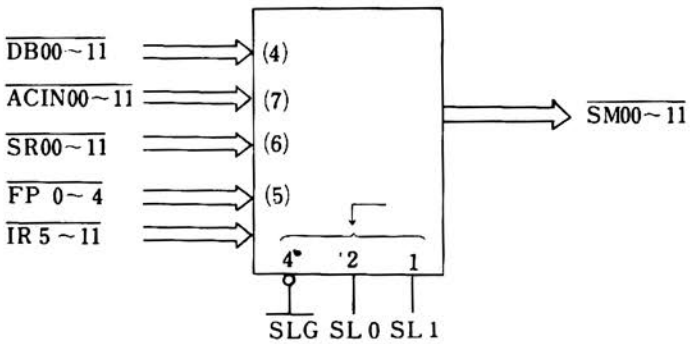

START KEY
KRO
S̄
CONT KEY
KR 1
S
INT
Eの終了時
に受付
INST STEP
KEY ON
F 0
S̄
F 0
S̄
F 1
S
F 1
S
RQ̄
RQ·IR3
E 0
S̄
RQ, IR3̄
E 0 D
S̄
E 1
S

PUブロック図の説明

SM：SUMの略である.

このブロックはDATAをSELECTする.

DM8214（NS）6個より構成されている.

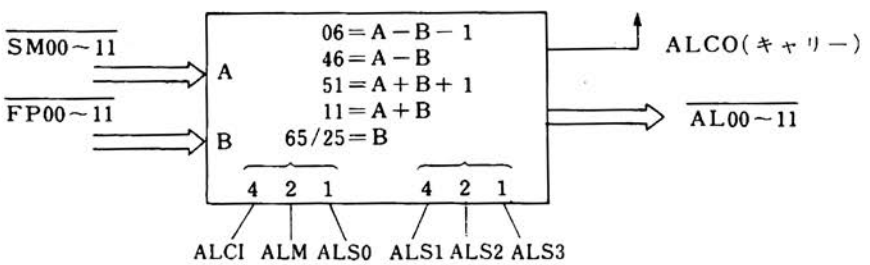


		SLG(4)	SL 0(2)	SL 1(1)
DB00～11	4	0	0	0
FP 0～4,IR 5～11	5	0	0	1
SR00～11	6	0	1	0
ACIN00～11	7	0	1	1

SLG＝1のときはDATA情報はSELECTは出来ない（受付けない）希望するDATAをSELECTした場合はSLGを0にしてSL 0, SL 1をデコードしてやればよい.

AL：ARITHMETIC LOGICの略

このブロックは加減算の演算を行う. SN74181（T.I）3個より構成

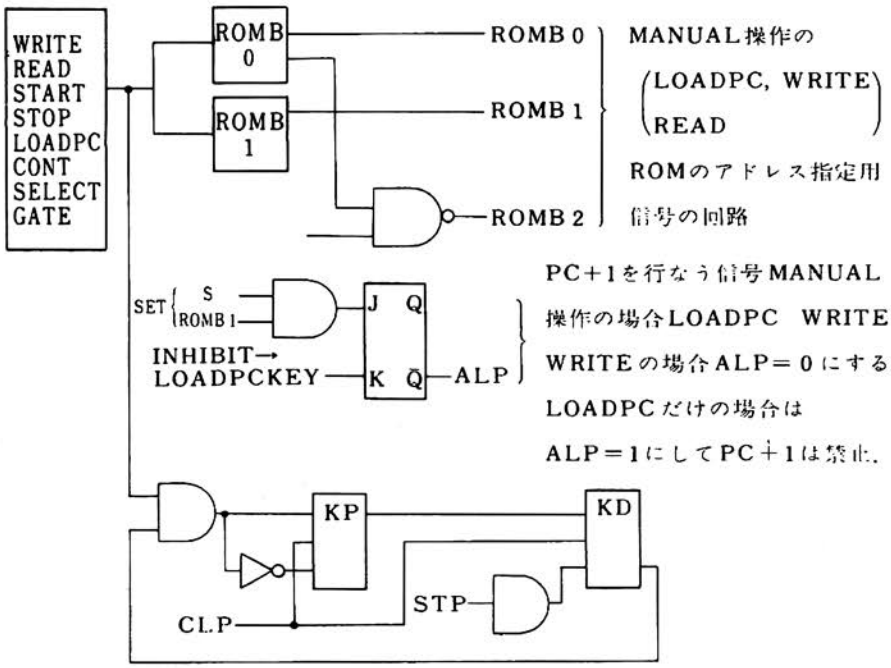


Aの情報とBの情報を加算したい場合下記のようにすればよい.

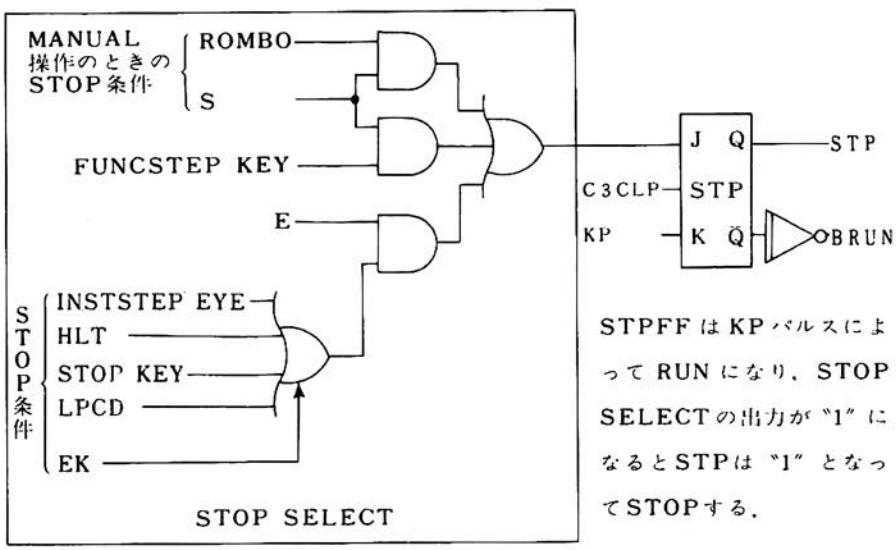
$$\left.\begin{array}{l}\text{ALCI}=0\\ \text{ALM}=0\\ \text{ALSO}=1\end{array}\right\}1 \qquad \left.\begin{array}{l}\text{ALS1}\\ \text{ALS0}\\ \text{ALS1}\end{array}\right\}1$$

ALCOはキャリーでLNKへ転送する.

START/STOP：このブロックは操作パネルのSTART, STOP, READ, WRITE, LOADPC, CONT, INST STEP, FUNC STEP等のKEYを押したときの動作を定める. PUがSTOP状態からRUN状態に入るまでの準備動作を行う. このブロックは, 基本的には6個のREGISTERで構成されている.



KPFF…このFFは微分パルスを作る．操作パネルのKEY—を押したとき1発だけ発生．KDFFはKPパルスを1発だけ発生させるためのFFでSTP＝1のとき解除．



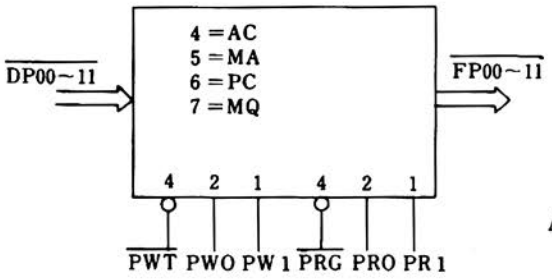
EK：Eをクリヤすると同時にSELECTGATEをあける．

FP：REGISTER FILEの略でPは意味はない．

SN74170 3個より構成

このブロックはAC，PC，MA，MQの各REGISTERがFILEされている．

希望するREGISTERからDATAのREAD（WRITE）を実行するには$\overline{\text{PRG}}$（$\overline{\text{PWG}}$）をϕにしてPRO，PR1（PWO，PW1）をデコードしてやればよい．



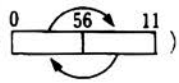
PC から DATA を読出したい場合は

$\overline{PRG} = 0$
PRO ＝ 1 } 6
PR1 ＝ 0

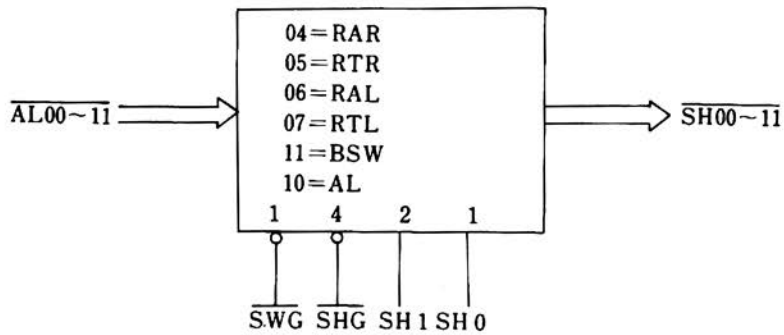
AC に DATA を書込みたい場合

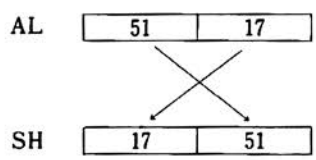
$\overline{PWT} = 0$
PWO ＝ 0 } 4
PW1 ＝ 0

S H：SHIFTの略

DM8214 6 個，DM8213 3 個より構成このブロックはALのDATAを（ 

）左右にシフトさせたりバイトスワップさせたりする.





BSWを実行する場合　RARを実行する場合

$\overline{SWG} = 0$ } 1　　$\overline{SWG} = 1$ } 0
$\overline{SHG} = 1$　　$\overline{SHG} = 0$
SH1 ＝ 0　　SH1 ＝ 0
SH0 ＝ 1 } 1　　SH0 ＝ 0 } 4

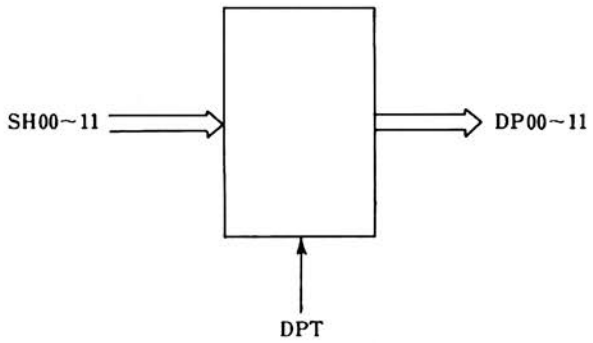
バイトスワップ（BSW）

DP:DISPLAYの略

SN74175　3 個より構成

このブロックは操作パネルの表示用に用いられる.

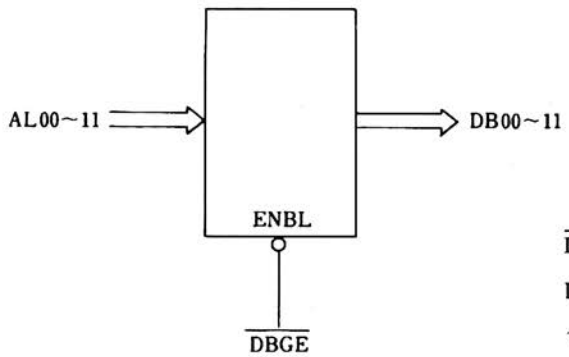
AC，PC，MA，MQの各REGISTERの内容およびメモリの内容はこのDPを経由して操作パネルに表示される．このブロックはREGISTERで構成されている



DB:DATA BITの略

DM8831（NS）3個で構成

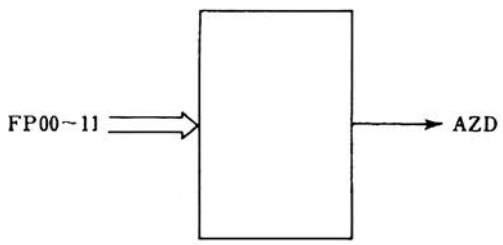
ADRESS情報とDATA情報はこのブロックを経由してメモリへ転送される.



$\overline{DBGE}$＝0のときメモリへDATA転送それ以外は1にしておく.

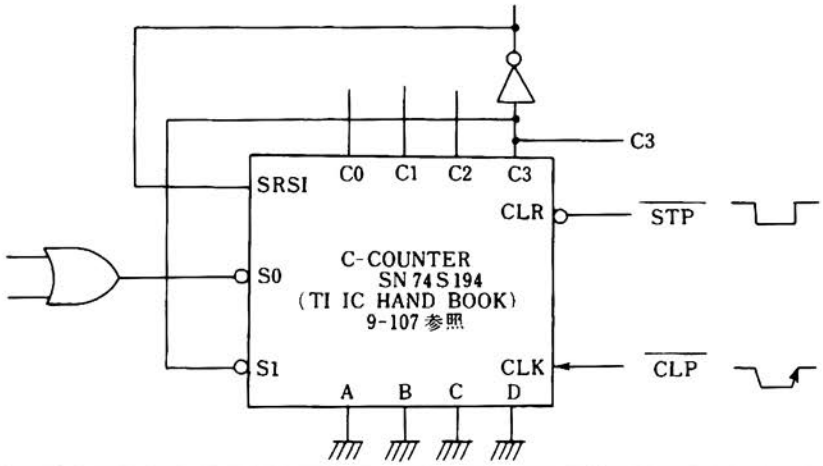
AZD：ACUMULATOR ZERO DETECTの略

AC＝0を検出する回路である.



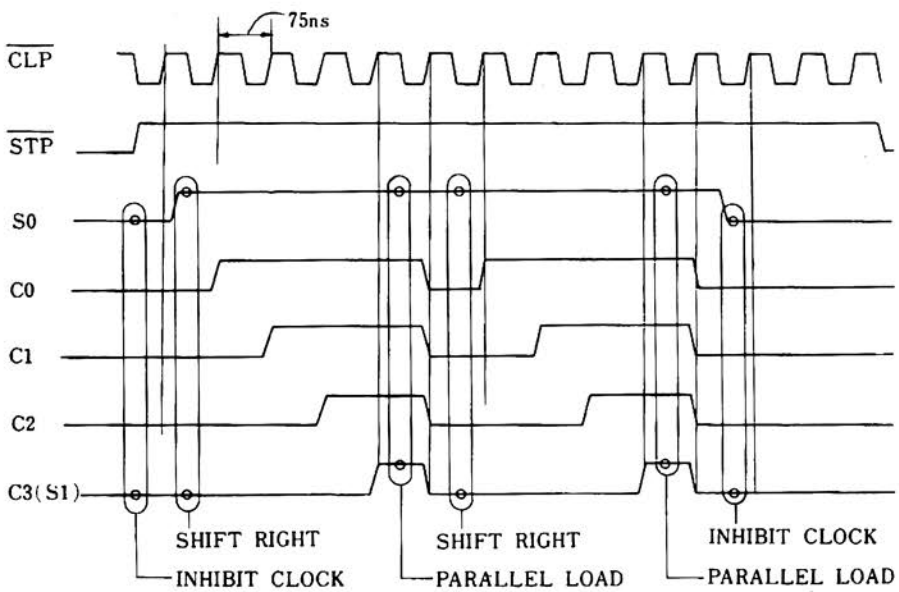
ACの内容が0である場合はAZD＝1となる

C-COUNTER…この回路はPU内の動作を定めるタイミング信号の発生およびメモリとのインタフェイス信号を作っている.



	S 1	S 0	動　　作
INHIBIT CLOCK	0	0	C 0～C 3のタイミングは発生しない
SHIFT RIGHT SERIAL IN	0	1	
SHIFT LEFT SERIAL IN	1	0	555 H では使用せず
PARALLEL LOAD(A-D)	1	1	(A－D) の入力信号がC 0 ～C 3 に OUT PUTされる

C-COUNTER部はメモリとのインタフェイスと密接にからんでいるのでインタフェイスの理解が大切である．特に，S 1，S 0のコントロールで，S 1 ＝ 1，S 0 ＝ 0の状態には絶対にしてはならない．C-COUNTERの動作についてC-COUNTERのSTARTは，S 0 ＝ 1，S 1 ＝ 0の状態から，STARTする．



C-COUNTERは，S0を0にするか，1にするかによって動作を定めている．

1） C-COUNTERのWAIT条件（S0＝0）

$$W(S0=0)=\underline{RQ\cdot C_2+ES_n(BDAVL+BDRCV+BAKN)}\ C_2$$
$$=C_2(RQ+ES_n+DIR)\cdot(DRCV\cdot BAKN)+DAVL\cdot C_1RQ\cdot ES_n$$
$$+DAVL\cdot C_1RQ\cdot ES_nDIR-①$$

2） WAITの解除条件（S0＝1）

$$W(S0=1)=\underline{RQ\cdot ES_n}+ES_n\{(BAKN+DRCV)(DIR\cdot RQ)$$
$$\underline{+RQ\cdot DIR\cdot DAVL}\}-②$$

上記の①と②式の組合せによって，COUNTERは，START又は，STOPする．

タイムチャート回路図を参照して①と②式の組合せについて説明．

① C-COUNTERが，STARTする場合②式の条件が必要である．

イ） $RQ\cdot ES_n$ …… RQSTの発生もなく，ES_nの信号もないときは，C-COUNTERは，無条件にSTARTする．(OP1,2,3 IOT 命令実行の場合)

ロ） $C2\cdot RQ(DRCV\cdot BAKN)$ ① …RQSTが発生すると，C2の点で，C-COUNTERはDRCVまたは，BAKNの信号がくるまで待っている．

ハ） $ES_n\cdot BAKN\cdot RQ$ ② …… BAKNの信号がくると，C3を発生し，PUは次の仕事($\overline{S}\rightarrow S$)に進み，C1の点またはC2の点で待つ．(C1，C2の見分は，DIRの方向によって定まる．DRCVの場合は，C2の点，DAVLの場合は，C1の点でSTOPする．)

ヘ） $DAVL\cdot C1RQ\cdot ES_nDIR$ ① … ㋑の項でBAKNが返ってきて，S0は1となって，C-COUNTERは進みPUはCHへDIRの転送指示によってメモリからDATAがくるのを待っている．C1の点で待っている条件が左記の項である．

ト） $DAVL\cdot ES_nDIR\cdot RQ$ ② ……メモリからDAVLのタイミング信号と一緒にDATAが送られてくると，C-COUNTERは，S0＝1となって進む．

DIRの条件

RUNの時（ROMB＝1の時）		TRANSF DIRの条件	
JMS or DCA＝1	→DIR＝1	(DRCV,AKN)F0，F1	→READ指定
RQ＝1	→DIR＝1	TAD, AND (E0, E)	→READ指定
STOPの時（ROMA＝1の時）		DCA	→WRITE
ROMB2＝0	→DIR＝1	ISZ	→READ

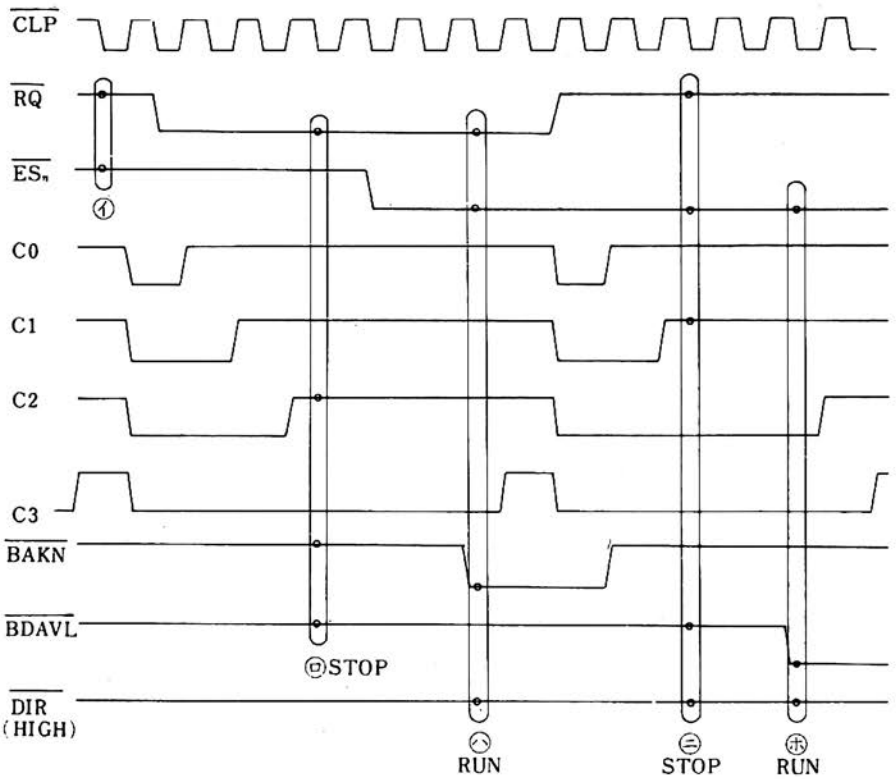
DIRとTRANSF DIRの違い．

TRANSF DIRはあくまでもPUとMEMORYとのDATA転送の方向を示すものでADDRESSについては転送指示は必要はない．C-COUNTERのC1，C2の点で，STOPさせたりするには，TRANSF DIRと違った信号が必要である．

C-COUNTERなどの点でSTOPさせるかの指示信号としてDIRが必要である．

TRANSF DIRの指示は，AKNが返ってきてから指示してもよいが，DIRの場合それをやると速度が遅くなるのでそれはできない．

以上の事柄を念頭におき，C-COUNTERの回路図を参照しながら基本的なタイムチャートを書く（時間関係については，インタフェースマニュアルおよびCH部のタイムチャートを参照のこと）．



INHCAIC…AUTO INDEXのとき〝1〃にする
TWC　　2CYC指定のとき　　〝0〃にする
INC　　MBINCRのとき　　〝0〃にする
} 詳細はインタフェースマニュアルを参照．

LNK…TAD命令実行時のオーバフローのSET以外に，下記のOPR 1，命令の組合せを満足しなければならない．

① CLA，CLL
② CMA，CML
③ IAC

④ RAR, RAL, RTR, RTL, BSW

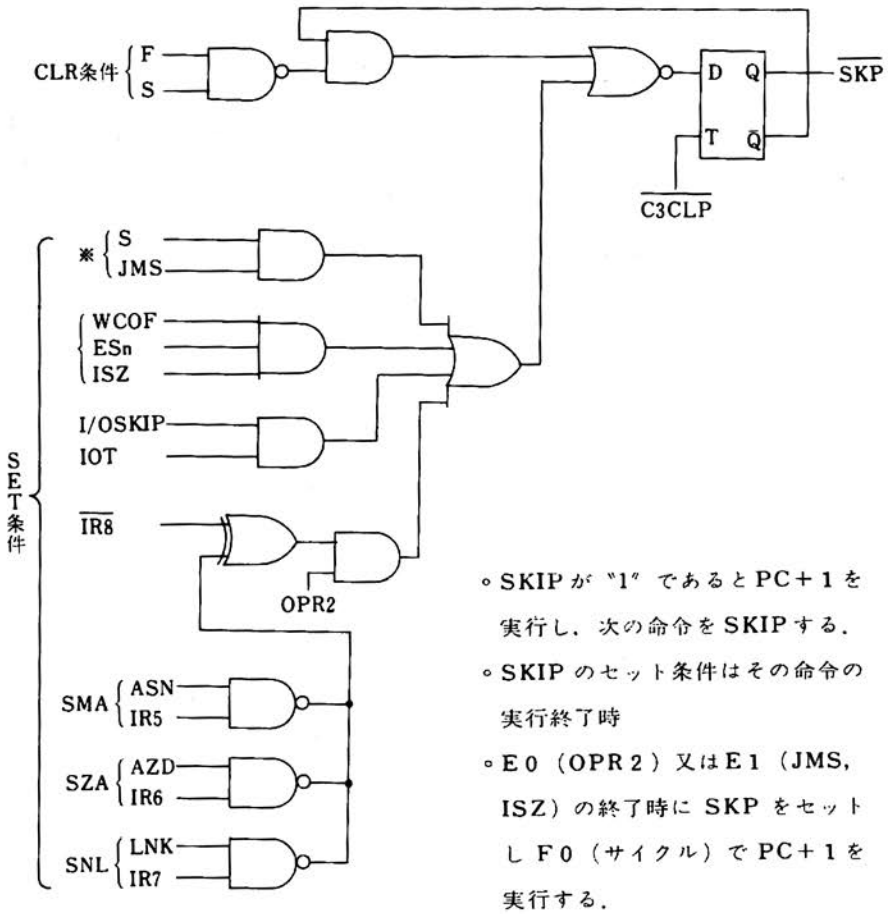
CLL (IR5)	CML (IR7)	IAC (IR11)	ALCO	LNK (t)	LNK (t+1)	
0	0	0	0	0	0	TAD命令実行
			1	0	1	シフト命令
0	0	1	0	0	0	
			1	0	1	
0	1	0	——	0	1	CML
				1	0	
0	1	1	0	0	1	
			1	1	0	
1	0	0	——	0	0	
				1	0	
1	1	0	——	0	1	CLL
				1	1	
1	1	1	0	0	1	STL命令
			1	1	1	

LNKのセット条件は上記のテーブルを満足すればよい.

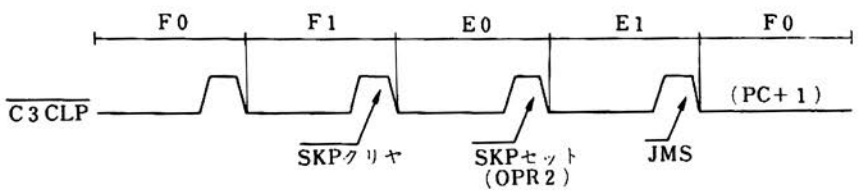
STL CIAは上記のテーブルの中に含む.

$$\text{STL CIA} = \underbrace{\text{CLA}\cdot\overbrace{\text{CLL}\cdot\text{CMA}}^{\text{STL}}\ \text{CML}\ \text{IAC}}_{\text{CIA}}$$

SKIP　下記に示す回転構成になっている. SKIPはALのALC1のタイミングと密接にからんでいる.



- SKIPが"1"であるとPC＋1を実行し，次の命令をSKIPする．
- SKIPのセット条件はその命令の実行終了時
- E0（OPR2）又はE1（JMS，ISZ）の終了時にSKPをセットしF0（サイクル）でPC＋1を実行する．

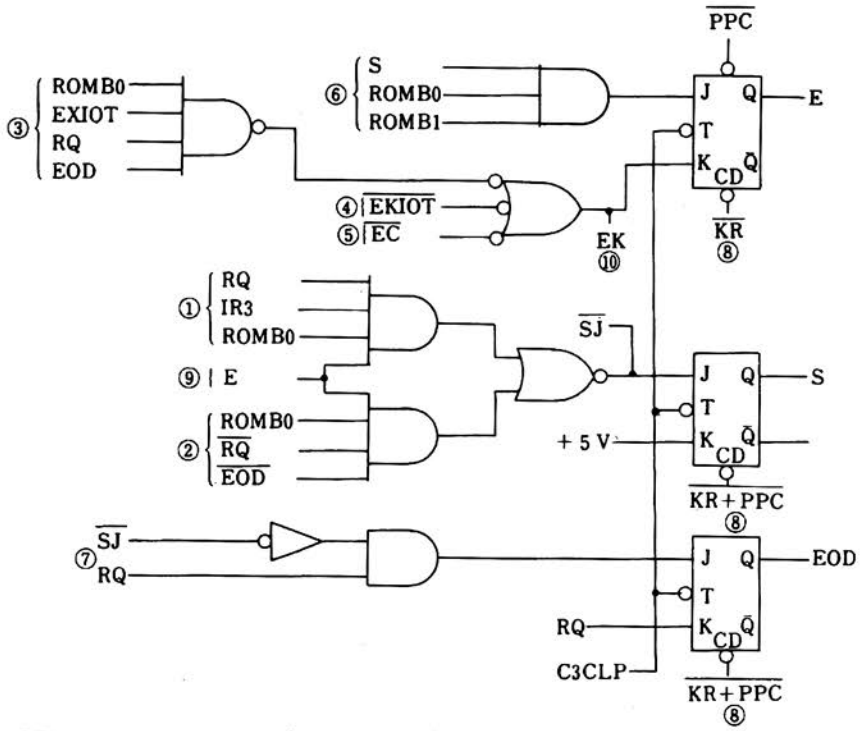


※ JMS，Sのセット条件

JMS Yを実行した場合

JMS YはY番地に（PC＋1）を書込み，Y＋1から実行．

Y＋1はF0（サイクル）で実行する．そのため，SKIPをセットする必要があるのでこの条件は必要．



STATUS GEN…このブロックは下記に示す回路構成から成立っている．E，S，EODの3つのFFだけで，CPU内の仕事のスケジュールを定めている．

①②はSFFのクリヤ条件である．

①はインダイレクト指定，②はIOT，OP1，OP2，OP3，JAPの場合である．

③④⑤はEのクリヤー条件である．

③はOP1，OP2，IOT（IOP，1，2，4を使用しない場合）のとき実行終了後Eをクリヤする必要がある．④はIOT（IOP1，2，4使用）のとき実行終了時に発生　はOP3，JAP命令のとき実行終了時に発生．

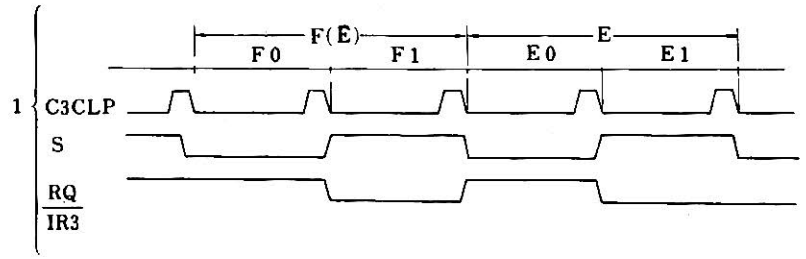
⑥はEのセット条件でF1からE0に入るときの必要条件である．

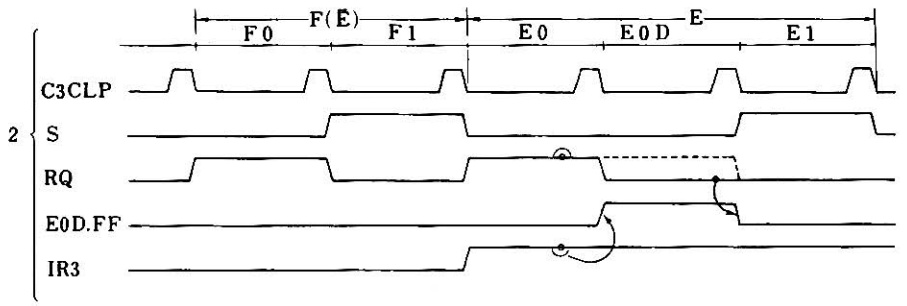
⑦はインダイレクト指定のときで，①の項が満足されたときEODFFのセット条件となる．

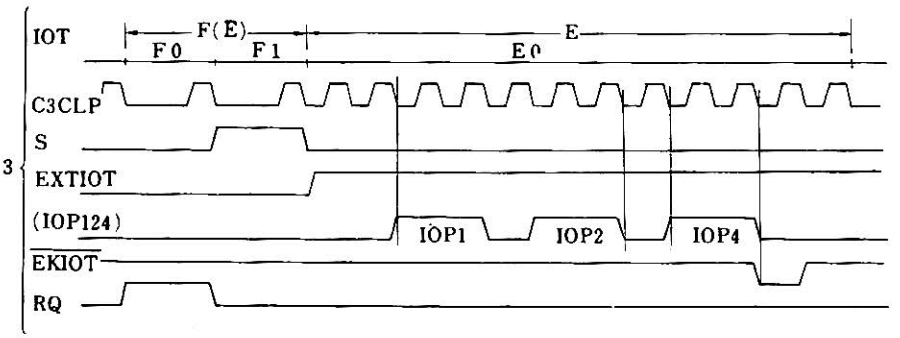
⑧イニシャライズでKRは前面パネルのSTART KEYを押したとき必ず発生する．

⑨はマニュアル操作の場合はE＝0で，SJ＝1のため，Sは0→1と繰返すだけである．

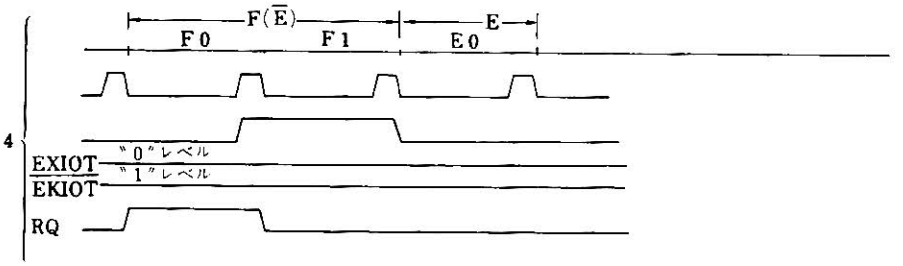
⑩はこの信号は，RUNをSTOPさせたい場合，命令実行終了時にEをクリヤすると同時にSTART/STOP回路のSTPFFのSTOPSELECTのゲートを開く信号に用いられる．

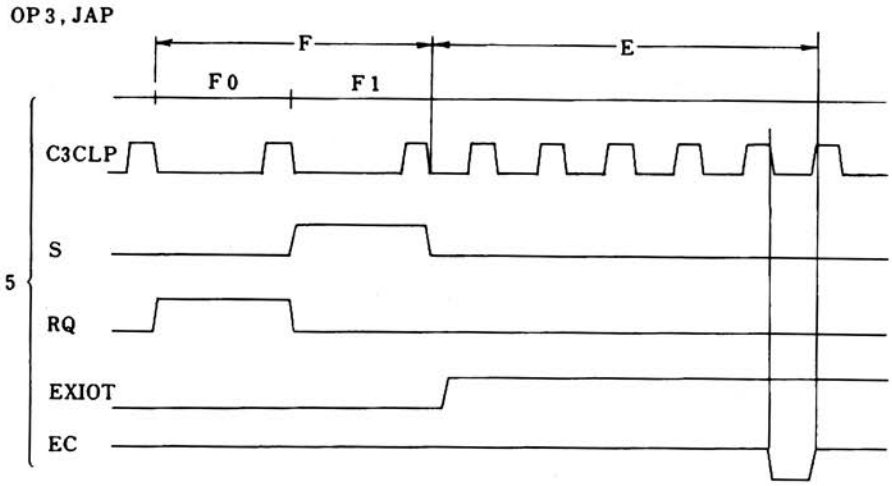






IOT（IOP1，2，4を使用しない場合），OP1，OP2





(1) のタイムチャートはMRI命令の直接アドレス指定の場合である.

(2) のタイムチャートはMRI命令の間接アドレス指定の場合である.

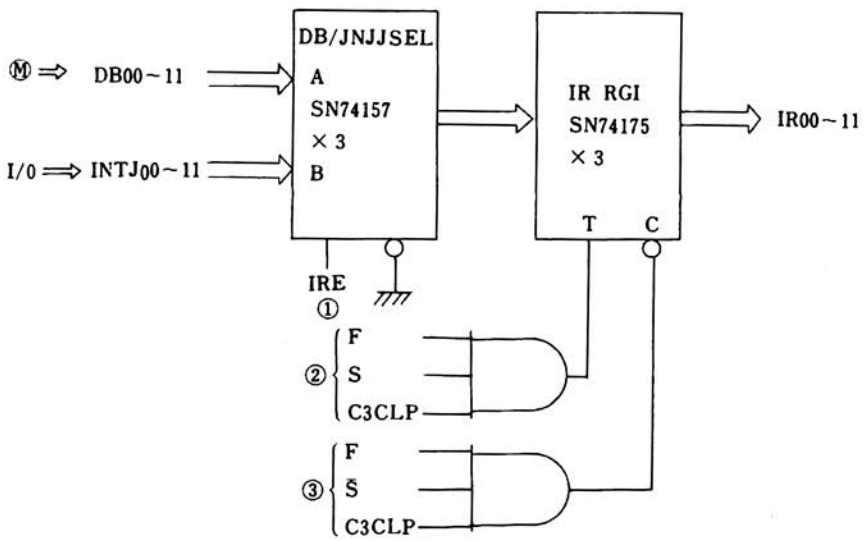
(3) のタイムチャートはIOT命令（IOP 1，2，4）の場合である.

(4) のタイムチャートはIOT命令(IOP 1，2，4 使用せず)OP 1，OP 2.の場合である.

(5) のタイムチャートはJAP，OP 3 の場合である.

IR…命令を解読する.

回路構成は下記のようになっている.

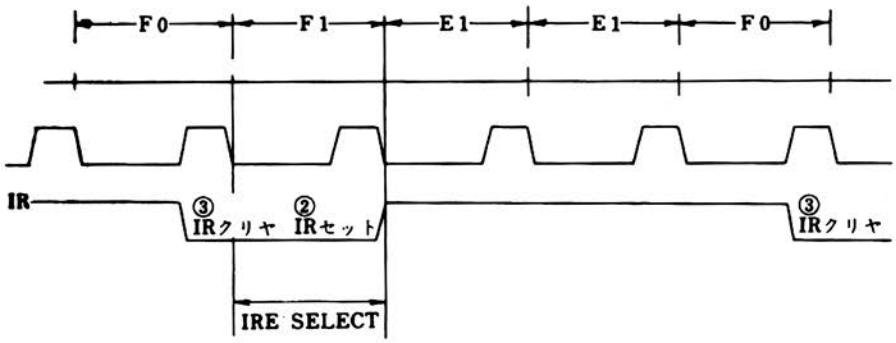


① IREは，DBかINTJを，SELECTする信号でROMの出力信号である. IRE＝0のとき，A側をSELECT，IRE＝1のとき，INTJ側をSELECTする.

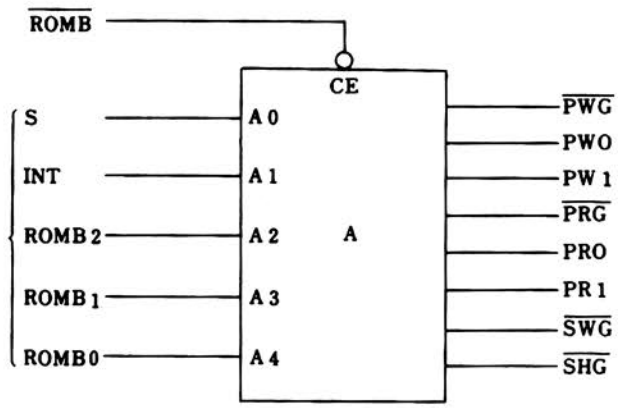
イ）PCで指定された，ADDRESSから読み出した命令をIRにセットするには，A側をSELECTする.

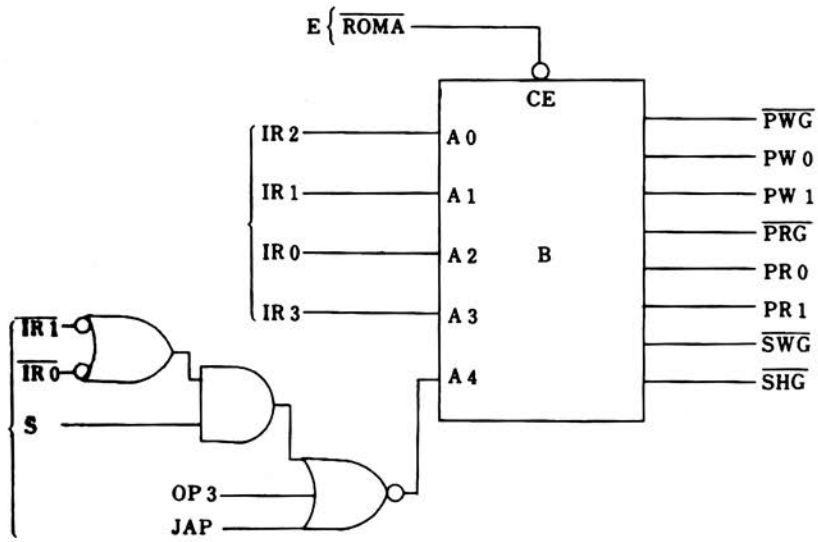
ロ）INTERRUPTのとき，I/Oから送られる，JMSXXをIRにSETするときB側をSELECTする.

② はIRのセット条件，③ はクリヤ条件である.



ROM…PU内の各ブロックのゲート信号をADDRESSを指定することによって選択する．8ビット32ワードで構成される．



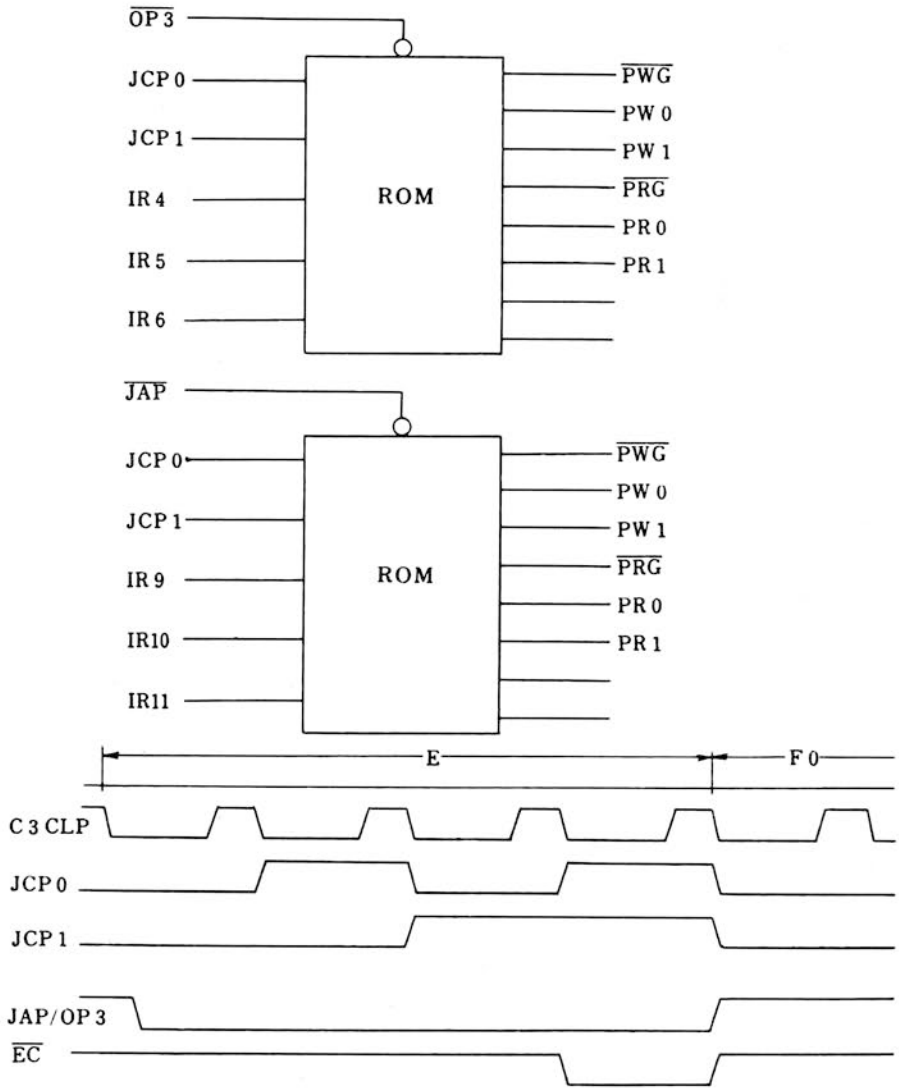


1）AはMANUAL操作，FETCHサイクルにS，INT，ROMB 0～2のADDRESS信号によって，各ゲート信号を発生させる．

2）BはEサイクルの各ブロックのゲート信号をIR 2 ～IR 0，IR 3，OP 3，JAP等のADDRESS信号によって発生させる．各ブロックのADDRESSによるゲート信号の状態はROM TABLEを参照のこと．

ROM CONT…ROMのブロックではMANUAL操作F，Eサイクルの各ブロックのゲート信号を発生させているが，それでもおぎなえないゲート条件はこのブロックで作られる．

OP & JAP GEN…命令の解読およびOP 3，JAPのゲート信号発生．



JCP COUNTERはOP 3，JAPのADDRESS信号として用いられている．アドレス信号に対する，各ゲートの状態は，ROM TABLE参照のこと．

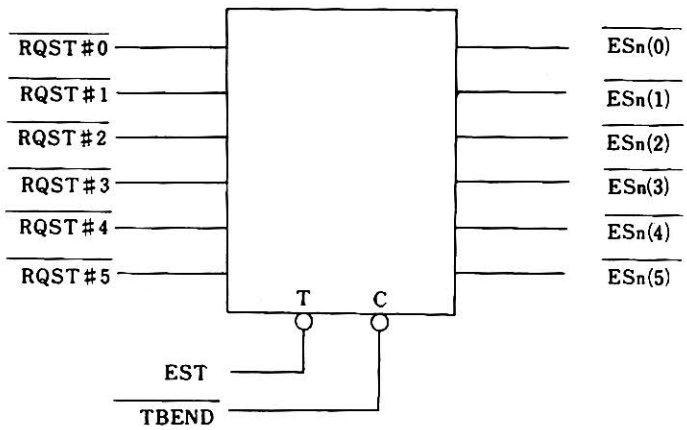
5—4 CH部

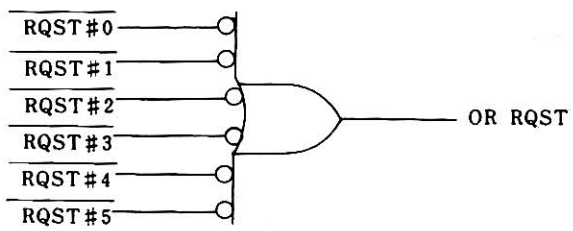
ES*n*…EQUIPMENT SELECTの略

このブロックはI/Oから，REQUESTがくると，REQUESTを受付けたという返事として，ESn信号をI/Oへ転送する．

ESn回路は，優先順位がもうけられる．I/Oから同時に，REQUESTがきた時は，優先順位の高い方から受付け，データ転送が終了するまで，他のREQUESTは受付けない．

優先順位は番号の若い方から高くなっている．

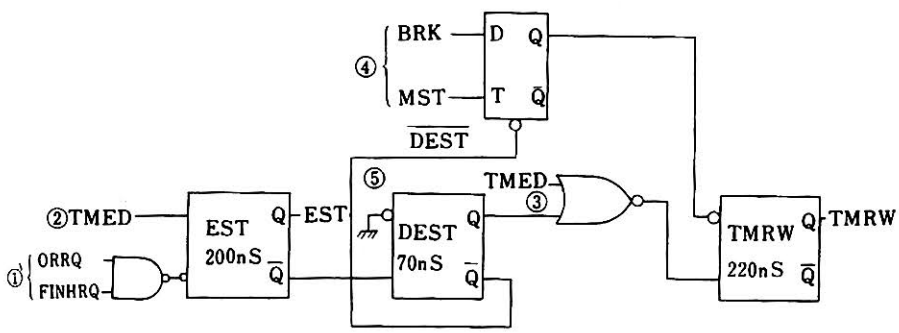




EST　　：RQST#nがくると，ESTタイミングが発生．

TBEND：BRKサイクルの後縁で発生．

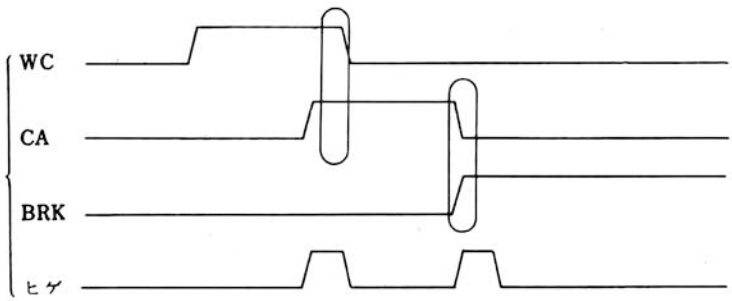
TIMING GEN　各ブロックのGATE条件を作るための基本タイミングを発生．



① はESTのタイミング発生条件，WC，CA，BRKのいずれか動作中はESTタイミングは発生せず（REQUESTは受付けない）．

② は，WC→CA→BRKと変化時にESTタイミング発生条件に，ヒゲがでてもESTタイミングが

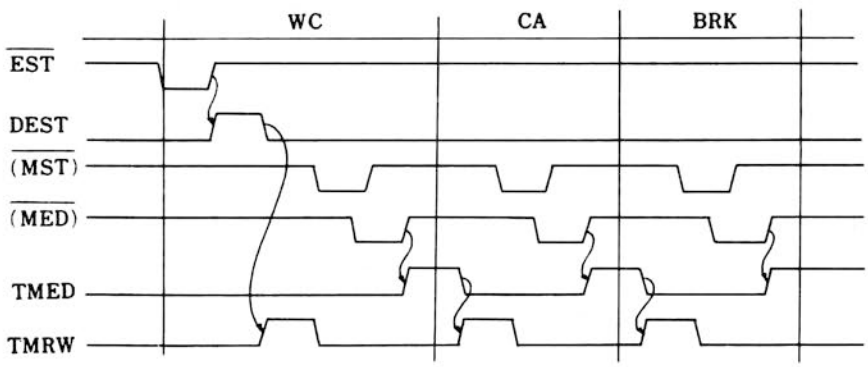
誤動作しないための防止である.

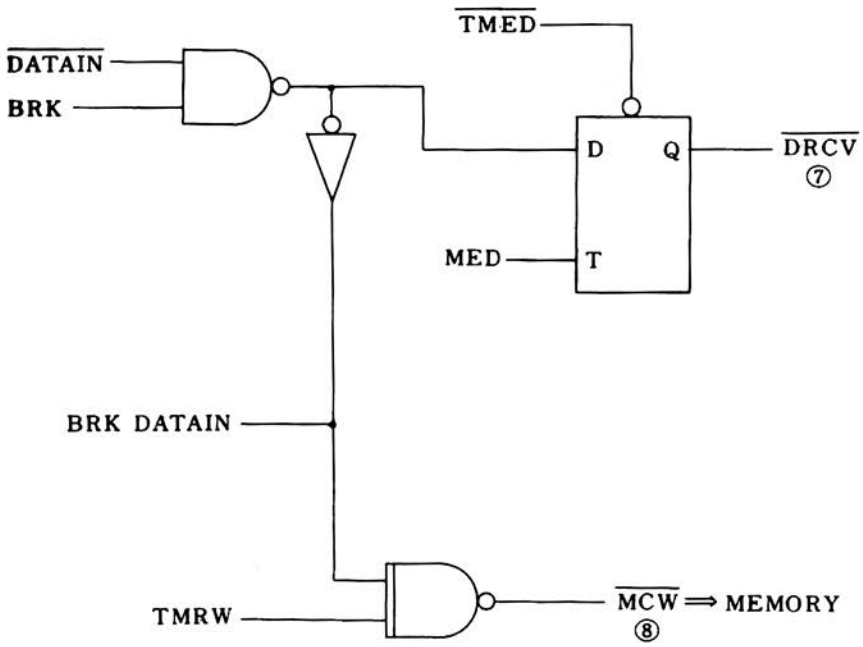


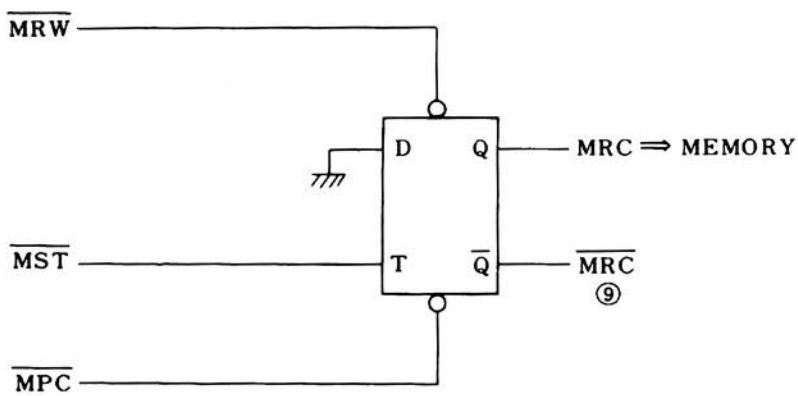
③ TMRWを発生させる条件

④ BRKサイクル終了後にもTMEDが発生するため，BRKサイクル終了時にはTMRWを発生防止の回路.

⑤ ESTが発生したとき，TMRW信号も発生させる必要がある．そのための解除である.





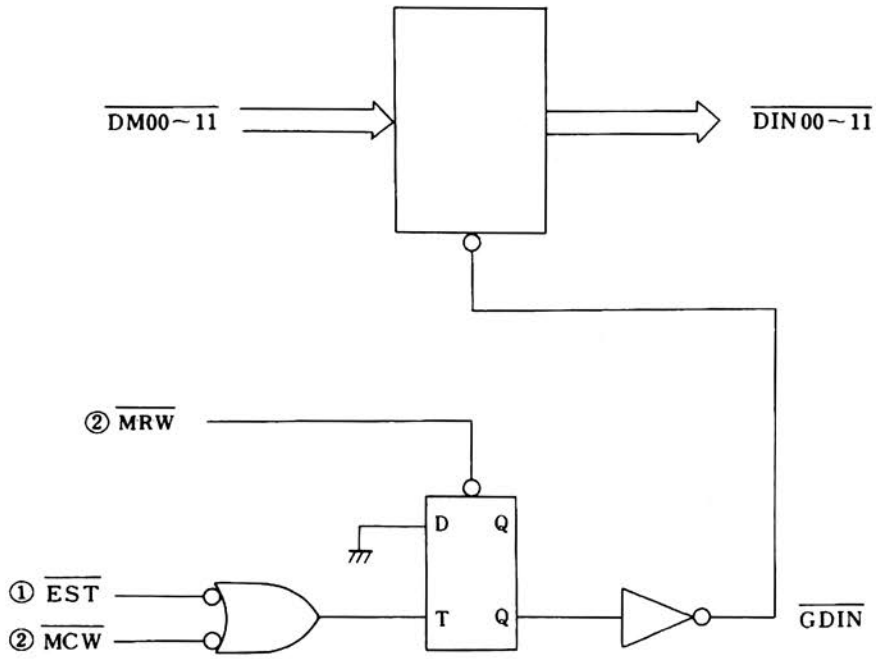


⑦ DRCVはTLDMBでDMB COUNTERにDATAが書込まれた後DRCVの信号を発生DRCVはデータを受取ったという返事としてI/Oへ転送する.

⑧ MCW信号はMEMORYへ送る信号（MCW＝0で送る).

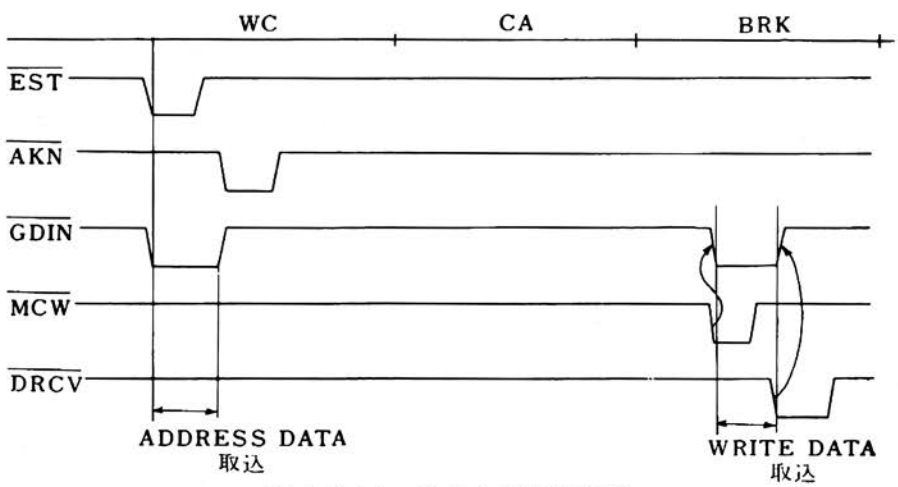
⑨ MRC信号はI/OからDATAの書込指定があったときBRKサイクルでMEMORYからのデータを禁止する. MRC信号はMEMORYへ転送する（MRC＝1 WC, CAサイクルではMRCはONである〈MEMORYからのデータ転送可〉).

DATA SELECT (I)…I/Oから転送されてくるDATAをADDRESS DATAかWRITE DATAかSELECTする.



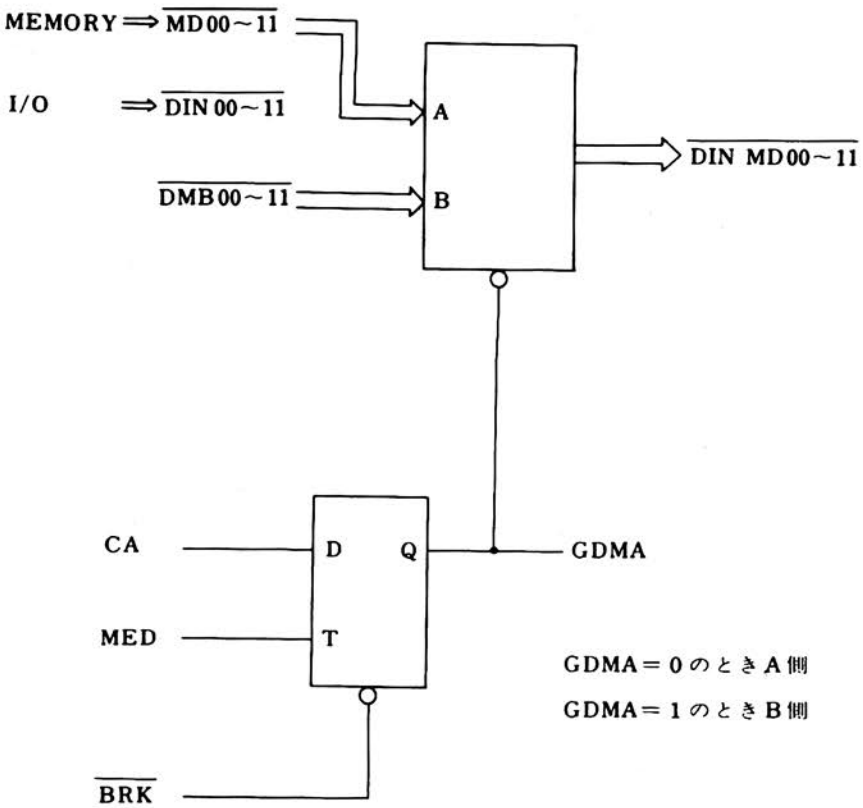
① ESTが発生するときは必ずADDRESS DATAであるからGDINをONしMRWでGDINをOFFにする.

② I/OからWRITE DATAであるならばMCW信号によってGDINをONしてWRITE DATAを取込む.



（1サイクル，2サイクルも同じ）

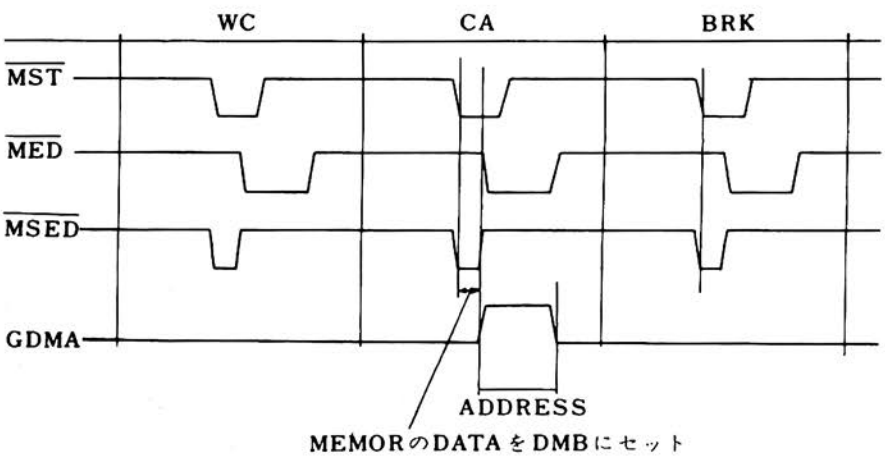
DATA SELECT (II) …DIN (MD) かDMBかをSELECTする.



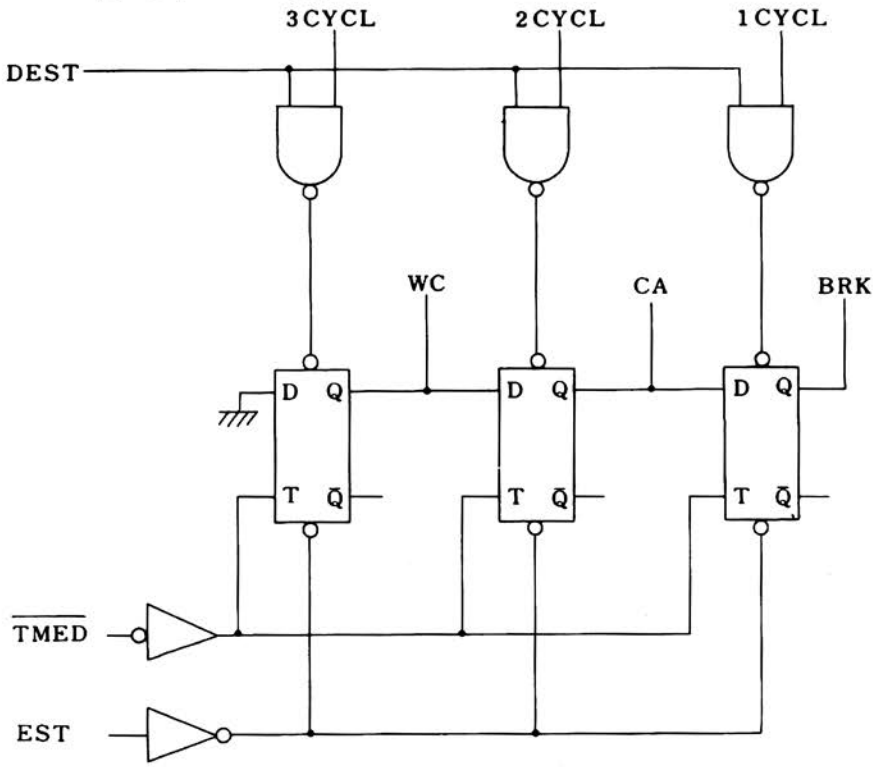
B側をSELECTする条件

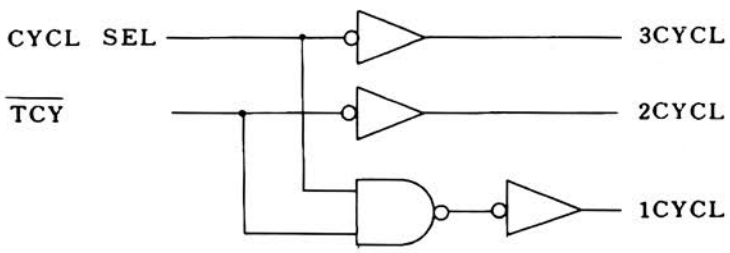
3CYCまたは2CYC指定のときCAサイクルでB側をSELECTする.

B側のDATAはADDRESS DATAとしてDMAに転送する.



CYCLGEN…CH内の基本サイクルを作る.





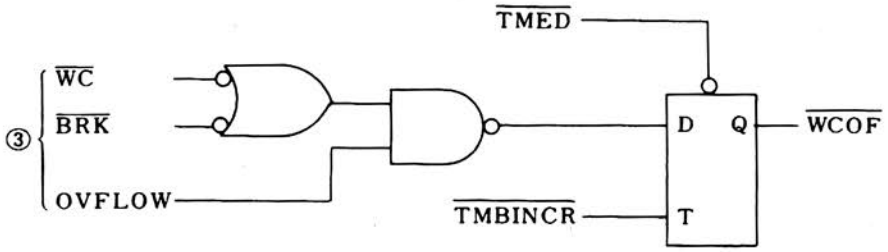
3 CYCL, 2 CYCL, 1 CYCLは必ずいずれか1つのみ1であること.
同時に2つ以上1になってはならない.

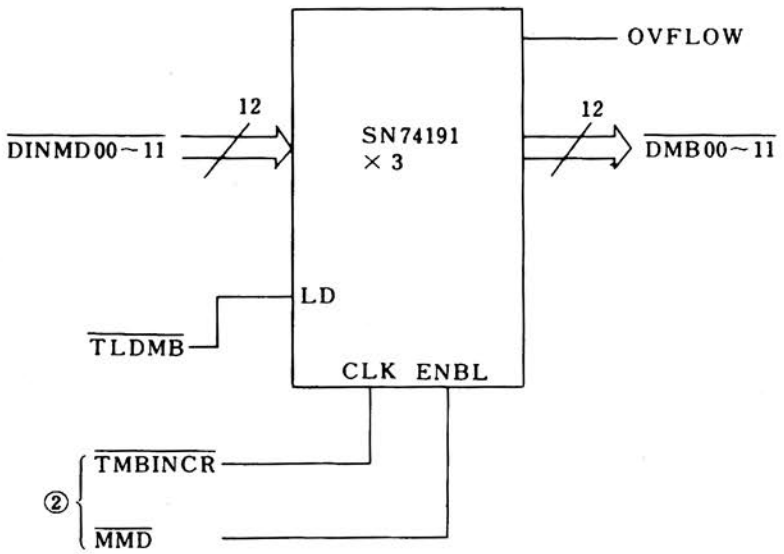
DEST：サイクルフリップフロップのセットタイミング

TMED：サイクルの切換タイミング

EST：クリヤ条件

DMB COUNTER…このブロックはI/OからのWRITE DATAまたはREAD DATAとしてI/Oへ転送する場合すべてこのDMB COUNTERを経由する. DMBにセットされたDATAを＋1したりするのもこのCOUNTERの仕事である.





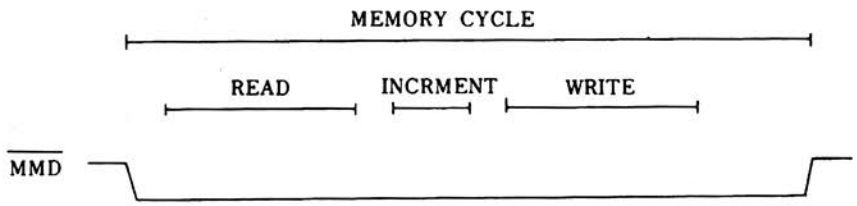
① DMB COUNTERにDATAをセットする条件でI/OからのWRITE DATAをセットするときは$\overline{\text{TLDMB}}$でセットする.

② は①によってセットされたDATAを、INCREMENTしたい場合は$\overline{\text{MMD}}$=0にして$\overline{\text{TMBINC}}$の（ ）後縁でINCREMENTする.

③ WCOFのセット条件で②でINCREMENTの結果OVFLOWが発生すれば，上記の条件でWCOFは発生する.

TIMING GATE…各ブロックの基本動作を定める.
TIMINGの，OR GATEである.

① MMD =WC+CA·CAINCR+BRK·MBINCR
MMDは，MEMORYのCYCLE TIMEを可変する信号である.
MMD=0のとき，CYCLE TIMEは長くなる.
INCRMENTするときの時間分だけ長くなる.



② TLDMA=$\overline{EST}$+CA·MED

TLDMAは，DMA REGISTERに，ADDRESS DATAをSETするタイミング信号として使用される．

③ AKN =TMRW（WC+2CYC·CA+1CYC·BRK）

AKNは，I/Oからの，ADDRESS DATAを受取ったという返事として，I/Oへ転送する信号である．

④ MRW =TMRW（WC+CA+BRK·DATA OUT）

MRWは，MEMORYへ転送する信号である．MEMORYはこの信号を受取ると動作開始する．（MEMORYへの，$\overline{REQUEST}$信号になる．）

⑤ DAVL =(CA+BRK·DATA OUT)·$\overline{MED}$·$\overline{MST}$·MMD

DAVLは，I/OへREAD DATAを転送するとき，この信号を発生する．

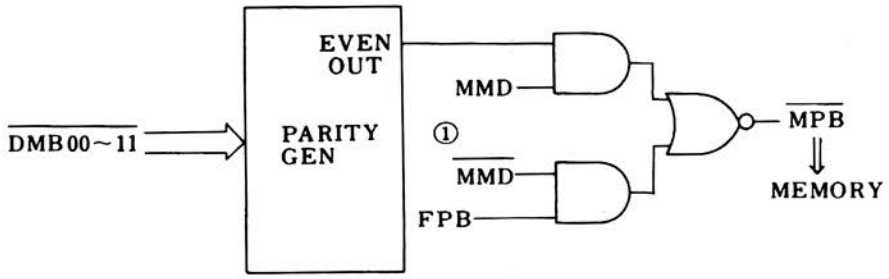
⑥ TMBINC=MMD·MST·MED

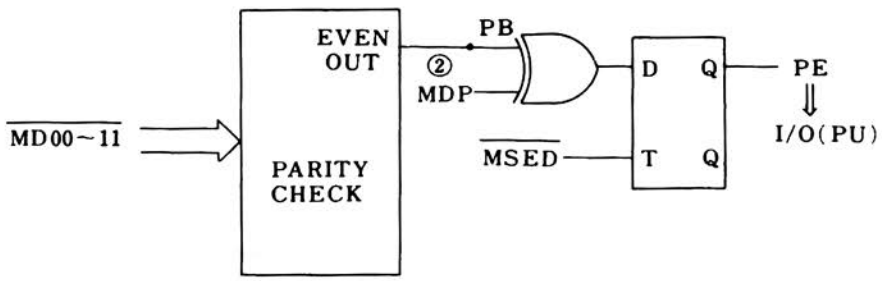
DMB COUNTERにSETされているDATAをINCREMENTしたいとき，この信号を用いる．MMD=0のとき，この信号は発生しない．

PARITY GEN……MEMORYへPARITY BITを入れる．

PARITY CHECK…MEMORYに転送されたDATAが誤りなくWRITEされたか調べる．

PARITYはすべてEVEN PARITYである．



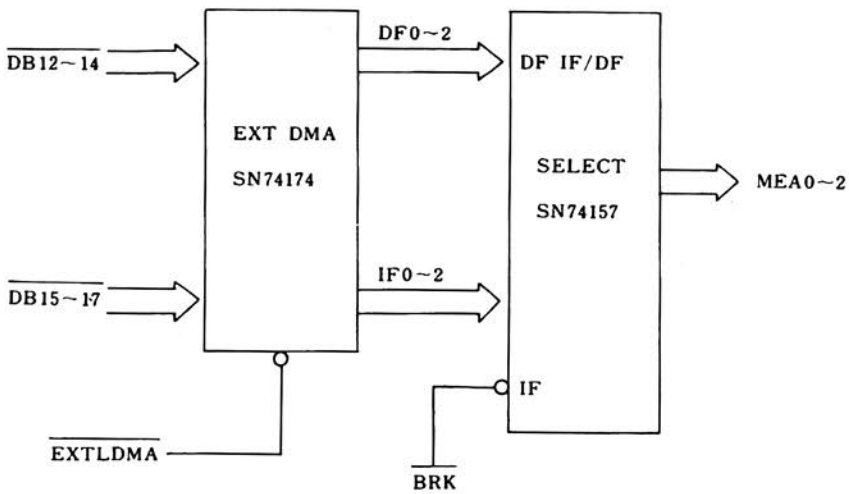


① DMB00～11，12本のDATAで“1”の数が偶数個あった場合EVEN OUTは“1”となる．
$\overline{MMD}$＝1のとき$\overline{MPB}$＝0としてMEMORYにかく．
$\overline{MMD}$＝1のときはMEMORYへ書込み時間の関係からPARITY CHECKの回路よりPARITY BITをもらう（$\overline{MMD}$＝1のときMEMORYのCYCLETIMEは早くなる）．

② MDPはMEMORYから転送されてくるPARITY CHECK DATAである．
①で書込まれたPARITY BIT（MPB）と対応している．
PBとMDPが共に1の場合はPARITY ERROR（PE）にはならない．

EXT DMA（I/DF SELECTも含む）

555Hでは，1FIELD，4KW単位で，8FIELD，32KWまで，MEMORYを増設することができる．MEMORYが増設された時，どのFIELDを使用するかをSELECTすることが必要になる．EXTLDMAはFIELD SELECTである．



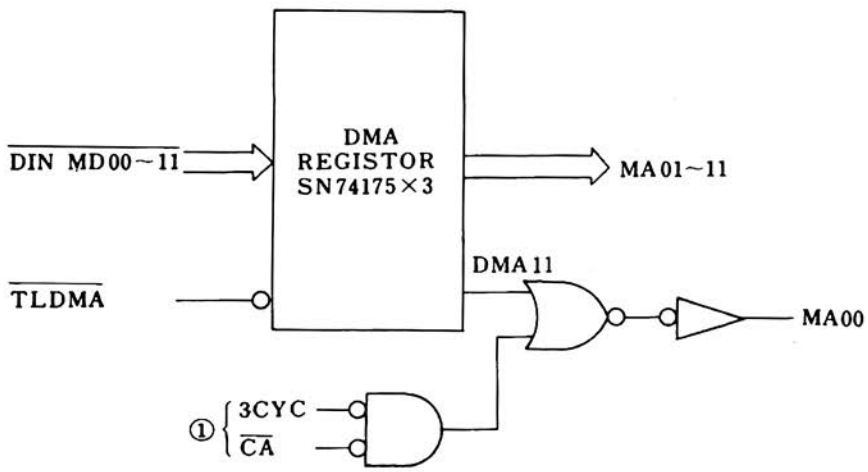
$\overline{BRK}$=0 → DF側

$\overline{BRK}$=1 → IF側

① EXTLDMA＝DEST（WC＋CA＋BRK）の条件である.

② DF IFは、BRKサイクルのときは、DF側をSELECTする. その他の場合は、IF側である.
WC、CAサイクルは、IF（INSTRUCTION FIELD）として扱われる.
BRKサイクルは、DF（DATA FIELD）として扱われる.

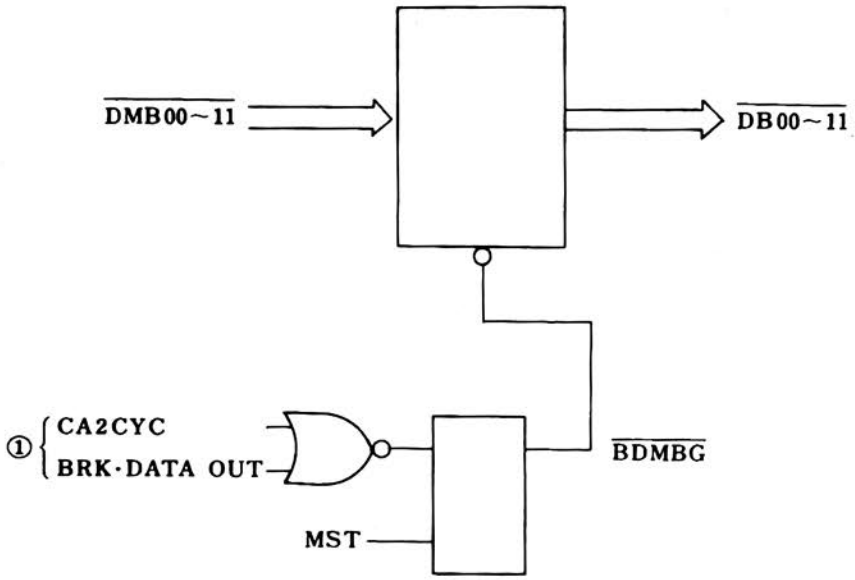
DMA REGISTER…12BITのREGISTERでADDRESS DATAとしてMEMORYへ転送する.



① 3CYCLE指定のときWC（WORD COUNTER）の書込まれているMEMORY ADDRESSは偶数番地としCA（CURRENT ADDRESS）の書込まれているMEMORY ADDRESSはWCの次の番地であると決められている. そのために必要な回路である. 3CYCLE指定以外は用いない.

② TLDMAの条件はTIMING GATEの項に記載.

BUFFER DMB…I/OからDATAOUT PUT指定があったときREAD DATAをI/Oへ転送する.



① I/OへREAD DATAを転送するときの，DMBGのON条件である．READ DATAは，DATA OUT PUT指定で，BRKサイクルのとき，MSTによってGATEを開ける．CA・2CYCの条件はCPUがJMSIXYを実行するとき必要となる．

転送方向	信号名	本数	内容	レベル
周辺装置 → 5555H	RQST n＝0～5	1本／CH	DSAの要求信号，各周辺装置独立の信号	+3V / 0V
	CYCSEL	1本	1CYCLEか，3CYCLEが指定する信号 0V：3CYCLE指定 3V：1CYCLE指定	+3V / 0V
	TRANSF DIR	1本	DATA転送方向を指定する信号 0V：DATA OUT PUT指定 (READ DATA) 3V：DATA IN PUT (WRITE DATA)	+3V / 0V
	INHCAINCR	1本	2CYCLEおよび3CYCLE指定の時CACYCLEにおいて，DMBの内容に，INCRIMENTを禁止する信号 3V：禁止	+3V / 0V
	MBINCR	1本	WRITE DATAまたはMEMORYからのREAD DATAをMODIFYする信号．DSA CYCLEの時有効．WC,CAサイクルの時無効	+3V / 0V

転送方向	信号名	本数	内容	レベル
555H → 周辺装置	※ $\overline{DB00\sim11}$	12本	ADDRESSDATA, WRITE DATA, READ DATAは，すべてこのDATA LINEにのせる	+3V 0V
	$\overline{DB12\sim14}$	3本	MEMORYのFIELD指定4KW単位，8FIELD，32KWまで，指定することができる．	+3V 0V
	$\overline{ES_{n=0\sim5}}$	1本／CH	I/OからのRQSTを〝受付けた〟という返事信号，ES_Nが続行中は他のRQSTは受付けない．	+3V 0V
	$\overline{BAKN}$	1本	ADDRESS DATAを〝受付けた〟という返事信号	+3V 0V
	$\overline{BDAVL}$	1本	TRANSF DIRがDATA OUT PUT指定の時，READ DATAのSTROBEとして，この信号が発生	+3V 0V
	$\overline{BDRCV}$	1本	TRANSF DIRが，DATA IN PUT指定の時，WRITE DATAを〝受取った〟という返事信号	+3V 0V
	$\overline{BBRK}$	1本	DSA CYCLE期間信号で，この時間帯にREAD DATAおよびWRITE DATAが定まる．	+3V 0V
	$\overline{BWCOF}$	1本	3CYCLE指定の時WC CYCLEにおいて，WORD COUNTEREGISTERがオーバーフローした時，または1CYCLE指定で，MBINCRで，READ DATAをMODIFYした時，DMB COUNTERがオーバーフロした時，この信号が発生．	+3V 0V
	BTS 1	1本	TIMING信号． DATAをCLEARしたり，CONTROL信号に用いてもよい．	+3V 0V
	BTS 2	1本	TIMING信号	+3V 0V
	※ $\overline{DB00\sim11}$	12本	READ DATAを転送する． BUSLINE	+3V 0V

5—5 I/O部

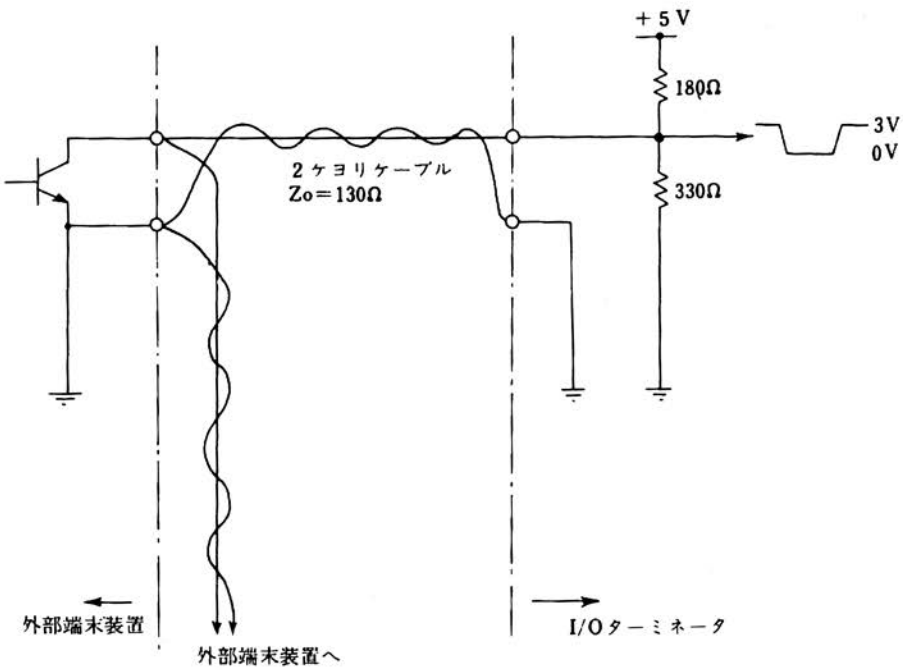
5—5—1 概 要

555HI/O部にはテレタイプ（TTY），高速紙テープパンチャ（PTP），高速紙テープリーダ（PTR）のコントロール，リアルタイムクロック，スーパーバイザーコール，インタラプトコントロール，フィールドセレクトおよびIOPパルスジェネレータなどの回路が含まれる．外部端末装置から，PUアキュムレータへのデータインプット，インタラプトリクエスト，スキップ等の信号はすべてI/O部に入力される．DSA方式によるデータ転送はCH部経由で行われる．

5—5—2 各部の説明

(1) INT. SKP. ACBUS. TERMINATOR

555Hと外部端末装置とのデータ転送は，バスラインを通して行われる．外部装置からI/Oへの，信号入力のインターフェースは1図の通りである（インタフェースの詳細は3．I/Oインタフェースの項を参照）．

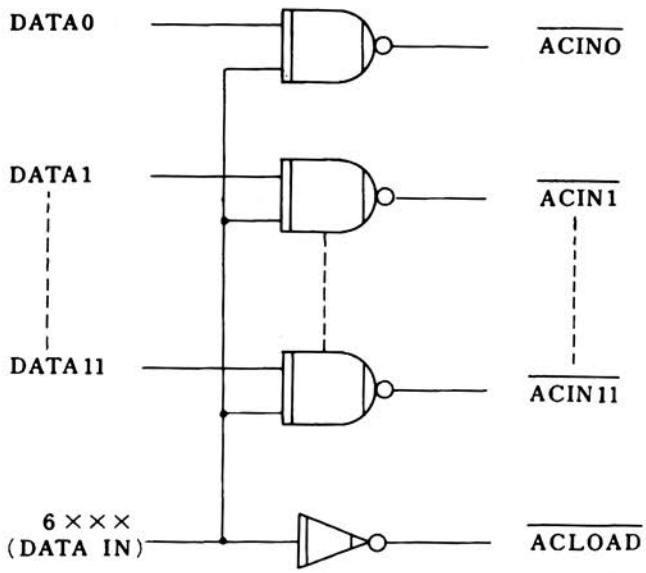


1図 外部端末装置→I/O信号入力インタフェース

(2) AC IN. ACLOAD. ACCLR. IO SKIP

外部装置からの入力信号，TTY，PTR，PTP，EXTCLOCK，SVC等の信号を信号種別ごとに1本にまとめ，PUへ送る回路で，オアゲードで構成される．

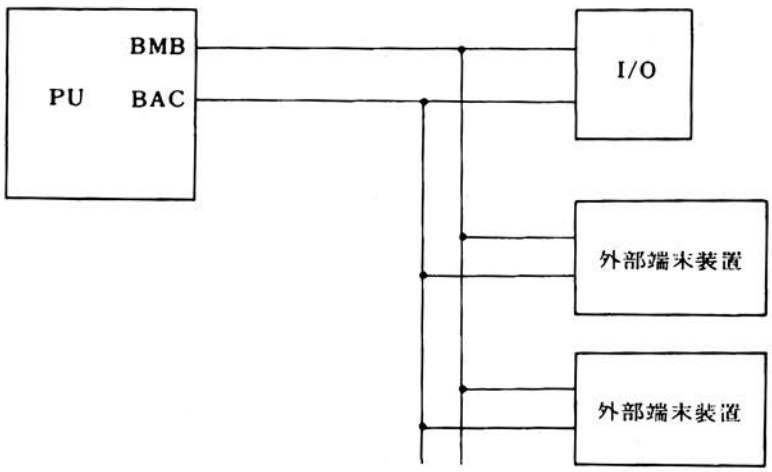
従来の555計算機と異って，データをACに書き込む場合，555Hではデータと共にACLOAD信号が必要である．



2図 AC LOAD

(3) **MB AC RECEIVER**

BMB, BAC信号は, PUより出力される. この信号はI/Oだけでなく, 外部装置も使用するため, I/Oのロードファクターは1に止めてある.

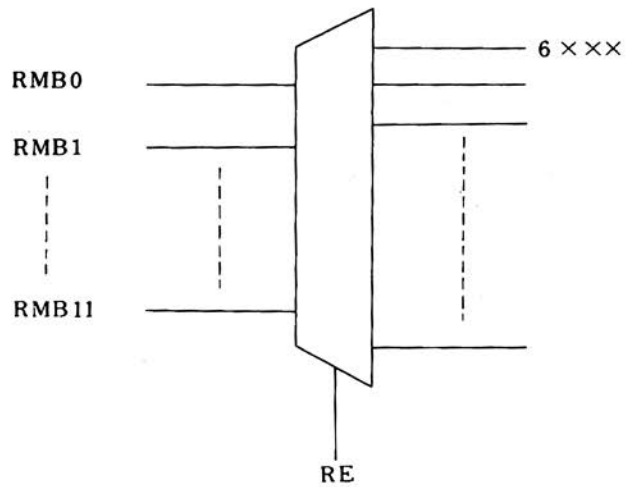


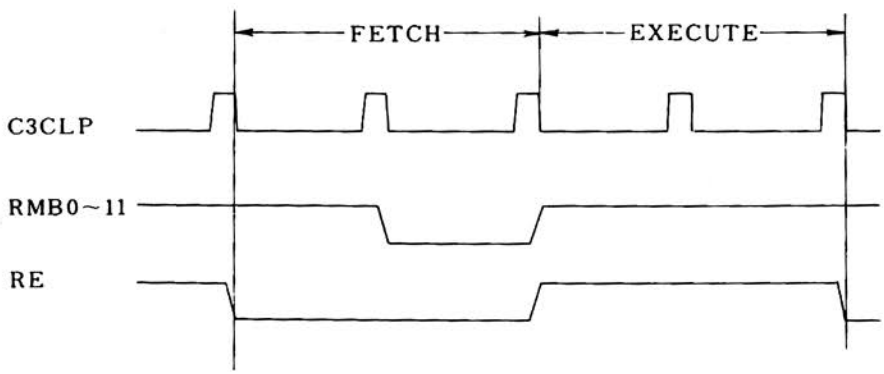
3図 BMB, BAC

(4) **DEVICE SELECTOR 1, 2**

I/O内部で使用される命令コードのデコーダである. 命令コードの詳細は, 8命令コード表を参照のこと.

BMB信号 (I/O内部ではRMB) は, BE (I/O内部ではRE) (EXECUTECYCLEの間1) で同期をとっている.



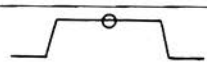
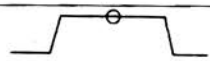
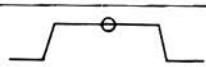
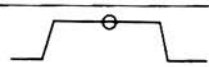
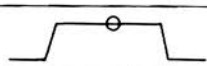
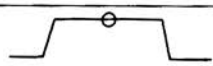


4図 IOTデコーダ

(5) **DF, IF**

データフィールドレジスタ (FDF0, 1, 2), インストラクションフィールドレジスタ (FIF0, 1, 2), セイブフィールドレジスタ (FSF0, 1, 2, 3, 4, 5) およびインストラクションバッファレジスタ (FIB0, 1, 2) で構成される.

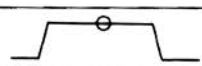
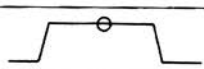
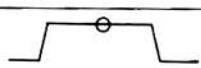
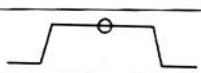
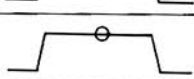
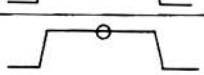
(A) データフィールドレジスタ（FDF0, 1, 2）

命令又はパルス	FDFに書き込まれるデータ	書き込みのタイミング
LD	SRDF0, 1, 2	
RAST	0	
INT	0	
CDF	MB06, 07, 08	
RMF	FSF3, 4, 5	
RTF	AC09, 10, 11	

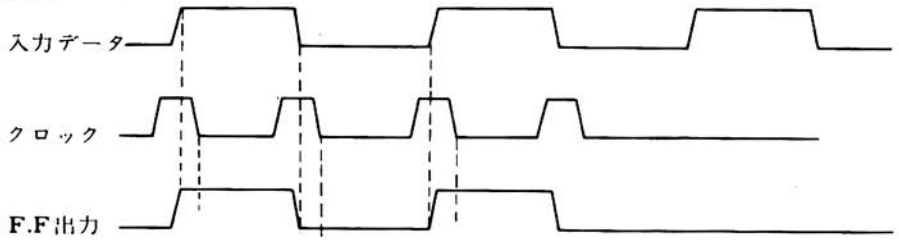
(B) インストラクションフィールドレジスタ（FIF0, 1, 2）

命令又はパルス	FIFに書き込まれるデータ	書き込みのタイミング
LD	SRIF0, 1, 2	
RAST	0	
INT	0	
JMP/JMS	FIB0, 1, 2	

(C) インストラクションバッファレジスタ（FIB0, 1, 2）

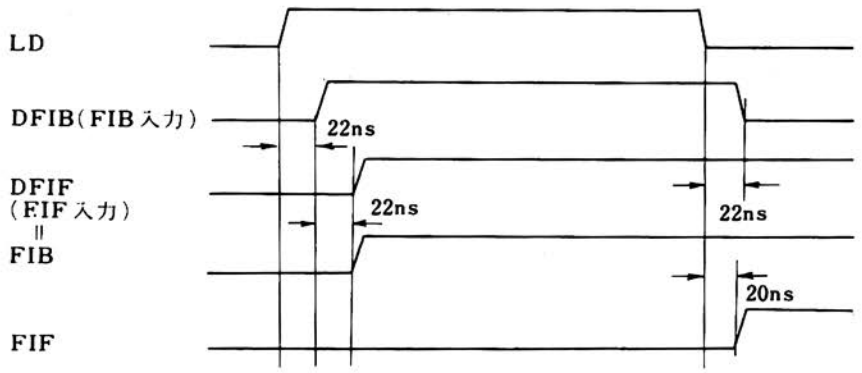
命令又はパルス	FIBに書き込まれるデータ	書き込みのタイミング
LD	SRIF0, 1, 2	
RAST	0	
INT	0	
CIF	MB06, 07, 08	
RMF	FSF0, 1, 2	
RTF	AC06, 07, 08	

FDFとFIBは，LATCH TYPE F.Fで，入力データと書込みクロックとのタイミング関係は，5図の通り．



5図　LATCH TYPE FFタイムチャート

クロックが1の間のみ入力データは有効で，クロックの終端時のデータがF.Fに保持される．
FIFとFSFは，EDGE TRIGGER TYE F.Fである．
LD実行時の時間関係を6図に示す．



6図　LD実行時のタイムチャート

(6)　IF、DF SELECT

①　CH部に対する，IF，DFの指定を行う．

インストラクションフィールド

FIFレジスタを，ES(5)でゲートして，CH部へ送る．

データフィールド

JMPまたはJMS命令以外の時に，ダイレクト指定ならばFIFレジスタを，インダイレクト指定ならばFDFを，ES(5)でゲートしてCH部へ送る．

JMPまたは，JMS命令の時には，FIBを，ES(5)でゲートして，CH部へ送る．

例1 （IF＝0，DF＝0）

```
CDF   ／DF＝1
TAD  Y／YはIF（0フィールド）から選ばれる．
TAD  iY／YはIF（0フィールド）から選ばれる．
```

例2 （IF＝0，DF＝0）

```
CIF  1  ／IF＝0  IB＝1
JMP  Y／JMPは，IF（0フィールド）から読み出される．
          Yは，IB（1フィールド）から選ばれる．
          JMP. Y実行後，IF＝1となってPCは，1フィールドのY
          番地になっている．
```

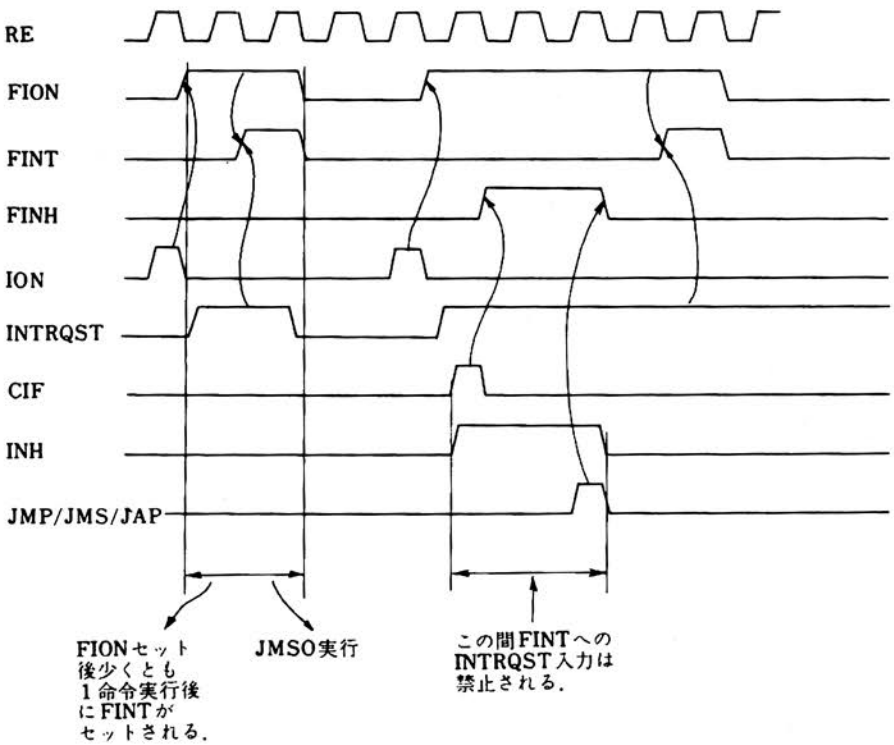
② IOT命令によって，IF，DF，SFを，ACに書き込む．

命令		RDF	RIF	RIB	GTF
AC	0				LNK
	1				INT RQST1
	2				INT RQST
	3				INH
	4				ION
	5				ION1
	6	FDF0	FDF0	FSF0	FSF0
	7	FDF1	FIF 1	FSF1	FSF1
	8	FDF2	FIF 2	FSF2	FSF2
	9			FSF3	FSF3
	10			FSF4	FSF4
	11			FSF4	FSF5

注）INT発生後，GTFで，ACにデータを読み込んだ時には，ION，ION1は，タリヤされている．ION，ION1を読むには．INT発生前に．GTFを実行すること．

(7) **INT. INT 1**

① INT発生時のタイムチャート



7図　INTタイムチャート

ION命令でFION（INTERRUPT ENABLE FF）がセットされる．この状態で，INTRQSTがあれば，REの後縁で，FINTがセットされる．ただし，FIONがセットされてからFINTがセットされるまで，少なくとも1命令時間必要である．FIONが1でINTRQSTがあっても，CIFまたはRTF命令が実行されるとFINTはセットされず，FINHがセットされ，INTRQSTは禁止される．FINHは，JMP，JMS，JAP命令のいずれかでクリヤされる．FION 1，FINT 1についても動作は同じであるがFINT 1によりJMS 0のかわりに，JMSXX（またはJMPXX，HLT，etc．半固定のワイヤリング方式で，任意の命令を設定できる．ただし，コードは，4，5，6，7に限られる）が実行される．

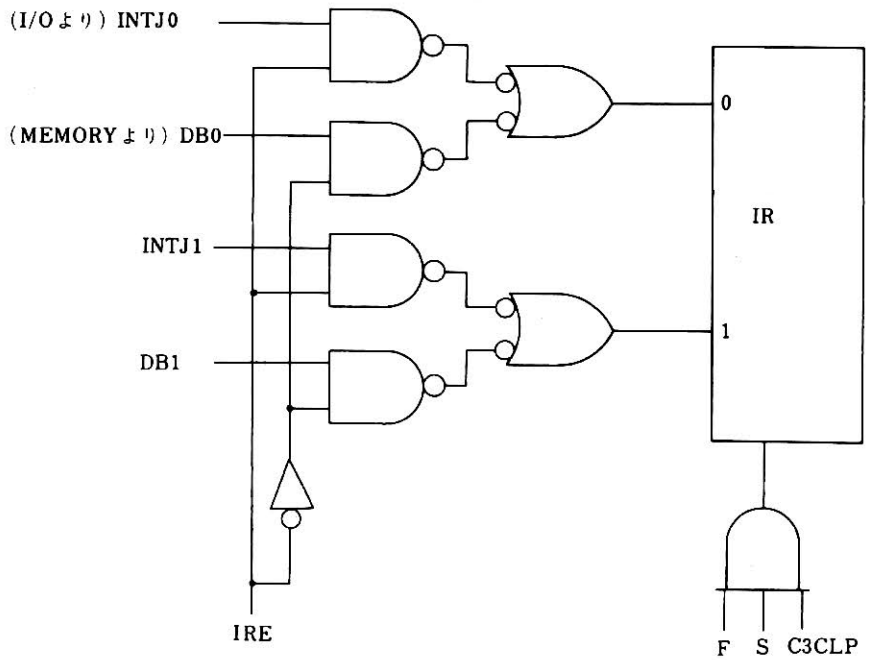
② FINTによるJMS 0の実行

FINTがセットされるとPUにINT信号が送られる．

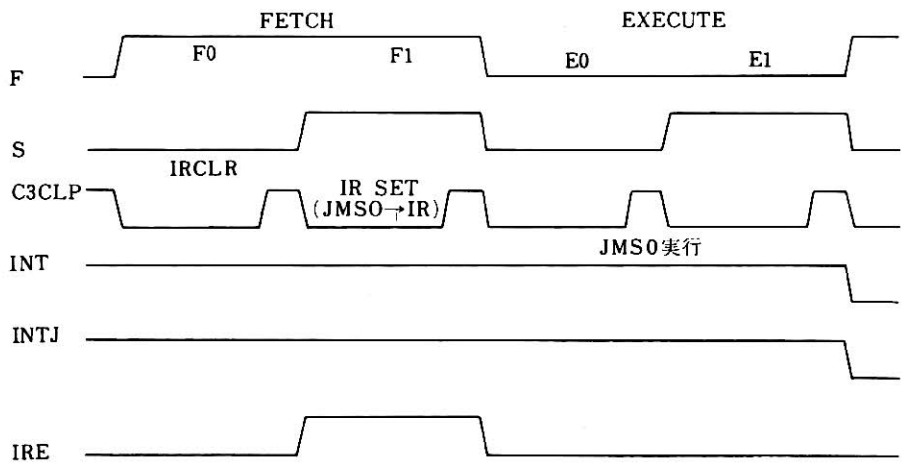
INT＝1の時，F 1サイクルでは，IRE信号が1となって，INTJ信号が選ばれる．

INTJ信号には，I/Oより，JMS 0（4000）が入力されている．

IR SET信号で，IRに4000が書き込まれ，Eサイクルで，JMS 0が実行される．



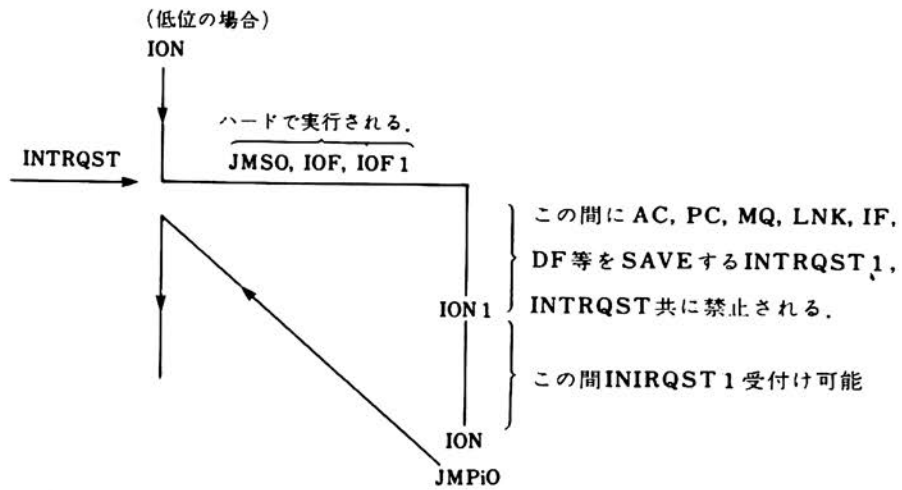
8図　IR(INSTRUCTION REGISTER)

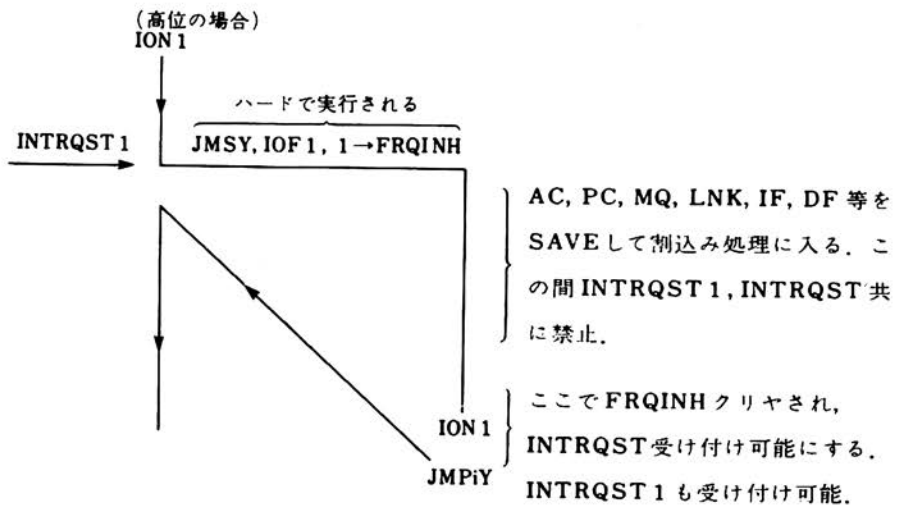


9図　INTにおけるJMS0の実行時のタイムチャート

③ INTとINT 1の優先順位

555H計算機では割込みラインは2本あってそれぞれINTRQST 1，INTRQSTとなっている．INTRQST 1はINTRQSTに優先する．即ちINTRQST処理中であってもINTRQST 1は受付け可能であるがINTRQST 1処理中はINTRQSTは禁止される．



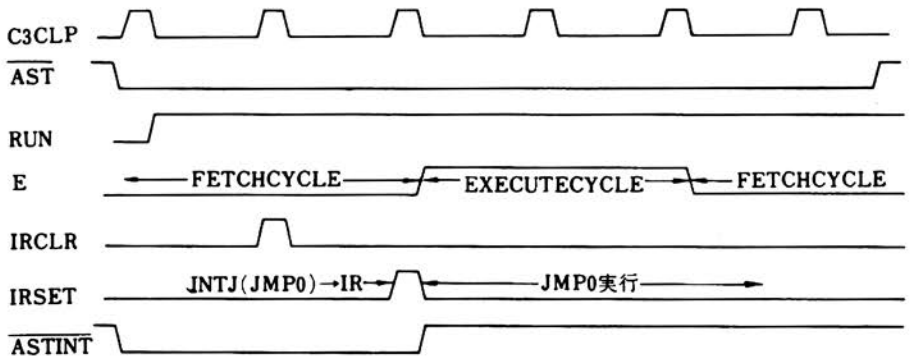


低位の割込み処理中，少なくともAC，PC，MQ，LNK，IF，DFをSAVEするまでは高位の割込みを禁止する必要がある．そこで低位の割込みが発生したときに，IOF 1を実行して（ハードで）一たん高位の割込みを禁止し，AC，PC，MQ，LNK等をSAVEし終った時にION 1（プログラム）で元に戻す．低位の割込みが発生した時には，高位は必ずION 1の状態であったはずだから無条件でION 1命令を実行してもよい．高位の割込みが発生したときに，低位がIONであったかIOFであったかは一意的には決まらない．従って高位の割込み発生時にハードでIOFを実行して低位の割込みを禁止しても元に戻す時に，IONは無条件で実行できないことになる．そこでFRQINH FFによって，高位の割込み処理中低位の割込みを禁止する．処理終

了後ION 1でFRQINHをクリヤして禁止を解除する．高位の割込み処理中，低位がIONであったかIOFであったかは無関係となる．FINT 1，FINTは同時にセットされることもあり得るが，この場合FINT 1が優先する（JMS 0よりもJMSYが優先する）．停電割込み（LPC，LOW POWER），メモリパリティエラー割込み（MPE）はINTRQST 1に属する．

④ AUTO START

AUTO STARTパルスは外部（電源部）より入力される．パルス幅 2 μs以上でTTLレベル（LOW TRUE）である．



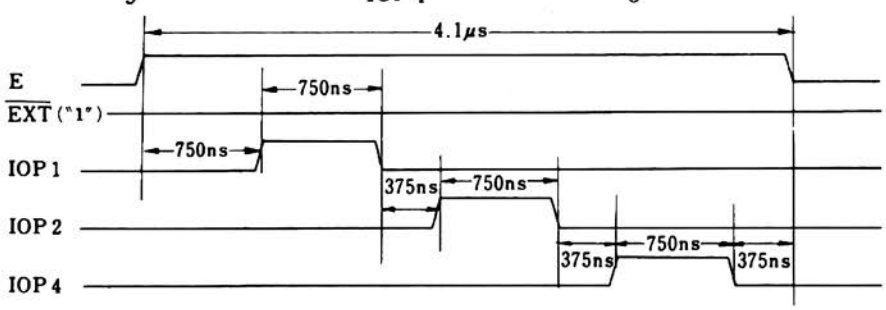
10図　AUTO STARTタイムチャート

ASTパルスによって，PUはRUN状態に入る．I/O部ではASTによってASTINT信号が作られ，これによってINT（疑似INT）信号がPUに送られ，同時にINTJ（5000＝JMP 0）がIRに入力される。INT時にはIR入力信号はINTJ側が選択されIRに5000（JMP 0）がセットされる．

(8) **IOT（IOPパルス）**

IOT命令（コード 6）はPUと端末装置間のデータ転送およびコントロール等を行うが，IOT命令の下 3 ビットに対応してIOPパルスが発生される．

IOT命令ビット	ＩＯＰパルス	発生順序
11	IOP 1	1
10	IOP 2	2
9	IOP 4	3



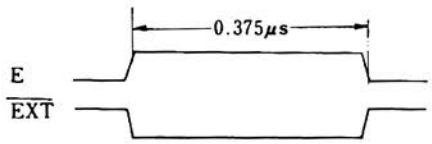
11図　LモードIOTサイクルタイムチャート

555H計算機ではＩＯＴの実行時間を２通りに選択できる.

Hモード　0.375μs；IOPパルス発生せず；$\overline{EXT}=0$

Lモード　4.1μs　；IOP１．２．４発生；$\overline{EXT}=1$

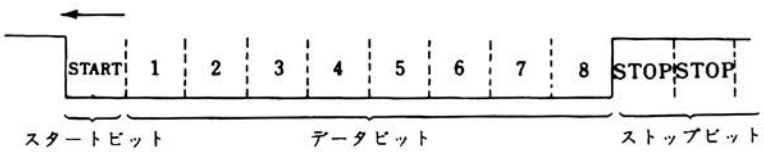
TTY, PTR, PTPコントロール, インタラプトコントロール, フィールドセレクト等の555H内部で使用されるIOT命令はすべてHモードである.



12図　HモードIOTサイクルタイムチャート

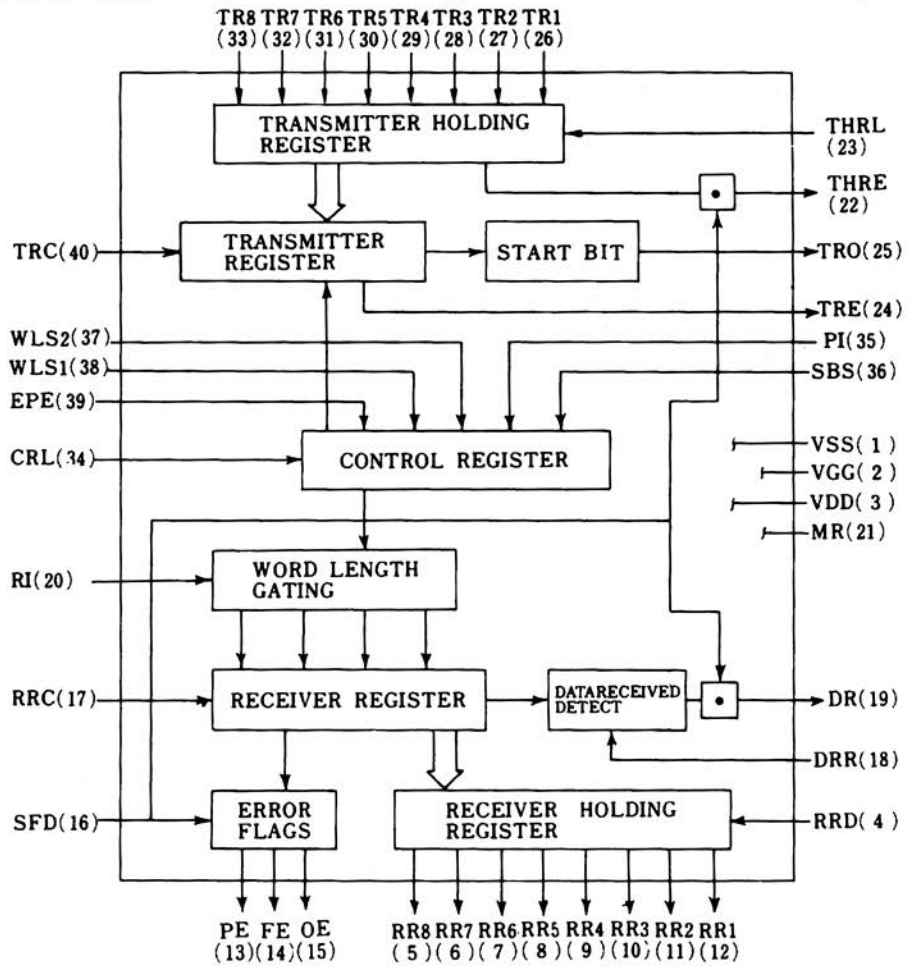
(9) **TTY CONTROL**

PU（並列データ）とTTY（直列データ）とのデータ転送をコントロールする直列，並列変換回路である.



13図　TTYデータ

直列，並列変換はMOS，LSI（TR1402）で行われる．TR1402Aの仕様は下記の通りである．



14図　TR1402A(ASYNCHRONOUS RECEIVER/TRANSMITTER)
ブロック図（　）内番号はPIN番号を示す．

PIN	NAME	SYMBOL	FUNCTION
1	Vss POWER SUPPLY	Vss	+5V
2	V_{GG} POWER SUPPLY	V_{GG}	−12V
3	V_{DD} POWER SUPPLY	V_{DD}	GND
4	RECEIVER REGISTER DISCONNECT	RRD	RRD端子が，HIGH(3.5〜5V）の時，RECEIVER HOLDING REGISTERの出力とRR^{8}〜RR^{1}出力とは切離される．
5〜12	RECEIVER HOLDING REGISTER DATA	RR_8〜RR_1	RRD端子が，LOW(0〜0.8V）の時，RECEIVER HOLDING REGISTER

			の内容は，パラレルに，RR_8，RR_1端子に出力される．RR_1が，LSBである．
13	PARITY ERROR	PE	STATUS FLAGS DISCONNECT端子(16ピン)がLOW の時，PARITY CHECK回路の出力が，PEに出力される．PEが，HIGH の時，PARITY ERRORである．
14	FRAMING ERROR	FE	STATUS FLAGS DISCONNECT端子がLOWの時，STOP BIT CHECK回路の出力が，FEに出力される．FEがHIGHの時，入力されたデータのストップビット数に誤りがあったことを示す．
15	OVERRUN ERROR	OE	STATUS FLAGS DISCONNECT端子がLOWの時，DATA RECEIVED回路の出力がOEに出力されOEがHIGHの時，前のデータがRECEIVER HOLDING REGISTERにあってまた，PUに読み込まれていないうちに，次のデータがRECEIVER HOLDING REGISTER に書き込まれたことを示す．
16	STATUS FLAGS DISCONNECT	SFD	SFD端子が，HIGHの時，PE, FE, DE, DR, THRE出力は，禁止される．
17	RECEIVER REGISTER CLOCK	RRC	データ転送速度の，16倍の周波数のクロックを入力する．
18	DATA RECEIVED REST	DRR	DRR端子が，LOWの時，DR LINEは，リセットされる．
19	DATA RECEIVED	DR	データがすべて読み込まれ，RECEIVER REGISTERに転送完了すると，DR端子は，HIGHになる．

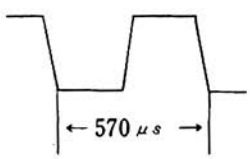
20	RECEIVER INPUT	RI	直列入力データは，RI端子よりRECEIVER REGISTERに読み込まれる．データがないとき，RI端子は，HIGH であること．
21	MASTER RESET	MR	MR端子が，HIGHになるとロッジクはすべてリセットされる．直列データ出力端子(TRC)は，HIGHになる．
22	TRANSMITTER HOLDING REGISTER EMPTY	$\overline{\text{THRE}}$	TRANSMITTES HOLDING REGISTERのデータが，TRANSMITTER REGISTERに転送され，次の新しいデータが受け入れ可能状態になると，THRE端子はHIGH となる．
23	TRANSMITTER HOLDING REGISTER LOAD	$\overline{\text{THRL}}$	THRL 端子が，LOWになると，データはTRANSMITTER HOLDING REGISTERに入る．LOWから HIGH に変るとTRANSMITTER REGISTERに移される．ただし，データ転送中の場合，それが完了するまで待たされる．
24	TRANSMITTER REGISTER EMPTY	$\overline{\text{TRE}}$	TRANSMITTER REGISTERの内容がすべて，外部に転送されると，TRE端子は，HIGHになる．次の転送が開始されるまでHIGHのままである．
25	TRANSMITTER REGISTER OUT PUT	TRO	TRANSMITTER REGISTERの内容（STARTビット，DATAビット，STOPビット）は，TRO端子より出力される．データがない時，TRO端子はHIGHであるHIGH→LOWでSTARTを意味する．
26～33	TRANSMITTER REGISTER DATA INPUT	TR_1-TR_8	並列の8ビットのデータは，TR_1～TR_8端子より，TRANSMITTER HOLD-

			ING REGISTERに入力される.
34	CONTROL REGISTER LOAD	CRL	CRL端子が, HIGHになると, CONTROL ビット (WLS_1, WLS_2, EPE PI, SBS)がCONTROL REGISTERにセットされる.
35	PARITY INHIBIT	PI	PIがHIGHの時, PARITY GENERATION, VERIFI 共に禁止されてる.
36	STOP BITS SELECT	SBS	SBSがHIGHの時, STOP ビットは2LOWの時, 1が選ばれる.
37～38	WORD LENGTH SELECT	WLS_2–WLS_1	1ワードのビット数を選択する.

WLS_2 WLS_1 WORD LENGTH
LOW LOW 5 bit
LOW HIGH 6 bit
HIGH LOW 7 bit
HIGH HIGH 8 bit

39	EVEN PARITY ENABLE	EPE	EPEが HIGH の時, 偶数 PARITY, LOWの時, 奇数 PARITY
40	TRANSMITTER REGISTER CLOCK	TRC	データ転送速度の, 16倍の周波数のクロックを必要とする.

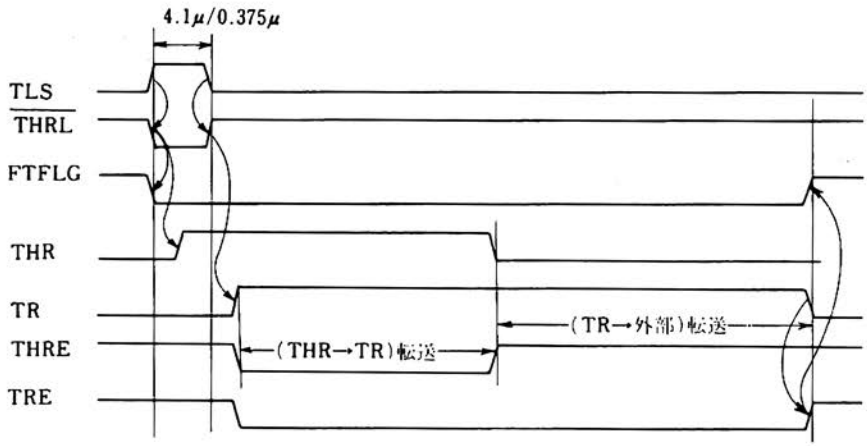
クロック周波数は, 転速送度の16倍である.
10キャラクタ／秒＝10×11ビット／秒 (TTYのデータ転送速度)
10×11×16＝1.76KHz (570 μs)



となる.

① PUからTTYへのデータ入力

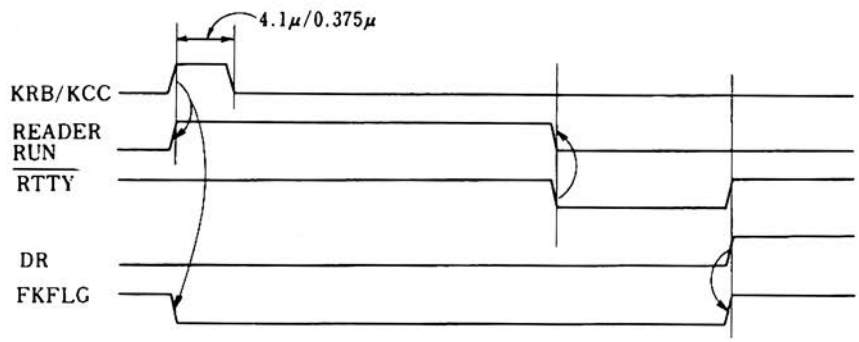
AC 4 ～11ビットにデータを置いて，TLS命令を出すと，TLSの前縁で，TRANSMITTER HOLDING REGISTER（THR）にデータが書き込まれる．同時にパンチャーフラグ（FTFLG）はクリヤされる．TLSの後縁でTHRからTRANSMITTER REGISTER（TR）に転送が開始される．転送が完了すると，THREがHIGHとなってFTFLGをセットし，PUに１ワードのデータ転送完了を知らせる．



15図　TTYへのデータ転送タイムチャート

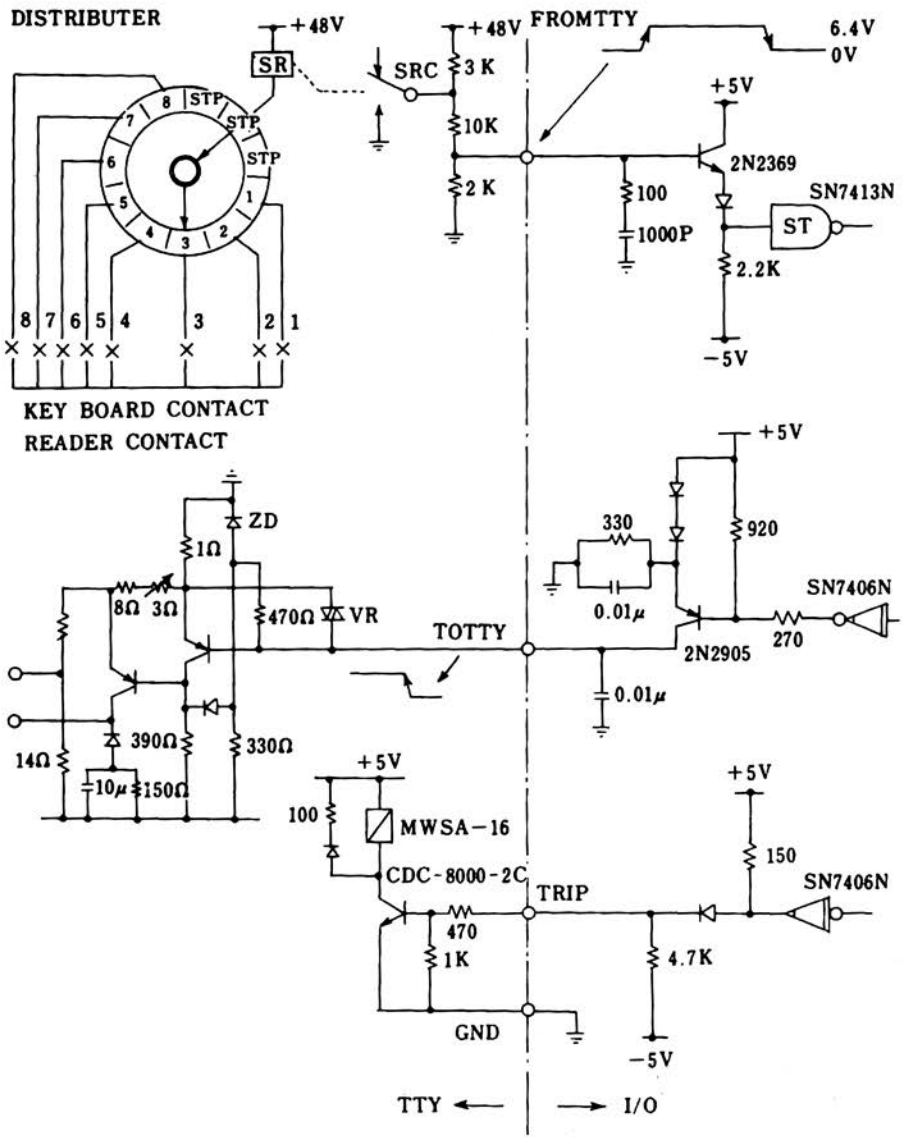
② TTYからPUへのデータ転送

KRBまたはKCC命令の前縁で，READER RUN FFがセットされ，TTYのTRIPMAGNETが駆動される．同時にキーボードフラグ（FKFLG）がクリヤされる．TTYのデータ出力（RTTY）がLOWになると（STARTビットが入力されると）READER RUN FFはクリヤされる．データはRECEIVER REGISTERに直列に読み込まれ，さらにRECEIVER HOLDING REGISTERに転送が完了すると，DR信号がHIGHになってキーボードフラグがセットされ，PUに１ワード読み込み完了を知らせる．



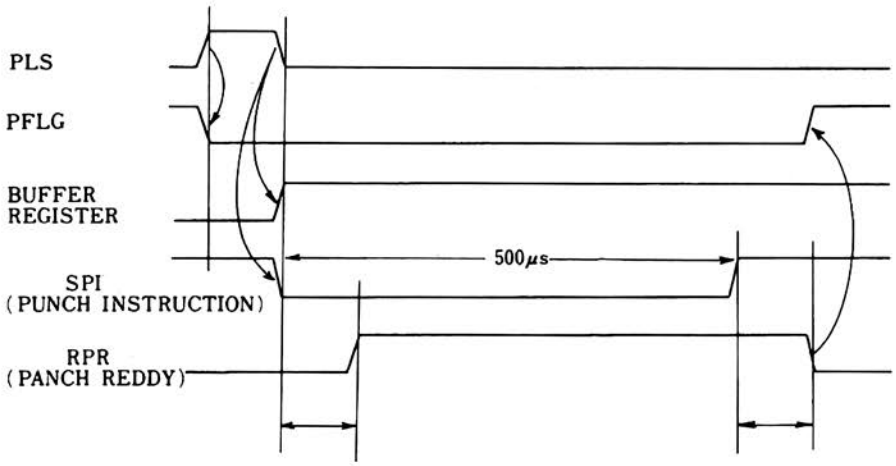
16図　TTYからのデータ転送タイムチャート

③ TTYとI/Oのインタフェース



17図 TTYインタフェース

⑽PTP CONTROL



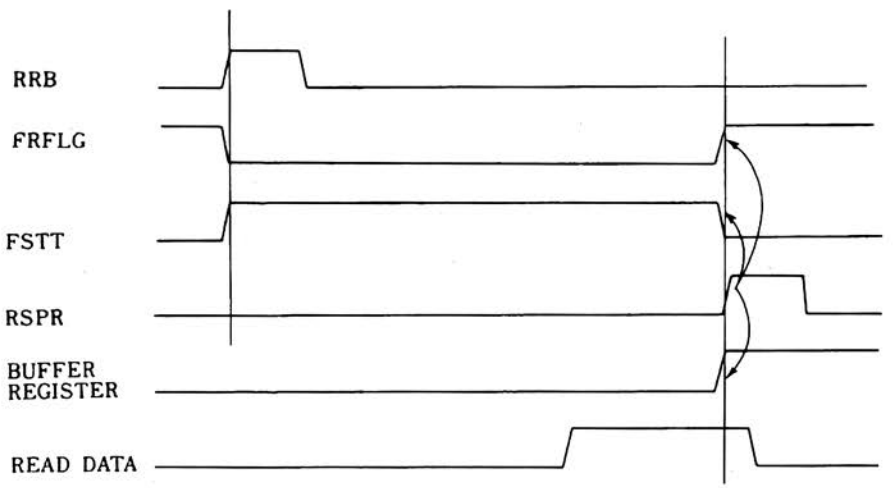
18図　PTPデータパンチタイムチャート

AC 4 ～11にデータを置いてPLS命令を出すと，PLSの前縁でパンチャーフラグ（PFLG）がクリヤされ，後縁で，ACデータはパンチャーのバッファレジスターに書き込まれる．同時に，MMVが駆動され，パンチャーにPI（PUNCH INSTRUCTION）信号が送出される．

パンチが終了すると，PR（PUNCH READY）信号が送り返されてくる．

PRの後縁で，PFLGをセットし，１ワードパンチ終了を，PUに知らせる．

⑾PTR CONTEOL



19図　PTRデータリードタイムチャート

RFC命令の前縁でリーダフラグ（FRFLG）がクリヤされ，リーダスタートF.F（FSTT）がセットされる．FSTTが１の間，リーダはRUN状態となって，データと共にスプロケット（RSPR）信号がI/Oに送られてくる．

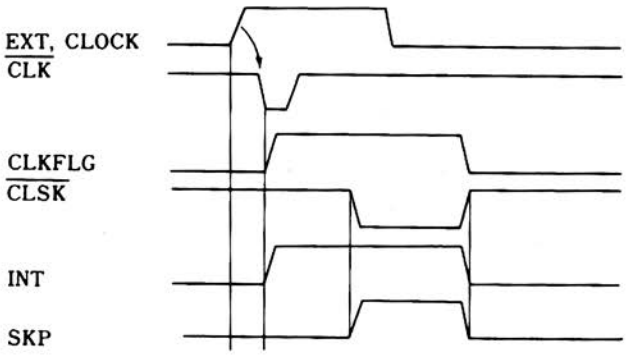
RSPRの前縁で，データをバッファレジスタに書き込みと同時にFRELGをセットして１ワードデータリード完了を，PUに知らせる．

(12)REAL TIME CLOCK

SUPERVISOR CALL

① REAL TIME CLOCK

リアルタイムクロックは外部より入力される.



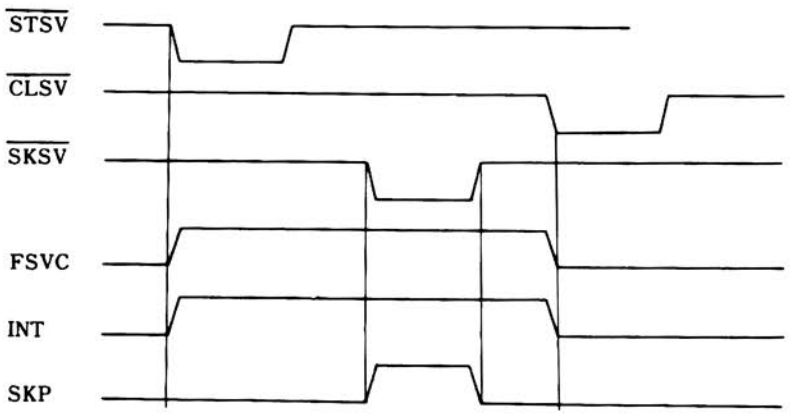
20図　リアルタイムクロックタイムチャート

外部クロックの前縁（立上り）が微分されて，CLKパルスが作られる．CLKでCLKFLGがセットされて，INTRQSTがINT回路に送られる．

ただし，CLEI命令によって，CLOCKINTがENABLEになっていることが必要である．CLDI命令でDISABLEとなる．

② SUPERVISOR CALL

プログラムによる割込みである.



21図　スーパーバイザーコールタイムチャート

③ I/Oインターフェース

外部端末装置と，555Hとのデータ転送（AC経由）に関する信号は以下の10種類である.

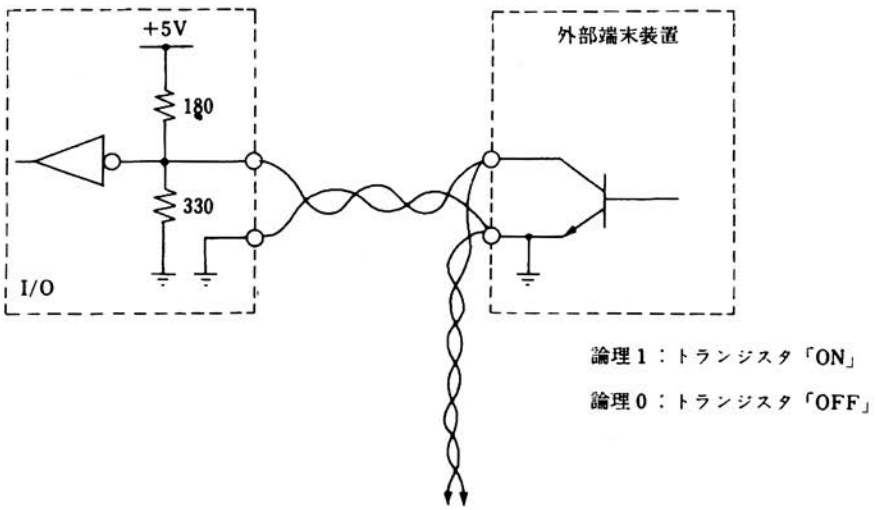
〔1〕 5 5 5H→外部装置

1.1 BAC	ACCUMURATER出力	12本
1.2 BMB	MEMORY BUFFER REGISTER出力	12本
1.3 BE	BMBの同期信号	1本
1.4 BIOP1,2,4	IOPパルス	3本
1.5 BINITIALIZE	●外部入 出力装置へのリセット信号	1本

〔2〕外部装置→５５５Ｈ

2.1 AC BUS	ACCUMURATERへの出力	12本
2.2 ACLOAD BUS	ACCUMURATERへのデータ書き込み信号	1本
2.3 INT BUS	INTERRUPT入力	2本
2.4 SKP BUS	SKIP入力	1本
2.5 EXTIOT BUS	IOT実行速度の切換	1本
2.6 AC CLR	ACクリヤ	1本

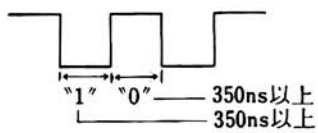
(1) AC INPUT, ACLOAD, AC CLR, INT, SKP, EXTIOT信号入力規格



22図　信号入力インタフェース

(A) I/O側規格

①	入力形式	IC（終端抵抗付）
②	流出電流	約 27mA
③	入力スレッシュホールドレベル	約 0.8V
④	応答時間	約 50ns
⑤	パルス入力条件	



(B) 外部端末装置側規格

①	出力形式	接地形トランジスタスイッチ
②	電流容量	50mA以上
③	耐圧	15V以上
④	飽和電圧	1V以下

(C) 接続ケーブル

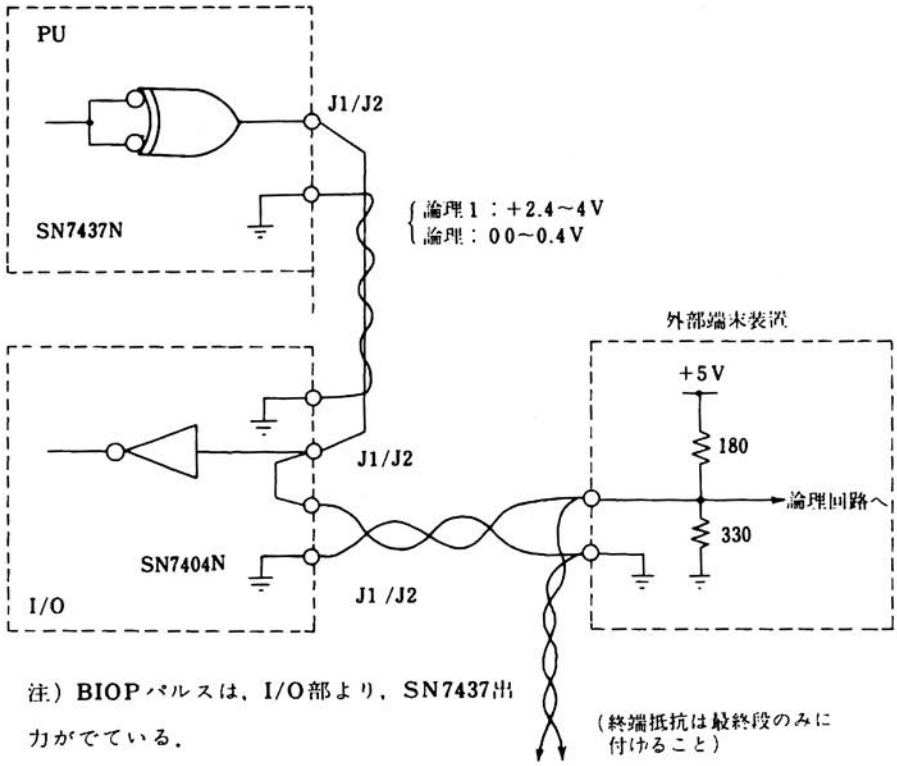
① 浮遊容量　100PH/m以下であること.

② ２ケヨリ線で構成され，多芯一括シールドケーブルを使用することが望ましい.

③ ケーブルの特性インピーダンスは，130Ωとする.

④ ケーブルの最大長は，５mとする.

(2) BAC, BMB, BE, BIOP, BINITIALIZE信号出力規格



23図　信号出力インタフェース

(A) I/O側規格

① 出力形式　　IC (SN7437N QUADRUPLE 2-INPUT POSITIVE
② 　　NAND BUFFERS
③ 最大印加電圧　5.25V
④ 最大負荷電流　46mA
④ 飽和電圧　　0.4V（最大負荷電流のとき）

(B) 外部端末装置側規格

① 入力形式　　IC（終端抵抗付）
② 入力スレッシュホールドレベル　0.8V＋ケーブル抵抗による電圧降下分

(C) 接続ケーブル

① 浮遊容量　100PF/m以下であること.
② 2ヶヨリ線で構成され，多芯一括シールドケーブルを使用することが望ましい.
③ ケーブルの特性インピーダンスは，130Ωとする.
④ ケーブルの最大長は5mとする.

6. 5 5 5 H　入出力コネクター一覧表

(1) PU

J1A

0		GND
1		BAC00
2		GND
3		BAC01
4		GND
5		BAC02
6		GND
7		BAC03
8		GND
9		BAC04
10		GND
11		BAC05
12		GND
13		BAC06
14		GND
15		BAC07
16		GND
17		BAC08
18		GND
19		BAC09
20		GND
21		BAC10
22		GND
23		BAC11
24		GND
25		BRUN
26		GND
27		BINIT
28		GND

2

J2A

0		GND
1		BMB00
2		GND
3		BMB01
4		GND
5		BMB02
6		GND
7		BMB03
8		GND
9		BMB04
10		GND
11		BMB05
12		GND
13		BMB06
14		GND
15		BMB07
16		GND
17		BMB08
18		GND
19		BMB09
20		GND
21		BMB10
22		GND
23		BMB11
24		GND
25		
26		GND
27		BE
29		GND

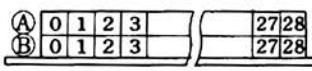
J1B

0		GND
1		
2		GND
3		
4		GND
5		
6		GND
7		
8		GND
9		
10		GND
11		
12		GND
13		
14		GND
15		
16		GND
17		
18		GND
19		
20		GND
21		
22		GND
23		
24		GND
25		
26		GND
27		
28		GND

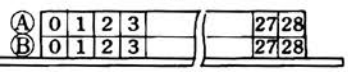
J2B

1		GND
2		
3		GND
4		
5		GND
6		
7		GND
8		
9		GND
10		
11		GND
12		
13		GND
14		
15		GND
16		
17		GND
18		
19		GND
20		
21		GND
22		
23		GND
24		
25		GND
26		
27		GND
28		
28		GND

J1



J2



(2) CH

J1A

0		GND
1		RQST#0
2		GND
3		RQST#1
4		GND
5		RQST#2
6		GND
7		RQST#3
8		GND
9		RQST#4
10		GND
11		
12		GND
13		ES. (0)
14		GND
15		ES. (1)
16		GND
17		ES. (2)
18		GND
19		ES. (3)
20		GND
21		ES. (4)
22		GND
23		
24		GND
25		BTS1
26		GND
27		BTS3
28		GND

J2A

0		GND
1		BAKN
2		GND
3		BWCOF
4		GND
5		BBRK
6		GND
7		BDAVL
8		GND
9		BDRCV
10		GND
11		TRANSF DIR
12		GND
13		MBNICR
14		GND
15		CAINCR
16		GND
17		CYC SEL
18		GND
19		
20		GND
21		
22		GND
23		DB12
24		GND
25		DB13
26		GND
27		DB14
28		GND

J1B

0		GND
1		DB00
2		GND
3		DB01
4		GND
5		DB02
6		GND
7		DB03
8		GND
9		DB04
10		GND
11		DB05
12		GND
13		DB06
14		GND
15		DB07
16		GND
17		DB08
18		GND
19		DB09
20		GND
21		DB10
22		GND
23		DB11
24		GND
25		
26		GND
27		
28		GND

J2B

0		GND
1		
2		GND
3		
4		GND
5		
6		GND
7		
8		GND
9		
10		GND
11		
12		GND
13		
14		GND
15		
16		GND
17		
18		GND
19		
20		GND
21		
22		GND
23		
24		GND
25		
26		GND
27		
28		GND

(3) I/O

J1A

0		GND
1		BAC00
2		GND
3		BAC01
4		GND
5		BAC02
6		GND
7		BAC03
8		GND
9		BAC04
10		GND
11		BAC05
12		GND
13		BAC06
14		GND
15		BAC07
16		GND
17		BAC08
18		GND
19		BAC09
20		GND
21		BAC10
22		GND
23		BAC11
24		GND
25		$\overline{\text{BRUN}}$
26		GND
27		BINIT
28		GND

J2A

0		GND
1		BMB00
2		GND
3		BMB01
4		GND
5		BMB02
6		GND
7		BMB03
8		GND
9		BMB04
10		GND
11		BMB05
12		GND
13		BMB06
14		GND
15		BMB07
16		GND
17		BMB08
18		GND
19		BMB09
20		GND
21		BMB10
22		GND
23		BMB11
24		GND
25		
26		GND
27		BE
28		GND

CN1 (CS420)

A		PD1
C		GND
B		PD2
D		GND
E		PD3
H		GND
F		PD4
J		GND
K		PD5
M		GND
L		PD6
N		GND
P		PD7
S		GND
R		PD8
T		GND
U		PD9
W		GND
V		SD
X		GND
Y		PI
d		GND
z		PR
b		GND
c		TL
d		GND

CN2 (CS408)

A		$\overline{\text{SPR}}$
C		GND
B		TAPE END
D		GND
E		a1
H		GND
F		a2
J		GND
K		a3
M		GND
L		a4
N		GND
P		a5
S		GND
R		a6
T		GND
U		a7
W		GND
V		a8
X		GND
Y		$\overline{\text{START}}$
d		GND
z		$\overline{\text{STOP}}$
b		GND
c		MNUAL
d		GND

CN3 (CS100)

A		FROM TTY
C		GND
B		
D		
E		
H		
F		+5V POWER
J		GND
K		TRP
M		GND
L		
N		
P		
S		
R		
T		
U		TO TTY
W		GND
V		
X		
Y		
d		
z		
b		
c		$\overline{\text{TTY ON}}$
d		GND

J1B

0		GND
1		$\overline{\text{AC00 BUS}}$
2		GND
3		$\overline{\text{AC01 BUS}}$
4		GND
5		$\overline{\text{AC02 BUS}}$
6		GND
7		$\overline{\text{AC03 BUS}}$
8		GND
9		$\overline{\text{AC04 BUS}}$
10		GND
11		$\overline{\text{AC05 BUS}}$
12		GND
13		$\overline{\text{AC06 BUS}}$
14		GND
15		$\overline{\text{AC07 BUS}}$
16		GND
17		$\overline{\text{AC08 BUS}}$
18		GND
19		$\overline{\text{AC09 BUS}}$
20		GND
21		$\overline{\text{AC10 BUS}}$
22		GND
23		$\overline{\text{AC11 BUS}}$
24		GND
25		$\overline{\text{ACCLR BUS}}$
26		GND
27		$\overline{\text{AC LOAD BUS}}$
28		GND

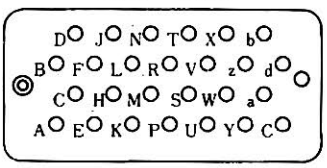
J2B

0		GND
1		
2		GND
3		
4		GND
5		
6		GND
7		
8		GND
9		
10		GND
11		
12		GND
13		BIOP4
14		GND
15		BIOP2
16		GND
17		BIOP1
18		GND
19		$\overline{\text{EXTIOT BUS}}$
20		GND
21		$\overline{\text{SKP BUS}}$
22		GND
23		$\overline{\text{INTRQST BU}}$
24		GND
25		$\overline{\text{INTRQST1 BUS}}$
26		GND
27		
28		GND

CN2

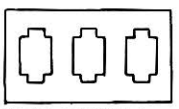
1		EXT. CLK
2		GND
3		

CN1 CN3



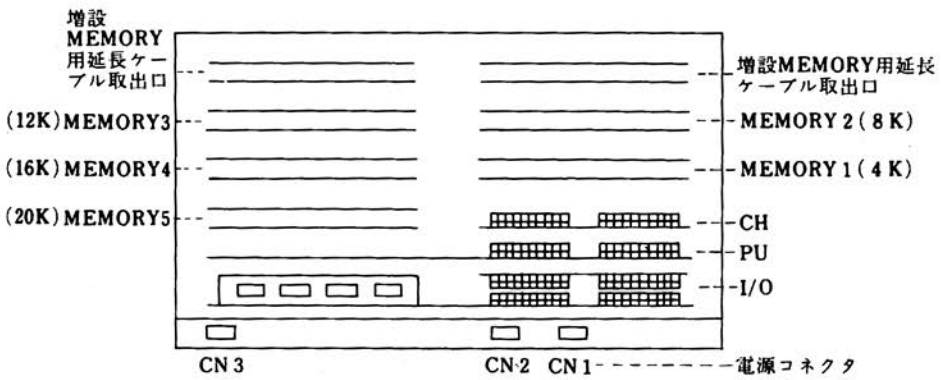
AMP-200512-2(筐体)
AMP-200513-2(ケーブル)

CN4

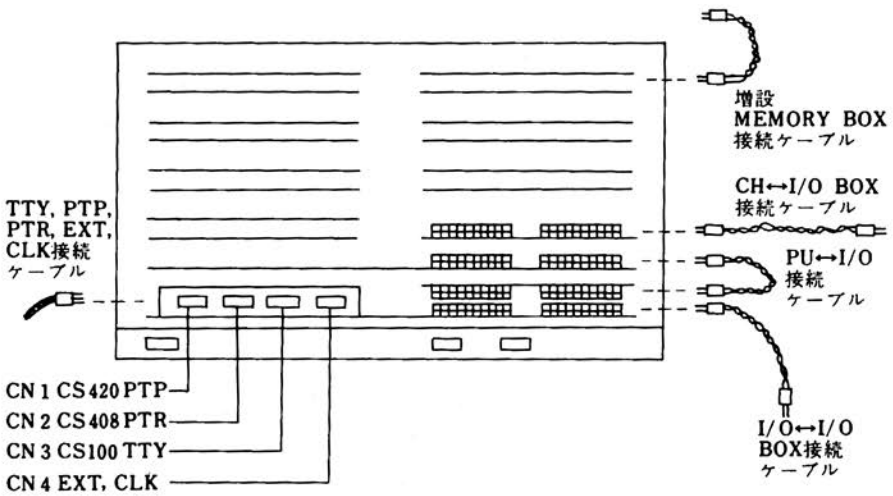


1-480004-0(筐体)
1-480305-0(ケーブル)

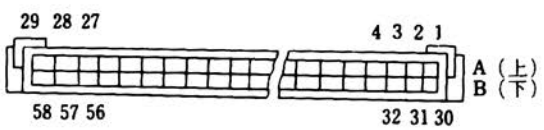
ケーブル接続



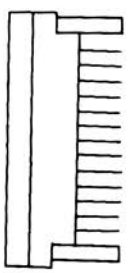
555H　背面図



555H　ケーブル接続



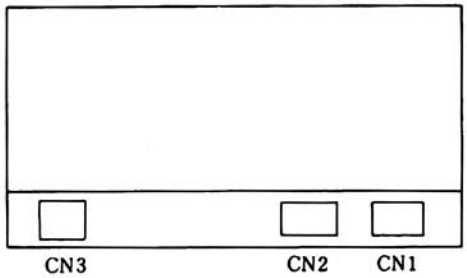
A-MP 1　86671－3　(オス)



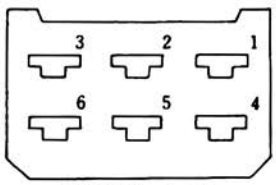
上面 (オス)

J 1, J 2 (58PIN) コネクタ

7. 電源コネクタ

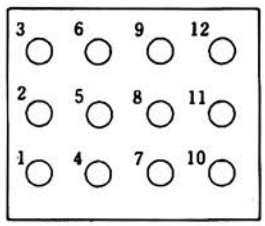


555H　電源コネクタ



(480194　オス)
(480195　メス)

CN1，CN2 ピン配置図



(480278　オス)
(480275　メス)

CN3 ピン配置図

ピン番号	CN1	ピン番号	CN2
1	+5V	1	+22V
2	+5V	2	+12V
3	−5V	3	
4	LGND	4	MGND1
5	LGND	5	MGND2
6	−12V	6	

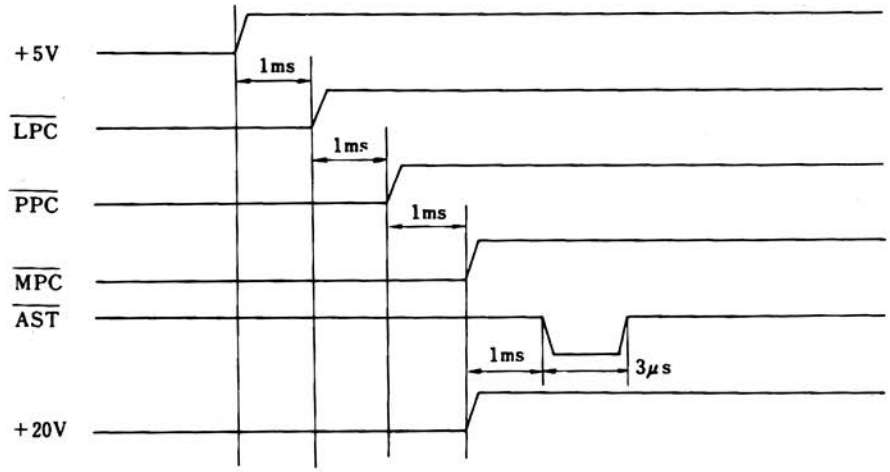
注）LGND：ロジックランド　　MGND：メモリーグランド

ピン番号	CN3	ピン番号	CN3	ピン番号	CN3
1	$\overline{\text{PPC}}$	5	$\overline{\text{AST}}$	9	EARTH
2	$\overline{\text{LPC}}$	6	LGND	10	POWER ON
3	$\overline{\text{LPCD}}$	7	POWER ALM	11	POWER SW
4	$\overline{\text{MPC}}$	8		12	POWER SW

(C-COMMON)

8．電源投入時と切断時におけるシーケンス

8－1　電源投入時におけるシーケンス



コンソールパネル，パワースイッチオンでAC電源が投入されると，＋5V，－12V，－5Vがオンになる．約1ms後，$\overline{\text{LPC}}$がHIGHに，さらに1ms後に，$\overline{\text{PPC}}$がHIGHになる．$\overline{\text{PPC}}$が，LOWの間（電源オンの後，約2msの間）計算機のロジックはリセット状態に置かれる．

$\overline{\text{PPC}}$がHIGHになってからさらに，約1ms後，$\overline{\text{MPC}}$がHIGHになる．

$\overline{\text{MPC}}$がLOWの間，メモリー部のロジックはリセット状態に置かれる．

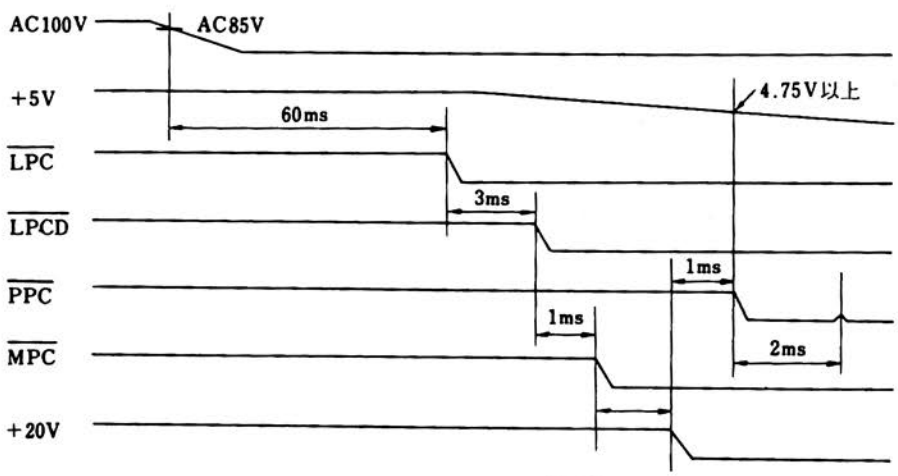
$\overline{\text{MPC}}$がHIGHになると，メモリー電源（+20V）がオンになって，メモリーへのリードライトが可能になる．

ここで始めて計算機は，READY状態になるが，$\overline{\text{MPC}}$がHIGHになってから約1ms後，電源部より計算機に，ASTパルスが入力される．

ただし，電源部のAUTO　STARTスイッチオンの場合である．ASTパルスによって計算機は，RUNフリップフロップがセットされて，RUN状態になり，続いて，JMP 0が実行される．

AUTO STARTスイッチオフの場合，計算機はREADY状態で待機する．

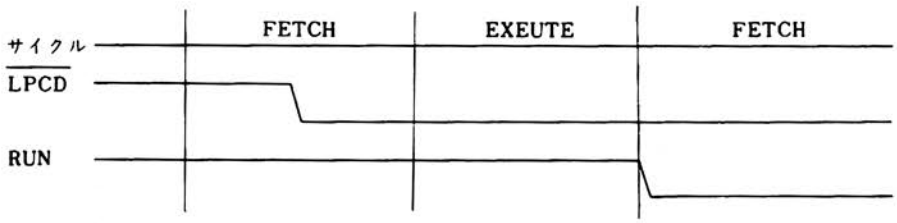
8－2 電源切断時におけるシーケンス



AC 100V電源，85V以下に落ちた時点から，60ms後に，$\overline{\text{LPC}}$がLOWになる．

ただし，60ms以内に，AC 100V電源が85V以上に復帰した場合LPCはLOWにならない．$\overline{\text{LPC}}$がLOWになると，INTRQST 1 信号が計算機に入力される．

ここで，プログラムがインタラプトを処理していてかつ停電処理を含むならば問題はないが，インタラプト処理をしていない場合，計算機は，RUNのまま$\overline{\text{PPC}}$によってリセットすることになるので，少なくとも$\overline{\text{MPC}}$がLOWになる前に，RUNを止めることが必要である．そこで，LPCがLOWになってから，約3 ms後に，$\overline{\text{LPCD}}$がLOWになって，そのときの，実行サイクルの終了時に，RUNを止める．従って，$\overline{\text{LPC}}$によって，インタラプトがかかってから，停電処理を終了するまで約3 ms以内に行うように，プログラムを組むことが必要である．



$\overline{\text{LPCD}}$によって計算機がSTPしてから約1 ms後，$\overline{\text{MPC}}$がLOWになってメモリー部をリセットし，続いて，＋20V電源がオフになってから約1 ms後に，PPCがLOWになって，計算機をリセットする．

$\overline{\text{PPC}}$がLOWになるまで，＋5 Vは95%以上（4.75V以上）を保持していることが必要である．

$\overline{\text{PPC}}$がLOWになってから，約2 ms後に若干のノイズが出る可能性がある．

＋5 Vが，4.75V以下に落ちることによて，電源部シーケンス回路が正が正常動作範囲からはずれるためである．ただし，2 ms以前にノイズが出ることはない．

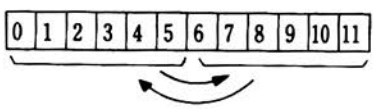
9. 命令コード表

(1) メモリー参照命令

記　号	8進コード	動　作	実行時間 直接	間接	オートインデックス
AND Y	0	(Y)と(AC)との間の2の補数演算による論理積をとる．結果は，ACに残る．	2.4μs	3.4μs	3.6μs
TAD Y	1	(Y) との間の2の補数演算による論理和をとる．結果は，ＡＣに残る．	2.4 μs	3.4 μs	3.6 μs
ISZ Y	2	(Y)を2の補数演算で＋1する．結果が0であれば次の命令をスキップする．	2.5 μs	3.5 μs	3.7 μs
DCA Y	3	(ＡＣ)の内容をＹ番地に書き込む．実行後ＡＣの内容はクリヤされる．	2.4 μs	3.4 μs	3.6 μs
JMS Y	4	PC＋1をYに書き込み，Y＋1にジャンプする．	2.4 μs	3.4 μs	3.6 μs
JMP Y	5	Yにジャンプする．	1.7 μs	2.4 μs	2.6 μs

(2) グループ1マクロインストラクション（実行時間 1.6 μs）

記　号	8進コード	動　作
NPP	7000	無効命令1.6 μsの遅延を行う．
IAC	7001	（ＡＣ）を2の補数演算で＋1する．
RAL	7004	ACとLNKを左に1つシフトする．
PTL	7006	ACとLNKを左に2つシフトする．
RAR	7010	ACとLNKを右に1つシフトする．
RTR	7112	ACとLNKを右に2つシフトする．
BSW	7002	バイトスワップ．ACの(0～5)と(6～11)を入れかえる．



CML	7020	LNKの補数をとる．
CMA	7040	(AC)の1の補数をとる．
CIA	7041	(AC) の2の補数をとる．
CLL	7100	LNKをクリヤする．

CLL RAL	7104	LNKをクリヤし，ACとLNKを左に1つシフトする．
CLL RTL	7106	LNKをクリヤし，ACとLNKを左に2つシフトする．
CLL RAR	7110	LNKをクリヤし，ACとLNKを左に1つシフトする．
CLL RTR	7112	LNKをクリヤし，ACとLNKを右に2つシフトする．
STL	7120	LNKを1セットする．
CLA	7200	ACをクリヤする．
CLA IAC	7201	ACを1にセットする．
GLK	7204	(LNK)を，AC11にロードする．
STA	7240	ACを1にセットする．
CLA CLL	7300	ACとLNKをクリヤする．

注）命令の優先順位

1 CLA, CLL

2 CMA, CML

3 IAC

4 RAR, RAL, RTR, RTL, BSW

(3) グループ2 マイクロインストラクション（実行時間 1.6 μs）

記 号	8進コード	動 作
HLT	7402	プログラムの実行を停止する．
OSR	7404	スイッチレジスタと(AC)との論理和をとり結果はACに残る．
SKP	7410	無条件スキップ，次の命令を無条件でスキップする．
SNL	7420	LNK≠0の時，次の命令をスキップする．
SZL	7430	LNK＝0の時，次の命令をスキップする．
SZA	7440	(AC)＝0の時，次の命令をスキップする．
SNA	7450	(AC)≠0の時，次の命令をスキップする．
SZA SNL	7460	(AC)＝0またはLNK≠0の時，次の命令をスキップする．
SNA SZL	7470	(AC)≠0でかつLNK＝0の時，次の命令をスキップする．
SMA	7500	(AC)＜0の時，次の命令をスキップする．
SPA	7510	(AC)≧0の時，次の命令をスキップする．
SMA SNL	7520	(AC)＜0またはLNK≠0の時，次の命令をスキップする．
SPA SZL	7530	(AC)≧0でかつLNK＝0の時，次の命令をスキップする．
SMA SZA	7540	(AC)≦0の時，次の命令をスキップする．
SPA SNA	7550	(AC)＞0の時，次の命令をスキップする．
(CLA)	7600	ACをクリヤする．

LAS	7604	ACの内容をクリアし，スイッチレジスタの内容をACにロードする．（CLA, OSR）
SZA CLA	7640	(AC)＝0の時，次の命令をスキップし，ACをクリヤする．
SNA CLA	7650	(AC)≠0の時，次の命令をスキップし，ACをクリヤする．
SMA CLA	7700	(AC)<0の時，次の命令をスキップし，ACをクリヤする．
SPA CLA	7710	(AC)≧0の時，次の命令をスキップし，ACをクリヤする．

注）命令の優先順位

1 (BIT 8 = 0)：SMA, SZA, SNL

2 (BIT 8 = 1)：SPA, SNA, SZL

3 CLA

4 OSR, HLT

(4) グループ3 マイクロインストラクション（実行時間 2.7 μs）

記　号	8進コード	動　作
(NOP)	7401	無効動作．2.7 μs遅延を行う．
MQL	7421	(AC)をMQにロードする．ACはクリヤする．
MQA	7501	(MQ)をACにロードする．
SWP	7521	(AC)と(MQ)を入れかえる．
(CLA)	7601	ACをクリヤする．
CLA MQL	7621	ACとMQをクリヤする．
CLA MQA	7701	ACをクリヤし，その後(MQ)をACにロードする．
CLA SWP	7721	ACをクリヤし，(MQ)をACにロードする．MQはクリヤされる．

注）命令の優先順位

1 CLA

2 MQA, MQL

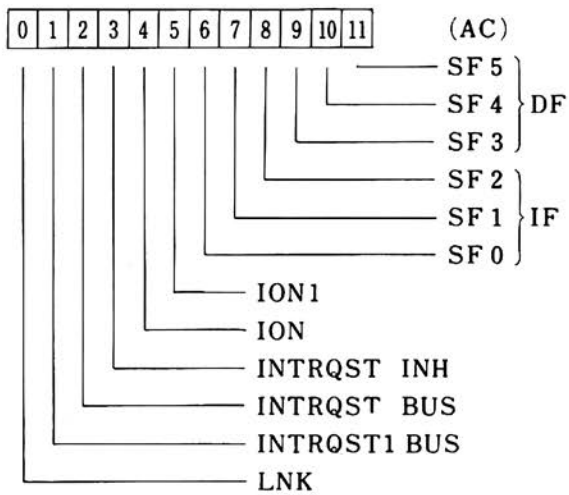
(5) JAP インストラクション（実行時間 2.7 μs）

記　号	8進コード	動　作
(NOP)	6770	無効動作．2.7 μs遅延を行う．
MPQ	6771	(PC)をMQにロードする．
MAP	6772	(AC)をPCにロードする．
MAPQ	6773	(AC)をPCにロードし，(PC)をMQにロードする．
(MQA)	6774	(MQ)をACにロードする．
MPQA	6775	(PC)をMQに，(MQ)をACにロードする．

MQAP	6776	(MQ) をACに，(AC) をPCにロードする．
RAPQ	6777	(MQ) をACに，(AC)をPCに，(PC)をMQにロードする．

(6) プログラムインタラプトに関する命令（実行時間 1.6 μs）

記　号	8進コード	動　作
SKON	6000	インタラプトまたはインタラプト1がイネイブルの時次の命令をスキップする．実行後インタラプトオフとなる．
ION	6001	インタラプトイネイブルにする．プログラムインタラプタに応じる．
IOF	6002	インタラプトオフにする．プログラムインタラプトを無視する．
SRQ	6003	インタラプトリクエストまたはインタラプトリクエスト1があれば，次の命令をスキップする．
GTF	6004	マシーンステートを，ACに読み込む．



RTF	6005	GTFの逆命令，命令実行後，インタライトイネイブルとなる．
CAF	6007	ACとLNKをクリヤする．
ION 1	6106	インタラプト1イネイブルにする．プログラムインタラプト1に応じる．
IOF 1	6107	インタラプト1オフにする．プログラムインタラプト1を無視する．

(7) メモリフィールドに関する命令（実行時間 1.6 μs）

記　号	8進コード	動　作
CDF	62N1	データフィールドを，N(N＝0～7)に切換える．

CIF	62N2	インストラクションフィールドを,N(N＝0～7)に切換える.
CDF CIF	62N3	データフィールド，インストラクションフィールドを，Nに切換える.
RDF	6214	データフィールドを，AC(6～8)に読み込む.
RIF	6224	インストラクションフィールドを，AC(6～8)に読み込む.
RIB	6234	インタラプトバッファレジスタのうち，インストラクションフィールドをAC(6~8)に，データフィールドをAC(9～11)に読み込む.
RMF	6244	インタラプトバッファレジスタを，インストラクションフィールドおよびデータフィールドに戻す.

(8) メモリパリティに関する命令（実行時間 1.6 μs）

記　号	8進コード	動　作
SMP	6101	メモリパリティエラーのない時，次の命令をスキップする.

(9) 電源異常に関する命令（実行時間 1.6 μs）

記　号	8進コード	動　作
SPL	6102	$\overline{\text{LPC}}$＝0の時(電源断または瞬停の時)次の命令をスキップする.

(10) 外部クロックに関する命令（実行時間 1.6 μs）

記　号	8進コード	動　作
CLEI	6131	外部クロックによるフラグセットを，可能にする.
GLDI	6132	外部クロックによるフラグセットを，無視する.
CLSK	6133	クロックフラグが1の時，次の命令をスキップする．フラグはクリヤされる.

(11) スーパーバイザコールに関する命令（実行時間 1.6 μs）

記　号	8進コード	動　作
STSV	6103	スーパーバイザーコールフラグを，1にする.
SKSV	6104	スーパーバイザコールフラグが，1の時，次の命令をスキップする.
CLSV	6105	スーパーバイザコールフラグをクリヤする.

(12) TTY キーボードリーダーに関する命令（実行時間 1.6 μs）

記　号	8進コード	動　作
KCF	6030	キーボードフラグをクリヤする.
KSF	6031	キーボードフラグが1の時，次の命令をスキップする.
KCC	6032	キーボードフラグとACをクリヤし,リーダラン指令を出す.
KRS	6034	キーボードバッファの内容を，AC(4～11)に読み込む.

KIE	6035	AC11が1の時，インタラプトイネイブルになる． AC11が0の時，インタラプトディスエイブルになる．
KRB	6036	キーボードバッファを，AC(4～11)に読み込み，フラグをクリヤする．

(13) TTY プリンタ／パンチャに関する命令（実行時間 1.6 μs）

記　号	8進コード	動　作
TFL	6040	プリンタ／パンチャフラグをセットする．
TSF	6041	プリンタ／パンチャフラグが1の時，次の命令をスキップする．
TCF	6042	プリンタ／パンチャフラグをクリヤする．
TPC	6044	AC(4～11)の内容を印刷またはせん孔する．
TSK	6045	プリンタ／パンチャフラグまたはキーボードフラグが1の時次の命令をスキップする．
TLS	6046	AC(4～11)の内容を印刷またはせん孔し，プリンタ／パンチャフラグをクリヤする．

(14) 高速紙テープリーダに関する命令（実行時間 1.6 μs）

記　号	8進コード	動　作
RPE	6010	リーダ／パンチャインタラプイネイブルにする．
RSF	6011	リーダフラグが1の時，次の命令をスキップする．
RRB	6012	リーダバッファの内容を，AC(4～11)に読み込む．
RFC	6014	リーダフラグをクリヤし，リード指令を出す．
RRB RFC	6016	リーダバッファの内容を，AC(4～11)に読み込み，リーダフラグをクリヤして，リード指令を出す．

(15) 高速紙テープパンチャに関する命令（実行時間 1.6 μs）

記　号	8進コード	動　作
PCE	6020	リーダ／パンチャインタラプトディスエイブルにする．
PSF	6021	パンチャフラグが1の時，次の命令をスキップする．
PCF	6022	パンチャフラグをクリヤする．
PPC	6024	AC(4～11)の内容をせん孔する．
PLS	6026	フラグをクリヤし，AC(4～11)の内容をせん孔する．

(16) テレタイプコード

ASCII コード

文字	コード	記号	コード
A	301	6	266
B	302	7	267
C	303	8	270
D	304	9	271
E	305	!	241
F	306	"	242
G	307	#	243
H	310	$	244
I	311	%	245
J	312	&	246
K	313	'	247
L	314	(	250
M	315	)	251
N	316	*	252
O	317	+	253
P	320	,	254
Q	321	—	255
R	322	.	256
S	323	/	257
T	324	:	272
U	325	;	273
V	326	<	274
W	327	=	275
X	330	>	276
Y	331	?	277
Z	332	@	300
0	260	Line Feed	212
1	261	Return	215
2	262	Space	240
3	263	ALT MODE	275
4	264	RUb OUT	277
5	265	ESC	233

ASCII コード

キー操作		記号	コード
Shift	Control	ⓐ…NULI/IDLE	200
	Control	A…SOM	201
	Control	B…EOA	202
	Control	C…EOM	203
	Control	D…EOT	204
	Control	E…WRU	205
	Control	F…RU	206
	Control	G…BELL	207
	Control	H…FE	210
	Control	I…HT(TAB)	211
	Control	J…LF(Line Feed)	212
	Control	K…V TAB	213
	Control	L…FF(FORM)	214
	Control	M…CR(carriage Return)	215
	Control	N…SO	216
	Control	O…SI	217
	Control	P…DC0	220
	Control	Q…DC1(X ON)	221
	Control	R…DC2(TAPE)	222
	Control	S…DC3(X OFF)	223
	Control	T…DC4(NOT TAPE)	224
	Control	U…ERR	225
	Control	V…SYNS	226
	Control	W…LEM	227
	Control	X…S0	230
	Control	Y…S1	231
	Control	Z…S2	232
Shift	Control	K…S3	233
Shift	Control	L…S4	234
Shift	Control	M…S5	235
Shift	Control	N…S6	236
Shift	Control	O…S7	237
Shift	Control	K…［	333
Shift		L…／	334
Shift		M…］	335
Shift		N…↑	336
		O…←	337

(17) RIM LOADER

番地	低速	高速
7756	6032	6014
7757	6031	6011

```
7760    5357    5357
7761    6036    6016
7762    7106    7106
7763    7006    7006
7764    7510    7510
7765    5357    5357
7766    7006    7006
7767    6031    6011
7770    5367    5367
7771    6034    6016
7772    7420    7420
7773    3776    3776
7774    3376    3376
7775    5356    5357
7776    0000    0000
```

(18) **BINARY LOADER**

```
                *7612       0
7612  0000      SWICH       0
7613  0000      MEMTEM      0
7614  0000      CHAR        0
7615  0000      CHKSUM      0
                ORIGIN
                *7620
/EXTRACT ERRORS, FIELD, L/T
7626  0000      BEGG,       0
7627  3212                  DCA SWITCH      /SET SWITCH
7630  4260                  JMS READ        /GET A CHARACTER
7631  1300                  TAD  M376       /TEST FOR 377
7632  7750                  SPA SNA CLA
7633  5237                  JMP. + 4        /NO
7634  2212                  ISZ SWITCH      /YES : COMPLEMENT SWI
7635  7040                  CMA
7636  5227                  JMP BEGG+ 1     /NOT 377
7637  1212                  TAD SWITCH      /IS SWITCH SET
7640  7640                  SZA CLA         /YES, IGNORE
7641  5230                  JMP BEGG+ 2     /NO, TEST FOR CODE
7642  1214                  TAD CHAR        /TYPES
7643  0274                  AND MASK
7644  1341                  TAD M200
7645  7510                  SPA
7646  2226                  ISZ BEGG        /DATA OR ORIGIN
7647  7750                  SPA SNA CLA     /DATA ORIGIN, or L/T
7650  5626                  JMP1 BEGG       /FIELD SETTING
```

```
7651  1214            TAD CHAR
7652  0256            AND FMASK
7653  1257            TAD CHANGE
7654  3213            DCA MEMTEM  /CONTINUE INPUT
7655  5230            JMP BEGG+ 2
7656  0070  FMASK,     70
7657  6201  CHANFE,    CDF
7660  0000  READ,      0
7661  0000            0
7662  6031  LOR       KSF          /WAIT FOR FLAG
7663  5262            JMP. - 1
7664  6036            KRB
7665  3214            DCA CHAR
7666  1214            TAD CHAR
7667  5660            JMPI READ
7670  6011  HIR,       RSF
7671  5270            JMP. - 1
7672  6016            RRB REC F
7673  5265            JMP LOR + 3
7674  0300  MASK      300
        /TRAILER CODE SEEN
7675  4343  BEND,     JMS ASSEMB
7676  7041            CIA
7677  1215            TAD CHKSUM
7700  7402  M376,     HLT
7701  6032  BEGIN,    KCC
7702  6014            RFC
7703  6214            RDF
7704  1257            TAD CHANGE
7705  3213            DCA MEMTEM  /SAVE FIELD INSTRUCTION
7706  7604            CLA OSR
7707  7700            SMA  CLA
7710  1353            TAD HIRI
7711  1352            TAD LORI
7712  3261            DCA READ + 1
7713  4226            JMS BEGG
7714  5313            JMP. - 1        /IGNORE LEADER(L/T)
7715  3215  GO,       DCA CHKSUM
7716  1213            TAD MEMTEM
7717  3336            DCA MEMFLD
7720  1214            TAD CHAR
7721  3376            DCA WORDI
7722  4260            JMS READ
```

```
7723  3355              DCA WORD2
7724  4226              JMS BEGG        /LOOK AHEAD
7725  5275              JMP BEND        /TRAILER, END
7726  4343              JMS ASSEMB
7727  7420              SNL
7730  5336              JMP MEMFLD
7731  3216              DCA ORIGIN
7732  1376  CHEX,       TAD WORD1
7733  1355              TAD WORD2
7734  1215              TAD CHKSUM
7735  5315              JMP GO
7736  0000  MEMFLD,     0
7731  3616  7737        DCAI ORIGIN
7740  2216              ISZ ORIGIN
7741  7600  M200,       7600
7742  5332              JMP CHEX
7743  0000  ASSEMB,     0
7744  1376              TAD WORD1
7745  7106              CLL RTL
7746  7006              RTL
7747  7006              RTL
7750  1355              TAD WORD2
7751  5743              JMPI ASSEMB
7752  5262  LOR 1,      JMP LOR
7753  0006  HIRI,        HIR—LOR
7754  0000              0
            WORD1=7776
7755  0000  WORD2,      0
            *7777
7777  5301              JMP BEGIN
```

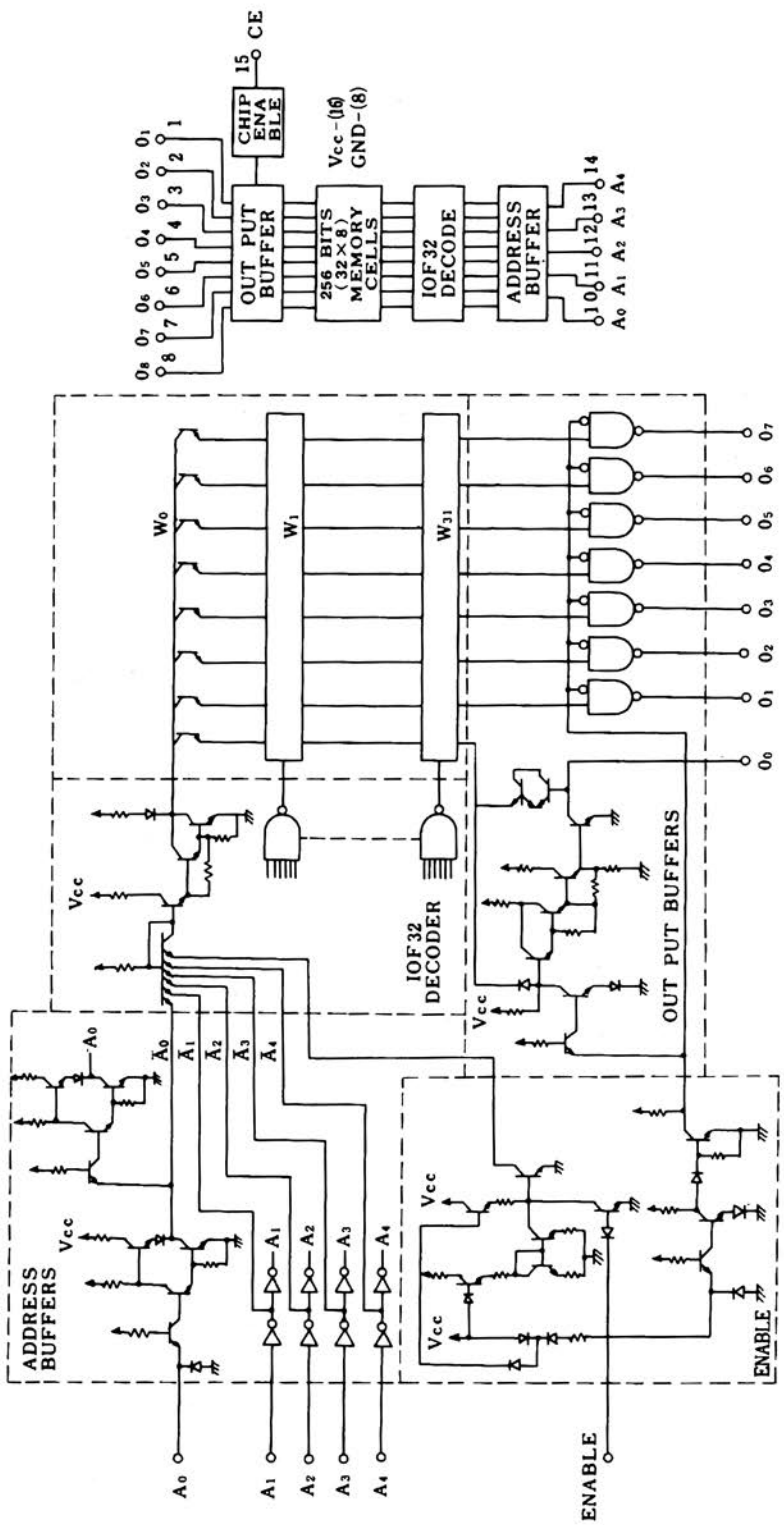
ASSEMB	7743
BEGG	7626
BEGIN	7701
BEND	7675
CHANGE	7657
CHAR	7614
CHEX	7732
CHKSUM	7615
FMASK	7656
GO	7715
HIR	7670
HIRI	7753
LOR	7662
LOR1	7752
MASK	7674
MEMFLD	7736
MEMTEM	7613
M200	7741
M376	7700
ORIGIN	7616
READ	7660
SWITCH	7612
WORD 1	7776
WORD 2	7755

10. 參考資料

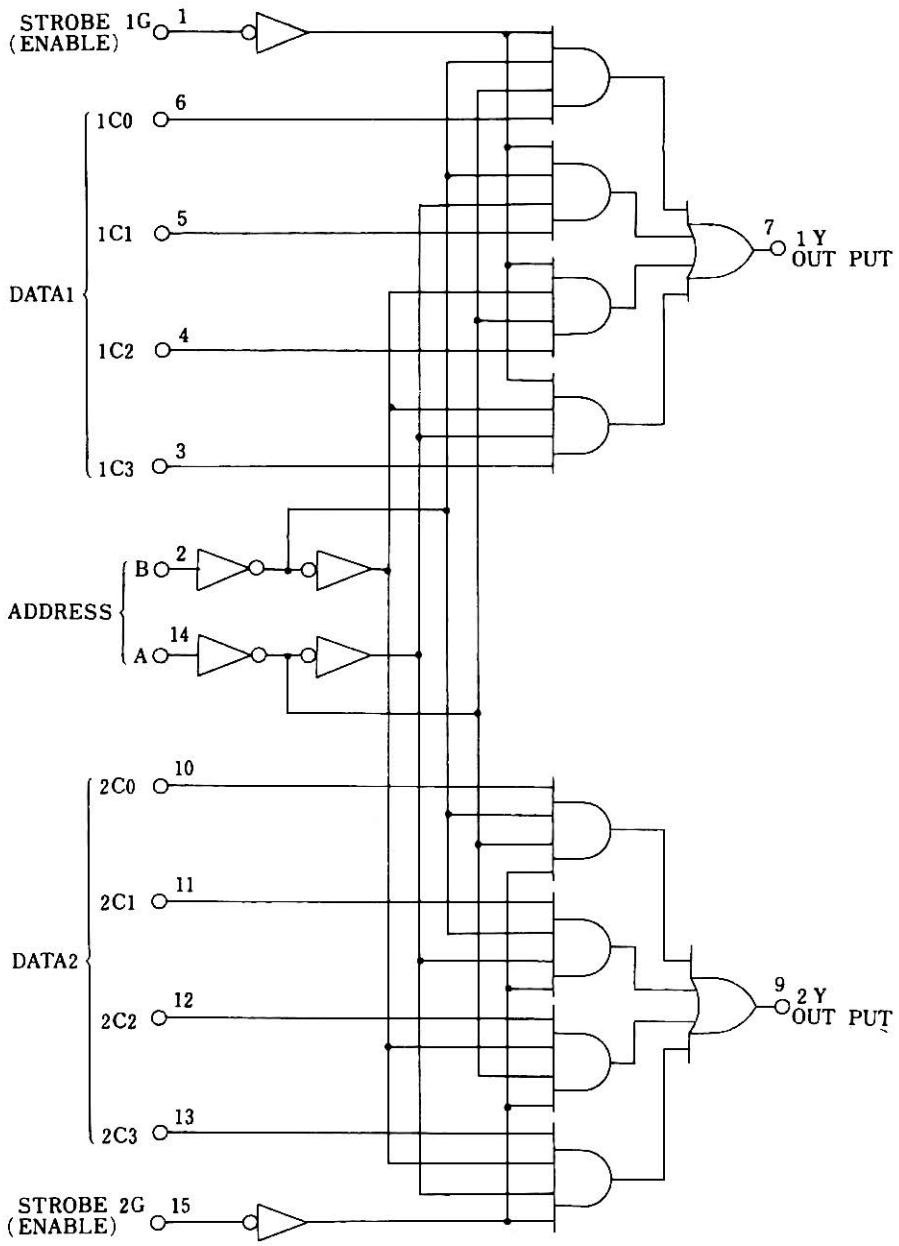
			t_{PLH}			t_{PHL}		
			MIN (ns)	TYP (ns)	MAX (ns)	MIX (ns)	TYP (ns)	MAX (ns)
SN74174 74175		CLOCK		20	30		21	30
		CRER		16	25		23	35
SN74123	Q	AINPUT		22	33			
		BINPUT		19	28			
	Q	AINPUT					30	40
		BINPUT					27	36
74H74		CLOCK	4	8.6	15	7	13	20
		CLER or RESET			20			30
7474		CLOCK	10	14	25	10	20	40
		CLER or PRESET			25			40
74H00N				5.9	10		6.2	10
7400N				11	12		7	15
74S04			2	3	4.5		3	5
74H04				6	10		6.5	10
7440				13	22		8	15
7437				13	22		8	15
DM8831				13	25		13	25

SN 74181 TABLE II

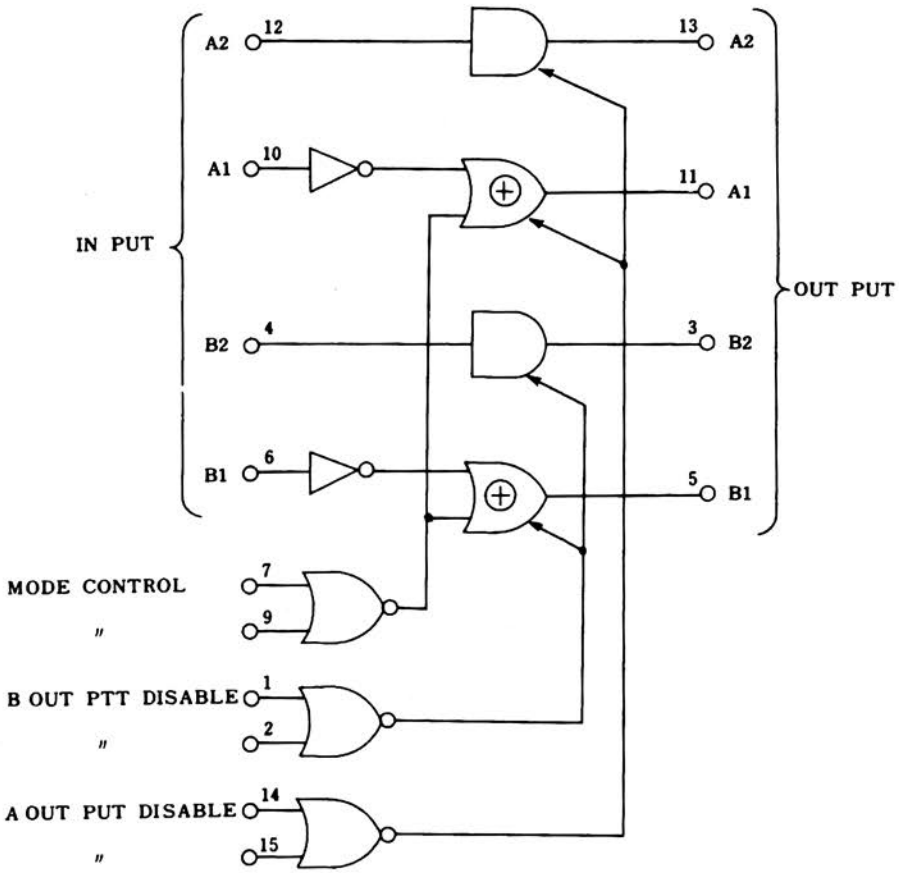
SELECTION S_3 S_2 S_1 S_0	ACTIVE-LOW DATA		
	M=H LOGIC FUNCTIONS	M=L ARITHMETIC OPERATIONS	
		C_n=0 C_n=0=L	C_n=1 C_n=1=H
L L L L	F=$\bar{A}$	F=A MINUS1	F=A
L L L H	F=AB	F=AB MINUS1 MI	F=AB
L L H L	F=A+B	F=A$\bar{B}$ MINUS1	F=A$\bar{B}$
L L H H	F=1	F=MINUS1(2s COMP)	F=ZERO
L H L L	F=A+B	F=A PLUS(A+$\bar{B}$)	F=A PLUS(A+$\bar{B}$)PLUS1
L H L H	F=$\bar{B}$	F=AB PLUS(A+$\bar{B}$)	F=AB PLUS(A+$\bar{B}$)PLUS1
L H H L	F=A$\oplus$B	F=A MINUS B MINUS1	F=A MINUS B
L H H H	F=A+$\bar{B}$	F=A+$\bar{B}$	F=(A+$\bar{B}$)PLUS1
H L L L	F=$\bar{A}$B	F=A PLUS(A+B)	F=A PLUS(A+B)PLUS1
H L L H	F=A$\oplus$B	F=A PLUS B	F=A PLUS B PLUS1
H L H L	F=B	F=A$\bar{B}$ PLUS(A+B)	F=A$\bar{B}$ PLUS(A+B)PLUS1
H L H H	F=A+B	F=A+B	F=(A+B)PLUS1
H H L L	F=0	F=A PLUS A	F=A PLUS A PLUS1
H H L H	F=A$\bar{B}$	F=AB PLUS A	F=AB PLUS A PLUS1
H H H L	F=AB	F=A$\bar{B}$ PLUS A	F=A$\bar{B}$ PLUS A PLUS1
H H H H	F=A	F=A	F=A PLUS1


15 CE
CHIP ENABLE
Vcc-(16)
GND-(8)
OUT PUT BUFFER
256 BITS (32×8) MEMORY CELLS
IOF32 DECODE
ADDRESS BUFFER
14
A0 A1 A2 A3 A4
O8 O7 O6 O5 O4 O3 O2 O1
1 2 3 4 5 6 7 8
W0
W1
W31
O7 O6 O5 O4 O3 O2 O1 O0
Vcc
IOF32 DECODER
OUT PUT BUFFERS
ADDRESS BUFFERS
A0 A1 A2 A3 A4
ENABLE

IM 5600


STROBE 1G
(ENABLE)
1
1C0
6
1C1
5
1C2
4
1C3
3
DATA1
7
1 Y
OUT PUT
B
2
A
14
ADDRESS
2C0
10
2C1
11
2C2
12
2C3
13
DATA2
9
2 Y
OUT PUT
STROBE 2G
(ENABLE)
15

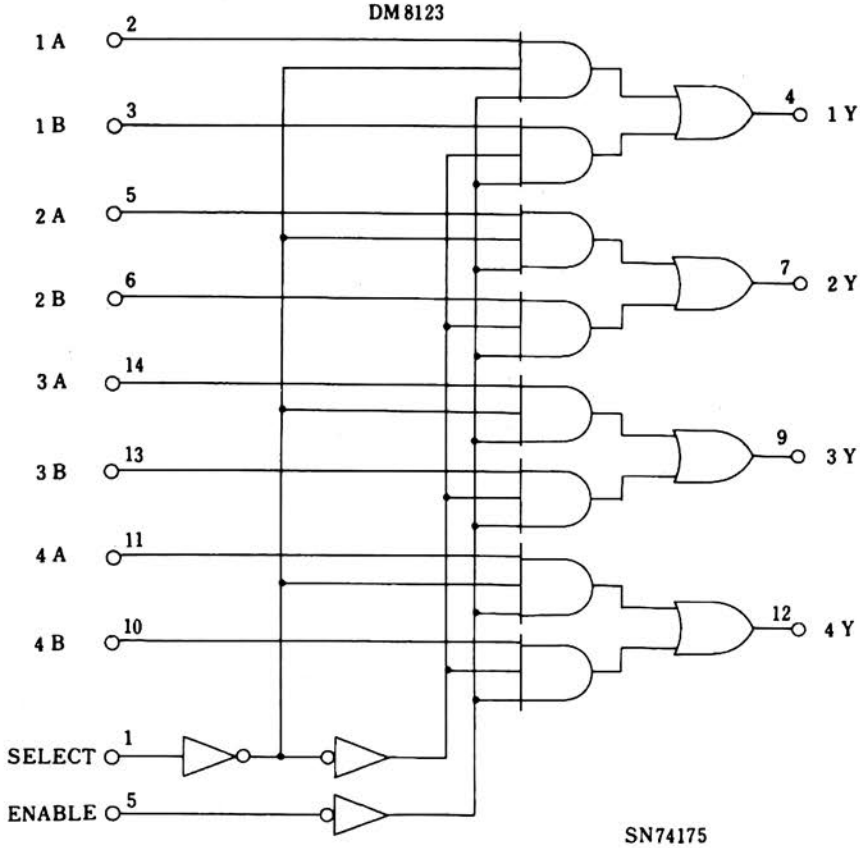
DM 8214



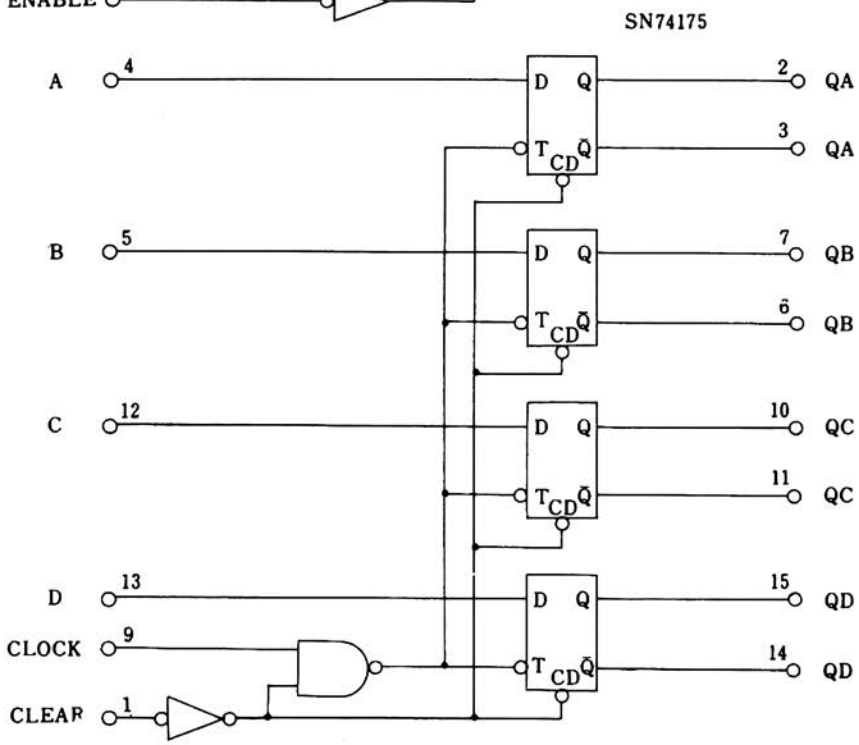
A OUT PUT DISABLE	MODE CONTROL	IN PUT A1	OUT PUT A1	IN PUT A2	OUT PUT A2
0 0	0 0	1 or 0	Same as in put A1	1 or 0	Same as in put A2
0 0	× 1 1 ×	1 or 0	opposite of in put A1	1 or 0	Same as in put A2
1	× ×	×	High impedance shart	×	High inpedance start

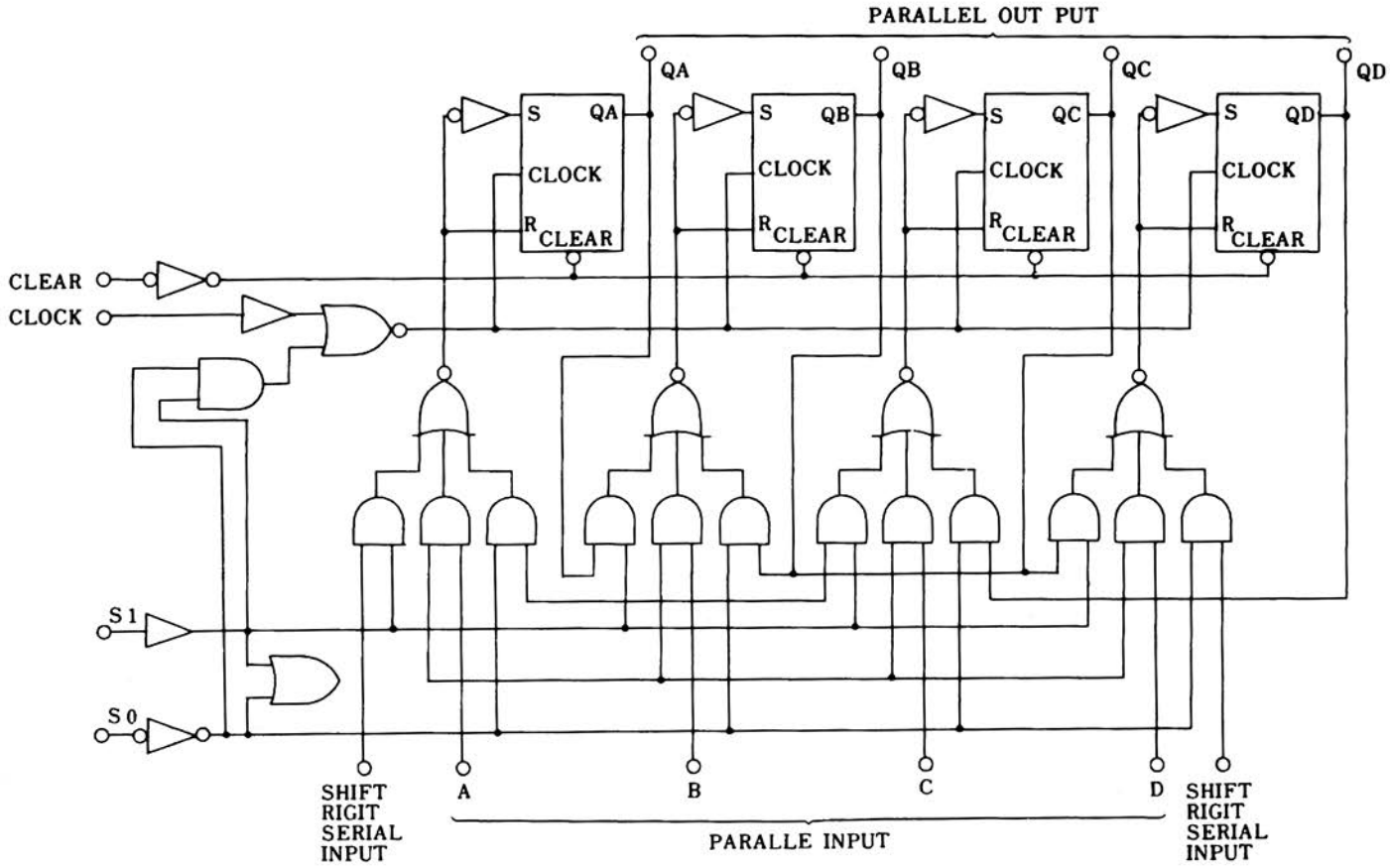
× = Dont Care

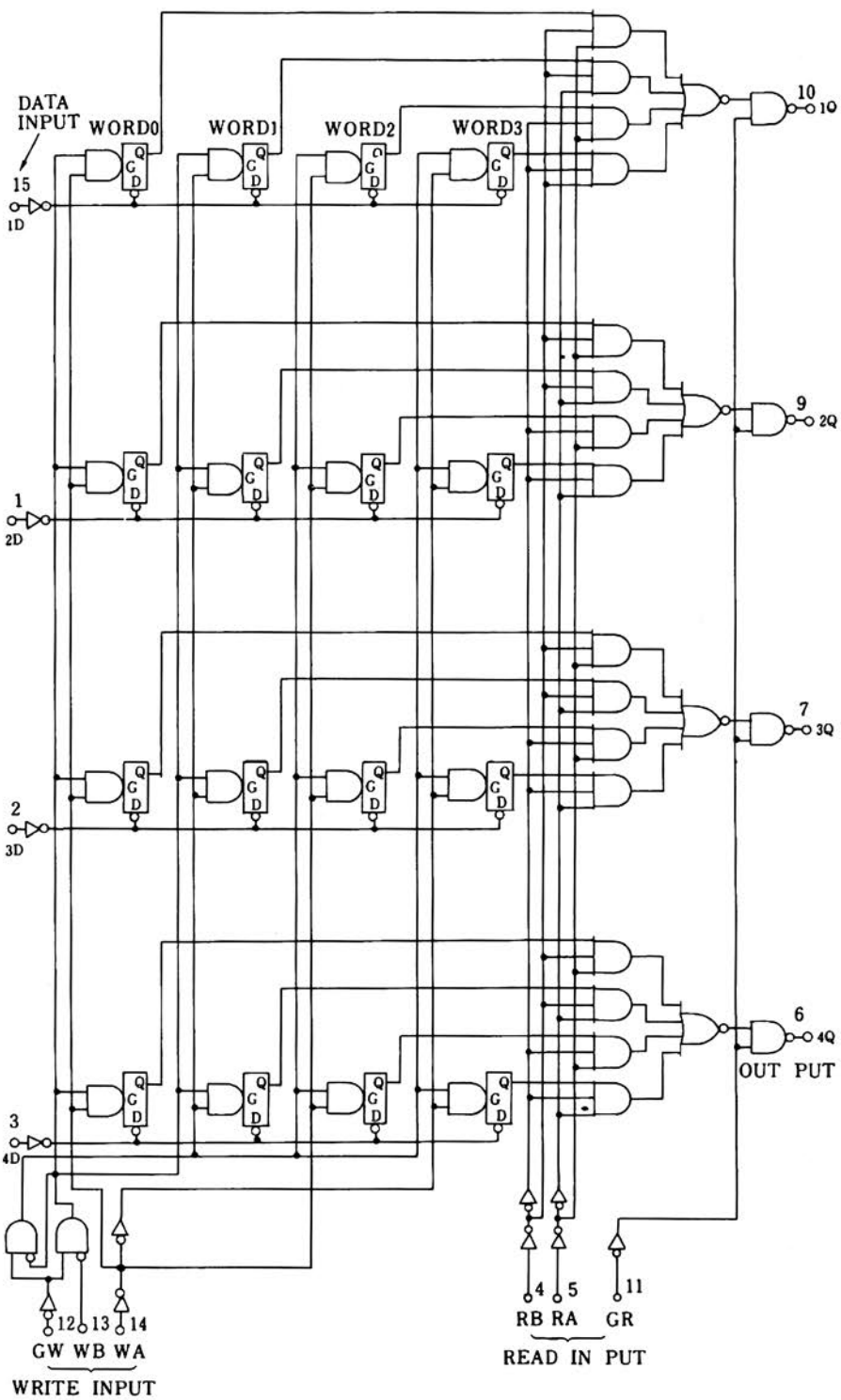
DM 8123



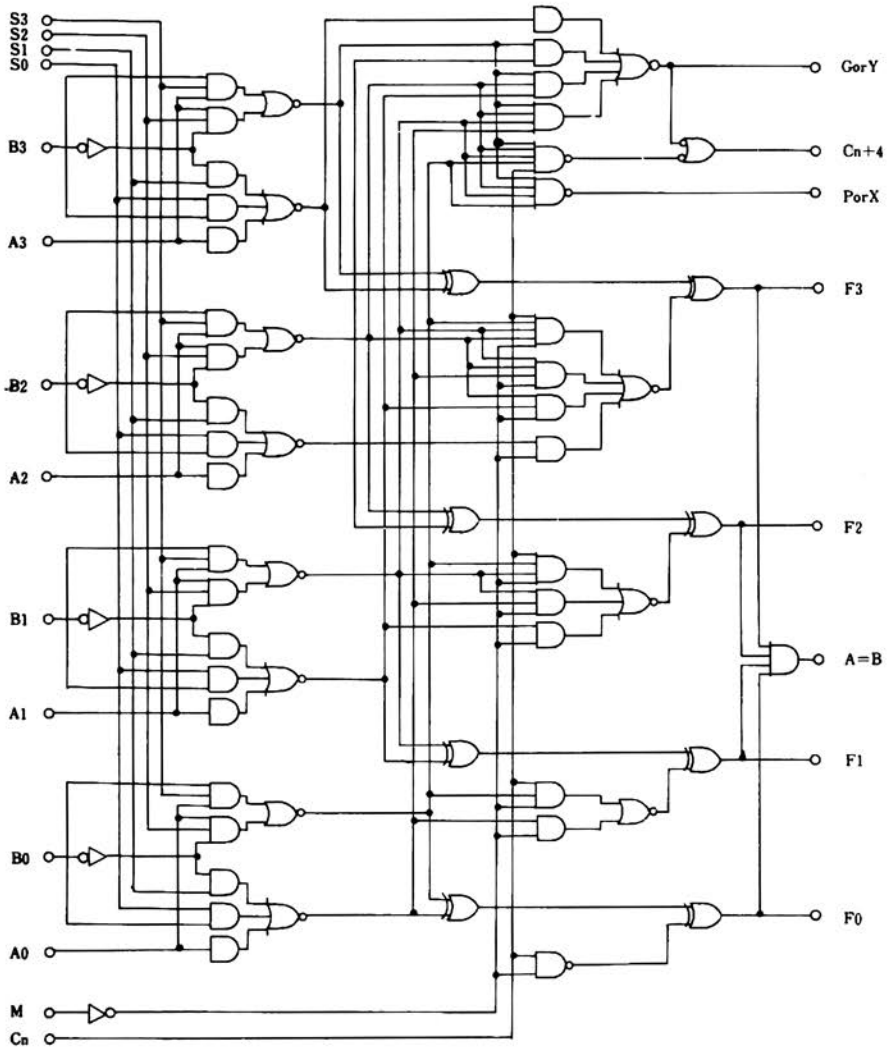
SN74175




PARALLEL OUT PUT
QA
QB
QC
QD
S
QA
CLOCK
R
CLEAR
S
QB
CLOCK
R
CLEAR
S
QC
CLOCK
R
CLEAR
S
QD
CLOCK
R
CLEAR
CLEAR
CLOCK
S 1
S 0
SHIFT
RIGIT
SERIAL
INPUT
A
B
C
D
SHIFT
RIGIT
SERIAL
INPUT
PARALLE INPUT


DATA INPUT
WORD0
WORD1
WORD2
WORD3
15
1D
1
2D
2
3D
3
4D
10
1Q
9
2Q
7
3Q
6
4Q
OUT PUT
12
13
14
GW WB WA
WRITE INPUT
4
5
11
RB RA GR
READ IN PUT

functional BLOCK diagram



11. インターフェース

11—1　CEC555H入出力インターフェースの概要

本章では，555H入出力インターフェースの機構と動作原理，接続法を説明する．

1．周辺装置がCEC555Hと情報交換するには2種類のインターフェースがある．
　1つは低速入出力データ転送であり，もう1つは高速入出力データ転送である．

2．低速入出力データ転送はCEC555Hのプログラムにより制御され，データの転送はACCUMULATOR（AC）を経由して行われる．
　低速入出力データ転送には，転送速度によってHモード，Lモードがある．

3．高速入出力データ転送は，周辺装置と555H記憶装置との間で直接データ転送を行う．転送速度は最大1000KW/Sで転送することができる．

4．高速出力データ転送は，6チャネル実装され，それぞれのチャネルには優先順位が設けられている．

11—2　入出力インターフェース機構図

図—1に入出力インターフェース機構図を示す．

低速入出力データ転送はCEC555H低速入出力制御装置を通して，処理装置（PU）のAC間でデータ転送を行う．高速入出力データ転送は高速入出力制御装置を通して記憶装置との間でデータ転送を行う．

高速入出力データ転送はブロック転送が主であるためプログラム制御下には入らないが，データ転送開始時および所定転送語数終了時には低速入出力データ転送機構を通してIOT命令による制御が必要となる．

11—3　低速入出力データ転送

低速入出力データ転送はIOT命令によって実行され，データは処理装置（PU）のアキュムレータ（AC）と入出力装置のバッファレジスタの間で転送される．

図—2にIOT命令の形式を示す．



図－2　IOT命令の形式

OP (Operation Code) は $(110)_2 = (6)_2$ である．

DVN (Device Number) は入出力装置の機番である．

ODR (Order) はDVNで指定された機番に対する入出力命令でODRによってIOT命令は実行される．

DVNは6ビットで64種類の機番を指定できる．

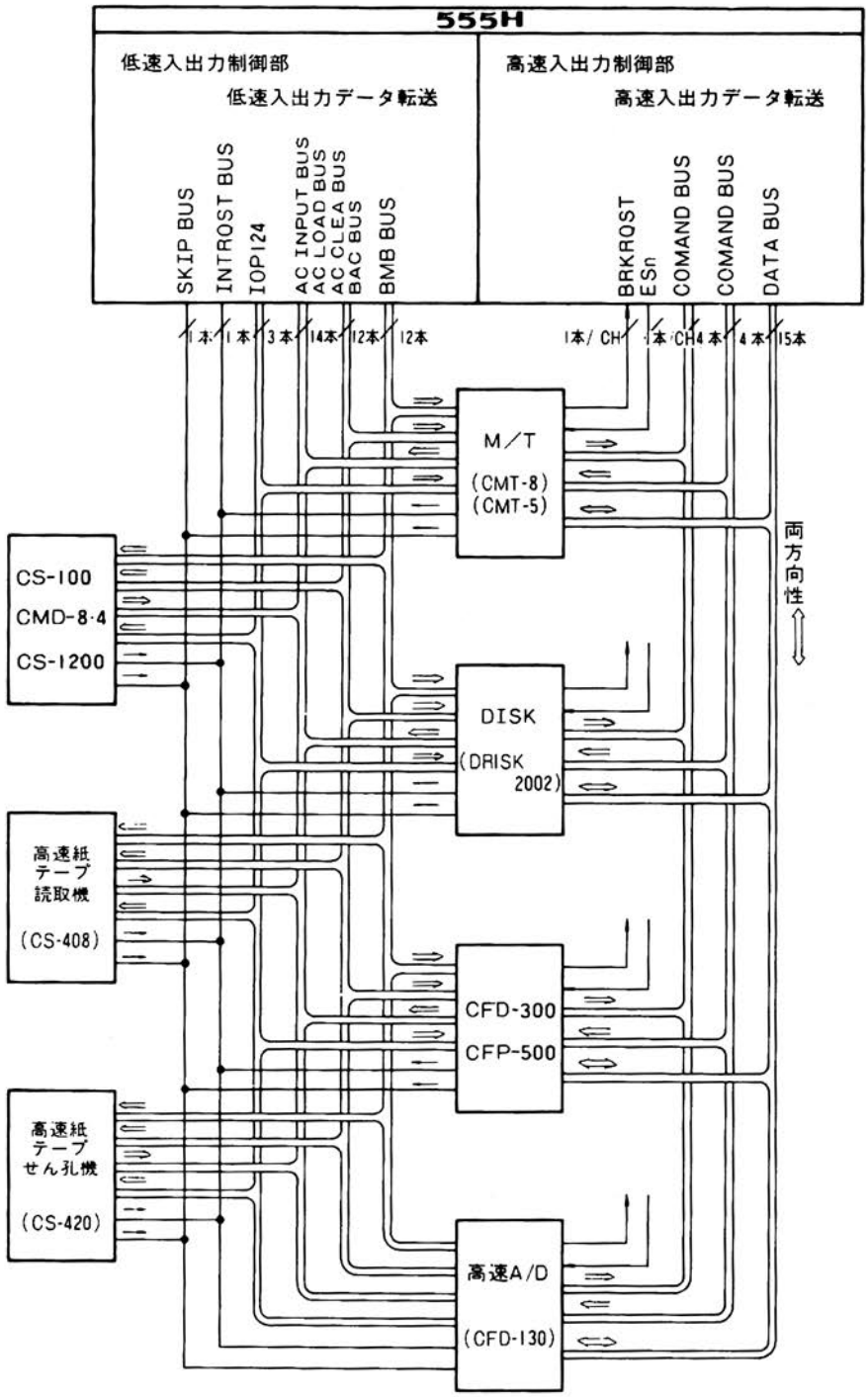
ODRは3ビットで8種類の命令を作ることができる．

11—3—1　低速入出力データ転送モード

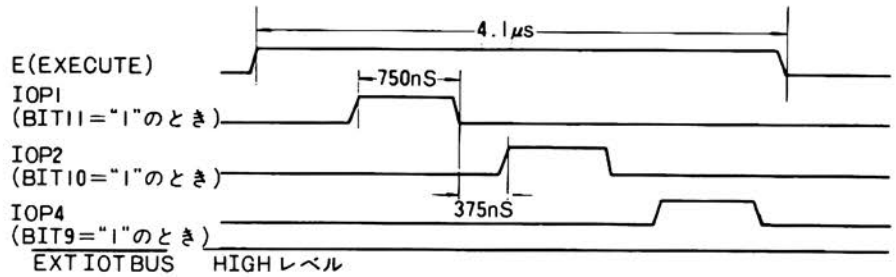
低速入出力データ転送には，転送速度によって2つのモードを選択することができる．1つはHモードIOTサイクル，もう1つはLモードIOTサイクルと呼ばれる．

1．LモードIOTサイクル

LモードIOTサイクルは$\overline{\text{EXT IOT BUS}}$がHIGHレベル（3.5V～4V）のとき，このサイクルが選択され，ODRビットに対応してIOPパルス（750ns）が発生する．



図－1　インターフェース機構図



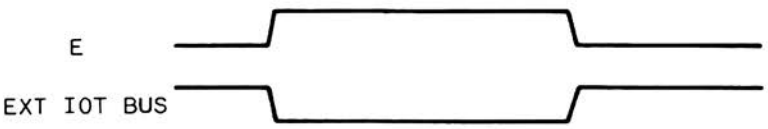
図－3　Lモード IOTサイクルタイムチャート

2．HモードIOTサイクル

HモードIOTサイクルは$\overline{\text{EXT IOT BUS}}$がLOWレベル（0～0.4V）のとき，このサイクルが選択され，IOPパルスは発生しない．555Hに基本構成として装備されているテレタイプ，PTR PTP等のコントロールはすべて，HモードIOTサイクルで実行される．

実行時間約1.3μSである．

〔FETCH CYCLE(0.9μs)＋EXECUTE CYCLE(0.375)＝1.27μs〕



図－4　HモードIOTサイクルタイムチャート

11－3－2　低速入出力機構の説明

紙テープ読取機，紙テープせん孔機等の低速入出力装置は，紙テープから1字読み取るとか，紙テープに1字パンチする等の動作を，すべてCPUのプログラムの制御のもとで行う．低速入出力インターフェースの説明をするに当って，入力データ転送（I/O→555H）は高速紙テープ読取機を，出力データ転送（555H→I/O）は高速紙テープせん孔機を例にとる．

(1) 低速入力データ転送

高速紙テープ読取機から紙テープの内容を読み取って，555Hに転送する場合いくつかのIOT命令が必要である．

IOT命令として次の三つを定義する．

表－2

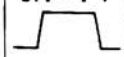
RFC(6014)$_8$	フラグと DATA BUFFER REGISTER をクリヤし紙テープから1字リードし DATA BUFFER に入れフラグをセットする
RSF(6011)$_8$	フラグが"1"のとき次の命令をスキップする
RRB(6012)$_8$	DATA BUFFER REGISTER の内容を AC に読みとりフラグをクリヤする

図－9 タイムチャート，図－6，7 制御部回路構成および図－5 のフローチャートを参照して低速入力データの手順を説明する．

《手順》

① CPUからPTR制御部（図－6 参照）にスタート指令としてRFC命令を出す．RFC命令を出

表－1　低速入出力データ転送信号名一覧表

転送方向	信号名	本数	内容	転送レベル
555H ↓ 外部入出力装置	BAC00～11	12	555HPU部のAC出力で12ビットのデータとして入力装置に転送される．信号レベルは HIGH TRUE（論理1＝2.4～4V）で転送される．	2.4～4 V 
	BMB00～11	12	BACがデータであるのに対してBMBはインストラクションである．BMB00～11の12本の信号でIOT命令を作る．HIGH TRUEで転送される．	
	BE	1	BMBの同期信号でEXECUTEサイクルの間HIGH.	
	BIOP 1,2,4	3	IOT命令のODR部3ビット（BIT 9,10,11）に対応して出力されるパルス．EXT IOT BUSがLOWの時は発生しない．	
	BINITIALTZE	1	外部入出力装置へのリセット信号 555Hの電源投入時とスタート時に出力される． HIGH TRUEで転送される．	
	$\overline{\text{BRUN}}$	1	555HPU部が動作中のとき0． 信号はLOW TRUEで転送．	
外部入出力装置 ↓ 555H	$\overline{\text{AC INPUT BUS 00～11}}$	12	外部入出力装置から555HPU部のACに転送する12ビットデータである．信号はLOW TRUE（論理"1"＝0V）で転送のこと．	
	$\overline{\text{AC LOAD BUS}}$	1	ACへのデータ書込み信号で，AC INPUT BUSにデータをのせAC LOAD BUSを"0"にするとデータがACに書込まれる．但しAC LOAD信号はIOT命令で同期をとること． 非同期でAC LOAD BUSを"0"にしても無効	
	$\overline{\text{I/O SKP BUS}}$	1	外部入出力装置の状態をセンスするために用いられる． I/O SKP BUSが"0"のときSKP-FFがセットされる． IOT命令で同期をとる事が必要．	
	$\overline{\text{INTRQST BUS}}$	2	外部入出力装置から555Hへの割込要求信号． LOW TRUE	
	$\overline{\text{AC CLR BUS}}$	1	555HPU部のACをクリヤする． 信号はLOW TRUEで転送． IOT命令で同期をとる事が必要．	
	$\overline{\text{EXT IOT BUS}}$	1	IOTの実行時間の選択． LOWでHモード，HIGHでLモード．	

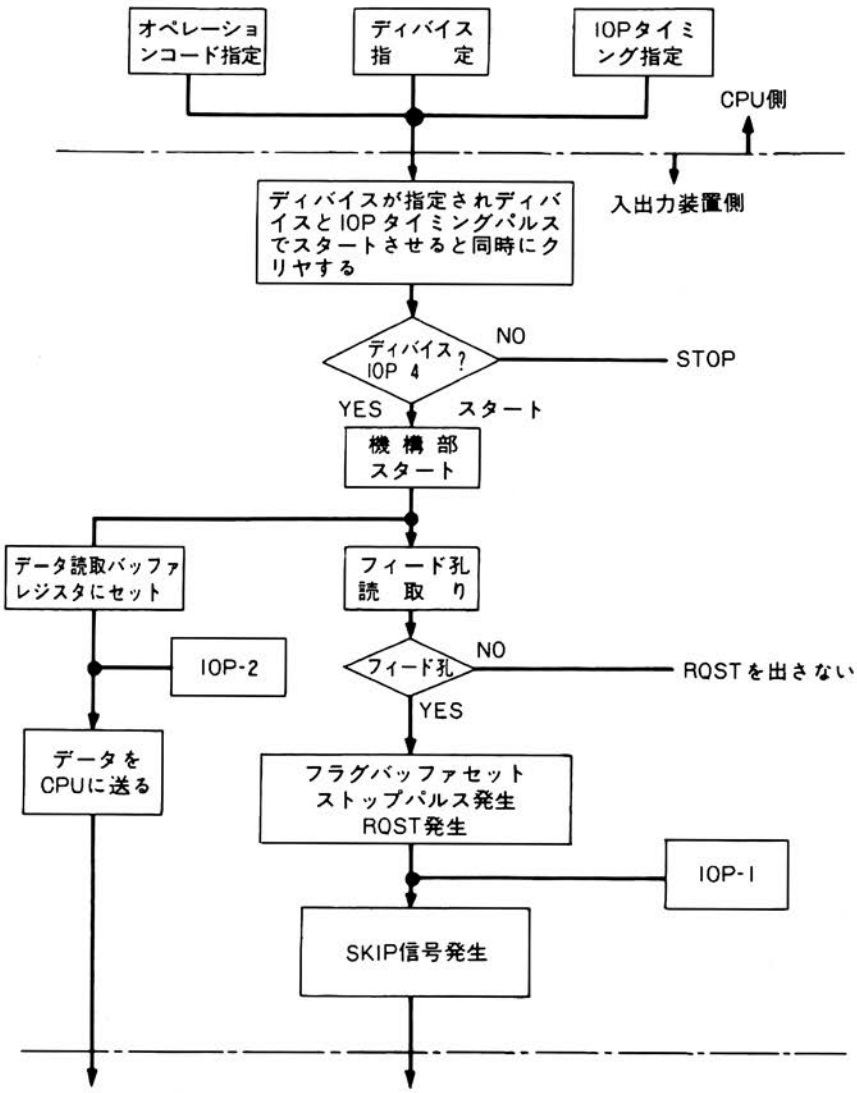
すとき，ACの内容はクリヤしておくことが必要である．

② PTR制御部ではRFC命令を受け取るとデコーダでデコードし，PTR本体（機構部とその付属回路）にスタート信号（READER START）を出すと同時にDATA BUFFER REGISTERの内容をIOP4の立上りでクリヤする．

③ PTR本体では紙テープの穴を電気信号に変えてPTR制御部のDATA BUFFER REGISTERに入れる．

④ PTR制御部ではDATA BUFFER REGISTERにDATAが入ると，このDATAをCPUに転送するためのフラグ（FLAG）セットし，割込要求信号（INTRQST）を出す（FLAG信号はINTRQSTラインに直結されていると同時にゲートを通してSKIPラインにも接続されていなければならない）．

⑤ CPUはPTR制御部からINTRQST信号があると区切のよいところで仕事を中断して，割込要



図－5　フローチャートによる高速紙テープ読取機動作説明

求の出ている入出力機器をさがすためにフラグセンス命令（RSF命令）を出す.

⑥ PTR制御部から出されているINTRQST信号とCPUから出されたフラグセンス命令（RSF命令）をデコードして，一致するとPTR制御部からSKIP信号を出す（SKIP BUSはCPUのスキップ機構に接続されておりSKIP BUSが“0”であると次に実行すべき命令をスキップする).

⑦ 入出力機器（PTR制御部）からSKIP信号が出され，SKIP BUSが“0”になるとCPUは次の命令をSKIPし，PTR制御部のDATA BUFFER REGISTERに入っているDATAをACに転送させるためにリード命令としてRRB命令を出し，DATA BUFFER REGISTERの内容をAC INPUT BUSにのせると同時にAC LOAD BUSを“0”にしてCPUのACに転送させる（ACに転送準備がととのえば，INTRQST信号は必要がないのでIOP 2の立上がレベルでINTRQST信号をクリヤする）

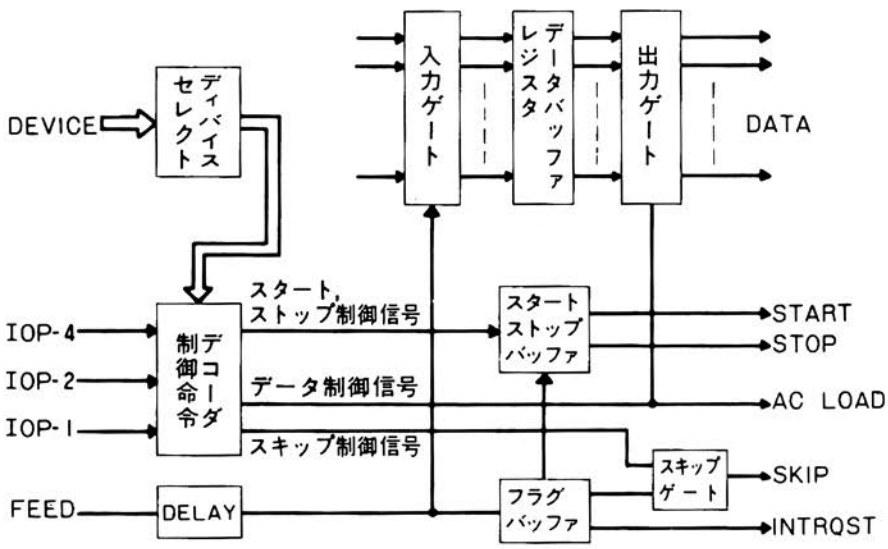
《低速入力データ転送のプログラム例》

```
        :
        :
    CLA
RTN, RFC          /FETCH CHARACTER
    RSF           /SKIP ON READER FLAG
    JMP0,-1
    RRB           /READ READER BUFFER
    DCA I SAVE    /DEPOSIT DATA
      :
      :
```

(2) 低速入力データ転送機構の具体的回路例

低速入力データ転送の具体的回路は紙テープ読取機の回路を例に上げる.

紙テープ読取機の制御部回路構成を**図—6**に示す.

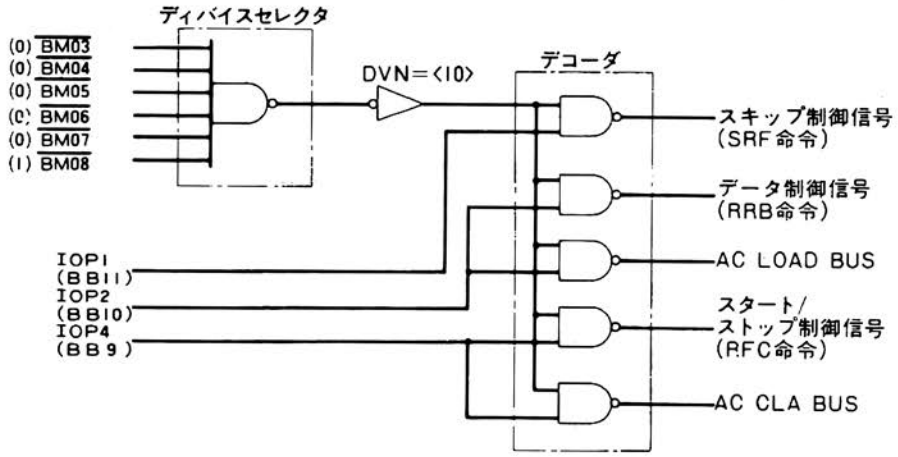


図—6 高速紙テープ読取機の制御部回路構成

(3) デバイス（DVC）選択／命令デコーダの具体的回路例の説明

CPUのBMB BUSによりディバイス指定をうけるとPTR制御部にその目的にあった回路を作らな

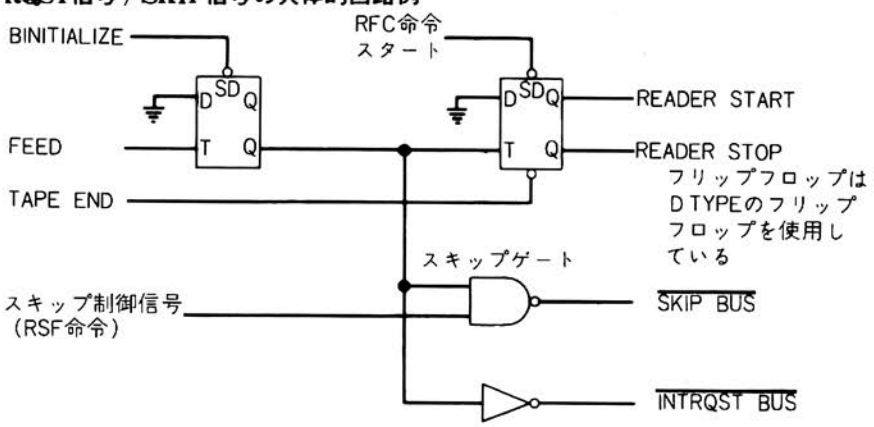
ければならない．具体的回路例を図—7に示す．NANDゲートの出力信号はIOP信号を組合せてスタート，ストップ制御信号，データ転送制御信号，スキップ信号およびAC LOAD信号を作りだす．



機番（DVN）を（010）$_8$に割当てる．

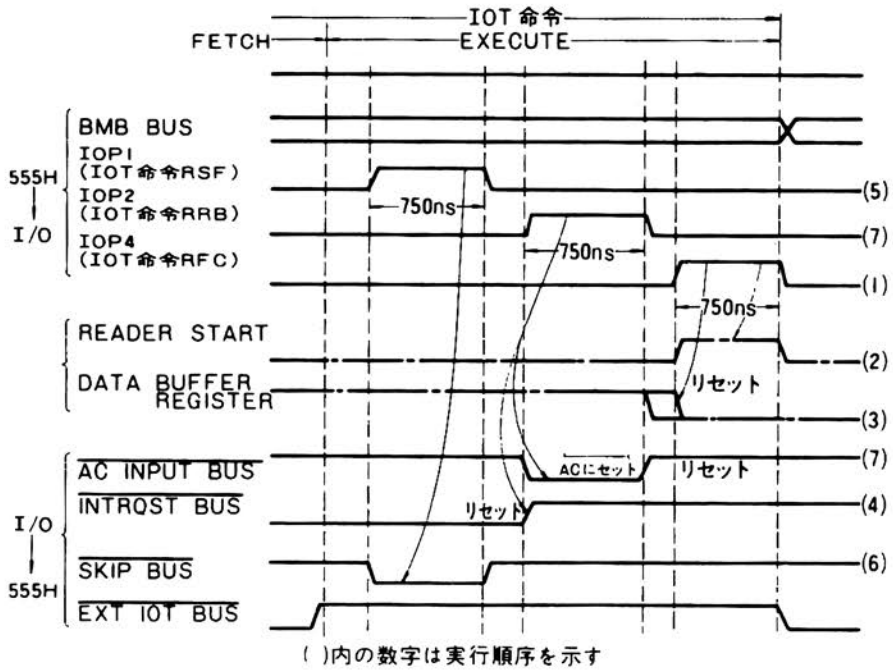
図－7　ディバイス選択／命令デコーダ回路例

(4) INTRQST信号／SKIP信号の具体的回路例



図－8　INTRQST／SKIP信号回路例

スタートストップ制御命令により，スタート駆動マグネットを駆動させると紙テープは動き，紙テープのフィード孔を読むとフラグフリップフロップがセットされ，ストップ信号を出して停止する．セットされたフラグフリップフロップの出力はINTRQST信号とスキップ制御ゲートに送られSKIP BUS，INTRQST BUSとしてCPUに転送される．



図－9　Lモード低速入力データ転送タイムチャート

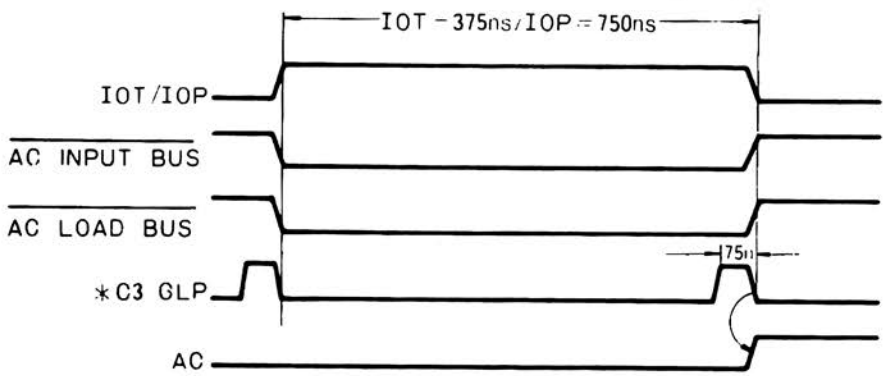
(5) AC LOAD BUS

データをIOT命令でACに書き込む場合のタイムチャートを図—10に示す.

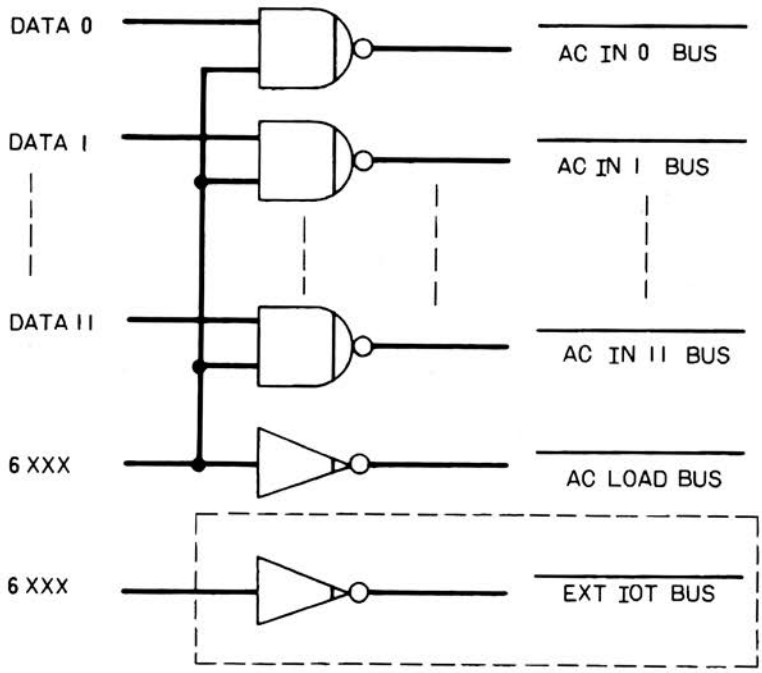
データはIOT命令（またはIOPパルス）の終端でACに書き込まれる.

データとともにAC LOAD信号が必要である.

HモードIOTサイクルを選択する場合６××で$\overline{\text{EXTIOT BUS}}$を「０」にする.



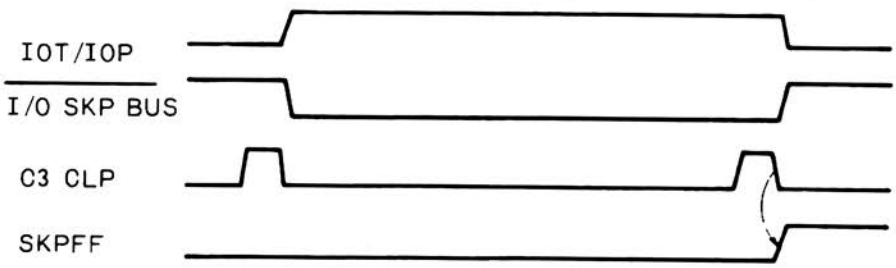
図－10　AC LOADタイムチャート



図－11　AC LOADの回路例

(6) SKIP BUS

外部入出力装置の状態をSKIP機構を用いてチェックする場合のタイムチャートを図－12に示す.



図－12　I／O　SKIPタイムチャート

IOT命令（またはIOPパルス）の終端でSKIP F.Fがセットされ，次のFETCHサイクルでメモリアドレスはPC＋2が指定される（SKIP F.F＝0の時PC＋1が指定される＝通常動作).

11－3－3　低速出力データ転送

出力データ転送は入力データ転送の場合と同じでCPUからテープせん孔機にDATAを転送するには命令が必要である．IOT命令として次の三つを定義する.

表－3

TPC(6044)$_8$	ACの内容（A（04～11））をせん孔する
TSF(6041)$_8$	フラグが1のとき次の命令をスキップする
TCF(6042)$_8$	フラグをクリヤする

図—13の出力転送タイムチャートおよび図—14の紙テープせん孔機制御部回路構成を参照して低速出力データ転送の手順を説明する.

《手順》

① CPUからPTP制御部（図—14参照）にスタート指令としてTPC命令を出す. TPC命令を出すときCPUのBAC OUT PUT BUSにはDATAが確定されていなければならない.

② PTP制御部ではTPC命令を受取るとデコーダでデコードし，PTP本体にスタート信号（PUNCHER START）を出すと同時に，IOP 4 の立上りでDATA BUFFER REGISTERにBAC OUTPUT DATAをセットする.

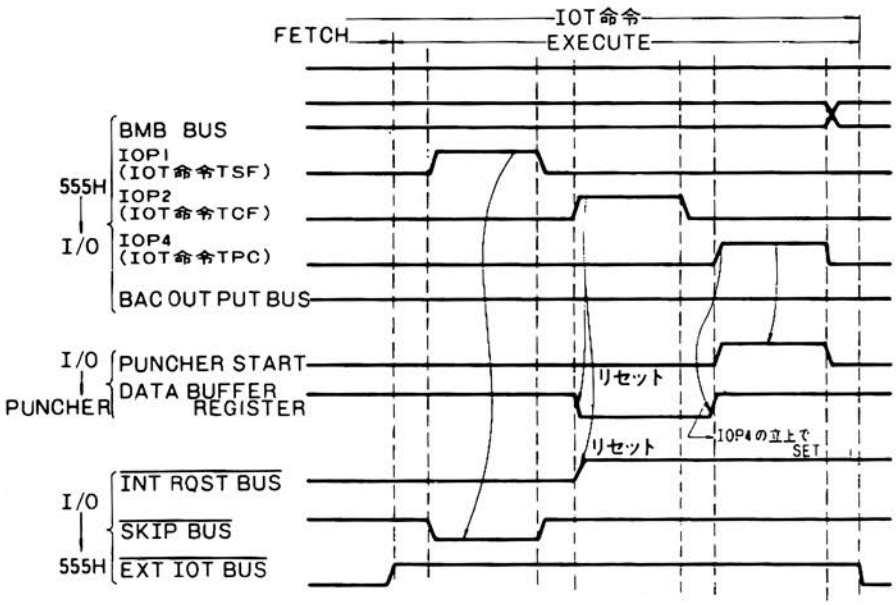
③ PTP本体ではPTP制御部のDATA BUFFER REGISTERの内容をPUNCHし終ると，パンチ終了信号PRをPTP制御部フラグフリップフロップに送りフラグフリップフロップをセットし割込要求信号（INTRQST）を出す.

④ CPUはPTP制御部からINTRQST信号がくると区切のよいところで仕事を中断して割込要求の出ている入出力機器をさがすためフラグセンス命令，TSF命令を出す.

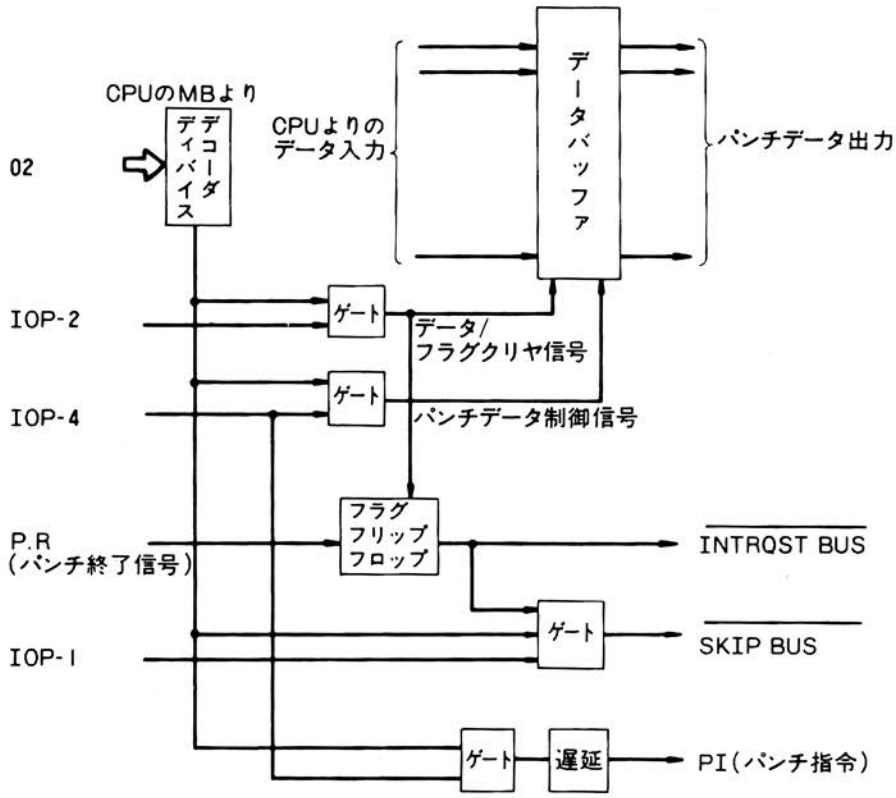
⑤ PTP制御部から出された割込要求信号（INTRQST）と，CPUからのフラグセンス命令（TSF命令）をデコードして，一致すると，PTP制御部からSKIP信号を出す.

⑥ PTP制御部からSKIP信号が出され，SKIP BUSが“ 0 ”になるとCPUは次の命令をSKIPし，PTP制御部のINTRQST信号およびDATA BUFFER REGISTERの内容をクリヤする信号としてTCF命令を出す. IOP 2 のレベルでフラグフリップフロップおよびDATA BUFFER REGISTERをクリヤする.

以上で1字のパンチ処理が終了，次に1字パンチするときもまた①から始まる.



図－13 Lモード低速出力データ転送タイムチャート



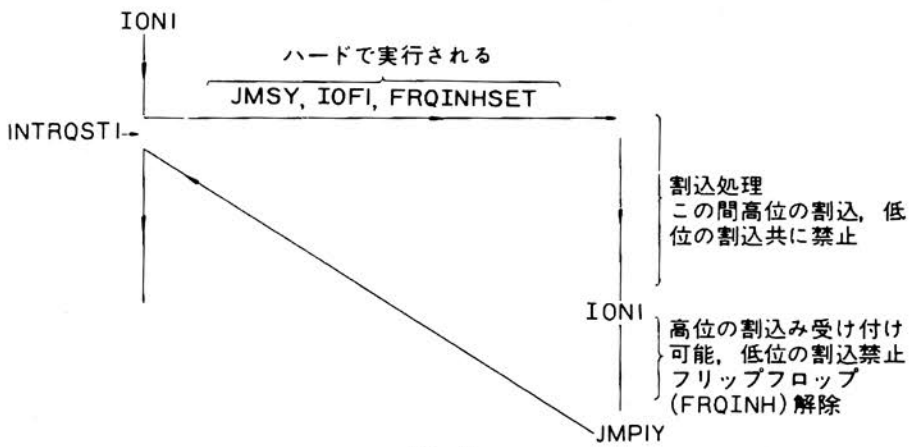
図－14　紙テープせん孔機制御部回路構成

11－4　インタラプト

外部入出力装置からPUへの割込み要求信号でINTRQST 1，INTRQSTの２本である．INTRQST 1はINTRQSTに優先して受付けられる．

インタラプトラインを１本のみ使用する場合はINTRQST 1，またはINTRQSTどちらを用いても従来の割込み処理と変るところはない．２本使用する場合には優先順位を考慮して入出力装置のスピードに合わせてインタラプトラインを選択し，プログラムを組むことが必要になる．

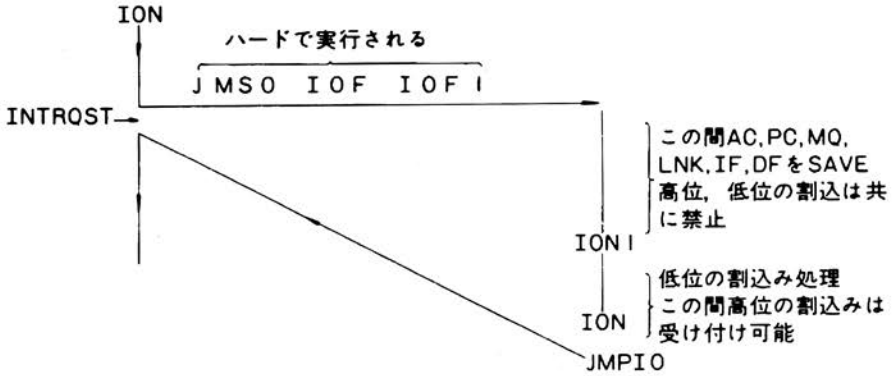
11－4－1　高位の割込処理



図－15

高位の割込みが発生すると，JMSY（注）が実行され次の命令は0フィールドのY＋1番地から読み出される．同時に高位，低位の割込みは禁止され，高位の割込処理が終って再びION 1命令が実行されるまで割込みは受付けられない．ION 1で高位は割込可能状態となるが，低位は禁止を解除するのみであって，高位の割込みが発生したときの低位の状態に戻すだけである．従って割込可能状態になるとは限らない．

11—4—2　低位の割込み処理



図—16

低位の割込みが発生するとJMS 0が実行され，次の命令は0フィードの1番地から読み出される．同時に高位，低位の割込みは禁止される．

低位の割込み処理に先だってAC，PC，MQ，LNK，IF，DFなどをメモリにセイブするが，この後にION 1で高位の割込みを受付け可能にしておく．

低位の割込みが発生した時点では，高位は必ず割込み可能状態であったはずであるから，無条件でION 1としてよい．ION 1とするまでは高位は禁止されるので注意を要する．

（注）高位の割込みが発生するとJMSYが実行されるが，JMSY以外に任意の命令を設定できる．
ただし半固定のワイヤリング方式（ICソケット使用）でプログラマブルではない．
命令コードは4(JMS)，5(JMP)，6(IOT)，7(OPR）のいずれかである．電源異常割込み，メモリパリティエラー割込みはINTRQST 1に属する．

11—5　I/Oインターフェース

外部端末装置と５５５Ｈとのデータ転送（AC経由）に関する信号は以下の10種類である．

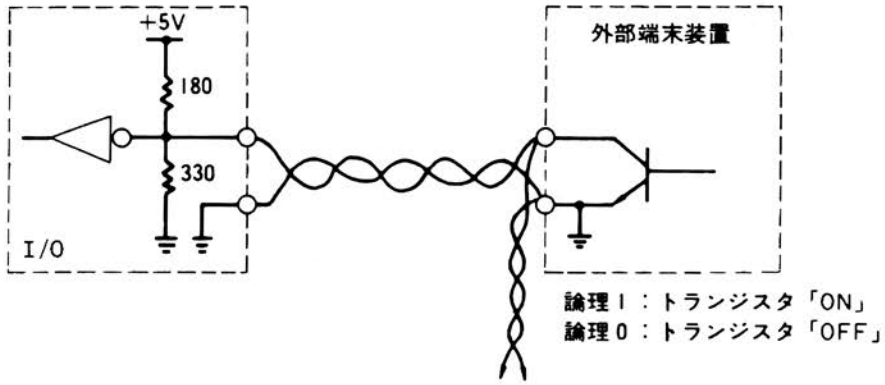
11—5—1　５５５Ｈ→外部装置

1.1	BAC	ACCUMULATOR 出力	12本
1.2	BMB	MEMORY BUFFER REGISTER 出力	12本
1.3	BE	BMB の同期信号	1本
1.4	BIOP1.2.4	IOP パルス	3本

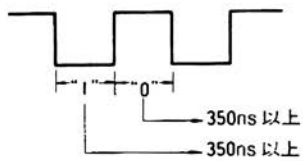
11—5—2　外部装置→５５５Ｈ

2.1	AC BUS	ACCUMULATOR への入力	12本
2.2	AC LOAD BUS	ACCUMULATOR へのデータ書き込み信号	1本
2.3	INT BUS	INTERRUPT 入力	2本
2.4	SKIP BUS	SKIP 入力	1本
2.5	EXTIOT BUS	IOT 実行速度の切換	1本
2.6	AC CLR	AC クリヤ	1本

1. AC IN PUT. AC LOAD, AC CLR, INT, SKP, EXTIO信号入力規格



図－17　信号入力インターフェース



図－18

(1) I/O側規格

①	入力形式	ＩＣ（終端抵抗付）
②	流出電流	約27mA
③	入力スレッシュホールドレベル	約0.8V
④	応答時間	約50ns
⑤	パルス入力条件	図－18参照

(2) 外部端末装置側規格

①	出力形式	接地形トランジスタスイッチ
②	電流容量	50mA以上
③	耐　　圧	1.5V以上
④	飽和電圧	１Ｖ以下

(3) 接続ケーブル

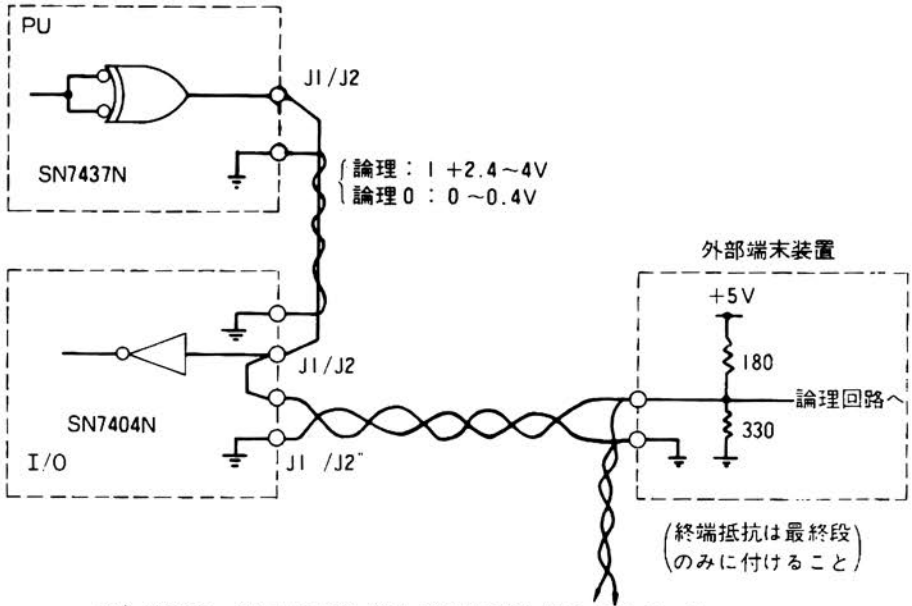
① 浮遊容量100PF/m以下であること.

② ２ヶヨリ線で構成され，多芯一括シールドケーブルを使用することが望ましい.

③ ケーブルの特性インピーダンスは130Ωとする.

④ ケーブルの最大長は５ｍとする.

2．BAC，BMB，BE，BIOP，BINITIALIZE信号出力規格



注）BIOPパルスはI/O部よりSN7437出力がでている．

図－19　信号出力インターフェース

(1) I/O側規格

① 出力形式　　IC (SN7437N QUADRUPLE 2-INPUT POSITIVE NAND BUFFERS)

② 最大印加電圧　5.25V

③ 最大負荷電流　46mA

④ 飽和電圧　　0.4V（最大負荷電流のとき）

(2) 外部端末装置側規格

① 入力形式　　IC（終端抵抗付）

② 入力スレッシュホールドレベル 0.8V＋ケーブル抵抗による電圧降下分

(3) 接続ケーブル

① 浮遊容量100PF/m以下であること．

② 2ヶヨリ線で構成され，多芯一括シールドケーブルを使用することが望ましい．

③ ケーブルの特性インピーダンスは130Ωとする．

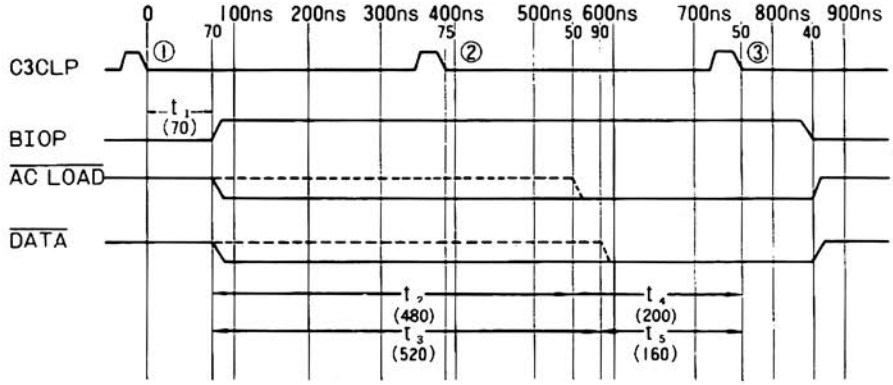
④ ケーブルの最大長は5mとする．

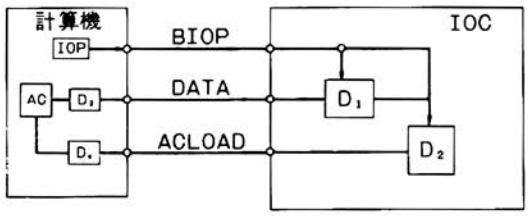
11－6　低速入出力信号の遅れ時間に関する規定

11－6－1　Lモード データ転送

IOPパルスを用いて，IOCからデータをアキュムレータに書き込む場合の時間関係について規定する．

計算機内部の動作は，すべてC3CLPパルスの後縁で規定される．C3CLP①の後縁から，約70ns遅れてBIOPパルスがIOCに送られる．C3CLP③の後縁でデータはアキュムレータに書き込まれるが，データに関する計算機内部の遅れ時間（計算機のインプットにデータが入力されてから，アキュムレータに達するまでの時間）が約160nsあるので，少なくともC3CLP③の後縁の160ns前までにデータが，計算機にインプットされなければならない．したがってIOCにBIOPパルスが入力され





$$\begin{cases} D_1 = t_3 - t_{DELAY\cdot CABLE} = 520ns - t_{DLY\cdot CBL} \\ D_2 = t_2 - t_{DLY\cdot CBL} = 480ns - t_{DLY\cdot CBL} \\ D_3 = t_5 = 160ns \\ D_4 = t_4 = 200ns \end{cases}$$

($t_{DLY\cdot CBL}$ は往復の遅れ時間)

図－20

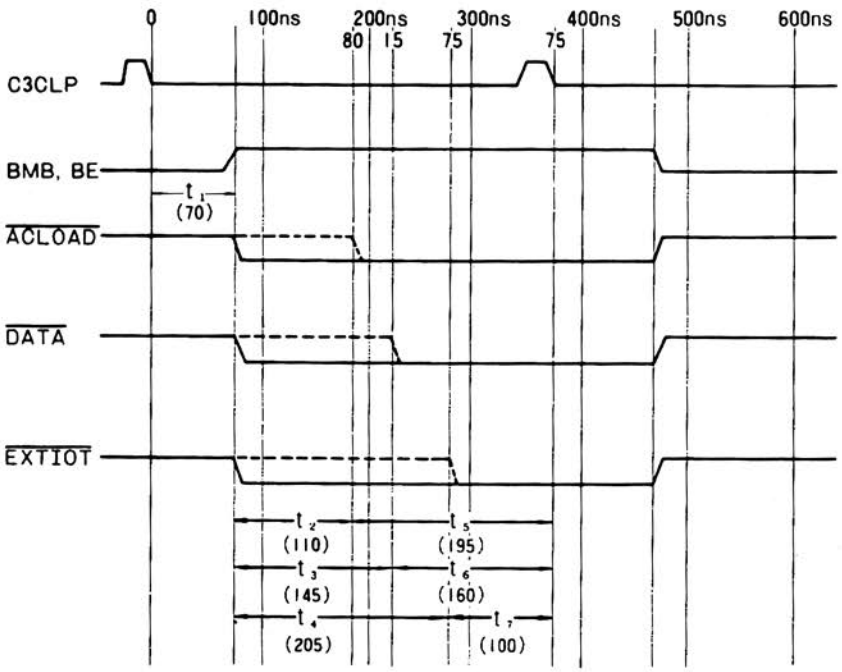
てから，データを送り出すまでのIOC内部での最大遅れ時間は約520ns－（ケーブルの遅れ時間）となる．

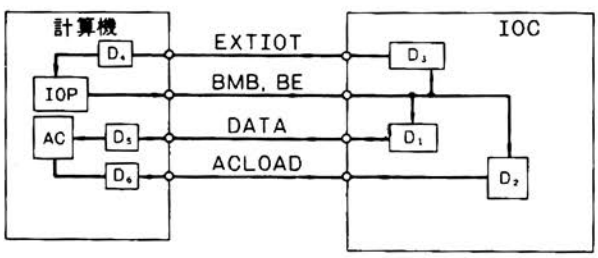
ケーブルの遅れ時間は 6 ns/mとして計算する．

AC LOAD信号が計算機に入力されてから，アキュムレータの書き込みゲートが開くまでの時間は約200nsである．したがって，IOCにBIOPパルスが入力されてからAC LOAD信号を送り出すまでの，IOC内部での最大遅れ時間は約480ns－（ケーブルの遅れ時間）となる．

11—6—2 Hモード データ転送

HモードIOTサイクルでIOCからデータをアキュムレータに書き込む場合の時間関係について規定する．





図－21

$D_1 = t_3 - t_{DELAY \cdot CABLE} = 140ns - t_{DLY \cdot CBL}$

$D_2 = t_2 - t_{DLY \cdot CBL} = 100ns - t_{DLY \cdot CBL}$

$D_3 = t_4 - t_{DLY \cdot CBL} = 200ns - t_{DLY \cdot CBL}$

$D_4 = t_7 = 100ns$

$D_5 = t_6 = 160ns$

$D_6 = t_5 = 200ns$

Hモードデータ転送の場合，新たに$\overline{\text{EXT IOT}}$信号が必要である（$\overline{\text{EXT IOT}} = 0$にする）．BMB，BEパルスがIOCに入力されてから，データを送り出すまでのIOC内部の最大遅れ時間は約140ns－（ケーブルの遅れ）である．同様にAC LOAD信号については，IOC内部の最大遅れ時間は，約100ns－（ケーブルの遅れ），EXTIOT信号については約200ns－（ケーブルの遅れ）となる．

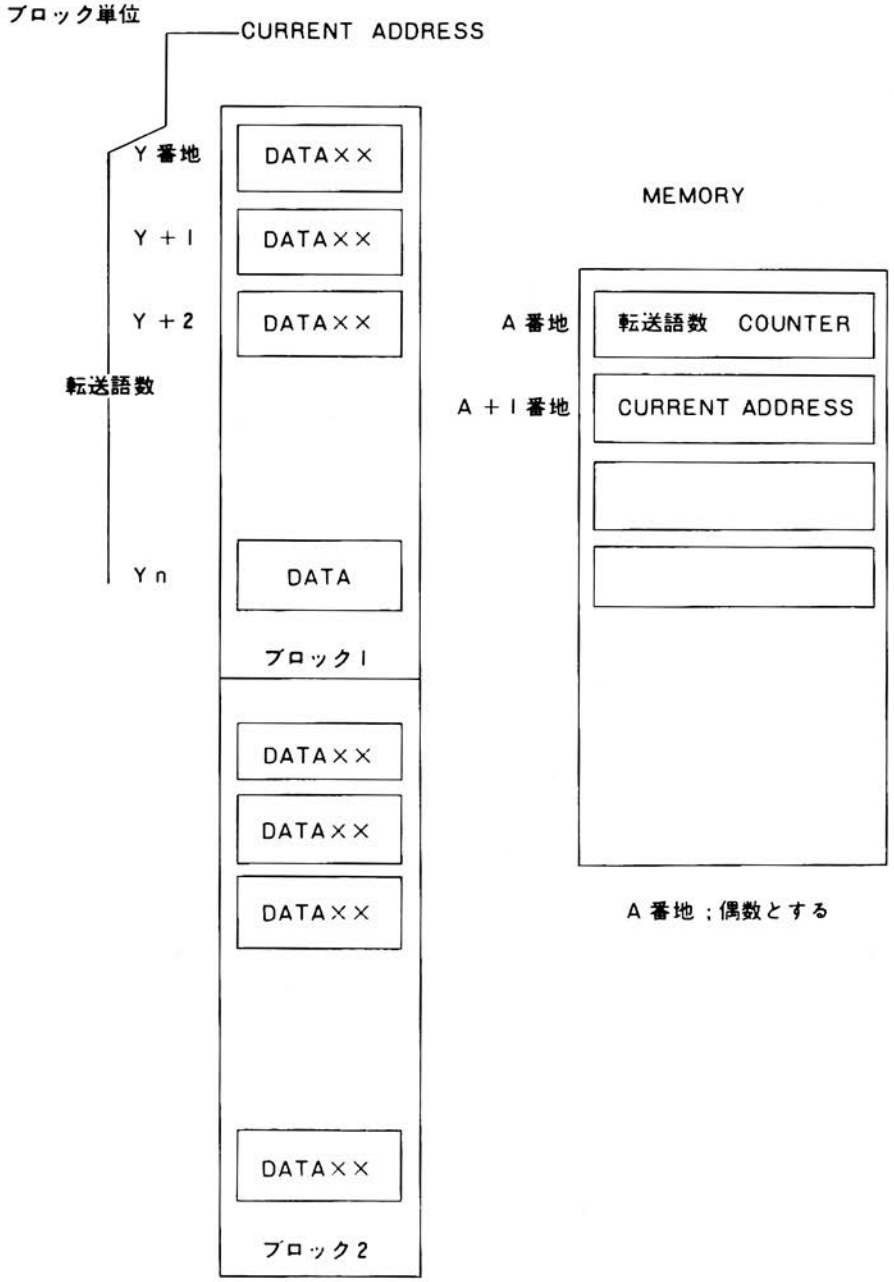
11—7　高速入出力データ転送

11—7—1　概　要

CEC 555H計算機は低速入出力データ転送機構のほかに，高速の入出力データ処理用としてDIRECT STORAGE ACCESS機構（DSA）が標準装備されており，入出力装置6台まで接続することができる．DSA方式によるデータ転送はプログラム制御による低速入出力転送と違って入出力装

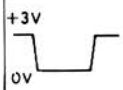
置と５５５H記憶装置との間で，直接データ転送が行われ，最大1000KW/Sで転送することができる．

ADDRESS DATA, READ DATAの転送はすべて両方向性DATA BUS LINEで行われるためケーブルの本数コネクタが少なく非常に経済的である．

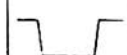


図－22

表－4〈その1〉　高速データ転送信号名一覧表

転送方向	信号名	本数	内容	転送レベル
入出力装置 ↓ 5 5 5 H へ	$\overline{\text{RQSTn}}$ ＝0～5	1本 /CH	DSA の要求信号，各周辺装置独立の信号	
	CYC SEL	1本	1 CYCLE か 3 CYCLE か指定する信号 0V：3 CYCLE 指定 3 V：1 CYCLE 指定	
	$\overline{\text{TRANSF DIR}}$	1本	DATA 転送方向を指定する信号 0 V：DATA OUT PUT 指定(READ DATA) 3 V：DATA IN PUT 指定(WRITE DATA)	
	$\overline{\text{INHCAINCR}}$	1本	3 CYCLE 指定のとき CA CYCLE において DMB の内容に INCREMENT を禁止する信号． 3 V：禁止	
	$\overline{\text{MBINCR}}$	1本	WRITE DATA 又は MEMORY からの READ DATA を MODIFY する信号．	
	$\overline{\text{DB 00～11}}$	12本	ADDRESS DATA WRITE DATA はすべてこの DATA LINE にのせる．	
	$\overline{\text{DB 12～14}}$	3本	MEMORY の FIELD 指定 4KW 単位 8 FILD 32KW まで指定することができる．	

表－4 〈その2〉　高速データ転送信号名一覧表

転送方向	信号名	本数	内容	転送レベル
555H→入出力装置へ	$\overline{ESn}$ ＝0～5	1本/CH	I/Oからの RQSTを"受付けた"という応答信号 ESn が続行中は他の RQST は受付けない.	
	$\overline{BAKN}$	1本	ADDRESS DATAを"受付けた"という返事.	
	$\overline{BDAVL}$	1本	TRANSF DIR が DATA OUT PUT 指定のとき READ DATA の STROBE としてこの信号が発生.	
	$\overline{BDRCV}$	1本	TRANSF DIR が DATA INPUT指定のとき	
	$\overline{BBRK}$	1本	DSA CYCLE 期間信号でこの時間帯に READ DATA 及び WRITE DATA が定まる.	
	$\overline{BWCOF}$	1本	3CYCLE 指定のとき WCCYCLE において WORD COUNTE REGISTER がオーバーフローしたとき又は 1CYCLE指定で MBINCR で READ DATA を MODIFY したとき DMB COUNTERがオーバーフローしたときこの信号が発生	
	BTSI1	1本	TIMING 信号 DATA をクリヤしたり CONTROL 信号に用いてもよい.	
	BTS 3	1本	TIMING 信号	
	$\overline{DB00～11}$	12本	DATA を転送する BUS LINE	

DSAによる転送方法は低速入出力データ転送と違ってブロック転送（図—22参照）が主であるため転送語数カウンタ（WORD COUNT REGISTER）およびADDRESS COUNT REGISTERが必要となる．転送語数カウンタ，ADDRESS COUNT REGISTERを入出力装置側で持つか，555H側で持つかによって転送方法は3通りに分けられ，それぞれ1サイクル，2サイクル，3サイクルデータ転送と呼ばれる．

1サイクルデータ転送は：転送語数カウンタ，ADDRESS COUNT REGISTERは入出力装置側で持ち，転送速度は1000KW/Sとなる．

2サイクルデータ転送は：転送語数カウンタは入出力装置側で持ちADDRESS COUNT REGISTERは555H側で持つ，転送速度500KW/S

3サイクルデータ転送は：転送語数カウンタ，ADDRESS COUNT REGISTERは555H側で持つ，転送速度300KW/Sとなる．

処理する仕事の内容によって，それぞれのサイクルには名弥がついている．

WCサイクル　CAサイクル　DSAサイクル



1サイクルデータ転送は，DSAサイクルからスタートする．

2サイクルデータ転送は，CAサイクルからスタートする．

3サイクルデータ転送は，WCサイクルからスタートする．

(1) 1CYCLE DATA転送

1CYCLE DATA転送で記憶装置と情報交換する場合に入出力装置側で用意するものとしてメモリのADDRESS指定REGISTER（15BIT）および転送語数を数えるカウンタおよびDSAのインターフェースに必要な信号をそろえる必要がある．

(2) 1CYCLE WRITE DATA転送

1CYCLE WRITE DATA転送は，入出力装置から555Hに情報を転送する．

1CYCLE WRITE DATA転送のTIME CHARTを（**図—23**）に示し，このTIME CHARTによって，1CYCLE WRITE DATA転送の手順を説明する．

① DSA要求信号（BRK RQST）を入力装置から555Hに転送する．

② BRK RQST信号が受付けられると，$\overline{\text{ESn}}$（EQUEPMENT SELECT）が555Hより出力装置へ転送される．入出力装置側は$\overline{\text{ESn}}$信号によってBRK RQST信号をクリヤしなければならない．

③ $\overline{\text{ESn}}$信号をGATE条件としてCYCSEL，$\overline{\text{TRANSF DIR}}$，$\overline{\text{MBINCR}}$およびADDRESS DATAを定める．

CYCSEL，$\overline{\text{MBINCR}}$は$\overline{\text{ESn}}$の立下りからMAX80ns以内（555H側に到達する遅延時間も含む）に確定しなければならない．

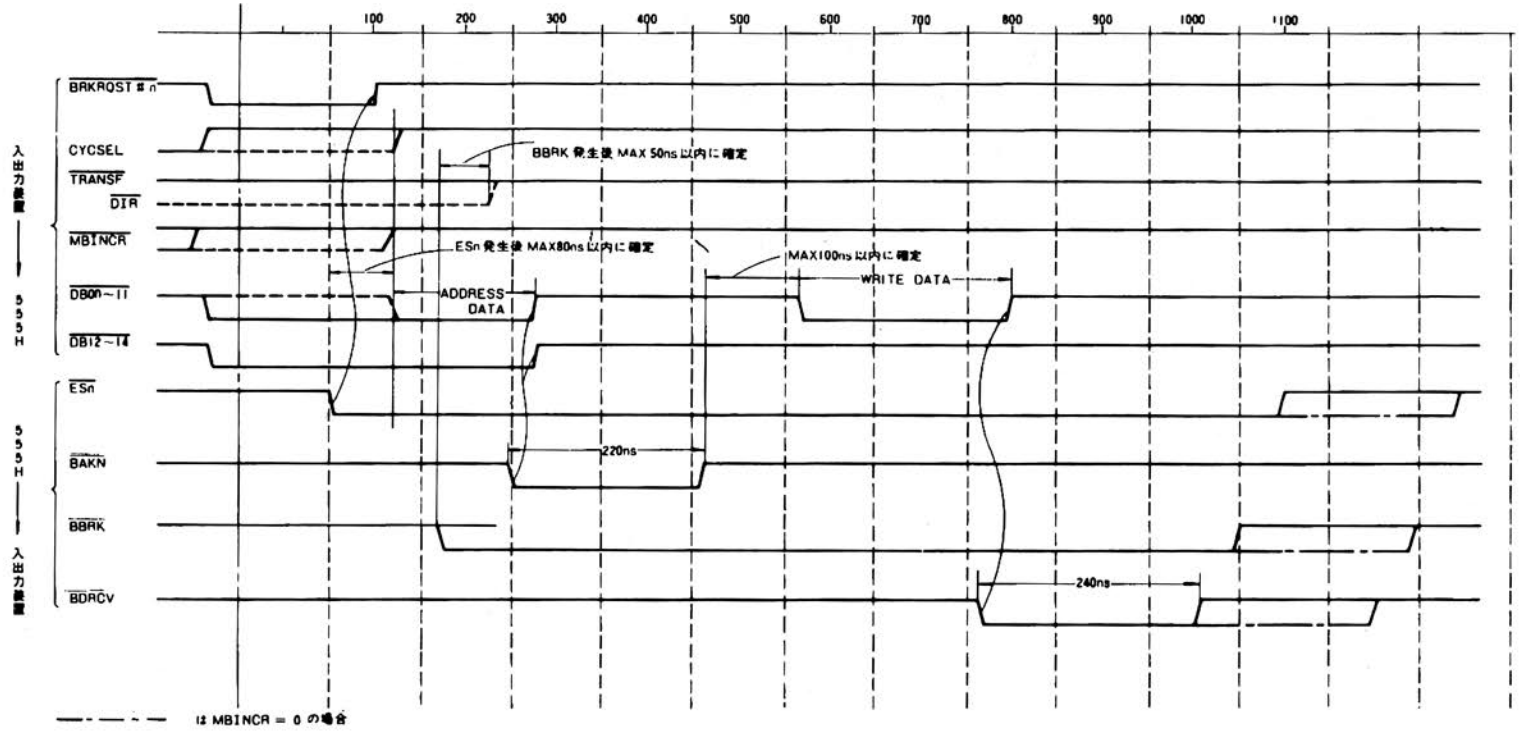
$\overline{\text{TRANSF DIR}}$はBBRK発生後50ns以内に定まればよい．

④ 入出力装置よりADDRESS DATAが転送されると555H側は“受取った”という返事信号として$\overline{\text{BAKN}}$を入出力装置へ転送する．

⑤ 入出力装置は$\overline{\text{BAKN}}$信号の前縁またはレベルでADDRESS DATAのGATE条件をOFFにし，WRITE DATAのGATE条件をONしてWRITE DATAを555Hに転送する．

WRITE DATAは$\overline{\text{BAKN}}$の後縁からMAX100ns以内に確定しなければならない．

⑥ WRITE DATAが555H側に転送されてくるとWRITE DATAを「受取った」という返事信号として$\overline{\text{BDRCV}}$信号を入出力装置へ転送する．入出力装置は$\overline{\text{BDRCV}}$信号によってWRITE DATAのGATE条件をOFFにする．



図－23 1CYCLE WRITE DATA TRANSFER TIME CHART

(3) 1 CYCLE READ DATA転送

1 CYCLE READ DATA転送は555Hから情報を入出力装置に転送する.

2 CYCLE READ DATA転送のTIME CHARTを（**図—24**）に示し，このTIME CHARTによって，1 CYCLE READ DATA転送の手順を示す.

手順の①～④はWRITE DATA転送と同じである.

⑤ 入出力装置は$\overline{\text{BAKN}}$信号の前縁またはレベルでADDRESS DATA BUFFFRのGATE条件をOFFにする. $\overline{\text{BAKN}}$信号の後縁でREAD DATA BUFFFRのGATE条件をONしてREAD DATAが転送されてくるのを待つ.

⑥ 555HからREAD DATAと一緒に$\overline{\text{BDAVL}}$信号が転送される. 入出力装置は$\overline{\text{BDAVL}}$のレベルまたは後縁でREAD DATA REGISTERにセットすればよい.

(4) 2 CYCLE DATA転送

2 CYCLE DATA転送はDATAをメモリの指定されたADDRESSにWRITE（またはREAD）したりするとき，ADDRESS REGISTERとしてメモリの一部を使用する. そのため，メモリのACCESSを2回行う.

1回目の仕事はメモリにSTOREされているADDRESSをREADし，そのREAD DATAをADDRESS DATAとしてDSA REGISTERにセットする（CAサイクルと呼ぶ).

2回目，すなわちBRKサイクルに入って，入出力装置からのWRITE DATA（または555Hから入出力装置へ転送するREAD DATA）を指定されたADDRESSへWRITE（またはREAD）する.

5. 2 CYCLE WRITE DATA転送

2 CYCLE WRITE DATA転送は入出力装置から555Hへ DATAを転送する.

2 CYCLE WRITE DATA転送のTIME CHARTを（**図—25**）に示す. この図によって2 CYCLE WRITE DATA転送の手順を説明する.

① DSA要求信号（$\overline{\text{BRK RQST}}$ ♯n）を入出力装置から555Hに転送する.

② $\overline{\text{BRK RQST}}$ ♯n信号が受付けられると“受付けた”という返事信号$\overline{\text{ESn}}$を555Hから入出力装置へ転送する. 入出力装置側は$\overline{\text{ESn}}$の前縁またはレベルで$\overline{\text{BRKRQST}}$ ♯n信号をクリヤする.

③ $\overline{\text{ESn}}$信号をGATE条件として$\overline{\text{2 CYC}}$, $\overline{\text{CAINCR}}$, $\overline{\text{TRANSF DIR}}$, $\overline{\text{MBINCR}}$およびADDRESS DATAを転送するDATA BUS LINE（DB00～11, DATA12～14, DATA15～17）のGATEをONする.

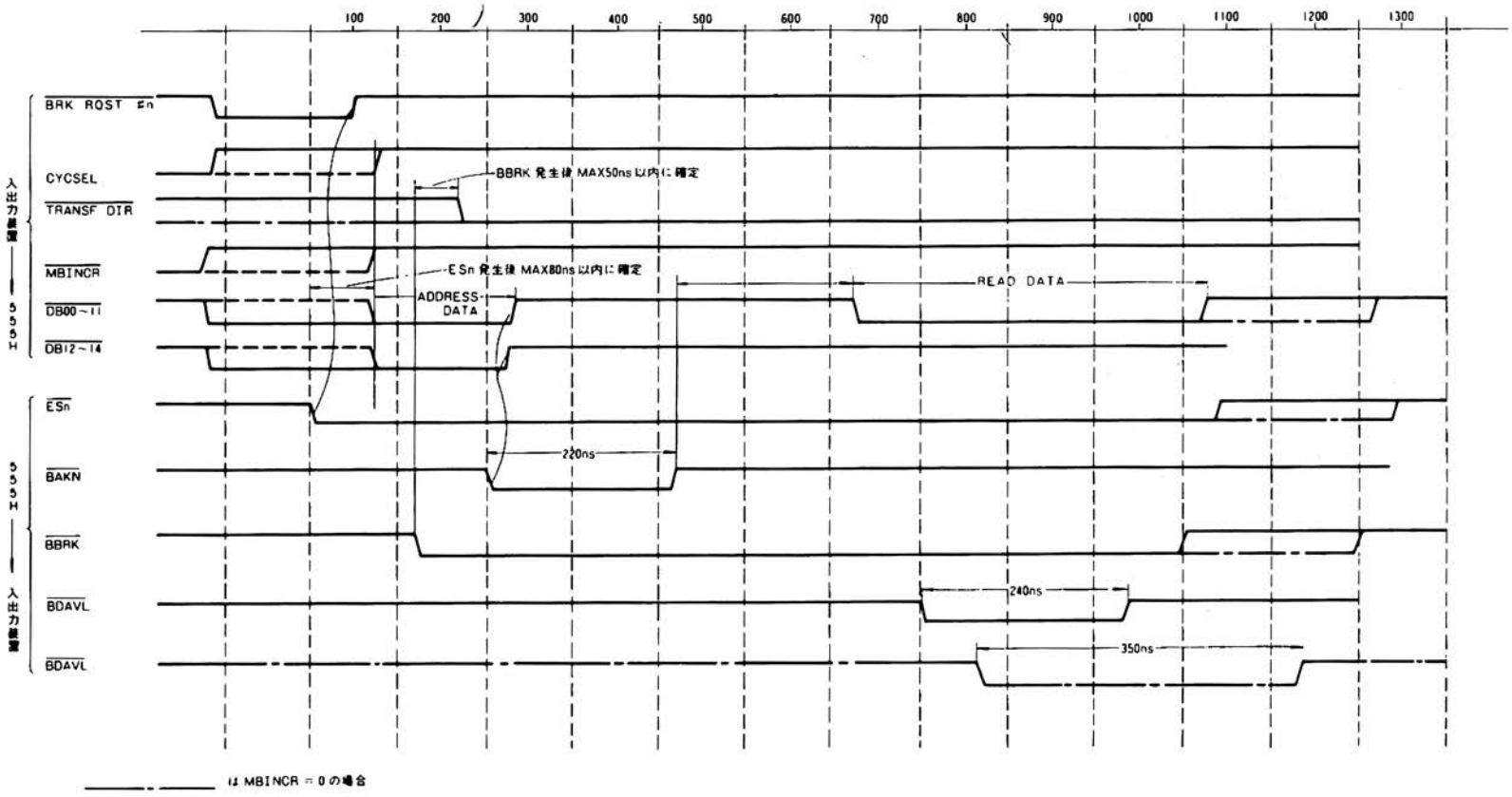
④ 入出力装置からADDRESS DATAが転送されてくると555Hは“受取った”という返事信号$\overline{\text{BAKN}}$信号を入出力装置へ転送する.

⑤ 入出力装置は$\overline{\text{BAKN}}$信号の前縁またはレベルでDATA BUS LINEのGATE条件をOFFにする.

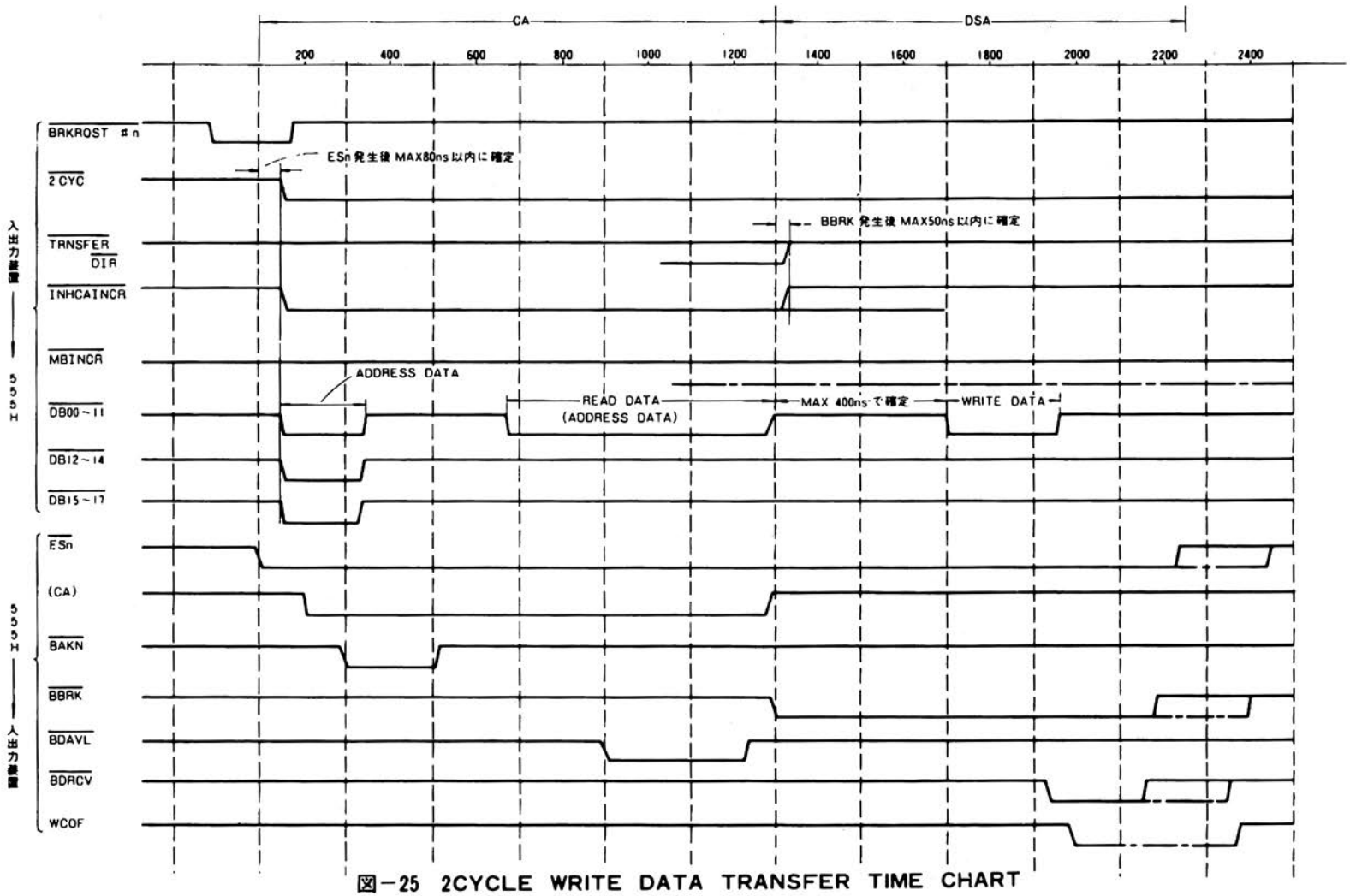
⑥ $\overline{\text{BAKN}}$信号によって受取られたADDRESS DATAによって指定されたメモリから，READ DATAがADDRESS DATAとして555H内のDSA REGISTERにセットされる. このADDRESS DATAをREAD DTAとして入出力装置へ転送する. 入出力装置はこのREAD DATAを使用するかしないかは自由である. $\overline{\text{CAINCR}}$＝0の場合はREAD DATAはINCREMENT（＋1）されてADDRESS DATAとしてDSA RE・REGISTERされると同時に指定されたADDRESSへ再書き込みされる.

⑦ $\overline{\text{BBRK}}$サイクルに入ったら入出力装置は$\overline{\text{BBRK}}$サイクルの前縁またはレベルでDATA BUS LINEのGATE条件をONにする. WRITE DATAは，$\overline{\text{BBRK}}$発生後400ns以内に転送すればよい.

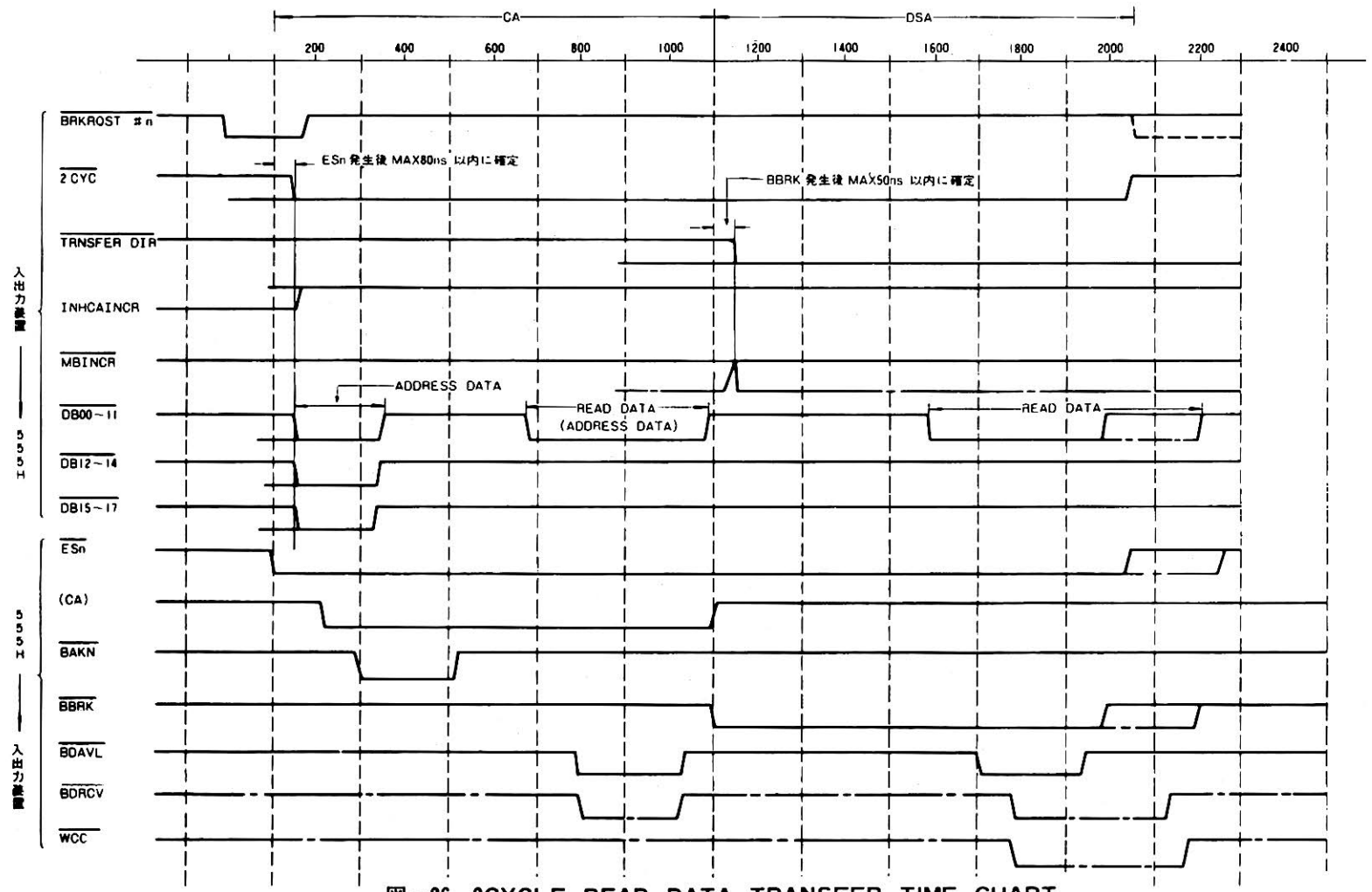
⑧ 555Hは入出力装置から転送されたWRITE DATAを“受取った”という返事信号として$\overline{\text{BDRCV}}$信号を入出力装置に転送する. 入出力装置は$\overline{\text{BDRCV}}$信号によって，DATA BUS LINE



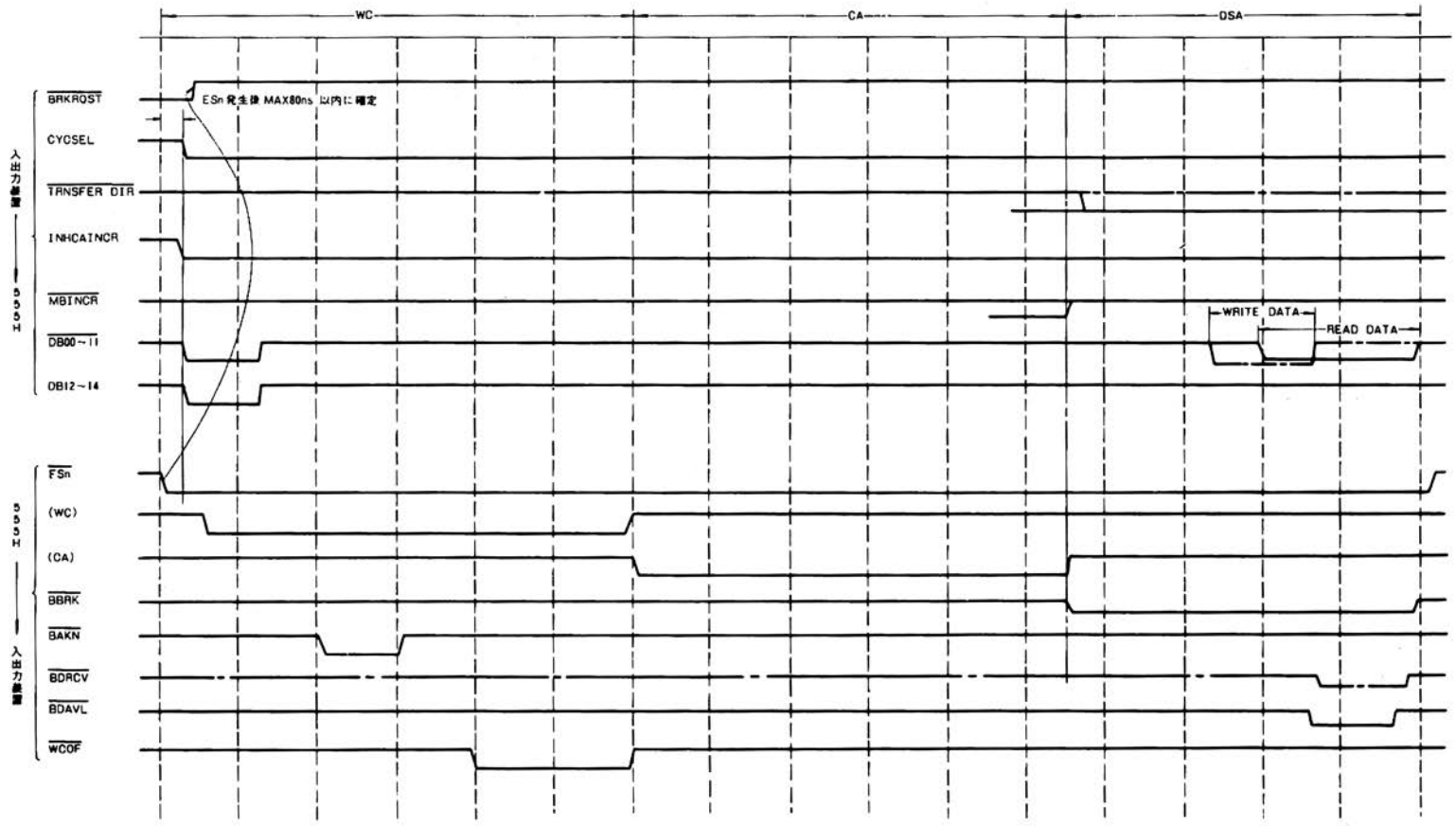
図－24 1CYCLE READ DATA TRANSFER TIME CHART



図－25 2CYCLE WRITE DATA TRANSFER TIME CHART



図－26 2CYCLE READ DATA TRANSFER TIME CHART



図－27 3CYCLE READ DATA(WRITE DATA) TRANSFER TIME CHART

のGATE条件をOFFにする.

$\overline{\text{MBINCR}}$＝0の場合はWRITE DATA＋1されてWRITEされる.

もしINCREMENTして0になればWCOF信号を入出力装置へ転送する.

(6) 3 CYCLE DATA転送

3 CYCLE DATA転送はMEMORYのACCESSを3回行ってDATAを転送する方式である.
転送速度は1 CYCLE DATA転送の1/3になる.

DSA方式によるDATA転送はブロック転送（DATAをブロック単位にまとめ転送）が主であるため，DATAの転送語数を数えるCOUNTERおよびそのDATAをMEMORYの何番地から規則正しく入れる（または読出）ためにはADDRESS DATAが必要となるがこれらの操作は555H側で実行するところに3 CYCLE DATA転送の特徴がある.

3 CYCLE DATA転送は，転送語数COUNTER（WORD COUNTER＝WCと略す）ADDRESS DATA (CURRENT ADDRESS＝CA) は555H MEMORYの特定番地をREGISTERとして用いるためにMEMORYのACCESSは3回必要となり，3回のACCESSする時間帯には次のような名弥がついている.

1回目のACCESSする時間帯をＷＣサイクル

2回目のACCESSする時間帯をＣＡサイクル

3回目のACCESSする時間帯をＤＳＡサイクル

① ＷＣサイクルの仕事

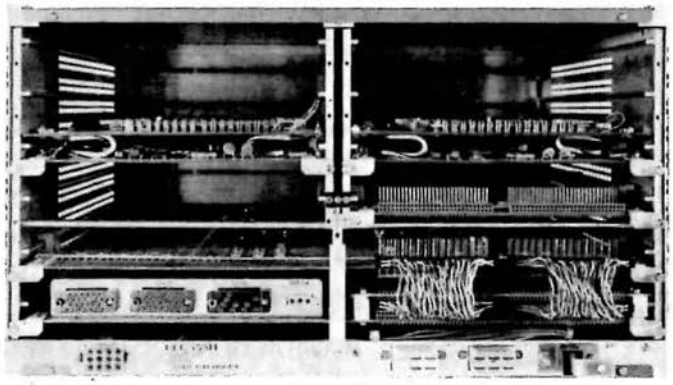
入出力装置から，転送語数COUNTERを指定するADDRESS DATA（特定番地で偶数番地であること・通常7750）が転送され，READ DATA（転送語数）をDMB REGISTERにセットする．そしてINCREMENT（＋1する）して，またMEMORYにWRITEする．INCREMENTした結果DMB REGISTERがOVER FLOWすればWCOF信号を入出力装置に転送する．ここまでがＷＣの仕事である.

② ＣＡサイクルの仕事

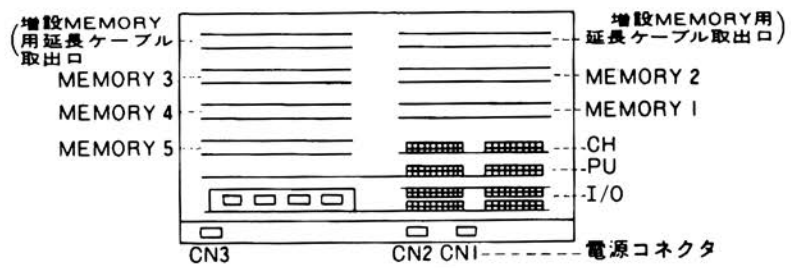
ＣＡサイクルは，ブロック単位で転送されてくるDATAをMEMORYにWRITE（またはREAD）するために特定番地（A＋1番地）にSTOREされているCURRENT ADDRESSをREADしINCREMENT（＋1）して，ADDRESS DATAとしてDMA REGISTERにセットする．ここまでがCAサイクルの仕事である.

③ DSAサイクルの仕事

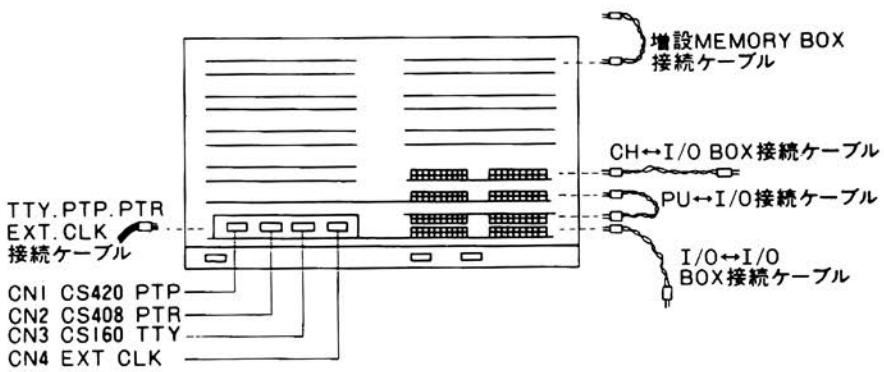
CAサイクルでDMA REGISTERにセットされたADDRESS DATAでMEMORYからのREAD DATA（またはWRITE）をDMB REGISTERにセットし，そして入出力装置に転送する.



555H背面



555H 背 面 図



555H ケーブル接続図



A-MP-I 86671-3(オス)

J1, J2(58PIN) コネクタ

図－28

(1) PU

J1A

0		GND
1		BAC 00
2		GND
3		BAC 01
4		GND
5		BAC 02
6		GND
7		BAC 03
8		GND
9		BAC 04
10		GND
11		BAC 05
12		GND
13		BAC 06
14		GND
15		BAC 07
16		GND
17		BAC 08
18		GND
19		BAC 09
20		GND
21		BAC 10
22		GND
23		BAC 11
24		GND
25		$\overline{\text{BRUN}}$
26		GND
27		BINIT
28		GND

J2A

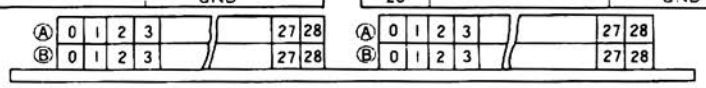
0		GND
1		BMB 00
2		GND
3		BMB 01
4		GND
5		BMB 02
6		GND
7		BMB 03
8		GND
9		BMB 04
10		GND
11		BMB 05
12		GND
13		BMB 06
14		GND
15		BMB 07
16		GND
17		BMB 08
18		GND
19		BMB 09
20		GND
21		BMB 10
22		GND
23		BMB 11
24		GND
25		
26		GND
27		BE
28		GND

J1B

0		GND
1		
2		GND
3		
4		GND
5		
6		GND
7		
8		GND
9		
10		GND
11		
12		GND
13		
14		GND
15		
16		GND
17		
18		GND
19		
20		GND
21		
22		GND
23		
24		GND
25		
26		GND
27		
28		GND

J2B

0		GND
1		
2		GND
3		
4		GND
5		
6		GND
7		
8		GND
9		
10		GND
11		
12		GND
13		
14		GND
15		
16		GND
17		
18		GND
19		
20		GND
21		
22		GND
23		
24		GND
25		
26		GND
27		
28		GND



入出力コネクタ信号表

(2) CH

J1A

0		GND
1		RQST #0
2		GND
3		RQST #1
4		GND
5		RQST #2
6		GND
7		RQST #3
8		GND
9		RQST #4
10		GND
11		
12		GND
13		ESn (0)
14		GND
15		ESn (1)
16		GND
17		ESn (2)
18		GND
19		ESn (3)
20		GND
21		ESn (4)
22		GND
23		
24		GND
25		BTS 1
26		GND
27		BTS 3
28		GND

J2A

0		GND
1		BAKN
2		GND
3		BWCOF
4		GND
5		BBRK
6		GND
7		BDAVL
8		GND
9		BDRCV
10		GND
11		TRANSF DIR
12		GND
13		MBINCR
14		GND
15		INRCAINCR
16		GND
17		CYC SEL
18		GND
19		
20		GND
21		
22		GND
23		DB 12
24		GND
25		DB 13
26		GND
27		DB 14
28		GND

J1B

0		GND
1		DB 00
2		GND
3		DB 01
4		GND
5		DB 02
6		GND
7		DB 03
8		GND
9		DB 04
10		GND
11		DB 05
12		GND
13		DB 06
14		GND
15		DB 07
16		GND
17		DB 08
18		GND
19		DB 09
20		GND
21		DB 10
22		GND
23		DB 11
24		GND
25		
26		GND
27		
28		GND

J2B

0		GND
1		
2		GND
3		
4		GND
5		
6		GND
7		
8		GND
9		
10		GND
11		
12		GND
13		
14		GND
15		
16		GND
17		
18		GND
19		
20		GND
21		
22		GND
23		
24		GND
25		
26		GND
27		
28		GND

入出力コネクタ信号表

(3) I/O

J1A

0		GND
1		BAC 00
2		GND
3		BAC 01
4		GND
5		BAC 02
6		GND
7		BAC 03
8		GND
9		BAC 04
10		GND
11		BAC 05
12		GND
13		BAC 06
14		GND
15		BAC 07
16		GND
17		BAC 08
18		GND
19		BAC 09
20		GND
21		BAC 10
22		GND
23		BAC 11
24		GND
25		$\overline{\text{BRUN}}$
26		GND
27		BINIT
28		GND

J2A

0		GND
1		BMB 00
2		GND
3		BMB 01
4		GND
5		BMB 02
6		GND
7		BMB 03
8		GND
9		BMB 04
10		GND
11		BMB 05
12		GND
13		BMB 06
14		GND
15		BMB 07
16		GND
17		BMB 08
18		GND
19		BMB 09
20		GND
21		BMB 10
22		GND
23		BMB 11
24		GND
25		
26		GND
27		BE
28		GND

J1B

0		GND
1		$\overline{\text{AC00BUS}}$
2		GND
3		$\overline{\text{AC01BUS}}$
4		GND
5		$\overline{\text{AC02BUS}}$
6		GND
7		$\overline{\text{AC03BUS}}$
8		GND
9		$\overline{\text{AC04BUS}}$
10		GND
11		$\overline{\text{AC05BUS}}$
12		GND
13		$\overline{\text{AC06BUS}}$
14		GND
15		$\overline{\text{AC07BUS}}$
16		GND
17		$\overline{\text{AC08BUS}}$
18		GND
19		$\overline{\text{AC09BUS}}$
20		GND
21		$\overline{\text{AC10BUS}}$
22		GND
23		$\overline{\text{AC11BUS}}$
24		GND
25		$\overline{\text{ACCLR BUS}}$
26		GND
27		$\overline{\text{ACLOAD BUS}}$
28		GND

J2B

0		GND
1		
2		GND
3		
4		GND
5		
6		GND
7		
8		GND
9		
10		GND
11		
12		GND
13		BIOP 4
14		GND
15		BIOP 2
16		GND
17		BIOP 1
18		GND
19		$\overline{\text{EXTIOT BUS}}$
20		GND
21		$\overline{\text{SKP BUS}}$
22		GND
23		$\overline{\text{INTRQST BUS}}$
24		GND
25		$\overline{\text{INTRQST1 BUS}}$
26		GND
27		
28		GND

CN 1 (CS420)

A		Ch 1
C		GND
B		Ch 2
D		GND
E		Ch 3
H		GND
F		Ch 4
J		GND
K		Ch 5
M		GND
L		Ch 6
N		GND
P		Ch 7
S		GND
R		Ch 8
T		GND
U		Ch 9
W		GND
V		SD
X		GND
Y		PI
d		GND
z		PR
b		GND
c		TL
d		GND

CN 2 (CS408)

A		$\overline{\text{FEED}}$
C		GND
B		$\overline{\text{TAPE END}}$
D		GND
E		$\overline{\text{a 1}}$
H		GND
F		$\overline{\text{a 2}}$
J		GND
K		$\overline{\text{a 3}}$
M		GND
L		$\overline{\text{a 4}}$
N		GND
P		$\overline{\text{a 5}}$
S		GND
R		$\overline{\text{a 6}}$
T		GND
U		$\overline{\text{a 7}}$
W		GND
V		$\overline{\text{a 8}}$
X		GND
Y		$\overline{\text{STOP}}$
d		GND
Z		$\overline{\text{START}}$
b		GND
c		$\overline{\text{R ON LINE}}$
d		GND

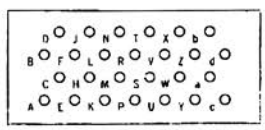
CN 3 (CS100)

A		FROM TTY
C		GND
B		
D		
E		
H		
F		+5V POWER
J		GND
K		TRP
M		GND
L		
N		
P		
S		
R		
T		
U		TO TTY
W		GND
V		
X		
Y		
d		
Z		
b		
c		$\overline{\text{TTY ON}}$
d		GND

CN 4

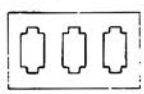
1		EXT CLK
2		
3		GND

CN 1 ～CN 3



AMP-200512-2 (筐体)
AMP-200513-2 (ケーブル)

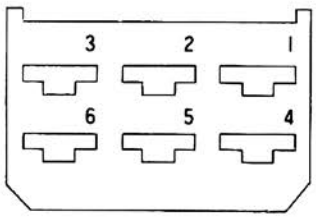
CN 4



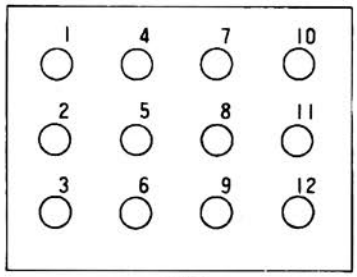
1-480304-0 (筐体)
1-480305-0 (ケーブル)

入出力コネクタ信号表

電源コネクタ信号表



CN1, CN2
(型名 メス480195 オス 480194)



CN3
(型名 メス 480275 オス 480278)

ピン番号	CN1	ピン番号	CN2
1	+5V	1	+22V
2	+5V	2	+12V
3	−5V	3	
4	LGND	4	MGND 1(+22VRTN)
5	LGND	5	MGND 2(+12VRTN)
6	−12V	6	

(LGND：ロジックグランド　MGND：メモリグランド)

ピン番号	CN3	ピン番号	CN3
1	$\overline{\text{PPC}}$	7	POWER ALM
2	$\overline{\text{LPC}}$	8	
3	$\overline{\text{LPCD}}$	9	EARTH
4	$\overline{\text{MPC}}$	10	POWER ON (LAMP)
5	$\overline{\text{AST}}$	11	POWER (SWITCH)
6	LGND	12	CCOM

❷ 555A超小型電子計算機 ハードウェア

1. 概　要

CEC555Aミニコンピュータは，新時代の多様化するニーズに応えるため設計された，集約的構造をもつ，低価格，高速演算処理を可能とする新鋭機である．

特にメンティナンス，オペレーション機能を重視し，1枚の大型基板に8K語固定メモリ，DSAチャネル1チャネル，高速紙テープリーダコントロール，高速紙テープパンチャコントロール，システムコンソール（TTY）コントロールおよびIPL（イニシアルプログラムローダ），ACによるモデファイ可能なINDEX　REGISTER 4個を内蔵し，かつ外部より供給される電源は＋5V8A1系統とし，他の必要とする電圧は内部においてDC—DCコンバータにより得る方式をとり，内部の配線，必要最少限のコネクタ方式で，大幅なコストの低減化と，すぐれた信頼性を確保している．

また，DSAチャネルによりDRISK—2002磁気ディスク装置（3.2Mワード），CFD—300ミニディスク装置（256Kワード），CFP—500フロッピーディスク（130Kワード），CMT—8磁気テープ装置（2400フィート）などの補助記憶装置との接続を可能としており，超小型ながらEDPシステムの演算処理部，あらゆる自動化省力化装置の演算制御装置として広範な分野で長期にわたり利用することができる．



2. 構　成

555Aの基本構成は，処理装置部（PU），記憶部（MU），低速入出力制御部（IO），高速入出力制御部（CH）であり，各部を1枚の大型基板上に実装した方式である．

(1) ＭＵ部：スタティック型ICメモリ８Ｋ語実装

(2) ＣＨ部：直接メモリと情報交換をする時，CH部を経由し外部記憶装置との間でデータの交換を行う．

(3) ＩＯ部：プログラム制御による低速入出力データ転送は，すべてIO　BUSを経由して行う．PTR，PTP，システムコンソール（CS—100，CS—1200Ｂ，CMD—４，ＣＭＤ—８）のインタフェースが標準実装されている．

(4) ＰＵ部：並列演算回路を持ち，0.68μsのサイクルタイムによる高速演算処理ができる．

3. 性　能

(1) 記憶装置（MU部）

素子：低電力型スタティックICメモリ

語長：12ビット

サイクルタイム：0.68μs

容量：８Ｋ語固定

(2) 処理装置（PU部）

演算方式：並列２進法，２の補数表示，固定小数点

命令数：基本命令８種類

演算速度：加算2.72μs　減算4.08μs

アドレス間接修飾2.04μs

アドレス方式：直接，間接，インデックス

インデックスレジスタ：４個（ACの内容による修飾可能）

オートインデックス：８（記憶装置上）

割込み：１レベル

(3) 入出力制御

低速入出力制御（IO）

接続チャネル数：64

転送速度：50kHz

高速入出力制御

２種類（１サイクル，３サイクル）

接続チャネル数：１チャネル

転送速度：１サイクル　500kHz

３サイクル　210kHz

(4) 設置条件

周囲温度：０～40℃

湿度：35～85%

電源：100Ｖ±10Ｖ

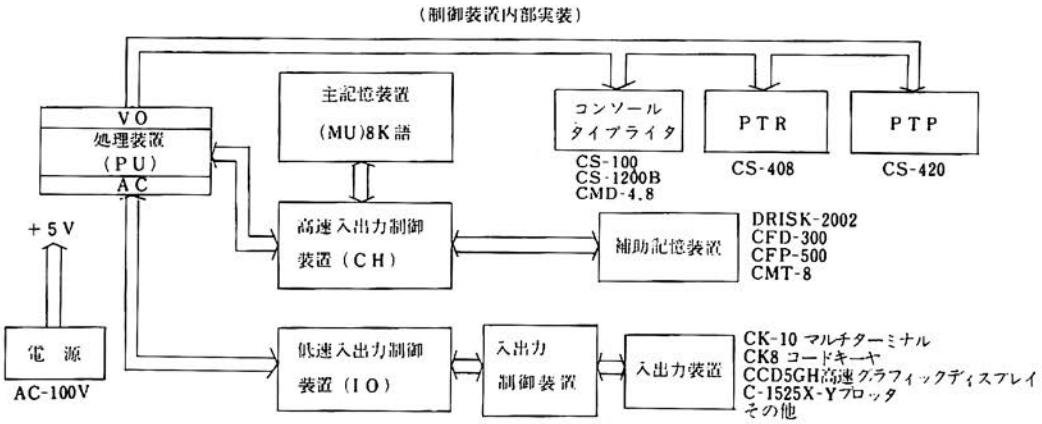
(5) 寸法

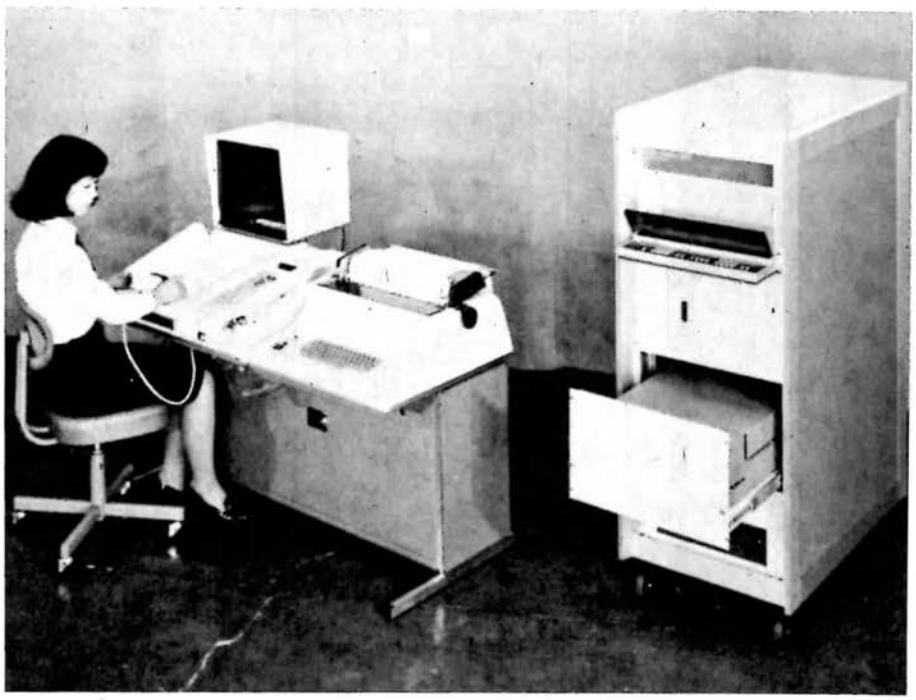
幅：430㎜

高さ：52㎜

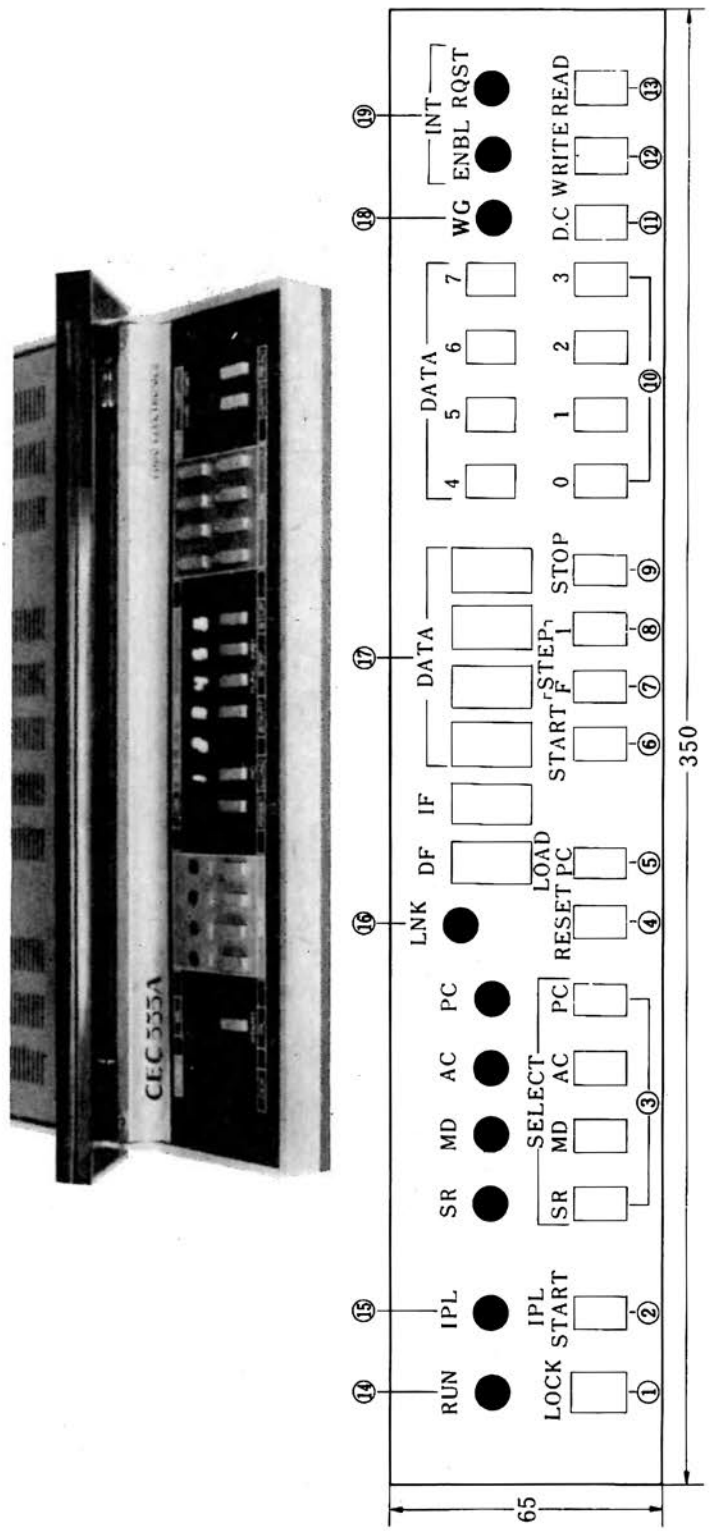
奥行：650㎜

4. 基本構成図





5. オペレータコンソール上の制御・インジケータ



パネルスイッチ群図

番号	名　称	内　容
①	LOCK	スライドスイッチを手前に引くとKEY LOCKがきく. ただし, ③, ④, ⑤, ⑥, ⑬はLOCKされない. 元に戻すとLOCKは解除される.
②	IPL START	INITIAL PROGRAM LOADERをKEYによりSTARTさせることができる.
③	SELECT	DATA表示の内容をSELECTし, SELECTされていることを示すためにKEYの上にある緑色の発光ダイオードが点灯する. SR……SWITCH REGISTERが選ばれる. MD……MEMORY DATAが選ばれる. AC……ACCUMULATORが選ばれる. PC……PROGRAM COUNTERが選ばれる.
④	RESET	RUN以外のFFをすべてCLRする (KEY INITIALIZE).
⑤	PC LOAD	SWITCH REGISTERの内容をPROGRAM COUNTERにLOADする.
⑥	START	現在の状態でそのまま続行する. RUNはSETされるがINITIALIZEはされないので, 初期状態からSTARTするときはRESETを押してからSTARTを押す.
⑦	FUNC STEP	INSTRUCTIONの各STATUSごとRUNがSTOPする. 保守用のKEYであるため通常は使用してはならない.
⑧	INST STEP	1つのINSTRUCTIONを実行し終った後にRUNはSTOPする.
⑨	STOP	その時実行しているINSTRUCTIONの終りでRUNはSTOPする.
⑩	DATA	0～7までの8進ディジ スイッチで1回押すごとにSWITCH REGISTERのLSDから入っていき, 次々にSHIFTされ, OVER FLOWは無視される.
⑪	D.C	DATA CLEARの略でSWITCH REGISTERの内容をCLEARする. PCで指定されたメモリのアドレスにSWITCH REGISTERの内容を書き込む. PCは＋1 INCREMENTされる. MDには表示されない.
⑫	WRITE	PCで指定されたメモリのアドレスから読み出してくる. 表示はMDに示される.
⑬	READ	PCで指定されたメモリのアドレスから読み出してくる. 表示はMDに示される.
⑭	RUN	計算機がRUNしていることをINDICATEするために赤い発光ダイオードが点灯する.
⑮	IPL	INITIAL PROGRAM LOADERが動作していることをINDICATEするために赤い発光ダイオードが点灯する.
⑯	LNK	リンクをINDICATEするために赤い発光ダイオードが点灯する.
⑰	DISPLAY	DFはデータフィールド. IFはインストラクションフィールドを表示する. DATAは③でSELECTされた緑色の発光ダイオードが点灯しているREGISTERを表示する.
⑱	W.G	WATCH DOGの略で計算機が正常に動作をしていることを監視するためのTIMER INDICATORで赤い発光ダイオードが点灯する.
⑲	INT	INTERRUPT REQUESTおよびINTERRUPTENABLEを表示する.

6. 命令およびコード表

1. メモリアドレスを含む命令

記　号	8進コード	実　行　内　容
A N D	0××××	ACの内容と指定したアドレスをもつロケーションの内容間でビットごとの論理積をとる演算が行なわれる。結果はACに立つ。指定したロケーションの内容はもとのままである。
T A D	1××××	指定したアドレスをもつロケーションの内容がACの内容に，2の補数演算で加算される。結果はACに立つ。指定したロケーションの内容はもとのままである。
I S Z	2××××	指定したアドレスをもつロケーションの内容に2の補数演算で1を加える。その結果が0であればPCの内容を1だけ増す。つまり次の命令をスキップする。結果が0でなければ次の命令に進む。何れの場合もACの内容は変らない。
D C A	3××××	ACの内容を指定したアドレスをもつロケーションにストアする。ACはクリアされる。もとのロケーションの内容は失われる。
J M S	4××××	PCの内容を1増し，それを指定したアドレスをもつロケーションにストアし，次の命令を指定したアドレス＋1のロケーションからとる。ACの内容は変らない。
J M P	5××××	次の命令を指定したアドレスをもつロケーションからとる。

2. メモリアドレスを含まない演算命令

記　号	8進コード	実　行　内　容
N O P	7000	無効命令。
I A C	7001	ACの内容は2の補数演算で1増される。
R A L	7004	ACとLの内容は左へ1つシフトされる。
R T L	7006	左へ2つシフトされる。RALの命令を2つ実行したのと同じ。
R A R	7010	ACとLの内容は1つ右へシフトされる。
R T R	7012	右へ2つシフトされる。RARの命令を2つ実行したのと同じ。
C M L	7020	Lの補数をとる。
C M A	7040	ACの内容が現在の内容の1の補数にセットされる。
C I A	7041	ACの2の補数をとる。
C L L	7100	Lをクリアする。
CLL RAL	7104	Lをクリアし，ACとLの内容を1つ左へシフトする。
CLL RTL	7106	Lをクリアし，2つ左へシフトする。
CLL RAR	7110	Lをクリアし，ACとLの内容を1つ右へシフトする。
CLL RTR	7112	Lをクリアし，2つ右へシフトする。
S T L	7120	Lを1にセットする。
C L A	7200	ACをクリアする。
CLA IAC	7201	ACを1にセットする。
G L K	7204	LをAC11に移す。
CLA CLL	7300	ACとLをクリアする。
S T A	7240	ACを－1にセットする。

3. 入出力命令

種類	記号	8進コード	実行内容
インターラプト	ＩＯＮ	６００１	プログラムインタラプト要求に応じる。
	ＩＯＦ	６００２	プログラムインタラプト要求を無視する。
ハイスピードリーダ	ＲＳＦ	６０１１	フラグが１のとき次の命令をスキップする。
	ＲＲＢ	６０１２	バッファの内容を読み取りフラグをクリアする。
	ＲＦＣ	６０１４	フラグとバッファをクリアし，紙テープから１語リードし，バッファに入れフラグをセットする。
テレタイプキーボードリーダ	ＫＳＦ	６０３１	フラグが１のとき次の命令をスキップする。
	ＫＣＣ	６０３２	ＡＣとフラグをクリアする。
	ＫＲＳ	６０３４	バッファの内容をＡＣ４－11に読み取る。
	ＫＲＢ	６０３４	ＡＣとフラグをクリアし，バッファの内容をＡＣ４－11に読み取る。
テレタイププリンタパンチ	ＴＳＦ	６０４１	フラグが１のとき次の命令をスキップする。
	ＴＣＦ	６０４２	フラグをクリアする。
	ＴＰＣ	６０４４	ＡＣ４－11の内容を印刷またはせん孔する。
	ＴＬＳ	６０４６	フラグをクリアし，ＡＣ４－11の内容を印刷またはせん孔する。
メモリーパリティ	ＳＭＰ	６１０１	エラーフラグが０のとき次の命令をスキップする。
	ＷＧ	６１０４	ＣＰＵがＤＯＷＮしたことを周辺装置に警報する。
増設メモリー	ＣＤＦ	６２ｎ１	データフィールドをｎに変える。
	ＣＩＦ	６２ｎ２	インストラクションフィールドをｎに変える。
	ＲＤＦ	６２１４	データフィールドをＡＣ７－８に読み込む。
	ＲＩＦ	６２２４	インストラクションフィールドをＡＣ７－８に読み込む。
	ＲＭＦ	６２４４	もとのメモリーフィールドに戻す。
	ＲＩＢ	６２３４	インターラプトバッファをＡＣ７－８およびＡＣ10－11に読みとる。
インディックスレジスタ		６２ｎ６	インディックスレジスタの書き込み
		６６ｎ６	インディックスレジスタの読出し。
		６２ｎ７	アドレス修飾
ＩＢＭ	ＴＩＳＦ	６３１１	フラグが１ならばスキップする。
タイプライタ	ＴＩＣＦ	６３１２	フラグをクリアする。
	ＴＩＰＣ	６３１４	ＡＣ４－11を印刷する。印刷後フラグが１になる。
ＡＤ変換器	ＡＤＳ	６３０１	フラグが１のときスキップする。
	ＡＤＣ	６３０２	フラグをクリアし，Ａ－Ｄ変換器をスタートする。変換終了後，フラグを１にする。
	ＡＤＲ	６３０４	Ａ－Ｄ変換器の内容をＡＣ０－11に読み取る。

ディスクメモリー（32K）	DCMA	6601	ディスクのアドレスレジスタ，パリティエラーフラグ，データ転送完了フラグ，ブレークリクエストフラグ，インタラプトフラグをクリアする。
	DMAR	6603	ディスクメモリーアドレスレジスタをロードし，ディスクの内容をメインメモリーへ転送する。
	DMAW	6605	ディスクメモリーアドレスレジスタをロードし，メインメモリーの内容をディスクへ転送する。
	DCEA	6611	ディスクエクステンディッドアドレスレジスタとメモリーアドレスエクステンションレジスタをクリアする。
	DSAC	6612	アドレス確認フラグが1のとき次の命令をスキップする。
	DEAL	6615	ディスクエクステンディッドアドレスとメモリーアドレスエクステンションレジスタをクリアし，ACの内容を入れる。
	DEAC	6616	ディスクエクステンディッドアドレスレジスタの内容を読みとる。
	DFSE	6621	エラーフラグの内容が0のとき次の命令をスキップする。
	DFSC	6622	データ転送完了フラグが1のとき次の命令をスキップする。
	DMAC	6626	ディスクアドレスレジスタの内容を読みとる。

4. プログラムの制御に関する命令

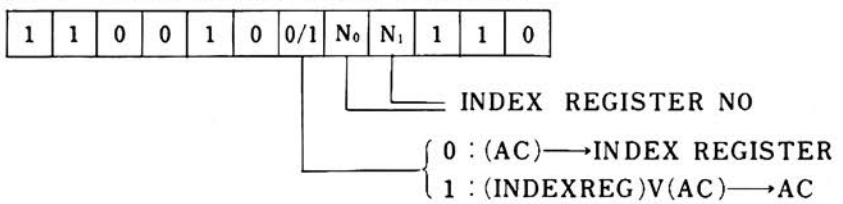
記　号	8進コード	実　行　内　容
HLT	7402	プログラムを停止する。
OSR	7404	DATAの内容とACの内容との間でビットごとに論理和をとるORが行なわれる。その結果はACに残る。
SKP	7410	無条件スキップ。その次の命令はスキップされる。
SNL	7420	L≠0のとき次の命令をスキップする。
SZL	7430	L＝0のとき次の命令をスキップする。
SZA	7440	AC＝0のとき次の命令をスキップする。
SNA	7450	AC≠0のとき次の命令をスキップする。
SZA SNL	7460	AC＝0またはL＝1のとき次の命令をスキップする。
SNA SZL	7470	AC≠0でL＝0のとき次の命令をスキップする。
SMA	7500	AC＜0のとき次の命令をスキップする。
SPA	7510	AC≧0のとき次の命令をスキップする。
SMA SNL	7520	AC＜0またはL＝1のとき次の命令をスキップする。
SPA SZL	7530	AC≧0でL＝0のとき次の命令をスキップする。
SMA SZA	7540	AC≦0のとき次の命令をスキップする。
SPA SNA	7550	AC＞0のとき次の命令をスキップする。

CLA	7600	ACをクリアする。
LAS	7604	DATAの内容をACに入れる。
SZA CLA	7640	AC＝0のとき次の命令をスキップし，それからACをクリアする。
SNA CLA	7650	AC≠0のとき次の命令をスキップし，それからACをクリアする。
SMA CLA	7700	AC＜0のとき次の命令をスキップし，それからACをクリアする。
SPA CLA	7710	AC≧0のとき次の命令をスキップし，それからACをクリアする。

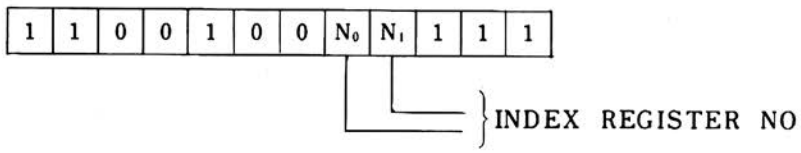
※印は従来の555シリーズに新たに追加された命令である.

4つのINDEX REGISTERが追加されたことにより，下記のような命令が追加された.

(1) 62N6 READ／WRITE INDEX REGISTER



(2) 62N7 MEMORY THE ADDRESS PART OF THE NEXT INSTRUCTION



注1） 62N7は次の命令と1対2の意味をもつ.

注2） 62N7と次の命令との間でINTERRUPTまたはDMA CYCLEに入ることはない.

注3） 62N7の次の命令はAND，TAD，ISZ，DCA，JMSまたはJMPに限る.

注4） INDEX REGISTERの内容とADDRESS部の和がOVER FLOWしたとき，OVER FLOW BITは捨てられる.

ex)

```
        6246    ／INDEX REGISTER #4 READ
                ／(XR #4)V(AC)⟶AC
        6216    ／(XR #1) WRITE
                ／(AC) ⟶ XR #1
1命令と 6207    ／XR #0
等価    AND Y   ／((XR #0)+Y)∧(AC)⟶AC
```

WATCH　DOG

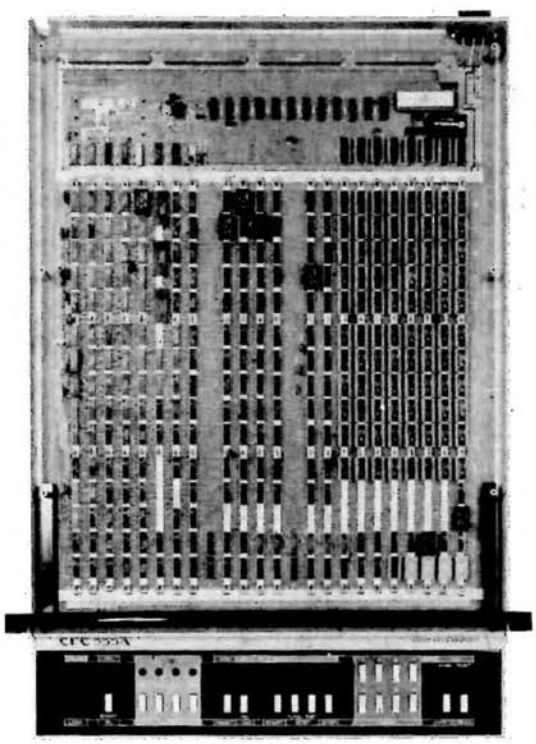
CPUがDOWNしたことを周辺装置に警報するためのものである.

IOT104（6104）の命令が周期的（10ms）に来る間はH　LEVELを保つ.

WATCH　DOGがH　LEVELの間は赤い発光ダイオートが点灯する.

7. INSTRUCTION実行時間

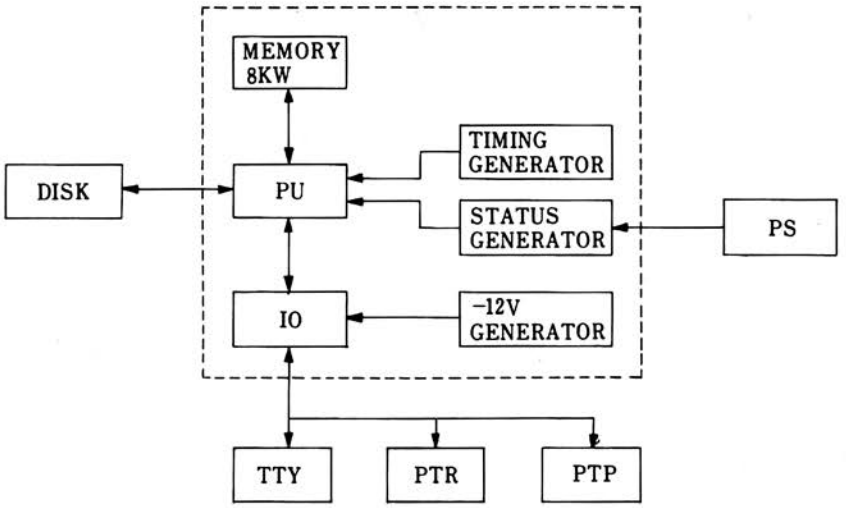
	DIRECT	INDIRECT	AUTO INDEX
AND TAD DCA	1.4μs	2.1	2.8
ISZ	2.1	2.8	3.6
JMS	1.9	2.7	3.4
JMP	1.2	1.9	2.7
OP	1.4		
IOT	2.1		3.6
3 CYCLE	3.9		
1 CYCLE	2.1		



8. 555A インタフェース信号表

	低速 I/O BUS				D M A			
	J1−A	J1−B	J2−A	J2−B	J3−A	J3−B	J4−A	J4−B
0	GND	GND	GND	GND	GND	GND	GND	GND
1	BAC00	$\overline{\text{AC00BUS}}$	BMB00	$\overline{\text{BMB03}}$	$\overline{\text{DADD00}}$	BMB00	$\overline{\text{DATA00}}$	$\overline{\text{BRKRQST}}$
2	GND	GND	GND	GND	GND	GND	GND	GND
3	BAC01	$\overline{\text{AC01BUS}}$	BMB01	$\overline{\text{BMB04}}$	DADD01	BMB01	$\overline{\text{DATA01}}$	$\overline{\text{DIR}}$
4	GND	GND	GND	GND	GND	GND	GND	GND
5	BAC02	$\overline{\text{AC02BUS}}$	BMB02	$\overline{\text{BMB05}}$	$\overline{\text{DADD02}}$	BMB02	$\overline{\text{DATA02}}$	$\overline{\text{MBINC}}$
6	GND	GND	GND	GND	GND	GND	GND	GND
7	BAC03	$\overline{\text{AC03BUS}}$	BMB03	$\overline{\text{BMB06}}$	$\overline{\text{DADD03}}$	BMB03	$\overline{\text{DATA03}}$	$\overline{\text{CYCLE}}$
8	GND	GND	GND	GND	GND	GND	GND	GND
9	BAC04	$\overline{\text{AC04BUS}}$	BMB04	$\overline{\text{BMB07}}$	$\overline{\text{DADD04}}$	BMB04	$\overline{\text{DATA04}}$	$\overline{\text{CAINC}}$
10	GND	GND	GND	GND	GND	GND	GND	GND
11	BAC05	$\overline{\text{AC05BUS}}$	BMB05	$\overline{\text{BMB08}}$	$\overline{\text{DADD05}}$	BMB05	$\overline{\text{DATA05}}$	BTS 1
12	GND	GND	GND	GND	GND	GND	GND	GND
13	BAC06	$\overline{\text{AC06BUS}}$	BMB06	BIOP 4	$\overline{\text{DADD06}}$	BMB06	$\overline{\text{DATA06}}$	BTS 3
14	GND	GND	GND	GND	GND	GND	GND	GND
15	BAC07	$\overline{\text{AC07BUS}}$	BMB07	BIOP 2	$\overline{\text{DADD07}}$	BMB07	$\overline{\text{DATA07}}$	$\overline{\text{BBRK}}$
16	GND	GND	GND	GND	GND	GND	GND	GND
17	BAC08	$\overline{\text{AC08BUS}}$	BMB08	BIOP 1	$\overline{\text{DADD08}}$	BMB08	$\overline{\text{DATA08}}$	$\overline{\text{ADDAC10T}}$
18	GND	GND	GND	GND	GND	GND	GND	GND
19	BAC09	$\overline{\text{AC09BUS}}$	BMB09	WATCH-DOG	$\overline{\text{DADD09}}$	BMB09	$\overline{\text{DATA09}}$	$\overline{\text{BWCOF}}$
20	GND	GND	GND	GND	GND	GND	GND	GND
21	BAC10	$\overline{\text{AC10BUS}}$	BMB10	$\overline{\text{SKIPBUS}}$	$\overline{\text{DADD10}}$	BMB10	$\overline{\text{DATA10}}$	
22	GND	GND	GND	GND	GND	GND	GND	GND
23	BAC11	$\overline{\text{AC11BUS}}$	BMB11	$\overline{\text{INTREQST}}$	$\overline{\text{DADD11}}$	BMB11	$\overline{\text{DATA11}}$	
24	GND	GND	GND	GND	GND	GND	GND	GND
25		$\overline{\text{ACLLRBUS}}$		$\overline{\text{BRUN}}$				$\overline{\text{FXDATA02}}$
26	GND	GND	GND	GND	GND	GND	GND	GND
27			BIOPX	BINITIALIZE				
28	GND	GND	GND	GND	GND	GND	GND	GND

9. 全体構造図


MEMORY
8KW
TIMING
GENERATOR
DISK
PU
STATUS
GENERATOR
PS
IO
−12V
GENERATOR
TTY
PTR
PTP

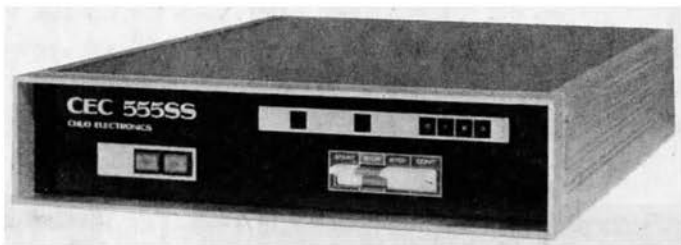
❸ 555SSマイクロプロセッサ

1. 概　要

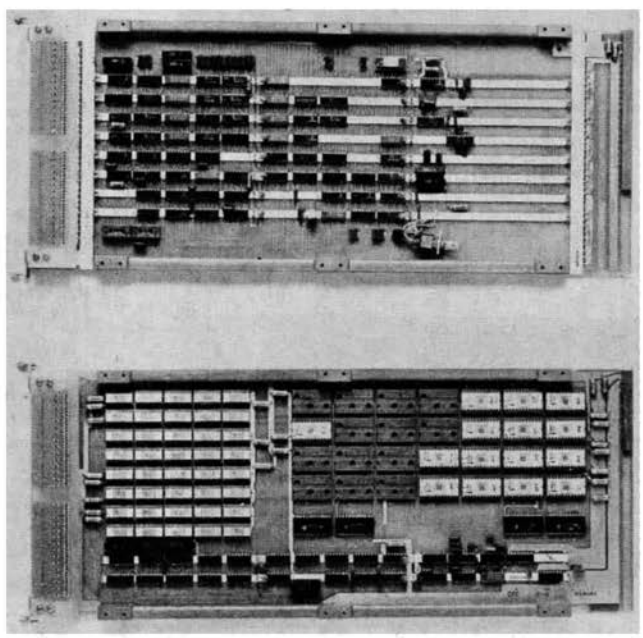
CEC 555 SSマイクロコンピュータは，超小型のコンピュータで，555のターミナルコンピュータとして，また種々の周辺装置や測定器，分析器，複雑な制御装置，省力化機械等に接続してインテリジェントな機器を構成できるように設計されている．マイクロコンピュータは，1語8ビットで，メモリは最大16Kバイトまで，直接アドレシングできる．

555 SS—Iの構成は，処理装置（PU）と記憶装置（MEMORY），555 SS—ⅡはPU，MEMORY，I／OP，E・Mから構成されており，1つのMEMORYに対して4台のPUを接続することができる．MEMORYの情報のやりとりは，すべてバスラインを通して行う．

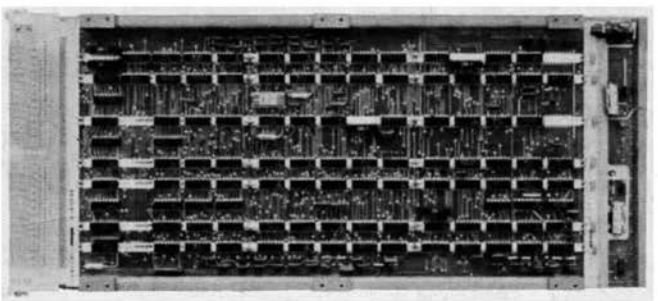
CEC 555 SSマイクロコンピュータは，555 SS—Ⅰ型および555 SS—Ⅱ型の2機種が用意されている．



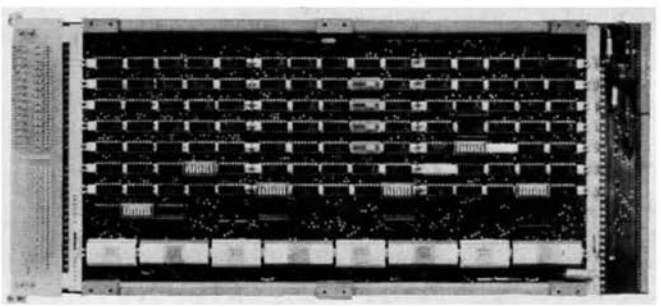
CEC 555 SS—Ⅰ　筐体収納



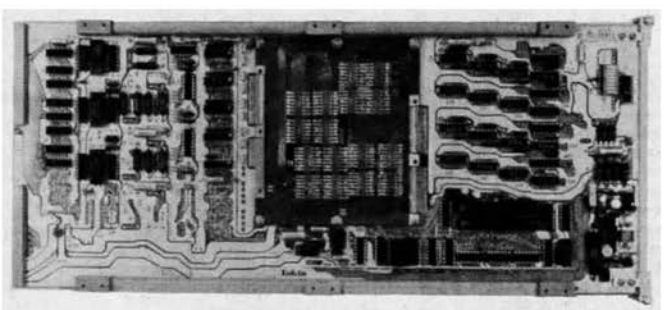
CEC 555 SS　パッケージ



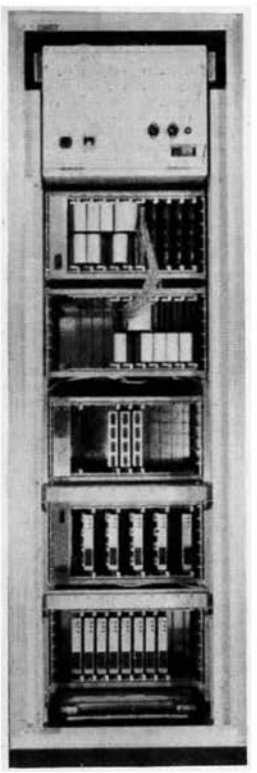
プロセッサユニット



IOユニット



メモリユニット



ラック実装図

555 SS-Ⅱ

1—1 仕 様

方　　式：プログラム記憶方式
演 算 方 式：8ビット並列2進
割　込　み：1レベル4ライン
入出力制御方式：プログラムコントロールモード，ダイレクトストレージ，アクセス方式
メモリサイクルタイム：3μs
メ モ リ 容 量：8.25K BYTE（MAX　16K BYTE）
メ モ リ 素 子：IC（PROM，RAM）
命　令　数：48種類
実 行 時 間：MIN　18.8μs～MAX65.8μs

1—2 特 長

1．演算処理部はMOS LSIを使用しているため，非常に小型化されている．
2．プログラムはPROMに書き込まれているため，プログラムの読込みや複雑な操作の必要はない．
3．割込みラインは4ラインあり，それぞれ独立にマスクできる．
4．マルチプロセッサとして作られているため，低コストで高度な処理ができ，応用分野が広い．
5．MEMORYと演算処理部はダイレクトストレージアクセス方式で接続されているため，MEMORYの空き時間を他の処理装置やI/Oを使用することができる．

1—3 応 用

1．ターミナルコンピュータとしての使用
2．データロガー，省力化装置，製品試験，検査機械等の演算コントロール装置としての使用
3．あらゆる機器に接続してインテリジェントな機器を構成できる．

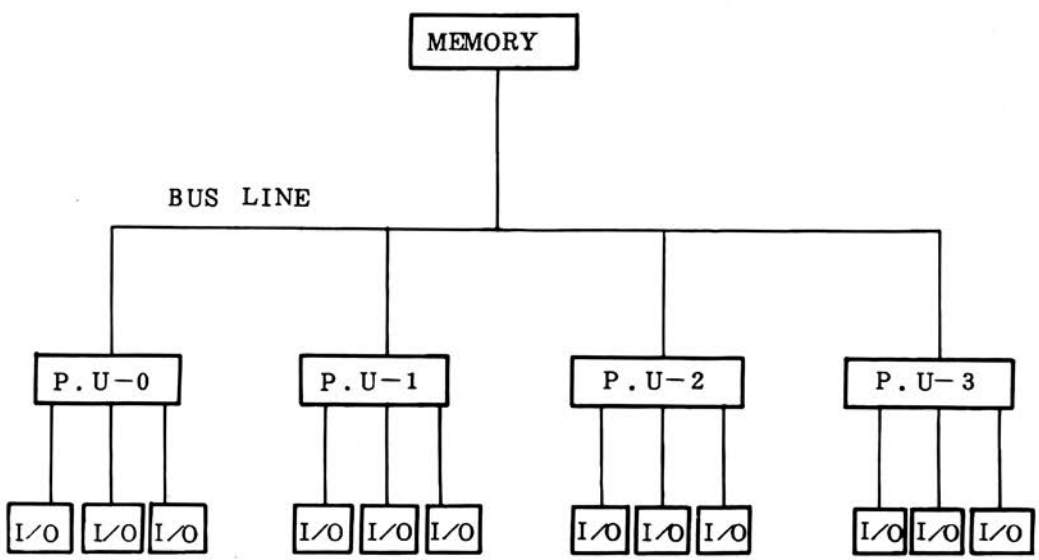


図2—1　555 SS—Ⅰマイクロコンピュータ接続図

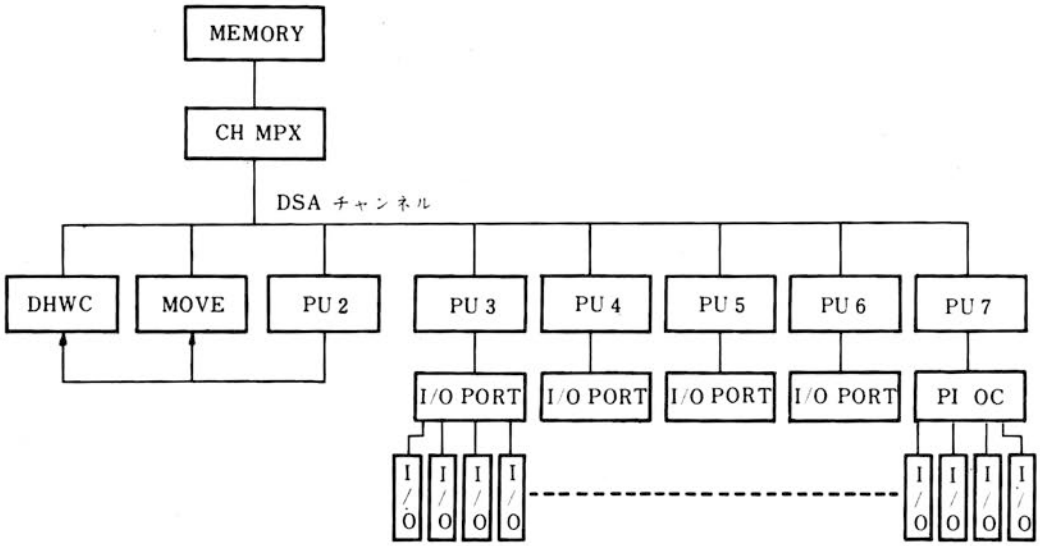


図2—2　555　SS—Ⅱマイクロコンピュータ接続図

2. 構　成

CEC—555SSマイクロコンピュータの構成は図2—1、2—2図のごとくMEMORYとPUから成り立っている．すべての算術演算，論理演算システムの制御命令は，PUで行われる．またプログラム制御によるPUと周辺装置との間の個々のデータの転送は，すべてAレジスタ経由で行われる．PUおよび周辺装置がMEMORYとの間で情報のやりとりを行う場合は，バスライン経由によるデータストレージアクセス方式（DSA）で行う．

3. 記憶装置

3—1　記憶装置の構成

記憶装置は，ICメモリ，アドレスレジスタおよびDSAコントロールから構成される．メモリ容

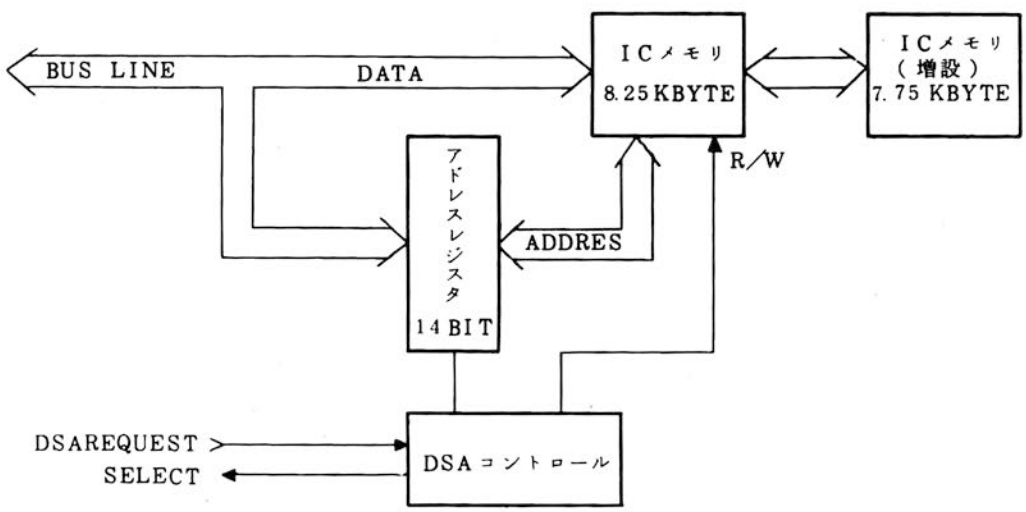


図3—1　555　SS－I記憶装置の構成

量は8.25K BYTEあり，基板を追加することにより16K BYTEまで増設できる．また16K BYTEのワイヤメモリの接続も可能である．**図3－1**は555 SS－Iの記憶装置の構成図である．

3－2　記憶装置の動作説明（以下MODEL 555 SS－I型について説明する）

周辺装置が，メモリを使用する場合，メモリにDSA REQUEST信号を出す．メモリが他の周辺装置によって使用されていなければ，SELECT信号がメモリから出される．他の周辺装置がメモリを使用している場合は使用が完了するまでSELECTは出ない．

SELECT信号を受けたら，周辺装置はADDRESS情報をメモリに送る．MEMORYがADDRESS情報を受け取るとADDACCEPTED信号を出す．メモリをリードする場合はADDACCEPTED信号に続いて，メモリからのDATAとDATA AVAILABLE信号が出される．

メモリにDATAを書き込む場合は，ADDACCEPTED信号を受けたら，DATAとWRITEENABLE信号をメモリに送ると，DATAはメモリに書き込まれる．MEMORY READの場合とWRITEの場合のTIMECHARTを**図3－2**および**図3－3**に示す．

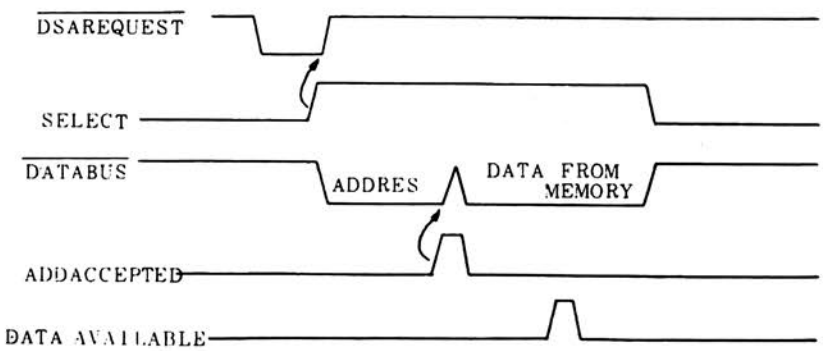


図3－2　MEMORYをREADする場合のTIMECHART

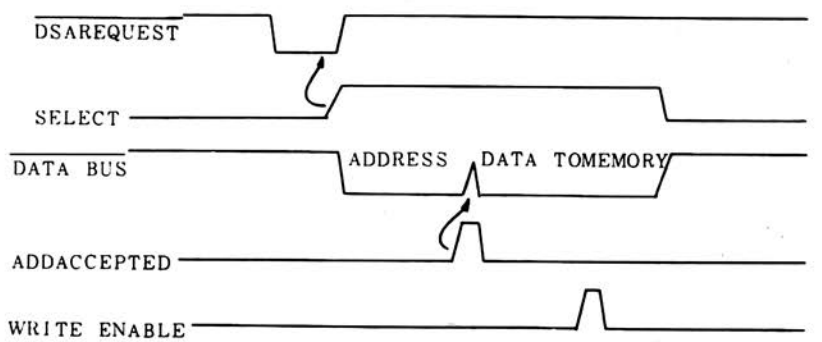


図3－3　MEMORYをWRITEする場合のTIMECHART

3－3　DSAコントロール回路

1つのメモリに対して，周辺装置は4台まで，「いもづる式」に接続することができる．BUSLINE上には，4台の周辺装置からのADDRESS情報と，MEMORYへのDATAおよびMEMORYからのDATAが時分割で存在する．そのコントロールをDSA機構によって行う．DSAコントロール回路は，DCAREQUEST信号によって動作を開始する．DSAREQUESTの信号LINEは，DSARQST0～3までの4本あり，その優先順位はDSARQST3，DSARQST2，DSARQST1，DSARQST0の順になっている．したがってまったく同時に2台以上の周辺装置からDSAREQUEST信号が出た場合，この優先順位の順にSELECT信号が出る．**図3－4**は優先順位を決めるための回路である．

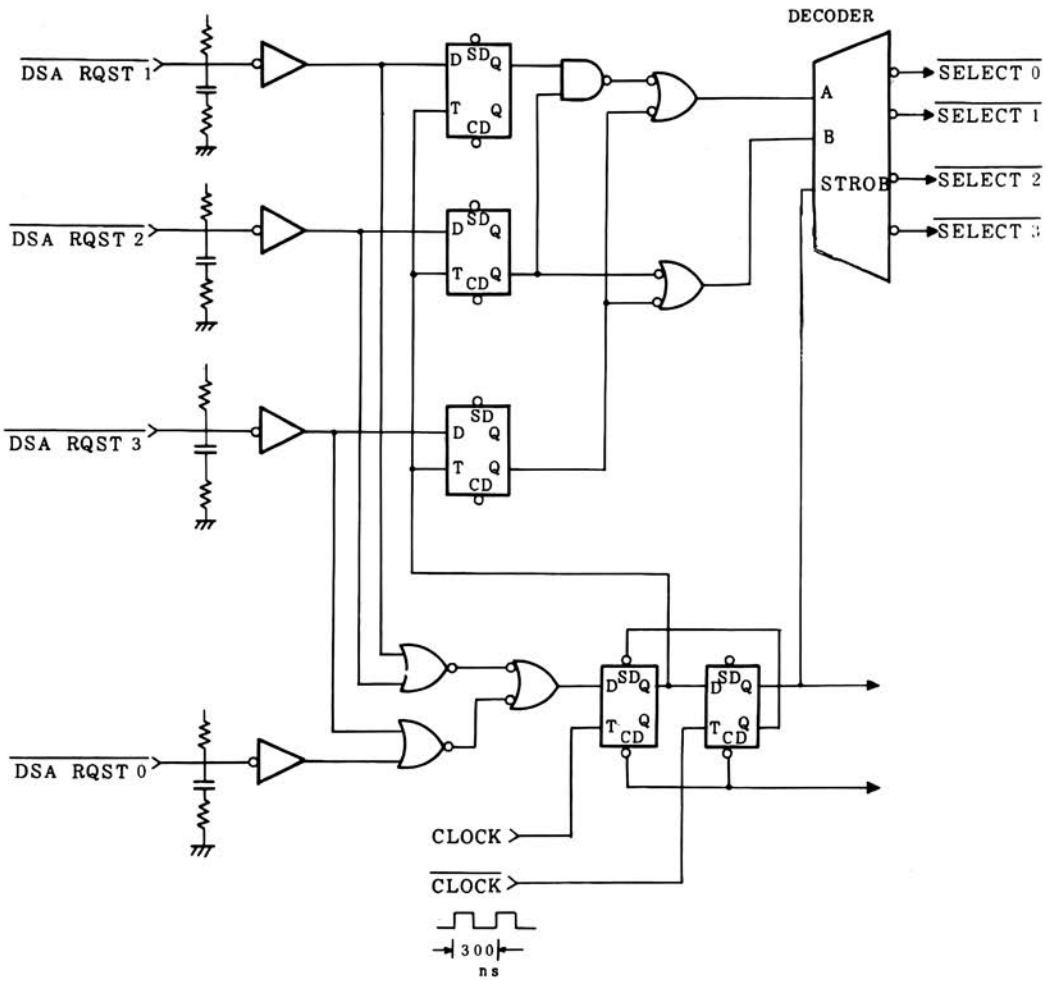


図3－4　優先順位の決め方

4．処理装置（PU）

4－1　処理装置の構成

処理装置の論理演算部は，SINGLE CHIPのMOS LSIを用いて作られており，6つの8 BIT DATA REGISTERと，1つのACCUMULATORと，8 BITのパラレル論理演算UNITと，8つの14 BITのADDRESS STACKと，4つのFLAGBITを持っている．MEMORYは16K WORDまで直接アドレシィングできる．MEMORYと，PUの間のデータのやり取りは，DSA機構を通して行う．

INTERRUPTは，4 LINEあり，それぞれ独立にマスクすることができる．

図4－1に処理装置の構成を示す．

4－2　INSTRUCTION REGISTERの機能

以下にCEC555SS—Iについて詳述する．

MEMORYから読み出した命令をINSTRUCTION REGISTERに記憶し，MEMORYを使用する命令と，ALUを使用する命令に分解し，8つの状態（T1，T1I，T2，WAIT，STOPED，T4，T5）のコントロールを行う．

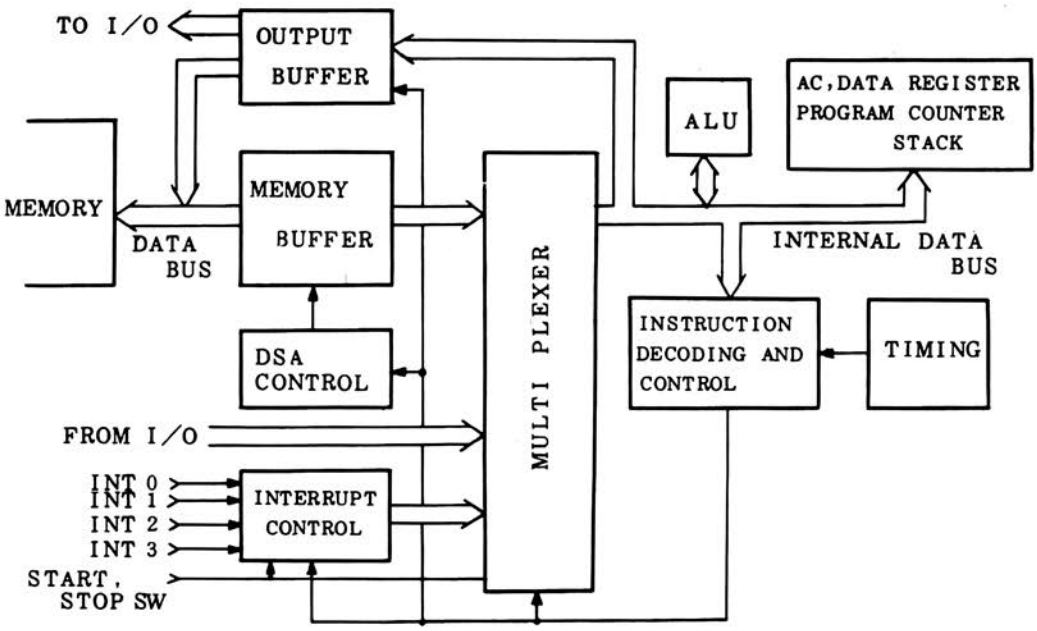


図4－1　555　SS－I　PU構成図

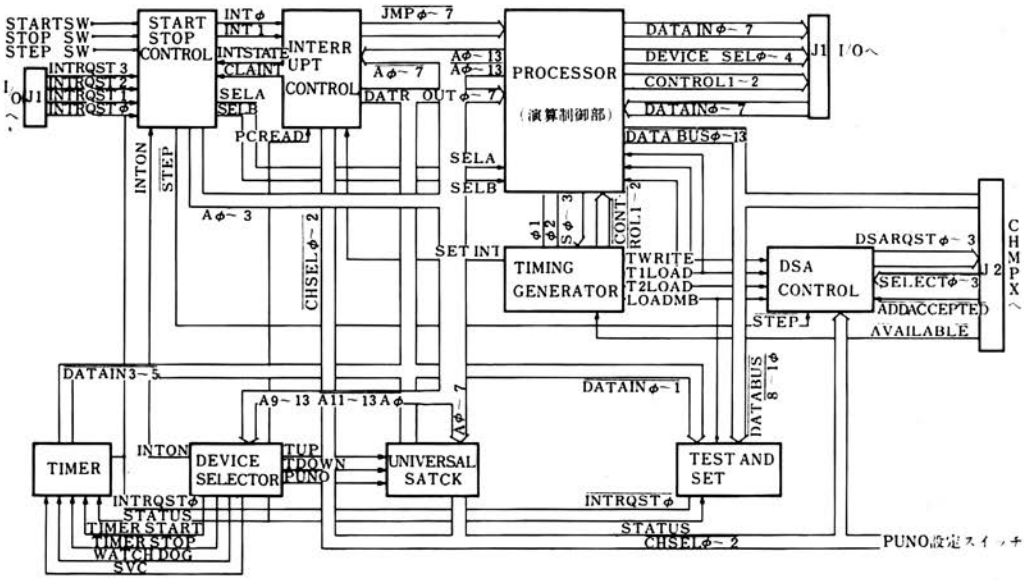


図4－2　555　SS－II　PU構成図

4－3　ADDRESS STACKの機能

ADDRESS STACKは，8つの14BITのレジスタで，そのうちの1つはPROGRAM COUNTERとして使用し，他の7つはSUBROUTINEから出るときのADDRESS POINTERとして使用する．

CALL INSTRUCTIONを実行するとPROGRAM COUNTERの内容がSTACKの中に1レベルPUSHDOWNする．そして，RETURN INSTRUCTIONを実行するとSTACKの内容がPROGRAM COUNTERにHOP UPされる．INTERRUPT信号を受付けた場合も，PROGRAM COUNTERの内容はSTACKに記憶される．

4－4　INDEX REGISTERの機能

555SS計算機システムにはACCUMULATOR（A REGISTER）のほかに6つの8 BITのREGISTER（B, C, D, E, H, L）を持っている.

この6つのREGISTERは, ACCUMULATORとの間で, ADD, SUBTRACT, AND, EXCLUSIVEOR, ORおよびCOMPAREができ, またそれぞれINCREMENT, DECREMENTができる. REGISTERのHとLは, MEMORYにDATAをSTORする場合のADDRESS POINTERとして使用する. この場合REGISTER Lは, 下位8BITを指示し, REGISTER Hは, 上位6BITを指示する（この場合BIT6, BIT7は無関係である）.

4－5　ARITHMETIC LOGIC UNIT（ALU）の機能

ARITHMETIC AND LOGICAL OPERATIONS (ADD, ADD WITH CHARRY, SUBTRACT, SUBTRACT WITH BORROW, AND, EXCLUSIVE OR, OR, COMPARE INCREMENT, DECREMENT）は, 8 BITのパラレル演算ユニットを用いて行われる. これらの演算結果によって, CARRY（C）, ZERO（Z）, SIGN（S）, PARITY（P）の4つのFLAG BITがセットされる. この4つのFLAG BITは, 条件付きCALL, JMP, RETURN命令に使用する. またCARRY BITは倍長演算のCARRYとして使用する.

4－6　INTERRUPT CONTROL回路の機能

INTERRUPT ON, OFF INSTRUCTIONは次のように行なわれる. まずA REGISTERにCOMMANDを持ってきて, OUT 37を実行すると, INTERRUPT ON, またはOFFの状態になる.

INTERRUPT REQUESTが受付けられると, INTERRUPT OFFの状態になる.

OUT 37

COMMAND (A REGISTER)

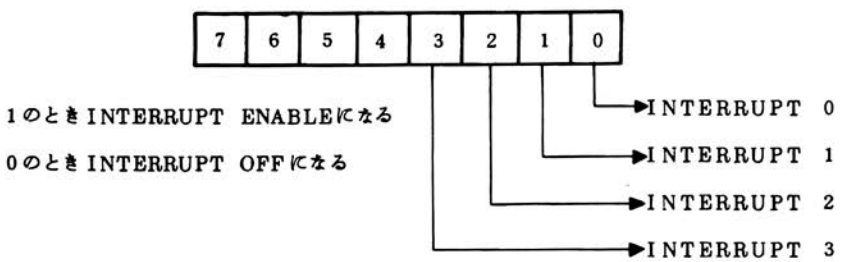


図4－3　INTERRUPT ON,OFF INSTRUCTION

INTERRUPT REQUESTが受付けられると, PUは, 現在実行しているPROGRAMを中断し, RESTART INSTRUCTIONを実行する. PROGRAM COUNTERの内容は, STACKにPUSH DOWNする. 各INTERRUPT LEVELに対するRESTART番地は次のようになる.

PU－3（MASTER）のRESTART番地

INTERRUPT 0 ――― 0_8番地
INTERRUPT 1 ―――10_8 〃
INTERRUPT 2 ―――20_8 〃
INTERRUPT 3 ―――30_8 〃

PU－2（SLAVE）のRESTART番地

INTERRUPT 0 ―――40_8番地
INTERRUPT 1 ―――50_8 〃
INTERRUPT 2 ―――60_8 〃
INTERRUPT 3 ―――70_8 〃

PU－1 (SLAVE) のRESTART番地

INTERRUPT 0 ――1000_8番地

INTERRUPT 1 ――1010_8 〃

INTERRUPT 2 ――1020_8 〃

INTERRUPT 3 ――1030_8 〃

UP－0 (SLAVE) のRESTART番地

INTERRUPT 0 ――1040_8番地

INTERRUPT 1 ――1050_8 〃

INTERRUPT 2 ――1060_8 〃

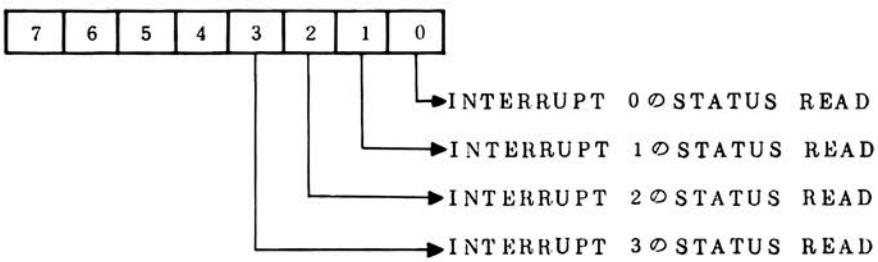
INTERRUPT 3 ――1070_8 〃

4－7 STATUS READ

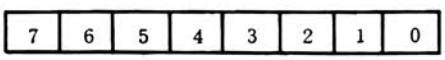
どのDEVICEからのINTERRUPTかを調べるためA REGISTERにSTATUSを読み込む．これは，A REGISTERにCOMMANDを持ってきて，INP 0を行なうとA REGISTERにSTATUSが読み込まれる．STATUSを読み込むと，INTERRUPT REQUEST信号はCLEARされる．

INP 0

COMMAND (A REGISTER)



STATUS(A REGISTER)



A REGISTERにSTATUSが読み込まれる。

図4－4 STATUS READ

I/O側では，INTERRUPTとSTATUSは**図4－5**のような回路でPUに送り出す．

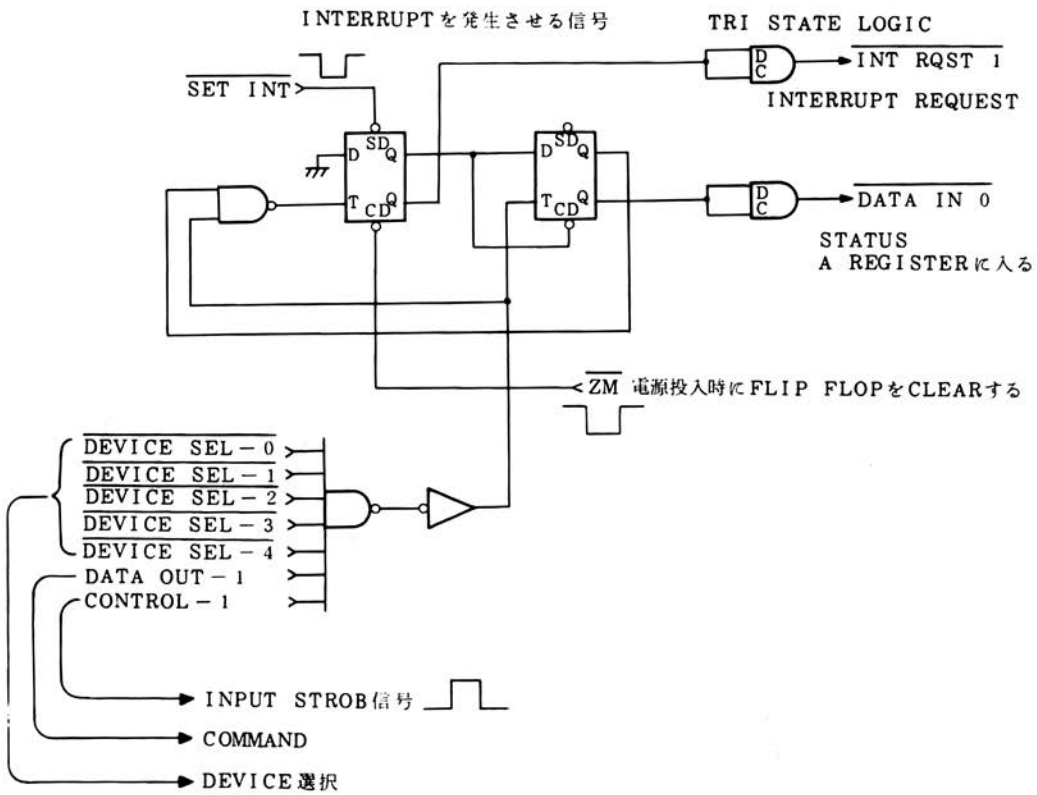


図4－5　I/O側でのINTERRUPTとSTATUS

4－8　I/O OPERATION

I/O命令を実行すると，CONTROL 1，CONTROL 2の2発のI/O CONTROL PULSEが出る．CONTROL 1は，A REGISTERへの入力STROB信号として使用する．A REGISTERにDATAを転送できるDEVICEは，DEVICE №0～7までの8 DEVICEである．残りのDEVICE№10₈～37₈までの24 DEVICEは，OUTPUT専用である．図4－5にI/O CONTROL PULSUのTIMINGを示す．

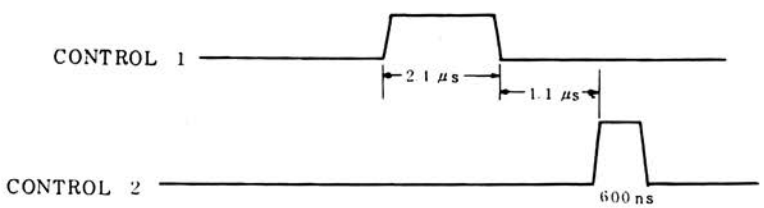


図4－6　I/O CONTROL PULSE

4－9　SVC ON INSTRUCTION

SUPERVISOR CALL INTERRUPTIONをかける．

この主な目的は，SLAVE MODEから，MASTER MODEへの切り替えや，PU間の同期をとるために使用する．

この命令はA REGISTERにCOMMANDを持ってきて，OUT 36を行なうと，SVC ON INSTRUCTIONになる．

OUT 36

COMMAND (A REGISTER)

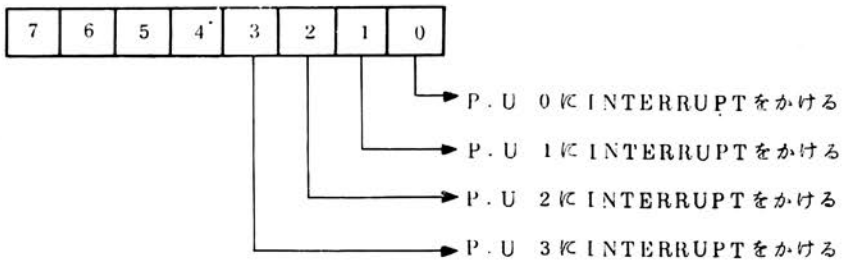
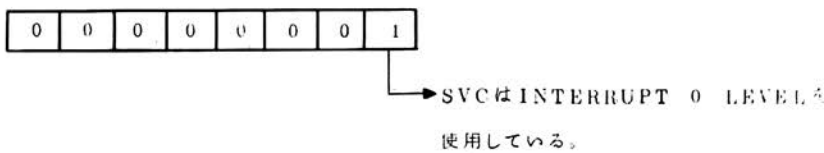


図4－7　SVC ON INSTRUCTION

SVCによる割り込みがかかった場合のSTATUSは，**図4－8**のようになる．ただし，PUのSTART SWITCHによるINTERRUPTのSTATUSは0である．

INP 0

COMMAND (A REGISTER)



STATUS(TO A REGISTER)

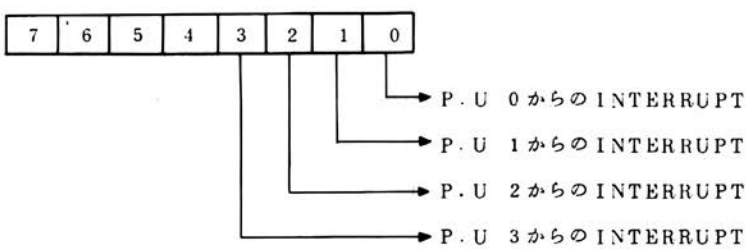


図4－8　SVC COMMANDおよびSTATUS

4－10　STEP OPERATION

STEPスイッチを押すと，PUはINSTRUCTION FETCH CYCLEでWAIT STATEに入り停止する．次にCONTINUEスイッチを押すと，スイッチを1回押すごとに命令を1つずつ実行していく．STEPスイッチを元にもどして，CONTINUEスイッチを押すと，再び命令を連続的に実行する．これは，ソフトのデバックや，ハードのチェック，I/O機器の調整に使用すると便利である．

5．命　令

5－1　命令体系

555SS命令は，INDEX REGISTER INSTRUCTION，ACCUMULATOR GROOP INSTRUCTION，PROGRAM COUNTER AND STACK CONTROL INSTRUCTION，INPUT/OUTPUT INSTRUCTIONおよびMACHINE INSTRUCTIONの5種類ある．

5－2　命令の説明

各命令の説明に使用する記号を下に示す．

〈B2〉　2バイト目のインストラクション

〈B 3〉　3バイト目のインストラクション

r　SCRATCH PAD REGISTER A, B, C, D, E, H, Lを示す.

C　FLAG FLIP-FLOP C, Z, S, Pを示す.

C_4C_3

0 0 CARRY

0 1 ZERO

1 0 SIGN

1 1 PARITY

M　レジスタHとLによって示されたメモリロケーション

( )　メモリまたはレジスタの内用

Am　Aレジスタのmビット目を示す.

STACK PUSHDOWN REGISTER

P　PROGRAM COUNTER

S S S　SOURCE REGISTER

D D D　DESTINATION REGISTER

S S S	D D D
0 0 0	A
0 0 1	B
0 1 0	C
0 1 1	D
1 0 0	E
1 0 1	H
1 1 0	L

5－2－1　INDEX REGISTER INSTRUCTIONS

LOAD DATA TO INDEX REGISTERS　ONE BYTE

DATAはINDEX REGISTER間, またはMEMORYとINDEX REGISTERの間を移動する.

Lr_1r_2　11 DDD SS　(r_1) ← (r_2)

REGISTER r_2の内容をr_1にLOADする. r_2の内容は変らないで保持される. もしS S S＝D D DならNOP (NO OPERATION) 命令になる.

LrM　11 DDD 111　(r) ← (M)

REGISTER HとLによって示されたMEMORY LOCATIONの内容をREGISTER rにLOADする.

EMr　11 111 SSS　(M) ← (r)

REGISTER rの内容をREGISTER HとLによって示されたMEMORY LOCATIONにSTORする.

LOAD DATA IMMEDIATE　TWO BYTE

この命令の次のLOCATIONにあるDATAをREGISTERまたはMEMORYにLOADする.

LrI　00 DDD 110　(r) ← 〈B_2〉

〈B_1〉　〈B_2〉をREGISTER rにLOADする.

LMI　00 11 110　〈B_2〉をREGISTER HとLで示されたMEMORYLOCATIONにSTORする.

〈B_2〉

INCREMENT INDEX REGISTER　ONE BYTE

INr　00 DDD 000　　(r) ← (r) + 1

REGISTER r の内容に 1 を加える.

この結果はCARRY以外のFLAG FLIP—FLOPに影響を与える.

ただしDDD≠000

DDD≠111

DECREMENT INDEX REGISTER　　ONE BYTE

DCr　00 DDD 001　　(r) ← (r) − 1

REGISTER r の内容から 1 を引く

この結果は, CARRY以外のFLAG FLIP—FLOPに影響を与える.

ただしDDD≠000

DDD≠111

5 − 2 − 2　ACCUMULATOR GROUP INSTRUCTIONS

この動作の結果により, CONDITION FLIP—FLOPS C, Z, S, PがSETされる. LOGICAL OPERATION (NDr, XRr, ORr) は, CARRY FLIP—FLOPを 0 にSETする. LOTATE OPERATIONは, CARRY FLIP—FLOPのみに影響を与える.

ALU INDEX REGISTER INSTRUCTIONS—ONE BYTE

この命令はACCUMULATORとINDEX REGISTERの間で行なわれる.

REGISTER SSSのもとの内容は失われない.

ADr　10 001 SSS　　(A) ← (A) + (r)

REGISTER r の内容とREGISTER Aの内容を加算する. その結果はREGISTER Aに入る.

ACr　10 001 SSS　　(A) ← (A) + (r) + (CARRY)

REGISTER r の内容と, CARRY FLIP—FLOPの内容と, REGISTER Aの内容を加算する. その結果はREGISTER Aに入る.

SUr　10 010 SSS　　(A) ← (A) − (r)

REGISTER Aの内容からREGISTER r の内容を引く. その結果はREGISTER Aに入る.

SBr　10 011 SSS　　(A) ← (A) − (r) − (BORROW)

REGISTER Aの内容から, REGISTER r とCARRY FLIP—FLOPの内容を引く. その結果はREGISTER Aに入る.

NDr　10 100 SSS　　(A) ← (A) ∧ (r)

REGISTER r の内容とREGISTER Aの内容の論理積. 結果はREGISTER Aに入る.

XRr　10 101 SSS　　(A) ← (A) ∨ (r)

REGISTER AとREGISTER r のEXCLUSIVEORを取る. 結果はREGISTER Aに入る.

ORr　10 110 SSS　　(A) ← (A) ∨ (r)

REGISTER AとREGISTER r のINCLUSIVEORを取る. 結果はREGISTER Aに入る.

CPr　10 111 SSS　　(A) ← (r)

REGISTER AとREGISTER rの比較を行なう.
REGISTER Aの内容は変化しない. その結果により
FLAG FLIP—FLOPがSETされる.

ALU OPERATIONS WITH MEMORY ONE BYTE

REGISTER HとLで示されたMEMORY LOCATIONの内容とACCUMLATORの間でのARITHMATIC AND LOGICAL OPERATION

ADM	10 000 111	(A) ← (A) + (M)
ACM	10 001 111	(A) ← (A) + (M) + (CARRY)
SUM	10 010 111	(A) ← (A) — (M)
SBM	10 011 111	(A) ← (A) — (M) — (BORROW)
NDM	10 100 111	(A) ← (A) ∧ (M)
XRM	10 101 111	(A) ← (A) ∨ (M)
ORM	10 110 111	(A) ← (A) ∨ (M)
CPM	10 111 111	(A) — (M) COMPARE

ALU IMMEDIATE INSTRUCTION

この命令の次のMEMORY LOCATIONの内容と, ACCUMULATORの間でのARITHMATIC AND LOGICAL OPERATION

ADI	00 000 100 〈B_2〉	(A) ← (A) + 〈B_2〉
ACI	00 001 100 〈B_2〉	(A) ← (A) + 〈B_2〉 + (CARRY)
SUI	00 010 100 〈B_2〉	(A) ← (A) — 〈B_2〉
SBI	00 011 100 〈B_2〉	(A) ← (A) — 〈B_2〉 — (BORROW)
NDI	00 100 100 〈B_2〉	(A) ← (A) ∧ 〈B_2〉
XRI	00 101 100 〈B_2〉	(A) ← (A) ∨ 〈B_2〉
ORI	00 110 100 〈B_2〉	(A) ← (A) ∨ 〈B_2〉
CPI	00 111 100 〈B_2〉	(A) — 〈B_2〉 COMPARE

ROTATE INSTRUCTIONS ONE BYTE

ACCUMULATORの内容は, 右または左に1つSHIFTする. この命令はCARRY FLIP—FLOPのみ影響をあたえる. 他のFLAG FLIP—FLOPは変化しない.

RLC	00 000 010	$Am+_1 \leftarrow Am$, $A_0 \leftarrow A_7$, (CARRY$\leftarrow A_7$) REGISTER Aの内容は左に1つSHIFTする. A_7はA_0とCARRY FLIP—FLOPに入る.
RRC	00 001 010	$Am \leftarrow Am+_1$, $A_7 \leftarrow A_0$, (CARRY) $\leftarrow A_0$ REGISTER Aの内容は右に1つSHIFTする. A_0はA_7とCARRY FLIP—FLOPに入る.
RAL	00 010 010	$Am+^1 \leftarrow Am$ $A_0 \leftarrow$ (CARRY), (CARRY) $\leftarrow A_7$

REGISTER Aの内容は左に1つSHIFTする.
CARRYの内容はA_0に入り、A_7はCARRYに入る.

RAR 00 011 010 Am←Am+1, A_7← (CARRY), (CARRY) ←A_0
REGISTER Aの内容はA_7に入り、A_0はCARRYに入る.

5－2－3 PROGRAM COUNTER AND STACK CONTROL INSTUCTIONS

PROGRAM COUNTER AND STACK CONTROL INSTRUCTIONS JMP INSTRUCTIONS THREE BYTES

この命令の2 BYTE目と3 BYTE目に示されたADDRESS LOCATIONにJUMPする.

JMP	01 XXX 100	(P) ← ⟨B_3⟩⟨B_2⟩
	⟨B_2⟩	2BYTE目と3BYTE目に示されたMEMORYLOCATIONに無
	⟨B_3⟩	条件にJMPする.
JFc	01 OC_4C_3 000	IF (C) = 0, (P) ← ⟨B_3⟩⟨B_2⟩
	⟨B_2⟩	OTHERWISE, (P) = (P) + 3
	⟨B_3⟩	もしFLAG FLIP—FLOP Cが0ならば, MEMORY LOCA
		TION ⟨B_3⟩⟨B_2⟩ にJMPする.
		他の場合は次の命令を実行する.
JTc	01 $1C_4C_3$ 000	IF (C) = 1, (P) ← ⟨B_3⟩⟨B_2⟩
	⟨B_2⟩	OTHERWISE, (P) = (P) + 3
	⟨B_3⟩	もしFLAG FLIP—FLOP Cの内容が1ならば
		MEMORY LOCATION ⟨B_3⟩⟨B_2⟩ にJMPする.
		他の場合は命令を実行する.

CALL INSTRUCTIONS THREE BYTES

SUBROUTINEをCALLする命令, STACKには7LEVELまでのRETURN ADDRESSを入れる能力がある.

CAL	01 XXX 110	(STACK) ← (P), (P) ← ⟨B_3⟩⟨B_2⟩
	⟨B_2⟩	PROGRAM COUNTERの内容はSTACKにPUSHDOWNす
	⟨B_3⟩	る. そして2 BYTE目と3 BYTE目に示されたMEMORY L
		OCATIONに無条件にJMPする.
CFc	01 $0C_4C_3$ 0I0	IF (C) = 0, (STACK) ← (P), (P) ← ⟨B_3⟩⟨B_2⟩
	⟨B_2⟩	OTHERWISE, (P) = (P) + 3
	⟨B_3⟩	もしFLAG FLIP—FLOP Cが0ならPROGRAM COUNTE
		Rの内容はSTACKにPUSHDOWNする.
		そして2 BYTE目と3 BYTE目に示されたMEMORY LOCA
		TIONにJMPする
CTc	01 IC_4C_3 0I0	IF (C) = I, (STACK) ← (P), (P) ← ⟨B_3⟩⟨B_2⟩ OT
	⟨B_2⟩	HERWISE (P) = (P) + 3
	⟨B_3⟩	もしFLAG FLIP—FLOP Cが1ならPROGRAM COUNTE
		Rの内容はSTACKにHOPUPする.
		そして2 BYTE目と3 BYTE目に示されたMEMORY LOCA
		TIONにJMPする.

JAMPとCALL INSTRUCTIONにおける ⟨B_2⟩ は下位のADDRESS BITであり、⟨B_3⟩ は上位のADDRESS BITである. ADDRESSは14 BITを使用するので, ⟨B_3⟩のBIT6, とBIT7は無関係である.

RETURN INSTRUCTIONS　　ONE BYTE

SUBROUTINEから出るときの命令として使用する．STACKの内容はILEVEL HOP—UPする．

RET　00 XXX 111　　(P) ← (STACK)
STACKの最上位にある内容がPLOGRAM COUNTERに入り，そのMEMORY LOCATIONから次の命令を実行する．
STACKの内容は1LEVEL HOP—UPする．

RFc　00 $0C_4C_3$ 011　　IF (C) = 0, (P) ← (STACK)
OTHERWISE (P) = (P) + 1
もしFLAG FLIP—FLOP Cが0なら，STACKの最上位にある内容がPROGRAM COUNTERにHOP—UPし，そのMEMORY LOCATIONから命令を実行する．
もしFLAG FLIP—FLOPが1なら次の命令を実行する．

RTc　00 $1C_4C_3$ 011　　IF (C) = 1, (P) ← (STACK)
OTHERWISE (P) = (P) + 1
もしFLAG FLIP—FLOP Cが1なら，STACKの最上位にある内容がPROGRAM COUNTERにHOP—UPし，そのMEMORY LOCATIONから命令を実行する．もしFLAG FLIP—FLOPが0なら次の命令を実行する．

RESTART INSTRUCTION　　ONE BYTE

RESTART命令は，0ページ (FIRST 256 INSTRUCTION WORD) の，特定番地に，IBYTEでJUMP SUBROUTINEすることができる．

RST　00 AAA 101　　(STACK) ← (P), (P) ← (00000 00AAA000) PROGRAM COUNTERの内容はSTACKにPUSHDOWNされる．
INSTRUCTION REGISTER内のAAAはPROGRAM COUNTERのBIT 3から5にSHIFTされる．
他のBITは0にSETされる．

5—2—4　INPUT/OUTPUT INSTRUCTION—ONE BYTE

入力命令は，8つの入力DEVICEを選択することができる．

INP　01 00M MM1　　(OUTPUT DATA LINES) ← (A)
(A) ← (INPUT DATA LINE)
A REGISTERの内容は，OUTPUT DATA LINEを通して周辺装置に送り出される．次にINPUT DATA LINEからのDATAをA REGISTERに読み込む．
MMMはDEVICE NUMBERである．

出力命令は，24のOUTPUT DEVICEを選択することができる．

OUT　01 RRM MMI　　(OUTPUT DATA LINES) ← (A)
A REGISTERの内容は，OUTPUT DATA LINEを通して周辺装置に送り出される．

5—2—5　MACHINE INSTRUCTION

HALT INSTRUCTION　　ONE BYTE

HALT　00 000 00X　　計算機はHALTする．
OR　　すべてのREGISTERの内容は保持される．

11 111 111

6. プログラミング

6－1　アッセンブラ

555SSのプログラミングを容易にするために，555SSクロスアッセンブラが用意されている．これは，CEC555を使用してシンボリックランゲージで書かれたソーステープから，オブジェクトテープとプログラムリスティング，および，PROM作製用テープを作ることができる．

6－2　ODT

555SSのデバックギィングのために，555—SS　O，D，Tが用意されている．これはCEC555と，555SSマイクロコンピュータをON LINEに接続してデバックを行なうためのプログラムである．テレタイプからの指示により，メモリち内容を変更したり，テレタイプに打ち出したり，またPTPからバイナリーフォーマットのテープを取り出すことができる．ODTとオブジェクトプログラムは同時に走らせることができる．

7. 入出力コネクタ

５５５－ＳＳのコネクタからの信号，高速入出力用，低速入出力用，メモリ拡張用，SVC用の４種類ある．高速および低速入出力用コネクタは，SUBPKGを使用して「いもずる式」に接続する．

高速入出力ラインは，最大，250KHzでデータ転送できる

低速入出力ラインは，５KHz以下のデータ転送に使用するのが望ましい。

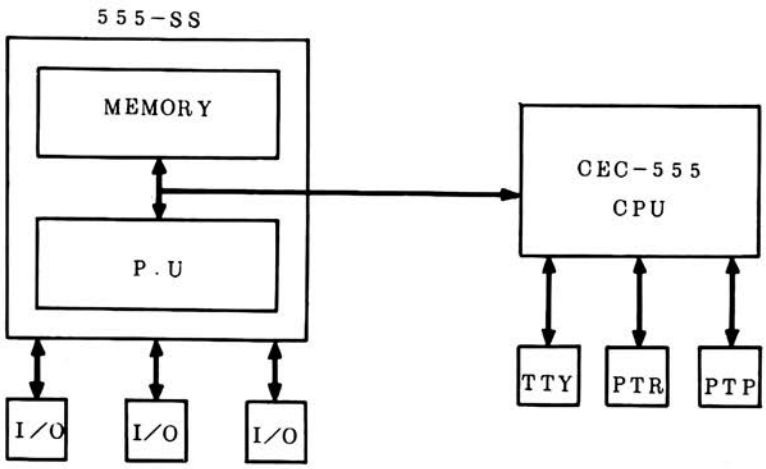


図６－１　ODTを使用する場合の構成

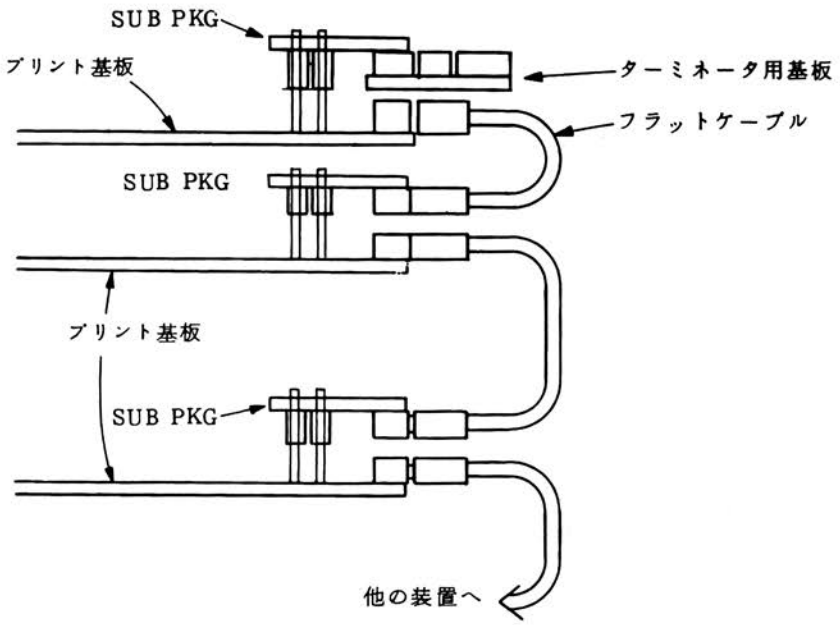


図7－1　コネクタ接続方法

表７－１　高速入出力用コネクタ

J２－A

0	$\overline{\text{DATA BUS 0}}$
1	GND
2	$\overline{\text{DATA BUS 1}}$
3	GND
4	$\overline{\text{DATA BUS 2}}$
5	GND
6	$\overline{\text{DATA BUS 3}}$
7	GND
8	$\overline{\text{DATA BUS 4}}$
9	GND
10	$\overline{\text{DATA BUS 5}}$
11	GND
12	$\overline{\text{DATA BUS 6}}$
13	GND
14	$\overline{\text{DATA BUS 7}}$
15	GND
16	$\overline{\text{DATA BUS 8}}$
17	GND
18	$\overline{\text{DATA BUS 9}}$
19	GND
20	$\overline{\text{DATA BUS 10}}$
21	GND
22	$\overline{\text{DATA BUS 11}}$
23	GND
24	$\overline{\text{DATA BUS 12}}$
25	GND
26	$\overline{\text{DATA BUS 13}}$
27	GND
28	Vcc

J２－B

0	GND
1	$\overline{\text{DSARQST0}}$
2	GND
3	$\overline{\text{DSARQST1}}$
4	GND
5	$\overline{\text{DSARQST2}}$
6	GND
7	$\overline{\text{DSARQST3}}$
8	GND
9	$\overline{\text{SELECT 0}}$
10	GND
11	$\overline{\text{SELECT 1}}$
12	GND
13	$\overline{\text{SELECT 2}}$
14	GND
15	$\overline{\text{SELECT 3}}$
16	GND
17	$\overline{\text{DATA IN}}$
18	GND
19	$\overline{\text{ADD ACCEPTED}}$
20	GND
21	$\overline{\text{DATA AVAILABLE}}$
22	GND
23	$\overline{\text{WRITE ENABLE}}$
24	GND
25	N.C
26	GND
27	N.C
28	GNG

表7－2　低速入出力用コネクタ

J1－A

0	$\overline{\text{DATA IN 0}}$
1	GND
2	$\overline{\text{DATA IN 1}}$
3	GND
4	$\overline{\text{DATA IN 2}}$
5	GND
6	$\overline{\text{DATA IN 3}}$
7	GND
8	$\overline{\text{DATA IN 4}}$
9	GND
10	$\overline{\text{DATA IN 5}}$
11	GND
12	$\overline{\text{DATA IN 6}}$
13	GND
14	$\overline{\text{DATA IN 7}}$
15	GND
16	$\overline{\text{INTRQST 0}}$
17	GND
18	$\overline{\text{INTRQST 1}}$
19	GND
20	$\overline{\text{INTRQST 2}}$
21	GND
22	$\overline{\text{INTRQST 3}}$
23	GND
24	CONTROL 1
25	GND
26	CONTROL 2
27	GND
28	Vcc

J1－B

0	GND
1	DATA OUT 0
2	GND
3	DATA OUT 1
4	GND
5	DATA OUT 2
6	GND
7	DATA OUT 3
8	GND
9	DATA OUT 4
10	GND
11	DATA OUT 5
12	GND
13	DATA OUT 6
14	GND
15	DATA OUT 7
16	GND
17	DEVICE SEL 0
18	GND
19	DEVICE SEL 1
20	GND
21	DEVICE SEL 2
22	GND
23	DEVICE SEL 3
24	GND
25	DEVICE SEL 4
26	GND
27	HALT
28	GND

表7－3　メモリ拡張用コネクタ〈その1〉

J1-A			J1-B		
0	ADDR	0	0	GND	
1	GND		1	ADDR	1
2	ADDR	2	2	GND	
3	GND		3	ADDR	3
4	ADDR	4	4	GND	
5	GND		5	ADDR	5
6	ADDR	6	6	GND	
7	GND		7	ADDR	7
8	MRD	0	8	GND	
9	GND		9	MRD	1
10	MRD	2	10	GND	
11	GND		11	MRD	3
12	MRD	4	12	GND	
13	GND		13	MRD	5
14	MRD	6	14	GND	
15	GND		15	MRD	7
16	MWD	0	16	GND	
17	GND		17	MWD	1
18	MWD	2	18	GND	
19	GND		19	MWD	3
20	MWD	4	20	GND	
21	GND		21	MWD	5
22	MWD	6	22	GND	
23	GND		23	MWD	7
24	R/W		24	GND	
25	GND		25	$\overline{\text{CS 33}}$	
26	$\overline{\text{CS 34}}$		26	GND	
27	GND		27	$\overline{\text{CS 35}}$	
28	NC		28	GND	
29			29		
30			30		

表7－3　メモリ拡張用コネクタ〈その2〉

	J I－C
0	$\overline{\text{CS 36}}$
1	GND
2	$\overline{\text{CS 38}}$
3	GND
4	$\overline{\text{CS 40}}$
5	GND
6	$\overline{\text{CS 42}}$
7	GND
8	$\overline{\text{CS 44}}$
9	GND
10	$\overline{\text{CS 46}}$
11	GND
12	$\overline{\text{CS 48}}$
13	GND
14	$\overline{\text{CS 50}}$
15	GND
16	$\overline{\text{CS 52}}$
17	GND
18	$\overline{\text{CS 54}}$
19	GND
20	$\overline{\text{CS 56}}$
21	GND
22	$\overline{\text{CS 58}}$
23	GND
24	$\overline{\text{CS 60}}$
25	GND
26	$\overline{\text{CS 62}}$
27	GND
28	N．C
29	
30	

	J I－D
0	GND
1	$\overline{\text{CS 37}}$
2	GND
3	$\overline{\text{CS 39}}$
4	GND
5	$\overline{\text{CS 41}}$
6	GND
7	$\overline{\text{CS 43}}$
8	GND
9	$\overline{\text{CS 45}}$
10	GND
11	$\overline{\text{CS 47}}$
12	GND
13	$\overline{\text{CS 49}}$
14	GND
15	$\overline{\text{CS 51}}$
16	GND
17	$\overline{\text{CS 53}}$
18	GND
19	$\overline{\text{CS 55}}$
20	GND
21	$\overline{\text{CS 57}}$
22	GND
23	$\overline{\text{CS 59}}$
24	GND
25	$\overline{\text{CS 61}}$
26	GND
27	$\overline{\text{CS 63}}$
28	GND
29	
30	

CEC 4000 ROMプログラム

1．概　要

CEC MODEL 4000 ROM PROGRAMMERはインテル社製PROM 1602A，1702Aおよび同等品専用のプログラマで，制御にCEC555シリーズ小型電子計算機を用いている．すべての制御はコンソールタイプライタCS—100によって行われ，プログラマには電源スィッチしかついていない．したがってオペレータはコンソールタイプライタを通して，会話形式の操作を行うので，わずか10分ぐらいの説明で操作方法をマスターすることができる．また各ステップごとにソフトウェアによるチェック機能があるので，誤操作をさけることができる．

このプログラマは，特にシステムハウスやシステムユーザでひんぱんにＲＯＭの変更が行われるところに非常に便利である．以下にシステム構成および取扱い方法について説明を行う．

2．システム構成

本章ではハードウェアおよびソフトウェアのシステム構成について述べる．

2—1　ハードウェアシステム構成

MODEL 4000 ROM PROGRAMMERは，CEC555シリーズ計算機のアキュムレータ系バスによって接続され，動作する．その最少システム構成は

1．4KW CPU
2．CS—100 コンソールタイプライタ
3．MODEL 4000 ROM PROGRAMMER

で構成される．ただしオプションによっては，さらに

4．CS—408 高速紙テープリーダ
5．CS—420 高速紙テープパンチャ
6．MT，DISC外部メモリ

等の高速大容量メモリＩ／Ｏ機器を増設して，便利に使用することもできる．

2—2 ソフトウェアシステム構成

ソフトウェアはプログラマコントロール用パッケージがあり，その他オプションとしてHEXDECIMAL用プログラマパッケージと，インテルフォーマットの紙テープをCEC4000用紙テープに変換するパッケージがある．

これをリストすると，

A．PROM PROGRAMMER PACKAGE 標準
　　(OCTAL VERSION)
B．PROM PROGRAMMER PACKAGE オプション
C．PROM TAPE CONVERSION オプション

なお，プログラマコントロールパッケージの操作説明については**4**．に述べる．

3．入力紙テープ

プログラムされるROMの内容は，下記3種の方法でプログラマシステムに入力することができるが，保管および他のシステムからの受渡しのことを考えると，紙テープベースが便利と思われる．

《CEC4000へのプログラムデータ入力方法》

1．紙テープによる．

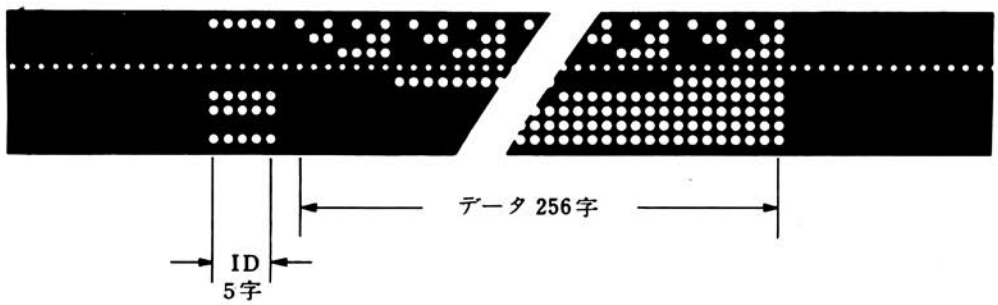
2．ソケット＃0でROMによる．

3．MODIFYで，TTYよりKEY　INする．

CEC4000用紙テープのフォーマットを**図1**に示す．このフォーマットは最初の3字をCEC4000用IDコードとし，次の2字はユーザ用IDコードとする．その次よりROMのアドレス順に書き込みデータを並べる．

CEC用IDコードは図1に示す3字が続けてパンチされていない場合は，正しい紙テープとして受けつけない．ユーザ用IDコードは何がパンチされていようとかまわない（読み込まれない）．

紙テープはもし書き込みデータが256番地分ない場合でも，書き込まない部分を零として，256番地分の長さがあることが必要である．また256番地以上ある場合は，プログラマシステムの方で，256番地以上になると自動的に読み込みを停止する．



図—1　CECフォーマットテープ

4．操作説明

本章ではシステムプログラムをロードする方法と，システムプログラムの使用方法の2項にわけて説明する．

4—1　システムプログラムのロードについて

ここで説明することは計算機の一般的使用法で，計算機の操作説明と重なるが，特にCEC4000システムだけを使用する人のために記述する．なお通常はシステムプログラムはロードされているので，本節で述べてある操作は必要でない．特にオペレータが計算機を誤操作した場合や，他の目的で使用した場合に必要である．

4—1—1　HELPローダのロード

表1に示す27番地より始まる数ステップのプログラムを，計算機のパネル操作によりロードする．操作法はデジスイッチでメモリ番地をセットし“LOAD　PC”KEYを押す．この時FUNCTION切替スイッチを，PCにセットしておくと，正しくメモリ番地がセットされたか，表示されるので確認できる

正しくメモリ番地がセットされたら，次にその番地に書き込むデータを，デジスイッチにセットして，FUNCTION切替スイッチをM　BUFFERにセットし，“STOR”KEYを押す．この操作で，データはメモリに書き込まれ，表示にデータが表われる．メモリ番地は自動的に1番地進められる．したがって次のデータをデジスイッチにセットして“STOR”を押す操作を，最後まで続ける．

最後まで行ったら再び0027番地をセットして，次に“READ”KEYを押すと，書き込みの場合と同様にして，すでに書き込まれたデータを表示し読み出すことができる．

もし書き込みミスが発見された場合は，その番地をセットして修正を行う．正しく入力できたら2節で述べるV—ローダをロードする．

メモリ番地		データ
0027	／	6031
0030	／	5027
0031	／	6036
0032	／	7450
0033	／	5027
0034	／	7012
0035	／	7010
0036	／	3007
0037	／	2036
0040	／	5027

表1

4－1－2　Vローダのロード

CS—100の左スイッチをFREEにセットし，Vローダ紙テープをリーダにセットする．
計算機のデジスイッチを0027にセットし“LOAD　PC”ボタンを押してスタート番地をセットする．

CS—100の左スイッチをSTARTにセットし，計算機押ボタンスイッチ“START”を押す．

CS—100は紙テープ（V—ローダ）を読み込み始める。

読み終ったところでCS—100のスイッチをFREEにし，紙テープをはずす．

以上でVローダのロードは完了したので，次にシステムプログラムテープを3節に従ってロードする．

2節の操作は3節の操作を行っても，計算機が正常に動作しない場合（原因としては誤操作によりローダ部分をこわした等）にのみ行い，通常システムプログラムを，入れ替える場合には行う必要がない．

4－1－3　システムプログラムのロード

CS—100にVローダの場合のようにプログラムテープのトレーラ部をセットする．

CS—100のスイッチをSTARTにセットし，計算機のデジスイッチを7777にセットし，“LOAD PC”を行った後“START”ボタンを押す．

紙テープの読み込みが開始され，ただちにCS—100上に読み込まれている紙テープの名前を印字する．引続きプログラムの読み込みが開始される．終りのトレーラのところまで読まれると，自動的に読み込みは終了する．

またCS—408高速リーダが付いている場合は，CS—100にセットせずに紙テープリーダにセットして，他はCS—100と同様に行う．

なお当社のローダは高低速紙テープリーダを自動選択するので，いずれか一方に紙テープをセットすればあえて計算機は，どちらで書き込みするか，自動的に決定する．

4－2　システムプログラムの操作

本節では，PROM　PROGRAMMER　PACKAGE＜OCTAL　VERSIN＞について述べる．OCTAL　VERSIONはすべての数字を8進数で扱う．

このプログラムは**図—2**に示すごとく，3段階のMODEにわかれており，通常の操作は第1MODEで処理できる．特に一度読み込まれたデータテープの変更等の場合は，MODIFY命令で第2，3MODEの操作をする．

なお，これ等の命令は将来改良のために変更の場合もある．

COMMAND MODE

```
 L       ：＃0ソケットのＲＯＭの内容を読み込む。
*X       ：＃＊ソケットのＲＯＭは消されているかチェックする．
*W       ：＃＊ソケットのＲＯＭへ書き込む．
*V       ：＃＊ソケットの書き込み内容をチェックする．
*S       ：＃＊ソケットのＲＯＭで書き込まれてない番地を探す．
*P       ：＃＊ソケットのＲＯＭの内容をプリントする．
*Z       ：上記Ｌ～Ｐまでの命令をＲＯＭ　＊番地にのみ行う．
*, **Z   ：上記Ｌ～Ｐまでの命令をＲＯＭ　＊～ＸＸ番地にのみ行う．
 Z       ：上記Ｌ～Ｐまでの命令をＲＯＭ　全番地に行う．
 R       ：紙テープの読み込みを行う．
 M       ：ＭＯＤＩＦＹ　ＭＯＤＥに移る．
*N       ：＃＊ソケットにＸ＋Ｗ＋Ｖの複合命令を行う．
*C       ：＃＊ソケットにＬ＋Ｘ＋Ｗ＋Ｖの複合命令を行う．
CONTP    ：ＣＯＭＭＡＮＤの一覧をプリントする．

   ↳ ＭＯＤＩＦＹ　ＭＯＤＥ
        CONT C：ＣＯＭＭＡＮＤ　ＭＯＤＥにもどる．
               Ｅ：バッファの内容を消す．
               Ｄ：バッファの内容をプリントする．
               Ｔ：バッファの内容を紙テープにパンチする．
               Ａ：ＡＲＲＡＮＧ　ＭＯＤＥに移る．

                  ↳ＡＲＲＡＮＧ　ＭＯＤＥ
                       CONT C：ＭＯＤＩＦＹ　ＭＯＤＥＥにもどる．
                            ＣＲ：ＫＥＹ　ＩＮデータを書き込む．
                            ＬＦ：ＫＥＹ　ＩＮしたデータを新しい
                                 番地とする．
                              Ｉ：現在の番地の内容をプリントする．
```

図－2

4－2－1　プログラムの起動

4－1で述べた方法で、システムプログラムがロードされていれば、プログラムは起動できる．起動後はすべてCS—100コンソールタイプライタのキーボードによって制御がおこなわれる．

起動は計算機のデジスイッチで0200番地をセットして、“LOAD　PC”ボタンを押し、“START”ボタンを押す．プログラムが起動すると、ただちにプログラム名と命令一覧をプリントする．

プリント一覧の印刷が終ると、COMMAND：と印字してCS—100は停止する．これはシステムがCOMMAND　MODEにある．

4－2－2　COMMAND　MODEの命令について

COMMAND　MODEに入ったら，以下に述べる命令をKEY　INすることにより操作が行える．

L：ロード命令で0番ソケットに挿入されたＲＯＭの内容を，システムのバッファメモリへ読み込む．動作完了後再びCOMMAND；と印字する．

COMMAND　MODEの命令を実行後次の命令を受けて良い状態になるシステムはCOMMAND：と印字する．

*X：　*番ソケットに挿入されているROMが完全に消されたかチェックを行う．もし消えていないところがあると，その番地と書かれている内容を印字する．なお*は8進数を示す．

消えている場合にはCHECK　OVERと印字する．なお*印はソケット番号の指示で1～6までである．ソケット数はハードウェアのオプションであり，全実装されていない場合もある．ソケット番号の指示は，以下で述べるW，V，S，P，N，C，命令に共通であるので，以下の説明は省略する．

*W：　*番ソケットにバッファに読み込まれたデータを書き込む命令，*＝7の場合は，1～6番のソケット全部同時に書き込む．

完了後正しく書き込まれると，WRITE　OVERと印字する．もし書き込み不良があるとその番地と内容を印字する．

書き込み不良の場合は，再書き込みで良品となる場合がある．

*V：　*番ソケットのROMの内容とバッファの内容を比較する試験で一致すれば，VERIFY　OVERと印字する．不一致の場合にはその番地と，バッファの内容と，ROMの内容を印字する．

*S：　*番ソケットのROM中で，書き込まれていない番地を探し，その番地を印字する．

*P：　*番ソケットのROMの内容をプリントする．

*Z　　*，**Z　　Z：　ROMの番地を指定する命令で，*はソケット番号でなく番地を示す．

L，X，W，V，S，P，N，C，命令はここで指定された番地についてのみしか実行されない．ただし，特に指定しない場合は全番地について行う．

*Zは指定された1つの番地のみ

*，**Zは指定された番地の間について

この指定は次の番地指定がくるまで保持されるので必ず解除する必要がある．

Zは全番地についておこなわれるので，解除命令として使うこともある．

R：　紙テープをバッファに読み込む．CS—408高速紙テープリーダがない場合は，CS—100のリーダ部から読み込ませる．CS—100の場合テープ読み込み完了とともにシステムはCOMMANDを印字してくるが，紙テープの残りの部分があるとそれを不正命令として読み込むので，いそいで左スイッチをFREEまでもどす必要がある．

M：　バッファの内容を変更する．MODIFY　MODEに移る．

N：　*番のソケットについてＸ＋Ｗ＋Ｖ命令を複合形式で実行する．

C：　*番のソケットについてＬ＋Ｘ＋Ｗ＋Ｖ命令を複合形式で実行する．

この複合命令はＲＯＭのコピーを作る時に便利である．

CONT　P：　命令の一覧表を見たい時にKEY　INすると印字する．

4－2－3　MODIFY　MODEの命令について

CONT　C：COMMAND　MODEにもどす．

E：バッファメモリの内容を全部消す．これは紙テープや，コピーするROMがなく，バッファにデータをKEY　INして作る時に，不要なところを消しておくのに便利である．

D：バッファの内容をプリントする．

T：バッファの内容を紙テープにパンチする．

A：ARRANGE　MODEに移る.

4－2－4　ARRANG　MODEの命令について

CONT　C：MODIFY　MODEにもどす.

CR：ARRANGE　MODEでは必ずCS—100は，番地を印字するので，次にキーボードより番地，または書込むべきデータの数字をKEY　INする.

その後，CRであれば，数字は書込みデータとして書き込まれる.

番地は自動的に一番地進む.

LF　：数字の後にLFであれば新しい番地として，番地変更がおこなわれる.

/　：数字をKEY　INせずに“/”をKEY　INすると番地の内容をプリントする.

以下にARRANG　MODEの例を示す.

021	334	CR	021番地へ334が書き込まれる.
022	311	CR	022番地へ311が書き込まれる.
023	012	LF	番地が012へ変る.
012/	123		012番地の内容123を印字
012			

—— はシステムから自動的にプリントされるもの

無印　KEY　INするもの

5．ROMの消し方

添付紫外線ランプで消すことができる.

赤ボタンを押し，ランプがスタートしたこと確認の上，ROMの石英の窓を通して紫外線をチップにあてる．約5分ぐらいで消すことができる．完全に消えたかプログラムによって確認する必要がある.

＊W　：＃＊ソケットのROMへ書き込む.

＊V　：＃＊ソケットの書き込み内容をチェックする.

＊S　：＃＊ソケットのROMでき込まれてない番地を探す.

＊P　：＃＊ソケットのROMの内容をプリントする.

＊Z　：上記L～Pまでの命令をROM　＊番地にのみ行う.

＊XXZ　：　＊～XX番地にのみ行う.

Z　：　全番地に行う.

R　：紙テープの読み込みを行う.

M　：書き込みバッファーの内容の変更を行う.

＊N　：＃＊ソケットにX＋W＋Vの複合命令を行う.

＊　：＃＊ソケットにL＋X＋W＋Vの複合命令を行う.

CONTP：COMMANDの一覧をプリントする.

❹ 555S電子計算機ハードウェア

1. 概　要

CEC—555Sは，大型の計算システムの情報処理および複雑なプロセスのシステムの制御用に設計された汎用の小型計算機である．

基本構成としてはPU（処理装置），メモリー，ＣＳ—100型入出力タイプライタより構成される．PUは演算機能，論理機能，システム制御機能を持っている．メモリーの動作は，PUからの要求があるごとに１リード／ライトサイクルが実行される．

インターフェースの回路を設計することによって，種々の周辺装置を接続することができるが，各装置用のデバイスコードを選択しなければならない．周辺装置と計算機との間のデータ転送はプログラム制御のもとで，入出力バスを通して行われる．そして周辺装置は計算機のプログラム進行中に割込み（インタラプト）をかけることができる．

PUの各ワードタイムにおけるデータ転送は直列に行われるが，PUとメモリー，PUとコンソール，そしてPUと入出力装置との間のデータ転送は並列である．PUとメモリーの間で処理される情報には次の諸特性がある．

語　長

ＰＵ	12ビット	
メモリー	12ビット	

命令形式

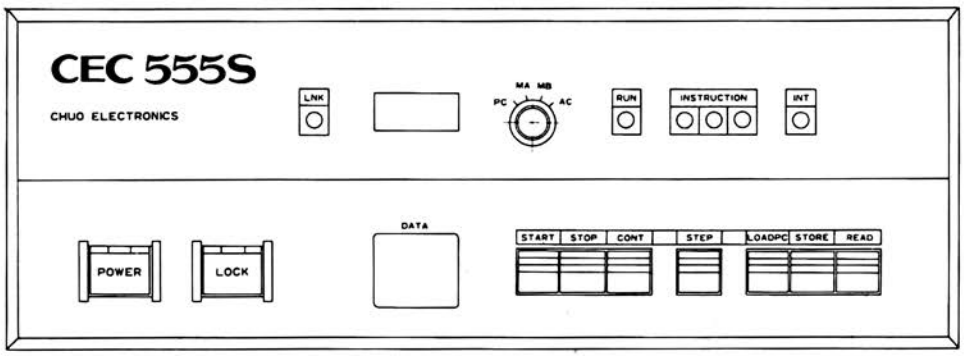
メモリー参照命令	命令コード	３ビット
	間接コード	１ビット
	メモリーアドレス	８ビット
メモリーアドレス		
を含まない命令	命令コード	12ビット
入出力命令	命令コード	12ビット
内部数値形式	２進	
マイナス表示	２の補数	
数値表示	符号１ビット	数値11ビット

PU内のタイミングはすべて同期方式であるが，PUとメモリーはそれぞれ異ったクロックで動作し，かつ同時に走ることはない．したがってPUがメモリーにアクセルするときは，PUが再スタートする．PUの１ワードタイムは2.4μsで，これは１ワード（12ビット）を直列に転送する時間である．各命令の実行時間は各命令の必要とするPUおよびメモリーサイクルの回数によって，それぞれ異る．また与えられた命令が間接アドレスをとるか，インデックスアドレスをとるかによっても異る．

PUは周辺装置から，または周辺装置へのデータ転送を行わなければならないが，各周辺装置から要求があるときは，プログラムインタラプトによってPUに信号を送るので，PUは待時間を必要としない．即ち，PUと周辺装置は平行して動作することができる．基本構成の計算機に付属する周辺装置としてＣＳ—100入出力タイプライタがあるが，これにはキーボード，プリンタ，リーダ，パンチを含んでいる．入出力タイプライタの処理速度は，10キャラクタ（８ビット）／秒である．

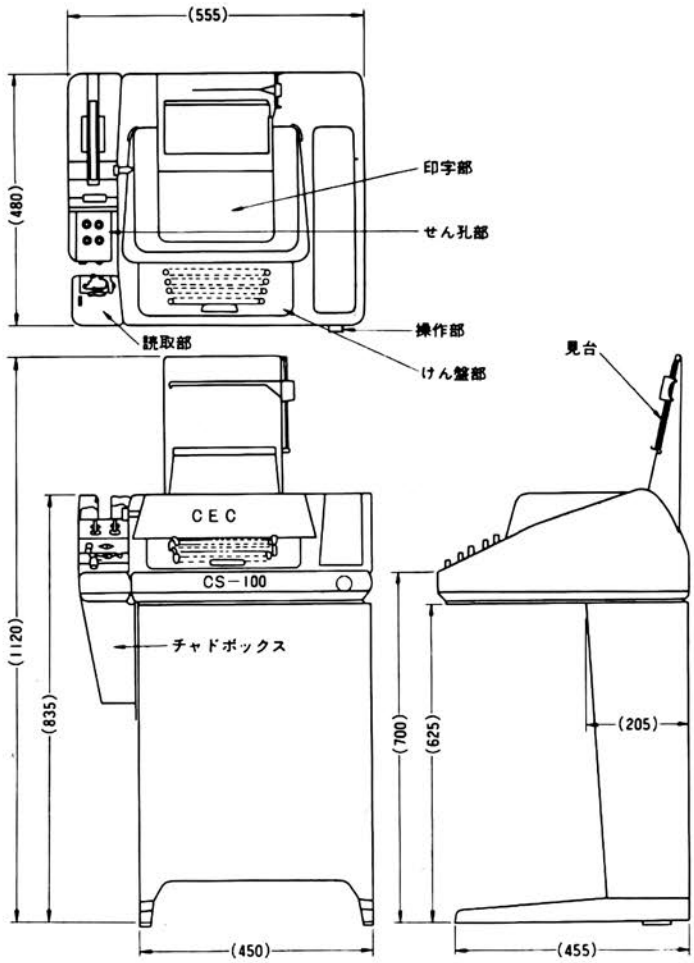
1—1　物理的特性

	CEC－555 S	CS－100
高　サ	150mm	1120mm
横	430mm	555mm
奥　行	560mm	480mm
重　サ	16kg	44kg
周囲温度	10～40℃	
湿　度	35～85%	



1—1図　555S外形寸法

1—1図、1—2図、CEC—555SとＣＳ—100の概略寸法を示す.



1－2図　CS－100

1－2　電気的特性

CEC－555Sの電源の標準入力は，単相100V±10VAC，50または60Hzである．電源はCEC製555－4CD型で，この電源への入力スイッチは計算機前面にある．

消費電力および電流は下記の通りである．

	CEC－555 S	入出力タイプライタ(テレタイプ)
ライン電流	3 A(AC100V)	2.6A(AC100V) (サージ電流　7 A)
消費電力	300W	140W
ロジック電圧		
＋5 V	4.0A	
－5 V	0.4A	
＋12V	0.4A	
＋24V	0.2A	
ロジックレベル		
"1"	＋2.5V～＋5 V	
"0"	0～±0.4V	

2．PUの動作

2－1図にCEC—555Sの各レジスタとデータの流れを示す．

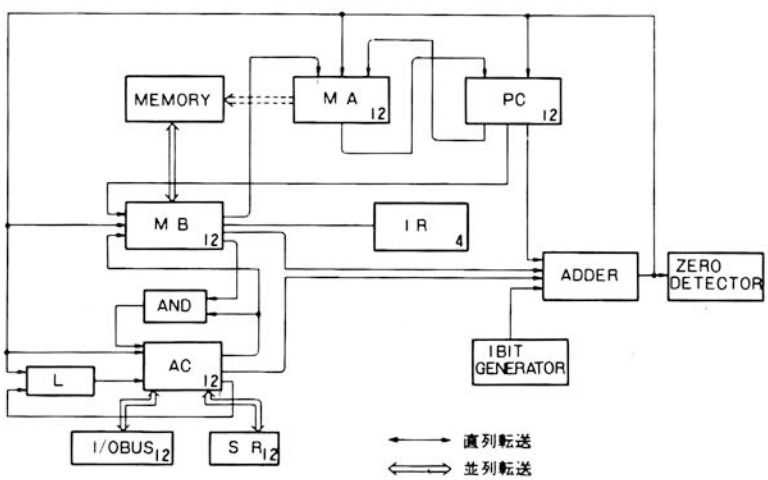
計算機がすべての周辺機器を制御し，プログラムを進行させ，すべての演算およびロジック操作を行う．

PUは12ビットのワードを扱い，メモリーは4096ワードの記憶容量を持っている．ワードのビット番号は左から右へ，ビット0～ビット11と表現し，ワードが扱われるレジスタの各フリップフロップの番号を表示する．ワードは計算機のプログラムの命令，アドレス，オペランド（即ちプログラムのためのデータ）として種々用いられる．

PUはプログラムカウンタによって数えられる連続したメモリーロケーションから読み出された命令を実行してプログラムを進めていく．しかし，テストスキップ命令によってPCをインデックス（PCを1インクリメントすると同意）したり，ジャンプ命令によってPCをある指定したアドレスに変更するようなPCのアドレスの連続性を変えるプログラムの進行もある．メモリーからの読み出しまたはメモリーへの書き込みを行うために，PUはメモリーサイクルを要求し，メモリーアドレスレジスタ（MA）からメモリーへアドレスを指定する．PUとメモリーの間のデータ転送はすべてメモリーバッファを通して行われる．ワードが命令としてMBに読み出され，その左の4ビットがインストラクションレジスタ（IR）に通される．IRの4ビットが命令を実行するためにデコードされる．メモリー参照命令ではMBに残された8ビットがアドレスとしてMAに供給される．

PUの心臓部は，メモリーバッファMB，アキュムレータACと，1ビットの直列加算器である．PUと周辺機器との間のデータ転送はすべてACを通して行われ，外部にはI/Oバスラインを通して行われる．アキュムレータはまた，スイッチレジスタ（SR）に接続されており，このレジスタを通してオペレータが，コンソールから計算機にデータとアドレスを送り込むことができるようになっている．計算機内の並列転送部はMBとメモリー，ACとI/Oバス，SRとACの間のみでそれ以外の転送は直列である．レジスタはすべてシフトレジスタで，二つのレジスタを同時に右へシフトすることによって，1つのレジスタから他のレジスタに情報を転送する．したがって情報はレジスタの右端のビット（L，S，B）から転出され，他のレジスタの左端のビット（M，S，B）から転入される．**2－1図**には示されていないが，転送中にレジスタがその内容を保持したいときは，いつでもPUは，レジスタをそれ自身で循環させることができる．

AND命令（論理積）を行うために，MBとACの内容を単純なANDゲートを通してシフトし，そ



2－1図　命令およびデータの流れ

の結果をACに入れる．他のすべての操作は直列加算器を通して行われる．加算の場合，MBとACの内容が同時に1ビットずつ加算され，その和がACにシフトされる．加算器はまたAC，MB，PCをインクリメント（＋1する）したり，ACをコンプリメント（1の補数をとる）したり，スキップテスト命令の場合MBまたはACの内容をゼロ検知するときに用いられる．

リンク（L）はACに連結した1ビットのレジスタである．リンクは加算またはACのインクリメントのときのオーバーフロー（桁上げ）フラグである．ACとLは13ビットのレジスタとして，右に左に回転させることができる（左の回転は少ない，右回転によって行われる）．1ビットごとの積または商はプログラムで行うが，このときLを使用する．

プログラムとその命令を実行するレジスタの他に，PUには周辺機器，電源故障によって生ずるプログラムインタラプトシステムを持っている．そのようなインタラプトが生じ，ION（プログラムインタラプト要求に応ずる）がONであるとき，PUはPCの内容（プログラムの次の命令が書かれているアドレス）がロケーション0000に貯えられ，ロケーション0001（1番地）の命令が実行される．

PUとメモリーを動かすタイミングは2つのタイミングパルスによって供給される．ワードの直列転送はPUクロックの制御の下で1ワードタイム内に行われる．メモリーアクセスが要求されると，PUはそのタイミングパルスを止め，メモリーサイクルに入り，メモリータイミングパルスの制御の下でメモリーサイクルが実行される．メモリーサイクルが完了すると，メモリータイミングパルスは止り，他のワードタイムを実行するために，PUタイミングパルスが再スタートする．

2—1　プログラミング

プログラムとはある仕事をさせるために用いられる一連の命令群であり，これらはメモリーに貯えられる．メモリー内の各ワードはアドレスと同様8進で0000から7777までである．

プログラムを実行するために，計算機は連続的に増加する（インクリメントされる）ロケーションから命令を引き出してくる．しかし，命令自身がプログラムの流れを変えた場合は，他のロケーションから連続操作を続ける．　**2—1表**に命令とその実行内容を示す．

2—1表　命令およびコード表

1．メモリアドレスを含む命令

記　号	8進コード	実　行　内　容
AND	0××××	ACの内容と指定したアドレスをもつロケーションの内容間でビットごとの論理積をとる演算が行なわれる．結果はACに立つ．指定したロケーションの内容はもとのままである．
TAD	1××××	指定したアドレスをもつロケーションの内容がACの内容に，2の補数演算で加算される．結果はACに立つ．指定したロケーションの内容はもとのままである．
ISZ	2××××	指定したアドレスをもつロケーションの内容に2の補数演算で1を加える．その結果が0であればPCの内容を1だけ増す．つまり次の命令をスキップする．結果が0でなければ次の命令に進む．何れの場合もACの内容は変らない．
DCA	3××××	ACの内容を指定したアドレスをもつロケーションにストアする．ACはクリアされる．もとのロケーションの内容は失われる．
JMS	4××××	PCの内容を1増し，それを指定したアドレスをもつロケーションにストアし，次の命令を指定したアドレス＋1のロケーションからとる．ACの内容は変らない．
JMP	5××××	次の命令を指定したアドレスをもつロケーションからとる．

2. メモリアドレスを含まない演算命令

記　号	8進コード	実　行　内　容
NOP	7000	無効命令
IAC	7001	ACの内容は2の補数演算で1増される.
RAL	7004	ACとLの内容は左へ1つシフトされる.
RTL	7006	左へ2つシフトされる. RAL命令を2つ実行したのと同じ.
RAR	7010	ACとLの内容は1つ右へシフトされる.
RTR	7012	右へ2つシフトされる. RAR命令を2つ実行したと同じ.
CML	7020	Lの補教をとる.
CMA	7040	ACの内容が現在の内容の1の補数にセットされる.
CIA	7041	ACの2の補数をとる.
CLL	7100	Lをクリアする.
CLLRAL	7104	Lをクリアし, ACとLの内容を1つ左へシフトする.
CLLRTL	7106	Lをクリアし, 2つ左へシフトする.
CLLRAR	7110	Lをクリアし, ACとLの内容を1つ右へシフトする.
CLLRTR	7112	Lをクリアし, 2つ右へシフトする.
STL	7120	Lを1にセットする.
CLA	7200	ACをクリアする.
CLAIAC	7201	ACを1にセットする.
GLK	7204	LをAC1に移す.
CLACLL	7300	ACとLをクリアする.
STA	7240	ACを−1にセットする.

3. 入出力命令

種　類	記　号	8進コード	実　行　内　容
インターラプト	ION	6001	プログラムインタラプト要求に応じる.
	IOF	6002	プログラムインタラプト要求を無視する.
ハイスピードリーダ	RSF	6011	フラグが1のとき次の命令をスキップする.
	RRB	6012	バッファの内容を読み取りフラグをクリアする.
	RFC	6014	フラグとバッファをクリアし, 紙テープから1語リードし, バッファに入れフラッグをセットする.
ハイスピードパンチ	PSF	6021	フラグが1のときスキップする.
	PFC	6022	フラグをクリアする.
	PPB	6024	AC4−11の内容をせん孔し, フラグをセットする.
テレタイプキーボード・リーダ	KSF	6031	フラグが1のとき次の命令をスキップする.
	KCC	6032	ACとフラグをクリアする.
	KRS	6034	バッファの内容をAC4−11に読み取る.
	KRB	6036	ACとフラグをクリアし, バッファの内容をAC4−11に読み取る.
テレタイプライター・パンチ	TSF	6041	フラグが1のとき次の命令をスキップする.
	TCF	6042	フラグをクリアする.
	TPC	6044	AC4−11の内容を印刷またはせん孔する.
	TLS	6046	フラグをクリアし, AC4−11の内容を印刷又はせん孔する.

4．プログラムの制御に関する命令

記　号	8進コード	実　行　内　容
HLT	7402	プログラムを停止する．
OSR	7404	DATAの内容とACの内容との間でビットごとに論理和をとるORが行なわれる．その結果はACに残る．
SKP	7410	無条件スキップ．その次の命令はスキップされる．
SNL	7420	L≠0のとき次の命令をスキップする．
SZL	7430	L＝0のとき次の命令をスキップする．
SZA	7440	AC＝0のとき次の命令をスキップする．
SNA	7450	AC≠0のとき次の命令をスキップする．
SZASNL	7460	AC＝0またはL＝1のとき次の命令をスキップする．
SNASZL	7470	AC≠0でL＝0のとき次の命令をスキップする．
SMA	7500	AC＜0のとき次の命令をスキップする．
SPA	7510	AC≧0のとき次の命令をスキップする．
SMASNL	7520	AC＜0またはL＝1のとき次の命令をスキップする．
SPASZL	7530	AC≧0でL＝0のとき次の命令をスキップする．
SMASZA	7540	AC≦0のとき次の命令をスキップする．
SPASNA	7550	AC＞0のとき次の命令をスキップする．
CLA	7600	ACをクリアする．
LAS	7604	DATAの内容をACに入れる．
SZACLA	7640	AC＝0のとき次の命令をスキップし，それからACをクリアする．
SNACLA	7650	AC≠0のとき次の命令をスキップし，それからACをクリアする．
SMACLA	7700	AC＜0のとき次の命令をスキップし，それからACをクリアする．
SPACLA	7710	AC≧0のとき次の命令をスキップし，それからACをクリアする．

2－1－1　数　表　示

CEC—555Sの2進演算には，2の補数および固定小数点法を用いる．ワードが数値として用いられない場合，ビット0（最左端ビット）が符号を表し，ビット0が0なら正，1なら負を表す．正の数において，残った11個のビットが，2進の数の大きさを表している．ある数の負数は2の補数をとることによって得られる．今仮りにXがn桁の2進数だとすると，その2の補数は2^n-Xで，1の補数は$(2^n-1)-X$，または$(2^n-X)-1$である．(2^n-1)からある数を減ずるということは論理的にコンプリメントとすることであり，すべての0を1に，すべての1を0に変えることと同じである．従って，2の補数を作るには符号を含めたワード全体を論理的にコンプリメントし，その結果に1を加えればよい．負の数においては，符号のビットは1で残りのビットの2の補数がその絶対値である．

$+153_{10}=+231_8=$ 0 0 0 0 1 0 0 1 1 0 0 1

0　　11

$-153_{10}=-231_8=$ 1 1 1 1 0 1 1 0 0 1 1 1

0　　11

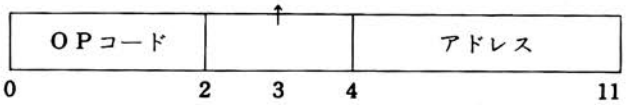
ゼロはすべてのビットが0のワードによって表わされる．この数の補数はすべてのビットが1で，1を加えることによって（2の補数）0に再び戻る．ゼロの表示の場合その符号は正である．正負に限らず，すべての偶数は最後のビットが0，奇数は1である．正の数と負の数は同じ数あるので，0を除く正数の数は負数の数より1つ少ない．

2－1－2　命令の形式

計算機の命令は1ワードをメモリーに貯えるとか，算術的または論理的操作を使うような，基本的な機能から成り立っている．もし命令がメモリーアドレスを必要とする場合は，上位の3ビット（0～2）がオペレーションコードを示し，ビット3がアドレス方式を指定し，残りの8ビットが実効アドレスまたは実効アドレスが貯えられているアドレス（間接アドレス）を指定する．実効アドレスはオペランドを準備するために用いられる実際のアドレスであり，プログラムの流れを変え

るためのアドレスでもある.

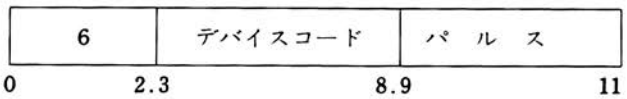
アドレス方式



OP命令とIOT命令は，メモリーアドレスを必要としない.

OP命令においてビット0～2まですべて1（8進で7）で，残りのビットが個々の操作を指定する．従って，いろいろな操作を組合せることによって，細かな命令をつくることができる.

IOT命令ではビット0～2が110（8進で6）で表わされ，ビット3～8が入出力デバイスを指定し，ビット9～11によってI/Oバスを通して，周辺機器に送られるタイミングパルスを選択する．3つの異なった時間に出るパルスが，ビット11，10，9の順序に従って選ばれる.



2－1－3　アドレス方式

メモリーのロケーションは，12ビットで指定され，8進でロケーション0000～7777（10進で0～4095）まで指定される．命令コードのアドレス部は8ビットが許される．プログラミングのために，メモリーを便宜上32ページに分ける．各ページには128（200_8）のロケーションが含まれている．命令コードの8つのアドレスビットによって256のロケーションから1ロケーション，即ち2ページから1ロケーションを選ぶことができる．もしビット4が0なら，ビット5～11で表示されるアドレスを0ページからとり，即ち命令のアドレス部分が0000～0177ロケーションの間のアドレスを表わしている.

ビット4が1なら，ビット5～11が命令が読み出された同じページのアドレスを表示している．12ビットのアドレスは，命令語のビット5～11と命令を読み出すために，PCによってMAに供給された前のアドレスのビット0～4との組合せによって作られる.

命令語のビット3は，ビット4～11に含まれるアドレスの形式を指定している．もしビット3が0ならアドレス指定は直接であり，実効アドレスビット4～11によって決定される12ビットのアドレスである．またもしビット3が1なら，アドレス指定は間接であり，命令のアドレス部分は，12ビットの実効アドレス内容とするロケーションを表わしている.

2－1－4　オートインデックス

プログラムは，0010～0017のロケーションのいずれかを，間接に指定することによって，自動的なインデックスアドレスを指定することができる．これらのロケーションの1つが，間接アドレスで指定されると，PUは指定されたロケーションの内容を読み出し，そのアドレスに含まれるワードをインデックスし（インクリメントする），インデックスされたワードを，メモリーの同じアドレスに書き込み，そのインデックスされたワードが，実効アドレスとして用いられる.

2－1－5　命令の実行時間

2－2表は，CEC—555Sの種々の命令の実行時間を，マイクロ秒単位で示している．表からわかるように，間接指定とオートインデックス指定の場合，必要とする時間は長い．実行時間の後に，（ ）で区切られた1対の数字は，命令に必要なPUとメモリーサイクルの回数を示している.

2—2表　CEC—555S命令実行時間

			間接アドレス		オートインデックス	
AND	8.4μs	2.2	12.6μs	3.3	16.8μs	4.4
TAD	8.4	2.2	12.6	3.3	16.8	4.4
ISZ	12.6	3.3	16.8	4.4	21.0	5.5
DCA	10.8	3.2	15.0	4.3	19.2	5.4
JMS	10.6	3.2	15.0	4.3	19.2	5.4
JMP	6.4	2.1	10.8	3.2	15.0	4.3
IOT	11.8	3.1				
OP1	6.0	2.1				
OP2	9.	3.1				

注）IOT命令のエクゼキュートタイムは2.4×２＝4.8μsである.

上記の時間は，メモリーのサイクルタイムを，2.4μsとして計算されている．プログラムインタラプトに要する時間は，命令の実行時間に8.4μs（2.2）を加える．例えば，TAD命令の最中にインタラプトがかかれば，TAD命令を記しインタラプトを処理し，次の命令の始めまでに要する時間は，8.4μs＋8.4μs＝16.8μsである.

2—2　ワードタイム

各ワードタイムは，14のビットタイム（BT00～BT13）から成り，この14ビットタイムの期間に170nsの間隔で，14発のタイミングパルス（A00～A13）を発生する．最初の12発（A00～A11）のパルスで１つまたはそれ以上の12ビットワードを直列に転送する．13番目のパルス（A12）でOP命令に要求されるある操作（CLAetc）およびメモリーサイクル要求を行う．メモリーサイクルが要求されると，プロセッサのタイミングパルスは止り，メモリー自身のクロックに従ってメモリーが作動する．メモリーサイクルが終ると，PUのタイミングパルスAは14発目（A13）から始り，このパルスによって次のワードタイムを決める．計算機がプログラムまたはオペレータによって止められるときは，BT12でこれを行う.

ワードが処理されるワードタイムは，そのワードがどんな情報を表わし（命令か，アドレスか，オペランドか），それがどんな働きをしなければならないのかによって６種に分けられる.

それらはフェッチ，インデックス，デファー，エクゼキュート，エンド，ブレークの各サイクルである．**命令の実行はフェッチタイムで始るが，プログラムのスタートはエンドタイムで始り，最初の命令を貯えているロケーションを決め，その命令を読み出すために，メモリーサイクルを要求する**．いいかえれば，フェッチタイムで命令を準備するのではなく，フェッチタイムではエンドタイムでメモリーから読み出された命令を処理する．メモリーバッファ（MB）に読み出された命令のOPコード（ビット０～３）をインストラクションレジスタ（IR）に移してデコードし，アドレスの部分（ビット５～11）をメモリーアドレスレジスタ（MA）に転送する．これと同時に，このワードタイムで，次のプログラムのロケーションを指定するために，プログラムカウンタ（PC）の内容に１を加え，PCに戻す.

フェッチタイムの終りに，その命令がアドレスを含む命令であれば，メモリーサイクルをPUが要求する．もしオートインデックスロケーション（10～17ロケーション）が間接的に指定されたときは，PUはインデックスタイムに入る．この他の間接アドレス指定の場合は，デファータイムに

入る．インデックスタイムとデファータイムの２つの場合以外は直接エクゼキュートタイムに入る．インデックスタイムでは，読み出されたアドレスが＋１されて再びメモリーの同じロケーションに書き込まれる．インデックスタイムの後にデファータイムに入り，新しいアドレスをMBよりMAに移し，オペランドをメモリーから読み出さなければならない場合はメモリーサイクルを要求する．命令によって指定された論理演算，算術演算，そしてプログラム操作がエクゼキュートタイムで行われ，オペランドをメモリーに貯える必要があれば，メモリーサイクルの要求を出す．**エンドタイムでPUは次の命令のロケーションを決定し**，このワードタイムの終りでその命令をメモリーから読み出し，準備する．ある場合には，エクゼキュートムとエンドタイムが同時に生ずる．同時でない場合はエクゼキュートタイムからエンドタイムに自動的に移る．

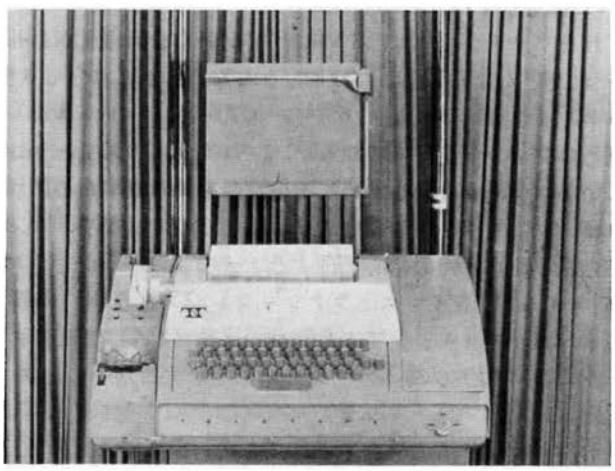
命令を読み出した後で，プログラムインタラプト要求がなければフェッチタイムに戻る．プログラムインタラプト要求がある場合はブレークタイムに入り，PCの内容をロケーション０００００に書き込み，ロケーション００００１の命令を読み出すためにエンドタイムに戻る．

コンソール操作のための特殊な３つのワードタイムがある．PCにスタートアドレスをロードするタイム，メモリーにワードを書き込むワードタイム，メモリーロケーションの内容を読み出すワードタイムの３つである．

I Oバスラインを通して行われる並列転送を含むI Oに要求される特殊な操作を制御するために，IOT命令のエクゼキュートタイムにおいて，1μs幅の特殊なパルスがある間隔をおいて３発出る．



2－2図　CEC－555Sコンソール



2－3図　CS－100コンソール

これら3発のパルスの1つ，あるいそれ以上がIOT命令を行うために，外部の周辺機器に送られるが，いずれのパルスを送るかはビット9～11のどのビットを指定するかによって決められる．

2—3　PUおよび入出力タイプライタの操作

CEC—555 Sの手動制御は，オペレータ・コンソール上のキーおよびスイッチを用いて行われる．機械の状態，コントロール・フリップ—フロップの可視表示器（インディケータ）も，このコンソール上にある．インディケータ・ランプが光っているということは，レジスタのビットもしくはコントロール・フリップ—フロップにバイナリ1が存在することを示す．これらの制御機能およびインディケータの位置を**2—2図**に示した．入出力タイプライタ・ユニット上のすべての制御機能およびインディケータについてはコンピュータの操作上必要な限り述べる．入出力タイプライタ・コンソールは**2—3図**に示してある．

2—3—1　オペレータ・コンソール上の制御キーおよびインディケータ制御キーおよびインディケータ

LOCK スイッチ　このスイッチを一度押すと，ランプは点灯し，スイッチレジスタを除いたすべてのオペレータ・コンソール上のキーおよびスイッチの機能が無効となる．この状態では，不注意によるキー操作をしてもプログラムがこわれることはない．しかし，プログラムはOSRインストラクションを実行することによって，SRの内容を監視できる．このスイッチをもう一度押すとランプは消え，すべてのオペレータ・コンソールキーおよびスイッチは正規に働く．

POWER スイッチ　このスイッチを押すと，ランプが点灯し，コンピュータの基本電源が入り，もう一度押すと，スイッチは消え，基本電源は切れる．

STARTキー　プログラムインタラプト回路をオフにし，L，MB，AC，IRをクリアする．またエンドステイトをセットしPCの内容をMAに転送し，メモリーサイクルを要求するRUNフリップ－フロップをセットする．しかる後コンピュータプログラムをスタートさせる．それゆえに，PCに現在保持しているアドレスにストアされたワードが最初のインストラクションとなる．

LOADPCキー　このキーを押すと，SRの内容がACにセットされ，RUNフリップ－フロップをセットし，ACの内容をPCにシリアル転送する．

STOREキー　このキーを押すと，PCの内容がシリアルにMAに送られ，そしてSRの内容がACに転送される．そしてACから1つのマシン・サイクルでMBにシフトされ，メモリーライトリクエストが生じてMBにあるデータがMAの指定したコアメモリにストアされる．しかる後PCの内容が1増加する．従って，STOREキーを繰返して操作することによって，連続したメモリーアドレスに情報をストアすることができる．

READキー　このキーを押すと，PCの内容が，MAにシリアルに送られ，そしてMAに入っているアドレスに存在するコアメモリーの内容をMBにセットする．しかる後PCの内容が1増加する．このREADキーを繰返し操作することによって連続したコアメモリーアドレスの内容を検査することができる．

CONTキー　このキーを押すと，RUNフリップ－フロップをセットし，コンソール上のインディケータに表示されているインストラクションそのままでPCにより現在指定されているアドレスからプログラムが継続して実行される．

STOPキー　このキーを押すと，進行中のフェッチサイクルの最後で，RUNフリップ－フロップがクリアされる．

STEPスイッチ　このスイッチは，上に上げるとオフである．下に下げると，そのスイッチは，次のインストラクション実行の始めでRUNフリップ－フロップがクリアされる．コンピュータは，実行すべきインストラクションをMBに置いた状態で常にフェッチステイトにあるように見える．CONTキーを繰返し操作すれば，プログラムを1回に，1インストラクション進める．

DATAスイッチ　8進4桁のデータスイッチである．各桁ともスイッチの回転によって0～7までセットされる．このレジスタの内容は，LOAD，PCキーで，PCにロードされ，STOREキーで，MBおよびコアメモリにロードされる．スイッチレジスタの内容は，OSRインストラクションのプログラムコントロール下で，ACにセットできる．

レジスタ切換用スイッチおよびレジスタ表示用インディケータ (PC)　コンソールの中央部に切換スイッチおよびレジスタの内容を表示する8進4桁のインディケータがある．

切換スイッチをPCにセットするとPCの内容がインディケータに表示される．コンピュータが停止したとき，PCの内容は，次にSTARTキーを押すときに，実行される最初のインストラクションのコアメモリーアドレスとなっている．CONTキーを押した後に実行されるインストラクションは，そのとき，MBにあるものであって，PCは次に実行すべきインストラクションを示すものである．

(MA)　切換スイッチをMAにセットすると，MAの内容が表示される．通常，MAの内容は，現在あるいは以前に読み取られ，または書き出されたワードのコアメモリーアドレスを示す．STOREキーまたは，READキーを押した後では，MAの内容は情報をライトまたはリードしたコアメモリーアドレスを示している．

(MB)　切換スイッチをMBにセットするとMBの内容が表示される．通常MBの内容は，MAに保持されているコアメモリーアドレスに丁度リードまたはライトしたばかりのワードが入っている．

(AC)　切換スイッチをACにセットするとACの内部が表示される．

LINK インディケータ　LINKの内容を表示する．

INSTRUCTION インディケータ　現在実行するインストラクションがIRでデコードされバイナリーで表示される．

AND	000
TAD	001
ISZ	010
DCA	011
JMS	100
JMP	101
IOT	110
OPR	111

INT インディケータ　INT．ENABLEフリップ－フロップが1になった時が表示される．表示がついた時，進行中のプログラムは，I/O機器からのプログラムインタラプトを受け付ける．

RUN インディケータ　RUNフリップ－フロップが1の状態になったことを示す．表示がついた時，内部のタイミング回路がイネイブルの状態となり，コンピュータは，インスト

ラクションを実行する.

2—3—2 入出力タイプライタ制御スイッチ

制御スイッチ	機能
REL. プッシュボタン	テープを取りはずしたり取りつけたりするためにパンチからテープを解放する.
B. SP. プッシュボタン	今パンチしたばかりのキャラクタを手動で修正または抹消するためにパンチ上のテープを1スペースだけバックスペースする.
OFF and ON プッシュボタン	テレタイプキーボードプリンタを連動して, テープパンチを動作させるか, させないかを制御する.
START／STOP／FREE	テレタイプと連動してテープリーダを動作させるかさせないかを制御する. FREEの位置にセットするとリーダは切り離され, テープをロードしたり取りはずしたりできる. 中間のSTOPの位置にセットすると, リーダ機構は動作しないがテープを束縛する. STARTの位置にセットすると, リーダはテープを束縛し, プログラムコントロール下で動作する.
キーボード	パンチONプッシュボタンを押した時, タイプライタおよびパンチングテープとして使用し用紙上に印刷する. そして, LINE/OFF/LOCALスイッチをLINEの位置にすれば, コンピュータに入力データを供給する用法となる.
LINE／OFF／LOCAL	テレタイプ基本電源を供給し, また中央処理機構とのデータの接続を制御する. LINEの位置にセットするとテレタイプは, コンピュータのI/O機器として, 接続される. OFFの位置にセットするとテレタイプの電源が切れる. LOCALの位置にセットするとテレタイプは, オフライン機器として動作し, 中央処理機構に対する信号接続は断たれる.

2—3—3 操作手順

CEC—555Sへ情報を入れたり取り出したりするのに色々な方法がある. いかなる方法を用いるかは, もちろん, その情報の形, 時間の制限, コンピュータに接続されている周辺機器に関係する.

次の手順はCEC—555Sのすべての用法にとって基本的である. それらは, プログラミングを行う際それ程多くは用いられず, またコンピュータの用法も複雑となるけれども, それらは, 入門段階でプログラムを下調べするとか, コンピュータの入出力転送の機能を学ぶのに重要である.

2—3—4 オートローダ

2—5図のように電源からの$\overline{\text{AST}}$によりROMコントロール内で$\overline{\text{PPC}}$を発生し, 0番地を指定する. 次にINSTRUCTION FIELDとして, ROMを指定しなければならないため, FAL (FIELD AUTO LOADER) をセットし, その後の$\overline{\text{PAST}}$で計算機は自動的にスタートする.

ROMコントロールPCB (NS-T-ROM) 内には,

RIM LOADER (32WORD) 1M5600 2個
BINARY COADER (256WORD) 1702A 2個
PROGRAM (2KWORD) 1702A 16個

のROMを使用し, 32WORDのRIMLOADERでBINARY LOADERをCOREにロードするためのローダである.

BINARY LOADERをCOREにロードし終ると, FALはクリアされる (AUTO LOADERの終了以後, INSTRUCTION FIELDはすべてCOREが指定される).

次に, COREにロードされたBINARY LOADERによりPROGRAMがCOREにロードされ, サムチェックがOKであると, PROGRAMは実行される. 万一, サムチェックがBADであった場合は再度PROGRAMをロードし直し, サムチェックがOKになるまで5回繰返す. 5回の間にサムチェックがOKになればPROGRAMは正常に実行される. 5回おこなってもサムチェックがBADのとき

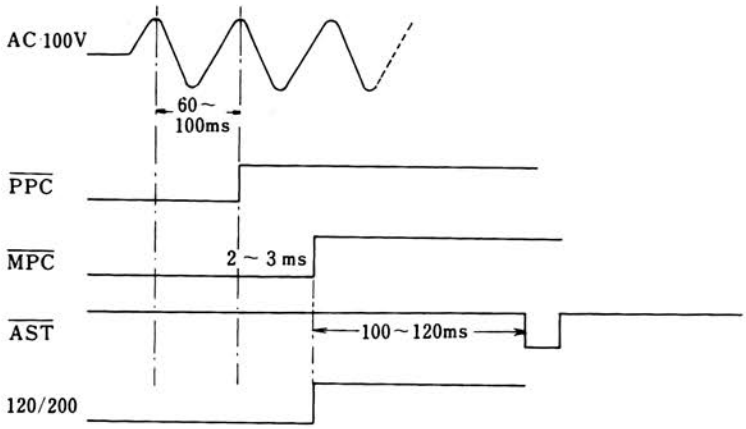
はERRORであるということで計算機はHLTする.

以上，CEC555Sは電源スイッチを押すだけでPROGRAMをCOREにロードし，実行するまでの一連の動作をすべて自動的におこなう．従って，オペレータは電源スイッチを押すだけで，後の操作は必要としない.

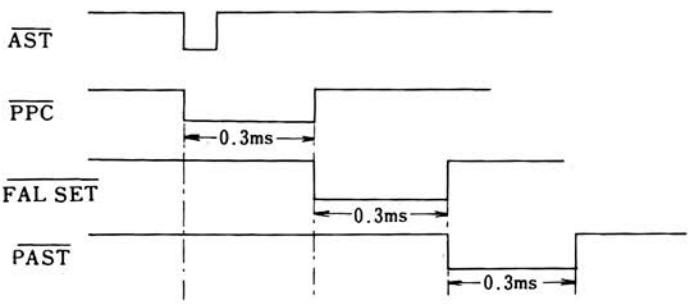
また，計算機自体も表面パネルは電源スイッチだけで，後は何もない（メクラ）．ただし，保守用として，標準操作パネル（CEC－555Sパネル）が別途用意されている.

〔**動作**〕

CEC－555Sでは，電源をONにすると電源より下図のようなシーケンスが発生する.



電源より$\overline{\text{AST}}$が出るとROMコントロール内で下図のようなシーケンスを発生する.



〔**操作手順**〕

SRからRIMローダをストアすることなしにハード化されたRIMローダで，自動的にRIMフォーマットのテープをロードすることができる機能であり，C61-52-20，C61-52-21，C61-52-22のPKGより構成されている.

以下，操作の手順を示す.

1．コンピュータのPOWERスイッチをONにする.

2．テイテイプのLINE／OFF／LOCALスイッチをLINEの位置にセットする.

3．START／STOP／FREEスイッチをFREEの位置にセットし，カバーガードを右にはずし，テレタイプリーダにA.L付のテープをセットし，スプロケットホィールティースをテープのフィードホールにかみ合わせる．その後，カバーガードを閉じ，3ポジションテレタイプリーダ・スイッチをSTARTの位置にセットする.

4．コンピュータのレジスタ切換用スイッチをA.Lの位置にセットし，SRを4000以上（PTRの場合は3777以下）にセットし，STARTキーを押すと，A.L付テープの前半のBINローダ（RIMフ

ォーマット）をコアメモリにロードし始める．このとき，レジスタ表示用インディケータにオートローダでロードしていることが青示され，BINローダのロードを完了すると表示は消える．その後，テープの後半のプログラム（BINフォーマット）をコアメモリにロードする．

5．以上の操作により，プログラムのロードが終了し，SRおよびLOAD PCキーによってPCの中にプログラムのスタートアドレスをロードし，STARTキーを押すことによって，プログラムの実行を開始することができる．

2—3—5　手動によるデータのストアおよび修正

プログラムおよびデータをオペレータコンソール上のスイッチ類によって，手動でストアしたり修正したりすることができる．手動によるデータのストアの主な用法は，コンピュータのコアメモリにリードインモードローダプログラムをロードすることである．

リードインモード（RIM）ローダは，RIMフォーマットでせん孔されたテープ上から，CEC—555Sにプログラムを自動的にロードするために用いられるプログラムである．このプログラムとRIMテープフォーマットについては，オペレーションマニュアルを参照のこと（オペレーション・マニアルに掲げたRIMプログラムを用いて手動でストアする練習をしてみるとよい）．

CEC—555Sのコアメモリに手動でデータをストアするには次の手順に従えばよい．

1．POWERスイッチをONにする．

2．ストアすべき最初のワードのアドレスビットに対応してDATA（SR）のビットスイッチをセットする．レジスタ表示切換用ロータリースイッチをPCにセットし，LOAD PCキーを押すと，SRによってセットされたアドレスが，PCに保持されるのがみられる．

3．PCに今アドレスをセットしたが，そのアドレスにストアすべきデータまたは，インストラクションワードに対して，SRをセットする．ロータリースイッチをMBにセットしSTOREキーを押すと，MB，従ってコアメモリに，SRにセットしたワードが入るのがみられる．

また，PCが1増加し，これにより次々とデータをSRにセットし　STOREキーを繰返し押すことによって連続したアドレスにストアできるのである．

コアメモリ内のアドレスの内容をチェックするためには，上記ステップ2におけるように，PCの中にアドレスをセットし，ロータリースイッチをMBにセットしてREADキーを押すと，そのアドレスの内容がメモリー・バッファーインディケータによって表示される．PCの内容は，READキーを操作することにより1増加する．従って最初の（または，始めの）アドレスをロードしたら，操作を繰返し行うことによって，連続したアドレスの内容を調べることができる．ステップ2と3の両方を繰返すことによって任意のアドレスの内容を修正することができる．

2—3—6　プログラムのコントロール下でデータをロードすること

前もってコアメモリ内にストアしてあるプログラムを始動するだけで，情報を自動的にコンピュータにストアしたり，あるいは修正したりすることができる．例えばコアメモリにRIMローダをストアしておくと，次のように，RIMフォーマットテープをロードすることができる．

1．POWERスイッチをONにする．

2．テレタイプのLINE　OFF　LOCALスイッチをLINEの位置にセットする．

3．START　STOP　FREEスイッチをFREEの位置にセットしラッチを右に押してカバーガードをはずし，テレタイプリーダにテープをセットする．その時，スプロケットホイールティースをテープのフィードホールにかみ合わせる．カバーガードを閉じ，STOPの位置にスイッチをセットする．テープは，リーダの後方からロードする．従って，リードされるにつれ，前方に動く．リーダに，テープを正しく置くと，スプロケットホイールの左に3つのビットポジションがあり，スプロケットホイールの右に5つのビットポジションがくる．

4．RIMローダ・プログラムのスターティング・アドレス（プログラムをロードすべきアドレス

ではない）をSRおよびLOAD　PCキーによってPCにロードする．

5．コンピュータのSTARTキーを押し，3ポジションテレタイプリーダ・スイッチをSTARTの位置にセットするとテープは，自動的にリードされる．

バイナリローダ（BIN）プログラムを自動的にストアするには，前に述べたようにRIMローダを用いる．シンボリックアセンブラでアセンブルされてバイナリフォーマットになったプログラムテープは，コアメモリにストアしたBINローダによって，前の手順に述べたと同様にストアされる．その際ステップ4でBINローダのスターティングアドレスは通常7777であることに注意する．このようにしてプログラムをストアし終った時，コンピュータは停止する．

その際プログラムが正しくストアされておればACは0となる筈である．コンピュータがAC中に0以外の数を含んだ状態で止った時は，チェックサムエラーが検出されたのである．プログラムをストアし終れば，SRおよびLOAD　PCキーによってPCの中にプログラムのスターティングアドレス（通常，テープの先頭に表示されている）をロードし，STARTキーを押すことによって，プログラムの実行を開始することができる．

2—3—7　オフラインの入出力タイプライタ（テレタイプ）操作

テレタイプは，CEC—555Sと切りはなして，タイプしたりテープせん孔を行ったり，あるいはテープを複写したりするためにも用いられる．以上に関するテレタイプの用法を次に述べる．

1．コンピュータのPOWERスイッチをOFFにする．

2．テレタイプLINE/OFF/LOCALスイッチをLOCALの位置にセットする．

3．パンチを用いる時まずカバーを上げて，手動でパンチのうしろにあるガードにロールの最初からテープをフィードする．そして，フリクションホイールを手で回して，テープをパンチを通して進ませ，その後カバーを閉じる．ONプッシュボタンを押して，パンチをオンにし約2フィート程のリーダを作る．リーダトレーラは，200または377というコードである．コード200リーダを作るには，左手でCTRLとSHIFTキーを同時に押し，そしてホールドし，REPTキーを押し，ホールドし，@キーを押しそして，リリースする．必要な量のリーダがパンチされれば，すべてのキーをリリースする．377コードを作るには，十分な量のリーダがパンチされるまで，REPTおよびRUBOUTキーの両方を同時に押しホールドすればよい．

テープをパンチしている間に，誤ってキーを打ち損じた場合，テープを次のようにして訂正する．すなわちそのエラーをNキャラクタタイプおよびパンチした後に気がついた時には，パンチB. SP.（バックスペース）プッシュ・ボタンをN＋1回押す．そして，キーボードRUBOUTキーをN＋1回打つ．その後誤ったキャラクタのところからタイプおよびパンチを続ける．

現存するテープリストをとったり複写したりするには次のようにする．すなわちこの節の先頭に書いてある手順を実行する．それから“プログラムのコントロール下でデータをロードすること”の手続きのステップ2で述べたように，複写するテープをロードする．リーダのSTART/STOP/FREEスイッチをSTARTの位置にセットすることによって，テープの複写を始める．複写すべきテープが完全にリードされると，パンチとテレプリンタは止る．

せん孔テープ上の情報を書き入れたり削除したりして修正するには，そのテープの正しい部分を複写し，かつ手で情報をパンチインをしたりあるいはまた削除すべき情報のパンチを行わないようにすればよい．これを行うには，テープがリードされる時テープを複写しながらタイプされてゆく情報を注意深く見なければならない．このようにしてリーダSTART/STOP/FREEスイッチを訂正すべき地点の直前でSTOPの位置にセットすることができる．その後書き入れるべき情報をキーボードを使って手でパンチすればよい．情報を削除するにはパンチOFFプッシュボタンを押し，テープ上の削除すべき部分がタイプされてしまうまで，リーダを動作させればよい．リーダが予定通り，正確に止まらなかった時には，新しいテープ上で，1つあるいは2つのキャラクタをバックスペー

スしたりラブアウトする必要がある．ラブアウトすべきキャラクタの数は，タイプされたコピーによって正確に決定される．タイプされたキャラクタを勘定する時には，スペースを誤りなく数えなければならない．修正を行った後，正規の方法でテープの複写を続行する．

新しく複写したテープ，または修正したせん孔テープは，それらをオフラインでリードし，タイプされたコピーを注意深く眺めて正しいかどうかを確かめなければならない．**2—3表**にテレタイプのコードを示す．

2—3表　テレタイプのコード

ASCII　コード

文　字	コード	文　字	コード
A	301	!	241
B	302	"	242
C	303	#	243
D	304	$	244
E	305	%	245
F	306	&	246
G	307	'	247
H	310	(	250
I	311	)	251
J	312	*	252
K	313	+	253
L	314	,	254
M	315	−	255
N	316	.	256
O	317	/	257
P	320	:	272
Q	321	;	273
R	322	<	274
S	323	=	275
T	324	>	276
U	325	?	277
V	326	a	300
W	327	Line Feed	212
X	330	Return	215
Y	331	Space	240
Z	332	ALT MOCE	375
0	260	Rub Out	377
1	261	Escape	233
2	262		
3	263		
4	264		
5	265		
6	266		
7	267		
8	270		
9	271		

ASCIIコード

キー操作			記号	コード
Shift	Control	(a	NULI IDLE	200
	Control	A	SOM	201
	Control	B	EOA	202
	Control	C	EOM	203
	Control	D	EOT	204
	Control	E	WRU	205
	Control	F	RU	206
	Control	G	BELL	207207
	Control	H	FE	210
	Control	I	HT(TAB)	211
	Control	J	LF(Line Feed)	212
	Control	K	VTAB	213
	Control	L	FF(FORM)	214
	Control	M	CR(carriage Return)	215
	Control	N	SO	216
	Control	O	SI	217
	Control	P	DCO	220
	Control	Q	DC 1 (X ON)	221
	Control	R	DC 2 (TAPE)	222
	Control	S	DC 3 (X OFF)	223
	Control	T	DC 4 (NOT TAPE)	224
	Control	U	ERR	225
	Control	V	SYNS	226
	Control	W	LEM	227
	Control	X	SO	230
	Control	Y	S 1	231
	Control	Z	S 2	232
Shift	Control	K	S 3	233
Shift	Control	L	S 4	234
Shift	Control	M	S 5	235
Shift	Control	N	S 6	236
Shift	Control	O	S 7	237
Shift		K	[	333
Shift		L	／	334
Shift		M	]	335
Shift		N	↑	336
Shift		O	←	337

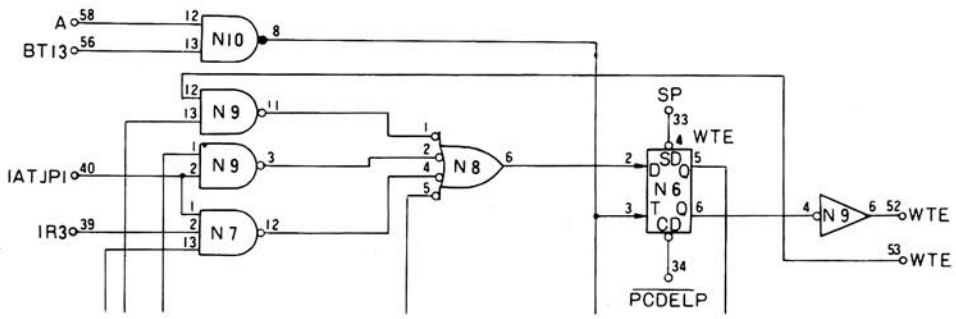
3．PUのロジック

3—1　ロジック図，フローチャート，タイムチャート

この章でのロジックの説明は，CEC—555S型電子計算機回路図（別冊）と附表のフローチャートおよびタイミングチャートを参考にして行う．ロジックの説明で，パッケージ番号（例C61—50—01）を引用するが，これは別冊の回路図の冒頭に示されるパッケージ配置図を参照されたい．

ロジック図は主として，インバータ（バッファ），NAND，D—Tフリップフロップより構成される．

下記に，WTEのロジック（実際とは少し違う）を示しているが，このロジックにおいて，N6，N7，N8，N9，N10はパッケージ内のIC番号を，その周辺の数字はICのピン番号を，また40%——はそのパッケージのピン番号をそれぞれ示している．N6のDは入力を，Tはトリガパルス入力，Qは出力，$\overline{Q}$はそのNOT出力を示している．SDは直接セット入力，CDは直接クリア入力である．N9，N10は通常のNANDでN8は$\overline{A}+\overline{B}=\overline{AB}$であるからNANDでいるが，出力が1レベルのNANDである．N7はインバーターで入力信号をNOT信号に変えるとともにバッファの役目もしている．NOT信号は頭に－を附する．N10の  の黒丸印は，ファンアウトの大きいNANDを意味する．



3—2　タイミング

PUを働かせるときに用いられるタイミングには2つある．各ワードタイムにおいて，連続的な制御を行うために，ビットタイムパルスを発生させるタイミングと，計算機の命令，周辺機器からの割り込み，コンソール機能の実行を制御するための異ったワードタイムを選ぶタイミングである．

主フローチャートの中央上部に見られるように，電源をON，OFFしたときに$\overline{\text{PPC}}$（PROCESSOR，POWER，CLEAR）のレベルが変るが，このレベルが0に落ちたとき，制御用のフリップフロップをクリアし，計算機をフェッチタイムにし，タイマー（リングカウンタ）Tを最初の状態にリセットする．この$\overline{\text{PPC}}$レベルは電源よりくる．この電源からはPUに＋5V，－5Vメモリーに＋20V，＋12V，＋5V，－5Vが供給されている．これらの電圧が規定の範囲内を越えたときはいつでもメモリーとプロセッサにパワークリア信号を出すようになっている．電源が入り，電圧が上りモータ，リレー等が働き始めPUが作動状態になってから約100msの間はメモリーおよびプロセッサのクリア出力は0レベルを保持している．まずメモリーへのクリア出力（$\overline{\text{MPC}}$）が1レベルに上り1～5ms後$\overline{\text{PPC}}$のレベルが1になる．PUが作動中に，電圧が一定値を保てなくなったときは，PPCは9～16msの間0レベルに落ちPUをクリアし1～5ms後に$\overline{\text{MPC}}$がメモリーの制御回路をクリアする．

電源はまた，低電圧状態のときに出るフラグ（LPC）があり，これは電圧が下った後で自動的に再復帰するときに用いられる．電圧が下ってから100msの間にLPCが0レベルに落ち，プログラム

インタラプトが可能であればLPCの変化がプログラムインタラプトの要求を出す（C61—50—11左上端部）.

この方法でパワークリアレベルが回復する前にプログラムは少なくともmsの間進むことができ，この時間内に，PUの種々のレジスタの内容を記憶させ，ロケーション0にジャンプする．パワークリアレベルが1に上ったとき，LPCが1レベルに上り計算機は0番地から再スタートする(オプションとして再スタート機能がついている場合).

3—2—1　ビット・タイミング

プリント板C61—50—13（CONTROL—2）にクロックを発生させるロジックを示し，C61—50—14（BIT TIMING）にタイマー（リング・カウンタ）と，タイマーをデコードしてビットタイムパルスを作るためのロジックを示す．C61—50—14に示されるクロック（信号名A）はメモリーサイクルの間を除いては休みなくパルスを発生する（メモリーサイクルに入るときにワードタイムを止めるためWTSがセットされる）．WTSがリセットされていて，RUNのフリップフロップ（C61—50—12CONTROL—1右上端部）がセットされている限り，リングカウンタTを右へ回すAパルスが連続して出る．C61—50—14からわかるように，T6の逆信号$\overline{T6}$がT0の入力として入り，他のビットへシフトされていく．

LOAD PC，STORE，READキーを押したときに出る最初のパルス（DELPタイミングチャート1参照）がタイマーの各フリップフロップをリセットする．T0の入力である$\overline{T6}=1$（STARTのキーを押したとき）の最初の状態から，Tのすべてが1になるまで，AパルスはTを左から順々に1にしていく．そしてさらにTのすべてがクリアされるまで0が左からシフトされていく．Tの状態によってビットタイム（BT00〜BT13）のレベルが決まるが，これについてはC61—50—14のロジック図および主フローチャートの左上端部およびタイムチャート2，3を参照されたい．

ビットタイムレベルが同じ番号をもつタイムパレスを作る．例えば，Tがクリアされるとと BT00は1になり次のAパルスでA00を発生する．それと同時にそのパルスがT0をセットしBT01を出す．PUのサイクルにおいてタイマーの初期状態は最後のビットタイムに依存している．従って最初のパルスは実際はA13であり，このパルスが次にくるワードタイムを選ぶ．

計算機がワードタイムの始めにスタートできるように，コンソールのSTARTキーによって出る最初のパルス（$\overline{ST}$）がT6をクリアする．タイマーが最初の状態にあるとき，即ちすべてのビットがクリアされているときのみオペレータがスタートキー押しているといえる．C61—50—12Bの左下部のロジックがTRレジスタをデコードしてタイムパルスをつくっている．これらのパルスはコンソールの操作キーを押すことによって作られる（タイムチャート1参照）.

3—2—2　ワードタイミング

C61—50—04Aに示される6つのフリップフロップが6つのワードタイムの実行を制御している．このロジック図とともに5—3ワードタイム状態遷移図をも参照されたい．C61—50—04において下から二番目のフリップフロップWTIは命令のアドレス部分がオートインデックス指定であるかどうかを決めている．各ワードタイムの終りで，各フリップフロップ出力（Q）の行先が次に入らねばならないワードタイムを決めている．セットされているワードタイムのフリップフロップがA13（AパルスとBT13によってつくられる）によってクリアされ，次のワードタイムのフリップフロップをセットしている．ワードタイムの直接の出力をインバータを通しバッファされすべてのロジックに用いている．フリップフロップの名前がそのままこれらのバッファされた出力信号名としてつけられている．バッファされた場合は逆信号となる（例．WTEの出力をバッファした場合WTEとなる．C61—50—11に示されるAIフリップフロップ入力ロジック網のゲート入力がすべて1のとき（即ちMA00〜04まで0，MB05〜MB07まで0で，MB08が1のとき），AIの入力ゲートは開かれる．オートインデックスロケーション10〜17の内の1ロケーションを間接アドレスとして指

定した命令のフェッチタイイムでAIがセットされる．AIをセットするパルスはA12で，A13でクリアする．またA13パルスは次のワードタイムのフリップフロップをセットし現在セットされているフリップフロップをクリアする．

PUがメモリーに1ワードを書き込むべきワードタイムになったとき，C61—50—05の左上部に示される**DC**．**WTI**．**WTB**．**WTX** IZDS等WTWRのレベルを1にする．同様に読み出すべきワードタイムにはWTRDを1にする（ロジック図には示されていないが，**MRRD**および**MRWR**の入力ゲートである）．メモリーのいずれかのサイクルが必要になったとき，BT12のタイミングで**MRRD**．**MRWR**が生じ，C61—50—14Aの左下部のWTSフリップフロップをセットしてPUのワードタイムおよびビットタイムを作るAクロックを止める．その後のBT12DDの（BT13に相当する）タイミングでメモリードまたはライトの要求信号（MRR/W，MRC/W）を送る．この信号メモリーが受けるとメモリー側のクロックが動き始める（C61—51—03C）．それと同時にメモリーからPUにBUSY信号が出される．メモリーサイクル中はこの信号は1レベルにあり，PUへはC61—50—14AにＭＧＯと信号を変えて入り，WTSのセットを維持する．メモリーサイクルが終ったときMGOは0レベルに落ちWTSはクリアされAクロックはA13からスタートする．

3—3　レジスタ

計算機のレジスタは種々の用途に用いられる．以下レジスタの各機能について詳細を説明する．レジスタはすべてシフトレジスタで右へシフトできるように各ビットが隣りのビットにつながっている．ある場合にレジスタのフリップフロップのセットダイレクトおよびクリアダイレクトが並列転送（SRからACへ転送するように）を行ったり，レジスタをクリアするときに用いられる．あるレジスタの内容を変えたり，他のレジスタに移したりするときはいつでも，情報が右へシフトされる．レジスタからの情報は右端のビット（LSB）から他のレジスタの左端のビット（MSB）にシフトされる．シフトしているときに入力がなければレジスタはクリアされる．しかしたまたまレジスタの入力とレジスタの出力が同じになることがあるが，これは情報を他のレジスタに移していると同時にそのレジスタ自身が循環していることを意味している．IRを除くすべてのレジスタは12ビットであるがACリングは13ビットのレジスタとして作動する．IRは4ビットである．

3—3—1　インストラクションレジスタ（IR）

各命令の実行中に，命令コードと間接指定ビットを保持しているレジスタがインストラクションレジスタ（IR）で，C61—50—02に示される，IRはコンソール操作の始めとか，プログラムインタラプトのときのような特殊な状態のときにクリアされる．各命令がメモリーからMBに読み出され準備されると，MBの内容が標準の12ビットシフトによってIRにシフトされる．このシフトによってワードの左4ビットがIRに残り右の8ビットのアドレス部分は押し出されてなくなる．

IRの左の3つのビットがデコードされ命令を決定する．デコーダ出力の1つはREADとSTOREが実行されているときを除いては，ある命令を指定しているため1レベルにある．IR03はオートインデックスの場合とオートインデックスでない場合の間接アドレス指定を選ぶためにAIの2つの状態（AIと$\overline{\text{AI}}$に関係している），C61—50—02の右上にあるQPレベルはOP1命令かOP2命令かを決定する．これもIR03の2つの状態によって決まる．OP1レベルはさらに特殊な命令を決めるためにMBのビットと結びついている（C61—50—03A参照）．

C61—50—03A（IR DECODER）に示されるように，ROTRはOP1，MB08が1で，MB10が1か，PUが最初のビットタイム（BT00）でないときのいずれかであるとき1レベルになる．このレベルとBT（00—01）によってAパルスがゲートされA（00—01）がACのシフトパルスとして発せられる（C61—50—06）．もし命令コードのビット10が1なら2つのシフトが生じ，ビット10が0ならBT01によって1つのシフトが行なわれる．

左へシフトするゲートはないから左へのローテーションは通常より短い右シフトによって作られ

る．ACとLは同時にシフトされるので完全なローテーションには13回のシフトが必要である．左へのローテーションのためA（00—11）によって右に12発シフトされる．L（リンク）を通してシフトしているから結果として左へ1つシフトしたことになる．MB10が0か，BT00が0のいずれかのときで左へのローテーション（OP・MB09＝1）のときに，ROTLが1になり，これによってシフトが制御される．もし命令のコードのビット10が1なら，ROTLはBT00以外では1になり，12回の右へのシフトが1回除かれ結果としてACおよびLを左へ2つシフトしたことになる．

3—3—2　メモリーバッファ（MB）

MB（C61—50—01A・ロケーション18）はPUとメモリーとの間のバッファなので，MBレジスタはリード　ライトサイクルの要求信号（MRR W）によってクリアされる（C61—50—07の右下部，MBCLR）．そしてメモリーから読み出された1ワードの並列転送が個々のMBビットをセットするメモリーの出力パルスによって行われる．MBレジスタのクリアはSTARTキーを押したときにも行われる．MBの各フリップフロップの直接の出力（Q）はシフトのための入力として用いられる．レジスタの出力はバッファを通しPU内の各コントロールおよび外部I/O用に送られる．I/O機器用のMB信号は1レベルのものと0レベルの両方がデバイス選択用として，6本および7本がバッファされて（$\overline{\text{MB03}}$〜$\overline{\text{MB08}}$とMB03〜MB09）背面コネクタにつながっている．

MBレジスタのロジック図（C61—50—01A）の左端部にMBシフトパルスおよびMB00へのシフト入力が示されている．MBはPCまたはACの内容を受け取り，それをメモリーに書き込んだり，インデックス動作のときはPCまたはACの内容を加算器を通して受け取る．またフェッチタイムでMBの内容がIRとMAに移されている間，MBはそれ自身の内容を受け入れる．即ち循環する．C61—50—07の左側に示されるように，MBをシフトするシフトゲートおよび入力ゲートはAND，TAD，JMP命令において，同時に開かれる．これらの命令はエクゼキュートタイム（WTX）とエンドタイム（WTE）が同時に起る．この2つのワードタイムにおいて，DCA，ISZ，JMSの3つの命令の場合のみMBに入力が供給される．そしてその内容が命令の実行に必要なものであるとき（IOTのデバイス選択とOPの各オペレーション選択のとき）を除いては入力ゲートを閉じ，シフトゲートを開くことによってMBをクリアする．

3—3—3　プログラムカウンタ（PC）

このレジスタ（C61—50—01Aロケーション20）はJMP命令のときに，MAからアドレスを受け取る．フェッチタイムでPCがインデックス（1インクリメント）されるとき（READとSTOREキーを押したとき），ブレークタイムでPCがアドレス0001を受け取るとき，そして（LOAD，PCキーを押したとき）SRからアドレスをACに移し加算器を通してPCが受け取るときは常に加算器の出力がPCの入力に入るようにゲートが開かれる．エンドタイム（WTE）の中，その中のインデックスが要求されるときは（スキップまたはJMS），いつでも，加算器のS出力をPCが受け取る．WTEでPCが，MAの内容を受け取るJMP命令はWTXとWTEが同時に起るので除かれるが，SKPまたはJMS以外の他のすべての命令はWTEにおいてPCはそれ自身で循環する．

3—3—4　メモリーアドレス（MA）

C61—50—01A（ロケーション19）に示されるMAレジスタは，メモリーサイクルにおいて，メモリーの4096ロケーションの1つを12ビットによって選ぶためのレジスタである．C61—50—09（ロケーション11）の左中央部にページゼロ（PGZ）のフリップフロップがあるが，これは命令コードのビット4が0であれば，フェッチタイムの最初のパルス（A00）によってセットされる．C61—50—07（ロケーション16）の左下部に示されるように，シフト入力（MAINS）にMBから7ビットのアドレスが最初の7つのパルスによって供給されるが，もしPGZフリップフロップがセットされていなければ，同じワードタイムの残りの時間BT（07—11）で前のワードタイムで入っていたMA00〜MA04の内容がそのまま循環される．PGZが1なら，MAの入力が禁止されMAはページ

ゼロのアドレスを指定する．MAはデファータイム（WTD）において，MBから12ビットのアドレスを受ける．またコンソールキーのREADおよびSTOREにおいては，メモリーアクセスをするためにMAはPCの内容を受け入れる．JMP命令においてMAの内容をPCに転送するが，それと同時にMA自身が循環している．

JMP命令を除くエンドタイム（WTE）では，次の命令が記憶されているアドレスを指定するためにMAはPCの内容を直接かまたは加算器を通し1インクリメントして（スキップの場合）受け入れる．ブレークタイムにおいてMAレジスタはMAの入力なしでシフトされるのでこのときのみクリアされる．

3—3—5　アキュムレータ（AC）とリンク（L）

C61—50—01A（ロケーション21）とC61—50—08（ロケーション22）にACとLを示す．これらを制御するパルスは別々に分れているが，加算とローテートのときは13ビットのレジスタとして一緒に働く．ACの12ビットはOP命令の2つ（7260と7400），IOT命令（例，6032），パワークリア（$\overline{\text{PPC}}$），コンソール操作(CONTINUEを除く)によって1度クリアされる．LOAD PC，STORE，またはOSR命令の場合，SRの内容をACにロードするためACRSRパルス（C61—50—13右上端部）を発生し，SRのONの部分によってゲートされ，個々のACビットをセットする（C61—58—02C，CONNECTOR—Bロケーション21参照)，外部周辺機器からのデータをACにロードするのは外部からの信号ありの部分だけがACを直接セットする．またバッファされたACの出力が周辺機器への情報の転送のためにバスラインに接続されている．ACのシフト入力ACINSは61—50—06の左上部，ACのシフトゲート（この出力とAパルスとBT（00—11）のANDがとられてシフトパルスとなる），ACSHGは右上部に示すように作られる．大抵のシフトは標準のA（00—11）パルスによって行われるが（ROTLはこれらの1つが除かれる），TAD命令のエクゼキュートタイム（WTX）では13ビットの加算を行うため（Lが含まれるので）A12によるシフトが1つ追加される．そして右へのローテーションにはA（00—01）を用いて1つ，または2つのシフトが行われる．

ローテート命令を行っているとき，LはACから情報を受け取り，ACは逆にLから情報を受け取る．TAD命令においてLは加算器のS出力を受け，これをACにシリアルに入れる．IACとCMA命令において，ACとLは異った時間に，加算器の出力を受け入れる．IACではACのインクリメントによってオーバフローが行われた場合A12でLをコンプリメントする．CMA命令では加算器を通してACはコンプリメントされるが，LへのCMA入力は不必要なものとなり，Lが単にそれ自身で循環している．AND命令ではACにシフトされたワードがMBの内容との間で1ビットごとのANDがとれ，その結果がACに入る．STOREの場合，ACの内容がMBにシフトされるが（READのときのシフトは不必要である），このときAC自身は循環している．OP2命令の中で，ACの内容が0かどうかテストする命令では，加算器を通してACがもとの内容を受け入れている．

C65—50—06中央部にリンクの制御パルスについて示す．

リンクをシフトするパルス$\overline{\text{LCLK}}$はIAC命令とCMA命令のA12のタイミングのときを除けば，ACのシフトパルスと同時に出る．Lがクリアされるのは，(1)電源投入時（PPCによる），(2)STARTキーを押したとき（SPによる），(3)Lがクリアされる命令が呼び出され実行されたとき（例7100），(4)Lが1にセットされていて，Lをコンプリメントする命令が実行されたときである．またLがセットされるのはLがクリアされていて，コンプリメントせよという命令が実行されたときと，クリアしてからコンプリメントせよという命令が実行されたときである．

3—4　制御回路

PUには，レジスタとタイミングロジックに加えて，直列加算器，スキップのロジック，コンソール関係の回路が含まれている．

3—4—1　加算器（F.A）

直列加算器（C61—50—08 FULL ADDER & LINK　ロケーション16）は1シフトパルスごとに，1対の数の和をつくる．各ステップごとに加算器には3つの入力がある．2つは加数および被加数のビットで他の1つは前のステップでの桁上げされたビットである（最初の段階では桁上げのビットはもちろん0である）．各ステップで2つの出力がつくられる．1つはX入力とY入力のビットごとの和S，もう一つは桁上げCであり，Cは桁上げ用フリップフロップに貯えられる．S出力は鉛筆と紙で行う2進法の加算と同じような方法で出てくる．和が奇数か，1つの入力だけが1か，3入力がすべて1なら和Sは1である．桁上げは和が2（10_2），即ち2つの入力がともに1であれば1である．

C61—50—08の左上部に加算器のX入力を示す．この入力は，TAD命令のときに，ACより入ってくるか，その他の実際に加算が行われるすべての場合に，XはBT00のときのみ1で，Y入力として入ってくる数に1を加えた結果になる．READとSTOREのときにPCを1インクリメントするのはX入力のBT00のゲート（DEF）が開かれて行われる．インクリメントが要求される他の命令と，ブレークタイム（WTB）でPCに0001を供給する場合BT00のゲートはWTINCR（C61—50—05の右下部）によって開かれる（この場合のY入力は0である）．X入力がなければY入力の値がそのままSにシフトされて出てくる．この操作はACをコンプリメントしたり，ACの内容が0であるかどうかテストしたり，ACの内容をPCに転送したりするときに用いられる．

Y入力にシフトされてくるのは，(1)READまたはSTOREのときのPCからと，(2)インデックスまたは加算のときのMBからと，(3)LOAD PCまたはOP命令のスキップのときのACからである．WT INCPC（C61—50—08左中央部およびC61—50—05右中央部）はフェッチタイム（WTF）で通常のプログラムカウントするときと，SKIPとJMP命令のエンドタイム（WTE）のときに，PCの出力（PC11P）をゲートして加算器のY入力とする．残りのY入力へのゲートはOP命令のIACとCMAのときにY入力を制御している．即ちC61—50—08に示される各ゲートがACからの情報をY入力に入れる制御をしている（CMA命令のときは，ACの内容の各ビットの補数，即ち$\overline{\text{AC11}}$をゲートを通してY入力としている．ただしLの内容をシフトしY入力とするビットタイムBT12の時を除く）．CMA命令のときのリンクシフトは必要ないが，IAC命令において桁上げ（オーバフロー）があるときは，LはA12によってコンプリメントされる．MB11が1でMB06が0のとき，IAC用のゲートが開かれる．加算器がコンプリメントされたものに1を加えるので，このときプログラムはコンプリメントとインクリメントを同時に行ったことになる（CIA命令の実行）．

C61—50—09の左下部のゼロインジケータのフリップフロップ（ZI）はフェッチタイム（WTF）で最後のパルス（BT12DD　タイミングチャート1参照）でセットされる．ISZまたはOPのスキップ命令のエクゼキュートタイムでもし加算器の和が1ならZIはクリアされる．

3—4—2　スキップロジック

SKPフリップフロップはC61—50—09（ロケーション11）の左上部に示される．このフリップフロップはスキップするために，エクゼキュートタイム（WTX）でPCのインクリメントを制御する．I/Oからスキップパルス（I/O SKP）は直接このフリップフロップをセットする．フリップフロップの左側にある入力網がOP命令のスキップ条件をつくり出す．

命令コードビット5の1がACの内容が負数（AC00がセット），ビット6の1がACの内容が0（ZIがセット），ビット7の1がLのセットをそれぞれテストする．その選択が命令の選択ビットを入力とする3つのANDゲートによって作られる．

完全なスキップ条件は，C61—50—09中央上部に示される，2つのNANDゲートの出力のOR出力によって決められる．命令コードのビット8は上記の選択ビットの逆条件でスキップするかどうかを指定する．従って最下位のNANDゲート（N4ピン3，4，5，6）が$\overline{\text{MB08}}$と同図の左上部での選択されたゲート出力（N1の8）を受け取る．一方その上の，NANDゲート（N4ピン番号

1，2，13，13）がMB08と選択されたゲート出力（N1の8）のインバート出力（N3の2）を受け取る．ビット8に0がプログラムされ，ビット5～ビット7によって選択された条件（N1の8）が1なら，SKPのQ入力が0になる．同様に選択条件が0でビット8が1ならSKPが0の信号を受け取る．従って左上端部の入力を満足しビット8に1がプログラムされればOP2命令のエクゼキュートタイムでA12パルスがSKPフリップフロップをセットする．このフリップフロップが1の状態にあるとき，OPまたはIOT命令のエンドタイムにおいて，特別にPCのインクリメントを起こさせる．

3—4—3　コンソール

CONSOLE CONTROL SWITCH（図番215644）にコンソールの回路図をC61—50—13にコンソールからPUへの入力を示している．またC61—50—12Bにコンソールのキーが計算機をRUNのフリップフロップを通してスタートさせたりストップさせたりするロジックを示している．C61—58—02C（ロケーション21）にはSRの内容がACRSR（C61—50—13右上部で作られる）パルスによってACに並列に転送されるロジックを示している．コンソールのスイッチレジスタの下部に操作用のキーとスイッチがある．その中のロックスイッチが閉じていれば（OFFの状態），各キーおよびスイッチのCOMMONがグラウンドに落ちるようになっている．ロックスイッチが開いていれば（ONの状態）キーおよびスイッチは働かない．

（CONSOLE CONTROL SWITCH　回路図参照）

STOPキーを除くキーがすべて閉じられると（キーが押されると），C—61—50—13の左端部に示されるR—Sフリップフロップを通して長いトリガパルスを発生する．しかし計算機がRUNの状態にあるときは，このパルスが発生しても計算機には何らの影響も与えない．R-Sフリップフロップからの各出力（$\overline{\text{FCONS}}$，$\overline{\text{FDEPS}}$，$\overline{\text{FEXS}}$等）がTPカウンター（C61—50—12B左上部）に入り，各カウンターの出力がRUNのQ出力（即ち$\overline{\text{RUN}}$）によってゲートされているためである．CONTINUEの場合はこのキーが押されると，TPがRUNをセットする．STARTの場合はメモリーサイクルのためにPUが止っているのでなければTPがSTARTパルスSPをつくる．SPパルスは大半のフリップフロップをクリアする．コンソールの操作によって発生するパルスおよびその働きについてはタイミングチャート1，2，3（附図）を参照のこと．

LOAD PCについてはTPが1対のロードパルス（LPとLPA）を340NSの間をおいて発生させる．同じような1対のパルスがSTORE，READのときも発生する．それはLPとLPA，EPとEPA，DPとDPAである．遅れた方のパルスがC61—50—12の右下部に示されるコンソール操作用のワードタイムフリップフロップ（LD，EX，DC）の1つをセットする．これらのワードタイムの最後に計算機を止めるためにA12でRUNをクリアし，その後のA14相当パルス（BT12DYとCLKのAND，コンソール操作のときにのみ用いられるパルス）で，これらのフリップフロップをクリアする．

HLT命令では，エンドタイム（WTE）のA12で，RUNのフリップフロップをクリアしプログラムを停止する．RUNフリップフロップの入力ゲート（C61—50—12B中央上部）からわかるように，STOPキーかSTEPスイッチがONであれば（このパッケージのピン番3にあるSTOSIKが1であれば）フェッチタイム（WTE）のA12（BT12とAのAND出力）でRUNがクリアされる．

3—5　PUの命令およびデータの流れ

この節ではメインフローチャート（5—2）を参考にして，コンソールの働き，命令の流れについて説明する．

3—5—1　コンソールの操作

メインフローチャート（5—2）の右上部に示されるように，計算機の動作はコンソールキーによって始る．CONTINUEでは計算機を走らせるためトリガパルスがRUNをセットする．

LOAD，PC，STORE，READにおいては，TPが1ワードタイムの間にPUを働かせる1対のパ

ルスを発生させる．3対のパルスの内の最初の各パルス（LP，DP，EP）がAC，WTF，WTEの各フリップフロップをクリアし，タイマー（T）をリセットする．さらにLPはワードタイムとIRそして他の制御用のフリップフロップをすべてクリアする．第2番目のパルス（LPA，DPA，EPA）の各々がRUNおよびコンソール操作によって作られるワードタイム（LD，DC，EX）のフリップフロップをセットする．DPAとLPAはSRの内容をACに転送するために用いられる．

LOAD PC，STORE，READの3つの働きを行うために，これらによって作られたパルスが3つのワードタイムをつくる．

LOAD PCにおいてA（00—11）の12発のパルスがSRにセットされたアドレスをAC（ACにはSRの内容が前に出たパルスLPAによってセットされている）から加算器を通してPCにシフトする．他の2つの操作（STORE，READ）では，この12発のパルスがPCの内容をMAにシフトし，同時にPCをインクリメントする．STOREの場合はさらにSRの内容をAC（SRの内容がDPAパルスによってすでにACにセットされている）からMBに移動させる．シフトが完了した後でA12がRUNをクリアし，A14相当パルスがコンソール用のフリップフロップ（LD，DC，EX）をクリアし，同じA14相当パルスがPUをフェッチタイムにする（WTFをセットする）．メモリーアクセスを必要とするREADとSTOREにおいてはA12が$\overline{\text{MRRD}}$または$\overline{\text{MRWR}}$をつくりWTSをセットする．これによってPUのAクロックパルスを止め，その後のA13によってメモリーリクエスト信号（MRR/W，MRC/W）をつくりメモリーに送ってメモリーサイクルに入る．読み出しの場合は，メモリーからのワードの読み出しを準備するためにMRR Wがメモリーへ送られると同時にMBをクリアする．

STARTキーを押すことによって，PCに指定されたロケーションの命令を読み出して計算機は正常の動作を開始する．もし現在メモリーサイクルが行われていなければキーからのトリガパルスが，スタートパルス（SP）を発生させる．SPによって計算機はエンドタイムに入り，SPがT6をクリアする．これと同時にメインフローチャート（5—2）の右上端に示されるように，入出力装置およびMB，AC，IRそしてPUなどの制御用フリップフロップをクリアする．最後にAクロックパルスをスタートさせるためにRUNをセットする．

（タイミングチャート1—2参照）

電源が瞬断後，ある一定のレベルに電圧が上ったとき，$\overline{\text{PPC}}$，$\overline{\text{LPC}}$が1になり$\overline{\text{AST}}$がつくられる．この$\overline{\text{AST}}$はSTARTキーを押したときにつくられるトリガパルスと同じ働きをする．

3—5—2　命令の流れ

メインフローチャート（5—2）の右上方部のループ図に，メモリーサイクル時以外は常に発生しているAパルスを示している．コンソール操作によってRUNがセットされるなら連続のクロックパルス（CLK）がAパルスをつくる．このAパルスが種々のビットタイムをつくるタイマーTを作動させている．メインフローチャートにおいてAを起点として右へ1本，あるいはそれ以上の線でつながり，これらはすべて各BT信号（BT00，BT（00—01）BT（00—11）等）が1のときである．そしてこのライン上に単一のワードタイム（WTF，WTX等）があり，その下にPU内の動作を表す．オートインデックスに必要な構成がMAとMBに存在しているならば，A12によってAIフリップフロップがセットされ，STEPがONになっていればRUNをクリアする．またA12によってタイマーTをリセットし，インタラプトイネーブル（ION）がONであり，低電圧状態（LPC）であれば，またI/Oバスからのインタラプトがあればプログラムインタラプトフリップフロップをセットする．そしてA13が次のワードタイムに移るために，多くの制御用フリップフロップをクリアする．

LOAD PC，STORE，READにおける特殊なワードタイムについてはすでに討論された．STARTキーが押された後で，PUはエンドタイムを実行し，PCの内容をMAに移し，最初の命令を読み出すために，リード/ライトサイクルをメモリーに要求する．

3—5—2—1　フェッチタイム（WTF）

命令がメモリーから読み出されると，その命令のビット4が0の場合にはA00がPGZフリップフロップをセットする．

プログラムの最初にION命令があればA01がインタラプトイネーブルをONにする（即ちIONを1にする）．ワードを転送するパルスA（00—11）が命令のオペレーションコード（3ビット）とインダイレクトビット（MB03）をIRに移すと同時にPCをインクリメントする（PCの内容を＋1してPCに入れる）．これらのパルスの最初の7発によってMBのアドレス部分をMAに移し，次の5発によってカーレントページの数（アドレスの先頭分）をMAからMAに入れるか，MA00—04をクリアする（0ページの場合）．このいずれかはPGZがセットされているかいないかによる．もしIRにOP1命令のコードが入っていて，ACおよびLをクリアしたり，セットしたりすることが指定されている場合はこの操作をA12またはA13で行う．STOPキーが押されたり，STEPのキーがONの状態でCONTが押された場合はA12が計算機を止める．命令が間接アドレスを指定して，新しいアドレスを読み出すためと，AND，TAD，ISZのオペランドを読み出すためにA13がリード／ライトサイクルをメモリーし要求する．このメモリーへアクセスする状態がメインフローチャートの中央下部に示される．

Aクロックが再スタートしたときに，A13がWTFをクリアし，ZIをクリアして，PUを次のワードタイムに移す．命令が間接にオートインデックスロケーションを指定するならA13がWTIをセットする．AIがクリアされた状態での間接アドレス指定であればフェッチタイムの後にデファータイムが続く．命令がOPかIOTであるか，直接アドレスを指定する命令であれば，A13がWTXをセットする．そして命令がOP1であるか直接アドレス指定のAND，TAD，JMP命令であればA13がWTEをセットする．

3—5—2—2　インデックスタイム（WTI）

このワードタイムはMBに読み出されたアドレスをインデックス（インクリメント）し，インクリメントされたアドレスをメモリーに再書き込みするために，クリア／ライト信号（MRC/W）を出す．そしてPUはデファータイム（WTD）に入る．

3—5—2—3　デファータイム（WTD）

このワードタイムで絶対アドレスをMBからMAに移す．そしてもし命令がAND，TAD，ISZのいずれかであればオペランド（データ）をメモリーから読み出す．PUはこの後エクゼキュートタイム（WTX）に入るが，AND，TAD，OPI，JMP命令の場合はエンドタイム（WTE）に同時に入る．

3—5—2—4　エクゼキュートタイム（WTX）

右へローテーション命令（RAR，RTR）において，A（00—01）の内の1発または2発がその要求に応じてACとLを一緒に右に1つ，または2つシフトする．このワードタイムでA（00—11）パルスが6つのメモリー参照命令に必ず用いられる．例えば，TAD命令はMBを加算器のY入力へ，ACをX入力へそれぞれシフトして，その和（S）をLを通してACに戻す．ISZ命令ではX入力がBT00のときのみ1で，MBがY入力にシフトされ，加算器によって＋1されてMBに戻る．もし加算器の出力に1のビットが1つでもあれば，ZIはクリアされる．

OP命令に要求される全ワード操作は，メインフローチャートにおいて12発のシフトパルスのWTXラインに表われる．IACはACに1を加え，CMAはACをコンプリメントする．いずれも加算器を通して行われる．この両方が一緒に指定されると（CIA命令），加算器が$\overline{AC}$を受け取り，IACによってBT100で1が加えられる．結果としてACの内容の2の補数にする．左へのローテート命令では，ACとLが左に11回または12回シフトされる．テストスキップグループ命令（SNL，SZL，SZA等）では，ACは加算器を通してローテートされ，そのS出力が1なら$\overline{ZI}$がセットされる．12ビ

ットのシフトの後で，命令を完成させるため，要求される加算操作はすべてA12で行われる．TAD命令で加算器を通して12ビットの結果をACに入れるが，最後のビットの加算で桁上げが行われる（このとき加算器の入力は桁上げとLで，LはACを通してシフトされてくる）．この場合最初の状態のLをコンプリメントするが，これはA12で行われる．このワードタイムで，AND，TAD，ISZはメモリーから読み出されたオペランドのパリティをチェックする．ISZ，DCA，JMSはオペランドをメモリーに書き込む．IACもACがオーバーフロー（桁上げ）した場合はLをコンプリメントする．スキップグループ命令コードのビット4が1ならA12によって，ACをクリアする．そして指定されたスキップ条件が満足されるとA12がSKPのフリップフロップをセットする．このエクゼキュートタイムでエンドタイムが同時に起きていなければ，A13によってWTEのフリップフロップをセットしてエンドタイムに入る．

3—5—2—5　エンドタイム（WTE）

OSR命令（ビット9が1）では，A01がスイッチレジスタSRの内容をACに転送する（C61—50—13右上部およびC60—58—02C参照）．このワードタイムでは次の命令を読み出すためにA（00—11）によってアドレスを決定する（PCをMAに移す）．JMS，ISZ，またはOPのSKP命令ではPCの内容をインクリメントしてMAに移す．SKPおよびJMP命令でなければPCはそのままMAに移される．JMP命令でMAからPCに新しいアドレスを移す場合はエンドタイムと同時に起るエクゼキュートタイムで行われる．エンドタイムで，MAはMA自身循環する．

A12でSKPフリップフロップをクリアし，HLT命令（ビット10が1）では同じパルスでRUNをクリアする（C61—50—09右上部およびC61—50—12B右上部参照）．そして次の命令を読み出すために，A13によって作られたメモリーリード　ライト信号（MRR W）をメモリーに送る．Aパルスが再びスタートし，インタラプト要求がありPIRがセットされて，ブレークタイム（WTB）に入ることがなければフェッチタイムに移る．

3—5—2　6　ブレークタイム（WTB）

このワードタイムに入るとA01がIRをクリアし，A（00—11）がMAをクリアし（MA入力がなくて12回のシフトを行うため）現在のプログラムロケーションをPCからMBにシフトし，PCには0001を入れる．そして，PCからMBにシフトされたアドレスをロケーション0000に書き込むために，メモリークリア　ライトサイクルを要求する．ロケーション0001の命令を実行するために，ブレークタイムの後で自動的にエンドタイムに戻る．以下は通常通りのフェッチ，エクゼキュートの各ワードタイムに移っていく．

4．保　守

4—1　電源故障の場合のPUランプの状態

1．前面のランプが全部点灯しない．
（1）＋5Vショート
（2）ファストン端子抜けおよび接触不良（電源ライン）
2．電源スイッチを入れた時はランプが点灯するが，すぐ消える．
（1）＋5Vラインオーバーロード
3．ランプは点灯するが記憶しない．
（1）+12Vショート（コアメモリー）
（2）－5Vショート
（3）MPC不良
4．電源投入時，POWER CLRがかからない．PC，ACが点灯する．
（1）PPC不良
5．インターラプトがかからすいまたは動作不良である．
（1）LPC不良

4—2　C61　PKG不良の場合のPU正面ランプの状態

1．電源投入時，POWER CLRがかからない．
（正常の場合は電源投入時，INSTRUCTION，RUN，INTRQST，LINK，は点灯しない．またロータリースイッチをまわして各レジスターの内容をみると，PC，ACは0000，MA MBはランダムな数を表示する）

	不良PKG	Loc
（1）全体的に点灯	C61—50—13	7
	〔A〕項PPC	電源
（2）ＰＣが点灯	C61—50—01A	20
（3）ＡＣが点灯	C61—50—01A	21
	C61—58—02C	27
（4）ＩＮＳＴＲＵＣＴＩＯＮが点灯	C61—50—02A	12
（5）ＲＵＮが点灯	C61—50—12B	8
（6）ＬＩＮＫが点灯	C61—50—08	22
	C61—50—06	17
（7）ＩＮＴ　ＲＱＳＴが点灯	C61—50—21E	23
	C61—50—09	11
（外部I/O接続の場合は点灯することがある）		
（8）INTERRUPT ENBLが点灯	C61—50—11	10

(2)〜(8)項までは，PPC信号のGATEによって，F.FのCDでクリアされる．

2．電源投入後，DATAスイッチをセットし，LOAD PCを押しても，AC，PCにDATAスイッチの内容が表示されない状態
（MB11が1レベルの時のみ，ACに表示される）

（1）常に0の場合	C61—58—02C	27
（ACRSR・SR信号がICの入力レベルで不良）		
	C61—50—13	7
（ACRSR信号が不良，CLKが不良，LAS，FLAS不良）		

C61—50—12B 8

(LP、LPA、DELP、LD、DELF信号不良)

C61—50—06 17

(ACINS、ACSHG、ACCLR不良)

C61—50—04A 14

(WTF信号不良)

C61—50—14A 9

(A信号によってBT12DD、BT00—11、BT13信号不良)

C61—50—01A 21

(2) DATAスイッチの内容と異なる場合

C61—58—02C 27

(ACXXBUS信号不良)

C61—50—01A 21

(シフトレジスタ動作不良)

3. DATAスイッチの内容がACにセットできるが、PCにセットできない.

(1) PCの内容が常に0である場合

C61—50—08 22

(AC11PとLD信号によりS信号不良)

C61—50—07 16

(PCINS信号不良)

C61—50—05A 15

(PCSHG、**WBDFI**信号不良)

C61—50—01A 20

(シフトレジスタ動作不良)

(2) PCの内容がDATAスイッチの内容と異なっている場合

C61—50—01A 20

(シフトレジスタ動作不良)

4. LOAD PCキーを押した時、RUNランプが点灯する.

◎C61—50—12B 8

(TP0～TP2DまでのF, Fの動作不良)

C61—50—14A 9

(CFFC、BT12信号不良)

C61—50—13 7

5. STOREキーを押した時、MB、ACにDATAスイッチの内容が表示されない.

(正常な場合はSTOREキーを押した時、DATAスイッチの内容は、MB、ACに表示され、MAには、LOAD PCキーを押したときのDATAスイッチの内容が表示され、それ以降はSTOREキーを押すごとにPCとともにアドワンする.)

(1) STOREキーを押した時AC、MBにDATAスイッチの内容が表示されない.

C61—50—13 7

(FDEPS、**FDEPS**不良)

C61—50—12B 8

(**DPA、DC、DC、DEP、DELP、DEF**、DEF、DELF、**DELF信号不良**)

C61—50—06 17

(DC、AC11P入力のACINS出力不良)

C61−50−05A　15

(PC入力により、MRC W、WTWR信号不良)

(注) LOAD P. Cキーによってac、PCにDATAスイッチの内容が表示されたと仮定する.

(2) STOREキーを押した時ACにDATAスイッチの内容が表示されるがMBに表示されない. またDATAの内容と違っている場合

C61−50−07　16

(AC11PとDC入力によるMBINS不良、MBSHG 不良、MBCLR 不良)

C61−50−01A　18

(シフトレジスタ不良)

C61−50−05A　15

(**WBDFI**信号不良)

(3) STOREキーを押した時、最初の一回でSTOREキーを押す前のPCの内容がMAに表示されない.

C61−50−08　22

(PC11PとDEF信号により、出力S不良)

C61−50−07　16

(PC11PとDEF信号によりMAINS不良)

C61−50−05A　15

(**DEF**信号を入力とするMASHG不良)

C61−50−01A　19

(シフトレジスタ不良)

(4) STORE、READキーを押した時、PC、MAはアドワンしない. またアドワンしてもPC、MAの内容が正常な動作をしない.

C61−50−08　22

(DEFとPC11P、WTINCR入力によるS信号不良)

C61−50−07　16

(PCINS MAINS信号不良)

C61−50−05A　15

(WBDFLP、WTINCR、WTINCPC信号不良)

C61−50−09　11

(SKP命令不良)

C61−50−14A　9

(BIT TIMMING誤動作)

C61−50−01A　19. 20

(シフトレジスタ動作)

(5) STORE、READキーを押した時、PC、MAが同じくアドワンする.

C61−50−03A　13

(OP2信号不良)

C61−50−02A　12

(インストラクションレジスタ誤動作)

(6) STOREキーを押した時、PC、MAとともにACも同時にアドワンする.

C61−50−03A　13

(OP 2信号不良)

6．LOAD PCキーを押した時，MBの内容が0になる．

◎C61—50—05A　15

($\overline{\text{MRRD}}$　$\overline{\text{MRR/W}}$信号不良)

C61—50—14A　9

(BIT TIMMING誤動作)

C61—50—04A　14

(WTD信号不良)

7．READキーを押した時，MBに先に読ませた内容を表示しない．または，その内容が正常でない．

(正常の場合はREADキーを押した時，MBには先に読みこませた内容を表示し，ACは0である．またPC，MAはアドワンする．)

(1) READキーを押しても，MBに表示しない（0である場合）．

C61—50—12B　8

(EPA，EX，$\overline{\text{EX}}$信号不良)

C61—50—13　7

(FEXS，$\overline{\text{FEXS}}$不良)

C61—50—05A　15

($\overline{\text{MRR/W}}$，$\overline{\text{MRRD}}$信号不良)

(注) STOREキーにより，MA，ACにDATAスイッチの内容を表示し，PC，MAは正常にアドワンすると仮定する．

(2) READキーを押してメモリーより読み出してもMBの内容が正常ではない．

C61—50—07　16

(MBINS信号不良)

C61—50—04A　14

(WTI，WTB信号不良この場合MBは7777を示す．)

(3) READキーを押したとき，ACにDATAが表示される．

C61—50—06　17

(ACCLR信号不良)

C61—50—05A　15

(**MRR/W**信号不良)

8．LOAD PCキーを押した後，STORE，READキーを押すと，再びLOAD PCキーが動作しない．

C61—50—04A　14

(WTF信号不良)

C61—50—05A　15

(**MRR/W**信号不良)

C61—50—14A　9

(BIT TIMING不良)

C61—50—06　17

(ACCLR信号不良)

9．STOREキーを押した時，INSTRUCTIONが点滅する．

C61—50—02A　12

(INSTRUCTION RESISTER誤動作)

C61—50—12B 8

(**DEF**信号不良)

C61—50—04A 14

(WTB信号不良)

10. STARTキーを押した時，RUNしない.

(1) STRATキーを押した時，RUNしない.

C61—50—13 7

(STT，**STT**信号不良)

C61—50—12B 8

(**SP**信号不良)

C61—50—04A 14

(**SP**信号により，WTE，**WTE**信号不良)

(注) LOADPC，STORE，READキーの動作は正常と仮定する.

(2) STARTキーを押した時，STEPキーを押さないのに，PCが1個ずつアドワンする.

C61—50—12B 8

(STOSIK信号回路不良)

C61—50—14A 9

(BT12信号誤動作)

C61—50—04A 14

(WTF信号誤動作)

11. RUN状態にてSTOPのキーがきかない.

C61—50—12B 8

(STOSIK信号回路不良)

C61—50—01A 18

(MB10信号不良)

4—3 TTY関係 (TTYは良として)

1. CPU電源ONの状態でTTYの電源をONにすると，TTYが動作する．その時，PUはスタートしていなく，TTYのREADERはSTOPまたはFREE状態にある.

C61—50—20B 24

(ONLINEを入力信号とするTOTTY信号不良)

2. 電源ONの状態でTTYのREADERをSTARTにすると，READERが動作する．PUはSTOP状態である.

C61—50—19A 25

(READER RUN CLR信号不良)

C61—50—21E 23

(TRP信号不良)

3. PUをSTART状態にして，TTYのREADERをONにしてもREADERは動作しない.

C61—50—19A 25

(SKP信号不良)

4. PUの内容をTTYにPUNCHしない，またPUNCHはするがミスをする.

C61—50—21E 23

(**TTOCLK** 4.6ms不良，TTOFLG不良)

C61—50—20B 24

(TTO＝0、信号不良)

5．TTYから読み込みPUに読み込みしない．また読み込みミスをする．

C61—50—19A 25

(FROMTTYの信号により**TT04～TT11**出力不良)

C61—50—21E 23

(TTIFLG不良)

C61—58—02C 27

(**TT04～TT11**を入力とする**ACS04～ACS11**不良)

C61—50—11 10

(IOP124信号不良)

6．TTYからPUにDATAを読み込むが，読み込んだDATAをPUNCHできない．

C61—50—21E 23

(TTINT信号不良)

7．RIM FORMATで書かれたTAPEを読むが，正確に読んでいない．

C61—50—21E 23

(TTIFLG回路不良，ACCLRBUS信号，SKIPBUS信号誤動作)

メモリーアドレスを含まない命令

リンク回路不良

C61—50—06 17

C61—50—02A 12

C61—50—03A 13

C61—50—08 22

C61—58—02C 27

(ACCLRBUS誤動作，IOSKS誤動作)

C61—50—09 11

8．RIM FORMATで書かれたTAPEは正常に読むがBINARY FORMATで書かれたTAPEを読まない．

C61—50—05A 15

(WXIOT不良)

9．BINARY FORMATで書かれたTAPEを最後まで読んでSUMCHECKがとれない．

C61—50—21E 23

(**TTOCLK** 4.6ms不良，**ACCLRBUS**不良)

C61—58—02C 27

(**ACCLRBUS**不良)

C61—50—06 17

ACCLR信号不良)

(注) ACXX，AC11P出力信号が他のICの入力レベル不良によってSUMCHECKがとれない時がある．

以上，TTYに関して，IOP1，IOP2，IOP4信号がC61—50—11（LOC10）において正常と仮定した．

4—4 メンテナンスプログラムによる不良PKGのチェック

メンテナンスプログラムの扱い方については別冊「基本保守診断プログラム」を参照されたい．

1．MEMORY CHECKER BOARD(HIGH)(LOW) を読込まない（実行しない）

主として　メモリー部
(主にICの不安定)
本体　C61—50—01　19

2．RANDOM ISZ TESTを読込まない（実行しない）
(ISZ命令をTESTする)
C61—50—02A　12
C61—50—03A　13
C61—50—05A　15
C61—50—09　11
◎LPC不良

3．RANDOM JMP TEST、および、RANDOM JMP—JMS TESTを読込まない（実行しない).
(JMP命令、JMS命令をTESTする)
◎C61—50—11　10
C61—50—02A　12
C61—50—03A　13
C61—50—05A　15

4．INSTRUCTION TEST1を読込まない（実行しない).
(AND、TAD、OPERATE命令をTESTする.)
C61—50—02A　12
C61—50—03A　13
C61—50—05A　15
C61—50—06　17
C61—50—09　11
C61—50—08　22

5．INSTRUCTION TEST 2 を読込まない（実行しない).
(AND、TAD、ISZ、JMP、JMS命令、インターラプトをTESTする)
C61—50—02A　12
C61—50—03A　13
C61—50—05A　15
C61—50—09　11
C61—50—11　10

5. 付表および付図

5－1　CEC555S信号リスト

プロセッサー

信号名	ロケーション	説　　明
A	3,5,6,8,12 13,14,15,16,17	1ワードタイム中の14発のクロック
AC00～11	15,16,18	ACレジスタの各ビット出力
ACCLR	11,15	ACをクリアする信号
ACINS	11, 15	直列なAC入力
AC00～11BUS	15, 19, 21	周辺装置，TTY，SRよりのAC入力バス
ACSHG	11, 15	ACのシフトゲート
AND	6,7	命令
AST	1	自動スタート
BINITIALIZE	10, 17	BUFFERED INITIALIZE
BIOP1,2,4	17, 21	BUFFERED IOP 1,2,4
BRUN	10	BUFFERED RUN
BT00	3,5,7,16	ビットタイム00
BT01	3,4,6	ビットタイム01
BT(00～01)	3,11	ビットタイム00～01
BT(00～06)	3,10	ビットタイム00～06
BT(07～11)	3,10	ビットタイム07～11
BT(00～11)	3,5,6,11,12,13,	ビットタイム00～11
BT12	3,4,5,9,11,16	ビットタイム12
BT13	3,8	ビットタイム13
BT1200	3,5,9,11,16	DELAYED BT12
BT12DY	1,2,5	DELAYED BT1200
CFFC	2,3	基本クロックを½に分周したパルス
CLK	1,2,3,5	基本クロック
CLL	7,11	命令
CML	7,11	命令
CONS	1,21	**CONT**
DC	2,10, 11	**STORE**のときのワードタイム
DCA	6,7,9	命令
DEF	2,9,10, 16	DC＋EX
DELF	1,2,11	DC＋EX＋LD
DELP	1,2,11	DP＋EP＋LP
DELSP	1,8	DP＋EP＋LP＋SP
DEPS	1,21	**STORE**
DP	2	**STORE**で最初に出るパルス
DPA	1,2	**STORE**でDPの後に出るパルス
EX	2,9	**READ**のときのワードタイム
EXS	1,21	**READ**

EP	2	**READ**で最初に出るパルス
EPA	2	**READ**でEP の後に出るパルス
FCONS	1,2	**CONT**でセットされるフリップフロップ
FDEPS	1,2	**STORE**でセットされるフリップフロップ
FEXS	1,2	**READ**でセットされるフリップフロップ
FLAS	1,2	**LOADPC**でセットされるフリップフロップ
FROMTTY	19	TTYからの直列入力
IATDNMFPZ	7,11	AND＋TAD＋PCA＋IAC＋CMA＋ROTL＋OPZ
IATJP1	7,8	AND＋TAD＋JMP＋OP1
IATZ	7,9	AND＋TAD＋ISZ
IAUG	7,8,10	IOT＋OP(OPコード0～5)
ICML	7,11	CLL＋CML
INITIALIZE	17,18,19	PCSP＝PPC＋SP
INMA	7,11,16	IAC＋CMA
INOCHPC	7,9	ISKIP＋JMP(PC を変更しない)
INOIND	7,8	IR 3＋IOT＋OP(間接指定にならない)
INTRQST	4,5	インタラプションリクエスト
INTRQSTBUS	5	インタラプションリクエストバス
INTRQSTR	/	INTRQSTの表示用出力
ION	4,21	インタラプションリクエストエネーブル
IOP1,2,4	4,21	IOPパルス1,2,4
IOP1,2,4	4,21	IOP1＋IOP 2＋IOP 4
IOSKS	4,5	SKPフリップフロップのSP入力
IOT	4,6,7,9	I/O 命令(OPコード6)
IR0～IR 3	6,7,8,9	IRレジスタ
IRCLR	4,6	IRレジスタのクリアパルス
IR 0,1,2R	6,21	IR 0,1,2,表示用出力
ISZ	6,7,9	命令
ITNMA	7,11	TAD＋ROTR＋ROTL
IZDS	7,9	ISZ＋DCA＋JMS
JMP	6,7,9	命令
JMS	6,7,9	命令
L	5,11,16,21	リンク
LAS	1,21	**LOADPC**
LCLK	11,16	Lのシフトパルス
LD	2,9,16	**LOADPC**のワードタイム
LINS	11,16	Lの直列入力
LPA	1,2	LPの後に出るパルス
LP	1,2	**LOADPC**で最初に出るパルス
LPC	4	低電圧状態で出るレベル
LR	21	リンクの表示用出力
MA00～11	4,13,21	MAレジスタの各ビット出力
MAINS	10,13	MAレジスタの直列入力

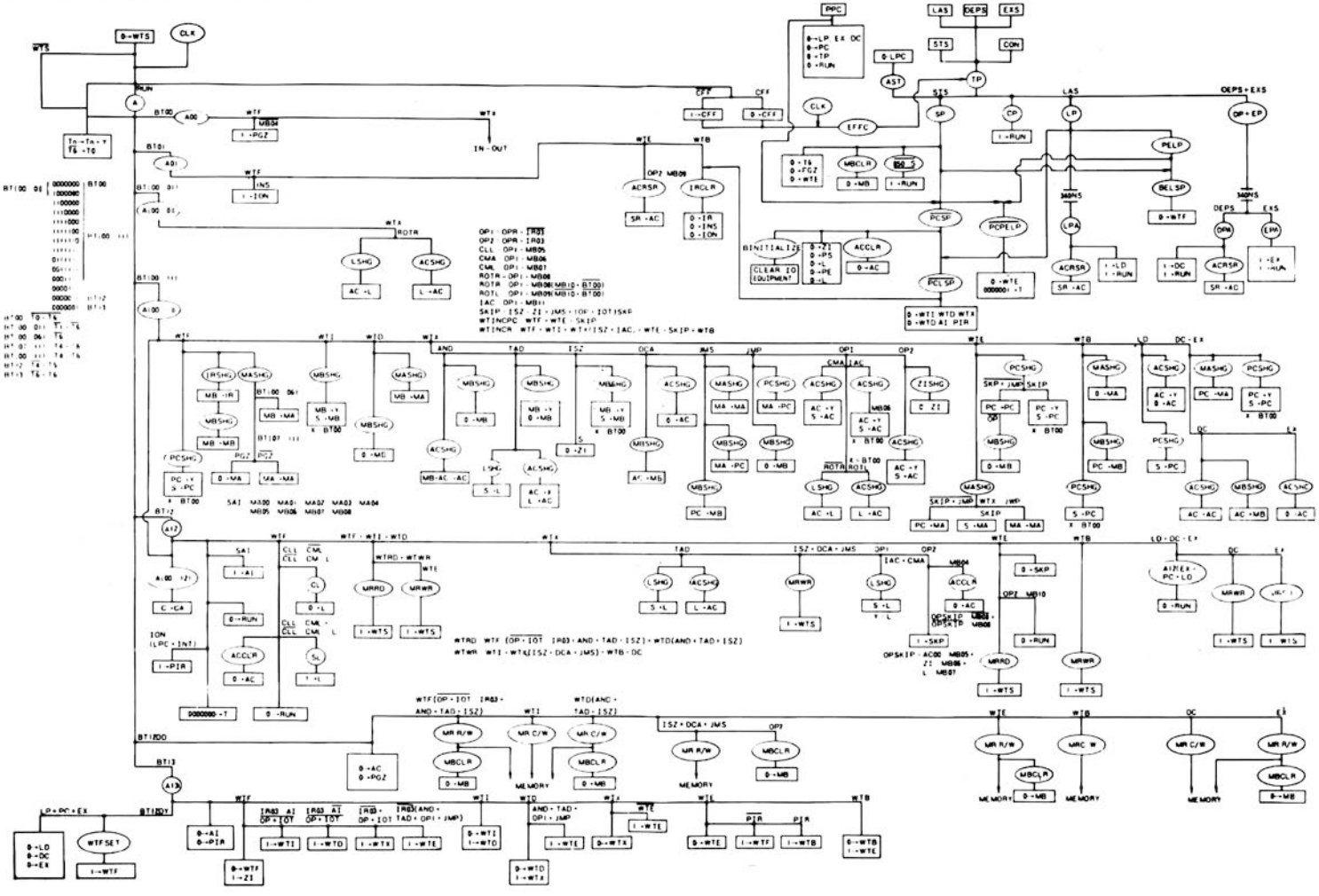
MA11P	10,13	MA11と同じ
MASHG	9,13	MAレジスタのシフトゲート
MB00～11	1,3,4,5,7,9, 11,16,17,21	MBレジスタの各ビット出力
MBCLK	12	MBレジスタのシフトパルス
MBCLR	10,12	MBレジスタのクリアパルス
MBINS	10,12	MBレジスタの直列入力
MB11P	6,9,10,11,12,16	MB11と同じ
MBSHG	10,12	MBレジスタのシフトゲート
MGO	3	メモリーからのBUSY信号
MRC／W	9	メモリーへのクリア／ライト信号
MRRD	9	MRR／Wの直前に出るパルス
MRR／W	9,10	メモリーへのリード／リライト信号
MRWR	9	MRC／Wの直前に出るパルス
OP	6,7,10	OPコード7の命令
OP1,2	1,2,7,9,11,16	OP命令の1および2
PC00～11	14,21	PCレジスタの各ビット出力
PCDELP	1,3,8	PPC+DP+EP+LP
PCINS	10,14	PCレジスタの直列入力
PCLSP	1,4,6,8,	PPC+LP+SP
PCSHG	9,14	PCレジスタのシフトゲート
PC11P	10,14,16	PC11と同じ
PCSP	1,3,5,11,16	PPC+SP
PIR	4,8	プログラムインタラプト
PMA00～11	21	PC,MA,MB,ACの選択された表示出力
PPC	1,2,14	電源が断のときに出るパルス
READERUNCLR	17,19	TTYリーダのクリア
ROTR	7,11	命令
RUN	2,3	計算機が動作している状態
RUNR	2	RUNの表示用出力
S	5,10,11,16	加算器のSUM出力
SKP	5,7	SKIP条件
SKIPBUS	4,17	周辺装置からのスキップバス入力
SNS00～11	12	コアメモリーのセンスアンプ出力
SP	1,2,3,5,8,11	**START**で出るパルス
SPMA00,01	20	PMA00～11の選択信号レベル
SR00～11	21	**DATA** 0～11
STOSIK	2,21	**STOP**+**STEP**
STS	1,21	**START**
STT	1,2	**START**+AST
T0～T6	3,4	ビットタイムをつくるフリップフロップ
TAD	6,7,9,11	命令
TO TTY	18	TTYへの直列出力

TRP	17	TTYのリーダクラッチ信号
TTOCLK	17,18	TTYの出力用クロック
TTICLK	17,19	TTYの入力用クロック
TTINT	5,17	TTYよりのインタラプション信号
TTOPRESET	17,18	TTY出力用レジスタへのAC出力ゲート
TTIREADIN	17,19	TTY入力用レジスタの出力ゲート
TTOSHIFT	17,18	TTY出力用レジスタのシフトパルス
TTISHIFT	17,19	TTY入力用レジスタのシフトパルス
WBDFI	9,10	WTI+WTD+WTB+WTF
WBFDELF	9,10	WTF+WTB+LD+DEF
WESKIP	9,10	WTE·ISKP
WFNAUG	8,9	WTF·IAUG
WI×TZ	9,16	WTI+WTX(TAD+ISZ)
WTI	8,9,10	ワードタイムインデックス
WTB	6,8,9,10	ワードタイムインタラプション
WTD	8,9,10	ワードタイムデファー(ニインダイレクト)
WTE	1,2,5,8,9,10	ワードタイムエンド
WTF	2,4,5,6,8,9,10,11	ワードタイムフェッチ
WTFSET	1,8	WTFのSD入力=PPC+DELF
WTINCPC	9,16	WTF+WTE·ISKP=PCをインクリメントすべきワードタイム
WTINCR	9,16	PC,AC,MBなどをインクリメントすべきワードタイム
WTNOCHPC	9,10	WTE·INOCHPC
WTS	2,3	ワードタイムストップ
WTX	8,9,10,11,16	ワードタイムエクゼキュート
WTWR	9	メモリーへ書き込みを行うワードタイム
WTRD	9	メモリーより読み出しを行うワードタイム
WXDCA	9,10	WTX·DCA
WXIOT	3,4,9	WTX·IOT
WXISZ	5,9,10	WTX·ISZ
WXJMPS	9,10	WTX·JMPS
WXJMS	9,10	WTX·JMS
WX·OPZ	5,9,11,16	WTX·OPZ
WX·TAD	9,11,16	WTX·TAD
X	16	加算器のX入力
Y	16	加算器のY入力
ZI	5,7	ゼロ検知用フリップフロップ
+3V	5	
−5V	17,19	

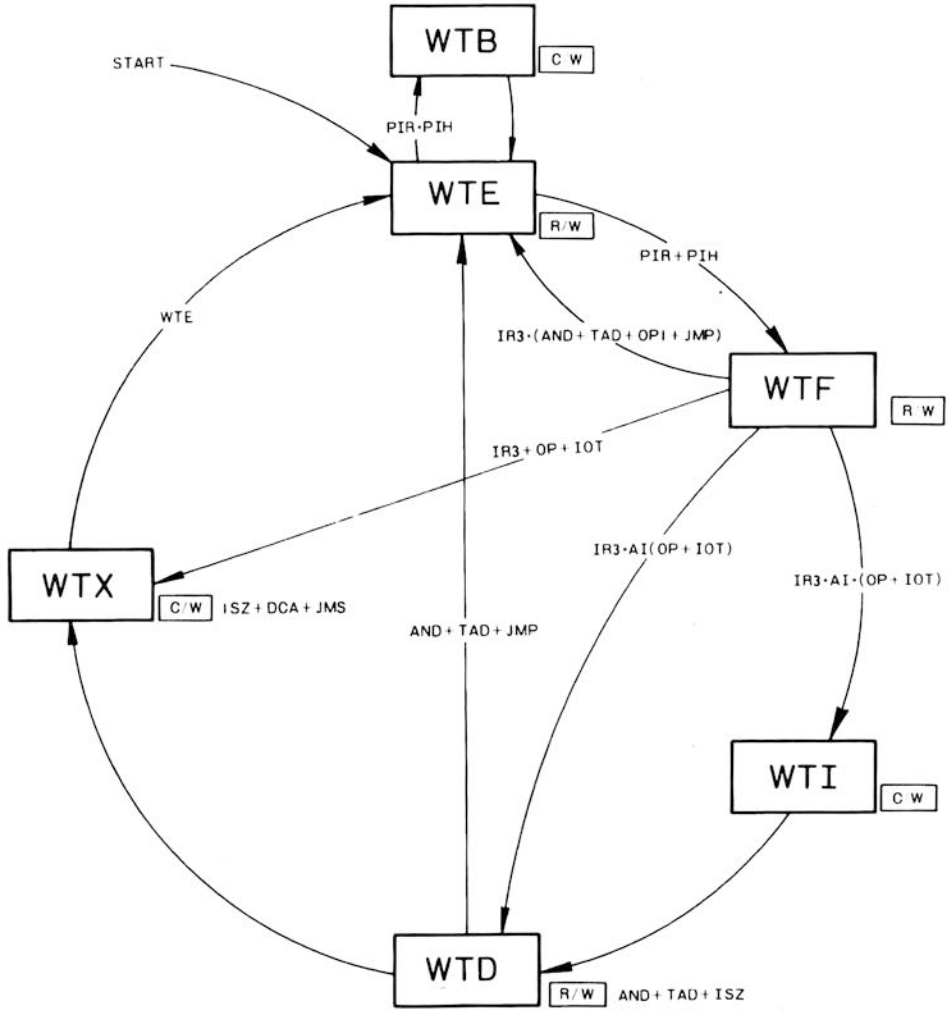
メモリー

AT	3,5,6,7,8,	アドレス選択のタイミングパルス
BSY	3	メモリーのBUSY信号＝MGO
CD	3	メモリーパワークリア＝MPC
C/W	3	CPUよりのクリアノライト信号＝MRC/W
IHBP	3,9	インヒビット電流を流すためのタイミングパルス
IHB 1 ～12	9,10	インヒビット電流1～12
MA00～11	5,7,6,8	CPUよりのMA出力(アドレス指定)
MB00～11	1,9	CPUよりのMB出力(書き込みデータ)
MSNS00～11	1,9	コアメモリーのセンスアンプ出力
R	2,5,7,6,8,11	リード電流
RP	2,3,11	リード電流のタイミングパルス
R/W	3	CPUよりのリード/リライト信号＝MRR/W
SA 1 ～12	4	コアメモリーのセンス出力
SA 1′～12′	4	〃
STP	3,4	センスアンプ出力のストローブパルス
X 1 ～12	6,7	リードおよびライト電流の流れるX線の入口
X 1′～12′	6,7	〃 出口
Y 1 ～12	5,8	リードおよびライト電流の流れるY線の入口
Y 1′～12′	5,8	〃 出口
W	2,5,6,7,8,11	ライト電流
WP	2,3,11	ライト電流のためのタイミングパルス

5－2　メインフローチャート



5－3　ワードタイム状態遷移図

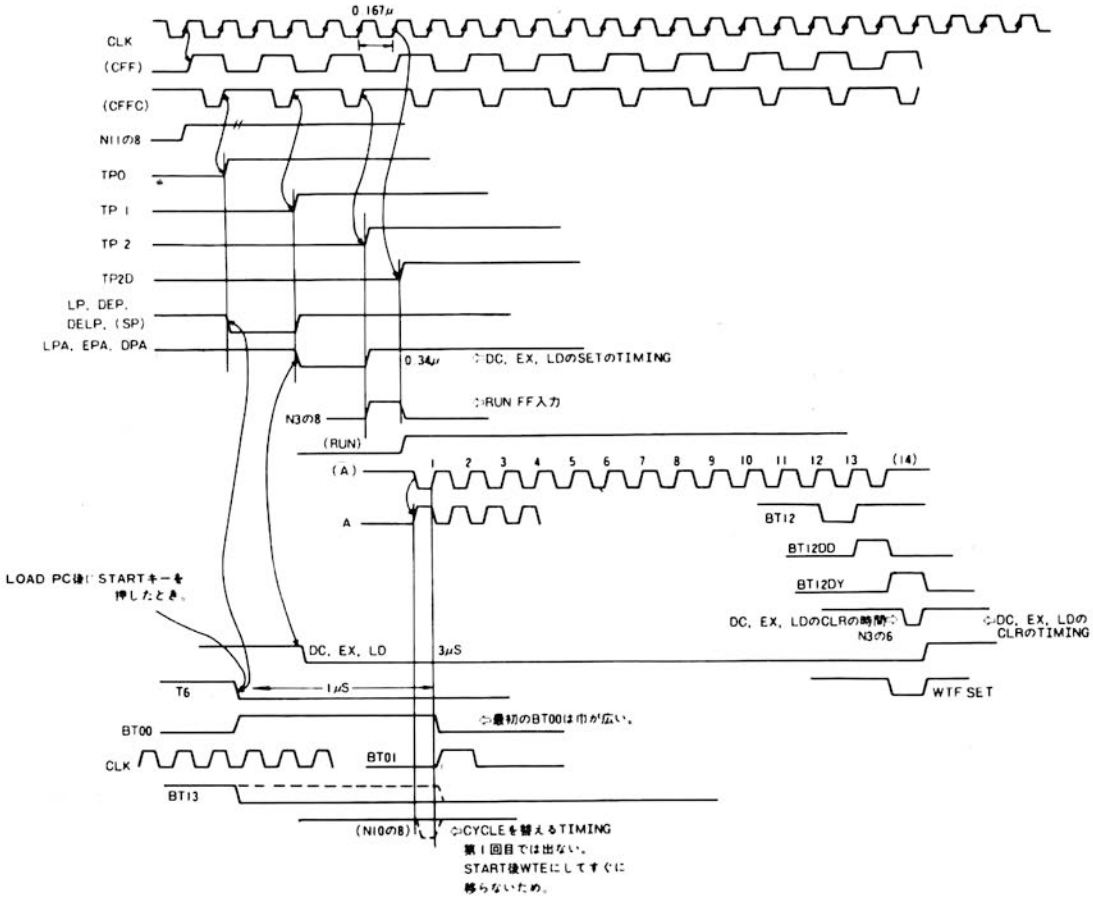


注）状態の遷移はAI3で行われる。

5－4　タイミングチャート

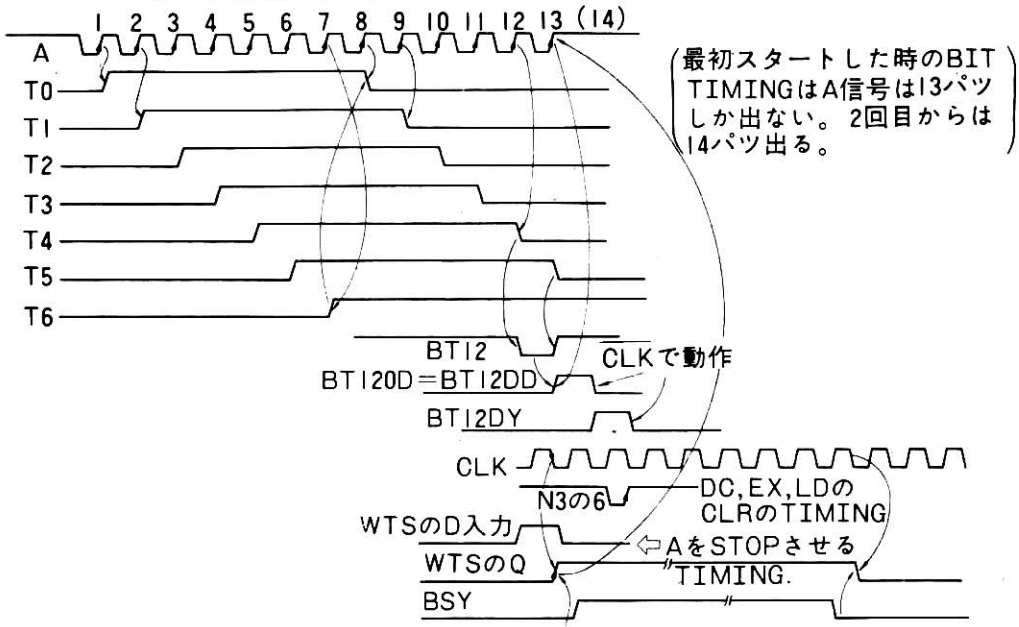
5－4－1　タイミングチャート1

コンソール操作時のタイミング（LOAD PC STORE READ）



5－4－2　タイミングチャート2

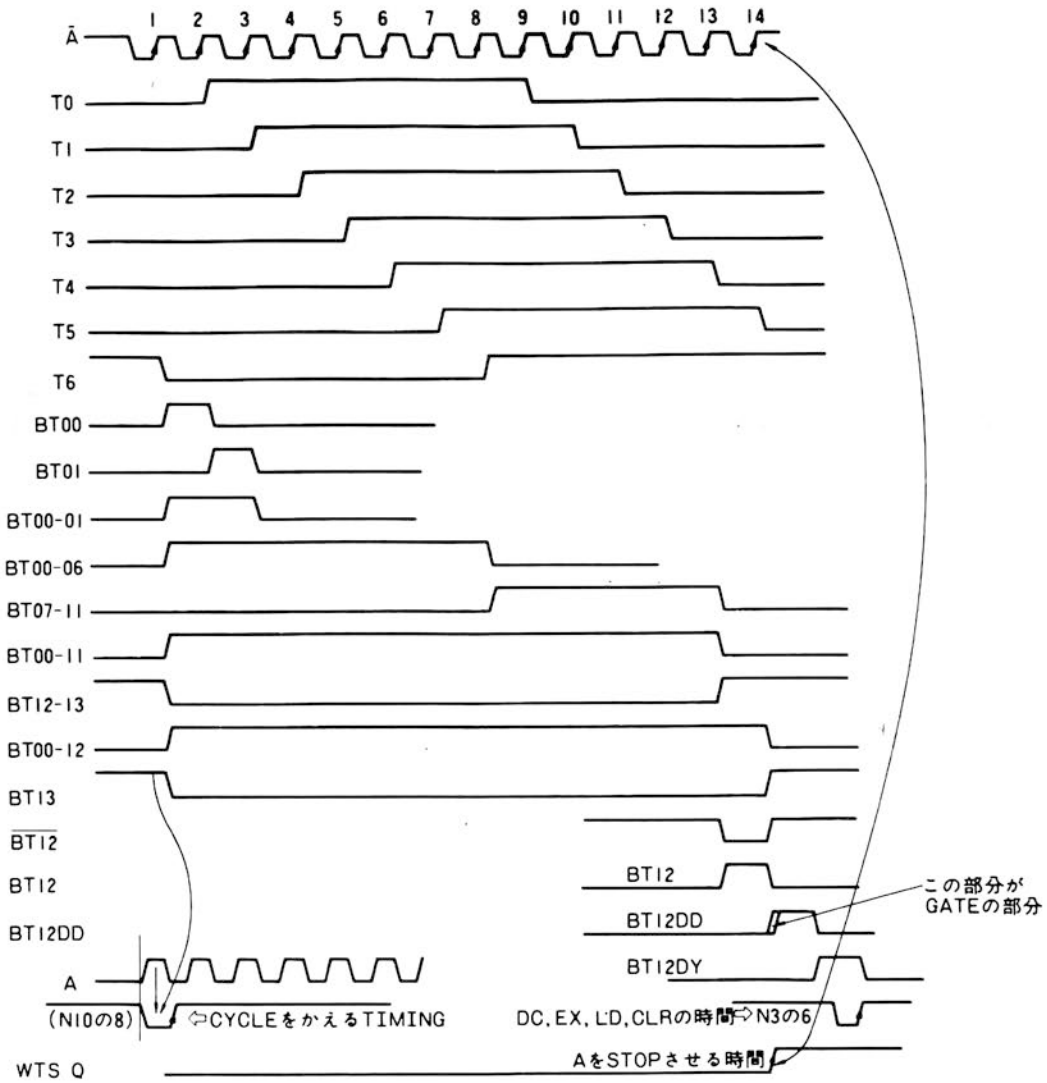
スタート時のタイミング



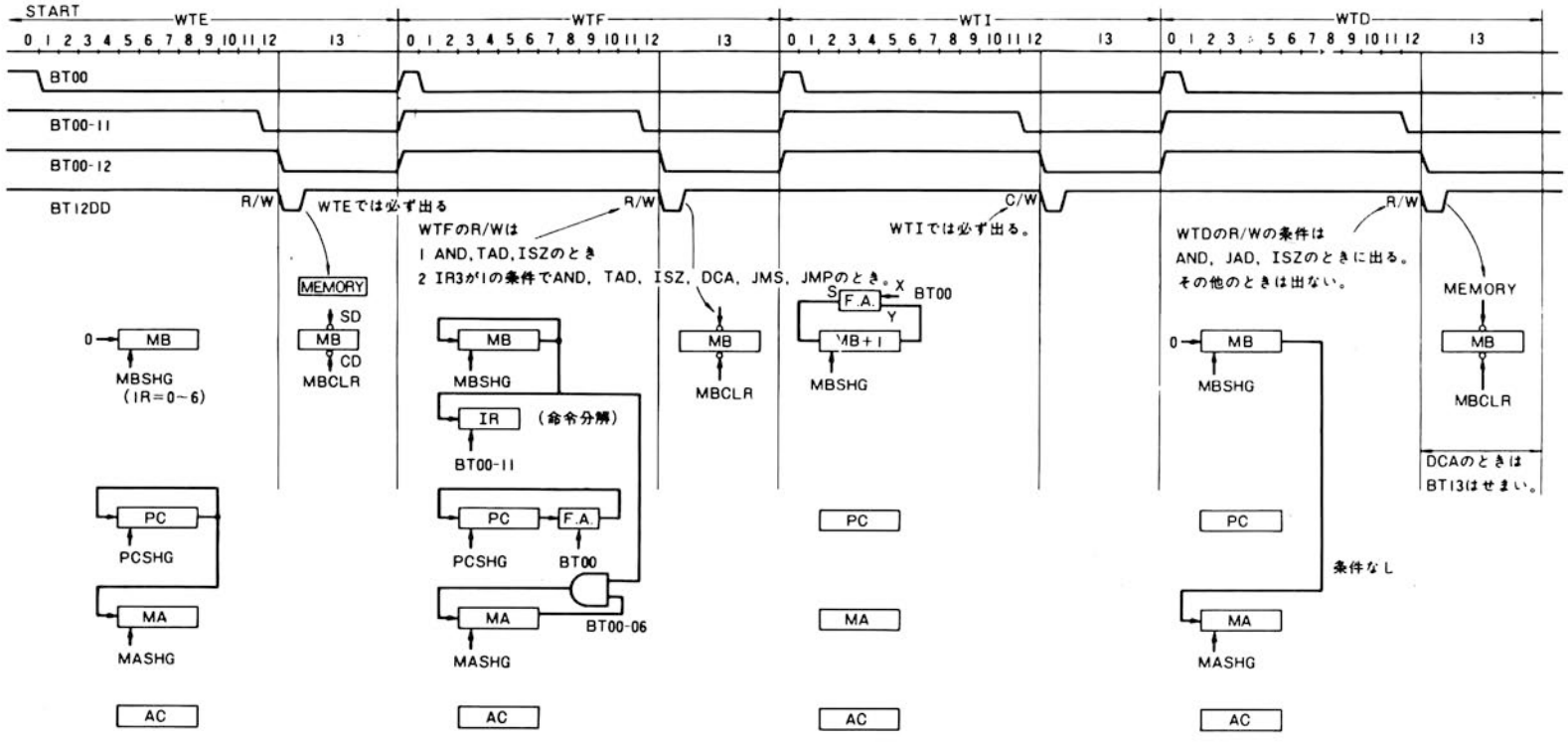
（最初スタートした時のBIT TIMINGはA信号は13パツしか出ない。2回目からは14パツ出る。）

このTIMINGはBIT TIMINGをSTOPさせるのとPC STORE READのときのRUNのSTOPのTIMINGでもある。

5－4－3　タイミングチャート３
ビットタイミング



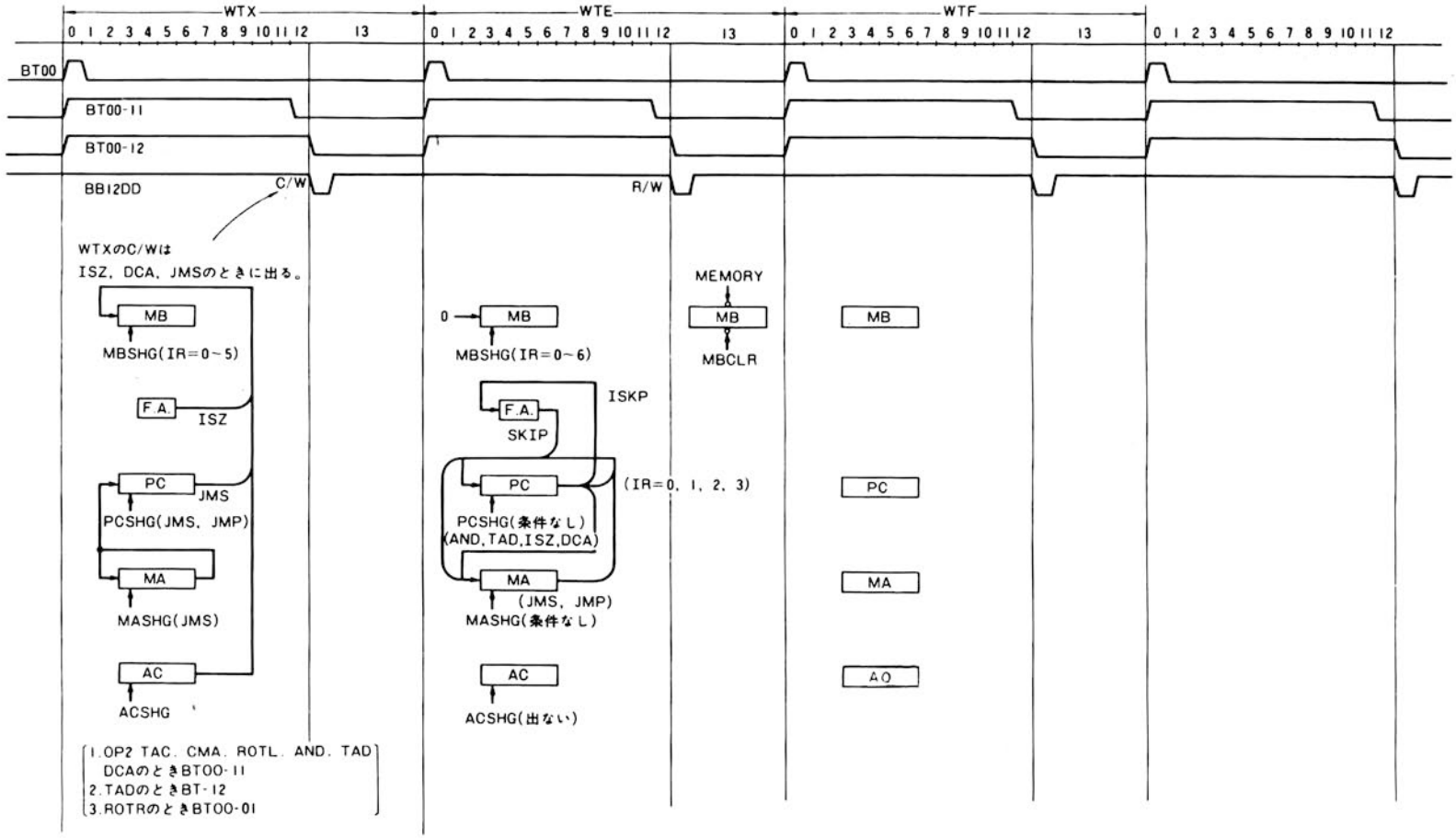
5－4－4　タイミングチャート　4(1)
命令3410を実行した場合のタイミング



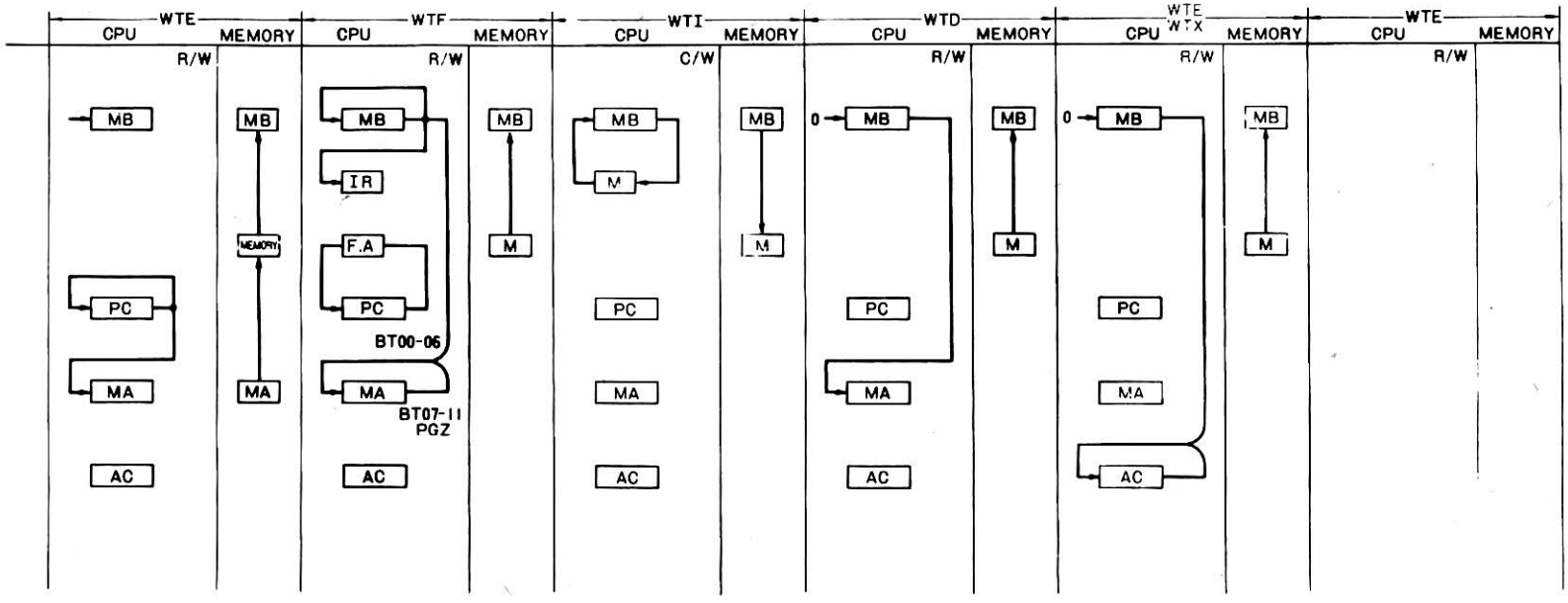
0番地　3410　　10番地　1000
1番地　7000　　1001番地　1111　(　　)

	WTE	WTF	WTI	WTD	WTX	WTE	WTF	
PC	0	1	1	1	1	1	2	
MA	0	10	10	1001	1001	1	1	
MB	3410	1000	1001	0	0	7000		
AC	0	0	0	0	0	0		

5－4－4　タイミングチャート4(2)
命令3410を実行した場合のタイミング



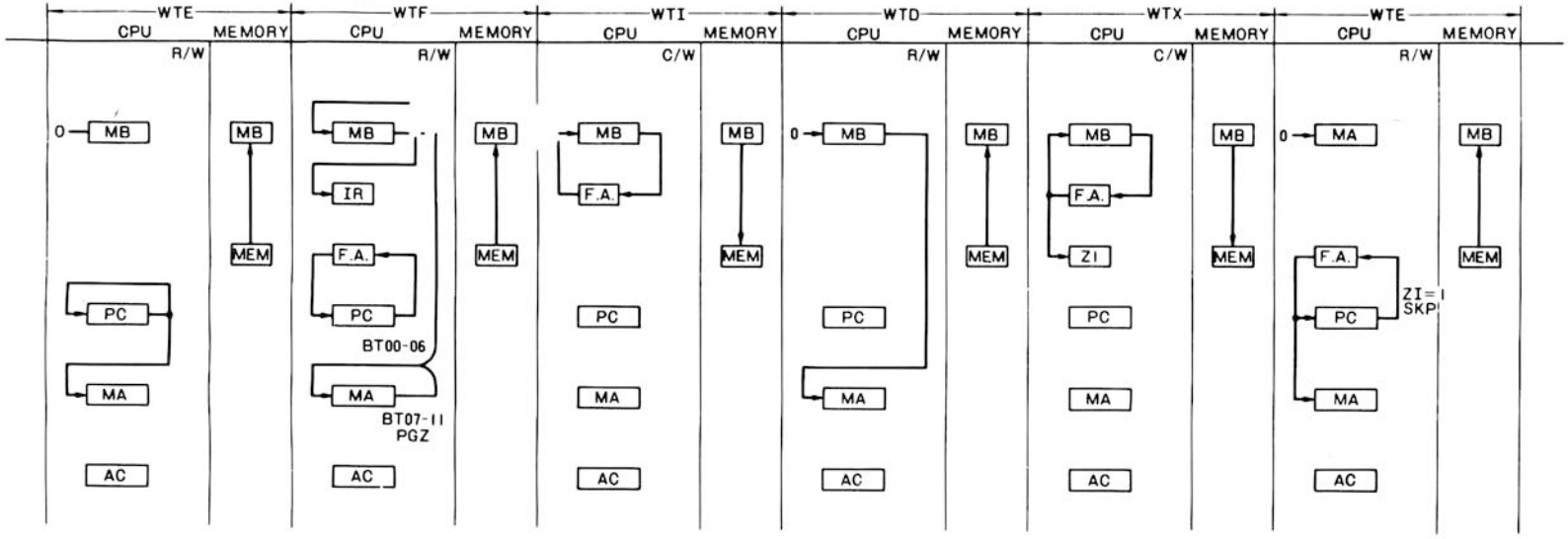
5－4－5　タイミングチャート5
ＡＮＤ命令における各レジスタの動き



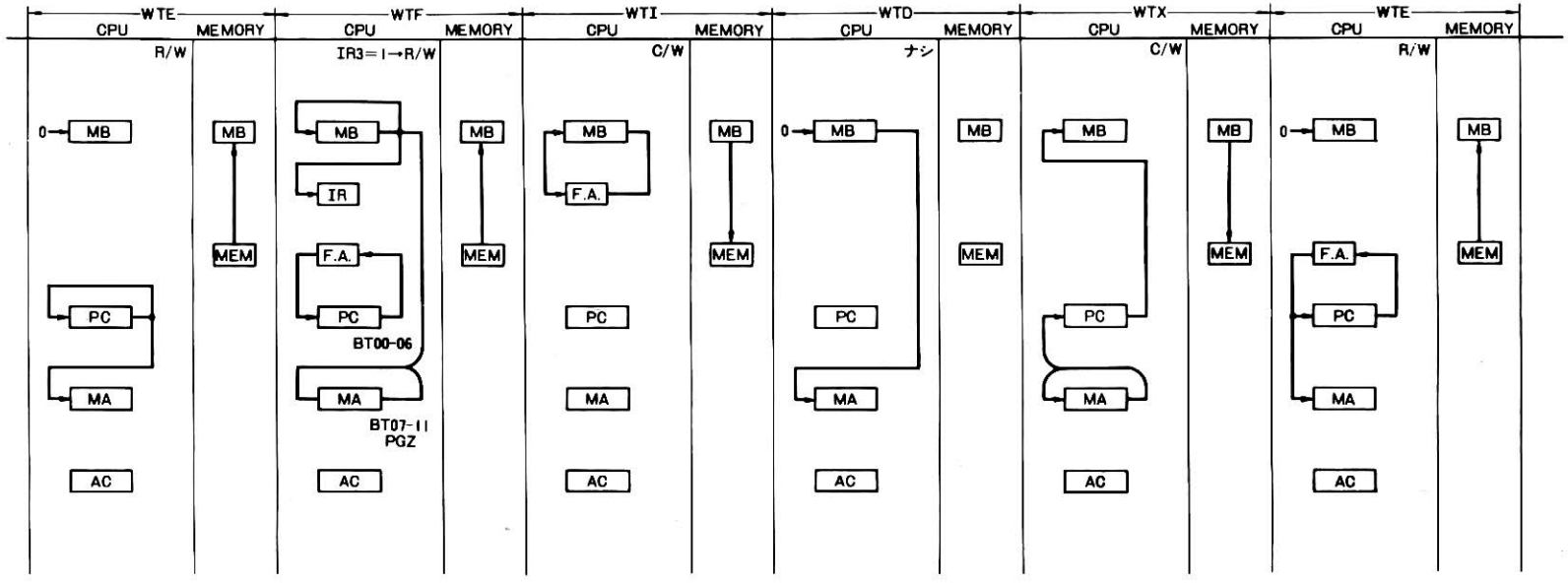
ANDの動作

1 WTE→WTF→WTE　2P+2M
2 WTE→WTF→WTD→WTE (WTX)　3P+3M
3 WTF→WTF→WTT→WTD→WTE (WTX)　4P+4M

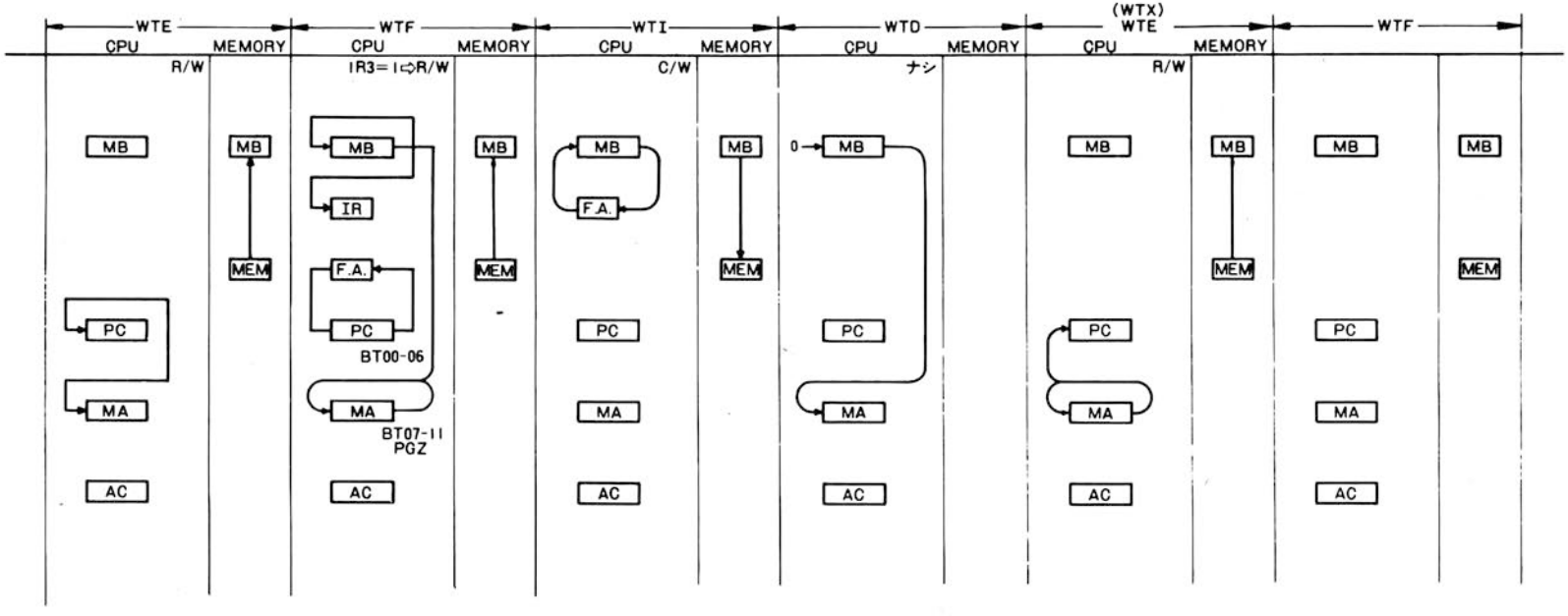
5－4－6　タイミングチャート6
ISZ命令における各レジスタの動き



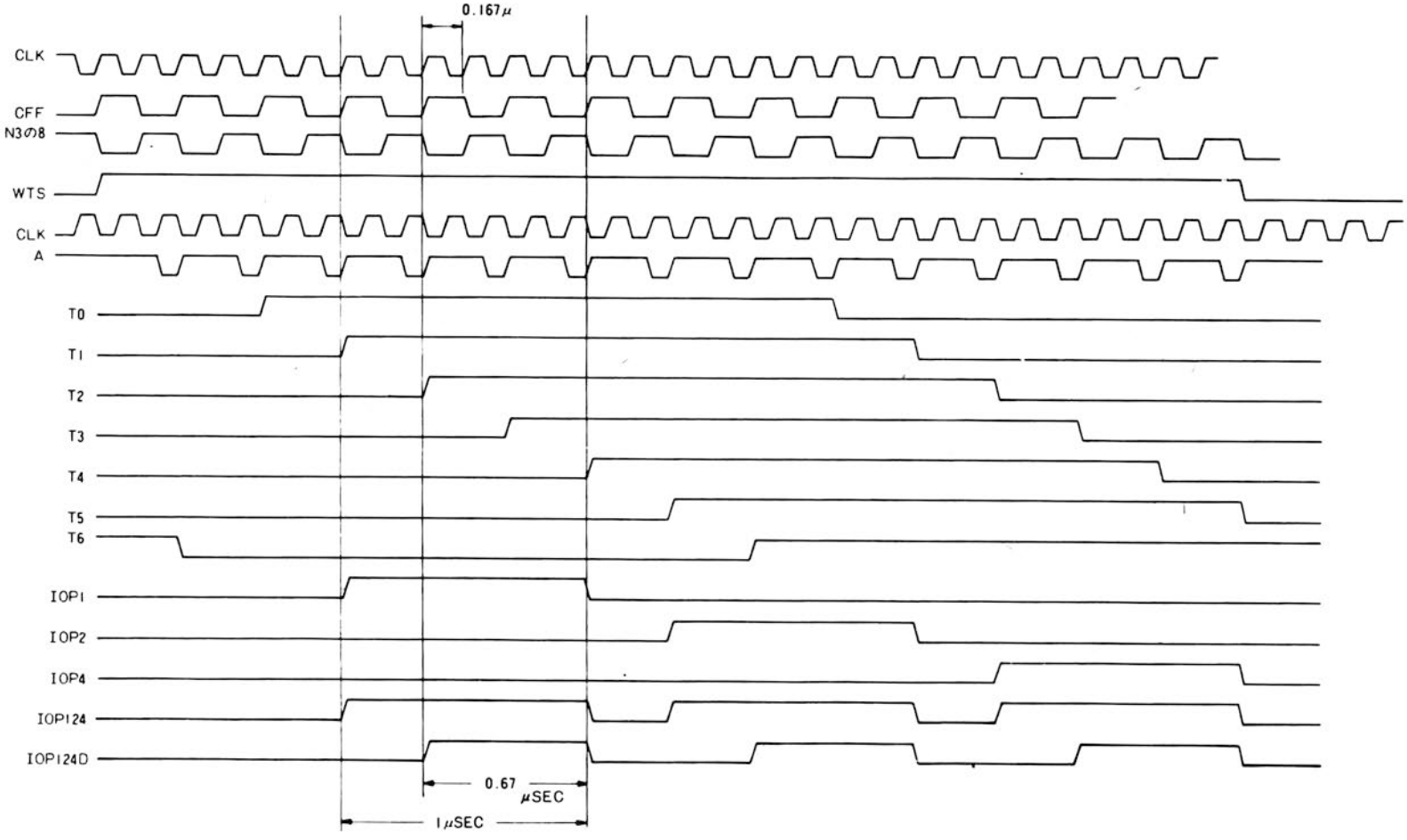
5－4－7　タイミングチャート7
JMS命令における各レジスタの動き



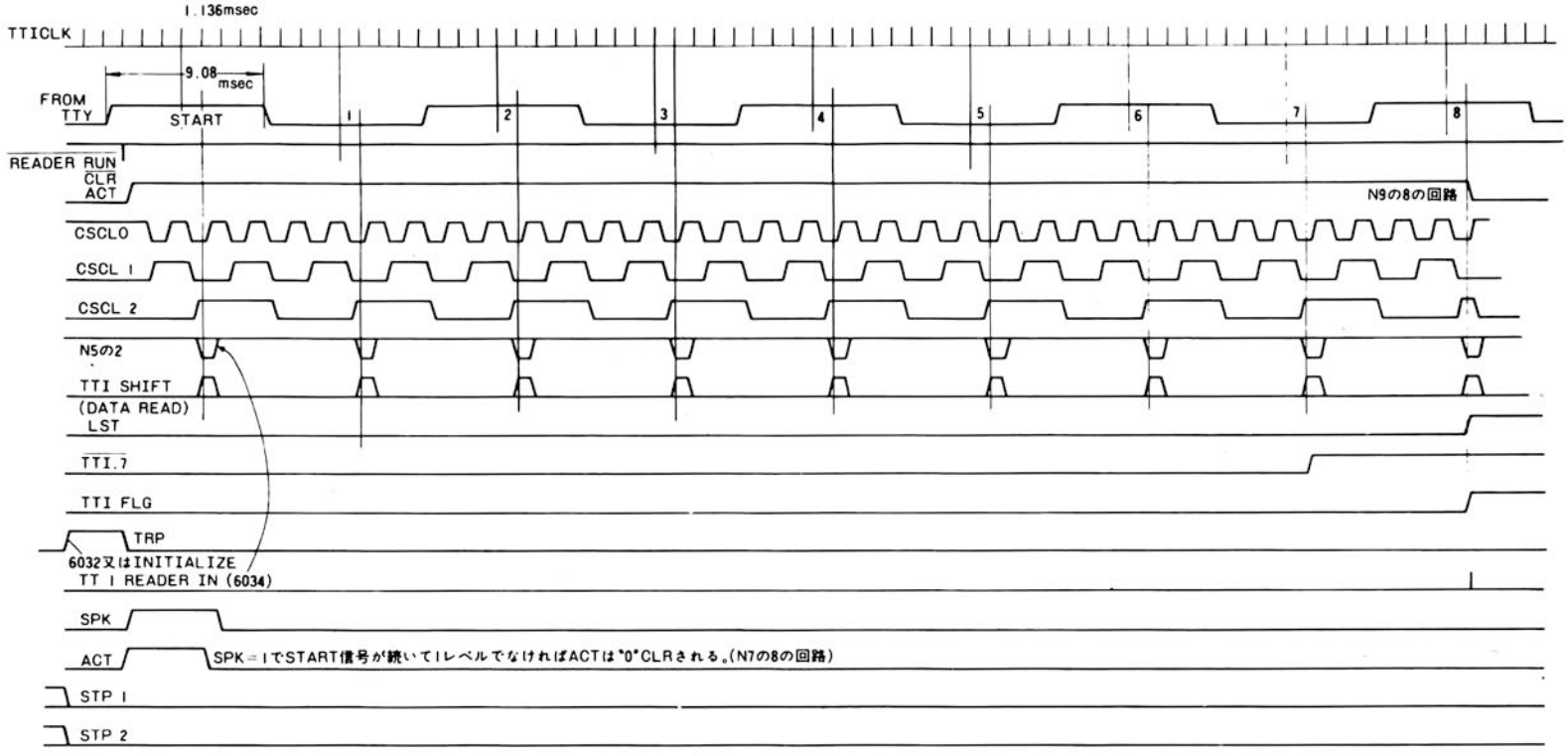
5－4－8　タイミングチャート8
JMP命令における各レジスタの動き



5－4－9　タイミングチャート9
入出力制御パルス(IOP)のタイミング

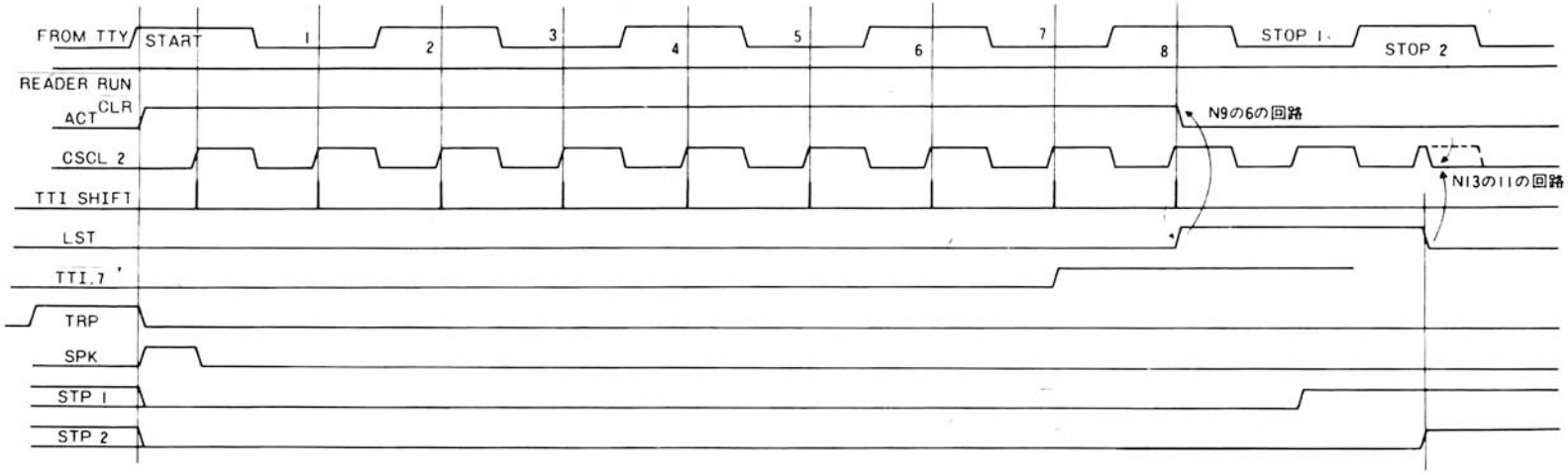


5－4－10　タイミングチャート10
テレタイプ(入力)のタイミング(1)

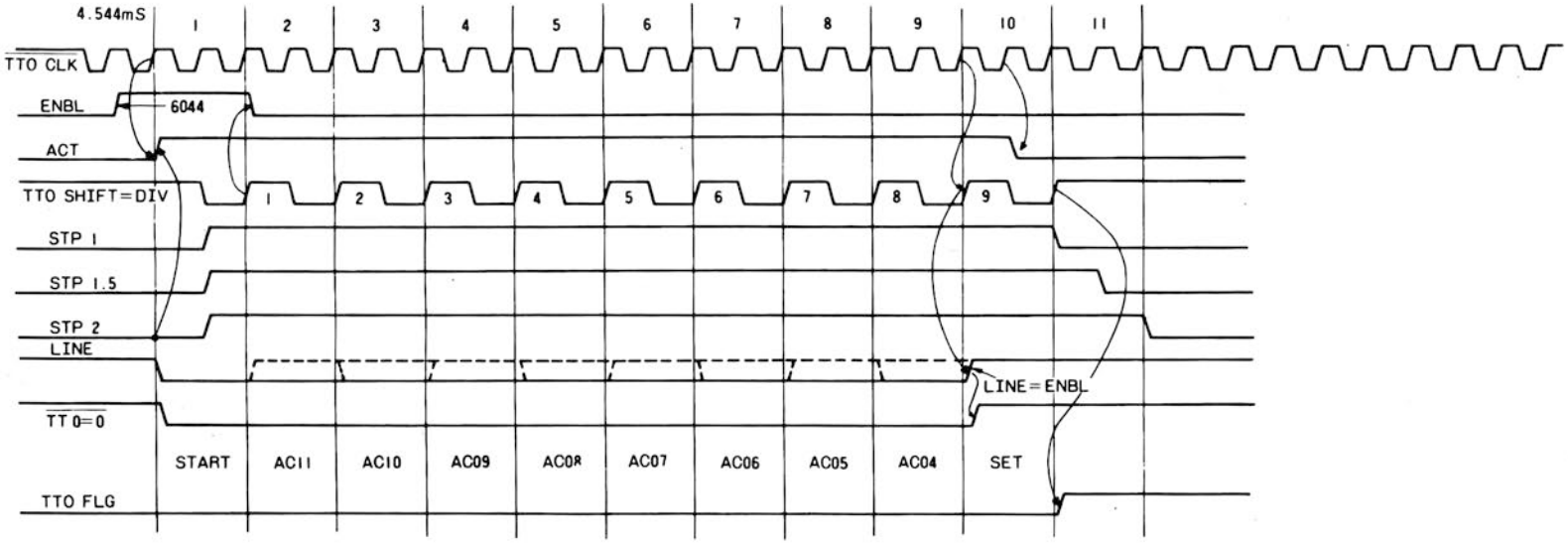


○TTYからの信号はSTART＝1　DATA＝8　STOP＝2のシリアル信号である。
○1個のパルスの中に8個のCLKパルスを含んでいるのはDATAの中間でREADパルスを作るためである。
○N5の2のパルスは巾が広くなっているが実際はμの巾である。

5－4－11　タイミングチャート11
テレタイプ(入力)のタイミング(2)



5－4－12　タイミングチャート12
テレタイプ(入力)のタイミング



周辺装置編

周辺装置編目次

① CKシリーズコードキーヤ …… 229
② CKシリーズマルチターミナル …… 238
④ DRISK-2002カートリッジ式磁気ディスク装置 …… 242
④ CFD-300ミニディスク装置…… 254
⑤ CFP-500フロッピィディスク装置 …… 256
⑥ CFD-130磁気ドラム記憶装置…… 261
⑦ CMT-8磁気テープ装置 …… 264
⑧ CMT-5磁気テープ装置 …… 267
⑨ CMT-100カセット磁気テープ…… 270
⑩ CS-1200入出力タイプライタ …… 274
⑪ CMD8, CMD4キャラクタディスプレイ …… 277
⑫ CS-408高速紙テープリーダ …… 281
⑬ CS-420高速紙テープパンチャ …… 283
⑭ CS-100入出力タイプライタ …… 285
⑮ CS-1000入出力タイプライタ …… 288
⑯ CLP-4100ラインプリンタ…… 291
⑰ CCD-5Sカラーキャラクタディスプレイ …… 294
⑱ CCD-5Gカラーグラフィックディスプレイ…… 296
⑲ CCD-5GH高速カラーグラフィックディスプレイ…… 298
⑳ C-1525デジタルX-Yプロッター …… 300
㉑ CS-60カードリーダ…… 302
㉒ CS-50, CS-80キーボード入力装置 …… 303

❶ CKシリーズコードキーヤ

(CK-3, CK-5, CK-6, CK-8)

脱パンチカード、脱文字キーボード、脱マークカードで情報入力の能率、コストを画期的に改善する入力端末装置!!

概　要

情報入力の手段としては、従来の常識的な手法、例えばキーボードや、OCR、OMRが一般的な方法として利用されているが、前者はアルファベット、数字、記号を主体とするキーボードを操作して入力する方法であり、けんしょう炎などの職業病の発生や、入力ミスが多発するなどの問題があり、加えて、わが国一般の言語習慣から誰れにでも操作技術上の格差なしに、簡単に、迅速に操作できるものではなく、この業務をキーパンチャなどの特殊な専門職に委託するのが通常とされており、またOCR、OMRについても、マークカード、マークシートといった中間媒体を必要とする関係上、コストや能率の点で効果的な手段とはいいがたい。

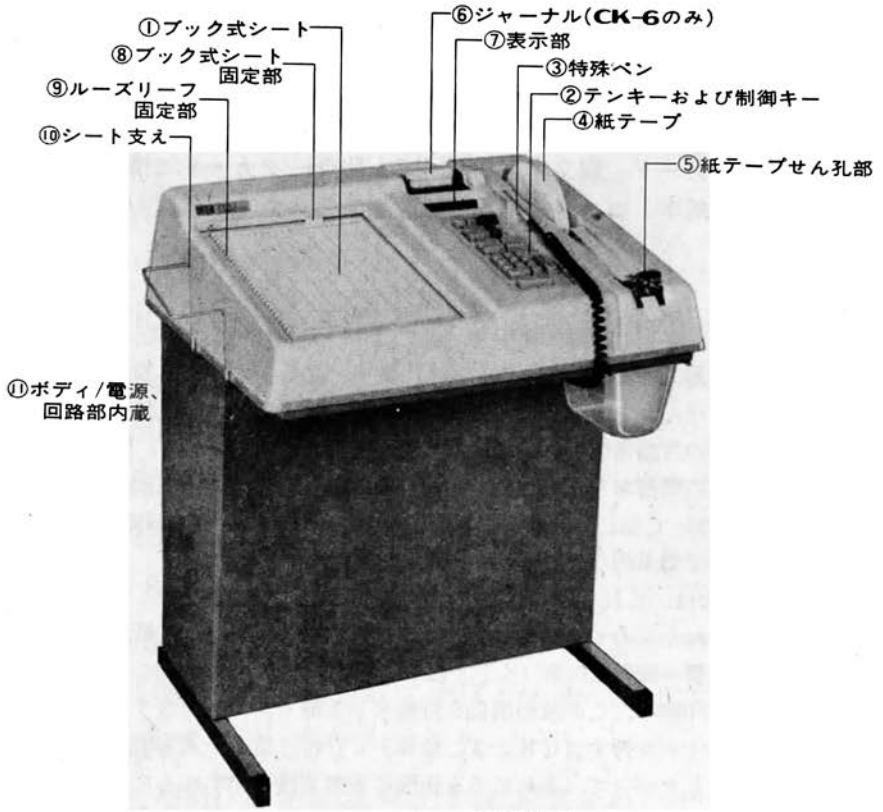
CKシリーズコードキーヤは、これらの問題点を、実際に使用する立場に立って、現場の作業の流れを十分検討し、開発されたもので、情報入力は文字ごとの取扱いでなく、紙製のブック式シート、あるいはアルミニウム製＝硬質板0.5t（もしくは厚紙製）のパネルシートに、項目名、商品名、部品名、取引先名を実名で印刷し、この該当項目を特殊ペンで軽くタッチする（CK－6、CK－8）か、あるいは対応するキーを押す（CK－3、CK－5）だけでデータ入力ができる画期的な万能入力端末装置である。したがって、誰れにでも正確に入力業務をすすめることができ、電子計算機システムの運用にあたって、システムを特殊な専門要員のものとしない非専従化を実現し、さらに情報入力の速度、能率、正確さ、汎用性を大幅に改善することができる。シートには1枚当り256項目（CK－6、CK－8＝8×32、CK－3、CK－5＝16×16)、シフトキーを利用して512項目が印刷できる。また、ブック式シートは約20枚を積み重ねて使用することができるので、1冊で約10000項目のデータを扱うことが可能である。またブック式シートやパネルシートを取替えることにより、データ量の増加や、他の業務との共同利用にも容易に対処することができるため、マスタファイルをもった電子計算機システムのあらゆる業種の情報入力装置として極めて有効であり、ユーザは長期にわたり、大幅に電子計算機の運用コストを引き下げ、電子計算機導入の大きな効果を得ることができる。

CKシリーズにはブック式シート利用のCK－6（オフライン用)、CK－8（オンライン用)、パネルシート利用のCK－3(オフライン用)、CK－5（オンライン用）の各種が用意されている。

1．CK－6、CK－8コードキーヤ

CK-6･8 コードキーヤは昭和49年度通産省「電子計算機等開発促進補助金」を受けて開発された。

1—1　コードキーヤ構成図



コードキーヤ構成図

1—2 特　長

- ●マスタファイルをもったあらゆる電子計算機システムの入力端末装置として「誰れでも」「どこでも」「何にでも」手軽に利用できる.
- ●所要目的によって正確なシートの原稿を作成しておくことにより，ブック式シートを交換して1台のマシンを各部門で共同利用することができる.
- ●けんしょう炎などの職業病の発生を防止できる.
- ●マスタファイルの追加，シートの品目変更の手続がきわめて簡単.
- ●標準1綴20枚，上下段シフトキーの利用により総計で10000項目を取扱うことができる（CK-6，CK-8).

1—3 仕　様

- ●**紙テープせん孔部**／入力されたデータの内容がトランザクションテープとして出力される.
  - せん孔速度：12.5字/秒以上
  - コード：8単位，ASCII，ISO
- ●**簡易プリンタ**／入力されたデータの内容が，ジャーナルとして記録される．データのチェックおよび控として用いる.
  - 行字数：20字/行
  - 印字速度：0.4秒/行
  - 文字記号：数字および記号若干
- ●**表示**／数字キーのデータのみを表示する（データの確認に用いる).
  - 桁数：12桁　表示文字：数字



CK－6（オフライン用）



CK－8（オンライン用）

●**特殊ペン**／コイル、スイッチを内蔵、シートの該当位置にペン先をタッチし座標コードを検出.

●**ブック式シート**／縦32、横8、1枚当り256項目、標準20枚を1綴として使用する. したがって総計10,240項目が入力可能.

用紙：合成紙B 4判135kg標準ルーズリーフ形式に綴込み使用.

●**主コントロール部**／すべての諸元のコントロールを含み入力から出力に至る制御部を有する.

主要素はTTL ICにより高い信頼度を有する.

●**コード変換部**／ASCII、ISOを標準とするが、既設の電子計算機コードに簡単に合致できるような分割したユニットが準備されているから指定によって、コードを変えることができる.

(ASCII、ISOほかはオプション)

●**電源部**

消費電力：約250VA

AC入力：AC100V、±10V、50～60Hz

1－4　ブック式シート

（項目名は手書き、タイプ印字、あるいは印刷して利用する）

●資材管理システムにおける利用例

SHEET No.	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	01		02		03			04			05		06			07			08	
01	アサヒ電機	エルコ・インターナショナル	協和電器								新光商事		ソニー・テクトロ			ターミック			東京電機通信	
02	秋元電機	エコー電子	共和産業								島田理化		ソフト・ウェア工業			多摩電子工業			東京コスモス	
03	アイ・エフ		極東貿易								伸和電気								東京電気化学	
04	アルプス・モトローラ		木住野商事								信州精器		田中貴金属工業			中央電材			東北金属	
05	アイ・エス・アイ		協和商会		阪根産業						昭和総業		高木商会			中央グローブ			東和蓄電器	
06	アサヒ・エレクトロニクス	オリエンタル・サービス	協立電業		三栄測器						昭和光機		大日電子			チェルベルジ			東海通信機材	
07	アンドール	大塚通商						山洋電気					大東通信機			千代田情報			東立通信	
08		OK電機						三協インター					大喜工業						東洋計器	
09		沖田 S/S	京浜アートワーク		山紫工業						スタンレー電気		高千穂交易			鶴賀電機			東光電子	
10	伊藤忠商事							三栄電子			住友スリー・エム		タカセ印刷						東和電線	
11	岩崎シンクロ		幸伸工業		三陽電機						住友商事		田代電機						東京特殊電線	
12	池上通信機		光電メディカル		佐藤抵抗								橘商会			寺田電機			東陽通商	
13	浦浜電子	カンン電子	国際変圧器		三信電機						精電舎		大都商事			デンケイ			トベ電機	
14		関東通信工業	小糸 S/S								千住金属		太栄商事			TDKフェアチャイルド			東亜金属	
15		長谷エレクトロニクス	小林 S/S										大和通信			ティアック			東芝ベックマン	
16		ウイダ医療	小島 S/S										太陽エレクトロ						東京プリント	
17			近田電機										大日制御機器						東京原子工業	
18																			東京エレクトロン	
19	日本パルスモータ	日本真空光学	ノア電気		日立工機								宮田 S/S			ユエオン電気			東京協立商事	
20	日本抵抗器	日本ディスカル	野田工業		日立製作所								三井物産電器販売						東洋印刷電信	
21	日本サーモ	日本ユニバック																	東洋プリント	
22	日本ホーロー抵抗	日産自動車販売														横河ヒューレット・パッカード			東和電子	
23	日本エーエム・ピー	日本オイル・シール						武川商事					森松電子			吉沢精機			鳥居電業	
24	日本通信光管		伯東		富士電業								森本電機						東芝アンペックス	
25	日興電子		バイオレット電機		双葉産業						北陸電気工業									
26	日本アイ・ピー・エム		花岡無線		フクダ・エム・イー						宝商					理研電具				
27	日本インター		パネトロン		扶桑商事											菱三電気				
28	日曜産業		浜松テレビ		富士通部品商事						マルコン電子		ヤマキ電気			理経			六甲電子	
29	エッコーム		ハイレル		フジコン						松村金属		山田工業			菱藤電機			和光電材	
30	ニットー		バンテック		福島電気						丸文		谷村新興			菱洋電機			和宏電子	
31	日本サイリスタ		八紘電業		富士弘商事						マミック		谷村新興サービス			リコー電子			渡辺測器	
32	日商エレクトロ				富士化学紙						科学放送		ヤセック・エレクトロニクス			ヤコベルグ				

CEC CK-6,CK-8 SHEET

●診療報酬請求書作成システムでの利用例

SHEET No.	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

	01	02	03	04	05	06	07	08
01								テープ終了
02	患者登録	項目変更	本人	家族	国保世帯主	国保その他		8
03	入院	外来	患者名	生年月日	保険者名	保険記号		7
04	内科	外科	産婦人科					6
05	男	女					直接指示	5
06	政府	給員	日雇一般	日雇特殊	組合	共済	日	4
07	自衛官	一般国保(東京都	一般国保(他府県)	国保組合	生活保護	結核予防一般	回	3
08	大気汚染	養育医療	自賠	労災			消去	No 1
09		200円	50円	0%	20%	30%	患者登録終了	No 2
10	ネブライザー	ネブライザー96		処置料				
11					内服	指導料	乳幼児加算	患者No
12					外用	内科再診(二度目)	時間外加算	
13					屯服	再診(二度目)	深夜加算	
14					注射料	再診(電話指示)	休日加算	
15					皮下筋肉注	往診料	診察料	
16					静主	暴風雨雪	初診	
17					リンゲル等	離島	再診	
18					点滴加算	投薬料	内科再診	
19								診療終了
20	0	1	2	3	4	5	6	7
21							8	9
22	ア	イ	ウ	エ	オ	A	B	C
23	カ	キ	ク	ケ	コ	D	E	F
24	サ	シ	ス	セ	ソ	G	H	I
25	タ	チ	ツ	テ	ト	J	K	L
26	ナ	ニ	ヌ	ネ	ノ	M	N	O
27	ハ	ヒ	フ	ヘ	ホ	P	Q	R
28	マ	ミ	ム	メ	モ	S	T	U
29	ヤ		ユ		ヨ	V	W	X
30	ラ	リ	ル	レ	ロ	Y	Z	
31	ワ	ン	゛	゜	¥	スペース	漢	ゟ
32	(	)	※	＋		−		／

CEC CK-6,CK-8 SHEET

1－5　CK－6，CK－8の操作

●使用するブック式シートを指定箇所に装着し，固定器具で押える．この時，座標位置がいちじるしくずれないように注意する．

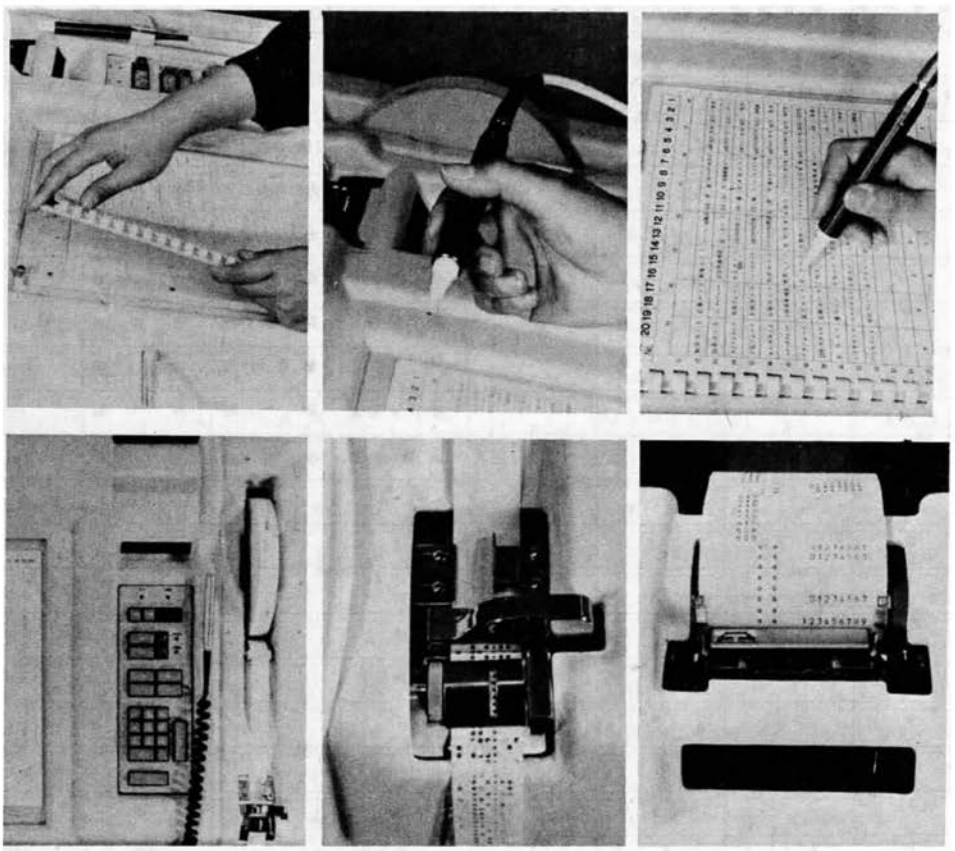
●特殊ペンをもつ，写真のように普通のペンをもつ要領でよい．

●原始伝票を見て，該当するシートを開き，該当項目をペン先で軽くタッチする．シートには正しい位置の表示が印刷されているので正確にタッチする．

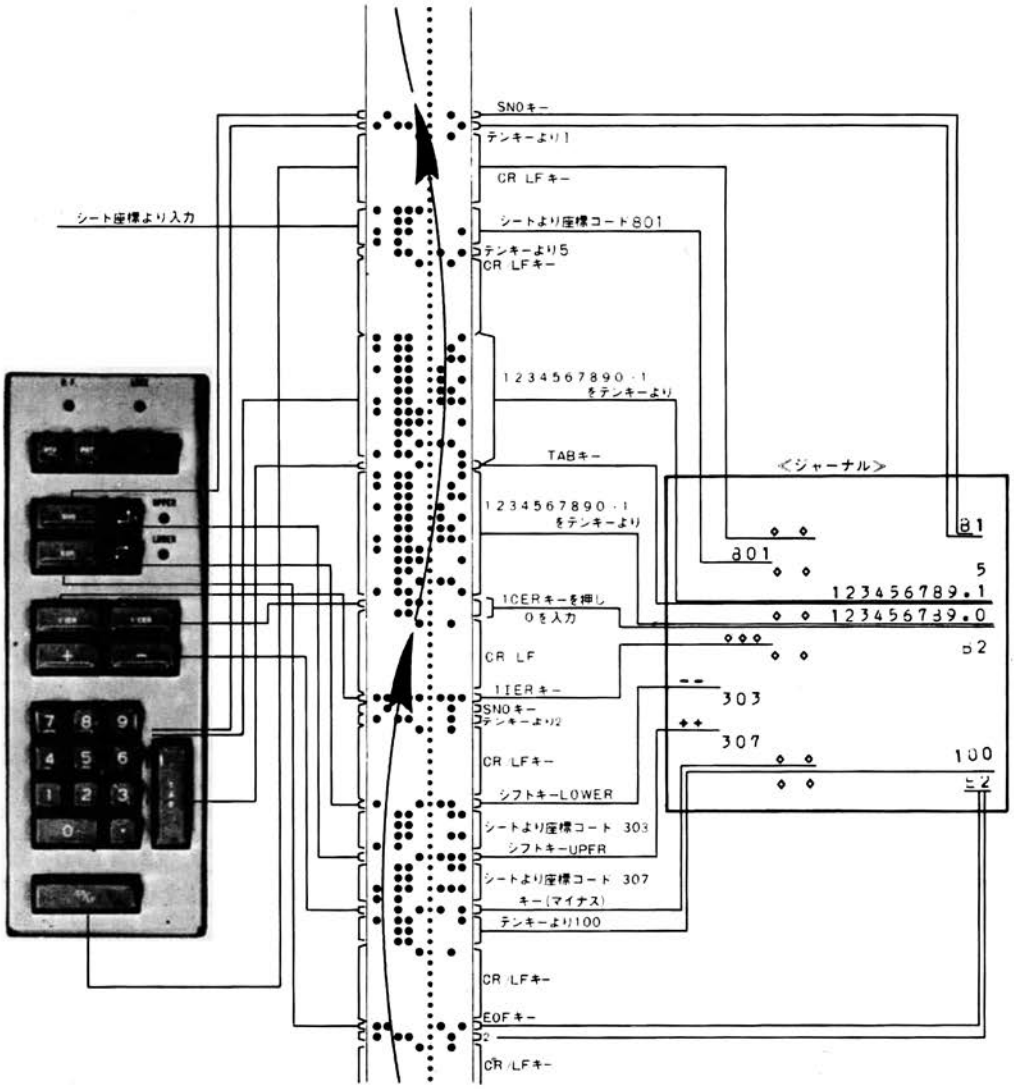
●数量を入力する場合は，該当数値をテンキーから入力し，定められた制御キーを押して1項目のデータ入力を完了する．

●オフライン使用の場合はデータは紙テープにせん孔される．オンラインの場合はそのまま電子計算機に入力．

●数値データは，入力時に表示部に表示されるので入力ミスがないか目視チェックする．またCK—6（オフライン）はジャーナルに印字される．

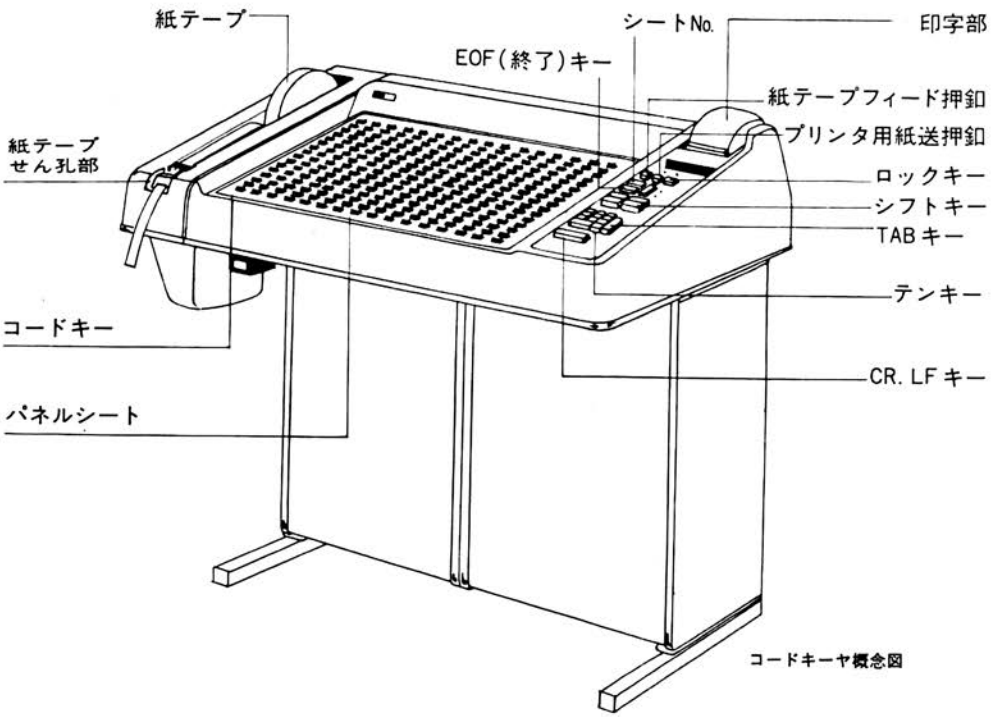


1—6　ブック式シート、テンキー、制御キーより入力されたコードと紙テープフォーマット、ジャーナルの対照



2．CK－3．CK－5コードキーヤ

2　1　CK－3．CK－5コードキーヤ構成図



コードキーヤ概念図

2－2　仕　様

左右16項目，天地16項目が実名で印刷されているパネルシートと対応したキーを押して入力するシリーズ．処理項目の少ないシステムの入力端末装置として手軽に利用できる．

●キーボード群／入力情報をキーインする．

群　別：パネルシート256品目×上下段シフト＝512品目×4096枚．

制御用キー6種または12種標　　数字キー　11種類(・，0～9)



CK－3コードキーヤ



CK－5コードキーヤ

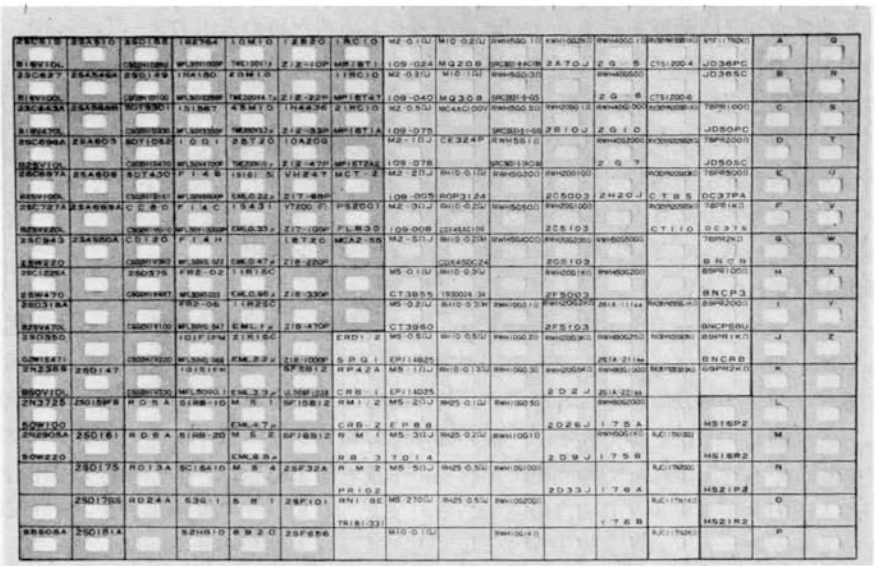
●パネルシート／パネルスイッチ群に必要とする品目のシートを装着し，使用する．

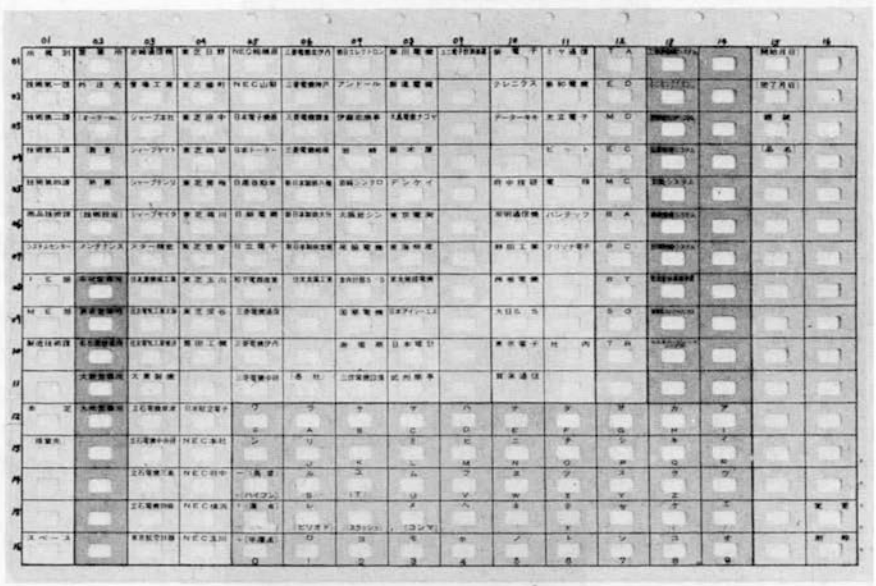
材　質：アルミニウム硬質板0.5t塗装クリーム下地，または厚紙シート印刷は各色可能．

2－3　適用例

財務会計処理の勘定科目のように規則に定められた総数が数百項目の場合に最適．

2－4　CK－3，CK－5用パネルシート





❷ CKシリーズマルチターミナル

(CK-10, CK-12, CK-15, CK-20)

概要

電子計算機のシステム機能を，より効率よく発揮せしめるための入出力ターミナルで，基本的に，ＣＳ－1200入出力タイプライタ，ＣＭＤシリーズ，キャラクタディスプレイ，ＣＫ－８コードキーヤの組合せで構成され，機能別に，

ＣＫ—10：ＣＫ—８，ＣＳ—1200，ＣＭＤ—８ターミナル専用

ＣＫ—12：ＣＫ—８，ＣＭＤ—８　印字機能なし

ＣＫ—15：ＣＫ—８，ＣＳ—1200表示機能なし

ＣＫ—20：ＣＫ—８，ＣＳ—1200，ＣＭＤ—８センタマシン専用

の４機能が用意されている．

ＣＫシリーズマルチターミナルの利用により，問合せ業務や，情報検索は，すべて実名で印刷されているＣＫシリーズのブック式シートの該当項目を，ディスプレイ上に表示された指示に従ってタッチするだけですべて表示することができ，特に必要とするもののみ印字する．全く無駄のないシステム方式を確立することができる．一般の操作はＣＭＤシリーズキャラクタディスプレイとの対話形式が基本であり，誰れにでも簡単に，容易に作業をすすめることができるもので，特にオフィス専用の入出力ターミナルとしては最適である．また，ＣＫシリーズマルチターミナルは，標準で８台までリンクすることができ，カプラ，モデムなどデータ通信機能を利用して，データ通信システムのターミナルとしても有効に利用できる．

1．マルチターミナル構成と機能

ＣＫ—10マルチターミナル

構成：ＣＫ—８コードキーヤ／ＣＭＤ—４キャラクタディスプレイ／ＣＳ—1200入出力タイプライタ

ターミナル専用のマシンとして使用する．入力はもちろんコードキーヤから入力，データは必要に応じてディスプレイ上にすべて表示される．ＣＫ—20形と同じく必要なもののみ印字部より印字する方式



CK－12マルチターミナル

構成：CK—8コードキーヤ／CMD—4キャラクタディスプレイ

入出力用のマルチターミナルとして使用する．構成はCKシリーズコードキーヤとCMDシリーズキャラクタディスプレイを組み合わせたので，ターミナルを多数分散して使用する場合効率がよい．



CK—15マルチターミナル

構成：CK—8コードキーヤ／CS—1200入出力タイプライタ

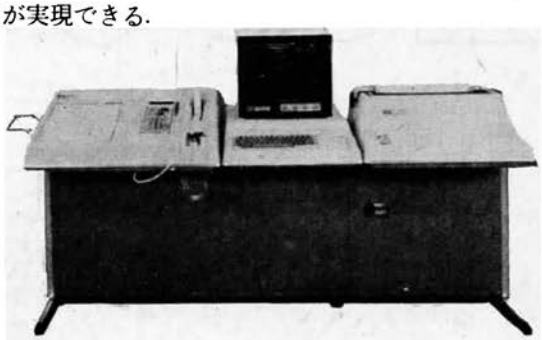
入出力用のターミナルとして使用する．出力印字の機能に重点をおいたターミナルとして利用する．



CK—20マルチターミナル

構成：CK—8コードキーヤ／CMD—4キャラクタディスプレイ／CS—1200入出力タイプライタ

センターマシン用のマルチターミナル，左からCKシリーズコードキーヤ，CMDシリーズキャラクタディスプレイ，CS—1200プリンタが配置されている．システムコンソールとしてCMDシリーズキャラクタディスプレイを共用して必要があるもののみ印字部で印字する．全くむだのないオペレーションが実現できる．



2　CKシリーズマルチターミナル各部仕様

データ入力部（CK－8コードキーヤ）

シ　　ー　　ト：1枚8（横）×32（縦）計256項目
　　1級標準20枚のブック式

入　力　方　式：特殊ペンにより入力項目の座標を検出

紙テープせん孔：入力データの内容をトランザクションテープとしてせん孔（オフライン使用）

数　字　キ　ー：数値を入力，および0～9の11種

コントロールキー：左右シフト，修正，その他

ブック式シート：用紙B4判標準135kg相当　1級20枚以下

表示部（CMD－4，CMD－8キャラクタディスプレイ）

表示文字数：800字（40字×20行－CMD－4）
　　2560字（80字×32行－CMD－8）

表示文字数：英・数・記号　64字　　カナ・カナ記号　64文字　　合計128字

印字部（CS－1200入出力タイプライタ）

印　字　速　度：30文字/秒

文字の種類：96種（ASCII標準，カナ文字オプション）

1行の印字数：132行/インチ

用　紙　幅：15インチ標準

キーボード：標準JIS規格配列

公衆通信回線，専用回線を利用

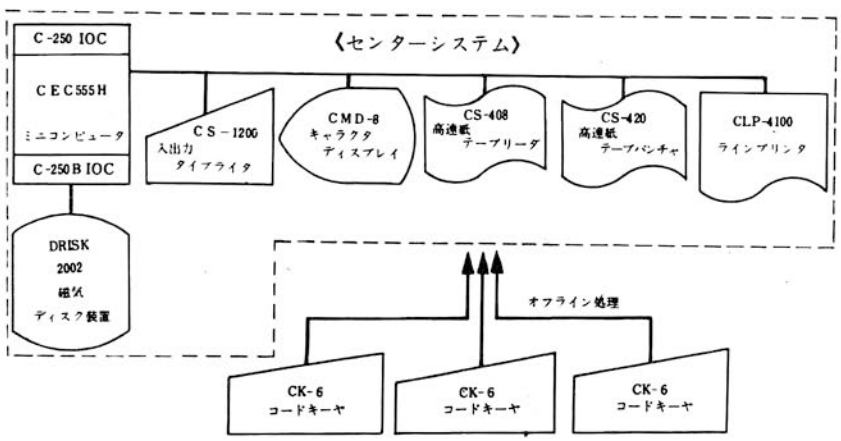
標準品CK－M2カプラ　200ボー通常電話回線利用

CK－M12モデム	1200ボー	公衆通信回線
CK－M24モデム	2400ボー	
CK－M48モデム	4800ボー	専用回線利用
CK－M96モデム	9600ボー	

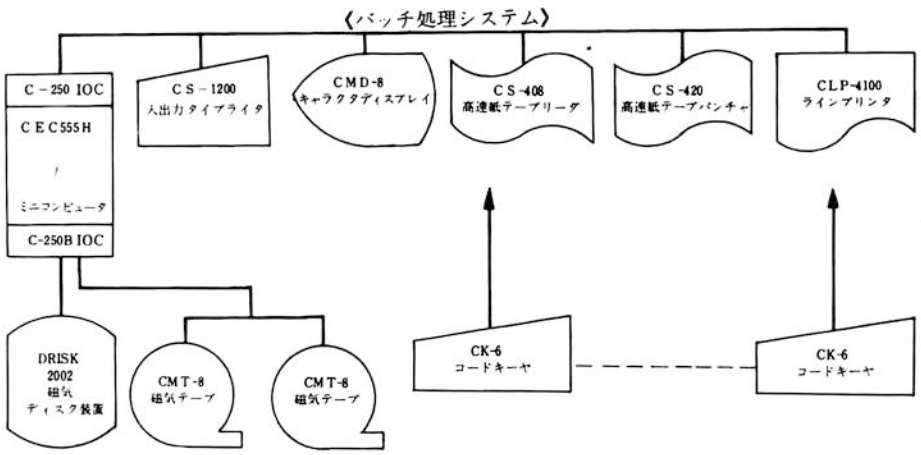
※公衆通信回線，専用回線用コントロールも用意されている（CEC555SSマイクロコンピュータ）．

3．CKシリーズコードキーヤ，CKシリーズマルチターミナルの利用例

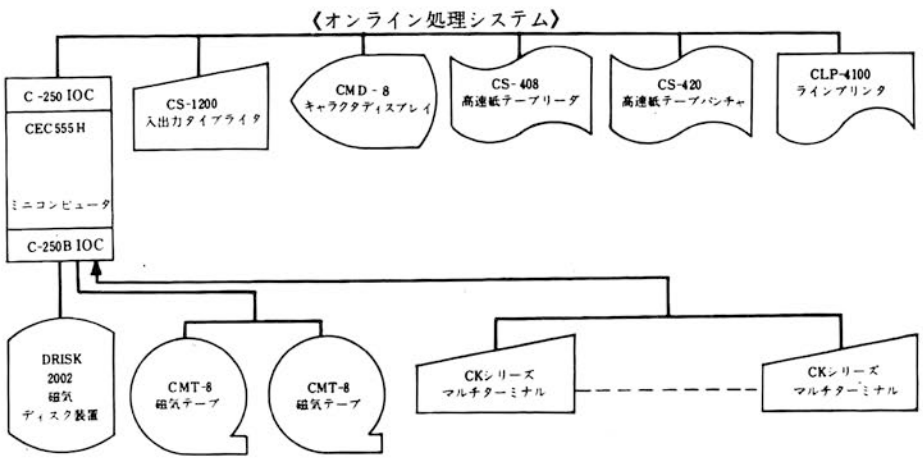
(1)　オフライン用



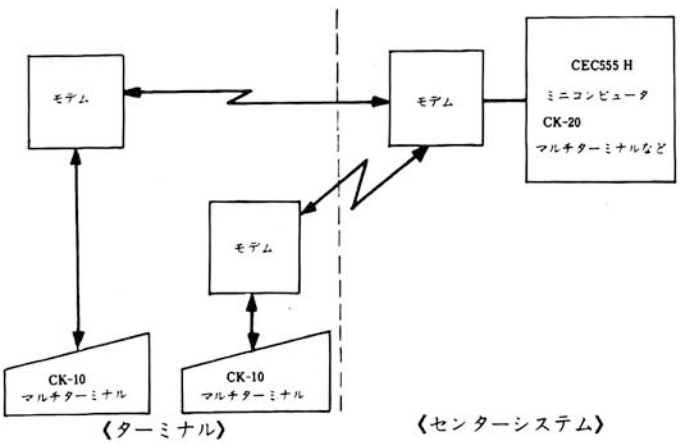
(2) バッチ処理用



(3) オンラインシステム用



(4) データ通信システムの入出力端末装置として



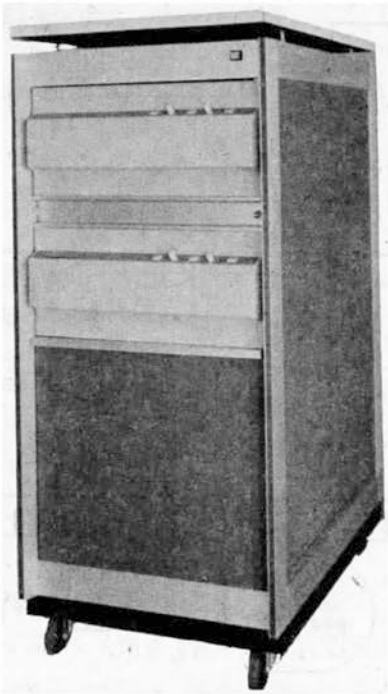
※ CKシリーズマルチターミナル，CKシリーズコードキーヤの代表的な利用例は応用編CEC EDPシステムの項を参照のこと

❸ DRISK－2002カートリッジ式磁気ディスク装置

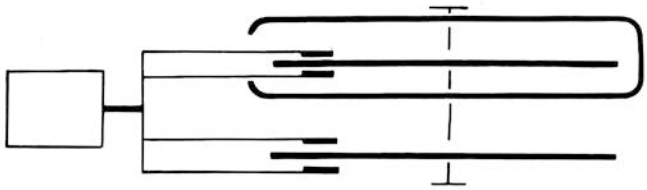
概　要

DRISK-2002ディスク記憶装置はCEC555シリーズ用の低価格，ランダム・アクセスの大量記憶装置である．

DRISK-2002磁気ディスク装置は，固定ディスクと交換可能ディスクとの組合わせで最高4,010







万ビットの情報を貯えることができる．２つのディスクは，ディスクの各表面に対して１つの腕のヘッドをもった１つの動くアクチュエイタの機構によりアクセスされる．この独特の設計により１台の装置で交換可能のカートリッジと固定ディスク間のデータ交換を可能にしている．従って交換可能のカートリッジをデータの出し入れに使う一方，固定ディスクをデータのコピーとプログラムがデータのオンライン記憶装置として使うことができる．

特　長

1．１台の装置で，4,010万ビットの容量を収納できる．
2．カートリッジ・ディスクと固定ディスク間のデータ交換が可能
3．種々のデータ・フォーマット
4．スライドにのせて19インチのラックに取りつけることができ，キャビネットの正面からだけで完全に操作可能．
5．〝完全〟エアフィルター・システムにより，周囲からの汚染を防ぐ．
6．安全検出とオプションとしてのデータ・プロテクトができる．
7．簡潔な機構による信頼性の保障

仕　様

容　量

204　シリンダ
816　トラック
13,056　セクタ　　(16セクタ／トラックモード)

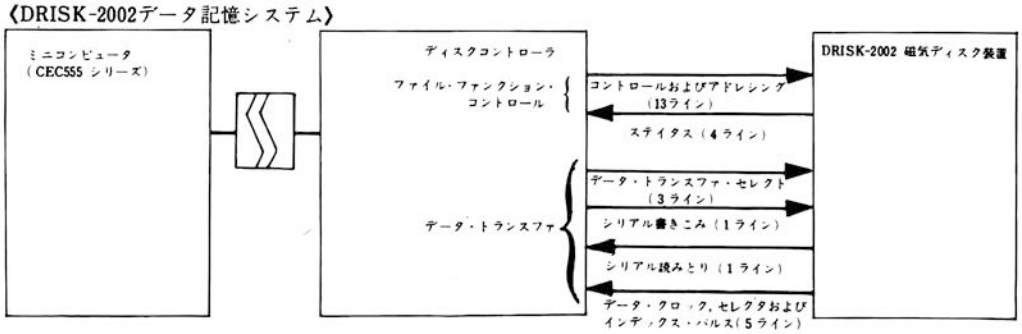
ドライブ

ビット……40,108,032
ワード（12ビット）…3,342,336

シリンダ

ビット……98,304
ワード（12ビット）…8,192

《DRISK-2002データ記憶システム》



トラック

ビット……49,152

ワード（12ビット）…4,096

セクタ

ビット……3,072

ワード（12ビット）…256

（上記の数字は実際に使えるデータの容量を示すもので，フォーマッティングのために使うことのできる追加の容量は入れていない）

アクセク・タイム（ヘット安定化時間を含む）

トラック間……35ms

平均ランダム移動……95ms

平均回転遅延……20ms

記録方法

ビット密度……2,200ビット／インチ

トラック密度……100トラック／インチ

記録方式……周波数変調

データ移送速度……1.5625Mビット／秒

ビット・セル・タイム…640ナノ秒

ディスク回転速度……1,500回／秒

カートリッジ直径……15インチ

ディスク直径……14インチ

ヘッド数……4

アクセス機構

アクチュエイタ……電磁気式ボイス・コイル・モータ

位置ぎめ機構……光学式

マウンティング……耐ショック・マウントのアルミ鋳造のベース板

衝撃および振動

14Hz 以下の周波数…0.03インチ　ピーク間（持続的）

0.3インチ　ピーク間（断続的）

14Hz 以上の周波数…0.3Gピーク（5秒以上の持続）

0.75Gピーク（5秒以下の断続）

冷　却……設置場所周囲の室内空気による強制空冷

70～90立方フィート／分（ダクト付）

入力電源……AC100V±10%

周波数……50または60Hz切替要（プリー替え）

消費電力……START時5秒間1400VA，500VA（コントロール付），本体300VA.

外形寸法……W483×H223×D301mm（DISKのみ）

周囲温度……動作時15°～35°C

湿　度……動作時30～80%

I／O命令一覧

1．DISK COMMANDの種類

（1）SENSE STATUS

使用中のDISKの状態をコントロールを通してCHECKする.

（2）WRITE DATA

DATAをDISKに書く時に用いられ，WRITE ADDRESS COMMANDでINITIALIZEされた後に有効.

（3）READ DATA

DISKからDATAを読み出すのに用いられ，DATA BREAK BUSを通してCOMPUTERにDATAが送られる.

（4）POSITION ACCESS

DISK DRIVEのREAD WRILE HEADを動かす.

（5）REFANE DATA

DISK DRIVEの特定のSECTORのDATAをRECOVERYする時，ERROR RECOVERY PROGRAMとして使用.

（6）CHECK DATA

RECOVERYされたDATAをCHECKする時に用いて，DATAとCRCのCHECKを行う.

（7）WRITE ADDRESS

書かれていないCYLINDERをINITIALIZEし，必要な時はDCIを発生する.

2. INSTRUCTION CODE

2－1 CODEについて

（1）6153 DISK CONTROLのHEAD SECTOR REGISTERにACCUMULATORの内容をSETする.

（2）6154 DISK CONTROLのCYLINDER REGISTERにACCUMULATORの内容をSETする.

（3）6174 DISK DRIVEのSTATUSをACCUMULATORにSETする.

（4）6164～6167 ERROR FLAGがSETされると次の命令をSKIP，またはOPERATIONが完了すれば次の命令をSKIP

6164	DRIVE 0
6165	DRIVE 1
6166	DRIVE 2
6167	DRIVE 3

（5）6173 ACCUMULATORの内容をCOMMAND REGISTERにSETし，DISK OPERATIONを始める

2－2 READ STATUS (6174)

この命令はDISK CONTROLからのSTATUSの内容がACCUMULATORに送られる．この命令は6173の次に使用される.

2－3 SKIP ON ATTEMTION OR ERROR (6164～6167)

この命令は動作が完了したことを示すのと，INTERRUPTが発生したことを示すATTENTIONを調べる.

ATTENTIONがSETされるとIOT命令の時SKIP信号をCOMPUTERに送り6164～6167の命令でCOMPUTERはINTERRUPTがどのDRIVEから出るのかをCHECKできる.

SELECTされたDRIVEにERRORがあると，このDRIVEのIOT命令が出た時SKIP信号が出る.

次のSTATUSでERROR FLAGがSETされる.

DATA ERROR

OVERRUN

ACCESS ERROR
ADDRESS ERROR
FLAGGED CYLINDER
END OF CYLINDER
NOT READY

3. COMMAND SET

COMMANDは6173の命令によってACCUMULATORから送られ、DISK CONTROLのCOMMAND REGISTERにIOP 1のPIRIODでSTROBEされIOP 2のPIRIODでDISKの動作が始る.

AC BITS

COMMAND

0	1	2	3	4	5	6	7	8	9	10	11	
										0	0	DRIVE 0
										0	1	DRIVE 1
										1	0	DRIVE 2
										1	1	DRIVE 3
0	0	0	0	0	0	—	—	—	—			SENSE STATUS
0	0	0	1	0	0	×	×	×	—			WRITE DATA
0	0	1	0	0	0	×	×	×	—			READ DATA
0	0	1	1	0	0	—	—	—	—			POSITION ACCESS
0	1	0	1	0	0	—	—	—	—			REFINE DATA
0	1	1	0	0	0	—	—	—	—			CHECK DATA
1	0	0	1	0	0	—	—	—	—			WRITE ADDRESS
1	0	0	1	0	1	—	—	—	—			WRITE ADDRESS

MEMORY EXTENSION BIT DCT

AC BITS

6	7	8	
0	0	0	4K
0	0	1	8K
0	1	0	12K
0	1	1	16K
1	0	0	20K
1	0	1	24K
1	1	0	28K
1	1	1	32K

(1) SENSE STATUS

このCOMMANDが出るとDISK CONTROLがSELECTされ、SELECTされたDISKのSTATUSの情報が6174でACCUMULATORに送られる.

AC BITS

0	DATA ERROR
1	OVERRUN
2	ACCESS ERROR

```
3
4       ADDRESS ERROR
5       FLAGGED CYLINDER
6       END OF CYLINDER
7       BUSY
8       NOT READY
9
10
11
```

（2）WRITE DATA

このCOMMANDはDISKがWRITE ADDRESSでINITIALIZEされた後に使用されCOMMANDが出るとDRIVEが選択され，HEAD SECTORの値がDISKに送られる．DISK CONTROLは指定のSECTORがみつかるまで待ち，指定の1つ前のSECTORを読んでハードCHECKを行う．この時，BREAK RQSTは出ない．

CYLINDER，SECTOR，HEADのREGISTERの内容と読んできた値とをCHECKし，DATAとCRCをCHECKし，DCIのBITがなくERRORがなければHEAD—SECTOR REGISTERの内容が＋1され，このところへDATAのWRITEが行われCOMPUTERとのDATA転送が行われる．

動作が終了すればINTERRUPTが発生する．

転送するDATAが256ワードの時は，1．SECTORの中全部に書かれる．256ワードより少ない時は，転送数までDATAが書かれ，256ワードになるまでZEROが書かれる．256ワード以上の時は，256ワード単位で書かれる．この場合，SECTORは自動的に＋1が行われ，これは次の状態で終了する．

1）WORD COUNTER OVER FLOWが出る．

2）ERRORが検出される．

3）END OF CYLINDERが検出される．

（3）READ DATA

このCOMMANDはDISKから読んだDATAをCOMPUTERに送る時使用され，COMMANDが出るとHEAD—SECTOR REGISTERの内容が＋1され，DISKが選択され，HEAD—SECTORの値がDISKに送られる．

DISK CONTROLは指定のSECTORがみつかるまで待ち，みつかるとDISKから読んだCYLINDER SECTOR HEADの値をCHECKする．この値が等しく，DCIがなく，ERRORがなければCOMPUTERとのDATA転送を始め，DATAがCOMPUTERに入力される．

転送が終了すればINTERRUPTが発生する．

転送するDATAが256ワードより少ない時は，その数だけCOMPUTERにDATAが送られ，256ワード以上の時は256ワード単位に転送される．この場合，SECTORは自動的に＋1が行われ，これは次の状態まで続く．

1）WORD COUNT OVER FLOWが出るまで．

2）ERRORが検出される時．

3）END OF CYLINDERが検出されるまで．

（4）POSITION ACCESS

このCOMMANDは指定されたDISKのREAD WRITE HEADを動かしCOMMANDが出ると

POSITION動作が始まる.

このCOMMANDの前にCYLINDER REGISTERとHEAD—SECTOR REGISTERをSETしておくこと.

この動作が完了した時, INTERRUPTが発生する. INTERRUPTは指定のSECTORの1つ前2.5msの間出る. 2.5ms以内にFLAG CHECKをしないと次は40ms後になる.

(5) REFINE DATA

このCOMMANDは, DATA CHARACTERをRECOVERYする場合に使用され, COMMANDが出るとCONTROLのHEAD—SECTOR REGISTERの内容が+1され, DISKが選択される.

指定のSECTORがみつかると, このSECTORに対してTUNNEL ERASEが実行される.

MULTIPLE SECTOR OPERATIONは, この命令の場合実行できない. 指定のSECTORが終るとINTERRUPTが発生して, この動作は終了する.

COMPUTERとのDATA転送はない.

(6) CHECK DATA

CHECK DATA COMMANDはRECOVERYされたDATA CHARACTERをCHECKするためにDISKにSTOREされたDATAとCRCのCHECKを行うためのものである.

このCOMMANDの実行中は, COMPUTERとのDATA転送はない.

このCOMMANDが出ると, CONTROLのHEAD-SECTOR REGISTERの内容が+1され, DISKが選択され, CONTROLは指定のSECTORがみつかるまで待つ.

指定のSECTORがみつかると, DISKの内容を読んでCONTROLのCYLINDER REGISTER, HEAD-SECTOR REGISTERの内容とをCHECKし, DATAとCRCのCHECKが行われる.

この時, COMPUTERのWORD-COUNTERの値が+1され, SECTORごとに+1が行なわれる.

これは次の状態まで続く.

1) WORD COUNTER OVERFLOWが出るまで

2) ERRORが検出されるまで

この状態のどれか1つでも検出されると, 現在のSECTORの終りで動作は終了し, INTERRUPTが発生する.

(7) WRITE ADDRESS

WRITE ADDRESS COMMANDは, 書かれていないCYLINDERをINITIALIZEするために使用する. また, DCI (DEFECTIVE CYLINDER INDICATER) を発生するのに使うCOMMANDである.

このCOMMANDは, 最初のSECTORに対してADDRESS CHECK, DCI CHECK, CRC CHECKをしないで, 指定のSECTORにDATA WRITEを送る. この部分を除けばWRITE DATA COMMANDと同じ動作を行う。

このCOMMANDが出るとDISKが選択され, 指定のSECTORがみつかるとアドレスフィールドにCYLINDER REGISTER, HEAD-SECTOR REGISTERの内容が書かれる.

また, 6173命令のACCUMLATORのBIT 5が1の時はDCIが書かれる.

(8) LOAD HEAD-SECTOR REGISTER (6153)

この命令は, CONTROLのHEAD-SECTOR REGISTERにACCUMULATORの内容をSETする. この命令は, POSITION ACCESS COMMANDを出す前に出される. また, 次のCOMMANDの前にも出すことができる.

WRITE ADDRESS
WRITE DATA
CHECK DATA
REFIND DATA

このCOMMANDにおいて，6151でHEAD-SECTORの内容がCREARされ，6152でACCUMULATORの内容がこのREGISTERにSETされる．ACCUMULATORのBITは次のようになつている．

AC BITS

0 1 2 3 4 5 6 7 8 9 10 11

HEAD 0～3 SECTOR 0～17

SECTOR INCREMENTにおいて，以下のBITまで回路でできる．

AC BITS

7 8 9 10 11

即ち，SECTORは0から37まで指定できる．

HEAD指定BITにおいて

1）AC BIT 7はHEADの上下を示し
"0"でBOTOM，"1"でTOPを表わす．

2）AC BIT 6はDISKの上下を示し，
"0"でREMOVAL，"1"でFIXを表わす．

（9）LOAD CYLINDER REGISTER (6154)

この命令はCONTROLのCYLINDER REGISTERにACCUMULATORの内容をSETしPOSITION ACCESS COMMANDの前に出される．

CYLINDER ADDRESSを指定するために，次のCOMMANDの前にも出すことができる．

WRITE ADDRESS
WRITE DATA
READ DATA
CHECK DATA
REFINE DATA

このCOMMANDが出ると，ACCUMULATORの内容がREGISTERに新しくSETされる．ACCUMULATERのBITは次のようになっている．

AC BITS

0 1 2 3 4 5 6 7 8 9 10 11

CYLINDER 000～203

4．PROGRAMING DESIGN

（1）THREE CYCLE DATA BREAK

DATAはCOMPUTERとCONTROLの間でDATA BREAK BUSを経由して転送される．転送に対して，次のMEMORY LOCATIONが割り当てられていて，これはデータの転送の前にSETされなければならない．

(WORD COUNTER = 7750
CORRENT ADDRESS = 7751)

（2）WRITE CHECKING

MEMORYの中のDATAと全く同じDATAが，DISKに書かれたかをCHECKする方法について，PROGRAM CHECKが必要である．

重要なDATAに関しては，READ-BACK CHECKを行って，MEMORYの内容とWORDごとのCHECKが行われなければならない．これは，DISKに対してシリアルDATA転送をCHECKするだけでなく，COMPUTERとのPARALLEL DATA転送CHECKが必要である．その他に関しては，DISKに書かれたDATAのCRC CHECKのためのCHECK DATA COMMANDを出すだけで十分である．これは，DATA CHARACTERとCRCのCHECKが行われる．

（3）DISK INITIALIZE

SYSTEMを使用する前に，NEWパック（カートリッジ）はDISK INITIALIZE PROGRAMを使用しなければならない．

このPROGRAMは，全てのSECTORをINITIALIZEし，各々のSECTORにADDRESSが書かれる．

CYLINDERに欠陥がある場合には，DCI（DEFECTIVE CYLI NDER INDICATER）が書かれなければならない．

これは，DATA TESTを通してみつけられる．

1） DISK DATA TEST

DATA TESTは，各CYLINDERに対して次の6回のTESTを行う．

（a） 各 CYLINDERは，全てのDATA PATTERNを書く．

（b） 各 CYLINDERの全てのDATAに対して，10回CHECK DATAを行う．

OCTAL
0 0 0 0
7 7 7 7
1 2 5 2
5 5 5 5
7 0 7 0
7 4 1 7

このルーチンを実行中にERRORが出れば，このCYLINDERはDCIが表示されなければならない．

2） DIC （FLAGGED CYLINDER）

CYLINDERに欠陥があれば，CYLINDERの各SECTORは表示されなければならない．

この場合，AC BIT 5＝1でWRITE ADDRESS COMMANDを使用する．

（4）ERROR RECOVERY

ERROR RECOVERY PROGRAMは，発生したERRORを確めなければならない．

ERRORの状態を確めるには，以下の手順をすすめる．

1） DATA ERRORか

READ DATA COMMANDでERRORが出れば，16回DATAをREADする．

くり返しERRORが出れば，REFINE DATA COMMANDを出して，READ DATA COMMANDを実行する．

依然としてERRORがあれば，このＣＯＭＭＡＮＤは終る．

WRITE DATAかWRITE ADDRESS COMMAND 実行中にERRORが出る時

は，再度このCOMMANDを実行する．

2） WRITE CHECKING時のERROR（4－2 参照）

WRITE CHECKルーチンにおて，READ DATAか，またはCHECK DATA COMMANDの時にDATA ERRORが発生すれば，完全なWRITEをするために16回実行する．

3）ADDRESS ERRORかACCESS ERRORの時

READ DATAかCHECK DATAの時，ADDRESS ERROSが出れば再度実行する．

4）OVERRUNの時

再度実行する．

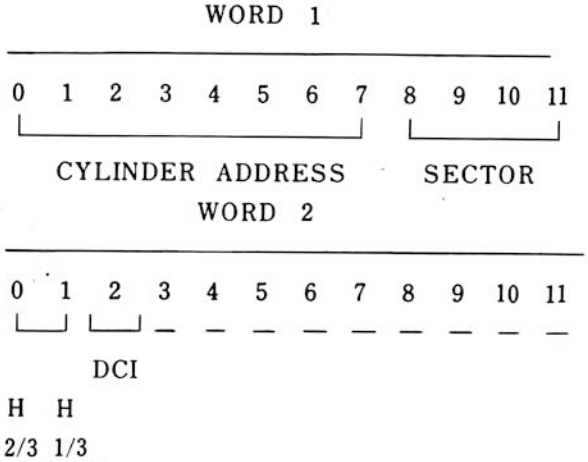
（5）ERROR STATUS CHECKについて

STATUS WORDの中に出るERRORを確める1つの方法として，確実に動作するDISKを接続すれば，ERRORの状態がCHECKできる．

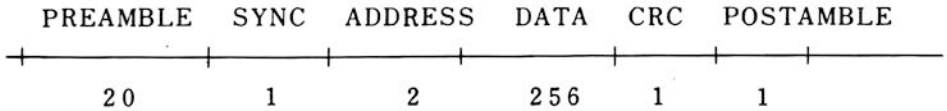
5．FORMATINGについて

1）ADDRESS FIELD

各SECTORのADDRESSは24 BITSで構成されて，この内容は，CYLINDER SECTOR，HEADが含まれていて，場合に依つてDCIが加えられる．



2）SECTORについて



6．ERRORについて

（1）DATA ERROR

このERRORはSYNC ERROR，RD WR DISABLE，DATA ERRORが含まれている．

(a) SYNC ERROR

READにおいて，SYNC MARKがみつからなかった時，あるいはSYNC MARK以外のものの時に出る．

(b) RD WR DICABLE

POWER FAILおよびRD WR ENABLEでない時に，SYNC REC CRC AREAが出ている時に出る．

(c) DATA ERROR

DATA CHARACTERとCRCが同じでない時に出る．

（2）OVERRUN

これには、DATA OVERRUNとSECTOR OVERRUNがある.

(a) DATA OVERRUN

2 WORD READY以上になった時に出る.

(b) SECTOR OVERRUN

RD WR GATEが次のSECTORにひっかかる時に出る.

（3）ACCESS ERROR

POSITION ACCESSのERRORを示し、SENSE COMMANDの6174でCLEARされる.

（4）ADDRESS ERROR

HEAD、SECTOR、CYLINDERのどれかのERRORを表わし、SENSE COMMANDの6174でCLEARされる.

（5）FLAGGED CYLINDER

DCI BITが検出されたことを示し、SENSE STATUS COMMANDの6174でCLEARされる.

（6）END OF CYLINDER

SECTORの指定が規定より大きくなっていることを示す.（これについては3－8）参照

7）BUSY

DISK DRAVE BUSYかCOMMAND AND BUSYを示す.

8）NOT READY

DISKがREADYになっていないことを示す.

取扱注意事項

1．OPERATION上の注意

（1） DISK ARTRIDGE MTは使用前に必ず何か不具合がないかどうか外観を目視検査して下さい.

（2）DISK CARTRIDGEは**カバーなし**で決して取扱わない下さい.

（3）コーヒー、その他の飲物をDRIVE、TAPE TRANSPORTやDISC CARTRIDGEやMTに近づけないようにして下さい.

汚れた手、鉛筆その他も同様DISK面、MT面に近づけないようにして下さい.

衝撃、圧力などはDISK面に歪みや、傷をつける原因になります.

（4）タバコやその灰などはDISK面、MT面を汚す原因になります. 従って、DISK、MTを使用の部屋では禁煙にして下さい.

（5）DISK CARTIDGE、MTにラベルが貼られることがありますが、ラベル上の表示変更を消しゴムを使ってCARTIDGE、DRIVE、MT、TRANSPORTの近くで行なわないようにして下さい.

消しゴムや紙くず類はDISK CARTRIDGE、MTを汚す原因になります.

（6）DRIVEのAIR FILTERはコンピュータルームで使用されている場合は1年で交換して下さい. そうでない場合は4～6ケ月ごとに交換しなければなりません.

（7）MT取付け時のテープは正しくクランプしている事を確認して下さい.

またテープのエッジがリールに接触していないように注意して下さい.

テープリールの装着時には中心部を抑えるようにして下さい.

（8）マシン・ルームは毎回ぬれ雑巾で清掃して下さい. ワックス磨きの回数はできるだけ減ら

し，ほうきや，羽根製のハタキなどで掃除しないで下さい．ほこりをたてることはさけて下さい．バフ磨きの場合は針金製の研磨材を使用しないで下さい．

2．操作上の注意

（1）CARTRIDGE，MTを使用する前に2時間はマシン・ルームの温度へならす必要があります．正確なREAD WRITEを行うためには是非必要です．

（2）CARTRIDGEをDRIVEへ，MTをTRANSPOTにINSTALLする前に，カバーのホコリ，汚れを落すようにして下さい．

（3）MTをTRANSPORTへ装着する場合，フィンガーホールドを使ってリールを回転させ，BOT EOT間のテープ面には絶対に手を触れないで下さい．

（4）CARTRIDGE，MTの脱着は必ず，回転がSTOPするまで待って行なうようにして下さい．

（5）CARTRICGEをはずした後は直ちに，BOTTON COVERをつけて下さい．

3．　DISK DRIVEの点検

（1）6ケ月点検（メンテナンス　マニュアル参照）

① SERVO POSITIONER MECHANISM

② READ WRITE READ INSPECTION

③ DISK CLEANING BRUSHES

④ DETENT MASK，CHECKEN AND CLEANED

ユーザー側でおこなうべき保守

1．AREフィルター交換

2．HEADクリーニング

3．ブラシ交換

4．DISK面のクリーニング（必要に応じ）

5．ファン

本I／Oを使用するにあたっての計算機最小構成

TTY＋CPU（555H or 555M）＋DISK-2002

❹ CFD−300ミニディスク装置

概　要

CFD-300ミニディスク装置は固定ヘッド磁気ディスク駆動装置である．この装置の特色は，トラックごとにヘッドが設けられており，平均10msの高速でのランダムアクセスが可能であり，価格もきわめて低廉で補助記憶装置として最適である．記憶装置は最大1.6Mビットまで収納可能で信頼性も高く，事務システムなどの広汎な分野で利用できる．

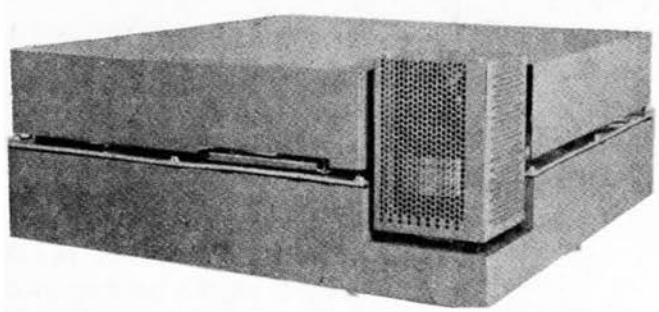
特　長

1．小形ながら，最大3.2Mビットまで記憶可能である．
2．トラックごとにヘッドを設け，平均アクセス時間が10msと高速のランダムアクセスが可能である．
3．密閉構造により，周期からの汚染防止が万全である．
4．機械的部分は回転体のみで，信頼性の向上を計っている．
5．横型，堅型共に実装が可能である．
6．50/60Hzの周波数変換が容易にできる．

仕　様

情報用トラック…32または　64
情報用予備トラック…4





タイミング用トラック…1　　1
タイミング用予備トラック…2　　2
総記憶容量……1,536Mビットまたは3.072Mビット
128K語，または256K語
1トラック当りの記憶容量…48Kビット
4K語
1セクター当りの記憶容量…3,072ビット
256語
（上記数字は実際にユーザーの使用できる容量を示すものである）
記録密度……1800ビット／インチ
転送速度……2.5Mビット／秒（50/60Hz）
記録方式……周波数変調
DISK回転速度……3,000rpm±5%
電源条件……AC100V±10%、50/60Hz、単相
起動時：5A以下
定常時：2A以下
DC +24V±5　1A
+5V　±5%2A
−5V　5%0.5A
DISK起動時間……2分（2分経過後，使用可能）
情報の誤り率……10回の読み直しで正しい読み出しができない誤り10^{11}ビットの読み出しを行って1件以下
入力電源……AC100V±10%
周波数……50〜60Hz，50または60Hz切替要（DISKのプーリー交換が必要）
消費電力……5A以下（起動時），3A以下（定常時）＊DISKを1台DRIVEする場合.
外形寸法……W413×H170×D413mm
重　量……18kg
周囲温度……10°〜40°C
湿　度……30〜80%

I/O命令一覧

6602　(SET　DISK　ADDRESS)
6605　(SET　WORD　COUNTER)
6606　(SET　CORE　ADDRESS)
6623　(SET　COMMAND)
6624　(READ　STATUS)
6614　(SKP　ON　ATTENTION　OR　ERROR)
6601　(READ　WORD　COUNTER)

取扱注意事項

運搬時にはDISKをとりはずして別に梱包すること.

ユーザー側でおこなうべき保守

ベルト点検（1回／1年）

本I/Oを使用するにあたっての計算機最小構成

555H，TTY

❺ CFP−500フロッピィディスク装置

概　要

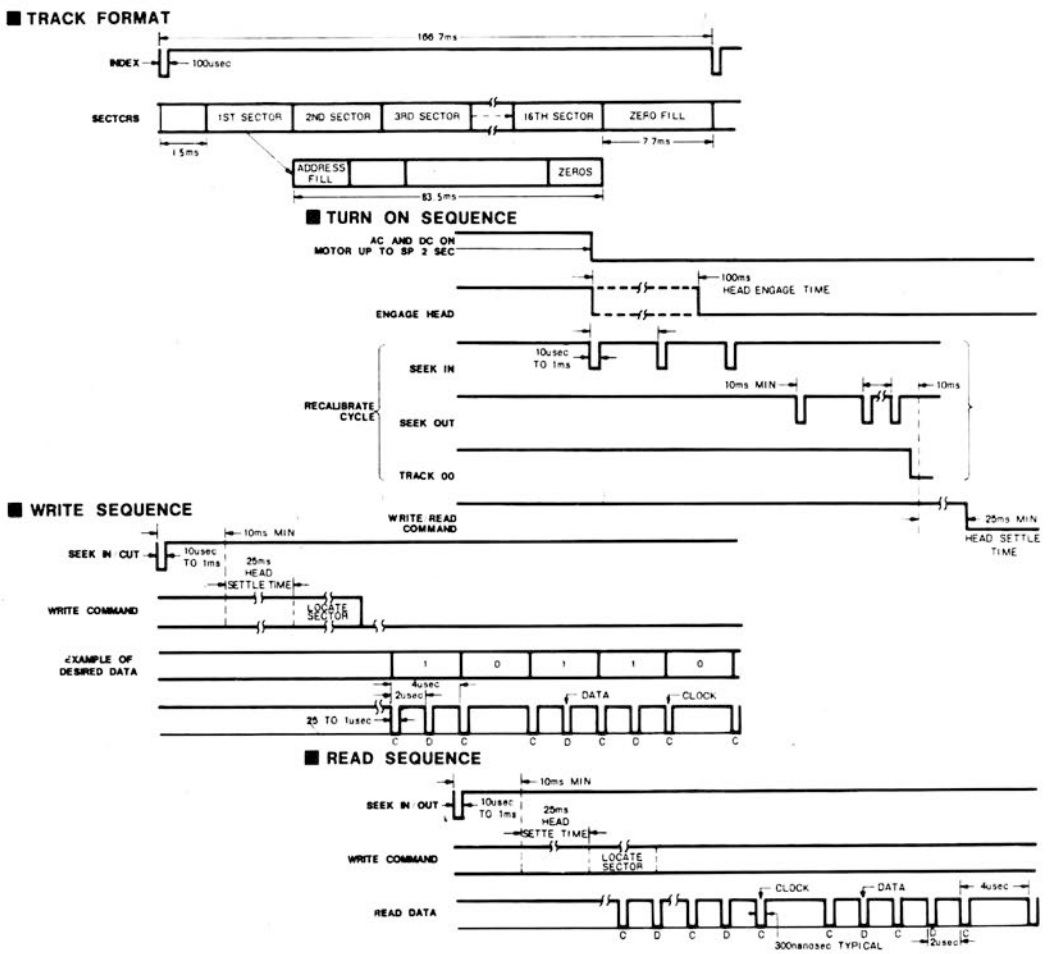
CFP-500フロッピィディスク装置は、CEC555シリーズ用の低価格、ランダムアクセスの記憶装置である.

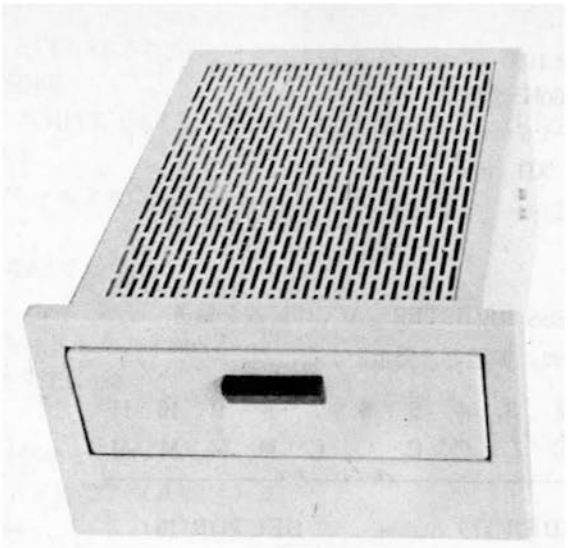
1ユニット150KW（12ビット／word）の記憶容量をもち、4ユニットまで接続することができる. また、ディスケットを自由に交換することができる. 小型システムのデータファイルに便利に利用できる.

データ転送はDSA方式によりおこない、命令はIOバスを通しておこなわれる.

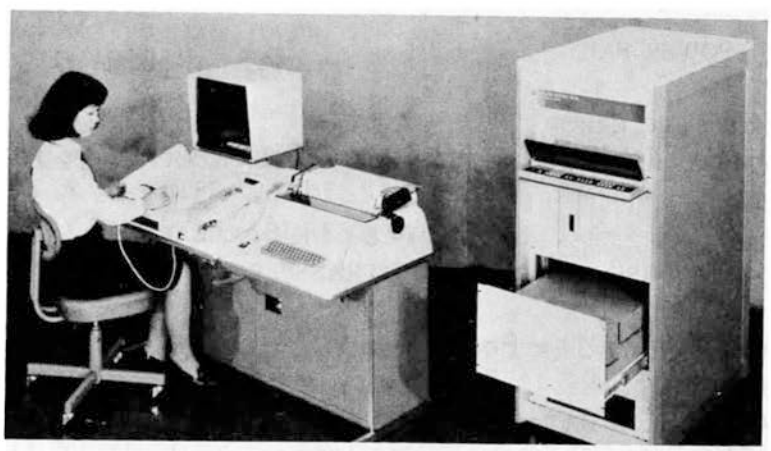
仕　様

容　量……トラック48T／インチ…77トラック
トラック容量……2KW（12ビット／語）
セクタ……16セクタ（128語／秒）





記録密度……3270ビット／インチ　max
回転送度……360rpm.　167ms／r
記録方式……周波数変調
データ転送速度…250.000ビット／秒
アクセスタイム…トラック－トラック間10ms
セットリングタイム…25ms
ヘッドロード．ヘッドアップ……100ms
電　源……AC100V，50／60Hz



周囲湿度……10℃～40℃（40°—100F）
相対湿度……10～80%
入力電源……AC100V±10%
周波数……50または60Hz切替要
消費電力……約320VA（ドライブ1台のとき）
外形寸法……W　XH　XD　mm
周囲温度……10°～309C
湿　度……10～80%

I／O命令一覧

6554　RECORD ADDRESS REGISTERへACCの情報を転送.
この命令の実行後に，割込みは発生しない.

ACC	0	1	2	3	4	5	6	7	8	9	10	11
		C	C	C	C	C	C	C	M	M	M	M
		CYLINDER(77)							SECTOR(16)			

6574　READ STATUS
CONTROLLERのSTATUSをACCに転送ACCに転送する前にACCの内容をCLEAR

ACC	STATUS
0	SYNC ERROR
1	CRC ERROR
2	DATA OVER RUN
3	SECTOR OVER RUN
4	ADDRESS ERROR
5	DCI
6	END OF CYLINDER
7	BUSY
8	SEEK EROR
9	——
10	POWER FAIL
11	——

1．SYNC ERROR，CRC ERROR
WRITE ADDRESS，WRITE DETA，READ DATA，CHECK DATA命令を実行中に，ERORを検出した場合.
ERORは，インストラクション6574を実行することによりCLEARされる.

2．DATA OVER RUN SECTOR OVER RUN TRANSFER LATEとDATA CLOCKをCHECK.
CLEARは6574を実行することによりCLEARされる.

3．ADDRESS ERROR
READ DATA，CHECK DATA，WRITE DATAの実行中にCYLINDER，SECTORをDISKから読み出し，その内容とRARが一致しない場合にSETする.
6574を実行することによりCLEARされる.

4．DCI

WRITE DATA、READ DATA、CHECK DATA実行中にDISKから読みだしたDATAにDCIを検出するとSET.

6574を実行するとCLEARされる.

5. ENDOF CYLINDER

READ DATA、WRITE DATA、WRITE ADDRESS、CHECK DATA実行中にSECTORが15を越える時にSET.

6574を実行することによりCLEARされる.

6. BUSY

DISK DRIVEがBUSYのとき.

7. SEEK ERROR

SEEK ERRORを起こしたときSET.

O CYLINDERをSEEKするとCLEARされる.

8. POWER FAIL

AC100Vが85V以下になったときSET.

6574を実行することによりCLEARされる.

6573 (L COM)

COMMAND REGISTERへACCの内容を転送する.

1. COMMAND

ACC	0	1	2	3	4	5	6	7	8	9	10	11
	0	0	0	0	X	X	X	X	X	X	M	M
	0	0	0	1	X	X	N	N	N	X	M	M
	0	0	1	0	X	X	N	N	N	X	M	M
	0	0	1	1	X	X	X	X	X	X	M	M
	0	1	1	0	X	X	X	X	X	X	M	M
	1	0	0	1	X	X	N	N	N	X	M	M
	1	0	0	1	X	1	N	N	N	X	M	M
	COMMAND						FIELD				UNIT	

0 0 X X SENSE STATUS

0 4 X X WRITE DATA

1 0 X X READ DATA

1 4 X X POSITION DATA

3 0 X X CHECK DATA

4 4 X X WRITE ADDRESS

4 5 X X WRITE ADDRESS WITH DCI

2. COMMADの内容

(1)SENSE STATUS

選択されたUNITEのSTATUSをSETする. そしてこの命令の後にACCに転送するために、6574の命令を実行する. この命令実行後に割込みはしない.

(2)WRITE DATA

WRITE ADDRESS命令によってINTIALIZEされた後に有効となる. WRITE DATAの命令は、CPUからの情報を書き込む. 情報は1SECTORに128WORDSを1群として書き込み、もしCPUからの情報が128WORDSに満たなかった場合は、その部分にダミーとして零を

書き込む．128 WORDSを越える場合は，次のSECTORに移り，情報を書き込む．
WORD COUNT OVERFLOW ERRORが発生した場合，割込みを発生する．

(3)READ DETA

READ DETAは，DISKからCPUへ情報を転送する．情報は1 SECTORに128WORDSを1群として読み込む．もし，DISKからの情報が128WORDSに満たない場合は転送を終了する．128WORDSを越える場合は，次のSECTORに移り，情報を読み込む．
WORD COUNT OVERFLOW，ERRORが発生した場合は，転送を終了し，割り込みを発生する．

(4)POSITION ACCECC

NEW ADDRESSがすでに転送されている場合において行なう．POS ITIONが終了したところで割込みを発生する．

(5)CHECK DATA

CRC CHECKを行なうだけでDATAの転送は行なわれない．SECTOR数をカウントし，WCOFがでるまで行なう．

取扱注意事項

1．メディアの取扱いについて

フロッピィディスクは，磁気テープ，磁気ディスクと同様の取扱いが必要である．

(1)熱や太陽熱の直接にさらさない．
(2)磁界にさらさない．限界は50エルステッド．
(3)ディスク面にさわらないこと．クリーニングはしないこと．
(4)外部ラベルに情報を記入する場合は，ディスクに貼りつける前に書いておくこと．
(5)紙クリップ等ではさまないこと．
(6)使用済み，あるいは記録済みのディスクは，すぐに保護用紙封筒に入れておくこと．
(7)折り曲げないこと．積み重ねる場合は10枚またはそれ以下する．保管棚では垂直に立てるか，あるいは，あるいは納入の際に使われる5パック用の紙箱に入れる．
(8)新しいラベルを貼るときは，古いラベルをとってから貼りつけること．

2．設置について

フロッピーディスクは，ほこりの多い所で使用すると寿命が短かくなり，ドライブのヘッドにも影響するので一定の環境の整備が必要である．

動作範囲が10°～35°Cであるので，温度差の大きいところでは使用できない．

装置は良好の接地をとる必要がある．

ユーザー側でおこなうべき保守

1．筐体のフィルタは環境の良いところで15日に1度掃除すること．
2．ドライブのヘッドは半年に1度はクリーニングすること．

本I/Oを使用するにあたっての計算機最小構成

・555シリーズ（メモリブレーク装置のあるもの）．
・CS-100

❻ CFD－130磁気ドラム記憶装置

概　要

磁気ドラム記憶装置は，主記装置の容量を補う目的で電子計算機システム内に接続され，主に呼出し頻度の高いシステムプログラムなどの記憶に使用される．

情報呼出し速度は磁気テープ装置にくらべ約10～100倍程度の速さをもち，かつ，大量のデータが記憶できるので経済性の面からも効果的な補助記憶装置である．

また，磁気テープ装置など他種の補助記憶装置に比べ構造が簡単で故障を除くあらゆる環境条件にも適応できる．信頼度の高い補助装置である．

特　長

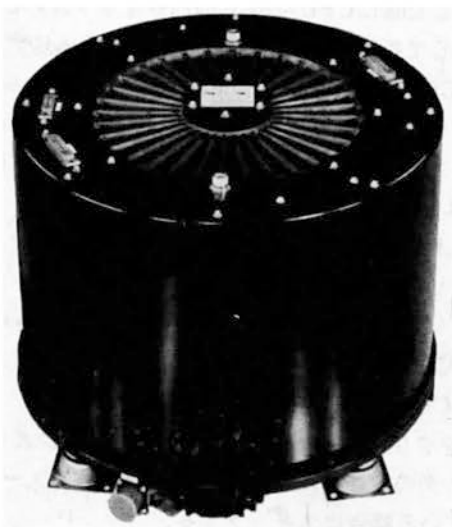
1．電源を切っても情報が消失しない．
2．非破壊読み出しが可能．
3．高速で書き込み，呼出しがおこなえる．
4．機構的に信頼度が高い．

用　途

1．コボルシステム用の補助記憶装置
2．事務計算システムの外部記憶装置として
3．オンラインシステムのファイルメモリ用
4．小型システムの主記憶装置

仕　様

記憶容量……131，072語
（12ビット＋1ビット／語）
2.048語／トラック



転送速度……50Hz　38.9μs／語
　　60Hz　31.2μs／語
平均アクセスタイム…50Hz　10.3ms
　　60Hz　8.6ms
トラック数……データ64トラック
　　（予備8トラック）
　　タイミング3トラック
　　（予備1トラック）
記憶方式……FM
振動（短時間）…動作時　0.3g以下
　　非動作時2g以下
関連機構……CFD-64磁気ドラム装置64K語．アクセスタイム10ms
入力電源……AC100V±10%
周波数……50または60Hz専用
消費電力……300VA
外形寸法……W600×H725×D750㎜
周囲温度……25℃
湿　度……45%

I／O命令一覧表

DCMA	6601	ドラムメモリアドレス・パリティエラー，転送終了インタラプットフラグ，転送速度レジスタ，アドレス一致レジスタ，ドラムエナージレジスタをクリアする．
DMAR	6603	DCMAとコンバインした命令であって，クリアはDCMAに準ずる．クリアした後にCPUのACCの内容をドラムメモリアドレスに転送する．そして指定のデータをドラムからCPUにデータ転送を開始する． ACC（00～11）⟶DMA（00～11）
DMAW	6605	DCMAとコンバインした命令であって，クリアはDCMAに準ずる．クリアした後にCPUのACCの内容をドラムメモリアドレスに転送する．そして指定のデータをCPUからドラムにデータ転送を開始する． ACC（00～11）⟶DMA（00～11）
DCEA	6611	ドラムトラックレジスタとフィールド指定レジスタをクリアする．
DSAC	6612	アドレス一致レジスタがセットされていればスキップする．そしてACCはクリアされる．
DEAL	6615	ドラムトラックレジスタとフィールド指定レジスタをクリアする．クリアした後にACCの内容をドラムトラックレジスタとフィールド指定レジスタに転送するACCの内容は変らない． ACC（06～08）⟶EF（3～1） ACC（01～05）⟶DT（5～1）
DEAC	6616	ACCをクリアした後に，ドラムトラックレジスタとフィールド指定レジスタの内容をACCに転送する．アドレス一致レジスタがセットされている場合はスキップする．

EF（3～1）⟶ACC（6～8）
DT（5～1）⟶ACC（1～5）
シンクマーク⟶ACC 00
転送速度レジスタ⟶ACC 09
書込み状態において書込み禁止SW〝ON〟⟶ACC 10
パリティエラー⟶ACC 11

DFSE 6621 パリティエラーなしで転送速度レジスタが正常で，かつ書込み禁止SW 〝OFF〟の場合にスキップする．

DFSC 6622 転送終了インタラプットフラッグがセットされている場合はスキップする．

DMAC 6626 ACCをクリアした後にドラムアドレスレジスタの内容をACCに転送する．

DMA（00～11）⟶ACC（00～11）

読み出しの場合は最終アドレスに〝1〟を加えた内容となり書込みの場合は最終アドレスとなる．

ブートストラップ

```
*200    6603
 201    6622
 202    5201
 203    5604
 204    7600
7750    7576
7751    7576
```

取扱注意事項

・動作中は強い振動を与えないこと．
・ACラインはできる限り1SYSTEMを独立して引くこと．

ユーザー側でおこなうべき保守

必要なし

本I/Oを使用するにあたっての計算機最小構成

・CPU4K、DSAまたはDMA付き
・CS-100、CS-408
・CFD-130

❼ CMT－8磁気テープ装置

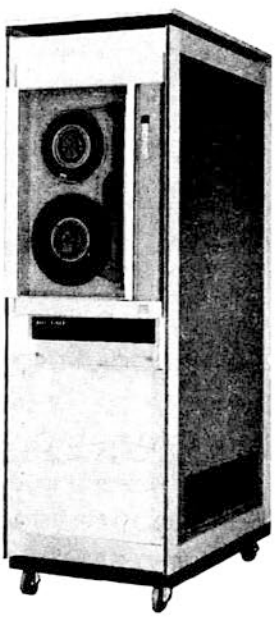
概　要

ミニコンピュータシステムで大量の情報量を処理するための外部（補助）記憶装置，あるいはデータ集録自動処理システムのデータギャザリングユニットとして使用する．

磁気テープ装置は，他の外部（補助）記憶装置に比べ記録容量が高く，大量データ収集に適し，また，テープがリールごとに交換できるので，データのファイル，情報の交換用として非常に便利である．IBMコンパティブルのデータファイルが作製できる．

CMT-8　磁気装置は，最大4台までのテープハンドラーが制御でき，CEC555シリーズミニコンピュータシステムを中型なみの機能に拡張することができる．

テープの駆動はシングルキャプスタンダイレクト方式で一定の速度に加速するまでわずか伸び，摩擦しか生じない．





また，急激な加速，停止時の慣性を吸収するように，スライド・キャリッジとテンション・アーム機構を採用している．

特　長

1．大容量・高密度である．
2．最大4台までのテープハンドラーが制御可能である．
3．テープの駆動はシングル・キャプスタンダイレクト方式である．
4．システムはサブルーチンソフトウェアパッケージで制御される．
5．フルインタラプト機能をもっている．
6．高速のリード・アフターライト・ノッドを備えている．
7．密度選択が可能である．
8．パリティ選択が可能である．

用　途

1．データ検索
2．プロセスコントロール
3．データ集録
4．大容量のデータ集積
5．データ分析
6．データのファイル
7．プログラム選択

仕　様

記録ヘッド……リード・アフターライト・ノンコンタクトイレーズヘッド
記録密度……9トラック800ビット／インチまたは7トラック800／556／200ビット／インチ
増設台数……最大　テープハンドラ　4台
テープ速度……75インチ／秒（45／325インチ／秒）25、37.5、45、75インチ／秒の各種
データーレート…60KHz:800ビット／インチ
リワイド速度……300インチ／秒、2400フィートテープで100秒以内（リワイド中テープはヘッドと接触せず）
テープ形式……IBMコンパチブル
パリティ……VRC
（リード／ライト）LRCまたはCRC
テープ幅……1 2インチ
リールサイズ……10 1 2インチ
入力電源……AC100V±5%
周波数……50／60Hz、50または60Hz切替要
消費電力……約500VA
外形寸法……W570×H1510×D590㎜
重　量……136kg
周囲温度……15°～35°C
湿　度……20～80%

I/O命令一覧

6703　Load and execute command
6704　Read status into Acc.
6711　Skip if MT. flag= 1
6716　Load record blocks (2'complement) and clear Acc.
6725　Load WC.(2'complement) and clear Acc.
6726　Load CA.(−1) and clear Acc.
6731　RELECT SELECTED a CPU.
6732　SELECT CPU When the other CPU not SELECTED
6734　Skip if selected a CPU.

ユーザー側でおこなうべき保守

ヘッドクリーニング（毎朝）

本I/Oを使用するためにあたっての計算機最小構成

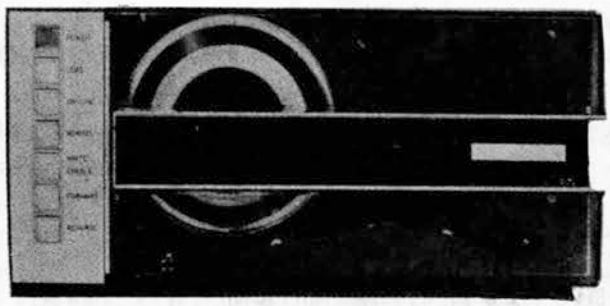
DSA付きのCPU

❽ CMT－5磁気テープ装置

概　要

データ量が飛躍的に増大した今日，磁気テープはデータの記録，再生，保存には理想的な記録媒体であり，磁気テープを有効にシステムに接続することにより，従来ミニコンピュータでは処理できなかった大量のデータ処理も可能になった．

CMT-5は小形，軽量，低価格，高性能という画期的なテープメモリシステムでCECのあらゆるシリーズに接続することができ，計測データ，試験データなどをはじめとしてあらゆる産業用，医学用その他広範囲な分野で使用でできる，データ集録ターミナルである．





特　長

1．低価格
2．小形・軽量
3．高性能
4．CECのあらゆるシリーズに接続可能
5．IBMコンパティブル
6．小型で800BPIの記録密度

仕　様

テープスピード…12.5インチ／秒（6.25～12.5インケ／秒可能）
ISV±3％以内
LSV：±1％以内
REWIND：50インチ／秒（600フィートテープ2分30秒以内）
記録密度……800ビット／インチ（556、200ビット／インチオプション）
記録方式……9トラック　ASCⅡ0.6インチIRG（IBM-2400シリーズと互換性あり）
7トラック　NR(D)I0.75インチIRG（IBM-7330、729シリーズと互換性あり）
データ転送速度

	200BPI	556BPI	800BPI
12.5IPS	2.5 kc	6.95kc	10kc

ヘッド……デュアルスタックヘッド
READ/WRITEギャップスペース
7トラック　0.150±0.005インチ
9トラック　0.150±0.005インチ
WRITE／ERASEスペース
7トラック　0.330±0.005インチ
9トラック　0.340±0.005インチ
スタート／ストラップ距離…0.19±0.02インチ（12.5インチ／秒）
スタート／ストップ時間…30ms＋2ms（12.5インチ／秒）
使用テープ……幅：½インチ、厚さ：1.5ミル、長さ：600フィート（以下）
リール：IBMタイプ7インチ（以下）
入力電源……AC100V±5％
周波数……50～60Hz
消費電力……200W
外形寸法……W483×H237×D370㎜（トランスポート）
W430×H150×D590㎜（コントロール）
重　量……15.5kg（トランスポート）
18kg　（コントロール）
周囲温度……5°～45°C
湿　度……20～95％

I O命令一覧

MSF　(6701)：FLAG　SENNSE命令である．これは"AC"の内容で3つのFLAG　SENSE命令を

定めており、FLAGをSENSEとね結果ONであればスキップする.

MWB(6702)：READ／WRITE系命令である. これは"AC"の内容でDATA　COMMANDをMACへ転送する命令を定めている.

MRB(6704)：入出力装置の情報をCPUに送るために使用される命令である. これは"AC"の内容でDATAとSTATUSをCPUに転送する命令の2種類を定めている.

取扱注意事項

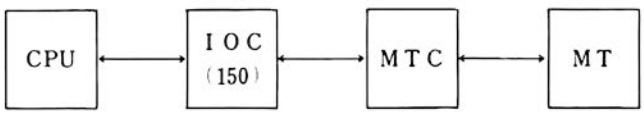
1. テープは確実に着装すること.
2. 急激な振動を与えないこと.
3. 磁性面に手や指で触れないこと.
4. ゴミ、ホコリ、繊維くず等が付かないように囲りをきれいにしておくこと.
5. 強い磁界の発生する機器の近くで使用しないこと.

ユーザー側でおこなうべき保守

テープパスの清掃：毎日使用する初めか少なくとも8時間ごとに指定清掃液で清掃すること.

1. ヘッド……ヘッドクリーナを綿棒にしみ込ませ表面をふくこと.
2. ガイド……ヘッドクリーナを綿棒にしみ込ませ表面をふくこと.
   特にセラミックの表面を酸化物が残らないように注意すること.
3. キャプスタン…純粋アルコールでキャプスタン表面を掃除すること.

本IOを使用するにあたっての計算機最小構成



❾ CMT－100カセット磁気テープ

概　要

ＣＭＴ－100カセット磁気テープ装置はダイレクトアクセスのできるカセット式の磁気テープ装置で，ミニコンピュータの補助記憶装置として，システムの記憶能力，データ処理機能を大幅にレベルアップすることができる．

本装置のデータ呼出時間は，300フィートのテープの場合，両方向平均12秒と高速でテープの上のどこからでも所定の位置にダイレクトアクセスする．

ＣＭＴ－100磁気カセットテープはあらゆるデータのファイル，検索が手軽におこなえる利点があり，事務計算，あるいは教育面で有効に使用することができる．

特　長

1．ダイレクトアクセスができる．
2．高速で両方向の作動が可能である．
3．ダイレクトアクセスで130K語以上をテープ上に記録できる．
4．データの転送率は500キャラクタ／秒
5．データの記録密度は100BPI.
6．読取り，書き込み速度は５IPS.
7．プログラムコントロールによりブロック長可変が可能である．

動作仕様

カセット数……１台のCMT－100当り１個



データ記録方法…ダイレクトアクセスもしくは逐次記録

ダイレクトアクセス…高速に両方向へ動作し，任意のデータファイルまたはデータ・ブロックを探し出せる．ソフトウェアによって，コンピュータはテープ・アドレス信号を調べて，所定の位置を検出する．この信号はカセットのスピンドルにつけてある光電検出装置から送り出される．

アクセス・タイム…300フィート・テープにおいて平均アクセス・タイム12秒以下

ブロック長……プログラムコントロールによって変化可能

システムとしての動作…ソフトウェアにより，１台のコンピュータについて最高８台までのＣＭＴ－100を同時に接続できる．複数台の装置を同時に動作させることが可能である．

同時動作……同時に複数台のＣＭＴ－100が，読み書きを任意におこなうことができる．この動作は別々のＩ／Ｏ領域からおこなうことができるので，別々の仕事を同時におこなわせることが可能である．

システムとしての容量…28.8Mビットまたは2.88M字（１字８ビット＋ビットパリティ）（８台接続した場合）

クロック……データから作る．

誤り検出……ハードウェア的にパリティビットおよび同期信号ビットを検出

デッキ仕様

テープヘッド数…１

レコード・トラック数…２

モータ……３モータ（シンクロナスキャプスタンモータ：１　インダクションモータ：２）

記録方式……ビット　シリアル
フェイズ　エンコード方式

記録密度……1000ビット／インチまたは2000磁束反転／インチ

読み／書き速度…５インチ／秒

早送り／巻戻し速度…120インチ／秒

スタート時間……160ms

停止時間……210ms

スタート距離……0.75インチ

停止距離……0.2インチ

高速よりの停止時間…260ms max.

ＥＯＴ検出……光電検出装置

ジッタ……±1.8%rms／ビット以下

速度変動……±0.2%以下

カセット仕様

テープ仕様……コンピュータグレード
幅：150mil　厚さ：0.7mil　長さ：300feet

テープ・カセット…フィリップス　タイプ（4″×２½″×⅝″）

テープ容量……3.6Mビットまたは360Kキャラクタ　（１キャラクタ８ビット＋２ビットパリティ）

テープ保護……アドレス・トラック，データトラックに各々別に保護が可能である．

環境および寸法重量等

論理回路……シリコンソリッドステート回路
マウント……卓上型
入力電源……AC100V±10%
周波数……50または60Hz切替要
消費電力……約130VA
外形寸法……W205×H215×D605mm
重　量……13.5kg
周囲温度……5°～38°C
湿　度……20～90%

I/O命令一覧

6301　ACCの内容をCMT-100に転送する.
6304　CMT-100のステイタスをACCに読み込む.
6302　フラグ検出
6301　CMT-100にACCの内容（命令）を転送

命　　令	ACCの内容
FORWORD	0414
REWIND	0440
FAST FORWORD	0450
WRITE TRACK B	0470
WRITE TRACK B	0471
READ TRACK A	0460
READ TRACK B	0461
STOP (FORWARD)	0410
STOP (REWIND)	0400

BIT3=0
6304　BIT4～11=DATA
CMT-100→ACCにステイタスを読む.
BIT 3 =1
BIT5=テープスピード不安定
6=キャラクタエラー
7=EOT
8=トラックプロテクト
9=MOTION
10=テープアドレス信号
11=データサービス
6304　CMT-100からACCにデータを転送
BIT3=0

取扱注意事項

1．カセットテープの取扱い方法

(1) 磁気テープはキャビネットに入れ，チリ，ゴミが入らないように保管し，使用するとき以外はキャビネットから出さないこと．

(2) 温度，湿度変化の激しい場所に放置しないこと．

(3) カセット装置から取りはずす場合はできるだけ巻戻すこと．

(4) カセットテープは磁性テープの始端，終端にそれぞれBOT．EOTマーカのさん孔されていないものを使用すること．

(5) カセットテープはセンシング泊のないものを使用すること．

(6) カセットは確実に装置にセットすること．

2．設置条件

(1) 空調設備のある標準の計算機室で作業されることが望ましく，これにより空気中の塵芥による汚れを保護することができる．

(2) 仕様の条件にあった環境で使用すること．

ユーザ側でおこなうべき保守

1．走行部分（プレッシャーローラ，テープヘッド，テープガイド）は日常掃除すること（クリーナはTEAC　TZ-251など）．

2．プラスチックフィルタは環境の良いところで使用するときは6カ月ごとに掃除すること．

3．もし，CMT-100の機械がおかしいと感じたらU CHECKプログラムでチェックしてみること．

本I/Oを使用するにあたっての計算機最小構成

・CEC-555

・CS-100

⑩ CS－1200入出力タイプライタ

概　要

印字機構は、プリントヘッドの位置決め、印字文字の選択、用紙の送りなどすべて電子制御でおこない正確で信頼性が高く、さらに通常のタイプライタのようにみられるメカ部分の部品点数を大幅に削除したことから長期にわたり上質の出力印字が保証されている.

とくにハンマ機構による打字方式文字と異なりプラスチック円盤状に配された文字をえらび静止印字する方式であり、まったく静かで震動もなく30字/秒の高速で印字できることから小型事務処理システム、データ処理システムの入出力カナタイプライタとしては最高の機能を発揮する.

仕　様

印字速度……30字/秒　平均Text

文字の種類……96種（ASCⅡによるアルファニューメリックを標準とする. カナコンビはオプション）

1行の印字数…132コラム/1インチに10文字の場合

用紙幅……15インチ標準

キャリッジリターン時間…400ms/132コラム

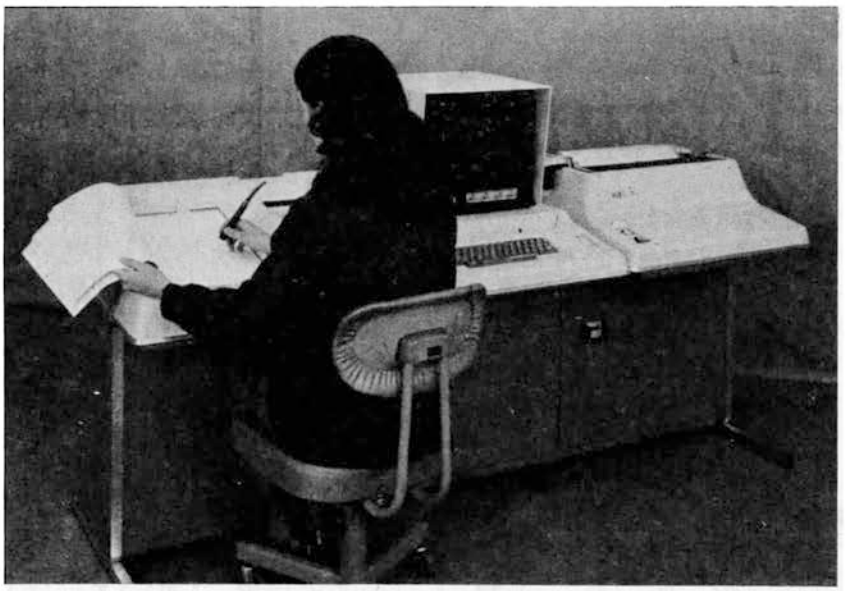
タブレーション…左右へコラムアドレスに対応して直接移動（メカニカルストップではない）

タブレーション時間…400ms/132コラム

微動コラムスペーシング…1/60インチ

ペーパーフィード…上下へアドレスに対応して直接移動





CS-1200接続例

微動ペーパーフィード幅…1/48インチ（マニュアルでは1/2行送り）
ペーパーフィード時間…4インチ/秒＋45m秒の設定遅れ
消費電力……100W平均
プラテン……標準型またはピンフォード型に変換可
補助フィード装置…トラクター型フォームフィード装置取付可
活字部……オペレータによる簡単な交換可
リボン……カートリッジ式
リボンの種類…ナイロン布製またはワンタイム
リボンの色……1色（黒），または赤／黒
入力電源……AC100V±10%
周　波　数……50～60Hz
消費電力……300W平均
外形寸法……W571×H215×D343mm
重　量……13.6kg
周囲温度……0～40°C（保存—50°～60°C）
湿　度……10～90%（NH）

I/O命令一覧

1．キーボード　リード

a）6031，6331：フラグが1の時，次の命令をスキップする．
b）6032，6332：キーボードのフラグとACをクリアする．
c）6034，6334：バッファの内容をAC04～11に読みこむ．
d）6036，6336：b），c）の複合命令で，キーボードフラグとACをクリアし，その後バッファの内容をAC04～11に読みこむ．

2．プリンタ

a）6041，6341：フラグが1の時次の命令をスキップする．

b）6042，6342：プリンタのフラグをクリアする．

c）6044，6344：AC04～11内容をプリンタに印字する．

d）6046，6346：ｂ），とｃ）の複合命令で，プリンタのフラグをクリアし，その後AC04～11の内容をプリンタに印字する．

e）6141，6142，6144：TAB動作をする．

注意事項

1．運搬時に，あまり大きな振動を与えないようにする．

2．プリンタの内部には，むやみに手などを入れないようにする．

3．リボンカートリッジの交換時には，交換後リボンにゆるみがないか確認する．もしゆるみがあれば巻取ノブを矢印の方向に巻き，リボンを常にはった状態にする．

4．ピンフィードプラテン，またはトラック型フォームフィード装置を使用する場合は，PAPER RELEASE LEVERは，RELEASE状態にする．

保持について

定期保守は，500時間使用，または６ケ月のどちらかにいたるごとにおこなう．

使用上の最小構成

CUP（IOC）含む

CS-1200

⓫ CMD8, CMD4キャラクタディスプレイ

概　要

コンピュータシステムの入出力端末として有効に利用できるキャラクタディスプレイ装置である.

構成は、標準17インチのブラウン管およびJIS準拠のキーボードからなり、オペレーション、問い合わせ、処理結果の表示などの業務をスムーズに遂行することができる.

標準として表示色は白黒であるが、オプションとして7色のカラーを表示することも可能である. 表示速度は最高9600ビット／秒であり、オンラインシステムにおける情報のリアルタイム処理に最高の効果を発揮する.

用　途

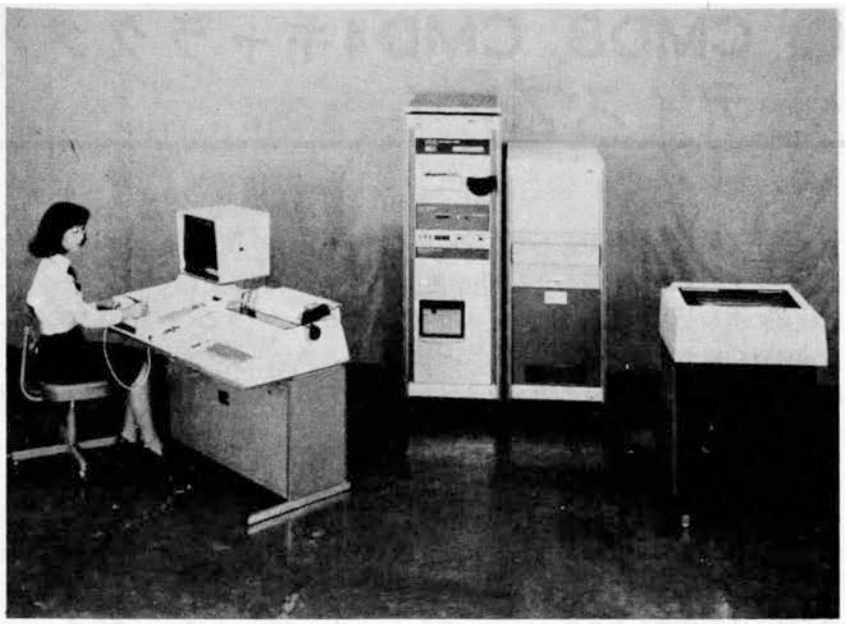
1. 事務処理システムの高速表示
2. 事務処理システム、オンラインデータ処理システムのオペレーションコンソールとして.
3. 各種のモニタリング用端末として.
4. 病院、学校、飛行場などのオンライン情報処理システム表示端末

特　長

1. 高速表示が可能、870字～960字／秒



CMD-4キャラクタディスプレイ



CMD-8接続例

2．価格が低廉
3．白黒またはカラー（CMD－4のみ）が選択できる．
4．JIS準扱いのキーボードが標準として装着されている．
5．ライトペンの使用が可能（オプション）

仕　様

CMD－4

表示式……CRT，CRT選択可能（ITV，PICTURE　MONITER等）
表示文字数……800字（40字×20行）
表示字種……英数字，英記号，64字（SP含む）カナ，カナ記号　64字（SP含む）計128文字
表示色……白黒，またはカラー7色（オプション）
（R，G，B，CY，MG，YEL，WHITE）
映像極性切換……表示色が白黒のとき／白地に黒，黒地に白　} スイッチ切換
表示文字フリッカー…可能（オプション）
文字表示方式……5×7ドットパターン
7×9ドットパターン（オプション）
TV方式……標準方式（カラーはR，G，B）
映像出力信号……白黒　VS1.0V（PP）
R，G，B　0.7V（PP）
SYNC　4.0V（PP）
符　号……JIS－C　6220（8単位）
インターフェース…調歩同期方式（110，150，300，600，1200，2400，4800，9600BPS選択可）
メモリ……・リフレッシュメモリ
12ビット　IK語　半導体

・キャラクタジェネレータ
英数字、英記号（5×7）64字（SP含む）
英数字、英記号（7×9）64字（SP含む）
カナ、カナ記号（5×7）64字（SP含む）
カナ、カナ記号（7×9）64字（SP含む）
（7×9ドットはオプション）
計128文字

キーボード……JIS C 6220

カーソル制御機能…ライト、レフト、アップ、ダウン、ホーム、CR、LF

電　源……100V±10%、50／60Hz

消費電力……約130VA（コントロール、キーボード部）

周囲条件……0°～+40°C

外形寸法……コントロール部
W430×H99×D392㎜
キーボード部
W430×H35×D260㎜

重　量……コントロール部　約12kg
キーボード部　約4kg

CMD－8

表示方式……CRT（標準17インチ）
CRT選択可能

表示文字数……2560字（80字×32行）

表示字種……英数字、英記号　64字（SP含む）
カナ、カナ記号　64字（SP含む）
計128字

表示色……白黒

映像極性切換……表示色が白黒のとき
白地に黒／黒地に白　スイッチ切換

表示文字フリッカー…可能（オプション）

文字表示方式……5×7ドットパターン

TV方式……標準方式

映像出力信号……白黒　VS1.0V（P－P）

符　号……JIS－C－6220（8単位）

インターフェース…調歩同期方式（110、150、300、600、1200、2400、4800、9600BPS選択可）

メモリ……・リフレッシュメモリ
8ビット　3K語　半導体
・キャラクタジェネレータ英数字、英記号（5×7）64字（SP含む）
カナ、カナ記号（5×7）64字（SP含む）
計128字

キーボード……JIS－C－6220

カーソル制御機能…ライト、レフト、アップ、ダウン、ホーム、CR、LF

電　源……100V±10%　50／60Hz

消費電力……約130VA（コントロール、キーボード部）
周囲条件……0°～＋40°C
外形寸法……コントロール部
W430×H99×D392㎜
キーボード部
W430×H35×D260㎜
重　量……コントロール部　約12kg
キーボード部　約4kg

⑫ CS－408高速紙テープリーダ

概　要

CS-408高速紙テープリーダは電子計算機システムおよび自動制御装置などの入力装置として使用される光電式せん孔テープ高速読取り装置で，毎秒300字の速度でせん孔テープを読取れる．

本装置はテープの駆動，停止機構の安定化を図り，電気回路はIC化されている．

特　長

1．読取り機に必要な電源はすべて内蔵している．
2．ランプの自動光量制御をしているから劣化による光量の減少などの現象はない．
3．読取りアンプにはIC化コンパレータを使用しているからリードミスがなく安定している．
4．内蔵安定化直流電源には，IC化レギュレータを使用しているから信頼度が大きく安定している．
5．キャプスタンローラには耐磨耗性に秀れた特殊２層ゴムを使用しているので摩擦力が大きく，駆動力が安定しており長期間の使用に耐える．
6．フライホィールおよびベルト等の機構には独特の技術が活かされているので，安定かつ長期の使用に耐える．

仕　様

読取り速度……連続読み：300字／秒
　　インクリメント：100字／秒以上
使用テープ……８単位JIS規格品（JIS　C　6243）
　　せん孔テープ
　　各色テープ読取り可能
読取り方式……シリコン光電素子による光電読取り



テープエンド検出…テープエンド検出スイッチでテープの終端を検出する.

入力電源……AC100V ±10%

周波数……50または60Hz切替要

消費電力……200VA以下

外形寸法……W248×H248×D367㎜

重　量……約14kg

周囲温度……0°～50°C

I/O命令一覧

6014　フラグとバッファをクリアしリーダから1語読み，バッファに入れフラグをセットする.

6012　バッファの内容を読みとりフラグをクリアする.

6011　フラグが1のとき次の命令をスキップする.

取扱注意事項

・ほこりの多い場所は避ける.

ユーザー側でおこなうべき保守

・使用日には最初にセンサ部のゴミをよくはらう.

本I/Oを使用するにあたっての計算機最小構成

・CPU

・IOC



ラック実装

⑬ CS－420高速紙テープパンチャ

概　要

CS-420高速紙テープパンチャはCEC555シリーズの高速のデータテープ作成に使用される高速紙テープせん孔機で、パンチブロック、せん孔選択機構、テープフィード機構から構成されている.

パンチ機構およびフィード機構は独立してマグネットの磁力で動作し、すべてのテープ幅、厚みを自動調整し、高速でパンチする.

また、ノズルレベルが低く、テープ供給巻取りリールは上部に内蔵され保守が大変容易である.

仕　様

作動速度……最高75キャラクタ／秒
テープ幅……8トラック（25.4mm±0.1mm）
テープの厚み……0.08～0.11㎜
テープの種類……紙テープ（ISOスタンダード）
テープの供給……約120,000キャラクタで300m
マークキャラクタ…リクエストによりフィードホール付またはなしで設定することができる.
送り精度……隣りのキャラクタで±3%
10キャラクタで±1%
50キャラクタで±0.5%
（ISOスタンダード）
入力電源……AC100V±10%
周波数……50～100Hz
消費電力……100W
外形寸法……W220×H190×D432㎜
重　量……13.5kg



周囲温度……5°～40°C

湿　度……20～80%

I/O命令一覧

6024　　1語パンチしてフラグをセットする.

6022　　フラグとバッファの内容をクリアする.

6011　　フラグが1のとき次の命令をスキップする.

取扱注意事項

・10巻程度で内部のゴミをきれいに掃除する.

本I/Oを使用するにあたっての計算機最小構成

・CPU

・IOC



⑭ CS－100入出力タイプライタ

概　要

CS—100入出力タイプライタはCEC555シリーズ用の入出力タイプライタである．CS—100は，キーボードの手動操作，あるいは，せん孔テープを読み取って，情報を送信する装置で，テープ・リーダ，テープ・パンチ機能を備えており，鍵盤によるローカル通信の際，テープせん孔と同時に，コピーができる．また，せん孔されたテープ送信のとき，コピーとせん孔テープの複数が同時におこなえる．

仕　様

単位数……8単位（ASCIIコード）
速　度……110ボー（600字／分
入力電源……AC100V　I 10%
周波数……50/60Hz
消費電力……200VA
外形寸法……560W×213H×470Dmm



鍵盤部

キー数……63字（数字、英字、記号）

ファンクションキー…9

印字部

印字方式……タイプホイール方式

行間隔……4.23㎜

印字間隔……2.54㎜

1行最大印字数…69字

用紙規格……幅213㎜，最大巻径127㎜のロール紙

リーダ部

テープ規格……幅25.4㎜

テープ・タイト…検出可能

テープアウト……検出可能

せん孔部

テープ規格……幅25.4㎜

せん孔規格……符号孔1.83ϕ

フィード孔1.18ϕ

I／O命令一覧

キーボードリーダ

a）KSF　6031：フラグが1の時次の命令をスキップする.

b）KCO　6032：ACとフラグをクリアする.

c）KRS　6034：バッファの内容をAC4～11に読みこむ.

d）KRB　6036：KCCとKRSの複合命令

プリンタ・パンチャ

a）TSF　6041：フラグが1の時次の命令をスキップする.

b）TCF　6042：フラグをクリアする.

c）TPC　6044：AC4-11の内容をプリントまたはせん孔する.

d）TLS　6046：TCFとTPCの複合命令

本I/Oを使用するにあたっての最小構成

CPU（4K語）

CS-100

取扱注意事項

1．受信紙の設置方向に注意すること.

2．リボンは印字鮮明度が落ちた場合と、リボンの縁がいたんで糸くずが出た時は直ちに交換すること.

3．横1行印字文字数は72文字であり、60～61文字目でエンドオブラインベルが鳴ったあとは1行に打ち過ぎないように注意する.

4．電源SWがOFFのとき、キーボードのキーを打つと、その後電源をONにしたとき不必要な文字を印字するので注意する.

5．テープリーダが動作中は、キーボードのキーを打たないこと、誤って打けんすると誤動作する.

ユーザ側での保守

1．3ケ月に1回、各機構のフェルト部に注油する．特に、タイプホイル軸のフェルトとモータカバー両側の注油孔を忘れないこと．使用オイルは「パンモータオイル10W—30」が良い.

2．注油のための本体カバーの脱着は次の通りである.

1) 印字用紙，せん孔紙を取り外す.
2) プラテンノブを外す.
3) 3Wノブを外す.
4) 前面の化粧板を手前に引きながら下げて外す.
5) 化粧板で隠れていたネジを4本外す.
6) 背面の化粧板ネジを3本外す.
7) 左側面のリーダカバー取付ネジ1本外す.
8) カバーを静かに引き上げることにより，取り外すことができる.
9) 取付けのときは，各所の組合せ部分が正しく組合されたかどうか注意してネジをしめる.

3．日常の清掃と点検

1) カバーウィンドキートップ，キーボード
汚れのひどい時は，石鹸水をしたした布で取ること.
※シンナー，ベンジン等は使用しないこと.
2) プラテンとその周辺
プラテン周辺の紙くず，ゴム等は取除き，プラテンの汚れのひどい時は石鹸水でふく.
3) キャリッジ周辺
キャリッジ周辺のインクリボンのきれはし，ゴム等を取り除き，インクの付着した所は石鹸水でふく.
4) 受信紙，せん孔紙，せん孔くず，インクリボンなど
受信紙，せん孔紙の有無を確認し，チャドボックス内のせん孔くずを拾てる．印字の文字が薄い場合インクリボンを交換する.
5) リーダ部
テープリッドを用いてトッププレートの溝の付近の紙くず，ゴミを取り除く.
6) タイプホイル
活字面のインクリボンのゴミを硬いブラシで掃除する.

⑮ CS－1000入出力タイプライタ

概　要

ＣＳ－1000入出力タイプライタはCEC555シリーズの英・カナ文字入出力タイプライタで，事務用伝票を打鍵操作によって印字，作成するのと同時に，紙テープまたはエッジカードにせん孔したり，あるいはエッジカードを読取って自動的に印字作表をおこなう．

仕　様

単位数……８単位，　６単位
コード……ASC Ⅱ コード
印字速度……1000字／分
読取速度……1000字／分
せん孔速度……1000字／分
電源……AC100V　±10%
　　50または60Hz
入力電源……AC100V±10%
周波数……50／60Hz
消費電力……350VA
外形寸法……900W×850H×580Dmm

鍵盤部

キー・配列……ISO形（英文・カナ文字両用）





キー・数……64個

印字部

印字方式……タイプホイール打字方式

行間隔……4.23mm

タイプリボン……黒

プラテン……バリアブルスプロケット

1行最大印字数…最大132字

用紙幅……最大381mm

複写枚数……5枚

活字収容数……最大132字

リーダ部

読取チェック……パリティ・チェック

テープ・アウト…検出可能

エッジカード……読取可能

せん孔部

エッジカード…せん孔可能

I/O命令一覧

CS-100に準ずる.

取扱注意事項

1．移動の時にプラテンノブを持たないこと

2．電源OFFの時キーボード操作をしないこと

3．タブストップした時は復帰以外のキーを押さないこと.

4．リーダおよびパンチを使用しない時，おのおのSWをOFFにしておくこと（SWのないときはコネクタを外す）

5．プラテンに印字用紙がセットされていない時には印字をしないこと.

6．チャドボックスのパンチくずは早めに捨てること，くずのガイド筒にくずがつまると，パンチピンを損傷することがある．

7．印字用紙は背面板の棚の上に乗せること．

8．インクリボンは印字鮮度が落ちた時およびリボンの縁から糸くずが出てきた場合は交換すること．

9．印字用紙で両側にスプロケット孔のある用紙はスプロケットガイドにセットし，ペーパーリリーズレバーを閉にする．

10.CS-1000（PTS-1000）の取扱説明書に記載のせん孔のセット方法のうち，せん孔のルートを説明書の逆に（反時計方向）　にセットするとCS-100と同じ面にせん孔される．

ユーザ側での保守

1．メンティナンス契約外の場合は，3ケ月に1回各機構のフェルト部分に注油する．使用オイルは「パンモータオイル10W-30」がよい．フェルト以外の注油は1年1回サービスマンに依頼しておこなう．注油のためのカバーの取はずしは「PTS-1000取扱説明書」を参照のこと．

2．オイル不足を防ぐため，不必要な時は，SWをOFFにすること．

3．終業時等は，防塵カバーを被せること

4．パワーオンができない時は，本体裏側のヒューズを点検し，断線時は交換する．

⑯ CLP−4100ラインプリンタ

概　要

CLP-4100ラインプリンタはCEC555シリーズ用として設計されたコンパクトタイプの小型ラインプリンタである.

機構的には印字機構に活字ドラムを使用し，インパクトハンマ，トリガ電磁石などがモジュール化されているのでメンテナンスが容易である.

ハンマ機構は128桁ASCII64文字を毎分300行のスピードで印字する.

制御回路はMSIを活用し，カナ文字の使用，リボン反転装置など中型クラスのプリンタ機能を備えている.

機種は64文字ASCII用CLP-4100と96文字カナ入り用CLP-4100Aの二種がある.

仕　様

印字型式……回転活字ドラム
印字速度……300行1／分（ASCII64文字ドラム）CLP-4100
　　220行1／分（英，数字，カナ，96文字ドラム）CLP-4100A
文字数……64／96文字（カナ文字）
文字桁……132桁／行
文字型と寸法……ゴシック文字
　　2.5（高）×1.7（幅）mm
ハンマ……オン・サプライ方式4行／1モジュール
紙送り時間……連続送り60行／秒（10インチ／秒），1行送り





紙送り型式……ピンタイプ・スプロケット電磁クラッチ・ブレーキ使用
行間数……6行／インチ（4.23±0.2mm）
用紙寸法……幅3インチ（76.2mm）～15インチ（381.0mm）
　　長さ任意（フォーマットテープ使用）
複写枚数……最大5部
フォーマット・コントロール……光電式フォーマット・コントロール
インク・リボン…357mm（幅）×13.5m（長）自動反転指定リボン（メック4100リボン）製造元
　　（富士化学紙（株））

外部条件

入力電源……AC100V±10%
周波数……50／60Hz
消費電力……350VA
外形寸法……660W×930H×5100m／m
周囲温度……5～40℃
湿度……40～90%

I／O命令一覧

a) 6051：フラグが1のとき次の命令をスキップする.
b) 6054：アキュムレータの内容をLPに転送する.
c) 6052：フラグをクリアする.

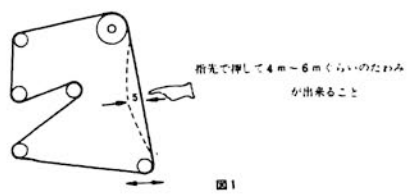


図1

取扱注意事項

1. 上蓋の開閉は静かにおこなうこと.
2. ケース前板の開閉は上蓋を開けてからおこなう.
3. 印字中，リボンが左または右に片寄っていたら機械を一旦停止させ，リボンを巻き直すこと.
4. フォーマットテープの装着は張り過ぎないこと（図1参照）.
5. 用紙，インクリボンを装着しない状態で印字しないこと.
6. 筐体グランドは必ず設置すること.
7. 用紙またはインクリボンを交換するため，ドラムユニットを手前に開くときは，必ずケース前板を手前に用いてからおこなうこと.
8. ドラムユニットは静かに開閉すること.
9. EMEDGランプが点灯している時の印字は絶対にさけること.

10. ホチキス等で止めてあるコピー用紙を使用するのはやめること.
11. 印字された用紙は背面の紙受金具の上を通し，紙の習性を利用して自動的に，キチンとたためるようにすること．これを怠ると用紙送り不良となる場合がある.
12. 納入時にセットされているドラムユニット固定金具は必ず保存しておき，オーバーホール等のため輸送する時に使用する.
13. 本体を移動するときは，指定場所を持つこと．背面板把手，前とびら等に手をかけてはならない.

ユーザ側でおこなう保守

1．活字ドラムの清掃

月1回，活字ドラムを清掃する.

歯ブラシで清掃液（トリクレン，トリオール，またはフレオンシンナ）をかけながら，活字間につまった汚れを落し，すべての文字がきれいになるようにする.

〔注〕金属ブラシは使用しない

リボン反転検出器にキズをつけぬこと.

2．クラッチブレーキの磨耗粉の除去

月1回，必ずリセットクラッチブレーキと，紙送りクラッチブレーキの磨耗粉を真空掃除機を使って除去する.

1) ケース上蓋をはずす.
2) 背右上部に左右各1ケの空色のモータ状のものがクラッチブレーキであり，左が紙送り用，右がリセット用
3) 各々のクラッチブレーキのビニールパイプの先端の栓を外し，電気掃除機で磨耗粉を吸い取る．この時プーリーを手で廻しながら吸い取るとよい.

3．静電除去器の清掃

静電除去器の針先にカーボンがたまっていたらガーゼか歯ブラシで清掃する．針の先端が丸くなって放電が弱くなったものは，針を交換するか，静電除去器を新しいものと交換する.

4．露出部分の清掃

本体についているゴミ，鉄粉，紙くず，ホコリなどを掃除器で吸い取る.

《点検箇所》

1) 紙送りユニット周辺の紙くず.
2) 入力電源（特にクラッチブレーキ，モータの下）のベルトくず，およびクラッチブレーキの鉄粉
3) 入力電源のファンのホコリ
4) フォーマットユニットの検出部分の紙くず，ホコリ

《清掃上の注意》

1) エアーガン等は使用しないこと.
2) ケース上蓋の清掃は，フレオンはピッチクリーナ等を用い，アルコールシンナ等の強い溶剤は使用しないこと.

⑰ CCD−5Sカラーキャラクタディスプレイ

概　要

本装置は，CEC555シリーズ用カラーキャラクタディスプレイでコンピュータシステム，データ通信システムの端末として効果的な装置である．

装置の特長は，音が全くない．1文字8×10ポイントで画像がきわめて鮮明である．表示速度が高速である．7色のカラーに色別表示できることなどで，制御用システムのモニタから，データ分析，図柄分析，医用情報システム，公害データ監視システムなど，広い分野で活用できる．





また，受像機は4台まで分岐表示が可能でサイズも大型，小型の選択は自由である．

特　長

1．表示端末の増設が簡単である．ＲＦ出力を使用し市販のカラー（白黒でも可）受像機のアンテナ入力端子にケーブル1本を接続するだけである．
2．ライトペン（オプション）の使用により表示文字の修正が容易である（ＣＣＤ－５型）．
3．データの受信速度が高速であるため1画面表示するのに約50msの速さである．

仕　様

表示端末……18インチカラーブラウン管
表示方式……NTSC方式に準ずる．
　　カラー受像機用出力
　　R.G.B出力
　　RF出力
　　NTSCカラー映像信号出力等が可能．従って表示端は市販のカラー（または白黒でも可）受像機でよい．さらにRF出力を用いケーブル1本で数台の表示端末へ接続することができる．
表示文字……JIS準拠
　　アルファベットおよび数字…36
　　カナ文字……55
　　特殊記号……35
　　合計126字
表示色……7色（R，G，CY，MG，YEL，WHITE）
表示記憶文字数…1画面512文字（32字×16行）を2画面・合計1024字
文字コード……ＡＳＣ II コード（8単位）
文字発生方式……12ビット1Ｋ語の内蔵半導体メモリ，リードオンメモリ（126語：1語8×10ドット）
キーボード……ホール素子
　　コントロールキーを含め71キー
ライトペン……使用可能（オプション）

⑱ CCD－5Gカラーグラフィック ディスプレイ装置

概　要

ＣＣＤ　５Ｇカラーグラフィックディスプレイ装置は，コンピュータからの出力情報のカラーグラフック表示はもちろんのこと，オフライン時に任意にブラウン管表面に描かれた図形，模様などをコンピュータに入力することもできる表示端末装置である．

従って，コンピュータによって計算された図形，デザインなどの表示を人間の目視による修整を加えて逆にコンピュータに入力することができ，複雑な図形処理，あるいは図形を情報としてコンピュータに入力することを必要とする分野において極めて効果的に利用できる．

表示色は，赤，青，黄，マゼンダ，シアン，白に黒を加えた８色で，これを操作パネルの24個の色調整ダイヤルにより任意の色調に調整することができる．表示は320×224のポイントにより像を構成し，これを９個のカーソル制御キーとジョイステイックコントロールキーにより像の調整をおこなう方式をとっているのでライトペンに比べて正確度が高く，能率的に正確に調整することができる．用途としては，各種織物のデザイン，修整，建築，車輌，船舶，航空機等のデザイン，構造設計，放送，医学，実験，試験データの解析表示，各種モニタシステムの表示端末などの広範囲の分野で活用することができる．

特　長

１．24個の色調整器により，任意の色調に調整することができる．
２．インターフェースを内蔵している．



4800BPS……スタンダード

2400、1200、600、300、150、110BPS等が可能である.

調歩式、または同期式.

3．オフラインで、ジョイステイックカーソル制御キーを使用して、自由に図形を描くことができる.

4．同一システム内で複数台の装置を制御運転することができる.

5．ライトペンの代りにカーソル制御キーを使用しているので、微妙な図形の修整が正確におこなえる.

6．価格も低廉で、信頼も高く、CEC555シリーズの他、あらゆるシステムで活用できる.

仕　様

表示部

表示装置

表示方式……水平……15.75KHz

垂直……60Hz

走査線……256本

インタレース１：１

毎秒画数……60枚／秒

解像度……水平

垂直

グラフイックコントローラ

表示色……７色（赤、緑、青、マゼンダ、シアン、白）

表示画素数……320×224

リフレッシュメモリ…210K　BITS（320×224×3）

カーソル……十字カーソル

ディスプレイモード

イ）ポイントプロット……X、Yアドレス、色指定

ロ）インクリメンタルプロット…８方向（色、方向指定）

ハ）ラスタスキャンニング

データ送信モード

イ）カーソル…十字カーソル交点の色、アドレスを出力する.

ロ）グラフィックデータ…画面左上より順次色情報を出力する.

操作パネル

LINE／LOCAL…切換キー

色指定キー……８キー　７色＋黒（消去）

カーソル制御キー…９キー　８方向＋「HOME」HOME以外はリピート機能あり.

マーカーON／OFFキー

クリヤキー……全画面のクリヤ

データ送信キー…２キー　カーソル、グラフイックデータ

色調整器……８×３個しぼり切り色消去

ジョイステイックコントロール…オプション

⑲ CCD−5GH高速カラーグラフィックディスプレイ

概　要

CCD-5GHカラーグラフィックディスプレイ装置はコンピュータからの出力情報のカラーグラフィック表示はもちろん，オフライン時に任意にブラウン管面に描かれた図形，模様などをコンピュータに入力することもできる表示端末装置である．

したがって，コンピュータによって計算処理された図形，デザインなどの表示を人間の目視による修整を加えて逆にコンピュータに入力することができ，複雑な図形処理や図形を情報としてコンピュータに入力することが必要な分野において極めて効果的に利用できる．

表示色は赤，青，黄，マゼンタ，シアン，白に黒を加えた８色で，これを操作パネル上の24個の色調整ダイヤルにより任意の色調に調整することができる．表示は320×240のポイントにより像を構成し，これを９個のカーソル制御キーとライトペンにより像の修整を能率的に正確に修整することができる．用途としては，各種識物のデザイン，修整，建築，車両，船舶，航空機等のデザイン，構造設計，放送，医学，実験，試験データの解析表示，各種モニタシステムの表示端末など，広範囲の分野での活用に適合できる．

特　長

1．24個の色調整器により，任意の色調に調整することができる．
2．オフラインで，カーソル制御キーを使用して自由に図形を描くことができる．
3．同一システム内で複数台の装置を制御運転できる．



4．ライトペンの代りにカーソル制御キーを使用して微妙な図形の修整が正確におこなえる．
5．価格も低廉で，信頼度も高くCEC555シリーズの他，あらゆるシステムに活用できる．
6．標準TV方式を採用し，表示部の増設が簡単にできる（RGBモニタ）．
7．表示された情報を計算機に入力することもできる．

仕　様

表示部

20インチRGBモニタ	水平	15.73426KHz
	垂直	60Hz
	走査線	525本
	水平解像度	350本以上

グラフィックコントローラ

表示方式	標準TV方式
表示色	8色（任意の8色）
表示画素数	320×240
リフレッシュメモリ	240K　BITS（320×256×3）

ディスプレイ

1）ポイントプロット　X，Yアドレス，色指定
2）インクリメンタルプロット　方向，色指定
3）カラープロット　色指定

データ出力

1）X，Yアドレス
2）ライトペン座標出力（ライトペン：オプション）
3）ディスプレイデータ出力

キーボード

38キー（表示付）｛方向キー／色キー／その他

その他

1．FULL　WHITE信号
　格子じま，ドット信号を内蔵しており，格子ドットは表面にスーパーすることもできる．
2．各色の設定状態を一目で確認できる基準色を表示できる．

⑳ C－1525デジタルX-Yプロッタ

概　要

C1525ＸＹプロッターは計算機の演算結果をＸ，Ｙ座標に作図することができる．

用　途

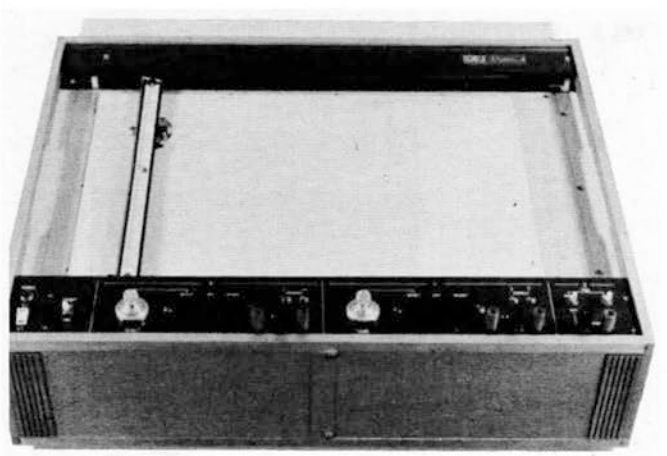
1．統計，管理の図表作成
2．測量計算における作図
3．NC装置用テープのチェック
4．各種技術計算の図形処理
5．各種データの解析
6．各種測定における特性曲線，分布図，状態図の自動作画

特　長

1．水平，垂直，傾斜のいずれの位置にても使用可能である．
2．600パルス／scc．ステップ幅0.1mmの高速で動作する．
3．任意の位置に零点設定が容易である．
4．リセットもフルスケール１秒以内で高速で原点に復帰できる．
5．オーバースケール保護回路によりスケールアウトによるレコーダの損傷がない．またスケールアウトの信号ができる．
6．Ｘ，Ｙ，Ｚ軸，入力端子のほかにインターフェースからの接続用のコネクタが付いている．
7．装置の大きさはAO判から希望により選択できる．

仕　様

記録紙固定方式…静電吸着方式
記録紙寸法……有効記録幅　Ｘ軸400mm
　　Ｙ軸280mm



Ａ３判

(JIS A 3判)

ステップ幅……0.1mm／1パルス．1～2相交互励磁方式

プロット速度……600パルス／sec（X，Y軸共通）

パルス……X軸4,000パルス／フルスケール

Y軸2,800パルス／フルスケール

入力パルス……（X，Y軸共通）

高レベル　4～6V（正パルスTTLレベル）

低レベル　0～0.2V

パルス幅　50μsec以上

立上り時間　10μsec以下

パルス入力抵抗　10KΩ

ペン……インク記録方式

Z軸　プロット速度約0.2sec

ペンのup-down：手動切換可能

自動の場合　高レベル＋4～＋6 Vdown　低レベル0～＋0.2Vup

いずれも入力抵抗10KΩ

零点位置……任意の位置に容易に設定可能

リセット……手動ボタン，または外部信号でフルスケール1秒内

DCサーボ方式，平衡回路により原点に復帰可能

オーバ　スケール…オーバ　スケール保護回路付

保護リミット回路…スケールアウトリミット信号出力：±X，±Y　高レベル：＋5V

入力電源……AC100V±10%

周波数……50～60Hz

消費電力……約70VA

外形寸法……W554×H470×D150mm

（A 3判）

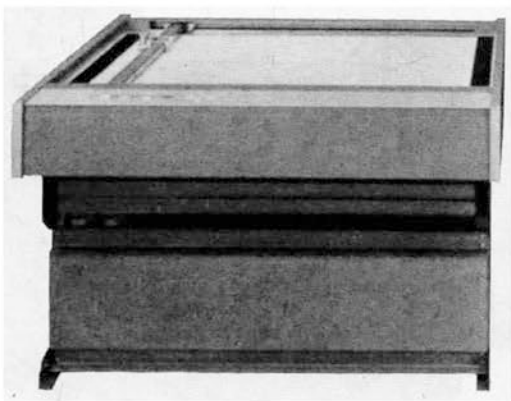
重　量……15kg

周囲温度……0～40℃

本I／Oを使用するにあたっての計算機最小構成

・555・IOC　UNIT

・CS—100（または高速紙テープ，外部記憶装置等）



AO判

㉑ CS－60カードリーダ

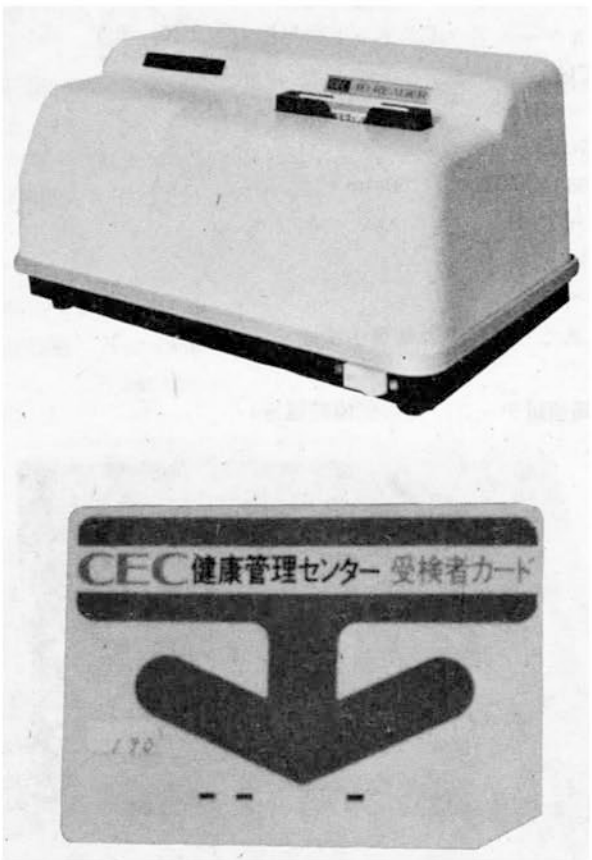
概　要

CS—60カードリーダは指定されたカードをリーダに挿入して多数のデータ入力時のセレクトに使用される装置で，個々のデータにカードの番号を付加し電子計算機に入力する．

この装置によりリンクできるデータ数は発行されるカードの数によって最大800まで自由に変更することができる．またCS—50キーボードユニットと接続使用することができる．

仕　様

コード……12ビット（1パリティビット）
　　BCD 3 桁（ただし最大799）
所要電力……AC100 V ±10 V　50／60Hz　単相　約50VA.



㉒ CS－50, CS－80キーボード入力装置

概　要

CS—50　キーボード入力装置は，コンピュータとオンラインで接続できない測定器，検査機器，試験器などの入力装置として利用する入力端末装置である．

装置はキーボード部の項目キー，数値キー，コントロールキー，アラーム表示およびパネル部の入力確認用のディスプレイから構成され，小型・軽量であり，項目キーに入力項目を設定することによりオンラインで入力できないあらゆるデータ発生個所での入力装置として使用することができる．この他にCS—80が用意されている．

仕　様

項目キー数……10個（指定による）
数字キー数……10個
記号キー……＋－・スペース
コントロール……読み込み，訂正，警報解除
ディスプレイ……プラズマディスプレイ16桁×１行（CS—50），32桁×８行（CS—80）
警報表示……あり
所要電力……AC100V±10V，50／60Hz　単相　約100VA



ソフトウェア編

ソフトウェア編目次

① ソフトウェア概要…… 307
② 紙テープシステム…… 315
§1 ローダ…… 315
§2 エディタ…… 335
§3 アセンブラ…… 343
§4 オクタルデッバガ…… 395
§5 ユーティリティ…… 407
§6 基本保守診断…… 411
§7 その他…… 413
③ マークIIモニタシステム〈その1〉…… 415
④ マークIIモニタシステム〈その2〉…… 463
⑤ 演算インタープリータ…… 465
⑥ ココル（対話計算）…… 473
⑦ ベーシック…… 509
⑧ RPG(Report Program Generator)…… 515
⑨ 4Kフォートラン…… 547
⑩ マークIIフォートラン…… 579
⑪ コボル…… 627
⑫ リアルタイムモニタ…… 677

❶ ソフトウェア概要

1. 概　要

CEC555用のシステムプログラムは，大別して補助記憶を含まない構成で使用できる紙テープベースのシステムと，磁気ドラム，カートリッジ磁気ディスク，固定ヘッド磁気ディスク，フロッピー磁気ディスクなどの補助記憶を含む構成で利用できるモニタベースのシステムとからなる．

一般にモニタベースのシステムプログラムは紙テープベースの対応するシステムプログラムのスーパセットとなっており，２つのシステム間における融通性は非常に大きくなっている．特にモニタベースのシステムにおけるコア常駐部分は僅少で，紙テープベースにおけるバイナリローダより大きくないので，紙テープベースのプログラムはモニタベースのシステムでも常に実行可能である．

紙テープベースのシステムにおけるバイナリプログラム，モニタベースのシステムにおけるすべての紙テープにダンプされたプログラムは，必ず固有の標識がつき，これらを使用する際に操作誤りが絶対に起こらないようにしてあるのは，世界に先がけてCEC555のシステムのみがもつ特長である．

その他，あらゆるプログラムはミニコンの10数年の歴史的な発展をふまえ，その成果にさらに当社独自の経験を加えたものであり，あらゆる要求に対して期待を裏切らない実力を発揮する．

2. 紙テープ用システムプログラム

2—1　ローダ（システムプログラム①ローダ説明書参照）

ベリファイローダ（Vローダ）は128ステップでアセンブラの出力オブジェクト紙テープをコアにロードするために用意されている．

従来の紙テープローダと異り，紙テープのラベルを入出力タイプライタに印刷するので，不注意による紙テープの取違えなどを容易に防止できる．またチェックサムも入出力タイプライタに印刷するので，ラベル印刷と共に紙テープロード時の記録として有用である．

また，スイッチレジスタの選択により，ロードする内容の比較をおこなうベリファイ機能が使用でき，これはプログラムデバッグのための強力な武器の一つである．入力機器は入出力タイプライタ，または紙テープリーダを自動選択できる．オートスタートローダ（Sローダ）ではプログラムロード後自動的にオブジェクトプログラムの実行を開始することが可能であり，入出力タイプライタのないシステムでも使用できる．

これらのローダをコアにロードするためには，ノンリエントラントで９ステップ（入出力タイプライタ用では10ステップ）のXRIMローダを用いる．XRIMローダはスイッチレジスタを用いてコアにロードする．

2—2　エディタ（システムプログラム②エディタ説明書参照）

紙テープベースのコンピュータの基本プログラムの一つで，ソースプログラムテープを，作成・修正したり，コピーしたりするのに用いる．入出力機器のコマンドによる選択，フォーマットコントロールのオプションを始めとして，きわめて応用性に富み，各種のコマンドによって，ソーステープの修正が最少の労力で効果的におこなうことができる．操作はすべてキーボードオリエントになっており，コンピュータに手を触れることなく，各種の操作が可能である．

2—3　アセンブラ（システムプログラム③アセンブラ説明書参照）

SAP-Ⅱ，SAP-Ⅲの２つのバージョンがあり，SAP-Ⅱは４K語のメモリ用，SAP-Ⅲは８K語の

メモリ用のアセンブラである．入出力機器は，最少構成では入出力タイプライタのみでよいが，これに高速紙テープ読取機，高速紙テープせん孔機，高速印刷機などの接続使用も可能である．SAPは2パスアセンブラで，パス1でメモリアドレスの割当てをおこない，パス2でバイナリテープのせん孔およびリスティングの印刷をおこなう．操作はすべてキーボードオリエントになっており，コンピュータに手を触れることなく操作が可能である．SAPでは通常のアセンブラ機能の他に次のような広範囲のコントロールオプションが使用できる．

DECIMAL	数の10進変換指定
OCTAL	数の8進変換指定
*	オリジン指定
PAGE	オリジンを次のページのトップに指定
#	ラベル指定
$	アセンブル指定およびエントリ指定
FIELD	ロードするフィールドの指定
FIXTAB	シンボルテーブルの固定指定
I	間接アドレス指定
TEXT	ASC II 文字の6ビットパック指定
"	ASC II 文字の8ビット値指定
=	シンボル値指定
( ) [ ]	定数表の自動指定

ページ外のアドレス指定について間接アドレス指定ベクトルの自動設定をおこなう．

これらの機能によりソース言語で書かれたライブラリルーチンの自動的リンクが可能である．

2—4　オクタルデバッギングエード（COD，XOD）（システムプログラム④オクタルデバッガ説明書参照）

CODは，384ステップのコンパクトなプログラムであるが，コアの任意アドレスの内容の読出し，修正，書込みを始めとして，ある範囲のコアについて任意ビットパタン（1～12ビット）のサーチなどが可能である．またコア上の任意位置にブレークポイントとなづけるトラップを設け，プログラムがこのトラップを指定回数だけ通過したらCODに戻るように指定できるので，ユーザのプログラムを一部分ずつ実行させてオンラインデバッグを有効に実行できる．修正したプログラムはローダで読める形でテープにせん孔できる．

CODは，CODをロードしたフィールド（コアの4K語ごとの単位）内のみに有効なのに対して，XODは任意フィールドに対して有効であるので，4K語を超えるプログラムのデバッグに有力である．また，CODはインタラプトをオフとするのに対し，XODはブレークポイントの前後においてTTYフラッグおよびインタラプトの状態を保存するのでインタラプトを使用するプログラムのデバッグに威力を発揮する．サーチ機能はCODのサーチ機能に加え逆サーチ（ビットパタンの指定に合わないものを印刷）が可能となっている．また，ある連続した領域に同一データを格納する機能が加えられた．プログラムの大きさは640語である．

2—5　対話計算（COCOL）

一種のオンラインフォートランであるが，その機能は驚異的に高く，しかもデバッグや修正が極めて容易な特長を持っている．COCOL言語の基本機能を修得するには1～3時間の講義と3～6時間の実習で足りる．4K語のメモリと入出力タイプライタだけの最少構成で使用可能であるが，8K語のメモリ，グラフィックディスプレイ，XYプロッタなどを使用するための各種のユーティリティが用意されている．

2—6　4Kフォートラン

コンパイラとオペレーティングシステムの2つのプログラムからなる．1パスフォートランである．コンパイラはソーステープをオブジェクトテープに変換しオペレーティングシステムはオブジェクトテープを読みこんで実行をおこなう．各種のライブラリルーチンはオペレーティングシステムに内蔵されている．フォートラン言語の仕様はJIS3000を若干下回るが，4K語のメモリと入出力タイプライタだけの最少構成で使用可能である．

2—7　マークⅡフォートラン

コンパイラ，リロケータブルアセンブラ，リロケータブルローダの3つのプログラムからなる本格的なコンパイラで8K語以上のメモリで使用可能である．ライブラリサブルーチンやユーザのサブルーチンはフォートランのみでなく，アセンブラを利用して自由なプログラムを書くことが可能である．リロケータブルローダは別々にコンパイルおよびアセンブルした多数のプログラムをロードしてリンクすることができる．JIS3000フォートランをかなり上回る機能を持ち，特に紙テープ用としての使いやすさに細かい配慮がなされている．

モニタベースのマークⅡフォートランと完全な互換性がある．

3．マークⅡモニタシステムプログラム

3—1　マークⅡモニタ

モニタは256語（0フィールドおよび1フィールドの最終ページ）の常駐部分と必要に応じて補助記憶からコアに呼びだされる非常駐部分とからなる．非常駐部分がコアに呼び出される時には，コアのその部分は補助記憶に保存され，モニタの動作後自動的に回復されるので，ユーザプログラムの領域に対する制限はほとんどない．各種の補助記憶に格納される，各種データファイル，ソースファイル，オブジェクトファイルなどはすべて論理名で扱うことができるので，ユーザおよびユーザプログラムは補助記憶やその他の入出力機器の形態や，その物理的特性にわずらわされずに利用可能である．また，機器の論理名はユーザプログラムやシステムプログラムの実行時に自由に指定することができるので，プログラム開発は，入出力機器，周辺機器，補助記憶装置などに無関係に進めることが可能である．モニタレパートリとしてはプログラムの補助記憶への保存，コアへのロード，実行，修正，動的デバッグ，入出力機器の割当てなどが用意されている．モニタユーティリティとしてファイル管理を行うための強力なサブルーチンUSR，キーボードコマンドを解析するコマンドデコーダQCDが用意され，これらはユーザプログラムから容易に利用できる．

マークⅡモニタのシステム記憶としては固定ヘッド磁気ディスク，カートリッジ磁気ディスク，フロッピー磁気ディスクなどがいずれでも使用でき，その場合のシステムプログラム，アプリケーションプログラムはすべて共通である．システム構成がレベルアップしても，プログラムはすべて完全な互換性を保つ．

3—2　マークⅡエディタ（EDIT）

標準エディタに準ずるが入出力としては，補助記憶ファイルを含めすべての周辺機器が使用可能である．また，文字ストリングサーチの機能が加わった．

3—3　マークⅡアセンブラ（SAPD）

SAP—Ⅲに準ずるが，入力機器としてはすべての補助記憶ファイル，出力機器としては補助記憶ファイルを含めてすべての周辺機器が使用可能である．シンボル定義，バイナリ出力，リスティング出力は自動的にリンクされ，1パスとなっている．

3—4　マークⅡフォートラン

コンパイラ，アセンブラ，リンキングローダの各ステップを追う本格的フォートランであるが，コンパイル，アセンブル（リスティングを含めると2フェーズ），ロードおよびライブラリの自動

サーチ，オブジェクトの実行の各フェーズは，すべて自動的にリンクされ，1パスとなっている．入出力機器の割当てはオブジェクトプログラムの実行中に，プログラムによって指定することも可能である．JIS3000を充分満足するだけでなく，フォートランソース中にアセンブラコーディングを自由に混ぜることが可能であるので，場合によっては従来の中型機，大型機のフォートラン以上に強力である．チェイニング機能により，コア容量の数倍に達するプログラムの実行も可能である．

3—5　ファイルユーティリティ（CONVRT，IMAGE）

マークⅡモニタが管理できる数100個のファイルを有効に検索したり，削除したり，入出力機器間で転送したりするサービスをおこなうためのユーティリティプログラムである．

3—6　事務用フォートランシステム

マークⅡフォートランとコンパイラ，アセンブラ，リンキングローダを共用し，ライブラリを新しくすることにより，36ビット（$\pm 3 \times 10^{10}$）の整数演算を可能としたシステムである．事務用に有用な，フォーマッティング仕様が付加されている．

3—7　マークⅣフォートラン

JIS7000レベルを完全に満足するフォートランシステムで，各種の補助記憶をプログラムおよびデータ用として有効に利用できるので，非常に大きなプログラムの実行が可能となっている．各種周辺機器の実行時における割付はマークⅡフォートランにくらべ，さらに自由におこなうことができる．

オプションとしてCPUに金物の倍長演算，浮動演算パッケージを付加することにより，演算とデータ収集などの並行処置ができることになり，その機能は飛躍的に増大する．

3—8　レポートプログラムジェネレータ（RPG）

RPGはIBM社が事務処理用としてプログラミング不要のシステムとして開発，発展させてきたものであるが，これをミニコン用として実用的な事務処理システムとして完成したものである．

RPGシステムの採用により，日常の事務処理の大部分は専門的なプログラム技術者をわずらわすことなく，開発，運用ができる．

3—9　マークⅡコボル（準備中）

3—10　ソートユーティリティ

事務処理のかなりの部分を占めるソート業務を容易におこなうためのユーティリティとして用意されている．カートリッジ磁気ディスクの特性を考慮して，ソート時間が最少となるように設計した．ソートキーの位置，昇順，降順，バイトデータ，語データなど自由に混用できる．

3—11　オクタルデバッガ（POD）

紙テープシステムのCOD，XODと同様の目的を果すが，コア常駐部が0という特長がある．

4．保守診断プログラム

保守診断プログラムはCPUをはじめ，システムを構成する機器の各々についてその機能をすべてチェックすることができる．システムの保全，故障個所の発見が容易におこなうことができ，異常がなければ一応正常な機能を果していると考えられる．

4—1　基本保守診断

CPU，主記憶装置およびテレタイプの動作診断用で次のようなテストがおこなわれる．

1. 命令テストNO. 1
2. 〃　NO. 2
3. 〃　NO. 3　すべての命令群の実行テスト，間接アドレス
4. 〃　NO. 4　オートインデックス機構等をテストする．
5. 〃　NO. 5

6. 命令テストNO. 6
7. 電源接断に対するメモリ保護のテスト：電源ON/OFF時のメモリ保護をチェックする.
8. メモリアドレステスト：コアのロケーション（アドレス）をチェックする.
9. メモリチェックボードテスト：コアのメモリエラーをチェックする.
10. テレタイプテスト　NO. 1：テレタイプのリーダおよびパンチャをチェックする.
11. テレタイプテスト　NO. 2：印字，キーボードおよび総合テスト

4—2　高速紙テープリーダ，高速紙テープパンチャ，テスト

高速紙テープリーダおよびパンチャの機能テストおよびコントロール部のテスト，増幅器のテスト等をおこなう.

テスト　1　高速紙テープリーダおよびリーダコントロールに関する基本的なテスト
〃　2　高速紙テープパンチおよびパンチコントロールの基本テスト
〃　3　特殊な紙テープを使ったリーダテスト
〃　4　特殊な紙テープをパンチするテスト
〃　5　テスト4のパンチしたテープをチェックする.
〃　6　ランダムなCHARACTERをパンチさせるテスト
〃　7　テスト6のパンチしたテープをチェックする.
〃　8　特殊な紙テープのパンチとリードの結合テスト
〃　9　リーダ増幅器の調整テスト
〃　10　SRにセットした任意のCHARACTERをパンチさせるテスト
〃　11　特殊な紙テープをパンチさせる.
〃　12　高速紙テープリーダのスピード測定
〃　13　高速紙テープパンチャのパンチスピード測定
〃　14　任意の長さのブロックを，任意の待ち時間でブロック読みをするテスト
〃　15　高速紙テープリーダのリーダ速度を測定
〃　16　高速紙テープパンチャのパンチ速度を測定

4—3　カセットシステムテスト各種

4—4　磁気ドラムテスト各種

4—5　拡張メモリ用テスト各種

4—6　ラインプリンタテスト各種

4—7　磁気テープ装置（IBMコンパチブル）テスト各種

5．各種ユーティリティプログラム

5—1　バイナリパンチプログラム

任意のロケーションの内容をバイナリフォーマットの紙テープとして出力する．出力装置はPTP・TTYを自動選択し，データフィールドの指定，パンチ指令，サムチェック，エントリー指定等が可能であり，すべての操作はキーボードオリエントになっている.

5—2　デュープおよびベリファイプログラム

バイナリテープからマスタバイナリテープの作製，マスタバイナリテープの複製とそのチェックをおこなうことができる.

5—3　バイナリプログラム（IDBIN）

バイナリテープを読んで，コメント，使用コアロケーションおよびチェックサムのプリントをおこなう.

5—4　3倍長整数演算システム

インタプリータ形式によるソフトウェアパッケージで有効数字10桁の演算が可能である.

5—5 フローティングポイントパッケージ

インタプリータ形式によるソフトウェアパッケージで出力形式を任意に指定でき有効6桁の実数計算ができる. 指数表示では0.999999×10^{616}～0.999999×10^{-616}の範囲を一定数として扱うことができ, 他の拡張函数パッケージと接続して, 科学技術演算用プログラムとして使用される.

5—6 SAPライブラリおよびライブラリエディタ

SAPライブラリ（ASC II ソース）の中から必要なサブルーチンのソースのみをコピーするユーティリティプログラム.

6. 各種アプリケーションプログラム

6—1 発券機群管理システム

200～600台の発券機からの発売データをオンラインで主記憶装置に登録すると同時に, オッズ表示を時々刻々におこなう. さらにコンソールからの指令によって合計10種の帳票の自動作成を即時におこなう. 特に有機的なバックアップの設計によりシステムのどの部分の事故に対しても複数個の処理ルートを用意してあるのでサービスが中断されない.

6—2 高速自動データ集録処理システム

最高64個のアナログ入力を任意に指定し, 最高20Kサンプル/secの速度でデータ集録をおこなう. 磁気テープに集録されたデータは各チャネルごとに分離し, 各種の計算をおこなう. 中央処理装置はマルチプログラム可能の555M形を使用, 磁気テープ装置は75″/sec800BPI, A/D変換装置は15ビット, 50μsの性能を有する.

6—3 建築・土木・測量計算システム

建築・土木・測量計算を電子計算機によりおこなうもので, プログラムとして下記のものが用意されている.

1. 多角測量座標計算（閉合トラバース, 結合トラバース）
2. 多角測量座標計算（閉放トラバース）
3. 細部多角測量座標計算（開放トラバース）
4. 縦横線および高低計算
5. 多角網平均座標計算（H型, A型, θ型）
6. 多角網平均座標計算（X型, Y型）
7. 面積計算（経緯距法）
8. 座標差による距離と方向角の計算（逆計算）
9. 交点座標計算
10. 剛結構造計算（多元一次連立方程式解法計算）
11. 土積計算
12. 配水管網計算
13. その他

6—4 事務管理システム

6—5 病院管理システム

6—6 マイクロコンピュータ用

各種クロスアセンブラ, クロスコンパイラ, シミュレータ

6—7 その他

1. プリント基板自動試験システム
2. 電子式卓上計算機自動試験システム

3. 振動分析自動処理システム
4. 試験・計測データ自動処理システム
5. 情報処理装置用自動配線機システム
6. 自動配線試験システム
7. 自動図柄分析システム
8. 丸編機群管理システム
9. 脳波自動解析
10. 心電図自動解析
11. PROM書込みシステム
12. 回線制御データ通信システム

❷ 紙テープシステム

§1 ローダ

1. RIMローダ

RIMローダはRIM形式にせん孔された紙テープをコアに格納するためのプログラムで，通常はスイッチレジスタを通してコアに格納して用いる．このため，プログラムの長さができるだけ短くなるように設計されている．

1—1 RIM形式

RIM形式の紙テープはリーダ部と，4字で1フレームを構成するデータ部，トレーラ部の3つの部分から成る．リーダ部，トレーラ部はオクタルコード200の連続から成り，トレーラ部は紙テープの最後までせん孔されていることが必要である．

データ部は1XY，0ZW，0AB，0CDの4字組の繰返しから成り，オクタルABCDのデータをオクタルXYZWの番地に格納する．RIM形式のテープは〝CEC PUNCH〟プログラムで作製できる．

1—2 周辺機器

入出力タイプライタ，または高速紙テープリーダ

1—3 操 作

7756番地～7775番地にプログラムをスイッチレジスタによって格納する．プログラムは入出力タイプライタ用と高速紙テープリーダ用と2種類あるので，該当するプログラムを使用すること，プログラムに対応したリーダ紙テープをセットし，SA＝7756としてスタートする．紙テープを読終っても，計算機はHALTしないのでSTOPを押す．RIMローダではなんらチェック機能がないので，テープのせん孔誤りのないテープを使用すること．またロードしたプログラムの動作がおかしい場合は再ロードを試みるのがよい．

RIMローダでロードするプログラムは通常BINローダ，またはCBNローダ，カセットローダなどである．

1—4 注 意

RIMローダは繰返し使用が可能である（リエントラント）．したがって7756～7775番地の内容が破壊されていなければ，使用の都度格納する必要はない．RIMローダはオクタルコード200のフィード記号以外でも200のビットにせん孔があれば，これを無視するのでリーダ部にASCII文字によるコメントを設けても，なんら差支えがないのでラベルとして利用できる．

RIMローダおよびこれによって読込まれるローダはDF，IFによって指定する任意のフィールドでよい．

```
                    #RIM LOADER-TTY

                    *7756
7756 6032     BEGIN,  KCC
7757 6031             KSF
7760 5357             JMP .-1
7761 6036             KRB
7762 7106             CLL RTL
7763 7006             RTL
7764 7510             SPA
7765 5357             JMP BEGIN+1
7766 7006             RTL
7767 6031             KSF
7770 5367             JMP .-1
7771 6034             KRS
7772 7420             SNL
7773 3776             DCA I TEMP
7774 3376             DCA TEMP
7775 5356             JMP BEGIN
7776 0000     TEMP,   0

BEGIN   7756    TEMP    7776
```

```
                    #RIM LOADER-RTR

                    *7756
7756 6014     BEGIN,  RFC
7757 6011             RSF
7760 5357             JMP .-1
7761 6016             RRB RFC
7762 7106             CLL RTL
7763 7006             RTL
7764 7510             SPA
7765 5357             JMP BEGIN+1
7766 7006             RTL
7767 6011             RSF
7770 5367             JMP .-1
7771 6016             RRB RFC
7772 7420             SNL
7773 3776             DCA I TEMP
7774 3376             DCA TEMP
7775 5357             JMP BEGIN+1
7776 0000     TEMP,   0

BEGIN   7756    TEMP    7776
```

2. HELPローダ

HELPローダはHELP形式でせん孔された紙テープをコアに格納するためのプログラムで，通常はスイッチレジスタによって格納して用いる．RIMローダに比較するとスイッチレジスタを通して格納するプログラムがRIMローダの16ロケーションに対して9ロケーションと少ないことと，読込む紙テープの長さが約½と短いので，入出力タイプライタによってBINローダ，CBNローダ，Vローダなどを格納する場合に時間が少なくてすむ．HELPローダはリエントラントでないので使用の都度コアに格納する必要がある．入力機器としては入出力タイプライタだけが許される．

2—1 HELP形式

HELP GENERATORプログラムによって作製した7600～7777番地のプログラムに対応するテープをいう．HELPローダによってコアにロードすることができる．

2—2 HELP GENERATOR

本プログラムは7600～7777番地に格納されているプログラムをHELP形式で入出力タイプライタを用いて紙テープにせん孔する．

1. **周辺機器**　入出力タイプライタ
2. **メモリロケーション**　7400～7577番地
3. **操　作**

コアにロード後SA＝7400とする．入出力タイプライタはON-LINE，PUNCH ONとする．CPUをスタートさせると，7600～7777番地のコアメモリの内容をHELP形式でせん孔してHALTする．HELP GENERATORおよびこれによってせん孔されるプログラムはDF，IFによって指定する任意のフィールドでよい．

2—3 HELP LOADER

HELP LOADERは，27～40番地にスイッチレジスタによって格納されるブートストラップ部とHELP形式の一部として紙テープにせん孔されており，ブートストラップ部でコアの6～30番地に読込まれる本体部分とから成る．

1. **周辺機器**　入出力タイプライタ
2. **メモリロケーション**　6～40番地
3. **操　作**

27～40番地にスイッチレジスタを通して次のプログラムをロードする．入出力タイプライタのリーダ部分にHELP形式のテープをセットしてSTARTとする．SA＝0027としてCPUをスタートさせる．紙テープリーダは最後のリーダ部分まで読込んでCPUはHALTする．

```
                    #HELP LOADER BOOTSTRAP

                    TOP=    7

                    *27
0027 6031           GO,     KSF
0030 5000                   JMP .-1
0031 6036                   KRB
0032 7450                   SNA
0033 5027                   JMP GO
0034 7012                   RTR
0035 7010                   RAR
0036 3007           STORE,  DCA TOP
0037 2036                   ISZ STORE
0040 5027                   JMP GO

GO      0027    STORE   0036    TOP     0007
```

3. XRIMローダ

XRIMローダは、RIMローダ、HELPローダと同様にローダ自体をコアに格納するためのプログラムで、XRIM形式のテープを入出力タイプライタ、または高速紙テープリーダを使用して読み込むことができる.

3—1 XRIM形式

XRIM形式の紙テープはリーダ部（ブランクフィールド）と4字で1フレームを構成するデータ部、終端部、トレーラ部からなる.

データ部、終端部は、1AB、000、000、OCDの4字組の繰返しでオクタルABCDのデータを表現する. XRIM形式の紙テープは指定番地から順に格納され、最後にXRIMローダ自身の一部を書き換えて、停止または自動スタートを実行する.

3—2 周辺機器

入出力タイプライタ、または高速紙テープリーダ

3—3 操作

入力機器が高速か低速かに応じ、0—10番地または1—12番地にスイッチレジスタからプログラムを格納する. 対応する入力機器に紙テープをセットして、0番地（高速）、または1番地（低速）から実行を開始する. トレーラ部分の前まで読み込んで計算機は停止する. XRIMローダも、他のブートストラップローダと同様にチェック機能がないのでテープのせん孔の誤りのないテープを使用すること、またロードしたプログラムの動作がおかしい場合は、再ロードを試みるのがよい.

3—4 注意

XRIMローダは、いったん実行すると、XRIMローダ自体の内容を一部破壊する. したがって、XRIMローダは使用の都度格納する必要がある.

```
                #XRIM LOADER-PTR
                SA=7600
                *0
0000 6014               RFC
0001 6011       LOOP,   RSF
0002 5001               JMP .-1
0003 6016               RRB RFC
0004 7430               SZL
0005 3410               DCA I AX
0006 7106               CLL RTL
0007 5001               JMP LOOP
0010 7577       AX,     SA-1

   AX       0010     LOOP     0001     SA       7600
$
```

```
                #XRIM LOADER-TTY
                SA=7600
                *1
0001  6031      LOOP,   KSF
0002  5001              JMP .-1
0003  3013              DCA W
0004  6036              KRB
0005  1013              TAD W
0006  7430              SZL
0007  3412              DCA I AX
0010  7106              CLL RTL
0011  5001              JMP LOOP
0012  7577      AX,     SA-1
0013  0000      W,      0

 AX       0012     LOOP     0001     SA       7600     W        0013
```

4. BIN形式

BIN形式とは紙テープに保存したプログラム，またはデータを直接コアメモリに読込むために定めた紙テープ上の一つの形式である．

BIN形式の紙テープとは

1. 見出し部（オプション）
2. リーダ部
3. ラベル部（オプション）
4. データ部
5. チェックサム部
6. エントリ指定部（オプション）
7. トレーラ部

から成り立つ．

4—1 見出し部

紙テープを目で見たときに直接読めるように，ASCII文字の形にせん孔した部分をいう．〝CEC PUNCH〟プログラムを利用して作製することができる．見出し部分のデータに対する唯一の条件は紙テープ上のビット8（オクタル値200）にせん孔がないことである．この見出し部分の前，またはその途中の任意の部分から読ませるとエラーを生ずるので，この部分の次のリーダ部分からリーダで読取らねばならない．

4—2 リーダ部

1個以上のフィードコード（オクタル値200）からなり，必ずこの部分から読始めなければならない．SAPアセンブラを使用して作製されたテープには自動的に付加されている．〝CEC PUNCH〟，〝ODT〟，などに含まれるBINパンチのプログラムでも各々必要なコマンドにより作製できる．

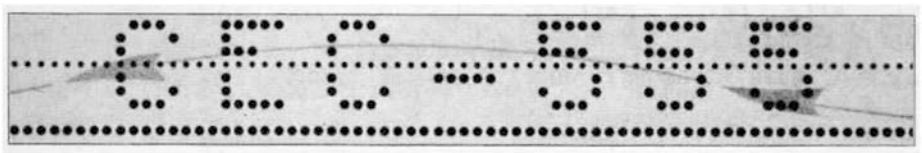
4—3 ラベル部

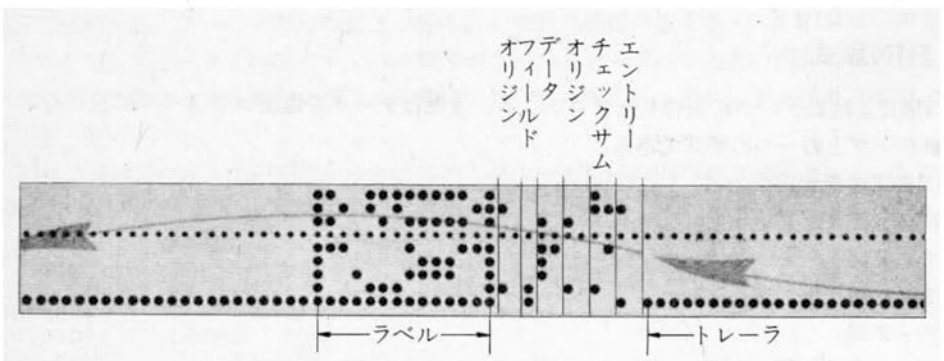
両端をRUBOUTコード（オクタル377）で囲まれた任意のデータであって，その後にリーダ部と同じフィードコードの連続を伴う．コアにロードされる内容とは無関係である．ラベル部分のデー

タに対する唯一の制限はRUBOUTコードを含んではならないことである.

BINローダではこの部分を無視するので，この部分の前後のどちらのリーダ部分から読取りを開始してもよい.

CBNローダまたはVローダを使用する場合は，ラベル部の前のリーダ部分から読取りを開始すれば，ラベル部分を入出力タイプライタに印刷するのでロードした紙テーププログラムの識別に役立つ．ラベル部分の後のリーダ部から読取りを開始してもコア上のデータに関しては同一である.





またオフラインでこのラベル部分を印刷させて紙テーププログラムの識別に役立てることができる.

SAPアセンブラを使用した場合，＃疑似命令によってこのラベル部分を作製することができる.

"CEC PUNCH" プログラムではCコマンドによって，ラベル部分を作製することができる.

4—4　データ部

データ部分は，フィールド指定（オプション），オリジン指定と，これに続くデータの連続の3つの部分から成り立つ.

1.　フィールド指定（オプション）

オクタルコード3XYの形式のせん孔はローダに対して，以下のデータを格納すべきコアのフィールドを指定する，Yの部分は無視される.

SAPアセンブラではFIELD疑似命令で作製される.

2.　オリジン指定

オクタルコード1XY，0ZWの2字の連続した組合わせで1フレームを構成し，オクタルXYZWのロケーションに以下のデータを格納することを指定する.

SAPアセンブラでは*疑似命令またはPAGE疑似命令によって作製される.

3.　データ

オクタルコード0XY，0ZWの2字の連続した組合わせで1フレームを構成し，オクタルXYZWのデータとしてコアに順次格納される．

4—5 チェックサム部

トレーラ部またはエントリ指定部の直前の2字0XY，0ZWで1フレームのせん孔であって，XYZWがチェックサムの内容となる．チェックサムに関係するのは，オリジン指定部およびデータのすべてのせん孔であって，各せん孔の8ビット値をすべて加算した値をチェックサムとする．12ビットを超えてオーバフローした部分は無視する．ローダは読取り時に各字の値を積算し，このチェックサム部のデータと比較して，読取りエラーの有無を検査する．

4—6 エントリ指定部

2XY，0ZWの形式の2字で1フレームを構成し，オクタルXYZWの値によって，エントリ番地の指定を可能にする．SAPアセンブラでは$擬似命令によってエントリ指定部のせん孔を行う．〝CEC PUNCH〟プログラムではGコマンドによる．

4—7 トレーラ部

リーダ部に等しい．ローダはトレーラ部，またはエントリ指定部の検出によりチェックサムの照合および動作の停止を行う．

5. BIN LOADER

5—1 目 的

BIN形式のプログラム紙テープをコアにロードするために用いる．

5—2 周辺機器

入出力タイプライタ，または高速紙テープリーダ

5—3 操 作

1. BIN形式のプログラム紙テープを入出力タイプライタまたは高速紙テープリーダに正しくセットする．リーダコード（オクタル200）の部分をセットする．
2. 入出力タイプライタまたは紙テープリーダのセット
   紙テープ入力として入出力タイプライタのリーダ部を使用する場合はレバーをスタートに置く．
   高速紙テープリーダを使用する場合は高速紙テープリーダのスイッチを電源ON，CPUとする．
   DFはプログラムをロードするフィルド．IFはローダのあるフィルド．
3. SA＝7777とする．
4. 入力機器として入出力タイプライタを用いる場合は，SR_0＝1（すなわちSA＝7777にセットしたそのまま），紙テープリーダを用いる場合はSR_0＝0（すなわちSA＝7777にセットした後0777または3777とする）としてスタートさせる．
5. RUBOUTで囲まれている部分は無視される．
6. 読込みが終了するとRUNが消えるから，AC＝0を表示していることを確認する．AC≠0の時は，読取り時にエラーを生じたか，BINテープにせん孔誤りがあるかのいずれかであるから1.に戻って再び試みる．

5—4 注 意

1. 入出力タイプライタのリーダ部を用いてプログラムを読込んでいる時には，入出力タイプライタのキーボードを押すとエラーとなるので手を触れないこと．
2. BINローダはRIM形式，HELP形式，BIN形式の3種類が用意されている．各々の形式に対応するローダを使用してコアに読込む．
3. ローダのあるフィールドとオブジェクトプログラムを読込むフィールドの選択については任意である．オブジェクトテープにフィールド指定のせん孔があればDFの指定に優先する．

```
                #       BIN-LOADER (MAY 16A, 1972 PUNCH)

                *7612
7612 0000       SWITCH, 0
7613 0000       MEMTEM, 0
7614 0000       CHAR,   0
7615 0000       CHKSUM, 0
7616 0000       ORIGIN, 0

                *7626
                /EXTRACT ERROR,FIELD,L/T
7626 0000       BEGG,   0
7627 3212               DCA SWITCH          /SET SWITCH
7630 4260               JMS READ            /GET CHARACTER
7631 1300               TAD M376            /TEST FOR 377
7632 7750               SPA SNA CLA
7633 5237               JMP .+4             /NO
7634 2212               ISZ SWITCH          /YES: COMPLEMENT SWITCH
7635 7040               CMA
7636 5227               JMP BEGG+1
7637 1212               TAD SWITCH          /NOT 377
7640 7640               SZA CLA             /IS SWITCH SET
7641 5230               JMP BEGG+2          /YES:IGNORE
7642 1214               TAD CHAR            /NO:TEST FOR CODE
7643 0274               AND MASK            /TYPES
7644 1341               TAD M200
7645 7510               SPA
7646 2226               ISZ BEGG            /DATA FOR ORIGIN
7647 7750               SPA SNA CLA
7650 5626               JMP I BEGG          /DATA,ORIGIN,OR L/T
7651 1214               TAD CHAR            /FIELD SETTING
7652 0256               AND FMASK
7653 1257               TAD CHANGE
7654 3213               DCA MEMTEM
7655 5230               JMP BEGG+2          /CONTINUE INPUT
7656 0070       FMASK,  70
7657 6201       CHANGE, 6201                /CDF CHANGE DATA FIELD
7660 0000       READ,   0
7661 0000               0
7662 6031       LOP,    KSF                 /WAIT FOR FLAG
7663 5262               JMP .-1
7664 6036               KRB
7665 3214               DCA CHAR
7666 1214               TAD CHAR
7667 5660               JMP I READ
7670 6011       FIF,    RSF
7671 5270               JMP .-1
7672 6016               RRB RFC
7673 5265               JMP LOP+3
7674 0300       MASK,   300

                /TRAILER CODE SEEN
7675 4343       BEND,   JMS ASSEMB
7676 7041               CIA
7677 1215               TAD CHKSUM
7700 7402       M376,   HLT
7701 6032       BEGIN,  KCC
7702 6014               RFC
7703 6214               6214                /CDF READ DATA FIELD
7704 1257               TAD CHANGE
7705 3213               DCA MEMTEM          /SAVE FIELD INSTRACTION
7706 7604               CLA OSR
```

```
7707 7700           SMA CLA
7710 1353           TAD HIR1
7711 1352           TAD LOR1
7712 3261           DCA READ+1
7713 4226           JMS BEGG
7714 5313           JMP .-1             /IGNORE LEADER

7715 3215     CK,   DCA CHKSUM
7716 1213           TAD MEMTEM
7717 3336           DCA MEMFLD
7720 1214           TAD CHAR
7721 3376           DCA WORD1
7722 4260           JMS READ
7723 3355           DCA WORD2
7724 4226           JMS BEGG            /LOOK AHEAD
7725 5275           JMP BEND            /TRAILER END
7726 4343           JMS ASSEMB
7727 7420           SNL
7730 5336           JMP MEMFLD
7731 3216           DCA ORIGIN
7732 1376     CHEX, TAD WORD1
7733 1355           TAD WORD2
7734 1215           TAD CHKSUM
7735 5315           JMP GO

7736 0000     MEMFLD, 0
7737 3616           DCA I ORIGIN
7740 2216           ISZ ORIGIN
7741 7600     M200, 7600
7742 5332           JMP CHEX

7743 0000     ASSEMB, 0
7744 1376           TAD WORD1
7745 7106           CLL RTL
7746 7006           RTL
7747 7006           RTL
7750 1355           TAD WORD2
7751 5743           JMP I ASSEMB.
7752 5262     LOR1, JMP LOR
7753 0600     HIR1, HIR-LOR
7754 0000           0
              WORD1=7776
7755 0000     WORD2, 0

              *7777
7777 5401           JMP BEGIN
```

ASSEMB	7743	BEGG	7626	BEGIN	7701	BEND	7675
CHANGE	7617	CHAR	7614	CHEX	7732	CHKSUM	7615
EMASK	7656	GO	7715	HIR	7670	HIR1	7753
LOR	7662	LOR1	7752	MASK	7674	MEMFLD	7736
MEMTEM	7613	M200	7741	M376	7700	ORIGIN	7616
READ	7660	SWITCH	7612	WORD1	7776	WORD2	7755

6. CBN LOADER

6—1 目 的

BIN形式のプログラム紙テープをコアにロードするために用いる．BINローダに対する相異は次の通りである．

1. 入力機器の選択（高速紙テープリーダまたはTTY）をスイッチレジスタを使用せず，BINテープをセットした側から自動的に読込むようにした．
2. RUBOUT（コード377）で囲まれたラベルの部分をTTYに印刷する．これによりロードしたプログラムの識別が容易となる．
3. CHECKSUMの後にENTRYがパンチしてあるBINテープでは，スイッチレジスタにスタート番地をセットしてLOAD PCを行うことなく，単にCONTを押すだけで，オブジェクトプログラムのスタートが可能である．この場合，DFおよびIFの指定は最後に与えられたFIELD指定による．

6—2 周辺機器

入出力タイプライタ

高速紙テープリーダ（オプション）

6—3 操 作

1. BIN形式のプログラム紙テープを入出力タイプライタ，または高速紙テープリーダに正しくセットする．必ずリーダコード（オクタル200）の部分をセットする．
2. 入出力タイプライタはON-LINE
3. 紙テープ入力として入出力タイプライタのリーダ部を使用する場合はレバーをスタートに置く．高速紙テープリーダを使用する場合は高速紙テープリーダのスイッチを電源ON，CPUとする．DFはプログラムをロードするフィールド，IFはローダのあるフィールド．
4. SA＝7777としてスタートさせる．
5. RUBOUTで囲まれたラベルを含んだBINテープであればそれを入出力タイプライタに印刷する．RUBOUTで囲まれたアセンブラのエラーメッセージについても同様である．
6. BINテープの読込みを終了した場合，チェックサムが合わない時はCPUのRUNが消えない．読取り誤りを生じたか，BINテープのせん孔誤りのいずれかであるから，1.に戻って再ロードを試みる．
7. チェックサムOKの場合はPC＝7624でRUNが消える．
8. ENTRYがせん孔されたテープの場合はACにその値が表示されている．
   CONTを押せばその番地からスタートする．プログラムをスタートさせるのではなく，さらに別のプログラムを読込むのであれば1.に戻る．
9. ENTRYがせん孔されていないテープの場合はAC＝7777が表示されている．さらにプログラムをロードするのであれば1.に戻る．プログラムをスタートさせるのであればSRを指定された値にセットしてスタートさせる．

6—4 注 意

1. プログラムを読込んでいる時に入出力タイプライタのキーボードを押すとエラーとなるので手を触れないこと．
2. CBNローダはRIM形式，HELP形式，BIN形式の3種類が用意されている．各々の形式に対応するローダを使用してコアに読込む．
3. ローダのあるフィールドとオブジェクトプログラムを読込むフィールドの選択については任意である．オブジェクトテープにフィールド指定のせん孔があればDFの指定に優先する．

```
                #CBN+RIM-H LOADER: MAY 2CA, 1972
                /       OCCUPIES LOCATIONS 7600-7754, 7777
                /AUTOMATICALLY SELECTS PTR OR TTY
                /PRINTS COMMENT
                /DYNAMIC STOP WITH ERROR
                /HALT, ENTRY POINT AT AC; PRESS CONTINUE TO GO
                /SA=7777

                *7600
                /END
7600 4332       BEND,   JMS ASSEMB
7601 7041               CIA
7602 1350               TAD CHKSUM
7603 7440               SZA
7604 5204               JMP .           /DYNAMIC STOP: SUMCHECK ERROR
7605 1261               TAD CHAR
7606 1257               TAD M200
7607 3345               DCA WORD1
7610 4267               JMS BEGG
7611 5226               JMP NOGO        /NO ENTRY PUNCH
7612 1261               TAD CHAR
7613 3346               DCA WORD2
7614 6214               6214            /RDF
7615 1354               TAD XCIF
7616 3220               DCA HALT-1
7617 4332               JMS ASSEMB
7620 0000               0
7621 3344       HALT,   DCA ENTRY
7622 1344               TAD ENTRY
7623 7402       M376,   HLT
7624 7200               CLA
7625 5744               JMP I ENTRY
                /NO ENTRY POINT
7626 7360       NOGO,   STA STL
7627 5221               JMP HALT
                /START
7630 6032       BEGIN,  KCC
7631 6014               RFC
7632 1347               TAD XNOP
7633 3271               DCA CDFN
7634 4267               JMS BEGG
7635 5234               JMP .-1         /LEADER
7636 3350       GO,     DCA CHKSUM
7637 1261               TAD CHAR
7640 3345               DCA WORD1
7641 4317               JMS READ
7642 3346               DCA WORD2
 643 4267               JMS BEGG
7644 5200               JMP BEND        /TRAILER
7645 4332               JMS ASSEMB
7646 7420               SNL
7647 5255               JMP STORE
7650 3344               DCA ORIGIN      /SET ORIGIN
7651 1345       CHEX,   TAD WORD1
7652 1346               TAD WORD2
7653 1350               TAD CHKSUM
7654 5236               JMP GO
7655 3744       STORE,  DCA I ORIGIN
7656 2344               ISZ ORIGIN
7657 7600       M200,   7600            /CLA
7660 5251               JMP CHEX
                /PRINT 1 CHARACTER
```

```
                    CHAR,
7661 0000           PRINT,  0
7662 6046                   TLS
7663 7200                   CLA
7664 6041                   TSF
7665 5264                   JMP .-1
7666 5661                   JMP I PRINT
                    /CHECK L/T, COMMENT & FIELD-SET
7667 0000           BEGG,   0
7670 3332                   DCA SWITCH
7671 0000           CDFN,   0                   /NOP OR CDF N
7672 4317           BEGG2,  JMS READ
7673 1223                   TAD M376
7674 7750                   SPA SNA CLA         /BUBOUT?
7675 5301                   JMP BEGG1           /NO
7676 2332                   ISZ SWITCH          /SET OR RESET COMMENT SWITCH
7677 7240                   STA
7700 5270                   JMP BEGG+1
7701 1332           BEGG1,  TAD SWITCH
7702 7640                   SZA CLA             /COMMENT?
7703 5341                   JMP COMENT          /YES
7704 1261                   TAD CHAR            /NO
7705 0353                   AND MASK
7706 1257                   TAD M200
7707 7510                   SPA
7710 2267                   ISZ BEGG            /DATA CHARACTER
7711 7750                   SPA SNA CLA
7712 5667                   JMP I BEGG
7713 1261                   TAD CHAR            /SET FIELD
7714 1351                   TAD XCDF
7715 3271                   DCA CDFN
7716 5272                   JMP BEGG2
                    /READ 1 CHARACTER
7717 0000           READ,   0
7720 6031           RDO,    KSF
7721 5326                   JMP RD2
7722 6036                   KRB
7723 3261           RD1,    DCA CHAR
7724 1261                   TAD CHAR
7725 5717                   JMP I READ
7726 6011           RD2,    RSF
7727 5320                   JMP RDO
7730 6016                   RRB RFC
7731 5323                   JMP RD1
                    /COMBINE 2 CHARACTERS
                    SWITCH,
7732 0000           ASSEMB, 0
7733 1345                   TAD WORD1
7734 7106                   CLL RTL
7735 7006                   RTL
7736 7006                   RTL
7737 1346                   TAD WORD2
7740 5732                   JMP I ASSEMB
                    /
7741 1261           COMENT, TAD CHAR
7742 4261                   JMS PRINT
7743 5271                   JMP BEGG+2
                    /
                    ENTRY,
7744 0000           ORIGIN, 0
7745 0000           WORD1,  0
7746 0000           WORD2,  0
7747 7000           XNOP,   NOP
7750 0000           CHKSUM, 0
```

```
7751 5701     XCDF,    6201-300
7752 7773     M5,      -5
7753 0300     MASK,    300
7754 6202     XCIF,    6202
              /RIM LOADER, PTR
              *7756
7756 6014     RIM,     RFC
7757 6011              RSF
7760 5357              JMP .-1
7761 6016              RRB RFC
7762 7106              CLL RTL
7763 7006              RTL
7764 7510              SPA
7765 5357              JMP RIM+1
7766 7006              RTL
7767 6011              RSF
7770 5367              JMP .-1
7771 6016              RRB RFC
7772 7420              SNL
7773 3776              DCA I RIMW
7774 3376              DCA RIMW
7775 5357              JMP RIM+1
7776 0000     RIMW,    0
              *7777
              /ENTRY
7777 5230              JMP BEGIN
```

```
ASSEMB  7732    BEGG    7667    BEGG1   7701    BEGG2   7672
BEGIN   7630    BEND    7600    CDFN    7671    CHAR    7661
CHEX    7651    CHKSUM  7750    COMENT  7741    ENTRY   7744
GO      7636    HALT    7621    MASK    7753    M200    7657
M376    7623    M5      7752    NOGO    7626    ORIGIN  7744
PRINT   7661    RD0     7720    RD1     7723    RD2     7726
READ    7717    RIM     7756    RIMW    7776    STORE   7655
SWITCH  7732    WORD1   7745    WORD2   7746    XCDF    7751
XCIF    7754    XNOP    7747
```

$

7. VERIFY LOADER

7—1 目 的

BIN形式のプログラム紙テープをコアにロードするために用いる．BINローダに対する相異は次の通りである．

1. 入力機器の選択（高速紙テープリーダまたはTTY）にスイッチレジスタを使用せず，BINテープをセットした側から自動的に読込むようにした．
2. RUBOUT（コード377）で囲まれたラベルの部分をTTYに印刷する．これによりロードしたプログラムの識別が容易となる．
3. 必要があればスイッチレジスタSR_0の操作によりBINテープの内容とコアの内容を比較し，異なっている場合はその番地，コア内容，テープ内容を並べて印刷させるベリファイ機能がある．これによりある種のプログラムデバッグが容易となる．

7—2 周辺機器

入出力タイプライタ

高速紙テープリーダ（オプション）

7—3 操 作

1. BIN形式のプログラムテープを入出力タイプライタまたは高速紙テープリーダに正しくセットする．必ずリーダコード（オクタル200）の部分にセットする．
2. 入出力タイプライタはON-LINE.
3. 紙テープ入力として入出力タイプライタのリーダ部を使用する場合はレバーをスタートに置く．高速紙テープリーダを使用する場合は高速紙テープリーダのスイッチを電源ON，CPUとする．DFはプログラムロードするフィールド，IFはローダのあるフィールド．
4. SA＝7777としてスタートさせる．

   この時，ベリファイ機能を利用する場合はSR_0＝0とする．

   SR_0のセッティングはプログラムをロードしている間に任意に変化させて，ベリファイ機能をON-OFFさせることができる．
5. ベリファイ機能をONとしている間，ベリファイローダは，データをコアに格納する前に，コアの内容と紙テープから読取ったデータとを比較する．等しければ何も行わないで先に進む．異なっている場合はコアの番地，コアの内容，紙テープから読取ったデータの3つをオクタル表示で印刷してから，紙テープからのデータをコアに格納する．
6. BINテープの読込みを終了した時，RUNが消えるので，AC＝0を確認する．AC≠0の場合は，紙テープの読取り時に誤りを生じたか，BINテープのせん孔に誤りがあるかのいずれかであるから1.に戻って再び試みる．

7—4 注 意

1. プログラムを読込んでいる時に入出力タイプライタのキーボードを押すとエラーとなるので手を触れないこと．
2. VローダはHELP形式，BIN形式の2種類が用意されている．各々の形式に対応するローダを使用してコアに読込む．
3. ローダのあるフィールドとオブジェクトプログラムを読込むフィールドの選択については任意である．オブジェクトテープにフィールド指定のせん孔があればDFの指定に優先する．
4. ベリファイ機能をコントロールするSR_0のセッティングは動作中に任意に変化させてよい．

```
                    /VERIFY LOADER:  OCT 31, 1973
                    /       OCCUPIES LOCATIONS 7600-7777
                    /AUTOMATICALLY SELECTS PTR OR TTY
                    /PRINTS COMMENT
                    /SR=1: ORDINARY LOADER
                    /SR=0: PRINTS PC, DIR & NEW CONTENT IF DIFFERENT
                    /SR=7777

                    *7600
                    /
                    /PRINT 1 CHARACTER
7600 0000   PRINT,  0
7601 6046           TLS
7602 7200           CLA
7603 6041           TSF
7604 5203           JMP .-1
7605 5600           JMP I PRINT
                    /
7606 6011   RD2,    RSF
7607 5247           JMP RDC
7610 6016           RRB RFC
7611 5252           JMP RD1
                    /
7612 5701   XDF,    6201-300
7613 0000   CHKSUM, 0
7614 0007   P7,     7
7615 0300   MASK,   300
                    /CHECK L/T, COMMENT & FIELD-SET
                    DCTW,
7616 0000   BEGG,   0
7617 3302           DCA SWITCH
7620 0000   XDF,    0                   /CLA OR CDF N.
7621 4246   BEGG2,  JMS READ
7622 1320           TAD M376
7623 7750           SPA SNA CLA         /RUBOUT?
7624 5230           JMP BEGG1           /NO
7625 2302           ISZ SWITCH          /SET OR RESET COMMENT SWITCH
7626 7240           STA
7627 5217           JMP BEGG+1
7630 1302   BEGG1,  TAD SWITCH
7631 7640   SPACE,  SZA CLA             /COMMENT?
7632 5311           JMP COMENT          /YES
7633 1376           TAD CHAR            /NO
7634 0215   CR,     AND MASK
7635 1374           TAD M200
7636 7510           SPA
7637 2216           ISZ BEGG            /DATA CHARACTER
7640 7750           SPA SNA CLA
7641 5616           JMP I BEGG
7642 1376           TAD CHAR            /SET FIELD
7643 1212   LF,     TAD XDF
7644 3220           DCA XDF
7645 5221           JMP BEGG2
                    /READ 1 CHARACTER
                    DIGCT,
7646 0000   READ,   0
7647 6031   RDC,    KSF
7650 5206           JMP RD2
7651 6036           KRB
7652 337E   RD1,    DCA CHAR
7653 1376           TAD CHAR
7654 5646           JMP I READ
```

```
                .
7655 0000       ORIGIN, 0
7656 0000       WORD1,  0
7657 0000       WORD2,  0
                /OCTAL PRINT ACC AFTER SPACE
7660 0000       OCTPR,  0
7661 7004               RAL
7662 3216               DCA OCTW
7663 1347               TAD M4
7664 3246               DCA DIGCT
7665 1231               TAD SPACE
7666 4200               JMS PRINT
7667 1216       OCTPR1, TAD OCTW
7670 7004               RAL
7671 7006               RTL
7672 3216               DCA OCTW
7673 1216               TAD OCTW
7674 0214               AND P7
7675 1301               TAD ZERO
7676 4200       OCTPR2, JMS PRINT
7677 2246               ISZ DIGCT
7700 5267               JMP OCTPR1
7701 5660       ZERO,   JMP I OCTPR
                /COMBINE 2 CHARACTERS
                PINW,
                SWITCH,
7702 0000       ASSEMB, 0
7703 1256               TAD WORD1
7704 7106               CLL RTL
7705 7006               RTL
7706 7006               RTL
7707 1257               TAD WORD2
7710 5702               JMP I ASSEMB
                /
7711 1376       COMENT, TAD CHAR
7712 4200               JMS PRINT
7713 5220               JMP BEGG+2
                /END
7714 4302       BEND,   JMS ASSEMB
7715 7041               CIA
7716 1213               TAD CHKSUM
7717 4260               JMS OCTPR
7720 7402       M376,   HLT
                /START
7721 6032       BEGIN,  KCC
7722 6014               RFC
7723 1374               TAD M200        /=CLA
7724 3220               DCA XOF
7725 4216               JMS BEGG
7726 5325               JMP .-1         /LEADER
7727 3213       CL,     DCA CHKSUM
7730 1376               TAD CHAR
7731 3256               DCA WORD1
7732 4246               JMS READ
7733 3257               DCA WORD2
7734 4216               JMS BEGG
7735 5314               JMP BEND        /TRAILER
7736 4302               JMS ASSEMB
7737 7420               SNL
7740 5346               JMP DATA
7741 3255               DCA ORIGIN      /SET ORIGIN
7742 1256       CHEX,   TAD WORD1
7743 1257               TAD WORD2
7744 1213               TAD CHKSUM
```

```
7745 5327           JMP GO
7746 3302   DATA,   DCA BINW
7747 7774   M4,     7774    / LAS
7750 7710           SPA CLA         /PRINT DIFFERENCE?
7751 5371           JMP STORE       /NO
7752 1302           TAD BINW
7753 7041           CIA
7754 1655           TAD I ORIGIN
7755 7650           SNA CLA         /ANY DIFFERENCE?
7756 5371           JMP STORE       /NO
7757 1234           TAD CR
7760 4200           JMS PRINT
7761 1243           TAD LF
7762 4200           JMS PRINT
7763 1255           TAD ORIGIN
7764 4260           JMS OCTPR
7765 1655           TAD I ORIGIN
7766 4260           JMS OCTPR
7767 1302           TAD BINW
7770 4260           JMS OCTPR
7771 1302   STORE,  TAD BINW
7772 3655           DCA I ORIGIN
7773 2255           ISZ ORIGIN
7774 7600   M200,   7600
7775 5342           JMP CHEX
7776 0000   CHAR,   0
            *7777
            /ENTRY
7777 5321           JMP BEGIN
```

ASSEMB	7702	BEGG	7616	BEGG1	7630	BEGG2	7621
BEGIN	7721	BEND	7714	BINW	7702	CHAR	7776
CHEX	7742	CHKSUM	7613	COMENT	7711	CR	7634
DATA	7746	DIGCT	7646	GO	7727	LF	7643
MASK	7615	M200	7774	M376	7720	M4	7747
OCTPR	7660	OCTPR1	7667	OCTPR2	7676	OCTW	7616
ORIGIN	7655	PRINT	7600	P7	7614	RDO	7647
RD1	7652	RD2	7606	READ	7646	SPACE	7631
STORE	7771	SWITCH	7702	WORD1	7656	WORD2	7657
XCDF	7612	XDF	7620	ZERO	7701		

$

8. BINテープのローディング手順

BIN形式で，紙テープに保存したプログラム，またはデータを，ローダを用いてメイン・メモリに読み込むための手順を示す．これは，ハードウェアによるオートローダや磁気ドラム・ディスク・テープ等の補助メモリを持たないシステムに適用される．

手順1． ロードすべき紙テープをさがす．

可視ラベルによって紙テープは，みわけることができる．可視ラベルは，プログラムのタイトル，製造年月日を直読できるような形にパンチしたものであり，SAPシリーズのアセンブラの“ ＃ ”疑似命令，または“BINパンチ”プログラムの“ C ”命令で作ることができる．

手順2． 紙テープのリーダ部（200のビットに穴のある部分）を読取機（TTYまたはPTR）にかけて，オンライン状態とする．

ここで，テープが走り出すようなら，CPUをSTOPにして，改めてテープをかけなおす．

手順3． ローダが常駐しているフィールドを，インストラクション・フィールド（IF）にセットする．

常駐フィールドが不明なら，まずIF＝0とする．できればすべてのフィールドにローダを常駐させておくのが賢明である．

手順4． テープをロードしたいフィールドを，データ・フィールド（DF）にセットする．

テープ上にフィールドの指定がある時は，その方が優先する．一般にはDF＝0でよい．

手順5． スイッチ・レジスタを7600にセットし，LOAD PCを押す．

すでに，CPUがRUN状態にある時は，STOPを押してからLOAD PCを押す必要がある．

手順6． ベリファイオプションを選択する時は，スイッチ・レジスタの最上ビットを0にする．

ベリファイオプションを選択すると，メインメモリの既存のデータと紙テープから読んだデータを照合できる．トグルスイッチのときはSR＝3600，ディジスイッチのときは，SR＝0600とすればよい．

手順7． STARTを押す．

CPUがRUNし，テープが走り出す．ここでテープが動かないとすれば，①操作ミス，②ローダがこわれている，③読取機がこわれている，④CPUがこわれているの順で疑ってみる．ローダが，こわれているときは，別のフィールドにローダが入っていれば，それを利用してローダを修復して後，手順1から繰返す．別のフィールドがないとき，別のフィールドにローダが入っていないときは，第3章の手順でローダをロードする．

注意：テープが走り出したら，TTYのキーボードに手をふれてはならない．読取りエラーを起す．

続いてTTYからタイトルがタイプ・アウトされる．

手順8． 期待したテープであることを確認する．

ちがっているときは，CPUをSTOPし，テープを巻もどして所定の場所に収納し，手順1にもどる．照合が選択されていて，メイン・メモリの既存のデータと紙テープから読んだデータに差がある場合，

FNNNN AAAA BBBB

とTTYからタイプアウトする．ここで，Fはデータ・フィールド，Nは番地，Aは既存のデータ，Bは新しいデータである．タイプアウト終了後に，FフィールドのN番地にBをストアする．

手順9． 照合がわずらわしくなったときは，いつでもスイッチ・レジスタの最上ビットを1にして照合をやめることができる．また再開もできる．

テープの最後で，チェック・サムを8進4桁でタイプアウトする．0000がタイプされれば，正しくロードされたことを示す．0以外の数字がタイプされた場合はエラーである．エラーの発生原因

は，①操作ミス，②テープの破損，③ローダがこわれている，④読取機がこわれている，⑤CPUがこわれているの順に疑ってみる．

テープがよくないときは，マスターテープがあればDUPEプログラムで複製する．ソーステープがあれば，アセンブル（またはコンパイル）しなおす．どちらもないときは，のりとはさみでテープを補修してから，DUPEプログラムで複製してマスターテープをつくる．

エントリの指定があれば，チェック・サムのタイプアウト後，指定された番地にコントロールを移すので，その先の手順は，ロードしたプログラムの説明書によるが，次のステップだけは必要である．

手順10．　ロードし終ったテープは巻きもどして所定の場所に収納する．

エントリの指定のないテープでは，PC＝7777を示してCPUは停止する．エントリ指定のあるテープでも，エラーが発生したときは停止することもある．

手順11．　続いて他のテープをロードしたいときは，テープをセットして，CONTINUEを押すだけでよい．

この場合，フィールドは指定のない限り，最後にロードしたフィールドのままになる．何本かのBINテープを続けてロードすることが多いときは，中間にブランクを入れずに，テープをつないでおけば，テープをセットする手順が省けて，CONTINUEを押すだけでよい．

9.　XRIMローダによるローダのローディング

ローダは，メイン・メモリの最終ページ（7600—7777）に常駐させておくが，何らかの理由で破壊されたときには，ローダをロードする必要が生ずる．

手順1．　ローダの紙テープをさがす．

ローダの紙テープは，前半がXRIM形成，後半がBIN形成で，中央部に可視ラベルが入っている．

手順2．　テープを読取機（TTYまたはPTR）にかけて，オンライン状態とする．

テープは，ブランク・リーダの位置から読み始めるようにセットしなければならない．

TTY用

スイッチレジスタ	操　作
0001	LOAD PC
6031	WRITE
5001	WRITE
3013	WRITE
6036	WRITE
1013	WRITE
7434	WRITE
3412	WRITE
7106	WRITE
5001	WRITE
7577	WRITE
0001	LOAD PC
	START

PTR用

スイッチレジスタ	操　作
0000	LOAD PC
6014	WRITE
6011	WRITE
5001	WRITE
6016	WRITE
7430	WRITE
3410	WRITE
7106	WRITE
5001	WRITE
7577	WRITE
0000	LOAD PC
	START

手順3.　ローダをロードしたいフィールドに，IFとDFをセットする．

一般にはIF＝DF＝0でよい．

手順4.　イニシアル・ローダをスイッチ・レジスタからロードして，STARTさせる．

読取機がテープを読み始め，テープが走り出す．

BIN形式の部分に移り，タイトルをTTYからタイプ・アウトする．最後に0000をタイプアウトする．PC＝7777を示してCPUが停止する．

以上の一連の動作が正常に進めば，ローダが7600—7777にロードできただけでなく，そのローダを使って，BIN形式のローダが自己ロードされ，照合されて，機能が確認されたことになる．

エラーがあれば，手順1から繰返す．

マルチ・フィールドのCPUでは，複数のフィールドにローダを常駐させておきたい，そうすれば，1つのフィールドのローダが破壊されても，ローダ・テープの後半のBIN形式の部分を用いて，別のフィールドのローダによって，第2章の手順で，ローダを復元できるので，XRIMローダをスイッチ・レジスタからロードする手間が省ける．

（参考）　これまで使われてきたローダとの比較表を次にかかげる．

NBINローダは，RIMと共存できないこと（リエントラント可能なイニシアル・ローダをもたないこと）を除いてBIN，CBN，V，の各ローダより優れている．

XRIMローダは，RIMと比べてリエントラントでないこと，HELPと比べてテープが長いことの2点を除いて，RIM．HELP等のイニシアル・ローダより優れている．

	B I N	C B N	V	N B I N
チェックサム・プリント	×	×	○	○
タイトル・プリント	×	○	○	○
入力自動選択	×	○	○	○
オート・エントリ	×	○	×	○
照　　合	×	×	○	○
5桁番地	×	×	×	○
RIMとの共存	○	○	×	×
7777Start	○	○	○	○
7600Start	×	×	×	○

	R I M	H E L P	X R I M
リエントラント	○	×	×
ローダ本体のエリア	×	○	○
ステップ数	×	○	○
テープの読みやすさ	○	×	○
テープの長さ	×	○	×
テープのTTY・PTR互換性	○	×	○
自動停止	×	○	○
自動チェック	×	×	○

§2 エディタ

1. 概 要

エディタは入出力タイプライタ（以下TTYと略），高速紙テープリーダ（PTR），高速紙テープパンチ（PTP）などを利用してプログラムのソーステープを作製または修正する作業を容易にするためのプログラムである．

エディタを使用するソーステープの作製はオフラインのテレタイプによる場合にくらべて，操作手順の簡易化や，ミスタッチの修正の容易さなどの利点がある．さらにいったん作製したソーステープをその後修正する場合に，オフラインのTTYによる作業では，修正箇所を発見するだけでも容易でないが，エディタによる作業では，これが極めて容易になっている．エディタはソーステープ中の任意の箇所で修正，除去，付加，順序の交換などを自由に実行できる．

2. エディタの動作

2—1 コマンドモードとテクストモード

エディタの動作はコマンドモードとテクストモードの2つに大別される．コマンドモードではテレタイプからの入力はすべてコマンド（エディタに対する指令）として解釈される．エディタを最初にスタートさせた時はコマンドモードである．あるコマンドをTTYから入力すると，エディタは通常テクストモードに変り，以後のTTYからの入力はソースプログラムの一部として，バッファに付け加えたり，挿入したりする．テクストモードでFFコード（CTRL/FORM）の入力があると，エディタはベルを鳴らして，コマンドモードに戻る．

コマンドモードで正しくない入力（コマンドの規定に合致しないもの）があるとエディタは？を印刷，改行して次のコマンドを待つ．コマンドはすべてRETURNコード（オクタル215，以後↵で示す）を入力して，始めて実行に移される．

2—2 入力コマンド

2—2—1 READ

R↵テクストモードに変り，またはPTRからソースプログラムを入力してバッファに付け加える．TTYまたはPTRの選択は，PTRにソーステープが用意されていれば，PTRが選択され，そうでなければTTYが選択される．FFコード（オクタル214CTRL/FORM）が入力されるとコマンドモードに戻る．またPTRが選択された時は紙テープの終りでもコマンドモードに戻る．

PTRにソーステープをかけても，ON—LINEにしないでRコマンドを発した時は，エディタはTTY入力を待ち受ける状態が選ばれているので，PTRをON—LINEとしても受付けない．この時はTTYからFFを入力してコマンドモードに戻してから，あらためてRコマンドを実行させなければならない．

入力コードの中の特殊コードについてはRUBOUT（オクタル 377）が無視される点を除いてテクストモードの通則（3—3参照）に従う．

オプション2，3，または6が選択されていれば，2箇以上連続するスペース（オクタル240）はTAB（オクタル211）に置換えられる．

1R↵ 入力ディバイスとしてPTRの有無にかかわらずTTYが選ばれる．

2—2—2 APPEND

A↵テクストモードに変り，以後のTTYからの入力をバッファに付け加える．FFコードでコマ

ンドモードに戻る．　/は最後の行，・は現在の行とすると　n行付け加えた時　/＝/＋n，(・＝/）となる．

2－2－3　挿入

I↙　テクストモードに変り，以後のTTYからの入力をバッファの前につけ加える．FFコードでコマンドモードに戻る．n行挿入した時，従来の1～l行はn＋1～l＋n行となり，・＝n，＝/＋nとなる．

nI↙　テクストモードに変り，以後のTTYからの入力は第n行以前につけ加えられる．FFコードでコマンドモードに戻る．m行挿入した時　・＝n＋m－1，/＝/＋m，従来のn～l行はn＋m～l＋m行となる．

2－3　出力コマンド

2－3－1　パンチ

T↙　約10cmのブランクテープをパンチする．TTYの場合は↙を入力したら直ちにPUNCH ONとし，パンチが終ったら次のコマンドを入力する前にPUNCH OFFとする．

1T↙　パンチ出力をTTYにしてTコマンドを実行する．最初にエディタをスタートした時はこの状態になっている．

2T↙　パンチ出力をPTPに指定してTコマンドを実行する．次に1T↙を行うまでパンチ出力はすべてPTPに出る．

P↙□　バッファの内容をすべてパンチする．パンチ後・＝1

nP↙□　第n行のみパンチする．パンチ後・＝n

m，nP↙□第m行から第n行までパンチする．パンチ後　・＝n

他のコマンドと異なり↙の後に□（スペース，実はどのキーでも同じである）を入力して始めてパンチを開始する．これはTTYをPUNCH ONとする余裕を与えるためで，↙後PUNCH ONとして次に□を入力するとパンチが始まる．TTYではパンチ終了後次のコマンドを入力する前にPUNCH OFFとする．

各CR後にはLFが付加されてパンチされる．オプション1，3，5，または7が選択されていれば，この後にさらにブランク（オクタル0）が4箇ずつ付加される．オプション4，5または6が選択されていればTAB（オクタル211）は8字相当分のスペースに置換えられてパンチされる．その他の場合にはTABはTAB/RUBOUTとせん孔される．このRUBOUTは時間調整のために付加されているものでコマンドRで読込む際は無視される．

F↙　ブランク4箇　FFコード（オクタル214），約5cmのブランクをパンチする．TTYの場合は↙後直ちにPUNCH ONとし，パンチ終了後PUNCH OFFとする．FFコードはいったんパンチしたテープを次にエディタで読込む場合，自動的にコマンドモードに戻すために必要である．

N↙□　P↙F↙K↙R↙の組み合わせをこの順に実行する．K↙は後述するがバッファの内容をクリアするコマンドである．□についてはPコマンドに同じである．

nN↙□　上述の動作を連続してn回実行してコマンドモードに戻る．nが入力テープのブロック数よりも多い時は最後のブロックをパンチした後，入力がPTRの場合は自動的にコマンドモードに戻る．TTYの場合はFFコードを入力させてコマンドモードに戻す．

2－4　修正コマンド

2－4－1　LIST

L↙　バッファの内容をすべて印刷する．印刷後・＝1

nL↙　第n行を印刷する．印刷後・＝n

m，nL↙　第m行から第n行まで印刷する．印刷後は・＝n

TABは8字ごとにTABが設定されているかのようにスペースに置換えて印刷される．

2—4—2 GET

nG↙ 第n行から順にチェックしてタグまたはオリジンで始まる行（最初の1字が/，TAB，SPACEでない行）を印刷する．・はその行の値に等しくなっている．

G↙ ・＋1 G↙に等しい．次のタグまたはオリジンで始まる行を印刷する．

2—4—3 抹消

K↙ バッファをクリアする．実行後・＝/＝0

nD↙ 第n行を抹消する．第n＋1行以下は順次繰上げられる．実行後 ・＝n－1，/＝/－1

m，nD↙ 第m行から第n行まで抹消する．第n＋1行以下は順次繰上げられる．実行後 ・＝m－1，/＝/＋m－n－1

2—4—4 変更

nC↙ nD↙と，nI↙の複合動作

m，nC↙m，nD↙，mI↙の複合動作

2—4—5 行の移動

m，n$kM m～n行のテクストをk行の前に移動．

m＝nならば1行だけ移動される．

k＝/＋1ならばバッファの最後に移動．

各行の№はつけかえられるが・の値は変化しない．

2—4—6 修正

nS↙ 第n行について指定する字まで複製する．

m，nS↙ 第m行から第n行までについて指定する字まで複製する．

それ以降については，その行の内容をさか上って抹消する，新しく字を挿入する，その行の以降の内容を捨て，新しい行として作る，さらに新しい字を指定して続行するなどの処置が可能であるが，詳細は2．の操作の項を参照されたい．

S↙ バッファ内のテクストをバッファ内の先頭に整理して並べ直す．

3. 操 作

3—1 ロード

VERIFYローダでコアに読込み，SA＝200番地からスタートさせる．200番地からスタートさせた場合はオプション0が選ばれる．オプション指定を用いる場合は176でスタートさせ，OPT：と印刷されたら0～7のオプションのいずれかを入力する．0～7以外を入力すると再びOPT：と印刷する．

オプション	テープ出力時に各行の後にブランク4個をせん孔	入力時に連続したスペースをTABに変換	テープ出力時にTABをスペースに変換
0	×	×	×
1	○	×	×
2	×	○	×
3	○	○	×
4	×	×	○
5	○	×	○
6	×	○	○
7	○	○	○

何らかの理由でエディタを再スタートさせたい場合はSA＝177 で再スタートさせればよい．SA＝176 で再スタートさせるとオプションの再指定ができる．

3－2 入力

エディタのバッファにテクストを入力するには，(1)キーボード，(2)高速リーダから紙テープ，(3)テレタイプより紙テープの3つの場合がある．

3－2－1 キーボードからのテクスト入力

コマンドIまたはAを用いる．Iは現在のバッファ内のテクストの前に付け加え，Aは現在のバッファ内のテクストの後につけ加える．バッファが空の時はどちらも同じである．またnIとすれば第n－1行と第n行の間に新しいテクストが付け加えられ，第n行以下は順次繰り下げられる．各行は↙を入力し，始めてバッファに繰込まれる．RUBOUTは1箇について最終入力文字から順次1字ずつ消去して，各々\を印字する．その行が空になった以降のRUBOUTは無視され\は印字されない．← (SHIFT O) はその行のすでに印刷した部分を抹消する．

LINE FEEDは無視される．

FFコードでコマンドモードに戻る．

2－2－2 高速リーダ，またはTTYからの紙テープによるテクスト入力

コマンドRを用いる．現在のバッファ内のテクストの後に紙テープから読んだテクストを付け加える．この場合，高速紙テープリーダにソーステープがセットされていれば，高速紙テープリーダが選ばれ，セットされていない場合や，セットされていても，オンライン状態になっていなかった時はTTYが選ばれる．もし誤って高速紙テープリーダを選べなかった時はTTYからFFコードを入力することによってコマンドモードに戻る．

入力として高速紙テープリーダが選ばれた場合は，FFのコードを読んだ時およびテープが終りになった時にコマンドモードに戻る．

コマンドRとコマンドAの違いは次の通りである．

1．コマンドRではTTYへの印刷（エコー バック）は行われない．

2．コマンドAでは入力はTTYに限られている．

3．コマンドRではRUBOUT（377）コードは無視される．

高速紙テープリーダに紙テープがセットされている時にTTYからの紙テープ入力を受付けるに

は1Rのコマンドを用いる.

3－3 テクストの修正

3－3－1 全　般

テクストの修正は行単位または連続したいくつかの行に対して行うことができる. 問題の行はその行の番号によって指定できるが, これを能率的に行うためにいくつかの便利な補助手段が用意されている.

1. 行の指定は行番号だけでなく, 現在修正している行の値を・で表わし・からの相対表示ができる. たとえば5L↙としての第5行目を印刷させると・＝5となるから・－2L↙とすれば3L↙と同じになり・5L↙または・＋5L↙とすれば10L↙と同じになる.
2. 行の指定は最終行の値を/で表わし/からの相対表示ができる. たとえば/＝125の時に/－3L↙は122L↙と同じである.
3. nG↙とすれば, n行から順にチェックして最初のTAGのついた行を印刷し, .はその行の値となる. またG↙は, .G↙と同じである.

3－3－2 一つまたは連続した行の修正（Sコマンド）

1. nS↙ 第n行をサーチする. RETURNを押した後でサーチする文字をTTYから入力する. この文字は印刷されず, その行のテクストを, その指定された文字まで印刷してからテクストモードになる.
2. RUBOUTを入力するとRUBOUT1箇について\を1箇印刷し, テクストの印刷されている部分について後から1字ずつ抹消する. その行のテクストが空になれば\は印刷されず, その前の行に抹消は及ばない.
3. ← を入力するとその行のすでに印刷された部分がすべて抹消される.
4. CTRL/FORM を入力すると, 指定された文字が次に出現するまで印刷する.（サーチ続行）
5. CTRL/BELL を入力し, 続けてサーチ文字の変更ができる. 入力した文字は印刷されず, 指定された文字まで印刷されて, コマンドモードに戻る.
   （新しい指定によるサーチ続行）
6. RETURNを入力するとその行の残りの部分が抹消される.（その行についてのサーチ終り）
7. LINE FEEDを入力すると, その行のすでに印刷された部分を1行とする.
   残りの部分はさらに次の行として付け加えられ, サーチ続行が可能である. ただし・の値は最初の値のままで変化しない.
8. ESC を入力するとその行の残りの部分はRETURNまでそのまま付け加えられる(その行についてのサーチ終り).
9. その他の文字を入力すると, そのままテクストのその部分に付け加えられる.
10. Sコマンドはn, mS↙のように第n行から第m行までについて一括して上記の操作を行うことができる.
11. S↙は, バッファ内のテクストを並べ直す. バッファに現在のテクストの大きさに等しいだけの余裕が必要である.

3－3－3 一つまたは連続した行の抹消（Dコマンド）

nD↙ 第n行を抹消, この時 /＝/－1となり, 第n＋1行～最終行はいずれも順次送られる. ・はn－1に変化する.

m, nD↙ 第m行から第n行まで抹消. この時/＝/＋m－n－1, ・＝m－1, 第n＋1行以下は順次繰上げられる.

3－3－4 新しい行の挿入（Iコマンド）

nI↙ 第n行の前に新しい行を何行でも挿入できる. RETURNを入力した時にはじめて新しいテ

クストは有効となる.

FFコードでコマンドモードに戻る.

Ｉ↙　は１Ｉ↙に同じである.

最終行の後に新しい行を設けるにはＡコマンドを用いる.

3－3－5　行の内容の変更（Cコマンド）

nC↙　nD↙を行ってからnI↙を行なうのと同じである.

m，nC↙　m，nD↙を行ってからmI↙を行なうのと同じである.

いずれでも抹消した行数と新しく挿入する行数とは無関係である.

3－3－6　行の順序の変更（Mコマンド）

m，n＄kM↙　第m～n行のテクストを第k行の前に移動させる．各行の番号は付け変えられるが・の値は変化しない.

m，n＄／１M↙　第m～n行はテクストの最後に移動する.

3－3－7　バッファ全体の抹消（Kコマンド）

K↙　バッファの内容はすべて抹消され　．＝／＝0となる.

3－4　出力

3－4－1　出力ディバイスの選択（Tコマンド）

１T↙　出力ディバイスをTTYとしトレーラをパンチする.

２T↙　出力ディバイスをPTPとしトレーラをパンチする.

最初にエディタのプログラムテープをロードした時はTTYが選ばれている.

3－4－2　トレーラ（Tコマンド）

T↙　約10㎝のブランクテープをパンチする．TTYの場合は↙を入力して直ちにPUNCH ONとし，パンチが終ったら次のコマンドを入力する前にPUNCH OFFとする.

3－4－3　本文のパンチ（Pコマンド）

P↙□　バッファの内容をすべてパンチする．パンチ後　・＝/

nP↙□　第n行をパンチする．　パンチ後　・＝n

m，nP↙□　第m行から第n行までをパンチする．パンチ後　・＝n

他のコマンドと異なり，↙の後に□（スペース，実はどのキーでもよい）を入力して始めてパンチを開始する．これはTTYについてPUNCH ONとする時間の余裕を与えるためで，↙後　PUNCH ONとしてから次に□を入力するとパンチが始まる．TTYではパンチ終了後，次のコマンドを入力する前にPUNCH OFFとする.

3－4－4　FFコードのパンチ（Fコマンド）

F↙　ブランク4箇，FFコード（オクタル214），約5㎝のブランクをパンチする．TTYの場合は↙後直ちにPUNCH ONとし，パンチ終了後，次のコマンドを入力する前にPUNCH OFFとする.

3－4－5　複合コマンド（Nコマンド）

N↙□

P↙F↙K↙R↙の組合わせをこの順に実行する．□についてはPコマンドに同じである.

nN↙　上述の動作を連続してn回実行してコマンドモードに戻る．入力がPTRの場合は，テープが終りになれば自動的にコマンドモードに戻るから，nとして実際のブロック数よりも大きい数を指定してやれば入力テープの最後までコピーできる.

4.　まとめ

4－1　コマンドーらん

A……TTYのキーボードからテクストを入力してバッファに付け加える.

nC……第n行を抹消してキーボードからのテクストで置きかえる.
m、nC……第m行から第n行までを抹消してキーボードからのテクストで置きかえる.
nD……第n行を抹消
m、nD……第m行から第n行を抹消
F……FFコードパンチ
G……次のTAGのついた行を印刷
nG……n行の後で次のTAGのついた行を印刷
I……第1行の前にキーボードからのテクスト挿入
nI……第n行の前にキーボードからのテクスト挿入
K……バッファ クリア
L……バッファを全部印刷
nL……第n行を印刷
m、nL……第m行から第n行までを印刷
m、n$kM……第m行から第n行までを第k行の前に移動
N……P、F、K、R
nN……n回, N動作を行う
P……バッファ パンチ
nP……第n行パンチ
m、nP……第m行から第n行パンチ
R……TTYまたはPTRからテクストを読んでバッファに付け加える
1R……TTYからテクストを読んでバッファに付け加える
S……バッファのテクスト整理
nS……第n行サーチ
m、nS……第m行から第n行サーチ
T……トレーラパンチ
1T……TTYを選択してトレーラパンチ
2T……PTPを選択してトレーラパンチ

4—2 特殊キー

4—2—1 コマンドモード一般

RETURN……コマンド実行
. ……現在の行の値
/……バッファの最終行の値
LINE FEED…次の行を印刷. .1Lに同じ.
ESC……次の行を印刷. .1Lに同じ.
>……次の行を印刷. .1Lに同じ.
<……前の行を印刷. .—1Lに同じ.
=……. または/の値を調べる.
:……. または/の値を調べる.

4—2—2 コマンド モード（サーチ コマンドの中）

RUBOUT……1字ずつ抹消（\エコー）
RETURN……その行の残りを抹消
LINE FEED…その行の残りを新しい行とする.
CTRL/U……その行のすでに印刷した部分を抹消 (SFIFT O)

ＥＳＣ……その行の残りの部分をそのまま生かす.
ＣＴＲＬ／ＦＯＲＭ…サーチ続行
ＣＴＲＬ／ＢＥＬＬ…サーチする文字を再指定する.

4－2－3 テクスト モード

ＲＥＴＵＲＮ……その行のテクストをバッファに登録
ＣＴＲＬ／Ｕ……その行の印刷した部分を抹消
ＲＵＢＯＵＴ……１字ずつ抹消 （＼エコー）
ＣＴＲＬ／ＦＯＲＭ…コマンド モードに戻る.
ＣＴＲＬ／ＢＥＬＬ…コマンド モードに戻る.
ＥＳＣ……コマンド モードに戻る.
ＬＩＮＥ ＦＥＥＤ…無視

§3　アセンブラ

1. 概　要

SAP—Ⅱ（Symbolic Assembly Program versionⅡ）は，4K語のメモリを持ったCEC555によって，シンボリックソースプログラムを，バイナリテープに変換するためのアセンブラである．変換はソーステープを2回読むことによっておこなわれる．第1回目には，コア上のプログラムの番地割当てと，プログラムに含まれたシンボルに対する値の指定とをおこない，第2回目には，ローダによってコアに読み込める形のバイナリテープと，リスティングとを出力する．出力機器が入出力タイプライタのみの場合は，第2回目には，バイナリテープのみが出力され，第3回目に，リスティングを出力する．

アセンブラに対する周辺機器として，入力は，入出力タイプライタまたは高速紙テープリーダ，出力は，入出力タイプライタ，高速紙テープパンチャ，高速ラインプリンタを使用できるが最少構成では入出力タイプライタのみでもよい．

CEC555は1ページ128語構成のページ式の小型電子計算機であるが，SAP—Ⅱは，このためのプログラミング上の制限をできるだけ緩和し，単なるシンボリック言語およびシンボリック番地によるプログラミングという基本機能の他に，次のような高度の機能をそなえている．

1．直接指定できるページの外の番地に対する間接指定を自動的におこなう．
2．プログラム中の定数指定を0ページまたは現在ページの最後から自動的に設定する．
3．文字のASCIIコード値の自動設定
4．文字ストリングのASCIIコード値を6ビットに圧縮して2字／1語とした連続したバイナリデータの自動設定
5．ローダに対する指定として自由なプログラム名のバイナリテープへの転送，これはバイナリテープのラベル（認識票）として極めて有効である．
6．ローダに対する指定としてのプログラムエントリ番地のバイナリテープへの転送
7．タブコードのプログラムによる翻訳
8．リスティングにおけるページ割付け
9．ソース言語で書かれたライブラリサブルーチン各種が用意されており，それらはユーザプログラムに自由に結合ができる．

2. 使用する文字

SAP—Ⅱは次のASCII文字を使用する．

1．英字　A～Z
2．数字　0～9
3．特殊文字
(1)　印字のあるもの

＋	オクタル	253
－		255
,		254
.		256
＝		275

*	252
;	273
／	257
$	244
#	243
(	250
〔	333
!	241
〃	242

(2) 印字のないもの、　説明中の表示

スペース	240	␣
タブ	211	→
復帰	215	↵

(3) 無視するもの

FORM	214
空白	000
抹消	377
改行	212
フィード	200
シフト	216
シフト	217

上記以外のすべての文字は、次の4つの場合を除いて、すべて禁止される.

1．コメント内（/記号の後）
2．テクスト内（TEXTの後の文字ストリング）
3．ラベル内（#記号の後）
4．ASCII文字（〃記号の直後）

アセンブラは不当に使用された文字に対しては……

IC　××××　AT　nnnn

とエラー表示を印刷する．ここで，××××は不当に使用された文字のオクタル表示，nnnnはこれの発生した時のプログラムカウンタのオクタル表示である.

3．区切り文字

＋　－　，　．　＝　！　；　／　スペース　タブ　復帰の11種類の文字はひとつの数またはシンボル記号の終りを示す.

さらに＋，－，スペース，タブは演算子としての機能をあわせて持つ．すなわち，前後のシンボルの値の和(+)，差(-)，および論理和（スペースおよびタブ）を計算する．演算子の前のシンボルが番地参照命令コードの場合は演算子の種類にかかわらず論理和となる．2つ以上の演算子が重複した場合は，最後の演算子が優先する．・は区切り文字であると同時に，その時のプログラムカウンタの値に等しいシンボルでもある．区切り文字としての機能はスペースに等しい．；および復帰は区切り文字であると同時に一つのアセンブルの区切りをも兼る.

4．数　値

数字の連続を数値と呼ぶ．通常はオクタル変換をしたバイナリ値が取られるが，後述のDECIM

AL疑似命令の後では10進変換をしたバイナリ値が取られる.

DECIMAL疑似命令は次にOCTAL疑似命令が発行されるまで，あるいは一つのパスの終了まで有効である.

バイナリ値については，10000（10進 4096）のモジュロが取られる．負の数は2の補数表示である.

本説明書で使用する数値は特別の指定がない限りオクタル数とする.

オクタル変換の場合，8，9は各々オクタルの10，11と解釈される．数字の中に，英字，特殊文字が混入するとICエラーとなる.

IC××××，AT nnnn，××××は英字または特殊文字の8ビットASCII値，nnnnはこの時のプログラムカウンタのオクタル表示である.

5．シンボル

英字で始まる英字または数字の連続をシンボルと呼ぶ．6字以上連続する場合は7字以降は区切り文字の前まで無視される.

シンボルは次の4種から成る.

1．疑似命令（付録A参照）

2．番地参照命令（付録B参照）

3．その他の命令（付録C参照）

4．1.，2.，3.に含まれないものでプログラムでその値を設定したもの

（ユーザのシンボル）

3．項の命令はFIXTABの疑似命令で増加設定することができる．4．項のシンボルの値の設定は，，および＝の各疑似命令によっておこなう．4．項のシンボルで値を設定しないで使用されたものは，パス1の終りに，US××××××　AT　nnnnの形でエラー表示がおこなわれる．××××××は使用されたシンボルでnnnnはそれが最初に用いられた時のプログラムカウンタの値である．値を設定しなかったシンボルの値はパス2,3では0として取扱われる．1.，2.，3.項で設定してあるシンボル記号，またはすでに値を設定したシンボル記号をユーザのプログラムでさらに値を設定した場合，＝および，，疑似命令に対して各々

(1) RD××××××　AT nnnn

(2) ID××××××　AT nnnn

の形のエラー表示がおこなわれる.

（注）SAP-DではUS表示はパス2で引用されたその都度おこなわれる.

6．ASC II文字，TEXTストリング

〃記号と任意の文字1字の組み合わせをASCII文字と名づける．ASCII文字の値は〃記号の後の文字のASCIIコードのオクタル値（8ビット）とする．〃の後の文字は何を使ってもよいが，無印字のコードは使用しない方が望ましい.

TEXT疑似命令はそれに続くASCIIストリングを6ビット／字，2字／語にパックしてアセンブルする．ストリング中の文字は印字（240—337），無印字（201—237，340—376）を問わず，そのオクタル値の6LSBをパックするので無印字の文字が含まれないように注意を要する．TEXTの次の文字（スペースまたはタブ）は区切り文字，その次の文字は終端文字で，再びこの文字が現われるまで（終端文字は除く）をアセンブルする．2回目の終端文字はその行に対するCRで兼用しても良く省略できる．文字数が奇数個の場合は最後の半端には0が詰められる．文字数が偶数個の場合には最後の語の後にさらに0000が1語付加される.

例

```
*DEMO  .TM  2015 *
#DEMO, ASCII CONVSTANT & TEXT STRING:  APR 1, 1975


                        PAGE
                /ASCII CONSTANT
0200 0303
0201 0305
0202 0303               "C:"E:"C
                /TEXT STRING, '/' IS TERMINATER
0203 0305
0204 0300       TEXT    /CEC/
                /EVEN CHARACTERS GENERATES AN EXTRA '0'
0205 0305
0206 0365
0207 6565
0210 0000       TEXT    /CEC555/
                /2ND TERMINATING CHAR, '?' MAY BE OMITTED
0211 0305
0212 0365
0213 6565
0214 0000       TEXT    ?CEC555
                /ANYTHING AFTER 2ND TERMINATER IS NEGLECTED
0215 0305
0216 0300       TEXT    ?CEC?555
                /BAD EXAMPLES
0217 0500       TEXT    CEC     /TERMINATER IS MISSING, 'C' STANDS FOR IT
0220 0500       TEXT    CEC     /SPACES IN STEAD OF TAB
0221 0305
0222 0311
0223 0000       TEXT    /CEC    /       /TAB IS ASSEMBLED AS '11' OR 'I'
```

$

7. 定数

(または〔で始まる表現を定数と呼ぶ．アセンブラは，(または〔の右の表現を評価してその値を0ページ定数表および現在ページ定数表に登録してあるかどうかをチェックして，もし登録してなければ登録する．ただし〔の場合は0ページ定数表のみに関係する．

(または〔を含む行のアセンブルに関しては，(または〔で始まる表現は定数表に登録された番地を値として処理される．

現在ページ定数表はそのページの最後の番地から順にさかのぼって確保される．0002, 7766, 0301の3つの数値は下記の例では各々，377，376，375，番地に格納され，アセンブラは次のリスティングに示すように処理をおこなう．

```
#DEMO. LITERAL CONSTANT:  JAN 1. 1975


                PAGE
0200 7200               CLA
0201 1377               TAD (2
0202 3210               DCA X
0203 1376               TAD (-12
0204 3211               DCA Y
0205 1375               TAD ("A
0206 3212               DCA Z
0207 7402               HLT
0210 0000       X,      0
0211 0000       Y,      0
0212 0000       Z,      0

0375 0301
0376 7766
0377 0002


   X        0210    Y        0211    Z        0212
$
```

定数表データは，*疑似命令，PAGE疑似命令，FIELD　疑似命令，$疑似命令などで，バイナリテープおよびリスティングに出力される．

定数を現在ページでなく0ページにとりたい時は，(の代りに〔を使う．また0ページ定数表は!疑似命令によっても設定できる．!疑似命令を使用すると本文のプログラムは同一のままで定数を共通に0ページにとることも現在ページに別々にとることも指定できる．各ページに許される定数の数は最大128個である．なるべく[よりも!を使う方がよい．

```
#DEMO. LITERAL CONSTNT AT ZERO PAGE:  JAN 1, 1975

2!      -12!    "A!.



          PAGE
0200 7200           CLA
0201 1177           TAD (2
0202 3210           DCA X
0203 1176           TAD (-12
0204 3211           DCA Y
0205 1175           TAD ("A
0206 3212           DCA Z
0207 7402           HLT
0210 0000     X,    0
0211 0000     Y,    0
0212 0000     Z,    0



0175 0301
0176 7766
0177 0002



  X       0210    Y       0211    Z       0212
$
```

8．プログラムカウンタ

SAP—Ⅱで書かれたソースプログラムは，原則として，１行が計算機メモリの１語に翻訳され，連続した行はメモリの連続した番地に割当てられる．メモリのどの番地にアセンブルした結果を割当てるかを指定するレジスタが，SAP—Ⅱの中に用意されているが，これをプログラムカウンタと呼び，その値を・の記号で表す．プログラムカウンタは*疑似命令または，PAGE疑似命令で任意に指定できる．プログラムカウンタの値はSAP—Ⅱが一語のアセンブルされた結果を処理し終る度に１ずつ増加される．プログラムカウンタの値は正で，10000のモジュロが取られる．

プログラムカウンタの値と定数表ポインタとが重なった場合はPEエラーとなり

PE×××× AT nnnn の表示がなされる．

プログラムカウンタの値の指定をおこなわなかった場合は，プログラムカウンタは200に指定されており，*200があった場合と同様になる．

＊疑似命令

＊に続く表現の値を解釈してそれをプログラムカウンタにセットする．プログラムカウンタが新しいページに移る場合には，現在ページ定数表をリセットする．パス２では現在ページ定数表と新しいプログラムカウンタのオリジン指定をバイナリテープに出力する．また定数表をリスティングに印刷する．

PAGE疑似命令

現在ページ定数表をリセットし，プログラムカウンタの右７ビット（LSB）が０でない場合はプログラムカウンタを次のページの最初の番地にセットする．パス２では現在ページ定数表をバイナリパンチし，リスティングに印刷する．

例

```
#DEMO. * & PAGE PSEUDO-OP:  JAN 1. 1975



*400
        PC1=.



PAGE
        PC2=.
*.+5
        PC3=.



PAGE
        PC4=.
```

```
  PC1     0400    PC2     0400    PC3     0405    PC4     0600
$
```

9. ラベル

SAP—Ⅱアセンブラの出力はバイナリテープ（オブジェクトテープ）とリスティングである．バイナリテープは通常その区別が容易でなく，従来は人間が手でメモを書込んだりラベル用紙に書込んだものを貼ったりしたことが多かったが，しばしば重大な間違いを生ずる原因となっていた．

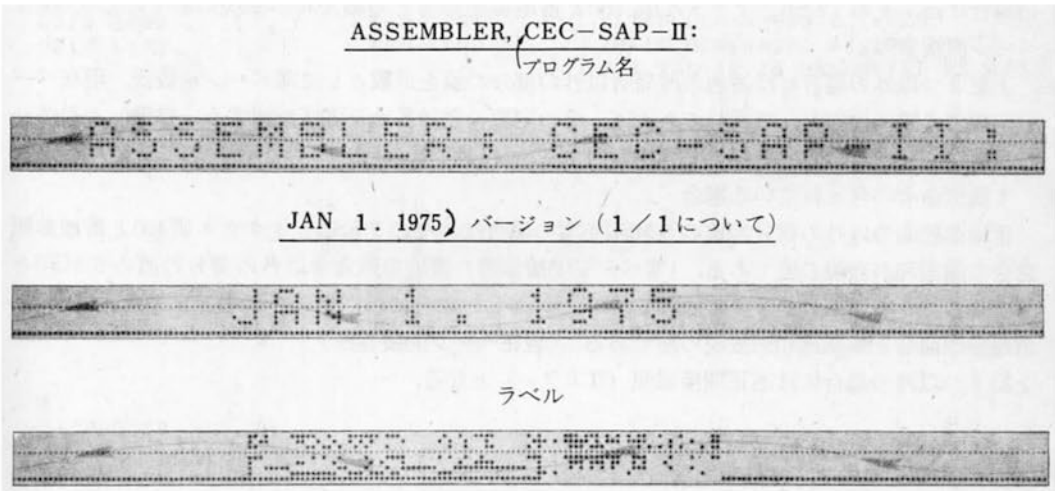
SAP—Ⅱではこの点を解決するために，アセンブラ出力のバイナリテープに自動的にラベルを付けるための便利な手段を用意してある．バイナリテープに付けられたラベルはバイナリテープをそのままオフラインの入出力タイプライタで印字させて内容を読むことができるだけでなく，ローダで自動的に印字させたり，種々の目的に利用される．

ラベルを生成するためのラベル行は＃記号で始まり／記号で終る．1行以上にわたる時は各行毎に＃記号を必要とする．ラベル行はアセンブラに対するあらゆる入力の前に位置しなければならない．

SAP—Ⅱのパス1でラベル行はそのまま入出力タイプライタにコピーされる．これはアセンブラのエラー表示についてそれがどのソースプログラムに対するものかを示すために有効である．パス2ではバイナリテープに，リーダ，見出し，リーダ，抹消コード，CR—LFを出力した後，ラベルの内容をそのままコピーし，最後に抹消コードを出力して終る．

ラベルを有効に活用するためには内容の少しでも異なるソーステープに同じラベル行をつけないように厳重に注意することが必要である．これはラベル行の内容としてプログラム名の他に，日付，バーション番号などを詳細に用意することによって解決できる．

例



10. 表　現

数，シンボル，ASCII文字，定数，・（プログラムカウンタの値）を区切り文字で連結したものを表現と呼び，表現はそれの持つオクタル値を代表する．

11. オブジェクトプログラムのコア割付

7．定数の説明で述べたようにSAP—Ⅱアセンブラでは疑似命令で自由に定数を現在ページに設定できる．

またこの現在ページ定数表には 12. 番地参照命令の処理で説明されるように，アセンブラが自動

的に設定した間接参照のためのベクトルも含まれている.

これらの定数はアセンブルしているプログラムがロードされるべきコアページの最終番地から逆に次々とコアが割当てられる. 一方プログラムは*疑似命令またはPAGE疑似命令によって設定されたオリジンから順に次々とコアが割当てられる. PAGE疑似命令によって現在ページ定数表は出力されクリアされる. *疑似命令によって番地が新しいページに移った時も同じである. 従って同一ページのプログラムを2個所以上に分けてコーディングするのは一般に危険である.

FIELD疑似命令および$疑似命令によって0ページ定数表は出力され, クリアされる. 従って同一フィールドのプログラムの間に異なったフィールドのプログラムをはさむのは一般に危険である.

12. 番地参照命令の処理

AND, TAD, ISZ, DCA, JMS, JMPの6種類の命令および浮動小数点演算インタプリータ, 3倍長演算インタプリータ用のFGET, FPUT, FADD, FSUB, FMPY, FDIV, FISZ, FJMP, FJMSの9種類の命令は番地参照命令としてSAP—IIの固定シンボル表に登録されている.

番地参照命令を含む表現のオクタル値の評価は, I疑似命令と番地参照命令とその他の部分とを別々に評価した後に次の規則に従って結合する.

1. I疑似命令の含まれない場合

   番地参照命令以外の部分の値の5 MSBが零の場合は, その7 LSBと番地参照命令の論理和が表現の値である. (零ページ直接参照)

   番地参照命令以外の部分の値の5 MSBとその時のプログラムカウンタの値の5 MSBとが等しい場合には, その7 LSBとオクタル値 200と番地参照命令との論理和が表現の値である. (現在ページ直接参照)

   上記2つ以外の場合には番地参照命令以外の部分の値を定数として零ページ定数表, 現在ページ定数表の順に検索し, 登録してあれば, その定数の登録番地を間接参照する. 登録してなければ, 現在ページ定数表に登録して間接参照する. (自動間接参照)

2. I疑似命令の含まれている場合

   番地参照命令以外の部分の値の5 MSBが零の場合は, その7 LSBとオクタル値400と番地参照命令の論理和が表現の値である. (零ページ間接参照) 番地参照命令以外の部分の値の5 MSBとその時のプログラムカウンタの値の5 MSBとが等しい場合には, その7 LSBとオクタル値600と番地参照命令との論理和が表現の値である. (現在ページ間接参照)

   上記2つ以外の場合には不正間接参照 (IIエラー) となる.

例

```
                    #DEMO. MRI:  JAN 1. 1975

                    CEC=     555
                    CDP=     552
                    CDP!
                    CS=      100
                    CF=      250
```

```
                    PAGE
0200 1250           BEGIN,  TAD CF
0201 1100                   TAD CS
0202 1577 0552              TAD CDP          /TAD I (CDP; CDP AT 177
0203 1577                   TAD I (CDP
0204 1777\0555              TAD CEC          /TAD I (CEC; CEC AT 377
0205 1777                   TAD I (CEC
0206 1752                   TAD I CDP        /"II ERROR"
0207 1755                   TAD I CEC        /"II ERROR"
0210 3000                   TAD+DCA          /TAD DCA
0211 3000                   TAD-DCA          /TAD DCA
0212 3000                   TAD DCA
0213 3400                   TAD I DCA        /(TAD DCA)=3000 & I=400
0214 1375                   TAD (TAD (2      /2 IS GENERATED AT 376
                                             / & TAD (2 IS GENERATED AT 375

0375 1376
0376 0002
0377 0555



0177 0552
```

```
BEGIN   0200    CDP     0552    CEC     0555    CF      0250
CS      0100
$
```

注(1) 番地参照命令とそれ以外の部分との結合子は＋や－でもスペースと見なされOR結合がとられる.

注(2) CDPはページ外であるが，CDPのデータは！で0ページの177番地にとられているのでこれ

を間接引用する.

注(3)　CECはページ外であるので377番地にデータを確保して間接引用する.

注(4)　TAD　I　(CECはCECがページ外ならばTAD　CECと同じである.

注(5)　2を376番地に作り、データTAD（2を375番地に作る.

13. その他のシンボルの処理

番地参照命令以外の命令および疑似命令およびユーザが定義したシンボルについては，＋（和），－（差），スペース（論理和）など各演算子の機能に従った値が計算される．

例

```
                    #DEMO. SYMBOL PROCESS:  JAN 1. 1975

                    CEC=    555
                    CFD=    532
                    CLP=    501



                    PAGE
0200 0577                   CEC CFD
0201 1307                   CEC+CFD
0202 0023                   CEC-CFD
0203 0555                   CEC CLP
0204 1256                   CEC+CLP
0205 0054                   CEC-CLP
0206 0206                   .
0207 0214                   .+5



0210 0211                   .1              /(.) OR (1)
0211 0211                   .1              /(.) OR (1)



  CEC     0555    CFD     0532    CLP     0501
$
```

14. マイクロ複合命令プログラミング

マイクロ複合命令については，アキュームレータの内容を反転したり，シフトしたりする第1類とアキュームレータの内容により，次の命令をスキップする第2類とにわかれている．第1類の命令，および第2類の命令の各々について，一つの類の命令相互間ではいくつかの複合動作が可能である．

555Hでは第1類，第2類の他に第3類の命令系を使用できる．555H以外の機種ではこれらの命令はNOP動作である．

第3類の命令系ではCLAとの複合動作が許され，CLAを先に実行する．

また，入出力命令中のJAP命令群は動作上はマイクロ複合命令群に類似である．

SAP—Ⅱは一つの命令語として，2個以上のマイクロ命令が指定されている場合，それらのOR結合のビットパタンを出力する．複合動作が可能かどうかについてはチェックしていないので，充分注意する必要がある．

第1類マイクロ複合命令の組合わせ

	CLL	CML	GMA	IAC	BSW	RAR	RAL	RTR	RTL	実行順序 555H	実行順序 555H以外
CLA	○	○	STA	○	△	○	GLK	○	○	1	1
	CLL	STL	○	○	△	○	×	○	○		
		CML	○	○	△	○	○	○	○	2	2
			CMA	CIA	△	△	△	△	△		3
				IAC	△	△	△	△	△	3	
					BSW	×	×	×	×	4	
						RAR	×	×	×		
							RAL	×	×		
								RTR	×		
									RTL		

縦横の命令の交点はその組み合わせを示し，そこに命令が書いてあるもの，および，○印は複合動作が可能であることを示す．△および×印の組み合わせについては結果は複合動作にはならない．BSW命令は555Hのみ，また△印の組み合わせは555Hについてのみ許される．

上記の組み合わせによって，ACに0，1．2．3，2000，4000，7777を1命令で得ることができる．

```
*DEMO  .TM    4/ 1/75 *
#DEMO, OPR GROUP 1:  APR 1, 1975

/(*) INDICATES THE COMBINED INSTRUCTION
/       IS WORKABLE WITH 555H ONLY

0200 7300           CLL CLA
0201 7402           HLT                 /LINK & AC: 0 0000
0202 7301           CLA CLL IAC
0203 7402           HLT                 / 0 0001
0204 7326           CLA STL RTL
0205 7402           HLT                 / 0 0002
0206 7305           CLA CLL IAC RAL
0207 7402           HLT                 / 0 0002 (*)
0210 7325           CLA STL IAC RAL
0211 7402           HLT                 / 0 0003 (*)
0212 7307           CLA CLL IAC RTL
0213 7402           HLT                 / 0 0004 (*)
0214 7327           CLA STL IAC RTL
0215 7402           HLT                 / 0 0006 (*)
0216 7303           CLA CLL IAC BSW
0217 7402           HLT                 / 0 0100 (*)
0220 7332           CLA STL RTR
0221 7402           HLT                 / 0 2000
0222 7350           CLL STA RAR
0223 7402           HLT                 / 1 3777 (*)
0224 7330           CLA STL RAR
0225 7402           HLT                 / 0 4000
0226 7352           CLL STA RTR
0227 7402           HLT                 / 1 5777 (*)
0230 7333           CLA STL IAC RTR
0231 7402           HLT                 / 0 6000 (*)
0232 7346           CLL STA RTL
0233 7402           HLT                 / 1 7775 (*)
0234 7344           CLL STA RAL
0235 7402           HLT                 / 1 7776 (*)
0236 7340           CLL STA
0237 7402           HLT                 / 0 7777
0240 7360           STL STA
0241 7402           HLT                 / 1 7777

$
```

第2類マイクロ複合命令の組合わせ

	SZA	SNL	SPA	SNA	SZL	SKP	CLA	OSR	HLT	実行順序
SMA	○	○	×	×	×	×	○	○	○	1
	SZA	○	×	×	×	×	○	○	○	
		SNL	×	×	×	×	○	○	○	
			SPA	○	○	×	○	○	○	
				SNA	○	×	○	○	○	
					SZL	×	○	○	○	
						SKP	○	○	○	
							CLA	LAS	○	2
								OSR	○	3
									HLT	4

スキップ命令については，SMA，SZA，SNLのOR類とSPA，SNA，SZLのAND類とSKP（無条件）の3類があり，一つの類の中の命令相互間のみ複合動作が可能である．その他の命令についてはすべて複合可能である．ANDとORの区別については，たとえばSPA，SNAはOR結合と仮定すると，無条件スキップで無意味になることから，AND類だと思い出すようにすると良い．

15. 入出力プログラミング

電子計算機が計算または制御をおこなうために使用するデータは一般に何らかの方法で主記憶装置に読みこまれることが必要である．一部のプログラムでは，データの全部がプログラム自体に内蔵され，プログラムと同時に主記憶装置にロードされるが，一般には実行時に必要とするデータの一部または大部分を周辺装置から読みこまねばならない．

また計算結果または制御用出力は何れかの出力媒体または変換器（トランスデューサ）に転送することが必要である．一般に外部機器のデータ転送速度は計算機速度にくらべてかなりおそいことと，計算機の内部構造が同期式であることの理由によってバッファレジスタ，および同期用フラッグを含むインターフェースを必要とする．

周辺機器と主記憶装置間のデータ転送を一語毎にプログラムによりアキュームレータを通しておこなうか，直接主記憶装置に対しておこなうかにより

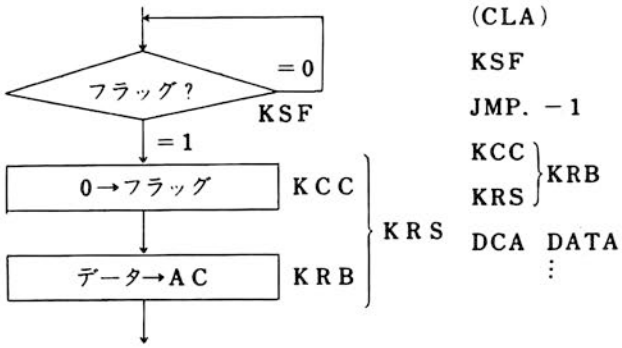
1. プログラムによる入出力と
2. データブレークによる入出力とに分類する

また，プログラムによる入出力は周辺装置との同期のとり方によりフラッグセンスによる方法と割込み制御による方法とに細分できる．本章ではフラッグセンスによるプログラム入出力について述べる．

先ず最初に周辺装置の用意したデータをプログラムで読み取る場合を考えよう．一般に周辺装置のデータ転送速度は演算速度にくらべると非常におそいので，プログラムはデータが用意されたかどうかを先ずチェックしてOKならば読み取るという手順が必要である．入出力タイプライタ（TTY）を例にとると，キーインまたは紙テープ読み取りによって一字8ビットのデータが用意できるとフラッグをセットする．このフラッグをチェックする命令がKSFで，この命令はフラッグが0

の場合はJMP.＋1と等価でありフラッグが1の場合はJMP.＋2と等価である．フラッグ＝1の場合はプログラムは，フラッグを0にして（そうでないと次の入力がいつ用意できたのかわからなくなる）データをACに読みこむ．

プログラム例としては次のようになる．

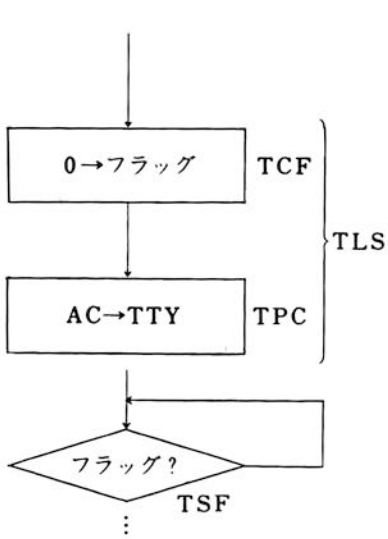


通常のインタフェースの設計ではこのTTYの例のようにフラッグクリアとデータ入力を複合できるようにすることが多いがそうでないこともある．TTYの命令のKCCにはACクリアが含まれているので，CLAは不要であるが一般にはCLAを要する．次にCPUの用意したデータを周辺装置に送る場合を考える．前例とは逆にデータを周辺装置に送った後にそのデータの処理が終ったかどうかをCPUはチェックしなければならない．

TTYのプリンタを例にとると次のようになる．

フラッグをクリアしてデータを送る．データの印刷が終るとフラッグがセットされる．

ここでプログラムは必ずしも印刷の終了を待つことは必要がないので＊の部分に他の仕事を割込ませてもよい．このためにはTLSとTSFの順序を逆にして出力をする直前にフラッグをチェックするようにすればよい．この場合注意しなければならないのは，CPUをスタートさせた時にTTYは動作可能状態にあるにもかかわらずフラッグはリセットされているので，TLSを一回実行させてフラッグをセットする（イニシャラズ）をおこなう必要があることである．




```
        ⋮
        TLS
        TSF
        JMP. －1
        ⋮
        TLS    /INITIALIZE
        ⋮
        TSF
        JMP. －1
        TLS
        CLA
        ⋮
        TSF
        JMP. －1
        TLS
        ⋮
```

16. 割込処理プログラム

前章のプログラミング例では，通常周辺装置の動作が終了するまでの時間の方が通常CPUが計算に要する時間よりも長く，したがってプログラムはTSF; JMP.—1のようなループで無駄に待っている．この時間に実行できる他の仕事をかかえている場合には，割込処理の方法を使うことによって，計算機をより有効に使うことが可能となる．

割込機能とは，いずれかの周辺装置のフラッグが1となった時に，実行中のプログラムを一時中断してJMS 0を実行する機能であってION（6001）の命令を実行することにより割込可能モードとなる．IOF（6002）は逆に割込禁止モードとする命令であって，割込禁止モードでは，周辺装置のフラッグが1となっても，プログラムの実行は何等影響を受けることはない．コンソールパネルのスタートスイッチで計算機の動作を開始させた場合には，割込禁止モードとなっている．

IONの命令を実行すると，IONの次の命令は周辺装置のフラッグがたとえ1であっても通常通り実行される．つまりIONの命令実行後1命令だけおくれて割込可能モードとなる．割込可能モードでは，1つの命令が終了した時に周辺装置のフラッグの状態をチェックし，その中に1のものがあれば，割込禁止モードとしてからその時のプログラムカウンタの値を0番地に書込み1番地に実行が移される．1番地以降にはいわゆる割込処理プログラムが置かれ，この中でフラッグが1となった周辺装置がどれであるかを決定し，それに対する処理プログラムを実行後ION; JMP I 0によって割込発生前のプログラムを続行する．

割込処理プログラムの中で，通常おこなわなければならない事項は次の通りである．

1．割込発生時のアキュームレータおよびリンクの状態を保存し，処理終了後これらを復元して，もとのプログラム（バックグラウンドプロ）に戻る．
2．割込原因となった周辺装置（フラッグが1となったデバイス）を割り出して，このフラッグを0とする．
3．この周辺装置について，入力または出力をおこなう必要があれば，対応する処理をおこなう．
4．もとのプログラムに戻る時にIONを実行する．

キーボードからの入力コマンドにより，バックグラウンドプロの制御およびプリンタ制御をおこなうデモプロを例として示す．

コマンドGによって印刷を開始する．
コマンドEは印刷中にこれを打切る．
コマンドRはコンソールのランプ表示の右シフト．
コマンドLは同じく左シフト．
コマンドSは停止を指示している．
プログラムスタート時には，ランプパタンは右シフトにセットされている．

（注）555Hについては割込処理に関してより高度の命令群の用意がある．

#DEMO. INTERRUPT HANDLER: JAN 1.

FIELD 0

```
                *1
00001 5577 0227         JMP INTRPT
                *20
00020 0000       PATTERN,0
00021 0000       CTR1,   0
00022 0000       CTR2,   0
00023 0000       AC,     0
00024 0000       LINK,   0
00025 0000       TTY,    0
00026 0000       HILO,   0
```

```
                     PAGE
                /INITIALIZE
00200 6001       BEGIN, ION
                /BACKGROUND JOB
00201 3020       A1,    DCA PATTERN
00202 1377              TAD (-3
```

```
00203 3021              DCA CTR1
00204 1020              TAD PATTERN
00205 2022              ISZ CTR2
00206 5205              JMP .-1
00207 2021              ISZ CTR1
00210 5205              JMP .-3
00211 7030      A2,     CML RAR
00212 5201              JMP A1
                /
                MESSAGE,
**********************
00226 2200      TEXT    /%%CEC-555 MINI-COMPUTER
                /INTERRUPT SERVICE ROUTINE
00227 3023      INTRPT,         DCA AC
00230 7204              GLK
00231 3024              DCA LINK
00232 6031              KSF
00233 5236              JMP .+3
00234 6032              KCC
00235 5251              JMP KEY
00236 6041              TSF
00237 5242              JMP .+3
00240 6042              TCF
00241 5425              JMP I TTY
00242 7402              HLT             /UNIDENTIFIED
                /INTERRUPT EXIT
00243 7200      EXIT,   CLA
00244 1024              TAD LINK
00245 7110              CLL RAR
00246 1023              TAD AC
00247 6001              ION
00250 5400              JMP I 0
                /
00251 6036      KEY,    KRB
00252 1376              TAD (-"E
00253 7450              SNA
00254 5300              JMP END
00255 1375              TAD (-"G+"E
00256 7450              SNA
00257 5272              JMP GO
00260 1374              TAD (-"L+"G
00261 7450              SNA
00262 5303              JMP LEFT
00263 1373              TAD (-"R+"L
00264 7450              SNA
00265 5304              JMP RIGHT
00266 1372              TAD (-"S+"R
00267 7450              SNA
00270 5311              JMP STILL
00271 5243              JMP EXIT
                /START PRINT
00272 6046      GO,     TLS
00273 1371              TAD (MESSAGE
00274 3344              DCA PTR
00275 7240              STA
00276 3026              DCA HILO
00277 1370              TAD (TTY1-EXIT
                /STOP PRINT
00300 1367      END,    TAD (EXIT
00301 3025              DCA TTY
00302 5243              JMP EXIT
                /ROTATE LEFT
00303 1366      LEFT,   TAD (RAL-RAR
                /ROTATE RIGHT
```

```
00304 1365      RIGHT, TAD (CML RAR
00305 3211             DCA A2
00306 3024             DCA LINK
00307 3023             DCA AC
00310 5243             JMP EXIT
                /STOP PATTERN
00311 1364      STILL, TAD (NOP
00312 3211             DCA A2
00313 5243             JMP EXIT
                /PRINT SERVICE
00314 2026      TTY1,  ISZ HILO
00315 5334             JMP TTY2
00316 1744             TAD I PTR
00317 7450             SNA
00320 5300             JMP END
00321 7012
00322 7012
00323 7012             RTR;RTR;RTR
00324 1363      TTY3,  TAD (240
00325 0362             AND (77
00326 1374             TAD (240-"%
00327 7450             SNA
00330 5345             JMP TTY4        /% IS FOR CR-LF
00331 1361             TAD ("%
00332 6046      TTY3A, TLS
00333 5243             JMP EXIT
                /6LSB
00334 7240      TTY2,  STA
00335 3026             DCA HILO
00336 1744             TAD I PTR
00337 2344             ISZ PTR
00340 0362             AND (77
00341 7450             SNA
00342 5300             JMP END
00343 5324             JMP TTY3
00344 0000      PTR,   0
                /CR
00345 1360      TTY4,  TAD (TTY5
00346 3025             DCA TTY
00347 1357             TAD (215
00350 5332             JMP TTY3A
                /LF
00351 1356      TTY5,  TAD (TTY1
00352 3025             DCA TTY
00353 1355             TAD (212
00354 5332             JMP TTY3A


00355 0212
00356 0314
00357 0215
00360 0351
00361 0245
00362 0077
00363 0240
00364 7000
00365 7030
00366 7774
00367 0243
00370 0051
00371 0213
00372 7777
00373 7772
00374 7773
```

```
00375 7776
00376 7473
00377 7775



00177 0227



  AC      0023    A1      0201    A2      0211    BEGIN   0200
  CTR1    0021    CTR2    0022    END     0300    EXIT    0243
  GO      0272    HILO    0026    INTRPT  0227    KEY     0251
  LEFT    0303    LINK    0024    MESSAG  0213    PATTER  0020
  PTR     0344    RIGHT   0304    STILL   0311    TTY     0025
  TTY1    0314    TTY2    0334    TTY3    0324    TTY3A   0332
  TTY4    0345    TTY5    0351
$
```

17. データブレークによる入出力プログラミング

前2章に述べた入出力ではアキュームレータを通じて，データの転送がおこなわれる．したがって，一つのデータごとに5個～10個以上の命令が必要で，データ転送速度も10K/秒～100K/秒程度に制限される．割込処理による場合は最高転送速度はさらに低いものとなる．

磁気テープ装置，磁気ドラム装置，磁気ディスク装置など高速，大容量の周辺機器については，データブレーク機能を利用することにより，アキュームレータ経由では実現できないような高速度のデータ転送がおこなわれるだけでなく，1個1個のデータ転送にプログラムのステップ数を要しない利点とが得られる．

データブレークによるデータ転送ではプログラムは，データを入出力するためのコア上の連続したバッファの先頭アドレスと，その語数を指定することと，入力または出力の開始を周辺機器の制御装置に命令するだけで，1個1個のデータ転送には関知しない，周辺機器の制御装置は，アキュームレータ経由でなく，直接主記憶装置のアドレスを指定してデータの読み出し，書込みを行う．プログラムも主記憶装置の読み出し，書き込みをおこなう．周辺装置または，CPUが主記憶装置とデータの転送中は，他側は終るまで待たされる．周辺装置とCPUが同時に要求をした場合は周辺装置からの要求が優先する．複数個の周辺装置から同時にデータブレークの要求があった場合は，あらかじめハードウェアで決められた優先度に従って順に処理される．

転送の開始時に指定された語数の転送が終了すると，周辺装置はフラッグを1とすることにより，プログラムにブロック転送の終了を知らせる．プログラム側は割込処理を使用し，あるいはフラッグセンスによって，その終了を知る．

データブレークには，1サイクル式と3サイクル式とがあり，各々1語のデータ転送について主記憶装置を1回または3回使用する．通常は3サイクル式が用いられるが，1サイクル式の方が，より高速の転送に適する．

1サイクル式では，前述のコアアドレス，および転送語数は，周辺機器の制御装置の内蔵するハードウェアのレジスタにIOT命令によって与えられる．

制御装置はこれらのレジスタを主記憶装置のアドレス指定および，語数カウントに使用し，一回ごとに1ずつ増加する．

3サイクル式では，転送語数，およびコアアドレスを指定するレジスタとして，主記憶装置上の連続した2語が割当てられる点が1サイクル式と異なる．この2語のアドレスは各周辺機器について別々に，あらかじめハードウェア的に設定される．周辺装置は1語の転送ごとにカウンタを読み出して，1増加させ（ISZ），アドレスポインタの内容を1増加させて，その内容を番地に書込み，または読出しをおこなう（オートインデックスに相当）．カウンタを1増加させた結果が0であれば，ブロック転送の終了であり，対応するフラッグを1とする．

アドレスポインタは先頭番地の値－1，カウンタは転送語数の2の補数で与える．3サイクルデータブレークを使用する磁気ドラムによるプログラム例を示しておく．130K語の磁気ドラムのアドレスは17ビットあるので，この5MSBがADDR 1，12LSBがADDR 2であるとする．磁気ドラム上のデータを読み出して主記憶装置上のALPHA番地からBETA番地までに書き込むものとする．磁気ドラムについては，前述のカウンタおよび，ポインタの番地は各々，7750，および7751と決められている．ブロック転送の終了は6622でセンスできるが，割込処理を使用できることは勿論である．

```
                    #DEMO. 3-CYCLE DATA BREAK:  JAN 1. 1975
                    /
                    /CFD-130 IS ASSUMED
                    /
                    /COUNTER LOCATION IS 7750
                    /POINTER LOCATION IS 7751
                    /
                    /DRUM HIGH ADRESS (TRACK #) BE AT AC1-5
                    /       & TRANSFERD TO CONTROLLER BY 6615
                    /
                    /DRUM LOW ADRESS BE HELD AT AC
                    /       & TRIGGERED FOR READ BY 6603
                    /               FOR WRITE BY 6605
                    /
                    /END OF TRANSFER IS SENSED BY 6622
                    /
                    /SUCCESSFUL TRANSFER (NO ERROR) IS SENSED BY 6621
                    /

                    ALPHA= 1000
                    BETA=  1777
                    ADDR1= 20
                    ADDR2= 1000
```

```
                         PAGE
0200 7200          BEGIN, CLA
0201 1377                 TAI (ALPHA-BETA-1
0202 3776\7750            DCA 7750
0203 1375                 TAD (ALPHA-1
0204 3774\7751            DCA 7751        /START ADDRESS-1
0205 1373                 TAD (ADDR1
0206 7106
0207 7006
0210 7006                 CLL RTL;RTL;RTL
0211 6615                 6615
0212 7200                 CLA
0213 1372                 TAD (ADDR2
0214 6603                 6603            /AC IS CLEARED
0215 6622                 6622
0216 5215                 JMP .-1                 /WAIT FOR COMPLETION
0217 6621                 6621            /ANY ERROR?
0220 5222                 JMP ERROR        /YES!!
0221 5223                 JMP NEXT         /NO
0222 7402          ERROR, HLT
0223 7402          NEXT,  HLT
                   /

0372 1000
0373 0020
0374 7751
0375 0777
0376 7750
0377 7000
   ADDR1    0020     ADDR2    1000     ALPHA    1000     BEGIN    0200
   BETA     1777     ERROR    0222     NEXT     0223
$
```

18. 拡張メモリ使用のプログラミング

4K語の主メモリのみを使用するCEC555では，12ビット1語にデータとして記憶した番地を間接参照することによって0～7777番地のすべてを引用できるから問題ない．すでに説明したようにSAP-Ⅱではページ外を自動的に間接参照する機能をそなえているので，プログラマは目的シンボルの実際の番地が，ページの中にあるか，外にあるか意識せずに自由に直接参照するプログラムを書くことができる．

4K語以上の主メモリを制御するためには，12ビット以上の番地レジスタが必要であるが，CEC555では，これを補うために，インストラクションフィールドおよびデータフィールドと呼ぶ各3ビットの2つのレジスタを用意する．このフィールドレジスタの助けにより番地レジスタは12＋3ビットとなり，32K語の主メモリを，データおよびプログラム記憶に使用することが可能となっている．

18—1 FIELD疑似命令

アセンブラの出力バイナリにローダに対するフィールド指定をパンチする．書式は$FIELD_n$である．

また，FIELD疑似命令によって現在ページ定数表，零ページ定数表の出力，リセットがおこなわれる．また，リスティングのプログラムカウンタはFIELD疑似命令以降はn（フィールド指定）を含めて5桁で印字される．

18—2 データフィールドレジスタ

データフィールドレジスタ（DF）は最高3ビット（拡張メモリを含め8K語以下なら1ビット，16K以下なら2ビットで充分である．）のレジスタであってAND TAD，DCA，ISZの4種の命令の間接参照で用いた時に番地レジスタのMSBとして作用する．すなわち自分のフィールド内を参照して得た12ビット番地に，MSBとして，DFを連結した13～15ビットの番地が実際の参照番地となる．直接参照ではDFの値は無関係である．SAP－Ⅱでは直接参照の形でプログラムしても，実際は間接参照になっていることがある．この場合は当然DFの値に影響されるので注意しなければならない．

DFのセッティングは当初CPUをスタートした時のDFセッティングによるが，命令では，CDF（62X1）によって任意の値Xにセットできる．また現在のDFの値は，RDF（6214）でAC_6～AC_8に読み出すことができる．

18—3 インストラクションフィールドレジスタ

インストラクションフィールドレジスタ（IF）はDFと同じく最高3ビットのレジスタであって，DFがMAの延長であったのに対して，PCの延長の役目を果たす．

IFのセッティングは当初CPUをスタートした時のIFセッティングによるが，命令では，CIF（62X2）によって任意の値Xに指定できる．ここで注意しなければならないのはCIFの命令が実効を発するのは，実行後の最初のJMPまたはJMSの命令であって，その他の命令には無関係である．JMPまたは，JMSの命令が直接参照で用いられた場合は，指定されたフィールドの参照番地にコントロールが移るが，間接参照の場合は自分のフィールドを参照して得られた番地を使って指定フィールドにコントロールを移す．

CIFの命令から次のJMPまたはJMSの命令の終了までは自動的に割込禁止状態となっている．

18—4 インタラプトバッファ

インタラプトバッファIBは6ビットのレジスタであって，割込発生時のIF，DFの保存をおこなう．割込が発生すると，その時のIF，DFの値にかかわらず，IFおよびDFを0として，その時のPCの値を0フィールドの0番地に書込み，0フィールドの1番地にコントロールが移る．この時のIF，

DFの保存の役目を果たすのがIBであって，IBの値はRMF（6244）命令によって再びIF，DFに戻される．多重割込みを許す場合は，さらにIBの保存をおこなう必要があるので，このためにRIB（6234）命令を用意してある．

18—5　命令の説明

CDF　62X1　　X → DF
CIF　62X2　　X → IF
（実際のPCへの代入は次のJMSまたはJMP命令で起る．）
RDF　6214　　DF → AC_{6-8}
（AC_{0-5}，AC_{9-11}は無変化）
RIF　6224　　IF → AC_{6-8}
（AC_{0-5}，AC_{9-11} は無変化）
RIB　6234　　IF → AC_{6-8}
　　DF → AC_{9-11}
RMP　6244　　IF → IF
　　DF → DF

18—6　例

1．プログラムをフィールド0に置き，フィールド1にある連続した100個のデータをフィールド2のバッファに移す．フィールド1のデータの最初のデータをACに置いて停止する．

```
#DEMO. EXTENLED DATA FIELD:  JAN 1. 1975



FIELD 0

BUF1=    200              /DATA FIELD 1
BUF2=    400              /BUFFER FIELD 2
```

```
                        PAGE
00200 7200      BEGIN,  CLA
00201 1377              TAD (-144       /DECIMAL 100
00202 3221              DCA COUNT
00203 1376              TAD (BUF1-1
00204 3010              DCA 10
00205 1375              TAD (BUF2-1
00206 3011              DCA 11
00207 6211      LOOP,   6211            /CDF 10
00210 14                TAD I 10
00211 6221              6221            /CDF 20
00212 3411              DCA I 11
00213 2221              ISZ COUNT
00214 5207              JMP LOOP
00215 6211              6211
00216 1774              TAD I (BUF1
00217 6201              6201
00220 7402              HLT
00221 0000      COUNT,  0
                /

00374 0200
00375 0377
00376 0177
00377 7634
```

```
  BEGIN   0200    BUF1    0200    BUF2    0400    COUNT   0221
  LOOP    0207
$
```

(注) このように書くことによって，BUF1が見掛上直接参照可能であっても間接参照の形にアセンブルされる.

2．JMP命令

0フィールドの1ページからフィールド1の0ページ，1ページおよび2ページにコントロールを移すコーディング例を示す．DFの値は無関係である．

```
                FIELD 0


                PAGE
00200 6212              6212
00201 5020              JMP 20          /JUMP TO 10020
00202 6212              6212
00203 5203              JMP .           /JUMP TO 10203
00204 6212              6212
00205 5606              JMP I .+1       /JUMP TO 10500
00206 0500              500
00207 6212              6212
00210 5777\0700         JMP 700         /JUMP TO 10700

00377 0700
```

3．JMS命令

JMS命令は2．項と同様にして実行できるが，コールされたサブルーチン側では12ビットの戻り番地しか与えられないので，一般には主プログラムへの復帰が可能である．これを解決する一つの方法は，戻り番地のフィールドをデータフィールドに置いて，サブルーチンコールをおこなうことである．プログラムがフィールド1，サブルーチンがフィールド2にあるとすれば，そのプログラムの1例は次のようになる．

```
#DEMO. EXTENDED FIELD. JMS:  JAN 1. 1975



FIELD 1


                PAGE
10200 6211              6211            /MAIN DATA FIELD
10201 6222              6222            /SUBROUTINE INSTRUCTION FIELD
10202 4777\0400         JMS SUBR
10203 7402              HLT

10377 0400
```

FIELD 2

```
                *400
20400 0000      SUBR,   0
20401 3213              DCA SAVEAC
20402 6214              6214            /RDF
20403 1377              TAD (6203
20404 3211              DCA EXIT        /CDF CIF
20405 1213              TAD SAVEAC
20406 7000              NOP             /ANY PROGRAM
20407 7000              NOP
20410 7000              NOP
20411 0000      EXIT,   0
20412 5600              JMP I SUBR
20413 0000      SAVEAC, 0

20577 6203
```

```
  EXIT    0411    SAVEAC  0413    SUBR    0400
$
```

4．割込処理

割込みが発生するとそのPCは0フィールドの0番地に保存され，割込発生時のIFおよびDFはIBに保存される．IBに保存されたIFおよびDFの回復はRMF命令によっておこなう．

```
                #DEMO. EXTENDED FIELD. INTERRUPTION:  JAN 1. 1975



                FIELD 0




                *1
00001 5577 0200         JMP INTHDL




                PAGE
00200 3215      INTHDL, DCA SAVEAC
00201 7204              GLK
00202 3216              DCA SAVEL
00203 7000              NOP             /FLAG CHECK & APPROPRIATE PROCES
00204 7000              NOP
00205 7000              NOP
00206 7300      EXIT,   CLA CLL
00207 1216              TAD SAVEL
00210 7110              CLL RAR
00211 1215              TAD SAVEAC
00212 6244              6244
00213 6001              ION             /RMF
00214 5400              JMP I 0
00215 0000      SAVEAC, 0
00216 0000      SAVEL,  0
00177 0200



  EXIT    0206    INTHDL  0200    SAVEAC  0215    SAVEL   0216

$
```

（注）555Hについては割込処理に関してより高度の命令群の用意がある．

19. SAP－Ⅱの操作法

SAP—Ⅱはバイナリテープとして供給される．入出力タイプライタまたは高速紙テープリーダからバイナリローダによって計算機の任意のフィールドにロードする．

スタート番地およびIF，DFをロードしたフィールドの200にセットし，STARTを押すとプログラムはPASS：と印刷するからこれに対してアセンブルしようとするソーステープをリーダにセットし，次のいずれかのキーインをおこなう．

1．パス1

パス1はアセンブラの中のシンボル表をセットするために常に必要である．高速紙テープリーダの場合“1”をキーインするとプログラムは直ちにソーステープの読取りを開始する．入出力タイプライタのテープ読取機を利用する場合は“1”をキーインしてからリーダをSTARTにする．入出力タイプライタのキーボードと高速または低速紙テープリーダは並列となっているのでソーステープ読込中にキーに手を触れてはならない．

読込中もしエラーを発見すれば随時エラー表示を入出力タイプライタに出力する．最後に$を読込むとプログラムはそれまでに入力のあったシンボルを走査し，未定義のものがあればこれに対するエラー表示をおこなう．これを終了するとプログラムはPASS：と印刷して次のコマンドを待つ状態となる．

2．パス2

(1) パス2における最も一般的な方法は，ソーステープをリーダにセットし，スペースをキーインすることである．この場合アセンブラは最も効果的な出力デバイスを自動選択してバイナリテープおよびリスティングを出力する．バイナリテープは高速紙テープパンチャがあればこれに出力されるが，なければ入出力タイプライタに出力される．

リスティングはラインプリンタがあればこれに出力される．なければ入出力タイプライタに出力される．出力機器として入出力タイプライタのみしかない時は，バイナリテープのみが出力され，リスティングは出力されない．パス2におけるエラー表示は入出力タイプライタに出力される．入出力タイプライタがバイナリテープ出力用として用いられている場合はこのエラー表示はバイナリテープの中間に残ることになるが，これはバイナリローダによって無視されるように配慮されている．

(2) バイナリテープのみを必要とする時はソーステープをリーダにセットし“2”をキーインする．出力機器は自動的に高速紙テープパンチャまたは入出力タイプライタが選択される．

(3) リスティングのみを必要とする時は，ソーステープをリーダにセットし，“3”をキーインする．プログラムはさらにOUT：と印刷してくるので出力機器の指定は“1”（入出力タイプライタ）“2”（高速紙テープパンチャ）または“3”（ラインプリンタ）のいずれかをキーインしておこなう．

(4) シンボル表のみを必要とする時はソーステープをセットする必要はない．“3”をキーインし，さらにOUT：に対して出力機器の指定をおこなう．さらに$↵とキーインすると直ちにシンボル表が出力される．

(5) パス2の(1)～(4)の各操作は任意の順序で必要回数だけ重複しておこなってよい．

注(1) SAP—Ⅲは操作上はSAP—Ⅱと同じである．

注(2) SAP—Dはソース入力としてドラムまたはディスクを使用する点がSAP—Ⅱと異なり，最初にファイル名を入力する操作が加わるが，ファイル入力では見掛上1パスである．

USのエラー表示はパス1の最後ではおこなわずパス2で未定義シンボルを参照の都度おこなう．

操作についてはマークⅡモニタ説明書を参照されたい.

20. SAP－Ⅱのソースプログラムの作成

SAP—Ⅱのソースプログラムは通常エディタを使用して入出力タイプライタよりキーインして作成する. SAP—Ⅱはカードペースのアセンブラと異なり, ラベル, 命令, アーギュメント, コメントなどの各フィールドは相互間の順序を除いては自由な形で使用できる. しかし, できあがったプログラムのリスティングの読みやすさとエディタの機能を考慮して通常は次のようにするのが良い.

1. コア上の1ページ分（128語）程度, あるいはそれ以下の区切りの良い部分毎にエディタで処理する1ページとする.
2. 各行はアドレスラベル, 命令およびオペランド, コメントの3つのフィールドで構成し, 各フィールドの間はTABで区切る. 命令およびオペランドのフィールドは16字分取ることとする. 従って命令およびアーギュメントの合計が7字以下の場合はコメントフィールドの前にTABを2つ挿入する.
3. コメントフィールドのみの行は1字目に / を置いて直ちにコメントの本文に続ける.
4. TEXTの行または；を含む行は複数の語にアセンブルされ, リスティングはその最終行に印刷されるのでアドレスラベルはつけない方がよい. 必要のある場合はその前にアドレスラベルのみの1行を設ける.
5. プログラムの流れの論理的な切れ目ごとに空行, / のみの行, またはコメント行を挿入する.
6. プログラムの一番最初に（最初のページの第1行）＃で始まるラベル行を置く. ラベルはプログラム名：日付の順に書き, 同一日に修正する場合には日の後にA, B……と添字をつける.
7. ラベルの次に, 1行あけその次の行から＝疑似命令によるシンボル定義を1行に1個ずつおこなう. これは通常次の各項が含まれる.

(1) SAP—Ⅱシンボル表に含まれていない計算機命令の定義

Ⅰ）オプションの金物に関する命令

Ⅱ）プログラムの都合により標準と異る命令法を採用している命令

CDF＝6201

CDF 2 ＝6221

SXA＝SPA SNA

RRB＝6016（RDメッセージが出る.）

(2) 別にアセンブルされた他のプログラムとの連絡のために必要なプログラム入口番地, レジスタ番地などの定義

〔例〕

HSR＝6215

同時に使用する他のプログラムで6215番地から高速紙テープリーダのサブルーチンがある.

(3) 同時にアセンブルするプログラムの中で同一番地を別の名前で引用するための定義

(4) そのプログラムをアセンブルする時にパラメータとして与えるシンボルの定義

```
〔例〕N = 3        /FIELD TO BE LOADED
        ⋮
        FIELD N
        ⋮
```

(5)プログラムを読みやすくするための呼びかえ.

例　POP=　JMS I POPI

　　⋮

　　POPI, POPUP

ただし，POPUPを設定しようとするページにプログラムカウンタを置いた状態で

　POP=　JMS POPUP

とすれば上記は自動的に行われる．

あるいはSAP－Ⅰを使用する時は！擬似命令およびページ外参照自動の機能がないので，0ページに参照ベクトルを割当てる段階で次のようにするのがよい．

```
*20
    ⋮
POP=JMS I
    POPUP
    ⋮
```

8．7．項までをソースプログラムの第1ページとすると，プログラム全体に対するコメントその他はこの中に含める．

9．第2ページ目までは0ページに関するデータを扱う．もしフィールド指定があれば，これを1行目でおこなう．

10．次に1行あけて，！疑似命令による0ページ定数表の割当をおこなう．1行に何回書いてもよいが，各シンボルまたは各数値ごとに1を前後に付け加える．リスティングの見やすさから，1行に7個以下とし，各々の間は！の後にTABを挿入すると良い．

11．！疑似命令で設定した定数値を使用する表現の定義をおこなう．（7．—(5)項参照）

12．*でオリジンを設定してコアの0ページに置くプログラムを書く．

7．項の定義は0ページにオリジンを設定した後でおこなえば！疑似命令による定数の設定は不要である．

13．以下コアの各ページごとに1ページにまとめてゆく．各ページの第1行は，PAGE疑似命令とする．FIELD　擬似命令で0ページ定数表が出力されてしまうので，1つのフィールドに置くプログラムは全部続けて，間に他のフィールドのプログラムはなるべくはさまない方が安全である．途中必要に応じて疑似命令によるオリジンの再設定をおこなってよい．

付録A．SAP—II　疑似命令一らん

ＤＥＣＩＭＡＬ	以降の数値は10進数として処理される．
ＯＣＴＡＬ	以降の数値は8進数として処理される．
ＦＩＥＬＤｎ	パス1では各定数表のポインタをリセットする．パス2は現在ページおよび0ページの定数表をバイナリパンチする．各定数表のポインタはリセットされる．ローダに対するnフィールド指定コードをパンチする．
＊××	××が現在ページの中か外かでその動作は異なる．××が現在ページの中の場合には単にプログラムカウンタを××にセットする．パス2ではさらにローダに対するオリジン指定コードをパンチする．××が現在ページの外にある場合に次の動作をおこなう．パス1では現在ページ定数表をリセットし，プログラムポイントを××にセットする．パス2では現在ページの定数表をバイナリパンチし，定数表のポインタをリセットする．プログラムカウンタを××にセットし，ローダに対するオリジン指定コードをパンチする．

PAGE　パス１では現在ページ定数表をリセットし，プログラムカウンタを次ページの最初にセットする．（そのページにデータが無ければプログラムカウンタはそのままである）パス２では現在ページ定数表をバイナリパンチし定数表のポインタをリセットする．プログラムカウンタを連続する次ページの最初の番地にセットする．

$××　パス１ではユーザの使用したシンボルをすべてチェックし，もし未定義のものがあればエラー表示をおこない，アセンブルのパス１を終了する．パス２では現在ページおよび０ページの定数表をバイナリパンチし，チェックサムおよびエントリ番地××をパンチ後，トレーラをパンチする．リスティングの最後にユーザの使用したシンボルとその値とをABC順にリストアップする．アセンブル終了，$ $は使用しないこと．

PAUSE　アセンブルを一時中止する．
ソーステープが何本にも分かれている場合に各テープの前後に用いる．次のテープの読込みを再開させるためには入出力タイプライタに何かキーインすればよい（入力が入出力タイプライタの場合は不要）．

FIXTAB　パス１に関してのみ有効で，これ以前に登録されたユーザのシンボルはすべて固定シンボル扱いとなり以後のパス１でリセットされない．またリスティングの最後のシンボル表にも出力されない．

I　オクタル値400のシンボルであるが，さらにこのシンボルの含まれた表現については間接参照命令としてのチェックをおこなう．

#　パス１ではこれに続く文字ストリングをCRまで入出力タイプライタにコピーする．パス２ではバイナリテープに見出しおよびラベルとしてコピーする．

″　これに続く文字のASCII値を表現の値とする．

!　直前のシンボルまたは数の値を０ページ定数表に登録する．！によるシンボル値の登録はその値を引用する命令よりも前におこなうか，あるいはそのシンボル値が定義されるよりも後でおこなうかしないとアセンブル結果が正しくない可能性がある．

(　これに続く表現の値を０ページ定数表に登録されているかどうかをチェックする．未登録ならば，さらに現在ページ定数表に登録されているかどうかをチェックし，未登録ならば登録する．（で始まる表現の値は０ページまたは現在ページ定数表の登録番地である．なお（が０ページで用いられた場合は，〔と同じに解釈される．

〔　これに続く表現の値を０ページ定数表に登録されているかどうかをチェックし，未登録ならば登録する．〔で始まる表現の値は定数表の登録番地である．

TEXT　TEXTの次の特殊文字は区切り文字としてのみ解釈される．その次の文字はこれに続くストリングの終端記号である．従ってTEXTの，最後のTの次から数えて３字目以降の文字は，そのASCII値の下６ビットを２字ずつ１語として順次連続したメモリ番地に割当てられる．
終端記号が再び現れるか，その行の終りに達すると上記の変換は打切られ，最後に６ビットまたは12ビットの０が付加される．TEXTの次の区切り文字としては，スペース，またはタブを用いるのが望ましい．その次の終端記号はストリングの中に含まれない文字ならば何でも良いが通常は／，？などを用いるのが良い．

```
                PAGE
******
 0203 0000      TEXT    /CEC555/
                /TERMINATING CHARACTER MAY BE OMITTED
******
 0207 0000      TEXT    ?CEC555
**
 0211 0300      TEXT    /CEC

 0212 0500      TEXT    CEC
```

	直前のシンボルの値を，プログラムカウンタの値に等しく設定する．，の前には一つ以上のシンボルがあってはならない．直前のシンボルの値がすでにシンボル表に登録されている場合にはIDエラーとなる．シンボル表に登録されている値は変更されない．
＝	直前のシンボルの値を＝の右側の表現の値に等しく設定する．直前のシンボルの値がすでにシンボル表に登録されている場合にはRDエラーの表示をする．RDエラーは単に警告であって，登録されている値は，新しく指定された値に変更される．
／	／以下の文字はすべてコメントとしてアセンブラは無視する．ただし／が＃の後，〃の直後，TEXTの中に現れる場合を除く．
ＮＯＬＩＳＴ	以降リスティングは印刷されない． (SAP—IIIおよびSAP—Dのみ)
ＥＮＬＩＳＴ	以降リスティングは印刷される．（通常の状態に戻る．） (SAP—IIIおよびSAP—Dのみ)

付録B．SAP—IIの番地参照命令一らん

記法：以下の記号によって表現する．

A→B	レジスタAの内容をレジスタBにいれる．
0→A	レジスタAの内容を０とする．
A j	レジスタAのj番目のビット（j＝０～11）の内容．
$A_{6\text{-}11}$	レジスタAのビット６～11の内容．
$A_{6\text{-}11}\rightarrow B_{0\text{-}5}$	レジスタAのビット６～11をレジスタBのビット０～５にその順に入れる．
ＡＣ	アキュームレータ（累算器）
ＭＱ	ＭＱレジスタ
ＰＣ	プログラムカウンタ
Ｙ	任意のメモリYの内容
(Y＋１)	Y＋１番地の内容
∨	論理和
⊕	排他的論理和
∧	論理積

$\overline{A}$　　レジスタAの内容の1の補数（各ビット）

PAC　　インタプリータの疑似アキュームレータ（3語）

QT　　整数演算インタプリータの除算の商

シンボル	値	内　容
AND	0000	ビットごとの論理積．ACj∧Yi → ACj
TAD	1000	和をとる．桁上げはLINKに及ぶ． $L,AC_{0-11} + Y_{0-11} \rightarrow L,AC_{0-11}$
ISZ	2000	メモリの内容に1足す．結果が0となった時は次の命令をスキップする．ACおよびLINKは不変． Y＋1→Y， その結果Y＝0 ならば，PC＋1 → PC
DCA	3000	ACの内容をメモリに書込み，ACを0とする． LINKは不変． AC→Y 0 →AC
JMS	4000	サブルーチンジャンプ，参照番地の次の番地を書込み参照番地の次の命令にコントロールを移す． PC ＋ 1 →Y Y ＋ 1 →PC
JMP	5000	無条件ジャンプ Y→PC

浮動演算，3倍長演算インタプリータ用命令

FISZ	0000	
FJMP	0000	
FADD	1000	加算　PAC＋(Y,Y＋1,Y＋2)→PAC
FSUB	2000	減算　PAC－(Y,Y＋1,Y＋2)→PAC
FMPY	3000	乗算　PAC＊(Y,Y＋1,Y＋2)→PAC
FDIV	4000	除算　PAC／(Y,Y＋1,Y＋2)→PAC(QT)
FGET	5000	メモリ→擬似AC　(Y,Y＋1,Y＋2)→PAC
FPUT	6000	擬似AC→メモリ　PAC→Y,Y＋1,Y＋2
FJMS	7000	サブルーチンコール

（注）FISZは20番～177番地の直接参照に限る．

FJMP
FJMS　} は0ページの直接参照は不可

アセンブラは上記の制限について何等チェックをおこなわないので使用するときは注意すること．

FDIV(3倍長演算)では剰余がPACに，商がQTに残される．

付録C．SAP-IIのその他の命令一らん

NOP 7000　　何もしない．

IAC 7001　　ACに1加算

AC＋1→AC（桁上げはLINKに及ぶ）

BSW 7002 †　　ACの右半分と左半分をいれかえる．

$AC_J \rightarrow AC_{J+6}$　J＝0～5

$AC_J \rightarrow AC_{J-6}$　J＝6～11

RAL 7004　　ACとLINKをあわせて左に1ビット回転シフト

$AC_J \rightarrow AC_{J-1}$, J＝1～11

$AC_0 \rightarrow L$

$L \rightarrow AC_{11}$

RTL 7006　　ACとLINKをあわせて左に2ビット回転シフト

$AC_J \rightarrow AC_{J-2}$, J＝2～11

$AC_1 \rightarrow L$

$AC_0 \rightarrow AC_{11}$

$L \rightarrow AC_{10}$

RAR 7010　　ACとLINKをあわせて右に1ビット回転シフト

$AC_J \rightarrow AC_{J+1}$,　J＝0～10

$AC_{11} \rightarrow L$

$L \rightarrow AC_0$

RTR 7012　　ACとLINKをあわせて右に2ビット回転シフト

$AC_J \rightarrow AC_{J+2}$,　J＝0～9

$AC_{10} \rightarrow L$

$AC_{11} \rightarrow AC_0$

$L \rightarrow AC_1$

CML 7020　　LINKビットを反転

$\overline{L} \rightarrow L$

CMA 7040　　ACCの各ビットを反転（1の補数）

$\overline{AC}_J \rightarrow AC_J$, J＝0～11

CLL 7100　　LINKビットを0にする．

$0 \rightarrow L$

CLA 7200　　ACを0にする．

$0 \rightarrow AC$

HLT 7402　　停　止

$0 \rightarrow RUN$

OSR 7404　　ACとSRのビット論理和をACに置く．

$AC_J \wedge SR_J \rightarrow AC_J$

SKP 7410　　次の命令をとばす．

PC＋1 → PC

SNL 7420　　LINK≒1ならばSKP, L≒1ならばPC＋1 → PC

SZL 7430　　LINK＝0ならばSKP, L＝0ならばPC＋1 → PC

SNA 7450　　AC≒0ならばSKP, AC≒0ならばPC＋1 → PC

SZA 7440　　AC＝0ならばSKP, AC＝0ならばPC＋1 → PC

SPA 7510　　AC≧0ならばSKP, AC_0＝0ならばPC＋1 → PC

SMA 7500　　AC＜0ならばSKP, AC_0＝1ならばPC＋1 → PC

CIA 7041　　ACの2の補数をとってACに置く．(CMA IAC)−AC → AC
(桁上げはLINKに及ぶ)

STL 7120　　LINK＝1とする.(CLL CML) 1 → L

STA 7240　　ACの各ビットを1とする．(CLA CMA)
1 → AC_J, J＝0〜11 (−1 → AC)

LAS 7604　　SRの値をACに置く.(CLA LAS)
SR_J → AC_J, J＝0〜11

GLK 7204　　LINKをAC11に移し，LINKおよびAC 0〜10を0とする．
(CLA RAL)
0 → AC_J, J＝0〜10
L → AC_{11}
0 → L

MQL 7421†　　ACの内容をMQへ転送し，転送終了後ACの内容をクリアする．
AC→MQ, 0→AC

MQA 7501†　　MQとACの論理和をACにいれる．
AC∨MQ → AC

SWP 7521†　　ACとMQの内容をいれかえる．
AC ⇄ MQ

CLA MQL 7621†　　ACとMQをクリアする．
0 → AC, 0 → MQ

CLA MQA 7701†　　MQをACに移す
MQ → AC

CLA SWP 7721†　　MQをACに移し，MQをクリアする．
MQ → AC, 0 →AC

MPQ 6771†　　PCの内容をMQに転送する．
PC → MQ

MAP 6772†　　ACの内容をPCに転送する．
AC → PC

MAPQ 6773†　　ACの内容をPCに，PCの内容をMQに転送する．
AC → PC → MQ

——— 6774†　　MQの内容をACに転送する．
MQ → AC

MPQA 6775†　　PCの内容をMQに，MQの内容をACに転送する．
PC → MQ → AC

MQAP 6776†　　MQの内容をACに，ACの内容をPCに転送する．
MQ → AC → PC

RAPQ 6777†　　MQの内容をACに，ACの内容をPCに，PCの内容をMQに転送する．
MQ → AC → PC → MQ

SKON 6000†　　IONまたはION 1の時にSKPおよびIOFまたはIOF 1

ION 6001　　次の命令を実行し終ったら，割込みを許す．

IOF 6002　　割込みを禁止する．

SRQ 6003 †　　インタラプトリクエストがあればSKP

GTF 6004 †　　フラグ，DF，IFなどをACに読込む．

0	1	2	3	4	5	6	7	8	9	10	11
LINK	INT1 BUS	INT BUS	INT INH F·F	ION	ION1	SF0 (IF0)	SF1 (IF1)	SF2 (IF2)	SF3 (DF0)	SF4 (DF1)	SF5 (DF2)

RTF 6005 †　　ACの内容をフラグなどにセットしてION，GTFの逆命令

CAF 6007 †　　AC，LINK，フラグ類をリセットする．

KCF 6030 †　　キーボードフラグをクリアする．

KSF 6031　　入出力タイプライタ(以下TTYと略す)のキーボードフラグが1ならばSKP

KCC 6032　　TTYのキーボードフラグとACを0として次の字を読ませる．

KRS 6034　　TTYのキーボードバッファとACの論理和をACCに置く．

KIE 6035 †　　$AC_{11}=1$ならばキーボード，パンチエネーブル
(ONの時，KBD，TTYから割込み可，標準状態)
$AC_{11}=0$ならばキーボードパンチ割込み禁止
(ONでもKBD，TTYからの割込み禁止)

KRB 6036　　TTYのキーボードバッファをACに移し，キーボードフラグを0とし，次の字を読ませる．

TFL 6040 †　　TTYフラグをセットする．

TSF 6041　　TTYのプリンタフラグが1ならばSKP

TCF 6042　　TTYプリンタフラグを0とする．

TPC 6044　　AC_{4-11}をTTYで印刷およびパンチする．

TSK 6045 †　　プリンタ，またはキーボードフラグが1で，かつキーボード・パンチインタラプトエネーブル フリップ フロップが1つの時次の命令をスキップする．

TLS 6046　　TTYプリンタフラグを0としAC_{4-11}をTTYで印刷する．終ったらプリンタフラグを1とする．

RPE 6010 †　　リーダ／パンチインタラプト エネーブル フリップフロップをセットする．

RSF 6011　　高速紙テープリーダフラグが1ならばSKP

RRB 6012　　高速紙テープリーダバッファとACの論理和をACに置き，リーダフラグを0とする．

RFC 6014　　リータフラグを0とし，紙テープから1字読み取ってバッファに置く．終ったらリーダフラグを1とする．

PCE 6020 †　　リーダ／パンチインタラプト エネーブル フリップフロップをクリアする．

PSF 6021　　パンチャフラグ1ならばSKP

PCF 6022　　パンチャフラグとパンチャバッファを0とする．

PPC 6024　　ACとパンチャバッファの論理和をバッファに入れパンチする．終ったらフラグを1とする．

PLS 6026　　パンチャフラグを0とし，ACの内容をバッファに移してパンチする．終ったらフラグを1とする．

SMP 6101　　メモリパリティエラーでない時SKP

SPL	6102	電源異常の時SKP
STSU	6103†	スーパーバイザコールセット
SKSU	6104†	スーパーバイザコールオンならばSKP
CLSU	6105†	スーパーバイザコールリセット
CDF	62N1•	データフィールドレジスタにNをロードする.
CIF	62N2•	インストラクションフィールドをNに変更する. 実際の実行は次のJMPまたはJMSが命令が与えられた時に起る. またその時まで割込み禁止となる. 割込み禁止フリップフロップの状態を知らないで一時的に割込み禁止とするためにも用いられる.
RDF	6214•	データフィールド値をAC6～8に読込む. AC0～5 およびAC9～11は不変
RIF	6224•	インストラクションフィールドの値をAC6～8に読込む. AC0～5およびAC9～11は不変
RIB	6234•	インタラプトバッファの値をAC6～11に読込む.割込み発生時のインストラクションフィールドがAC6～8にデータフィールドがAC9～11に読込まれる.
RMF	6244•	インタラプトバッファに保存されたインストラクションフィールドおよびデータフィールドを各ロードする.

浮動演算, 3倍長演算インタプリータ用

FEXT	0000	インタプリータモードから通常のモードに戻る.
FCIA	0006	符号反転－PAC → PAC
FCLA	0007	PACクリア　0 → PAC
FSZA	7050	PAC＝0ならばSKP
FSPA	7100	PAC≧0ならばSKP
FSNA	7040	PAC≠0ならばSKP
FSMA	7110	PAC＜0ならばSKP
FSKP	7020	SKP
FCDF	70N1	インタプリータモードのデータフィールドセット
FCHG	7005	PAC ⇄ QT（3倍長インタプリータのみ）
FNOR	7000	正規化（浮動インタプリータのみ）
FNOP	7007	NOP

（注）FSPAとFSNA, FSMAとFSZAのみ複合可.

• SAP-ⅢおよびSAP-Dのみ.

† 〃 (555Hのみ)

付録D　SAP-Ⅱのエラー表示

IC××××ATnnnn不正文字 (Illegal Character)

××××は不正に使用された文字のオクタル値で, nnnnは発見された時のプログラムカウンタのオクタル値である. 不正文字はSAP—IIで許されていない文字がコメント, テクスト, ラベル, ASCII文字のいずれかの中以外で使用された場合か, 数の中に英字が用いられた場合である. 不正文字は区切り文字として処理される.

RD××××ATnnnn　重複定義 (Re Define)

××××は重ねて＝疑似命令によって定義されたシンボリック記号，nnnnはその時のプログラムカウンタの値である．

シンボリック記号××××の値は再定義される．

ID××××ATnnnn　不正定義 (Illegal Define)

××××は，疑似命令によって異なる場所で再定義されたシンボリック記号，nnnnはその時のプログラムカウンタの値である．シンボリック記号××××の値は最初に定義されたまま変らない．

PE××××ATnnnn　ページ溢れ (Page Exceeded)

××××は無意味，nnnnはプログラムカウンタの値である．ページ外参照用ベクトルおよび（疑似命令で設定される定数表は現在のページの最終番地から順次最大128個まで使用できるが，これがプログラムと重複した場合に表示される．

II××××ATnnnn　不正間接参照 (Illegal Indirect)

××××はページ外のシンボリック記号，nnnnは××××を間接参照した時のプログラムカウンタ，参照する番地がページ外であっても直接参照であればベクトルをページ内に設定し，間接参照とすることによって，SAP—IIはアセンブルをおこなうが，ページ外の間接参照は不可能である．

SE××××ATnnnn　シンボル表溢れ (Symbol table Exceeded)

××××はシンボリック記号，nnnnはこれが発生した時のプログラムカウンタのオクタル値である．

SAP—IIは，414個のユーザシンボリック記号を許しているが，これを越えて使用した場合にSEとなる．このソースプログラムについてのアセンブルは打切られる．

PO××××ATnnnn　プッシュダウン表溢れ
(Pushdown list Overflow)

PU××××ATnnnn　プッシュダウン表溢れ
(Pushdown list Underflow)

プッシュダウン表の溢れ，nnnnはその時のプログラムカウンタのオクタル値，このソースプログラムについてのアセンブルは打切られる．

US××××ATnnnn　未定義シンボル (Undefined Symbol)

××××は値の定義されなかったシンボリック記号，nnnnはこれが最初に参照された時のプログラムカウンタのオクタル値．

パス１の最後にシンボル表を走査してこのエラーが発見される．

シンボリック記号の値としては，ゼロが与えられる．

IR××××ATnnnn　ページ外参照 (Illegal Referenece) (SAP—Iのみ)

××××はページ外のシンボリック記号，nnnnは××××を参照した時のプログラムカウンタ，SAP—I以外では，××××の値をページ内に自動的に設定するので，ページ外の番地を間接参照しない限りエラーとはならない．

（この場合はIIエラー）

パス１であらわれるエラー

IC，RD，PE*，SE，US，***，ID

パス２，３であらわれるエラー

II*，PE*，IR**，US****，

・SAP Iを除く

＊・SAP Iのみ

＊＊＊　SAP Dを除く

＊＊＊＊SAP Dのみ

付録E. SAPファミリイのアセンブラ

§SAP—I

SAP—IはSAP—IIから現在ページおよび0ページ定数表を取除いたもので，(および[はICエラーとなる．また番地参照命令で参照番地が0ページ，または現在ページに無い場合にはIRエラーとなる．

ユーザ用のシンボル表の容量は580個でSAP—IIよりも多少大きいが，SAP—IIにおける定数および間接参照ベクトルもシンボルをつけなければならないので実効的には必ずしも大きくない．

§SAP—Ⅲ

SAP—ⅢはSAP—Ⅱのシンボル表をフィールド1に移し，8K語用としたものである．ユーザ用のシンボル表の大きさが920個と大きくなっている．

またNOLISTおよびENLIST疑似命令が追加された．FIELD疑似命令が使用された後のリスティングではPCの値はフィールドを含め5桁で印刷される．各パスの最初にPCの印刷ルーチンは通常の4桁印刷にリセットされる．

§SAP—M

SAP—Ⅱの入出力機器としてカセットを用いられるようにしたもので，カセットモニタのもとで動作する．

入力にカセットのファイルを指定した場合には，最高4個のファイルについて各々パス1およびパス2を連続して順次実行する．

§SAP—D

SAP—Ⅲの入出力機器としてドラムまたはディスクを用いられるようにしたもので，マークⅡモニタのもとで動作する．コアは8K語必要である．入力はドラムまたはディスクのファイルで，パス1およびパス2は連続して実行される．

NOLISTおよびENLISTが追加された，SAP—Ⅱ，Ⅲ，Mと異なりPASS1の終りにシンボル表の未定義シンボルを印刷するのをおこなわない．未定義シンボルがPASS2で引用されると，その都度エラーメッセージを印刷する．

SAP—Ⅲと同様にFIELD疑似命令が使用された後のリスティングではPCの値はフィールドを含め5桁で印刷される．

ＳＡＰファミリアセンブラの比較

		ＳＡＰ－II	ＳＡＰ－III	ＳＡＰ－Ｍ	ＳＡＰ－Ｄ
主記憶容量		４　Ｋ　語 （任意フィールド）	８　Ｋ　語 （0.1フィールド）	４　Ｋ　語 （任意フィールド）	８　Ｋ　語 （0.1フィールド）
補助記憶		×	×	ＣＭＴ-100	ＣＦＤ，ＣＦＰ ＣＦＸまたはＣＸＤ
周辺機	ＴＴＹ	ＣＳ-100 またはＣＳ-1000	ＣＳ-100 またはＣＳ-1000	ＣＳ-100 またはＣＳ-1000	ＣＳ-100 またはＣＳ-1000
	ＰＴＲ	（ＣＳ 408）	（ＣＳ 408）	（ＣＳ 408）	（ＣＳ 408）
	ＰＴＰ	（ＣＳ 420）	（ＣＳ 420）	（ＣＳ 420）	（ＣＳ 420）
	ＬＰ	｛ＣＬＰ 4100または ＣＬＰ 100｝	｛ＣＬＰ 4100または ＣＬＰ 100｝	｛ＣＬＰ 4100または ＣＬＰ 100｝	｛ＣＬＰ 4100または ＣＬＰ 100｝
ＰＡＳＳ1エラー		IC, RD, PE, SE, US, ID	←	←	IC, RD, SE, ID
ＰＡＳＳ2エラー		II, PE, IR, IC	←	←	II, PE, IR, US
拡張メモリ関係命令コード		×	内　蔵	×	内　蔵
プログラム番地桁数（リスティング）		4	4または5	4	4または5

付録F．ライブラリ　サブルーチンとその使用法

SAP　アセンブラ（SAP—Iを除く）のプログラミングの特長の一つにソースプログラムレベルのリロケーション機能がある．アセンブラソースのレベルでのリロケーションは通常当然のことと考えられがちであるが，実際上は必ずしもそうではない．SAP—Ｉを例にとって見ても次のような問題点があるので，ある特定の機能のサブルーチンを一定の形で用意して置くことが不可能である．

1．　メインプログラムからのコーリング

サブルーチンとメインプログラムが同一ページにあれば問題ないが，そうでない場合は間接指定ベクトルを用意してこれにシンボルSUBRIをつけ，ＪＭＳ　Ｉ　SUBRIの形でコールしなければならない．ベクトルを０ページに用意できれば良いが，そうでない場合は各ページごとに異なる名前で呼ばなくてはならない．

2．ライブラリサブルーチンから他のライブラリサブルーチンのコーリング

１つのライブラリサブルーチンから他のライブラリ サブルーチンをコールする場合についても１．項と同様の問題が生じる．これを避けるためには各ライブラリ自体に内蔵させるか，いくつかのライブラリを一まとめにして扱うかをしなければならず，いずれにしても不当に無駄なコアスペースを消費する可能性が多い．

3．ライブラリ　サブルーチン内の定数

サブルーチンが使用する定数はすべて内蔵する必要があるので，同一ページの他のサブルーチンと定数を共用することができず，不当にコアスペースを消費する．また各定数にシンボルを割当てるとすれば，これが他のルーチンやメイン　プログラムと重ならないようにする必要があり，シン

ボルを割当てないようにするとすれば，サブルーチンの保守上問題が大きい．

以上のような理由により従来のアセンブラではライブラリ　ルーチンを使用する場合，多少の手直しをおこなうことが避けられない．

SAP—Ⅱ，Ⅲ，D，Mなどにあっては，他のサブルーチンをコールするのに，ページの内外の区別無しに直接引用でソース　プログラムをコーディングできるので1.，2．の問題点が解決される．また（による定数定義ができるので3．の問題点も解決される．さらに，これらのベクトルおよび定数のコア　ロケーションは現在ページに設定する場合も零ページに設定する場合もコーディング自体は共通で良く，メインプログラムでゼロページに設定したい場合は！によってその選択ができるのでコアスペースは従来のコーディングよりも無駄なく利用される．

ライブラリサブルーチンはASCⅡソース　テープの形で供給される．UTCプログラムをコアにロードして200番地からスタートさせるとNAMEと印刷される．ライブラリテープを高速紙テープリーダにセットしてから，必要なサブルーチン名のすべてをコンマを区切り符号としてキーインする．最後にCRをキーインするとライブラリ　テープを読み始め，指定されたサブルーチンを発見すると，そのサブルーチン名をタイプライタに印刷すると共に高速紙テープパンチにソースをコピーする．コピーされた紙テープは各サブルーチン毎にエディタの1ページとしてパンチされるので，エディタを使用して，メイン　プログラムと容易に結合できる．

ライブラリサブルーチンーらん表

A）1字出力するルーチン

```
JMS  PRINT     (TTY)
JMS  LPOUT     (ラインプリンタ)
```

B）オクタル4桁印字

```
JMS  OCTPR
```

C）ヘクサデシマル3桁印字

```
JMS  HEXPR
```

D）デシマル印字

```
JMS  DECPR     (単語長符号付)
JMS  SDPRNT    (倍長符号付)
```

E）BCD印字（4 LSBを1字）

```
JMS  BCDPR
```

F）文字ストリング印字（240～337）

```
JMS  STRING
TEXT  /.....
```

G）文字ストリング印字（201～376）

```
JMS  TEXTPR
TEXT  ADDRESS
```

H）1字入力するルーチン

```
JMS  KEYIN     (TTY，印字は)
JMS  TREAD     (TTYまたはPTR)
```

I）オクタルまたはデシマル入力ルーチン

```
JMS  NOMBIN    (単語長AC=0ならばオクタル入力　AC≠ならばデシマル入力)
JMS  DICONU    (倍長デシマル符号付)
```

J）分岐ルーチン

```
JMS  SRJMP
```

```
        "A ; TESTA
        "B ; TESTB
          .
          .
          .
        0または負数（END）
```

K）BCD→BIN変換

```
        JMS   BCDBIN     （単語長）
        JMS   DOUBLE     （倍長）
```

L）BIN→BCD変換

```
        JMS   BINBCD     （単語長）
```

M）乗算（符号なし、12ビット×12ビット→24ビット）

```
        JMS   MULT
```

N）除算（符号なし、24ビット÷12ビット→12ビット）

```
        JMS   DIVIDE
```

O）シフト演算

```
        JMS   SPSL       （単語長左シフト）
        JMS   SPSR       （単語長右シフト）
        JMS   LSRSP      （単語長右論理シフト）
        JMS   DPSL       （倍長左シフト）
        JMS   DPSR       （倍長右シフト）
        JMS   LSRDP      （倍長右論理シフト）
```

P）論理和

```
        JMS   INCOR      （論理和）
        JMS   EXCOR      （排他的論理和）
```

Q）平方根

```
        JMS   SQRT.
```

付録G. SAP-II　コア割当(任意フィールド)

0～3311 プログラム本体
3312－3351 擬似命令シンボル表
3352－3441 番地参照命令シンボル表
3442－3705 その他の命令シンボル表
3706－7077 ユーザ用シンボル表 （414個）
7100－7277 現在ページ定数表
7300－7477 0ページ定数表
7500－7577 ソースプログラムバッファ
7600－7777 ロ　ー　ダ

§ SAP-III コア割当

0－3477 プログラム本体
3500－4133 シンボル表
6200－6377 現在ページ定数表
6400－6577 0ページ定数表
6600－6777 テキストバッファ
（COD）
7600－7777 ローダ
10000－10043 擬似命令シンボル表
10044－10133 番地参照命令シンボル表
10134－10433 その他の命令シンボル表
10434－17577 ユーザ用シンボル表 （920個）
17600－17777 ローダ

§ SAP-D コア割当

<table>
<tr><td colspan="2">0－3677
プログラム本体</td><td rowspan="3">システムファイル
へのセーブエリア</td></tr>
<tr><td colspan="2">3700－4327
シンボル表</td></tr>
<tr><td colspan="2"></td></tr>
<tr><td rowspan="4">6000－6777
コマンドデコーダ</td><td>現在ページ定数表</td><td rowspan="4"></td></tr>
<tr><td>0ページ定数表</td></tr>
<tr><td>テキスト用バッファ</td></tr>
<tr><td>ドラム用バッファ</td></tr>
<tr><td colspan="2">（COD）</td><td></td></tr>
<tr><td colspan="2">7600－7777
モニタ</td><td></td></tr>
<tr><td colspan="2">10000－10043
擬似命令シンボル表</td><td></td></tr>
<tr><td colspan="2">10044－10133
番地参照命令シンボル表</td><td></td></tr>
<tr><td colspan="2">10134－10427
その他の命令シンボル表</td><td></td></tr>
<tr><td colspan="2">10430－17577
ユーザ用シンボル表
（921個）</td><td></td></tr>
<tr><td colspan="2">17600－17777
ローダ</td><td></td></tr>
</table>

付録H. 参考事項

1. 零ページ定数表はFIELD疑似命令または$疑似命令によって出力にパンチされる. 従って, 一つのフィールドのプログラムの中間に他のフィールドのプログラムをはさむことはできるだけ避けた方が安全である.

2. 命令を定数としてとる場合にエラーメッセージが出ずに正しくアセンブルできない場合がある.

例

```
                                #DEMO. INCORRECT ASSEMBLING:  JAN 1. 1975


                PAGE
0200 1376\0400          TAD (JMP 400
0201 3775\0500          DCA 500

0375 0500
0376 5777
0377 0400
```

この場合, JMP 400は5200とアセンブルすることを期待してもプログラム自体のカウンタは200にあるので, 5777 (377に400) とコーディングされてしまう. JMP400の命令を200～311番地の中で使用する場合にはアセンブラのコーディングは正しい.

3. 命令を定数としてとる場合にエラーメッセージが出るが正しくアセンブルされる場合がある.

例

```
                                #DEMO. INCORRECT ASSEMBLING:  JAN 1. 1975


                PAGE
0200 0200        0200
0201 1377               TAD (JMP I 400
0202 3776\0520          DCA 520

0376 0520
0377 5600
```

この場合IIのエラーメッセージがでるがJMP I 400自体は5600と期待通りにアセンブルされる.

この場合JMP I 400の命令を200—377番地のリスティング中で使用するとすれば, 当然ながら誤りである.

4．2．3．に対する正しいコーディングは次の通りである．

```
                #DEMO. CORRECT CODING:  JAN 1. 1975


                PAGE
0200 1377               TAD (INST1      /JMP 400
0201 3776\0500          DCA 500
0202 1375               TAD (INST2      /JMP I 400
0203 3774\0520          DCA 520
0204 1372\0400          TAD (JMP 400
0205 3300               DCA 300

0372 5773
0373 0400
0374 0520
0375 5600
0376 0500
0377 5200



               PAGE
               INST1= JMP 400
               INST2= JMP I 400



   INST1   5200    INST2   5600
$
```

5．定数表は*疑似命令またはPAGE疑似命令がくるまでリセットされないので２ページ以上にわたる連続したプログラムで定数または自動間接を使用した場合，その間の異なるページの定数表がすべて共通にとられ，またPEエラーは各ページの終りの部分毎に発生する．

§4　オクタルディバッカ

1．はしがき

COD（CEC OCTAL DEBUGGER）およびXOD（EXTENDED OCTAL DEBUGGER）は，入出力タイプライタ（TTY）を使ってユーザプログラムの任意の部分の実行，任意の番地およびレジスタの読出し，変更などの動的デバッグを行うためのプログラムである．

アセンブラによってアセンブルされたバイナリオブジェクトプログラムに含まれる論理ミスを発見，修正するためには通常次の3つの方法が有効で，しかも後の方法ほど有効性が高い．

1．机上デバッグ

フローチャートおよびリスティングによって論理フローをチェックする．やっかいで，しかも経験を積まないと必ずしも有効とは限らない．

2．静的デバック

オブジェクトプログラムを適当な個所から実行，その動作を調べる．また所期の結果が得られない時に，プログラムがどこでループしているか，各ソフトウェアレジスタがどのような状態になっているか，プログラムのいずれかが破壊されていないかなどを調べる．この目的のためにはベリファイローダの使用がきわめて能率的である．

3．動的デバッグ

COD，またはXODを使えばプログラムの任意の番地から任意の番地まで任意の回数を実行させて，この時のアキュームレータおよびその他のソフトウェアレジスタの値をチェックしながらプログラムの流れを追うことができる．コンソールスイッチを使って1ステップずつ実行してプログラムの流れを追う原始的な方法に比べて，ソフトウェアを使用した動的デバッグは数千倍以上も能率的であるといえる．また，プログラムに誤りを発見した場合のプログラムの部分的変更と，そのテストランに関してもCODおよびXODはコンソールスイッチを使う場合に比べてはるかに能率的である．CODおよびXODにおいてはすべての数値はオクタル数によって表示およびキーインされるので，コンソールを使用する場合に比べて人間の負担は大幅に軽減され，誤りのチャンスを減らすことができる．

CODはデバッグされるプログラムの0ページに1ロケーション（通常は4番地）およびデバッグするプログラムと同じフィールドの任意の3ページ（通常は7000～7577番地）を占有する．デバッグされるプログラムは実行途中で他のフィールドを使用してもよいが，CODによってランを行う部分の入口および出口におけるIFおよびDFはCODのロードされたフィールドと一致している必要がある．

XODはCODの動作領域を任意フィールドを選択できるように拡張したもので，デバッグされるプログラムの0ページに3ロケーション（通常は4～6番地）と任意フィールドの任意の5ページ（通常は6400～7577番地）を占有する．デバッグされるプログラムはXODのロードされたフィールドと無関係に任意のフィールドで動作が行える．またXODとデバッグされるプログラムとの間でコントロールが移る場合，XODはインタラプトのON-OFF，TTYフラッグのON-OFF，さらにDFおよびIFなどを正しく保存する．

2．CODの機能

CODは次の諸機能を持っている．

1．CODのロードされたフィールド内の任意のロケーションの内容の印刷と内容の変更．
2．CODのロードされたフィールド内の任意のブロックのバイナリパンチ．出力テープはオリジンおよびチェックサムを含み，バイナリローダ（CBNまたはV）でコアにロードできる．
3．CODのロードされたフィールド内の任意の連続ロケーションのオクタルダンプおよびサーチ．
4．CODのロードされたフィールド内の任意のロケーションにブレークポイントを設定することおよびユーザプログラムの実行．ユーザプログラムの実行を開始する時のアキュームレータ（AC）およびリンク（LK）の値は任意に指定できる．ユーザプログラムにおいて，実行が設定したブレークポイントロケーションに達した時，そのロケーションの命令は実行せずにブレークポイントロケーションとAC，LKをオクタルプリントしてコントロールをCODに戻す．ユーザはユーザプログラムの任意の部分のチェックおよび変更を行って，さらにプログラムを続行できる．プログラムを続行する時にはACおよびLKは正しく復元されるが他の値に変更もできる．ブレークポイントからの続行に当っては新しいブレークポイントを指定することもできる．また指定した回数だけブレークポイントを通過した後に，コントロールをCODに戻すようにすることができる．ブレークポイントはユーザのプログラムで変更される命令，浮動小数点演算および倍長演算の疑似命令，データフィールドまたはインストラクションバッファがCODのロードされたフィールドと異なる命令など，小数の制限を除きほとんどの命令に設定できる．

3．CODの操作

CODでユーザのプログラムのデバッグを行うためには，次のものを用意する．

1．ユーザのプログラムのバイナリテープ
2．ユーザのプログラムのリスティング
3．ユーザのプログラムとコア上で重なり合わないCODのバイナリテープ

バイナリローダによって，ユーザのプログラムおよびCODをコアにロードし，CODのスタート番地（通常は7000）からスタートさせる．以下のコマンドを任意に実行する．

4．CODのコマンド

1．／：任意番地のオープン

／の直前にキーインされたオクタル数（0～7の数字の組合わせ，5桁以上の場合は最後の4桁のみが有効．3桁以下の場合はその前に0が補われる）で示されるロケーションがオープンされ，その内容がオクタル4桁で印刷される．オープンされたロケーションの内容は，任意のオクタル数をキーインし，クローズ（CR，LF，↑，←による）することによって変更されないでクローズされる．

／コマンドがオクタル数を伴わずに単独に用いられた時は最後にオープンしたロケーション（B，C，E，G，P，T，Wの各コマンドを実行しても変化ない）が再びオープンされる．

2．↙（CR）：オープンしたロケーションのクローズ

CRの前にオクタル数のキーインがあればオープンされたロケーションの内容をキーインされた内容に変更してクローズする．CRのみをキーインした場合はオープンされたロケーションの内容はそのままでクローズされる．

例

```
1200/5641 ↵
/5641 ↵
1200/5641 0 ↵
1200/0000
```

3．↓（ＬＦ）：クローズして次のロケーションをオープン

例

```
400/0200 ↓
0401 /3252 2345 ↓
0402 /1253 ↵
401/2345
```

4．↑（SHIFT/N）：クローズしてその内容のアドレス部分に示されたロケーションをオープン

例

```
401/2345 ↑
0545 /5457 1205↑
0405 /3253 ↵
545/1205
```

5．←（SHIFT/O）：クローズしてその内容のロケーションをオープン

例

```
401/2345 ←
2345 /7616 123←
0123 /7600 ↵
2345/0123
```

6．A：CODの内部ACレジスタをオープン

CODの内部にはACおよびLKの２つのレジスタをソフト的に用意してあり(LKのレジスタは7202番地)，後述のGおよびCコマンドでユーザープログラムにコントロールを移す前にこれらの値をACおよびLKにセットする．Aコマンドはこの COD内部のACレジスタをオープンし，変更を可能とする．LKレジスタは直接オープンすることはできないが，AコマンドでACレジスタをオープンした後，LFによってオープンできる．AおよびＬＫの値は当初はおのおの０である．

例

```
A0000 1234↓
7202 /0000 1 ↵
A1234 ↓
7202 /0001
```

7．M：COD内部のサーチマスクのオープン

次項に説明するWコマンドに使用するためのサーチマスクMをオープンする．前項の場合と同様Mで内容が印刷され変更可能，さらにLFによりサーチ範囲の下限が，さらにLFにより，サーチ範囲の上限がオープンできる．Mおよびサーチ範囲は当初おのおの7777，0001および6777となってい

る．7402番地および7403番地はそれぞれサーチ範囲の上限，下限を格納するレジスタとして使われている．

なお，サーチ範囲は7777→0000とつながり，もし下限に7770，上限に0010が指定された場合7770～7777および0000～0010が範囲としてとられる．

例

```
M7777↵
7402 /0001 1000 ↵
7403 /6777 1777 ↵
```

8．xxxW：ワードサーチ

指定する範囲でxxxのデータがどこにあるかをサーチする．前項のMコマンドおよびそれに連続するLFで設定した範囲のコア内容とxxxを比較して一致すればそのロケーションと内容をすべて印刷する．ただし比較はMで指定した値の1のビットについてのみ行う．あるロケーションのコアの内容をY，Mの値をM，Wで指定するデータをDとした時，

$$Y \cdot M = D \cdot M$$

なら印刷される（・は論理積）．

またワードサーチのマスクを0とすればサーチ範囲のすべてを印刷するから，コア内容のオクタルダンプとしても利用できる．

例

```
M7777 0 ↓
7402 /1000 7770 ↓
7403 /1777 10↲

7770 /3350
7771 /1351
7772 /3753
7773 /6201
7774 /1350
7775 /3746
7776 /5763
7777 /7777
0000 /0000
0001 /5402
0002 /0200
0003 /0000
0004 /7250
0005 /0000
0006 /7776
0007 /0500
0010 /0000

M0000 7777 ↲
5763W
7776 /5763

M7777 7700 ↲
3700W
7772 /3753
7775 /3746

M7700 7000 ↲
3000W
7770 /3350
7772 /3753
7775 /3746

M7000 70 ↲
50W
7770 /3350
7771 /1351
7772 /3753
7774 /1350
0004 /7250
```

Wコマンドでサーチおよび印刷を行っている時，任意のキー入力（たとえばスペースバー）によって，Wコマンドの動作を中止させることができる．何かのキー入力があるとCODはサーチを中止して？を印刷する．このキー入力の内容は無視される．

9．T：リーダ（トレーラ）パンチ

バイナリテープのリーダをパンチする．テレタイプに出力されるから，Tをキーインした後に，PUNCH ONとする．適当な長さのリーダが得られたら，PUNCH OFFとしてからCPUをSTOPさせる．CODのスタート番地から再スタートする．

10．xx；xxxP：コア内容のパンチ

xxおよびxxxはおのおのオクタル数とする，CODはxx番地からxxx番地までをオリジンxxをつけてバイナリパンチする．xx；xxxPとキーインするとCPUはストップしTTYをPUNCH ONとする余裕を与える．TTYをPUNCH ONとしてからCPUのCONTキーを押すとバイナリパンチを行う．パンチが終ったら次のコマンドをキーインする前にTTYをPUNCH OFFとしないとバイナリテープにASCIIコードがパンチされ使用できなくなるので注意が必要である．

コアの別々の幾つかのブロックを続けてパンチするにはPコマンドを繰返せばよい．

xx；xxxPとした時xxxがxxに等しければ，xxの1番地だけがオリジンとともにパンチされる．

xxxがxxよりも小さい場合は7777番地までパンチされる．

xxPとした時の結果は無意味である．

11．E：チェックサムおよびトレーラのパンチ

Eをキーインすると，TTYをPUNCH ONとするためにCPUがストップするのはPコマンドと同様である．PUNCH ONとしてからCONTを押す．適当な長さのトレーラが得られたら，TTYをPUNCH OFFとしCPUをSTOPさせる．CODを再スタートさせる．

12．xxB：xx番地にブレークポイント設定

ブレークポイントとはユーザのプログラムが進行してプログラムカウンタが設定したブレークポイントに達した時にコントロールをCODに移し，AC，LKをはじめとし，コアの各種ソフトウェアレジスタの内容の点検，変更，さらにはプログラムの修正などCODを使用する動的デバッグのキーポイントをなすものである．

xxBとすることによりxx番地にブレークポイントの設定が行われるが，実際のソフトウェア処理は次に述べるGコマンド，Cコマンドが実行された時にはじめて行われるので，ブレークポイントの有無にかかわらずコアの任意のロケーションの修正は自由である．Gコマンドが実行されるとCODは設定されたブレークポイントの番地にCODへのジャンプ命令を挿入し，ACおよびLKをAコマンドで設定された値（または直前のブレークで持ち込まれた値）に復元して指定された番地にジャンプする．

ユーザプログラムの実行中にブレークポイントに達すると，その時のACおよびLKの値をCOD内部のレジスタに保存してブレークポイントのユーザプログラムを復元する．さらに，ブレークポイントロケーションをオクタル4桁，LK，ACをオクタル5桁で印刷してCODにコントロールを移す．

ユーザはCODの任意の機能を利用してAC，LKの変更を含め，点検・修正を行うことができる．さらにプログラムを続行する必要があればBコマンドにより，ブレークポイントを変更した後にCコマンドを実行させる．CODはGコマンドの時と同様に，AC，LKを復元した後に，ブレークポイントにあった命令を正しく実行して，ユーザプログラムのしかるべき番地にコントロールを移す．

オクタル数を伴わずBのみを指定されるとCODは前に設定したブレークポイントを取り除く．

ブレークポイントを設定しユーザプログラムを実行して，コントロールがブレークポイントに達しなかった場合は，CODを再スタートすることによってブレークポイントのユーザプログラムは復元される．

13. xxG：xx番地からプログラム実行

14. xC：実行再開

直前のブレークポイントからプログラム実行を再開し，新しいブレークポイントをx回通過した後，ブレークポイントでCODにコントロールを移す．オクタル数を伴わずCのみであれば最初にブレークポイントに達した時にCODにコントロールは戻る．ブレークポイントによってCODに戻ったのでない時にCコマンドを実行した時の結果は保証されない．

5．CODについての補足

1．TTYフラッグ

CODのGコマンドおよびCコマンドでユーザのプログラムにコントロールを移す時のTTYフラッグ（テレタイプのプリントフラッグ）は1となっている．ある種のプログラムでTTYフラッグが1では困る場合には7350番地（ORG+350）の内容7000を6042に変えることによって目的が達せられる．この変更はCODの／コマンドを使って行ってよい．

2．ブレークポイント

ブレークポイントは以下の例外を除いてCODと同じフィールドであれば任意のロケーションに設定できる．

(1) 現実に実行される命令でないもの
　たとえば，インタプリータの疑似命令は実行命令でなくＪＭＳで呼ばれたサブルーチンのデータであるからブレークポイントを設定できない．

(2) ユーザのプログラムで変更される命令

(3) ユーザのプログラムでデータに兼用されている命令

(4) CIF命令およびその後の最初のJMSまたはJMPまでのすべての命令

(5) CDF命令およびDFがIFと違う部分にあるすべての命令

(6) オートインデックスを使用するJMS命令(現実に意味があるかは不明)

3．インタラプト

インタラプトオンのプログラムにブレークポイントを設定する場合は充分の注意が必要である．

(1) ユーザプログラムのインタラプト処理ルーチンの組み立て方によってはブレークポイントが取除かれるチャンスがある．

(2) (1)と同様の理由によってブレークポイントを通過する回数が正しくカウントされない可能性がある．

(3) ブレークポイントを通じてCODに戻った時にインタラプトは禁止状態になる．

4．ＣＯＤの再スタート

任意の時にランニング中のユーザのプログラムをストップさせてCODを再スタートさせることができる．ユーザのプログラムをストップさせるにはCPUのコンソールのSTOPキーを押し，さらにCODの実行開始番地をLOAD PC, STARTとしてCODを再スタートさせる．もしブレークポイントを設定してあればユーザプログラムのその番地の命令は復元される．

5．ユーザプログラムの再ロード

ブレークポイントを設定した状態でCODを再スタートさせるとブレークポイントの命令は復元される．したがって1つのプログラムでブレークポイントを設定し，他のプログラムをロードしてCODを再スタールさせると，そのプログラムが破壊される場合がある．

6．CODのリロケート

CODの使用する0ページ内の1ロケーションはソーステープのBRLOCの値を変更することによって0～7および20～177番地の任意のロケーションに変更できる．COD本体はORGの値を0ペー

ジおよび最終2ページを除く任意のページの先頭に設定することによって任意のコアロケーションに変更できる．実行開始番地はORGの値に等しい．

7．エラー

CODは0～7の数字および規定されたコマンド以外の入力に対しては？を印刷してエラーを表示する．

6．XODの機能

XODはCODの諸機能に加えて次の諸機能を持っている．

1．XODのロードされたフィールドだけでなく，すべてのフィールドの任意のロケーションの内容の印刷と内容の変更．

2．すべてのフィールドの任意の連続ロケーションについて，オクタルダンプおよびサーチ．サーチは指定パターンに合致したものを印刷する機能の他に合致しないものを印刷する機能も用意されている．

3．すべてのフィールドの任意の連続ロケーションにすべて一定の指定データを書き込むこと．

4．すべてのフィールドの任意のロケーションにブレークポイントを設定することおよびユーザプログラムの実行．CODではブレークポイント設定に伴い0ページの1ロケーションを使用したのに対し，XODでは0ページに連続した3ロケーションを使用することを除いて同様である．

5．サーチのためのマスク，サーチ範囲の上限，下限，ブレークポイントにおけるACおよびLINKなどはCODのようにオープンすることはできないが，読み出すことおよび任意の値に設定することができる．

6．CODにおけるP，E，Tの諸コマンド（パンチ）は使用できない．

7．ブレークポイントにおいてはDFおよびIF，インタラプトのON-OFF，TTYフラッグのON-OFFを保存する．

7．XODの操作

XODでユーザのプログラムのデバッグを行うには次のものを用意する．

1．ユーザのプログラムのバイナリテープ

2．ユーザのプログラムのリスティング

3．ユーザのプログラムとコア上で重なり合わないXODのバイナリテープ

バイナリローダによってユーザのプログラムおよびXODをコアにロードし，XODのスタート番地（通常はロードされたフィールドの7000番地からスタートさせる（DFセッティングは任意）．以下のコマンドを任意に実行する．

8．XODのコマンド

XODのコマンドは2，3の相違を除けばCODのコマンドと同様であるが，すべてのコマンドについてXOD内部のフィールドレジスタFLDDによってその動作が修飾される点が異なる．

1．F：フィールドセッティング

Fの直前にキーインされたオクタル数を内部のソフトウェアフィールドレジスタFLDDにセットする．オクタル数は最後の1桁のみが有効である．オクタル数を伴わないでFのみをキーインした時は現在のFLDDの値を印刷する．また計算機のコンソールパネルのDF表示は常にこのFLDDの値を表示している．XODをロードしてスタートさせた直後の状態ではDFセッティングにかかわらず，

FLDD＝0

となっている

2．／：任意番地のオープン

3．↙：オープンしたロケーションのクローズ

4．↓：クローズして次のロケーションをオープン

5．↑：クローズしてその内容のアドレス部分に示されたロケーションをオープン

6．←：クローズしてその内容のロケーションをオープン

2．～6．のコマンドはおのおのFLDDの内容で修飾される点を除けばCODの対応するコマンドと同じである．ただしオクタル数を伴わずに単独で用いられた／はエラーである．最後にオープンしたロケーションを再びオープンするには次項のコマンドを用いる．

7．・：再オープン

最後にオープンしたロケーション，あるいは現在オープンしたままになっているロケーションを再びオープンする．

オクタル数の直後にキーインすれば，最後にオープンしたロケーションにオクタル数を加えたロケーションを新しくオープンする．

なお，最後にオープンしたロケーションとは，W，Vコマンドの後ではそれによって最後に印刷したロケーションとなっている点がCODと異なっている．

8．A：XOD内部のACレジスタのセット　初期値は　0000

9．Y：　〃　LINK　〃　〃　0000

10．M：　〃　MASK　〃　〃　7777

11．L：　〃　LLIM　〃　〃　0000

12．U：　〃　HLIM　〃　〃　6377

8．～12．までのコマンドはCODのAコマンドおよびMコマンドと同じ目的に使用できる操作法が異なる．これらのレジスタはCODと異なり，オープンすることはできないが読み出すことと任意の値にセットすることができる．8．～12．までのコマンドをオクタル数を伴わずに使用すると対応するレジスタの内容をオクタル印刷する．オクタル数の直後にこれらのコマンドを使用すると，対応するレジスタの内容はそのオクタル数にセットされる．

13．xxxW：ワードサーチ

14．xxxV：ネガティブワードサーチ

Wコマンドの動作はサーチ範囲がLLIM，HLIM，FLDDと指定されていることを除いてCODと同じである．LLIMおよびHLIMはおのおのサーチ範囲のスタートおよびエンドでこれらのフィールドはFLDDで指定される．

Vコマンドの動作はWコマンドでは条件に合致したものを印刷していたのに対し，条件に合致しないものを印刷する点だけが異なる．

15．xxx%：ブロックストア

LLIM，HLIM，FLDDで指定されている範囲にすべてxxxをストアする．%のみをキーインすれば上記の範囲をすべて0にクリアする．

16．xxB：ブレークポイント

ブレークポイントの設定においてはCODが0ページの4番地をCODにコントロールを移すために使用したのに対し，XODではブレークポイントを設定するロケーションの含まれているフィールドの0ページの4，5，6番地を使用する点が異なる．設定ロケーションのフィールド指定ではBコマンドを使用する時のFLDDの値に従う．Bコマンドを使用した後で，FコマンドによってFLDDの値を変更してもブレークポイントは影響されない．

またブレークポイントのために使用される4，5，6番地はBコマンドを使用するまでは使われない点もCODと異なる．

ブレークポイントを設定してユーザのプログラムをランさせた時にブレークポイントに達しない場合，XODを再スタートさせることができる．XODを再スタートさせるとブレークポイントの命令は復元され，ブレークポイントの設定コマンド自体がキャンセルされる．

17. xxG：xx番地からプログラム実行

18. xxC：直前のブレークポイントからプログラム実行を再回し，新しいブレークポイントをx回通過した後，ブレークポイントでXODにコントロールを移す．

9．XODについての補足

1．T，P，Eコマンド

CODのこれらのコマンドはXODでは使用できない．

2．ブレークポイント

ブレークポイントは以下の例外を除いて任意のフィールドの任意のロケーションに設定できる．

(1) 現実に実行される命令でないもの

(2) ユーザのプログラムで変更される命令

(3) ユーザのプログラムでデータに兼用されている命令

(4) CIF命令およびその後の最初のJMSまたはJMP命令までのすべての命令

これらの各項の制限の内容についてはCODの場合と同じである．

ブレークポイントの機能はCODの場合よりも大幅に高級化している．

(1) 任意のフィールドに設定できる．

(2) ブレークポイントから再スタートする時にTTYフラッグ，IF，DF，インタラプトの状態はいずれも保存される．

(3) ブレークポイントを通じてXODに戻る時，ブレークポイントロケーションはフィールドを含めて5桁で印刷される．

3．インタラプト

XODは設定したブレークポイントにおけるインタラプトのON-OFFの状態を保存して，コントロールがXODの中にある間はインタラプト禁止の状態におく．Cコマンドによってユーザのプログラムにコントロールを移す時には保存してあったインタラプトの状態を復元する．

4．XODの再スタート

任意の時にランニング中のユーザのプログラムをストップさせてXODを再スタートさせることができる．XODを再スタートさせると，以前のブレークポイントは解除される．またAC，LINK，TTY，FLAGなどをクリアする．

10．CODおよびXODコマンドのまとめ

A	ACオーブンおよびLINKオーブンの前操作（COD）
	ACのチェックまたは指定（XOD）
xxB	ブレークポイント設定
xxC	ブレークポイントからのプログラム続行およびブレークポイント通過回数指定
E	チェックサムおよびトレーラのパンチ（CODのみ）
xF	フィールド指定（XODのみ）
xxG	ユーザプログラムへのジャンプ
L	サーチ範囲の下限番地のチェックまたは指定（XODのみ）
M	サーチマスクのオーブンおよびサーチ範囲の下限，上限レジスタオーブンの前操作（COD）

	サーチマスクのチェックまたは指定（XOD）
x;xxP	オリジンおよびデータのパンチ（CODのみ）
T	リーダパンチ（CODのみ）
U	サーチ範囲の上限番地のチェックまたは指定（XODのみ）
xxV	ネガティブワードサーチ（XODのみ）
xxW	ワードサーチ
Y	LINKのチェックまたは指定（XODのみ）
;	Pコマンドの区切り（CODのみ）
↑	オペランドアドレスオープン
←	インダイレクトオープン
LF	次番地オープン
CR	クローズ
xx	オープン
	再オープンまたはxx番地先をオープン（xx· ）（XODのみ）

§5　ユーティリティ

1. パンチプログラム

1－1　目　的

4K語またはそれ以上の主記憶装置を持つCEC555シリーズ計算機を使用して，任意のロケーションをBIN形式，またはRIM形式でせん孔する．

1－2　周辺機器

入出力タイプライタ（TTY）

高速紙テープパンチャ（オプション）（PTP）

1－3　使用メモリ

いずれかのフィールドの6000-7577

1－4　操　作

ロード後，IFおよびDFをロードしたフィールドに設定して，SA＝7200でスタートさせる．TTYに＃を印刷するので，コマンドをキーボードから入力する．

T	リーダコード（オクタル200）を約15cmせん孔する．
C	リーダコードを約15cmせん孔後RUBOUT CR／LFをせん孔する，以後CRのキーインがあるまでTTYからのキーインをせん孔する．CRのキーインでRUBOUT，約15cmのリーダせん孔を行って，次のコマンドに移る．
nF	n＝0～7　パンチしたプログラムをロードする時にフィールド指定を行いたい場合に用いる．Dの指定とは無関係である．
nD	n＝0～7　出力するフィールドを指定する．ロードした直後のデータフィールドは0になっている．
nP	nは1～4桁のオクタル数．n番地の内容をBIN形式で出力する．
m；np	m，nは1～4桁のオクタル数．n番地からm番地までの内容をBIN形式で出力する．m＞nの場合はm～7777，0～nまでを出力する．
nR	nPと同様の機能であるがRIM形式で出力する．
m；nR	m，nRと同様の機能であるが，RIM形で出力する．
E	チェックサムおよびトレーラを出力する．
nG	nは1～4桁のオクタル数．チェックサム，エントリn番地，トレーラを出力する．

1－5　注意事項

1．PコマンドとRコマンドを混合して用いてはならない．

2．BIN形式の時はTで始め，EまたはGで終了する．
RIM形式の時はTで始め，Tで終了する．C，F，E，Gコマンドを使用してはならない．

3．PTPが用意されている時は直ちにPTPにパンチされる．PTPがない時は，コマンド入力後まずTTYのパンチャをONにしてからスペースバーを押すとTTYのパンチを開始する．リーダのパンチ終了後，まずパンチOFFとしてから，スペースバーを押すとプログラムはCR/LFを出して次のコマンドを受付ける状態となる．

2．デュープおよびベリファイプログラム

2—1　目　的

4 Kまたはそれ以上の主記憶装置を持つCEC555シリーズ計算機を使用して，紙テープからマスターテープの作製，マスターテープからの複製，マスターテープまたは複製のチェックを行う．チェック方法として，CRC-16ビットを使用しているので，いかなるフォーマットの紙テープを複製することが可能である．

2—2　周辺機器

入出力タイプライタ（TTY）
高速紙テープリーダ（PTR）
高速紙テープパンチャ（PTP）

2—3　使用メモリ

0002～7577

2—4　操　作

ロード後，DF（データフィールド）およびIF（イレストラクションフィールド）をロードしたフィールドに設定し，SA＝0200でスタートさせると
TTYに
*OPT：
と印字する．

コマンド　：　M…マスターテープの作成
　　V…マスターテープおよび複製のチェック
　　C…マスターテープからの複製

1．マスターテープの作製（MASTER）
　複製すべき紙テープをPTRにセットし，＊OPT：に対して，M（MASTER）をキーインすると，マスターテープを作製する．
　マスターテープには2文字のチェックキャラクタがテープエンドに付いている．
　作製が終了するとTTYに
MASTER　CREATED
*OPT：
と印字する．

作製したマスターテープは照合機にかけるか，目視によって原紙テープとのチェックを行う必要がある．もしOKならばさらに次項のVコマンドによりCRCのチェックを行う．この2つのチェックがOKならばマスターテープとして完全である．

2．マスターテープおよび複製のチェック（VERIFY）
　チェックすべき紙テープのブランクの部分をPTRにセットし，
　*OPT：
に対してV（VERIFY）をキーインすると複製した紙テープのチェックを行う．
　＊チェックの結果が正しければ，TTYに
　VERIFY　OK
　*OPT：
と印字し，誤りであれば
　VERIFY　ERROR
　*OPT：

と印字する.

紙テープPTRにセットする時，誤ってブランク以下の部分をPTRにセットした場合は，TTYに，

BAD SETTING TAPE

*OPT :

と印字するので，この場合には紙テープをPTRに正しくセットしなおして，チェックを行う.

チェックの結果がエラーの場合は，再度ベリファイを行い，それでもエラーならばこの紙テープは破棄する.

3．マスターテープからの複製（COPY）

作製したマスターテープのブランクの部分をPTRにセットし，

*OPT :

に対してC（COPY）をキーインすると，テープの複製を行う．複製が終了すると，マスターテープが長い場合は，

COPY END

*OPT :

と印字する．マスターテープが短い場合には

COPY END

*NUMBER OF MORE COPIES

と印字するので，0，1～9の数字をキーインする．最大10本まで一度に複製することができる．0をキーインするとイニシャルにもどる．0以外の数字をキーインすると複製が終了するごとに残り数を印字する.

2—5 注意事項

1．オプションCの場合には複製すると同時にチェックも行うので，エラーが生じた場合は複製途中でエラーメッセージ（*VERIFY ERROR*）を印字し，イニシャルにもどる．この場合は，再度試みるかマスターテープの再作製が必要である.

例

```
*OPT:M
*MASTER CREATED*

*OPT:V
*VERIFY OK*

*OPT:C
*COPY END*

*NUMBER OF MORE COPIES : 3

*REMAIN : 2
*REMAIN : 1
*COPY END*

*OPT:V
*VERIFY OK*

*OPT:
```

3. I.D. BINプログラム

3—1 目 的

BINテープを読んで,

1. コメントのプリント
2. コアロケーションのプリント
3. チェックサムのプリント

を行う.

3—2 周辺機器

入出力タイプライタ

高速テープリーダ（オプション）

3—3 使用メモリ

いずれかのフィールドの0—0377

3—4 操 作

ロード後，IFおよびDFをロードしたフィールドでSA＝0200を設定する．チェックするBINテープを入出力タイプライタ，または高速紙テープリーダにセットし，CPUをスタートさせる．

BINテープでフィールド指定がない場合にはコアロケーションは4桁のオクタル数でプリントされる．フィールド指定がある場合は，

FIELD n

と印刷し，以後のフィールド指定があるまでn××××／n××××のように印刷する．

1つのBINテープについてチェックサムを印刷後PC＝254にHALTする．

CONTINUEで次のBINテープについて印刷する．

I.D. BIN プログラム自体の使用例

```
I.D. BIN:  FEB 22, 1973

 0200
 0000/ 0160
 0200/ 0372
 0373/ 0377
 0162/ 0177

CHECKSUM 0000

ENTRY 0200
```

§6　基本保守診断

保守診断プログラムはＣＰＵをはじめ，システムを構成する機器の各々についてその機能をすべてチェックすることができる．システムの保全，故障個所の発見が容易におこなうことができ，異常がなければ一応正常な機能を果していると考えられる．

1．基本保守診断

ＣＰＵ，主記憶装置およびテレタイプの動作診断用で次のようなテストがおこなわれる．

1．命令テストNO. 1
2．〃　NO. 2
3．〃　NO. 3　すべての命令群の実行テスト，間接アドレス
4．〃　NO. 4　オートインデックス機構等をテストする．
5．〃　NO. 5
6．〃　NO. 6
7．電源接断に対するメモリ保護のテスト：電源ON/OFF時のメモリ保護をチェックする．
8．メモリアドレステスト：コアのロケーション（アドレス）をチェックする．
9．メモリチェックボードテスト：コアのメモリエラーをチェックする．
10．テレタイプテスト　NO. 1：テレタイプのリーダおよびパンチャをチェックする．
11．テレタイプテスト　NO. 2：印字，キーボードおよび総合テスト

2．高速紙テープリーダ，高速紙テープパンチャ，テスト

高速紙テープリーダおよびパンチャの機能テストおよびコントロール部のテスト，増幅器のテスト等をおこなう．

テスト　1　高速紙テープリーダおよびリーダコントロールに関する基本的なテスト
〃　2　高速紙テープパンチおよびパンチコントロールの基本テスト
〃　3　特殊な紙テープを使ったリーダテスト
〃　4　特殊な紙テープをパンチするテスト
〃　5　テスト4のパンチしたテープをチェックする．
〃　6　ランダムなCHARACTERをパンチさせるテスト
〃　7　テスト6のパンチしたテープをチェックする．
〃　8　特殊な紙テープのパンチとリードの結合テスト
〃　9　リーダ増幅器の調整テスト
〃　10　SRにセットした任意のCHARACTERをパンチさせるテスト
〃　11　特殊な紙テープをパンチさせる．
〃　12　高速紙テープリーダのスピード測定
〃　13　高速紙テープパンチャのパンチスピード測定
〃　14　任意の長さのブロックを，任意の待ち時間でブロック読みをするテスト
〃　15　高速紙テープリーダのリーダ速度を測定
〃　16　高速紙テープパンチャのパンチ速度を測定

3．その他

3－1　カセットシステムテスト各種
3－2　磁気ドラムテスト各種
3－3　拡張メモリ用テスト各種
3－4　ラインプリンタテスト各種
3－5　磁気テープ装置（IBMコンパチブル）テスト各種

§7　その他

1. インタプリタパッケージ

CEC555紙テープシステムには2つの浮動小数点演算パッケージ，2つの3倍長整数演算パッケージ，倍長演算パッケージが用意されている．これらの5つのパッケージはいずれもインタプリータ形式で，通常のアセンブラ言語に類似の命令群によるプログラミングを可能としている．

浮動小数点演算パッケージ，および3倍長整数演算パッケージには，各々単一フィールド用と複数フィールド用の2つのバージョンを目的に応じて使うことができる．

(1) 浮動小数点パッケージ(A)
　単一フィールド用，指数部24ビット仮数部24ビット，精度は10進6桁．

(2) 浮動小数点パッケージ(B)
　複数フィールド用，指数部8ビット仮数部28ビット，精度は10進8桁．

(3) 3倍長整数演算パッケージ(A)
　単一フィールド用，精度は10進10桁．

(4) 3倍長整数演算パッケージ(B)
　複数フィールド用，精度は10進10桁．

(5) 倍長整数パッケージ
　単一フィールド用，精度は10進6桁．

2. コンパイラ

CEC555紙テープシステムでは，2つのFORTRAN言語と2つのCOCOL言語を使うことができる．

4K FORTRANは，JIS-3000フォートランのサブセットであるが，必要最小限のフォーマット指定をそなえ，実行時の入出力タイプライタからの入出力について，懇切な配慮がある．また，実行の速度はアセンブラプログラムに比べ，ほとんど遜色がない．必要に応じアセンブラ言語で書いたサブルーチンを容易にリンクすることができる．

マークII　FORTRANは主記憶容量8K語～32K語を有効に利用できるように設計された本格コンパイラで，1パスのコンパイラ，1パスのアセンブラ，1パスのリンキングローダから成る．言語仕様はJIS-3000フォートランのスーパーセットで，特にソースプログラムに直接アセンブラ命令文を混用できるので，従来その類をみない豊富な応用可能性をもっている．リンキングローダは主プログラムと別に用意したサブプログラム，ライブラリプログラムなどを必要なだけロードして利用できる．

COCOLは特に入出力タイプライタを通じた会話的な用途に適したインタプリータ形式でフォートラン式の簡易言語で，最低2時間程度の講習で実用できる特長がある．

命令語はわずか12個に過ぎないが，これらを駆使することにより，非常に高度の用途にも充分こたえられるだけの機能をそなえている．COCOLはコアレジデントのソースプログラムと，これのオンラインエディティング，オンラインデバッグなどの機能がある．ソースプログラムは紙テープに保存，再使用なども容易にできる．

❸ マークIIモニタシステム〈その1〉

1.はしがき

CEC555電子計算機のモニタ（マークII）は，補助記憶装置と8K語以上の主記憶装置を持つシステムのためのファイル管理プログラムである．モニタ（マークII）は上記のシステムを使ったプログラム開発のための有用な武器であり，アプリケーションシステム構成のためには，能率的な構成要素の一つとなる．

モニタ（マークII）は従来の130K語の磁気ドラムのためのモニタの能力を大幅に改善したもので，以下のような特長を持っている．

1. プログラムに対する入出力機器の割当は，そのプログラムの実行段階で行うことができる．したがって，プログラムは，入出力機器とは無関係に開発できるし，ある特定の入出力機器について開発したプログラムは，他の入出力機器に対してもほとんどすべて，モニタ（マークII）のもとで実行可能である．
2. 入出力機器をプログラムで参照するには，システム構成から決まっているデバイス名または内部デバイス番号で行うことができる他に，任意のデバイス名を割当ててこれを使うことができる．
3. システムプログラムの設計に当っては，デフォールト（DEFAULT）の概念を，できる限り採用し，必要な指定を最少限にとどめるようにしてある．
4. 大容量の磁気ディスクカードリッジ装置の性能を有効に利用できるように配慮し，磁気ディスクへのアクセス回数をできるだけ少なくするような用意がある．
5. 固定ヘッド磁気ディスクの高速アクセス，磁気ディスクカートリッジの大容量，フロッピ磁気ディスクの簡便性などの諸特長を目的に応じて選定，組み合わせることができる．
6. 紙テープを全然使用しないシステム運用にも対処できる．
7. マークIIフォートランはJIS3000を上回る性能があり，コンパイル，アセンブル，リンクエディト，実行が自動的になった（1パス）．
8. マークIVフォートランはJIS7000を上回る性能があり，16レベルの自動オーバレー，システムストレージの配列に対する自動アクセスなどの機能を備えている．

本説明書の第1部ではモニタ（マークII）を使ってシステムライブラリおよび，これによって開発したユーザプログラムを実行するために必要な事項を説明する．モニタ（マークII）を使って，ファイル管理を行うためのプログラム技法については第2部で説明する．

2. システム構成

2—1 機器構成

モニタ（マークII）はシステム用の補助記憶（システムデバイス）として，CFD-130磁気ドラムまたはDRISK-2002カートリッジ磁気ディスク装置，CFD-300固定ヘッド磁気ディスク，CFP-フロッピ磁気ディスクのいずれかを使用する．将来，システムデバイスとして適当な補助記憶が開発されれば，その時点で，それらを使用するためのソフトウェアを供給する．

モニタ（マークII）のシステム構成に利用できる機器は次の通りである．

1. 電子計算機

555D/555Hで主記憶8K語～32K語

2. 入出力タイプライタ

CS-1200, CS-100, CS-1000, CMD-8

3. システムデバイス

DRISK-2002磁気ディスク装置, またはCFD-130, CFD-300, CFD500

4. その他の補助記憶（オプション）

(1) DRISK-2002形磁気ディスク装置（最大6.4M語）

(2) CFD-500形フロッピ磁気ディスク装置（1～4台）

(3) CFD-300形固定ヘッド磁気ディスク装置（130K語または260K語×最大4台）

(4) CMT-5形磁気テープ装置（1～4台）（最大3.2M語×4）

(5) CMT-8磁気テープ装置

(6) CMT-100形カセット磁気テープ装置（1～8台）（最大130K語×8）

5. その他の周辺機器（オプション）

(1) CMD-8キャラクタディスプレイ

(2) 紙テープリーダ

CS-408形

(3) 紙テープパンチ

CS-420形

(4) ラインプリンタ

CLP-4100形, CLP-4100A形, CLP-3000形

(5) カード読取機

2—2 プログラム構成

モニタ（マークⅡ）は

1. システムレパートリを受持つキーボードモニタ（KM）
2. システムライブラリやユーザプログラムにキーボードから与える指令を受けいれるための補助プログラムであるコマンドデコーダ（CD）
3. システムライブラリやユーザプログラムがファイル管理を行うための補助プログラムであるユーザサービスルーチン（USR）
4. 入出力機器との間のデータ転送を行うためのサービスプログラムであるI/Oドライバ（各機器ごとに用意される）
5. システムライブラリ（エディタ, アセンブラ, ローダ, ファイルユーティリティ, フォートラン等）
6. KM, CD, USRなどをスタートさせるためのコア常駐部分（シスレジデント）から成り立つ.

モニタ（マークⅡ）のシスレジデントは, フィールド0および1の最終ページ（7600～7777）計256語に過ぎない. その他の部分はすべてユーザプログラムに使用できる. したがって, 従来のドラムモニタまたは紙テープベースシステムで使用したプログラムはほとんどすべて, モニタ（マークⅡ）のもとでも実行可能である. モニタ（マークⅡ）の下でKM, CD, USR, PODなどが動作する時には0フィールドおよび1フィールドの0000～1777番地を使用するが, その都度, 主記憶とシステムデバイスとの間で自動的にスワッピングを行っているので, ユーザがこれらの主記憶を使用する支障とはならない.

2—3 用語の説明

2—3—1 語, ページ, ブロック

12ビットのデータを1語といい, CEC555計算機の通常の処理の単位である. モニタ（マークⅡ）のI/Oドライバは128語の整数倍の語数のデータ転送を行うように設計されており, 128語を1ページと呼ぶ. モニタ（マークⅡ）の管理するファイルは256語（2ページまたは1ブロック）単位の

構成をとり，ファイルの大きさはブロック数で表示される．ファイルとのデータ転送に当っては，奇数ページの転送は，各ブロックの先頭からのみが許され，最終ブロックの半端は0が詰められるか（ファイルへの転送），無視される（ファイルからの転送）．これらのデバイス名はシステムの中の表に登録されており，KM，CD，USRなどを通じて参照できる．

2—3—2　デバイス名

モニタ（マークⅡ）の下では各入出力機器は以下のデバイス名で呼ばれる．

SYS	システムデバイス CFD-130磁気ドラムまたはDRISK-2002磁気ディスク，CFP-500，CFD-300． 磁気ディスクの場合はカートリッジ1個を論理的に2ユニットに分けその前半（IOD0またはIOD2）をシステムデバイスとする．
DSK	デフォールトデバイス（デバイス名を指定しない時のデバイス）磁気ドラムのシステムではSYS=DSK，磁気デスクのシステムではカートリッジを論理的に2ユニットに分けたその前半または後半（IOD0～3）
TTY	入出力タイプライタ，キーボード入力および印字
PTP	紙テープパンチ（CS-408） 高速紙テープパンチのないシステムでは入出力タイプライタのパンチ部
PTR	紙テープリーダ（CS-420） 高速紙テープ読取器のないシステムでは入出力タイプライタのリーダ部
LPT	ラインプリンタ（CLP-4100）
CDR	カード読取器
IOD0～7	磁気ディスク装置（CXD　1.6/3.2）
CFP0～9	
MTA0～3	磁気テープ装置（CMT-5）
CTA0～3	磁気カセット装置（CMT-6）
CTB0～7	磁気カセット装置（CMT-100）

```
                DIMENSION  K (10)
                DO  5      I＝1，10
SWAIT,          6131
S               JMP  WA1T
S               6136
S               DCA\J
                K(I)=J
5               CONTINUE
```

1命令文の終り

スペース，タブ	シンボルのセパレータ
LF，FF	リスティング以外には無視
RUBOUT	無視

上記以外の文字はコメントの中（／からCRまで）ASCII定数（”の直後），TEXTの中（TEXTの次の次の文字から，同じ文字まで）以外では不正文字となる．

2—3—3　ファイル構成デバイス

SYS，DSKおよびIOD0～7，CFP0～3の各デバイスでは各ブロックごとにブロック番号が割当てられており，任意のブロックに直接アクセス可能である．これらのデバイスをファイル構成デバイスと呼び，登録されたデータは通常ファイル名で参照される．

その他のデバイス上のデータは通常はファイル名がなく，1回のアクセスについて，その都度用意されたファイルを1個だけ使用できる．例外として，イメージ形式のデータでは紙テープや磁気テープであってもファイル名が付属するものとして取扱われる．

2—3—4　ファイル名および形名

ファイル構成のデバイスに登録するデータは最少1ページからデバイスの許す最大の大きさまで任意することができ，この一連のデータはファイル名および形名で参照される．

ファイル名は1字以上の英数字から成り，第7番目以下の文字は無視される．形名はあってもなくてもよく，ファイル名の後にピリオッドとそれに続く英数字2字とする．形名は，これもファイル名の一部として，ファイル名が最大8字として扱ってもよい．システムプログラムでは，以下のデフォールト形名を想定することによって，キーボードからの入力を省略できるようにしてある．

形　名	説明(デフォールト形名)
.SV	コアイメージファイル，モニタレパートリのR,RUX,SAVE
.SA	SAP-Dアセンブラソース
.RA	RAP-D アセンブラソース
.FT	フォートランソース
.BN	バイナリ (LOAD 入力およびSAP-D 出力)
.RL	リロケータブルバイナリ (LLDR 入力およびRAP-D 出力)
.LS	リスティング (SAPD および RAPD リスティング出力)
.MP	ローディングマップ (LLDR 出力)
.TM	一時ファイル フォートラン出力およびRAPD出力，その他，上記以外のすべての場合

システムレパートリおよびシステムプログラムでは入力に指定されたファイル名に形名のない時にはデフォールト形名を付加したものを入力とする．

システムレパートリおよびシステムプログラムでは出力ファイル名に形名の指定がない時には，デフォールト形名を自動的に付加する．たとえばSAVE SYS PROGAとすると，PROGA.SVとして登録される．もちろん，SAVE SYS PROGA. ××も可能である．

形名のないファイルはCECの供給するシステムプログラム，またはアプリケーションプログラムに限って用いられる．システムレパートリのR，RUN，GETを除き，すべてのシステムプログラムおよびユーザのプログラムで形名のないファイルを参照することは許されない．

唯一の例外は，CECの供給するイメージ形式の紙テープを，SYS BUILDER，またはIMAGEプログラムの/Rオプションで，ファイルに読み込む場合である．

2—3—5　ダイレクトリ（DIR）

ファイル構成デバイスには，ダイレクトリと名づける特殊ブロックがあり，デバイスに収容しているファイル名とその大きさ，日付けを記録している．ダイレクトリの内容はファイルユーティリティのCONVRTで出力することができる．

2—3—6　システムデート（DATE）

システムレパートリのDATEコマンドで更新され，システムレパートリのSAVEコマンドやその他のシステムライブラリ，ユーザプログラムでも使用できる．

2—3—7　ジョブステータスワード（JSW）

SV形のファイルとして登録されたファイルについて、KM、USR、CD、PODの実行に際して、コアのスワッピングの必要の有無を指定する.

2—3—8　コアコントロールブロック（CCB）

SV形のファイルの最初のブロックをコアコントロールブロックと呼び、第2ブロック以下のデータのコア上の位置、スタート番地、JSWなどを保存する.

3.　システムレパートリ（KM）

モニタ（マークⅡ）の通常のスタート番地は0フィールドの7600番地で、ここから計算機をスタートさせること. プログラムは必要なコアの保存を行った後、入出力タイプライタにピリオッドを印刷して、ユーザのコマンドを待受ける状態となる. ユーザコマンドを処理するプログラムをキーボードモニタ（KM）と呼び、次の9つのシステムレパートリを持っている.

.DATE
.SAVE
.GET
.START
.RUN
.R
.ASSIGN
.RESET
.POD

システムユーティリティ、I/Oドライバおよびシステムライブラリ中のプログラムが動作している時には、常にコントロールC（↑Cとタイプライタに印字される. 以降↑Cと表示する）のキーインによりキーボードモニタに戻る.

3—1　コマンド行

キーボードモニタに対するコマンド行は、コマンドとこれに続く0～2個の指定とから成り、そのキーイン、修正が容易なように配慮されている.

コマンドは最初の2字だけがモニタによって解釈され、以後は冗長である. コマンド行はRETURN（CR/LFをエコー）またはESC（$とエコー）で終端され、キーボードモニタに渡される. RETURNまたはESCをキーインするまでは、キーインの誤りを容易に修正できる.

コントロールU（↑U）はキーインをすべてキャンセルする. RUBOUTはすでに行ったキーインを最後から1字ずつキャンセルする. RUBOUTに対しては1字ごとに＼がエコーされる. LFは修正して読み難くなった行を清書する. 清書された後さらにキーイン、修正、終端のいずれも自由である. キーボードモニタがコマンド行を解釈できない時はエラーメッセージを印刷する.

3—2　DATE　コマンド

DATEコマンドは次の書式による.

.DATE mm/dd/yy

ここでmmは1～12、ddは1～31、yyは74～77の10進数で各々月、日、年を表わす.

また、DATE↵ではシステムデートを印刷する.

例　DATE↵2/3/74

3—3　SAVE　コマンド

SAVEコマンドは次の書式による

.SAVE dev file. ext a—b,c……;,s=n

ここでdevはデバイス名（システム名でもユーザ名でも可）. file. extは形名を含めたファイル名

である．形名がなければ，ＳＶが付加される．a,b……は主記憶装置上のセーブすべき番地を表わし，フィールドを含め５桁で表示する．セーブする範囲はページ単位で処理され，そのページの１番地でも指定されれば，そのページ全部がセーブされる．また奇数ページはその直前の偶数ページを含めてセーブされる．－で連結されたセーブ範囲は２つのフィールドにまたがってはならない．，で併記された各範囲はオーバラップしてはならない．sはプログラムのスタート番地でフィールドを含め５桁で表示する．ｎはジョブステータスワード（JSW）で次の意味を持つ．

ビット０＝１　フィールド０の０－1777にプログラムが存在しない．

ビット１＝１　フィールド１の０－1777にプログラムが存在しない．

ビット２＝１　プログラムは再スタートできない．セーブコマンドでは指定があっても０

ビット10＝１　フィールド０の０－1777を保存する必要がない（CDおよびPOD）

ビット11＝１　フィールド１の０－1777を保存する必要がない（USR）

コアの範囲，スタート番地，およびJSWについては指定がなければ，システムのCCB（GETコマンド，RUNコマンド，LOAD，LLDRなどで作られている）の数値が用いられる．

例

```
.SAVE SYS XYZ
.SAVE SYS XYZ.PX
.SAVE IOD3 XY 55, 10100－11005;10200＝0
```

（セーブされる範囲は00000－00177，10000－11177）

3－4　GETコマンド

GETコマンドは次の書式による

.GET dev file.ext

または .GET dev file

GETコマンドはコアイメージファイル（SAVEコマンドで作ったファイル）をコアにロードする．形名のない時には形名のないファイルをさがし，なければSVと解釈する．JSWを含めたCCBはシステムに保存され，次のSAVEコマンドに使用される．

GETコマンドでロードされたコアイメージはSTARTコマンドで実行するか，PODで修正する．SAVEコマンドで新しいファイルに登録するなどに用いられる．

3－5　STARTコマンド

START　コマンドは次の書式による．

.START nnnnn

nnnnnはフィールドを含め５桁のスタート番地である．nnnnnがない時はCCBのスタート番地が用いられる．

3－6　RUNコマンド

RUNコマンドは次の書式による．

.RUN dev file.ext

または .RUN dev file

RUNコマンドはGET＋STARTを行う．形名のない時は.SVと解釈する．

3－7　Rコマンド

Rコマンドは次の書式による．

.R file

または .R file. ext

これは .RUN SYS fileに類似しているがCCBが保存されない点が異る．

3－8　ASSIGN

ASSIGNコマンドはシステムデバイスにユーザネームを割当てたり，割当てたユーザネームを除いたりする．書式は次の通りである．

(1) ASSIGN dev udev

(2) ASSIGN dev

(1)はシステムデバイスdevにユーザネームudevを割当てる．(2)はシステムデバイスdevにもし割当てられたユーザネームがあれば，これを取除く．

デバイスネームは内部では1語にパックしているので1字または2字のネームについては問題が起らないが，3字または4字のユーザネームはすでに登録されているネームとの混同の起きる危険性がある．すでに登録されているネームと重ならないことのチェックは，

(3) ASSIGN udev

としてudev NOT AVAILABLEのメッセージが印刷されればシステムネームとの混同は起きないことが保証できる．メッセージがなければ(3)は(2)と同じである．

ユーザネームとしてシステムネームを使用することは自由である．例えばここにTTYを想定したプログラムがあり，この出力をLPTに一時的に変更したいとすれば，単に，

.ASSIGN LPT TTY

としてからそのプログラムを実行すればよい．ここではLPTもTTYもLPTを参照する．さらに，上に加えて，

.ASSIGN TTY PTP

とすれば，プログラムで指定するTTY，PTPの代りに各々LPT，TTYが使用される．

3－9 RESET

書式は次の通りである．

.RESET

デバイスネームはすべてシステムネームのみとなる．

またPODでブレークポイントを使用した時，ユーザネームテーブルがスクラッチとして使用されたのをクリアするのにも用いられる．

3—10 POD

PODはPhantom Octal Debuggerの意味で.PODで呼び出される．

3—10—1 PODの概要

PODは機能的にはCODとXODの中間に位するが，ブレークポイント機能を用いる時以外はユーザのコアエリアを全然消費しない点が優れる．ブレークポイント機能を用いる場合も，ブレークポイントを設定したフィールドの4，5，6番地を要するだけである．

1. PODはコアを要しない（CODは3ページ，XODは5ページ要する）
2. 複数フィールドで使用できる（XODと同等，CODでは不可）
3. ブレークポイント設定時にはそのフィールドの4，5，6番地を使用する
（XODに同じ，CODでは4番地のみ）
4. 割込機能を用いるプログラムのデバッグには利用できない（CODに同じ，XODよりも劣る）．
5. ブレークポイントにおけるTTYフラッグを保存する（XODに同じ，CODでは保存しない）．
6. XODのVコマンドがない（CODに同じ）．
7. XODの%コマンドがない（CODに同じ）．
8. CODのPコマンドがない（XODに同じ）．

その他

9. ユーザネームを使うプログラムのデバッグには利用できない．

3—10—2　PODコマンド

mnnnn	mフィールドのnnnn番地オープン
⤶(CR)	クローズ
↓(LF)	クローズして次の番地オープン
↑(シフトN)	クローズして参照番地オープン
←(シフトO)	クローズして内容番地オープン (新しくオープンされる番地のフィールドはFレジスタの内容に等しい)
n＋	クローズして．＋n番地オープン
n－	クローズして．－n番地オープン
mnnnnB	mフィールドのnnnn番地にブレークポイント設定
B	ブレークポイント除去
mnnnnG	mフィールドnnnn番地にジャンプ
C	ブレークポイントからの再スタート
nC	ブレークポイントから再スタートして新しいブレークポイントをn回通過した後にPODに戻る．
D	ブレーク時のデータフィールドオープン
F	ブレーク時のインストラクションフィールドオープン．またこのレシスタは←およびWコマンドに関係する．また，DおよびFでフィールド指定はn0とすること．
A	ブレーク時のACオープン
L	ブレーク時のLINKオープン
M	Wコマンドのためのマスクオープン，さらにLFでサーチ下限番地，またLFでサーチ上限番地オープン
nnnnW	ワードサーチ，コア内容にMをかけた値がnnnnに等しければアドレスと内容を印刷する．
↑O	印刷を中断してPODコマンドモードに戻る．
↑C	モニタに戻る．

3—10—3　POD使用上の注意

ブレークポイントを設定した時に，PODとユーザプログラムとの間のスワッピングの要，不要はJSWのビット10による．ビット10が0のプログラムでブレークポイントを使用すると，かなり時間のかかるのは避けられない．特にnCとした時にこれは目立つ．

また当初0フィールドの0～1777を使用していなかったプログラムに対し，0～1777にパッチを追加した時にはJSWを修正する必要が生ずる．JSWは07746番地にありPODでオープン，変更ができる．またCCBの中の実行開始番地（SAVEの時の;s）は07744および07745番地に保存されており，この2つの番地もPODで変更できる．

ブレークポイントはCIF命令およびこれの後のJMP，またはJMS命令実行後までに設定してはならない．

4.　システムユーティリティ（USR，QCD）

モニタ（マークⅡ）では，システムユーティリティとして，ファイル管理およびI/Oドライバの管理を行うためのサブルーチンの集合をユーザサービスルーチン（USR）となづけている．

USRの仲介により，システムライブラリプログラムおよびユーザプログラムは入出力ファイルに関するユーザ（人間）の指定を受け取ること，さらにファイルのダイレクトリを管理すること，

プログラムが必要とするI/Oドライバの管理などを容易に行うことができるようになっている．USRの詳細レパートリおよびその使用技術の説明は本書の第2部にゆずり，ここではシステムライブラリプログラムを使うために最少限必要なコマンド行の約束について述べる．

システムライブラリプログラムのすべては動作後，ただちに*を行の左端に印刷してコマンド行のキーインを要求する．このコマンド行の要求とその解釈はUSRの中のコマンドデコーダ（QCD）機能にもとづく．

4—1　コマンド行の特殊キー

コマンド行をキーインするに当ってはKMに対するコマンド行と同様の約束に従う．これを下表に示す．

キ　ー	エコー	機　能
RUBOUT	＼	最後から1字ずつ消去
コントロールU	↑U	その行を全部消去
LF		清書を行う
コントロールC	↑C	.（KM）に戻る
RETURN	CR-LF	コマンド行処理
ESC	$	〃

4—2　コマンド行の要素

コマンド行は0～3個の出力ファイル指定，0～9個の入力ファイル指定と幾つかのオプション指定とから成り，出力ファイル,＜記号，入力ファイルの順にキーインされる．

4—3　出力ファイルの指定

出力ファイル指定はdev:file.ext〔nn〕の形で行う．デバイス名（dev）はシステムデバイス名またはユーザデバイス名とする．ファイル構成でないデバイスについては：以下は不要である．ファイル構成のデバイスについては通常ファイル名（file）を必要とする．ファイル形名（.ext）を省略した場合は，システムプログラムまたはユーザプログラムにデフォールト形名の約束があれば，自動的にその形名が付加される．〔　〕で囲まれた部分は出力のファイルの大きさについての指定であって1～377の8進数とする．ファイル管理プログラム（USR）はこの指定を満足する範囲で一番小さな空きブロックを割当てる．〔　〕を指定しない場合，および377よりも大きい数を指定した場合には空きブロックの中で一番大きいものが割当てられる．

出力デバイスにデフォールトデバイス（DSK:）を指定する場合には，出力デバイス指定を省略して，ファイル名のみ指定すればよい．

2個以上の出力ファイルを指定する場合には上記の指定の間に，，を入れて区切って指定する．プログラムが1個以上の出力ファイルを要求する場合には，順序関係が意味を持つのが通常であって，そのいずれかに無指定を選ぶこともあり得る．

出力ファイルの指定の最後には＜記号が必要である．

4—4　入力ファイルの指定

入力ファイルの指定は〔　〕指定がないことを除き，出力ファイルの指定に準ずるが無指定の取扱いが多少異る．

入力ファイルの指定は＜の記号の右に行うが，出力ファイル無指定の場合には＜記号を省略してもよい．

デバイス名の無指定は，最初の入力ファイルについてはデフォールトデバイス（DSK:），2つ目以降の入力ファイルについては直前の入力デバイスと同じデバイスを意味する．ファイル形名が無指定の場合には，プログラムで規定するデフォールト入力形名を付加して，入力ファイルを探す．

(注) 正確には，コマンドデコーダが直接行うのは入力ファイルおよび出力ファイルについてデフォールトデバイスの割当てと通常モードで入力ファイルについてデフオールト形名の付加とである．その他のデフオールトやファイル長の指定などはコマンドデコーダの出力に基づいて，システムプログラム，またはユーザープログラムがUSRを使って行う．

4—5 文字オプション

コマンド行の機能としては英字（26個）と数字（10個）を受けいれる必要がある．各オプションの解釈はライブラリプログラムまたはユーザプログラムの規定に従う．

オプション指定には2つの方法が許されている．第1の方法では / の後の1文字はオプション指定と解釈される．第2の方法では（から）まで任意の文字数のオプション指定が許される．なお，これらのオプション指定は，入出力ファイル指定のストリングと無関係で，その中間の任意の個所に指定することも許される．

4—6 オクタル数オプション

オクタル数オプションは最高23ビットのオクタル数を指定することを許す．オクタル数オプションは＝を入力し，これに続けた0～7の数字ストリングで指定する．＝0は無指定に同じである．

オクタル数オプションの解釈はこれを受けいれるライブラリプログラムまたはユーザプログラムの規定に従う．

4—7 終端記号オプション

コマンド行はRETURNまたはESCによって終端されてプログラムに渡される．RETURNはCR/LF,ESCは $ とエコーされ，この終端条件もオプションの一つとしてプログラムに渡される．このオプションの解釈も，受けいれるプログラム次第であるが，通常の約束では，RETURNはさらにコマンド行が続くことを表し，ESCはコマンド行の指定がこれですべてであることを表す．

4—8 コマンド行の例

* LPT：＜SYS：INPUT.LS/A
  出力はラインプリンタ，入力はシステムデバイスのINPUT.LSファイル，オプションA
* OUT＜PTR：,，X，SYS:Y，Z
  出力はDSK:OUT，入力はPTR:,PTR:,PTR:X,SYS:Y,
  SYS:Z（PTR:XはPTR:に同じである）
* X/DX＜
  出力はDSK:XX，オプションD，入力はなし
* XYZ,ABC,IOD 2 ：XY,YZ
  出力は指定なし，入力はDSK:XYZ，DSK:ABC，IOD2:XY,
  IOD 2 ：YZ

5．ファイルユーティリティ（CONVRT、IMAGE）

5－1　CONVRT

5－1－1　概要

CONVRTはモニタMARKⅡの下で働き，デバイス間のファイル転送，ダイレクトリーの印刷，ファイルの抹消etc.の機能を持つユーティリティプログラムである．

5－1－2　CONVRT　プログラムの登録

CONVRTは次の方法で登録する．

・R␣LOAD↵

*PTR：（89）＝12000＄

↑（↑（upper arrow）に対しては任意の文字，例えば␣(space）をキーインする）

・SAVE␣SYS␣CONURT↵

5－1－3　CONVRTプログラムの実行

CONVRTは

・R␣CONVRT↵

で呼び出されて，「＊（asterisk)」を印刷し，以下に述べる書式に従ったコマンドを，入出力タイプライタより受け入れる．

（コマンド・デコーダ（QCD）解説書参照）

指定された動作が完了すると，再び「*」または「・」に戻る．

（コマンド・ラインの終端文字が「CR」ならば「*」に，「ESC」ならば「・」に戻る）

コマンド指定については，出力指定1個と最大5個の入力指定およびオプション指定を原則としている．

5－1－4　コマンド詳説

1．ダイレクトリーの印刷1（オプション/D）

次の書式でコマンドを与える．

*output　device:（file-name.ext）＜input device:/D

(1) 指定された入力デバイスのダイレクトリーをすべて印刷する．

(2) 入力デバイスの指定は最大5個までである．

(3) 入力デバイス指定で，1番目の入力デバイスについては無指定ならばDSK：が選ばれる．

(4) 出力デバイスが無指定ならば，入出力タイプライタ（TTY）が出力デバイスとなる．

(5) 印刷されるファイル長はオクタル表示である．

例

*/D……DSK：のダイレクトリーをTTY：に印刷

*LPT:＜/D……DSK：のダイレクトリーをLPT:に印刷

*LPT:＜IOD0:，IOD1:，IOD2:，IOD3:/D

IOD0～3のダイレクトリーをLPT:に印刷

*IOD0:DIREC.DT＜SYS:，DSK:/D

SYS:およびDSK:のダイレクトリーをIOD0のDIREC.DTに格納する．

```
.R CONVRT
*/D

 6/18/75
--
  IMAGE          11    2/ 6/75
  CONVRT         13   11/29/74
  SQZ             5    2/ 5/75
  EDIT           14    6/ 6/75
  SAPD           15    9/20/74
  DUPE            3    8/22/74
  BOOT            2    1/17/75
  BOOTBN          2    2/15/75
  HEXE   .BN     17    6/18/75
  DESI   .BN     16    6/17/75
  PEN    .SA     60    6/17/75
  NANO   .SA     43    6/17/75
  <EMPTY>      5677
```

2．ダイレクトリーの印刷2（オプション/D）

次の書式でコマンドを与える

output device:＜Input device:. ext/D

(1) 指定された入力デバイス上の指定された形名（.ext）が合致するすべてのファイルのダイレクトリーデータを印刷する.

(2) 入力デバイスの指定は最大5個までである.

(3) 入力ファイルの第1番組のファイル名は「＊（asterisk)」とする.
入力デバイスを複数個指定する場合には，第1番目のファイル名だけ「＊」を指定すればよい.
（入力デバイスが複数個の時，各デバイスに対して違った形名を指定することはできない）

(4) .extは形名の指定である.

(5) 入力デバイス指定で，1番目のファイルについては無指定ならばDSK:が選ばれる.

(6) 出力デバイスが無指定ならば，入出力タイプライタ（TTY:）が出力デバイスとなる.

例

**.SA/D……DSK:上の形名．SAを持つすべてのファイルのダイレクトリー・データをTTY:に印刷.

LPT:＜IOD0:. BN, IOD1:/D
……IOD0およびIOD1上の形名BNを持つすべてのファイルのダイレクトリーデータをLPT:に印刷.

注. ここでファイル名を「*」とした代りに，任意の名前を付けて実行することも可能である．この場合には「/D」に加え，「/X」を指定する.

**.SA/Dは

*TEM.SA/D/X

と同値である．ここでファイル名「TEM」は何の意味も持たない.
仮のファイル名である.

```
.R CONVRT
**.SA/D

 6/18/75
--
  PEN    .SA    60    6/17/75
  NANO   .SA    43    6/17/75

.R CONVRT
**.BN/D

 6/18/75
--
  HEXE   .BN    17    6/18/75
  DESI   .BN    16    6/17/75
```

3．指定ファイルの有無（オプション/D）

次の書式でコマンドを与える．

*Input device:file-name.ext/D

(1) 指定された，入力デバイス上の指定ファイルを検索し，そのファイルの有無を調べる．

(2) 入力デバイスの指定は1個のみであり，入力デバイス指定が無指定ならばDSK:が選ばれる．

(3) .extは形名の指定であり，無指定ならば「.SA」として扱われる．

(4) 指定ファイルが存在するならば入出力タイプライタに

FOUND

と印刷する．存在しなければ，

NOT FOUND

を印刷する．

例

*CONVRT/D……DSK:上のCONVRT.SAの有無を調べる．

*CONVRT.BN/D……DSK:上のCONVRT.BNの有無を調べる．

*IOD0:CONVRT/D…IOD上のCONVRT.SAの有無を調べる．

```
.R CONVRT
*NANO.SA/D

FOUND !*NANO.LS/D

NOT FOUND*
```

4．ヘッダ付加（オプション/L）

以下に述べるファイル転送にオプションLを複合させるとビジブル・ヘッダーを付加した後，実際のファイル転送を行う．

コマンド・ラインで「/L」を指定すると，テレタイプに「# (SHARP)」を印刷し，以下「CR」までの文字をヘッダーとして取り扱う．

例

*……/L

＃CEC—555

「CEC—555」がヘッダーとして扱われる.

5．ファイル転送1（オプション無指定）

次の書式でコマンドを与える.

*output device:file-name.ext

＜input device:file-name.ext

(1) すべての形式のファイルをそのままの形で転送する.

(2) 入力ファイル指定は，1個のみ有効とし，入力デバイス指定がない場合には，DSK:が選ばれる.

(3) 出力デバイス指定がない場合にはPTP:が選ばれる.

(4) 形名指定がない場合には，入力，出力とも「.SA」として扱われる.

例

*CONVRT↵……PTP:にDSK:上のCONVRT.SAを出力する.

*IOD0:CONVRT＜CONVRT

……DSK:上のCONVRT.SAをIOD0にコピーし，CONVRT.SAというファイル名を与える.

*SYS:CONVRT.BN↵……SYS:上のCONVRT.BNをPTP:に出力する.

6．ファイル転送2（オプション/T）

次の書式でコマンドを与える.

*output device:file-name.ext

< Input device:file-name.ext/T.

ASCII形ファイルの印刷に用いる．オプションTによる転送では，

(1) オクタル値200以上のコードが入力される以前の200以下のコードはすべて無視される.

(2) オクタル値0，177，200，377の各コードは無視される.

(3) FF（オクタル214）は，LF（オクタル212）9個に変換される.

(4) VT（オクタル213）は，LF 5個に変換される.

(5) TAB）オクタル211）は，space（オクタル240）8個に換算される.

(6) ファイルの終りは，CONTROL Z（オクタル232）による.

(7) 入力ファイル指定は1個のみ有効とし，入力デバイス指定がない場合には，DSK:が選ばれる.

(8) 出力デバイス指定がない場合には，PTP:が選ばれる.

(9) 形名指定がない場合には，入力，出力とも形名「.SA」として扱われる.

例

*TTY:＜CONVRT/T……DSK:上のCONVRT.SAをTTY:に印刷する.

*LPT:＜IOD0:CONVRT.SA / T

……IOD0:上のCONVRT.SAをLPT:に印刷する.

7．ファイル転送3（オプション/A）

次の書式でコマンドを与える.

*output device:file-name.xt

-< Input device:file-name.ext/A/L.

ASCII形のファイルについての転送に用い，複数個のファイルの合成ができる.

オプションAによる転送では

(1) オクタル値200以上のコードが入力される以前の200以下のコードはすべて無視される.

(2) オクタル値0，177，200，377の各コードは無視される.

(3) 出力が紙テープの場合はFF（オクタル214）の後にブランク（オクタル0000）10個を付け加えると共にファイルの前後にブランク・フィードを付け加える.

(4) ファイルの終りはCONTROL Z（オクタル232）による.

(5) 入力ファイル指定は最大5個とし，1番目のファイルについて，入力デバイス指定が無指定ならばDSK:が選ばれる.

(6) 出力デバイス指定がない場合にはPTP：が選ばれる.

(7) 形名指定がない場合には，入力，出力とも形名「.SA」として扱われる.

(8) /Lオプションを複合できる.

例

*CONVRT/A/L……ビジブル・ヘッダーを付加し，DSK:上のCONVRT.SAをPTP:に出力する.

*TTY：<CONVRT / A / L……TTY:出力の場合

*X3<X1，X2 / A……DSK：上のX1.SA.とX2.SAを合成し，X3.SAを作り出す.

8．ファイル転送4（/B）

次の書式でコマンドを与える.

*output device:file-name.ext

<Intut device:file-name.ext/B/L.

バイナリー形のファイルについて，ヘッダー部を取り除き，規格化されたリーダおよびトレーラを付加して転送する.

(1) 入力ファイル指定は1個のみ有効とし，入力デバイス指定がない場合にはDSK:が選ばれる.

(2) 出力デバイス指定がない場合にはPTP:が選ばれる.

(3) 形名指定がない場合には入力，出力とも形名「.BN」として扱われる.

(4) ファイルの終りはCONTROL Z（オクタル232）による.

(5) /Lオプションを複合できる.

例

*SYS:CONVRT.BN/B/L

……新しいビジダブル・ヘッダーを付加しSYS:上のCONVRT.BNをPTP:に出力する.

*CONVRT<PTR:/B……BIN形式の紙テープをDSK:上のCONVRT.BNに登録する.

9. ファイル抹消1（/E）

次の書式でコマンドを与える.

*Input device：file-name.ext/E.

指定されたファイルを抹消する.

(1) 入力ファイル指定は最大5個とし，1番目の入力デバイス指定がない場合には，DSK：が選ばれる.

(2) 出力ファイル指定は無視される.

例

*CONVRT・BN/E…DSK：上のCONVRT・BNを抹消.

*X1.SA，X2.SA，X3.SA/E…DSK：上のX1.SA，X2.SA，X3.SA，を抹消する.

*IOD0：X1.SA，IOD1：X2.SA/E…IOD0：上のX1.SAおよびIOD1：上のX2.SAを抹消する.

10. ファイル抹消2（/E）

次の書式でコマンドを与える.

**. ext/E

↓

(file-name)

指定された形名の合致するすべてのファイルを抹消する.

(1) 入力ファイル指定は1個のみ有効とし，そのファイル名として「* (asterisk)」を与える.

(2) extにより，形名を指定する.

形名が「.SA」の時には，入出力タイプライターに SURE ?をプリントするから「Y (yes)」または，「N (No)」をキインする.

(3) 入力デバイス指定がない場合には，DSK：が選ばれる.

(4) 出力ファイル指定は無視される.

例

**.LS/E…DSK：上の形名「.LS」を持ったファイルをすべて抹消する.

注 ここで，ファイル名「*」とした代りに任意の名前を付けて

この場合には「/E」に加えて「/X」を指定する必要がある.

上の例は

*TEM.LS/E/X

と同値であり，ここでファイル名「TEM」は意味を持たない仮のファイル名である.

```
.R CONVRT
*HEXE.BN/E
*/D

 6/16/75
--
  IMAGE          11    2/ 6/75
  CONVRT         13   11/29/74
  SQZ             5    2/ 5/75
  EDIT           14    6/ 6/75
  SAPL           15    9/20/74
  DUPE            3    8/22/74
  BOOT            2    1/17/75
  BOOTBN          2    2/15/75
  <EMPTY>        17
  DESI   .BN     16    6/17/75
  PEN    .SA     60    6/17/75
  NANO   .SA     43    6/17/75
  <EMPTY>      5677
```

```
.R CONVRT
**.BN/E

 6/16/75
--
  IMAGE          11    2/ 6/75
  CONVRT         13   11/29/74
  SQZ             5    2/ 5/75
  EDIT           14    6/ 6/75
  SAPL           15    9/20/74
  DUPE            3    8/22/74
  BOOT            2    1/17/75
  BOOTBN          2    2/15/75
  <EMPTY>        35
  PEN    .SA     60    6/17/75
  NANO   .SA     43    6/17/75
  <EMPTY>      5677
```

11. 最大空きファイルの印刷（/N）

次の書式でコマンドを与える.

*output device :<Input device : /N.

指定されたデバイスのダイレクトリーから最大の空きファイル（NULLファイル）を探し，その大きさをオクタル表示する.

(1) 入力デバイス指定は最大5個であり，入力デバイス指定がない場合にはDSK：が選ばれる.

(2) 出力デバイス指定がない場合にはTTY：が選ばれる.

(3) 入力，出力ともファイル名および形名は無視される.

例

*/N…DSK：上の最大NULLファイルの大きさをTTY：に表示する.

*IOD 0 :, IOD 1 :, IOD 2 : /N

```
.R CONVRT
*/N

MAX NULL FILE = 5677*
```

12. ヘッダープリント（/H）

次の書式でコマンドを与える.

*output device :<Input device : file-name.ext/H.

ASCII形ファイルの「# (SHARP) 行」を印刷する.

(1) 入力ファイルは最大5個とし1番目のファイルについてデバイス指定がない場合には，DSK：が選ばれる.

(2) 入力ファイルの形名が指定されなければ「.SA」として扱われる.

(3) 出力デバイス指定が無指定ならばTTY：が選ばれる.

例

*CONVT/H…DSK：上のCONVRT.SAの#行をTTY：にプリント.

```
.R CONVRT
*NANO.SA/H

#NANO  (69)=12000: JAN 1, 1975
```

13. ファイル転送5（/S）

次の書式によりコマンドを与える.

*output device : file-name.ext

<Input device : file-name.ext/S/L

コアイメージ，ファイルのBIN変換転送を行う.

(1) 入力ファイルは1個のみ有効とし，デバイス指定がない場合には，DSK：が選ばれる．形名指定がない場合には「.SV」として扱われる.

(2) 出力デバイス指定がない場合には，PTP：が選ばれる.

(3) /Lオプションを複合できる.

例

*CONVRT.SU/S/L

……ビジブル，ヘッダーを付加し，DSK：上のSV形式ファイルCONVRT.SVをPTP：にBIN形式でとり出す.

14. ファイル転送 6 （/F）

次の書式でコマンドを与える.

*output device : file-name.ext

〈Input device : file-name.ext/F/L.

COPY形式ファイルをASCII形ファイルに変換する.

(1) 入力ファイルは1個のみ有効とし，デバイス指定がない場合にはDSK：が選ばれる.

(2) 出力デバイス指定がない場合には，PTP：が選ばれる.

(3) /Lオプションを複合できる.

例

*PTR：/F/L…ビジブル，ヘッダーを付加し，PTR：よりCOPY形式のファイルを入力し，ASCII形をファイルに変換し，PTP：に出力する.

*COPY.SA＜PTR：/F…PTR：よりCOPY形式ファイル入力し，DSK：上のCOPY.SAにASCII形ファイルに変換し登録する.

5－2 IMAGE

5－2－1 目的

IMAGEは，モニターMARKⅡの下で働き，ファイル登録の能率化を目的として開発されたプログラムである.

5－2－2 概要

IMAGEは，以下に述べる4つの機能を持つ.

(1) IMAGE形式紙テープファイルの生成

(2) IMAGE形式紙テープファイルのヴェリファイ

(3) IMAGE形式紙テープファイルの登録

(4) IMAGE形式紙テープファイルのヘッダ情報の印刷

このIMAGEプログラムによって生成された，紙テープファイルの登録に当っては，オペレータの介入をできるだけ少なくするように設計されており，ファイル名指定等のダイレクトリー情報はあらかじめ規定された情報により登録される.

5－2－3 IMAGEプログラムの登録

IMAGEは次の方法で登録する.

・R LOAD↓

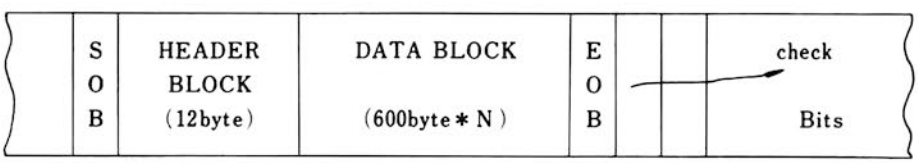
*PTR：（89）＝12000$

↑（↑（upper arrow）に対しては，任意の文字，例えば␣（space）を，キーインする）

・SAVE SYS・IMAGE.

5－2－4 イメージ，フォーマット

以下のような形式を持つファイルをイメージ形式と呼ぶ.



(16bits)

1. SOB (start of BLOCK)

ブロックの始りを示す.（SOB＝2018）

2. HEADER BLOCK

12byteのブロックでダイレクトリー情報を収納している．12byteは，3byte/2 Wに変換され，以下のダイレクトリー情報を示している．

	DEV #	
S_1	S_2	file −name
S_3	S_4	
S_5	S_6	
E_1	E_2	−extention −name
DATE		
−File Length		
0		(DUMMY WORD)

3. DATA BLOCK

ファイル形式デバイスに，登録すべきデータが3 byte/2 Wの形で，そのファイルのブロック数だけ存在する．

4. EOB (End of Block)

ブロックの終りを示す．(EOB=378_8)

5. CHECK BITS

16bitsのcheck bitsである．

5−2−5 IMAGEプログラムの実行

IMAGEは

・R␣IMAGE↵

で呼び出されて「＊(asterisk)」を印刷し，以上に述べるコマンドを，入力タイプライタより指定する．

(コマンド・デコーダ解説書参照)

なお，コマンド・ラインの終端文字が「CR」ならば「＊」に，「ESC」ならば「・(dot)」に戻る．

5−2−6 コマンド詳説

1. IMAGE形式紙テープ・ファイルの生成

次の書式でコマンドを与える．

＊DEVICE:FILE-NAME. EXT/I

(1) 入力ファイルは最大5個まで指定できる．

(2) Device指定が省略されている場合には，SYS：が選ばれる．

(3) 出力デバイスはPTP：固定である．

(4) /Lオプションを複合できる．

上述のコマンドに/Lオプションを複合させると，入出力タイプライタに「＃(SHARP)」をプリントし，以下CRまでのキーインされた文字をビジブル・ヘッダーとして扱う．

例.

＊CONVRT. SV/I/L↵

＃CONVRT IMAGE FORMAT↵…ビジブル・ヘッダー

＊

2. IMAGE形式紙テープ，ファイルのベリファイ

次の書式でコマンドを与える.

*/V

(1) デフォールト入力デバイスはPTR：である.

(2) 入力ファイルの終りは，PTR：のタイム・オーバーによる.

(3) チェックされたファイルに対しては，以下のメッセージをプリントする.

VERIFY␣OK␣FILE♯␣NNNN

VERIFY␣ERR␣FILE♯␣NNNN

FORMAT␣ERR␣FILE♯␣NNNN

（入力ファイルがIMAGE形式でない）

注，ＮＮＮＮは何番目の入力ファイルであるかを示す.

3. IMAGE形式紙テープ・ファイルの登録

次の書式でコマンドを与える

*DEVICE：</R

(1) 入力デバイスはPTR：固定である.

(2) 入力ファイルの終りはPTR：のタイム・オーバーによる.

(3) 出力デバイスの指定があると，指定デバイスに規定ファイル名を与えて登録する．出力デバイス指定がない場合には紙テープファイルのHEADER BLOCKに収納されているデバイスに登録する.

(4) リード・エラーが生じた場合には

READ ERR FILE♯ NNNN

をプリントし，「*」または「・」に戻る.

4. IMAGE形式紙テープ・ファイルのヘッダ情報の印刷

次の書式でコマンドを与える.

*/H

(1) 入力デバイスはPTR：固定である.

(2) 入力ファイルの終りはPTR：のタイム・オーバーによる.

(3) 以下のフォーマットでヘッダ情報を印刷する.

DEVICE♯NNNN……NNNNはDevice Noである.

FILE—NAME ××××××．EX……××××××はファイル名，EXは拡張名である.

DATE MM/DD/Y……MMは月，DDは日，Yは年である.

FILE LENGHT llll……llllはファイル長である.

6. ローダ（LOAD）

バイナリ形式のファイル，または紙テープを入力してコアイメージを作るためのプログラムで，CCBも同時に作成する．CCBについて必要なデータはオプションとしてコマンド行に入力される．

コマンド行をCRで終端すれば，指定されたファイルを入力後再び*に戻り，ESCで終端すればモニタに戻る．モニタはコアイメージをSAVEするかプログラムをSTARTするか，PODで制御するかなどが可能である．オプションGを指定すれば終端オプションにかかわらずロードしたプログラムを実行する．

すでにイメージ形で保存されているプログラムにパッチ（部分的な修正用のバイナリプロ）を加えるための手段として，LOADはオプションIによって，イメージ形のファイルを1個だけ入力することを許している．この場合には他の入力に先立ってイメージ形のファイルのロードを実行しなくてはならない．

バイナリ形の入力ファイルは1つのコマンド行に対して，最高9個まで許される．コマンド行は何行にわたっても良い．

LOADは一つの入力ファイルについてひとつのバイナリファイルのみを許す．途中にリーダ，トレーラを含む紙テープは複数個のファイルとして扱い1個ごとにリーダの読取り開始位置を確める必要がある．

ファイル形名は無指定ならば．BNを付加する．

6－1－1　ローディング・フィールドの指定

/n（ただしn＝1～7）で行う．無指定ならば0フィールドとなる．また，入力ファイル中にフィールド指定があれば，以後のすべての入力ファイルはこの新しい指定に従う．

6－1－2　実行開始番地（SA）の指定

＝nnnnnで行う．フィールドを含め5桁で指定する．無指定ならば00200となる．

6－1－3　JSWの指定

プログラムがフィールド0，およびフィールド1の0～1777番地を使用するか否かは，USR，PODその他のシステムが，この範囲を使用する際にシステムスクラッチに保存する必要の有無を決定する．このための情報はJSWに蓄えられるLOADに対してはフィールド0および1の各々に対して/8および/9で指定する．

6－1－4　例

```
. R  LOAD                  PROGAをバイナリ紙テープから
*PTR:$                     SYS:PROGA.SVとして保存，実行開始番地
. SAVE  SYS  PROGA         は0フィールドの200
. R  LOAD
*PTR:=13000(89)$           PIPの実行開始番地は13000
. SAVE  SYS  PIP
. R  LOAD
*SYS:PIP.SV/I              1オプションは第1行のみ
*PTR:=13000(89)$           1オプションでは古いSVファイルのSAおよ
. SAVE  SYS  PIP           びJSWは捨てられるので新しく指定が必要
. R  LOAD                  PROGAをロードして200番地から
*SYS:PROGA.BN(89)/G        実行開始
```

7. エディタ（EDIT）

エディタはSAPD，RAPD，FORTRANなどの入力となるASCIIソースプログラムを作ったり，変更したりするためのプログラムである．

従来の紙テープエディタの機能に加えて，文字ストリングサーチ，ページスキップの機能が新しく追加された．

7－1　入出力指定

次の書式による

*odev：file. ext＜idev：file. ext / B

出力デバイスおよび出力ファイルを指定しない場合は，入力デバイス，入力ファイルと同じに指定される．出力ファイル名を省略した時には，指定の出力デバイスに，入力ファイルと同名の出力ファイルを指定される．

Bオプションの指定は，入力テクストの2個以上連続するスペースをタブに変換する．Bオプションを指定しなければ何らコード変換は行わない．

出力デバイスのみ指定することもあり得る．この場合，入力テクストはAコマンドにより，TTYキーボードから与えられる．

出力ファイルの永久登録はEコマンドまたはQコマンドによってモニタに戻る直前に行われ，その際に，入力ファイルと出力ファイルが同名の場合および出力ファイルと同名の他のファイルが存在する場合には，これらの抹消が実行される．従ってエディタの実行の途中でモニタに戻した場合には出力ファイルは抹消され，入力ファイルは何の影響も受けない．この時エディタを.STARTで再スタートできる．もし，モニタに戻る時にエディタのバッファにテクストが残っていれば，これは再スタートしたエディタで再びアクセスできる．

入力デバイスとしてTTY：（あるいは高速紙テープリーダのないシステムでのPTR：）を指定することは避けなければならない．これは，あるブロックの入力が終った後，エディタのコマンドモードに戻り，入力テープの一部をコマンドして読みとることが避けられないためである．このような場合には，CONVRTで一旦システムデバイス上のファイルとして登録し，これをエディタの入力ファイルとして指定するようにすればよい．

7－2　エディタの動作

エディタはページと名づける論理ブロックごとにデバイスとのデータ転送を行うものと想定する．実際のデバイスとのデータ転送は入力バッファまたは出力バッファが空きまたは満杯になるごとに行われるので，エディタの論理的なデータ転送とは直接の関係はない．またエディタのページは，エディタのコマンドでアクセスできる論理ブロックであって，計算機の1ページ（128語）とは何の関係もない．エディタの1ページの最大の大きさは，エディタの用意するバッファの大きさで制限され約5000字である．このバッファの大きさは，SAPDのソースプログラムの約1ページ（128語分）を収容して，エディティングのための余地を残すに充分である．もちろん，コメントが非常に詳細であれば，この限りでない．

エディタの1ページは各ページの最後のＦＦコードによって区切られる．出力コマンドＰによって出力される時には，ＦＦコードはその出力テクストの最後に自動的に付加される．

エディタの出力ファイルの最後にはEOFコード（↑Ｚ）が自動的に付加される．

エディタは*に対するコマンド行が終端されると直ちにコマンドモードにはいる．エディタのコマンドモードは＃の印刷によって示される．ユーザは＃に対し，必要なエディタコマンドをキーインすることによって，エディタに動作の指示を行う．エディタコマンドは一般に＃ｎ，ｍ×↙の形をとり，×↙の2字分の組み合わせで受付けが行われる．

7－2－1　入力コマンド（R，A，I）

Rコマンドは入力デバイスのファイルから1ページだけエディタのバッファに転送することを指示する．もし，エディタのバッファにテクストが存在すれば新しく転送されたテクストはその後に付加される．入力に際し，000，200，212，377の各文字コードは無視され，転送されない．

Aコマンドはキーボードからテクスト入力を受付けて，これをエディタのテクストバッファの最後に順次付加する．コマンドモードに戻るにはＦＦ（↑Ｌ）による．キーボードからの入力はエコーバックされてプリンタに印字される．TAB（↑Ｉ）はスペース8個に換算してエコーバックされる．ＣＲはＣＲ－ＬＦとエコーバックされＬＦは無視される．

Ｉコマンドはエディタのテクストの第1行目の前にキーボードからのテクストを挿入する点が異る．コマンドモードへの戻りとエコーバックについてはAコマンドの場合と同じである．nIとすれば第n行目の前に何行でも挿入される．

7－2－2　出力コマンド（P）

Pコマンドは，エディタのテクストバッファの内容の最後にＦＦコードを付して出力デバイスに転送する．ｎ，mpとすれば第n行から第m行までにFFコードをつけて出力デバイスに転送する．nPでは第n行にＦＦコードをつけて出力デバイスに転送する．テクストバッファが空の時のPコマンドはエラーとなる．

7－2－3　抹消コマンド（D）

Dコマンドはエディタのテクストの任意の行を抹消するのに用いる．nDは第n行を抹消する．ｎ，mDは第n行から第m行までを抹消する．

7－2－4　リセットコマンド（K）

エディタのバッファをすべて抹消して，リセットする．通常は後述のNコマンド，Yコマンドに複合されている．

7－2－5　印刷コマンド（L，V）

Lコマンドはテクストバッファの内容をTTYに印刷する．TABコードはスペース8個に換算して出力される．nLは第n行のみ印刷．n,mLは第n行から第m行まで印刷する．

Vコマンドは出力がLPT：である他は，Lコマンドに同じ．

7－2－7　終結コマンド（↑C，Q，E）

↑Cは直ちにモニタに戻る．出力中であったファイルは抹消される．モニタからSTARTで再スタートを行えば，再び*に戻り，入出力ファイルを新しく指定できる．↑Cをキーインした時点のテクストバッファの内容は，そのまま残っている．

Qコマンドは現在のテクストバッファの内容を出力して，出力ファイルを正式に登録した後にモニタに戻る．もし，入出力名が同一の場合には，出力ファイルを正式登録する時点で入力ファイルは抹消されるので，まだ読んでいない残りの部分はすべて捨てられる．モニタから．STARTで再スタートを行った場合，テクストバッファの内容は残っている．

Eコマンドは，入力ファイルの内容がなくなるまでNコマンドを繰返し，その後Qコマンドを実行する．

7－2－8　サーチコマンド1（G）

Gコマンドはアセンブラのソースプログラムでタグ（シンボリックアドレス）のついた行を探すのに用いる．書式はnG↵またはG↵である．

nGは第n行以降で最初にタグのついた行を印刷する．タグのついた行とは，第1字がスペース，タブ，およびスラッシュでない行のことである．Gのみでは．＋1Gを意味する．すなわち現在行の次の行からサーチを開始する．

7－2－9　サーチコマンド2（S）

Sは指定した文字までコピーをするコマンドで，行の途中を一部分抹消したり修正したりするのに用いる．

nS　とした後でサーチ文字をキーインして指定を行う（サーチ文字はエコーされない）と第n行をサーチ文字まで印刷して止まる．この時のエディタをサーチモードとなずける．

もしサーチ文字が発見されなければ，行の最後まで印刷してコマンドモード（＃）に戻る．

サーチ文字が止った時にユーザは次の各項いずれかを実行できる．

1. 新しい文字を続けてキーインする．キーインした文字はすべて新しいテクストの一部となる．
2. ↑Lでサーチを続行する．再びサーチ文字が出てくれば止る．
3. ↑Gの次に新しいサーチ文字を入力する（サーチ文字はエコーバックされない），サーチ文字が出てくれば止まる．
4. ↵はその行の以降のテクストを捨てて，新しい行に移る．
5. ＬＦはそれまででその行を閉じ，以降のテクストは新しい行とする．
6. ↑Uはその行のすでに印刷された部分を抹消する．
7. RUBOUTは①とエコーバックされその行のすでに印刷された部分を後から1字ずつ抹消する．

n，mSは第n行から第m行にわたって上記のサーチを実行する．Sはバッファのすべての行についてサーチを実行する．

7—2—10　サーチコマンド3　ESC．′，″

ESCは文字ストリングのサーチを実行する．ESCは＄とエコーバックされ，任意の文字ストリング（ただし，′および″を含むことは許されない．またストリングの長さは20文字までが有効でそれ以上はタイプライタに印刷されるが無視される）指定を受けいれる．文字ストリングの最後は′または″で終端される．エディタは′では第1行から，″では現在の次から，指定された文字ストリングを探し，＄から′または″の代りに，発見した行の行数を返す．この時，現在行の値は発見した行の値に等しくなる．＃＄ABC′，＄XYZ″Lとすればabcのはいった行からXYZのはいった行まで印刷する．同じ文字ストリングは再指定しなくても単に″だけでも良い．

＃＄ABC′，″LはABCのはいった行から次にABCのはいった行までを印刷する．サーチが不成功の場合は？で＃に戻る．

7—2—11　サーチコマンド4（J，F）

Jコマンドは前節のサーチに類似であるが，発見できない場合は自動的にNコマンドを実行し，発見できるまでテクストの転送を行う点が異る．発見した時にはそのまま，コマンドモードに戻る．EOFまで探して発見できなければ？で＃に戻る．この場合は出力ファイルはEまたはQで閉じる必要がある．J↵とするとエディタは＄を印刷して文字ストリングのキーインを待つ．文字ストリングを′で終端するとサーチが開始される．同じ文字ストリングについてのサーチ続行はF↵で指定する．

7—2—12　コピーコマンド（N）

Nコマンドは，Pコマンド，Kコマンド，Rコマンドの複合コマンドで，P，K，Rの順に行う．

nNはP，K，Rのn回繰返し．

7—2—13　スキップコマンド（Y）

Yコマンドは，Kコマンド，Rコマンドの複合コマンドで，K，Rの順に行う．

nYはK，Rのn回繰返し．

7—2—14　フィードコマンド（T）

32個のブランクフィードを出力する（紙テープ出力）

7—2—12　エディタの特殊文字

次の幾つかの文字はエディタのモードにより特殊の意味を持つ.

	テクストモード	コマンドモード	サーチモード
LF	無視	+1Lに同じ	現在行を閉じる
CR	CR-LF		現在行を閉じる その行の以下の部分は抹消
↑U	その行のすでに入力した部分を抹消		テクストモードに同じ
RUBOUT	Rコマンド中では無視 キーイン中には最後から1字ずつ抹消		最後から1字ずつ抹消
↑G	コマンドモードに戻る		サーチ文字変更
↑L	コマンドに戻る		サーチ続行
↑C	モニタモードに戻る	モニタに戻る	モニタに戻る
↑O	コマンドモードに戻る		コマンドモードに戻る
.		現在行の値	
>		.+1Lに同じ	
<		.−1Lに同じ	
／		バッファの最後行の値	
:		:の左側の値をデシマル印刷 (たとえば #/:035)	

7—2—13　出力デバイスオーバーフローの処理

出力ファイルとして割当てられたスペースに出力がはいりきらない場合には，そのファイルを閉じた後にFULLのエラメッセージを印刷してコマンドデコーダ*に戻る．新しい出力ファイルを指定してやれば，テクストの損失を招くことなくエディティングを続行できる．このようにして分散したファイルは，CONVRTのオプションAによって1つのファイルにまとめればよい．分散したままのファイルは，行の中間で2つのファイルに分れている可能性があり，これをエディタの入力とすると一部のテクストが失われる．CONVRTのオプションAで一つにまとめたファイルについては問題は起きない．

8.　アセンブラ（SAPD）

SAPDはSAPファミリのアセンブラの一つで，アセンブラ言語によって書かれたソースプログラムを絶対バイナリ形のオブジェクトプログラムに変換する．他のSAPアァミリのアセンブラと同様

に２パス形式のアセンブラであるが，入力としてファイル構造デバイス上のASCＩＩファイルを使用する点が異る．２つのパスは自動的に連続して行われるので，見掛上は１パスとなっている．

マークⅡモニタシステムに特有の入力指定と操作を除けば，他のSAPファミリアセンブラと仕様はほとんど共通で，コンパティブルである．

SAPDでは最高５個までの別々のプログラムを連続してアセンブルすることが可能である．一方，別々のファイルにわかれている一つのプログラムを連続してアセンブルすることは許さない．CONVRTプログラムで一つのファイルにまとめておいてから，アセンブラにかけるようにする．

８－１　SAPDの入出力指定

.R SAPDで呼び出されるとすぐに＊を印刷して，コマンド行を要求する．コマンド行は１個から５個の入力ファイルとオプションから成る．個々の入力ファイルはすべてASCⅡ形のソースファイルで各々が独立したプログラムであることが必要である．

アセンブラの出力はコマンド行がＣＲで終端されるか，ESCで終端されるかによって，自動的に選択され，必要ならば適当な出力ファイル名が割当てられる．

入力ファイルについては形名は指定がなければ.TM，出力ファイルの形名はバイナリオブジェクトおよびリスティングについて.BNおよび.LSとする．

８－１－１　コマンド行のオプション

SAPDに対するオプションは次の通りである．

無指定	バイナリもリスティングも出力
/B	バイナリのみ出力
/L	リスティングのみ出力
/N	バイナリもリスティングも不要
/T	リスティングはTTY形式 (ページフィードを入れる)

これらのオプション指定は，入力ファイルのすべてに共通で，個々のファイルごとに別々に指定することはできない．

８－１－２　CRで終端したコマンド行

コマンド行をＣＲで終端した場合には，バイナリ出力は紙テープ（PTP：）に，リスティングは行印刷機（LPT：）に出力される．

８－１－３　ESCで終端したコマンド行

コマンド行をESCで終端（＄とエコーバックされる）した場合には，バイナリはシステムデバイス（SYS：）に，リスティングはデフォールトデバイス（DSK：）に出力される．ファイル名は入力ファイル名と同じで，形名は各々.BNおよび.LSとなる．同名のファイルがすでに存在する場合には，それらは新しいファイルによって自動的に抹消される．

８－１－４　SAPDの内蔵シンボル表

SAPDのシンボル表はSAP—Ⅲに次のUSR関係の命令を加えたものである．

QFETCH, RSERCH, QOPEN, QCLOSE, QCD, QCHAIN, QERROR,QLOCK, QFIRE, QUIZ, QRESET.

次の擬似命令が追加された．QBLOCK ｎ（ｎはオクタル数，またはシンボル，あるいは表現）ｎの価の個数の０を確保する．

バイナリ出力には，ｎ個の０が付加される．リスティング上では，プログラムカウンタがｎだけ増される．

ATEXT．　TEXT擬似命令と同様であるが，８ビットのASCII値を，そのまま，１文字／１語として，アセンブルする．終りに０を追加することはない．

8－1－5　ライブラリエディタ（LIBEDT）

ライブラリエディタはSAPアセンブラ用サブルーチン集（LIBSAP），あるいは，その他のプログラム集から必要なパッケージを抜粋するためのユーティリティである．

ライブラリエディタは入力ファイルをエディタのページごと（ＦＦコードで区切られるサブルーチン名）にコアにロードして，指定されたシンボル定義の有無をチェックする．あればこのページを出力ファイルに送り，なければこれを無視する．出力ファイルは，指定されたサブルーチンだけから成り，必要とするソースプログラムと容易に連結できる．

コマンド行の指定は次の書式による．

*odev：file＜idev：file

出力デバイス無指定の場合は，PTP：が選ばれる．入力デバイス，ファイル名無指定の場合はSYS：LIBSAPが選ばれる．

LIBEDTでエディトする入力ファイルの各ページは3.7k字以内でなければならない．入力ファイルのページが大きすぎる場合は，エラーメッセージを印刷してモニタに戻る．出力ファイルは抹消される．

例

```
.R  LIBEDT
*S                              入力ファイルはSYS：LIBSAP
NAME：STRING , PRINT,OCTPR      出力はPTP
PRINT                           PRINT.OCTPR.STRINGの順に出力
OCTPR                           された．
STRING
.R  LIBEDT
*TEST＜IOD1 ：LIBRAS            入力ファイルは IOD1 ：LIBRA.TM
NAME：HEXPR,KEYIN               出力ファイルはDSK：TEST.TM
HEXPR
KEYIN
```

9.　フォートラン（マークⅡ）

マークⅡフォートランシステムはコンパイラ（FORT），リロケータブルアセンブラ（RAPD），リンキングローダ（LLDR），ライブラリ（LIB555）から成り立つ．

コンパイラはアセンブラ，およびローダに自動チェインの機能を持ち，中間出力は自動的にシステムデバイス上の一時ファイルとして登録され，使用後抹消される．コンパイラに対し，必要なオプションを指定して置けば，コンパイルから実行までは全部自動的に実行される．また必要により，リンキングローダによって作られたコアイメージをそのままコアイメージファイルとしてセーブすることもできる．

9－1　マークⅡフォートラン言語の文法（抄）

9－1－1　定数

マークⅡフォートランでは整数形定数，実数形定数およびホラリス形定数の３種類の定数を許す．

整数形定数は１～４字から成り，符号があってもよい．整数形の定数は±2047以下の数値でなければならない．

実数形定数は$\pm 1.7\times 10^{38}$以下の実数（小数点付き）有効数字８桁まで変換され，それ以降は無視される．表現はＦ形（通常小数点付10進数）でもＥ形（Ｆ形にさらにE±NNの形で10の指数を

付けたもの）のどちらでもよい.

ホラリス形の定数は′で囲んだ6字以下のASCⅡ文字（ただし′は除く）のストリングで実数形の定数として扱われる.

9－1－2　変数（配列を含む）

変数名は英字で始まる5字以下のストリング（6字以降はあってもよいが無視される）とし，最初の英字がI，J，K，L，M，Nのいずれかであれば整数形，それ以外は実数形とする．添字は1次元および2次元が許される．添字付変数はDIMENSIONであらかじめ定義しておく.

9－1－3　ライブラリ関数

マークⅡフォートランは以下のライブラリ関数を持つ，関数名がIで始まる関数の値は整数，その他の関係の値は実数である．変数I，Jは整数形，変数X，Yは実数形とする.

ABS(X)	絶対値（実数）
IABS(I)	〃　（整数）
FLOAT(I)	実数に変換
IFIX(X)	整数に変換
IREM(　)	直前の整数割算の余り
IREM(I/J)	I/Jの余り
EXP(X)	e^X
ALOG(X)	log e X
SIN(X)	sin　X　Xはラジアン
COS(X)	cos　X　〃
TAN(X)	tan　X　〃
ATAN(X)	$Tan^{-1}X$　結果はラジアン
SQRT(X)	$\sqrt{X}$
JRDSW(　)	スイッチレジスタの値

9－1－4　算術式

算術式は定数または変数を演算子で連結したもので（および）で演算順序を指示することができる.

演算子の順位は

＊＊	べき乗
－	負号（他の演算子に続いて現れる）
＊および／	乗除
＋および－	加減
＝	置換

で上にある演算が先に実行される．同位の演算子は左から右へと実行される．一つの表現の中ではI＊＊XおよびX＊＊Iを除いて，実数形の変数，定数，表現が混在することは許されない.

実数演算ではオーバーフローはチェックされ，エラーメッセージが印刷される．整数演算ではオーバーフローはチェックされず，そのままの値で計算が続行される.

(2047＋2047→－2)

X＊＊IおよびJ＊＊Iは乗算の繰返しとして実行される．X＊＊YおよびJ＊＊YはEXP関数およびALOGを使って計算される．XまたはJが負数の場合もエラーとはならず，絶対値をとって計算する.

((－3.0)　＊＊3.0→27.0)

実数演算の精度は符号を除いて27ビットである．従って，10進7桁の精度が保証される．ライブ

ラリ関数の精度は10進6桁である.

9－1－5　制御文

GOTO文はGOTO nとGOTO (n1, n2…), Jの2つの形が許される.

IF文は, IF (expression) n1, n2, n3の形の算術IF文だけが許される.

DO文はDO n I=m1, m2, m3は整数形定数または整数形変数のいずれかに限る. m3は省略すれば1とみなされる.

PAUSE, PAUSE n, はプログラムの実行を一時停止させる. 続行は, コンソールのCONTを押すことによる. PAUSE文がPTRへのデータテープのかけかえのために用いられることがある. この場合にはPAUSE文の直後にCALL OPENを置いて, PTRのイニシャライズを行う必要がある.

STOP文は, コントロールをモニタに戻す, CALL EXIT文に同じである.

END文はソースプログラムの一番最後に置かれ, コンパイルの終りを知らせる. または, END文は自動的にSTOP文を挿入する.

9－1－6　入出力文

入出力文は

READ (unit, format) list

WRITE (unit, format) list

またはWRITE (unit, format)

の形をとる.

unitはデバイス番号で

1. TTY
2. PTR/PTP
3. LPT/CDR
4. I/Oドライバによる万能入出力

を意味する. TTY入力ではエコーバックが印字される. LPT出力では各行の第1字は改行制御文字で印字されない. スペースは通常のシングル改行, ＋は改行なし (重ね打ち), 0はダブル改行, 1はフォームフィード (次頁にスキップ), その他の文字はシングル改行を意味する.

formatはFORMATの番号である.

listは値を転送すべき変数名または表現のリストであって, その各項は次のいずれかの形をとる.

1. 算術式. ただし入力については変数または添字付変数のいずれかに限る.
2. DOループ付算術式.

たとえば　(S1, S2, …, Sn, i＝m1, m2, m3) の形をとる.

たとえば　DIMENSION A (3, 6)

WRITE (1, 100) I (A (J, I), J＝1, 3)

では

I, A (1, I), A (2, I), A (3, I) が出力される.

3. 配列名, 指定された配列のすべてを意味する.

DIMENSION A (3, 6)

READ (1, 100) Aでは

A (1, 1), A (2, 1), A (3, 1), A (1, 2) …A (3, 6)

のすべてが入力される.

listのない場合にはFORMAT文による.

万能入出力については別に述べる.

9－1－7　FORMAT文

FORMAT（S1，S2，……Sn）の形とする．

Siはデータの形式指定でE，F，I，A，X，Hの各変換および′で囲まれた英数字ストリングが許される．

1. E変換

Ew.dの形とする．wは与えられたフィールドの大きさ（全字数），dは小数点の右の字数である．　w≧d＋6でなければならない．

E16．8の例を示す．

␣␣␣・1 2 3 4 5 6 7 8 E＋02

␣␣－・8 7 6 5 4 3 2 1 E－10

入力についてはE変換は次のF変換と等しい．

2. F変換

Fw．dの形とする．F 8．5の例を示す．

␣1．2 3 4 5 6

－2．3 4 5 6 7

wの値が充分に大きくない時には左側に溢れた数字は印字されない．

入力時に，小数点が入力されれば，dの値にかかわらず優先する．以下の例はF 6．2またはE 6．2でいずれでも100.00と解釈される．

100.␣␣

100E 2␣

1 .E␣␣

␣10000

100 .↙

100E 2↙

1．E 2↙

10000↙

3. I変換

Iwの形をとり，整数形の変数の入出力に用いられる．

4. A変換

Amの形をとり変数が整数形であれば，wは1または2，実数形であればwは1～6である．文字の値はASCIIの6 LSBがとられ2字/語にパックされる．値は左寄せで，空きの部分にはスペース（40）が詰められる．

5. H変換およびホラリス定数表現

H変換はASCII文字をそのまま印字するのに用いられnH×××××…×の形とする．nはスペースも含めた字数である．H変換の代りに両端を′で囲み′××××′とすれば字数を数える必要がない．H変換は本来WRITE文に関連するが，マークⅡフォートランでは，READ文に用いてキャプションを印字するのを許している．たとえば，

READ（1，100）A，B

100　FORMET（′A＝′，F 7．2/′B＝′，F 7．2）

ではAおよびBの値の入力の前にA＝およびB＝が印字される．

6. X変換

nXの形をとり，出力ではn個のスペース印字，入力ではn文字読み飛ばしを意味する．

9－1－8　FUNCTIONサブプログラム

FUNCTIONサブプログラムはMAINプログラムと別にコンパイル，アセンブルされる．

FUNCTIONサブプログラムは1個の結果を与える関数でライブラリ関数と同様に、FUNCTION名で式の中に用いられる.

FUNCTIONサブプログラムはFUNCTION文で始まり、途中に1個以上のFUNCTION文を含み、END文で終る.

FUNCTION名は変数名と同じ規則に従う.

9－1－9　SUBROUTINEサブプログラム

SUBROUTINEサブプログラムはMAINプログラムと別にコンパイル、アセンブルされる.

SUBROUTINEサブプログラムはCALL文でもってMAINプログラムから参照される.

SUBROUTINEサブプログラムはSUBROUTINE文で始まり、途中に1個以上のRETURN文を含み、END文で終る.

9－1－10　CALL EXIT文

オブジェクトプログラムの実行終了後モニタに戻すにはCALL EXITを用いる.

9－1－11　CALL CHAIN

CALL CHAINはコアに収容しきれない大きなプログラムを幾つかのセグメントに分け、これらを順にコアにロードして実行するために用いる. CALL CHAINによって、実行中のプログラムの代りに新しいプログラムがコアにロードされ、コントロールが移される. この時COMMON領域に確保された変数は新しくロードされたプログラムで使用できる.

CALL CHAIN（'PROG'）の形で用いる. 呼ばれるプログラムはシステムデバイスにSYS：PROG・SVの形で登録されていなければならない.

9－1－12　万能入出力

マークⅡフォートランではI/Oドライバを使用する万能入出力としてユニット番号4のデバイスの使用を許している.

この万能入出力はファイル構造のデバイスでも、その他でもよく、しかも、実行段階で指定することも可能である.

1. 入力デバイスの割当（IOPEN）

   入力デバイスの割当てはCALL IOPEN（A、B）の形で行い、以降READ（4、X）の入力はデバイスAのファイルBから入力される. AおよびBはホラリス定数の形で与える. またファイルBの形名は、DAとする.

   CALL IOPEN（'DSK'、DATA'）

   とすれば、入力は　DSK：DATA.DAである.

   CALL IOPEN（'PTR'、0）

   では入力はPTR：が割当てられる.

   デバイス名およびファイル名を実行段階でREAD文で読むとすれば、各A6FOR MATとし、入力時には6字になるまで@を補う必要がある.

2. 出力デバイスの割当て（OOPEN）

   出力デバイスの割当てはCALL OOPEN（A、B）の形でIOPENに準じる. ファイルの形名は自動的に.DAとなる.

3. 出力デバイスの切り離し（OCLOSE）

   OOPENで作ったファイルはOCLOSEでダイレクトリに登録される. OCLOSEを行わない出力ファイルはモニタに戻った時点で削除される. OCLOSEを行わないで重ねてOOPENを行った場合はエラーとなる. OCLOSEは引数を持たない.

   すなわち

   CALL OCLOSE

4. フォートランのデータファイル

WRITE（4，X）で作ったデータファイルは，後にREAD（4，X）の形で再び入力するためのみであれば，実数形の変数はA6，整数形の変数はA2で転送すると，データ量も小さく，2進10進変換も行われないので転送時間，演算時間共に大幅に節約される．

9—1—13 COMMON宣言文

COMMON文は次の書式による

COMMON V_1，V_2，……V_N

V_iは変数名または配列名である．COMMON文はフォートランプログラムの一番始め（COMMENT文を除く）に位置しなければならない．

9—1—14 DIMENSION宣言文

配列の大きさを規定する．書式は次の通りである．

DIMENSION a_1，a_2，……a_N

a_iは配列名とその大きさである．

9—1—15 EQUIVALENCE宣言文

EQUIVALENCE文は2つ以上の変数が同じ場所を共有することを指示し，次の書式による．

EQUIVALENCE（u_1，u_2…），（　　）……

u_nは変数名または配列の要素である．配列の添字は整定数に限る．COMMON文にあらわれた変数名または配列名はEQUIVALENCE文にあらわれてはならない．

また，同じ変数が2度以上EQUIVALENCE文にあらわれてはならない．

EQUIVALENCE（A，B），（B，C）

は

EQUIVALENCE（A，B，C）

とする．

9—1—16 アセンブラコード

MARK Ⅱ フォートランでは，ソースプログラムの中にアセンブラコーディングを自由に混ぜて用いることができる．これによって，フォートランシステムに含まれない入出力制御や論理演算などを容易に使用することが可能となる．フォートランのソース文で第1コラムがSであると，この行はRAPDのソース行であることが，コンパイラに示され，コンパイラはこの行を何等処理をせずに，コンパイラ出力（RAPDソース）に転送する．

たとえば

```
                DIMENSION  K(10)
                DO    5    I=1, 10
   SWAIT,       6131
   S            JMP  WAIT
   S            6136
   S            DCA  \J
                K( I )=J
    5           CONTINUE
```

このコーディングは，デバイス13から10個のデータを入力して配列Kに格納する．

この操作はサブルーチンサブプログラムの形で使用するよりもはるかに容易で，能率もよい．

フォートランソース中の変数名および文番号はアセンブラのソースコーディング中では各変数名および文番号の前に＼(バックスラッシュ，SHIFT/L)を付けたラベルで扱われることに注意する．

9—2 RAPD アセンブラの文法

RAPDアセンブラはフォートランコンパイラの出力を自動的にリロケータブルオブジェクトに変換するのに用いられるが，それ自体としてプログラマがコーディングしたRAPDソースをアセンブルしたり，あるいは，フォートランコンパイラの出力ファイルをエディタで修正したファイルをアセンブルしたりするのに用いられる．

RAPDアセンブラの命令語はSAPDとほとんど共通であるが，

1. ページおよびフィールドの処理はアセンブラおよびローダが自動的に行う．
2. オブジェクトプログラムは，リロケータブルバイナリ形式でリンキングローダ（LLDR）によってコアにロードすることができる．
3. 実行に必要なライブラリルーチンは，リンキングローダによって，自動的に必要なものだけがコアにロードされる．
4. アセンブルの形式は1パス方式であるが，リスティングを必要とする時は，第2パスで行う．
5. 次の擬似命令が用意されている．

ABSYM ARG BLOCK CALL COMMN CPAGE DECIM DUMMY
EAP END ENTRY FORTR I IF LAP OCTAL OPDEF PAGE
PAUSE REORG RETRN SKPDF TEXT

9—2—1 使用する文字

アルファベット26文字＋数字10文字および次の特殊文字

,	シンボル番地を定義する．またCALLの引数の数を定義する．
/	以下はコメント（RAPDは無視，ただしリスティングには印刷される．
(	リテラル定数，数の変換はその時のOCTALまたはDECIM定義の状態に従う．(Dnnでは上記にかかわらずデシマル定数，(Knnでは上記にかかわらずオクタル定数，(nn，(Dnn，(Knn，などでnnは数に限る．
〃	ASCII定数，次の1文字のASCII値を作る．
—	定数，ASCII定数について－1をかける．
#	直前のシンボル番地＋1をとる．××#はSAPの××＋1に相当する．
CR	1行の終り
	1命令文の終り
スペース，タブ	シンボルのセパレータ
LF，FF	リスティング以外には無視
RUBOUT	無視

上記以外の文字はコメントの中（/からCRまで）ASCII定数（〃の直後），TEXTの中（TEXTの次の次の文字から，同じ文字まで）以外では不正文字となる．

9—2—2 命令文

命令文シンボル番地，コンマ，命令語，オペランド，コメントから成る．

シンボル番地の定義と用法はSAPDに準じる．命令語は

1. 番地参照命令，RAPDに内蔵されている命令およびOPDEFまたはSKPDEFによりプログラムで定義したもの，(間接参照のIも含むと考える)
2. OPR（7×××）およびIOT（6×××）命令および，その複合命令
3. 擬似命令

から成る．

オペランドは，シンボル番地，リテラル定数または数である．SAPDと異り，オペランド部分での演算は許されない．

9—2—3 数

数は1～4字の字数ストリングで＋または－の符号があってもよい．数の変換は指定がなければオクタル変換とする．DECIM擬似命令，があるとその次のOCTAL擬似命令，またはEND擬似命令まで10進変換となる．

9－2－4　各種の疑似命令の概略

ABSYM	絶対番地定義
例	ABSYM TEM 177 シンボルTEMには絶対番地177が割当てられる．
ARG	サブルーチンコール（CALL）に続く引数指定，ARGに続くオペランドはフィールド指定を含めて2語にアセンブルされる．
BLOCK	オペランドで指定された数だけ0をアセンブルする．
CALL	外部サブルーチンコール 例　CALL　3，SUB 外部サブルーチン名SUB，引数は3個
COMMN	ＣＯＭＭＮ領域（フィールド1の200番地以降）への割当て，BLOKに類する．これはすべての外部プログラムに共通に定義される． 例　A，COMMN　20
CPAGE	以下の語数を同一ページに確保する．確保できなければ，新しいページを割当てる（PAGE擬似命令を実行）． 例　CPAGE　20 これに続く20語が連続した番地にアセンブルされることを保証する．
DECIM	SAPDのDECIMALに同じ．
DUMMY	シンボル番地の内容はフィールドを含む2語に表わされるとして扱う． 例　DUMMY　A A，　　BLOCK　2
EAP	1つのページから次のページへのジャンプを自動的にアセンブルする（RAPDの通常のモード）．
END	SAPDの＄に相当
ENTRY	サブルーチンの入口シンボルであることを指示する．ENTRYで指定されたシンボルはLLDRで外部シンボルとして処理される． 例　ENTRY　SUB SUB，　　BLOCK　2
FORTR	フォートランコンパイラの出力にあらわれる．
I	間接指定
IF	条件付アセンブル指定 例　IF NAME　7 NAMEがすでに定義されていれば，何もしない．未定義ならば，これに続く7命令は無視する（アセンブルしない）．
LAP	EAPの逆指定，ページからページへの自動スキップを行わない．
OCTAL	SAPDのOCTALに同じ．
OPDEF	通常の命令定義（スキップが起らない命令） 例　OPDEF TADI 1400
PAGE	次のPAGEへスキップ（SAPDのPAGEに類似）
REORG	SAPDの＊に類似，ただしREORGに続く数はページの先頭（オクタル200の倍

数）でなければならない.

RETRN　サブプログラムからの戻り指定

例　ENTRY　SUB

SUB,　BLOCK　2　BLOCK

⋮

RETRN　SUB

SKPDF　スキップが起るかもしれない命令の定義

例　SKPDF　DFSC　6 6 2 2

TEXT　SAPDのTEXTに同じ.

ただし、文字数が、偶数個の場合には、最後に0データを付加しない.

ABS　ABSYMで定義されたもの

COM　COMMONで定義されたもの

EXT　CALLで引用されたもの

ENTRYで定義されたもの

OP　OPDEFまたはSKPDEFで定義されたもの

UNDF　未定義（EXTを除く）、プログラムエラーである.

9－2－5　シンボル表

シンボル表はパス1の最後にABC順に印刷される. 絶対番地（ABSYM）および命令（OPDEFおよびSKPDEF）以外はプログラムの先頭番地を200とて計算される. 通常のシンボル番地以外のものについては以下の注が添えられる.

ABS　ABSYMで定義されたもの

COM　COMMONで定義されたもの

EXT　CALLで引用されたもの

ENTRYで定義されたもの

OP　OPDEFまたはSKPDEFで定義されたもの

UNDF　未定義(EXTを除く)、プログラムエラーである.

9－3　MARK II. フォートランの操作

9－3－1　ソースプログラム

フォートランのソースプログラムは通常EDITを利用して作成される. コンパイラの入力として紙テープ（PTR：）、IBMカード（CDR：）その他システム構成に含まれる非ファイル形の入力機器、およびシステム構成に含まれるファイル形の機器が使用可能である. ファイルの場合はファイル名の形名は.FTであれば、入力指定に省略できる.

EDITを使ってソースプログラムを作成する場合には、通常のフォートラン文法に従ってコラムを数える代りにTABでアドレス部と命令文本体とを区切ることが許される.

1. 各行の第1字がCであればコメント行である(コンパイラは無視).
2. 各行の第1字がSであれば、RAPDソースである（コンパイラはそのまま出力に転送する).
3. 各行の第1字が数字であれば、文番号である（文番号は1～4桁の10進数で2047以下). 文番号と命令文の間はTABをいれる.
4. 各行の第1字がTABであれば文番号がない命令文である.
5. 継続行の場合には第1字がTAB、第2字は1～9の数字とする. 継続行の本文は第3字から始まる.
6. 通常のフォートラン文法に従って、コラムワイズにソースを作成してもよい. この場合継続行は第6字に0以外の英数字で指定できる. 文番号は第2字から第5字に与える.

7. 命令文の中では，FORMAT文の中，およびホラリスフィールド（H指定または' 指定）の中以外では，すべて無視される.

9—3—2 コンパイラの実行

コンパイラはFORTでロードされる.

例

```
.R FORT      FORTがSYS：にセーブされている場合
*
.RUN IOD3FORT FORTがSYS：以外にセーブされている場合
*
```

*に対しては0～3個の出力ファイル名と1個の入力ファイル名，および必要数のオプションを指定する.

1．出力ファイル指定

出力ファイルの1番目はRAPDの出力オブジェクト用で形名無指定の場合は.RLが付加される. ファイル無指定で/Lまたは/Gが指定された場合は，SYS:FORTRL.TMを作り，実行後FORTRL.TMを抹消する.

出力ファイルの2番目はRAPDのリスティング出力用で，形名無指定の場合は，.LSが付加される. ファイル無指定の場合はリスティングパスを実行しない.

出力ファイルの3番目はLLDRのマップリスティング用で形名無指定の場合は，.MPが付加される. ファイル無指定の場合はTTY：が指定される. /M, /U, /Pのいずれかのオプションを指定しない時は不要である. 入力ファイルは形名指定がない時には，.FTが付加される.

オプションは次の通りである.

/A	リースプログラムをラインプリンタに出力する.
/K	コンパイラ出力（RAPD入力）SYS:FORTRN. TMに格納され，通常はRAPDのアセンブル終了後，自動的に抹消される. Kオプションを指定すると，SYS:FORTRN.TMは抹消されない.
/L	RAPDのアセンブル終了後. LLDRを呼び，オブジェクトをロードする. ロード後，コマンド行の終がESCかCRかによりキーボードモニタ（・）またはLLDR（*）に戻る.
/G	RAPDのアセンブル終了後，LLDRを呼んでRAPDオブジェクトをロードし，さらに必要なライブラリ関数をSYS:LIB555からロードして実行する. 実行開始番地は＝オプションで指定しなければMAINが自動的に指定される.
/N	RAPDのリスティング出力はシンボル表だけ（リスティングパスを実行しない）
/S	RAPDのリスティング出力はシンボル表を除く（N，Sの両オプションはリスティング出力の指定が無ければ無意味である）.

以下のオプションはLまたはGオプションを指定した時のみ有効である.

/I	万能入力ルーチンを使用する.
/O	万能出力ルーチンを使用
/H	万能入出力ルーチンは2ページドライバを使用する.
＝n	n 5桁のオクタル番地. 実行開始番地の指定
/n	nは1～7の数字，LLDRはn−1以下のフィールドにはロードしない.
/M	ローディングマップ（EXTシンボルの絶対番地表示）を出力する.
/P	各フィールドの残りのページ数を出力する（M. U, Pの各オプションで出力ファイルの指定がない時はTTY：が指定される）.

例

*TEST/G

コンパイラはDSK：TEST.FTをコンパイルしてSYS：FORTRN.TMを作る.

RAPDはFORTRN.TMをアセンブルしてFORTRN.TMを作り、FORTRN.TMを抹消する.

LLDRはFORTRL.TMをロードしてFORTRL.TMを抹消する. さらに必要なライブラリ関数をロードして、MAINから実行を開始する.

*,, LPT：＜TEST/L/M

前例と同様であるが、ローディングマップをLPTに出力する. ただしライブラリ関数はロードされない. LLDRのコマンド行に戻る（*).

*,,LPT:TESTL/M$

前例と同様であるが最後はESC、ライブラリ関数もロードして、ローディングマップをLPTに出力する. キーボードモニタ（.）に戻る. 以後、SAVE、START、PODなどが実行できる.

*BIN,LIST,MAP＜TEST（KMG)

入力ファイル	DSK:TEST.FT
バイナリ出力ファイル	DSK:BIN.RL
リスティング出力ファイル	DSK:LIST.LS
ローディングマップ出力ファイル	DSK:MAP.MP
フォートラン出力ファイル	SYS:FORTRN.TM

コンパイル、アセンブル、ロード、実行の順に自動的に行う.

*TEST/K

DESK:TEST.FTをコンパイルしてDSK:FORTRN.TMを作る. さらにRAPDを呼びアセンブルもリスティングも出力しない. コンパイラとアセンブラのエラーメッセージはTTYに出力される.

9—3—3　アセンブラの実行

アセンブラRAPDはフォートランコンパイラから自動的に呼出される以外に、単独のアセンブラとして、フォートランコンパイラ出力をあらためてアセンブルしたり、プログラマのコーディングしたソースプログラムをアセンブルしたりするのにも用いられる.

.R RAPDでロードされる.

コマンド行の出力ファイル指定（0～3個）についてはコンパイラの場合と同じである. 入力ファイルは形名指定がない場合は.RAが付加される.

オプション指定は次の通りである.

/L、/G、/N、/S、/I、/O、＝n、/M、/U、/P、/nについてはコンパイラの場合と同じである.

/F　入力ファイルはフォートランコンパイラの出力である.

9—3—4　ローダの実行

リンキングローダ（LLDR）はコンパイラ（FORT）またはアセンブラ（RAPD）で/Lまたは/Gが指定されれば、自動的にロードされるが、単独にも操作することができる.

コマンド行には0または1個の出力ファイルと1～9個の入力ファイルが指定できる. 出力ファイルはFORTまたはRAPDに対する3番目の出力ファイル指定と同じである. 入力ファイルの数は1つのコマンド行については9個以下であるが、何行にもわたることは差支えない. 入力ファイルは形名指定のない時は.RLが付加される.

オプション指定は次の通りである.

/G、/M、/U、/P、/n、＝nについてはコンパイラの場合と同じである.

/Iおよび/Oコンパイラに対するオプション指定と同じであるが、最初のコマンド行でのみ指定できる.

/L　入力ファイルはライブラリファイル（ダイレクトリ付の特殊形式ファイルで必要なものだけが選択的にロードされる）で，入力ファイルが1個以上指定されても最初のファイル以外は無視される．

10．モニタの操作運用

10—1　紙テープからのシステムビルディング

マークⅡモニタシステムはいくつかのブロックにわかれたシステムビルダ紙テープ，およびイメージ形式ライブラリ紙テープの形で供給される．

フロッピーディスクを含むシステムでは，システムおよびライブラリはそれらを収容したフロッピーディスクの形で供給される．

10—1—1　紙テープローダのフードストラップ

1. システムビルダ紙テープを高速紙テープ読取機（あるいは，コンソールタイプライタのテープ読取部）にセットする．(POWER ON, ON-LINE)
2. 次項のブートストラップをスイッチレジスタからセットする．

   システムデバイスがカートリッジ磁気ディスク固定ヘッド，磁気ディスク，フロッピー磁気ディスクのいずれかである場合はあらかじめアドレス書込みをおこなっておく．アドレス書込みのルーチンは各機器の診断プログラムに含まれている．
3. 0番地（コンソールタイプライタの場合は1番地）から計算機をスタートさせる．コンソールタイプライタの場合は紙テープ読取部をSTARTにする．

10—1—2　シスビルディングにおける指定

1. シスビルダ紙テープの前半（Part 1）を読みこむとコンソールタイプライタにDATE:と印刷されるから，月／日／年と数字（年は西暦の下2桁）で間にスラッシュをいれてキーインする．コンソールタイプライタの紙テープ部分から読みこんだ場合はPart 1とPart 2の間のリーダ部分で読取り機をストップさせてから日付を入力する．月は1〜12，日は1〜31，年は74〜77とする．最後に↵（RETURN）を入力すると形式が正しければ先に進む．正しくない時は再びDATE:と印刷される．

   紙テープの読取りエラーの場合はREAD ERROR, TRY AGAINと印刷されるから10—1—1に戻ってやり直す．
2. HIGHEST FIELD:と印刷されたら主記憶装置の容量に応じて1〜7の数字を入力する．この数字は実際の記憶容量に等しいかそれよりも小さければさしつかえない．たとえば，16K語のシステムで2と入力すれば，モニタおよびシステムプログラムは0，1，2の3フィールドのみを使用して最後のフィールドを使用することはない．実際の記憶容量も大きい値を指定した場合にはプログラムによっては正しく動作できない可能性がある．
3. KEEP OLD DIRECTORY？と印刷されたらYまたはNを入力する．Yとするとシステムの書き直しだけをおこない，ダイレクトリとローダ（LOAD）はそのままとする．
4. PTR？，PTP？，LPT？に対しては各々システム構成にしたがってYまたはNを入力する．PTR？に対してNと答えた場合にはPTPもないものとしてPTP？は省略される．

   PTR？，PTP？，LPT？の各々についてNと答えた場合はPTR：，PTP：，LPT：の各デバイス名はTTYの対応する部分を意味するようにシステム構成がおこなわれる．
5. SYS：と印刷されたら，システム構成にしたがって

固定ヘッド磁気ドラム	CFD
カートリッジ磁気ディスク	IOD 0 またはIOD 2
固定ヘッド磁気ディスク	CFX

フロッピー磁気ディスク　　CFP

のいずれかを入力する．カートリッジ磁気ディスクを除いてSYS：，DSK：のいずれに対しても指定されたデバイスが割当てられる．また，単にCRを入力した場合にはSYS＝IOD 0．DSK＝IOD 1の割当がおこなわれる．

カートリッジ磁気ディスクの場合はさらにDSK：と印刷されるからSYS，IOD 0，IOD 1，IOD 2，IOD 3のいずれかを指定する．

6. システムビルダ紙テープの後半を自動的に読込み，システムビルディングを終了し，入出力タイプライタに↑を印刷する．↑はイメージ形式のシステムテープ読込みの用意ができたことを示す．

Part 2の読込みに読取りエラーが起った場合には

READ ERROR, SET TAPE & CONTINUE

と印刷される．

紙テープのPart 1とPart 2の間のリーダ部分を読取機にセットする．高速紙テープ読取機の場合は入出力タイプライタのスペースバーを押すことにより読込みを開始する．低速読取機（入出力タイプライタのリーダ部）の場合は単にリーダ部のスイッチをSTARTとすることにより読込みを開始する．

10—1—3　システムプログラムの読込み

イメージ形式システムプログラムライブラリ紙テープを読取機にセットしてスペースバーを押すと読込みを開始する．各ファイルごとにファイル名とファイルデートを印刷し，正しく読込みが終了すれば次のファイルに進む．紙テープの読取りエラーを起した場合にはBCC ERRORと印刷して↑に戻るから，一つ前のヘッダ部分を読取機にセットして再度読込みを試みる．

イメージ形式の紙テープを終りまで正しく読込んだ場合には↑に戻るので次のテープを読込ませる．必要なすべてのイメージ形式紙テープの読込みが終った時にはコントロールCを入力することによってモニタにコントロールが移り・（ドット）が印刷される．

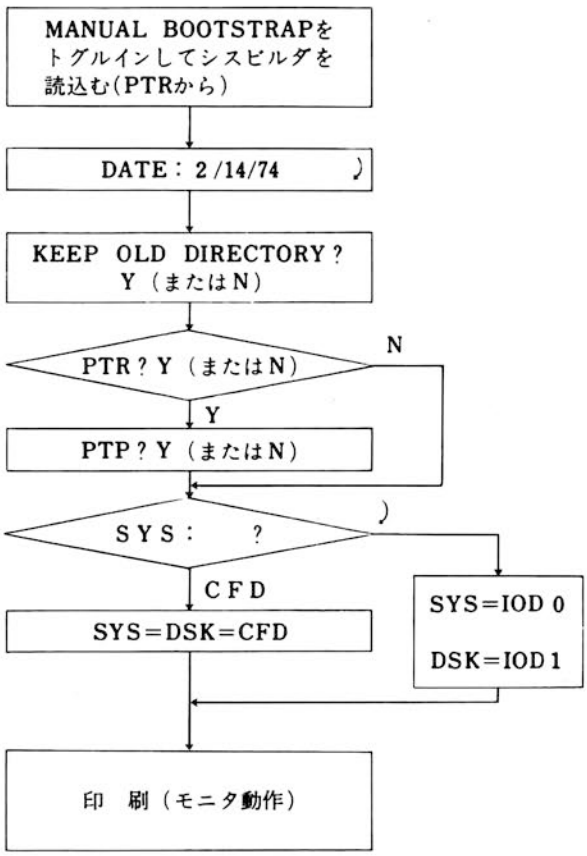
10—1—4　ダイレクトリのイニシャライズ

SYS：デバイスについてはシステムビルダがダイレクトリをイニシャライズしている．その他のデバイスが含まれる場合（たとえばカートリッジディスクのその他のデバイス）にはCONVRTプログラムの／Zオプションにより，ダイレクトリのイニシャライズをおこなう．

例　(SYS＝IOD 2の場合)

```
.R  CONVRT
*IOD 0 :, IOD 1 :, IOD 3 :/Z  ↵
```

10—1—5　システムビルディングのフロー図



10—1—6　フロッピーディスクを含むシステム

フロッピーディスクを含むシステムではシステムおよびライブラリは紙テープを使わずに能率的なシステムビルディングが可能である.

1.　フロッピーディスクをシステムデバイスとして使用する場合

後述のフロッピーディスク用ブートストラップ（10—2—4）によってシスヘッドをレストアする.

2.　フロッピーディスク以外のデバイスをシステムデバイスとする場合

1.と同様にフロッピーディスクをシステムデバイスとして一旦システムスタートをおこなう.

プログラムCONFIGを使用してシステムの再構成をおこなう.

例

```
.R　CONFIG
SYS：=IOD 2
DSK：=IOD 3
OTHERS：CFP 0，CFP 1
```

10—2　再スタートブートストラップ

何等かのミスオペレーションでフィールド0，またはフィールド1の最終ページ（7600～7777）の一部を破壊した場合にはシステムレジデント部分のブートストラップをおこなう必要がある.

10—2—1　磁気ドラム

以下のブートラトラップをスイッチレジスタからトグルインして07750からスタートする.

番地　／　内容
07750 ／ 7600
07751 ／ 6603
07752 ／ 6622
07753 ／ 5352
07754 ／ 5752

10—2—2　固定ヘッド磁気ディスク

以下のブートストラップをトグルインして07600からスタートする.

番地　／　内容
07600 ／ 7600
07601 ／ 6602
07602 ／ 1200
07603 ／ 6607
07604 ／ 1207
07605 ／ 6623
07606 ／ 5206
07607 ／ 1000

10—2—3　カートリッジ磁気ディスク　（IOD 0 をSYS：としている場合）

番地　／　内容
07746 ／ 6157
07747 ／ 1354
07750 ／ 6173
07751 ／ 7746
07752 ／ 6173
07753 ／ 5353
07754 ／ 1400
07752 ／ 6173
07753 ／ 5353
07754 ／ 1400

7746をPCにセットしてからSR＝1000とし，START，CONTと順に押す.

10—2—4　フロッピー磁気ディスク

番地　／　内容
07746 ／ 6557
07747 ／ 1354
07750 ／ 6573
07751 ／ 7748
07752 ／ 6573
07753 ／ 5353

07754 ／ 1400

7746にPCをセットしてからSR＝1000とし，START，CONTと順に押す.

10—2—5 XRIM形式ブートストラップ紙テープによる法

システムビルディング後，BOOTプログラムを呼び，XRIM形式紙テープを作っておく．この紙テープをリーダにセットし次のブートストラップをトグルインし0番地（コンソールタイプライタ入力の場合は1番地）からスタートする.

高速紙テープ読取機

番地	内容
0000	6014
0001	6011
0002	5001
0003	6016
0004	7530
0005	3410
0006	7106
0007	5001
0010	7377

コンソールタイプライタのリーダ部

番地	内容
0001	6031
0002	5001
0003	3013
0004	6036
0005	1013
0006	7530
0007	3412
0010	7106
0011	5001
0012	7377

10—2—6 BIN形式ブートストラップ紙テープによる法

12K語以上の主記憶を含むシステムでは，フィールド2以降の最終ページにBIN紙テープローダを常駐させておくことができるので，この紙テープローダによってSYSヘッドBINテープを読込み07600から再スタートさせる.

BIN形式紙テープはシステムライブラリのBOOTBNによってパンチされる.

付録A エラーメッセージ

A—1 シスビルダ

システムデバイスに対する入出力エラーの場合RUNが消える（HLT）．システム構成の指定についての形式不良に対しては？を印刷して，質問を繰返す.

A—2 キーボードモニタ（システムレパートリ）

SYSTEM IO ERROR	入出力エラー
××××？	指定された9個以外のコマンド
TOO FEW ARG	指定が不足
DEVICE NOT AVAILABLE	指定されたDEVICEがない.
FILE NOT FOUND	指定されたFILEがない.
BAD CORE IMAGE	CCBが正しくない（たとえば，SV形のファイルをCONVRTの /Aまたは/Bでコピーした場合など）.
USER ERROR 0 AT××××	ロード中に入力エラー
BAD ARGS ILLEGAL ARG.	SAVEの指定が正しくない.
SAVE ERROR	SAVE時のI/Oエラー，コア上のプログラムは不変
BAD DATE	DATEの指定が正しくない.
NO!!	再スタート禁止のプログラムにSTARTを指示した.

A—3 コマンドデコーダ（コマンド行）

ILLEGAL SYNTAX	コマンド行の形式不良
TOO MANY FILES	指定ファイル数過大
DEVICE DOES NOT EXIST	指定されたDEVICEがない.
FILE NOT FOUND	指定されたFILEがない.

A—4 エディタ（EDIT）

通常の正しくないコマンドに対しては？を印刷して＃に戻る.

重大なエラーについては？n↑C（nは0～4の数字）を印刷してモニタ（・）に戻る.

nの意味は次の通りである.

0	入力時のエラー
1	出力時のエラー
2	ファイルクローズのエラー
3	ファイルオープンのエラー
4	入出力ドライバのエラー

出力デバイスの割当てられたスペースがいっぱいになった時はFULLと印刷してコマンドデコーダ（*）に戻る. 新しい出力デバイスを指定すれば，作業を続行できる. この時，出力デバイスの指定を誤まると，キーボードモニタに戻り，以後の出力ファイルは失われる. 入出力ファイル名が同じ場合には，ファイルの失われた部分は回復不可能となる.

A—5 アセンブラ（SAPD）

アセンブラのエラーメッセージはデバイスTTY：に出力される. 以下でAAAAAAは6字以下のシンボル名，fnnnnは5桁のオクタル数字でエラーの発見された番地，CCCCは4桁のオクタル数字で不正文字の値，××××は4桁のオクタル数字で意味がない.

US AAAAAA AT fnnnn	未定義シンボルが使われた.
IC cccc AT fnnnn	不正文字が使用された.
RD AAAAAA AT fnnnn	＝による再定義が行われた.
ID AAAAAA AT fnnnn	すでに定義されたシンボルを再び，で定義しようとした.
II AAAAAA AT fnnnn	現在ページに無いシンボルの間接引用
PE xxxx AT fnnnn	現在ページ定数表とアセンブルされた命令語とが同じ番地に重なった.
PO xxxx AT fnnnn	プッシュダウンリストの溢れ，モニタに戻る.
SE AAAAAA AT fnnnn	シンボル表の溢れ，モニタに戻る.
IO xxxx AT fnnnn	入出力エラー，モニタに戻る.
LE xxxx AT fnnnn	リスティング出力デバイスの割当てられた場所が無余地，リスティングを出力しない.
BE xxxx AT fnnnn	バイナリ出力デバイスの割当てられた場所が無余地，バイナリを出力しない.

ファイル名AAAAAAが発見できない(・に戻る).

A—6 バイナリコーダ（LOAD）

I/O ERROR FILE ＃n	n番目のファイルをロード中に入力エラー
BAD INPUT. FILE ＃n	n番目のファイルがバイナリ形式でない.
BAD CHECKSUM,FILE＃n	n番目のファイナのチェックサムエラー
NO INPUT	入力ファイルなし.

ローダはエラメッセージの印刷後コマンドデコーダ(＊)に戻る.

A—7　フォートランコンパイラ（FORT）

コンパイラはエラーを発見すると、その行および以下のいずれかのエラーメッセージを印刷する.
以降もエラーチェックはプログラムの終りまで行われる.

ARITHMETIC EXPRESSION TOO COMPLEX
EXCESSIVE SUBSCRIPTS
ILLEGAL ARITHMETIC EXPRESSION
ILLEGAL CONSTANT
ILLEGAL CONTINUATION
ILLEGAL EQUIVALENCING
ILLEGAL OR EXCESSIVE DO NESTING
ILLEGAL STATEMENT
ILLEGAL STATEMENT NUMBER
ILLEGAL VARIABLE
MIXED MODE EXPRESSION
SYMBOL TABLE EXCEEDED
SYNTAX ERROR
I/O ERROR
NO ROOM FOR OUTPUT
RAPD NOT FOUND
NO END STATEMENT

未定義文番号はRAPDでアセンブルするまで発見されない.
なお、**A—8**、**A—9**、**A—10**も参照のこと.

A—8　リロケータブルアセンブラ（RAPD）

RAPDでエラーが発見されると次の形のエラーメッセージが入出力タイプライタに印刷される.

× AT AAAAAA +nnnn

×はエラーコード、AAAAAAはシンボル名、nnnnはオクタル数でシンボルが定義されて以後の行数を示す. エラーコードの意味は次の通りである.

A	CALLの後のＡＲＧの数の不一致
C	不正文字
D	入出力エラー
L	/Lまたは/GでSYS：LLDR.SVがない.
M	シンボルの多重定義
I	擬似命令のアーギュメント不良およびOPRの命令の不正複合
E	END文なし.
S	シンボル表溢れまたは出力ファイル溢れ
U	未定義シンボルあり.

なお、シンボル表が出力された場合には未定義
シンボルには各々にUNDFの注が付く.
なお、**A—9**、**A—10**も参照のこと.

A—9　リンキングローダ（LLDR）

リンキングローダは次の形のエラーメッセージを印刷する.

ERROR nnnn

nnnnは4桁のオクタル数で以下の意味を持つ.

0000	/Iまた/Oがコマンド行の第1行目以外で指定された.
0001	シンボル表溢れ（EXTシンボルが64個以上）
0002	プログラム過大（コア不足）
0003	最大のCOMMON指定が第1ファイル以外で行われた.
0004	チェックサムエラー
0005	バイナリファイル形式不良
0006	出力エラー
0007	入力エラー
0010	MAINがなく，実行開始番地の指定も行われなかった.
0011	I/Oドライバのロードエラー
0012	入出力エラー

USER ERROR 1 AT ××××××

ロードされていないサブルーチンへのコール00005に停止　ローダスクラッチ（#40—47）のリーダエラー

なおA—10も参照のこと

なお，A—8も参照のこと.

A—10　フォートランライブラリ（LIB555.）

フォートランまたはリロケータブルアセンブラのオブジェクト実行時に，ライブラリサブルーチンの中でエラーが発見されると××××ERROR AT nnnnnの形のエラーメッセージを印刷する. ××××はエラーコードで以下に説明する. nnnnnは5桁のオクタル数字でエラーの発見されたフィードルおよび番地を示す. 重大なエラーについてはモニタ（・）に戻る.

1．モニタに戻るもの.

ALOG	負数の対数
IOER	i）万能入出力指定で/Iまたは/Oが指定されなかった.
	ii）IOPENまたはOOPENの指定不良
	iii）入出力エラー
CHER	CHAIN不良(指定ファイルなし)
FMT1	FORMAT文不良
FMT2	I形式入力での不正文字
FMT3	F形式入力またはE形式入力での不正文字

ただし，EMT2またはFMT3で入力が入出タイプライタの場合はそのREAD文を最初から繰返す.

2．実行を続行するもの.

DIVZ	0で割る（最大値を結果とする）
EXP	結果過大（　〃　）
OVFL	〃　（　〃　）
FLPW	負数の実数べき（絶対値を用いる）
SQRT	負数の平方根　（　〃　）
FIX	結果過大（−2047または2047を結果とする）

付録B　入出力タイプライタにCS1000を用いる場合の操作

CS—1000形入出力タイプライタはカナ文字を含め132字／行の印刷が可能でピンフィードも内蔵されている. 標準のCS—100とくらべるとキーボード配列が異り，操作に多少の相異がある.

CS100とCS1000のキー操作の相異

1．シフトキーの代りに英記号を用いる．英数を押して戻さないとシフトのままとなる．

2．以下の文字については印字およびキー操作が異る（下段，英数側にあるのでシフト不要である）．

CS100	CS1000	
↑（シフトN）	＜	
←（シフトO）	—	（英数）
〔（シフトK）	〔	（英数）
〕（シフトM）	〕	（英数）
＼（シフトL）	¥	（英数）
ESC	EOT	（英数）

付録C　SAPDおよびRAPDの内蔵シンボル表

C—1　文字セット

1．英字

A—Zの26文字と〔＼〕↑←

SAPDでは〔を除く．RAPDでは←を除く．

2．数字

0—9

3．特殊文字

，／（″—＃；スペース　タブ　CR　．＋＊！＝〔＄（SAPDのみ）

4．無視文字

LF．FF．RUBOUT

217および216（SAPDのみ）

5．不正文字（TEXT，コメントおよび″の直後を除く）

上記以外のすべての文字

C—2　擬似命令シンボル

DECIMAL（RAPDではDECIM）OCTAL　PAUSE　I　PAGE　TEXT（以上共通）

FIXTAB　ENLIST　NOLIST　QBLOCK　ATEXT　FIELD（以上SAPDのみ）

ABSYM　ARG　BLOCK　CALL　COMMN　CPAGE　DVMMY　EAP　END　ENTRY　FORTR　IF　LAP　OPDEF　REORG　RETRN　SKPDF　ACH　ACM　ACL（以上RAPDのみ）

C—3　番地参照命令シンボル

AND　TAD　ISZ　DCA　JMS　JMP（共通）

FADD　FSUB　FMPY　FDIV　FGET　FPUT　FNOR　FJMP　FJMS　FISZ（以上SAPDのみ）

INC（RAPDのみ）

C—4　その他の命令シンボル

NOP　IAC　PAL　RTL　RAR　RTR　CML　CMA　CLL　CLA　HLT　OSR　SKP　SNL　SZA　SNA　SMA　SPA　CIA　STL　STA　ION　IOF　KSF　KRB　TSF　TLS　RSF　RRB　RFC　PSF　PLS（共通）

GLK　LAS　KCC　KRS　TCF　TPC　PCF　PPC　CDF　CIF　RDF　RIF　RIB　RMF　QFETCH　QSERCH　QOPEN　QCLOSE　QCD　QCHAIN　QERROR　QLOCK　QFIRE　QUIZ　QRESET（SAPDのみ）

SPC（RAPDのみ）

付録D　フォートランおよびRAPDライブラリーらん

1. READ　WRITE　IOH
2. FAD　FSB　FMP　FDV　STO　FLOT　FLOAT　FIX　IFIX　IFAD　ISTO　ABS　CHS
3. OPEN　GENIO　EXIT　ERROR　CKIO
4. IFPOW　FFPOW　EXP　ALOG
5. IREM　IABS　DIV　MPY　IRDSW　CLEAR　SUBSC
6. SIN　COS　TAN
7. ATAN
8. SQRT
9. IIPOW　FIPOW
10. IOPEN　OOPEN　OCLOS　CHAIN

❹ マークIIモニタシステム〈その2〉

本章では、モニタシステムのUSR(USER SERVICE ROUTINE)について説明する．ユーザはその機能を利用することにより，プログラム作成にあたってマークIIをより有効に活用することができる．その内容の概略は下表の通りである．

USR機能一覧表

Function Code	Name	Operation
1	QFETCH	I/Oドライバを指定エリアにローディングし，そのドライバのエントリーアドレスを知る．
2	QSERCH＊	指定デバイスのダイレクトリを検索し，指定されたファイルのスターティングブロックNO.とファイル長を知る．
3	QOPEN＊	指定デバイスに仮登録ファイルを作る．
4	QCLOSE＊	指定デバイス上の仮登録ファイルをパーマネントファイルにする．その時，同一名のファイルは抹消される．
5	QCD	コマンド・デコーダを呼び出す．
6	QCHAIN	システムデバイス上の指定されたコアイメージファイルをローディングし，コントロールを，そのプログラムに引き渡す．
7	QERROR	以下の形式のエラーメッセージをコンソールタイプライタにプリントする． USER ERROR n AT LOCATION fnnnn
10	QLOCK	USRをコアに常駐させる． 以降USRはJMS TO 10200によりコールされる．
11	QFIRE	コアからUSRを取り除き，フィールド1の0000～1777の内容はQLOCKがなされる前の状態に復元される．
12	QUIZ	指定されたデバイスが存在するか(システムに登録されているか否か？)，もしくは，そのドライバがコアにあるかどうかの確認を行う．
13	QRESET	システムテーブルを初期状態にする．
14～17		for future use

❺ 演算インタープリータ

§ 1　3倍長整数演算，浮動小数点システム

1．概　要

本システムは，3倍長整数演算パッケージA，3倍長整数演算パッケージB，浮動小数点演算パッケージの3種類で構成される．

3倍長整数演算パッケージ2種は，それぞれ有効数字10桁の整数演算および入出力を行う，インタプリタ形式のソフトウェアパッケージである．浮動小数点演算パッケージは小数部6桁，指数部は−615から＋615までの浮動小数点演算および入出力を行うインタプリタ形式のソフトウエアパッケージである．

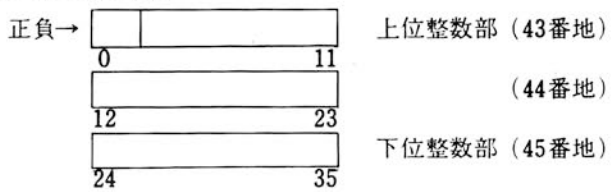
2．使用機器構成

CEC555シリーズの記憶容量4KW以上および入出力タイプライタの構成から使用可能である．

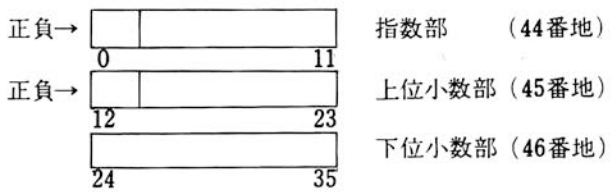
3．処理方法

各パッケージはそれぞれ3ワードの擬似アキュムレータPACをもつ．命令語は整数演算用PAC，浮動小数点演算用PACに対して機能する．

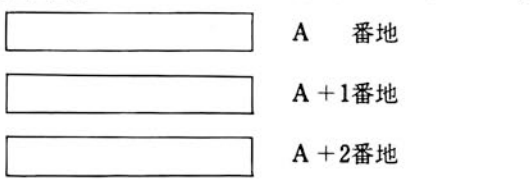
整数演算用PAC



浮動小数点演算用PAC



演算命令のオペランド部で指定する記憶装置内のアドレスは，数値データ3ワードの先頭を示す．例えば，加算命令FADD　AではPACとA，A＋1，A＋2番地の加算を行う．



4．命令一覧

ＦＧＥＴ　PACの内容をクリアし実効アドレスの内容をPACに転送する．実効アドレスの内容は不変である．

ＦＰＵＴ　PACの内容を実効番地に格納する．PACの内容は不変である．

ＦＡＤＤ　PACの内容と実効番地の内容を加算し，結果をPACに格納する．

ＦＳＵＢ　PACの内容から実効アドレスの内容を減算し，結果をPACに格納する．

ＦＭＰＹ　PACの内容と実効アドレスの内容を乗算し，結果をPACに格納する．オーバーフローのチェックは，行っていないので注意が必要である．

ＦＤＩＶ　PACの内容を実効アドレスの内容で除算する．浮動小数点演算パッケージでは商はPACに格納される．

3倍長整数演算パッケージでは商はQT（46.47.50番地）に格納され，剰余はPACに格納される．

商の符号は除数，被除数の積，剰余の符号は被除数の符号に等しい．

ＳＱＵＡＲＥ　PACの内容を2乗してPACに格納する．

ＳＱＲＯＯＴ　PACの内容を½乗して，PACに格納する．

ＳＩＮ　SIN（PAC）をPACに格納する．演算単位はラジアンとして行う．

ＣＯＳ　COS（PAC）をPACに格納する．演算単位はラジアンとして行う．

ＡＴＡＮ　TAN^{-1}（PAC）をPACに格納する．演算単位はラジアンとして行う．

ＥＸＰ　exp（PAC）を，PACに格納する．

ＬＯＧ　log（PAC）を，PACに格納する．

ＦＪＭＰ　3倍長モードの中の指定された番地にジャンプする．直接アドレス指定の時，飛先番地が，0ページであってはならない．

ＦＩＳＺ　指定された番地の内容に1を加算し，結果が0ならば，3倍長モードの中でスキップする．0ページ内番地に適用することはできない．

ＦＪＭＳ　指定された番地に，そのときのPC＋1をセットし，指定された番地の次の命令にジャンプする．直接アドレス指定の時，0ページ内の番地に適用することはできない．

ＦＡＢＳ　PACの内容の絶対値をPACに格納する．

ＩＮＴＧＲ　PACの整数部分のみをとる．

ＦＲＡＣＴ　PACの小数部分のみをとる．

ＦＩＸ　浮動小数点で表現されたPACの内容を単精度整数に変換し，45番地に格納する．

ＦＬＯＡＴ　45番地の内容を浮動小数点で表現し，PACに格納する．

ＦＮＥＧ　PACの正負を反転する．

ＦＣＩＡ　PACの正負を反転する．

ＦＣＬＡ　PACの内容をゼロにする．

ＦＣＲＬ　TTY（LPT）にCR／LFを印刷する．

ＦＳＴＹ　出力機器をTTYにセットする．

ＦＳＬＰ　出力機器をLPTにセットする．

ＦＣＤＦ　インタプリータ内のデータフィールドチェンジを行う（FEXTで，CURRENT FIELDにRESETされる．FADD，FSUB，FPUT，FGET，FMPY，FDIVの間接参照に対して有効である）．

ＦＨＬＴ　プログラムを停止する．

ＦＣＨＧ　PACとQTの内容を入れかえる．

ＦＮＯＰ　無効命令

ＦＳＫＰ　無条件に次の命令をスキップする.

ＦＳＭＡ　PACの内容が負の時に，次の命令をスキップする.

ＦＳＰＡ　PACの内容が０以上の時に，次の命令をスキップする.

ＦＳＮＡ　PACの内容が０でない時に，次の命令をスキップする.

ＦＳＺＡ　PACの内容が０の時，次の命令をスキップする.

ＦＳＭＡ　ＦＳＺＡ　FSMAとFSZAの複合動作

ＦＳＰＡ　ＦＳＮＡ　FSPAとFSNAの複合動作

ＦＩＮ　TTYからの入力をPACに格納する．抹消の入力があると←を印字し，PACをゼロに戻し，再度入力を行う.

	3倍長整数演算パッケージ　A	3倍長整数演算パッケージ　B	浮動小数点演算パッケージ
入力可能な文字	数字，－，コンマ	数字，－，コンマ	数字，＋，－，E，小数点
終了文字	上記以外	上記以外	上記以外
最終入力ASCII値	57番地に記憶	56番地に記憶	57番地に記憶
入力の終了指示	60番地≒0で終了	57番地≒0で終了	60番地≒0で終了
CRの入力に対する処置	CR，LFを出力	CR，LFを出力	56番地≒0でCR，LF出力

ＦＢＣＤ　PACの内容を符号および12桁の４ビットBCDに変換して，51～55番地に格納する．３倍長整数演算パッケージにだけ有効である．パッケージAでは，PACの内容は保存されないが，パッケージBでは保存される.

ＦＢＯＵＴ　51～55番地の内容とTTY（LPT）に印字する.

符号は負の場合，有効数字の直前に印字される．パッケージA命令では右端から，３桁ごとにカンマが挿入される．ゼロ，サプレスされたものが出力されるが，符号およびコンマを含めた全印字桁数は，パッケージBでは61番地，パッケージAでは62番地にあらかじめ指定しておく.

ＦＯＵＴ　３倍長整数演算パッケージではFBCD，FBOUTの複合動作を行う.

浮動小数点演算パッケージでは，55番地がゼロでない時は，出力後CR／LFを出力する.

62番地にはあらかじめ出力すべき桁数を入れておく.

ただし，小数点，符号は含まない.

62番地＝０の時は，標準のE形式で出力する.

E形式　±０．××××××E±××

62番地は，始めはゼロにセットされている．またACの内容は10進小数点の右側の桁数を入れておく.

AC＝0の時，小数点は印字されない.

PAC＝（678.234）$_{10}$

62番地	AC	出力
3	0	67.8
4	1	678.2

0　　0.678234E+03

＋符号は，スペースとして印字され，－符号の有効数字の直前に印字される．

62番地の内容が，PACの整数部の桁数より小さいときは〝×〟が印字される．

FNOR　PACの内容を正規化（NORMALIZE）する．

正規化とは，浮動小数点表現における小数部の値を下記のようにすることである．

$\frac{1}{2} \leqq |$小数部$| < 1$

正規化により，小数部の有効桁数を多くすることができる．

FEXT　3倍長整数演算パッケージ，浮動小数点パッケージ使用の終了を示す．

5．使用法

インタプリタの各エントリポイントに対して，JMS命令で制御を移行する．

3倍長整数演算パッケージA…JMS　6200

〃　　B…JMS　6400

浮動小数点演算パッケージ……JMS　5600

いったんインタプリタの制御下に入ると，通常の命令は使用できない．インタプリタの制御を離れるためにはFEXT命令を用いる．あらかじめエントリポイントを7番地に設定しておくと，コーディングが容易になる．

```
#  TEST
 CALL=JMS  I  7
*7
        5600
*200
START,  CLA  IAC
        DCA  55
        IAC
        DCA  56
        DCA  62
        CALL
        INPUT
        SQROOT
        OUTPUT
        FEXT
        JMP  START
$
```

6．パッケージ別命令一覧

	3倍長整数演算パッケージA	3倍長整数演算パッケージB	浮動小数点演算パッケージ
FGET	○	◎	○
FPUT	○	◎	○
FADD	○	◎	○
FSUB	○	◎	○
FMPY	○	◎	○

	3倍長整数演算パッケージA	3倍長整数演算パッケージB	浮動小数点演算パッケージ
FDIV	○	◎	○
SQUARE	×	×	1
SQROOT	×	×	2
SIN	×	×	3
COS	×	×	4
ATAN	×	×	5
EXP	×	×	6
LOG	×	×	7
FJMP	×	◎	×
FISZ	×		×
FJMS	×		×
FABS	×	×	10
INTGR	×	×	11
FRACT	×	×	12
FIX	×	×	15
FLOAT	×	×	16
FNEG	× 6	×	17
FCIA	×	−	×
FCLA	×		×
CLEAR	7	×	×
FCRL	×	10	×
FSTY	×	11	×
FSLP	×	12	×
FCDF	×		×
FHLT	×		×
FCHG	×		×
FNOP	×		×
FSKP	×		×
FSMA	×		×
FSPA	×		×
FSNA	×		×
FSZA	×		×
FSMA	×		×
FSPA	×		×
FIN	1	1	13 JMS7400
FBCD	2	2	×
FBOUT	3	4	×
FOUT	5	14	14 JMS7200
FNOR	×	×	
FEXT			

× 使用不可　　○ 間接指定可

◎ オートインデックス可　　nn

アセンブラで使用する時に定義が必要

JMSnnnn 単独のTMS命令でも実行可能

7. 使用メモリ

	3倍長整数演算 パッケージA	3倍長整数演算 パッケージB	浮動小数点演算 パッケージ
0　　頁	40～73	40～64	15, 40～62
インタプリタ	6200 ～ 7577	5600 ～ 7577	4466 ～ 7577
P　A、C	43, 44, 45	43, 44, 45	44, 45, 46
QT（除算の商）	46, 47, 50	46, 47, 50	————
FIN最終入力ASCII値	57	56	57
FIN終了指示	60	57	60
FBCD出力	51,52,53,54,55	51,52,53,54,55	————
FBOUT桁数指定	62	60	————
FOUTCR/LF出力指定	————	————	55(初期値7777)
FOUT桁数指定	62(初期値0021)	60(初期値0021)	62
FINCR/LF出力指定	————	————	56
演算エラー指示	————	————	61

8. 注意事項

8—1　パッケージの混用

1つのプログラムで使用できるパッケージは1種類のみである.

8—2　フィールド

3倍長整数演算パッケージA，浮動小数演算パッケージでは，それぞれパッケージの存在するフィールド内でのみ使用できる.

3倍長整数演算パッケージBでは，パッケージの存在するフィールド以外から使用することができる.

```
CDF  ×0    / パッケージの存在するフィールドと同一であれば不要
CIF  N0    / パッケージの存在するフィールドを指定
JMS  6400
```

他フィールドからパッケージBをコールする方法として，次のような擬似エントリを設けると，CIFを意識せずにコールできる.

```
    #   TEST
CALL=JMS    I    7
*7          XFLPT
*200        CALL
              :
              :
            FEXT
```

```
XFLPT,0
        CLA
        TAD
        62N3
        DCA  I   (6400
        62×1
        JMP  6401
```

8－3　3倍長整数演算パッケージAで出力時のコンマを印刷しない方法

次のオーバレイを使用する.

```
*7250
      TAD    7357
*7323
      JMP  I  7266
*7355
      －15
```

8－4　3倍長整数演算パッケージBの出力命令

OPコード	出力番地	コンマ	ゼロサプレス
4	51～55	無	有
13	51～55	無	無
5	PAC	無	有
14	PAC	無	無
15	PAC	有	有

8－5　浮動小数点演算パッケージの入力形式

数値726.7は次の形式のどれでも入力できる.

```
726.7
.7267E3
.7267E+03
+7267E-1
```

8－6　出力命令

パッケージ内でのタイプライタ出力命令は次のようになっているので，出力命令FOUT等を使用する場合には，あらかじめTSL命令を実行しておく必要がある.

```
TSF
JMP  .  -1
TLS
```

8－7　演算エラー

3倍長整数演算パッケージで除数がゼロの除算を実行した場合，結果は保障できない.

浮動小数点演算パッケージで除数がゼロの除算を実行すると，PACには最大値が格納され，61番地は1となる．また負数の平方計算を行うと，絶対値に関して計算を実行後61番地を1とする．正常に終了した場合61番地はゼロとなっている.

9. プログラム例

```
/SOLVE THE QUADRATIC EQUATION
/AX*X+BX+C=0     FOR VALUES
/OF A,B,C.       TYPE OUT ROOTS

SOROOT=0002
SQUARE=0001
*7       5600            /POINTER TO INTERPRETER
*200     KCC
         TLS
BEGIN,   JMS 7400        /INPUT A
         JMS I 7         /ENTER INTERPRETER
         FMPY TWO        /2.0
         FPUT A          /STORE 2.0.A
         FEXT
         JMS 7400        /INPUT B
         JMS I 7         /ENTER INTERPRETER
         FPUT B          /STORE B
         FEXT
         JMS 7400        /INPUT C
         JMS I 7         /ENTER INTERPRETER
         FMPY TWO        /MULTIPLY BY2.
         FMPY A          /MULTIPLY BY2A
         FPUT TEMP       /4AC
         FGET B
         SQUARE          /B SQUARED
         FSUB TEMP       /B SQUARED -4AC
         SQROOT          /TAKE SQUARE ROOT
         FPUT TEMP       /STORE IT
         FGET B
         FMPY MIN1       /-1.B=-B
         FPUT B          /STORE
         FSUB TEMP       /-B-SQUARE ROOT
         FDIV A
         FEXT            /ANSWER IN FAC
         JMS 7200        /OUTPUT IT
         JMS I 7         /ENTER INTERPRETER
         FGET B          /-B
         FADD TEMP       /-B+SQRARE ROOT
         FDIV A
         FEXT            /ANSWER IN AC
         JMS 7200        /OUTPUT IT
         JMP BEGIN       /CONTINUE
TWO,     0002
         2000
         0000
MIN,     0001
         6000
         0000
TEMP,    0
         0
         0
A,       0
         0
         0
B,       0
         0
         0
C,       0
         0
         0
$
```

❻ ココル（対話計算）

1．概要

COCOL（ココル）は，Conversational Computing Languageの略語である．この言語は，技術者や研究者が，問題解決の手段として，コンピュータをあたかも電卓を使用するような気持で使用できるように工夫されている．

普通，コンピュータを使用するに当っては，解決しようとする問題を正確かつ精密に理解することのほかに，データのやりとりをする，入力，出力の機能についても厳密な論理解析が必要である．しかし，このCOCOLでは，入，出力の部分については，機械内部で自動的に処理され，利用者は，タイプライタの前に座って以下にのべる命令語の組合わせをタイプインすることによって，コンピュータと対話形式でデータのやりとりをしながら問題解決ができるようになっている．また，利用者の組んだプログラム（命令語の組合わせ）を，紙テープの形でとり出せるので，時間にゆとりがある時に設計計算の基本的なパターンをあらかじめ作っておき，利用時にはその紙テープをCPU（計算機本体）に読み込んで，数値だけを入力すれば，もっと楽に使用できる．

以下，その使用法とコマンド（命令語），機能，制限について述べる．

1－1　使用できる函数

1－1－1　基本函数

1．四則演算（×，÷，＋，－）

加算：＋で表わす．　　A＝B＋C　のように表現する．

減算：－で表わす．　　A＝B－C　のように表現する．

乗算：＊（アスタリスク記号）A＝B＊C　のように表現する．

割算：／（スラッシュ記号）〃A＝B／C

2．指数演算（a^n……aのn乗）

↑（アップアロー記号）で表わす．

A＝B↑C，のように表現する．この場合，**Cは正の整数でなければならない**．

その他のA^{-2}，$A^{\frac{1}{3}}$，等については後でLog x，e^xの応用のところで説明する．

3．平方根（$\sqrt{a}$）

A＝FSQT（B），のように表現する．

4．絶対値（|a|）

A＝FABS（B）と表現する．

5．符号部（aの正，負のみを得る）

A＝FSGN（B）（正の時は1.0負の時は－1.0と答えてくる）

6．整数部（aという数値の整数部のみをとる）

A＝FITR（B），と表現する．

〈例〉A＝FITR（45.2713）→A＝45

7．乱数発生器（乱数を発生する）

A＝FRAN（␣）と表現する

↖ブランク（空らん）　Aは任意の数になる．

サイコロバクチのプログラム，統計用のプログラム等に使用される．

1－1－2　特殊函数

1．指数函数（$f(x)=e^x$）

A＝FEXP（X）

eのx乗の数を作る.

2．対数函数（$f(x)=\log e\ X$）

A＝FLOG（X）

e対数をとる.

3．サイン（$f(x)=\mathrm{Sin}X$）

A＝FSIN（X）

ラジアンで表わされたXについてサインの値をとる.

4．コサイン（$f(x)=\cos X$）

A＝FCOS（X）

ラジアンで表わされたXについてコサインの値をとる.

5．アークタンジェント（逆タンジェント（$f(x)=\tan^{-1}X$）

A＝FATN（X）

Xに対するアークタンジェントをラジアンで作る.

1－1－3　その他の函数

その他の特殊な函数は1－1－2の函数の組合せで作る．一般の数学的な手法を用いる.

例

$$\tan X=\frac{\sin X}{\cos X}\quad A=\mathrm{FSIN}(x)/\mathrm{FCOS}(X)$$

$$\log_{10}X=\frac{\log e\ X}{\log e\ 10}\quad A=\mathrm{FLOG}(x)/\mathrm{FLOG}(10)$$

以上のように初等的な公式を使用する他に，微分学の力を利用したマクローリン展開公式や，行列式，漸化式を利用する方法もある．その手法については，各種の参考書が数多く出ているので参照されたい.

1－2　命令語（コマンド）

1－2－1　プログラミング用命令語

これは，利用者がプログラムを作るのに用いられる．すべてのプログラムは，以下の語と，前章の函数の組合わせによって作られる.

1：ASK（アスク）

英語のHe asked me the way（彼は，私に道をたずねた.）のASKと同様の意味で，「問う」という形で使用される.

☆使用例

ASK　X

この命令に出合うと，コンピュータは，タイプライタに：（コロン）を打ち，Xというものの数値を問うてくる．このことは：のあとに，タイプライタにより数（例えば：3.1415）を打ち込むと，コンピュータはXをその数として（X＝3.1415）記憶する.

この命令は，同じ計算式に対し，数値を，ランダムにくりかえし計算したい時に使用される.

He（↓コンピュータ）　Asked　me（↓利用者）　the way（↓×）

答として3.1415を打つことになる.

2：DO（ドウー）

You do Kill him!!（お前がヤツを殺せ！）のdoと同様，……をせよ!!の意味を持つ．しかし，

CoCoLのDOは，何かをしたあとの意味も持っていて，DOの命令を行ったあとは，必ずすぐ次にある命令を実行するようになっている．

例

```
.
.
.
DO  4.1
ASK X
.
.
.
```

このような時，4．1という場所に置いてある仕事をして次の命令，ASK Xに移る．

●直接命令と間接命令

直接命令とは，その命令に出合った時，直接その命令を実行する命令の方法であり，間接命令とは，その命令に出合った時，他の命令の中継を受けて命令を実行する命令の方法である．

岡っ引きが，半七親分に「オマエ，ヤツをつけまわしてショッ引いて来い!!」といわれてスッ飛んで行くのが，直接命令でCIA式に「東京○○町の橋の下のナントカ氏からある人物の写真を受けとり，その写真の人物からつぎの命令を聞け．あとはお前にまかせる!!」という形が間接命令である．

COCOLの直接命令，間接命令も同じ感覚で理解していただければよい．

間接命令の場合は，命令の1行ごとに，ライン番号が付き，プログラムの中でその実行順序を自由にコントロールできる．

直接命令では，ライン番号がなく，命令は書かれた順に実行され，とちゅうで，別の道の仕事をさせることができない．

前の例をあげれば，半七親分から命令を受けた岡っ引きは，指定された人物をつかまえることだけしかできず，つぎの命令は，また親分のところに帰って来なければ実行できない．後半のCIA方式では，写真の人物に「命令を受けた人物が来たらあなたの命令を行ったあと，ホンコンの×××氏に会って命令を受けるように」とことづけておくことによって，殺し屋氏は親分の所へ帰ることなく，命令を実行できる．このCIA方式が間接命令の形であり，普通，プログラムは，ほとんどこの形式で書かれる．COCOLのDOは，この形式のもっとも典型的な例である．この例の中で親分に当るのは，COCOLのプログラムを動かしている．「あなた」自身のことであり，橋の下のナントカ氏や，ホンコンの×××氏は，ライン番号のことである．

◎例題は，プログラム例の項を参照のこと．

3：FOR（フォー）

You must knock the door for three times.（あなたは，ドアを3回ノックせねばならない）のforとまったく同じ使用法である．プログラムの中では，DOやASK等の命令と一体となって使用される．

例

```
1．1  FOR  J＝0，5，50；DO  8．1
```

これは，「Jが0から5ずつ50になるまで8．1にある命令をくりかえしなさい」という命令になる．この形は数表を作る時によく使用される形である．

例えば，角度を10度ずつ0度から50度まで変化させたサイン（sin）の表を作るとすれば，

```
2．1  FOR  J＝0，10，50；DO  4．1
```

と書き，4．1のところに「FSIN（J）をタイプせよ」という命令を書けば良い．

FOR　J＝l，m，n　という時

l：始めの数
m：変化させる量｝となるように定められている．
n：終りの数

だから，同じ仕事をn回させようと思ったら

FOR　K＝1，1，n：DO　Xと書くことになる．

この場合，1つずつ変化させて行く時は中の1（mに当るもの）は略してもよい．（FOR　K＝1，nと書く）

4：GOTO（ゴートゥー）

I will go to the station.（私は駅へ行くでしょう）のgo toとまったく同じ意味で使用される．ある仕事が終ったあと，別のライン番号のところへ行ってまた仕事を始める．前頁の親分と殺し屋氏の例でいえば写真の人物に，親分が「ホンコンの×××氏の所へ行け」という命令を与える，そのことがGOTOの意味である．

例

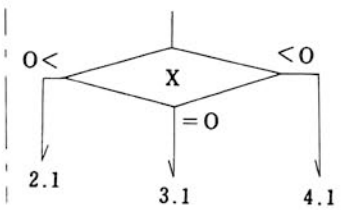
1．1　DO　2．0：GOTO　4．1

2．0の仕事を終えたら4．1へ行け，ということである．

5：IF（イフ）

If your purse empty, you must goto the pownshop.（もしサイフがカラッポになったら質屋へ行け），このIFについては，IFに対する状態が，purse empty（サイフがカラになる）一つであり，その結果は，質屋に行く．一つである．CoCoLの場合，サイフは，マイナスになるか，カラになるか，まだ残っているのか，三つに対して三つの行き場所が一度に指定できる．

```
1・1　IF（X）2.1，3.1，4.1
          X＜0　→　2.1へ行け
          ×＝0　→　3.1へ行け
          X＞0　→　4.1へ行け
```



6：QUIT（クイット）

英語の"Quit!!"（離れろ!!）を略した形であるが，意味はまったく同じで文字通りプログラムをQUITの書いてあるラインで停止させる．

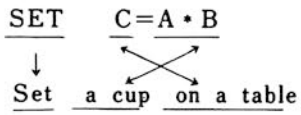
7：SET（セット）

I set a cup on a table.（私は，カップをテーブルの上に置く）のsetとは少しニュアンスは異なるが，"・・・を×××と定義する"という意味を持っている．

例

1．1 ASK A，B

1．2 SET C＝A＊B（Cに対しA＊Bを定義する）これは，Cに対して，AとBの積を定義したことになる．だから，今，Aに5，Bに3を答えたら，Cの内要は，15になっている．（必ず＝の左に対して，右側の結果を定義する．逆は不可）．



英語を例にとれば，カップがAとBの積であり，Cがテーブルである．この形を使用すれば，なり複雑な計算式も，きわめて単純に処理できる．

マニングの公式について，CoCoLを使用して書いてみる．nは固定値で0.02，RとIを任意に力するものとする．

$$V=\frac{1}{n}\ R^{\frac{2}{3}}\ I^{\frac{1}{2}}$$

```
1．1　SET　N＝0.02　　（n＝0.02）
1．2　ASK　R，I　　（R：？　I：？）
1．3　SET　K＝FSQT（I）　　（K＝I^½＝√I）
1．4　SET　M＝FLOG（R）＊2／3　　（M＝2/3 log R）
1．5　SET　H＝FEXP（M）　　（H＝e^M）
1．6　SET　V＝K＊H／N　　（V＝K X H / N）
```

あとは，Vを出力させればよい．

$R^{\frac{2}{3}}$の計算法

$X=R^{\frac{2}{3}}$ とおく

$\log_e X=\frac{2}{3}\log_e R \quad \rightarrow \quad X=e^{(\frac{2}{3}\log R)}$

：TYPE（タイプ）

今までの命令は，計算をさせるための手順に使用する命令であったが，結果は計算機の中にたくわえられていて，人間の目には見えない．これを人間の目に見えるように，タイプライタで文字を打たせる命令が，TYPEである．TYPE命令は，出力するものを指定する部分と，それをどのような形式で出力するかを指定する2つの部分からなっている．7の例の続きを例にする．

```
1．7　TYPE　%8.03，V
```

%8.03　がVの形式であり“全桁数8桁で，小数以下3位でVをタイプせよ”という命令になる．

%A．B　はAが全桁数で，Bが小数以下の桁数である．

TYPE　命令は，数字の他に，英文字による文章を打たせることもできる．

例

```
1．7　TYPE “RYUSOKU”，%8.03，V“M／S”，！
```

Vが2.567であったとすると1．7の命令によって以下のように出力される．

```
RYUSOKU　　2.567M／S↙
                    ↑復帰と改行を行う．
```

“（クォーテーションマーク）でかこまれた文字列は，そのまま出力される．これはASK命令にも共通している．

```
1．2　ASK “KEISHIN”，R，“KOBAI”，I
  ↓
KEISHIN　：＿＿＿＿　KOBAI　：＿＿＿＿　↙
          ↑入力する          ↑入力する
```

注　！記号は，CoCoLにCRLF（復帰，改行）を行わせる命令であり，ASK，TYPE　命令に付随して使用される．その他に，この種の記号として，＃記号（復帰のみ行う）もある．

1－2－2　制御用命令語

この種の命令語は，間接命令からなるプログラムの中には書かれないが，直接命令の形（ライン番号を持たない）で，コンピュータの動き（プログラムの動き）を制御したり，プログラムの修正

をするのに使用される.

1：ERASE（簡略形……E）（イレーズ）

プログラムを消去するのに使用される.

ERASE　ALL……プログラムすべてを消去

ERASE　4.0……4.01～4.99まで，すなわち，4のグループすべてを消去

ERASE　4.10……4.10のラインの命令を消去

2：GO（簡略形……G）（ゴー）

行け！　文字通り，プログラムをスタートさせる．この場合，間接命令で書かれたプログラムを，ライン番号の一番若いものからスタートする.

3：MODIFY（簡略形……M）（モディファイ）

プログラムの書直しに使用する.

MODIFY　1.51　↙　（1.51の行を修正可能にする）

A　　（1.51の行のAの文字まえでタイプライタがとまる）

（修正例は，オペレーションの項を参照すること）

4：WRITE（簡略形……W）（ライト）

プログラムをタイプアウトするのに使用する.

WRITE　ALL…すべてをタイプアウト

WRITE　4.0…4のグループすべてをタイプアウト

WRITE　4.1…4.1の行をタイプアウト

この他に，RETURN，COMMENTの命令があるが，これらは，後部のプログラム例を追いながら説明する.

命令語は両種ともCoCoL内部では頭の1字で理解される．SETをSEP，S，SE，ST，……どのように書いても頭でSであるかぎりSETと解釈される．すべてのコマンドに対してそのようになっているのでTYPEをT，ASKをAと書いてもよい.

1－3　出力形式

TYPE命令によって出力の形式は任意に選べるが，内部に記憶される数値の大きさは，少なからず制限を受ける．正確には，内部に記憶される数値の大きさはほぼ無限といってよいが，その数値を，タイプライタで打ち出せるような形にして記憶しておく場所が，8桁分（有効数字6桁分）しか用意されていないため，有効数字がそれより大になるようなプログラムでは，出力形式に大きな制限を受ける.

出力の形式については，言葉と理論でいいまわすより，例を引いた方が理解しやすいので，例を添えて説明する.

1．整数部の0はスペースに置換えられ，小数部の後方には，0をつめて出力してくる.

SET　A＝12.53；SET　B＝B1

TYPE　A，B

＝␣␣␣　12.5300　＝␣␣␣31.0000

↓

本来00のはずである

このTYPE命令には，出力形式の指定（％A．B）がしてないので，CoCoL内部の初期指定形式によって，整数部4桁，小数部4桁，即ち％8.04の形式で打たれてくる.

TYPE命令を

TYPE ％6.04（全桁数6桁，小数部4桁）

に書きかえると，

＝␣12.5300＝␣31.0000

と出力される．出力形式の指定は，以後改めて指定されるまで最後に指定した形式を保持する．

＝␣12.5300

＋の記号はCOCOLによって自動的に付けられてくるが，現在使用しているCOCOLプログラムは，内部でスペースにおきかわるように直してあるので，＝␣12.5300のように出力されてくる．また，マイナス記号（－）は，＋の入るべき位置に入るので，結果には，「＋のみがスペースに変わり，＋記号は，付いてこない」と考えて良い．

```
SET      A＝12.53     B＝－12.53
TYPE        % 4. 02
 (＋)
␣␣12.53␣ －12.53
TYPE  %10. 03  にすると
␣␣␣␣␣␣␣␣12.530␣－␣␣␣␣␣12.530
```

10桁　　10桁

・・・・出力形式の制限・・・・

TYPE　%A．Bの時，Aは19まで，したがってBは18までである．

```
SET  A＝////. ////
TYPE  %10.05  A
␣␣␣|||||.||||
```

有効6桁

2．6桁以上表示し得るが，有効数字は6桁である．

例で示す通り，Aに定義された数値は，8桁であっても，高位の6桁のみが，有効数字として出力されてくる．

また，%A．BのA（全桁数）は，19桁以下でなければならない．

3．オーバーフロー，これは出力すべき数値が，出力形式より大きくなってしまったことをいうが，この場合，指定式にしたがって，×を出力してくる．

```
SET  A＝2345
SET  B＝A＊10        →Bは23450になっている．
TYPE  %4.00, B
```

××××　　と出力してくる．この場合，MODIFYコマンドを用いて，TYPE%6.00，Bと直せば，

␣␣␣23450と打ってくる．

4．指数形式による出力，科学計算等において，きわめて大きな値や，非常に小さな値を表現するのに，$K\times10^n$という形の指数形式で表現することがある．CoCoLの内部では，数値は，すべてそのような形で記憶され，処理されているので，出力も，そのような形で取り出すことが可能である．$K\times10^n$を例にとれば，Kの値は，一般形と同じ，有効桁数6桁である．

nは±619までとれるので，天文学的数値から極小の数値まで，有効数字6桁の範囲ならば自由自在にとりあつかえる．指数形式は，TYPE　%，A，の形の命令で得られる．

```
例1.{ SET    A＝125
     { TYPE  %, A
       ␣␣0.   125000E␣03 → 0.125×10³
例2.{ SET    A＝625.3874
     { TYPE  %, A
```

␣␣0．625387　E　03

有効数字6桁

3.で説明したようなオーバーフローをなくするために，プログラムを作る時，計算結果を入力値のとりうる最大値を入力して指数形式で求め，その結果に基づいて出力形式を定めるとよい．

例3．
```
SET    A＝123．5 ／10↑10
TYPE   %，A
```
␣␣0．123500 E － 07 →0．1235×10^{-7}

↓

×$\frac{1}{10^7}$

指数形式の場合K×10^nと表わされているが，以上3つの例でわかる通り．

K・・・・0.999999までの数値

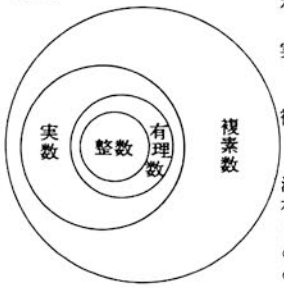
n・・・・−619≦n≦619までの整数値の範囲で表示が可能である．

1−4　浮動小数点処理と整数処理

一般に計算機というものを大別すると，そろばんに代表されるデジタル計算と，計算尺に代表されるアナログ計算機とがある．アナグロ計算機とは，物理的な量を計ることによって計算を行い，デジタル計算機は，整数（0，±1，±2，±3……）計算の組み合わせによって計算を行う．アナログ，デジタルということの本質は，連続か不連続かということである．

測量のデータは，巻尺，トランシット等で測るアナログデータであり，そのままの形では小数以下無限大まで数値がつながっている．そこで切り捨て，四捨五入等の処理を介して，そろばん処理をするためのデジタル変換を行って処理している．一般のコンピュータは，スイッチがONかOFFかの判断を基本とするデジタル計算機である．したがって，加減算と乗算をする時には，そろばんと同様フルカウント（最後まで計算すること）で処理できるが，割算をするときには，必ずあまりが出る．このような処理を整数処理とよび，たとえば2.583は2523という量としてコンピュータ内部でとりあつかわれる．しかし，このような処理はsin，cos，log等の函数計算には，近似計算をしたとしても割算を最後まで割りきってしまう，実数処理計算（アナログ的な処理）を要求される．これをデジタル計算機であるコンピュータにやらせる方法として考えられたのが，指数計算を応用した浮動小数点処理である．この処理は，デジタルな計算処理でアナロジックな量を作るため必ず誤差をともなってくる．CoCoLでは，特殊な函数を自由にとりあつかえるようにするために，内部の計算処理はすべて浮動小数点処理で行われている．だから，タテ，ヨコ，たしてピッタリというような集計処理をともなう計算をすると少なからず誤差がかさみ，計算書の矛盾を生じる．したがって，CoCoLを利用する時には，自分の行おうとしている処理が，整数処理（集計，四捨五入，あまりの出る割算等の処理）をするべきか，実数（浮動小数点）処理（構造計算に代表される科学計算）をするべきかをはっきり認識しなければならない．しかし，内部処理の構造上，入出力値の小さい数値（有効数字6ケタ，出力4ケタ）ならば整数処理が可能なので，ごく部分的な計算処理ならばプログラムを動かしても，なんらさしつかえない．

数の構成



整数……0，±1，＋2，…．
有理数…分数で表わされる数．
実数……$\sqrt{2}$，$\sqrt[3]{5}$ 等分数で表わされない数をふくむ．
複素数…$\sqrt{-2}$ 等もはや初等的には考えられない数をふくむ．
浮動小数点処理というのは有理数と実数の中間に位置している．コンピュータそのものは，整数集合の内部のものである．

1—5　その他のきまり

1．SETコマンドの式，数値定義記号の使用について

SET　A＝2.371
　　↘（この記号のつけ方について）

(1)　1字または2字の英数字で構成されていること
　　M，I，MI，A1，A2，B1，B6，AB，……etc

(2)　最初の文字はF以外の英文字でなければならない．
　　だから，1字の場合，数字と，Fは使用できない．
　　F1，1A，3A，FA，……等はゆるされない．

この他に，アレイという定義の方法がある．

記号　A（J）
　　F以外の英数字またそれよりなる式で構成される．

例えば，FOR　J＝1，10；ASK　A（J）

これは

ASK　A（1）
ASK　A（2）　} を連続して行ったことと同じである．
：　：
ASK　A（10）

注　A（6）とA6とはちがった定義記号としてあつかわれる．

FOR　J＝1，10；TYPE　A（J＋10）
TYPE　A（11）
TYPE　A（12）
TYPE　A（13）　} と同じ意味
：
TYPE　A（20）

注

① FOR　I＝1，10；ASK　A（I）と

② FOR　K＝1，10；ASK　A（K）とは，まったく同じ記号としてあつかわれる．

なぜならば，

① ASK　A（1）
　ASK　A（2）
　：
　ASK　A（10）

② ASK　A（1）
　ASK　A（2）
　：
　ASK　A（10）

であるから，アレイによって分類定義するには，A（J＋10）A（J＋20）等のように，合同法によらなければならない．

合同法

J=1～5	A(J)	1	2	3	4	5
J=1～5	A(J+5)	6	7	8	9	10
J=1～5	A(J+10)	11	12	13	14	15

} このように分類する方法

{ 1と6 / 2と7 / 9と14 / 5と10 } は5を法として合同であるという

2. 二重演算子を使用してはならない．

A＋＋B，　A 1／＋B　　A＊－B

これらを二重演算子という

3. 数式の中に，複数の演算子がふくまれているときは，演算順位の高いものから行なわれ，同位のものは左から右へ行われる．

演算順位
　カッコ内　函数，指数計算，乗算，除算，加，減算

例1　A＋B＊C＋D＝A＋（B＊C）＋D
　　（A＋B）＊C＋DやA＋B＊（C＋D）ではない．

例2　$A / B*C = \frac{A}{B*C}$
　　$\frac{A*C}{B}$ではない．$\frac{A*C}{B}$はA＊C／Bである．

例3　A↑2＊C＝A^{2}＊C
　　A^{2*C}ではない．

もしA^{2*C}を行うならば，A↑（2＊C）と書く．

また，カッコは直列型（A＋（B＋C＊（M＋N））のような時は内部のカッコの中より計算を行い，並列型（A＋B）＊（C＋D）の時は，同等にあつかわれる．

例4　FSIN（A＊B↑2）＊C＝C×sin（A＋B^{2}）
　　sin（A＊B^{2}＊C）ではない．

4. 各コマンドのすぐうしろはスペースでなければならない．

TYPE␣%4.0, A
　　↑このスペース
ASK␣〝A〞，×

5. 同じライン（行）の中を；によって命令を書くことができる．

↘命令を区切る．

1．1　SET　A＝3；SET　B＝A＊2；SET……
　その場合，異なった命令でもかまわない．

1．2　SET　A＝3；ASK　C；TYPE　A＊C

6. 1行は，72文字以内でなければならない．

7. TYPE命令は，出力形式が一定であれば，複数の定義記号の内容をタイプさせることができる．
　TYPE　%6.02, A1, A2, A3, /

（%6.02の形式で，A1，A2，A3の内容を打ってくる）

TYPE　%6.02，A1，%4.03，A2！

のような命令は書けない．

8．ASK命令は1ラインで複数の数値を聞くことができる．

ＡＳＫ　Ａ，Ｂ，Ｃ，Ｄ，！

（Ａ，Ｂ，Ｃ，Ｄの内容を問うてくる）

（SET命令にはこのような使用法は適用できない．）

9．ライン番号のふり方

間接命令でプログラミングをする時，各行（ライン）にライン番号を付ける．

例

```
1. 01　ＡＳＫ　Ａ
1. 02　ＡＳＫ　Ｂ
1. 10　ＳＥＴ　Ｃ1＝Ａ＋Ｂ
1. 20　ＳＥＴ　Ｃ2＝Ａ＊Ｂ
1. 30　ＤＯ　4．0
1. 40　ＤＯ　5．0
1. 50　ＱＵＩＴ
4. 10　ＴＹＰＥ　%6.04，Ｃ1，！
5. 10　ＴＹＰＥ　%6.04，Ｃ2，！
```

↓

Ａ．Ｂのような形で番号をつける．

Ａは，１～15まで｝　任意のライン番号を

Ｂは，01～99まで｝　付けることができる．

プログラム内部で分類されて実行されるのは，Ａの番号による分類だけである．

Ａ．０という形は，命令の中で使用され，ライン番号の頭がＡであるグループ，即ち，1.0であるならば，1.01～1.99まで全部をさす．

2．プログラミング

この章では前章で説明したコマンドを実際に使用して，プログラムを組みながら，論理解析（アルゴリズムの分析）の方法を学んでゆく．

2－1　基本的な数学公式のプログラム

1．三平方（ピタゴラス）の定理－1

☆　プログラムの仕様

直角三角形の底辺と高さを入力し，斜辺の長さを計算する．

底辺……最高　全桁数4桁　小数以下2桁

高さ……　〃

公式　$A^2=B^2+C^2$　（Ｂ，Ｃ　底辺，高さ）

出力形式

ＴＥＩＨＥＮ＝：99．99（ＣＰＵが聞く）

ＴＡＫＡＳＡ＝：99．99（　〃　）

＊＊＊＊ＳＨＡＨＥＮ＊＊＊＊999.99

斜辺については，有効数字をそろえるために，小数以下2位とする．全桁数は，5桁以上になりえないので，斜辺の出力形式は，%5.02でよい．

プログラム

```
1．1．ASK〝TEIHEN＝〃，B，！
1．2　ASK〝TAKASA＝〃，C，！
1．3　SET　A＝FSQT（B∧2＋C∧2）→（√(B²＋C²)）
1．4　TYPE〃　　****SHAHEN****〃%5.02，A，！
1．5　ＱＵＩＴ
```

プログラム

```
COCOL 1971

01.10 ASK "TEIHEN=", B, !
01.20 ASK "TAKASA=", C, !
01.30 SET A=FSQT(B^2 +C^2)
01.40 TYPE "    ****SHAHEN****", %5.02, A, !
01.50 QUIT
```

プログラムを動かした結果

```
•GO          ┌ ここでとまる
TEIHEN=:3.00 ┐
TAKASA=:4.00 ┘ 打ち込む
   ****SHAHEN****        5.00
•
```

2．ピタゴラスの定理―2

☆　プログラム仕様

直角三角形の斜辺と底辺を入力し，高さを計算してから，その三角形の面積を計算する．

(1)　入力

斜辺	全桁数	4．	小数以下	2
底辺	〃	4．	〃	2

(2)　出力

高さ	全桁数	4．	小数以下	2
面積	〃	6．	〃	2

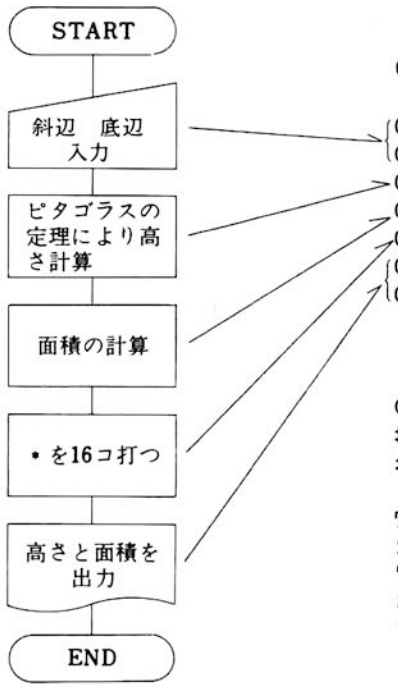
出力形式

```
＃ＳＨＡＨＥＮ……99．99
＃ＴＥＩＨＥＮ……99．99
****************
ＴＡＫＡＳＡ＝99．99
$␣␣ＭＥＮＳＥＫＩ***9999．99
```

公式　$H=\sqrt{L^2+T^2}$　（Ｌ…斜，Ｔ…底）

$S=L\times H\div 2$

流れ図(フローチャート)

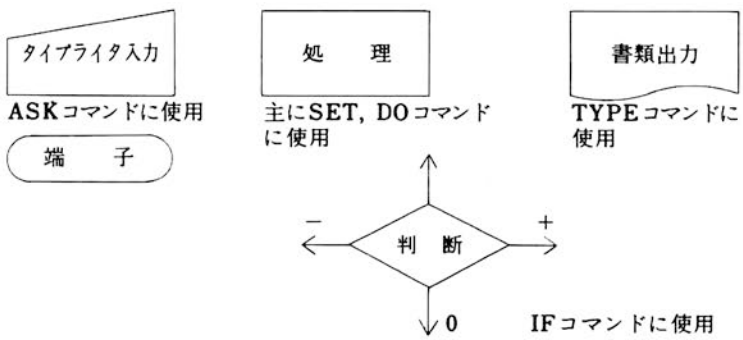


プログラム

```
COCOL 1971
01.10 ASK  "#SHAHEN...", L, !
01.20 ASK  "#TEIHEN...", T, !
01.30 SET  H=FSQT(L^2-T^2)
01.40 SET  S=T*H/2
01.50 FOR  J=1,16;TYPE  "*"
01.60 TYPE !, "TAKASA=", %4.02, H, !
01.70 TYPE  "$   MENSEKI***", %6.02, S, !
*

GO
#SHAHEN...:5.00
#TEIHEN...:4.00
****************
TAKASA=    3.00
$   MENSEKI***        6.00
TAKASA=3.00
$ MENSEKI*** 6.00
*
```

1.50の命令は，TYPE "*" の
命令を16回繰り返せということ

フローチャートの見方



3．曲線設置計算

☆　プログラム仕様

道路中心測量（測角）の成果のうち，各変点における，方向角より交角を計算し，任意のRを入力することによって，切線長（T,L），曲線長（C,L），正割長（S,L）を計算する．

(1)　入力

IP（交点）番号：番号であり，計算数値ではない．

方向角　　：度，分で入力

R（半径）　　：全桁数，6位，小数以下3位までとする．

(2)　出力

切線長：小数以下2位

曲線長：　〃　2位

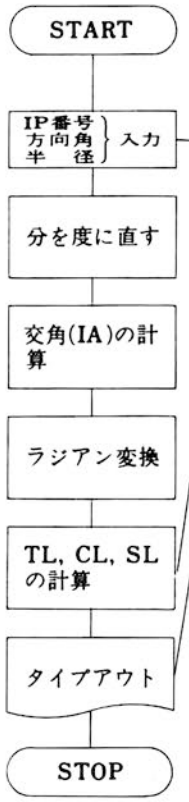
正割長：　〃　2位

交角の計算をして出力させるためには，IF文による分析計算が必要なため，この節では行わない．（2－3，3節を見よ）

(3)　出力形式

```
***ＩＰ　９９９***
Ａ＝９９９ＤＯ９９ＦＵＮ
Ｒ＝９９９．９９９Ｍ
ＴＬ＝９９９．９９␣␣ＣＬ＝９９９．９９␣␣ＳＬ＝９９９．９９
```

フローチャート　　プログラム




```
COCOL 1971

01.10 ASK  "***IP ", B, "***", !!
01.20 ASK  "A=", A1, "DO", A2, "FUN", !
01.30 ASK  "R=", R, "M", !!
01.40 SET  IA=FABS(180-(A1+A2/60))*3.14159/180
01.50 SET  T=R*FSIN(IA/2)/FCOS(IA/2)
01.60 SET  C=R*IA
01.70 SET  S=R*(1/FCOS(IA/2)-1)
01.80 TYPE %5.02, "  TL=", T, " CL =",C," SL=", S, !!!
*
*       実行結果
*
*GO
***IP  :15  ***

A=:147  DO:35  FUN
R=:40  M

   TL=    11.63   CL=   22.63   SL=    1.66


*GO
***IP  :16  ***

A=:236  DO:45  FUN
R=:20  M

   TL=    10.80   CL=    19.81   SL=    2.73
```

TL=R*tan IA/2　（IAは，度のラジアン）$\tan X = \frac{\sin Y}{\cos X}$

CL=R*IA

SL=R*(Sec IA/2−1)　$\mathrm{Sec}\, X = \frac{1}{\cos X}$

ラジアン変換

IA=｛度+(分)/60｝×π/180

π=3.14159

1.40　IA=FABS（180……

のFARSはカッコ内の数値の絶対値をとる．これは，方向角が180以上であっても以下であっても両方にこの式を使用できることを意味している．

2—2　作表プログラム

数表を作るということは，入力データを一定規則に従って変化させながら，まったく同じ仕事をし，その結果を規則正しく並べて書き込むことである．この節では，DO，FOR，コマンドを使用して，作業プログラムの解説をする．



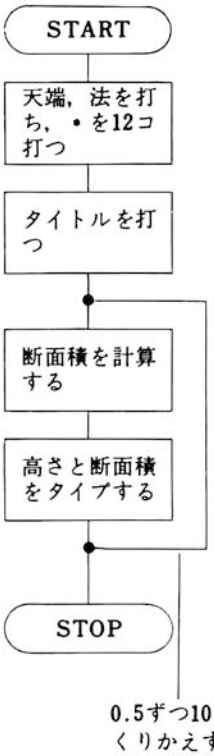
左の図のようなコンクリートウォールの断面積を，高さを 0.5m から 5.0m まで 0.5m ずつ変化させながら表にする．

1. 固定データ　　天端長　0.30m
　　法勾配　1：0.2
2. 変化データ　　壁　高　H（0.5≦H≦5.0）
3. 出力形式

```
TENBA     0.30
NORI      0.20
************
TAKASA     DANMEN
  0.50     99.999
  1.00        〃
  1.50        〃
   )          )
   )          )
  5.00     99.999
```

4. 出力データ

H	全桁数 3 桁	小数以下 2 位
断面積	〃 5 桁	〃 3 位

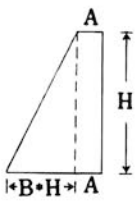



```
COCOL 1971

01.10 TYPE  "TENBA...0.30", !
01.20 TYPE  "NORI....1:0.20", !
01.30 FOR  I=1,14;TYPE  "•"
01.40 TYPE  !!!, "TAKASA    DANMEN", !
01.50 FOR  H=0.5, 0.5, 5.0;DO  2.0
01.60 QUIT

02.10 SET  S=(0.2•H+2•0.30)•H/2
02.20 TYPE  %3.02, H
02.30 TYPE  %5.03, S, !
•
•
•
•
•GO
TENBA...0.30
NORI....1:0.20
••••••••••••••

TAKASA    DANMEN
   0.50    0.175
   1.00    0.400
   1.50    0.675
   2.00    1.000
   2.50    1.375
   3.00    1.800
   3.50    2.275
   4.00    2.800
   4.50    3.375
   5.00    4.000
•
```

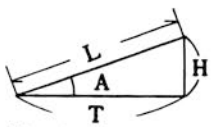
1. 断面積の計算

天端長　A　(0. 30)
法勾配　B　(0. 20)　} とする.
高さ　H

$S=(B+H+2A)\cdot H/2$



2．斜距離改算表を作る



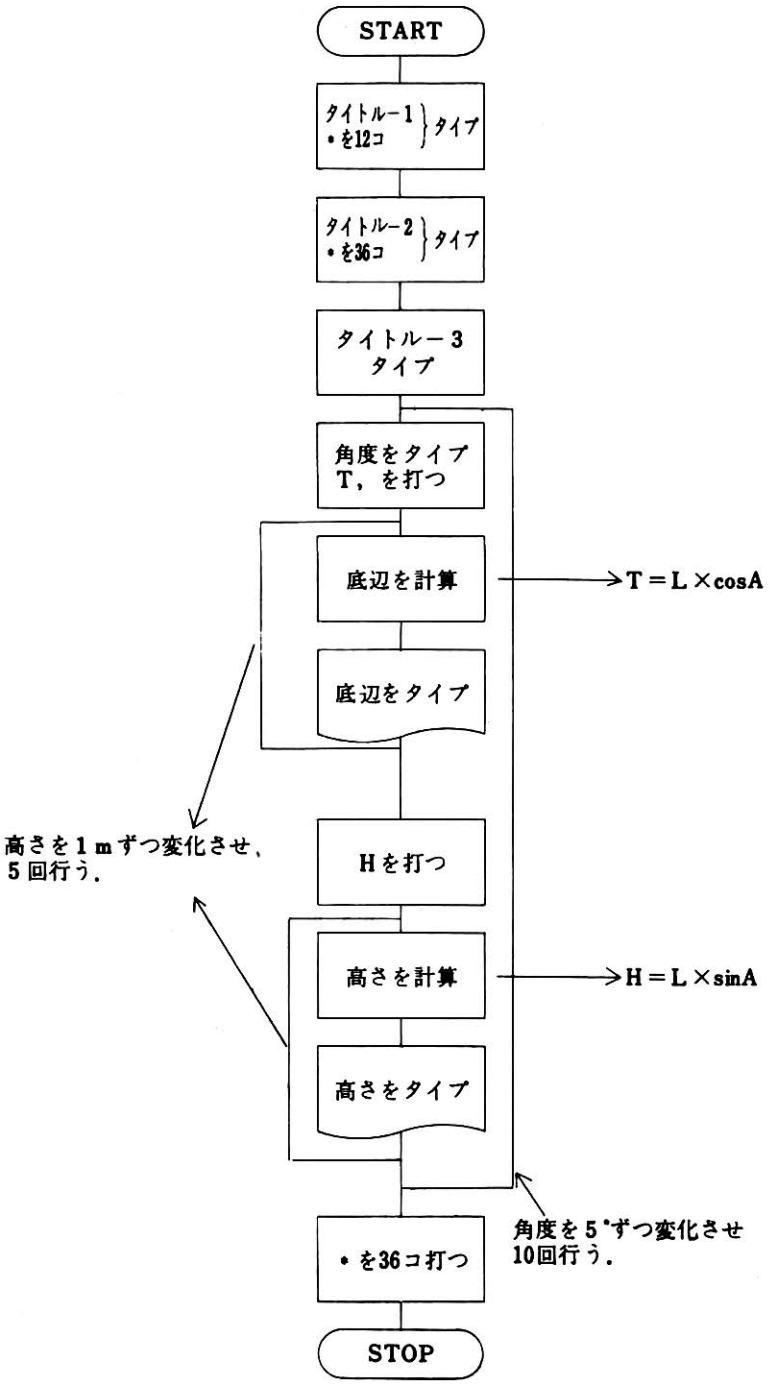
斜距離Lと角度Aを変化させながら
水平距離T　高さHを計算して表にする．

(1) 変化データ　　斜距離　L　（1≦L≦5）　1mずつ
角度　A　（5°≦A≦50°）　5°ずつ

(2) 出力形式

```
SHAKYORI KAISANHYO
••••••••••••••••••••

SHAKYORI••1M     2M     3M     4M     5M
••••••••••••••••••••••••••••••••••••••••
KAKU
 5DO T  9.99   9.99   9.99   9.99   9.99
     H  9.99   9.99   9.99   9.99   9.99
10DO T    ∫      ∫      ∫      ∫      ∫
     H
  ∫
50DO T
     H
••••••••••••••••••••••••••••••••••••••••
```

(3) 出力データ
水平距離　T　　全桁数　3桁　小数以下　2位　　T＝L× cos A
高さ　　H　　H＝L× sin A


START
タイトル－1 ・を12コ タイプ
タイトル－2 ・を36コ タイプ
タイトル－3 タイプ
角度をタイプ T，を打つ
底辺を計算
T＝L×cosA
底辺をタイプ
高さを1mずつ変化させ、5回行う．
Hを打つ
高さを計算
H＝L×sinA
高さをタイプ
角度を5°ずつ変化させ10回行う．
・を36コ打つ
STOP

```
COCOO 1971

01.10 TYPE "SHAKYOKI KAISANKYO",!
01.20 FOR I=1,16;LO 3.1
01.30 TYPE !!!"SHAKYOKI**1%     2%     3%     4%     5%",!
01.40 FOR I=1,36;LO 3.1
01.50 TYPE !,"KAKO",!
01.60 FOR J=5,5,50;LO 2.0
01.70 FOR I=1,36;LO 3.1
01.80 TYPE !!!!!!;COLT

02.05 SET AN=[illegible]*3.1/159/180
02.10 TYPE X2.00,2," T"
02.20 FOR C=1,5;LO 4.0
02.30 TYPE !,"      H"
02.40 FOR L=1,5;LO 3.0
02.50 TYPE !

03.10 TYPE "*"

04.20 SET I=L*FCO (AN)
04.30 TYPE X3.02,T

05.10 SET H=L*FCIN(AN)
05.20 TYPE X3.02,H
*GO
SHAKYOKI KAISANKYO
*******************


SHAKYOKI**1%     2%     3%     4%     5%
*****************************************
KAKO
=  5 T= 1.00= 1.[illegible]= 2.99= 3.99= 4.[illegible]
     H= 0.0[illegible]= 0.1[illegible]= 0.[illegible]= 0.35= 0.[illegible]
= 10 T= 0.99= 1.97= [illegible]= 3.97= 4.93
     H= 0.17= 0.35= 0.52= 0.70= 0.87
= 15 T= 0.97= 1.93= 2.90= 3.86= 4.83
     H= 0.26= 0.52= 0.78= 1.04= 1.30
= 20 T= 0.94= 1.[illegible]= [illegible]= [illegible].76= 4.7[illegible]
     H= 0.34= 0.69= 1.03= 1.37= 1.71
= 25 T= 0.91= 1.81= [illegible].72= 3.63= 4.53
     H= 0.42= [illegible].85= 1.[illegible]7= 1.69= 2.11
= 30 T= 0.87= 1.73= [illegible]= 3.47= 4.33
     H= 0.50= 1.00= 1.50= 2.00= 2.50
= 35 T= 0.[illegible]= 1.64= 2.46= 3.28= 4.10
     H= 0.57= 1.1[illegible]= 1.72= [illegible].30= [illegible].87
= 40 T= 0.77= 1.53= [illegible].3[illegible]= 3.07= 3.83
     H= 0.6[illegible]= 1.[illegible]= 1.93= 2.57= 3.21
= 45 T= 0.71= 1.42= 2.1[illegible]= 2.83= 3.54
     H= 0.71= 1.42= 2.12= 2.83= 3.54
= 50 T= 0.64= 1.[illegible]= 1.93= 2.57= 3.21
     H= 0.77= 1.53= 2.30= 3.07= 3.83
*****************************************
```

"2のグループをすべて実行せよ！" の意味

2—3　分析（IF文）をふくむプログラム

1.　算数テストのプログラム

☆プログラム仕様

2ケタと1ケタの数のかけ算の問題をコンピュータが出しその解答をタイプライタで打ち込むと，正誤を分析し，メッセージをタイプする．誤りの時は，正解が出るまでやり直す．2ケタと1ケタの数は任意の数であるから，乱数発生関数を使用する．

乱数発生函数

SET A＝FRAN（　）

この命令に出合うとAに　－1＜A＜＋1の任意の数が定義される．だから2ケタの任意の数を作るのであるならば，Aを100倍してその整数値をとればよい．

2ケタの整数乱数発生

SET A1＝FRAN（　）

SET A2＝FITR(100×A1)（FITRは小数以下を切りすてる）

A1に　0.675935……が定義されたとすると

0.675935…×100＝67.5935……

↓FITRでこの部分だけとり出す．

A2＝67になっている．

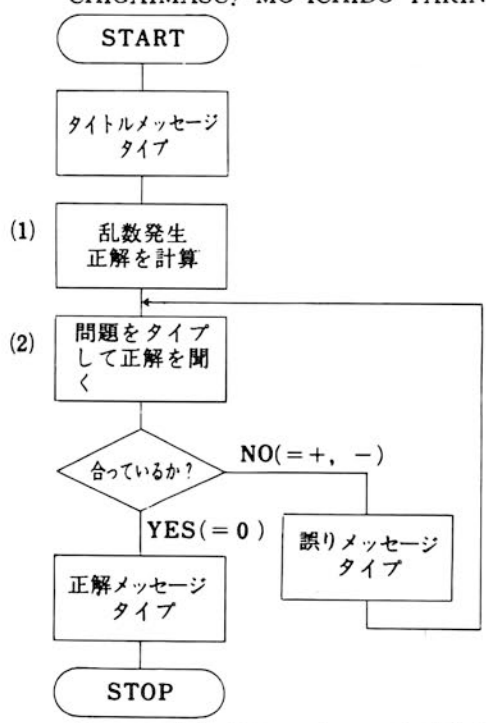
メッセージ

正解の時　よくできました．

YOKU DEKIMASHITA!

誤りの時　ちがいます，もう一度やりなおし！

CHIGAIMASU! MO-ICHIDO YARINAOSHI!



タイトールメッセージ

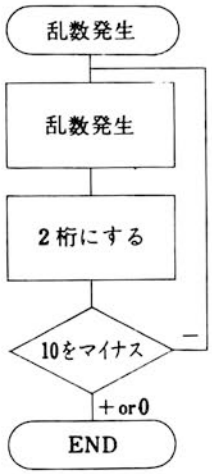
かけ算のべんきょうをしましょう．

問題の形

99×9＝

この中で(1)を洪水流量，(2)をパイプの設置条件にすればマニングの公式を使用した設計計算プログラムが作れる．これは研究課題としておく．

2. 1.のプログラムのうちで，乱数を発生するとき，2桁の乱数は，必ず2桁になっているとはかぎらない．なぜならA＝FRAN（　）でA＝0.0175……になっていれば，乱数は，1桁になってしまう．それを必ず2桁になるようにする．




```
4．1　SET　A1＝FRAN(－)
4．2　SET　A2＝FITR(A1＊100)
4．3　IF　(A2－10)4.1，4：4，4.4
4．4　RETURN
```

A2は2桁になっている．

COCOL 1971

```
01.10 TYPE !!!!!!,"KAKEZAN NO BENKYO O"
01.20 TYPE " SHIMASHO !!",!!!
01.30 DO 4.0
01.40 DO 5.0
01.50 SET C=A*B
01.60 TYPE "MONDAI****",%2.00,A," X",B," ="
01.70 ASK AN,!
01.80 IF (AN-C)2.1,1.9,2.1
01.90 TYPE !!!,"YOKU DEKIMASHITA",!!!
01.95 QUIT

02.10 TYPE !!!,"CHIGAIMASU!!!!!",!
02.20 TYPE "MO-ICHIDO YARINAOSHI!",!!!
02.30 GOTO 1.6

04.10 SET A1=FRAN( )
04.20 SET A=FITR(100*A1)
04.30 IF (A-10)4.1,4.4,4.4
04.40 RETURN

05.10 SET B1=FRAN( )
05.20 SET B=FITR(10*B1)
05.30 IF (B)5.4,5.1,5.4
05.40 RETURN
*GO
```

2桁の乱数発生（正の数のみ）

0でない1桁の乱数発生

5.1にすると正の数のみになる．現在は正・負両方の乱数を発生する．

```
KAKEZAN NO BENKYO O SHIMASHO !!

MONDAI****  25 X  1 =:35

CHIGAIMASU!!!!!
MO-ICHIDO YARINAOSHI!

MONDAI****  25 X  1=:25

YOKU DEKIMASHITA
```

3. 角度の四則演算

角度と角度の加算や引算をする時，度，分，秒の60進法にからんで整数処理系の割算（商とあまりのある割算）が必要になってくる．しかしCoCoLでは内部の演算機能，浮動小数点方式（割算の時あまりを出さずに，最後まで割ってしまう）を使用しているためA/Bの形で割算すると度分秒の表示が不能になってしまう．

例．A＝20°45′

B＝31°56′

このA，Bの和を計算する．

度の部分　　20＋31＝51

分の部分　　45＋56＝101

102÷60＝1あまり41

∴　51°＋1°＝52°　52°41′

または，　20°×60′＋45′＝1245′

31°×60′＋56′＝1916′

1245＋1916＝3161

3161÷60＝52あまり41

52°41′

CoCoLで計算すると

```
SET   A＝20×60＋45
SET   B＝31×60＋56        結果は，52.6833
SET   C＝（A＋B）/60      小数以下については0.6833×00°＝
TYPE  %8.04, C            可能であるが近似計算になってしまう．
```

そこで，角度計算を行うのに引算を基本とした割算，即ちそろばんで行うような方式の割算プログラムを組まねばならない．

31を割られる，数4を割る数としてこの割算を説明する．

1）　A＝31－4＝27　＞0　C＝1（＋1）

2）　A＝27－4＝23　＞0　C＝2（＋1）

3）　A＝23－4＝19　＞0　C＝3（＋1）

4）　A＝19－4＝15　＞0　C＝4（＋1）

5）　A＝15－4＝11　＞0　C＝5（＋1）

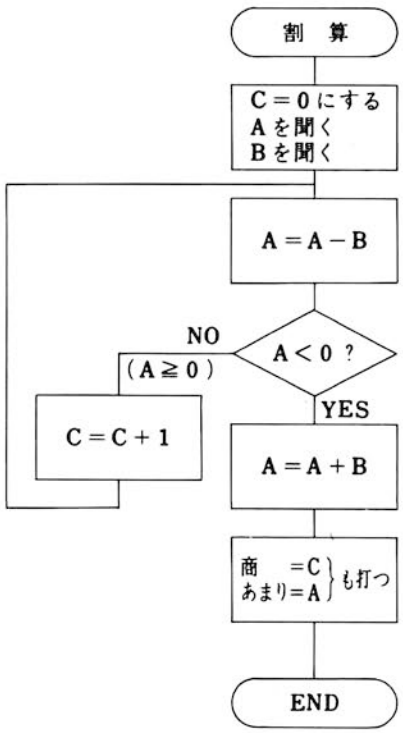
6）　A＝11－4＝7　＞0　C＝6（＋1）

7）　A＝7－4＝3　＞0　C＝　（＋1）

8）　A＝3－4＝－1　＜0　STOP

C＝7……商　A＋.4＝3……あまり

元々，コンピュータ内部のかけ算，割算は，これとまったく同じパターンで行われているのである．この割算の内容をフローチャートに書いてプログラミングをしてみよう．



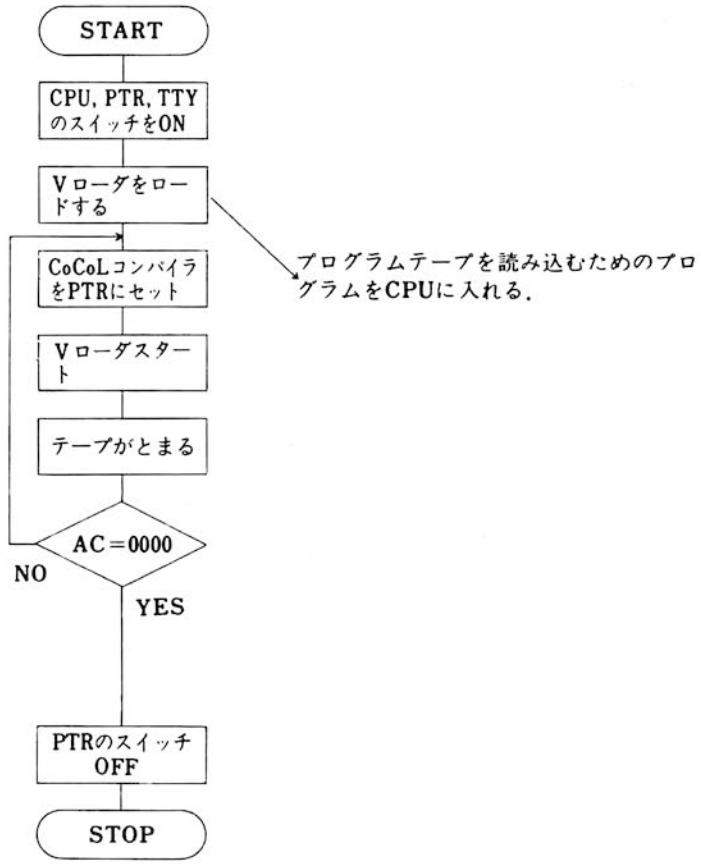
A割られる数
B割る数
} 両方とも整数値でなければならない.

```
COCOL 1971

01.10 SET C=0
01.20 ASK A,"/",B,!
01.30 SET A=A-B
01.40 IF (A)1.80,1.70,1.70
01.50 TYPE %6.00,"SHO",C,"  :AMARI",A,!
01.60 QUIT
01.70 SET C=C+1
01.71 GOTO 1.3
01.80 SET A=A+B
01.81 GOTO 1.5
*
*
*GO
:237 /:45
SHO        5    :AMARI      12
GO
:43 /:8
SHO        5    :AMARI       3
*5\GO
:50 /:5
SHO       10    :AMARI       0
*
```

3．オペレーション

3－1　CoCoLの読み込み

ココルプログラムを使用するには，ココルコンパイラをCPUにロード（読み込み）しなければならない．その手順は，CEC—555の一般のプログラムと同様である．下の流れ図はその手順を示す．



注　1）ココルコンパイラ

われわれは，ココル言語を使用してプログラムを組む．しかし，コンピュータは，直接そのプログラムを理解できないので，それを翻訳せねばならない．その翻訳するためのプログラムのことをいう．

3－2　スタートと初期対話

ココルコンパイラのスタートは，0200番地である．スタートさせると，TTY（タイプライタ）に初期対話の質問が打たれてくる．質問は，3つあり，それぞれにYES，NOをタイプすることによってこたえる．

```
CONGRATULATIONS!!
YOU HAVE SUCCESSFULLY LOADED 'COCOL 1971' ON A CEC-555 COMPUTER.

SHALL I RETAIN LOG, EXP, ATN ?:YYS

PLEASE ANSWER 'YES' OR 'NO' :NO

SHALL I RETAIN SINE, COSINE ?:YES

PROCEED.

*
```

特殊函数が必要な場合は，YES，不必要な場合はNOと答える．NOと答えた時には，特殊函数の分だけよけいにプログラムを入力することができる．

初期対話の内容は書きかえることができないので，質問の答は，慎重に行なわねばならない．多種類のプログラムをとりまぜて処理を行う時には，特殊函数を使用するものと，しないものに分けて処理することが望ましい．しかし，プログラムがそれほど大きくない場合（ライン数にして30行ぐらい）には，YESと答えておいて，混合して使用しても良い．

3－3　修正

1.　誤った文字を打った時

プログラムをタイプインしている時に，文字や記号を誤った場合「抹消」のキーを打つと，¥記号が打たれて，すぐ左の1文字を消去することができる．続けて「抹消」を打った時は，すぐ左から，¥記号の個数だけ消去したことになる．また「←」のキーを打つと，その行の左側をすべて消去する．

例(1)

```
1.1  SET  AZ=B*K¥H
```

Kを消去したことになる

したがって1.1は，SET　AZ=B*H　になっている

例(2)

```
1.1  SET  AZ=B*K¥¥¥A/Z
```

このように消去したことになる

したがって

```
1.1  SET  AZ=A/Z
```

になっている

例(3)

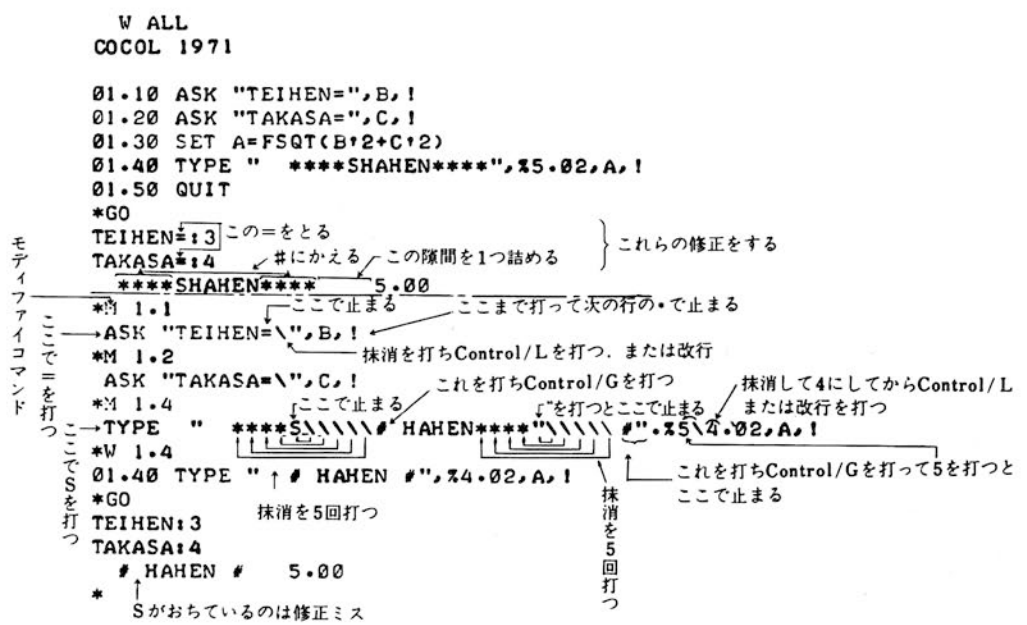
```
1.3  TYPE   %8. 2-
```

すべてを消去したことになる

2.　ラインの消去

ERASEコマンドを使用する．使用法，性質はERASEの項を見ること．

3. MODIJY（モディファイ）コマンドの使用法



☆Sを挿入するには，どのようなオペレーションをしたらよいか？

Cotrol/L，ControL/Gについて，

タイプライタのコントロールキーを押しながらL，またはGを打つこと．ControL/Gを打つとタイプライターのチャイムが鳴る．

このコマンドは，指定した行のいくつかの文字を修正，あるいは追加するのに，他の文字に影響なく簡単に操作できるように作られている．

修正したい行は，ライン番号で指定する．

M(MODIFY)␣ A.'B）←復帰

↑ ↖ライン番号

必ずブランクを1つあけること

ここまで打つと，タイプライタは，復帰，改行を行ってストップする．そのあと，検索するべき文字のキーを打つと，その文字がみつかるまで，その行の内容をタイプし，その文字でとまる．その時点で，われわれは，7種類の操作が可能になる．

(1) すでにタイプアウトされた文字の次に，新しい文字を追加できる．

(2) Control/Lを打つと，検索文字（はじめに指定した文字）がまた出てくるまでその行の内容をタイプしてくる．このとき，その文字がない場合は，最終字までタイプされてきて操作は自動的に終る．

(3) Control/Gを打つと，検索文字を変更できる．以後は始めと同じ．

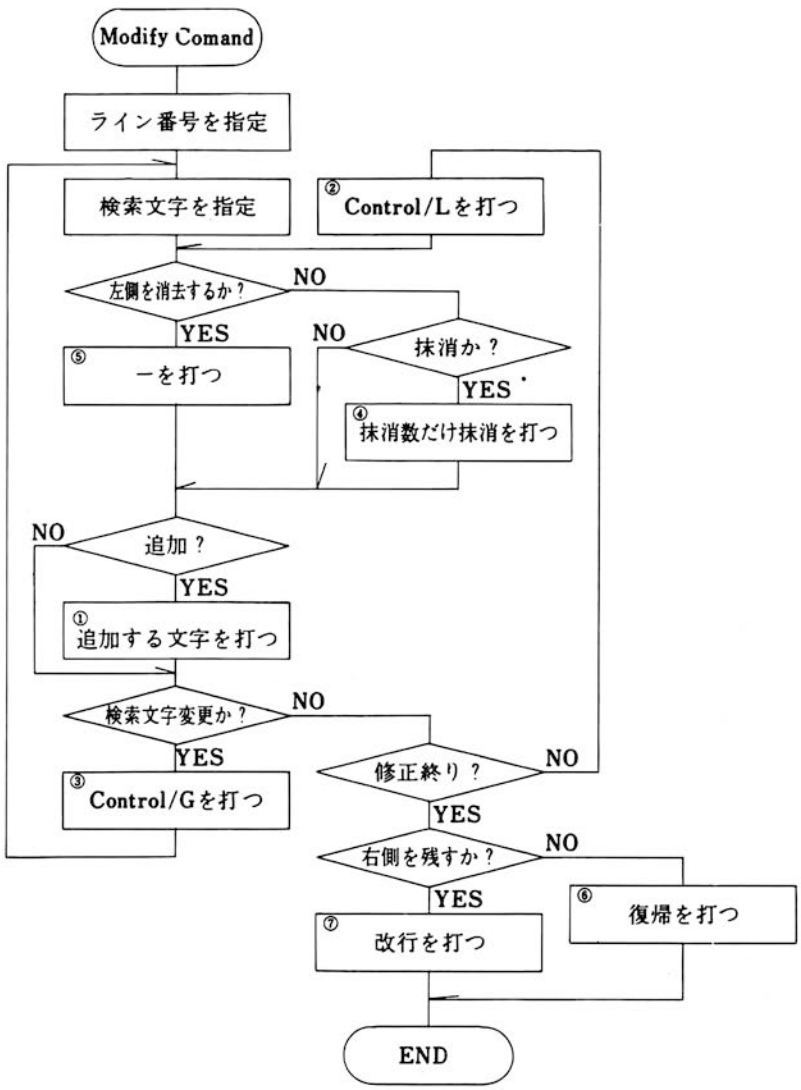
(4) 抹消を打つごとに，左の方へ1字ずつ消える．そのとき1字につき（スペースもふくむ）¥記号を1つタイプしてくる．

(5) ー（マイナスではない）を打つと，行の左側が全部消える．

(6) 復帰を打つと，それを打った時点の行の右側の残りはすべて抹消される．

(7) 改行を打つと，行の残りの部分は保存される．（タイプアウトされてくる）

MODIFYコマンド操作の流れ図



4. ASKコマンドに対する数値入力とその修正

ASKコマンドに対し，数値を入力している時，まちがった数値を入力することが多々ある．その数値は，抹消キーでは消去することができない．

抹消の方法は二通りある．数値に対するターミネータ（データのくぎり，注参照）を打ったあとでは，処理が先に進んでしまっているのでControl/Cのキーを打って，プログラムを最初からやり直す．もう一つは，ターミネータを打つ前に←キーを打つことによって，前に打った数値を0にできる．その場合，正しい数値は，つづけて打てばよい．

注 ASKコマンドの数値入力に対するターミネータ

1．1 ASK “TEIHEN”，A

この命令を実行すると，タイプライタは下のように打つ．

ＴＥＩＨＥＮ：　ここから後に数値を入力した時，数値の切れめを示さなければならない．

ＴＥＩＨＥＮ：39，14　この後にはまだ続くのか終りかを示さなければならない．

これを示す記号のことをターミネータと呼ぶ．

CoCoLでは，英数字，・，¥，以外のキーはほとんどターミネータとして受け入れるが，ふつう，．；スペース，復帰等が使用される．入力データの見やすさ，処理の能率を考えるとスペースが最も良く使用される．

```
 ?00.00
*W ALL
COCOL 1971

01.10 ASK "SU 1",A
01.20 ASK "*** SU 2",B,!
01.30 SET C=A*B
01.40 TYPE %6.04,"KOTAE    ",C,!
*GO
SU 1:25 *** SU 2:4
KOTAE       100.000
*GO
SU 1:35-*** SU 2:4
KOTAE       140.000
*GO
SU 1:35-25 *** SU 2:4
KOTAE       100.000
*GO
SU 1:35 *** SU 2:4

KOTAE      ?01.00
*GO
SU 1:25 *** SU 2:4
KOTAE       100.000
*GO
SU 1:32.56-32.54 *** SU 2:0.045
KOTAE         1.4643
*GO
SU 1:45.67 *** SU ?01.00
*GO
SU 1:35.67 *** SU 2:0.2
KOTAE         7.1340
*
```

－をマイナス(－)とまちがえて打ったため，ターミネータとして扱われ，35×4＝140の計算を行ってしまった

この数値を－によって消去し，新たに25を入力した

数値を入力し終ってからミスに気がつき，途中でControl/Cを打った

打ち直したもの

スペース以外のターミネータ

前の数値（SU1）にミスをみつけControl/Cを打った

このへん

やり直した分

5. エラー診断

利用者がココルプログラムを組んで，それを実行させた時，誤った命令の仕方に出あうと，その誤りがどんな種類のものかを判断して修正するように要求してくる．その時，直接プログラムであると，最初からプログラムを打ち直しすることになるが間接プログラムの場合は，誤りに出合ったところで，エラー診断コードを行ってくるので，そのコード表（別表）を見て，モディファイコマンドを使用して直せば良い．

例

```
 W ALL
COCOL 1971

01.10 ASK "SU 1",A
01.20 ASK "*** SU 2",B,!
01.30 DET C=A*B
01.40 TYPE %6.04,"KOTAE    ",V,!
* GO
SU 1:25 *** SU 2:4
?02.61
*M 1.3
 D\SET C=A*B
*GO
SU 1:25 *** SU 2:4
KOTAE       0.0000
*M 1.4
 TYPE %6.04,"KOTAE     ",V\C,!
*GO
SU 1:25 *** SU 2:4
KOTAE      100.000
*W ALL
C-TAIWA-KEISAN

01.10 ASK "SU 1",A
01.20 ASK "*** SU 2",B,!
01.30 SET C=A*B
01.40 TIPE %6.04,"KOTAE    ",C,!
*
```

1.20まで実行してきて，1.30のDETにぶつかってこの診断を下した．この場合，CoCoLでは命令語の頭の1字で命令の種類を分類しているのでDOコマンドとまちがえて解釈されている．

正しい数を入力したにもかかわらず答がおかしい．Vという記号は数値がなにも定義されていないので0を打ってきた．

注

エラー診断

コード	意　味
?００．００	コンソールから手動スタート
?０１．００	キーボードからCTRL/Cを介しての割込み
?０１．４０	規則違反のライン番号フォーマット
?０１．７８	グループ数あるいはリテラルが大きすぎる．
?０１．９６	ライン番号にピリオドが2つ以上ある．
?０１．：５	ライン番号が大きすぎる．
?０１．：４	ライン番号に0がある．
?０２．３２	存在しないグループをDOにより参照した．
?０２．５２	存在しないライン番号をDOにより参照した．
?０２．７９	変数ストレージを超過した．
?０３．０５	存在しないライン番号をGOTOまたはIFにより参照した．
?０３．２８	不正コマンド使用
?０４．３９	FORまたはSETコマンド中の"＝"の左側に不正がある．
?０４．５２	"＝"の右側に不正がある．
?０４．６０	FORコマンドの中に不正がある．

?０４．：３　　DISPLAYコマンドの中に不正がある．
?０５．４８　　MODIFYコマンドの中に不正がある．
?０６．０６　　関数または数字の不正使用がある．
?０６．５４　　変数の数が過大である．
?０７．２２　　数字の表現が不正である．
?０７．３８　　カッコの前に演算子がない．
?０７．：９　　関数の中に引数がない．
?０７．：６　　２重演算子の使用，または存在しない関数を使用した．
?０８．４７　　カッコが足りない．
?０９．１１　　ERASEコマンドの中に不正がある．
?１０．：５　　コマンドの数が過大である．
?１１．３５　　入力する数値が過大である．
?２０．３４　　０の対数を要求した．
?２３．３６　　変数名が長すぎる．
?２６．９９　　数値が大きすぎる．
?２８．７３　　０による除算を要求した．
?３０．０５　　負の平方根を要求した．
?３１．＜７　　不正文字が含まれている．不正コマンドの使用
　　不正関数の使用

6．プログラムテープの保存

間接プログラムは，WRITEコマンドを使用して，プログラムをタイプライタで打ち出すことができる．この時，TTYのさん孔キーをONにしておけば紙テープとして保存できる．また，次にそのプログラムを使用する時には，TTYのリーダから読り込ませて使用することができる．

テープの取り方



WRITE ALL と打っても良い.

7. Control/Cの使用について

プログラムを実行している時は，TTYからCoCoLを直接コントロールすることができない．プログラム実行中に，何かミスを見つけたり，とちゅうでストップさせたりするのには，コンピュータの動きそのものをストップさせることもできるが，内部の記憶を破壊してしまうことがあるので，これはさけた方がよい．そのような時には，Control/Cのキーを打つと．＊?01．00と打たれてくる．そのあとならば，TTYから，CoCoLを自由にコントロールすることができる．

記号と意味

記号	よ び 方	意 味
%	パーセント	出力形式を示す
"	コーテーションマーク	文字列の区切り
!	エックスクラメーション	復帰，改行を行う
#	ナンバーサイン	復帰のみを行う
,	カンマ	式の区切り
;	セミコロン	命令文の区切り
:	コロン	ASK命令に出合った時の入力OKのサイン
$	ドルマーク	シンボルテーブルの内容をタイプする
*	アスタリスク	①式の中ではかけ算を意味する ②行のはじめでは，CoCoLのコントロールが使用者にもどされたこと
／	スラッシュ	わり算の記号
^	アップアロー	指数演算記号
.	ピリオド	小数点，ライン番号の区切り，出力形式の区切り

❼ ベーシック

1．概　要

MARK Ⅱ BASICはアメリカのダートマス大学で開発されたBASICを基礎として開発された使いやすい会話形式のプログラム言語である．経営や科学的問題を解くための道具として，あるいはコンピュータ学習の道具として広く用いられている．

BASICはCOCOLやFORTRANのようなプログラム言語と良く似ているが，特に利用者がコンピュータを簡単に使いこなせるように設計されている．

BASICは平易な英語と一般的な数式でプログラムを作成する．ユーザーは直接にテレタイプでプログラムをタイプすることによりコンピュータを利用する．

BASICは最も習得しやすいプログラム言語の一つであるが，習熟すればより効果的な高級な手法を用いることもできる．

MARK Ⅱ BASICは次の５つから構成されている．

1．エディタ
2．コンパイラ
3．ローダ
4．ランタイム・システム（BRTS）
5．同上　オーバーレー部分

エディタはソース・プログラムのタイプを受けとり，あるいは記憶装置から呼び出して編集し，RUNコマンドによりコンパイラにプログラムをつなげる．

コンパイラは，ソース・プログラムを仮の命令に翻訳してローダにつなげる．

ローダは，このコンパイルされたプログラムと，ランタイム・システムを一緒にして，コアにロードする．

ランタム・システムは，命令を一つずつ解いて，必要ならば，オーバーレーを呼び込んで実行する．実行が終了するとエディタにコントロールを返す．

表-1 BASIC ステートメント要約

ステートメント	一 般 形 式	説 明
CHAIN	CHIN DEV:Filename, ex	プログラムの実行を中止して，指定した記憶装置から指定したプログラムを呼び出して，実行する．
DATA	DATA $X_1, X_2, X_3, \cdots, Xn$	X_1～Xnは READ ステートメントの変数に対応した数値，またはストリングで，ストリングは必ず引用符(〃)で囲まれる．
DEF	DEF FNz(v)=X	ユーザ関数の定義で，zは任意の英字でXは式である．引数vは4つまでの単一数値変数と2つまでの単一文字変数が許される．
DIM	(数値変数) DIM $v_1(n_1), v_2(n_2, m_2)$	2次元までの配列を定義でき，vは配列変数で，n，mは要素の数である．一度に定義する配列の数は任意．
	(ストリング) DIM $v_1\$(I), v_2\$(K, L)$	Iはストリング $v_1\$$ の長さで，8桁を超えるストリングは，DIM で定義が必要．Kは配列 $v_2\$$ の要素の数で，Lはその長さでストリングの配列は1次元のみ．
END		プログラムの最終ステートメントで，無いとエラーになる．
FOR−TO−STEP	FOR $v=X_1$ TO X_2 STEP X_3	プログラム・ループを形成するもので，NEXT v ステートメントまでのループ・グループを繰返し実行する．vは初期値を式X_1の値で繰返しの都度式X_3の値を加え，式X_2の値を超えるとループの実行を完了する．X_3 を省略すると$X_3=1$と見なされる．
GOSUB	GOSUB X	サブルーチンを実行するステートメントで，Xはサブルーチンの入口のライン番号．
GO TO 又は GOTO	GO TO n 又は GOTO n	ライン番号nのステートメントへ分岐する．
IF GO TO IF THEN	IF f_1 r f_2 GO TO n IF f_1 r f_2 THEN n	f_1とf_2を関係演算子rで比較し，結果が真であればライン番号nのステートメントへ分岐する．f_1，f_2がストリングの時は ASCII コードの値を用いる．
INPUT	INPUT $v_1, v_2, \cdots, vn$	疑問符(？)を印刷して変数vに対応する値をテレタイプから入力させる．
LET	LET v=f	等号の右側の変数，定数あるいは式の値を左側の変数に入れる．
NEXT	NEXT v	FOR ステートメントに対応して，プログラム・ループの終りのステートメントを示す．変数vはFOR ステートメントのvと同じであることが必要．
PRINT	PRINT $a_1d_1a_2d_2\cdots dn-1an$	a_1～anまでのデータをデリミッタd(，または；) に従った形式で印刷する．
RANDO−MIZE	RANDOMIZ	RND 関数の初期値を与える．

ステートメント	一 般 形 式	説 明
READ	READ $v_1, v_2, \cdots, vn$	変数 v の初期値を対応するデータリストから読み込む.
REM	REM ……	プログラムに注釈文をつける.
RESTORE	RESTORE	データリストからデータを読む位置を先頭に戻す.
RETURN	RETURN	サブルーチンの出口のステートメントで実行されたGOSUB ステートメントの次のステートメントへ分岐する.
STOP	STOP	プログラムの実行を終了される.
UDEF	UDEF function name	アセンブラーで作られたユーザ関数を連結させる.
USE	USE $v_1, v_2 \cdots, vn$	配列 v をUDEF で宣義するユーザ関数で使えるようにする.

表-2 BASIC ファイルステートメント要約

ステートメント	一 般 形 式	説 明
CLOSE #	CLOSE #N	FILE #Nステートメントで開かれているファイルNを閉じる. Nはファイル番号である. (N=1~4)
FILE #	FILE # n:s	ファイルをオープンするステートメントで, 順に固定長ASCII, 可変長ASCII, 固定長数値, 可変長数値の各形式のファイルに適用される. nは0~4までのファイル番号で, SはDEV:FILE. EXを表すストリングである.
FILEV #	FILEV # n:s	
FILEN #	FILEN # n:s	
FILEVN #	FILEVN # n:s	
INPUT #	INPUT #N : $v_1, v_2, \cdots, vn$	v_1~vnまでの変数に, ファイル番号Nのファイルからデータを読み込む.
IF END #	IF END #N THEN n	ファイル番号Nのファイルの最後のデータを超えて, さらに入出力を試みた時, あるいは入出力に誤りがあった時ライン番号 n のステートメントへ分岐する.
PRINT #	PRINT #N : $a_1, a_2, \cdots an$	a_1~anまでのデータをファイル番号Nのファイルへ出力する.
RESTORE #	RESTORE #N	ファイル番号N のファイルのデータの位置を示すポインターを先頭に戻す.

表-3 BASIC 制御コマンド要約

コマンド	一般形式	説明
BYE	BYE	BASICの使用を終了し，キーボードモニターに制御を戻す.
CTRL／C	CTRL／C	プログラムの実行を中止し，OS/8BASICモニターに制御を戻す．BASICモニターの制御下の時は，キーボード・モニターに制御を戻す.
CTRL／O	CTRL／O	テキストの印刷を中止して，制御をBASICモニターへ戻す.
LIST LI	LIST LI	ヘディング付きで，プログラムリストを印刷させる.
LIST n LI n	LIST n LI n	ヘディング付きで，プログラムリストをライン番号nから印刷する.
LISTNH		ヘディング無しの他は，LIST と同じ.
LISTNHn		ヘディング無しの他はLIST nと同じ.
NAME NA	NAME FILE.EX NA FILE.EX	ロードされているプログラムの名前を変更する.
NEW NE	NEW FILE.EX NE FILE.EX	新しく作成するプログラムに名前をつける.
OLD OL	OLD DEV：FILE.EX OL DEV：FILE.EX	既に作成されているプログラムを記憶装置から呼び出す.
RUN RU	RUN RU	ロードされているプログラムのヘディングを印刷して，コンパイルと実行を行なう.
RUNNH	RUNNH	ヘディングを印刷しない他はRUNと全く同じ.
SAVE SA	SAVE DEV:FILE.EX SA DEV:FILE.EX	ロードされているプログラムを指定した記憶装置に保存する.
SCRATCH SC	SCRATCH SC	ロードされているプログラムの全てのステートメントを除去する.

表-4 BASIC 関数の要約

関数名	説明
ABS（X）	引数Xの絶対値を与える.
ASC（X）	文字XのASCIIコードでの位置を10進数で与える.（附表－Aを参照）
ATN（X）	正接がXの角度(ラジアン)を与える.
CHR$（X）	ASCIIコードの位置が数字XであるASCII文字を与える.（附表－Aを参照）
COS（X）	角度がX(ラジアン)の余弦値を与える.
DAT$（X）	MM/DD/YYの形式で日付を与える. 引数Xはダミーである.
EXP（X）	e^Xを与える.（e=2.71828）
FNA（X）	DEFステートメントによりユーザー関数を定義する. 以降は他の関数と同様に使える.
INT（X）	引数Xを超えない最大の整数値を与える.
LOG（X）	Xの自然対数値(lnX)を与える.
PNT（X）	PRINTステートメントでのみ使用できる. 10進数表示XのASCIIコードを出力する. 印刷しない制御コード等の出力に便利である.
POS（X$,Y$,Z）	ストリングY$のZ番目の文字が, ストリングX$に最初に表われる位置を与える.
RND（X）	0～1の間の乱数を与える.
SEG$（X$,Y,Z）	ストリングX$のY～Z番目までのサブ・ストリングを与える.
SGN（X）	Xの符号により－1(X<0),0(X=0),1(X>0)の数値を与える.
SIN（X）	角度X(ラジアン)の正弦値を与える.
SQR（X）	Xの絶対値の正の2乗根を与える.
STR（X）	数値Xの各桁をストリング表示する.
TAB（X）	PRINTステートメントのみで使用する. 印刷用のプリント・ヘッドを行の先頭からX桁目へ移動させる.
TRC（X）	X=1でON, X=0でOFFになり, ONの時に実行するステートメントのライン番号をすべて印刷するデバッグ用のトレース機能である.
VAL（X$）	ストリング表示の数値を数値タイプにする.

表-5　コンパイル時のエラーメッセージ

エラーメッセージはXX YY と印刷される．{XXはエラーコード / YYはエラーのライン番号}

エラーコード	説明
CH	CHAINステートメントの誤り
DE	DFFステートメントの誤り
DI	DIMステートメントの誤り，または72桁以上のストリング，または配列の重複定義
FN	ファイル番号，ファイル名の誤り
FP	FORステートメントの誤り
FR	未定義の関数の使用，または引数の誤り
IF	THEN，またはGOTOの未指定
IO	I/Oエラー
LS	LETステートメントで符号なし
LT	80文字以上のステートメント
MD	ライン番号の重複定義
ME	ENDステートメント無し
MO	予期されるオペランドが無い
MP	括弧が無いか，誤った使用
MT	数値タイプと文字タイプ・データの混用
NF	対応するFORステートメントの無いNEXTステートメント
NM	GOTO，GOSUB，THENの後にライン番号の指定無し
OF	出力ファイルエラー
PD	ステートメントが複雑すぎるか，FORループのNESTが多すぎて解析不能
QS	ストリング・リテラルが長すぎるか，終りの引用符無し
SS	添字または関数の誤り
ST	変数，ライン番号，リテラル等が多すぎて，コンパイル不能
SY	システムが不完全{BASIC,SD / BCOMP,SD}等が無い
TB	プログラムが大きすぎる
TD	DATAが多すぎる
TS	ストリングリテラルの総文字数が多い
UD	UDEFステートメントの誤り
UF	対応するNEXTの無いFORループ
US	定義されていないライン番号を使用
UU	USEステートメントの配列の誤り
XC	ステートメントに余分の文字がある

表-6　実行時のエラー・メッセージ

エラーメッセージは XX AT LINE YYYYY {XXはエラーコード / YYYYYはエラーのあるライン番号}

エラーコード	説明
BO	ファイル・バッファが使用不能
CI	CHAINの実行不能，装置無し
CL	CHAINの実行不能，プログラム無し
DA	データ・リストの不足
DE	I/Oエラー
DO	使用可能な記憶装置の不足
DV	ゼロの割算
EF	ファイルの終了(END of File)
EM	指数が負
EN	ファイル・オープンのエラー装置がないか，すでに使用されているファイルがない
FB	指定したファイルが既に使用されている
FC	可変長ファイルのクローズの誤りまたはクローズ済みのファイルを読む
FE	ファイル・オープンの誤り，装置が無いか，コアにスペースが無い
FI	オープンしてないファイルのクローズ
FM	ファイル番号が負
FN	ファイル番号が0,1,2,3,4 でない
FO	ファイル番号が4095を超える
GR	対応するGOSUBの無RETURN
GS	GOSUBのネストが10を超える
IA	UDEF関数の引数の誤り
IF	DEV：Filename. EXの指定の誤り
IN	ファイルのオープンで装置が無い．
IO	テレタイプの入力バッファがオーバー
LM	ゼロか負の数のLOG計算
OE	SYS装置のI/Oエラー
OV	数値，または入力のオーバーフロー
PA	POS関数の引数エラー
RE	終りになったファイルをさらに入力
SC	ストリングが72桁を超える
SL	ストリングが長すぎるか，定義なし
SR	数値ファイルからストリングを読む
ST	入力時のストリングの切捨て，最大桁数まで入力する
SU	添字の値が，DIMの定義を超えた
SW	数値ファイルへ，ストリングを出力
VR	可変長ファイルを読もうとした
WE	終りになったファイルをさらに出力

❽ RPG（Report Program Generator）

1．概要

RPG（レポート　プログラム　ジェネレータ）は1960年代の初期にIBMによって事務処理用の問題向き言語として開発されたもので，次のような特長がある．

1. 商業計算，事務計算のデータ処理に適した編集，演算，ファイルの取扱いの容易さなどの点ですぐれている．
2. プログラミングやデータ処理について特別な知識や経験がなくとも使える．
3. プログラミングというより様式にデータやパラメータを記入するだけでよいので，作業が容易迅速である．
4. 電子計算機そのものの機械的機能とあまり関係がなく，問題中心の考え方ができる．

ＣＥＣ５５５　RPGはIBM社の最新版であるRPGⅡの仕様を参考として，ミニコン用に新しく設計開発したものである．ＣＥＣ５５５の基本オペレーティングシステムであるマークⅡモニタの管理下でモニタの内蔵する機能を有効に利用して能率的な動作をおこなっている．

適用システムの最少構成はＣＥＣ５５５（D,H,M）主記憶８Ｋ語以上，システム補助記憶100Ｋ語以上（ＣＦＤ－１３０磁気ドラム装置，ＣＦＤ－３００磁気ディスク装置，ＣＦＰ－５００フロッピディスク装置，ＤＲＩＳＫ－２００２磁気ディスク装置のいずれか），コンソールタイプライタ（ＣＳ－１００，ＣＳ－１０００，ＣＳ－１２００，ＣＭＤ－８のいずれか）である．最大構成ではＤＲＩＳＫ－２００２磁気ディスク２台，ＣＭＴ－５またはＣＭＴ－８磁気テープ装置などを含むすべての周辺機器を含むことが可能である．

本RPGシステムではコンパイラとオブジェクトタイムシステムとは自動的にリンクして１パス動作が可能であるが，ソースコンピュータとオブジェクトコンピュータとの分離動作も可能である．

2．RPG言語の紹介

RPGは請求書作成とか給与計算といった業務をおこなうための，効果的な，使いやすい手段として作られたプログラミング言語である．

電子計算機によるデータ処理は，入力（処理されるもの），処理および出力（必要な結果）から成り立っている．

RPGでプログラムを作成する場合，これらのステップを次のような順序で考える．

1. どのような出力が必要であるか（出力）
2. 処理されるものとして何があるか（入力）
3. どのような情報が不足しており，したがって，作り出さなければならないか（処理）

従来のプログラミング言語と異り，これらの条件を仕様書と名づける用紙にデータとして書き込むだけで，コーディングの作業は終了する．ＲＰＧを使っての作業では，プログラミングもフローチャートも不要であるといわれるゆえんである．

3．作業の計画立て

3―1　ステップ１：最終結果

まず，RPGによる作業で，どのような出力が必要であるかを定義しなければならない．ある作業をRPG言語で記述する場合，どのような情報を印刷あるいは穿孔させたいかを決めなければな

らない．次に，出力の体裁は，どのようにすべきか決める．

例えば，社員の給与報告書を作り出す場合を考えてみよう．報告書には，社員名，各社員の就労時間数（正規の時間または時間外），および各社員の給料を入れなければならないだろう．すると，どのような情報（氏名，時間数，給料）を印刷させるかが決まる．

次に，それらの出力を特別な並べ方にしたいだろう．つまり，名前（NAME），時間数（JIKAN），給与（KYUYO）といった3つの情報を，各欄の頭に見出しとしてつけ，各印刷行は1行おきに印刷する．

3－2　ステップ2：使用可能な情報

入力テープ上には，どのような情報が入っているかを考えてみる．

必要な情報がすでに紙テープ上にあれば，何も計算をおこなう必要はない．電子計算機は，その情報を紙テープから印刷報告書に移すことができる．入力紙テープには，社員名，出勤日数，残業時間数およびその他の情報が記録されている．

必要な出力の一部は，電子計算機に命令して，名前および時間数を紙テープから報告書に移させることができる．しかし，各社員の総支払額もまた必要であり，これは入力紙テープ上には入っていない．

3－3　ステップ3：必要な情報

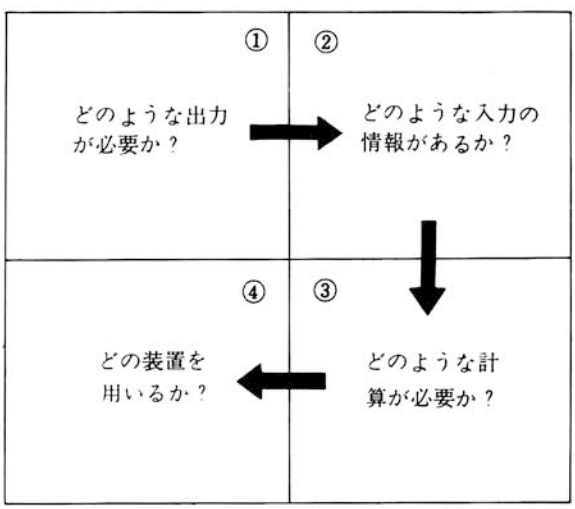
今度は，どのような情報がまだ不足しているか，あるいは，必要な出力を出すには何が必要であるか，を決めることである．

総支払額を出すには，本給＋諸手当の計算をおこなわなければならない（残業手当は，残業時間数×時間給の計算で計算される）．この結果がその月に支払われる総支払額になる．したがって，計算することが出力のために欠けている情報を与える方法になるわけである．

3－4　ステップ4：システムの諸装置

最後に，必要な出力を出すために，どのような装置が必要であるか知る必要がある．RPGでは，どの装置から出力を出すかを記述しなければならない．また入力はどこからということも記述しなければならない．上記4段階をまとめたものが図である．

RPG言語で作業を記述する場合，通常4種類の記入用紙（仕様書）が用いられる．



1図　RPGによる作業の4つのステップ

4．RPG仕様書

出力仕様書には，出力としてどのような情報をのせるか，また，それをどのような体裁にするかを記入する．

入力仕様書では，どのような情報がすでに入力テープに入っているかを記述する．

また，演算仕様書では，どのような計算をおこなわせたいかを指定する．ファイル仕様書には，どの装置が入力と出力を取り扱うかを記入する．

RPGプログラムを書く場合，ファイルとか，フィールドとか，標識とかいったことについて，知っていなければならない．したがって，このことについてまず説明する．

4－1　ファイル

ファイルとは，一群の関連性のあるレコードの集りである．ファイルには入力と出力の2種類がある．入力ファイルは，編成された情報を内容とするデータ群である．

出力ファイルは，データ群の場合と報告書の場合の2つの形式に分かれる．RPGでは，入力および出力の各ファイルにそれぞれ違った固有の名前を与えて，各ファイルを電子計算機に区別しなければならない．この名前は，RPGでプログラムを書く人が割り当てる．

4－2　フィールド

RPG言語で作業を記述する場合，フィールドということを理解しておくことが大切である．カード上のフィールドとは，あるデータ項目を表わすためにカード上にとったある数のカード桁（カラム）のことをいう．

フィールドの長さは，何桁のカード桁であっても構わない．

例えば，1桁の場合，6桁の場合，20桁の場合，あるいは，96桁の場合などいろいろの場合がある．

しかし，数字フィールドは，15桁を越えることは許されない．

4－3　標　識

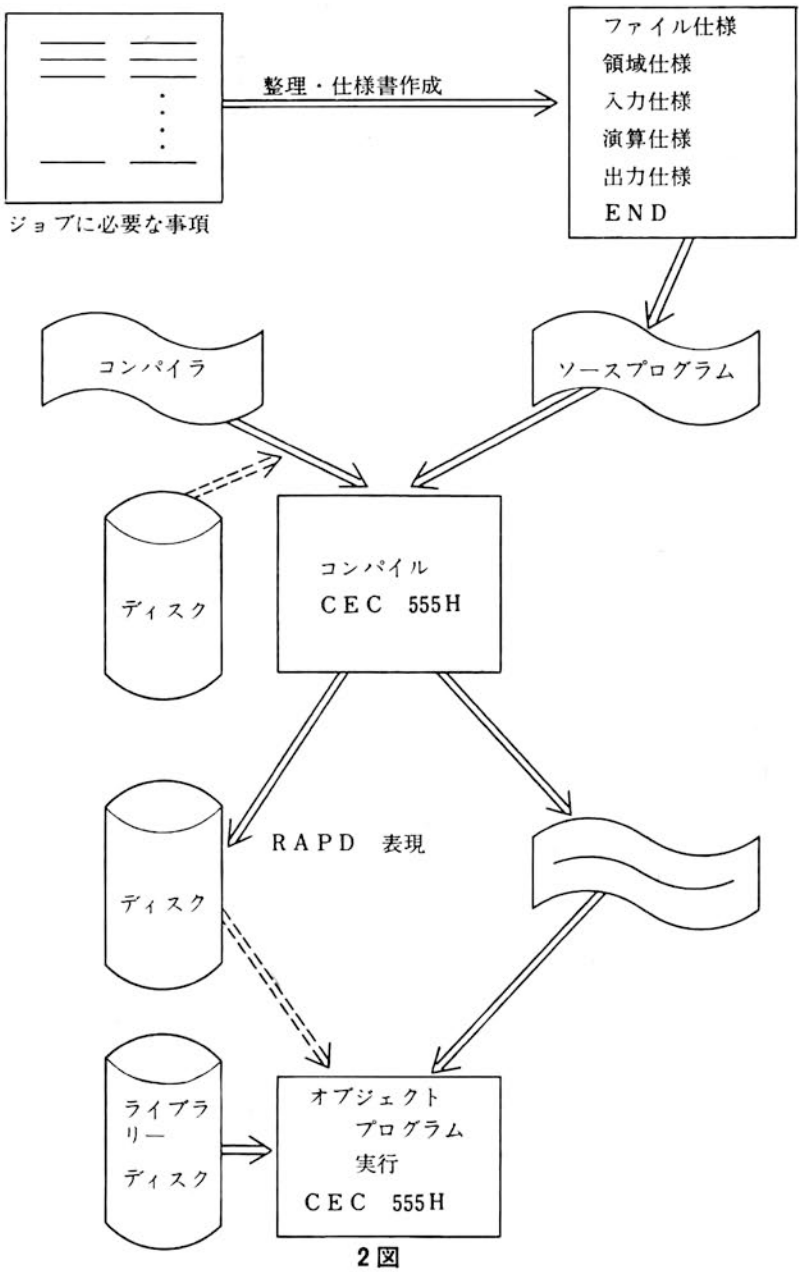
標識は，ついたり，消えたりするライトにたとえることができる．

これは，条件が起ったとき，またはある条件が起らなかったとき，電子計算機に対して合図を送る．例えば，データが1レコード読まれるたびに，ある標識が“ON”になるように割り当てたとする．そうすれば，この標識が“ON”になるたびに，電子計算機にはデータが1レコード読まれたということが合図される．

そうすれば，電子計算機は，この標識がONのときにおこなわなければならない処理操作を全ておこなうことができる．

このような処理操作には，その入力情報を報告書に印刷したりする場合があるだろう．これらの処理操作が終ると，標識は次のデータが読まれて再びONになるまでOFFの状態にされる．この過程は，ファイル内のすべてのレコードが読み取られてしまうまで，繰り返しおこなわれる．

5. プログラム作成手順



2図

6. 例題

例題 1

3図および**4図**のような入力フォーマットのデータが2つある．各々の商品コードを見比べて一致したら数量と単価の積（＝金額）を行わせ，**5図**のような出力フォーマットで出力させる．その際，得意先別にページを変え，得意先の変わるごとに数量と金額の合計を，また，全処理終了後にその総合計を出力させること．ただし，2つのデータのうち1つは補助記憶装置にTANKM.MFとして登録されており，他の1つは高速紙テープリーダからの入力とする．

入力フォーマット

トランザクションファイル：PTR

	得意先 TŌKU X(12)	T A B	商品コード SHŌH 9(4)	T A B	数量 SURYŌ 9(6)	CL RF	

3図

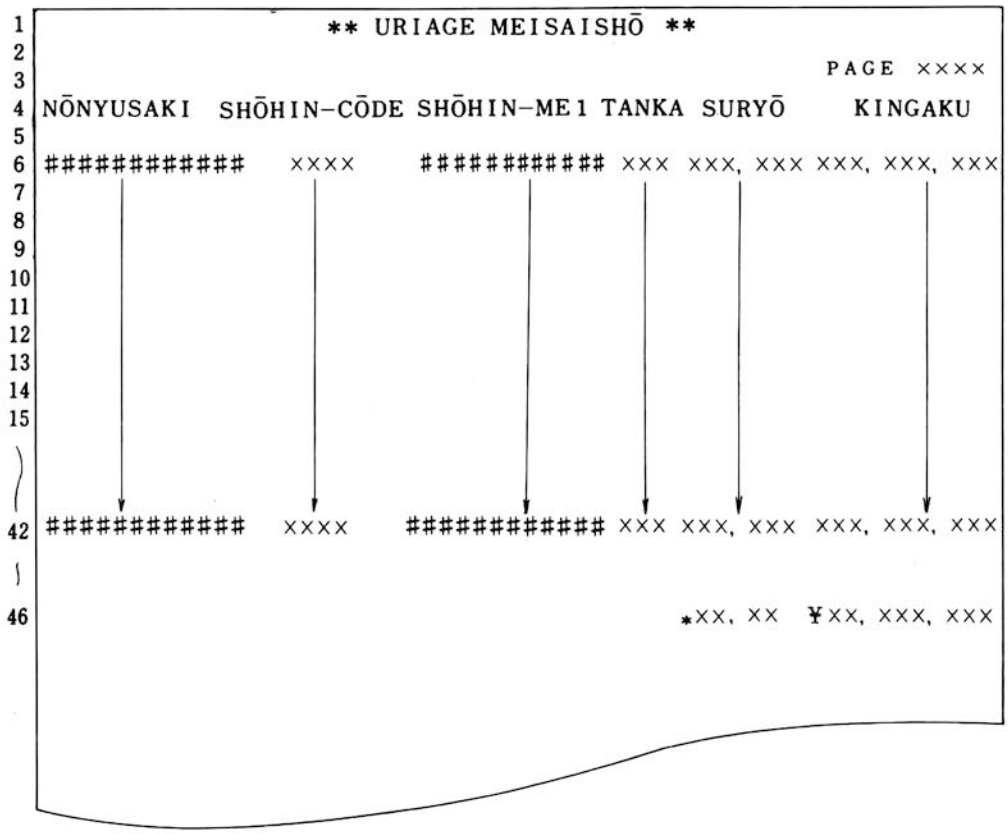
マスターファイル：TANKM.MF

	商品コード TSHŌH 9(4)	単価 TANK1 9(3)	商品名 SNAME X(12)	CL RF	

X(8)……英文字
9(8)……数字
N……長さ

4図

出力様式〈例〉



5図

例題 2

1ケ月分の売上げファイル（トランザクション）と前月の売上げマスターファイル（旧マスター）を補助記憶装置から読取り、個人CODEによる突合せを行い、新しい売上げマスターファイル（新マスター）を補助記憶装置に書込むと同時に、変更・追加のあるレコードをラインプリンタに打ち出しその月の売上げ一覧表を作成する.

〈処理概要〉

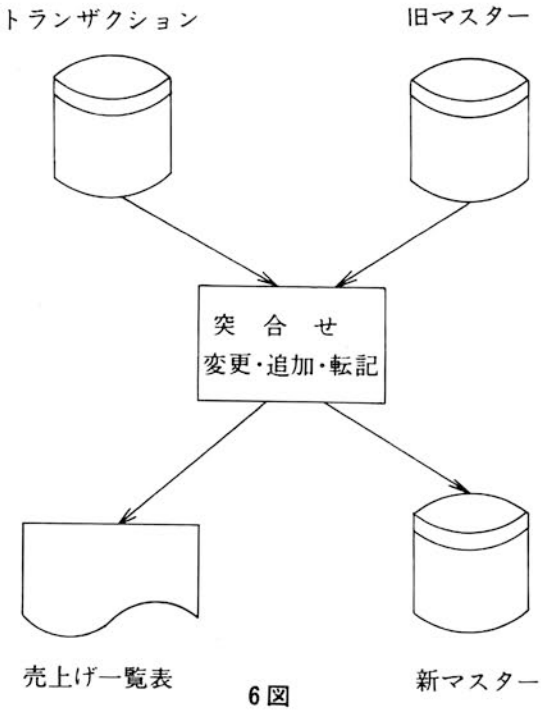
1. マスターファイルはソートされているものとする.
2. トランザクションファイルは、マスターファイルとMATCHINGするレコードと、新しくマスターファイルに追加するレコードからなっている.

   レコードがMATCHINGする場合は売上げ高を更新し、新マスターファイルのレコードとする. 追加するレコードはそのまま新マスターファイルのレコードとする.

   変更のあったレコードと追加のレコードはラインプリンタにも出力する. 補助記憶装置には全てのレコードを書き込む.

   ラインプリンタに出す帳表には売上げの個人別前月比（%）と当月分の総計も打ち出す.
3. トランザクションファイルの中でマスターファイルとMATCHINGするレコードとして入れたレコードが、MATCHINGしなかった場合には該当なしとしてそのレコードのCODEをテレタイプに打ち出した後に削除する.

〈プロセスフローチャート〉



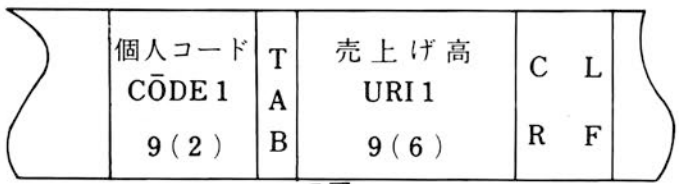
6図

〈入出力フォーマット〉

（入力フォーマット）

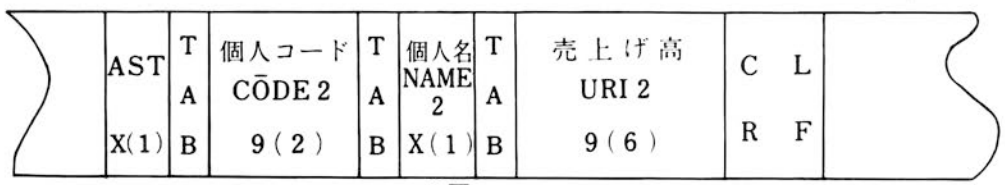
トランザクションファイル

その1　更新するレコードフォーマット



7図

その2　追加するレコードフォーマット



8図

マスターファイル



9図

(出力フォーマット)

マスターファイル

その1　更新・転記のレコードフォーマット



10図

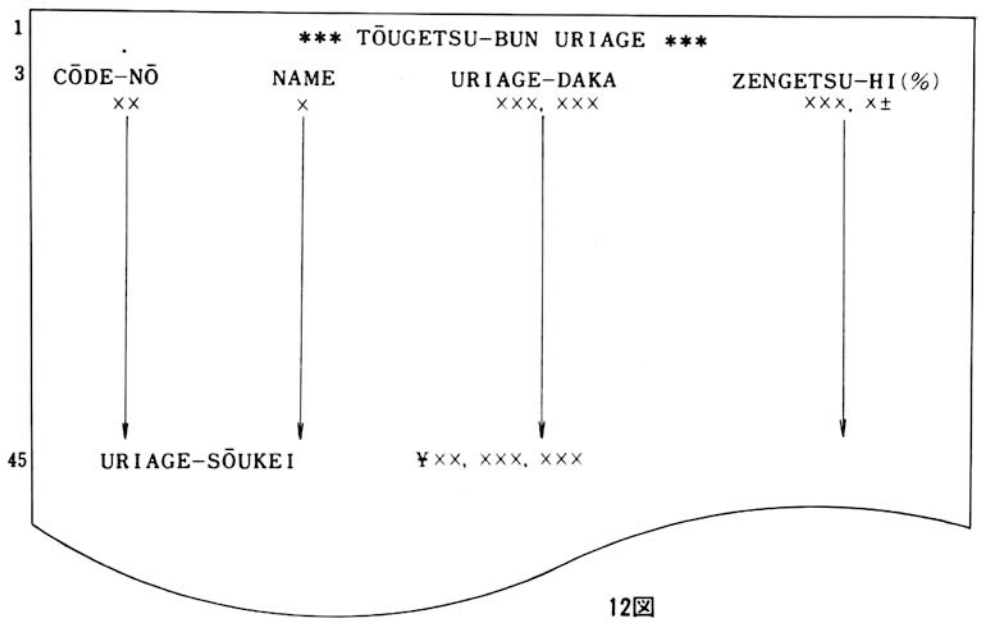
その2　追加のレコードフォーマット



11図

(出力フォーマット)

ラインプリンタ



12図

プログラム名 URIAGE MEISAI PROGRAM **CEC555 RPG**

作成日 50年 3月 31日　プログラマ M.KAWASHIMA　**ファイル 仕様書**　1/6

行	A 仕様書区分 / ファイルの数	B ファイルの種類 I/O	B ファイル名	C デバイス名：〔ファイル名〕	D	E ファイルの指定 P/S	E ファイルの最後の処理 (E)	E 配列 (A/D)	E 形式 (F/V)	F レコードの長さ	F TAB	G 行数	G 最終行	G OF標識	H コメント	I
00	FILE,3	I,	URT	PTR:		P,	E,	,	V	24						
05		I,	TANK	SYS:TANKM.MF		S,	,	,	F	22						
10		O,	LPF	LPT:		,	,	,	F	80			,47	,OF		
15																
20																
25																
30																
35																
40																
45																
50																
55																
60																
65																
70																
75																
80																
85																
90																
95																

第1行目：URTという入力ファイル名を指定し，これをプライマリとする．プライマリ終了時に全処理終了としPTRよりの入力としている．データは可変長である．

第2行目：TANKという入力ファイル名を指定し，補助記憶装置よりTANKM.MFというファイル名での入力とする．これをセカンダリーとしデータは固定長である．

第3行目：LPFという出力ファイル名を指定し，LPTに出力させる．47行でOF標識がONになる．

プログラム名 URIAGE MEISAI PROGRAM **CEC555 RPG**

作成日 58年 3月 31日　プログラマ M.KAWASHIMA　作業域・定数仕様書　2/6

行	A 仕様書区分 TAB	B フィールド名 TAB	C フィールドの長さ	C 小数部桁数	D 定数値
00	WŌRK	TŌTAL	10	,Ø	Ø
05		PAGE	4	,Ø	1
10		CŌUNT	1	,Ø	1
15		YAS	8	,Ø	Ø
20		SUM	10	,Ø	Ø
25		KING	10	,Ø	Ø
30		WŌRKT	6	,Ø	Ø
35		HEAÐ	23		"** uRIAGE MEISAISHŌ **"
40		ABC	4		"PAGE"
45		KEI	6		"SŌUKEI"
50		LABEL	71		"NŌNYUSAKI SHŌHIN-CŌÐE SHŌHIN-MEI"-
55					" TANKA SURYŌ KINGAKU"
60					
65					
70					
75					
80					
85					
90					
95					

入力レコードにない処理に必要なフィールドを確保するものであり，数字の場合は必ず小数位を指定する．英文字では " で囲まれた字数とフィールドの長さが一致するようにする．

プログラム名 URIAGE MEISAI PROGRAM **CEC555 RPG**

作成日 50年 3月 31日　プログラマ M.KAWASHIMA　**入力仕様書**　

行	A 仕様書区分	B ファイル名	C レコード標識	D レコード識別条件 位置	否定	文字	AND/OR	E ～	F	G 制御レベル	突合レベル	H	I
	TAB	TAB	TAB	フィールド名	TAB	始め	長さ	小数桁					
00	INPUT	URT	Ø1	1									
05				TŌKU		1	12			L1			
10				SHŌH		13	4	Ø			M1		
15				SURYŌ		17	6	Ø					
20				CRLF		23	2						
25		TANK	Ø2	1									
30				TSHŌH		1	4	Ø			M1		
35				TANK1		5	3	Ø					
40				SNAME		8	12						
45				CRF		20	2						
50													
55													
60													
65													
70													
75													
80													
85	ファイル仕様書で指定した入力ファイルは全てここで定義されなければならない．URT，TANKの各々のレコード標識01，02は1桁目に＊がなければONになる．制御レベルL1はTOKUに対して設けられ，この内容の変化あるごとにL1で指定された処理を行なう．M1はSHOHとTSHOHとの突合わせを行なうためのものである．この時，数字なら数字と，英文字なら英文字との組合わせでなくてはならない．フィールドの長さの指定は図1を参照．												
90													
95													

CEC555 RPG

プログラム名 URIAGE MEISAI PROGRAM

作成日 58年 3月 31日　プログラマ M. KAWASHIMA　演算仕様書

行	A 仕様書区分 TAB	B 制御標識 H/D/T/E/G/I TAB	C 演算ルーチン・各命令の実行条件 否定 標識名, AND/OR, 否定 標識名,	D 演算命令	E フィールド名1 (標識名/タグ名/リテラル)	F フィールド名2 (標識名/タグ名/リテラル)	G フィールド名3(結果) (標識 <,=,>)	H	I
00	COMPUTE	D	Ø1,&,ME						
05				MPY	SURYŌ	TANK1	KING		
10				ADD	KING	TŌTAL	TŌTAL		
15				ADD	SURYŌ	WŌRKT	WŌRKT		
20		T	L1						
25				ADD	PAGE	CŌUNT	PAGE		
30				ADD	TŌTAL	SUM	SUM		
35				ADD	WŌRKT	YAS	YAS		
40		D	Ø1,&,L1						
45				WŌPEN	TANK	URT			
50									
55									
60									
65									
70									
75									
80									
85									
90									
95									

明細演算(D)
- 01, &, MEは01がONで突合わせが成立した時に指定の演算を行なわせる.
  MPY－乗算　SURYO×TANK1＝KING
  ADD－加算　KING＋TOTAL＝TOTAL
- 01, &, L1は01とL1が共にONのとき, この場合, TOKUが変わってもURTのデータが存在する時指定の命令を行なう.

合計演算(T)　T L1
L1がONの時に指定の演算を行なわせる.

- WOPEN TANK URT
  URTファイルの01標識をOFFにしTANKファイルを再オープンさせる（URTファイルをそのままにし, TANKファイルが最初から始まる）.

CEC555 RPG

プログラム名 URIAGE MEISAI PROGRAM　　出力仕様書　　ページ 5/6

作成日 58年 3月 31日　　プログラマ M.KAWASHIMA

行	A 仕様書区分 TAB	B ファイル名 TAB	C 制御標識 HDITEGI	改行(改頁)制御 紙送り前	紙送り後	D レコード・フィールド出力条件 否定 標識名 AND/OR 否定 標識名 ～	E 始め TAB	F フィールド名	G 編集語
00	OUTPUT	LPF	H	,2	,+1	OF			
05	ファイル仕様書で指定された出力ファイルは全てここで定義されなければならない。						25	HEAD	
10			H	,3	,+1	OF			
15	H, 2, +1 OF						60	ABC	
20	H, 3, +1 OF						66	PAGE	ZZZ9
25	H, 3, +2 OF								
30	OF標識がONになったときに見出しが出力される。一番最初はOF標識がONになっている。		H	,5	,+2	OF			
35	D 01, &, ME						1	LABEL	
40	L1, OF		D			01,&,ME			
45	01とME共にONで、L1あるいはOFのどちらかがONになったときに明細を出力する。					L1,,OF			
50	61 KING B ZZZ, ZZZ, ZZ9						1	TOKU	
55	BはKINGを出力した後にKINGの内容をゼロあるいはブランクにセットする。Z, 9は編集語でZはゼロサプレス、9はそのままの数を印刷する。また＊は＊に対応するものが最初の有効数字ならば、それ以降は全て数字に変換され、それ以外は＊に変換される。￥は＊と始んど同じであるが、最初の有効数字の前に1つだけ￥が印刷される。						17	SHOH	ZZZ9
60							29	SNAME	
65							41	TANK1	ZZZ,ZZ9
70							52	SURYO	ZZZ,ZZ9
75							61	KING,B	ZZZ,ZZZ,ZZ9
80	T LR		T	,46	,+2	L1			
85	CONTROL Zで全データの終了が告げられると、LRがONになる。それと共にL1もONになり、L1に指定された処理とLRに指定された処理が順次行なわれる。						52	WORKT,B	＊＊＊,＊＊9
90							61	TOTAL,B	¥¥¥,¥¥¥,¥¥9
95			T			LR			
							41	KEI	

プログラム名 URIAGE MEISAI PRŌGRAM **CEC555 RPG**

作成日 50年 3月 31日　プログラマ M. KAWASHIMA　出力仕様書　6/6

行	A 仕様書区分 TAB	B ファイル名 TAB	C 制御標識 H/D/T/F/G/I 改行(改更)制御 (紙送り前, 紙送り後)	D レコード・フィールド出力条件 否定 標識名, AND/OR, 否定 標識名, ～	E 始め	E TAB	F フィールド名	G 編集語
00	OUTPUT				49		YAS,B	***,***,**9
05					61		SUM	¥¥¥,¥¥¥,¥¥9
10	END							
15								
20								
25								
30								
35								
40								
45								
50								
55								
60								
65								
70								
75								
80								
85								
90								
95								

例題1のソースリスト

```
/** URIAGE MEISAI PROGRAM (3.31.75) **
/
FILE,3   I,URT    PTR:              P,E,,V  24
         I,TANK   SYS:TANKM.MF      S,,,F   22
         O,LPF    LPT:              ,,,F    80        ,47,OF
/
WORK     TOTAL    10,0
         PAGE     4,0       1
         COUNT    1,0       1
         YAS      8,0
         SUM      10,0
         KING     10,0
         WORKT    6,0
         HEAD     23        "** URIAGE MEISAISHO **"
         ABC      4         "PAGE"
         KEI      6         "SOUKEI"
         LABEL    71        "NONYUSAKI      SHOHIN-CODE    SHOHIN-MEI"-
                            "      TANKA       SURYO      KINGAKU"
/
INPUT    URT      01        1,#*
                            TOKU     1,12              L1
                            SHOH     13,4,O            ,M1
                            SURYO    17,6,O
                            CRLF     23,2
         TANK     02        1,#*
                            TSHOH    1,4,O             ,M1
                            TANK1    5,3,O
                            SNAME    8,12
                            CRF      20,2
/
COMPUTE  D        01,&,ME
                            MPY      SURYO    TANK1    KING
                            ADD      KING     TOTAL    TOTAL
                            ADD      SURYO    WORKT    WORKT
         T        L1
                            ADD      PAGE     COUNT    PAGE
                            ADD      TOTAL    SUM      SUM
                            ADD      WORKT    YAS      YAS
         D        01,&,L1
                            WOPEN    TANK     URT
/
OUTPUT   LPF      H,2,+1    OF
                                     25       HEAD
                  H,3,+1    OF
                                     60       ABC
                                     66       PAGE     ZZZ9
                  H,5,+2    OF
                                     1        LABEL
                  D         01,&,ME
                            L1,,OF
                                     1        TOKU
                                     17       SHOH     ZZZ9
                                     29       SNAME
                                     41       TANK1    ZZZ,ZZ9
                                     52       SURYO    ZZZ,ZZ9
                                     61       KING,B   ZZZ,ZZZ,ZZ9
                  T,46,+2   L1
                                     52       WORKT,B  ***,**9
                                     61       TOTAL,B  ¥¥¥,¥¥¥,¥¥9
                  T         LR
                                     41       KEI
                                     49       YAS,B    ***,***,**9
                                     61       SUM      ¥¥¥,¥¥¥,¥¥9
END
```

例題1のマスターファイル

（TANKM, MF）

```
0001030GOLDEN BAT
0002030MINORI
0003030ASAHI
0004030HIKARI
0005050SHINSEI
0006050IKOI
0007050ECHO
0008050FUJI
0009050VIOLET
0010050LON
0011050HOPE
0012050PEACE
0013060WAKABA
0014060HITOON
0015060ULMA
0016080HILITE
0017080000ZORA
0018080MF
0019100KOHAKU
0020100IZUMI
0021100YAMATO
0022100PEARL
0023100HILITE L
0024100SEVEN STARS
0025100LONG HOPE
0026100LONG PEACE
0027100LUNA
0028100CHERRY
0029100MARINA
0030100CURRENT
0031100MINI STAR
0032120RAN
0033120HILITE EXPO
0034120MINE
0035120MR SLIME
0036150MARL BORO
0037150OLD SPLENDER
```

例題1のトランザクションデータ

```
YAMA-SHOKAI     1       100
YAMA-SHOKAI     2       200
YAMA-SHOKAI     3       300
YAMA-SHOKAI     4       150
YAMA-SHOKAI     5       250
YAMA-SHOKAI     6       350
YAMA-SHOKAI     7       400
YAMA-SHOKAI     8       500
YAMA-SHOKAI     9       600
YAMA-SHOKAI     10      700
YAMA-SHOKAI     11      800
YAMA-SHOKAI     12      900
YAMA-SHOKAI     13      450
YAMA-SHOKAI     14      550
YAMA-SHOKAI     15      650
YAMA-SHOKAI     16      750
YAMA-SHOKAI     17      850
YAMA-SHOKAI     18      950
YAMA-SHOKAI     19      1000
YAMA-SHOKAI     20      100
YAMA-SHOKAI     21      200
YAMA-SHOKAI     22      300
YAMA-SHOKAI     23      400
YAMA-SHOKAI     24      500
YAMA-SHOKAI     25      600
YAMA-SHOKAI     26      700
YAMA-SHOKAI     27      800
YAMA-SHOKAI     28      900
YAMA-SHOKAI     29      1000
YAMA-SHOKAI     30      2000
YAMA-SHOKAI     31      3000
YAMA-SHOKAI     32      1500
YAMA-SHOKAI     33      500
YAMA-SHOKAI     34      2500
YAMA-SHOKAI     35      300
YAMA-SHOKAI     36      700
YAMA-SHOKAI     37      800
KK.KAWA 1       1000
KK.KAWA 2       100
KK.KAWA 3       150
KK.KAWA 4       200
KK.KAWA 5       250
KK.KAWA 6       300
KK.KAWA 7       350
KK.KAWA 8       400
KK.KAWA 9       450
KK.KAWA 10      500
KK.KAWA 11      550
KK.KAWA 12      600
KK.KAWA 13      650
KK.KAWA 14      700
KK.KAWA 15      750
KK.KAWA 16      800
KK.KAWA 17      850
KK.KAWA 18      900
KK.KAWA 19      950
KK.KAWA 20      2000
KK.KAWA 21      1500
KK.KAWA 22      110
KK.KAWA 23      120
KK.KAWA 24      130
KK.KAWA 25      140
KK.KAWA 26      150
KK.KAWA 27      160
```

KK.KAWA	28	170
KK.KAWA	29	180
KK.KAWA	30	190
KK.KAWA	31	200
KK.KAWA	32	1500
KK.KAWA	33	2500
KK.KAWA	34	3500
KK.KAWA	35	4500
KK.KAWA	36	5000
KK.KAWA	37	1000
SORA CO,LTD.	1	1000
SORA CO,LTD.	2	950
SORA CO,LTD.	3	900
SORA CO,LTD.	4	850
SORA CO,LTD.	5	850
SORA CO,LTD.	6	750
SORA CO,LTD.	7	750
SORA CO,LTD.	8	650
SORA CO,LTD.	9	600
SORA CO,LTD.	10	550
SORA CO,LTD.	11	500
SORA CO,LTD.	12	450
SORA CO,LTD.	13	400
SORA CO,LTD.	14	350
SORA CO,LTD.	15	300
SORA CO,LTD.	16	250
SORA CO,LTD.	17	200
SORA CO,LTD.	18	150
SORA CO,LTD.	19	100
SORA CO,LTD.	20	300
SORA CO,LTD.	21	500
SORA CO,LTD.	22	700
SORA CO,LTD.	23	1000
SORA CO,LTD.	24	1300
SORA CO,LTD.	25	2000
SORA CO,LTD.	26	2100
SORA CO,LTD.	27	2300
SORA CO,LTD.	28	2500
SORA CO,LTD.	29	2700
SORA CO,LTD.	30	3000
DREAM.KK	1	100
DREAM.KK	2	300
DREAM.KK	3	200
DREAM.KK	4	500
DREAM.KK	5	400
DREAM.KK	6	700
DREAM.KK	7	600
DREAM.KK	8	900
DREAM.KK	9	800
DREAM.KK	10	1000
DREAM.KK	11	2000
DREAM.KK	12	3000
DREAM.KK	13	4000
DREAM.KK	14	5000
DREAM.KK	15	6000
DREAM.KK	16	7000
DREAM.KK	17	8000
DREAM.KK	18	9000
DREAM.KK	19	9200
DREAM.KK	20	9400
DREAM.KK	21	9600
DREAM.KK	22	9800
DREAM.KK	25	10000
DREAM.KK	28	300
DREAM.KK	32	400
DREAM.KK	36	2000
DREAM.KK	37	200

例題１の実行結果

** URIAGE MEISAISHO **

PAGE 1

NONYUSAKI	SHOHIN-CODE	SHOHIN-MEI	TANKA	SURYO	KINGAKU
YAMA-SHOKAI	1	GOLDEN BAT ,	30	100	3,000
	2	MINORI	30	200	6,000
	3	ASAHI	30	300	9,000
	4	HIKARI	30	150	4,500
	5	SHINSEI	50	250	12,500
	6	IKOI	50	350	17,500
	7	ECHO	50	400	20,000
	8	FUJI	50	500	25,000
	9	VIOLET	50	600	30,000
	10	LON	50	700	35,000
	11	HOPE	50	800	40,000
	12	PEACE	50	900	45,000
	13	WAKABA	60	450	27,000
	14	HITOON	60	550	33,000
	15	ULMA	60	650	39,000
	16	HILITE	80	750	60,000
	17	OOZORA	80	850	68,000
	18	MF	80	950	76,000
	19	KOHAKU	100	1,000	100,000
	20	IZUMI	100	100	10,000
	21	YAMATO	100	200	20,000
	22	PEARL	100	300	30,000
	23	HILITE L	100	400	40,000
	24	SEVEN STARS	100	500	50,000
	25	LONG HOPE	100	600	60,000
	26	LONG PEACE	100	700	70,000
	27	LUNA	100	800	80,000
	28	CHERRY	100	900	90,000
	29	MARINA	100	1,000	100,000
	30	CURRENT	100	2,000	200,000
	31	MINI STAR	100	3,000	300,000
	32	RAN	120	1,500	180,000
	33	HILITE EXPO	120	500	60,000
	34	MINE	120	2,500	300,000
	35	MR SLIME	120	300	36,000
	36	MARL BORO	150	700	105,000
	37	OLD SPLENDER	150	800	120,000
				*27,250	¥2,501,500

** URIAGE MEISAISHO **

NONYUSAKI	SHOHIN-CODE	SHOHIN-MEI	TANKA	SURYO	KINGAKU
KK.KAWA	1	GOLDEN BAT	30	1,000	30,000
	2	MINORI	30	100	3,000
	3	ASAHI	30	150	4,500
	4	HIKARI	30	200	6,000
	5	SHINSEI	50	250	12,500
	6	IKOI	50	300	15,000
	7	ECHO	50	350	17,500
	8	FUJI	50	400	20,000
	9	VIOLET	50	450	22,500
	10	LON	50	500	25,000
	11	HOPE	50	550	27,500
	12	PEACE	50	600	30,000
	13	WAKABA	60	650	39,000
	14	HITOON	60	700	42,000
	15	ULMA	60	750	45,000
	16	HILITE	80	800	64,000
	17	OOZORA	80	850	68,000
	18	MF	80	900	72,000
	19	KOHAKU	100	950	95,000
	20	IZUMI	100	2,000	200,000
	21	YAMATO	100	1,500	150,000
	22	PEARL	100	110	11,000
	23	HILITE L	100	120	12,000
	24	SEVEN STARS	100	130	13,000
	25	LONG HOPE	100	140	14,000
	26	LONG PEACE	100	150	15,000
	27	LUNA	100	160	16,000
	28	CHERRY	100	170	17,000
	29	MARINA	100	180	18,000
	30	CURRENT	100	190	19,000
	31	MINI STAR	100	200	20,000
	32	RAN	120	1,500	180,000
	33	HILITE EXPO	120	2,500	300,000
	34	MINE	120	3,500	420,000
	35	MR SLIME	120	4,500	540,000
	36	MARL BORO	150	5,000	750,000
	37	OLD SPLENDER	150	1,000	150,000
				*33,500	¥3,483,500

** URIAGE MEISAISHŌ **

NŌNYUSAKI	SHŌHIN-CŌDE	SHŌHIN-MEI	TANKA	SURYŌ	KINGAKU
SŌRA CŌ,LTD.	1	GŌLDEN BAT	30	1,000	30,000
	2	MINŌRI	30	950	28,500
	3	ASAHI	30	900	27,000
	4	HIKARI	30	850	25,500
	5	SHINSEI	50	850	42,500
	6	IKŌI	50	750	37,500
	7	ECHŌ	50	750	37,500
	8	FUJI	50	650	32,500
	9	VIŌLET	50	600	30,000
	10	LŌN	50	550	27,500
	11	HŌPE	50	500	25,000
	12	PEACE	50	450	22,500
	13	WAKABA	60	400	24,000
	14	HITŌON	60	350	21,000
	15	ULMA	60	300	18,000
	16	HILITE	80	250	20,000
	17	ŌŌZŌRA	80	200	16,000
	18	MF	80	150	12,000
	19	KŌHAKU	100	100	10,000
	20	IZUMI	100	300	30,000
	21	YAMATŌ	100	500	50,000
	22	PEARL	100	700	70,000
	23	HILITE L	100	1,000	100,000
	24	SEVEN STARS	100	1,500	150,000
	25	LŌNG HŌPE	100	2,000	200,000
	26	LŌNG PEACE	100	2,100	210,000
	27	LUNA	100	2,300	230,000
	28	CHERRY	100	2,500	250,000
	29	MARINA	100	2,700	270,000
	30	CURRENT	100	3,000	300,000
				*29,150	¥2,347,000

** URIAGE MEISAISHO **

NONYUSAKI	SHOHIN-CODE	SHOHIN-MEI	TANKA	SURYO	KINGAKU
DREAM.KK	1	GOLDEN BAT	30	100	3,000
	2	MINORI	30	300	9,000
	3	ASAHI	30	200	6,000
	4	HIKARI	30	500	15,000
	5	SHINSEI	50	400	20,000
	6	IKOI	50	700	35,000
	7	ECHO	50	600	30,000
	8	FUJI	50	900	45,000
	9	VIOLET	50	800	40,000
	10	LON	50	1,000	50,000
	11	HOPE	50	2,000	100,000
	12	PEACE	50	3,000	150,000
	13	WAKABA	60	4,000	240,000
	14	HITOON	60	5,000	300,000
	15	ULMA	60	6,000	360,000
	16	HILITE	80	7,000	560,000
	17	OOZORA	80	8,000	640,000
	18	MF	80	9,000	720,000
	19	KOHAKU	100	9,200	920,000
	20	IZUMI	100	9,400	940,000
	21	YAMATO	100	9,600	960,000
	22	PEARL	100	9,800	980,000
	25	LONG HOPE	100	10,000	1,000,000
	28	CHERRY	100	300	30,000
	32	RAN	120	400	48,000
	36	MARL BORO	150	2,000	300,000
	37	OLD SPLENDER	150	200	30,000
				100,400	¥8,531,000
		SOUKEI		****190,300	¥16,863,000

プログラム名 URIAGE MASTER NŌ KŌUSHIN **CEC555 RPG**

作成日 50年 4月 1日　プログラマ H. NAKAWATASE　**ファイル 仕様書**　

行	A 仕様書区分 / ファイルの数	B ファイルの種類 / I/O	B ファイル名	C デバイス名:〔ファイル名〕	D	E ファイルの指定	E ファイルの最後の処理	E 配列	E 形式	F レコードの長さ	F TAB	G 行数	G 最終行	G OF標識	H コメント	I
00	FILE,4	I,	TRAN	SYS:TRAN.NA		P	,	,	,V	20						
05		I,	MAST	SYS:MAST.NA			,	,	,F	11						
10		Ō,	NMAST	SYS:NMAST.NA			,	,	,F	11						
15		Ō,	HYŌ	LPT:			,	,	,F	45			,60	,ŌF		
20																
25																
30																
35																
40																
45																
50																
55																
60																
65																
70																
75																
80																
85		INPUTファイルのTRANをSYS：TRAN. NA, MASTをSYS：ABLE, 後者をFIXとする.														
90		MAST. NAと指定し, 前者をプライマリファイル, 後者をセカ														
95		ンダリーファイルとする. またファイルの形式は前者をVARI														

OUTPUTファイルのNMASTをSYS：NMAST. NA, HYOをL
PT：と指定する. OFはオーバーフロー標識である.

プログラム名 URIAGE MASTER NŌ KŌUSHIN **CEC555 RPG**

作成日 50年 4月 1日 プログラマ H.NAKAWATASE 作業域・定数 仕様書 ページ 2/6

行	A 仕様書区分 TAB	B フィールド名 TAB	C フィールドの長さ	C 小数部桁数	D 〔定数値〕
00		WŌRK	SA	6	,Ø	
05			RITSU	6	,3	
10			SUM	8	,Ø	Ø
15			SA1ØØ	3	,Ø	1ØØ
20			HEAÐ	29		"*** TŌUGETSU-BUN URIAGE *** "
25			HEAÐA	45		"CŌÐE-NŌ NAME URIAGE-ÐAKA ZENGETSU-HI(%)"
30			KEI	13		"URIAGE-SŌUKEI "
35						
40						
45						
50						
55						
60						
65						
70						
75						
80						
85						
90						
95						

WORK AREAとして用いるフィールドのフィールド名，大きさ，初期値，ヘッダーなどを定義している．

プログラム名 URIAGE MASTER NŌ KŌUSHIN **CEC555 RPG**

作成日 50年 4月 1日　プログラマ H.NAKAWATASE　**入力仕様書**　ページ 3/6

行	A 仕様書区分	B ファイル名	C レコード標識	D レコード識別条件 位置	D 否定	D 文字	D AND/OR	E	F	G	H	I
	TAB	TAB	TAB	フィールド名	TAB			始め	長さ	(小数桁)	(制御レベル)	(突合レベル)
00	INPUT	TRAN	Ø1	1								
05				CŌÐE1				1	2	Ø	L1	M1
10				uRI1				3	6	Ø		
15				CR1				9	2			
20			Ø2	1		*						
25				AST				1	1			
30				CŌÐE2				2	2	Ø		
35				NAME2				4	1			
40				uRI2				5	6	Ø		
45				CR2				11	2			
50				XXX				2	11			
55		MAST	Ø3	1		#*						
60				CŌÐE				1	2	Ø	M1	
65				NAME				3	1			
70				uRI				4	6	Ø		
75				CR				10	2			
80												
85												
90												
95												

TRANと名づけたファイルのレコードの各フィールド名とその大きさを定義しMASTとMATCHINGするレコードに01の標識をつけ，CODE1に制御レベル（コントロールフッティング）"L1"とMATCHINGレベル "M1" をつけている．もう一方のレコード（MASTに追加するレコード）には02の標識をつけている．

ここでXXXというフィールドはCODE2からCR2までの全てのフィールドを含んでいる．また，MASTと名づけたファイルのレコードの各フィールド名とその大きさを定義し，03の標識をつけ，CODEにはM1の標識をつけてCODE1とMATCHINGさせている．

プログラム名 URIAGE MASTER NŌ KŌUSHIN **CEC555 RPG**

作成日 50年 4月 1日　プログラマ H.NAKAWATASE　**演算仕様書**　ページ 4/6

行	A 仕様書区分 TAB	B 制御標識 H1/H2/TE/GT1 TAB	C 演算ルーチン・各命令の実行条件 否定 標識名 AND/OR 否定 標識名	D 演算命令	E フィールド名1 (標識名 タグ名 リテラル)	F フィールド名2 (標識名 タグ名 リテラル)	G フィールド名3(結果) (標識 <,=,>)	H	I
00	COMPUTE	Đ	Ø1,&,ME						
05				SŌF	Ø3				
10				ADĐ	URI1	SUM	SUM		
15				SUB	URI1	URI	SA		
20				ĐIV	SA	URI	RITSU		
25				MPY	RITSU	SU1ØØ			
30				MŌVE	URI1	URI			
35		Đ	Ø1,&,#ME						
40				ĐSPLY	CŌĐE1				
45		Đ	Ø2						
50				ADĐ	URI2	SUM	SUM		
55									
60									
65									
70									
75									
80									
85									
90									
95									

DはDETAIL TIME（明細処理時間）の時にONになる標識である．MATCHINGが取れた場合には03をOFFにしADDで合計の演算を行ないSUB, DIV, MPYの各演算で前月比の計算をさせている．また，次のMOVE命令で売上げ高を転移している．ここで03をOFFにするのは次の処理サイクルで03をOFFのままにするためである．MATCHINGが取れなかった場合というのはMASTにないレコードがTRANに入っていた場合（追加という形ではなくて）なのでそのコードをタイプライタに打出し削除する．02がONのときは追加のレコードであるから合計の演算のみをさせている．

プログラム名 URIAGE MASTER NŌ KŌUSHIN **CEC555 RPG**

作成日 58年 4月 1日　プログラマ H. NAKAWATASE　出力仕様書　ページ 5/6

行	A 仕様書区分 TAB	B ファイル名 TAB	C 制御機識 HDITEIGI 改行(改頁)制御 送り前	C 送り後	D レコード・フィールド出力条件 否定 標識名 AND/OR 否定 標識名 ～	E 始め	E TAB	F フィールド名	G 編集語
00	OUTPUT	HYŌ	H	1 +1	ŌF				
05						7		HEAD	
10			H	3 +1	ŌF				
15	・HYOの出力に関して					1		HEADA	
20	・H標識がONの時に見出しを		D		Ø1 & ME				
25	打出し、D標識がONの時に					4		CŌDE	99
30	MATCHINGが取れたなら、					13		NAME	
35	CODE、NAME、URI、RITSU					22		URI	ZZZ,ZZ9
40	を出力する。					37		RITSU	ZZ9.9+
45	・02がONになる時は、追加レ		D		Ø2				
50	コードがあった場合だから					4		CŌDE2	99
55	そのまま打出す。次にT標識が					13		NAME2	
60	ONでトータルタイムでLR					22		URI2	ZZZ,ZZ9
65	がONの時にTRANの処理		T	45 +1	LR				
70	が終った時にONになる。					3		KEI	
75	には総計を打出す。					18		SUM	¥¥¥,999,999
80	NMASTの出力に関して	NMAST	D		Ø1 & ME 03				
85	・01、&、ME、03の条件によ					1		CŌDE	
90	ってMATCHINGがとれた場合					3		NAME	
95	とMASTから、そのままレコ					4		URI	
	ードを転記する場合との出力								
	をさせ、02がONの場合には								
	追加のレコードを出力させる。								

プログラム名 URIAGE MASTER NŌ KŌUSHIN **CEC555 RPG**

作成日 50年 4月 1日　プログラマ H. NAKAWATASE　出力仕様書　ページ 6/6

行	A 仕様書区分 TAB	B ファイル名 TAB	C 制御標識 H/D/T/E/G/I	改行(改要)制御 紙送り前	紙送り後	D レコード・フィールド出力条件 否定 標識名 AND/OR 否定 標識名 ～	E 始め	TAB	F フィールド名	G〜I 〔編集語〕
00	OUTPUT		D				10		CR	
05						02				
10							1		XXX	
15	END									
20										
25										
30										
35										
40										
45										
50										
55										
60										
65										
70										
75										
80										
85										
90										
95										

例題2のソースリスト

```
/ FILE MAINTENANCE
/       KOSHIN, TENKI,TSUIKA
FILE,4  I,TRAN  SYS:TRAN.VA     P,,,V   20
        I,MAST  SYS:MAST.VA     ,,,F    11
        O,NMAST SYS:NMAST.VA    ,,,F    10
        O,HYO   LPT:            ,,,F    45      ,60,0F
WORK    SA      6,0
        RITSU   6,3
        SUM     8,0     0
        SU100   3,0     100
        HEAD    29      "***  TOUGETSU-BUN URIAGE  ***"
        HEADA   45      "CODE-NO   NAME   URIAGE-DAKA   ZENGETSU-HI(%)"
        KEI     13      "URIAGE-SOUKEI"
INPUT   TRAN    01      1,##
                        CODE1   1,2,0           L1,M1
                        URI1    3,6,0
                        CR1     9,2
                02      1,*
                        AST     1,1
                        CODE2   2,2,0
                        NAME2   4,1
                URI2    5,6,0
                        CR2     11,2
                        XXX     2,11
        MAST    03      1,##
                        CODE    1,2,0           ,M1
                        NAME    3,1
                        URI     4,6,0
                        CR      10,2
COMPUTE D       01,&,ME
                        SOF     03
                        ADD     URI1    SUM     SUM
                        SUB     URI1    URI     SA
                        DIV     SA      URI     RITSU
                        MPY     RITSU   SU100   RITSU
                        MOVE    URI1    URI
        D       01,&,#ME
                        DSPLY   CODE1
        D       02
                        ADD     URI2    SUM     SUM
OUTPUT  HYO     H,1,+1  0F
                                7       HEAD
                H,3,+1  0F
                                1       HEADA
                D       01,&,ME
                                4       CODE    99
                                13      NAME
                                22      URI     ZZZ,ZZ9
                                36      RITSU   ZZ9.9+
                D       02
                                4       CODE2   99
                                13      NAME2
                                22      URI2    ZZZ,ZZ9
                T,45,+1 LR
                                3       KEI
                                18      SUM     ¥¥¥,999,999
        NMAST   D       01,&,ME,,03
                                1       CODE
                                3       NAME
                                4       URI
                                10      CR
                D       02
                                1       XXX
END
```

例題２の旧マスタファイル

```
01A415000
02B505000
03C690000
04D780000
05E675000
06F550000
07G550000
08H550000
091550000
10J715000
11K655000
12L585000
13M470000
14N456000
150367000
16P548000
17Q444000
18R950000
19S470000
20T400000
21U400000
22V400000
23W400000
24X400000
25Y400000
26Z400000
```

例題２のトランザクションデータ

```
01415000
02505000
03690000
04780000
05675000
10715000
11655000
12585000
13470000
*14N456000
*150367000
*16P548000
*17Q444000
*18R950000
*19S470000
*20T400000
*21U400000
*22V400000
*23W400000
*24X400000
*25Y400000
*26Z400000
```

例題２の実行結果

(イ) 新マスタファイル

```
01A550000
02B550000
03C550000
04D550000
05E550000
06F550000
07G550000
08H550000
09I550000
10J550000
11K550000
12L550000
13M550000
```

(ロ) 当月分売上げリスト

*** TOUGETSU-BUN URIAGE ***

CODE-NO	NAME	URIAGE-DAKA	ZENGETSU-HI (%)
01	A	415,000	24.5-
02	B	505,000	8.1-
03	C	690,000	25.4+
04	D	780,000	41.8+
05	E	675,000	22.7+
10	J	715,000	30.0+
11	K	655,000	19.0+
12	L	585,000	6.3+
13	M	470,000	14.5-
14	N	456,000	
15	Q	367,000	
16	P	548,000	
17	Q	444,000	
18	R	950,000	
19	S	470,000	
20	T	400,000	
21	U	400,000	
22	V	400,000	
23	W	400,000	
24	X	400,000	
25	Y	400,000	
26	Z	400,000	

URIAGE-SOUKEI ¥11,525,000

❾ 4Kフォートラン

1．序　論

コンピュータを用いて問題を解くためには，一般に次のような一連の段階をふまねばならない．

a．採用する手続き，つまり数式やデータの取扱い，結果の表現等を適正に決定する．

b．手続きを適当な順序に整理すること．

c．指定した操作を実行させるためのコンピュータインストラクションの列を決定する．

d．コンピュータを動作させるために，インストラクションの列を2進表現に変換すること．この際，コンピュータが読むことができるような媒体上に変換することが必要である．

プログラミングを楽に行えるように多くの研究開発を行い，上述の段階をなるべく多くコンピュータに実行させる方法が見出され，プログラマは問題そのものにより多く集中することができるようになった．

当初のプログラマはインストラクションをコンピュータコンソールを用いて直接入力するか，または直接入力するためのプログラムを用意しているのが常だった．

その後コンピュータインストラクションの集合に対するシンボリックノーテーションが開発された．そしてアセンブラとよばれるプログラムが開発された．これはこの表示法によってパンチされたテープやカードを解釈して2進表現に変換するものであった．

これらのアセンブラはシンボリックインストラクションを機械命令に翻訳したり，実行できるプログラムにアセンブルしたりする．すなわちアセンブラは段階cからの仕事ををすることができる．

アセンブラを使うためには，プログラマはコンピュータ固有のインストラクションセットをよく覚えなければならない．

同じ問題を，他のコンピュータを使って解くには，通常完全にプログラミングをやり直すことが必要である．

機械を使って問題を解決するに際して，プログラマは与えられたコンピュータの機械語命令，またはアセンブラ言語を覚える必要があるが，この厄介さからプログラマを解放するためにコンパイラが開発されたのであった．

これを用いればコンピュータに問題の原形に近い形で情報を入力することができる．

FORTRANは，そのようなコンパイラの一つである．それは数式によく似たstatementという状態で入力することができる（FORTRANは**FOR**mula **TRAN**slatorの略である）．そして指定した手続きを実行するのに必要なインストラクションの列をコンパイル（編集）する．

上記にあげた段階でいえばFORTRANソースプログラムは段階bの所産でありコンピュータが段階cとdを実行する．

FORTRANコンパイラは，ソースプログラムを実行可能なプログラムに変換するがゆえにどのようなコンピュータにでも使うことができる．つまりプログラミングをやり直す必要はない．同一のソースプログラムはほとんど変更することなく他の機械でコンパイルすることができるからである．

4K　FORTRANシステムは2つのサブシステム，すなわちコンパイラとオブジェクトタイムシステムから成り立っている．

FORTRANコンパイラは問題をFORTRANのソースプログラムで記述したものをオブジェクトプログラムに翻訳する仕事をする．

コンパイラはstatementの形式や用法の中に誤りを発見したとき，エラーメッセージ（付録G）をプリントアウトする．

プログラマやオペレータはこれによりプログラムを正しく修正する手段を講ずることができるのである．

コンパイル後，オブジェクトタイムシステムがプログラムを実行する．このシステムはインタプリータ，アリスメティックファンクションサブルーチン，インプット・アウトプットパッケージを含んでいる．

プログラムの実行が要求されると，それと一緒に働くオブジェクトタイムシステムとオブジェクトプログラムがコンピュータの中へロードされる．

ソースランゲージテープを一度読むだけでオブジェクトテープを作りだすシステムをone-passコンパイラというが，本FORTRANはone-passコンパイラである．

コンパイラはソースランゲージテープを処理し，オブジェクトプログラムテープを作成する．このオブジェクトプログラムテープは，オブジェクトタイムシステムによっていつでも簡単に実行させることができる．

2．FORTRANランゲージ

2—1　文（ステートメント）

第1図はFORTRANプログラムの一例であって注釈行，プログラムの主要部，END文とから成っている．

プログラムの最初の行は注釈行であって，このプログラムを利用する人に対してプログラムの内容を教えるために書く．これは実際に実行するプログラムには影響を及ぼさない．プログラムの主要部は，数式またはプログラムの流れを制御したり，このプログラムを正しく働かすのに必要な他の仕事を実行したりする一連の文である．

ENDはあらゆるFORTRANプログラムの最後の文として必ず存在しなければならない．この機能は，これより後にはプログラムが存在しないことをコンパイラに知らせることである．

```
C;      THIS PROGRAM CALCULATES FACTORIALS
5;      TYPE 200
10;     ACCEPT 300,X
        FACT=Y=1.
        IF(X) 5,32,30
30;     IF(X-Y) 41,32,33
32;     TYPE 400,X,FACT
        GO TO 30
33;     FACT=FACT*(Y=Y+1)
        GO TO 30
41;     PAUSE
        GO TO 5
200;    FORMAT(/,"PLEASE TYPE A POSITIVE NUMBER",/)
300;    FORMAT(E)
400;    FORMAT(/,E,"FACTORIAL IS",/E)
        END
```

第1図　FORTRAN　プログラムの一例

2—2　プログラムの形式

各行には2つのフィールドが存在する．

すなわち端から始まる最初のフィールドはidentificationフィールドである．2番目のフィールドは文を書く所である（第1図を見よ）．

identificationフィールドは左端から始まってセミコロン記号で終る．このフィールドは空白のま

まにしておくこともできるし，文番号またはコメント指定のidentificationを書くこともできる．

1. 文番号

この数は1から2047までの正の整数でなければならない．この数によってプログラムの他の部分からこの行に存在する文を照合するのである．文番号はプログラムの制御のために使用され，またプログラマが自分のプログラムの各部分を区別するのに使用される．文番号は最高40個まで使用できる．

2. 文字C

この行はコメント（注釈）であることを表わす．

（コメントの項を見よ）

セミコロンはそのstatementがstatement numberをもっているか，またはcomment statementであるときに必要である．

（もしidentification fieldが空白であるときはセミコロンはなくてもよい）

statement fieldはセミコロンの後からキャリジリターンまである．

Continuation記号（'）を使えば2つ，またはそれ以上の行を接続して1つのstatementとすることができる．しかし，2個以上のstatementを1行の中に書くことはできない．またコメント行は（'）で継続できない．

2—3 文の型

FORTRAN文は以下に示すようにその機能によって各々の型に分類される．

1. 算術代入文は代数式に似た形をしていて，実行すべき演算を記述する．
2. 制御文はプログラムの実行順序を制御する．
3. 宣言文はデータのためにストレージを割り当てたり，変数やデータの様式を決定したり入出力フォーマットを指定する．
4. 入出力文はコンピュータと外部との間の情報の転送を制御する．

2—4 コメント（注釈）

FORTRANプログラムは英語や数式記号を利用しているために，アセンブラソースプログラムより読みやすく理解しやすい．また，プログラムには豊富なコメントを付け加えて，用いた手続きを説明することは，非常に有用である．

コメントは行の最初の位置にCを置くことによって識別されるが，この行はコンパイラにより無視され，実行プログラムに影響を与えない．

```
      .
      .
C;    CALCULATE PERCENTAGE OF CORRECT RESPONSES
C;    PERCENTAGE=-1 IF THERE ARE NO ITEMS
      DO 47 I=1,57
      DO 46 J=1,6
      IF(ITMS(J)) 46,46,51
46;   PRCN(I)=-1.
      .
      .
      .
```

第2図 コメントの例

2—5 文の継続

文が長くて1行に入りきれない場合に，single quote記号(')をキャリジリターンの前に，1行の最後に書くと次の行はその文の継続として取扱われる（第3図参照）．

一つのstatementは必要なだけ行を続けても良いが，その行の字数は最大128をこえてはならない．

```
          .
          .
          .
          IF(NR(1)-1) 2,2,3
2;        AP=-14.73
          GO TO 6
3;        IF(NR(1)-2) 4,4,5
4;        AP=-44.19
          GO TO 6
5;        AP=SH(2)*3.-AG(4)/'
          AG(1)+SQFT(AG(14))
          .
          .
          .
```

第3図　文を接続した例

2—6　使用できる文字

ASCIIに属し，FORTRANで使われている記号は付録Eにリストしてある．これらの解釈可能な記号には英字AからZまでと数字0～9まで，記号　セミコロン(；)，キャリジリターン(CR)，ラインフィード(LF)，single quote (')，double quote(")，左カッコ ((), 右カッコ()), ピリオド(.)，コンマ (,)とプラス (+)，マイナス (—)，スラッシュ(／)，星印(*)，等号(=)，がある．

スペースは，FORMAT statementを除いては，すべて無視されるが，プログラムを読みやすくするために自由に使ってよい．

他の全ての記号は，FORMAT statementのHollerith informationを除いては，コンパイラは無視する（エラー12）．

3．FORTRANで用いられる演算

3—1　算術演算式

数式

$$〔5a+4b\ (x^2-x_0)\ 〕／2a\ \sin\theta$$

はFORTRANの算術演算式では次のように書く．

```
(5.*A+4.*B*(X**2-XZRO))/(2.*A*SINF(1h1A))
```

両方の式の構造は同じである．異なるのは記法である．算術演算式には定数，変数，演算子，関数の4つの要素がある．

一つの式は，定数，変数，または関数を任意の数の演算子によって関係づけたものから成る．

次の例は算術演算式の性質を示すものである．各々にはそれに対応する代数式が示してある．

	対応式
`A*X**2+B*X+C`	AX^2+BX+C
`(A**2-B**2)/(A+B)**2`	$(A^2-B^2)/(A+B)^2$
`4.*PI*..**2/3.`	$4\pi R^2/3$

```
(3.*X*PI-2(X+Y))/4.25
```

$[3\pi-2(X+Y)]/4.25$

```
A*SINF(THTA)+2.*A*COSF(THTA-.785398)
```

$A\sin(\theta)+2A\cos(\theta-0.785398)$

```
2.*SQTF(X)/3.
```

$2\sqrt{X}/3$

3—2 定　数

定数は数字だけで表現した数値である.

それらは整数, 小数点付の10進数, または10の指数付の数である. 例を次に示す.

整　数

```
        5                    -70                    2047
```

小数点付10進数

```
      18.75                3.14159                -.00025
```

10の指数付10進数

```
      1.66E-16
```

数の表現法のいろいろな形式については第4章で述べる.

3—4 変　数

変数はプログラムによって, その値を与えられる数である.

変数の名前は次の三つの規則によって1つ, またはそれ以上の文字で組み立てられる.

a. 変数の名前に使用される文字はA～Zと0～9である.

b. 最初の字は英字でなければならない. (すなわちA～Z)

c. 変数名は最初の4文字に意味がある.

それ以後のすべての文字はコンパイラによって無視される. 変数名の実例を以下に示す.

```
A        XZRO    DCSB    EPSL
R        LST8
THTA     PS1     XSUM
```

EX ITは2個でなく1個の変数を表わす (スペースはFORTRANでは無視). すなわちEX IT, EXIT, EXI T, はコンパイラに関するかぎり, まったく同一の名前である.

名前EPSILONはコンパイラによりEPSIと解釈される. すなわち最初の4つの文字だけが意味をもつのである. たとえば2つの名前XSUM1とXSUM2は同一と考えられる.

以下は正しくない変数名の例である.

```
9SHT      最初の文字が英字でない

GO(S      規則で許されていない文字が含まれている.(()

CSH$      規則で許されていない文字が含まれている.($)
```

3—5 演算子

演算子は通常算術演算を表現する記号である．FORTRANの算術演算式に関する重要な規則は“あらゆる演算は演算子によって明白に表現されなければならない”ということである．乗算記号は決して省略できないべき指数に対する記法が与えられている．この規則を説明するためにこの節の算術演算式の項で与えた代数式とそのFORTRANをここに示す．

〔5.＊A＋4.＊B＊〔X＊＊2—XZRO〕〕／〔2.＊A＊SING(THTA)〕

$[5a+4b(x^2-x_0)]/2a\ \sin\theta$

一般にFORTRANの演算式は演算子の優先度が同じであれば，代数式と同じように左から右へ演算が行われる．計算の優先順序は次のようである．

1）	括弧の中に入っている部分	（ ）
2）	符号としての－	－
3）	べ き 乗	＊＊
4）	乗 除 算	＊および／
5）	加 減 算	＋および－

符号としての－とは，－53，－GAMME，－Kのように，単にその値の符号を逆にすることを指定する演算式である．通常の－がA－Bのように自身の両側に関係するのに対照的に符号としての－は，それ自身の後にくる定数または変数にのみ関係する．符号としての－は他の演算子の直後に続くことで特徴づけられる．

A＋B＊＊－2／C－D

最初のminus（負の指数を示す）は符号演算子である．

また，二番目（減法を示す）は減算演算子である．

“左から右へ”という規則は次に示すようにより正確に述べることができる．すなわち等しい優先度の演算子の列は左から右へ演算される．演算の順序を変更するときはカッコを必要とする．

A－B＊Cは（A－B）＊Cではなく A－(B＊C）と演算される．下の実例は“左から右へ”という規則の実例である．

式	計算される順序
A/B*C	(A/B)*C
A/B/C	(A/B)/C
A**B**C	(A**B)**C

3—6 括弧の用法

次の例の括弧の使用法に注意しよう．括弧は配列Dの添字を囲むために使用されたり，A，B，Cを含む演算式の演算順序を記述したり関数の引数を囲むために使用されたりする．

D(1＋J）＝(A＋B）＊＊C＋SINF(X)

数字では大括弧〔 〕や中括弧｛ ｝のようにsubexpressionした時のlevelを区別するものがある．しかしFORTRANでは小括弧のみが使用可能であり，したがってプログラマは括弧が正しい組になっているかどうかということに対して，特に注意しなければならない．一つの式で左括弧の数と右括弧の数とは等しくなければならない．

3—7 関 数

FORTRANでは関数を一般数学で使っているように使用する．すなわち算術演算式の中の変数として使用する．関数名は計算を実行する特定のサブプログラムを表現する．

4K FORTRANでは，平方根，正弦，余弦，逆正接，指数関数および自然対数を関数として使用

できる．関数の引数は単純な変数でも添字のついた配列であっても式であってもよい．引数は浮動小数点形式でなければならない．FORTRANは括弧で囲まれた引数を後に従え，かつ最後の字がFで終っており，かつ前もって定義しておいたシンボルを関数として認識する（もしFかその項か落ちているならばそのシンボルは添字のついた配列として取扱われる）．

4K FORTRANライブラリは現在次の関数群から成っている．

関数名	その意味
ＳＱＴＦ(X)	Xの平方根
ＳＩＮＦ(X)	Xの正弦，Xラジアン
ＣＯＳＦ(X)	Xの余弦，Xラジアン
ＡＴＮＦ(X)	逆正接，結果はラジアン
ＥＸＰＦ(X)	e^x
ＬＯＧＦ(X)	$\log_e x$

3—8 算術代入文

算術代入文は記号〔＝〕により算術式Ｅの結果を変数Ｖに代入する．

Ｖ＝Ｅ

この文は数学の方程式によく似ているがその内容は異っている．

FORTRANで使う＝記号は左辺と右辺の数の間の関連を表現しているのではなく代入することを記述しているのである．

〔注　意〕算術代入文においては＝記号の右にある演算式の値が左辺の変数の値としておきかえられることを意味する．このことは左辺の変数の値が算術代入文を実行した後では変化することを意味している．

算術代入文のいくつかの実例を以下に与えて置く．

```
a. VMA=VO+AMT
b. T=2.*PI*SQTF(1./G)
c. PI=3.14159
d. THTA=OMGA+ALPH*T**2/2.
e. MIN=MINO
f. INDX=INDX+2
```

前に定義した＝記号の意味によれば例ｆでは，たとえばINDXの値が，文が実行される前に40であるなら，実行の後には42となる．

＝記号を別の角度から眺めるとその使用法をよく説明することになる．算術式の中で通常の演算子はその左右にオペランドを必要とする．算術代入文の＝記号も，また演算子の一つであると考えることができる．この演算子に関する解釈を含めて表とし，次に示しておく．

演算子	用法	解　　釈
－(符号)	－Ａ	Ａの符号をかえる．
＊＊	Ａ＊＊Ｂ	ＡをＢ乗ずる．
＊	Ａ＊Ｂ	ＡにＢを乗ずる．
／	Ａ／Ｂ	ＡをＢで除す．
＋	Ａ＋Ｂ	ＡにＢを加える．
－(減算)	Ａ－Ｂ	ＡからＢを引く．
＝	Ａ＝Ｂ	ＡをＢにおきかえる．

このように解釈する時＝記号はすべての演算子の中で最小の優先度をもつものと考えられる．このことは右辺の式が，＝によって示される演算子を実行する前に計算されることを意味する．＝記号を演算子として扱うことのもっとも重要な結果は，＝を算術代入文の中で1回以上使用しても良いということである．次を考察して見よう．

```
CPRM=(CKL-CKG)/(CPG=P*(Q+1.))
```

内部の算術式，CPG＝P＊〔Q＋1.〕が括弧により仕切られている．

この文は次に示すように通常数学上の手続きを表現するのによく用いられる型を簡単に表現している．

$$C' = \frac{C_{Kl} - C_{Kg}}{C_{pq}}$$

ただし　$C_{pq} = p\ (q+1)$

このように変数の数値を求めるのにいくつかの条件があるならば，その関係をFORTRANでは単一の算術代入文で表現することができる．

＝記号を演算子として取扱うことによりもたらされるもう一つの重要な結果は，この演算は継続して実行可能であるということである．ちょうど連続の加算A＋B＋Cがあるように，連続してA＝B＝C＝Dとおくことができる．＝記号の左の変数は変数でなければならないが，最右端のオペランド（上記の例ではD）のみは算術式であってもよい．この文は次のような意味である．すなわち式Dの値を変数Cの値とし，さらに変数Bの値は左のCの値で置きかえられる．以下同様である．換言すれば最右端の式または変数の値がそれより左にある変数の値となるのである．この構文をよく使用するのは最初の値をセットする時である．

```
XZR0=SZR0=AZR0=0.
T=T1=T2=T3=60.
P=PP=4.*ATM-AK
```

ただしA(1)＝A(2)＝R(1)＝0.123の型の文は不可能であって，正しくコンパイルできない．

＝記号を演算子として扱うことのもう一つの利点は，＝記号の右側の値は左側の変数の型に変換されることである．たとえば

A＝M　　整数型→実数型

K＝B　　実数型→整数型

4．数の表現と変数の型

4—1　整数と実数

数学においては，多くの数の範疇がある．たとえば，正数や負数，有理数や無理数，整数や分数などである．FORTRANにおいて，数の取扱い方は2種あって整数と実数に分かれ次のように分類される．

1．整数は小数点をもたない正，および負の数である．

次の数9，17，147，1024.2047は整数である．最後の数(2047)は4KFORTRANの整数として表現することができる最も大きな量である．小数や2047($2^{11}-1$）より大きな数を表現するために第2の型の数が必要である．整数の計算をする時，小数部分はすべて切り捨てられる．

たとえば

M＝N／3はN＝8ならM＝2となる．

2．実数は２つの形態をもつ．一つは0.0025,.4, 57.,2.71828のような単純小数であり，他は指数型の小数である．指数型小数は単純小数に10のべきを掛けた形のものである．

実　例

数学表現	FORTRAN表現
6.023×10^{28}	6.023E28
1.66×10^{-16}	1.66　E－１６
$72.\times10^{12}$	72E12

一般に指数型小数における浮動小数点はNE±Kの形をしている．ただしNは整数か単純小数であればよく，またKは０から99までの整数である．記法NE±Kは，数N×10^{K}を表現する．次は数19の浮動小数点のいくつかの表現である．

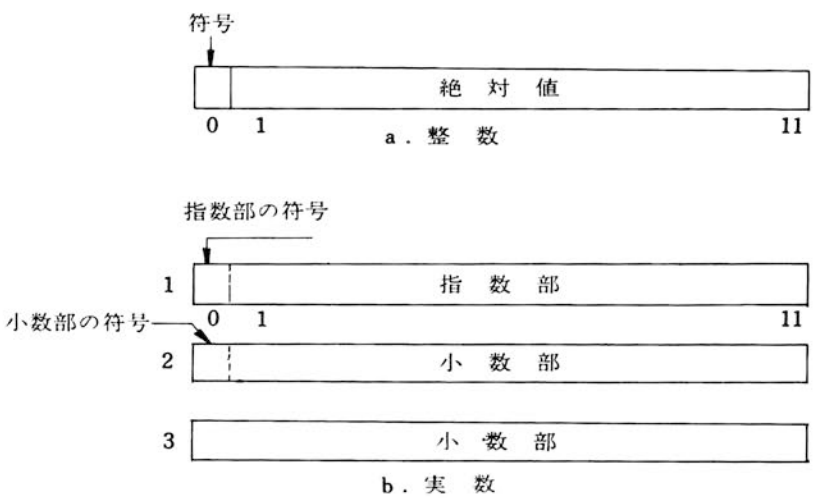
```
19.
.19E2
1.9E+1
1900E-2
190.0E-1
```

4—2　固定および浮動小数点表現

FORTRANにおける整数と実数の間の相違は，コアメモリでそれぞれを表現する方法の相違である．

FORTRANの整数は２の補数形式で12ビットの１語にストアされる．第4図で示すこの表現を固定小数点とよぶ．なぜなら小数点は最右端の数字の右に存在すると考えるからである．

FORTRANの整数は－2048から＋2047の範囲を超えてはならない．計算途中で上記の範囲をこえた場合には4096を法として取扱う（すなわち2049は－2047となり4099は３となる）．



第４図　数の表現

浮動小数点の型式は２つの部分から成り立っている．すなわち指数部と小数部である．

小数部は，最左端の数字の左に小数点が存在すると仮定した10進小数である．小数部は常に正規化されている．すなわち符号を除いた最左端のビットが常に有効数字となるように位取りを調整した形でストアされるのである．指数部分は２のべき乗を表現する．これを小数部に乗ずれば真の値が得られる．

指数部と小数部はそれぞれ２の補数形式でストアされる．

4—3　変数の型

変数は数量を表現するものであるから，何かある方法により表現の型を記述せねばならない．FORTRANでの通常のプログラミングにおいて変数の型は次のような約束にもとづいてきめられる．

1．整数の変数の名前は文字Ｉ，Ｊ，Ｋ，Ｌ，Ｍ，Ｎ，の中の一つで始まる．

2．浮動小数点の変数は上記以外の他の文字で始まる．すなわち，INDX，KDTA，M359は整数型の変数名である．

また，ZXPO，CONT，FICAは実数型の変数名である．整数はべき指数や配列の添字をのぞいて実数型の式の中に存在してはならない．べき指数は整数型でなければならない．

いくつかの式の正しい例，誤りの例を次に掲げる．

式		型
A(1)*B(J)**2	正	実数
I(M)*K(N)	正	整数
4.*J	誤	
I+L	誤	
16.*B	正	実数
(K+16)*3	正	整数
A**(1+2)/B	正	実数
B*A	誤	

5．添字付変数

5—1　配　例

配例とはデータを，一まとめにしたものである．数の列，ベクトル，リスト，行列はすべて配列である．数学において配列の要素は，配列を表わす記号とその要素の位置を示す添字によって表現するのが常である．たとえば，ベクトルの６番目の要素はV_6と記述される．

FORTRANにおいても，配列の要素は同様にして識別される．配列は第３章および第４章で述べた変数の名前のつけ方と同様の規則に従って名前をつけることができる．配列の要素を識別する添字はカッコで囲んでおく．

上記の要素はFORTRANでは次のようにして表わす．

V(6)

そのような名前を添字付変数といい計算において単純変数と同じように使用することができる．下の実例は若干の添字のついた変数の使い方を示したものである．

a. A(I+L)=A(1)+ALPH(I)*F(I)

b. A(1+3)=A(1+2)+A(1+1)/2.

c. C(J)=(I*J+3)

d. A=A(6)

5—2 添 字

上の例で見るように添字は様々な形をしている．

実際に正の整数値をとる限りどんな長いFORTRAN算術式でも添字であり得る．

5—3 Dimension 文

配列の名前は，FORTRANコンパイラにより配列の名前として識別されなければならない．

配列を使用しているプログラムにおいては次の２つの情報が与えられなければならない．

1．どれが添字のついた変数であるか？

2．添字の最大の大きさはどれだけか？

配列を使用する時，その要素のために記憶装置を必要なだけ確保しておかなければならない．

```
C;      MATRIX MULTIPLY
        DIMENSION A(36),B(36),C(36)
        ACCEPT 1,I
1;      FORMAT(I)
        DO 10 M=1,I
        DO 10 N=1,I
        INDX=M+I*(N-1)
        ACCEPT 2,A(INDX)
2;      FORMAT(E)
10;     CONTINUE

        TYPE 15
15;     FORMAT(/,/,/)
        DO 20 M=1,I
        DO 20 N=1,I
        INDX=M+I*(N-1)
        ACCEPT 2,B(INDX)
        C(INDX)=0.
20;     CONTINUE
        DO 30 M=1,I
        DO 30 N=1,I
        DO 30 K=1,I
        IC=N+I*(M-1)
        IA=K+I*(M-1)
        IB=N+I*(K-1)
        C(IC)=C(IC)+A(IA)*B(IB)
30;     CONTINUE
        TYPE 15
        DO 40 M=1,I
        TYPE 21
        DO 40 N=1,I
        INDX=N+I*(M-1)
        TYPE 2,C(INDX)
40;     CONTINUE
21;     FORMAT(/)
        TYPE 15
        END
```

第５図　２次元の添字計算

したがってこの最大値が必要なのである.

以上の情報はすべて次の宣言文により与えられる.

すなわち,

DIMENSION　A(20), B(15)

AとBは配列の名前であり, 整数20と15は各添字の最大の大きさである.

配列の名前とDIMENSION文の使い方を支配する規則は次のようである. すなわち, すべてのアレイの名前をDIMENSION文の中に書き込まなければならない. DIMENSIONは何回使ってもよいし, また, 配列を引用している文の前でありさえすれば, その配列を含むDIMENSIONはFORTRANプログラムのどこに存在してもよい.

DIMENSION　LIST(30), MAT(100), REGR(20)

上記の例の文において, 名前LISTとMATは整数型の配列を表わしている. すなわち各要素は整数である. 3番目の名前REGRは実数型の配列を表わしている. 最初の配列は30個の要素をもつ配列である. すなわち30語分の記憶装置がそのために確保されている. 実数型である3番目の配列は, その中に20個の要素がある. この配列は実数型であるから各要素は3語分を必要とし, 都合60語をこの配列のために確保しているのである.

4K　FORTRANでは, 2個以上の添字をもつ変数を用いることはできない. 2個の変数を用いるためにはプログラマは, **第5図**で説明するようにソースプログラムの中に添字計算式を使わなければならない.

この例で, 行列はメモリにコラムワイズにストアされている. すなわちメモリ内の一連のロケーションは次のようになっている.

要　素	メモリ内の相対位置 (INDX)
a_{11}	1
a_{21}	2
a_{31}	3
a_{41}	4
a_{41}	5
a_{61}	6
a_{12}	7
a_{22}	8
.	.
.	.
.	.
a_{56}	35
a_{66}	36

仮にこの配列でa_{56}を指定したいとすれば配列でM＝5, N＝6 (6×6配列に対しては) とすれば,

INDX＝M＋1＊(N－1)＝5＋6＊5＝35

また，指定したい要素をa_{22}とすれば，

INDX＝２＋６＊１＝８

となる．

６．プログラムコントロール

今まで，独立したものとしてFORTRAN statementを説明してきたかが，この章ではプログラムの中での正しい使い方について述べる．

FORTRANの文は特別の命令を与えない限り，書かれた順序に従って実行される．特別な命令は制御文によって与えられ，プログラムの実行順序を変更したり，ある部分を反復させたり，演算の実行を一時中止したり，あるいは完全に停止させたりするものである．

６―１　プログラムの終結

最後に書かれたstatementが最後に実行され，かつただ一つの停止場所であるようなプログラムにとっては特に終了の指示は必要ではない．END statementが自動的に最終停止であることを示すからである．しかしながらたいていのプログラムはループやブランチを含んでいて，最後に実行されるstamentはプログラムの中間にあることが多い．しばしばプログラムが停止する点が二つ以上あることもある．このような終結は次の文によって指示される．

STOP

これは最終的な停止であり，これ以上計算を進めることはできない．

STOPがプログラムの実行時に現われると，システムは入出力タイプライタ上に感嘆符〔！〕を印刷することによってSTOPが起ったことを表わす．

STOP文は，それが実行されるとそれ以後もはや計算は実行されない．

しかしながら演算を一時的に中止させしかる後手動でプログラムを再スタートさせる方法がある．この手続きはプログラムの実行の途中でテープを取り付けたり，はずしたりする仕事をしなければならない時，しばしば必要となる．この種の一時的な停止は次の文で与えられる．

PAUSE

これはプログラムを停止させるが，オペレータはコンピュータ・コンソールのCONTINUEを押すことによって，プログラムを続行させることができる．

またPAUSEはアセンブラ言語で書かれたサブルーチンとの連結にも用いられる（第８章参照）．

６―２　END　文

ENDは１行に単独に書く．これはFORTRANコンパイラへそのソースプログラムの終りを示す．その後にはキャリジリターンがなければならない．あらゆるプログラムはEND文を含まなければならない．

６―３　ブランチとループ

GO　TO文

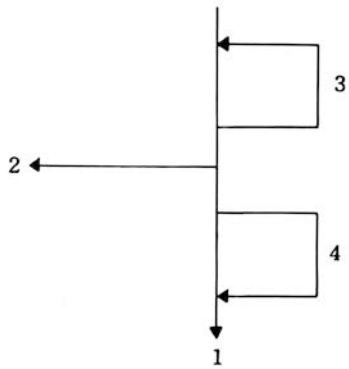
プログラムの流れを指示するには種々の方法がある．

第６図にみられるようにプログラムは直接的順序（１）であることもあるし，またそれらはまったく異った系列（２）へ枝わかれすることもあるし，前にとおった地点（３）へ帰ることもあり得るし，先の地点（４）へ飛ぶこともあり得る．

それらのブランチは文を使用していろいろな方法で実行することができる．その最も簡単なものは次のGO　TO文を用いるものである．

GO　TO　N

ここでNはプログラムで使用されている文番号である．この文の使用法を次の例で説明する．第６図№３で表われた型のプログラムブランチをループと呼ぶがこのループ構造の説明にもなってい



第6図　プログラム　ブランチの図式表現

る.

6－4　整数の和

下の実例は連続した整数の総和を計算するものである．計算部分は文2で始まる3個の命令から成るループである.

このループの前にある文は実行開始条件を与えるものである．これはイニシャライゼーションと呼ばれる．総和0にセットし，そして最初の整数に1という値を与える．そこでループは総和にその整数を加え，その整数に1を増加させ，…という具合に繰返す.

```
C; SUM OF FIRST N INTEGERS BY ITERATION
        KSUM=0
        INUM=1
2;      KSUM=KSUM+INUM
        INUM=INUM+1
        GO TO 2
```

第7図　整数の和

6－5　限界およびその決定（IF文）

前述の例で示したプログラムは必要な計算を実行するが，このループはいつまでたっても終らないという欠点がある．ループを出るために，ユーザは繰返しを止めるのはいつで，その後何をするかを知っている必要がある．IF文はこの両方の条件を与える．それは次の型で表わされる.

IF（E)K，L，M

ここでEは変数の名前か算術式のいずれかで実数でも整数でもよい．K，L，Mは文番号である．この文は次のように解釈される.

Eの値が0より小さい場合は文Kへ行く.

Eの値が0に等しい場合は文Lへ行く.

Eの値が0より大きい場合は文Mへ行く.

それゆえIF文は表現を計算することよりいつ停止するかを決定し，かつ計算の結果いかんでプロ

```
C;SUM OF THE FIRST 50 INTEGERS
        KSUM=0
        INUM=1
2;      KSUM=KSUM+INUM
        INUM=INUM+1
        IF(INUM-50) 2,2,3
3;      STOP
```

第8図　ＩＦ文を使った整数和をもとめるプログラム

グラムのブランチを選択するものである.

この例で初期条件設定とメインループは，最初のプログラムのGO　TO文をIF文により置きかえたものと全く同じである．この文はもし変数INUMの値が50より小さいか等しいならば（INUM50の値が0より小さいか等しいということと同じこと），文2へ行きそして計算を続行せよ，もしその値が50より大きいときは停止せよという意味である.

＝演算子の利用によって上記の例はさらに改良することができる.

```
C;SUM OF THE FIRST 50 INTEGERS
        KSUM=0
        INUM=50
2;      KSUM=KSUM+INUM
        IF(INUM=INUM-1) 3,3,2
3;      STOP
```

第9図　代入文を用いたＩＦ文

この例で総和は大きい方から順に加えていくようになっているが結果は同じである．初期条件設定はINUMは0でなくて50の値で始まるように変えなくてはならない．こうすれば文INUM＝INUM＋1はもはや必要ではない.

同様にループは一連の値を計算するためにも使用される．次の例はFibonacci数列の項を生成するプログラムである．この数列では各項は2つの項の和になっている.

```
C;      FIBONACCI SERIES, 100 TERMS
        DIMENSION FIB(100)
        FIB(1)=1.
        FIB(2)=1.
5;      FIB(K)=FIB(K-1)+FIB(K-2)
6;      K=K+1
        IF(K-100) 5,5,10
10;     STOP
```

第10図　Fibonacci 数列

このプログラムにおいては初期条件設定は記憶装置にスペースをとって置くためのDIMENSION文と，数列を生成するに必要な初期値をあたえるための2つの文からなっている．各項が計算されるごとに次々に引きつづいてロケーションにストアされて行くように添字がインデックスされる．添字が100に到達すると計算は終了する.

6－6　DOループ

上記の例のループと同じような繰返し手続きが必要である場合，簡潔に表現することが可能であ

る．上の例では3つの文が必要であってそれぞれ添字をイニシャライズしそれを増加させ終了のためにテストするものであった．

次の形のstatementはこれら3つの機能を全て含んでいる．

DO n　I＝K1，K2，K3

ここではnは文番号でありIは整数型の変形でありK1，K2，K3はインデックスパラメータあって順番にIの初期値，Iの最終値，Iの増分である．増分が1の場合には，K3を省いてもいい．文nはCONTINUE文でなければならない．

```
C; FIBONACCI SERIES. 100 TERMS
        DIMENSION FIB(100)
        FIB(1)=1.
        FIB(1)=1.
        DO 5 K=3,100
        FIB(K)=FIB(K-1)+FIB(K-2)
5;      CONTINUE
        STOP
```

第11　DOループを用いた Fibonacci 紙数計算プログラム

要するに上記DO文は次のことを命令するものである．すなわち文5に出合うまでの全ての文をK＝3に対して行え．つづいて，もしK＋1が100以下であるならばK＝K＋1とセットしDOの後の最初の文を実行することによりプログラムを続けよ．もしK＋1が100より大きければ文5につづく次の文が実行される．この例では次の文はSTOPである．

DOループは通常添字付変数をもった計算に使用される．

この場合一般にはループ内でループを実行することがあり得るが，FORTRANはそのようなループのネスティングを用いる．

```
        DO 10 I=1,20
        A(I)=0.
        DO 5 J=2,40,2
        A(I)=A(I)+(B(J)-Z(J))**2
5;      CONTINUE
        A(I)=A(I)**2+C(I)
10;     CONTINUE
```

第12図　ネストされたDOループ

前の例でXにおける各要素はBとZの各要素の差の平方を合計することにより求められる．

上記ループに対する数式は次のようである．

$$A_i = X_i^2 - C_i \; ; \quad i = 1, 2, 3, \cdots\cdots, 20$$

ここで

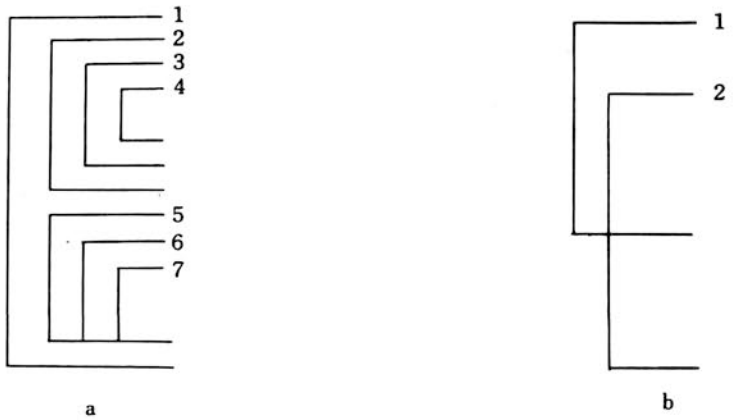
$$X_i = \sum_{j=2}^{40} (b_j - Z_j \; ; \quad j = 2, 4, 6, \cdots\cdots, 40$$

DOループにおいては次の一般的な規則を守らなければならない．

1．DOループはネストすることができる．オーバラップしてはならない．ネストされたループは同じ文で終ってもよいが，内のループは外のループの最後の文をこえて延長してはならない．

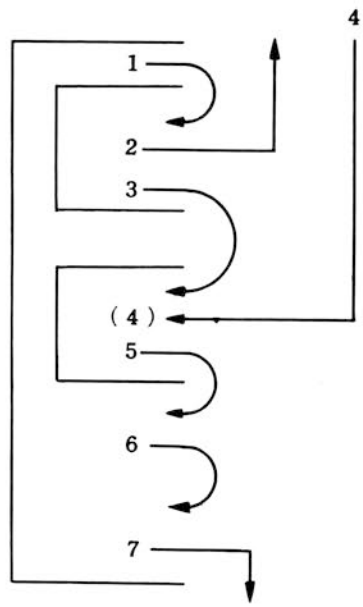
第13図は正しい例を図示したものである．

2．ユーザがDOのレンジの中へトランスファしたならば変数IはDO文において指定されたように初期設定がされない．DOのレンジの中へトランスファすることは，以下の場合にのみ許され



第13図　ループ

正しくないＤＯのネスティング



ブランチ　２，５，６，７，は正しい．

ブランチ　１，３，４，　は正しくない．

第14図　ＤＯループ中のプログラム　ブラシチ

る．

a．増加することやテストすることがＩの現在値で始まってもよい場合

b．実行順序がもともとそのDOから出てもどることによりそのDOを完了する場合

3．一つのDOループはCONTINUE文で終らなければならない．

a．の中のループは正しい．この場合ループ５，６，７は同じstatementで終っている．

b．の中のループは正しくない．ループ２はループ１の範囲の外のstatementで終っているからである．

6—7　CONTINUE　文

CONTINUE文は無意味でDOループの終りとして用いられる.

CONTINUE

計算型 DO TO

ブランチやループについての項で述べたGO TO文は無条件であって二者択一ではない. IF文は最大3つのブランチポイントを与えるものである. 多者択一を与える他の方法は計算型GO TOを使用することである. これは次の形をしている.

GO TO (K1, K2, K3, ……Kn), J

ここでは文番号でありKjは整数型変数であり, それより前の計算結果に従って1, 2, 3, ……n, の値を取る. たとえば,

IVAR＝1, 3, または5の時statement 5へ,

IVAR＝2, 4, または6の時statement 7へ,

IVAR＝7の時statement 10へbranchを生じる.

IVAR＝7

IVARが1より小さかったり, また7より大きいような時には, GO TO文のすぐ次の文が実行される.

7. 入力および出力

プログラマはプログラム本体の動作に合わせてデータの入力や結果の出力について規定する必要がある. 一般に入出力の手続きというものに対しては, いろいろの項目を指定しなければならない.

1. データはコアメモリの中のどこから読み取りあるいはどこへ読み込むのか?

   コンピュータの記憶装置の中のデータの総量とそのロケーションを指定しておかねばならない.

2. データはどんな型式で表現されているか?

   数値としてのデータである実数型および整数型の他に, 英字や文章を転送するために用いられるホラリス型がある.

最初の項目は, 入出力文により記述され, 後の項目はFORMAT文により決定される.

4K FORTRANは次の方法でプログラムから, あるいはプログラムへデータを伝達する.

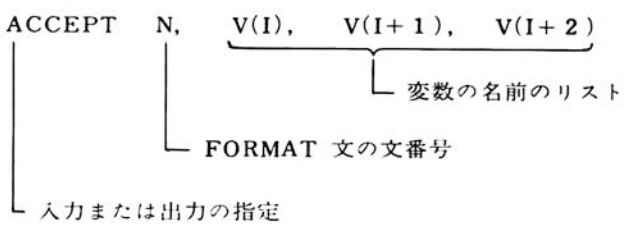
ASCII Coded Data (付録E) はデータをプログラムに転送するのに用いられる. その場合入出力タイプのキーボードにユーザが直接データをタイプするか, または前もってパンチしておいた紙テープをタイプライタのリーダから読み込むかどちらか一方が可能である.

データをプログラムからタイプライタにとり出すことができる. その場合それに対応する紙テープを作ることも, 省くこともできる (これはパンチがオンになっているかどうかで決定される).

また, 高速紙テープリーダ・パンチは, 紙テープによるデータの転送を行うものである. 出力が高速パンチの時はコピーはプリントされない.

7－1 入出力文

入出力文は情報の転送を制御するものである.



第15図 入出力 statement

第15図で図示するように入出力文は情報の３つの基本項目から成り立っている．すなわち用いるべき装置とその方向，データの配列を規定する**FORMAT**文の文番号；出力すべき値の変数名，あるいは新しく入力した値で更新すべき変数名のリストから成っている．

7－2　装置の選定と転送の行先

ACCEPTとTYPEはともにテレタイプとCEC―555の間で情報の転送を行う．

ACCEPTは情報を入出力タイプの紙テープリーダ，キーボード，紙テープリーダのいずれか１つからコアメモリの中へ受け入れるものである．

装置の選定はプログラムの実行時にスイッチレジスタのセッティングによる．TYPEは情報をコアメモリから入出力タイプのプリンタおよび紙テープパンチ（パンチがONになっているかいないかによる），または高速パンチのいずれかへ転送するものである．この場合の装置の選択はプログラム実行時にスイッチレジスタのセッティングによる．

7－3　Format　文の文番号

転送の方向を決める命令（ACCEPTまたはTYPE）に引き続いてFORMAT文の文番号がある．これは転送すべき情報の配列を規定するものである．

```
          ACCEPT  10, A
    10;       FORMAT (F)
```

あらゆる入出力文はFORMAT文を照合する．

7－4　リ　ス　ト

入出力文で次に存在する項目は，転送すべき変数やアレイの要素の名前である．これらの名前は値を転送する順序でリストに書けばよい．リストの中の変数について，その個数や型について別に制限はないが，対応するFORMAT文の指定と一致していなければならない．

```
                .
                .
                .
                TYPE 10,A,I,B,C(I+K),N(J+L)
    10;         FORMAT(E,I,E,/)
                .
                .
                .
                A=A+C(J)**2-C(N)**2
                .
                .
                .
                TYPE 10,A,J,B,C(N)
                .
                .
                .
```

第16図　リストの一例

もしリストがFORMAT文の中の要素より多くの名前を含んでいる場合は，要素がFORMAT文を使いつくした時あらためてFORMAT文の最初の要素をリストの中の次の名前に対応させる．

たとえば，前の例で変数Ｂの値をＥ形式でタイプした後制御記号のスラッシュ（/）はキャリジリターン，ラインフィードを起す．

そこからFORMAT文はあらためてイニシャライズされ，アレイ要素Ｃ（１＋Ｋ）がＥ型式でタイプされる．

もしFORMAT文の中の要素よりリストの名前が少ないならばプログラムの入出力文の動作は完結し，次に続くFORMAT文へ進行する．次にFORMAT文を照合する他の入出力文を実行する場合にはあらためてFORMAT文はその最初要素から照合される．

7－5　FORMAT　文

入出力文のところですでに述べたように，FORMAT文は転送する情報の配列と型式を規定する．

入出力文のリストの中に現われる名前の値は，FORMAT文の中の該当する要素によって示される型式で転送される．

これらの制御のための要素は記号E，I，スラッシュ（/），引用符（''）である．要素の集合はカッコの中に書き，かつカンマによって区切られていなければならない．例えば，

FORMAT（E，I，/，''HOLLERITH''）

7－6　制御要素EとI

制御要素EとIは転送すべきデータのモードを定義するために使用される．一つの変数がE型式で転送される時，それは実数型としてストアされたり出力されたりする．もし変数がI型式で転送される時，それは整数型としてストアされたりする．入力と出力の際のモードの変換が可能である．なぜならFORMAT文における要素はデータの型式を定義し変数の型を無視するからである．

例

```
    TYPE  10, A
10; FORMAT (I)
```

変数Aは整数型に変換されてタイプされ小数部分は切り捨てられる．たとえば，もしAが14.96という値をもっているならば整数部分の14がタイプされる．

7－7　入　力

入力データワードは符号，10進数，指数の値（E型のみ）およびスペースのような終結記号で成り立っている．数字，小数点，符号，E以外の記号はラブアウト記号を除いては1つのデータの終りを示すのに用いられる．

データをタイプする時，より読みやすくするためにスペースやその他の文字を符号や10進数の前にタイプすればこれらの文字は無視される．

これらの入力ワードはタイプライタの紙テープリーダ，キーボード，紙テープリーダのいずれかからコアメモリへ転送することができる．その場合整数や小数として固定小数点か，浮動小数点からのどちらかのモードで入力することができる．それらをどのモードでストアするかということはFORMAT文の該当する要素によって規定されるのである．

整数値―固定小数点―FORMAT（I）

整数のデータは符号（＋，－またはスペース），4ケタの10進数から成り立っている．整数のいくつかの実例を下に示す．

タイプされた数	入力された値
－2001	－2001
－40	－0040
－0040	－0040
16	0016
－2047	－2047

＊プラス符号は＋またはスペースによって表わすことができる．マイナス符号は－によって表わされる．

もしデータに符号が存在しなければデータは正数としてストアされる．

10進小数値―浮動小数点―FORMAT（E）

浮動小数点入力データは1つの符号＊7ケタまでの10進数によるデータの値，もし指数を含んでいるならばE（指数の記号），データに掛けるべき10の指数から成り立っている．たとえば，

dddd，dddEnn

dはデータの中の数字を現わし．nは10のベキを表わしている．符号，小数点および指数部の全体は省略することができる．もし符号が省略されているならばその数は正ということである．また，小数点を省略すればその数の小数点はもっとも右の数字の後に存在するということである．指数が省略されているならば10のベキは0ということである．

浮動小数点の値の例を次に示す．

タイプされた数	入力した値
16	0.16×10^{2}
16E02	0.16×10^{2}
1600. E−02	0.16×10^{2}

7—8 タイプエラーの修正

FORTRANプログラムの中でデータを間違ってタイプした場合，終結記号をタイプする前にラブアウト記号をタイプすることによって誤りを訂正できるリストの中にある名前に対する操作はそれまでに入力されたデータをリセットし新しいデータと終結記号を完全に受け終るまで，次に続いている名前へは進行しない．

出　力

データ　ワード出力　浮動小数点および固定小数点モード

−FORMAT（E）およびFORMAT（I）

整数の値は，符号と頭の0をスペースにおきかえた最大4桁の数として印刷される．

浮動小数点の値は，符号，頭の0.7ケタの10進数，記号E，指数の符号（−か＋），2桁の指数値から成り立っている浮動小数形式で印刷される．たとえば，

整　数　値	出力形式
−1043	−1043
−0016	− 16
+0016	+ 16

浮動小数点の値は下の例のように印刷される．

S0. ddddddEsxx

ここで，Sは符号，，符号かスペース

dはデータの7ケタの10進小数

sは指数符号

xxは指数値

出力形式	10進値
−0.8388608E+07	−8,388,608.0
0.1192092E−06	+0000001192092

7—9 他のFORMAT 制御要素

たいていの場合データは，データシート上に適当に分離したり，配列したり表題をつけたりしなければならない．これをFORTRANで行うために文章情報やスペースをデータワードの間にタイプする機能をそなえている．

これらの特徴は，特別のFORMAT制御要素である引用符（''）やslash（/）により与えられる．

引用符（''）ホラリス出力

文章情報がFORMATの一部分として含まれ，かつこの情報が引用符でかこまれているならば，それはstatement上に書いてあるそのままの形で指定された装置へ出力される．この出力はTYPE文が文章を含んでいるFORMAT statementを照合し，かつそのFORMAT中でその文章より前にあるすべての要素が使われている時に生ずる．

```
          .
          .
          .
          TYPE 10
10;       FORMAT(/,,"THIS IS HOLLERITH",/)
          .
          .
          .
          TYPE 100,AMIN,AMAX
100;      FORMAT(/,"MINIMUM=",E,/,"MAXIMUM=",E,/)
          .
          .
          .
          TYPE 210
210;      FORMAT(/,/,"    CUMULATIVE DISTRIBUTION",/,/'
          "       INCREMENTS      FREQULENCY",/)
          DO 220 K=1,100
          TYPE 250,K,VALUE(L),VALUE(K+1),COUNT(K)
250;      FORMAT(I,"      ",E,"  ",E,"  ",E,/)
200;      CONTINUE
          .
          .
          .
```

第17図　引用符と“　”の例

正当なタイプライタ記号であればすべて引用符の中に書くことができ、そして文章として出力することができる.

もう一つの文がもう一つの行にわたって続いているとしてもホラリスfieldは継続記号（′）をタイプする前に終らなければならない.

したがって、次の行であらためてホラリスfieldを開始すればよいのである. 文章が出力されるためにはホラリス情報の前に存在するFORMAT文の要素は必ず使用されねばならない. たとえば、

```
          TYPE 10,VA..,SL
10;       FORMAT(E,E,E,"VARIANCE AND STANDARD
          DEVIATION",/)
```

この例で、文章はタイプされない. なぜならばE要素の一つが使用されていないからである.

Slash（/）

slash記号は、タイプライタの用紙を進めるためのキャリジリターンとラインフィードをタイプするために使用される. FORMAT文の中に/があらわれるごとに一つのキャリジリターンおよびラインフィードがタイプされる. たとえば、

```
          TYPE 10,A,B
10;       FORMAT(/,/,/,E,/,/,E,/,/)
```

3つのキャリジリターン、ラインフィードがAの値の前にタイプされる. オブジェクトタイプシ

ステムのインプットサブルーチンはデータワードの終結記号以外すべての数でない文字を無視する.

これにより，入力データには文章やキャリジリターン，ラインフィードを間にはさんだり，スペースをあけたりめじるしをつけたりすることができるのである.

8．FORTRAN　シンボルプリント

FORTRANシンボルプリントは，コンパイルされたFORTRANオブジェクトプログラムがメモリのどこかにストアされているか，どのメモリロケーションが変数に割当てられたか，また，このFORTRANプログラムによって使われていないコアメモリ部分を見出すのに有用な助けとなる.

シンボルプリントはFORTRANコンパイラをオーバーしてロードし，アドレス600から始まっている．次はタイプアウトの例である.

Variableの名前のリスト　　割りあてられたロケーション

```
X       7574
FACT    7571
Y       7566
6154    7563
```

最後の2つの8進定数はインタープレティブメモの中でそのプログラムによって使用されたもっとも高いアドレスと，データのために使用されたもっとも低いアドレスをそれぞれ表わしている．それゆえにそれらの2つのアドレスの間の範囲はまだ使用可能である．たとえば，

7563−6154− 1 ＝1406

個のロケーションが使われていない．マシンランゲージサブルーチンはこの使われていないスペースを占めることが可能である．PAUSEを用いればFORTRANプログラムをサブルーチンに結びつけることが可能である．PAUSEの後に一つの数（10進数）を書いておけばFORTRANはそのアドレスへJMSするようにコンパイルする．たとえば，

PAUSE 3328

はJMS6400を生ぜしめる．ロケーション6400以下は次のようにコーディングされているとする.

```
SUBR,  0
       .
       .
       .
       JMP I SUBR
```

これにより希望するアセンブラプログラムを連絡することができる.

付録A．BIN　ローダの操作手順

4K FORTRANのシステムテープはバイナリ形式の紙テープとして供給されている．コンパイラ，シンボルプリントおよびオペレーティングシステムの各テープはベリファイローダあるいはその他のバイナリローダで計算機に読込んで使用する.

ベリファイローダはCEC−555の最も汎用的なバイナリローダで(1)テープラベルの入出力タイプライタへの印刷，(2)リーダの読誤りに対するチェックサムによる検査，(3)紙テープリーダまたは入出力タイプライタのいずれからでも切替なしでのテープ読込み，(4)紙テープでロードしようとする内容との比較，プリントアウト（オプション）の諸機能をそなえている.

ベリファイローダを使ってバイナリテープを読込むためには，まずベリファイローダが正しくコアに常駐していると考えてよいが，要すればHELPローダを使ってコアにロードする.

バイナリローダの操作手順は次の通りである.

1．バイナリテープ（コンパイラ，シンボルプリントまたはオペレーティングシステム）を紙テープリーダまたは入出力タイプライタのリーダ部のいずれかにセットしてCONTまたはスター

トとする．この時リーダの読取り部分はバイナリテープの見出し部分またはリーダ部分のところになければならない．

2．スイッチレジスタを7777にセット（STOP，LOAD PC）してSTARTを押せばリーダはスタートし，トレーラ部分の頭の部分で止まる．止まらない場合はバイナリローダの不良であるからローダをロードし直す．

3．この状態でACをチェックして0ならばチェックサムOK（読誤りはないと思われる）であるが，0でなければ1のステップからやり直す．

4．チェックサムOKならば演算開始番地をセットして動作を開始できる．

ベリファイローダを計算機に読込むためのHELPローダの操作手順は次の通りである．

1．HELP形式のベリファイローダを，用意し，入出力タイプライタのリーダ部にセットしてスタートとする．読取り部分はテープのラベル部分の後のブランク部分のところになければならない．

2．スイッチレジスタおよびストアのキーを用いてコンソールから27～40番地に次のブートストラップをストアする．

```
0027 /6031
0030 /5027
0031 /6036
0032 /7450
0033 /5027
0034 /7012
0035 /7010
0036 /3007
0037 /2036
0040 /5027
```

3．演算開始番地を27にセットして計算機をスタートさせると紙テープを読みこみトレーラ部分で停止する．ベリファイローダは7600～7777番地に格納された．

なお，ベリファイローダおよびHELPローダの詳細についてはローダ説明書を参照されたい．

付録B．ソーステープの準備

ソーステープはエディタを使用して作製するのが便利である．

エディタは176番地スタートでOPT：4と指定するとタブコード（CTRL I）を使って体裁良く形式を整えるのが楽にできる．タブコードはテープ出力時には，適当な数のスペースとしてせん孔される．

エディタの操作の詳細についてはエディタ説明書を参照されたい．また，入出力タイプライタをオフラインで使用してのソーステープの修正はエディタの使用が望ましい．

付録C．FORTRANの操作手順

1．バイナリローダでコンパイラをロードする（付録Aを参照）．

2．計算開始番地を200にセットする．

3．ソース入力に紙テープリーダを使用する時は，スイッチレジスタ1（SR1：2000）を1として，ソーステープを紙テープリーダにセットしてスイッチをCとする．入出力タイプライタのリーダを使用する時は，スイッチレジスタを0としてソーステープをリーダ部分にセットして

リーダスタートとする.

4．オブジェクトテープ出力に紙テープパンチを使用する時には，スイッチレジスタ2（SR2：1000）を1とする．入出力タイプライタを使用する場合にはスイッチレジスタを0としてタイプライタをパンチオンとする.

5．STARTを押す.

6．コンパイルが終ったらコンピュータは停止する.

7．さらにまた，プログラムをコンパイルするにはソーステープを正しくリーダにセットして，CONTINUEを押す．ただしI/O選択は，コンパイラをあらためてロードしなおさなければかえることはできない.

シンボルプリント

シンボルプリントは一つのプログラムをコンパイルした後，他のプログラムをコンパイルしたりオペレーティング　システムをロードする前にすぐに使用しなければならない.

a．バイナリローダでシンボルプリントをロードする.

b．計算機を600番地からスタートさせる.

c．印刷内容について8章参照のこと.

オペレーテイング　システム（オブジエクトタイム　システム）

1．コンパイルされたオブジェクトプログラムのロード

a．バイナリローダを使用してFORTRANオペレーシィングシステムを読み込む.

b．計算開始番地を200にセットする.

c．入出力タイプライタ，あるいは紙テープリーダ上にコンパイラアウトプット（オブジェクトテープ）をセットし，スイッチレジスタ1（SR:2000）を0（入出力タイプの場合），または1（紙テープリーダの場合）にセットする.

e．STARTを押すとコンパイラのアウトプットが読みこまれる.

f．ローディングが終ると計算機は停止する．ACに表われるチェックサムが0であればローディングは正しく行われたことになる.

2．オブジェクトプログラムの実行

a．入力データをキーボードまたは入出力タイプライタのリーダ部分からいれる場合にはスイッチレジスタ1（SR:2000）を0とする．紙テープリーダからデータを入力させる場合は1とする.

b．出力データを入出力タイプライタで印刷またはパンチする場合にはスイッチレジスタ2（SR:1000）を0とする．紙テープパンチでパンチをする場合は1とする.

c．CONTINUEを押す.

〔注意〕オペレーティングシステムで2つ以上のプログラムを継続して実行させるには（ステップ1b）から再スタートすればよい.

3．ロードされたプログラムの再実行

〔注意〕一度ロードしたらプログラムは何回でも実行できる.

a．演算開始番地を201にセットする.

b．入出力に関してスイッチレジスタを（2a）（2b）項に準じてセットする.

c．STARTを押す.

FORTRANオペレーティングシステムはスタックオーバーフローもスタックアンダーフローも起らないようにするために，各々のインストラクションの実行後に内部のスタックの各々をチェックする．もしFORTRANプログラムがもうすでに充分デッバグされて確実に実行できることがわかっていれば，このテストは400番地の内容を7000（NOP）と変えることにより

省略される．このことはプログラムの処理速度を約２倍にはやめることになる．

付録D．オブジェクト　テープ（コンパイラ出力）の型式

1．200コード（リーダ，これはローダにより無視される）
2．データ　ブロックはそれぞれ次のようである．
   a．オリジン（２フレーム，最初のものはビットポジション７がパンチされている．）
   b．データワード（２フレーム／ワード）
3．アドレステーブル：最初のフレームはビットポジション７と８だけパンチされている．
4．チェックサム
   a．最初のフレームはビットポジション６，７，８だけパンチされている．
   b．次の２つのフレームはチェックサムである．
5．200コード（トレーラ，チェックサムの後でローダを停止させる）
6．エラー　コメントがもしあればASCIIで表現される．

付録E．タイプライタ８ビットキャラクタ　セット
入出力タイプライタ(テレタイプ)

キャラクタ	８ビットコード（８進）	キャラクタ	８ビットコード（８進）
A	301	!	241
B	302	〃	242
C	303	#	243
D	304	$	244
E	305	%	245
F	306	&	246
G	307		247
H	310	(	250
I	311	)	251
J	312	*	252
K	313	+	253
L	314	'	254
M	315	−	255
N	316	.	256
O	317	／	257

P	320	:	272
Q	321	;	273
R	322	<	274
S	323	=	275
T	324	>	276
U	325	?	277
V	326	@	300
W	327	〔	333
X	330	／	334
Y	331	〕	335
Z	332	↑	336

キャラクタ	9ビットコード（8進）	キャラクタ	8ビットコード（8進）
0	260	リーダ/トレーラ	337
1	261	ライスフィート	200 *
2	262	キャリジリターン	212 *
3	263	スペース	215
4	264	ラブアウト	240
5	265	ブランク	377 *
6	266		000 *
7	267	*コンパイラにより無視される.	
8	270		
9	271		

付録F. 4K FORTRANソース　プログラムの制限条件

a. 変数および定数によるコアの使用量が896語より多くないこと. これはDIMENSION のついた変数, ユーザによって定義された変数, 定数, DOループ定数などすべてを含める.

b. 一つのプログラムの中で未定義文番号に対する参照が20個以下であること. 未定義文番号とはそのプログラムで, 引用する部分までにまだ文番号フィールドに出現していない文番号のことである.

c. 一つのプログラムで使用できる変数名64より多くないこと.

d. 一つの命令文について文字数が128より多くないこと（ただしスペースは, FORMAT 文以外ではカウントされない).

e. 一つのプログラムにつき文番号は数は40より多くないこと.

コンパイラは次のコアロケーションを占める.

3——7600　コンパイラそのものとそのテーブル

7200——7600　コンパイラ　テーブル（アンデファインド　フォワード　リアァレンス　テーブル等)

オペレーティングシステムは次のロケーションを占める.

0——5200　紙テープ入出力のためのオペレーティングシステム

ロケーション5200—7576は紙テープ入出力を使用する時ユーザのプログラムのために使える.

付録G. エラーメッセージ

プログラムのコンパイル時に，プログラマを助けるための診断ルーチンがコンパイラの中に用意されている．コンパイラがFORTRANソースプログラムの中にエラーを発見すると，入出力タイプライタにエラーメッセージを印字する.

これらのメッセージはエラーの原因や内容を指摘してプログラマにエラー訂正の手がかりを与える．コンパイラの過程の進行を促進するためにコンパイラはエラーコードだけをプリントする．そこでプログラマは表A—1のコードに対応するエラーメッセージを参照して適当な訂正手段をとることになる.

コンパイル中のエラー訂正

ユーザは2つの方法，すなわちノーマルモードまたはダイナミックコレクションモードのいずれか一方でコンパイルすることを選択できる．後者ではソースエラを含む行をコンパイル中に訂正できる．この機能は高速紙テープリーダを用いた場合には利用できない.

ダイナミックコレクションモードを選択するためには，

1．コンソールスイッチにコンパイラのスタートアドレス（0200）をロードする．そしてLOAD ADDRESSを押す.

2．SRのビットポジション11を1にセットしSTARTを押す（低速紙テープ入出力でのみ使用可).

もしエラーが発見されたら診断ルーチンはノーマルモードの場合と同じ方法でエラーメッセージをプリントしコンピュータは停止する.

文を訂正するためには

1．READERスイッチをFREEに切り変える.

2．READERスイッチをFREEにしたままCONTINUEを押す.

3．1行全部を全く新しくタイプし直す．その場合もちろんソースランゲージに対する全ての規則に従いキャリジリターンラインフィードでstatementを終了する.

4．リーダをONに切り換える．そうすればコンパイルは続行して行われる.

ダイナミックコレクションモードの状態から抜け出すためには，ノーマルモードでコンパイルを再スタートしなければならない.

＊その文が文番号をもっていればそれが誤りでない限り，文番号を新しく入れ直してはならない.

コンパイル時のエラーメッセージ

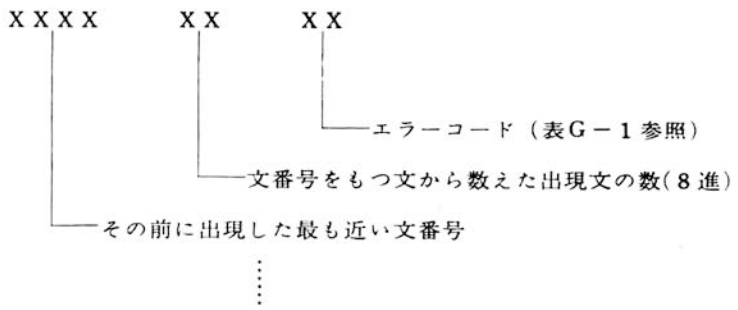
メッセージの形式

XXXX　　XX　　XX

エラーコード（表G－1参照）

文番号をもつ文から数えた出現文の数(8進)

その前に出現した最も近い文番号



⋮

例

```
10 ; A = I (J + I)
     B = A * (B + SINF(THTA)
```

⋮

上記のプログラムをコンパイルすれば次のエラーコードがプリントされる.

10　11　11

これは文10があるところから数えて8進で11個目（10進で9個目）にある文の中にエラーがあることを示している. コード11はその文の中で左カッコの数と右のカッコ数が等しくないことを示す.

表G－1

診断コード	条　　件
00	• 固定小数点モードと浮動小数点モードとが一つの表現の中で混在している.
01	• 2つの演算子が互に隣接して現われている. たとえば　A＝C＋＊D
02	• 原因不明. コンパイラをロードし直す.
03	• 算術文での中でコンマが使われた.
04	• 1つの文の中で演算子の数が多すぎてコンパイラが処理できない.
05	• 関数の引数として不当に固定小数点モードが使われた. たとえばSINF（INC）
06	• 配列の添字に浮動小数点モードが使われた.
07	• 1つのプログラムの中に変数名が64個以上使われた.
10	• プログラムが大きすぎてコア割当てができない.
11	• 1つの文の中で左右のカッコの数が等しくない.
12	• FORTRANで許されていない記号が発見された. コンパイルにはこれを無視する.
13	• 文中のエラーのためにコンパイラはこの文を処理できない.
14	• 文番号の重複.
15	• 添字付変数がDIMENSION 文の前で使われているか, DIMENSION 文に定義されていない.
16	• 文が長すぎる. すなわち1つのstatementが128キャラクタ以上になっている. この場合FORMAT文の中にあるスペース以外はスペースを勘定に入れない. FORMAT文の中ではすべてのTTYキャラクタを勘定する.
17	• 固定小数点モードでなければならないところに浮動小数点モードが使われた.
20	• その文番号がプログラムの中にない. 引用された文番号をXXXXとするとエラープリントはXXXX77　20となる.
21	• 文番号が40個以上使われた.
22	• 1つの文の中に完結していない演算が多すぎる. たとえば　C＝A＋（C＋（D＋（E＋……
23	＊定義される前に照合されている文番号が多すぎる.

オペレーティングシステム　エラーメッセージ

プログラムの実行中に正規には実行できないエラーが起った時，コンピュータはエラーメッセージを印刷して停止する．CONTINUEを押せば次の表に定義されているように動作を続行する．

表G－2

エラーコード	原　　因	動　　作
11	• 0での計算	• 商は，コンピュータで表現できる最大数（±）にセットされる．しかる後，次のインストラクションへ進む．
12	• 入力した浮動小数点の指数部が±2047より大きい．	
13	• 規則にないオペレーションコード（コンピュータエラーか，またはデータがプログラム部分まではみ出してストアされたか，あるいはデータセクションにトランスファーされたかのいずれか）である．	
14	• コアーロケーションの0番地，あるいは1番地にトランスファーした．	• ハードウェアのエラー？
15	• FORMATとしてFORMAT statementでないものが使用された．	
16	• FORMAT statementの誤り．	
17	• 浮動小数点数を固定小数点数に変換する時のオーバフロー（2048以上）	
20	• 負数の平方根	• 絶対値の平方根を計算する．
21	• 負数のべき乗	• その絶対値のべき乗を計算する．
22	• 0や負数の対数	• その絶対値の対数を計算する．
76	• システムによって使用されているスタックの1つがアンダーフローである（すなわちスタック上に配置されたデータにより多くのデータが要求されている）．	
77	• スタックの1つがオーバーフローである（コンピュータの記憶装置に存在するより多くのデータがスタックの上に置かれている）．	＊エラー76と同様

⑩ マークIIフォートラン

1. MARK II FORTRAN

FORTRANはFORmula TRANslatorの略であって，FORTRAN言語（文法）と翻訳言語，即ち，コンパイラの二つの意味がある．

MARK II FORTRANコンパイラは，プログラマが，MARK II FORTRAN文法に従って作成したプログラムを，計算機の受け入れることのできる言語に翻訳するプログラムである．

従って，問題を解くためにこのプログラムを使うことにすれば，プログラマは問題をMARK II FORTRAN言語を用いて書きあらわせばよいが，MARK II FORTRAN言語の表現は，英文字や数学で用いる表式が用いられているので，人間が日常使用する表現に非常に類似した表現で問題を書き表わすことができる．

このようにして，FORTRAN言語で書いたプログラムをソースプログラムといい，これは次の手順でコンピュータにロードされ，問題がコンピュータで解かれることになる．

まず，ソースプログラムはコンパイラによって，シンボル言語のプログラムに翻訳される．次にこのシンボル言語のプログラムは，RAPアセンブラによってコンピュータ自身の言語である機械語のリロケータブルバイナリコードに変換される．最後に問題の答を得るために，リロケータブルバイナリコードのプログラムをコンピュータに入力させ，問題を解く．

MARK II FORTRANシステムは次の特長をもっている．

1．サブルーチン
2．2次元の添字
3．関数副プログラム
4．入出力プログラム
5．リンキングローダによりロードされるリロケータブル出力
6．COMMON文
7．I，F，E，A，X，H　FORMAT仕様
8．算術演算と三角関数ライブラリサブルーチン

MARK II FORTRANシステム（以後FORTRANと呼ぶ）は，1パスFORTRANコンパイラ，RAPアセンブラ，リンキングローダおよびサブプログラムのライブラリから構成されている．

このFORTRANシステムは，最少2フィールドのコアメモリ容量のコンピュータの外に，入出力タイプライタ，高速紙テープリーダとパンチャが必要であるが，磁気ドラム，磁気ディスクなどの外部記憶装置を含むシステムでは，紙テープリーダ，紙テープパンチャはなくてもよい．

1—1　コーディング

FORTRANプログラムの一例を図—1に示す．プログラムは図のような，コーディング用紙（コーディングシートと呼ばれる）に書いた方が見やすく，ソーステープを作る場合にも間違いが少ない．コーディング用紙の1行は80桁である．桁をcol.と略称して呼ぶことにする．

1—2　行フォーマット

FORTRANでは1行はcol.1からcol.72までであり，1行に書かれるデータは72文字以下でなければならない．各行は，図—1に示すように，3つの欄からなり，**ステートメント番号欄**，**行継続欄**と**ステートメント欄**がある．

1—2—1　ステートメント番号（文番号）

FORTRAN CODING FORM

DECK ID. 73 75 CEC 555

C FOR COMMENT STATEMENT NO (1 2 5)	CONT OR (6)	FORTRAN STATEMENT (7 11 21 41 61 72)	DECK ID (73 75 76 80)
C		THIS PROGRAM WILL SORT AN ARRAY OF NUMBERS INTO ASCENDING ORDER.	
C		FIRST READ THE NUMBERS, THEN SORT THE NUMBERS,	
C		LAST, WRITE THE NUMBERS IN ORDER.	
C			
		DIMENSION A(100)	
		ND=2	
		N=100	
		DO 10 i=1,N,2	
10		READ (ND,12) A(i),A(i+1)	
12		FORMAT(2E12.0)	
		DO 30 K=2,N	
		J=K-1	
20		IF(A(J)-A(J+1))30,30,32	
22		TEM=A(J)	
40		DO 42 i=1,N,2	
42		WRITE (ND,44) A(i),A(i+1)	
		STOP	
44		FORMAT(2E16.8)	
		END	

Remarks :

図 1

あるステートメントをプログラムの他のステートメントの中で参照するために，ステートメントに番号を付けることができる．これをステートメント番号という．ステートメント番号は，col.1～col.5のステートメント番号欄に1～2047の自然数で記入する．

この場合，必ずしもステートメントの順序に従わなくてもよいが，同一の**ステートメント番号**を他のステートメントに二重に用いることは許されない．ステートメント番号に引続き，TABキーを用いて，TABの直後からステートメントを書くことができる．またステートメント番号を書かない場合もTABキーを用いれば，TABキーの直後からステートメントを書くことができる．

1—2—2　行継続欄

ステートメントが長くて，1行のステートメント欄に納まらない時は，文を完結するために，必要なだけ行を続けることができる．この場合は，継続する行のcol.6に0以外の英字または数字を記入する（番号欄にTABを使う場合はTABの直後に1～9までの数字）．最初の行，および行を続けない場合の行は，col.6に0を記入するか，あるいは空欄のままにする．

1—2—3　FORTRANステートメント

ステートメントは，各行のcol.7～col.72のステートメント欄に書くが，TABキーを使う場合は，TABの直後から書くことができる．個々の一単位のステートメントはそれぞれ別々の行に書かなければならない．ホラリス欄の中のデータを除いてスペースは無視されるので，ステートメントの文を見やすくするために，スペースを自由に用いてもよい．

FORTRANで許されるステートメントには，次の5つの型がある．

1．算術ステートメント；行うべき計算を定義する．
2．制御ステートメント；プログラムの中で，ステートメントの実行の順序を制御する．
3．入力/出力ステートメント；入出力装置とプログラムとの間の直接の連絡を制御する．
4．仕様ステートメント；プログラムに納められているデータの内容と形を記述する．
5．副プログラムステートメント；副プログラムやサブルーチンの形式や引用を定義する．

上記の，各ステートメントについて，それぞれの章で詳しく述べる．

1—3　コメント（注釈文）

プログラムの中で，プログラムの実行とは関係のないプログラムの標題をつけたり，注釈を書き入れたい時が多くある．この場合は，行のcol.1に英文字Cを記入すると，その行（col.2からcol.72まで）に書かれるステートメントはコメント（注釈文）となる．**コメントステートメント**は，機械語に翻訳するコンパイルの過程では無視される．コメントの行の数には制限はない．

1—4　文　字

FORTRAN語で使用できる文字．

1．英文字
　　A B C ……X Y Z
2．数字
　　0 1 2 …… 9
3．その他の記号

!	′	=	・
″	(	#	,
$	)	;	<
%	+	:	>
&	—	?	BLANK (SPACE)
*	／	CR	TAB

2. 言語の要素

定数および変数を定義する規則と式を作る規則について以下述べる.

2—1 定　数

定数は，プログラマ自身がステートメントの中に書き入れた「数値」をいう．FORTRANのソースプログラムでは，**整数型**と**実数型**の2つの型だけが許される.

2—1—1 整数型定数

整数（固定小数点）型定数は，小数点をもたない10進数であって，1～4桁の−2047～＋2047の範囲の値でなければならない．＋符号はつけてもつけなくともよい.

例　47，−2，−0（＝0）

＋47　0434（＝434，先頭の0は無視される）

2—1—2 実数型定数

実数型定数は，次の2つの型で表わすことができる.

1．小数点をもった10進数（基本実定数という）

例　＋4.52，0.23，17.

2．基本実定数と10のベキ乗を示す整定数の指数の組合せ

例　7.2×10^{3}は7.2E＋03と書く.

2.5E−7（2.5×10^{-7}）

◎実数型定数の有効数字は8桁である.

◎実数型定数は0.14×10^{-38}～1.7×10^{38}までの範囲の値でなければならない.

例　＋4.50（＋符号は任意である）

4.50

−23.09

−3.0E14（-3.0×10^{14}）

2—1—3 ホラリス定数

ホラリス定数は，1～6字の英数字（スペースを含む）を，′記号で囲んだもので，扱いは実数形となる．ホラリス定数は＝記号の右側に使って実数形変数に値を代入すること，およびサブルーチンの引数として用いることができる.

2—2 変　数

変数は，その値が，プログラム実行中に変るものである.

1．**変数名**——変数の名前は，1あるいはそれ以上の英文字，数字で表わされる.

◎ただし，最初の文字は，英文字でなければならない.

◎文字の個数はいくつでもよいが，最初の5文字だけが，名前として受け取られ，それ以上は無視される.

2．変数の型（整数型と実数型）は**変数名**の最初の文字で決められる．最初の文字が，I，J，K，L，M，Nである変数は**整数型**である．これ以外の他の文字は**実数型変数**を表わす.

3．**変数**はスカラ，あるいは配列変数のいずれにもなれるが，配列変数である場合には，**DIMENSIONステートメント**で，使用前にあらかじめ定義しておかなければならない.

2—2—1 整数型変数（整変数）

1．**整変数**は，その値は小数点をもたず，算術計算では，小数部分が自動的に切捨てられる.

例　K＝5，J＝9とすると，J/Kは1となる.

2．**整変数**の取り得る値は，−2047～＋2047である.

3．**整変数**は**関数FLOAT**，あるいは**算術ステートメント**で**実数型変数**に変換することができる.

2—2—2　実数型変数

1．**実数型変数**の名前の先頭の大文字は，Ｉ，Ｊ，Ｋ，Ｌ，Ｍ，Ｎ以外の文字でなければならない．

2．**実数型変数**は**関数IFIX**，あるいは**算術ステートメント**で**整数型変数**に変換することができる．

2—2—3　スカラ変数

スカラ変数は，単一の量を表わす．整数型，実数型のどちらもとれる．

例　LM，
　　A
　　G 2
　　TOTAL
　　J

2—2—4　配列整数

1．通常ベクトル（一次元配列）や，行列（２次元配列）を**配列**という．この配列を構成する各々の要素を**配列要素**と呼び，**配列名**に（　）でくくった**添字**をつけたものである．例えば，行列のM行，Ｎ列の要素は，Ａ（Ｍ，Ｎ）と書く．

2．**配列変数**は，ある量の変数の配列の要素をいう．配列要素である配列変数は(1)のように，**配列名**に（　）の中の**添字**をつけて表わす．

3．**添字**は，１あるいは２個以上の整数型の式である．２個以上の場合はコンマで区切られる．

◎ここで使用する整数型の式とは整数型変数，あるいは，整数型定数を算術的に結合したものである．

◎式の値は添字を示し，参照する配列要素を選び出す．

例　Y(1)
　　RORT（K）
　　A（3＊K＋2，1）

4．上の配列（Y，RORT，A）は実行ステートメントの中で使われる前に，あらかじめ**DIMENSIONステートメント**で定義しておかなければならない．

◎**DIMENSIONステートメント**は配列の要素の個数を定義し，配列要素が格納される記憶場所をコアメモリに確保する．

◎配列要素がメモリに格納される順序は，二次元配列Ｂ（J,K）では次の通りである．

例　B（J,K）
　　B（1,1）　B（2,1）……B（J,1），B（1,2），B（2,2）…B（J，2）
　　……B（J,K）

2—3　式

算術式は，数字の慣習と以下に述べる規則に従い，定数や変数および関数の引用形代数演算子とカッコを使って結合したものである．

1．（　）のない算術式では，次の優先順位で算術演算は実行される．

1．＊＊　　ベキ乗
2．－　　負の符号
3．＊と／　掛算，割算
4．＋と－　加算，減算
5．＝　　置換（代入）記号

◎（　）の中の演算がはじめに実行される．（　）を用いることにより，演算の優先順位を変えることができる．

◎同順位間での演算は，左から右へ順に実行される．ベキ乗の計算でも同様である．

2．**算術式の形式**

◎整数および実数は，いずれも整数のベキ乗および実数のベキ乗が可能である．

A**B

は，算術式A^Bを意味し，A，Bが共に整数でなければ実数である．

◎指数e^xと自然数log e（x）の関数は副プログラムとして与えられている．

（付録E参照）

◎**（ベキ乗）を除いて，2番目の演算記号が＋，あるいは，－でない限り，2つの算術演算記号を連続して続けて用いてはならない．

◎算術式の型には整数型，あるいは実数型のどちらか一方だけが許され，次の例外を除いて，一つの算術式の中に整数，実数型が混在してはならない．

(1) 実数型変数は，整数ベキ乗をとることができる．

A**2

(2) 関数IFIX，またはFLOATを用いて，型を変換することが出来る．

A*FLOAT（I）

◎全ての算術式は（ ）でくくることができて，（ ）でくくられたものは，一つの基本要素として扱われる．

例 IFIX（X＋Y）/2

(ZETA)

(COS (SIN (PI*EM) ＋X))

◎算術式はただ一個の要素（定数，変数，関数の引用形）で構成することもできる．

例 2.71828

Z（N）

TAN（THETA）

◎演算記号を用いて，基本的要素を結びつけ，複合した算術式を作ることができる．

例 X＋3.

TOTAL/A

TAN（PI*EM）

◎＋，あるいは，－の符号を前につけた英文字の式も，算術式である．

例 －X，

－（ALPHA*BEAT）

－SQRT（－GAMMA）

◎算術演算記号及び関数の引用形を用いている典型的な算術式の例として，2次方程式の最大根

$$\frac{-b+\sqrt{b^2-4ac}}{2a}$$

をもとめる式をあげよう．これは，

（－B＋SQRT（B**2－4．*A*C））/（2．*A）

と書かれる．

2—4 ライブラリ副プログラム

FORTRANは，次の関数ライブラリを持っており，これらは単に引数を持った関数名を書くことによって，算術式の中に自由に使用することができる．

EXP（X） e^x

ALOG（X） log e X

SIN（X）　sin X　Xはラジアン
COS（X）　cos X　〃
TAN（X）　tan X
ATAN（X）　$\tan^{-1}X$　結果はラジアン
SQRT（X）　$\sqrt{X}$
IRDSW（　）スイッチレジスタの値
ABS（X）　絶対値（実数）
IABS（I）　〃　（整数）
FLOAT（I）実数に変換
IFIX（X）　整数に変換
IREM（　）　直前の整数割算の余り
IREM（I／J）　I／Jの余り

ライブラリ関数は，プログラムの実行時に主プログラムにつけ加えてコアにロードする．主プログラムで何回引用しても，ロードされる副プログラムは，コアの1つの位置を占めるだけである（閉じたサブルーチン）．

補助メモリを含むシステムでは，ライブラリ関数のコアへのロードは完全自動であって，実行時に必要な関数だけが選択される．紙テープベースのシステムでは，使用者が関数ライブラリの紙テープの中から，必要な部分を選択してロードする必要がある．

FORTRANシステムでは，上記関数以外の四則演算なども，閉じたサブルーチンとして関数ライブラリの中に含まれており，実行時に必要な部分がロードされる．

整数形演算は，2の補数表示で12ビットのモジュロがとられる．オーバフローは起ってもそのまま継続して実行される．

実数形演算では，オーバフローはエラーメッセージを入出力タイプライタに印刷して演算を打切る．実数形演算の精度は10進7桁（27ビットおよび符号ビット）である．実数形関数の精度は10進6桁である．

3．算術ステートメント

FORTRANの算術計算は，算術ステートメントで定義される．

算術ステートメントは，次の形を取る．

```
        V＝e
V；変数名，あるいは，配列要素名
e；算術式
＝；代入の演算記号
```

1．等号（＝）は，右辺の算術式を計算し，その結果を左辺の変数名Vの値とすることを意味する．
　例　A＝A＋B，　A＝A＊B
は，まずA＋B，1＊Bを計算し，この値をAの値とする．即ち，
　（A＋B）→A，（A＊B）→A
ここで，変数Aの値は演算終了後変化している．

2．右辺の算術式の値は，左辺に代入する前に左辺の変数の型（整数型あるいは実数型）に合わせられる．
　例　META＝W＊（ABETA＋E）

METAは整数型，右辺は実数型算術式であるから，右辺の値は，整数型に変換されてMETAに代入される．

3．**使用例**

```
Y＝1.1*Y
P＝X**2＋3.*X＋2.0
X（N）＝EN*ZETA*（ALPHA＋EM/PI）
```

A＝B　　；B→A

I＝B　　；Bの小数点以下を切捨て，整数型に変換してIに代入

A＝I　　；Iの内容を実数型に変換して，Aに代入

I＝I＋1；I＋1をIに代入

A＝5.0*B；（5.0×B）をAに代入

4．制御ステートメント

FORTRANによりコンパイルされたプログラムの実行順序は，通常は，各ステートメントが，コンパイルされる順である．しかし，**制御ステートメント**を用いることにより，プログラムの実行順序を変更することができる．このような制御ステートメントは，次のものが許される．

1．GO TOステートメント
2．IF　　〃
3．DO　　〃
4．PAUSE　　〃
5．STOP　　〃

4—1　GO TOステートメント

GO TOステートメントには，次の二つがある．

1．無条件GO TOステートメント
2．計算型GO TO　　〃

4—1—1　無条件GO TOステートメント

無条件GO TOステートメントは，

```
GO TO　n
n；実行可能なステートメントのステートメント番号
```

の形で与えられる．

このステートメントによって，実行ステートメントのステートメント番号nに飛び，そこから順次プログラムを実行する．

4—1—2　計算型GO TOステートメント

計算型GO TOステートメントは，

```
GO TO (n1, n2……nk), J
n1, n2……nk；ステートメント番号
J；添字なしの整数型変数
```

の形である．

このステートメントの指示の内容は，『Jの値が1, 2, ……，kのいずれかを取るかによって，そ

れに対応したn_1, n_2……n_kのステートメント番号に飛び，そのステートメントから順次プログラムを実行する』である．

例　ステートメント

GO TO (20, 10, 5), K

は　K＝1であれば　ステートメント番号20

K＝2　〃　〃　10

K＝3　〃　〃　5

へ各々飛ぶ（ジャンプ）ことを指示する．

4—2　IFステートメント

IFステートメントは，次の形をもつ．

IF (e) n_1, n_2, n_3
n_1, n_2, n_3：実行ステートメント番号
e；算術式もしくは変数

このステートメントの指示の内容は，次の通りである．算術式eの値に対応して，

e＜0　ならば　ステートメント番号　n_1

e＝0　〃　〃　n_2

e＞0　〃　〃　n_3

の実行ステートメントにジャンプする．

例　IF (ETA) 4, 7, 12

IF (KAPPA—L(10)) 20, 14, 14

4—3　DOステートメント

あるプログラムを反復して何回も実行させようとする場合，DOステートメントを用いると，プログラムの組立は簡単になる．

DOステートメントは，次の形である．

DO n　i＝m_1, m_2, m_3
あるいは
DO n　i＝m_1, m_2
n；ステートメント番号
i；整数型変数
m_1, m_2, m_3；整数型定数か添字なしの整数型変数

DOステートメントは，そのすぐ後に続くステートメントからステートメント番号nまでのプログラムの実行をくり返し行うものである．

◎m_3が定義されていない場合は，m_3＝1とみなす．

◎iをインデックスといいm_1, m_2, m_3をそれぞれインデックスiの初期値，終値，増分と呼ぶ．$m_2 \geqq m_1$, $m_3 > 0$でなければならない．

例　DO　10　I＝J, 5, 2

DO　20　I＝J, K, 5

DO　100　I＝J, K, L

1．**DOの範囲**

DOステートメントの次のステートメントとDOステートメントで指定されたステートメント番号nまでの範囲にある全ての実行ステートメントの集りをDOステートメント範囲という.

2．DOステートメントの実行は次の3つの過程で行われる.

(1) DO範囲が実行される前に，インデックスiに初期値m_1が代入される．（i＝m_1）

(2) 以後増分m_3ずつインデックスiを増加する．（i＝i＋m_3）

iが終値m_2を越えない最大数まで，DO範囲の実行をくり返し行う.

(3) iがm_2以上になるとDOは満足され，ステートメント番号nの次のステートメントの実行に移る.

◎3のようなDO範囲からの脱出をノーマルイグジットという.

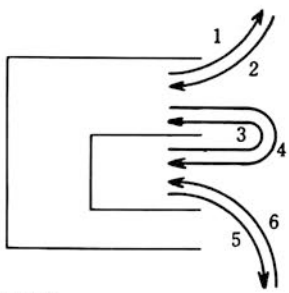
(4) $m_1 > m_2$の時はDO範囲は1回も実行しないで脱出する.

3．DO範囲のステートメントに対する規則

(1) DO範囲内に，別のDOステートメントを含むことができる．この場合，内側のDO範囲は外側のDO範囲に完全に納まり，互いに交わってはいけない.

(2) DOの範囲の外から，中に入ることは許されない.

例 34



1，3，5は許される.

2，4，6は許されない.

(3) DO範囲内ではインデックスiは普通の変数として扱われる．DOを満足しないで，DO範囲から脱出した時のインデックスは，その時点での値を保持し，DO範囲の外で変数として用いることが出来る.

◎DOが満足された場合，インデックスの値は不定となり，再定義されるまで用いられない.

(4) DO範囲内では，インデックス初期値，終値，増分パラメータの値を変えるステートメントを書いてはならない.

(5) DO範囲の最後のステートメントは，実行ステートメントでなければならず.

a．IFステートメント

b．GO TOステートメント

c．DOステートメント

であってはならない．このようなステートメントとなる場合はダミーステートメントとして通常CONTINUEステートメントが用いられる.

4—4 PAUSEステートメント

PAUSEステートメントは，連続的，自動的に行われるプログラムの実行を一時停止させるために用いる．PAUSEステートメントは，

```
PAUSE
あるいは
PAUSEn
n；符号をもたない1～4桁の10進数
    (1≦n≦2047)
```

の形を取る.

後者のPAUSEステートメントは，実行されると，コンピュータのコンソール上のアキュムレータに，10進数nと等価な8進数を表示して，プログラムの実行を一時停止する.

実行の再開はプログラムでは制御できず，オペレータがコンソール上のCONTを押すことにより，PAUSEステートメントの次の実行ステートメントからプログラムを再び継続することができる.

紙テープリーダにデータテープを，かけかえるためにもPAUSEステートメントが用いられる. この時はPAUSE文の次に，CALL OPENステートメントを用いて紙テープリーダを再起動させる必要がある.

4—5 CONTINEUステートメント

CONTINUEステートメントは，通常ジャンプの目標として，あるいはDOステートメント範囲の最後のステートメントとして用いられる.

```
n CONTINUE
n；ステートメント番号
```

```
例  DO  7  K=INIT,LIMIT
        :
    IF (X(K)) 22,13,7
        :
    7  CONTINUE
```

この例では，X(K)の値が正の場合，ステートメント番号7へジャンプし，このステートメントを実行する. CONTINUEステートメントはIFステートメントの目標番地であり，DOステートメント範囲の終結となっている.

4—6 STOPステートメント

STOPステートメントは，

```
STOP
```

の形をもつ. STOPステートメントでプログラムを終了させる. モニタシステムではコントロールはモニタに戻る.

4—7 CALL EXIT文

オブジェクトプログラムの実行終了後モニタに戻すにはSTOP文の代りにCALL EXITを用いる.

4—8 CALL CHAIN

CALL CHAINはコアに収容しきれない大きなプログラムを幾つかのセグメントに分け，これらを順にコアにロードして実行するために用いる. CALL CHAINによって，実行中のプログラムの代りに新しいプログラムがコアロードされ，コントロールが移される. この時COMMON領域に確

保された変数は新しくロードされたプログラムで使用できる.

CALL CHAIN ('PROG') の形で用いる. 呼ばれるプログラムはシステムデバイスに SYS:PROG·SVの形で登録されていなければならない.

5. ENDステートメント

ENDステートメントは,

```
END
```

の形をもつ.

ENDステートメントはコンパイラにソースプログラムの終りを指示する. ENDステートメントは一単位のプログラムの最後のステートメントであり, 必ずプログラムの終りに書かなければならない.

6. 入出力ステートメント

入出力 (I/O) ステートメントは, コンピュータと周辺機器の間のデータの転送と入出力データの書式を指定するために用いられる.

入出力ステートメントは次の2種類に分けられる.

1. FORMATステートメント; 非実行ステートメントで, コアメモリと外部との間でデータを転送する時に, データの形式の変換, 編集の仕方, (書式) を指定する.
2. READ, WRITEステートメント; コンピュータと入出力機器の間のデータの転送の方法を規定するデータ転送ステートメントである.

6—1 非実行ステートメント

非実行FORMATステートメントにより, 指定した外部機器上にあるデータの形式と, 配列を定義することができる.

6—1—1 FORMATステートメント

FORMATステートメントの形は,

```
n FORMAT (s1, s2……sn／s1, s2……sn / ……)
n ; ステートメント番号
 (s1, s2, ……sn…… / …… / ) ; 書式指定
s1, s2…… ; 欄記述子
 / 記録の区切の指定する.
```

◎FORMATステートメントはソースプログラム中のどこに置いてもよい.

◎FORMATステートメントの ( ) の書式指定が入出力転送される英数文字だけからなるデータの時は, データ転送ステートメントのリストと共に用いる.

◎**記録**は入出力の時のデータの集りで, 次のものに相当する.

1. 2ヶの**CR／LF** (キャリジリタン／ラインフィード) の間にはさまれた紙テープコード

2. タイプライタキーボードを用いてプリントした1行 (最高72文字)

◎**欄記述子**には, 次の3種類がある.

1. 数値欄述子
2. 文字欄述子

3. 空白欄記述子

詳細は**6—1—2**で説明する.

データを転送している間, オブジェクトプログラムは, 指定したFORMATステートメントの( )の書式指定を走査する. 書式指定の中に数値欄記述子があって, データ転送ステートメントのリストの中に, まだ転送されず残っている項があると, 欄記述子に従って転送が行われる. リストした項について, 全て転送し終ると, データ転送ステートメントの実行は終了する.

◎FORMATステートメントは, データ転送ステートメントのリストの中に書かれている項よりも多くの欄記述子をもつことができる.

◎FORMATステートメントは, データ転送ステートメントのリストの中に書かれた項よりも少ない欄記述子をもつことができる. この場合, フォーマットコントロールは最も右の左カッコにかえる.

◎FORMATステートメントは, 数値欄記述子, 文字欄記述子を混合して用いてもよい.
多重記録フォーマット, スキップする文字, スペース挿入および反復を取り扱うことができる.

◎入力リストの要求する一つのデータの文字の個数が多く, 入力機器が要求の全てを供給できない場合, 不足分にブランクが挿入される.

6—1—2 数値欄記述子

数値として取扱われるデータに対する変換形式と, それに対応するコンピュータの内部および外部での数値の形を次表に示す.

変換種類	内部での形	外部での形
E	浮動小数点2進数	"E指数"で表現される実数型10進数(324E+10)
F	〃	指数をもたない実数型10進数
I	固定小数点2進整数	10進整数

これらの変換は, 数値欄記述子.

```
rEw.d
rFw.d
rIw
```

で書かれる. ただしrはくり返し回数, wは数値の占有する全桁数(欄の長さ)で, 符号, 小数点, 指数部も含めて算定する. dは小数点以下の桁数である.

例 印刷のFORMATステートメントとして,

FORMAT (**I5**, **F10.2**, **E16.8**)

とすると, 次のように印刷される.

```
 I 5      F10.2          E16.8
bbb 32   bbbb -17.60  bbb .59625476 E+03
```

(ただし, bは空白を表わすものとする.)

◎E, F変換での入力の際, 欄記述子の小数点の位置と, 実際のデータの小数点の位置が異っている場合は, データの小数点の位置が正しいとみなされる.

◎変換された数値の桁数が, 欄の長さwより大きい場合は, 数値は左側にオーバーフローし, 越

えた桁は消失する.

数値の桁数の長さwより小さい場合は，数値は右側につめられる.

6—1—3 文字欄記述子

英数字のデータは，

> rAw
> ただし，
> r ；くり返し回数
> w ：欄の中の文字の個数

の形の文字欄記述子をFORMATステートメントに用いることにより，数値データと同じ方法で転送することができる.

この場合英数文字は，入出力リストの変数の値として転送される（リストの変数は整数型，実数型どちらもとれる).

◎転送される文字の最大の個数は，リスト変数の型で決る.

1．実数型変数では，最大6文字

2．整数型変数では，最大2文字

である（通常，一つの実数変数，整変数を表現するのにそれぞれ3語および1語のメモリが割り当てられている．一語のメモリには2文字納めることができる).

◎wが最大個数を越した場合

入力では，左側に超過した文字は入力されない.

出力では，文字の右側に空白が置かれる.

◎wが最大個数以下である場合

入力では，最大個数に達するまで，文字の右側に空白が置かれる.

出力では，左からw個の文字が外部機器に伝達される.

6—1—4 ホラリス変換

ホラリス変換（H変換）を用いて英文字の文章はFORMATステートメントの（ ）の書式指定の中に書くことにより，直接出力させることができる．H変換は，WRITEステートメントだけに用いる.

H交換は，

> n Hh_1, h_2……, h_n
> n ；文字の個数（空白も含める.）
> h_1, h_2……h_n 伝達すべき文字

例 FORMAT(17HbPROGRAM bCOMPLETE)

17文字

のステートメントで，␣PROGRAM␣COMPLETE とプリントできる.（␣：空白）

◎ホラリスの文章では，8K FORTRANで使用可能な全ての文字，記号を使うことができる．スペースは文章の中では，一文字として数えられる.

◎H変換の代りに′で両側を囲んだ文字ストリングを用いることができる．この場合は文字数を数える必要がない.

◎H変換は，本来WRITEステートメントに用いられるものであるが，MARKⅡ FORTRANで

は入力文に組み合わせてキャプションを印刷するのに用いることができる.

6—1—5　複合欄記述子

文字欄記述子は，他の別の欄記述子の中に書くことができる.

例　FORMAT（I 5，7HbFORCE=F10.5）

を用いると，

```
bbb 22  bFORCE=  bb17.68901
 I 5      7 H       F10.5
```

のように出力させることができる.

◎H変換の場合は，次に続く欄記述子との間の区切コンマは省略できる.

6—1—6　欄記述子のくり返し

同じ形の欄記述子をくり返して書く場合，くり返しの回数を変換記号E，F，I，Aの前に置くことにより，くり返しを指定できる.

例　FORMAT（2E12.4，3I5）の書式指定は，

FORMAT（E12.4，E12.4，I5，I5，I5）

と同じである.

6—1—7　グループのくり返し

いくつかの欄記述子の組の全体を（　）で囲み，その前にくり返し回数を書いて，欄記述子の組全体のくり返しを指定できる.

例　FORMAT（2I8，2（E15.5，2F8.6））は，

FORMAT（2I8，E15.5，2F8.6，E15.5，2F8.6）

と同じである.

6—1—8　多重記録フォーマット

各々の記録が，それぞれの一連の欄記述子をもっている場合，これらの記録の書式をまとめて指定することができる．この場合，この一連の欄記述子のあとにスラッシュ"/"を書く．"/"は記録の区切を示す.

例　FORMAT（3I8/I5，2F8.6）

は最初の記録に対しての，

FORMAT（3I8）

次の記録に対しnの，

FORMAT（I5，2F8.6）

を，まとめたものである.

◎スラッシュを用いる時は，コンマは省略できる.

◎n個のスラッシュは，書式指定の中の置かれる位置によって，次のような意味をもつ.

1．書式指定の中の最初，あるいは終り

出力の時（WRITEステートメントと一緒に用いる）はn個の空行が書かれる.

入力の時（READステートメントと……）は無視される.

2．中間にある場合

出力では（n－1）個の空白が書かれる.

入力では（n－1）個の記録が無視される.

3．書式指定の終りでの一個のスラッシュと，閉じカッコは記録の終りを示す.

◎入出力ステートメントのリストがFORMATステートメントの欄記述子より多くあって，FORMATステートメントの（　）を走査した時に閉じカッコ　）　に到達した場合は，フォーマ

ットはレベル1の最も前の開きカッコ　（　にもどって反復される.

例えば，次のステートメント，

```
FORMAT (F7.2,  (2 (E15.5, E15.4), I7) ) レベル0
        ↑          ↑              ↑
      レベル0    レベル1        レベル1
```

は第一のレコードに対しては，フォーマット，

F7.2, 2 (E15.5, E15.4), I7

が用いられ，続くレコード対しては，フォーマット，

2 (E15.5, E15.4), I7

が用いられる.

例　FORMAT (F7.2/ (2 (E15.5, E15.4), I7))

は，最初のレコードに対しては，フォーマット

F7.2

であり，後に続くレコードに対しては，フォーマット

2 (E15.5, E15.4), I7

である.

6—1—9　空白，スキップ欄記述子

空白欄記述子によっては，出力記録に空白を書いたり入力記録の文字を読み飛ばすことができる.
空白欄記述子は，

nX n；空白の個数，あるいは 読み飛ばす文字の個数 （n＞0）

である.

例　FORMAT(5HbSTEP 15,10×2HY＝F7.3)は，

$$\underbrace{\text{bSTEP}}_{5\text{H}}\ \underbrace{\text{bbb 28}}_{\text{I 5}}\ \underbrace{\text{bbbbbbbbbb}}_{10\text{X}}\ \underbrace{\text{Y=}}_{2\text{H}}\ \underbrace{\text{bb3.872}}_{\text{F7.3}}$$

と，出力することができる.

6—2　データ転送ステートメント

データの転送ステートメントには，

READとWRITE

の2つのステートメントがある．データ転送ステートメントは，FORMATステートメントの書式指定に従ってデータの入出力転送を行う.

転送する量を表わす変数のリストが，データ転送ステートメントに書かれる.

データは外部機器上に記録（レコード）の形である.

1．入出力リスト

入出力ステートメントのリストは変数値の転送の順序を指定し，書かれている順に左から右へ実行される.

リストは値を転送すべき変数名または表現のリストであって，その各項は次のいずれかの形をとる.

(1)　算術式．ただし入力については変数または添字付変数のいずれかに限る.

(2) ＤＯループ付算術式

たとえば（S1，S2…，Sn，i＝m1，m2，m3）の形をとる．

たとえばDIMENSION A（3，6）

WRITE（1，100）I，（A（J，I），J＝1，3）

では

I，A（1，I），A（2，I），A（3，I）が出力される

(3) 配列名．指定された配列のすべてを意味する．

たとえば

DIMENSION A（3，6）

READ（1，100）Aでは

A（1，1），A（2，1），A（3，1），A（1，2）…A（3，6）

のすべてが入力される．

◎入力している間，リストの中のある変数の値を，リストの中で，その変数より後に（右に）書かれた変数に対する添字としてあるいは制御ステートメントの式の中で用いることができる．

紙テープシステムでは，入，出力リスト中のインプライドDOは実行されない．

例えば，READ（2,1000）L，A(L)，B(L＋1)

リスト

Lの値を読み，この値をAおよびBの添字の中に用いることができる．

2．入出力記録（レコード）

入力時に存在する全ての情報の一単位は一つの記録単位（レコード）単位となっている．

プリンタへの出力では1レコードとは1行である．ASCIIレコードに含まれている情報の全ては，FORMATおよび入出力リストによって規定される．入力，出力ステートメントを実行すると，データレコードの転送が始まる．従って，次のステートメント，

READ（1，100）　FIRST，SECOND，THIRD

は，次の一群のステートメントと等しくない．

READ（1，100）　FIRST

READ（1，100）　SECOND

READ（1，100）　THIRD

なぜなら，後者は，少なくとも3個のレコードを要求しており，一方単一ステートメント，

READ（d，f）FIRST，SECOND，THIRD

は，FORMAT fの指定によって，何個でもレコードを要求できるからである．

入出力ステートメントの要求が，レコードの情報より少ない場合，レコードの中で要求されなかった部は消失し，再びそのレコードを同じ位置にセットしないかぎり，もう一度入出力ステートメントを用いても，取りもどすことは出来ない．

入出力リストの要求が，ASCIIレコードの情報より多い場合は，引き続きレコードが読まれる．

6—2—1　READステートメント

READステートメントは，選択された入力機器から内部メモリへ変数，配列，配列要素などリストに従って情報の転送を規定する．READステートメントは次の形をとる．

READ（d、 f）list
d ；入力機器を指定する番号
f ；FORMAT番号
list；変数のリスト

READステートメントは、ASCIIコードで書かれたデータを指定された機器から読み取り、リストの中の変数としてメモリへ格納する．データはFORMATステートメントの指定の形に変換され，コンピュータの内部へ読み込まれる．

6—2—2 WRITEステートメント

WRITEステートメントは、コンピュータから、選択された出力機器に情報を転送するのに用いられる．

WRITEステートメントは、次の形である．

WRITE（d、 f）list
WIRTE（d、 f）
d ；機器番号
f ；FORMAT番号
list：変数のリスト

WRITEステートメントの最初の形は、リストの中の変数の値をメモリから読み取り、ASCIIコードの形式で選択した機器上に書き下す．データは、FORMATステートメントで指定された形に変換されて出力される．

2番目の形は、特定のフォーマットから読み取り、ASCII形式で外部機器上に書き下す．

6—2—3 選択機器番号

機器番号は、READとWRITEステートメントの中に用いられる．

機器番号

1． 入出力タイプライタおよび低速リーダ、パンチ
2． 高速度リーダ、パンチ
3． カードリーダ、ラインプリンタ
4． I/Oドライバによる万能入出力（紙テープシステムを除く）

6—2—4 入出力タイプライタ（機器番号1）

入出力タイプライタでは、READステートメントについて次の特長がある．

1．キーボードまたは紙テープ読取部から入力されたデータは、印刷部にエコーバックされる．
2．FORMATステートメントの中のH変換およびホラリス定数部分（クオーテーションで囲まれる）は印刷される．

例 READ（1、30）A、B
　　30 FORMAT（4HA␣=␣、F7.2/′B␣=␣′、F7.2）

はデータを読みとる前に，

　　A= および B=

を印刷するので、直接キーボードから行う入力について、どのデータが要求されているかをユーザに示すのに用いるとよい．

6—2—5 高速紙テープリーダ、パンチ（機器番号2）

紙テープリーダからデータを入力し，あるいはデータを出力するのに用いる．FORTRANで，コンパイルしたMAINプログラムでは自動的にCALL OPENが挿入されており，これでフラグのイニシャライズを行う．プログラムの途中でPAUSE文をいれ紙テープのかけかえを行う場合には，PAUSE文の後にCALL OPEN文を入れて，フラグのイニシャライズを行う必要がある．

6—2—6　ラインプリンタ，カードリーダ（機器番号3）

ラインプリンタでは，出力されるデータの各行の第1字は改行制御に用いられ印刷されない．第1字が0ならば2行改行，1ならば次のページにスキップ，＋は改行無し（重ね打ち），その他の文字では1行改行を印刷の前に行う．

6—2—7　万能入出力（機器番号4）

モニタシステムではI/Oドライバを使用する万能入出力として機器番号4のデバイスの使用を許している．

この万能入出力はファイル構造のデバイスでも，その他でも良く，しかも，実行段階で指定することも可能である．

1．入力デバイスの割当て（IOPEN）

入力デバイスの割当てはCALL IOPEN（A，B）の形で行い，以降READ（4，X）の入力はデバイスAのファイルBから入力される．AおよびBはホラリス定数の形で与える．またファイルBの形名は，DAとする．

CALL IOPEN ('DSK'，'DATA')

とすれば，入力は　DSK：DATA.DAである．

CALL IOPEN ('PTR'，0)

では入力はRTR：が割当てられる．

デバイス名およびファイル名を実行段階でREADステートメントで読むとすれば，各A6FORMATとし，入力時には6字になるまで@を補う必要がある．

2．出力デバイスの割当て（OOPEN）

出力デバイスの割当てはCALL OOPEN（A，B）の形でIOPENに準じる．ファイルの形名は自動的に.DAとなる．

3．出力デバイスの切り離し（OCLOSE）

OOPENで作ったファイルはOCLOSEでダイレクトリに登録される．OCLOSEを行わない出力ファイルはモニタに戻った時点で削除される．OCLOSEを行わないで重ねてOOPENを行った場合はエラーとなる．OCLOSEは引数を持たない．すなわち

CALL OCLOSE

4．フォートランのデータファイル

WRITE（4，X）で作ったデータファイルは，後にREAD（4，X）の形で再び入力するためのみであれば，実数形の変数はA6，整数形の変数はA2で転送すると，データ量も小さく，2進10進変換も行われないので転送時間，演算時間共に大幅に節約される．

7．宣言ステーメント

宣言文は，非実行ステートメントで，プログラムの中では，あらかじめ，全実行ステートメントの前に置かれ，変数や定数に関する情報をコンパイラに知らせメモリの確保や割当を行う．

DIMENSION ステートメント，COMMON ステートメント，EQUIVALENCE ステートメントの宣言ステートメントが使用できる．

7—1　COMMONステートメント

COMMON ステートメントは，二つ以上のプログラムのそれぞれの中で定義された変数や，配

列の記憶場所を，共有させるステートメントである．

例えば，COMMON ステートメントを使用すれことにより，主プログラムのデータとその副プログラムのデータは，共通の記憶場所を共有できる．

COMMON ステートメントは次の書式による．

COMMON V_1, V_2, ……V_N

Viは変数名または配列名である．COMMON ステートメントはフォートランプログラムの一番始め（コメントステートメントを除く）に位置しなければならない．

◎COMMON ステートメントの中の変数の記憶場所は，フィールド1の200番地から順次割当てられる．（付録D参照）

7—2 DIMENSIONステートメント

DIMENSION ステートメントは，配列のための記憶場所を確保する情報を与えると共に，その宣言子名が，配列名であることをコンパイラに知らせる．

DIMENSION ステートメントは，

DIMENSION S_1, S_2, ……S_n S_1, S_2, ……S_n；配列宣言子

例 DIMENSION A (100)

DIMENSION Y(10), PORT(25), A(10, 10)

配列宣言は，

1．宣言子名

2．宣言子添字

からなり，配列名，配列の次元，大きさを規定する．次の形である．

S(i) ただし S；宣言子名 i；宣言子添字

◎COMMON ストレージの中の変数が，次元をもつ場合COMMON ステートメントは，DIMENSION ステートメントの前に書く．

7—3 EQUIVALENCE ステートメント

EQUIVALENCE ステートメントは，ある一つのプログラムの中の幾つかの変数の記憶場所として，同じ記憶場所を共有させる．

EQUIVALENCE ステートメントは，

EQUIVALENCE (V_1, V_2, ……), (V_k, V_{K+1}, ……) V；変数名

の形をもち，（ ）の中の変数の組が同じ記憶場所を共有する．

例 EQIVALENCE (RED, BLUE)

は，変数RED と変数 BLUE が同じ場所に格納されることを示している．

◎配列変数の添字は整定数である．

例 EQUIVALENCE (X, A(3), Y (2,1))

◎宣言子は，EQUIVELENCE ステートメントおよび COMMON ステートメントの両方に用いてはならない．

また，同じ変数が2度以上EQUIVALENCEステートメントにあらわれてはならない．

EQUIVALENCE（A，B），（B，C）

は

EQUIVALENCE（A，B，C）

とする．

8．副プログラム

外部副プログラムは，引用される主プログラムとは独立に存在する一つの完全なプログラムである．これは，“閉じた副プログラム”と呼ばれ，メモリーのある一つの場所に記憶され，使用するごとに，そこから呼び出される．外部副プログラムは，

1．FUNCTION ステートメント

2．SUBROUTINEステートメント

を，プログラムの先頭に置くことにより，定義される．

8—1 仮引数

副プログラムの中で使用される引数を，仮引数と呼び，その種類，使用方法は，副プログラムを定義するステートメント1，2により規定される．

仮引数は，副プログラムの中では普通の引数と同じように用いてよく，副プログラムを実際に使用する時には，実引数と置き換えられる．

8—2 関数副プログラム

関数副プログラムは一価の関数であり，算術式の中に，

FUNC（N）

（ただし，FUNC；関数名，N；引数）

のように，関数名として副プログラムの名前と引数を書くことによって引用できる．

（ここに，FUNC は，引数Nの関数値を求める副プログラムの名前．）

関数副プログラムは先頭にFUNCTION ステートメントが置かれ，END ステートメントで終る．

関数副プログラムから主プログラムにもどるには，RETURN ステートメントを用いて行われる．

8—3 FUNCTIONステートメント

FOUNCTIONステートメントは，次の形をとる．

FUNCTION name（a_1，a_2，……，a_n）
name；関数の名前
a_1，a_2，……a_n；仮引数

この文によって，この文以下のプログラムが関数副プログラムであることを宣言する．

◎“name”は定義する関数の名前である．副プログラム実行中は，“name”はスカラー変数であり，関数値が代入される．

◎（　）の中の仮引数a_1，a_2，……a_nは関数の引数を表わし，算術式，及び配列名であってもよいが，少なくとも一つはなければならない．

◎仮引数は，引用している主プログラムの実引数とその個数，順序，型が一致していなければならない．

◎仮引数が，配列名である時は，副プログラムの中で，DIMENSIONステートメントを用いて，

あらかじめ，配列名であることを宣言しておかなければならない．この場合，配列の添字は定整数型であって，その最大値は，主プログラムの対応する実引数の配列の添字の最大値を越してはいけない．

◎関数副プログラムの中の関数は，引用しているFORTRAN 算術式の引数を変えてはならない．

8—4 関数の型

関数の型は，変数の名前の付け方と同じで，関数の名前“name”の先頭の文字で決る．

8—5 SUBROUTINEステートメント

SUBROUTINE ステートメントは，次の形をとる．

SUBROUTINE name (a_1, a_2, ……a_n) name；SUBROUTINE の名前 a_1, a_2……a_n；仮引数

このステートメントによって，以下のプログラムがSUBROUTINE 副プログラムであることを宣言する．

◎仮引数は，呼び出し主プログラムの実引数と個数，型順序が一致しなければならない．

◎SUBROUTINE 副プログラムは，引数として，式や配列名を書くことができるが，仮引数は，スカラー宣言子，あるいは配列宣言子である．

◎仮引数が配列宣言子であるなら，副プログラムの中で，**DIMENSION ステートメント**を用いて，その大きさを定義しなければならない．

仮引数は，SUBROUTINE副プログラムの中の，**EQUIVALENCE ステートメント**および**COMMONステートメント**の中に用いることは出来ない．

8—6 SUBROUTINE副プログラム

SUBROUTINE副プログラムも，それ自体一つの独立したプログラムであり，関数副プログラムと多くの点で類似しているが，いくつか相異点がある．それは，

1．多価関数である．

2．引用する場合，CALL ステートメントで呼び出される．

多価関数であるから，同時にいくつもの値を取ることができるから複雑な入出力をまとめて扱うには便利である．

SUBROUTINE 副プログラムの構成は，

SUBROUTINE ステートメント

で始まり，

END ステートメント

で終る．

引用している，主プログラムにもどるには，

RETURN ステートメント

を用いる．SUBROUTINE副プログラムの中には，RETURN ステートメントは，少なくとも１つ以上なければならない．

8—7 SUBROUTINEステートメント

SUUBROUTINE ステートメントは，次の形をとる．

SUBROUTINE name (a_1, a_2, ……a_n)
name；サブルーチンの名前
a_1, a_2, ……；仮引数

このステートメントによって，以下のプログラムが，サブルーチン副プログラムであることを宣言する．

◎サブルーチンの中で使われる引数，即ち仮引数a_1, a_2, ……a_nは，引用している主プログラムの実引数と，(i)個数，(ii)順序，(iii)型が一致しなければならない．

◎仮引数は，(i)定数，(ii)変数名，(iii)配列要素名の他に，(iv)配列名，(v)算術式であるのでよい．

◎SUBROUTINE ステートメントの中の仮引数が，配列名であるなら，副プログラムの中で，**DIMENSION ステートメント**を用いて，配列名であることを宣言して置かなければならない．

8—8 CALLステートメント

CALL ステートメントは，次の２つの形を取り，主プログラムがサブルーチン副プログラムを引用する時に用いる．

CALL name
あるいは，
CALL name (a_1, a_2, ……, a_n)
name；サブルーチン副プログラムの名前
a_1, a_2……；仮引数

◎引数（a_1, ……, a_n）は，算術式や配列名，添字つき変数，各種定数などである．引数の個数，順序，型及び配列の大きさは，そのサブルーチンの仕様に一致しなければならない．

◎サブルーチン・副プログラムは，関数副プログラムと異なり，同時にいくつもの値を取ることができる．プログラム名自身は，値を取らないので，式の中に引用することはできない．サブプログラムの結果は，引数を媒体として，主プログラムにもどされる．

◎引用する時に，引数を必要としなければ□の中の最初の形を用いる．

例 CALL EXIST
CALL EXIST (VALUE, ……)

◎サブルーチン名を示す"name"は，型をもたない．また，引数の型とは無関係である．

◎定数，あるいは算術式の形で書かれた引数は，サブルーチンの中で変化されてはならない．

8—9 RETURNステートメント

RETURN ステートメントは，次の形を取り，副プログラムを実行した後に，これを呼び出している主プログラムに帰すためのステートメントである．

RETURN

◎RETURN ステートメントは通常，副プログラムの中では，最後の実行ステートメントとなるが，何カ所で使われてもよい．

◎RETURN ステートメントは，主プログラムの中に用いることはできない．

9. MARK Ⅱ FORTRANシステムの操作手順（モニタシステム）

9—1 ソースプログラム

フォートランのソースプログラムは通常EDITを利用して作成される.

EDITを使ってソースプログラムを作成する場合には，通常のフォートラン文法に従ってコラムを数える代りにTABでアドレス部と命令文本体とを区切ることが許される.

1. 各行の第1字がCであればコメント行である（コンパイラは無視).
2. 各行の第1字がSであれば，RAPDソースである（コンパイラはそのまま出力に転送する).
3. 各行の第1字が数字であれば，文番号である（文番号は1～4桁の10進数で2047以下). 文番号と命令文の間はTABを入れる.
4. 各行の第1字がTABであれば文番号がない命令文である.
5. 継続行の場合には第1字がTAB，第2字は1～9の数字とする. 継続行の本文は第3字から始まる.
6. 通常のフォートラン文法に従って，コラムワイズにソースを作成してもよい. この場合継続行は第6字に0以外の英数字で指定できる. 文番号は第2字から第5字に与える.
7. FORMATステートメントの中およびホラリス定数の中以外では，スペースはすべて無視される.

9—2 コンパイラの実行

コンパイラはFORTでロードされる.

例

```
.R FORT      FORTがSYS：にセーブされている場合
*
.RUN IOD3 FORT      FORTがSYS：以外にセーブされている場合
*
```

*に対しては0～3個の出力ファイル名と1個の入力ファイル名および必要数のオプションを指定する.

1. 出力ファイル指定

出力ファイルの1番目はRAPDの出力オブジェクト用で形名無指定の場合は.RLが付加される. ファイル無指定で/Lまたは/Gが指定された場合は，SYS：FORTRL.TMを作り，実行後FORTRL.TMを抹消する.

出力ファイルの2番目はRAPDのリスティング出力用で，形名無指定の場合は，.LSが付加される. ファイル無指定の場合はリスティングパスを実行しない.

出力ファイルの3番目はLLDRのマップリスティング用で形名無指定の場合は，.MPが付加される. ファイル無指定の場合はTTY：が指定される. /M，/U，/Pのいずれかのオプションを指定しない時は不要である.

入力ファイルは形名指定がない時には，.FTが付加される.

オプションは次の通りである.

／A	ソースリストをラインプリンタに印刷する.
／K	コンパイラ出力（RAPD入力）は，SYS：FORTRN.TMに格納され，通常はRAPDのアセンブル終了後，自動的に抹消される. Kオプションを指定すると，SYS：FORTRN.TMは抹消されない.
／L	RAPDのアセンブル終了後. LLDRを呼び，オブジェクトをロードする. ロード後，コマンド行の終りがESCかCRかによりキーボードモニタ（・）またはLLDR（*）

に戻る.

／G　RAPDのアセンブル終了後，LLDRを呼んでRAPDオブジェクトをロードし，さらに必要なライブラリ関数をSYS：LIB555からロードして実行する．実行開始番地は＝オプションで指定しなければMAINが自動的に指定される.

／N　RAPDのリスティング出力はシンボル表だけ（リスティングパスを実行しない).

／S　RAPDのリスティング出力はシンボル表を除く（N，S，の両オプションはリスティング出力の指定がなければ無意味である).

以下のオプションはLまたはGオプションを指定した時のみ有効である.

／I　万能入力ルーチンを使用する.

／O　万能出力ルーチンを使用する.

／H　万能入出力ルーチンは2ページドライバを使用する.

＝n　nは5桁のオクタル番地．実行開始番地の指定

／n　nは1～7の数字，LLDRはn−1以下のフィールドにはロードしない.

／M　ローディングマップ（EXTシンボルの絶対番地表示）を出力する.

／U　ローディングされていないサブルーチン名を出力する.

／P　各フィールドの残りのページ数を出力する（M，U，Pの各オプションで出力ファイルの指定がない時はTTY：が指定される).

例

*TEST/G

コンパイラは，DSK：TEST.FTをコンパイルして SYS：FORTRN.TMを作る．RAPDはSYS：FORTRN.TMをアセンブルしてSYS：FORTRL.TMを作り，SYS：FORTRN.TMを抹消する.

LLDRはSYS：FORTRL.TMをロードしてSYS：FORTRL.TMを抹消する．さらに必要なライブラリ関数をロードしてMAINから実行を開始する.

*,,LPT：＜TEST/L/M

前例と同様であるが，ローディングマップをLPTに出力する．ただしライブラリ関数はロードされない．LLDRのコマンド行に戻る．(*)

*,,LPT：＜TEST/L/M $

前例と同様であるが最後はESC，ライブラリ関数もロードして，ローディングマップをLPTに出力する．キーボードモニタ（.）に戻る．以後，SAVE，START，PODなどが実行できる.

*BIN，LIST，MAP＜TEST（KMG）

入力ファイル	DSK：TEST.FT
バイナリ出力ファイル	DSK：BIN.RL
リスティング出力ファイル	DSK：LIST.LS
ローディングマップ出力ファイル	DSK：MAP.MP
フォートラン出力ファイル	SYS：FORTRN.TM

コンパイル，アセンブル，ロード，実行の順に自動的に行う.

*TEST/K

DSK：TEST.FTをコンパイルしてSYS：FORTRN.TMを作る．さらにRAPDを呼びアセンブルするが，バイナリもリスティングも出力しない．コンパイラとアセンブラのエラーメッセージはTTYに出力される.

9—3　アセンブラの実行

アセンブラRAPDはフォートランコンパイラから自動的に呼出される以外に，単独のアセンブラとして，フォートランコンパイラ出力をあらためてアセンブルしたり，プログラマのコーディング

したソースプログラムをアセンブルしたりするのにも用いられる.

.R RAPDでロードされる.

コマンド行の出力ファイル指定（0～3個）についてはコンパイラの場合と同じである. 入力ファイルは形名指定がない場合は.RAが付加される.

オプション指定は次の通りである.

/L, /G, /N, /S, /I, /O, /H, =n, /M, /U, /P, /nについてはコンパイラの場合と同じである.

/F 入力ファイルはフォートランコンパイラの出力である.

9—4 ローダの実行

リンキングローダ（LLDR）はコンパイラ（FORT）またはアセンブラ（RAPD）で/Lまたは/Gが指定されれば, 自動的にロードされるが, 単独にも操作することができる.

コマンド行には0または1個の出力ファイルと1～9個の入力ファイルが指定できる. 出力ファイルはFORTまたはRAPDに対する3番目の出力ファイル指定と同じである. 入力ファイルの数は1つのコマンド行については9個以下であるが, 何行にもわたることは差支えない. 入力ファイルは形名指定のない時は.RLが付加される.

オプション指定は次の通りである.

/G, /M, /U, /P, /n, =nについてはコンパイラの場合と同じである.

/I, /Oおよび/Hは, コンパイラに対するオプション指定と同じであるが, 最初のコマンド行でのみ指定できる.

/L 入力ファイルはライブラリファイル（ダイレクトリ付の特殊形式ファイルで必要なものだけが選択的にロードされる）で, 入力ファイルが1個以上指定されても最初のファイル以外は無視される.

10. MARK Ⅱ FORTRANシステムの操作手順について（紙テープシステム）

この章では,

1. MARK Ⅱ FORTRANコンパイラ
2. RAPアセンブラ
3. リンキングローダ

のシステムプログラムを用いて, FORTRANプログラムをコンパイル, アセンブルおよび実行する方法を述べる.

リンキングローダをロードする場合を除いて, コンソール上のDATA FIELDスイッチのセッティングは無視される. それは, リンキングローダを除いて, 他のシステムテープは, テープ自身フィールドの設定をもつからである.

10—1 コンパイラのロード

1. Vローダがいずれかのフィールド0または1のメモリ内に正しくロードされていることを確認する.
2. FORTRANコンパイラテープをリーダにセットする.
3. コンソール・スイッチを次のようにセットする.
   (1) INST FIELDスイッチ（IF）＝n（nはローダの存在するフィールド）(DFは無視される.)
   (2) DATAスイッチ（DS）=7777 (8)
4. LOAD PCを押す.

5. STARTを押す.

6. 以上により，FORTRANコンパイラは，BINローダにより読み取られ，メモリに格納される．コンパイラは，フィールド0とフィード1へ格納される．

10—2 コンパイラの操作

コンパイルするソースプログラムは，主プログラムと，いくつかのサブプログラムからなり，各々ASCIIフォーマットで紙テープにパンチされてあるものとする（この場合各ソースプログラムは，テープの終りにEND文をもたなければならない）．

10—1の手順でメモリにロードされたコンパイラは，FORTRANステートメントをRAPアセンブラの命令に翻訳する．コンパイルされた結果（コンパイラ出力）は，紙テープ上に2つの部分に分けられてパンチされる．この場合，初めの部分（実行コード）は，ソーステープを読みながら，2番目の部分（変数，定数表）は，ソーステープが全部読まれた後で，パンチされる．2つの部分の間に約90cmの空白がある．

1. コンパイルの手順

(1) コンソール上のスイッチを，次のようにセットする．

DATA FIELD ＝0
INST FIELD ＝0
DATA (DS) ＝0200

(2) ソースプログラムをテープリーダまたはカードリーダにセットし，テープパンチをONにする．

(3) LOAD PCを押し，STARTを押す．

(4) ソースプログラムの入力機器を指定する．

CEC—555 FORTRAN COMPILER MARK—Ⅱ
CDR INPUT？
上記のようにタイプアウトしてくる．YまたはNをキーインする．
Y ； カードリーダから入力する．
N ； テープリーダから入力する．

(5) ソースリストの有無を指定する．

SOURCE LIST？
Nをキーインするとその時点からコンパイルを開始する．
Yをキーインするとラインプリンタ，入出力タイプライタの選択を行う．
ON LPT？
Y ； ラインプリンタに出力する．
N ； 入出力タイプライタに出力する．

(6) コンパイルを開始する．コンパイラ出力は，通常ソースプログラムの数倍の長さである．

(7) RAPのソースプログラムを出力し，停止する．

(8) 正常に終了した場合は，CONTキーを押すと再スタートする．

コンパイル途中でSTOPキーなどを押した場合は，DFキーを0，IFキーを0，DSを1600にセットし，LOAD PCキー，STARTキーを押すと再スタートする．

10—3 RAPアセンブラのロード

1. Vローダーがいずれかのフィールドのメモリ内にロードされていることを確認する．

2. コンソール上のスイッチを，次のようにセットする．

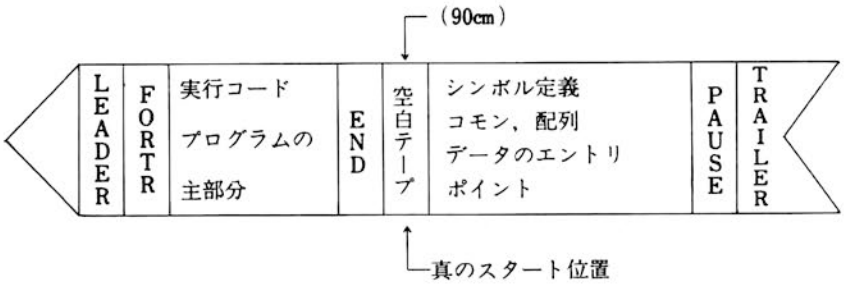
INST FIELD＝n（nはローダのフィールド，DFは無視される）
DATA スイッチ (DS) ＝7777

3. LOAD　PCを押す.
4. RAPアセンブラテープをリーダにセットする.
5. STARTを押す.
6. 以上より，RAPアセンブラは，メモリ内へ格納される（フィールド0とフィールド1にまたがって格納されている）.

10—4　RAPアセンブラの操作

RAPは独立したアセンブラであるが，さらにFORTRANコンパイルのパス2としても，用いられる．この目的の場合は，FORTRANコンパイラ出力のフォーマットが通常のものと違うことにより，RAPは，RAPマニュアルに記してある使用法とは少し違ってくる.

パス1のコンパイルで，ユーザのFORTRANソースプログラムは，シンボルプログラムがテープに変換される．その次に，RAPは，このシンボルテープをリロケータブルバイナリーに変換する．しかし，コンパイラによって作られたシンボルテープは下図に示すように，標準フォーマットのRAP言語のテープでない.



このようなテープの形になっているのは，テープの終りにあるデータを実行コードの中に入れることができないことや，テープの初めになければならないデータも，最後まで解らないからである.

このために，シンボルプログラムの真のスタート位置は，図に示される位置である．このようなテープをRAPでアセンブルするには，次に述べる3つの方法がある．実際の一般的な手順はRAPマニュアルの中に書かれているのと大体同じであるが，ある特定な部分の詳細が異なるから，以下で説明する3つの方法で，その相違をカバーする.

10—4—1　方法—1

最も簡単な方法は，図のようなシンボルテープを，シンボル定義表と実行コードの間の“空白”の真中で2つの部分に切断することである.

図のテープの後の部分（シンボル定義表）をセクション1，前の部分（実行コード）を，セクション2とする．アセンブルはこの2つに分けたシンボルテープで以下に述べる手順で行う.

メモリ内へロードしたRAPを用いて，コンパイラ出力をアセンブルするが，RAPのパス1で，ユーザのソースプログラムのリロケータブルバイナリ変換テープを作り，続いてシンボル表をタイプおよびパンチする．パス2は，リスティングパスである.

アセンブルは，次の手順で行う.

1. コンソール上のスイッチを，次のようにセットする.
   DATA　FIELD (DF) ＝0
   INST　FIELD (IF) ＝0
   DATA (DS) ＝0200
2. LOAD　PCを押し，STARTを押す.
3. RAPは，次の質問を順にタイプしてくる.

"PTR?"
"PTP?"
"NEED LISTING?"
"ON LPT?"
"ON PTP?"

質問に対する答が"YES"であるなら，

"Y"をタイプする．それ以外のどんな答も"NO"とみなされるので，答が"NO"の場合，例えば"N"をタイプすれば良い．4，5番目の質問は，3番目の質問に対する答が"Y"である場合だけにタイプされる．

4. 以下の手続からパス1である．
   RAPによって，ソーステープが読まれ，バイナリテープがパンチされる．
   そのあと続いて，RAPプログラムシンボル表が出力される．
5. ソーステープがいくつかの部分（最後を除き，終りにPAUSEをもったテープ）からなる場合，それぞれの部分の終りで，コンピュータは停止するので，この時に，ユーザは次の部分のテープをリーダにかけ，CONTINUEを押し，4. のステップを続ける．(注)
6. 以上でRAPによるパス1の手続が終り，コンピュータは停止する．(注)
7. アセンブリリストを必要とする場合は，次のパス2を行う．
   ソーステープをリーダに設置しなおし，CONTINUEを押す．
8. パス2が終ると，コンピュータは停止する．さらに，他のプログラムをアセンブルするには，CONTINUEを押し，コンピュータを再スタートさせる．(注)
9. RAPを再スタートさせるには，STOPを押し，一時コンピュータを停止させ，(i)DATEスイッチ(DS)＝0200にセットし，(ii)LOAD PC，STARTを押す．パス1から繰返されることになる．
10. RAPでは，最後の値に対して，答を与えたら，ユーザは，すぐに，パンチデバイス，あるいはASRリーダをONにしなければならない．低速リーダは，余り長くONにしないままでいると，エラーメッセージEをタイプする．ユーザは，始めから，スタートし直さなければならない．

(注) RAPの最新バージョンではコンピュータは停止せずに，その代り↑を入出力タイプライタに印字する．続行はスペースバーを押すことによってなされる．

10—4—2 方法—2

シンボルテープは図に示したように，約90cmの空白部分があるので，この部分を真のスタート位置に取ることにより，上で説明した方法1でシンボルテープを切断せずに，アセンブルすることができる．シンボルテープの終端にある，**PAUSEステートメント**でコンピュータが一時停止したら，シンボルテープの先端から，リーダに設置し直す．次に，CONTINUEを押すと，シンボルテープの空白部分まで読み取られ，パス1のアセンブリパスが実行される．

アセンブリリストを得るには，同じ方法でCONTINUEを押してパス2のリスティングパスをスタートさせる．

10—4—3 方法—3

8K FORTRANコンパイラによるシンボルテープ出力は，最初の行に特別な**疑似命令FORTR**を持つ．この疑似命令は，次に示すような働きをする命令であるから，アセンブルの各パスに対して，シンボルテープを2度通す必要がある．

第3の方法は，次のようにして行われる．

1. シンボルテープを先端からリーダへかける．
2. **10—4—1**の1.～5. のステップで，シンボルテープの読み取りを始める．
3. シンボルテープの最初の行のFORTRステートメントが読まれると，ENDステートメントまで

にある全てのプログラムは，無視される．続いてRAPによって，シンボルテープの後半のシンボル定義等が読まれ，その処理が行われて，アセンブルが開始される．

4. コンピュータは，シンボルテープの終端にあるPAUSEステートメントでいったん停止する．この時に，シンボルテープを先端から，リーダにかけ直し，CONTINUEを押す．
   RAPにより，テープの実行コードの部分のアセンブルが行われる．
5. アセンブルリストの必要がある場合は，パス1のアセンブルパスのあとで，方法—2と同じ手続きをふめばよい．

◎ここで，発見されるエラーは，未定義シンボルである．RAPはワンパスアセンブラであるから，アセンブリパスが終るまで，未定義かどうかは決定することができない．従って，シンボル表リストに，エラー診断UNDFが与えられることになる．

10—5 リンキングローダ

RAPにより変換されたリロケータブルバイナリプログラムは，MARK—Ⅱリンキングローダにより，メモリへロードされるが，リンキングローダは，次の役割を果たす．

1. ユーザプログラムや副プログラムをメモリへ格納したり結合する．
2. 記憶装置の記憶の状態，すなわち，
   (1) ストレージマップ
   (2) コアアベイラビリティ（使用可能なメモリ）
   を取り出す．

1. リンキングローダは多数のユーザプログラムやライブラリプログラムをいかなるフィールドのメモリへもロードできる．
   ◎各プログラムのロードするフィールドは，キーボードからのオプション指定によって選択される．
   ◎各フィールドに数個のプログラムをロードできるが，フィールド0の利用可能スペースは，他のフィールドより3ページ少ない（リンケージルーチンは，フィールド0に3ページのスペースを占めている）．
   ◎ロードされるプログラムの中で宣言されたCOMMONの格納場所を確保する（最大のCOMMONをもつプログラムを最初にロードする）．
   ◎リンキングローダ上にプログラムがロードされる場合には，いったんBINフォーマットで紙テープにパンチアウトする．このテープは全てのリロケータブルバイナリプログラムをロードした後にオプションBを用いてロードすることができる．
   ◎RAPプログラムで使用されるライブラリサブプログラムは，他のリロケータブルバイナリプログラムと同じ方法でロードされる．
   ◎ユーザのプログラムに実際に引用されているライブラリプログラムは，必ずロードされなければならない．
   ◎オブジェクトプログラムを実行するための，ランタイムリンケージルーチンは，オブジェクトプログラムの一部として，リンキングローダにより，全てのフィールド内の適当なエリアに自動的にロードされる．
2. リンキングローダによるロード操作により，どのサブルーチンがメモリのどこにロードされたのかというメモリの状態を知るためにM，U，Cのオプションがあり，キーボードからの指定によってそれらを選択する．

10—6 リンキングローダのロード

リンキングローダは，メモリフィールドの最終フィールドにロードする．

1. ベリファイローダが最終フィールドのメモリにあることを確認する．

2. コンソール上のスイッチをセットする.

DATA FIELD=h（hは最終フィールドの値）

INST FIELD=n

DATA　=7777

3. LOAD PCを押す.
4. リンキングローダのテープをリーダに設置する.
5. STARTを押す.

以上より，リンキングローダはメモリにロードされる.

10—7　リンキングローダの操作

リンキングローダはキーボードからのオプション指定によって，ユーザのリロケータブルプログラムおよびライブラリサブプログラムをロードする．この場合，最大のCOMMONストレージをもつプログラムを最初にロードする.

1. コンソール上のスイッチを次のようにセットする.

DATA FIELD= h（存在する最終フィールド）

INST FIELD= h

DATA (DS) =0200

2. LOAD　PCを押し，STARTを押す.
3. リロケータブルバイナリプログラムの入力機器，リンキングローダ上にロードされるバイナリテープの出力機器を指定する.

C E C - 5 5 5　LINKING LOADER

PTR?

PTP?

上記のようにタイプアウトしてくる．それぞれ高速紙テープデバイスを使用する時はY，使用しない時はNをキーインする．この指定はリンキングローダを再ロードしない限り変更できない.

4. OPT：と印刷後，次のコマンド1字をキーインすると指定業務を実行する.

OPT：n　　nフィールドにロードする.

OPT：M　　すでにロードされたプログラムで使用しているサブルーチン名を印刷する．ロードされていないサブルーチンについてはアドレスの部分にUと印刷される.

```
OPT：M
MAIN        10300
OPEN        U
WRITE       U
  ：
  ：
  ：
EXIT        U
0033
0033
```

OPT：C　　各フィールドの空ページ数を印刷する.

```
OPT：C
0033        （フィールド0の空ページ数）
0033        （フィールド1の空ページ数）
```

OPT：N　　ライブラリテープのように複数のサブルーチンを1本のテープにしてテープリーダにセットして使用している場合，不要のサブルーチンブロックを1つ読みとばす.

OPT：G　MAINの位置から実行開始する.

OPT：S　任意のアドレスから実行開始する. Sをキーイン後実行開始番地を, 5桁のオクタル数でキーインする.

OPT：B　リンキングローダ上にBINフォーマットの紙テープをロードする. この時DSのビット0をONにする.
動作終了後CPUはHLTするのでプログラムのスタートは, コンソール上のDF, IF, DSを手で操作する必要がある.

10—8　ライブラリ

CEC-555 フォートランシステムの標準ライブラリは, 次の10個のモジュールから構成され, 必要なモジュールだけを, いずれかのフィールドにロードして使用することができる.

	ルーチン名	必要とする他のライブラリ	ページ数（オクタル）
1.	READ, WRITE, IOH	2.，3.，5.	11
2.	FSB, FAD, ABS, ISTO, STO, FIX IFIX, FMP, FDV, FLOAT, FLOT IFAD, CHS	3.	5
3.	GENIO, OPEN, EXIT, ERROR, CKIO	5.	3
4.	ALOG, EXP, IFPOW, FFPOW	2.，3.，9.	3
5.	MPY, DIV, IREM, IABS, IRDSW CLEAR, SUBSC	3.	2
6.	SIN, COS, TAN	2.	2
7.	ATAN	2.	2
8.	SQRT	2.，3.	1
9.	FIPOW, IIPOW	2.，5.	1
10.	IOPEN, OOPEN, OCLOS, CHAIN	3.	1

通常のプログラムでは, 最低1.，2.，3.，5.の4つのモジュールを必要とする. 計算のみで入出力を伴わないプログラムでも2.，3.，5.の3つのモジュールが必要である.

10. のモジュールは, モニタベースのFORTRANシステムでのみ使用できる.

11. プログラムの実例

ここでは, 1から34までの偶数整数の階乗を計算するプログラムを, 一つの実例とする. このプログラムの実際のテレタイプのプリントアウトを下に示す. 主プログラムは, 関数副プログラムを引用して計算を行う.

1. ソース・プログラム

```
C       FORTRAN DEMONSTRATION PROGRAM
        DO 10 N=2,34,2
        A=FACT(N)
10      WRITE(1,60) N,A
        STOP
60      FORMAT(I3,'! = ',E16.8)
        END
```

```
C       FORTRAN FUNCTION TO COMPUTE FACTORIALS
        FUNCTION FACT(N)
        IF(N-34) 1,5,5
1       IF(N) 2,4,2
2       FACT=1.
        DO 3 K=1,N
        FK=FLOAT(K)
3       FACT=FACT*FK
        RETURN
4       FACT=1.
        RETURN
5       WRITE(1,6) N
        FACT=0.
        RETURN
6       FORMAT(I5,' EXCEEDS CAPACITY OF MACHINE!')
        END
```

2. FORTRANコンパイラをロードし、両ソース、プログラムをコンパイルする.

3. 2. から得られたシンボル言語をRAPでアセンブルする.

```
EXIT     0000EXT
FACT     0217EXT
FAD      0000EXT
FLOAT    0000EXT
FMP      0000EXT
IOH      0000EXT
OPEN     0000EXT
STO      0000EXT
WRITE    0000EXT
[0       0452
\FACT    0200
\FK      0204
\K       0203
\N       0450
\1       0246
\2       0253
\3       0307
\4       0332
\5       0350
\6       0423
]3       0212
]6       0207
↑A       0264
↑B       0324
↑C       0364
↑D       0446
```

```
EXIT     0000EXT
FACT     0000EXT
IOH      0000EXT
MAIN     0204EXT
OPEN     0000EXT
STO      0000EXT
WRITE    0000EXT
[0       0271
\A       0201
\N       0200
\10      0230
\60      0256
↑A       0210
↑B       0253
↑C       0267
```

4. リンキングローダがメインプログラムと必要なサブルーチンをロードする.

メモリマップのプリンタ出力

```
MAIN        1004
OPEN        5433
FACT        1217
STO         4425
WRITE       1664
IOH         3213
EXIT        5733
FAD         4016
FLOAT       5024
FMP         4606
REAL        1653
ERROR       5613
DIV         6051
IREM        6154
CLEAR       6240
FLOT        5046
IST0        4400
GENIO       5203
IFAD        5067
FDV         4673
FSB         4005
FIX         4501
IFIX        4537
ABS         4160
COS         5137
CKIO        5741
IABS        6200
XPY         6006
IRLSW       6221
SUBSC       6263
0005
0036
```

演算結果

```
 2! =    0.20000000E+01
 4! =    0.24000000E+02
 6! =    0.71999999E+03
 8! =    0.40320000E+05
10! =    0.36287999E+07
12! =    0.47900160E+09
14! =    0.87178291E+11
16! =    0.20922789E+14
18! =    0.64023736E+16
20! =    0.24329020E+19
22! =    0.11240007E+22
24! =    0.62044841E+24
26! =    0.40329146E+27
28! =    0.30488833E+30
30! =    0.26525286E+33
32! =    0.26313084E+36
  34 EXCEEDS CAPACITY OF MACHINE!
34! =    0.00000000E-00
```

付録A　キャラクタセットの10進および8進表現

MARK II FORTRANでは，6単位（ビット）のキャラクタを使用する．240～337の範囲の8単位コードのASCIIのキャラクタは6単位に縮められて，次表のように変換されている．

キャラクタ	10進	8進	キャラクタ	10進	8進	キャラクタ	10進	8進
Carriage	0	00	⊏	27	33	0	48	60
Return			＼	28	34	1	49	61
A	1	01	⊐	29	35	2	50	62
B	2	02	↑	30	36	3	51	63
C	3	03	←	31	37	4	52	64
D	4	04	Space	32	40	5	53	65
E	5	05	!	33	41	6	54	66
F	6	06	"	34	42	7	55	67
G	7	07	#	35	43	8	56	70
H	8	10	$	36	44	9	57	71
I	9	11	%	37	45	:	58	72
J	10	12	&	38	46	;	59	73
K	11	13	'	39	47	>	60	74
L	12	14	(	40	50	=	61	75
M	13	15	)	41	51	>	62	76
N	14	16	*	42	52	?	63	77
O	15	17	+	43	53			
P	16	20	,	44	54			
Q	17	21	−	45	55			
R	18	22	.	46	56			
S	19	23	／	47	57			
T	20	24						
U	21	25						
V	22	26						
W	23	27						
X	24	30						
Y	25	31						
Z	26	32						

付録B　ステートメント一覧表

ステートメント	形　　式	機　　能
コメント	Col.1に"C"を書く.	Col.2～Col.72までのデータは無視される.
CONTINUE	CONTINUE	通常DOの端末ステートメントとして用いる.
算術ステートメント	V＝e	eの計算結末を変数名Vに代入する.
GO TO	GO TO n	ステートメントnへ無条件に飛び越す.
	GO TO $(n_1,\cdots,n_n)$, i	$1\leqq i\leqq m$; $i=i_0$ のとき, ステートメントn_{i_0}へ飛び越す.
I F	IF(e)n_1, n_2, n_3	eの値が, $\left\{\begin{matrix}e<0\\e=0\\e>0\end{matrix}\right\}$に従い$\left\{\begin{matrix}n_1\\n_2\\n_3\end{matrix}\right\}$へ飛び越す
DO	DO n i＝m_1, m_2, m_3	ステートメントnまでを, i＝m_1からm_3ずつ増加し, m_2を越えない最大数ま繰返し実行する. m_1, m_2, m_3及びiは添字を持たない.
	DO n i＝m_1, m_2	m_3＝1とみなされる.
PAUSE STOP	PAUSE	プログラムの一時停止, CONTキーで再スタートできる.
	PAUSE n	整数nと等価な8進数を表示してプログラムの実行を一時停止する.
	STOP	主プログラムの実行を終了する.
	STOP n	整数nの等価な8進数を表示して終了する.
END	END	プログラム単位の終りをコンパイラに指示する.
READ WRITE	READ(d, f)e WRITE(d, f)e	dは装置番号　　fは書式指定のステートメント番号, eは入出力データの変数名リスト
FORMAT	FORMAT(R_1, …, R_n)	R_1, …R_nは書式指完
COMMON	COMMON a, b, …, n	a, …, nは添字なしの変数名.
DIMENSION	DIMENSION $a_1(R_1)$, …, $a_n(R_n)$	$a_1(R_1)$,……$a_n(R_n)$, は配列宣言子;配列であることを登録する.
FUNCTION	FUNCTION name(a_1, …, a_n)	a_1……a_n; 仮引数, 関数名は関数値を取ることのできる変数として, 定義される.
サブルーチン	SUBROUTINE name(a_1, a_n)	a_1……a_n仮引数.
CALL	CALL name(a_1, …a_n)	a_1…a_nはサブルーチンの実引数術
RETURN	RETURN	サブルーチンの場合, CALLの次のステートメントにもどる.
RETURN	RUTURN	関数の場合, 引用プログラムの式の計算にもどる.
EQUIVALENCE	EQUIVALENCE (V_1, …, V_n) ……(V_m, …, V_p)	同一プログラム単位内の2つ以上の要素に記憶場所を共有させる.

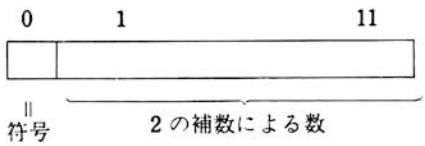
付録C　書式指定一覧表

種　　類	形　　式	機　　能
整　数	r I w	r；反復回数 w；欄の幅(数字の桁数) I；欄記述子
浮動小数点数	r F w . d	r；反復回数 w；符号，小数点を含めた欄の幅 d；小数部の桁数
指数表示	r E w . d	r；反復回数 w；符号，小数点を含めた欄の幅 d；指数部の桁数
英文字	r A w	r；反復回数　w；欄の幅
H（ホレリス，あるいはリテラル）	n H a b c d 'a b c d'	n；Hに続く文字の全個数 '（SHIFT/7）で囲まれた文字もプリントされる．
カッコ	n（書式指定）	（　）の中の書式指定がn回くり返される．
キャリジコントロール	/	新しいデータコードの始まりを示す．

付録D

D－1

整数は，1つにつき1ワード（12ビット）割り当てられ，2の補数形式による2進数で表現される．



符号のためにビット0が割り当てられる，

ビット0＝0　　正

ビット0＝1　　負

となっている．

従って



$3777_{(8)}=+2047_{(10)}$は最大の正の整数である．

$3777_{(8)}=+2047_{(10)}$は最大の正の整数である．

負の数は，1の補数に1を加えた数（2の補数）で表わされる．

例

＋1は　| 0 | 00000000001 |

－1は　| 1 | 11111111110 |　＋1＝　| 1 | 11111111111 |

＋1の1の補数　　＝7777(8)

最大の負の数は，

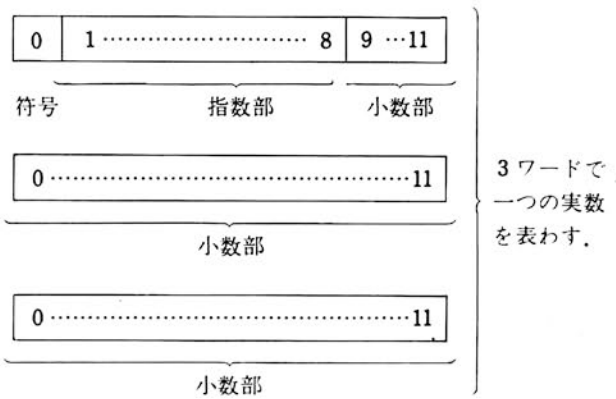
| 1 | 0 …… 0 |　＝4000(8)

で，－2047 (10) である．（計算結果では－2048が有り得る）

D—2 実　数

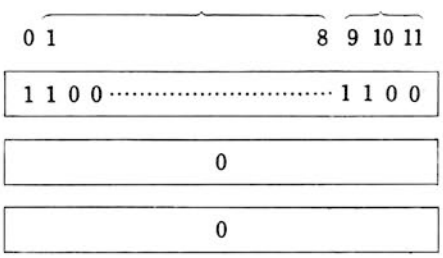
実数は，3ワード割り当てられる．

この指数部（2進）と2進小数の積の形で，次のようにして表現される．



符号は，ワード1のビット0である（0＝＋，1＝－）．指数部の符号と値は，ワード1のビット1～8の数から10,000,000 (2) ＝（200 (8) ）を引くことにより得られる．

（例1）



符号も（ワード1の0ビット）＝1 (2) ＝

指数部（ワード1の1～8ビット）＝10……01 (2) ＝

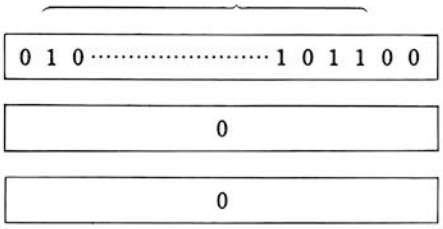
小数部＝.100 (2)

指数部＝201 (8) －200 (8) ＝1 (8)

小数部＝．1 (8)

数＝－.4 (8) ×2^{1} (8) ＝5（例2．）

（例２）



符　号;0 (2)

指数部;10,000　101 (2)

小数部：0.100 (2)

指数部＝205 (8)　－200 (8)　＝5 (8)

小数部＝.4 (8)

数　＝.4 (8)　×2^5 (8)　＝½×32＝16

D—3　配列のストレージ

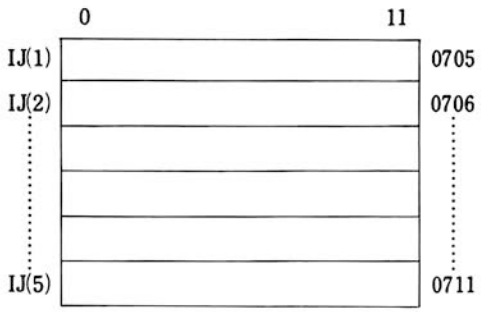
1. 整数型配列

配列順序はJIS標準に従って，コアメモリの中に上から下へ納められていく．

例えば，配列IJ，

DIMENSION IJ (5)

は，0705番地から始まると，



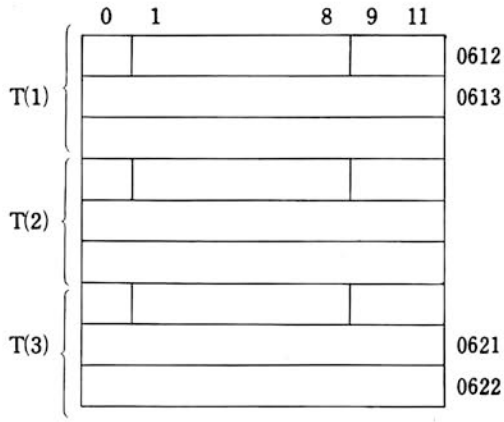
のように割り当てられる．

2. 実数型配列

実数の配列T

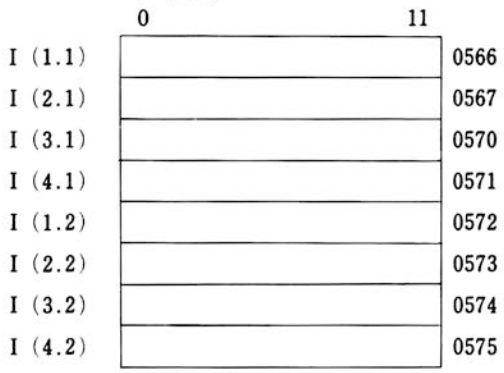
DIMENSION T (3)

は，0612番地から始まるとすると，次のように割り当てられる．



3. 2次元配列は、下の例のように割り当てられる.

DIMENSION I (4.2)



◎配列

A (M (J, K))

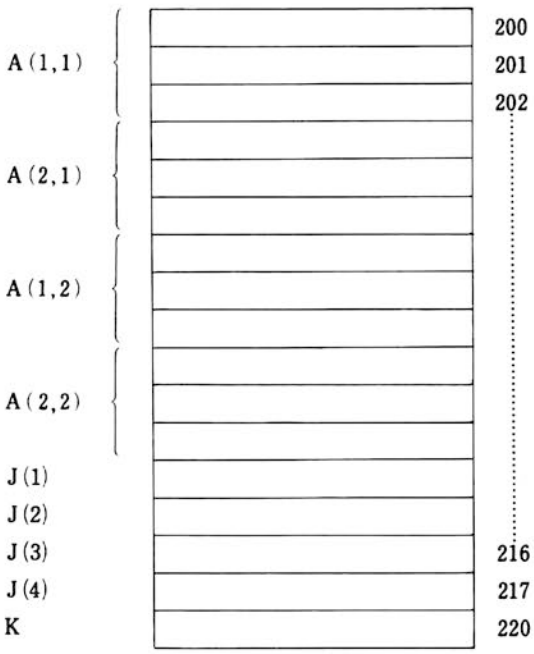
の場合, Mは上に示したように割り当てられる2次元の整数配列である. 今, 要素M (3.4) が, M (3.4) ＝7であるならば, A (M (3.4)) は, A (7) として計算される. ここで＝M (J,K) に割り当てられている整数の最大は, Aの次元を超えてはならない.

D—4 COMMONストレージのメモリ割り当て

COMMONストレージのメモリ割り当ては, フィールド1の絶対番地200から始まる. 変数は, COMMONステートメントに登録してある順序で, COMMONストレージのメモリ割り当てに配置される.

例 COMMON A,J,K
DIMENSION A (2,2) ,J (4)

は, 次のように納められる.



今，別の副プログラムの中で，COMMONステートメントが，

```
COMMON J
DIMENSION J (5)
```

のように定義しているとすると，J（1）～J（5）は，200～204番地に置かれる．即ち，変数A（1，1）とA（2，1）のうち5ワード共有されることになる．

付録E　エラーメッセージ

E－1

1つのステートメントをコンパイルしている時に，エラーが発見されると，誤りのあるステートメントとエラーメッセージがプリントされる．そのステートメントのコンパイルが終ると，それ以後の残りのコンパイルには出力は現れない．

しかし，コンパイラはエラーを求めて，残りのステートメントを走査し，エラーがあれば，そのステートメントをプリントする．

例　エラーメッセージの例を示す．

```
A = B + M(6) + N(1)
              ↑
MIXEND MODE EXPRESSION
```

（注）↑は誤りのあるステートメントの下に直接プリントされる．

(1) 例は，ステートメントの始めから↑印までのどこかにエラーが生じたことを示している．

(2) ↑印は，規則に反した文字，あるいはワードを直接にさし示すこともある．

(3) 一連の継続行の中にエラーがある場合，そのステートメントの残りの全ての行がエラーメッセージで，

```
ILLEGAL CONTINUATION
```

をつけて，プリントされる．

(4) コンパイラエラーメッセージを以下に示す．

ARITHMETIC EXPRESSION TOO COMPLEX
EXCESSIVE SUBSCRIPTS
ILLEGAL ARITHMETIC EXPRESSION
ILLEGAL CONSTANT
ILLEGAL CONTINUATION
ILLEGAL EQUIVALENCING
ILLEGAL OR EXCESSIVE DO NESTING
ILLEGAL STATEMENT
ILLEGAL STATEMENT NUMBER
ILLEGAL VARIABLE
MIXED MODE EXPRESSION
SYMBOL TABLE EXCEEDED
SYNTAX ERROR
I O ERROR
NO ROOM FOR OUTPUT
RAPD NOT FOUND
NO END STATEMENT

未定義文番号はRAPDでアセンブルするまで発見されない.

E-2 RAP

RAPは、バイナリリロケータブルプログラムのための、ワンパスオートマティックページングアセンブラであるので、エラーの処理は少しむずかしくなる.

ソースプログラムにエラーがある場合、アセンブルされたバイナリコードは事実上使用不能である. 特に、EとSのエラーは致命的であって、このエラーに出合うと、アセンブルは停止してしまう. しかしこれ以外のエラーが発見された行はコメントに同じように扱われ、アセンブルの過程で無視される.

従って、そのような行にある番地登録は未定義のままになり、即ち、誤りのあるデータはバイナリ出力の中にスペースを取ることはない.

アセンブリのパスの間で、エラーが発見されると次の形式でエラー診断がタイプされる.

```
C AT \10+0004
```

これは番地TAG\10より数えて4番目の命令にタイプCのエラーが発生したことを示している.

\10は、ソースプログラムのステートメント10に対応する.

リストを取るパスの間に、命令行の番地欄にエラーの種類を示す文字記号をタイプする.

エラー診断には、次のものがある.

A	CALLの後のARGの数の不一致
C	不正文字
D	入出力エラー
L	LまたはGでSYS:LLDRがない.
M	シンボルの多重定義
I	疑似命令のアーギュメント不良およびOPR命令の不正複合.
E	END文なし.
S	シンボル表溢れまたは出力ファイル溢れ
U	未定義シンボルあり.

なお、シンボル表が出力された場合には未定義

シンボルには各々にUNDFの注が付く.

◎RAPは，ワンパスアセンブラであるので，未定義シンボルアセンブリパスが終わるまで決まらない，これは，テーブルリスティングの中で与えられる.

E—3 リンキングローダ

リンキングローダによって，プログラムあるいはサブプログラムをロードする過程で，エラーが発見されると，ユーザはエラーメッセージによって，その内容を知ることができる．ロードしかけのプログラムは，その時点で無視され，そのフィールドから消される．コアメモリはフリーになる．エラーメッセージは，次の形でタイプされ，OPT：に戻る.

ERROR nnnn

nnnnは4桁のオクタル数で以下の意味を持つ.

0000	/ Iまた / Oがコマンド行の第1行目以外で指定された.
0001	シンボル表溢れ（EXTシンボルが64個以上）
0002	プログラム過大（コア不足）
0003	最大のCOMMON指定が第1ファイル以外で行われた.
0004	チェックサムエラー
0005	バイナリファイル形式不良
0006	コマンド指定不良
0010	MAINがなく，実行開始番地の指定も行われなかった.

リロケータブルバイナリテープの不良か，あるいはユーザが不正な取扱いをした場合に起る.

例　テープの始めからセットしなかった.
あるいはリーダエラー，あるいは，パンチエラーがある.
再ロードしてみて，エラーが起こるなら，再アセンブリする.

◎エラー2,4,5を正しく修正するには，テープをリーダに正しくかけなおして，オプション指定をする．この場合，エラーのあるプログラムを再ロードする前に，他のプログラムをロードしてはならない．エラー2の場合は別のフィールドを適当に選んで処理を続行する.

◎紙テープベースのシステムではロードの全過程は，次の手続きでリンキングローダをイニシャライズすることにより，再スタートできる.

a．DATA FIELD= h
　INST FIELD= h
　DS =6200
　} にセットする.

b．LOAD PCを押し，STARTを押す.

E—4 ライブラリプログラム

フォートランまたはリロケータブルアセンブラのオブジェクト実行時に，ライブラリサブルーチンの中でエラーが発見されると×××× ERROR AT nnnnの形のエラーメッセージを印刷する．××××はエラーコードで以下に説明する．nnnnnは5桁のオクタル数字でエラーの発見されたフィードルおよび番地を示す．重大なエラーについてはモニタ（・）に戻る（紙テープベースのシステムでは停止）.

1．モニタに戻るもの

ALOG	負数の対数
IOER	i）万能入出力指定で / Iまたは / Oが指定されなかった. ii）IOPENまたはOOPENの指定不良 iii）入出力エラー
CHER	CHAIN不良（指定ファイルなし）

FMT1　　FORMAT文不良
FMT2　　I形式入力での不正文字
FMT3　　F形式入力またはE形式入力での不正文字

ただし，FMT 2またはFMT 3で入力が入出力タイプライタの場合はそのREAD文を最初から繰返す．

2. 実行を続行するもの

DIVZ　　0で割る（最大値を結果とする）
EXP　　結果過大（最大値を結果とする）
OVFL　　〃　（　　〃　　）
FLPW　　負数の実数べき（絶対値を用いる）
SQRT　　負数の平方根　（　　〃　　）
FIX　　結果過大（−2047または2047を結果とする）

(1) ストレージマップから，最前の最少番号の番地（外部シンボル）を見つける．
これがエラーのあるプログラムのエントリポイントである．

(2) エラーメッセージの中のエラー番地LOCから，エラーのあるプログラムのエントリーポイントの番地を，8進数演算で差し引く．

(3) アセンブリシンボルテープから(1)で見つけた外部シンボルの相対番地を見つけ，ステップ(2)の結果に加える．

(4) (3)の結果がエラーの相対番地であるので，プログラム中のステートメントの相対番地と比較できる．

付録F　補足説明

F—1　入力／出力

F—1—1　数値入力変換

一般に数値入力変換は，他のFORTRANコンパイラと共用性をもっているが，しかし次に掲げる若干の例外がある．

1. 空白は，欄の桁数の決定の時にのみ意味をもつ．数値は欄の右側につめられて並べかえられる．
例えばF 5.2フォーマットの場合．

bbb12
12bbb
b12bb
00012

は全て0.12として読まれる．

2. 2つのCR/LFによりデリミットされた無効行は一行全部のスペースとして扱われる，また，FORMATステートメントを満たすために，データの後に必要なだけのスペースが補われる．
例えば，

12 (CR LF)
12bbb
bbb12

は，フォーマットF 5.2に関しては全て同等である．
行全体がスペースの場合，この行からの数値データは，ゼロとして読まれる．

3. 入力においては，EとFフォーマットの間に差異はない．

100.
100E2
1.E2

10000

は，フォーマットF5.2，あるいは，E5.2に関しては同様に読まれる．

4. キーボード入力についてはFORMATステートメントに拘らず，1つの数値ごとに一つのCRを必要とする．

F—1—2 数値出力

1．Eフォーマット出力および1.0より小さい数値に対する，Fフォーマット出力では小数点の左にゼロを印刷する．

2．nに，0のフォーマットでは小数点まで印刷される．

F—2

1. 0の0乗は1となる．
2. 0の0以外の数でのベキ乗は，0である．
3. ある数の整数ベキ乗は，その数をベキ乗数回だけ繰返し掛算することで計算される．
4. ある数の実数乗は，関数EXPとALOGを引用することにより計算される．
5. 負数の実数乗は，絶対値を用いて計算され，エラーメッセージが書かれない．

例　　(−3.0) **3.0は＋27となる．

F—3 添 字

添字の使いすぎは，メモリ効率を悪くするので添字付変数は，よく考えて用いるべきである．

例えば，ステートメント，

```
A＝((B(I)+C2)*B(I)+C1)*B(I)
```

は，コアメモリおよび実行時間の節約を考慮するなら，次のように書きかえた方がよい．

```
T＝B(I)
A＝((T+C2)*T+C1)*T
```

F—4 アセンブラコード

MARKⅡ FORTRANでは，ソースプログラムの中にアセンブラコーディングを自由に混ぜて用いることができる．これによって，フォートランシステムに含まれない入出力制御や論理演算などを容易に使用することが可能となる．フォートランのソース文で第1コラムがSであると，この行はRAPのソース行であることが，コンパイラに示され，コンパイラはこの行を何ら処理をせずに，コンパイラ出力（RAPソース）に転送する．

たとえば

```
          DIMENSION  K(10)
           DO  5        I＝1，10
SWAIT,    6131
S         JMP WAIT
S         6136
S         DCA  \ J
          K(I)＝J
5         CONTINUE
```

このコーディングは，デバイス13から10個のデータを入力して配列Kに格納する．この操作はサブルーチンサブプログラムの形で使用するよりもはるかに容易で，能率もよい．

F—5 SAPサブルーチンとのリンク

1．RAP言語で書かれたプログラムは，次の種々の方法でSAPサブルーチンを呼び出すことができる．

(1) DIMENSIONステートメントでSAPプログラムのエリアを確保しておき，RAP言語でJMS命

令を用い，SAPプログラムにリンクする．

```
      DIMENSION  A (20)
          :
S     JMS   \A
          :
      END
```

FORTRANの実行前に配列に，SAPプログラムをロードしておく必要がある．

(2) CONTINUEあるいはPAUSEステートメントを，FORTRANプログラムに挿入しておき，実行前に，スイッチレジスタからJMS命令に置換する．

F—6 FORMAT処理

READ，WRITEステートメントは，キャリッジリターン／ラインフィード（CR／LF）で動作を終了するが，ステートメントを（コンマ）で終結することにより，CR／LFを省略することができる．

1．READステートメントの場合

READステートメントの最後を（コンマ）で終結する．

```
      READ (1, 100) IA
100   FORMAT (A2)
```

2．WRITEステートメントの場合

WRITEステートメントの最後にダミー変数リストを記述し，FORMATステートメント中ではゼロとする．

```
      WRITE (1, 100) IA, I,
100   FORMAT (A2, 1XYZ, I0)
```

（注）WRITEステートメントの中のI，がないと，IAは出力されるがXYZは印刷されない．

F—7 エラー

コンパイル，アセンブルおよび実行時に発見されるエラーは全て致命的であるので，ソースプログラムが本当に正しいと信じられるまで，コンパイラやアセンブラのパンチ出力を抑えることが望ましい．

特殊な命令は，9章を参照．

コンパイラは，未定義のステートメント番号を検知しないので注意する．従って，プログラムをロードし実行する前に，未定義シンボルに対するアセンブリシンボル表を調べる必要がある．

アセンブリエラーをもつプログラムをロードやランさせてはならない．また実行時のエラーの後で，CONTを押して，続行してはならない．どちらの場合もどのような結果となるか予測できない．

⑪ コボル

1. 概　要

従来，コボルコンパイラは，一般事務用として中型，あるいは大型システムの主要ソフトウェアとして考えられていたが，ミニコンピュータシステムの利用範囲，応用分野の拡大にともなって一般事務計算，管理システムの分野でもミニコンピュータの利用度が高くなり，また，商業高校における情報処理科の学習課程においてもコボルベースの学習が必要となるなど，諸般の要請により，CECが他社に先がけてミニコンピュータコボルを実用レベルで完成したものである．

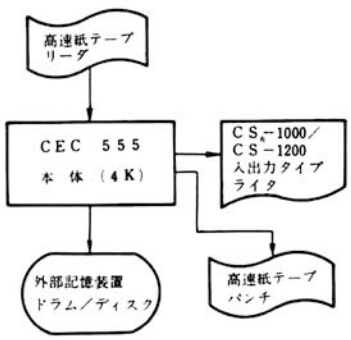
現代電子計算機の使用は大手，中小企業をとわず著しく，目ざましく発展している．

CECでは大型コンピュータで使用されている機能を盛り込み，CEC555COBOLを完成した．

CEC555シリーズは，丸編機・発券機の群管理，各種自動試験システム，自動計測システム，データ集録自動処理システム，心電図自動解析などアプリケーションシステムに多くの実績をもち，それぞれの分野で高く評価されているが，その主要ソフトウェアのなかにコボルコンパイラが加えられることにより利用範囲が数段拡張されたことになる．

2. システム構成

2—1　基本システム構成



※機能説明

1. CEC 555　電子計算機

記憶容量：4～16K語

サイクルタイム：0.8μs

演算語長：12ビット　1パリティ

加算時間：1.8μs

割込レベル：2

入出力装置数：64

基本命令：8種類

2. CS—408　高速紙テープリーダ

計算処理を記憶させるためのプログラムの転送およびデータテープの読取りに使用される．300字／秒，8単位

3. CS—1000　入出力タイプライタ（TTY）

データを電子計算機へ入力させるためのデータテープ作成，直接手動またはテープによる計算機へのデータ転送，また計算結果の帳票類作成のために使用される．16.6字／秒読取機，せん孔機，キーボード，印字の4機能

4．CS—420　高速紙テープパンチャ

計算処理結果のデータのパンチ，ソースプログラムを入力し，コンパイルされた後のオブジェクトプログラムのパンチなど，75字／秒，8単位

5．CFD—130　磁気ドラム記憶装置

固定ヘッド型磁気ドラム容量1トラック2K語　64トラック外部記憶容量として128K語使用できる．

6．用　　途

見積計算
在庫計算
部品計算
給与計算
経理計算

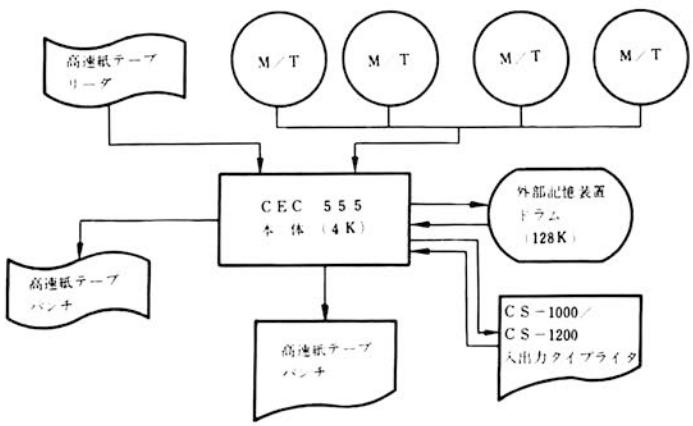
上記計算などは，簡単にコボル言語で計算可能である．また，機能の拡張により，計算業務だけでなく管理業務にも適用することが容易である．

〔例〕

1．生産管理，2．部品管理，3．部品発注，4．未納管理，5．経理業務全般など

CEC 555 COBOLの特色……転送，判断，計算などに両オペランドに添字が使用できる．また，関数も使用できる．

2—2　拡張システム構成



1．磁気テープ装置

外部記憶装置（M　T）9トラック　800BPI（密度）75インチ sec

2．CLP—4100　ラインプリンタ

1行　131桁　英数字，特殊文字，300行／分

コボルは各デバイスが増設された場合でも，ドラム接続して使用することができる．

3. COBOL入門

3－1　CEC 555 COBOL機能

COBOL (COmmon Business Oriented Language) は，商業用プログラム言語の一つである．

COBOLは，通常の英語に近い表現でプログラムを書くことができ，また複雑な入出力命令が簡単に書ける特長をもっている．

また，COBOLはその名が示すように電子計算機組織の個々の大きさ，特長に依存する部分が少なく，システムの変換に伴うプログラムの変換が容易である．

このような利点からCOBOLは事務計算用プログラムを書く手段として広く使用されている．

COBOLで書かれたプログラム (COBOL Source Program) を，機械語プログラム (Object Program) に翻訳するために，COBOL Compilerが用意されている．

3－2　COBOLプログラムの書き方

3－2－1　COBOLプログラムの構成

COBOLプログラムはDIVISIONとよばれる各独立した4つの部門から構成される．

IDENTIFICATION DIVISION：

プログラムに見出しをつけ，このプログラムの作成者，作成年月日などのコメントを書く．

```
iÐENTiFiCATiōN ÐiViSiōN.
PRŌGRAM-iÐ. EX1.
AUTHŌR. KŌBAYASHi.
REMARKS. SAMPLE PRŌGRAM.
```

このプログラムに対してEX1という名前をつけている．その他コメントが書かれている．

ENVIRONMENT DIVISION：

使用する電子計算機組織と，働かせる装置について記述する．

〔例〕

SEQUENCE NUMBER 1–6	– 7	A 8–11	B 12–72 COBOL STATEMENT
01			ENViRŌNMENT ÐiViSiōN.
02			CŌNFiGURATiōN SECTiōN
03			SŌURCE-CŌMPUTER. CEC555.
04			ŌBJECT-CŌMPUTER. CEC555.
05			iNPUT-ŌUTPUT SECTiōN.
06			FiLE-CŌNTRŌL.
07			SELECT iN-FiLE ASSiGN TŌ PTR-L.
08			SELECT ÕUT-FiLE ASSiGN TŌ PTP-L.
09			

このプログラムを使用する計算機はCEC 555である．入力ファイルは低速紙テープ読取装置，出力ファイルは低速紙テープせん孔機を使用することを，COMPILERに教えている．

DATA DIVISION：

扱うファイルの特性，レコードの形式，データの性質などに関する情報を与える．

〔例〕

```
SEQUENCE NUMBER  -*  A        B                    CŌBŌL STATEMENT
1      6         7  8     11 12        22        32        42        52
01   DATA DIVISIŌN
02   FILE SECTIŌN
03   FD  IN-FILE RECŌRDING MŌDE IS S.
04   LABEL RECŌRD ŌMITTED.
05   DATA RECŌRD IS IN-REC.
06   Ø1  IN-REC.
07       Ø2 IN-1      PICTURE XXXXXXXXXX.
08       Ø2 IN-2      PICTURE XX.
09
10   FD  ŌUT-FILE RECŌRDING MŌDE IS S.
11   LABEL RECŌRD ŌMITTED.
12   DATA RECŌRD IS ŌUT-REC.
13   Ø1  ŌUT-REC.
14       Ø2 ŌUT-1     PICTURE XXXXXXXXXX.
15       Ø2 ŌUT-2     PICTURE XX.
16
```

この例題のプログラムで，入力紙テープファイルはIN-FILEと名づけられ，1レコード 12桁のエリアが定義されている．

出力紙テープファイルも同様で，これはOUT-FILEと名づけられている．

PROCEDURE DIVISION：

入力，計算処理，出力の全般にわたっての命令を記述する．

```
PRŌCEDURE DIVISIŌN.
START.
    ŌPEN IN-FILE.
    ŌPEN ŌUT-FILE.
PRŌCESS.
    READ IN-FILE END GŌ TŌ END1.
    MŌVE IN-REC TŌ ŌUT-REC.
    WRITE ŌUT-REC.
END1.
    CLŌSE IN-FILE.
    CLŌSE ŌUT-FILE.
    STŌP RUN.
END CŌBŌL.
```

このPROCEDURE DIVISIONは3つの部分から成り立っている．すなわち，STARTで使用ファイルのオープンを行い，PROCESSでは入力ファイルを読み入力エリアを出力エリアに移し出力ファイルに書き出す．入力ファイルが最後まで読まれたらEND1で，ファイルのクローズを行い，このプログラムを終了する．

3－2－2　COBOLの言語構造

COBOLプログラムは，通常の文章と同様にWord（語）とWordを組み合わせて，Clause（節）やSentence（文）を作ることによって作文される．

本節ではCOBOL言語の構成要素であるWord，Wordを構成するCharacter（文字）およびそれらの結合規則を説明する．

1．使用する文字

COBOLプログラムとして使用することのできる文字は次の48文字である．

数　字	0～9
英　字	A～Z
特殊文字	
スペース	
＋	プラス記号
－	マイナス記号，ハイフン
＊	アスタリスク
／	スラッシュ
＝	等　号
¥	円記号
,	カンマ
.	ピリオド
″	引用符
(	左括弧
)	右括弧

2．Word（語）

Word（語）とは，次の文字の組み合わせからなる12字以内のものをいう。Wordはブランクによって一語の終わりとなる．

数　字
英　字
ハイフン

・COBOL Word（COBOL reserved Word）

これらの語は，COBOL Compilerに対して，あらかじめ特別の意味を持つように取決められている．すなわち，これらのWordの意味と用法は厳格に決っており，プログラマが自分勝手に，これを本来の意味と異なる目的で使用してはならない．

COBOL Wordはプログラムを書く使用法上3種に分けられる．

Key Word
Optional Word
Figurative, Constant

***KEY Word（基本語）**

一つの文を書く時に，必ず書かなければならない基本語がCOBOLにはいくつかある．たとえばMOVE a TO b. におけるMOVEとTOは移送という操作で必ず書かなければならない語である．これらの絶対に省略できない語を**Key Word**とよぶ．

***Optional Word（補助語）**

COBOLプログラムを書きやすくまた読みやすくするために適時挿入される語を**Optional Word**とよぶ．たとえばIF a IS EQUAL TO b ……
においてIF，EQUALは省略できないKey Wordであるが，IS，TOは書いても書かなくてもよい．COBOL CompilerにとってIF a EQUAL bは前文と同じである．しかしOptional WordもCOBOL Compilerとの間に一定の約束のある．COBOL Wordであるから正しい綴字で，正しい用法によっ

て使用しないとCompile時にエラーとなる.

＊Figurative Constant（Compileの既定の定数名）

COBOL CompiIeに対して命令や情報を与える前出の２種のCOBOL Wordに異なり，Figurative Constantは一種のデータである．このデータは，データ内容と，呼び名（data-name）がすでに約束されている.

ZEROは，数値項目としても，非数値項目としても使用できる.

SPACEは非数値項目としてだけ使用できる.

ZERO　　ゼロ

SPACE　　スペース

Figurative Constantは，相手の項目の桁数に等しくなる.

MOVE SPACE TO a

aの桁数分だけスペースが移送される.

・COBOL Wordの以外の語―Nameの作り方

下記に挙げる規則に従って，プログラマは好きな綴字による語を作り，これを使用することができる．プログラマが作る語はすべて名詞であり，NameとLiteralに大別される.

NameにはFile-Name，Data-Name，Procedure-Name，Record-Nameの４種がある.

＊File-Name

COBOLプログラムで扱うファイルに対して与えられる名称である．命名の規則は後述する.

＊Data-Name

COBOLプログラムで扱うデータに対して与えられる名称である．命名の規則は後述する.

＊Procedure-Name

PROCEDURE DIVISIONの中で処理手順のまとまりに対してつけられる名称である.

命名規則

① 使用できる文字は０～９，Ａ～Ｚ，ハイフンである.

② 長さは12文字以下である．（ただし先頭の６文字はユニークでなければならない）

③ 最初と最後にハイフンを置いてはいけない.

④ Procedure-Nameを除き，Nameは１文字以上英字を含まなければならない．Procedure-Nameは数字だけでもよい.

・COBOL語以外の語―Literalの書き方

プログラムの内で定数を必要とする時，その定数そのものを命令文中に書くことができる.

命令文中にその内容のものが書かれる時，これをLiteralという．たとえば“＊”を送りたい時MOVE“＊”TO A．において＊はデータの名称ではなく，データそのものであるからLiteralである．Literalには非数値Literal，数値Literalがある.

＊非数値Literal

① 最初と最後を引用符（”）で囲む.

② 引用符間にあるスペースはLiteralの一部として扱われる.

③ 長さは24文字以下である.

④ 非数値Literalは内容が数字のみであっても計算に使用することはできない.

＊数値Literal

① 数値Literalは，０～９の数字と負符号および小数点からなる.

② 負符号は最左端に―を書く．符号がない時は正数とみなされる.

③ 小数点は１コだけ書ける．小数点のおき場所は，最右端を除きどこでも良い．小数点のないLiteralは整数として扱われる.

④ 少なくとも1桁の数字をもち21桁以内であること.

3—2—3 COBOLプログラムシートの記入法

COBOLでプログラムを書くときは，COBOL Program Sheetを用いて定められた規則に従って書く.

COBOL Programは，比較的自由な形式で書くことができるが，書き出し地点，句読法は厳密に守らないとCompilerが正しく翻訳しない.

① 1～6 Column

Sequence Numberであるが，CEC555 COBOL CompilerがSequenceで付けるのでProgramerは書いても書かなくてもよい.

② 7 Column 使用しない.

③ 8～72Column COBOL Source Progamを書く.

3—2—4 句読法

COBOLプログラムを書く際には次の句読法を守らなければならない.

① 本書で，フォーマットの中に句読点が書かれている場合は，その句読点は省略できない.

② 左括弧のすぐ後，右括弧のすぐ前にスペースをおいてはならない.

③ WordとWord，または括弧をもった表現，あるいはLiteralの間には1つ以上のスペースをおかなければならない.

④ 演算子と符号の前後にはスペースをおく.

⑤ 編集のPicture-Clauseにピリオド，カンマ等を使うときは上記の規則ではなく報告書項目の規則による.

3—2—5 凡 例

本節では，今後COBOLの各部分の説明に先立って一般的な書式を書く．この一般形にしたがって，プログラマはStatementを書く．一般形の表現形式について以下に記述する.

① Word全体が大文字で書かれているものはCOBOL Wordである．これはすべてCOBOL Compilerに対して意味をもっておりその用法も決っている.

② COBOL Wordの中で下線が施されている語は省略できない．これらは省いたり，誤った綴りで書くとエラーになる．下線の引かれていないCOBOL WordはOptionであるから書いても書かなくてもよいが，記述する場合は綴りを間違えてはならない.

③ すべての句読点および特殊文字は，使用されている場合には必ず書く.

④ Data-Nameなど書式中の小文字の部分はプログラマが適当な名前もしくは値を与える.

⑤ 角括弧〔 〕に囲まれた事項は書いても書かなくてもよい．角括弧の中に2つ以上の事項が入っている場合は，どれか1つのみを選択するか，何も書かないのいずれかである.

⑥ 中括弧{ }にいくつかの事項が入っている場合は，その中のどれか1つを選んで必ず書かなくてはならない.

⑦ 各書式の意味，その使用法については本文中に説明する.

4. COBOL基本機能

4—1 DATA DIVISION

4—1—1 DATA DIVISIONの構成

1. Data DivisionはIdentification Division，Environment Divisionに続く第3番目のDivisionである．このDivisionでは，問題処理のためのプログラムステップを書くのに先立って問題処理に必要なデータの性質，その所在位置に関する情報をCompilerに与える.

2. Data Divisionの先頭にはまずDATA Divisionと書き続けすぐ後にピリオドをうつ．この残り

の場所はスペースのままにしておかなければならない．Data Divisionの後から次のDivision名，すなわちPROCEDURE Divisionまでの全体がData Divisionである．

3．Data Divisionの中は大きく2つの部分に分けられる．この部分のことをSectionと呼ぶSectionはそれぞれ既定の名称を持っておりData Division内で書かれる順番が決っている．

・**DATA DIVISION**

・**FILE SECTION**

File Description Entry,

Record Description Entry

WORKING-STORAGE SECTION

Record Description Entry

・各Sectionの役割は概略次のとおりである．

1．**File Section**

このSectionでは，Source Programで扱うFileの構成，内容について述べる．

2．**Working-Storage Section**

このSectionでは，プログラム実行中に，中間演算結果や，その他の項目を一時的に貯えたりする場所，いわゆるWorking Areaを設定する．また，ここで定数を定義して，その定数にData-Nameに与えることもできる．

4—1—2　データの構成

1．データの種類

・COBOLでプログラムを書くには，プログラマは扱うデータをデータ間の従属関係によってData-Item（項目），Record（レコード），File（ファイル）という3段階に分類する必要がある．

・Data-Item（項目）は，COBOLで扱うデータの内で最も低い段階のデータである．たとえば商品番号，商品名，単価などの，ひとつひとつの項目をData-Itemという．ひとつのData-Itemには，他のData-Itemとの従属関係を明らかにするために，必ずレベル番号をつける．レベル番号の高低によってData-Itemは，さらに基本項目とグループ項目に分けられる．

・RecordはData-Itemによって構成される情報の1単位である．たとえば，給与計算業務において，ひとりの従業員につき，従業員番号，氏名，基本給など，いくつかのData-Itemが必要である．これらのData-Itemが集まって，ある従業員に関する給与記録が出来上っている．このData-Itemの集まりを1単位として，Recordと呼ぶ．

・Fileとは，Recordの集まりをいう．

2．データのレベル

・COBOLのプログラムではData Divisionでデータを記述する際に，すべてのデータにレベル番号をつけている．レベル番号は，データの段階，およびData-Item間の従属関係を明らかにするものである．

データのレベルが高いということは，そのデータがさらに細分化できる段階のものであるという意味であり，データのレベルが低いということは，そのデータが即ちデータの一部として，他のデータに従属しているという意味である．

・レベル番号01

レベル番号01は各Sectionの中の次のような意味を持つ．

① **File Section**

このENTRYがLogical Recordを記述していることを示す．すなわちFile Sectionの中ではLogical Recordを定義するレベル番号01を書く．

② **Working-storage Section**

集団項目（Group Item）のまとまりとして書く.

```
01  A
    02  B  Picture  9・
    02  C  Picture  ×・
```

・レベル番号　02～49

02から49までの番号は，Data-Itemに対して与える番号である.

レベル番号は，数値が大きい程レベルは下になる．従って，他の項目との従属関係によって，02～49の中から適当な番号を選ばなければならない．その際レベル番号として連続した数値を使う必要はなく，大小関係さえ正しければ，02～49の中から自由な番号を選んでよい.

・レベル番号　77

レベル番号77は，Working-storage Sectionで独立した作業域（WORK AREA）を設定するために使用する.

・Data-Itemはレベル番号によって，従属関係が明らかになり，グループ項目，基本項目の別ができる．グループ項目とは，いくつかの項目を包含している項目をいう.

つまり，グループ項目は，さらに分解されて基本項目，あるいは，他のグループ項目になり得る項目である．いいかえると直後に続くレベル番号より小さいレベル番号を持つ項目はグループ項目である.

・グループ項目でないData-Itemは基本項目である．基本項目はそれ以下には分割されない項目である.

3．添　字（SUBSCRIPT）

・Recordの中に桁数，小数点の位置など性質が全く同じであるような項目が並んでいる場合がある．たとえば社員一人一人についての週間勤務状況として，月曜から土曜まで毎日の勤労時間を記録しているRecordを考える.

勤労時間をどの曜日も小数点以上2桁，小数点以下2桁で記述するとすれば，4桁の同種類項目が6コできることになる．このように同種類が連続的に並んでいる場合には，それらに個々別々の名称を与えずに同じ名称を与える．Procedure Divisionで使用するときには，Data-Name（添字）の形で使用するのが便利である.

添字（SUBSCRIPT）は，同種類項目の何番目の項目かを表わす.

・いくつかの同種項に同一の名称を与えるには，Record Description EntryでOccursを使用する．Occursを持つData-NameをProcedure Divisionで使用する時は，必ず添字をつけなければならない.

・添字を書くには，Data-Nameの後に1コ以上のスペースをおいてから添字を括弧で囲む．添字には整数を値するLiteral，Data-Nameがある.

4－1－3　FILE SECTION

・File Sectionは，プログラムで使用するすべての入出力Fileを規定し，その中に含まれるRecord，Recordを構成しているData-Itemについて，その性質を記述するSectionである.

・File SectionにFile，Record，Data-Itemを記述することによって

① File，Rcecord，Data-Itemに対して以後の呼称が決まる.

② Data-Itemの相互従属関係が決定する.

③ 記憶装置に必要なAreaが確保されてCompilerにとってFile SectionのData NameがDataを貯える場所の名称となる．たとえばCOBOLプログラマが，あるデータを計算要素として使うために，そのデータのData Nameを書くとCOBOL CompilerはFile Sectionの登録内容を参照して，データの場所，データの桁数，小数点の位置などを知り，必要な計算が行えるように機械語命令

を作り出して行くことができる.

1. **File Section Entry**

FD File-Name RECORDING MODE is S.
LABEL RECORD is OMITTED.
DATA RECORD is record-name.

・RECORDING MODE
S……Sequential

・LABEL RECORD
OMITTED……Labelは存在しない.

・DATA RECORD
ここでの01レベルのLogical Recordに名称をつける.

2. **Record Description Entry**

・Record Description Entryは, RecordおよびRecordに含まれるData-Itemの性質をCompilerに指示する.

ひとつひとつのデータごとにレベル番号, Data-Nameを書き, さらに項目の大きさおよびクラスを共に記述する必要がある. ただし項目がさらに細分化されている場合には, 最低のレベルで性質を述べればよいので, より高いレベルのEntryでは, レベル番号とData-Nameのみを書き, ただちにピリオドを打ってよい.

・レベル番号 (01～49) は, 8 Columnより書き始めてよいがData-Nameは12Columnより後に書く. 一般的には, Data-Itemの従属形態を視覚的に表わすために, レベルの低い項目ほど右にずらして, レベル番号を書く方法がとられている.

① Level Number

Level Number { FILLER / Data-Name }

・File Sectionではレベル番号77は使用できない. レベル番号の次に必ずData—Name または FILLERを書く.

FILLERは, 特にData—Nameを与える必要のない項目に対して使われる

② PICTURE

PICTURE is { Alpha—Numeric Form / Numeric Form / Report Form }

・Alpha—Numeric Formは項目が英数字項目であることを示す.
英数字項目には, その桁数だけXを書く.

・Numeric Formは項目が固定小数点であることを示す.
PICTUREには次の組合わせの符号をかく.
9：符号9は, 1桁の数字に対応する.
データがどのような記憶形態をとるときもPICTUREでは常に10進数で考えた場合の桁数と小数点位を書く.
V：符号Vは, 仮定の小数点位を示す.
数値項目は, 数字と符号しか含めないので実際の小数点をその中に加えることができない. そこで, 小数点を置くべき位置Vを置いてそこに小数点があるものとして, この項目を処理するようにCompilerに指示する.

従って記憶装置に小数点がおかれるわけではないので，Vは項目の大きさには含まれない．

S：符号Sは正負符号のあることを示す．

Sを書く場合は，PICTUREの最左端でなければならない．PICTUREにSが書かれている時のみ，符号が有効になる．この項目に符号Sが書かれていないと，出力には現われないが正符号となる．

・Report—Formは，データを読みやすい形に編集して印刷するためのものである．

編集用文字の意味は次のとおりである．

・：小数点は編集されるデータを小数点の基準にあわせ，実際の小数点を指定された位置におく．Vと異なり，小数点は項目の大きさに加えられる．同一のPICTUREには，ただ一つだけである．

Z：符号Zは，ゼロ消去記号である．

ゼロ消去とは，データの中の有効数字の左側の不心要なゼロをスペースに置き換えることをいう．すなわちPICTUREの各Z桁には，ソース・データ1桁が入るが，それが不要なゼロである場合には，その桁をスペースにする．

有効数字の右側のゼロは消去されない．また，ゼロ消去操作は，実際の小数点にぶつかり次第終了するから，ＺＺ・ＺＺはＺＺ・99と同じである．ただし，ソース・データが全桁0の場合は，スペースになってしまう．

—：マイナス記号は，PICTUREに1個だけ現われる．

¥：円記号は，PICTUREの最左端におく．そうすると編集の結果マイナス記号，円記号はそこに残って印刷される．

③ VALUE

VALUE is Literal

・編集用項目にVALUEを書いてはならない．

・OCCURSをもつ項目，あるいは，これらの項目に付随している項目に，VALUEを書くことはできない．

・VALUEのLiteralの長さは，PICTUREで指定されている長さに等しくなくてはならない．

④ OCCURS

OCCURS nnn

・データの中には性質が全く同じ項目が，いくつか連続している場合がある．同じPICTUREをもつこれらの項目を記述する際に，同類項がいくつ続いているかをOCCURSで指定すると，データ処理が簡単になる．

次の2例は全く同じデータの記述である．

```
02    A 1   PICTURE   XX.
      A 2
      A 3
      A 4
      A 5
02    A 6   PICTURE   XX.
02    A     OCCURS      6.
03    A N   PICTURE   XX.
```

・OCCURSで記述されたData-Name（An）をProcedure Divisionで使用する場合には，添字をつけなければならない．
・OCCURSはレベル番号01または77レベルには書けない．またOCCURSを持つ項目，またはこれに付随しいている項目にVALUEを書くことはできない．
・nnnは項目が繰り返される数を指定する．

4－1－4　WORKING-STORAGE SECTION

・Working—Storage Sectionは，データを処理するための中間的作業域（Work—Area）や，定数(Constant）を確保するためのSectionである．
・Work Areaや定数を確保するためには，File Sectionと全く同じようにRecord Description Entryで指定を行う．このためにFile Sectionで説明したすべてのレベル番号，およびPICTUREなどを使うことができる．
・Working—Storage Sectionで指定する．Areaや定数は，グループごとにまとめられるものと，独立しているものとの2種に分けられる．
他の項目に含まれず，かつ他の項目を含まない独立の定数には，レベル番号77を与える．これに反して，他の項目に含まれるか，または，他の項目を含んでいる場合は，File Sectionと同じようにレベル番号01～49を与える．
レベル番号77は，レベル番号01～49の項目より先に書かなければならない．

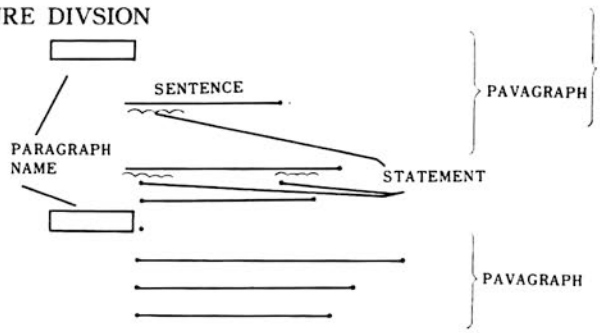
4－2　PROCEDURE DIVISION

1．Procedure DivisionはCOBOLプログラムの最後のDivisionである．このプログラムで使用するシステム，データの性質などに関してはProcedure Divisionの前の3つのDivisionで記述してある．
そこでProcedure Divisionでは，まったくシステムから離れて問題を解くためのステップを書くことができる．
2．Procedure DivisionはPROCEDURE DIVISIONという見出しで始まる．
PROCEDURE DIVISIONという文字の直後にピリオドを打つ．
3．PROCEDURE DIVISION.からSourceプログラムの最後まではPocedure DivisionのStatamemtであり，以下に述べる構文規則にしたがって書かなくてはならない．

4—2—1　構文規則

1．Procedure Divisionでは計算機に対する命令を文章と式で書き現わす．文章および式は英語文，数式の書き方とよく似ているが，独特のCOBOL文法の制約を受けている．COBOL文法とは要するに，Compilerとの約束であり，Compilerとの間に定められている語句を定められている文法で連結するための規則である．この規則を無視すると，Sourceプログラムを Compile するときに Compilerはエラー・メッセージを印刷してプログラマの注意を促す．
2．Procedure Divisionの構成は次の通りである．
(1)　ディビジョン　(Division)
(2)　パラグラフ　(Paragraph)
(3)　文　(Sentence)
(4)　ステートメント　(Statement)
により構成される．

PROCEDURE DIVSION

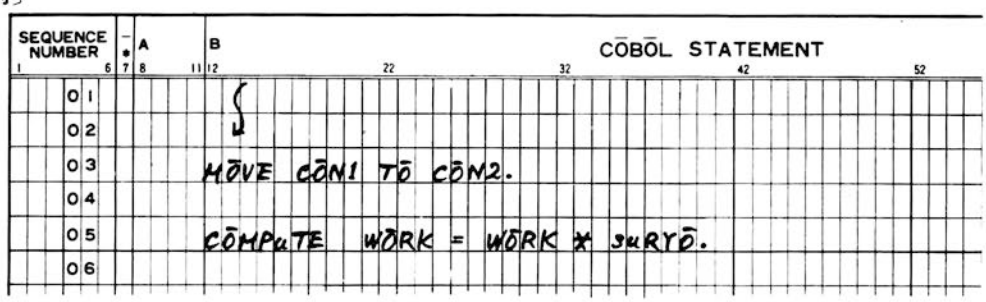


3．StatementはProcedure Divisionにおける命令の最少単位の少なくとも１つのCOBOL動詞（COBOL WORD）を含むものをいう．

4．Sentenceとは，１つまたは２つ以上の命令（Statement）が連続し，最後はピリオドで終わっているものをいう．したがって１命令で１つの文となるものもあれば，複数の命令で１つの文となるものもある．

文は12カラムから書き始める．

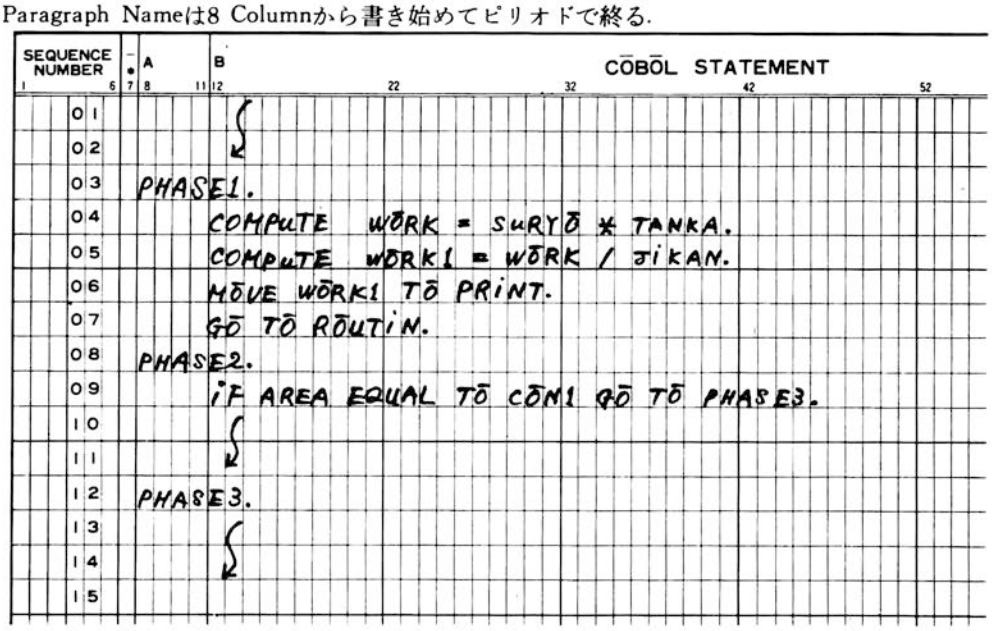
〔例〕



5．Paragraphは１つ以上の文の集まりでその始めに手続名（Paragraph Name）をつけられる．

Paragraph Nameから次のParagraph NameまでがParagraphである．

Paragraph Nameは8 Columnから書き始めてピリオドで終る．



4—2—2 STATEMENT

1．Statementとは少なくとも１つのCOBOL動詞を含む命令の最少単位である．COBOL動詞と定めている語を次に掲げる．

OPEN
READ
WRITE
CLOSE
MOVE
IF
COMPUTE
GO TO
PERFORM　　*注
EXIT　　*注　　がある．

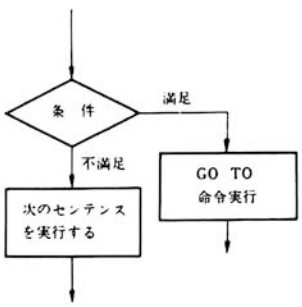
2．IF Statement

ある条件を調べて，その条件が満足されるか，満足されないかによって次に実行すべきステートメントを指定する．

形　式

```
IF { 項目名－A }  { IS  GREATER THAN }  { 項目名－B }
   {           }  {     LESS    THAN }  {           } GO TO
   { 定数(*注) }  {     EQUAL   TO   }  { 定数(*注) }
```

条件によってプログラムの流れを変えるのがIF命令である．
条件について比較した結果満足すればGO TO命令を実行する．
条件を満たさない場合は次のSentenceを実行する．



1．比較条件

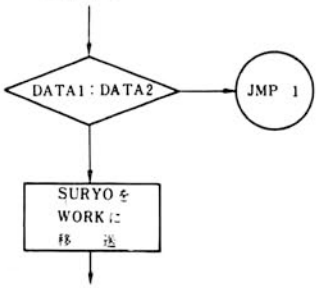
大きい---------→GREATER
小さい---------→LESS
等しい---------→EQUAL

2．比較する両項目のフォーマット（PICTURE）は等しくなければならない．

3．添字比較が出来る．

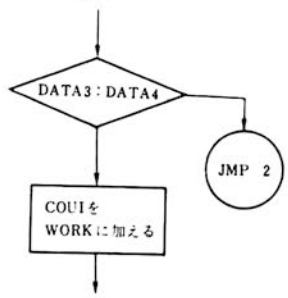
〔例 1〕

〔図 1〕



〔例 2〕

〔図 2〕

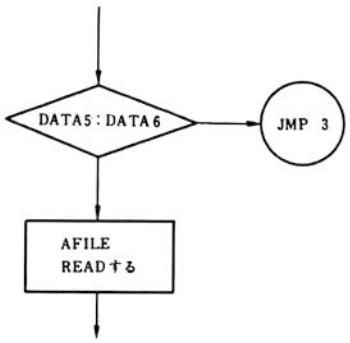


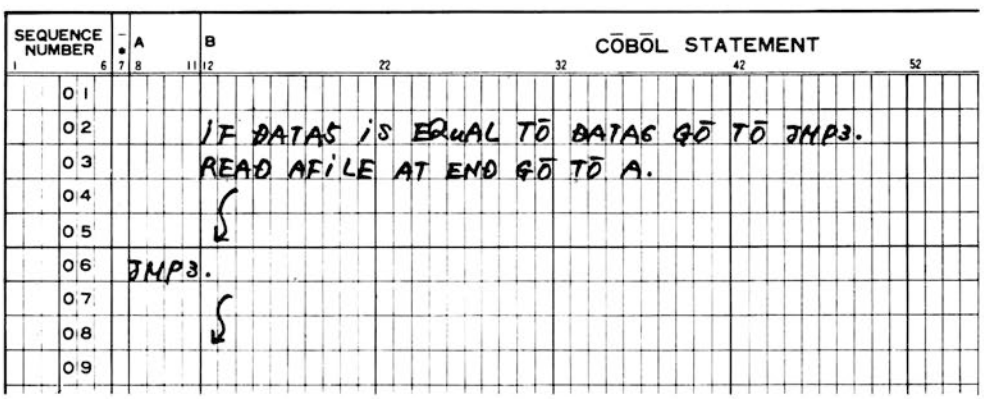
SEQUENCE NUMBER	A	B	COBOL STATEMENT
01			
02			IF DATA1 IS GREATER THAN DATA2 GO TO JMP1.
03			MOVE SURYO TO WORK.
04			
05			
06		JMP1.	
07			
08			
09			

SEQUENCE NUMBER	A	B	COBOL STATEMENT
01			
02			IF DATA3 IS LESS THAN DATA4 GO TO JMP2.
03			COMPUTE WORK = WORK + CON1.
04			
05			
06		JMP2.	
07			
08			
09			

〔例3〕

〔図3〕





2—2—3 MOVE Statement

データ項目の値または，定数をあるデータ項目に移したいことがよくある．この機能をもったのがMOVE命令である．

形　式

MOVE { 項目の名前 / 定数(＊注) } TO 項目の名前

1．MOVEの規則はつぎのようになる．

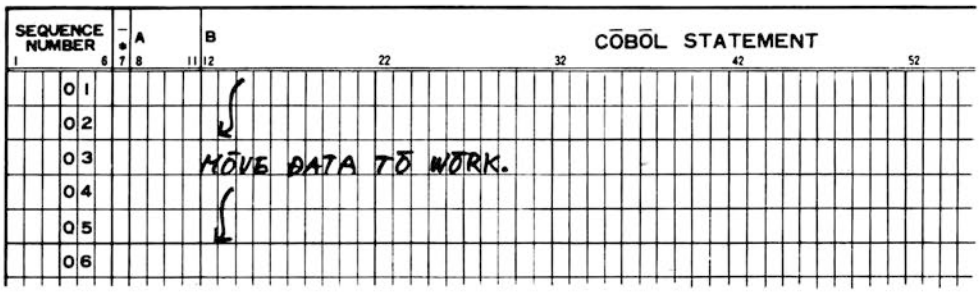
受取り / 送出し	文字項目	数字項目	編集項目	集団項目
文字項目	可	不可	不可	不可
数字項目	可	可	可	不可
編集項目	可	不可	不可	不可
集団項目	不可	不可	不可	可

集団項目移送の時，両方の基本項目のPICTUREが等しくないと正しい移送は行わない．

2．数字項目を移送するとき，その数字項目は整数部16桁小数部５桁までである．文字項目は24桁までである．

3．MOVE命令文で添字付の名前を使用することができる．（両方の項目に添字を使用することが出来る）．

〔例〕　DATAの内容をWORKエリアに移送する．



データの転送

送出し領域	送出しデータ	受取り領域	結　果
9999	1234	9999	1234
9999	1234	999999	001234
9999	1234	999	234
××××	ABCD	××××	ABCD
××××	ABCD	×××××××	ABCD△△
××××	ABCD	×××	ABC
999	100	×××	100
99999	01234	ZZZZZ	△1234
99999	01234	ZZZ,ZZZ	△△1,234
S9999	1234−	−ZZZZZ	−△1234
S9999	1234+	−ZZZZZ	△△1234
S99999	12345−	−ZZZZ	12345
999	100	¥ZZZZ	¥△100
9999	1235	¥ZZ,ZZZ	¥△1,235
99999	12345	¥ZZZZ	12345

2—2—4 COMPUTE Statement

四則演算を行う命令である．

形　式

COMPUTE 項目の名前 ＝ 四則演算式

1．使用可能な記号

等　号	＝
加算記号	＋
減算記号	−
乗算記号	＊
除算記号	／

これらの記号の直前，直後は空白でなければならない．添字計算に右カッコ）．左カッコ，（，が使用され，右カッコは直後，左カッコ直前空白でなければならない．

2．COMPUTE命令の左辺は数値項目か編集項目である．

3．COMPUTE命令の右辺は数値項目でなければならない．

4．演算できる数値の長さは整数部16桁，小数部５桁以内である．

5．COMPUTE命令の左辺，右辺で使用される項目の長さは異ってもよい．

6．COMPUTE命令の中で演算記号は１コである．（10コまで拡張の予定）

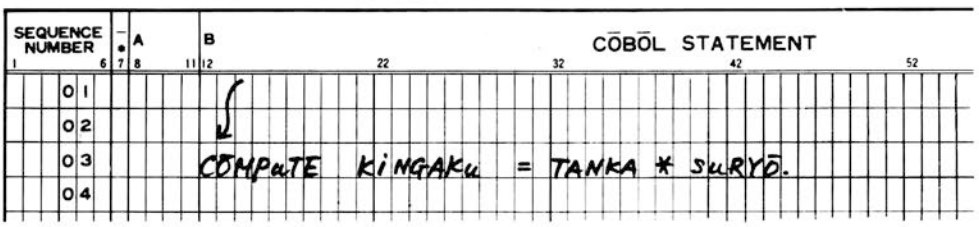
7．COMPUTE命令の左辺は定数であってはならない．

8．添字計算が出来る．

〔**例１**〕

単　価 × 数　量 ＝ 金　額

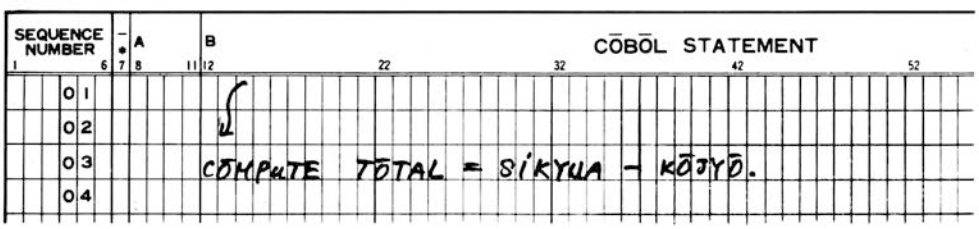
TANKA　　SURYO　　KINGAK



〔**例２**〕

支給額計 － 控除額計 ＝ 支払額

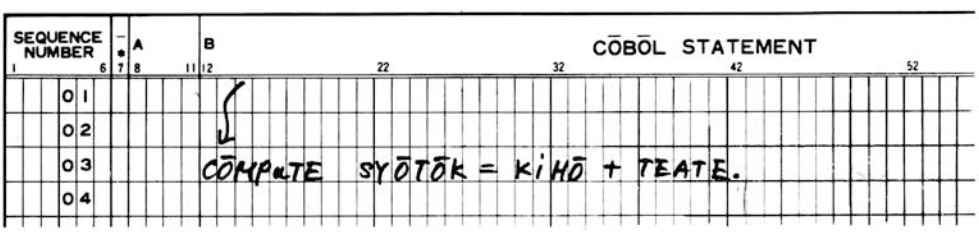
SIKYUA　　KOJYO　　TOTAL



〔**例３**〕

基本給 ＋ 手　当 ＝ 所　得

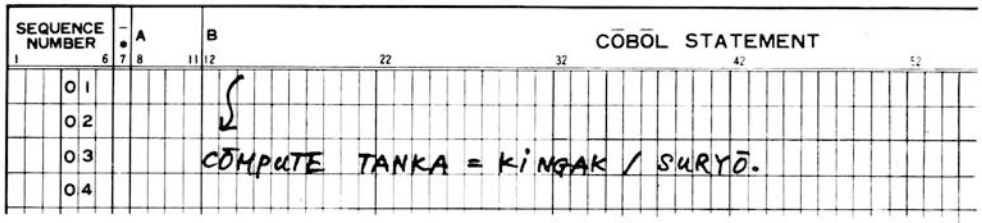
KIHON　　TEATE　　SYOTOK



〔例4〕

金　額 ÷ 数　量 = 単　価

KINGAK / SURYO = TANKA

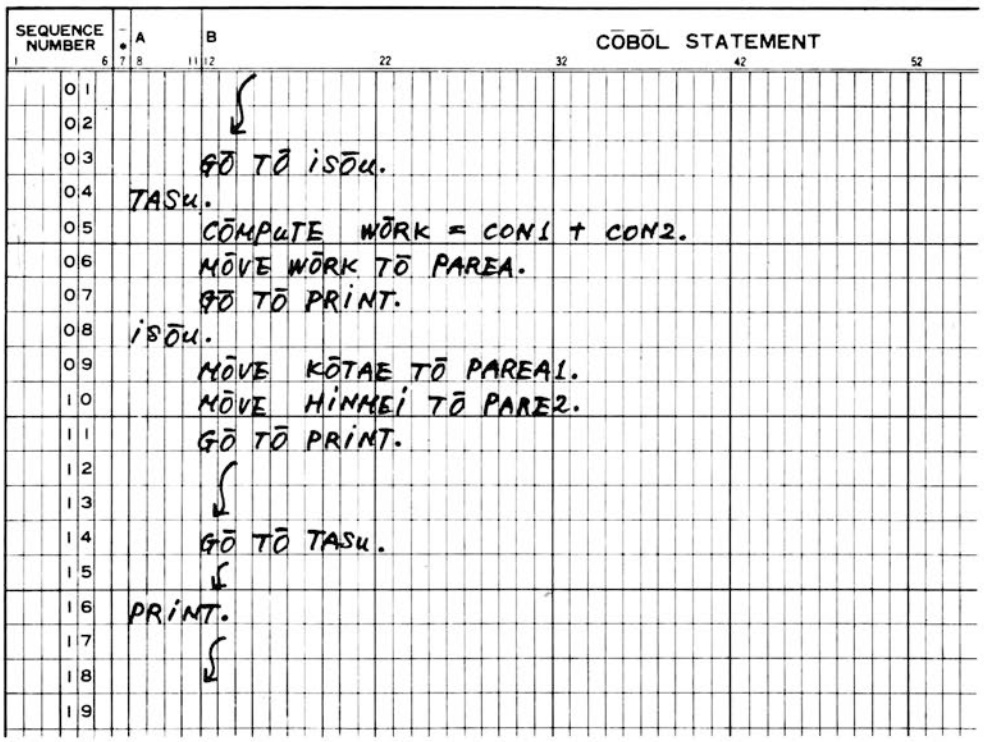


4—2—5

1. GO TO Statement

COBOLプログラムの通常の実行順序から無条件に順序をかえるときに使用する.

〔例〕



形　式

GO TO　パラグラフ名

2. PERFORM Statement

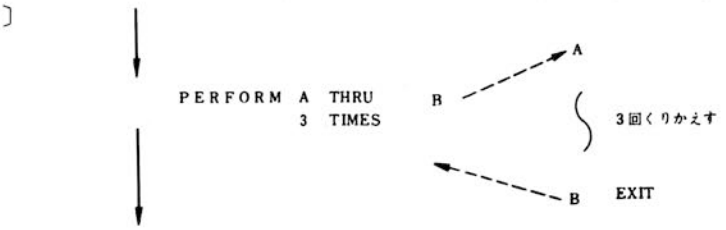
通常のプログラム実行順序からはなれて一連の手続きを実行し、再びもとの順序に戻る.

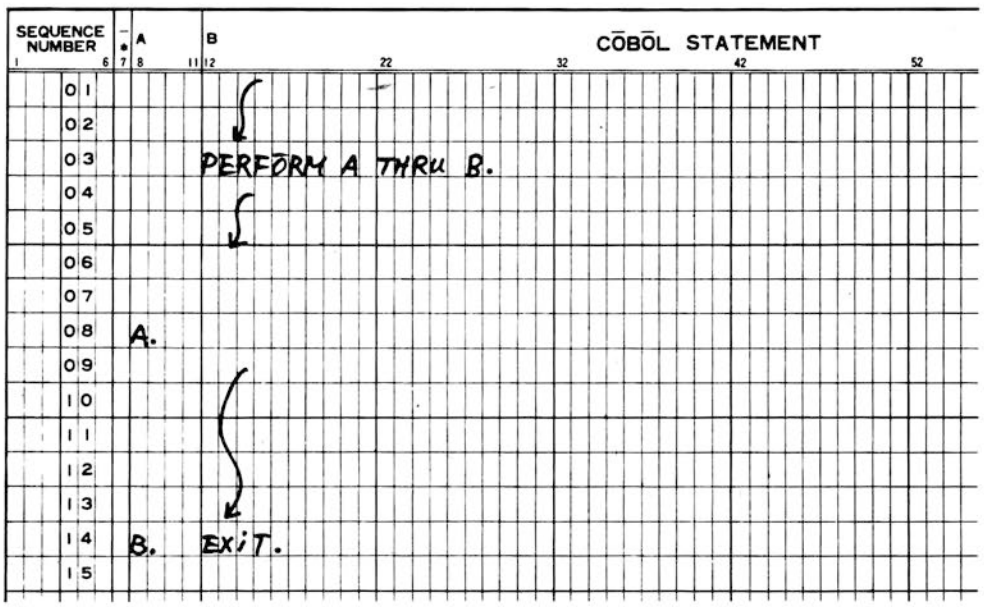
形　式

PERFORM　パラグラフ名—1　THRUパラグラフ名　2整数TIMES

1．整数で指定した回数だけその手続きを実行する．
2．整数は正の整数でなければならない．
3．指定できる回数は999までである．
4．指定された回数だけ手続きを実行すると元の実行順序（Perform命令のつぎの命令）にもどる．
5．Perform命令で実行されるサブルーチンの最後の命令が，IF命令やGO TO命令であってはいけない．そのような場合はExit命令を最後に書いておくこと．
6．Perform命令で実行されるサブルーチンの中で，さらに，Perform 命令を使用することは出来ない．
7．元の順序に戻るとき，Exit命令を用いパラグラフ名—2をExit命令につけた名前とする．

〔例〕





3．STOP STATEMENT

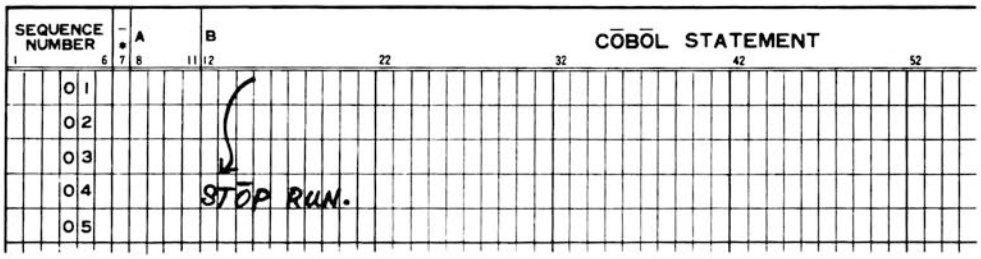
プログラムを終了させたり，一時的に止めたりする場合使用する．

形　式

STOP { RUN / 定数(＊注) }

1．RUNを使用した場合，プログラムの実行は完全にホルトする．
2．定数を使用した場合，定数をタイプアウトし一時停止する．
再開は次のステートメントが実行される．

〔例〕　プログラムを終了させる.



4－2－6

1. OPEN STATEMENT

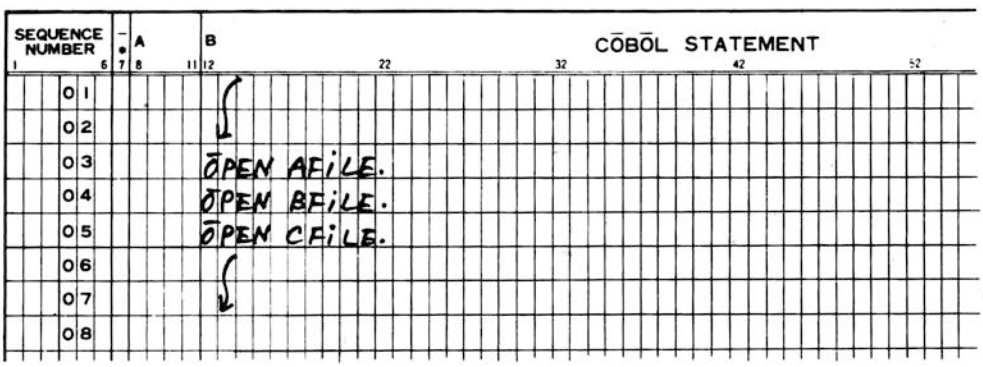
各入力ファイル，出力ファイルがREAD, WRITEできるように準備をする命令である.

形　式

OPEN　ファイル名

1. READ命令，WRITE命令を実行する前に，必ずファイルをOPENしておかねばならない.
2. 一度OPENしたファイルはCLOSEしないかぎり，再びOPENしてはいけない.
3. 紙テープファイルにおいてOPEN命令が実行されると，最初のCR，LFが検出されるまで紙テープを読みとばす.

〔例〕



4. ドラムファイルにおいて，OPEN命令が実行されると，ドラムファイル領域の先頭に位置付けされる.

2. CLOSE Statement

入力および出力ファイル処理を終わらせる.

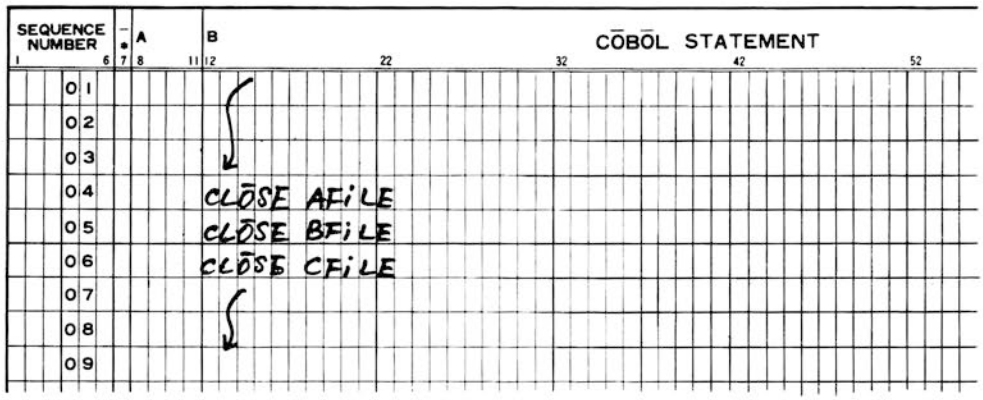
形　式

CLOSE　ファイル名

1. CLOSEはファイルの最終的な終了手続を行うため，すべてのファイルは使用済みになれば必ずCLOSEを行う.
2. 一度CLOSEしたファイルをもう一度使うためにはあらためてOPENしなければならない.
3. 出力ファイルが紙テープの場合，CLOSE命令を実行すると，ベル，リターン，ラインフィー

ドのコードがパンチアウトされる．出力ファイルがドラムの場合，CLOSE 命令を実行すると，3377(8)が書かれる．（エンドマーク）

〔例〕



3．READ Statement

入力ファイルのレコードを読み込む．またエンドマークを検出すると指定したParagraph Nameにジャンプする．

形　式

READ　ファイル名　AT END GO TO　パラグラフ名

1．最初のREAD命令を実行する前にそのファイルはOPENされていなければならない．

2．ファイル名はDATA DIVISIONのFDで書かれていること．またENVIRONMENT DIVISIONでSELECTされていなければならない．

3．フローチャート上から絶対にENDにならなくとも，AT END GO TOパラグラフ名はすべてのREAD命令に書かなければならない．

紙テープファイルのREAD

1．READされる紙テープファイルの最後には，ベル，リターン，ラインフィード　コードがファイルのエンド符号としてパンチされていなければならない．

2．最初の桁にベルコードのあるレコードをREADすると，AT END GO TO命令が実行される．

3．１つのファイルには一種類のレコードのみ許される．（１つのFDに１つの01レベル）

4．フィードだけ，またRub out，CONTROL@コードはデータに入らず読みとばされ，RETURN（215）は！（241）に，LINE FEED（212）は%（245）にそれぞれ変換されて読み込まれる．

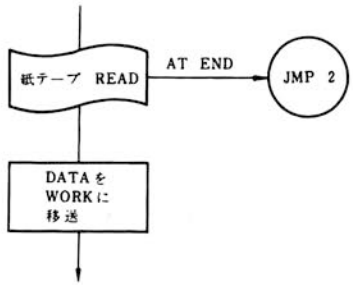
ドラムファイルのREAD

1．ドラムファイルのREADにおいてDATA DIVISIONのFDでとる01レベルは１コのみ許される．

2．ドラムファイルの領域をすべて読み終るか，その途中でも，最初の桁に3377(8)があるレコードをREADすると，AT END GO TO命令が実行される．

〔例〕

紙テープを高速リーダでREADしエンドマークが検出された時JMP 2にジャンプする．エンドマークでなかったならばDATAをWORKに移送する．




```
SEQUENCE NUMBER  -*  A        B                    COBOL STATEMENT
1      6         7  8      11 12        22        32        42        52
01
02                    SELECT AFILE ASSIGN TO PTR-H.
03
04
05
06              PROCEDURE DIVISION.
07
08                    OPEN AFILE.
09
10
11                    READ AFILE AT END GO TO JMP2.
12                    MOVE DATA TO WORK
13
14
15              JMP2.
16
17
18
```

4. WRITE STATEMENT

出力ファイルのレコードを書き込む.

形　式

①	WRITE	レコード名.		
②	WRITE	レコード名.	BEFORE	整　数.
③	WRITE	レコード名.	AFTER	整　数.

1. 最初のWRITE命令を実行する前に，そのファイルはOPENされていなければならない.
2. レコード名はDATA DIVISIONのFDで書かれていること．またENVIRONMENT DIVISIONでSELECTされていなければならない.

紙テープファイルのWRITE

1. 使用可能な形式は①のみとする.
2. 1つのファイルには一種類のレコードのみ許される．（1つのFDに1つの01レベル）
3. ！（241）はRETURN（215）に，％（245）はLINE FEED（212）にそれぞれ変換されて書き込まれる.

ドラム　ファイルのREAD

1．使用可能な形式は①のみとする.

2．1つのファイルには，一種類のレコードのみ許される．（1つのFDに1つの01レベル）

3．！（241）はRETURN（215）に，％（245）はLINE FEED（212）にそれぞれ変換されて書き込まれる.

4．出力ファイル　エリアとして，32Kword用意されている.

プリンタ　ファイルのWRITE

1．使用可能な形式は②，③とする.

2．1つのファイルには一種類のレコードのみ許される．（1つのFDに1つの01レベル）

3．！（241）はRETURN（215）％（245）はLINE FEED（212）にそれぞれ変換されて書き込まれる.

4．整数にゆるされる範囲は0～15までとする.

AFTER	0	HEAD SKIPしてプリント		
	1	1行改行してプリント		
	2	2	〃	
	3	3	〃	
	4	4	〃	
	5	5	〃	
	6	6	〃	
	7	7	〃	
	8	8	〃	
	9	9	〃	
	10	10	〃	
	11	11	〃	
	12	12	〃	
	13	13	〃	
	14	14	〃	
	15	15	〃	
BEFORE	0	プリント後HEAD SKIP		
	1	プリント後	1行	改行
	2	〃	2	〃
	3	〃	3	〃
	4	〃	4	〃
	5	〃	5	〃
	6	〃	6	〃
	7	〃	7	〃
	8	〃	8	〃
	9	〃	9	〃
	10	〃	10	〃
	11	〃	11	〃
	12	〃	12	〃
	13	〃	13	〃
	14	〃	14	〃

15　　〃　　15　　〃

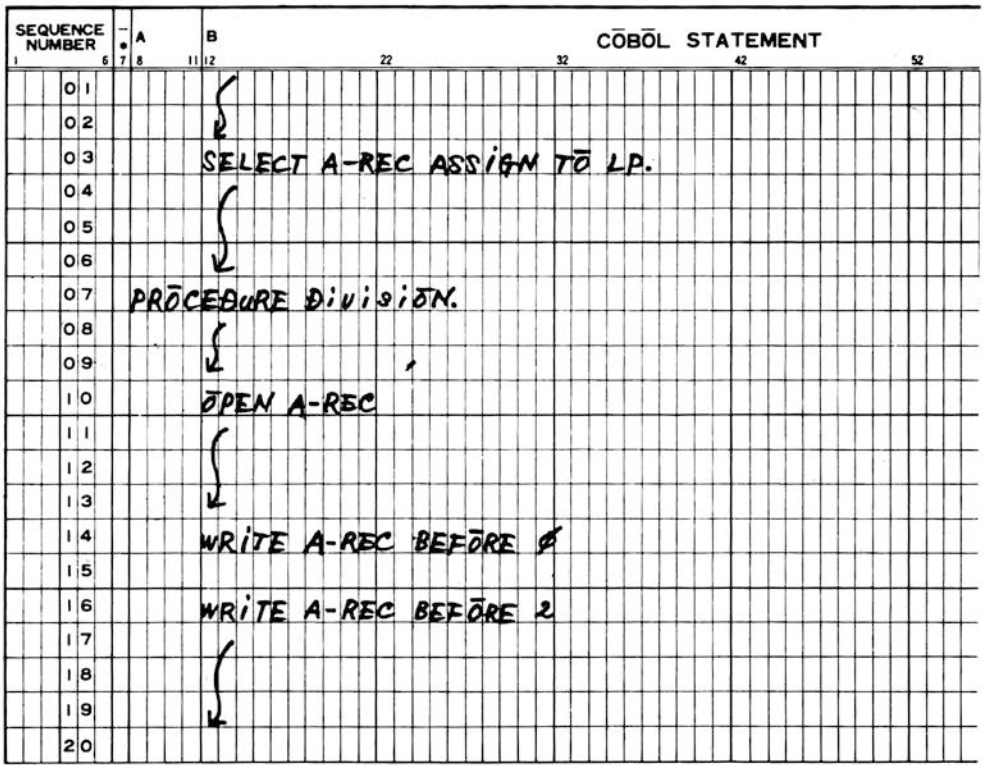
これ以外はエラーとする.

下段記号は内部では　$46_{(8)}$とする.

上段　〃　　〃　　$47_{(8)}$　　〃

〔例〕

プリンタ出力において，HEAD SKIPを行い2行改行して印字を行う.



4－2－7

1．EXIT Statement

サブルーチンの出口を示す.

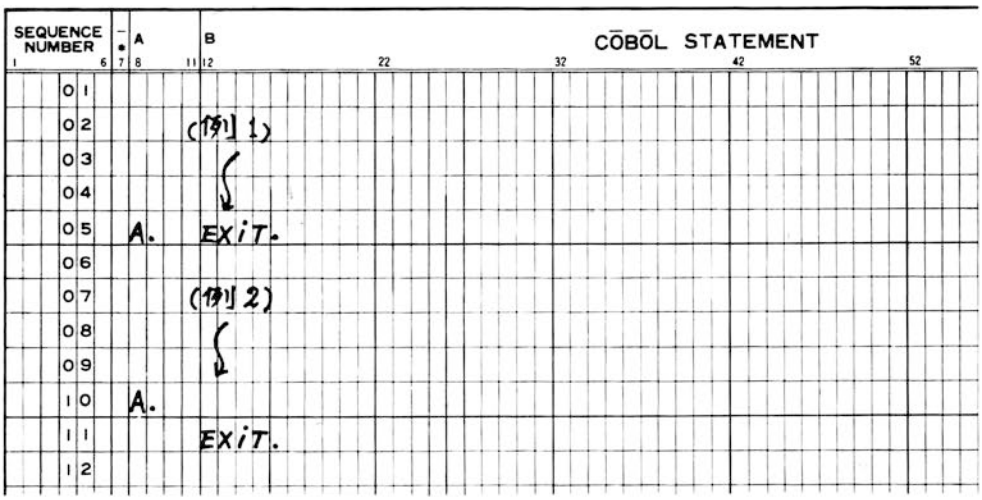
形　式

パラグラフ名　EXIT

1．EXIT命令にはパラグラフ名をつけ，そのパラグラフにはEXITだけを書く.

2．EXIT命令だけで1つの文になっていなければならない.

〔例〕



2. END COBOL

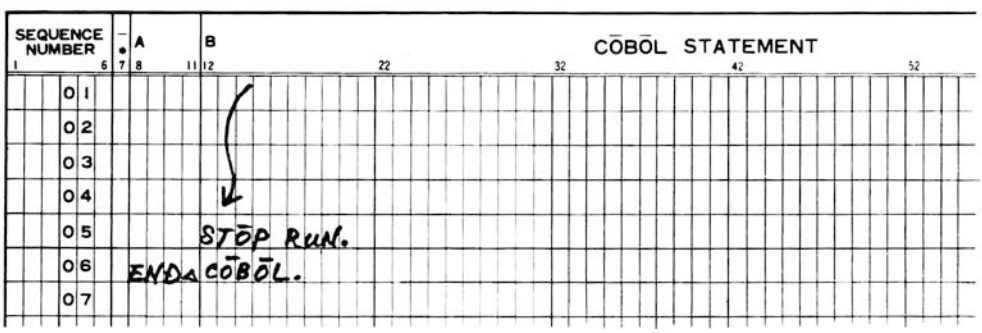
形　式

END COBOL

1．ソースプログラムの最後には必ずEND COBOLを書く.

2．END COBOLは，8カラムから必ず書き始める.

〔例〕



5. ENVIRONMENT DIVISION

5—1　ENVIRONMENT DIVISIONの構成

Environment Divisionは，電子計算機組織の機械的特長についての情報を記述し，Compilerに提供する.

1．Environment Divisionは2つのSectionから構成される．それぞれのSectionは既定の名前をもつParagraphから成る．Division,Section,Paragraphの名前はColumn. 8 (A)から書き最後にピリオド(.)を打たなければならない.

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER.

OBJECT-COMPUTER.
INPUT-OUTPUT SECTION.
FILE-CONTROL.

2．Configuration sectionでは，SourceプログラムをCompileする計算機（Source-Computer）と，Compilerによって作られたObjectプログラム，データを処理するときの計算機（Object-Computer）のModelを教える．

3．Input-Output Sectionは，入出力データがどのような入出力装置によって読み書きされるのかを教える．

5－2　CONFIGURATION SECTION

CONFIGURATION SECTION.
SOURCE-COMPUTER. CEC555.
OBJECT-COMPUTER. CEC555.

1．SOURCE-COMPUTERには，COBOL SourceプログラムをCompileする際に使用するシステムを指定する．

2．OBJECT-COMPUTERには，問題処理のために使用するシステムを指定する．

5－3　INPUT-OUTPUT SECTION

INPUT-OUTPUT SECTION.
FILE-CONTROL.
　SELECTファイル名 ASSIGN TO. 入出力装置.

1．入出力ファイルを使用する場合には，Eile—Control. のSelect～Assign toの節は，必ず書かなくてはならない．また，Select句はCol. 12（B）から書き，最後にピリオド（.）を打たなければならない．

5－3－1　SELECT

1．FIle—Control Paragraphは，この言葉で始まる．ファイル名は入出力ファイルに与えられる名前で一意でなければならない．Data DivisionのFile SectionでFDに記述されるファイル名を全部1つずつSELECTし，必要な節を列記し最後にピリオドを打つ．逆にいうとSELECTされたすべてのファイル名はData DivisionのFDで記述されなければならない．

2．**例**

INPUT—OUTPUT　SECTION.
FILE—CONTROL.
　SELECT MASTER—FILE ASSIGN TO 入出力装置.
　SELECT SUB—FILE ASSIGN TO 入出力装置.

5－3－2　ASSIGN

ASSIGN TO 入出力装置.

1．ASSIGN……は，COBOLプログラマーがSOURCEプログラムの中で使用するファイル名を，Operating System上の名前と結びつけ，これによってデータを特定の記憶媒体に移送するのに適当なRoutineをCompilerが作る．

2．**入出力装置の指定**

PTR—L　　低速紙テープ読取装置
PTR—H　　高速紙テープ読取装置
PTP—L　　低速紙テープ穿孔装置
PTP—H　　高速紙テープ穿孔装置
DRUM—R　磁気ドラム装置（入力用）

DRUM—W　磁気ドラム装置（出力用）
LP　　ラインプリンタ

3．**例**

```
INPUT—OUTPUT SECTION.
FILE—CONTROL.
  SELECT MASTER—FILE ASSIGN TO PTR—H.
  SELECT SUB—FILE ASSIGN TO DRUM—R.
  SELECT OUTPUT ASSIGN TO PTP—L.
```

6．IDENTIFICATION DIVISION

```
IDENTIFICATION DIVISION.
PROGRAM—ID.          プログラム名.
〔AUTHOR.              文.  〕
〔INSTALLATION.        文.  〕
〔DATA—WRITTEN.        文.  〕
〔DATA—COMPILED.       文.  〕
〔SECURITY.            文.  〕
〔REMARKS.             文.  〕
```

1．Identification divisionはCOBOLプログラムの最初のDivisionである．このDivisionはプログラムの表紙ともいうべき部分であり，プログラムに名前をつけたり，プログラム保守に関する注意事項を書きとめたりする．

2．このDivisionには既定のParagraph名をもつ７つのParagraphがあるが，必ず書かなければならない．

ParagraphはPROGRAM—IDだけである．これらのParagraphとDivisionの名前はMargin Aから書く．

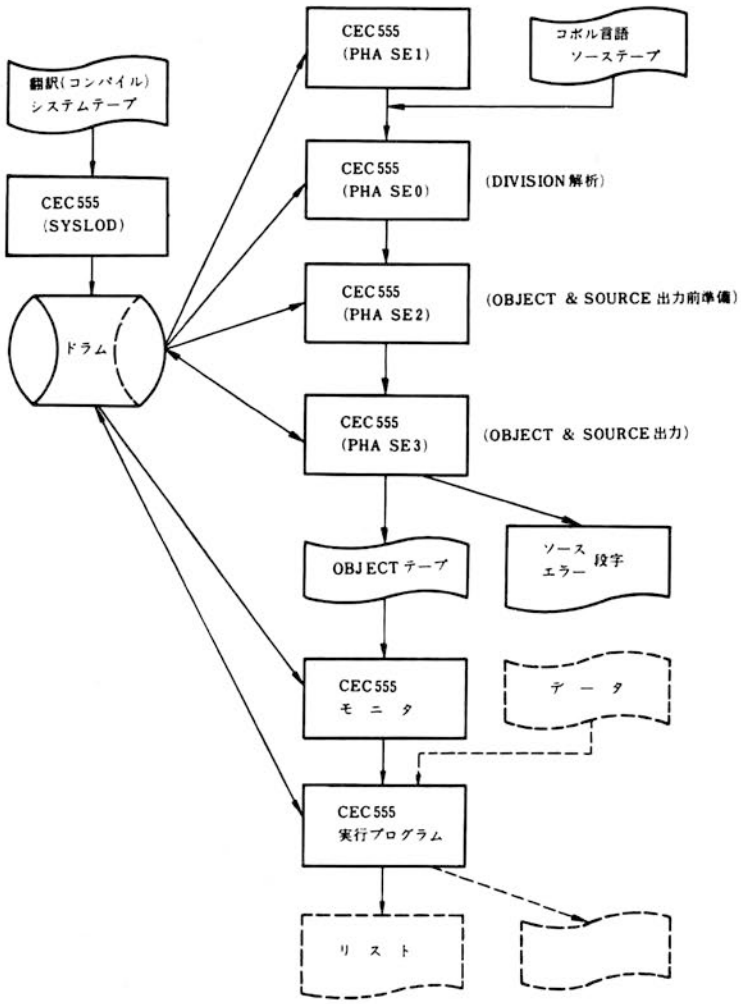
3．PROGRAM—ID.（Program Identification）はIdentification Divisionの中でただ一つ省略できないParagraphであり，かつ常に最初のParagraphである．

4．その他のParagraphはプログラム・リストに印刷されて注釈として役立つだけで，プログラムの実行には直接の関係はなく，書いても書かなくてもよい．

5．**例**

```
IDENTIFICATION DIVISION.
PROGRAM—ID. EXAMPLE—1.
AUTHOR. HANAKO.
INSTALLATION. CEC EDP CENTER.
DATA—WRITTEN. 71—1—01.
```

7. CEC555 COBOL操作法



7—1 コボル翻訳工程

7—2 コボル翻訳プログラムの読込み

コボル翻訳プログラムは，8単位の紙テープにバイナリー形式でパンチされている．コボルの翻訳を行うためには，この翻訳プログラムをSyslod（Compilerをドラム上に格納するプログラム）によりドラムファイル上の所定の位置に格納する．

そのための操作は次のとおりである．

7—2—1 コボル翻訳プログラムのロード手順

1．Rim Loaderをマニュアル入力する．

2．Binary Loaderを紙テープリーダにセットしSR＝7756

Load pc on, Start

3．Syslod（Drum格納プログラム）を紙テープリーダにセットしSR＝7777

Load pc on, 高速リーダ速用時には3777（0ビットOFF）にし，Start

4．コボル翻訳プログラムを紙テープリーダにセットしてSR＝1000

Load pc on, Start

7－3 コボル翻訳プログラムの実行

コボル翻訳プログラムをドラムファイルに格納すると，次の操作によって，翻訳を行うことができる．

7－3－1 翻訳プログラムの実行（COMPILE）

1．Phase 0のプログラムを紙テープリーダにセットし，SR＝7777 Load pc on，高速リーダ使用時は0ビットOFFにしてStart

（Phase 0プログラムはRUN操作等によりこわされない限り再ロード不要）

2．コボルソースプログラムを紙テープリーダにセットし，SR＝7200

Load pc on，Start

3．**その後次のキーを選択しテレタイプ上より入力する．**

OPTION　0……PHASE　1，2，3全て実行
　　1……PHASE　1，3　　実行
　　2……PHASE　1，2　　〃

（上記以外のステップにコントロールが移らない．）

4．PHASE 1 / PHASE 2 / PHASE 3 ― ボタンの選択により異なるがテレタイプ上に出力（自動RUNを行うので操作不要）

OBJECT TAPE，ソースリストおよびエラーリスト出力

END OF JOB

5．翻訳終了

7－4　翻訳の実行における付加操作

操作パネルのデータ部を操作することによりソースリストおよびエラーリスト入出力の選択ができる．

DATA部　0ビット　ON　；低速紙テープ出力
　　OFF；高速　〃
DATA部　10ビット　ON　；ソースリストの省略
　　OFF；　〃　省略せず
DATA部　11ビット　ON　；LINE PRINTER出力
　　OFF；　〃　出力せず

7－5　翻訳済みプログラム実行操作

翻訳され出力された紙テープはオブジェクト形式になっている．（コボル翻訳プログラムの読込みで1，2の操作については何かの原因でこわされない限り不要）

1．出力された紙テープを紙テープリーダにセットしてSR＝7777 Load pc on，高速リーダ使用時は0ビットOFFにしStart

2．コントロールプログラムロード後一担停止，SR＝2000 Load pc on，start

3．テレタイプ上にHalt Option?が出力されたならN，またはYキーを入力する．

N；プログラムロード後停止せずに実行

Y；　〃　停止する．

7－6　エラーリストの見方

CEC 555のコボルで翻訳すれば，もとのプログラムのリストをとる，とらないにかかわらず，CEC 555が自発的に人間の犯したプログラム上の誤りを指摘してくれる仕組みになっている．

プログラム作成上考えられる誤りは，①文法違反，②論理ミスであるが，CEC 555が自発的に誤りを指摘してくれるのは①の方である．

例

０００１１０……30

００１０１０……26

左方はラインナンバー，右方はエラー番号である．

エラー番号は別表を参照されたい．

7－7　コンパイルおよび実行時におけるエラーメッセージ

実行時におけるエラーメッセージ

1．COMPILE FRROR

2．PROGRAM ERROR

3．READ ERROR

4．DRUM I—O ERROR

5．IRREGULAR ACCESS

コンパイル時におけるエラーメッセージ

1．DRUM I—O ERROR

2．I—TABLE OVERFLOW

3．AREA OVERFLOW

4．OBJECT SIZE OVER

注意コード

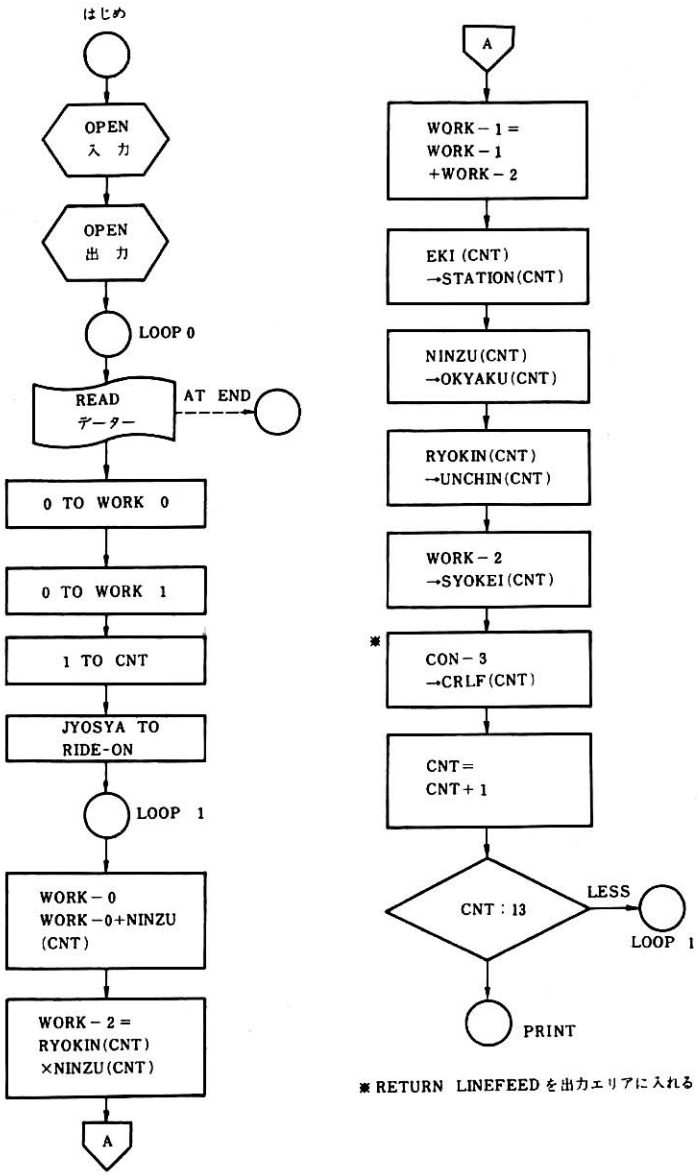
出力コード	内部コード	内　　容
01	1	データの名前の桁数は12桁までという制限を超えている
02	2	1つの数値定数の桁数が21桁までという制限を超えている
03	3	1つの文字定数が24桁までという制限を超えている
05	5	ステートメント数が制限を超過している
10	100	ENVIRONMENT DIVISIONの中で書き方の順序か綴りにおかしいところがある
11	101	ENVIRONMENT DIVISIONの中でASSIGN以下書き方におかしいところがある
12	102	DATA DIVISIONの書き方が違う（FDステートメントなどが消えている）
13	103	DATA DIVISIONのFD以下の書き方がおかしい
15	105	DATA DIVISIONのレベル番号以下の書き方がおかしい
16	106	DATA DIVISIONのPICTUREの書き方がおかしい
19	111	データエリヤの制限を超過した
20	200	命令文の名前の桁数が6桁までという制限を超えている
21	201	PROCEDURE DIVISIONの定数でデータエリヤが超過した
22	202	データの名前（項目の名前）の個数が超過した
23	203	命令文の名前の個数が超過した
24	204	住所不定の名前がでてきた
25	205	ステートメントの書き方がおかしい
26	206	MOVEステートメントの書き方がおかしい
27	207	COMPUTE　〃
28	210	PERFORM　〃
29	211	READ WRITE　〃

30	300	OPEN/CLOSE	〃
31	301	IF	〃
32	302	GO TO	〃
33	303	STOP	〃
34	304	EXIT	〃
35	305	PROCEDURE DIVISIONのステートメント数が超過	
36	306	リテラルの書き方がおかしい	

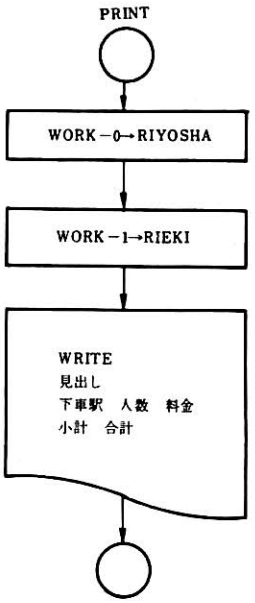
8．CEC555 COBOL例題

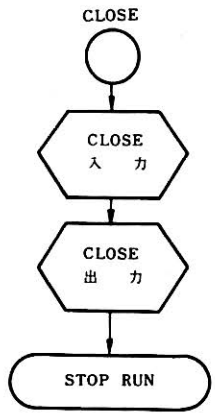
データとして，乗車駅名，下車駅名，乗車駅より下車駅までの区間利用人数と運賃を入力する．
区間ごとの収入と，乗車駅での利用者総数及び総収入を求める．
入力データ形式，( )の中はカラム数
乗車駅名（10）復帰改行
下車駅名1（10）　ブランク（2）　下車人数（7）　ブランク（2）　料金（6）
復帰改行
下車駅名は，12駅あるものとする
出力は次の形式で出力する

```
          カラム1
            ↓              駅　名
          ジョウシャ　エキ　═══
1行あける（
          ゲシャ　エキ　　ニンズウ　　リョウキン　　ショウ　ケイ
1行あける（
            駅名1　　人数×××　　料金×××　　小計×××

            駅名12　　×××　　×××　　×××
1行あける（
          ＊　TOTAL　＊　　リヨウシャ　　×××××　　ソウ　リエキ×××××××
          （
3行あける（
          （
          ジョウシャ　エキ　　駅名
```



※ RETURN LINEFEED を出力エリアに入れる


PRINT
WORK－0→RIYOSHA
WORK－1→RIEKI
WRITE
見出し
下車駅　人数　料金
小計　合計


CLOSE
CLOSE
入　力
CLOSE
出　力
STOP RUN

1．運賃集計計算

```
        ***   CEC-555  COBOL  COMPILE  LIST   ***

SEQ NO          COBOL  SOURCE  PROGRAM                PAGE 001
000010 IDENTIFICATION DIVISION.
000020 PROGRAM-ID.      NEW TOKAIDO LINE.
000030 AUTHOR.                T-KATAGAMI.
000040 DATE-WRITTEN.        1 / 01/ 75.
000050 ENVIRONMENT DIVISION.
000060 CONFIGURASION SECTION.
000070 SOURCE-COMPUTER. CEC555.
000080 OBJECT-COMPUTER. CEC555.
000090 INPUT-OUTPUT SECTION.
000100 FILE-CONTROL.
000110     SELECT PTR-F  ASSIGN TO PTR-H.
000120     SELECT PTP-F  ASSIGN TO PTP-L.
000130 DATA DIVISION.
000140 FILE SECSION.
000150 FD  PTR-F  RECORDING MODE IS S.
000160 LABEL RECORD IS OMITTED.
000170 DATA  RECORD IS INPUT.
000180 01  INPUT.
000190     02  JYOSYA       PICTURE  XXXXXXXXXX.
000200     02  FILLER       PICTURE            XX.
000210     02  DATA-1       OCCURS  12.
000220       03  EKI        PICTURE  XXXXXXXXXX.
000230       03  FILLER     PICTURE            XX.
000240       03  NINZU      PICTURE       9999999.
000250       03  FILLER     PICTURE            XX.
000260       03  RYOKIN     PICTURE        999999.
000270       03  FILLER     PICTURE            XX.
000280 FD  PTP-F  RECORDING MODE IS S.
000290 LABEL RECORD IS OMITTED.
000300 DATA  RECORD IS OUTPUT.
000310 01  OUTPUT.
000320     02  HEADING.
000330       03  FILLER  PICTURE XXXXXXXXXXX VALUE "ｼﾞﾖｳｼﾔ ｴｷ".
000340       03  FILLER  PICTURE            XX.
000350       03  RIDE-ON PICTURE  XXXXXXXXXX.
000360       03  FILLER  PICTURE          XXXX VALUE "!%!%".
000370     02  TYTLE.
000380       03  FILLER  PICTURE  XXXXXXXXXX VALUE " ｹﾞｼﾔ ｴｷ".
000390       03  FILLER  PICTURE          XXXX.
000400       03  FILLER  PICTURE       XXXXXXX VALUE "ﾆﾝｽﾞｳ".
000410       03  FILLER  PICTURE          XXXX.
000420       03  FILLER  PICTURE       XXXXXXX VALUE "ﾘﾖｳｷﾝ".
000430       03  FILLER  PICTURE         XXXXX.
000440       03  FILLER  PICTURE      XXXXXXXX VALUE "ｼﾖｳ ｹｲ".
000450       03  FILLER  PICTURE          XXXX VALUE "!%!%".
000460     02  KEKKA      OCCURS  12.
000470       03  STATION PICTURE  XXXXXXXXXX.
000480       03  FILLER  PICTURE            XX.
000490       03  OKYAKU  PICTURE       ZZZZZZ9.
000500       03  FILLER  PICTURE            XX.
000510       03  UNCHIN  PICTURE        ZZZZZ9.
000520       03  FILLER  PICTURE           XXX.
000530       03  SYOKEI  PICTURE      ZZZZZZZ9.
000540       03  CRLF    PICTURE            XX.
000550     02  TOTAL.
```

```
        ***   CEC-555  COBOL  COMPILE  LIST   ***

SEQ NO          COBOL  SOURCE  PROGRAM              PAGE 002

000560        03  FILLER  PICTURE XXXXXXXXXXX VALUE "!%* TOTAL *".
000570        03  FILLER  PICTURE          XXX.
000580        03  FILLER  PICTURE   XXXXXXXXX VALUE "ﾘｴｷｼｬ  ".
000590        03  RIYOSHA PICTURE   ZZZZZZZZ9.
000600        03  FILLER  PICTURE          XXX.
000610        03  FILLER  PICTURE  XXXXXXXXXX VALUE "ｿｳ ｳﾘｱｹ  ".
000620        03  RIEKI   PICTURE ¥ZZZZZZZZZ9.
000630        03  FILLER  PICTURE    XXXXXXXX VALUE "!%!%!%!%".
000640 WORKING-STORAGE SECTION.
000650 77  CON-0    PICTURE        9999.
000660 77  CON-1    PICTURE        9999  VALUE       0001.
000670 77  CON-2    PICTURE        9999  VALUE       0013.
000680 77  CON-3    PICTURE          XX  VALUE       "!%".
000690 77  WORK-0   PICTURE    99999999.
000700 77  WORK-1   PICTURE    99999999.
000710 77  WORK-2   PICTURE    99999999.
000720 77  CNT      PICTURE        9999.
000730 PROCEDURE DIVISION.
000740     OPEN PTR-F.
000750     OPEN PTP-F.
000760 LOOPO.
000770     READ PTR-F  AT END GO TO CLOS.
000780     MOVE      CON-0         TO  WORK-0.
000790     MOVE      CON-0         TO  WORK-1.
000800     MOVE      CON-1         TO      CNT.
000810     MOVE     JYOSYA         TO RIDE-SN.
000820 LOOP1.
000830     COMPUTE WORK-0 =  WORK-0        +    NINZU (CNT).
000840     COMPUTE WORK-2 =  RYOKIN (CNT)  *    NINZU (CNT).
000850     COMPUTE WORK-1 =  WORK-1        +   WORK-2.
000860     MOVE        EKI (CNT)  TO STATION (CNT).
000870     MOVE      NINZU (CNT)  TO  OKYAKU (CNT).
000880     MOVE     RYOKIN (CNT)  TO  UNCHIN (CNT).
000890     MOVE     WORK-2        TO  SYOKEI (CNT).
000900     MOVE      CON-3        TO    CRLF (CNT).
000910     COMPUTE     CNT =      CNT       +    CON-1.
000920     IF CNT IS  LESS   THAN CON-2 GO TO LOOP1.
000930 PRINT.
000940     MOVE     WORK-0         TO RIYOSHA.
000950     MOVE     WORK-1         TO   RIEKI.
000960     WRITE OUTPUT.
000970     GO TO LOOPO.
000980 CLOS.
000990     CLOSE PTR-F.
001000     CLOSE PTP-F.
001010     STOP RUN.
001020 END COBOL.


        **  0000 ERRORS  **

        *** CEC-555 COBOL END ***

        * END OF JOB

        HALT OPTION? Y
```

ｼﾞﾖｳｼﾔ ｴｷ　　ﾄｳｷﾖｳ

ｹﾞｼﾔ ｴｷ	ﾆﾝｽﾞｳ	ﾘﾖｳｷﾝ	ｼﾖｳ ｹｲ
ｼﾝ ﾖｺﾊﾏ	98	520	50960
ｵﾀﾞﾜﾗ	102	1060	108120
ｱﾀﾐ	111	1170	129870
ﾐｼﾏ	48	1250	60000
ｼｽﾞｵｶ	81	1500	121500
ﾊﾏﾏﾂ	73	1950	142350
ﾄﾖﾊｼ	50	2120	106000
ﾅｺﾞﾔ	128	2660	340480
ｷﾞﾌ ﾊｼﾏ	67	2740	183580
ﾏｲﾊﾞﾗ	32	3190	102080
ｷﾖｳﾄ	100	3650	365000
ｼﾝ ｵｵｻｶ	198	3730	738540

＊ TOTAL ＊　ﾘﾖｳｼﾔ　1088　ｿｳ ﾘｴｷ　¥2448480

ｼﾞﾖｳｼﾔ ｴｷ　ｼﾝ ﾖｺﾊﾏ

ｹﾞｼﾔ ｴｷ	ﾆﾝｽﾞｳ	ﾘﾖｳｷﾝ	ｼﾖｳ ｹｲ
ﾄｳｷﾖｳ	87	520	45240
ｵﾀﾞﾜﾗ	72	640	46080
ｱﾀﾐ	28	1020	28560
ﾐｼﾏ	7	1100	7700
ｼｽﾞｵｶ	34	1330	45220
ﾊﾏﾏﾂ	22	1870	41140
ﾄﾖﾊｼ	13	2040	26520
ﾅｺﾞﾔ	41	2490	102090
ｷﾞﾌ ﾊｼﾏ	18	2660	47880
ﾏｲﾊﾞﾗ	14	3030	42420
ｷﾖｳﾄ	55	3360	184800
ｼﾝ ｵｵｻｶ	118	3650	430700

＊ TOTAL ＊　ﾘﾖｳｼﾔ　509　ｿｳ ﾘｴｷ　¥1048350

ｼﾞﾖｳｼﾔ ｴｷ　ｵﾀﾞﾜﾗ

ｹﾞｼﾔ ｴｷ	ﾆﾝｽﾞｳ	ﾘﾖｳｷﾝ	ｼﾖｳ ｹｲ
ﾄｳｷﾖｳ	124	1060	131440
ｼﾝ ﾖｺﾊﾏ	52	640	33280
ｱﾀﾐ	33	500	16600
ﾐｼﾏ	9	860	7740
ｼｽﾞｵｶ	27	1100	29700
ﾊﾏﾏﾂ	14	1420	19880
ﾄﾖﾊｼ	10	1790	17900
ﾅｺﾞﾔ	38	2120	80560
ｷﾞﾌ ﾊｼﾏ	23	2410	55430
ﾏｲﾊﾞﾗ	14	2660	37240
ｷﾖｳﾄ	29	3110	90190
ｼﾝ ｵｵｻｶ	91	3280	298480

＊ TOTAL ＊　ﾘﾖｳｼﾔ　434　ｿｳ ﾘｴｷ　¥818340

ｼﾞﾖｳｼﾔ ｴｷ　　アタミ

ｹﾞｼﾔ ｴｷ	ﾆﾝｽﾞｳ	ﾘﾖｳｷﾝ	ｼﾖｳ ｹｲ
ﾄｳｷﾖｳ	181	1170	211770
ｼﾝ ﾖｺﾊﾏ	48	1020	48960
ｵﾀﾞﾜﾗ	22	500	11000
ﾐｼﾏ	6	480	2880
ｼｽﾞｵｶ	17	1020	17340
ﾊﾏﾏﾂ	15	1330	19950
ﾄﾖﾊｼ	9	1500	13500
ﾅｺﾞﾔ	63	2040	128520
ｷﾞﾌ ﾊｼﾏ	27	2120	57240
ﾏｲﾊﾞﾗ	21	2570	53970
ｷﾖｳﾄ	30	3030	90900
ｼﾝ ｵｵｻｶ	103	3190	328570

＊ TOTAL ＊　ﾘﾖｳｼﾔ　542　ｿｳ ﾘｴｷ　¥994600

ｼﾞﾖｳｼﾔ ｴｷ　　ﾐｼﾏ

ｹﾞｼﾔ ｴｷ	ﾆﾝｽﾞｳ	ﾘﾖｳｷﾝ	ｼﾖｳ ｹｲ
ﾄｳｷﾖｳ	100	1220	122000
ｼﾝ ﾖｺﾊﾏ	32	1100	35200
ｵﾀﾞﾜﾗ	32	860	27520
ｱﾀﾐ	24	480	11520
ｼｽﾞｵｶ	13	640	8320
ﾊﾏﾏﾂ	7	1250	8750
ﾄﾖﾊｼ	5	1420	7100
ﾅｺﾞﾔ	63	1950	122850
ｷﾞﾌ ﾊｼﾏ	19	2040	38760
ﾏｲﾊﾞﾗ	8	2490	19920
ｷﾖｳﾄ	56	2740	153440
ｼﾝ ｵｵｻｶ	78	3110	242580

＊ TOTAL ＊　ﾘﾖｳｼﾔ　437　ｿｳ ﾘｴｷ　¥797960

ｼﾞﾖｳｼﾔ ｴｷ　　ｼｽﾞｵｶ

ｹﾞｼﾔ ｴｷ	ﾆﾝｽﾞｳ	ﾘﾖｳｷﾝ	ｼﾖｳ ｹｲ
ﾄｳｷﾖｳ	82	1500	123000
ｼﾝ ﾖｺﾊﾏ	18	1330	23940
ｵﾀﾞﾜﾗ	11	1100	12100
ｱﾀﾐ	22	1020	22440
ﾐｼﾏ	4	640	2560
ﾊﾏﾏﾂ	17	720	12240
ﾄﾖﾊｼ	2	1170	2340
ﾅｺﾞﾔ	31	1500	46500
ｷﾞﾌ ﾊｼﾏ	11	1790	19690
ﾏｲﾊﾞﾗ	10	2040	20400
ｷﾖｳﾄ	40	2490	99600
ｼﾝ ｵｵｻｶ	59	2660	156940

＊ TOTAL ＊　ﾘﾖｳｼﾔ　307　ｿｳ ﾘｴｷ　¥541750

SIN，COS，TANの計算例

```
        ***   CEC-555  COBOL  COMPILE  LIST   ***

 SEQ NO          COBOL  SOURCE  PROGRAM                PAGE 001

000010 IDENTIFICATION DIVISION.
000020 PROGRAM-ID.      SANKAK-KANSU HYOU.
000030 AUTHOR.          T-KATAGAMI.
000040 DATE-WRITTEN.    1/1/75
000050 REMARKS.         BUSINESS COMPUTING CENTER.
000060 ENVIRONMENT DIVISION.
000070 CONFIG ATION SECTION.
000080 SOURCE-COMPUTER. CEC555.
000090 OBJECT-COMPUTER. CEC555.
000100 INPUT-OUTPUT SECTION.
000110 FILE-CONTROL.
000120     SELECT  PTP-F  ASSIGN TO PTP-L.
000130 DATA DIVISION.
000140 FILE SECTION.
000150 FD  PTP-F  RECORDING MODE IS S.
000160 LABEL RECORD IS OMITTED.
000170 DATA  RECORD IS PTP-W.
000180 01  PTP-W.
000190     02  OUT-1  PICTURE  XXX.
000200     02  DUMY-1 PICTURE  XXX.
000210     02  OUT-2  PICTURE  XXXXXXXX.
000220     02  DUMY-2 PICTURE  XXXXXXXX.
000230     02  OUT-3  PICTURE  XXXXXXX.
000240     02  DUMY-3 PICTURE  XXXXXXX.
000250     02  OUT-4  PICTURE  XXXXXXX.
000260     02  DUMY-4 PICTURE  XXXXXXX.
000270     02  OUT-5  PICTURE  XXXXXXXXX.
000280     02  DUMY-5 PICTURE  XXXXXXXXX.
000290     02  FILLER PICTURE  XX.
000300 WORKING-STORAGE SECTION.
000310 77  CON-1       PICTURE  S9V99999     VALUE  1.57080.
000320 77  CON-2       PICTURE  S9V99999     VALUE -0.64596.
000330 77  CON-3       PICTURE  S9V99999     VALUE  0.07969.
000340 77  CON-4       PICTURE  S9V99999     VALUE -0.00467.
000350 77  CON-5       PICTURE  S9V99999     VALUE  0.00016.
000360 77  X-DIG       PICTURE  999          VALUE   000.
000370 77  X-RAD       PICTURE  99V99999.
000380 77  X-INC       PICTURE  9            VALUE   1.
000390 77  SIN-X       PICTURE  9V99999.
000400 77  COS-X       PICTURE  9V99999.
000410 77  TAN-X       PICTURE  999V99999.
000420 77  PAI         PICTURE  9V99999      VALUE   3.14159.
000430 77  X-TPI       PICTURE  99V99999.
000440 77  X-PEN       PICTURE  9999V99999.
000450 77  X-HEN       PICTURE  9999V99999.
000460 77  X-NOW       PICTURE  9999V99999.
000470 77  WORK-1      PICTURE  99999999999999V99999.
000480 77  WORK-2      PICTURE  99999999999999V99999.
000490 77  WORK-3      PICTURE  99999999999999V99999.
000500 77  WORK-4      PICTURE  99999999999999V99999.
000510 77  WORK-5      PICTURE  99999999999999V99999.
000520 77  WORK-6      PICTURE  99999999999999V99999.
000530 77  CON-6       PICTURE  9999         VALUE 0180.
000540 77  CON-7       PICTURE  99           VALUE 02.
000550 77  CON-8       PICTURE  999          VALUE 090.
```

```
        ***   CEC-555  COBOL  COMPILE  LIST   ***

SEQ NO          COBOL  SOURCE  PROGRAM                PAGE 002

000560 77  CNT-1       PICTURE  999          VALUE 010.
000570 77  CNT-2       PICTURE  999.
000580 77  CNT-3       PICTURE  999          VALUE 000.
000590 77  BLANK       PICTURE  XXXXXXXXXXXXXXXX.
000600 01  CLEAR.
000610      02  FILLER PICTURE  XXX.
000620      02  FILLER PICTURE  XXX.
000630      02  FILLER PICTURE  XXXXXXXX.
000640      02  FILLER PICTURE  XXXXXXXX.
000650      02  FILLER PICTURE  XXXXXXX.
000660      02  FILLER PICTURE  XXXXXXX.
000670      02  FILLER PICTURE  XXXXXXX.
000680      02  FILLER PICTURE  XXXXXXX.
000690      02  FILLER PICTURE  XXXXXXXXX.
000700      02  FILLER PICTURE  XXXXXXXXX.
000710      02  FILLER PICTURE  XX           VALUE "!%".
000720 01  MES-1.
000730      02  FILLER PICTURE  XXX          VALUE "  X".
000740      02  FILLER PICTURE  XXX.
000750      02  FILLER PICTURE  XXXXXXXX     VALUE "X-RADIAN".
000760      02  FILLER PICTURE  XXXXXXXX.
000770      02  FILLER PICTURE  XXXXXXX      VALUE "  SIN  ".
000780      02  FILLER PICTURE  XXXXXXX.
000790      02  FILLER PICTURE  XXXXXXX      VALUE "  COS  ".
000800      02  FILLER PICTURE  XXXXXXX.
000810      02  FILLER PICTURE  XXXXXXXXX    VALUE "   TAN    ".
000820      02  FILLER PICTURE  XXXXXXXXX.
000830      02  FILLER PICTURE  XX           VALUE "!%".
000840 01  MES-2.
000850      02  OPT-1   PICTURE  Z79.
000860      02  OPT-2   PICTURE  Z9.99999.
000870      02  OPT-3   PICTURE  9.99999.
000880      02  OPT-4   PICTURE  9.99999.
000890      02  OPT-5   PICTURE  Z79.99999.
000900      02  FILLER PICTURE  XX           VALUE "!%".
000910 PROCEDURE DIVISION.
000920      OPEN PTP-F.
000930 START.
000940      MOVE  CLEAR TO PTP-W.
000950      WRITE PTP-W.
000960      MOVE  MES-1 TO PTP-W.
000970      WRITE PTP-W.
000980      MOVE  CLEAR TO PTP-W.
000990      WRITE PTP-W.
001000      MOVE  CNT-1 TO CNT-2.
001010 LOOP1.
001020      COMPUTE  X-RAD = X-DIG / CON-6.
001030      COMPUTE  X-RAD = X-RAD * PAI.
001040      MOVE  X-DIG TO OPT-1.
001050      MOVE  X-RAD TO OPT-2.
001060      COMPUTE  X-RAD = X-RAD * CON-7.
001070      COMPUTE  X-RAD = X-RAD / PAI.
001080      COMPUTE  WORK-1 = CON-1 * X-RAD.
001090      COMPUTE  X-TRI = X-RAD * X-RAD.
001100      COMPUTE  X-TRI = X-TRI * X-RAD.
```

```
001110      COMPUTE  WORK-2 = CON-2 * X-TRI.
001120      COMPUTE  X-PEN  = X-TRI * X-RAD.
001130      COMPUTE  X-PEN  = X-PEN * X-RAD.
001140      COMPUTE  WORK-3 = CON-3 * X-PEN.
001150      COMPUTE  X-HEP  = X-PEN * X-RAD.
001160      COMPUTE  X-HEP  = X-HEP * X-RAD.
001170      COMPUTE  WORK-4 = CON-4 * X-HEP.
001180      COMPUTE  X-NON  = X-HEP * X-RAD.
001190      COMPUTE  X-NON  = X-NON * X-RAD.
001200      COMPUTE  WORK-5 = CON-5 * X-NON.
001210      COMPUTE  WORK-6 = WORK-1 + WORK-2.
001220      COMPUTE  WORK-6 = WORK-6 + WORK-3.
001230      COMPUTE  WORK-6 = WORK-6 + WORK-4.
001240      COMPUTE  SIN-X  = WORK-6 + WORK-5.
001250      MOVE  SIN-X TO OPT-3.
001260      COMPUTE  X-RAD  = X-DIG / CON-6.
001270      COMPUTE  X-RAD  = X-RAD * PAI.
001280      COMPUTE  WORK-6 = PAI   / CON-7.
001290      COMPUTE  X-RAD  = WORK-6 - X-RAD.
001300      COMPUTE  X-RAD  = X-RAD * CON-7.
001310      COMPUTE  X-RAD  = X-RAD / PAI.
001320      COMPUTE  WORK-1 = CON-1 * X-RAD.
001330      COMPUTE  X-TRI  = X-RAD * X-RAD.
001340      COMPUTE  X-TRI  = X-TRI * X-RAD.
001350      COMPUTE  WORK-2 = CON-2 * X-TRI.
001360      COMPUTE  X-PEN  = X-TRI * X-RAD.
001370      COMPUTE  X-PEN  = X-PEN * X-RAD.
001380      COMPUTE  WORK-3 = CON-3 * X-PEN.
001390      COMPUTE  X-HEP  = X-PEN * X-RAD.
001400      COMPUTE  X-HEP  = X-HEP * X-RAD.
001410      COMPUTE  WORK-4 = CON-4 * X-HEP.
001420      COMPUTE  X-NON  = X-HEP * X-RAD.
001430      COMPUTE  X-NON  = X-NON * X-RAD.
001440      COMPUTE  WORK-5 = CON-5 * X-NON.
001450      COMPUTE  WORK-6 = WORK-1 + WORK-2.
001460      COMPUTE  WORK-6 = WORK-6 + WORK-3.
001470      COMPUTE  WORK-6 = WORK-6 + WORK-4.
001480      COMPUTE  COS-X  = WORK-6 + WORK-5.
001490      MOVE  COS-X TO OPT-4.
001500      COMPUTE  TAN-X  = SIN-X / COS-X.
001510      MOVE  TAN-X TO OPT-5.
001520      MOVE  OPT-1 TO OUT-1.
001530      MOVE  OPT-2 TO OUT-2.
001540      MOVE  OPT-3 TO OUT-3.
001550      MOVE  OPT-4 TO OUT-4.
001560      MOVE  OPT-5 TO OUT-5.
001570      WRITE PTP-W.
001580      COMPUTE  X-DIG  = X-DIG + X-INC.
001590      IF  X-DIG IS GREATER THAN CON-8  GO TO OWARI.
001600      COMPUTE  CNT-2  = CNT-2 - X-INC.
001610      IF  CNT-2 IS GREATER THAN CNT-3  GO TO LOOP1.
001620      MOVE  CLEAR TO PTP-W.
001630      WRITE PTP-W.
001640      GO TO START.
```

```
001650 OWARI.
```

```
       ***   CEC-555  COBUL  COMPILE  LIST   ***

SEQ NO           CCPOL  SOURCE  PROGRAM                PAGE 004

001660      CLOSE PTP-F.
001670      STOP RUN.
001680 END COBOL.
```

X	X-RADIAN	SIN	COS	TAN
0	0.00000	0.00000	1.00001	0.00000
1	0.01743	0.01742	0.99987	0.01742
2	0.03490	0.03488	0.99942	0.03490
3	0.05233	0.05231	0.99864	0.05238
4	0.06980	0.06974	0.99758	0.06990
5	0.08724	0.08712	0.99621	0.08745
6	0.10470	0.10451	0.99454	0.10508
7	0.12214	0.12183	0.99257	0.12274
8	0.13961	0.13914	0.99028	0.14050
9	0.15707	0.15643	0.98770	0.15837

X	X-RADIAN	SIN	COS	TAN
10	0.17451	0.17362	0.98482	0.17629
11	0.19198	0.19079	0.98163	0.19436
12	0.20941	0.20788	0.97817	0.21251
13	0.22688	0.22493	0.97437	0.23084
14	0.24432	0.24186	0.97031	0.24928
15	0.26178	0.25879	0.96594	0.26791
16	0.27922	0.27559	0.96127	0.28669
17	0.29669	0.29234	0.95632	0.30569
18	0.31415	0.30900	0.95107	0.32489
19	0.33159	0.32554	0.94552	0.34429

X	X-RADIAN	SIN	COS	TAN
20	0.34906	0.34200	0.93969	0.36394
21	0.36649	0.35834	0.93360	0.38382
22	0.38396	0.37458	0.92720	0.40399
23	0.40140	0.39069	0.92052	0.42442
24	0.41886	0.40671	0.91356	0.44519
25	0.43630	0.42258	0.90631	0.46626
26	0.45377	0.43835	0.89881	0.48770
27	0.47123	0.45398	0.89101	0.50951
28	0.48867	0.46945	0.88295	0.53168
29	0.50614	0.48479	0.87462	0.55428

X	X-RADIAN	SIN	COS	TAN
30	0.52357	0.49997	0.86604	0.57730
31	0.54104	0.51503	0.85716	0.60085
32	0.55848	0.52989	0.84806	0.62482
33	0.57594	0.54462	0.83868	0.64937
34	0.59338	0.55916	0.82905	0.67445
35	0.61085	0.57355	0.81917	0.70015
36	0.62831	0.58778	0.80902	0.72653
37	0.64575	0.60180	0.79864	0.75353
38	0.66322	0.61564	0.78801	0.78125
39	0.68065	0.62930	0.77717	0.80973

X	X-RADIAN	SIN	COS	TAN
40	0.69812	0.64279	0.76605	0.83909
41	0.71555	0.65604	0.75473	0.86923
42	0.73302	0.66912	0.74316	0.90037
43	0.75046	0.68197	0.73137	0.93245
44	0.76793	0.69464	0.71934	0.96566
45	0.78539	0.70710	0.70712	0.99997
46	0.80283	0.71932	0.69466	1.03549
47	0.82030	0.73135	0.68200	1.07236
48	0.83773	0.74313	0.66916	1.11054
49	0.85520	0.75471	0.65606	1.15036

X	X-RADIAN	SIN	COS	TAN
50	0.87263	0.76602	0.64282	1.19165
51	0.89010	0.77714	0.62933	1.23486
52	0.90754	0.78798	0.61566	1.27989
53	0.92500	0.79863	0.60183	1.32700
54	0.94247	0.80901	0.58779	1.37635
55	0.95991	0.81915	0.57358	1.42813
56	0.97738	0.82904	0.55919	1.48257
57	0.99481	0.83865	0.54466	1.53976
58	1.01228	0.84804	0.52992	1.60031
59	1.02971	0.85715	0.51505	1.66420

X	X-RADIAN	SIN	COS	TAN
60	1.04718	0.86602	0.50002	1.73197
61	1.06462	0.87460	0.48482	1.80396
62	1.08208	0.88294	0.46949	1.88063
63	1.09955	0.89100	0.45399	1.96259
64	1.11699	0.89879	0.43838	2.05025
65	1.13445	0.90630	0.42262	2.14447
66	1.15189	0.91352	0.40676	2.24584
67	1.16936	0.92050	0.39072	2.35590
68	1.18679	0.92717	0.37462	2.47496
69	1.20426	0.93358	0.35838	2.60500

X	X-RADIAN	SIN	COS	TAN
70	1.22170	0.93967	0.34203	2.74733
71	1.23916	0.94551	0.32558	2.90407
72	1.25663	0.95107	0.30902	3.07769
73	1.27407	0.95631	0.29237	3.27088
74	1.29153	0.96126	0.27564	3.48737
75	1.30897	0.96593	0.25882	3.73205
76	1.32644	0.97031	0.24191	4.01103
77	1.34387	0.97437	0.22497	4.33111
78	1.36134	0.97815	0.20793	4.70422
79	1.37876	0.98161	0.19082	5.14416

λ	λ-RADIAN	SIN	COS	TAN
80	1.39624	0.98482	0.17366	5.67096
81	1.41371	0.98770	0.15644	6.31360
82	1.43115	0.99027	0.13917	7.11554
83	1.44861	0.99255	0.12188	8.14366
84	1.46605	0.99454	0.10454	9.51348
85	1.48352	0.99622	0.08715	11.43109
86	1.50095	0.99758	0.06978	14.29607
87	1.51842	0.99864	0.05234	19.07986
88	1.53586	0.99941	0.03491	28.62818
89	1.55332	0.99987	0.01746	57.26632

λ	λ-RADIAN	SIN	COS	TAN
90	1.57079	1.00001	0.00000	0.00000

加、減、乘、除算例

```
        ***   CEC-555  COBOL  COMPILE  LIST   ***

SEQ NO          COBOL  SOURCE  PROGRAM                PAGE 001

000010 IDENTIFICATION DIVISION.
000020 PROGRAM-ID. TEST0.
000030 ENVIRONMENT DIVISION.
000040 CONFIGURATION SECTION.
000050 SOURCE-COMPUTER. CEC555.
000060 OBJECT-COMPUTER. CEC555.
000070 INPUT-OUTPUT SECTION.
000080 FILE-CONTROL.
000090     SELECT PTR-F ASSIGN TO PTR-H.
000100     SELECT PTP-F ASSIGN TO PTP-L.
000110 DATA DIVISION.
000120 FILE SECTION.
000130 FD  PTR-F RECORDING MODE IS S.
000140 LABEL RECORD IS OMITTED.
000150 DATA RECORD IS PTR-R.
000160 01  PTR-R.
000170     02  AA1          PICTURE XXXXX.
000180     02  AA2          PICTURE 999.
000190     02  AA3          PICTURE 999.
000200     02  AA4          PICTURE 999.
000210     02  AA5          PICTURE 999.
000220     02  AA6          PICTURE XX.
000230 FD  PTP-F RECORDING MODE IS S.
000240 LABEL RECORD IS OMITTED.
000250 DATA RECORD IS PTP-R.
000260 01  PTP-R.
000270     02  BB1           PICTURE XXXXX.
000280     02  FILLER        PICTURE XX.
000290     02  BB2           PICTURE XXX.
000300     02  FILLER        PICTURE XX.
000310     02  BB3           PICTURE XXX.
000320     02  FILLER        PICTURE XX.
000330     02  BB4           PICTURE XXX.
000340     02  FILLER        PICTURE XX.
000350     02  BB5           PICTURE XXX.
000360     02  FILLER        PICTURE XX.
000370     02  BB6           PICTURE XXXXXXX.
000380     02  BB7           PICTURE XX     VALUE "!%".
000390 WORKING-STORAGE SECTION.
000400 77  WORK5             PICTURE 99999.
000410 77  WORK6             PICTURE 99999.
000420 77  CONO              PICTURE 9.
000430 77  CNT               PICTURE 99.
000440 77  CMP               PICTURE 99     VALUE 10.
000450 77  CONGR             PICTURE XX     VALUE "!%".
000460 77  CON1              PICTURE 99     VALUE 01.
000470 77  CONA              PICTURE XXX    VALUE "END".
000480 77  CONB              PICTURE XXX    VALUE "OF ".
000490 77  CONC              PICTURE XXX    VALUE "LST".
000500 77  WORK2             PICTURE -ZZZZZ9.
000510 77  WORK3             PICTURE ZZ9.
000520 01  MSG3.
000530     02  FILLER       PICTURE XXXXX.
000540     02  FILLER       PICTURE XX.
000550     02  FILLER       PICTURE XXX.
```

```
000560      02  FILLER       PICTURE XX.
000570      02  FILLER       PICTURE XXX.
000580      02  FILLER       PICTURE XX.
000590      02  FILLER       PICTURE XXX.
000600      02  FILLER       PICTURE XX.
000610      02  FILLER       PICTURE XXX.
000620      02  FILLER       PICTURE XX.
000630      02  FILLER       PICTURE XXXXXXX.
000640      02  FILLER       PICTURE XX     VALUE "1%".
000650 01   MSG2.
000660      02  FILLER       PICTURE XXXXX VALUE "NAME ".
000670      02  FILLER       PICTURE XX.
000680      02  FILLER       PICTURE XXX    VALUE "DT1".
000690      02  FILLER       PICTURE XX.
000700      02  FILLER       PICTURE XXX    VALUE "DT2".
000710      02  FILLER       PICTURE XX.
000720      02  FILLER       PICTURE XXX    VALUE "DT3".
000730      02  FILLER       PICTURE XX.
000740      02  FILLER       PICTURE XXX    VALUE "DT4".
000750      02  FILLER       PICTURE XX.
000760      02  FILLER       PICTURE XXXXXXX VALUE " T0TAL".
000770      02  FILLER       PICTURE XX     VALUE "1%".
000780 01   MSG1.
000790      02  FILLER       PICTURE XXXXX.
000800      02  FILLER       PICTURE XX     VALUE "CE".
000810      02  FILLER       PICTURE XXX    VALUE "C55".
000820      02  FILLER       PICTURE XX     VALUE "S ".
000830      02  FILLER       PICTURE XXX    VALUE "C0B".
000840      02  FILLER       PICTURE XX     VALUE "0L".
000850      02  FILLER       PICTURE XXX    VALUE " TE".
000860      02  FILLER       PICTURE XX     VALUE "ST".
000870      02  FILLER       PICTURE XXX.
000880      02  FILLER       PICTURE XX.
000890      02  FILLER       PICTURE XXXXXXX.
000900      02  FILLER       PICTURE XX     VALUE "1%".
000910 PR0CEDURE DIVISI0N.
000920 START.
000930      0PEN  PTR-F.
000940      0PEN  PTP-F.
000950 XXX.
000960      M0VE  C0N0  T0 CNT.
000970      M0VE  MSG3  T0 PTP-R.
000980      WRITE PTP-R.
000990      WRITE PTP-R.
001000      M0VE  MSG1  T0 PTP-R.
001010      WRITE PTP-R.
001020      M0VE  MSG2  T0 PTP-R.
001030      WRITE PTP-R.
001040      M0VE  MSG3  T0 PTP-R.
001050      WRITE PTP-R.
001060 R1.
001070      READ  PTR-F AT END  G0 T0 END1.
001080      M0VE  C0N0  T0 W0RK5.
001090      C0MPUTE  W0RK5 = AA2 - AA3.
001100      C0MPUTE  W0RK5 = W0RK5 * AA4.
```

```
      ***   CEC-555  COBOL  COMPILE  LIST   ***

SEQ NO          COBOL  SOURCE  PROGRAM              PAGE 003

001110      COMPUTE  WORK5 = WORK5 / AA5.
001120      COMPUTE  WORK6 = WORK5 + WORK6.
001130      MOVE  AA1  TO BB1.
001140      MOVE  AA2  TO WORK3.
001150      MOVE  WORK3  TO BB2.
001160      MOVE  AA3    TO WORK3.
001170      MOVE  WORK3  TO BB3.
001180      MOVE  AA4    TO WORK3.
001190      MOVE  WORK3  TO BB4.
001200      MOVE  AA5    TO WORK3.
001210      MOVE  WORK3  TO BB5.
001220      MOVE  WORK5  TO WORK2.
001230      MOVE  WORK2  TO BB6.
001240      WRITE PTP-R.
001250      COMPUTE  CNT = CON1 + CNT.
001260      IF  CNT EQUAL CMP  GO TO XXX.
001270      GO TO R1.
001280 END1.
001290      MOVE  MSG3  TO PTP-R.
001300      MOVE  CONA  TO BB2.
001310      MOVE  CONB  TO BB3.
001320      MOVE  CONC  TO BB4.
001330      WRITE PTP-R.
001340      CLOSE PTP-F.
001350      CLOSE PTR-F.
001360      STOP RUN.
001370 END COBOL.



**  0000 ERRORS   **

*** CEC-555 COBOL END ***

* END OF JOB
```

```
HALT OPTION? Y


     CEC555 COBOL TEST
NAME    DT1   DT2   DT3   DT4     TOTAL

AAAA1   100    90    30     2       150
AAAA2   200   300    20    10   -   200
AAAA3   234   123    22    33        74
AAAA4   567     2     3     4       423
AAAA3     2     0   500    20        50
AAAA4.  333   222    11     1      1221
AAAA5   445     5     3     4       330
AAAA6   345   123    99    20      1098
AAAA7   678   999   200    10   -  6420
AAAA8   900   300   100   200       300


     CEC555 COBOL TEST
NAME    DT1   DT2   DT3   DT4     TOTAL

AAAA9   465   111    11     4       973
AAAAO   565   555   222     3       740
BBBB1   876   435    43     2      9481
BBBB2   657   444    33    40       175
BBBB3    23    11     1     0         0
BBBB4   957   345   100    22      2781
BBBB5   867   435    43    77       241
BBBB6   659   111    99    33      1644
BBBB7   253   999    32     1   - 23872
BBBB8   666   333   222    11      6720


     CEC555 COBOL TEAT
NAME    DT1   DT2   DT3   DT4     TOTAL

BBBB9   645   243    87     4      8743
BBBBC    34   867    44    22   -  1666
CCCC1   756   333    67    40       708
CCCC2   555   444    22    11       222
CCCC3   666   611   101    32       173
CCCC4   456   123    56    31       601
CCCC5   857   222    25     2      7937
        END   OF   LST
```

⑫ リアルタイムモニタ

1. 概　要

CEC555リアルタイムモニタシステムは，複数個のタスクをユーザが指定する優先度にしたがって管理をおこない，リアルタイムで実行を要求される仕事を含めて有効なリソースの管理を行うシステムプログラムである.

本モニタシステムは，CEC555マークⅡフォートラン，SAPDアセンブラ，ココル（対話計算），エディタをはじめとするマークⅡモニタシステムに含まれるすべてのシステムプログラムをサポートするとともに，マークⅡモニタシステムに含まれるすべての周辺機器を管理することができる.

入出力機器としてはCS—100，CS—1200入出力タイプライタ，SS—408高速紙テープリーダ，CS—420高速紙テープパンチャ，CLP—4100ラインプリンタが含まれる.（図1）

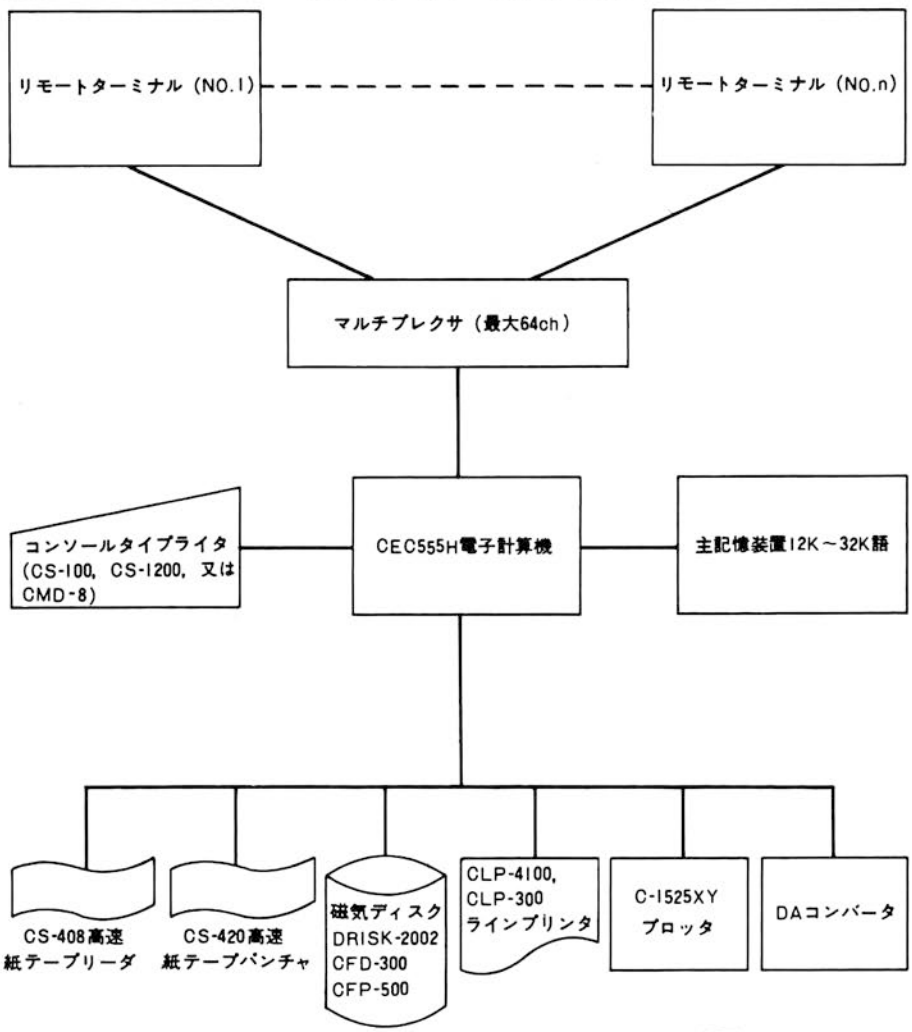


図 1　CEC555Hリアルタイムモニタシステム　構成図

補助記憶装置としては，DRISK-2002磁気ディスク装置（可動ヘッド型），CFD—300ミニディスク装置（固定ヘッド型），CFP—500フロッピーディスクが使用できる．

最小システム構成は，システム記憶装置として上記の3種のなかの，任意の補助記憶装置を含むCEC555H電子計算機（主記憶装置12K語以上），CS—100入出力タイプライタ，CS—1200入出力タイプライタ，またはCMD—8キャラクタディスプレイ装置（キーボード付）である．

2．システムの構成（図2）

本モニタシステムは，割込処理とタスク管理の2部分から構成されている．

割込処理部はさらに割込受付ルーチン，入出力機器管理ルーチン，コマンド受付ルーチンに細別される．

タスク管理部は3つの部分から構成される．第1の部分は実行ルーチンおよび待機ルーチンであり，実行ルーチンは，第2部分であるタスク実行部，または，第3部分であるローダ，ロールイン，ロールアウトルーチンなどにコントロールを渡す．また第2部分は，割込処理部からの指令によって起動される．

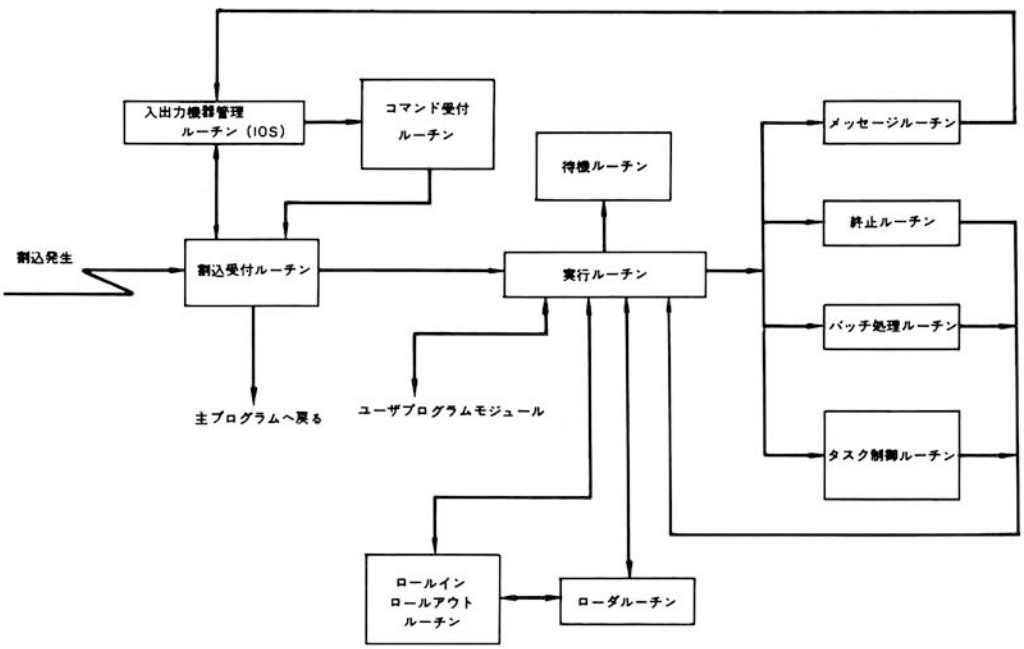


図2　ゼネラルフローチャート

2－1　割込受付ルーチン（図3－1、図3－2）

割込受付ルーチンは，CEC555の割込バスラインに接続されているすべての周辺機器の受付管理をおこなう．

1．クロック割込が発生した場合は，クロックを使用しているプログラムに対応するアドレスを探

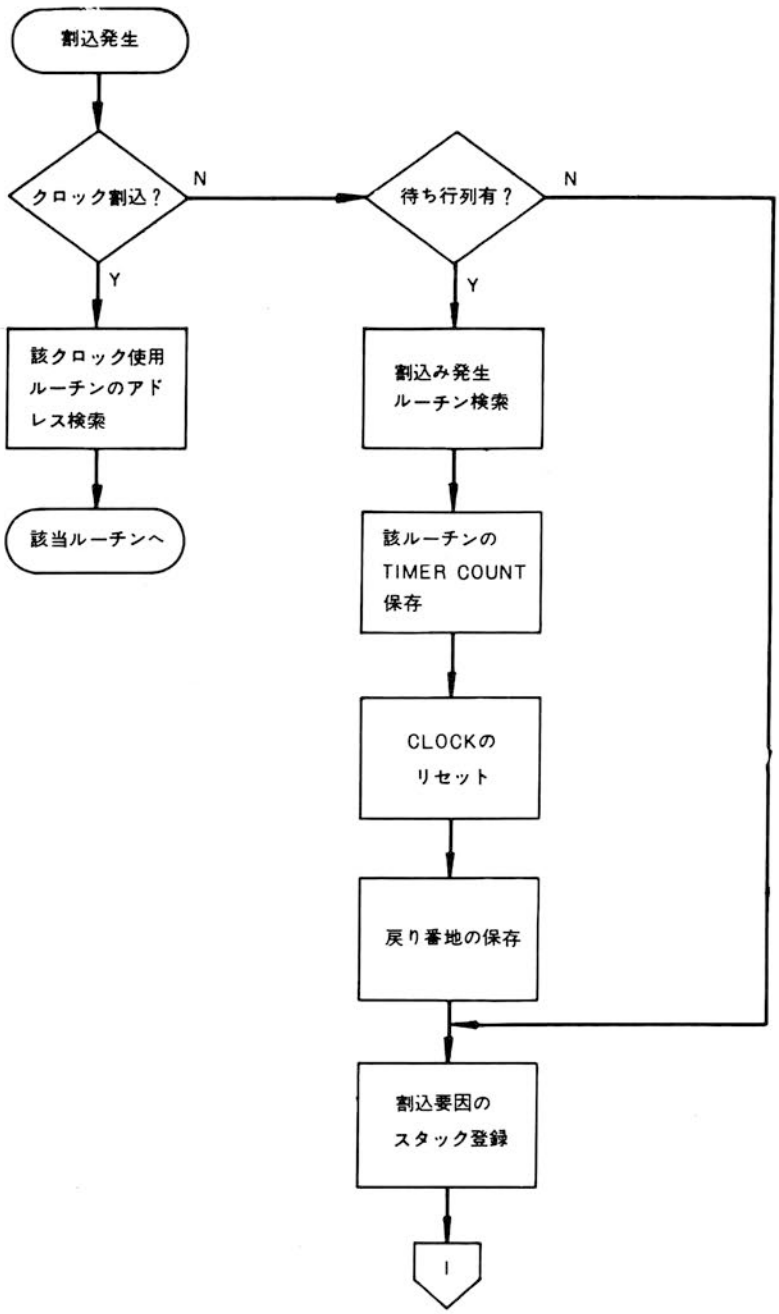


図3-1　割込受付ルーチン-1

し，これにコントロールを移す.

2．その他の周辺装置の動作終了割込については，割込待ち行列により処理をおこない，入出力機器管理ルーチン，または実行ルーチンにコントロールを移す.

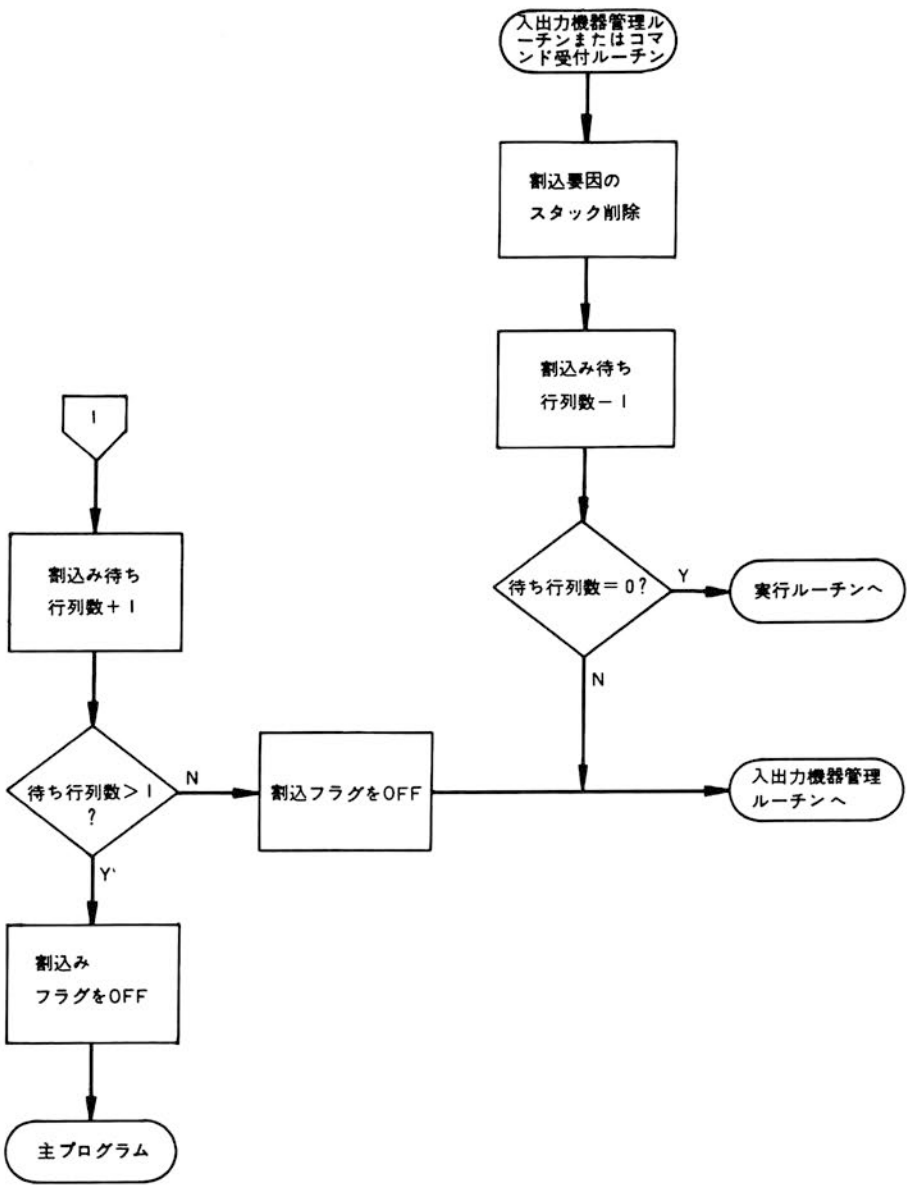


図3-2　割込受付ルーチン-2

2—2　入出力機器管理ルーチン（図4）

入出力機器管理ルーチンは，入力動作または出力動作の管理をおこなう．入力動作は，

1．データバッファから1字読みとってそのデバイスに対応する入力バッファに移す．バッファポインタをアップデートしてハードウェアフラグとコントロールテーブルのソフトウェアフラグとをクリアする．

コンソールタイプライタについてはＥＯＦならば，コマンド受付ルーチンへ，ＥＯＦでなければ割込受付ルーチンにコントロールを移す．コンソールタイプライタ以外の入力機器では，次の

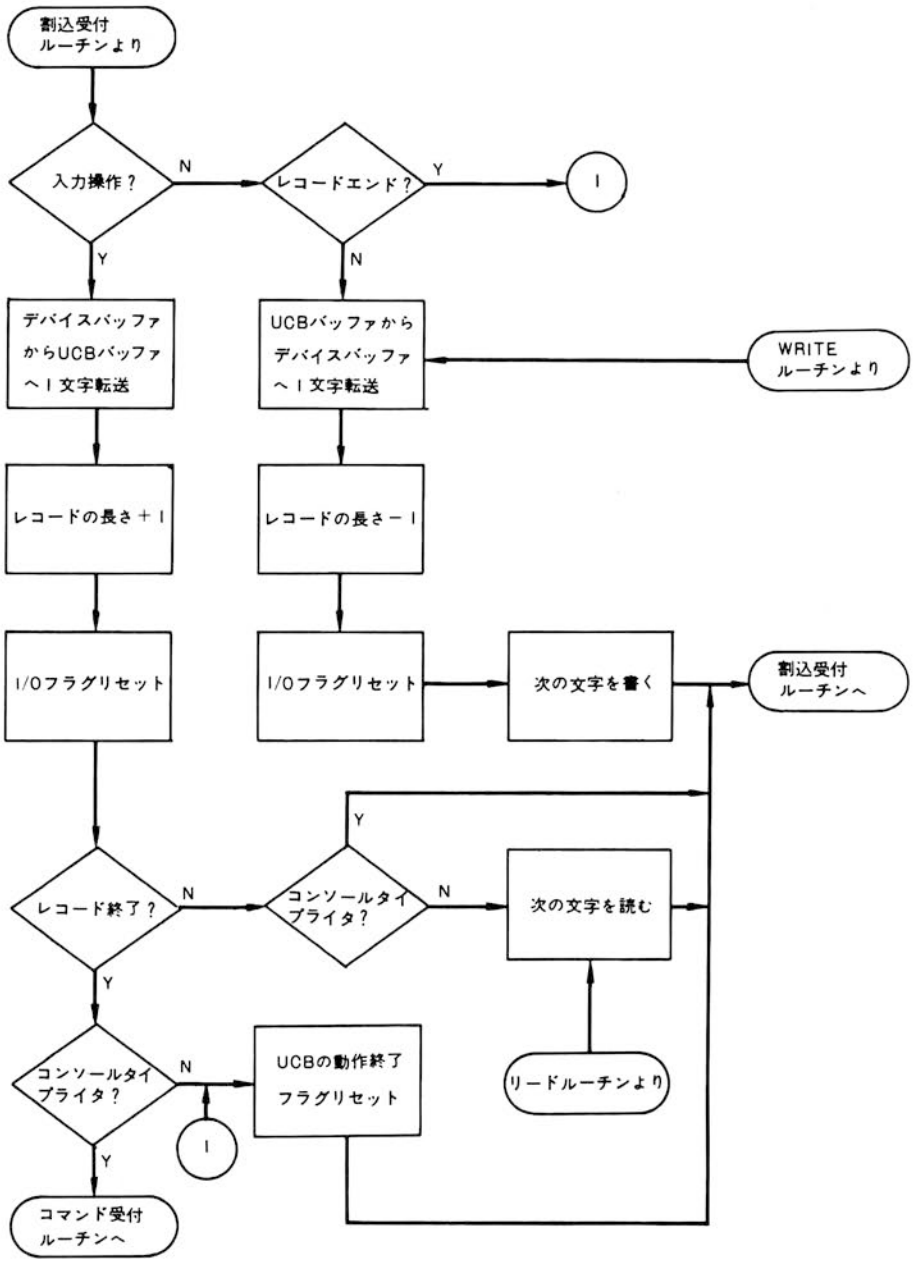


図4　入出力器機管理ルーチン

文字の入力命令を発行した後，割込受付ルーチンにコントロールを移す．

出力動作でＥＯＦならば，対応するコントロールテーブルのブロック転送フラグに記入して割込受付ルーチンにコントロールを移す．ＥＯＦでなければ，出力バッファから１字周辺機器に転送命令を発行するとともに，ポインタのアップデート，ハードウェアフラグ，コントロールテーブルのソフトウェアフラグをクリアして割込受付ルーチンにコントロールを移す．

入力（READ），または出力（WRITE）マクロ命令の処理は，その処理が完了した時にコントロールテーブル内のブロック転送フラグによってプログラムに完了を通知する．

２—３　コマンド受付ルーチン（図５，図６—１，図６—２）

コマンド受付ルーチンは，コンソールタイプライタからのコマンドを受けるために用意され，各コンソールごとに USER IDENTIFICATION ELEMENT（UIE）を作成する．

UIEは，コンソール番号，ステータス，ユーザのパスワード，優先度からなる．ステータスは次の10個のステータスが可能である．

０：使用中でない
１：コマンド要求受付中
２：ユーザ確認済
３：コマンド要求の実行待ち
４：コマンド待ち

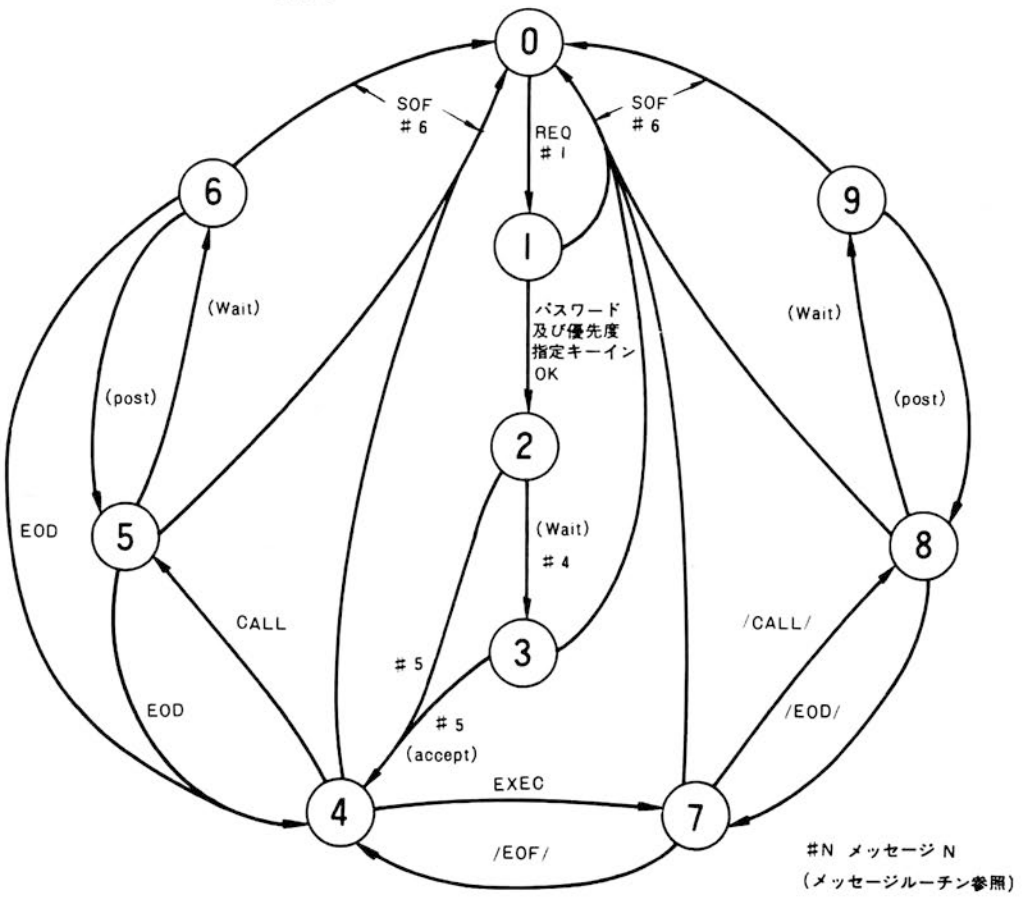


図５　UIEステータスせん移図

5：コンソール会話中
6：コンソール会話待ち
7：バッチコマンド受付ルーチン動作中
8：バッチコマンド受付中
9：バッチコマンド実行待ち

UIEステータスが0で受付可能のコマンドは，REQコマンドのみである．UIEステータスが1では，コンソールキーインを受付け，ユーザ確認が終れば，UIEステータスは2に変る．UIE 2ではコマンド受付可能でなく，優先度管理ルーチンにより3に進められた後，実行ルーチンにより4に変化されてはじめてコマンド受付可能の状態となる．

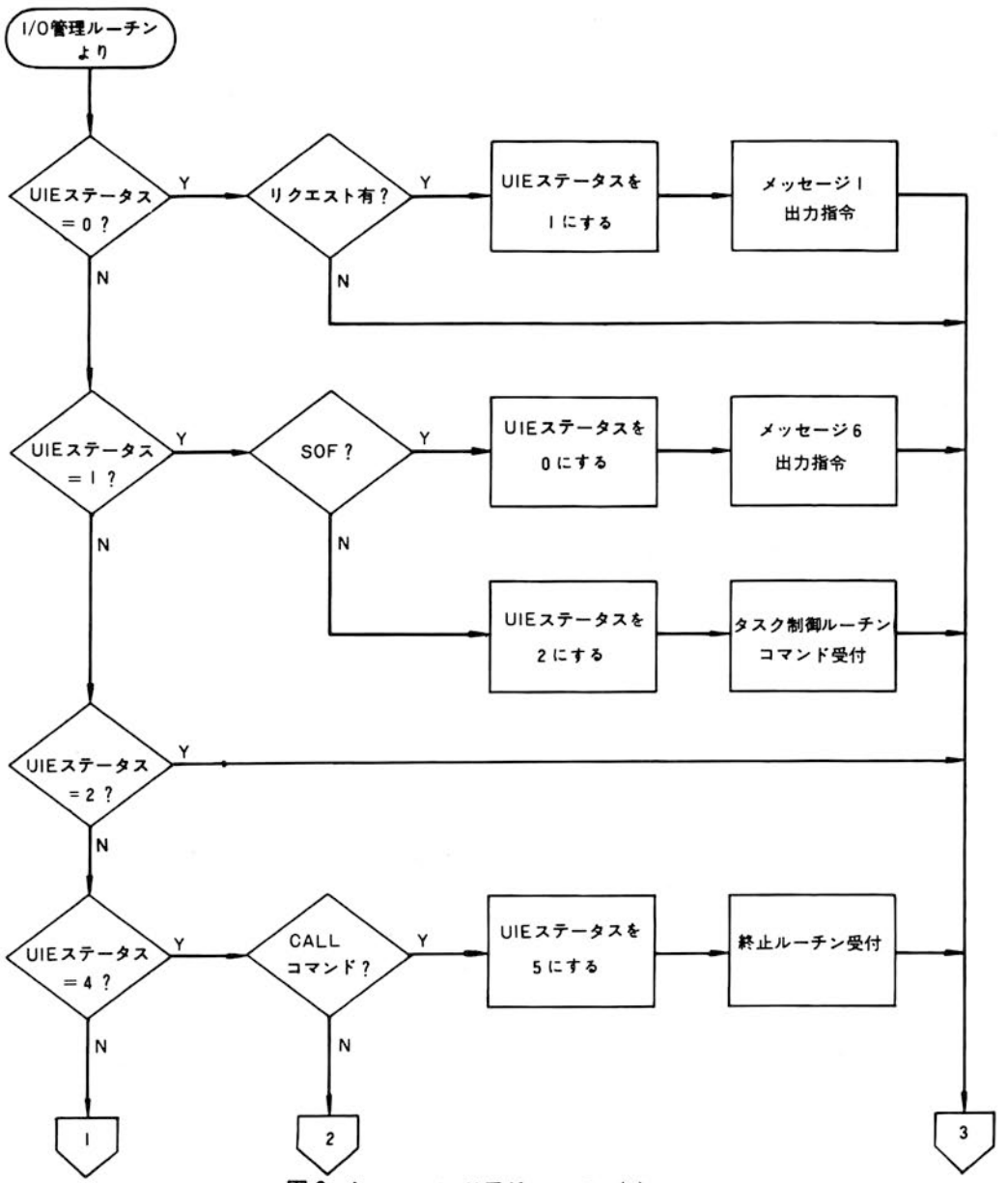


図6-1　コマンド受付ルーチン(1)

UIE４でCALLコマンドを受付けると，UIEは５となりEXECコマンドを受付けるとＵＩＥは７となる．UIEが５または６でデータエンドとなると，UIEは４にもどる．ＵＩＥが８または９でデータエンドとなるとUIEは７にもどる．UIEが７でＥＯＦになるとＵＩＥは４にもどる．

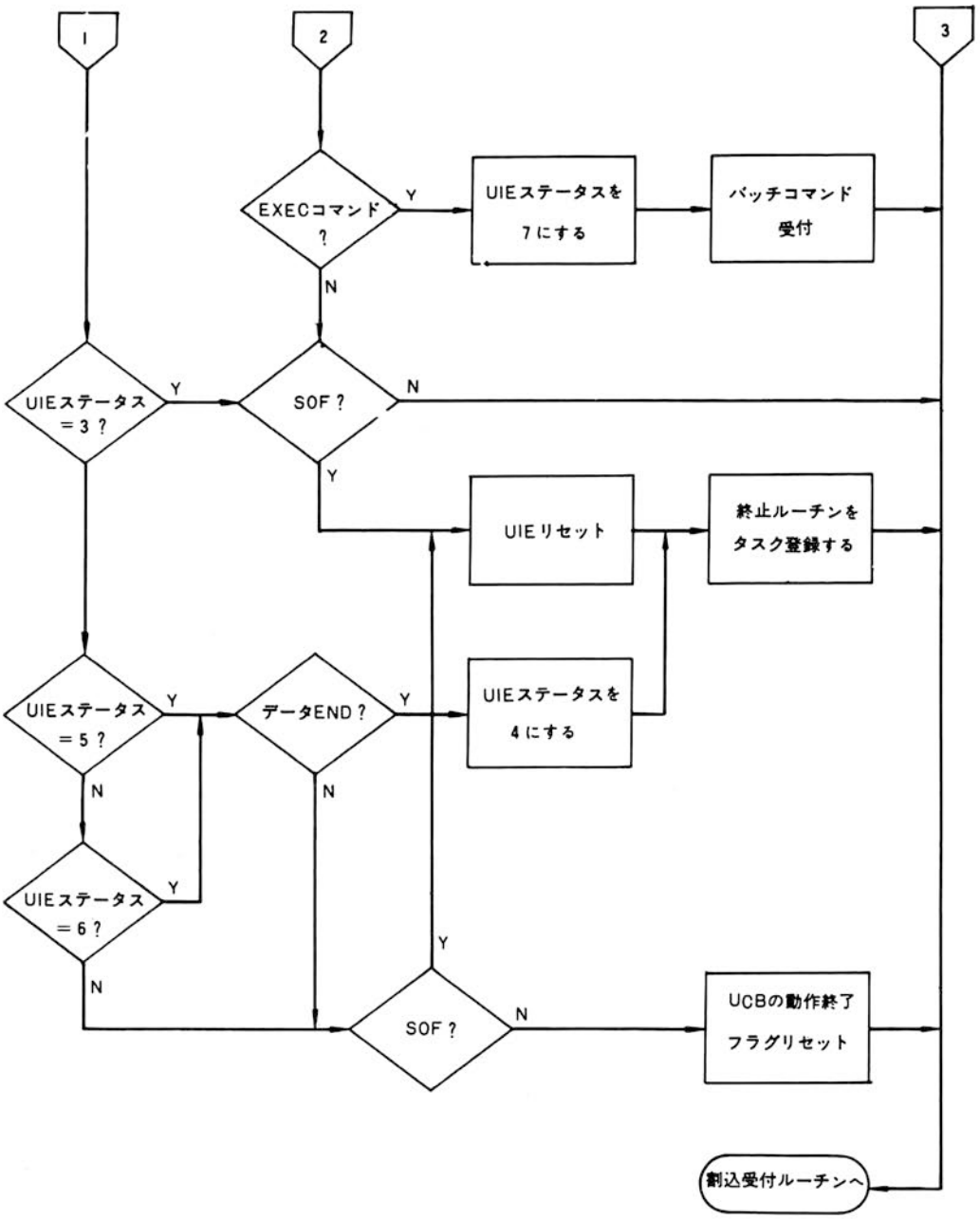


図６-２　コマンド受付ルーチン(2)

2—4　機器コントロールテーブル（UCB）

機器コントロールテーブルは，各入出力機器ごとに，対応した情報を蓄えるためのもので以下の要素から構成されている．

1．デバイスの内部番号
2．インタラプトフラグ（ハードウェアの割込によってセットされる）
3．動作終了フラグ（1つのレコードの転送が完了した時にセットされる）
4．割当てられたUIEアドレス
5．入出力バッファポインタ
6．入出力バッファサイズ
7．転送ワードカウント

2—5　UIE（USER IDENTIFICATION ELEMENT）

UIEはシステムに含まれる各コンソールごとに用意され，各UIEはコンソール番号，ステータス，ユーザのパスワード，優先度からなる．

2—6　実行ルーチン（図7—1，図7—2）

実行ルーチンは，タスク管理をおこなうためのもので，ある時刻に，どのモジュールにコントロールを与えるべきかを決定する．

モニタ機能は，一番優先度の高いレベルが与えられており，実行ルーチンはまずモニタ機能のなかで，待ち状態にあるものがないかをチェックする．モニタ機能の各モジュールは，動作すると，タスクテーブルを作る．

タスクテーブルはタスクの内容を記述するものであり，モジュールの内部番号，入口アドレス，管理データ，UIE，戻りアドレスなどからなる．

複数個のタスクの並行処理では（EXECUTION CONTROL TABLE（ECT））によってタスク間のリンケージが保たれる．割込みが発生した時は優先度によって直ちにコントロールを移すべきか，待ち行列に加えるべきかを決定し，必要ならばROLL-OUT，ROLL-INをおこなってコントロールを移す．ひとつの処理が終ったら待ち行列をチェックし，未サービスのものがあればサービスをおこなって待ち行列から除く．

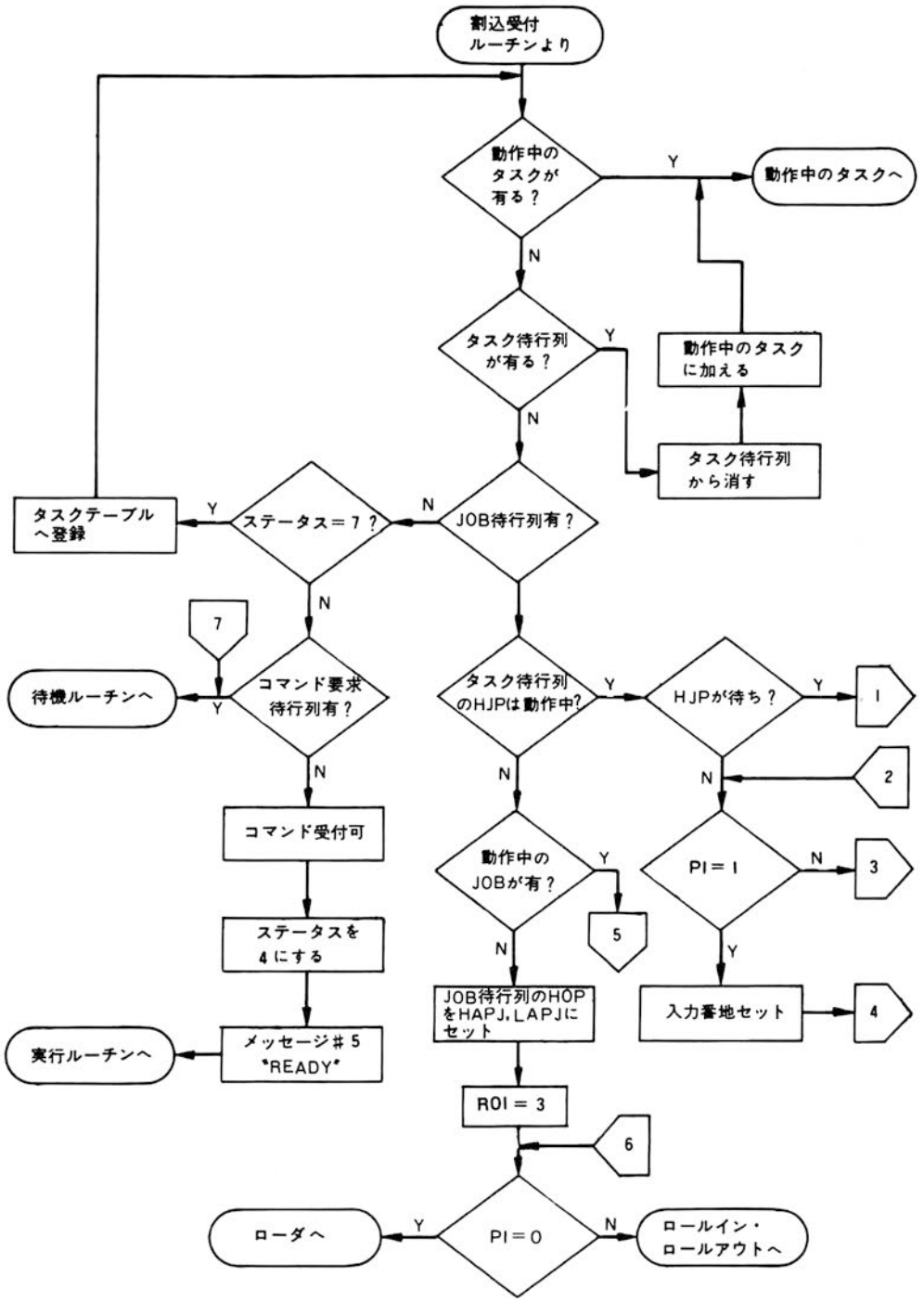

割込受付
ルーチンより
動作中の
タスクが
有る？
Y
動作中のタスクへ
N
タスク待行列
が有る？
Y
動作中のタスク
に加える
N
タスク待行列
から消す
タスクテーブル
へ登録
Y
ステータス＝7？
N
JOB待行列有？
N
7
待機ルーチンへ
Y
コマンド要求
待行列有？
タスク待行列
のHJPは動作中?
Y
HJPが待ち？
Y
1
N
2
N
コマンド受付可
動作中の
JOBが有？
Y
5
PI＝1
N
3
ステータスを
4にする
N
Y
JOB待行列のHOP
をHAPJ,LAPJに
セット
入力番地セット
4
実行ルーチンへ
メッセージ#5
"READY"
ROI＝3
6
ローダへ
Y
PI＝0
N
ロールイン・
ロールアウトへ

図 7 - 1　実行ルーチン

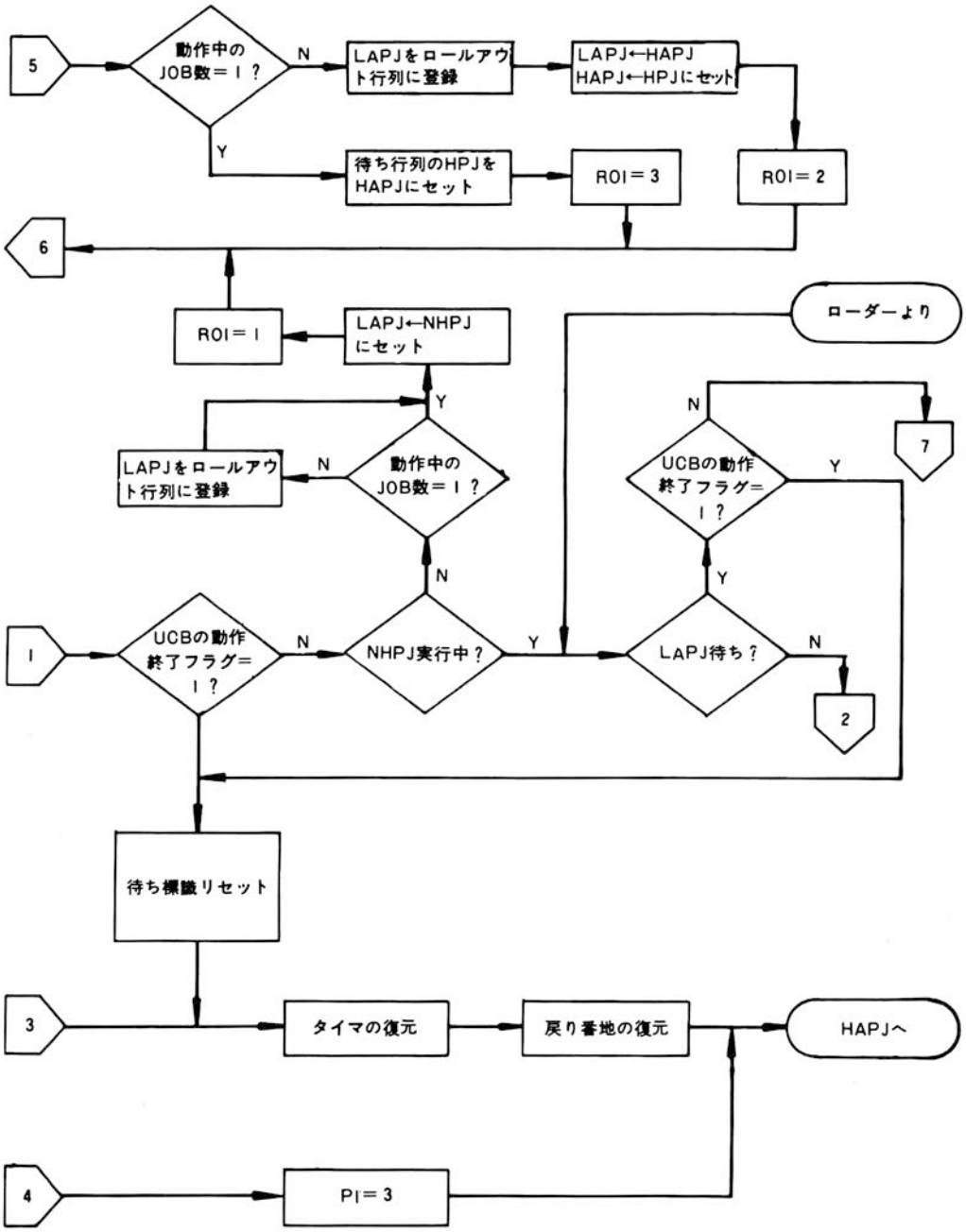

5
動作中の JOB数＝１？
N
LAPJをロールアウト行列に登録
LAPJ←HAPJ HAPJ←HPJにセット
Y
待ち行列のHPJをHAPJにセット
R0I＝3
R0I＝2
6
R0I＝1
LAPJ←NHPJにセット
ローダーより
Y
N
7
LAPJをロールアウト行列に登録
N
動作中の JOB数＝１？
UCBの動作終了フラグ＝１？
Y
N
Y
1
UCBの動作終了フラグ＝１？
N
NHPJ実行中？
Y
LAPJ待ち？
N
2
待ち標識リセット
3
タイマの復元
戻り番地の復元
HAPJへ
4
PI＝3

図７-２　実行ルーチン

2—7　ローダ（図8）

ローダはダイレクトリによって指定されたプログラムモジュールを，コアの空きスペースにロードする機能をもつ．プログラムモジュールは，ページリロケーダブルフォーマットをもち，ローダによって絶対アドレスを与えられる．コアの空きスペースが充分でない時は，それよりも優先度の低いプログラムをROLL-OUTすることによってスペースを確保する．コアのすべてのスペースはページ単位の割当てアップによって管理する．

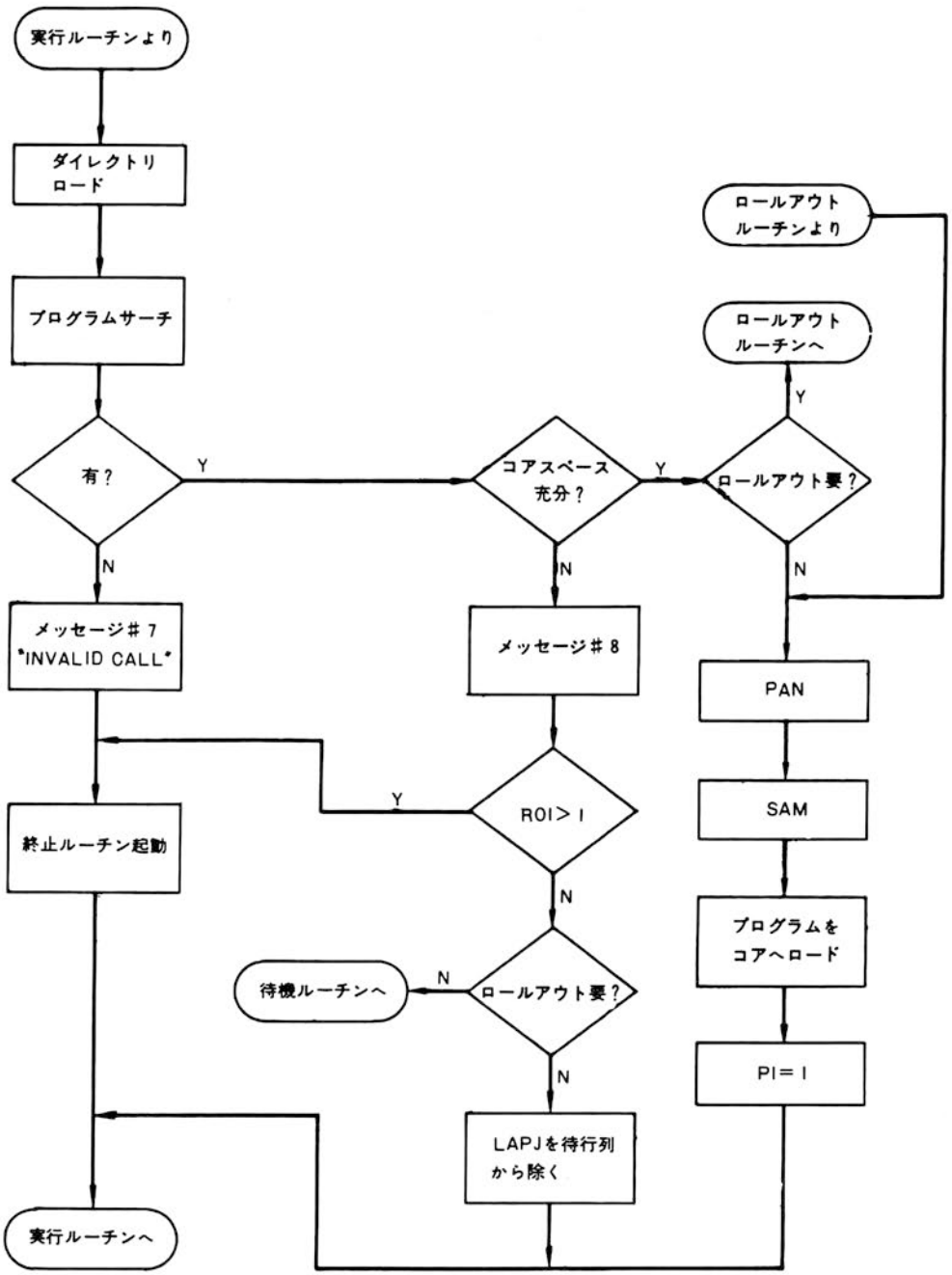


図8　ローダ

2—8　ROLL-INルーチン（図9）

実行ルーチンからROLL-INルーチンにコントロールが移されると，空きスペースが充分であればオブジェクトモジュールをロードして実行ルーチンにコントロールをもどす．空きスペースが充分でなければ，待ちルーチンにコントロールを移す．

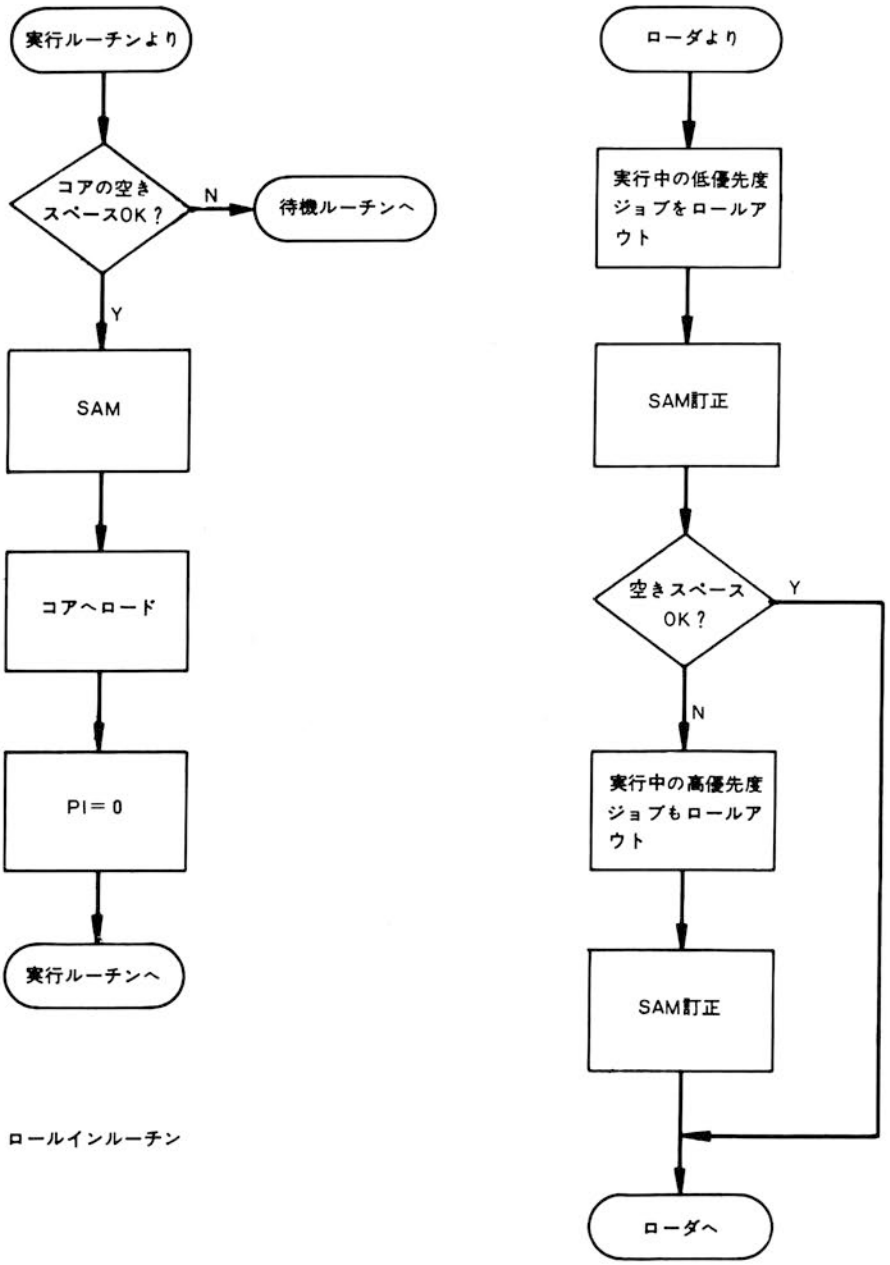


図9　ロールアウトルーチン

2－9　ROLL-OUTルーチン

ROLL-OUTルーチンはローダから呼ばれ，優先度の低いモジュールから順にコアから排除して必要なスペースを確保してローダにコントロールを戻す．

2－10　メッセージルーチン（図10）

メッセージルーチンは，コンソールに以下のいずれかのメッセージを印刷するために用いられ，1～9の番号で指定される．

1：LOG IN
2：UNAUTHORIZED USER, REQUEST REJECTED
3：UNAUTHORIZED PRIORITY, REQUEST REJECTED
4：WAIT, HIGHER PRIORITY JOB BEING PROCESSED
5：READY
6：SIGN OFF
7：INVALID CALL
8：UNAVAILABLE STORAGE
9：INVALID COMMAND

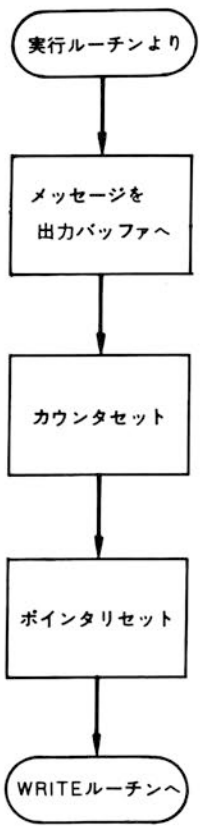


図10　メッセージルーチン

2—11　終止ルーチン（図11）

終止ルーチンは，実行ルーチンから呼ばれ，ジョブの待ち行列からの抹消，メモリマップのアップデートをおこなう．データエンドでUIEステータスの８の時はUIEステータスを７に変更する．ＳＯＦ（サインオブファイル）で呼ばれた時は，UIEを待ち行列から取り除く．終止ルーチンからの戻りはいずれの場合も実行ルーチンにコントロールが移される．

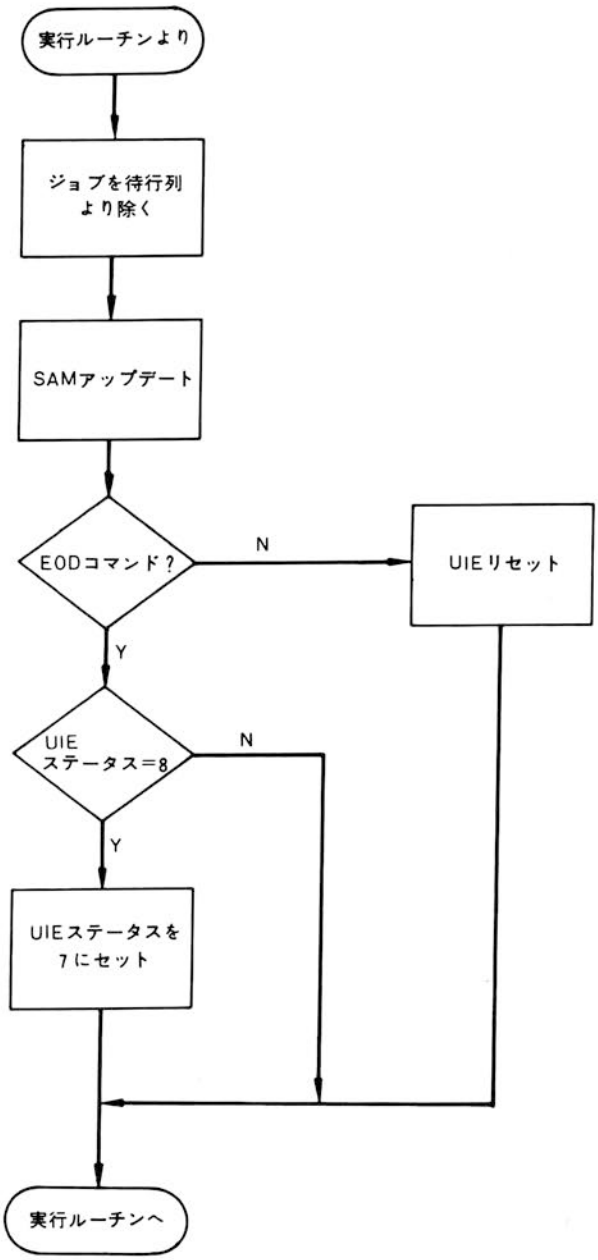


図11　終止ルーチン

2—12　バッチ処理ルーチン（図12）

バッチ処理ルーチンは実行ルーチンから呼ばれ，バッチ処理（コンソール以外の任意のファイルからのジョブ指定）を可能にする．UIEステータスが7でCALLコマンドであればUIEステータスを8にしてJOBを起動する．UIEステータスが8以外でEODコマンド以外の時はレコードをシステムスクラッチにストアする．

EODコマンドの時は実行ルーチンにコントロールを戻す．UIEステータスが7でEOFコマンドが与えられるとUIEステータスを4にして実行ルーチンに戻る．

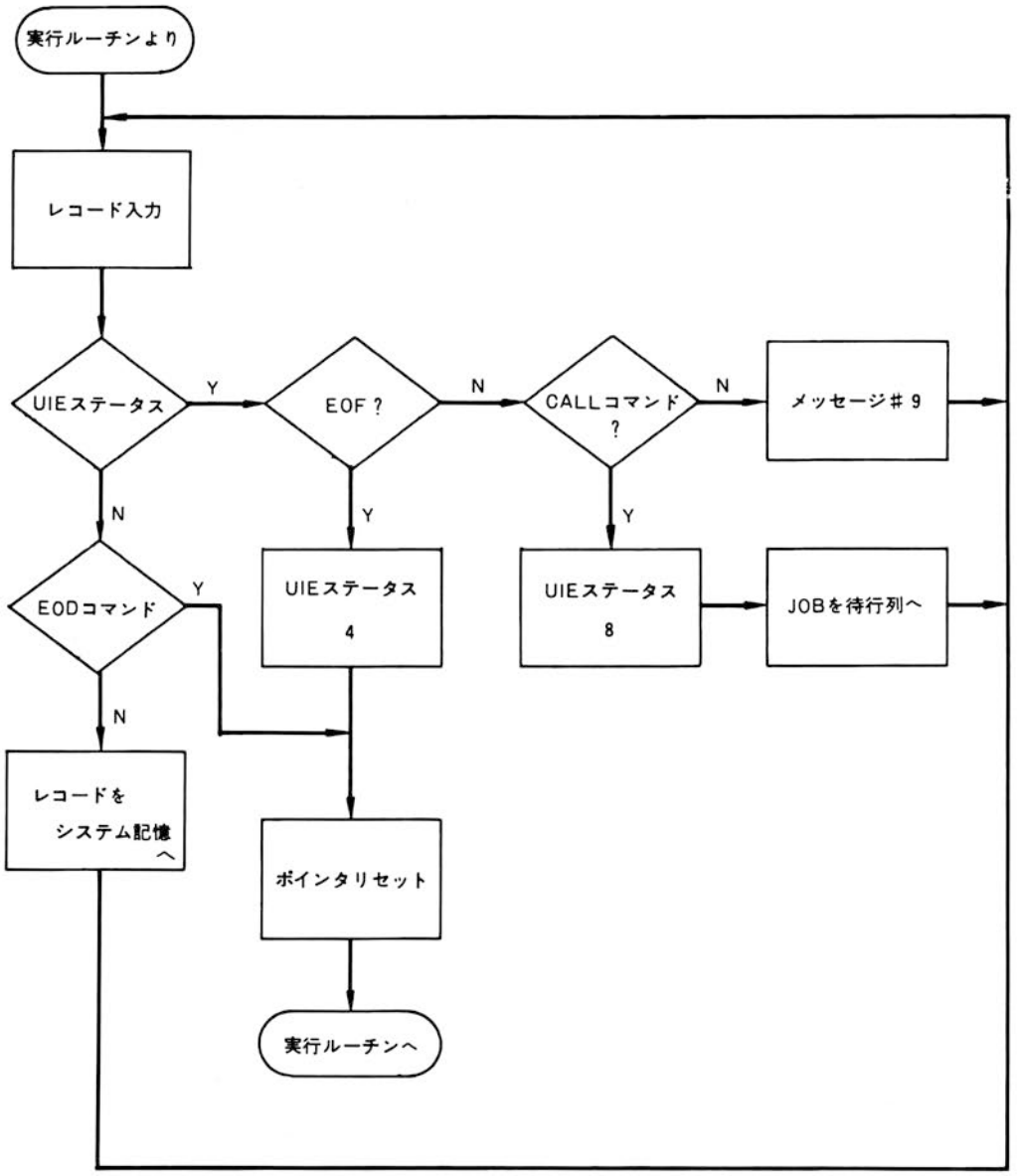


図12　バッチ処理ルーチン

2—13　タスク制御ルーチン（図13）

タスク制御ルーチンはユーザから与えられたパスワードおよび優先度のチェックをおこない，以下の必要な処理をおこなう．

パスワードおよび優先度チェックが適正で，指定されたジョブよりも優先度の高いジョブが実行中でなければ，UIEステータスを4にセットし，コントロールを実行ルーチンに戻す．

より優先度の高いジョブを実行中であればUIEステータスを3にセットし，リクエスト待ち行列

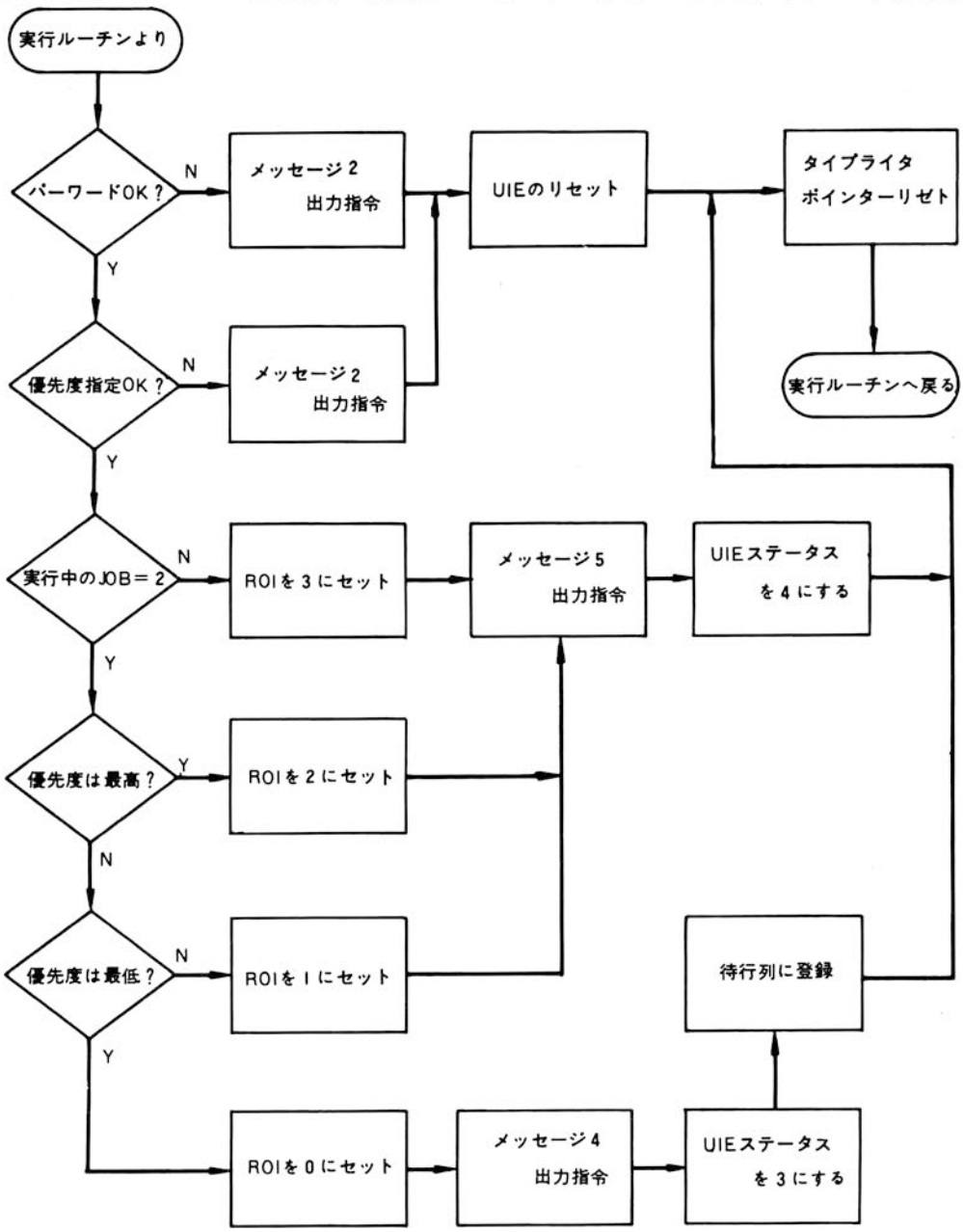


図13　タスク制御ルーチン

に登録して実行ルーチンに戻る．パスワードあるいは優先度指定が不適当であれば，エラーメッセージを印刷してUIEを抹消し，実行ルーチンに戻る．

付録

《略記号一らん》

DAC	D／Aコンバータ
DDT	システム記憶ダイレクトリ
DEC	－1する
DEQ	待ち行列から除く
EOD	レコードエンド
EOF	ファイルエンド
ENQ	待ち行列に加える
EXEC	実行コマンド
HAPJ	実行中の最高優先度ジョブ
HPJ	最高優先度ジョブ
IC	待ち行列カウント
INC	＋1する
IOS	入出力機器管理ルーチン
JT	ジョブトランザクション
JTQ	ジョブ待ち行列
LAPJ	実行中の最低優先度ジョブ
NHPJ	待ち行列の中の最優先度ジョブ
PAM	プログラム割当マップ
PI	動作標識
PPS	タスク制御ルーチン
PPT	タスク制御テーブル
RAR	コマンド受付ルーチン
ROI	ロールアウト標識
ROQ	ロールアウト行列
RQ	コマンド要求待ち行列
SAM	主記憶割当マップ
SOF	ジョブ終了
TT	タスクトランザクション
TTQ	タオク待ち行列
UCB	機器コントロールテーブル
UIE	USER IDENTIFICATION ELEMENT

応　用　編
（アプリケーションシステム）

応　用　編　目　次

① 工場自動群制御システム…… 702
② トータシステム…… 706
③ CEC EDPシステム…… 710
④ 医療情報システム…… 740
⑤ 繊維産業システム…… 763
⑥ 計測制御システム…… 769
⑦ 自動車エレクトロニクス…… 778
⑧ 文教システム…… 781

CEC555ミニコンピュータアプリケーションシステム

CECは創業以来15年間，一貫して情報処理システムの基礎技術であるパルス技術，電源技術，特殊情報処理技術の蓄積に努力を傾注し，パルス・電源機器，情報処理用ＩＣパッケージ，その他の周辺機器などの各部門について標準化を完了している．

CECの情報処理システムは，これらの基礎技術と多年にわたる豊富な径験のもとに緻密微細に設計・製作され，すでに各種の自動省力システムをはじめ，数々の電子計算機システムが各分野において活躍している．

特　長

用途に応じて適切な機種が選択できる

最新鋭 555H(並列演算，0.8μS 4～32K語)，マイクロコンピュータ555SS(8ビット並列8.25Kバイト～16Kバイト)，メンテナンスの容易な大型基板の555A（８K固定ICメモリ)，その他用途に応じておえらびいただける豊富な機種を用意している．

豊富なソフトウェアを用意

事務計算用から制御用まで抜群の威力を発揮するモニタシステムIIをはじめ使いやすいベーシックソフト，および豊富なアプリケーションソフトが準備されている．

事務計算にも強いソフトウェアとハードウェア

事務処理システム専用のCECEDPシステム10／60は事業規模と処理目的によっておえらびいただける20システムを標準化．

小型ながら中型機なみの性能

主記憶装置は必要に応じて32K語まで拡張でき，さらに磁気ディスク，磁気テープなどを接続することにより中型機なみの性能を発揮．

これから５～10年有効利用できる標準専用システム

技術革新や，将来新らしい周辺装置が出現しても必要充分な拡張，増設の機能が組込まれている．さらにアプリケーションソフトウェアの追加も容易である．

システムインターフェース用ICパッケージ類の標準化

C41，C61，サブシステム150シリーズなど約350種におよぶ情報処理用ICパッケージが標準化されており，CPU～周辺装置，システム～システムの接続が容易．

アナログデータの収集・解析・監視・制御が容易

A/Dコンバータ，マルチプレクサ，その他計測用オプションとして各種ICパッケージが標準化されている．

システムの拡張が自由自在にできる豊富な周辺装置群

必要に応じて各種の周辺機器が接続できるので機動力と経済性は抜群．

公衆通信回線，専用回線を利用した広域オンラインシステムが形成できる（参考図参照)．

NC用コントローラ，プログラムシーケンサに

小型，軽量で高い信頼性をもち，しかも価格が低廉なので各種自動省力システムの標準ユニットとして手頃に利用できる．

豊富な基礎技術の集積

16年間にわたる電源技術，パルス技術，情報処理技術の集積が新しいシステム創造に貢献．

完全なアフターサービスの保証

担当プロジェクトチーム，SEグループの充分なソフトウェアサポートとサービス，およびシステム障害発生時の緊急連絡網など保守体制が完備している．

CEC555の利用分野

1．高校・大学・学校・研究所・機械・重工業・自動車・車輌・材料関係

コンピュータ教育
コンピュータ研究
実験データ処理
データロガ
ロボット制御
ＮＣ装置
原子炉測定データ処理
Ｘ線データ処理
水理研究
振動分析
車輌運動解析
人間工学，ＭＥ
地震測定
統計計算
自動試験
試験データ解析
数学，物理計算
データ通信
座標計算
熱伝導，熱力学計算
自動運航・荷役制御
材料力学計算
歯車，ゼンマイ計算
図形認識
応力計算
分子運動計算
金属分析
流体計算
エンジンデータ処理
排気ガス分析制御
熱交換計算
空調計算制御
倉庫自動制御
プラント自動制御
特性計算分析
弾道計算
構造力学計算
光学計算
照明制御
土木計算
建築計算
自動作図

2．原料・食品・繊維・化学・石油化学・鉱業関係

気体分析測定計算
液体分析測定計算
カラー図柄分析
織機自動制御
化学分析
化学式計算
結晶分析
パルス波高分析
計算制御

3．電子・電気・データ通信関係

データ通信端末
計測，数値分析処理
設計計算
回路特性計算
半導体　Ｉ／Ｃ自動試験
波形，音声測定分析合成

4．経営・金融関係

経営分析計算
経営計画計算
料率，利息計算
見積計算
原価計算
在庫管理
販売業務

5．保健関係

心電図判読
脳波解析
細胞診
血液分析
病院管理・情報検索

6．環境，サービス，レジャー関係

公害データ処理
販売機群管理
天候，気象データ処理
案内サービス
交通制御
保安管理

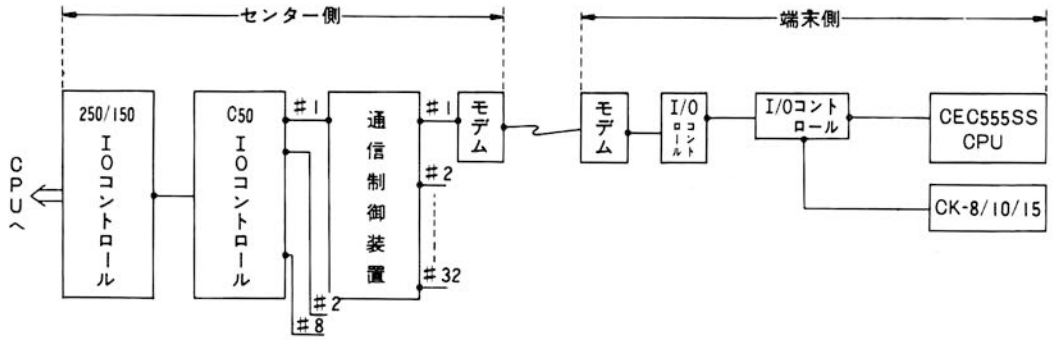
データ通信システム

標準品CK-M２カプラ　200ボー通常電話回線利用

CK-M12モデム　1200ボー
CK-M24モデム　2400ボー　公衆通信回線
CK-M48モデム　4800ボー　専用回線利用
CK-M96モデム　9600ボー

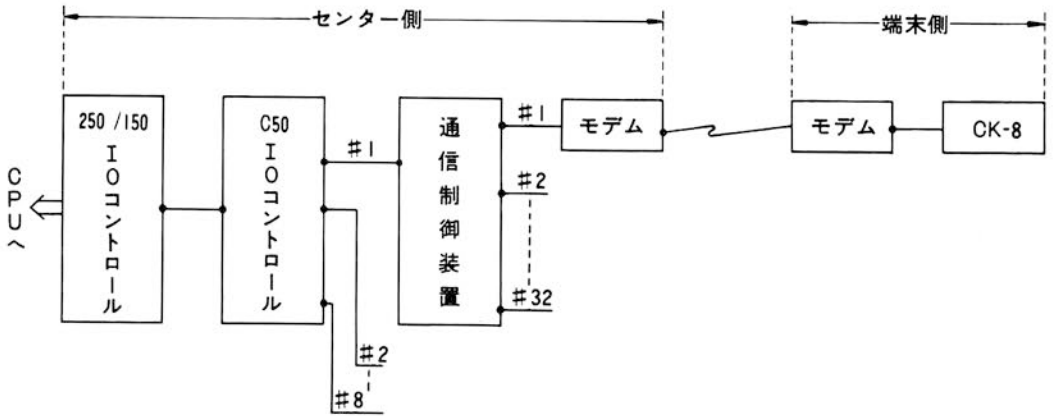
＊公衆通信回線，専用回線用コントローラも用意されている（CEC555SSマイクロコンピュータ）.

専用回線を設ける場合 A形



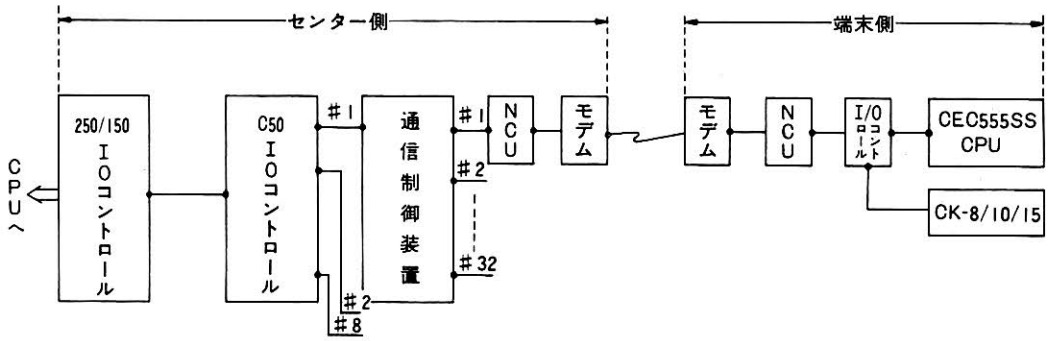
① 通信制御装置は基本4回線で4回線単位で増設可能.

専用回線を設ける場合 B形



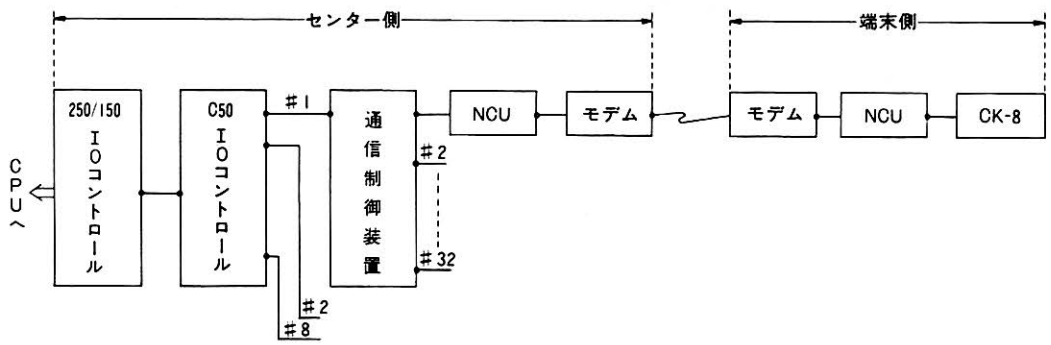
① 通信制御装置は基本4回線で4回線単位で増設可能.

電話回線と共用する場合A形



① 通信制御装置は基本4回線で4回線単位で増設可能.
② NCUは簡易型, MM型, AM型, AA型の4種類ある.

電話回線と共用する場合B形



① 通信制御装置は基本4回線で4回線単位で増設可能.
② NCUは簡易型, MM型, AM型, AA型の4種類ある.

❶ 工場自動群制御システム

《工場群制御システム》

本システムはCEC555-SSマイクロコンピュータと，CS-80キーボード入力装置（プラズマディスプレイ）を利用した工場群制御のモデルシステムである．

システムは，センタシステムから各種の指令(作業指令，生産工程指示など)によりCEC555-SSマイクロコンピュータのコントロール下で，複数台接続された端末装置を稼動し，それぞれの処理目的によって各作業現場への作業指示と各工程の最適運転管理が自動的におこなえるシステムである．

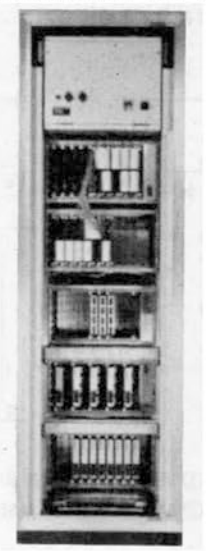
端末装置（CS-80キーボード入力装置）には，作業内容，各工程毎に所定項目をキーボード上に印刷し，（パネルシートを作成して1台の端末を多目的に利用する方式も用意されている）情報入力の簡便化と能率の向上をはかり，自動化による労力軽減をはかりながら，より高度な情報処理作業を実現することができる．

標準システムでは1台のマイクロコンピュータにより最大20台の端末装置を接続することができ，業務の内容，規模により増設が可能である．

またシステム内に異常検出回路機構が設けられており，障害発生時にただちに警報が発せられ適切な処置がおこなえ，24時間操業の各種工場現場の自動化に最適な自動化システムである．

《主な適用分野》

化学工場
石油精製工場
製鉄所
製粉工場
自動車工場
製紙工場
繊維産業
食料品生産
その他あらゆる生産ラインの自動化に適用



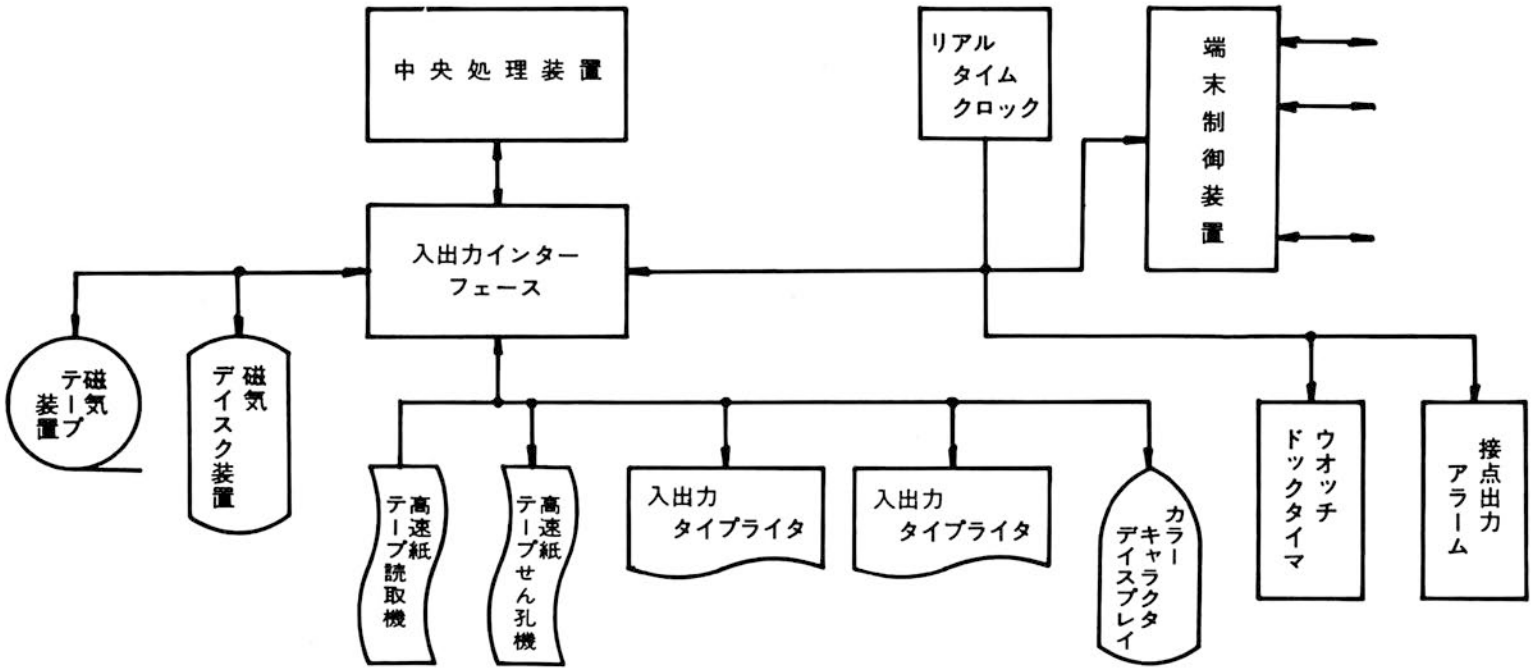
コントロールユニット
(555SS-IIマイクロコンピュータ)



出力プリンタ



キーボード入力端末


中央処理装置
リアルタイムクロック
端末制御装置
入出力インターフェース
磁気テープ装置
磁気ディスク装置
高速紙テープ読取機
高速紙テープせん孔機
入出力タイプライタ
入出力タイプライタ
カラーキャラクタディスプレイ
ウオッチドックタイマ
接点出力アラーム

《倉庫自動管理システム》

DCCシステムにおけるCEC555ミニコンピュータの利用

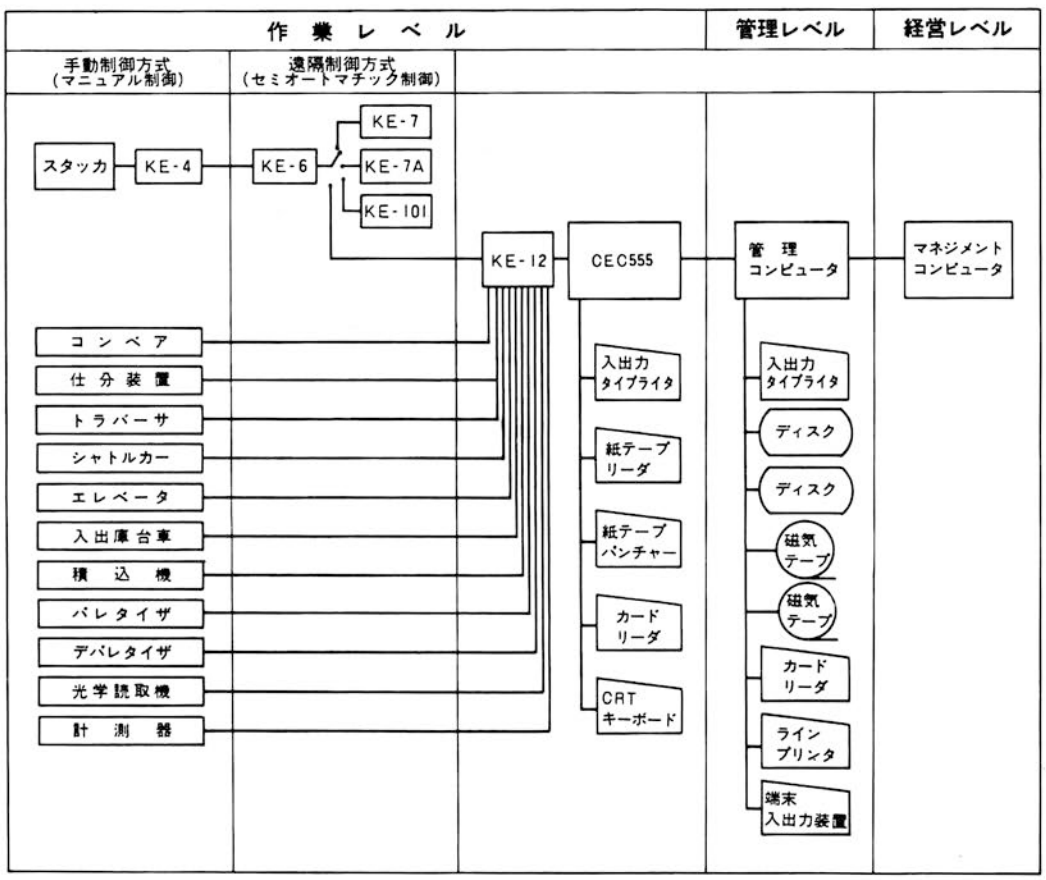
概　要

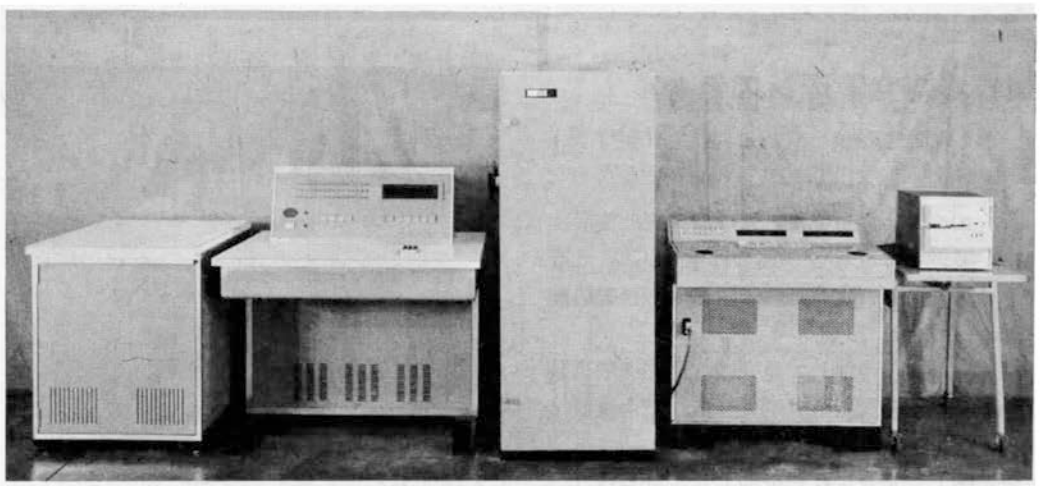
DCCシステムは，CEC555 ミニコンピュータと管理センターをベースに，自動倉庫・マテリアルハンドリングの稼動における効率性と経済性を追求する新しい倉庫自動制御システムである．システムは，次の3レベルを主要構成要素としている．

① 作業レベル：CEC555ミニコンピュータにより，スタッカの最適な運転の指示，および監視を行ない，コンベヤのトラッキング，仕分装置，情報表示装置をリアルタイムで制御する．

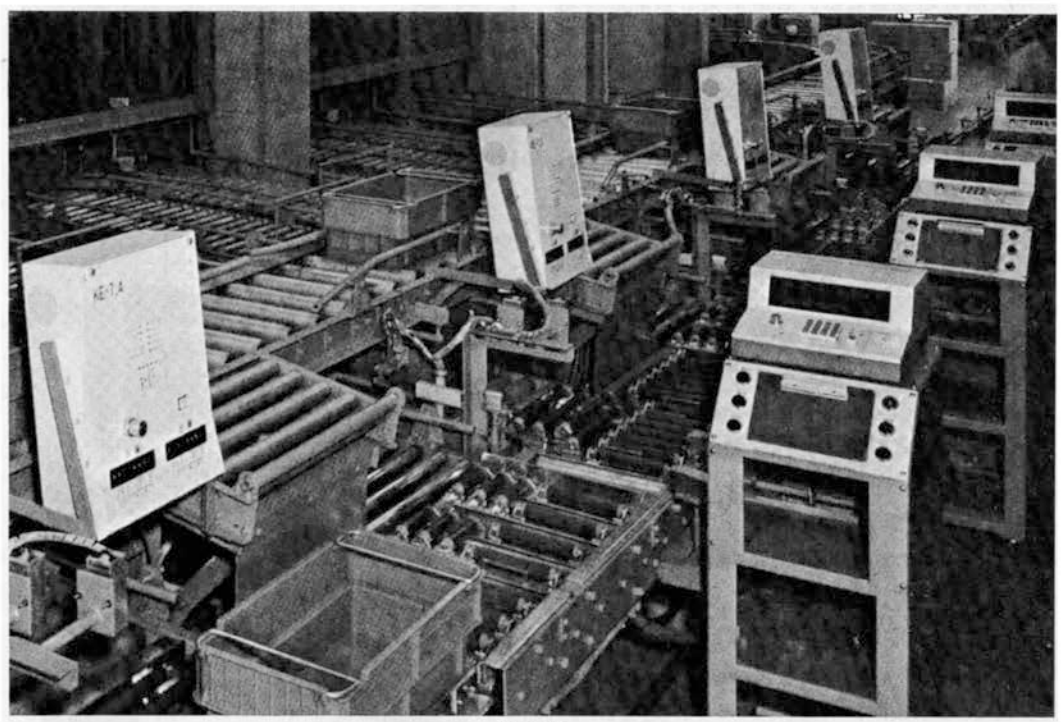
② 管理レベル：各作業レベルの作業事項のリアルタイム制御，作業内容の割当，発生データの収集・組織化がおこなわれ，さらに作業レベルとの対話により，必要な場合，作業レベルに対してバックアップをおこなう，また，経営レベルに正確なデータを供給し，倉庫・生産・出荷工程に関する最新のデータによる経営の分析，評価をおこなうことができる．

③ 経営レベル：経営・財務に関するデータについて通信回線によるオンライン・リアルタイム処理(あるいは磁気テープ，紙テープ等によるオフライン処理)で倉庫や工場内に設置されている管理コンピュータとの情報交換．





8 t スタッカクレーン制御装置



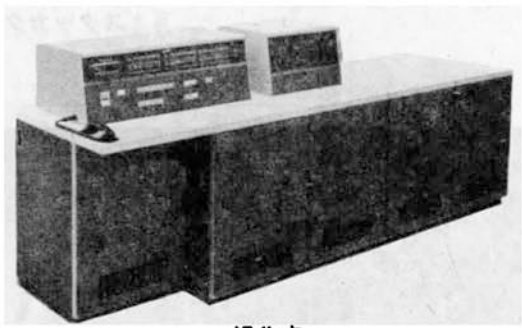
操作盤

❷ トータシステム

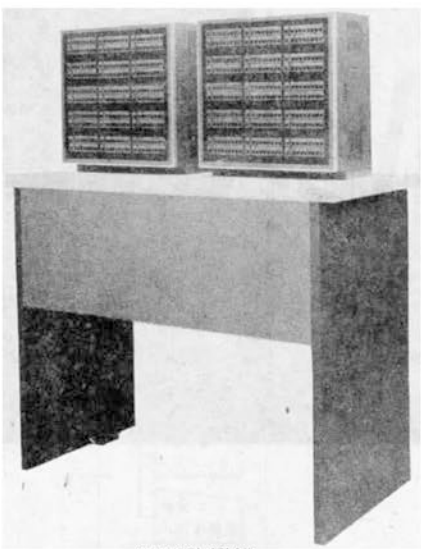
《競技場総合管理システム》



中央処理装置



操作卓



運用監視盤

概　要

本システムは発売業務における発売，集計，計算，帳表の作成など一連の作業を CEC 555 電子計算機を中央演算処理装置として自動処理を計るシステムである．

システムは，大別して発売機，発売機制御装置，群別集計装置，中央処理装置，指令装置，表示装置などにより構成される．

発売機より発売された金額，または券数は，オンラインで即刻群別集計装置に集計され，同時に中央演算処理装置に送られる．

中央演算処理装置は，各群別集計装置から送られたデータをもとに総合集計をおこない，発売終了と同時に集計結果をタイプライタにより自動印字，作表をおこなう．

中央処理装置では，総合集計のほかに，発送途中の集計，計算をおこなう機能をも備え，必要ある場合には中間の集計結果の表示，作表をすることができる．

本システムは，システム運用の特殊性について特に考慮をはらい，いかなる事態においても発売業務に支障を生じないよう設計され，システム各部についても中央部は二重構成，端末部は細分化の方式を採用し，危険の分散を計っている．また，データの取扱いは二重記憶方式により，その信頼性を高めてあり，万一システムの一部に不測の事故，障害発生がおこっても，システム全体に波及することのないようサブシステムの独立性に重点をおいて設計されている．

システム構成

1．中央処理装置

発売機などを集中管理，制御する中央処理装置は，　CEC555H　を 2 台並列で使用する．発売機からのデータは，主入出力接続装置を通しDSA (DIRECT STORAGE ACCE

SS) チャネルから入力される．DSAチャネルの処理能力は端末機器1,000台程度を有するので，相当大形のシステムでも運用が可能である．

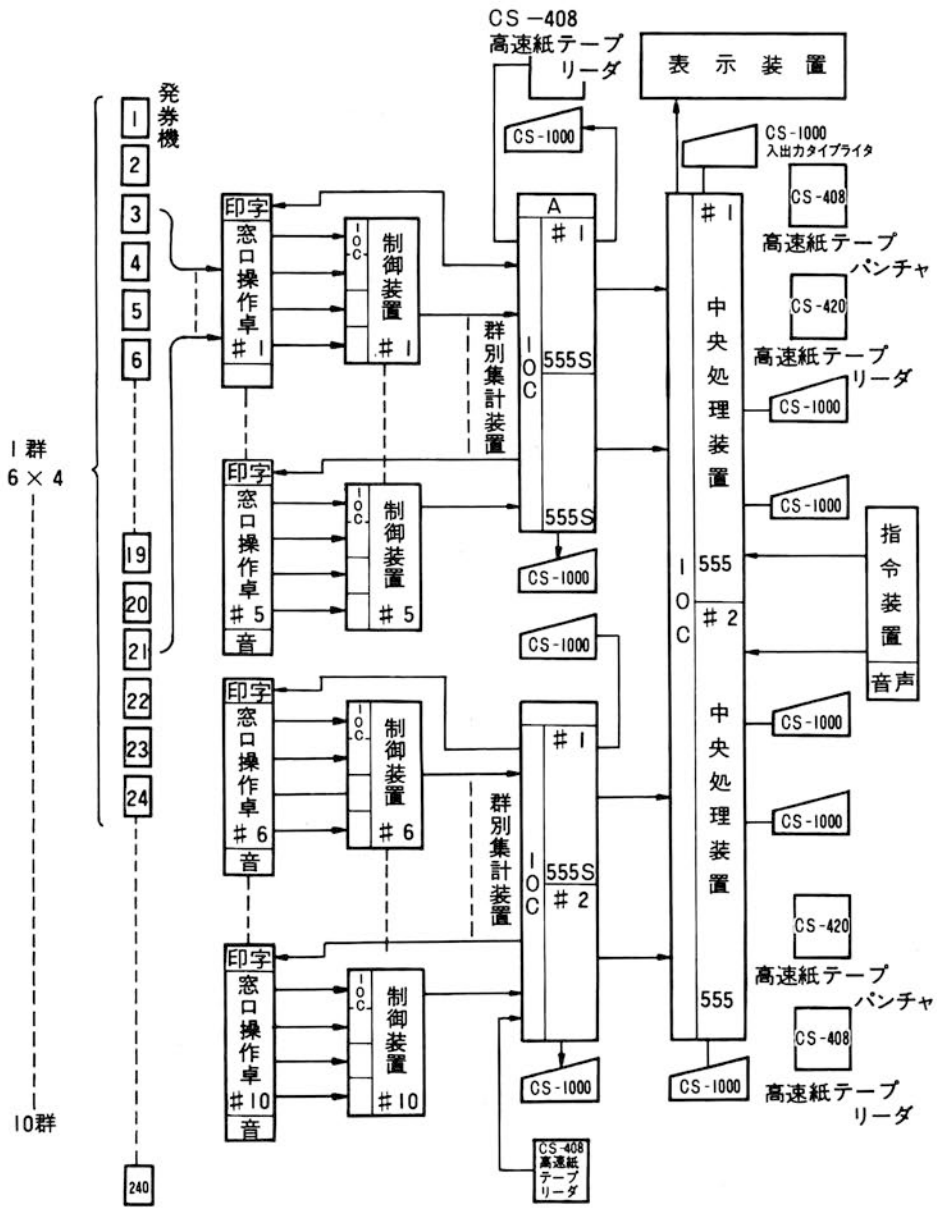
2．中央操作卓

システム全体の運営をおこなう．機能としては，中央処理装置への設定データの入力，プログラム変更の指令，売場の操作卓との連絡，およびシステム全体の動作状況の監視などがあげられる．

通常，中央処理装置の管理プログラムは，中央操作卓上の進行状態を示す押ボタンスイッチの監視，およびその時点で処理すべきプログラム実行の機能を有する．

3．表示装置

主入出力接続装置に接続され，中央処理装置で集計，計算されたデータを表示する．データは固定小数点10進3桁で1～2分ごとに



更新表示が可能である.

4．コンソールタイプライタ

コンソールタイプライタはシステム全体の保守，サービス用としてのみでなく，中央処理装置に事故が発生した場合のバックアップ用としても使用される.

5．紙テープリーダ

中央処理装置へのプログラムローディング，およびデータの読込み用として使用する.

6．タイプライタ

タイプライタは，オンラインで出力専用として使用され，所定の各種帳表の印刷，作表をおこなう．タイプライタは中央処理装置1台につき2台接続され，作業時間の短縮を計っている.

7．紙テープパンチャ

1日分のデータは，中央処理装置の主記憶装置で記憶し，終了時にタイプライタで作表されるが，万一の事故に備えて個々の紙テープとして出力しておき，1日終了と同時にバッチ処理できるようにされている.

8．群別集計機

群別集計機は，最大5台までの発売機制御装置を接続し，中央処理装置への中継をおこなう.

主な機能は，中央処理装置へ送るデータのマルチプロセッシング，パリティ検査，部分集計用計算機（CEC 555S）による部分集計，作表，すなわち発売機別集計，操作卓別集計，種類別集計をおこない，その結果を印字作表するとともに，紙テープにせん孔することもできる.

9．制御装置

発売機からのスタート信号を受け，プリント指令，発売機へのアラーム信号の伝達，中央装置への分類，集計の要求，および発売機の動作不良に対する各種の検査機能などを有する.

制御装置はシステムの性質上，端末装置の異常，故障などが中央の各装置へ影響しないよう電気的に分離する機構として設計され，制御する端末（発売機）装置は5～6台程度に分割制御する方式をとっている.

10．窓口操作卓

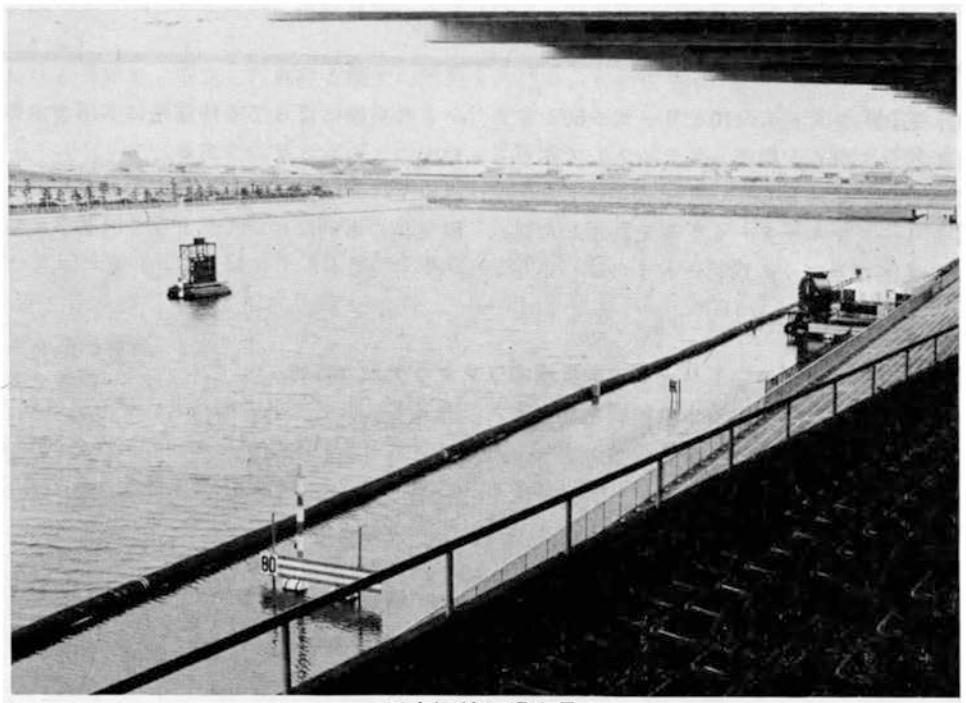
発売機最高24台を1群として1帖場を作り，各帖場単位に1台ずつ設置する．機能としては各発売機の状態監視と制御のほか，誤発売などの減算処理などがあげられる.

11．発売窓口

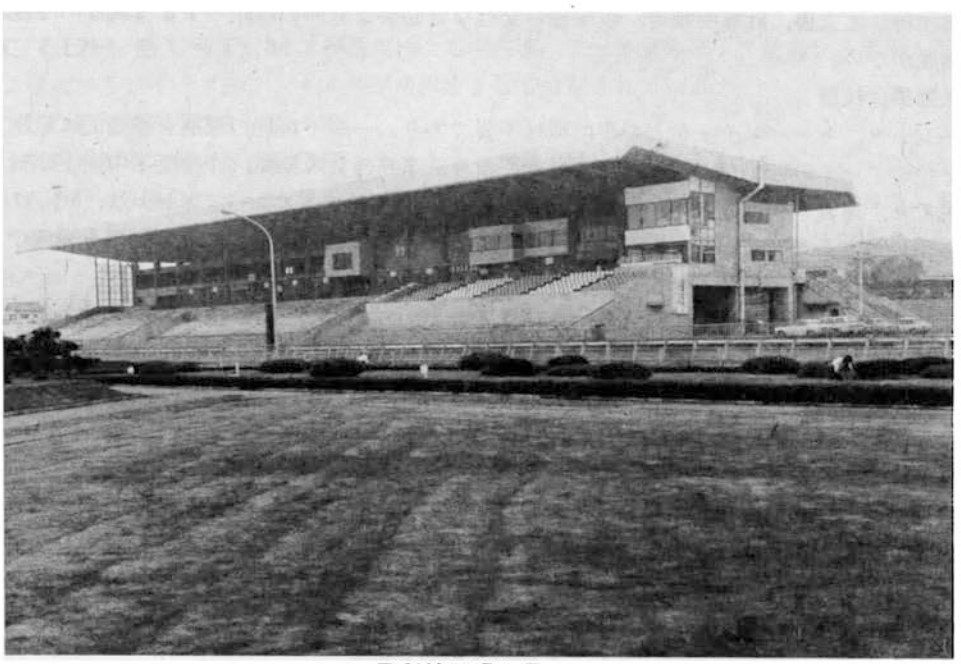
各種類の発券，発売業務をおこなう．各発売機から発売と同時発信される信号は制御装置でチェックの上，発売機に返送され，券をプリントする.



オッズ制御装置



浜名湖競艇場全景



足利競馬場全景

❸ CEC EDPシステム

CEC, EDP システムは10シリーズ～60シリーズまで20システムが標準システムとして用意されている．いずれもCKシリーズコードキーヤ　CKシリーズマルチターミナルを利用した誰れにでも操作できブック式シートまたはパネルシートの交換によって多種業務に共用できる導入効率のよいシステムである．

以下にCEC EDPシステムの特長，および各種業務の事例について紹介する．

1．特長

誰でも簡単にデータエントリーができる実名ワンタッチ入力方式

データ入力はブック式シートに印刷された実名の商品名、取引先名、部品名、データ処理の項目などの該当項目を特殊ペンで軽くタッチし、あとは数量を入力するテンキーの操作だけで完了するワンタッチ・ダイレクトエントリー方式であり、誰れにでも気軽に入力業務をすすめることができる．

システム運営に全員が参加できる

誰れにでも簡単に操作できるので，現在のオフィスの構成員そのままでスムーズに，**EDP**化に移行することが可能，特に専門要員を採用したり，養成する必要はない．

全く無駄のないシステム方式

問合わせ，検索には実名で記入されている**CK**シリーズのブック式コードキーヤの項目を特殊ペンで軽くタッチするだけですべての内容が**CMD**シリーズキャラクタディスプレイに表示され，出荷指示書，注文書，計算明細書，領収書の発行など必要なもののみ印字する方式なので運用上全く無駄がない．

導入効果は抜群

オペレータ，キーパンチャなどの専門職は不要であり，一事不再理の原則を徹底的に追求し，コード化やデータの転記などの作業は一切なくなり，事務経費の節減，作業能率の向上が無理なく実現でき，運営コストも大巾に低減される．また，**CK**シリーズコードキーヤは，ブック式シートを差しかえることにより全部門で共用することができるので，抜群の導入効果を発揮できる．

システム操作はすべてディスプレイとの対話形式で

システム操作はすべて**CMD**シリーズキャラクタディスプレイでおこなわれる．問合わせ業務，処理結果の表示はもちろんリアルタイム処理，対話形式で操作できるのでミス防止も万全，さらに，より機能的にオペレーションをすすめることのできる入力，表示，印字の各機能を1セットにまとめた**CK-10**，**CK-20**，**CK-12**（印字機能なし），**CK-15**（表示機能なし）のマルチターミナルも用意されている．

公衆通信回線，専用回線を利用した広域地域を結ぶオンラインシステムが構成できる

CKシリーズマルチターミナルは標準で8台までリンクすることができる．カプラ，モデムなどデータ通信機能がオプションとして用意されており，データ通信システムのターミナルとして効果的に利用できる．

システムの拡張，機器の増設に対しても十分な機能を充実

業務の拡張，仕事量の増加に対してシステムの手直しなしに簡単にジョブに見合う機器を接続できる機動力を備えている．

ヴァラエティに富んだソフトウェア

財務会計処理，販売在庫管理，商品管理，工程管理などの豊富なソフトウェアパッケージが用意されている．

これから5～10年，安定した利益と最大の効果をあげることができる

貴社専用のペーパーシートで社員が誰れでも操作できるワンタッチ入力方式，用紙類の無駄のないディスプレイによる対話形式のオペレーション，フロッピーディスク，ミニディスク，カートリッジディスク，磁気テープなど業務のスケールに応じて選択できる機動性，どんなシステムにも共通して使用できる便利なオペレーティングシステム・モニタマークII，これらにより長期間にわたりユーザー利益を保証することができる．

2．主な適用業務

1．売上管理	10．資金管理	19．ガス・電気・水道料金計算
2．仕入管理	11．固定資産管理	20．各種税徴収業務
3．販売在庫管理	12．財務会計	21．診療報酬請求
4．商品管理	13．経費管理	22．入院会計
5．生産計画	14．債権・債務管理	23．外来会計
6．工程管理	15．棚卸資産管理	24．検診データ管理
7．外注購買管理	16．給与計算	25．各種金融業務
8．原価計算	17．人事管理	26．その他
9．手形管理	18．各種統計	

3．システム構成

CEC EDP SYSTEMは事業規模，処理内容，データ量等により最適システムが選択できるよう次の5シリーズ20システムが標準構成として用意されている．

（イ） EDPシステム10シリーズ

小規模事業所向けの専用**EDP**システムとして最小構成のシステムである．データファイルに**CFP-500**フロッピディスク，**CFD-300**ミニディスクが接続されており，オフィスマシンとして充分なデータファイル機能を備えている．また，ユーザープログラムの作成，オペレーションがより能率的におこなえる**CS-408**高速紙テープリーダ，**CS-420**高速紙テープパンチャ，**CLP-4100A**ラインプリンタなどの接続システムも用意されている．

（ロ） EDPシステム20シリーズ

データのファイルに**DRISK-2002**カートリッジ磁気ディスクが接続されたシステムである．

はなれた，個別のセクションでの専用端末として**CK**シリーズコードキーヤ，**CMD**シリーズキャラクタディスプレイなどが複数台接続可能であり，各事業部，セクションを総合的にリンクした**EDP**システムを構成することができる．

（ハ） EDPシステム30シリーズ

CPUの記憶装置を16K語まで増設し，カートリッジ磁気ディスクを2台接続したシステム中

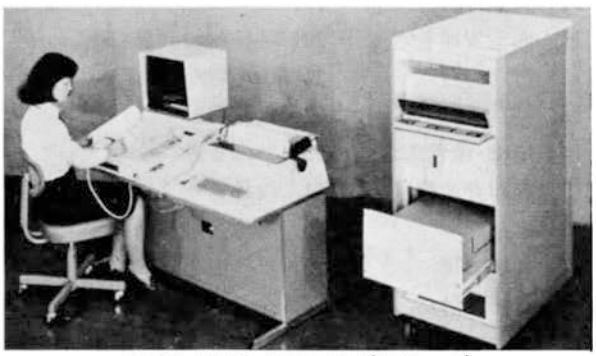
規模の事業所の専用マシンとしては充分なファイル機能をもち，各部，各セクションを連けいした総合的なEDP化が推進できる．

(ニ) EDPシステム50シリーズ

カートリッジデスクを2台，さらに入出力装置を複数台設置したシステム特に502システムはCMT-8磁気テープ装置が加えられ大容量のデータファイルが可能である．

(ホ) EDPシステム60シリーズ

デュアルシステムが容易に構成できる基本の601システムと，バックアップ機能を備えた602，603システムが用意されている．磁気テープ装置は標準4台まで接続可能であり，システム機能は中型機に匹敵する．



CEC EDP システム10シリーズ



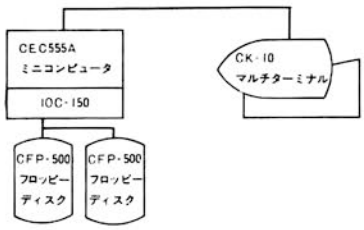
CEC EDP システム20シリーズ



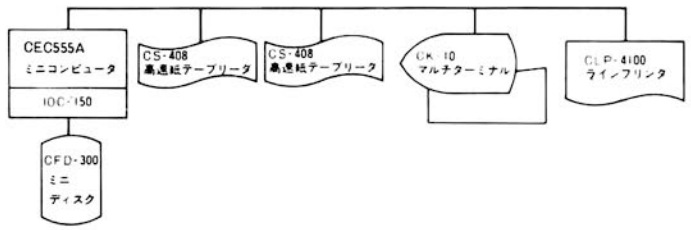
CEC EDP システム60シリーズ

CEC EDP SYSTEM 101

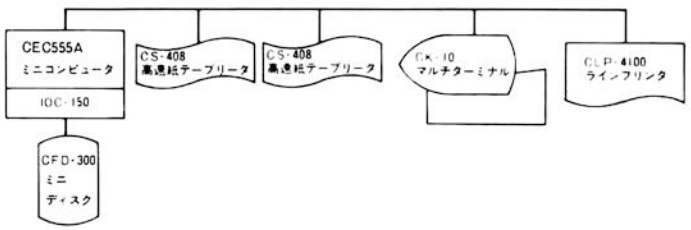
A：130K語
B：260K語



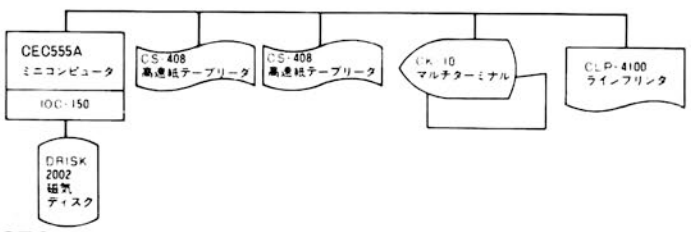
CEC EDP SYSTEM 102（A,B）



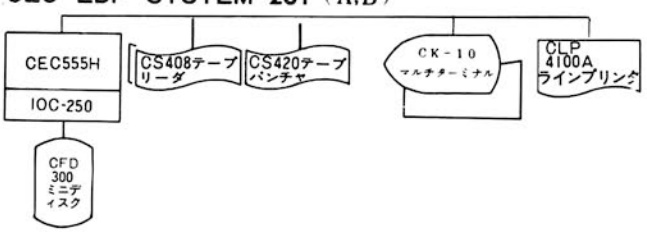
CEC EDP SYSTEM 103（A,B）



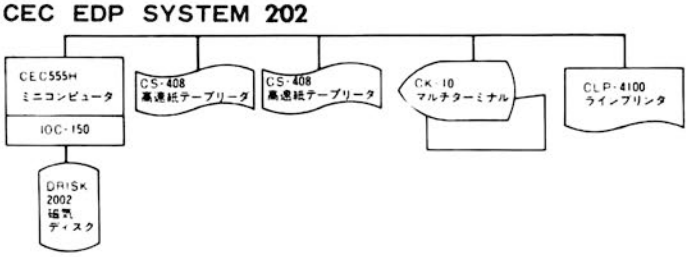
CEC EDP SYSTEM 104



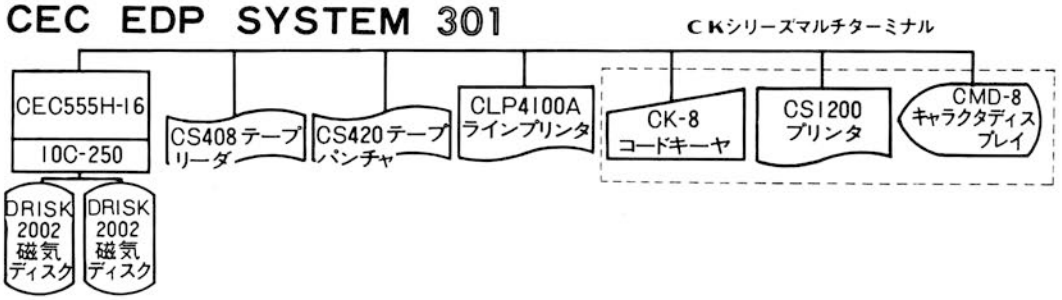
CEC EDP SYSTEM 201（A,B）



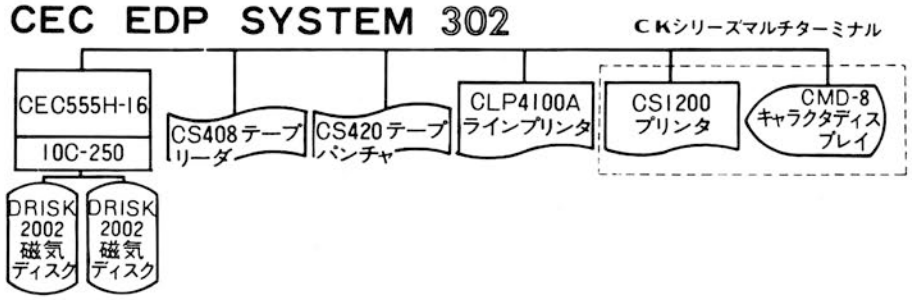
CEC EDP SYSTEM 202



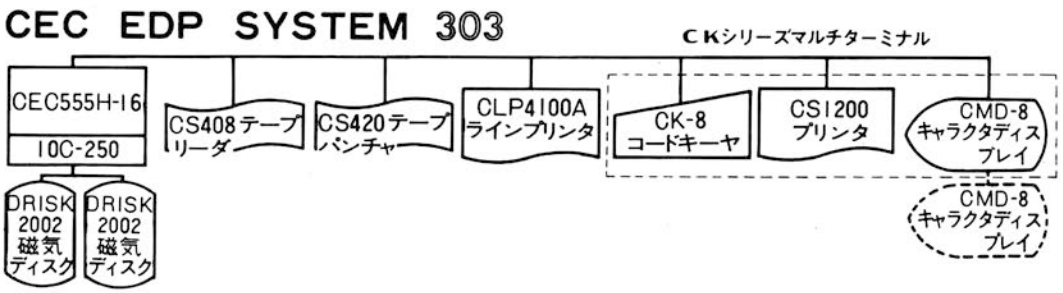
CEC EDP SYSTEM 301



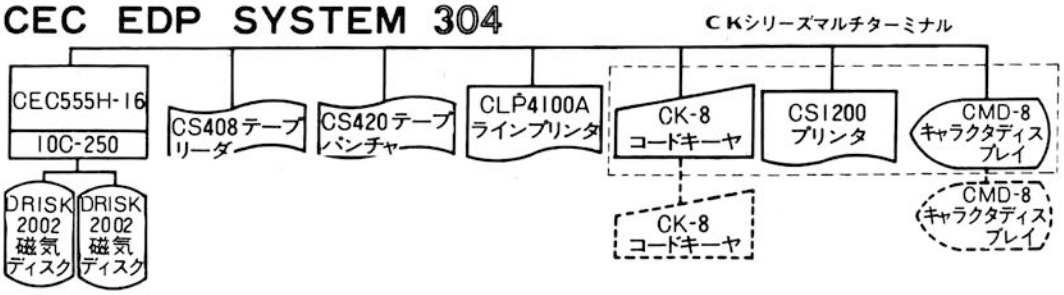
CEC EDP SYSTEM 302



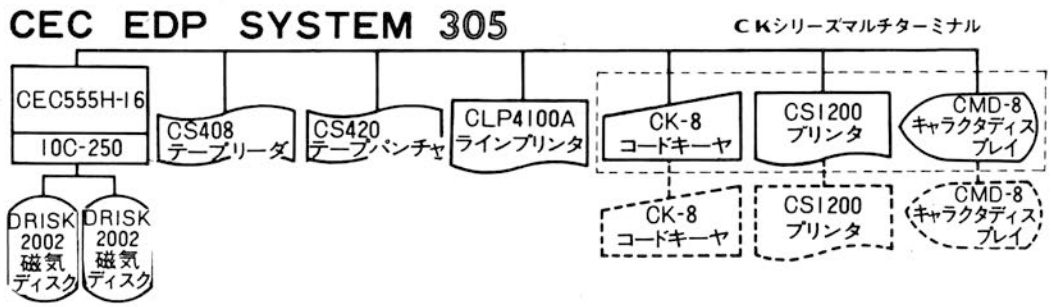
CEC EDP SYSTEM 303



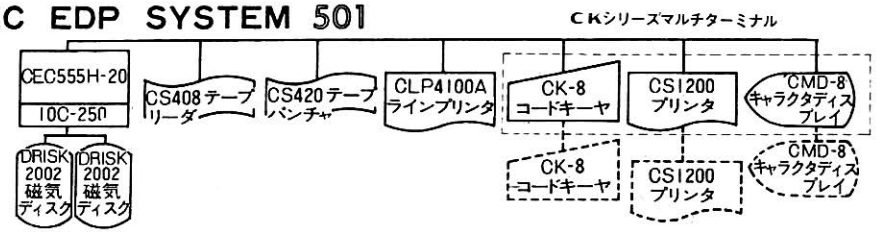
CEC EDP SYSTEM 304



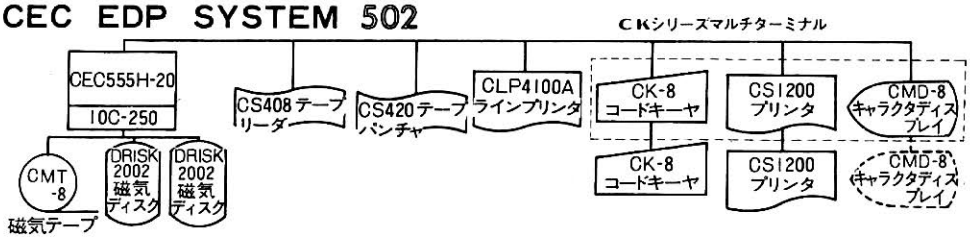
CEC EDP SYSTEM 305



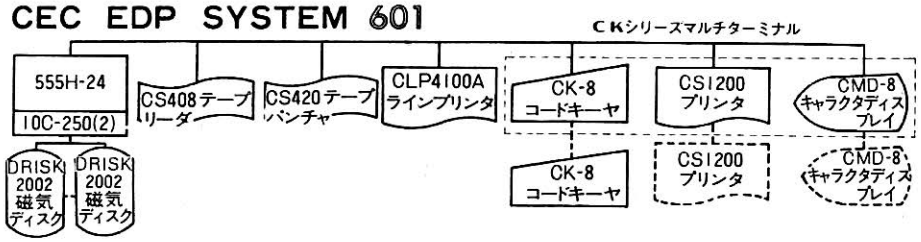
CEC EDP SYSTEM 501



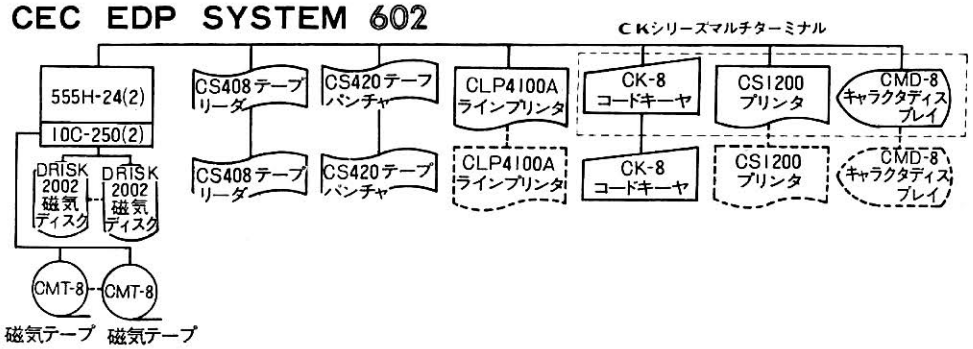
CEC EDP SYSTEM 502



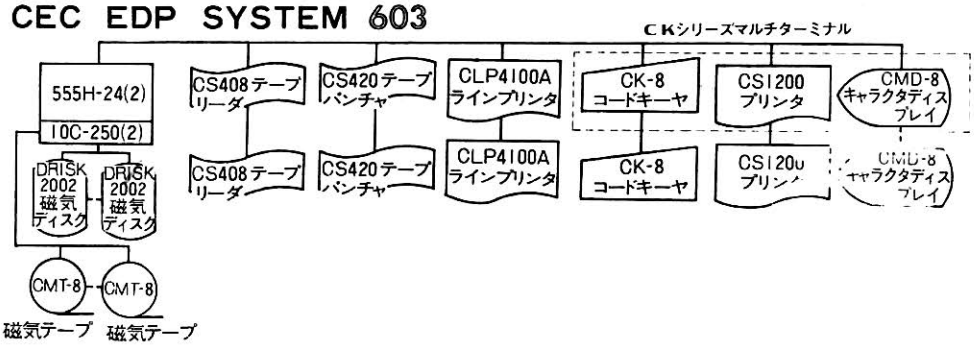
CEC EDP SYSTEM 601



CEC EDP SYSTEM 602



CEC EDP SYSTEM 603



事例－1　財務会計処理

ＣＥＣ財務会計処理システムは，コンピュータ化への 移行に際して,規模，内容に応じて最適システム，最適方式が提供できること，運営に当って，もっとも繁雑な手続きを必要とするデータ入力部門は万能入力端末装置コードキーヤによって，いつでも，誰にでも入力業務が遂行できる機能を備えたことが特長である．従ってシステム運営にあたって特別な専門職を配置する必要がないため，コンピュータ化への移行を円滑に推進することができる．

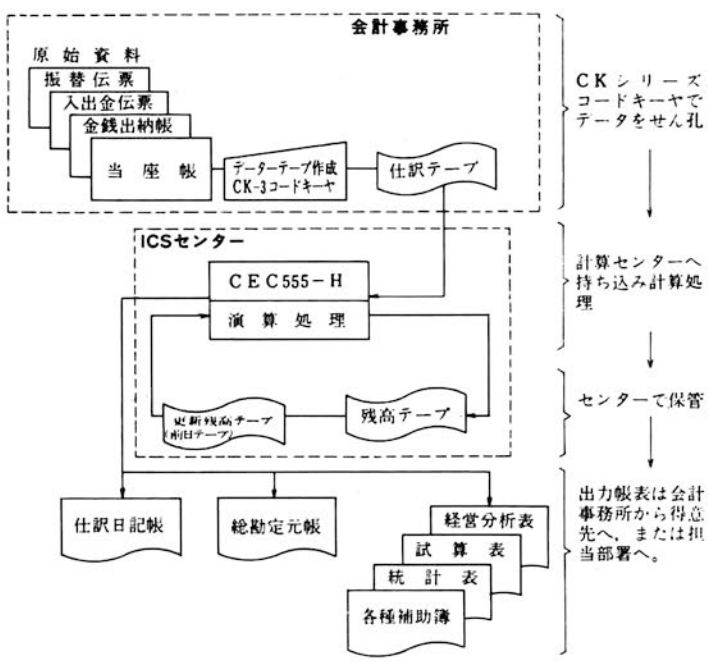
省力化の最大の眼目であるデータ入力部に使用される万能入力端末装置コードキーヤは，パネルシートに実名の語で印刷されている項目に対応したキーと，数値を入力するテンキーの操作だけで正確なデータが入力できる装置で，誰にでも操作でき，キーパンチャ，オペレータなど特殊な専門職を全く必要としないばかりでなく，原票以外のパンチカードなど紙の浪費もなく記入や転記の無駄も省くことができ，本格的な省資源，知識集約形，情報化社会の移行の担い手として各分野の専門家に高く評価されている．

特　長

1．通常仕訳はもちろんのこと金銭出納帳，当座帳からでも入力が可能である．
2．振替伝票（入出金伝票，金銭出納帳，当座帳）の勘定科目名そのものをCK－3コードキーヤ の科目名キーを押すだけで入力できる．
3．現在事務所内でご使用の科目名を探し出す要領で操作できる．
4．科目をワンタッチで処理できるためにタッチ数が少なく従来の端末機にくらべ，1／3～1／4の手間で済む．

標準処理業務

1．仕訳日記帳の作成
2．総勘定元帳の作成
3．試算表の作成
　(1)　貸借対照表の作成
　(2)　損益計算書の作成
4．経営分析表
　(1)　資金繰実績表
　(2)　前年実績対比分析表
5．各種補助簿の作成



工具・器具	流動資産３	通知預金Ｊ	通知預金Ｉ
什器備品	流動資産２	その他預金Ｔ	普通預金Ｈ
車輌運搬具	流動資産１	その他預金Ｓ	普通預金Ｇ
附属設備	不渡手形	定期預金Ｒ	普通預金Ｆ
その他流動資産2	仮払金	定期預金Ｑ	普通預金Ｅ
その他流動資産1	未収入金	定期預金Ｐ	当座預金Ｄ
流動資産８	立替金	定期預金Ｏ	当座預金Ｃ
流動資産７	貸付金１	積立預金Ｎ	当座預金Ｂ
流動資産６	前渡金	積立預金Ｍ	当座預金Ａ

会社名 _____ 会社コード 00000000

試　算　表・貸借対照表

起票	検印	担当者	作成者

昭和 50 年 05 月 31 日 現在

勘定科目	準標コード	前月残高	当月発生 借方	当月発生 貸方	当月残高	構成比率
流動資産 現金	1610	25163	4322735	3939189	408709	2
当座預金	1609	4796090	9986500	9158010	5624580	32
普通預金	1605	471460		20000	451460	2
通知預金	1601	200000			200000	1
積立預金	1511	1516000			1516000	8
定期預金	1507	8800000		5000000	3800000	21
その他預金	1503	876713			876713	4
受取手形	1411	6696709	2400000	2300000	6796709	38
売掛金	1410	9998671	17950000	9090500	18858171	107
商品・製品	0408A	2413000			2413000	13
仕掛品	0409A	7386454			7386454	42
原材料	0409B					
貯蔵品	0410A					
前渡金	1409	469782		200000	269782	1
貸付金	1408	3998500			3998500	22
立替金	1407	110463			110463	
未収入金	1406	211550			211550	1
仮払金	1405	176494	20000		196494	1
前払費用	0418B					
不渡手形	1404	500000			500000	2
	1403	3601345			3601345	20
	1402					
	1401					
	1311					
	1310					
	1309					
	1308					
	1307					
その他流動資産	1306					
貸倒引当金	0411B					
流動資産合計		52248394	34679235	29707699	57219930	326
固定資産 建物	0311B	21165000			21165000	120
附属設備	1304	22462800			22462800	127
構築物	0311A					
機械装置	0310B					
車輌運搬具	1303	3843560	537000		4380560	24
工具・器具・備品	1302	3984200	236790		4220990	24
土地	0309B	62240489			62240489	354
建設仮勘定	0309A					
その他有形固定資産	0307B					
減価償却引当金	0307A					
	0308B					
	0308A					
電話加入権	0306B	371000			371000	2
出資金	0306A	2127000			2127000	12
投資有価証券	0305B	303000			303000	1
保証金	0305A	1000000			1000000	5
	0304B					
固定資産合計		117497049	773790		118270839	673
繰延資産 権利金	0304A					
創業費	0303B					
開発費	0303A					
	0302B					
繰延資産合計						
資産の部合計		169745443	35453025	29707699	175490769	1000

勘定科目	準標コード	前月残高	当月発生 借方	当月発生 貸方	当月残高	構成比率
流動負債 支払手形	1211	5579479●		12737900	18317379●	104
買掛金	1210	13615210●	15630000	7868960	5854170●	33
短期借入金	1209	15218000●	2500000		12718000●	72
未払金	1208	7743084●		186790	7929874●	45
前受金	1207	563000●			563000●	3
預り金	1206	1752277●		302735	2055012●	11
仮受金	1204	231927●			231927●	1
未払費用	0302A					
未払配当金	0301B	3000000●		2426000	5426000●	30
未払役員賞与	0301B	2000000●			2000000●	11
法人税等充当金	1203	2000000●			2000000●	11
割引手形	1202	700000●			700000●	3
	1111	3600●			3600●	
	1110					
	1109					
	1108					
	1107					
	1106					
	1105					
	0301A					
	0201A					
その他流動負債	1201	4003227●			4003227●	22
流動負債合計		56409804●	18130000	23522385	61802189●	352
固定負債 社債	0202A					
長期借入金	1104	80000000●			80000000●	455
	1102					
	1101					
	0202B					
その他固定負債	1103					
固定負債合計		80000000●			80000000●	455
引当金 賞与引当金	0203A					
退職給与引当金	0203B	876000●			876000●	4
価格変動準備金	0204A					
修繕引当金	0204B					
	0205A	503000●			503000●	2
	0205B					
引当金合計		1379000●			1379000●	7
負債の部合計		137788804●	18130000	23522385	143181189●	815
資本金	0206A	20000000●			20000000●	113
法定準備金 資本準備金	0206B	1500000●			1500000●	8
利益準備金	0207A	600000●		250000	850000●	4
その他法定準備金	0207B					
別途積立金	0208A					
剰余金 退職積立金	0208B	1300000●		244000	1544000●	8
	0209A					
	0209B					
	0210A					
又は欠損金 前期繰越利益	0210B	388113●			388113●	2
前期繰越損失	0211A					
未処分利益	0211B	4750000●	2920000		1830000●	10
未処理損失	0111B					
当期純利益又は当期純損失	0111A	34185261●		2778941	61976671●	35
資本の部合計		31956639●	2920000	3272941	32309580●	185
負債・資本の部合計		169745443●	21050000	26795326	175490769●	1000
合計			56503025	56503025		

会社名 ＿＿＿＿＿＿　　会社コード 00000000

試 算 表・損益計算書

社長	部長	課長	作成者

自 昭和50年05月01日
至 昭和50年05月31日

勘定科目	整理コード	前月迄の累計	当月発生 借方	当月発生 貸方	当月迄の累計	対売上高比率
[illegible]	0101A					
売上原価 仕入高	0911	5754300	14617490		20371790	436*
	0910	1563213			1563213	33*
仕入値引・戻し高	0909	256300*		4000000	4256300*	91
[illegible]	0407B					
[illegible]	0402A					
売上総利益		[illegible]	[illegible]	21950000	29032515*	621
販売費及び一般管理費		18980374	4515144		23495518	502*
営業利益(損失)		2744631*	19157634	21950000	5536997*	118
営業外損失		263134	13425		276559	5*
経常利益		3418526*	19171059	21950000	6197467*	132
特別損失 [illegible]	0107B					
固定資産売却損	0107A					
	0106B					
	0106A					
税引前当期利益(損失)		3418526*	19171059	21950000	6197467*	132
法人税等充当額	0105B					
当期利益(損失)		3418526*	19171059	21950000	6197467*	132

勘定科目	整理コード	前月迄の累計	当月発生 借方	当月発生 貸方	当月迄の累計	対売上高比率
営業収益 売上高	1011	28924218*		17950000	46874218*	1003
	1010					
	1009					
	1008					
	1007					
	1006					
売上値引・戻り高	1005	138000	25000		163000	3*
純売上高		28786218*	25000	17950000	46711218*	1000
営業外収益 受取利息	1003	213613*			213613*	*
受取配当金	010B					
雑収入	1002	723416*			723416*	15
その他営業外収益	1001					
営業外収益合計		937029*			937029*	20
特別利益 貸倒引当金戻入	0110A					
[illegible]	0109B					
固定資産売却益	0109A					
	0108B					
	0108A					
特別利益合計						

販売費及び一般管理費明細

勘定科目	整理コード	前月迄の累計	当月発生 借方	当月発生 貸方	当月迄の累計	対売上高比率
役員報酬	0907	1036213			1036213	22*
給料手当	0906	10439940	3711000		14150940	302*
退職金	0905	73621			73621	1*
法定福利費	0904	1636657			1636657	35*
福利厚生費	0903	1254935	236478		1491413	31*
広告宣伝費	0902	13621			13621	
運賃	0901	430172			430172	9*
荷造包装費	0811		39447		39447	
旅費交通費	0810	416764	60618		477382	10*
燃料費	0809	89939			89939	1*
通信費	0808	103673	40		103713	2*
水道光熱費	0807	84317	14359		98676	2*
租税公課	0806	23633	20000		43633	
接待交際費	0805	83796	37239		121035	2*
消耗品費	0804	35641	5100		40741	
事務用品費	0803	23143			23143	
地代・家賃	0802	1542692	330000		1872692	40*
修繕費	0801	545465	22400		567865	12*
保険料	0711	160528			160528	3*
支払手数料	0710	43600			43600	
	0709	887933			887933	19*
	0708					
	0707					
	0706					
	0705					
	0704					
	0703					
	0702					
雑費	0701	54091	38463		92554	1*
減価償却費	0104B					
販売費・一般管理費合計		18980374	4515144		23495518	502*
営業外損失 支払利息・割引料	0403B	190483	13425		203908	4*
貸倒引当金繰入	0404A					
雑損失	0401A	72651			72651	1*
貸倒損失	0403A					
	0404B					
営業外損失合計		263134	13425		276559	5*

製 造 原 価 明 細

勘定科目	整理コード	前月迄の累計	当月発生 借方	当月発生 貸方	当月迄の累計	対売上高比率
期首材料棚卸高	0102A					
材料仕入高	0611					
副資材仕入	0610					
期末材料棚卸高	0104A					
賃金手当	0609					
退職手当	0608					
雑給	0607					
法定福利費	0606					
福利厚生費	0605					
	0504					
	0503					
外注加工費	0604					
電力料	0603					
水道光熱費	0602					
運賃	0601					
修繕費	0511					
租税公課	0510					
地代・家賃	0509					
保険料	0508					
消耗品費	0507					
	0502					
旅費交通費	0506					
通信費	0505					
	0501					
	0407A					
	0406B					
	0406A					
	0405B					
	0405A					
雑費	0401B					
減価償却費	0105A					
期首仕掛品棚卸高	0102B					
期末仕掛品棚卸高	0103B					
当期製品製造原価						
備考						

経営分析表

所長	統括	担当者	作成者

昭和 50 年 05 月分

会社名 ＿＿＿＿＿＿＿＿ 会社コード 00000000

経営分析指数表

	分析項目	分析値		分析値グラフ
財務比率	流動比率	925*	%	==========
	当座比率	623*	%	==========
	売上債権対買入債務比率	1061*	%	==========
	固定比率	3660*	%	==========
	固定長期適合率	1053*	%	==========
	負債比率	4431	%	==========
	借入金依存度	2869	%	==========
	現金預金バランス	208*	%	==========
	手形バランス	296	%	==========
回転率	売上債権回転率	18*	回	==========
	棚卸資産回転率	47*	回	==========
	流動資産回転率	8*	回	========
	固定資産回転率	3*	回	==========
	総資本回転率	2	回	==========
	自己資本回転率	14	回	==========
利益率	売上高総利益率	621	%	==========
	売上高営業利益率	118	%	==========
	総資本営業利益率	31	%	======
	総資本純利益率	35	%	=======

(＊印は棒グラフ超過・分析値参照)

資金繰実績表

		分析項目	月実績	月実績	月実績	当月実績
		前月繰越高(1)		4111010	3921998	5492713
営業資金	収入	現金売上				
		売掛金回収		846463	1240000	560000
		受取手形期日落		1000000	1066000	2300000
		前受金				
		雑収入		32986		
		諸口合計				11105500
		その他		13236936	18243955	302735
		小計2		19227395	20549955	14268235
	支出	現金仕入				747630
		買掛金支払		1482369	1500250	1464000
		支払手形決済				
		前渡金				
		人件費		235407	237932	3947478
		製造経費		1076115	1168411	
		販売・一般管理費		174536	209625	223307
		金融費用		78000	150000	
		固定資産購入		100000	201000	50000
		法人税・市府民税				
		諸口合計				6623784
		その他		16119980	16033482	20000
		小計3		19266407	19500700	13076199
		(1)+(2)−(3)差引過不足(4)		4071998	4971233	6684749
財務資金	収入	前年性預金引出				
		借入金				
		手形減引				
		小計5				
	支出	前年性預金引入		150000	150000	
		借入金返済				
		小計6		150000	150000	
		(4)−(5)−(6)当月繰越高		3921998	4821253	6684749

前年実績対比分析表

項目	前年度実績 金額	前年度実績 対売上比率	当月実績 金額	当月実績 対売上比率	増減益額	対比率
純売上高	15743900*		17925000*	1000	2181100*	1138
売上原価	12826425		10617490	592*	2208935*	827
売上総利益	2917475*		7307510*	407	4390035*	2504
一般管理費	1879543		4515144	251*	2635601	2402
内人件費	1233666		3947478	220*	2713812	3199
製造経費	2516981				2516981*	
内人件費	1613782				1613782*	
内外注費	610536				610536*	
営業利益	1037932*		2792366*	155	1754434*	2690
支払利息・割引	98642		13425		85217*	136
当期利益	939290*		2778941*	155	1839651*	2958





元　　帳

会社コード　00000000　　科目コード　　科目名　シフクニツキチヨウ　　P.　1

月　日	伝票 No.	借（カモク）方	貸（カモク）方	残（キンガク）高	（相手科目名）	摘　　要
05.01	0001	ﾘｮﾋ.ｺｳｿｳﾋ	ｹﾝｷﾝ	750	ｵｵｻｶｺｳﾂｳ	
	0002	ｼｮｳﾓｳﾋﾝﾋ	ｹﾝｷﾝ	500	ﾔﾉｶﾅﾓﾉ	
05.02	0003	ｹﾝｷﾝ	ﾌｼｳﾖｷﾝ-E	20,000	ﾋｷﾀﾞｼ	
	0004	ﾄｳｻﾞﾖｷﾝｰA	ｹﾝｷﾝ	21,000	ｱｽﾞｻｲﾚ	
05.10	0005	ｼﾞﾕｳｷ.ﾋﾞﾋﾝ	ﾄｳｻﾞﾖｷﾝｰA	50,000	ｱｵｲｼﾞﾑ ﾛｯｶｰ	
	0006	ｾﾂﾀｲｺｳｻｲﾋ	ｹﾝｷﾝ	450	ｷﾂｻ ｲｽﾞﾐ ｵｵﾀ	
		ｿｳｼﾝﾋ	ｹﾝｷﾝ	40	ﾃﾙﾀﾞｲ	
05.12	0007	ﾄｳｻﾞﾖｷﾝｰA	ｳｹﾄﾘﾃｶﾞﾀ	300,000	ｵｵｻｶK.K 2/12	
	0008	ｻﾞﾂﾋﾞ	ｹﾝｷﾝ	2,345	ｼﾐｽﾞｼｮｰﾃﾝ	
05.15	0009	ｼｮｳﾓｳﾋﾝﾋ	ｹﾝｷﾝ	4,600	ﾔﾉｶﾅﾓﾉ ｵｵﾀ	
		ﾄｳｻﾞﾖｷﾝｰA	ｼｮｸﾞﾁ	1,000,000	ﾃｲｷ ｶｲﾄｸ	
		ｹﾝｷﾝ	ｼｮｸﾞﾁ	4,000,000	ﾃｲｷ ｶｲﾄｸ	
		ｼｮｸﾞﾁ	ﾃｲｷﾖｷﾝｰD	5,000,000	ﾃｲｷ ｶｲﾄｸ	
		ｳﾘｶｹｷﾝ	ｳﾘｱｹﾞﾀﾞｶｰ1	3,600,000	ｶﾝｻｲ K.K	
	0010	ﾄｳｻﾞﾖｷﾝｰA	ｼｮｸﾞﾁ	3,462,000	ﾉﾑﾗｼｮｰｼﾞ	
		ｳｹﾄﾘﾃｶﾞﾀ	ｼｮｸﾞﾁ	2,000,000	ﾉﾑﾗｼｮｰｼﾞ 8/20	
		ｼｮｸﾞﾁ	ｳﾘｶｹｷﾝ	5,462,000	ﾉﾑﾗｼｮｰｼﾞ	
05.17	0011	ｶﾘﾊﾞﾗｲｷﾝ	ｹﾝｷﾝ	20,000	ﾔﾏﾓﾄ ｶﾘﾊﾞﾗｲ	
		ｶｲｶｹｷﾝ	ｼｲﾚﾈﾋﾞｷ	4,000,000	ﾔﾊﾀ ｼｮｰｶｲ K.K	
	0012	ﾀﾝｷｶﾘｲﾚｷﾝ	ｼｮｸﾞﾁ	2,500,000	ｻﾝﾌｷﾞﾝｺｳ	—
		ｼﾊﾗｲﾘｿｸ	ｼｮｸﾞﾁ	13,425	ｻﾝﾌｷﾞﾝｺｳ	
		ｼｮｸﾞﾁ	ﾄｳｻﾞﾖｷﾝｰA	2,513,425	ｻﾝﾌ ﾊﾝｻｲ	
	0013	ｼﾞﾀｲ.ﾄｷﾝ	ｼｮｸﾞﾁ	330,000	ｲｼｲ ｼｮｰｼﾞ	
		ｽｲﾄｰｳｺｳｷｿﾋ	ｼｮｸﾞﾁ	9,860	ｶﾞｽ ﾀﾞｲ	

ICSセンター → 連絡事項

元　　帳

会社コード　00000000　　科目コード　　科目名　シフクニツキチヨウ　　P.　2

月　日	伝票 No.	借（カモク）方	貸（カモク）方	残（キンガク）高	（相手科目名）	摘　　要
05.17	0013	ｽｲﾄｰｳｺｳｷｿﾋ	ｼｮｸﾞﾁ	4,499	ｽｲﾄｰﾀﾞｲ	
		ｼｮｸﾞﾁ	ﾄｳｻﾞﾖｷﾝｰA	344,359	ｲｼｲ ｼｮｰｼﾞ	
05.18	0014	ｳﾘｶｹｷﾝ	ｳﾘｱｹﾞﾀﾞｶｰ1	3,600,000	ｵｵｻｶ K.K	
		ｳﾘｶｹｷﾝ	ｳﾘｱｹﾞﾀﾞｶｰ1	1,500,000	ﾌｶﾔﾏ ｼｮｰｶｲ	
		ﾘｮﾋ.ｺｳｿｳﾋ	ｹﾝｷﾝ	2,000	ﾄｳｷﾖｰ ｴｷ	
05.19	0015	ｼｲﾚﾀﾞｶｰ1	ﾄｳｻﾞﾖｷﾝｰA	300,000	ﾔﾏﾄK.K	
		ｻﾞﾂﾋﾞ	ﾄｳｻﾞﾖｷﾝｰA	36,118	ｲｼﾀﾞ ｼｮｰﾃﾝ	
		ｼｲﾚﾀﾞｶｰ1	ﾄｳｻﾞﾖｷﾝｰA	447,630	ﾔﾏｶ K.K	
	0016	ｼﾅﾘｮｳ.ｳﾝﾊﾟﾝｸﾞ	ｼｮｸﾞﾁ	537,000	ﾄﾖﾀｵｰﾄ ｶﾛｰﾗﾊﾟﾝ	
		ｼｮｸﾞﾁ	ﾏｴｳﾘｼｷﾝ	200,000	ﾄﾖﾀ ｵｰﾄ	
		ｼｮｸﾞﾁ	ｼﾊﾗｲﾃｶﾞﾀ	337,000	ﾄﾖﾀ ｵｰﾄ9/19	
		ﾆｭｰｸﾘﾈｳｿｳﾋ	ｹﾝｷﾝ	39,447	ﾎﾘ ｳﾝｿｳ	
	0017	ｳﾘｶｹｷﾝ	ｳﾘｱｹﾞﾀﾞｶｰ1	4,600,000	ﾄﾊﾞ ｼｮｰｼﾞK.K	
		ｳﾘｱｹﾞﾈﾋﾞｷ	ｳﾘｶｹｷﾝ	25,000	ｵｵｻｶK.K	
05.20	0018	ｶｲｶｹｷﾝ	ｼｮｸﾞﾁ	3,600,000	ｷﾖｰﾄK.K	
		ｼｮｸﾞﾁ	ｼﾊﾗｲﾃｶﾞﾀ	2,400,000	ｷﾖｰﾄK.K	
		ｼｮｸﾞﾁ	ﾄｳｻﾞﾖｷﾝｰA	1,200,000	ｷﾖｰﾄK.K	
		ｶｲｶｹｷﾝ	ﾄｳｻﾞﾖｷﾝｰA	600,000	ﾔﾏｶﾐｼｮｰｶｲ	
		ｶｲｶｹｷﾝ	ﾄｳｻﾞﾖｷﾝｰA	864,000	ｼﾓﾀﾞK.K	
	0019	ｶｲｶｹｷﾝ	ｼｮｸﾞﾁ	6,566,000	ﾅﾗ ｷｺｳ	
		ｼｮｸﾞﾁ	ﾄｳｻﾞﾖｷﾝｰA	2,566,000	ﾅﾗ ｷｺｳK.K	
		ｼｮｸﾞﾁ	ｼﾊﾗｲﾃｶﾞﾀ	2,000,000	ﾅﾗ ｷｺｳ 8/20	
		ｼｮｸﾞﾁ	ｼﾊﾗｲﾃｶﾞﾀ	2,000,000	ﾅﾗ ｷｺｳ9/20	
	0020	ｿｾﾞｲｺｳｶ	ｹﾝｷﾝ	20,000	ｲﾝｼ ﾀﾞｲ	

事例－2　販売在庫管理

概　要

傘下に多数の営業店をもち，かつ商品が多岐多種にわたる業種においては，各営業店の正確な売上予測，それに伴う供給側の在庫量，生産加工量の適正な管理がきわめて重要な課題になっている．従って日々の商品のデータを整理処理して，正確に，早期にマネージメント担当に正確な情報として現場よりフィードバックする必要がある．また，営業店への仕切価格の変動等も誤りなく伝票上に記載処理することが販売政策上の重要問題のひとつであり，このような複雑多岐にわたる流通段階での事務処理に費される労力と時間はたいへんなものとなりコストアップの要因のひとつになっている．

このシステムでは，これらの問題点を適切に解決するために

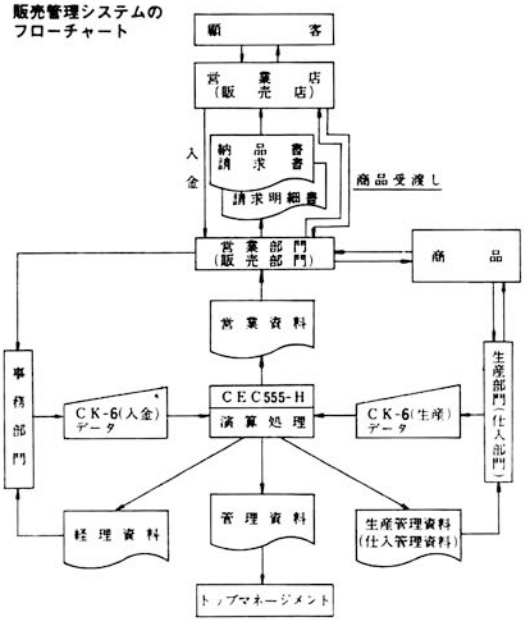
1．日計業務としてブロック別，コース別の売上明細書の作成（品種別の数量，金額等）
2．週計業務としてコース別の入金状況一覧表の作成（標準５日ごと）
3．月計業務として
　(1)　得意先売上明細書の作成
　(2)　得意先台帳（総括表の作成）
　(3)　請求書，領収書，入金明細表の自動発行
　(4)　品種別売上実績表の作成
　(5)　集金予定表の作成
　(6)　地域別売上実績表の作成
　(7)　その他給与計算，財務緒表の作成

等の販売管理上の主なデータ処理をCEC555電子計算機に万能入力端末装置コードキーヤを併設した新しいシステム方式により一貫しておこなうシステムである．

主な処理業務

1．日計ごとの業務
　(1)　ブロック別．コース別，品種別，数量，金額．
　(2)　大口店，納品明細表
2．週計ごとの業務
　(1)　コース別，入金状況表（５日ごと）
3．月計ごとの業務
　(1)　ブロック別，コース別，品種別，数量，金額（得意先別売上明細表に打ちだす）
　(2)　請求書，領収書，入金明細書
　(3)　得意先台帳（総括表）
　(4)　品種別売上実績表
　(5)　集金予定表
　(6)　給与計算
　(7)　地区別売上実績表
4．その他
　(1)　工場内課別処理量と金額
　(2)　財務諸表の作成
　(3)　給与計算

販売管理システムのフローチャート



販売在庫管理用ブック式シート

SHEET No.	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

	01	02	03	04	05	06	07	08
01								
02	飯田カメラ	近藤カメラ	松屋カメラ	ユニフォート高麗店	OS 芦屋	塚口カメラ塚口店	ABCカメラ淡路	KCU 淡路
03	四条カメラ	ラッキーフォート	山田写真機店	宝 カメラ	OS 立花	宝塚映画カラー	ABCカメラ天六	KCU 天六
04	クラウンカメラ	南海フォート	大正カメラ	ヒカリカメラ牧方	OS 桃谷	東部カラー	ABCカメラ梅田	KCU 梅田
05	小牧フォート	竹田カメラ	フォートタマダ	ABCカメラ北千里	OS 塚口	フジカメラ八尾	ABCカメラ長崎屋	KCU 長崎屋
06	ローズカメラ	太陽カメラ	和田カメラ	ユニフォート内本町店	社内用	塚本カメラ本店	ABCカメラ茨木	KCU 茨木
07	ユニークフォート	小野写真機店	野崎カメラ	田村写真	大阪トイ	さくらカラー南大阪	ABCカメラ南千里	KCU 南千里
08	アキフォート	国立カメラ	都 カメラ	中野カメラ	ベッキカメラ	ニチイ高槻	ABCカメラ北千里	KCU 北千里
09	DPEサクラヤ	近鉄カメラ	三和カメラ	高岡カメラ	ヒカリカメラ正省	渡辺写真館	マルシンカメラ	AO 淡路
10	東 カメラ	藤本カメラ	中西カメラ	大石カメラ	ヨドカワフォート	トキワフォート	大阪フォート	AO 天六
11	トラヤカメラ	スターカメラ	吉田カメラ	上田カメラ	ユニフォート平野店	里内写真機店	ナンバDPコーナー	AO 梅田
12	パールカメラ	岩田フォート	田中カメラ	田辺カメラ	ユニフォート伏見店	河原カメラ梅田	塚口カメラ本店	AO 長崎屋
13								
14								
15	A	B	C	D	E	F	G	H
16	I	J	K	L	M	N	O	P
17	Q	R	S	T	U	V	W	X
18	Y	Z						
19								
20	ア	イ	ウ	エ	オ			
21	カ	キ	ク	ケ	コ			
22	サ	シ	ス	セ	ソ			
23	タ	チ	ツ	テ	ト			
24	ナ	ニ	ヌ	ネ	ノ			
25	ハ	ヒ	フ	ヘ	ホ			
26	マ	ミ	ム	メ	モ			
27	ヤ		ユ		ヨ			
28	ラ	リ	ル	レ	ロ			
29	ワ				ヲ			
30	ン							
31								
32								

CEC CK-6,CK-8 SHEET

ﾌﾞﾛｯｸNo. 1

ｺｰｽNo. 1

得意先別売上明細表

昭和 56年 03月分　　　52頁

品名 / 得意先コード	サービスサイズ	手焼	CPR	CPS	CNO	CRO	8ミリ 16ミリ	マウント	チューブ インターネガ	他社	カメラ修理	雑品	合計	フイルム代金	その他商品	当月合計
12017	ﾊﾝｼﾝCS(ｲﾏｽﾞﾃﾝ)				ｻﾝﾀﾞｲ ﾏｻｺ			ﾆｼﾉﾐﾔｼ ｲﾏｽﾞｱﾗﾎﾞﾌﾁｮｳ 3-13								
数量	134	0	0	0	6	0	0	0	0	0	0	0		0	0	
金額	5622	0	0	0	840	0	0	0	0	0	0	0	6462	0	0	6462
12025	ﾊﾝｼﾝDS(ﾌﾅﾊﾞｼﾃﾝ)				ﾊﾏﾆｼ ﾉﾎﾞﾙ			ﾋﾗｶﾀｼ ﾐﾅﾐﾌﾅﾊﾞｼ 1-1								
	0	0	0	0	0	0	0	0	0	0	0	0		0	0	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12033	ﾊﾝｼﾝDS(ｱﾏｶﾞｻｷﾃﾝ)							ｱﾏｶﾞｻｷｼ ﾐｿﾉﾁｮｳ 12								
	124	0	0	0	[illegible]	0	0	0	0	0	0	0		0	0	
	5208	0	0	0	910	0	0	0	0	0	0	0	6118	0	0	6118
12041	ﾊﾝｼﾝDS(ｻｶｾｶﾞﾜﾃﾝ)				ｽｷﾞﾘ ﾄﾐｺ			ﾀｶﾗﾂﾞｶｼ ｻｶｾｶﾞﾜ 2-6-6								
	0	0	0	0	0	0	0	0	0	0	0	0		0	0	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12050	ﾊﾝｼﾝDS(ｲﾅﾉﾃﾝ)				ｼﾂｲ			ｲﾀﾐｼ ﾐﾅﾐﾏﾁｮｳ ｱｻﾞﾄｽﾉﾏｴ								
	66	0	0	0	2	0	0	0	0	0	0	0		0	0	
	2772	0	0	0	280	0	0	0	0	0	0	0	3052	0	0	3052
12068	ﾊﾝｼﾝDS(ﾉﾏﾃﾝ)				ｺｳﾉ ｽﾐ			ｲﾀﾐｼ ﾉﾏ ｱｻﾞｸﾎﾏｯｶ 516								
	296	0	0	0	14	0	0	0	0	0	0	0		0	0	
	12576	0	0	0	1960	0	0	0	0	0	0	0	14536	0	0	14536
12076	ﾊﾝｼﾝDS(ｹｲﾕｳｼﾖｳ)				ﾊｼﾓﾄ ｹｲｿﾞｳ			ｵｵｻｶｼ ﾃﾝﾉｳｼﾞｸ ﾄﾞｳｼﾞﾝﾊﾞｼﾁｮｳ 251								
	0	0	0	0	0	0	0	0	0	0	0	0		0	0	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12106	ﾊﾝｼﾝDS(ﾉｶﾞﾐﾃﾝ)				ﾒｸﾞﾛ ﾏｻｺ			ﾀｶﾗｽﾞｶｼ ｱﾗｶﾞﾐ 2-10-15								
	0	0	0	0	0	0	0	0	0	0	0	0		0	0	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12114	ﾊﾝｼﾝDS(ﾓﾘﾐｾ)							ｺｳﾍﾞｼ ﾋｶﾞｼﾅﾀﾞｸ ｽﾐﾖｼﾁｮｳ ﾌｶｴｷﾀﾄﾞｳﾘ								
	0	0	0	0	0	0	0	0	0	0	0	0		0	0	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12122	ﾊﾝｼﾝCS(ﾆｼﾉﾐﾔﾃﾝ)				ﾆｼﾀﾞ ﾀﾀﾞｼ			ﾆｼﾉﾐﾔｼ ﾆｼｱｸﾁｮｳ 5-10								
	0	0	0	0	0	0	0	0	0	0	0	0		0	0	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12131	ﾊﾝｼﾝDS(ﾐｽﾞﾃﾝ)							ｲﾀﾐｼ ﾐｽﾞﾁｮｳ 1 ﾐｽﾞﾛｰﾄﾞﾚｲ								
	19	0	0	0	2	0	0	0	0	0	0	0		0	0	
	756	0	0	0	280	0	0	0	0	0	0	0	1036	0	0	1036
12157	ﾊﾝｼﾝDS(ｷｮｳﾊﾞｼﾃﾝ)				ﾅﾙﾑﾗ ｼｽﾞｵ			ｵｵｻｶｼ ﾐﾔｺｼﾞﾏｸ ﾋｶﾞｼﾉﾀﾞﾁｮｳ 3-4-21								
	0	0	0	0	0	0	0	0	0	0	0	0		0	0	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12165	ﾊﾝｼﾝDS(ﾌﾙｶﾜﾊﾞｼﾃﾝ)							ｶﾄﾞﾏｼ ｶﾈｳﾁﾁｮｳ 7-3-1								
	0	0	0	0	0	0	0	0	0	0	0	0		0	0	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12173	ﾊﾝｼﾝCS(ｵｵﾄﾘﾃﾝ)							ｽｲｶﾞｼﾞ ﾎｽﾞﾐﾁｮｳ 10								
	0	0	0	0	0	0	0	0	0	0	0	0		0	0	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12181	ﾊﾝｼﾝCS(ｱｼﾔﾃﾝ)				ﾆｼｸﾞﾁ ｶｽﾞｴ			ｱｼﾔｼ ﾂｷﾄﾞﾁｮｳ 2-19								
	68	0	0	0	0	0	0	0	0	78	0	0		0	0	
	2856	0	0	0	0	0	0	0	0	5312	0	0	8168	0	0	8168
12190	ﾊﾝｼﾝDS(ﾆｼﾉﾐﾔｷﾀｸﾞﾁﾃﾝ)							ﾆｼﾉﾐﾔｼ ﾄﾖﾂﾁｮｳ 3								
	0	0	0	0	0	0	0	0	0	0	0	0		0	0	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12203	ﾊﾝｼﾝDS(ﾌﾙｻﾜﾃﾝ)				ｵｵﾅﾘ ｼﾞｭﾝｺ			ﾀｶﾂｷｼ ﾀﾞｲｶﾝﾁｮｳ 2-23								
	0	0	0	0	0	0	0	0	0	0	0	0		0	0	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12211	ﾊﾝｼﾝDS(ﾀｶﾂｷﾃﾝ)				ﾔﾏﾓﾄ ﾐﾉﾙ			ﾀｶﾂｷｼ ｷﾀｿﾞﾉﾁｮｳ 14-4								
	110	0	0	0	10	0	0	0	0	0	0	0		0	0	
	4620	0	0	0	1400	0	0	0	0	0	0	0	6020	0	0	6020

CEC中央電子株式会社　　CEC555

納品請求明細表

ブロックNo. 2　　コースNo. 7

得意先コード 03425

得意先名 モリノフォート 様

昭和 50年 5月 15日

CEC中央電子株式会社
八王子市元本郷町1～9～9
電話 八王子(0426)23-1211
TELEX 2862-529

月日	受注No.	内訳 品名コード	品名	数量	単価	同P現	金額
310	051	305	KﾌﾟﾘﾝﾄBR1	16	60.00		960.00
310	051	331	K.DP	1	160.00		160.00
310	301	101	SE	6	38.00	140	368.00
310	302	101	SE	21	38.00	140	938.00
310	311	101	SE	2	38.00	140	216.00
310	312	101	SE	11	38.00	140	558.00
310	360	119	2L	2	105.00		210.00
310	366	101	SE	1	38.00	140	178.00
310	376	107	P	14	41.40	140	719.60
		1	ｻｰﾋﾞｽﾌﾟﾘﾝﾄ	55			2137.00
		101	SE	41	38.00		1558.00
		107	P	14	41.40		579.00
		2	ﾃﾅｷﾌﾟﾘﾝﾄ	2			210.00
		119	2L	2	105.00		210.00
		5	ﾈｶﾞｸﾞﾝｿﾞｳ	6			840.00
		401	DP35 12.20	6	140.00		840.00
		12	ﾀｼﾅｾｲﾋﾝ	17			1120.00
		305	KﾌﾟﾘﾝﾄBR1	16	60.00		960.00
		331	K.DP	1	160.00		160.00
						合計	4307

月日	受注No.	内訳 品名コード	品名	数量	単価	同P現	金額
						合計	

CEC555

事例－3　輸送業務管理

概　要

輸送業務を合理的に運用発展させることは資源，商品の輸送コスト，流通コストを引き下げるための重要な課題となっている．従って運用効率を高くすることが必要であり，これによって物資流通の担い手として社会的要請が大きい輸送業務の適切な安定供給が約束される．

このシステムは，輸送業務上発生する日々のデータを基本的な情報発生源とし，これらのデータを有効に活用することにより，日常の業務処理と，経営管理のための情報処理を適切におこなうことを主な目的としている．

また，輸送業界の特質にあわせ，それぞれの事業規模，事業内容に適したデータ処理ができるシステムであること，データ入力に際して各項目別のコード化，伝票類の転記等の繁雑な事前作業がなく，誰にでも自由に入力作業をすすめることができる入力装置を使用していること，業務の拡張によって給与計算，財務諸表の作成等まで処理することのできる拡張機能を有すること等を特長としている．

導入による効果

1．事務管理面における効果
- (1) 人件費の節減
- (2) 事務コストの節減
- (3) 事務処理時間の短縮
- (4) 社内事務手続，制度の標準化
- (5) 事務の合理化，情報化の推進

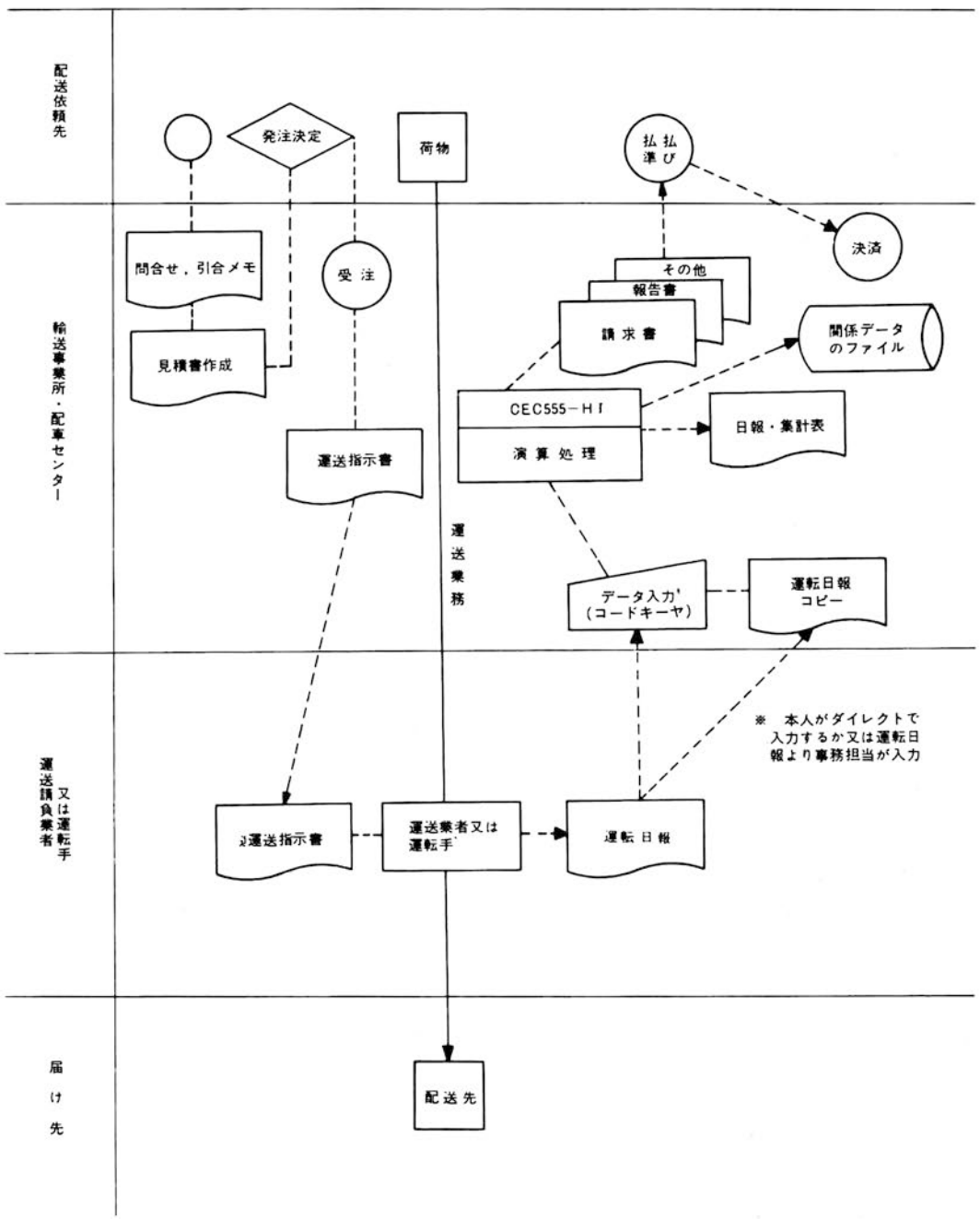
2．業務管理面における効果
- (1) 管理手続，ルールの標準化
- (2) 計数管理による最適操業
- (3) 統計による業務実体の正確な把握
- (4) 管理の計画化
- (5) 輸送コストの低減

3．経営上の効果
- (1) 経営判断，意志決定の迅速化，最適化
- (2) 経営実態の正確，迅速な把握
- (3) 原価，採算の正確な把握
- (4) 客観的基本データにもとづく経営方針決定
- (5) 経営近代化の促進
- (6) 情報化の促進

主な処理業務

1．運転手別稼動状況表
2．得意先別管理表
3．採算管理表
4．トラック別稼動状況表
5．トラック別収支状況表
6．トラック別原価状況表
7．品名別売上一覧表

業務処理のフローチャート



事例－4　商品管理

概　要

製品あるいは商品に関する種々の情報，例えば在庫状況，在庫量などを正確に把握するために電子計算機を利用し，庫入，問い合わせ，在庫移動，売上集計，返品処理，在庫販売予約，在庫一覧表の作成，棚卸伝票の発行等の業務を自動的におこなうもので，ＣＫシリーズマルチターミナルの使用により，どのような規模，形態の事業所，事務所でも，また大規模な事業所では中央計算センターにリンクし，一種の端末処理装置用としても運用することのできる機動力の高い管理システムである．
理システムである．

商品管理業務は，営業上の骨幹をなす要素のひとつであり，現状況を正確に迅速に把握するためにも，時々刻々，ランダムに発生する業務を遅滞，誤りもなく遂行する必要がある．

このため，このシステムの入力部では入力業務の省力化をはかるため，万能入力端末装置コードキーヤを使用している．

これは，ブック式に綴じられた紙製のシートの上に実名で処理項目名，商品名，部品名，取引先名などが印刷されており，それぞれの実名の語または語の集合の横を特殊電磁ペンにより軽くタッチするだけで該当項目のデータ入力が正確にはたせるものである．特殊ペンは普通の万年筆程度の太さであり，通常のペンを持つ要領でもち，指定項目を軽くタッチしながらテンキーから数値を入れてゆく方式で項目名商品名，部品名，取引先名はすべて実名で印刷されているから誰が使っても入力ミスはおこらないし能率も大幅に向上する方式のものである．各部所で発行された原始伝票より直接，簡単にデータが入力でき，従来のようなコード化やカードパンチ等の繁雑な前処理は一切不要である．従ってキーパンチャやオペレータなどの専門職の必要もなく誰にでもなく誰にでも操作でき，しかも実名の語との対応で入力されるので入力ミスの防止にも役立つ．

問い合わせ業務は，コードキーヤの該当項目と操作キーを押すだけでキャラクターディスプレイ上に在庫等の要求したデータが表示される．

特　長

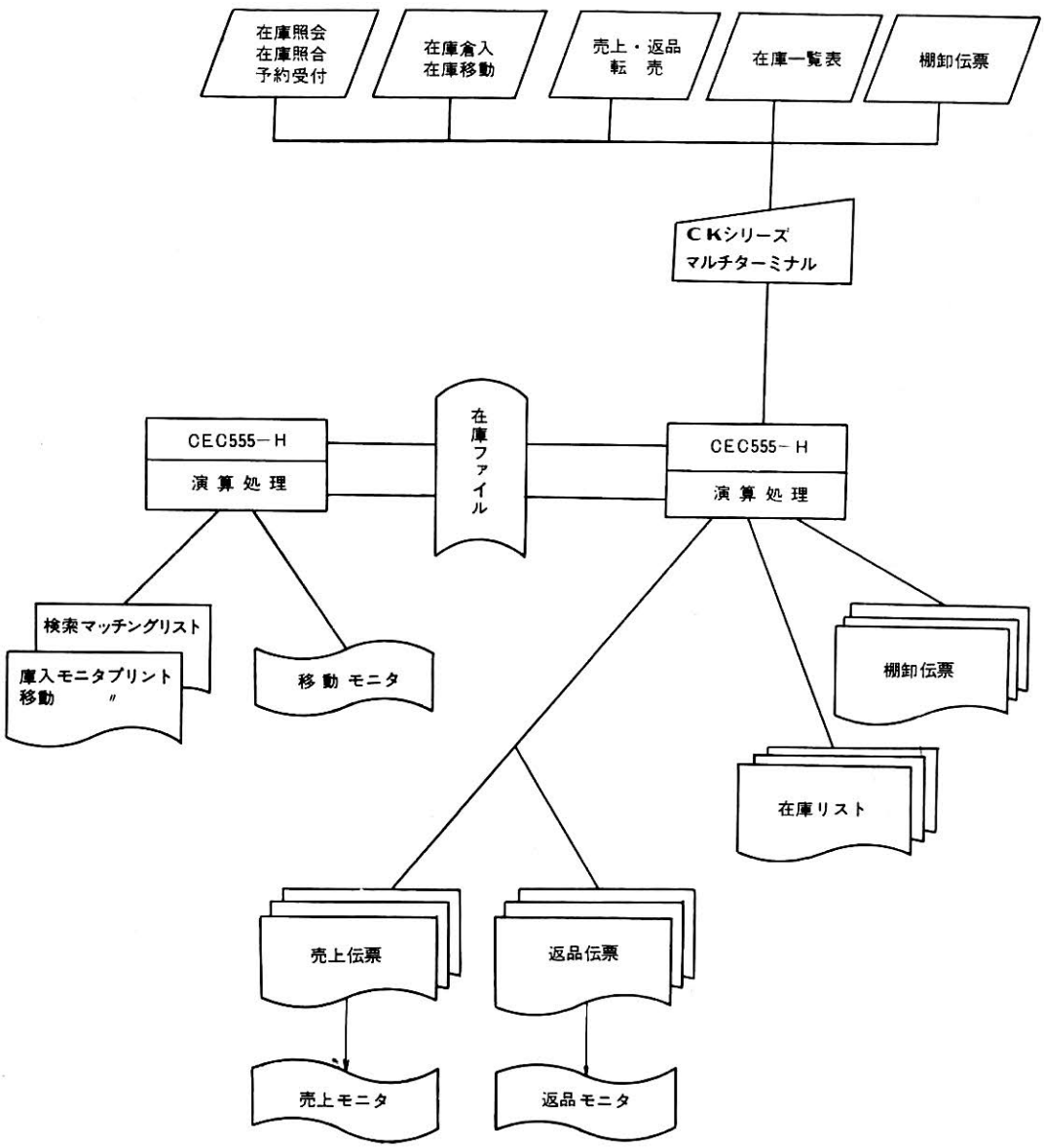
このシステムは工場で製品化し，入庫がおこなわれた時点よりその商品に関する在庫状況を正確に把握することができる．

1．本システムにはソートマージを始めとした効率のよいユーティリテイプログラムが用意されている．
2．商品管理用にあらたにHUT　ASK等の融通性に富んだ使いやすいアプリケーションプログラムが開発されておりシステムを容易に運用できる．
3．販売業務における情報の正確さと迅速さが保証される．
4．売上伝票（売上伝票４枚，納品書，請求書を含む）および返品伝票（赤伝）の発行が正確におこなえるために伝票上のミスによる浪費が皆無になる．
5．上記納品書，請求書等も同時に発行されるので販売管理上の省力化に役立つ．
6．同時に上位計算機への売上情報データも（紙テープ，または磁気カートリッジディスク等）得られるため，経理上の管理も円滑におこなえる．
7．随時在庫照会がおこなわれ，直ちに正確な照会結果が得られるので注文に対して敏速な処理がおこなえる．
8．在庫状況の正確な把握が充分な生産管理を保証する．
9．オペレーションが非常に簡単ですから計算機の専門知識はまったく必要とせず誰でも操作できる．

主な処理業務

1．庫入
2．問い合わせ業務
3．在庫移動
4．売上業務
5．返品業務
6．転売業務
7．在庫販売予約業務
8．在庫一覧表の作成
9．棚卸伝票の発行

標準処理フローチャート



事例－5　総合事務EDPシステム

このシステムは，社内における事務管理業務の自動化をはかるシステムで，データ入力装置にCKシリーズコードキーヤまたはCKシリーズマルチターミナルを利用することにより，営業部門，資材部門，財務部門，生産管理部門などすべての部門でのEDP化をスムーズに推進できるものである．プログラムは各部門ごとの処理業務がパッケージ化されており，重複処理のないスッキリしたEDP化への移行が実現できる．本章では，営業管理業務，資材管理業の一部について紹介されているが，この他に財務部門における給与計算，生産管理部門における工程管理業務の各種プログラムも用意されている．また，会社規模，または事業規模によってCECEDPシステム10～60シリーズより最適構成が選択できるので導入効率は抜群であり，社内，事業所の各部門を連係した総合的なEDPシステムを構成することができる．

営業管理部門

受注処理，受注実績処理，受注残高処理，出荷処理，入金回収処理，売上計画処理，売上実績処理，売掛残高処理など各業務用のプログラムがパッチージ化されている．入力データとしては，受注票，出荷票などのデータをCKシリーズコードキーヤにより入力し，CLP4100ラインプリンタによって工場注文書（製造指示書），納品書，請求書，回収実績表，売上計画表，売上実績表，受注実績表，受注残高一覧表，売掛残高一覧表，などを所定のフォーマットに従って印字する．

資材管理部門

在庫処理業務，入出庫処理業務，発注点処理業務，納入処理業務および未納処理業務などのプログラムがパッケージ化されている．入力データは注文伝票，入庫伝票，出庫伝票などのデータをCKシリーズコードキーヤより入力し，CLP－4100ラインプリンタによって注文書，未納日報，入出庫一覧表，在庫明細一覧表，取引先別納入一覧表などを所定のフォーマットに従って印字する．

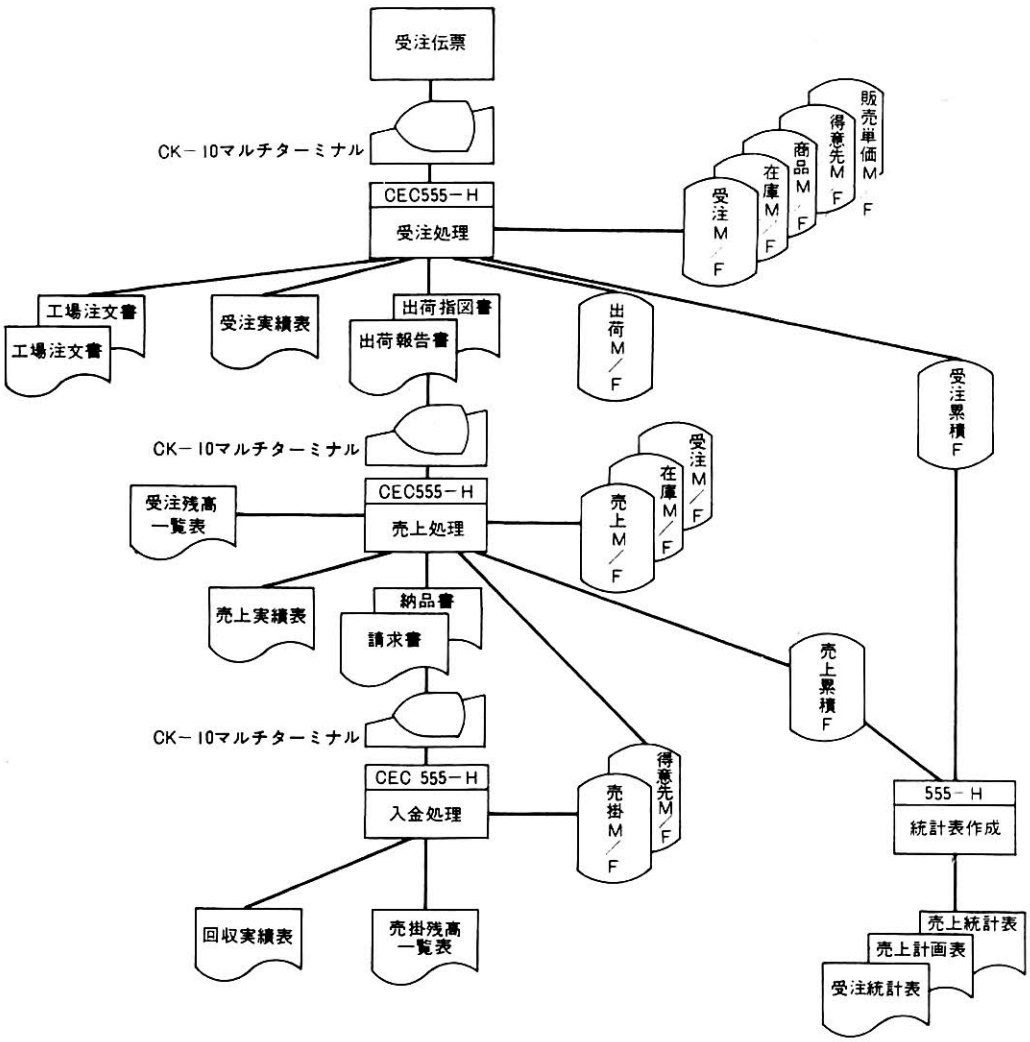
生産管理部門

生産日程の管理をおこなうもので，入力データとして，各所属における製造日程等をCKシリーズコードキーヤにより入力し，総合工程管理表，営業所別工程表，得意先別工程表，所属別工程表，外注先別工程表をCLP－4100ラインプリンタにより印字する．

財務部門

給与計算，財務諸表の作成．

《営業管理業務システムフローチャート》

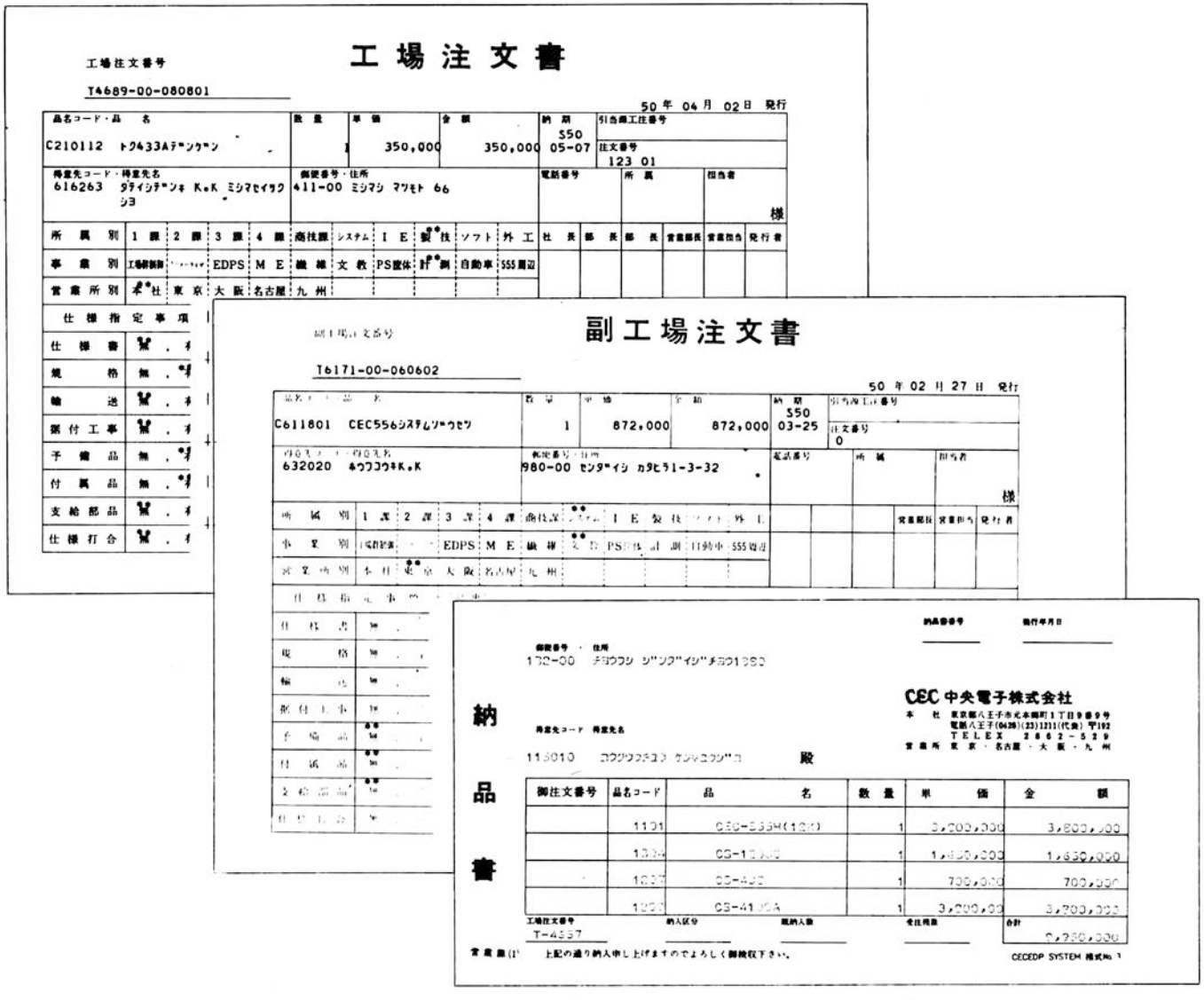


営業管理業務用ブック式シート

SHEET No.	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

	01	02	03	04	05	06	07	08
01	宇宙開発事業団	航空局	東京航空局	航空保安大学校	国際事業団	航空研	防衛庁調達本部	円意先地！
02	防衛庁技本第3研究所	航空自衛隊第3補給処	海上保安庁	機械技術研究所	鐵高研	電総研	大阪工業技術試験所	
03	日本原子力研究所	鉄研	日本専売公社	電々公社	東京データ通信局	武蔵野通研	茨城通研	
04	NHK	NHK総合技術研究所	電波研					
05	共同通信社	機械振興協会	高エネルギー物理学研究所	電力中研				
06	青山学院大	茨城高専	航空大学校	大阪市立大	大阪大学	学習院大学	京大	
07	九州大学	神戸大学	自治医科大学	信州大学	白石高専	静岡大学	成蹊大学	
08	防衛大学校	電通大	都立上科短大	東京高専	東京大学	都立大		
09	東京農大	東京農工大	東邦大	東洋大	独協医科大	長崎大	名古屋大	
10	日本大学	新潟大学	武蔵工大	明治大学	立命館大	早稲田大学		
11	宇部工高	戸畑工高	前橋工高	都城高校				
12	総合会津中央病院	石塚病院	国立大阪病院	鹿児島県成人病予防協会	群馬県成人病研究所	国立精神衛生研究所	鶴見病院	
13	日本成人病予防会	PL大学健康管理センタ	PL東京健康管理センタ	ヘルツ健康管理センタ	宝迫医院	みどり健康管理センタ	宮崎県健康増進センタ	
14	生駒保健所	若林医院						
15	アクト技術	アマダ	石川島播磨	岩通本社	岩通八王子	大倉電気	沖電気研究所	
16	沖電気八王子	沖電気芝浦	沖電気品川	沖電気八王子	大阪変圧器	神ユニバック	オニバック	
17	オリンパス光学工業	電場工業	川島織物	キーノン	久保田鉄工	国際電気	小西六写真工業	
18	三栄測器	三洋電機	三洋電気中研	三洋電気開発所	産業輸送開発	塩野義製薬	シャープ本社	
19	シャープ奈良	シャープ天理	シャープ矢板	新日鉄	新日鉄八幡	新日鉄大分	新日鉄室蘭	
20	新日本電気	新日本電気上野	沖信光	住友重機工業	住電大阪	住電機浜	スガー精密	
21	住友ゴム工業	ソニーテクトロ	ソニー厚木	ソニー中研	大東製機			
22	立石草津	立石中研	立石三島	立石四条	大同マルタ	高岳製作所	田原電気	
23	ティアック特機	中央電機SS	椿本チェーン	帝王通信	テルエンジニア	テレニクス		
24	ワ技研	東海ゴム工業	東京計器	東京航空計器	東芝日野	東芝綱町	東芝府中	
25	東芝総研	東芝青梅	東芝堀川	東芝音響	東芝玉川	東芝西谷	東京精密	
26	（下請）東京電子工業	東京電波	東京特殊電線	東京電子工業	東英工業	東邦電子	東北令属東京	
27	東洋通信機	鳥取三洋	東海理化電機	戸上電機SS	東通エンジニ	東洋鋼鉄	東レ滋賀	
28	豊田工機							
29	日本アビオトロ		日本工営	日本航空電子	日本電気本店	日本電気府中	日本電気機元	
30	日本電気玉川	日本電気相模原	日本電気山梨	日本電気精器	日本電子機器	日本トータ	日本ビクタ八王子	
31	日本飛行機	日本分光	日本無線	日本ギュレータ	日電機工	日産自動車	日産車体	
32	日新電気		浜松テレビ	パタトロン				

CEC CK-6,CK-8 SHEET



工場注文書

工場注文番号

T4689-00-080801

50年04月02日 発行

品名コード・品名	数量	単価	金額	納期	引当源工注番号
C210112 トク433Aデンケン	1	350,000	350,000	S50 05-07	注文番号 123 01

得意先コード・得意先名	郵便番号・住所	電話番号	所属	担当者
616263 ダイシデンキ K.K ミシマセイサクショ	411-00 ミシマシ マツモト 66			様

所属別	1課	2課	3課	4課	商技課	システム	IE	製技	ソフト	外工	社長	部長	部長	営業部長	営業担当	発行者
事業別	工場制御	[illegible]	EDPS	ME	繊維	文教	PS筐体	計測	自動車	SSS周辺						
営業所別	本社	東京	大阪	名古屋	九州											

仕様指定事項	
仕様書	無
規格	無
輸送	無
据付工事	無
予備品	無
付属品	無
支給部品	無
仕様打合	無

副工場注文書

副工場注文番号

T6171-00-060602

50年02月27日 発行

品名コード・品名	数量	単価	金額	納期	引当源工注番号
C611801 CEC556システムゾウセツ	1	872,000	872,000	S50 03-25	注文番号 0

得意先コード・得意先名	郵便番号・住所	電話番号	所属	担当者
632020 ホウフコウキK.K	980-00 センダイシ カタヒラ1-3-32			様

所属別	1課	2課	3課	4課	商技課	システム	IE	製技	ソフト	外工	営業部長	営業担当	発行者
事業別	工場制御	[illegible]	EDPS	ME	繊維	文教	PS筐体	計測	自動車	SSS周辺			
営業所別	本社	東京	大阪	名古屋	九州								

仕様指定事項	
仕様書	無
規格	無
輸送	無
据付工事	無
予備品	無
付属品	無
支給部品	無
仕様打合	無

納品書

納品番号　　発行年月日

郵便番号・住所
132-00 チヨウフシ ジンダイシ チヨウ1980

CEC 中央電子株式会社
本社 東京都八王子市元本郷町1丁目9番9号
電話八王子(0426)(23)1211(代表) 〒192
TELEX 2862-529
営業所 東京・名古屋・大阪・九州

得意先コード 得意先名
115010 コウクウフチヨウ ケンキユウジヨ 殿

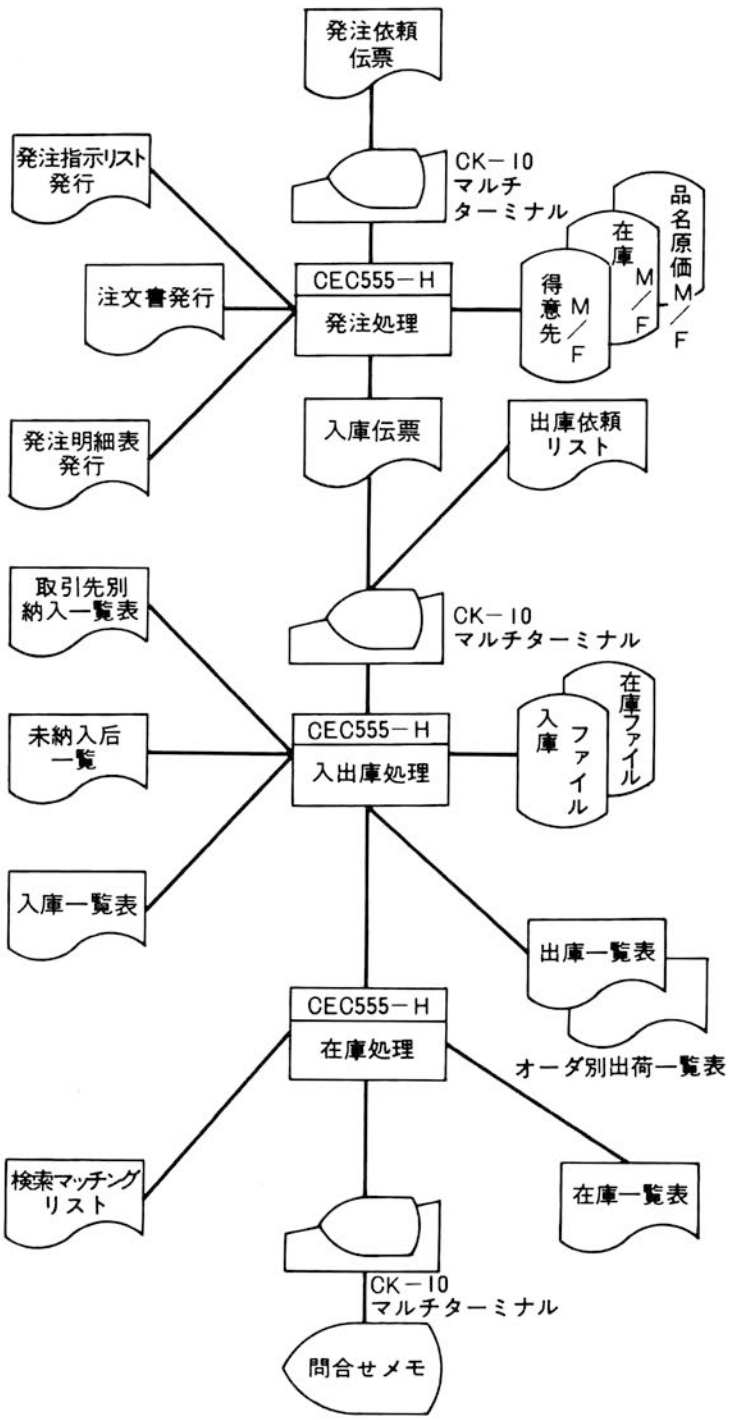
御注文番号	品名コード	品名	数量	単価	金額
	1101	CEC-555H(12K)	1	3,200,000	3,800,000
	1304	CS-1[illegible]	1	1,650,000	1,650,000
	1207	CS-4[illegible]	1	700,000	700,000
	1200	CS-41[illegible]A	1	3,200,000	3,800,000

工場注文番号	納入区分	既納入数	受注残数	合計
T-4557				9,950,000

営業課(1) 上記の通り納入申し上げますのでよろしく御検収下さい。

CECEDP SYSTEM 様式No. 3

《資材管理業務システムフローチャート》



資材管理用ブック式シート例

SHEET No.	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

	01	02	03	04	05	06	07	08
01	SN7400N	SN7473N	SN74151N	8008	SN74S11N	SN74LS181N	MCT2	部品問い合せ
02	SN7401N	SN7474N	SN74154N	8008-1	SN74S112N	SN74LS191N	MCA231	注　文
03	SN7402N	SN7475N	SN74155N	8080	SN74S124N	SN74LS194N	PS2201	工注 A
04	SN7403N	SN7476N	SN74157N	1702A	SN74S133N	SN74LS295N	MCS2	常備品 ST
05	SN7404N	SN7480N	SN74160N	IM5600	SN74S138N		MAN6A	S
06	SN7405N	SN7481N	SN74161N	MPD404D	SN74S158N		MAN66A	T
07	SN7406N	SN7482N	SN74162N	TR1602A	SN74S194N	SL1112	TLR301	担当者
08	SN7407N	SN7483N	SN74163N	AY-5-1012	SN74S257N		MV5024	取引先
09	SN7408N	SN7484N	SN74164N	SN75451P	SN74LS00N		SL103	部品名
10	SN7409N	SN7485N	SN74167N	SN75452P	SN74LS02N		OLD3101	数　量
11	SN7410N	SN7486N	SN74170N	SN75453P	SN74LS04N		OLD4101	納　期
12	SN7412N	SN7489N	SN74172N	DM8820	SN74LS08N		OLD7101	継　続
13	SN7413N	SN7490N	SN74173N	DM8830	SN74LS10N		DB1	入　庫
14	SN7414N	SN7491N	SN74174N	DM8831	SN74LS11N		DB2	出　庫
15	SN7420N	SN7492N	SN74175N	SN74H00N	SN74LS14N		CRB1	購入先一覧表
16	SN7425N	SN7493N	SN74180N	SN74H01	SN74LS20N		CRB2	在庫管理
17	SN7427N	SN7494N	SN74181N	SN74H02	SN74LS27N		RB3	①
18	SN7430N	SN7495N	SN74182N	SN74H04	SN74LS28N		PR102	
19	SN7432N	SN7496N	SN74184N	SN74H10	SN74LS30N		T181-331	未納管理
20	SN7437N	SN7497N	SN74185N	SN74H20N	SN74LS32N		RA9	
21	SN7438N	SN74107N	SN74190N	SN74H30N	SN74LS37N		RA2	
22	SN7440N	SN74110N	SN74191N	SN74H40N	SN74LS38N		PR332	
23	SN7442N	SN74111N	SN74192N	SN74H50N	SN74LS42N		PR103	
24	SN7445N	SN74120N	SN74193N	SN74H54N	SN74LS74N		SSD1000-0041	
25	SN7447N	SN74123N	SN74194N	SN74H55N	SN74LS86N		SSD0132-0040	
26	SN7448N	SN74125N	SN74195N	SN74H60N	SN74LS109N		BDS-40832-200	
27	SN7450N	SN74132N	SN74196N	SN74H62N	SN74LS153N		BDS-40832-PS1	
28	SN7451N	SN74141N	SN74197N	SN74H74N	SN74LS157N		SR127	
29	SN7453N	SN74142N	SN74198N	SN74H101N	SN74LS164N		SR132	
30	SN7454N	SN74145N	SN74283N	SN74H183N	SN74LS170N			
31	SN7470N	SN74148N	SN74284N	SN74S00N	SN74LS174N			
32	SN7472N	SN74150N	SN74285N	SN74S04N	SN74LS175N			

CEC CK-6,CK-8 SHEET

在庫明細一覧表

ｻﾞｲｺ ﾒｲｻｲ ｲﾁﾗﾝﾋｮｳ

S50-07-10

ﾌﾞﾋﾝｺｰﾄﾞ	ﾌﾞﾋﾝﾒｲ	ｻﾞｲｺｽｳ	ﾀﾝｶ	ｷﾝｶﾞｸ	ﾐﾆﾏﾑｻﾞｲｺ	ｱﾗｰﾑ
A010010	7400PC	156	48.00	7,488.00	500	***
A010020	SN7401N	1,843	57.00	105,051.00	200	
A010030	7402PC	265	48.00	12,720.00	200	
A010040	SN7403N	610	57.00	34,770.00	200	
A010050	7404PC	1,061	55.00	58,355.00	500	
A010060	SN7405N	178	67.00	11,926.00	200	***
A010070	SN7406N	185	251.00	46,435.00	100	
A010080	SN7407N	70	251.00	17,570.00	100	***
A010090	SN7408N	30	80.00	2,400.00	0	
A010100	SN7409N	0	80.00	0.00	50	***
A010110	7410PC	184	48.00	8,832.00	500	***
A010120	SN7412N	155	57.00	8,835.00	0	
A010130	SN7413N	217	172.00	37,324.00	50	
A010140	SN7414N	209	456.00	95,304.00	50	
A010150	SN7416N	0	193.00	0.00	0	
A010160	SN7417N	0	193.00	0.00	0	
A010170	7420PC	211	48.00	10,128.00	300	***
A010180	SN7423N	0	107.00	0.00	0	
A010190	SN7425N	117	89.00	10,413.00	100	
A010200	SN7426N	0	85.00	0.00	0	
A010210	SN7427N	0	89.00	0.00	0	
A010220	7430PC	27	48.00	1,296.00	400	***
A010230	SN7432N	3	89.00	267.00	50	***
A010240	SN7433N	0	106.00	0.00	0	
A010250	SN7437N	966	96.00	92,736.00	300	
A010260	SN7438N	300	96.00	28,800.00	300	
A010270	SN7440N	740	57.00	42,180.00	300	
A010280	SN7442N	123	255.00	31,365.00	100	
A010281	SN7441N	0	450.00	0.00	0	
A010290	SN7443N	0	350.00	0.00	0	
A010300	SN7444N	0	396.00	0.00	0	
A010310	SN7445N	0	460.00	0.00	0	
A010320	SN7446N	0	326.00	0.00	0	
A010330	SN7447N	332	305.00	101,260.00	100	
A010340	SN7448N	0	553.00	0.00	0	
A010350	SN7450N	1,355	57.00	77,235.00	0	
A010360	SN7451N	0	57.00	0.00	100	***
A010370	SN7453N	445	57.00	25,365.00	100	
A010380	SN7454N	228	57.00	12,996.00	100	
A010390	SN7470N	328	72.00	23,616.00	0	
A010400	SN7472N	263	82.00	21,566.00	0	
A010410	SN7473N	120	118.00	14,160.00	100	
A010420	7474PC	1,001	95.00	95,095.00	500	
A010430	SN7475N	238	144.00	34,272.00	100	
A010440	SN7476N	0	144.00	0.00	0	
A010450	SN7480N	56	140.00	7,840.00	0	
A010460	SN7481N	0	255.00	0.00	0	
A010470	SN7482N	23	263.00	6,049.00	0	
A010480	SN7483N	641	290.00	185,890.00	50	
A010490	SN7484N	283	450.00	127,350.00	0	
A010500	SN7485N	6	528.00	3,168.00	30	***
A010510	SN7486N	8	116.00	928.00	50	***
		ﾍﾟｰｼﾞ ﾄｰﾀﾙ		1,600,985.00		

《土木測量計算システム》

概　要

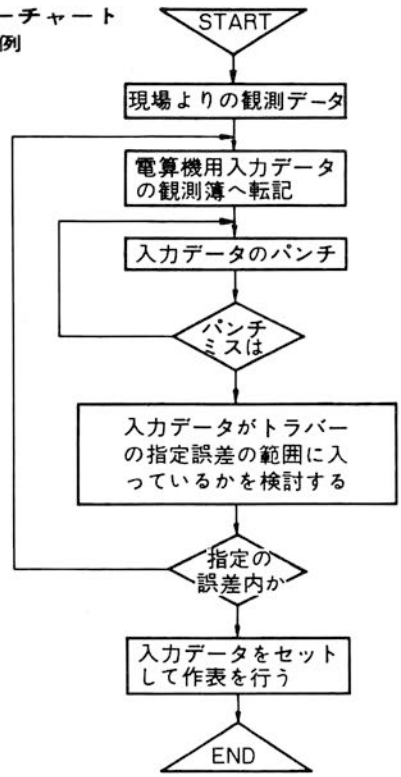
本システムはＣＥＣ５５５電子計算機を中央演算処理装置として各種の業務の計算処理，および帳票の作成を自動的におこなうもので，従来，多くの日数を必要とした計算業務を大幅に時間短縮し，精度の飛躍的向上が期待できる．

システムはソフトウェアにより運営されるので，プログラムによって多角測量座標計算，縦横線および高低計算，面積計算，剛結構造計算，配水管網計算などあらゆる業務の計算処理をおこなうことができる．

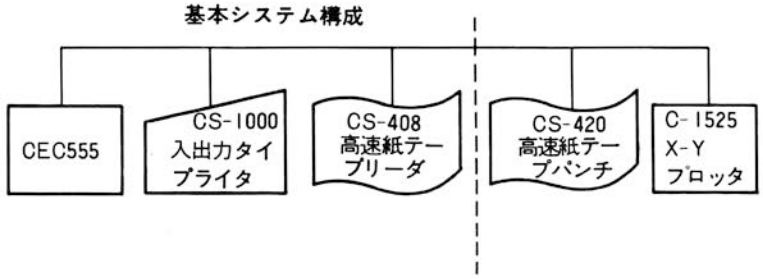
また間接業務である給与計算，見積り計算，伝票整理計算，原価計算等の計算処理をおこなうこともできる．

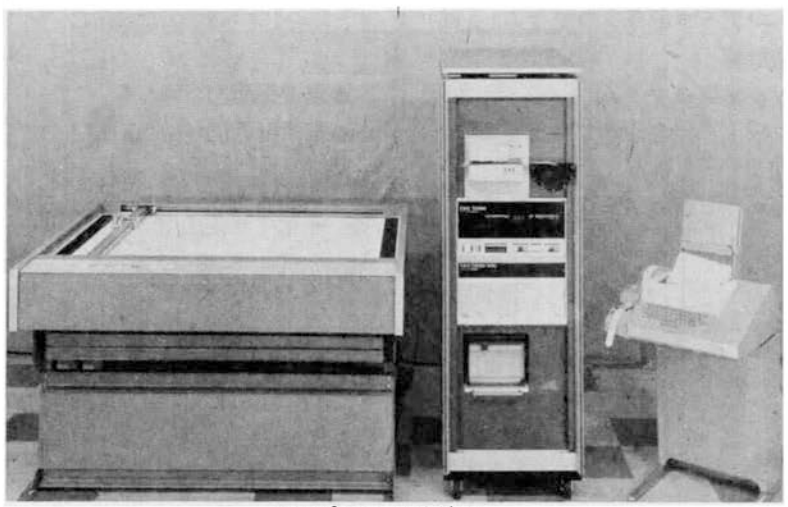
さらに，Ｘ—Ｙプロッタによる自動作図，カセット磁気テープ装置，あるいはデータ通信網を利用した遠隔地との情報自動処理も容易である．

フローチャートの一例



基本システム構成





ＸＹプロッタ接続例

土木測量計算プログラム

- ●多角測量座標計算(閉合、結合)
- ●多角測量座標計算(開放)
- ●多角測量座標計算(国土調査用、地籍用)
- ●細部多角測量座標計算(開放)
- ●多角網平均計算(X－Y型、H－A－O型)
- ●縦横線および高低計算(オフセット付)
- ●水平距離および高低差出力計算
- ●座標差による距離と方向角の計算
- ●基準点成果計算(含む標高)
- ●トラバー逆計算
- ●水準測量計算(閉合)
- ●交点座標計算(直線と直線)
- ●交点座標計算(直線と曲線)
- ●交点座標計算(クロイソドと直線)
- ●交点座標計算(クロソイドと曲線)
- ●交点座標計算(クロイソドとクロソイド)
- ●曲線設置計算(単曲線)
- ●クロソイド計算
- ●ヘロンの公式
- ●面積計算(経緯距法)
- ●三斜底辺、高さ面積計算
- ●定面積調整計算
- ●土積計算
- ●切取、盛土床堀面積計算
- ●切取、盛土床堀数量計算
- ●盛土流量計画計算
- ●層厚調整計算
- ●道路縦断計算
- ●道路法面積計算
  取付道路計算
- ●盛土法面仕上計算
- ●街区計算
- ●二個三角形平均計算
- ●スタジア測量計算
- ●境界測量計算
- ●空中三角測量計算
- ●新・旧(旧・新)座標変換
- ●剛結構造計算
- ●ラーメン計算
- ●配水管網計算
- ●流量速計算
- ●流量計算(マニングの公式)
- ●流量計算(合理式、断面決定)
- ●ボックス・カルバート応力計算
- ●橋脚設計計算
- ●橋台設計計算
- ●横断暗渠工計算

座標読取装置用プログラム

- ●実座標出力計算
- ●デイジフラニ・データ編集プログラム(各種)
- ●定面積調整計算
- ●土地改良土積計算
- ●三斜法面積計算

X－Yプロッタ用プログラム

- ●図根点プロットと点名印字プログラム
- ●道路横断図
- ●現況平面図(用地図)
- ●都市計画図(宅地造成図)

※ここに記載のプログラムは主要のもののみであり、この他約200点におよぶ各種プログラムが用意されている.

多角測量座標計算（トラベー計算）

地積用　閉合、結合型

多角測量座標計算簿

路線名 U7U7　12	次数	精度区分	方向角の閉合差 0 22	n = 11	公差 5 29	合・否
測　点 *A　から　C	1	CTU 1	座標の閉合差 111252	水平距離の和 1044.750	公差 1.150	合・否

測点	方向点	夾角	補正量	補正方向角	水平距離	真数 cos	真数 sin	⊿X	⊿Y	補正量 εx	補正量 εy	X	Y	座標点
*B	*A			121 9 47								-142481.770	-10082.930	*A
*A	1	45 5 8	-2	346 14 53	44.390	.971334	.237719	43.118	-10.552	-4	-3	-142438.656	-10093.485	1
1	2	139 59 42	-2	306 14 33	111.080	.591204	.806522	65.671	-89.588	-10	-6	-142372.995	-10183.079	2
2	3	214 27 40	-2	340 42 11	128.250	.943819	.330464	121.045	-42.382	-11	-8	-142251.961	-10225.469	3
3	4	187 39 42	-2	348 21 51	106.040	.979449	.201691	103.861	-21.397	-9	-7	-142148.109	-10246.863	4
4	5	191 29 55	-2	359 51 44	121.900	.999997	.002405	121.900	-.293	-11	-7	-142026.220	-10247.163	5
5	6	165 32 38	-2	345 24 20	106.940	.967734	.251976	103.499	-26.946	-9	-7	-141922.740	-10274.116	6
6	7	161 46 42	-2	327 11 0	106.750	.840409	.541953	89.714	-57.853	-9	-6	-141833.035	-10331.975	7
7	8	172 55 32	-2	320 6 30	122.950	.767258	.641338	94.334	-78.853	-11	-8	-141738.712	-10410.836	8
8	9	193 33 40	-2	333 40 8	75.010	.896246	.443558	67.227	-33.271	-6	-4	-141671.491	-10444.111	9
9	C	180 0 2	-2	333 40 8	121.440	.896246	.443558	108.840	-53.866	-11	-8	-141562.662	-10497.985	C
C	D	188 50 40	-2	342 30 46										

観測方向角	342 31 8		342 30 46	1044.750	⊿X or ⊿Y	919.199	-414.991	-91	-64	-141562.662	-10497.985	C
既知方向角	342 30 46				DX or DY	919.108	-415.055					
夾角の補正量	0 22				閉合差	.091	.064			-142481.770	-10082.930	*A
配付点数	11				配布点数	10		既知の座標差		DX 919.108	DY -415.055	
平均補正量	-2.000				補正量	-.000087	-.000061	精度		1/ 9391		

計画機関　　実施機関　　（計算機中央電子株式会社　CEC555）

❹医療情報システム

《CEC552D心電・心音自動診断システム》

概　要

本システムは，集団検診のスクリーニングから精密検診データの解析まで幅広く活用できる心電図自動診断システムである．

成人安静時心電図，負荷心電図，小児心電図，心音等をCEC 555Hミニコンピュータのプログラム処理により自動的に解析診断し，診断結果，および正常，異常の有無，さらに異常の場合の程度を自動的に印字する．

また，プログラムにより精密解析データの印字も自動的におこなえる．

移動検診僻地検診は，カセット式心電図記録装置の併用と電話回線などの利用により，能率よくおこなえ，心電図検診の広域普及化と老人，成人，学生，学童にいたる各階層の心電図検診を正確に，能率よくすすめることができる心電図自動診断システムである．

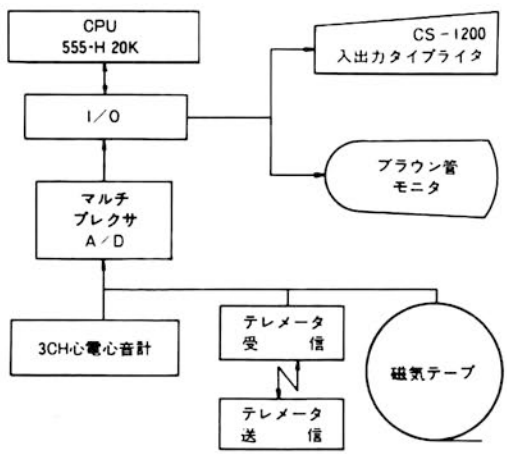
特　長

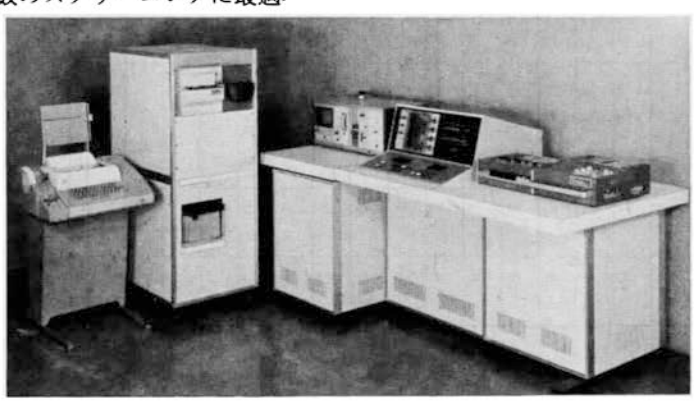
1．テレメータで負荷ロード中に心電異常が発見された場合，アラーム信号を発生し，ただちに負荷を中止する機能がある．
2．８Ｋ語の主記憶装置の増設で２人の測定が可能である．
3．電話回線を利用して検診の広域普及ができる．
4．診断結果の印字は，ミネソタコード以外に正常，異常および異常の程度が印字されるので，多人数のスクリーニングに最適．

診断項目

1．成人安静時心電図のミネソタコードによる印字
2．診断名の印字
3．精密解析データの印字
4．ミネソタコードによる負荷心電図の診断印字
5．負荷心電図の精密解析データの印字
6．小児心電図解析
7．精密データの解析印字
8．心音の解析

システム構成





診断出力印字例

CEC-CARDIOLYZER

SERIAL NO: 12-34-56

CASE NO: 750714

NAME: AGE: SEX:

HEART RATE : 53

MINNESOTA CODE

VENTRICULAR CONDUCTION DEFECT
7-5 : V1

JUDGEMENT ; ABNORMAL1 ECG

	1	2	3	AVR	AVL	AVF	V1	V2	V3	V4	V5	V6
PA	10	10	0	-12	0	7	11	9	12	11	9	12
PD	64	64	0	64	0	46	64	64	64	64	64	64
P'A	-	-	-	-	-	-	-	-	-	-	-	-
P'D	-	-	-	-	-	-	-	-	-	-	-	-
PQ-S	0	1	0	-1	0	0	1	1	1	1	2	1
QA	0	0	0	-78	0	0	0	0	0	0	0	0
QD	0	0	0	40	0	0	0	0	0	0	0	0
RA	67	72	28	14	19	60	52	40	53	51	39	54
RD	48	52	36	20	44	40	28	28	28	28	28	28
SA	0	0	-23	0	0	-23	-121	-92	-126	-121	-93	-125
SD	0	0	24	0	0	24	36	44	32	36	44	32
R'A	0	0	0	0	0	0	10	0	11	11	0	12
R'D	0	0	0	0	0	0	20	0	16	20	0	16
S'A	-	-	-	-	-	-	-	-	-	-	-	-
S'D	-	-	-	-	-	-	-	-	-	-	-	-
ST-J	1	1	-1	-1	1	-1	-1	-4	-2	-1	-5	-2
ST1	0	0	-1	0	1	0	-1	-2	-1	-1	-4	-1
ST2	0	0	-1	-1	0	0	2	-1	2	2	-3	2
TA	21	36	19	-31	0	29	97	95	100	97	95	100
T'A	-	-	-	-	-	-	-	-	-	-	-	-
UA	-	-	-	-	-	-	-	-	-	-	-	-
PP	1088	1088	0	1080	0	1088	1080	1088	1080	1084	1092	1088
RR	1116	1116	1116	1112	1112	1112	1116	1116	1116	1116	1116	1116
PR	156	152	0	152	0	152	152	152	152	156	152	156
VAT	32	32	24	60	28	28	24	24	20	24	24	20
QRS	48	52	60	60	44	64	84	72	76	84	72	76
QT	416	440	424	432	0	428	424	492	424	424	492	424
IC	1	1	1	1	1	1	1	1	1	1	1	1

CEC 552 心電図自動診断装置

CEC552 心電図自動診断装置は，日本循環器管理研究協議会，世界保健機構(WHO)で採用されているミネソタコードを診断基準に，測定された心電図を自動的に解析診断し，結果をコード印字する装置である．カルテは印字されたコードによって簡単に集計・分類ができるので国家的規模の健康管理の資料として保健行政機関・循環器専門医師・その他の研究機関で有意義に活用できる．また小型車で運搬・移動することができますのであらゆる場所・地域での運用が可能で，広く国民全体の健康管理に貢献することができる．

診断印字例

・・・1－0－0	ミネソタコード
・・・・・058	脈　拍
・・・00003	番　号
50・07・14	検査年月日

■速い処理時間

測定開始後1人／約1分で完了

■保守への配慮が完璧

CEC 標準化ＩＣパッケージの使用により故障が少なく，保守，点検も簡単．

■二様式の操作が可能

AUTO, MANUALいずれの操作も可能．

■操作は簡単

スタートボタンを押すだけで自動的に測定し診断結果をタイプ　字する．操作は，3日間（正味15時間程度）の講習で簡単に習得できる．

■設置はどこでも

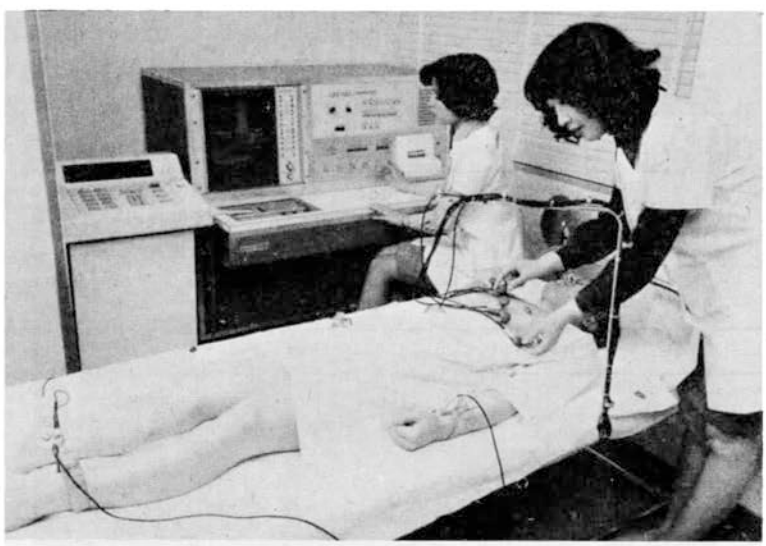
10m²(約3坪)程度の部屋で設置・運用が可能

■巡回検診が可能

小型車に搭載して移動・出張検診が手軽にできる．

■データの分類・統計が容易

コードは個人別カードに自動印字されますので集計・整理・統計・分類が簡単．



CEC552Sカセット式心電図記録装置

携帯形の心電図記録装置で，移動検診時にはカセットテープに心電図を集録し，中央の診断システムで診断をおこなう．巡回検診，移動検診，僻地医療には大変便利な装置であり，心電図検診の広域普及に効力を発揮する．

特　長

1．小型・軽量で持ち運びが容易．
2．CEC552心電図自動診断装置，CEC552D心電・心音自動診断装置との接続が容易で，簡単にデータ入力ができる．
3．移動検診時，または遠隔地，山間僻地の検診に便利．
4．カセットの差しかえにより，多人数のデータ集録，分類が簡単
5．信頼性が高く，長期の運用が可能
6．ノイズの影響が少く，忠実に心電図波形を再生．

※　CEC552Sカセット式心電図記録装置は，次の3種が用意されている．
552S　1チャネル用
552D　3チャネル用
552D-3P心電・心音用

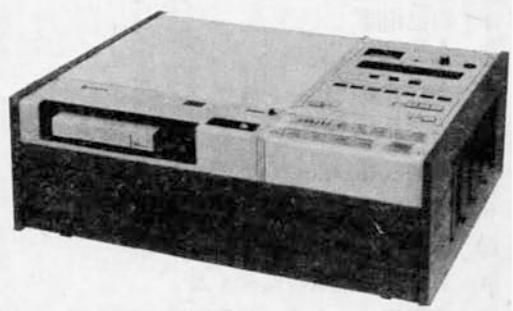
CEC540Sカセット式脳波記録装置（13CH）

脳波，または脳波集団検診時におけるデータの記録を主目的に設計された装置で，集団検診時におけるデータの整理や測定状態のチェックが容易におこなえる．また各種のアナログデータの記録にも利用することができる．

特　長

1．医療用として，脳波データ，心電図データのほか，筋電図（オプション）動脈圧の記録，眼球運動の記録にも利用できる．
2．医療用のみならず，地震，振動，制御信号などあらゆるアナログデータの記録にも利用できる．
3．電子計算機またはその他の測定器による制御が容易．
4．記録時に被検者番号，較正電圧などが同時に記録できる．
5．SERCH機構により記録された位置が容易に検出できる．
6．音声チャネルにより，データ測定状態が記録できる
7．カセット式のため操作，および保存が便利





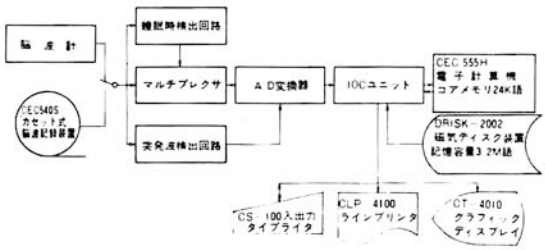
《CEC540脳波自動診断装置》

臨床脳波の解析をCEC 555H電子計算機によるプログラム処理とスパイクウェーブ検出回路などにより，自動診断をおこなうシステムである．

脳波の解析は基礎律動，突発波の解析を新開発の特殊回路により併行しておこない，このデータについてあらかじめ設定されている診断基準により正常，異常の判定と，異常者に対する精密検診データをタイプライタに印字する．また，判定結果は印字と併行してディスプレイパネル上にも描画される．また一定時間内に大量のデータ処理がおこなえるので臨床検査や脳波研究に便利に利用できる．

システムは13ch脳波計CEC 555H電子計算機，磁気ディスク装置，ディスプレイおよびスパイクウェーブ検出回路，論理診断回路，マルチプレクサ，A—D変換器などの主要回路が同一筐体内にコンパクトにまとめられており，脳波臨床検査，集団検診，移動検診が容易に能率的におこなえる．





〈異常波〉

1．スパイク波（個数，平均周期，振幅，パーセントタイム，連続性）
2．スパイクおよびウェーブ（個数，平均周期，振幅，パーセントタイム，連続性）
3．シャープウェーブ（個数，平均周期，振幅，パーセントタイム，連続性）
4．シャープおよびウェーブ（個数，平均周期，振幅，パーセントタイム，連続性）
5．バーストウェーブ（各帯域の個数，振幅，平均振幅，パーセントタイム，連続性）
6．アーチファクト検出，パーセントタイム，その他
7．トータル，平均振幅
8．睡眠，検出
9．臨床診断用プログラム
10．パワースペクトル＞周期
　　自己相関
　　相互相関
　　（FETを用いたもの，
　　高速フーリエ変換を用いたもの）

〈ウォルター型分析プログラム〉
〈眼球運動の解析プログラム〉
〈その他〉 心電図，心音図，呼吸の測定分析

データ処理の内容

1．刺激反応時間分布
2．潜伏時間分布
3．繰返し時間分布
4．自己相関
5．相互相関
6．眼球運動の積分，他
7．周期分布視察法による（藤森法改良型）
8．周期振幅分布
9．振幅と周波数帯域による分布
10．周波数帯域による平均振幅分布
11．各周波数帯域の%タイム
12．高電圧シータ波の個数振幅，%タイム，周期の連続性
13．高電圧α波の個数振幅，%タイム，周期の連続性，振幅の連続性

DISTRIBUTION OF AMPLITUDE AND FREQUENCY　　印字例（精神神経学雑誌より転載）

	DELTA .5-3.5	THETA2 4.-5.5	THETA1 6.-7.5	ALPHA1 8.-8.5	ALPHA2 9-11	ALPHA3 11.5-12.5	BETA1 13-17.5	BETA2 18-30
00 - 10		5	14	3	22	11	10	4
10 - 20	3	14	20	21	76	15	9	4
20 - 30	1	6	6	13	66	17		
30 - 40	1		1	17	63	2	1	
40 - 50				9	53	3		
50 - 60			1	8	25	1	1	
60 - 70				1	4			
70 - 80					3			
80 - 90								
90 -100								
100 -								
MEAN	22	14	14	29	29	19	13	10
TOTAL N	5	25	42	72	312	49	21	8
% TIME	2.5	7.8	9.4	13.2	49.5	6.4	2.3	0.6
TOTAL T	1.6	4.9	5.9	8.3	31.0	4.0	1.4	0.3

THETA WAVE > 30MMV

TOTAL N	:	2
% TIME	:	0.4

AMPLITUDE

MAXIMUM	:	52
MEAN	:	44

CONTINUITY

MAXIMUM	:	0
MEAN	:	0
BLOCK N	:	0
TOTAL N	:	0

ALPHA WAVE REGULARITY

MAXIMUM	:	39
MINIMUM	:	10
MEAN	:	20
TOTAL N	:	204
GROUP N	:	10
% TIME	:	32.8
TOTAL TIME	:	20.5

CONTINUITY OF FREQUENCY BAND

	DELTA 0.5-3.5	THETA 4.0-7.5	ALPHA 8.0-12.5	BETA(1) 13.0-17.5	BETA(2) 18.0-30.0
MAXIMUM N	0	4	80	0	0
MEAN N	0	2	14	0	0
BLOCK N	0	10	26	0	0
TOTAL N	0	26	384	0	0
% TIME	0.0	6.9	61.3	0.0	0.0
TOTAL TIME	0.0	4.3	38.4	0.0	0.0

上段は図7の脳波の周波数帯域と振幅，%time との関係を示し，下段は各帯域の波の持続性とその %time を示す．周波数帯域は，実際には 0.5≦DELTA＜4.0，4.0≦THETA 2＜6.0，6.0≦THETA 1＜8.0，8.0≦ALPHA 1＜9.0，9.0≦ALPHA 2＜11.5，11.5≦ALPHA 3＜13.0，13.0≦BETA 1＜18.0，18.0≦BETA 2＜30 を示している

ALPHA WAVE REGULARITY

NO	MAXIMAM	MINIMAM	DURATION MEAN	DURATION TOTAL	DURATION IREGULAR	AMPLITUDE REGULAR	AMPLITUDE IREGULAR	AMPLITUDE CONSTANT
1	10.9	8.2	9.5	12	6	+	-	-
2	11.2	8.7	9.8	12	3	+	-	-
3	12.8	8.5	9.6	10	3	-	-	-
4	12.0	8.8	10.0	13	2	-	-	-
5	12.9	8.4	10.0	39	11	-	4	23
6	11.4	8.5	10.0	10	3	-	1	3
7	11.6	8.8	9.9	27	3	-	2	17
8	12.8	8.5	9.9	35	12	-	1	29
9	12.5	8.0	9.9	22	7	-	1	15
10	12.2	8.3	10.1	24	5	-	1	13

30μV 以上の θ 波（上段左），30μV 以上の α 波で10個以上連続する群（上段右）についての集計と，α 波についてはその詳細（下段）を打ち出している

《CEC512血液生化学自動分析装置》

概　要

本装置は、血液、生化学検査における測定と、処理をCEC555SSマイクロコンピュータにより自動的におこなう．簡単な操作で，正確な，精度の高いデータを得ることのできる自動検査装置である．

測定は光電比色法を採用し、温度制御、反応時間など反応機構の管理制御から、各検査項目に対応する関数演算、光源ランプの補正、検体番号および検査データの表示などの制御をミニコンピュータにより自動処理する．従って、検査能率の大幅な向上と誤差のない検査データを得ることができ、さらに健康管理システムなどの電子計算機とのオンライン接続により、目的にかなったデータ処理が容易におこなえる．

また、機構的に、試薬管理、血清添加の容易さ、完璧な廃液処理，任意の検査項目を自由に選択できる測定の柔軟性などの特長をもち，健康管理システム、検診システム等、オンライン接続の検査機としてはもちろんのこと，比較的小規模な保健所，疫学研究所，病院の臨床検査室などでも効率よく使用できる．

主要仕様

検査項目

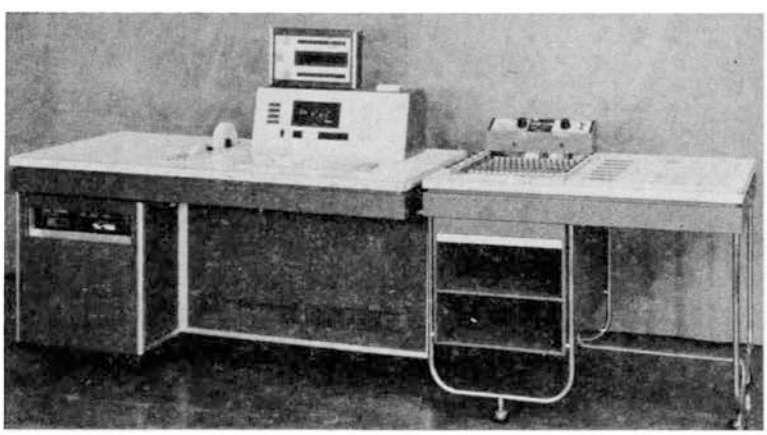
ヘモグロビン (Hemoglobin)
総たんぱく (Total protein)
アルブミン (Albumin)
Z.T.T (Zinc Turbidity Test)
TTT (Thymol Turbidity Test)
GOT (Glutamic Oxalacetic Transaminase)
GPT (Glutamic Pyruvic Transaminase)
アルカリフォスファターゼ
LDH (Lactic Dehydrogenase)
LAP (Leucine Amino Peptidase)
グルコース (Glucose)
BUN
ビリルビン (Total Bilirubin)
コレステロール (Cholesterol)

処理時間……標準６項目測定の場合……30検体／１時間．１日当り約240人の処理可能．

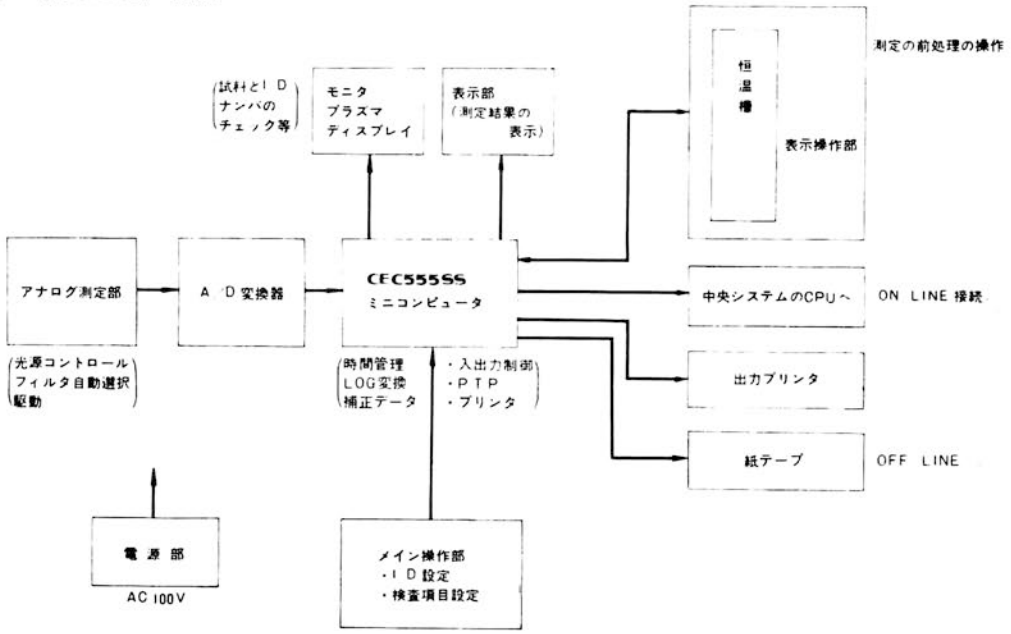
インキュベータ……37℃

１チャネル30液体６チャネルを同時処理可能．

表示精度……１％（±１digit）



装置の構成と各部の機能



出力データの例

オフラインの例

グループNO.	シリアルNO.	ファイルNO.	検査項目	データ
101	00000101		01	18.3
			03	5.37
			05	1.75
102	00000102		01	18.3
			03	10.0
			05	0.48
103	00000103		01	18.3
			03	3.97
			05	1.59
104	00000104		01	18.3
			03	3.97
			05	2.91
105	00000105		01	18.3
			03	4.02
			05	2.90
106	00000106		01	18.3
			03	4.02
			05	2.89
107	00000107		01	18.3
			03	4.02
			05	2.85
108	00000108		01	18.3
			03	4.02
			05	1.64

(ヘモグロビン、アルブミン，TTT,3項目の測定例)

オンラインの例

グループNo.	シリアルNo.	ID. NO.	検査項目	データ
203		003	01	18.1
			02	5.05
			04	3.39
204		004	01	18.1
			02	5.05
			04	3.36
205		005	01	18.1
			02	10.1
			04	5.91
206		006	01	14.0
			02	10.1
			04	5.92
207		007	01	14.0
			02	5.11
			04	5.90
208		008	01	20.5
			02	9.74
			04	9.96
209		009	01	14.0
			02	10.1
			04	9.96
210		010	01	18.3
			02	7.97
			04	2.17

(ヘモグロビン，総たんぱく，ZTT,3項目の測定例)

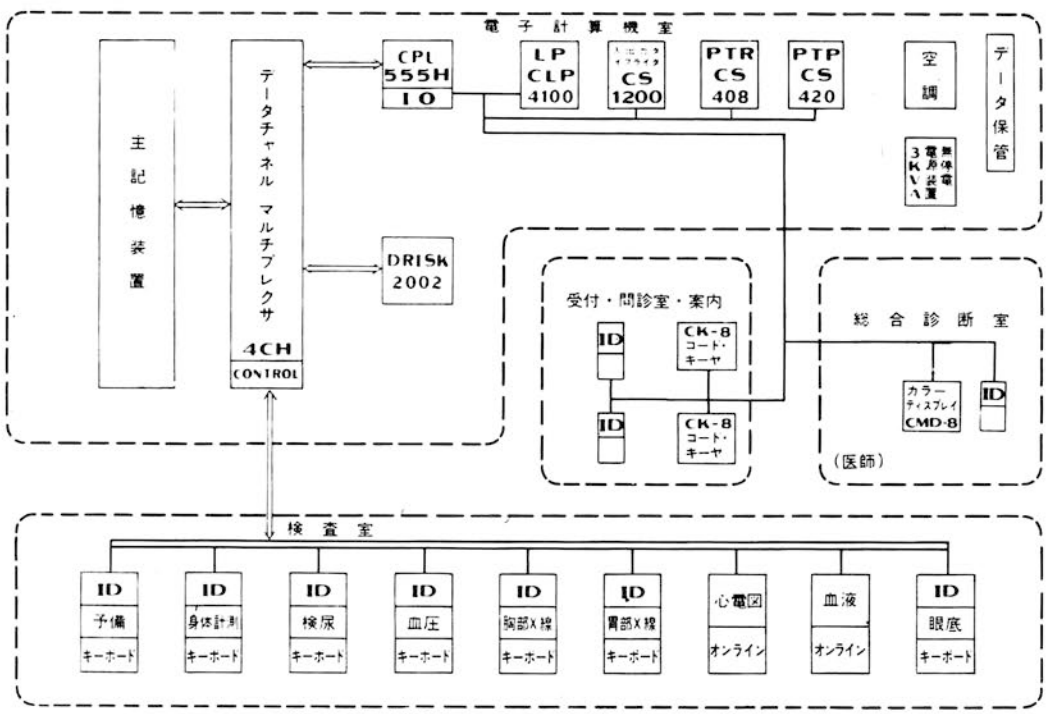
《CEC健康管理センター》

概　要

CEC普及型健康管理システムは，成人病ドックと集団検診のデータ処理の自動化システムである．特に成人病検診を対象とする特質からシステム機能の重点を病歴管理，つまり個人別，団体別検診データのファイルが，正確かつ大量に整理・保管でき，連続的な追跡・比較・企業体・地域集団ごとの統計・調査が容易にできる情報検索機能にすぐれていること，検診の流れとそのデータ処理が無理なく，スムーズにおこなえ，個人検診，企業団体等の集団検診にも共用できることなどにおき，さらに事業目的の性格上設置運営後のシステムライフや，既存設備の共用性，技術革新や新しい検査機器などの追加拡充に対しても容易に対処できるシステムアーキテクチャーを採用するなどの考慮がはらわれており，小型電子計算機システムながら中型機に匹敵する高性能の健康管理システムで，地域行政単位，あるいは企業体単位の健康管理システムとして効率よく運用することができる．

検診項目を標準検診コースと精密検診コースにわけ，検診がスムーズにおこなわれるように配置されている．従って1日当りの検診者数は150人程度が可能になり集団検診に最適である．

カートリッジ磁気ディスク記憶装置は，ファイルの更新，情報検索機能にすぐれ，通知や経過の追跡が容易におこなえる．



普及形健康管理システム　標準システムブロック図》

システム構成（図参照）

このシステムは大別すると，①電子計算機とその端末装置，②各種検査機器と付設の端末装置から構成されている．実際のところ，オンラインでデータ転送が出来ないX線撮影装置，眼底検査（眼底カメラ）などは検査機器に付設されたキーボードよりデータをマニュアルで入力する方式であり，電子計算機室の入出力タイプライター，ラインプリンターとは性格が異なる端末装置である．

以下に各機器のシステム内における機能の概略を説明する

1) 電子計算機室

a) CEC555Hミニコンピュータ20K語：システム全体の制御と演算処理を行なう．検査装置よりオンラインでデータが転送される検査項目は，CEC552心電図自動診断装置による心電図検査，CEC512血液生化学自動分析装置による血液生化学検査，それに身体計測 CK－5コードキーヤによる問診データ等であり，その他の検査項目はキーボードより所定のデータをマニュアルで入力する．電子計算機はこれらの検診者のデータを検診者ごとに整理し，所定のフォーマットにより印字させる．(印字例参照)

b) DRISK－2002カートリッジ磁気ディスク装置(カートリッジ式3.2M語)：この装置はシステム全体を制御するシステムプログラムの収納と，カートリッジディスクに検診者ごとのデータをファイルしておく機能を持つ．カートリッジ磁気ディスクは1個で1.6M語のデータが収納出来る．これは通常のデータ記録を行なうとすれば1個当り約600人分程度が記録出来ることになる．

c) CLP－4100ラインプリンタ：この装置は電子計算機により演算処理されたデータを規定のフォーマットによって印字する装置である．

b) CS－1200入出力タイプライタ

e) CS－408高速紙テープリーダ

f) CS－420高速紙テープパンチャ

以上の各装置はシステム制御，情報の入出力に使用される．

2) 各種検査機器および付設端末装置

a) CK－5コードキーヤ：この装置はあらかじめ検診者によって記入されたデータを計算機に入力する装置である．設問に対応する項目がパネルシートに実名の語で記入されてをり，担当者はそのキーと，テンキーの操作のみでデータが入力出来る．これはこのシステムの新しい試みであり，誰れにでもデータインプットが遂行出来るものである．（入力業務にはCK-6，CK-8コードキーヤを利用する場合もある：装置の概要については周辺装置編コードキーヤの項目を参照のこと．）

b) キャラクターディスプレイ：総合診断室に設置し，一巡の検査結果をディスプレイ上に描画する．

担当医師はこのデータとレントゲン結果を参考にして総合診断を行なう．

c) キーボード：オンラインで接続されない検査機器については，キーボードよりデータ入力を行なう．これも各キー上に実名で項目が印刷されているのでその項目と数値を入力すればよい．

d) 心電図検査装置(CEC552心電図自動診断装置)：出力データは3桁のミネソタコードであり，このデータはオンラインで電子計算機へ送られる．

e) 血液生化学検査(CEC512血液生化学自動分析装置)：この装置はCEC555SSマイクロプロセッサ（ワンチップCPU）内蔵の分析装置であり，温度制御，反応時間制御，測定項目に対応する関数演算を自動的に行なう装置である．検査項目は，Hemoglobin，total protein，Albumin，

ZTT, GOT, GPT, アルカリフォスファターゼ, LDH, LAP, Glucose, BUN, Total Bilirubin, Cholesterolの14項で随意検査項目を選定することが出来る. 出力は検査データ5桁, その他I.D. 番号, 検査項目番号等のデータが電子計算機に転送される.

f) 身体計測と肺機能：身長体重測定には自動計測装置を用い, 身長測定用レバーが上部より降下し, 身長値に達した時点にて, 体重値と身長値が同時に電子計算機に送られる. 肺機能については, 現在はオフラインの装置を用いているためC項で述べたキーボードより入力する方法が用いられる.

g) その他の機能：その他の機能として, 被検者番号(I.D)リーダーが各種検査室に配置されており, 検査機器の測定結果はI.D.カード リーダー によって, 計算機に被検者番号を指示し, これによって, 各個人個人のファイル記憶する. また検査室では, はじめに被検者が入室した際にI.D カードをリーダーに挿入すると入口に使用中を示すランプ表示が点 し, 同一検査室に被検者が二人以上入らないように考慮している.

■CEC555Hミニコンピュータ,CEC555SSマイクロコンピュータの仕様についてはハードウェア編の該当項を, IDカードリーダ, キーボード入力装置については周辺装置編の該当項目をそれぞれ参照のこと.

■CEC552心電図自動診断装置. CEC552D心電・心音自動診断装置, CEC512血液生化学自動分析装置については応用編（アプリケーション）の該当項目を参照のこと.

特　長

1. 検診項目を標準検診コースと精密検診コースにわけ, 検診がスムーズにおこなわれるように配置されている.
2. 従って1日当りの検診者数は150人程度が可能になり集団検診に最適である.
3. 移動検診のデータ処理にも共用できる（カセット式心電図記憶装置等）
4. 病院と併設の場合は病院業務のデータ処理にも共用できる.
5. 新らしい検査装置が開発されたとき容易に拡張し, 接続することができる.
6. カートリッジ磁気デイスク記憶装置は, ファイルの更新, 情報検索機能にすぐれ, 通知や経過の追跡が容易におこなえる.
7. 低額の設備投資で設置運営ができる.

標準検査項目例

(イ)標準検診コース

①問診　②身体計測　③採尿　④眼底　⑤採血　⑥心電図　⑦血圧　⑧胸部X線　⑨診察

※必要に応じて検査項目と, 検査装置を追加増設することができる.

(ロ)精密検診コース

①～⑨までの標準検診コースに⑩糖飲負荷　⑪胃部レントゲン　⑫負荷心電図　⑬その他の検査　⑭診察, が追加される.

検査項目中オンラインで接続されない検査機器の入力は、**CS－50キーボード入力装置**により入力する.

■この他に運動機能, 体力測定等の検査機器を附設したCEC健康増進センタの標準システムも用意されている.

健診結果の印字例

検　診　成　績　書

項目		検査内容					
身体計測	身長	157 cm	体重	57.3 kg	肥満度	109 %	
	皮厚 腹部	20 mm	肩胛間部	mm	上腕部	mm	
肺機能	肺活量	2650 cc	努力肺活量	2550 cc	予測肺活量	3620 cc	
	一秒率	78 %	一秒量	2000 cc	%肺活量	73 %	
聴力	右 250Hz	7 dB	500Hz	6 dB	1 kHz	0 dB	
	右 2 kHz	7 dB	4 kHz	17 dB			
	左 250Hz	0 dB	500Hz	4 dB	1 kHz	0 dB	
	左 2 kHz	5 dB	4 kHz	17 dB			
検尿	蛋白	-	糖	-	尿潜血		
	pH	6	ウロビリノーゲン	-			
眼底検査	高血圧変化	0 度	動脈硬化変化	1 度	K.W.	1 度	眼圧 5 mmHg
血液検査	Hb(血色素量)	14.6 g/dl	総コレステロール	185 mg/dl	中性脂肪	240 mg/dl	
	総蛋白	7.2 g/dl	ヘマトクリット	43 %			
	Albumin	4.1	ZTT	5.	TTT	5	GOT 20
	GPT	19	ALP	9.23	LDH	235	% 1.32
	Glucose	108	BUN	9.	Bilirubin	0.5	尿酸 5.3
血圧	最高	130 mmHg	最低	80 mmHg	体温	℃	
心電図	心拍数	072	ミネソタコード	3-1-0			
胸部X線	肺所見	-	大動脈硬化	-	心胸比	50 %	
胃部X線	食道	-	胃	-	十二指腸	-	
負荷心電図							
その他	梅毒反応		ASLO	u.	RA		CRP

総合判定：
ｲｼﾞｮｳｱﾘﾏｾﾝ

指導事項：
ﾄｳﾆｮｳﾋﾞﾖｳﾉﾗﾝﾂｵ ﾓｳｲﾁﾄﾞｼﾃﾗｼｱｹﾞﾏｽ

様　コース　C

年令　41 オ

ID番号　012

8日

理センター

元本郷町1丁目8番地7号

-23-1211



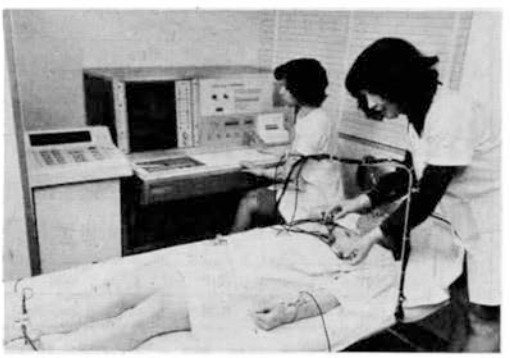
受付・問診（CK-5利用）



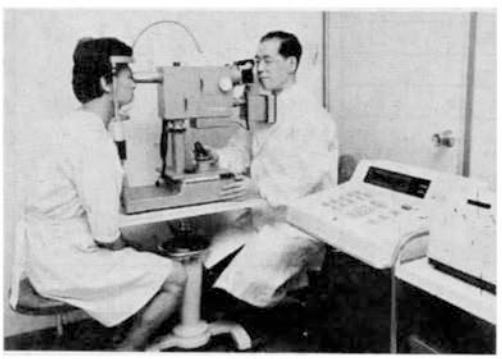
総合診断（CCD-5S利用）



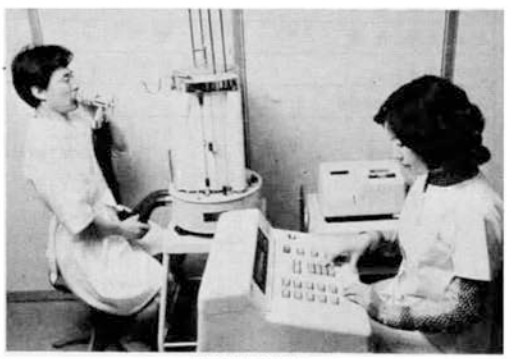
血液生化学検査（CEC512）



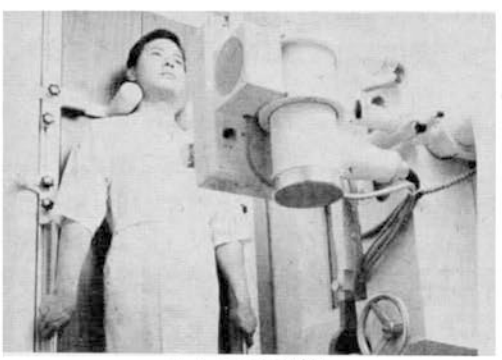
心電図検査（CEC552）



眼底カメラ



肺機能検査



レントゲン検査



IDカード（CS-50）

《病院事務処理システム》

概　要

病院における情報処理システムは，病院機能にもとづく特殊性，例えば情報発生源が多岐にわたること，情報の種類が多種多様であること，処理業務の多面性と相互干渉，組織形態の特殊性などがあり，一般の事務処理システムと比較して複雑になる傾向にある．加えて，近年医療の高度化，患者数の増加，それに伴う病院業務の拡大等の原因により事務処理の膨張と複雑化にいっそうの拍車がかけられているのが現状である．

CEC 555電子計算機による病院事務処理システムは，各種の処理業務のうち，一番煩雑である窓口会計処理業務と，診療報酬明細書の作成（レセプト作成）を第一の課題とし，その他の処理業務はそれぞれの機能，形態に即応した形で病院経理事務，医薬品の在庫管理，各種の日計業務，資材管理，給食栄養管理業務，経営分析にいたるまで必要に応じて随意に導入することのできる柔軟なソフトウェアシステムとして整備されており，さらにシステム運営を容易にするためデータの入力機構に万能入力端末装置コードキーヤを使用し，システム運営上一番問題視されている入力業務の簡素化と入力ミスの発生を防ぐ方式を採用するなど，導入・運用に関して経済面，運用面で効率よく使用できるよう配慮を加えたシステムである．

万能入力端末装置コードキーヤは，従来のようにパンチカード，あるいはタイプライタを使用せず，ブック式シートに実名の語で印刷された項目を特殊ペンで軽くタッチし，数値キーの操作で正確にデータが入力できるもので，品名の記入や転記などの手続きも不要であり，面倒な作業が一切なく，キーパンチャ，オペレータなど特殊な専門職の必要もなく，実際に利用する立場のユーザから，入力業務が大幅に改善されたと高く評価されているものである．

システムは，当面の規模と必要度にマッチした最適システムで，導入後業務の拡張に即応した機器を必要に応じて増設できる拡張機能を備えており，比較的小規模の病院から総合病院にいたるまで幅広い層で利用できる総合的な病院事務処理システムである．

主な処理業務

1．窓口会計業務（標準システム処理）

各帳表類

(1)	自己負担請求書	毎月
(2)	診療科別患者日報	毎日
(3)	保険種別収入金	毎日
(4)	診療報酬明細書	毎月
(5)	レセプト集計表	毎月

2．入院請求業務（拡張システム）

(1)	自己負担入院料金請求書	10日毎
(2)	診療報酬明細書	毎月
(3)	レセプト集計表	毎月

3．医薬品在庫管理業務（拡張システム）

(1)	医薬品出庫一覧表	毎日
(2)	注文指示書	毎日
(3)	発注一覧表	毎日
(4)	未納日報表	毎日
(5)	入在庫日報表	毎日
(6)	在庫一覧表	毎月

4．給食栄養業務（拡張システム）

(1)	献立大枠表	毎週
(2)	栄養出納表	毎週
(3)	病種別献立明細書	毎週
(4)	給食日報	毎日
(5)	材料注文	毎日
(6)	日計表	毎日
(7)	納入一覧表	毎日
(8)	材料納入表	毎月
(9)	給食栄養状況表	毎月

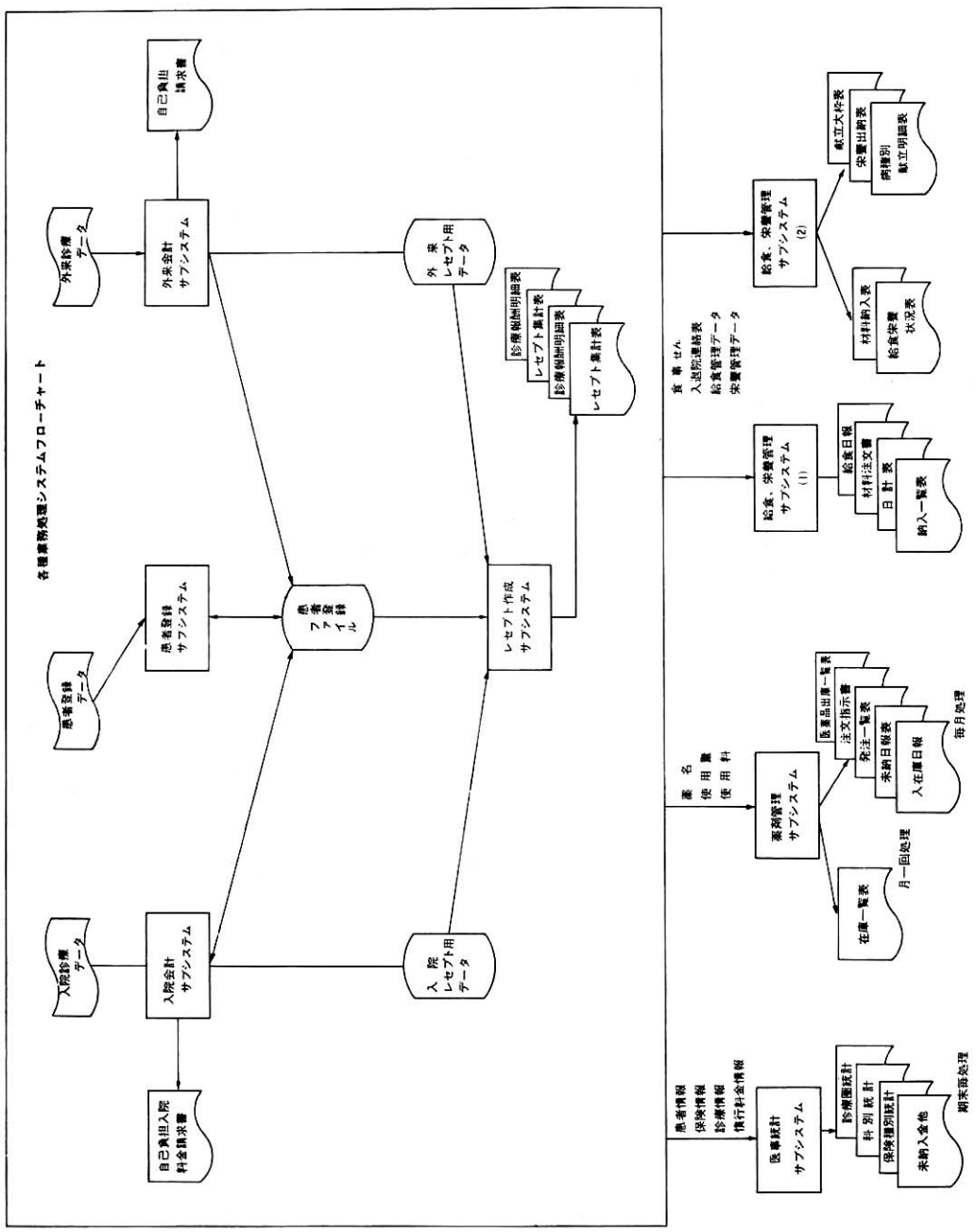
5．経営分析管理資料　毎月

前項で述べた業務をおこなうことにより患者数に関する統計，傷病に関する統計，診療関連統計等を作成できる．

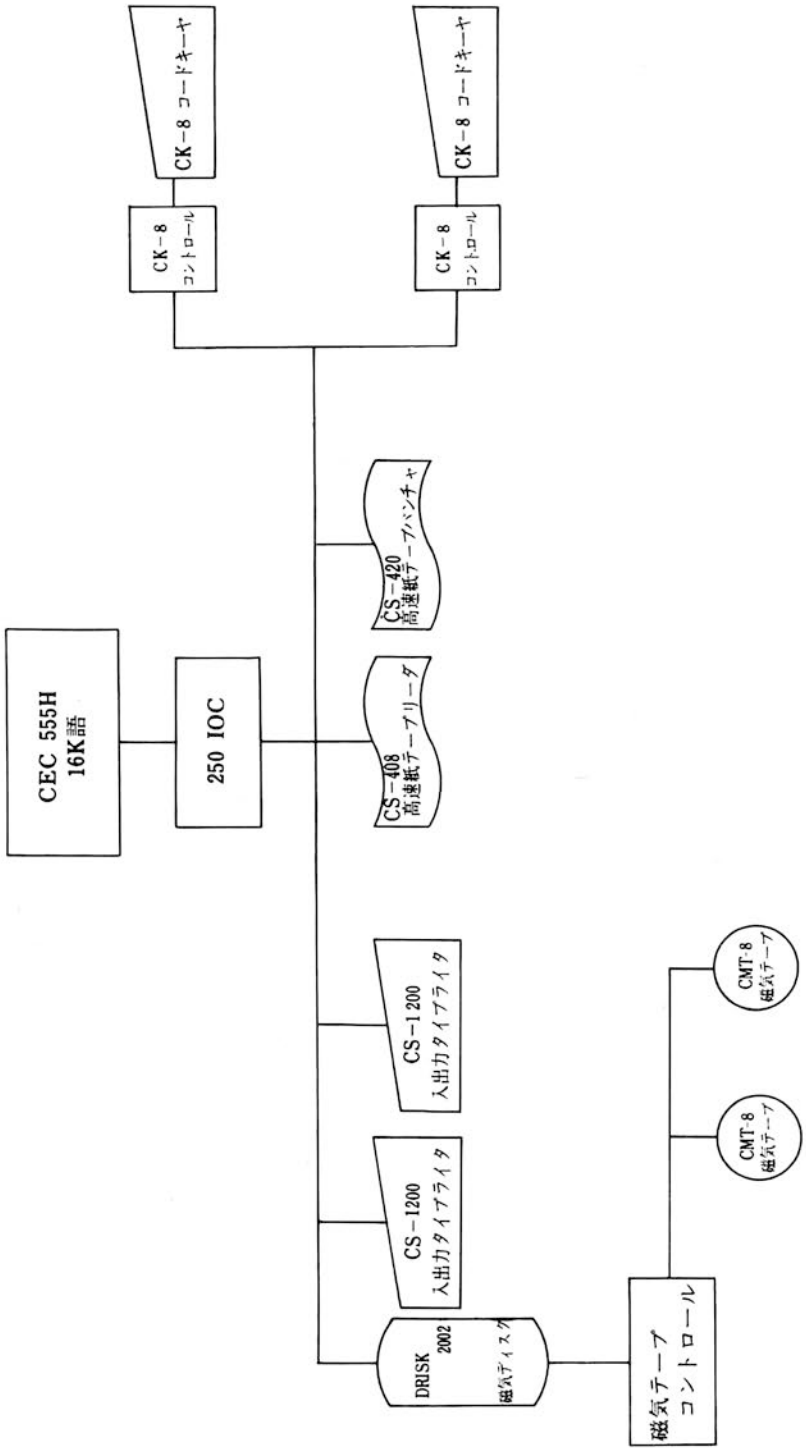
（拡張システム）

(1) 診療圏統計
(2) 科別統計
(3) 保険種別統計
(4) 未収入金

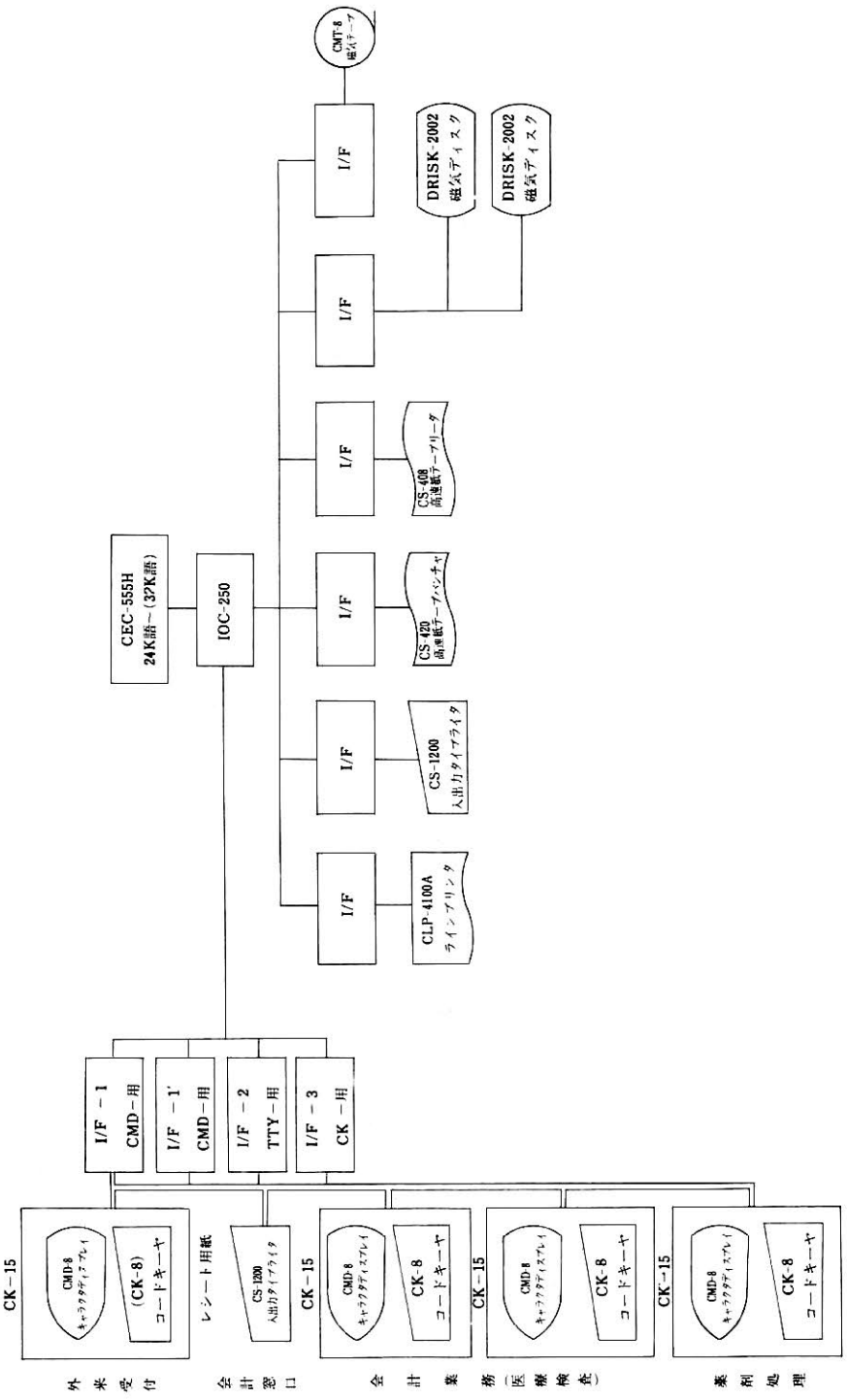
総合システムフローチャート



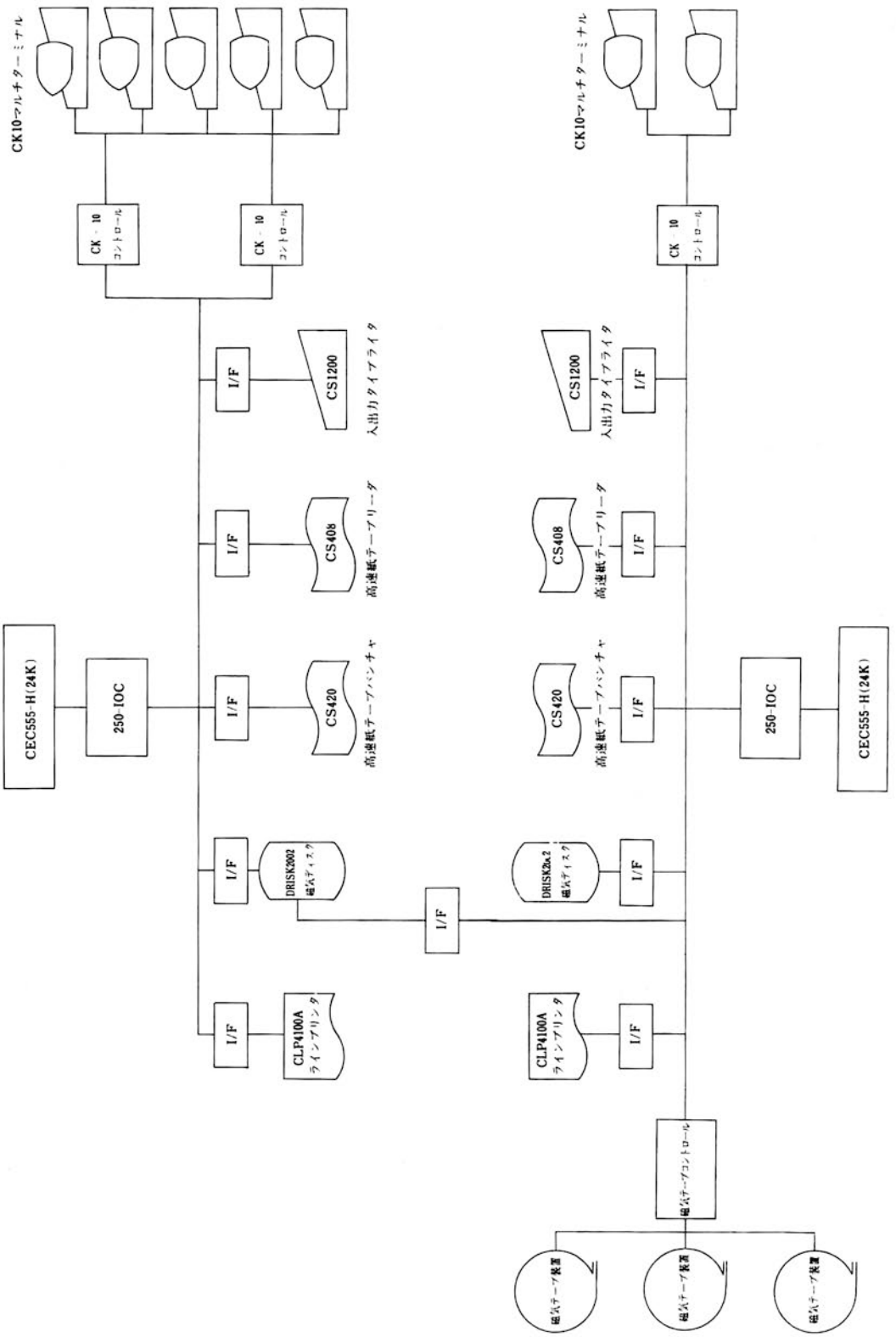
小型システムの構成例


CEC 555H
16K語
250 IOC
CK-8 コントロール
CK-8 コードキーヤ
CK-8 コントロール
CK-8 コードキーヤ
CS-420
高速紙テープパンチャ
CS-408
高速紙テープリーダ
CS-1200
入出力タイプライタ
CS-1200
入出力タイプライタ
DRUSK
2002
磁気ディスク
磁気テープ
コントロール
CMT-8
磁気テープ
CMT-8
磁気テープ

中型システムの構成例


CMT-8
磁気テープ
I/F
DRISK-2002
磁気ディスク
DRISK-2002
磁気ディスク
I/F
I/F
CS-408
高速紙テープリーダ
I/F
CS-420
高速紙テープパンチャ
I/F
CS-1200
入出力タイプライタ
I/F
CLP-4100A
ラインプリンタ
CEC-555H
24K語～(32K語)
IOC-250
I/F －1
CMD－用
I/F －1'
CMD－用
I/F －2
TTY－用
I/F －3
CK －用
CK－15
CMD-8
キャラクタディスプレイ
(CK-8)
コードキーヤ
レシート用紙
CS-1200
入出力タイプライタ
CK－15
CMD-8
キャラクタディスプレイ
CK-8
コードキーヤ
CK－15
CMD-8
キャラクタディスプレイ
CK-8
コードキーヤ
CK－15
CMD-8
キャラクタディスプレイ
CK-8
コードキーヤ
外来受付
会計窓口
会計業務
(医療検査)
薬剤処理

大型システムの構成例


CK10マルチターミナル
CK-10 コントロール
CK-10 コントロール
CK10マルチターミナル
CK-10 コントロール
I/F
CS1200
入出力タイプライタ
I/F
CS408
高速紙テープリーダ
I/F
CS420
高速紙テープパンチャ
CEC555-H(24K)
250-IOC
I/F
DRISK2002
磁気ディスク
I/F
CLP4100A
ラインプリンタ
I/F
CS1200
入出力タイプライタ
I/F
CS408
高速紙テープリーダ
I/F
CS420
高速紙テープパンチャ
250-IOC
CEC555-H(24K)
I/F
DRISK20x2
磁気ディスク
I/F
CLP4100A
ラインプリンタ
I/F
磁気テープコントロール
磁気テープ装置
磁気テープ装置
磁気テープ装置

パネルシート例　処置

SHEET No.	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

	01	02	03	04	05	06	07	08
01			Hb	Na, Cl, K	BSP	鍼　　灸		
02	そ　の　他	手　術　料	Ht	mg	RA.CRP.ASLO	脳　　波	検　査　料	
03	電気療法	切開（小）	Tp	Ca	尿　　酸	精尿眼底	尿　一　般	
04	ジアテルミー	〃（中）	アルブミン	WAR（1）	T　G	眼底カメラ	尿　沈　渣	
05	長　　波	〃（大）	ZTT	WAR（2）	B－リポ	眼圧測定	尿蛋白(半定量)	
06	紫　外　線	創傷処理（小）	TTT	総ビリルビン	BUN	ガストロテスト	尿蛋白（定量）	
07	マッサージ	〃（中）	GOT	Hb	アルブミン	出血時間	尿ジアスターゼ	
08	麻　酔　料	〃（大）	GPT	Ht	LDH	凝固時間	便　鏡　検	
09		腹部手術	AI－P	ZTT	LAP	血糖定量	便　虫　卵	
10			LAP	TTT	TC	血　　沈	便ヘマテスト	
11			LDH	γ－GTR	TP	ヘマトクリット	細菌培養	
12			TC	アミラーゼ	GOT	血　小　板	細菌培養同定	
13			TG	赤　血　球	GPT	血型（AOB）	心　電　図	
14			β リポ	白　血　球	AL－P	血型（RH）	心電図負荷	
15				採血（静）7	ルゴール反応	W.R.Hb.BB	心　音　図	
16				採血（耳）3	CHE	MG		
17	KX－P大4F×1	上腕骨4F×12R	背椎　4F×2		ネブライザー			
18	4F×1 BICO	6F×12R	6F×1	レントゲン料	変形機械矯正術	湿布処置10	処　置　料	
19	胸部大角F×1	前腕骨4F×12R	骨盤大4F×1	胸部大4F×1	胃　洗　浄	15	創傷処置10点	
20	MX－D4F×6 6F×2（スポ）	6　×12R	4F×1	大4F×2	浣　　腸	20	15点	
21	MX－D4F×6 6F×3（スポ）	肘関節4F×12R	大腿骨4F×2	4F×1	高位浣腸	30	20点	
22	MX－D4F×7 6F×2（スポ）	6F×12R	6F×2	8F×12R	洗　　腸	50	30点	
23	MX－D4F×7 6F×3（スポ）	手関節4F×12R	下腿骨4F×12R	4F×2	注　　腸	96	40点	
24	頸椎 6F×4	6F×12R	6F×12R	4F×22R	肛門処置	眼　処　置	50点	
25		手指　8F×12R	膝関部4F×12R	6F×1	吸　　入	耳　処　置	60点	
26		キヤビネ×12R	6F×12R	腹部大4F×1	酸素吸入	耳管処置	100点	診療終了
27		MX－D4F×6	足関部4F×12R	4F×1	腰椎穿刺	耳管ブジー法	192点	消　　去
28		6F×2（ス）B300	6F×12R	6F×1	胃十二指腸ゾンデ34点	鼻　処　置	皮膚科処置10点	直接指示
29		MX－D4F×5	足背部4F×12R	頭部　4F×1	〃　25点	口腔処理	15点	日
30		6F×1（ス）B250	6F×12R	4F×2		咽頭処置	20点	回
31		TX－P4F×2	足指　8F×12R	6F×2		副鼻腔洗浄	30点	NO1
32		TX－P6F×1	キヤビネ×12R	頸椎　6F×2		喉頭処置		NO2

CEC CK-6,CK-8 SHEET

診療報酬請求明細書出力印字例

昭和 49年 09月分 **診療報酬請求明細書** （被保険者／被保険者であった者 分・入院外） ㋑

1098

被保険者証・被保険者手帳の記号番号	
記号	番号 21473
氏名	（男・女）明・大・昭 3[illegible]年 月生
職務上外の別	上/外 乗船中発病 下船中発病

医療機関記号番号 35, 1661, 4

管掌別	政・組・日・船・共・自
保険者名	394

様式第四

保険医療機関の所在地および名称開設者氏名

東京都日野市多摩平3丁目11の2
石塚病院
医師 石塚民幸
電話 日野（0425）82－0563 番

傷病名		診療開始日		診療実日数
(1)		(1)	年 月 日	9 日
(2)		(2)	年 月 日	転帰
(3)		(3)	年 月 日	治ゆ 死亡 中止

⑩診察料	⑪ 初診	時間外・深夜	2回	130点
	⑫ 再診	再診 5×	7回	105
		内科再診 13×	回	
		時間外加算 3×	回	
		深夜加算 60×	1回	100
	⑬指導	26×	回	
	⑭ 往診	普通	回	
		夜間	回	
		深夜・暴風雨雪・難路	回	
		同一家屋	回	
⑳投薬料	㉑ 内服	薬剤	6単位	141
		調・処 12×	2回	24
	㉒ 屯服	薬剤	4単位	4
		調・処 5×	1回	5
	㉓ 外用	薬剤	単位	
		調・処 6×	回	
	麻薬・毒薬加算	2×	回	
㉚注射料	㉛皮下筋肉内		回	
	㉜静脈内		回	
	㉝その他		回	
㊵処置料			7回	105
			回	
			回	
	薬剤		回	
㊿手術料			1回	51
			回	
	薬剤		回	
60検査料			回	
			回	
	薬剤		回	
70レントゲン料			回	
			回	
			回	
80その他				

21 エリスロシン チフアブル 100MG 6T
アスメトン 3C
19 × 3 = 54
CM 250MG 3T
プロクターゼロ 3C
29 × 3 = 87
22 1 × 4 = 4
40 グカ ショチ
15 × 7 = 105
50 ホウゴウ
51 × 1 = 51

公費負担額	結核	点	結核予防法 第34条による申請	承認・不承認
	精神	点		
合計	点 665	決定	点	初診時・部負担金額 200円・50円

《検診データ自動処理システム》

概　要

医用情報システムには，病院事務処理業務，生体現象の自動処理から各種の自動診断，健康管理システムにいたるまでさまざまな利用例があり，また，その目的についても多岐にわたっている．

このシステムは，疫学面，臨床面での利用を主とした検診データの個人別・団体別のファイル・検索，あるいは各種疫学データの統計処理を自動的におこなうシステムである．

入力された各種項目のデータは被検者ごとの総合判定，あるいは医師による指導事項を含めて，個人別・集団別一覧表として出力されるほか，平均値と標準偏差など，各種の統計処理をおこない公衆衛生上重要な疫学資料を自動的に作成することができる．

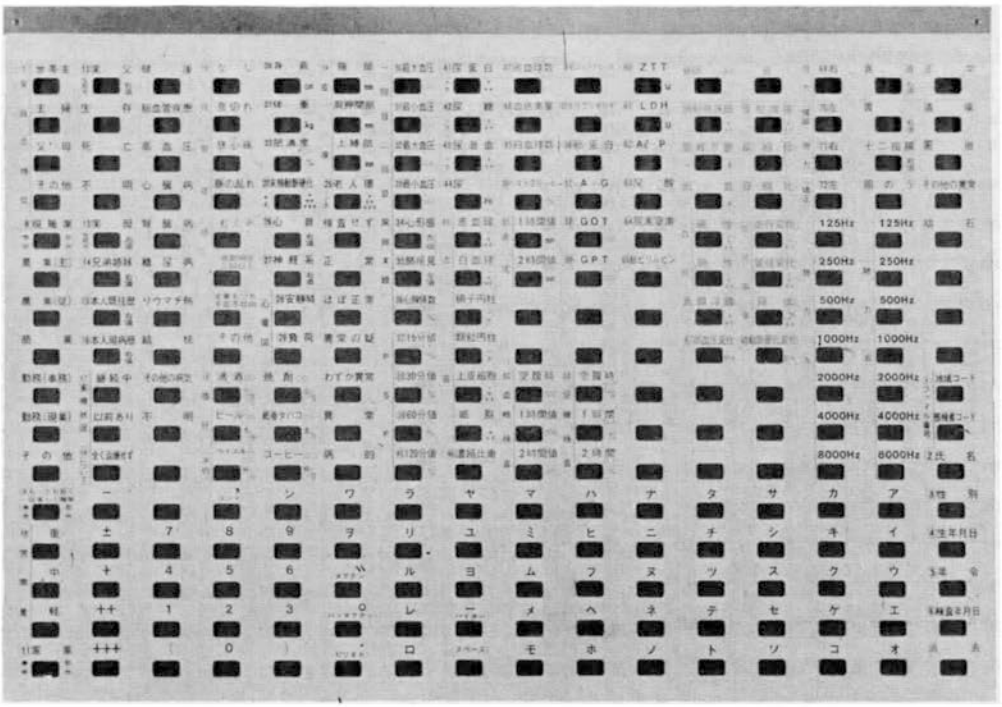
データの入力は誰れにでも容易に入力作業がおこなえる万能入力端末装置コードキーヤを使用している．この装置は，パネルシートに実名の語で印刷された検診項目や問診項目などに対応するキーの操作と，数値をテンキーより入力するだけでデータとして紙テープにせん孔し，あるいはオンラインで磁気ディスクに入力することができるもので，従来の電子計算機システムのようなデータのコード化等の繁雑な前作業は皆無であり，手書きによる順序整理の必要もなく，容易に操作することができ，システムの導入にあたってオペレータ，キーパンチャなど特殊な専門職を必要としない．このシステムは疫学，予防衛生上の極めて重要な資料作成のみならず，病院，医院，診療所，保健所などでの情報処理に利用され，さらに各種の病院事務管理業務にも共用することができ，導入経費が安く運営が容易であり長期的にわたり健康福祉に大きな役割を果すことが期待される．また将来の拡張性も考慮されており，かつ経済効率の高いシステムである．

データ処理の内容

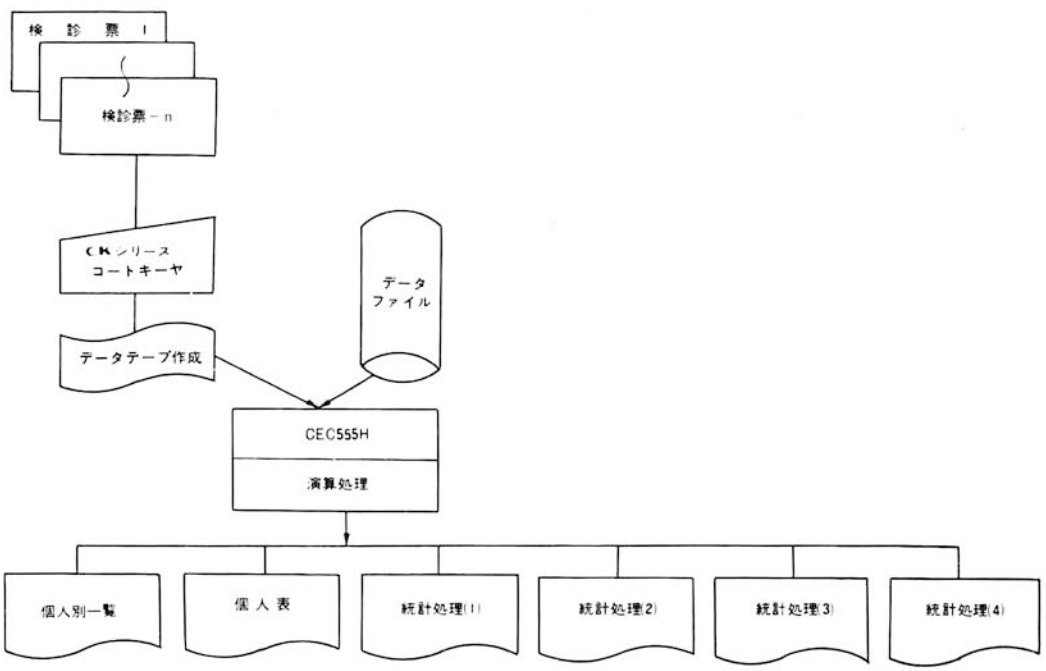
(1)　検診データの内容

1．検診地域
2．患者番号，ファイル番号
3．氏　名
4．性　別
5．年号，生 年月日
6．年　令
7．検診年月日
8．身体計測（身長，体重，腹部，肩胛間部上膊部）
9．血圧，MAX，MIN
10．検尿（蛋白，糖，潜血，PH）
11．心電図（ミネソタコード）
12．胸部X線（心形態，心胸部係数，肺所見）
13．血液検査（血液素量，総コレステロール中性脂肪）
14．眼底検査（フィルム番号，高血圧変化，動脈硬化変化）
15．診察所見（老人環，心音，神経系，末梢動脈硬化）

パネルシート例(CK-3)



システムフロチャート



検診データ出力印字例

6　出力帳票例

検診地区　0001

個人別一覧表

昭和47年11月14日検診　2頁

No.	氏名	性別	年令	身長 (cm)	体重 (kg)	肥満度 (%)	血圧 MAX	血圧 MIN.	尿所見 糖	尿所見 蛋白	尿所見 潜血	心電図 (ミネソタコード)	眼底 H	眼底 S	血液 比重	血液 Hb g/dl	血液 ヘマトクリット %	総合判定
0019		M	60	163.4	54.0	- 5	198	118	-	-	-	7-2-0	2	2		14.3	120.0	
0020		F	68	145.0	52.0	+14	148	80	-	-	-		0	2		13.3	190.0	
0021		F	67	140.9	49.0	+12	160	104	++	-	-	3-1-0 9-4-2				12.2	220.0	
0022		F	49	145.9	44.0	- 4	120	70	-	-	-	5-4-0	1	1		11.3	260.0	
0023		F	55	155.8	55.0	+ 7	162	100	-	+++	+	5-5-0	2	1		13.6	220.0	
0024		F	53	154.8	51.0	0	162	106	-	+++	-	3-3-0				13.8	195.0	
0025		F	48	152.2	56.0	+13	128	110	-	-	-		1	0		12.2	165.0	
0026		F	58	144.0	43.0	- 5	122	82	-	-	-	1-0-0	1	1		12.6	230.0	
0027		F	50	138.9	33.0	-22	114	72	-	-	+		1	1		12.4	165.0	
0028		F	41	151.0	44.0	-10	106	60	-	-	-		0	0		12.2	190.0	
0029		F	68	146.3	63.0	+26	170	104	-	-	-	3-1-0 4-2-0 5-2-0 7-3-0	2	1		13.4	280.0	
0030		F	65	141.2	54.0	+24	180	94	++	-	-	3-3-0	0	2		12.8	220.0	
0031		F	70	146.0	45.0	- 2	122	80	-	-	-	1-0-0	1	2		13.2	210.0	
0032		M	41	170.0	66.0	+10	130	86	-	-	-	9-4-2	0	0		15.4	280.0	
0033		F	37	163.4	54.0	- 4	122	74	-	-	+	8-9-0	1	1		14.5	165.0	
0034		F	50	150.3	54.0	+12	156	90	-	-	-	1-0-0	1	1		14.8	210.0	
0035		F	70	143.4	62.0	+26	162	98	++	-	-	3-1-0 4-1-0 5-2-0 9-4-2	2	2		14.2	190.0	
0036		F	56	148.0	51.0	+ 8	140	88	-	-	+	1-3-2 9-2-0	0	0		13.1	230.0	

❺繊維産業システム

《CEC510カラーパターンアナライザー》

CEC510は，デザイナーによって描かれた絵柄について，その色合と形状を光学的にスキャンニングし，CEC555電子計算機により自動的に柄信号の紙テープを作成する装置で，従来熟練した技術者が長時間を要して造りあげた紋紙のせん孔や，ニット機械の制御，その他，絵柄の焼つけ，印刷，塗布など高い熟練度が要求されていた手作業を制御機器，端末機器との組合せで自動的に，短時間で，忠実におこなうことのできる画期的な自動図柄分析装置である．

この装置は紋紙の自動せん孔，丸編機の制御などのほか，印刷などの機器と連動して広く生活文化産業の分野で利用することができる．

CEC510シリーズの特長

図案の描き方は従来のまま

図案の描き方はこれまで通り，フリーハンドの図案でも意匠図でもOK．

どんな色でも思いのままに使用できる．

識別は任意の16色，同系色もミニコンピュタで判別でき，図案にはどんな色も自由にできる．

X軸，Y軸は自由に設定可能

同一の図案からX軸，Y軸の分割を，独立して設定できる．

取り扱いが簡単

誰れにでも簡単に，確実に操作できる．

織物の場合糸の使用量が計算できる

絵素の数を色別に計算でき，それぞれの糸の使用量を計算することができる．

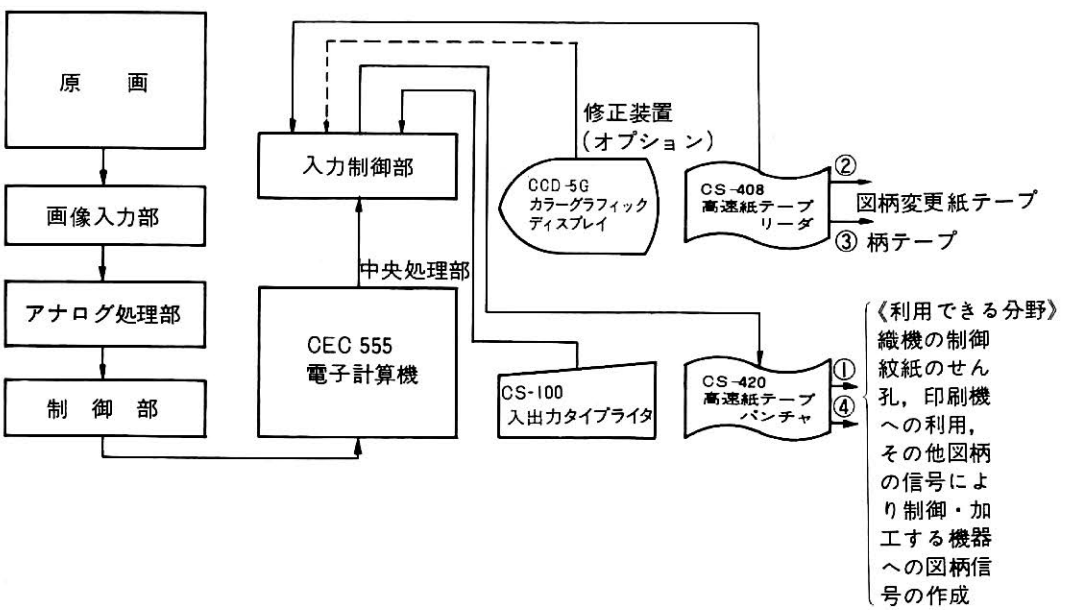
省力化，生産性のアップに抜群の威力を発揮

製品の多様化，寿命サイクルの早さ，高賃金労働力不足，熟練者不足，コスト高騰など山積する問題を一挙に解決して，将来にわたる知識集約形への効率よい移行が可能．

	CEC510 B3-II型	CEC510 A1-II型
図案寸法	B3 サイズ以下	A1(準B1)サイズ以下
ドラム寸法	390×515mm	645×1020mm
分割数	3900×5150以下	6450×10200以下
駆動	0.1mm/ステップ	0.1mm/ステップ
走査速度X	400ステップ/秒	400ステップ/秒
走査速度Y	500ステップ/秒	400ステップ/秒
走査速度連続	1200ステップ/秒	1200ステップ/秒
SLOW/OP時間	0.5秒	0.5秒
変換濃度	0～2.0	0～2.0
色分解	R，G，B 3チャンネル	R，G，B 3チャンネル
分解能	0.2φ又は0.3φ	0.2φ又は0.3φ
1ラインの時間	約4秒	約6.5秒
意匠図読取り	可能（オプション）	可能（オプション）
意匠図のます目	1mm×1mm以上 （用紙は用意してあります）	1mm×1mm以上 （用紙は用意してあります）
使用色	任意の16色	任意の16色



システム構成


原　画
画像入力部
アナログ処理部
制　御　部
入力制御部
中央処理部
CEC 555
電子計算機
修正装置
（オプション）
CCD-5G
カラーグラフィック
ディスプレイ
CS-408
高速紙テープ
リーダ
②
図柄変更紙テープ
③ 柄テープ
CS-100
入出力タイプライタ
CS-420
高速紙テープ
パンチャ
①
④
《利用できる分野》
織機の制御
紋紙のせん
孔，印刷機
への利用，
その他図柄
の信号によ
り制御・加
工する機器
への図柄信
号の作成

CEC510カラーパターンアナライザは各種の繊維産業に利用されている.

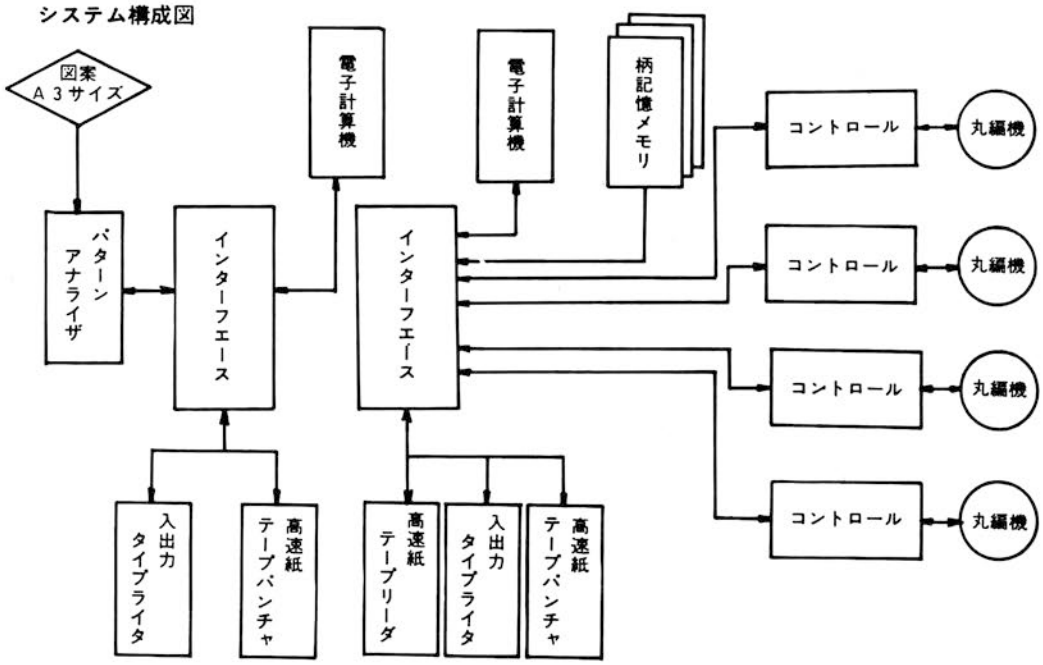
《電子式自動丸編機システム》

図柄分析装置で出力された柄データを入力することによって丸編機を1台から最大4台まで独立して任意な柄を編むことができるシステム。

柄　色　数：任意の4色以内
柄メモリ：標準　64Kステッチ×64Kステッチ／1枚で32枚まで増設可能
丸編機の接続台数：最大4台
柄の転回：有
柄の倍長：有
柄の組合せ：有



システム構成図



《自動紋紙せん孔システム》

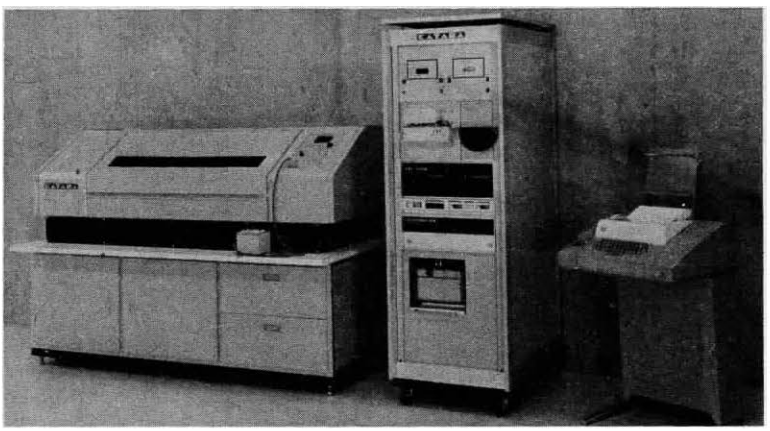
図柄分析装置で出力された柄データを入力することによって自動的に紋紙をせん孔するシステム。

柄　色　数：任意の16色以内

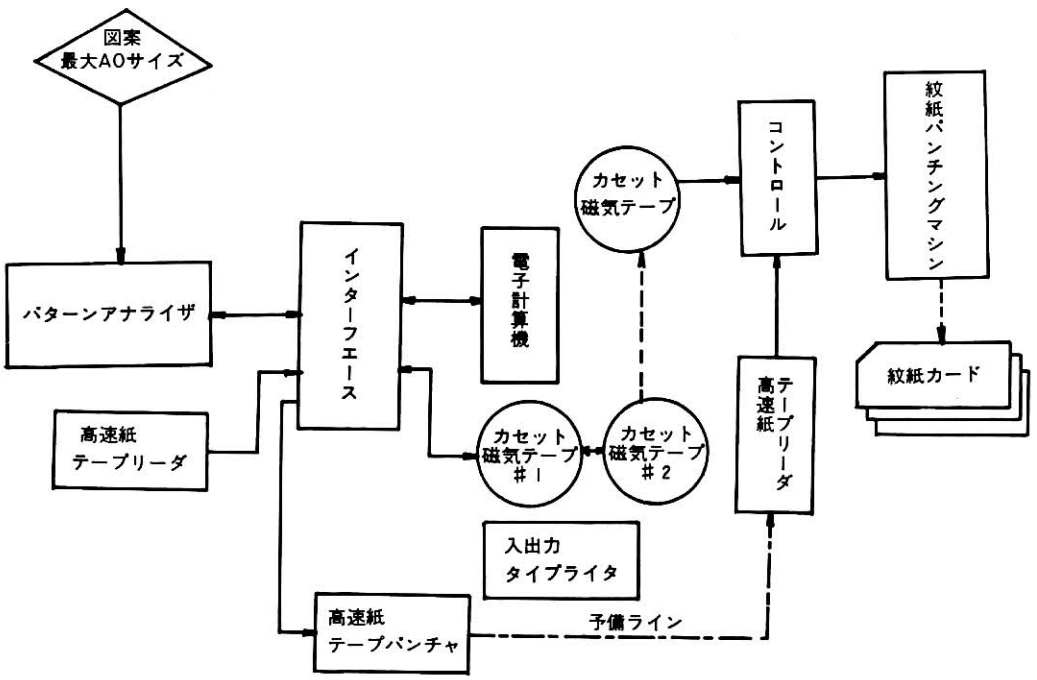
紋紙の種類：関西、関東、バンサンジ各種可能

紋切り速度：横彫り方式　15枚／分

ワンパンチ方式　30枚／分



システム構成



《ラッセル自動編システム》

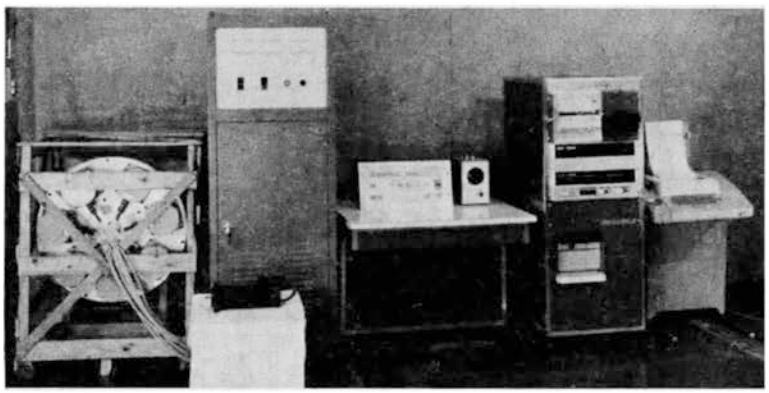
図柄分析装置で出力された柄データを入力することによって既設のラッセル機をコントロールし自動的に柄出しを行なうシステム。

制　御　軸：12軸（標準）

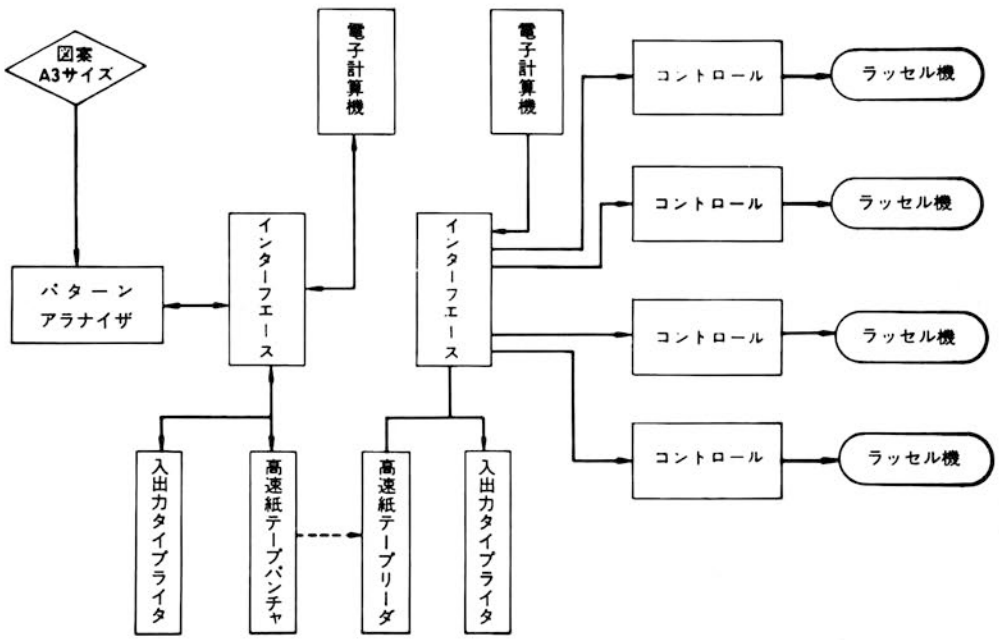
ガイド移動距離：60㎜（14針／ROUND）

1 コ ー ス：1回転

1ピッチ(針)の距離：1.4㎜



システム構成



《自動捺染システム》

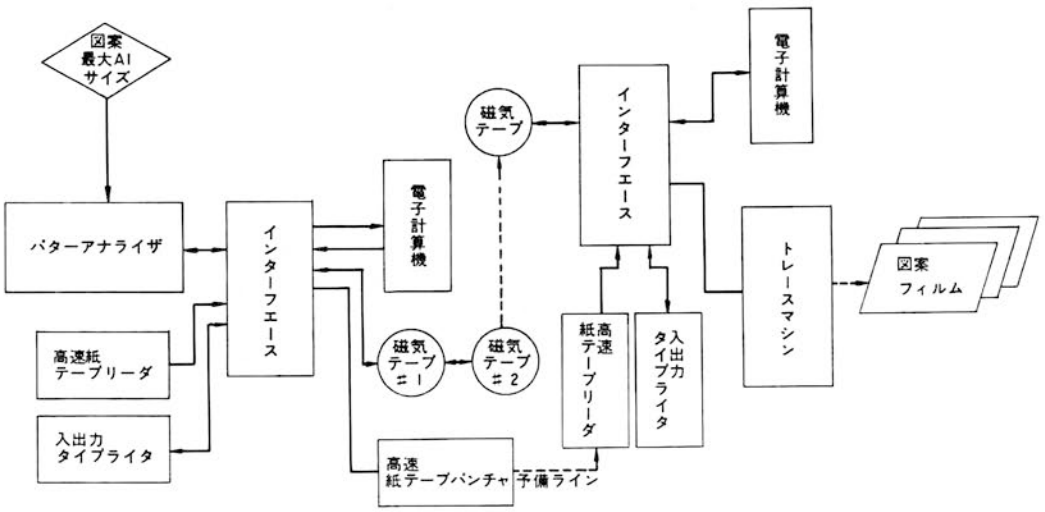
図柄分析装置で出力された柄データを入力することによって自動的に捺染用フイルムを作成するシステム。

柄　色　数：任意の16色以内

フイルムサイズ：839㎜×594㎜以内

1日当りのフイルム作成：96枚／8時間

システム構成



❻計測制御システム

《データ集録自動処理システム》

概　要

本システムは，分析機器，計測器，その他の計器類より集録した多チャネルにわたるアナログデータを，CEC 555H電子計算機を中央演算処理装置とするミニコンピュータシステムにより自動的集中的に集計，整理，解析，演算処理をおこない，あらかじめ定められたフォーマットに従って印字，作表，記録あるいは結果のパネル表示，警報の発生など一連の処理を自動的におこなうシステムである．

本システムの運用により，研究機関の実験データの処理，工場の検査部門，品質管理部門，公害関係などにおけるデータ処理，資料の収集作業など，従来多くの日数と多人数の延人員を必要とした該当部門の業務が大幅に省力化され，精度面でも飛躍的な向上が期待できる．

中央演算処理装置に使用されるCEC 555H電子計算機は高信頼性と機動力のある拡張性にすぐれた電子計算機システムを構成している．

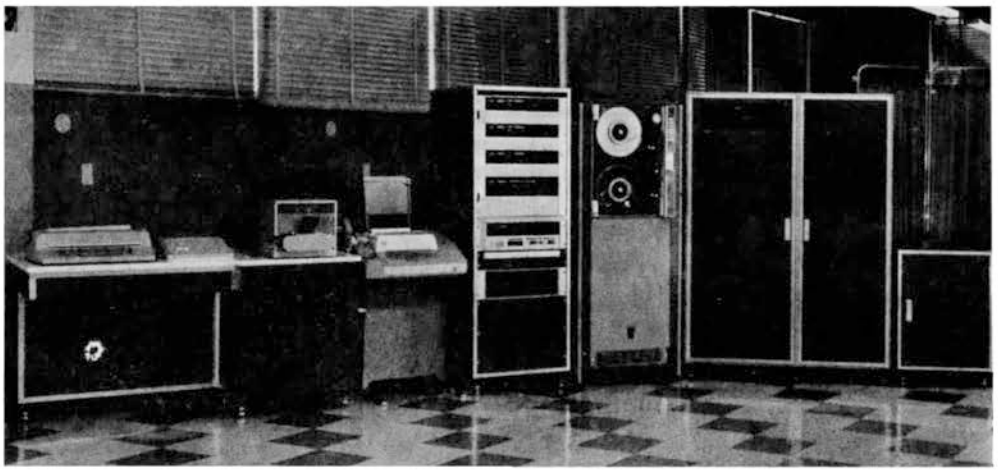
特にCEC 555H電子計算機はマルチプロセッシングシステム用として開発されたもので多チャネルのデータ処理が容易におこなえる．

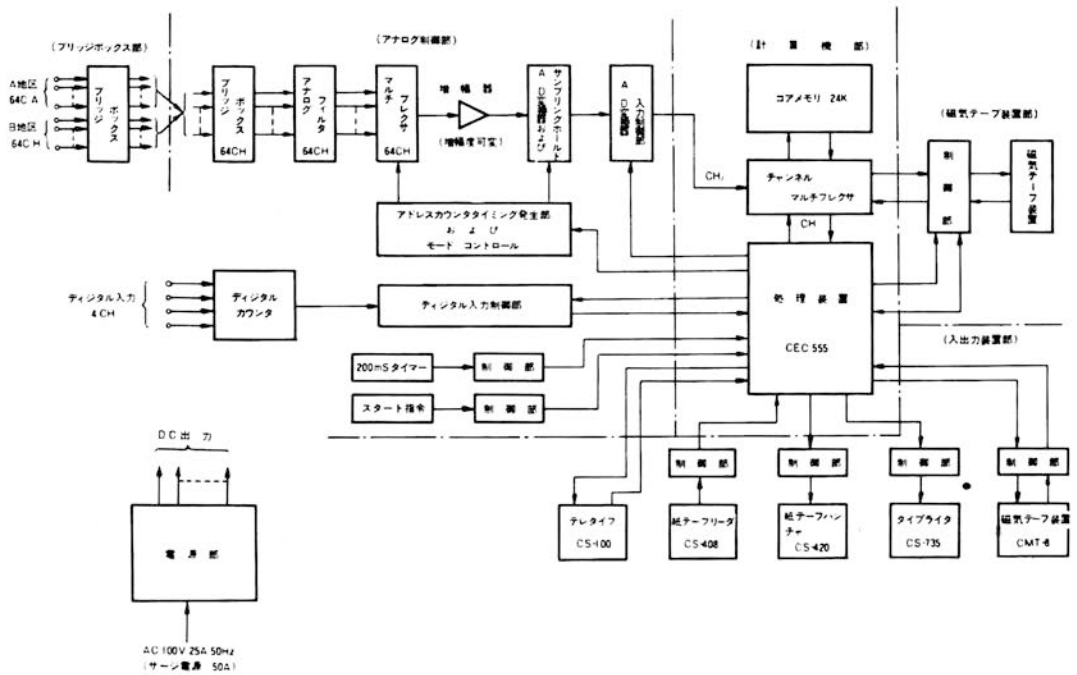
特　長

1. 多量のデータを高速で集録，処理が可能．
2. データソースが複雑で，多点数の処理が必要の場合でも容易に処理できる．
3. データソースが複雑に変化する場合でもプログラムにより適確な処理ができる．
4. 演算処理のデータをリアルタイムでデータソースにフィードバックできる．
5. 修正計算を含めた測定処理が容易．
6. 出力データのファイル，検索が簡単．
7. ソフトウェアが完備している．
8. 信頼性が高い．

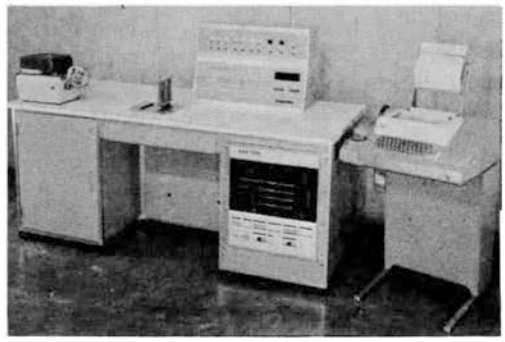
システム構成

1. ブリッジ　ボックス……1式
2. バッファ　アンプ……64台
3. アナログフィルタ……64台
4. アナログマルチプレクサ（標準64CH）……1式
5. 前置増幅器……1台
6. 基準電源……1台
7. A／D変換器……1台
8. A／D変換器入力制御部……1式
9. アドレスカウンタ　タイミング発生部およびモードコントロール部……1式
10. 200msタイマ……1式
11. ディジタル入力カウンタ……4CH
12. ディジタル入力制御部……1式
13. スタートマーク検出部……1式
14. コンソール部……1式
15. 中央演算処理部（CEC 555H電子計算機）……1式
16. 高速紙テープリーダ（制御部含む，CS-408高速紙テープリーダ）……1台
17. 入出力タイプライタ（制御部含む，CS-100入出力タイプライタ）……1台
18. 高速紙テープパンチャ（制御部含む，CS-420高速紙テープパンチャ）……1台
19. タイプライタ（制御部含む，CS-735タイプライタ）……1台
20. 磁気テープ装置（制御部含む，CMT-8磁気テープ装置）……1台
21. 電源部……1式




(フリッジボックス部)
A地区
64C A
B地区
64C H
フリッジ ボックス
フリッジ ボックス 64CH
アナログ フィルタ 64CH
マルチ プレクサ 64CH
(アナログ制御部)
増幅器
(増幅度可変)
サンプリングホールド および
A D 変換部
A D 入力制御部
(計 算 機 部)
コアメモリ 24K
チャンネル マルチプレクサ
CH
CH
(磁気テープ装置部)
制御部
磁気テープ装置
アドレスカウンタタイミング発生部 および モード コントロール
ディジタル入力 4CH
ディジタル カウンタ
ディジタル入力制御部
処理装置 CEC 555
(入出力装置部)
200mS タイマー
制御部
スタート指令
制御部
制御部
制御部
制御部
制御部
テレタイプ CS-100
紙テープリーダ CS-408
紙テープパンチャ CS-420
タイプライタ CS-735
磁気テープ装置 CMT-6
DC 出力
電源部
AC 100V 25A 50Hz
(サーン電源 50A)

《プリント基板自動試験システム》



概　要

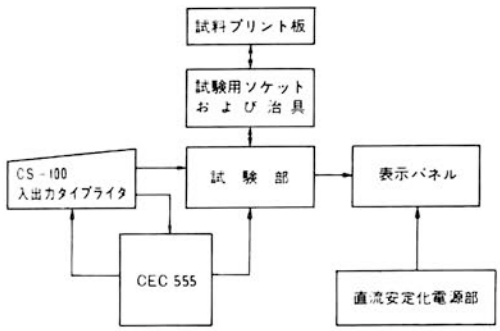
本装置は，各種ディジタル応用機器用の論理回路モジュール，たとえばパッケージ（プリント板）IC，LSIなどの機能試験をCEC 555電子計算機により自動的におこない，良否（GO/NO GO）の判定結果をパネル上に表示，またはタイプライタに印字するシステムで，プログラムを用意することによりあらゆる項目の試験が自動的におこなえる．

試験は，試験対象となるICやモジュールに対して用意されたプログラムに従い，次のようにおこなわれる．

試験プログラムをCEC 555に記憶させ試料をソケットまたはハンドラにセットして，STARTボタンを押す．スタートとともに，同時に試料に試料電源が供給され，電子計算機のプログラムに従って

1．試料の入力ピンに，プログラムで指定された入力記号パターンを加える．
2．入力パターンを加えると同時に，試料の出力ピンからの出力信号パターンとプログラムによって与えられた正しい出力パターンとを比較照合し，誤りがあればそのピンに対応した照合結果レジスタのフリップ・フロップを1にセットし（このレジスタの各フリップ・フロップの状態は表示パネルにランプで表示されている），同時にGO，NO GO信号を発生し，ランプ表示する．
3．停止命令（STOP）を読み取ると試験を停止し試験電源をOFFにする．

なお，試験中NO GOが出るとただちに試験を停止させるためのFAIL STOPスイッチ，テープ中の停止命令を無効にするためのAUTO STOP ON-OFFスイッチなどのほか入出力パターンを1ステップずつ進める押しボタンなどがついている．また，試料のピンの入力・出力および不要なピンの指定は，操作パネル上のピンボードでおこなう．



LSIテストシステム



《圃場最底温度予報システム》

概　要

特定圃場における日没前後の気象要素を測定し，熱収支式から導いた最低気温予報式により翌日の最低温度を予報するシステムで，同時に気象要素を継続的に整理記録するものである．データ入力は56チャネル設けてあり，それぞれについて気象データ信号レベルに適合したアナログアンプにより一定レベルに変換した後，アナログマルチプレクサ部より信号をとり出し，A／D 変換をおこないディジタル信号にし，定められたフォーマットで磁気テープに集録し，同時に目的に従った種々の計算処理をおこなうシステムである．主な処理目的は①，風による熱の輸送の測定，②気象要素をディジタルデータに変換し，記録する，③変動の激しい気象要素や，遅く変化する気象要素をそれぞれの変動に合致した時間間隔での測定，④測定したデータの記録と同時に計算処理をおこない最低温度の予報値の算出，⑤その他の気象データの記録・分析などであり，圃場の他に各種の農芸に関する気象条件の記録，分析に利用することができる．

測定対称の入力信号とアンプの出力電圧

	入力チャネル	測定間隔	入力電圧(MAX)	出力電圧
気温(半分は湿球)	26点	1 ms	1 mV	10 V
風向・風速	6点	1 ms	± 10V	± 10 V
表面温度	1点	10ms	100mV	10 V
地温	11点	1 s	10mV	10 V
放射	4点	1 s	± 100mV	± 10 V
地中熱流	2点	1 s	± 10mV	± 10 V
土じょう水分	2点	1 s	1 mA	10 V

システムは，図の様に計測装置(PU ＃1)，チャンネルマルチプレクサ（CH)，記憶装置（メモリ16Kワード)，から構成されている．各装置は全て独立しており，全ての算術演算，論理演算，システムの制御指令は CEC 555 H で行なわれる．なお，CEC 555 H と周辺装置のインターフェースとの，プログラムによる個々のデータの転送は，アキュムレータ(AC)経由で行なわれ，A/D コンバータ，磁気テープ，メモリ，リアルタイムクロックの転送は高速のバスライン方式で行なわれる．

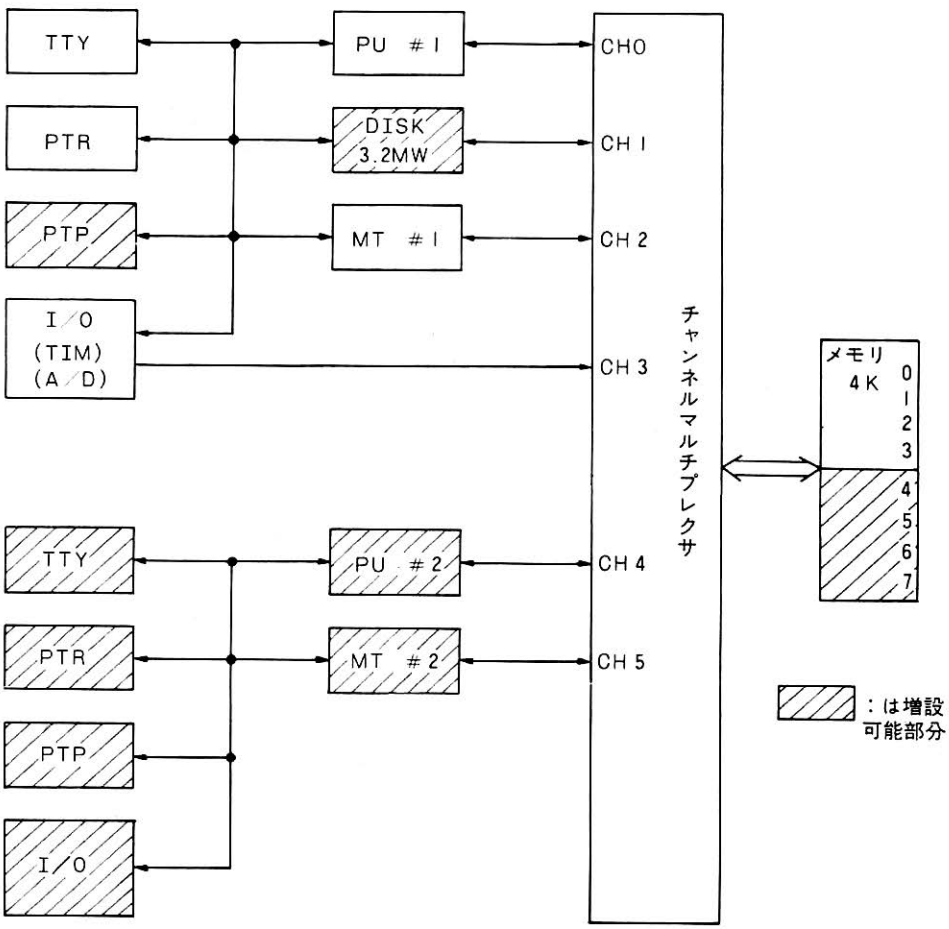
動作原理

データインプット方法

入力レベルの異なった56チャンネルのアナログ信号をアンプ及びダイレクトで 0～±10Vに増幅する．増幅した56チャンネルのアナログ信号は，マルチプレクサで必要チャンネルを選択し，1チャンネルづつ取り出し，サンプルホールドしてA／D変換器でディジタル信号(BIN, 12B IT)に変換され，演算装置内のディジタルマルチプレクサを通して記憶装置にストアされる．
また，入力データと同時にリアルタイムクロックも記憶される．
計測装置の記憶データは，フィールドごとにふり分けられ，データ入力，データ出力をCEC555Hでコントロールし，データ出力は CMT-8磁気テープに転送される．

システム構成




TTY
PTR
PTP
I/O
(TIM)
(A/D)
PU #1
DISK
3.2MW
MT #1
CH0
CH1
CH2
CH3
チャンネルマルチプレクサ
メモリ
4K
0
1
2
3
4
5
6
7
TTY
PTR
PTP
I/O
PU #2
MT #2
CH4
CH5
：は増設
可能部分

《振動分析自動処理システム》

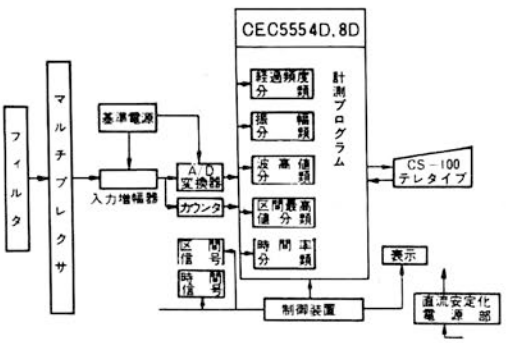
概　要

本システムは振動・歪みなどのデータをアナログ波形入力とし、CEC 555電子計算機によりオンラインで自動的に分類、演算処理する装置である．分類項目は経過頻度，振幅，波高値，区間最高値，および時間率などである．



システム構成

1. 入力増幅器
2. スライサ
3. 基準電源
4. 分類器
5. 計数器
6. 走査器
7. 表示器
8. 直流電源
9. 制御パネル
10. 区間，時間信号発生部
11. フィルタ
12. CEC 555電子計算機



主要仕様

入力増幅器…入力インピーダンス　10kΩ一定，利得　0～10，0～1，0～0.1倍，入力電圧　0～0.4，0～±4，0～±40V，周波数　0～10kHz±0.01%，ドリフト　0.01V／H

波形モニタ…ブラウン管　3インチ，垂直感度　0.01V，0.1V，1V，10V／Div，入力インピータンス　100kΩ，50pF以下，周波数　DC～100kHz　+0.5dB，－3dB，掃引周波数　1Hz～10kHz

電　源……入力　AC100V±10%，50／60Hz，出力　＋5V±0.5%（理論回路用），±24V±1%（アナログ回路用），リップル1mV$_{p-p}$以下

区間信号……出力　0.1，0.2，0.5，1秒，精度　±0.005%

時間信号……クロック信号　20kHz，10kHHz，5kHz，出力　5，10，29秒，精度　±0.005%

分類器……測定範囲周波数　DC～1kHz

計数器……桁数10進6桁，段数　12，カウント周波数　1MHz以下

走査器……チャネル数　12，速度　約0.5秒／チャネル

入力信号とスライスレベルとの関係

入力信号電圧＞スライスレベル
判定〈1〉
入力信号電圧＞スライスレベル
判定〈0〉

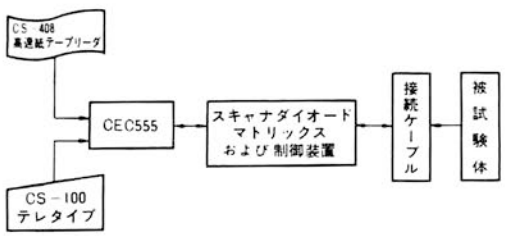
$$\text{スライスレベル} = \frac{\text{スライス電圧}}{\text{入力増幅器のゲイン（倍）}}$$

周辺温度：0～40℃

《自動配線試験システム》

本装置は電子計算機，電子交換機，情報処理周辺装置など，プリント基板を大量に用いる装置において端子間の配線の余配線と未配線の有無を自動的に短時間で検査し，かつ余配線や未配線があった場合は自動的にその“配線順序番号”“FROM端子”“TO端子”番号をそれぞれタイプライタで印字する．装置は“配線順序番号”“FROM端子”“TO端子”など布線情報がせん孔された紙テープリーダで読み取り自動的に論理演算制御しダイオードマトリクス群などを利用して純電子的に検査するもので，試験担当者は安心して安全な配線検査を完了することができ，特に手作業では発見困難な余配線も容易に検出できて円滑な検査業務ができる．余配線または未配線があった場合，修正すべき情報がタイプライタに直ちに印字されるからリアルタイムで補正，修正することができ，そのあとさらに本装置で検査し，合格“GO”を確認して次の作業工程に送ることができる．





システム構成

1．制御装置
- (1) CEC 555電子計算機
- (2) 制御部
- (3) ダイオードマトリクス部
- (4) 直流安定化電源部

2．入出力装置
- (1) 紙テープ読取機
- (2) テレタイプ
- (3) 入出力制御部

3．検査用治具
検査用プリント板（リード接栓付）

主要仕様

表示：各機能，動作に基づいて表示

マトリクス駆動部

制御部：メイン制御，PTR読取制御，印字など

検出マトリクス：誤配線検出は8.020点検出可能

システム構成

マトリクス駆動：５Ｖ信号を入力として駆動
S側マトリクスドライバ，入力信号　＋５Ｖ4mA以下，出力信号　＋６Ｖ600mA以上
R側マトリクスドライバ，入力信号　＋５Ｖ4mA以下，出力信号　－６Ｖ30mA以上
立下り時間　500μ以下

センスアンプ部

センスライン：8.020個のマトリクスを駆動した結果をとり出す．OR回路を経て最終１センスラインにまとめる．

CCD-5GH高速カラーグラフィックディスプレイによる

《選挙速報システム》

このシステムは，CEC555Hミニコンピュータとセンタシステムとの間でデータの交換をおこない，CEC555Hミニコンピュータによりモデム回線の制御とグラフィックディスプレイの制御とおこなってセンタシステムから送られるデータの解析，処理，実行結果の表示をおこなうシステムである．

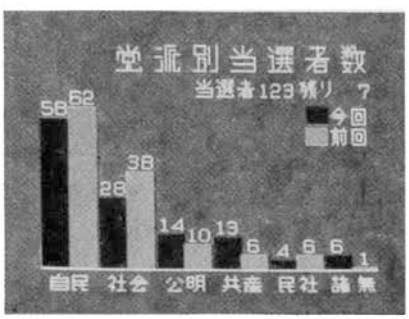
本システムにおける放送用の出力は RGB信号および，NTSC信号の2種であり，モデムで接続された両システム間のデータ転送スピードは4800BPS である．

ディスプレイシステムの構成図，および構成機器は次の通りである．

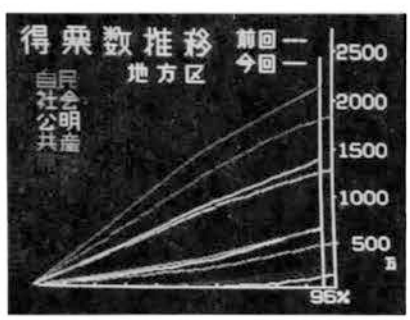
① 小型電子計算機……CEC-555H
② グラフィックディスプレイ……CCD-5GH
③ モデム……DATAX 4804M
④ コンソールタイプライタ……CS-100(ASR 33)
⑤ 光学式紙テープリーダ……CS-408
⑥ カラーエンコーダ……CC-2A
⑦ 電源及び収容筐体一式

CCD-5GH の特長

① TV 方式が標準方式であるため表示部が容易かつ LOW COST である．
② カラーカメラ用のエンコーダと組合せ，NTSC カラー映像信号を得られる．
③ 外部同期可能であり，スーパー，特殊効果等他の映像との合成により情報源を拡大したかたちでのディスプレイが出来る．
④ キーボードおよびサムホイールコントローラを使用して(オフライン)絵を描くことが出来る．(DISPLAY 単体の機能チェックが簡単に出来る．)
⑤ リフレッシュメモリを内蔵しているため，CPUはディスプレイ用の BUFFER を特に要しない．
⑥ またリフレッシュメモリーの内容を自由に読み取ることが出来る．
⑦ 入出力信号はそれぞれ8単位（8単位目は予備……パリティビット等に使用可）であり，計算機と接続のための制御装置を非常に簡単にしている．

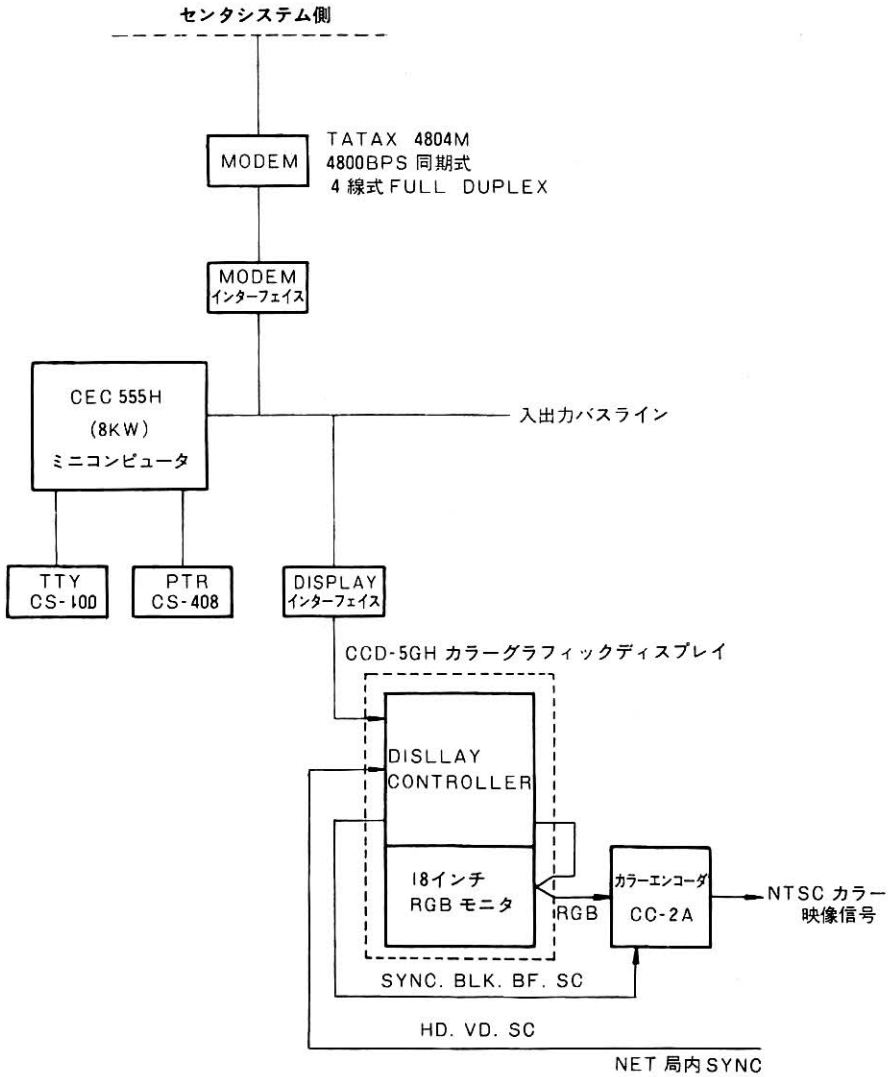


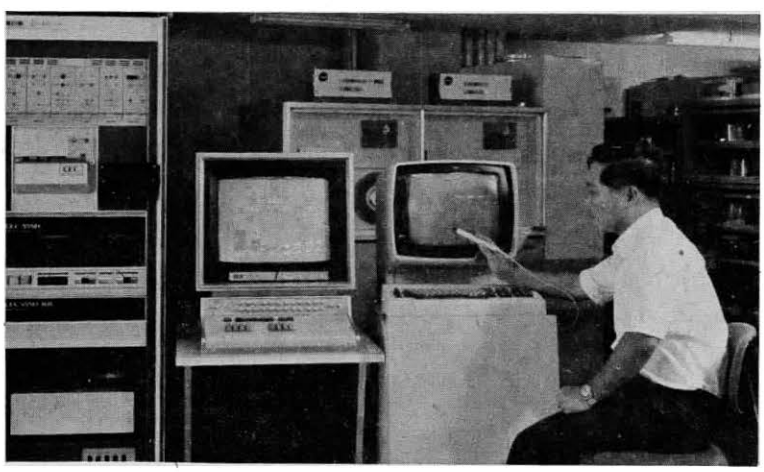
表示例-1　党派別当選者数の棒グラフ



表示例-2　得票数の推移の折線グラフ

システム構成


センタシステム側
MODEM
TATAX 4804M
4800BPS 同期式
4線式 FULL DUPLEX
MODEM
インターフェイス
CEC 555H
(8KW)
ミニコンピュータ
入出力バスライン
TTY
CS-100
PTR
CS-408
DISPLAY
インターフェイス
CCD-5GH カラーグラフィックディスプレイ
DISLLAY
CONTROLLER
18インチ
RGB モニタ
RGB
カラーエンコーダ
CC-2A
NTSC カラー
映像信号
SYNC. BLK. BF. SC
HD. VD. SC
NET 局内SYNC



❼自動車エレクトロニクス

《ドアー耐久試験機制御システム》

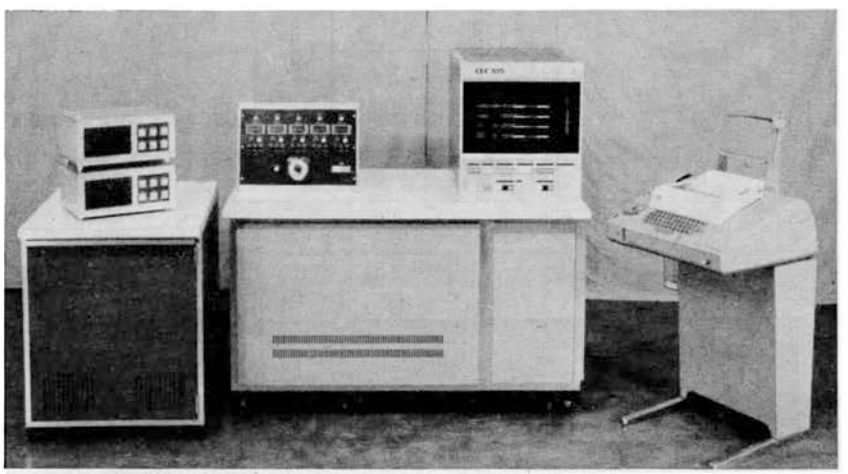
概　要

本装置は，ドア耐久試験機類の操作，計測系を自動化し，全試験機の集中制御を行ない，試験機運転上の無人化を目的とするシステムである．

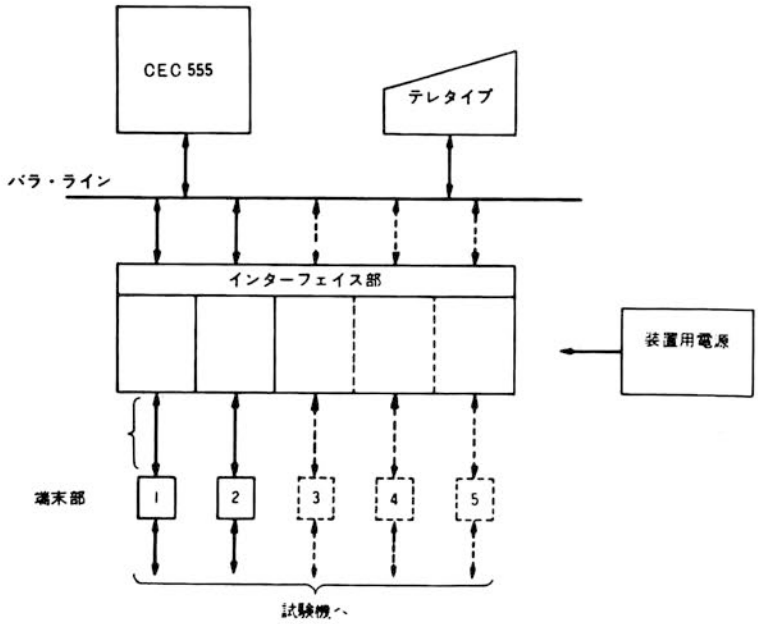
操作，計測等の指令判断は，すべて CEC 555 ミニコンピュータで行ない，ソフトウエアによって運用され，総計 5 チャンネルの端末部を各チャンネル毎に，独立に試験することが出来る．

構　成

(1)　CEC 555-4
(2)　CS-100 入出力タイプライタ
(3)　インターフェイス部
(4)　端末部
(5)　装置用電源部

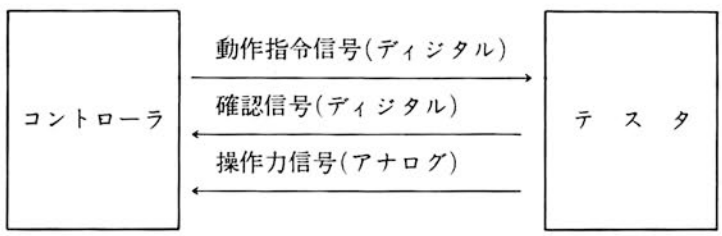


システム構成



機　能

本装置(コントローラ)は，試験機(テスタ)との間で，3系統の信号の交換を行なう．



(1)　動作指令信号

テスタに動作の指令を与える信号で，コントローラの電磁弁ドライバによって，テスタの電磁弁をON-OFFし，被試験体に機械的な動作をさせる．

(2)　確認信号

(1)の信号によって生じた動作が，正しく行なわれたか否かを，コントローラで確認するために，テスタで発生させる接点信号である．

(3)　操作力信号

(1)によって生じた動作により，被試験体の各部に発生する操作力をストレンゲージ等によって電気信号に変換した信号であり，これはコントローラによってAD変換の後，処理される．

このような，基本的な信号のやりとりを，各種組み合せることによって(テスト・パタン)，複雑な機械系を動作させることができる．また，これ等は各CH毎に独立に設定できる．

CEC555システムの自動車産業における応用事例



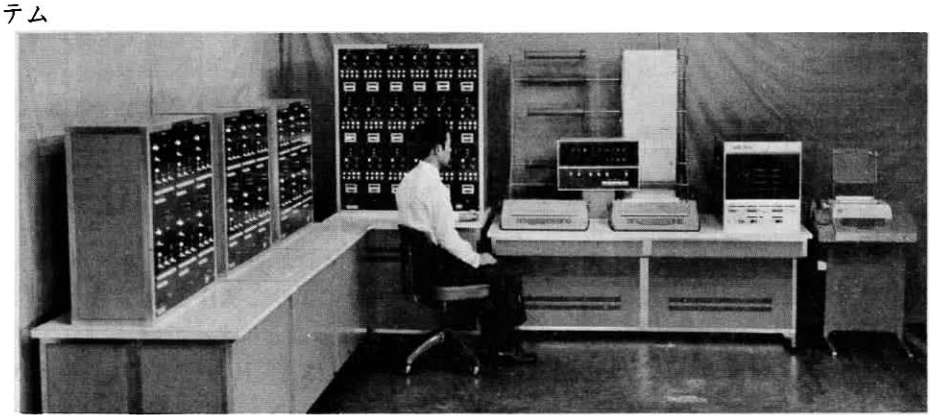
車輌性能データ処理システム

車輌性能を測定し，ミニコンピュータにより定められた補整演算をおこなったのちX－Yプロッタに結果を作図するシステム



検査データオンラインシステム

定められた基準値に対する製品の立体的寸法・精度について総合的にチェックをおこなうシステム



耐久試験装置　自動車部品の耐久性をテストするシステム

⑧文教システム

① 豊富で廉価な情報処理教育用周辺装置　拡張性をもった高性能のハードウエアシステム

豊富な実績をもつ高性能ミニコン CEC 555（記憶容量4,000語～32,000語）を中央処理装置とし，これに教育用に適切な入出力タイプライタ，万能入力端末装置コードキーヤ高速紙テープ読取装置，高速紙テープせん孔装置，磁気ドラム装置，磁気テープ装置，カセット磁気テープ装置，ラインプリンタ，カラーキャラクタディスプレイ，グラフィックディスプレイ，XY プロッタ，各種制御用インタフェイスさらに教育用表示装置などの豊富な周辺装置が，しかも廉価に接続して使用でき，これらが総合されて情報処理教育用の理想的ハードウエアシステムをなしており，かつ予算額に応じ，350万位から2,500万円位の各種のシステム構成が用意されている。

② 整備された豊富で体系的な情報処理用ソフトウエアシステム

情報処理教育の成否をきめるのは，ハードウエアシステムとともにそれ以上にソフトウエアシステムが重要な要素である．CEC 情報処理教育システムはオペレーティングシステムモニタII，フォートラン，RPG（Report Program Generator）などの基本言語は勿論豊富なシステムプログラムを揃えているうえ，各種のサービスルーチン，サービスプログラムライブラリが用意されている．

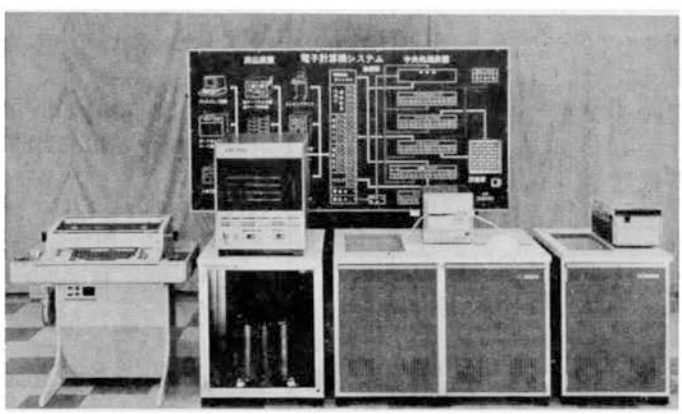
③ 体系だった教育方法と体系だった学習テキストなどの豊富な教材

CEC のコンピュータのハードウエア，ソフトウエアに習熟するための教育ではなく，CEC のシステムを用いてコンピュータの構造と機能原理，コンピュータによる情報処理の基本原理と基本技術を習得させるための教育方法がたてられており，しかもこれを実施していくためのテキストが用意されている．

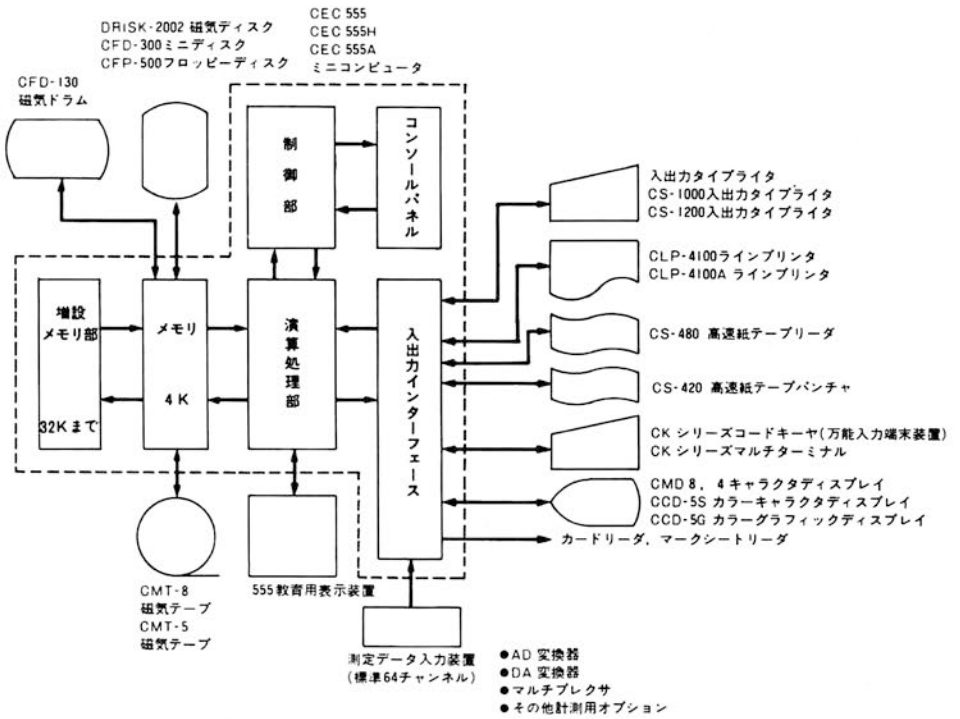
CEC 556 教育用電子計算機表示装置

黒板大の表示装置は実際のプログラムを電子計算機にかけその動きを3段階に切換えて，適した速度で学習のできる独特の表示装置．電子計算機の内容は10メートル以上離れた位置で通常の視力の人が容易に読みとれるように表示される．論理演算の過程は通常の速度のほかに，(1)約 0.3 秒/1ビット，(2)手動(1ステップごと）の 3段階に速度が切換えられ，教育のレベル・目的に適した速度で観測できる．ただ通常速度の場合には過程は判別できない．

※その他の構成装置については，ハードウエア編，周辺装置編を参照のこと．



教育用に利用できる機器



教育用に利用できるソフトウェア

情報処理教育用ソフトウエアシステム
- オペレーティングシステムII-1，II-2
- ベーシックアセンブラ
- マクロアセンブラ
- 対語型 FORTRAN（COCOL）
- 4K FORTRAN
- 8K FORTRAN
- COBOL
- R. P. G（Report Program Generator）
- 各種ユーティリティプログラム
- ドラムオペレーティングシステム
- 各種事務処理用ライブラリ
- 各種科学計算ライブラリ
- 各種制御用専用コンパイラ及アプリケーションライブラリ

経営数学・統計解析プログラム各種

教育用言語システム
教育用サービスルーチン
教育用ユーティリティプログラム
教育用サービスライブラリ

標準カリキュラムと学習テキスト・教材
（大学，短大，工専用）
電子計算機入門，プログラミング学習
FORTRAN COBOL
ASSEMBLER，システムプログラム学習
値計算プログラミング
事務処理プログラミング
制御用プログラミング
システム計算法
（工業高校用）
プログラミング，電子計算機，数値計算法
プログラム理論，システム工学
（商業高校用）
電子計算機一般，プログラミングI
プログラミングII，**経営数学**，システム設計
（共通教材）
コンピュータシステム黒板，プログラミング教育用黒板，標準テンプレート
標準フローチャート用紙
コーディングシート

その他

マネージメント編

マネージメント編目次

1．CEC555電子計算機システムについて …… 785
2．CEC医療情報システムについて …… 787
3．CEC　EDPシステムについて …… 788
4．システム障害対策，および保守・サービス体制 …… 790
5．販売代理店 …… 794
6．リースおよびローンについて …… 796
7．CEC中央電子株式会社の概要 …… 799
8．関連会社について …… 818

付録A．売買契約書 …… 820
付録B．CEC装置保守，サービス基本契約書 …… 825

1．CEC555電子計算機システムについて

1－1　概　況

CECは，昭和35年設立以来，一貫してエレクトロニクスを媒体として高度に発達した個別の科学を連携するシステムの専門メーカを目指してきた．これは，近年情報処理システムと呼ばれるようになったが，CECは，情報処理システムの専門メーカーとして既に16年間重要な実績経験を積んできたといえる．ちなみに，CECの安定化電源，パルス発生器，パルス応用装置，アルミ筐体は情報処理システムの基礎技術として不可欠のものであり，専門業界には広く知られている．

昭和45年に至って，TTL集積回路の出現と電子計算機の方式が確立したのを契機としてCEC555電子計算機を開発した．当時から，今後10年～20年間，産業社会の広域分野にわたり大きな利用効果があるとの確信をもとに，着々と各種の情報処理システムに実績を記し，充分なサービスを指針として今日に至っている．したがって，汎用電子計算機メーカーとしては，知名度は必ずしも高くないが，特殊な情報処理システムを指向するか，もしくは高度の情報処理システム技術を指向する本格派ユーザーには高い評価を受けている．つまり，情報処理システムについては，従来のような電子計算機概論的商品概念としてではなく，電子計算機の理論と現代の周辺装置群および集積回路等電子部品群の機能，品質の技術革新の可能性および利用分野の実際的需要等を包括して，長期的に的確な判断をすることのできる専門家には広く認識されている．

現在，純国産のミニコンおよびその応用システムの専門メーカーとしては本邦唯一であり，通産省の重要技術研究補助金および日本開発銀行より高度化資金の融資をはじめ，情報処理振興協会の保証による日本興業銀行よりのソフトウェア開発資金など政府資金の融資を受け成果をあげている．これら一連の通産省の制度融資を受けていることは，これからの広域の産業社会における情報化社会への移行，知識集約形産業構造への転換をはかるため，CEC555ミニコンとその応用システムが各方面で活躍し，ユーザーがそれぞれ満足している実績を客観的に認められている結果であり，将来にわたる大きな期待が寄せられている．

1－2　特　長

1．徹底したユーザーオリエンテッドの情報処理システムをつくる．

従来，メーカーの方法はメーカーオリエンテッドで商品をつくってきたため，レンタルバックという言葉に象徴されるごとく使いすてや目標とする利用効果があげられない情報処理システムが多かったきらいがある．

CECはCEC555ミニコンをつくるのと同時に多種の周辺装置を開発し，ユーザーの期待を第1順位として，ユーザーの将来計画を含めて，システムの最適設計を目標として過去5年間に多数の応用システムを開発してきた．したがって現在CEC555ミニコンシステムは，実名でワンタッチでデータ入力を可能としたCKシリーズコードキーヤ，紙クズを出さないCMD-8ディスプレイ装置，最も単価の安いDRISK2002磁気ディスク装置などに代表される豊富な標準周辺端末群を用意して，自由自在にユーザーのニーズに応じて最適な情報処理システムを供給する態勢を整えている．

2．長期利用に耐えられる対策を完了している．

従来，ユーザーの業務拡大や時の流れによって新しい利用目的があったときユーザーの期待に添えない例が多かった．

新しい周辺装置が出現しても，業務拡大があってもCEC555ミニコンシステムは充分な機能を備えている．

特にCEC555Hミニコンシステムは新しい周辺装置や，新しいLSIが出現しても接続しやすくユーザーにとって今後5～10年間はこのシステムが必要充分条件を備えていると確信をもってい

る．加えてこれ以上小型化しても，手が使えなかったり逆の効果も予測されるところから，現在時点から積極的な普及に乗り出したものである．一方産業社会全般にわたり製造経費，販売経費ともに上昇している現今，早期に利用効果をあげる必然性もある．

3．　各事業分野で共用できる．

容量が適切ならば，ユーザーの各事業分野で共用できる．また，CEC555ミニコンシステムに機種統一すれば，営業部門と製造部門でも長期的に標準化して利用できるため，組織の分断が生じないし，ソフトの開発効率および投資効率ともに相乗効果があげられる唯一の機種である．ちなみに制御システム，オンラインシステム，計測システム，事務EDPシステムの多方面の豊富なシステム実績がある．（主要実績表参照）

4．　故障が極めて少なく，信頼性が高い．

CEC555ミニコンシリーズは業界中最も多機種完備していて，周辺装置群も多種あり，加えて筐体から電源，プリント基板まで一貫してCEC技術部で設計生産されていることと，設立後16年来わが国電子工業の代表的研究機関である日本電信電話公社通研，電総研，国鉄技研，NHK技研等に納入実績も多いことから，基礎技術が確立していることが明らかであり，高いシステム技術力と応用システム設計の機動力がユーザーに広く認識されている．

5．　サービスが良い．

今までの広いユーザー層全般にサービス態勢の評判がよい．

2. CEC医療情報システムについて

2—1 概況

昭和35年設立以来，一貫してエレクトロニクスを媒体として高度に発達した個別の科学を連携するシステムの専門メーカーを目指してきた．現在，純国産のCEC555Hシリーズ，CEC555SSシリーズミニコンピュータおよびその応用システムメーカーとしては本邦唯一であり，CEC552心電図自動診断装置，CEC540脳波自動診断装置，CEC512血液生化学自動分析装置，CEC普及形健康管理センターシステムなど，国が定めた開発テーマの主力メーカーとして指名され，長期にわたる行政官庁・医学専門家との協力プロジェクトとして完了して実のある普及成果をあげて現在に至っている．

CEC555H，CEC555SSに代表されるミニコンピュータも通産省の情報処理振興政策の一環として指名され，日本開発銀行により政府資金の融資を受けて開発し，知識集約形，省資源形産業構造への移行の担い手として各方面にその応用システムを普及して，成果をあげている．

従来，医療情報システムは，個々の機器を寄せ集めて単に組立てたものが多く，大味であり，ユーザー専門家の批判を受けているが，CECの医療情報システムは，すでに10年以上，上記のごとき医療情報システム構成上重要なシステム装置群について慎重に研究実績を重ねて完成したものと，一貫生産によるミニコンピュータ技術を適切に利用して，かつ将来5年～10年までユーザーの発展を想定して拡張や増設が自由にできるような準備ができているところに特長がある．

2—2 特 長

1. 医療情報システムの重要主構成要素

   CEC555H，CEC555SSミニコンピュータシステム

   CEC552心電図自動診断装置

   CEC540脳波自動診断装置

   CEC512血液生化学自動分析装置

   などであり，いずれも一貫生産しているから，将来にわたり，生産中止や変更によるサービス障害，拡張・増設の場合の障害などがなく，ユーザーへの行き届いたサービスを保証している．

   従来，一般に事業部がそれぞれ異なりいったんシステムを完了しても事業部が独立しているため，充分なシステム技術の利点を活用できないままのものが多いが，CECは一貫生産しているメーカーであるため，充分なサービス体制を敷いている．

   したがって，一般事務用民需向のコンピュータシステムは大手メーカーの実績が多く，CECの実績よりはるかに大きいが，国家レベルの新しいコンピュータシステムの開発テーマは対等以上の実績が専門家筋には高く評価されている．CECはこのような専門メーカーであるから，当座の目標だけでなく，将来にわたり有効に利用していこうとするユーザーは，CEC情報システムを信頼して採用することができる．

2. 医療機器については，本邦の業界の努力によって多様な機種が完成しているため，ユーザーの志向するものを自由に選択し，システムとして組み込むことができる．したがって，自社製のものを無理に組み込まないことがユーザーに好評である．

3．CEC EDPシステムについて

3－1概　要

現在，人件費の上昇，公共料金，租税公課等諸経費の上昇は，相当企業経営を圧迫している．そして，今後も更に上昇する趨勢にあることは否定し得ない．したがって，今直ちに事務管理部門と販売部門をもう一度見直してこれから長期にわたって，コスト上昇を吸収し，かつ安定した利益を確保する対策を実施することが必要である．

そのためにはCEC EDP SYSTEMを採用して関連部門全員が気持よく参加して抜群の利用効果を上げることが最善だと確信する．

CEC EDP SYTSEMの採用にいたる評価は次のとおりである．

3－2特　長

(1)現有職員1人当りの費用の追跡

1人当り年間(1)人　件　費…250万円（直接賃金＋企業負担福利費＋源泉税他）

(2)経　　費…200万円｛商業・特殊法人……一般管理費＋販売費用
　　製造業……製造直接，間接費＋一般管理販売費

(3)利子割引料…　50万円（借入金利息＋割引利息）

計500万円…月当り費用42万円（概算）

CEC EDP SYSTEMを採用すれば，初年度より最低3人～5人の省力が可能であり，省力効果を，これからの最重要業務である**市場確保に集中**することができる．そして市場確保を条件として，3年リース，5年リース換算の月額費用は，3人の場合，42万円×3人＝126万円

5人の場合，42万円×5人＝220万円

この範囲内であれば省力効果と見合うことになる。

そして2年度，3年度と順次組織全体が運用に精通することによって，大巾な省力の相乗効果を挙げることが明らかである．

因みに買取価額，2500万円のCEC EDP SYSTEM20の**5年リース月額**は**約63万円**であり，買取価額，1200万円のCEC EDP SYSTEM10の**5年リース月額**は**約30万円**である．したがって省力効果に対して極めてコストは安い．

(2)全員参加で重複処理のない明るい力のある組織の確立

従来，原始データが発生してからの**日常の処理，週間の処理，月間の処理，通期の処理**を通じて，重複処理が多く，加えて販売活動に直接関係のない大量の転記，累積計算，伝票帖票作成業務が存在していたが，ここで，もう一度洗い直してスッキリしたものとして将来に備える必要がある．実際には原始データの発生時点でデータ入力が「**全員参加で職員格差がなくスピーディに,しかも正確にキーボード,パンチカード,マークシートなど一切の中間媒体を介在せずに完了する**」ことと一旦データ入力が完了したら「**日常の処理，週間の処理，月間の処理および通期の処理は，すべて一事不再理原則でコンピュータによって自動処理される**」ことが必要条件である．

これらの2つの条件を満足するには，CEC CK**シリーズコードキーヤ**により紙製のブックに記入された実名をみながら特殊ペンで，ワンタッチで入力する方法とCEC EDP SYSTEM10～60により一切の自動データ処理を一貫して行なう方法が最善である．

特に欧米の企業の実体が，「**読み書きする人よりもキーボードを打てる人が多い**」のに対して日本の実体は「**キーボードを打てる人が僅かで読み書きは全員が達者**」という特質を活して日々の省力効果を累積することによって，従来と比較して莫大な利益を確保することができる．

さらに，重複や無駄を徹底的に排除することは，全職員をして必然的に本来の重要業務に集中させることになり，健全な愛社心を盛り上げ，明るい力のある組織を確立することができる．

(3)今がCEC EDP SYSTEM採用の絶好期

SYSTEM構成は，最近20年の歴史をかけてめざましい技術革新を重ねてきた高度化した電子工業，とりわけミニコンピュータ，LSI，記憶装置，周辺装置群を結集したもので，現時点が丁度SYSTEM全体のコスト，品質，機能，サービス面で一応の完成期に当っている．そして現在から将来5～10年先を見通した事務管理，販売管理面の理想的利用に対して必要充分条件を具備した段階である．

将来，現在以上にLSIが更に大形化したり，記憶密度の高いものができたり，印刷速度が更に高速になっても利用面で技本的な改善にはならないと思われる．したがって，むしろ早期に採用し，徒らな経費の流出を防止することによって事業利益を確保する積極策を実施する絶好期である．

因みにCEC EDP SYSTEMの構成は，データ入力装置からミニコンピュータ，記憶装置，ディスプレイ，印字装置に至るまで，多様な業務への利用に対して共用性が高く，消耗するものがなく，電子的に高速で自動プログラム処理を行い，用紙も価値の高い必要なものだけに圧縮できる方式を確立しているから一見高価とみられるものの，運用費が安く個別に分析すると極めて相乗効果の高い装置の集合である．

(4)ユーザ利益保護の思想と長期利用の安全性

CEC EDP SYSTEMは従来装置，もしくは，他の製品に比較して徹底したユーザー利益保護の思想を貫いていることが特長であり，特に将来の業務の拡張やユーザー企業の規模と特質に合った機種の撰択に対する入力制御機構についても，安全な対策が準備されていることと，秀れたＯＳが用意されていることが特筆される．したがってCKシリーズコードキーヤ利用のCEC EDP SYSTEMはこれから5～10年の長期にわたる有効適切な事務管理および販売管理組織のトータルシステムを考える上で最適である．

4．システム障害対策，および保守・サービス体制

4－1概　　要

CECは過去16年にわたり，情報処理システムの主要構成諸元である筐体，電源，パルス応用装置，各種データ処理装置を一貫して研究・開発・生産してきた本邦唯一のミニコンピュータシステムの専門メーカである．さらに電子回路系については，信頼性の高いTTL IC化を完了し，多くのシステム製作の経験を積んできており，これらの基礎技術をもとにして，CEC555H，CEC555A CEC555S，CEC555SSをはじめとする中央処理装置，各種周辺装置群，筐体，電源，その他AD変換器，DA変換器，スキャナ，リアルタイムクロック，などの計測用オプション，オンライン接続制御装置にいたるまで，主要な部分はすべて社内で一貫生産していることが特長である．

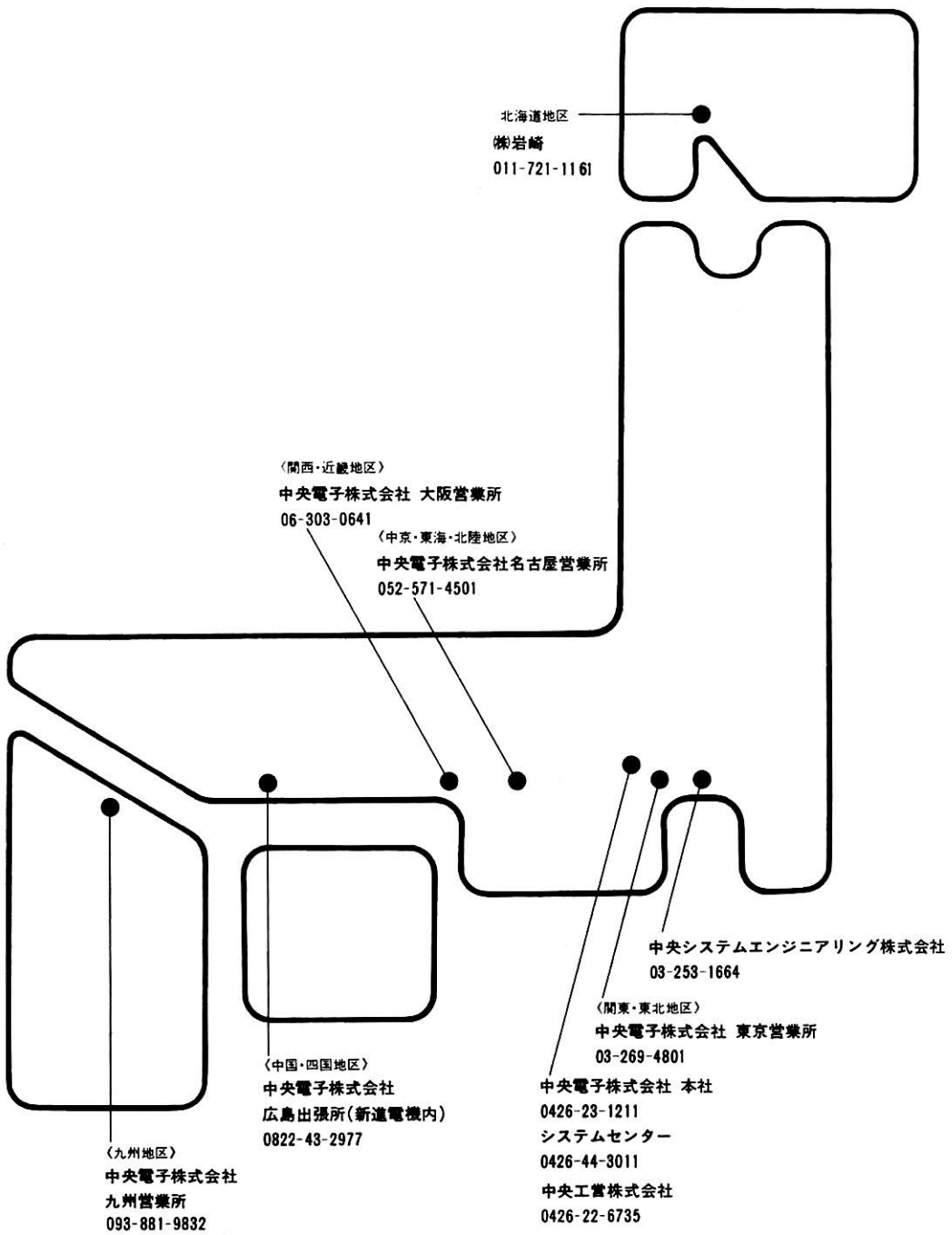
このことは，とりもなおさずシステムが高い信頼性をもつことを裏付けている．またユーザの事業拡大にともなう，機器の増設，システムの拡張や，万が一発生する障害に対しても適確，適切な対策が用意されている．したがってCECの行きとどいたサービスによって，ユーザーは安心して利用効果を挙げることができる．

4－2保守体制

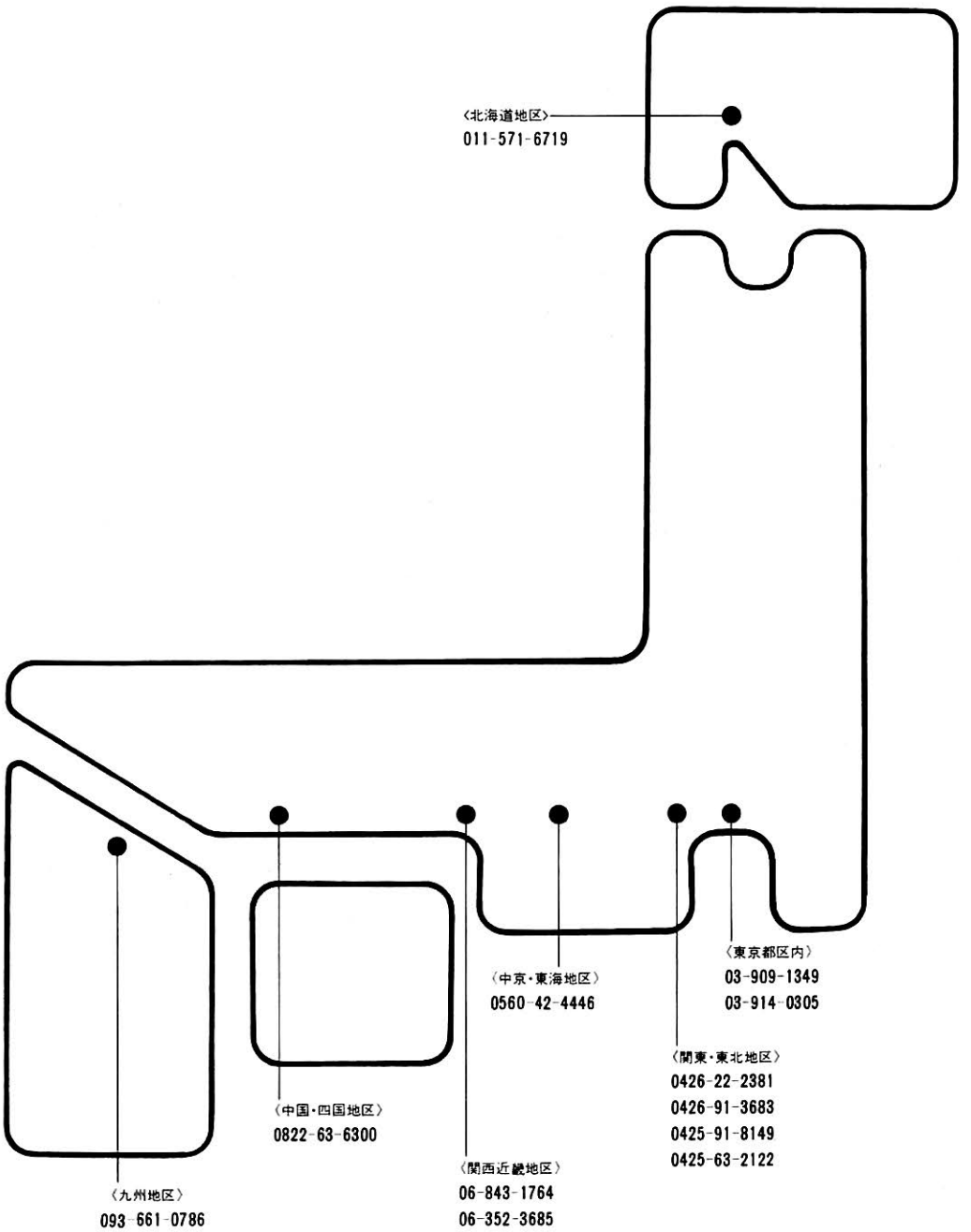
CEC555ミニコンピュータシステムは，数多くの実績を通じてユーザーより高い信頼性を評価されているが，万一発生する障害対策として全国ネットのサービス体制を組織し，不測の事故に備えている．具体的には，各営業所，代理店に補給部品を常備し，ユーザのご理解とご協力を前提として「CEC装置保守・サービス基本 契約（付録2を参照）にもとずく定期サービスを実施している．定期サービスのほか，万一の 障害，故障に対しては，近地においては数時間以内，遠地においても24時間以内の復旧を原則とし，また夜間における障害発生に備え各地域拠点ごとに夜間連絡網を組織しシステムサービスに 万全を期している．

緊急時における連絡網は次表の通りである．

■緊急連絡体制(AM8:30～PM5:30)



■夜間(PM5:00～AM8:30)における緊急連絡



4－3サービス体制

○教育・オリエンテーション

CECではユーザのシステム担当者，システム管理者を対象に講習会をおこなっている．

（1）定期講習会

毎年1月，5月，9月の3回，期間は5日間を原則とし，主要言語の解説，プログラム作成，OSについておこなう．

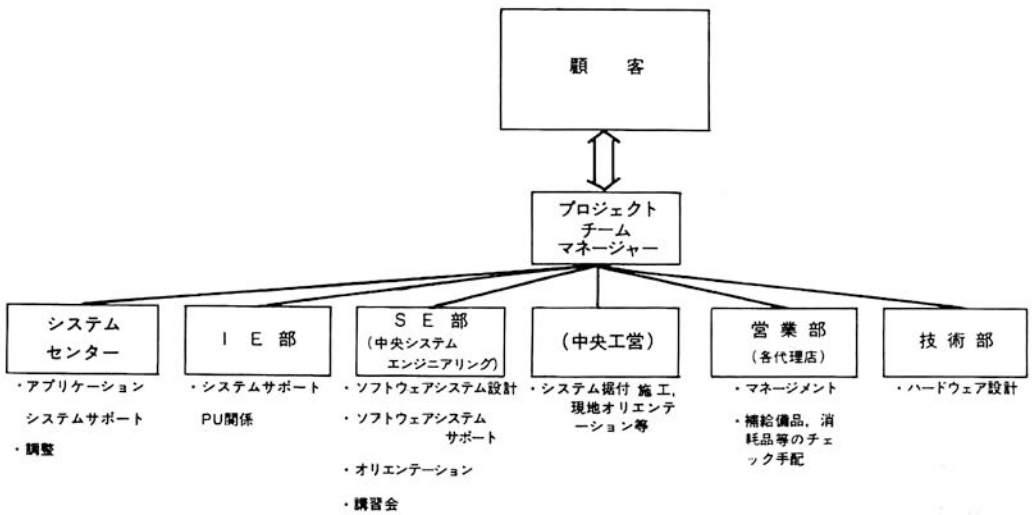
（2）不定期講習会

ユーザの意向により随意おこなう講習会で日時，期間講習内容についてはその都度打合せの上決定する．

※1)，(2)の講習会に関する直接費用は原則として無料でおこなう。

○サービス体制

CECでは契約から納入・設置・調整・オリエンテーション・稼動中のシステムサポート，ソフトウェアサポートに関してシステム毎にプロジェクトチームを編成しユーザーので理解とご協力を前提として，適切なサービス体制をとっている．



5．販売代理店

■EDPシステム，医療システム関係

日本アイシーエス株式会社	〒543 大阪市天王寺区上本町5丁目3番21号 TEL06-761-5895
株式会社文祥堂 オフィス用品事業部	〒104 東京都中央区銀座３－４－12 TEL03-563-1511
株式会社岩崎	〒065 札幌市東区北六条東２丁目８番地 （札幌綜合卸センター内） TEL011-721-2531
武州商事株式会社	〒103 東京都中央区日本橋室町１－３－３ （大栄ビル） TEL03-279-3841
伊藤忠商事株式会社 産業電子機器部第３課	〒103 東京都中央区日本橋本町２－４ TEL03-662-5111
第一実業株式会社 産業機械第２部情報システムグループ	〒101 東京都千代田区神田錦町3丁目20番地 TEL03-296-6163，6167（ダイヤルイン）
光電メディカル株式会社	〒191 東京都日野市東平山２－25－４ TEL0425-84-1381
フクダエム・イー工業株式会社	〒112 東京都文京区本駒込２丁目９番９号 TEL03-946-0681

■商品（電源・筐体・計測器・計測システム関係）

株式会社データキキ	〒151 東京都渋谷区笹塚１－47－１ （京王重機ビル） TEL03-469-4961㈹
岩崎シンクロスコープ販売株式会社	〒105 東京都港区芝西久保巴町18 （第２松田ビル） TEL03-436-2461㈹
株式会社デンケイ	〒110 東京都千代田区神田須田町１－28 TEL03-256-2311㈹
日本電計株式会社	〒110 東京都台東区上野５－14－12 TEL03-834-6331
株式会社栃木屋	〒101 東京都千代田区内神田２－11－１ TEL03-254-2041

三益産商株式会社	〒370 群馬県高崎市上並榎町397 TEL0273-61-7511
大阪岩崎シンクロスコープ販売株式会社	〒530 大阪市北区南森町11 (八千代第一ビル) TEL06-358-0181
国華電機株式会社	〒530 大阪市北区空心町1－16 TEL06-353-5551
東海電子計測工業株式会社	〒465 名古屋市名東区猪高町猪子石亀鳥42-1 TEL052-772-0501
朝日エレクトロン株式会社	〒461 名古屋市東区布池町32 (カドノビル) TEL052-935-9281
株式会社金井計器製作所	〒940 長岡市柏町1－5－17 TEL0258-33-3525
轟産業株式会社	〒910 福井市毛天町3－2－4 TEL0776-36-5522
有限会社紫光通信	〒920-03 金沢市専光寺町2－13－1 TEL0762-68-4311
有限会社寿電商	〒802 北九州市小倉北区三郎丸1－8－26 (白石ビル) TEL093-931-1472
株式会社三木屋金物工具	〒892 鹿児島市千日町8の12 TEL0992-26-1091
大豊産業株式会社	〒760 高松市御坊町7の1 TEL0878-22-1313
新進電機株式会社	〒730 広島市千田町3－9－25 (広島工業会館) TEL0822-43-2977
尾脇電機株式会社	〒680 鳥取市片原町1の114 TEL0857-23-0841
東北無線電機株式会社	〒980 仙台市一番町1－14－23 TEL0222-21-7241
株式会社岩崎	〒065 札幌市東区北六条東2丁目8番地 (札幌綜合卸センター内) TEL011-721-2531

6. リースおよびローンについて

6－1 リースについて

6－1－1 概 要

経営にとって，設備，機械の投資は繁栄の大きな決め手である．かといって，現代は技術革新の時代．日に日に機械，設備も新しくなっているが，資金の問題や減価償却の問題があるので，そのつど設備投資をくりかえすわけにはいかない．設備や機械は，使うことによって始めて価値が生まれるものである．この考え方に基づいて，上記のような問題を解決する方法はないか？ そこで経営の知恵として登場したのがリースシステムである．

このシステムは，合理主義の徹底したアメリカで誕生したもので，第二次大戦後，急速に発展し，いまでは最も近代的な経営手段の一つとなっている．企業にとって，リースシステムをいかにうまく使いこなすか，そして，いつタイミングよく取り入れるかが，上手な経営施策なのではないだろうか．そのような時，リース経験と実績の生かされたノウハウが力を発揮する．

CECリースシステムは，購入資金なしで企業に必要なCEC555電子計算機システム，周辺端末機器，および付属設備を，自ら購入した場合とまったく同様に，自由に使用できるシステムである．委託リース会社がユーザーの要望により，中央電子から新たにCEC555電子計算機システム，周辺端末機器，および付属設備を買いとり，２～５年間の長期にわたり賃貸するもので，月々一定の賃貸料がリース料となる．さらに，電子計算機システムの所有に伴う減価償却，固定資産税，保険などの事務手続，および支払いなどはすべてリース会社でおこなうので，ユーザーはわずらわしい管理事務から解放されて，合理化を進めることになる．物件のアフターサービスについては，ユーザーと，中央電子，またはディーラーとの間で直接「CEC装置保守・サービス基本契約」に基づく契約を結んでいただき，ユーザーが購入した場合と同様に，中央電子，またはディーラーから充分な保守・サービスを直接受けることができる．

6－1－2 手続き

リース引き合い	ユーザー	リース会社	機種，金額等設備内容の提示 中央電子，ディーラーよりの見積書提出
↓			
リース説明	ユーザー	リース会社	リース料見積り，説明
↓			
リース申込	ユーザー		会社経歴書，過去２期分の財務諸表提出
↓			
審査	リース会社		リース会社の社内審査
↓			
リース契約	ユーザー	リース会社	契約締結，契約金の支払
↓			
発注	リース会社		中央電子，ディーラーへ発注
↓			
納入	中央電子		物件は直接，ユーザーの指定する場所へ期日迄納入
↓			
検収	ユーザー	リース会社	リース会社の立合い検査，確認
↓			
リース開始	ユーザー		借受証の発行，リース料手形の支払

6－1－3　リース料換算表（概算）

リース料率年11%（月利，3年3.69%，4年3.00%，5年2.58%）

リース期間 / 購入価格	3年		4年		5年	
	月額リース料	総額リース料	月額リース料	総額リース料	月額リース料	総額リース料
1,500,000	55,400	1,994,400	45,000	2,160,000	38,800	2,328,000
2,000,000	73,900	2,660,400	60,000	2,880,000	51,700	3,102,000
3,000,000	110,800	3,988,800	90,000	4,320,000	77,500	4,650,000
4,000,000	147,800	5,320,800	120,000	5,760,000	103,300	6,198,000
5,000,000	184,700	6,649,200	150,000	7,200,000	129,200	7,752,000
6,000,000	221,600	7,977,600	180,000	8,640,000	155,000	9,300,000
7,000,000	258,600	9,309,600	210,000	10,080,000	180,800	10,848,000
8,000,000	295,500	10,638,000	240,000	11,520,000	206,600	12,396,000
9,000,000	332,500	11,970,000	270,000	12,960,000	232,500	13,950,000
10,000,000	369,400	13,298,400	300,000	14,400,000	258,300	15,498,000
11,000,000	406,300	14,626,800	330,000	15,840,000	284,100	17,046,000
12,000,000	443,300	15,958,800	360,000	17,280,000	310,000	18,600,000
13,000,000	480,200	17,287,200	390,000	18,720,000	335,800	20,148,000
14,000,000	517,200	18,619,200	420,000	20,160,000	361,600	21,696,000
15,000,000	554,100	19,947,600	450,000	21,600,000	387,500	23,250,000
16,000,000	591,000	21,276,000	480,000	23,040,000	413,300	24,798,000
17,000,000	628,000	22,608,000	510,000	24,480,000	439,100	26,346,000
18,000,000	664,900	23,936,400	540,000	25,920,000	464,900	27,894,000
19,000,000	701,900	25,268,400	570,000	27,360,000	490,800	29,448,000
20,000,000	738,800	26,596,800	600,000	28,800,000	515,600	30,996,000

6－2　サイギン・中央電子コードキーヤローン

6－2－1　特長と手続き

1. 担保，保証人は不要．申し込み手続に繁雑な手間がかからず，気軽に利用できる．
2. 元利均等返済方式のため月々の負担が軽い（別表参照）．返済の期間は6カ月から3年まで，半年ごとの6種類が設定されている．したがって一番都合のよい期間を自由に選定できる．

ローンは，コードキーヤおよびそれに付随する機器の購入に利用するもので，10万円から200万円までローンの対象となる．

4. 手続きが非常に簡単である．ＣＥＣ営業部，または代理店に申し込み必要書類の所定の箇所に必要事項を記入するだけで手続きは完了する．
5. その他．

(1) 毎月の返済日は6日，16日，26日のうちのいずれかの日を指定できる．

(2) 指定日より振込みが遅れると延滞金が加算される.

(3) 購入台数が多い場合は別途協議とする.

6－2－2 元利均等償還金額一覧表（月額概算）

・利息年7.8%（月利0.65%）

●最終回の返済金額は若干変更がある.

返済回数 / 金額	6回(0.5年)	12回(1年)	18回(1.5年)	24回(2年)	30回(2.5年)	36回(3年)
10万	17,047	8,689	5,904	4,513	3,679	3,124
20万	34,095	17,379	11,809	9,027	7,359	6,248
30万	51,143	26,068	17,714	13,540	11,039	9,373
40万	68,191	34,758	23,619	18,054	14,718	12,497
50万	85,239	43,447	29,524	22,568	18,398	15,622
60万	102,287	52,137	35,429	27,081	22,078	18,746
70万	119,335	60,827	41,334	31,595	25,757	21,870
80万	136,383	69,516	47,239	36,108	29,437	24,995
90万	153,430	78,206	53,144	40,622	33,117	27,119
100万	170,478	86,895	59,049	45,136	36,796	30,244
110万	187,526	95,585	64,953	49,649	40,476	33,368
120万	204,574	104,275	60,858	54,163	44,156	36,492
130万	221,622	112,964	76,763	58,676	47,835	49,617
140万	238,670	121,654	72,668	63,190	51,515	43,741
150万	255,718	130,343	88,573	67,704	55,195	46,866
160万	272,766	139,033	84,478	72,217	58,874	49,990
170万	289,813	147,723	190,383	76,731	62,554	52,114
180万	306,861	156,412	106,288	81,244	66,234	56,239
190万	323,909	165,102	102,193	85,758	69,913	59,363
200万	340,957	173,791	118,098	90,272	73,593	62,488

（注） 現在CK－3，CK－6のほかにCK－10，12，15，20について交渉中であり，近く1件500万円までに拡張される予定である

7. CEC中央電子株式会社の概要

7－1 会社の概況

1.	会社の名称	中央電子株式会社
2.	会社の設立年月日	昭和35年4月5日
3.	会社の目的	工業用・医療用・民生用電子機器の設計，製造，販売ならびにこれに付帯する一切の業務.
4.	代表者名	代表取締役社長　石井　一恵
5.	所在地　《本社》	東京都八王子市元本郷町1丁目9番9号　〒192 電話（0426）23—1211（代）　テレックス2862—529
	《営業所》東京営業所	東京都新宿区市ヶ谷本村町28　ホワイトレジデンスビル　〒162 電話（03）269—4801（代）
	大阪営業所	大阪市淀川区西中島3丁目8番14号犬飼ビル　〒532 電話（06）303—0641（代）　テレックス523—4289
	九州営業所	北九州市戸畑区福柳木2丁目2番地　〒804 電話（093）881—9832
	名古屋営業所	名古屋市西区牛島町86番地カネヒサビル　〒451 電話（052）571—4501（代）
	《工場》本社第一工場	東京都八王子市元本郷町2丁目6番24号　〒192 電話（0426）23—1211（代）
	本社第二工場	東京都八王子市元本郷町2丁目5番4号
	本社第三工場	東京都八王子市元本郷町1丁目14番地
	システムセンター	東京都八王子市明神町4丁目471番地　〒192 電話（0426）44—3011（代）
	商品工場	東京都八王子市千人町3丁目5番13号　〒193 電話（0426）61—0611（代）
	機械工場	東京都八王子市平岡町18－14　〒192 電話（0426）23—1624
	千人工場	東京都八王子市千人町1丁目4番8号　〒193 電話（0426）64—0374（代）
	本郷工場	東京都八王子市元本郷町3丁目34番11号
	藤野製作所	神奈川県津久井郡藤野町沢井字日野176番地　〒199—02 電話（04268）7—2341（代）
	《施設》	
	ＣＥＣ健康管理センター	東京都八王子市元本郷町1丁目8番6号　〒192 電話（0426）23—1211（代）
	ユーザ研修クラブ	東京都八王子市中野町3468番地　〒192 電話（0426）22—0212
6.	資本金	払込済資本金　320,000千円
7.	主な株主	日産自動車（株），伊藤忠商事（株），住友電気工業（株），日本電気（株），（株）埼玉銀行，岩崎通信機（株），武州商事（株），日立電子（株），石井一恵.
8.	主要取引金融機関	商工組合中央金庫八王子支店，埼玉銀行八王子支店，富士銀行八王子

支店，太陽神戸銀行八王子支店，三井銀行八王子支店，三菱銀行八王子支店，東海銀行八王子支店，第一勧業銀行八王子支店，協和銀行八王子支店，八王子信用金庫，日本開発銀行，日本興業銀行．

9．役員名および略歴

代表取締役社長	石井　一恵	昭和28年３月	一橋大学法学部卒
常務取締役	山口　幸也	昭和27年３月	東京大学工学部卒
常務取締役	高橋　啓	昭和34年３月	東北大学工学部卒
取締役	小林　宏	昭和31年３月	電気通信大学卒
取締役	吉原　亘	昭和34年３月	早稲田大学政経学部卒
取締役	前川　豊	昭和29年３月	神戸大学法学部卒
取締役（非常勤）	若林　玄修		
取締役（非常勤）	田中健次郎		
取締役（非常勤）	尾崎　知三		
監査役（非常勤）	野田　賢一		

10．従業員の状況

性別	事務・管理・営業系			技術系			作業職系			計		合計
	男	女	計	男	女	計	男	女	計	男	女	
社員	60	21	81	129	11	138	61	11	72	250	43	293
社友・嘱託	1	4	5	1	3	4	4	44	48	6	51	57
計	61	25	86	130	14	142	65	55	120	256	94	350

（昭和50年４月現在）

7　2　会社の沿革

昭和35年４月　東京都八王子市元本郷町に中央電子株式会社を設立，本社および工場を新設して産業用電子機器の設計製作，販売，ならびにこれに付帯する業務を開始，資本金4,000千円，従業員13名（役員を除く），石井一恵代表取締役社長となる．

昭和35年７月　計算機用安定化直流電源を開発した．

35年12月　八王子平岡町に機械工作を専門とする機械工場を新設．

36年４月　パルス技術，高周波技術，半導体技術が高く評価され始め，諸官庁研究所の指名を受けた．

36年11月　2,000千円増資し，資本金6,000千円となる．

37年６月　東京都千代田区神田に東京営業所を開設．

37年12月　八王子市元本郷町に社屋を新設し本社を移転する．

38年２月　海外進出の第一歩としてパルス機器を初輸出する．

38年９月　10,000千円増資し，資本金16,000千円となる．

39年８月　8,000千円増資し，資本金24,000千円となる．

40年11月　営業面の拡充を計るため大阪市北区に大阪営業所を開設．

41年４月　技術部が第１，第２，第３，第４，商品技術および製造技術課の６課となる．

41年６月　構造部品の標準化の一環として，アルミサッシを材料とした筐体を採用した．実願３件．

41年11月　ディジタル応用装置をはじめ各種電子応用機器の設計に全面的に集積回路を採用．

42年４月　21,000千円増資し，資本金45,000千円となる．

42年６月　八王子市元本郷町に敷地を買収，本社工場を新設し，工場の分散を集中した．

43年９月　神奈川県津久井郡藤野町にアルミサッシ材加工専門の工場を設ける．

44年3月　事業の拡大に伴い，八王子市元本郷町に伸和ビルを設置して技術部および営業部のフロアーを増設.

44年4月　35,000千円増資し，資本金80,000千円，従業員159名となる.

44年5月　神奈川県津久井郡藤野町に新工場建設のため土地1884平方米を購入し建設に着手する.

45年1月　神奈川県津久井郡藤野町に藤野製作所を増設し，アルミサッシ標準筐体の生産体制を拡充した.

45年2月　八王子市明神町にシステムセンターを建設し，CEC555形電子計算機をはじめ情報システムの生産体制を拡充した.

45年2月　日本電信電話公社データ通信本部にCEC556B教育用電子計算機システムを納入した.

45年2月　情報システムの需要が増大し従業員180名となる.

45年3月　日本国有鉄道にCEC555電子計算機を納入した.

45年6月　CEC555電子計算機を大阪大学，信州大学，明治大学等大学，官公庁，民間各社に納入し，本格的な情報システム産業としての活動をはじめた.

46年2月　CEC555電子計算機を4K語単位で年間100セット販売した.

46年2月　日本電信電話公社近畿データ通信本部にCEC556B教育用電子計算機システムを納入した.

46年3月　日本開発銀行から電振法によるCEC555電子計算機の生産設備資金の借入れを受けた.

46年4月　従業員230名となり，東京営業所を新宿区市ヶ谷に，大阪営業所を淀川区西中島にそれぞれ移転拡充し，販売・サービス面を強化した.

46年4月　CEC552心電図自動診断システムを群馬県立前橋病院に納入し，脳卒中と心臓病の予防のための医用電子システムの販売を開始した.

46年6月　CEC555電子計算機による脳波検診システムを開発した.

46年11月　資本金80,000千円を増資し，新資本金160,000千円となる.

46年11月　藤野製作所第2期工事が完成し，CEC555電子計算機の専門工場として操業を開始した.

47年4月　従来の500シリーズ，400シリーズの直流安定化電源に，装置用多出力電源を加え，電源機器の商品充実をはかった.

47年4月　従業員251名となる.

47年5月　電子計算機用の万能入力端末装置，CKシリーズコードキーヤを開発し，周辺機器の販売拡充をはかった.

47年6月　本社工場隣接地に新工場を増設し，情報処理システムの専用工場として生産体制を拡充した.

47年11月　電子式自動紋紙作成システムを商品化し，萱場工業（株）に納入（日刊工業新聞社1972年度10大製品選定）.

48年3月　普及健康管理システムが，通産省新規機械の商品化試作の指定を受け，日本開発銀行より国産技術振興資金の融資を受けた.

48年4月　従業員267名となる.

48年4月　九州営業所を北九州市戸畑区に新設した.

48年4月　東京霞が関ビルにおいて医用情報システムの個展を開き多大の反響を呼んだ.

48年6月　北京市で開催の日本自動化機器，医療機器展に参加し中国親善の一役を担った.

48年８月　文部省後援の教職員夏季実技講習会に「経営数学」を担当して初参加し，好評を得た.

48年９月　名古屋営業所を名古屋市西区に新設した.

48年10月　電子技術総合研究所，東京医科歯科大学，東邦大学，新潟大学の協力と指導のもとに「脳波自動診断装置」を開発，日本脳波筋電図学会に発表した.

48年12月　資本金160，000千円を増資し，新資本金230,000千円となる.

49年１月　ＣＫシリーズコードキーヤ利用の病院事務処理システムを開発し，第１号システムを日野市石塚病院に納入，財務会計処理システムとあわせて事務処理システム部門の商品化をはかった.

49年３月　ＣＫシリーズコードキーヤにオンライン用ＣＫ―５型，ページ式を加え強化をはかった.

49年４月　CEC555SS応用システムであるCEC512血液生化学自動分析装置を開発，商品化し，心電図，脳波，医用事務処理を含む総合医用情報処理システムメーカーとして新たにＭＥ部門を新設し，拡販にのりだした.

49年４月　従業員数291名となる.

49年５月　CEC健康管理センターが完成し，地域健康福祉，成人病予防のモデルシステムとしてただちに運営に入った.

49年６月　コードキーヤの新シリーズＣＫ―６，ＣＫ―８が49年度通産省「電子計算機等開発促進補助金」の適用を受けた.

49年12月　機電法によるCEC555電子計算機の生産設備資金の第２次開銀融資を受けた.

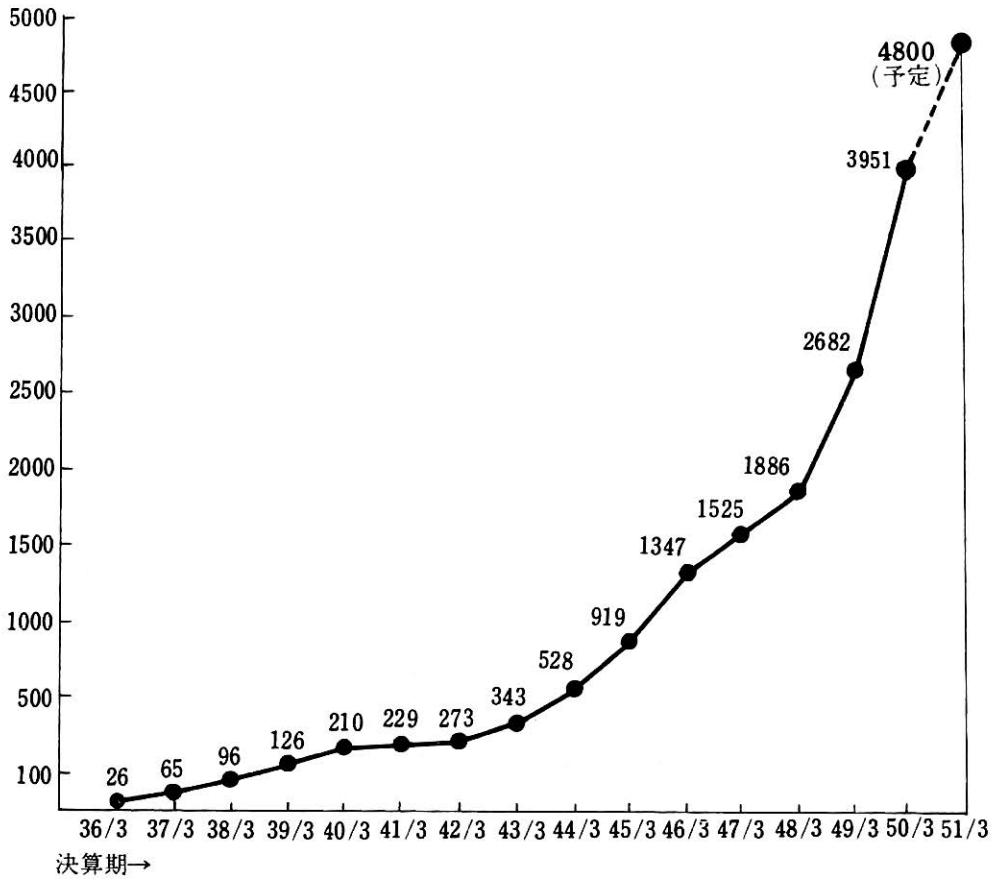
49年12月　コードキーヤ利用のCEC　EDP　SYSTEM10～60シリーズを商品化し，事務計算部門への販売拡大にのり出す.

50年５月　健康管理システムの第２号システムを厚生連鶴見病院に納入する.

50年５月　「電子計算機等開発促進補助金」適用のＣＫ―６，ＣＫ―８ブック式コードキーヤが完成，発表と同時に業界より注視をあびた（実用新案９件）.

50年６月　「異種細胞自動検出診断装置に関する研究試作」が通産省の昭和50年度重要技術研究開発費補助金の適用を受けた.

7－3　生産高推移（単位：100万円）



7－4　主な事業分野

事　　業	指　　定　　商　　品
1．工場制御システム	1．工場群制御システム 2．プラント計算制御システム 3．工場，倉庫集中管理システム
2．トータシステム	1．競技場総合管理システム 2．競技場前売システム 3．競技場加算機システム 4．競技場テレビモニタシステム 5．競技場人員管理システム 6．競技場窓口集計システム
3．EDPシステム	1．一般事務管理処理システム 2．CEC　EDPシステム （10，20，30，50，60シリーズ） (1)　財務会計処理 (2)　販売管理 (3)　在庫管理 (4)　輸送業務管理 (5)　給与計算 (6)　仕入管理 (7)　人員管理 (8)　工程管理 (9)　工業簿記 (10)　商業簿記 (11)　特殊法人会計 (12)　経営数学 (13)　土木測量計算 (14)　その他各種
4．医療情報システム	1．CEC 552心電図自動診断装置 2．CEC 552D心電心音自動診断システム 3．CEC 540脳波自動診断システム 4．CEC健康管理センター（成人病予防センタ） 5．病院事務処理自動化システム 6．レセプト作成システム 7．CEC 512血液生化学分析装置 8．その他医療用自動診断，解析システム各種

5．繊維産業システム	1．電子式自動柄出し丸編機システム 2．自動図柄分析システム 3．電子紋紙自動穿孔システム 4．ラッセル自動編システム 5．捺染自動システム 6．柄準編機システム 7．その他繊維産業システム
6．計測制御システム	1．データ集録自動処理システム 2．ＮＣ制御システム 3．PKG自動検査システム 4．計測制御システム 5．通信，制御システム 6．データ通信システム
7．自動車エレクトロニクス	1．電子，電装品，自動試験システム 2．電子，電装品，研究開発システム 3．フイールドサービスシステム 4．自動車，部品販売管理システム
8．文教システム	1．教育用電子計算機システム 2．データ統計解析システム 3．計測制御システム
9．555シリーズ電子計算機	1．CEC555シリーズ電子計算機 2．各種周辺機器
10．装置用電源，筐体基礎技術機器	1．各種標準直流電源 2．各種装置用電源装置 3．各種標準筐体 4．各種特殊筐体 5．パルス応用機器 6．基礎技術装置

7－5　主な納入実績（主要なもののみ納入順に記載，敬称略）

① 工場制御システム

システム名	数量	納入先	主な構成内容
監視用システム	1 式	日本工営㈱	555(8D)＋CS100＋CCD5S
自動倉庫システム	3 式	住友重機械工業㈱	555T＋CS100＋CS408 他
〃	1 式	〃	555T＋CS100＋CS408 他
監視用システム	1 式	日本工営㈱	555T＋CS100＋CCD5S 他
ロボット制御装置	1 式	住友重機械工業㈱	555T＋CS100＋CS408＋CS420
自動倉庫システム	1 式	〃	555T＋CS100＋CS408 他
形状検出システム	1 式	新日本製鉄㈱	555S＋CS100
分塊ビレット探傷信号システム	1 式	住金物産㈱	555SS-I 他
データ通信・制御システム	1 式	新日本製鉄㈱	555SS-II，CS-80 キーボード　その他

② トータライザシステム

システム名	数量	納入先	主な構成内容
桐生競艇場トータライザシステム	1 式	日本トータ株式会社	555 8D(2)＋555S(6)＋CS-1000(8) 他
三国競艇場トータライザシステム	1 式	〃	555 8D(2)＋555S(6)＋CS-1000(6)
津競艇場トータライザシステム	1 式	〃	555 8D(2)＋555S(6)＋CS-1000(6)＋CCD-5S 他
蒲郡競艇場トータライザシステム	1 式	〃	555 8D(2)＋555S(8)＋CS-1000(6)＋CCD-5S 他
足利競馬場トータライザシステム	1 式	〃	555 12D(2)＋555S(2)＋CLP-4100(2) 他
金沢競馬場トータライザシステム	1 式	〃	555H-16(2)＋555S(10)＋CLP-4100(2)＋CMD-8 他
浜名湖競艇場トータライザシステム	1 式	〃	555H-16(2)＋555S(10)＋CLP-4100(2) 他

③ EDPシステム

その1　事務処理システム関係

システム名	数量	納入先	主な構成内容
ソフトウェア作成用システム	1 式	㈱アンドールシステムサポート	555(8)＋CS100
CEC555(8)システム	1 式	丸善㈱	555(8)＋CS1000
販売管理システム	1 式	東京シャツ	555S＋CS1000
在庫管理システム	1 式	住友重機械工業㈱	555S＋CS100
事務処理システム	1 式	シャープ㈱	555S＋CS100
在庫管理システム	1 式	住友電気工業㈱	555(4)＋CS100
事務処理システム	1 式	シャープ㈱	555S＋CS400＋CS408
在庫管理システム	2 式	住友電気工業㈱	555(4)＋CS100＋CS408＋CS420
事務処理システム	1 式	シャープ㈱	555S＋CS1000＋CS408
〃	3 式	〃	555S＋CS1000
〃	4 式	〃	555S＋CS100
在庫管理システム	1 式	住友電気工業㈱	555M＋CS100＋CLP4100 CS408＋CS420
〃	1 式	〃	555(4)＋CS1000＋CS408＋CS420 CFD130
財務会計処理システム	1 式	日本アイシーエス㈱	555(8D)＋CS1000＋CFD300 CS408＋CS420＋CK3＋CLP4100
事務処理用システム	5 式	伊藤忠商事㈱	555S＋CS1000＋CS408
〃	1 式	〃	555T＋CS1000＋CS408
〃	1 式	シャープ㈱	555S
〃	1 式	〃	555S＋CS1000＋CS408＋CS420
在庫管理システム	1 式	住友電気工業㈱	555(4D)＋CFD130＋CS1000
〃	1 式	〃	555M＋CFD130＋CS100＋CS408＋CS420
財務会計処理システム	2 式	日本アイシーエス㈱	555(8D)＋CFD130＋CS1000 CS408＋CS420＋CK3＋CLP4100
事務処理用システム	3 式	伊藤忠商事経由各社	555(8D)＋CFD130＋CS1000 CLP4100＋CS408＋CS420
在庫管理システム	1 式	住友電気工業㈱	555(4D)＋CS1000＋CFD130
〃	1 式	日本特殊鋼㈱	555M＋CS1000＋CS408 他

財務会計処理システム	1 式	日本アイシーエス㈱	555(8D)＋CFD130＋CS1000＋CLP4100＋CS408＋CS420＋CK3
〃	2 式	〃	555(8D)＋CFD130＋CS1000 CLP4100＋CS408＋CS420＋CK3
〃	1 式	〃	555H(8)＋DRISK2002＋CS1000＋CS408＋CS420＋CK3
在庫管理システム	1 式	住友電気工業㈱	555H(8)＋CS100 他
財務会計処理システム	1 式	日本アイシーエス㈱	555H(8)＋DRISK2002 CLP4100＋CS1000＋CS408　CS420＋CK3
ソフトウェア開発用システム	1 式	国際経研㈱	555(4D)＋CS100
プログラム開発システム	1 式	事務計算センター	555(8D)＋CFD130＋CLP4100＋CS1000＋CS408＋CS420＋CK3
販売在庫管理システム	1 式	㈱さくらカラー大阪	555H＋DRISK2002＋CS100 CS408＋CS420＋CK3
在庫管理システム	1 式	住友電気工業㈱	555M＋CS1000 他
財務会計処理システム	1 式	伊藤忠商事㈱	555(8D)＋CFD130 他
料金計算システム	1 式	立石電機㈱	555S＋CS100＋CS408
事務処理システム	1 式	住友電気工業㈱	555H(8)＋DRISK2002＋CS100＋CLP4100 他
財務会計処理システム	1 式	日本アイシーエス㈱	555H＋DRISK2002＋CLP4100＋CK3 他
〃	1 式	〃	555H(8)＋DRISK2002＋CLP4100 他
輸送業務管理システム	1 式	産業輸送開発㈱	555H＋CS1000＋DRISK2002＋CLP4100＋CS408＋CK3
財務会計処理システム	2 式	日本アイシーエス㈱	555H(8)＋DRISK2002＋CLP4100 CS1000＋CS408＋CK3
事務処理システム	1 式	ユニ電子計測産業㈱	555H(8)＋DRISK2002＋CLP4100 CS408＋CK3
在庫管理システム	1 式	住友電気工業㈱	555H(8)＋DRISK2002＋CLP4100 他
財務会計処理システム	5 式	伊藤忠商事㈱	555H＋DRISK2002＋CLP4100＋CS1000＋CS408　CK3 他
事務システムプログラム作成システム	1 式	㈱事務計算センター	555H＋DRISK2002＋CLP4100 CS1000＋CS408＋CS420＋CK3 他
事務処理用プログラム作成システム	1 式	伊藤忠商事㈱	555H(8)＋DRISK2002＋CLP4100 他
各種計算業務委託	1 式	斉藤電算センター	
〃	1 式	中央電算センター	555H(24)＋DRISK2002(2)＋CLP4100＋CMT8＋CK10 他
財務会計処理システム	1 式	日本アイシーエス㈱	555A＋CFD130＋CK5 他
〃	1 式	〃	555H＋CLP4100 他
〃	1 式	〃	555H(8)＋CFD300B 他
〃	1 式	〃	

訓練用システム	1 式	中央工営㈱	555(8)＋CS408＋CS420＋CS1000
財務会計処理システム	1 式	日本アイシーエス㈱	555H(8)＋CFD130＋CLP4100 他

その2　建築土木測量計算システム関係

システム名	数量	納入先	主な構成内容
土木測量計算システム	1 式	㈲山下測量	555S＋CS1000
〃	1 式	東邦測量㈱	555S＋CS1000
〃	1 式	北欧コンサルタント	555S＋CS1000
〃	1 式	岩崎㈱経由各社	555S＋CS1000＋CS408
〃	3 式	〃	555(4)＋CS1000
〃	1 式	〃	555S＋CS1000
〃	5 式	〃	555S＋CS1000
〃	6 式	〃	555(4)＋CS1000
〃	1 式	〃	555S＋CS1000
〃	1 式	〃	555(4)＋CS1000
〃	3 式	〃	555(4)＋CS1000＋CS408
〃	1 式	〃	555T＋CS1000
〃	2 式	〃	555(8)＋CS1000＋CS408
〃	1 式	〃	555S＋CS1000＋CS408
〃	1 式	〃	555T＋CS1000＋CS408
〃	1 式	〃	555(4)＋CS1000＋CS408
〃	5 式	〃	555(4)＋CS1000＋CS408
〃	4 式	〃	555S＋CS1000＋CS408
〃	3 式	〃	555T＋CS1000＋CS408
〃	1 式	〃	555S＋CS1000＋CS408
〃	2 式	〃	555(8)＋CS1000
〃	1 式	〃	555S＋CS1000＋CS408
自動作図システム	1 式	〃	555(8D)＋C1525XYプロッタ
土木測量計算システム	1 式	国際商事経由中国農水産部	555(8D)＋CFD130＋CS1000＋CS408＋CS420
〃	2 式	㈱岩崎経由各社	555(8)＋CS1000

土木測量計算システム	3 式	伊藤忠商事㈱経由各社	555(8)＋CS1000
〃	6 式	㈱ 岩 崎 経 由 各 社	555(8)＋CS1000
自動作図システム	3 式	〃	555(8)＋CS100＋XYプロッタ
土木測量計算システム	3 式	伊藤忠商事㈱経由各社	555(8)＋CS1000＋CS408
〃	3 式	㈱ 岩 崎 経 由 各 社	555(8)＋CS1000＋CS408
〃	5 式	〃	555S＋CS1000 他
自動作図システム	1 式	〃	555H＋CS100＋C1525XYプロッタ
〃	1 式	㈱ 岩 崎	555H＋CS100＋XYプロッタ
〃	1 式	国際商事経由中国農水産部	555H(8)＋CS1000＋C-1525 他
土木測量計算システム	1 式	㈱ 岩 崎 経 由 各 社	555H(8)＋CS1000＋CS408 他

④ 医療情報システム

その1　検診センター中央処理システム関係

システム名	数量	納入先	主な構成内容
成人病健康管理システム	1 式	みどり会健康管理センター	CEC552　CEC555-8D(2台)
〃	1 式	ヘルシー健康管理センター	552, 552S　CEC555H　DRISK-2002　カートリッジ磁気 ディスク他
検診データ自動処理システム	1 式	群馬県成人病研究所	552, 552S　CEC555H　CK-3コードキーヤ　CFD-130磁気 その他
成人病健康管理システム	1 式	大分県農業厚生連　鶴見病院併設	552,CEC555H　DRISK-2002 他
CEC健康管理センター	1 式	若 林 医 院	552, 512　CEC555H　DRISK-2002 他
検診データ自動処理システム	1 式	東京医事管理センター	555H(24)＋DRISK-2002(2)　CMT-8＋CK-20

その2　心電図自動検診装置関係

システム名	数量	納入先	主な構成内容
CEC552心電図自動検診装置	1 式	群馬県立前橋病院成人病研究所	552＋552S
〃	1 式	協栄生命保険㈱健康管理センター	552

ＣＥＣ552心電図自動検診装置	1 式	ＰＬ健康管理センター	552
〃	1 式	大和郡山市	552＋552S
〃	1 式	みどり会健康管理センター	552
〃	1 式	東洋工業中央病院	552
〃	1 式	渋川病院	552＋552S
〃	1 式	大日本塗料㈱	552
〃	1 式	岡山県新見市阿新健康管理センター	552＋552S
〃	1 式	四国電力	552＋552S
〃	1 式	日興証券㈱	552
〃	1 式	長野県厚生農業協同組合連合会 佐久総合病院	552＋552S（2台）
〃	1 式	久保田鉄工㈱	552＋552S
CEC552D心電・心音自動診断システム	1 式	宮崎県健康増進センター	552D＋552S
CEC552心電図自動診断装置	1 式	ヘルシー健康管理センター	552
〃	1 式	自治医科大学	552
〃	1 式	中国衛生院	552
CEC552D心電・心音自動診断システム	1 式	鹿児島県成人病予防協会	552D＋552S

〔注〕552 （心電図自動検診装置） 552S （カセット式心電図記録装置）

その3 脳波自動診断システム関係

システム名	数量	納入先	主な構成内容
CEC540脳波自動診断装置	1 式	東京医科歯科大学	厚生省，電子技術総合研究所，東京医科歯科大学，東邦大学，新潟大学の協力 CEC555
〃	1 式	東邦医大学	CEC555利用
脳波データ処理システム	1 式	東京都精神医学研究所	555H＋CS100＋CFD130＋CS408＋CS420＋CLP4100
CEC540脳波自動診断装置	1 式	新潟医大	CEC555利用
〃	1 式	独協大学医学部	CEC555H利用
〃	1 式	自治医科大学	CEC555H利用

その4　血液生化学自動分析装置関係

システム名	数量	納入先	主な構成内容
CEC512血液生化学 自動分析装置	1 式	CEC健康管理センター	CEC555SS利用　14項目

その5　医療事務自動処理システム

システム名	数量	納入先	主な構成内容
病院事務自動化システム	1 式	日野市石塚病院	CEC555H, DRISK2002, CLP4100, CK-5, CS1000
〃	1 式	前橋市群馬県立前橋病院	CEC555H, DRISK2002, CMD-8, CLP4100, CK-5
〃	1 式	会津若松市　総合会津中央病院	CEC555H, DRISK2002, CMT-8, CLP4100, CK-5
〃	1 式	若林医院	健康管理センタシステムと併用
〃	1 式	自治医科大学付属病院	555H(24)＋CK-10＋DRISK2002(2) 他
病院事務自動化装置	多数	各病院　医院	CK-3コードキーヤ　CK-6コードキーヤ

⑤ 繊維産業システム

システム名	数量	納入先	主な構成内容
電子式自動柄出丸編機システム	1 式	機械振興協会	555(4)＋CS100
〃	1 式	〃	555(8D)＋CS100
〃	1 式	大東製機㈱	555(4)＋CS100
自動図柄分析システム	1 式	萱場工業㈱	555T2台＋CS100＋CS408 555S＋CS100＋CCD5G
〃	3 式	大東製機㈱	555S＋CS100＋CS408　CCD5G
自動柄出丸編機システム	4 式	〃	555T＋CS100＋CS408＋CS420
自動図柄分析システム	1 式	萱場工業㈱	555T＋CS100＋CS408　CCD5G
〃	1 式	金井計器製作所	555(8)＋CS100＋CS408　CCD5G
〃	1 式	大東製機㈱	555T＋CS100＋CS408　CCD5G

自動図柄分析システム	1 式	大東製機㈱	555S＋CS100＋CS408　CCD5G
自動紋紙せん孔システム	1 式	萱場工業㈱	555H＋CS100＋CS408
自動柄出丸編機システム	3 式	大東製機㈱	555T＋CS100＋CS408＋その他
〃	8 式	〃	555S＋CS100
自動図柄分析システム	4 式	〃	555T＋CS100＋CS408　CCD5G
自動紋紙せん孔システム	1 式	萱場工業㈱	555M＋CS100＋CS408＋その他
〃	1 式	〃	555H(8)＋CS100＋CS408＋その他
自動柄出丸編機システム	5 式	大東製機㈱	555T＋CS100＋CS408　CCD5G
自動図柄分析システム	2 式	〃	555S＋CS100＋CS408＋CCD5G
レース編自動化装置	1 式	松浦機械製作所	555T＋CS100＋CS408 その他
自動柄出丸編機システム	1 式	大東製機㈱	555H(8)＋CS100＋CS408 その他
自動図柄分析システム	2 式	〃	555S＋CS100＋CCD5G
〃	1 式	萱場工業㈱	555H(8)＋CS100＋CCD5G
〃	3 式	大東製機㈱	555S＋CS100＋CS408 他　CCD5G
〃	1 式	〃	555S＋CS100＋CS408　CCD5G
〃	1 式	〃	555S＋CS100＋CS408 他　CCD5G
自動柄出丸編機システム	2 式	〃	555S＋CS100＋CS408
			555T＋CS100＋CS408
〃	1 式	〃	555S 他
自動図柄分析システム	2 式	〃	555S＋CS100＋CS408 他　CCD5G
自動柄出丸編機システム	2 式	〃	555T＋CS100＋CS408 他
高速図柄分析システム	1 式	大同マルタ染工㈱	555H＋CS100＋CS408＋CCD5GH 他
もん紙システムプログラム作成システム	1 式	萱場工業㈱	555H＋CS100＋CS408＋CS420 他
自動図柄分析システム	1 式	〃	555H＋CS100＋CS408＋CCD5G 他
〃	1 式	大東製機㈱	555H(16)＋DRISK-2002＋CCD-5GH＋ディジタイザ 他

⑥ 計測制御システム

システム名	数量	納入先	主な構成内容
データ処理装置	1 式	ソニーテクトロニクス㈱	555(8)＋CS-100
電卓ＬＳＩチェッカ	1 式	三菱電機㈱	555S＋CS-100

データ処理システム	1 式	住友商事㈱	555(8)＋CS-100
電卓チェッカ	1 式	三菱電機㈱	555(8)＋CS-100＋CS-408
研究用システム	1 式	松下電器産業㈱	555(4)＋CS-100
実験用システム	1 式	新電子㈱	555S＋CS-100
路面すべり測定システム	2 式	シャープ㈱	555S＋CS-100
熱疲労試験システム	1 式	イスズ電機㈱	555S＋CS-1000　CS408＋CS-420
実験用システム	1 式	朝日エレクトロン㈱	555(8)＋CS-1000＋CS-408＋CS-420
試験用システム	1 式	㈱アサカ	555S＋CS-100
パターン認識装置	1 式	シャープ㈱	555(4)＋CS-100
ＬＳＩ単体チェッカ	7 式	〃	555S＋CS-100
データ処理システム	2 式	住友電気工業㈱	555(4)＋CS-100＋CS-408＋CS-420　CS-408
試験用システム	1 式	テルエンジニアリング㈱	555T＋CS-100
ラボ自動処理システム	1 式	小西六写真工業㈱	555S＋CS-100
実験用システム	1 式	シャープ㈱	555S＋CS-100
試験用システム	1 式	ファンテックデータサイエンス	555S＋CS-100
実験用システム	1 式	住友重機㈱	555T＋CS-100
試験用システム	1 式	テルエンジニアリング㈱	555T＋CS-100
〃	1 式	〃	555(8)＋CS-100＋CS-408＋CS-420
実験用システム	1 式	大和通信㈱	555(4)
試験用システム	1 式	テルエンジニアリング㈱	555(8D)＋CS-100＋CS-408＋CS-420　CFD-130
実験用システム	1 式	中央システムエンジニアリング	555(8D)＋CFD-130＋CS-1000＋CS-408＋CS-420
試験用システム	1 式	テルエンジニアリング㈱	555T＋CS-100
〃	1 式	三和電子製作所	555S＋CS-100
〃	1 式	〃	555(8)＋CS-100＋CS-408
公害監視用データ処理装置	4 式	柳本製作所	555S＋CS-100＋その他
実験用システム	1 式	シャープ㈱	555S＋CS-100
〃	1 式	〃	555S
データ処理システム	1 式	立石電機㈱	555S＋CS-100＋CS-408
実験用システム	1 式	シャープ㈱	555T＋CS-100
試験用システム	1 式	パネトロン㈱	555T＋CS-100
〃	1 式	三洋電機㈱	555S＋CS-100
公害監視データ処理システム	3 式	柳本製作所	555S＋CS-100＋その他
プリント基板自動試験システム	1 式	国際商事経由中国北京電子管工場	555(8)＋CS-408＋CS-100

試験用システム	1 式	大阪工業試験所	555(12D)+CS-1000+CFD-130
〃	1 式	メックエンジニアリング	555S+CS-100+CS-408
実験用システム	1 式	㈱岩崎	555S 他
〃	1 式	朝日エレクトロン㈱	555(4D) 他
〃	1 式	マークランド工業㈱	555S 他
試験用システム	1 式	シャープ㈱	555S
NC制御多軸ボール盤	1 式	国際商事経由北京自動車工場	555(4)+CS-100+CS-408+その他
調整用システム	1 式	日本電子機器㈱	555T 他
調整機システム	1 式	〃	555T 他
ビル監視盤装置	1 式	西日本コンピュータ㈱	555S+CS-100 他一式
振動分析システム	1 式	日本国有鉄道	555+CS-100
データ処理装置	1 式	電子技術総合研究所	555T+CS-100
SSRR距測実驗装置	1 式	郵政省電波研究所	555S+CS-100
スタジオ照明制御装置	1 式	NHK	555T+CS-100
選挙速報システム	1 式	NETテレビ	555H(8)+CS-100+CS-408+CCD5GH
その他多種多数			
リーダパンチャ試験システム	1 式	リコー電子工業㈱	555T 他
自動作図システム	1 式	㈱岩崎	555H+DRISK-2002+C-1525D 他
コークス炉データ処理装置	1 式	新日本製鉄㈱	555S+CS-100 他

⑦ 自動車エレクトロニクス

システム名	数量	納入先	主な構成内容
自動計測記録装置	1 式	日産自動車㈱	555(4)+CS-100
データ集録自動処理システム	1 式	〃	555M+CS-100+CS-408 CS-420+CMT-8+IBM-735
部品耐久試験システム	1 式	〃	555(8)+CS-100+CS-408 CS-420
自動検査システム	1 式	日本エアブレーキ㈱	555(4)+CS-100+CS-408 他
データ処理システム	1 式	日産自動車㈱	555T+CS-100+CS-408+CS-420
ロボット制御用システム	1 式	〃	555(8)+CS-100+CS-408+CS-420
検査データ処理システム	1 式	〃	555T+CS-100+CS-408+CS-420
エンジン燃焼解析装置	1 式	〃	555H(12)+CS-100 他

⑧ 文教システム

その 1

システム名	数量	納入先	主な構成内容
データ通信教育・展示	1 式	日本電々公社 データ通信本部	555-8D, DISK, 展示ボード, TTY, PTR
教育・実験用	1 式	明治大学工学部	555-8D, DRUM, TTY, カセットMT, PTR, LP
〃	1 式	信州大学繊維工学部	555-8, TTY, X-Yプロッタ 教育用ボード, PTR
高校電算機教育	1 式	都城工業高校	555-8, TTY, COLORDISPLAY, PTR, PTP
教育・実験用	1 式	大阪大学工学部	555-8D, TTY, PTR, PTP各2システム, DRUM, DISK
専門学校教育用	1 式	千代田学園	556-4, TTY, 教育用ボード
データ通信教育・展示	1 式	日本電々公社近畿データ通信本部	556-8D, TTY, DISK, PTR, PTP, 教育用ボード
工専における電算機教育	1 式	東京高等工業専門学校	555-8, DRUM, PTR, PTP, TTY2台, 教育用ボード
教育・実験用	1 式	北海道大学工学部	555S, TTY, 2システム
〃	1 式	青山学院大学理工学部	555-8, TTY, PTR, PTP, 教育用ボード
工業高校電算機教育	1 式	戸畑工業高校	556-4, TTY, 教育用ボード PTR
専門教育・実習	1 式	白石職業訓練校	556-4, TTY, 教育用ボード, PTR, PTP, マークカードリーダ
教育・実験用	1 式	韓国麗南大学	556-4, TTY, 教育用ボード, PTR
工業高校電算機教育実習	1 式	前橋工業高校	555-8, TTY, 教育用ボード, PTR, DISK
専門教育・実習	1 式	宮古専修職業訓練校	555-4, TTY, 教育用ボード
教育・研究用	1 式	立命館大学	555-8, TTY
教育・研究・実習	1 式	大阪大学工学部	555H-8, DISK, PTR, PTP, LP
教育・実習	1 式	茨城工専	555-8, TTY, PTR, X-Yプロッタ
脳波研究	1 式	新潟大学	555-8, PTR, PTP, A／D, D／Aユニット, TTY
〃	1 式	東邦大学	555-T, TTY, A／D, D／Aユニット, PTR
〃	1 式	東京医科歯科大学	555H-16, TTY, DISK, PTR, PTP, LP
研究用	1 式	東京大学	555T, TTY
脳波研究	1 式	独協大学	555H-16, TTY, PTR, PTP, DISK, LP, DISPLAY
〃	1 式	自治医大	555H-16, TTY, PTR, PTP, DISK, LP, DISPLAY
教育実習用	1 式	航空保安大学	555-8D, DRUM, PTR, PTP, MT, LP, DISPLAY, 教育用ボード
〃	1 式	神戸大学	555-8D, TTY, PTR, PTP, DRUM
生物実験データ処理装置	1 式	東京大学	
教育実習用	1 式	九州産業大学	555H(24)＋CMT-8＋CMD-8＋C-1525＋CK-8 他
ガスクロマトグラフ制御装置	1 式	東京大学	555(4)＋CS-100 他

その 2

システム名	納入先名	システム名	納入先名
1051教育用電子計算機	岐阜西工業高等学校	1051教育用電子計算機	嘉穂工業高等学校
〃	津工業高等学校	〃	愛知工業大学
〃	四日市工業高等学校	〃	立命館大学
〃	洲本実業高等学校	〃	名古屋電気通信工学院
〃	神戸村野工業高等学校	〃	墨田工業高等学校
〃	主寺工業高等学校	〃	豊田工業高等学校
〃	正強高等学校	〃	釧路工業高等学校
〃	宇部工業高等学校	〃	岐阜工業高等学校
〃	新居浜工業高等学校	〃	国立長岡工業高等専門学校
〃	御影工業高等学校	〃	可児工業高校
〃	壬生川工業高等学校		

8. 関連会社について

8－1 中央システムエンジニアリング株式会社

8－1－1 会社の概況

1. 会社の名称　中央システムエンジニアリング株式会社
2. 会社の設立年月日　昭和46年12月24日
3. 会社の目的　電子機器，および装置の設計・販売・サービス，不動産，動産の管理に関する事業，およびこれに付帯する一切の業務
4. 代表者氏名　代表取締役　山口　幸也
5. 所在地
   - 本　社　東京都八王子市元本郷町1丁目9番9号　〒192
     電話（0426）23—1211
   - 東京事務所　東京都千代田区神田練塀町55番地　若林ビル
   - 電子計算機センタ　電話（03）253—1664
   - 東京出張所　東京都新宿区市ケ谷本村町28番地　ホワイトレジデンスビル　〒162
     電話（03）269—4801
   - 大阪出張所　大阪市淀川区西中島3丁目8番14号　犬飼ビル　〒532
     電話（06）303—0461
   - 名古屋出張所　名古屋市西区牛島町86番地　カネヒサビル　〒451
     電話（052）571—4501
   - 九州出張所　北九州市戸畑区福柳木2丁目2番地　〒804
     電話（093）881—9832
6. 資本金　授権資本金　12,000千円
   払込済資本金　3,000千円
7. 主な株主　日本トーター株式会社
   中央電子株式会社
8. 主要取引金融機関　東海銀行八王子支店
9. 役員名および略歴

代表取締役	山口　幸也	昭和27年3月	東京大学工学部卒
取締役	坂倉　越夫	昭和13年3月	三重県立松坂工業学校卒
取締役	住田　幸男	昭和19年3月	神戸高等工業専門学校中退
取締役	吉原　亘	昭和34年3月	早稲田大学政経学部卒
監査役	前川　豊	昭和29年3月	神戸大学法学部卒

10. 従業員の状況

	男	女	計
社　員	26	2	28

（昭和50年4月現在）

8－1－2 営業の現況

トータリゼータシステム，自動丸編機システム，普及型健康管理システム，CEC EDPシステム，教育用電子計算システム，自動倉庫システム，土木測量計算システム等に代表されるCEC555電子計算機応用システムのソフトウェアに関する一切のサポート，およびこれを含めた総合的な保守・メンテナンス，また新規に開発される応用システムのソフトウェアシステムの設計を担当している．

現在，新日本製鉄大分製鉄所工場プロセス制御システムをはじめ桐生，津，三国，蒲郡，浜名湖，芦屋，下関等の競艇場，足利競馬場，金沢競馬場のトータリゼータシステム，自動丸編機システム，自動紋紙せん孔システム，その他財務会計処理をはじめとするEDPシステム，土木測量計算用に多

数のシステムが稼動中であり，全般的なこれらのシステムサポートを主体とする営業活動を続けている．

8－2　中央工営株式会社

8－2－1　会社の概況

1. 会社の名称　　中央工営株式会社
2. 会社の設立年月日　　昭和48年12月26日
3. 会社の目的　　電子機器，および装置の設計，製造，据付，施工等にともなう一切の工事
4. 代表者氏名　　代表取締役　村田　肇
5. 所在地
   - 本　社　東京都八王子市八木町3番15号　〒192
     電話(0426)22—6735
   - 東京出張所　東京都新宿区市ケ谷本村町28番地　ホワイトレジデンスビル　〒162
     電話(03)269—4801
   - 大阪出張所　大阪市淀川区西中島3丁目8番14号　犬飼ビル　〒532
     電話(06)303-0641
   - 名古屋出張所　名古屋市西区牛島町86番地　カネヒサビル　〒451
     電話(052)571—4501
   - 九州出張所　北九州市戸畑区福柳木2丁目2番地　〒804
     電話(093)881—9832
6. 資本金　　授権資本金　12,000千円
   払込済資本金　3,000千円
7. 主な株主　　中央電子株式会社
8. 主要取引金融機関　　埼玉銀行八王子支店
9. 役員名および略歴

代表取締役	村田　肇	昭和26年3月	芝浦工業大学電気科卒
取締役	塩田　正訓	昭和38年3月	東京都立第二商業高校卒
取締役	小林　宏	昭和31年3月	電気通信大学卒
取締役	吉原　亘	昭和34年3月	早稲田大学政経学部卒
監査役	波多野森雄	昭和33年3月	東京都第二商業高校卒

10. 従業員の状況

		男	女	計
社員		5	1	6
協力社友	工事系	65	—	65
	検査系	20	—	20

（昭和50年4月現在）

8－2－2　営業の現況

CEC555電子計算機による各種アプリケーションシステム，応用システムの据付工事，施工および現地オリエンテーションを主体とした営業活動が主力である．

特に，近年の情報処理システムはその利用分野の拡大にともない，高精度化と広域化を志向する傾向にあり，設置後の保守，運用を容易にするために利用目的に合致した万全な管理体制による据付施工が望まれている．

当社は，桐生，蒲郡競艇場などのトータリゼータシステム，自動丸編機システム，財務会計処理システム，健康管理センターシステム，運輸省航空局空港管制システムなど多数のシステム，装置の据付施行実績があり，これらの経験をもとに，より高い情報化社会創造の一翼を担っている．

見本

売買契約書

印　紙
５０円

売　買　契　約　書

見本

（以下甲という）と　　*
中央電子株式会社（以下乙という）とは
（仕様書および打合議事録による）*
（以下装置という）の製作および納入に関し次のとおり契約する。

（契約の目的）

第１条　甲は乙に対し本契約条件にしたがい第２条に規定する装置を注文し、乙はこれを製作完成の上甲に納入し、甲はこれに対し代金を支払うものとする。

（装置の明細）

第２条　本契約に基づき製作すべき装置の明細は、次のとおりとする。

（イ）　名称および数量　　*

（ロ）　性能および仕様　仕様書および打合議事録による

（装置の製作）

第３条　乙は、本契約締結後速やかに装置に関する設計図および甲の指示する図面を提出して甲の承認を得るものとする。

２．甲は前項の承認にあたり速やかに乙の意見を聞いた上、装置の設計製作および納入が円滑に遂行できるよう協力するものとする。

（仕様・性能の変更）

第４条　本契約締結後装置を完成するまでの間に、甲または乙が装置の仕様・性能に関する条項を一部変更する必要があると認めた場合、甲は乙の意見を聞いた上でこれを変更するこ

見本

とができる。

ただし、乙が製作に着手した部分については変更はできないものとする。

2. 前項の場合、甲は、乙の意見を聞いて第5条に定める納入期限、第7条に定める装置の代金、その他本契約条項を変更することができる。

（納入場所および納入期限）

第 5 条　装置の納入場所は次のとおりとする。

2. 装置の納入期限は、昭和　　年　　月　　日とする。
製作工程上納入期限が遅延する場合は、予め乙は甲に申し出て、甲・乙協議の上納入期限を変更するものとする。

（納入方法および検収）

第 6 条　乙は装置を前条に定める納入期限までに同条の定める場所に納入し、据付調整をするものとする。

2. 乙が装置の据付調整完了後、甲は速やかに検収を行い、甲が検収合格と認めたとき納入が完了したものとする。

3. 前2項による運搬・据付・調整に要する費用は乙の負担とする。検収に要する費用は甲・乙それぞれ負担する。

（装置の代金）

第 7 条　装置の代金は、総額金　　円也とし、その内訳は別紙見積書のとおりとする。
装置の代金は第4条の定めにより仕様・性能の追加または変更がある場合増減されるものとする。

見本

（装置の代金の支払方法）

第 8 条　甲は乙に対し前条代金前渡金（内金）を含めて次のとおり支払うものとする。

1．契約時　　金　　円　　＊

2．納入時　　金　　円　　＊

3．検収後　　日以内　金　　円　　＊

（所有権）

第 9 条　装置は前条の前渡金に相当する部分について、甲の所有に属するものとし、前条3項の代金が支払われたとき装置は甲の所有に属するものとする。

（危険負担）

第 10 条　納入完了前に装置が滅失・毀損した場合、これによって生じた損失は、すべて乙の負担とする。

ただし、甲の責に帰すべき事由によるときは、その程度に応じ甲がその損失の全部または一部を負担する。

（設置条件）

第 11 条　甲は、装置の信頼性および安全運転を確保するために乙が別に指定する設置条件をまもるものとする。

（保　証）

第 12 条　乙は装置の納入完了までに甲と協力し甲が指定する運用者に対して装置の運転および保守要領の指導を行うものとする。

2．乙は、1カ年間設計製作上に起因する装置の故障に対しては無償にて修理改修を行うものとする。

ただし、外部条件、設置条件または運用上操作上の誤り

見本

による改障については有償とする。
消耗品については甲の負担とする。
納入後2年目以降は甲の意思により別に定める保証契約によることができる。

（新技術の相互協力）

第13　条　装置の納入後乙の責に帰さない事由により装置の改造が必要であると甲が判断した場合、乙はこれを引受けるものとし、改造費用は甲の負担とする。

第 14 条　乙の責に帰すべき事由により装置の納入後装置の改造が必要になった場合は、乙は乙の費用および負担でこれを行なうものとする。ただし、甲の判断により改造費用の50%以内の額を甲が負担することがある。

第 15 条　装置の完成後、甲乙互いに協力して装置の市場開拓をはかり、共存共栄の実を挙げるよう努力するものとする。

（信義の原則）

第 16 条　本契約に定めのない事項について、疑義が生じたときは、信義誠実の原則に従い甲乙協議の上決定するものとする。

本契約締結の証として甲乙記名捺印の上それぞれ各壱通を保有する。

昭和　　年　　月　　日

甲

乙　東京都八王子市元本郷町1丁目9番9号
中央電子株式会社
代表取締役　石井一恵

CEC装置 保守，サービス基本契約書 見本

（以下甲と云う）と中央電子
株式会社（以下乙と云う）とは次の通り保守サービス契約を締結する。

中央電子株式会社

システム保守サービス基本契約書

見本

を甲とし（以下「甲」と云う）と中央電子株式会社を乙とし（以下「乙」と云う）甲と乙の間に次の通り保守サービスについて基本契約を締結する。

第1条　適用範囲

本契約は乙が甲に納入した　　（以下「装置」と云う）が甲により長期間安全確実に運用されて、装置の使用目的を達成するための保守、サービスに適用する。

第2条　保守、サービス契約発効の条件

本契約は乙が装置を甲の指定する場所に納入設置し甲が装置の検収を行った時点より効力を生ずるものとする。

但し甲の指定する場所とは、国内とし、国外については別の基本契約書に基くものとする。

第3条　保守、サービスの種類

保守、サービスは次の項目によって実施するものとする。

第1項　保証修理

甲の指定する場所に納入設置し甲が装置の検収を行った時点から1年以内に生じた設計上、および製造上など、乙の責に帰す事故または、故障について適用し、乙の負担において修理を行う。

ただし次の何れかに該当する場合は、この保証修理の範囲外とし、有償とする。

(1)　甲の不適当な取扱いまたは使用による場合

(2)　乙以外の事由による場合

(3)　乙以外の改造または修理による場合

(4) 指定の設置条件および運用条件を超えた使用による場合

(5) 設計仕様条件を超えた過酷な環境下における取扱い、保管、あるいは使用の場合

(6) 火災、水害、地震、落雷、その他天災、地変による場合

(7) 機構部品、ランプ類、表示器、継電器、ブラウン管などの真空管類および、光電子増倍管、ならびにヒューズ類、スイッチ類などの減耗のはなはだしい部品、またはせん孔用紙テープ、印字用紙、記録紙ならびにタイプライタのリボンテープ、磁気ディスク用カートリッジ、磁気テープおよび保守用オイル等の消耗品。

ただし乙以外の製造業者（以下「丙」と云う）の製造した製品の保証期間および保証範囲は本項にかかわらず甲、丙間または乙、丙間の保守、サービスの契約によることがある。

第2項　不定期点検サービス（通常保守サービス）

第1項の期間を経過したのち、甲は装置の正常な機能を常に監視し異常が発生した時はすみやかに乙に連絡を行い、乙は装置の診断、修理、調整ならびに部品の交換、補充ならびに試験もしくは技術指導を行う場合に適用する。

第3項　定期点検サービス（通常保守サービス）

第1項の期間を経過したのち、甲は装置の正常な機能を維持するため、予め甲、乙間協議の上定期的に装置の診断、調整、清掃および部品の交換、補充ならびに試験もしくは技術指導を行う場合に適用する。

第4項　不定期修理サービス（特別保守サービス）

第1項の期間を経過したのち、甲は装置の正常な機能を常に監視し異常が発生した時はすみやかに乙に連絡を行い、乙は装置の診断、修理、調整を行う場合に適用する。

第4条　保守サービスの料金

通常保守サービスは第3条第2項および第3項と併行するものとし、この場

見本

合の料金は年間当り、工事代金を除いた装置の総金額の４％とし、甲は乙の請求によりサービス開始時点を第１回として６ケ月毎に½の金額を支払うものとする。

特例保守サービスは第３条第４項とし、この場合の料金は、異常発生の都度、甲は乙の請求により支払うものとする。

ただし、第３条第１項(7)に記載の部品、材料については別途実費とし甲は乙の請求により支払うものとする。

なお、この料金は社会情勢の変化等により変更が生じた場合は甲乙協議の上改定されるものとする。

第５条　故障、事故時の回復

甲は装置の故障、事故時はすみやかに乙に連絡を行い、甲、乙間調整の上、原則として、24時間以内に回復を行うものとする。

ただし故障、事故の内容、交通事情、および甲の連絡が乙の休日ならびに時間外の場合はこの限りでない。

第６条　契約の解除

甲あるいは乙は相手方が本契約に定める条項に違反した場合本契約の全部または一部を解除することができる。また甲、あるいは乙は必要と認めた場合相手方と協議の上契約内容の全部もしくは一部の変更解除ができる。

第７条　期　　間

本契約は、契約締結の日から２年間有効とし期間満了１ケ月以前に甲または乙から本契約を終了させる意思を書面を以って通知しない場合は更に１年間存続するものとし、その後も同様とする。

第８条　紛争又は疑義の解決

本契約について甲、乙間に紛争又は疑義を生じた場合には社会信義の原則により協義解決する。

以上の契約の締結を証するため、この証書弐通作成し双方記名捺印の上各自壱通保有するものとする。

見本

昭和　　年　　月　　日

甲

乙　　東京都八王子市元本郷町1丁目9番9号
中央電子株式会社
代表取締役　　石井一恵

CEC555　ミニコンピュータ　ハンドブック

1975年7月　発行
発行者　中央電子株式会社
発行所　中央電子株式会社
〒192 東京都八王子市元本郷町1－9－9
TEL　0426－23－1211
印　刷　株式会社　有朋社

© 中央電子株式会社
（落丁本・乱丁本がありましたらご連絡下さい。また，追補，訂正事項についてはその都度ご連絡いたします。）

CSC 中央電子株式会社

本社工場	東京都八王子市元本郷町1丁目9番9号 〒192 電話 八王子(0426)23−1211(代) TELEX 2862-529
東京営業所	東京都新宿区市ヶ谷本村町28番地 ホワイトレジデンスビル 〒162 電話 東京 (03)269−4801(代)
大阪営業所	大阪市淀川区西中島3丁目8番14号(犬飼ビル) 〒532 電話 大阪 (06)303−0641(代) TELEX 523-4289
九州営業所	北九州市戸畑区福柳木2丁目2番地 〒804 電話 北九州(093)881−9832(代)
名古屋営業所	名古屋市西区牛島町86番地 カネヒサビル 〒451 電話 名古屋(052)571−4501(代)

©中央電子株式会社 1975年7月 Printed in Japan