

経情協46-22(1/2)

一般管理者向け

企業内コンピュータ教育

テキスト (Aコース用)

昭和47年11月

財団法人 日本経営情報開発協会



## は　じ　め　に

本協会では、企業をはじめとするコンピュータの各分野におけるめざましい発展・普及に対処するため、その一環として、企業内におけるコンピュータ教育の重要性を認識し、昭和44年度に「企業内コンピュータ教育委員会（委員長大野達男氏）」を設け、同問題につき調査研究を進めてきた。

44年度は、一般管理者およびコンピュータ専門要員に関するコンピュータ教育の現状を調査した。その結果、一般管理者向けのコンピュータ教育の必要性を痛感し、同カリキュラムを開発することとなり、カリキュラムのスケルトンを作成した。

45年度は、上記スケルトンに検討を加え、カリキュラム概要を作成した（表1参照）。

本年度は、前記「一般管理者向け企業内コンピュータ教育カリキュラム概要」に則り、A・Bコースについてテキストおよびスライドを作成した（表2参照）。これらテキスト・スライドの開発に先立って、46年7月14日から16日までの3日間、委員長を中心として、前年度に開発したカリキュラム概要に即して、「一般管理者向け企業内コンピュータ教育特別講座」を実施した。委員会では、科目別に作業部会（表3参照）を設け、上記講義録および受講者の意見等を参考にして、教材作成のための基本方針を検討した。この基本方針にもとづいて、教材の開発を委員会の構成メンバーによって行なうことを意図したが、各委員とも時間的制約が多く、具体的作業を進めることが困難となったので、やむなく外部へ具体的作業を依頼することとした。依頼先については、委員会の承認を得て、この種の教材の作成に経験と実績を有し、かつ、前年度に「本カリキュラムの概要」の作成を手がけている㈱フジミックに、具体的作業を依頼することとした。各作業部会では、テキスト・スライドの内容の検討および修正を行なった。

本テキストは、上記のような経過により開発されたものを、A・Bコース別に2冊にとりまとめたものである。

おわりに、カリキュラムおよび教材の作成につき、終始ご尽力いただいた委員をはじめ各位の方々ならびに㈱フジミックに対し、厚くお礼申し上げる次第である。

昭和47年5月

財団法人 日本経営情報開発協会

(1)

科目別・コースの目的と時間

科目 \ コース	A	B	C
〔講義〕 1. データ・プロセ シング概説	コンピュータおよびデ ータ・プロセシングに 関する基礎知識を習得 する 5 H	データ・プロセシング・ システムに関する具体 的事例を研究する 3 H	
2. プロジェクト ・マネジメント	プロジェクト・マネジ メントに関する各社の 具体的事例の研究を行 ない、全般的基礎知識 を習得する 9 H		プロジェクト・マネジ メントに関する自社例 を研究する 7 H
3. システム設計	情報処理システムの活 用に関する基礎知識を 習得する 7 H	情報処理システムの活 用に関する技術的問題 を取扱う 4 H	自社における情報処理 システムの活用例につ いて研究する 5 ~ 10 H
4. マネジメント・ サイエンスの応用		マネジメント・サイエ ンスに関する基礎知識 を習得する 4 H	マネジメント・サイエ ンスの応用に関する具 体的事例（自社例およ び各社の事例）を研究 する 6 ~ 8 H
5. カレント・ トピックス		コンピュータに関する 最近の一般的動向を研 究する 3 H	自社に関するコンピ ュータ関係の新技術を研 究する 3 H
6. プログラム 実習			プログラムを実際に組 むことによってコンピ ュータの概念を具体的 に理解する 2 ~ 6 H
	21 H	14 H	23 ~ 34 H
〔自習〕 7. データ・プロセ シング概説	コンピュータおよびデ ータ・プロセシングに 関する基礎知識を補足 する 4 H		
8. システム設計	情報処理システムの活 用に関する基礎知識を 補足する 4 H		
計	8 H	—	—
合計	29 H	14 H	23 ~ 34 H

表 2 本年度に開発された教材

		Aコース	Bコース
講	データ・プロセッシング概説	○ テキスト ○ スライド	○ テキスト
	プロジェクト・マネジメント	○ テキスト ○ スライド	
義	システム設計	○ テキスト ○ スライド	○ テキスト
	マネジメント・サイエンスの応用		○ テキスト ○ スライド
自	データ・プロセッシング概説	○ テキスト	
	システム設計	○ テキスト	

表 3 科目別作業部会構成メンバー

〔データ・プロセッシング概説〕

西村委員、米口委員

〔プロジェクト・マネジメント〕

石原委員、生田委員

角 信 郎 氏 (日本ユニパック(株)システム・サービス統轄部長)

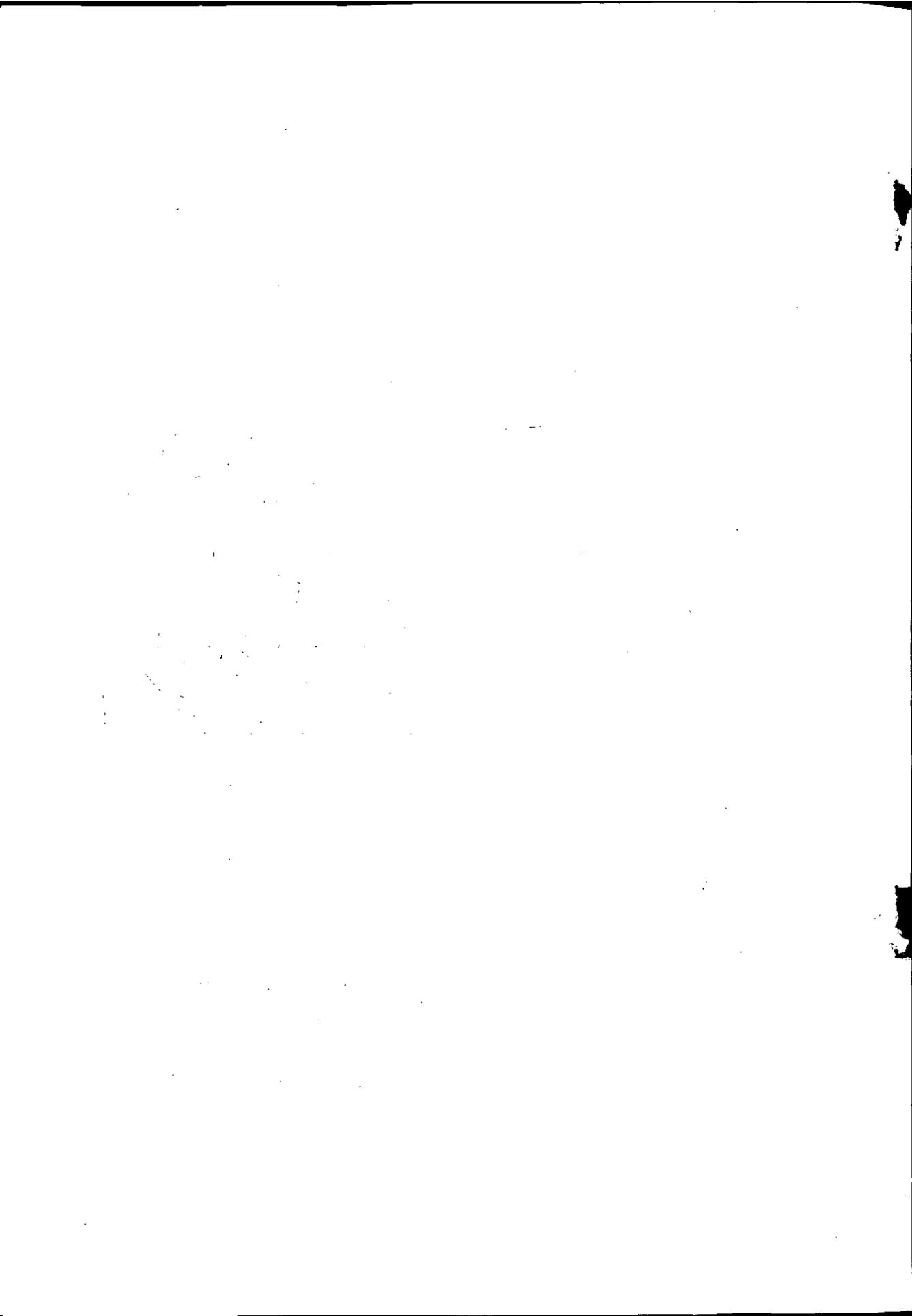
石 川 栄 三 氏 (株日立製作所 東京教育課長)

〔システム設計〕

今村委員、永野委員(代理として山王堂氏)、新井委員、後藤委員

〔マネジメント・サイエンスの応用〕

額田委員、石崎委員、鈴木委員



企業内コンピュータ教育カリキュラム  
作成委員会委員名簿

(五十音順・敬称略)

委員長

大野 達 男 (株)野村電子計算センター・社長

副委員長

石原 善太郎 三井東圧化学(株)・取締役システム部長・海外事業部長

額田 巖 (株)日本経営データセンター・代表取締役社長

委員

新井 進 (株)野村電子計算センター・営業部・課長代理

生田 努 (株)日立製作所・コンピュータ事業部・教育センター部長

石崎 純 夫 (株)富士銀行・業務管理部・部長代理

今村 茂 雄 (財)情報処理研修センター・常務理事

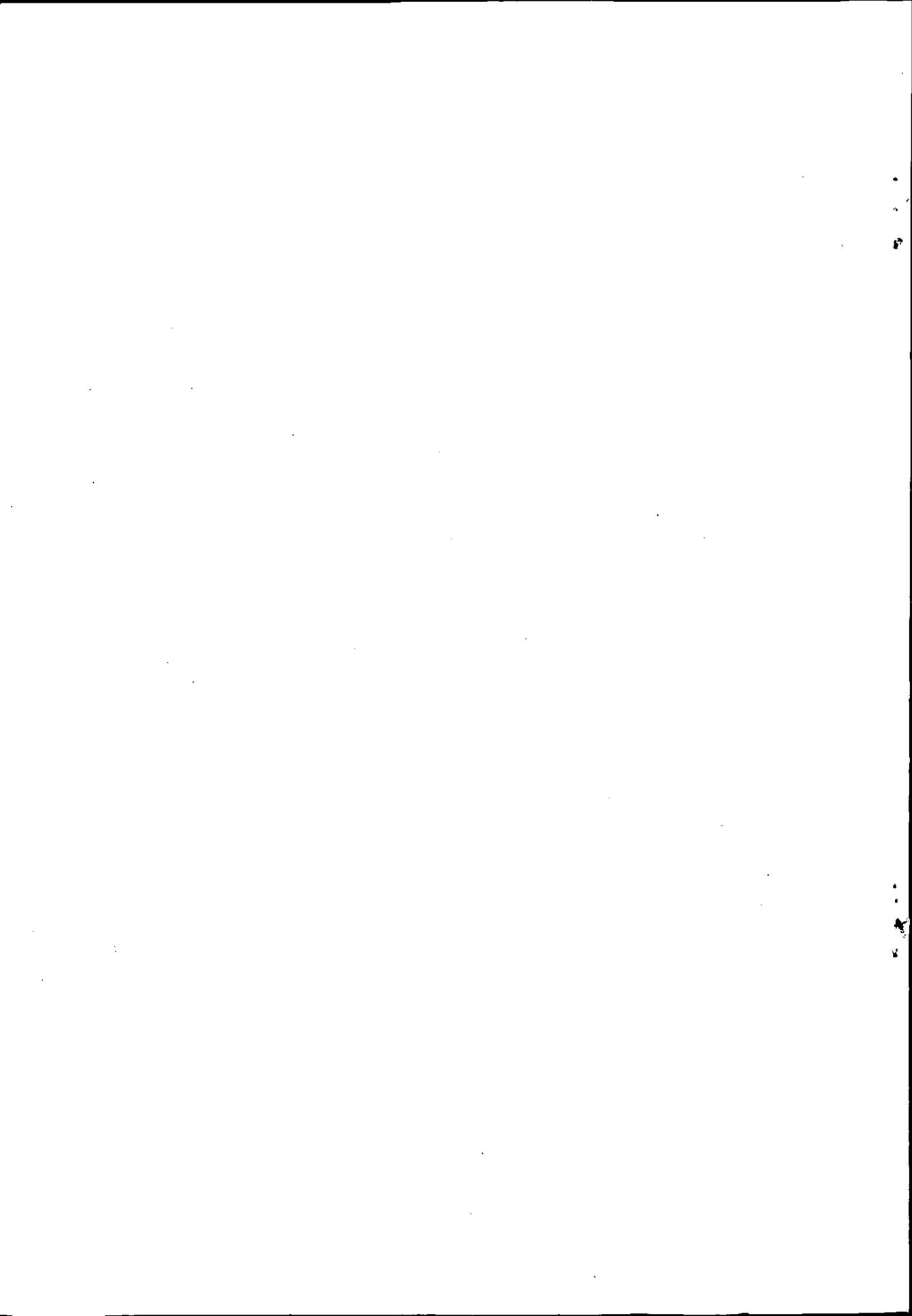
後藤 榛 男 (株)学習研究社・電算植字事業部・課長

鈴木 毅 日産自動車(株)・計算企画課長

永野 元 義 東京芝浦電気(株)・教育研修部長

西村 真一郎 富士通ファコム(株)・取締役・ソフトウェア開発部長

米口 肇 (株)日本ユニバック総合研究所・取締役所長



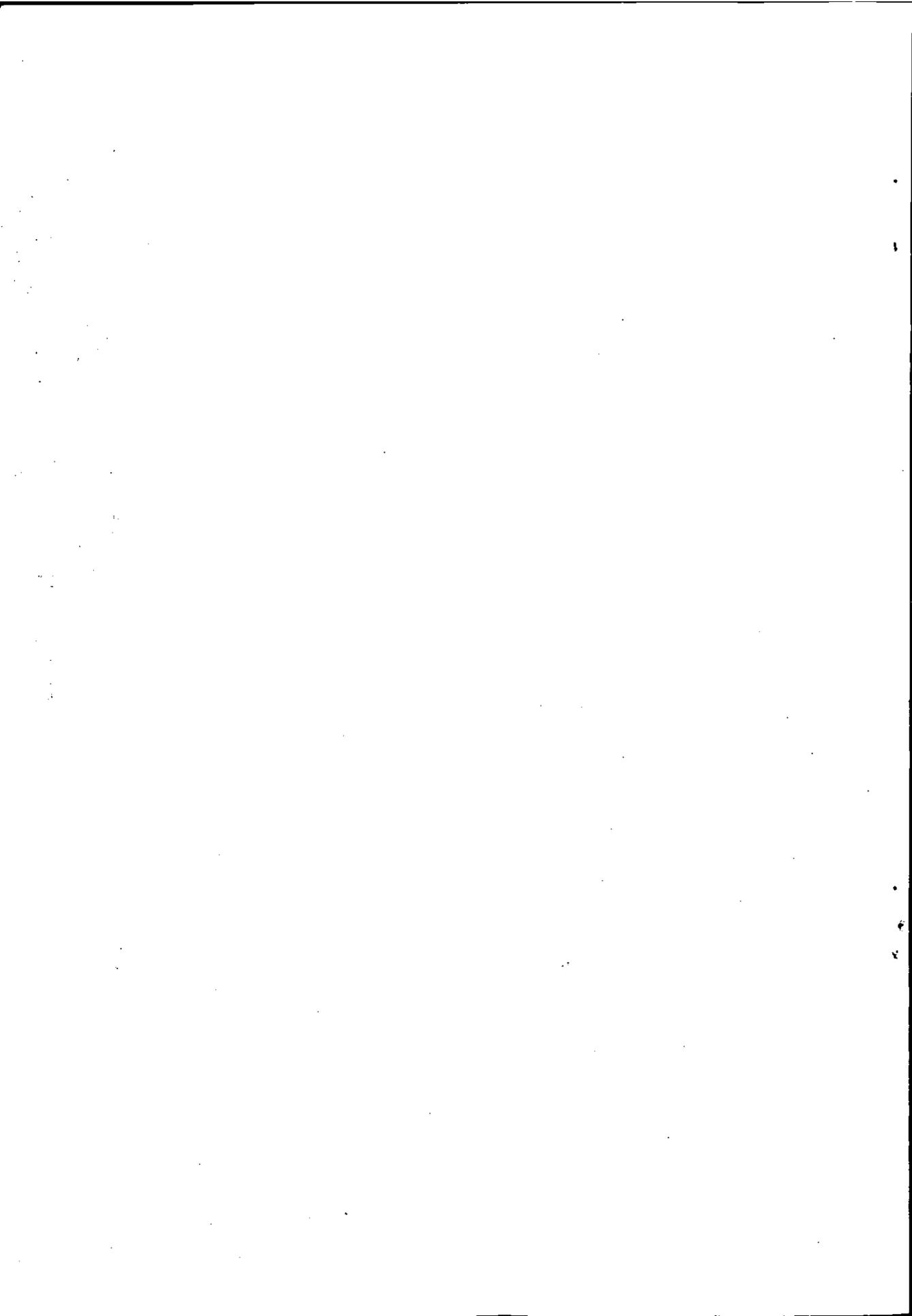
# 総目次

## 〔 A コース 〕

第 1 部	データ・プロセッシング概説.....	3
I	コンピュータ概説.....	3
1.	コンピュータ.....	3
2.	コンピュータの原理と構成.....	16
3.	ソフトウェア.....	32
4.	コンピュータとその周辺技術の発達.....	37
II	データ・プロセッシング.....	45
1.	情報処理の目的と方法.....	45
2.	処理方法.....	53
第 2 部	プロジェクト・マネジメント.....	65
I	プロジェクト・マネジメント概説.....	65
1.	プロジェクト・マネジメントとは.....	65
2.	システム・アナリシス.....	74
3.	プロジェクト・マネジャ, 組織.....	88
II	プロジェクト・マネジメントの技法.....	103
1.	PERT と CPM.....	103
2.	線型計画法.....	107
3.	シミュレーション.....	110
4.	インダストリアル・ダイナミックス.....	115
5.	LP の実例.....	117
III	経営情報システム ( M I S ).....	130
1.	情報化時代と新時代経営.....	130
2.	対象業務システムの分類.....	131
3.	M I S 設計の特徴.....	132

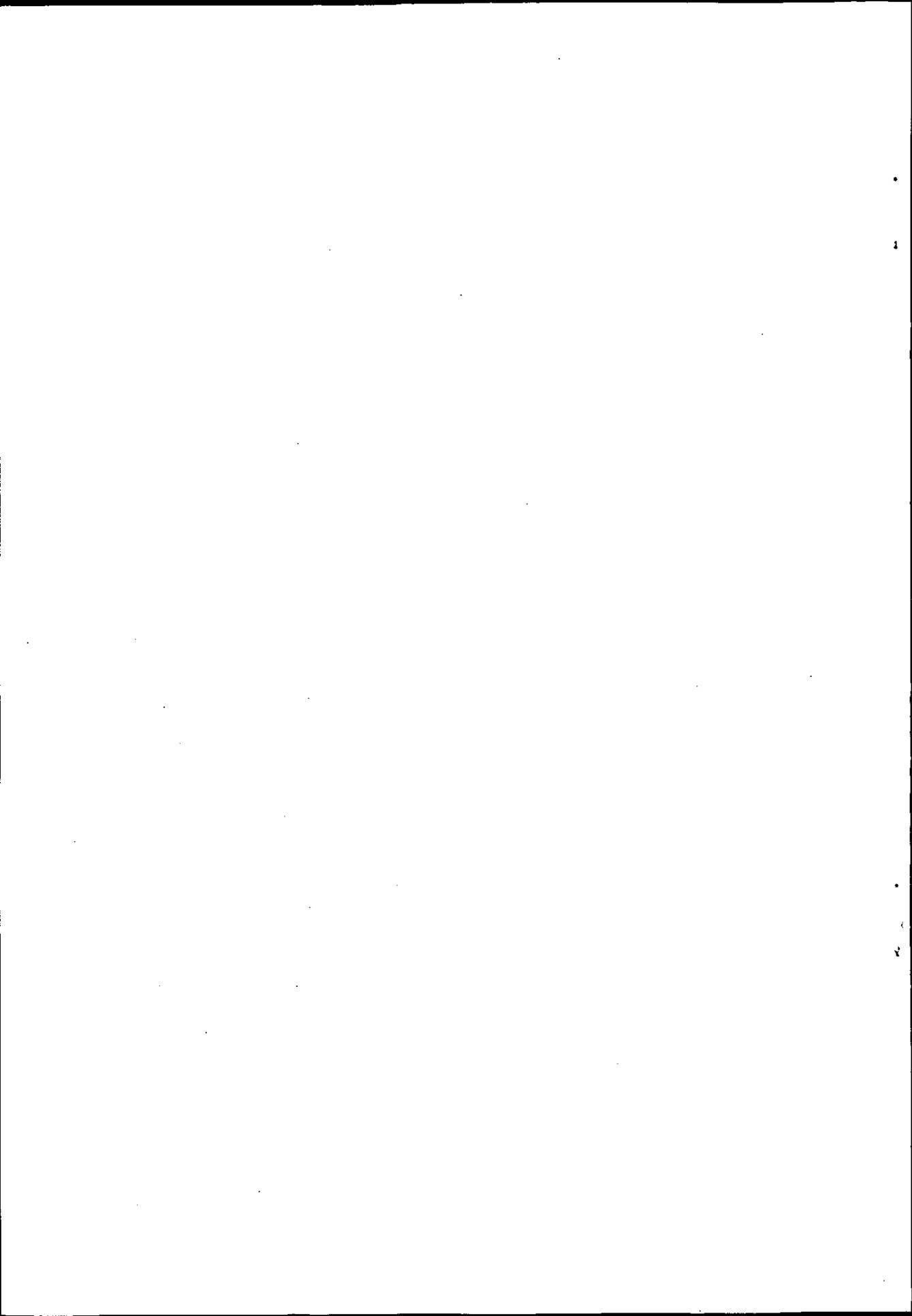
4.	経営管理の機能.....	133
5.	システム弾力化の手法.....	140
6.	M I S 設計の手順.....	147
7.	システム設計の注意事項.....	152
第3部	システム設計.....	156
I	基礎調査段階.....	156
1.	システム設計の目的.....	156
2.	現状調査.....	160
3.	現状分析.....	162
II	設計推進段階.....	170
1.	概要設計.....	170
2.	詳細設計.....	182
III	実施準備段階.....	197
1.	プログラミング.....	197
2.	プログラムテスト.....	199
3.	ドキュメンテーション.....	200
4.	機械導入のための諸準備.....	202
5.	コンピュータ要員.....	205
IV	実施および評価段階.....	207
1.	実施結果の検討と反省.....	207
2.	効果とコストの比較.....	208
3.	システムの改善.....	211
[ A コース自習 ]		
第4部	データ・プロセッシング.....	216
1.	プログラミング.....	216
2.	ハードウェア.....	221

第 5 部 システム設計.....	2 4 0
1. システム設計.....	2 4 0
2. 演習問題.....	2 5 9



第 1 部

データ・プロセッシング概説



# 目 次

I	コンピュータ概説	3
1	コンピュータ	3
1-1	計算作業の構造	3
1-2	コンピュータの歴史	7
1-2-1	卓上計算機の発明	7
1-2-2	プログラムによる制御の自動化	7
1-2-3	パンチカードの採用	9
1-2-4	コンピュータの発明	11
1-2-5	プログラム記憶方式	12
2	コンピュータの原理と構成	16
2-1	情報の表現	16
2-2	演算処理装置	23
2-2-1	主記憶装置	23
2-2-2	演算制御装置	24
2-2-3	入出力制御装置	25
2-3	入出力装置	26
3	ソフトウェア	32
3-1	ハードウェアとソフトウェア	32
3-2	ソフトウェアの構成	32
4	コンピュータとその周辺技術の発達	37
II	データ・プロセッシング	45
1	情報処理の目的と方法	45
1-1	情報の流れ	45
1-2	生産性の向上	47
1-3	経営情報のレベルアップ	48
1-4	新技術の促進	52

2 処 理 手 法 .....	5 3
2-1 バッチ処理 .....	5 5
2-2 リアルタイム処理 .....	5 6
2-3 タイム・シェアリング・システム .....	5 8
2-4 情報の蓄積と検索 .....	6 0

# I コンピュータ概説

## 1 コンピュータ

### 1-1 計算作業の構造

コンピュータは、単純なソロバンではない。あるいは、単に、計算するだけではないということが一般にいわれている。しかし、コンピュータを1つの機械あるいは、道具として見た場合、それはやはり、“計算する機械”あるいは“計算をさせる機械”であることには間違いはない。では、ここで、この“計算をする時につかう機械”についてまず考えてみよう。

われわれは、昔からいろいろなタイプの“計算をする時につかう機械”を持っている。ソロバンをはじめとして、手回しの卓上計算機や、最近では電動の計算機、電子式卓上型計算機といったような、いろいろな段階でいろいろなタイプの機械を持っている。その中で、コンピュータはどのような位置にあるかということをもまず最初に明らかにしたい。そのためには、まず計算というものが、どういうしくみになっているかを最初に考えよう。

計算作業は、一般に、 $+$ 、 $-$ 、 $\times$ 、 $\div$ というような演算をする作業であると考えられている。そして、それも確かにそうだが、計算“作業”ということを考えるならば、実際はそうではない。たとえば、ソロバンを使って計算するというケースを考えてみると、ソロバンはそれ自身演算能力をもっているわけではない。それにもかかわらず、ソロバンを使うと、いろいろな計算作業が能率よく運ぶということは事実である。

そこで、どうしてソロバンを使うことによって計算が容易になるのか、ソロバンを用いての計算作業を調べてみよう。

ソロバンで計算する場合、まず計算するためのデータ、つまりもとの数字を帳簿から読み取る、あるいは読み上げたものを耳から聞くという形で、われわれは指先ではじき入れる。一番最初にソロバンをご破算(クリア)にして、帳簿から読み取る、あるいは耳から聞いて指先ではじき、データをソロバンに置くということが必要になる。そういう動作のことをわれわれは“入力(INPUT)”と呼んでいる(一般にシステムの中に何か外側からデータやパワーを入れることを、通信工学関係で入力ということばを使っている)。

これは単純にデータを入れるということである。また、はじめた結果をみると、珠の形で記録されているというふうに考えられる。あとで振ったり落したりしない限り、ソロバンの珠の形で、永久にはじき入れられた数字は残っている。こういう動作のことを一般に「記憶」と呼んでいる。つまりある時点で入ってきた情報というものは、一定の時間遅れてあとからそれを取り出すことができる。そういうものを「記憶」と呼んでいる。

われわれが実際にソロバンを使って計算する場合、記憶されたデータについて、 $+$ 、 $-$ 、 $\times$ 、 $\div$ ということを指先の動作をまじえて行なうわけである。ソロバンを使う場合、演算をしているのは人間の頭の中である。最終的な結果はソロバンの珠の形で残る。そのままの状態では使いものにならないので、計算した結果を帳簿に書き取るとか、結果は幾らだとか口で伝える。つまり計算の結果をほかに伝えるという動作が必要になる。

一般に情報が持っている性質は、その時点では何の価値を生み出すものでもないが、これを他人に伝えることによって初めて1つの価値を生み出すものである。他のものに伝える動作を「出力 (OUTPUT)」と呼んでいる。そして、入力と出力を一緒にして「I/O」と略称している。

これだけで計算ができるかというとは実はそうではなくて、最も基本的なものが忘れられている。いままでに説明した動作はすべて機械的な動作であるので、これらのことが機械で実現できたとしても、それほど不思議ではない。ところが、帳簿上のどのデータをいつ読みとるか、あるいは、人が読み上げた数字を正確に聞きとってそれをソロバンのどこに置くかというような、計算動作以外の動作を、われわれは自分で考えながらやっている。また、耳から入ってきた「タセ」、「ヒケ」に従って、頭の中にある計算機構が、足し算、引き算を行なっている。となると、計算作業には2つの動作が入っていることになる。

1つは、 $+$ 、 $-$ 、 $\times$ 、 $\div$ という計算そのものであり、もう1つは、それらをどのような順番で行なえば目的の答が得られるか、あるいは、ソロバンのどの桁をつかえばいいかなどの計算作業を「制御」するタイプの動作である。つまり、計算作業は、

○計算動作

### ○ 制御動作

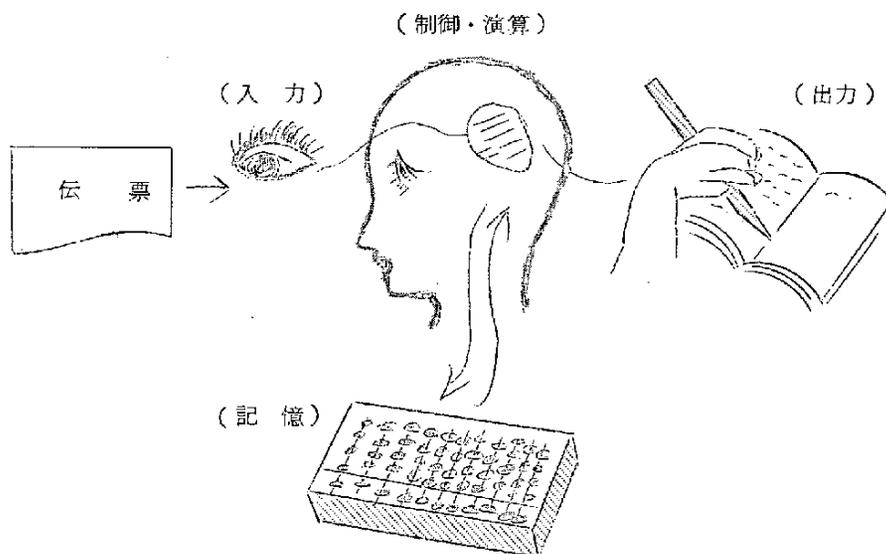
の2つからなっており、制御動作は“ どのようなふうにソロバンを入れる ” というようなことの面倒をみているのである。

ソロバンを使う場合、この制御動作の命令は、人間の脳細胞から発せられ、それが神経を伝わって指先なり、目なり、耳なり、口なりに伝わるという形で動作が行なわれる。いま、ソロバンを例にとって説明したが、暗算で計算する場合でも、道具を使って計算する場合でも、一応計算という作業はこれだけのものから成り立っているということは疑う余地がない。

ソロバンは、以上の動作の中で、記憶という動作だけを機械化しているにすぎない。入力も出力も制御も演算も全部人間がしているわけである。

しかし、ソロバンを使えば非常に計算の効率が上がるということは、計算という仕事の中で記憶という動作が非常に大きな意味をもっているからである。

図 I - 1

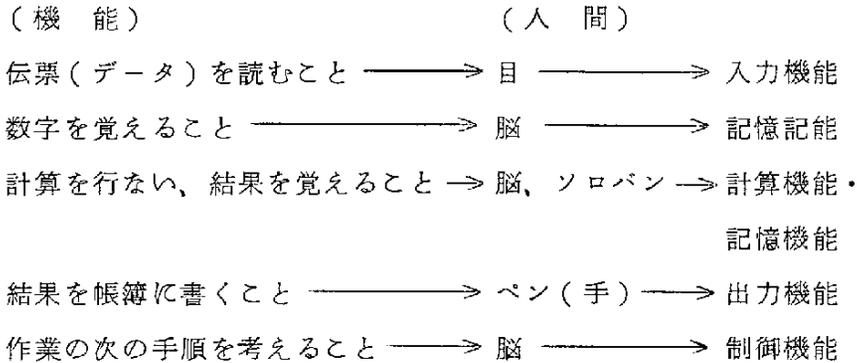


さて、今まで見てきたように、人間が、ソロバンを使って行なう計算作業は、

- ① 伝票を1枚取り上げる

- ② その数字を読んで覚える
- ③ 覚えた数字をソロバンに入れる
- ④ 計算結果を覚える
- ⑤ 覚えた結果数字を帳簿に記入する

となり、この過程で活用されている機能は次の通りである。



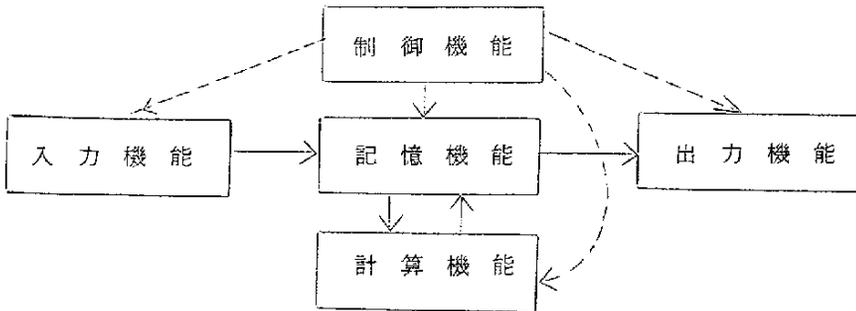
すなわち、入力、記憶、計算、出力の機能と、これらを順序よく使いこなす制御の機能があればよいということになる。しかし、ここでもう一つ考えてみたいことがある。それは、計算を行なうときに、計算順序をだれが考え、だれが覚えていたかということである。いうまでもなく、人間が考え、人間がその頭の中で覚えていた。両方とも人間の役目であった。

そこで、さらに次のようなことを考える。機械に計算の手順を考えろといっても、これはムリだ。だが、機械には覚えるところがあるのだから、人間の考えた一連の計算手続を覚えこませることができるだろう。——データと同じように“記憶”してもらえばいいのではないか。

このようにすれば、後はデータを与えてやればよい。そうすれば、自分が覚えている計算手続にしだがつて、一連の計算を行なり機械になるのではないか。

このような考えに基づいて作られたものが、コンピュータであり、このコンピュータに覚えこませ一連の計算手続のことをプログラムといい、そしてそれを考え出す人をプログラマと呼ぶのである。だから、コンピュータのもっている機能を図に書けば図1-2のようになる。そして、これがコンピュータの基本的な構成なのである。

図 1-2 コンピュータの機能



## 1-2 コンピュータの歴史

### 1-2-1 卓上計算機の発明

ソロバンの歴史も源をたどれば、非常に古く、その原形は北周の時代の文献にも現われており、一説には、さらにさかのぼって、インドに起源を発するといわれている。欧州では、1617年にスコットランド人のナビアが考案した、ナビアの骨と呼ばれる乗算の道具が、計算の機械化の先祖と考えられている。

現在の卓上型計算機の原形が一番最初に出たのは、17世紀半ばで、名前として有名なのは、フランスのパスカルである。このパスカルが、1642年に歯車をつかった加算機を試作したという記録がある。また1671年には、哲学者、数学者として有名なライプニッツが、段付歯車式計算機をつくった。これは、最近までつかわれていた、ユークリッドというメーカーから出された電動計算機に使われているメカニズムと全く同じ設計であり、四則演算力ができる計算機であった。

### 1-2-2 プログラムによる制御の自動化

昔の手回しのものから、いまの電子式卓上型計算機に至るまで、ずいぶんタイプは違うが、共通していえることは、記憶という動作と演算という動作を機械化しているが、制御はまだ機械化されていないという点である。この制御を自動化する機械は、つくれないだろうか考えた人がいた。それが、チャールズ・バベッジというイギリス人で、バベッジは記憶、演算を機械でできても、ふしぎではない。それよりも、人間がやっている制御ということ、うまく機械化できないだろうか考えた。

チャールズ・バベッジの発想は、きわめて自然な形がとられている。ところで、制御というものを機械化する場合に必要なことは、2通りある。1つは、計算しようとする意思である。あるいは、何か問題を解決しようとする意思である。このことは、機械的には絶対に生まれてこないものである。

しかし、意思の部分を除くと、あとはやろうと思った手続を、ただ順番通りにやっているだけにすぎない。したがって、そういう手順を何らかの形で、計算機構の中に覚えさせておき、順番に処理していけば、うまくいくのではないかということが、最初の発想であった。

ところが、この考え方には1つ大きな問題がある。どのような計算でも、実際にデータの大きさを見る前に、一般的にその手続を正確に書くことができない。つまり、途中でデータの大きさによって条件が分かれる場合があるからである。この解決法をバベッジは、次のように考えた。2つのものを比較して、どちらが大きいか小さいか。もっと基本的にいうと、等しいか等しくないかという比較をした。その結果によって、それ以下の計算手続はどのような手順をとるかを選択させた。この選択する手順を拾わせる機械を、従来の四則演算にプラスすると、すべての問題は解決するということを発見したのである。このプロセスを書き上げたものをバベッジは「計算のプログラム」と呼んだ。つまり、これがプログラムのはじまりであり、プログラムによる制御の自動化ということである。

今日では、こういうプログラムによって、制御が自動化されているような計算機械のことを自動計算機(Automatic Computer)と呼んでいる。制御の自動化は、実用的にも非常に大きく、これによって計算の能率は飛躍的に向上するといえる。

バベッジは、実際にこのプログラムを与える方法としてカードに穴をあけたパンチカードを使った。プログラムをカードの上にあらわし、それを機械の中に差し込むと動くというようなプロセデュアを考えたが、実現しなかった。

実際にこういった自動計算機が最初に完成したのは、1944年でハーバード大学で、ASCC Mark I (Automatic Sequence Control

lled Calculator) という計算機である。これは歯車の組み合わせではなく、電話交換機などで古くから使われているリレーという小型のスイッチをうまく組み合わせて、一連の動作を機械化しようということであった。ASCO Mark I では、プログラムは紙テープで与えられており、長い紙テープに順番に計算のプログラムを打ち込んで、それをループ状態にする。その紙テープをかけると順番通り処理していく。一種の外部的な記憶という形でプログラムが与えられたということである。

### 1-2-3 パンチカードの採用

一方、多量のデータを処理する方法として、パンチカードが一般に広く使われるようになったのは、19世紀の後半である。米国の人口統計局にいたハーマン・ホレリスという人がカードを使って各種の統計業務をやらうとした。

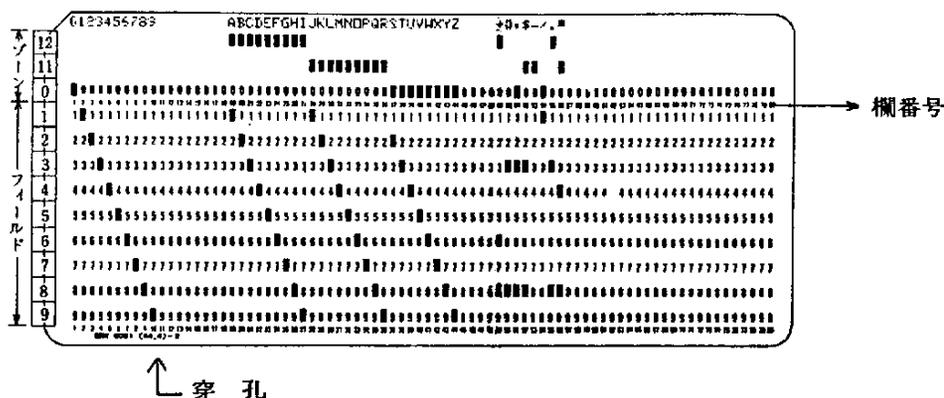
このホレリスが発明したパンチカードを使って、いろいろな統計業務を行なう機械が、後にパンチカードシステム(通常その頭文字をとってPCSと呼ばれている)という一つの大きな事務データの処理機械群に発展してきている。コンピュータが出現する以前には、大量の集計だとか情報の処理はすべて、このパンチカード・システムによって行なわれていた。もちろん、現在でもコンピュータの入力媒体としてパンチカードは非常に重要な、あるいは最も重要な媒体の1つであるといえる。

ホレリスの場合には、大量のデータを速く処理するという目的のためにパンチカードを使用したわけであるが、別な見方からすると、これは非常に重要な意味をもっている。その1つは、パンチカードというのは、カードに穴があいているか、あいていないかということによって情報を表現することである。現在のカードは、ポジションが縦に12段あり、1桁ずつになっている。標準のカードでは80桁のものが表示できるが、この桁にどういふデータが入っているかというのは、この穴がどのような状態であっているかということによってわかるわけである。つまり、かりにあいている状態を1、あいていない状態を0という数字で表現すると、0か1かというような、たった2つの安定した状態しかないようなものを組み合わせて、いろいろな数字なり、情報なりを表現するということである。このように、2つの

状態、1と0を用いて表わす方法を一般に「二進表現(Binary)」と称する。

図 I - 3

穿孔段



よく見かける例でいうと電光ニュースも二進表現になっている。電光ニュースは普通のものとは違って、個々の電球がついているかいないかという、その組合わせによって、図形すなわち文字を表わしている。この二進表現が実際の情報処理の上で採用されたことが、このパンチカードの大きな意味である。

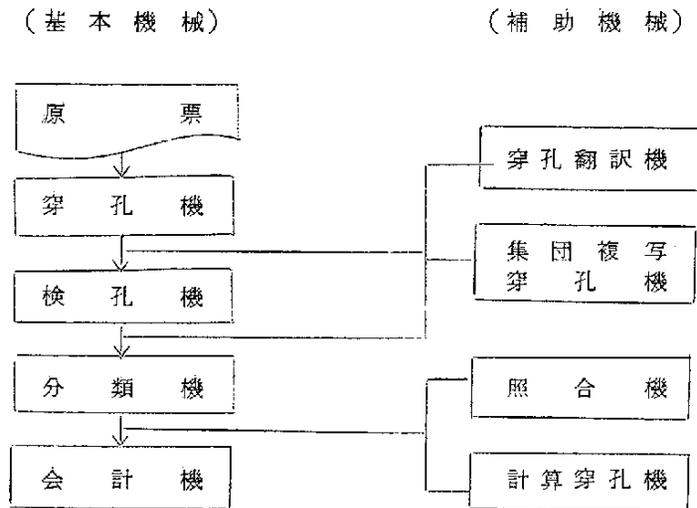
もう1つの大きな意味は、機械相互間で共通に交換し得るようなことばを作ったことである。パンチカードに打たれたことばは、人間のことばや文字ではない。しかし、機械相互間には、それで自由に通用する。PCSの機械を使えば仕事が非常に能率的に運ぶという1つの理由は、こういう共通のことばがあるために、その間で人間が通訳のような形で介在する必要はないからである。データが自動的に交換されていくということに非常に大きい意味がある。

現在では、機械相互間で交換しうるような共通のことばは、パンチカー

ドのことばだけではなく、紙テープのことば、あるいは磁気テープのことばというように、いろいろなことばがある。

このようにパンチカードは、多量のデータを速く処理することができたということとは別に、いま述べたような2つの大きな意味を情報処理の歴史上でもっている。

図 I - 4 P C S 機械組織図



#### 1 - 2 - 4 コンピュータの発明

今までは機械的な自動計算機であったが、1946年に初めて電子部品によるコンピュータが出現した。これはElectronic Numerical Integrator And Calculator というのがもとの名前で、その頭をとってENIACとした。これが世界で最初のコンピュータである(オートマチック・コンピュータというのは、それ以前にハーバード大学でできているので、別にふしぎはないが)。ハーバード大学の機械ではリレーのようなスイッチを使って計算していた部分を、このENIACでは真空管による電子回路に置きかえた。これは電子計算機の歴史上、特筆されるべき事柄であった。たとえば、電子回路に置きかえることによってスピードが速くなった。ハーバード大学のMark Iに比べてENIACは、1,000倍ぐらいの速さになっている。

## 資料 1

### ENIACの仕様

使用真空管	約 18,800 本
所要電力	120 KVA
所要空調設備	24 馬力
演算速度	10 桁 ± 10 桁      0.2 mS
	10 桁 × 10 桁      3 mS
	10 桁 ÷ 10 桁      20 mS

但し

1 mS	1 / 1,000 S
1 μS	1 / 1,000,000 S
1 nano Second	1 / 1,000,000,000 S
1 pico Second	1 / 1,000,000,000,000 S

#### 1-2-5 プログラム記憶方式

この ENIAC で、先に述べたリレーから電子部品への発展の過程にいくつもの新しい考え方が登場した。その一つに、プログラムをシステムの外部から与える方法をあげることができる。それまでのプログラムは紙テープによって与えていたが、ENIAC では穴がたくさんあいたプラグボードに配線をすることによってプログラムを与えた。しかし、この方法だけではいろいろなことが不便で、たとえば計算の途中でプログラムそのものを変更したいということもできない。結果によっていくつもの計算手続の中の1つをえらんだりするだけではなく、プログラムそのものを変更したいというようなことも実際に必要になってくる。このような場合、外部からプログラムを与える方式では非常に不便である。そこで、エッカート、モークリの両名が、この外部からプログラムを与える方法以外になにかうまい方法はないものかと研究していたときに、フォン・ノイマンという有名な数学者が、エッカート、モークリに助言して1つの重要なアイデアを出した。それがプログラム記憶方式である。1945年のことであった。

ノイマンのアイデアは次のようなことであった。人間が何か頭の中で計算する場合、人間の頭の中に記憶されているのはデータだけではなく、どのような方法で計算していくかということも、実は頭の中に入っている。つまり、計算のプログラムそのものも頭の中に入っているわけである。

つまり、記憶装置に入力データや出力データ、あるいは演算の中間結果などをたくわえるだけではなく、演算を進めていくプログラムを同時にここに記憶させておいたらどうだろうかということである。そして記憶されたプログラムを逆に制御装置へ1つずつ呼び出して、その命令に従って制御装置を動かせばよいということである。

その場合、いろいろな演算の手続を命令という形で表現し記憶する。これは命令を1つの情報として貯えるということである。足せ、引け、掛けろ、割れという命令、つまり実際に紙テープに穴をあけるとか、プラグボードにピンを刺すというようなことではなく、これを命令というような情報の形にしてたくわえておき、その命令を制御装置で読みとって実行していくということである。後に、エッカート、モークリは実際にプログラム記憶方式のコンピュータとしてEDVACというものの設計をしたが、実際に完成したのは1951年であった。

実際に完成した最初のプログラム記憶方式の電子計算機は、イギリスのケンブリッジ大学のEDSACという機械で、1949年5月に最初の計算をしたと伝えられている。これはウィルクスという人が作り上げたコンピュータで1949年に完成している。今は一般にコンピュータというのは、このプログラム記憶方式の電子計算機のことをさしている。

資 料 2

電 子 計 算 機 の 歴 史

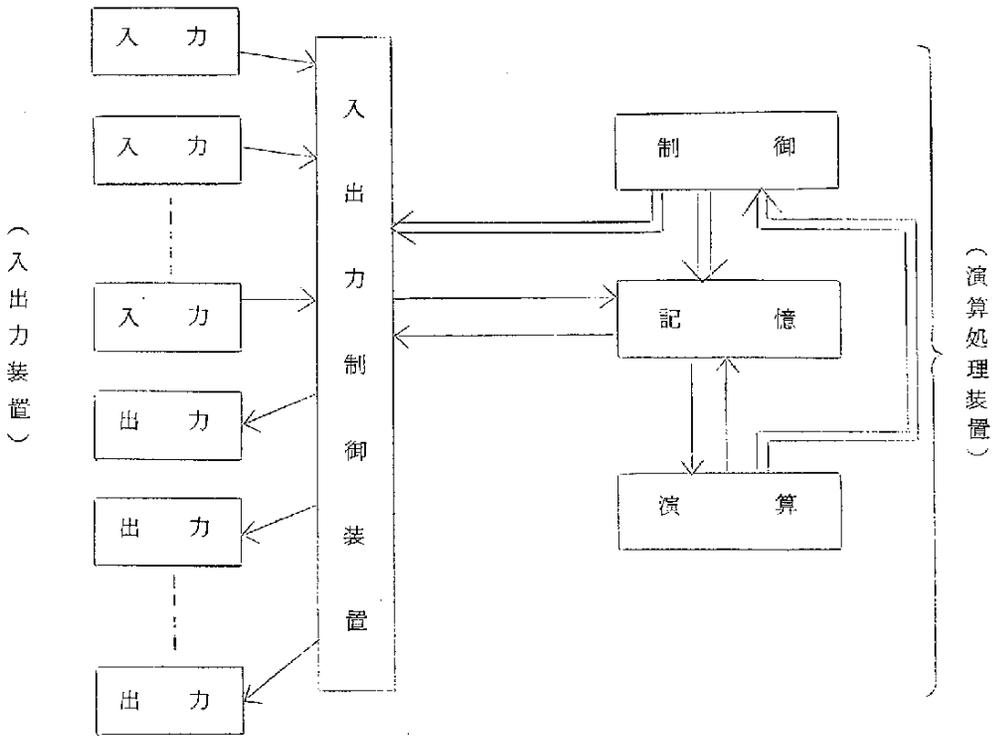
- 古 代 手、指、石などを利用して数字を表現  
石から算盤へ
- 1614年 John Napier ( 1550 ~ 1617 ) による骨牌の発明
- 1642年 Blaise Pascal ( 1623 ~ 1662 ) による10進数歯車の計算機 ( Machine Arithmetique ) の発明
- 1672年 G・W・Lebnig ( 1646 ~ 1716 ) による卓上計算機 ( Steped Reckoner ) の工夫1694年完成
- 1834年 Charles Babbage ( 1791 ~ 1871 ) によるAnalytical Engineの考案
- 1850年 Palmary 計算機の発明
- 1872年 Boldwin リバーシブル四則計算機の発明
- 1884年 Burroughs の作表会計機 ( Listing Calculator ) の開発
- 1889年 H・Hollerith によるパンチカード法の発明
- 1917年 電動式分類機の発明
- 1919年 Eceles & Jordan による真空管計数回路 ( Flip flop ) の発明
- 1919年 Vannerar Bush による大砲射程の研究
- 1938年 G・R・Stibitz が Relay 計算機を提唱
- 1940年 Nobert Winer による自動計算機の構想
- 1940年 S・B・Williams の設計による Model 完成
- 1944年 ハーバード大学 H・H・Aiken の助力を得て IBM は Harvard Mark I を完成
- 1945年 J・P・Eckert および John・W・Mauckly の手によって電子管式本格的電子計算機 ENIAC 誕生
- 1948年 Harvard Mark II を完成
- 1949年 水銀遅延回路をもつ EDSAC がケンブリッジ大学で完成
- 1950年 J・P・Eckert & J・W・Mauckly が BINAC を完成
- 1951年 EDVAC の完成
- 1951年 ブラウン管記憶装置がマンチェスター大学 F・C・Williams & Ferrantige の協同で開発

資料 3

	第一世代 1954 ~ 1959	第二世代 1959 ~ 1964	第三世代 1964 ~ ?
主要電子回路	真空管	トランジスタ	トランジスタ マイクロサーキット
内部記憶装置	磁気ドラム ミリセカンド	磁気コア マイクロセカンド	磁気コア 磁気ナノセカンド シンフィルム
記憶装置の内容	4,000語	82,000語	50,000語
外部記憶装置	磁気テープ	磁気テープ 高価なランダム・ ストレージ	安価なランダム・ ストレージ
プロダクト・ラ イン特性	個別的な無関係 な機械	科学用および商 業用計算の傾向	ワン・システム・ コンセプト
システム特性	パンチド・カー ド装置と単独コ ンピュータ	地点におけるデ ータプロセシン グ大型機と衛星 コンピュータ	統合されたフレ イムワーク 遠隔入出力装置をも つオン・ライン処理
ソフトウェア	機 械 語	アッセンブリー システムズ	タイム・シェアリン グ・システムズ アプリケーション・ システムズ
アプリケーション の型	管理資料給与発 注処理など	オペレーショナ ルな在庫および 生産コントロール	総合的マネジメン ト・インフォメーショ ン・システム

## 2 コンピュータの原理と構成

図 I - 5



コンピュータは同時にいろいろな媒体に記憶されたデータを読み込み、同時にいくつものデータを書き出すことを実際にやらなければならない。

それで一般にコンピュータでは、入口も出口も非常にたくさんある。図 I-5 のように、いくつかの入力装置といくつかの出力装置をまとめて、入出力装置 (I/O) と一般に呼んでいる。一方、制御、記憶、演算という、人間でいえば頭に対応するところがある。この部分を、演算処理装置、あるいは中央演算処理装置 (セントラル・プロセッサ) と呼んでいる。

### 2-1 情報の表現

コンピュータでの情報の表現というのは、どのような形で行なわれているかということを考えよう。これは、基本的には先に述べたパンチカードと同じことである。電子的な回路素子の場合も、安定状態は2つしかない。つまりオン (On) とかオフ (Off) とかいうような形である。オンの状態、オフの状態を利用していろいろな計算を進めていくわけであるから、情報の表現

は2進表現になってくる。一般にコンピュータの回路素子と呼ばれているものは、大ざっぱに分けると3つのタイプに分けることができる。

1つは、論理素子と普通呼ばれている。つまり、足す、引く、掛ける、割る、あるいは、大きいか、小さいか、等しいかを判断するような役割りをつとめているのがこの論理素子である。

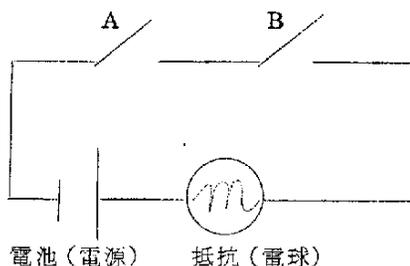
それに対して、電氣的な意味でいろいろな論理素子を補充するような整形素子、あるいは増幅素子といったものがある。電氣的な信号は、たとえばトランジスタや、ダイオード、コンデンサーなどを通過すると、波の形がくずれるとともに抵抗があるので、それだけ高さも低くなってしまふ。そうすると電氣的な信号が一般の雑音と区別がつかなくなってしまふので、1つの素子を通過するたびにうまく形を整えて、かつ、もとの大きさにまで増幅するものが必要で、そのような素子を増幅素子という。

もう1つは、遅延素子、あるいは記憶素子とも呼ばれており、コンピュータの内部でのタイミングの調整に使われている。いろいろな回路を通過する信号は、そのパスの長さによって、あるいは途中に入っているエレメントの数によって時間が狂ってくる。これは真空中を通る速さでは走らないわけである。パスが長くなればそれだけ遅くなり、エレメントが多くなるとそれだけ遅くなるので、1ヶ所に集まったときにタイミングが合わなくなる。それを調整するために早い方を少し遅らせる。そういうものが遅延素子であるが、これが記憶素子といわれているゆえんは、記憶というのは、ある時点に送った情報をタイミングをずらして取り出せるという動作であるので、しばらく遅らせること、つまり待たせるということは、ちょうど記憶という状態と同じである。それで記憶素子という名前が使われている。

電氣的なものの解消のために整形素子や遅延素子などが使われるのであるが、直接演算とかプログラムの制御とかに関与しているわけではない。そういうことをやるのは論理素子である。コンピュータの回路はトランジスタとか、I/O(集積回路)とか、あるいは真空管であるというようなことがよくいわれるが、それはこういう回路素子と呼ばれるものが、半導体であるか、トランジスタであるか、あるいは集積回路であるか、あるいは真空管であるか、ということの意味しているのである。基本的に、論理素子は3種類のも

のから成り立っている。1つはアンド(AND)と普通呼ばれている素子で、スイッチ回路のような形で書くと図 I - 6 のような回路になる。この回路には、スイッチの他に電池(電源)と抵抗(たとえば電球のようなもの)とから構成されている。この回路は、A というスイッチと B というスイッチのどちらも閉じないと回路全体が閉じない。つまり、抵抗を電球だとすると、この電球がつくのは A、B どちらも閉じた場合である。つまり A かつ B ということである。こういう動作は、何もスイッチではなく、トランジスタや真空管を使っても容易に実現できるが、この動作をする素子のことを AND 回路

図 I - 6



といっている。これを回路記号であらわすと図 I - 7 のようになる。

これに対応するものとしてオア(OR)という素子がある。図 I - 8 でいくと電球がつくのは、両方閉じて電球はつくわけであるが、A または B のどちらかが閉じればよいことになる。こういう回路のことを OR 回路といっている。これを回路記号であらわすと図 I - 9 のようになる。

もう1つ基本的なものはノット(NOT)である。いま図 I - 10 のような回路を考える。この場合 A をふさぐと、コイルに電流が流れて電磁石となり、スイッチを吸い上げて B の回路を離してしまふ。つまり、何かインプット・シグナルがあると回路が切れてしまふような回路である。これを NO

図 I - 7 AND 回路

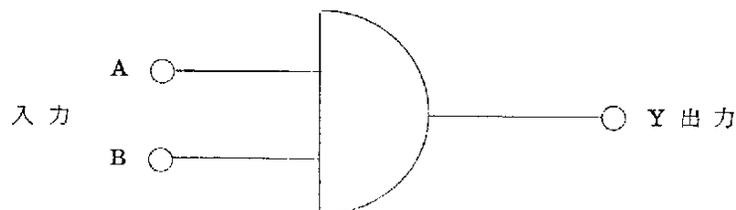


図 I - 8 O R 回路

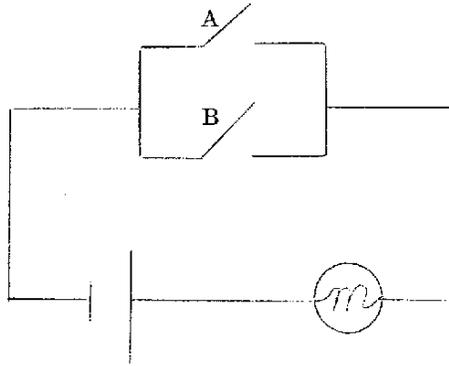


図 I - 9 O R 回路

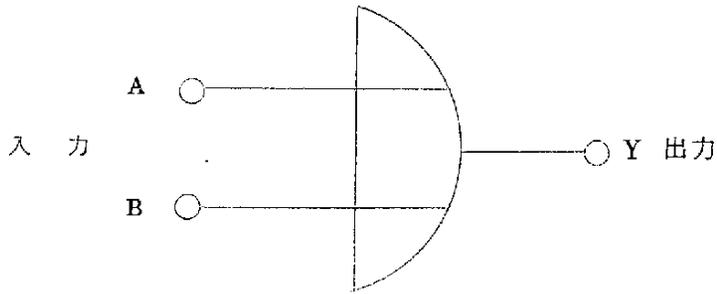
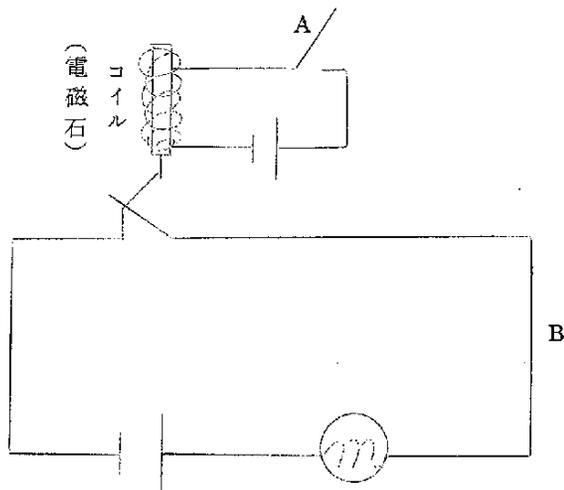
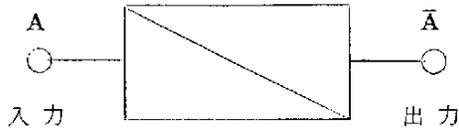


図 I - 10 NOT 回路



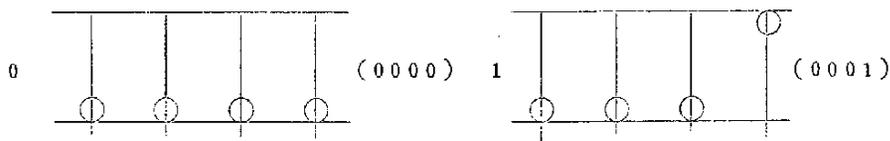
T回路（否定回路）といい、回路記号であらわすと図 I-11 のようになる。

図 I-11 NOT回路

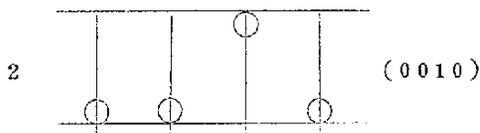


この3種類のものが基本的なタイプであるが、現在では、こういったものを組み合わせたようないろいろな素子が使われている。

次に、こういうものの組み合わせによってどうして計算が行なわれるか、つまり実際に2進法による計算を考えてみよう。



いまここで、4つ玉のソロバンではなく、上図のようなソロバンを考えると、ゼロというのは全部玉が下にある状態をいい、1というのは一番右はじを1つはじけばよいということになる。次に2はどうするかというと、1を加えるのだが、もう玉がないので、つまり4つ玉の9と同じ状態なので、

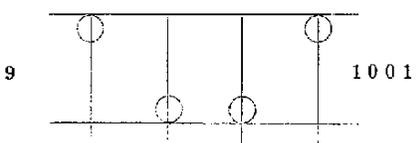
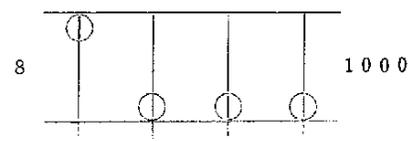
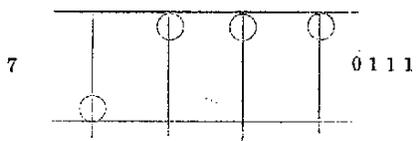
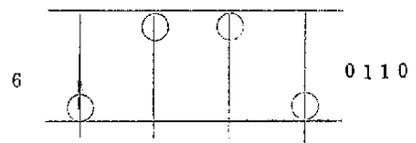
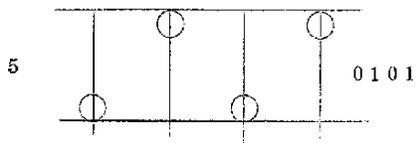
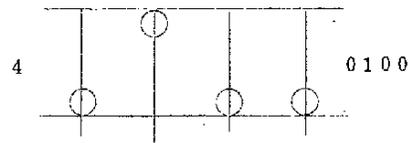
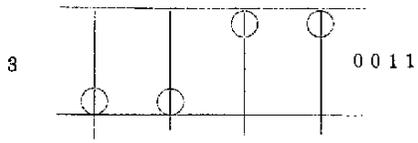


右はじの玉をはらって、次のけたに1を入れるわけである。すると上図のようになる。以下同じようにすると、次頁の図のように、0と1の組み合わせ、つまり2進法で10進全部があらわせる。これでわかるように、2進法演算の規則をまとめてみると、

$$0 + 0 = 0$$

$$1 + 0 = 1$$

$$1 + 1 = 10$$



となる。

実際にこの規則を使って、2進数で10進数と同じような計算ができるかどうかやってみよう。

たとえば、 $3 + 5$  というような計算をやってみると、10進数では

$$\begin{array}{r} 3 \\ + 5 \\ \hline 8 \end{array} \quad \text{となる。}$$

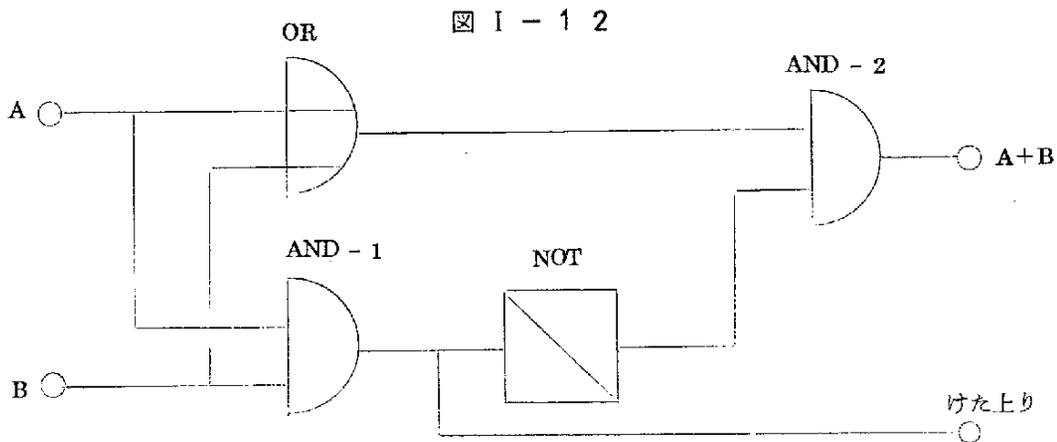
これを2進数で行なうと

$$\begin{array}{r} 0011 \\ + 0101 \\ \hline 1000 \end{array}$$

となり、1000は10進数では8になる。そこでコンピュータでやらなければいけないことは、この演算規則に従うような演算が、論理素子の組み合

わせでできるかどうかということである。これを図 I - 1 2 のような回路でチェックしてみる

$A = 0$ 、 $B = 0$  を考えると、OR 回路からは 0 が出て AND - 2 けた上がり回路に入る。AND - 1 回路からは 0 が出て NOT 回路に入り、そこから 1 が出て AND 回路に入る。そして、AND 回路からは 0 が出て  $A + B$  は 0 になる。けた上がりは 0 である。 $A = 1 \cdot B = 0$ 、 $A = 0 \cdot B = 1$ 、 $A = 1 \cdot B = 1$  も同様である。



コンピュータは、このような回路を数多くつなぎ合わせたものである。

次に、情報の単位について考えてみよう。

0 から 9 までの数字をあらわすためには、2 進法では 4 つのエレメントがあればよい。2 進法のこういって 1 つ 1 つのエレメントを 10 進法のけたと区別するために「ビット (bit)」と呼ぶ。つまり、4 ビットで 10 進法の 0 から 9 までの数字は表現できるということである。ところが、われわれがコンピュータで扱うのは数字だけではなく、英文字や記号など情報の範囲は広く、4 ビットでは足りない。4 ビットで表現し得る情報の種類は 16 種類、5 ビットだと倍になって 32、6 ビットでは 64、8 ビットになると 256 種類の情報を表現できる。最近では、文字をあらわす単位として 8 ビットを 1 つの単位としている。英文字やほかの記号をあらわすときには 8 ビットをフルに使い、数字だけのときは 8 ビットを半分に割って、4 ビットを 1 つの単位として扱うことができる。したがって、数字だけ扱う場合には、8 ビット

トで2けたの数字を入れることができるので、コンピュータ全般として非常に無駄がない。特に8ビットひとかたまりで文字をあらわすというような場合に、8ビットという単位に対してバイトという名前がつけられている。コンピュータによっては、演算の効果を上げるために、2進法の計算をそのまま使っているものもある。一般に、1回に演算し得る単位、つまり長さというようなものを、コンピュータでは、語あるいは「ワード(Word)」という単位で呼んでいる。つまり、コンピュータで1回に取り扱い得る情報の長さを語と称する。これに対して、4ビットとか8ビットなど、語を形成する単位を「キャラクタ(character)」と呼んでいる。コンピュータによっては、語という単位をとらないで、キャラクタだけという単位で計算をしているような機械もある。これは1けたずつ処理をしては元に送り返すというようなやり方をする。その場合、1つの命令で取扱い得るけた数には制限はなくなる。しかし、そのような計算方法をとると非常に遅くなる。比較的小型のコンピュータで記憶容量に余裕がない場合にはこの方法がとられているが、大型の機械では、たいてい語(ワード)という単位で計算をする方法がとられている。その理由は計算が速いからである。

ところで、2進表現や10進表現のように文字や数字といったものを組合わせて表現できるような情報を、一般に、計数型の情報、あるいはデジタル・インフォメーションという。われわれは普通コンピュータと呼んでいるが、それにはアナログ型(相似型)コンピュータとデジタル型(計数型)コンピュータの2種類がある。しかし、普通一般にコンピュータというとき後者をさす。

## 2-2 演算処理装置

### 2-2-1 主記憶装置

演算処理装置にある記憶という機能を果たす部分は、普通、コンピュータでは「主記憶装置」と呼ばれている。単純に記憶装置と呼ぶ場合もあるが、他の記憶装置、たとえば補助記憶装置と区別する場合には主記憶装置と呼んでいる。主記憶装置は、ものをたくさん覚えるためにあるものではなく、ものを考えるためにある。ちょうど人間の脳にあたる部分である。人間の脳というのは、いろいろなことを考え出すことができるが、コンピ

ータの主記憶装置もそれが目的である。

コンピュータが考えるというのはプログラムを収容するということである。複雑なプログラムを収容するためには手続が多いため、それだけ大きな記憶容量がいるわけである。主記憶装置の最大の目的はプログラムの収容ということで、主記憶装置の容量が少ないということは、複雑なことが1度に考えられないということである。したがって、本来1回で処理が終ることでも、2回、3回と分けて処理をしなければならなくなり、手順も複雑になり、また、処理に必要な時間も大幅に増加する。

次に付随する仕事として、入出力データ、あるいは演算に必要な定数、演算の中間結果、こういったことを一時的に覚えるのが2番目の目的である。主記憶装置に対する性能的要請は速いということである。いかにたくさん複雑なプロセデュアを覚えられたとしても、反応の鈍いのは困る。コンピュータの速さは主記憶装置の動作の速さ、つまりものを覚える、あるいは覚えているものを取り出すための速さのことである。最も速いものでも、0.1マイクロセカンド、つまり100ナノセカンド前後の速さしか出せない。それ以上のものは実験的にしかまだ存在していない。

## 2-2-2 演算制御装置

演算制御装置というのは、演算装置と制御装置をひっくるめていっているが、これは実際にプログラムで与えられたことを実行していくということ、つまり四則演算および論理演算の実行が1つの大きな目的である。

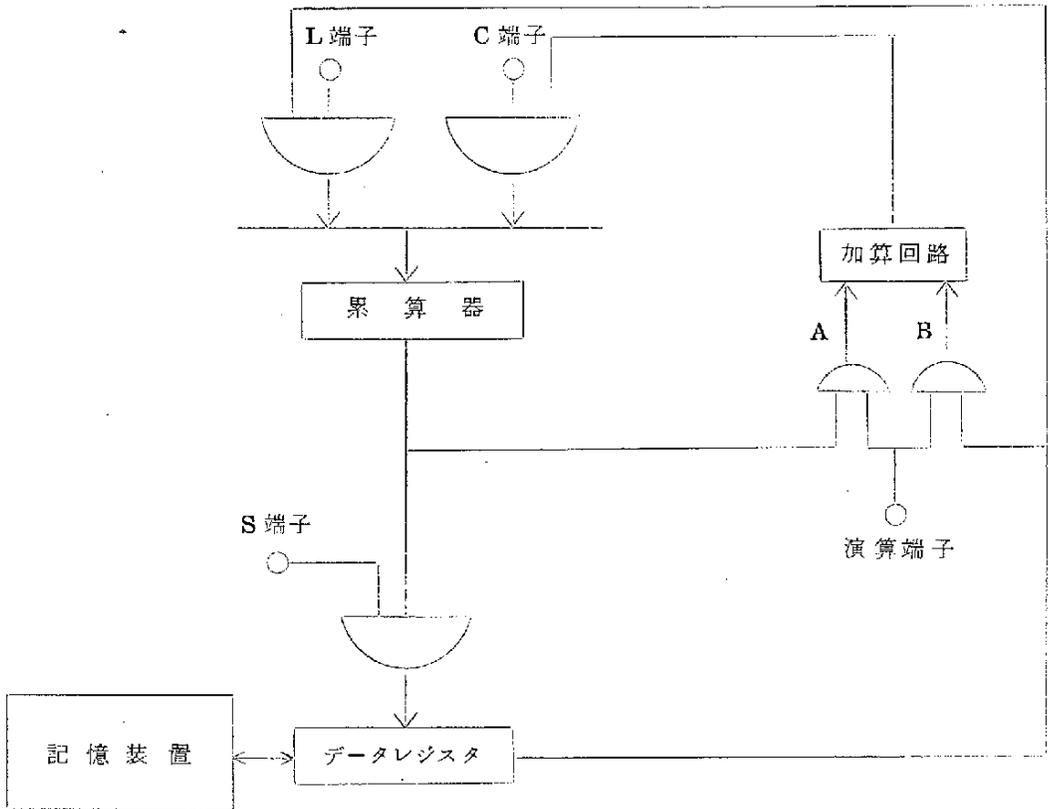
それ以外に、各種の誤動作、つまりいろいろな間違いが起こったときの検出処理をすることも重要な仕事である。コンピュータでは非常に多くの仕事をするので、そのためのいろいろなチェックは自動的に処理しないと困るのである。

演算装置は、四則演算を行なう加算回路、レジスターの1種で演算の結果を蓄える累算器、シフトレジスターから構成されている。

$X + Y$  の演算について、演算の仕組みを考えてみよう。

まずデータ  $X$  が記憶装置からデータレジスターにより出される。L端子に電圧がかかるとデータレジスターの内容、すなわちデータ  $X$  が累算器に入り、次にデータ  $Y$  が記憶装置からデータレジスターにより出される。

図 1 - 1 3



端 子

- L : Load 端子
- C : Calculate 端子
- S : Store 端子

加算回路の演算端子に電圧が加えられると、累算器内のデータ X が加算回路の A 側から加算回路に入り、データレジスタ内のデータ Y は、加算回路の B 側から加算回路に入る。そこでデータ X とデータ Y が加えられ、C 端子に電圧がかかると加算された結果が累算器に入る。したがって累算器に最初に入った内容データ X は加算結果  $X + Y$  によって置きかえられる。S 端子に電圧がかかると、累算器の内容 ( $X + Y$ ) が記憶装置の定められた記憶場所に蓄えられる。

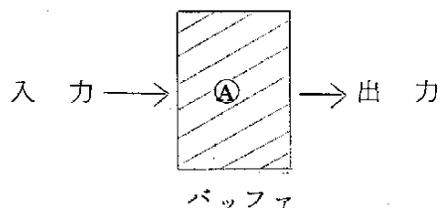
### 2 - 2 - 3 入出力制御装置

入出力制御装置は、入出力機器や補助記憶装置、通信回線などを制御するものである。入出力装置から送られてくるデータ、あるいはそこへ送り

出すデータをうまく交通整理して、正しく取り扱われるようにするものである。

もう1つの重要な働きは、タイミングの調整である。コンピュータの演算処理装置は、大体100万分の1秒ぐらいの速さで動いているが(あるいは最近のコンピュータでは、もっと速い1,000万分の1秒ぐらいの速さで計算している)、入出力装置はそれほど速く動かない。カードで1分間に1,000枚読むのが限度である。磁気テープの装置のような速いものでも、1秒間に20万~30万字ぐらいしか読めない。このようにコンピュータの演算速度と入出力装置との速さの間には、大きなアンバランスがある。国鉄のみどりの窓口のように実際に人間が手でタイプし、コンピュータにデータを送るという場合になるともっと遅くなる。このアンバランスをうまく調整するということが必要になる。

次にこのアンバランスを調整する方法を考えてみよう。下の図の入力側



からポツポツと単発にデータが送られてくるものを、1度Aで蓄える。ある程度たまったところで、これを一斉に演算処理装置に短いタイミングで送ってしまう。たとえば、データを全部送るのに1秒間かかったとすると、1秒間A(バッファ)へ蓄えておき、この間セントラル・プロセッサとは縁を切る。1秒間のデータがたまったところで、1ミリ秒なり、数10マイクロ秒の速さで一斉にたくわえたデータを主記憶装置に移す。このように、ある時間データなどを蓄えることをバッファリングと呼んでいる。入出力装置が非常に遅く、演算処理装置が非常に速い場合、その時間差をバッファを設けることによってうまく調整するという仕事も入出力制御装置の部分では、やっているのである。

### 2-3 入出力装置

入出力装置は、人間とコンピュータとのインターフェースになるので、

これが不便だと中央処理装置の性能がどんなによくてもうまいかない。したがって非常に大事な部分であり、それだけ多種多様に富んでいる。入出力機器、つまり実際にデータを出し入れする機械は2つに分かれている。1つは一括処理（バッチ処理）のための入出力機器である。一括処理とは、同種類のデータをまとめて一度に処理するものである。多くの企業で多量のデータを短時間で処理するために使っている方法である。入出力装置には、カード読取装置、カード穿孔装置、紙テープ読取り装置、紙テープ穿孔装置、光学文字読取装置（OCR）、磁気文字読取装置（MICR）、ライン・プリンター、プロッター、ドラフターなどがある。

バッチ処理をする入出力機器の要件の1つは早く処理できるということである。2つには信頼性が高い、つまり故障しないということである。早く処理できることと、堅牢であるということが非常に大きな要件として要求される。

もう1つの入出力機器には、会話型処理あるいは対話型処理の入出力機器がある。国鉄のみどりの窓口のように、お客さんの要求を中央のコンピュータに送って、その結果を受けとるといようにコンピュータとみどりの窓口にある機械が対話をする形で処理を進めていく。このような処理のしかたを対話型の処理、会話型処理、カンバセーショナル・プロセスという。

この会話の処理に適している入出力装置として、タイプライターや映像表示装置がよく使われている。これはテレビのブラウン管と同じで、CRTディスプレイ（カソードレイチューブ・ディスプレイ）といわれている。

会話型処理における最大の要件は人間が直接さわれるので、まず使いやすいということである。堅牢さとか速さということも必要であるが、業務目的に合い使いやすいということが会話型端末機器に対する一番大きな要請である。したがって一般に個々の業務目的に合った特別な機械がつくられるというケースが多い。たとえば、国鉄のみどりの窓口にある機械、銀行の普通預金口座を扱うテーラー・マシンなどがそうであり、一般に端末機械あるいは端末機器と称している。またこれら端末機器は通常コンピュータと通信回線で結ばれている。

コンピュータ本体の性能が高いということはもちろん重要であるが、それ

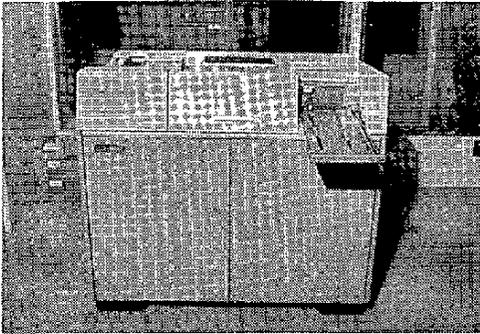
と並行してもう1つ大事なことは、端末機器がいかに使いやすく優秀であるかということが大きな問題になってくる。さらに業務に適應したものであるということや、使用台数が多いため値段の安いということが要請される。

コンピュータの構成で大事なものに補助記憶装置がある。情報処理システムを構成する大事なものとして、ファイル(台帳)があるが、そういったものを蓄えるのが補助記憶装置である。何億字、何十億字という、たくさんの文字をたくわえる装置がこの補助記憶装置である。補助記憶装置は入出力機器と同じように必要に応じて何台も付け加えることができるようになっていゝる。補助記憶装置には、磁気テープ装置のように多くの文字を収容することができるが、必要な情報を直接取り出すことができないものがある。これは録音テープと同じで、巻かれているテープの最初から順を追って、情報を取り出したり入れたりすることしかできないものである。このような形でファイルするものをシーケンシャル・ファイルと称する。シーケンシャル・ファイルの形で取扱うだけでは不便なことが多いため、直接、情報を読み書きできる補助記憶装置が非常に重要なポジションを占めるようになってきた。こういうものを、ダイレクト・アクセス・ストレージ(直接呼び出しができるストレージ)と一般に呼ばれている。あるいは、ランダム・アクセス・ストレージという名前で呼んでいる。これには、磁気ディスク、磁気ドラムといったようなタイプのものが使われている。それぞれ大きさによって、あるいは記録密度によって容量は違いますが、何十億字というものが記録できるダイレクト・アクセス・ストレージが最近ではつくられている。

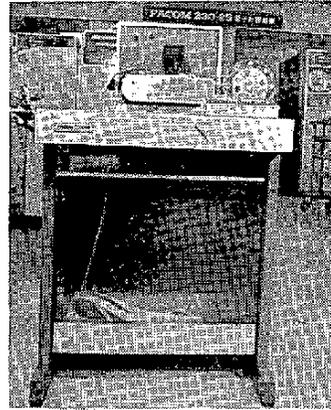
資料 4

1) 入出力装置

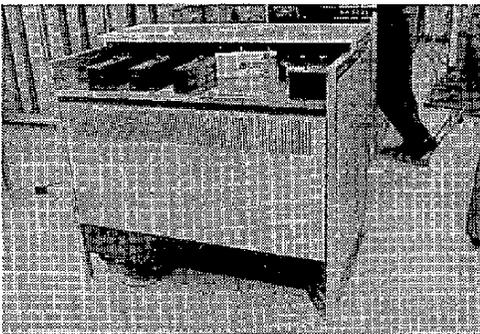
カード読取装置



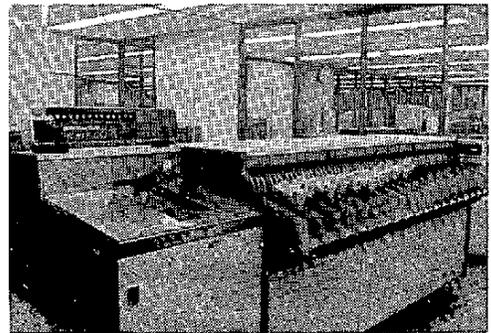
紙テープ読取装置



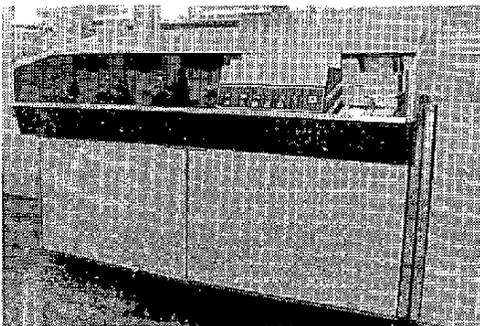
光学文字読取装置 (OCR)



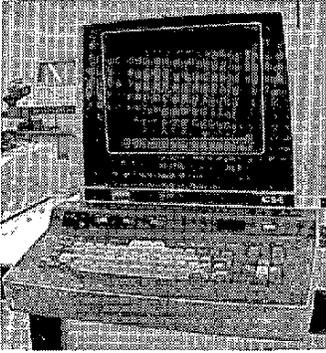
磁気文字読取装置 (MICR)



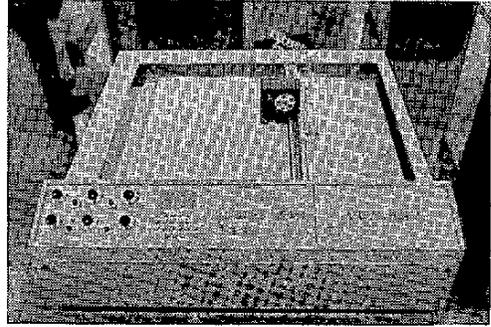
光学マーク読取装置 (OMR)



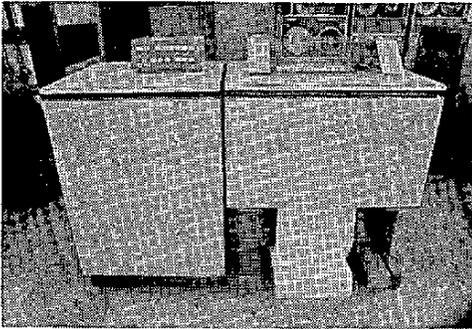
映像表示装置



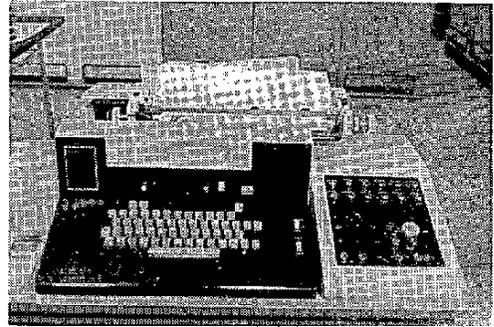
プロッタ



ライン・プリンタ

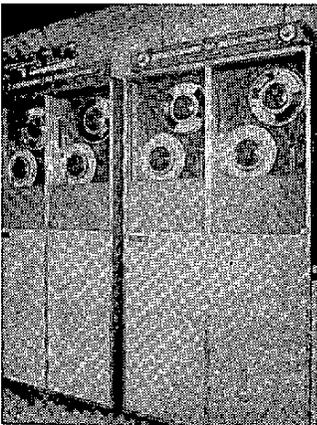


コンソール・タイプライタ

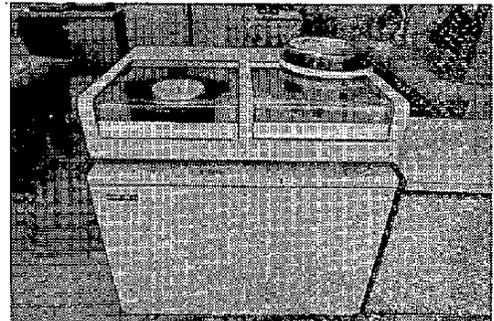


2) 補助記憶装置

磁気テープ装置



磁気ディスク装置



3) カード・穿孔テープ・磁気テープの比較表

項 目	媒体の種類			
	カ	ー	ド	
情報の記録密度				
入出力速度	入	力		
	出	力		
記録の独立性				
記録の保存				
記録の長さ				
媒体の値段				
装置の値段				

○ — 優れている    △ — 普通    × — 劣っている

4) 補助記憶装置の比較

記憶装置	アクセスタイム	容量	装置の大きさ	1けた当りの価格
磁気ドラム	約 10 mS 小	小	大	高 価
磁気ディスク	約 10 mS	↑	↑	↑
磁気テープ	大きい 大	↓	↓	↓
		大	小	安 価

### 3 ソフトウェア

#### 3-1 ハードウェアとソフトウェア

コンピュータがほかの機械と大きく違うところは金物（ハードウェア Hardware）、つまり機械そのものを買ってきて、それに電源を通じただけでは全く仕事をしないということである。それでは仕事をさせるためには何が必要かという、ハードウェアを実際に動かすプログラムである。このプログラムやコンピュータの使用技術のことを総称して、ソフトウェアという。

コンピュータができたごく初期のころは、仕事があるごとにそのプログラムを書いていた。ところがだんだんとコンピュータが方々で利用されるようになると、プログラムを書くという作業が非常に大きな負担になってきて、仕事がいまぐれいなくなる。また、同じようなプログラムをいろいろなところで、同じような苦勞をしながら作成するという無駄が生じてくる。それから、特別な技術が必要とするようなプログラムは特定のところでしか作れない。そこで、プログラムを標準的なもの、一般的なものとして作り、どのコンピュータでも使えるようにしようということにソフトウェアの発想がある。

ハードウェア：コンピュータを構成する機器、電子回路など

ソフトウェア：コンピュータを活用するために必要なプログラム使用技術など

#### 3-2 ソフトウェアの構成

コンピュータに附随している直接的なソフトウェアは次のような構成になっている。

##### 1) 言語処理プログラム

プログラムを作成することのむずかしさは、大きく分けると2つある。

その1つは、実際にコンピュータで処理する仕事の手順を正しくフローチャート（流れ図）に書きあげることである。これが非常にむずかしいのである。実際にプログラムを作成する作業のうち8割位がそれにかかっているといっても過言ではない。

もう1つのむずかしいことは、書きあげられた手順をコンピュータにわかることばであらわすということである。コンピュータのプログラムは、

プログラム記憶方式の場合、命令の形で書かれるが、その命令はわれわれ普通の人間には非常にわかりにくいものである。たとえば、足し算や引き算の命令もその通り書くのではなく、0、1の組み合わせで書いていく。この0、1で書いたことばが、直接コンピュータがわかることばである。ところがコンピュータのことば、つまり0と1の組み合わせを勉強するという事は非常に困難である。また人間の場合で考えると、日本人とアメリカ人がお互いにインフォメーションを交換する場合、日本語でも英語でもないもの、たとえばエスペラント語のような第3のことばを使ってもお互に通じるはずである。コンピュータの場合にもそれと同じようなことをやろうというわけである。人間のことばでもない、コンピュータのことばでもない中間のもので、人間には使いやすくし、コンピュータにもわかるようにする。このようなことばを使用すれば、プログラムを作成する作業が楽になる。このプログラムのための専用のことばを、一般にプログラム言語と称する。これはいうまでもなく、自然言語ではなく、文法的に見てもプログラム用のことば、人工のことばである。人間はプログラム言語を使ってプログラムを表現する。現在プログラム言語として、一般によく知られているものとして、COBOL語(Common Business Oriented Language)というものがある。これは主として事務計算に使われている。また、科学計算用としてFORTRAN(Formula Translation)がある。その他にALGOL語、PL/1、RPGなどのことばがあるが、これらはいずれもプログラム言語である。いまあげたのは、一般にレベルが非常に高く、人間のことばにわりあいに近いことばである。以上のようなプログラム用言語を使ってプログラムを書く。そのプログラムをコンピュータのことばで書かれた、つまりコンピュータの命令で書かれたプログラムに翻訳することが必要になってくる。この翻訳するプログラムのことを言語処理プログラムと一般に呼ぶ。

## 2) サービス・プログラム

実際にコンピュータを使用する立場から見ると、どういうプログラム言語でプログラムを書くかということが最も大事なことである。

もう1つ非常に大事なことは、コンピュータの入出力データの編集だと

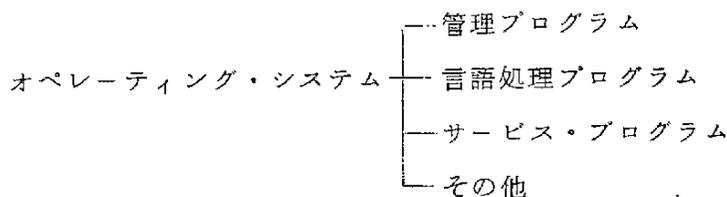
か分類だとか、入出力装置の制御、ファイルの処理、こういったものほどのようなデータ処理をする場合でも必ず必要になる。このようなプログラムは、標準的なプログラムとして作っておけば便利である。パッケージ化してあらかじめ準備してあるこのようなプログラムのことを、サービス・プログラムと呼んでいる。

### 3) 診断プログラム

ディバグ用プログラムなどコンピュータにどこか悪いところはないかというようなことを診断するために使われるプログラムが、診断プログラムである。また、実際に書かれたプログラムのどこかに間違いがないかどうか、というようなことをチェックするのがディバグ用のプログラムである。

### 4) 管理プログラム

最近のコンピュータでは、管理プログラム、あるいは制御プログラム、一般にコントロール・プログラムという名前と呼ばれているものがある。このプログラムによって統一的に処理が行なわれる。つまり、各装置に対する仕事の割り振りを制御し、その潜在的な能力を最も効果的に使えるようにするプログラムである。管理プログラムができたことによって得られる最大のメリットは、オペレータによる仕事の中断が少なくなったということである。仕事の順序(ジョブ・ストリーム)を、このコンピュータにあらかじめ与えておくと、管理プログラムは順番にピックアップして行って、オペレータの中断なしに次から次へと仕事をやっていく。ソフトウェアの中で、管理プログラムによって引き入れられた言語処理プログラム以下のプログラムで、1つの大きな体系になっているわけである。これを一般にオペレーティング・システムと呼んでいる。



### 5) オペレーティング・システム

オペレーティング・システムというのは、管理プログラムがあって、

その下に言語処理プログラム、サービス・プログラムなどの体系をもっている。プログラムのサイズとしても非常に大きなプログラムになる。

#### 6) アプリケーション・プログラム

ユーザー・サイドから見て大きいのは、アプリケーション・プログラムである。実際にコンピュータで仕事を行なうのに必要なプログラムである。これは大体2通りに分けることができる。

アプリケーション・プログラム  $\left\{ \begin{array}{l} \text{プログラム・ライブラリ} \\ \text{ユーザー・プログラム} \end{array} \right.$

1つはプログラム・ライブラリと呼ばれているものである。特定の大きな内容をもった対象業務を共通に処理しうるように作成したプログラムである。ある一定の計算手順で一定の結果が得られるものをあらかじめプログラムしておく。ちょうど既製品を使用する時の考え方と同じで、必要な時必要なプログラムをとり出して使用することができる。特に1つのプログラムで大きなものをつくるには、いろいろな技術がいる。特別な計算技法や仕事そのものに対するノウハウが必要であるといった場合、プログラム・ライブラリは、非常に貴重な価値をもってくる。現在標準的なプログラム・ライブラリが作成されている。大きなソフトウェア・パッケージとしては、情報管理の分野、データ・ベースの管理、計量経済予測、いろいろなものを予測するためのプログラムシミュレーションのためのプログラム、あるいは統計解析、数理計画、数値制御、構造解析などいろいろな大きなパッケージの開発が進んでいる。

アプリケーション・プログラムのもう1つの分野は、ユーザーズ・プログラムである。ユーザーが自らの企業の中で開発したアプリケーション・パッケージである。

以上述べたように、ソフトウェアはいろいろな目的のためのプログラムの集団としてみてもいいわけである。このソフトウェアの問題点は、まず第1にハードウェアとのバランスである。中型や小型の機械で、非常に大きいソフトウェア・パッケージをつくってみても意味がないということである。

次に大事なことは、プログラムの標準化ということである。特にプログ

ラム言語の標準化は非常に大事なことである。つまり、どのメーカーのどの機械でも、このことばでプログラムを書けば安心だというような標準語がほしいことである。現在、標準語として、一般に認められているのは、COBOL、FORTRAN、ALGOL、PL/1 ということばである。特にCOBOLとFORTRANは国際的に大体1つの標準はできあがっている。標準語でプログラムを書けば、多くのコンピュータでそのままそのプログラムを使うことができるようになる時代が、近い将来実現するであろう。尚、日本においては、FORTRAN、ALGOLがJISに定められており、COBOLのJIS化も近い。

3番目に大事なことは、基礎技術の開発である。コンピュータは、ハードウェアやソフトウェアというようにものももちろん大事であるが、さらにそれをバックアップする基礎技術を開発することが非常に重要である。たとえば、100万トンタンカを設計したいという場合、当然その構造計算を綿密にしなければならない。しかし、その構造計算をどういう方法で行なったらよいかというのは現在はっきりわかっているわけではない。近似的な方法や、やりくりする方法はあるが、それでは果たして実際に大丈夫かどうかということは、わからないわけである。そのために非常に大きいセーフティ・ファクタ(安全率)をかけてやることも行なわなければならない。そういうことを正確に行なわないと、いかにコンピュータが発達しても、うまく活用していくことがむずかしいのである。これは一般の情報処理の場合にもいえる。データ・ベースというようにものをどういう構造につくるべきか、あるいはその検索はどういう方法でやるべきかというようなことは、特にMISと呼ばれるような大きな情報システムをつくる場合には、非常に大きな問題になってくる。したがって基礎技術や基礎理論といったようなものを無視しては、決してうまくいかないということである。

最後に大事なことは、ユーザーは開発することを怠ってはいけないということである。つまり、真に企業のための機械としてコンピュータを使っていく場合、非常に大きな意味をもっているということである。

#### 4 コンピュータとその周辺技術の発達

コンピュータが他の機械と違う大きな特徴は、ソフトウェアが必要であるということである。もう1つ大きな特徴は、ENIAC以降のコンピュータの発達は非常にめざましいということである。つまり、同一種類のものでこれほどまでに著しい発達をした機械はほかにないだろう。たとえば自動車にしても、フォードが一番先に走らせた自動車と現在の自動車を比べても、平均的な値でいえば、速さは10対1にもなっていない。しかしコンピュータの場合、人間とENIACの計算能力の差よりも、ENIACと現在のコンピュータとの計算能力の差の方がはるかに大きいわけである。コンピュータの発展過程を数字的に見ると(1)~(3)表のようになる。(次頁)。

まず一番著しい特徴として処理能力について見ると、処理能力は2つあって、1つは速さ、1つはものを考える能力である。演算速度は、1950年代の一番速いもので大体 $1.000\mu\text{S}$ 、つまり1mSかかった。これは10桁ぐらいの数をプラス・マイナスする速さが大体1mSかかったということである。これは、10年後に大体50倍ぐらいの速さで速くなってきている。現在では、大体 $0.4\mu\text{S}$ ぐらいのものが、大型の計算機ではごく普通に使われている。 $0.4\mu\text{S}$ というと、1秒間に250万回のプラス・マイナスができるという速さである。1980年には、演算速度が $0.01\mu\text{S}$ のものが開発されているに違いないと思う。これが1つの大きな特徴である。

2番目に頭の良さであるが、大体10年間に1桁ずつぐらいの割合で良くなってきているということである。一番最初のプログラム容量は $10^3$ ぐらい、つまり1,000ぐらいである。これは、わりあいに複雑な給与計算を1つのプログラムで書けるぐらいの容量である。給与計算を1つのまとまったプログラムとして処理し得るぐらいの容量が、1950年代の最高限度だったわけである。これが大体10年ごとに1桁ぐらいずつ上がってきて、現在では、数十万といったプログラム・ステップが実際に実行できるようなコンピュータがある。近い将来には、数百万というようなことになってくると考えられる。

つぎに、補助記憶容量、つまりどれぐらいのファイルを記憶できるかということである。これは、大体10年間に2桁、100倍ずつぐらいの割合で伸びて、現在では10年間数百億字になっている。

## (1) 処理能力の向上

	1950	1960	1970	
演算速度	1,000 $\mu$ S	20 $\mu$ S	0.4 $\mu$ S	0.01 $\mu$ S
プログラム容量	10 <sup>3</sup>	10 <sup>4</sup>	10 <sup>5</sup>	10 <sup>6</sup>
補助記憶容量	10 <sup>5</sup>	10 <sup>7</sup>	10 <sup>9</sup>	10 <sup>11</sup>
線型計算の処理能力	100	1,000	10,000	100,000

## (2) 機能の多様化

	1950	1960	1970	
入出力情報	計数的情報	物理的情報	図形的情報	パターン情報
処理方式	一括処理	オンライン処理	会話型処理	時分割処理
プログラム言語	機械的言語	汎用コンパイラ言語		問題向言語

## (3) 信頼性、経済性の向上

	1950	1960	1970	
回路素子	真空管回路	トランジスタ回路	集積回路	高密度集積回路
M T B F	200時間	1,000時間	5,000時間	10,000時間
素子の価格	10ドル	1ドル	0.1ドル	0.01ドル
消費電力	100	10	5	1

以上の3つの性能はハードウェアの能力であるが、能力の向上ではソフトウェアの向上もみのがせない。たとえば、連立一次方程式を解く場合、1950年代くらいには百元くらいのものを解く計算技術しかなかったのである。その誤差をキャンセルするようによく計算していかないと、この計算はできない。コンピュータの性能ではなくて、誤差をうまくとって行く計算技術、つまりソフ

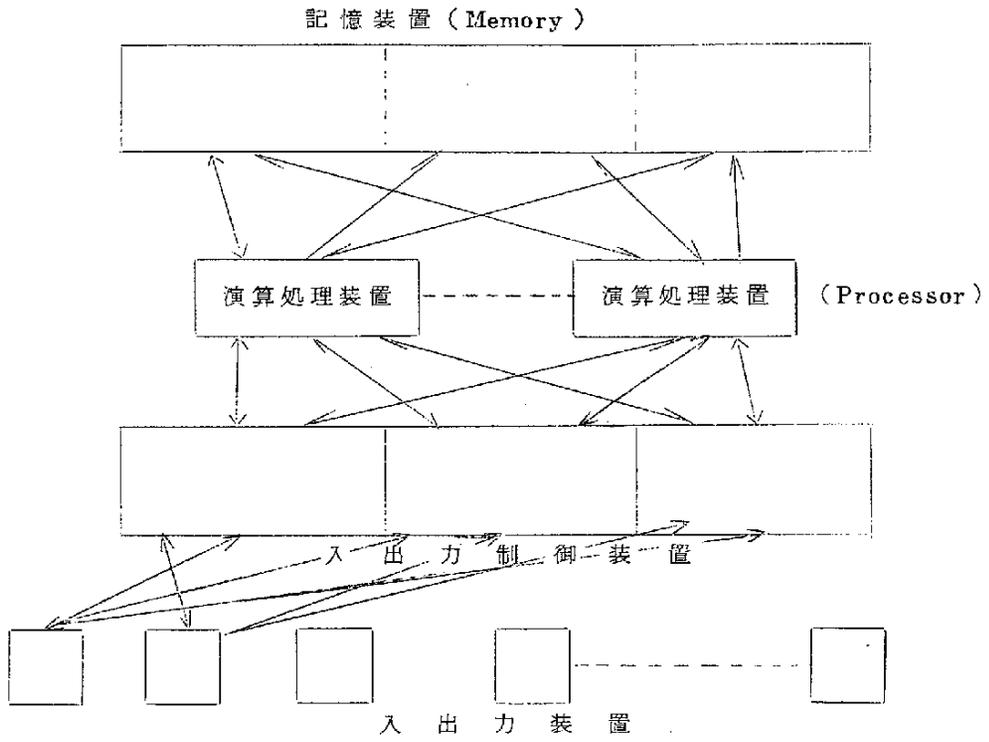
トウェアの方が問題になるわけである。50年代には大体百円くらいのものをさばくソフトウェアが限度であったが、それが60年代になると千円くらいのものができるようになり、現在では一万元位の線型計算が取り扱えるようになってきている。

さて、先にふれたコンピュータの計算スピードであるが、この上昇のスピードは今後にぶるだろう。なぜなら、現在のスピードそのものが、限界に近づいていると考えてもいいからである。では、コンピュータの処理能力はもうこれ以上あがらないかという、決してそうではない。コンピュータの処理能力はまだまだ向上する。それが個々のスピードは同じでも、一時にたくさんの仕事ができるようにと設計の方向が向いているからである。丁度、自動車のスピードをあげるよりも、積載量を多くする、また、道路を広くして一時に数台の自動車を走らす、そうした全運送量を大きくしようとするのと同じ考え方である。一時にいろいろなプログラムが互いに関連しながら仕事ができれば、コンピュータの処理能力は増大するのである。この方向に向って、これから登場し、常識になろうとしているのが、多重プロセサ・システムである。すでに数年前から、米国のウェスティングハウスのソロモン・コンピュータ、イリノイ大学のイリアック等実験コンピュータのいくつかがあり、最近では、この思想を実現した一般用コンピュータも続々と登場している。

これは要するに、図I-14の型をもつもので、従来の唯一の演算・制御装置にかわり、これ(Processor)をたくさんもったものである。この種のコンピュータでは、互いに関連のないプログラムを同時にいくつか走らすことができるのはいうまでもないことであるが、1つのプログラムをいくつかのモジュールに分解し、互いに関連をもたせながら(データにより、あるいは時間により)仕事を行なわせることが可能になっている。共通にしなければならない部分(記憶装置、入出力装置)は共通にし、個々別々に持たせた方が効果的な部分(演算、制御装置)を別々に持たせて、しかも、1つのコンピュータとして制御できるようにまとめたものが、多重プロセサ・システムなのである。

このような形でデータ処理を行なう技術が、だんだんと一般の汎用コンピュータでも使われている。2番目にめざましいことは、機能の多様化ということである。つまり、いろいろなことができるようになってきたということである。

圖 I - 14



1950年頃のコンピュータは、計数型の情報だけを処理するように考えられてきたわけである。もちろん、現在でもコンピュータは基本的には計数型の情報しか処理しないが、60年代になると、それに加えて物理的な情報がコンピュータで扱われるようになった。物理的な量、たとえば電流の量だとか電圧だとかいったものを、直接コンピュータの入力情報として使うことができるようになったということである。たとえば、化学工場で無人制御などを行なうためには絶対に必要であり、電力プラントの制御をするためには、いちいちそれを計数型の情報に直してそれをコンピュータにあらためて入れるという方法では間に合わない。また、現在では図形的な情報が取り扱えるようになった。映像表示装置で、グラフを表示するというような装置である。映像表示装置に瞬間的に図形をかくことができるようになったが、さらに図形そのものを、コンピュータにインプットとして使うことができるようになった。一般にタブレットと呼ばれている入力機器で、一番最初にランドコーポレーションで開発したものであるが、図形そのものをコンピュータの入力情報として使った。

今後の問題は、パターン情報と称するものである。パターンで与えられている情報をうまく処理しようということ、現在でも部分的には行なわれるようになってきている。たとえば、手書きOCRというようなものがある、手で書いた数字をうまくコンピュータで読み取るということである。あるいは、郵便番号の認識機械なども一種のパターン認識だといえる。パターンというと、図形と同じものだと思われるが、実は大きな違いがある。図形は形そのもののものをあらわしているが、パターンはそうではない。たとえば、ここに「あ」という字を書く。これが大きくても小さくても、情報としては全部同じ情報である。つまりこれは、線のつながりによって1つの情報があらわされているわけで、形ではない。どこがどう連絡しているかというようなことで、1つの情報が与えられている。このような前後の情報がコンピュータによって与えられている情報のことを、一般にパターン情報と称する。音声も代表的なパターン情報である。これらをどのようにうまくコンピュータで処理するかということが、これからの重要な課題である。パターン認識は、現在のところ、まだそれほど実用段階になるところまではいっていないが、それがだんだんと実用化されてきているということである。

次に、処理方式では、昔はバッチ処理だけであったが、60年代になってから、通信回線とコンピュータを直結して、遠隔地からデータを直接コンピュータに送り込むことができるようになってきた。さらに現在では、一般に会話型処理、つまり、リアルタイム処理が行なわれるようになってきた。将来の姿としては、時分割処理(タイム・シェアリング処理)、TSSということがあげられている。これは、1つの大きなコンピュータにいろいろな端末機器を数多く結び、情報だけでなく処理するプログラムまで送ってしまうものである。独立に各端末機から勝手に、処理したいと思う仕事のプログラムを中央に送る。そこで処理させて、また結果を送ってもらう。すなわち、1台の端末機器を持っているだけで、あたかも大きいコンピュータを持っているのと同じように仕事ができるわけである。業務量としてはコンピュータの1割ぐらいしか使えないけれど、仕事の質で大型のコンピュータでなければできない仕事もすることができ、非常に都合がよくなる。異なったメーカ、異なった企業が、1つの大きなコンピュータを共同で使えるということであり、高性能のコンピュータでなければできないものが使えるようになるということは、これから1つの大きな方向であろうと思われる。

プログラム言語は、一番最初機械向き言語しかなかったということである。プログラム言語といっても、機械の言葉そのものか、あるいは機械向きにつくられた言葉、すなわちアセンブラ言語と呼ばれる低いレベルの言葉だった。60年代になって、一般に汎用コンパイラ言語になった。プログラム言語をコンピュータの言語に落とす言語処理プログラムのことを、汎用コンパイラ言語と一般に呼んでいる。

COBOL、FORTRAN、ALGOL、PL/1といったものがそれである。これらは、一般に問題向き言語(Problem Oriented Language-POL)というふうに呼んでいるが、厳密にいうと、問題向きではなくてあくまで汎用なのである。つまり、どの問題に向くということではなくて、人間が使いやすいようにできていることばである。真の問題向き言語は、たとえば、データ・ベースの完了するために非常に都合のよいプログラム言語とか、あるいはシミュレーションをやるためのプログラム言語(たとえばSIMSCRIPT II)というように、もっと狭い範囲の問題を処理するために都合のよいプログラム言語

がある。そのようなプログラムの開発が、これからだんだんと活発になるであろう。

次に、ハイブリッド・システムの実用化である。アナログ的情報、つまり物理的な量であらわされているような情報と計数的な情報が混在するような1つのシステムを、ハイブリッド・システムと呼ぶ。今までのところは特別な制御の分野でしか使われていなかったが、たとえば、数値解析の難しいものを取り扱う場合には、こういうハイブリッド・システムで解くことはかなり有用である。このハイブリッド・システムは、これから実用化されていくであろう。

適応制御は、状況に応じてそれに合うような形でコンピュータが対応して、自分でプログラムをつくり出していってくれるというようなシステムである。現在考えられているのはそれのごく初歩で、コンピュータにものを学ばせる学習系ランゲージ・システムの開発が実験的に行なわれている。

もう1つの重要なことは、信頼性、経済性の向上である。回路素子は、1950年代には真空管回路が使われて、60年代になってから、一般に固体回路化されたトランジスタ回路、あるいはトランジスタ・ダイオード回路が使われるようになった。現在では集積回路、いわゆるICが一般に使われている。ICはトランジスタ回路に換算すると数十個分にあたる。将来の方向としては、トランジスタ回路に直すと数千個分にあたるLSI（高密度集積回路）が考えられている。

回路素子の発達により得られる大きなメリットは、信頼性が向上したということと、もう1つ、コンピュータそのものを安くつくることができるようになったということである。

信頼性については、普通MTBFというような指数を使う。MTBFというのはMean Time Between Failureという言葉の略で、平均故障間隔あるいは平均故障率というように呼び方が一般に行なわれている。つまりこれは、一度故障が起こってから次に故障が起こるまでの無事故運転の平均の長さである。50年代の真空管時代には、コンピュータは連続運転200時間が限度であったが、60年代に入ると1,000時間位のものがあらわれ、現在では数千時間というようなものもでてくる。将来これは1万時間になるであろうということである。故障の間隔が長くなることによる問題点は、故障を直すのが非常

に難かしくなるということである。MTBFの非常に長いコンピュータになると、いかにしてメンテナンス・エンジニアを訓練するかというのが1つの大きな問題になってきている。

もう1つの問題は、MTTR (Mean Time To Recover)、つまり平均的修復時間である。つまり、故障してそれを直すまでの平均の時間である。この時間は短くないと困る。故障率と一般に呼ばれているものをあげると、

$$\text{故障率} = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}}$$

という式で関連づけられる。

次に素子の価格であるが、1個あたり10ドルぐらいしたのが大体10年ごとに1けたぐらいずつ下がってきて、現在では、論理素子1個あたり10セントぐらいでつくられるようになってきた。将来はさらに1けたぐらい下がるであろうと考えられている。コンピュータの消費電力も、50年代を100とすると、現在は10とか5とか1とかの指数になってきている。

その他、現在この分野における最も重要な問題点は、ソフトウェアの開発コストの問題である。ハードウェアの開発コストはどんどん下がっているが、逆にソフトウェアのコストは上がっている。50年代には、ソフトウェアの開発コストは、コンピュータ開発コストのほんの数%にすぎなかった。現在ではすでに50%を越え、将来は、おそらくコンピュータの開発コストの70~80%がこのソフトウェアで占められると思われる。ソフトウェアの製造工程を見ると、家内工業式が大きくなったというような製造の仕方しか行なわれていない。この辺は1つの大きな問題だろうと思われる。ハードウェアとソフトウェアの標準化は、コンピュータのコストを下げるのに非常に重要なことである。特に、これから先コンピュータがオンライン使用された場合、回線と接続して使っていくことが非常に大事になってくる。コンピュータが変われば端末も変わるということでは困るわけである。通信回線なり、端末機器に対するインターフェースなど、標準化してできるだけユーザー・サイドでうまくコンピュータが使えるようになっていないと困るわけである。

以上あげたような分野で、コンピュータ・メーカーおよび関連企業は、日夜努力を続けている。

## Ⅱ データ・プロセッシング

### 1. 情報処理の目的と方法

情報処理の目的は生産性向上、経営情報のレベルアップ、新技術の促進など、いろいろなことを目的としている。

その中で、コンピュータだからこそこできるという特色は、高速性、記憶が大容量、すなわち機械的に処理ができる形で多くの情報がストアされ、いろいろなものの相互関係をつかまえることができるということである。

たとえば、旅館の予約、列車の予約といったものを考えてみると、コンピュータがなければ列車はあいているかどうか、電話をかけて調べなければならない。旅館にも電話をかけなければならない。電話がなければ手紙で尋ねる。それであいてなければ、旅行計画を少し変更してもう一度くりかえす。それがコンピュータの中にいろいろなデータがストアされ、ハイスピードで相互比較ができれば、1つのことがぐあい悪ければ、次の方法、それでも悪ければまた別の方法をとるというふうにしらべていくことができる。こういうのがコンピュータの特色を生かした方法で、この形での情報処理の向上がはかられている。

もう1つの形は、いろいろなI/Oの装置を利用して行なわれる教育用のシステム、CAI（コンピュータ、エーデット、インストラクション）で、飛行機の操縦のシミュレータとか自動車の練習、そういった人間との対話を中心とした情報処理方式がある。

#### 1-1 情報の流れ

また別の方法でデータ、プロセス（情報処理）をみると、一般には、生産工場や倉庫、あるいはサービスなどの窓口、つまり実際の作業をしている所との関係でおこなわれているものがある。その処理の過程の1つの形は、作業結果についての報告書が各部所から出され、それがだんだんと集約されて、何らかの形にまとめられる。つまり各部所から出された情報を集約していき、最終的に社長への報告書になる。この過程は、こまかな情報は社長には必要ないので、社長が心要とする情報に集約して持っていく。こういう作業は結果の報告である。こういうのを後始末型の処理といっている。

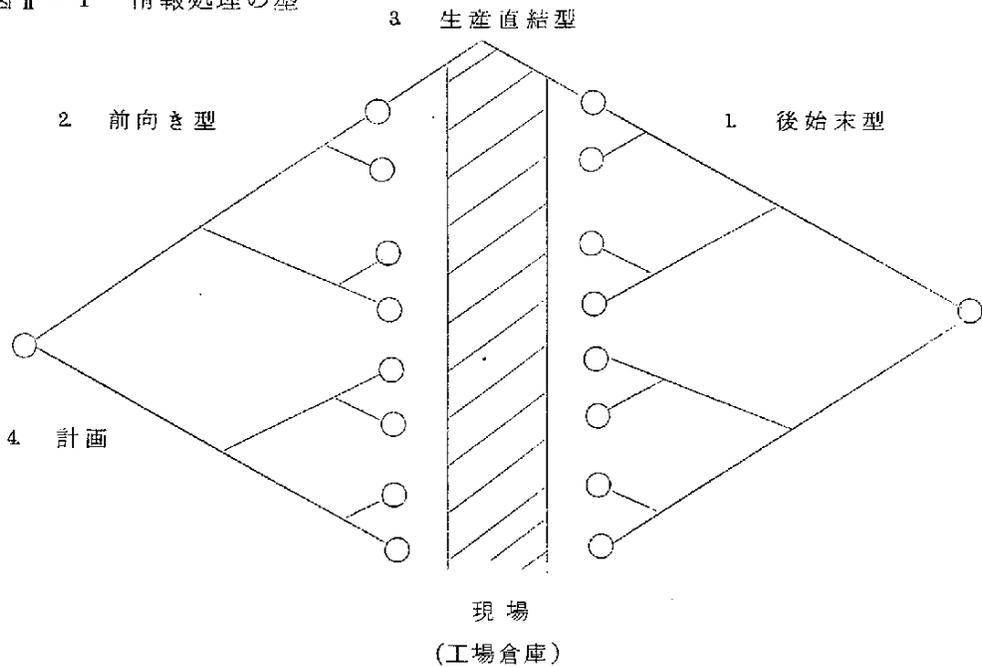
それからもう1つは、働いている人たちにどういう仕事をしなさいという命令が当然あるわけで、自動車工場で、きょうの仕事はこれこれだということ、1人1人に知らせる作業命令書のようなものがある。それはどういうふうに個人個人の手元に届いているかという、社長がまず何かの計画を立てる。今日の自動車の生産は10万台で、そのうち乗用車が何台、トラックが何台、バスが何台というような計画を立てる。すると社長は各課長が何をすべきかという命令を出し、各課長は部下の個人個人が何をすればよいかということ、つまり情報がだんだん拡大されていく形がある。これを前向きの処理という。このように情報を集約するのと情報を展開するのと2つの基本型がある。

その他に、生産やサービスのために直接コンピュータを使い工場の自動化（プロセス制御）や、無人倉庫のためにコンピュータを使い型すなわち生産直結型がある。情報処理の型は、図Ⅱ-1のようになる。

1950年ごろまでの処理方式は、後始末型の情報処理であった。次に生産工場で問題になってきたのが情報を展開していく方法である。さらに無人工場あるいは無人倉庫など省力化が大きなテーマになってきたのが1970年代であり、これからの情報処理の大きな目的になってくるのではないかと考えられる。

ここで問題になるのは、コンピュータがなくて人間がやっている場合には、同じルートを通じて情報が展開されまた集約されてかえってくる場合である。たとえば、課長が命令を下して部下からレポートをもらう。それをまた部長に持っていく。それがまた社長の所にまとまってくる。この場合、課長や部長のところでは自分の仕事の範囲内の情報は、すぐその場で手に入る。何か変なことがあった場合、自分のところで手が打てる範囲内の問題はすぐに手を打つ。ところが、コンピュータを使用して2つの別の情報のルートをつくると、変なことがあった場合、最後の結果をまたなければならぬ。そこでわかったとしてもどこがおかしいかは、調べ直さなければならぬ。そこで各部分で必要な情報を適時にフィードバックする必要がある。システムを設計するとき、そのフィードバックの方法を組みこんでおくことが必要であり、部、課長の管理職は、そのようなフィードバックを促進する方法を身

図 II - 1 情報処理の型



につけなければならない。今まではすぐその場で情報がつかめたのに、コンピュータを使うようになった時さっぱり情報が流れなくなったということがあるとするれば、やはりその辺の必要な情報の流し方のシステム設計のミスとか、使う立場の人々の理解の不足からということが考えられる。

また前向きと後始末型を一体化してシステムをデザインし、何かトラブルが起った場合、— トラブルというのはプログラム・エラーもあるし、ハードウェアのエラーもある — 全体の進行がとまるおそれがある。たとえば、設計する仕事の部分がおかしいときには、前向きの情報は現場へストレートに流しておいて、統計処理はまたあとで回復することを考える方が普通なので、システムをつくるときには、お互いに関係がなければいけないが、それが干渉し合ってもいけないという、非常に奇妙なものをつくり上げていかなければならない。

### 1 - 2 生産性の向上

生産直結型を見ると、まず第一次産業革命のときには大量生産ということでもコストダウンがはかられ、性能とか規格が統一されたよいものができた。

それに対して最近、ユーザーのいろいろな希望、趣味を取り入れなければならない。そうすると多品種の少量生産ということが要求されることとなる。自動車であれば、フォードが最初つくったのはT型フォード1種類だけであった。それが、最近ではデラックス、スタンダード、その他にオプションと、いろいろなものをくっつけて、しかもそれが大量生産と同じぐらいの速さと安さでできるようになった。そういう要望に対して無人工場とか無人倉庫などにおけるNC（工作機械の数値制御）技法などが非常に大きな影響を与え初めた。

これはどういうことかと言うと、いろいろなものをつくるということは倉庫に各種の部品を持っていて、要求に応じてそれを組み立てる。ところが部品の在庫を持つということと、数値制御の工作機械でプログラムの紙テープを持つことは同じことになる。ある種の部品がほしいと言われたときに、倉庫から部品をつくるプログラムのある紙テープを持ってきて工作機械にかけるのとの違いだけである。要するに部品在庫を紙テープで置きかえることができる。

もっとおもしろいことは、古い補修部品をとっておくのはたいへんである。ところが、これが紙テープになっていれば、いつまでもとっておける。そんな意味で、現場を自動化するといういき方が、生産性向上そのものの考え方、倉庫とか部品というものの考え方に非常に大きな変化を与えてきている。そういった総合的な会社システムの中におけるコンピュータの役割を理解しなければならない。

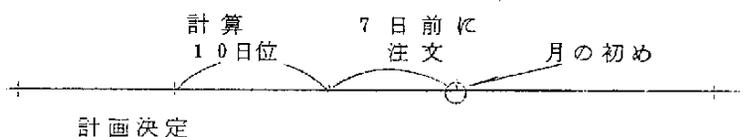
コンピュータは、プログラムがなければ動かない。そのプログラムがあることによって後始末、前始末生産、そしてその生産を通じて部品というものに対して大きなイメージ・チェンジが行なわれつつあるのである。もちろん、多品種を間違いなく管理することもやはりコンピュータの大容量性と高速性、および人間以上の正確性、これを活用しているのである。

### －3 経営情報のレベルアップ

コンピュータの大容量記憶と高速性を利用して、各種の分析作業をしなければならないが、その場合、高速性がどういうふうに生かされるのか？各種の分析をするだけなら時間をかければ手でもできる。コンピュータを使っ

て、人手なら、1000時間かかるものが1時間でできるというような高速性は、どういふふうに生かされるだろうか。

営業情報などがコンピュータを使うことによって、1週間早く社長の手元に届けられることは、確かにコンピュータの効能であるが、それは一体金に換算できるだろうか。社長が机の引き出しに1週間入れっぱなししておいたら同じことになる。そうすると、コンピュータの高速性を生かすためには、社長つまりレポートをもらった人が、1週間早くもらったことによって、それだけ利益を生み出す人でなければ無意味となる。つまり、時間を金に換算する能力がなければならない。また、いろいろな見方の分析をすることは、その結果をフィードバックできる能力が要求されることとなる。コンピュータそのものがどのように活用されるかというのは、逆にコンピュータを使う立場にある人がどう活用できるか、その人の能力によって非常に左右される。そういう点がコンピュータを情報処理に活用するときのポイントとなる。ある月の生産計画をいつ立てたらよいだろうか。部品を注文する時間があるから、



7日ぐらい前に注文しないと間にあわない。計画を立てて、どれだけの部品が必要だということ計算するのに10日ぐらいかかる。そうすると、計画決定は17日前となる。この日数が正しいかどうかは別として、少なくとも幾日か前に計画を決定しなければならない。そこでコンピュータを入れて、この計算を1日で済ます。そうすると決定を8日前にまで縮めることができる。コンピュータ費用が同じだとすると、表面の利益は何もないが、実はこの9日の間に世の中の大勢がどう変わるか、何か変化があったとすると、この間にすぐ計画変更ができる。前の方法だと17日前に決定しているので、その次の月のところで変更しなければならない。この10日間と9日間が生きて使えることとなり、やはりこれも時間を金に換算していると言える。

それからもう1つの型がある。たとえば社長はまず計画を立てる。計画を立てれば、自分の立てた計画が良いか、悪いかということを実行前に知りたい。つまり、計画を検討するという型の使い方である。もう少し詳しくいう

と、工場の作業がコンピュータの中に完全に再現されていたとする。社長は、今月1万5000台の乗用車、2万5000台の貨物自動車、500台のキャンピングカーを生産しようという決心をする。それで、工場に命令を下す前に、こういったものをコンピュータの中で再現してみる。順調に流れればそれでよいが、いざ計算してみると、倉庫にある部品が足りなくなる場合もある。そんな現象もコンピュータの中でやってみればわかる。

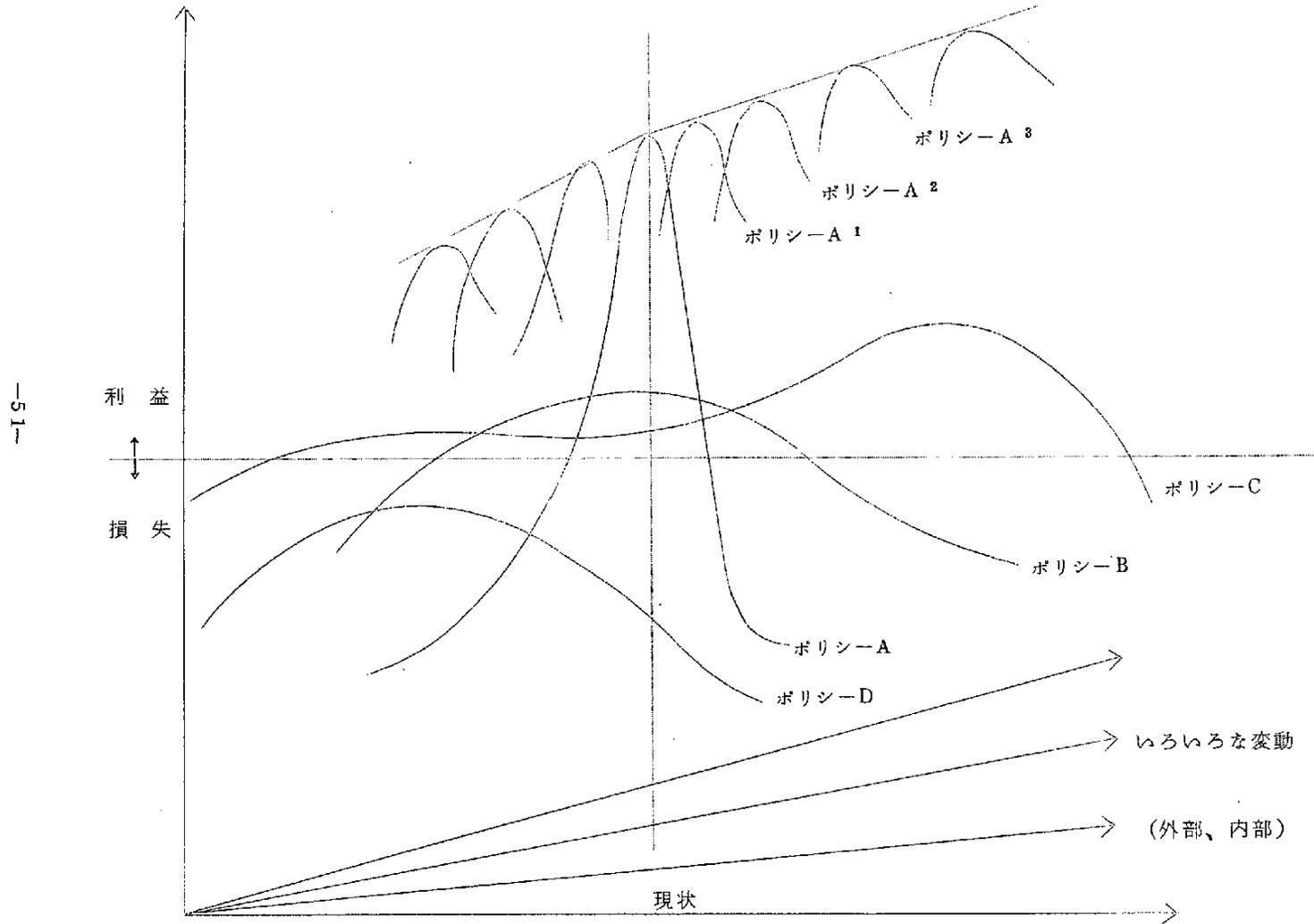
そこでコンピュータの使い方として、工場のシミュレーションをして、計画の実現可能性を検討することができる。外部情報、乗用車の販売予測、景気の変動などからして、2000台削ってもよいという判断や、2000台削ると工場がアイドルになり、次の月が忙しくなるので、ストックにしておけばよいという判断ができる。こういったシミュレーションを活用して、コンピュータは計画に使われていかなければならない。

このシミュレーションには2通りある。

それは図Ⅱ-2のごとく、たとえば、会社の運用ポリシーをAとすると、世の中の情勢に応じたポリシーでやると利益のカーブが求められる。このポリシーをきめるのは別にコンピュータを使う必要はなく、勘できめることもできる。コンピュータを使うときのメリットはどこにあるかという、幾つかのポリシーを短時間のうちに比較、検討することができる点にある。中には赤字がでるものもあるので、そういうポリシーは採用しないことになる。さて、図Ⅱ-2のようにいろいろなカーブができたときに一体どうするかという問題がある。それは世の中の変動の幅をどう見るかの問題で、大きく変動しないという見込みが立てば、社長はAのポリシーを採用するであろう。しかし、変動が起こり得るし、その幅が相当に広いとなると、ポリシーBを採用することとなろう。またもしもっと変動が大きいと考えれば、Cの方が安全だということになる。ある与えられた条件における最適なオペレーションをやるのが目的ではなくて、いろいろな変動があっても、利益が確保できるやり方でオペレーションをすることになる。これをさがすのもシミュレーションの利用である。

また、社会情勢が変わった場合には、すぐポリシーを切りかえる必要がある。その場合、図Ⅱ-2における最高のところをずっとつないでいくことがで

図 II-2 シミュレーションの例



きれば、(ポリシー  $A \rightarrow A^1 \rightarrow A^2 \rightarrow A^3$  というように)、変化にうまく追従していくことになる。こういったことをする場合にも、コンピュータの高速性が活用できる。自分のポリシーがどのように利益に結びつくかを考えてコンピュータで前もって計算しておいて、その結果を長期にわたって使っていけばいいというような会社もあるだろうし、あるいは毎週毎週、毎日毎日、いろいろな情報を利用しながらやっていった方がいいという会社もあるだろう。これは実はコンピュータに使う費用の問題もからむわけである。うまくポリシーを変える方式をやろうと思えば、相当大きなコンピュータを持って複雑なシステムデザインをしなければならない。

#### 4 新技術の促進

技術計算のほりに目を向けてみると、設計という問題や、CAD (Computer Aided Design) という問題がある。コンピュータを設計に使う場合の特色は、開発のスピード・アップができることである。エンジニアは一步步着実に安全に設計計算をしていかなければならないが、そこに大きな飛躍があると理論と実際に食い違いが出てくる。それがコンピュータを使って解析の精度を上げていけば、安全に飛躍できる範囲が広がる。タンカーなどでみると3万トンのタンカーができるとすぐに、それを10万トンに引き上げ、さらに30万トンへと巨大化のステップを早めたのもコンピュータの力であり、また36階の高層ビルを作ったのもコンピュータの能力によるものである。

さらに、エンジニアのもつ情報の蓄積をコンピュータに処理させることにより、より広く、より多くの情報を設計にとり入れることができるようになった。

こういった考え方を一層拡大したものがCADで、そこでは1種のシミュレーションが行なわれる。物をつくらないうちで、コンピュータの中で架空の物をつくり、その形状をしらべ、その性能を計算し、それに人間-機械の相互会話機能を入れて、新しいよりよい物をつくり出していくことができる。エンジニアのための情報処理システムは、非常に重要な意味を持つようになり、積極的な意味で利益をだしてくれるシステムとなるのである。

以上述べた情報処理の目的を整理してみると次の表Ⅱ-1のようになる。

表 II - 1 情報処理の目的

生産性向上	経営情報のレベル・アップ	新技術の促進
<ul style="list-style-type: none"> <li>○ 省力化</li> <li>○ 迅速化</li> <li>○ 正確化</li> <li>○ 適性化</li> </ul>	<ul style="list-style-type: none"> <li>○ データ・バンク</li> <li>○ マネージメント・サイエンス 管理手法 予測手法 統計、分析手法</li> </ul>	<ul style="list-style-type: none"> <li>○ 新製品の開発 要求の分析、マーケット リサーチ CAD シミュ レーション (適用性の確認)</li> <li>○ 新技術の開発 テスト・プラントのスケー ルの向上 (運搬、建設)</li> <li>○ 新種サービスの促進 予約 (フィードバックの速さ) I R</li> </ul>

2. 処理手法

ここでコンピュータの仕事というのはい体どういう形で普通行われているかということを考えてみよう。

典型的な例 (図 II - 3) はまず元帳 (マスタ・ファイル) がありそして、トランザクション・ファイルがある場合である。

このトランザクション・ファイルの内容に従ってマスタ・ファイル (ニューマスタ・ファイル) を書き直す。これがコンピュータの作業の基本型である。

それからもう1つの作業 (図 II - 4) は、この結果できたマスタ・ファイルなし、トランザクション・ファイルから何か情報を出す、つまり報告書をつくる作業である。コンピュータの作業は基本的にはこの2つである。たとえば、ある期の貸借対照表があり、それに対する営業活動があって、翌期の貸借対照表が出る。それがまた次の期へ行くというような繰り返しである。すべての会社の情報が伝票になっていたとすると、その伝票をマスタ・ファイル、あるいはトランザクション・ファイルとして次の処理をするということは、会社の中の動きをコンピュータの内部で再現しているということである。

レポートというものは、そういった時間的な動きがあるものの中か

図 II - 3

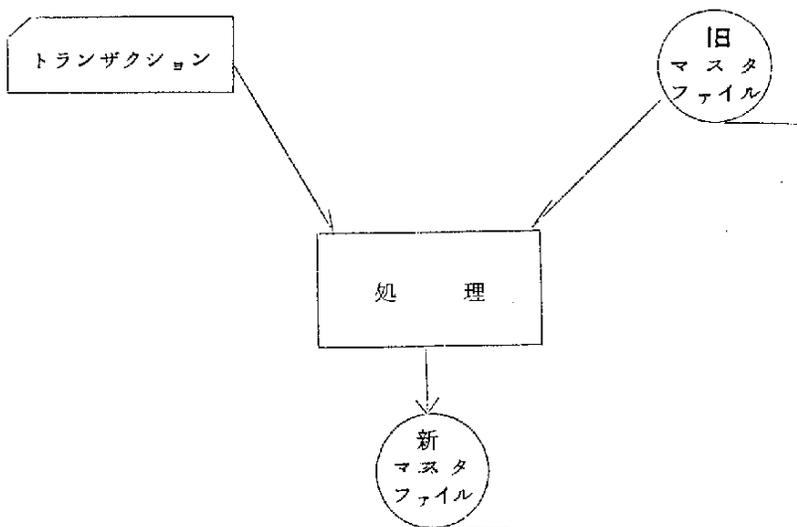
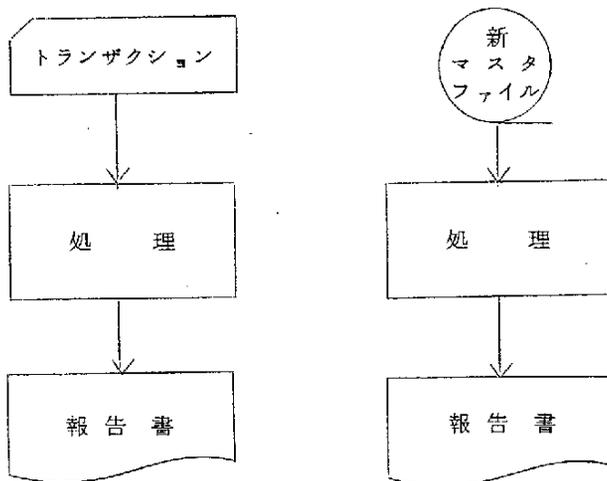


図 II - 4



ら必要な情報を取り出して報告するということである。

この基本作業をするのに1番大事な操作は何かというと、分類 (SORT) という操作である。分類という操作は、データをキーコード順に順序よく並べておくということである。たとえば、給与計算用のマスタ・ファイルを従業員番号順に並べておくことなどがそうである。次に給料計算のトランザクション・カードが発生した場合、これも従業員番号と同じ順序に並べておくと作業スピードが非常に上がってくる。従って、分類つまり順序よく並べるというのが次のべるバッチ・プロセス方式で、情報処理を行なう場合の非常に重要な作業となる。それに反し、リアル・タイムまたはオンラインという処理を行なうのであるから、分類というのは報告書をつくる時にわかりやすいまとめ方、または分析の立場々々に応じて、データの集計を行う手がかりとしての意味と、特定のデータをすばやくとりだして加工してしまうときの手がかりとしての意味をもってくる。

このように分類 (コードに応じた集計) というのは、情報処理の基本の処理であると同時に、統計分析のための重要な手がかりとなるものであるので、充分将来の新しい分析に応じられるものであると共に、現時点でのシステムの流れに応じたものを設計しなければならない。それゆえに、コード設計の良し悪しによって、そのシステムが成功するかしないかがほとんど決まってしまうと言われている。

## 2-1 バッチ処理

バッチ処理というのは、1ヶ月とか1週間とかのデータをためておいて、一度に処理する方法である。これを在庫管理を例にとって考えてみよう。まず、出庫カード、入庫カードをつくり、前の残高を計算して新しい残高を作る。これが1月ごと、あるいは1週間ごとというように定期的に行なわれる。いま、その途中で現在在庫がいくらあるかという質問を受けた時、倉庫へ行って品物を見て知らせるか、倉庫係が在庫帳に常時記録しておいて、いくつ残っているかを調べるのである。この様に、バッチ処理ではある期間ごとに処理されるため、必要な情報を必要な時に取り出すことが出来ないという欠点がある。その点、リアルタイム処理の場合は必要な時に必要な処理がすぐ出来る。ところがコンピュータの実働効率を考えると、バッチ処理の場

合の方が非常に効率がよい。というのは、リアルタイム処理の場合、データの入り方がまちまちであるが、バッチ処理の場合データをまとめて処理できるので、遊びの時間が非常に少なくなるからである。

バッチ処理とリアルタイム処理は、時間的、距離的に見て非常に処理の形態が異なるので、それぞれの特性を考えて使わなければならない。たとえば、給与計算のような場合には、給与が支給されるのは月に1度であるので、修正、任意控除や勤怠などのデータは、月に1度まとめて処理されればよい。これをリアルタイムで処理するメリットは全くないのである。ところが在庫管理の場合を考えると、特に倉庫がたくさんあり、しかもそれぞれ離れている場合には、常に正確に在庫をとらえておくことは非常に困難になる。

ところが、在庫管理を月1度のバッチ処理で行なったとすると、その1ヶ月の間にある品物が急に売れた場合、品切れをおこす危険性がある。だからといって多量の商品をもった場合、在庫費用がかかり、しかも急に売れなくなった場合には、多量の在庫をかかえこむことになる。この様なことをふせぐには、きめの細かい在庫管理を行なわなければならない。つまり、出庫や入庫の状態を常に正確に知っておいて、いつも適性在庫維持をしなければならない。この様な場合、リアルタイム処理が非常に有効になってくるのである。

また、旅館の予約の場合でも、バッチ処理を行なったのでは、空室を正確につかむことは出来ない。この様に、情報処理の目的に合わせて処理の形態を考えなければならない。

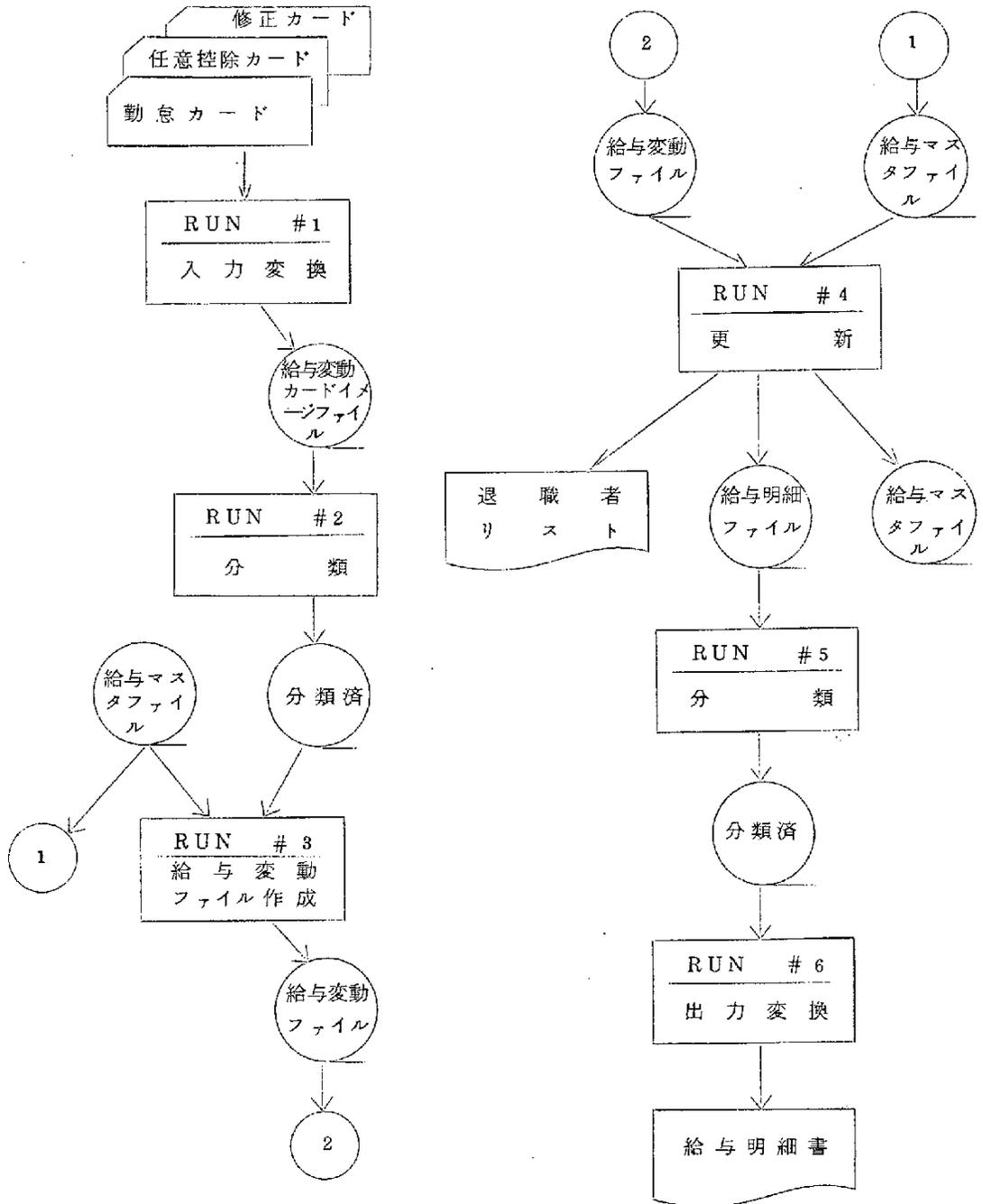
ここで、バッチ処理の典型的例として、給与計算のシステム・フローチャートを上げよう。この処理は、1ヶ月間にまとめた修正カード、任意控除カード、勤怠カードから給与変動ファイルというトランザクション・ファイルを作成し、それによって給与マスタ・ファイルを更新しながら、給与明細ファイルを作成し、それから給与明細書を作る一連の処理である。(図II-5)

## 2-2 リアルタイム処理

リアル・タイムというのは、そのデータが発生したときに、そのつど処理をする方法である。これが最初に使われたのはアメリカで、対空防衛システムとして、レーダとコンピュータを直結したものであった。

このリアル・タイムの考え方を在庫管理を例にとりて考えてみよう。

図II-5 給与計算フローチャート



いま部品1つに1枚のカードを使う。そのカードを箱の中に順序よく入れておく。これは1対1で対応している。まず部品が欲しいという人は、その対応しているカードを持って倉庫へ行き、部品と引きかえに持って行く。ここで歩いている時間を無視すると、部品とカードは一致する。問い合わせがあると、カードを見て在庫がすぐわかる。新しいものが倉庫へ入ってきたら、カードを追加すればよいのである。これは倉庫のシミュレータである。倉庫と全く同じような作用をするものをカードで表わす。データ・プロセッシングという意味は、データをプロセスするというイメージよりも、いかにして現物と同じような作用をつくるかという思想でなければならない。

たとえば、同じ種類の部品カードがある場合、その10枚目のところに赤いカードを入れておく。カードを抜いているうちに赤いカードが出てくる。赤いカードは、実は部品に対応しているのではなくて、注文書のカードである。ある一定レベルまで在庫が下がってくると注文書が出てくる。すなわちこれは自動発注方式にもなる。もう少し在庫が下がっていくと、今度は電話催促カードが出てくる。よく考えてみると、ディス・バックを使ってオンライン・リアルタイム方式で倉庫管理をしているときに、やはり同じようなことをしているのである。リアルタイム・プロセスは、コンピュータと直結して工場をコントロールし、発電所をコントロールし、あるいはデータを記録しておくことなどに使われていたが、あまり効率が良くなかった。1年に何千回問い合わせがあっても、高性能のコンピュータにとっては全労働の何分の1でしか仕事をしていないということである。リアル・タイムの思想、メリットは非常に良かったが、コストと釣り合いがとれなかったのである。しかし、現在ではみどりの窓口は全国に何百となく散らばっていて、1台のコンピュータで処理するのに都合がよかったので、非常に効果をあげている。また、銀行の預金システムにも大きな効果をあげている。そして、これからはますます他方面に利用されていくであろう。

### 3 タイム・シェアリング・システム(TSS)

コンピュータの使用方式として、一つはバッチ処理の方向があった。また、それと独立して、オンライン的な使い方が発達してきた。

もう1つの展開は、小型、高速、安価な集積回路の採用により、コンピ

ータは一段と大型化し高速化した。ただし、これは中央処理装置の方であって、その他の入出力装置の方はそれほど変ってはいない。そこで、この差を埋めて有効に使う工夫はないものかということで、タイム・シェアリング・システム（TSS）という考え方が生まれたのである。

タイム・シェアリングというのは、ある仕事をしている時に優先順位の高い仕事が入ってきた場合、今やっている仕事を途中でやめて優先的にそちらの仕事をし、それが終ると途中でやめていた仕事をまた始める。つまり、複数の仕事を、それぞれの短い時間ずつに区切って断続的に処理する方式である。1台の中央処理装置に数十、数百の通信回線を直結し、同時に大勢の人がオンラインでコンピュータを使用する。しかも、各人に全く自分1人でコンピュータを使っているような感じをいだかせる方式、それがタイム・シェアリング方式である。

つまり、バッチ処理とリアル・タイム処理の場合には、使用する人の数だけコンピュータが必要であるという使い方であるが、TSSというのは、コンピュータは1台しかないが、そのターミナル（端末機器）を使っている人から見ると、それぞれコンピュータが1台ずつあるといったような感じなのである。

TSSの使い方として、バッチ処理とリアル・タイム処理をコンバインした形と、各ターミナルのユーザーが1台ずつ計算機を占有しているんだという形と、2通りの使い方がある。

TSSのメリットとしては、データが1台の端末に1つずつついているのではなくて、多くのターミナルから1つのデータを全員が使えるということでもある。たとえば、技術計算をする場合、いろいろな技術データは全部共通のところにおいてあって全員が使える。

また、コンピュータで自動的に診断する場合、診断の新しいデータとか、プログラムをストアしておく。医者はターミナルを使って診断する。ある医者が新しい方式を考えてこれを取り換えると、全部一斉に交換される。ところが、小型コンピュータなどをそれぞれが使っていた場合には、新しいプログラムやデータを各人が作成しなければならないということである。

次にデメリットであるが、TSSは、コンピュータの使用効率からすると

非常に悪い。ターミナルの切り換えに要する遊び時間が入るからである。その上待ち時間からいえば、問題はもっと悪くなる。極端な例でいうと、1時間かかるような計算が4つあったとする。すると答えが出るのは4時間あとである。もし、バッチ処理でやっていくと、1時間目に最初の答えが出る。2時間目に次の答えが出る。平均待ち時間は2.5時間である。TSSの場合、平均待ち時間は4時間である。TSSの考え方は非常に良い考え方であるが、実用的に使うには、そういった面を考えながらデザインしなければならない。実用的に使うにはTSSの特色を生かし、デメリットの少ない仕事に使うべきである。

そのような応用面は、

- a エンジニアの創造力を助ける。(会話型)
- b 経営者、管理者に適確な情報と予測を与える。(IR型)
- c 共通のファイルに遠方からアクセスできる。(データ・プログラムの共有)
- d 仕事量、スケールから専用機では不経済な場合。(リモート・バッチ)などという所にある。

## 2-4 情報の蓄積と検索

現代は情報過多の時代といわれ、われわれの回りでもいろいろな情報がはん乱している。そして、物質的生産、エネルギー生産に代って情報活動が大きな比重を占めるようになってきた。

そこで情報とは何かとすると、たとえば別荘地が売り出されたニュースがあったとする。ところが、このニュースは、いま別荘地を買いたい人、あるいは将来買おうとしている人にとっては価値があるが、全く買う予定のない人にとってはほとんど価値を持たない。また、株式投資をやっている人にとっては株式情報が大きな価値を持つが、興味のない人にとっては単なるニュースとしか受け取れない。

このように、情報の価値というのは、その情報を受け取る人の行動によって決定されるのである。

逆にいうと、情報は、人間や組織体が行動を決定する時、必ず重要な役割りを果しているということである。たとえば、買物をする場合の広告やチラ

シの役割り、あるいは企業活動における外部情報ないし内部情報などがそうである。

ところが、現代において膨大な量のデータがはん乱しており、そのために自分の欲している情報をつかむことが困難になってきている。つまり、膨大なデータのおかげで、かえって情報不足という現象が起きているのである。

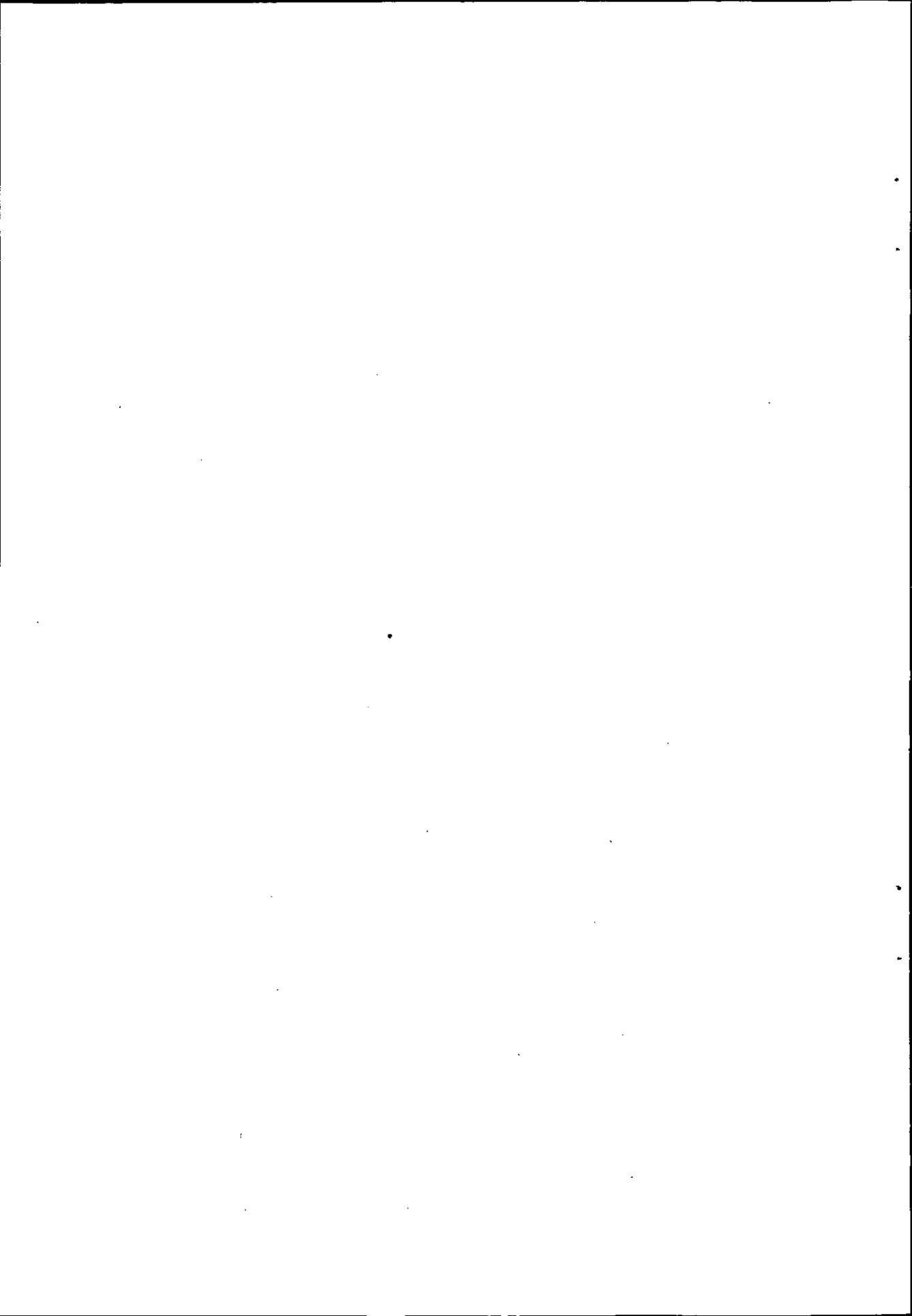
そこで、このような膨大なデータの中から必要なデータを集め、しかも利用しやすいような形に加工し、いつでも必要なときに取り出すことができるように考えられたのが、コンピュータによる情報の検索（IR: Information Retrieval）である。情報検索というのは、つまり、「蓄積してあるデータの中から必要に応じて特定のデータを取り出し、それを利用できる形にする」ということである。広く言えば、百科事典によってある項目の説明を求めると、電話帳を引いてある個人の番号を見つけること、ファイルの中から特定の書類を取り出すことなども、すべて情報検索といつてよい。

これを可能にしたのは、ほかでもなく、コンピュータの大記憶容量、及び高速性、正確性である。しかも、コンピュータの働きであるデータの収集、処理、検索、加工、判断の各機能が、このシステムを可能にしたのである。

ところで、情報検索システムの機能は2つの部分に分れる。その1つは、検索しやすいようにデータを分析整理して蓄積することであり、もう1つは、蓄積されたデータから目的にあった情報を採し出すことである。特に前者は、このシステムを効率よく働かすには重要なカギになっているのである。

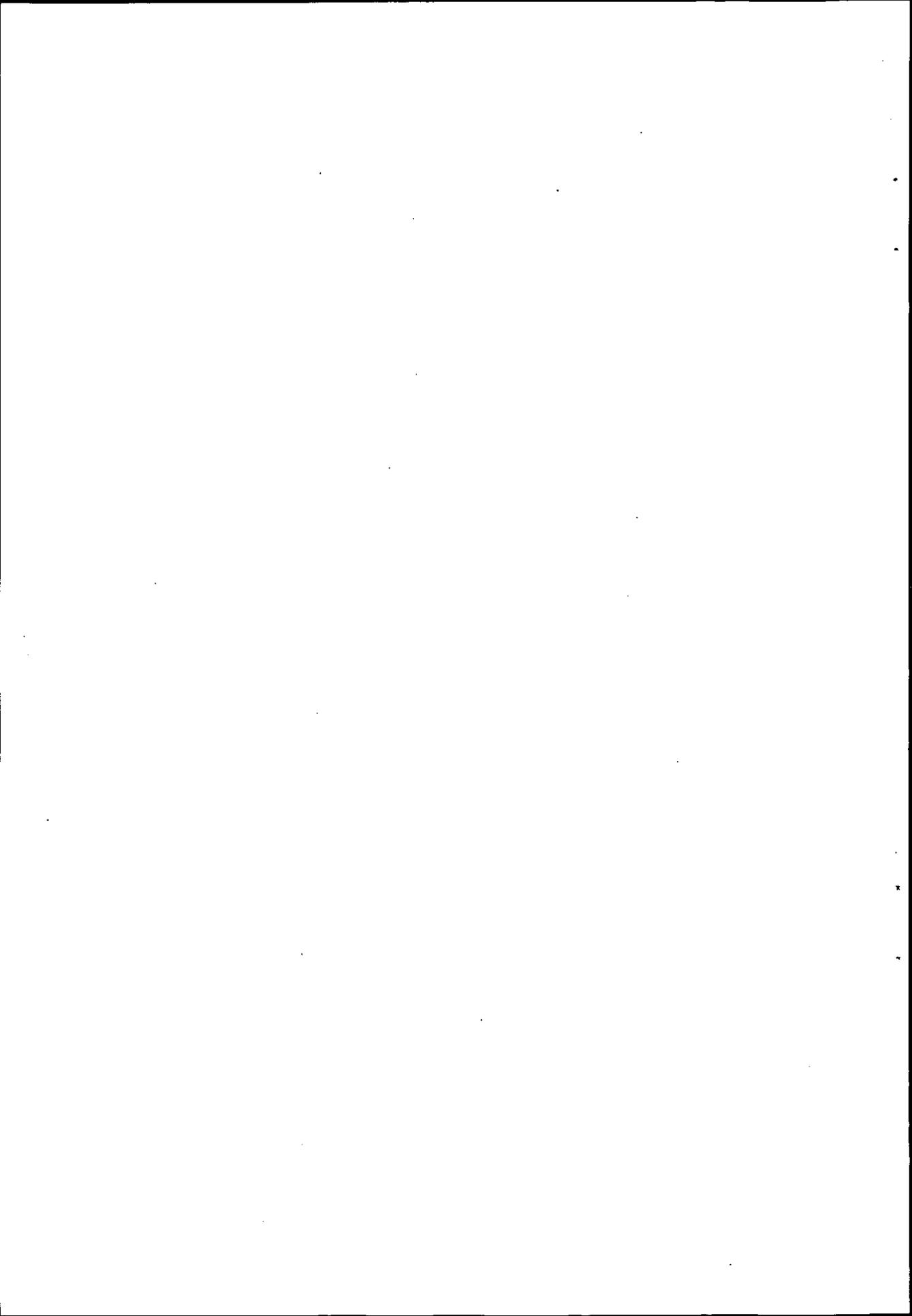
そこで、コンピュータによる情報検索のメリットを考えてみると、1つは、検索速度が速く多数の検索を同時に処理することができる。2つには、多次元の検索質問ができ、質の高い検索ができるということである。

コンピュータによる検索の他に、データの蓄積が少ない場合に用いられるカード方式や、文書や図面など資料類の検索に用いられるマイクロフィルム方式があるが、いずれも量、スピードに限界があり、これからは、コンピュータによる情報検索の必要性がますます高まっていくと思われる。



第 2 部

プロジェクト・マネジメント



# 目 次

I プロジェクト・マネジメント概説 .....	6 5
1. プロジェクト・マネジメントとは .....	6 5
1-1 定型的マネジメントと非定型的マネジメント .....	6 5
1-2 システム思考によるマネジメント .....	6 6
1-2-1 システムとは何か .....	6 6
1-2-2 システムの種類 .....	6 8
1-2-3 システムの特徴 .....	6 9
1-2-4 システム・アプローチの特徴 .....	7 0
2. システム・アナリシス .....	7 4
2-1 システム・アナリシスの誕生 .....	7 4
2-2 システム・アナリシスの内容とプロセス .....	7 8
2-3 システム・アナリシスの特徴 .....	8 5
2-3-1 システム・アナリシスと判断 .....	8 5
2-3-2 定性的分析と定量的分析 .....	8 6
2-3-3 不確定性の問題 .....	8 6
2-3-4 システム・アナリシスのモデル .....	8 6
2-3-5 代替手段の列挙と比較 .....	8 7
2-3-6 比較の基準 .....	8 7
3. プロジェクト・マネジャ、組織 .....	8 8
3-1 プロジェクト・マネジャ .....	8 8
3-2 プロジェクト組織 .....	9 5
3-2-1 プロジェクト組織の基本 .....	9 5
3-2-2 プロジェクト組織への移行プロセス .....	1 0 0
II プロジェクト・マネジメントの技法 .....	1 0 3
1. PERTとCPM .....	1 0 3
1-1 PERT .....	1 0 3
1-2 CPM .....	1 0 6
2. 線型計画法 (LP) .....	1 0 7
3. シミュレーション .....	1 1 0

4.	インダストリアル・ダイナミックス	115
5.	LPの実例	117
Ⅲ	経営情報システム(MIS)	130
1.	情報化時代と新時代経営	130
1-1	情報化時代とは	130
1-2	新時代経営の特徴	130
2.	対象業務システムの分類	131
2-1	大量作業事務	131
2-2	科学技術計算	131
2-3	経営管理	131
3.	MIS設計の特徴	132
4.	経営管理の機能	133
4-1	総合管理	133
4-2	定型的管理(オペレーショナル・マネジメント)	134
4-3	計画	138
5.	システム弾力化の手法	140
5-1	経営情報システム弾力化の例	140
5-1-1	TSS	140
5-1-2	データ・ベース(データ・バンク)	142
5-1-3	モジュラ・プログラミング	145
5-2	経営組織弾力化の例	146
6.	MIS設計の手順	147
6-1	概要調査	147
6-2	基本設計	148
6-3	システム分割	148
6-4	システム構想の確立	149
6-5	長期計画の策定	149
6-6	各種サブ・システムのシステム設計	150
7.	システム設計の注意事項	152
7-1	MIS設計上の基本事項	152
7-2	システム設計上の要諦	152

# I プロジェクト・マネジメント概説

## 1. プロジェクト・マネジメントとは

### 1-1 定型的マネジメントと非定型的マネジメント

マネジメントという考え方には、マネジメント対象の反復性という観点から、定型的マネジメントと非定型的マネジメントとに分類可能である。その場合、プロジェクト・マネジメントは非定型的マネジメントのひとつの方法であるといえる。

定型的という意味には繰り返しがあるということで、プロジェクト・マネジメントには、そのような反復性のある日常業務的なことではなくて、ひとつの目的をもった、いわば例外的なもの・一時的な問題解決を対象としたものであり、非定型的なマネジメントに属する考え方であると考えられる。

それでは、非定型的マネジメントについて学んでいくわけであるが、その場合の組織構造はどうなるかということについて考える。ここに、古い静態組織と新しいプロジェクト型組織の対比についてD. I. クリーランドとW. R. キングによってまとめられた表があるので参照したい。

プロジェクト・マネジメント	機能的部門別マネジメント
<p>〔ラインとスタッフ〕</p> <p>従来の階級的序列制度の痕跡は残っているが、機能的部門のラインは主流としてよりも、支援組織としての役割を果たす。</p>	<p>企業活動はラインの責任のもとに行なわれる。命令系統はラインにあり、スタッフはラインに対し、助言組織としての役割を果たす。</p>
<p>〔階級的序列制度〕</p> <p>縦の関係も存在するが、横のあるいは斜めの関係が重要な役割を果たす。重要な企業活動は、すべてプロジェクトの実施を中心に行なわれる。</p>	<p>企業体のあらゆる分野において、責任と権限は縦に流れる。重要な企業活動は、すべて上下の関係で行なわれる。</p>
<p>〔上役と部下の関係〕</p> <p>同僚と同僚、仲間と仲間、マネジャーと技術者等々の、横の関係が重要である。重要な活動のほとんどが、横の関係で処理されていく。</p>	<p>上役と部下の関係がもっとも重要である。この関係さえうまくいっていれば、成功は保証されたと同様である。重要な活動のすべてが、縦の関係で処理されていく。</p>
<p>〔組織の目的〕</p> <p>プロジェクトのマネジメントは、なんらかの</p>	<p>企業全体の目的が追求される。したがって、</p>

独立性を有するいくつかの組織の共同作業として行なわれる。

〔目的の統合度〕

プロジェクト・マネジャーは、各部門各組織の境界を越えてマネジメントを行ない、プロジェクトの実施という単一の目的に向かう。

〔権限と責任〕

責任が、与えられた権限以上に、重く多様なものとなることがある。各部門からのプロジェクト参加者の給与や勤務評定や昇進に関し、それぞれの部門の長に対し、報告もしくは勧告を行なうという責任さえ、ときとして与えられる。

〔永続性〕

プロジェクトには終りがある。したがって、そのための組織にも終りがある。

目的は単一である。

ゼネラル・マネジャーは、同一の目的をもつ複数の活動をまとめる役割を果たす。

部門別組織図に示すとおり、権限と責任が与えられている。権限と責任はラインにあり、スタッフは助言を行なう。上役と部下の関係が、すべての活動の中心となる。

永遠につづく可能性をもっている。部門別の組織は、それぞれ、それ自身つねにより強力な存在たろうとする。

## 1-2 システム思考によるマネジメント

### 1-2-1 システムとは何か

システムとは、非常にわかりにくい言葉で、コンピュータの場合、コンピュータそのもの、物理的存在としてのハードウェアをシステムということもあり、またコンピュータにのせる対象となる適用業務をシステムということもあり、時としてはプログラムをシステム（特にシステム・プログラムという場合にはプログラムのうちコンピュータ・コントロールを行なうプログラムのことをいう）ということもあるように、はっきりとした定義づけがしにくいものである。

ここでは、一応システム・エンジニアリング関係の本において、システムという用語がどのように定義されているかという、システムの概括的な考え方を紹介する。

- ① マクミランとゴンザレス Mc MILLAN and GONZALEZ )は、「ものおよびその属性が互いに何らかの関係をもって結合されている場合、その集合がシステムである。」（「システム分析より」）と定義をくだしている。

② アーサー・D・ホール ( Arthur D. Holl ) は、

「システムとは要素の集まりで、その要素間あるいは要素の属性間に相互関係が存在するものである。」( "A Methodology for Systems Engineering" より ) と考え、システムとは大規模で複雑な要素の集合体であると端的に割り切っている。

③ ウェブスター辞典は、

「複合的または統一体を構成する事物や部分の集合または組み合わせである」

と定義をくだしている。

④ ジョンソン、カスト、ローゼンツバイク ( Johnson, Kast, Rosenzweig ) は、

「複雑な統一体を形成するもの、または部分の集合または連合であり、組織化された複雑な統一体である。」( "The Theory and Management of System" より ) と定義をくだしているが、同時に著書の中に「システムという考え方はマネジメントのための考え方であり、内的、外的環境要素を統一体として具象化する場合の枠組となるものである。」と考えており、システムとは、具体的に、客観的に存在するという考え方よりも、A WAY OF THINKING つまり、システムというのは、ひとつの考え方なんだととらえている。さらに、つき進めて考えるならば、

⑤ ハーバート・A・サイモン ( Herbert A. Simon ) も、システムを

「一つまたはそれ以上の関係をもつ要素の組み合わせ」

と考えている点は他の学者とほぼ同様であるが、注目すべきはもっと実益のある考え方、つまり実務的観点からさらに突込んだ考え方の展開が、サイモン教授から期待できる点である。

同教授は制御できる変数すなわちコントロールできる変数をシステム要素と名づけ、制御できないつまりコントロールできない変数を環境要素というように分ける考え方である。このことは問題を解く場合に、何が制御できる変数であり、何が制御できない変数であるか——変数とは定量的な観念からそういっているのである。変数といわないで要素といってもよい——を考慮することが大事なポイントであ

る。制御できないものをいくら考えてみてもしかたがなく、結局制御できないものに対してはそれに適応していくより手がないのであるから、適応の仕方や方法を発見することが非常に大事なポイントとなる。と同時に、制御できる変数はできるだけ目標に合致するようにコントロールするということが大切になる。

このように、制御できる変数と制御できない変数にものごとを分けて考え、そこから問題解決の糸口を見つけ出すという方法論が非常に実益のある考え方である。

その意味で、システム的な考え方は A WAY OF THINKING であるという考え方は、実益があるといえる。

#### 1-2-2 システムの種類

システム概念について考える場合、システムを次のように分けると有効であろう。

##### ① 自然のシステムと人工のシステム

今、自然のシステムが人工のシステムの相対的な巨大化のために追いやられているということが、公害問題等で盛んに論議され始めている。このことは、人工のシステムというのが人間生活にとって重要性を増し、それだけに熱中しすぎて、自然のシステムの循環的システムが無視されてきたことのために生じた問題である。その結果、問題が多く起こってきたのであるからまた自然に返れというわけであるが、そう簡単ではなく自然のシステムと人工システムとの調和が問題である。さらに、裏返していえば、われわれの日常生活において人工のシステムが非常に大きなウエイトを占めつつあることを表わしているともいえる。

##### ② 開いたシステム(open system)と閉じたシステム(closed system)

「ほとんどのシステムは、その環境とのあいだで、物質、エネルギー、情報を交換しあうという意味において、開いたシステムである。もし、システムがでんな形においても、情報、熱、物理的物質のようなエネルギーを出したり、入れたりしなければ、そのシステムは閉じたシステムである。」  
(“A Methodology for Systems Engineering”より)

問題を端的に説明するために、前出の公害問題を取り上げると、今まで

地球は無限の大きさを持つ open system であると考えられてきた。

それに対して、人間の営みは相対的に小さいものと考えられてきた。ところが、人間の活動が非常に広範囲にわたり、能率のために高密度化した。大規模化・高密度化の結果、だんだん地球は有限な大きさを持つ closed system に変わりつつある。closed system といっても、太陽エネルギーはもらうわけであるが、地球は地球として全体のエネルギーおよび物質のバランスを保たなければならないということで、無限の大きさを持つと考えられてきた地球が、有限の大きさであることが認識され始めたのである。

つまり、人間の営みの相対的の巨大化によって、前述のようになったということは、経済・技術のいままでの論理というものが、地球は open system であるという考え方のもとに、経済・技術の論理を発展していったところに問題がある。結局、地球が closed system ならば、われわれの経済・技術の論理も closed system でなければならないということで、この open system と closed system という考え方は、今後の技術・経済の論理の展開上非常に大事な要素となった。

### ③ 適応システムと非適応システム

適応システムとは、ひとつの問題を考える場合に、制御できない変数というのが環境要素であるから、環境の変化に対していかに制御できる変数を対応せしめていくかということである。その場合、環境から全く遊離したものが、非適応システムである。非適応システムとはシステムの機能を失った存在価値のない過去の形骸ということになる。

#### 1-2-3 システムの特徴

システムの特徴は、まず第1に定義でも述べたように、まず何らかの形で要素の集まりであるということから集合性ということがその第1の特徴である。第2に要素はひとつでは困るので、要素間において、インターアクションというか、相互関連性が必要である。第3にシステムはひとつの目標をもち、目標の追求性を持っていなければならない。なぜ、目標を持つかというならば、目標を持っていなければ、相互関連性をどう調整したらよいかという価値基準がなくなるからである。第4に環境の変化に対して、目標

に照らして、どのようにシステム要素をコンパインしたらよいか、その行動をどのようにコーディネートしたらよいかということが問題になるので、環境適応性をもたなければならない。

以上をまとめてみるならば、システムの特徴は、

- ① 集合性
- ② 相互関連性
- ③ 目標追求性
- ④ 環境適応性

であり、問題の解決のしかたやその設計を考えるには、以上の4点がどういふところにあるかを具体的につきつめてみるのが、問題をはっきり定義づけることになるわけである。

#### 1-2-4 システムズ・アプローチの特徴

システムを考える場合、a way of thinking であるという考え方をとることが、実益がある旨を前述したが、もう少しシステムズ・アプローチについての注意点を述べる。

##### ① システム要素と環境要素との弁別（権限、時間、状況）

たとえば権限の有無によって、ある場合はシステム要素であり、他の場合にはその同一要素が環境要素ともなりうるということである。権限のない者にとっては環境要素つまりコントロールできない要素であるものが、権限のある者にとってはコントロールできる要素となるので、権限の有無によってシステムの境界が変わるということである。ところが、わが国の場合、これを意思決定について考えた場合、制御できる要素の小さい人が意思決定をして、制御できる要素の大きい人が承認行為をよく行なっているが、これは能率の問題以外に有害なことである。一般に人間は自分の規準に合わせて行動するものであるから、制御できる要素の大きい人が意思決定をしない場合、企業にとって安全性を見込みすぎてマイナスではないかということである。このように権限によって、システム要素と環境要素とは、いれかわるのだということ为先づ考えるべきである。次に、時間について考えるならば、短期計画と長期計画とは一般に非常に矛盾しあうもので、短期計画の利益の積み重ねは長期的な利益になるとは限らない。ひと

つの問題を考えてみても、どの程度の時間を考慮するかによって、システム要素と環境要素はいれかわるわけである。

長期的なものを考えれば、人間の能力は教育によって制御できるシステム要素となし得るけれども、短期的に考えると与件としての環境要素たらざるを得なくなるからである。状況について考えるならば、成長している企業においては制御できる要素はかなり多く、たとえば長期資金の調達は容易であるが、行き詰まっている企業においてはそれは制御できない要素ということになるからである。

したがって、システムというものは、客観的に存在するのではなくて、むしろ問題解決にあたってのひとつの枠組（フレーム・ワーク）、a way of thinking だという考え方のほうが実益が多いと考えている。

## ② 有効性 (effectiveness)

有効性という考え方は能率 (efficiency) を上廻る上位概念で、efficiency という考え方は要素的な思考、effectiveness という考え方はシステム的な思考の概念である。たとえば、コンピュータを使って事務管理の能率をあげるということは要素的思考においてはきわめて大切なことであるが、はたして企業全体に役立つかどうかということとは別問題になってくる。売り上げ高を早く知ることひとつをとっても、マーケット・オリエンテッドの企業で打つべき手のある企業とプロセス・オリエンテッドの企業で対応策の少ない企業とでは、その有効性が違ってくるということになる。あるひとつの要素における能率性の向上が即全体の企業に大きな貢献度をもたらすかどうかは、その企業体の性質、状況によっても違うわけである。われわれがねらうべきことは、むしろ efficiency よりも effectiveness でなければならないということになる。オペレーションズ・リサーチの場合は、メジャーオブエフェクティブネス (measure of effectiveness = 有効性の尺度) といい、決して measure of efficiency とはいわないことに留意されたい。

有効性という考え方はシステムのミッションというか目標というか、使命に対してその達成度合がどのようなか、またその貢献度がどの程度であったかということが判断の基準になるべきで、ただ能率をよくするだけでは

足りないことになるわけである。

### ③ 要素間の争剋 ( conflict )

日本の経営を考えた場合、予定調和説といって各人が努力すればするほど会社はよくなるというように考えられているが、日本経済や企業の現実面を見た場合に各人が努力すればするほど悪くなっている例が多数ある。これについては、P R P T手法のところで一層明確に学ぶことになるが、ここでは、この一番よい例としてチャーチマン教授らの在庫政策に対する組織単位間の矛盾を引用しておこう。

在庫政策に関する各部門のもっているビヘイビアとかあるいは政策というものを考えてみるときに、単位組織間のコンフリクトが明瞭に浮び上がってくるからである。チャーチマン教授らの例を追って行くなら、生産部はいかに安いコストで製品をつくるかということが第1任務となる。コスト・ミニマムの生産を行なうには、ロットの大きさを大きくすればよい、つまり一連の生産をなるべく多くつくれば準備費、あとかたづけの費用が最小になるので、ロットの大きさを大きくすることが、コスト・ミニマムになる有力な手段であるが、その結果は在庫は大となる。次に、販売部は販売に際して売りそこない防止をすることが第1任務であるが、そのためには在庫がどの製品についてもほしいということになり、その結果は在庫は大とならざるを得ない。また、人事部では雇用の安定を希望することにより、需要期と不需要期とが存在する季節的製品の場合には、不需要期の生産ということになり、その結果は在庫はまたも大となる。しかし、財務部や経理部が考えることは資金の固定化防止であり、在庫が多いということは資金がそれだけ固定化するが故に、何時も悪とされ、在庫は少なければ少ないほどよいというわけで、はじめて在庫は小なるべしという主張が出てくる。そして、これらの主張を定量的に追求して、その釣り合いを発見することがO Rの任務であるが、これは計算してみないと誰れもわからない問題で、実際問題としては実力者が勘で決め、他の者は任務を忘れて妥協しているのに過ぎない。些細な問題はそれでもよいが、非常に重要な決定である場合には、計算してみるとどこで調整したら一番システム全体として売りそこないも少なく、在庫も少なく、コストもミニマムな点が発見でき

るかという技法は既に存在している。併しORも費用がかかるから重要な問題についてのみ適用するのが賢明である。

問題は、本来組織というものは、各自の目標を追求すれば追求するほど争烈する点が必ず出てくるもので、その事実認識の上にそれらのコンフリクトをどこで調整するかということを経験することによって、実はシステム的な a way of thinking の意義があるといえる。

#### ④ モデルの利用（数式モデル、決定論的モデルと確率論的モデル）

コンピュータの出現により、社会科学の諸現象も自然科学と同じく実験科学の性質をおびてきた。すなわち、数式モデルの使用により将来の予測や決定因子に落ちがないかどうかということを実験的に追求することが可能になってきたわけである。数式モデルは普通2つの種類があり、決定論的モデルと確率論的モデルとがある。

決定論的モデルというのは、すでにわかっている法則の中から答えが一義的に出てくる性質を有するものであり、確率論的モデルというのは明確なルールがなく、時と場合によっていろいろと異った結果を伴うものであるが、それを全体としてはこういう傾向になるだろうという推定をするモデルであり、例としてはモンテ・カルロ法がある。

〔モンテ・カルロ法〕

J. von Neumann と S. M. Ulam によって1945年ごろシミュレーションの中に導入された語で、“決定論的な数学問題の処理に乱数を用いること”と定義した。現在ではむしろ乱数などを中心とする確率論的問題で、方程式をたてることが困難な場合、あるいは方程式をたてても解くことが困難な場合に、デジタル計算機によるシミュレーションで解く場合を、モンテ・カルロ法とよぶ。（岩波数学辞典より）

#### ⑤ 科学的方法（Scientific Approach）

システムズ・アプローチの敵は何かというと、先入観とか固定観念である。元来、科学的方法とか、科学精神というものの本質はいまのべたような観念を取り去り、実験と観察の中からよりよい可能性を見つけ出そうとする努力である。

#### ⑥ 学際的方法（Interdisciplinary Team Approach）

当初は、異専門領域間の研究と訳されていた語であるが、最近では学際的方法ということ、通用するようになった。これは、違った学問をおさめた人達は違った思考方法、発想法を持っている。だから新しい問題に挑戦する場合に、違った発想法を持っている人々の集団の方が有効であるとの考え方から生まれた方法である。つまり、量的な問題を解決する場合には、同じ専門領域の人（同質性）が集まっていたほうが効率がよく、質的な問題解決には異質的な素養をもったグループの方が有利であるという考え方である。また今日の複雑な社会にあっては、一人の天才よりも異質の集団頭脳に依拠すべしという考え方も同時に存在しての発想でもある。

ただし、このアプローチの方法もミッション・オリエンテッドな態度があって始めて有効であって、日本の縦社会構造の中では難かしい、なかなか実行しがたい方法論である。しかし、これはシステムズ・アプローチの重要な方法論になっているので、こういうことを体得し実行可能にしなければ、とてもわが国では新しい問題の解決はできないことは確かであるので、どうしても70年代には克服しなくてはならない創造性発揮の関門といえよう。

## 2. システム・アナリシス

### 2-1 システム・アナリシスの誕生

システム・アナリシスは、1948年、C・J・ヒッチ、S・エンク博士などによって開発された。1961年、マクナマラ国防長官の下で具体的にはPPBS（Program-Planning-Budgeting System）の中に組み入れられたのである。

その内容は、大別4つの部門から成り立っている。目的、任務（プログラム）、費用—効果、予算である。以下その概要をみてみよう。

目的（ミッション）を達成するために、まず計画を立てる。米国における国防の例では、戦略報復兵力、一般目的兵力、情報収集及び通信、などに分けた9つのプログラム別成業計画（任務別計画ともいう）を立てている。すなわち国防という任務を達成するための企画と行動を選り分けて、具体的な計画に再組み分けすることが必要なのである。それらはすべて1つの統一活

動としての「国家安全保障」を目的として関連づけるのである。プログラム別成業計画を構成する要素は、2つの面から考える必要がある。1つは部隊の数、航空隊の数、発射台のミサイルの数、艦艇の数といった構成要素の具体的な定員を数字で明確に示すことである。他の1つは、それに要する費用を明確にすることである。費用は3つに区分して検討し、作成する。

- ・ 研究開発用費用
- ・ 投資費用
- ・ 運用費用

である。

研究開発用費用は、新しい兵器体系または新しい兵器を開発して、戦争能力を実際に利用可能な状態にするまでの費用である。すなわち一般企業での新製品開発に必要な費用の見積りと同じであって、これを開発し、具体化するまでには莫大な額の費用がかかる。目的に対して、適切なものであるのかといった点が大切になり、全体規模での開発、すなわち開発の最終段階に進むか否かは重大な意思決定である。そこでの評価を終わってから投資が決定されるのである。

投資費用は開発が終わってから、これを作戦用途に導入するまでに必要な費用である。これには兵器体系の調達、陣地の建設などが含まれている。

運用費用は年々の運用維持、兵員のための費用である。歩兵師団のように一年間の運用費用が初年度投資よりも大きなものもある。

これら3つの必要費用は5ヶ年間程度にわたって見積られる。以上のようなプログラム構成要素と費用見積りは、概念的に図I-1のようになる。

こうしたプログラム要素を組み立て、長期にわたるデータを揃えて、戦闘計画の内容を明らかにし、同時に同様な任務を達成するために考えられる他の代替体系と比較して、そのプログラム別成業計画の費用—有効度を評価することになる。それが平等の配分（資源の配分）についての適正度を判定する基準になる。

これらのプログラム別成業計画を予算という期間計画へ、年間毎の費用として把握するために、費用項目を予算の支出項目に分ける必要がある。費用項目を予算の支出科目区分に従って分けておく。図でもそれぞれの費用項

図 I-1 プログラム要素

プログラム要素 F-202									
	年度 68	64	65	66	67	68	69	70	71
大 隊						1	2	6	6
中隊(20機/中隊)					1	8	6	18	18
航空機計					20	60	120	360	360
地对空ミサイルGAM-99 (25/中隊)									
ミサイル計					25	75	150	450	450

全 債 務 権 限 (100万ドル)

研究および開発									
研究、開発、試験、審査	100	200	300	200	50				
建 設									
計	100	200	300	200	50				
投 資									
航空機調達			150	500	900	750			
ミサイル調達			15	20	25	25			
その他調達					1	1			
建 設									
計			165	520	926	776			
運 用									
運用および維持									
航空機調達				1	2	3			
ミサイル調達			1	3	5	7			
その他調達						1			
軍事要員					1	5			
計			1	4	8	16			
全債務権限計	100	200	466	724	984	792			

	68	64	65	66	67	68	69	70	71
人 員 (1000人)									
軍 人									
将 校					0.1	0.3			
下士官兵					0.3	1.5			
シビリアン									
直 備									
契 約									

目を予算書の支出項目区分に対応させているのである。

しかし国家予算の編成で使われてきている用途別の積算方法と、マクナマラ方式での目的達成に必要な任務別なプログラム事業計画の予算積算方法とは、別々の予算体系となっている。それでヒッチは、両立てにしてそれぞれの長所を生かすシステムを考え出した。「この結果、予算編成にあたっては、任務別計画を用途別予算に翻訳し交換するという事務的めんどうが生じ、また承認せられた予算を任務別計画に反映するためには、逆の変換を行なわなければならない。しかしこれはいわば機械的な手間であって、現行予算の持つ利点は、このような手ばを十分償うものである。しかもこの種のめんどうは、コンピュータの利用によって十分克服することのできるものである。かようにして、任務別に計画し、資源別に年度予算を組むことによって、国防省は任務と戦略という文脈において管理を行なうと同時に、資源別による管理をも遺憾なく行なうことができる。」としたのである。

以上概要をみたように、3軍（陸海空）の国家防衛目的への貢献という立場から、統一した目的達成の方法を手順化して明確化したのが、PPBSといわれるものである。こうしたPPBSの編成のステップの中で研究され、検討され、具体的に展開されたのが、システム・アナリシスなのである。すなわち

1. 目的—任務
- ↓
2. 代替手段—費用分析（経営性分析）
- ↓
3. 選択手段
- ↓
4. モデル化
- ↓
5. 費用—効果分析
- ↓
6. 予算化

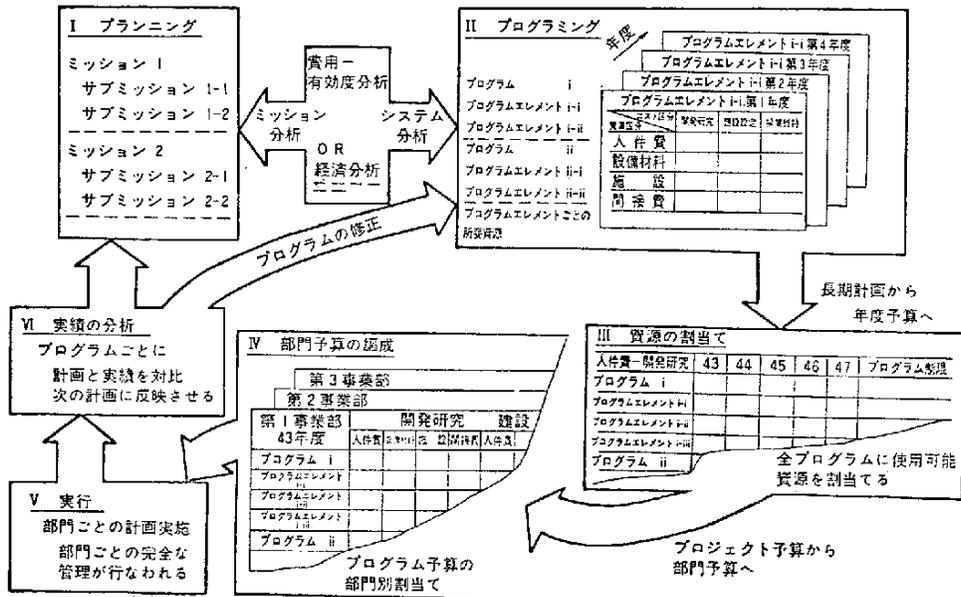
といった内容が含まれ、1→6がそのステップと考えることができよう。これらがくりかえされたり、部分的な評価が任務へのフィード・バックが行なわれる。そして内容が規定されて目的が達成されるのである。

こうした三軍統一を企画して、目的達成をねらったPPBSの展開を中心に開発されたシステム・アナリシスの方法や、スケジュール管理の考え方や方法は、一般企業においても、企業経営の立場で企業目的を達成するための

用具として理解されだした。すなわち企業経営のシステム化とその効率化のために利用され初めたのである。

目的をじゅうぶんに検討し、その中から必要な「任務別計画」を設定する。それらの具体的諸資源を数量的に把握し、それを費用として、大別3つのステップで計算する。そしてそのプログラムの構成要素を、目的別に明確に費用—効果分析を行なって最適化し、それを予算化する、といったステップが企業経営の多角化か競争戦略の設定でも有効な創造的、代替効果を含むアプローチであると同時に、そこを流れる基本的目的、目標志向性企業活動の本質を決定するものなのである。それは、経営学的な効果性の分析、PERTの利用、ORの利用などその諸方法が総合されており、最も効果的な戦略設定に役立つものといえよう。

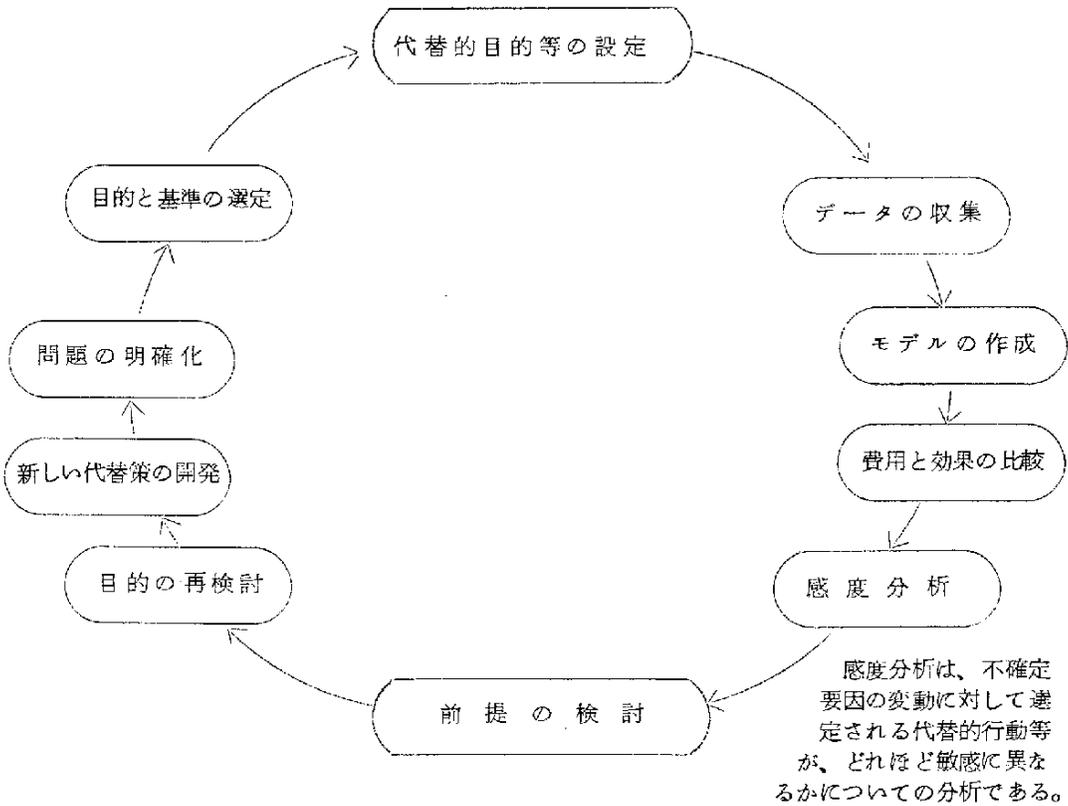
図 I-2 PRBSのサイクル



2-2 システム・アナリシスの内容とプロセス

アメリカのランド・コーポレーション (Rand Corporation) のクエード (Quade, E・S) は、次のように定義している。「システム分析は、行

図 I-3 システム・アナリシスのサイクル



動の目標を明らかにし、その目標を達成するための諸方策を列挙し、それらの費用や有効度や危険度を比較して、目標達成のための方策を提示することをねらいとした、意思決定の問題に関する探求の1つの形態である。そして、それは不確定性の環境の中で、複雑な選択の問題を解決するためのアプローチの方法や考え方を提示するものである。」そして図 I-3 のようなサイクル活動を示している。

システム・アナリシスの特徴とされる「費用・効果（便益）分析」は、P P B S の予算編成の過程で行なわれ、代替策を徹底的に比較検討することによって、その効果が認められたものである。

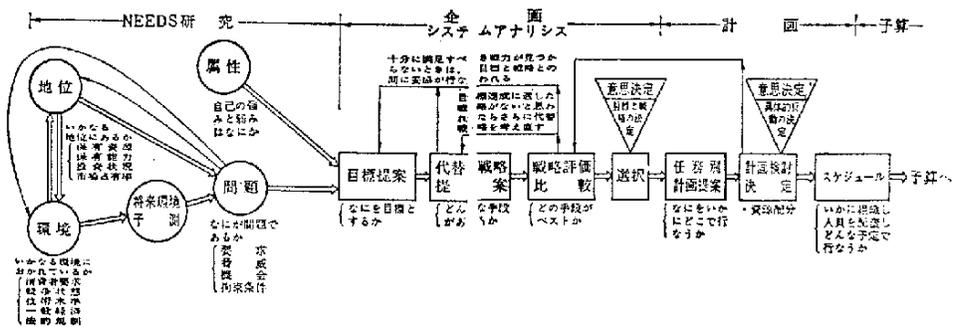
S・L・オプナーは、「定性的かつ定量的属性をもった混合問題を解くための技法がシステム・アナリシスである。」と述べている。またクリーランド、キングは、「システムの概念が、マネジメントのプランニングの段階で適用された時、それはシステム・アナリシスと呼ばれる。システム・アナリ

シスの目的は、マネジメントのプランニングの段階において、意思決定問題を解決することにある。」と述べている。以上の内容からシステム・アナリシスのステップを次のように設定することができる。

1. 目標・目的の設定（達成しようとする目標がある）
- ↓
2. 目標・目的達成の手段の列挙（達成する手段方法はいくつがある）
- ↓
3. 目標と手段の相互関係の明確化（目的と手段・方法・それに必要な資源の相互関係を明らかにする、費用—効果分析を含む）
- ↓
4. 選択の判定基準の設定（各々を比較検討し選択する基準）
- ↓
5. 不確定要因の列挙と可能性検討（不確定なものを明らかにし、その可能性を検討し、その扱い方を見極める）
- ↓
6. 資源としての時間要素の検討（タイミングを考える）
- ↓
7. 総合的な検討（部分的な検討だけでなく総合的に考える）

以上のようなプロセスで、システム・アナリシスを行なうことができる。システム・アナリシスはPPBSの開発にその端を発していることは前述し

図 I-4 システム・アナリシスのプロセス



た通りであるが、その基本的な目的志向をPPBSに範を取って、システム的に戦略設定を行なった企業がある。インターナショナル・ミネラルズ・アンド・ケミカル社の場合は、独自に図I-4のような展開を行っている。

システム・アナリシスの第一のプロセスが「ニーズの研究」である。すなわちニーズの研究が最も大切なシステム・アナリシスの前提として示されている。次のステップは、企業によってその展開すべきシステムはなんであるかをハッキリさせることの必要性を示している。

企業の中で考えるシステムは、戦略としての企業目標・目的に合致させることによって成り立つ。PPBSが「国家安全保障をどう確保するか」の目標からスタートしたと同じように、企業の目標・目的の設定が大切なのである。

企業活動の中では、目標・目的の設定・すなわち環境適応のステップから考える必要がある。「戦略設定を含むシステム・アナリシスの手順」について以下説明する。(図I-5参照)

まず第1は現状の分析である。企業のおかれている社会的経済的諸条件の検討を必要とする。すなわち企業をとりまく環境がどう変わってきているかを分析するのである。この分析によって、存続と成長を願う企業が適応すべき環境が問題となるのである。企業の主体制において、いかなる業種に属しているか、それは外部適応に於いて問題はないか、市場や顧客がどうであるか、などが環境要因として分析される。(環境要因分析)

第2にそれを受けて立つ企業主体側に、一体その環境変化を受け入れる体制ができているのか、企業の果たしている役割が分析される。(企業主体性分析)

第3に必要なことは「変化予測」である。将来の変化要因とその変化によって、どんな体制や市場が需要構造になるのかを予測することである。経済、文化、社会、政治の面での変化、また最近問題になっている「技術予測」も将来の変動予測である。(変化予測)以上の内容を大きく「環境研究」と考えることができる。

第4に対応検討である。客観的变化に対して、自社をどう順応せしめるか。又はチャレンジするかを検討することである。それには企業の長所・短所を

分析することが必要である。アンソフは、設備や施設、人的技能、組織的能力、管理能力に区分して、それを一般管理財務、研究開発、生産、マーケティングに区分してチェック項目を示している。

	一般管理財務	研究開発	生産	マーケティング
設備や施設				
人的能力				
組織的能力				
管理能力				

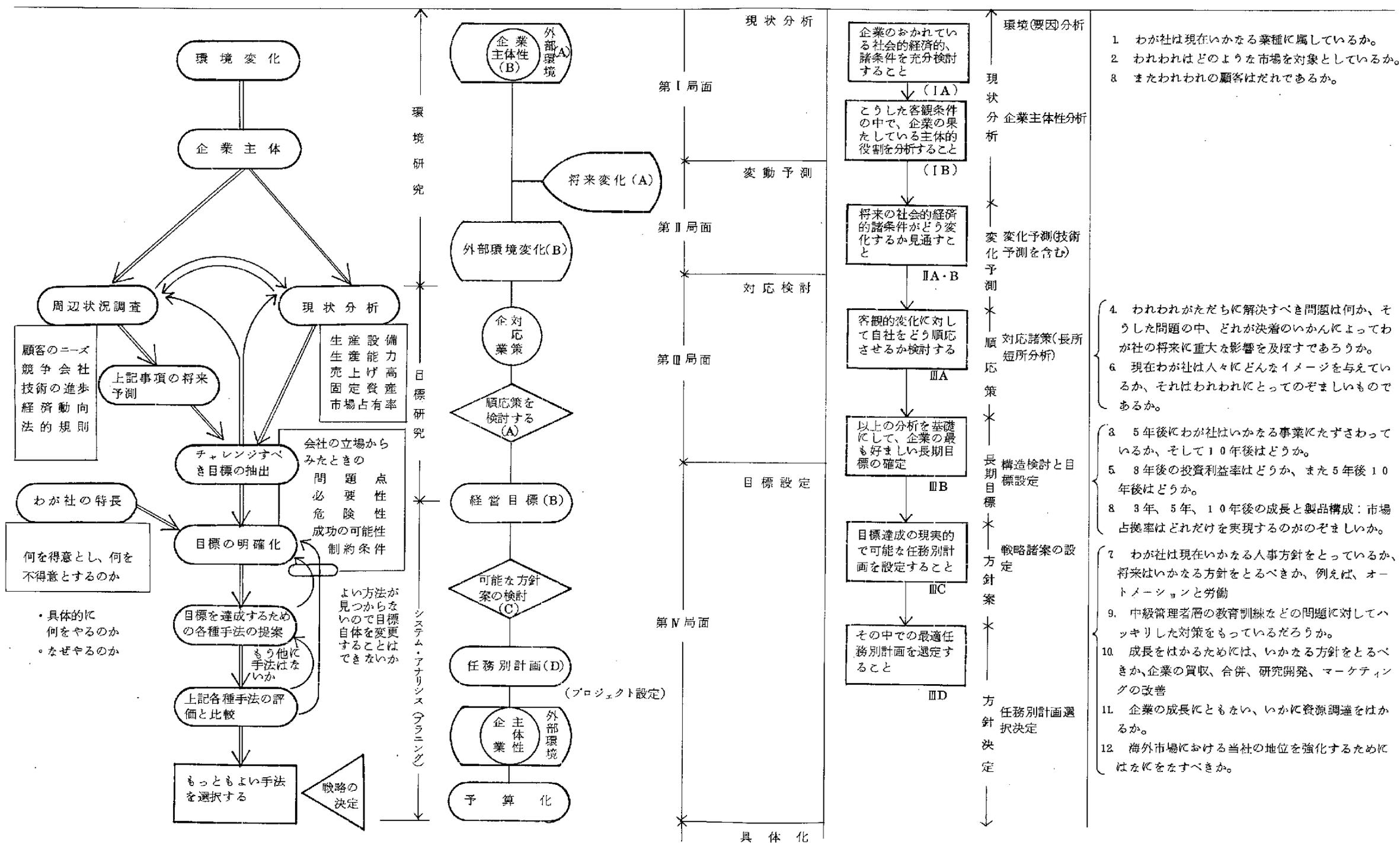
こうして、自社の現状を把握することによって、「敵を知り己を知らば、百戦してあやうからず」の体制ができるのである。企業の最も好ましい長期目標が確定されるのである。すなわち、構造検討と目標設定がなされる。

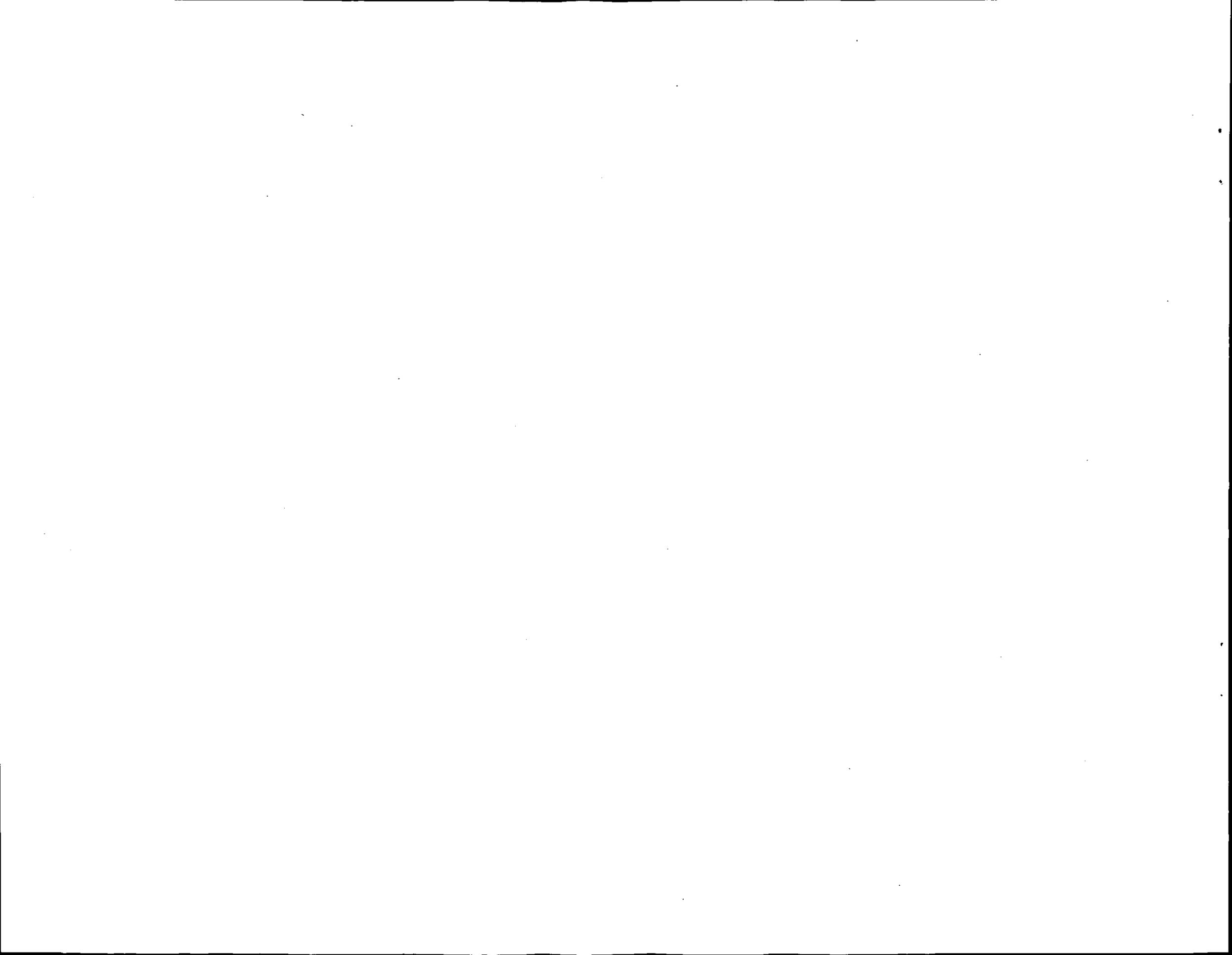
以上の内容は、外部環境変化に対応する企業側の対応と目標設定のプロセスであり、戦略が設定される過程である。このプロセスで目標研究とシステム・アナリシスとが必要となる。特に最も有利な戦略と資源の再配分を行うための構造（組織）の検討が必要となる。（目標設定）

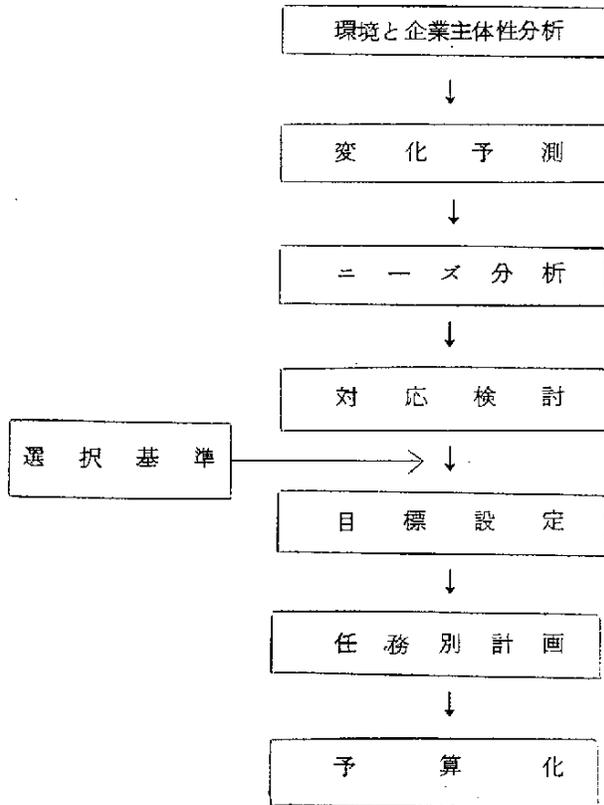
最後のステップは、期間計画としての予算化へのステップである。このステップは企業の場合、効率性を中心にプロジェクトが評価されて、期間の中で具体化が決定され、実施されるステップである。（具体化）

以上のステップを経て、企業の戦略決定の中でシステム・アナリシスの役割が完成する。システム・アナリシスは、今までのヒューリスティックな意思決定に対して、具体的な内容をもつプロジェクトを、創造性と客観的な経済的妥当の上に、代替策を考慮に入れた効果的な意思決定をしようとするものである。

図 I-5 戦略設定におけるシステム・アナリシスの手順







### 2-3 システム・アナリシスの特徴

システム・アナリシスは従来の分析と幾つかの点で異なった特徴を持っている。例えば、

1. 経済的な問題としての資源配分（費用—効果分析）
2. 不確定性と選択権（OR、基準決定）
3. 代替計画
4. 予算との結びつき

などは、前述したようにそれらはいづれも従来のアプローチと異なった内容をもつものである。ここではクリーテンド、キングの所論に従ってさらに補足検討してみよう。

#### 2-3-1 システム・アナリシスと判断

人間の判断は2つのかたちで、システム・アナリシスの中に持ち込まれる。1つは、「蓋然性の判断」であり、「明日は雨だろう」といった種類の判断

である。他の1つは「価値の判断」で、マーケット・シェアか利益率かの問題で、どちらを取るべきかは、デシジョン・メーカーの手で行なわれるものである。

システム・アナリシスのメリットの1つは、こうした「蓋然性の判断」と「価値の判断」の2つの種類の判断がハッキリ区別され、それぞれ独立して行なわれるようになってきている点である。

### 2-3-2 定性的分析と定量的分析

システム・アナリシスは科学的な分析手法である。科学的分析は現実の複雑な問題を、抽象化し単純にするための論理的手法である。この科学的分析の結果と分析の対象にならない要素に対する判断を総合して、最善の意思決定を行なうことができる。科学的な客観的分析と非科学的な主観的分析を自由に駆使し、最善の意思決定を行なうことができる。

### 2-3-3 不確定性の問題

プランニングは、必ず未来にかかわりを持つ。未来は不確定性の世界である。意思決定が分析に基づいて科学的に行なわれても、直観だけに依存して行なわれても、不確実であることに変わりはない。しかしシステム・アナリシスの利点は、この不確実なものを徹底的に分析する。主観的なものにしろ客観的なものにしろ、意思決定の各種の方法論において、不確実性の分析を含めているのは、システム・アナリシス以外にない。

### 2-3-4 システム・アナリシスのモデル

システム・アナリシスの選定は、モデルによる。モデルは、目的と戦略と外的条件との組み合わせである。モデルを使うことによって、定量的な予測は、それを使わない場合よりも、はるかに容易に行なわれる。システム・アナリシスでは、2種類のモデルが使われる。コスト・モデルとベネフィットを予測するベネフィット・モデルである。モデルの価値は、基本的には、それが現実の世界のあるもの、すなわちシステムの代りの役を果たすことにある。従って、問題解決のための分析にもっとも必要なのは、現実の世界のシステムを模倣するところのモデルである。モデルを使うことによって、アナリストは、システムの構成要素の相互関係を明らかにし、それらの要素の変化が、システム全体の働きにいかなる影響を与えるかを知ることができる。つまり

現実のシステムの代用品を使い、その代用品によって実験を行ない、システムの構成要素における変化が、システム全体の働きにいかなる影響を与えるかを予測することができる。

### 2-3-5 代替手段の列挙と比較

目的達成のための手段をあげて、その手段の達成度合いを比較するプロセスは、一見する以上にむづかしいものである。選択の対象となる代替手段の数が比較的限定的である場合には、列挙されている中から1つを選択するという比較的簡単なプロセスになる。列挙された代替手段のほか、まったく新しい手段を考えなければならないという場合には、すでにあげられた幾つかの案では、目的をじゅうぶんに満足させることができない場合、或いはそれらの手段ではコストがかかりすぎる場合などが考えられる。こうしたまったく新しい代替手段を考察する場合には、非常に多くの人間の力を必要とする。とても1人の人間があらゆる必要な能力を身につけるといいうわけにはいかない。そこで、システム・アナリシスでは、各方針の専門家をあつめた混合チーム（プロジェクト・チーム）が大きな役割を果たす。

新しい代替手段を開発する段階だけでなく、各プロセスで各種の専門家が必要とされる。

従って、プロジェクト組織にはマネジャがこうしたシステム・アナリシスのプロセスを円滑化させるためにも是非必要なのである。

### 2-3-6 比較の基準

目的達成のための諸案を評価する場合の基準は、コストとベネフィットでなければならない。コストはその計画案を採用した場合の資源（人・もの・金）であり、ベネフィットとは、その手段を用いた結果をもたらされる利益であり、便益であり、価値である。この2つが代替案を評価・比較する場合に、不可欠の基準である。目的達成のためには、この両面から同時に評価することが必要である。例えば、家を立てる場合にいくらコストがかかるからといって、水道管やガス管の配管をしなかったり、また入口がない家を建てる人はいない。しかし、コストだけをシステムとしてとらえると、こうしたことがおこるのである。

システム・アナリシスが非常に広い範囲のものであることを示す例として、

戦時中に軍の戦略行動を決定する方法として考えられたORとの比較を図I-6に示す。

上記の内容から、システム・アナリシスは企業活動の基本部分へアプローチするものであることがわかるだろう。

図 I-6 システム・アナリシスの特徴

	システム・アナリシス	O. R
問題の範囲	広い	狭い
問題の次元	高い	低い
問題の期間	将来の問題が多い	当面の問題
問題の定量化	必ずしも可能でない	可能
問題の目標	目標の検討そのものが課題となりうる	明確で動かすことができない
分析の目的	What to do	How to do
分析の手法	固有の分析手法をもたない	固有の手法をもつ
分析者	各領域の専門家	数学・統計学・工学などの専門家

以上、幾つかの点において、システム・アナリシスの特徴についてのべたが、これらの特徴もシステム・アナリシスを深く研究することによって、じゅうぶん理解できるものである。要は問題へのアプローチとして、システムとしての見方や態度が必要であり、この点が見失なわれると、内容や技法は何の役にも立たないものになってしまうのである。

### 3. プロジェクト・マネジャ、組織

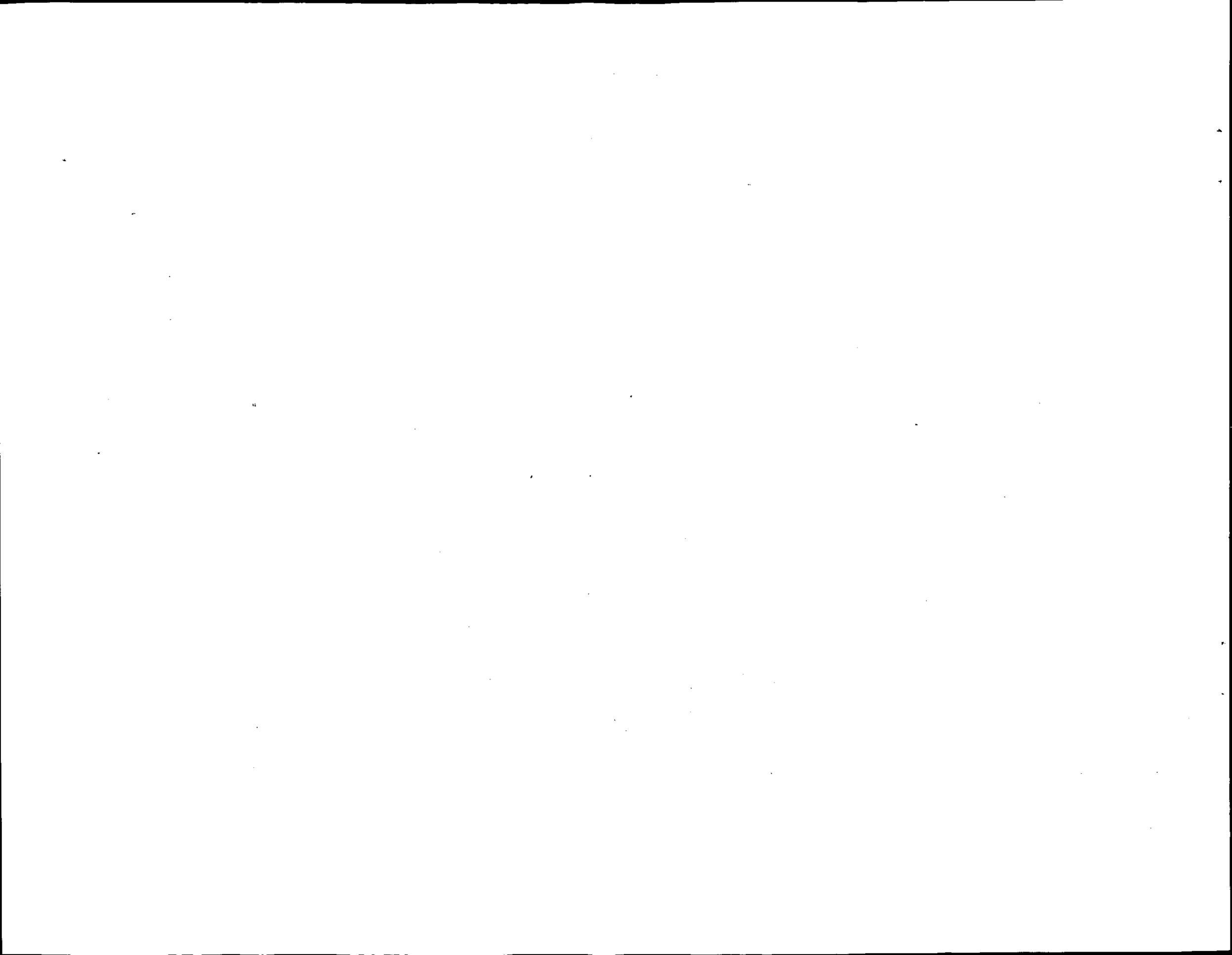
#### 3-1 プロジェクト・マネジャ

プロジェクト・マネジャは、プロジェクトを管理する役割を持ち、その職務を遂行するために、人・もの・金の資源を有効に活用して、一定の品質のプロジェクトを決められた納期、決められたコスト内において完成するよう指揮・監督する。

すなわち、プロジェクトの大小にかかわらず、そのプロジェクトについて、質・コスト・工期を管理し、プロジェクトの着手から完成まで最適かつ最短

表 I - 1 プロジェクト・マネジャの職務

基本任務と権限	プロジェクト・マネジャの仕事	プロジェクト・マネジャーの職務権限	プロジェクト・マネジャーの任務と権限
<ol style="list-style-type: none"> <li>1. プロジェクト活動内容の決定</li> <li>2. 組織の機軸強化</li> <li>3. 管理と技術に関する意思決定参加</li> <li>4. 予算、日程に関する意思決定参加</li> <li>5. 人員の獲得と配置</li> <li>6. プロジェクト・チームの決定</li> <li>7. プロジェクト・チームの統一性維持</li> <li>8. プロジェクト・プランの作成</li> <li>9. プロジェクトに関する情報システムの確立</li> <li>10. プロジェクト・オーガニゼーションの形態決定</li> <li>11. 顧客との連絡・接触</li> </ol>	<ol style="list-style-type: none"> <li>1. プロジェクト仕様書の作成</li> <li>2. プロジェクトに必要な技術水準設計</li> <li>3. プロジェクト見積</li> <li>4. プロジェクト工程表</li> <li>5. 資材・設備の購入、レンタル契約</li> <li>6. 下請業者の選定</li> <li>7. プロジェクトの進捗</li> <li>8. プロジェクトの監督</li> <li>9. プロジェクト予算の管理</li> <li>10. プロジェクトの進捗状況報告</li> </ol>	<ol style="list-style-type: none"> <li>1. 受注交渉、顧客との技術的折衝</li> <li>2. 積算及び見積書の提出</li> <li>3. 契約の締結(契約の更新、解除を含む)</li> <li>4. プロセスオーナー、顧客、下請業者とのライセンス契約、秘密保持協定書の締結</li> <li>5. 関係官庁、団体協会などとの対外接渉</li> <li>6. プロジェクト運営方針、人員編成の立案</li> <li>7. 設計・提携プロセスの選定</li> <li>8. 資材・機器の調達</li> <li>9. 建設用車輛、機器、仮設材の調達</li> <li>10. 総合工程表、施工計画書の作成</li> <li>11. 下請業者の選定及び発注</li> </ol>	<ol style="list-style-type: none"> <li>1. プロジェクト関係活動の中心的位置を定めること</li> <li>2. 機能的諸部門と折衝を行なうこと</li> <li>3. プロジェクトの統一性を保持し、そのため関係活動の調整を行なうこと</li> <li>4. プロジェクトの重要な意思決定に必ず積極的に参画すること</li> <li>5. 必要な人材を確保すること</li> <li>6. 資源配分、支出をコントロールすること</li> <li>7. 下請業者の選択と契約条件の折衝を行なうこと</li> <li>8. プロジェクトの実施を妨げる意見の対立を解決すること</li> <li>9. プロジェクト情報システムを確立すること</li> <li>10. 全期間の管理・技術の改善を行なうこと</li> <li>11. 活動分析とフォーマル権限を文書によって明らかにすること</li> </ol>
<p style="text-align: center;">具 体 的 任 務</p>	<p style="text-align: center;">グレメックスの実務能力</p>		
<ol style="list-style-type: none"> <li>1. 日程の変更</li> <li>2. プロジェクト関係活動の優先順位づけ</li> <li>3. 要求・性能の変更</li> <li>4. 予算化</li> <li>5. 契約条件の変更</li> <li>6. 資源配分の変更</li> <li>7. 調査方法(社内生産・社外調達)の決定</li> <li>8. プロジェクト活動の重点変更</li> <li>9. プロジェクト支援活動</li> <li>10. 設計の変更</li> <li>11. 下請発注の中止</li> <li>12. 人員の採用・配置</li> <li>13. 人員の機能的部門の復帰</li> <li>14. 貢献者に対する報奨</li> <li>15. 予算の変更</li> <li>16. 目標原価の再調整</li> <li>17. その他</li> </ol>	<ol style="list-style-type: none"> <li>1. 契約者の選定法</li> <li>2. 契約の型の比較</li> <li>3. 研究開発実行の際の戦略</li> <li>4. 計画上の余裕、信頼性、費用、性能、予算期日、及び指定された性能間のバランスの取り方</li> <li>5. 不測の事故に対する対処のし方</li> <li>6. Pert.net workの活用</li> <li>7. 情報の詳細の程度と価値の認識</li> <li>8. 計画の統制</li> <li>9. 費用、日程へのファクターの認識</li> <li>10. 技術的性能と契約に伴うリスク比較</li> </ol>	<ol style="list-style-type: none"> <li>12. 実行予算書の作成</li> <li>13. 工事中各種保険の付保決定</li> <li>14. 試運転要領書の設定</li> <li>15. 余剰資材、残材の処分</li> <li>16. 所属員の人事管理・教育訓練</li> </ol>	
<p style="text-align: center;">「システム・マネジメント」P 325」</p>		<p style="text-align: center;">千代田化工建設株式会社</p>	<p style="text-align: center;">「システム・マネジメント」P 313</p>



の条件を満足させて遂行する。そうした全責任を持つのがプロジェクト・マネージャである。

プロジェクト・マネージャは、プロジェクトを管理するスペシャリストであり、特に組織能力が最も必要とされる。一般的に、数人の部下から、200～300人程度のチームを編成して、特定のプロジェクトを担当する時（または幾つかのプロジェクト・リーダーの責任者をいう場合もある）、そのチーム・リーダーがプロジェクト・マネージャと呼ばれる。プロジェクト・マネージャはそのプロジェクトについての全責任者であるが、全責任を持つといってもただ1人でプロジェクトに関するあらゆる仕事を処理するわけではない。別の言い方をすれば、そのプロジェクトに関するコーディネータといってもよい。つまり、そのプロジェクトに参加する技術者や、関係するあらゆる部門のいろいろな担当者の協力のもとに、プロジェクトは計画され、遂行され、完成されるのである。その中心となって運営する責任者が、プロジェクト・マネージャなのである。

プロジェクト・チームが編成されると、チーム・メンバーに対するプロジェクトに関する一切の指示命令権は、プロジェクト・マネージャが持つ。一定期間と一定のコストで決められた技術レベルのプロジェクトを完成させるためには、プロジェクト権限の行使がどうしても必要になる。その権限を行使することによって、参加者に命令し、必要な資材を調達して、期間内で仕事が進められるのである。

ただ、プロジェクトとして仕事を進める場合は、参加者の自由な活動が必要であり、特に研究者や専門家の参加が必要な場合、相当程度の自由創造が求められる。そうした配慮がないと、参加者の創意やモラルが低下し、参画意欲を失い、結局技術レベルの確保がむずかしくなる。従って、プロジェクトの進行のための権限の行使と、参加者の意思を尊重した自由な活動との両面をバランスさせながら管理する事が大切になる。

また外部の業者に対しても、このプロジェクトに関する一切の関係について会社を代表する窓口となる。そこでの交渉やコスト・品質・納期の責任と進度の管理が要求される。

プロジェクト・マネージャの仕事の内容は

- ・プロジェクト仕様書の作成
- ・プロジェクトに必要な技術レベルの設計
- ・プロジェクト見積もり
- ・プロジェクト工程表
- ・資材・設備の購入請求、m/c レンタル契約
- ・下請業者の選定
- ・プロジェクトの進捗
- ・プロジェクトの監督
- ・プロジェクト予算の管理
- ・プロジェクトの進捗状況の報告

などがあげられる。すなわち、

- ・コスト
- ・品質
- ・納期

を中心とした活動が管理基準となる。マネジメントの活動は一般に計画・組織・指揮・統制の職能をもっている。あらゆる階層のすべての管理者がその役割を果たすためには、4つの職能をも遂行しなければならない。

表 I-2 プロジェクト・マネジャの仕事

	計 画	組 織	指 揮	統 制
品質、技術(Q)				
費用(C)				
納期(D)				

大型のプロジェクトの場合には、トップ層がみずからプロジェクト・マネジャとなって活躍することになるし、小型のプロジェクトになるに従って、部課長クラスがプロジェクト・マネジャになる。プロジェクトの効果的運用のためには、企業内の組織やそのプロジェクトの内容、特に技術的なレベルに精通したベテランがそのプロジェクトのマネジャになる必要がある。

リーダーとしてプロジェクトをマネジメントし、担当する人間のパーソナリ

ティも大きく影響する。一般にいわれているリーダーの資格要件としては、次のものがあげられる。

- 与えられたプロジェクトに対する熱意と興味を持ち、実行力のある人
- プロジェクトの内容に対する十分な理解力があり、同時に全社的な判断のできる人
- プロジェクトのメンバをまとめて、目標達成に協力させる指導力のある人
- 従来からの定常組織の協力をうるための渉外力とトップへの説得力のある人

より少ない投資額で、より多くのアウトプットを生み出すようなプロジェクトの完成を、出来るだけ最短工期で完成させるかどうかは、プロジェクト・マネージャの力量に左右される事がきわめて大きい。

こうした能力・資質をもったマネージャの育成が必要である。それは専門領域での経験の積み重ねが必要となり、人材の確保がむずかしいのである。

(注) プロジェクト(Project)とは何かについて

日本IBM、システム開発部長藤枝氏は次のように述べている。

「プロジェクトは“特殊任務事業計画”として理解されているが、それは通常の作業、計画、あるいはプログラムとどう異なるのであろうか。それは、次の“プロジェクトの特性”から考える事ができる。

- (1) 長期的開発作業計画
- (2) 戦略的で企業ポリシー的特務事業
- (3) 各種機能の融合と統制
- (4) 高額投資と競合性
- (5) 未知な開発分析、導入管理技術とそのリスク

また、プロジェクトの言葉の意味から、“前面、将来に向かってある実存の典型として、思考を、計画を、設計を、長期的に投げ上げ、突出せしめ、投影し、主観的存在を客観的、生産的存在に転換する作業をプロジェクトという」

ケース・スタディ

次のケース・スタディは「マンハッタン計画」のプロジェクト・マネージャであった。グローヴス将軍の人となりとやり方である。

〔設問〕次のケース・スタディの場合プロジェクト・マネージャとしての資質要件やマネジメント・システムからみて、何が問題かを考えて下さい。

グローヴスがタフでしかも人のことに無頓着なことは有名であった。グローヴスは自分の評判を知らないわけではなかったが、それだからといって自分をかえるつもりは毛頭なかった。彼の関心事はただ1つ、仕事の完成、だけであった。たとえ他人の感情を傷つけても「仕事が完了した」かどうかだけに気をくばった。グローヴスの部下は大抵彼をおそれた。彼を好きな人はほんの少数だった。しかしグローヴスはこの方針をかえなかった。彼は自分を尊敬することだけを強要した。そしてグローヴスは尊敬された。目上の人からも、部下からも、また同僚からも。

グローヴスは1918年のウェスト・ポイントのクラスを4番で卒業していらい、その模範的な働きぶりと、能率のよさと軍に対する献身とで有名であった。彼が仕事を引き受きうけたときけば、上官は人間にできる仕事である限り、その仕事は完成するにちがいないという確信をもった。実際彼は仕事を立派にかつ遅滞なくやりとげた。しかしグローヴスはその仕事をどのようにしてやりとげたかは別の問題であった。彼は必要とあらば無神経で、無情であり暴君でかつ情容赦がなかった。ちょっとでもおくれたらがまんしないし、ぐちはきかないし、礼儀正しい形式をふむ余裕もなくまた考慮しなかった。グローヴスは仕事に対する自からの異常な包括力によって、すべての人間をその能力や感情を無視し、最高に生かして働かせた。自分自身も精力的に働いたことはもちろんである。

軍部の中のグローヴスの部下はいやおうなしに彼の強引な方針に従うばかりではなかった。しかし、建設会社や工業家でもグローヴスと仕事をいっしょにやるときは、まったく同じ方針に従わされたのである。しばしばグローヴスは民間のエンジニアに対して建設計画のための旅行に同行することをさそうことがあった。彼は午前1時のような法外な出発時間をえらんでエンジニアをろうばいさせるのであった。こんなときたとえば彼らは終着駅に午前6時にやっと到着し、つかれ果てた同行者がほとんど夜通し眠れなかったのでせめて上品な朝食にありつこうと望んでいると、グローヴスは完全に回復していてニコニコしながらいうのだった。「朝食をとる時間

がないね、さあ会議に出かけよう」

すじがき通りに、おくれずに仕事を完了できるという能力がみとめられ、グローヴスはペンタゴンの建設工事の指揮を命ぜられた。そしてトマス・ロビンソン將軍の代理として、グローヴスは国内のすべての軍事的建設事業を統轄した。たいていの場合、グローヴスの上官は彼がきらいだったが、その勇気と責任感はみとめざるをえなかった。

早川書房刊「マンハッタン計画」より

### 3-2 プロジェクト組織

#### 3-2-1 プロジェクト組織の基本

プロジェクト組織は、従来の組織と本質的に次の点において異なっている。

- 1) プロジェクト達成のための組織であること。
- 2) メンバーは、従来の組織に関係なく最適の担当者が割り当てられていること。
- 3) 期限を決められた臨時組織であること。(従って終われば解散するものであること)

の3つがあげられる。

- 1) プロジェクトとして選定されるものには、

工場建設

新製品開発、改良

プラント建設工事

工場倉庫適正配置

製品流通合理化

事務合理化推進

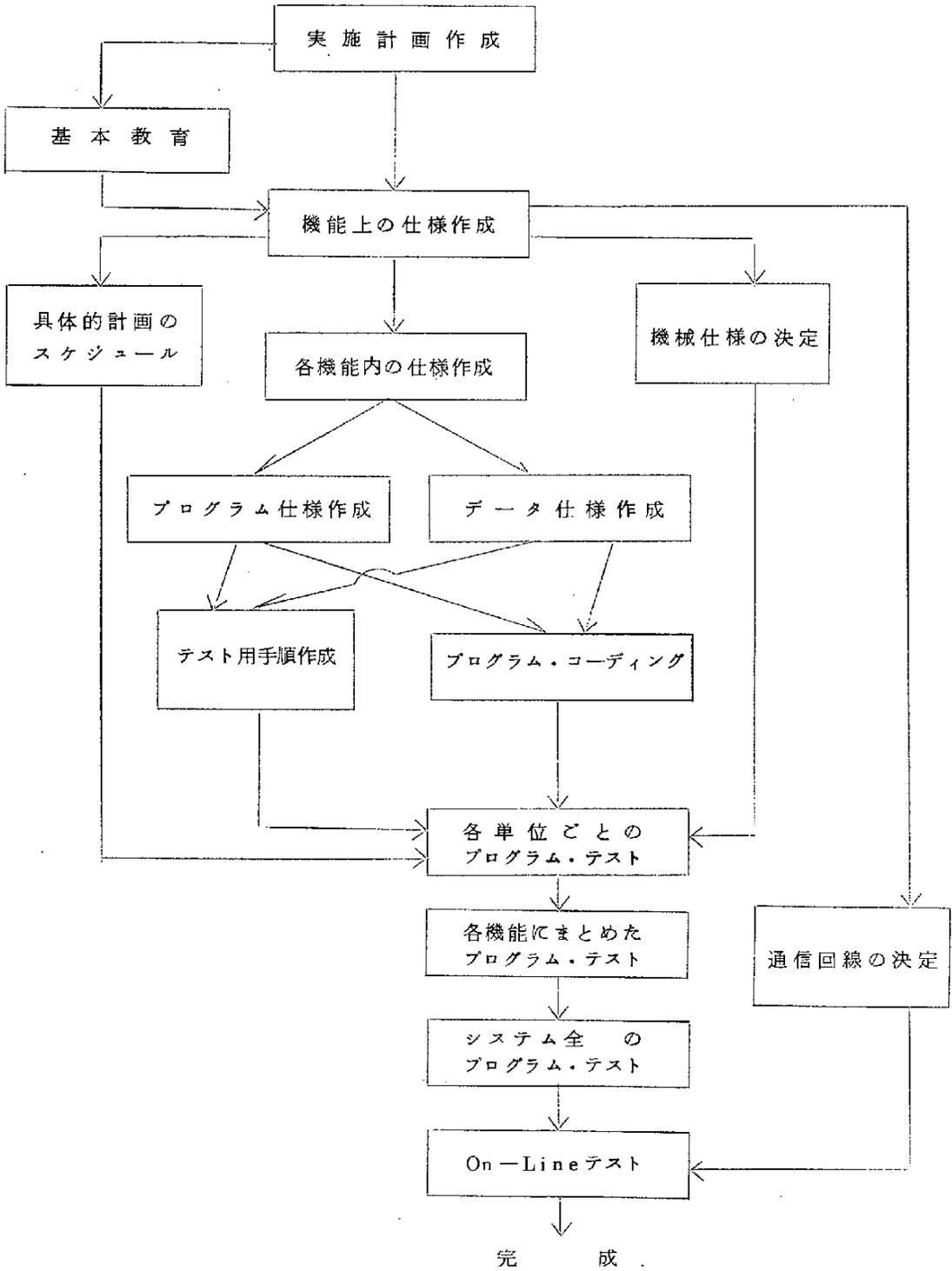
ZD推進

ソフトウェアの調査・研究・開発

コンサルティング活動

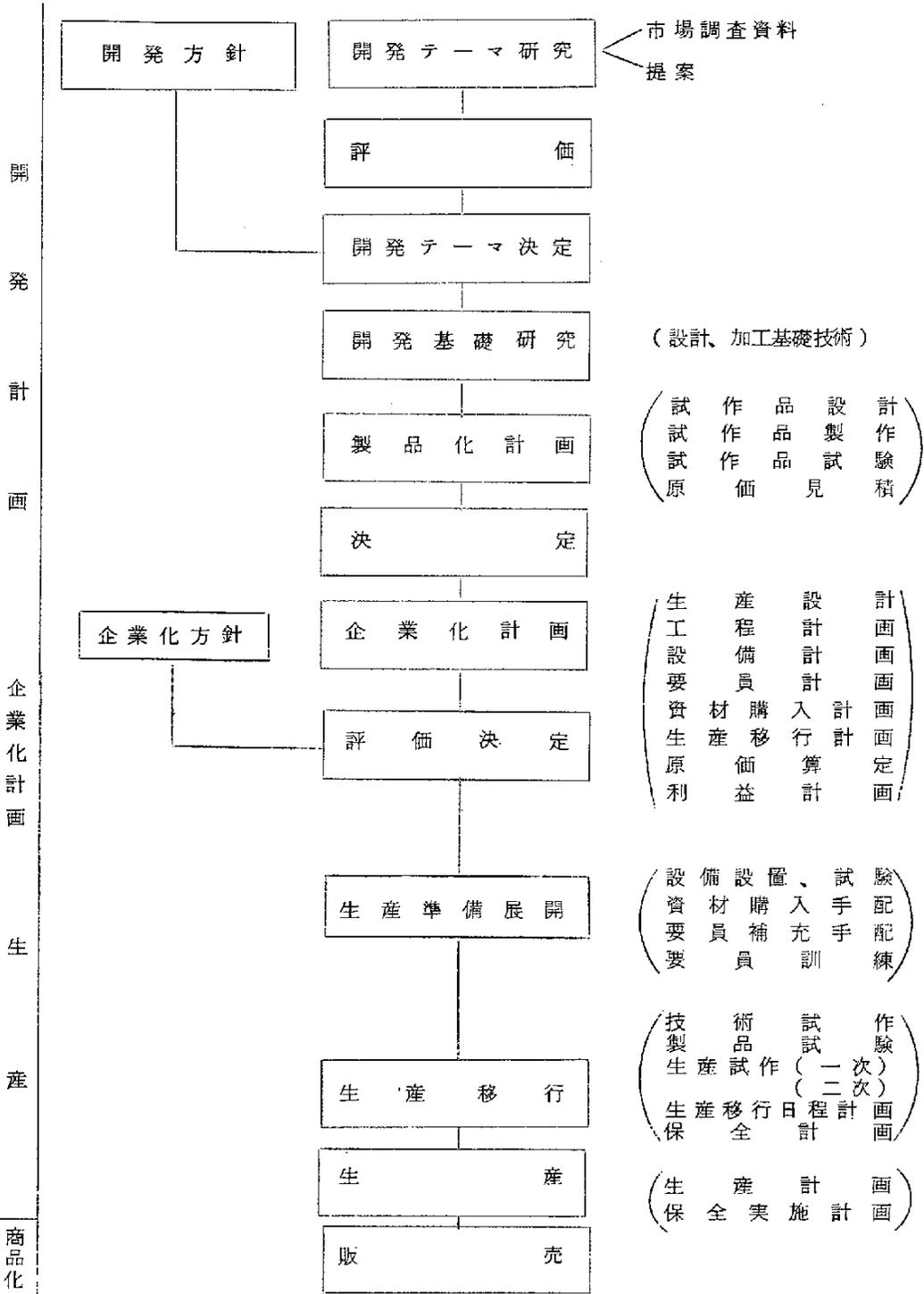
などが考えられる。これらのプロジェクトをプロジェクトとして担当する場合に、研究調査から実施まで担当するものもあれば、調査までの段階の場合もある。プロジェクトとして有名なマンハッタン計画は、原子爆弾を作るという決定がなされてから、プロジェクト・マネージャとしてグローヴ

図 I-7 導入までに必要な作業 (Activity)



\* データ・プロセッシング・マネジメント \* より

図 I - 8 新製品開発の手順



中島清一著「プラントエンジニアリング」より

図 I-9 自社開発および技術導入による新規事業企業化の手順

自社開発の場合		技術導入の場合		
企業化進展段階	処理・審査事項	企業化進展段階	処理・審査事項	
テーマ	アイデア収集	アイデア	アイデア収集	
	文献調査		基礎的研究	
	総合評価		予備的調査	
研	テーマの決定	調	コンタクト	
	基礎研究		技術的・経済的調査	
	応用研究		導入方針検討資料作成	
	研究結果報告書作成		第1次審査	
	技術的・経済的調査		導入方針決定	
	研究結果の評価		コンタクトの強化	
	パイロットプラント計画書作成		関連部門の調整	
究	パイロットプラント設置決定	導入交渉企画	最終の技術的・経済的調整 (中間報告)	
	パイロットプラント設計建設		企業化計画書1次案作成	
	パイロットプラント運転		第2次審査	
	パイロットプラント結果報告書作成		企業化提案	
	パイロットプラント結果評価		企業化方針決定	
	セミコマーシャルプラント計画書作成		技術導入契約締結	
	第2次審査		交渉結果報告書作成	
	セミコマーシャルプラント設置決定		企業化決定	申請
	セミコマーシャルプラント設計建設			
	セミコマーシャルプラント運転			
最終的調査調整				
セミコマーシャルプラント結果報告書作成				
セミコマーシャルプラントの結果の評価				
企業化計画書最終案作成				
企業化準備	第3次審査	申請	官庁申請書作成	
	企業化提案		官庁申請	
	企業化計画決定			
	担当職制決定			
	人事配置			
	企業化準備			
	設備投資計画書の作成			
建設工事	第4次審査(最終審査)			
	設備投資決定			
	建設月報			
	1次修正工事予算書(2次)			

中島清一著「プラントエンジニアリング」より

ス陸軍准将が選任されて、研究調査が開始された。そして、1945年8月6日、広島に投下されるまでがその期間であり、責任であった。

工場建設やプラント建設の場合は、一般に、(研究開発)→(仕様決定)→(詳細設計)→(建設)までがプロジェクトである。運転(その設備を使っての生産活動)や保全活動(その設備の保守)は別々の分担体制が組織化されている場合が多い。

新製品開発・改良やソフトウェア開発などは、(調査・研究)→(製品化研究)→(企業化研究)→(生産準備)などのステップが考えられ、それまでの作業が必要となる。

プロジェクトのパターン別にその手順を示すと、図I-7のようになる。それぞれ、テスト完了、生産体制移行、建設工事終了、工場引渡し、販売開始などがプロジェクトの終了という事になる。

国内におけるプロジェクトの例としては、鹿島建設の霞ヶ関ビルの建設、世界貿易センタービルの建設、電気化学工業の接着剤開発、関西電力の都市配電近代化、松下電器のホーム・ファックス開発、帝人の新蛋白食品の開発、日本石油の重油脱硫プロセスの開発などがある。

国外では、アメリカ陸軍のマンハッタン計画、NASAのアポロ計画、AEC社のLMFBR計画や高速増殖炉開発、テキサス・インスツルメント社の電子・光学機器システムの開発、ゼネラル・ダイナミックス社のATLAS開発などがある。ソフトウェアの開発としては、フォード社の排気ガス処理システム・交通輸送システム、ロッキード社のコンピュータによるNETWORKシステムなどがある。

## 2) メンバー構成

プロジェクトが決まって、プロジェクトの目標が明示されると、次に必要なのがメンバーの編成である。メンバーの編成はプロジェクトの成否を完全に支配する要因である。従来の機能別組織の場合でも組織構成員の能力は、その成果を左右する重要な要因であったが、プロジェクトの場合は期限が限られており、しかも少数精鋭の特別な専門的スキルにプロジェクトの成果が託されている。すなわち、従来からの組織では解決されない困難な課題であるだけに、一層高いレベルの能力に依存して、長期的構造的総合的ま

たは高額投資的競合的な問題を解決しようとするものである。こうした専門的能力が各参加メンバーに要求されるのである。これらの参加メンバーは、どのような条件を備えていなければならないであろうか。プロジェクトの内容や要求される質的レベルにもよって異なるが、一般には次のような資格要件が必要とされている。

1. 物事を客観的かつ公正に判断できる人物
2. チーム・ワークに徹した協力的な人物
3. 困難な課題に積極的に取り組むファイトのある人物
4. 特定の分野における専門的知識または技能を有する専門家

### 3-2-2 プロジェクト組織への移行プロセス

ダイナミック組織は、従来からの機能別組織に対応するものとして、ここ数年来急に注目されだした。企業環境が激しく変化している時に、企業目的を達成する方法である組織が固定的で、環境変化に対応できないとすれば、場合によっては企業競争にらくごする危険性は十分にありうる。

東商第173回常議員会議である「経営組織のあり方についての提言」では、次のようにいっている。

1. 企業が全社的に最高の効率をあげるためには、まずシステムの組織観の確立が必要である。
2. 企業が変化に即応して、ダイナミックな経営を展開するためには、機動的組織の導入が必要である。
3. 経営組織を変化に即応させるためには、経営者は既存の組織を絶えず再検討し、改善していく心構えが必要である。
4. 組織内のコミュニケーションを改善し、トップの意思決定を敏速、正確に行ない得る組織の簡素化と管理階層の短縮をはかる事が必要である。
5. 従業員の能力を開発、活用し、組織に活力を与える人事管理上の配慮が必要である。

また、「今後の経営組織はどうあるべきか」との設問に対しては、次のようにいっている。

「今後の組織変革に際して形態的な変化のみならず、あるいは形態的変化の導入を通して、こうした経営体質の変革へと導き出さなければならない事

が明らかであろう。上述の経営環境に対応して、かつここに述べた様な既存組織の問題点を克服して行くための具体的方策としては、

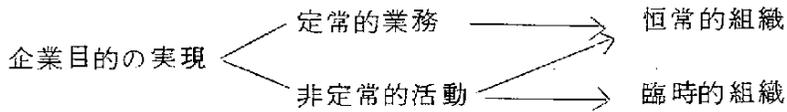
1. 課題中心組織（プロジェクト・チーム、タスク・フォース）を編成し、従来の職能部門の壁を越えた有機的、流動的活用を推進して行く。
2. 独立採算制、プロフィット・センター方式による管理（事業部制）を推進し、製造、営業、技術などの統合的な適用を図って行く。
3. 責任権限を明確化し、権限委譲を大幅に促進して行く。さらには課制廃止などの階層短縮にまで進む企業もある。
4. 経営情報システムの導入により、情報処理、伝達の総合性、連繋性を高めて行く。
5. 目標管理を推進し、各部門、各個人が企業の期待する目標をはっきりつかみ、しかもみずからの啓発目標、努力目標をも盛り込める様な体制に持って行く。
6. 経営企画表などのトップ・マネジメント補佐機関を設けまた将来の企業態様に対処し得る戦略的企画力を高めるなどの事があげられよう。」と述べられている。

従来の組織はそれなりのよい点もあった。しかしビジネスの組織においては、権限構造が人間能力の発揮の面、コミュニケーションにおいても、目的別の仕事を合理的にすすめる体制になっていない。すなわち、従来の機能別組織は次のような特徴を持っている。

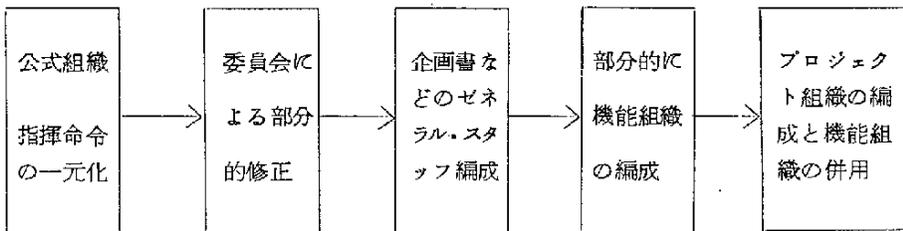
1. 組織構造上、いかなる構成員といえども、他の構成員すべてを知ることができない。
2. 組織の構成員は、その属する組織から得る収入で生計をたて、その組織内において昇進する事を第一の目的とする。彼は組織と密接不可分の関係を持ち、組織内の支配的な思想と対立するような言動はとらない。
3. 組織は階級的に構成されており、昇進は、組織内においてその人間に与えられる評価によって決まる。個人の目的は組織の目的に従属する。組織の構成員には、能力と忠誠と意欲が要求される。しかし、もちろん現実には、自己中心に働く者もいれば、忠誠心から働く者もいる。
4. 組織は、その組織が真に必要なかどうかという基本的な問題とは関係を

く、常に生存をし続けようとする。

企業の目的を達成するために、定常的な繰り返し作業の活動と、市場適応をねらう非定常的な活動が必要となる。そこでそれに対応して組織も成立することになる。すなわちつぎの様になる。



こうして組織は、従来からの公式的固定的な組織から徐々に非公式的弾力的なチーム・ワーク中心のプロジェクト組織に移行してきているのである。



## II プロジェクト・マネジメントの技法

### 1 PERTとCPM

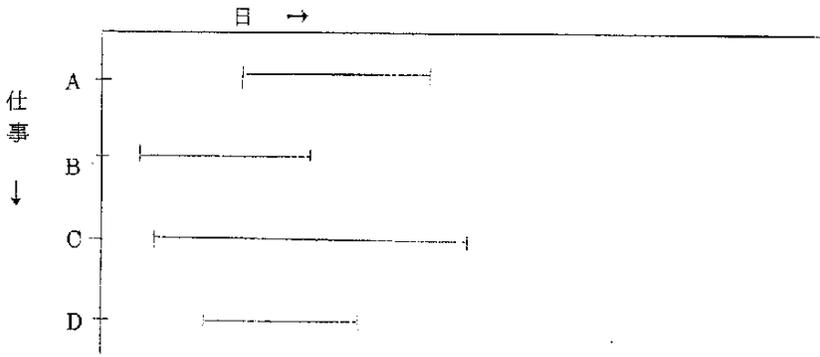
#### 1-1 PERT (Program Evaluation and Review Technique)

PERTは

- ・土木、建築等の工事日程の管理
- ・装置の新設、修理等の日程計画
- ・原料、部品等の搬入計画

等の問題を解決する手法として、開発されたものである。

簡単な日程管理手法としては、ガント・チャートがある。



これは、横軸のほうに月日、縦軸のほうに仕事をとりその仕事をA・B・C・Dという作業に分解し、Aは何日から始まって何日で終わらなければならぬか、Bはどうかといった具合になり、管理者・監督者がこれを使って仕事の進み具合を簡単に管理することができる。

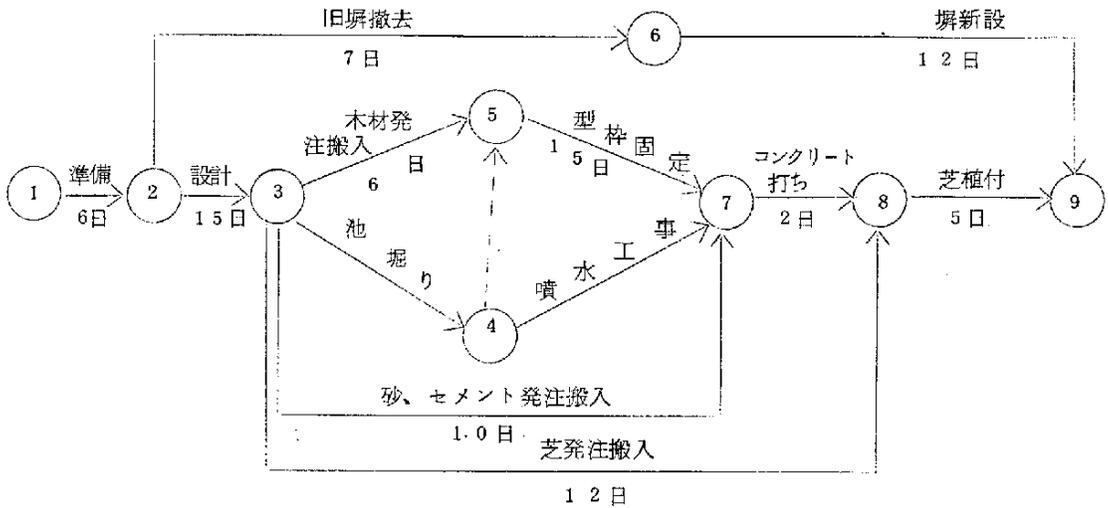
しかし、これには2つの欠点があり、

- (1) どの作業が終わらないとどの作業が始められないという作業間の前後関係が不明確
- (2) 特定の作業、たとえばCの完了が1日延びると他の作業にどのような影響を及ぼすかわからない。

ということである。この場合も作業数が少なければ問題とはならないが、作業数の多い仕事になると影響してくるわけで、その解決を図る手法として、

PERTが開発されたのである。

図 II - 1 ネット・ワーク



PERTは仕事を細かく分解し、図1のようにマルと矢印で網目状のネット・ワークを作成し、各作業の所要日数から、各作業の余裕日数、ネックになる作業の流れなどを見出し、重点的、総合的な日程管理を行なう手法である。

図II-1は、今まであった古いへいを取りこわして新しいへいを作り、そのときついでに庭に噴水のある池を作る仕事の場合のネット・ワークである。この仕事を行なう場合に、こまかく作業に分けて、それぞれの作業が何日ぐらいかかるかで見込みの数字を入れたのであるが、これを見るとガント・チャートの2つの欠点が解消されているのがわかる。

1番目は、仕事の前後関係がすぐわかることである。たとえば、②から⑥への矢印“古い塀を取りこわせ・7日間で”という作業は②から③への矢印とは別方向に出ているので、③への作業と同時点で行なってよいということをお知らせしているからである。この様子を図を見ることによって、どの作業が終わらないとどの作業が始められないかどうかということが、一目りょう然となる。

2番目は、どの作業が遅れたら他の作業にどの程度の影響を及ぼすかということであるが、足し算を行なうことによってはっきりとする。①がスタートで⑨が完成であるが、①から⑨までに到達する経路は何本もある。具体的に書くならば①→②→⑥→⑨、①→②→③→⑤→⑦→⑧→⑨、①→②→③→④→⑦→⑧→⑨、①→②→③→⑦→⑧→⑨そして①→②→③→⑧→⑨の5

本である。

次に、経路の日数をそれぞれ足してみると、①→②→③→⑤→⑦→⑧→⑨が最大49日となる。最大の経路のことを、PERTでは通常クリティカル・パスという。管理者・監督者の役目はこのクリティカル・パスの作業に目をひからせることである。

すなわち、①②③⑤⑦⑧⑨という経路の各作業は最大経路のため、どの作業でも1日遅れれば納期が1日遅れることである。だから、①②③⑤⑦⑧⑨という各作業に目をひからせることになる。では、他の作業はどうかということになる、引き算をすることにより他の作業の余裕が出てくることになる。その余裕のことをPERTではフロートと呼ぶ。たとえば、④から⑦にいく噴水工事の作業は、4日間という様に見積もったのであるが、後の方から逆に引き算をしていくと、11日の余裕があることがわかる。その場合、たとえば5人でこの作業を行なうとなっていれば、2人を配置転換して、クリティカル・パスの作業に割り当てるとか、資材や機械力を割り当てるとかのごとく、実際のアクションにつながるような情報が足し算するだけ、引き算するだけですぐわかるという利点がある。

ネット・ワークの例では、①から②という作業には6日かかるという具合に、1つの数字で見積っていたがこれを通常1点見積もりという。

それに対して3点見積りというのがある。ある作業を行なう場合に、普通の状態ならば10日間かかればできるはずであるが、非常に条件がよくうまく行けば8日間であり、何か悪条件があれば15日間かかるかも知れないというのが3点見積りである。この場合、8日間を楽観値、10日間を最可能値、15日間を悲観値という。

1点見積りの場合、見積る人に負担がかかる感じであるが、3点見積りの場合には、そのような精神的負担を軽くする利点もある。計算は少し複雑になるが、考え方は同じであり、3点見積りから1点見積りのような期待値を求めるなら以下の公式で計算される。

$$\text{期待値} = \frac{\text{楽観値} + \text{最可能値} \times 4 + \text{悲観値}}{6}$$

楽観値 = 8日 最可能値 = 10日 悲観値 = 15日 を考えてみると

$$\text{期待値} = \frac{8 + 10 \times 4 + 15}{6} = 10.5$$

となり、期待値は10.5日となる。

このPERTの親戚にCPM(Critical Path Method)というものがある。PERTは、何日かかる、何ヶ月かかるといった時間だけのパラメータを使ったが、CPMは、時間と費用という2つのパラメータを使用する。例えば、納期を2日短縮する為にはどの作業の所要日数を短縮することが最も費用を安く出来るかということを考える日程管理の手法である。

PERT、CPMの実施例としては土木、建築関係では日程管理のためよく使われているが、部品や材料の搬入計画にも適用されている。変わった例で、設計図を何日、誰に渡してやればよいか — 出国管理 — という問題にも使われている。

#### PERTの具体例

- 霞ヶ関ビルなどの高層建築の日程管理
- 建築材料の搬入計画

高層建築作成の場合、大都市においては材料を置いておくための広い場所がないので、どの材料をいつ持ってくるかを計画する。

- 図面管理

大きい機械製作の場合、多数ある設計図をいつの時点で誰に渡すかを管理する。それを出図管理ともいう。

#### — 2 CPM(Critical Path Method)

PERTは、パラメータ(要素)として何日、何ヶ月という様に時間だけの要素でできている。

それに対して、CPMは費用も考えるものである。たとえば1日遅れたら費用はどの程度かかり、1日作業を早く終わらせるためには費用がどの程度かかるかということ、時間と費用という2つの要素を同時に考えることである。当然、使い方はPERTより複雑である。たとえば、契約にあたって納期を約束した場合に、納期遅延ならば罰金を払うという契約がある。その際に、罰金を払って納期を遅らせたほうが得になるか、あるいは、罰金を払

われない様に何日か短縮すると費用はいくらかかるかという様に、考えてみる  
 ことである。その高度化したものとしても1つパラメータを増し、時間と  
 と費用のほかに資源を加えて考えるものがある。人間が何名、機械力はどの  
 くらい、ブルドーザは何台、トラクタは何台といった資源まで考えるもの  
 がある。それをランプス(RAMPS "Resource Allocation and  
 Multi-Project Scheduling")という。しかし、パラメータが増  
 すと、操作が非常に複雑になるので、ランプスが使われたという例は少ない。

## 2 線型計画法(LP, Linear Programming)

LPとは

- ・利益が最大になるような生産計画
- ・輸送コストを最小にするような輸送計画
- ・原料費を最小にするような原料配分計画

を解決するためによく使われている手法である。その意味は、量の間  
 の関係で取り扱われる数式が、すべて1次式の形、すなわち、直線である  
 ことに由来するものである。

簡単な例をあげて考えてみる。

"太郎さんと花子さんという恋人がいます。太郎さんは、毎月デート代として、  
 1,000円が自由になり、花さんはデートのために毎月20時間が自由  
 になります。デートする時に2人は喫茶店か、映画に行きます。喫茶店は1人  
 1回50円、映画は1人1回100円です。2人は取りとめなく話をするので、  
 喫茶店では毎回4時間話をします。映画は入れ替え制のために毎回3時間で追  
 い出されます。2人がデートするときの楽しみ度は、喫茶店の時は1.0、映画  
 の時は1.5です。

以上の条件のもとで、2人がデートした場合に、楽しみを最大にするにはど  
 うしたらよいでしょうか。"

	喫茶店	映画	使用可能量
太郎	100円	200円	1,000円
花子	4	3	20H
楽しみ	1.0	1.5	

〔解決法〕

喫茶店でデートする回数を $X_1$ 回、映画でデートする回数を $X_2$ 回とする。この $X_1$ 、 $X_2$ は未知数である。

- (1) お金について考えてみると、喫茶店に2人で行くと1回100円かかり、 $X_1$ 回行くから100 $X_1$ が喫茶店でデートする費用である。同様に、200 $X_2$ が映画の費用である。ただし、その合計は太郎が所持している1,000円の中から払うのであるから、1,000円に等しいか、小さくしなければならない。

$$100X_1 + 200X_2 \leq 1,000 \dots\dots (式1)$$

- (2) 時間について考えてみると、喫茶店に1回行くと4時間かかり、 $X_1$ 回行くから4 $X_1$ が喫茶店でデートする時間である。同様に、3 $X_2$ が映画の時間である。ただし、その合計は花子が自由になる20時間に等しいか、少しでなければならない。

$$4X_1 + 3X_2 \leq 20 \dots\dots (式2)$$

- (3) LPの特徴が、ここに出てくるわけであるが、 $X_1$ 、 $X_2$ という変数はマイナスになってはいけないことである。

$$X_1 \times X_2 \geq 0 \dots\dots (式3)$$

何故ならば、もし喫茶店に5回、映画にマイナス3回という答えの場合、実行することは明らかに不可能だからである。

普通、(式1)(式2)(式3)を条件式と呼ぶ。このような条件式のもとで目的は2人の楽しみを最大にすることである。喫茶店でデートした場合の楽しみの量が、1.0 $X_1$ であり、同様に映画では1.5 $X_2$ であり、これらを加えたものが最大(MAXIMUM)となるようにすることである。

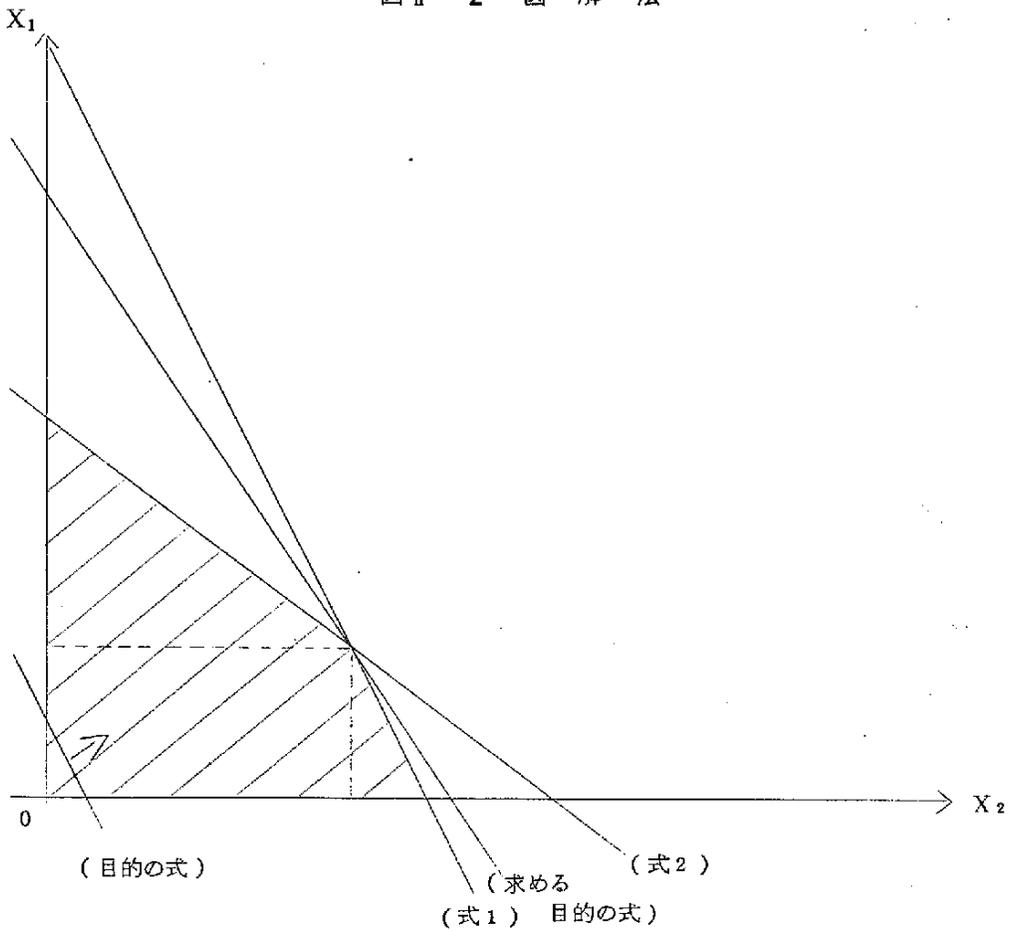
$$\text{目的 } 1.0X_1 + 1.5X_2 \rightarrow \text{Max}$$

これでLPのモデルができたわけである。一般に、与えられた条件が複雑かつ膨大であっても、以上の様な式構成になる。

この例の場合には、未知数が2つであるから、むずかしい計算を行わなくても、グラフを使うことにより、簡単に解くことができる。しかし、未知数が3つ以上になると簡単に書くわけには行かなくなる。

(式1)をグラフに書くと、図Ⅱ-2の様になり、1,000円に等しいか、

図Ⅱ-2 図解法



それ以下ということで、直線の下の部分になり、(式2)のグラフも同様に直線の下部分になる。また、(式3)の条件をも満足するために、該当する部分は斜線部分である。この範囲内に最大値があることが必要条件になる。

次に、目的の式を考えるのであるが、たとえば目的の式をイコール1と考える。 $(1.0 X_1 + 1.5 X_2 = 1)$  このグラフは上の様になる。最大値を求めたいのであるから、上へ平行移動して行き、(式1)(式2)の交点を通った時に最大になる。その際の値を読めば、 $X_1 = 2$ 、 $X_2 = 4$  ということであり、月に喫茶店に2回、映画を4回という具合にデートすれば、その時に限って2人の楽しみは最大になり、最大値は8で交わっているので、楽しみ最大値は8になる。実際の問題の場合は、未知数が50個~1,000個ぐらい出てくるが、アプリケーション・プログラムを使用すればある要素、数字を投入することによ

り、どういつ時に最大値がいくらであるという結果が計算されて、出てくる仕組みになっている。

#### LPの具体例

##### ○日本の専売公社の葉たばこの配合問題

数十種類の葉たばこをまぜあわせてたばこを作る場合に、ハイライト、ピースなど各種銘柄をおのおの何本かずつ作り、同時に、コストを最小にするまぜあわせ方は、どのようにすればよいかということである。

##### ○日本専売公社の葉たばこの輸送計画

どこの葉たばこの乾燥倉庫から、どこのたばこ製造工場へ輸送するのが輸送コストを最小にするかという輸送計画問題

##### ○原油の加工関係

工場での原油の処理能力は一定であり、石油化学工業は原油をもとにして、種々の製品を作る。それぞれの製品の価格がわかっている時に、どういつ製品をどのくらい作れば、きょう現在の各製品の価格で利益が最大になるかを計算し、生産計画を立てる。

##### ○京都市立病院の給食の献立

##### ○私鉄の運賃値上げ

##### ○工業立地問題

### 3 シミュレーション (Simulation)

シミュレーションとは

##### ○サービス要因問題

##### ○長期収支計画

##### ○窓口増設問題

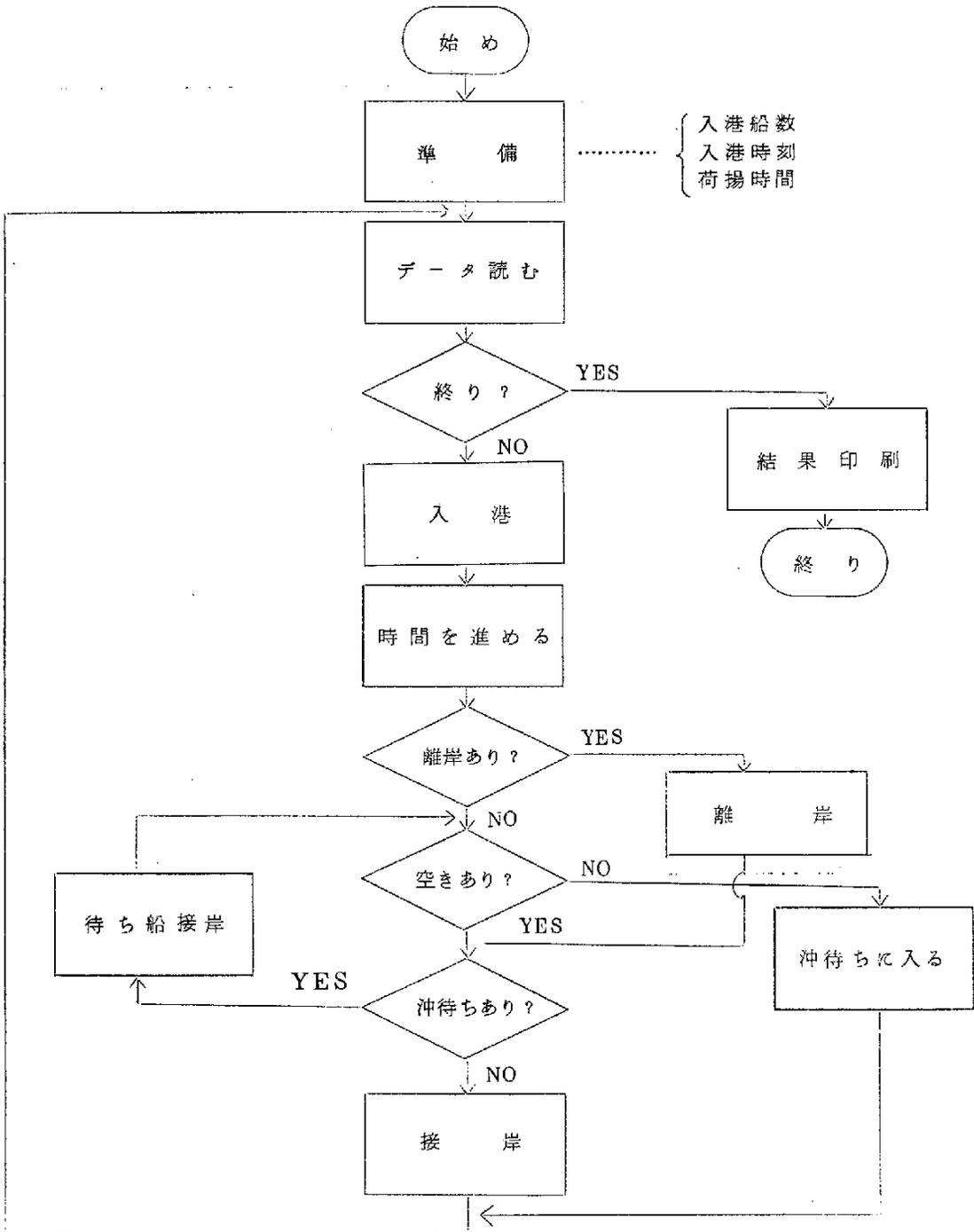
等について、現実のものではなく、その模型(モデル)を構成してある現象を模倣する模擬実験のことである。

シミュレーションは、金がかかりすぎるとか、危険をとまなうとか、時間がかかりすぎるとか、解析的に解く方法が見つからないというような問題を解決するためよく使われる手法である。

例をあげて考えてみる。

次ページの流れ図は岸壁の長さの決定を目的としている。これは、実際に北海道大学で、八戸漁港の棧橋延長問題で行なったいわゆる待ち合わせの問題である。その当時、八戸漁港棧橋は150隻分の岸壁しかなかったために、朝方になると操業が終った船で港は混雑し、荷揚げすることも出来ずに沖で待っている船がたくさんあった。そこで棧橋の長さを延ばせばよいわけであるが、むやみやたらに延ばせば、当然費用の問題がからんでくる訳で、そのためにモデルを構成して、長さと費用を最も効率的に生かせるように実験するのである。

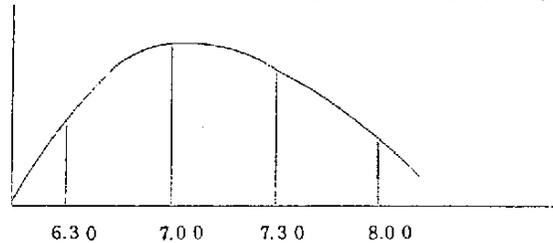
図 II - 3 岸壁の長さ決定の流れ図



シミュレーションを行なう場合には、まずデータの整理を行なう。

- (1) 今までに、何隻ぐらい毎日入港していたか(入港船数)の調査
  - (2) 入港船数が平均300隻だとわかっていても、朝から晩まで平均して入ってくるのであれば問題ないが、ラッシュの時間があるはずなので入港の時刻の調査
  - (3) 入港した船は荷揚げをするのであるから、荷揚げにかかる時間調査
- つぎに、八戸漁港の現状に即した模擬実験を行なうため、上記のデータを使って、まず今日何隻入港したかを乱数をもちいてきめる。

そして入港時刻も下図のようなデータをもとに乱数を使って、それぞれの船



に対して、入港時刻を割り当てる。

荷揚げについても同様に所要時間を決める作業を行なうのである。

以上でシミュレーション用のデータの準備が出来たので、シミュレーションを開始する。ここでは、簡単に流れ図の読み方を説明する。

- ① 第1番目のデータを読む。このデータには何隻目であるかの番号、入港時刻、荷揚げ時間が入っている。
- ② 何隻目の船であるかのチェックをする。データが終りという合図がなければ、NOの矢印の方に進む。
- ③ 読み込んだデータを入港の状態にセットし、時間を進める。
- ④ 第1番目の入港であるから、今から荷揚げをすることになり、難岸はNOであるので、下の矢印に進む。
- ⑤ 同じく、第1番目であるから、岸壁の空きは当然ありYesの矢印に進む。
- ⑥ 沖に待っている船があるか、これについても第1番目であるからNOの矢印に進み、接岸する。

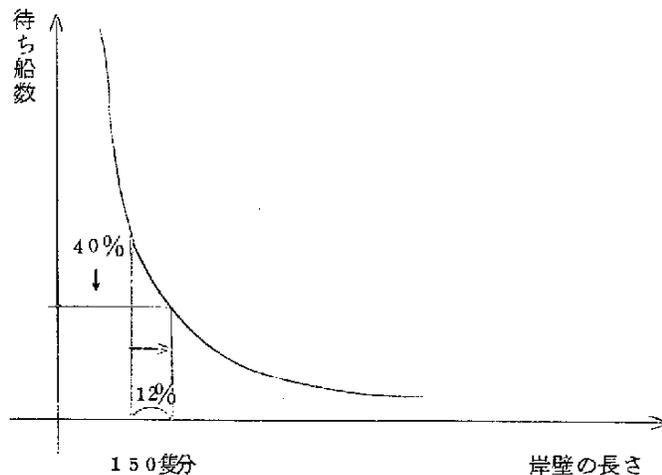
という様になる。これをコンピュータで行なわせるのである。

たとえば、1日分として今日は300隻入港したということであれば、300隻分が終ると1日分の模擬実験が終ったことになるのである。以上の模擬実験を何回もくり返して、1ヶ月分とか2ヶ月分を計算すると同時に、1割岸壁を延長したら沖で待っている船はどのくらい減ったか、1割5分延長したらどうかというデータをとるわけである。

すでに学んだLP、PERTとの大きな違いがここにあるわけでLPの場合だところすればよいという結果がすぐ出てくるが、シミュレーションの場合は、確率的な問題として精度を上げるために、何度も乱数を使って実験をくり返すのである。

実際に八戸漁港で行なったシミュレーションの結果は、横軸に岸壁の長さ、

表Ⅱ-1 シミュレーション結果



縦軸に沖で待っている船の数をとってグラフに描くならば、表Ⅱ-1の様になったのである。上図でわかる様に、岸壁の長さを12% 延ばすことにより、待ち船の数が40% 減るということである。

次に、シミュレーションの精度ということであるが、確率的に問題を解く以上、模擬実験の回数が多ければ多いほどよいということは明らかである。しかし、実験費用がそれにともなって、多くかかることも明らかであるので、一般的には、誤差を $\frac{1}{2}$ 、精度を2倍にする場合は、4倍の回数行ない、誤差を $\frac{1}{10}$ 、精度を10倍にする場合は100倍の回数行なうわけである。

#### シミュレーションの具体例

##### ○東京オリンピックの際のメイン・スタジアムの入口の数

開会式等の入場者を行列させることなく、管理、費用を最小にするためである。

##### ○需要予測

##### ○在庫管理

##### ○国鉄の通勤輸送問題

##### ○日本航空の就航計画

##### ○道路の信号制御方式

#### 4 インダストリアル・ダイナミックス

ジェイ・フォレスター (Jay W. Forrester) によるインダストリアル・ダイナミックス (Industrial Dynamics) では企業モデルのシステム要素を受注、人、物、金、設備および情報の六要素に分け、それらをフローとして把握することにより、意思決定ポリシーの変化がどのように企業システムに影響を与えるかを分析し、よりよい企業システムのデザインを行なわんとするものである。

コンピュータ・モデルには、オペレーション・モデルと、デザイン・モデルとがあり、ポニーニヤサイヤート・マーチのモデルがオペレーション・モデルであるのに対して、インダストリアル・ダイナミックスは解析にも利用できるが、モデル・デザインのほりに有利なダイナモ (Dynamo) というコンパイラ (compiler 人間語に近い文章表現でコンピュータのプログラムが書けるように配慮した一種の文法) が用意されていて、システム・デザインにもきわめて有利である。

インダストリアル・ダイナミックスも前二者と同様に、コンピュータ・シミュレーション・モデルであるが、世の中のあらゆるシステムがインフォメーション・フィードバックを受けるとの仮設が特徴で、これに発散型の正のフィードバックと、収斂型の負のフィードバックとが考えられている。そして、インフォメーション・フィードバック・システムには、時間遅れと増幅という、二

つの現象を伴うものであるという、現実分析の結果が取り入れられている。

インダストリアル・ダイナミクス・モデルの実例は企業の秘密性に災いされて、あまり紹介されていないのであるが、米国ではかなり実用化されておると言われており、わが国でもいろいろと研究されているようである。しかし何分にも、フォレスター教授の書物が名著でもあり、邦訳もなく、ダイナモのマニュアルもマスターしなくてはならぬので、やや普及に難色があるが、モデル化には有力な武器となろう。

## 5 LPの実例

### (1) 問題

独立した3つの生産過程に関連して、千葉工場に新しく年産6,000トン  
の中間製品（B製品）の工場を建設した時の生産、販売および輸送（社内  
の他工場送りを含む）を企業全体として最適化（製造原価および輸送費の  
和を最小化）になるように、各工場別の生産量、需要地区別販売量および各  
製品の工場間の転送量を、線型計画法（Linear Programming）で  
一義的に決定することを目的とする。

### (2) 問題解決に必要なデータ

① 独立した3つの製品をA、B、Cとし、B製品はA製品の0.53原単位  
で製造され、C製品の種類はかなりあるので、原単位的に把握しないで、  
工場別需要量として把握することにした。

② A・B製品の工場別設備能力および製造原価中の変動費部分は下記のと

A製品の工場別設備能力およびコスト

項目 \ 工場	千葉第1	千葉第2	神奈川	山口第1	山口第2	九州
設備能力 (トン/年)	15,800	15,800	8,400	7,500	1,100	6,600
変動費 (円/トン)	13,380	19,380	38,000	28,810	38,000	144

おりと仮定する。

千葉第1、千葉第2、たんに千葉工場および山口第1、山口第  
2、たんに山口工場と呼称しても、それぞれ、千葉、山口の同一地域にあ  
る工場を意味し、A製品に関して、製造コストの相違で、第1、第2と区  
別したのにすぎない。

B製品の工場別設備能力およびコスト

項目 \ 工場	千葉建設予定	神奈川	山口
設備能力 (トン/年)	6,000	29,600	26,500
変動費 (円/トン)	2,260	720	370

③ A製品およびB製品の地区別需要量(トン/年)

製品名 \ 販売地域	北海道	関東	関西	九州
A製品	408	13,332	5,796	4,296
B製品	-	8,880	10,380	1,380

④ C製品用の自社必要量(必要転送量)(トン/年)

千葉工場 必要量	神奈川工場 必要量	新潟工場 必要量	山口工場 必要量	九州工場 必要量
2,400	14,965	1,000	9,600	2,100

⑤ A製品の工場別・地区別運賃表(円/トン)

出荷元 \ 仕向地	北海道	関東	関西	九州	転送		
					千葉工場	神奈川工場	山口工場
千葉工場	5,450	1,700	4,630	4,950	-	1,000	3,700
山口工場	7,850	4,850	3,940	780	3,700	3,300	-
九州工場	8,370	5,500	4,450	700	4,700	4,300	1,000

⑥ B製品の工場別・地区別運賃表(円/トン)

出荷元 \ 仕向先	関東	関西	九州	転送				
				千葉工場	神奈川工場	山口工場	九州工場	新潟工場
千葉建設 予定工場	2,100	4,160	5,140	-	900	3,300	4,200	1,800
神奈川工場	1,860	3,620	4,720	900	-	3,000	3,870	1,500
山口工場	4,860	3,980	1,040	3,300	3,000	-	900	4,000

⑦ 以上の前提条件のうえに、問題を数式化するための未知数を設定する。

その場合、ペースを合わせるため、B製品は0.53でA製品に換算して数量をあらわすものとする。

(1) 未知数の設定

① A製品の販売量をあらわす未知数(トン/年)

A製品出荷元 \ 販売地区	北海道	関東	関西	九州
千葉第1工場	$x_{100}$	$x_{111}$	$x_{112}$	$x_{113}$
千葉第2工場	$x_{110}$	$x_{114}$	$x_{115}$	$x_{116}$
山口第1工場	$x_{120}$	$x_{117}$	$x_{118}$	$x_{119}$
山口第2工場	$x_{130}$	$x_{121}$	$x_{122}$	$x_{123}$
九州工場	$x_{140}$	$x_{124}$	$x_{125}$	$x_{126}$
計	408	13,332	5,796	4,296

神奈川工場のA製品については、荷役設備の不良などの理由により、外販および他工場送りを考慮しないことにした。

② B製品のA製品製造工場別・地区別販売量をあらわす未知数(トン/年)

A製品製造工場名 \ B製品製造工場名	千葉建設予定工場			神奈川工場			山口工場		
	関東	関西	九州	関東	関西	九州	関東	関西	九州
千葉第1工場	$x_{131}$	$x_{132}$	$x_{133}$	$x_{134}$	$x_{135}$	$x_{136}$	$x_{141}$	$x_{142}$	$x_{143}$
千葉第2工場	$x_{147}$	$x_{148}$	$x_{149}$	$x_{151}$	$x_{152}$	$x_{153}$	$x_{157}$	$x_{158}$	$x_{159}$
神奈川工場	—	—	—	$x_{164}$	$x_{165}$	$x_{166}$	—	—	—
山口第1工場	$x_{171}$	$x_{172}$	$x_{173}$	$x_{174}$	$x_{175}$	$x_{176}$	$x_{181}$	$x_{182}$	$x_{183}$
山口第2工場	$x_{187}$	$x_{188}$	$x_{189}$	$x_{191}$	$x_{192}$	$x_{193}$	$x_{197}$	$x_{198}$	$x_{199}$
九州工場	$x_{204}$	$x_{205}$	$x_{206}$	$x_{207}$	$x_{208}$	$x_{209}$	$x_{214}$	$x_{215}$	$x_{216}$
計	関東地区 = 8,880 関西 = 10,380 九州 = 1,380								

たとえば、 $x_{131}$ トンは、千葉第1工場で生産されたA製品のうち千葉建設予定のB製品工場でB製品の原料として消費され、B製品として関東地区に輸送すべきものをA製品の量で示したものである。したがって、B製品の量で示すならば、 $x_{131} \times \frac{1}{0.53}$ トンとなる。8,880トン、10,380トンおよび1,380トンはB製品としての数量。

③ B製品のA製品製造工場別・神奈川工場向け転送量をあらわす未知数

(トン/年)

B製品の加工 A製品の 製造工場名	千葉建設予定工場	神奈川工場	山口工場
千葉第1工場	$x_{221}$	$x_{222}$	$x_{224}$
千葉第2工場	$x_{226}$	$x_{227}$	$x_{229}$
神奈川工場	—	$x_{231}$	—
山口第1工場	$x_{233}$	$x_{234}$	$x_{236}$
山口第2工場	$x_{235}$	$x_{239}$	$x_{241}$
九州工場	$x_{243}$	$x_{244}$	$x_{246}$
計	全体として14,965トンの要転送量		

たとえば、 $x_{233}$  トンは、山口第1工場で生産されたA製品のうち千葉建設予定工場に送られてB製品の原料として消費され、さらに、神奈川工場におけるB製品のdown flow用に転送・消費さるべきものをA製品の量で示したものである。したがって、B製品の量で示すならば $x_{234} \frac{1}{0.53}$  トンとなる。  
14,965トンはB製品としての数量(以下考え方は同じ)。

④ B製品のA製品製造工場別・千葉工場向け転送量をあらわす未知数

(トン/年)

B製品の加工 A製品の 製造工場名	千葉建設予定工場	神奈川工場	山口工場
千葉第1工場	$x_{248}$	$x_{249}$	$x_{251}$
千葉第2工場	$x_{253}$	$x_{254}$	$x_{256}$
神奈川工場	—	$x_{259}$	—
山口第1工場	$x_{263}$	$x_{264}$	$x_{266}$
山口第2工場	$x_{268}$	$x_{269}$	$x_{271}$
九州工場	$x_{273}$	$x_{274}$	$x_{276}$
計	全体として2,400トンの要転送量		

⑤ B製品のA製品工場別・九州工場向け転送量をあらわす未知数(トン/年)

B製品の加工工場名 A製品製造工場名	千葉建設予定工場	神奈川工場	山口工場
千葉第1工場	x 278	x 279	x 281
千葉第2工場	x 283	x 284	x 286
神奈川工場	—	x 289	—
山口第1工場	x 293	x 294	x 296
山口第2工場	x 298	x 299	x 301
九州工場	x 303	x 304	x 306
計	全体として2,100トンの要転送量		

⑥ B製品のA製品工場別・新潟工場向け転送量をあらわす未知数(トン/年)

B製品の加工工場名 A製品製造工場名	千葉建設予定工場	神奈川工場	山口工場
千葉第1工場	x 308	x 309	x 311
千葉第2工場	x 313	x 314	x 316
神奈川工場		x 318	—
山口第1工場	x 320	x 321	x 323
山口第2工場	x 325	x 326	x 328
九州工場	x 330	x 331	x 333
計	全体として1,000トンの要転送量		

⑦ B製品のA製品製造工場別・山口工場向け転送量をあらわす未知数(トン/年)

A製品製造工場名 \ B製品の加工工場名	千葉建設予定工場	神奈川工場	山口工場
千葉第1工場	$x_{335}$	$x_{336}$	$x_{338}$
千葉第2工場	$x_{340}$	$x_{341}$	$x_{343}$
神奈川工場	—	$x_{346}$	—
山口第1工場	$x_{350}$	$x_{351}$	$x_{353}$
山口第2工場	$x_{355}$	$x_{356}$	$x_{358}$
九州工場	$x_{360}$	$x_{361}$	$x_{363}$
計	全体として9,600トンの要転送量		

(4) 数式化

(3)で設定した未知数と(2)で示されているデータにより、生産能力、地区別需要量および自社必要量につき、つぎの21個の制限条件を設定する。

① 千葉第1工場のA製品製造設備の能力制限

$$\begin{aligned}
 &x_{100} + x_{111} + x_{112} + x_{113} + x_{131} + x_{132} + x_{133} + x_{134} + x_{135} + x_{136} \\
 &+ x_{141} + x_{142} + x_{143} + x_{221} + x_{222} + x_{224} + x_{246} + x_{248} + x_{251} + x_{276} \\
 &+ x_{279} + x_{281} + x_{308} + x_{309} + x_{311} + x_{335} + x_{336} + x_{338} \leq 1,5800
 \end{aligned}$$

(以下の制限式は $x_{101}$ は101と示す)

② 千葉第2工場のA製品製造設備の能力制限

$$\begin{aligned}
 &110 + 114 + 115 + 116 + 147 + 148 + 149 + 151 + 152 + 153 + 157 + 158 \\
 &+ 159 + 226 + 227 + 229 + 253 + 254 + 256 + 283 + 284 + 286 + 313 + 314 \\
 &+ 316 + 340 + 341 + 343 \leq 15,800
 \end{aligned}$$

③ 神奈川工場のA製品製造設備の能力制限

$$164 + 165 + 166 + 231 + 259 + 289 + 318 + 346 \leq 8,400$$

④ 山口第1工場のA製品製造設備の能力制限

$$\begin{aligned}
 &120 + 117 + 118 + 119 + 171 + 172 + 173 + 174 + 175 + 176 + 181 + 182 \\
 &+ 183 + 233 + 234 + 236 + 263 + 264 + 266 + 293 + 294 + 296 + 320 + 321 \\
 &+ 323 + 350 + 351 + 353 \leq 7,500
 \end{aligned}$$

- ⑤ 山口第2工場のA製品製造設備の能力制限

$$130+121+122+123+187+188+189+191+192+193+197+198 \\ +199+238+239+241+268+269+271+298+299+301+325+326 \\ +328+355+356+358 \leq 1,100$$

- ⑥ 九州工場のA製品製造設備の能力制限

$$140+124+125+126+204+205+206+207+208+209+214+215 \\ +216+243+244+246+273+274+276+303+304+306+330+331 \\ +333+360+361+363 \leq 6,600$$

- ⑦ 千葉建設予定工場のB製品製造設備の能力制限

$$\frac{1}{0.53} (131+132+133+147+148+149+171+172+173+187+188 \\ +189+204+205+206+221+226+233+238+243+248+253+263 \\ +268+273+278+283+293+298+303+308+313+320+325+330 \\ +335+340+350+355+360.) \leq 6,000$$

- ⑧ 神奈川工場のB製品製造設備の能力制限

$$\frac{1}{0.53} (134+135+136+151+152+153+164+165+166+174+175 \\ +176+191+192+193+207+208+209+222+227+231+234+239 \\ +244+249+254+259+264+269+274+279+284+289+294+299 \\ +304+309+314+318+321+326+331+336+341+346+351+356 \\ +361.) \leq 29,600$$

- ⑨ 山口工場のB製品製造設備の能力制限

$$\frac{1}{0.53} (141+142+143+157+158+159+181+182+183+197+198 \\ +199+214+215+216+224+229+236+241+246+251+256+266 \\ +271+276+281+286+296+301+306+311+316+323+328+333 \\ +338+343+353+358+363.) \leq 26,500$$

- ⑩ 神奈川工場のB製品自家消費必要量の条件

$$\frac{1}{0.53} (221+222+224+226+227+229+231+233+234+236+238 \\ +239+241+243+244+246) = 1,496.5$$

- ⑪ 千葉工場のB製品自家消費必要量の条件

$$\frac{1}{0.53} (248+249+251+253+254+256+259+263+264+266+268 \\ +269+271+273+274+276) = 2,400$$

- ⑫ 山口工場のB製品自家消費必要量の条件

$$\frac{1}{0.53} (335+336+338+340+341+343+346+350+351+353+355 \\ +356+358+360+361+363)=9,600$$

- ⑬ 関東地区A製品需要量の制限

$$111+114+117+121+124=1,332$$

- ⑭ 関西地区A製品需要量の制限

$$112+115+118+122+125=5,796$$

- ⑮ 九州地区A製品需要量の制限

$$113+116+119+123+126=4,296$$

- ⑯ 北海道地区A製品需要量の制限

$$100+110+120+130+140=408$$

- ⑰ 関東地区B製品需要量の制限

$$\frac{1}{0.53} (131+134+141+147+151+157+164+171+174+181+187+191 \\ +197+204+207+214)=8,880$$

- ⑱ 関西地区B製品需要量の制限

$$\frac{1}{0.53} (132+135+142+148+152+158+165+172+175+182+188 \\ +192+198+205+208+215)=10,380$$

- ⑲ 九州地区B製品需要量の制限

$$\frac{1}{0.53} (133+136+143+149+153+159+166+173+176+183+189 \\ +193+199+206+209+216)=1,380$$

- ⑳ 九州工場B製品自家消費必要量の条件

$$\frac{1}{0.53} (278+279+281+283+284+286+289+293+294+296+298 \\ +299+301+303+304+306)=2,100$$

- ㉑ 新潟工場B製品自家消費必要量の条件

$$\frac{1}{0.53} (308+309+311+313+314+316+318+320+321+323+325 \\ +326+328+330+331+333)=1,000$$

- ㉒ 目的関数の設定

以上の制限条件式に含まれる各未知数にかかわる係数(製造原価中変動費+輸送費)を(2)のデータから算出して、下記の目的関数を作る。

$$\begin{aligned}
& \text{目的関数 } Z = 1,8330x_{100} + 2,4830x_{110} + 1,5080x_{111} + 1,8010x_{112} \\
& + 1,8330x_{113} + 2,1080x_{114} + 2,4010x_{115} + 2,4330x_{116} + 3,3660x_{117} \\
& + 3,2750x_{118} + 2,9590x_{119} + 3,6600x_{120} + 4,2850x_{121} + 4,1940x_{122} \\
& + 3,8780x_{123} + 5,644x_{124} + 4,594x_{125} + 8,44x_{126} + 4,5850x_{130} \\
& + 2,1608x_{131} + 2,5495x_{132} + 2,7343x_{133} + 1,9,249x_{134} + 2,2570x_{135} \\
& + 2,4646x_{136} + 8,514x_{140} + 2,6949x_{141} + 2,5288x_{142} + 1,9,710x_{143} \\
& + 2,7608x_{147} + 3,1495x_{148} + 3,3343x_{149} + 2,5,249x_{151} + 2,8,570x_{152} \\
& + 3,0,646x_{153} + 3,2,949x_{157} + 3,1,288x_{158} + 2,5,740x_{159} + 4,2,869x_{164} \\
& + 4,6,190x_{165} + 4,8,266x_{166} + 4,0,738x_{171} + 4,4,625x_{172} + 4,6,473x_{173} \\
& + 3,6,979x_{174} + 4,0,300x_{175} + 4,2,376x_{176} + 3,8,679x_{181} + 3,7,081x_{182} \\
& + 3,1,470x_{183} + 4,9,928x_{187} + 5,3,815x_{188} + 5,5,663x_{189} + 4,6,169x_{191} \\
& + 4,9,490x_{192} + 5,1,566x_{193} + 4,7,869x_{197} + 46,208x_{198} + 4,0,660x_{199} \\
& + 1,3,072x_{204} + 1,6,959x_{205} + 1,8,807x_{206} + 9,3,13x_{207} + 1,2,634x_{208} \\
& + 1,4,710x_{209} + 1,1,013x_{214} + 9,3,52x_{215} + 3,8,04x_{216} + 1,9,343x_{221} \\
& + 1,5,739x_{222} + 2,3,439x_{224} + 2,5,343x_{226} + 2,1,739x_{227} + 2,9,439x_{229} \\
& + 3,9,359x_{231} + 3,8,473x_{233} + 3,3,469x_{234} + 3,5,169x_{236} + 4,7,663x_{238} \\
& + 4,2,659x_{239} + 4,4,359x_{241} + 1,0,807x_{243} + 5,8,03x_{244} + 7,5,03x_{246} \\
& + 1,7,645x_{248} + 1,7,437x_{249} + 2,4,005x_{251} + 2,3,645x_{253} + 2,3,437x_{254} \\
& + 3,0,000x_{256} + 4,1,057x_{259} + 3,6,775x_{263} + 3,5,167x_{264} + 3,5,735x_{266} \\
& + 4,5,965x_{268} + 4,4,357x_{269} + 4,4,925x_{271} + 9,1,09x_{273} + 7,5,01x_{274} \\
& + 8,0,69x_{276} + 2,5,570x_{278} + 2,3,041x_{279} + 1,9,476x_{281} + 3,1,570x_{283} \\
& + 2,9,041x_{284} + 2,5,476x_{286} + 4,6,661x_{289} + 4,4,700x_{293} + 4,0,771x_{294} \\
& + 3,1,206x_{296} + 5,3,890x_{298} + 4,9,961x_{299} + 4,0,396x_{301} + 1,7,034x_{303} \\
& + 1,3,105x_{304} + 3,5,40x_{306} + 2,1,041x_{308} + 1,8,569x_{309} + 2,5,326x_{311} \\
& + 2,7,041x_{313} + 2,4,569x_{314} + 3,1,326x_{316} + 4,2,189x_{318} + 4,0,171x_{320} \\
& + 3,6,299x_{321} + 3,7,056x_{323} + 4,9,361x_{325} + 4,5,489x_{326} + 4,6,246x_{328} \\
& + 1,2,505x_{330} + 8,6,33x_{331} + 9,3,90x_{333} + 2,3,872x_{335} + 2,1,400x_{336} \\
& + 1,7,778x_{338} + 2,9,872x_{340} + 2,7,400x_{341} + 2,3,748x_{343} + 4,5,020x_{346}
\end{aligned}$$

\*  $x_{138}$  の係数 ¥ 2,464.6 はつぎのように算定される。

千葉第1工場のA製品→神奈川工場にてB製品に加工→九州地区に販売 = [千葉第1工場A製品製造原価(②の②表) + 千葉工場から神奈川工場へのA製品運賃(②の⑤表)] + [神奈川工場におけるB製品の加工賃(②の②表) + 神奈川工場から九州地区までのB製品の運賃(②の⑥表)] ÷ 原単位  
 = [ ¥ 1,338.0 + ¥ 1,000.0 ] + [ ¥ 720 + ¥ 4,720 ] ÷ 0.53 = ¥ 2,464.6

+ 4,300.2  $x_{350}$  + 3,913.0  $x_{351}$  + 2,950.8  $x_{353}$  + 5,219.2  $x_{355}$  + 4,832.0  $x_{358}$

+ 3,869.8  $x_{358}$  + 1,533.6  $x_{360}$  + 1,146.4  $x_{361}$  + 1,842.2  $x_{363}$  固定費 → Minimum

最適生産・販売・輸送量(トン/年)

製品	設備区分	能力 限度	最適 生産	販 売 量				自家消費 輸送量*	idle capacity
				関東	関西	九州	北海道		
A 製 品	千葉第1工場	15,800	15,800	6,992	5,796		408	2,604	—
	千葉第2工場	15,800	15,800	6,340				9,460	—
	神奈川工場	8,400	3,904	/	/	/		3,904	4,496
	山口第1工場	7,500	7,500					7,500	—
	山口第2工場	1,100	1,100					1,100	—
	九州工場	6,600	6,600			4,296		2,304	—
	計	55,200	50,704	13,332	5,796	4,296	408	26,872	4,496
B 製 品	千葉建設予定工場	6,000	528				/	528	5,472
	神奈川工場	29,600	29,600	8,880	2,883		/	17,837	—
	山口工場	26,500	20,577		7,497	1,380	/	11,700	5,923
	計	62,100	50,705	8,880	10,380	1,380	/	30,065	11,395

\* はつぎの表およびそのつぎの表を参照

最適計画における製造原価・輸送費の総計 ¥ 1,087,013,000円(この例の場合ほ費用の最少化)。

A 製品自家消費輸送量の内訳 (トン/年)

設備区分	自家消費 輸送量	転送先		
		千葉B製品設備 建設予定工場	神奈川B製品 製造工場	山口B製品製 造工場
千葉第1工場	2,004	280	2,324	
千葉第2工場	9,460		9,460	
神奈川工場	3,904		3,904	
山口第1工場	7,500			7,500
山口第2工場	1,100			1,100
九州工場	2,304			2,304
計	26,872	280	15,688	10,904

B 製品自家消費転送量

設備区分	自家消費 転送量	転送先				
		千葉工場用	九州工場用	新潟工場用	神奈川工場用	山口工場用
千葉B製品設備 建設予定工場	528	528 (280)				
神奈川工場	17,700	1,872 (991)		1,000 (530)	14,965 (7,931)	
山口工場	11,700		2,100 (1,113)			9,600 (5,087)
計	30,065	2,400	2,100	1,000	14,965	9,600

( )内はA製品の換算量を示す。

(5) 結論

- ① A製品の神奈川工場は約4,500トンのidle capacityを生ずる。これについては需要の拡大をはかるとともに荷役設備の整備の検討を必要とする。
- ② 千葉に建設予定のB製品工場は計画規模(6,000トン/年が最小経済規模)に対して、稼働率がきわめて低位にとどまる見込みとなった。したがって当面の建設は中止する。

③ 上記②の生産downをcoverするため、神奈川工場A製品の荷役設備の整備を行なうか、または、A製品を外部から購入するかなどにより、山口工場のB製品工場の生産を増加させる必要がある。

備考 このLP計算の解法のためIBM360-50Hを使用したか、所要計算時間は約50秒であった(UNIVAC-60で10年前に同様な計算をしたときは30時間以上を要した)。

表 II - 2 O R 模 型

在庫模型	配分模型
在庫保持に要する費用と注文費(自家生産の場合は段取費)および品切損失との間に費用の釣合いを考慮しながら発注(または生産)の時期および数量を決定する模型	なすべき行動や方法が多数あって、それを利用しうる資源や設備が不足している場合に、全体としての行動を最適化する模型。 線型計画(Linear Programming)や輸送問題の解法、さらにインテジャー・プログラミング(Integer Programming)、ダイナミック・プログラミング(Dynamic Programming)、二次計画法(Quadratic Programming)など
待合せ模型	取替え模型
サービス地点において、サービスを受ける側の費用とサービスを提供する側の施設費との和を最小ならしめる模型	修繕費その他の維持費用および非能率によるマイナス面と新しく取換える費用との合計額を最小ならしめる模型
競争模型	
一方の決定による効果が他の決定によって影響を受けるような問題に適用できるOR模型で、これに入札模型とゲームの理論とがある。	

表 II - 3 LP の計算時間とコスト

計 算 機 名	LPの解ける最大サイズ	LPサイズ	計算時間	推定計算コスト	備 考
UNIVAC-60		22(式)×192	30時間		計算穿孔機といっ て電子式の計算機 ではあるがプログ ラム内蔵方式ではない
UNIVAC-120	20式	15(式)×20	12時間		
Bendex-G15	50式	48(式)×75	25時間	¥750,000	第1世代のコンピ ュータ
USSC-60	48式	48(式)×75	4時間	¥200,000	
IBM-7090	1,024式	48(式)×75	1.2分	¥5,000	第2世代のコンピ ュータ
CDC-3600	4,096式	48(式)×75	1.2秒	¥1,000	第3世代のコンピ ュータ
IBM-360-50H	4,095式	22(式)×192	50秒	¥1,000	
IBM-360-75I	4,095式	48(式)×75	4秒	¥400	

三菱石油新野氏の論文中的表に私見を追加して作成した(『事務管理』1971年1月号)

### Ⅲ 経営情報システム (MIS)

#### 1. 情報化時代と新時代経営

##### 1. - 1 情報化時代とは

情報化時代の特徴をあげてみるならば

第1に、私達は従来物の生産あるいはエネルギーの生産に非常に力を入れてきたのであるが、現在はむしろ情報の価値を追求する時代になっていることである。

第2に、重要な情報の量が爆発的に増加してきており、これをいかに処理して、実際のポリシーなり行動に結びつけていくかということである。

第3に、週5日制の企業増加により、人間の生きる意味あるいは人類の目的は何かという従来は哲学として扱われてきた問題が、社会的現象として問題になってきたことである。

要するに、激動の時代であり、変化の速度が非常に速くなっていることである。

##### 1. - 2 新時代経営の特徴 (必要条件)

激動の時代において、企業をいかに長期的に指導していくかは、経営者・管理者の職責であるが、いかなる企業体・組織体においても必要になるものが2つある。ただし、これは必要条件であって、十分条件ではない。

第1は、マスタ・プランである。企業は一般に、3年あるいは5年先にどのように会社を持って行くかという経営の長期計画を立てる。この長期計画とは、3年あるいは5年後の企業目的を設定し、それを実現するための年次計画であり、その前には必ず明確な定量的目標がなければならず、これをマスタ・プランというのである。

第2に経営組織と経営情報組織の弾力化である。目標を達成するために、会社の組織をいかにするかということが問題になる。また、これと表裏一体をなすものとして、経営情報組織の問題がある。いくらマスタ・プランを立てても、世の中は予想外に変化するものであるために、迅速にそれらの変化に対して対応できる必要がある。これが、経営組織と経営情報組織の弾力化である。

## 2. 対象業務システムの分類

一般企業において、合理化の対象になるものをシステムの設計対象と呼び、大きく分けると3つになる。以下、順次説明する。

### 2-1 大量作業事務

社員の給与計算、顧客への請求書発行事務、株式計算などで、過去における機械化の大部分はこれを機械化したことである。この対象業務の特徴は企業なり組織体の生理現象であることで、なぜかというならば、その業務は絶対に企業としては不可欠ではあるが、その機械化を行なったところで特別な利益がでるわけでもないからである。しかも、月のある時期までに出来ればよいという性質のもので、また比較的初歩的な機械化であるが、最近非常にこれが脚光を浴びているのである。なぜならば、過去においては人の採用は企業が自由に行なえたわけであるが、現在は人手不足であり、また将来にわたっての日本の人口構成もきまっていることから、経済成長を持続して行くには労働装備率を高くして、個人の生産性を上げて行かざるを得ないからである。結局は省力化である。このように、今日の投資は生産力増強の投資というよりも、省力化・合理化投資に力が入っているのであり、その意味からも大量作業事務が大事な対象になっているのである。

### 2-2 科学技術計算

科学技術の急激な発達と共に、造船・建築・設計などの計算は非常に複雑なものとなり、コンピュータがなければ仕事は出来なくなってきたのである。この機械化の利益は非常に大きい。

ただ、大量作業事務および科学技術計算の機械化は企業全体から見た場合、大きい投資ではあるが、及ぼす影響は局部的であり、次にのべる経営管理の機械化とは大きく違っていることを理解すべきである。

### 2-3 経営管理

経営管理の機械化の特徴は、ある一部分について始めたとしても、それだけではおさまらないことである。たとえば、いま、困っている問題が在庫管理であり、製品在庫の管理がまずいために成績が上がらない場合、これを合理化するために機械化を始めたとする。このような機械化を個別機械化と呼ぶ。この様な機械化で在庫管理が一応うまくいった場合、在庫が置ったのは

よいが生産管理が問題だと気がつき、また生産管理が直ると販売管理がよくないと気がつき、最終的にはこれは全体的問題ではないかとなってくるのである。つまり、管理の機械化は部分的に始めても、結局はその重要なところだけは全社的な機械化に発展し、個別機械化から総合機械化へ、いわゆるトータル・システムなりM I Sと一般に呼ばれることになるのである。

さらに、もう1つの特徴は、工場のオートメーション化とは違って、管理を前提としての事務処理は企業の神経系統であるために、組織の末端にまでつながっており、影響は全社に及ぶということである。そのために組織は当然変り、企業の体質改善になり、大問題となるために、トップが中心となってこの企画を推進するということは当然のことになってくるのである。

### 3. M I S 設計の特徴

企業は、人事、経理、資材、生産等の様々な管理機能を持っている。この管理機能はそれぞれ所管範囲を持って管理をしているが、それらはある1つのはっきりした目標に従って、お互いに相助けあって、企業目的に奉仕している。従って、こういうシステムを部分最適の考え方で設計した場合、全体として非常につながりの悪い、効率の悪いものになるのである。

M I S 設計を理論的に考えるならば、3年あるいは5年後の会社のあるべき姿を決め、それを行なう企業の目的から出発し、その最終目的を達成するためにはどのような組織化を行ない、各部門にはどのような目標を与えたらよいかという2次目的を決め、さらに3次目標を決めて企業目的の方から演繹的に計画を立てるのである。それにより、一般企業の場合、幾つかのファンクション、たとえば販売管理関係、経理関係、人事関係、製造関係等に分かれる。個々のサブ・システムの誕生である。個々のサブ・システムはそれぞれにある仕様が与えられ、その仕様に従って合理化・機械化が決められ、それぞれのシステム設計チームを任命し、その仕様書に従って、個別のサブ・システムの設計を行えば、素直につながるといえる考え方である。

このように、目的から出発して手段の方へ演繹的に計画を立て、実行は長期計画に従って1番下のところから積み上げていくというやり方を、M I S 設計

というのである。この場合、気をつけなければならないのは、経営組織と経営情報組織とは表裏一体であるので、当然一緒に設計しなければならないということが基本的な条件になっているのである。

過去には、個別の機械化を積み重ねて、それらを横につなげば全社的な機械化が出来るのではないかという考え方として、トータル・システムがあった。この場合、必要最少限の組織の変更は考えられたが、初めから組織を再設計するという様には考えなかったのである。確かに、情報組織はどのような企業組織でも一応はサービスできるが最適の保証はないのであり、最適の保証のないシステムは作成しても効果の保証がないのである。

次に考えなければならないのは、M I Sを教科書的に行なうことが不可能な場合、応用動作を考える必要があるが、その場合でも教科書は試行錯誤を少なくするための指針であるということである。

#### 4. 経営管理の機能

経営管理を機能的に分類すると、(1)総合管理・(2)定型的管理・(3)計画の3つになる。以下、順次説明する。

##### 4-1 総合管理

総合管理とは、トップが日常行なっている非常にむずかしい管理である。なぜならば一般に、総合管理とは、お互いに関係があり、しかも相矛盾しており、あちらを立てればこちらが立たないという仕組みになっている幾つかの案件の中から、最適の決定あるいはセレクションを行なうという意思決定の機能だからである。また、一義的に決まるものはすでに下の方で決め、トップのところには本質的にむずかしいものしかこないからである。したがって、どうにか満足できる程度の決定を是非行わねばならないのであって、最適決定などというものは、とうてい人間にはできないものである。

それでは、このようなむずかしい総合管理をコンピュータのシステム技術、あるいは経営科学手法等の情報処理機能を持って援助をすることができるかという問題であるが、部分的には現在すでにうまくいっているところもあるが、非常にむずかしいというのが現状である。なぜなら、1つは、このよう

な意思決定は必ずしもロジカルに行なわれるとは限らないからである。個人のフィロソフィあるいはパーソナリティというものが、意思決定に非常に大きな影響を及ぼしているからである。

ロジカルに決定する部分については、これを助ける手法はいろいろあり、例を上げればPERT、シミュレーションなどであるが、むしろこの分野が進むことによって、トップの必要性はかえって増すことになるのである。

次に、情報処理の面から見ると、いかなる意思決定がなされるにしても、必要な情報が幾つかなければ不可能である。たとえば、ある重要な1つの情報が間違って遅れて来たらどうであろうか。ほかの情報は非常に早く正確に来て、トップの意思決定の精度が一番悪いもので決まることになるのである。このことは、トップの意思決定のために必要な情報は、同じ早さ、同じ正確度で来なければ困るということで、情報源となる所を合理化し、機械化しなければならないということになるのである。MIS設計の特徴で学んだことであるが、A部門とB部門は合理化され、機械化されたが、C部門とD部門はされていないということでは、同じ早さ、同じ正確度では情報が来ないために、トップの意思決定には役立たないということである。確かに、一部門において機械化されることはその部門においては利益があるが、トップの意思決定には直接役立たないことであり、役立たせるためには一とおり重要なところだけでも、機械化する必要があるという事である。これが、管理の機械化というものは個別機械化から出発しても、結局は全社的な総合機械化に進まざるを得ないという最大の理由の1つである。

#### 4-2 定型的管理 (オペレーショナル・マネジメント)

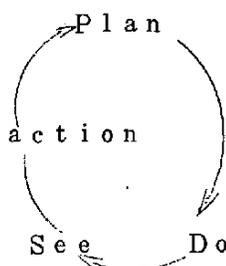
定型的管理とは、各部門で行なわれており、一般に課長、係長が行なっている管理で、合理化・機械化によって得られる利益は非常に大きいものである。

第1に、会社のいたるところで定型的管理が行なわれているので、それらをすつきりとしたシステムにすれば、得られる利益がいたるところで上がるわけだから、総合的に非常に利益が大になるからである。

第2に機械化作業であれ手作業であれ、管理の手段としての事務を行なった場合に、事務のアウトプットは何らかの形のレポートである。それらのリ

ポートはトップへ直接行くことはなく、大部分はミドルの行なっている定型的な管理につながっているのである。これをすっきりすると、物理的な情報の量が激減し、情報処理が楽になり、手作業の場合は事務要員が減り、機械化の場合は設備投資が減ることになり、経済的に非常に大きな利益になるのである。

第3に定型的な管理は機械化にまじめに取りかかれば、必ずよいシステムができることである。定型管理の最も基本的な合理化手法を以下述べる。



定型的な管理というのは、会社の目的が決まると、その目的を達成するために組織化が行なわれ、各部門ができ、各部門に対して部門目標が与えられる。その与えられた目標をいかにして実際に達成するかという管理のことである。

たとえば、予算とか目標あるいは管理限界という言葉を使用するが、このプランに属するものは、目標毎に1つのスタンダードをつくることである。そして実際に経営活動を行ない (Do)、その結果オブザベーションにする (See)。Seeの過程は実績データを集め情報処理を行ない、その結果をレポートの形式等の何らかの方法で、プランすなわちスタンダードにぶつけるわけである。計画どおりに行く事はほとんどありえないことで、くい違いが起こる。そこで目標・実績の差異の分析を行なうわけである。これが普通定型的な管理の典型的な形である。

ただ、差異分析を行なうだけではなくて、その差異分析の結果、本能的に差異の悪いほうから大きいものに対して、管理上のアクションをとるわけであり、管理上のアクションとは、「うまく行かなかった現場に対してそれがうまく行くように、管理レベルから援助すること」である。すなわち、一番

うまく行かないところからアクションを取って行くので、全体がよくなるというしかけになるわけで、これで一応管理ができるわけである。

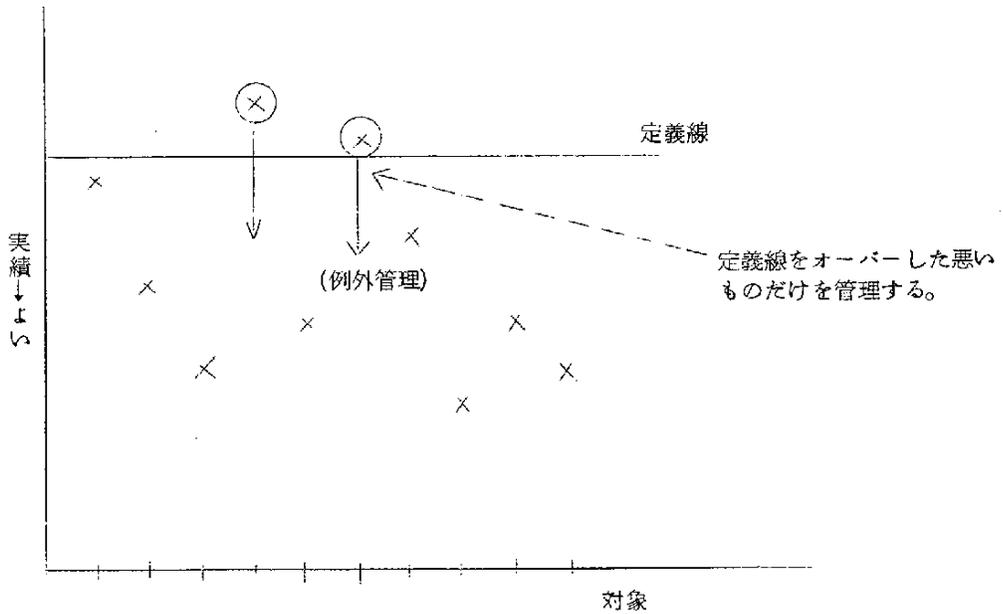
ところが問題点としては、本能的に悪いほうからアクションをとっていくことである。科学的管理というのは、計数的管理すなわち定量管理であるべきで、本能的管理では困るのである。そこで、これを計数化し、定量化して科学的管理に持って行く一般的な方法について説明する。

最初に、横軸に管理する対象（支店、工場、セールスマン等）をとり、縦軸に実績値をとるのである。実績というのは、普通上へ行くほどよいものと、下へ行くほどよいものがあるが、今ここでは下へ行くほどよいとする。もし売り上げみたいに上へ行くほどよいものがあれば、逆数をとればよいわけである。この図においては、上の方が悪くなるが、これをはっきりするために模線を一本引き、これを定義線と名付ける。

この定義は、この線の上にあるものは悪く、この線の下のはよいということである。この線は動くのであるから、これを下げると悪いのが増え、上げると悪いのが減るといっしかけである。この図はいわゆる品質管理（QC）の管理図である。デスク・ワークも品質管理を行なえるというので出来たのがTQCである。ただ、人間を対象にする管理は物ほどの確には行かないので、ある程度ソフトな定量管理が一番適当である。たとえば、全管理対象の1割をアクションの対象して、前期のデータをプロットし、その1割がこの線の上に出て来るところへ定義線を引く。そして、この定義線のみを残して前期の実績点を消し、今期のデータをプロットする。それによって、ちょうど1割にはならないであろうが、その前後のものは定義線の上に飛び出してくるであろうから、これだけを管理すればよくなるのである。このように、この線から上に出たものだけを管理の対象とし、管理しないものの報告は必要なくなる。これを例外管理（Management by Exception）というのである。

これによって、報告書の紙だけでも約9割近く節約になるのであるが、本質的に学ばなければならぬことは、必要のない紙くずの山を一杯作ることに、管理を非常に毒している元凶を断たねばならぬことであり、管理につながらない、またはアクションにつながらないものはいくら作っても

図 III - I 科学的管理方法



意味がないということである。

次に、定型的管理を3つに分類して考えるならば、管理として進歩している順にならべると、第1はコントロール・システム、第2はコントロール・レポート、第3はゼネラル・レポートである。

コントロール・システムとは、ある目標ごとに管理限界（基準）を設定しその定義線を出たものについて、コンピュータからアクション指令が出される状態になったときをさし、その指令に基づいてアクションをとるのは人間でもよい。コンピュータがアクション指令を出すためには、まず何がよい、悪いという目標ごとの基準（定義線）を明確にして、コンピュータに教えておく必要がある。そして定義線を切った場合のアクション指令の内容も教えておく必要があり、以上2つを教えておけば、母集団になるもとデータは全部コンピュータの中にあるわけであるから、定義線の上に出たものにアクション指令を出すことはコンピュータが行なうのである。

次に、コントロール・レポート（管理報告）であるが、何がよい、悪いは決めたにしても、アクションは報告書を人間がもらって、人間が判断してアクション指令を出すという段階もあるわけで、この段階をコントロール・リポ

ートという。

最後に、ゼネラル・レポートであるが、これは何がよいか悪いかわかってなく、悪ければどうするかも決まってない段階である。そして重要なことは、コントロール・システム、コントロール・レポートについては例外管理原則が適用されるわけであるが、ゼネラル・レポートには例外管理原則が適用されないのである。このゼネラル・レポートは一般に管理に役立たないといつてよい。

#### 4-3 計 画

今まで色々と学んできたわけであるが、何がよいか悪いかを決めたり、悪かった場合どういうアクションをとるかということを決めるのは結局、計画であるということである。

計画は大きく分けて2つある。その第1は戦略的計画で、トップ・マネジメントで行なう総合管理をバック・アップするシステムである。すべてについて戦略を立てればよいわけではなくて、企業の性格あるいはその企業の置かれている立場によって、計画を立てるのが重要なのである。たとえば、固定運営費の会社では設備投資、いわゆる設備計画とか立地計画等が重要である。ただ最近、一般的には人材の質および量をいかにして確保するかという、人事シミュレーション等の計画が脚光をあびている。

第2は定型的管理の計画で、パラメトリック・システムというのがある。このパラメトリック・システムは、定型的管理の特にコントロール・システムをバック・アップする計画システムであり、非常に新しい分野であるために、世界中の企業全体をつうじて、全部が完成しているシステムはまだない様である。

コントロール・システムにおいては、ある管理限界を設ける定義線を引き、その線を出たものについてはコンピュータから指令が出てアクションをとり、悪い方がよくなるのであるが、しばらく行なうことにより定義線にひっかからなくなってくるわけである。ひっかからなくなると、今度はコントロール・システムがオペレートしなくなり、定義線を下げることになる。どこへ下げるかというのは計画の仕事であるが、それはコンピュータに「全体の1割をひっかけなさい。」とか、「標準偏差の2倍で切りなさい。」と指令を与え

ることによって、母集団たるもとデータはコンピュータの中にあるわけだから、簡単に計算して新しい定義線を引いてくれるのである。この定義線を引く仕事が、パラメトリック・システムの原型である。

定義線が引かれると、コンピュータの中でコントロール・システムに移せばよいのである。これを繰り返せばよいので、これによって自動計画、自動管理が行なわれるのである。ところがこれには1つの欠点があり、実際の経営の場合を考えればすぐ判るように、ある特定の目標がよくなればよくなるほど、よいということはないわけである。たとえば、時計をつくることを考えると、どのような性能の時計をつくるのか（質的な目標）、何個つくるのか（量的な目標）、いつまでにつくるのか（納期目標）、そして原価目標とお互いに因果関係があるのであり、納期は短いほどよいと半分にしたら、質が悪くなり、コストが高くなるようなことである。

それではどうすればよいかという問題であるが、目標が4つならその各々について、コントロール・システムは動いているわけである。では、その目標をどのように決めるかということは、結局4つの相関のある目標の間に、あるバランスが保たれる必要がある。そのバランスをとるパラメトリック・システムについての考え方であるが、比較的相関の強い目標群の中でバランスを何に対してとるかというのと、上位目標たとえば工場の利益最大あるいは売り上げ最大に対して最適になるように、いまの目標の間のバランスを設定するわけである。これは目標間の因果関係というものを公式にして、シミュレーション・モデルを設定するのである。

それに対していろいろな数字を当てはめて、シミュレーションを行ない、上位目標が最適になるところでストップすれば、4つなら4つの目標が同時にきまる。このようなシミュレーション・モデルをビジネス・モデルといい、会社の中には、実際は1つではなく沢山あるのである。しかも、大会社になると何段階も経営階層があるので、その段階ごとに沢山のビジネス・モデルがあり、それらがすべて階層的につながるわけであるから、そのようなビジネス・モデルの連鎖たる1つの大きな計画システム、会社全体の計画システムが考えられるわけで、その全体をパラメトリック・システムと呼ぶのである。

結局、このようなことを行なうためにも、その出発点は例外管理であって、合理化の第1歩として絶対に行なう必要があり、せめてコントロール・リポートの段階に持つて行くことが合理化の1つの大きなキイ・ポイントである。

## 5. システム弾力化の手法

システム弾力化が考えられてきた背景として、コンピュータは大量作業事務あるいは科学技術計算等の場合は、超高速で作業を行ない、また膨大な資料を何時でも抜き出せる形で持つているので問題はあまりないが、経営管理の場合はシステムが非常に硬直化するので問題がある。コンピュータはプログラムの一字たりとも間違っていれば、間違いとして処理するので、「偉大なる低能児」といわれるのである。そのため、そのようなコンピュータを基本として、システムを作ると、非常に硬直化したシステムが出来上るのである。

一方、経営は変化するのが普通であり、今日作ったものが明日役に立たなくなることも、たびたびある。しかし、欠点は多々あるが、それを補って余りある程、コンピュータは利点を持つている。しかも、硬直化しがちのシステムをいかに弾力化するかが、システム設計の腕であって、弾力化手法はシステム設計のあらゆる段階で重要な設計ポイントになる。

詳細については別項に譲り、ここでは大きな形で開発されてきたシステムの弾力化手法について説明する。

### 5-1 経営情報システム弾力化の例

#### 5-1-1 TSS (Time-Sharing System [時分割システム])

TSSはコンピュータのCPUを、多数の人が、離れたところから、同時に、即時に、共用する考え方である。

多数の人とは何百人であってもよく、離れたところとはテルスタを使って太平洋を横断してもよく、同時とは同じ時刻に一斉にという意味であり、即時とはデータなりプログラムを入れて、数秒待つている間に返事がくるといふことである。さらに、TSSの非常に重要な条件は、ある1つのコンピュータを何百人が一緒に使用していても、使用者が専有感を持つて使用できることである。

TSSを考へてきた背景であるが、コンピュータは絶対価格が相当高く、処理速度が早いために遊びの時間（アイドル・タイムとかハンドリング・タイム等）が長いので、これを有効に使うと非常に効率が上がる。そのため、ユーザの場合、コンピュータをフルに稼働させるために、大日程、中日程、小日程、最後には分単位の時間管理で仕事が流れるように、工程管理や作業管理を行なうのである。また、メーカーは、同じ目的でコンピュータを遊ばせないために、モニタリング・システムを開発したのである。モニタリング・システムとは、仕事を行なっていて、エラー等に引っかけた時、コンソール・タイプに即座にどのようなエラーかを打ち出し、エラー・リストをつけて送り返すのである。その間コンピュータは、すぐ次の仕事にかかれるように前の仕事をハジキだしてしまう。

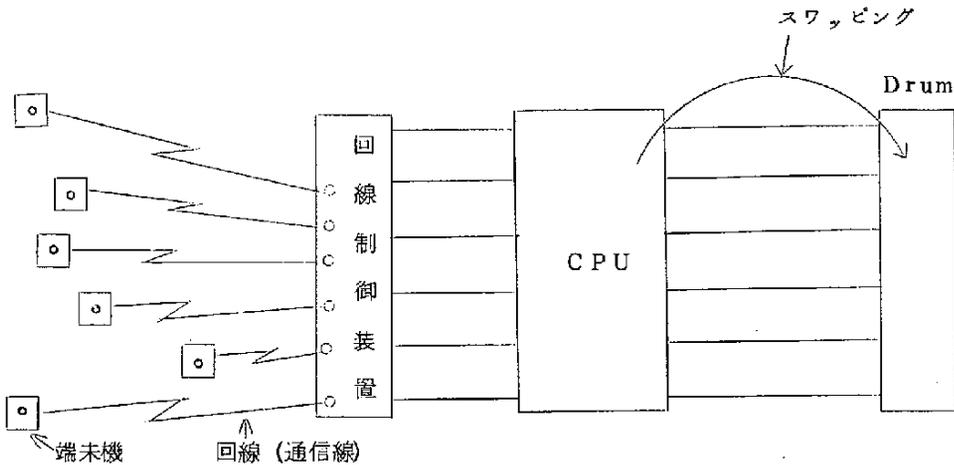
その結果、コンピュータの稼働はよくなったのであるが、仕事を頼んでも長く待たされ、いつ出来てくるかわからないために、コンピュータの利用者である研究者とか企画者等を遊ばせることになり、全体効率としてはまずいことになったのである。それらを解決するものとして考へられたのが、アメリカのマサチューセツ工科大学（MIT）のプロジェクトMACである。この考へ方の設計思想の基本は、データとかプログラムを入れると即座に返事が返ってくることを絶対条件とし、かつ、コンピュータの効率もなるべく落とさないようにするというものである。

なぜ、その様な事が行なえるかを概念的に説明する。

回線とコンピュータをつなぐ場合、回線制御装置を間に置き、その回線に端末機を必要な台数つなぎ、回線制御装置とCPUをつなぐのである。端末機タイプライタかビデオ・ターミナルを使用する。人間がそのタイプライタを打って入力するので、速さはおそく、またこの通信線の速度は制約される。現在、一般に使用されている通信線の速度は、50ビット・パー・セカンド（50BPS）、あるいは200BPSで、50BPSとは1秒間に50短点送るといふ早さのことである。ところが、コンピュータの処理速度は大体マイクロ・セカンド（=百万分の1秒）あるいはナノ・セカンド（=10億分の1秒）のオーダーであり、通信回線の速度と大幅な隔りがある。

回線制御装置にはバッファ（一種のメモリ）があり、送られてきたデータ

図 II - 2 コンピュータと通信回線との結合



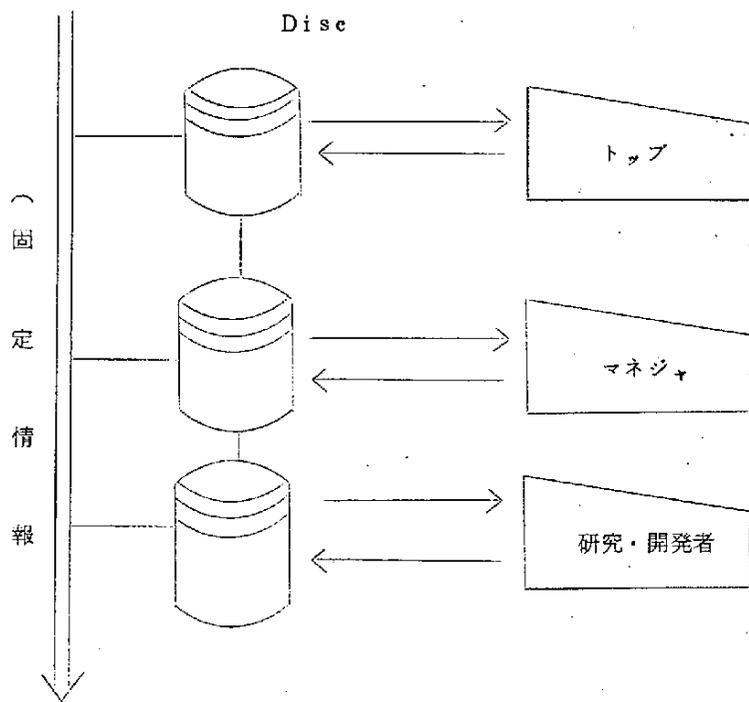
は一度ここにためられて、一杯になるとCPUの内部に入れてその問題を処理し、結果を送り返すのである。問題を処理し、結果を送り返す時間であるが、これはTSS設計の基礎であり、いろいろ設定できるが、かりに20ミリセカンド(50分の1秒)で処理して返すというシステムを作っても、決して非常識ではない。たとえば、20ミリセカンドは50BPSで送った1短点の早さであるから、結局、全然待ったという感じがなくて、専有感を持って使えることになるわけである。

しかし、その場合にも、時には20ミリセカンドで終わらないものがある。これらの処理にはランダム・メモリの補助記憶装置として磁気ドラムを置き、計算途中の式と数値とプログラムを退避(スワッピング)させるのである。そして、このあいたバッファへすぐ次の問題を入れて処理するのである。この様に行なう場合、処理装置にあきが出る時期があり、その時に退避していたものを処理して返すのである。簡単なようであるが、全然違う仕事を並行的にCPUは行なっているので、それらを間違えないように監視するソフトウェアは膨大な量になる。しかし、これが行なえることにより、使用者へのフィード・バック時間は早くなり、システムは非常に弾力的になるのである。

#### 5. - 1. - 2 データ・ベース (データ・バンク)

データ・ベースは設備的に、またシステム設計上、非常に大きな問題に今後なるであろうと言うことが出来る。

図 III - 2 データ・ベース



上図の矢印は固定情報処理を意味する。固定情報処理とは、会計決算あるいは工程管理の一部の様に、きまり切ったプログラムできまり切った処理を行えばよいというものである。これはコンピュータの非常に得意な分野で、良いプログラムを作っておけばすぐに処理できるのである。しかし、ルーチン作業中に臨時の要求が割り込んでくるとトラブルが起きたりするために、これを解決する方法として、必要であろうと予想されるものを抜き出してためておくのである。

たとえば、ランダム・メモリの比較的大容量の磁気ディスクを使用するのである。設計によって設置数はいろいろあるが、この例では3つ設置したとする。トップにはトップ専用を、研究開発者には研究開発者用の専用をとり具合にである。

あるトップ・マネジメントが、ビデオ・ターミナルかタイプライターを打って、北海道地区におけるある商品の売り上げは、現在幾らになっているかあるいは、その工程はいまどこまで進んでいるかということを知りたい場合に、

返答は即座（リアル・タイム）に出てくる。なぜ、リアル・タイムに出るか  
というと、ルーチン作業ではなくて、ディスクから直接に返答するからであ  
る。ただ、この場合に、聞かれそうな答を、最適の配列で入れておく必要  
がある。聞かれそうな答といってもいろいろとあるわけで、システム設計者  
として注意しなければならないことは、あのトップは過去においてこうい  
うことをよく聞いたとか、あのトップはいつもこういうことに関心を持って  
いるからたぶん聞いてくるだろうとか、その他トップとして当然知っているべ  
きだという事を選択することである。最初は少しトラブルが起きるだろうが、  
よく聞かれるものは詳しく、全然聞かれなないものは取り去り、トップが進歩  
して高度になれば、除々に高度なものに、入れ替えてやればよく、こうし  
ているうちに急速にディスクの内容がトップ・オリエンテッドになってくる  
のである。

しかし、トップが変わるとシステム設計はやり直すことになる。もちろん  
トップのデータを管理者の所へ廻したり、その逆も出来る。内容によっては、  
ロックすることも可能である。

次に、データ・ベースの使い方について説明する。

第1は、単発データの問合せである。普通、トップ・マネジメント等があ  
のデータは今どうなっているかと聞くことで、一種のIR (Informa-  
tion Retrieval) のため、比較的抜き出し易いのである。

第2は、課長又は企画者レベルの場合、単発データを聞いただけでは仕事に  
ならず、たとえば表頭と表側があって、これに数字の入ったような形で、表  
形式のアウトプットが必要となるわけである。単発データの抽出はコマンド  
と言ってごく簡単なもので出来るが、表形式のアウトプットには一寸したプ  
ログラムが必要である。たとえば、RRG (レポート・プログラム・ジェネ  
レータ) の簡単なものとして、DPG (ディストリビュート・プログラム・  
ジェネレータ) を使用する場合もある。

第3は、加工した結果—この加工というのはたとへばシミュレーションの結  
果を打出す場合等—をアウトプットするということである。そのためには、  
シミュレーション・モデルをいれておく必要がある。

第4は、アラームである。第1から第3までは使用者からデータ・ベースに

向かってアクセスするのであるが、これはコンピュータの側からくるのである。この端的な例は、例外管理で管理限界を切った場合に、アラームを自動的に出すことである。

### 5-1-3 モジュラ・プログラミング

モジュラ・プログラミングは、プログラムやシステム設計を非常に簡単にしようという手法である。これは、たとえば、磁気テープの内容をプリントアウトせよとか、2つの数列を一緒にして大きさの順位に並べよとかの様に、幾つかの単位作業の集積になっているわけであるから、事務作業を部品の段階つまり単位作業にブレイク・ダウンして、標準化してプログラムを組むわけである。これをモジュラプログラムという。一般にこの種のプログラムは出来合いであるが、部分的にはパラメータを変えることにより、多少の変更は可能であるが、制限はもちろんある。

効果であるが、システム設計からプログラムにかかる仕事が4週間の場合に、この種の手法を用いると4時間程度で終わることになる。しかし、欠点もあり、既成のプログラムをつなぐために、プログラムが比較的長くなり、ランの数もふえるので、コンピュータ・タイムが、1割から2割ぐらい長くなる。しかし、4週間かかるものが、4時間でできるので、非常に弾力的であり、急に何かが起こった時に利用するには、非常に効果的である。

次に、モジュラ・プログラムの使い方について説明する。

第1は、システム弾力化の使用で、急に必要が起こった場合に、すぐ対応できるというのが1つの特徴である。

第2は、システム設計上の有利な点で、普通システム設計が一番困るのが計画の初期である。つまり、初めは皆、素人ばかりであって、どうすればよいかかわからず困ってしまうのであるが、モジュラ・プログラムは簡単なプログラムの組み合わせであるから、すぐ理解でき、実行できるのである。また、そういうことを行なっている内に、皆の腕があがってくるから、常時プログラムをかえないですむものは、アッセンブラとかコボルに切りかえればよく、たびたび変わるものは、つくりかえる方が大変であるから、モジュラ・プログラミングを使用した方がよい。

現在、この技術がどんどん進みつつあるわけだが、おそらく第4世代のコ

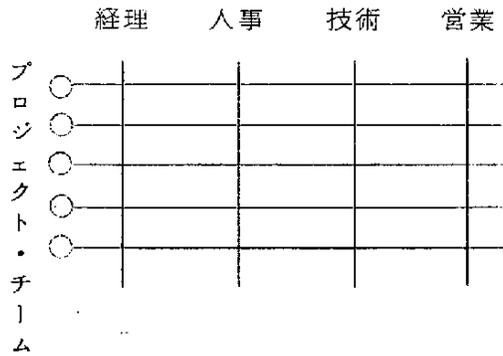
ンピュータになると、想像であるが、今までとは逆に、既成のプログラムを一層こまかい形で標準化して、それをLSI (Large Scale Integrated Circuit) で組むようになるであろう。従来は、ハードで行なったものを、ソフトに変えて行くのがコンピュータの常道だったのであるが、今度はソフトの開発が困難になったので、逆にハードに戻すというわけである。これをファーム・ウェアという。これにより、基礎的なコンピュータの本体にLSIのパッケージをつけて、自分の会社にマッチしたコンピュータをつくることも可能になると思われるのである。

#### 5. - 2 経営組織弾力化の例

最近、多くの会社においてはプロジェクト・チームとかタスクフォースが出来ている。そのメンバとして、研究者、設計者、製造エンジニア等が入っている。つまり開発を伴う、期限の非常にやかましい大きなシステムの場合、プロジェクト・チームのような非常に目的指向型の組織で行なっているのが普通である。なぜなら、ラインの仕事を考えてみるならば、それは大小のプロジェクトの集合なのである。従来の縦型組織の場合、人に関するのは全部人事、金は経理、製造は工場であり、また金は試算表では、要素別か何かに分かれて、人件費がどう、物件費がどうということになり、プロジェクト中心の管理には全然使いものにならないのである。そこで、その様なきびしい条件のもとで開発を伴うむずかしい仕事は、プロジェクト・チームで行なうことになり、さらに徹底した場合、ラインの仕事は全部プロジェクトの集合であるから、プロジェクト・チームの集合形体の組織がもっとも目的指向的でありという考え方になり、現在徐々にダイナミック・オーガニゼーションという型に移りつつある。

結局、プロジェクト・チームがたくさんできることになり、これだとある目的を持って、しかも必要な人間が全部一斉にその目的に向かって仕事をするわけであるから、非常に効率がよくなるのである。しかし、この組織には欠陥があり、プロジェクトであるから、いつか終わるわけで、終わると解散することになり、解散した人間が戻って行く縦型組織がないことになるのである。したがって、縦断的に見るものすなわち経理、人事、技術、営業という形で縦に見るものも必要ではないかということである。そして、仕事は横

に行なうということで、これをマトリックス・オーガニゼーションという。



例としては、すでに学んだPPBSや、NHKの番組編成の定型管理システムであるTOPICSなどがある。TOPICSは定型管理システムであるが、例えば土曜日の7時半から8時、8時から9時という様にチームがあり、端末機（ビデオ・ターミナル）を使って番組編成を行なう。シナリオは何を、スタジオは何番を、俳優は誰と誰を、放送範囲はどこどこということ打ち込むのである。それらはコンピュータに入力され、中で分解され、たとえばスタジオ手配係とか、放送線管理係とか、大道具・小道具あるいは俳優手配係とかに分解され、それぞれの手配をするのである。そして、普通一般にはあまり起らないとのことであるが、リソースが足りなくなると、調整係で調整するのである。

結局、従来のような縦型の職能組織ではどうにもならないわけで、情報組織と一緒に経営組織そのものが、非常にダイナミックになりつつあることであり、システム設計と同時に設計しなければ、情報組織のみがちりちりしても何にもならず、本当の効率的な仕組みにはならないということである。

## 6. M I S 設計の手順

M I S 設計の手順は、概要調査、基本設計、システム分割、システム構想の確立、長期計画の策定、各種サブ・システムの設計という形になる。各設計手順について細部手順、注意事項を順次説明する。

### 6-1 概要調査

現状調査を行なうわけであるが、純粹に目的から演繹することが重要な

であって、あまり現状に足を引っ張られてはならない。また、設計の始まりは基本設計からであって、概要設計と基本設計はバラレルに行なってよい。

## 6-2 基本設計 (Master Plan)

### (1) 企業目的の明文化

一般の会社においては企業目的は明文化されているわけであるが、もう一度将来に向かって、これでよいかどうかを再検討することである。

### (2) 主体条件の調査

たとえば、社内にどの様なリソースがあるか、自分の会社の性格はどうかという事で、たとえば、資産をどれだけ持っているか、優秀な研究者はいるか等を調査するのである。

### (3) 環境条件の調査

たとえば、経済状況、競争会社の状況、将来5年間の技術動向、マーケット動向とかの調査をするのである。

### (4) 長期基本方針の確立

以上の3つをもとにして、どの製品分野に進出し、どの方面は中止し、どの程度の売り上げの絶対額にして、利益率をどのするかというようなことをトップが中心になり、企画者とかシステム設計者の一部を入れて、マスタ・プランの本体を作るのである。

## 6-3 システム分割

### (1) メジャ・システムズ (major systems) への分割

前段階の結果から、組織を再検討し、会社の全体の守備範囲を決め分野に分けるのである。この分割の仕方は種々あって組織論の問題になるわけであるが、製品別分割、地域分割、職能分割等が考えれる。第1次の分割単位をメジャ・システムまたは主要系列業務といい、この分割も幾つかの案があり、代替案つまりオルタネティブをつくり、その中でどれが一番企業目的に合うかということを検討して決めて行くのである。1次分割を終わると、2次分割を行なうのである。製品別分割を1次で行なうと、職能分割を2次で行なう事になり、たとえば、事業部の経理とか、事業部部の営業という様な形に分かれて行くのである。しかし、2次分割であっても、システム設計単位としては大きすぎるので、普通は小規模な場合で

も、3次分割がシステム設計単位になる。システム設計単位とは、3人から多くとも10人位の1チームでもって、数カ月、あるいは1年以上かけてシステム設計をして機械化する単位である。もちろん分割したものでも機械化しないと決めることもある。

#### (2) サブ・システムズ (sub systems) への分割

この場合は代替案の検討を行なって、各メジャ・システム、サブ・システム毎に部門目標、管理方式 (基準・標準)、アクション内容、成果測定方式等を決めるのである。

### 6. - 4 システム構想の確立

概要設計からシステム分割まではコンピュータを全然考える必要はなく、この段階になって考えるのである。たとえば、

- (1) コンピュータ・センタを中央に集中した方がよいのか、分散した方がよいのか、工場に入れたらよいのか、本社に入れたらよいのか。
- (2) オンライン・リアル・タイム・システムをどの分野に適用するか、適用しないか。
- (3) データ・ベースはどのような形で作るか。
- (4) モジュラ・プログラミングの手法をどの様に活用するか。
- (5) 各種ディスプレイを含む端末等をどの様に考えるか。
- (6) 例外管理の手法をどの様に活用し、コントロール・システムをどのような形で作っていくか。
- (7) 運用システム (=各種の色々な規定類で全部改定せねばならない) をどう作っていくか。

ということ、マン・マシン・インターフェースに注意して、計画するのである。

### 6. - 5 長期計画の策定

ここでは以上の資料を全部もとにして、(1)システム設計計画、(2)設備計画、(3)要員計画、(4)組織計画、(5)教育計画、(6)運用計画、(7)P・R計画、(8)資金計画、(9)その他では全体の新技術開発計画とか、マーケティング計画等を考えるのである。

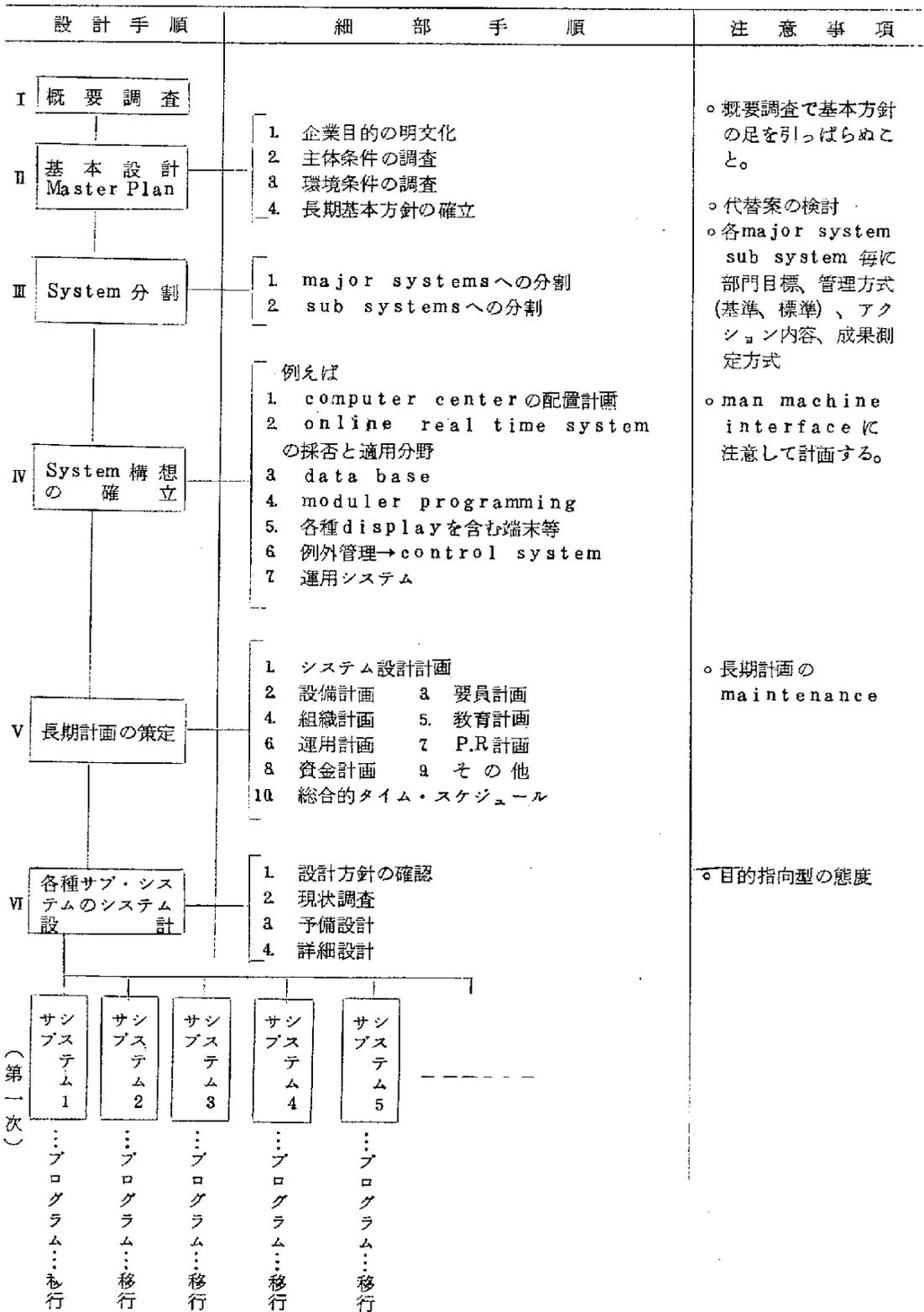
これらは全部因果関係があり、たとえば、建物が建たないとコンピュータ

は入らず、教育が終わらないとシステム設計が出来ないという様に、相互関係があり、しかもクローズド・システム（閉じたシステム）となっていることである。これらの総合的タイム・スケジュールにはPERT法が一番よいのである。

#### 6. - 6 各種サブ・システムのシステム設計

ここまでくると、第1次に行なうシステムは何と何であるということが決まっているので、システム設計計画に従い、第1次の設計を行なう。たとえば、サブ・システムが5つあるとすると、それぞれにシステム設計チームをアサインして正規のシステム設計を行ない、プログラムが出来て、並行実施を行ない移行するのであるが、出発点は同じであっても早かったり遅かったりするので、1つ終わるとすぐ次の第2次計画の方へアサインして進んで行くのである。（図Ⅲ-4参照）

図Ⅲ-4 MIS設計の手順



## 7. システム設計の注意事項

### 7-1 M I S 設計上の基本事項

(1) トップみずからの仕事である。 会社の長期計画を M I S 化するために、プロジェクト・マネジャあるいはプロモータには、経営者の 1 人がなるのが普通である。

(2) 教育に始まり、教育に終わる。

(3) 長期的経営のニーズが先行する。

(4) M I S に既製品はない。 他社の行なったことは参考にはなるが、その事例を聞いてまねすれば出来るというものではない。

(5) 優秀なスタッフを専任すること。 優秀といってもいろいろあるが、絶対に欠くことの出来ないのは、経営上の長期の見通し、将来の見とおしをするというインテンションを持った人が 1 人は必要である。

(6) M I S はそれ自体成長体である。 3 年計画、5 年計画は要するに、マイル・ストーン（一里塚）であって、マスタ・プランは 5 年であれば常に 5 年先に必ず目標が移動するのであり、永遠のプロジェクトである。

(7) 経営組織と情報組織を一緒に設計すること。

(8) 人を信用すること。 人を信用するというポリシーの上でのみ合理化は成り立つのであって、疑ってかかれればきりがないのである。

(9) マスタ・プラン樹立困難な時は応用動作。 マスタ・プランの確立困難の場合は、個別機械化で開始し、将来ある時点でマスタ・プランの原点に戻ればよいのである。

(10) 外部情報は決定的に不足している。

### 7-2 システム設計上の要諦

(1) システムの目的を明確にすること。 目的が明らかで、仕様書がはっきりしているものをシステム設計するのはローレベルの仕事で、目的のわからない所を目的を捜して行くのがシステム設計の仕事なのである。また、評価基準を決めておく必要がある。目的のサブ・システムとしての評価基準とは、システムが成功したとはどういうことかということである。これを初めに決めておかなければ、全然違う評価基準を持ってきて、責められる危険がある。

(2) 管理者が自分の仕事と心得ること。 システム設計は管理者の基本的職

責である。管理者は基本的に2つの職責を負っているわけで、第1は組織をつうじて仕事をすることで、組織とは情報の処理とそれに基づく意思決定のメカニズムのことであり、自分の所管範囲の組織をよりよく設計し、よりよく運用するのがシステム設計であり、管理者の基本的職責なのである。第2は上司、同僚、部下とよい人間関係を保って、コミュニケーションをよくすることであり、その前半はシステム設計である。従って、システム設計者の教育は管理者教育であると考えらるべきである。

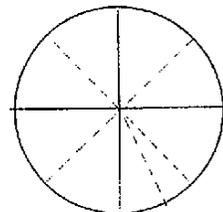
(3) 目的指向型の態度が各段階の設計者に必要である。経営情報システム設計の場合には目的を失う事はまずないが、個別のシステム設計に入ると、目的指向型から手続き指向型になる傾向がある。極端な場合、手作業をまる写しにした様なシステムをつくる危険があり、能率化ははかれたが、効率化ははかれないことになるのである。

(4) できるだけ弾力的なシステムを設計すること（既に述べてある。）

(5) できるだけ将来を見て進歩的なシステムを設計すること。  
将来使いシステムを設計するのであるから、現状に妥協してばかりいたら、いざ使い時には陳腐化して役に立たないからである。しかも最初の概要設計のとき、最も進歩的なシステムを考えていないと途中で様々をひっかかりがあり、後退することはあっても前進しないからである。

(6) 定型管理のアウト・プットは最少限におさえること。例外管理などはその例で、アウト・プットを少なくするのは、管理をよくするというのが目的なのである。

(7) システム設計単位はできるだけ大きくとること。大きくとるのは、アウト・プットを減らすためである。たとえば、MIS設計では、業務全体をいくつかのメジャ・システムに分ける。これを、仮にシステム設計単位とする。今、この円を会社全体の仕事とする。これを4つに分けて、それをシステム設計単位とする。そうすると、仕事と仕事の境目、たとえば資材と製造との関係、資材と経理



との関係、それをインターフェースという。インターフェースはシステム間のデータのやりとりである。この場合は4つの組合せである。ところで、これをさらに分割してシステム設計することは可能である。これにより、サブシステムができ、サブ・システム相互間に全部インターフェースができる。インターフェースはすべてデータのやりとりであり、普通は手作業を介して行なわれるので、分割すればするほど境界現象が多くなり、インプットとアウトプットがふえることになる。大きなシステム設計単位をとると境界現象が少なくなりシステムの中にはあるが、それはコンピュータの中でのやりとりである—アウトプットは減ることになるのである。

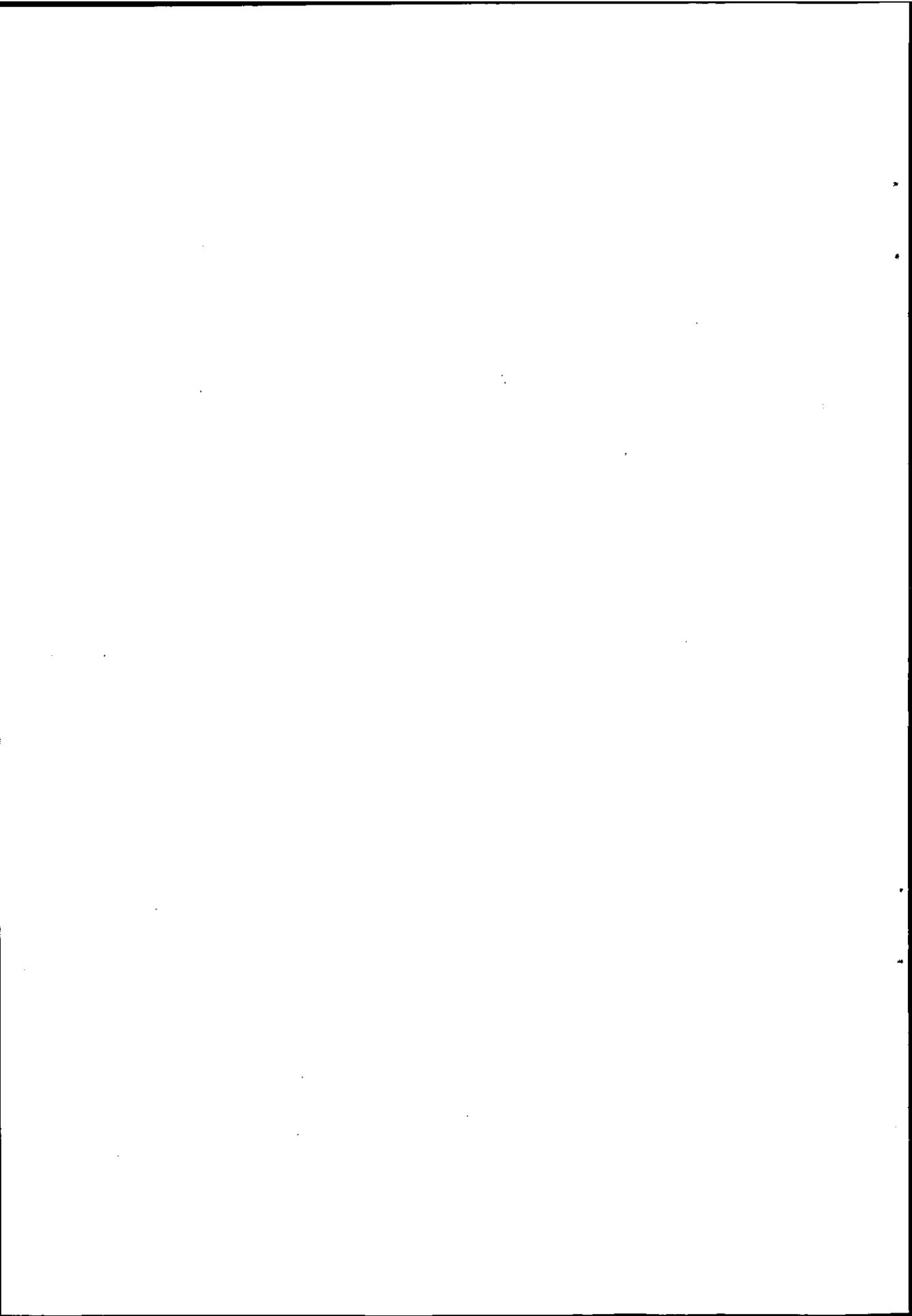
(8) 合理化とはアキラメルことである。合理化とは目的に対する寄与度の少ないものを切ることである。たとえば、資材のA、B、C管理を考えてみても効率の高いところ、20%だけでもすくうと実質的には8割ぐらいの管理ができるのである。アウトプットの場合も、責任権限を明確にした決定者のところへ届けるのが、合理化なのである。つまり、なるべくパスを少なくするということを考えないと、よいシステムはできないのである。

(9) 計算センタの保安および監査上の配慮を設計におりこむこと。保安は非常に大事で、火事を考えなければいけない。次に、監査であるが、設計の当初から監査を入れなければならず、方法は追跡調査的になっており、プログラムそのものを監査したり、特定の監査プログラムをつくり、コンピュータの中へデータを追っかけて歩くという様になってきているのである。

(10) サブ・システム設計上の重点。(a) 経営管理上はアウトプットである。役に立つアウトプット・要点をついているアウトプットが経営管理につながるものであり、アウトプットの設計は非常に重要である。(b) システム運用上はインプット、特にインプットのエラーである。このインプット・エラーを減らす方法としては、システム設計上の配慮の外に機械化移行の際の並行実施期間をいかにうまくマネジメントするかにかかっているのである(短かすぎるのがいけない)。(c) システムの寿命上はコード設計である。経営管理上の要求が変われば当然システムは変わるが、そういうことがなくて、システムの寿命がくるのは、コードのオーバースケールである。長すぎても、短かすぎてもよくない。その辺のところ設計上のポイントになるのである。

第 3 部

シ ス テ ム 設 計



# 目 次

I . 基礎調査段階	156
1. システム設計の目的	156
2. 現状調査	160
3. 現状分析	162
II . 設計推進段階	170
1. 概要設計	170
1-1 システム推進母体の組織化	170
1-2 一般的流れ図の作成	172
1-3 必要な作成報告書の検討と決定	179
1-4 機械化費用の見積りと効果の測定	180
1-5 移行計画の立案、検討	181
1-6 最終の調整と確認	182
2. 詳細設計	182
2-1 コード設計	183
2-2 アウトプット設計	186
2-3 インプット設計	188
2-4 ファイル設計	191
2-5 データチェック	193
2-6 例外処理	196
III . 実施準備段階	197
1. プログラミング	197
2. プログラムテスト	199
3. ドキュメンテーション	200
4. 機械導入のための諸準備	202
5. コンピュータ要員	205
IV . 実施および評価段階	207
1. 実施結果の検討と反省	207
2. 効果とコストの比較	208
3. システムの改善	211

## I 基礎調査段階

### 1. システム設計の目的

システムという言葉は、一般に体系、組織、制度などと訳されているが、いったいシステムとはどんなものをいうのであろうか、システムとは、「互に関連するいくつかの要素によって、ある目的のために構成されているもの」である。つまりあるものがシステムであるためには、1つには、ある目的が与えられていること。もう1つには、いくつかの要素から構成されていること、最後に、それらの要素が互に何らかの関連を持っていること。以上3つの条件が満たされなくてはならない。

それでは、企業システムはどうであろうか。企業は需要者に対して経済的に財貨あるいはサービスを提供し、利潤獲得を目的とする一つのシステムである。そして企業システムも財務活動、生産活動、販売活動のような要素から構成され、それぞれが互に関連をもって動いているのである。

近年、経済社会の国際化に伴い、企業間の競争はますます激化し、一方これに対処するため企業はますます大規模化する傾向にある。この激しい競争を勝ち抜くためには意思決定に必要な情報を迅速に得なければならない。すなわち、情報の遅れは企業に利潤獲得のチャンスを失わせるばかりでなく、場合によっては致命的な打撃を与えかねない。一方、企業の大規模化は情報伝達の経路を長く複雑にし、情報の遅れが生じがちになる。また情報処理（すなわち事務）に携わる人員の増加も企業の利潤を圧迫する要因になってくる。

大規模化した企業の中で意思決定に必要な情報を迅速に得るということは、従来の本とソロバンによる情報処理システムではほとんど不可能である。そこでコンピュータを導入して情報処理を機械化する必要が生じたのである。ここでいう情報処理システムとは、人間が意思決定を行なうために直接役立つ情報を意思決定者に供給するシステムである。

それでは、ここで情報処理システムの設計について考えてみよう。コンピュータのためのシステム設計とは、企業活動に直接必要な情報を収集し、処理し、伝達する方法などのすべてを、コンピュータシステムにマッチした形に設計することである。したがって、システム設計は、コンピュータに直接つながる面

だけでなく、原始伝票を投入する部門における手作業の事務処理方法や、作成された報告書のフィードバックの方法、さらに進んで、報告書に基づく意思決定の部門にまでわたって総合的に設計されるべきである。

それには、

- ① 新システムの目的は何か。
- ② 現状ではどんな問題があるか。
- ③ その解決策は何によって可能か。
- ④ いつまで実施すべきか。
- ⑤ システムライフはどのくらいか

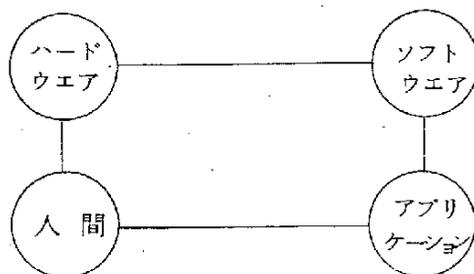
などの点から出発しなければならない。

それからシステム設計を考える上で非常に大事なことは、従来情報処理というのは、手作業で行なっていた仕事を、より情密に、より詳細に分析して、コンピュータにのせていくといった形であったのが最近では、たとえば、コンテナ船のコンテナの配送計画、ジャンボ輸送機による輸送の問題などにみられるように、コンピュータによる情報処理を前提として、技術革新が行なわれている。つまり従来の処理をコンピュータにおきかえたのではなくて、新しい技術革新のときに、すでにその技術革新がコンピュータによる情報処理を要求してきた、それによって解決をはかるうということである。

これから述べるシステム設計は、主に前者について説明することとする。

ではコンピュータ・システムの運営にあたって、システム設計がどんな役割をはたし、どんな位置にあるのかを見よう。

一般に、コンピュータ・システムの運営にあたって、つぎの4本の柱が重要である。



ハードウェアとは、コンピュータそのもの、つまり金物的部分をいい、ソフトウェアとは、そのハードウェアに指示を与えるためのプログラム上のシステムをいう。この二つがコンピュータ・システムの両車輪である。アプリケーションとはコンピュータ・システムに種々の業務をのせ、全体としての情報処理システムを作り上げることで、いわゆる「システム設計」は、この部分の仕事である。人間とは、コンピュータ・システムを動かす人間関係をさすもので、最近とくに重要視されてきた部分である。その中には、システム設計を担当するプランナ、プログラムを担当するプログラマ、コンピュータを操作するオペレータ、その他間接的に接する人達も含まれる。

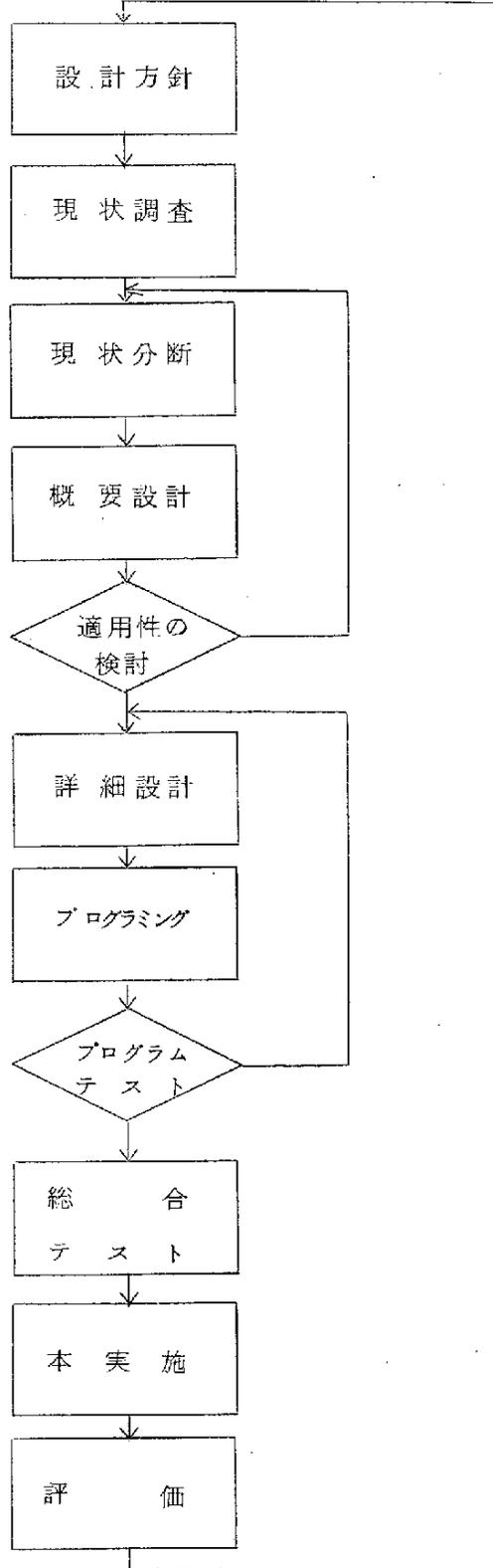
これらの4本の柱は、それぞれ重要な役割をもっており、いずれを除いても、その運営は不可能である。

特にアプリケーションとしてのシステム設計は、コンピュータ・システムが、真に有効な働きをするかどうかを決定する重要な部分である。つまりシステム設計は、いわば「使い方の科学」であり、最適な使い方によって真の効果を発揮させるのがシステム設計の役割である。

まず、新システムの方針を決定し、それに基づいて、現状の調査を行なう。次にその調査結果を分析し、問題点を探し、システムの基本的構造を明らかにする。そして概要設計では、現状調査によって明らかにされたシステムの機能要求にもとづき、コンピュータの導入を前提として、システムの機能向上のための概略的な設計を行なう。ここで、そのシステムの適用性を検討し、OKであれば、コンピュータの性能を考慮しながら、コンピュータで処理できるように細部にわたる設計を行なう。その後、それに基づいて実際にプログラミングをし、テストを行ない、OKであれば総合テストに移って、本実施にはいる。最後にそのシステムの評価を行なうのである。

次にシステム設計の手順を見てみよう。

図1 システム設計の手順



## 2. 現状調査

システム設計にあたって、事前に現状の正確な把握が必要になってくる。つまり現状を詳細に調査し、それを分析して、そこにおける問題点は何か、またシステム設計にあたって必要な条件は何か、などをつかんでおかなければならない。

したがって現状調査の目的として、

- ① 設計の対象となる情報処理システムの基本的な構造の解明
- ② 設計の対象となる情報処理システムの問題点の把握
- ③ 情報処理の手順の解明

があげられる。

それでは、これらの目的を達成するには、どんなことを調査しなければならないであろうか。企業は、複雑な組織、手続に関する諸規定のもとで、必要な情報処理を行なっているばかりでなく、外部条件によっても左右されるのである。

### 1) 外部条件

- ① 経済動向
- ② 業界の背景
- ③ 官庁規則及び法律

### 2) 内部条件

- ① マクロ的見地
  - ① 企業の規模、組織、成長率
  - ② 企業の新製品、新規業種の開発
  - ③ 従業員の生産率とその消長
  - ④ 労使関係
  - ⑤ 関連企業との関係
  - ⑥ 企業の長期計画または経営目標
  - ⑦ 企業経営上の問題点

この段階での調査員は、特に厳選された従業員をあてることが必要であり、多くの場合、管理職員が当るのが普通である。

### ② ミクロ的見地

- ④ 適用業務組織の調査
- ⑤ 適用業務手続の調査
- ⑥ 情報処理のための手順体系の調査
- ⑦ 情報処理システムの効率の調査

まず部門間の情報処理業務の結びつきを組織的に把握するために、業務組織の調査を行なう。業務組織の調査が終わったら、情報がどのような手続きによってどのように処理されているか、そしてどの部門からどの部門に流れているか等業務手続の調査を行ない、システム設計に必要な基本事項を明確にする。

そして業務手続の調査によって、各業務における情報の流通径路および流通に要する時間、また情報の質、量、加工方法などがわかり、処理手順および判断基準などの情報処理システムの設計に必要な基本事項が明確になる。次に情報処理のための手順体系の調査を行なうが、現状の情報処理システムの問題点を把握することなしにシステムの改善をすることはできないので、情報処理のための手順体系の調査は、システム設計を行なうとき非常に重要になってくる。情報処理のための手順体系の調査が終わったならば、ある部門で必要とする情報が、他の部門から効率よく伝達されているかどうかを調査するために、情報処理システムの効率の調査を行なう。

それでは、次に調査方法についてみてみよう。

### 1) 面接法

これは、直接各担当者に面接して、各調査事項についての資料を得る方法である。この方法は、プランナーが被面接者に信頼されている場合には、もっとも効率的な方法である。この場合、面接は、経営層のトップからはじめて、ミドル・ロワー、担当者という順に行なうのがよい。

### 2) 質問紙法

これは、いろいろな調査事項を書いてある質問紙を配布して、調査を行なう方法である。質問紙に書く質問は、その解答を容易に分類できるように工夫しておくことが大切である。

### 3) 観測法

これは情報処理システムで行なわれる動作を観察、観測し、情報処理に必要な作業を知る方法であり、作業量を測定する場合にもよく用いられる。

#### 4) 資料法

これは経理規程、文書作成規則など企業における情報処理に関する諸資料を調査し、情報処理システムの構造、処理手順などを知る方法である。また社外の情報源には業界刊行物、官庁統計、仲買報告書、信用調査書などがある。さらにもっと略式の情報源としては、原料供給者、顧客、保険契約者、預金者などがある。

### 3. 現状分析

現状分析とは、現状調査によって得られた情報をまとめ分析することによって新システムの設計方針を決定することである。

現状分析の目的として、

- 1) 現状調査によって得た知識を整理することによって、現在の情報処理システムの問題点を正確に把握することができる。
- 2) 現状調査によって得た知識を体系的にまとめておくことにより、対象企業の現在の情報処理システムの問題点が明確になり、新システムを設計する場合の基礎条件になる。
- 3) 現状調査によって得た知識を体系的にまとめることによって、各業務や各情報相互間のつながりを明確にすることができ、また現状調査における調査もれを発見することができる。

次に現状調査で得た知識を体系的にまとめるためには、現状調査によって得た資料を分析しなければならない。その方法としては以下のものがあげられる。

#### 1) 組織分析

企業組織は製造、営業、経理、財務、人事、庶務などの各单位組織から構成されている。組織分析は、各单位組織で発生する情報の関連性と、各单位組織が果たしている機能を知るために行なり。

組織分析では、現状調査によって得た知識をもとに、十分な組織上の分析を行ない、問題点をもれなく把握しなければならない。組織分析では、

通常つぎの項目について分析を行なう。

- ① 企業組織が各単位組織に要求している機能は何か。
- ② 各単位組織の機能には重複あるいはモレがないか。
- ③ 各単位組織における機能はどのようなつながりをもっているか。
- ④ 各単位組織は、与えられた機能をどの程度果たしているか。

## 2) 職務分析

各組織における職務を明確にするため、各職務の要点を記述する資料を分析し、確認し、報告する過程である。この資料には通常4種類ある。

- ① 職務記述書……行なうべき作業の簡単なリスト
- ② 職務明細書……行なうべき作業を列挙したリスト
- ③ 職務資格要件…おのおのの職務に必要な特性
- ④ その他の分析事項…各職場に含まれる機能相互間の関係の分析

各職務に関する4種類の資料にもとづいて行なわれた職務分析によって、職務のモレ・重複などをなくし、職務の合理的な再編成を行なう。

## 3) 業務分析

主要業務における単位業務の種類にわかれ、単位業務における事務手続の流れ、必要な技能経験、用いられている伝票・報告書などを明確に確認する。その目的は3つにまとめられる。

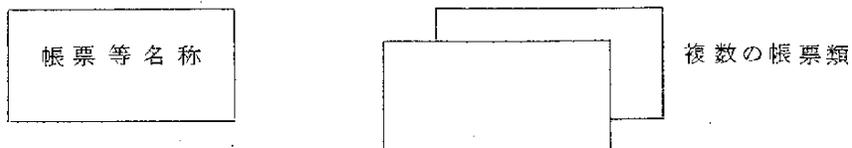
- ① 事務手続の流れの明確化
- ② 必要な技能経験の明確化
- ③ 伝票・報告書の明確化

業務分析のまとめ方として、伝票・報告書作成などの事務作業における、それぞれの情報処理を記号であらわしたものがある。これを事務処理現状分析図（事務のフローチャート）という。

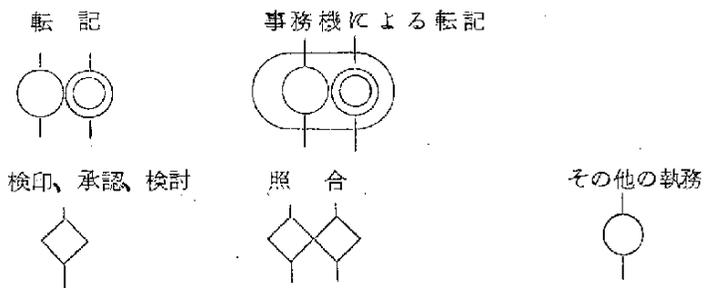
ここではNOM A式（日本事務能率協会）を参考に説明する。

図2 事務のフローチャート記号（NOMA式）

① 帳票記号



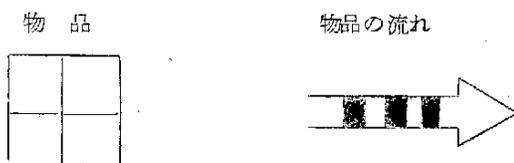
② 作業記号



③ 電話記号



④ 物品記号



⑤ その他

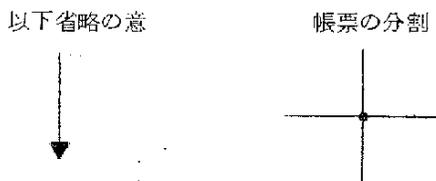
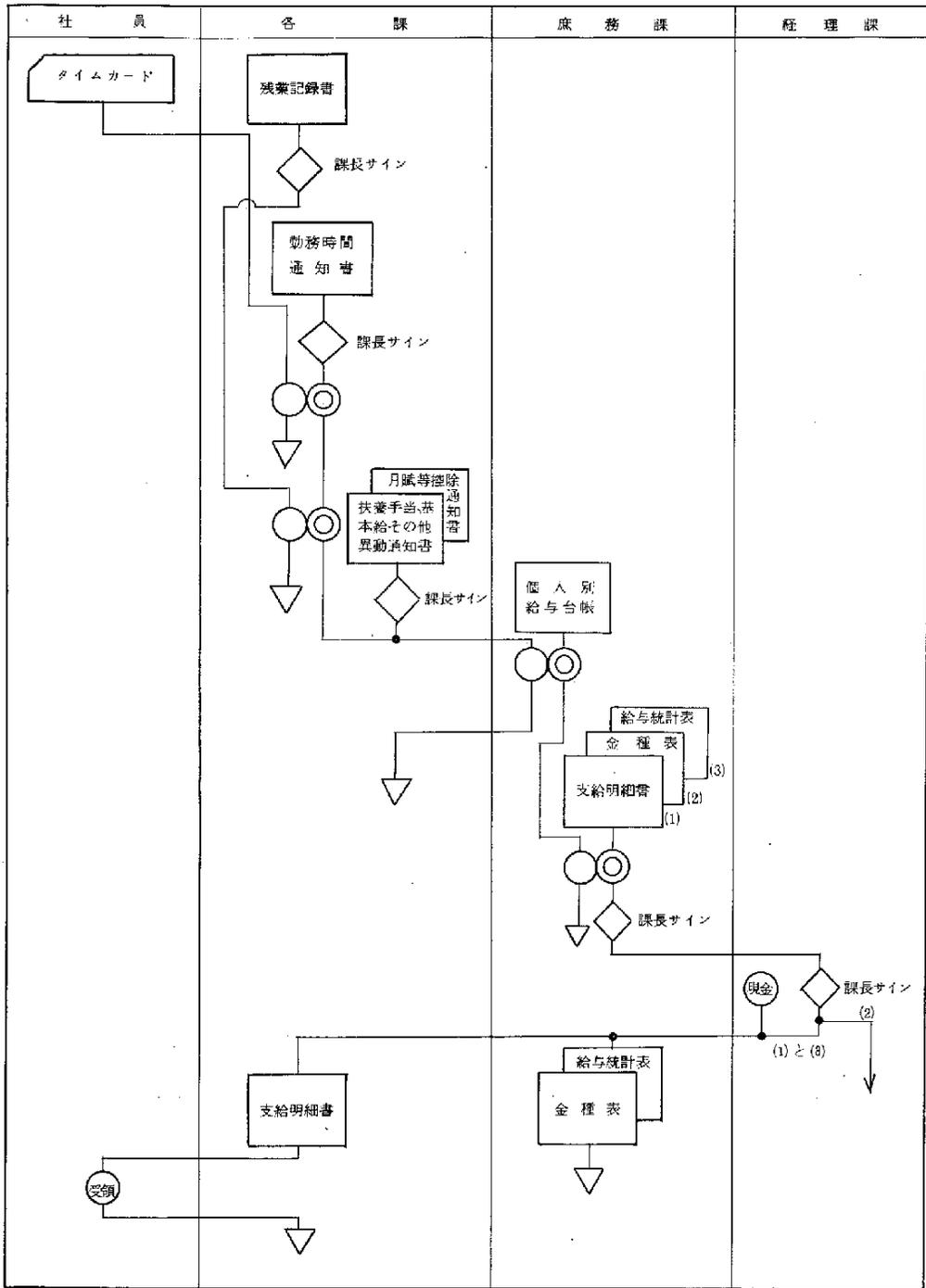


図3 給与計算事務フローチャートの実例



#### 4) 帳票分析

企業内外の環境が複雑化すればするほど、企業内において発生する情報は増大し、情報の媒体として使われる伝票・報告書などの帳票の管理が重要になってくる。この情報の増大とともにふえてきた帳票の管理を再検討するために、帳票の簡素化、標準化を目的として、帳票の処理過程、組織上の経路、項目などを明確にしなければならない。このように現状の情報処理システムで用いられている帳票の簡素化・標準化を目的として、各帳票の処理過程および関連を明確にするのが帳票分析である。

帳票分析を行なう場合、つぎの2つの面から分析していく。

- ① 帳票の処理過程の分析
- ② 帳票の記入項目についての分析

帳票の処理過程で、不明確な点があれば、実際の帳票処理の担当者から助言を得て明確にする。また各帳票の記入項目について分析するときには、単に個々の帳票を見てもはっきりしないので、入力と出力という関係をとらえる。これを I/O 分析という。

I/O 分析をすすむことによって、項目のモレや重複を発見すると同時に、必要のない項目を発見したり、あるいは今後必要と考えられる項目を発見する基礎資料が得られる。

#### 5) 手順分析

組織分析、職務分析、業務分析、帳票分析によって得た資料にもとづき、事務の細部手順を分析することによって、論理的処理がなされているかどうかを明確にする。

細部手順を記述する場合、一般につぎのことに注意しなければならない。

- ① たとえ以前に制定した規定・手続があっても、分析にあたっては現状において実際に行なわれている手順を記述すること。
- ② 記述するときの表現方法は簡単かつ具体的であること。
- ③ 使用する用語はできるかぎり簡明なものとし、全社的統一をはかること。
- ④ 使用する用語があいまいなものは、明確に定義づけること。
- ⑤ なるべく箇条書きの方針をとること。

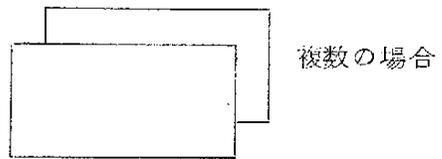
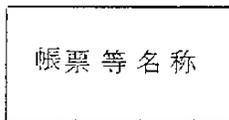
- ⑥ 手順の書き方、使用する器具、用紙などについても明確な規定をすること。
- ⑦ 流れ図を利用するときには、その書き方、用いる記号などを明確にし、全社的統一をはかること。

細部手順を明確に文書化する方法として、ブロックダイアグラムによる方法がある。ここではNOMA式を説明する。

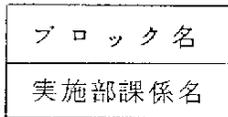
○ 基本記号

① 帳票記号

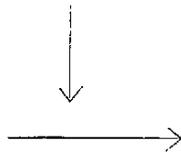
帳票、帳簿、報告書類



② 事務処理ブロック記号



③ 方向表示記号



6) 事務量測定

個々の事務作業の所要時間を測定し、またその繁閑の程度を調査することによって、現在の情報処理システムにおける問題点を明らかにするのが事務量測定の目的である。

事務量測定方法には、つぎの方法がある。

① 見積法

個々の事務作業の担当者の報告にもとづいて、所要時間を見積る方法である。

② 実測法

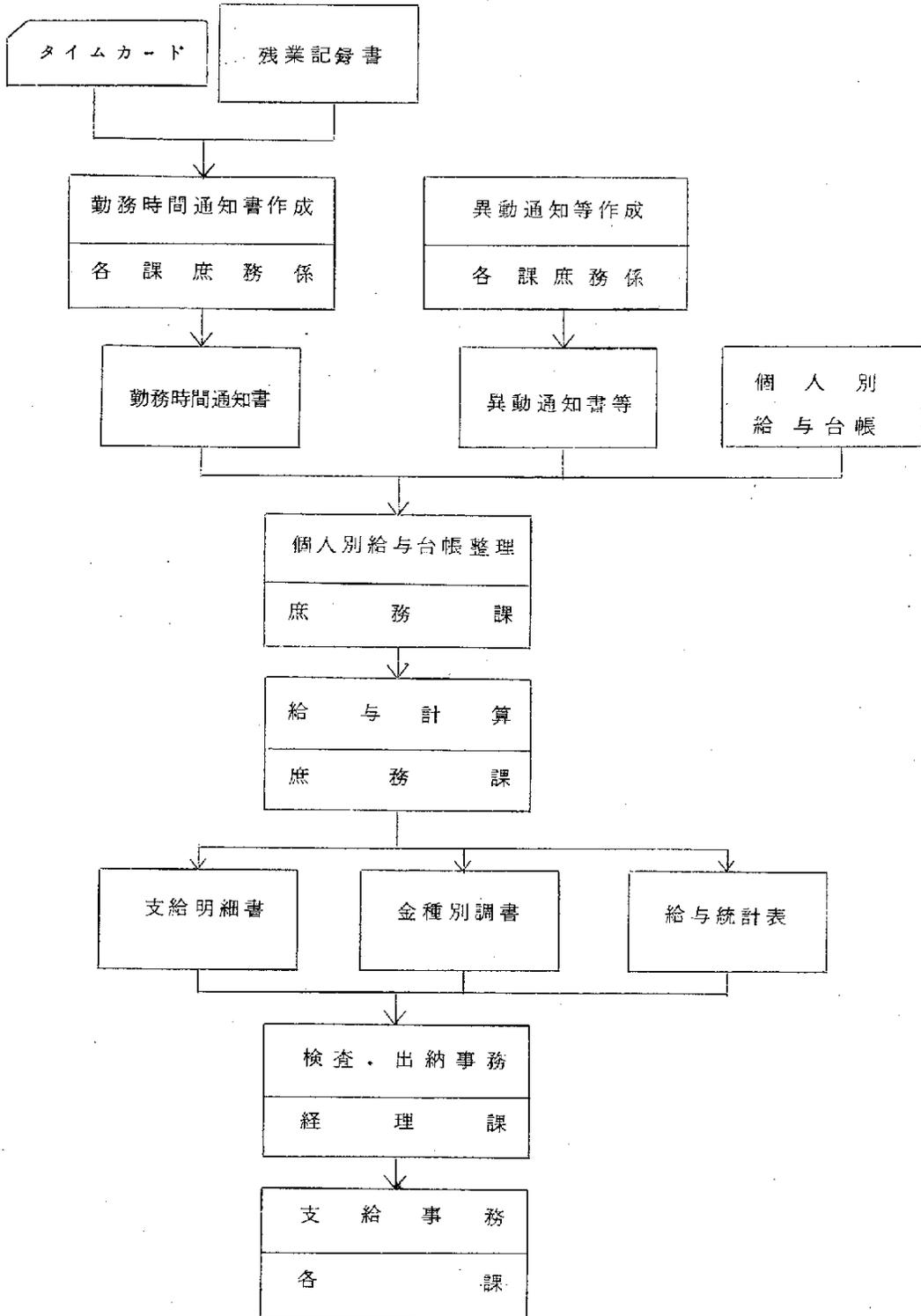
個々の事務作業の担当者のそばで、ストップ・ウォッチなどにより、

その作業の所要時間を測る方法である。

③ 既定時間法

事務量を測定しようとする事務作業を動作要素に分解し、事務作業に必要な動作にたいして、決めておいた時間をあてはめ、その作業の所要時間を求める方法である。

図4 プロツクダイアグラムの実例



## Ⅱ 設計推進段階

### 1 概要設計

#### 1-1 システム推進母体の組織化

まず新システムを設計するにあたって、最初に考えなければならないことは、このシステムを実施するための推進母体をどのように組織化するかということである。つまり、そのシステム推進母体を形成する各々のメンバーをどのように結びつけ、相互の関連性、一貫性、総合性というものをどのような形でまとめて1つのシステムをつくり上げていくか、ということが問題になってくるのである。

一般には次の3通りに分けられる。

#### 1) 専門部課による推進

専門部課の専門担当者（システム・プランナー）が、調査分析改善案の作成から、電子計算組織導入稼働までのすべてについて、関係者の協力を得ながら中心的役割をはたす方法である。

#### ○ 長所

- ① 専門家の集まりであるから、意思の統一がとれる。
- ② 短期間でまとまりがつく。

#### ○ 短所

- ① 現場の実務家の参画が取り入れにくい。
- ② 社内の抵抗、急激な変化に対する反応が起こりやすい。
- ③ システム・プランナーがその企業に精通することが困難である。

#### 2) 委員会制度による推進

通常の職制活動に籍をおきながらシステム設計の推進に参加する方法で、システム・プランナーは単に、メンバー、アドバイザー、事務局として参加することが多い。

#### ○ 長所

- ① 各実務現場の意見が取り入れられる。
- ② それぞれの現場の参画意識が高まるので、社内の盛り上りが得られる。

○ 短所

- ① 各員が全体の利益、総合的な目的を考えないで、部門の利益代表に終わってしまう。
- ② 時間がかかるおそれがある。
- ③ 委員長に強力なリーダーシップが要求される。

委員会制度による推進というのは、時間的な要素、決断力という面からみて、必ずしも日本の企業では成功していない。

3) プロジェクト・チームによる推進

関連部門より適任者をえらび、システムの専門家、業務の精通者が一団となり、同一目的をもつチームを編成し、システムが完成するまでは、現業からはなれて、システムの設計・推進に専従する方法である。

○ 長所

- ① きわめて能率的である。
- ② 意思の統一がとれる。
- ③ 目標を見失うことが少ない。

○ 短所

- ① 優秀な人材を各部門から相当期間にわたって専従させるために、各部門からの抵抗が大きい。
- ② 特に日本の場合、プロジェクトチーム制に慣れていない。

これら組織の編成方法は、それぞれの企業の体質、新システムの性格にあわせて考えていかななくてはならない。

その場合特に留意しなくてはならない点は、トップマネジメントより公式に任命し、社内に広く公示すべきこと、任務、責任、行動範囲を明確にしておくこと、対象業務の部門が常に中心であることの自覚と認識を持つこと等である。つまり販売システムをつくる場合、コンピュータの企画室がつくっているのではなくて、営業部門が自分の仕事として考えるという自覚と認識を常に特たせておく必要がある。このようにしていろいろなシステムの内容のより詳細な検討、立案が行なわれていくのである。

その場合の内容としては。

- ① 基礎調査の結果の再検討
  - ② 現場のより詳細な分析
  - ③ 改善しなければならない問題点の完全なる把握
  - ④ 改善すべき方法の解決策（たとえば制度、手続、組織、経費）
  - ⑤ 新システムにもりこむべき内容の検討、優先順位の設定
- などがあげられる。

このようなプロジェクトチームという考え方が、従来の日本の企業の縦割り社会にどの程度定着していくかというのが、これからの課題であろう。

## 1-2 一般的流れ図の作成

基本的にはいろいろなシステム設計を行なうにあたり、どのような複雑なシステムに対しても、仕事の流れを分析しておく必要がある。

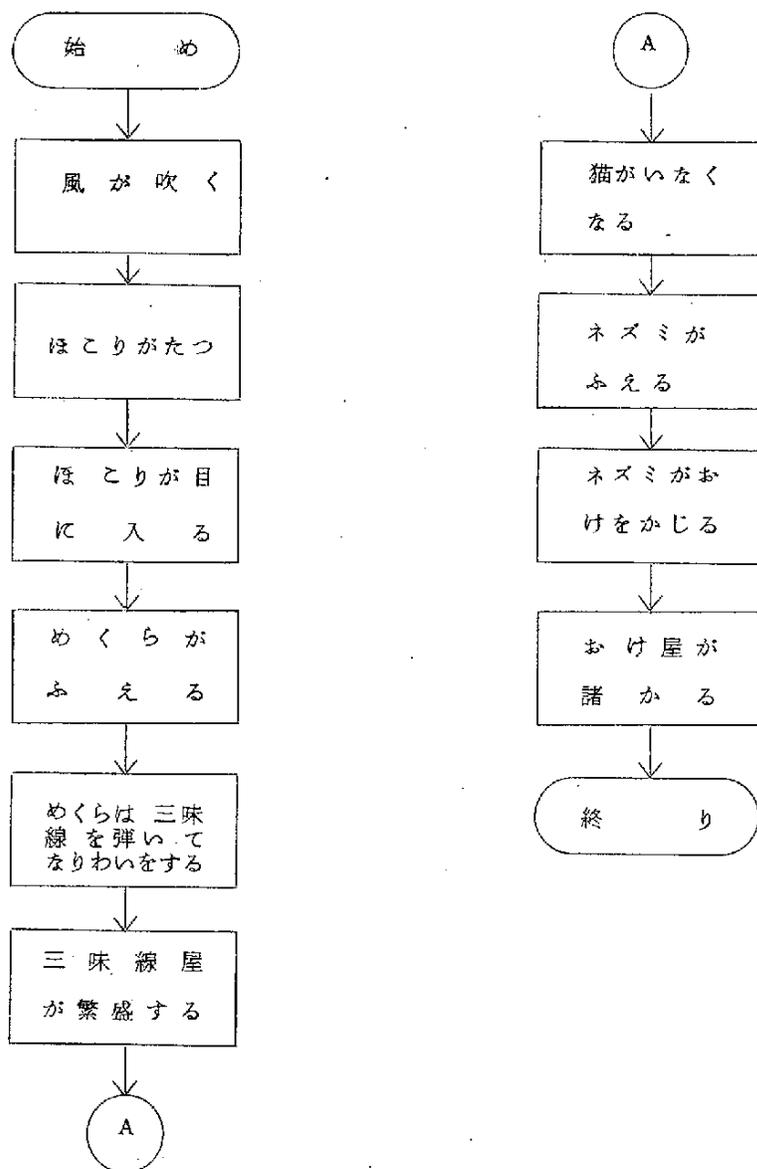
この仕事の中には、仕事の手順、仕事の手順によって処理されるデータの2つがある。つまり給与計算という仕事には、給与計算をすための手順（A社給与計算規則）、給与計算される各人のデータ（出勤日数、残業時間、遅刻の回数）がある。このように仕事は、手順とデータからできているが、その中でデータがどのように処理されるかという手続の部分だけを、できるだけわかりやすく書いたものが流れ図（フローチャート）である。たとえば「風が吹けばおけ屋が儲かる」の流れ図の実例は図5のようになる。

このように、手順を最もわかりやすい図解の形で、しかも誰が読んでも間違わないように客観化して図示しておく、この場合、流れ図の中には、時間的あるいは数量的また形式的な要素が含まれる場合もある。

このような流れ図で分析しておく、誰が見ても流れを追いかけることができ、客観化される。またトップマネジメントや関連部門の代表者に説明する場合、視覚に訴えて説明することで非常にわかりやすい。つまり、これから設計していこうとするシステムの一般的な流れ図というものを作成し、さらに新しい方式と古い方式とのシステムについてそれぞれを作成し、いかに改善の効果が上がるかということをややすく比較するような方法に訴えると非常にトップに説明しやすいわけである。

流れ図というのは前後関係、これは時間的な関係も表わすと同時に、数量的な関係を表わすこともできる。これはしばしば仕事の分析とか定量的な関係で

図5 流れ図の実例



しか使われないと考えがちであるが、量や計数的な関係だけではなくて、定性的な流れを表わす場合にも、用いられる。たとえば、南米で飢饉が起きたらそれが国際農業機構をどう動かして、それが日本の農業にどう影響を及ぼすか、そしてそれが日本の肥料産業にどのような影響を及ぼすかという問題はあくまで定性的なとらえ方であり、この定性的な流れ図を見ても、何がどのように刺激するかという、刺激する源はわかるが、それによって南米の農業生産が10%落ちたから、日本の肥料工業に対して何%の影響があるというような量

的な結びつき方はわからない。定性的な流れ図というのは、政治の問題とか社会工学的な問題を取り上げる場合に多く作られる。

一般に、企業、団体、役所などで考えられる普通のシステムの場合には、前の計算ででてきた数値をもってきて、それをどのようにして次にもっていくかという定量的な流れ図である。普通の場合は、この流れ図の中には全て計数的な記号とか要素（給与計算のやり方とか、販売方式とか、技術計算など）が含まれてこなければならない。

ここでいう一般的流れ図は、むしろ全体の流れをとらえるので、定量的というよりは、やや定性的な流れ図になる。つまり、仕事のプロセスの関連がどういうつながりになっているか、ということをつまみにわからせるためのマクロ的な表わし方であるから、定量的というほどこまかくなくて、定性的なものになるのであろう。

フローチャート（流れ図）の種類には(1)コンピュータにどのようなプロセスで仕事をさせるかを図示したプロセス・フローチャート、(2)各プロセスごとにコンピュータの内部で、どのような方法によって処理を行なうかを図示したゼネラル・フローチャート、(3)プログラミングに使われるディテール・フローチャートがある。

プロセス・フローチャート、ゼネラル・フローチャート・ディテール・フローチャートの例（これは、商品コード、売上数量、商品単価がパンチされているカードファイルを入力することにより、商品別の売上げを計算する流れ図である。）

図6 プロセス・フローチャート

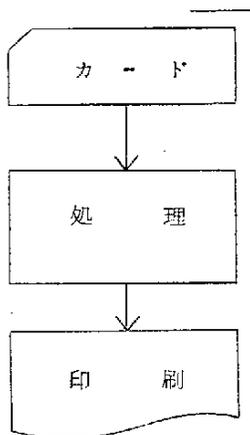


図7 ゼネラル フローチャート

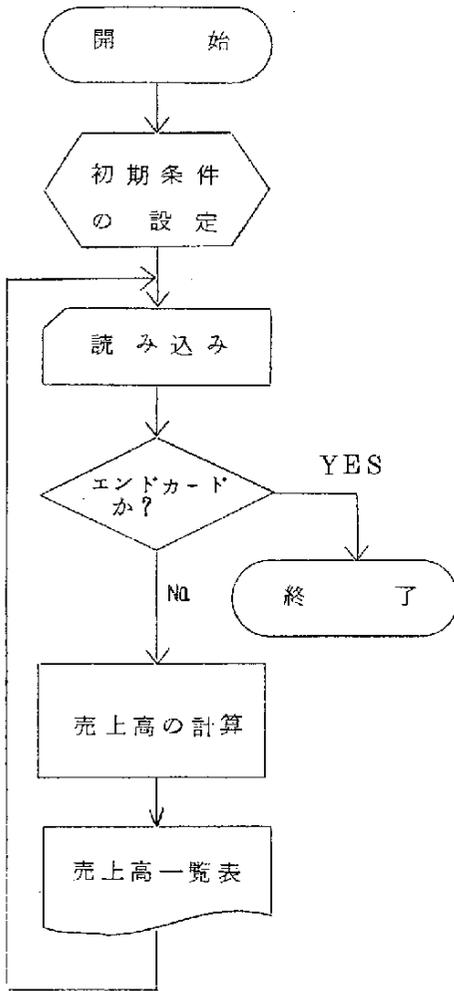
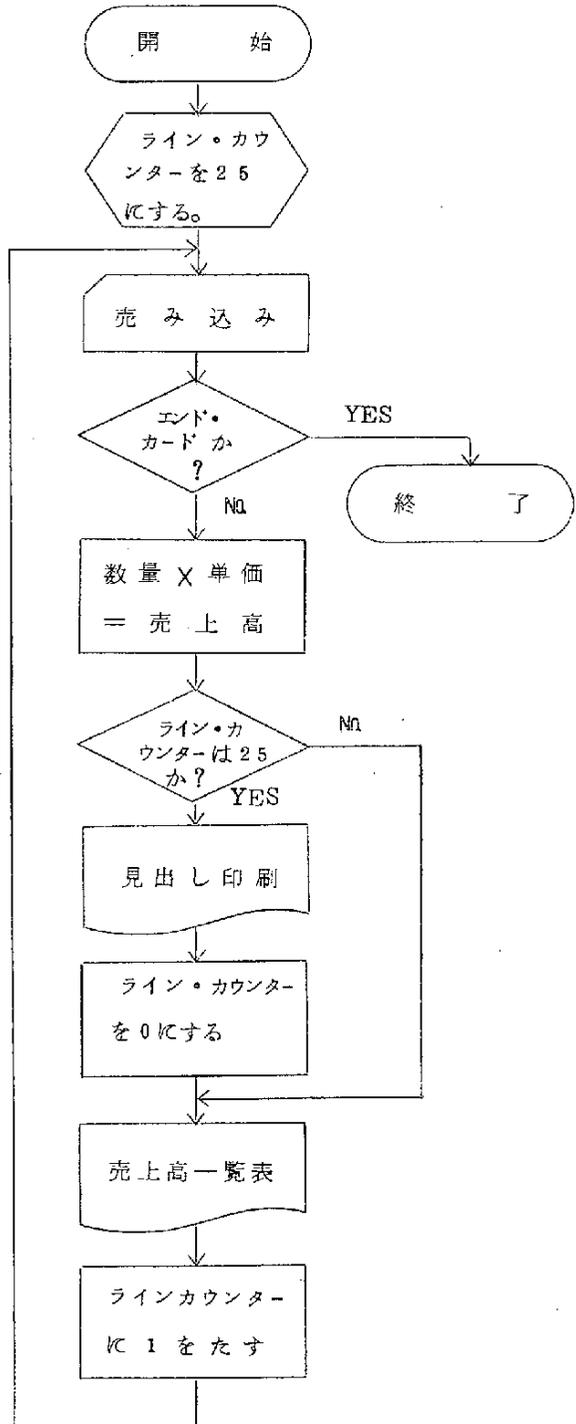
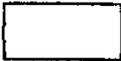
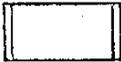
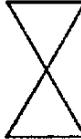
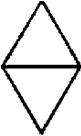
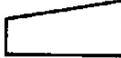
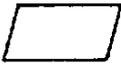
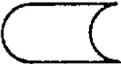
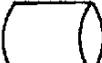
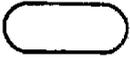


図8 デイテールフローチャート



番号	記号	意味
1		処 理 (process) あらゆる種類の処理機能を表わす。
2		判 断 (decision) いくつかの択一的経路のうち、どの経路をとらせるかを定める判断、またはスイッチ式の操作を表わす。
3		準 備 (preparation) スイッチの設定、指標レジスタの変更、ルーチンの初期値の設定など、プログラム自身を変えるための命令、または命令群を修飾する機能を表わす。
4		定義済み処理 (predefined process) サブルーチンなど、別の場所で定義されている命令群、またはいくつかの操作からなる命名された処理過程を表わす。
5		手 作 業 (manual operation) 機械的な手段を用いない手作業によるオフラインの処理過程を表わす。
6		補助操作 (auxiliary operation) 中央処理装置で直接に制御されていない装置によって行なうオフライン操作を表わす。
7		組 合 せ (merge) 記録の二つ以上の集合を一つの集合に組み合わせることを表わす。
8		抜 出 し (extract) いくつかの記録からなる一つの集合のなかから、特定のいくつかの集合を取り分けることを表わす。
9		照 合 (collate) 組み合わせたり抜き出ししたりしながら、記録の二つ以上の集合から二つ以上の集合を作り出すことを表わす。
10		分 類 (sort) 集合中の記録をある定められた順序に並べかえることを表わす。
11		手操作入力 (manual input) オンラインけん整、スイッチ、押しボタンなどを手で操作して、処理中に情報を入れる入力機能を表わす。

番号	記号	意味
12		入出力 (input/output) 情報を処理可能にする入力機能, または処理済みの情報を記録する出力機能を表わす。
13		オンライン記憶 (online storage) 磁気テープ, 磁気ドラム, 磁気ディスクなど, あるいは他の種類のオンライン記憶を利用する入出力機能を表わす。
14		オフライン記憶 (offline storage) 情報を任意のオフラインの記憶媒体に記憶する機能を表わす。
15		書類 (document) 書類を媒体とする入出力機能を表わす。
16		紙カード (punched card) 紙カード (マークセンスカード, パーシャルカード, スタブカード, マークスキャンカードなどを除く) を媒体とする入出力機能を表わす。
17		カードの組 (deck of cards) 紙カードの集まりを表わす。
18		カードのファイル (file of cards) 紙カード上の関連ある記録の集まりを表わす。
19		紙テープ (punched tape) 紙テープを媒体とする入出力機能を表わす。
20		磁気テープ (magnetic tape) 磁気テープを媒体とする入出力機能を表わす。
21		磁気ドラム (magnetic drum) 磁気ドラムを媒体とする入出力機能を表わす。
22		磁気ディスク (magnetic disk) 磁気ディスクを媒体とする入出力機能を表わす。
23		磁心 (core) 磁心を媒体とする入出力機能を表わす。

番号	記号	意味
24		表示 (display) 処理回路、オンライン表示機、映像表示装置、操作車の印字機、作図機などを通じて、情報を人間が利用できるように表示する入出力機能を表わす。
25		流れ線 (flow line) 記号を結びつける機能を表わす。
26		並行処理 (parallel mode) 二つ以上の同時操作の始まり、または終りを表わす (左端には流れ線は示していない)。
27		通信 (communication link) 情報を通信線で伝送する機能を表わす。
28		結合子 (connector) 流れ図のほかの場所への出口、あるいはほかの場所からの入口を表わす。
29		端子 (terminal, interrupt) 開始、終了、停止、遅延、中断など、流れ図の端子を表わす。
30		注釈 (comment, annotation) 明りょうにするために、説明または注意を加える機能を表わす。

### 1-3 必要な作成報告書の検討と決定

コンピュータの役目には、「大量の事務処理」というのがある。これは領収書とか請求書の作成ということで事務そのものが生産であるというような場合、コンピュータは印刷機のように大量の書類をつくっていく。コンピュータを「印刷する機械」であるというように使い方に対しては批判があるが、コンピュータのもつ目的は、大量の事務処理、高度の計数的な数理計画、経営科学を駆使して長期計画をつくること等いくつかの段階に分けられるわけで、大量の事務処理をする場合には、コンピュータはプリンター的に使われていいし、それが最も経済的な使われ方である。たとえば磁気テープやカードに収められたデータのファイルをコンピュータで処理して宛名をつくったりプリントする場合、人手で台帳から転記するのと比較してみるとはるかに前者の方が時間的にも経済的にも効率的である。最近では、コンピュータを使って百科辞典を作るというような方法も考えられている。つまり、コンピュータに自動的に編集させるという機能をもたせる。30行打ったら次へ、また30行打ったら次へ、というように行なりわけであるが、これを人間が組版すると非常に手間がかかる。このようなことを考え合せると、ますますコンピュータが持つプリンター的な性格とか要素は重要性を増してくるのである。

そこで考えなくてはならないことは、それらのコンピュータで打ち出されてくる報告書は、それぞれ経営レベルに応じて作られなければならないということである。

たとえば、在庫管理の場合を考えてみると、在庫管理というのは一定の基準で考えられた限界保有量を、一定の期間保持することが目的であって、その場合全品目をプリントアウトする必要はないのである。場合によっては、年に1度ぐらいしか蔵出しの行なわれない品目もあるはずで、そのような品目を毎日、毎週リストを出したところで意味はない。このような場合、むしろ足りなくなった品物、あるいはあり余っている品物についてだけウォーニングリスト（警告リスト）を出すだけでよいのである。

また人事管理の場合においても、ある一定の人事管理のための基準を越えるような人だけをリストアップするだけで事が足りるかもしれない。全従業員を打ち出す必要はないのである。このような意味で作成報告書の内容の検討

とか決定の場合に、一体その報告書は誰が見るのか、また何のために、いつ必要かということが重要になってくるのである。

そこで作成報告書の留意点として、

- 1) 管理上の区分、項目等を充分検討すること。
- 2) 従来の作成報告書の存廃をきめること。
- 3) 作成報告書の内容の統合化をはかること。
- 4) 作成報告書の経営レベルをきめること。
- 5) 表現方法、表示方法をよく検討すること。
- 6) 報告書の内容は現場実務者にはよく理解されなければならない。

などがあげられる。

#### 1-4 機械化費用の見積りと効果の測定

これはシステム設計の段階でも特に見落されやすい問題で、やはりある締めくくりにおいては、経済性というもの、「ミニマックス」、つまり最小の経費と最大の効果ということを目途に考えなくてはならない。そこで問題になってくるのは、機械化した場合の発生効果の評価をどうするか、ということである。これは、正確に機械化されるべき仕事から、人員の節減数、被代替機械の節約、その他金額換算可能な効果のみを正確に事前に把握すべきことである。この正確な金額をもって予測し得る利益を、タンジブル・プロフィットという。要するに、投下した資本と投下した経費による効果（金額に換算可能な効果）だけで、プラス・マイナスを考えるべきである。これは非常に困難なことであるが、ともすると情報処理の場合は、あいまいな表現、中途はんばな手続きで終わってしまうので、この点を特に注意したい。

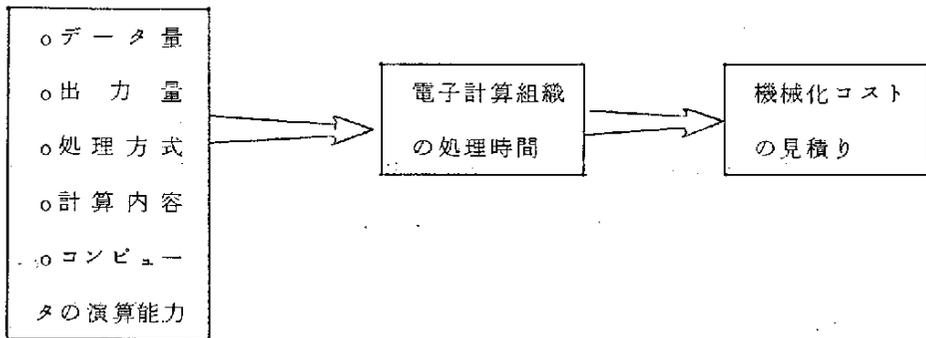
機械化の期待効果は、作成資料を判断する経営者管理部門の責任者によって作り出されるものであるから、コンピュータ部門の責任者は、あまり事前に吹聴してはいけな。むしろそうすることによってコンピュータ部門の信頼度はますます上がってくるのである。

機械化コストの事前把握と、機械化効果の事前把握とは、システム設計のしめくりである。往々にしてこのことが充分に分折されないままに、機械化に邁進してしまう怒れがあるので、この点も注意しなければならない。

最後に重要なことは、実際に機械化してから、実際コストと実際効果と比較

対比して、予測と実際とを一致させていくような努力を、常に払わなければならないということである。

ここで機械コストの見積りのファクターを上げておこう。



#### 1-5 移行計画の立案・検討

移行とは旧システムから新システムへ移ることであるが、これには、移行の日程計画をたて、これに従ってファイル作成の準備、業務実施要領の作成、物品の調達、業務指導などの移行に必要な準備が含まれる。ここではそれらの移行の形態について述べる。

移行には次の三つの形態がある。

- ① 手作業（会計機も含む）→電子計算化
- ② PCDP（穿孔カード機械方式）→電子計算化
- ③ 電子計算組織→より高度の電子計算化

①及び②の形態、つまりいままで手で計算していたものをコンピュータにのせるとか、PCDP（PCS）のものをコンピュータにのせるといった場合には、手作業のスピードあるいはPCDPの穿孔カード方式のスピードと、コンピュータのスピードの差に、非常に大きなギャップがあるので、並行的な作業をしていても、その処理の時間のタイミングの差で移行を容易に行なうことができる。

しかし③の型体の電子計算組織からさらに高度のコンピュータシステムに移る場合は、他の二つの形態よりも容易に見えるが、これが一番困難なのである。というのは、前の二者よりも移行時に非常に大きな危険が伴う。もう

一つは、いわゆる設備投資の面でも、コンピュータを2台同時に置いて、漸次、切りかえていかななくてはならないという経済的な負担がある。この経済的負担というのは、コンピュータの値段だけの問題でなく、スペースの問題、人の問題も含まれてくるのである。

このように移行の問題は、それぞれの形態に応じて慎重に考えなくてはならない。

#### 1-6 最終の調整と確認

この段階では、まずこれまでの機械化システムについてのいろいろな資料をドキュメントしておかなくてはならない。

##### — 目 的 —

- ① 経営者に説明するため
- ② 各部門の関係者の理解のより所とするため
- ③ 各部門の担当者の諒解事項とするため
- ④ 機械化推進の原典となるため（プログラムのより所となる）
- ⑤ システム設計の詳細記録として保管するため
- ⑥ 後継者の教育用とするため

このようにシステムの概要をドキュメントしておいて、最終的には関連部門との確認を行ない、公的な会社の承認を取りつけておく必要がある。これをおろそかにすると、実施の段階において、各部門からのいろいろな抵抗を受けることにもなる。

##### — 内 容 —

- ① 前途の機械化システム資料に基づく最終確認
- ② 必要ならば一部の手直し
- ③ 関連部門責任者に対する説明会
- ④ 会社としての承認（トップマネジメントの決裁）

## 2. 詳細設計

現状調査、分析および概要設計が終ると、詳細設計にはいる。詳細設計とは、概要設計で明確にされた構想を具体化して、プログラミングができるようにす

るとともに、残置する手作業についても具体的な処理方法を設計し、対象業務全体が円滑に機械化できるようにすることである。したがって、その内容については抽象的表現を避け、詳細かつ具体的でなければならない。

## 2-1 コード設計

コードとは、“もの”あるいは“ことがら”をあらわす符号や記号のことをいう。われわれが日常使っている言葉をコンピュータに直接理解させることはできない。そこで、情報処理のために必要な分類項目などの重要な項目をコード化し、コンピュータが容易に理解できるようにすることが必要になってくるのである。それ故、コード設計の良し悪しは、情報処理の機械化の成否に大きな影響をもたらすのである。

次にコード体系の条件を考えてみよう。

経営において、各業務は何らかの関係を持っている。したがって、コードも複数の業務に関連をもつこととなり、コード体系は複数の業務に対して共通に適用しうるように統一性をもっていなければならない。またコードのケタ数が長いと、記入ミス、パンチミスなど誤りが生じやすくなる。それ故、コードのケタ数は、なるべく短く簡潔でなければならない。それから事業の拡張や新商品の開発などの理由により、コードをすべて変更していたのでは混乱するばかりである。そこでコードとコードの間に空き番号をもうけるなどしてコードに変更可能性をもたせることが必要である。また、コードは、各々、それによって表現されているものに確実に対応していかなければならない。たとえば、1つの商品には1つのコードしかつけてはならない。これを識別可能性という。また、情報処理の目的のためには、個々のものが識別可能であるばかりでなく、同じ性質を持った一群のものか他のものか識別できるようなコードをつけることが必要である。最後に、現在使われているコンピュータは、漢字や特殊な記号を解読するための入力装置をもっていないのが普通である。それ故、数字や英字のようにコンピュータの機械的な制約に適合したコードを使用することが大切である。

以上をまとめると、

- ① 統一性
- ② 簡潔性

- ③ 変更可能性
- ④ 識別可能性
- ⑤ 機械に対する適合性

となる。

それでは次にコードの種類を見てみよう。

(1) 順番(連番)コード (Sequence Code)

これは、与えられた項目を一定の基準で配列し、これに1から連続番号をつけていくものである。

たとえば、

- 0 1 北 海 道
- 0 2 青 森
- 0 3 秋 田
- 0 4 山 形
- 0 5 岩 手
- 0 6 宮 城
- 0 7 福 島

これは、簡単で明解であるが、分類機能がなく、変更可能性がないので、項目数が比較的少なく、項目の内容が長期間にわたって不変であることが予想されるときに有効である。

(2) 区分コード (Block Code)

これは、順番コードに分類機能をもたせた方式で、正順に並ぶ数値をいくつかのブロックに分け、各ブロックにより項目の分類区分を示す。

たとえば、

- 0 1 ~ 2 9 農 林 水 産
- 3 0 ~ 5 9 鉱 業
- 6 0 ~ 8 9 食 料 品

この方式は、少ない桁数で多くのグループを表わすことができ、後で追加が生じるおそれのある場合には、グループごとに空き番号をもうけることによって、一定の制限内ではあるが追加可能である。

### (3) 桁別コード (Group Classification Code)

コードの各桁にそれぞれ分類上の意味を持たせ、順次連続番号をふっていく。  
たとえば、

1 0 0	文 科 系	2 0 0	理 科 系
1 1 1	法 律	2 1 1	化 学
1 1 2	政 治	2 1 2	物 理
1 2 3	経 済	2 2 3	機 械
1 2 4	経 営	2 3 4	薬 学

これは、各桁が分類上のグループを示すのでわかりやすく、機械処理に非常に適しているが、一般に桁数が多くなる。

### (4) 10進コード (Decimal Code)

これは、図書の分類に使われ、桁別コードの各分類をあらわすために、10進法の数字を用いる。

- 0、 一般事項
- 1、 哲 学
- 2、 社会科学
- 3、 自然科学
- 3、 1 数 学
- 3、 2 天 文 学
- 3、 3 物 理 学

10進コードは、いくらでも細分化できるので変更可能であるが、桁数が不安定であるという欠点がある。

### (5) 表意コード (Mnemonic Code)

これは、記憶に役立つものを、コードの構成部分とする。コードは、数字や文字の組合せになることが多い。

たとえば

製品 サイズ 型

T V 0 9 A …… テレビ、9インチ、A型

T V 1 2 A …… # 12 # #

T V 1 4 B …… # 14 # B型

TV 16 C テレビ、16インチ C型

TV 19 C " 19 " "

表意コードは、対象項目が多くなるという欠点はあるが、識別が容易である。

## 2-2 アウトプット設計

アウトプットの設計とは、出力装置からどのような様式で情報を出力させるかを定めることである。それにはまず、経営者や管理者が、どのような報告書を求めているかを知ることが必要である。それには、必要な情報を必要な時に、見やすい形で印刷しなければならない。

それでは、ここで、代表的な出力装置であるラインプリンタの場合についてのアウトプット設計を考えてみよう。

### (1) ラインプリンタの機能概要

項 目	機 能
印 刷 桁 数	100、120、132、144 桁など
活 字 の 種 類	英字、数字、カナ文字、特殊記号
1 桁 の 幅	1 / 10 インチ
行 間	1 / 6 または 1 / 8 インチ
用 紙 の 最 大 幅	機種によって異なるが最大 $18 \frac{3}{4}$ インチ程度
用 紙 の 最 大 長	" " 22 インチ程度

### (2) ラインプリンタによるアウトプット設計

- ① ラインプリンタの用紙を連続用紙といい、上から下へと印刷する。
- ② ラインプリンタで印刷できる文字の種類は、機械によって限られている。
- ③ ラインプリンタは、1行に印刷できる桁数が限られている。
- ④ 行間は次のようにとることができる。

以上のハード的な制約を考慮に入れながら、スペイニング・チャートとよばれる用紙に出力様式を設計するのであるが、設計上の留意点としては、

- ① 帳票は、見やすいこと、使いやすいことが最も重要である。

○ シングルスペース

○ ダブルスペース

○ トリプルスペース

数 量	金 額
× × × ×	× × × × × ×
× × × ×	× × × × × ×
× × × ×	× × × × × ×
× × × ×	× × × × × ×
× × × ×	× × × × × ×

数 量	金 額
× × × ×	× × × × × ×
× × × ×	× × × × × ×
× × × ×	× × × × × ×
× × × ×	× × × × × ×

数 量	金 額
× × × ×	× × × × × ×
× × × ×	× × × × × ×
× × × ×	× × × × × ×
× × × ×	× × × × × ×

そのためには、項目の配置や区分けに注意しなければならない。

- ② スペイシング・チャートに印刷する場合、コンマ、ピリオド、マイナスの表示方法や、カナ文字データの扱いにも注意しなければならない。
- ③ 印字時間の短縮、帳票の大きさにも注意しなければならない。
- ④ 帳票のフォーマットを印刷してそれに印字する場合は、その帳票の性格をよく考えなければならない。

表 1 納品書の例

	0			1			2			3			4			5			6												
	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	
1																															
2																															
3																															
4																															
5	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
6	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
7	*																														
8	*																														
9	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
10	*																														
11	*																														
12	*																														
13	*																														
14	*																														
15	*																														
16	*																														
17	*																														
18	*																														
19	*																														
20	*																														
21	*																														
22	*																														
23	*																														
24	*																														
25	*																														
26	*																														
27	*																														
28	*																														
29	*																														
30	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
31																															
32																															
33																															

## 2-3 インプット設計

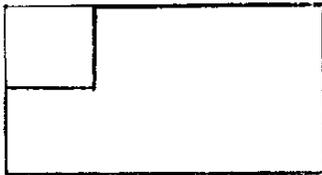
インプット設計とは、コンピュータにデータを入力するための媒体（原票、カードなど）の様式を設計することである。

ここでは、原票にもとづきカードにパンチし、パンチされたカードをコンピュータに入力する場合を考え、原票の設計とカードの設計を考えてみよう。

### (1) 原票の設計

原票を設計する場合に考慮しなければならない点は、原票で発生したデータミスが、論理的にチェックしうるミスを除いて、そのまま出力されてしまうので、原票でのデータミスは許されないということである。記入・転記の際に発生するエラーは、すべてのエラーの90%をしめているといわれ、この記入・転記ミスをなくすために記入項目、転記項目の配列を考慮する必要がある。

そのために、各記入・転記をする項目の部分を、太線で囲んだり、ケタの区切りをあらわすために、点線を使ったりする。



記入担当部門でかならず必要なコードは、あらかじめ原票上に印刷しておくことも有効である。しかし原票が良く設計されていても、どうしても間違いが生じがちである。そこで記入、転記担当者を集め、原始データのミスは、最後までひびくことをよく理解させ、記入方法の講習をするのも1つの手段である。

つぎに、この原票にもとづき、パンチャーの手でデータがカードに穿孔される。そのため、パンチャーにとって読みやすく、パンチしやすい原票でなければならない。左から右へ、上から下へ読んでいけるように、項目が配列されていることが大切である。

(良い例)

1	2	3
4	5	6

(悪い例)

4	1	5
6	3	2

またパンチする項目を太線で囲んだり、点数をつかってケタの区切りを明確にし、コラム数を入れてパンチしやすいようにする。

社員番号	1	2	3	4
------	---	---	---	---

原票設計で重要なことは、原票にデータとして必要な項目が網羅されていることである。その場合、出力項目から逆に入力項目をあらい出すことによって、データとして必要な項目が網羅されている原票を作成することができる。

表2 個人別基本ファイル作成書設計例

個人別基本ファイル作成書設計例		備考	
年月日	14181812310928	氏名	
部コード	01	コード	
カードNo.	01	氏名カナ文字	
カードNo.	02	銀行コード	

- (説明)
- この帳票は、各支店に分散する全職員の給与計算業務の基本ファイルを作成するための最初の投入物の一部である。
  - ウラ郡で囲んだ部分がカードにパンチされる情報であり、1-80の数字はカードのコラム番号を示す。(実際はカード01は70桁まで、02は76桁まで使用)
  - 氏名はカナ文字で投入されるので「」で囲む。



## 2-4 ファイル設計

ファイルとは記録の集まりのことで、ある目的に沿って組織的に集められた情報をいう。コンピュータ・システムにおいては、補助記憶装置の記憶情報をさす場合が多い。

具体的にファイルとはどんなものであるか、製品在庫台張を例にとってみると、月日、入庫数量、出庫数量、在庫数量などを項目という。そしてこの項目が集まったものをレコードといい、レコードが集まったものをブロックという。1ブロックの情報は丁度台張の1頁に相当する。そしてこのブロックがいくつが集まって、1つのファイルになるわけである。

それでは、このファイルにどのようなレコードの編成がなされているか見てみよう。

(ファイルの形態)

### (1) 順次編成ファイル (シーケンシャルファイル)

ファイル中の記録の見出しとなる項目 (キー項目) が、正しい順序に従って収録されているファイルを、シーケンシャルファイルという。この方式のファイルとしては、一般に磁気テープが多く用いられている。

1	データ	2	データ	3	データ	4	データ
---	-----	---	-----	---	-----	---	-----

☆数字はキー・コード

### (2) 索引順次編成ファイル (インデックス・シーケンシャルファイル)

これは、索引 (インデックス) を持った順次編成ファイルで、順次に処理することも、索引を用いて個々のレコードを直接アクセスすることも、同様に迅速に行なうことができる。この方式のファイルとしては、磁気ディスクや磁気ドラムが用いられる。

### (3) 直接編成ファイル (ランダム・ファイル)

ランダム・ファイルは、データ・レコードを記憶する場所が、キー・コードの分類機能などをもとにして決められているファイルである。したがって直接、目的のレコードを読み込んだり、書き出したりすることができる。こ

の方式のファイルとしては、磁気ディスクや磁気ドラムが用いられる。

磁気テープを媒体としたシーケンシャルファイルの設計

磁気テープの場合、レコードは一般につきのような形態で記録される。



磁気テープは、リール状になっているので、巻き取りながら読み書きをブロック単位で行なうため、ブロック間にデータの記録されない部分がつくられる。この部分のことをIRG (Inter Record Gap)という。このギャップの長さは、0.75 インチで一定になっている。

磁気テープの場合、一般にブロックの大きさは数レコードになる。1ブロック当たりのレコード数を多くすれば、1巻当たりのIRGが少なくてすむため、大量のレコードを記録することが可能である。しかし磁気テープの場合、読み書きはブロック単位で行なうのでブロックが大きくなると、記憶装置の入出力領域も大きくしなければならない。入出力領域を大きくすると、プログラムを入れるプログラム領域を小さくしなければならないので、複雑な処理を行なうことができなくなってくる。そのため、ブロックをあまり大きくすることも、よいとはかぎらない。

1ブロックの桁数(大きさ)が一定ならば、1レコードの桁数が小さいほど、1ブロック当たりのレコード数はふえる。したがって、ブロックの大きさによって、磁気テープに記録できるレコード数が左右される。

磁気テープ1巻に記録できるレコード数は、次の計算式により求めることができる。

$$1 \text{ 巻のレコード数} = \left\{ \begin{array}{l} (\text{テープの長さ}) \div \left( \frac{1 \text{ ブロックの桁数}}{\text{記録密度}} + \text{ギャップの長さ} \right) \\ \times (\text{ブロック内のレコード数}) \end{array} \right\}$$

☆記録密度；1インチの長さに記録できるケタ数

以上のように、磁気テープ・ファイルの設計を行なう時には、ブロックの大きさを何桁にするかが重要になってくるのである。

レコード内の項目の配列

処理が容易にできるように、磁気テープ上のレコードにおいては、表示項目（商品番号、社員番号など）を先頭におき、つぎに計算項目（数量、単価など）をグループにまとめて配列する。また表示項目でキー・コードとなりうる項目がいくつかある場合には、グループ化しておく。

表示項目	計算項目
------	------

磁気テープ・ファイルの設計を行なう場合の注意点

- (1) 1ブロックを構成するレコードの数を決める。
- (2) 1レコードを構成する項目の種類、桁数およびその配置を決める。

以上で磁気テープを使用するファイルの設計について述べてきたわけであるが、ここでファイル設計全般について、考慮しなければならない点をあげると、

- (1) 作成するファイルの性格はどうか、
- (2) ファイルの編成方法はどのようにするか、
- (3) ファイルの媒体としては何を使用したらよいか、
- (4) ファイルに収めるべきデータ項目の決定
- (5) ファイルにどのくらいのデータが収まるか、
- (6) 入出力の時間はどのくらいかかるか、

2-5 データ・チェック

コンピュータを使う場合に、忘れられやすいのは、オーディティングいわゆる監査という機能である。たとえば、手計算の場合は、必ず検算を行なう。そして、コンピュータの場合でも、オーデット（検算）の部分を必ず入れておかなければならないわけだが、コンピュータでやるのだから間違いはないということで、この部分をつけないでしまう。これは非常に間違った考え方で、コン

コンピュータで行なう場合にも手計算の場合と同様に、必ずオーデットの機能を処理の流れの中に当然組み込んでおかななくてはならない。

このような配慮がなされないと、Aさんの残業手配がBさんに支払われたり、Cさんの口座にはいるべき預金がDさんの口座に振り込まれるようなとりかえしのつかない結果になってしまうのである。

そこでデータ・チェックの必要性が生まれてくるわけであるが、チェック方法の設計とは、主としてコンピュータが行なうチェックの範囲、内容を設計することである。すなわち、どの項目について、どのようなチェックを行なうか、ミスが発見されたときにどのような事後措置をとるかを定めることである。

ところが、「コンピュータは万能である」と思い込んで、ミス発生のあるゆる場合を想定してチェックを設けるというやり方は、一見確かなようであるが、経済性ということを考えると必ずしも得策ではない。たとえば、100項目についてチェックするようプログラムした業務があるとする。ところが実際に処理してみるとエラー発生は10項目程度であって、他の90項目はほとんどエラーを起こさなかった。このような場合は、チェックに費す時間のうち90%がむだになっていることになるので、90項目は思いきってチェックをやめるべきである。

### チェック方法の種類

#### (1) コンピュータにインプットするまでのチェック

- ① 手書き帳票の場合は、最もエラーが発生しやすいので、あらかじめ書体を統一しておく事が大事である。
- ② コード誤りによるエラーは計算精度を致命的なものとしてしまう恐れがあるので、なるべくゴム印を使用すること。
- ③ データの投入責任を明確にしておいて、定められたチェックポイントを点検のうえ投入することを義務づける。
- ④ 投入データの脱落紛失を防ぐ手段を講ずること。

#### (2) プログラムによるチェック

##### ① トータルチェック

インプットするデータの合計をあらかじめ求めておき、コンピュータで



② 出力された結果に代替するもの（過去の統計結果など）と比較する。

③ 出力された結果を手操作で検討する。たとえば、送状、請求書などがその例である。

## 2-6 例外処理

例外処理は、プログラムに組んだ命令で、特定の処理を拒否するような結果のことであり、場合によっては、処理が何らかの性質を持っていてプログラムの編集手順にパスしない、又はプログラムにその対処方法を書いていないことの結果生ずるものもある。

それ故、例外処理はそれぞれ調査を加え、誤りを直し、そして通常はもう一度処理し直す必要がある。あるいは、性質の違った処理ということで、別に手作業で行なう場合もある。

例外事項を処理するには、大きく分けて2通りある。

1つは、例外が出てくると、それを全部プリントアウトし、そしてそれを手作業で行なう方法と、もう1つには、考えられるすべての例外事項を分析してフローチャートへ組み込んでおく方法である。たとえば、給料計算の例でいくと、従業員1万人の会社の中で、自動車の運転手が10人いたとする。自動車の運転手というのは、断続勤務だから給料の計算方法が違ってくる。実際、こういう人々の給与計算もすべておりに込んで計算するか、つまり、1万人の中で10人の人の計算をはたしてこの中におりに込んでいくかどうかである。たとえば職員ナンバー12345番の人は運転手だから計算はしない、というリストを出した後、手で計算する方法と、そのような例外もすべて含んで流れの中に組み込んである、という二つの方法があるわけである。

実際に、このどちらの方法をとるかというのは、あくまで全体に占める例外の起こる率をどう把握するかによる。たとえば時間があって分析の暇さえあれば、この10人というのは必ず毎月起こる例外だから入れておこうという考え方になる。しかし別の場合、万に1つも起りそうもない例外だから、その例外が起きた場合手作業で処理すればいいということもある。その場合には、むしろ前者の例外事項の抽出ということで処理してしまう。

このような例外処理のシステム設計にあたって問題になってくるのは、とにかくシステム・アナリストが行なう場合は、そう例外処理もシステムの中に組み入れて、システムを大きくしてしまう傾向にある。ところが現場の人は、例外は別に処理すればよいと考える。そのようなときに適正な判断ができるのは、システム設計の専門家ではなくて、業務に精通している人であるということである。

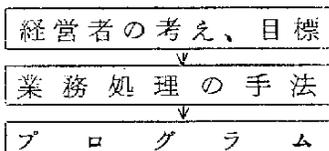
例外処理というのは、あくまでケースバイケースであって、むしろその処理にあたっては、実際に業務の専門家の方が正確に判断できる場合があることに注意しなければならない。

### Ⅲ 実施準備段階

#### 1 プログラミング

詳細なシステム設計が終ると、それにもとづいて、プログラムを作成しなければならない。

プログラムとは、経営者の経営目標、経営者の考え方をあらわすものである。



つまり経営者の考えとか、経営の目標が、管理者に伝えられて、業務処理の手法に置きかれられ、そしてそれが実際にコンピュータを動かすプログラムという形になるわけである。

プログラムというのは、客観化された業務の処理手順である。したがって、プログラムが作成されると、それをコンピュータに入れることによって、誰がやっても同じ結果がでる。たとえば、従来手でやっていた場合には、××会社原価計算要領とか、〇〇会社給与規程などのような規則によって定義づけられ、それが人間の頭の中にはいついて、それに基づいて処理されていた。それがコンピュータで処理する場合には、この手順というものが流れ図に分析されていて、それがプログラムという形でコンピュータの中に入れられ、処理されるわけである。その大きな相異点は、人間の場合、人によって処理の結果の良し悪しが出てくるが、コンピュータで処理をした場合、誰がやってもボタンを押すだけで、同じ結果が出てくることである。そういう意味で、プログラムは、客観化された業務処理の手順といわれるわけである。

したがって、経営の目標が立てられ、定義されたならば、それをプログラムの形において行なうと、それが会社の一つの経営のやり方、あるいは規格、標準になる。そのためには、プログラムは、誰が見ても分かるように資料化され、ドキュメンテーションがなされていなければならない。

プログラミングを行なう手順

- (1) システムの内容を十分理解する。

- (2) プログラム設計（ディテールフローチャートの作成等）を行なう。
- (3) 詳細設計の条件を満足しているかどうかの確認をうける。
- (4) コーディングを行なう。
- (5) デバッグをくり返してプログラムを完成させる。

この場合注意しなければならないことは、システム設計に基づいてプログラムを見るばかりでなく、プログラミングからシステム設計をみる必要がある。というのは、プログラミングを無視してシステム設計を行なうと、プログラミングが困難だったり、場合によっては、不可能という場合がでてくる。そのためは、詳細設計の説明を受けたプログラマは、提示された資料が十分であるか、また設計されたシステムでそのままプログラムが作成できるかを検討し、その結果、必要があれば、資料の補充やシステムの変更について提案を行なわなければならない。

#### デバッグについて

プログラムは人間が行なうのであるから、細心の注意をはらっても、誤りの生ずる場合が多い。したがって、プログラミングが終了すると必ずプログラムのチェックをする必要がある。この作業のことをデバッグという。デバッグには、コンピュータを使用する前に、机の上でプログラムをチェックするデスク・デバッグと、実際にコンピュータを使い、その結果出たエラーリストを使ってチェックするマシン・デバッグとがある。

プログラムを書く言語には、たとえば、フォートラン（FORTRAN）という技術計算用の言語とか、コボル（COBOL）という事務計算用の言語がある。あるいはアルゴル（ALGOL）といった数学者向けの言語や、最近使われてきたPL/Iといった言語もある。ところがこれらを全部覚えるわけにはいかない。また1つの会社で2つも3つも違う言語を使っていたのでは、企業の標準化が困難になってくる。そこで最近では、コーポレート・ランゲージという全社共通のプログラム言語が必要になってきているといわれている。その言語は、専門家だけにわかるのではなく、経営者も管理者にも理解できるようにしなければならない。つまり、経営者、管理者が、ある程度勉強すれば、実行者が書いたプログラムをある程度読むことができ、チェックすることができる程度でなければならない。それによって、少なくとも経営者や管理者が考えていること

がプログラムという段階でも、確実に実行者によって実行されているかということが、マネジメントに対してフィードバックされるのである。

そういう意味で、最近 PL/I ということばを会社のコーポレート・ランゲージ、つまり企業の共通語にしようとしている動きがある。これは世界でもわりあい大きな企業がすでに 100 社ぐらい、又日本でも相当多くの一流会社が PL/I を共通の言語にしている。

## 2 プログラムテスト

プログラムが一応完成すると、テストデータによるプログラムテストを行なう。このプログラムテストは、プログラムに組み込んだあらゆるケースの事故を含む資料を用いて行ない、これらが、プログラミングされた通りに処理されるかどうかをテストするのである。

プログラムテストには 2 通りある。

1 つは、非常に大きなシステムの場合には、それらを小さなサブシステムに分けて、それをブロックにしながらかテストを進めていく方法、つまり、そのブロックのシステムを個別にテストする方法であり、もう 1 つは個別にでき上がったものを、さらに組み合わせながら、ちょうど埋もれ木細工のようにして、全体のシステムをつくっていく総合テストの 2 通りである。

たとえば、JALCOM(ジャルコム)の場合を見てみると、カウンターでチェック・インするときのお客と女子職員との応対の部分でのプログラムのチェック、そしてそれが実際呼び出されて予約されているパッセンジャー・リストとの比較の部分のテスト、さらにその人がその飛行機のみならず、その先で乗り継ぐ飛行機とのかみ合わせ等それぞれのブロック、ブロックでのプログラムを書いていく。そのテストが個別のテストであり、それらを組み合わせて、一人の人が予約してからカウンターに来て、チェック・インし、さらに行き先の飛行機の確認を取りつけるまでのすべての全体のシステムを通して総合テストを行なうのである。

### 3 ドキュメンテーション

プログラムが終ると次には実施の段階にはいるわけだが、その前に、業務担当者に業務内容を、的確に解説する文書を作成しておく必要がある。この文書の作成をドキュメンテーションという。つまりシステムを堅実に導入するためには、またシステムの導入後においても多くの適用業務を円滑に処理するためには、システムに関する詳細な事柄を標準化して文書にまとめておく必要がある。

#### 文書化の目的

- (1) 仕事の範囲、仕事の内容、使用される方法、予想される節減、今までの作業に対する変更箇所などに関して生ずる誤解を防ぐことに役立つ。
- (2) システム導入における進行状況の重要な記録の一部となる。
- (3) 新システムを業務担当者に周知徹底させる基準となる。
- (4) 新人の訓練、オペレータの教育、システムの一部変更などの基準参照資料とする。

#### 文書化を行なう際の留意点

- ① 文書化を行なう前に、事柄を標準化しておかなければならない。使う言葉の意味、まとめ方の形式などを標準化しておかないと誤解のもととなり混乱をまねく恐れがある。
- ② 記入する際には、今まで生じたすべての変更を記録しておかなくてはならない。それと同時に、最近の記録を確保しておかなければならない。
- ③ 文書化する場合、秩序立ったファイリング・システムを作る必要がある。それが乱雑になっていると、せっかくファイルしてあるのに、どこに何が書いてあるか分からないのでは、文書化の意味がなくなってくるのである。

文書化を行なう場合は、それぞれの目的にあった記録形式がとられるが、それには次のようなものがある。

(1) ラン・ブック

これは、ランごとに必要な諸項目を記述したもので、それには、プログラム名、プログラム・フローチャート、使用するアウトプット用紙、使用する入力データ、プログラム・メッセージの意味、再始動の方法、出力データの見本、チェックすべきコントロール合計、プログラム・リスト、コードの種類、などが記入されている。

これによってデータ処理の中味がすべてわかるようになっている。

(2) オペレーション・マニュアル

これは、ランごとの操作に必要な注意事項を記入したもので、オペレータのマニュアルとなる。

それには、入出力装置の番号、リール・ラベルの処理法、プログラム・メッセージの意味と処理方法、入出力の種類とその処理経路、コントロールカードの使用方法、各装置の準備の方法などが記入されている。

以上のように文書化を正確に行なっておくと、業務推行中に、担当者が変わっても業務を支障なく続けることができ、また新システム導入後、変更や改善があっても、それを効率よくまた確実に行なうことができるのである。

#### 4 機械導入のための諸準備

総合テストが終ると新システム導入のための準備を行なわなければならない。すなわち、移行の日程計画をたて、これに従ってファイル作成の準備、業務実施要領の作成、物品の調達、業務指導など移行に必要な準備を行なう。

新システム移行の場合は、試行という段階があって、現行システムと並行して処理を行ない、新システムが円滑に動くかどうかを確かめてから本格的に新システムに移るのである。

それではその準備を順を追って述べてみよう。

##### (1) ファイルの作成

ファイルには、システムを切りかえる時点で、最近の情報が貯えられていなくてはならない。このために、切替前に作成したファイルは、常に現行に維持し、いつでも使用できる状態に準備しておく必要がある。したがって、ファイルの作成時期が早すぎると、これを使用するまでの間の現行維持に多くの手数を要することになるので、ファイルの作成する時期には注意を要する。

##### (2) 業務処理基準書の作成

機械処理システム移行後の手作業部門の仕事のやり方を示すのが業務処理基準書である。業務処理基準書は、詳細設計書に基づいて、業務担当部門が、作成すべきものであり、通常、インプット作成までの事務処理方法と、インプット、アウトプットの送達管理方法がその主体となる。

インプット作成までの事務処理方法としては、従来の手作業処理を機械処理システムにマッチするように改善したものが、詳細設計に明示されているので、そのやり方を担当者が理解しやすいように具体的に記載することとし、仕事の手順を示すフローチャートや、インプット、アウトプットの様式を明示することが望ましい。

また、アウトプットが、インプットの作成部門にフィードバックされず、特定の管理部門に送付されるようなシステムにおいても、全体的なシステムがわかるような説明を行ない、現場で投入される資料がどのような形でアウトプットされ、管理資料としてどのように使用されるかを明らかにすることが望ましい。これは、投入段階の人たちに、自分の仕事の重要性を認識させ、

責任ある仕事をさせるうえに役立つのである。

### (3) 物品の調達

試行のために必要な物品は、大部分が切替後においても必要とするものであり、試行結果に伴う大幅なシステム変更がないかぎり、そのまま使用できるものである。したがって、通常は本実施のために必要な一定期間の量を見込んで準備しておく必要がある。

所要物品のうち、コンピュータ部門で使用する主なものは、カード、紙テープ、アウトプット用紙、である。各業務に共通に使用できる応用カードや応用用紙を使用する場合は、コンピュータ部門で一括購入するので、概算して購入すればよいが、その業務特有の様式のものを作成する場合は、それぞれ個々に購入することになるので、見込所要量の算出にあたって正確性を必要とするとともに、経済的な購入数量について特に留意する必要がある。また、現場で使用するおもな物品は、インプット用紙であるが、その業務に特有なものは、適正な購入数量、発注の時期などに留意する必要がある。

### (4) 業務指導

移行にあたって最も重要なことは、その仕事に関係あるすべての人々の積極的な協力を得ることであり、とくに現場の人たちの理解と協力がたいせつである。したがって、試行前に、機械化の必要性や効果について積極的に周知を行なうことが必要である。システムの周知にあたっては、現在実施している手作業が、機械化後どのように改善され、現場の負担がいかに軽減されるかに重点をおき、そのシステムの実施により業務全体がいかに合理化されるかを強調すべきである。

また、現場の作業は機械化に伴って大幅に変更されることになるので、現場に対する業務指導を十分に行なう必要がある。現場の人たちがとくに知りたいことは、新システムの移行に伴って、自分たちの仕事がどのように変わるかということであるから、仕事の内容について、業務処理基準書に基づいて詳細に説明し、新しい業務内容に十分習熟させることがたいせつである。また、現場の担当者は、新システム導入後、自分たちの果たすべき役割について認識する必要がある。そのためには、業務指導において、単に担当者として直接関係のある業務内容のみでなく、システム全体の構成とそのシステ

ムの目的について十分研究し、これらを認識したうえで、自分たちの仕事を進めていくことが望ましい。

## 5 コンピュータ要員

ここでもう一つ重要なことはコンピュータの要員の問題である。

ここでは要員としてどんな職性をもった人たちが必要か、またその選出の問題は何かを述べてみよう。

コンピュータの要員には、システム・エンジニア、プログラマ、オペレータキーパンチャーなどがあるが、これらの選出にあたっては、組織と人事移動に関するもので、公然とは打合せや討議ができない性質の問題である。

### (1) システム・エンジニア

システム・エンジニアとは、理想的なデータ処理システムを設計するにあたり、これを現実に移行する人のことをいい、その職務としては、

- 実務担当者と協力し、現状の情報処理システムの問題点を摘出し分析する。
- 新しい情報処理システムの設計を行なう。
- プログラマに、プログラムの内容を指示するプログラム作成仕様書を作成する。

またシステム・エンジニアに具備すべき条件としては、

- 優れた分析能力を持ち、論理的であること。
- 精神的にも安定し、優れた創造力、独創力、応用力を有すること。
- トップマネジメントや関連部門に対して、説得力や表現力があること。
- コンピュータに関して深い知識を持っていること。
- コンピュータ知識以外に広い知識を持っていること。

### (2) プログラマ

システム・エンジニアによって作成されたプログラム仕様書にもとづいてプログラミングをする人のことをいう。

その職務としては、

- プログラミング
- デバッグング
- プログラムテスト
- オペレーション・マニュアルの作成

また具備すべき条件としては、

○分析能力を持ち、論理的で細かな仕事を正確に行なり能力を持っていること。

○忍耐強く、集中力があること。

○注意力がありしかもその転換能力を有すること。

○理解力と問題解決の能力を有すること。

(3) キーパンチャー

プログラマによって作成されたプログラムや、伝票などに書かれた情報をコンピュータに入力するためにカードや、紙テープにパンチ（せん孔）する人のことをいい、女性がほとんどである。

具備すべき条件としては、

○スピードと正確性

○単純作業の繰り返しに耐えられること。

○仕事の拘束性に耐えられること。

○特殊な人間関係にあるので協調性に富むこと。

(4) オペレータ

オペレータは、パンチャーが作成したカードや紙テープをコンピュータに入力し、プログラマが作成したプログラムを使用して、必要な情報処理を行なり。

具備すべき条件としては、

○誠実で反復的な仕事を正確に行なり能力を有すること

○冷静な判断力と責任感の強いこと。

○使用する装置をよく理解し、オペレーション上の特性と限界を知ること。

## Ⅳ 実施および評価段階

### 1 実施結果の検討と反省

実施準備が完了して、いよいよ試行の段階にはいるわけであるが、試行中は、試行計画に基づいて、コンピュータ室とデータの投入点の周辺について、試行の実施状況を点検しなければならない。試行の点検は、随時、現場におもむいて試行状況を調査するとともに、一定周期ごとに、つぎのような点を調査して検討するとともに、手作業とコンピュータ・システムによるアウトプットの照合によって、システムを点検しなければならない。

#### (1) インプットの到着状況

これはインプットデータが、締切日まで確実に到着しているかどうかを調べると同時に、どのデータが遅れているかをチェックして対策を考える。

#### (2) インプットの事故状況

これは、インプットデータのエラーの状況を調べることで、帳票段階では、記入もれ、記入ミス、記入不鮮明などの状況を調べ、パンチ段階では、パンチミスの状況などを調査する。

#### (3) アウトプットの発送状況

システム設計で定められた期日までに、報告書が作成されて、発送されているかどうかをチェックする。

#### (4) 機械処理時間の予定と実績

これは、データ処理中、各入出力装置や、中央処理装置が、システム設計で定められた時間内で処理されているかどうかを調査することで、この状況によっては、システム改善を考えなければならないことにもなりかねないので、タイムを計り正確にチェックする必要がある。

これらの他に調べねばならない重要なことは、システムの目的は、要求されたアウトプットを正しく作成することであるから、試行段階において、アウトプットがシステム設計で指示されたとおり作成されているかどうかを点検することである。

アウトプットの点検は、同一の原始情報に基づいて作成した手作業によるアウトプットと、コンピュータ処理によるアウトプットとを照合し、両者の

相違点を調べる。この場合、両者の間に著しい相違がある時は、コンピュータ・システムまたは、プログラムの不備が原因であることが多いが、アウトプットの数値に若干の誤差があるのは、手作業における誤りか、プログラムの計算式のたて方による誤差の場合が多い。

試行状況の検討は、プランナー、プログラマ、および業務部門の担当者が中心となつて行ない、その検討の結果次のような措置をとる。

(1) プログラムが適当でない場合

アウトプットの分析の結果、プログラムの不備によると思われるエラーが発見された場合、その部分についてのプログラムを再検討し、必要な修正を行なう。この修正が完了すれば、そのプログラムによって情報の再処理を行ない、手作業による作成資料との比較検討を行なう。

(2) システムの変更を要する場合

設計したシステムに不備があり、これを修正しなければ適切なコンピュータ処理システムが期待できないことが明らかになった場合は、当該部分についてのシステムの再検討を行ない、必要な変更を行なう。設計内容の変更に伴って、通常プログラムの修正を必要とするが、場合によっては手作業部分の変更のみで措置できることもある。

以上のように試行結果によってシステムの変更を行なうことは、変更のために、多くの費用と時間を要することが多いので、システム設計およびプログラミングの段階で十分検討しておくように注意しなければならない。

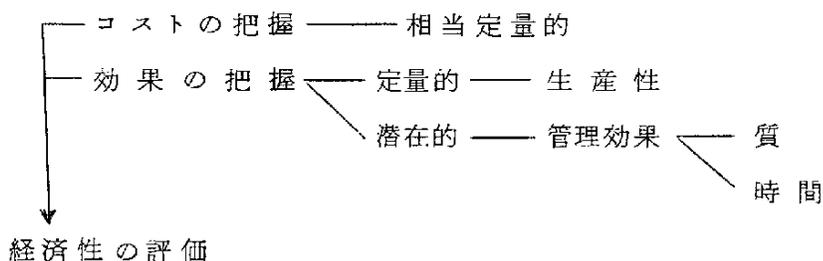
## 2 効果とコストの比較

最後の段階として、実際に実施にはいって、そのシステムをどのように評価するかが重要な問題になってくる。その評価の方法は、非常に難しく、現在でも標準的なシステム評価の方法には定説がない。いま現在考えられる点として、システムの効果とコストの比較がある。これには、実際の効果と事前評価との比較や、機械化コストの実際と事前評価との比較ということに関連して行くわけであるが、事前評価とか、事前予測については、システムをつくり上げ、これを推進するために非常に大きな問題であって、トップマネジメントも、一体このシステムにどのぐらいの費用がかかるのか、またどういふ効果があるか

ということは、大きな関心事ではあるが、しかし、実施に移って、事後の効果把握をおろそかにする場合が多い。本当は、実施後の効果がねらいであるはずなのに決裁のためだけに事前評価とか、事前予測だけでことが終るのでは意味がないのである。コストの把握というのは、相当定量的に行なわれるのであるが、新システムの導入後、一体どのくらい人間が減ったか、あるいは、現在機械で行なっている仕事をかりに人間に引き戻して考えた場合、どのくらいのコストがかかるか、というようにかなり明確にとらえることができる。

それはつまり、コストとそれから得られる効果から割り出すと、生産性ということになる。それらの効果は、はっきり数字で表わすことができるのでさほど問題ではないが、ここで重要なことは、潜在的な効果をどのようにみるかということである。つまりお金の換算することが困難な効果を一体どのように考えるかということである。

それは、管理効果として、いわゆる会社の仕事の質、あるいは、でき上がる時間内にどれだけの問題を、またどれだけの利益を与えているかという点から見るべきである。



たとえば、船をつくる場合、船の仕様書が与えられて、次に仕様書にもとづいて諸元ができ、それによって設計が進むわけである。だいたい初期のころのコンピュータはこの程度まで処理していたが、さらに進んでくると、この設計に基づいて、この船がどのような性能を持つかということも計算できて、さらに、出された仕様書と実際に合っているかどうか、あるいはその仕様書に与えられた許容誤差の範囲にはいるかどうかまでコンピュータで行なうのである。これがコンピュータを使った第2の時期である。

ところが、問題は船をつくるというのは、紙の上で諸元を出して設計して、

性能を計算するだけではなくて、それがはたしてできるかという加工性の問題可能性の問題も重要になってくるのである。つまり、こういう船をつくるためにはどのくらいのドックが必要か、船大工は何人いるか、あるいは資金がどのくらい必要かというようなもっと管理的なあるいは経営的な要素がはいつてくるのである。そしてさらに、船大工が足りなければ、設計を変えたり、仕様を変えたりしなければならぬかもしれない。あるいは他から船大工を借りて解決しなければならない。そうして初めて加工性というものが解決されて、いわゆる工程表だとか、材料、資料の割り当て表とか、資金繰りとか、人員の配置表などのこまかい、いろいろな管理の諸表が出てくるわけである。現在ではこのような段階までも可能になってきたのである。

従来、この範囲は完全な技術計算の範囲であった。しかしながらこれから以後になると、その工場なり造船所の持つ管理能力というものを加味した管理計算になってきたのである。つまり従来は技術計算であったものが、さらに仕事として拡張されてくると、技術管理計算という形になる。それがどういう効果をもたらすかということ、お客がいて、いわゆる仕様が出た場合、はたしてこれは会社として可能であるか、またそれを引き受けた場合どのくらいの利益があるかというマーケティングの基本的な問題に戻るわけである。

こういうような一連の技術管理システムというものが完成しているならば、お客からある仕様が出た場合に、これについては受注ができる、ある条件のもとでは受注できる、あるいは、これを受注しても決して利益はないけれど、受注したほうがいい、とか受注すべきではない、というような適切な経営判断のもとができるわけである。

現在では、これがすべて網羅されたものができてはいないが、一部すでに技術管理計算といったところまでなされている。

ここで以上述べたのは、実は、潜在的な効果というのは、そのマーケティングの原点に戻って、こういうような注文が出たときに、いち早く、それが受注可能であるかどうかということを見きわめる時間の問題が、どれだけ企業に対して効果をもたらすかという一例であり、これはきわめて定性的な問題なのである。しかし、これが会社にとってプラスであることは間違いないのであり、このような定性的な効果は、トップマネジメントが判断していく問題なのであ

る。

### 3 システムの改善

新システムの切替が完了し、全体的にスムーズに運営されていることが確認されると一応の仕事は終了したことになる。しかし、情報処理システムというのは、常に生きものである。常に新しい応用の開発とか、新しい手法の適用によってシステムが拡張されるのである。前項の例に見られるように、最初は技術計算であったものが、次第に拡張していった、技術管理計算、さらにはマーケティング、つまり営業のシステムと結びついて大きなシステムになっていったのである。

このようなシステム改善とか変更が、いわゆるシステムメンテナンスとかシステムリファインといわれる仕事である。システムメンテナンスが小さい補修をさすのに対して、システムリファインはシステムの大きな改造をいうのである。たとえば、給与計算業務における扶養家族手当や超過勤務手当の醸金変更などは、メンテナンスにあたり、投入資料の項目を変更して、統計資料を同時に作成するシステムに変更するような場合はリファインに該当する。一般に、メンテナンスの場合はプログラムの大幅な変更がなく、ファイルの内容補正などで対応できることが多いが、リファインはプログラムの大幅変更を必要とする。

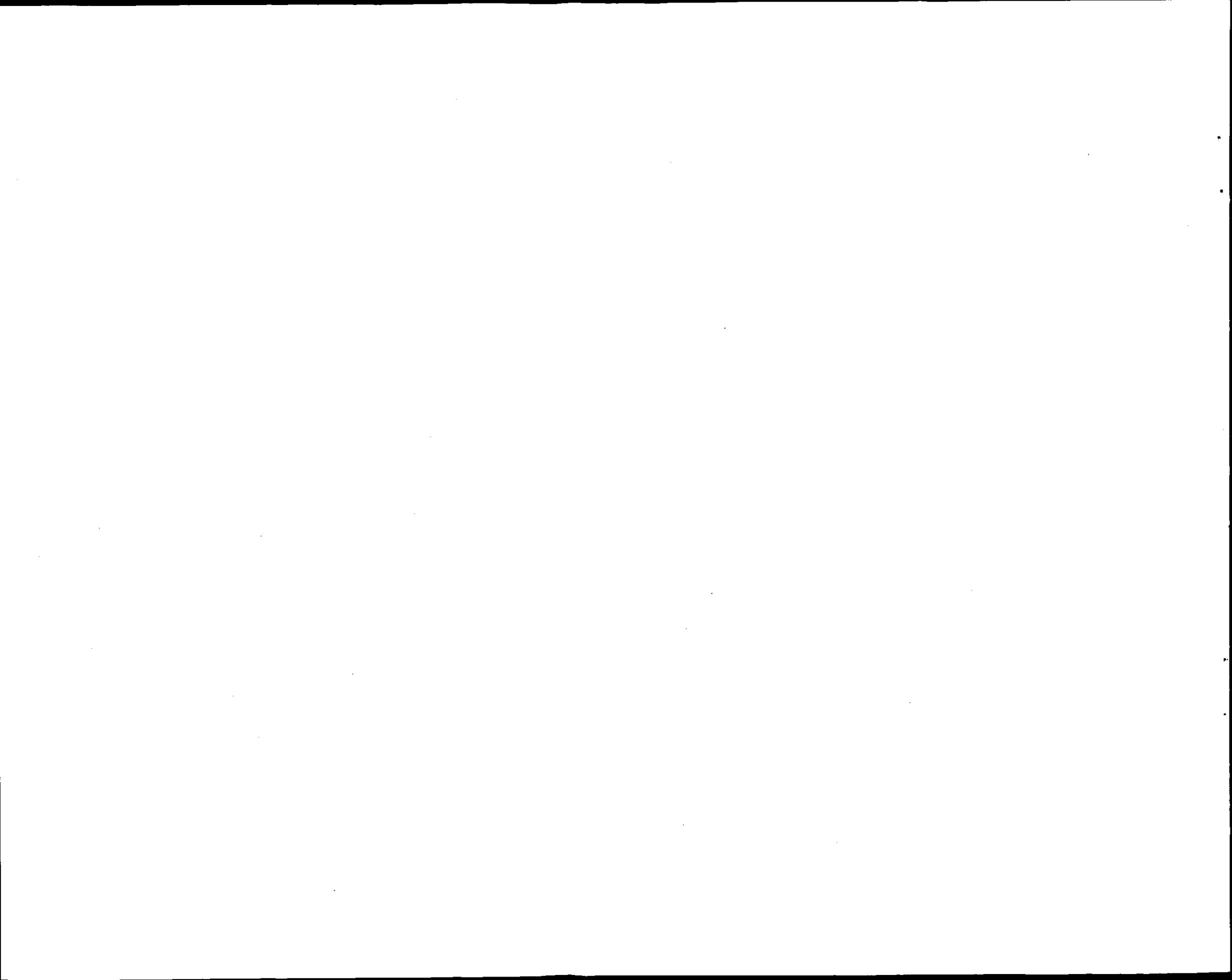
システムはなるべく長い将来を見通して設計し、設計したシステムの寿命が長いことが望ましい。しかし、生きている企業の管理資料を主眼とするシステムを、常に固定させておくことはきわめて困難であり、システムの変更要請は必ず起こってくる。この場合、当初のシステムが、将来の変更を見込んで設計されていれば、比較的スムーズに変更が可能であるが、そうでない場合は、システムの全面的なやり直しを必要とすることもある。

システムリファインの手順は、新しくシステム設計を行なう場合とほぼ同様であり、必要な現状分析と将来の管理方法に基づいて設計する。特にリファインは、現に実施しているシステムの大幅な変更であり、短期間のうちに何回もリファインを行なうことは許されないので、その必要性を十分検討するとともに、

変更がやむをえない場合は、将来を十分考慮したリファインを行なうよう留意しなければならぬ。

最後にこのようなシステム改善を可能にするのは、正確なドキュメンテーションがあつて初めて可能であるということを忘れてはならない。

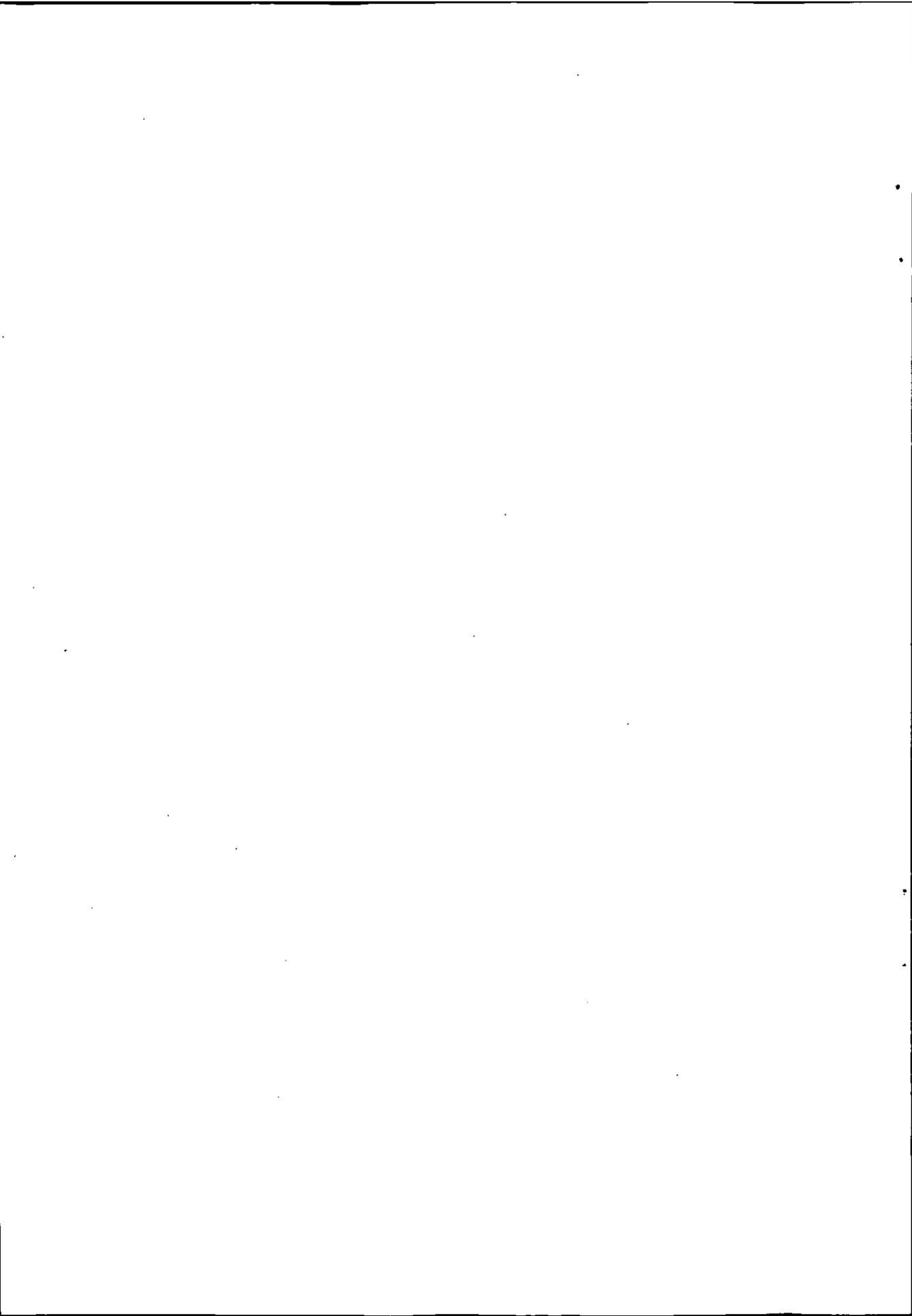




第 4 部

自 習

データ・プロセッシング



# 目 次

1	プログラミング	2 1 6
1-1	プログラム	2 1 6
1-2	流れ図 (フロー チャート)	2 1 6
1-3	コーディングとデバッグ	2 2 1
2	ハードウェア	2 2 1
2-1	中央処理装置	2 2 1
2-2	入出力装置	2 2 5
2-2-1	カード読取装置、カード穿孔装置	2 2 5
2-2-2	紙テープ読取装置、紙テープ穿孔装置	2 2 6
2-2-3	光学文字読取装置、磁気文字読取装置 光学マーク読取装置	2 2 7
2-2-4	ラインプリンタ	2 3 0
2-2-5	映像表示装置	2 3 0
2-3	補助記憶装置	2 3 1
2-3-1	磁気テープ装置	2 3 1
2-3-2	磁気ディスク装置	2 3 3
2-3-3	磁気ドラム装置	2 3 5
2-4	データ通信	2 3 6

## 1 プログラミング

### 1-1 プログラム

卓上計算機で計算をする場合、まず数字のキーを押す。つぎに、+、-、×、÷などのキーを押す、つぎに数字のキーを押して計算をする。この時に我々は、“加算しろ”、“減算しろ”といった指示を、卓上計算機に与える。つまり、卓上計算機に、“つぎはどのような事をしなさい”という指示(制御)をしていることになる。

同じような計算が数多くある場合には、何度も同じ操作を行なわなければならない。そこで、この操作の手順を、あらかじめ覚えこませておき、データをまとめて与えれば、計算できることになる。また、計算が複雑で長いものは、手操作に誤りが生じやすい。この場合にも、操作手順を覚えこませておけば、誤りがなくなる。しかし、卓上計算機には、この操作手順を覚えこませる機能がない。コンピュータには、この機能があり、さらに覚えた命令を間違いなく実行する機能もある。

一連の計算手順を、コンピュータにわかる言葉で、覚えこませて処理をすれば、誤りなく計算することができるわけである。

この一連の計算手順のことを、プログラムという。そして、プログラムを作成する人のことを、プログラマとよび、この作業のことを、プログラミングと言う。

### 1-2 流れ図(フローチャート)

複雑な一連の処理手順を、何も見ないで、コンピュータにわかることばで書くことは、困難なことであり、誤りが生じやすいものである。そこで、まず一連の処理手順を、記号で表わした図にする。この図の事を、流れ図(フローチャート)という。

流れ図を見ながら、コンピュータにわかることばに書き直せば、誤りは減少し、プログラムをはやく作成することができる。したがって、流れ図が正しく書ければ、ほぼプログラムができたとも言える。日常生活も流れ図に表わす事ができる。

〔洋服店で背広を作る例〕

たとえば背広を作る手順を、つぎのように分解することができる。

○洋服店へ行く

- 生地を選ぶ。
- 生地が気に入ったか。気に入れば色を選ぶ。気に入らなければ、さらに生地を選ぶ。
- 色を選ぶ。
- 色が気に入ったか。気に入れば店員をよぶ。気に入らなければ、さらに色を選ぶ。
- 店員をよぶ。
- サイズをはかる。
- お金を支払う。
- 洋服店を出る。

この分解例を、流れ図に表すと、つぎのようになる。

この例では、生地が気に入るまで生地を選ぶため、矢印を①のようにもどすことにより、生地を選択する作業が、くり返されていることを表わしている。色の場合も、同様である。

〔 1 から 10 までの数字を加算する例 〕

この加算では、まず最初の数 (1) に、つぎの数 (2) を加え ( $1 + 2 = 3$ )、その答えに、つぎの数 (3) を加える。このようなくり返しを 10 まで行なうことにより、計算することができる。

これを分解してみると、

「ある数までの合計」+「つぎの数」=「つぎの数までの合計」

であることがわかる。

この「ある数までの合計」を「T」とし、「つぎの数」を「N」とすると、つぎのような式になる。

$$T + N = T$$

したがって、Tには〇絶えず最新の合計が入っていることになる。

つぎに、加える数は、

$$1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10$$

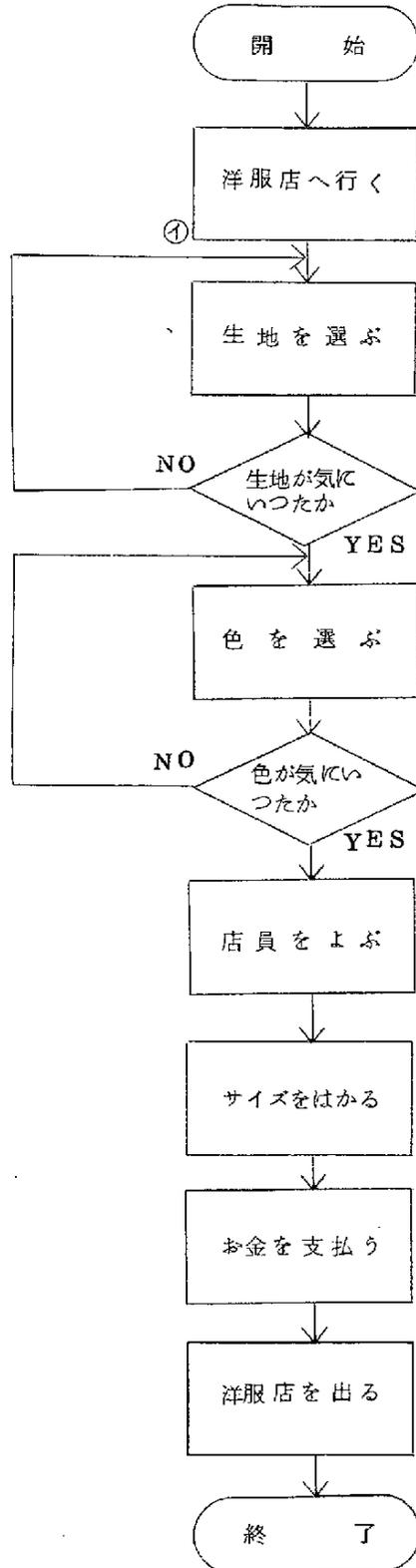
と、1つずつ増えている。

最初の数 (1) をNで表わすと、つぎの数 (2) は、

$$N + 1 = N$$

で表わすことができる。3以降の数字も、同様にして表わせる。このNの数

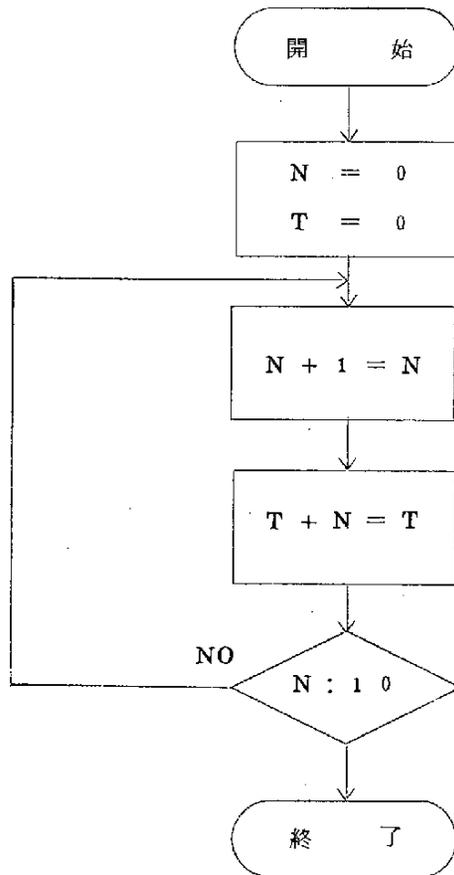
図1 背広を作る例



を見ることにより、今どこまで計算が進んでいるかがわかるのである。この計算例では、Nは10までである。

以上のように、1から10までの加算の式を分解できれば、これを流れ図に表わせばよいのである。

図2 数字の加算例



以上のような計算などが、コンピュータ内部では、どのように行なわれているかを見てみよう。

その手順としては、

- ① 主記憶装置から、命令を持ってくる。
- ② 命令を命令レジスタに入れる。

(命令レジスタは、制御装置の中にあるもので、記憶装置から読み出

された命令を、一時記憶する場所のことである。)

③ つぎに実行する命令の番地を記憶する。

④ 命令コードを解釈する。

⑤ 実行する。

の順に行なわれる。

A番地のデータを、B番地のデータに加算せよという命令は、次のようになる。

加算せよ	A 番 地	B 番 地
------	-------	-------

操作部                      番 地 部

このように、番地部に2つの番地を用いる方式を、2アドレス方式という。これに対して、1アドレス方式というのがあり、次のようになっている。

加算せよ	A 番 地
------	-------

操作部                      番地部

(例)

- ① 

移 せ	A 番 地
-----	-------

      まずA番地のデータを累算器に移せ。
- ② 

加えよ	B 番 地
-----	-------

      つぎに、B番地のデータを累算器に加算せよ。
- ③ 

移 せ	B 番 地
-----	-------

      累算器の加算結果をB番地へ移せ。

(累算器とは、演算をする場所のことである)

命令の順序を制御するために、制御装置内にプログラム・カウンタ (P-カウンタ) というものがある。プログラム・カウンタは、自動的につぎの命令が記憶されている番地を記憶する。今、2000番地の命令が実行されていて、つぎに実行されるものが、2001番地の命令である時、この2001

番地というものが、プログラム・カウンタに記憶されているということである。

### 1-3 コーディングとデバッグ

流れ図ができれば、流れ図を見ながらコンピュータにわかることばに書きなおす。この作業のことを、コーディングという。

コンピュータがわかることばとして、最も一般的に知られているのが、コボルとフォートランである。

コボル (COBOL) は、Common Business Oriented Language の略で、事務計算に適したことばである。

フォートラン (FORTRAN) は、Formula Translator の略で、技術計算に適したことばである。

流れ図にもとづいて、正しくコーディングされたプログラムが、正しい処理をするプログラムであるとは限らない。言うまでもないことだが、コンピュータは、誤った指示をすれば、誤った結果を出す。

そこで、プログラムが正しいかどうかを、十分調べる必要がある。このプログラムを調べることを、デバッグという。

デバッグには、

#### ①机上デバッグ

#### ②コンピュータによるデバッグ

がある。

机上デバッグは、コンピュータを使用しないで、人間がプログラムを調べ直すものである。コンピュータによるデバッグでは、テスト・データを用意し、テスト・データを処理することにより、正しく結果が出るかを調べるものである。さらに本番用のデータも使用して、プログラムが正しく作成されているかを調べるのである。

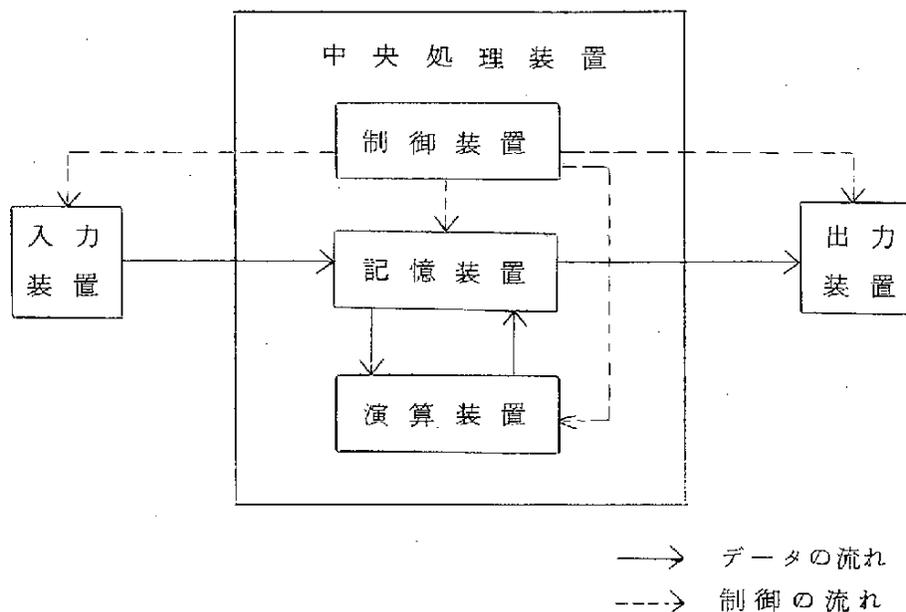
## 2 ハードウェア

### 2-1 中央処理装置

コンピュータの最も大切な中心部分は、中央処理装置 (Central

Processing Unit : CPU) とよばれるものである。中央処理装置は、記憶装置、演算装置、制御装置から構成されており、記憶、演算、制御の3つの機能をもっている。この関係を図示すると、つぎのようになる。

図3 コンピュータ構成図



データは、入力装置（カード読取装置、紙テープ読取装置など）から記憶装置に入る。演算をする必要のあるデータは、演算装置に送られ、演算が行なわれる。そして、再び記憶装置にもどり、出力装置（ラインプリンタなど）に出力される。

制御装置は、入力装置、記憶装置、演算装置、出力装置が指示どおり動くようにコントロールする。

補助的な記憶のために使われる磁気テープ装置や、磁気ディスク装置に対して、中央処理装置内の記憶装置のことを、主記憶装置とよんでいる。主記憶装置は、読み込まれたデータや、演算の中間結果や、書き出されるデータや、プログラムなど、データ処理に必要な情報を記憶する装置である。

主記憶装置の多くは、2進化された信号を記憶しておくために、磁気コアを使用している。磁気コアは、強磁性体（磁石になりやすい性質のもの）を図4のようにドーナツ状にしたものである。

磁気コアの中に電線を通す。その電線に一定以上の電流を流すと、電流の

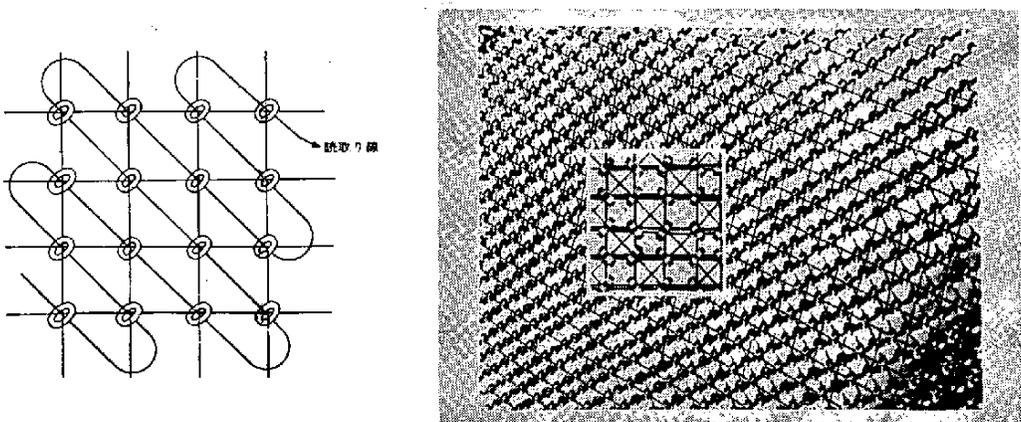
図 4 磁化による記憶の方法



流れる方向によって、磁気コアは一定の方向に磁化される。また磁気コアに一定以上の逆の電流を流すと、磁化の方向は逆になる。磁気コアの状態を読みとるために、読取り線を磁気コアの中に入れておく。読取り線は、磁気コアの磁化の方向が変わった時に、電流が流れる。この読取り線に、電流が流れるか、流れないかによって、磁気コアが、0の状態であるか、1の状態であることを知ることができる。つまり、2つの磁化の方向の状態を、2進数の0と1に対応させているのである。したがって、磁気コア1個で、1ビットの情報を表現することができる。

磁気コアを何個もならべて編んだものを、コア・プレーンといい図5のようなものである。このコア・プレーンが、何枚か重ねあわされたところに、データやプログラムが記憶されるのである。

図 5 コア・プレーン



我々が住んでいる町には、番地というものが付けられ、居住区域を区別するようになっている。一般に、個々の居住区域には、1軒の家があり1家族が住んでいる。貴方の住んでいる場所にも、一定の番地が付いているはずである。したがって、貴方の番地をさがすことにより、貴方を見つけることができる。下図の例では、34番地にAさんが住んでいることになる。

このように、主記憶装置にも各記憶場所に一定の番地が付いていて、その番地に記憶すべき内容（データやプログラム）を書き込んだり、その番地に記憶されている内容を読み取ったりする。

上の例では、20個の記憶場所があり、それぞれに番地が付けられている。34番地という記憶場所に入っている内容は、Aということである。コンピュータでは、主記憶装置へデータなどを書いたり、読んだりする場合、その記憶場所の番地を指定することにより行なわれる。

演算装置は、加算回路、累算器、シフトレジスタなどから構成されており、

11番地	21番地	31番地	41番地	51番地
12番地	22番地	32番地	42番地	52番地
13番地	23番地	33番地	43番地	53番地
14番地	24番地	34番地 A	44番地	54番地

データに対し四則演算、論理演算などの処理を行なう機能を持っている。加算回路は、主として四則演算をする時に用いられる。累算器は、レジスタの一種で、演算に先だてて数値をあらかじめ貯えておく機能も持っている。論理演算（または論理判断）とは、演算装置の持つ判断機能で、何らかの条件に応じて、あらかじめ決めてあるいくつかの処理手順のうち、一つを選択する操作のことをいう。

制御装置は、記憶装置、演算装置、入力装置、出力装置などの装置を制御するものである。制御装置は、コンピュータの各装置に、プログラムの実行

に必要な指令パルスを与える。この指令パルスは、制御装置がプログラムを構成している命令を、逐次解読することによってつくられる。

## 2-2 入出力装置

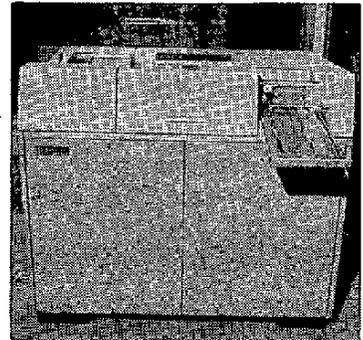
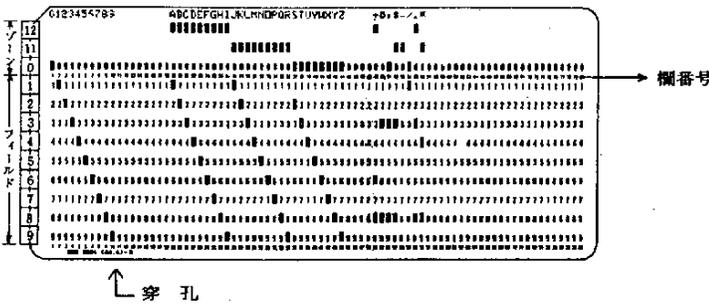
### 2-2-1 カード読取装置、カード穿孔装置

カード読取装置は、下図のような穿孔されたカードを読み、カードに穿孔されたデータや、プログラムの内容を、中央処理装置に送り込むための装置である。

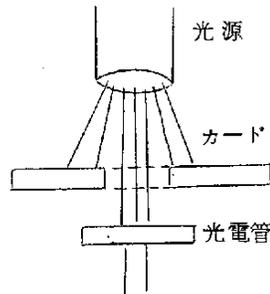
カードに穴があいているかいないか、どのような穴の組合せで穿孔されているかということ、何という文字が表わされているかを読取る。図のカードの1桁目は、0という数字を表わしているが、その情報は、0とカードに印刷された位置が、穿孔されていることで読取る。同様に、数字の1は、1の部分、2は2の部分で穿孔されている。19桁目のAという文字を表わすために、一番上の部分（ゾーン）と、1の部分との組合せで表現し、Bは、一番上の部分と、2の組合せで表現するといった具合である。

カードに穿孔されている状態を、2進数の1に対応させ、穿孔されてな

穿孔段



い状態を、0に対応させることにより、中央処理装置にその情報を伝達するのである。カードが穿孔されているかどうかを調べる方法として、光を利用する方法がある。



カード読取装置の速度は、機種によって異なるが、普通1分間に400枚から1500枚位を読むことができる。

中央処理装置に情報を送るカード読取装置に対して、逆に中央処理装置からの情報を、カードに穿孔する装置がある。これをカード穿孔装置といい、1分間に100枚位から500枚位のカードを穿孔する。

カードの特徴としては、つぎのようなことがいえる。

- ①カード上の原始データの修正、追加、変更、検孔が容易である。
- ②情報が人間の目で確認しやすい。
- ③原始データの編集が容易である。
- ④紙テープにくらべ高価である。

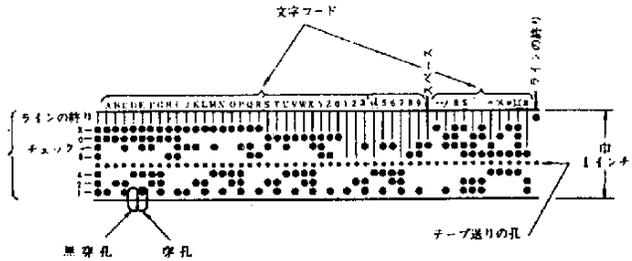
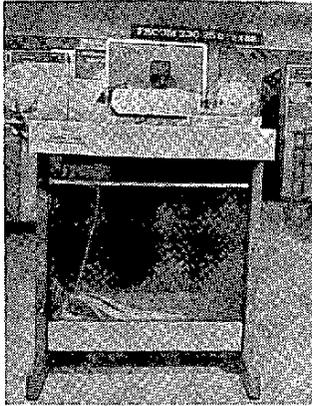
## 2-2-2 紙テープ読取装置、紙テープ穿孔装置

紙テープ読取装置は、図6のような細長い紙テープ上にあけられた穴の状態を読み取ることにより、情報を中央処理装置に伝達する装置である。

紙テープは、6単位のもの、8単位のもがよく使用される。6単位とは、6つの穴の組合せで文字や記号を表わすもので、上図の紙テープは、8単位の例である。紙テープ読取装置は、普通6単位のものでも、8単位のものでも取扱えるようになっている。穿孔されているかどうかを調べる方法は、カード読取装置と同様に、光を使う。

紙テープ読取装置の読取り速度は、1秒間に300字から1000字位

図6 紙テープと読取り装置



の速さである。

紙テープ穿孔装置は、カード穿孔装置と同様に、中央処理装置からの情報を出力するための装置である。穿孔速度は、1秒間に100字から300字の速さである。

紙テープの特徴としては、つぎのようなことがいえる。

- ①テープ上に任意の長さの情報を連続して穿孔できる。
- ②紙テープの価格が安い。
- ③原始データの変更、誤りの訂正が面倒である。

### 2-2-3 光学文字読取装置、磁気文字読取装置、光学マーク読取装置

光学文字読取装置 (Optical Character Reader : OCR) は、穿孔カードや紙テープのように、コード化したものを読ませるのではなく、直接文字や数字を入力する装置である。カードや紙テープのように、パンチャによる穿孔作業が省略できるため、データ作成のための時間が大幅に節約できる。読取り方法としては、強力な光源とレンズを使って、文字に反射する白と黒の模様を解説する方法と、書類の特定位置につけられた、

マークを読む方法とがある。読取り速度は、書類の大きさにもよるが、1秒間に330文字から2,400文字位である。

磁気文字読取装置 (Magnetic Ink Character Reader : MICR) は、OCRと同様に、文字や数字を直接読むことができる。これは磁気インクで印刷されている特殊な文字を読取る装置で、文字のパターンと特徴を解読して、入力するものである。読取り速度は、1分間に950枚から1,600枚位である。

OCRやMICRは、銀行の小切手やガス、水道、電気などの料金請求書などで使用されている。

光学マーク読取装置 (Optical Mark Reader : OMR) は、特定のカードや帳票にマークをするだけで、文字や数字などのデータを、直接、電子計算機に入力する装置である。読取り速度は、1分間に200枚位である。

図7 光学文字読取装置用の文字と読取装置

数字 0123456789  
英大文字 ABC-----XYZ

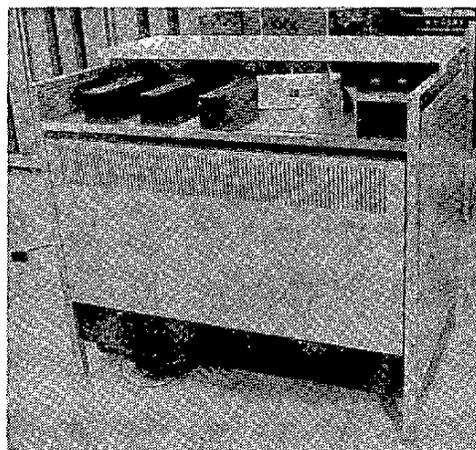


図 8 磁気文字読取用の帳票と読取装置

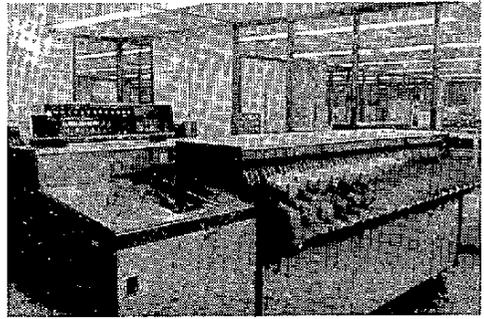
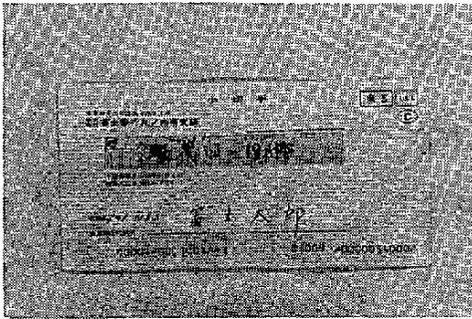
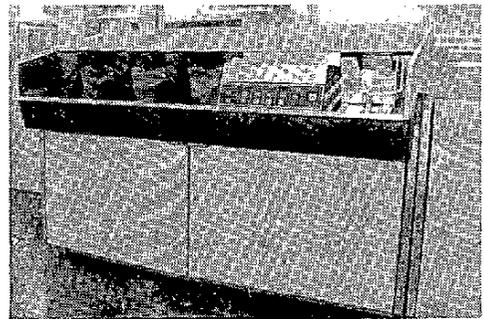
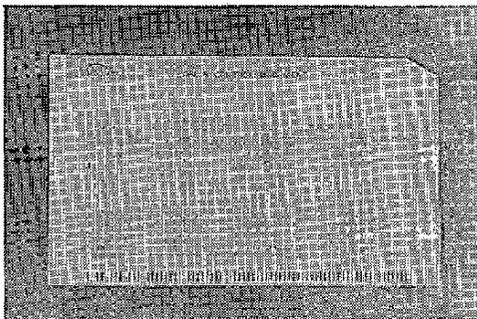


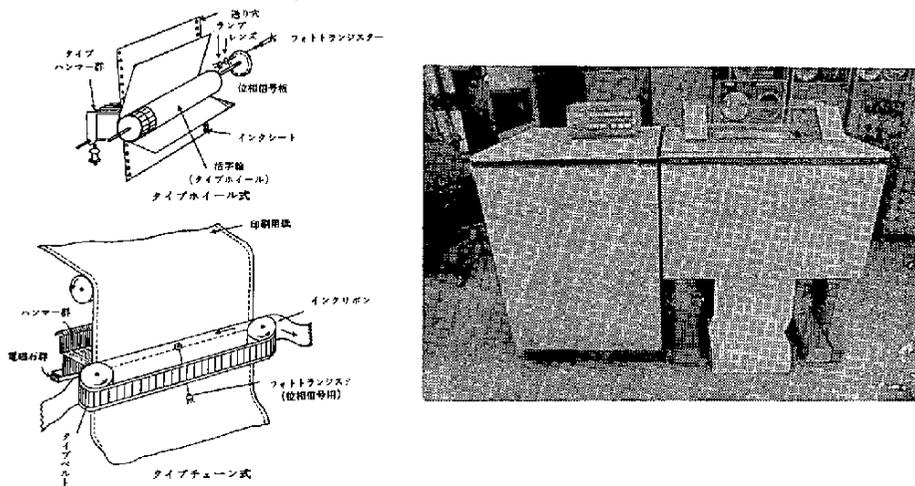
図 9 光学マーク読取用の帳票と読取装置



## 2-2-4 ラインプリンタ

ラインプリンタ（高速製表印字装置）は、中央処理装置からの情報を、1行単位で印刷する出力装置である。1行に120から144の文字や記号を、印刷することができる。印字機構としては、タイプ・ホイール式と、タイプ・チェーン式がある。前者は、横方向に文字が不ぞろいになりやすく、後者の場合は、逆に縦方向に不ぞろいになりやすい。タイプ・ホイール式は、活字をいっぱいはめ込んだ円筒が回転し、タイプ・チェーン式は、活字をチェーン状にしたものが、回転している。中央処理装置から伝達されてきた情報に対応する活字が、印字位置に来た時に、タイミングよく、用紙の裏からハンマで打つことにより、印字される。印字速度は、1分間に200行から1,000行位である。

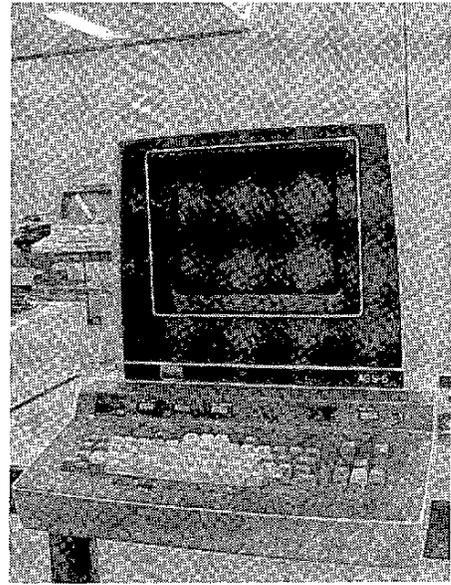
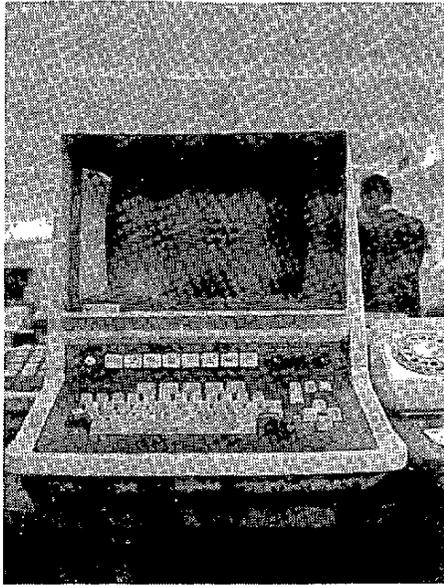
図10 ラインプリンタの構造と本体



## 2-2-5 映像表示装置

映像表示装置（CRTディスプレイ装置）は、図形、グラフ、文字、数字などの形で入出力できる装置である。入力方法としては、コンソールのキーボードからタイプして入力する方法と、ライト・ペンを使用して、ブラウン管上から入力する方法がある。出力方法は、中央処理装置からの情報を、ブラウン管上に図形やグラフや文字などの目で見える形で、表わす

図 1 1 映像表示装置



方法である。映像表示装置を機能的に分類すると、文字と数字のみが表示できるキャラクタ・ディスプレイ装置と、図形やグラフも表示できるグラフィック・ディスプレイ装置とに分けられる。

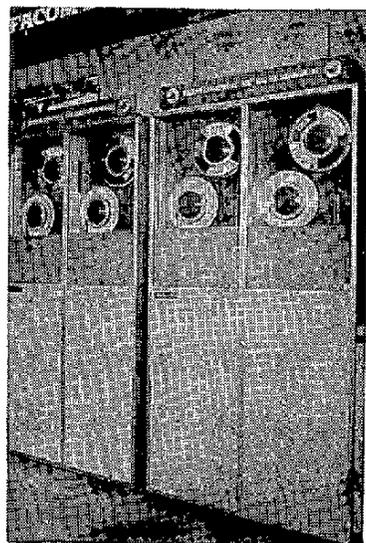
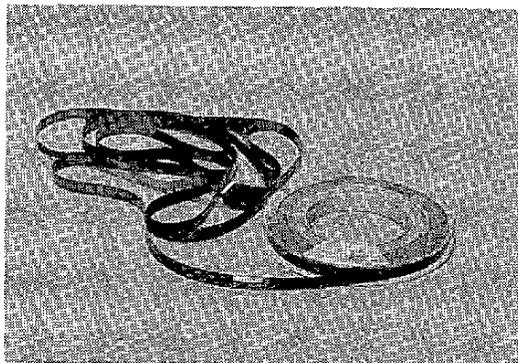
### 2-3 補助記憶装置

主記憶装置の記憶能力には限度がある。そこで、主記憶装置に記憶されているデータやプログラムを取り出して、別のところに記録、保存しておく必要がある。このように、主記憶装置を補助する機能を持った装置のことを、補助記憶装置または外部記憶装置という。補助記憶装置として代表的なものとして、磁気テープ装置、磁気ディスク装置、磁気ドラム装置がある。

#### 2-3-1 磁気テープ装置

磁気テープ装置は、入力、出力の両方の機能を持った装置である。磁気テープ装置は、データやプログラムを磁気テープに書き込んだり、磁気テープに書かれているデータや、プログラムを読み取ったりする装置である。磁気テープは、オープン・リールの録音テープと同じようなものである。幅が約  $13 \text{ mm}$  位で、長さが  $2,400$  フィート (約  $730 \text{ m}$ )、 $1,200$  フ

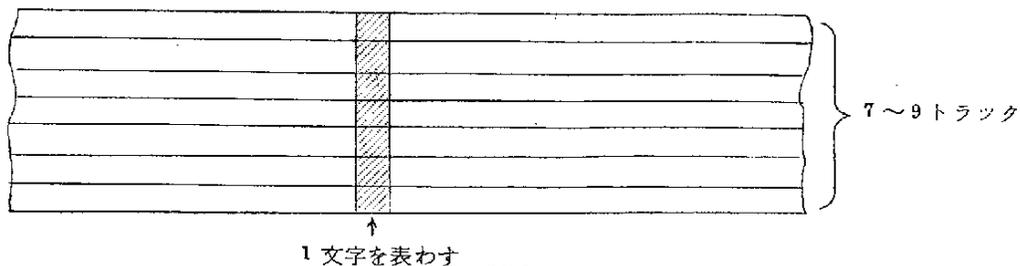
図 1 2 磁気テープと磁気テープ装置



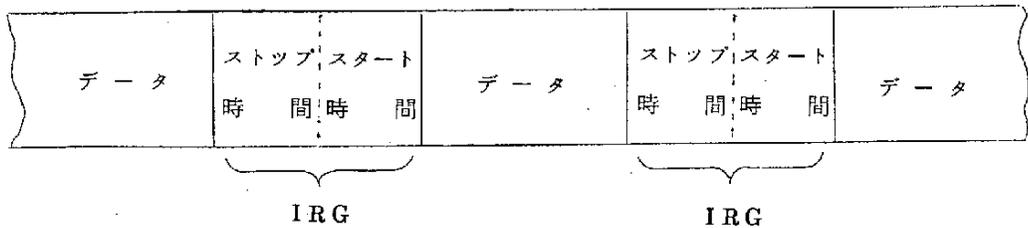
バイト、300バイトなどの種類がある。

磁気テープは、磁気テープの縦の1行で、1文字を表現する。この1文字に、磁気ヘッドの通過する位置が、7つあるものを7トラック、8つあるものを8トラック、9つあるものを9トラックという。

磁気テープは、動き出してから一定のスピードに達するまでに、時間（スタート時間）が必要である。また、停止するのにも、時間（ストップ時間）がかかる。そこで、データとデータの間には、スタート時間とストップ時間の部分が生じるわけである。これをIRG（Inter Record Gap）といい、通常、2cmほどの長さが必要である。



磁気テープ装置は、磁気テープを取りかえるだけで、大量のデータを記録することができる。シーケンシャルにデータ処理してよいものは、一般に磁気テープにファイルされ保存される。最近では、各種の台帳や伝票や帳簿などが、磁気テープに置きかえられてきている。



磁気テープの特徴としては、つぎのようなことがあげられる。

- ①ファイルの形式がシーケンシャルである。
- ②記録密度が高い。
- ③保存場所が比較的小さくてすむ。
- ④価格が安い。

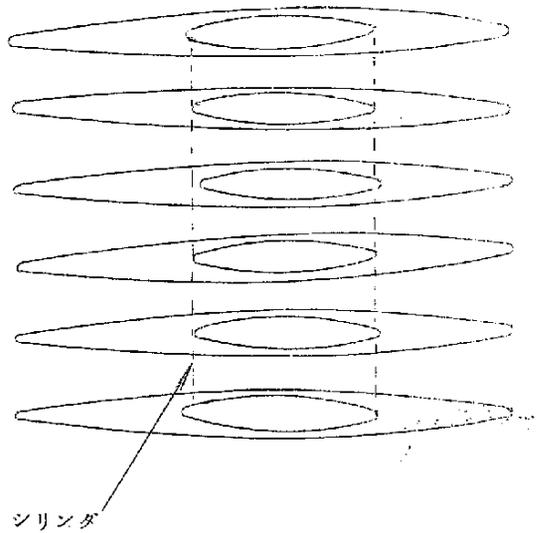
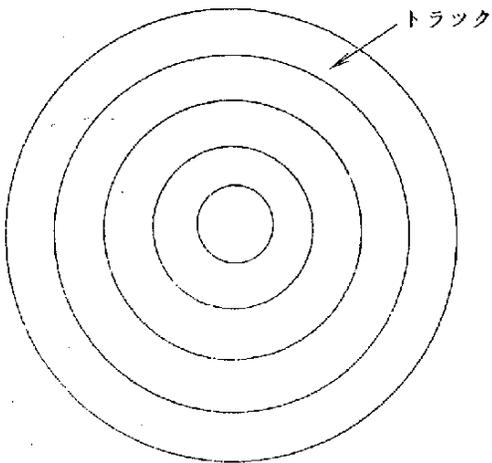
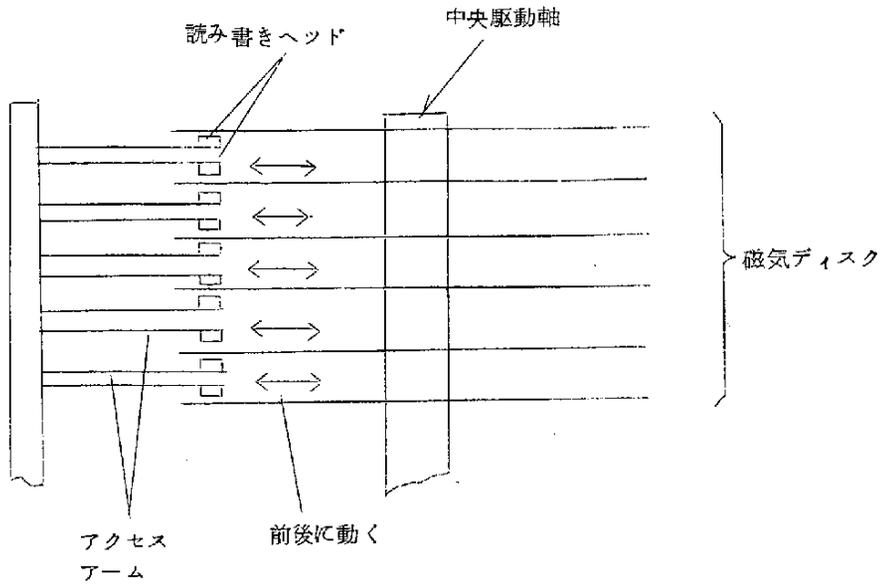
### 2-3-2 磁気ディスク装置

磁気ディスク装置は、表面と裏面を、磁性物質で被ったレコード盤のような円盤を、数枚重ねたものを回転させ、データやプログラムなどを記録する装置である。円盤の表面と裏面には、アクセスアームが出ている。アクセスアームの先端には、読み書き用の磁気ヘッドがついている。このヘッドをとおして、データなどが円盤上に書き込まれたり、記録されているデータなどを読み取ったりする。

レコード盤は、溝が1本の渦巻線になっているが、磁気ディスクは、同心円になっている。この同心円の1つ1つを、トラックという。また、各盤上の同じトラックを、円筒に見たてて、それをシリンダという。

磁気ディスク装置には、数枚重ねあったレコード盤のようなものが、固定されているものと、取りはずしが出来るものがある。この取りはずしが出来るレコード盤のようなものを、ディスクパックという。したがってディスクパックを取りかえることにより、数億、数十億といった多量のデータも、記録保存することができるわけである。

図 1 3 磁気ディスクの構造

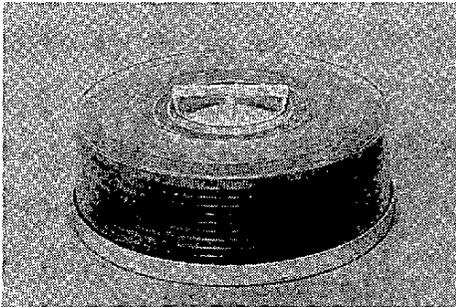


磁気テープにデータを書き込んだり、磁気テープからデータを読み込んだりする場合、磁気テープは、最初から順番に読み書きしなければならないが、磁気ディスク装置は、円盤のようなものが同心円で回転しているため、直接データを読み書きすることができる。このような処理方式をランダム処理という。

磁気ディスク装置の大きな特徴としては、つぎのことがあげられる。

- ①ランダム処理ができる。
- ②磁気テープ装置より読み書き速度が速い。
- ③記録されているデータを部分的に書きなおすことができる。

図 1 4 ディスクパックと磁気ディスク装置

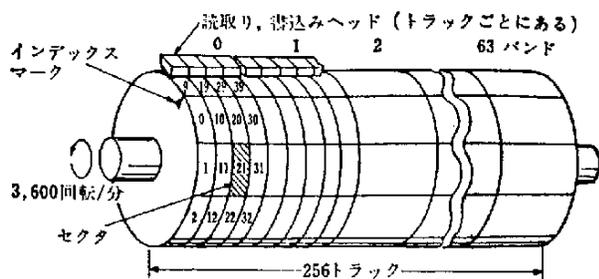


### 2-3-3 磁気ドラム装置

磁気ドラム装置は、表面が磁気被膜でおおわれた金属製の円筒を回転させ、その表面にデータを書き込んだり、読み込んだりする装置である。

磁気ドラムもデータなどの読み書きは、磁気ヘッドをとおして行なわれる。また磁気ディスク装置と同様に、ランダム処理ができる装置であるが、ディスクパックのように、取りはずして使用することはできない。

図 15 磁 気 ド ラ ム



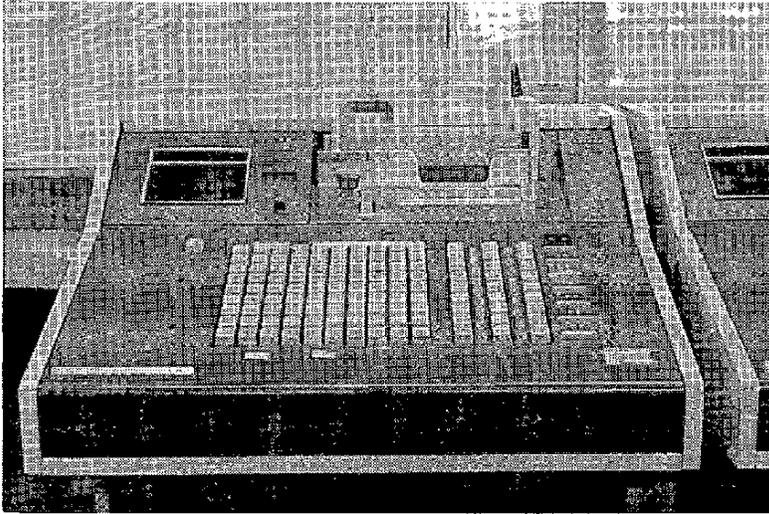
#### 2-4 データ通信

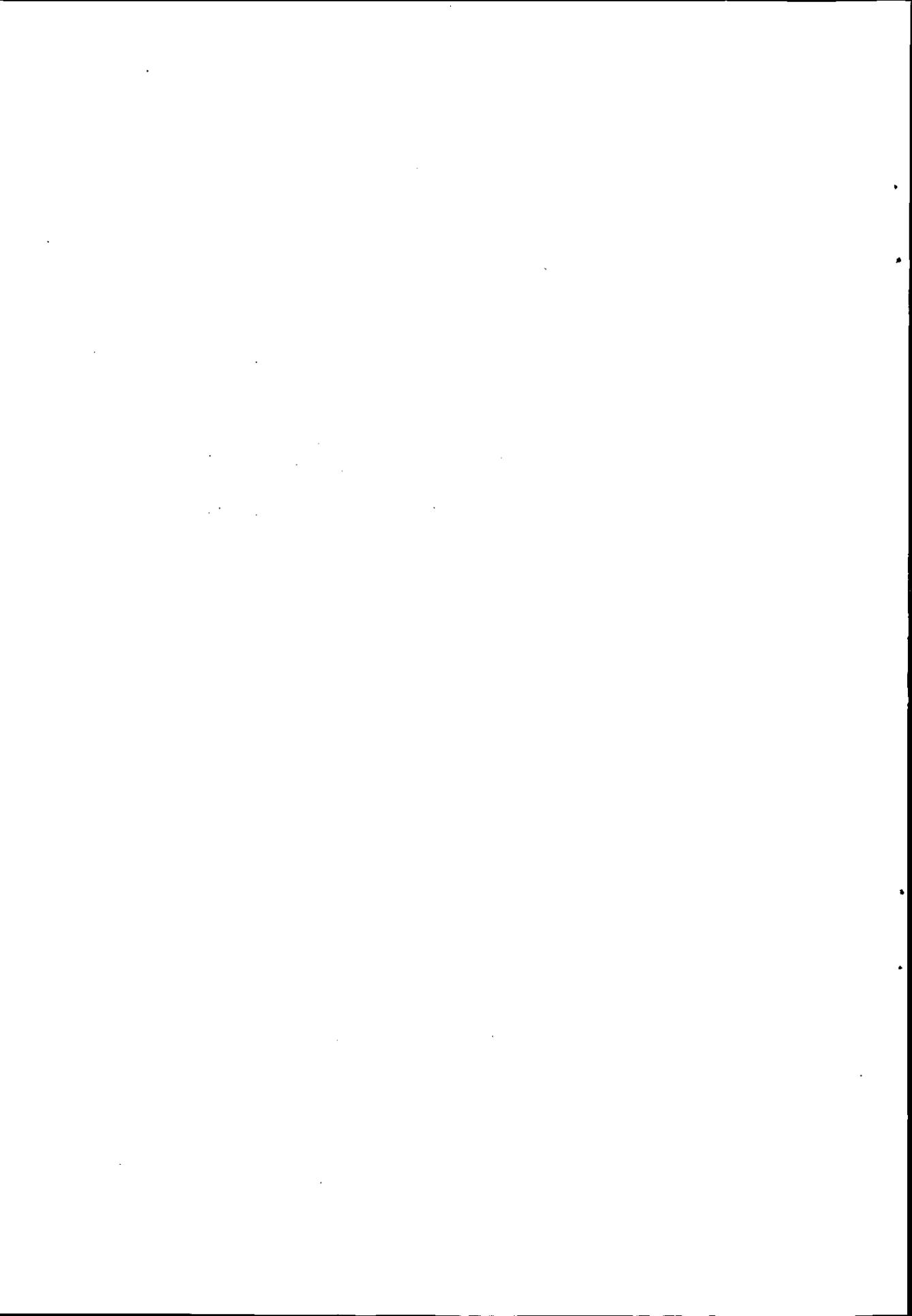
データ通信とは、中央のコンピュータと遠隔地の端末装置とを、電信や電話回線などの通信施設を使って結び、相互にデータを交換することをいう。広域に分散している支店、工場などで発生したデータを、正確にそして迅速に送る必要性がでてきた今日、データ通信は、重要な地位をしめるようになってきた。データ通信を使用しない場合には、データが発生した場所で、紙テープやカードに穿孔し、中央のコンピュータ室に送る方法と、原始伝票を中央のコンピュータ室に送り、そこでカードなどに穿孔する方法とがある。いずれにしても、データを郵送や運送する必要があるため、コンピュータにデータを入力するまでの時間がかかりすぎる。出力の場合も同様で、中央のコンピュータ室で作成された報告書やレポートなどを各地に郵送しなければならず、これも時間がかかる。これでは、コンピュータのもつ高速性が半減されてしまう。そこで、データの発生した場所で、端末装置から直接データを入力し、通信回線を通して、中央のコンピュータにデータを入れたり、逆に中央から遠隔地の端末装置に、直接出力するデータ通信の方法がとられるのである。

通信回線には、データを一方向だけにしか送ることができない単一回線や、両方向へデータを送ることができるが、同時に送ることができない半二重回線や、両方向へ同時に、データを送ることができる全二重回線がある。

データが発生した時、すぐに端末装置からデータを入力し、処理するものを、リアルタイム処理といい、ある程度データを貯め、一括処理するものをリモートバッチ処理という。リアルタイム処理は、経済的に高くつくため、一般にはリモートバッチで処理されるケースが多い。

图16 端末装置

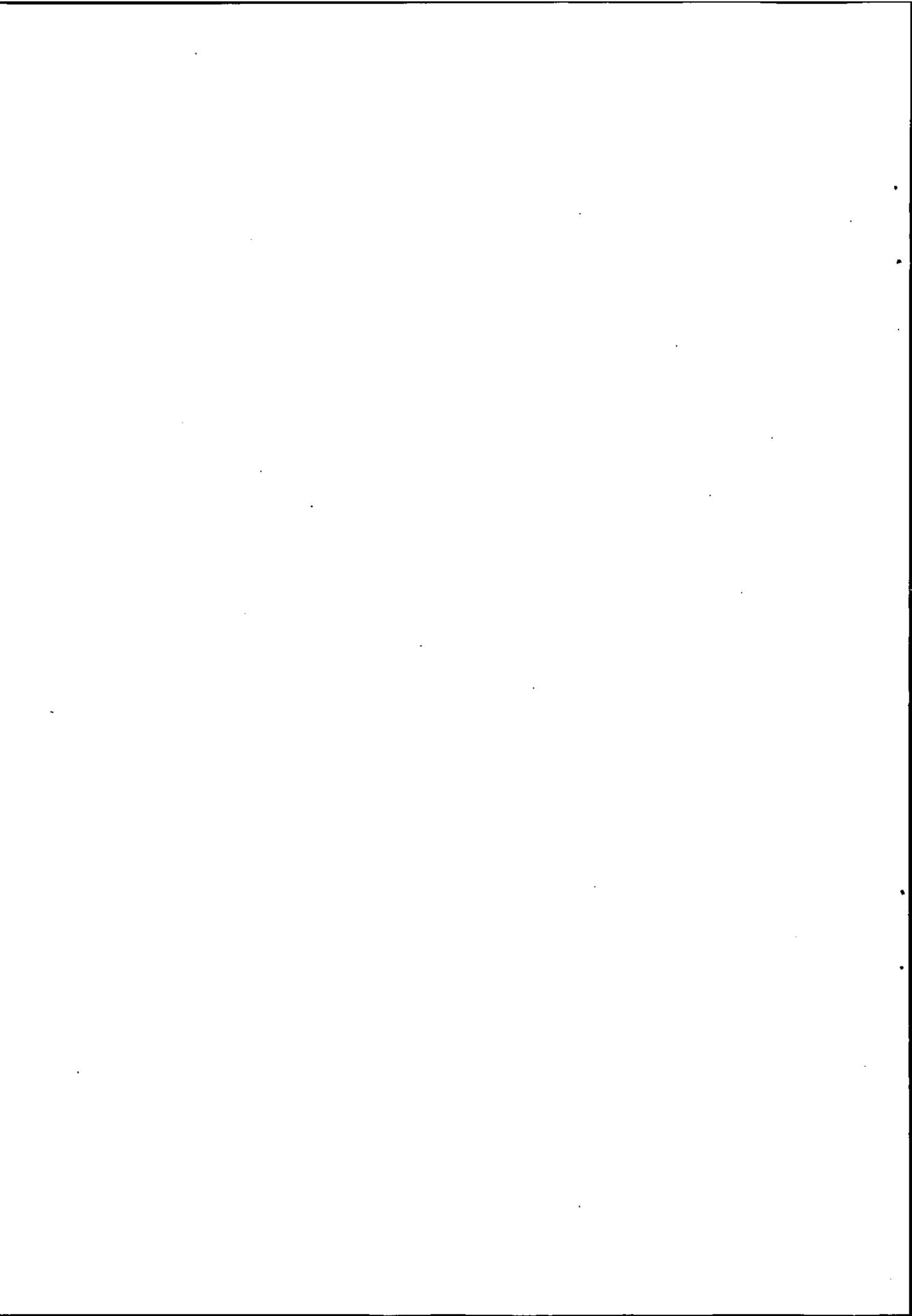




第 5 部

自 習

シ ス テ ム 設 計

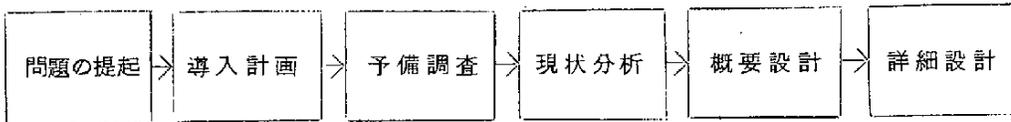


# 目 次

1. システム設計 .....	240
1-1 出力(アウトプット)の設計 .....	243
1-2 入力(インプット)の設計 .....	245
1-3 ファイルの設計 .....	248
1-4 コードの設計 .....	251
1-5 コスト分析 .....	258
2. 演習問題 .....	259

## 1. システム設計

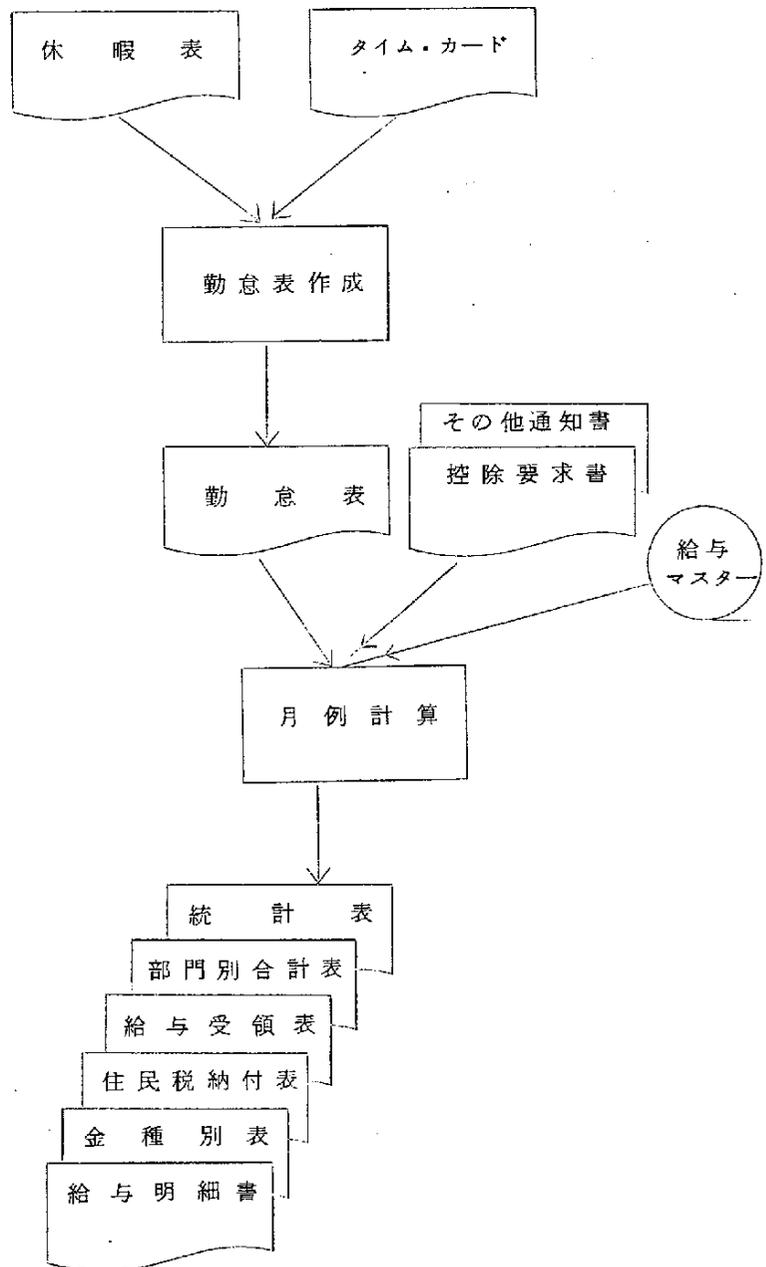
システム設計の手順は、問題の提起に始まり、導入計画、予備調査、現状分析、概要設計、詳細設計の順に行なわれる。



現状分析が終ると、基本システムの設計を行なう。これには、システム設計者の基本構想だけでなく、経営者の基本方針なども加味した上で、設計しなければならない。

基本システムの構想は、つぎのような流れ図（フローチャート）に表わされる。

図1 流れ図の例……給与計算の月例計算業務



基本システムの流れ図が出来れば、システムの概要設計、システムの詳細設計へと、作業を進めていく。ここでは、概要設計に関して学習する。

概要設計とは、現状調査と分析の結果、明らかになった現状の情報処理システムの問題点を解決し、企業目的に適合した情報システムの概要を定めていく過程のことをいう。

システムの概要設計の手順としては、出力（アウトプット）の設計から入り、入力（インプット）の設計、ファイルの設計、コードの設計を行ない、最後にコスト分析（またはシステムの評価）を行なうのが、一般的である。

## 1-1 出力の設計

出力の設計でまず考えなければならない事は、出力の目的は何かという事である。何のために出力するのか、誰のために出力するのか、出力を使用する人にとって価値ある情報もり込まれているかなどといった事を明確にしておく必要がある。そこで、つぎのような項目について個々に検討する。

### ① 対象業務

出力をしなければならない対象業務は何であるのか、また、どの範囲までを対象とするのかといった事を整理する。これは、システムの基本設計ができた時に自動的に決定されることであるが、再確認しておくといふ。

たとえば、対象業務に人事管理を取り上げてみても、人事、勤労、厚生などもあり範囲は広い。給与計算を対象にしても、月例の給与計算事務もあれば賞与事務や年末調整事務などもあり、どの範囲までを対象にするかを最初に確定しておく必要がある。

### ② 出力の種類

対象業務の範囲が確定すれば、つぎに対象業務内で必要とされる出力にはどのようなものがあるかを整理する。

月例の給与計算事務には、給与明細書、全種別表、住民税納付表、給与受領表、部門別合計表、統計表などがある。

### ③ 出力の作成周期および時期

各出力について、それぞれの作成周期（日、週、月、季、年）や時期を検討する。コンピュータによる出力の迅速性も大切だが、必要な出力をタイミングよく出力する事は、さらに重要な問題である。

給与明細書は、月に1回のサイクルで、毎月20日に作成するか、部門別合計表はいつ作成すればよいかといったことを検討する。

### ④ 出力の形態

出力の形態としては、帳票、カード、紙テープ、磁気テープ、磁気ディスク、磁気ドラム、映像表示装置などがある。帳票には、

① 見やすいように項目名や、枠どりの線があらかじめ印刷してある定型用紙

② 横線だけが入っている連続用紙

とがある。

ここでは、各出力がどの形態をとればよいかということを検討する。

目で見える形で出力しなければならないものは、普通、ラインプリンタ（行印字機）や、タイプライタによって、用紙に印刷される。外部に出す必要のないものは、連続用紙に印刷されることが多い。給与明細書や納品書などは、見ばえのよい定型用紙に印刷されることが多い。

一時的に、しかも迅速に情報を知る必要のあるものは、映像表示装置に出力される。

コンピュータからの出力情報に、他の情報を追加したものが、新しい入力情報となるシステムを、ターンアラウンド・システムという。このターンアラウンド・システムの一環として、出力が考えられる場合には、しばしばカードが用いられる。

出力の内容を遠方に送付しなければならない場合や、遠隔地に端末装置などの設備があり、データ通信が可能な場合には、紙テープが用いられる。

出力が同時にファイルとして取扱われるものは、磁気テープ、磁気ディスク、磁気ドラムなどに納められる。

#### ⑤ 出力の項目名と桁数

各出力の種類や形態が決定されると、各出力について、どのような項目があるのか、その項目の重要度はどうか、その項目の桁数は何桁必要か、使用する文字の種類（数字、カナ、アルファベット、記号）は何か、などといった事を検討する。

給与明細書の項目には、年月、社員番号、氏名、所属コード、本給、役付手当、時間外手当などがあり、それぞれ必要な桁数がある。

ラインプリンタやタイプライタで出力する場合には、1行に印刷できる桁数を調べておく必要がある。

#### ⑥ 出力の提出先および使用者

出力を使用する人にとって、使い易く、見やすいように設計することも

大切であるが、まずだれがその出力を使用するのかを検討する必要がある。経営者、管理者、顧客などによって、必要な情報が異なり、したがって出力の設計が異なってくることは言うまでもない。

#### ⑦ 出力の処理件数とコピー枚数

出力の種類を整理した後に、各出力の処理件数が何件あるかを検討する。1枚の用紙に出力される統計表のようなものもあれば、給与明細書のように従業員の数だけ出力されるものもある。処理件数を明確にすることにより処理時間が明らかになり、処理コストも算出できる。

控が必要な出力であるかどうか、控が必要な場合は何部必要かということも検討する。

### 1-2 入力設計

適切な出力を得るためにはそれに必要な入力データを作成しておかなければならない。

入力の方法として、普通一般に用いられている方法としては各現場で作成された原始伝票を集め、カードや紙テープに穿孔し、入力する方法である。

入力設計には

- ① 原始伝票の設計
- ② カードなどの設計

の2つがある。

原始伝票は、パンチャが穿孔しやすいものであると共に、まず第1に現場で原始伝票を使用する人々にとって取扱い易いものでなければならない。

カードなどの設計をする場合、つぎの事を検討して設計する。

#### ① 入力の種類

対象業務にはどのような入力があるかを明らかにする。給与計算を例にとると

##### ① 個人からのデータとして

タイムカード、扶養控除申告書、保険料申告書、家族異動届、住所変更など

##### ② 人事部門からのデータとして

入社退社通知書、人事異動表、賞与計算書など

##### ③ 地方自治体からのデータとして

住民税決定通知書など

がある。

## ② 入力周期および時期

ここでは、各入力について、その周期と時期を調べる。タイムカードを何日に締切するのか、カード穿孔を何日までにするのかといったことを検討する。

入力データの発生部門の作業スケジュールや、コンピュータ室およびパンチ室の作業能力なども検討しなければならない。

## ③ 入力の形態

入力の形態には、カード、紙テープ、直接入力帳票（OCR、OMR、MICRで使用する用紙）、端末装置、磁気テープ、磁気ディスクなどがある。入力の形態は、処理する対象業務の内容や、経費の問題や、ハードの機器構成などにより異なってくる。

各入力の特徴をあげると、つぎのようになる。

### 〔カード〕

- 追加訂正が簡単である。
- 検孔ができる。
- 分類機で分類できる。
- 1枚のカードに穿孔できる桁数に制約がある。
- 紙テープよりコスト高である。
- 保管や運搬が困難である。

### 〔紙テープ〕

- 追加訂正が困難である。
- 検孔が困難である。
- 分類できない。
- 桁数に制約がない。
- 保管や運搬が容易である。
- 伝票発行と同時に紙テープを作成することが出来る。
- データ伝送が可能である。

### 〔直接入力帳票：OCR、OMR、MICR〕

- パンチャが不用である。

- コストが高つく。
- 取扱いに制約がある。

〔端末装置〕

- 直接、入力できる。
- 入力する場所が端末装置のある所に限定される。
- コストが高つく。

〔磁気テープ、磁気ディスク〕

- カードより操作しやすい。
- 入力が速い。
- 少量のデータを取扱うには不向きである。
- データの内容を目で見ることができない。

④ 入力の項目名と桁数

各出力を作成するためにどのような入力項目が必要なのか、各項目はそれぞれ何桁必要なのか、使用する文字の種類は何か、といったことを検討する。

給与計算の勤怠カードの項目として

カード・ナンバー、社員番号、出勤日数、有給休暇日数、欠勤日数、遅刻早退回数、遅刻早退基準回数、早残時間、休出時間、深夜時間、徹夜代休回数

などが考えられる。

つぎに、上記の項目について、桁数や文字の種類を検討する。

桁数が固定のもの（固定長）を取扱う時はカードが使用され、可変長の時は紙テープが使用されることが多い。

⑤ 入力の発生部門

ここでは、コンピュータで処理をするために必要な入力データが、どの部門で発生するかを明確にする。各種のデータが各部門で発生するが、発生したデータのすべてがコンピュータ処理に必要な入力データであるとは限らない。ある程度人手で加工されたデータが入力データとなることもある。そこで、コンピュータ処理に必要な入力データの最終発生部門はどこか、ということを検討する。

## ⑥ 入力データの件数

各入力について、それぞれ何件のデータが発生するかを調べる。現在使用している伝票の枚数を把握する事や、把握できないものは推定でデータ件数をわり出しておく。

データ件数の把握は、コンピュータの処理時間の算出や、処理コストの算出のためにも必要である。

## 1-3 ファイルの設計

ファイルの設計をする時に検討しなければならないことは、つぎの事である。

### ① ファイルの種類

まず基本システムの設計書にもとづき、どのようなファイルがあるのかを調べる。給与計算システムでは、給与明細ファイル、給与変動ファイル、給与マスターファイルなどが考えられる。これらのファイルが、マスターファイルとして取扱われるものか、トランザクションファイルとして取扱われるものかといった事も同時に明確にしておく。

### ② ファイルの作成および更新の周期と時期

ここでは各ファイルについて、いつ作成するのか、いつ更新するのか、その周期(日、週、月、季、年)はどうかといった事を検討する。売上ファイルのように日毎に更新されるものもあれば給与明細ファイルのように月毎に更新されるものもある。

ファイルの更新時期を考える時に注意しなければならないのは、旧ファイルをいつまで保存する必要があるかということである。台帳や帳簿が磁気テープや磁気ディスクなどに置きかわったシステムでは、ファイルの保存・保管は重要な問題である。何かの手違いや事故や災害などで、ファイルが消されたり、破損したりすることがある。この時、すぐにファイルを再成したり、回復できなければ業務に大きな支障をきたすことになる。すべてのファイルをコピーすると費用がかかりすぎる。

したがって、旧ファイルをいつ消してよいかという問題を慎重に検討しなければならない。

### ③ ファイルの形態

ファイルの形態には、カード、紙テープ、磁気テープ、磁気ディスク

などがあり、それぞれの特徴を整理するとつぎのようになる。

[ カード ]

- 1枚のカードにファイルできる桁数に制約がある。
- 保管、運搬が困難である。
- 追加、訂正が比較的容易である。
- データの内容を目で読むことができる。
- 固定長のデータに向いている。

[ 紙テープ ]

- 桁数の制約がない。
- 保管、運搬が容易である。
- 追加、訂正が困難である。
- 紙質が弱いため、何度も使用することができない。
- 可変長のデータに向いている。

[ 磁気テープ ]

- 長期の保存ができる。
- 記録密度が高い。
- 磁気ディスクや磁気ドラムより安価である。
- 保管の場所が小さくてよい。
- バッチ処理に向いている。
- シークエンシャル（順次）な処理にしか向かない。

[ 磁気ディスク、磁気ドラム ]

- 必要なデータを、直接取り出したり、記録したりすることができる。
- 同心円で回転しているため、アクセス時間が均等である。
- 即時処理に向いている。
- 複数のファイルを一時に更新できる。

以上の各形態の特徴を理解すると共に、ファイルの更新（追加、訂正、削除）が頻繁に行なわれるものであるか、そのファイル自体が頻繁に使用されるものであるのか、ファイルの大きさはどうか、将来そのファイルはどの程度拡張されていくものかなどといったファイルの特性を加味した上で検討し

なければならない。

#### ④ ファイルの項目名と桁数

各ファイルには、それぞれどのような項目が必要であるか、また、それらの項目の中で必要度、利用度の高いものはどれかといったことを検討する。一般に、ファイルを設計する場合、必要度の高い項目から並べられることが多い。各項目の桁数や文字の種類も、同時に明確にしておけば、1件分のデータの長さが明らかになり、しいては、ファイルの大きさも把握できる。

#### ⑤ ファイルの件数

ここでは、ファイルすべきデータの件数を把握し、何件のファイルがあるかを明確にする。データの件数とデータの長さから、ファイル全体の大きさを知ることができる。したがって、そのファイルが1本の磁気テープ（または磁気ディスクなど）に納まるかどうか、または、何本必要かということがわかる。

#### ⑥ ファイルの編成方法

どのようなファイルに編成するかという問題は、ファイルを作成するための時間や費用、データ量などにより異なってくる。また、データの検索やファイルの使用頻度により編成方法を考えなければならない。

ファイルの編成方法には、**①**シークエンシャル・ファイル、**②**インデックスド・シークエンシャル・ファイル、**③**ランダム・ファイルなどがある。

その特徴はつぎのようなことである。

[ シークエンシャル・ファイル : Sequential file ]

- ファイルの多くの項目を、一度に処理するのによい。
- 必要な項目をランダムに抽出することが困難なため、情報検索的な処理には不向きである。
- 追加処理や削除処理をする場合、ファイルの全部を書き直さなければならない。

〔インデックスド・シークエンシャル・ファイル

： Indexed Sequential file 〕

- シークエンシャル処理とランダム処理の両方を処理することができる。
- オーバーフローエリアを使用することにより、項目の追加が容易にできる。
- 索引しながら処理をしていくので処理時間が長くなる。

〔ランダム・ファイル：Random file 〕

- 情報検索が容易にできる。
- ファイルを拡張しやすい。
- 記憶エリアの利用効率が悪い。

⑦ ファイルのレーベル

磁気テープや磁気ディスクなどに、集中的にファイルされたデータが破損したり消されたりしては、業務が停滞し混乱をおこすだけである。したがって、ファイルの保護策が必要となってくる。その一つはファイルの管理上の保護であり、他の一つはプログラム上の保護である。

プログラムによる保護は、まず各ファイルに特別なレーベルを付けておく。つぎに、プログラムでレーベルが一致しているかどうかを調べ、処理に必要なファイルかどうかを判断する。

ここでは、各ファイルにどのようなレーベルを付けるかを検討し整理しておくのである。

1-4 コードの設計

各データに記号を付けることにより、データの識別、分類、照合、集計などがしやすいように付けられたものをコードという。

コードをいかに統一的に、しかも利用しやすいように設計するかという事がここでの問題である。

コード設計で考慮しなければならないのは、つぎのような事である。

まず、コードは、統一された体系のもとで設計されなければならない。使用する記号の種類や桁数などすべて全社的に考えるべきものである。つ

ぎにコードは簡潔にする。コードは多くの人に使用されるものであるので使う人の立場になって設計されなければならない。そこでできるだけ簡潔なコードが望ましい。

コードは、事業や取扱商品の拡張などと共に量的に増加する可能性がある。それに対処できるように、コードの拡張性も考慮に入れる必要がある。またコードは、1つの対象物に対して1つのコードでなければならないことは、いうまでもないことである。コンピュータで処理しやすいコードにすることも大切なことである。

以上のことがらに注意しながら、つぎのような手順で、コードの説計をする。

#### ① コードの目的

何のためにコードが必要なのか、そのコードを使用することによりどのようなメリットがあるのかといった問題を明確にし、その目的を達成するために、都合のよいコードを設計する。

#### ② コードの付番件数

コードを付けなければならない対象物の件数は、現在どれ位あるのか、将来どれ位の幅で増減するのかといったことを検討する。

#### ③ コードの有効期間

一度設定したコードが永久的に不変であるということはなく、絶えず変化がある。またコードが目的に合わなくなり消滅する場合もある。したがってそのコードがいつまで有効であるかを明確にしておく必要がある。

#### ④ コードの種類

コードの種類には、順番コード、区分コード、桁別コード、10進コード、表意コードなどがある。しばしば、それらが組合わされて使用されている。どのコードを使用するかはコードの目的によって決まってくる。たとえば、大分類、中分類、小分類といった分類処理が頻繁に行なわれるものに付けるコードは、分類機能を持ったコードである必要がある。そこで各コードの機能を検討し、設計しなければならない。

⑤ コードの付け方

コードを付ける方法には、つぎの3種類の方法がある。

- コードを付ける専任担当者による方法
- 専任担当者以外の人による方法。
- コンピュータで自動的に付ける方法。

⑥ コードの管理

コードは変更や追加が発生する。この変更や追加処理を正確に、しかも迅速に行なう必要がある。このために、コードの管理担当者を決めコード一覧表を作成して管理するケースが多い。

出力、入力、ファイル、コードの概要設計をする場合、つぎのような設計書に整理し、検討するとよい。

[日本電気マニュアルES-5001-1より]

表1

## 出力設計書

1/2

資料 No	アウト・プットの名称		発行周期	発行時期	形態		
KO101	給与明細書		毎月	25日	LP		
作成部数	発生件数	レコード/件	件数/頁	用紙の種類			
2部	8,000	3/2	6	印刷用紙 裏カーボン			
送付先・使用者		送付方法	使用後の処理				
給与課		メールボーイ	1部を3年間保存後焼却				
順序	項目	桁数	文字区分	順序	項目	桁数	文字区分
1	年月	4	数値	11	減額	4	数値
2	社員番号	6	"	12	支給額	6	"
3	氏名	16	カナ文字	13	失業保険料	4	"
4	所属コード	4	数値	14	厚年・健保料	4	"
5	本給	6	"	15	所得税	5	"
6	扶養給	5	"	16	住民税	5	"
7	役付手当	5	"	17	組合費	4	"
8	時間外手当	5	"	18			
9	管精動手当	4	"	19	通勤費	4	数値
10	その他給与	6	数値1桁は符号	20	預金控除額	5	"
チェック項目	本給+扶養給+役付手当+時間外手当+管精動手当+その他給与(1)-減額=支給額 支給額-失業保険料-厚年・健保料-所得税-住民税-組合費+その他給与(2)+通勤費-預金控除額-貸付金返済額-家賃支払額-生協代金支払額-保険支払額-その他控除額+前月繰越額-翌月繰越額=振込額			特記事項	早寝・休出・深夜時間は小数点を含まず5桁で小数点以下2桁である。 氏名はソフト・コードを含まず、その他の項目は数値のみの桁数である。		
集計				区分			
	項目		項目		項目		項目
1		2		3		4	

表 2

## 入 力 設 計 書

データ No	データ名	インプット周期	使用後の処理	形態					
K4-10	勤怠カード	月	半年後廃棄	カード					
原始帳票名		発生部署	推定発生量						
勤怠表		給与課	MAX	MIN X 8,000					
No	項目	桁	固可	備考	No	項目	桁	固可	備考
1	カード No	5	固	K4-10	14				
2	社員番号	6	"		15				
3	出勤日数	2	"		16				
4	有給休暇日数	2	"		17				
5	欠勤日数	2	"		18				
6	遅刻・早退回数	2	"		19				
7	遅刻・早退基準回数	3	"		20				
8	早・残時間	5	"		21				
9	休出時間	5	"		22				
10	深夜時間	5	"		23				
11	徹夜代休回数	2	"		24				
12					25				
13					26				
チ エ ッ ク 項 目					特 記 事 項				

表 3

## フ ァ イ ル 設 計 書

1 / 2

ファイル No		ファイル名		作成・更新サイクル		新 別		形 態	
KF・C4		給 与 明 細		① 更 月		M・T (P・I)		M・T	
レコード件数	字/レコード	固・可区分	件/レコード	字/件	保 存 期 間				
1,600	900	①・V	5	180	3ヶ月				
No	項 目	桁数	固可区分	備 考	No	項 目	桁数	固可区分	備 考
1	社 員 番 号	6	固		13	時 間 外 手 当	5	固	
2	所 属 コード	4	#		14	減 額	4	#	
3	氏 名	18	#	カナ文字で、前後に上下段シフトコード有	15	休 出 時 間	5	#	
4	本 給	6	#		16	深 夜 時 間	5	#	
5	扶 養 給	5	#		17	徹 夜・代 休 回 数	2	#	
6	役 付 手 当	5	#		18	そ の 他 給 与	5	#	
7	出 勤 日 数	2	#		19	前 月 繰 越	2	#	
8	有 給 休 暇 日 数	2	#		20	通 勤 費	4	#	
9	欠 勤 日 数	2	#		21	支 給 額	6	#	
10	遅 刻・早 退 時 間	5	#	基準回数×6分	22	失 業 保 險 料	4	#	
11	早・残 時 間	5	#		23	厚 年・健 保 料	4	#	
12	皆・精 動 手 当	4	#		24	所 得 税	5	#	
分 類 項 目								期 限 後 処 理 方 法	
①	社 員 番 号	②	所 属 コード 役 職 コード 社 員 第 号						
レ ー ベ ル 特 記 事 項					関 連 帳 票・フ ァ イ ル 類				
ヘッダー	作 成 年 月				(F) 給与マスター・ファイル、給与変動ファイル (O) 給与明細書、金種別表、給与受領表、住民税 納付区分表、部門別合計表、総計表				
トレーラー									

表4

## コード設計書

コードの名称	一番方式	付番件数	桁数
所属コード	桁別式	80件	4桁
使用目的			
<p>給与明細書等、個人に配布する帳票類は、部門別にまとめて、作成するのも、事後処理を容易にする。また原価管理に関係する部門別合計表等の帳票作成に使用される。この様に部門別の統計を必要とする時に使用する。</p>			
構成・付番方法			
<p style="text-align: center;"> <math>A_1</math> <math>A_2</math>                      <math>A_3</math>                      <math>A_4</math>  <span style="border: 1px solid black; padding: 2px;">事業部</span>                      <span style="border: 1px solid black; padding: 2px;">部</span>                      <span style="border: 1px solid black; padding: 2px;">課</span> </p>			
<p>事業部 (<math>A_1 A_2</math>)</p> <p>本社所属の部、および事業部の区分を10進数2桁で行なり。現在これに属するものは11ヶある。</p> <p>部 (<math>A_3</math>)</p> <p>各事業部ごとに、その事業部に所属する部の区分を10進数1桁で行なり。なお、本社所属の部に対しては、この桁は、0にする。</p> <p>課 (<math>A_4</math>)</p> <p>各部ごとに、その部に属する課の区分を10進数1桁で行なり。</p>		<p>所属コードは、おもに部門別の統計を表わす為に使用され、大、中、小の分類の必要性がある。また機構改正等により、増減も考えられるので、10進組別式コードを採用する。各組の構成は前述の通りである。</p>	
その他			

## 1-5 コスト分析

ここでは、コンピュータを使用することによる費用と効果を算出して評価する。

コストとして考えられるものには、システムを開発する費用とシステムを運用する費用とに分けられる。

### ○システムを開発する費用

人件費、機械使用料、設備費など、

### ○システムを運用する費用

人件費、機械使用料、設備費、償却費、消耗品費、一般管理費など、コストを算出するためには、まず処理時間を見積らなければならない。処理時間の計算は、普通、中央処理装置の所要時間と、周辺装置の中で最も時間のかかるものとを比較して、大きい方の時間を処理時間とすることが多い。つぎに業務の日程計画をたて稼働時間を算出する。以上から、コンピュータの使用料が明確になる。

つぎに、それに要する人件費や消耗品費や一般管理費などのコストを算出する。

効果には、直接的な効果と間接的な効果とがある。

### 〔直接的効果〕

- 人件費、事務コスト、在庫費などの節減
- 仕掛品の減少
- 能率の向上、スピードアップ

### 〔間接的効果〕

- 組織、事務などの簡素化
- 管理の質的向上
- サービスの向上
- 信用の増加

## 2. 演習問題

### 〔演習1〕

貴社の給与システムを流れ図で表わして下さい。

### 〔演習2〕

貴社用の出力設計書、入力設計書、ファイル設計書、コード設計書を開発して下さい。

### 〔演習3〕

貴社の給与明細書の概要を出力設計書に整理して下さい。

### 〔演習4〕

上記の給与明細書を出力するためには、どのような入力データが必要ですか。

### 〔演習5〕

演習1のシステムを運用するためのコストを計算して下さい。

請求 番号	経 46-22(5)	登録 番号			
著者名	↑				
書名	一般管理者向け企業向けIT教育テキスト (Aコース用)				
所属	帯出者氏名	貸出日	返却 予定日	返却日	

禁無断転載

昭和47年11月

発行人 日本経営情報開発協会

東京都千代田区霞ヶ関3-2-5

(霞ヶ関ビル30階)

TEL (531)6401

印刷 三州社

東京都港区芝大門1-1-21

TEL (433)1481(代)

