53－S 001

分散型リソース処理技術の研究開発

昭和54年3月

JIPDEC

財団法人 日本情報処理開発協会

この報告書は，日本自転車振興会から競輪収益の一部である機械工業振興資金の補助を受けて昭和５３年度に実施した「分散型リソース処理技術の研究開発」の成果をとりまとめたものであります。

序

当財団は情報処理技術の研究開発の一環として、昭和５２年度より「分散型リソース処理技術の研究開発」に着手いたしました。

コンピュータ利用の高度化および複雑化にともない、ハードウェア、ソフトウェア、データベース等のリソースの集中処理はほぼ限界に達してまいりました。

分散処理システムの出現によって大量のデータおよび複雑な処理が可能になった反面、新たに発生した異機種に分散しているデータおよびプログラムをどのようにして効率よく、かつ容易に利用するかという問題の解決に迫られております。このような問題を解決するためには、分散処理システムのリソース全体を総合的にとらえる必要があります。

また、わが国の情報処理の特徴の一つである日本語情報処理について、低コストで操作性にすぐれた日本語端末の開発が必要であります。

当財団は、コンピュータ・ネットワークＪＩＰＮＥＴを有し、過去において日本語情報処理の研究開発をおこなっており、このような問題解決のための研究開発を実施しうる環境をそなえていると判断いたし、昭和５２年度より３ケ年計画で、分散処理におけるリソースの統合に関する研究開発をおこなうことといたしました。

本年度は、第二年度目にあたり、データベースの統合利用方式およびＮＤＤ（ネットワークデータディレクトリ）のシステム設計とモデルの作成、技術的諸問題の解明、海外調査を行ない、さらには日本語端末の機能検討と基本設計を行ないました。

本書は、研究開発の中間報告としてその成果をまとめたものであります。

本報告書がこの方面に興味ある方々に広く利用され、わが国情報処理技術向上の一助として寄与できることを念願いたす次第であります。

昭和５４年３月

財団法人　日本情報処理開発協会

会　長　上　野　幸　七

分散型リソース処理技術の研究開発

目　　次

I　分散型データベースシステムの研究開発

1. 総　　論 …… 1
2. 分散型データベースシステム …… 4
3. ネットワークデータディレクトリ …… 74
4. 海外の技術動向 …… 114
5. 論文・資料紹介 …… 177

II　日本語情報処理技術の研究開発

1. 総　　論 …… 189
2. 日本語情報処理技術の課題 …… 194
3. 日本語端末用文字発生方法の検討 …… 212
4. 日本語端末用入力方法の検討 …… 243
5. 日本語端末の基本設計 …… 274

I　分散型データベースシステムの研究開発

1. 総　論 …… 1
1.1 研究の背景と目的 …… 1
1.2 概　要 …… 3
2. 分散型データベースシステム …… 4
2.1 概　要 …… 4
2.2 グロス・アーキテクチャ …… 5
2.2.1 4層スキーマの記述 …… 7
2.2.2 スキーマのインプリメンテーション …… 8
2.3 ＬＩＳとＬＣＳ記述 …… 8
2.3.1 データモデルの一般構造 …… 8
2.3.2 ＬＣＳ記述 …… 9
2.3.3 ＬＣＳのリレーショナル記述 …… 11
2.3.4 変換例 …… 11
2.4 アクセス言語 …… 14
2.5 異種性問題 …… 16
2.5.1 問合せ変換 …… 16
2.5.2 異種性情報 …… 50
2.5.3 その他の問題 …… 51
2.6 分散問題 …… 52
2.6.1 セマンティックリンク …… 52
2.6.2 分散記述 …… 54
2.6.3 分散情報 …… 61
2.6.4 全体一局所問合せ分割 …… 63
2.7 システム・アーキテクチャ …… 71
2.8 ま と め …… 73
3. ネットワークデータディレクトリ …… 74
3.1 クリアリング情報生成システム／異種性情報生成サブシステム …… 74
3.1.1 処理概要 …… 75
3.1.2 ＬＣＳ／ＨＩ記述 …… 78
3.1.3 リレーショナル記述 …… 93
3.1.4 例　題 …… 94
3.2 クリアリング情報生成システム／分散情報生成サブシステム …… 103
3.2.1 ＤＤとその木構造表現 …… 103

3.2.2 対応情報 …… 111
3.2.3 スキーマ情報 …… 112
3.2.4 所在情報 …… 113
4. 海外の技術動向 …… 114
4.1 目 的 …… 114
4.2 日 程 …… 114
4.3 調査報告 …… 115
4.3.1 UC. Berkeley …… 115
4.3.2 SRI …… 124
4.3.3 CNRS …… 132
4.3.4 IRIA …… 138
4.3.5 Univ. of Grenoble …… 144
4.3.6 GMD …… 149
4.3.7 GLC …… 155
4.3.8 Berkeley Workshop と VLDB …… 162
4.3.9 まとめ …… 168
付記1. 入手資料 …… 168
付記2. 主要面接者 …… 174
5. 論文・資料紹介 …… 177
参考文献 …… 181
略語表 …… 185

1 総　　論

本章では当研究開発の背景とその目的、３ケ年計画の第二年度までの経過報告、来年度の作業項目、本報告書の構成等について記す。

1.1 研究の背景と目的

当財団では、日本自転車振興会の機械工業振興資金の補助を受けて、昭和４８年度より４年間に渡ってコンピュータネットワークＪＩＰＮＥＴを開発した。この技術成果として、インプリメント側に立ったコンピュータネットワークに関する多くのノウハウを得、それを発表した。過去４年間に渡って４０人年以上の工数をかけて作り上げたＪＩＰＮＥＴは、ホストおよび端末装置からのリソース（資源）の共有を可能ならしめたが、コンピュータになじみの薄い利用者の立場からの意見では、利用し易い形でサービスを受けようとした場合に、利用したいホストのシステムのマニュアルを読まなければならない等の少なからぬ困難さをともなうものであった。

確かにコンピュータネットワークによってリソースの共有が可能になったことから、利用者は意志さえあれば多くの情報や処理サービスを利用できるようになった。しかしながら、利用者に対して、それを使用するために多くの情報あるいはスキル等を利用者が持つことを要求することにもなった。

このような背景で、昭和５２年度から３ケ年計画で「分散型リソース処理技術の研究開発」のプロジェクトが発足された。

今までのコンピュータネットワークを中心とした新しいアーキテクチャがインプリメント側あるいは通信側指導型で進められてきたのに対して、このプロジェクトは利用者の立場から容易な利用形態を開発することを第一義に考えた処理技術の研究開発を目標としている。このプロジェクトでは過去にコンピュータネットワークに関して蓄積したノウハウおよび実験台として利用できるコンピュータネットワークＪＩＰＮＥＴを利用して、リソース統合方式（第Ⅰ部）及び日本語情報処理技術（第Ⅱ部）を主たるテーマにして研究開発を進めている。

以下、リソース統合方式の研究開発における第二年度までの経過報告と来年度の作業予定項目を示す。

第一年度目（昨年度）は、

① 分散処理に関する一般的検討

② 分散処理におけるデータ共有方式の検討

③ ＮＣＣ（ネットワークリアリングセンター）に関する検討

④ 標準制御言語に関する検討

等について調査研究を行なった。

①では、まず分散処理出現の背景を明らかにし、分散処理における最も重要なユーザとして一般ユーザ（computer naive user／casual user）を位置づけた。さらには、分散処理の主要なリソ

ースを検討し、データリソース（データベース）が特に重要であるという結論に達した。

②では、ネットワーク技術に代表される分散処理技術を概観し、ＤＢＭＳ出現の背景と歴史を展望した。また、ＤＢＭＳにおける代表的な三つのデータモデル（階層型モデル、網型モデル、関係モデル）の検討を行ない、データモデルがＡＮＳＩ／Ｘ３／ＳＰＡＲＣの提案する三層化（外部スキーマ、概念スキーマ、内部スキーマ）の傾向にあることを示した。分散型データベースシステムには、ＣＯＮＩＴやＥＵＲＯＮＥＴに見られる科学技術文献の共用を目指す専用アプローチと、ＰＯＬＹＰＨＥＭＥ、ＩＮＧＲＥＳ、ＳＤＤ－１、ＬＡＤＤＥＲ等関係モデルに基づくＤＢＭＳを用いた汎用アプローチがあり、当研究開発の方向は後者にあることを明らかにした。

③では、汎用の分散型データベースシステムにおけるＮＣＣ（ネットワーククリアリングセンター）の機能を明確にし、ＤＤ／Ｄ（データディクショナリ／ディレクトリ）とＤＢＭＳの関係、ＮＣＣの基本構成を明らかにした。

④では、異機種のコンピュータから構成するコンピュータネットワークのユーザが、まず直面する問題である制御言語の多様性を解決する技術、即ち、コンピュータの異機種性を克服するために設定した標準制御言語システムを示した。

以上の検討に基づいてリソース統合モデルにおけるデータ共有問題、即ち、分散型データベースシステムにおけるＤＢＭＳ統合の基本的視点を検討した。つまり、システム実現のための概念モデルと言語、分散問題と分散機能、異種性問題と異種性機能等について基本構想を示した。

第二年度（本年度）は前年度の検討に基づき、

① 分散型データベースシステムに関する検討

② ＮＤＤ（ネットワークデータディレクトリ）に関する検討

③ 海外における分散型リソース処理技術の動向調査

等について調査・研究開発を行なった。

①では、コンピュータネットワーク上に分散した異種のＤＢＭＳからなる分散型データベースシステムの検討を汎用・ボトムアップ（general purpose , bottom-up）の立場から行ない、アーキテクチャを確立した。このアーキテクチャの特徴は、分散及び異種の問題を解決するために導入した４層スキーマ構造と概念モデルとしてＥ－Ｒ（事象－関係性）モデルを採用した点にある。

②では、①で確立したアーキテクチャをサポートするシステムとしてＮＤＤを設定し、そのモデルを作成した。ＮＤＤは分散及び異種の問題を解決するための情報、即ち、分散情報及び異種性情報を維持する。これらの情報の記述、蓄積、検索方式等に関する検討を行ない、関係タイプのデータベースによる設計を行ない、モデルの作成を行なった。

③では、最新の分散型リソース処理技術及び評価技術とそのデータ等を収集する目的で、欧米４ケ国７研究機関を調査訪問するとともに、データベースに関する２つの国際会議に参加した。（訪問先：ＵＣ．Berkeley，ＳＲＩ，ＣＮＲＳ，ＩＲＩＡ，Univ. of Grenoble，ＧＭＤ，ＧＬＣ，国際会議：３rd Berkeley Workshop，４th ＶＬＤＢ）

なお、情報処理学会全国大会（第１９回、東京、５３．８）とＩＣＣＣ ’７８（第４回、京都、

53. 9）に於て研究成果の発表を行なった。

本プロジェクトの最終年度にあたる来年度は、

① 分散型データベースシステム実験モデルの作成

② 評価方式の調査・検討・実験・評価

③ 総合報告書の作成

等について行なう予定である。

①では、第二年度目までの検討に引き続いて分散型データベースシステム、特に、問合せ変換（query translation）と問合せ分割（query decomposition）の実験モデルを作成する。

②では、作成したシステムの評価方式について調査・検討・実験・評価を行なう。

③では、3カ年計画で実施した当プロジェクトの総合報告書の作成を行なう。

1.2 概　　要

第Ⅰ部は、「分散型リソース処理技術の研究開発」の一環として推進している、分散型データベースシステムの研究開発についてとりまとめたものである。

第Ⅰ部は、5つの章から構成されており、それぞれ次のような内容について記述している。

第1章「総論」では、当研究開発の背景とその目的について述べている。

第2章「分散型データベースシステム」では、データリソースを主体としたリソース統合システム、即ち、研究開発中の分散型データベースシステムについて述べる。特に、分散及び異種性の2つの問題を解決する4層スキーマ構造について述べている。さらに来年度モデル化を行なう問合せの全体ー局所分割と共通ー固有変換の概要を示している。

第3章「ネットワークデータディレクトリ（NDD）」では、第2章で述べた分散型データベースシステムをサポートするクリアリング情報生成システムについて述べる。このシステムは、異種性情報生成サブシステムと分散情報生成サブシステムの2つのサブシステムから構成されており、それぞれについて概要を述べている。

第4章「海外の技術動向」では、当研究開発の一環として昭和53年8月から9月にかけて実施した海外調査の結果について述べる。調査報告は、訪問した欧米4ケ国7研究機関ならびに参加した2つの国際会議についてそれぞれ述べている。

第5章「論文・資料紹介」では、当プロジェクト発足以来とりまとめた論文及び資料について紹介している。

2　分散型データベースシステム

本章では、4層スキーマ構造に基づいた分散型データベースシステムの全体アーキテクチャについて論じる。

2.1 概　　要

コンピュータネットワーク技術とデータベース管理技術のめざましい進歩により、複数のデータベースをネットワークを用いて統合することの可能性が最近とみに高まってきている。このようなシステムは分散型データベース（DDB）と呼ばれている。既に米国では、CCA社によるSDD－1〔ROTH77〕，UC．Berkeley におけるINGRES〔STON77〕，SRIにおけるLADDER〔SAGA77，HEND77〕等各種の分散型データベースシステムの研究開発が進められている。これらのシステムに共通している特徴は、同種のDBMS（実際には関係DBMS）を統合しようとしている点である。これに対して、仏のGrenoble 大学を中心として開発が進められているPOLYPHEME〔ADIB78b，c〕では、SOCRATE，IMS，DBTGといった異種のDBMSの統合を目指している。他の試みとしては、西独のSttutgart大学のPOREL〔NEUH77〕，Berlin工科大学のDISCO〔MUNZ78〕をあげることが出来る。こうした動きのなかで、我々の目指す分散型データベースシステムは、ネットワークを用いて、異種のDBMSを統合しようとするものである〔TAKI78，HAMA78〕。

分散型データベースを実現するにあたり、まず解決されるべき主要問題として、次の2つがある。

ⅰ）分散問題

ⅱ）異種性問題

分散問題とは、ユーザの必要とするデータが、ネットワーク上のどこに存在するかをどのようにして明らかにするかという問題である。また、ネットワークによって結合されたリソースの異種性をいかに克服するかという問題も同時に考えていかねばならない問題である。分散型データベースの実現において、その構成要素であるDBMSの相違が、最も重要な問題の1つとなっている。よって、異種性とは、DBMSの提供するデータモデルの相違と、言語の相違であると考える。この様なデータモデルと言語の各々の相違を克服することが、異種性問題の解決となる。

上述した問題点を解決するための我々のアプローチは、4層スキーマ構造（FSS：four-schema structure）概念〔TAKI78，79〕に基づいている。同時に、FSSは汎用かつボトムアップ〔ADIB78a〕な分散型データベースを構築するための基本思想でもある。FSSは、局所内部スキーマ（LIS）、局所概念スキーマ（LCS）、全体概念スキーマ（GCS）、及び外部スキーマ（EXS）の4つのスキーマから成り立っている。

各スキーマは、既存DBMSのスキーマレベルから、異種性問題と分散問題とが不可視（invisible）なレベルへの概念化（抽象化）の各レベルに対応している。この概念化（conceptualization）の主要部分は、同種化と統合化の2段階から成っている。第1の概念化は、同種化（ho-

mogenization）と呼ばれるものである。このプロセスにおいて、異なったデータモデルで記述された各サイトのDBMSのスキーマ（LIS）を事象一関係性モデル（entity-relationship model，以下E－Rモデルと記す)〔CHEN76〕を用いて共通記述することによって、各DBMSの異種性が除去される。この記述は、局所管理者（LA）によってなされる。この様にして共通記述された各サイトのスキーマが局所概念スキーマである。

第2の概念化である統合化（integration）とは、同種化された各サイトのLCSから、あるネットワーク共同体にとって意味のあるデータをE－Rモデルを用いて記述することによって、統合するプロセスである。この様にして統合されたスキーマが全体概念スキーマである。この統合化は、データのセマンティックスに基づいて、全体管理者（GA）によって行なわれる。ここで、データの所在を示す分散情報が除去される。

以下、本章では、FSS概念の概要を2.2節で述べ、2.3節では、同種化の例を示す。2.4節では、アクセス言語について触れ、2.5節では、異種性問題を問合せ変換を中心に論じる。2.6節では、統合化を含めた分散問題を論じる。最後に、2.7節において、分散型データベースのシステムアーキテクチャを示す。

2.2 グロス・アーキテクチャ（4層スキーマ構造）

我々が研究開発している分散型データベースシステム（DDB：Distributed Database）は、4層スキーマ構造（FSS）に基づいている〔TAKI78，HAMA78〕。我々は、ANSI／X3／SPARCの3層スキーマ概念〔TSIC78〕を次のように位置づけている。即ち、この概念はデータベースの物理構造の変更やユーザ・アプリケーションの変化（多様性）に関連する異種性問題を解決する手段である。FSSの基本概念は、単一サイトのDBMSに適用しているANSI／SPARCの概念を、異なるサイトにある1つ以上のDBMSを結合することにより生じる分散問題を解決するために、4つのスキーマで構成する構造に拡張したものである。

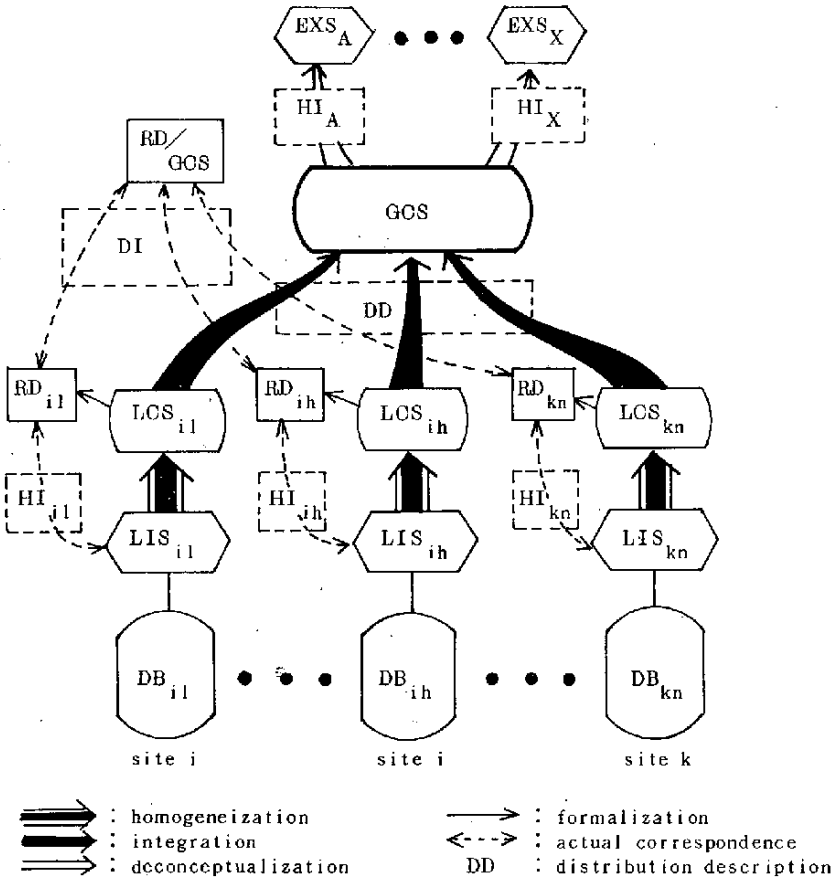


図2－1 The Gross Architecture of Distributed Database System

2.2.1　4層スキーマの記述

ＦＳＳにおける4つのスキーマとは、ＬＩＳ、ＬＣＳ、ＧＣＳ、ＥＸＳである。以下、各スキーマについて、その概要を示す。

第一に、ＬＩＳ（Local Internal Schema：局所内部スキーマ）とは、ローカルサイトのデータベースに格納され、ネットワーク環境下でアクセス可能なデータに関するデータモデルの記述である。ＬＩＳは既存のＤＢＭＳにおけるスキーマもしくはサブスキーマに相当する。さらに、ＬＩＳはＡＮＳＩ／Ｘ３／ＳＰＡＲＣ概念の外部スキーマ（external schema）あるいは概念スキーマ（conceptual schema）に対応する。我々はここでＬＣＳレベルでのＤＢＭＳは、１つのデータモデルとアクセス言語とによって特徴付けられると考える。アクセス言語としては、ホスト言語や問合せ言語が考えられる。

ＬＩＳの異種性問題を解決するには、共通概念モデルによる記述が必要である。我々は共通モデルとしてＥ－Ｒモデル（entity-relationship model）〔ＣＨＥＮ７６〕を採用したが、その理由は次のとうりである。即ち、第１にＥ－Ｒモデルは、Codd が提案しているｎ項関係モデル〔ＣＯＤＤ７０〕にエンタティ（事象）の概念が付加されているからである。従って、Ｅ－Ｒモデルの採用により、リレーショナル形式でのデータベースの記述やアクセスが容易になる。第２の理由は、エンタティとｎ項の関係性の概念が明らかに区別されるからである。このようなことから、我々はＥ－Ｒモデルを用いることにより、種々のＤＢＭＳにより特徴付けられる構文構造（syntactic structure）やデータのセマンティックス（semantics）の記述が可能となる。

第二に、ＬＣＳ（Local Conceptual Schema：局所概念スキーマ）とは、局所データベースのＬＩＳの構文構造をＥ－Ｒモデルを用いてフォーマルにしかも共通に記述するものである。ＬＣＳはＬＩＳと１：１に対応している。ＬＣＳでは、ＬＩＳの構文構造が１つのデータモデルにより特徴付けられる。また完全性条件（integrity constraints）のようにデータモデルでは表現出来ないものは、共通形式で記述され保持される。このように、ＬＣＳレベルでは、異種性問題がＥ－Ｒモデルにより解決されている。各サイトで行なわれるＬＩＳからＬＣＳへの変換プロセスを「同種化」（homogenization）と呼ぶ。この変換プロセスにおいて、ＤＢＭＳの異種性は、異種性情報（ＨＩ：Heterogeneity Information）として除去される。

第三に、ＧＣＳ（Global Conceptual Schema：全体概念スキーマ）とは、ＤＤＢのセマンティックスをＥ－Ｒモデルで記述したものである。即ち、特定のネットワーク共同体にとって意味のあるエンタティや関係性に関する記述である。このように、ＧＣＳはセマンティックスの記述であるので、データベースの全体管理者（ＧＡ：Global Administrator）が記述（定義）しなければならない。ＧＡはネットワーク上に分散しているＬＣＳを局所データベースのデータ管理者と協調してＧＣＳを構成する。これら局所データベースの管理者を局所管理者（ＬＡ：Local Administrator）と呼ぶ。このようにＧＣＳレベルでは、分散問題が解決する。ＧＡがＬＣＳをＧＣＳへ統合するプロセス、即ち、分散問題が分散情報（ＤＩ：Distribution Information）として除去されるプロセスを「統合化」（integration）と呼ぶ。

最後に、EXS（External Schema：外部スキーマ）とは、GCSのサブセットを記述したものであり、これは共同体のあるアプリケーションに特に関係のあるデータの記述である。また、EXSの記述はそのアプリケーションに適合したデータモデルで行なわれる。EXSはセキュリティや使用上の制約に基づいて設定されたGCSのサブセットと言える。従って、以下我々はこれら2つのスキーマ（GCS、EXS）を特に区別しない。

2.2.2. スキーマのインプリメンテーション

我々は、セマンティックスの記述に関してその能力を高めれば高めるほど、結果として記述の複雑さやデータに対するアクセスの困難さを招くと考えている。従って、セマンティックスはE－Rモデルのような概念モデルにより記述する必要がある。また、関係モデル（スキーマのリレーショナル記述（RD）と呼ぶ）はデータの記述やアクセスの手段として有効である。即ち、データベースは、E－Rモデルで記述されたセマンティック記述に基づいて、関係モデル（RD）を用いてアクセスすることができる。

2.3 LISとLCS記述

本節では、LIS_{ij}からLCS_{ij}への変換問題について議論する。ここでLIS_{ij}は、i番目のサイトのj番目のLISを表わしている。前節で述べたようにLIS_{ij}は、既存のデータモデル〔DATE77〕、即ち、関係モデル、網型モデル、階層型モデルを用いて記述される。LCS_{ij}は、LIS_{ij}の構文構造をE－Rモデルで記述したものである。

2.3.1 データモデルの一般構造

既存のDBMSが提供しているデータモデルを用いることにより、データベースの構文構造（syntactic structure）の大部分は表現可能である。あるデータベースのセマンティックに係わる大部分は、完全性条件（integrity constraints）又は規則として記述されなければならない。そして、一般的には文書のような形式で記述され、データモデルとは独立に扱われる。例えば、リレーションがエンタティなのかアトリビュート（属性）なのかはリレーションスキーム（relation scheme）〔BEER78〕だけでは決められず、何らかの付加情報が必要である。本項では幾つかのデータモデルの構文構造、特に構文的関係性を明らかにする。我々が検討するモデルは、網型モデルとしてCODASYL DBTGモデル、階層型モデルとしてIMSモデルである。

一般的にデータモデルとは、構成要素（component）とその要素間の関係性に関する記述とみることができる。我々はデータモデルを要素（elements）の実現値集合（O：occurrence sets）、要素（E）、要素のブロック（B：blocks）、そしてこれらの関係性（$\hat{R}$：relationships）から構成されると考える。

Oはデータベースに格納されているデータの実体である。従って、Oはスキーマのイクステンション（extension）である。また、これはE－Rモデルの値集合（value set）に対応している。

Eはデータベースにおいて名前付けられた最小単位であり、DBTGモデルのデータ項目名、関係モデルの属性名、IMSモデルのフィールド名に対応する。Bは要素の集合であり、レコード型、リレーション名、セグメント型にそれぞれ関連付けできる。$\hat{R}$は3要素、即ち、O、E、B間の関連を示す。個々のデータモデルは関係性により特徴付けられる。例えば、セット型(set-type)は、DBTGモデルにおける最も特徴的な関係性である。関係性には次に示すように4種類ある。即ち、2つのブロック間の関係($B\hat{R}B'$)、ブロックと要素間の関係($B\hat{R}E$)、要素とブロック間の関係($E\hat{R}B$)、要素間の関係($E\hat{R}E'$)の4種類である。表2-1にデータモデルと関係性との関係を示す。

表2－1　データモデルにおける関係性

relation-ships / models	$B\hat{R}B'$	$B\hat{R}E$	$E\hat{R}B$	$E\hat{R}E'$
relational model	$\hat{R}$: has a common domain with	$\hat{R}$: is composed of	$\hat{R}$: is {key, composite-key, non-key} of	$\hat{R}$: multi- or functionally determines
DBTG model	$\hat{R}$: is an owner of (set-type or link-record-type)	$\hat{R}$: is composed of	$\hat{R}$: is {key, composite-key, non-key} of	
IMS model	$\hat{R}$: is a parent of (hierarchical path)	$\hat{R}$: is composed of	$\hat{R}$: is {key, composite-key, non-key} of	

2.3.2　LCS記述

前項では、各データモデルの一般構造について明らかにした。この検討結果から、次の事が明白である。即ち、既存の3つのデータモデルの個々の構成要素は、E－Rモデルの要素と1：1対応が可能である（表2－2参照）。例えば、関係モデルの領域(domain)はE－Rモデルの値集合に、DBTGモデルのデータ項目名は属性(attribute)に関係付けられる。

1) エンタティセット（事象集合）の定義

事象集合（ESs）は次のように定義される。

es-name(at-name[/v-name]=format[/role]
{,at-name[/v-name]=format[/role]}).

ここで、es-name及びat-nameは、それぞれES名（事象集合名）、属性名を表わしている。v-nameは、値集合名を表わし、at-nameとv-nameが同一の場合にはv-nameを定義から除くことが可能である。roleは、事象集合における属性の役割を定めるもので、その属性がキー(key)、統合キー(composite-key)、非キー(non-key)の何れであるかを示す。

非キーである属性は定義する必要がない。またformatでは、その属性のタイプ（文字型、整数型）やバイト長が定義される。ESに関するこの定義をESスキーム（ESS：ES scheme）と呼び、BREとERBの2つの関係性はこのESスキームで記述される。

EMP（emp#：EMPLOYEE(I5)，name：NAME(C20)，
sal：SALARY(I10)）

このリレーションスキームをESスキームで記述すると次のようになる。

EMP（emp#／EMPLOYEE＝I5／key，
name／NAME＝C20，sal／SALARY＝I10）

2) リレーションシップ・セット（関係性集合）の定義

関係性集合（RS）は、ES間の構文的関連を示したものである。具体的には、DBTGのセット型、IMSの階層パスに相当し、関係モデルでは領域を共有した相互リレーション関係性（inter-relation relationship）がそれである（表2－2参照）。E－Rモデルにおいて、RSはn項の関係性にあるのが一般的である。しかしながら、既存のデータモデルにおいては、セット型や階層パスのような2つのES間に関係性が存在するので、2項関係で関係性を表現するだけで十分である。従って、2つのES間（ES_1, ES_2）の関係性RSを次のように定義することができる。

rs-name（es_1-name／(list-of-kat)〔／st_1〕，
es_2-name／(list-of-kat)〔／st_2〕)〔(rs-at)〕.

list-of-katは事象集合ES_i（i=1, 2）のキー属性のリストである。ES_iにキー属性が無い場合は、ES_iの全属性が、ES_iのキーとして定義される。RSが属性を持っている場合には、rs-atに定義される。また、幾つかのリレーションで共有されている領域名は、関係モデルにおけるrs-atで定義される。

rs-at::＝at-name〔／domain〕＝format
{，at-name〔／domain〕＝format}|shared-domain-name.

st_iは、データモデル固有の構造を表現したものである。DBTGモデルにおけるst_iとしては、set_i-name／direction が考えられる。即ち、set_i-nameはセット型の名前に相当し、RSとES_iに関連がある場合で、RSがリンクレコード型の場合には次のようになる。ES_iがセット$_i$の子レコード型の場合は方向がinでその他の場合は方向がoutになる。我々はこの定義をRSスキーム（RSS：RS scheme）と呼んでいる。またRSの関係性には方向性（$ES_1 \rightarrow ES_2$）が伴っていることも特徴の1つである。

一方、関係モデルにおいてはこのような関係性が明確には存在しない。2つのリレーション間の関係性の存在は、これらのリレーションが同一領域に属する属性を持っていることを意味している。その為には、ESスキームで記述した属性と値集合（領域）を用いてRSスキームを構築しなければならない。

表２－２　E－Rモデルと３つのデータモデルとの関係

general structure	E-R model	relational model	DBTG model	IMS model
O	value set	domain	occurrences of data-item	occurrences of field
E	attribute	attribute	data-item name	data-item name
B	entity-set	relation-name	record-type	segment-name
B R B′	relation-sbip-set	shared domains	set-type, link-record-type	hierarchical path

2.3.3　LCSのリレーショナル記述（RD）

我々のシステムにおいては、ES及びRSはリレーショナル記述（RD_{ij}：Relational Description of LCS_{ij}）と呼ばれるリレーショナル形式で表現される。LCSレベルでの個々のデータベースは、LCSのE－Rモデル記述に基づいたRDによりアクセスしたり記述される。これは、関係モデルがデータに対する記述及びアクセスに関して簡潔性を提供し得るからである。

RDは２種類のリレーション、即ち、事象集合リレーション（ESR）と関係性集合リレーション（RSR）を取り扱う。RDの形式は両者共通である。

```
rel-name(at-name[/domain[/role]]
        {,at-name[/domain[/role]]}).
```

RSRの属性は、RS自身の属性とRSを通して関連している２つのESに属しているキー属性から構成される。

2.3.4　変換例

本項では３つの異なるデータモデル（関係モデル、DBTGモデル、IMSモデル）で記述されたスキーマからE－Rモデルのスキーマ（LCS）への変換例を示す。

まず、ここで表２－３に示す３つのデータベースがコンピュータ・ネットワーク上に分散しているとする。

プロジェクト管理データベースPMDBは各研究所に存在し、その研究所でのプロジェクトに関する情報（プロジェクト・メンバー、報告書、プロジェクト・マネージャ等）を格納している。

MEMBER　(mem#, name, address, phone#, position, degree, deg-year, alma-mater, birthday, birthplace)

REPORT　(rep#, no-of-authors, title, abstract, content, source, vol, no, month, year, page, cost)

AUTHOR　(rep#, author-no, author)

PROJ　(proj#, pname, manager)

PROJ-MEM　(proj#, mem#)

図2－2　PMDBのスキーマ

表2－3　データモデルとデータベース名

データモデル	データベース名
関係モデル	PMDB(project management databases)
DBTGモデル	PRDB(project database)
IMSモデル	BLDB(bibliography database)

プロジェクトデータベースPRDBはクリアリングセンターに位置し、各研究所で進行中（含、終了）の研究プロジェクトに関する情報を提供する。図2－3は、2つのレコード型（KEYWORDSとPROJECT）が網構造（m:n）の関係にあることを示している。SPONSORとPROJECTの関係も同様にm:n の網構造となっている。また、レコード型PROJECTは、自分自身で網型のサイクル構造を表わしている。即ち、PROJ－KEYW－LINK、PROJ－SPON－LINK、PROJ－LINKの3つのレコード型はいわゆるリンクレコード型である。

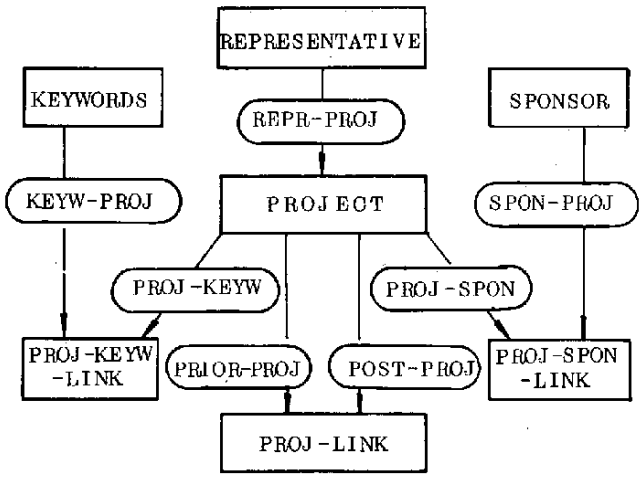


図2－3　PRDBのデータ構造図式(data structure diagram)

文献データベースＢＬＤＢは、ドキュメントセンターにあり各プロジェクトが発表した報告書に関する情報、即ち、文献の書誌的事項について管理している〔図２－４参照〕。

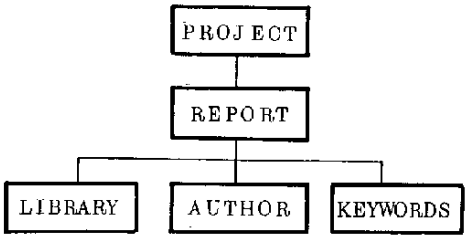


図２－４　ＢＬＤＢのデータ構造図式

ＬＣＳは表２－２を用いて３つのＬＩＳから変換される。図２－５及び２－６は、それぞれＰＭＤＢ、ＰＲＤＢのＬＣＳを表わし、図２－７は、これらＬＣＳのＲＤを示している。

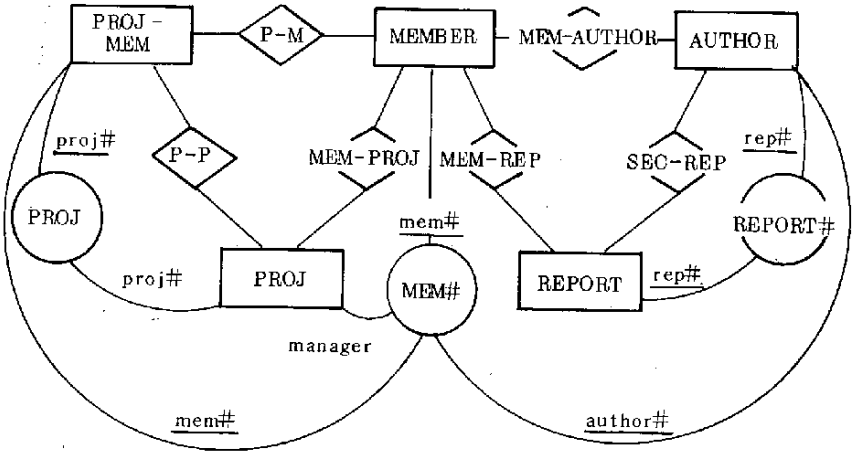


図２－５　ＰＭＤＢのＬＣＳのＥ－Ｒ図式

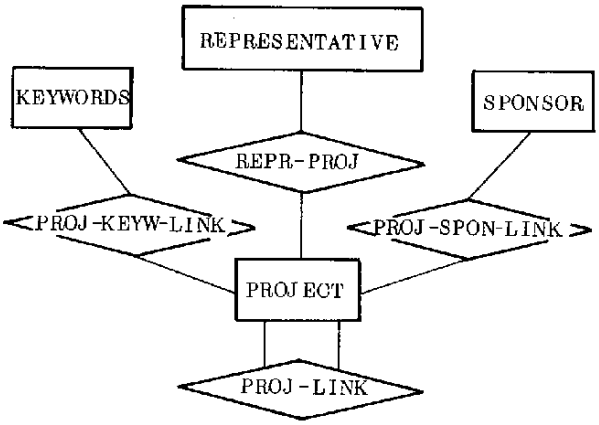


図２－６　ＰＲＤＢのＬＣＳのＥ－Ｒ図式

1) PMDB (relational database)

ESR schemes

MEMBER (mem#, name, address, phone#, position, degree, deg-year, alma-mater, birthday, birthplace)

PROJ—MEM (proj#, mem#)

PROJ (proj#, pname, manager/MEM#)

REPORT (rep#, no-of-authors, title, abstract, content, source, vol, no, month, year, page, cost)

AUTHOR (rep#/REP#, author-no, author/MEM#)

2) PRDB (DBTG database)

ESR schemes

REPRESENTATIVE (repr-no, reprname, repraddr)

PROJECT (proj-no, projname, projeyar, projsyar, projstat)

SPONSOR (sponname, sponorg, sponaddr, sponrepr)

KEYWORDS (keywords)

RSR schemes

PROJ—KEYW—LINK (keywords, proj-no)

PROJ-LINK (prior-no /PROJ—NO, post-no/PROJ—NO)

PROJ—SPON—LINK (sponname, proj-no, budget)

REPR—PROJ (repr-no, proj-no)

図2－7 PMDBとPRDBのRD

2.4 アクセス言語

2.2で述べたように、簡潔性のために、実際にはGCS及びLCSの両レベルで、データベースは関係モデルを用いてアクセスされる。我々は、関係計算言語(relational calculus language)が第1階述語計算理論(first predicate calculus)に基づいているために高度の検索記述機能を提供しうると考えている。関係代数言語(relational algebra language)は、リレーションに対する基本的な演算能力を提供しているが、関係計算言語より記述能力の点で劣っている。これは、関係計算言語に比べてより手続的であり、しかもインプリメンテーションの手段として提供されているからである。

以上の点から、我々はアクセス言語として関係計算言語を採用した。また、特に我々が採用した言語は限定作用素(∀、∃)を何ら必要としないQUEL〔YOUS77〕である。

QUELは、RANGE文、RETRIEVE文、更新文(APPEND, DELETE, RE-

PLACE) から成り立っている。

問合せ内におけるリレーションは、組変数(tuple variable)によって代表される。リレーションの属性は、属性名に、この組変数を修飾して使用される。組変数は、RANGE文によって定義される。1つのリレーションが複数の組変数を持ちうるが、この逆は成り立たない。RANGE文は次の形式を持つ。

RANGE文 ::= range of(〈var-list〉)(〈rel-list〉);

〈var-list〉と〈rel-list〉は、各々組変数とリレーション名との並びである。

RETRIEVE文は、いくつかのリレーションから必要なデータをみつけるものであり、次の形式をもっている。

RETRIEVE文 ::= retrieve into 〈rel-name〉(〈target-list〉)
where 〈qualification〉;

〈rel-name〉は、問合せの結果生成されるリレーション名である。これを結果リレーション(result-relation)と呼ぶ。〈target-list〉は、目標リストである。即ち、結果リレーションの属性名及びその定義を行なう。

〈target-list〉::=〈t-clause〉{,〈t-clause〉}

〈t-clause〉 ::=〈attribute〉=〈a-exp〉|〈v-attribute〉

〈attribute〉は、結果リレーションの属性で、結果属性(result-attribute) と呼ばれる。〈a-exp〉は、定数と属性の算術式である。

〈a-exp〉::=(〈a-exp〉)|〈a-exp〉〈aop〉〈a-exp〉|
−〈a-exp〉|〈a-function〉|
〈g-function〉|〈constant〉|〈v-attribute〉

〈aop〉 ::=+|−|＊|／|＊＊

〈a-function〉は、ABS、LOGといった算術関数であり、〈g-function〉は、MIN、MAXといった集合(aggregate)関数である。〈v-attribute〉は、RANGE文で宣言された組変数によって修飾された属性である。

〈v-attribute〉::=〈variable〉.〈attribute〉

〈qualification〉は、問合せの条件を表わしている。

〈qualification〉::=〈clause〉|(〈qualification〉)|
〈qualification〉∧〈qualification〉|
〈qualification〉∨〈qualification〉|
～〈qualification〉

〈clause〉 ::=〈a-exp〉〈cop〉〈a-exp〉

〈cop〉 ::=＜|≦|＝|≧|＞|≠

〈clause〉は、比較句(comparison clause)を表わす。

QUELの問合せは、次のことを意味している。即ち、RANGE文で定義されたリレーション

の直積をつくり、条件部の論理式によって制限（restriction）し、目標リストで必要な属性だけに射影（projection）する。このようにして生成された結果リレーションは、通常のリレーションとしてQUELによって問合せを行なえる。

更新に関しては、〔ＹＯＵＳ７７〕を参照のこと。

2.5 異種性問題（Heterogeneity problem）

異種ＤＢＭＳを統合するためには、既存の各種ＤＢＭＳの異種性（heterogeneity）を、まず克服する必要がある。2.1で定義した様に、異種性とは、ＤＢＭＳの提供するデータモデルと言語（ＤＤＬとＤＭＬ）の各々の相違である。データモデルの相違は、ＬＩＳをＥ－Ｒモデルによって共通記述することによって（ＬＣＳ）解決されている〔2.3〕。本節では、共通のデータモデルによって記述されたＬＣＳに基づいて出された問合せを、各ＤＢＭＳで実行可能なＬＩＳに基づいた問合せ又は親言語に変換し、実行させる問題を論じる。この様な問合せ言語又は親言語を、ＤＢＭＳのＤＭＬと呼ぶことにする。第１に、ＱＵＥＬ言語によって記述された非手続的問合せを、一連のＤＢＭＳ　ＤＭＬに変換する問題を、2.5.1で論じる。次に、2.5.2では、この変換のために必要となる異種性情報（ＨＩ）の内容とその管理問題を論じる。最後に、ＤＢＭＳの実際の起動、エラーの回復、ネットワークの支援機能等の制御問題を考える。

2.5.1 問合せ変換（ＱＴ）

ＧＣＳレベルにおけるリレーション（即ちＥＳＲとＲＳＲ）は、種々のサイトのＬＣＳのリレーションより構成されているviewとみなせる。従って、ＧＣＳレベルの全体問合せ（global query）は、１つのサイト内のリレーションのみを参照している局所問合せ（local query）に分割される必要がある。この全体問合せから局所問合せへの分割を、全体―局所問合せ分割（ＧＬＱＤ）と呼ぶ。全体と局所の両方の問合せは、ＱＵＥＬ言語〔ＹＯＵＳ７７〕〔2.4〕によって記述される。

ＧＬＱＤによって分割された局所問合せは、そのサイトのＤＢＭＳ（局所ＤＢＭＳと呼ぶ）によって実行可能な一連のＤＭＬに変換される。この変換を問合せ変換（ＱＴ）と呼ぶ。ＱＴは、さらに２つのプロセスから成り立っている。第１のプロセスは、局所―局所問合せ分割（ＬＬＱＤ）と呼ばれるものである。局所問合せは、一般に複数の局所リレーションを参照している（この様な問合せを、多変数問合せ（multi-variable query）と呼ぶ）。複数リレーションを参照している問合せは、一般的に、これらのリレーションの直積（Cartesian product）を作り、これに射影（projection）演算と制限（restriction)演算とを行なうことによって処理されることができる。しかし、リレーションの直積は、一般に、これらのリレーションのcardinalityの積に等しい組数をもつ。これは、リレーション数の増加によって、組合せ的に大きくなってしまう。よって、多変数問合せを、より少ないリレーションを参照している問合せに分割することは、組合せ的増大を防ぐために有効な方法と考えられる。ＬＬＱＤにおいて、多変数局所問合せは、一連のより

少ないリレーションを参照している問合せに分割される。

この様にして分割された問合せは、QUEL言語で記述されているために、局所DBMSでは一般に実行できない。よって、我々は、分割された問合せを、そのサイトで実行可能な一連のDMLに変換せねばならない。このプロセスを、局所—局所問合せ変換（LLQT）と呼ぶ。

これらの2つのプロセスにおける問題点は、分割された問合せと、変換されたDMLの最適な実行順序を決定することである。このことは、全体問合せから、局所問合せへの分割（即ちGLQD）においても同様のことをいうことができる。QTとGLQDとの差は、最適化の目的関数の相違である。GLQDにおいては、各ローカルサイトでの問合せの処理の並行度（parallelism）を最大化することと、各サイト間でのデータの転送の最小化をはかることが重要である〔HEVN78〕。これに対して、LLQTとLLQDにおいては、実際のデータベースの物理的なアクセスの効率化をはかることが重要となる。即ち、問合せの処理中に生成される中間結果を最少とするとともに、有効なアクセスパス(e.g. index,clustering index etc.)を用いて、アクセスの高速化をはかることである。

アクセス言語は、検索と更新（消去、置換、追加）機能を備えているが、問合せの処理としては、データベースに対する検索(retrieve)問合せについてのみ論じる。更新(update)は、更新されるためのデータを見つけるための検索演算と、これによって見つけられたデータへの書込み演算との対とみなすことができる。よって、問合せ処理において、検索演算を議論することは本質的である。

本節では、問合せ変換問題のうちのLLQDとLLQTとを議論する。特に、QUEL問合せを、一連のDBTGのDMLに変換することを議論する。

A. 局所--局所問合せ分割（LLQD）

LLQDは、局所（多変数）問合せを、変換目標のDBMSの構文構造を考慮しながら、一連のn変数（$1 \leq n \leq 3$）の問合せに分割するプロセスである。ユーザの問合せの条件部は、論理式(formula)であり、句の接合（∧）、離接（∨）、否定（～）より成り立っている。第1に、このような論理式を、離接正規化(disjunctive normalization)することを論じる。次に、離接正規化された問合せから、接合問合せ(conjunctive query)の集合に分割することを論じる。これを水平分割と呼ぶ。接合問合せは、問合せ図(query graph)を用いて、問合せの参照するリレーションに対する条件(qualification)づけを表わせる。次に、この問合せ図を定義し、問合せの図表現を示す。最後に、DBTGモデルを変換目標とした場合の、このモデルの構文構造を、問合せ分割に導入することを論じる。

1) 離接正規形(disjunctive normal form)

ユーザの問合せの条件部は、～、∧、∨と句からなる論理式である。問合せ処理上、論理式を離接(disjunctive)又は正接(conjunctive)正規化することは望ましい。正接（∧）は、関係代数の結合に、離接（∨）は論理和に対応させられる。我々は、結合演算の後に、これらの結果の和をとるように問合せを処理するために、問合せを離接正規化する。

ここで、我々は、全ての問合せの条件部(qualification part)は、離接正規形(dis-

junctive normal form）であると仮定する。即ち、条件部をQLとすると、$QL = QL_1 \vee QL_2 \vee \cdots\cdots \vee QL_n$。ここで各々の$QL_i$は、離接演算子（∨）を含まない論理式である。即ち、$QL_i = c_{i1} \wedge c_{i2} \wedge \cdots \wedge c_{in_i}$。$c_{ij}$は、〈a-exp〉〈cop〉〈a-exp〉の形式をもつ比較句であり、〈cop〉$\in \{<, \leq, =, \geq, >, \neq\}$、〈a-exp〉は、属性と定数とからなる算術式である〔2.4を参照〕。もちろん〈a-exp〉には、MAX、MIN、SUMといったライブラリ関数も含まれる。n_iは、QL_i内の比較句の数である。

ここで比較句〈a-exp〉〈cop〉〈constant〉は、制限句（restrictive clause）と呼ばれる。例えば、x.a≧5は、制限句である。制限句の特徴は、他のリレーションとは独立に、この句によって参照されるリレーションに制限を行なうことができることである。特に〈cop〉＝〝＝〟の場合、等価制限句（equi-restrictive clause）と呼ぶ。例えば、x.b＝〝TOM〟は、等価制限句である。

これに対して、比較演算子の第1及び第2被演算子の算術式内に、ともに属性が含まれている句を結合句（join clause）と呼ぶ。例えば、x.a≧y.bは、両辺に各々x.aとy.bという属性を持っているために結合句である。結合句の特徴は、両辺の各々が参照する2つのリレーションの関連（即ち結合）を表わしていることである。特に、〈cop〉＝〝＝〟の場合、この句は等価結合句（equi-join clause）と呼ばれる。例えば、x・a＝5＊y・bは等価結合句である。

2) 水平分割（horizontal decomposition）

離接正規化された問合せは、この問合せの条件部の各々の接合子（conjunct）（即ちQL_i, for $i = 1, \cdots\cdots, n$）を条件部として持つ問合せに分割できる。この分割を、水平問合せ分割（horizontal query decomposition：HQD）と呼ぶ。結果Rが、分割された問合せの結果の和（union）となるという意味で〝水平〟である。

即ち、問合せ

$$\left.\begin{array}{l} Q: \underline{\text{range of}}\ (x_1, \cdots\cdots, x_n)(X_1, \cdots\cdots, X_n); \\ \quad \underline{\text{retrieve into}}\ \ R\ (TL) \\ \quad \underline{\text{where}}\ \ \ QL_1 \wedge QL_2 \wedge \cdots\cdots \wedge QL_n; \end{array}\right\} \quad (1)$$

は、

$$\left\{\begin{array}{l} Q_i: \underline{\text{range of}}\ (x_1, \cdots\cdots, x_n)(X_1, \cdots\cdots, X_n); \\ \quad \underline{\text{retrieve into}}\ \ R_i\ (TL) \\ \quad \underline{\text{where}}\ \ \ QL_i\ ; \end{array}\right| \ i = 1, \cdots\cdots, n \Bigg\} \quad (2)$$

に水平分割される。ここでx_iは、リレーションX_iの組変数（tuple-variable）である。TLは目標リスト（target-list）である。

問合せQの結果として生ずるリレーションRと、Q_iの結果R_iは同一の属性構成をしている。何故なら、両者の目標リストは同一である。よってRは、R_i（$i = 1, \cdots\cdots, n$）の和集合（union）である。即ち、$R = R_1 \cup R_2 \cup \cdots\cdots \cup R_n$である。水平分割された問合せ$Q_i$の条件部は次の形式である。即ち、接合のみからなり、$QL_i = c_{i1} \wedge c_{i2} \wedge \cdots\cdots \wedge c_{in_i}$である。

先に述べたように、c_{ij}($j=1$, ……, n_j)は比較句である。このように、条件部が比較句の接合(conjunction)であることから、この問合せを接合問合せ(conjunctive query)と呼ぶ。

比較句の離接(disjunction)を考えてみよう。これは、$c_{i1} \vee c_{i2} \vee \cdots\cdots \vee c_{in_i}$と表わせる。cij(for $j=1$, ……, n_i)は比較句である。比較句には、制限句と結合句の2種類がある。さらに、これらの比較句が参照する組変数を考慮すると、制限句は、同一の組変数を参照する句の集合にグループ化できる。1つのグループ内の制限句は同一の組変数のみを参照している。例えば、次の論理式を考えてみよう。即ち、x.a=5∨y.b≧6∨x.a=8∨z.d="TOM"∨x.b=9. この論理式は、x.a=5∨x.a=8∨x.b=9, y.b≧6とz.d="TOM" の3つにグループ化できる。問合せ処理において、このような制限句のグループを離接正規形に変形することによって各句を異なった接合問合せに置くよりも、ある接合問合せの条件部の1つの句としてまとめた方が、アクセスの効率上望ましい。もし水平分割すれば、これらの同一組変数を参照する比較句を持つ問合せは、各々同一のリレーションを冗長にアクセスすることになる。

一方、結合句も、両方が参照する組変数の対によってグループ化できる。これらのグループ内の比較句は、どれもある2つの組変数のみを参照している。例えば、x.a=y.b∨x.b=z.e∨y.c=x.b∨y.f=z.f∨x.a=z.aを考えよう。この論理式は、x.a=y.b∨y.c=x.b, x.b=z.e∨x.a=z.a, とy.f=z.f の3つのグループに分けることができる。これらの各々のグループも、離接正規化することによって、各比較句に対応した問合せを生成するよりも、1つの問合せ内に1つのグループを持たせた方が処理効率上望ましい。理由は、制限句と同じく冗長なアクセスを防ぐためである。

以上の議論より、上述した様な比較句のグループを句としてもつ接合問合せを、modified接合問合せ(conjunctive query)と呼ぶ。modified接合問合せを生成するためには、先に述べた比較句の離接のグループを、論理式からくくり出さねばならない。このためには、論理式の記号処理が必要となる。

以降の議論では、接合問合せ(conjunctive query)のみを考えることにする。この問合せの比較句を、先に述べた同一グループの離接に置き換えれば、この議論を、modified問合せに拡張することは容易である。以降問合せは、接合問合せを意味するものとする。よって、問合せQは次のように表わせる。

$$\left.\begin{array}{l} Q : \underline{\text{range of}} \quad (x_1, \cdots\cdots, x_n)(X_1, \cdots\cdots, X_n); \\ \quad\ \underline{\text{retrieve into}} \quad R\,(TL) \\ \quad\ \underline{\text{where}} \quad\quad c_1 \wedge c_2 \wedge \cdots\cdots \wedge c_m\ ; \end{array}\right\} \quad (3)$$

ここで、Rは問合せQの結果生成されるリレーション名であり、c_i(for $i=1$, ……, m)は比較句(comparison clause)である。nは問合せ内の句の数である。問合せQはn コの組変数を参照するn変数問合せ(n-variable query)である。この問合せを、より少ない変数を参照する一連の問合せに分割することを、我々は、先に述べた水平分割に対して垂直問

合せ分割(vertical query decomposition：VQD)と呼ぶ。Qの結果Rは、分割された問合せによって生成される結果の結合(join)であるという意味で、〝垂直〟という言葉を用いた。

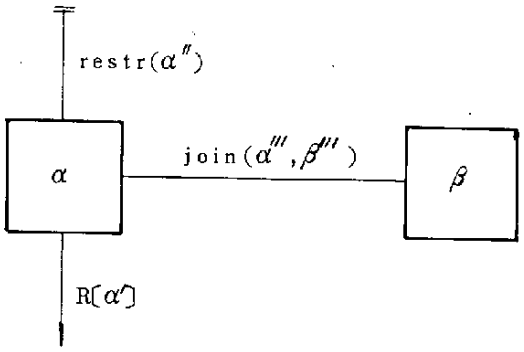
以降、問合せの垂直分割を議論する。以降本文中に表われる分割は、垂直分割を意味するものとする。

3) 問合せ図(query graph)

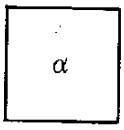
問合せ分割を論じるうえで、問合せを図形表示すると便利である。我々は、このために問合せ図〔WONG77〕を導入する。ここで、問合せ図は、前述した接合問合せ(又は、modified接合問合せ)のみを表わせることを注意しておく。

問合せ図は、ノードとリンクからなっている。ノードは、箱(i.e.□)で表わされ、1つの組変数を表わしている〔図2－8〕。リンクには、結果リンク(result-link),制限リンク(restriction-link)と、結合リンク(join-link)との3種類がある。結果リンクと制限リンクは、1つのノードとのみ結合している。結合リンクは、2つのノード間に存在している。結果リンク(——→)は、このリンクの出ているノードを表わすリレーションの属性のなかで、問合せの目標リスト内にも表われるものがあれば、それを表わしている。制限リンク(——|)は、このリンクの出ているノードに関連した条件部内の制限句(restrictive clause)の論理式を表わしている。結果リンクは、このリンクによって関連づけられる2つのノード間の結合句(join-clause)の論理式を表わしている。

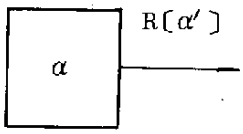
これらのリンクが、ある問合せ図に存在することは、次のことを意味している；これらのリンクが表わす比較句の論理式が、同時に全て満足されたものが結果リレーションとなる。即ち、これらの論理式の接合を図は表わしている。例えば、x.a＝y.b ∨ y.b＝z.cという条件を持つ問合せを、問合せ図は表わせない。



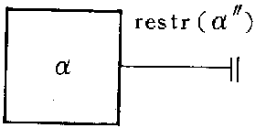
ここで



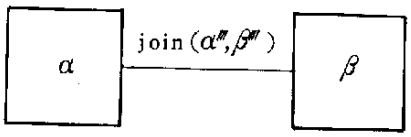
：　組変数αを表わすノード



：　結果リンク（ result-link ）
ここで、Rは、結果リレーション名



：　制限リンク（ restriction-link ）
restr（α″）は、属性名集合α″を参照する制限句の接合（ conjunction ）を表わしている。＊）



：　結合リンク（ join-link ）
join（α‴,β‴）は、属性名集合α‴とβ‴とを各々両辺で参照する結合句の接合を表わしている。＊）

α′,α″,α‴　：　組変数αの range リレーションの属性名の部分集合

＊）　modified 接合問合せ（ modified conjunctive query ）では、離接も可能である。

図２－８　問合せ図（Query Graph）

4) 問合せの例

例として、リレーションスキーム(relation scheme)〔BEER78〕は、図2－7のプロジェクトデータベース（PRDB）を考える。問合せとしては、次のものを考える。

〝１９７５年以降データベースを研究しているプロジェクトの名前とその代表者をもとめよ〟

対応するQUEL問合わせ（Q）は、次のように記述できる。

```
Q:  range of (j, r, pkl, rp)     (PROJECT, REPRESENTATIVE,
                                  PROJ-KEYW-LINK, REPR-PROJ);
    retrieve into RESULT (j.projname, r.reprname)
    where j.projsyar≧'1975' and j.projstat = "active"
    and j.proj-no = rp.proj-no and rp.repr-no = r.repr-no
    and j.proj-no = pkl.proj-no and pkl.keywords = "databases";
```

(4)

問合わせ分割は、この例に基づいて以降論ずる。

5) 垂直分割(vertical decomposition)

問合せの分割は、次のプロセスより成り立つ。

(i) 条件部の論理式より、推移関係性(transitivity)を用いて導出される制限比較句を、条件部に付加する。

(ii) 問合せに対応する問合せ図(query graph)を生成する。

(iii) 問合せ図における切断ノード(cut-node or cut-vertex)に基づいて、図を部分図(subgraph)に分割する。

(iv) 部分図の実行順序を定める。

(v) 部分図から問合せを生成する。

第１に、問合せの条件部を解析して、論理式の推移関係(transitivity)から導出できる制限比較句を付加する。例えば、論理式x.a＝5 ∧ x.a＝y.aより、y.a＝5 を導き出せる。この場合、y.a＝5は制限句であるために、これを条件部に接合によって付加することは有効である。特に、属性y.aがリレーションのキーであれば、さらに有効なものとなる。推移関係から、結合句の導出は行なわない。結合句は、リレーション間のアクセスパスを示しているために、推移性に基づいた結合句の導出は、意味のない又は関係のないアクセスパスをつくってしまう恐れがあるからである。この問合せQ内には、対応する推移性は存在しない。よって、制限句の付加は行なわれない。

次に、対応する問合せ図を生成する。これは、図２－９に表わされる。

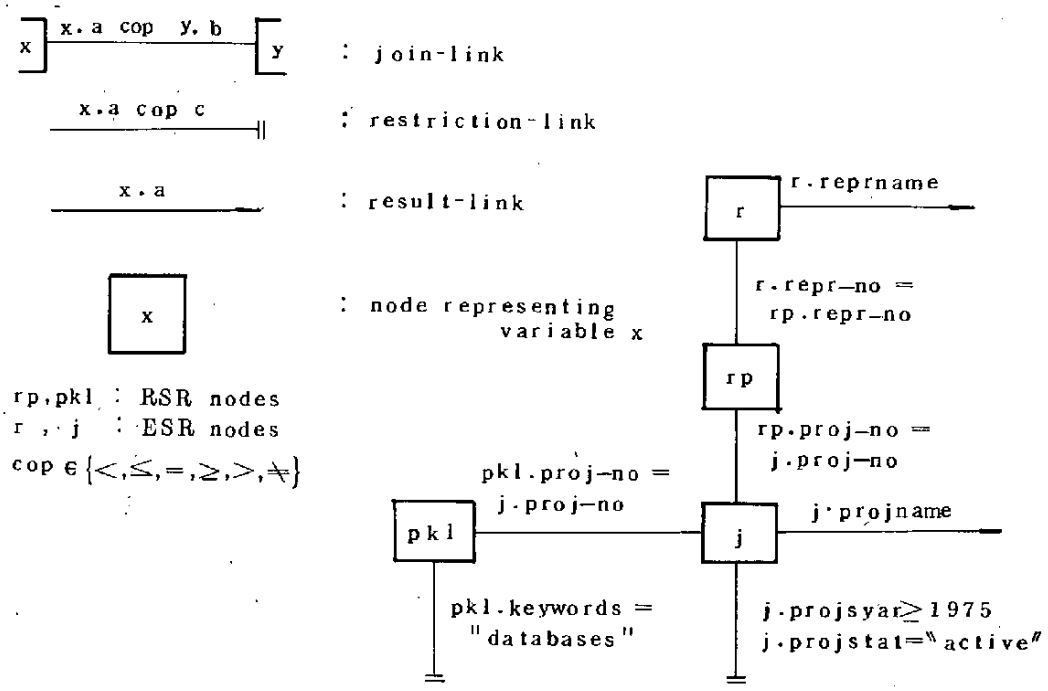
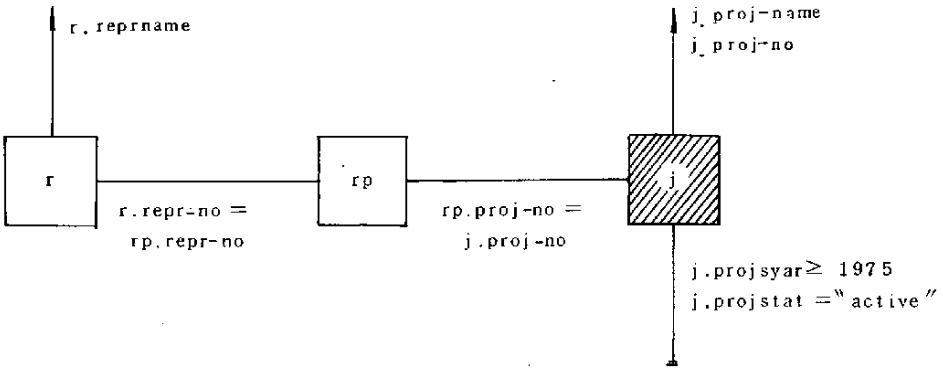


図２－９　(4)の問合せに対応する問合せ図

次に、この問合せ図を用いて、問合せの分割を行なう。このために、まず問合せ図の切断ノード(cut-vertex or cut-node)を見つける。グラフ理論によると〔DEO７４〕、切断ノードとは、そのノードを図から除去してしまうと残された図が連結(connect)されなくなってしまうノードと定義されている。先に述べたように〔2.2.2〕、我々はＥＳＲ(entity-set relation)とＲＳＲ(relationship-set relation)との２種類のリレーションをもっている。特にＬＣＳレベルにおけるＲＳＲは、事象集合(entity-set)（即ち、ＤＢＴＧモデルのレコード型、ＩＭＳモデルのセグメント型）間の構文的関係性を示している。構文的関係性とは、ＤＢＴＧモデルにおけるセット型、リンクレコード型、ＩＭＳモデルの階層パスを意味している。よってアクセス効率上、局所(local)ＤＢＭＳの構文的構造を考慮するためには、ＲＳＲノード（ＲＳＲを表わすノード）を切断ノードとして選択してはならない。ここで、我々は問合せ図における切断ノードを次のように再定義する。即ち、切断ノードとは、ＲＳＲノード以外（すなわちＥＳＲノード）のグラフ理論によるところの切断ノードである。図２－９の例では、ｒｐノードは、ＲＳＲノードのために切断ノードはｊノードだけである。よって、ｊノードを切断ノードとしてつくられる部分図は、ｐｋｌ－ｊ、ｊ－ｒｐ－ｒの２つである〔図２－１０〕。このような部分図は、各々分割された問合せに対応している。

この部分図の順序づけは、目標ＤＢＭＳの構文構造を考慮して決定されねばならない。

1) r－rp－j 部分図



2) pkl－j 部分図

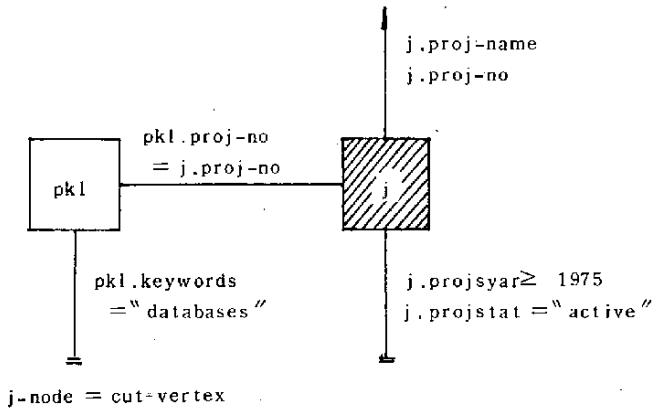
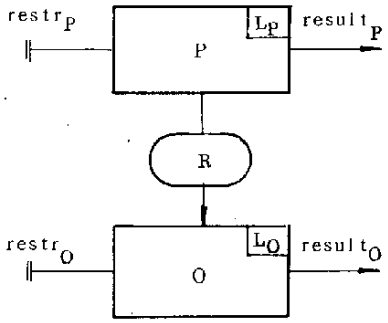


図２－１０　図２－９の部分図

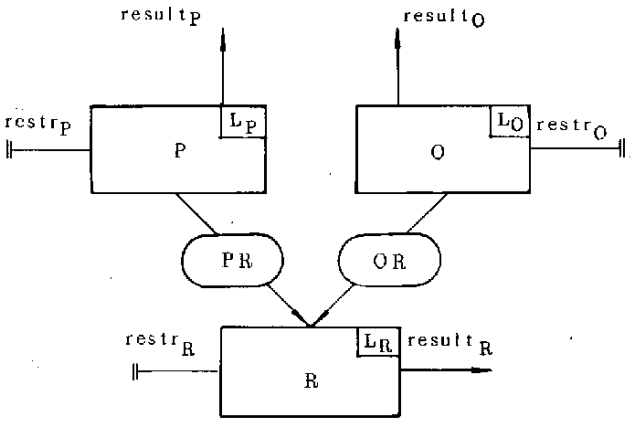
6) ＤＢＴＧ図式（DBTG diagram）

ここで、ＤＢＴＧモデルの構文構造と問合せ図の関係を考えてみる。切断ノードに基づいて分割された部分図は、目標とするモデルのアクセスの基本単位に相当している。ＤＢＴＧモデルでは、レコード型間の階層、ネットワーク、階層サイクル、ネットワークサイクルの４つの関係に対応している〔図２－１１〕。このアクセス単位に対応する問合せ図の部分図には、５つのタイプ（ＥＳ－ＲＳ－ＥＳ，ループＥＳ－ＲＳ－ＥＳ，ＥＳ－ＲＳ，ＲＳ，ＥＳブロック）がある〔図２－１２〕。これらのブロックと、ＤＢＴＧ図式（DBTG diagram）との関連は図２－１２に示されている。図２－１２における点線は、レコード型、又はセット型が問合せ内に明確に表われていないことを表わす。よって、この隠されたデータ構造は、このデータベースに関する異種性情報（ＨＩ）から導入される必要がある。

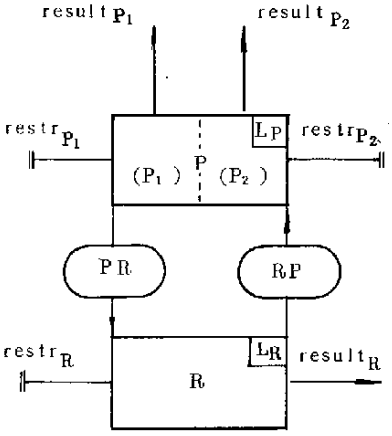
例えば、部分図ｐｋｌ－ｊ〔図２－１０〕を考えてみよう。ｐｋｌノードはＲＳＲノードであり、ＤＢＴＧスキーマ上のＰＲＯＪ－ＫＥＹＷ－ＬＩＮＫレコード型を表わしている。このレコード型は、ＰＲＯＪとＫＥＹＷとの間のｎ：ｍ（ネットワーク）関係性を表わしており、いわゆるリンクレコード型である。この部分図は、ＥＳ－ＲＳブロックである。この部分図に対応するＤＢＴＧ図式を図２－１３に示す。



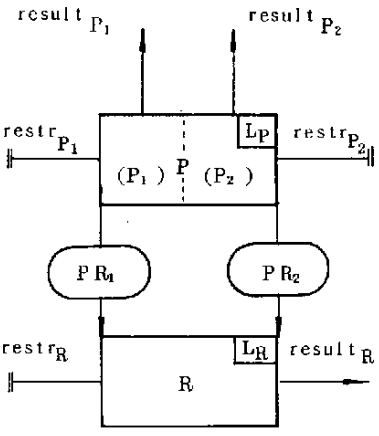
i) 階・層 関 係



ii) ネットワーク関係



(iii) 階層サイクル関係



(iv) ネットワークサイクル関係

：レコード型

：セット型

$restr_{\alpha}$：レコード型αに関する制限句
$result_{\alpha}$：レコード型αに関する結果項目のリスト

L_{α}：レコード型Pのロケーションモード

図2－11　DBTG図式

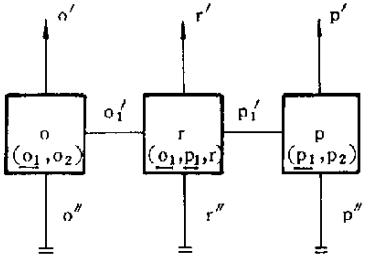
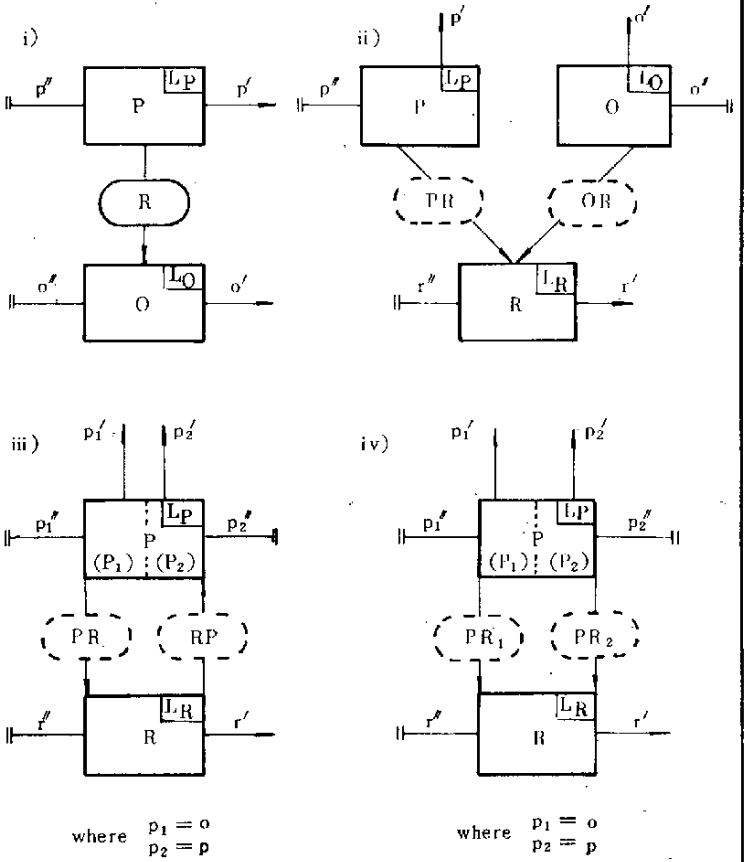
ブロック名	問合せ図	DBTG図式
ES—RS—ES ブロック	 o，r，pは、リレーションO，R，Pの組変数 OとPは、ESR．Rは、OとPを関連づける RSR． これらのリレーションは O(o1，o2)， R(o1，p1) or R(o1,p1,r) O(p1，p2)，である。 o1,p1は、OとPのkey属性の集合 o′とo″，r′とr″，p′とp″は、各々O，R，Pの属性の部分集合である。o1′とp1′は各々o1とp1の部分集合である。	

図2—12 ブロックとDBTG図式との対応(1)

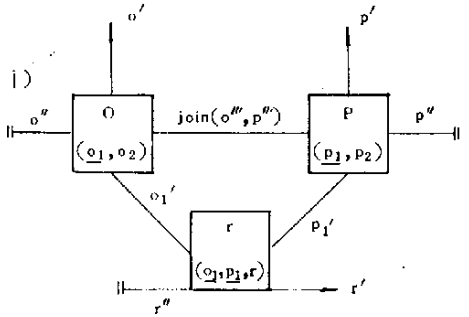
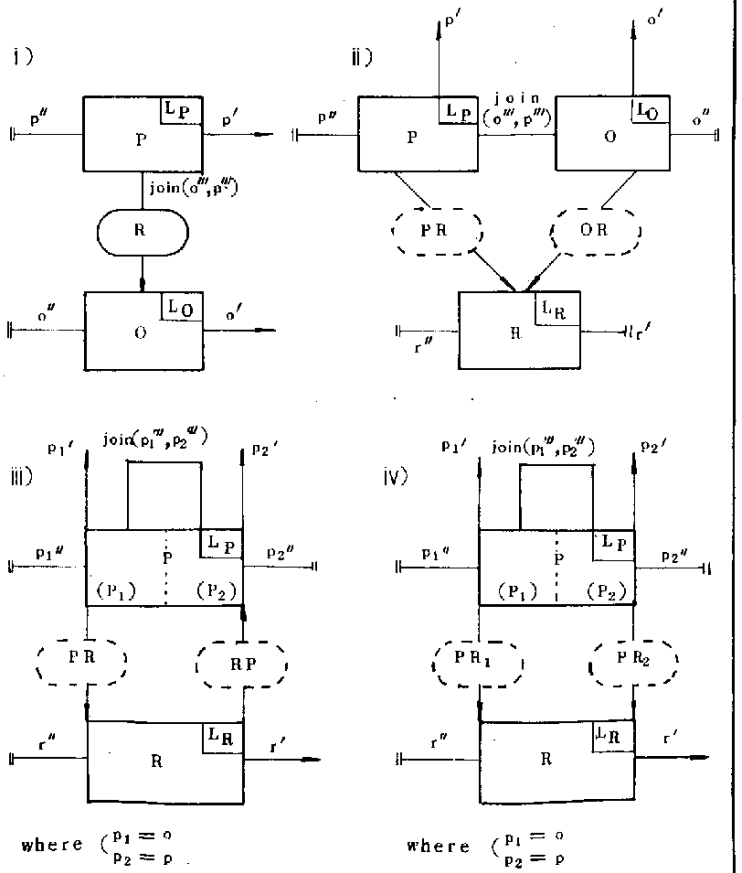
ブロック名	問合せ図	DBTG図式
ループ ES−RS−ES ブロック		

図 2－12 ブロックとDBTG図式との対応(2)

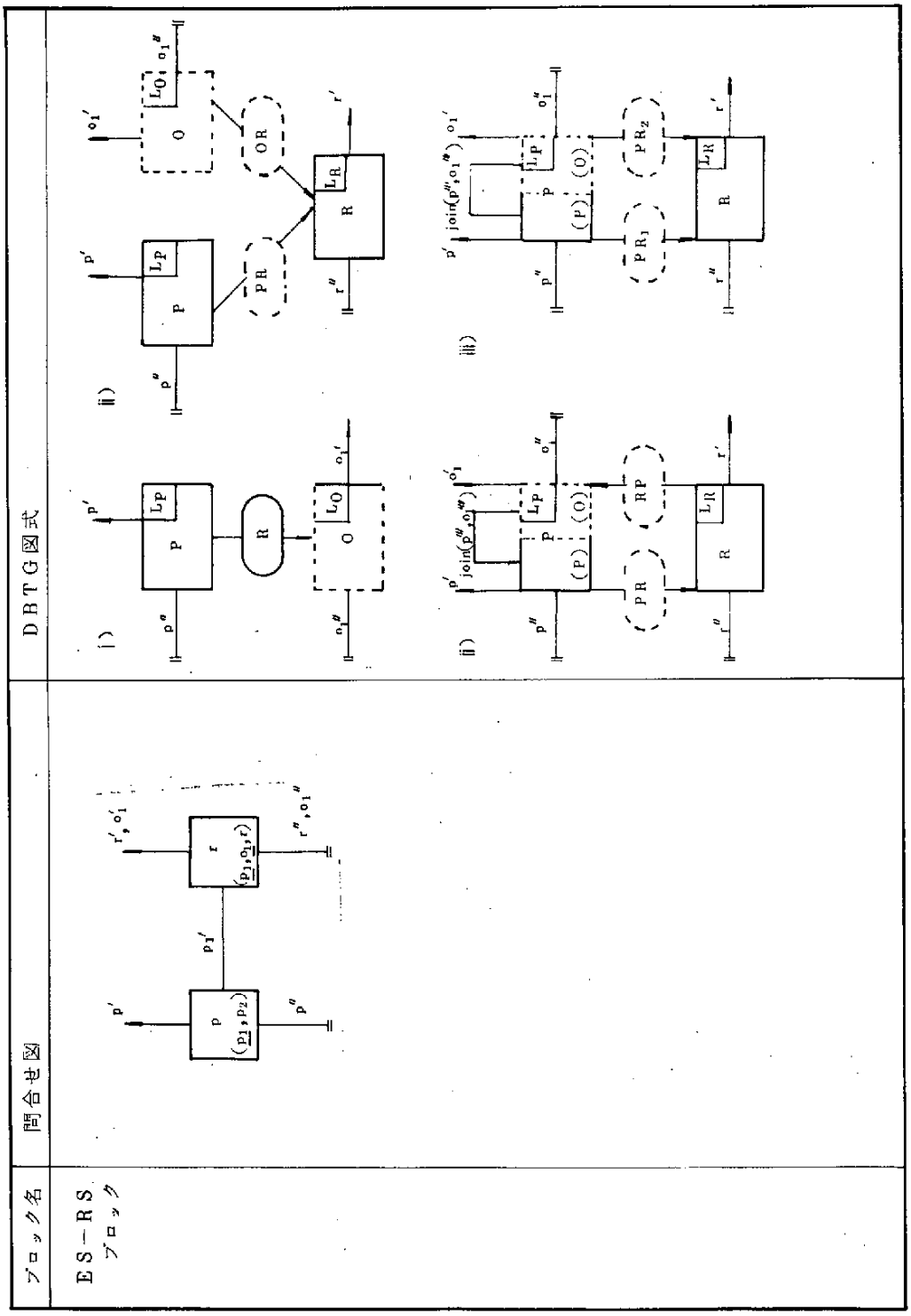


図２－12 ブロックとＤＢＴＧ図式との対応(3)

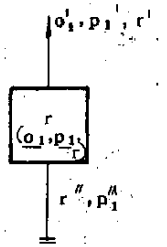
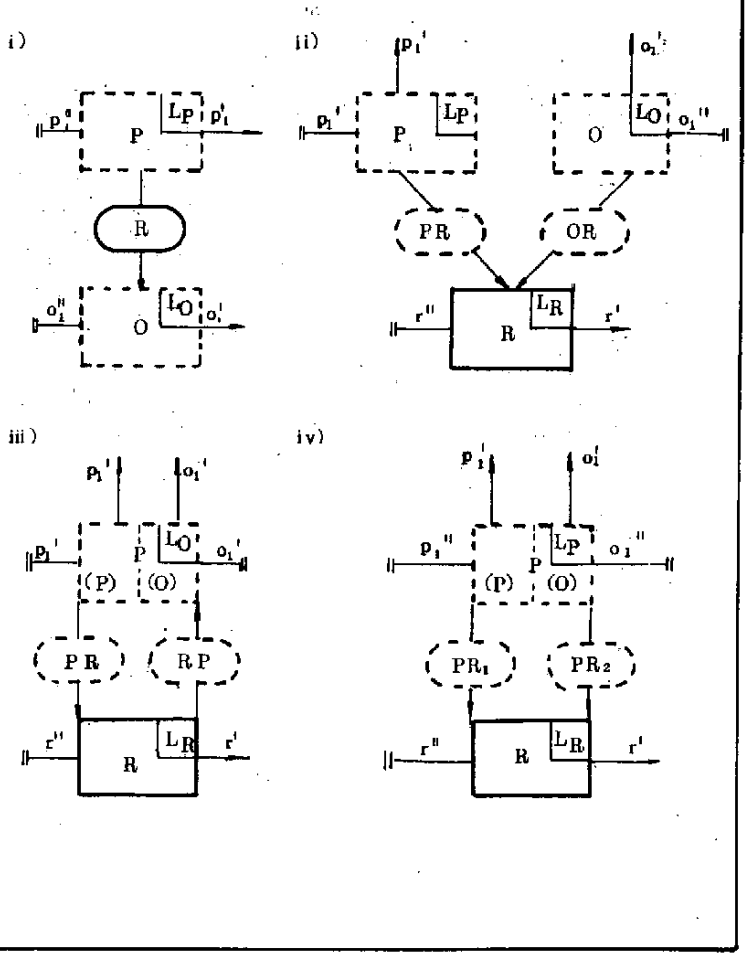
ブロック名	問合せ図	DBTG図式
RSブロック		

図2－12 ブロックとDBTG図式との対応(4)

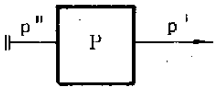
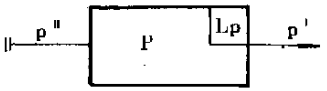
ブロック名	問合せ図	ＤＢＴＧ図式
ＥＳブロック	 リレーションPは　ESRである。 Pは、どのノードとも結合リンクをもっていない。 即ちPは、独立なレコード型(isolated record-type)を表わす。	

図２－１２ブロックとＤＢＴＧ図式との対応(5)

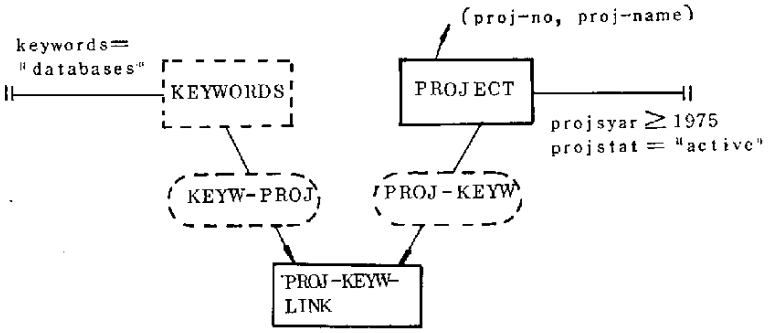


図２－１３　部分図（図２－１０）に対応するＤＢＴＧ図式

B. 分割された問合せの関係代数的表現

以上、この様にして分割された問合せの最適な実行順序を決定することが次の問題である。この実行順序の決定は、最適化問題である。この最適化において、各リレーションの組数（cardinality）、各属性のとり得る値（これの集合は領域（domain））の数、及びこのリレーション内のユニークな値の数についての情報が必要となる。ＲＳリレーションについてのこの種の情報は、このＲＳＲによって関連づけられる２つのＥＳの対応濃度（cardinality）の平均値を表わしている。これらの情報をもとに、最大数のユニークな実現値をもつ属性を結果リンクとして持つリレーションを、アクセスの終端とした場合の実現値（occurrence）の traverse 数が最少となるように順序を決定する。

問合せＱは、上述したようなアルゴリズムを用いて２つの問合せＱ１とＱ２に分割されたとする。ここで、Ｑ１が実行された後にＱ２が実行される。Ｑ１とＱ２はＱＵＥＬを用いて表わすと次のように記述できる。

```
Q1: retrieve into NEW-PROJ (j.projname,j.proj-no)
    where pkl.keywords = "databases"
    and pkl.proj-no=j.proj-no and j. projsyar ≧ 1975
    and j.projstat = "active" ;
Q2: range of(nj) (NEW-PROJ) ;
    retrieve into RESULT (nj.projname,r.reprname)
    where nj.proj-no=rp.proj-no and rp.repr-no=r.repr-no ;
```

(5)

図２－１４　分割された問合せ

分割された問合せには、前述したように、変換目標のＤＢＭＳのデータ構造を完全には表わしていない。即ち、隠れた構造をもっている。よって、これらの情報を対応するＨＩを用いて付加せねばならない。ここである問合せは、次の順番の問合せ別に分割されたとする：$Q_1 \rightarrow Q_2 \rightarrow \cdots \rightarrow Q_n$。各問合せ$Q_i$（$i=1, \cdots, n$）は、アクセス単位に対応している。ＤＢＴＧモデルでは、図２－１１のどれかのパターンに対応している。Q_iはQ_{i-1}と、切断ノードに対応

するリレーションを共有している（i＝2，3，…，n）。これらより、問合せQ_iは次のように記述できる。

$$
\left.
\begin{aligned}
Q_i : OUT_{i-1}[att_{i-1}] \longrightarrow & \\
& P[restr_P][join_{PR}]R[restr_R][join_{RO}] \\
& O[restr_O] \longrightarrow OUT_i[att_i] \\
Q_1 : P[restr_P][join_{PR}]R[restr_R][join_{RO}]O[restr_O] & \\
& \longrightarrow OUT_2(att_2)
\end{aligned}
\right\} \quad (6)
$$

ここで、PとOは、Q_iに対応するアクセス単位〔図２－１１〕内の事象集合（ES）でる。これらは、Q_iに対応する部分図では見えないかもしれない。Rは、PとOとの関係性を表わすリレーション（RSR）である。$restr_P$, $restr_R$, $restr_O$は、各々リレーションP、R、Oについての制限句の接合を表わしている。$join_{PR}$と$join_{RO}$は、各々PとR、RとPとの結合句の接合を表わしている。これらは、離接も可である。OUT_{i-1}は、Q_{i-1}によって生成されるリレーションであり、OUT_iは、Q_iによって生成されるリレーションである。att_{i-1}は、Q_{i-1}の結果の属性のリストである。att_{i-1}には、Q_1，……，Q_{i-1}の結果リストを包んでいる。att_iは、Q_iの結果の属性リストである。$att_i \supseteq att_{i-1}$である。さらに、$\{att_{i-1}, result_P, result_R, result_P\} \subseteq att_i$である。

Q_1とQ_2は、この記述法を用いて次の様に記述できる。

$$
\left.
\begin{aligned}
Q_1' : & \text{KEYWORDS [keywords = "databases"][keywords = keywords]} \\
& \text{PROJ-KEYW-LINK [proj-no = proj-no] PROJECT [projsyar} \geq 1975, \\
& \text{projstat = "active"]} \rightarrow \text{NEW-PROJ [j.proj-no, j.projname]} \\
Q_2' : & \text{NEW-PROJ [nj.proj-no, nj.projname]} \rightarrow \text{PROJECT} \\
& \text{[proj-no = proj-no] REPR-PROJ [repr-no = repr-no]} \\
& \text{REPRESENTATIVE} \rightarrow \text{RESULT [j.projname, r.reprname]}
\end{aligned}
\right\} \quad (7)
$$

C. 局所—局所問合せ変換（LLQT）

(6)の形式であらわされた問合せは、アクセスの基本単位に対応している。DBTGモデルではこの基本単位は図２－１１にのせられた４つのうちのいずれかである。

先ず、１つのアクセス単位について、アクセスパスを考えてみよう。アクセスパスは、次の事項に基づいて決定される。

(i) 各レコード型のロケーションモード（location mode）

(ii) 各レコード型に関連した制限句（restrictive clause）

ロケーションモードとしては、CALCモード（Cと記す）と、Secondaryモード（Sと記す）との２つを考える。リンク用のレコード型のロケーションモードは、Sとする。あるレコード型に関連した制限句がCALC項目を参照している等価制限句であるならば、アクセスパスとして

これを用いることは有効である。

アクセスパスを示すために、表２－４に示すような記述法を導入する。 円 は、レコード型を表わし、矢印はアクセスパスを表わす。この記述法に基づいて、図２－１１に示した各アクセス単位が持ち得るアクセスパスを表２－５に示す。さらに、この表は、各アクセスパスに対応するＤＢＴＧのＤＭＬも記されている。ネットワークにおけるレコード型ＯとＰ、ネットワークサイクルにおける変数ＯとＰ、について、対称性を考慮して逆の位置のものはのせてない。

例えば、階層構造をもつアクセスパスを考えてみよう。このアクセス単位は親レコード型Ｐ、子レコード型Ｏ、セット型Ｒとから成っている。これに対するアクセスパスは、エントリレコード型かＰ（１，３）か、Ｏ（３，４）か、両方（２）かに大きく分けれる。さらに、エントリが、ＣＡＬＣによるアクセス（１，３）か順次アクセス（３，４）かによって、２つに分けられる。アクセスパス２は、いわゆるｆｏｒｍａｔ７〔ＤＡＴＥ７７〕に基づいたもので、ＯとＰがともにＣＡＬＣ項目を参照する等価制限句をもっている場合である。

表２－４　ＤＢＴＧアクセスグラフ(1)

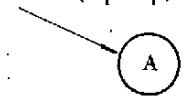
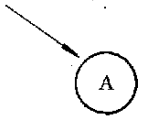
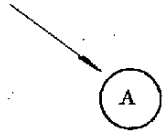
アクセス名	アクセスグラフ	説　　明	DBTG DML
first entry(1)	CALC(a_i=v_i)  	A　＝レコード型 a'_i　＝AのＣＡＬＣ項目名又はその集合 v_i　＝ＣＡＬＣの値 AはＣＡＬＣ項目 a'_i と値 v_i とを用いてアクセスされる。この時Aのロケーションモードは C（CALC）でなければならない。	MOVE vi TO ai IN A. FIND ANY A.
first entry(2)	first A  	レコード型Aの第１実現値がアクセスされる。この時、Aのロケーションモードは S（secondary）であるか、ＣＡＬＣ項目に対する等価制限句をもたない時である。	FIND FIRST A WITHIN PROAREA.
interative entry(1)	next CALC(a'_i=v_i)  	ＣＡＬＣ項目を参照する等価制限句が他にあれば、これを用いてアクセスする。	if no other CALC clause,goto QUIT, select CALC clause. MODE v_i TO a_i IN A. FIND ANY A.

表２－４　ＤＢＴＧアクセスグラフ(2)

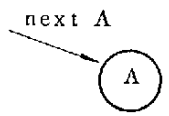
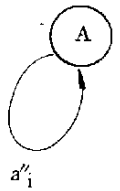
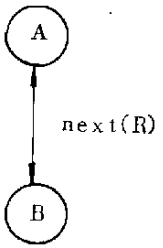
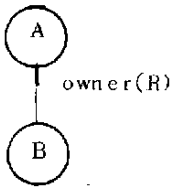
iterative entry (2)		レコード型Aの現在指示子（currency）の次の実現値をアクセスする。	FIND NEXT A.
restriction		制限句（restr（ai″））を用いて現在指示子の示す実現値に制限（restriction）を加える。 a''_i ＝レコード型Aの項目又はその集合	if an A occurrence just found by FIND satisfies all the restrictive clauses pertinent to item(s) ai″, then next step else next iterate.
traverse set (1)		AとBはレコード型、RはAとBの間のセット型である。 Aは、親レコード型とする。 Bは、子レコード型とする。 親から子へ、セット型Rを介してアクセスする。	FIND NEXT B WITHIN R.
traverse set (2)		子から親へ、セット型Rを介してアクセスする。	FIND OWNER WITHIN R.

表２－４　ＤＢＴＧアクセスグラフ(3)

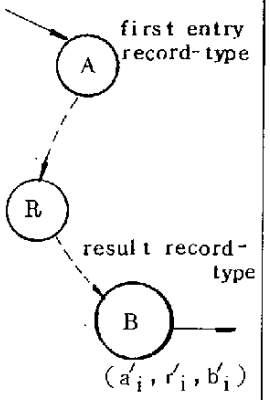
output		Bは、１つのアクセスパスの最終レコード型（我々は、結果レコード型と呼ぶ）である。 （ a'_i 、r'_i 、b'_i ）は、このアクセスによって生成されるデータの集合を表わす。a'_i b'_i 、r'_i は、各々レコード型A、B、Rのアイテム名又はその集合である。	write (a'_i, r'_i, b'_i)

表２－５　ＤＢＴＧ図式とアクセスパス(1)

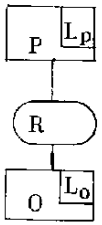
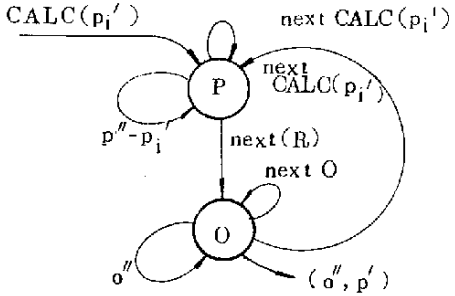
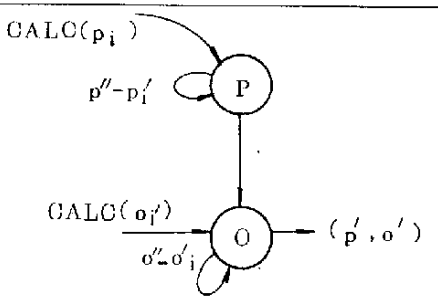
アクセス単位	アクセスパス番号	アクセスグラフ	DBTG DML
階層構造  	1	 p_i' は、レコード型ＰのＣＡＬＣ項目 $p''-p_i'$ は、p_i' に関する等価制限句を除いた制限句の接合（or離接）をあらわす。	A1 if no CALC clause exists, goto QUIT. MOVE v(p_i') TO p_i' IN P. FIND ANY P. if notfound='YES', goto A1. GET P. if ($p''-p_i'$) unsatisfied or R empty, goto A1. A2 FIND NEXT O WITHIN R. if endset='YES', goto A1. GET O. if p'' unsatisfied, goto A2. write (p', o'). goto A1.
	2	 *) このアクセスパスは、いわゆるFORMAT 7［DATE77］にもとづいている。	MOVE v(p_i') TO p_i' IN P. FIND ANY P. if notfound='YES', goto QUIT. GET P. if ($p''-pi'$) unsatisfied, goto QUIT. MOVE v(oi') TO o_i' IN O. FIND O WITHIN R CURRENT USING p_i' IN P. if notfound='YES', goto QUIT. GET O. if ($o''-oi'$) unsatisfied, goto QUIT. write (p', o').

表２－５　ＤＢＴＧ図式とアクセスパス(2)

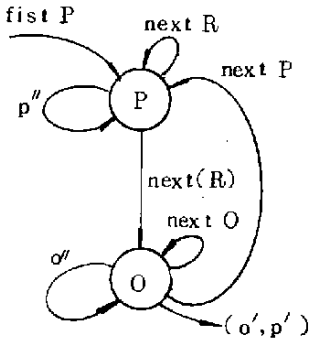
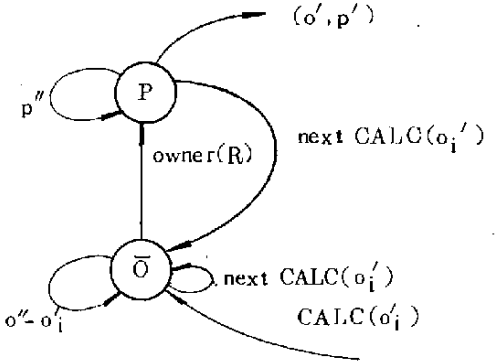
階　層	3		FIND FIRST P WITHIN PROAREA. goto A2. A1 FIND NEXT P. A2 if endset='YES',goto QUIT. GET P. if p″unsatisfied or R empty,goto A1. A3 FIND NEXT O WITHIN R. if endset='YES',goto A1. GET O. if o″ unsatisfied, goto A3. write (p′, o′). goto A3.
	4		A1 if no CALC clause exists, goto QUIT. MOVE v(o_i') TO o_i' IN O. FIND ANY O if notfound='YES',goto A1. GET O. if o″ unsatiqfied or R empty, goto A1. A3 FIND OWNER WITHIN R. if notfound='YES',goto A1. GET P. if p″ unsatisfied, goto A3. write (p′, o′). goto A3.

表２－５　ＤＢＴＧ図式とアクセスパス(3)

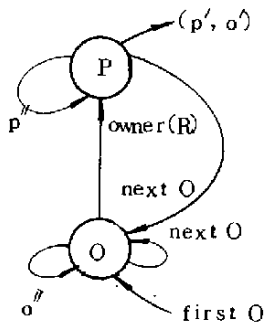
階　層	5		FIND FIRST O WITHIN PROAREA. goto A2. A1 FIND NEXT O. A2 if endset='YES', goto QUIT. GET O. if o″ unsatisfied or R empty,goto A1. A3 FIND OWNER WITHIN R. if notfound='YES', goto A1. GET P. if p″ unsatisfied, goto A1. write (p′, o′). goto A1.

表２－５　ＤＢＴＧ図式とアクセスグラフ(4)

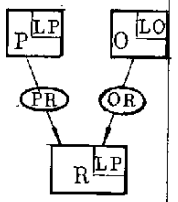
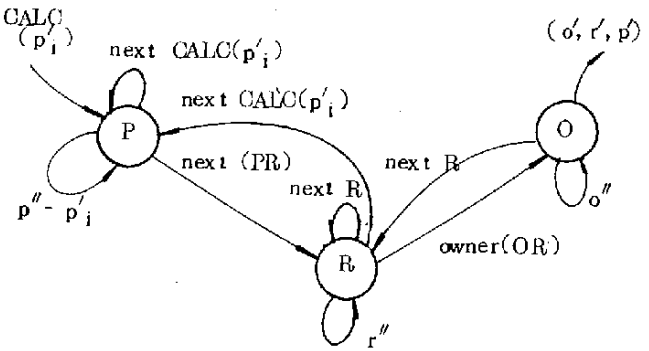
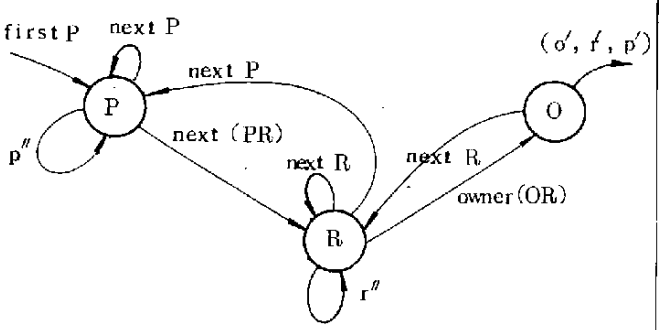
アクセス単位		アクセスグラフ	DPTG DML
ネットワーク	6	 	A1 if no CALC clause exists, goto QUIT. MODE v(p'_i) TO p'_i IN P. FIND ANY P. if notfound='YES', goto A1. GET P. if ($p''-p'_i$) unsatisfied or PR empty, goto A1. A3 FIND NEXT R WITHIN PR. if endset='YES', goto A1. GET R. if r'' unsatisfied or OR empty, goto A3. FIND OWNER WITHIN OR. if notfound='YES', goto A3. GET O. if o'' unsatisfied, goto A3. write (p', r', o'). goto A3.

表２－５　ＤＢＴＧ図式とアクセスグラフ(5)

ネットワーク	7	


```
    FIND FIRST P WITHIN PROAREA.
    goto A 2.
A1. FIND NEXT P.
A2. if endset='YES', goto QUIT.
    GET P.
    if p″ unsatisfied or PR empty, goto A 1.
A3. FIND NEXT R WITHIN PR.
    if endset='YES', goto A 1.
    GET R.
    if r″unsatisfied or OR empty, goto A 3.
    FIND OWNER WITHIN OR.
    if notfaund='YES', goto A 3.
    GET O.
    if o″ unsatisfied, goto A 3.
    write (p′, r′, o′).
    goto A 3.
```

表２－５　ＤＢＴＧ図式とアクセスグラフ(6)

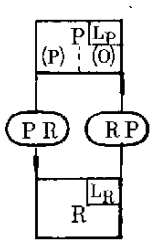
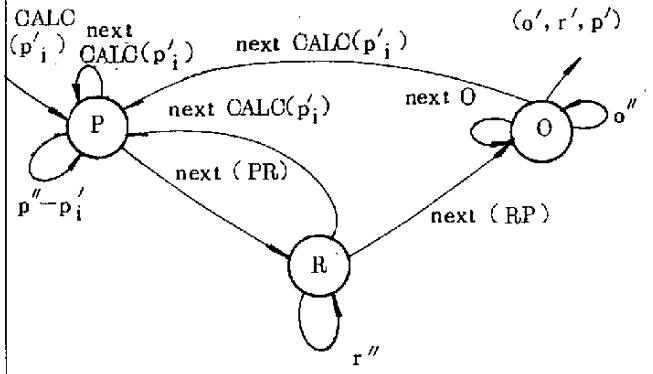
階層サイクル  	8	 	A1. if no CALC clause exists, goto QUIT. MOVE v(p'_i) TO p'_i IN P. FIND ANY P. if notfound='YES', goto A1. GET P. if ($p''-p'_i$) unsatisfied or PR empty, goto A1. FIND NEXT R WITHIN PR. if endset='YES', goto A1. GET R. if r″ unsatisfied or RP empty, goto A1. A4. FIND NEXT P WITHIN RP. if endset='YES', goto A1. GET P. if p″ unsatisfied, goto A4. goto A4.

表２－５ＤＤＢＴＧ図式とアクセスグラフ(7)

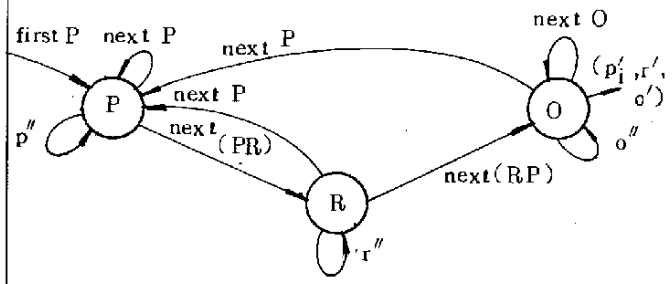
階層サイクル	9		FIND FIRST P WITHIN PROAREA. goto A2. A1 FIND NEXT P. A2 if endset='YES', goto QUIT. GET P. if p″ unsatisfied or PR empty, goto A1. FIND NEXT R WITHIN PR. if endset='YES', goto A2. GET R. if r″ unsatisfied or PR empty, goto A1. A4 FIND NEXT P WITHIN RP. if endset='YES', goto A1. GET P. if o″ unsatisfied, goto A4. write (c′, r′, o′,). goto A4.

表２－５　ＤＢＴＧ図式とアクセスグラフ(8)

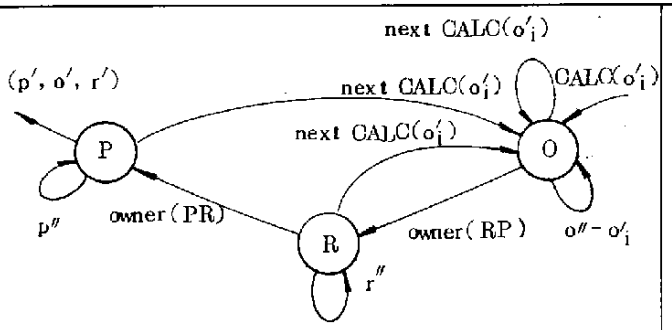
階層サイクル	10		A1 if no CALC clause exists, goto QUIT. MOVE v(o'_i) TO o'_i IN P. FIND ANY P. if notfound='YES', goto A1. GET P. if $o''-o'_i$ unsatisfied or PR empty, goto A1. FIND OWNER WITHIN RP. if notfound='YES', goto A1. GET R. if r'' unsatisfied or PR empty, goto A1. FIND OWNER WITHIN PR. if notfound='YES', goto A1. GET P. if p'' unsatisfied, goto A1. write (p', r', o'). goto A1.

表２－５　ＤＢＴＧ図式とアクセスグラフ(9)

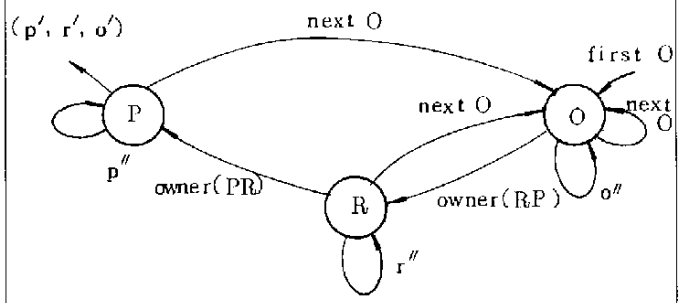
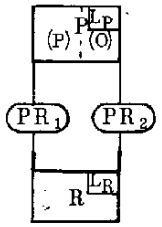
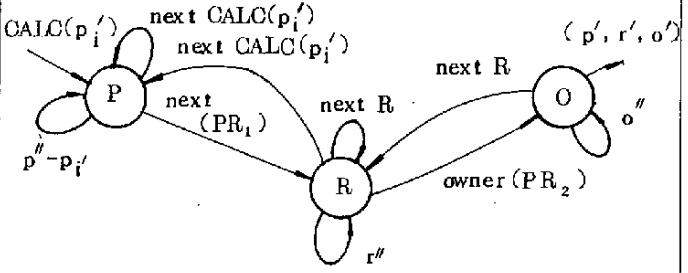
階層サイクル	11		FIND FIRST P WITHIN PROAREA. goto A 2. A1 FIND NEXT P. A2 if endset＝'YES', goto QUIT. GET P. if o″ unsatisfied or RP empty, goto A 1. FIND OWNER WITHIN RP. in notfound＝'YES', goto A 1. GET R. if r″ unsatisfied or PR empty, goto A1. FIND OWNER WITHIN PR. if notfound＝'YES', goto A 1. GET P. if p″ unsatisfied. goto A 1. write (p′, r′, o′). goto A 1.

表２－５　ＤＢＴＧ図式とアクセスグラフ(10)

ネットワークサイクル　12



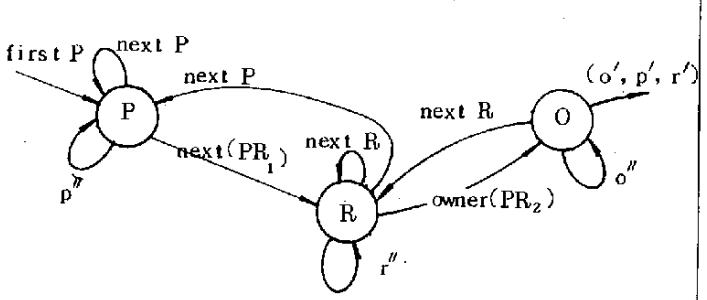


A1. if no CALC clause , goto QUIT.
MOVE v(pi′) TO p_i' IN P.
FIND ANY P.
if notfound＝'YES', goto A1.
GET P.
if (p″−pi′) unsatisfied or PR_1 empty,
goto A1.

A3. FIND NEXT R WITHIN PR_1.
if endset＝'YES', goto A1.
GET R
if r″ unsatisfied or PR_2empty,
goto A3.
FIND OWNER WITHIN PR_2.
if endset＝'YES', goto A3.
GET P.
if o″ unsatisfied, goto A3.
write (o′, p′, r′).
goto A3.

表２－５　ＤＢＴＧ図式とアクセスグラフ(11)

ネットワーク サイクル	13	




```
    FIND FIRST P WITHIN PROAREA.
    goto A2.
A1  FIND NEXT P.
A2  if endset='YES', goto QUIT
    GET P.
    if p″ unsatisfied or PR  empty, goto A1.
A3  FIND NEXT R WITHIN PR1.
    if endset='YES', goto A1.
    GET R.
    if r″ unsatisfied or PR2 empty, goto A3.
    FIND OWNER WITHIN PR2.
    if notfound='YES', goto A3.
    GET P.
    if o″ unsatisfied, goto A3.
    write (p′, r′, o′).
    goto A3.
```

表２－６は、i)とii)を入力情報として、表２－５にのせたアクセスパスがどのように決定されるかを示している。この表で $restr_P$ と $restr_O$ は、レコード型PとOに関する制限句のパターンを示している。Cは、制限句のなかに、ＣＡＬＣ項目を含む等価制限句があることを示している。ｎ－Cは、制限句をもっているが、ＣＡＬＣ項目を含む等価制限句はないことを示している。ｎは、制限句はもっていないことを示している。

表２－６　アクセスパスの決定テーブル

アクセス単位	$restr_P$	$restr_O$	アクセスパス番号
階層	C	C	2
	C	n － c	1
	C	n	1
	n － c	C	4
	n － c	n － c	3
	n － c	n	3
	n	C	4
	n	n － c	5
	n	n	3
ネットワーク	C	C	6
	C	n － c	6
	C	n	6
	n － c	n － c	7
	n － c	n	7
	n	n	7
階層サイクル	C	C	8(or 10)
	C	n － c	8
	C	n	8
	n － c	C	10
	n － c	n － c	9(or 11)
	n － c	n	9
	n	C	10
	n	n － c	11
	n	n	9(or 11)
ネットワークサイクル	C	C	12
	C	n － c	12
	C	n	12
	n － c	n － c	13
	n － c	n	13
	n	n	13

例えば、(7)のQ'_1を考えてみよう。レコード型KEYWORDSは、CALC項目を参照する等価制限句（即ち、keywords＝〝databases〞）を持つ。一方、PROJECTは、CALC項目に関する制限句を持たない。よって、このQ'_1に対応するアクセスパスは表２－６より６となる〔図2−15〕。このDBTG DMLは、図２－１６のようになる。

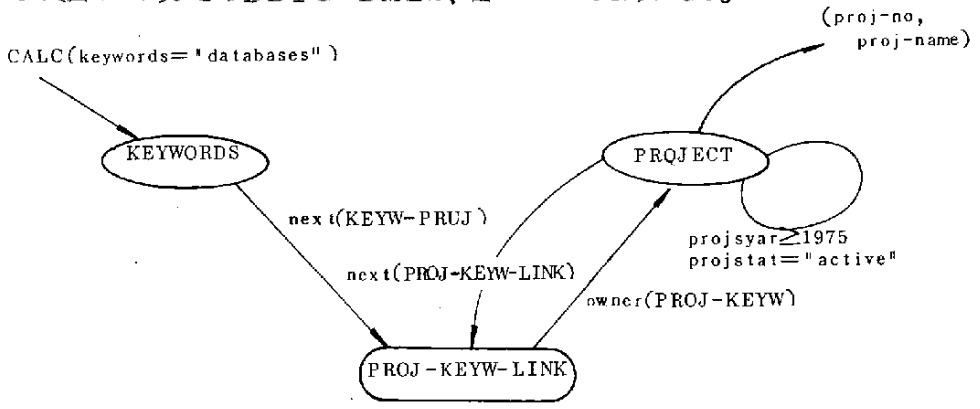


図２－15　Q'_1 (7)のアクセスパス

```
        move 〝databases〞 to keywords in KEYWORDS.
        FIND ANY KEYWORDS.
        if notfound='YES' or KEYW-PROJ empty, go to QUIT.
LOOP    FIND NEXT PROJ-KEYW-LINK WITHIN KEYW-PROJ.
        if endset='YES', go to QUIT.
        FIND OWNER WITHIN PROJ-KEYW.
        if notfound='YES', go to LOOP.
        GET PROJECT.
        if projsyar≧1975 and projstat=〝active〞,
        write NEW-PROJ (proj-no, projname).
        go to LOOP.
```

図２－１６　Q'_1に対応するDBTG DML

ここまでは、１つのアクセス単位について、可能なアクセスパスと、その最適なパス決定について論じてきた。次に、この様なアクセス単位を１つのブロックとした場合、こうしたブロック間のアクセスパスを考えねばならない。

DBTGモデルにおいて、そのDMLは、完全（complete）ではない。即ち、DBTG DMLによって導出（derive）された結果を、DBTGモデルにおけるレコード型としてDMLによってアクセス出来ない。このため、１つのブロックの処理ごとに、中間結果を生成しないことが望ましい。我々のブロック間のアクセスパスは次のようにして書かれる。結果レコード型（re-

sult record-type)は、次のブロックと共有されているものを選ぶ。DML記述のwrite(p',o')(又はwrite(p',o',r'))部を次のブロックのDML記述におきかえる。この方法は、問合せ図内にループが存在する場合には困難である。DBTGモデルにおいて、一般にこのような問合せは少ないと考える。

2.5.2 異種性情報(HI)

異種性情報(HI)は、局所内部スキーマ(LIS)と局所概念スキーマ(LCS)との間の変換情報である。HIは後述する分散情報(DI)とともに、ネットワークデータディレクトリ(NDD)として統一的に管理される。ここで、HI_{ij}をi番目のサイト内のj番目のデータベースに関する異種性情報とする。各HI_{ij}は各データベースサイトで、正規化されたリレーションとして管理される。HIは、次の5種類のリレーションから成り立っている。

(i) ESR
(ii) RSR
(iii) ATTRIBUTE
(iv) INTEGRITY
(v) PROTECTION

ESRは、事象集合(ES)の属性の数(degree)、組の数(cardinality)、キー属性の数等の情報をもっている。RSRは、関係性集合(RS)の関連づけるESとRSRの属性との情報をもっている。RSRは、対応するDBMSの構文的関係性を表わしている。ATTRIBUTEは、各属性のフォーマット、値集合、ESR又はRSR内の役目等の情報を保持している。INTEGRITYとPROTECTIONリレーションは、各リレーションに関する完全性(integrity)とプロテクションの条件を表わしている。これらの条件は関係計算形式で表わされており、問合せの処理時に、その条件部に接合される[STON76]。表2-7は、これらのリレーションのスキームを表わしている。

HIは、LISからLCSが生成される時に、各サイトの局所管理者(LA)によって同時に生成される[2.3.2を参照]。HIは、LCSとLISとの間のスキーマの要素の対応関係とともに、属性のrole,cardinality、size、width といったパフォーマンス情報も保持している。これらの情報は、問合せの処理、一致性及び同時実行性、エラー回復等において用いられる。

表2－7　HIのリレーションスキーム

relation name	attribute name	meaning
ESR	ESR−name	the name of ES relation
	owner	owner of the ESR
	no-keys	the number of keys
	degree	the number of attributes
	cardinality	the number of tuples
	width	the length of tuple
	size	width * cardinality
	protection	e.g. access control of DBTG, sensitivity of IMS
	integrity	e.g. check of DBTG
	ES−type*)	location mode, area name, root segment
ATTRIBUTE	ESR/RSR name	the name of ES/RS relation
	attribute-name	the name of attribute
	role	key, composite-key, or non-key
	v-name	value-set name to which the attribute belongs
	type	integer or character
	length	byte length of the attribute
	cardinality	the number of distinct values of the attribute
	protection	
	integrity	
RSR	RSR−name	the name of RS relation
	source-ESR-name	the names of two ESRs which are connected via the RSR
	dest-ESR-name	
	protection	
	integrity	
	set-type*)	source-set-name and dest-set-name
	shared-domain**)	the name of shared by the two ESRs

*) for DBTG mdel and IMS model
**) for relational mdel

2.5.3　その他の問題

以上で述べてきたことの他に、次のようなものがある。1つは、いかに実際のDBMS上でDMLを実行させるかである。これは、局所DBMSが、どのような通信レベルを提供しているかに依存している。DBMSが問合せ言語を備えていれば、この問題は比較的容易であろう。DBMSが親言語（host language）レベルのみをサポートしているならば、一連のDMLから成るプログラムを1つのジョブとして生成せねばならない。これは前者に対して、困難であると思われる。

もう1つは、障害の回復（error recovery）問題である。この問題については、障害を起こした演算の撤退（backout）、データベースのダンプ、交信の記録、チェックポイント、及びダンプを用いてデータベースを復旧し、最後のチェックポイントまでデータの更新の実行すること等を検討する必要がある。

さらに、ネットワークを使用するための支援機能（assistant）の問題がある。この機能として、

多重結合、メッセージの複数サイトへの拡散（broadcasting）、用いる通信プロトコルの選択、各ホストへの共通ログロン／オフ手順のサポート等をあげることができる。

2.6 分散問題 (Distribution problem)

本節では、種々のDBMSがコンピュータネットワークを介して結合されていることによって生ずる分散問題を論じる。各サイト単位で同種化（homogenize）されたDBMSスキーマ（即ち、局所概念スキーマ（LCS））を基にして、あるネットワーク共同体（network community）にとって意味のあるデータの記述、即ち全体概念スキーマ（GCS）を構成する問題（統合化）をまず論ずる〔2.6.1と2.6.2〕。2.6.3では、GCSとLCSの対応情報である分散情報について論ずる。さいごに、GCSに基づいた問合せ（全体問合せと呼ぶ）を、各LCSのRDのリレーションに対する問合せ（局所問合せと呼ぶ）に分割する問題（全体—局所問合せ分割（GLQD））を2.6.4で論ずる。

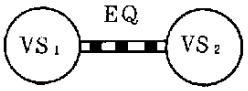
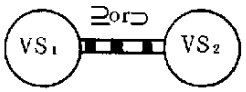
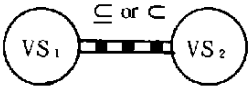
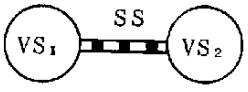
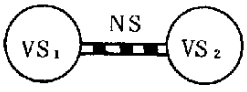
2.6.1 セマンティックリンク（semantic link）

GCSは、ネットワーク上に分散したLCSから、セマンティックリンクに基づいて全体管理者（global administrator：GA）によって統合される。セマンティックリンク（SL）は、異なったサイト（又は同一のサイト）内の値集合（value set：VS）間のリンクである。SLの存在は、これによって結合される2つの値集合が、意味論的に等価であることを表わしている。例えば、2.3の例のプロジェクト管理データベース（PMDB）と文献データベース（BLDB）とを考えてみよう。PMDBのLCS（LCS_{PMDB}と記述する）のESR MEMBERは、値集合MEM#を持っている。BLDBのLCS（LCS_{BLDB}と記す）のESR AUTHORは、値集合AUTHOR#を持っている。プロジェクトのメンバーは、各々幾つかのレポートを書き、このレポートは文献データベースに登録されているとする。この時、LCS_{PMDB}のMEM#とLCS_{BLDB}のAUTHOR#とは意味論的に等価であり、これらの間にセマンティックリンク（SL）が存在している。

さらに、SLは、これによって関連づけられる値集合の集合論的関係性も示している。これらは表2－8にまとめられる。SLは、2つの値集合VS_1とVS_2との間に存在し、集合論的関係性$\hat{R}$を表わしている時、我々は、SL：VS_1 $\hat{R}$ VS_2と記述する。Rは表2－8のように、$\hat{R} \in \{ \subset, \subseteq, \equiv, \supseteq, \supset, \cap, \cup \}$である。

$\hat{R}$が$\subset$, $\subseteq$, $\equiv$, $\supseteq$, $\supset$, 又は$\cap$である時、SLはこれによって関連づけられるリレーションが結合（join）することが可能であることを示している。これに対して$\hat{R}=NS(U)$の時、これによって関連づけられるリレーションの和（union）をとることが可能であることを示している。このため、前者を結合リンク（join-link）、後者を和リンク（union link）とも呼ぶ。

表 2 － 8　ＳＬの集合論的関係性

$VS_1 \overset{\triangle}{R} VS_2$	説　　明	
$VS_1 \equiv VS_2$ (EQ)		VS_1 の全ての値は，VS_2 にも存在し， VS_1 の全ての値は，VS_1 にも存在する。 即ち　$VS_1 \cap VS_2 \neq \emptyset$ $VS_1 - VS_2 = \emptyset$ $VS_2 - VS_1 = \emptyset$　である。
$VS_1 \left\{ \begin{matrix} \supseteq \\ \supset \end{matrix} \right\} VS_2$		VS_2 の全ての値は，VS_1 内にも存在する。 VS_2 は VS_1 の部分集合である。 即ち　$VS_1 \cap VS_2 \neq \emptyset$ $VS_1 - VS_2 \neq \emptyset$ $VS_2 - VS_1 = \emptyset$　である。
$VS_1 \left\{ \begin{matrix} \subseteq \\ \subset \end{matrix} \right\} VS_2$		VS_1 は VS_2 の部分集合である。 即ち　$VS_1 \cap VS_2 \neq \emptyset$ $VS_1 - VS_2 = \emptyset$ $VS_2 - VS_1 \neq \emptyset$　である。
$VS_1 \cap VS_2$ (SS)		VS_1 と VS_2 は各々共通の部分集合をもつ。 即ち　$VS_1 \cap VS_2 \neq \emptyset$ $VS_1 - VS_2 \neq \emptyset$ $VS_2 - VS_1 \neq \emptyset$　である。
$VS_1 \cup VS_2$ (NS)		VS_1 と VS_2 は，互いに素(disjoint)である。 即ち　$VS_1 \cap VS_2 = \emptyset$　である。

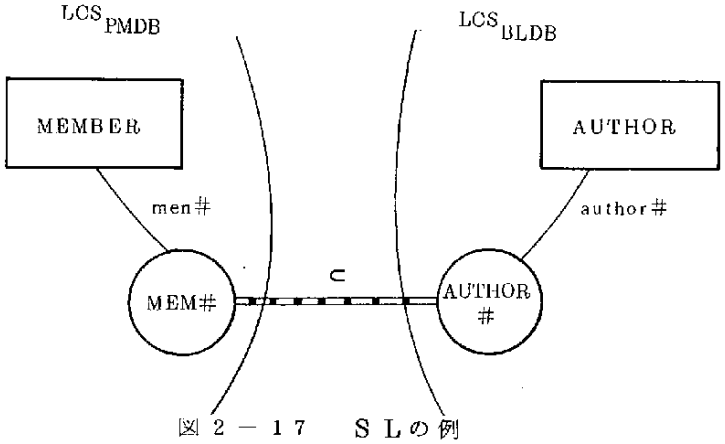


図 2 － 1 7　ＳＬの例

2.6.2 分散記述（Distribution description ; DD）

全体管理者（GA）は、セマンティックリンク（SL）を用いて、新たにESRとRSRとを定義する。あるGCSに関するこれらの定義文の集合は、分散記述（DD）と呼ばれる。構文的には、DDは、関係モデルにおけるviewの定義文に対応している。DDは、QUEL仕様に準拠した関係計算（relational calculus）の形式をもっている。 DDの形式的な定義は、図２－１８にのせてある。DDは、QUELと同様なRANGE文と、DEFINE文とからなっている。RANGE文は、LCSレベルの局所リレーションと、対応する組変数（tuple variable)及びその存在サイト番号とを宣言する。DEFINE文は、局所リレーションから、全体リレーション名とその全体属性とを関係計算形式（relational calculus form）で定義する。

RANGE文は、次の事項の定義を行なう。

1) 局所リレーション名と、その対応する組変数。
2) 局所リレーション名と、その所在サイト名。

DEFINE文は、次の事項の定義を行なう。

1) 全体リレーション名(<grel-name>) と、そのタイプ(<rel-type>) (即ち、ESR or RSR)。
2) 全体リレーション名を構成する属性（全体属性（global attribute）と呼ぶ）名及びそのフォーマットと領域名（<a-list>）。
3) 全体属性名と、局所属性（局所リレーションを構成する属性）名との対応（(<target-list>)）。
4) 全体リレーションを構成するための局所リレーションに対する条件づけ(<qualification>)。

RANGE文は、次の形式をもっている。

RANGE文::＝range of (<var-list>)(<lrel-list>);
<lrel-list>::＝<lrel-name>[:<site-no>][,<lrel-name>[:<site-no>]]

<var-list>は組変数の並びである。<lrel-name>は局所リレーション名であり、<site-no>はこのリレーションの存在するサイト番号である。

DEFINE文は、次の形式をもっている。

DEFINE文::＝define <rel-type> <grel-type> (<a-list>)
<g-constraint >[:<g-constraint>];

<grel-name>は、定義される全体リレーション名であり、その属性構成は<a-list>に示される。この全体リレーションのタイプ（RSR or ESR）は<rel-type>によって示される。
<g-constraint>は、次の形式である。

〈g-constraint〉::=〈target-list〉 where 〈qualification〉

〈target-list〉は、〈a-list〉で定義された全体属性名と、局所属性名との対応を示している。〈target-list〉は、一般に、全体属性名と、ある局所属性のとり得る値の集合(もちろん、この値は〈qualification〉部によって制限されている）とを対応させている。よって、ある全体属性の領域（値集合）が、2つの互いに素な局所属性の値集合から構成されているとすると、いわゆる〈target-list〉 where 〈qualification〉の形式では表現できない。この時、2つのVS間には、SS又はNS（和）リンクがSLとして存在している。gaをある全体属性、la_1, la_2, ・・・, la_nをn個の局所属性とする。又、VS（a）は、属性aの値集合を表わすとする。すると、

$$\left.\begin{array}{l} VS(ga) \subseteq VS(la_1) \cup VS(la_2) \cup \cdots \cup VS(la_n) \\ \forall_i \forall_j (j \neq i) : VS(la_i) - VS(la_j) \neq \emptyset \end{array}\right\} \quad (1)$$

である場合を、〈target-list〉 where 〈qualification〉では表わせない。表わせるのは、次の場合である。

$$VS(ga) \subseteq VS(la_i) \quad (2)$$

よって、〈target-list〉 where 〈qualification〉句を、我々は〈g-constraint〉と呼び、(1)を〈g-constraint〉の集合（即ち<g-constraint>{:<g-constraint>}）として表わす。即ち、(1)は、1つの〈g-constraint〉によって生成されるリレーションの和（union）として表わされる。複数のリレーションの和をとる場合には、これらのリレーションのスキームは同一でなければならない。このために、各<g-constraint>には、〈target-list〉がついている。

DD ::= <DD-statement>{<DD-statement>}

<DD-statement>::=<RANGE-statement>|<DEFINE-statement>|
<DROP-statement>

<RANGE-statement>::= RANGE OF (<var-list>)(<lrel-list>);

<var-list>::= <variable>{, <variable>}$_{\leq n}$

<lrel-list>::=<lrel-name>[:<site-no>]
{, <lrel-name>[:<site-no>]}$_{\leq n}$

<lrel-name>::=<rel-name>

<DEFINE-statement>::=DEFINE <rel-type><grel-name>(<a-list>)
<g-constraint>{:<g-constraint>};

<rel-type>::=ESR | RSR

<grel-name>::=<rel-name>

<a-list>::=<g-attribute>[/<domain-name>][=<format>]
{, <g-attribute>[/<domain-name>][=<format>]}

<g-attribute>::=<attribute>

<g-constraint>::=(<target-list>)WHERE<qualification>

<target-list>::=<t-clause>{, <t-clause>}

<t-clause>::=<g-attribute>=<a-exp>|<v-attribute>

<v-attribute>::=<variable>.<attribute>

<format>::=<f-type><f-length>

図２－１８　ＤＤの形式的定義（１）

```
<qualification> ::= <clause> | (<qualification>) |
                    NOT <qualification> |
                    <qualification> AND <qualification> |
                    <qualification> OR <qualification>

<clause> ::= <a-exp> <cop> <a-exp>

<cop> ::= LT | LE | EQ | = | GE | GT | NE

<a-exp> ::= <constant> | <v-attribute> |
            <a-exp> <aop> <a-exp> | (<a-exp>) |
            -<a-exp> | <a-function> | <g-function>

<aop> ::= + | - | * | / | **

<g-function> ::= <g-func-name> ( <a-exp>
                 [ BY <v-attribute> { BY <v-attribute> } ]
                 [ WHERE <qualification> ] )

<g-func-name> ::= ANY | AVER | COUNT | SUM | MAX | MIN |
                  UAVER | UCOUNT | USUM

<a-function> ::= <a-func-name> ( <a-exp> { , <a-exp> } )

<a-func-name> ::= ABS | MVD | EXP | LVG | ·····
```

図２－１８　ＤＤの形式的定義（２）

<alpha> ::= A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|
V|W|X|Y|Z|$

<digit> ::= 0|1|2|3|4|5|6|7|8|9

<special-letter> ::= _|#

<letter> ::= <alpha>|<digit>|<special-letter>

<variable> ::= <alpha>{<letter>}$_{\leq 3}$

<attribute> ::= <alpha>{<letter>}$_{\leq 19}$

<rel-name> ::= <alpha> {<letter>}$_{\leq 19}$

<f-type> ::= C|D

<f-lenght> ::= <digit>{<digit>}$_{\leq 3}$

<site-no> ::= <digit>{<digit>}$_{\leq 3}$

<constant> ::= <d-constant>|<c-constant>

<d-constant> ::= <digit>{<digit>}

<c-constant> ::= '' <b-letter>{<b-letter>} ''

<b-letter> ::= <letter>|ｦ|<s-letter>

<s-letter> ::= +|-|*|/|,|.|:|(|)|%|'|@|ﾞ|;

図２－１８　ＤＤの形式的定義（３）

DDの例として、次の３つの局所リレーションを統合して、全体リレーションをつくる問題を考えよう。

リレーションスキーム（relation scheme）

BOOK	(book#, name, year)	(サイト１)	(3)
DOCUMENT	(d#, dname, year)	(サイト２)	
AUTHOR	(author#, name, report#)	(サイト３)	

BOOKリレーションは、サイト１のＬＣＳ内にあり、本の番号（book#）と書名（name）及び発行年（year）とが格納されている。サイト１は文献の案内センターである。DOCUMENTリレーションは、サイト２にあり、ドキュメントの番号（d#）とその名前（dname）と発表年（year）が格納されている。サイト２はドキュメントセンターである。AUTHORリレーションはサイト３にあり、著者の番号（author#）、名前（name）と、彼の論文の番号（report#）が格納されている。論文は、同時に文献案内センターとドキュメントセンターに論文の番号と名前が登録されているので、２つのセンターは、同じ文献情報を共有している。この場合の３つのリレーションの関係は図２－１９に示される。即ち

$$VS〔AUTHOR.report\#〕 \subseteq VS〔BOOK.book\#〕$$
$$VS〔AUTHOR.report\#〕 \subseteq VS〔DOCUMENT.d\#〕$$
$$VS〔BOOK.book\#〕 \cap VS〔DOCUMENT.d\#〕 \neq \phi$$
$$VS〔BOOK.name〕 \cap VS〔DOCUMENT.dname〕 \neq \phi$$

である。

これらの３つの局所リレーションより、著者と著書（番号と名前）との関係性を表わす全体RSリレーションAUTHOR_REPの定義を行なう。このリレーションのスキームは、AUTHOR_REP（a#, aname, rep#, rep_name）である。a#は著者番号、anameは著者名、rep#はレポート番号、rep-nameはレポート名である。このDDは次のようになる。

```
range of    (a, b, d)(AUTHOR : 3, BOOK : 1, DOCUMENT : 2) ;
define      RSR AUTHOR_REP(a#, aname, rep#, rep_name, year)
(a#=a.author#, aname=a.name, rep#=a.report#,
 rep_name=b.name, year=b.year)
where       a.report#=b.book# ;
(a#=a.author#, aname=a.name, rep#=a.report#,
 rep_name=d.dname, year=d.year)
where       a.report#=d.d# ;
```

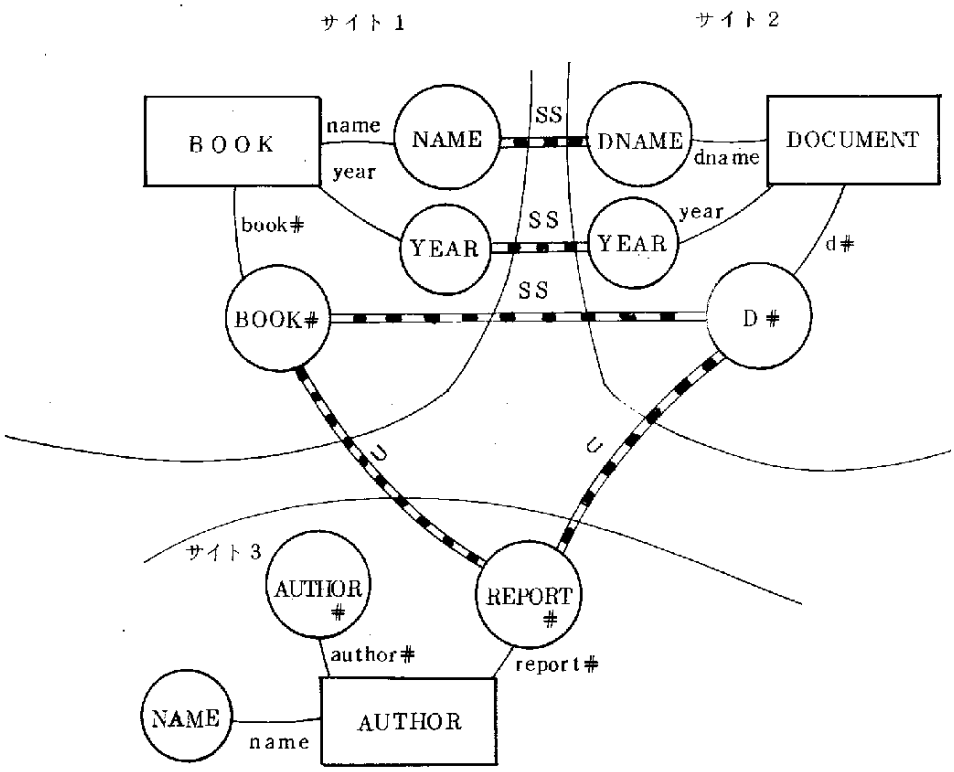


図２－１９　３つの局所リレーションの関係

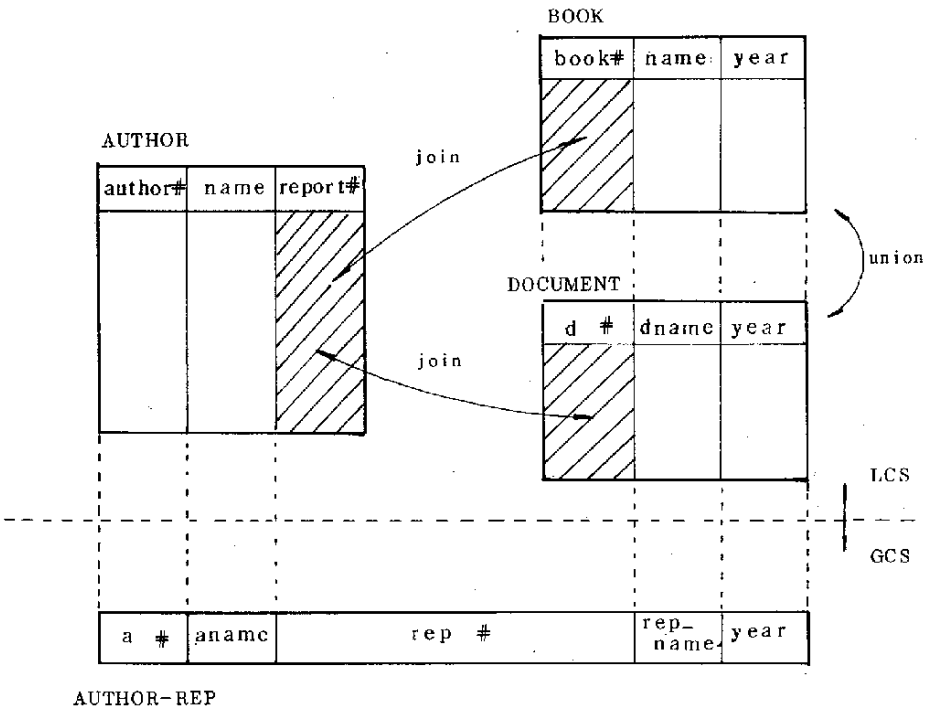


図２－２０　AUTHOR－REPと３つの局所リレーションとの関係

2.6.3 分散情報（ＤＩ）

分散情報（NDD／ＤＩと呼ぶ）は、ＧＣＳとＬＣＳとの対応情報である。NDD／ＤＩは、次の３種類の情報から成り立っている。

(i) NDD／ＤＩ／ＬＩ〔所在情報（location information）〕

(ii) NDD／ＤＩ／ＣＩ〔対応情報（correspondence information）〕

(iii) NDD／ＤＩ／ＳＩ〔スキーマ情報（schema information）〕

これらの情報は、ＨＩと同じく全てリレーションとして管理されている。ＬＩ、ＣＩ、ＤＩのスキーマを、図２－２１に示す。楕円は属性を表わし、矢印は従属関係を表わしている。

所在情報（ＬＩ）は、各サイトのＨＩから生成され、局所リレーションと局所属性名とそれらの所在サイトとの対応を示している。図において、斜線をひかれた属性は、ＨＩと共有されていることを示している。ＬＩＣＦは、完全性及びプロテクションの条件式をあらわしている。この式は、関係計算形式で表わされている。

対応情報（ＣＩ）は、全体リレーション名と、これを定義するＤＤ文（１つのＤＥＦＩＮＥ文とＲＡＮＧＥ文の集合）との対応を示している。システム内でＤＥＦＩＮＥ文は、木構造〔3.2を参照〕として表わされている。ＣＩにおける属性ＤＤは、このtree表現を実現値（instance）として持つ。l-variableは、ＤＤのＲＡＮＧＥ文によって定義された局所リレーションに対する組変数名を表わしている。

スキーマ情報（ＳＩ）は、ＧＣＳにおけるスキーマを表わしている。すなわち、全体リレーション名、次数、タイプ、及び属性名とその形式、値集合（領域）等を表わしている。これは、ＨＩのＥＳＲ、ＲＳＲ、ＡＴＴと、ほぼ等しい内容を持っている。ＳＩでは、ＨＩに存在したcardinality、size等のパフォーマンス情報を持たない。

ＤＩにおける次の問題は、前述した情報をどのようにネットワーク上で管理するか、すなわち、分散情報（ＤＩ）の分散形態である。ＤＩには、ＤＩ独自の情報（ＳＩ）と、各サイトで管理されているＨＩと共有部分を持つ情報（ＬＩとＣＩ）との２つにまず分けて考えられる。分散形態は、これらの情報の種類ごとに論じられる必要がある。ここで、分散問題を処理するプロセッサをＧＤＰ（global data processor）〔2.7を参照〕と呼ぶことにする。ＳＩは、ＧＤＰの存在するサイトには必ず必要である。

ＣＩも、問合せ処理の効率上、ＧＤＰの存在するサイトには必要である。ＣＩは、ＨＩとの共有属性として、局所リレーション名を持っている。

ＬＩは、大半の情報をＨＩと共有している。よって、ＧＤＰが全サイトについてのＬＩを持つことは、多くの格納スペースを必要とする。局所リレーション名（l-rel-name）と、その所在サイト名（site-no）との情報は、各ＧＤＰは持つべきであるが、他のcardinality、size等のパフォーマンス情報は、必要な時に各サイトから集め、一定時間格納しておくworking set〔ＳＨＡＷ７４〕方式を用いる。

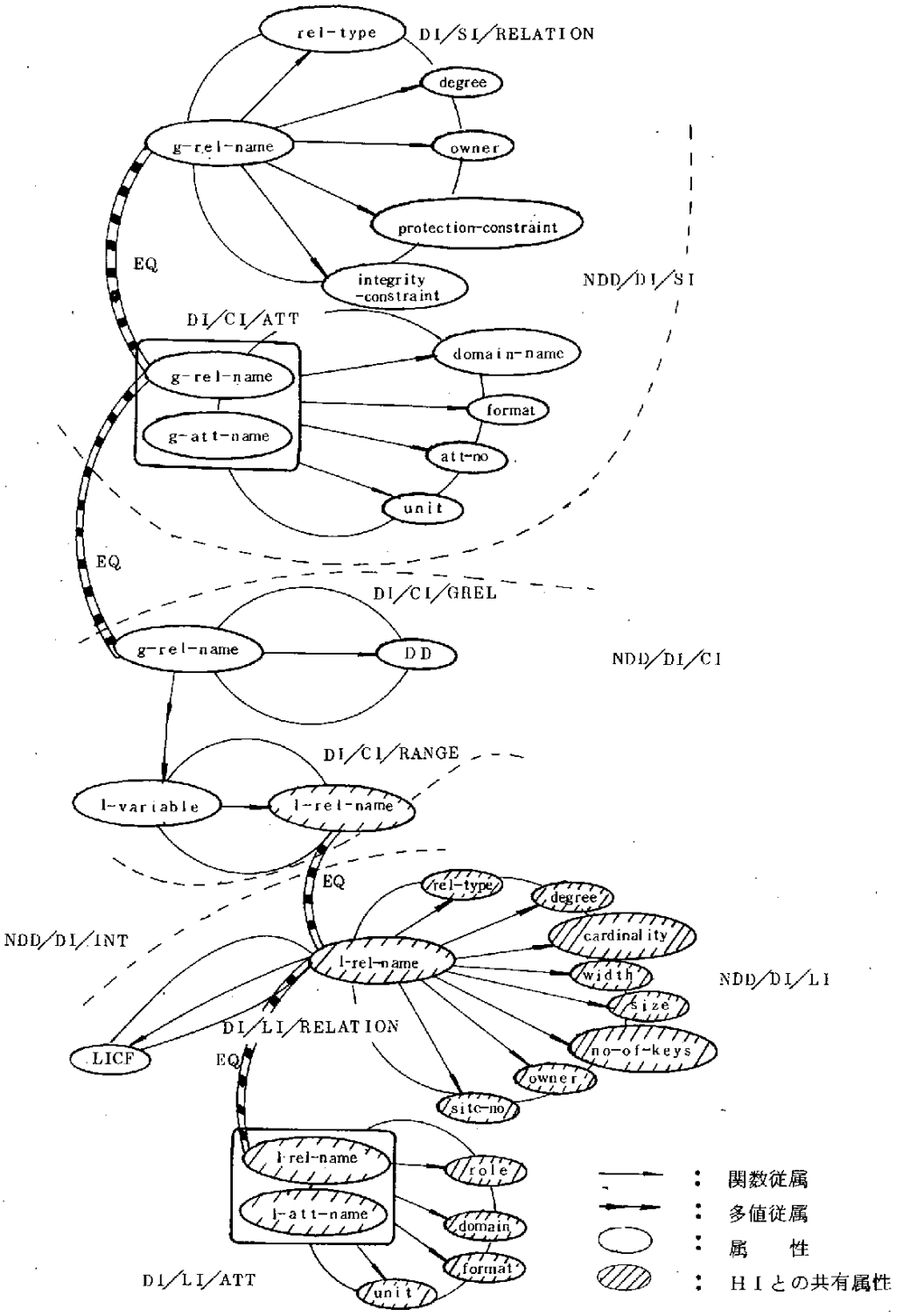

rel-type
DI/SI/RELATION
degree
g-rel-name
owner
protection-constraint
EQ
integrity
-constraint
NDD/DI/SI
DI/CI/ATT
domain-name
g-rel-name
format
g-att-name
att-no
unit
EQ
DI/CI/GREL
g-rel-name
DD
NDD/DI/CI
DI/CI/RANGE
l-variable
l-rel-name
EQ
rel-type
degree
cardinality
NDD/DI/INT
l-rel-name
width
NDD/DI/LI
size
DI/LI/RELATION
LICF
EQ
no-of-keys
owner
site-no
l-rel-name
role
l-att-name
domain
format
DI/LI/ATT
unit
：関数従属
：多値従属
：属　性
：ＨＩとの共有属性

図２－２１　ＮＤＤ／ＤＩ

2.6.4 全体ー局所問合せ分割（GLQD）

前項までに、GCS／RDの生成について論じてきた。本項では、この様にして統合されたGCSに対する問合せ（全体問合せ）を、逆に局所問合せに分割することを試みる。このプロセスを全体ー局所問合せ分割（GLQD）と呼ぶ。

A. 水平問合せ分割（HQD）

全体問合せは、GCS／RD内のESRとRSRとを参照している。DDは、全体リレーションと局所リレーションとの対応を関係計算形式（relational calculus form）の式によって表わしている。全体問合せは、まず、このDDを用いて、局所リレーションのみを参照する問合せに変換される。この問合せは、DD内の<g-constraint>句の数だけ生成される。全体問合せ内の全体属性は、対応するDEFINE文の目標リストを用いて局所属性に置き換えられる。さらに、この問合せの条件部に、<g-constraint>の条件部の論理式を接合（conjunction）する。このプロセスは、形式的には次のように示せる。Gは接合問合せ（conjunctive query）と仮定する。

全体問合せ G: range of $(g_1, \cdots, g_m)(G_1, \cdots, G_m)$;
retrieve into GR (GTL)
where GQL ;
} (1)

DD: range of $(l_1, \cdots, l_n)(L_1 : s_1, \cdots, L_n : s_n)$;
define $rtype_1$ G_1 (AL_1)
$GC_{11} : \cdots : GC_{1n_1}$;
•
•
•
define $rtype_m$ G_m (AL_m)
$GC_{m1} : \cdots : GC_{mn_m}$;
} (2)

(1)は全体問合せを表わし、(2)はDD（分散記述）を表わしている。

ここで、G_i（$i=1, \cdots, m$）は、全体リレーション名であり、g_i（$i=1, \cdots, m$）は、リレーションG_iに対する組変数である。ここで、各G_iは互いに異なる（distinct）必要はない。GRは、問合せGによって生成される結果のリレーション名である。GTLは目標リストであり、GQLは、条件式である。L_iは、サイト番号s_iに存在する局所リレーション名である。l_iはl_iの組変数である。$rtype_i$は、全体リレーションG_iがESRかRSRかを示している。AL_iは、G_iの属性リストである。GC_{ij}は、全体リレーションG_iのj番目の<g-constraint>である。n_iは、G_i内の<g-constraint>の数である。

この時、問合せGは次の問合せに分割される。

$$
\left.
\begin{array}{l}
\quad \underline{\text{range of}}\ \ (l_1, \cdots\cdots, l_n)(L_1 : s_1, \cdots\cdots, L_n : s_n)\ ; \\
G_h : \underline{\text{retrieve into}}\ \ GR_h\ (GTL') \\
\quad \underline{\text{where}}\ \ \ GQL' \wedge (GC_{ij_1} \wedge \cdots\cdots \wedge GC_{mi_m})\ ; \\
\\
\text{vh} \in \{1, 2, \cdots, n_1 \cdot n_2 ; \cdots \cdot n_m\}\ \text{v}\ (i_1, i_2, \cdots, i_m) \in \{(1,1,\cdots, 1,1), (1,1,\cdots, 1,2), \cdots \\
\qquad \cdots, (n_1, n_2, \cdots, n_{m-1}, n_m)\}
\end{array}
\right\} (3)
$$

ここでGTL′は、全体属性を、DD内の対応する目標リストALi :∃i∈{1, 2, ···,m}を用いて、局所属性の式におきかえたものである。GQL′も、同じく局所属性のみを参照するように変換された条件式である。全体問合せGの結果リレーションGRは、分割されたGRhの和(union)となる($1 \leq h \leq \prod_{j=i}^{m} i_j$)。

さらに、分割された問合せGhに、完全性(integrity)とプロテクション条件を加える。インテグリティは、個々の局所リレーション及び局所リレーション間に関して関係計算式として表わされる。この論理式を局所完全性条件式(local integrity constraint formula; LICF)と呼ぶ。LICFは、離接正規化されている。よって、問合せGhが参照する局所リレーションと、これらの間に関するLICFを、さらにGhの条件式に接合する。この時接合するLICFは、句の接合、即ち離接(OR)はふくまれないとする。このようなインテグリティの条件式をLICFhと記す。よってGhは、

$$
\left.
\begin{array}{l}
\quad \underline{\text{range of}}\ \ (l_1, \cdots, l_n)(L_1 : s_1, \cdots, L_n : s_n, \cdots, L_n : s_n)\ ; \\
G_h' : \underline{\text{retrieve into}}\ \ GR_h\ (GTL') \\
\quad \underline{\text{where}}\ \ \ GQL' \wedge GCli_1 \wedge \cdots \wedge GCmi_m \wedge LICF_h\ \ ;
\end{array}
\right\} (4)
$$

となる。同じく全体問合せの結果はGRhである。

問合せGhは、接合問合せであり、複数の局所リレーションを参照している。この問合せは、これらのリレーションが種々のサイトに存在していることを特徴としている。最終結果がG′hの結果の和(union)であることから、このプロセスを水平問合せ分割(HQD)と呼ぶ。

HQDされた問合せは、次のような形式に書き直せる。

$$
\begin{array}{l}
G : \ \underline{\text{range of}}\ (l_{11}, l_{12}, \cdots, l_{1t_1}, l_{21}, l_{22}, \cdots, l_{2t_2}, \cdots, , \\
\qquad\qquad l_{n1}, l_{n2}, \cdots, l_{nt_n}) \\
\qquad\qquad (L_{11} : 1, L_{12} : 1, \cdots, L_{1t_1} : 1, L_{21} : 2, L_{22} : 2, \cdots, \\
\qquad\qquad L_{2t_2} : 2, \cdots, L_{n1} : n, L_{n2} : n, \cdots, L_{nt_n} : n)\ ; \\
\quad \underline{\text{retrieve into}}\ \ \ RESULT\ (TL) \\
\quad \underline{\text{where}}\ \ \ \ QL\ \ ;
\end{array}
\qquad (5)
$$

ここで、l_{ij}は、サイトiのj番目の組変数である。t_iは、サイトi内の組変数の数である。QLは比較句の接合である。比較句には、1つのサイト内の組変数のみを参照しているサイト内句（one-site clause：OSC）と、サイト間にまたがっているサイト間句（two-site clause：TSC）とがある。$OSC_i(l_{i1}, \cdots, l_{it_i})$は、サイトiとjのサイト間句の接合である。よって

$$QL=\left[\bigvee_{i=1,\cdots,n} OSC_i(l_{i1}, \cdots, l_{it_i})\right]\left[\bigvee_{\substack{i=1,\cdots,n \\ j=i+1,\cdots,n}} TSC_{ij}(l_{i1},\cdots l_{it_i}, l_{j1}, \cdots, l_{jt_j})\right]$$

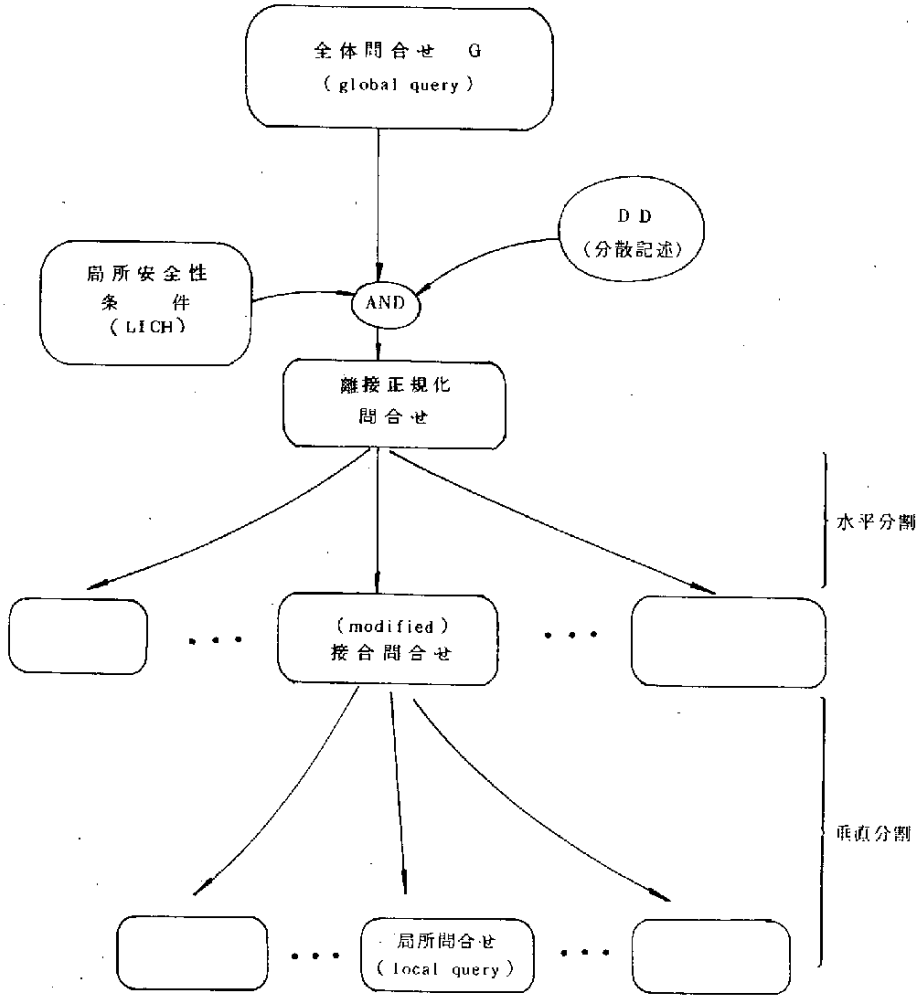


図2－22　GLQDの概要

HQDの例として、2.6.2における全体RSR　AUTHOR－REPに対する問合せを考えてみよう。これの定義式は、2.6.2の(4)式として与えられている。このAUTHOR－REPに対して、"TOMが1970年以前に書いたレポートの番号と名前をもとめよ"という問合せGは、次のように記述される。

```
    range of      (arp) (AUTHOR－REP) ;
G : retrieve into   RESULT(arp.aname, arp.rep#, arp.rep_name)
    where      arp.aname＝"TOM"  ∧  arp.year≦1970;
```

問合せGは、DD内の〈g-constraint〉に対応して、G_1とG_2の2つの問合せに水平分割される。

```
    range of    (a, b, d) (AUTHOR:3, BOOK:1, DOCUMENT:2);
G1 : retrieve into   GR1(aname＝a.name, rep#＝a.report#,
                                          rep_name＝b.name)
    where        a.name＝"TOM"                                  (6)
        ∧  a.report#＝b.book#  ∧  b.year≦1970;

G2 : retrieve into  GR2(aname＝a.name, rep#＝a.report#
                                          rep-name＝d.dname)
    where      a.name＝"TOM"  ∧                                 (7)
               a.report#＝d.d#  ∧  d.year≦1970;
```

ここで、AUTHOR、BOOK、DOCUMENTの3つの局所リレーションに対するインテグリティ条件は、次のようである。BOOKリレーションには、1900年以降出版された図書情報が格納されているが、DOCUMENTリレーションには、1975年以降のもののみである。これは、次のように記述される。

$$LICF = b.year \geq 1900 \ \wedge \ d.year \geq 1975 \tag{8}$$

これをG_1とG_2に付加すると、最終的に次の問合せを得る。

```
G1' : retrieve into  GR1(aname＝a.name, rep#＝a.report#
                                          rep_name＝b.name)
    where      a.name＝"TOM"∧ a.report#＝b.book#                (9)
        ∧ b.year≦1970  ∧ b.year≧1900 ;
```

$$
\left.
\begin{array}{l}
G_2' : \underline{\text{retrieve into}}\ \ GR_2(\text{aname}=\text{a.name},\ \text{rep\#}=\text{a.report\#},\\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{rep-name}=\text{d.dname})\\
\underline{\text{where}}\quad \text{a.name}=\text{“TOM”}\ \wedge\ \text{a.report\#}=\text{d.d\#}\\
\qquad \wedge\ \text{d.year}\leq 1970\ \wedge\ \text{d.year}\geq 1975\quad ;
\end{array}
\right\} \quad (10)
$$

問合せG_2'(10)の条件部は、常に偽(false)となるために、G_1'のみが生成される。

ここで、OSC_iを、サイトiに関するサイト内句とする($i \in \{1,2,3\}$)。TSC_{ij}を、サイトiとjとのサイト間句($i \in \{1,2,3\}$ and $j \neq i$)とする。HQDされた問合せのOSCとTSCは次のようになる。

$$
\left.
\begin{array}{l}
OSC_3(a) = \left[\text{a.name}=\text{“TOM”}\right]\\
OSC_1(b) = \left[\text{b.year}\leq 1970\ \wedge\ \text{b.year}\geq 1900\right]\\
TSC_{13}(a,b) = \left[\text{a.report\#}=\text{b.book\#}\right]
\end{array}
\right\} \quad (11)
$$

OSC_1とOSC_3は、各々サイト1とサイト3において独立に処理が可能である。次に、このような各サイト内で独立して行なえる問合せ処理、局所問合せ処理を論じる。局所問合せ処理の終了後、サイト間句、例えばTSC_{13}の処理を行なうために、問合せの垂直分割を行なう。

B. 局所問合せ処理(local query processing ; LQP)

問合せの条件式のうち、1つのサイトのみを参照している句(OSC)に関する局所処理をそのサイトで行なわせる。よって、サイトiには、$OSC_i(l_{i1},\cdots\cdots,l_{it_1})$を条件部としてもつ問合せが出される。局所問合せ処理を垂直分割する前に行なう理由は、通信コストに対して、各サイトでの処理コストは極めて小さいと仮定しているからである。ここで、次のことを考える必要がある。OSC_iから生成できる問合せ図が連結(connect)されているかどうかである。連結されているならば、このサイトには1つの問合せが出される。連結されていなければ、各々の非連結な部分図に対応して問合せを出す必要がある。この時、OSC_iは図に対応してm_iコに分割される。よって、一般に各サイトiには、次の問合せが並行して出される。

$$
\left\{
\begin{array}{l|l}
\underline{\text{range of}}\quad (l_{ij}^{1},\cdots,l_{ij}^{k})(L_{ij}^{1},\cdots,L_{ij}^{k})\ ; & \\
\underline{\text{retrieve into}}\quad R_{ij}(TL_{ij}) & j=1,2,\ldots,m_i\\
\underline{\text{where}}\qquad OSC_{ij}\quad ; &
\end{array}
\right\} \quad (12)
$$

ここでl_{ij}^{h}($h=1,\cdots,k$)は、OSC_{ij}及びTL_{ij}内で参照される組変数である。TL_{ij}内の属性には、残りのTSC_{ij}内とTL内で必要なものが含まれる。この問合せは、各サイトで処理され、結果としてリレーションR_{ij}が生成される。この局所問合せ処理によって生成された

リレーションに基づいて、サイト間にまたがった問合せの分割を行なう。

例として、問合せG_1'〔(9)〕を考えてみよう。G_1'内のOSCとTSCは、(11)に与えられている。図2－23は、G_1'の問合せ図を表わしている。

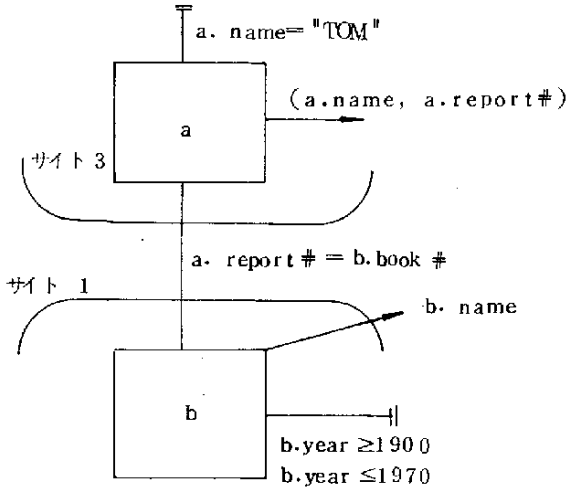


図2－23　G_1'に対応する問合せ図

このOSCより生成される局所問合せは次のようである。

G_{13}':　range of　(a) (A:3) ;
　　retrieve into　R_3(aname = a.name, rep#=a.report#)
　　where　a.name = "TOM" ;　(13)

G_{11}':　range of　(b) (BOOK:1) ;
　　retrieve into　R_1(rep# = b.book#, rep_name=b.name)
　　where　b.year ≥ 1900 ∧ b.year ≤ 1970 ;　(14)

G_{13}'とG_{11}'の局所処理が終了すると、サイト3にはリレーションR_3(aname, rep#)と、サイト1にはリレーションR_1(rep#, rep-name)が生成される。

残されたTSCに関する問合せは、次のようになる。

range of　(r_1, r_3)(R_1:1, R_3:3) ;
retrieve into　GR_1(r_1.aname, r_1.rep#, r_3.rep_name)
where　r_1.rep# = r_2.rep# ;　(15)

次に、このようにして生成されたＴＳＣのみを持つ問合せの分割について論じる。

C．水平分割における目的関数

問合せの垂直分割では、水平分割と異なり、分割された問合せの実行順序が問題となる。この実行順序の決定は、次の目的を達するようになされる必要がある〔ＨＥＶＮ７８〕。

(i) 各サイトでの問合せ処理の並行度を最大にする。

(ii) 各サイト間でのデータのやりとりを最少にする。

(i)は、応答時間を最少にすることである。(ii)は、全処理時間を最少にすることである。各サイトでの問合せ処理の並行度を高めると、一方では、サイト間の通信量の増大をおこす。よって、問合せの分散処理には、この２点のトレードオフが存在している。

我々の基本方法は次のようである。

(i) より小さいリレーションを、より大きいリレーションの存在するサイトにネットワークを用いて転送する。

(ii) 複数のサイトへ転送する場合は、並行しておこなう。

(iii) 必要な値集合のみ（即ち、属性単位）を転送する。

(iv) 組単位（tuple-at-a-time）では転送しない。

ネットワークに対しては、次の仮定がなりたつとする。

(i) 距離に独立な通信コスト（distant-independent）。即ち、$C_{ij}(x)$を サイトｉとサイトｊの間で、大きさｘのデータを転送する通信コストとすると、$C_{ij}(x)=C(x)= a+b\cdot x$である。ａとｂは、ある定数である。

(ii) サイト―サイト（site-to-site）ネットワーク。即ち、$C_{i\to\{j_1,j_2,\cdots,j_n\}}(x)$を、サイトｉからｎコのサイト$j_1,j_2,\cdots\cdots,j_n$に、大きさｘのデータを転送する通信コストとすると、$C_{i\to\{j_1,j_2,\cdots,j_n\}}(x)=n\cdot C(x)$である。ちなみに、放送（broadcast）ネットワークでは、$C_{i\to\{j_1,j_2,\cdots,j_n\}}(x)=C(x)$である。

(iii) 低負荷ネットワーク。ネットワーク内のトラフィック量は、各ノードで待行列が生成されない程度であるとする。

D．垂直分割（ＶＱＤ）

局所問合せ処理後の問合せは、サイト間句（ＴＳＣ）の接合より成り立っている。よって、この問合せを処理するためには、サイト間でデータを転送する必要がある。ここで論じる問合せの垂直分割（vertical query decomposition）とは、複数サイトにまたがった問合せを、２つのサイト（即ち、２つの組変数）を参照する問合せに分割し、その実行順序を定めることである。局所問合せ処理後の問合せG'_hは、次の形式を持つ。

$$\underline{\text{range of}}\ (r11,r12,\cdots,r1m_1,\ r21,r22,\cdots,r2m_2,\cdots,rnm_n)$$
$$(R_{11}:1,R_{12}:1,\cdots,R_{1m_1}:1,$$
$$R_{21}:2,R_{22}:2,\cdots,R_{2m_2}:2,\cdots,R_{nm_n}:n)\quad ;$$

$$
\begin{aligned}
G_h' : \ & \underline{\text{retrieve into}} \quad GR_h(TL') \\
& \underline{\text{where}} \quad TSC_{12}(r11, r12, \cdots, r1m_1, r21, r22, \cdots, r2m_2) \wedge \\
& \qquad TSC_{23}(r21, \cdots, r2m_2, r31, \cdots, r3m_3) \wedge \\
& \qquad \cdots\cdots \\
& \qquad \wedge TSC_{n-1,\,n}(rn-1,\,1, \cdots, rn-1m_{m-1}, rn1, \cdots, rnm_n);
\end{aligned}
\tag{16}
$$

ここで、R_{ij}は、サイトiのj番目のリレーション名である（$i=1,2,\cdots,n$, $j=1,2,\cdots,m_i$）。R_{ij}は、局所問合せ処理によって生成されたリレーションである。r_{ij}は、リレーションR_{ij}の組変数である。TL′は、目標リストである。TSC_{kl}は、サイトkとサイトlとを参照するサイト間句の接合である（$k=1,2,\cdots,n-1$, $l=k+1,k+2,\cdots,n$）。

次に、問合せG_h'の分割を行なう。問合せは、接合問合せであるので、LLQD（2.5.1）で論じたように、問合せ図によって表わされる。各TSC_{kl}は、サイトkとサイトl内のノード間の結合句を表わしている。この問合せ図は、サイト間句に対応する結合リンクと、目標リストに対応する結果リンクのみを持つ。問合せの分割は、図の切断ノードを用いて、部分図に分割することによってなされる。部分図内にループが存在する場合は、2つ以上のノードを包んでいる。この時は、さらに2つのノードのみをもつ部分図に分割せねばならない。

問合せの実行順序は、応答時間を最少にするとともに、全通信コストを最少にするように決定される。先に述べたように、我々の問合せ処理は、必要な属性の値の集合を、より大きなリレーションに、なるべく並行して転送しようとするものである。このために、各リレーションについて、次の情報を保持する必要がある。

(i) リレーションの属性数と属性構成

(ii) リレーションの組数（cardinality）

(iii) 属性のアクセスパス

(iv) 属性のとり得る値の数

(v) このリレーション内に実現されている異なった値（unique value）の数

これらの情報は、局所問合せ処理の応答として、問合せ処理のスケジューリングを行なうGDP（我々は統合GDPと呼ぶ）に転送される。この情報は、HIと同様な形式〔2.5.2を参照〕で問合せ処理の間管理される。

2.7 システムアーキテクチャ

我々の分散型データベース（DDB：distributed database）は、コンピュータ・ネットワークにより結合される幾つかのデータベースサイトから構成される。各サイトは1つ以上のDBMSで管理される幾つかのデータベースを持っている。DDBの管理上、個々のサイトには2種類のプロセッサが必要である。即ち、GDP（Global Data Processor：全体データプロセッサ）と呼ばれるものとLDP（Local Data Processor：局所データプロセッサ）と呼ばれるものである。各LDPは、1つのデータベースに対応している。

NDD（Network Data Directory）は、DI（分散情報）及びHI（異種性情報）を管理しており、これらの情報は通常のデータベースと同様に取り扱われる。さらに、DI及びHI情報はリレーショナルな形式になっている。GDPは必要な情報（DI）を、即ち必要なデータが何処にあるかを、そのデータを管理しているLDPの1つを通して獲得する。また、LDPは共通—固有（common-local）変換情報（HI）をそのサイトでHIを管理しているLDPから獲得する。実際、NDDはDBMSで管理されるデータベースもしくはその一部である。

GDPは分散問題や分散機能に関して次のような働きをする。

1) GDPはローカルサイトでのGCSもしくはEXSに基づいたユーザの要求を受け、それをリレーショナル形式の全体問合せ（relational global query）に変換する。
2) 次に、GDPはDIを用いて必要なデータの所在を決定し、幾つかのサイトで同時に実行できるように半順序付けされた（partially ordered）一連の局所問合せに分割する。
3) さらに、これらの局所問合せをネットワーク経由で対応するLDPへ発信する。
4) また、GDPは局所問合せの実行について同期をとるためにLDPを統括する。
5) そして、LDPから返された実行結果をユーザの要求に合うように合成する。

個々のLDPは局所DBMSの異種性に係わっている。また共通形式と局所形式の間の記述と言語の変換器として機能する。全体問合せが発信されたGDP（このGDPをcoordinate GDP：統合GDPと呼ぶ）の管理下で、各LDPは次のような働きをする。

1) LDPはGDPからリレーショナルな形式で局所問合せを受けとる。
2) 局所問合せは、HIを用いて一連の局所DBMS　DMLに変換される。
3) 次にLDPは局所DBMSを起動する。
4) そして局所DBMS　DMLの結果を局所問合せに合うように合成する。
5) 合成した結果はネットワーク経由で統合GDPに戻される。
6) 他のLDPに実行結果を転送したり、局所障害からの復帰のための機能も有している。

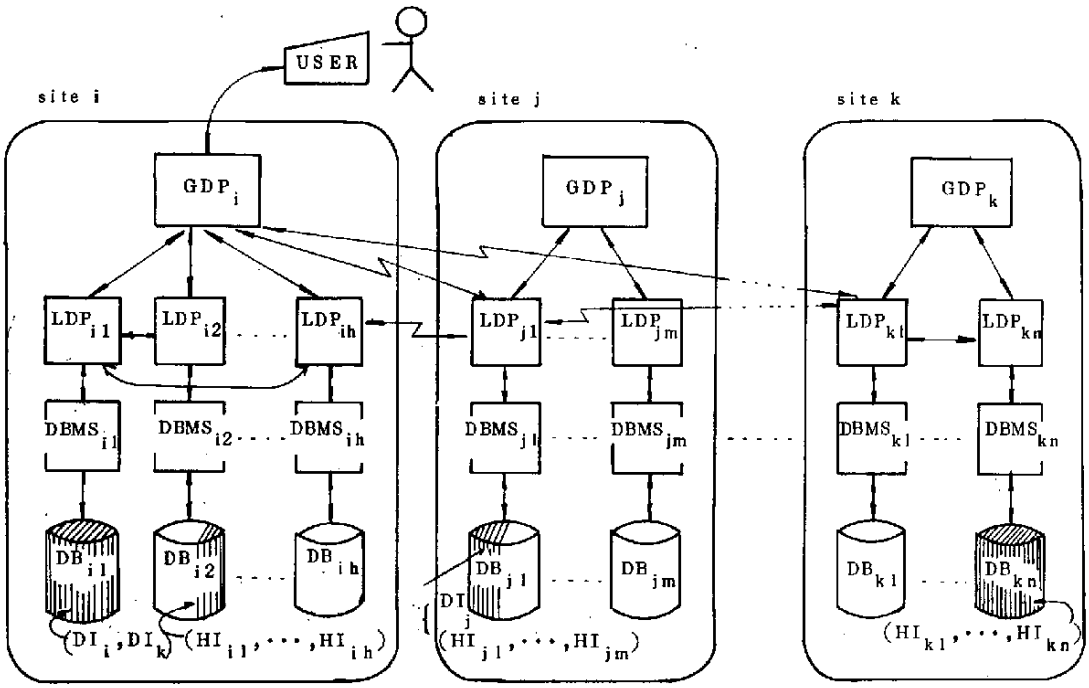


図２－２４　ＤＤＢのシステムアーキテクチャ

2.8 まとめ

本章では、異種のＤＢＭＳを統合し、コンピュータネットワークを介した共同システム（cooperative system）――分散型データベース――構築の可能性を示した。そして、汎用－統合型（general-purpose and bottom-up）の分散型データベースシステムの全体アーキテクチャとして４層スキーマ構造（ＦＳＳ）を提案した。我々の目的は、

1) Ｅ－Ｒモデルを用いて個々のスキーマの構文構造を記述することにより異種のＤＢＭＳを同種化することと、

2) 分散されたデータベースを１つのデータベースに統合することである。統合の際ユーザは分散と異種に関して不可視（invisible）の状態でアクセスできる。これは、特定のネットワーク共同体にとって意味のある統合データベースのセマンティックを記述することにより実現される。

異種性問題としては、ＦＳＳ概念に基づいて、ＬＩＳからＬＣＳへの記述過程を示した。さらに局所共通問合せから一連の局所ＤＢＭＳ　ＤＭＬへの変換過程を示した。特にＤＢＴＧモデルを採り上げ具体的に示した。分散問題としては、全体―局所問合せ分割（GLQD：global－local-query decomposition）及びＤＩ（分散情報）について示した。今後は、同期管理、障害回復、ＤＢＭＳの起動等に関する制御機能に関連する諸問題も検討・解決する必要がある。問合せ処理については、実験的に最適化を検討していくつもりである。

3　ネットワークデータディレクトリ

第2章では、4層スキーマ構造に基づく我々の分散型データベース構想を示した。本章では、この分散型データベースシステムをサポートするＮＤＤ（Network Data Directory）について述べる。ＮＤＤが管理する主な情報は、ＤＩ（分散情報）及びＨＩ（異種性情報）である。これらの情報は通常のデータベースもしくはその一部として取り扱われる。また形式は原則として正規化されたリレーションとする。

ＮＤＤが管理する情報を生成するシステムを“クリアリング情報生成システム”とし、ＨＩを生成するシステムを“異種性情報生成サブシステム”、ＤＩを生成するシステムを“分散情報生成サブシステム”という。以下、3.1では異種性情報生成サブシステムについて、3.2では分散情報生成サブシステムについて述べる。

3.1　クリアリング情報生成システム／異種性情報生成サブシステム（ＮＤＤ／ＨＩＰＳ）

分散型データベースシステムにおけるクリアリング情報生成システムの一環として作成する異種性情報生成サブシステムの名称をＮＤＤ／ＨＩＰＳ（Network Data Directory／Heterogeneity Information Processing System ――以下本サブシステムという）とする。

図３－１に４層スキーマ構造における位置付けを示す。

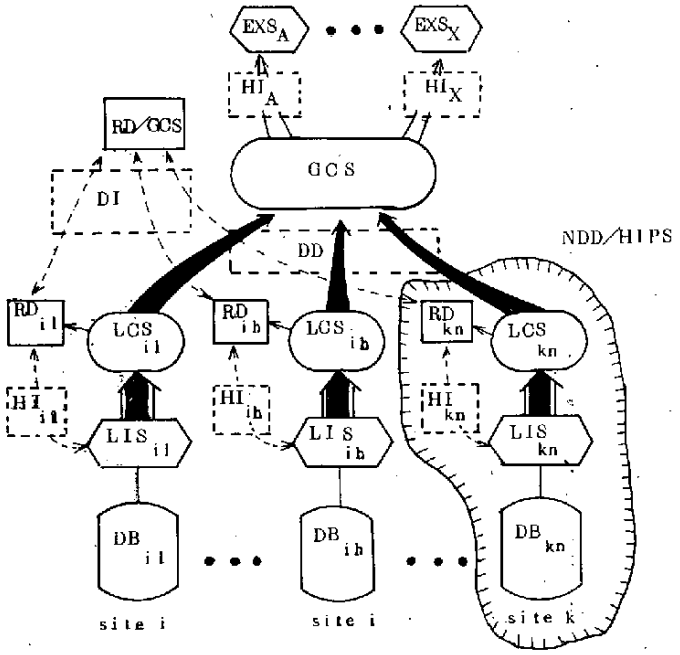


図３－１　４層スキーマ構造とＮＤＤ／ＨＩＰＳ

3.1.1 処理概要

本サブシステムは、ＬＣＳ／ＨＩ（異種性情報）に関する記述を解読し５種類１０個のＨＩ情報のリレーション（ファイル）を生成することと、このＨＩ情報からＬＣＳのリレーショナル記述（ＲＤ）を生成するものである。

図３－２にＮＤＤ／ＨＩＰＳの入出力過程を示す。入力するＬＣＳ記述は、各ＤＢＭＳ固有のスキーマ記述をＥ－Ｒモデルに基づいて記述したものである。

図３－３は、スキーマ構造と入出力過程を対応させることにより異種性情報が除去される過程を示している。

図３－４は、ＮＤＤ／ＨＩＰＳがＬＣＳ／ＨＩ記述を解読するために持っている４つの機能モジュールを中心に記述したものである。

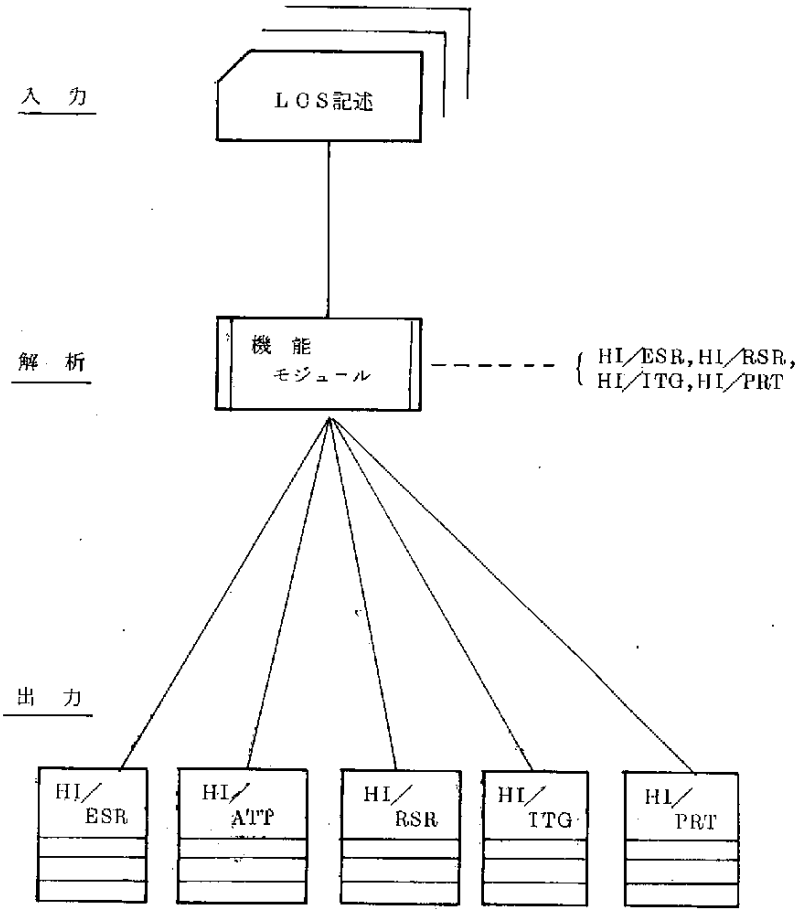


図３－２　ＮＤＤ／ＨＩＰＳ入出力過程

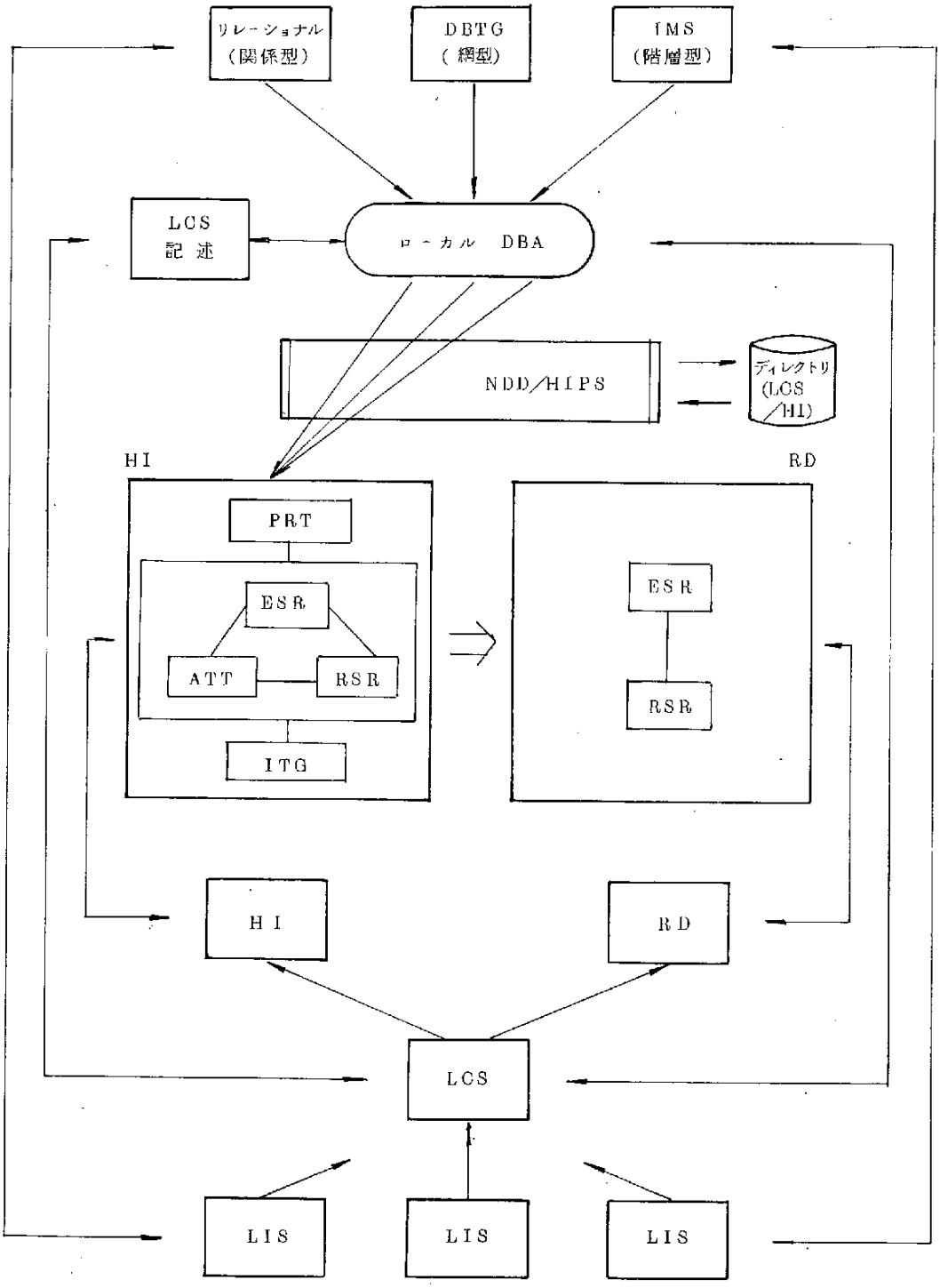


図３－３　スキーマ構造と入出力過程

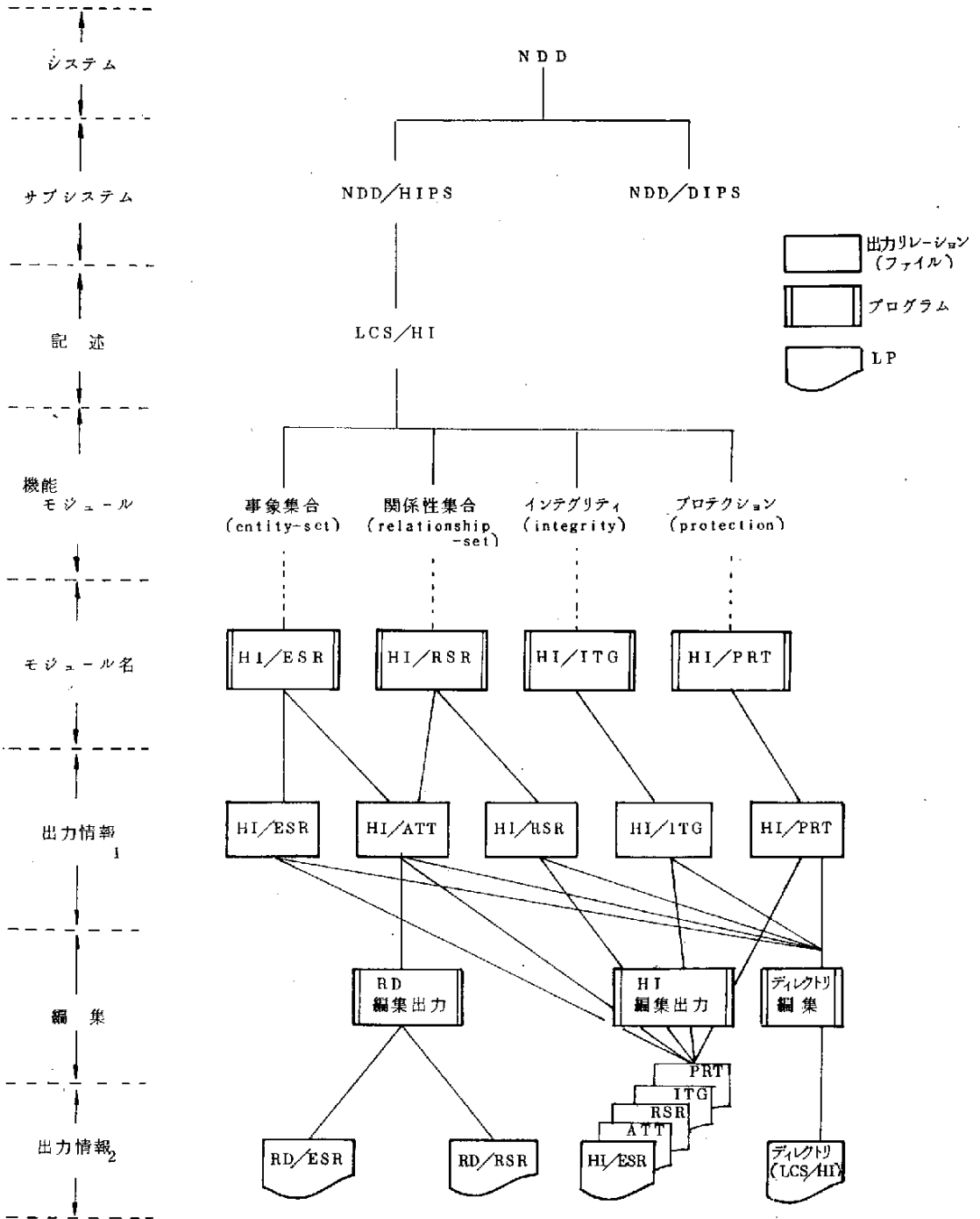


図3－4　NDD／HIPSモジュール関連図

3.1.2 ＬＣＳ／ＨＪ記述

ＬＣＳ／ＨＩ記述は次のような体系になっている。またこれを図示すると図３－５のようになる。

※ ＬＣＳ／ＨＩ記述の骨格

- １つ以上のＤＢエントリの記述

※※ ＤＢエントリ記述の骨格

- １つのＤＢステートメント（＃＃ＤＢ）の記述
- １つ以上のモジュールエントリの記述
- １つのＤＢエンドステートメント（＃＃ＥＮＤ）の記述

※※※ モジュールエントリ記述の骨格

- １つ以上の“事象集合／関係性集合に関するステートメント”の記述

表３－１は、各モジュールエントリを構成するステートメントの種類（番号）毎の許される最小、最大数を示したものである。

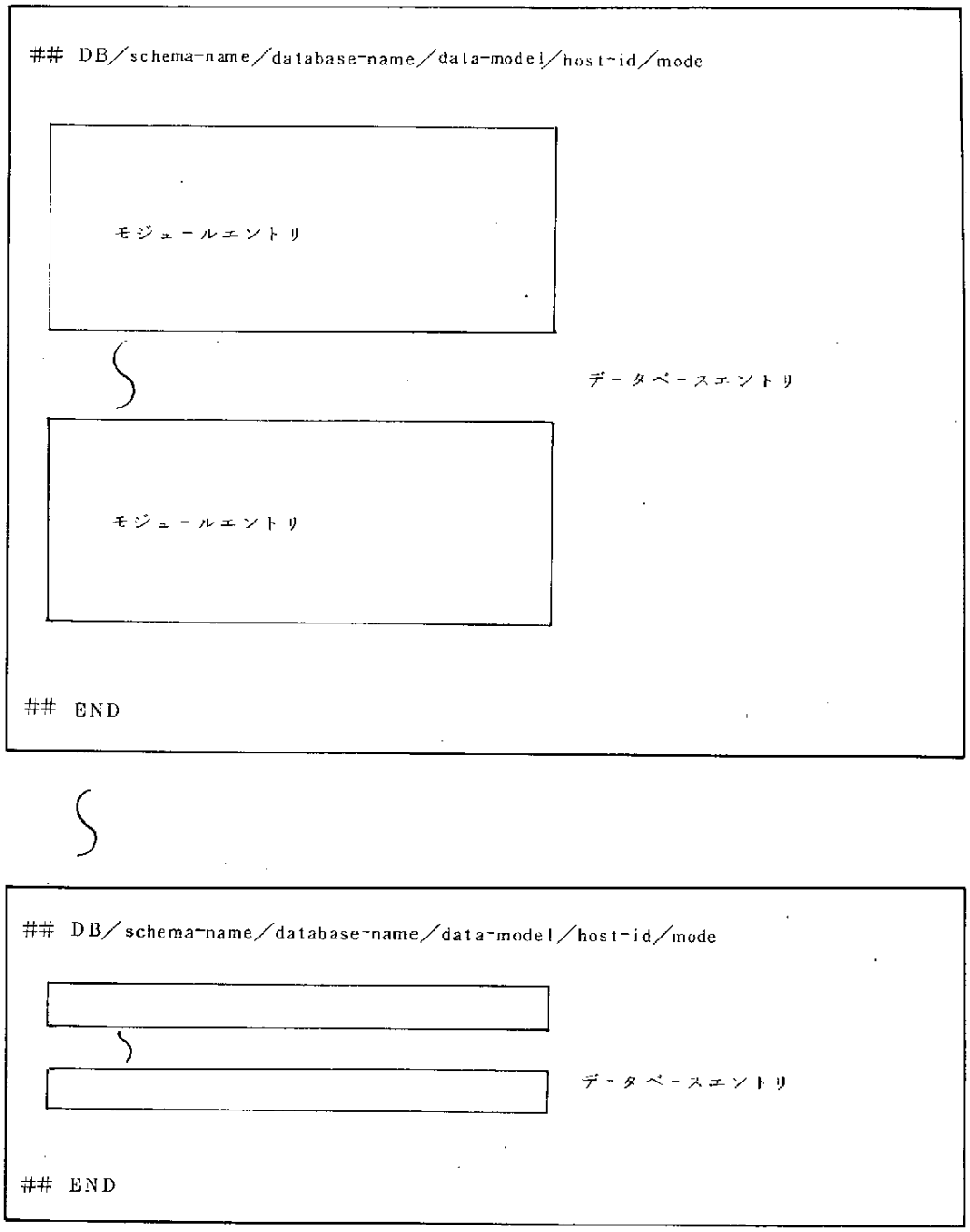


図３－５　ＬＣＳ／ＨＩ記述：構成例

表３－１　エントリと記述ステートメント数の関係

エントリ名	ステートメント	最大	最大	備　　考
データベース	##DB	1	N	－
	##END	0	N	
ESR	101	1	1	同一エンタティセット、もしくはリレーションシップセット内での条件である。
	201～299	1	99	
RSR	101	1	1	
	201	0	1	
	301～399	1	99	
	401～499	1	99	
	501～599	0	99	
ITG	101	0	1	
	201～299	0	99	
	901～999	0	99	
PRT	101～199	0	99	
	201	0	1	
	301～399	0	99	
	901～999	0	99	

さらに、各モジュールエントリのステートメント番号と出力ファイル（リレーション）との対応関係を図３－６に示す（→－；1:1に対応，→→；1：nに対応）。

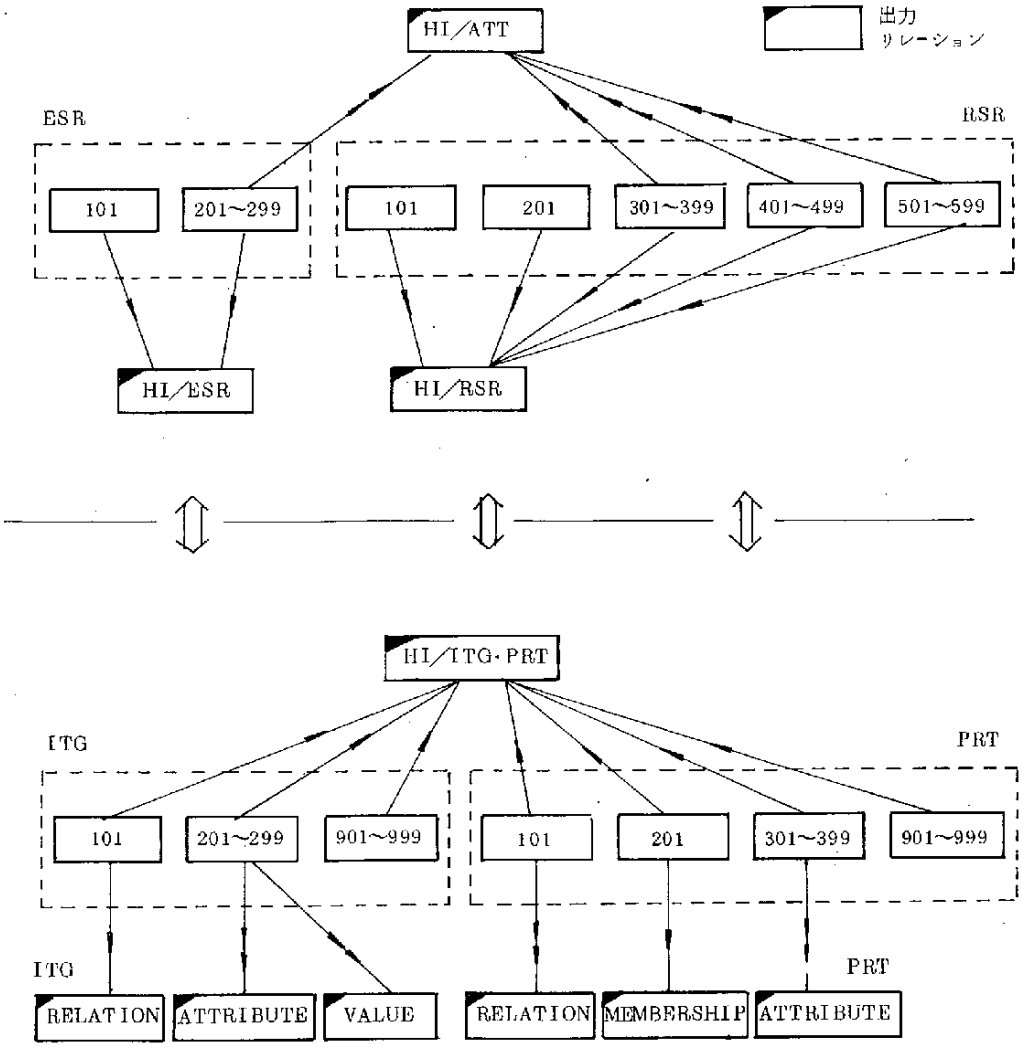


図３－６　ステートメント番号と出力ファイル（リレーション）の関係

1）データベースエントリ（DB entry）

DB entryは、NDD／HIPSで処理する単位の1つで、スキーマ（schema）毎に記述される。

一般形式：

##DB／schema-name／database-name／data-model／host-id／mode

・1つ以上のモジュールエントリの記述

##END

解説：

##DB ステートメントは本サブシステムで処理するスキーマ名（サブスキーマ名）、データベース名、データモデルの種類（DBTG、IMS、RELATIONAL）、ホスト名、処理モード（C：創成、U：更新、A：追加、D：削除）を記述する。

2）モジュールエントリ（module entry）

モジュールエントリは、DB entryの中で記述され、LCSのHI記述を具体的に行なうものである。

一般形式：

module-name mode { entity-set / relationship-set } detail-description

解説：

・module-name

HI／ESR、HI／RSR、HI／ITG、HI／PRTの何れかを指定する。

・mode

処理モード（C、U、A、D）を指定する。

・{ entity-set / relationship-set }

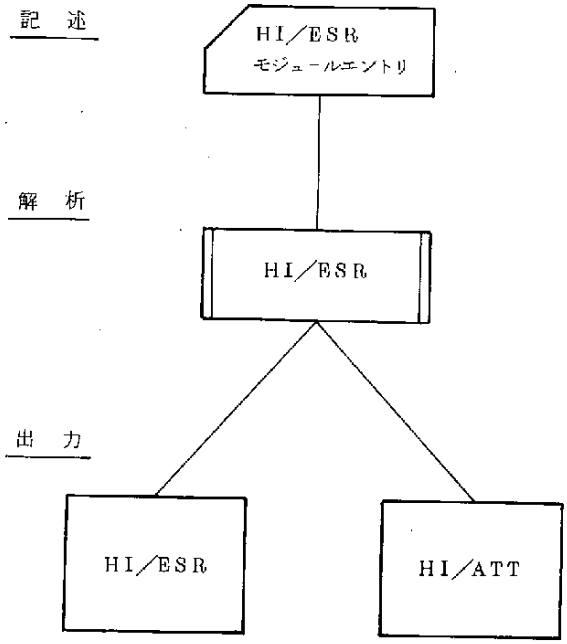
処理対象の事象集合名、関係性集合名を指定する。

・detail-description

LCS／HIに関する詳細記述を行なう。

a．ＨＩ／ＥＳＲモジュールエントリ

ＨＩ／ＥＳＲモジュールエントリでは、ＬＣＳレベルでの事象集合に関する記述を行なう。解析の結果は、ＨＩ／ＥＳＲおよびＨＩ／ＡＴＴに出力される。なお、ＨＩ／ＡＴＴには、ＨＩ／ＲＳＲモジュールエントリの結果も合わせて出力される

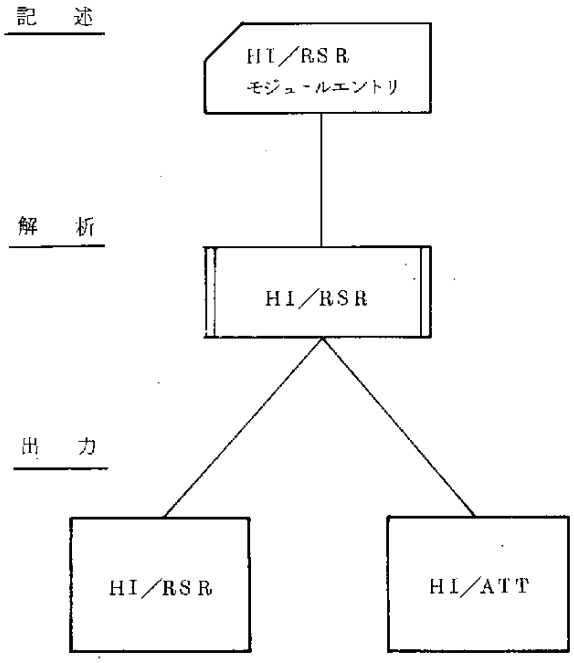



```
*
*              ************************
*              *  HI/ESR  SAVE AREA   *
*              ************************
*
 01  HI-ESR-SAVE-AREA.
     03  HI-ESR-101.
         05    HES-101-PROCESS-MODE        PIC  X.
         05    HES-101-ENTITY              PIC  X(16).
         05    HES-101-AREA-ROOT           PIC  X(16).
         05    HES-101-MODE                PIC  X(02).
         05    HES-101-PROTECTION          PIC  X(02).
         05    HES-101-INTEGRITY           PIC  X(02).
         05    HES-101-CARDINALITY         PIC  9(04).
*
*
     03  HI-ESR-201                  OCCURS  99 TIMES.
         05    HES-201-PROCESS-MODE        PIC  X.
         05    HES-201-ATTRIBUTE           PIC  X(16).
         05    HES-201-TYPE                PIC  X.
         05    HES-201-LENGTH              PIC  9(03).
         05    HES-201-KEY-TYPE            PIC  X(05).
         05    HES-201-PROTECTION          PIC  X(02).
         05    HES-201-INTEGRITY           PIC  X(02).
         05    HES-201-CARDINALITY         PIC  9(04).
         05    HES-201-VALUE-SET           PIC  X(12).
```

図３－７　ＨＩ／ＥＳＲの記述形式

b．ＨＩ／ＲＳＲモジュールエントリ

ＨＩ／ＲＳＲモジュールエントリでは、ＬＣＳレベルでの関係性集合に関する記述を行なう。解析の結果は、ＨＩ／ＲＳＲおよびＨＩ／ＡＴＴに出力される。



```
*
*            ************************
*            *  HI/RSR  SAVE AREA   *
*            ************************
*
 01  HI-RSR-SAVE-AREA
                                        REDEFINES   HI-ESR-SAVE-AREA.
     03  HI-RSR-101.
         05  HRS-101-PROCESS-MODE          PIC  X.
         05  HRS-101-RELATIONSHIP          PIC  X(16).
         05  HRS-101-SOURCE-ES             PIC  X(16).
         05  HRS-101-DEST-ES               PIC  X(16).
         05  HRS-101-MODE                  PIC  X(02).
         05  HRS-101-RS-TYPE               PIC  X(03).
         05  HRS-101-PROTECTION            PIC  X(02).
         05  HRS-101-INTEGRITY             PIC  X(02).
         05  HRS-101-CARDINALITY           PIC  9(04).
*
*
     03  HI-RSR-201.
         05  HRS-201-PROCESS-MODE          PIC  X.
         05  HRS-201-SOURCE-SET            PIC  X(16).
         05  HRS-201-DEST-SET              PIC  X(16).
*
*
     03  HI-RSR-301                  OCCURS  99 TIMES.
         05  HRS-301-ATTRIBUTE             PIC  X(16).
         05  HRS-301-TYPE                  PIC  X.
         05  HRS-301-LENGTH                PIC  9(03).
         05  HRS-301-KEY-TYPE              PIC  X(05).
         05  HRS-301-PROTECTION            PIC  X(02).
         05  HRS-301-INTEGRITY             PIC  X(02).
         05  HRS-301-CARDINALITY           PIC  9(04).
         05  HRS-301-VALUE-SET             PIC  X(12).
         05  HRS-301-DEFAULT               PIC  9.
*
     03  HI-RSR-401                  OCCURS  99 TIMES.
         05  HRS-401-ATTRIBUTE             PIC  X(16).
         05  HRS-401-TYPE                  PIC  X.
         05  HRS-401-LENGTH                PIC  9(03).
         05  HRS-401-KEY-TYPE              PIC  X(05).
         05  HRS-401-PROTECTION            PIC  X(02).
         05  HRS-401-INTEGRITY             PIC  X(02).
         05  HRS-401-CARDINALITY           PIC  9(04).
         05  HRS-401-VALUE-SET             PIC  X(12).
         05  HRS-401-DEFAULT               PIC  9.
*
*
     03  HI-RSR-501                  OCCURS  99 TIMES.
         05  HRS-501-ATTRIBUTE             PIC  X(16).
         05  HRS-501-TYPE                  PIC  X.
         05  HRS-501-LENGTH                PIC  9(03).
         05  HRS-501-KEY-TYPE              PIC  X(05).
         05  HRS-501-PROTECTION            PIC  X(02).
         05  HRS-501-INTEGRITY             PIC  X(02).
         05  HRS-501-CARDINALITY           PIC  9(04).
         05  HRS-501-VALUE-SET             PIC  X(12).
```

図３－８　ＨＩ／ＲＳＲの記述形式

c．ＨＩ／ＩＴＧとＨＩ／ＰＲＴ

① ＨＩ／ＩＴＧモジュールエントリ

ＨＩ／ＩＴＧモジュールエントリでは、ＬＣＳレベルでのインテグリティ（integrity）に関する記述を行なう。インテグリティ記述は、ＨＩ／ＥＳＲおよびＨＩ／ＲＳＲモジュールエントリでの記述に対応している。なお、ここでのインテグリティ記述は、データベースのスキーマで明確に定義されているものに限る。

スキーマ上で定義されていないインテグリティの記述は、ステートメント番号９０１以降で具体的に記述するようにする。例えば、項目間に何らかのインテグリティを設定する場合が考えられる。

```
*
*          ************************
*          *  HI/ITG  SAVE AREA   *
*          ************************
*
 01  HI-ITG-SAVE-AREA
                                    REDEFINES   HI-ESR-SAVE-AREA.
     03  HI-ITG-101.
         05  HIT-101-PROCESS-MODE        PIC  X.
         05  HIT-101-ES-RS-NAME          PIC  X(16).
         05  HIT-101-ES-RS-TYPE          PIC  X(03).
         05  HIT-101-KEY-TYPE            PIC  X.
         05  HIT-101-CARD-ES-RS          PIC  9(04).
         05  HIT-101-CARD-TUPLE          PIC  9(04).
*
*
     03  HI-ITG-201                  OCCURS  99 TIMES.
         05  HIT-201-PROCESS-MODE        PIC  X.
         05  HIT-201-ES-RS-NAME          PIC  X(16).
         05  HIT-201-ES-RS-TYPE          PIC  X(03).
         05  HIT-201-ATTRIBUTE           PIC  X(16).
         05  HIT-201-CARD-ATT            PIC  9(04).
         05  HIT-201-VALUE-TYPE          PIC  X.
         05  HIT-201-VALUE-TBL       OCCURS   3 TIMES.
             07  HIT-201-VALUE-KEY       PIC  9(03).
             07  HIT-201-VALUE           PIC  9(07).
*
*
     03  HI-ITG-901                  OCCURS  99 TIMES.
         05  HIT-901-PROCESS-MODE        PIC  X.
         05  HIT-901-ES-RS-NAME          PIC  X(16).
         05  HIT-901-ES-RS-TYPE          PIC  X(03).
         05  HIT-901-DETAIL              PIC  X(50).
```

図３－９　ＨＩ／ＩＴＧの記述形式

② ＨＩ／ＰＲＴモジュールエントリ

ＨＩ／ＰＲＴモジュールエントリでは、ＬＣＳレベルでのプロテクション（protection）に関する記述を行なう。プロテクション記述は、ＨＩ／ＥＳＲおよびＨＩ／ＲＳＲモジュールエントリでの記述に対応している。なお、ここでのプロテクション記述は、データベースのスキーマで明確に定義されているものに限る。

スキーマ上で定義されていないプロテクションの記述は、ステートメント番号９０１以降で具体的に記述するようにする。

プロテクションの記述に際しては、表３－２に示すプロテクションテーブルに基づいて行なう。

表３－２　プロテクションテーブル

	DBTG（網型）	IMS（階層型）	Relational（関係型）
事象集合（entity-set）	DE DELete　INSert FINd　MODify GET　REMove STOre	IE GET INSert DELete REPlace	RE RETrieve APPend DELete REPlace
関係性集合（relationship-set）	DR FINd INSert REMove ORDer	IR GET INSert DELete REPlace	RR RETrieve APPend DELete REPlace
子特性（membership）	DM storage-class / romoval-class AUTOMATIC / FIXED MANUAL / MANDATORY OPTIONAL		
属性（attribute）	DA GET MODify STOre	IA GET INSert DELete REPlace	RA RETrieve APPend DELete REPlace

※　プロテクションテーブル内の大文字部分のみが記述される。

```
*
*            ************************
*            *  HI-PRT  SAVE AREA   *
*            ************************
*
 01  HI-PRT-SAVE-AREA                    REDEFINES   HI-ESR-SAVE-AREA.
     03  HI-PRT-101                      OCCURS  99 TIMES.
         05  HPR-101-PROCESS-MODE            PIC  X.
         05  HPR-101-ES-RS-NAME              PIC  X(16).
         05  HPR-101-ES-RS-TYPE              PIC  X(03).
         05  HPR-101-MODE-KEY            OCCURS   3 TIMES.
             07  HPR-101-MODE                PIC  X(03).
             07  HPR-101-KEY                 PIC  X(10).
*
*
     03  HI-PRT-201.
         05  HPR-201-PROCESS-MODE            PIC  X.
         05  HPR-201-ES-RS-NAME              PIC  X(16).
         05  HPR-201-ES-RS-TYPE              PIC  X(03).
         05  HPR-201-STORAGE-CLASS           PIC  X(09).
*        05  HPR-201-REMOVAL-CLASS           PIC  X(09).
*
     03  HI-PRT-301                      OCCURS  99 TIMES.
         05  HPR-301-PROCESS-MODE            PIC  X.
         05  HPR-301-ES-RS-NAME              PIC  X(16).
         05  HPR-301-ES-RS-TYPE              PIC  X(03).
         05  HPR-301-ATTRIBUTE               PIC  X(16).
         05  HPR-301-MODE-KEY            OCCURS   2 TIMES.
             07  HPR-301-MODE                PIC  X(03).
             07  HPR-301-KEY                 PIC  X(10).
*
*
     03  HI-PRT-901                      OCCURS  99 TIMES.
         05  HPR-901-PROCESS-MODE            PIC  X.
         05  HPR-901-ES-RS-NAME              PIC  X(16).
         05  HPR-901-ES-RS-TYPE              PIC  X(03).
         05  HPR-901-DETAIL                  PIC  X(50).
```

図３－１０　ＨＩ／ＰＲＴの記述形式

```
*
*            ***************************
*            *  HI/ITG.PRT  SAVE AREA  *
*            ***************************
*
 01  HI-ITG-PRT-SAVE-AREA.
     03  HI-ITG-ALL                      OCCURS 200 TIMES.
         05  ITG-ALL-ES-RS-NAME              PIC  X(16).
         05  ITG-ALL-ES-RS-TYPE              PIC  X(02).
         05  ITG-ALL-ITG-LEVEL.
             07  ITG-ALL-ITG-L1              PIC  X.
             07  ITG-ALL-ITG-L2              PIC  X.
             07  ITG-ALL-ITG-L3              PIC  X.
         05  ITG-ALL-ITG-DL                  PIC  X(04).
*
     03  HI-PRT-ALL                      OCCURS 200 TIMES.
         05  PRT-ALL-ES-RS-NAME              PIC  X(16).
         05  PRT-ALL-ES-RS-TYPE              PIC  X(02).
         05  PRT-ALL-PRT-LEVEL.
             07  PRT-ALL-PRT-L1              PIC  X.
             07  PRT-ALL-PRT-L2              PIC  X.
             07  PRT-ALL-PRT-L3              PIC  X.
         05  PRT-ALL-PRT-DL                  PIC  X(04).
```

図３－１１　ＨＩ／ＩＴＧとＨＩ／ＰＲＴのための作業用リレーション形式

3）出力リレーション（output relation）

各モジュールで生成するリレーションのフォーマットを以下に示す。モジュールと出力リレーションとの対応は図３－６に示すとうりである。特に、ＨＩ／ＩＴＧとＨＩ／ＰＲＴから出力される７個のリレーションは、ＨＩ／ＩＴＧ・ＰＲＴを中心に管理される（図３－１３）。

```
*
*              ****************************
*              *  OUTPUT RELATION NAME    *
*              ****************************
*
****************
*    HI/ESR    *
**************** <RL=80>
*
 01  HI-ESR-PUT.
     03  HI-ESR-ENTITY-NAME              PIC  X(16).
     03  HI-ESR-AREA-ROOT                PIC  X(16).
     03  HI-ESR-NO-OF-KEYS               PIC  9(04).
     03  HI-ESR-MODE                     PIC  X(02).
     03  HI-ESR-DEGREE                   PIC  9(02).
     03  HI-ESR-WIDTH                    PIC  9(04).
     03  HI-ESR-SIZE                     PIC  9(08).
     03  HI-ESR-PROTECTION               PIC  X.
     03  HI-ESR-INTEGRITY                PIC  X.
     03  HI-ESR-CARDINALITY              PIC  9(04).
     03  HI-ESR-TID                      PIC  X(08).
     03  HI-ESR-FILLER                   PIC  X(14).
*
****************
*    HI/ATT    *
**************** <RL=80>
*
 01  HI-ATT-PUT.
     03  HI-ATT-ES-RS-NAME               PIC  X(16).
     03  HI-ATT-ES-RS-TYPE               PIC  X(02).
     03  HI-ATT-ATTRIBUTE                PIC  X(16).
     03  HI-ATT-VALUE-SET                PIC  X(12).
     03  HI-ATT-ROLE                     PIC  X(05).
     03  HI-ATT-TYPE                     PIC  X.
     03  HI-ATT-LENGTH                   PIC  9(03).
     03  HI-ATT-PROTECTION               PIC  X.
     03  HI-ATT-INTEGRITY                PIC  X.
     03  HI-ATT-CARDINALITY              PIC  9(04).
     03  HI-ATT-TID                      PIC  X(08).
     03  HI-ATT-FILLER                   PIC  X(11).
*
****************
*    HI/RSR    *
**************** <RL=120>
*
 01  HI-RSR-PUT.
     03  HI-RSR-RELATIONSHIP             PIC  X(16).
     03  HI-RSR-SOURCE-ES                PIC  X(16).
     03  HI-RSR-DEST-ES                  PIC  X(16).
     03  HI-RSR-MODE                     PIC  X(02).
     03  HI-RSR-RS-TYPE.
         05  HI-RSR-RS-T1                PIC  X.
         05  HI-RSR-RS-T2                PIC  X.
         05  HI-RSR-RS-T3                PIC  X.
     03  HI-RSR-DEGREE                   PIC  9(02).
     03  HI-RSR-WIDTH                    PIC  9(04).
     03  HI-RSR-SIZE                     PIC  9(08).
     03  HI-RSR-SOURCE-SET               PIC  X(16).
     03  HI-RSR-DEST-SET                 PIC  X(16).
     03  HI-RSR-PROTECTION               PIC  X.
     03  HI-RSR-INTEGRITY                PIC  X.
     03  HI-RSR-CARDINALITY              PIC  9(04).
     03  HI-RSR-TID                      PIC  X(08).
     03  HI-RSR-FILLER                   PIC  X(07).
```

図３－１２　出力リレーション形式

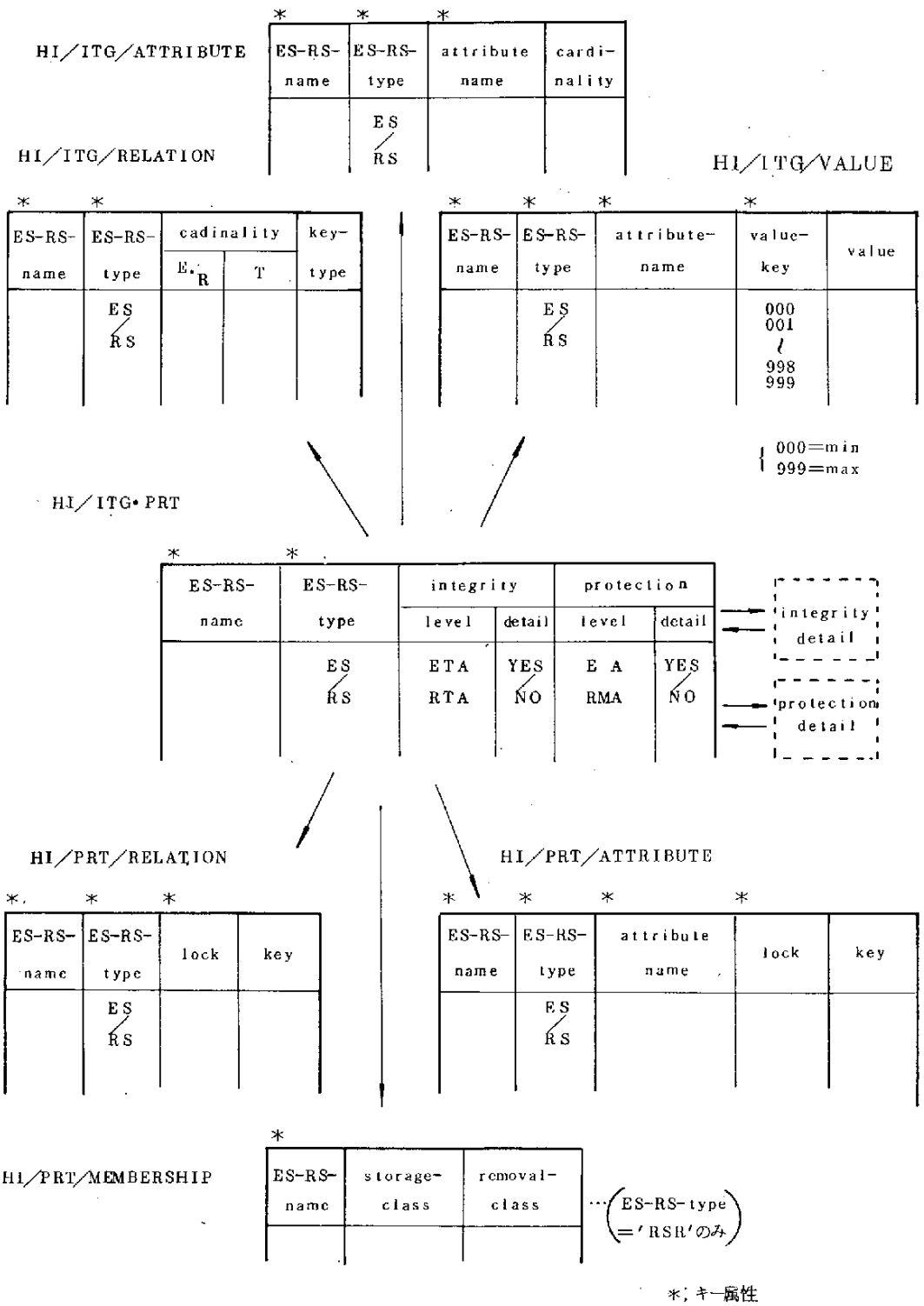


図３－１３　インテグリティとプロテクション（integrity and protection）

```
*
*********************
*    HI/ITG.PRT     *
********************* <RL=60>
*
 01  HI-ITG-PRT-PUT.
     03  ITG-PRT-ES-RS-NAME              PIC  X(16).
     03  ITG-PRT-ES-RS-TYPE              PIC  X(02).
     03  ITG-PRT-ITG-LEVEL.
         05  ITG-PRT-ITG-L1              PIC  X.
         05  ITG-PRT-ITG-L2              PIC  X.
         05  ITG-PRT-ITG-L3              PIC  X.
     03  ITG-PRT-ITG-DETAIL              PIC  X(04).
     03  ITG-PRT-PRT-LEVEL.
         05    ITG-PRT-PRT-L1             PIC  X.
         05    ITG-PRT-PRT-L2             PIC  X.
         05    ITG-PRT-PRT-L3             PIC  X.
     03  ITG-PRT-PRT-DETAIL              PIC  X(04).
     03  ITG-PRT-TID                     PIC  X(08).
     03  ITG-PRT-FILLER                  PIC  X(20).
*
************************
*    HI/PRT.RELATION   *
************************ <RL=60>
*
 01  HI-PRT-REL-PUT.
     03  PRT-REL-ES-RS-NAME              PIC  X(16).
     03  PRT-REL-ES-RS-TYPE              PIC  X(02).
     03  PRT-REL-LOCK                    PIC  X(03).
     03  PRT-REL-KEY                     PIC  X(10).
     03  PRT-REL-TID                     PIC  X(08).
     03  PRT-REL-FILLER                  PIC  X(21).
*
**************************
*    HI/PRT.MEMBERSHIP   *
************************** <RL-60>
*
 01  HI-PRT-MEM-PUT.
     03  PRT-MEM-ES-RS-NAME              PIC  X(16).
     03  PRT-MEM-STORAGE-CLASS           PIC  X(09).
     03  PRT-MEM-REMOVAL-CLASS           PIC  X(09).
     03  PRT-MEM-TID                     PIC  X(08).
     03  PRT-MEM-FILLER                  PIC  X(19).
*
************************
*    HI/PRT.ATTRIBUTE  *
************************ <RL=60>
*
 01  HI-PRT-ATT-PUT.
     03  PRT-ATT-ES-RS-NAME              PIC  X(16).
     03  PRT-ATT-ES-RS-TYPE              PIC  X(02).
     03  PRT-ATT-ATTRIBUTE               PIC  X(16).
     03  PRT-ATT-LOCK                    PIC  X(03).
     03  PRT-ATT-KEY                     PIC  X(10).
     03  PRT-ATT-TID                     PIC  X(08).
     03  PRT-ATT-FILLER                  PIC  X(05).
```

図３－１４　出力リレーション形式

```
*
************************
*    HI/ITG.RELATION   *
************************ <RL=60>
*
 01  HI-ITG-REL-PUT.
     03  ITG-REL-ES-RS-NAME          PIC  X(16).
     03  ITG-REL-ES-RS-TYPE          PIC  X(02).
     03  ITG-REL-CDNL-ES-RS          PIC  9(04).
     03  ITG-REL-CDNL-TUPLE          PIC  9(04).
     03  ITG-REL-KEY-TYPE            PIC  X.
     03  ITG-REL-TID                 PIC  X(08).
     03  ITG-REL-FILLER              PIC  X(25).
*
************************
*    HI/ITG.ATTRIBUTE  *
************************ <RL=60>
*
 01  HI-ITG-ATT-PUT.
     03  ITG-ATT-ES-RS-NAME          PIC  X(16).
     03  ITG-ATT-ES-RS-TYPE          PIC  X(02).
     03  ITG-ATT-ATTRIBUTE           PIC  X(16).
     03  ITG-ATT-CARDINALITY         PIC  9(04).
     03  ITG-ATT-TID                 PIC  X(08).
     03  ITG-ATT-FILLER              PIC  X(14).
*
**********************
*    HI/ITG.VALUE    *
********************** <RL=60>
*
 01  HI-ITG-VAL-PUT.
     03  ITG-VAL-ES-RS-NAME          PIC  X(16).
     03  ITG-VAL-ES-RS-TYPE          PIC  X(02).
     03  ITG-VAL-ATTRIBUTE           PIC  X(16).
     03  ITG-VAL-VALUE-KEY           PIC  9(03).
     03  ITG-VAL-VALUE               PIC  9(07).
     03  ITG-VAL-TID                 PIC  X(08).
     03  ITG-VAL-FILLER              PIC  X(08).
```

図３－１５　出力リレーション形式

3.1.3　リレーショナル記述

ＤＢエントリ及びモジュールエントリで記述されたＬＣＳ／ＨＩ情報は、ＮＤＤ／ＨＩＰＳサブシステムにより解析され、ＮＤＤのＨＩ情報として新たに生成される。リレーショナル記述、即ち、ＲＤとは事象集合や関係性集合の記述をユーザもしくは問合せ（query）がみるときの視点（view）であり、リレーションの一種である。本サブシステムでは、生成した各リレーションに対して、図３－１６の形式に基づいた問合せ機能を有している。指定項目は、対象リレーション名、リレーションタイプ、編集項目名等である。

形式：

HI／RELATION ({REL−TYPE / TYPE} = {ES / RS}, schema-name, database-name, host-id)

({ [rel-name [, rel-name] ･･･] / REL / RELATION } [, { ATT / ATTRIBUTE } [, { ROL / ROLE }]])

図３－１６　ＲＤの指定形式

次の例は、ＡＣＯＳ（host-id）にあるＰＲＯＪ（database-name）のＰＲＯＪＥＣＴ（schema-name）について、全ての関係性集合（ＲＥＬ－ＴＹＰＥ＝ＲＳ）を求めている。また、合わせてその属性構成と役割についても要求している。

```
        JIPDEC  NDD/HIPS    ( RELATIONAL DESCRIPTION )              DATE 02/15/79        PAGE    1

PARAMETER CARD LIST : HI/ATT ( REL-TYPE = RS, PROJECT, PROJ, ACOS )
                             ( RELATION, ATT, ROLE )
```

```
        JIPDEC  NDD/HIPS    ( RELATIONAL DESCRIPTION )              DATE 02/15/79        PAGE    2

RD/RSR : SCHEMA NAME = PROJECT          DATABASE NAME = PROJ          MODEL =          HOST = ACOS

        RELATIONSHIP SET    ATTRIBUTE NAME (DOMAIN)    ROLE OF KEY    ATTRIBUTE NAME (DOMAIN)    ROLE OF KEY

        REPR-PROJ           REPR-NO (REPR-NO )         K0101          PROJ-NO (PROJ-NO )         K0102

        PROJ-SPON-LINK      SPONNAME (SPONNAME )       K0102          PROJ-NO (PROJ-NO )         K0102

                            BUDGET (BUDGET )
```

図３－１７　RD 指定と表示例

3.1.4 例　　題

本項では、ＤＢＴＧデータモデルに基づいて記述したスキーマ（ＬＩＳ）を採り上げ、ＮＤＤ／ＨＩＰＳサブシステムによりＨＩ情報が生成される過程を示す。引用するデータベース（プロジェクトデータベース）のデータ構造図を図３－１８に示す。ただし例題の範囲は斜線で囲まれた部分である。図３－１９は、この部分のデータ構造図にスキーマを付加したものである。さらに、図３－２０は、Ｅ－Ｒモデルに基づいたＥ－Ｒ図（Entity-Relationship Diagram）である。

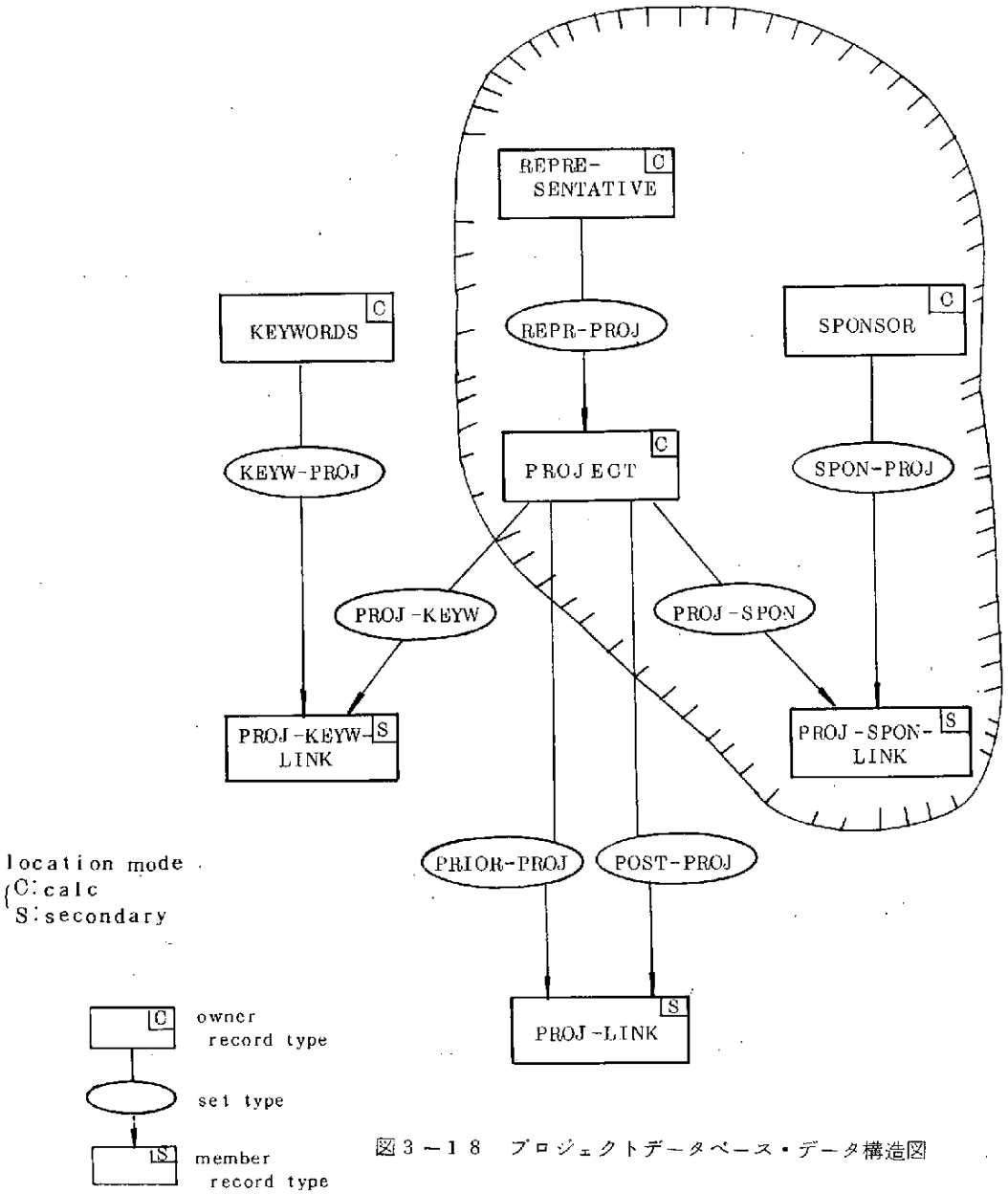


図３－１８　プロジェクトデータベース・データ構造図

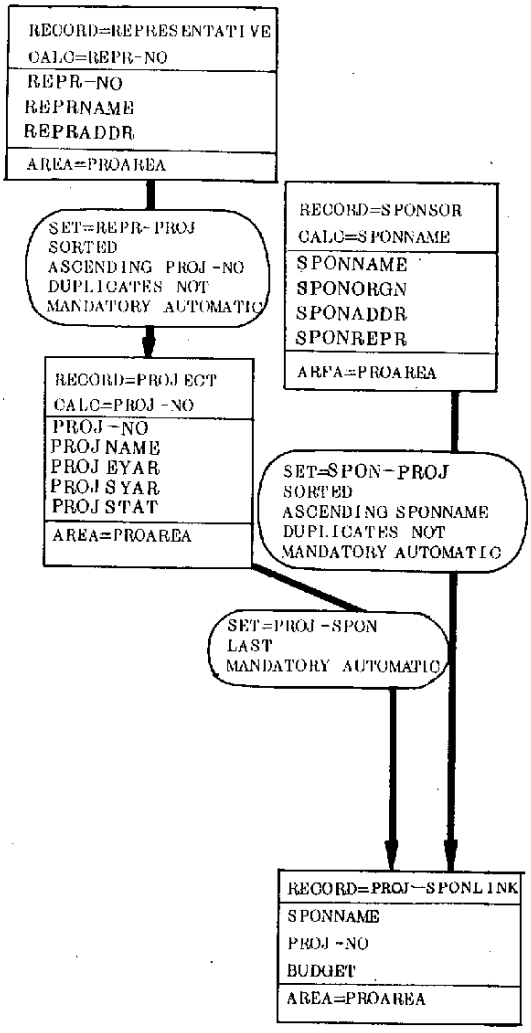


図３－１９　スキーマを伴なうデータ構造図

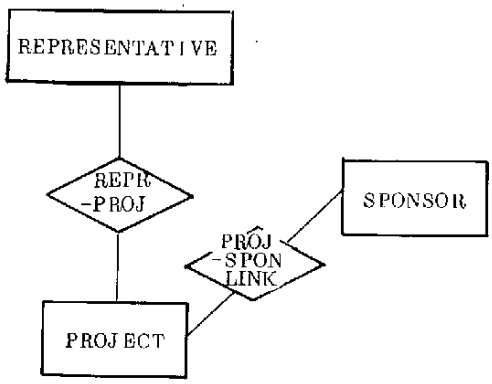


図３－２０　プロジェクトデータベース：LCS（E－R図）

「3.1.2 LCS/HI 記述」に基づいて、例題のLCS/HI記述を以下に示す。

```
##DB/PROJECT/PROJ/DBTG/ACOS/C
ESRC REPRESENTATIVE   101 PROAREA          CU PN IN 0100
ESRC REPRESENTATIVE   201 REPR-NO         C 020 K0001 PN IN
ESRC REPRESENTATIVE   202 REPRNAME        C 020       PN IN
ESRC REPRESENTATIVE   203 REPRADDR        C 020       PN IN
ESRC PROJECT          101 PROAREA          CU PN IN 0100
ESRC PROJECT          201 PROJ-NO          C 020 K0001 PN IN
ESRC PROJECT          202 PROJNAME         C 020       PN IN
ESRC PROJECT          203 PROJEYAR         D 006       PN IN
ESRC PROJECT          204 PROJSYAR         D 006       PN IN
ESRC PROJECT          205 PROJSTAT         D 002       PN IN
ESRC SPONSOR          101 PROAREA          CU PN IN 0100
ESRC SPONSOR          201 SPONNAME         C 020 K0001 PN IN
ESRC SPONSOR          202 SPONOGEN         C 020       PN IN
ESRC SPONSOR          203 SPONADDR         C 020       PN IN
ESRC SPONSOR          204 SPONREPR         C 020       PN IN
RSRC REPR-PROJ        101 REPRESENTATIVE   PROJECT          VS S   PN IN 0200
RSRC REPR-PROJ        301 REPR-NO          C 020 K0101 PN IN
RSRC REPR-PROJ        401 PROJ-NO          C 020 K0102 PN IN
RSRC PROJ-SPON-LINK   101 SPONSOR          PROJECT          VS LN  PN IN 0200
RSRC PROJ-SPON-LINK   201 SPON-PROJ        PROJ-SPON
RSRC PROJ-SPON-LINK   301 SPONNAME         C 020 K0102 PN IN
RSRC PROJ-SPON-LINK   401 PROJ-NO          C 020 K0102 PN IN
RSRC PROJ-SPON-LINK   501 BUDGET           D 010       PN IN
PRTC PROJECT          101 ESR DEL JIPDEC       REM JIPDEC
PRTC PROJECT          301 ESR PROJ-NO          MOD JIPDEC
PRTC REPR-PROJ        101 RSR FIN JIPDEC
PRTC REPR-PROJ        201 RSR AUTOMATIC MANDATORY
PRTC PROJ-SPON-LINK   101 RSR FIN JIPDEC
PRTC PROJ-SPON-LINK   201 RSR AUTOMATIC MANDATORY
PRTC PROJ-SPON-LINK   301 RSR SPONNAME         MOD JIPDEC
PRTC PROJ-SPON-LINK   302 RSR PROJ-NO          MOD JIPDEC
PRTC PROJ-SPON-LINK   901 RSR THIS RECORD TYPE MUST BE HANDLED BY DBA.
PRTC SPONSOR          101 ESR DEL JIPDEC       REM JIPDEC
PRTC SPONSOR          102 ESR INS JIPDEC
PRTC SPONSOR          201 ESR AUTOMATIC MANDATORY
PRTC SPONSOR          301 ESR SPONNAME         MOD JIPDEC     GET JIPDEC
PRTC SPONSOR          302 ESR                  STO JIPDEC
PRTC REPRESENTATIVE   101 ESR DEL JIPDEC
ITGC PROJECT          101 ESR M 0100 0100
ITGC PROJECT          201 ESR PROJ-NO          0100 R 0000001 0000100
ITGC PROJECT          901 ESR A PROJ-NO CORRESPONDS TO ONE OR MORE PROJECT.
ITGC REPRESENTATIVE   101 ESR U 0100 0100
ITGC REPRESENTATIVE   201 ESR REPR-NO          0100 E 0000004 0000005 0000020
ITGC REPRESENTATIVE   202 ESR                        0000100
ITGC REPRESENTATIVE   901 ESR A REPRESENTATIVE MUST LIVE IN TOKYO.
ITGC KEYWORDS         101 ESR M 0100 0100
ITGC SPONSOR          101 ESR U 0050 0050
ITGC SPONSOR          201 ESR PROJ-NO          0100 R 0000001 0000100
##END
```

図3－21　LCS／HI記述例

図3－22以降は、NDD／HIPSにより生成されたHI情報の各リレーションの内容を示したものである。また、図3－32に本サブシステムが生成したりレーションのディレクトリを示す。

JIPDEC NDD/HIPS (ENTITY SET RELATION) DATE 02/15/79 PAGE 1

HI/ESR : SCHEMA NAME = PROJECT DATABASE NAME = PROJ MODEL = DBTG HOST = M

ENTITY SET NAME	AREA ROOT NAME	NO. OF KEYS	MODE	DEGREE	WIDTH	SIZE	PRT	ITG	CARDINALITY	TID
REPRESENTATIVE	PROAREA	1	CU	3	60	6,000	N	N	100	
PROJECT	PROAREA	1	CU	5	54	5,400	N	N	100	
SPONSOR	PROAREA	1	CU	4	80	8,000	N	N	100	

図３－２２　リレーション：ＨＩ／ＥＳＲ

JIPDEC NDD/HIPS (RELATIONSHIP SET RELATION)

HI/RSR : SCHEMA NAME = PROJECT DATABASE NAME = PROJ

RELATIONSHIP SET	SOURCE ES	DESTINATION ES	MODE	TYPE	DEG	WIDTH
REPR-PROJ	REPRESENTATIVE	PROJECT	VS	S	2	40
PROJ-SPON-LINK	SPONSOR	PROJECT	VS	LN	3	50

図３－２３　リレーション：ＨＩ／ＲＳＲ

JIPDEC NDD/HIPS (ATTRIBUTE) DATE 02/15/79 PAGE 1

HI/ATT : SCHEMA NAME = PROJECT DATABASE NAME = PROJ MODEL = DBTG HOST = M

ES-RS NAME	ES/RS	ATTRIBUTE NAME	VALUE SET	ROLE OF KEY	TYPE	LENGTH	PRT	ITG	CARDINALITY	TID
REPRESENTATIVE	ES	REPR-NO	REPR-NO	K0001	C	20	N	N	0	
		REPRNAME	REPRNAME		C	20	N	N	0	
		REPRADDR	REPRADDR		C	20	N	N	0	
PROJECT	ES	PROJ-NO	PROJ-NO	K0001	C	20	N	N	0	
		PROJNAME	PROJNAME		C	20	N	N	0	
		PROJEYAR	PROJEYAR		D	6	N	N	0	
		PROJSYAR	PROJSYAR		D	6	N	N	0	
		PROJSTAT	PROJSTAT		D	2	N	N	0	
SPONSOR	ES	SPONNAME	SPONNAME	K0001	C	20	N	N	0	
		SPONOGEN	SPONOGEN		C	20	N	N	0	
		SPONADDR	SPONADDR		C	20	N	N	0	
		SPONREPR	SPONREPR		C	20	N	N	0	
REPR-PROJ	RS	REPR-NO	REPR-NO	K0101	C	20	N	N	0	
		PROJ-NO	PROJ-NO	K0102	C	20	N	N	0	
PROJ-SPON-LINK	RS	SPONNAME	SPONNAME	K0102	C	20	N	N	0	
		PROJ-NO	PROJ-NO	K0102	C	20	N	N	0	
		BUDGET	BUDGET		D	10	N	N	0	

図３－２４　リレーション：ＨＩ／ＡＴＴ

JIPDEC NDD/HIPS (INTEGRITY , PROTECTION) DATE 02/15/79 PAGE 1

HI/ITG,PRT : SCHEMA NAME = PROJECT DATABASE NAME = PROJ MODEL = DBTG HOST = M

ES-RS NAME	ES/RS	INTEGRITY LEVEL	DETAIL	PROTECTION LEVEL	DETAIL	TID
PROJECT	ES	ETA	Y	E A	N	
REPRESENTATIVE	ES	ETA	Y	E	N	
KEYWORDS	ES	ET	N			
SPONSOR	ES	ETA	N	EMA	N	
REPR-PROJ	RS			RM	N	
PROJ-SPON-LINK	RS			RMA	Y	

図３－２５　リレーション：ＨＩ／ＩＴＧ・ＰＲＴ

JIPDEC NDD/HIPS (HI/PRT/RELATION) DATE 02/15/79 PAGE 1

HI/PRT : SCHEMA NAME = PROJECT DATABASE NAME = PROJ MODEL = DBTG HOST = M

ES-RS NAME	ES/RS	LOCK	KEY	LOCK	KEY	LOCK	KEY
PROJECT	ES	DEL	JIPDEC	REM	JIPDEC		
REPR-PROJ	RS	FIN	JIPDEC				
PROJ-SPON-LINK	RS	FIN	JIPDEC				
SPONSOR	ES	DEL	JIPDEC	REM	JIPDEC	INS	JIPDEC
REPRESENTATIVE	ES	DEL	JIPDEC				

図３－２６　リレーション：ＨＩ／ＰＲＴ／ＲＥＬＡＴＩＯＮ

JIPDEC NDD/HIPS (HI/PRT/MEMBERSHIP) DATE 02/15/79 PAGE 1

HI/PRT : SCHEMA NAME = PROJECT DATABASE NAME = PROJ MODEL = DBTG HOST = M

ES-RS NAME	ES/RS	STORAGE CLASS	REMOVAL CLASS	TID
REPR-PROJ	RS	AUTOMATIC	MANDATORY	
PROJ-SPON-LINK	RS	AUTOMATIC	MANDATORY	
SPONSOR	RS	AUTOMATIC	MANDATORY	

図3－27　リレーション：HI／PRT／MEMBERSHIP

JIPDEC NDD/HIPS (HI/PRT/ATTRIBUTE) DATE 02/15/79 PAGE 1

HI/PRT : SCHEMA NAME = PROJECT DATABASE NAME = PROJ MODEL = DBTG HOST = M

ES-RS NAME	ES/RS	ATTRIBUTE NAME	LOCK	KEY	LOCK	KEY
PROJECT	ES	PROJ-NO	MOD	JIPDEC		
PROJ-SPON-LINK	RS	SPONNAME	MOD	JIPDEC		
PROJ-SPON-LINK	RS	PROJ-NO	MOD	JIPDEC		
SPONSOR	ES	SPONNAME	MOD	JIPDEC	GET	JIPDEC
			STO	JIPDEC		

図3－28　リレーション：HI／PRT／ATTRIBUTE

JIPDEC NDD/HIPS (HI/ITG/RELATION) DATE 02/15/79 PAGE 1

HI/ITG : SCHEMA NAME = PROJECT DATABASE NAME = PROJ MODEL = DBTG HOST = M

ES-RS NAME	ES/RS	CARDINALITY ES-RS	CARDINALITY TUPLE	KEY TYPE	TID
PROJECT	ES	100	100	M	
REPRESENTATIVE	ES	100	100	U	
KEYWORDS	ES	100	100	M	
SPONSOR	ES	50	50	U	

図3－29 リレーション：HI／ITG／RELATION

JIPDEC NDD/HIPS (HI/ITG/ATTRIBUTE) DATE 02/15/79 PAGE 1

HI/ITG : SCHEMA NAME = PROJECT DATABASE NAME = PROJ MODEL = DBTG HOST = M

ES-RS-NAME	ES/RS	ATTRIBUTE NAME	CARDINALITY	TID
PROJECT	ES	PROJ-NO	100	
REPRESENTATIVE	ES	REPR-NO	100	
SPONSOR	ES	PROJ-NO	100	

図3－30 リレーション：HI／ITG／ATTRIBUTE

JIPDEC NDD/HIPS (HI/ITG/VALUE) DATE 02/15/79 PAGE 1

HI/ITG : SCHEMA NAME = PROJECT DATABASE NAME = PROJ MODEL = DBTG HOST = M

ES-RS NAME	ES/RS	ATTRIBUTE NAME	VALUE KEY	VALUE	VALUE KEY	VALUE	VALUE KEY	VALUE
PROJECT	ES	PROJ-NO	000	1	999	100		
REPRESENTATIVE	ES	REPR-NO	001	4	002	5	003	20
			004	100				
SPONSOR	ES	PROJ-NO	000	1	999	100		

図３－３１　リレーション：ＨＩ／ＩＴＧ／ＶＡＬＵＥ

JIPDEC NDD/HIPS (DIRECTORY OF LCS/HI) DATE 02/15/79 PAGE 1

SCHEMA NAME	DATABASE NAME	MODEL	HOST	ESR	ATT	RSR	ITG REL	ITG ATT	ITG VAL	PRT REL	PRT ATT	PRT MEM
PROJECT	PROJ	DBTG	ACOS	YES	YES	YES	YES	YES	YES	YES	YES	YES

図３－３２　ＬＣＳ／ＨＩディレクトリ

3.2　クリアリング情報生成システム／分散情報生成サブシステム（NDD/DIPS）

ＮＤＤ／ＨＩＰＳは、各データベースサイトの異種情報（ＨＩ）をＤＢＭＳのスキーマから生成するシステムである。ここで述べるＤＩＰＳ（distribution information processing system）は、ＨＩＰＳによって生成されたＨＩと、全体管理者（ＧＡ）が定義した分散記述（ＤＤ）とをもとにして、ＧＣＳのＲＤ（relational description）と、分散情報（ＤＩ）とを生成するシステムである（図３－３３）。

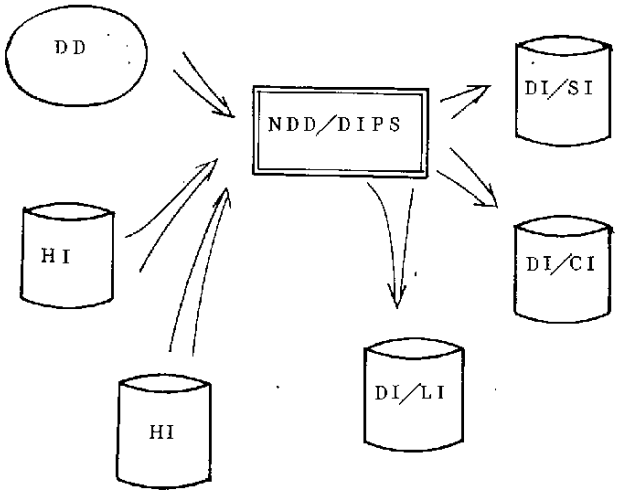


図３－３３　ＮＤＤ／ＤＩＰＳの概要

3.2.1　ＤＤとその木構造表現

ＤＤの形式的定義は、2.6.2に述べた。ＤＤは、ＱＵＥＬに基づいた仕様で記述され、ＲＡＮＧＥ文とＤＥＦＩＮＥ文の集合から成っている。これらのＤＤの文字列は、システム内では、２進木構造と、いくつかのテーブルから表わされる(図３－３４)。２進木構造とした理由は、tree 内の部分 tree（subtree）の共有を可能とするためである。

ＲＴＢＬ（range table）は、range 文によって宣言された組変数名とその range 局所リレーション名及びその所在サイト番号との対応表になっている。ＳＮＴＢＬ（site-no table）は、range 文内に表われたサイト番号のテーブルである。ＧＲＴＢＬは、define文Ｄによって定義された全体リレーション名、そのリレーションタイプ及び次数が格納されている。ＧＡＴＢＬは、define 文Ｄ内で定義された全体リレーションを構成する属性名、その領域名（値集合名）、キー情報、属性番号、フォーマットが格納されている。これらのリレーション・スキームは表３－３にまとめられている。

２進木構造として表わされるのは、＜g-constraint＞{ ：＜g-constraint＞}の文字列である。ＧＢＲノードは、１つの＜g-constraint＞を表わすtreeへの分岐ノードである。さらに＜g-constraint＞木は、次のように表わされる。

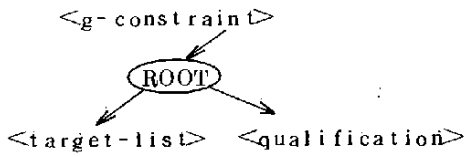


表3－3　テーブルのリレーション・スキーム

テーブル名	リレーション・スキーム	属　性　名	説　　名
RTBL	RTBL (VAR, RELNAME, SITENO)	VAR RELNAME SITENO	組変数名 局所リレーション名 サイト番号
SNTBL	SNTBL (SITENO)	SITENO	サイト番号
GRTBL	GRTBL(GRELNAME, RTYPE, DEGREE)	GRELNAME RTYPE DEGREE	全体リレーション名 リレーション型 (ESRor RSR) 次数（属性数）
GATBL	GATBL (ATTNO, GATTNAME, DNAME, FTYPE, FLENGTH)	ATTNO GATTNAME DNAME FTYPE FLENGTH	属性番号 属性名 領域名 属性のとる値の型 属性のとる値の長さ

DD

range of (l_i, ⋯, l_n)

(L_i:s_i, ⋯, L_n:s_n);

D : define rel−type G(ga_1/GA_1=t_1|l_1,

⋯, gam/GAm=tm|m)

< g constraint > { :<g constraint> } ;

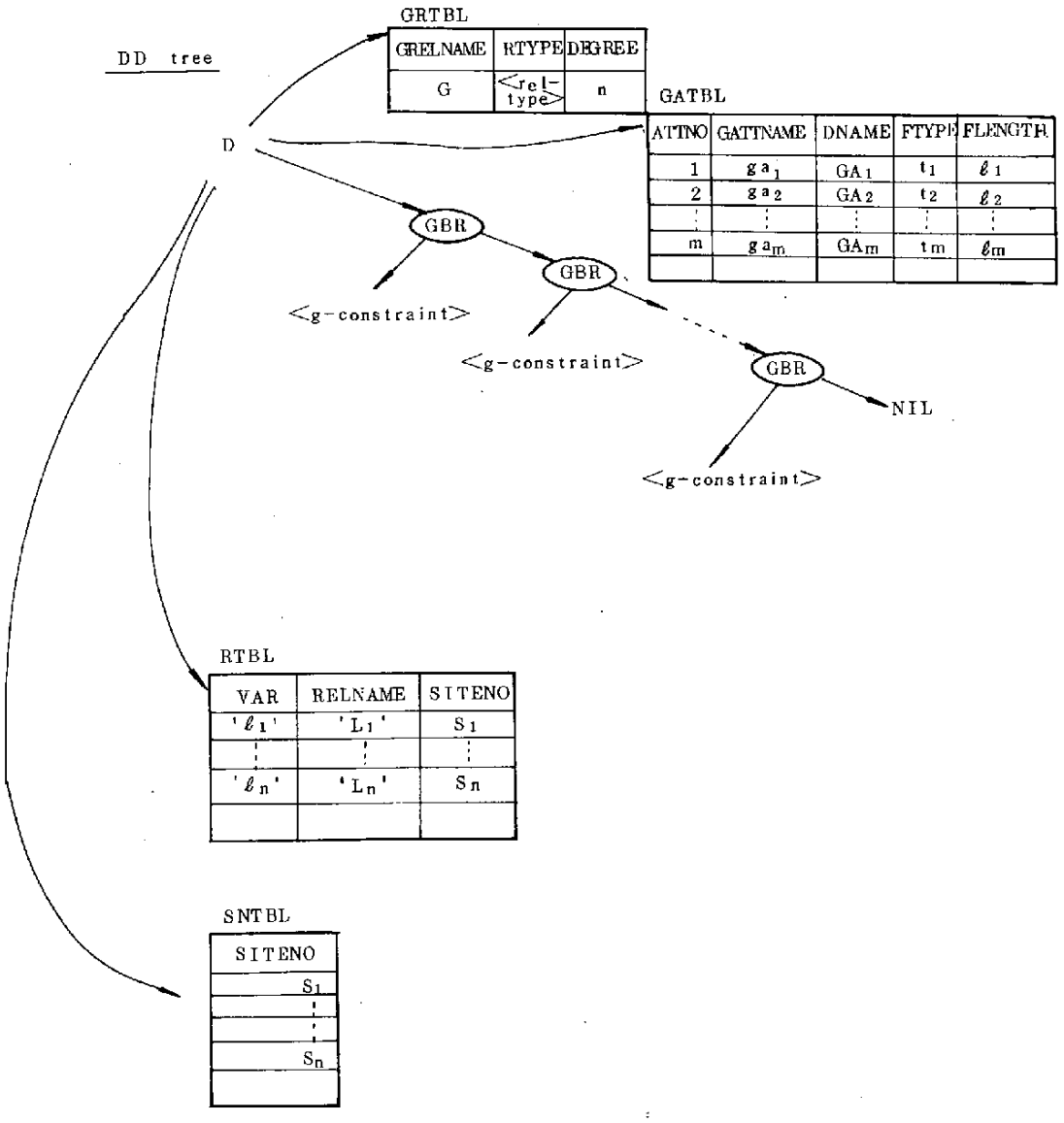
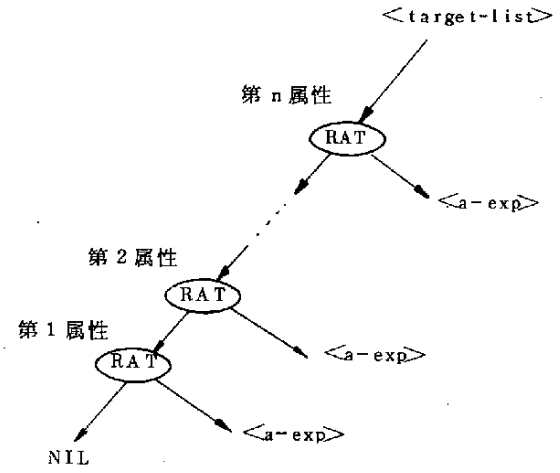
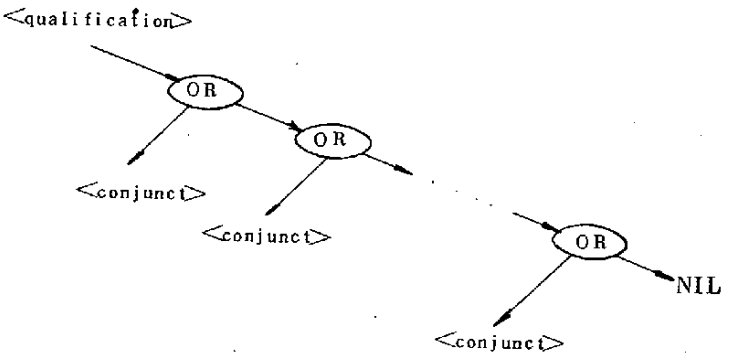


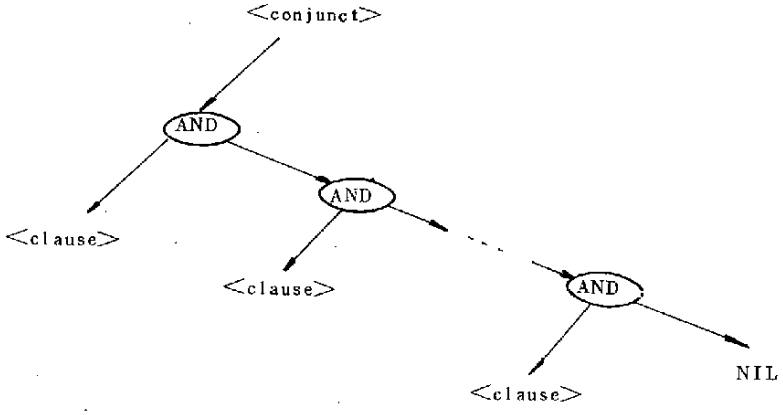
図3－34　DDとDD木

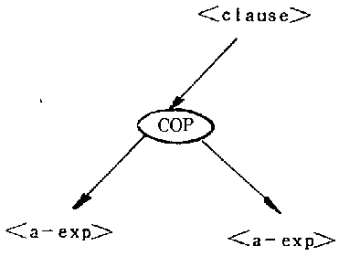


＜target-list＞ tree における ＲＡＴノード（result-attribute node）は、define 文内の＜a-list＞に対応して、最左下が第１属性（即ち、属性番号が１）、このＲＡＴの右上のＲＡＴが第２属性，…，最上部が最後の属性を表わしている。

＜qualification ＞部の論理式は、離接正規化されているために次の木構造をもつ。

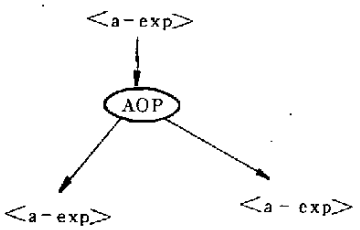






ここで、ＣＯＰは、比較演算子（＜、≦、＝、≧、＞、≠）を表わすノードである。

＜ａ－ｅｘｐ＞は、定数と属性（＜variable ＞，＜attribute ＞)の算術式を表わしている（関数もふくむ）。２項演算子（＋、＊、／、＊＊）に対する＜ａ－ｅｘｐ＞は次のように表わされる。

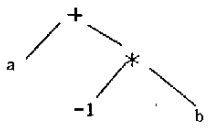


演算子（－）は、次のように（－１）をかけてあらわす。即ち、－ａは

と表わされる。

例えばａ－ｂは

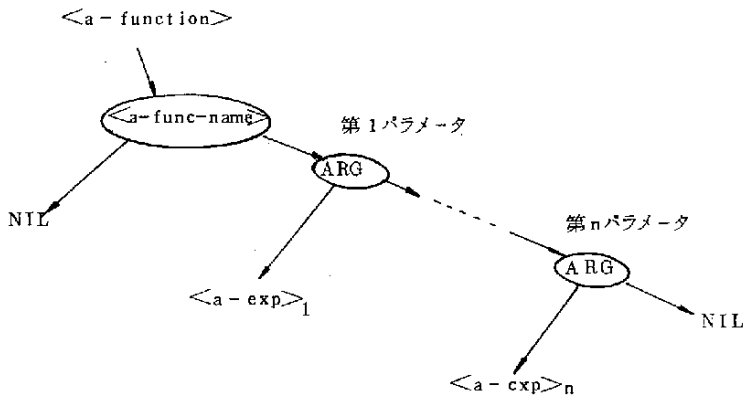
と表わされる。このような表現を用



いる理由は、式の簡約化（simplification）を容易とするためである。

関数の演算子には、２種類ある。１つは通常の算術関数（e.g. ABS, EXP, LOG, etc）である。これは、次のような形式をもつ。

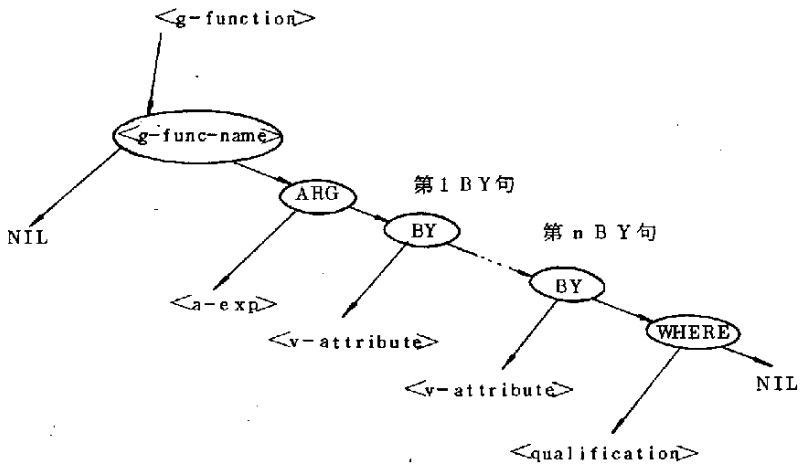
＜a-function＞ ::＝＜a-func-name＞(＜a-exp＞$_1$, … ,＜a-exp＞$_n$)対応する tree は、



他の関数は、aggregate 関数（e.g. SUM, COUNT, AVER, etc）である。これは次のような形式をもつ。

＜g-function＞ ::＝＜g-func-name＞ (＜a-exp＞
〔BY ＜v-attribute＞{ BY ＜v-attribute＞}〕
〔WHERE ＜qualification＞〕)

対応する tree は、次のようになる。



例えば、range of (a、b、d) (AUTHOR：3、BOOK：1、DOCUMENT：2)；

define RSR AUTHOR_REP(a#/AA#＝I5、aname＝C20，

rep#＝I5、rep_name＝C50)

(a#＝a.author #、aname＝a.name、rep#＝a.report #、

rep_name＝b.name) where a.report #＝b.book #

and b.book #≧5000

：(a#＝a.author #、aname＝a.name、rep#＝a.report #

rep_name＝d.dname) where a.report #＝d.d #；

は、図3－35のようになる。テーブルは、図3－36のようになる。

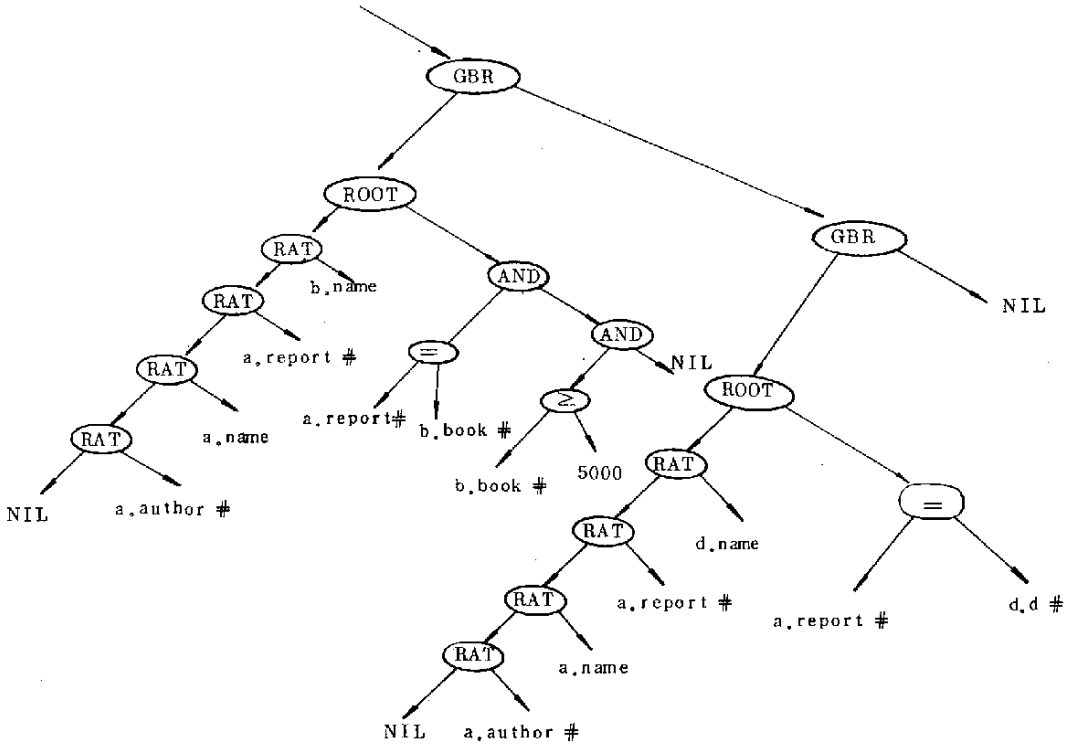


図3－35　DD－tree の例

RTBL

	VAR	RELNAME	SITENO
1	'a'	'AUTHOR'	3
2	'b'	'BOOK'	1
3	'd'	'DOCUMENT'	2

SNTBL

	SITENO
1	3
2	1
3	2

GRTBL

GRELNAME	RTYPE	DEGREE
'AUTHOR_REP'	'RSR'	4

GATBL

ATTNO	GATTNAME	DNAME	FTYPE	FLENGTH
1	'a#'	'AA#'	I	5
2	'aname'	'ANAME'	C	20
3	'rep#'	'REP#'	I	5
4	'rep_name'	'REP_NAME'	C	50

図３－３６　テーブルの例

木（tree）は、メモリ内では一連の３語（４bytes／語）から成るノードによって表わされる。ノードは、原子ノード（atomic node）と演算子ノード（operator node）との２種類がある。

原子ノードは、木の葉（leaf）に対応するノードで、属性、定数等を表わす。属性ノード（attribute node：ATTN）は、目標リスト及び条件部の＜a-exp＞（算術式）内に表われる属性を表わしている。ＡＴＴＮが持つ情報として、組変数名、属性名、属性のrole（key, non-key, composite-key）、インデックス情報、属性のとりうる値のフォーマットがある。

定数ノード（constant node：CNTN）は、整数又は文字定数を表わす。４つのノードは、９ケタまでの整数と４文字までの文字を格納できる。これを超えるものは、複数のＣＮＴＮをチェインして表わす。

演算子ノードは、２進の分岐を表わすノードである。この種のノードとしては、算術演算子ノード（ＡＯＰＮ）は、比較演算子ノード（ＣＯＰＮ）、論理演算子ノード（ＬＯＰＮ）、算術関数ノ

ード（AFNCN）、集合関数ノード（GFNCN）、＜target-list＞と＜qualification＞木とのルートを表わすルートノード（ROOTN）、結果属性ノード（RATN）、＜g-constraint＞を表わすGBRN、関数内のパラメータを表わすノード（ARGN, WHEREN, BYN）とがある。これらは、演算子の種類とともに、サイト数、変数の数等のパフォーマンス情報をもっている。

3.2.2 対応情報（NDD／DI／CI）

対応情報は、全体リレーションと局所リレーションとの対応情報である。3.2.1で述べたように、DDとして表わされたRANGE文とDEFINE文の文字列は、木構造として内部的に表わされる。問合せの処理においても、問合せは、DD−tree と同様な木構造として表わされ、DD−tree の必要部分（subtree）が問合せ tree に接合される。即ち、問合せは、システム内では、木として処理される。よって、対応情報はメモリ内で tree として表わされる。

DI／CIは、ファイルとして存在するために、tree 構造を、sequential なノードの列に変換して格納する。これは、tree 内のノードを、postorder〔KNUT73〕に基づいて番号づけし、その順に格納することによってなされる。各々ノードは、DI__CI__GRELリレーションの1つの組（tuple）として格納される。DI__CI__GREL．NODEは、このノードの内容を表わす属性である。NODE−NO は、このノードのDD− tree 内での postorder に基づいた順番を表わしている。

GLOBAL_RELNAMEとES_RS_TYPE は、全体リレーション名と、このリレーションの型（ESR or RSR）を表わしている。これらの値は、GRTBLより得られる。

DI__CI__RANGEリレーションは、RTBL（range table）である。DD−tree 内の属性の持つ組変数と、局リレーション名、その所在サイトとの対応表である。DI__CI__SITE リレーションは、SNTBLより生成される。

表3−4に、NDD／DI／CIのスキーマをのせる。

表3−4　NDD／DI／CIスキーマ

リレーション名	リレーション・スキーム	属性名	説明
DI_CI_GREL	DI_CI_GREL	GLOBAL_RELNAME	全体リレーション名
	(GLOBAL−RELNAME,	ES_RS_TYPE	ESR or RSR
	ES_RS_TYPE,		
	NO_OF_NODES,	NO_OF_NODES	このDD−tree 内の全ノード数
	NODE_NO, NODE)	NODE_NO	このノードの tree 内の番号（in postorder）
		NODE	ノードの中味
DI_CI_RANGE	DI_CI_RANGE	VAR	組変数名
	(VAR, LOCAL−RELNAME,		
	LOCATION−ID)	LOCAL_RELNAME	局所リレーション名
		LOCATION_ID	サイト番号
DI_CI_SITE	DI−CI−SITE	POSITION	SNTBL内の position
	(POSITION,		
	LOCATION_ID)	LOCATION_ID	サイト番号

3.2.3 スキーマ情報（NDD／DI／SI）

スキーマ情報は、GCSにおけるスキーマを表わしている。これらは、GRTBLとGATLLより構成される。SI_RELATION リレーションは、全体リレーション名と、そのタイプと次数とを表わしている。SI_ATTリレーションは、全体リレーションを構成する属性の情報をもっている。表3－5は、NDD／DI／SIのスキーマを表わしている。

表3－5　NDD／DI／SIスキーマ

リレーション名	リレーション・スキーム	属性名	説明
SI_RELATION	SI_RELATION (GLOBAL_RELNAME, ES_RS_TYPE, DEGREE)	GLOBAL_RELNAME ES_RS_TYPE DEGREE	全体リレーション名 ESR or RSR 次数
SI_ATT	SI_ATT (GLOBAL_RELNAME, ES_RS_TYPE GLOBAL_ATT_NAME, TYPE, LENGTH, DOMAIN, ATTRIBUTE_NO)	GLOBAL_RELNAME ES_RS_TYPE GLOBAL_ATT_NAME TYPE LENGTH DOMAIM ATTRIBUTE_NO	全体リレーション名 ESR or RSR 全体属性名 データの型 データの長さ 領域（値集合）名 属性番号

3.2.4 所在情報（NDD／CI／LI）

所在情報は、各サイトのHIより生成される。LI_RELATION リレーションは、局所リレーションとその所在サイト番号（LOCATION_ID）とを示している。LI_ATTリレーションは、局所属性と、その所在サイト番号とを示している。

表3－6 NDD／CI／LIスキーマ

リレーション名	リレーション・スキーム	属性名	説明
LI_RELATION	LI_RELATION	LOCAL_RELNAME	局所リレーション名
	(LOCAL_RELNAME,	ES_RS_TYPE	リレーションタイプ（ESR or RSR）
	ES_RS_TYPE, LOCATION_ID,	LOCATION_ID	サイト番号
	OWNER, DEGREE,		
	NO_OF_KEYS, WIDTH,	OWNER	所有者
	CARDINALITY, SIZE)	DEGREE	次数
		NO_OF_KEYS	キーの数
		WIDTH	組のサイズ
		CARDINALITY	組数
		SIZE	WIDTH* CARDINALITY
LI_ATT	LI_ATT	LOCAL_RELNAME	局所リレーション名
	(LOCAL_RELNAME,	ES-RS-TYPE	リレーションタイプ
	ES_RS_TYRE,	LOCATION-ID	サイト番号
	LOCATION_ID,	LOCAL_ATT_NAME	局所属性名
	LOCAL_ATT_NAME,		
	ROLE, TYPE, LENGTH,	ROLE	属性の役割（key，non-key, c-key）
	DOMAIN, ATTRIBUTE_NO)	TYPE	データの型
		LENGTH	データの長さ
		DOMAIN	領域名
		ATTRIBUTE_NO	属性番号

4　海外の技術動向

4.1　目　　的

今回実施した海外調査の目的は、最新の分散型リソース処理技術、特に分散型データベースシステムを比較評価する技術について研究を行なうために、欧米の先進的研究機関を訪問して、その技術及びデータを収集することにある。

訪問先の候補は次のように数多くのぼった。即ち、UC. Berkeley（INGRESシステム）を始め、IBM San Jose(System－R、SEQUEL)、SRI（LADDER）、CCA（SDD－1)、IBM U. K.（PRTV)、IRIA(SIRIUS)、CNRS(PASCAL)、Univ. of Paris－VI、Univ. of Grenoble（POLYPHEME)、GMD（INGVER、FIDAS)、DIMDI(EURONET)、Univ. of Stuttgart(POREL)、Univ. of Toulouse(SYNTEX－2)、Technical University Berlin、西独Nuclear Research Center(DISCO)が調査の対象に考えられた。

しかし、①分散型データベースを運用もしくは志向していること、②特に関係データベースの研究を行なっていること、の2点を最重要項目にした結果、また、日程上の関係から4.2に示す機関を訪問することに決定した。また同時期に開催される2つの国際会議を日程に組み込むことにより海外調査を充実することができた。4.3で訪問先別及び参加した2つの国際会議の報告を行なう。

4.2　日　　程

表 4－1　海外の技術動向調査日程

月　日	訪問先／国際会議	所在地
8. 2,8	U. C. Berkeley	San Francisco
8. 29 〜 8. 31	the 3rd Berkeley Workshop on Distributed Data Management and Computer Networks	San Francisco
9. 1	SRI	San Francisco
9. 4	CNRS	Paris
9. 5	IRIA	Paris
9. 8	Univ. of Grenoble	Grenoble
9. 11	GMD	Bonn
9. 13 〜 9. 14	the 4th International Conf. on VLDB	West Berlin
9. 15	GLC	London

4.3 調査報告

4.3.1 UC Berkeley

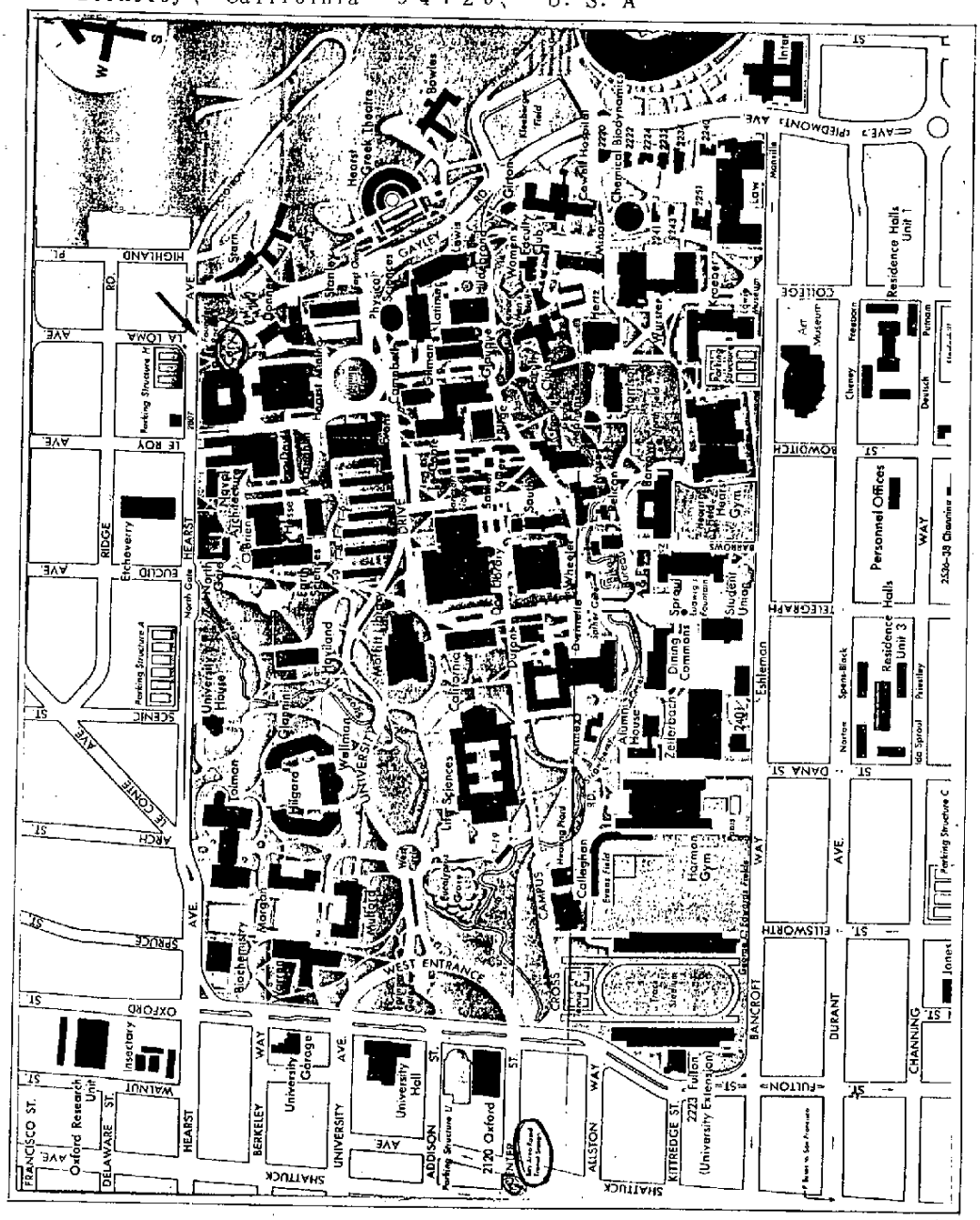
Electronics Research Laboratory

College of Engineering

University of California、Berkeley

1）所 在 地

Berkeley、California 94720、U.S.A



2) 調査目的

UC. Berkeley では Stonebraker教授を中心として、1973年よりリレーショナルデータベース管理システム INGRESを開発し、現在は、これの分散型データベースシステム(DDB)への拡張を試みている。今回の調査は、本プロジェクトの主要な研究課題となっている分散型データベースシステムの問題点である。

i) DDBへのアプローチ(bottom－upかtop－downか,
汎用か専用か)

ii) 共通データモデルと共通アクセス言語

iii) 分散viewの問題

iv) ディレクトリ、concurrency、consistency、問合せ分割

等について、INGRES ではどのように問題解決を試みているかを、議論を通して明らかにすることである。及び、すでに種々のアプリケーションにおいて実用されている(non－distributed) INGRESの実動状況を調査することである。

3) INGRES プロジェクト

Stonebraker教授を中心に、20人近いスタッフによって、データベース一般に関して幅広い研究を行なっている。このなかには、CCAにおける分散型データベースSDD－1のプロジェクトのスタッフとして、問合せの分割に関する研究を行なっている著名なWong博士もいる。研究はPDP－11上にインプリメントしたINGRESを中心に、完全性、プロテクション、問合せ処理等の種々の新しいアイディアを実験を通して、実証的にすすめられている。リレーショナルDBMSに関する研究成果は、SDD－1をはじめ多大な影響をデータベースシステムの開発に与えている。

4) 分散型INGRESのアプローチ

分散型INGRESは、汎用、top－down アプローチをとっている。即ち、論理的なあるリレーションを、ある目的関数のもとで最適なように、いくつかのサイトに分割を行なう。この1つのサイト内に格納されたリレーションの一部をフラグメント(fragment)と呼ぶ。INGRESは、ミニコンピュータ(PDP－11)を用いており、その処理能力の制約から、複数のコンピュータにデータを分散させることによって全体的な処理能力を高めようとしている。即ち、ある1つのリレーションをn個の等しいサイズのフラグメントに分割し、n個のサイトで同時に検索すれば、理想的には、1つのリレーションを検索するよりもn倍速くなる。さらに、冗長コピーを設けることによって、フェイルソフトと検索効率の向上をはかることができる。このような動機から、分散型INGRESへの拡張を試みている。

このような点からINGRESでは、異種のDBMS(例えばIMS、DBTG)の統合は考えていない。又、現在PDP－11では、このようなDBMSをサポートしていないことも理由になっている。

5) データモデル

INGRESは、リレーショナルDBMSであることから、共通データモデルはリレーショナル

モデルである。INGRESがリレーショナルモデルを採用した理由は、このモデルがデータの記述とデータのアクセスの簡潔さ（simplicity）を備えていることである。この簡潔さは、強調されるべきである。現在種々のデータモデル（例えば、Abrial の data semantics、Chen のE－Rモデル etc）が提案されているが、これらはセマンティックスをモデル内で表わそうとするものである。データモデルとセマンティックスとにおける重要な点は、モデルが多くのセマンティックスを持てば、一方ではモデルの複雑化をもたらしてしまうことである。又、モデルがセマンティックスを持たず抽象代数に基づくならば、いわゆるセマンティックス問題（意味のあいまいさ）をもたらすが、モデルは非常に簡単なものとなる。Stonebraker教授の意見は、セマンティックスとは、極めて幅が広くかつ多いので、データモデルに持ちこんだとしても表わしきれるものではなく、データモデル外の規則（rule）として存在すべきであり、データモデルの簡潔さが一番重要であるとのことであった。よって、他のモデルの検討はあまり行なわれていない。この教授の指摘は、我々のFSS概念にも重要な影響を与えている。

関係計算に基づいたアクセス言語（QUELという）と正規化されたリレーションからなるINGRESは、完全な（complete）なリレーショナルDBMSである。即ち、リレーションとして格納されたどのデータも、QUELによって新たなリレーションとしてあいまいさなくとり出せる。この意味で、リレーショナルDBMSは、データベース・システムの核となるものである。実際、SDCにおいて開発されているEUFID(End-User Friendly Interface to Data Management)は、INGRESを核とした自然言語による問合せシステムである。このように、データのセマンティックスは、構文的に完全なリレーショナルDBMSを核として、表現されていこうとしているようにみえる。

6）言　語

INGRESは、非手続的な問合せ言語QUEL、手続的なEQUELをサポートしている。学生はQUELをよく用い、ルーティンワークにはEQUELが用いられている。簡単な問合せには、非手続言語は適しているが、実際のアプリケーションのもとで複雑な使用（sophisticate use）又はルーティンワークには、手続的言語EQUELが適しているようである。

7）分散view

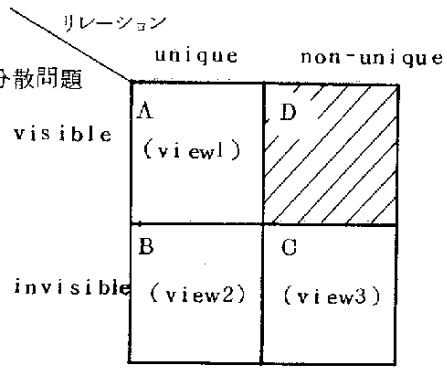


図4－1　INGRESの分散view

INGRESにおけるview（1～3）は、図4－1のようにまとめなおせる。縦軸は、分散問題（即ち、必要なリレーションがどこに存在するかという問題）が可視か（visible）か不可視（invisible）かを示している。横軸は、リレーションが1つのサイト内にあるか（unique）、又はいくつかのサイトにまたがっているか（non-unique）を示している。斜線をひいたDに対するview（4）は考えていない。

8）分散機能

INGRESにおけるリレーションの分散の単位（即ち、fragment）は、tupleである。フラグメントは、リレーションの制限（restriction）であり、射影（projection）ではない。リレーションの意味論的理由から、tuple単位としている。従って、冗長の単位もtupleである。リレーションの分散は、分散criteriaとして関係計算形式で定義されることができる。

ディレクトリは、リレーション形式で管理されている。ディレクトリへのアクセスは、通常のユーザリレーションに対するのと同様に行なえる。

DDBにおいて、一致性（consistency）制御は、冗長コピーが存在する場合に特に重要である。INGRESでは、冗長コピーに対しては主コピー（primary copy）を1つもうけている。検索は、一番近いコピーに対してなされるが、更新は、必ず主コピーに対してなされる。主コピーの更新中は、他のコピーをlockするものである。この前提には、更新は局所的であるということがある。このアルゴリズムは、この週に開催された第3回Berkeley Workshopで発表された。

9）INGRESのデモンストレーション

現在INGRESは、ERLのPDP－11／70上にインプリメントされている。我々は、E. Allman博士によるINGRESのデモンストレーションを見ることができた。Harvard大学、New York電話会社等で実用されているだけあって、機能的に（完全性、プロテクション等）完備されたものであった。実際のリレーショナルモデルの実働をみれたことは、我々にとっても初めてのことであり大きな収獲であった。感想としては、リレーショナルモデルは、そのパフォーマンスの点で種々問題が指摘されているが思った以上にレスポンスはよかった。（実際リレーションが小さいこともあるが）。デモの例を図4－2に示す。

10）ま と め

現在、UC. Berkeleyの計算センタにある2台のPDP－11とERLのものとを回線で結合して、分散型データベースシステムの実験を行なおうとしているが、政治的問題があって結合されていない。よって、分散型INGRESに関しては、現在設計中であるといえることができる。DDBにおいて、そのパフォーマンスの問題が種々指摘されるが、現在はDDB構築の可能性を示そうとしている時であるように思えた。このために、問合せの分割のアルゴリズム、冗長コピーの更新アルゴリズムの研究が行なわれている。DDBが従来の集中型DBに対して、どのような長所と欠点をもつかは、これらの実験システムが実動してから大きな議論をもたらすだろう。INGRESグループの特徴は、常に実証的に研究をすすめていることにあり、理論だけを追求したり、インプリメントだけを目的としていない姿勢は貴重な教訓となった。

```
>@eric
Password:
INGRES UNIX 11/70
System will be down 08:00-10:00, Tues., 8/22 for disk maint.
System will be down from 5-7 am Sat.,Sun.,Tues.,&Thurs..
Erase set to control-H
# ingres demo
INGRES version 6.1/10 login
Mon Aug 28 15:23:47 1978
go
* range of e is employee          } eとdを各々リレーション employee
* range of d is dept              } とdeptの組変数とする.
* help employee
* helpd dept
* \g                              ← go
Executing . . .


Relation:                 employee
Owner:                    ingres
Tuple width:              30
Saved until:              Tue Jan  1 00:00:00 1980
Number of tuples:         27
Storage structure:        paged heap
Relation type:            user relation

attribute name    type  length  keyno.

birthdate          i       2
manager            i       2
name               c      20
number             i       2
salary             i       2
startdate          i       2

employee リレーションについての情報


Relation:                 dept
Owner:                    ingres
Tuple width:              28
Saved until:              Tue Jan  1 00:00:00 1980
Number of tuples:         19
Storage structure:        paged heap
Relation type:            user relation

attribute name    type  length  keyno.

floor              i       2
manager            i       2
name               c      20
number             i       2
store              i       2

deptリレーションについての情報
```

図4－2　INGRESのデモンストレーション(1)

employee リレーション

number	name	salary	manage	birthd	startd
157	Jones, Tim	12000	199	1940	1960
1110	Smith, Paul	6000	33	1952	1973
35	Evans, Michael	5000	32	1952	1974
129	Thomas, Tom	10000	199	1941	1962
13	Edwards, Peter	9000	199	1928	1958
215	Collins, Joanne	7000	10	1950	1971
55	James, Mary	12000	199	1920	1969
26	Thompson, Bob	13000	199	1930	1970
98	Williams, Judy	9000	199	1935	1969
32	Smythe, Carol	9050	199	1929	1967
33	Hayes, Evelyn	10100	199	1931	1963
199	Bullock, J.D.	27000	0	1920	1920
4901	Bailey, Chas M.	8377	32	1956	1975
843	Schmidt, Herman	11204	26	1936	1956
2398	Wallace, Maggie J.	7880	26	1940	1959
1639	Choy, Wanda	11160	55	1947	1970
5119	Ferro, Tony	13621	55	1939	1963
37	Raveen, Lemont	11985	26	1950	1974
5219	Williams, Bruce	13374	33	1944	1959
1523	Zugnoni, Arthur A.	19868	129	1928	1949
430	Brunet, Paul C.	17674	129	1938	1959
994	Iwano, Masahiro	15641	129	1944	1970
1330	Onstad, Richard	8779	13	1952	1971
10	Ross, Stanley	15908	199	1927	1945
11	Ross, Stuart	12067	0	1931	1932

```
continue
* retrieve (e.number, e.name)
* where e.salary > 12000
* \g
Executing . . .
```

給料が12,000ドル以上の社員の番号と名前をもとめよ

number	name
10	Ross, Stanley
11	Ross, Stuart
843	Schmidt, Herman
994	Iwano, Masahiro
1523	Zugnoni, Arthur A.
5119	Ferro, Tony
55	James, Mary
1639	Choy, Wanda
26	Thompson, Bob
430	Brunet, Paul C.
37	Raveen, Lemont
157	Jones, Tim
129	Thomas, Tom
199	Bullock, J.D.
5219	Williams, Bruce
1000	Ries, Dan

この問合せ結果

図4－2　INGRESのデモンストレーション(2)

```
xxx relation

|ename                |dname                |
|---------------------|---------------------|
|Edwards, Peter       |Candy                |
|Evans, Michael       |Toys                 |
|Hayes, Evelyn        |Jewelry              |
|Hayes, Evelyn        |Stationary           |
|James, Mary          |Book                 |
|Jones, Tim           |Linens               |
|Ross, Stanley        |Children's           |
|Ross, Stanley        |Sportswear           |
|Ross, Stanley        |Women's              |
|Smythe, Carol        |Children's           |
|Smythe, Carol        |Women's              |
|Thomas, Tom          |Junior Miss          |
|Thomas, Tom          |Men's                |
|Thompson, Bob        |Furniture            |
|Thompson, Bob        |Major Appliances     |
|Williams, Judy       |Giftwrap             |

continue
* range of x is xxx
* delete x where x.dname = "Candy"
* \g
Executing . . .

* print xxx
* \g
Executing . . .

xxx relation

|ename                |dname                |
|---------------------|---------------------|
|Evans, Michael       |Toys                 |
|Hayes, Evelyn        |Jewelry              |
|Hayes, Evelyn        |Stationary           |
|James, Mary          |Book                 |
|Jones, Tim           |Linens               |
|Ross, Stanley        |Children's           |
|Ross, Stanley        |Sportswear           |
|Ross, Stanley        |Women's              |
|Smythe, Carol        |Children's           |
|Smythe, Carol        |Women's              |
|Thomas, Tom          |Junior Miss          |
|Thomas, Tom          |Men's                |
|Thompson, Bob        |Furniture            |
|Thompson, Bob        |Major Appliances     |
|Williams, Judy       |Giftwrap             |
```

xxxリレーション内のdnameが"Candy"であるtupleをdeleteする

この更新結果

図4－2　INGRESのデモンストレーション (3)

```
replace x (ename = "Smith, Carol")
where x.ename = "Smythe, Carol"
continue
* \g
Executing . . .
```

xxxリレーションの
enameが"Smythe,Carol"である
tupleのenameを
"Smith, Carol"に
replaceする。

```
continue
* print xxx
* \g
Executing . . .
```

xxx relation

ename	dname
Smith, Carol	Children's
Evans, Michael	Toys
Hayes, Evelyn	Jewelry
Hayes, Evelyn	Stationary
James, Mary	Book
Jones, Tim	Linens
Ross, Stanley	Children's
Ross, Stanley	Sportswear
Ross, Stanley	Women's
Smith, Carol	Women's
Thomas, Tom	Junior Miss
Thomas, Tom	Men's
Thompson, Bob	Furniture
Thompson, Bob	Major Appliances
Williams, Judy	Giftwrap

図４－２　ＩＮＧＲＥＳのデモンストレーション (4)

```
continue
* retrieve (e.salary, e.manager)
* \g
Executing . . .
```

salary	manage
17498	199
9656	13
13273	0
13556	26
10950	199
17205	129
9534	26
9900	199
11110	199
24039	129
14983	55
7260	33
5500	32
13200	199
10441	10
12276	55
9214	32
15730	199
19441	129
13183	26
13200	199
10890	199
12100	199
29700	0
16182	33
19264	0
10	0

```
continue
* retrieve (e.manager, av = avg(elsalary by e.manager))
* \p
retrieve (e.manager, av = avg(e.salary by e.manager))
continue
* \g
Executing . . .
```

マネージャーと、その部下の給料の平均値をもとめる。avgは、aggregate関数

manage	av
0	15561.750
10	10441.000
32	7357.000
13	9656.000
33	11721.000
55	13629.500
26	12091.000
199	12730.888
129	20228.333

図4－2　INGRESのデモンストレーション (5)

4.3.2 ＳＲＩ．スタンフォード研究所

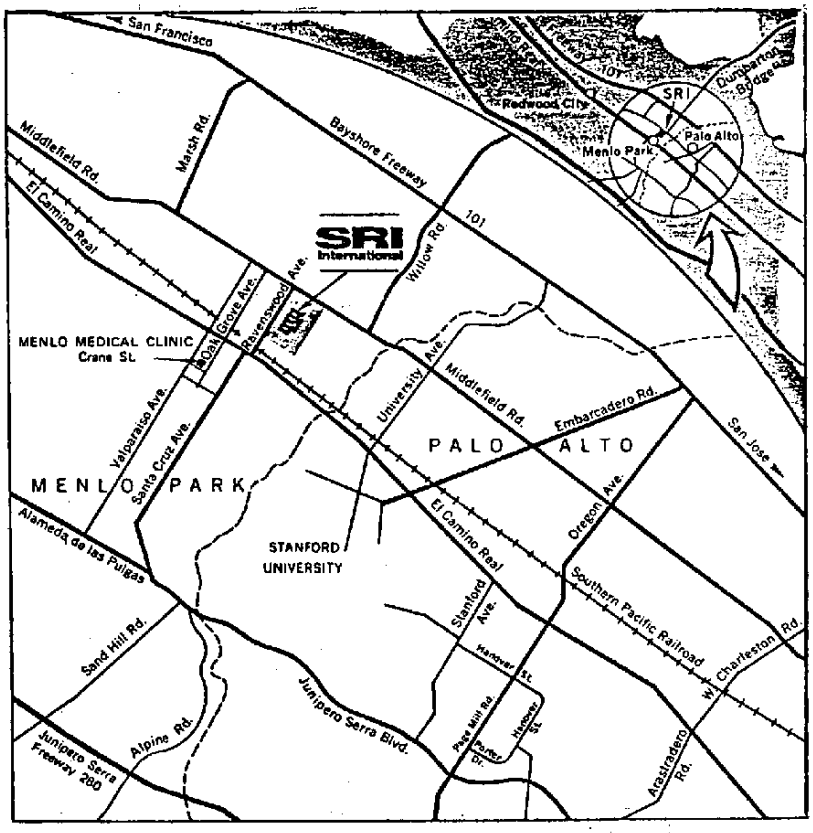
Artificial Intelligence Center, Stanford Research Institute, Internatinal

1) 所　在　地

333　Ravenswood Avenue Menlo Park,

California　94025, U. S. A.

Tel : 415 − 326 − 6200



2) 調査目的

ＳＲＩでは、分散型データベースシステムＬＡＤＤＥＲを開発している。 現時点では、同種のＤＢＭＳ（データコンピュータ）のみを サポートしているが、将来は異種のＤＢＭＳ もサポートする計画にある。このシステムは自然言語（英語、将来は仏語も）を入力としており、データベースの分野に留まらず人工知能（ＡＩ）の分野からも注目されている。今回は、このＬＡＤＤＥＲの現状と問題点、将来計画等についての調査を目的とした。

3) LADDERシステム

① 現　状

LADDER（Language Access to Distributed Data with Error Recovery）は、SRIにおいて開発された米国海軍のための分散型データベースシステムである。LADDERの目的は、一般ユーザに対して、種々のDBMSの管理下で複数のコンピュータ上に格納された情報への容易なアクセス手段を提供することである。SRI／AIセンターには2台のKL－10（PDP－10）があり、キャッシュメモリ付きのKL－10上にLADDERをインプリメントしている。

データベースは、CCAとSRIに海軍関係の情報（船等）を格納したデータコンピュータ（datacomputer）があり、関係データベースで管理している。データコンピュータは、PDP－10上のDBMSであり、CCAでは、専用の大規模メモリterabitを開発している。現在LADDERがARPA－NET経由で使用できるデータコンピュータは、CCA、SRI、USC、NOSCの4台である。これらのデータベースは、up－to－dateなものでなく、データベースの更新は各データベースサイトのDBAによって行なわれている。

今日のLADDERシステムを構築するのにINTERLISPを用いて約5人年の工数で実現しており、現在でもこのプロジェクトに10人のスタッフが配属され、5人が専任となっている。なお、参考までにSRIのAIセンターには約50人が研究に従事しているとのことである。

② アーキテクチャ

図4－3にLADDERの全体概念図を示す。LADDERは、このようにINLAND、IDA、FAMの独立した3つの要素から成る。以下、各要素の概要を示すが、詳細は前年度の報告書〔52–S001〕及び参考文献を参照されたい。

現在LADDERがサポートできるDBMSはデータコンピュータだけであるが、PDP–10用のDBMS20（DBTGタイプ）への拡張に対する検討が行なわれている。

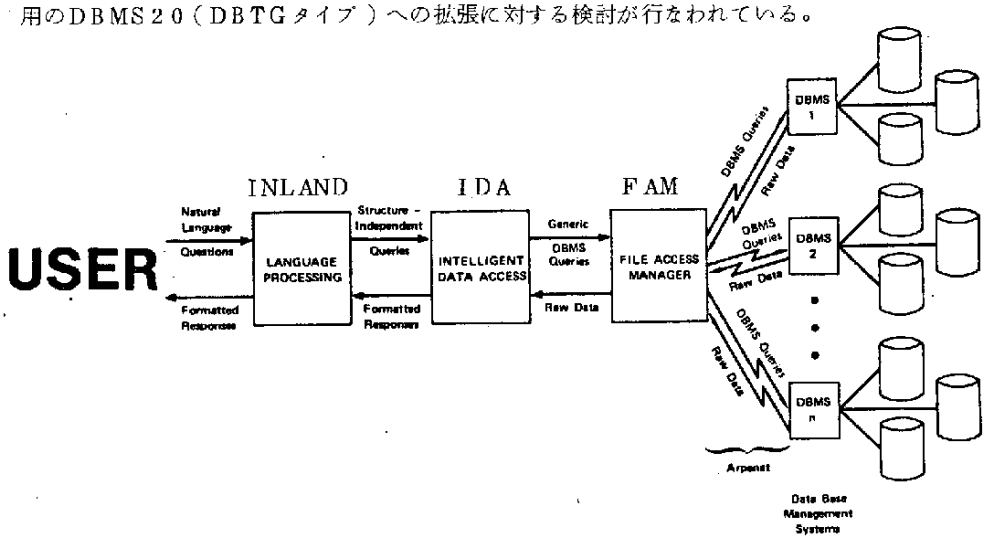


図4－3　LADDERの全体概念図

INLAND

INLAND(Information Natural Language Access to Navy Data)は、LADDERにおける第1番目の要素であり、コンピュータネットワーク上に分散したデータに対して自然言語に近い形でのアクセスを提供する知的インタフェースである。INLANDは、LIFFER(Language Interface Facility with Ellipsis and Recursion)と呼ばれる自然言語インタフェースを用いている。LIFFERは、他のコンピュータソフトウェアへの自然言語インターフェース生成機能を持っている。INLANDは、このような自然言語問合せを入力として、ネットワーク上に分散したデータベース全体に対するひとかたまりの高水準形式的問合せ群を生成する。生成される問合せは、必要なデータ項目名だけの情報を持ち、項目の構成方法、即ち情報構造に関する情報を何等持たない。

現在SRIでは仏語を入力としたFrench INLANDの開発を計画している。このようにSRI/AIセンターにおけるLADDERは、AIの立場から研究を進めており、興味の対象は容易なインタフェースにある。

IDA

LADDERにおける第2の要素であるIDA(Intelligent Data Access)(aidaと読む)は、データ項目名だけで、データベースからデータベースの構造とは独立に情報を検索できる機能を有している。IDAは、項目名だけの問合せにデータベースの構造を付加して実際のDBMS問合せを生成する。生成される問合せは、一般DBMS問合せ(generic DBMS query)と呼ばれ、CCAにおいて開発されたデータコンピュータのアクセス言語、データランゲージ(datalanguage)である。IDAでは、データの存在するファイルを一般ファイル名(generic file name)で参照する。1つの一般ファイル名は、主ファイル及び冗長コピー(バックアップ)全体の総称であり実際のファイルの所在情報は付加されていない。所在情報は、第3の要素FAMによって明らかになる。IDAはさらに一般DBMS問合せの実行結果をユーザの要求に合うような結果の生成も行なう。このようにLADDERにおける統一モデルは、IDAとINLAND(ユーザ)とのインタフェースに存在すると考えられる。図4－4に現在のIDAのアーキテクチャを示す。

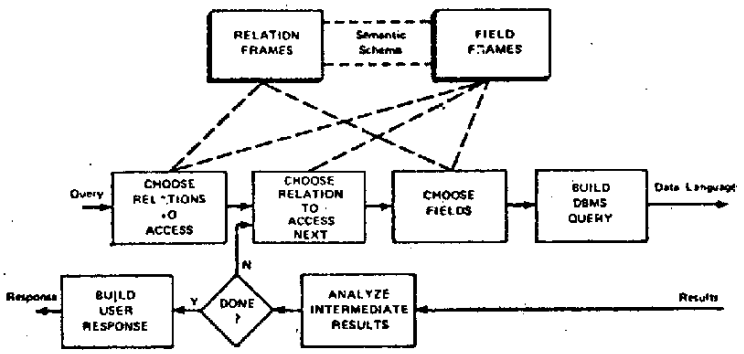


図４－４　現在のＩＤＡアーキテクチャ

現在、異種のＤＢＭＳをサポートするために図４－５に示すようなＩＤＡ（new－ＩＤＡ）の開発に取組んでいる。即ち、ＩＮＬＡＮＤからのone－relation query （項目名だけからなる問合せ）を実際のリレーションへのquery(multi－relation query：複数リレーションにまたがる問合せ）に変換するのを従来のＩＤＡ（old－ＩＤＡ）が行なう。このqueryから異種のＤＢＭＳへの変換をnew－ＩＤＡが行なう。現在検討している異種のＤＢＭＳはＤＢＴＧタイプのＤＢＭＳ２０であり、ＤＢＭＳ２０のレコードタイプをリレーションに対応させている。

one－relation qneryからmulti－relation queryへの変換は、マルチパス(multi－path)の問題を生じさせる。この問題はセマンティックス（semantics）の問題も絡んでおり難しい問題である。このためone－relation queryも幾つかのリレーションから成るqueryに変更する動きも見られる。

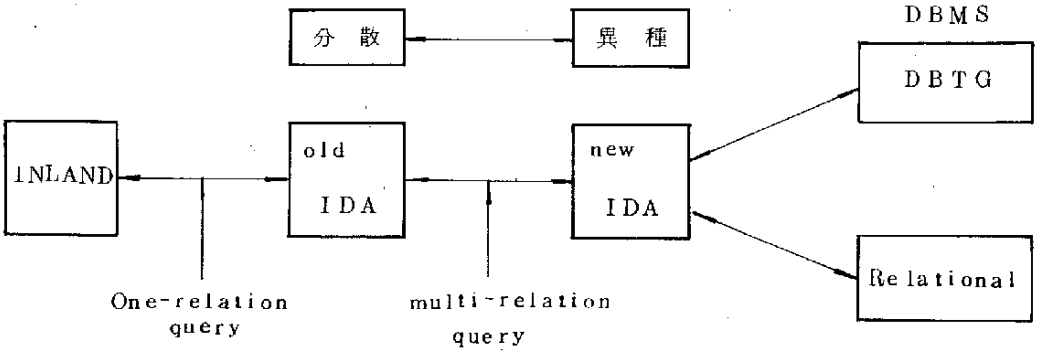


図４－５　研究開発中のnew ＩＤＡ

FAM

LADDERにおける第3番目の要素であるFAM(File Access Manager）の目的は、ユーザとプログラムに対して、データベースの非本質的事項（inessential details)を解放するとともに分散型データベースに信頼性の高いアクセスを提供することである。FAMのユーザ（IDAも含む）は、一般ファイル名によってファイルの指定が可能である。FAMの主要な役目は、コンピュータネットワーク上のファイルの所在を明らかにし、実際のネットワークのデータベースサイトに向けて問合せを発進することである。ファイルのオープン、処理の実行、ログイン等をつかさどったり、主ファイル及び冗長コピーの所在を知っているFAMは、CCAによって開発された分散型データベースシステムSDD－1と同様の機能を有している。現在FAMはCCAにおいて開発されたデータコンピュータのみをサポートしているが、PDP－10用のDBMS20への拡張に対する検討が行なわれている。

4) デモンストレーション

今回訪問した際行なわれたデモンストレーションの結果の一部を図4－6に示す。なお、PDP－10の端末は、SRI内に90台が設置されており、このデモンストレーション中には65台の端末が稼働していた。また、図4－7はLADDERシステムが受け入れ得る問合せの例を示したものである〔HEND78〕。

5) ま と め

現在のLADDERは、ユーザがデータベースの在るサイトを指定しなければならないが、開発中のnew－LADDER（主にIDA）では、サイトを意識しなくてよいように改良されている。また、異種のDBMS（DBMS20－DBTGタイプ）のサポートも計画しており、異種のDBMSから成る分散型データベースシステムとしての稼働も含め、エラー回復機能の充実やFrench INLANDの開発、より説明的(explanatory)な解答の実現等多くの課題があるが、それらの多くは近い将来実現されることになろう。

LADDERでは、データベースの更新等のDDBの制御問題を検討していないが、これは、目的が容易なユーザインタフェースをAIの立場から開発することだからである。やはり、LADDERの核としては、リレーショナルモデルを用いている。

LADDERとの対話開始（'78.9.1. 9:57AM）

ユーザ名：SAGALOWICZ

FOX(船名)の長さは？ (INLAND)

(IDA)

(FAM:以下省略)

(Datalanguage)

FOXの長さは＝547(フィート)

```
[PHOTO:  Recording initiated  Fri 1-Sep-78 9:57AM]

LINK FROM SAGALOWICZ, TTY 105

 TOPS-20 Command processor 1018(117)
@LADDER.EXE.2

        Language Access to Distributed Data with Error Recovery
                        -- SRI International --

Please type in your name:
" 3-Aug-78 16:54:03"
Do you want instructions? (type FIRST LETTER of response)
No
Do you want to use 2 Data Computers? ...No
Do you want to specify a current location (default = Norfolk)?  No
Do you wish distance/direction calculations to default to GREAT CIRCLE,
or RHUMB LINE? (you can override by specifying in the query) ...Great Circle

1_WHAT IS THE LENGTH OF THE FOX
PARSED!
Parse time: .462 seconds

For SHIP equal to FOX, give the LEN.

 Connecting to Datacomputer at CCA:
>> ;0031 780901170604   IONETI: CONNECTED TO SRI-KL-16700010
>> ;J150 780901170605   FCRUN: V="DC-5/01.00.13" J=4 DT="FRIDAY, SEPTEMBER 1,
                                                  ** 19
78 13:06:05-EDT" S="CCA"
>> ;J200 780901170606   RHRUN: READY FOR REQUEST

*> Set parameters
*<      Exit
CCA: -Z
*> Set parameters
*< V    Verbosity (-1 to 5):   2
*< P    PROCEED with Datalanguage  [confirm with <CR>]
CCA: LOGIN %TOP.ACCAT.GUEST ;
CCA: OPEN %TOP.ACCAT.SAGALOWICZ.NSTDPORT1 WRITE;
CCA: OPEN %TOP.ACCAT.SHIP READ;
CCA: FOR NSTDPORT1 , SHIP WITH (NAM EQ "FOX") BEGIN STRING1 = LGHN END;
*> Total bytes transferred: 9
Computation time for query: 2.509 seconds.
Real time for query: 158.020 seconds
LEN = 547
```

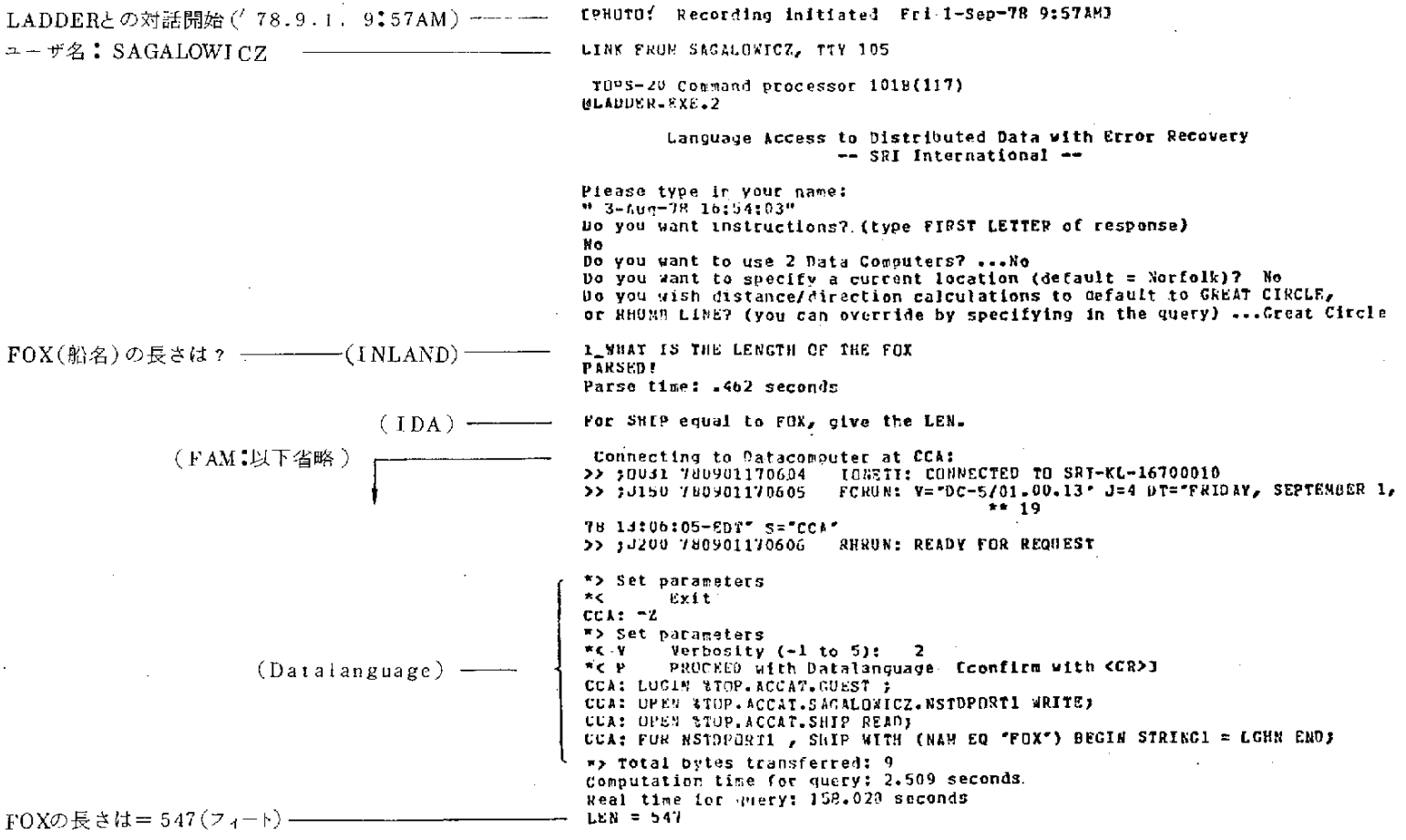


図４－６　ＬＡＤＤＥＲのデモンストレーション(1)

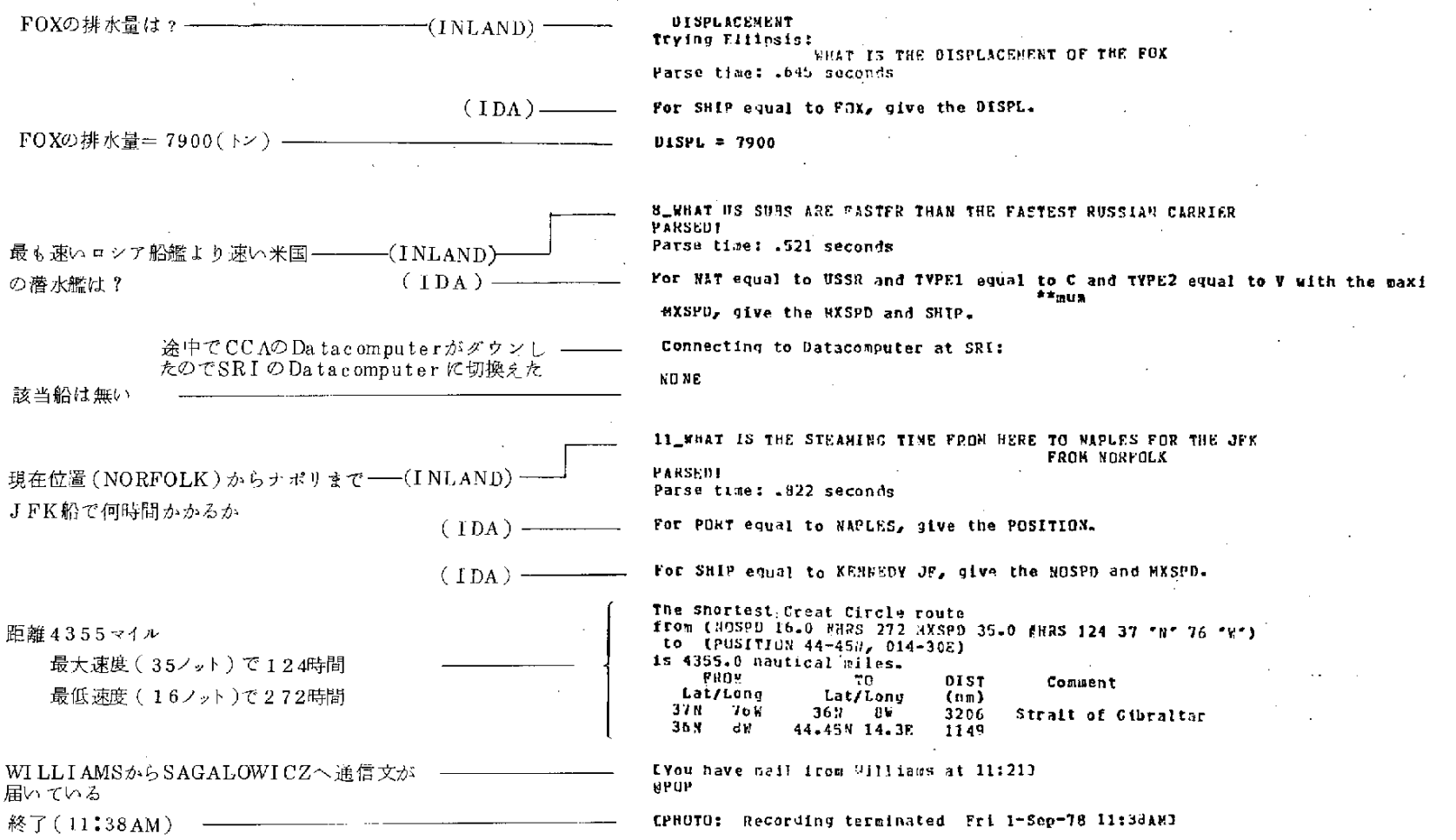


図4－6　LADDERのデモンストレーション(2)

What kind of information do you know about
Is there a doctor on board the Biddle
Display all the American cruisers in the North Atlantic
What is the name and location of the carrier nearest to New York
What is the commanding officer's name
Who commands the Kennedy
What is the Kennedy's beam
When will the Los Angeles reach Norfolk
Tell me when Taru is scheduled to leave port
Where is she scheduled to go
When will Los Angeles arrive in its home port
When will the Sturgeon arrive on station
What aircraft units are embarked on the Constellation
To which task organization is Knox assigned
Where is the Sellers
Where is Luanda
What is the next port of call of the Santa Inez
When will Tarifa get underway
Which convoy escorts have inoperative sonar systems
When will they be repaired
Which U.S. Navy DDGs have casreps involving radar systems
What Soviet ship has hull number 855
To what class does the Soviet ship Minsk belong
What class does the Whale belong to?
What is the normal steaming time for the Wainwright from Gibraltar to Norfolk
What American ships are carrying vanadium ore
How far is it to Norfolk
How far away is Norfolk
How many nautical miles is it to Norfolk
How many miles is it to Norfolk from here
How close is the Baton Rouge to Norfolk
How far is the Adams from the Aspro
What is the distance from Gibraltar to Norfolk
What is the nearest oiler
What is the nearest oiler to the Constellation
How far is it from Naples to 23-00N, 45-00W
What is the distance from the Kitty Hawk to Naples
How long would it take the Independence to reach 35-00N, 20-00W
How long is the Philadelphia
How long would it take the Aspro to join Kennedy
What is the nearest ship to Naples with a doctor on board
What is the nearest USN ship to the Enterprise with an operational air search radar
What is known about that ship
How many merchant ships are within 400 miles of the Hepburn
What are their identities and last reported locations
What cargo does the Pecos have
Who is CTG 67.3
What are the length, width, and draft of the Kitty Hawk
To whom is the Harry E. Yarnell attached
What type ships are in the Knox class
Where are the Charles F. Adams class ships
What are their current assignments

図4－7　Sample of acceptable inputs to LADDER

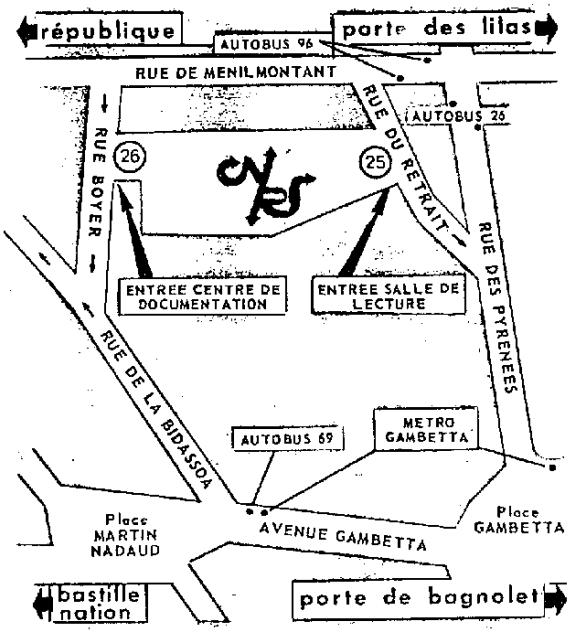
4.3.3 CNRS 科学技術情報センター

Centre National de la Recherche Scientifique、Centre de Documentation Scientifique et Technique、Informascience

1) 所 在 地

26, Rue Boyer - 75971 Paris Cedex 20; France

Tel : 797. 35. 59



SALLE DE LECTURE : ENTRÉE 25, RUE DU RETRAIT
Métro : Gambetta Autobus : 26 - 69 - 96

2) 調 査 目 的

CNRSではPASCALファイルを用いたオンライン検索システムPASCALINE（PASCAL on LINE）を運用している。PASCALINEは伊のFrascatiのESA/SDSにおける一連の流れと、仏のGrenoble大学における実験の2つの系統がある。また、欧州には科学技術情報の交流を目指したEURONETが存在し、仏国内をみてもIRIAが中心に開発したCYCLADESがある。このようなネットワーク環境にあるCNRSにおける分散型データベースシステムの調査を第一の目的とし、さらには仏の国家プロジェクトSIRIUSにおけるCNRSの役割も合わせて調査することを目的とした。

3) CNRS Informascienceの組織図

PASCALシステムの改良を続けているCDSTは、現在500名を超えるスタッフを擁している。ただし約半数が外部協力者または自由契約の研究員である。

INFORMASCIENCE

CENTRE DE DOCUMENTATION
SCIENTIFIQUE ET TECHNIQUE DU C.N.R.S.

26, rue Boyer — 75971 PARIS CEDEX 20
Tél: 797.35.59 — Télex: CNRSDOC 220880

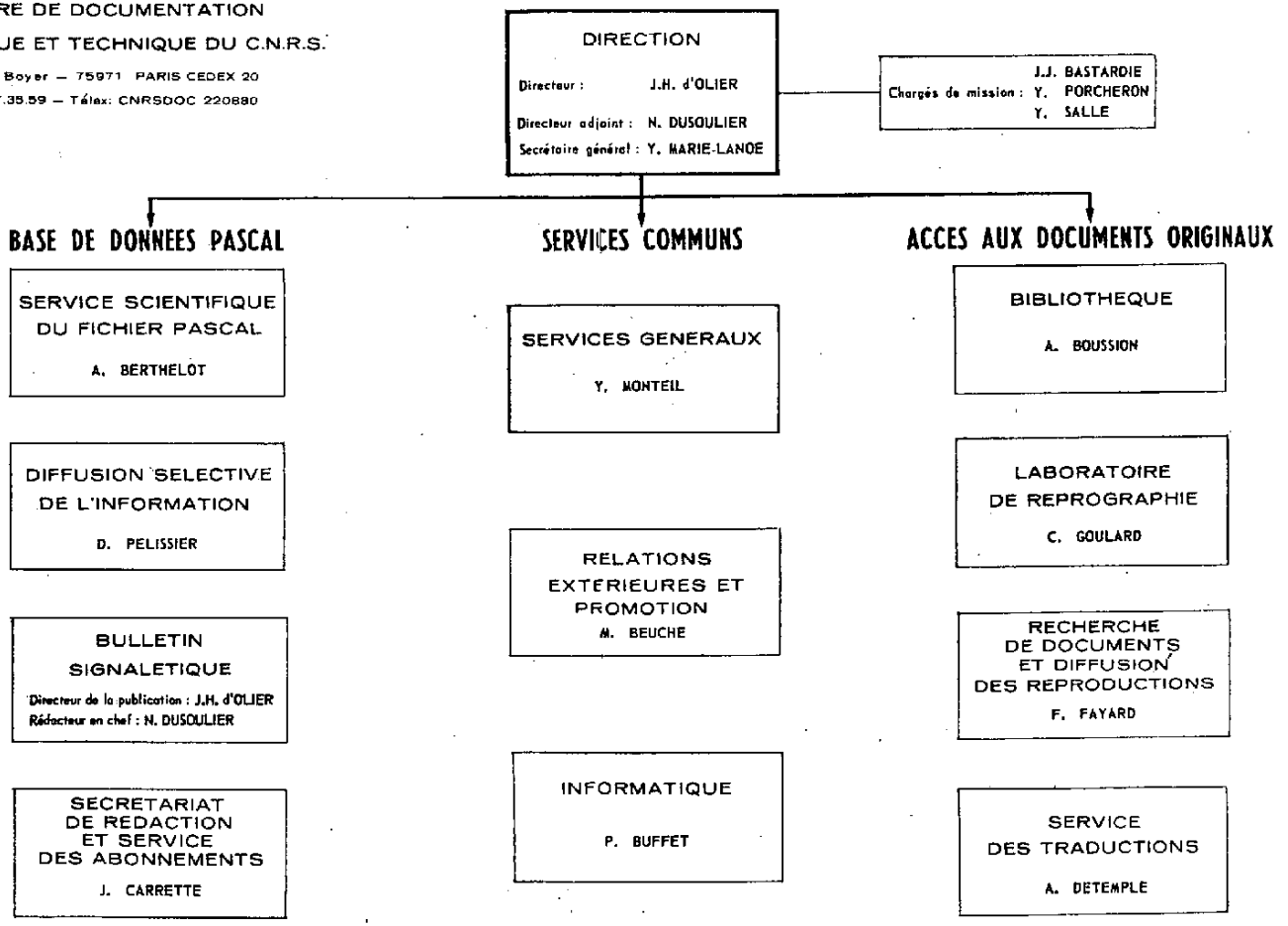


図4－8　CNRS／CDSTの組織図

4) ＣＮＲＳとＰＡＳＣＡＬシステム

ＣＮＲＳは１９３９年国家的援助により設立された。ＣＤＳＴも同時に設置されＣＮＲＳの科学者への情報提供を目指した。今日ではＣＮＲＳの情報処理サービスのほぼ全般を指してＰＡＳＣＡＬと呼んでいるが、本来は科学技術文献の機械処理に対する名称であった。　ＰＡＳＣＡＬ（Programme Appliqué à la Sélection et a la Compilation、Automatiques de la Literature ── Program Applied to the Automatic Selection and Compilation of the Literature）システムは、１９７１年にＰＡＳＣＡＬ－１として稼働しＰＡＳＣＡＬ－２を経て現在のＰＡＳＣＡＬ－３に至っている。

機械化の始まりは１９６２年の機械可読な「ＢＳ」作成の機械化実験計画であり、１９６８年には写植装置を導入している。「ＢＳ」の出版は１９６９年から１９７３年にかけて全部門の機械化が完了した。検索システムの機械化は１９７０年のグルノーブルにおけるＭＩＳＴＲＡＬ－１に端を発している。

5) ＣＮＲＳの提供するサービス

ＣＮＲＳの実施している情報提供サービスの中心は抄録誌「ＢＳ」の提供である。現在「BS」は約５０部門に渡っている（表 4－2 参照）。この他次のようなサービスを行なっている。

・ＰＡＳＣＡＬテープの提供

ＢＳの分類に従って作成しており、その提供先はポーランドを始めカナダ、スウェーデン、米国、ブラジルとなっている。

・シソーラスの提供

分野別で年１～２回の更新を行なっている。

・翻訳サービス

筆記、口述、テープへの吹込みによる翻訳。

・書誌目録リストの編集

・閲覧・複写サービス

・検索サービス

ＲＳサービス（遡及検索）

ＳＤＩサービス（選択情報提供サービス）

表４－２　Ｂ Ｓ 50 部 門 の 一 部

Bulletin Signaletique	
部門No.	部　　門
101	情報科学、ドキュメンテーション
110	コンピューター科学、オペレーションリサーチ、マネージメント
120	天文学、宇宙物理学、地球物理学
130	物理学
140	エレクトロテクニクス
145	エレクトロニクス
160	固体物理学
161	結晶学
165	原子および分子物理学
170	化学
310	生物医学工学およびコンピュータ科学
320	生物化学、生物物理学
330	薬理学、毒物学
340	微生物学、ビールス学、免疫学
761	電子顕微鏡、電子回析
780	高分子
880	化学工学、化学工業および異性体化学工業
885	公害
890	機械工学、建築工学、土木工学、輸送
900	翻訳誌

6）ＣＮＲＳと分散型データベース

ＣＮＲＳにおいては汎用ＤＢＭＳを用いた分散型データベースの研究は行なわれておらず、ＥＵＲＯＮＥＴに見られるデータの共有に精力を注いでいる。現在、ＥＳＡ／ＳＤＳ（伊、Frascati）にＰＡＳＣＡＬデータベースを配置してＥＵＲＯＮＥＴへの参加を実現している。現在運用されているデータベースは全ＰＡＳＣＡＬファイルの一部にすぎないが、全ファイルを用いて運用するのも近いといわれている。図４－９はＰＡＳＣＡＬＩＮＥの接続形態を示し、図４－10にＰＡＳＣＡＬファイル（分野：エレクトロテクニクス）を用いた検索結果を示す。

なお、現在Niceでの ＣＮＲＳのデータベースの整備（センター化）が進んでいる。

comment se connecter à PASCALINE ?

Par l'intermédiaire de trois réseaux accessibles par téléphone :

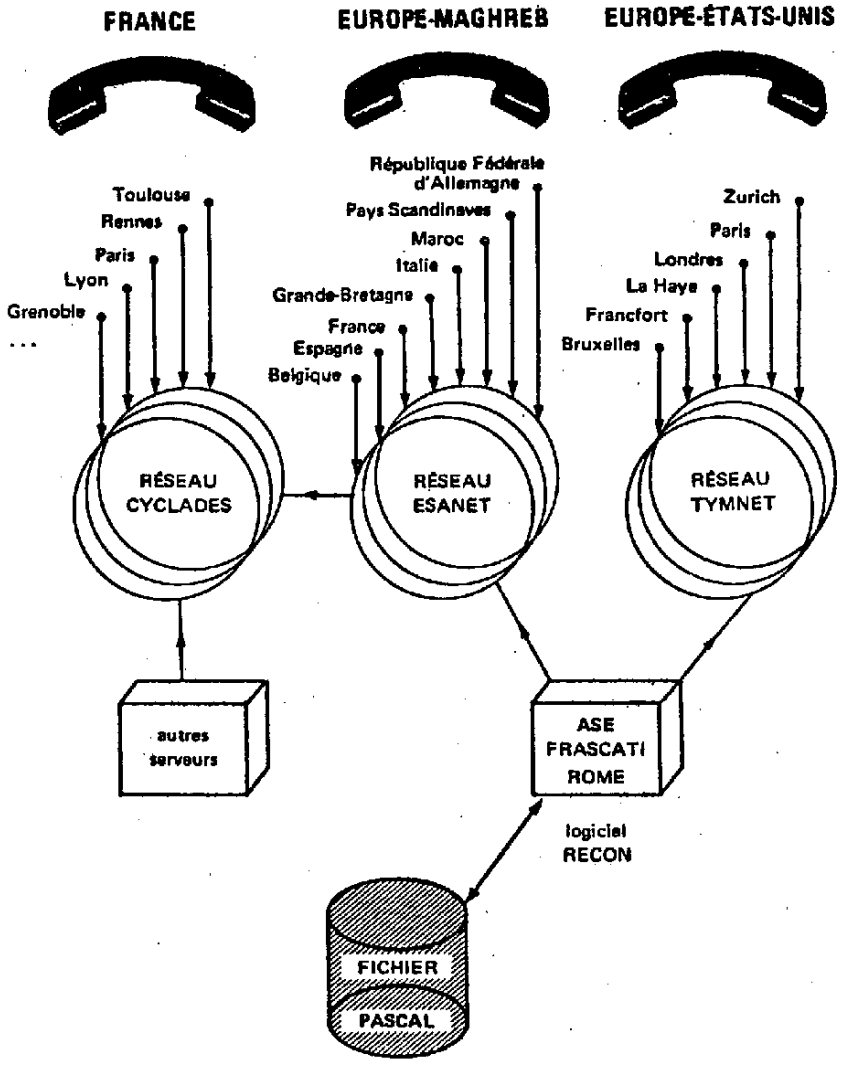


図4－9　PASCAL INEへのアクセスパターン

入力キーワード ＝ 〝JAPAN〟 AND 〝COMPUTER NETWORK〟

```
19/4/1-5

78-3-00406621 P.A.S.C.A.L. 1381295
  NETWORKING FOR INTER-UNIVERSITY COMPUTER CENTRES IN JAPAN.
  INOSE (H.); ASANO (S.); SAKAI (T.); KATO (M.)
  AFF: UNIV. TOKYO, JAP.
  IN:  ADV.  THROUGH  RESOUR. SHARING. INT. CONF. COMPUT. COMMUN. 3.
PROC.;  TORONTO;  1976;  WASHINGTON,  D.C.;  Ed: INT. COUNC. COMPUT.

COMMUN.; Date: 1976; p.: 286-291; 4 REF.; Cote: Y013194;
  CC: 110.C.04.G.02
  DS:  TRANSMISSION  DONNEE  /  RESEAU  TRANSMISSION DONNEE / RESEAU
ORDINATEUR  /  ENSEIGNEMENT  / JAPON / COMPUTER NETWORK / TEACHING /
JAPAN / UNIVERSITE TOKYO

                    LOGOFF

----------04Sep78  11:13:41  User0407---
     7.24 AU 10.86 Minutes in File14
     3.62 AU Telecomm charge
    10.86 AU approx Total

RECON session terminated at 11:13:50

%I

%%RECON-(0001 10) 0284 0501    DECONNECTE 0005
%
```

File14 ＝Electronics & Electro-technics

10.86AU＝1303(U.S.$)
(1AU＝1.2(U.S.$))

図４－10　PASCALINE(EURONET)デモンストレーション

7) PASCALINEの開発経緯

① 伊Frascatiにおける PASCALINE

INFORMASCIENCEとESA／SDS(European Space Ageney / Space Documentation Service)は、1973年に最初の接渉をして以来、1975年まで宇宙科学に関するファイルを用いた実験を行なうことを決定した。その後1975年、1976年の改良を経て実システムの運用に至っている。最終バージョンでは多国語、即ち、英語、仏語、独語の何れでも検索が可能となっている(図４－11　参照)。1977年7月には100万文献がアクセスできるシステムとなった。

```
Access keys :
Personal author                          (AU=...)
Affiliation                              (AF=...)
Classification code                      (CC=...)
Corporate source                         (CS=...)
Patent assignee                          (PT=...)
Descriptor                               (.../DE)
Term from English title                  (.../ET)
Term from French title                   (.../FT)
Term from German title                   (.../GT)
Uniterm of descriptor and/or title : no affix (prefix or suffix).
```

図４－11　PASCALINEのアクセス・キー

② 仏GrenobleにおけるPASCALINE

Grenobleでの開発システムは一貫してMISTRALと呼ばれている。1970年にはCIIのソフトウェアMISTRAL−1の研究が開始された。このソフトウェアはSDIサービスとRS検索を目的としていた。1972年から1975年にかけての実験は、Lyonsに移され計算機もCII IRIS−50からCII IRIS−80へ大型化した。同時にSDIサービスを目的とした2つのソフトウェア、CRISTALとPASCAL−VIRAも開発された。なお、この時期に研究されたMISTRAL−2はRS検索専用のシステムである。1975年の末にはグルノーブル大学での実験に移った。この実験では約10万文献を対象にし、CYCLADESとの'semi−interactive'システムも実現した。1976年にはPASCALファイルを用いてMISTRAL−3の最初の実験が行われた。この最新バージョンは会話型であり、1977年5月以降幾つかのPASCALファイルを運用している。

8) ま と め

ドキメンテーションセンターの主業務は、PASCALデータベースの作成とテープ形式での配布である。このテープは抄録誌「BS」に対応している。データベースの運用に関しては、現在ESA/SDSにデータベースを設置している。近い将来に全部門(約50)のオンライン化が実現するといわれている。今後は米国系の商用オンラインサービスとの競合も充分考えられ、その動向が注目される。CNRSでは、いわゆるDBMSを用いた分散型データベースシステムの研究開発を行なっていない。しかし、仏が進めている分散型データベースに関する国家プロジェクトSIRIUSに、CNRSのDaniel Meyer氏が参加し、集中型のデータベースから分散型データベースへの変換について可能性を追求している。

4.3.4 IRIA 情報処理自動化研究所

Institut de Recherche d' Informatique et d' Automatique

1) 所 在 地

Domaine de Voluceau − Rocquencourt

B. P. 105−78150 Le Chesnay − France

Tel. 954. 90. 20

2) 調査目的

IRIAでは、1976年6月より分散型データベースプロジェクト〝SIRIUS〟を発足させている。このSIRIUSは、前プロジェクト〝CYCLADES〟の継続プロジェクトとして4年計画で研究開発が進められている。今回は、このSIRIUSプロジェクトの現状と将来計画について調査することを目的とした。

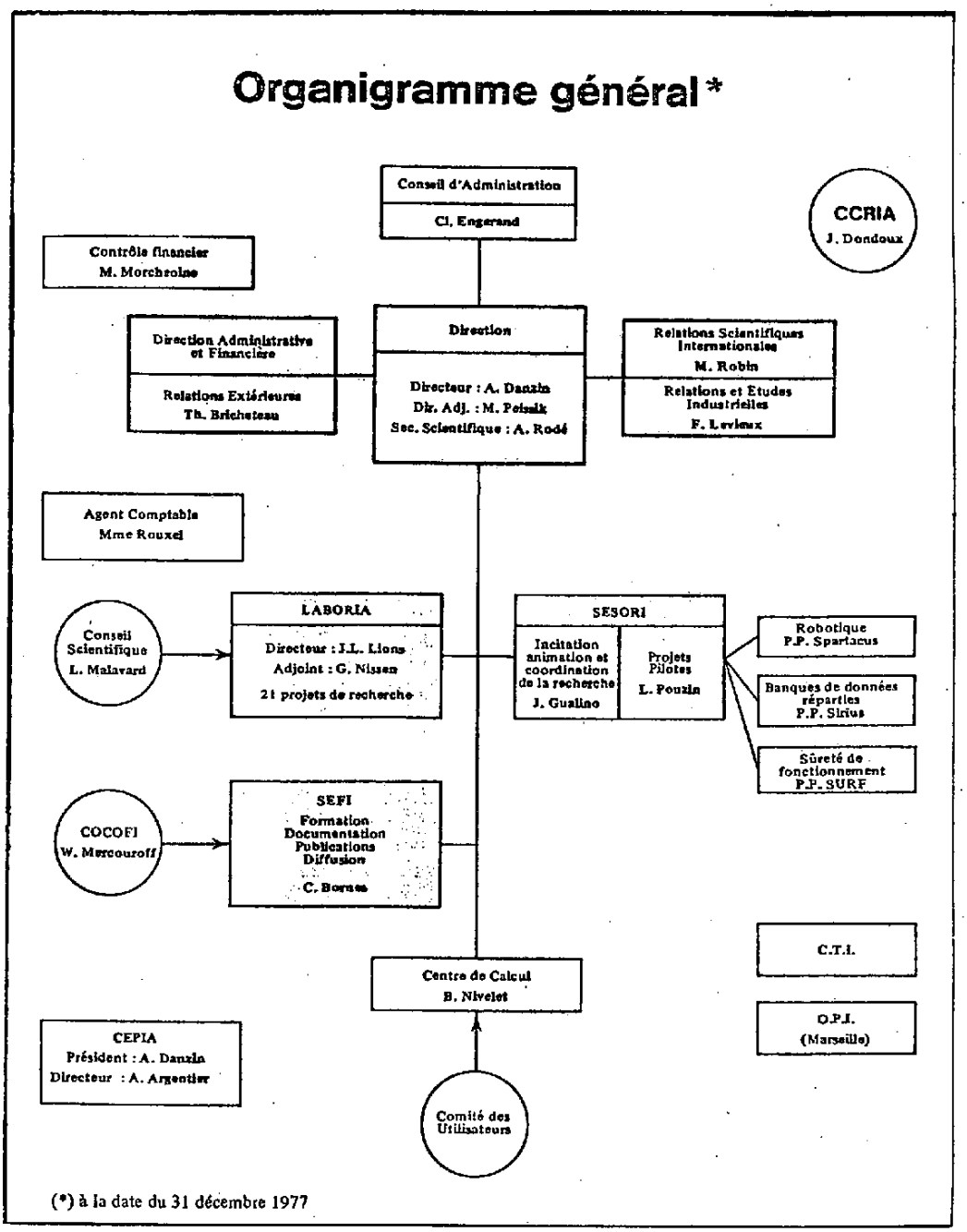


図4－12　IRIAの組織図

3) ＩＲＩＡの組織

図４－１２にＩＲＩＡの組織図を示す。図中に用いている略語の意味は次のとおりである。

LABORIA ; Laboratoire de Recherche en Informatique et Automatique.

SEFI ; Service Formation Information.

SESORI ; Service de Synthèse et d'Orientation de la Recherche en Informatique.

CCRIA ; Comité Consultatif de la Recherche en Informatique et Automatique.

CEPIA ; Centre d'Etudes Pratiques d'Informatique et d'Automatique.

COCOFI ; Comité Consultatif de la Formation et de l'Information.

CTI ; Centre Technique Informatique.

4) ＳＩＲＩＵＳの予算と組織

① 予算とプロジェクト計画

ＳＩＲＩＵＳは１９７６年６月から４年計画で進められているが、全体予算は約３０M.Frs(約７５０万US.$)で、ＩＲＩＡと産業省(Ministry of Industry)がスポンサーとなっている。 ４年計画の概要を図４－１３に示す。

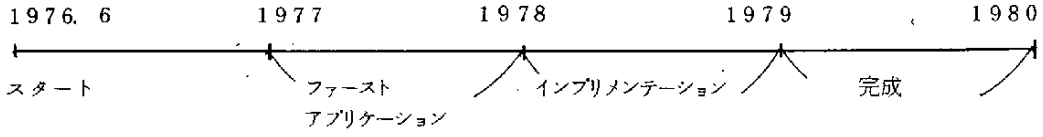


図４－１３ ＳＩＲＩＵＳの４年計画

② ＳＩＲＩＵＳの構成

現時点ではＩＲＩＡを中心に４大学から成っている。これらの構成メンバーは、何れも〝ＣＹＣＬＡＤＥＳ〟の研究・推進グループである。

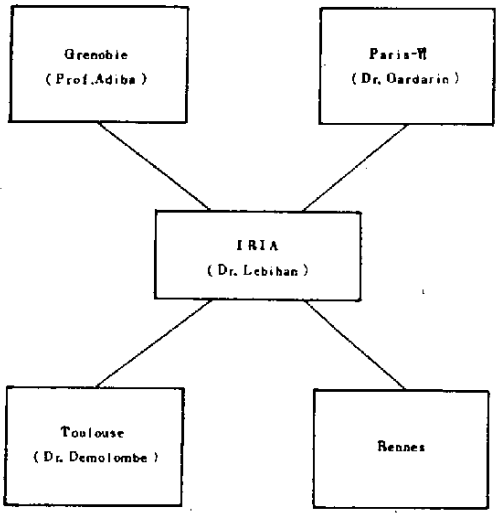


図４－１４　ＩＲＩＡと構成メンバーの代表者

グルノーブル大学には、Adiba 教授の下にＰＯＬＹＰＨＥＭＥプロジェクトがある。パリ第６大学にはＳＩＲＩＵＳのほかに、Ｇａｒｄａｒｉｎ博士の下でデータベース設計技法の研究開発プロジェクトＡＳＴＩ(Ａｎａｌｙｓｉｓ ｏｆ Ｓｙｓｔｅｍｓ ｏｎ ｔｈｅ Ｔｒｅａｔｍｅｎｔ ｏｆ Ｉｎｆｏｒｍａｔｉｏｎ)が進められている。トゥールズ大学では、同大学のＣＥＲＴセンター(Ｃｅｎｔｒｅ ｄ' Ｅｔｕｄｅｓ ｅｔ ｄｅ Ｒｅｃｈｅｒｃｈｅ à Ｔｏｕｌｏｕｓｅ)で関係モデルに基づいたＤＢＭＳのプロトタイプＳｙｎｔｅｘ－２を開発済みである。このＤＢＭＳのアーキテクチャは、ＡＮＳＩ／＊３／ＳＰＡＲＣの提案に基づいており、各レベルで必要とする情報と機能について研究が行なわれた。なお、このアーキテクチャは伝統的なＤＢＭＳ　Ｓｙｎｔｅｘ－２に基づいている。さらに、レンヌ大学においても分散型データベースに関する積極的な研究が押し進められている。

ＳＩＲＩＵＳプロジェクトの構成メンバーはフランス各地に分散しており、統合チームにより連係が保たれている。

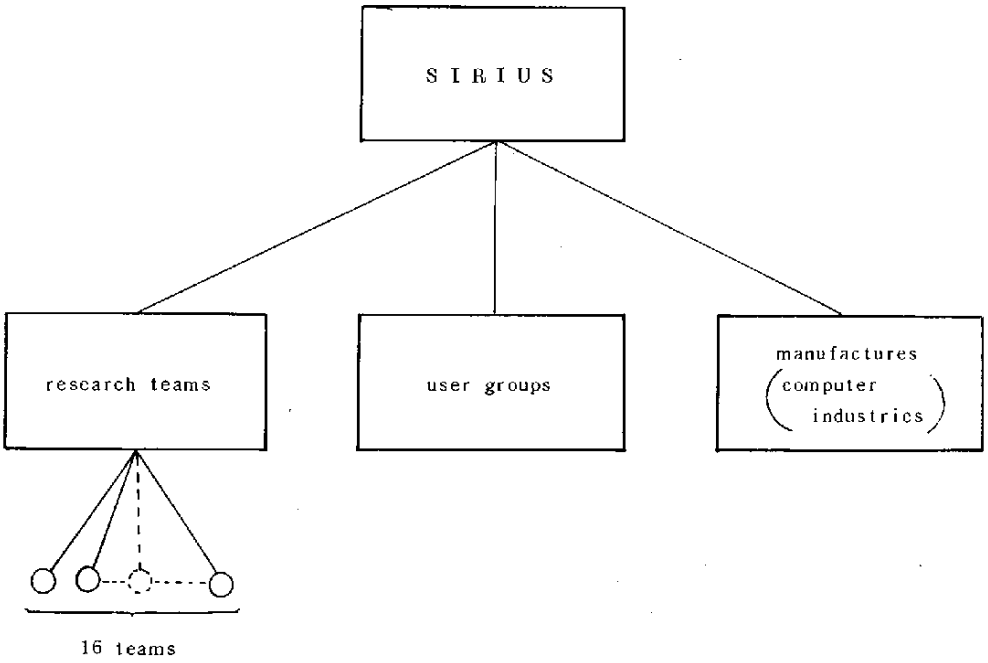


図４－１５　ＳＩＲＩＵＳの構成

図４－１５に示すようにＳＩＲＩＵＳは、１６の研究チームと幾つかのユーザグループ、さらにはコンピュータメーカから構成されている。特に４０人のスタッフが従事している研究チームのテーマは次のとうりである。

・分散型データベースにおけるデータモデルの定義

・分散型データ管理システムのアーキテクチャ

・異種のデータベースに対するアクセスプロトコル

・分散型データベースの保全、機密保護、プライバシ

・分散型システムメカニズム等

なおＩＲＩＡでは５～６人のスタッフが当プロジェクトを担当している。

5）分散型データベースシステムと共通モデル

異種のＤＢＭＳから成る分散型データベースシステムには共通モデルが必要である。

ＳＩＲＩＵＳではこの共通モデルの候補として次の４つをあげている。

・関係モデル

・Ｅ－Ｒ（事象・関係性）モデル

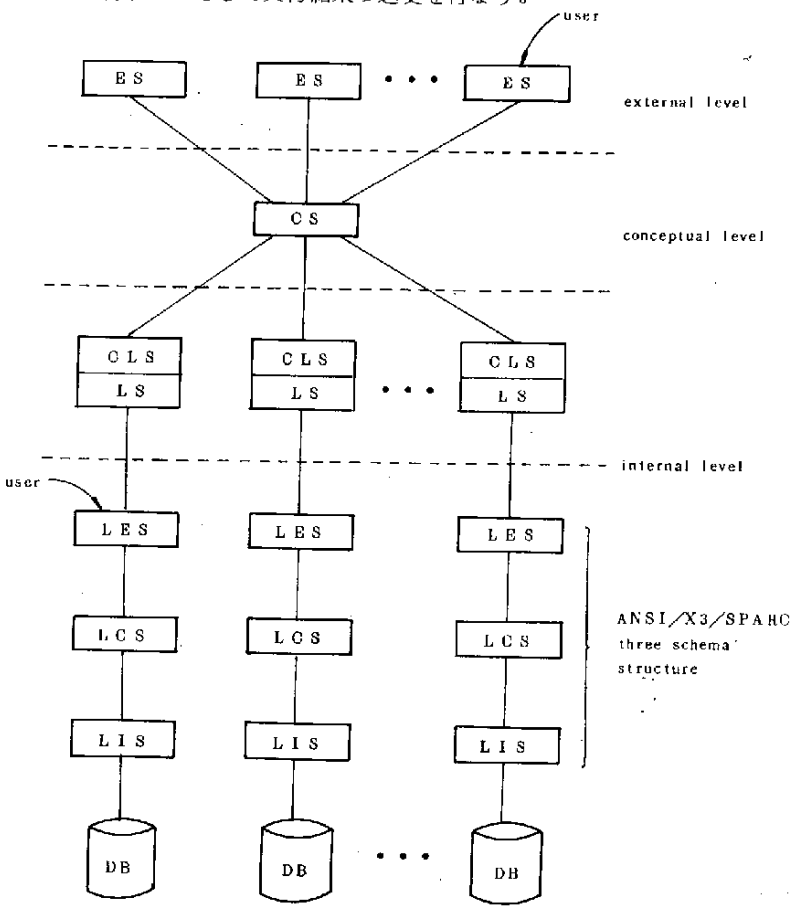
・セマンティック・ネットワーク（MOGADOR）

・他のモデル

なお、ＭＯＧＡＤＯＲとは分散型データベースの概念モデルとして設計された２項関係モデルである。このＭＯＧＡＤＯＲは局所性を排除するｆｉｌｔｅｒｉｎｇ ｔｅｃｈｎｉｑｕｅとして有効である。

分散型データベースシステム実現のためには、この共通モデルの決定以外に、通信回線のスピードや、保全、機密保護、プライバシ、同期制御に関してより一層の機能向上を目指す必要がある。

ＳＩＲＩＵＳのＧＤＤＢＭＳ(Global Distributed DBMS)の基本アーキテクチャは、ＡＮＳＩ／Ｘ３／ＳＰＡＲＣの３層スキーマに基づいており図４－１６に示すとおりである。ＧＤＤＢＭＳでは、ユーザの問合せを、まずExternalレベルからConceptualレベルへ変換（ＥＳ→ＣＳ）する。この問合せは共通言語により各サイトでの実行単位に分割される。次に実行制御のためにホスト言語を組込む。さらに、一連のローカルＤＭＬｓに展開し、ローカルプログラムを実行する。そして実行結果の送受を行なう。



ES：External Schema　　LES：Local External Schema
CS：Conceptual Schema　　LCS：Local Conceptual Schema
CLS：Conceptual Local Schema　　LIS：Local Internal Schema
LS：Local Schema　　DB：Data Base

図４－１６　ＧＤＤＢＭＳの基本アーキテクチャ

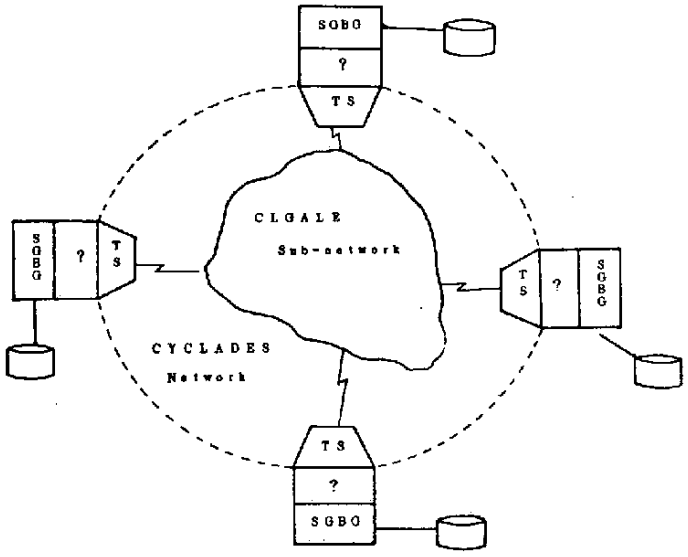


図４－１７　System Architecture

6）ま と め

ＩＲＩＡではコンピュータネットワークの研究開発に取組み成功した‘ＣＹＣＬＡＤＥＳ’の継続プロジェクトとして‘ＳＩＲＩＵＳ’が発足している。この分散型データベースの研究開発を目的としたプロジェクトは参加各大学の積極的協力により、着実に成果をあげてきている。特に、グルノーブル大学のAdiba教授率いるＰＯＬＹＰＨＥＭＥプロジェクトの データベースグループは、ＳＩＲＩＵＳの中心的地位を確立してきている。

プロジェクトＳＩＲＩＵＳは４年計画で進められているが、本年度（１９７９年）中にもインプリメンテーションが行なわれ、プロジェクトの最終年度にあたる来年度にはシステムとして完成するものと思われる。

4.3.5 Univ.of Grenoble　グルノーブル大学

Laboratoire d' Informatique

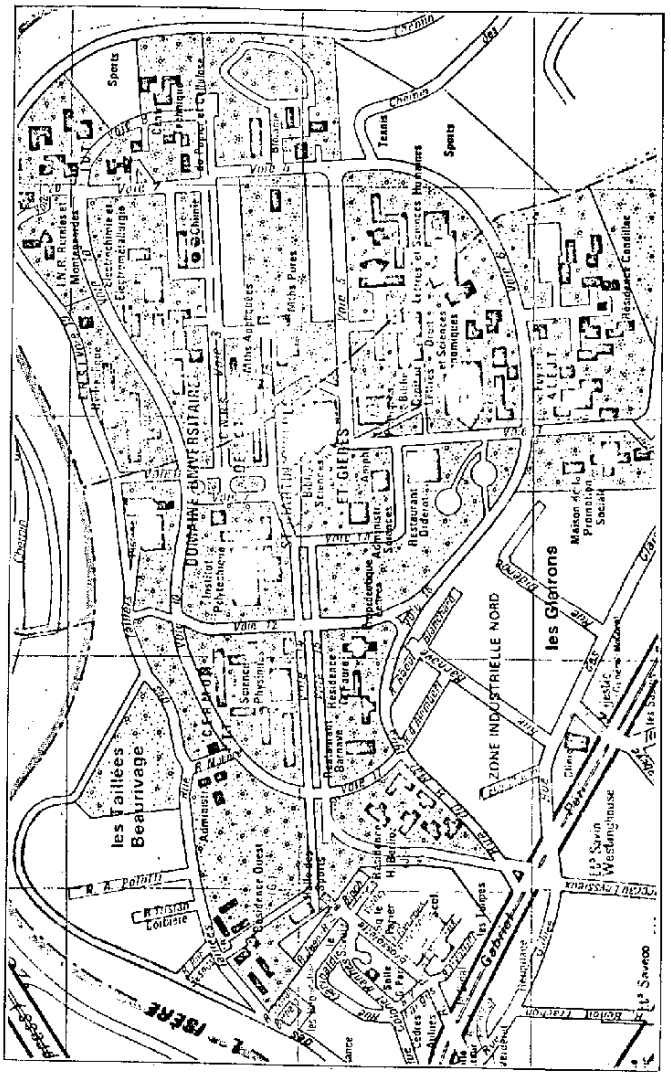
Universite' Scientifique et Medicale de Grenoble

1）所 在 地

B. P. 53

３８０４１　Grenoble Cedex,　France

Tel : (76) 54　81　45



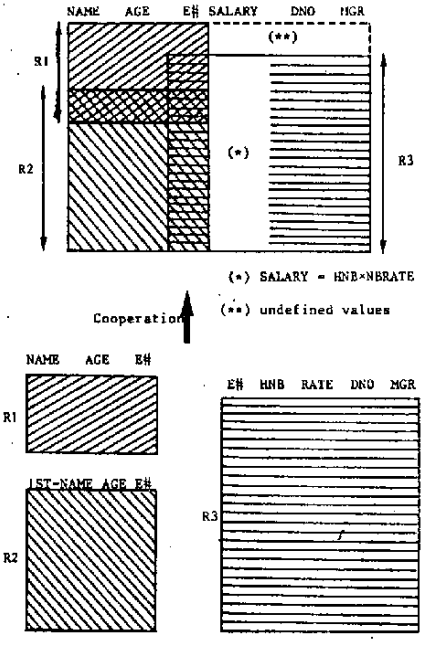
2) 調査目的

グルノーブル大学では、1976年より、IMS, DBTG, SOCRATE, SYSTEM2000, TOTAL, 関係DBMSといった異種DBMSの統合を目指す分散型データベースシステムPOLYPHEMEプロジェクトをAdiba教授のもとにすすめている。POLYPHEMEアプローチの背景には、ARPA, CYCLADES, TRANSPAC, EURONET等のデータ 通信機能を備えたコンピュータネットワークの発展がある。このPOLYPHEMEも、こうした既存ネットワークの基本通信手段に基づいている。異種のDBMSをボトム・アップ(bottom−up)の立場から統合しようとしていること(図4−18)が他のSDD−1、INGRES、LADDERに対

する大きな特徴である。今回は、このPOLYPHEMEに関して、そのアプローチ、共通データモデルと言語、分散問題、異種性問題について調査することが目的である。

3) POLYPHEMEプロジェクト

POLYPHEMEの名は、ギリシャ神話からとっている。このプロジェクトは、1976年にスタートし1979年6月ごろプロトタイプを完成させる予定である。これ以前グルノーブル大学では、Abrial, Delobelらによるデータのセマンティックス、関係モデルに関する理論的研究が進められていた。この様な研究の成果の上に、POLYPHEMEプロジェクトが発足したわけである。プロジェクトは大きく分けて、データベースとネットワークの2つのチームから成っている。データベースチームは5人、ネットワークチームは2人、この両チームのリーダーとして、Adiba教授がいる。なお、当プロジェクトのシステム開発は、CIIと協同して行なわれている。



ローカルリレーション；

R1 (E#, NAME, AGE)

R2 (E#, AGE, FIRST-NAME)

R3 (E#, HNB, RATE, DNO, MGR)

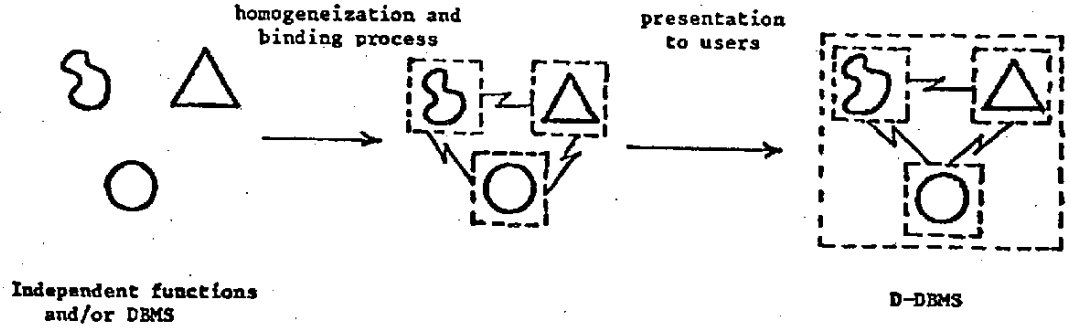


図4－18 汎用・ボトムアップの統合過程

4) データモデル

POLYPHEMEにおける共通モデルは、Abrial博士のデータ意味論(data semantics)に基づいたMOGADOR(General Model for Distributed Database)と呼ばれるものである。これは、Coddのn項関係性に対して、2項関係性に基づいている。各サイト単位に、MOGADORを用いてDBMSのスキーマを共通記述し(local view),さらにこれらからネットワーク全体のデータの意味を同じくMOGADORを用いて記述(global view)している(図4－19)。このようにMOGADORの目的は、各サイトのDBMSの同種化と、DDB全体の統一記述を行なうこと、これらのアクセスパスの記述、及び共有データに対する各サイトのDBMSのアクセス形式とを共通記述することである。

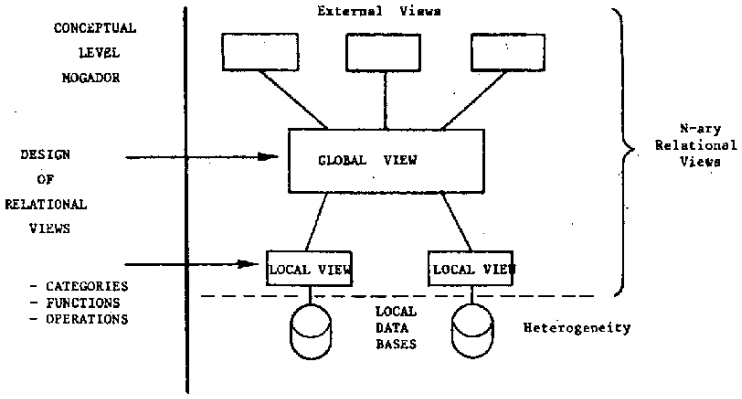


図4－19　MOGADOR：Global viewとLocal view

5) 分散問題と異種性問題

POLYPHEMEにおける重要な特徴は、データベースの異種性の問題と、ネットワークを用いてデータベースを共働させるために生ずる分散の問題とを分離して扱っている点である。まず個々のDBMSを共通データモデルによって記述する。この段階にDBMSの異種性問題が表われる。次にこの記述されたローカルデータベースから全体的な視点を記述する。この段階に、データベースの分散の問題がある。このアプローチは、2つの問題を明確に分離したという意味で価値があるものと考える。

① 分散問題

POLYPHEMEにおける分散問題は、全体視点（global view；我々のGCSに相当）とローカルデータベース記述（local view）との対応である。全体視点における全体カテゴリとローカルカテゴリ及び全体アクセス関数とローカルアクセス関数との対応である。この対応は多重グラフによって示され個々のローカルデータベースの記述とあるアプリケーション向けの全体視点の記述内の情報を用いて構成される。POLYPHEMEにおける分散問題は、多重グラフとこれに対応した記述とによって表現され、この2つをディレクトリと考えることが出来る。

リレーションの分散形態としては、INGRESと異なり、リレーションの制限（restriction）と射影（projection）とを考えている。冗長コピーも考えている。

global view レベルでのDMLとしては、関係代数URANVSを用いている。このDMLからローカルレベルへの分割プロセスと、分割されたDMLの分散実行（distributed execution）を行なっている。

POLYPHEMEでは、さらに、設計とimplementationとを分離している。即ち、設計はMOGADORでなされ、実際のアクセスとデータの記述は、リレーショナルモデルを用いてなされる。Stonebraker教授の指摘と同じく、これはリレーショナルモデルのもつ簡潔さによっている。即ち、セマンティックスは、MOGADORによって記述し、実際のアクセスは、MOGADOR記述を参照しながら、リレーショナルモデルを用いて行なわれる。

② 異種性問題

POLYPHEMEにおける異種性問題は、2項関係モデルによって記述されたローカルデータベースと、実際のDBMS（CODASYL—DBTG，S2000、SOCRATE等）との対応関係である。異種性の問題は、共通↔ローカルの変換の問題である。POLYPHEMEのアプローチは、各種DBMSを2項関係モデルによって、サイト単位に記述する。各種（関係、網型、階層型）データモデルを2項関係モデルで記述することの可能性は、Adiba等により示されている。

現在、CYCLADES上のいくつかのSOCRATE（階層型DBMS）データベースの統合を進めている。SOCRATEに関しては、リレーショナルインターフェースURANVSが完成しており、これを同種化されたリレーショナルマシンとして用いる。これは、CIIによってな

された。

5) ま と め

グルノーブル大学で進められているPOLYPHEMEプロジェクトは、Abrial博士やDelobel博士らにより1970年初期から進められているデータ意味論（data semantics）や関係モデル等に関する理論的研究成果を統合したプロジェクトである。このプロジェクトは、異種DBMSの統合を目指しており、フランスにおける最も大きくかつ重要な分散型データベースの研究プロジェクトである。1979年6月にはCIIとの協力のもとに、CYCLADES上のSOCRATEデータベースの統合を計画している。この段階では同種のDBMS統合としてインプリメントされるが、最終的には汎用・ボトム・アップの立場から異種のDBMS統合の実現を目指している。

POLYPHEMEの特徴は、2項関係性に基づいたMOGADORを共通概念モデルとして用いていることである。また、我々のシステムにおけるLCS（局所概念スキーマ）には、CIIが開発したリレーショナルインタフェースURANVSを用いる計画である。このようなことからグルノーブル大学では、分散問題に多くの力を注いでいる。

なお、Adiba教授は「異種のDBMSの統合が実現されれば、パフォーマンスロスも多いと思われるがそれは支払うべき代償であり、統合することにより受ける恩恵の方がはるかに高い」と述べている。

4.3.6 GMD 西独数理データ処理公社

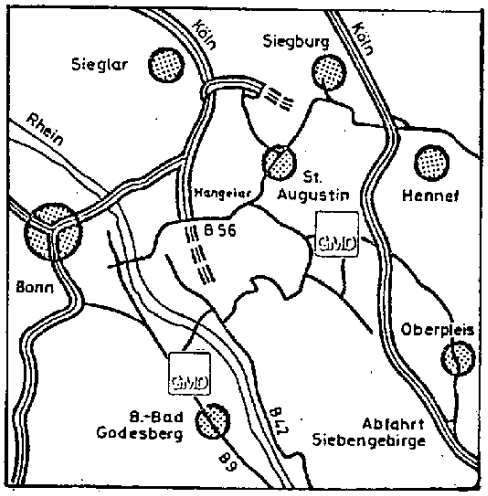
Gesellschaft für Mathematik und Datenverarbeitung mbh Bonn

1) 所 在 地

PO BOX 1240, Schloss Birlinghoven,

D－5205 St. Augustin 1, West Germany

Tel：（0 22 41）14－1



2）調査目的

ＧＭＤでは分散型データベースシステム研究開発プロジェクト〝ＩＮＧＶＥＲ〟を発足させている。調査の第一目的は、このＩＮＧＶＥＲの概要を知ることである。また、ＧＭＤが開発したデータバンクシステムＦＩＤＡＳについても合わせて調査するものとした。

3）ＧＭＤの組織

ＧＭＤは第一次情報処理計画期間中の１９６８年４月に設置された連邦、州両政府出資（連邦、９０％、州１０％）の公的研究所である。

ＧＭＤには１１の研究所と５つの部門があり広範な研究を行なっている。これらの研究を支えるスタッフは約６００人に達する。

Institute for Education in Informatics (IK)
Institute for Teleprocessing (IFV)
Institute for Data Processing and Law (IDR)
Institute for Graphical Data Processing and Structure Recognition (IGS)
Institute for Information Systems (IIS)
Institute for Information Systems Research (ISF)
Institute for Mathematics (IMA)
Institute for Planning and Decision Systems (IPES)
Institute for Computer and Program Structures (IRP)
Institute for Software Technology (IST)
Computing Centres (RZ)
Department for Information, Documentation and Publication (AIW)
Department for Project Management (APM)
Technical Department (TA)
Department for Administration (VA)
Central Services (ZA)

図４－２０　ＧＭＤの研究所と部門

4）データバンクシステムＦＩＤＡＳ

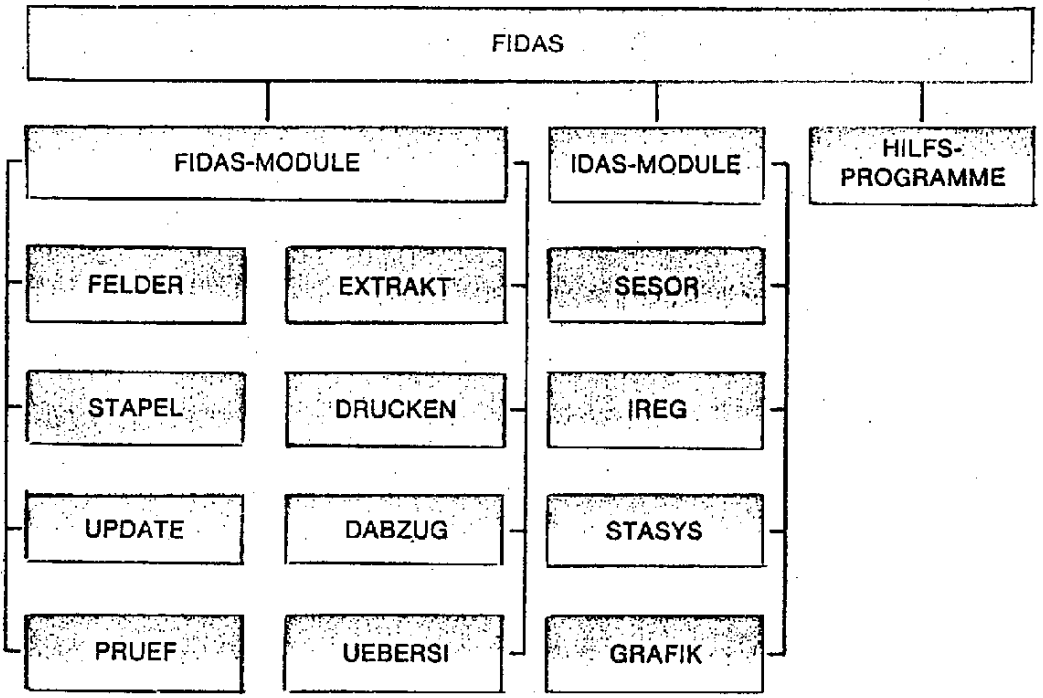
ＦＩＤＡＳ(Forms-oriented Interactive Data Base System)は、ＧＭＤで開発したデータバンクシステムである。このＦＩＤＡＳは現在西独国内で数十の企業や研究所で、合わせて数百のアプリケーションに用いられている。ＦＩＤＡＳが持つデータベースアクセス言語はＣＯＤＡＳＹＬ／ＥＵＦＴＧ(ＣＯＤＡＳＹＬ　Ｅｎｄ　Ｕｓｅｒ　Ｆａｃｉｌｉｔｙ　Ｔａｓｋ　Ｇｒｏｕｐ)が提案しているものに類似しているが、独自に開発されたものである。

ＧＭＤではこのＦＩＤＡＳの機能向上とメインテナンスのために、ＦＩＤＡＳのユーザ会を年数回開催している。

ＦＩＤＡＳとの対話は、既形式化されたいわゆるシステムリード型で行なわれるので、初心者にとっては有効であるが、慣れてきたり熟練者にとってはわずらわしさがある。〝ＳＴＡＳＹＳ〟や〝ＧＲＡＦＩＫ〟モジュールを用いるとヒストグラムや折線グラフの表示が可能であるので、数値

情報の統計処理には有効であろう。図４－２１にＦＩＤＡＳのモジュール構成と機能概要を示す。

モジュール構成



モジュール名	機　能
FIDAS Module	Lademodul. Aufruf der FIDAS-Module über Parameter
FELDER	Definition u. Änderung der Satzstruktur, der Feldnamen u. Feldlängen
STAPEL	Dateneingabe und Änderungsdienst im Stapel-Betrieb
UPDATE	Parametergesteuerte Bildschirmanzeige der Datensätze, Eingabe, Änderungsdienst im Dialogbetrieb
PRUEF	Plausibilitätskontrollen
EXTRAKT	Extrahieren von Datensätzen, Zusammenfügen von Dateien nach vorgebbaren Kriterien
DRUCKEN	Dateien ganz oder in Teilen drucken (Druckformat)
DABZUG	Dateien ganz oder in Teilen drucken (Bildschirmformat)
UEBERSI	Dateiinhalte in Übersichtstabellen darstellen
SESOR	Selektieren, Sortieren, Rechnen, mit und ohne Deskriptorenangabe
IREG	Layout und Berechnungen für Ausgabelisten beliebigen Formats
STASYS	Statistische Auswertungen
GRAFIK	Grafische Ausgabe auf Bildschirm und Listen (Druckformat) sowie über Plotter
HILFS-PROGRAMME	Module für verschiedene Arten von Dateimanipulationen

図４－２１　ＦＩＤＡＳのモジュール構成と機能概要

5) 分散型データベースシステム INGVER

GMDでは、FIDASとは別に分散型データベースシステムの研究開発が進められている。このプロジェクトは〝INGVER〟(Interactive Systems based on Network) といい、異種のDBMSから成る分散型データベースシステムの構築を目指している。

当研究の基本概念は、ハードウェア、ソフトウェア、特にユーザインタフェース(問合せ言語)に対して大幅な変更を生じさせないで、異種のDBMS間の情報交換手段を提供することにある。また、システムのインプリメントにあたっては、英IBMのピータリー研究所で開発された言語CIL(Common Intermediate Language)を用いている。このCILを採用した理由は新たな言語を開発しなくて済む時間の節約と実験システムとしては機能的に充分だからである。

INGVERのインプリメンテーションにあたっては、中・大型のコンピュータを前提とし、さらには異種のDBMSから成る分散型データベースについても配慮している。従って、グルノーブル大学におけるPOLYPHEMEの流れにあり、UC. BerkeleyのINGRESのように小型のコンピュータを用い、しかも同様のDBMSから成るアプローチとは異なっている。

図4－22に分散型データベースシステム(DBS)の機能関連図を示す。

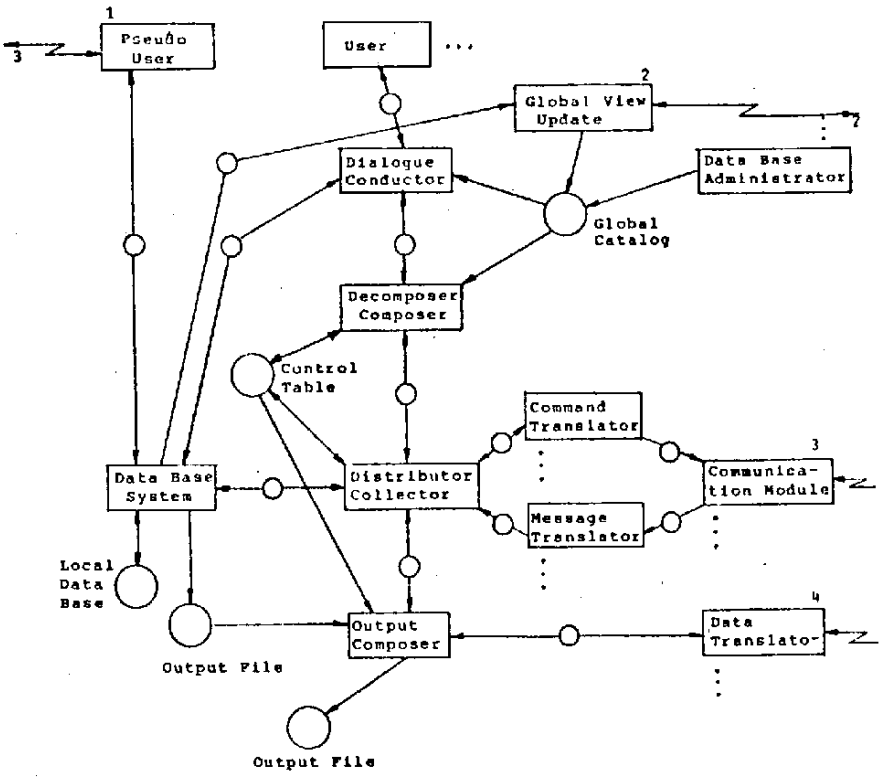


図4－22 Structure of the DBS interworking Controller

6) GMDの研究開発における今後の課題

GMDのK. Kruckeberg氏は次の事項を指摘している。

① データベースアクセス用ユーザ言語は、ユーザ本位でなければならない。即ち、CODASYLのEUFTGが提案しているようなものである。GMDにおいてはFIDASでこれを実現しているが、今後より一層の操作性が期待されている。

② データベースシステムは、ユーザが最適なデータモデルを選択できるように、種々のアクセスレベルで幾つかのデータモデルをサポートすべきである。従って、データベースシステムのモデルの共存や多層アーキテクチャの研究は重要である。

③ データベースのデータ構造には、企業の組織や管理構造を理想的な形で反映すべきである。このためには、データベースに対する同時アクセス、一致性、保全性の解決が必要である。さらに、プログラム・モジュールやデータ構造の標準化の必要性を明確化する必要がある。

④ 大規模データベースの運用は、新しい蓄積、検索手法をサポートする新しいハードウェア技術に大いに依存する。特に、連想メモリ、同時アクセスアルゴリズムが主要なアプローチとなろう。しかし、大規模データベースは幾つかの非技術的に重要な面を持っている。特に政治的、社会的、経済的構造に与えるインパクトについて配慮すべきである。

7) ま と め

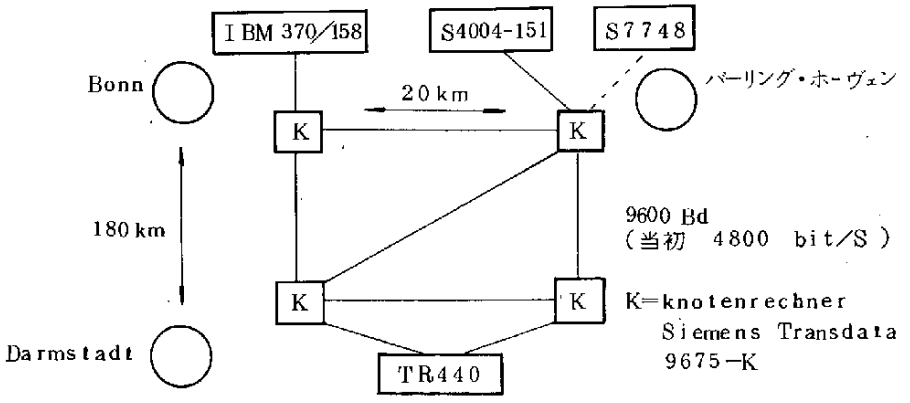
今回調査したFIDASは汎用データバンクシステムとして、西独国内における地位を確立しつつある。現在の実績からみて今後益々ユーザ数が増加すると思われる。一方、分散型データベースシステムを研究開発中のINGVERも数年の内にインプリメンテーションを完了し、将来的にはGMD－netとの結合も考えられる。

参　考

～　GMD－Net　～

GMDは、第3次情報処理計画のデータ遠隔処理システムのプロジェクト推進機関としての役割を持たされている。GMD－Netは、こうしたデータ遠隔処理システム計画の一プロジェクトとしてGMDが中心となって開発を進めているリソース・シェアリングをめざした、異機種コンピュータ・ネットワーク・システムである。

こうした意味で、GMD－Netは、同種のコンピュータ・ネットワークに対して、ベースの技術を提供する実験システムとしての役割を果たすとともに、西独の技術水準を反映する焦点となっている。



ネットコントロールセンター

GMD－Net 全体を監視し
管理する役割を持つ計算機である。
特にモニタ機能と中央のネット
ソフトウェア開発手段をもっている。

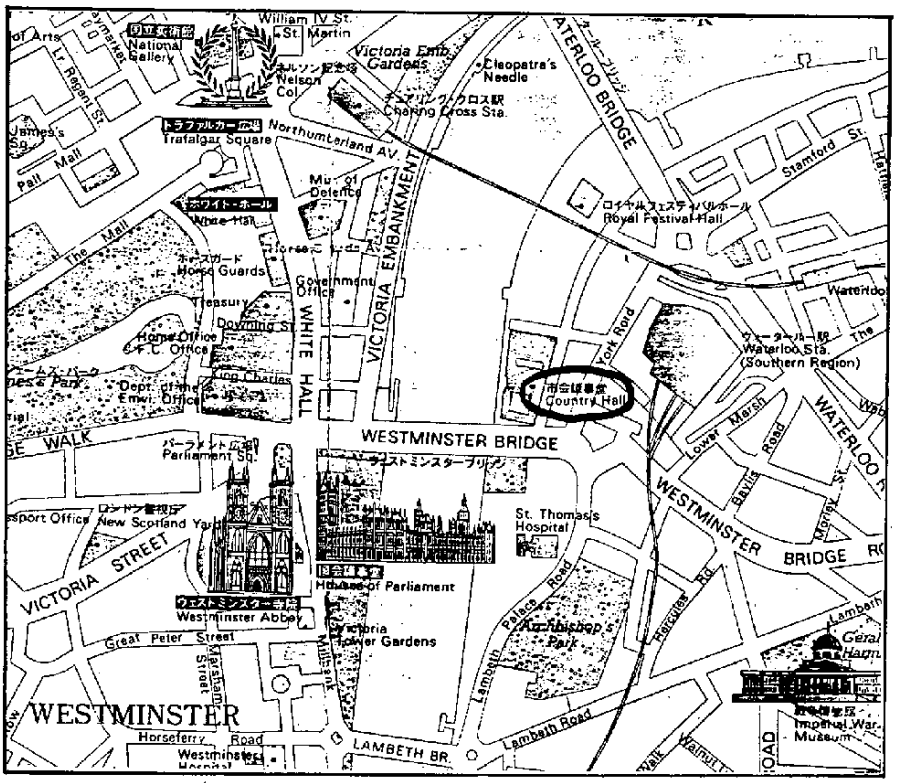
図4－23　GMDネットワーク構成

4.3.7 GLC 大ロンドン市議会

Greater London Council, Central Computer Service

1) 所 在 地

The County Hall London SE1 7PB, U.K.



2) 調査目的

GLCでは英IBMサイエンティフィックセンターの協力を得て開発したDBMS，即ち、IPS及びPRTVを運用している。今回の目的はこれらのDBMSの稼働状況をユーザサイドから調査するとともに、商用のDBMS（ADABAS，IMS）の利用状況を調査することにある。

3) GLCの組織

GLCの組織は図4－24に示すとおり、8グループ13の専門委員会がGLCの直轄になっている。また、政策・財源委員会（Policy and Resources Committee）を除く12の委員会には7つの実行委員会が属している。特に、住宅問題実行委員会には、都市計画や医療関係の14部が属している。

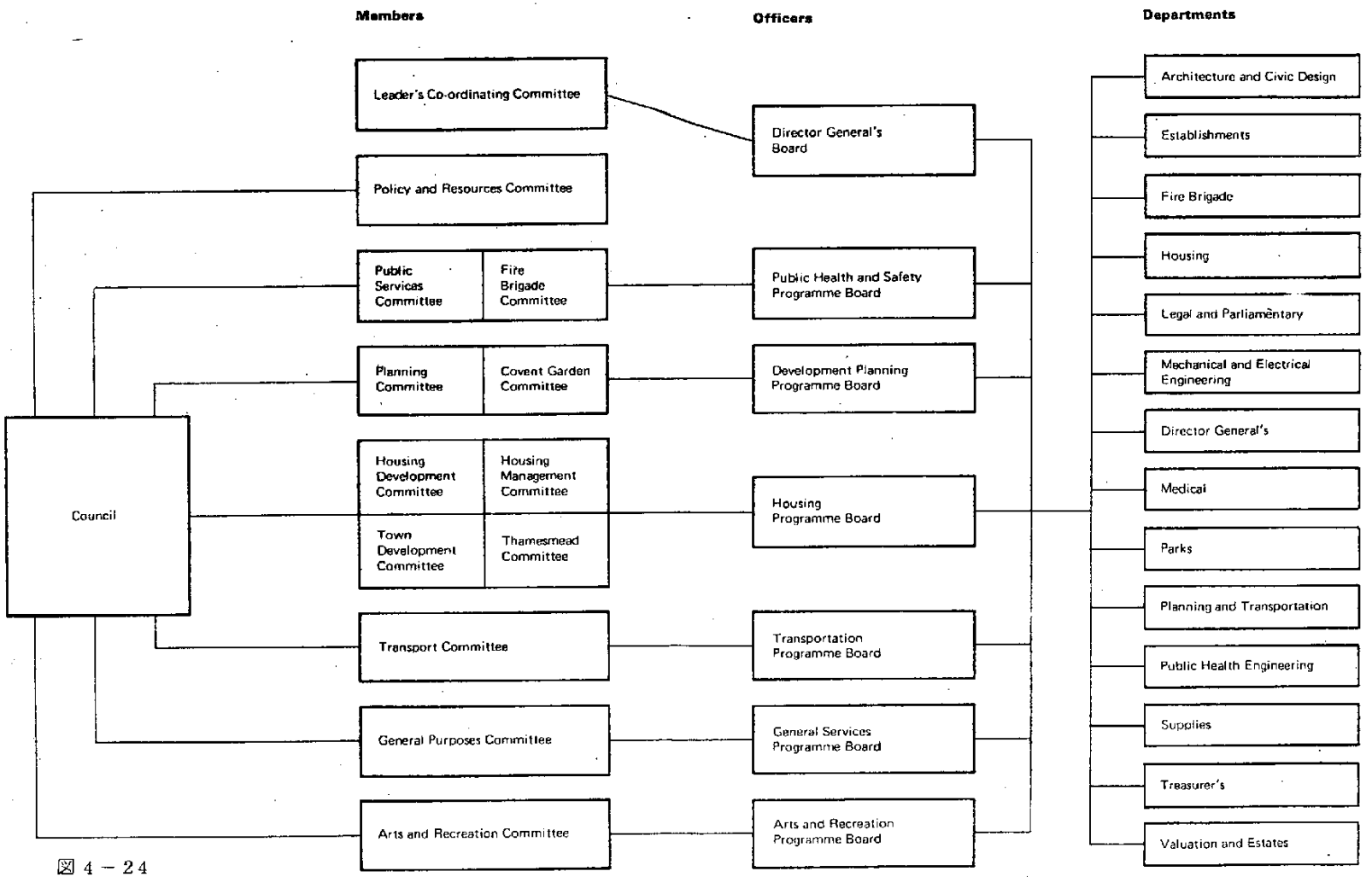


図 4－24
Management Structure of
the Greater London Council

4) ＧＬＣにおけるＤＢＭＳ

① ＩＰＳ

ＩＰＳ(Interactive Planning System)は英国ＩＢＭサイエンティフィックセンターの協力を得て開発されたシステムで、

i) 大容量データの分析

ii) 分析プログラマの充当

に対処できるものである。また、表形式でアクセス可能なエンドユーザ言語をもつＤＢＭＳである。機能的特徴としては次の２点に集約できる。

i) 単純な表表現でデータ操作が可能

ii) データ間の関係性を動的に設定可能

ＩＰＳはいわゆる関係モデルのＤＢＭＳに属するが関係モデルで用いる用語とは名称が異なる。図４－２５にＩＰＳの基本データ構成を示し、（　）内に関係モデルの用語を併記する。

	Attributes			
	AGE	NAME	WRIGHT	
	16	FRED	12.42	)
Block name PEOPLE	18	JOHN	13.62	) Entries
	21	SUE	8.94	) (Tuples)
(Relation)	15	ANN	7.98	)

図４－２５　ＩＰＳの基本データ構成

ＩＰＳのエンドユーザ言語が持っているコマンドは、エントリを選択するＳ(select)，演算を行なうＩ(intersection)，Ｕ(union) 等がある。以下、ＩＰＳのユーザガイドから代表的なコマンドとその機能を示す。

```
a.  SINGLE BLOCK OPERATIONS
    *I<-<: S  :Expression:>Block
           P
           A
           K
           R
           N
           C

    "S" - Select entries

    Expression::=logical  test using
        or      &  and
```

= is equal to, > is greater than,
= is not equal to, < is less than,
<= is less than or equal to,
>= is greater than or equal to

"P" - sample entries
Expression:: = variable or constant %age
"N" - rename attributes
Expression:: = old attributes list;
new attributes list
"A" - aggregate attributes
Expression:: = total attributes list;
key attributes list
"K" - keep selected attributes
Expression:: = kept attributes list
"R" - remove selected attributes
Expression:: = removed attributes list
"O" - calculate attributes
Expression:: = new attribute;
attribute expression

b. INTER-BLOCK OPERATIONS

Block <- Block Simple block copy
*T <- Block1 <: I : B1 atts; B2 atts :> Block 2
D
E
U

"I" - Intersection
Find entries common to two blocks.
"D" - Difference
Find entries in block 1 but not in block 2
"E" - Exolusion
Find entries in block 1 or block 2
but not in both
"U" - Union
Find entries in either block 1 or block 2
or both.
Join Operator
*T <- Block 1 <:J: Selection criteria:> Block 2
Expands selected entries in Block 1
by adding all attributes from corresponding

entry in block 2
Link Operator
*T<- Block 1<:L: Atts. to be compared:>Block 2
Attributes have some names in both blocks
and are only tested for equality

c. OUTPUT FORMATS

*T<- *F List Format Values
*F<- x1,x2,x3,x4,x5

		DEFAULT
x1	Page Size	50
x2	Output Limit	50
x3	Line Length(*T)	76
x4	Line Length(*P)	120
x5	Output Order	U
F<-,*	Reset x1,x2 to defaults	

d. SPECIFYING OUTPUT

*T(att1,att2,...)<-Block
List specified attributes
*T(att1, att2,...; key1, key2)<-Block
Output in key1,key2 order
with sub-totals

e. SYSTEM COMMANDS

*CLEAR reloads the permanent workfile
*COM prefixes comments
*SAVE sends blocks to workfile
*DESTROY deletes blocks in workfile
*OFF terminates a session

f. RUNNING A REPORT
*T<-*R([SPACE.]BLOCKS[(B1,...)])
SPACE tells you what percentage
of the workfile has been used.
BLOCKS gives information about
block attributes and size.

g. EXAMINING BLOCKS

*T<-*B		List Absolute Blocks
<-*V		" " Variables
*T=-*B		" Defined Blocks
T<-		" last block assigned
*T<-N(Blk)		Display No. of entries
A(Blk)		" " " attributes
D(Blk)		List Block attributes
Block		" " entries
*T<-AVG	(Blk,attr)	Display average
MAX		maximun
MIN		minimun
TOT		total
V		sample valus

h. SPECIAL-PURPOSE FUNCTIONS

CHAR(att) converts fl-pt. or integer attribute entries to character.
FLOAT(att,p) converts character or integer attribute entries to fl-pt.
LOG(att) calculates log to the base 10 for all entries and returns fl-pt. no.
SEQ requires no parameters; returns set of positive integers.
SUBSTR(att,start,length) returns substring of character attribute in given position with length supplied.
INT(att) converts fl-pt. attribute entries to integer.

i. ATTRIBUTE ASSIGNMENT

Blockname. attribute < - expression
The source block is extended by an additional attribute derived from the expression.

② ＰＲＴＶ

ＰＲＴＶ(Peterlee Relational Test Vehicle)は、英国Peterlee にあるＩＢＭサイエンティフィックセンターで開発された関係モデルに基づくＤＢＭＳで、従来はＩＳ／１と呼ばれていた。ＧＬＣでは、都市計画に関するデータベース（約5000件）をこのＰＲＴＶで運用している。なお、ＧＬＣでの説明によれば、ＰＲＴＶとＩＰＳは機能的に共通点が多いとのことである。

ＰＲＴＶは、１９７３年８月以来本格稼働しており本来の目的は次の機能を提供することにある。

i) 高度かつ柔軟性のあるデータサポート

ii) 機能拡張性が容易である。

iii) 効率的である

ＰＲＴＶへのインターフェースは、いわゆる関係代数に基づいた言語である。演算は、union(+), intersection(・), difference(−), projection(%), selection(:), join(*) が可能である。

BOOKS （ACQ#, AUTHOR, TITLE）

LOANS （ACQ#, BORROWER, DATE−OUT）

この２つのリレーションに対する問合せ例を考える。

▼Who has ▼▼Persuasion ▼▼ on loan ? ▼

この問合せは次のように表わされる。

BOOKS*LOANS: TITLE = ▼Persuasion▼

ＰＲＴＶにはＩＳＢＬ(Information System Base Language）という言語がユーザに解放されている。

次の例では図書館の貸出し業務管理のために３つのリレーションを処理している。

```
LOANS_TODAY(ACQ_NO,NAME:BORROWER)
RETURNS_TODAY(ACQ_NO)
NEW_BOOKS(ACQ_NO,NAME:AUTHOR,NAME:TITLE).

The end-of-day processing would be as follows:

1 LOGON LIB,LIB
2 BOOKS = BOOKS + NEW_BOOKS
3 LOANS_TODAY1 = LOANS_TODAY * <DATE=741121>
4 LOANS = LOANS + LOANS_TODAY1 − RETURNS_TODAY
5 PUT BOOKS
6 PUT LOANS
7 LOGOFF
```

図４－２６　図書館における貸出し業務の例

③ ＡＤＡＢＡＳとＩＭＳ

ＧＬＣではこの他ＡＤＡＢＡＳとＩＭＳの２つのＤＢＭＳを導入している。ＡＤＡＢＡＳは住宅問題に関係したデータベースを取り扱っている。

5) ＰＲＥＳＴＥＬ

今回の訪問では、ＰＲＥＳＴＥＬのデモンストレーションを見る機会が得られた。ＰＲＥＳＴＥＬ（Presentation Telecommunication，旧名称 Viewdata）は、英国郵政省が推進しているテレビと電話を利用した新しい形態の情報提供システムである。このＰＲＥＳＴＥＬに情報提供の予定会社数は既に１００以上に達しており、１９７９年末頃には約60,000ページの情報をもつシステムとして稼働が予定されている。なお提供予定の情報の一部を次に示す。

News and Weather	Education
TV and Radio Guide	Houses
Sport	Jobs & Careers
Going Out	Facts & Figures
Holidays	Food Guide
Travel	Money
Hobbies	Stock Exchange & Commodities
Jokes and Games	Company & Market Information
Market Place	Manufacturing Industry Guide
Buying Advice	Service Industry Guide
Cars	Reference Information
Community Services	Government Information
Home and Family	Calculations

6) ま と め

ＧＬＣでは、ＩＰＳ，ＰＲＴＶ，ＡＤＡＢＡＳ，ＩＭＳと数多くのＤＢＭＳを導入し行政処理に運用している。今回の調査では、特にＩＰＳについて詳しい情報が得られた。ＧＬＣの端末室におけるデモンストレーションを見た限りでは、関係代数に基づいたエンドユーザ言語も機能的に充分と判断した。今後、これら異種のＤＢＭＳからなるＧＬＣデータベースをどう運用（統合）していくのか興味深いが、何れにしても分散型データベースとしての本格稼働は未だこれからの段階といえる。

4.3.8 Berkeley Workshop と ＶＬＤＢ

① The 3rd Berkeley Workshop on Distributed Data Management and Computer Networks，
San Francisco Aug. 1978

主催　Lawrence Berkeley Laboratory，University of California
Office of Energy Research，U. S. Department of Energy

期間　１９７８年８月２９日 ～ ３１日

場所　Shelaton Palace Hotel

概要　この会議は、いわゆる分散型データベース（ＤＤＢ）に関する問題を議論することを目的と

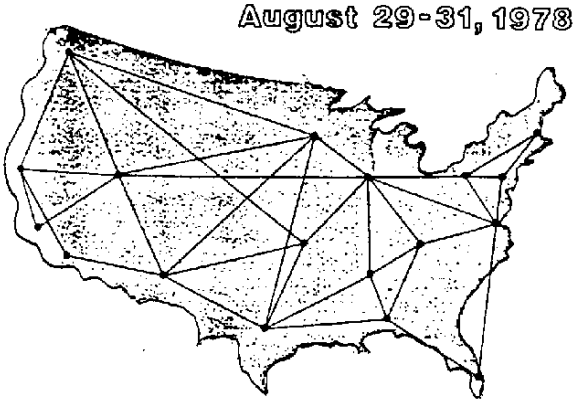
して、今回で3回目である。

今回の会議の特徴は、DDBにおける制御機能、特にconcurrency制御の問題に多くの論文を集めていた。UC. BerkeleyのStonebrakerは、多重コピーをもつDDBにおける更新アルゴリズムを発表された。UCLAのMenasce教授は、DDBにおけるデットロックの検出アルゴリズムの発表を行なった。この他にも、New York州立大学のYao博士からは、各サイトのリレーションの各属性の値の数、selectivityに基づいた問合せの分散処理のアルゴリズムの発表があり興味をひかれた。招待講演として、HISのBachman博士のISO／TC97／SC16のオープンシステムアーキテクチャの講演と、IBMのCodd博士から、自然言語による問合せシステム〝ランデブー〟の講演があった。

今回の会議を通して、感じられることはDDBの実現の可能性を示すための種々のシステムの制御機能が大きくとり上げられ、近いうちにDDBは実験段階に入ると思われる。

THIRD
BERKELEY WORKSHOP
on
DISTRIBUTED
DATA MANAGEMENT AND
COMPUTER NETWORKS

August 29-31, 1978



LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA, BERKELEY

PREPARED FOR THE U.S. DEPARTMENT OF ENERGY UNDER CONTRACT W-7405-ENG-48

INVITED TALK

Provisional Model of Open System Architecture
Charles W. Bachman

NETWORK PROTOCOLS AND ARCHITECTURE
Chairperson: *Carl Sunshine*

Software Architecture for a Microprocessor-Based Packet Network
Holger Opderbeck, Jay H. Hoffmeier, and Roy L. Spitzer

Protocol Converters and User Interface in the CYCLADES Network
N. Naffah and H. Zimmerman

An Architecture for Secure Packet-Switched Ne vorks
M. A. Padlipsky

PERFORMANCE ISSUES IN DISTRIBUTED DATABASES
Chairperson: *Peter Kreps*

Analysis of Update Synchronization for Multiple Copy Data-Bases
Erol Gelenbe and Ken Sevcik

Query Processing on a Distributed Database
Alan R. Hevner and S. Bing Yao

Performance Comparison of Two Update Algorithms for Distributed Databases
Hector Garcia-Molina

LOCAL NETWORKS
Chairperson: *Dennis Hall*

Interaction Between Protocol Levels in a Prioritized CSMA Broadcast Network*
James E. Donnelley and Jeffry W. Yeh

The Pacific Northwest Laboratory Minicomputer Network
L. H. Gerhardstein, J. C. Schroeder, and A. J. Boland

COMMUNICATION TECHNIQUES
Chairperson: *Peter Alsberg*

Inter-Network Fragmentation and the TCP*
John F. Shoch

Dynamic Management of Packet Radio Slots
Glenn Ricart and Ashok K. Agrawala

Failsafe Distributed Algorithms for Routing in Communication Networks
Adrian Segall

CONCURRENCY CONTROL I
Chairperson: *Bea Yormark*

A Formal Model of Concurrency Control Mechanisms for Database Systems†
Philip A. Bernstein and David W. Shipman

図４－２７　Berkeley Workshop 発表論文 (1)

Maximally Concurrent Transaction Processing
Toshimi Minoura

Locking and Deadlock Detection in Distributed Databases†
Daniel A. Menasce and Richard R. Muntz

CONCURRENCY CONTROL II
Chairperson: *Peter Ting*

Concurrency Control and Consistency of Multiple Copies of Data in Distributed INGRES†
Michael Stonebraker

Algorithms for Distributed Data-Sharing Systems Which Use Tickets
Gérard Le Lann

A Proposal for Distributed Concurrency Control for Partially Redundant Distributed Data Base Systems
D. Z. Badal and G. J. Popek

NETWORK MODELS AND PERFORMANCE
Chairperson: *Jon Postel*

On the Design of Elementary Distributed Systems*
Randolph T. Yeh and K. Mani Chandy

Synchronization in Distributed System Modules*
Gregor V. Bochmann

Performance Limits of the TMM Protocol: Modelling and Measurement
Erol Gelenbe, Jean-Louis Grangé, and Patrice Mussard

Distributed Simulation Using a Network of Microcomputers*
J. K. Peacock, J. W. Wong, and Eric Manning

図4－27　Berkeley Workshop 発表論文(2)

② The 4th International Conference on Very Large Data Bases

West Berlin, Sept. 1978

主催及び後援

Association for Computing Machinery (ACM) and its Special Interest Groups (SIGs) on Business Data Processing (SIGBDP), Information Retrieval (SIGIR), and Management of Data (SIGMOD)
Association Francaise pour la Cybernétique Economique et Technique (AFCET)
British Computer Society (BCS)
Chalmers University of Technology
Computer Society of India (CSI)
Defense Advanced Research Project Agency, USA
Gesellschaft für Informatik (GI)
Hahn-Meitner-Institut (HMI)
IBM, Germany
IEEE Computer Society and its Technical Committee on Database Engineering
Information Processing Society of Japan (IPSJ)
Institut de Recherche d'Informatique et d'Automatique (IRIA)
International Federation for Information Processing (IFIP)
Japan's Special Research Project on Scientific Information Systems
National Bureau of Standards, USA
Siemens AG, Germany
Swedish Society for Information Processing (SSI)
U.S. Army European Research Office

期間　1978年9月13日 ～ 15日

場所　International Congress Hall

概要　第4回VLDB(Fourth International Conference on Very Large Data Bases)は、9月13日から3日間にわたり西ベルリンで開催された。VLDBは、大規模データベースの研究と理解を深めるために発足したもので、データベースの国際会議としては回を重ねるごとに充実してきている。実際、今大会には欧州におけるデータベースの研究者のほとんどが参加し、我々が訪問した仏のIRIA、ブルノーブル大学、西独のGMDからも大挙して参加していた。発表論文は27セッション約90論文で会議参加者は14ケ国約400名の多きを数えたが、日本や米国本土(特に西海岸)からの参加者は少なかった。なお、第5回大会は1979年10月3日から3日間、ブラジルのリオ・デ・ジャネイロでの開催が予定されている。

表4－3　VLDB開催都市一覧

回数	開催年月	開催国・都市
第1回	1975. 9	米国、Cambridge, MA.
第2回	1976. 9	ベルギー、Brussels
第3回	1977. 10	日本、東京
第4回	1978. 9	西ベルリン
第5回	1979. 10	ブラジル、Rio de Janeiro

FOURTH INTERNATIONAL CONFERENCE ON VERY LARGE DATA BASES

WEST-BERLIN **SEPTEMBER 13-15, 1978**

SURVEY SESSIONS

DATA BASE DESIGN
DATA BASE SOFTWARE ENGINEERING
DATA BASE SECURITY AND PRIVACY
IMPACT OF NEW TECHNOLOGIES
DISTRIBUTED DATA BASES
DATA BASE THEORY AND MODELING

PANEL SESSIONS

WHAT END-USER FACILITIES DO WE NEED?
IMPLEMENTING A THREE-SCHEMA DBMS FRAMEWORK
EFFICIENT ORGANIZATION OF INTERNAL DBMS STRUCTURE
WHAT IS A "GOOD" SEMANTIC OR META MODEL?
WHAT IS THE USE OF ABSTRACT DATA TYPES?
DATA BASE APPLICATIONS IN HUMANITIES
DATA BASE RESEARCH PROJECTS AND FUNDING

TECHNICAL SESSIONS

SEMANTICS AND MODELING I
SEMANTICS AND MODELING II
"VERY SMALL" DATA BASES
INTERFACES AND ARCHITECTURE
DATA BASE MACHINES I
DATA BASE MACHINES II
REALITY, USER REQUIREMENTS AND LOGICAL DATA BASE DESI
RELATIONAL VIEWS.
HIGH LEVEL USER INTERFACES
DATA BASE SOFTWARE
DISTRIBUTED DATA BASE SYSTEMS
PHYSICAL DESIGN TECHNIQUES I
PHYSICAL DESIGN TECHNIQUES II
EQUIVALENCE AND TRANSFORMATION

図４－２８　第４回ＶＬＤＢのセッション一覧

4.3.9 ま と め

以上欧米の4カ国7機関の調査概要及び二つの国際会議について報告した。訪問先のほとんどが大学及び公的研究所であり、強力なスタッフと優れた環境の中での意欲的な研究姿勢がうかがわれ、施設等も羨む限りである。当研究開発のテーマである分散型データベースシステムの現状は、まだ研究開発段階にあり評価方式云々以前に実現化の問題に直面している。INGRES, LADDER, FIDAS, IPS, PRTVの各システムは既に稼働しているが、分散型データベースとしては実働していない。POLYPHEMEは1979年6月にプロトタイプの完成を目指し、またIRIAのSIRIUSプロジェクトやGMDのINGVERプロジェクトによるシステムも実現化の方向に向って積極的に進められている。

複数(同種/異種)のDBMSをコンピュータネットワークを介して統合利用するためには、未解決の技術的問題が多い。特に、異種のDBMSを統合するには共通モデルや共通言語の設定、分散問題や異種性問題を解決するそれぞれの機能が必要であることはいうまでもない。グルノーブル大学のM.Adiba教授は、「異種のDBMSの統合が実現されれば、パフォーマンスロスも多いと思われるがそれは支払うべき代償であり、統合することにより受ける恩恵の方がはるかに高い。」と述べている。

システムの評価尺度に関しては、ほとんどの研究者がシステムの長寿命性、システムの信頼性、通信コスト/通信量、応答性を指摘している。今回の海外調査で、研究開発の第一線にいる研究者達と行った技術討論及び二つの国際会議に出席したことは、当研究開発に多くの技術的成果をもたらすと共に、大いに刺激となり研究開発の意義も再認識するに至った。

○調査担当メンバー

浜中栄治, 滝沢誠・開発部

付記1. 入手資料

UC. Berkeley

〔MCDO 74〕
McDonald, N., and Stonebraker, M., "CUPID-The Friendly Query Language," UC.Berkeley, No. ERL-M487, Oct. 1974.

〔GO 75〕
Go, A., Stonebraker, M., and Williams, C., "An Approach to Implementing a Geo-date System," UC.Berkeley, No. ERL-M 529, June 1975.

〔HELD 75〕
Held, G., and Stonebraker, M., "B-Trees Re-Examined," UC.Berkeley, No. ERL-M 528, July 1975.

[ALLM 75]
Allman, E., Stonebraker, M., and Held, G., "Embedding a Relational Data Sublanguage in a General Purpose Programming Language," UC.Berkeley, No. ERL-M 564, Oct. 1975.

[WONG 76]
Wong, E., and Youssefi, K., "Decomposition - A strategy for Query Processing," UC.Berkeley, No. ERL-M 574, Jan. 1976.

[PREN 77]
Prenner, C. J., "A Uniform Notation for Expressing Queries," UC. Berkeley, No. UCB/ERL M77/60, Sept. 1977.

[PREN 78]
Prenner, C. J., and Rowe, L. A., "Programming Languages for Relational Data Base Systems," UC. Berkeley, No. UCB/ERL M78/54, July 1978.

[EPST 77a]
Epstein, R., "A Tutorial on INGRES," UC. Berkeley, No. UCB/ERL M77/25, Dec. 1977.

[EPST 77b]
Epstein, R., "Creating and Maintaining A Database Using INGRES," UC. Berkeley, No. UCB/ERL M77/71, Dec. 1977.

[EPST 78]
Epstein, R., Stonebraker, M., and Wong, E., "Distributed Query Processing in a Relational Data Base System," UC. Berkeley, No. UCB/ERL M78/18, April 1978.

[YOUS 77]
Youssefi, K., Whyte, N., Ubell, M., Ries, D., Hawthorn, P., Epstein, B., Berman, R., and Allman, E., "INGRES Reference Manual---Version 6.1, "UC.Berkeley, No. ERL-M 579, Dec. 1977.

[YOUS 78]
Youssefi, K., and Wong, E., "Query Processing in A Relational Database Management System," UC.Berkeley, No. UCB/ERL M 78/17, March 1978.

[RIES 78]
Ries, D., and Epstein, R., "Evaluation of Distributed Criteria for Distributed Data Base Systems," UC.Berkeley, NO. UCB/ERL M 78/22, May 1978.

[STON 75]
Stonebraker, M., "Implementation of Integrity Constraints and Views by Query Modification," UC.Berkeley, No. ERL-M 514, March 1975.

[STON 76a]
Stonebraker, M., Wong, E., Kreps, P., and Held, G., "The Design and Implementation of INGRES," UC.Berkeley, No. ERL-M 577, Jan. 1976.

[STON 76b]
Stonebraker, M., and Rubinstein, P., "The INGRES Protection System," UC.Berkeley, No. ERL-M 594, July 1976.

[STON 77]
Stonebraker, M., and Rowe, L.A., "Observations on Data Manipulation Languges and Their Embedding in General Purpose Programming Languages," UC.Berkeley, No. UCB/ERL M 77/53, July 1977.

[STON 78a]
Stonebraker, M., "A Distributed Data Base Machine," UC.Berkeley, No. UCB/ERL M 78/23, May 1978.

[STON 78b]
Stonebraker, M., "Concurrency Control and Consistency of Multiple Copies of Data in Distributed INGRES," UC.Berkeley, No. UCB/ERL M 78/24, May 1978.

S R I

[SRI 78]
SRI, LADDER system - Demonstration, 1978.

C N R S

[CNRS 78a]
CNRS, "Bulletin Signalétique," Vol.39, No.5, 1978.

[CNRS 78b]
CNRS, "Bulletin Signalétique 1978-PASCAL Plan de classement," 1978.

[CNRS 78c]
CNRS, "Diffusion Sélective Information," 1978.

[CNRS 78d]
CNRS, "Systéme Pascal; PASCALINE, Bibliothèque, Traduction, Outils Documentaires Tarif 1978, Bandes Magnétiques Pascal, Diffusion Sélective de I'information Tarif 1978, Bibliographie Rétrospective Tarif 1978, Guide du lecture," 1978.

[BUFF 77]
Buffet, P., "PASCALINE, A Multidisciplinary European Data Base," 1st International On-Line Information Meeting, London, Dec. 1977. pp. 143-149.

I R I A

[IRIA 77]
IRIA, "Rapport du Conseil d'Administration"

[IRIA 78a]
IRIA, "Compte-Rendu d'Active 1977 des Equipes SIRIUS," Février 1978.

[IRIA 78b]
IRIA, "Bulletin de Liaison de la Recherche en Informatique et Automatique," Juin 1978.

[LITW 78a]
Litwin, W., "Virtual Hashing : A Hashing Which may be Dynamically Changed," CIPS Session '78, Alberta, Canada, May 1978.

[LITW 78b]
Litwin, W., "Virtual Hashing : A Dynamically Changing Hashing," 4th International Conference on VLDB, Berlin, Sept. 1978.

[PARE 77]
Parent, C., "Integrity in Distributed Data Bases," INT-I-002, AICA '77

[ADIB 78]
Adiba, M., Caleca, J. Y., and Euzet, C., "A Distributed Data Bases System Using Logical Relational Machines," Draft paper of VLDB 78, February 1978.

[HOLL 77]
Holler, E., "Report : French-German Seminar on Distributed Data Bases," November 1978.

Univ. of Grenoble

[ADIB 78a]
Adiba, M., "Modelling Approach for Distributed Data Bases,"
Laboratoire d' Informatique, USMG, Grenoble, Jan. 1978.
(to appear in Proc. of Information Systems Methodology, Venice, Oct. 1978.)

[ADIB 78b]
Adiba, M., "Un Modele Relationnel et Une Architecture Pour Les Systemes de Bases de Donnees Reparties, Application au Project POLYPHEME,"
USMG, Grenoble. Sept. 1978.

G M D

[GMD 77a]
GMD, "Wissenschaftlicher Eregebnisbericht 1977"

[GMD 77b]
GMD, "Datenbanksystem FIDAS Kurzbeschreibung," Institut für Informationssystem (IIS), 1977.

[GMD 78a]
GMD, "Datenbanksystem FIDAS Einführung," Institut für Informationssystem (IIS), 1978.

[GMD 78b]
GMD, "Mittrtlungen für GMD-FIDAS-Anwender, Nr. 9," June 1978.

[BREI 78]
Breitwieser, H., Drobnik, O., Keil, C., and Kersten, U., "Architektur und Operationales Modell eines Verteilten Dateiverwaltungssystems," Kernforschungszentrum Karlsruhe, Institut für Datenverarbeitung in der Technik, Karlsruhe, Bundesrepublik, 1978.

[CHRI 77]
Christiansen, S., Voss, K., and Wurch, G., "Bericht VERBIS 9, An Approch to Data Base Systems, Interworking," February 1977.

[CHRI 78a]
Christiansen, S., and Wurch, G., "Bericht VERBIS 14, Decomposer, Zerlegung von Relationalen Ausdrücken und Versorgung der Control Table mit Informationen, January 1978.

[CHRI 78b]
Christiasen, S., and Wurch, G., "Bericht VERBIS 17, Decomposer," July 1978.

G L C

〔GLC 76a〕
GLC, "Greater London Services 1976-77,"

〔GLC 76b〕
GLC, "GLC User Guide for the IBM Interactive Planning System (IPS)," GLC, October 1976.

〔GLC 76c〕
GLC, "IPS Reference Mannual; Language Syntax Definition"

〔GIBS 74〕
Gibson, C.F., and Nolan, R.L., "Managing the Four Stages of EDP Growth," Harvard Business Review, January-February 1974.

〔POST 77〕
British Post office, "Viewdata; A New Information Service from the British Post office," September 1977.

付記2. 主要面接者

UC. Berkeley

Prof. M. Stonebraker
UC. Berkeley, Electronics Research Laboratory.

Dr. E. Allmal
UC. Berkeley, Electronics Research Laboratory.

S R I

Dr. D. Sagalowicz
SRI, International, Artificial Intelligence Center.

C N R S

Dr. Y. Salle (女 史)
CNRS, Informascience, Chargés de mission

Dr. P. Buffet
CNRS, Informascience, Informatique

Dr. Y. Henry (女史)
CNRS, Informascience

I R I A

Dr. W. Litwin

Univ. of Grenoble

Prof. M. Adiba.
Laboratoire d'Informatique, Univ. of Grenoble.

Dr. C. Delobel
Laboratoire d'Informatique, Univ. of Grenoble.

G M D

Dr. K. Voss.
GMD, Chief, Chairman for EDP-systems Application in National Administrations.

S. Christiansen (女 史)
GMD, INGVER project.

G. Wurch
GMD, INGVER project.

Dr. H. Scholev
GMD, FIDAS Consultant Manager.

G L C

Dr. A. E. Mitson
GLC, Head of Computing Services

Dr. C. Doggrell
GLC, Computing Services

H. Kon （女史）
GLC, Computing Services, IPS Manager

5　論文・資料紹介

本プロジェクトは昭和５２年度から3年計画で遂行されている。本章では、現在までに外部に発表した論文、報告書、とりまとめた内部資料等について紹介する。特に、外部発表論文に関しては、投稿中もしくは投稿予定の論文も合わせて紹介する。

配布用資料（外部発表論文、報告書等）

1）分散型リソース処理技術の研究開発

昭和53年3月　報告書No. 52−S 001

2）ＪＩＰＮＥＴにおけるリソース統合システム

昭和53年8月　第19回情報処理学会全国大会

3）Resource Integration and Data Sharing on Heterogeneous Resource Sharing System

昭和53年9月　ＩＣＣＣ'７８（第4回大会）

4）欧・米の分散型リソース処理技術

昭和53年12月　ＪＩＰＤＤＣジャーナル　No. 36

5）The Four−Schema Concept as the Gross Architecture of Distributed Databases

昭和53年12月　投稿中

6）分散型リソース処理技術の研究開発（本報告書）

昭和54年3月　報告書　No. 53−S 001

7）4層スキーマ構造に基づいた分散型データベースシステム

昭和54年7月（予）　第20回情報処理学会全国大会

1）分散型リソース処理技術の研究開発（52−S001）

この報告書は、昭和５２年度から3ケ年計画で発足した「分散型リソース処理技術の研究開発」の第一年度目の報告書であり、分散処理、データ共有方式の検討、日本語情報処理の3部より構成されている。

第Ⅰ部の分散処理は、分散処理の一般論を述べている。第1章では分散処理の出現の背景を述べ、分散処理の定義をおこなっている。第2章では、ネットワークユーザの分類をおこない、ユーザの視点の重要性を論じている。第3章においては、ネットワークで共有されるリソースを検討し、データが特に重要なリソースであることを指摘している。

第Ⅱ部では、分散処理におけるデータ共有方式について議論している。まず第1章では、ユーザの視点からみた分散処理技術に関して、第Ⅰ部で分類を試みた分散処理関係者の各層に対応し

たシステムをそれぞれ紹介している。第2章では、データを管理する強力な道具としてデータベース管理システム（ＤＢＭＳ）をとりあげ議論している。第3章では、ネットワーク技術とデータベース技術を有効に組み合わせて実現している分散型データベースについて議論している。第4章では、ネットワーク上に散在するリソースを統合していく上で重要な機能の一つであるネットワークリアリングセンターについて議論している。特に、リソースの所在源情報管理の基本的技術としてＤＤ／Ｄに焦点をあて、ＤＢＭＳの結合形態やディレクトリの配置や管理方法について言及している。第5章では、異機種のコンピュータから構成されたコンピュータネットワークを利用するユーザが、まず直面する問題である制御言語の多様性を解決する技術について議論している。つまり、コンピュータの異機種を克服するために設定した標準制御言語のシステムを示している。第6章では、第5章までの検討に基づいて、異種コンピュータネットワークにおけるリソース統合問題を採り上げている。

2) ＪＩＰＮＥＴにおけるリソース統合システム（〝はじめに〃より引用）

当財団は、昭和48年度から4年間に渡り、リソースシェアリングコンピュータネットワークＪＩＰＮＥＴを開発した。ＪＩＰＮＥＴにより国産の3機種による異機種間のリソースシェアリングを実現した。当研究開発では、更に一歩進めて異機種コンピュータネットワークにおけるリソース統合問題を検討している。本論文では、ＪＩＰＮＥＴにおけるリソース統合システムとして、異種のＤＢＭＳから成る分散型データベースシステムを採り上げ、解決すべき問題点、システムの持つべき機能について検討を行なう。

3) Resource Integration and Data Sharing on Heterogeneous Resource Sharing System

There are two problems to discuss regarding the resource sharing system employing computer networks such as ARPANET and CYCLADES; resource heterogeneity problems accompanying the incorporation of various host computers, and distribution problems problems posed by resources combined through networks. In this paper, to solve the two above-mentioned problems, we classify existing approaches and then propose our approach. Our approach consists of two stages of conceptualization. First, the heterogeneity problem is removed by translating local models into common model. Secondly, the distribution problem is removed by integrating the common models for each site to the composed resource of interest to a network community. We apply this approach to the general-purpose distributed data base on networks.

4）欧・米の分散型リソース処理技術

当研究開発の一貫として昭和５３年８月下旬から約3週間に渡り調査訪問した欧米４ケ国計7つの先進的研究機関（大学及び研究所.）における調査概要について述べている。今回の調査は、分散型リソース処理技術及び評価技術とデータの収集を目的として行なった。報告の最後に、実用化への技術的問題点について述べている。

5）The Four－Schema Concept as the Gross Architecture of Distributed Databases

This paper presents a gross architecture of distributed data－bases based on four-schema structure (FSS). FSS is composed of local internal, local conceptual, global conceptual and external schemas. This paper discusses the process of translation of local internal schema into local conceptual schema, and the integration of local conceptual schemas into a global conceptual schema, and also the qucry dccomposition and translation.

In this paper, we try to show the possibility of organizing heterogeneous DBMS into a cooperative system through computer net-works. We discuss the homogenization of existing DBMS, and then the process of their conceptualization (or integration). The qucry translation problem in also discussed.

6）分散型リソース処理技術の研究開発（53－S001）

この報告書は、昭和５２年度から３ケ年計画で発足した「分散型リソース処理技術の研究開発」の第二年度目の報告書であり、分散型データベースシステム、日本語情報処理の２部より構成しされている。

第Ｉ部は分散型データベースシステムに関する研究開発についてとりまとめたものであり５つの章からなっており、それぞれ次のような内容について述べている。第１章では、当研究開発の背景とその目的について述べている。第２章では、分散及び異種性問題を解決するために設定した４層スキーマ構造に基づく分散型データベースシステムのアーキテクチャについて述べている。第３章では、第２章で示したシステムをサポートするネットワークデータディレクトリが保持する情報を生成する、クリアリング情報生成システムについて述べている。第４章では、海外調査に基づいて分散型リソース処理技術の技術動向について述べている。第５章では、当研究開発でとりまとめた論文・資料について紹介している。

7） 4層スキーマ構造に基づいた分散型データベースシステム

昭和５３年度に行なった研究成果を中心に発表する。２及び3章で述べているように、現在データリソースを主体にしたリソース統合システム、即ち、異種のＤＢＭＳから成る分散型データベースシステムの研究開発を行なっている。我々は、異種及び分散問題を解決するために４層スキーマ概念を導入した。本論文ではこの４層スキーマ構造を中心に、ＮＤＤ（ネットワークデータディレクトリ）等の関係も含めて発表する予定である。

内部資料

1） 海外調査概要報告書

昭和５3年１１月

2） クリアリング情報生成システム／異種性情報生成サブシステム（ＮＤＤ／ＨＩＰＳ）基本設計書

昭和５３年１２月

3） ＮＤＤ／ＨＩＰＳプログラム仕様書

昭和５４年２月

4） クリアリング情報生成システム／分散情報生成サブシステム（ＮＤＤ／ＤＩＰＳ）基本設計書

昭和５３年１２月

5） ＮＤＤ／ＤＩＰＳプログラム仕様書

昭和５４年３月

6） Query Translation

昭和５３年１２月

7） Query Processing

昭和５４年１月

参考文献

〔ADIB 78a〕
Adiba, M., Chupin, J.C., Demolombe, R., Gardarin, G., and Le Bihan, J.,
"Issues in Distributed Data Base Management Systems : A Technical Overview,"
Proc. of the 4th International Conf. on VLDB. Berlin, Sept. 1978, pp. 89-110.

〔ADIB 78b〕
Adiba, M., and Euzet, C.,
"A Distributed Data Base System Using Logical Relational Machines,"
Proc. of the 4th International Conf. on VLDB, Berlin Sept. 1978, pp. 450-461.

〔ADIB 78c〕
Adiba, M., "Modelling Approach for Distributed Data Bases,"
Laboratoire d'Informatique, USMG, Grenoble, Jan. 1978.
(to be published in ECI-78, Venice, Italy, Oct. 1978)

〔BEER 78〕
Beeri, C., Bernstein, P.A., and Goodman, N.,
"A Sophisticate's Introduction to Database Normalization Theory,"
Proc. of the 4th International Conf. on VLDB, Berlin, Sept. 1978, pp. 113-124.

〔CHEN 76〕
Chen, P.P.S., "The Entity-Relationship Model - Toward a Unified View of Data,"
ACM TODS. Vol. 1, No. 1, March 1976, pp.9-36.

〔CODD 70〕
Codd, E.F., "A Relational Model of Data for Large Shared Data Bank," Communications of the ACM, Vol.13, NO.6, June 1970, pp. 337-387.

〔DATE 77〕
Date, C.J., "An Introduction to Data Base System," (Second Edition),
Addison-Wesley, June 1977, 536p.

〔DEO 74〕
Deo, N.,
"Graph Theory with Applications to Engineering and Computer Science,"
Prentice-Hall, 1974, 478p.

〔HAMA 78〕
Hamanaka, E., Takizawa, M., and Ito, T.,
"Resource Integration System Using JIPNET," (Japanese)
Proc. of the 19th Annual Convention of Information Processing Soc. Japan, Aug. 1978.

〔HEND 77〕
Hendrix, G.G., Sacerdoti, E.D., Sagalowicz, D., and Slocum, J., "Developing a Natural Language Interface to Complex Data," Proc. of the 3rd International Conf. on VLDB(Vol.Ⅱ), Tokyo, Oct. 1977, pp. 37-58. (also available in ACM TODS, Vol.3, No.2, June 1978, pp. 105-147)

〔HEVN 78〕
Hevner, A.R. and Yao, S.B.,
"Query Processing in a Distributed System,"
to appear in IEEE Trans. on Software Engineering, 1978.

〔JIPD 78〕
日本情報処理開発協会, "分散型リソース処理技術の研究開発"
52－S－001, 昭和53年3月, 582P.

〔KNUT 73〕
Knuth, D,E., "Sorting and Searching,"
The Art of Computer Programming,
Addison-Wesley, Vol.3, 1973.

[KNUT 75]
Knuth, D.E., "Fundamental Algorithms," (2nd. edition), The Art of Computer Programming, Addison-Wesley, Vol.1., 1975.

[MUNZ 78]
Munz, R., "System Architectures for Managing Distributed Databases," GI-Fachtagung "Datenbanken in Rechnernetzen mit Kleinrechhern," Karlsruhe, April 1978.

[NEUH 77]
Neuhold, E.J., and Biller, H., "POREL : A Distributed Data Base on an Inhomogenecus Computer Network," Proc.of the 3rd International Conf. on VLDB, Tokyo, Japan, Oct. 1977, pp. 380-395.

[ROTH 77]
Rothnie, J.B. and Goodman, N., "An Overview of the Preliminary Design of SDD-1: A System for Distributed Databases," Proc. of the 2nd Berkeley Workshop on Distributed Data Management and Computer Networks, UC. Berkeley, California, May 1977, pp. 39-57.

[SAGA 77]
Sagalowicz. D., "IDA : An Intelligent Data Access Program," Proc. of the 3rd International Conf. on VLDB, Tokyo, Oct. 1977, pp. 293-302.

[SHAW 74]
Shaw, A.C., "The Logical Design of Operating Systems," Prentice-Hall, Inc., 1974.

[STON 76]
Stonebraker, M., Wong, E., Kreps, P., and Held, G., "The Design and Implementation of INGRES," ACM TODS, Vol. 1, No. 3, Sept. 1976, pp. 189-222.

[STON 77]
Stonebraker, M., and Neuhold, E., "A Distributed Data Base Version of INGRES," Proc. of the 2nd Berkeley Workshop on Distributed Data Management and Computer Networks, UC. Berkeley, California, May 1977, pp. 19-36.

[TAKI 78]
Takizawa, M., Hamanaka, E., and Ito, T., "Resource Integration and Data Sharing on Heterogeneous Resource Sharing System," Proc. of ICCC' 78, Kyoto, Japan, Sept. 1978, pp. 253-258.

[TAKI 79]
Takizawa, M., and Hamanaka, E., "The Four-Schema Concept as the Gross Architecture of Distributed Databases," JIPDEC Project Report 540001, Jan. 1979.

[TSIC 78]
Tsichritzis, D. and Klug, A. (eds.), "The ANSI/X3/SPARC DBMS Framework," Information Systems, Vol.3, No.3, 1978, pp. 173-191.

[WONG 77]
Wong, E., "Retrieving Dispersed Data from SDD-1: A System for Distributed Databases," Proc. of the 2nd Berkeley Workshop on Distributed Data Management and Computer Networks, UC. Berkeley, May, 1977, pp. 217-235.

[YOUS 77]
Youssefi, K., Whyte, N., Ubell, M., Ries, D., Hawthorn, P., Epstein, B., Berman, R., and Allman, E., "INGRES Reference Manual-Version 6.1," Memo ERL-M579, Electronics Research Lab., UC. Berkeley, Oct. 1977.

略 語 表

AOP	算術演算子(arithmetic operator)
CGDP	統合GDP(coordinate GDP)
CI	対応情報(correspondence information)
COP	比較演算子(comparison operator)
CS	対応情報スキーマ(CI schema)
DB	データベース(database)
DBMS	データベース管理システム(database management system)
DD	分散記述(distribution description)
DDB	分散型データベース(distributed database)
DDL	データ定義言語(data definition/description language)
DD−tree	分散記述木(distribution description tree)
DI	分散情報(distribution information)
DIPS	分散情報生成サブシステム(distribution information processing system)
DML	データ操作言語(data manipulation language)
ES	事象集合(entity set)
ESR	事象集合リレーション(entity set relation)
ESS	事象集合スキーム(entity−set seheme)
EXS	外部スキーマ(external schema)
E−R	事象関係性(entity−relationship)
FSS	4層スキーマ構造(four−schema structure)
GA	全体管理者(global administrator)
GCS	全体概念スキーマ(global conceptual schema)
GDP	全体データプロセッサ(global data processor)
GLQD	全体−局所問合せ分割(global−local query decomposition)
HI	異種性情報(heterogeneity information)
HIPS	異種性情報生成サブシステム(heterogeneity information processing system)
HQD	水平問合せ分割(horizontal query decomposition)
LA	局所管理者(local administrator)
LCS	局所概念スキーマ(local conceptual schema)
LDP	局所データプロセッサ(local data processor)
LI	所在情報(location information)
LICF	局所完全性条件式(local integrity constraint formula)
LIS	局所内部スキーマ(local internal schema)
LLQD	局所−局所問合せ分割(local−local query decomposition)

ＬＬＱＴ	局所－局所問合せ変換（local－local query translation）
ＬＯＰ	論理演算子（logical operator）
ＬＱＰ	局所問合せ処理（local query processing）
ＬＳ	所在情報スキーマ（ＬＩ schema）
ＮＤＤ	ネットワークデータディレクトリ（network data directory）
ＯＳＣ	サイト内句（one－site clause）
ＱＤ	問合せ分割（query decomposition） （本文中では、全体－局所問合せ分割（GLQD）を意味する）
ＱＴ	問合せ変換（query translation） （本文中では、共通－固有の問合せ変換、即ち、ＬＬＱＤとＬＬＱＴの総称である）
Ｑ－tree	問合せ木（query tree）
ＲＤ	リレーショナル記述（relational description）
ＲＳ	関係性集合（relationship set）
ＲＳＲ	関係性集合リレーション（relationship set relation）
ＲＳＳ	関係性集合スキーム（relationship set seheme）
ＳＩ	スキーマ情報（schema information）
ＳＬ	セマンティックリンク（semantic link）
ＳＳ	スキーマ情報スキーマ（ＳＩ schema）
ＴＳＣ	サイト間句（two－site clause）
ＶＱＤ	垂直問合せ分割（vertical query decomposition）
ＶＳ	値集合（value set）

Ⅱ　日本語情報処理技術の研究開発

1. 総論 …… 189
1.1 研究の背景と目的 …… 189
1.2 概要 …… 192
2. 日本語情報処理技術の課題 …… 194
2.1 技術動向 …… 194
2.2 最近の話題 …… 195
2.2.1 カナ入力方法（タッチ打法） …… 195
2.2.2 日本語ワードプロセッサ …… 198
2.2.3 画像通信システム …… 201
2.2.4 その他の話題 …… 202
2.3 分散処理用端末 …… 204
2.3.1 米国における情報サービスの発展 …… 205
2.3.2 通信回線開放への動き …… 206
2.3.3 技術的課題 …… 207
3. 日本語端末用文字発生方法の検討 …… 212
3.1 文字発生方式 …… 212
3.2 パターンの圧縮方法 …… 213
3.3 ランレングスコード方式 …… 215
3.4 行パターンによる圧縮方式 …… 225
3.5 ラインドット方式 …… 226
3.6 ハフマンの符号化法 …… 228
3.7 圧縮方式の検討 …… 240
4. 日本語端末用入力方法の検討 …… 243
4.1 入力方式の分類 …… 243
4.2 漢字仮名入力システムの概要 …… 243
4.2.1 システムの目的 …… 243
4.2.2 システムの構成 …… 244
4.2.3 入力変換サブシステム …… 246
4.2.4 変換辞書作成サブシステム …… 255
4.3 変換辞書の分析 …… 260
4.3.1 項目変換辞書 …… 260
4.3.2 カナ漢字変換辞書 …… 264
4.3.3 カナ入力変換辞書 …… 265

4.4 システムの評価検討 …… 268
5. 日本語端末の基本設計 …… 274
5.1 日本語端末の設計方針 …… 274
5.2 日本語情報処理システムの構成 …… 274
5.3 日本語端末の構成 …… 276
5.3.1 基本構成 …… 276
5.3.2 付加構成 …… 276
5.3.3 構成機器の概要 …… 277
5.4 日本語端末の特徴 …… 281
5.5 日本語端末の処理概要 …… 282
5.5.1 入力制御機能 …… 282
5.5.2 文字発生機能 …… 285
5.5.3 出力制御機能 …… 291
5.6 圧縮文字パターンの作成 …… 294
5.6.1 圧縮方法 …… 294
5.6.2 圧縮文字パターン作成処理 …… 297
5.7 入力変換辞書の構成 …… 301
付記1. 漢字出力システムの概要 …… 303
付記2. 日本語端末用文字種・コード表 …… 333
付記3. 文字パターン出力例 …… 356
付記4. マイクロコンピュータ機器構成 …… 363
参考文献 …… 369

1　総　　論

1.1　研究の背景と目的

最近のコンピュータ技術、周辺・端末装置の発展はあらゆる分野においての情報処理のコンピュータ化を急速に拡大してきた。従来そこにはコンピュータ処理といえば演算処理が中心となって利用されてきたが、適用分野の拡大によって一般事務処理へとコンピュータ化が進んでくると、コンピュータの出力結果に対する不満が生じて来る。それは日本人であるがゆえに、コンピュータから出力される結果といえども、国語の表記文字である漢字かなまじりの日本語情報が出力できるようでないと価値のない情報になりかねない。これまで欧米で開発されたコンピュータだから、あるいは計算をするための機械であるからと、コンピュータから出力される文字は、英字、数字と一部記号の扱いが一般的で、日本人向けにはカタカナの使用を許しているにすぎなかった。また逆にみれば、出力文字への不満よりも処理面の効率追求に力を注いだために、今日のコンピュータの進歩をもたらしてきたけれども、コンピュータの処理効率の限界に近づいてきたこともあって、ようやくというより日本人だからこそ出力文字の要求が必然的に高まってきたといえる。

日本人にとって、一部の知識人を除けば当然のことながら外国語で書かれた文章よりも日本語で書かれた文章のほうが理解しやすく、情報伝達も正確にスムーズに行なわれる。ところがコンピュータからの出力リストが英字と数字で書かれ、これをレポートとしてあるいは情報サービスの提供物として渡されたりしたら、受取人は読むことさえもわずらわしく感じる。しかたなく読解しようにも、注釈が日本語文で書かれていない限りこれまた困難で、無意味な情報として破棄されかねない。このようにわが国の情報処理においては、日本語情報の処理は必須条件であり、日本語情報処理技術の難題を少しずつでも解決して行かなければならない状況にある。

日本語情報を処理するための入出力装置、いわゆる漢字入出力装置の開発は、コンピュータ周辺・端末装置の急速な発展と機能向上の中で、あいかわらず低速度の進行状態である。そこにはあらゆる面で文字種の多さによる原因がある。普及の最重点的要素である入出力装置の価格が、従来の英数字カナ文字用に比べてあまりにも高価であったこと、漢字入力・データ作成に費用が多くかかること、などすべてが多字種によるものである。

日本語を構成している文字は、ひらがなやカタカナをはじめ、新旧漢字・俗字などを含めると最大5万字種近くあると言われている。もっとも一般的な制定漢字として当用漢字があるが、この制限枠内の漢字でさえも1,８５０字種である。また、新聞・雑誌に使用されている漢字の種類は、およそ3,０００字種前後といわれ、日本人の姓名に至っては約1万字種を使用しないと書き現わすことができない。

このように、欧米諸国での使用文字種が数十種というのに比較して、日本ではその数十倍から数百倍の文字種を扱わなければならないというギャップがある。このことは、コンピュータ処理において文字コード・配列順の問題が生起して、入力側では入力キーボードへの文字配列の問題が、出力側では文字発生機構の複雑化等の問題が派生してくる。こうした日本語情報の取扱い上の問題が

ある中で、日本語情報（漢字）処理の先駆的役割を果したのが、新聞・通信業界であった。当業界では昭和３４年に文字種・文字コードの制定を行なって（ＣＯ－５９コード）、新聞記事・原稿の通信印刷の手段に漢字入出力機器を使用するようになった。この当時開発されたのが漢字テレタイプライタ（漢テレ）であり、２千数百の漢字を一面のキーボード上に配置したフルキーボード型の入力機器が用いられた。

当初は、漢字入出力機器の製作・販売は数メーカに限られていたが、大手コンピュータ・メーカをはじめ周辺機器メーカもあいついで漢字処理システム機器の開発に参入してきた。その数は、現在約３０社にのぼる。その結果、出力装置に関しては従来のラインプリンタ以上の機能を有した、高速・高精度の装置が開発されて、ユーザの利用目的に合った選択ができるようになってきた。しかし一方の入力装置に至っては、現在でも相変らずフルキー方式が主流を占めている。両手を使った２打けん式の入力操作から、和文タイプ型、タブレット型、ペンタッチ型へと操作が容易になって素人でも扱えるようになり、記録媒体も紙テープから、カセットテープ、フロッピー・ディスクなどへと後処理がしやすいように変ってきたが、数千字の漢字が配置された盤面から目的とする漢字を探す方法はほとんど変わっていない。また操作性の向上を目指して、ローマ字入力・カタカナ入力方式も研究開発されてきたが同音異義語（同音異字）、分ち書きの問題をかかえて実用化されるまでに至っていない。

最近注目されている入力方式は、タッチ打法として最高速の入力が可能とされている連想コード方式である。カナあるいは英字を、各漢字に２字対応させて入力する方法で、キーボードに英数カナ・キーボードあるいはそのキートップだけを特殊化した形態でそのまま使用できるという簡便さがある。約５０キーの操作が容易であるばかりか、各漢字への２字の符号化の方法がユニークで、パンチャの最も連想しやすいあるいは覚えやすいカナを割り当てているために高速化がはかられた。しかしこの方式にも、数千字種分の符号を覚える教育・訓練の問題があって、現在議論のまっただなかにある。ラインプット、ＫＡＮＴＥＸ、ＫＩＳ、大計方式などと呼ばれているのがこれに類する方式である。

上記の文字種のギャップは、日本語情報処理用の各装置の価格を高額にしている。とくに出力装置に至っては、数千万円規模のものが多く、普及をさまたげている大きな原因になっている。出力装置の高価格化の最も大きな要因は文字種の多さであり、この要因は出力装置の主要機構である文字発生器の規模を大型化している。ディジタル型にしてもアナログ型の文字発生方式にしても、出力のために文字パターン（フォント）を字種分記憶（記録）しなければならず、英数字用の文字発生にくらべ、これだけでも１００倍の文字種分が必要である。それに加えて、漢字のパターンは複雑で、字画、点素でとらえても漢字表現のためには多くのパターン情報が必要である。漢字を格子に通して点素（ドット）に分解しようとすれば、約１万字種を表現しようとすると最低２０（幅方向ドット数）×２０（高さ方向ドット数）以上、標準的には２４×２４あるいは３０×３０、３２×３２といったドット構成が多い。２４×２４ドットで５７６ビット、英数文字を５×７ドットで３５ビットとしても、記憶するメモリ容量は５７６ビット×１０,０００字＝5,７６０,０００ビット、

３５ビット×６４字＝2,240ビット、となっておよそ2,500倍のちがいがある。文字発生器のメモリ・コストもこの差がそのまま現われることになるが、実際には、他の要素（字種を制限したり）があるため数倍から数十倍程度の差になっている。文字発生器のメモリ・コストは、そのまま漢字出力装置のコストに大きくえいきょうし、さらに日本語情報処理全体のコストにも及ぶことになってメーカ各社の苦心がここにある。ディジタル型の文字発生器による場合、ディジタル・メモリのコストが主要であり、その低廉化がはかれることが期待される。最近の半導体素子、ＬＳＩ技術の進歩によって、高密度化、大容量化がはかられつつあり、低廉化も進んでいる。また、周辺記憶装置もますます大容量化が進み、ビット当りの単価も次第に下がってきている。さらにプロセッサの機能向上と、ＬＳＩ化による低廉化も進んで、ディジタル型の文字発生器がもっとも有望な方式とみられている。

ディジタル型の文字発生器への期待は、さらに文字パターンの圧縮技術によって倍加する。現在のところ文字パターンの圧縮技術を優先させるか、ディジタル・メモリの低価格化を率先させるか判断に迷う段階である。実際に低価格をうたった漢字出力装置の例をみてもパターン圧縮によるものと、低価なメモリを採用したもの、とに大別される。そして両者の制御には、機能面からミニコンピュータを使用し、マルチステーション化によって低価格を前面に出す場合と、機能を多少落しても安価なマイクロコンピュータを導入して低価格を打ち出す場合がある。ここには、発展途上にあるマイクロコンピュータが、どこまで制御することができるか判断しにくい面がある。マイクロコンピュータも汎用コンピュータとみなすことができるか、あるいは現在多くの例にあるように、各種の機器・装置に組込まれて利用されるようなコントローラ用のプロセッサとしかないと判断するかである。ミニコンピュータ・メーカが主記憶をＬＳＩ化してマイクロコンピュータとするケースもあればマイクロコンピュータの機能を向上させてミニコンピュータを代用するという双方から近づきつつある。しかし現段階では、汎用コンピュータというよりも、ＯＥＭ用の制御プロセッサといった方が適当のようである。とはいえ、マイクロコンピュータの出現は、日本語情報処理はもとより、他の周辺、端末機器、各種アプリケーション・システムにおいても、注目されることである。

今後の日本語情報処理の適用で期待されるのは、コンピュータ端末装置の漢字化であり、そこにも漢字入出力装置の低価格化が大きな課題としてとらえることができる。

一般の英数カナ文字用の周辺端末装置の分野でも、最近のコンピュータの普及にともない、低価格の機器・装置や、簡単に操作できるものへとその傾向を示しており、市場におけるニーズも当然その方向を示している。またバッチ処理ばかりでなく、オンライン化傾向が強まってきており、同時に高速処理、データ処理内容の向上、スループットの向上を可能とする製品の開発が進んでいる。とくにオンライン化への進展により、データの分散処理が必要となり、端末機器のインテリジェント化が急速に進んでいることが大きな傾向としてみられる。分散処理端末機は、専門でない人たちが必要に応じていつでも操作を行なうことが前提となり、だれでも容易に操作できる端末機が望まれる。

これらのことは、日本語情報処理においてもいえることで、これまで実例の少なかった日本語のオンライン処理も、低価格の漢字入出力装置が出現することによって大きく変化するにちがいない。

コンピュータ周辺端末装置の漢字化にあたって、装置を構成する漢字入出力機器に要求されることは、"簡易な入出力方式によって端末装置が構成でき、そこには操作性と処理機能の最適化をはかって、低価格の日本語端末を構築する"ことである。

低価格化の課題とともに、日本語端末に重要なことは、ソフトウェアの処理機能の最適化をはかる点である。日本語端末の構成と処理機能は強い相関にあり、構成要素の入出力機器の選択はソフトウェア機能の増減に大きくえいきょうする。簡易な入力方式の採用は、内部でのコード変換処理を増加させ、出力機能を多様化すればソフトウェアも増加するように、相反するものと同調するものがある。また端末自身ばかりでなく、ホスト（センタ）側の処理機能との関連も強く、端末のインテリジェント化とホストの集中処理が相関する。一般にホスト側にアプリケーション・システムがあってデータベースを有し、端末側には端末の入出力機能・制御機能を持つことが多い。このセンタと端末の処理機能は、合せもってこそ日本語情報処理が可能であり、一方のみが漢字処理ができても実動することはできない。たとえば、日本語のデータベースを持つホストの処理は、データベースのアクセスを端末からの問合せに対して行ない、その結果を端末へ送る、といった機能をもって、一方の端末では、送られてきた情報を利用者に出力提供する機能をもたなければならない。その場合、入力する情報は問合せ情報で少量であるが、応答情報はそれよりも多量なことが一般的である。その出力情報が日本語情報であった場合、漢字コード情報なのか漢字パターン情報であるか、あるいはまた英数カナ・コードであるかによってそれぞれの機能が大きくちがってくる。それは端末の機能によるところが大きく、文字発生機能、英数カナ・コードからの変換機能、などが端末のプロセッサに必要である。そしてこれはまた、伝送速度の問題もあり、一般には伝送するには簡易なコードで行ない、端末で文字発生（コードからパターンへの変換）するようでないと実用的でない。カナ・コード伝送では端末にソフトウェア負担が多くなりこれも実用的でない。

このようにセンタと端末との間に、処理機能を分担させることは、多数端末を接続するセンタのコンピュータ負荷を平均化するという効果がある。また端末の用途も、アプリケーションによって種々であり、センタと端末との機能分担も、適用分野ごとに設計する必要もある。その場合は、日本語端末の入出力構成も機能を十分発揮できるように構築しておかなければその効果を活用することはできない。

以上のようなことから、当研究開発では、コンピュータ端末装置としての日本語情報処理アプリケーション向け入出力方式の確立をはかることを目的にしている。日本語端末の最適な入出力構成と、その入出力制御方式を検討し、ユーザにとって操作のしやすい方式と満足のいく出力結果が得られるようにし、かつ低価格化をはかって日本語情報処理の需要拡大を目指すものである。

当事業「分散型リソース処理技術の研究開発」においては、わが国のリソースシェアリングネットワークの特色の1つとして日本語情報の処理機能が必要不可欠なことである。とくに今後のネットワーク・システムには、データベースが重要なリソースとして期待され、そこに日本語情報のデ

ータベースが構築されることになる。この日本語情報のリソースを利用するために、日本語端末が必要である。

3ケ年計画で開始した当研究開発において、初年度は、日本語情報処理技術の全体的な調査を行なって、日本語端末に必要な構成要素と基本機能について検討を行なった。本年第2年度においては、日本語端末の低価格化をはかる目的で、文字パターンの発生方式の機能検討と、簡易入力方式の機能検討を行なった上で、これらに基づいた日本語端末の基本設計を行なった。3ケ年計画の最終年度となる昭和54年度には、日本語端末の具体的なモデルとして、特定アプリケーション向けに端末機能を設定し、スタンドアロン型で機能するワードプロセッサ的なシステムを開発して、その実用性を実験評価する計画である。

1.2 概　　要

第Ⅱ部は、「分散型リソース処理技術の研究開発」の一環として推進している、日本語情報処理技術に関する研究開発についてとりまとめたものであり、日本語端末の機能検討および日本語端末のモデル設計を中心に記述している。

第Ⅱ部の構成は、5つの章と4つの付記からなっており、それぞれ次のような内容について述べている。

第1章「総論」では、当研究開発の背景とその目的、および3ケ年計画の第2年度までの経過報告と来年度の作業内容、本報告書の構成等について述べている。

第2章「日本語情報処理技術の課題」では、日本語情報処理の技術動向、最近の話題、および分散処理用端末の課題について述べている。

第3章「日本語端末用文字発生方法の検討」では、日本語端末の低廉化の1つとして、文字発生器のパターン・メモリ量を節減するために、パターンの圧縮復元技術の検討結果について述べている。とくに圧縮の期待値を求めるための統計分析を中心にまとめている。

第4章「日本語端末用入力方法の検討」では、日本語端末の入力操作性向上を目的にして、入力方式の比較検討のために作成した実験システム“漢字仮名入力システム”の概要と、変換辞書の統計分析、および評価検討について述べている。

第5章「日本語端末の基本設計」では、第3章の文字発生方法と第4章の入力方法の機能検討から、文字パターンの圧縮復元技術および項目入力とカナ漢字変換の両方式を導入した、日本語端末の基本設計について概説している。

その他、付記1には、漢字出力機能を検討するために作成した“漢字出力システム”の概要を、付記2には、日本語端末で扱う文字種・コード表一覧を、付記3には、文字パターン（ドット・マトリックス・パターン）の出力例を添付し、付記4には、本年度購入して機能実験したマイクロコンピュータの機器構成を概説している。

2　日本語情報処理技術の課題

2.1　技術動向

日本語情報処理システムは、この十余年の間に先駆的なユーザによる試行的段階から、多数ユーザによる実用の段階にまで発展してきた。この背景には多様なユーザのニーズと、それを支えるメーカの数多くの技術開発とが相まって、発展を遂げてきたことがある。この間の目的と原因およびその結果は複雑にからみ合っているが、よくいわれているようにデータ処理におけるデータ項目の漢字化、印刷出版界における編集植字の電子化、および日本語による情報提供サービスがその中心的役割を果してきた。

そしてその過程で量的および質的に処理内容の多岐化と広範化をもたらし、かつ高度な技法の適用を実現するという変化をもたらした。現段階はすでにハードウェアおよびソフトウェアの基本的開発を終り、特定の分野での応用から、さまざまな分野への適用の拡大の段階に入ったといえよう。このような技術の普遍化は当然のことながら、システムおよび機器の仕様の統一つまり規格化が重要な課題となってくる。後に詳しく述べるオンラインシステムにおける日本語端末装置などは、規格化が遅れているために、技術は先行しているにもかかわらず量産が進まないという現状である。特に現段階では各メーカおよびユーザが、もっと真剣に日本語情報処理の規格化に取組むべきであろう。

従来の適用分野をみるとデータ処理にしろ編集植字処理にしろ、日本語情報処理への必然性の高い分野から日本語化が始まったため、大量処理・高速処理への指向が強く、このためより高性能な機器の開発に重点が置かれてきた。これによりディスプレイ装置もプリンタも極めて高性能な機器が出現したが、一方量産品への指向が弱く日本語情報処理の大衆化が遅れており、いまだに日本語情報処理は高級なシステムと考えられている。ＬＳＩの技術が電卓という大衆化路線によりその基盤を確立したことを、他山の石とすべきであろう。

さて最近になってこの大衆化のきざしが見え始めてきた。主なる方向は二つあり、一つはオフィスコンピュータへの組込みであり、日本語ワードプロセッサもその一種と考えられる。もう一つは日本語情報のオンライン提供サービスにおける端末機である。現在漢字出力機器におけるコストのネックは、その文字発生器におけるメモリであるが、その本命と見られているマスクＲＯＭおよび磁気バブルメモリの将来のコスト低下を見込んで、英数カナモードの現在の機器と大差ない価格を目標とするものである。

オフィスコンピュータはデータ処理におけるパーソナルコンピュータとして、また分散処理における簡易なノードコンピュータとして脚光を浴び、ここ二，三年急速な伸びを示しているものであるが、これに日本語情報処理の能力を組込むことはデータ処理の高度化という面のみでなく、真のパーソナルユースという面での意義も大きく、またワードプロセッサへの途を開くものである。オフィスコンピュータの場合は、日本語の出力機器の組込みとともに入力装置の方式も一つの大きな課題である。

情報処理システム全般のすう勢であるリアルタイム処理の普及は、当然日本語情報処理をもその中に包含する。検索や照会などに対する回答を、日本語情報処理によって表示することは将来の普及への絶対的な条件である。現在、特定多数を対象とする情報提供サービスを日本語情報により行なっている機関は、日本科学技術情報センター、日本特許情報センター、日本経済新聞社など数少ないが、既にいくつかの官公庁では、各種文書や申請書類などを日本語により入力し膨大なデータベースを作成している。これら官公庁での広域オンライン化のニーズは極めて高く、データ通信技術および主として端末機の技術の進展に伴い、ここ数年のうちに実用化が広まるものとみられている。

さらにオンラインによる情報サービスに関しては、以上のようなビジネスユースに止まらず、全くの不特定多数を対象とした個人向けのサービスの開発が進められている。これは始め英国の郵電公社により始められ、既にＰＲＥＳＴＥＬの名の下に実験サービスが行なわれている。わが国での同等のサービスは、電電公社のＶＲＳおよび郵政省のキャプテンズであるが、近く実験サービス開始の予定である。これらは既設の電話とテレビ受像機を用いるが、もちろんプリンタの接続も可能で、ＰＲＥＳＴＥＬではいわゆるオンライン情報検索システムとほぼ同等の機能を有している。しかしわが国では日本語の記述文字の問題があり同等の域には達していないが、明らかに日本語情報処理の一種であり、これに使用する日本語端末機は大衆化路線の最右翼に位するものであり、すでに試作も開始されている。これは超簡易型端末として将来数十万台の需要が見込まれるものである。

2.2 最近の話題

オンライン・システム関係の話題は2.3に譲り、それ以外の最近のトピックについて触れよう。

2.2.1 カナ入力方式（タッチ打法）

日本語情報処理においてはその入力がネックであることは現在でも変らない。既に開発の当初より入力問題に多くの関心が集まり、これまで数多くの方式が提案されてきた。その中で従来からの多段シフト式の漢テレ型フルキー式けん盤入力機と、同じくフルキー式の漢字タブレット入力機が主流を占めてきた。しかし最近になって日本語情報処理の普及と処理量の増大に伴って、英数カナけん盤を用いたタッチ打法による入力方式の有利性が認められて、実用化の傾向が目立ってきた。

情報処理業務の一般的傾向としてバッチ処理による大量情報の処理は、専門の情報処理センターへ委託する方向にあるが、漢字の入力作業もこの例外ではない。かかる状況となると、漢字入力を主たる業務とするセンターが多数現われ当然価格競争となる。タッチ打法は現実にはこのような現状におけるコスト追求の結果として脚光を浴びるに至ったものである。またこの背景としてキーツーディスクなど、入力システムの革新も見逃すことができない。

タッチ打法の利点として次のような点があげられる。

① 入力機器が英数カナモードと同一であり機器の利用効率がよい。

② ベリファイ、マシンチェック、テーブル付加など有利なサポートが利用できる。

③ ミスタッチの訂正が容易である。

④ 入力速度がフルキー方式より速い。

⑤ 機器のコストが安い。

⑥ 打けんの習熟期間もフルキー方式と大差がない。

このような利点があるため最近急速に普及が進んでいるが、中でもカンテック、九段コンピュータ・サービス（ＫＩＳ）、大阪計算センターが先駆的役割りを果している。またキーツーディスクなどの販売会社が、漢字入力システムをソフトウェアとして販売する動きも出てきている。

タッチ打法により日本語を入力する方法は特に目新しいものではない。方式は大別して二つあり一つはラインプットとして著名な、漢字に連想的なカナ２文字の読み（コード）を与える方式であり、他は新興製作所がＳＣ４０００として発売している漢字表と座標コードを用いる方式で、共に２ストロークで入力するように工夫されている。しかし実際のコード表は各センターで独自に開発されており、企業秘密ないしは販売の対象となっている。したがってコードの人間工学的面からの最適化については、今後とも議論に花が咲くものと思われる。

さてこのようなすう勢を受けて昭和５３年の情報処理学会全国大会で、「漢字入力法の人間工学的検討」と題するパネル討論が行なわれた。その内容の詳細は省略するとして、東京大学理学部情報科学科の山田尚勇氏が、タッチ打法の優位性について詳細な報告を行なっているので、その主要点を紹介しておこう。

まずタッチ打法とサイト打法（けん盤を見ながら打けんする方法）を表２－１のように比較する。

表２－１　入力打法の比較

	サイト打法	タッチ打法
入力機器	和文タイプ、漢テレ	英数カナタイプ
視点	原稿とけん盤	原稿のみ
作業	意識による決定が入る	反射的運動
リズム感	な し	大いにあり
体力消耗	多 い	少ない
作業速度	遅 い	速 い
心理的負担	大きい	小さい
職業病	出やすい	出にくい
習熟期間	比較的短かい	初期にみっちり訓練がいる
個人差	特に習うべきコツはない	コツを身につけると差が出る
職業意識	低 い	高 い

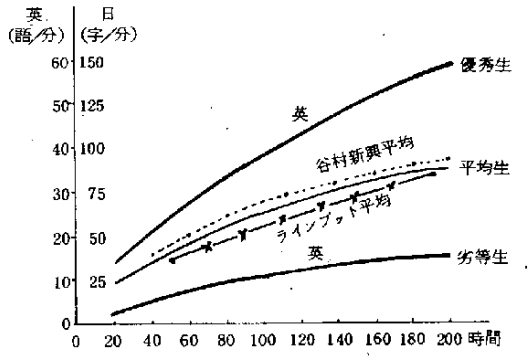


図２－１　タッチ打法の習熟曲線

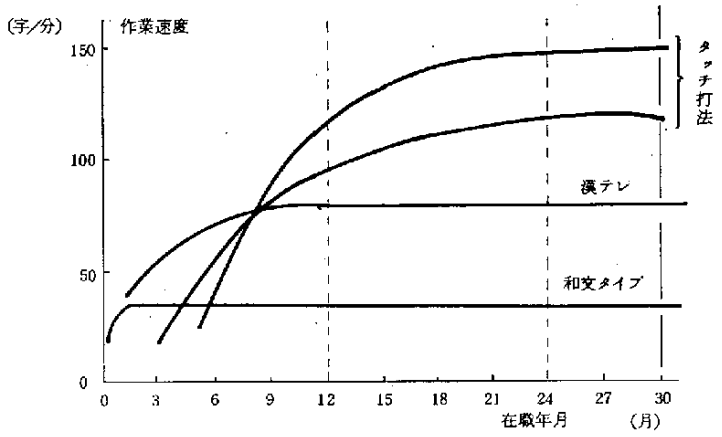


図２－２　各種入力方法の習熟曲線

この比較の中で特に重要なのは、体力消耗・心理的負担・職業病など身体に関する面で、サイト打法がすべて劣っていることである。氏は具体例を示してはいないが、これら労働衛生面での問題は重大であるので、今後議論を呼ぶものと思われる。

次に打けん速度であるが、米国における高校生の速度とラインプットおよび新興製作所の方式との比較として、図２－１を示している。これでみるといずれも平均値は大体同程度であり、漢字コードがあまり負担となっていないことがわかる。さらに３０か月にわたる和文タイプ、漢テレ、タッチ打法のそれぞれの打けん速度の比較として、図２－２を示している。タッチ打法は習熟による伸びが大きいだけでなく２年位にわたって習熟が続き、最終的には漢テレ方式の１.５～２倍の速度に達する。一般に打けん速度は正確な値を発表したがらない傾向があり比較はむずかしいが、この図からすると優劣の差は歴然としている。

この他氏の論点を以下に列挙しておく。

①　入力装置はプロの職業人が使用するものとして設計すべきである。

②　訓練期間は長くても当然である。打けんはプロの職業である。

③ 打けんは反射運動となることによって始めて精神的爽快感を味わえる。

④ 同音異字の処理を入力後に行なう方式はタッチ打法とはいえない。

⑤ 日本文中のカナ・かなも漢字同様２ストローク打けんがよい。

⑥ タッチ打法の漢字コードとキー配列の人間工学的最適化をはかり標準化すべきである。

⑦ 連想コードはタイピストの能力の発展に非常に悪い結果をもたらす。

⑧ タイピストは訓練方法によって能力レベルが左右される。

⑨ 漢字コードで打けんするのはよほど能力のある人でない限りできないと思うのは誤った先入観である。

タッチ打法についてはすでに英文タイプでその優秀性は認められているが、日本語への適用については、パンチ・センターや計算センターなどでの実績データおよび山田氏らの主張をふまえて、ここ当分の間議論が交されるであろう。その際特に労働衛生面からのデータの収集と分析にも、力を入れてもらいたいものである。

2.2.2 日本語ワードプロセッサ

最近米国およびヨーロッパにおいては、合理化の点で製造部門より大きく立遅れている事務部門の生産性向上をはかるべく、そのエースとして登場してきたのがワードプロセッサ（ＷＰ）である。ＷＰの潜在市場は極めて大きく、ここ数年の間に需要は顕在化して爆発的に拡大するとも予測されている（図２－３）。

ワードプロセッシングとはデータプロセッシング（ＤＰ）に対比される用語で文を処理するシステムの総称であるが、現在ＷＰといわれるものは、小規模な編集機能を持った編集タイプライタとでもいうべきものである。米国においてはＩＢＭ社製が寡占状態にあり、ＤＥＣ、ＸＥＲＯＸ、ＷＡＮＧＬＡＢ、Ａ．Ｂ．ＤＩＣＫ社などが参入している。

ＷＰの構成ブロックを図2－4に示す。プロセッサを中心としてキーボード、プリンタ、ファイル記憶からなる。ＩＢＭのＷＰはファイル記憶として安価な磁気カードを用いているのが特徴である。これら構成ブロックに対応してＷＰの基本機能は次の四項からなる。

入力機能	キーボード
編集校正機能	プロセッサ
文書保存機能	ファイル記憶
印字機能	プリンタ

英文のＷＰについて各機能をみてみると、キーボードは当然のことながら通常の英文タイプのキーボードと同一である。編集校正機能は各社が最も工夫をこらしているところで、プロセッサと内部記憶を活用してそれぞれ特徴を出している。基本的には語や行を任意に呼出して修正する校正機能、行編集・行末そろえ・自動マージン・タブセットなどの編集機能、ファイルからの自動印字および特定の語の挿入機能などからなる。

ファイル記憶としては磁気記憶を用い、着脱可能として記憶容量の増大をはかっている。ＩＢＭ

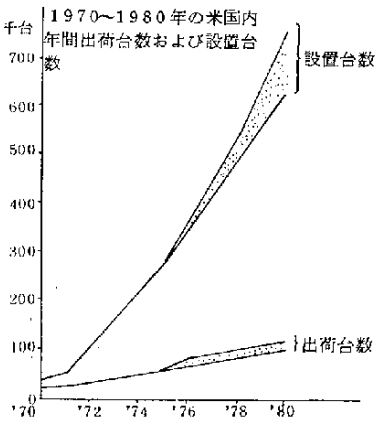


図２－３　ＷＰの需要と予測

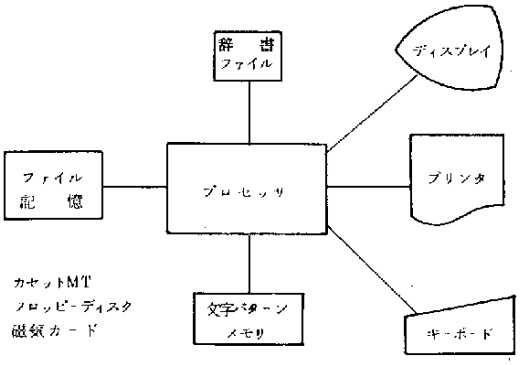


図２－４　日本語ＷＰの構成ブロック図

社の磁気カード（容量5,０００字）がその代表である。プリンタとしては活字インパクト方式が主流であるが、自動印字に対応して高速が要求され、また多くのタイプフェイスを用意している。ＷＰにおいては特に印字の美麗さが要求され中には植字装置を用いているものもある。

さてわが国においては、和文タイプライタは職業的専門家の操作する機器としてオフィス外にあり、事務部門の合理化の対象外と考えられていた。またメーカ側からは、日本語情報処理の技術的困難さと複雑さから非常に高価なものとなり、実用的製品の開発が遅れていた。日本語のＷＰを考える場合以下の項目は必須条件と考えられる。

(1)　日本語なるが故の操作の困難さをなくし利用者に繁雑感を与えないこと。

(2)　機能、性能にこだわって高価とならないこと。

(3)　印字された文書がそのまま社会に通用すること。

これらの需要を満したＷＰを現在の技術レベルで実現することはなかなか困難であるが、最近に到

って二，三のメーカからＷＰを目的とした機器が発表されるに至った。利用者側からのその評価は未知数であるが、英文のＷＰとそれほど大差ない価格で発表されたことは、今後の普及に大いに期待が持てると考えてよいであろう。なおＷＰとオフィス・コンピュータの日本語情報処理は、技術的にはほぼ同等と考えてよいであろう。

具体例として東芝の日本語ワードプロセッサＪＷ－１０の機能を次に示そう。

(1) 入力機能

入力は英数カナキーボードを用いたカナ漢字変換方式を用いている。日本文をその読みに従ってカナ文字により入力する方式で、操作の容易さとコストの低減をはかっている。漢字への変換は装置内に用意されている変換辞書を用いるが、変換は語単位で行なわれ標準で普通単語４万語、固有名８千語が用意されており、さらにユーザエリアを用意し最大８万語の登録を可能としている。

入力モードとしては文節ごとにカナ文字文で入力し、カナ漢字変換を行なう文節指定モードと、漢字を用いたい部分を指定して入力する漢字指定モードとがある。同音異義語の選択モードとしては、文の入力時に同音異義語が発生するつどその選択を行なう逐次選択モードと、入力完了後まとめて同音異義語を選択する一括選択モードがある。同音異義語はディスプレイ上にその使用ひん度順に表示され、その中から選択して指示し入力する。もし表示された同音語の最初の語が期待したものであれば、次の文の入力を続行すると自動的に選択入力される。

文字種はＪＩＳ　Ｃ－６２２６「情報交換用漢字符号系」の第１および第２水準漢字6,802字で、またある語が変換辞書に登録されてなく漢字に変換されない場合には、漢字１字ずつ音・訓読みにより入力することができる。

(2) 編集校正機能

編集校正機能は英文ＷＰと本質的に異なるところはない。

①訂正　②挿入　③削除　④移動　⑤全文対象

訂正・削除の直後にのみ有効で、その対象となった文字列と同一内容の文字列に対して、全文にわたり同一処理を行なう。

⑥タブ　⑦インデント（行頭位置指定）　⑧アンダーライン　⑨センタリング　⑩枠あけ

⑪切りばり

複数の文書の指定する部分をつなぎ合わせて一つの文書を作成する。

⑫さし込み

文書の指定個所に種々の文字列を自動的に挿入していく（例：入金通知書）

⑬作表　⑭禁則処理

(3) 文書保存機能

保存：本体内　　約２００ページ

フロッピーディスク　　約６０ページ／枚

呼出し：ページ単位、行単位

(4) 印字機能

プリンタ、ディスプレイともオフィス・コンピュータ「ＴＯＳＢＡＣ漢字システム１５」と共用に、新たに開発した低コスト高性能の装置を用いている。これら機器の開発によって日本語WPの製品化が可能となったといっても過言ではない。プリンタは２４ピンのドットマトリックスによるインパクト式シリアルプリンタで、横書き／縦書き、フォーマット変更、自動ページングなどの機能を有する。

〔ディスプレイ〕

画面サイズ	１２インチ
表示方式	２４×２４　ドットマトリックス
表示文字数	３２字×１４行：４４８字

〔プリンタ〕

印字方式	ワイヤドット・インパクト方式
文字構成	２４×２４　ドットマトリックス
文字サイズ	3.9×3.5㎜（約１１ポ）
印字速度	35 字／秒
印字字数	最大９０字／行
字　体	明朝体
印字方向	横書き／縦書き

本機の価格は６３０万円と発表されているが、英文の標準的WPが４００万円前後であることから、わが国においてもいよいよWPの時代が開幕するものと思われる。オフィス・コンピュータと併せて約１０年の歳月を経て、やっと普及時代が到来したといえよう。日本語WPの真の評価は今後に残されているが、新たな一歩を踏み出したことは確かである。

2.2.3 画像通信システム

2.1で簡単に触れた英国のＰＲＥＳＴＥＬ、郵政省のキャプテンズ、電電公社のＶＲＳなどは文字のみでなく画像も通信の主たる対象としているところから、一般に画像通信システムと総称されている。その代表としてキャプテンズのシステムについて述べよう。

これは基本的にはＰＲＥＳＴＥＬと同じ考え方に基づくもので、システム構成を図２－５に示す。利用者はまず電話でセンターを呼出し接続されたらキーパッドで検索を行なう。センターでは文字図形情報ファイルを探索し要求された情報を取出し画像処理部で画像パターンに変換して利用者に伝送する。信号はアダプタのリフレッシュメモリに蓄積され、テレビ受像機に画面として表示される。画面の表示容量は日本語で横１５字縦８行の１２０字で、伝送速度は4,８００ｂｐｓ、一画面分を送るのに１５秒を要する。

このように文字を送るのに、一般の日本語情報処理システムのように文字コードで伝送し端末側でパターンに変換する方式をとらないのは端末が高価となるためである。このため伝送に１５秒も

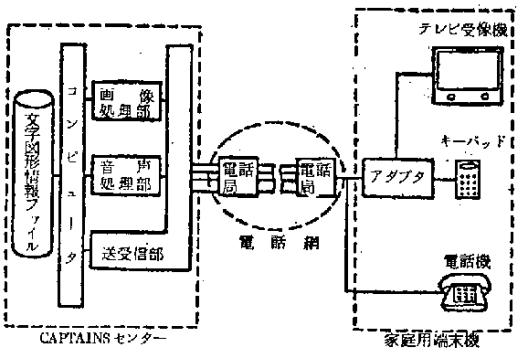


図２－５　キャプテンズ・システム

要し実用上は問題である。コード伝送にすれば１２０字を0.５秒で送れる。ＰＲＥＳＴＥＬでは英数字のみであるからコード伝送を行なっている。

郵政省では電電公社と協力して昭和５４年８月から1年ないし1年半にわたって東京都内の１千世帯を対象として実験を行なうことになっている。またこの実験のために報道出版関係の企業を中心として約７０社が協力を表明し、ファイルに１０万ページ分の情報を蓄積することを目標としている。このようにキャプテンズは一般家庭を対象とした簡易型のオンライン情報検索システムであるが、将来的にみると次のように日本語情報処理の発展と密接に関連している。

キャプテンズは上のようにテレビ受像機によるディスプレイのみであるが、実用上はどうしてもコピーのとれるプリンタが必要である。メモの代りという消極的な用途ではなく座席や宿泊などの予約券、入金の領収書など将来の発展を予想すると、家庭に設置できる超簡易型漢字プリンタの開発は必須の条件である。このため郵政省ではかかる端末機の開発を進めており、まず多摩ニュータウンで実験中の生活情報システム（ＣＣＩＳ）の端末機として近く実用化試験を行なう予定である。このプリンタの開発に当っては

１６×１６ドット

１５×１８ドット

でできる限り安価なものという条件を出し、すでに試作機を完成させた。価格の目標は当面1台１０万円、将来２～３万円としており、キャプテンズにもこれを採用する方針である。4,８００bpsで伝送すればドットパターンで送っても２０字／秒の印字速度が得られ、また１行の印字字数を少なくすれば装置は簡易化され実現の可能性が高い。

2.2.4　その他の話題

昭和５３年７月に情報処理学会の主催により「日本語情報処理」に関するシンポジウムが開催された。テーマとして入出力、応用、処理、日本語について約３０件の発表が行なわれた。このシンポジウムを通して今後推進すべき課題として次のテーマが取上げられている。

(1)　タッチ打法による入力方式

(2) 高品質の漢字フォントの開発

(3) 高品質低価格のプリンタの開発

(4) 用語辞典、英和辞典のファイルの整備

(5) 日本語による会話システムの研究

(6) 日本語の構文分析、意味論の研究

(7) 日本語によるデータベースの開発

(8) 機械翻訳

(9) 手書き日本文字の自動認識

(1)については2.2.1で、(2)および(3)については2.2.3でそれぞれ触れたところであり、さらに(3)についてはオンライン用の端末機として後に詳しく言及する。(4)~(8)は今後日本語の総合的情報処理を確立する上での基本的なソフトウェアであり、(9)は究極的なハードウェアである。ここでは実用面に焦点を合せて標準化の面から現状をみてみよう。

技術が発展し需要が喚起されて普及段階に入ると、標準化つまり規格化が非常に重要な要素となってくる。特に集積回路のように大量生産がコスト低減のかなめとなるような部品になると、規格化がキャスティングボードを握ることになる。漢字プリンタのコストのネックは、文字発生器の文字パターンの標準化が遅れているためである。

日本語情報処理における規格化は、昭和53年1月に制定されたC－6226「情報交換用漢字符号系」のみである。この規格では漢字の字種とその符号系が制定されたが、この1年間にその普及は目ざましいものがある。しかし一方その制定に伴って不満を訴える論評も二、三見受けられるが、プラスの効果の方がはるかに大きい。この他制御符号系の規格案がまとまり制定を待っているが、現在のところ規格化の動きはこれのみである。しかし規格化を急がねばならない項目は山積している。

まず入力についてはフルキー方式におけるけん盤配列がある。タッチ打法の項でも述べたように、大量バッチ入力においてはフルキー方式万能とはいえないが、用途によっては今後とも必要な方式である。不幸なことに和文タイプの活字配列には規格がなく、日本タイプライタ、写研を筆頭にして各メーカで独自な配列をとっている。しかし入力機となると活字の場合と異なり自由度が大きいため、このままではますます発散する恐れがあり、またメーカとしてもそのつど配列に頭を痛めねばならない。C－6226により漢字の字種とその順序が固定されたのであるから、まず第一に着手すべき規格化であろう。これと同時に外字入力の方式も定めそのためのキーの規格も必要であろうし、さらにパターン入力の方式も定まればこれに越したことはない。

ＪＩＳカナキーボードの配列は評判がかんばしくなく、また英文のキーボードについても非能率であるという論が強いことから、漢字のけん盤の場合には現在の技術水準にとらわれることなく、真に人間工学的な立場から充分に検討を重ねるべきである。さらに制御符号系の制定とともに制御符号キーが定まれば、入力ソフトウェアにとって極めて意義のあるものとなる。

この問題と関連の深いのが字体とそのドットパターンの規格である。C－6226では字体を規

定していないから、これは本規格の領域であり慎重を期さねばならない。パターンを定めるには、明朝体とかゴシック体とかの字体と文字のサイズを定め、それに従ってドットパターンの縦横の密度を定め、パターンを設定しなければならない。パターンを設定するには文字のグラフィカルなパターンの美麗さと、既存のデザイン権が存在し、なかなか大変な作業である。恐らく数種類のフォントが設定されると思われるが、この困難さを克服することが漢字プリンタのコスト低減と、利用者の便宜にとって不可欠な条件である。

現在漢字プリンタと英数カナプリンタとの価格差の大部分はその文字パターン発生器にある。パターン発生器はコードとパターンの変換テーブルのメモリであるが、両者を比較すると以下のようになる。

	英数カナ	漢　字
文字種	128	4,000
文字コード	8ビット／字	16ビット／字
文字パターン	5×7	20×20
メモリ容量	4.5Kビット	1.6Mビット

このように漢字の場合には高速大容量のメモリが必要となり、一般に磁気ドラム、ディスク、RAM、ROMが用いられているが、将来の本命はROMと考えられ、そのためには規格化されたマスクが絶対に必要であるといわれている。以上さし当って急がれている規格について考察した。

2.3 分散処理用端末

LSI製造技術の急速な発展に伴ってプロセッサのコストが年々低下し、従来では考えられなかったような分散処理が可能となってきた。この傾向と日本語情報処理との接点として当然のこととして、漢字ディスプレイや漢字プリンタを中心とする、日本語端末機がクローズアップされてきた。端末には大別して専用端末機と汎用端末機があり、専用機は銀行窓口用、予約用、POS用などそれぞれの業務に最も適合するように作られ、そのコストも本来業務のコストの中に吸収される。また特定業務用としてある使用範囲内での標準化ですみ、一般に使用ひん度も高い。これに反し汎用端末となると使用条件の特定がむずかしく、最も重要なファクタとして、端末の普及がサービスの普及を喚起するという関係がある。したがってシステムに先立って端末が開発されねばならないという厳しい条件に置かれる。

現在英数モードの汎用端末機はTSSサービスを中心とするリモート処理、データベースに照会する情報提供サービスに主として用いられている。近い将来社内のオンラインシステムにも逐次利用が開けると考えられるが、一般に専用機に比して利用ひん度が低く、会話型システムの場合にはそれほどの高速性も要求されず、また端末機のコストが直接情報の処理や入手のコストに付加して取扱われるため、安価なことが大きな条件となっている。

日本語端末機はもちろん当面は、日本語情報の取扱いが主たる業務となるシステムにおいて需要が強く、そのようなシステムとしては情報提供サービスが最右翼にあり、しかも需要喚起の面から

安価な端末が望まれている。したがって本節ではそのような端末の可能性を技術的に分析するが、それが可能な場合それより高級な端末は自ずと解決されるはずである。

2.3.1 米国における情報サービスの発展

米国における言語情報のオンラインサービスは、まずＭＩＴにおいてＴＩＰおよびＩＮＴＲＥＸとして実験的に開始され、次いでＮＡＳＡのアポロ計画の一環としてロッキード社によってＲＥ－ＣＯＮが開発された。さらにＳＤＣ社のＯＲＢＩＴ（ＮＬＭで採用)、ＩＢＭ社のＳＴＡＩＲＳが開発され、これらのシステムの開発とともに、続々とオンラインによる情報サービスが開始された。主なものは以下のとおりである。

ＬＩＳ社（昭和４７年開始）
　ＤＩＡＬＯＧ
　データベース　７５種　2,０００万記事

ＳＤＣ社（昭和４７年開始）
　ＯＲＢＩＴ
　データベース　４４種　1,０００万記事

ＢＲＳ社（昭和５２年開始）
　ＳＴＡＩＲＳ
　データベース　１１種　1,０００万記事

ＮＬＭ　（昭和４７年開始）
　ＥＬＨＩＬＬ（ＯＲＢＩＴ）
　データベース　９種　２００万記事

ＮＹタイムズ社（昭和４８年開始）
　データベース　１種　１５０万記事

かくて米国においては情報システムをいかに作るかから、いかにサービスを利用するかということに関心が移っている。普及の背景としてシステムの開発以外にＶＡＮ（付加価値ネットワーク）の発達とその利用、および電話網を利用できる安価な簡易端末の利用を見逃すことはできない。ちなみにこれらサービスは一つの例外もなく英語を用いている。

米国においてはシステムの安定・大規模化とともにサービス間の競争も激烈となり、料金の値下げ、データベースの拡充、機能の向上競争に入っている。ちなみにＬＩＳ社のサービスを受けた場合、１回１０分として通信料金を含めて５～１６ドルとなる。料金の違いはデータベースの使用料が異なるためである。一方端末機のリース料金は８５～１５０ドル／月であり、平均１回の料金を８ドルとすると１０～２０回分程度に相当する。ＬＩＳ社のサービス状況を以下に示す。

利用回数	１５０万回／年	
利用端末数	4,０００～5,０００台（推定）	
登録利用者数※	１６,０００	（※海外二十数か国を含む）

米国の情報サービスは国際通信回線を通して世界中にサービス網を広げており、近くわが国にも上陸が予想され賛否両論がかまびすしい。これら情報サービスに収録されているデータベースは、大部分が米国の情報を米国においてデータベース化したもので、その意味においては海外の重要な情報が広範囲に早くキャッチできる意義は大きい。しかし一方わが国にはわが国独自の情報が数多くあり、それらを日本語のデータベースとしサービスすることも重要なことである。しかも言語情報となるとカナ文字文では扱いにくく、漢字かな混り文によるサービスが必須となる。現在わが国で大規模なデータベースを日本語で構築している機関を以下に示す。

日本科学技術情報センター（ＪＩＣＳＴ）

雑誌記事　　オンライン／バッチ

日本特許情報センター（ＪＡＰＡＴＩＣ）

特許公報　　オンライン／バッチ

日本経済新聞社

新聞記事　　バッチ／オンライン（計画中）

わが国におけるオンライン情報サービスはやっと夜明けを迎えたばかりで、現在は先行投資としてデータベースおよびシステムを構築し、細々と需要の喚起を行なっているところである。問題点として指摘されているのは次の三点である。

(1) データベースの構築が遅れている

(2) 通信回線利用の制約が大きい

(3) 漢字システムのコストが高い

このうち特に特定多数に普及をはからなければならない、公衆電話網を利用する簡易日本語端末についてその可能性を分析するものである。

2.3.2 通信回線開放への動き

この問題も日本語情報処理を普及させるための、一つのかなめであるので簡単に触れておこう。

米国において情報サービスが爆発的に普及した最大の要因は、ＶＡＮにあるといわれている。ＬＩＳ社の場合ＶＡＮの利用料金はTymnetで8ドル／時、Telenet で5ドル／時で全米一律料金である。1回のサービスを１０分とすると円に換算（２００円／ドル）して、それぞれ２６７円と１６７円となる。別の計算をすると１０円でそれぞれ２2.5秒、３６秒使用できることになり、これは東京－鎌倉間および東京－横浜間の電話料金と同じである。しかも全米どこからでもこの料金であるからいかに安いかがわかろう。

米国のＶＡＮはＶＡＮ業者が通信業者から回線を借受け、これに各種の通信機能を付加して提供しているもので、わが国では公衆電話通信法によってこのようなサービスを業者が行なうことはできない。同法５５条の１１によって、２人以上の者が同一の特定通信回線を共同で利用する場合に厳しい制限を設け、また５５条の１３によって、業として特定通信回線を用いて他人の通信の媒介をするなどの他人使用を禁じている。ただし１つのコンピュータの本体 と１つの入出力装置との

間に終始する共同利用は認められているので、情報サービスは辛うじて実行されている状況である。

したがって、情報サービス業がそれぞれネットワークを個別に構築してサービスを行なうか、または公衆通信回線（電話網）を用いてサービスをすればよいわけであるが、わが国の通信回線利用料金は下のように距離によって非常に割高であり、コスト負担に耐えられない。

	電話通話料	特定回線料（D－1）
東京都区内	１８０秒／１０円	1.2万円／月（１０㎞内）
東京ー福岡	2.5秒／１０円	８６万円／月
倍　率	７２倍	７２倍

一方電電公社がＶＡＮ相当のサービスをすればよいわけであるが、新データ通信網（ＤＤＸ）が予定より大幅に遅れており、また使用料も発表されていない。

このような状況であるので昭和５３年に入ってにわかに

(1) 共同利用、他人使用など回線利用の自由化

(2) 回線料金の引き下げ

(3) 公社の情報処理サービスに対する規制

などを求める声が高くなってきた。まず８月に「データ通信会議」が提言し、１１月に日本情報センター協会、１２月に電気通信ユーザ協議会、ＥＤＰユーザ団体連合会等がそれぞれ要望書を提出した。これを受けて郵政省は「データ通信振興法案」を国会へ提出すべく準備中であり、これの骨子は郵政大臣が指定するデータ通信システムについては、回線の共同利用、他人使用、相互接続など一切の規制を外すというもので、全面的な制限緩和ではないが、情報サービスもその恩恵に浴せるものと期待されている。

このように通信回線の問題を取上げたのは、日本語によるオンラインサービスの需要を喚起するためには、簡易型日本語端末の普及が第一に必要な条件であるが、それ以前の問題としてネットワークの効率化と回線料金の引下げが、重要な因子であると考えられるからである。一般的にいえば、端末のリース料より回線料金の方が安いという状態を実現すべきではなかろうか。

2.3.3 技術的課題

(1) 印字機構

漢字プリンタの印字機構は低速用と中高速用に分かれ、低速用としては現在以下の方式が実用化されている。

インパクト方式

活字インパクト式

ワイヤドット式

ノンインパクト方式

サーマル（感熱）式

インクジェット式

この他にも電解記録式、放電破壊式などがあるが、簡易日本語端末用としては上記4方式とみてよい。

活字インパクト式はかつて漢字テレプリンタとして実用化されていたが、印字速度が5字／秒と遅く姿を消していた。最近になって軽金属ドラムの表面に感光樹脂版の活字マトリックスを装着した新製品が発表になった。しかし2,600字種で10字／秒と遅く高々300bps が限度と思われるが、一方価格は現在のところ文字発生器用のメモリが不要のため、他の方式に比し格段に安価である。

感熱式はいわゆる「あぶり出し」と同じで、熱により発色する有機化合物を塗布または含浸した感熱紙に、集積回路の発熱素子からなるプリントヘッドを接触させ、文字パターンに従ってヘッドを発熱させ印字させるものである。しかしヘッドの熱応答速度が遅いため600bps が当面限度であると思われる。感熱式の特徴は騒音が低いこと、構造が簡単なため信頼性が高く小型軽量であること、保守が容易なことなどであり、要約すれば非常に簡便なプリンタを構成できることである。したがって米国でもTI社のSilent 700シリーズを始めとして、多くの携帯用端末に採用されている。感熱式の不利な点はプリント用紙として特殊な感熱紙を必要とすることである。

ワイヤドット式はワイヤの先端を用いインクリボンを通して、マトリックス状に文字を打印するもので、英数カナプリンタ としては多くの製品が実用化されている。ワイヤを細くし数多く用いれば印字品質は向上するが、物理的な制約もありドット数は印字の大きさとワイヤの太さのかね合いにより定まる。一例をあげれば0.2㎜φのワイヤを用い、24×24ドットで16級（11ポ）の明朝体漢字を、35字／秒で印字するプリンタが製品化されている。複写能力はオリジナルを含めて5枚である。印字の方法としてはシリアルプリンタの場合、縦一列（または千鳥に二列）の横走査方式をとるが、横一列の縦走査のラインプリント方式をとれば印字速度は格段に早くなる。以上の技術レベルからワイヤドット方式は、1,200bps の印字速度も可能であると思われる。本方式は技術の安定性、複写能力、普通紙の使用などの点から、低速用プリンタとしては最も普及している。

インクジェット式は低速用としては電界制御方式と荷電量制御方式とがあるが、日本語端末としては印字速度の関係から後者が有望である。この荷電量制御方式を用いれば1,200bps 以上の速度も可能である。試作機ではあるが32×32ドットで9ポの明朝体漢字を、60字／秒で印字するプリンタが発表になっている。本方式は高速、高印字品質、低騒音と極めて優れた性能を持っており、特に漢字シリアルプリンタとして期待されるものであるが、インクの品質、信頼性などの点で将来に期待が持たれている。

以上各方式の得失を概観したが、それぞれの性能を一覧表として表2－2に示す。

表２－２　プリンタ性能比較

	活字インパクト	ワイヤドット	感　　熱	インクジェット
文　字　種	3,０００	ＣＧのメモリによる	〃	〃
文字品質	優	良	可	優
印字速度	１５字／秒	６０字／秒	３０字／秒	６０字／秒以上
用　　紙	普通紙	〃	感熱紙	普通紙（制限あり）
複写能力	３枚	５枚	２枚	１枚
騒　　音	６０～６５ホン	〃	５０～５５ホン	〃
ヘッド寿命	③	②	②	①
消　耗　品	インクリボン	〃	感熱紙	インク
予想価格	④	②	③	①
総合評価	③	①	①	②

この表から総合的に判断すると、当面簡易型に徹すれば性能面には目をつぶって感熱式が妥当であり、実用性に重点を置けばワイヤドット式ということになろう。インクジェット式は将来に期待し、活字インパクト式は字種の制限と印字速度の制約がきびしく、汎用性の面から問題が多いといえよう。

(2)　文字発生器（ＣＧ）用メモリ

ドットパターン方式の日本語端末機は将来の本命と目されているが、簡易型として端末機の性能を最少限に押えたとしても、ＣＧ用メモリを極端に減らすことはできない。かえって簡易型にすればする程コストに占めるＣＧ用メモリの割合は増加する。かくてどのメーカも異口同音にＣＧメモリのコストが下らない限り、日本語端末機の価格は下らないという発言となる。

現在低速の漢字プリンタのＣＧメモリとしては、以下のようなものが使われている。

磁気ディスク（含フロッピー）

磁気ドラム

ワイヤメモリ

磁性線ソレノイド

ＰＲＯＭ

磁気ディスクやドラムは簡易型という点からは不適当である。将来的にみてどうしてもＩＣメモリを使用したいというのが一致した意見である。

一方英数カナ用端末は６４字～１２８字分でよく、またドットマトリックスも５×７または７×９でよいためメモリ容量が少なくてすみ、すでにＣＧ用ＲＯＭとしてＭＯＳ　ＲＯＭが量産化されている。英数カナ用と漢字用とのメモリ容量を単純に比較すると、前にも述べたように

５×７×１２８　　：　４.５Ｋｂ

２０×２０×4,０００　：　1.6Mb

で約３００倍以上の容量を必要とする。あるメーカの試算によるとこれは約５０万円になるとのことである。マスクＲＯＭのいま一つの問題は、ＩＣの製作工程でフォトマスクを用いてパターンを書き込むため、ある量の生産がまとまらないとマスク代の負担が大きくなることである。このためには生産に見合う需要量の確保と、字種および文字パターンの規格化によるマスクの標準化が必要である。

このような状況のため現在のところ、簡易端末機にマスクＲＯＭを用いてコスト低減をはかる見通しは暗い。しかし端末機の標準化はこの問題に限るものではないから、関係機関は標準化に地道な努力を払わねばならない。

ここでメモリに関する一般的な動向をみてみよう。図２－６はビット当りのコストを、コンピュータの主メモリからＭＳＳに至る各種のメモリについて、概略を示したものである。図２－７は同じく各種メモリについてアクセスタイムの概略を示したものである。この２つの図をよく見てみると、漢字のＣＧ用メモリとしては磁気バブルが最も適しているように思われる。磁気バブルは高速のＩＣメモリと大容量の機械駆動型メモリの中間のエアポケットを埋めるものとして、後者に対しては信頼性の向上と高性能を、前者に対しては不揮発性と高密度化による低価格性を特長として開発が進められているもので、特にインテリジェント化の進む端末機器においては、小形大容量固体ファイルメモリとして大きな期待がよせられている。既に米国では製品化され始めたので近い将来に充分期待してよいであろう。

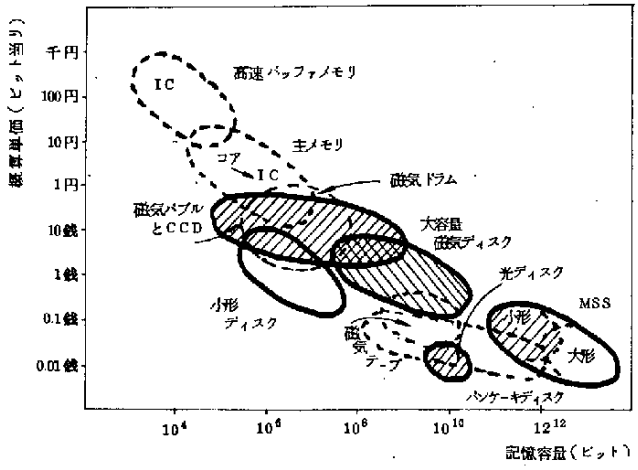


図２－６　ビット当りのメモリ・コスト

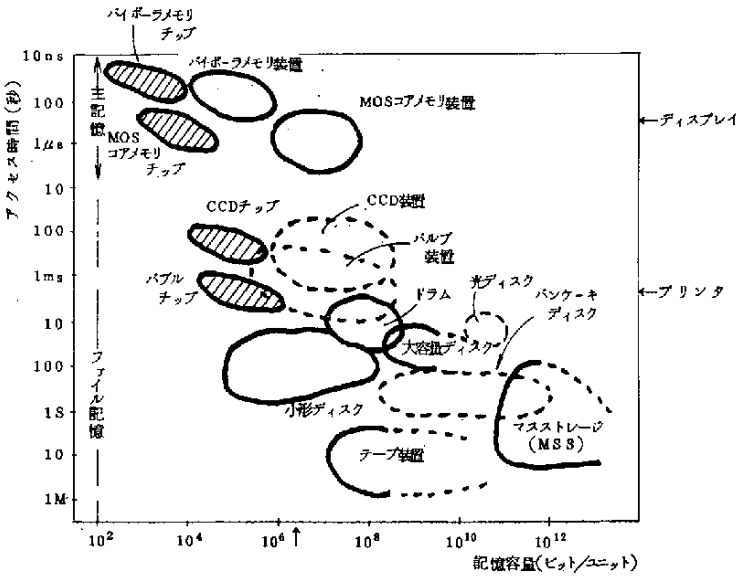


図２－７　各種メモリのアクセス・タイム

図２－８はＩＣメモリについて種々の容量のチップの価格について、その低下の推移（予測も含む）を相対的に示したものである。これによると大体５年で$\frac{1}{10}$に低下していることがわかる。したがって現在主流となっているＭＯＳのマスクＲＯＭも、大幅な価格低下が期待される。特に集積度が大きくなるとその度に、大幅な価格低下が起っていることに注目すべきである。すでに電電公社では１２８ＫｂのＲＯＭを試作しているので、ますます期待が持てるといえよう。

以上のようにＣＧ用メモリとしてＭＯＳ　ＲＯＭと磁気バブルに大きな期待が持て、近い将来簡易日本語端末機の低価格化に寄与するものと考えられる。同時に日本のオフィス・コンピュータも、漢字プリンタの装着が常識となることであろう。

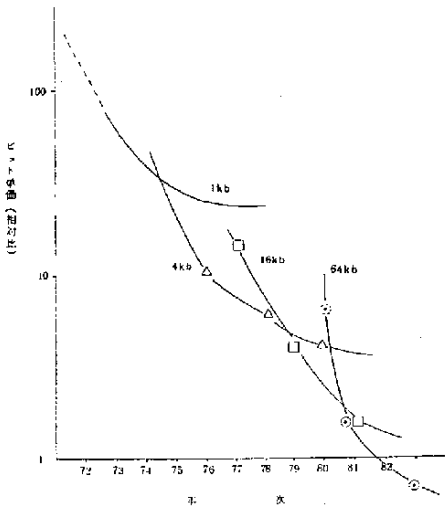


図２－８　ＩＣメモリのコスト低下

3　日本語端末用文字発生方法の検討

3.1　文字発生方式

日本語情報処理全般としても、また漢字出力装置・漢字出力方式に関する問題点の中でも、最も大きな問題としてとらえられているのが、文字発生装置の高価格化である。多量な文字種と複雑な文字パターンで構成される漢字を取扱うことから、どのような文字発生方式を採用しようとも、その構成は大規模化し、高度なハードウェア技術をもってようやく現在の日本語情報処理が可能になっている。

このような現状の中で、日本語端末としてその主要な役割りを果す文字発生装置を簡易化し、低価格化をはかるには、様々な要因がからんでくる。その中で最も大きな要因としてあげられるのが文字パターン（フォント）の記憶方法に係わることである。文字パターンの記憶には、文字発生方式によって種々の技術が採り入れられており、その中から日本語端末が採用する方法を見い出さなければならない。ここでは字母アナログ型とディジタル型とにしぼって日本語端末用の文字発生方式を検討してみる。

字母型の文字発生方式には、フライング・スポット方式、ビデコン方式、その他特殊管方式などがあるが、これらの共通的な特徴はディジタル型に比べて、高品質文字の印字ができることである。そこには、光学的な技術が採用されており、装置自体の規模が大型化して、総じて高額である。また日本語端末を想定した場合、複数端末を制御するようなマルチステーション化をはかろうとすると、字母型の文字発生装置では共用が困難とされている。汎用コンピュータのディジタル処理という観点からも、字母フォントへのアドレッシングにはディジタル処理ができても、それ以降の字母のスキャンから印字・表示機構にはコンピュータ処理はあまり生かされないという結果になる。

一方のディジタル型では、文字パターンをディジタル情報に変換して処理するために、磁気コア、磁気ディスク、その他ＩＣ、ＬＳＩメモリなどの一般の記憶素子が使用できるという特徴をもっている。しかも最近の半導体技術の進歩は急速で、今後ますます高密度・大容量メモリの開発が進んで、コンピュータ・メモリとしても普及すれば、記憶素子の低廉化が大いに期待できる。ディジタル型の文字発生方式には、ドット・マトリックス方式、ラインドット方式、ストローク方式などがある。このうち文字パターンの記憶方法が容易で、文字のアクセス表示速度がはやく、表現の自由さ、コンピュータ処理の扱いやすさ、さらには固定長記憶、表示・出力媒体の選択範囲とその多様さなどに特徴を有するドット・マトリックス方式が最も有効な方式として適用されよう。

ドット・マトリックス方式の最大の問題点はメモリを大量に使用しなければならないことである（これはディジタル型の文字発生方式すべてにいえる）。２４×２４ドットの文字パターンを例にとれば、日本語端末の取扱う文字種を2,０００字に制限したとしても、（２４×２４ビット／文字）×2,０００字で1,１５２Ｋビット、約１３２Ｋバイトという記憶容量になる。これは英数カナ文字のみを扱ってきた従来の処理系（７×９ドット／文字×１２８種＝8,０６４ビット≒１Ｋバイト）よりも１００倍以上ものメモリを必要とする。このメモリ規模だけでも文字発生装置の価格は高額

になってしまう。そのため各種の対処方法が考案されて、各メーカから漢字出力装置・文字発生装置が開発されてきた。

対処方法の代表的なものは、まず低価格の記憶素子を用いることであった。それはビット当りの単価が高いコア・メモリに代って外部記憶媒体を利用することである。しかし単に価格面を重視した装置が望まれるばかりでなく、これまでの漢字出力装置の開発は逆にどちらかというと高速化、多字種、多目的利用を目指して進んできたようである。ここに日本語端末の出現・普及が遅れている原因がある。最近ようやくこの傾向から脱皮して、オンライン利用向けに簡易で低価格の漢字処理装置が出現してきた。そこには前述したような半導体技術の進歩によるところが大きい。また周辺装置として低速であるがダイレクト・アクセスが可能で、小型であることから端末用周辺機器として発展のめざましい、フロッピー・ディスク（ディスケット）、ミニ・フロッピー・ディスクの出現が、今後とも高度な技術力をもって適用されるであろう。その他にも、高速不揮発性半導体メモリ、ＣＣＤ（電荷結合素子）、磁気バブル・メモリ等が期待されている。またホログラフィ技術を応用したホログラム方式は、ディジタルな磁気記録よりもメモリの高密度化が期待できる点で将来性があるといわれている。

もう一方の対処方法は、文字パターンのドット・マトリックスを圧縮して記憶する、いわゆるパターン圧縮・復元の技術を適用することである。パターンの圧縮が可能であればメモリの節約ができて、大量メモリの必要性もなくなり低価格化がはかれる。

以上のような現状からここでとりあげる日本語端末用文字発生方式としてドット・マトリックス方式を取りあげる。以下にその圧縮方式について述べる。

3.2 パターンの圧縮方法

パターンの圧縮方法は、現在のところ実用例はまだ数少なく、グラフィック技術や画像情報処理技術の応用としていくつかの方式が実験研究され検討されている段階である。その中から代表的な圧縮方式をみてみると次のように分けられる。

① 文字パターンを一種の図形模様とみて圧縮をする方法で、ファクシミリ伝送に広く使用されているランレングスコード法。

② 漢字は縦線よりも横線が多いこと、漢字には１ドットの点は存在しないこと、などの漢字パターン特有の性質を利用して、文字パターンの半分程度の点素（ドット）のみ記憶しておき、この圧縮パターンを補間法によって復元する方法。

③ 漢字の構造面からの特徴を利用して、「へん」や「つくり」などの漢字部首や、漢字の主要な構成要素を部分パターンとして用意しておき、これらを合成して文字パターンを表わそうという部分パターン合成方式。

④ 文字を直線の集合として近似するストローク方式。

⑤ 文字を横のラインで分割して、ラインのＹ座標とライン上での地の部分と文字部分の境界のＸ座標を記憶するラインドット方式。

⑥ 文字パターンを２×２程度の部分パターン（サブパターン）に分割して、出現頻度や隣接したサブパターン同士の結合度合等を利用して、サブパターンをハフマン（Huffman）の符号法によってコード化する方法。

ところで、ドット・マトリックス方式における文字パターンは１８×１８、２４×２４、６４×６４ドット、…… と種々のドット構成が種々の用途に応じて使われている。文字パターンのドット構成とその用途（文字品質）との関係は表３－１のようにまとめられる。

表３－１　ドット構成と文字品質の関係

ドット構成	文字の品質
１８×１８ドット	文字の骨組を表現することしかできない。画数の多い字に対しては略字体を用いなければ表現できない。たとえば、当用漢字の中でも「雲」「警」「響」「襲」の４字は表現できないほか、約4,０００字の漢字に対して約３０字の漢字が略字体となる。この文字は帳表用を主とした特殊用途の文字である。
２４×２４ドット	ほとんどの文字を略字体なしで正確に表現することができ、明朝体での表現もある程度できる。
３２×３２ドット	全ての文字を正確に表現することができ、明朝体、ゴシック体の区別が行なえる。しかし、２０画以上の文字になるとこの区別が難かしい文字もでてくる。斜線、曲線の表現が難かしい。
６４×６４ドット	書体の区別は確実に行なえるが、明朝体の線端の表現および、ひらがなの曲線の微妙な差異が表現しにくい。一般の事務用文書や帳票の版下として使用できる。
９６×９６ドット以上	線の太さ、線端のアクセント、曲線などにも十分に表現できる。商用印刷にも使用することも可能である。

文字パターンの圧縮についてパターンのドット構成を考えてみると、たとえば、ランレングスコード方式において、９０×９０ドット程度の植字用フォントでは３３～５０％（原パターンに対する圧縮後のドット数の割合）程度の圧縮率が得られるが１文字のマトリックスが１８×１８、２４×２４、３２×３２ドット程度では冗長度が少なく、圧縮にあまり大きな期待ができないという報告がある。ドット構成が大きな方がパターンに冗長度が高く、圧縮率がよいようだ。

この章において圧縮方法の調査に使われる文字パターンは２８×２４ドットのものであり、これは一般の文字パターンと同様に文字の外枠に余白を含んでいる。この余白は上下に１ドットずつ、左右も１ドットずつある。このパターンをさらに左右１ドットずつ省略して２４×２４ドットの文字パターンにしても文字の形はあまり変化しないので、以下の統計では原パターン（２８×２４）

とともに左右を２ドットずつ省略した修正パターンについても圧縮率を調査する。

ここで扱う圧縮方式は、ランレングスコード法、ラインドット法、パターンを小パターンに分割し、その小パターンをハフマンの符号化法によりコード付けて圧縮をはかる方法（以下、簡単にハフマン法と呼ぶ）、および前述の圧縮方式にはなかった方式であるが、パターンを、横に１行１行分割してできる行パターンを行の構成要素によってではなく行パターンの識別番号によって表わす方式、を取りあつかった。

ここで、これらの方式を採用したのは、いずれの方式も、圧縮・復元により、原パターンが変化しない方式であるからである。なお、文字種については漢字とその他の文字との特徴をみるために全文字種についてだけでなく、漢字のみ、またはかな・記号（ひらがな、カタカナ、英字、数字、記号）のみについても調べる。以下にこれらの方式の圧縮・復元の方法と統計結果、分析について述べる。

3.3 ランレングスコード方式

(1) 圧縮方式

この方法は１行あるいは２行を単位に端からみていって、同じパターン（１行単位の場合には白または黒、２行単位の場合には縦に２つのパターン、白白、白黒、黒白、黒黒）が連続する状態を調べる。この連続する部分をパターンの種類とその連続する長さによって表現してパターンの圧縮をはかろうとするものである。この方式では、連続する部分が多いほど圧縮率が良くなり、単独で存在する部分が多くなるほど圧縮率が悪くなる。普通、連続する回数を示す部分の長さは固定であり、たとえば、連続回数を表現する部分が３ビットであるとすると、表現できる回数は $2^3 = 8$ までである。これ以上、同じパターンが連続する場合には、２つ以上に分割される。連続回数を表現する部分の長さが３ビットであり、連続する回数が２０だとすると、これは８、８、４というように３つの部分に分割して考える必要がある。

(2) 統計方法とその結果

この圧縮方法における統計のとり方は、以下のようになる。

ｉ）１行単位

図３－１のようにパターンを行単位に分割し、各行ごとに、１ドットずつ左側からみていく。まず、連続する回数に対応したカウンタを、長さ１～２８までの２８個用意してゼロ・クリアしておく。

各行ごとに、１ドットずつみていき、変化しないうちは、連続する回数をカウントしておく。変化したならば連続した回数に対応するカウンタに加える。 以下同様に２４行全体についてカウントすれば１文字の統計が求まる。この操作を全文字種について行なう。ここでは、漢字１文字，１文字の統計はあまり問題とせず、文字種全体を対象にする。

図３－１は２８×２４ドット・パターンの例であるが、このパターンから両端２列をとった２４×２４ドットの場合についても調べる。

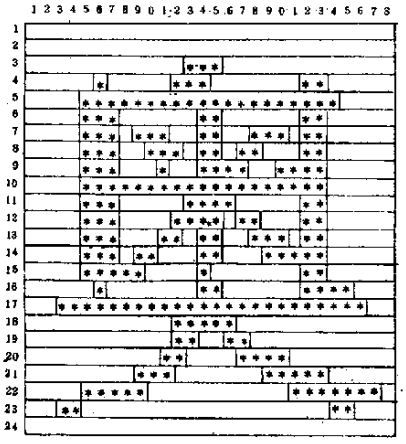


図３－１　１行単位の分割

ii）２行単位

図３－２のようにパターンを２行単位に分割して、各２行ごとに上下２ドットの連続性をみていく。変化しないうちは、連続する回数をカウントして、変化した時点で、連続した回数に対応したカウンタに１を加える。このように１２回行なえば、１文字の統計が求まる。この方法でも、１行単位の場合と同様に、２８×２４、２４×２４ドットの各パターン、および全文字種、漢字のみ、かな・記号のみについてそれぞれ統計を求める。また、この統計においては上下２ドットの種類と連続する回数についての情報も求める。

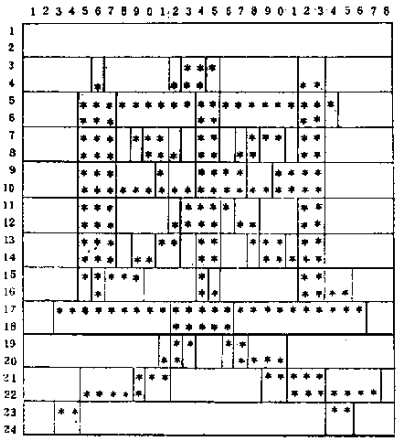


図３－２　２行単位の分割

表3－2　ランレングスコード法のためのデータ（1行単位、２４×２４ドットパターン、文字種全体）

連続する回数	(a)文字部分（黒）	(b)地の部分（白）	(a)+(b)
1	15771	27934	43705
2	82433	28694	111127
3	20600	29266	49866
4	4736	21253	25989
5	2558	14730	17288
6	1458	11359	12817
7	1267	9401	10668
8	1100	7316	8416
9	1141	4849	5990
10	1106	4074	5180
11	888	5781	6669
12	1044	2289	3333
13	878	1212	2090
14	842	1320	2162
15	640	968	1608
16	797	808	1605
17	575	586	1161
18	595	629	1224
19	273	662	935
20	316	395	711
21	204	273	477
22	265	124	389
23	284	55	339
24	734	9121	9855

表3－3　ランレングスコード法のためのデータ（2行単位、２４×２４ドットパターン、文字種全体）

連続する回数	(a) (白白) の部分	(b) (白黒) の部分	(c) (黒白) の部分	(d) (黒黒) の部分	(a)+(b)+(c)+(d)
1	16519	17217	17991	15074	66801
2	15688	6914	6756	43883	73241
3	14325	3310	2676	6868	27179
4	9912	1437	1574	940	13863
5	6194	1077	1148	275	8694
6	4572	949	1015	143	6679
7	3860	687	704	97	5348
8	2843	601	627	89	4160
9	1892	350	361	82	2685
10	1584	276	264	87	2211
11	2325	277	204	62	2868
12	919	227	176	128	1450
13	497	116	72	84	769
14	557	84	99	57	797
15	417	60	36	44	557
16	350	44	53	66	513
17	270	44	38	27	379
18	286	46	50	31	413
19	352	38	32	5	427
20	224	18	8	28	278
21	145	39	3	15	202
22	70	7	3	10	90
23	42	1	2	16	61
24	2881	3	37	48	2969

表3－4　ランレングスコード法のためのデータ（1行単位、24×24ドットパターン、漢字のみ）

連続する回数	(a)文字部分（黒）	(b)地の部分（白）	(a) ＋ (b)
1	14557	27096	41653
2	78326	27616	105942
3	18795	28312	47107
4	4308	20163	24471
5	2371	13581	15952
6	1263	10099	11362
7	1157	8258	9415
8	978	6331	7309
9	1074	4052	5126
10	1037	3228	4265
11	845	4638	5483
12	992	1785	2777
13	852	897	1749
14	811	946	1757
15	622	692	1314
16	768	586	1354
17	559	401	960
18	558	508	1066
19	264	558	822
20	298	301	599
21	199	212	411
22	252	84	336
23	279	39	318
24	694	6386	7080

表3－5　ランレングスコード法のためのデータ（2行単位、24×24ドットパターン、漢字のみ）

連続する回数	(a) (白白) の部分	(b) (白黒) の部分	(c) (黒白) の部分	(d) (黒黒) の部分	(a)+(b)+(c)+(d)
1	16004	15755	16382	13662	61803
2	15073	6389	6243	41997	69702
3	13809	3099	2471	6487	25866
4	9276	1360	1504	858	12998
5	5605	1021	1107	248	7981
6	3913	913	971	110	5907
7	3265	652	680	84	4681
8	2356	575	599	77	3607
9	1453	332	341	77	2203
10	1161	265	246	82	1754
11	1791	270	200	57	2318
12	688	223	171	115	1197
13	351	114	71	82	618
14	371	80	98	55	604
15	307	58	36	39	440
16	245	42	53	58	398
17	189	41	38	27	295
18	225	37	44	25	331
19	307	35	32	4	378
20	185	16	7	21	229
21	118	39	3	14	174
22	56	6	3	7	72
23	32	1	2	16	51
24	1708	0	36	34	1778

表3－6　ランレングスコード法のためのデータ（1行単位、24×24ドットパターン、かな・記号のみ）

連続する回数	(a)文字部分（黒）	(b)地の部分（白）	(a) ＋ (b)
1	1214	838	2052
2	4107	1078	5185
3	1805	954	2759
4	428	1090	1518
5	187	1149	1336
6	195	1260	1455
7	110	1143	1253
8	122	985	1107
9	67	797	864
10	69	846	915
11	43	1143	1186
12	52	504	556
13	26	315	341
14	31	374	405
15	18	276	294
16	29	222	251
17	16	185	201
18	37	121	158
19	9	104	113
20	18	94	112
21	5	61	66
22	13	40	53
23	5	16	21
24	40	2735	2775

表3－7　ランレングスコード法のためのデータ（2行単位、24×24ドットパターン、かな・記号のみ）

連続する回数	(a)（白白）の部分	(b)（白黒）の部分	(c)（黒白）の部分	(d)（黒黒）の部分	(a)+(b)+(c)+(d)
1	515	1462	1609	1412	4998
2	615	525	513	1886	3539
3	516	211	205	381	1313
4	636	77	70	82	865
5	589	56	41	27	713
6	659	36	44	33	772
7	595	35	24	13	667
8	487	26	28	12	553
9	439	18	20	5	482
10	423	11	18	5	457
11	534	7	4	5	550
12	231	4	5	13	253
13	146	2	1	2	151
14	186	4	1	2	193
15	110	2	0	5	117
16	105	2	0	8	115
17	81	3	0	0	84
18	61	9	6	6	82
19	45	3	0	1	49
20	39	2	1	7	49
21	27	0	0	1	28
22	14	1	0	3	18
23	10	0	0	0	10
24	1173	3	1	14	1191

表3－8　ランレングスコード法のためのデータ（1行単位、28×24ドットパターン、文字種全体）

連続する回数	(a)文字部分（黒）	(b)地の部分（白）	(a) ＋ (b)
1	15661	15705	31366
2	82594	30560	113154
3	20768	31773	52541
4	4901	26155	31056
5	2532	23298	25830
6	1556	15580	17136
7	1290	10408	11698
8	958	8549	9507
9	1178	7222	8400
10	1158	5358	6516
11	904	3220	4124
12	1087	3346	4433
13	856	5270	6126
14	866	1609	2475
15	619	944	1563
16	788	1044	1832
17	577	821	1398
18	620	739	1359
19	305	515	820
20	338	462	800
21	196	587	783
22	269	340	609
23	256	251	507
24	469	116	585
25	176	54	230
26	134	9	143
27	0	0	0
28	1	9112	9113

表3－9　ランレングスコード法のためのデータ（2行単位、28×24ドットパターン、文字種全体）

連続する回数	(a) (白/白) の部分	(b) (白/黒) の部分	(c) (黒/白) の部分	(d) (黒/黒) の部分	(a)+(b)+(c)+(d)
1	11511	17996	18289	15007	62803
2	18424	7067	6913	43924	76328
3	16362	3352	2542	6888	29144
4	12917	1529	1755	958	17159
5	10699	1093	1138	284	13214
6	6892	969	1043	150	9054
7	4360	669	710	87	5826
8	3473	585	629	86	4773
9	2942	385	377	94	3798
10	2103	283	268	76	2730
11	1320	252	182	72	1826
12	1309	257	210	131	1907
13	2156	116	72	78	2422
14	695	84	102	65	946
15	392	61	31	42	526
16	455	41	56	65	617
17	380	47	35	30	492
18	317	46	55	30	448
19	232	36	33	5	306
20	249	20	12	29	310
21	333	34	4	15	386
22	187	7	3	10	207
23	129	6	2	16	153
24	67	2	20	41	130
25	41	0	6	7	54
26	8	0	11	0	19
27	0	0	0	0	0
28	2873	1	0	0	2874

表3－１０　ランレングスコード法のためのデータ（１行単位、２８×２４ドットパターン、漢字のみ）

連続する回数	(a)文字部分（黒）	(b)地の部分（白）	(a) ＋ (b)
1	14469	15335	29804
2	78468	29796	108264
3	18948	30860	49808
4	4472	25142	29614
5	2345	22375	24720
6	1362	14470	15832
7	1179	9310	10489
8	836	7398	8234
9	1111	6209	7320
10	1089	4438	5527
11	861	2451	3312
12	1035	2527	3562
13	830	4214	5044
14	835	1213	2048
15	601	661	1262
16	759	668	1427
17	561	551	1112
18	583	528	1111
19	296	334	630
20	320	331	651
21	191	484	675
22	256	261	517
23	251	192	443
24	436	82	518
25	171	38	209
26	133	8	141
27	0	0	0
28	0	6378	6378

表3－１１　ランレングスコード法のためのデータ（2行単位、２８×２４ドットパターン、漢字のみ）

連続する回数	(a) (白/白) の部分	(b) (白/黒) の部分	(c) (黒/白) の部分	(d) (黒/黒) の部分	(a)+(b)+(c)+(d)
1	11256	16513	16689	13589	58047
2	17941	6540	6383	42038	72902
3	15871	3141	2335	6507	27854
4	12334	1452	1685	876	16347
5	10187	1037	1097	257	12578
6	6291	933	999	117	8340
7	3794	634	686	74	5188
8	2873	559	601	74	4107
9	2406	368	357	89	3220
10	1634	271	250	71	2226
11	911	245	178	67	1401
12	908	253	205	118	1484
13	1651	114	71	76	1912
14	504	80	101	63	748
15	259	59	31	37	386
16	270	39	56	57	422
17	274	44	35	30	383
18	219	37	49	24	329
19	152	35	33	4	224
20	180	16	11	22	229
21	289	34	4	14	341
22	157	7	3	7	174
23	102	5	2	16	125
24	55	0	20	27	102
25	31	0	5	7	43
26	7	0	11	0	18
27	0	0	0	0	0
28	1701	0	0	0	1701

表3－12　ランレングスコード法のためのデータ（1行単位、28×24ドットパターン、かな・記号のみ）

連続する回数	(a)文字部分（黒）	(b)地の部分（白）	(a) ＋ (b)
1	1192	370	1562
2	4126	764	4890
3	1820	913	2733
4	429	1013	1442
5	187	923	1110
6	194	1110	1304
7	111	1098	1209
8	122	1151	1273
9	67	1013	1080
10	69	920	989
11	43	769	812
12	52	819	871
13	26	1056	1082
14	31	396	427
15	18	283	301
16	29	376	405
17	16	270	286
18	37	211	248
19	9	181	190
20	18	131	149
21	5	103	108
22	13	79	92
23	5	59	64
24	33	34	67
25	5	16	21
26	1	1	2
27	0	0	0
28	1	2734	2735

表3－13　ランレングスコード法のためのデータ（2行単位、28×24ドットパターン、かな・記号のみ）

連続する回数	(a) (白/白) の部分	(b) (白/黒) の部分	(c) (黒/白) の部分	(d) (黒/黒) の部分	(a)+(b)+(c)+(d)
1	255	1483	1600	1418	4756
2	483	527	530	1886	3426
3	491	211	207	381	1290
4	583	77	70	82	812
5	512	56	41	27	636
6	601	36	44	33	714
7	566	35	24	13	638
8	600	26	28	12	666
9	536	17	20	5	578
10	469	12	18	5	504
11	409	7	4	5	425
12	401	4	5	13	423
13	505	2	1	2	510
14	191	4	1	2	193
15	133	2	0	5	140
16	185	2	0	8	195
17	106	3	0	0	109
18	98	9	6	6	119
19	80	1	0	1	82
20	69	4	1	7	81
21	44	0	0	1	45
22	30	0	0	3	33
23	27	1	0	0	28
24	12	2	0	14	28
25	10	0	1	0	11
26	1	0	0	0	1
27	0	0	0	0	0
28	1172	1	0	0	1173

(3) 統計結果の分析

1行のサイズは28×24ドット・パターンにおいて最大28で、長さを表わす部分は5ビットを必要とする。またパターンの種類を表わす部分は、1行単位の方法で1ビット、2行単位の方法で2ビット必要である。これに、連続する回数を表わす部分の長さに1、2、3、4、5ビットをそれぞれ想定して、文字種全体を表わすのに必要なビット数を求め、これと原パターンの全ドット数を比べると、この方法による圧縮率が求まる。

このようにして計算した圧縮率の期待値は表3－14、15、16のようになる。

表3－14　ランレングス法による圧縮率（全文字種）

圧縮の対象となる行数	パターンの大きさ	レングス部分の長さ（ビット）				
		1	2	3	4	5
1行単位	24×24	1.0946	1.0263	1.0282	1.1382	1.3050
	28×24	1.0833	0.9948	0.9719	1.0383	1.1895
2行単位	24×24	0.8669	0.7924	0.8170	0.9145	1.0459
	28×24	0.8099	0.7298	0.7445	0.8141	0.9281

表3－15　ランレングス法による圧縮率（かな・記号のみ）

圧縮の対象となる行数	パターンの大きさ	レングス部分の長さ（ビット）				
		1	2	3	4	5
1行単位	24×24	1.0537	0.8904	0.7536	0.7291	0.7675
	28×24	1.0464	0.8717	0.7393	0.6456	0.6704
2行単位	24×24	0.8204	0.6649	0.5784	0.5765	0.6184
	28×24	0.8109	0.6428	0.5531	0.5092	0.5415

表3－16　ランレングス法による圧縮率（漢字のみ）

圧縮の対象となる行数	パターンの大きさ	レングス部分の長さ（ビット）				
		1	2	3	4	5
1行単位	24×24	1.1008	1.0468	1.0696	1.2000	1.3862
	28×24	1.0894	1.0183	1.0071	1.0976	1.2679
2行単位	24×24	0.8740	0.8170	0.8570	0.9688	1.1123
	28×24	0.8611	0.8091	0.8191	0.9038	1.0374

文字種全体とかな・記号のみの圧縮率をレングス部分のビット数によってプロットしてみると図3－3、4になる。

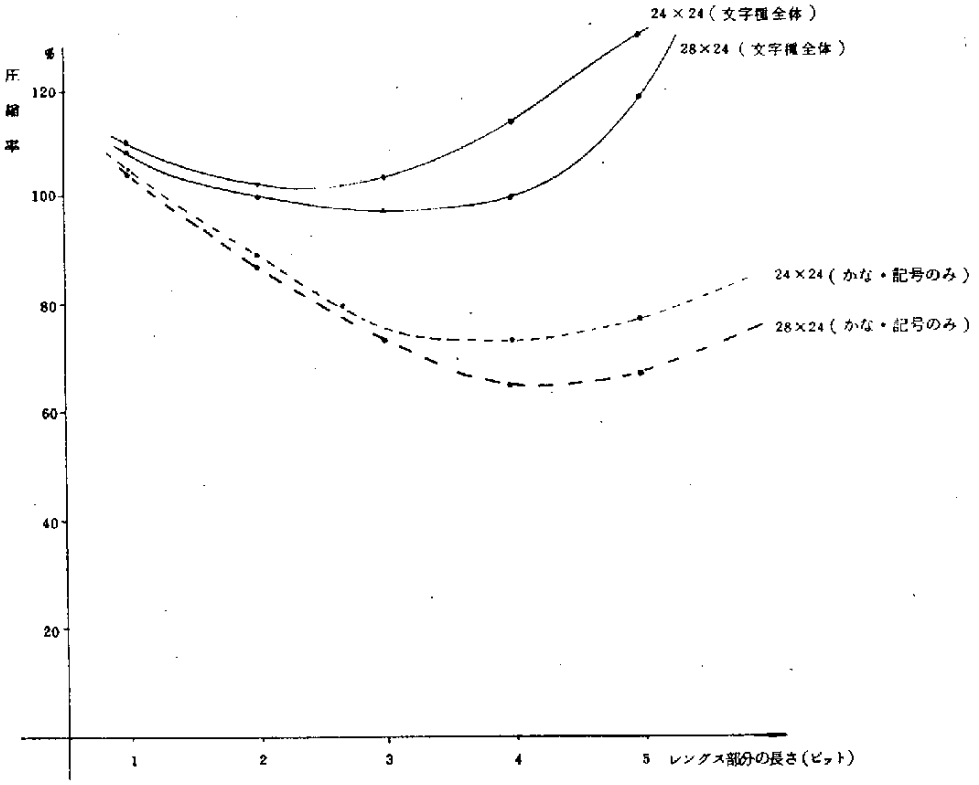


図３－３　ランレングス法（１行単位）の圧縮率

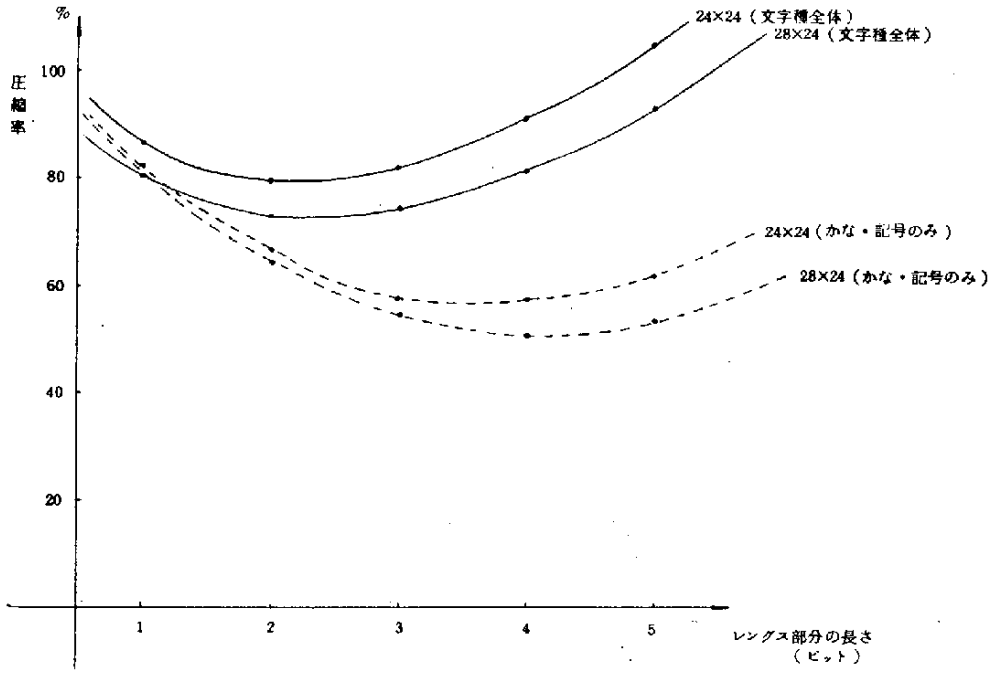


図３－４　ランレングス法（２行単位）の圧縮率

圧縮率の期待値からレングス部分の長さを適切に決定することが重要であることがわかる。また１行単位より２行単位のほうが圧縮率が多少良い。しかし、全文字種の場合、最も良いものでも７０％以上であり、２４×２４ドット程度の文字パターンの圧縮には、この方法はあまり期待できないようだ。しかし、空白部分の多いかな・記号類では圧縮率が５０％程度までいき、このことから、この方式は２４×２４ドット程度よりドット構成の大きな文字パターンあるいは行間の空間をも含んだ文章全体を圧縮の対象とした場合に適しているようだ。

3.4 行パターンによる圧縮方式

(1) 圧縮方式

この方式は、文字パターンを行ごとに分割してこの行パターンの種類の数を調べる。この圧縮方式の原理は、このシステムで扱う文字種はたかだか2,５８３種であるから、分割した全ての行パターンが異なるとしても2,５８３×２４＝６１,９９２種類であり、この行パターンに符号をふる場合に、１６ビットあれば表現できる。原パターンでは、この行パターンを表現するのに２４ビットまたは２８ビット使用してきたのであるから、これでも圧縮になる。実際に、同一の行パターンが多くあるはずであるから、種類は少なくかなりの圧縮が期待できる。というのが、この方式のねらいである。しかし、文字パターン自体を表現するには、これだけでよいのだが、復元するために、２４ビットのフル行パターンを行パターンの種類分だけ持たねばならず、パターン・テーブルの記憶量がばう大になる。よってこの方式の圧縮の効果は、行パターンの種類に依存している。

(2) 統計方法とその結果

この方式に必要な情報の求め方は、まず、文字パターンを行パターンに分割する。全文字種について分割が終了したら、この行パターンをバイナリ・ソートする。このソートされた行パターンを読んでいって、行パターンの種類、それぞれの行パターンの頻度を調べる。この調査の結果は表３－１７のとうりである。

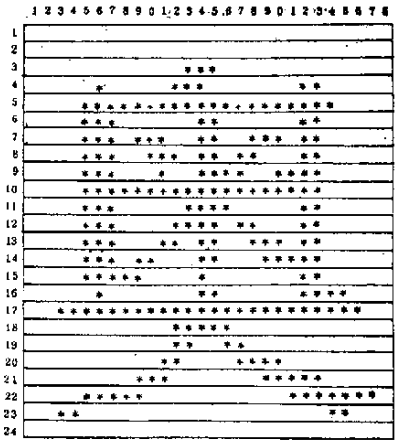


図３－５　行パターンによる圧縮方式の分割法

表３－１７　行パターンの統計結果

	全　体	漢字のみ	かな・記号のみ
２４×２４	２２０７０行	２１２１７行	１７３５行
２８×２４	２３０９０行	２２２１３行	１７５７行

(3)　統計結果の分析

この方式によるパターンの圧縮率を計算する方法は以下のようである。

この方式で文字フォントの記憶エリアはパターン記述エリア（行パターン番号の組合せでパターンを表わす）と行パターン記憶エリアに分かれる。パターン記述エリアは、行パターンの種類を表現できるビット数×全フォントの行パターンの数で表わされる。行パターン記憶エリアは、行パターンの種類×行パターンのビット数で表わされる。たとえば、２８×２４のパターン２５８３種の行パターンは２３０９０であるが、２３０９０＝2^{1449}であるから、行の種類を識別するのに１５ビット必要である。よってパターン記述エリアは１５×２５８３×２４＝９２９８８０ビット，行パターン記憶エリアは23090×28=646520ビットであり、合わせて１５７６４００ビット必要である。原フォントは２８×２４＝１７３５７７６であるからこれで割ると圧縮率９０.８１８％となる。このようにして、全てのデータについて圧縮率を計算すると表３－１８のようになる。

表３－１８　行パターンによる圧縮率

	パターンの構成	
	２４×２４	２８×２４
全文字種 ２５８３	0.９８１０１	0.９０８１８
漢字のみ ２２４４	1.０１８９６	0.９４８１７
かな・記号のみ ３３９	0.６７１５８	0.６０８８１

この結果から、この方式は、２８（２４）×２４ドット程度の文字パターンの圧縮はあまり期待できないことがわかる。これは行パターン・テーブルの記憶エリアが大きすぎ、これによってパターン記述部分が節減されても効果がなくなってしまった結果である。ただし、かな・記号類のみの圧縮率はよい。これにより、使用分野を限定して漢字の文字種を押えることができれば圧縮率が良くなることが予測できる。

3.5　ラインドット方式

(1)　圧縮方式

この方法は行単位に文字パターンを分割して、文字部分（パターン上の黒の部分）がはじまるX座標と、文字部分がきれるX座標、あるいは文字部分がはじまるX座標と連続する長さを示すこ

とにより文字パターン全体を表現しようとする方式である。一行の中に1度も文字部分が現われない空白行の場合は、Y座標が与えられないために、パターンの復元がむずかしくなる。このため、空白行がある場所に空白行があることを示すコードを入れる必要がある。

(2) 統計方法とその結果

この方式についての統計のとり方は、以下のようになる。

この方法において必要な情報は、行単位に分割した場合にその中に含まれる文字部分の数と空白行の数である。各行ごとに左はじからみていき、その行に含まれる文字部分の数を調べる。また、その行が空白行である場合はそのカウントをとる。これを文字種全体について求める。図3－6においては28×24ドットのパターンの例であるが、同様に、24×24ドットパターンについても調べる。文字種についても、全文字種についてだけでなく、漢字のみ、あるいは、かな・記号のみの場合についても調べる。

この調査の結果は表3－19のとうりである。

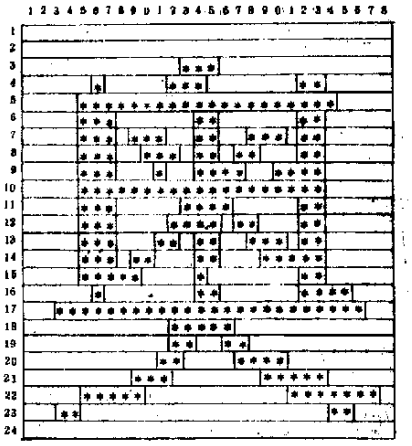


図3－6　ラインドット法の分割

表3－19　ラインドットの統計結果

		文字部分の数	空白行の数
24×24	全文字種 2583	140504	9121
	漢字のみ 2244	131859	6386
	かな・記号 339	8645	2735
28×24	全文字種 2583	141057	9112
	漢字のみ 2244	132398	6378
	かな・記号 339	8659	2734

(3) 統計結果の分析

この方式によるパターンの圧縮率を計算する方法は、文字パターンが28×24ドットとして

も行の長さは２８であるから、Ｘ座標を表わすにも、連続する長さを表わすのにしても５ビットで十分である。また空白行を他のものと区別するために、５ビット全てＯＮの状態のビット列にして使用する。すると、１文字を表記するために必要な情報を求める計算式は以下のようになる。

$$a \times 10 + b \times 5 \quad \begin{cases} a：文字部分の数 \\ b：空白行 \end{cases}$$

この計算式で求めた圧縮率の期待値は表３－２０のようである。

表３－２０　ラインドット方式による圧縮率

	パターンの構成	
	２４×２４	２８×２４
全文字種 ２５８３	0.９７５０	0.８３８９
漢字のみ ２２４４	1.０４４９	0.８９９１
かな・記号のみ ３３９	0.５１２８	0.４４０１

この結果、この方式では圧縮率があまり期待できないことがわかる。本来、英・数字・記号類用に使われていた方式であることから、文字種をかな・記号のみとした場合の圧縮率は大変よいが、字形の複雑な漢字、とくに明朝体に類する書体のパターンには適していないという結論になる。

3.6　ハフマンの符号化法

(1)　圧縮方式

この方法は、まず文字パターンのドット・マトリックスを２×２あるいは４×１のサブパターンに分割する。このサブパターンを表現するのに、もとの文字パターンでは４ビットを必要としているが、サブパターンの種類（１６種類）ごとの出現確率を文字種全体について求める。この確率から、ハフマンの符号化法によりコードをふり直すことによって、コードの平均語長（期待値）を最小にする方式である。この他にサブパターンの出現確率のみでなく、隣接するサブパターン同士の関連性を考慮する方式も試みる。これはサブパターン同士に順序（左から右、上から下）を決めておいて、この順序に従って、１６種のサブパターンそれぞれの後に続くサブパターンの種類とその確率を文字種全体について求める。このようにして求めた確率から、１６種のサブパターンごとに、その後に続くサブパターンのコードをハフマンの符号化法によってつけていく。つまり、１６種のサブパターンにはその前にあるサブパターンの種類によって１６種の異なったコードを持つことになる。文字パターンのサブパターンはサブパターンの順序に従って、その前のサブパターンの種類によって定まるコード系を用いてコードがつけられる。サブパターンの順

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	…	…	…	…	33	34	35	36
37	38	…	…	…	…	…	…	…	…	47	48
49											60
61											
											108
										119	120
121	…	…	…	…	…	…	…	129	130	131	132
133	134	…	…	…	…	139	140	141	142	143	144

図３－７　サブパターンの順序付け（２４×２４ドットマトリックスの場合）

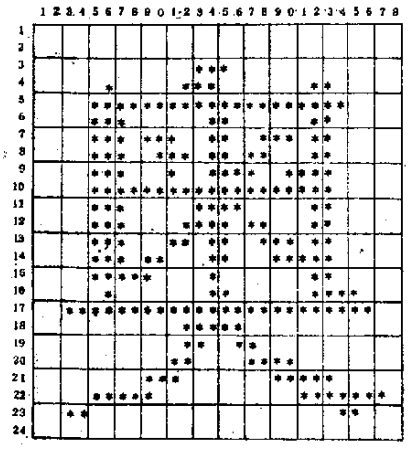


図３－８　２×２サブパターンへの分割

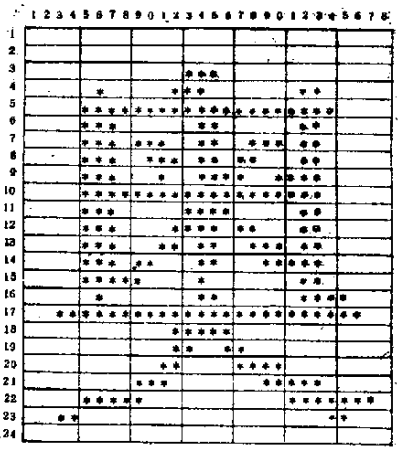


図３－９　４×１サブパターンへの分割

序で先頭のサブパターンはその前にサブパターンがないため、空白のサブパターンがそこにあったとしてコードを付っていく。文字パターンのドット構成が２８（横）×２４（縦）の場合には１６８、２４×２４の場合には１４４のサブパターンのコード付けが終了すると１文字のパターン圧縮が終わることになる。ところで、ハフマンの符号化法（以下、簡単にハフマン法と呼ぶ）とは、出現率が明らかな事象の集合の１つ１つの事象に対して、そのコード長が最小となるよう

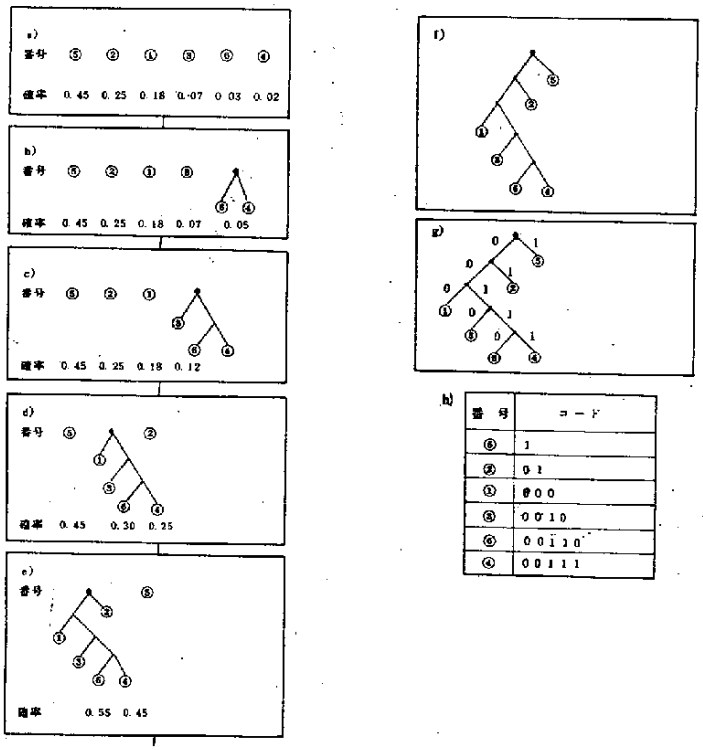
にコードを付けるものである。方法としては、起こりえる確率がわかっている事象を、確率の低い順に並べておく。この中から最も確率の低いもの２つを結びつけ、それら２つの確率を加えてできる合成要素の確率に合った所に挿入して、さらにこの中から最も確率の低いもの２つをみつけて結びつけ、それら２つの確率を加えてできる合成要素を合成確率に合った場所に入れる操作をくりかえす。この様にすると、１本のトリーができる。このトリーの根から、ある事象までの深さがその事象のコード長となる。トリーの根からたどっていき左側を０、右側を１とコード付けしていけば、ある事象のコードは根からたどっていき、その事象に至るまでの過程でコード付けすることができる。また、平均コード長とは、そのようにしてコード付けした場合のコード長の期待値のことである。

以後、各サブパターンの出現率のみにより圧縮する方式を単純ハフマン法、隣接するサブパターン同士の結びつきを考慮した圧縮方法を複合ハフマン法と呼ぶことにする。

図３－１０に６つ要素からなる集合を例にして、ハフマン法によるコード付けを説明する。

a）まず６つの要素を、その出現確率の低い順に並び変える。

b）出現確率の最も低い２つの要素を結びつける。この合成要素の確率は２つの要素の確率を加えたものとする。また合成要素の表現として要素が⑥と④の場合にはｂ）の右はじのようなトリー表現をつかうことにする。合成要素を作成した後にまた、出現確率の低い順に並び変える。



番号	コード
⑤	1
②	0 1
①	0 0 0
③	0 0 1 0
⑥	0 0 1 1 0
④	0 0 1 1 1

図３－１０　ハフマン法によるコード化例

c）～e）　b）の操作を次々に行なっていく。

f）このようにして要素数が1つになったらこの操作は終わりでトリーができあがる。

このトリーを根からたどっていって左に降るのを0、右に降るのを1とすれば、この例におけるトリーはg）のようになる。それぞれの単一要素のコードは、根からその要素に到る道すじで表わされるから、h）のようにコード付けされる。

この例の平均語長（期待値）はコード長×出現率の和で求まるから、これを計算すると2.02ビットになる。一般に、6つの要素を識別するには、3ビット必要であるから、この方法により多少、コード付けが圧縮されたことになる。

(2)　統計方法とその結果

この方法の対象となる文字パターンは、28×24ドットの文字パターンであるが、これとこの文字パターンの横の両端を2ドットずつ除いて作った24×24ドットの文字パターンも扱う。まず、これらの文字パターンを2×2あるいは4×1のサブパターンに分割し、このサブパターン同士の統計を求める。

この統計には大きく分けて2種類ある。1つはサブパターンそれぞれの先験確率を文字種全体について求めるか、あるいはかな・記号と漢字に分けてそれぞれの文字種全体について求めるものである。ここで、かな・記号と漢字を分けて統計をとる目的は漢字、かな・記号では文字パターンの複雑さが異なり、サブパターンに分けての統計にも違いが出てくる可能性があるためである。

第2の方法は、隣り合うサブパターン同士の結びつきに関するものである。このためにサブパターンを左から右、上から下に順序付けし、この順で隣り同士の結びつきを調べている。この場合、行の横方向の関係を調べる際には問題はないが、行が変わる際に右端の処理をどうするかで方法が変わる。たとえば、図3－7で24番目と25番目のサブパターン同士は実際には隣り合ってはいない。しかし、ここで扱っているのは文字パターンであり、文字の回りには空白があると仮定することができる。この特徴を用いると、この空白と隣り合っているパターン同士も結びつきが高いことになる。ここでは、この間に空白のパターン（空サブパターン）を入れた場合の統計と、行の末端（右端）と次の行の先端（左端）を接続した統計をそれぞれ求めている。

この隣り合うサブパターンの関係を調べるために、図示した順番にサブパターンを調べてゆく。あるサブパターンの次にどのサブパターンが続くのかを調べて統計を求める。なお、最初のサブパターンには比べるサブパターンがないが、空サブパターンを仮に置いている。この隣り合うサブパターン同士の関係を調べる際にも、文字種として、全体、漢字のみ、かな・記号のみの場合に分けて調べる。次に、複合ハフマン法の場合の圧縮率の期待値の求め方は以下のようである。まず、先行するサブパターンの種類とその後に続くサブパターンの種類を調べるために、先行するサブパターン16種類との対応関係の256通りについて、頻度を求める。対象とする全文字種

の文字パターンについて、この２５６種のカウント情報を調べ、これを基にして先行するサブパターンに対する後に続くサブパターンの確率を求める。この確率を使ってハフマンの符号化法により、コード付けを行なう。これによって、先行するサブパターンの種類ごとに１６種類のコード系ができる。そしてこのコード系ごとに平均コード長（期待値）を求める。一方、先行するサブパターンの種類ごとの確率も求め、これに、このサブパターンが先行する場合のコード系の平均語長を乗じた値を１６種類加えると複合ハフマン法全体の期待値が求まる。

以上の方法による例として、２８×２４ドットパターンを全文字種について２×２のサブパターンに分割して単純ハフマン法を適用したときの結果を表３－２１に示す。

表3－２１　単純ハフマン法用サブパターンの出現頻度表

パターン	パターン番号	頻　　度	割　　合
□	1	238456	0.54951
▨	16	35003	0.08066
▨	4	26696	0.06152
▨	13	25718	0.05927
▨	6	19336	0.04456
▨	11	19263	0.04439
▨	9	9938	0.02290
▨	2	9882	0.02277
▨	15	9635	0.02220
▨	8	9479	0.02184
▨	3	7501	0.01729
▨	12	7451	0.01717
▨	14	7149	0.01647
▨	5	7026	0.01619
▨	7	860	0.00198
▨	10	551	0.00127
—	計	433944	———

表3－２２　単純ハフマン法によるコード表

パターン番号	コード長	コ　ー　ド
1	1	0
16	4	1000
4	4	1010
13	4	1011
6	4	1101
11	4	1110
9	5	11000
2	5	11001
15	5	11110
8	5	11111
3	6	100101
12	6	100110
14	6	100111
5	7	1001000
7	8	10010010
10	8	10010011

なお、表３－２１、２２にあげた例での平均語長（期待値）は2.605ビットで、圧縮率の期待値は圧縮する前が４ビットであるから2.605÷4＝0.6512となる。同一条件における複合ハフマン法の結果として、先行するサブパターンの種類ごとの平均語長等を表３－２３に示す。

なお、複合ハフマン法の例では、空サブパターンを各行の始端に入れて行なっている。

表３－２３　複合ハフマン法の結果
（２８×２４ドット、全文字種、２×２サブパターン、各行の先頭サブパターンの前に空白サブパターンを置く）

先行するサブパターンの番号	(a) 頻度	(b) 割合	(c) 平均語長	(b) × (c)
1	240159	0.553443	1.727706	0.955581
2	9882	0.022277	3.105844	0.070073
3	6717	0.015448	1.666264	0.025744
4	26696	0.061522	2.292110	0.141101
5	7026	0.016119	3.065775	0.049644
6	19336	0.044456	1.972077	0.087787
7	860	0.001978	2.831399	0.005561
8	9479	0.021844	3.144400	0.068688
9	9242	0.021300	1.775055	0.037800
10	551	0.001127	2.943744	0.003374
11	19040	0.043888	1.821116	0.079911
12	7451	0.017177	2.545299	0.043700
13	25718	0.059277	2.258033	0.133822
14	7149	0.016477	2.957199	0.048722
15	9635	0.022220	2.476288	0.054988
16	35003	0.080666	3.002200	0.242166
計	433944	————	————	2.049922 ビット (51.24797%)

また同じデータから、先行するサブパターンの種類ごとに、全体で出現率が９０％になるまでのサブパターンの種類を示したのが図３－１１である。

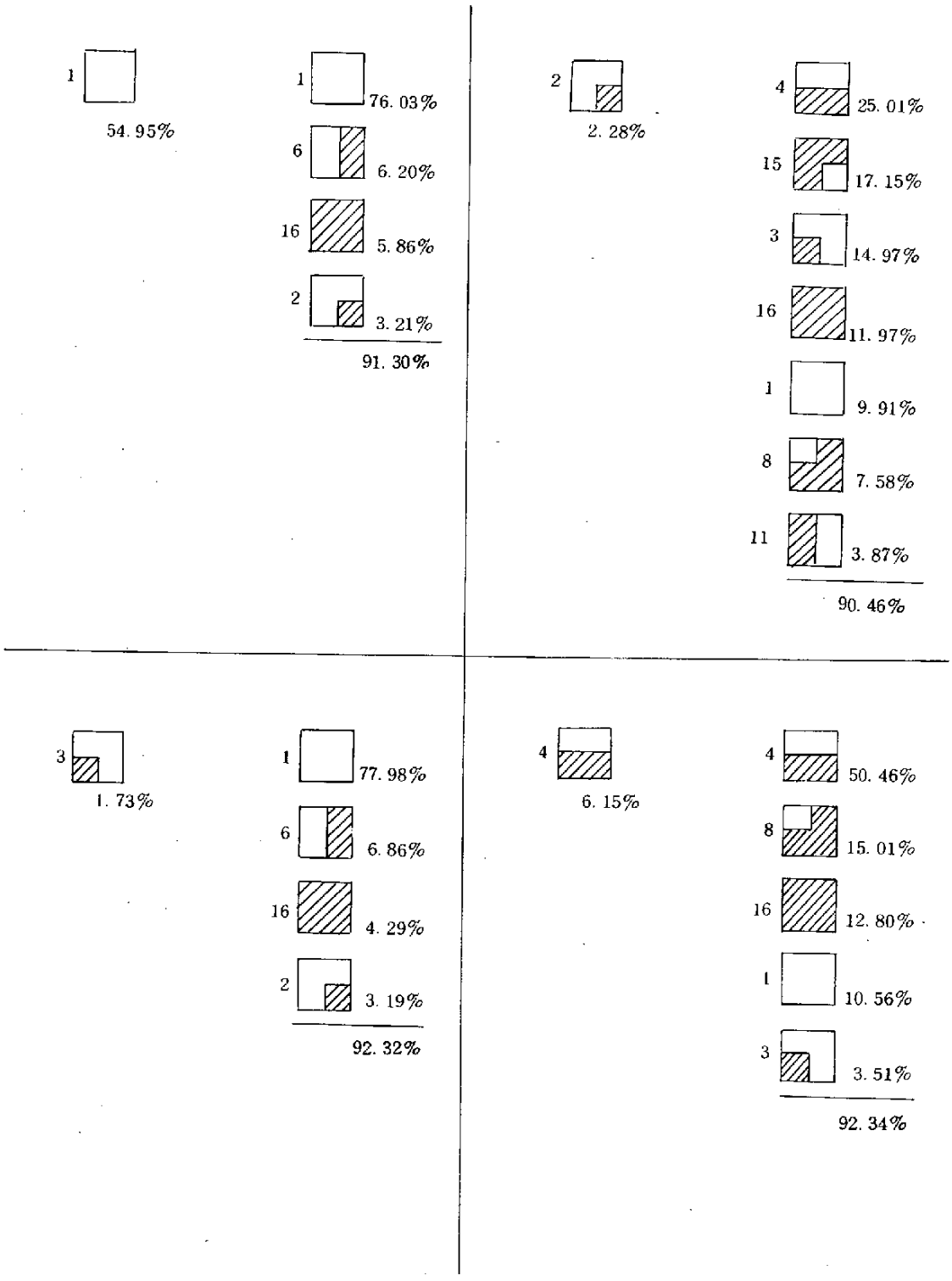
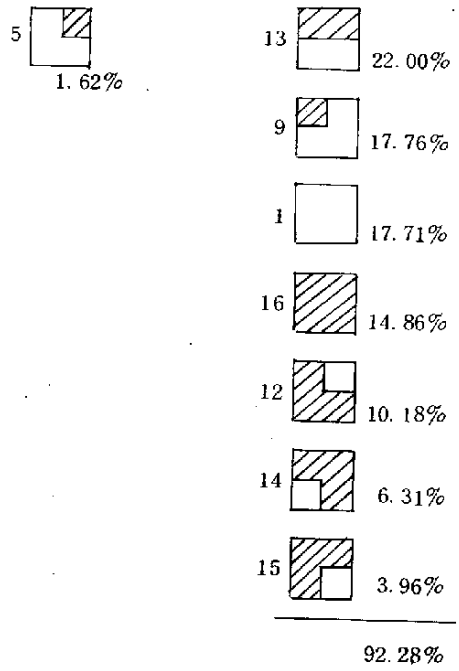
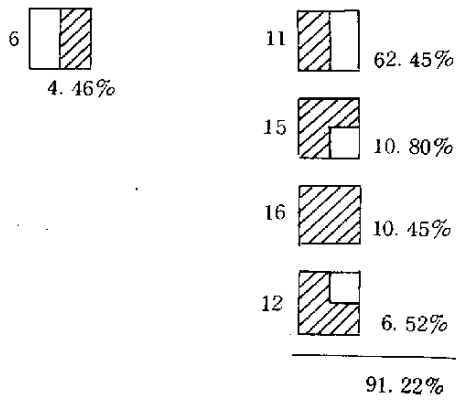
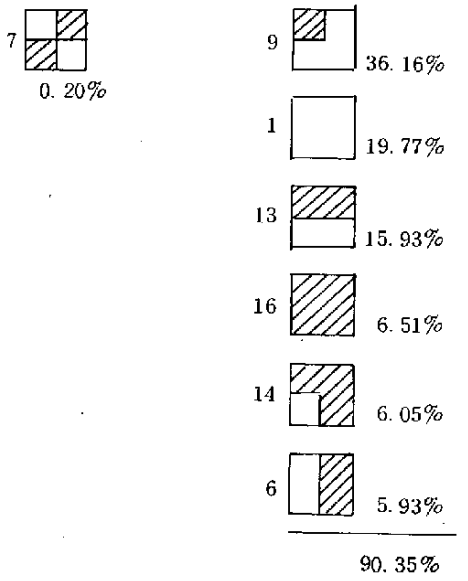


図３－１１　サブパターン同士の関係


5
1. 62%
13
22. 00%
9
17. 76%
1
17. 71%
16
14. 86%
12
10. 18%
14
6. 31%
15
3. 96%
92. 28%


6
4. 46%
11
62. 45%
15
10. 80%
16
10. 45%
12
6. 52%
91. 22%


7
0. 20%
9
36. 16%
1
19. 77%
13
15. 93%
16
6. 51%
14
6. 05%
6
5. 93%
90. 35%

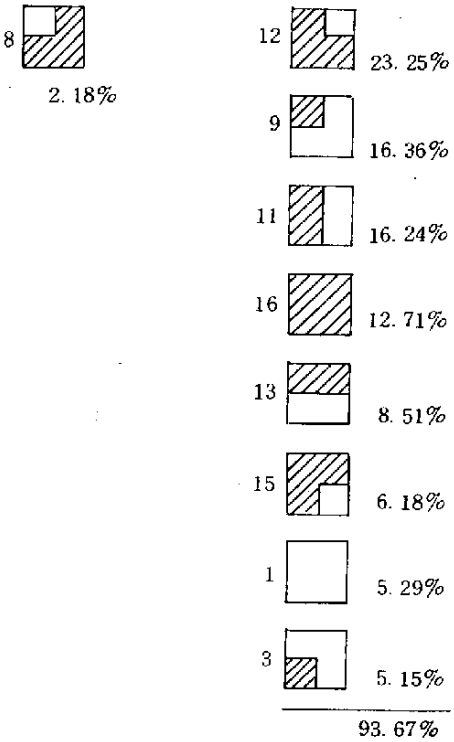
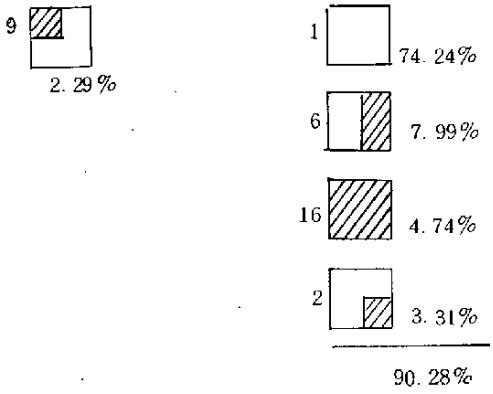
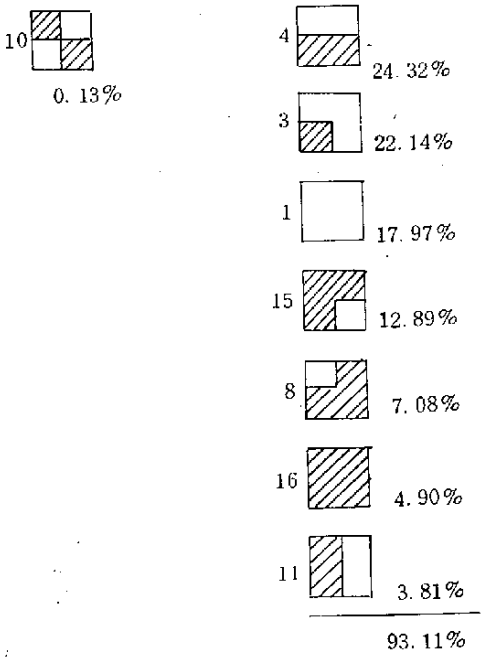
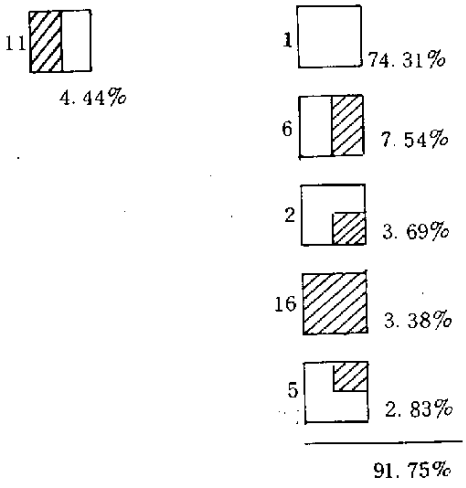

8
2. 18%
12
23. 25%
9
16. 36%
11
16. 24%
16
12. 71%
13
8. 51%
15
6. 18%
1
5. 29%
3
5. 15%
93. 67%

図３－１１（つづき）


9
2.29%
1
74.24%
6
7.99%
16
4.74%
2
3.31%
90.28%


10
0.13%
4
24.32%
3
22.14%
1
17.97%
15
12.89%
8
7.08%
16
4.90%
11
3.81%
93.11%


11
4.44%
1
74.31%
6
7.54%
2
3.69%
16
3.38%
5
2.83%
91.75%

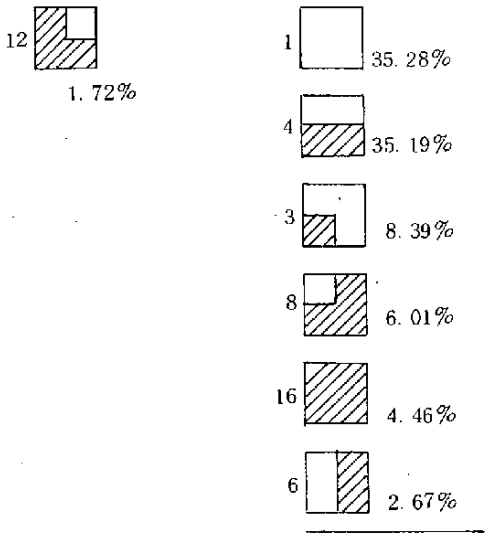

12
1.72%
1
35.28%
4
35.19%
3
8.39%
8
6.01%
16
4.46%
6
2.67%
92.00%

図３－１１（つづき）

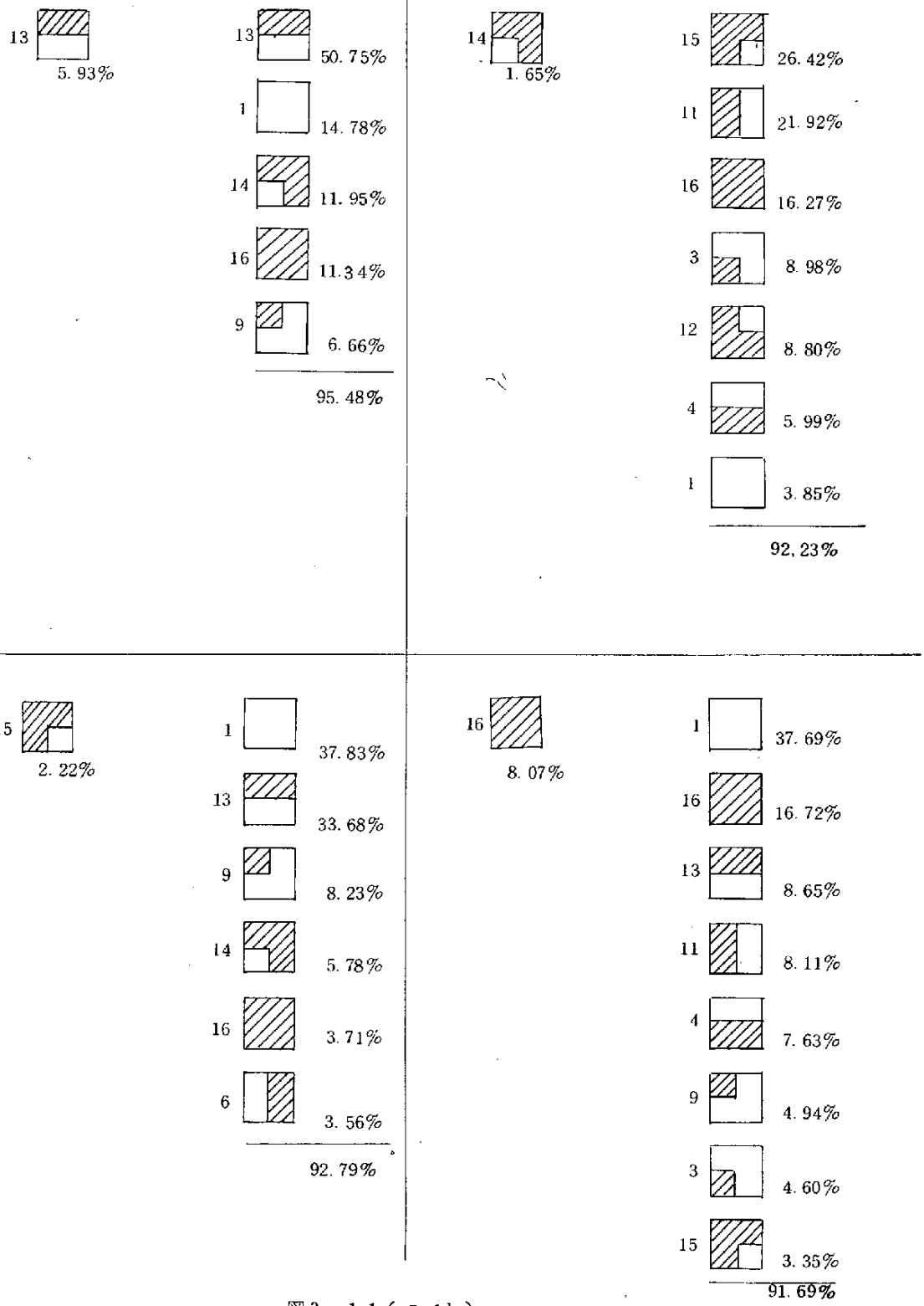

13
5. 93%
13
50. 75%
1
14. 78%
14
11. 95%
16
11.34%
9
6. 66%
95. 48%
14
1. 65%
15
26. 42%
11
21. 92%
16
16. 27%
3
8. 98%
12
8. 80%
4
5. 99%
1
3. 85%
92, 23%
15
2. 22%
1
37. 83%
13
33. 68%
9
8. 23%
14
5. 78%
16
3. 71%
6
3. 56%
92. 79%
16
8. 07%
1
37. 69%
16
16. 72%
13
8. 65%
11
8. 11%
4
7. 63%
9
4. 94%
3
4. 60%
15
3. 35%
91. 69%

図３－１１（つづき）

(3) 統計結果の分析

統計結果の圧縮率を表3－24、25に示す。なお、行の始端の処理とは、空サブパターンの挿入の処理をいう。

表3－24　単純ハフマン法の圧縮率の期待値

パターンの構成	サブパターンの構成	全体 2583	漢字 2244	かな・記号 339	混合
24×24	2×2	2.822ビット (70.56%)	2.965ビット (74.12%)	1.866ビット (46.66%)	2.821ビット (70.52%)
	4×1	2.863ビット (71.57%)	3.009ビット (75.24%)	1.877ビット (46.91%)	2.860ビット (71.51%)
28×24	2×2	2.605ビット (65.12%)	2.732ビット (68.30%)	1.747ビット (43.69%)	2.603ビット (65.07%)
	4×1	2.611ビット (65.29%)	2.743ビット (68.58%)	1.736ビット (43.40%)	2.611ビット (65.27%)

表3－25　複合ハフマン法による圧縮率の期待値

パターンの構成	サブパターンの構成	行の始端の処理	全体 2583	漢字 2244	かな・記号 339	混合
24×24	2×2	最初の行のみ	2.204ビット (55.09%)	2.286ビット (57.14%)	1.613ビット (40.31%)	2.198ビット (54.94%)
		各行	2.187ビット (54.66%)	2.267ビット (56.67%)	1.609ビット (40.22%)	2.181ビット (54.52%)
	4×1	最初の行のみ	2.426ビット (60.65%)	2.530ビット (63.25%)	1.680ビット (42.00%)	2.418ビット (60.46%)
		各行	2.407ビット (60.18%)	2.513ビット (62.82%)	1.680ビット (41.99%)	2.404ビット (60.09%)
28×24	2×2	最初の行のみ	2.050ビット (51.25%)	2.123ビット (53.09%)	1.528ビット (38.20%)	2.045ビット (51.12%)
		各行	2.050ビット (51.25%)	2.123ビット (53.08%)	1.528ビット (38.20%)	2.045ビット (51.12%)
	4×1	最初の行のみ	2.246ビット (56.15%)	2.340ビット (58.51%)	1.602ビット (40.05%)	2.243ビット (56.08%)
		各行	2.250ビット (56.25%)	2.346ビット (58.66%)	1.602ビット (40.05%)	2.248ビット (56.21%)

ところで、これらの2つの表に示した圧縮率の期待値はトリー・テーブルのサイズを考察していないので、ここでトリー・テーブルの記憶エリアの大きさを考慮した圧縮率の期待値を計算することにする。1種類のトリー・テーブルに対するメモリ容量は3バイト×16で十分であるから48バイト必要である。単純ハフマン法では、漢字とかな・記号類を分離して別々のトリーテーブルを使用する混合方式の場合で2種類、その他が1種類必要である。複合ハフマン法では、混

合方式の場合が３２種類、その他が１６種のトリー・テーブルを必要とする。

トリー・テーブルのサイズを考慮した圧縮率の期待値を表３－２６、２７に示す。

表３－２６　単純ハフマン法の期待値（トリー・テーブル・サイズを考慮）

パターンの構成	サブパターンの構成	全体 2583	漢字 2244	かな・記号 339	混合
24×24	2×2	70.59%	74.15%	46.86%	70.52%
	4×1	71.60%	75.27%	47.11%	71.56%
28×24	2×2	65.14%	68.33%	43.71%	65.11%
	4×1	65.31%	68.61%	43.42%	65.31%

表３－２７　複合ハフマン法の期待値（トリー・テーブル・サイズを考慮）

パターンの構成	サブパターンの構成	行の始端の処理	全体 2583	漢字 2244	かな・記号 339	混合
24×24	2×2	最初の行のみ	55.50%	57.62%	43.46%	55.77%
		各行	55.07%	57.15%	43.37%	55.35%
	4×1	最初の行のみ	61.06%	63.83%	45.15%	61.29%
		各行	60.59%	63.30%	45.14%	60.92%
28×24	2×2	最初の行のみ	51.60%	53.50%	40.90%	51.83%
		各行	51.60%	53.49%	40.90%	51.83%
	4×1	最初の行のみ	56.50%	58.92%	42.75%	56.79%
		各行	56.60%	59.07%	42.75%	56.92%

以上の結果からこの方式は前記3方式よりもかなり圧縮効果がある。しかし、単純ハフマン法（サブパターンの出現率のみによる方法）では２４×２４ドットのパターンで７０%程度、２８×２４ドットのパターンで６５%と圧縮率はそれほどでもなく、サブパターンのドット構成の種類（２×２、４（横）×１（縦）ドット）によってはあまり効果はない。また混合方式（漢字とかな・記号類の別のトリーテーブルで圧縮・復元する方式）と漢字とかな・記号類を同一トリー・テーブルで圧縮する場合との差もほとんどない。一方、複合ハフマン法（隣接するサブパターン同士の関連性を考慮した方法）では、２８×２４ドットのパターンの場合で圧縮率は５２%程度、２４×２４ドットのパターンの場合には５５%程度で、圧縮効果がかなり得られる。また混合方式では、混合方式をとらない方式に比べてトリー・テーブルの分だけ圧縮率が悪くなってしまい、圧縮効果を相殺してしまった。

ところで、サブパターンのドット構成の違いによって５%程度違い、４×１ドットの方が悪くなる。圧縮の段階でサブパターンをみていく際に、行の最後から次行の最初のサブパターンにい

くのにサブパターンは文字パターン上で連続しない。そこで、行が変わるごとに、ここに、空サブパターンを置いてやる必要があるかについては約0.5％程度の差しかないことがわかった。また、２８×２４ドットのパターンの場合には各行に空サブパターンを置いた方がかえって悪くなることがある。結局、２８×２４ドットで最も圧縮率がよいのがサブパターン構成が２×２ドットで、最初の行のみ空サブパターンを置いたものであり、２４×２４ドットのパターンの場合にはサブパターンの構成が２×２ドットで、各行の先頭に空サブパターンを置いて圧縮する方式がもっとも圧縮率がよい結果が得られた。

3.7 圧縮方式の検討

これまで述べてきた各圧縮方式を圧縮率で比較したのが表３－２８である。

この圧縮方式の実験でとりあげた方式のうちでは、ハフマンの符号化法による方式が圧縮率の点で最も優れている。特に、隣接するサブパターン同士の関連性を利用した複合ハフマン法では約１／２の圧縮率が得られた。次に圧縮率が良いのがランレングスコード法の２行単位に処理したもので７０％台の圧縮率がある。この方式は対象となる文字パターンのドット構成が大きくなると、この圧縮の効果も良くなることが予想される。その他の圧縮方式はあまり期待できそうにない。これらの結果は、圧縮の対象となった疑似明朝体の文字パターン（２８×２４あるいは両端を２ビットずつ切捨てた２４×２４ドット）の影響によるものもあろうが、圧縮率から圧縮方式の評価をするとすれば複合ハフマン法が選定される。

表3－28　各方式の圧縮率

方式		圧縮率(%) ２４×２４	２８×２４
ランレングスコード法	１行単位	１０２.６３	９７.１９
	２行単位	７９.２４	７２９８
行パターンによる方式		９８.１０	８０.８２
ラインドット方式		９７.５０	８３.８９
単純ハフマン法	サブパターン ２×２	７０.５９	６５.１４
	サブパターン ４×１	７１.６０	６５.３１
複合ハフマン法	サブパターン ２×２	５５.０７	５１.６０
	サブパターン ４×１	６０.５９	５６.５０

しかし、圧縮方式を選択する場合、圧縮率もさることながら、圧縮した文字パターンの復元処理の複雑さやその処理時間等も大きな要素となる。そこで以下に、ここで取上げた各方式の復元処理について考察してみる。

まず、ランレングスコード法であるが、この方式で２５８３字種の文字パターン（２４×２４ドット構成）を圧縮して記憶する場合、２行単位の処理で７２×０.７９２４×２５８３＝１４７Ｋバイトのメモリが必要になる。１文字分の圧縮文字パターンの大きさは最大１００バイト程度であるから記憶装置として補助記憶装置を使用した場合１回アクセスするだけで１文字分のビット列を主メモリにもってきて以下のような復元処理が行なえる。この方式の復元処理は基本的には、圧縮して記憶した文字パターン（圧縮文字パターン）のビット列を数ビットずつ（例えば４ビット）を持ってきて、これをパターン部とレングス部分（連続する長さを示した部分）に分け、パターンをレ

ングス部分で示される分だけ文字パターン復元エリアに移すことである。そして、この基本処理ルーチンを繰返して圧縮文字パターンの全てのビット列の解析が終ると文字パターン復元エリアに２４×２４ドットの文字パターンが復元される。１文字を復元するのに要する基本処理ルーチンの回数は２４×２４ドットの文字パターンで１行単位の場合が平均１９２回、２行単位の場合が平均１１４回となり、２８×２４ドットの文字パターンは１行単位の場合が平均１６３回、２行単位の場合が平均１２３回となる。

行パターンによる方式では圧縮文字パターンの記憶容量は２４×２４ドットの文字パターンを対象とする場合、$3\times\frac{13}{24}\times 24\times 2583\fallingdotseq 98.4$Ｋバイト必要である。また、行パターンの種類を記憶する行パターン・テーブルとして３×２２０７０≒６５Ｋバイト必要である。小型機以下のコンピュータでは、これらを主記憶上に持つことは難かしいため補助記憶装置を利用することになる。すると各文字の行パターンにはそれぞれ関連性はあまりないため、各行につき１回ずつ補助記憶装置にアクセスしなければならない。このため、補助記憶装置へのアクセス回数は圧縮文字パターン（１文字２４行分の行種区分からなる）のアクセスに１回、行パターンのアクセスに２４回、計２５回となる。このため、補助記憶装置として低速の装置を使用する場合には文字発生装置としての機能を満足することはできない。復元処理では１行分の行種区分から行パターンを選択し、この行パターンを文字パターン復元エリアに移す一連の操作を基本処理ルーチンとすると、この基本処理ルーチンは１文字（２４行）を復元するのに２４回必要である。なお、圧縮文字パターンから１行分の行種区分を識別するためビット操作が入るが、全体的に処理が簡単である。

ラインドット方式では、圧縮文字パターンを記憶するために、２４×２４ドット・パターンの場合７２×0.975×2583≒１７７Ｋバイト必要である。この方式でも圧縮文字パターンは補助記憶装置に記憶する必要がある。また、１文字の文字パターンは７０バイト前後であるから１回の補助記憶装置へのアクセスですむ。ラインドット方式による圧縮文字パターンは行単位に、文字部分の開始位置の座標と終端位置の座標（あるいは連続する長さ）からなっている。復元処理では開始位置と終端位置の座標を１組として読み込み、これを解析し、文字パターン復元エリアの対応する場所にビットを立てる。この処理を圧縮文字パターン全体について行なえば１文字復元することができる。１組の座標から文字パターン復元エリアの対応する位置にビットを操作するまでの一連の処理を基本処理ルーチンとすれば、１文字復元するためにこれを平均５８回くり返す必要があり、また、空白行の処理は平均４回必要である。

ハフマン法では圧縮文字パターンを記憶するために、２４×２４ドット・パターンの場合７２×0.55×2583≒１００Ｋバイト必要であるのでこれを補助記憶装置に記憶する。また、１文字分を記憶するには平均４０バイト程度で十分であるから補助記憶装置へのアクセスは１回ですむ。ハフマン法による圧縮文字パターンはサブパターンごとにコード化されたものが２４×２４ドット・パターンで１４４個（２８×２４ドット・パターンで１６８個）集まったものである。復元処理では１文字分の圧縮文字パターンをアクセスした上で、各サブパターンをコード化したものから２×２ドットのサブパターンを復元して文字パターン復元エリアの対応する位置に入れるまでの一連処理

を基本処理ルーチンとすれば１文字復元するために２４×２４ドット・パターンで１４４回（２８×２４ドット・パターンで１６８回）くり返えされる。この処理はビット単位の比較が入り多少複雑となる。

以上のようなことから圧縮方式を全般的にみると、どの圧縮方式もドット操作を含んでいる基本的な処理を６０～２００回程度くり返して文字を復元している。基本的な処理は各方式により多少処理の複雑性が異なり、また、処理のくり返し数も異なるため、どの圧縮方式が処理時間等が早いかは、あまりはっきりしない。ただ、行パターンによる方法は補助記憶装置に２５回もアクセスしなければならず、処理時間が多くかかることが予想されるので圧縮方式として適さないようだ。その他の方式では、処理の難易度に著しい差がないようなので、前述の圧縮率の比較からハフマン法が最も圧縮方式として適しているようだ。これは対象となる文字パターン（２８×２４あるいは２４×２４ドット）に対しての評価であるから、その他のドット構成の場合は確めていないのではっきりしない。なお、ここでは圧縮方式の検討を中心としたために復元の処理を実際に行なっていない。そのため処理時間やそのための処理プログラムの大きさについての評価検討は、今後の実験を通して得ることにする。

4　日本語端末用入力方法の検討

4.1　入力方式の分類

漢字かなまじりの日本語情報をコンピュータで処理するためには、何等かの方法で日本語情報を入力しなければならない。従来の英数カナ記号類だけの取扱いならば文字種が少なく、約５０個のキーで十分であったが、数千字種の漢字とかなが加わることでキーボードの配置や、コード、入力方式などのイメージをすべて変えなければならない。文字種の最大限は約５万字といわれる中で、ＪＩＳ漢字コードとして制定された6,３４９字の漢字と仮名・英字・数字・記号類４６３種、計6,８０２字種を標準文字として扱うことになるが、日常生活の中であるいは社内文書中に出現する約2,０００～3,０００字種という範囲内での入力が先決になる。

これまでに研究開発されている入力方式の分類は、日本語情報のデータ作成業務用、いわゆる専任パンチャが操作することを目的にした方式が中心となっていた。これに一般素人でも取扱いやすい方式が、オンライン化傾向とともに端末用などに要求されるようになってきたし、事務処理の中でも一般事務員が操作することが増加しつつある。

漢字テレタイプ方式、連想コード方式などを専任者用とすると、マルチストローク方式、カナ漢字変換方式などが素人用、和文タイプ方式やタブレット方式は兼用といった分類ができよう。素人用は、とくに漢字キーボードを使用しないという点から分類したが、１字１字を片手で容易に入力できるタブレット方式なども素人向けにはなっている。

数千字種という漢字をキーボードに配置し、この中から１字１字を探すという操作はとても素人には扱えないという判断から、日本語端末用の入力方式から除外し、これに代って最近のオフィス・コンピュータに多数採用されているコードレス入力方式を比較検討する。コードレス入力方式は別名をDe－Skilled（熟練を要しない）Type、ダイレクト・インプット、ワンタッチ方式、タッチイン方式、項目入力方式、あるいは熟語入力方式などと呼ばれており、いずれにせよ、入力者がコード化して入力する必要がなく、直接漢字かなまじりの日本語情報を記述したキーをタッチインする方式である。

以下に日本語端末用の入力方式をいくつかとりあげて比較検討する。

4.2　漢字仮名入力システムの概要

4.2.1　システムの目的

漢字仮名入力システムは、日本語端末のモデル開発にあたって、その入力方式を設定するために作成した模擬実験システムである。

日本語情報の入力方式は、前節4.1で述べたような各種の方式が研究開発されている。その中から日本語端末の入力方式として採用できそうなものを選定してみると、次の３つの方式が候補として挙げられる。

(1) コードレス（項目）入力方式

(2) カナ漢字（単語）変換方式

(3) カナ入力表示選択方式

これらはいずれも漢字キーボード（フルキー）を持たない簡易入力キーボードから入力する方法である。日本語端末の利用対象者が一般事務員や技術者であることを想定し、また利用方法がセンタあるいはネットワーク上のホスト・コンピュータに対して問合せすることが主体になること、などの理由から上記の3方式を原型にした入力方式が適用できる。

漢字仮名入力システムでは、コードレス入力方式を主方式にして、他の2つの方式（カナ漢字変換方式、カナ入力表示選択方式）のいずれか、あるいは2つの方式を合成した方式を副方式として模擬実験する計画であったが、主記憶の制約から実現した副方式は、カナ入力表示選択方式の機能を一部含んだカナ漢字変換方式になっている。なお変換辞書ファイルは3方式用を作成して後述の分析を行なった。

4.2.2 システムの構成

(1) ハードウエア／入出力機器構成

漢字仮名入力システムは、ＦＡＣＯＭ　Ｍ－１６０コンピュータ・システム（以下、M-160と呼ぶ）と、ＮＥＡＣシステム１００小型コンピュータ・システム（以下、Ｎ－１００と呼ぶ）の構成のもとに実行する。両コンピュータ・システムで使用する入出力機器は次のとおりである。

- Ｍ－１６０の入出力機器
  - カード・リーダ
  - ライン・プリンタ
  - 磁気テープ
  - 磁気ディスク
  - フロッピー・ディスク
- Ｎ－１００の入出力機器
  - カード・リーダ
  - シリアル・プリンタ
  - カートリッジ・ディスク
  - フロッピー・ディスク
  - ＣＲＴディスプレイ
  - キーボード（英数カナ・キーボード、キーマット、制御キー）

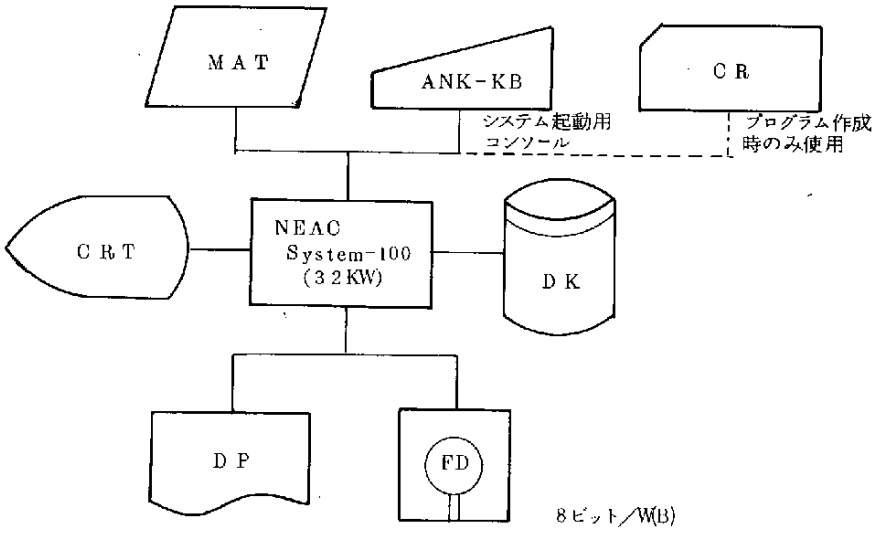


図４－１　ＮＥＡＣシステム１００構成

(2)　ソフトウエア構成

漢字仮名入力システムは、次の２つのサブシステムによって構成されている。

- 入力変換サブシステム
- 変換辞書作成サブシステム

入力変換サブシステム（以下、入力システムと呼ぶ）は、キーボードから入力される情報を、変換辞書を使って漢字仮名まじりの日本語情報（漢字コード列）に変換するシステムで、Ｎ－１００に構築されている。

変換辞書作成サブシステムは、入力システムで使用する変換辞書を作成し登録するシステムであり、Ｍ－１６０でフロッピーディスクに順編成ファイルを作成し、Ｎ－１００のユーティリティでカートリッジディスクに索引編成ファイルを作成する、

システム全体の処理フローを図４－２に示す。

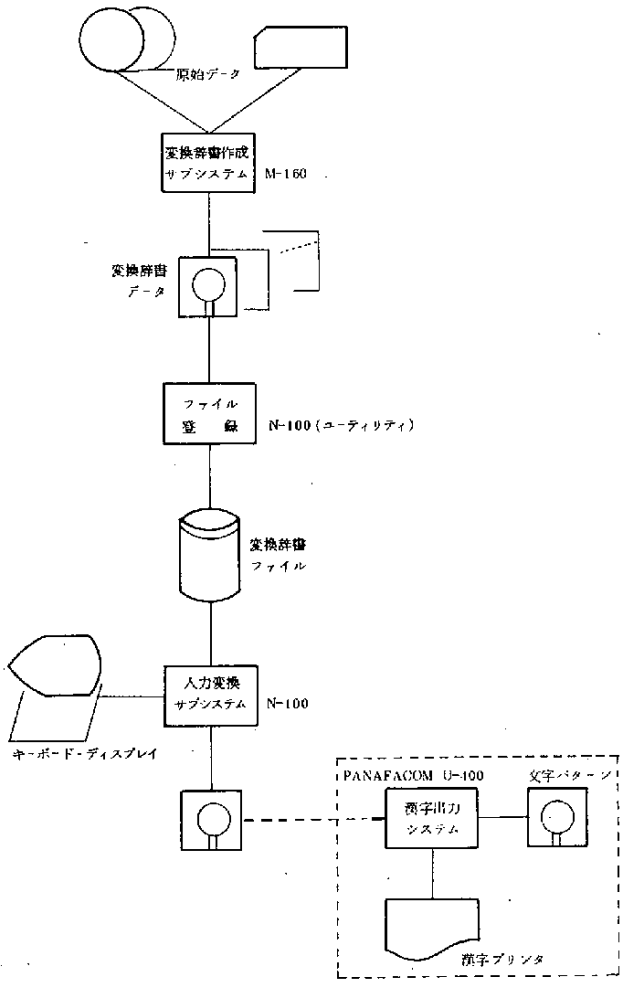


図４－２　システム・フロー

4.2.3　入力変換サブシステム

(1)　入力システムの概要

入力システムは、ブック型のキーマットを入力キーボードにして、容易に漢字仮名まじりの日本語情報を作成することができる。

キーマットには、最大９ページ1,４４０項目に、漢語や複合語が割当てられ、最終１ページ１６０項目に英数カナ記号と機能キーが割当てられている。システムは、キーの押下によって発生するマット情報から該当する変換辞書をアクセスして、漢字コード情報を得、フロッピー・ディスクにデータ出力する。

主とする入力方式は、コードレス入力方式（以下では項目入力方式と呼ぶ）であり、キーマットの索引ページを用いて該当項目（漢語・複合語）をワンタッチ入力できる。もう一つの方式は、

単語ファンクション・キーとともに漢字や単語の読仮名を入力するカナ漢字変換方式を採用している。

当システムは、N－１００で実動するが、出力・表示装置に漢字はいっさい出力できない。そのため漢字の識別は漢字コード（ＪＩＳ区点４桁／字）で行なわなければならない。

(2) 入力システムの機能

① 入力情報の識別

キーマットの各キーから入力される情報は、いずれも６桁の数値情報（マット情報）であり、これを解析してそれぞれに対応する変換処理を行なう。

a）キーマットの割当

・キーマット形態……ブック型

・キー数：{ページ左……１０行　８列＝８０キー
　　ページ右……１０行　８列＝８０キー
　　計１６０キー／ページ

・ページ数……１０ページ

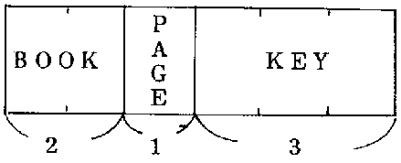
０～８（９ページ）……項目入力割当

項目数：１６０キー／ページ×９ページ＝1,４４０項目

９（１ページ）……仮名英数字記号および制御情報、ファンクション情報の入力割当

キー数：１６０キー

・マット情報



数値６桁（６バイト）

ＢＯＯＫ：ブック・コード……６３固定

ＰＡＧＥ：ページ・コード……０～９

０……第１ページ（最初のページを開いた状態）

１……　２
２……　３　｝（右から左に１枚づつ開ける）
｛

９……第１０ページ（最後のページを開いた状態）

ＫＥＹ　：キー・コード……０００～１５９

上位２桁：列番（００～１５）

下位１桁：行番（　０～　９）

行＼列	0	1	2	3	4	5	6	7
0	000	010	020	030	040	050	060	070
1	001							
2	002							
3	003							
4	004							
5	005							
6	006							
7	007							
8	008							
9	009	019	029	039	049	059	069	079

(左)

8	9	10	11	12	13	14	15
080	090	100	110	120	130	140	150
							151
							152
							153
							154
							155
							156
							157
							158
089	099	109	119	129	139	149	159

(右)

図４－３　キーマットのＫＥＹコード

マット情報の範囲……６３０００００～６３９１５９

(注) マット情報は項目キー１６０種の他に、マット制御キー２種（ＩＶキー、ＡＬＰＨＡキー）が使われても数値が入ってくる。ＩＶキー（コード０００００００）は入力終了キーとして使うがＡＬＰＨＡキー（９９０００００）は無視する。

b）項目入力情報

１４４０項目キーには、漢語・複合語が対応している。

割当方法は、漢語・複合語の読仮名順（昇順）になっており、マット情報の昇順に対応させてある。

当システムの実験用には、情報処理関連用語（検索キーワード）１４３２語が配置されている。

(例)	マット情報	用　語
第1ページ	６３０００００	アナログ通信
	６３０００１	アルゴリズム理論
	｜	
	６３０００９	安定判別
	｜	
	６３０１５９	記憶素子
	｜	
	６３８１５１	ＺＤ運動

c）仮名英数字記号情報

キーマット最終１ページの１６０キーのうち１２６キーにカタカナ、英字、数字、記号を

配置し、それぞれに対応したマット情報が入力される。

表４－１　仮名英数字記号の種類とマット情報

文　　字	種類	マット情報の範囲
カ タ カ ナ		
大　文　字	４６	６３９０８０～６３９１２８
		（６３９１１７、６３９１１９、６３９１２７を除く）
小　文　字	９	６３９１４４～６３９１４８
		６３９１５４～６３９１５７
ﾀﾞｸ点・半ﾀﾞｸ点	２	６３９１４２、６３９１５２
英　　字	２６	６３９０００～６３９００３
		６３９０１０～６３９０１３
		６３９０２０～６３９０２２
		｝
		６３９０７０～６３９０７２
数　　字	１０	６３９０３３、６３９０３４
		｝
		６３９０７３、６３９０７４
記　　号	３３	６３９００５～６３９００８
		｝
		６３９０２５～６３９０２８
		６３９０３５～６３９０３７
		｝
		６３９０７５～６３９０７７
		６３９１４０、６３９１４１、６３９１４３
		６３９１５０、６３９１５１、６３９１５３

d）制御情報／ファンクション情報

表示画面の制御と出力制御情報および入力ファンクション情報、計１１種２０キーが割当られている。

表４－２　制御／ファンクション・キー

キー	マット情報	
ＩＮＩＴ	６３９０３８	入力データ表示画面および入力データ出力バッファをクリア
Ｄ Ｓ Ｋ	６３９０４８	入力データ出力バッファをフロッピー・ディスクに出力
ＢＡＣＫ	６３９０１９	入力データ（表示画面、出力バッファ）を１字あるいは１モード単位にキャンセル後退
Ｎ Ｅ Ｌ	６３９０５８	復帰改行コードの挿入
ＦＦ	６３９０６８	改ページ（書式送り）コードの挿入
Ｋ Ｓ Ｐ	６３９０７８	スペース（漢字）コードの挿入
ＳＰ	６３９１３０ ～ ６３９１３９	表示画面にスペース挿入
カタカナ	６３９０３９	カタカナ・ファンクション・キー
外　字	６３９０４９	外字ファンクション・キー
小 文 字	６３９０５９	英小文字ファンクション・キー
Ｋ Ｋ Ｃ	６３９０６９	単語ファンクション・キー

② 入力変換機能

キーマットに割当られた項目キー、仮名英数字記号キー、制御キー、ファンクション・キーの各キーから入力される情報を解析して、それぞれに必要な変換辞書ファイルをアクセスし、漢字コード情報に変換する。変換機能の種類と概要を以下に述べる。なお変換に必要な変換辞書を表４－３に示す。

表４－３　変換機能と辞書

変換機能	変換辞書
項目変換	項目変換辞書
カナ漢字変換	カナ漢字変換辞書
ひらがな変換	平仮名群変換辞書
英字変換	平仮名群変換辞書
数字変換	平仮名群変換辞書
記号変換	平仮名群変換辞書
カタカナ変換	片仮名変換辞書
外字変換	外字変換辞書

a) 項目変換

キーマットの項目キーの押下によって発生するマット情報6桁をアクセス・キーにして項目変換辞書を探索して、漢字コード列を得る。

b) カナ漢字変換

単語ファンクション・キーの押下後に入力される読仮名（最大12桁）をアクセス・キーにして、カナ漢字変換辞書を探索して漢字コード列を得る。なお読仮名は、マット情報からの変換（マット文字変換）を行なって求める。

c) カタカナ変換

カタカナ・ファンクション・キーの押下後に入力されるカタカナを、片仮名変換辞書によって漢字コードのカタカナに変換する。このときダク点・半ダク点を先読みして、漢字コードのダク音・半ダク音のカタカナにする。

d) ひらがな変換

英数かなモード（ファンクション・キーでモードが変化していない状態）で入力されるカタカナを、平仮名群変換辞書によって漢字コードのひらがなに変換する。ダク点・半ダク点の処理はc) と同様。

e) 英数記号変換

英数かなモードで入力される英字、数字、記号を、平仮名群変換辞書によって、漢字コードの英字、数字、記号に変換する。

f) 英字小文字変換

小文字ファンクション・キーの押下後に入力される英字を、平仮名群変換辞書によって、漢字コードの英字小文字に変換する。

g) 外字変換

外字ファンクション・キーの押下後に入力される数字4桁（ＪＩＳ漢字コード区点4桁）をアクセス・キーにして、外字変換辞書を探索し、2バイトの漢字コードを得る。なお数字4桁は、マット情報からの変換（マット数字変換）を行なって求める。

h) 出力制御コード変換

入力システムで作成する漢字データ中に、漢字出力システムでプリントアウトする際の制御情報を挿入する機能である。

出力制御情報として、復帰改行、改ページ（書式送り）、スペース（漢字間隔）を指示することができる。

キーマットの最終ページに割当られた、ＮＥＬ、ＦＦ、ＫＳＰの各キーが対応し、パラメータによって複数個の情報を一度に入力できる。

システムのコード変換機能は、各キーを使って入力されるマット情報を識別して、対応する制御コードをパラメータで指定された個数分出力バッファに転送する。パラメータもマット数字変換を行なって数値を得る。

キー	マット情報	制御コード
		（16進）
NEL	639058	1C40
F F	639068	0C
KSP	639078	1C20

パラメータの最大値は64に制限し、これをこえると障害を起すことがある。

③ 画面表示機能

CRTディスプレイ装置の表示文字数は、40字／行×16行＝640字で、文字種はカタカナ（大文字、小文字）、英字、数字、記号、スペースの計127種である。

表示画面の割当を図4－4に示す。

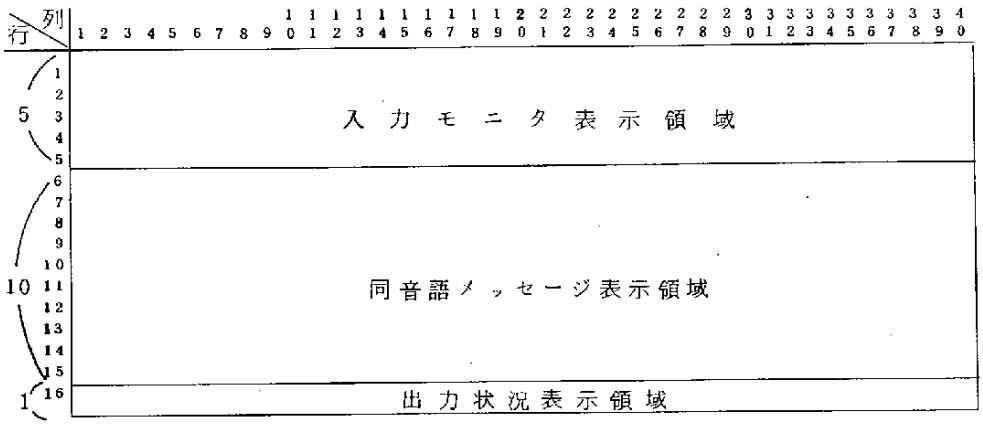


図4－4　画面割当

a）入力モニタ表示領域には、入力した文字情報や制御情報を表示するが、文字種の制約から表示マークをつけて識別できるようにする。

対応表を表4－4に示す。

表4－4　入力情報の表示法

入力文字	表示マーク	表　示　例	原　文
項　目	〔……〕	〔ジョウホウケンサク〕	情報検索
漢字単語	＜……＞	＜ケンキュウ＞	研究
カタカナ	＊	＊コ＊ン＊ピ＊ュ＊ー＊タ	コンピュータ
ひらがな	な　し	カナモジ	かなもじ
数　字	な　し	123	123
英　字			
大文字	な　し	ABC	ABC
小文字	・	・A・B・C	abc
記　号	な　し	¥－	¥－
外　字	@……@	@1601@	亜（区点コード(1601)）

制御文字	表示マーク	表示例	制御内容
ＮＥＬ	／……／	／ＮＥＬ／	復帰改行（１行）
Ｆ　Ｆ	／……／	／ＦＦ／	改ページ（１ページ）
ＫＳＰ	／……／	／ＫＳＰ２／	スペース（２桁）

b）同音語メッセージ表示領域には、カナ漢字変換によって漢字・単語を入力する際に発生する同音異義語（同音異字）を表示する。読仮名の入力に対して複数の漢字・単語が対応する場合、それらをすべて表示して、利用者からの選択指示を受けたうえで一意に変換する。同音語の漢字表示は、ＪＩＳ漢字コードの区点４桁の数字で代用する。

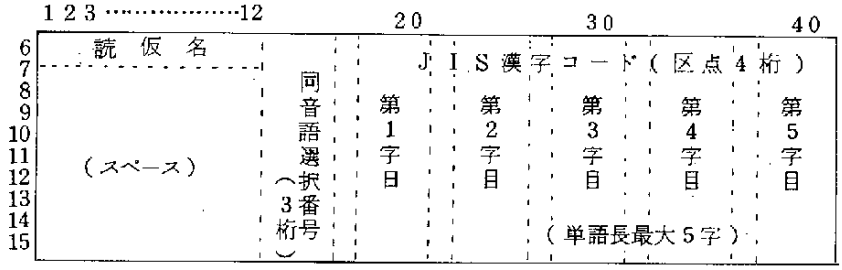


図４－５　同音異義語の表示画面

同音語選択番号（３桁以内）の入力は、キーマットの数字キーで行なわれ、これをマット数字変換によって１０進数値を得、同音語識別する。

選択された同音語のモニタ表示は、漢字単語の表示マーク内に挿入される。

ex)　＜ジョウホウ**1**＞
　　　└── 同音語選択番号

c）出力状況表示領域には、フロッピー・ディスクに出力する漢字データが、出力バッファ内に残り何文字分入力できるかを逐次表示する。

フロッピー・ディスク・ファイルのレコード・サイズは１２８バイトであり、２バイト／字の漢字データが最大６４字入る。

16行目　ＲＥＭＡＩＮ　ＣＨＡＲＡＣＴＥＲ　nn

（１≦nn≦６４）

d）なお入力モニタ表示領域と出力状況表示領域の内容は、画面制御の指示をうけて変化する。

ＩＮＩＴキーは、入力バッファと出力バッファをクリアすると同時に、入力モニタ表示領域を全面消去する。

ＤＳＫキーは、出力バッファの内容をフロッピーディスクに出力すると同時に、入力モニタ表示領域を全面消去し、出力状況表示領域には、出力バッファの移項後に残った入力可能文字数にセットしなおす。（移項がないときは６４）

ＢＡＣＫキーは、入力の誤りを修正するために１単位戻る機能である。この場合の１単位とは、項目、単語、外字、出力制御情報では１モード分、カタカナ、ひらがな、英数字記号では１字分を意味する。よって出力バッファを１単位バックすると同時に出力状況表示領域を１単位の文字数分減算させ、入力モニタ表示領域を１単位バックして消去する。

④ 漢字データ出力機能

入力（キーイン）―表示・選択―変換という処理を経て漢字データを得ることができたら、これをフロッピー・ディスクに出力する。

出力バッファ領域は、２バイト／字を１２８字分確保してあり、６４字分が記憶された時点でフロッピーディスクに自動出力する。あるいはまたＤＳＫキーが入力されると６４字分に満たなくとも出力する。

１２８字分の出力バッファは、項目キーが６４字目に入力されたり、出力制御情報（ＮＥＬ、ＦＦ、ＫＳＰ）のパラメータが入力されたときに対処するために余分にある。

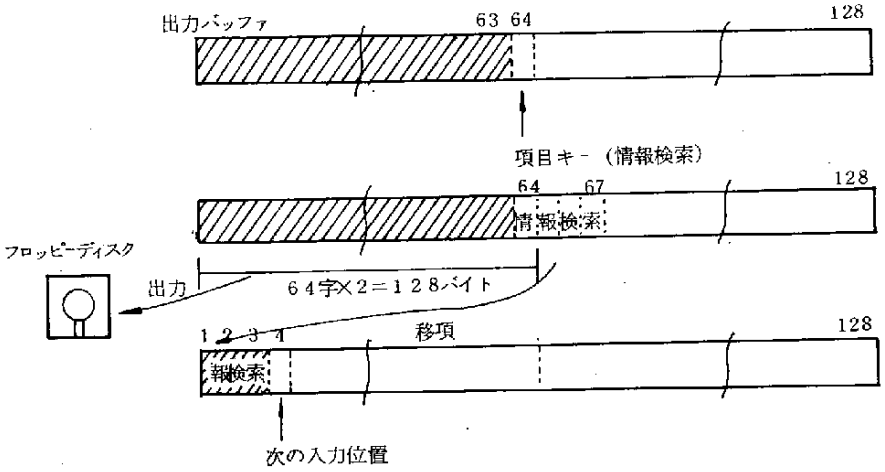


図４－６　漢字データ出力

また、フロッピー・ディスクへの出力（自動出力またはＤＳＫキー指示出力）と同時に、入力データのモニタをＣＲＴ表示バッファ（入力モニタ表示領域）から、ＤＰ（高速シリアル・プリンタ）へハード・コピーをとる。

出力情報は表示されていた５行と同一である。

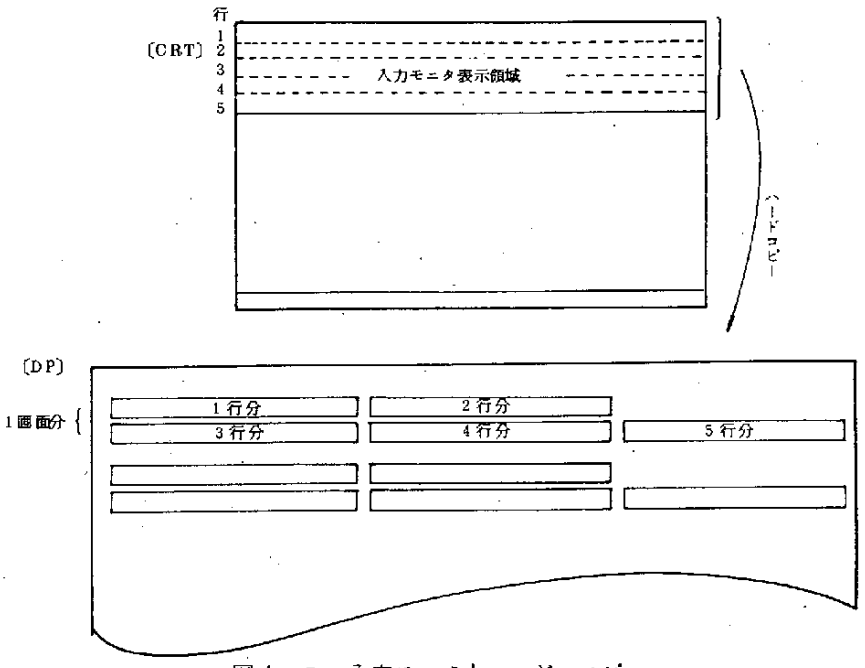


図4－7　入力モニタとハード・コピー

4.2.4　変換辞書作成サブシステム

(1)　辞書作成システムの概要

入力変換サブシステムで使用する変換辞書ファイルを作成するために、次のような6つのプログラム群で構成されている。

- ＪＩＳコード変換ファイル作成プログラム
- 項目変換辞書作成プログラム
- カナ漢字変換辞書作成プログラム
- カナ入力変換辞書作成プログラム
- 仮名英数字記号変換辞書作成プログラム
- 外字変換辞書作成プログラム

いずれもＭ－１６０のＦＯＲＴＲＡＮ　Ｍで作成したプログラムである。これらのプログラムで最終ファイルとしてフロッピー・ディスクに創成したあとは、Ｎ－１００のユーティリティ・プログラムで、ファイル変換（順編成→索引編成）し、カートリッジ磁気ディスクに登録する。

なお、項目変換辞書およびカナ漢字変換辞書の原始データは、昭和５０－５１年度の日本語情報文献検索システムで作成したキーワード・ファイルおよびカナ漢字変換テーブルから抽出したものである。

(2)　辞書作成システムの処理概要

a)　ＪＩＳコード変換ファイル作成プログラム

ＦＡＣＯＭ漢字コード（Ｆコード）から、ＪＩＳ漢字コード（Ｊコード）に変換するために

作成したコード変換データ・カードのチェックを行なって、最終ファイルとしてＦＪコード変換ファイルを作成する。

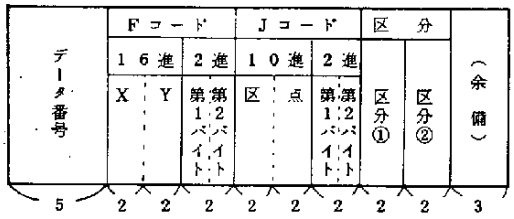
Ｆコード文字種：５３７６

Ｊコード文字種：６８０２

データ件数：約８０００件

カード枚数：約１６００枚

○ＦＪコード変換ファイル形式



レコードサイズ：２４バイト

・データ番号：コード変換データ番号・５桁

・Ｆコード：ＦＡＣＯＭ漢字コード

１６進コード ― Ｘ・２桁、Ｙ・２桁

２進コード ― Ｘバイナリ１桁、Ｙバイナリ１桁

・Ｊコード：ＪＩＳ漢字コード

１０進コード ― 区・２桁、点・２桁

２進コード ― 区バイナリ１桁、点バイナリ１桁

・区　分：漢字区分

区分① ― 第１区分
区分② ― 第２区分
）付記２　漢字区分表参照

・余　備：オール・ゼロ

b）項目変換辞書作成プログラム

日本語情報文献検索システム（ＫＩＮＤＳ……昭和５０、５１年度開発）で使用されたキーワード６４７５語の中から、情報処理に関連するカテゴリのキーワードを抽出して各種統計リストをとり、さらにしぼって項目変換辞書の最大容量１４４０語以内を抽出して作成する。

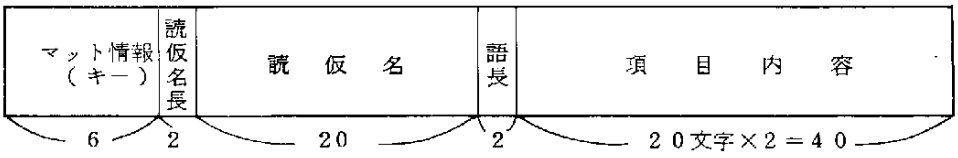
ＫＩＮＤＳのキーワード・ファイルでは、ＦＡＣＯＭ漢字コードが使用されていたので、これをＪＩＳ漢字コードに変換する。

ＫＩＮＤＳキーワード数　：　６４７５語

一次抽出語数　：　２５１１語

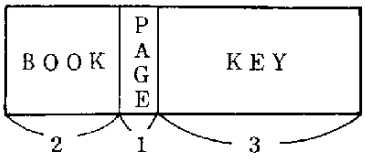
最終辞書登録語数　：　１４３２語

○項目変換辞書ファイル形式（索引編成）



レコード・サイズ：７０バイト

• マット情報：キーマットの項目キーに対応した数値６桁



• 読仮名長：読仮名の文字数（１～２０）

• 読 仮 名：項目内容（漢語・複合語）の読仮名

• 語　　長：項目の文字数（１～２０）

• 項目内容：漢語・複合語（最大２０字×2_B＝40_B）

c） カナ漢字変換辞書作成プログラム

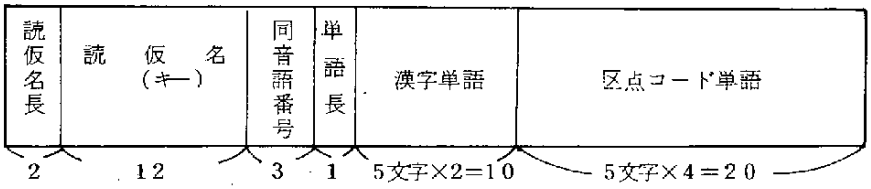
日本語情報文献検索システムおよびカナ漢字変換システム（昭和４６～４９年度開発）で使用したカナ漢字変換テーブルの中から、情報処理関連用語２７７９語を抽出し、ＦＡＣＯＭ漢字コードからＪＩＳ漢字コードへ変換して、カナ漢字変換辞書ファイルを作成する。

全 用 語 数 ： 約５０,０００語

抽 出 語 数 ： 2,７７９語

辞書登録語数 ： 2,７７９語

○カナ漢字変換辞書ファイル形式（索引編成）



レコード・サイズ：４８バイト

読 仮 名 長 ： 読仮名の文字数（１～１２）

読　仮　名 ： 漢字単語の読仮名

同 音 語 番 号 ： 同音異義語識別番号

０００ ― 同音異義語対応なし

001
〈 } 同音異義語対応シーケンス番号
999

単　語　長　：　漢字単語の文字数（１～５）

漢字単語　：　漢字コードの単語（最大５字×2_B＝10_B）

区点コード単語　：　ＪＩＳ漢字コード区点４桁表現した単語

（最大５字×４＝20_B）

d）カナ入力変換辞書作成プログラム

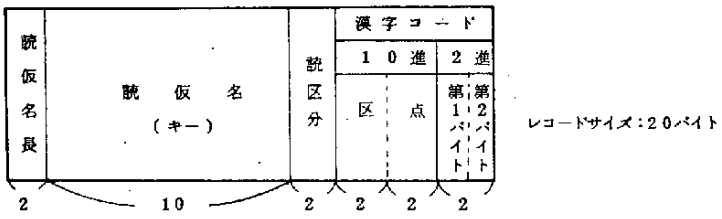
ＪＩＳ漢字コード表の第１水準漢字集合に入っている２９６５字について、漢和辞典（角川漢和中辞典）から音訓読を調べてデータ作成し、このデータをもとに各種統計リストをとったうえで、最終的にカナ入力変換辞書ファイルを作成する。

漢　字　数　：　２９６５字

読仮名総数　：　７６４０

（カードの枚数：２９９６枚）

○カナ入力変換辞書ファイル



・読仮名長　：　読仮名の文字数

・読　仮　名　：　漢字の読仮名

・読　区　分　：　読仮名の分類コード

００：当用漢字音訓表にある音読

０１：　〃　　訓読

１０：当用漢字音訓表にない音読あるいは表外漢字の音読

１１：当用漢字音訓表にない訓読あるいは表外漢字の訓読

・漢字コード　：　ＪＩＳ漢字コード { 区点各２桁 / ２進各１桁

e）仮名英数字記号変換辞書作成プログラム

仮名英数字記号の内部コード（ＥＢＣＤＩＣ）とＪＩＳ漢字コード（区点コード）に、アク

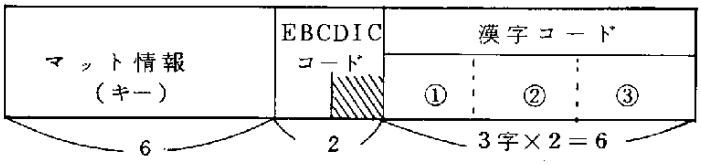
セス情報（キーマットのマット情報）を対応づけて作成したデータから、変換辞書ファイルを作成する。一部カードせん孔機にないコードは１６進数を対応させて内部変換する。

平仮名群……ひらがな、英字、数字、記号

片 仮 名……カタカナ

○平仮名群変換辞書ファイル形式（索引編成）

○片仮名変換辞書ファイル形式（索引編成）



レコード・サイズ：１４バイト

マ ッ ト 情 報 ： キーマットから発生する６桁の数値

ＥＢＣＤＩＣコード ： 文字コード（ＥＢＣＤＩＣ）

カタカナ（大・小文字）・英大文字・数字・記号

漢 字 コ ー ド ： ＥＢＣＤＩＣに対応する漢字コード

平仮名群：①英大文字、数字、記号、ひらがな

②英小文字、ひらがなダク音

③ひらがな半ダク音

片 仮 名：①カタカナ、一部記号

②カタカナ・ダク音

③カタカナ・半ダク音

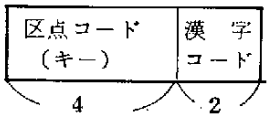
f）外字変換辞書作成プログラム

ＪＩＳ漢字コードを、１０進４桁（区点コード）で対応づけたレコードを内部で発生して、外字変換辞書ファイルを作成する。

文字種 ： ＪＩＳ漢字コード（ＪＩＳ外記号１）

第１水準	２９６５字
第２水準	３３８４字
仮名英数記号	４５４字種
計	６８０３字種

○外字変換辞書ファイル形式（索引編成）



区点コード：ＪＩＳ漢字コード区点４桁／字

漢字コード： 〃 （バイナリ）２桁／字

4.3 変換辞書の分析

4.3.1 項目変換辞書

項目変換辞書への登録語決定までに2段階のデータ抽出を行なった。第1段階では、６４７５語の原データ（キーワード・ファイル）から情報処理に関連するカテゴリに属するキーワード２５１１語を抽出して統計分析を行なっている。第2段階では、さらに登録条件を加えて１４３２語を最終登録語として決定し、これについての統計分析も行なった。

(1) 分析1

① データ件数……２５１１語

② 語　　長

読仮名長……平均8.49　　最大３３

漢字語長……平均4.95　　最大２４

③ 出現文字種

a) 読仮名項目

出現文字総数……２１３２４

文　字　種……　　７９

使 用 頻 度……平均２６９.９２

最大２０７４（ヽ）

b) 漢字項目

出現文字総数……１２４３９

文　字　種……　　９２５

使 用 頻 度……平均１３.４５

最大　３６２（一）

・文字種内訳

漢字　出現文字総数……７５０７（６０.３５％）

文　字　種……　７６６

使 用 頻 度……平均9.80

最大１８２（理）

その他　出現文字総数……４９３２（３９.６５％）

文　字　種……　１５９

使 用 頻 度……平均３１.０２

最大３６２（一）

・文字比率

	読仮名	漢　字
文字総数	（２１３２４）	（１２４３９）
	1.714	: 1
漢字数	（１６３９２）	（７５０７）
	2.18	: 1

④ 同音語

同音語総数……１２（0.48％）

同音語種類……6

同音語対応……（平均／最大）2

⑤ 語構成

漢字が１字も出現しない語数……４２８

漢字語長１～３の語数……５８４

漢字語長２１以上の語数……1

読仮名長２１以上の語数……7

(注)　これに該当する語を削除したのが分析２のデータ件数である。

(2) 分析２

① データ件数……１４３２語

② 語　長

読仮名長……平均9.56　最大１９

漢字語長……平均5.04　最大１４

③ 出現文字種

a) 読仮名項目

出現文字総数……１３６９３

文字種……72

使用頻度……平均１９０.１８

最大１３２６（ヽ）

b) 漢字項目

出現文字総数……７２１４

文字種……７７０

使用頻度……平均9.38

最大１６８（埋）

・文字種内訳

漢字　出現文字総数……５８０４

文字種……６４４

使 用 頻 度……平均9.0 1

最大1 6 8（理）

その他　出現文字総数……1 4 1 0

文　字　種……　1 2 6

使 用 頻 度……平均1 1.1 9

最大9 5（－）

• 文字比率

	読仮名	漢　字
文字総数	（1 3 6 9 3）	（7 2 1 4）
	1.8 9 8	： 1
漢 字 数	（1 2 2 8 3）	（5 8 0 4）
	2.1 1 6	： 1

④ 同 音 語

同音語総数……4（0.2 8％）

同音語種類……2

同音語対応……（平均／最大）2

表４－５　語長分布表（項目変換辞書）

語長	分析１		分析２	
	読仮名	漢字	読仮名	漢字
1	0	4		
2	23	302		
3	111	278		
4	140	854	1	767
5	155	341	5	292
6	202	237	45	162
7	317	159	191	96
8	353	90	264	47
9	366	100	305	40
10	285	49	228	13
11	204	36	157	7
12	112	24	84	6
13	84	15	55	1
14	55	5	40	1
15	44	7	28	0
16	19	5	15	0
17	13	2	5	0
18	8	1	6	0
19	10	0	3	0
20	3	0	0	0
21	1	1		
22	2	0		
23	0	0		
24	1	1		
25	0	0		
26	0	0		
27	0	0		
28	1	0		
29	0	0		
30	0	0		
31	0	0		
32	1	0		
33	1	0		
34～40	0	0		

4.3.2 カナ漢字変換辞書

カナ漢字変換テーブルに収録してある約5万語のデータの中から、情報処理関連用語２７７９語を抽出して統計分析を行なった。

① データ件数……２７７９語

② 語　　長

読仮名長……平均4.65　　最大１１

漢字語長……平均2.17　　最大　4

③ 出現文字種

a）読仮名項目

出現文字総数……１２６８３

文　字　種……　　４７

使 用 頻 度……平均２６９.８５

最大１３７０　（ウ）

b）漢字項目

出現文字総数……６０４０

文　字　種……１０２３

使 用 頻 度……平均5.90

最大７０　（機）

・文字種内訳

漢　字　出現文字総数……５９８８

文　字　種……　９９７

使 用 頻 度……平均6.01

最大７０　（機）

その他　出現文字総数……５２

文　字　種……２６

使 用 頻 度……平均2

最大８　（り）

・文字比率

	読仮名	漢　字
文字総数	（１２６８３）	（６０４０）
	2.10	: 1
漢字数	（１２６３１）	（５９８８）
	2.11	: 1

④ 同 音 語

同音語総数……４００（１４.３％）

同音語種類……１７４

同音語対応……平均2.３０

最大６

表４－６　語長分布表
（カナ漢字変換辞書）

語長	読仮名	漢字
1	３０	２１１
2	１４３	１８９５
3	４３７	６５３
4	８１２	２０
5	６９２	０
6	４０７	
7	１８０	
8	６０	
9	１５	
１０	２	
１１	１	
１２	０	

4.3.3　カナ入力変換辞書

ＪＩＳ漢字コード第１水準漢字集合２９６５字種について音訓読データを作成して、統計分析を行なった。対象データ区分として漢字区分と読区分で分類し、５種の分析を得た。

(1)　分　析　１

①　対象データ区分（付記２、漢字区分表・読区分表参照）

漢字区分……１０～１９

読 区 分……全区分（００、０１、１０、１１）

②　文 字 種……２９６５

③　読仮名数

音訓読総数……７６４０

読仮名対応数……平均2.５８

最大８　（下、生、納）

④　読仮名長（送り仮名接続、以下同様）

平均文字長……2.５７

文字長分布……

文字長	度数	文字長	度数
1	８４２	5	３００
2	３４３６	6	７８
3	２０５２	7	１１
4	９２１	8～10	０

⑤　索　　引

索引文字長	索 引 数	平均度数	最大度数（文字）
1	４４	１７３.６４	８１５（ シ ）
2	９６３	7.９３	２１５（ ジ ）
3	１９３６	3.９５	１２９（コウ）
4	２１１９	3.６１	１２９（コウ）
１０	２１２９	3.５９	１２９（コウ）

(2)　分　析　２

①　対象データ区分

漢字区分……１０～１８

読 区 分……全区分（００、０１、１０、１１）

② 文 字 種……２２４２

③ 読仮名数

音訓読総数……５９６６

読仮名対応数……平均 2.6 6

④ 読仮名長

平均文字長……2.5 7

文字長分布……

文字長	度数	文字長	度数
1	６７１	5	２３３
2	２６６７	6	５１
3	１６０４	7	6
4	７３４	8～10	0

⑤ 索　引

索引文字長	索引数	平均度数	最大度数（文字）
1	４４	１３５.５９	６５４（ シ ）
2	８８５	6.7４	１７５（ ジ ）
3	１６９４	3.5２	９７（コウ）
4	１８２５	3.2７	９７（コウ）
１０	１８３５	3.2５	９７（コウ）

(3) 分 析 3

① 対象データ区分

漢字区分……１０～１８

読 区 分……当用漢字 ― ００（音読）、０１（訓読）

表外漢字 ― １０（音読）、１１（訓読）

② 文 字 種……２２４２

③ 読仮名数

音訓読総数……５２９６

読仮名対応数……平均2.3６

④ 読仮名長

平均文字長……2.５４

文字長分布……

文字長	度数	文字数	度数
1	６０７	5	２０３
2	２４０４	6	４５
3	１３９８	7	6
4	６３３	8～10	0

⑤ 索　引

索引文字長	索引数	平均度数	最大度数（文字）
1	44	120.36	573（シ）
2	866	6.12	154（ジ）
3	1620	3.24	89（コウ）
4	1732	3.06	89（コウ）
10	1741	3.04	89（コウ）

(4) 分析 4

① 対象データ区分

漢字区分……10、11、15（当用漢字）

読区分……全区分（00、01、10、11）

② 文字種……1850

③ 読仮名数

音訓読総数……5042

読仮名対応数……平均2.73

④ 読仮名長

平均文字長……2.58

文字長分布……

文字長	度数	文字長	度数
1	541	5	203
2	2228	6	45
3	1378	7	6
4	643	8～10	0

⑤ 索　引

索引文字長	索引数	平均度数	最大度数（文字）
1	44	114.59	555（シ）
2	836	6.03	151（ジ）
3	1532	3.29	80（コウ、ショウ）
4	1646	3.06	80（コウ、ショウ）
10	1656	3.04	80（コウ、ショウ）

(5) 分析 5

① 対象データ区分

漢字区分……10、11、15

読区分……00、01

② 文字種……1850

③ 読仮名数

音訓読総数……３４９４

読仮名対応数……平均1.89

④ 読仮名長

平均文字長……2.49

文字長分布……

文字長	度数	文字長	度数
1	414	5	111
2	1640	6	22
3	918	7	3
4	386	8～10	0

⑤ 索　引

索引文字長	索引数	平均度数	最大度数（文字）
1	44	79.41	386（シ）
2	762	4.59	105（ジ）
3	1276	2.74	64（コウ）
4	1345	2.60	64（コウ）
10	1351	2.59	64（コウ）

4.4 システムの評価検討

日本語端末の入力方式を設定するために作成した漢字仮名入力システムをとおして得られた評価から、入力方式の比較検討を述べることにする。

漢字仮名入力システムに導入した入力方式は、キーマットを使ってのコードレス（項目）入力方式と、キーマットの最終１ページにカタカナを五十音順に、英字をＡＢＣ順に配列したキーボードを使って入力するカナ漢字変換方式、および外字の入力に数字キーを使った数字コード入力方式の３方式である。また実験に用いた小型コンピュータのメモリの制約などがあって、変換辞書は作成したものの導入できなかったカナ入力表示選択方式についても検討を加える。

(1) 項目入力方式の評価

実験に使用した項目入力方式用のキーボードは、ブック型のキーマットで、最大６３個のブックが交換可能になっている。このうちの１つのブックを使用して、１０行×１６列の１６０キーからなるページが１０ページ、計１６００項目に用語や文字記号を割当て実験を行なった。

１０ページのうち最初の９ページ１４４０項目に１４３２語のキーワード（漢語・複合語）を割当、最終１ページには、カナ英数字記号と制御キー、ファンクション・キーを割当た。この１４３２語の入力が項目入力方式に該当するものである。キーワードの割当は、システム作成者が機械的に並べただけのものであり、情報処理関連用語が読仮名順に項目キーに配列されれている。

入力操作手順は、入力する原文に漢語や複合語が出現したら、ブックのページに付られた索引

をみて該当ページを開く。次にそのページ内の項目に該当する語があるかどうか目視探索する。見つかった時にはその項目キーを押下するだけでその語の入力を完了する。もしなかった場合には、項目入力を断念して他の方法で入力しなければならない。ここまでの操作から気づくことは、各ページの１６０項目を目視探索することが容易でないことである。各項目キーの上方に漢字仮名まじり文で記されているので読解はできるが、キーの間隔がせまく、そのため記述している文字が小さくて素早く読みとることは容易でない。キーの間隔をせまくしてキーマットを小型にすることは、目の動作範囲を小さくする効果をもたらすが、文字が小さくなって読みにくい欠点が生じる。これはキーマットの評価といえるが、日本語端末のキーボードを考えた場合、キー数を減らして文字を読みやすくすることが必要である。

また各項目キーにどんな語が割当てあるのかを前もって知っておかないと、せっかくページを開いて探してみても無かったという無駄な操作が増加し、入力速度が遅くなる原因になる。これは項目キーへの語の割当を利用者が設定した方が効果的であるという指針である。システム的には、利用者からの項目割当機能や更新機能などが要求されることになる。

入力操作に要する動作は、ページを開ける動作が大きいことの欠点と、目視動作が多いという欠点を有するが、キーのタッチ数は最も少ない方式であり、ページの開閉を自動化したり、項目数の最適化をはかれば効果は大である。項目入力方式の特徴は、語や文の長さにかかわらず一様なスピードと少ないタッチ数で入力できることであり、これを最大限に生かすことが課題である。

一方、利用者側からは間接的になる変換辞書は、項目キーが発信するコードをアクセス・キーにしてダイレクト・アクセスできるようにファイル編成しなければならない。そのため外部記憶装置としては乱処理可能なディスク類が要求される。変換辞書の規模は、項目数に一致するが、割当る語や文の長さによって変化する。一般には漢語・複合語であれば、変換辞書の分析からわかるように平均５字、漢字仮名まじり語や文でも２０字を用意すれば十分と思われる。日本語の文書での区切りを読点にすると、およそ１５～３０字程度になる。このような文字長の変化は、結局はアプリケーションに依存する部分であり、日本語端末の適用アプリケーションで決定する必要がある。

なお、項目入力方式の変換処理は、ファイルのダイレクト・アクセス処理だけの簡単なアルゴリズムで可能である。

(2) カナ漢字変換方式の評価

入力方式の実験では、キーマットの最終１ページに割当たカナ英数字記号およびファンクション・キーを使ってカナ漢字変換処理を行なった。カナ漢字変換方式は、漢字の読仮名を入力して、これをアクセス・キーにしてカナ漢字変換辞書をアクセスし、漢字（コード）を得ることである。そのため漢字キーボードは不要で、カナ英数キーボードで十分である。カナ英数キーボードは、約５０キーを４段シフトして操作するのが一般的であるが、そのキー配置は、ＡＢＣ順でもなく五十音順にもなっていない、タイプライタ型である。英文タイピストやキーパンチャにとっては、疲労の少ない理想的な配置なのかも知れないが、一般の素人にとっては目視することからはじめ

るために、ＡＢＣ順や五十音順の方が入力速度は早くなる。また情報処理にたずさわる人であれば英数記号はタイプライタ型になれており、カナを五十音順にするだけで良い。

入力の操作手順は、一般の文書を原文としてそのデータを入力しようとすると、文書中に出現する文字構成に注意しながら、漢字・漢語の部分について先頭に漢字用ファンクション・キーを付したあと、読仮名をカタカナ・キーから入力する。漢字・漢語は平均2字の語であり、その読仮名は漢字１字について平均２字である。よって漢語の入力は平均４字のカタカナで入力できる。カタカナ（読仮名）で入力する方式の最大の難点は、この時発生する同音異義語（同音異字を含む）の問題である。一般の国語辞書のように汎用的な変換辞書を作ろうとすると、辞書の規模が大きくなればなるほどに同音異義語の発生が増加する。発生率は３０％以上にもなってしまう。これに対処するための工夫の１つは、汎用辞書を分割して専用分野・アプリケーション用の辞書にすることであり、同音異義語の発生率は１０％前後になる。

入力操作は、同音異義語の発生と同時に入力を中断し、同音異義語の選択操作を行なう。

(3) カナ入力表示選択方式の評価

カナ入力表示選択方式では、入力のための機器構成として、カナ・キーボードと漢字表示機器が必須となるため、漢字仮名入力システムで実験評価することは困難である。現実には、前述したようにこの方式を機能として採用しなかったが、カナ漢字変換方式における同音異義語の表示選択機能と同様の機能を各文字（漢字）の入力ごとに行なうわけである。

入力キーボードの配置などの考察も、カナ入力を基本とするカナ漢字変換方式と同様であり、付加すべき事項は、表示画面に対する操作法と、表示情報に対する選択情報の入力法を容易にすることである。そこにはカナ入力表示選択方式の入力情報としてどんな内容を指示できるかも影響する。これは変換辞書の構成についてもいえる。

カナ入力表示選択方式の入力操作は、基本的には各漢字の読仮名を入力し、その読仮名に対応する同音異字の漢字を表示させて、その中から該当文字を１字選択して入力する。この場合、読仮名の入力方法を、先頭文字１字を入力して表示選択するか、あるいは２字まで入力するかで機能は変化し、また同音異字の表示後に追加入力情報の機能として２字目、３字目の入力あるいは、他の読み方（音と訓のちがいや、他の同音・訓の読み方）を追加情報として入力し識別するか、または部首構成要素の名称を（１字）入力したり、その他のイメージ情報を入力したりして、同音異字の識別に対処する方法がとられている。これらの方法をとるには、そのための辞書を作らなければ対処できない。

漢字仮名入力システム用に作成した変換辞書は、読仮名対応辞書であるが、索引編成の構成を１字から２、３、４字および全文字（１０字）でとって、追加情報の入力に対処している。入力順序としてはまず最初に読仮名の第１字目を入力する。この１字の読仮名に対応する漢字（当用漢字１８５０字の場合　、平均８０種、最大３８６種）が表示される。この中から１字を探して選択入力するが、対応文字種が多いときは時間がかかる。そこで対応文字種を少なくしようと読仮名の第２字目を入力すると、平均4.6種、最大１０５種に減少する。この時点で大半の漢字を

限定することができるが、さらに範囲を狭くしようとすると、第３字目、第４字目と入力し、最大１０文字目（実際には７字目までが最長）まで可能である。

読仮名の一意性は、第１字目の入力では０、つまり、１字の読仮名を入力しただけでは、漢字の一意性が全くないことを意味する。第２字目まで入力すると、約３８％が一意に決定し、残６２％には平均6.７８個が対応し、第３字目まで入力すると約６６％が一意に決まり、残３４％に平均6.１１個が対応している。そして最大長７字目までを使って入力すると約７０％が一意に決まり、残３０％に平均6.３５個が対応している。

このような数字をみても明らかなように、各漢字を入力するのに、多くのタッチ数を要するかあるいは多くの表示文字の中から１字を探すという目視探索が要求されるため、時間はかかるし、各漢字ごとに同様なことが要求されるわずらわしさがある。

この方式の長所とする点は、文字種分は確実に入力できることである。しかも、音訓いずれの読みからでも索引できるために素人にとって操作方法は簡単である。

(4) 数字コード入力方式の評価

項目入力方式、カナ漢字変換方式（カナ入力表示選択方式）をベースにした入力方式でシステムを作成した場合、ある特異な漢字や記号を入力しようとすると、そのための変換辞書がないと入力できない。

統計分析の結果からみても、変換辞書に登録されている用語を記述している文字種は、延べ千数百種のオーダである。すると他の文字が文書に出現したら入力しようにもできないことになる。この点、フルキーボード方式などでは、キーボードに配置されている数千字の文字は確実に入力できるわけで、実際に評価されてデータ作成業務に多く使用されている。

これに対して前記３方式の目的はフルキーボードを使用しない簡易入力方式用を検討しているわけであり、また文字種も極力少な目におさえる方向にもある。よってあまり使用しない文字種の入力は、コードブックを参照しながらでもよいという判断のもとに、この数字コード入力方式を採用した。

数字コード入力方式の特徴は、テン・キーからの入力ができることであり、キー・タッチ・スピードが最高速である。そのため、漢字に対応した数字コードを覚えてしまえば、入力速度は他の入力方式をはるかにしのぐことになる。ただし、漢字の種類が多いために覚えるのが大変で、しかも漢字と数字コードの関連がないためにコード化にも留意しなければならない。

一般に、多く用いられている数字コードは、新字源に付られた４桁の数字が代表的であるが、ＪＩＳ漢字コードの制定によって、区点４桁のコードが対応しやすくなっている。

当システムの実験では区点４桁の数字コードから内部で変換辞書を用いてバイナリ２桁に変換する方法をとっている。変換辞書を用いているのは、入力変換システムのプログラム言語でバイナリ値が求められないためであり、他の言語（アセンブラなど）ならば区点コードからバイナリ値への変換は演算で可能である。

区点コード４桁の入力確認のためには、漢字表示機器がないと不可能であるからして、この方

式の評価も未確定である。

なお、変換辞書はＪＩＳ第１水準漢字集合２９６５字種分もっている。ＪＩＳ第１水準漢字集合の配列は、代表音訓読の五十音順であり、コード・ブックの索引も容易になっており、漢和字典方式よりも字種が少ないこともあって、探しやすい。

(5) 全体的評価

以上の方式から入力の操作性を比較検討してみる。比較項目は、主に入力タッチ数、入力速度、操作訓練、正確性などであり、相対的な評価を加えて日本語端末利用者のための入力方法を考察する。

ただし、漢字仮名入力システムで使用したキーボードおよびディスプレイで不十分な点は、若干改良したものとして話を進める。

まず入力操作に要するタッチ数は、漢字１字を入力するのに、項目入力方式の場合、索引に１タッチ項目に１タッチの計２タッチ、カナ漢字変換方式は、読仮名２タッチ同音語選択に１タッチの計３タッチ、カナ入力表示選択方式は、読仮名１タッチ選択に１タッチの計２タッチ、数字コード入力方式は、数字４タッチ、という結果になる。これを２字、３字、４字、８字についてまとめると表４－７のようになる。

表４－７　入力タッチ数

	１字	２字	３字	４字	８字
項目入力方式	2	2	2	2	2
カナ漢字変換方式	3	4.2	6.0 5	8.0 0 1	1 6
カナ入力表示選択方式	2	4	6	8	1 6
数字コード入力方式	4	8	1 2	1 6	3 2

なおカナ入力表示選択方式では、読仮名１字の入力に対応する漢字が１画面に表示しきれないときには、さらに継続表示を要求する操作が１タッチ必要になる。またカナ漢字変換方式では同音異義語の発生率をそれぞれ１００％、２０％、５％、0.1％、０％として加算してある。

次に入力操作速度を相対的に比較してみると表４－８のようになる。この場合の値は、項目入力方式の索引に１、項目の探索とタッチに２を割当、カナ漢字変換方式の読仮名１字につき１、同音異義語選択操作に１を、カナ入力表示選択方式の読仮名に１、漢字の探索と選択操作に２を、数字コード入力方式の数字に0.5を、それぞれ割当て比較している。

表４－８　入力速度比

	１字	２字	３字	４字	８字
項目入力方式	3	3	3	3	3
カナ漢字変換方式	3	4.2	6.0 5	8.0 0 1	1 6
カナ入力表示選択方式	3	6	9	1 2	2 4
数字コード入力方式	2	4	6	8	1 6

実際の入力速度は、これに原文の判読時間や目の移動時間、表示装置の表示時間などが加わることになろう。

これらの入力方式の操作を利用者が覚えるのにどれくらいの訓練を要するであろうか。いずれの方式もキーボードの操作という点では訓練は不要と判断してみると、項目入力方式では、項目の割当順を知っておくと入力速度が上昇し、カナ漢字変換方式とカナ入力表示選択方式は、漢字の読仮名に対する読解力によって入力速度が変化する。これは学校教育からの国語力の差が生じることになろう。また数字コード入力方式は別格で、漢字とコードの対応を暗誦することは長期の訓練といえども満足されないかも知れない。よってコード・ブックの参照ということになるから、入力方式の主流をなすものでないことは明らかである。

この他に、カナ漢字変換方式とカナ入力表示選択方式では、機能キーによる識別が必要とされることになろうから、タッチ数、速度がいく分変化する。またこの2方式ともカナ入力という共通点と、同音異字あるいは同音異義語の表示選択ということに類似点があり、入力方式の分類も同一にすることができる。これはカナ漢字変換方式を逐次処理の会話型にしているためであり、分ち書きのない自動変換と分類を異質にしている。こうしてみると、カナ漢字変換方式にカナ入力表示選択方式を包含して、カナ入力方式とし、前方一致・完全一致の読仮名から変換処理を行なうことができる。ただし、漢字1字単位の入力は、操作もめんどうであるし、入力速度も劣ることから、主体となるのは単語としての読仮名入力が適当である。

日本語端末からの入力の特徴は、日本語情報といえども短いメッセージ、パラメータ、キーワードに類するものであるから、適用アプリケーションによって項目入力キーボードを可変的に扱える方法を主体に考案したほうが効果的である。また日本語端末を出力端末的に扱うような場合は、入力は英数カナキーボードからの情報で十分になるであろうから、漢字コードに変換することもなくなって、従来のキーボードから英数カナ・コードを入力できればよいことになる。

5　日本語端末の基本設計

5.1　日本語端末の設計方針

センタ・コンピュータあるいはネットワーク上のホスト・コンピュータとオンラインで日本語情報処理を行なうことを目的にした、日本語情報の入出力処理を行なう端末装置、これを日本語端末と定義して呼称する。

日本語端末の構築目的は、現状の日本語情報処理技術の中で、日本語情報を入出力する装置（以下漢字入出力装置）の技術的課題が、入力の操作性向上と出力装置の低価格化にあること、そしてコンピュータの利用分野の拡大にともなって、利用者へのサービスとしてオンライン化傾向が強まっていることなどを総合的にとらえた結果として、オンライン日本語端末の開発に意義が高いと判断したためである。

ここで目指す日本語端末は、これらのニーズに応えるべく、入力操作性の向上として入力の簡易化をはかり、端末のコストを低下させるために、ハード／ソフト両面から機能検討を行なって、その可能性を究明することが目標である。入力の簡易化は利用者のレベルを一般事務員や技術者といったいわゆる素人（漢字入力の非専門家）に的をしぼって、入力速度よりもいかに簡単な操作で行なうことができるかを検討することにある。また端末の低価格化として、マイクロプロセッサによる制御機能の実用性を検討するとともに、日本語端末の中心的機能となる文字発生器の低廉化を試みて、文字パターン・メモリの削減にパターン圧縮技術を、記憶装置にRAMとフロッピーディスクを用いてその適用性を検討する。

5.2　日本語情報処理システムの構成

日本語情報をオンライン形態で処理するためのコンピュータおよび端末のシステム構成は、図5－1に示すようなオンライン日本語情報処理システムとして構築される。センタ・コンピュータ（センタあるいはネットワーク上のホスト・コンピュータを総称する）には、日本語情報データベースをはじめとして、変換辞書データベース、文字パターン・データベースなどが存在し、日本語端末ではこれらのデータベースを利用するアプリケーション・システムの端末装置として使用される。そのため、日本語端末の基本構成要素として、日本語情報を出力できる表示装置あるいは出力装置（プリンタ）が必須条件になり、これに入力装置、制御装置が接続構成する。

日本語端末の構成方法は、最小構成としての入出力機器をタイプライタ型で構成することも可能であり、またインテリジェント・ターミナルとして、プロセッサを端末に内蔵し、センタ・コンピュータと端末間の機能分担を最適にすることができる。

端末のインテリジェンス性は、入力－送信…受信－文字発生－表示・出力の各機能を定型的に（ハードウェア化）しないで、適用アプリケーションに依存する部分などをソフトウェア化して、動的な処理にも対処できるようにする。

一般にインテリジェント端末は、情報システムの効率化のために利用されるものであり、そのねらいの中にも、

- 自由なアプリケーションの拡張追加
- データ処理機能の分散
- 端末側での処理の分担

などがあげられている。これに最近のハードウェアの進歩でプロセッサが安価になり、端末の機能向上が容易になってきた。また情報処理システムの分散化傾向、システムの多様化に対する端末システムの拡張性、隔通性が要求されており、日本語情報処理にもその適用現象が見られている。むしろ、入出力処理の煩雑な日本語情報処理では、周辺装置・端末装置の機能向上と拡張性などが要求され、マイクロ・プロセッサやミニコンピュータを組込んで対処するのが必然的傾向である。

なお、日本語情報処理システムの構成図においてセンタ・コンピュータと日本語端末のソフトウェア構成、機能分担は、およそ次のようである。

- センタコンピュータ側
  - 日本語情報データベース管理システム
  - 変換辞書データベース管理システム
  - 文字パターン・データベース管理システム
  - 各種アプリケーション・システム
- 端末側
  - 入力変換モジュール
  - 文字パターン発生モジュール
  - 表示・出力編集モジュール
  - 変換辞書管理／文字パターン管理

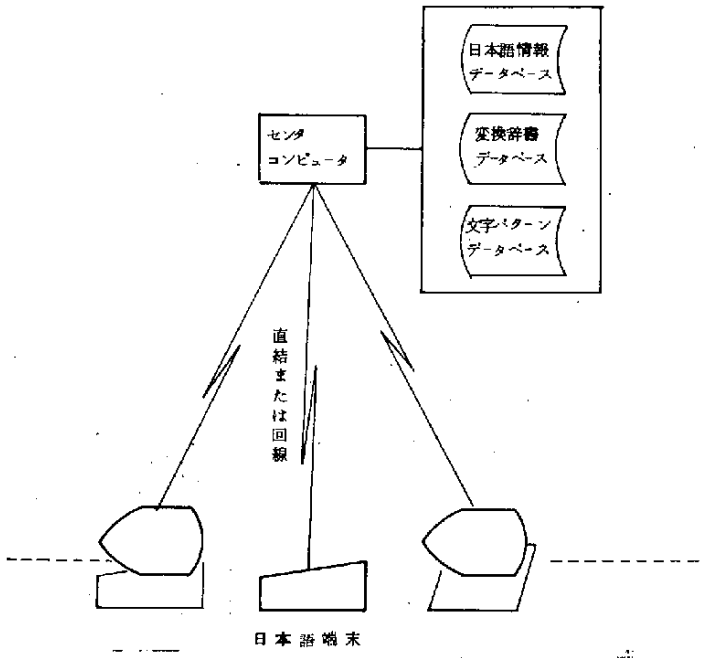


図５－１　日本語情報処理システムの構成図

5.3 日本語端末の構成

日本語端末の構成は、入力キーボードと印字機構を備えたタイプライタ型と、印字機構の代わりに表示装置を接続したディスプレイ型とに大別される。ここでは、インテリジェント・ディスプレイ・ターミナル型の日本語端末を構成し、次のような基本構成と付加構成をとってモデルを設計する。

5.3.1 基本構成

- 端末制御装置 （ＴＣ）
- 文字発生器 （ＣＧ）
- 入力キーボード （ＫＢ）
- ディスプレイ・モニタ （ＴＶ）

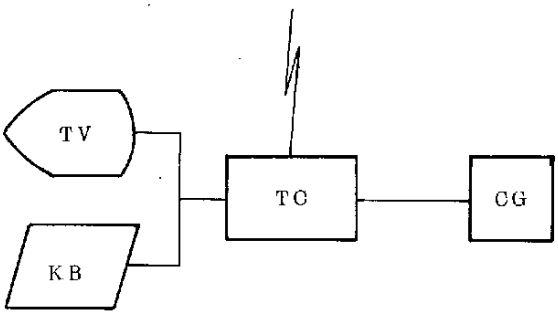


図５－２ 基本構成図

5.3.2 付加構成

- 基本構成機器
  - 端末制御装置
  - 文字発生器
  - 入力キーボード
  - ディスプレイ・モニタ
- 補助記憶装置 （ＤＫ）
- ハード・プリンタ （ＨＰ）

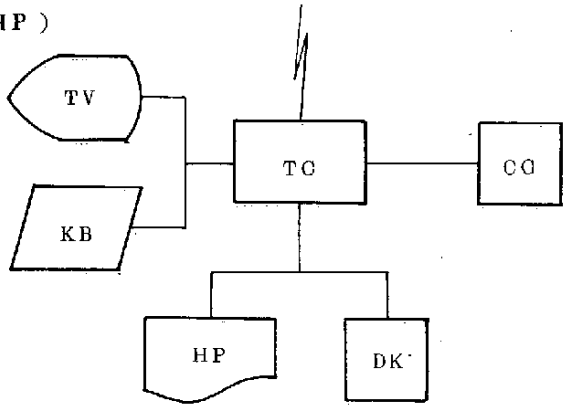


図５－３ 付加構成図

5.3.3 構成機器の概要

(1) 端末制御装置

マイクロプロセッサ

主記憶：６４ＫＢ

(2) 文字発生器

文字パターン記憶装置

主記憶：１６ＫＢ

（高頻度文字パターン記憶用）

補助記憶（フロッピーディスクを想定）：１４４ＫＢ

（低頻度文字パターン記憶用、索引テーブル、圧縮トリー・テーブル、他）

文字発生方式：ドット・マトリックス方式

文字パターン圧縮・復元方式：ハフマン符号化法

ドット・マトリックス構成：縦２４×横２４ドット

文字種：２４００字（最大３６６６字）

(3) 入力キーボード

文字キーボード：①ＪＩＳ型キーボード

文字種 { カタカナ / 英字 / 数字 / 記号 }

②簡易型順配列キーボード

文字種 { ひらがな / カタカナ / 英字 / 数字 / 記号 }

項目入力キーボード：

索引キー：１０／２０キー

項目キー：１００／５０キー

項目数：{ １０×１００ / ２０×５０ } ＝１０００項目

（項目面が索引によって作動する可変型）

制御機能キーボード：

{ 制御キー / 機能キー } ２０キー

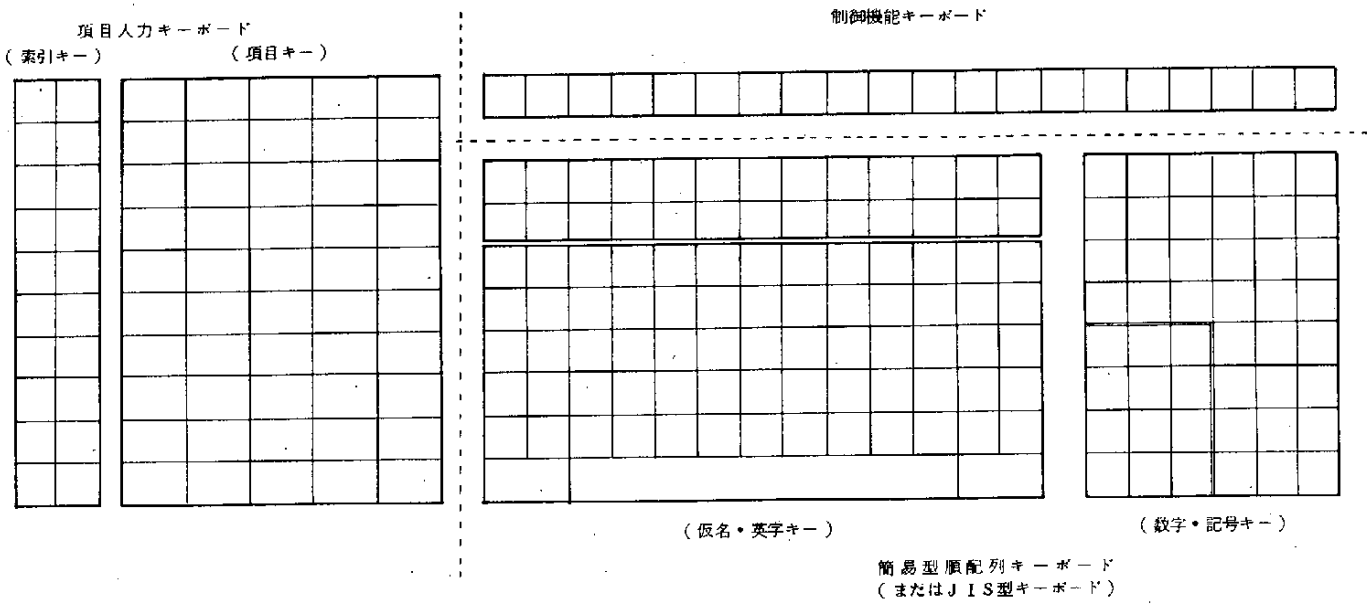


図５－４　入力キーボードの配置図

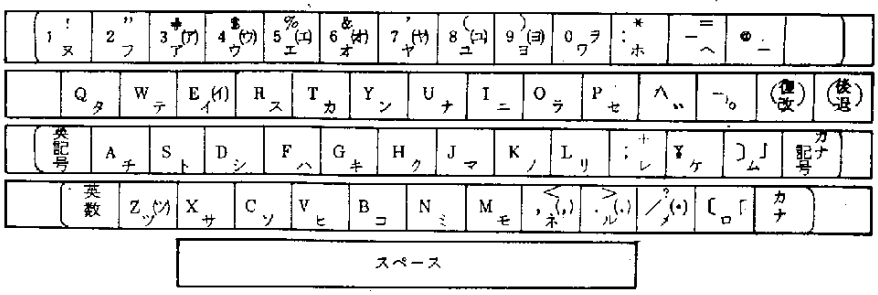


図５－５　ＪＩＳ型キーボード

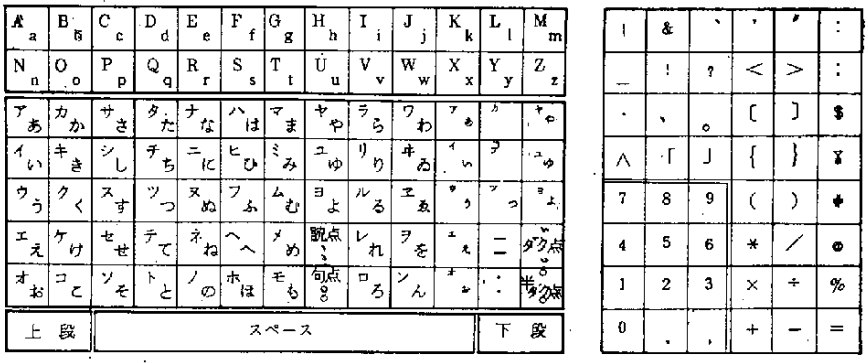


図５－６　簡易型順配列キーボード

索引キー　　項目キー

1	11
2	12
3	13
4	14
5	15
6	16
7	17
8	18
9	19
10	20

1	11	21	31	41
2	12	22	32	42
3	13	23	33	43
4	14	24	34	44
5	15	25	35	45
6	16	26	36	46
7	17	27	37	47
8	18	28	38	48
9	19	29	39	49
10	20	30	40	50

図５－７　項目入力キーボード（２０／５０型）

制御機能キーの種類：

問合せ

送信

消去…… １字後退削除

無効…… 入力データのキャンセル

画面消去

継続…… 画面の継続表示

後退…… 前画面の表示

復改

ハード・コピー

データ出力

モード設定

カナ漢字変換

外字入力

カタカナ入力…… 基本構成でひらがなとの識別に使用

(4) ディスプレイ・モニタ

表示方式：パターン・リフレッシュ方式

画面分解能：縦５１２×横５１２

パターン・リフレッシュ・メモリ：５１２×５１２＝２６２１４４ビット（３２ＫＢ）

表示文字構成：２４×２４ドット・マトリックス

表示文字数：２０字／行×１２行＝２４０字

文字間隔：０

行間隔：８ドット

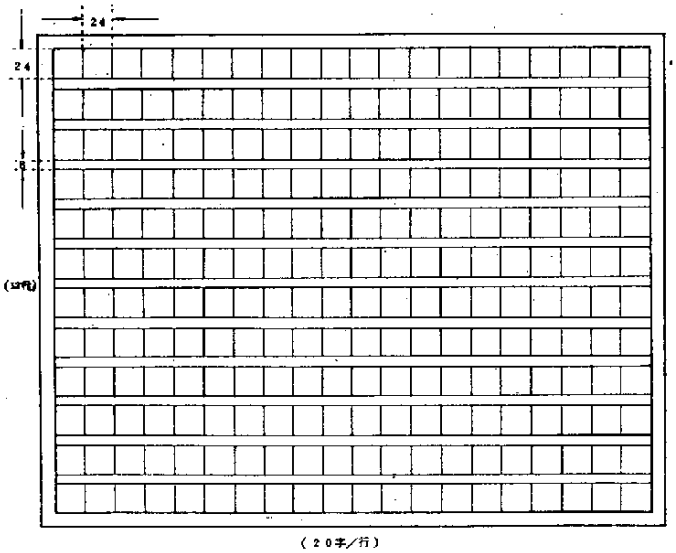


図５－８　表示画面構成

(5) 補助記憶装置

変換辞書ファイル記憶用

ランダム・アクセス・メモリ（フロッピーディスクを想定）

記憶容量：２５６ＫＢ

項目変換辞書：４２Ｂ／項目×１０００＝４２ＫＢ

カナ漢字変換辞書：２８Ｂ／語×５０００＝１４０ＫＢ

その他変換辞書：約４３ＫＢ

データ出力ファイル記憶用

記憶容量：１００ＫＢ

２Ｂ／字×５００００字＝１００ＫＢ

(6) ハード・プリンタ

出力方式：ドット・マトリックス方式

分解度：４本／㎜

印字数：４０字／行

印字文字構成：２４×２４ドット・マトリックス

文字間隔：０

行間隔：８ドット

5.4 日本語端末の特徴

日本語端末モデルの特徴を以下に列挙する。

(1) 入力の操作性を向上させるために、コードレス入力方式とカナ漢字変換方式を組合せた混合方式を採用している。

(2) 入力キーボードに、変更可能な項目入力キーボードと、五十音順・ＡＢＣ順の配列をとった簡易型順配列キーボードを使うことによって、素人でも簡単に操作ができる。

(3) 文字発生方式にドット・マトリックス方式を採用し、文字パターンの記憶容量と、文字発生速度に最適化をはかって低価格の文字発生器を構築している。

(4) 文字パターンの記憶に、ハフマン符号化法による圧縮技術を採用し、約４５％のメモリを節減している。（圧縮率５５％）。

(5) 日本語情報の表示機器としてモニタ・テレビを漢字ディスプレイ化し、入力の確認と、応答情報の理解が容易にできる。

(6) 表示・出力する文字パターンは、２４×２４ドット構成の疑似明朝体文字であり、読解がスムーズである。

(7) 入力変換辞書ファイルと文字パターン・ファイルの交換を容易にしてユーザ・アプリケーションに対処できる。

(8) 制御機能キーに、会話用コマンドや特殊制御機能を割当てることができ、操作が容易である。

5.5 日本語端末の処理概要

日本語端末の処理機能は大別すると次の3つの主要な機能に分類される。

- 入力制御機能
- 文字発生機能
- 出力制御機能

図5－9に処理ブロックの関連図を示す。

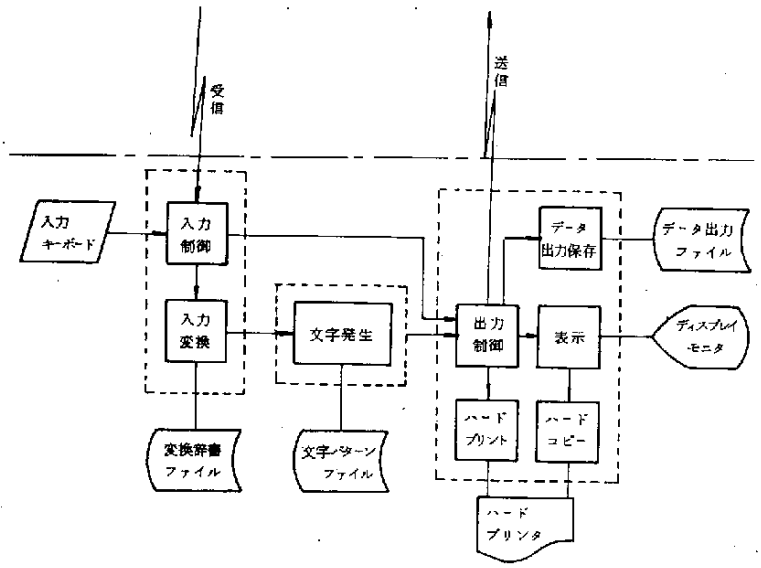


図5－9　日本語端末の処理ブロック関連図

5.5.1 入力制御機能

入力制御機能は、入力キーボードから発信されるコードの識別機能と、センタ・コンピュータから送られてくるコードの識別機能、および文字コード、項目コードから文字発生用コード（漢字コード）への入力変換機能を実行する。ただし入力変換機能は、日本語端末の構成が付加構成をとって補助記憶装置に変換辞書が登録されているときに限って動作する機能が大半を占める。

また入力制御機能の実行においては、日本語端末の動作開始時に、入力モードの設定を行なっておき、以降の入力はモード変更の指示があるまで同一モードで機能する。モードの設定および変更は、制御機能キーの1つに割当られた入力モード・キーとそのモード・パラメータを受けて行なう。

センタ・コンピュータから送られるデータについては、モード設定の制御コードを識別して、テキスト・データの内容を解析し、モード変更の指示があるまで（他のモードが設定されるまで）同一モードで機能する。

モードの種類は、漢字モードと文字モードに大別され、漢字モードは漢字コードの処理に、文字モードは文字コード（仮名英数字記号コード）と項目コードの処理に使用する。なお文字コードは

ＪＩＳ　Ｃ６２２０の8単位コードに準処したコードと、ＥＢＣＤＩＣコードに区分される。ＪＩＳに準処したコード（ＪＩＳ系文字コード）とは、カタカナの大小文字の他に、ひらがなの大小文字と一部記号を加えたコード体系である。ＥＢＣＤＩＣコードにはカタカナの小文字やひらがなは含まれない。また漢字コードはＪＩＳ　Ｃ６２２６の7単位2列コードを8単位2列にしたコード体系（ＪＩＳ漢字コード）を使用する。その他、制御コードはＪＩＳ　Ｃ６２２０とＪＩＳ漢字コード系の制御コード（現在審議中）を部分的に使用する。

入力および受信データの解析から入力変換機能までの処理フローを図５－１０に示す。

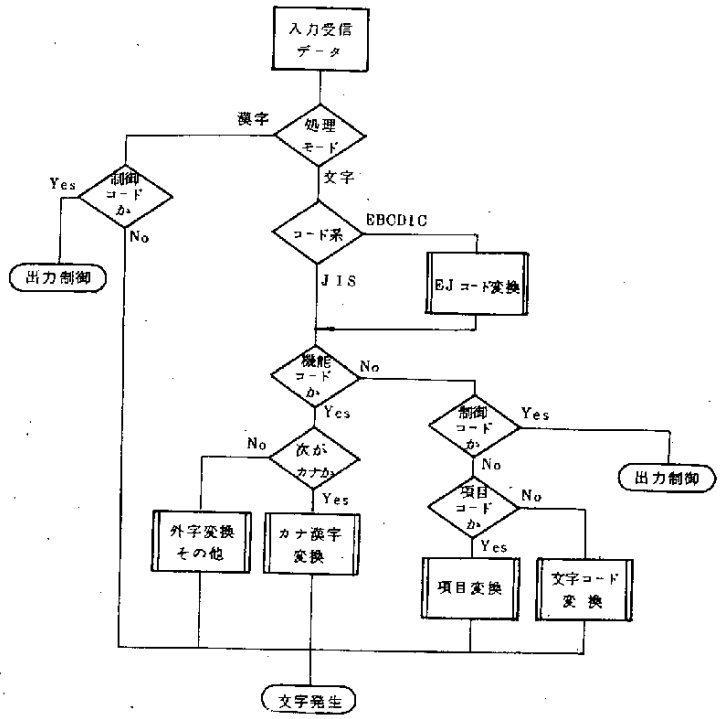


図５－１０　入力制御の解析フロー

以下に入力変換処理を機能ごとに概説する。

(1)　項目変換処理

項目変換処理は、漢語（漢字単語・複合語）や文が割当られた項目キーから発生する項目コードの2バイトをアクセス・キーにして、項目変換辞書をアクセスし、対応する漢字コード列を得る。項目コードは項目キーボードから3バイト・コードとして入力されるが、変換辞書をアクセスする際は、先頭1バイトを除去して、後2バイト（索引コード1バイト、キー・コード1バイト）を使用する。図５－１１に項目変換フローを示す。

(2)　カナ漢字変換処理

カナ漢字変換処理は、漢字単語の読仮名をキーにしてカナ漢字変換辞書をアクセスし、対応する漢字コード列を得る。この時、読仮名に対応する漢字単語が複数個あった場合、いわゆる同音異義語が発生するときには、文字発生・表示機能を作動して同音異義語をディスプレイに表示し、

選択番号の入力で一意に決定する処理を行なう。

図５－１２にカナ漢字変換フローを示す。

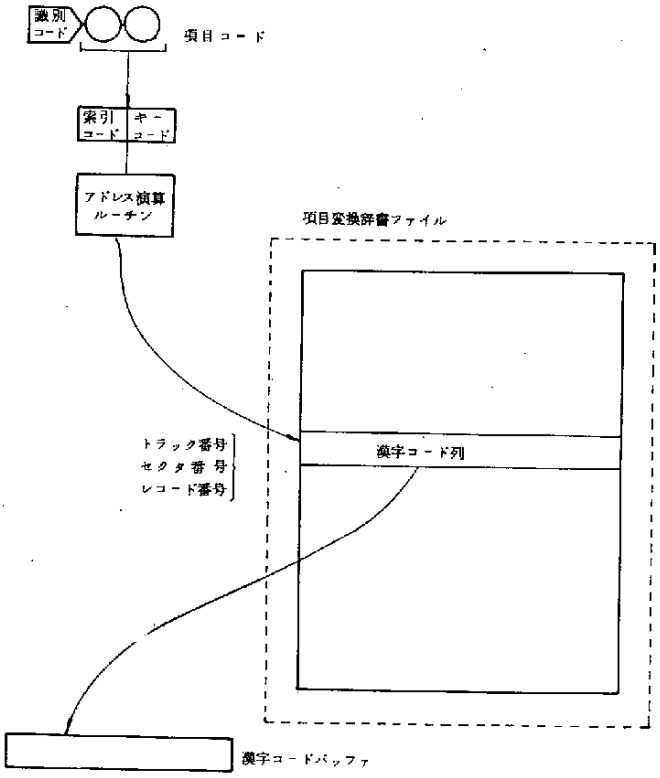


図５－１１　項目変換フロー

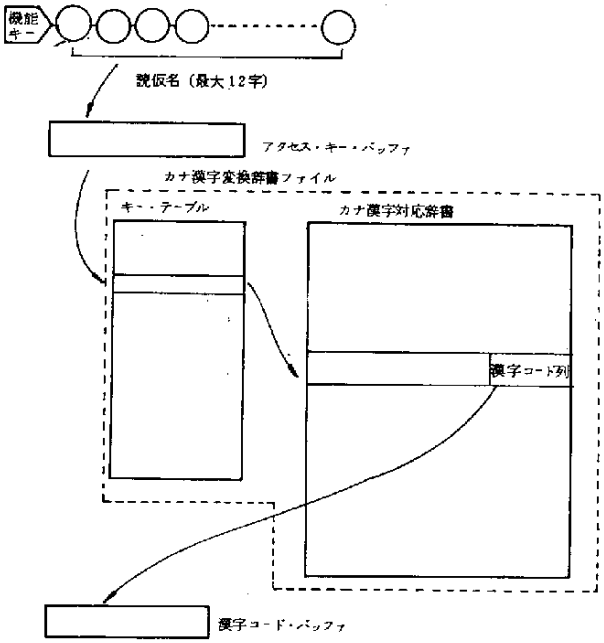


図５－１２　カナ漢字変換フロー

(3) 文字コード変換処理

文字コードの変換処理は、仮名英数字記号の1バイト・コードを、文字発生・表示させるために2バイト・コード（漢字コード）に変換する。このとき、1バイト・コードでは仮名（ひらがな、カタカナ）のダク点・半ダク点を1バイト・コードとして入力したものを、1文字前のコードと合成してダク音・半ダク音文字にする。図5－13に文字コード変換フローを示す。

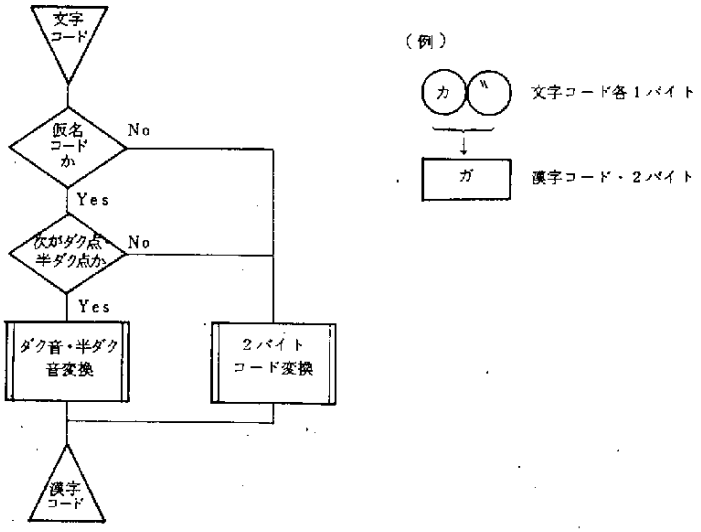


図5－13　文字コード変換フロー

(4) 外字変換処理

項目変換、カナ漢字変換、いずれからも入力できない漢字について、ＪＩＳ漢字コード表の区点コードから漢字コードへ変換する。入力される区点コードは数字4桁であり、これを演算によって2バイトの漢字コードに変換する。

なお、項目変換処理およびカナ漢字変換処理で使用する変換辞書構成については、5.7で概説する。

5.5.2 文字発生機能

文字発生機能は、入力制御機能から渡される漢字コードを受けて、そのコードに対応する文字パターンを文字パターン・ファイルからアクセスし、これをパターン復元エリアに復元して出力制御機能に渡すまでの処理を行なう。文字発生器の構成は図5－14に示すとうりである。各構成要素の内容は以下のようである。

① 圧縮文字パターン・ファイル …… ２４×２４ドットの文字パターンを圧縮して記憶したファイル。主記憶と補助記憶に割当られている。

② アドレス・テーブル …… 漢字コードと文字パターンの記憶アドレスを対応づけたテーブル。

③ トリー・テーブル …… 圧縮パターンを復元するために使用するテーブル。

④ サブパターン …… 圧縮パターンの原形となる2×2のドット・パターン（16種）。

⑤ パターン復元エリア …… 圧縮パターンを復元するための作業エリア。

文字発生機能は、圧縮文字パターン・フェッチ処理と文字パターン復元処理に大別され、以下にそれぞれの処理概要を述べる。

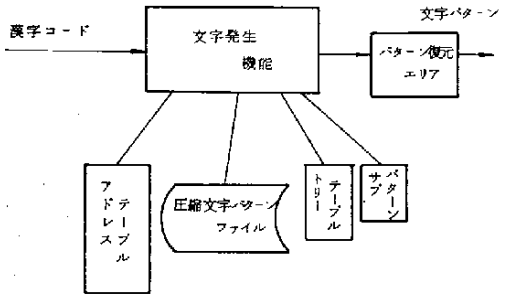


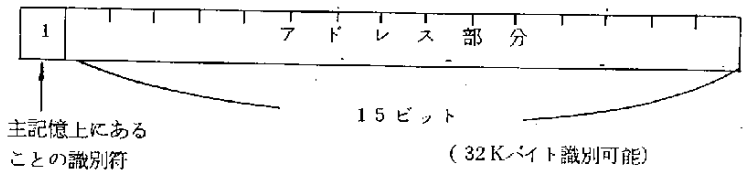
図5－14　文字発生器の構成

(1) 圧縮パターン・フェッチ処理

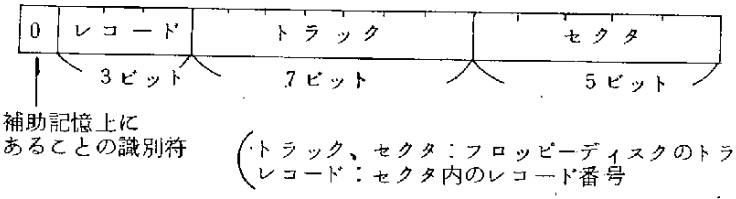
この処理はアドレス・テーブルと圧縮文字パターン・ファイルを使用して機能する。

アドレス・テーブルは、漢字コードに対応している。圧縮文字パターン・ファイルに記憶する文字種は第1水準の文字・記号に限定しているため、第1区から第7区、第16区から第47区までの39区、第1点から第94点までの94点、計3666区点のアドレス・テーブルを持つことにする。1文字に対するアドレスの表現方法は、圧縮パターンが主記憶上にあるか補助記憶（フロッピーディスク）にあるかによって異なる。

・主記憶用



・補助記憶用

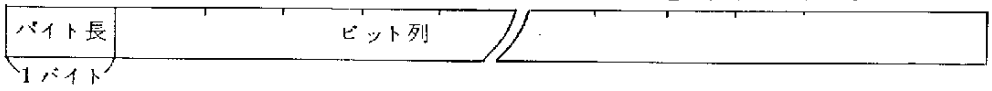


どちらも2バイトで表現する。そのため、アドレス・テーブルの大きさは

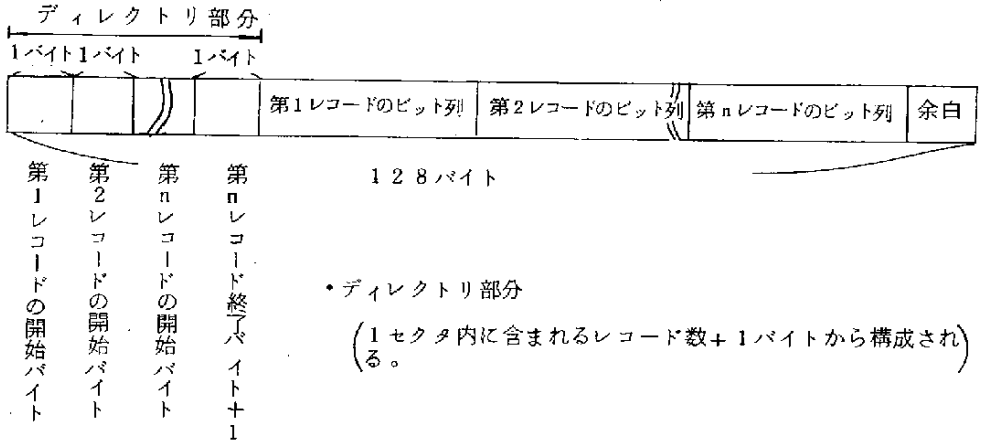
39×94×2＝7332バイトとなる。
区の数　点の数

JIS漢字コードにない文字や第1水準にない文字はコード表の空領域にコードを対応させて使用する。

圧縮文字パターン　・ファイルは使用頻度によって、主記憶内部に持つか、補助記憶内に持つかに分けてある。主記憶内部には高頻度文字約４００字種分を持ち文字発生速度を助けている。圧縮文字パターン・レコードは、バイト単位で表現され、その構造は以下のようである。



主記憶にない文字は補助記憶にあり１セクタ内に何文字分かが混在する。しかし、１文字の圧縮文字パターン・レコードが、２セクタ以上に渡って存在することはない。補助記憶に記憶されている圧縮文字パターン・レコードの構成は主記憶用とは異なり以下のようになる。



・ディレクトリ部分

（１セクタ内に含まれるレコード数＋１バイトから構成される。）

なお、圧縮文字パターン・レコードはバイト単位であり、残った部分には、０がうめられている。処理の流れは次のようである（図５－１５参照）。まず漢字コードを受けるとアドレス・テーブルをみにいく。そのコードに対応するアドレス指示部分の第１ビット目により主記憶または補助記憶の判別をし、第２ビット目から第１６ビット目までのアドレス部分によって圧縮文字パターンのビット列をアクセスしてくる。主記憶からのフェッチは圧縮文字パターン・レコードの1バイト目にレングスが入っているため、2バイト目からそのレングス（バイト単位）だけビット列用バッファに持ってくればよい。補助記憶の場合は、アドレス部のトラック番号、セクタ番号によって（フロッピーディスクから）１２８バイト単位で読み込み、レコード番号からその先頭にあるディレクトリ部分を参照（n レコードはnバイト目を参照）してそのレコードの開始位置を知ることができる。そのレコードのレングスはディレクトリ部分の次のバイトをみて、それと開始位置との差から計算して求める。この開始位置とレングスが求まったならば１レコード分のビット列用バッファに入れる。これで圧縮パターンフェッチ処理が終了し次の文字パターン復元処理に移る。

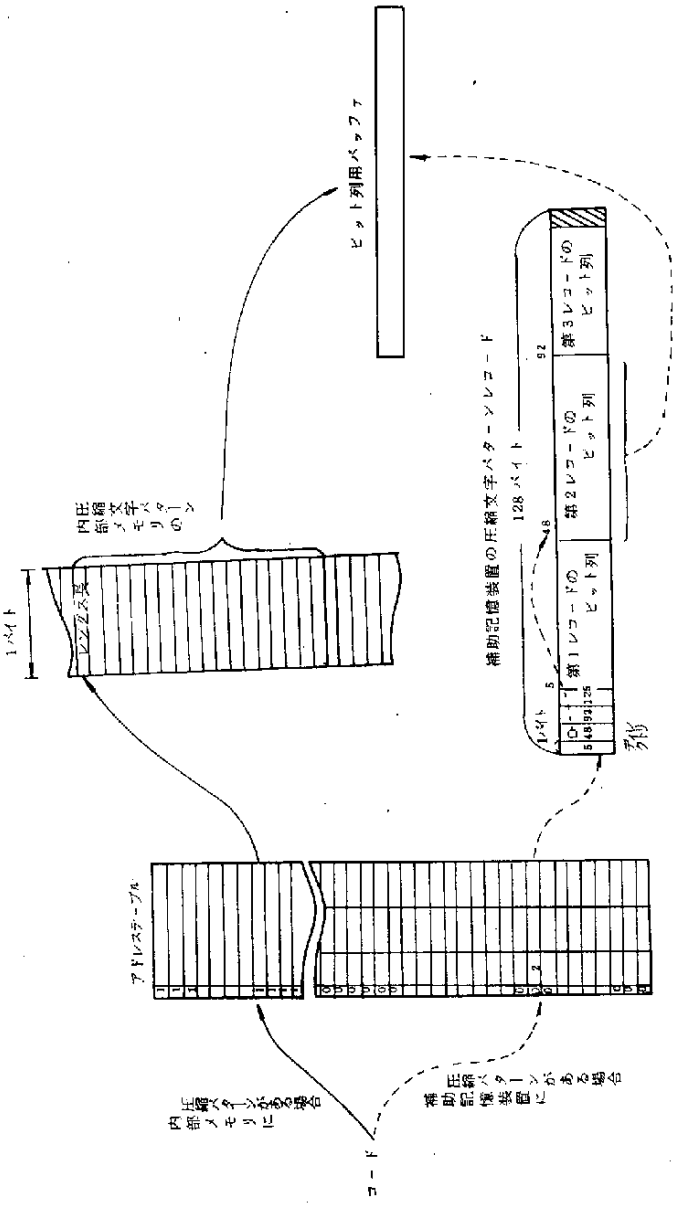


図５－１５　圧縮パターンのフェッチ

(2)　文字パターン復元処理

この処理はトリー・テーブルとサブパターンを使用して機能し、パターン復元エリアに文字パターンを復元する。

トリー・テーブル、パターン復元エリア、サブパターンは以下のようなものである。

・　トリー・テーブル

複合ハフマン法のトリー群の１つを示したもので、１６種類のトリー・テーブルがある。各トリー・テーブルには０から１５までの番号がつけられ、前出のサブパターン番号と対応している。１つのトリー・テーブルの構造は図５－１６のようである。

テーブルのパターン番号はサブパターンの番号を示し、レングス長はビット列中の有効なビット数を表わしている。なお図中、ビット列の右方空白には０がうまっている。

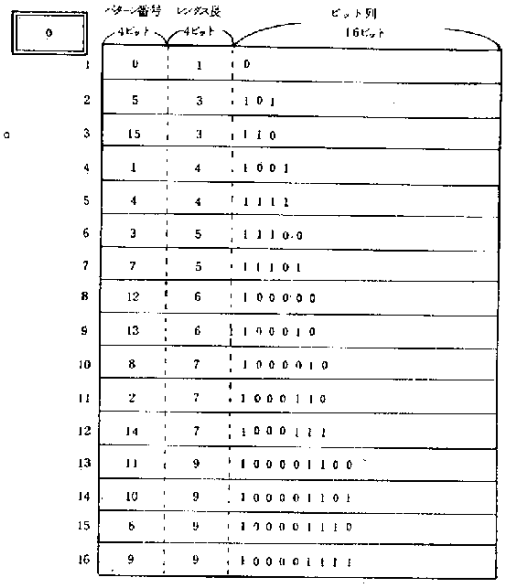


図５－１６　トリー・テーブル

・　パターン復元エリア

このエリアは、パターンを復元した後、次のステップにパターンを受けわたすバッファであり、復元処理の必要性から2×2ビットの小エリアで横１２×縦１２のマトリックス形になっている（図５－１７）。このエリアには復元のためにサブパターンを記憶していくが、その順番は左から右へ、上から下に決られている。

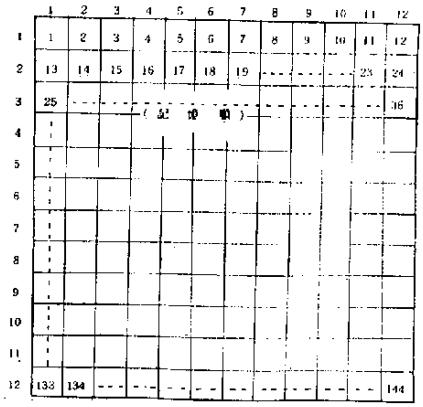


図５－１７　パターン復元エリア

• サブパターン

これはトリー・テーブル内のパターン番号（４ビット）で参照されることから０～１５の番号がつけられ、番号と対応するサブパターンは図５－１８のようである。

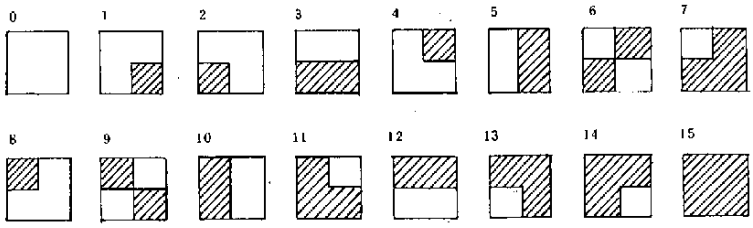


図５－１８　サブパターン

図中、パターンの黒の部分を１、白の部分を０で表現し、パターン番号との対応は２×２のマトリックス位置に次のようにビット付けをしている。

４ビット表現：００００　（０）

０００１　（１）

⋮

１１１１　（１５）

$\begin{pmatrix} \uparrow & \uparrow & \uparrow & \uparrow \\ 2^3 & 2^2 & 2^1 & 2^0 \end{pmatrix}$

文字発生機能の処理フローを図５－１９に示す。

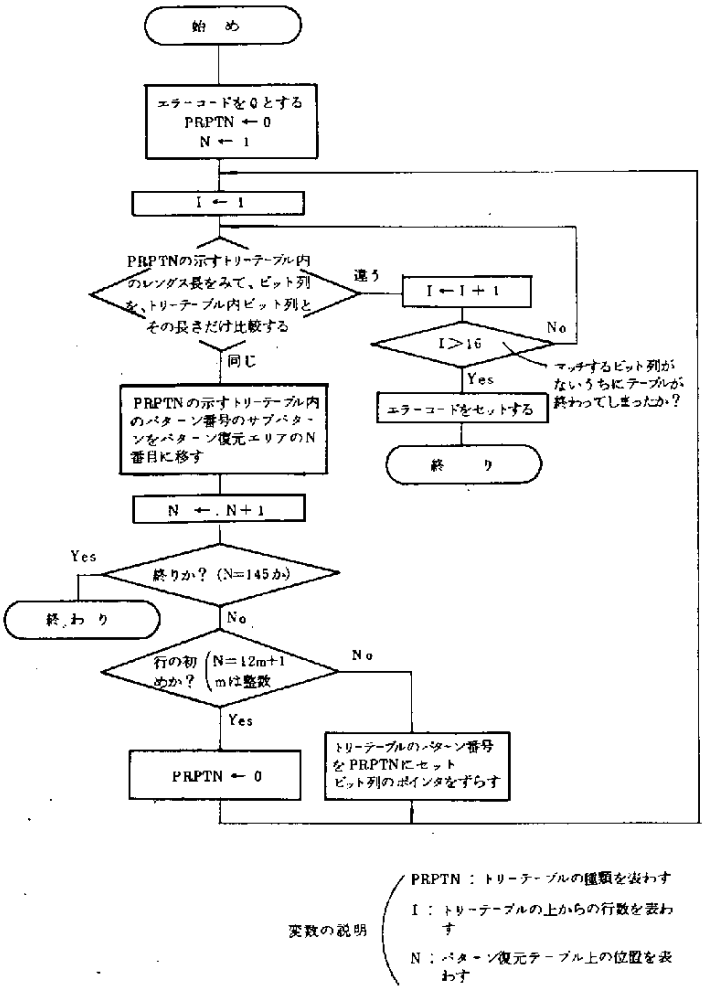


図５－１９　文字発生処理フロー

5.5.3　出力制御機能

出力制御機能は、入力制御機能と文字発生機能から受けた情報にもとずいて、センタ・コンピュータへの送信、ディスプレイ・モニタへの表示、ハード・プリンタへの出力、データ出力ファイルへのデータ出力保存といった各機能を実行する。

入力制御機能から渡される情報は、制御コードとデータ・コードであり、データ・コードには、仮名英数字記号用の文字コードと、漢字コード、項目コードの区別がある。これらは入力変換処理によってすべて漢字コードに変換することができるがセンタ・コンピュータへの問合せ方法によっては、変換を行なわないままで、送信することがある。またデータ出力保存機能においても、漢字コード・データを出力するばかりか、文字／漢字／項目の各コードと一部の制御コードを混在した

形態をとることがある。

文字発生機能から渡される情報は、漢字コードとそのパターン（ドット・マトリックス・パターン）である。パターン・データは、パターン復元エリアに２４×２４ドット・マトリックスのパターンが復元された形式で渡され、これを表示機能、ハード・プリント機能に受け渡して、ディスプレイ・モニタに表示したり、あるいはハード・プリンタに直接印刷したりする。

各処理ブロックごとの概略は以下のとおり。

(1) 出力制御ブロック

出力制御機能の全体をコントロールするブロックで、入力制御、文字発生の各機能からの情報を受けて、送信、表示、ハードプリント／コピー、データ出力の各処理ブロックを制御する。

(2) 送信ブロック

入力ー出力の制御機能（ブロック）を経由した送信情報を、センタ・コンピュータに送信するブロックである。送信情報には、制御コードと文字コードが含まれるが、入力変換ブロックを経たデータは漢字コードに変換されており、これを送信する。

(3) 表示ブロック

入力と文字発生の各機能から、漢字コードとその文字パターンを得て、これを表示画面のパターン・リフレッシュ・メモリに記憶させ、表示するブロックである。１画面は、５１２×５１２ドットのパターン・リフレッシュ・メモリに対応し、可視部分に２０字／行×１２行＝２４０字を表示する。

画面を制御する機能としては、入力モニタ用と受信データ表示用に、１字後退消去、全画面消去、入力無効（キャンセル）、復改、画面継続表示、画面後退表示、の各機能が動作する。

(4) ハードコピー・ブロック

ディスプレイ・モニタに表示されている２０字／行×１２行＝２４０字を、ハード・コピー制御コードの指示を受けて、ハード・プリンタに出力する。出力形式は表示形式と同一にし、ハード・プリンタの印字幅の左半分を使用する。

(5) ハード・プリント・ブロック

センタ・コンピュータからの受信データを、ハード・プリンタに直接出力するためのブロックである。この指示は、入力キーボードの制御キーからあるいは受信データ中の制御コードとしてハード・プリントの要求があった場合に実行し、ハード・プリンタの印字幅４０字／行に編集して出力する。

(6) データ出力保存ブロック

入力データあるいは受信データを外部記憶装置に保存するための機能ブロックである。出力制御ブロックからは、基本的には漢字コードと制御コードが混在した形で入ってくるが、この他、文字コード、項目コードを受けることもできる。

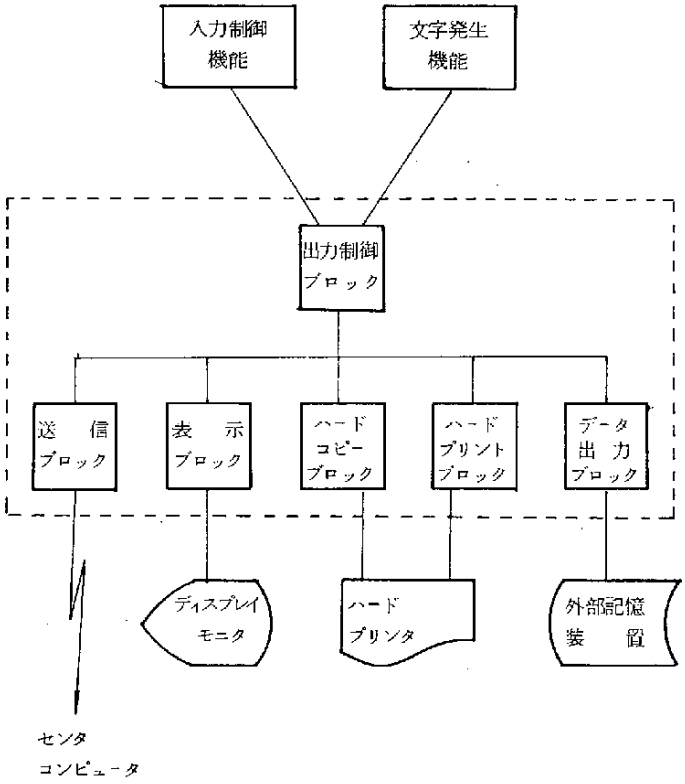


図５－２０　出力制御機能ブロック図

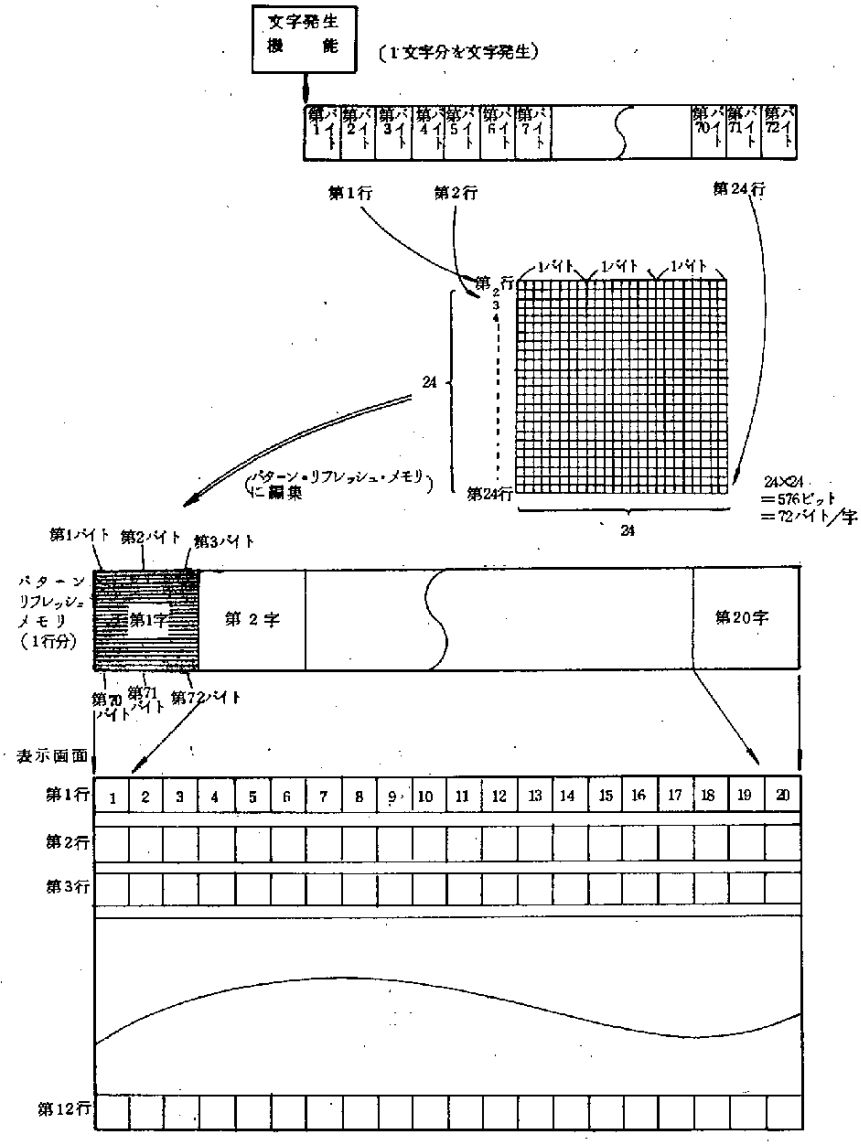


図５－２１　表示編集処理

5.6　圧縮文字パターンの作成

5.6.1　圧縮方法

日本語端末の文字発生機能で使用する圧縮文字パターン・ファイルの作成方法を述べる。

原文字パターンは２４×２４ドット・マトリックスのパターンであるが、メモリの節減をはかるためにドット・パターンを圧縮して記憶する。圧縮方式は複合ハフマン法（第３章を参照）を採用する。複合ハフマン法によって文字パターンを圧縮するには、前もって使用する文字パターン（全文字種）の統計を行なってトリーテーブルを作成しておかなければならない。この統計では、２４×２４ドット・マトリックスを２×２の小パターンの集合としてとらえ、この小パターンの前後関係の確率

を求めるものである。複合ハフマン法には合計１６種類のトリー・テーブルが必要になる。なお各トリー・テーブルは小パターンとしてとらえた１６種類のサブパターン番号順に並べたものを使用する。

圧縮処理の手順は、まず圧縮する原文字パターンの２４×２４ドット・マトリックスから２×２ドットの小パターンで分割した１２×１２のサブパターン番号の集合ファイル（サブパターン番号ファイル）を作成する。このファイルは、サブパターンの順序付けに従って、左から右、上から下へと順序がつけられて、あとの処理をこの順序で行なっていく。まずはじめに、サブパターン番号１についてコード化を行なうが、使用するトリー・テーブルは左端のサブパターンの場合０番を用いる。０番のトリー・テーブルを探索してサブパターン番号１と合致するビット列を取り出す。このビット列をビット列用バッファに移項して最初の処理が終る。次にサブパターン番号２についてのコード化では、前サブパターン番号１に対応するトリー・テーブルを使用し、同様にしてサブパターン番号２に合致するビット列を得てビット列用バッファに移項する。以下同様に、前サブパターン番号ｎ－１に対応するトリー・テーブルを使用して、サブパターン番号ｎのコード化を行なっていく。最終的にはｎ＝１４４まで同様の処理を行なっていくが、１行分の処理が終了して次の行へ進むとき、実際にはパターンは連続していないので、ここで前サブパターン番号に対応させるトリー・テーブルとして０番を使用する。

こうして１４４回の処理が終了すると、ビット列用バッファには、圧縮された文字パターンができあがる。このとき圧縮文字パターンの単位をバイト（８ビット）にして取扱うために、残ったビットを０にしてバイト調整（バイト・バウンダリ）を行ない、これを圧縮文字パターン・ファイルに記憶する。

なお文字パターンの圧縮処理は、センタ・コンピュータでバッチ処理するので、トリー・テーブルのメモリ・サイズはあまり気にせず、図５－２２のような構造にしている。

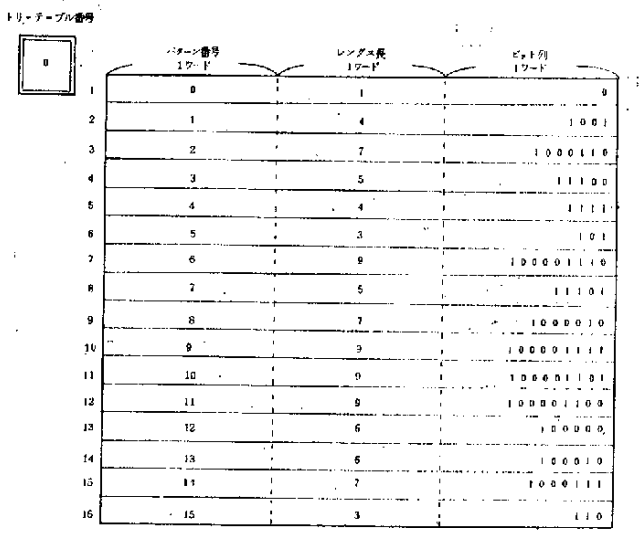


図５－２２　トリー・テーブルの構造

以上のような圧縮処理の段階で、各文字の漢字コードとそれに対応するビット列（圧縮パターン）、およびそのレングスが求められるので、圧縮文字パターンの復元処理（文字発生）で使用するアドレス・テーブルを作成することができる。しかし、この段階で圧縮パターン・ファイルを作成するのは圧縮文字パターン・ファイルのメモリ容量の効率が悪くなる。たとえばフロッピー・ディスクの1セクタに入れる順序を調整してレコードをつめ、1セクタ内の余白の部分を少なくすることができる。また使用頻度の高い文字パターンをフロッピー・ディスク内の近い場所に集めたり、関連の強い文字パターンを近い位置に記憶することなどに工夫して、フロッピー・ディスクのアームの移動を少なくし、文字発生時間の短縮を行なうことができる。これらは圧縮文字パターン・ファイルの効率を改善することになる。

このためには、ビット列のレングス情報、文字の使用頻度の情報を調整して文字パターンの配列順位を求め、それにより圧縮文字パターン・ファイルを作成する必要がある。

文字間の関連性や文字の使用頻度は、使用するアプリケーションによって異なり、ここでは具体的な圧縮文字パターン・ファイルの効率的配置については省略する。

圧縮の機能を流れ図で示すと図5－23のようになる。

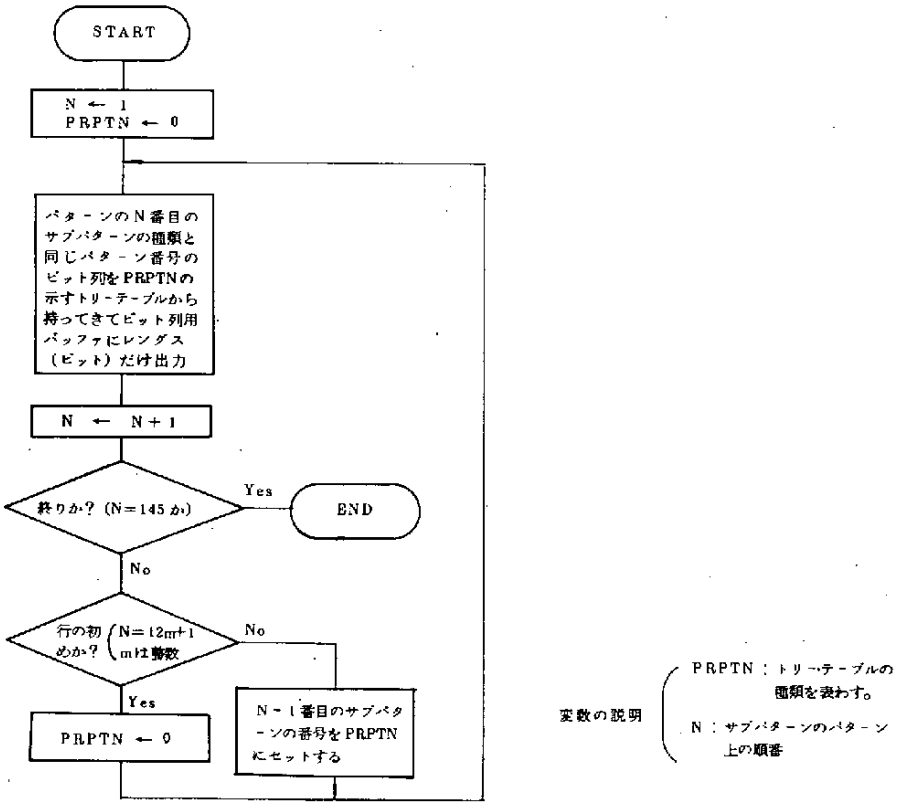


図5－23　圧縮処理フロー

5.6.2　圧縮文字パターン作成処理

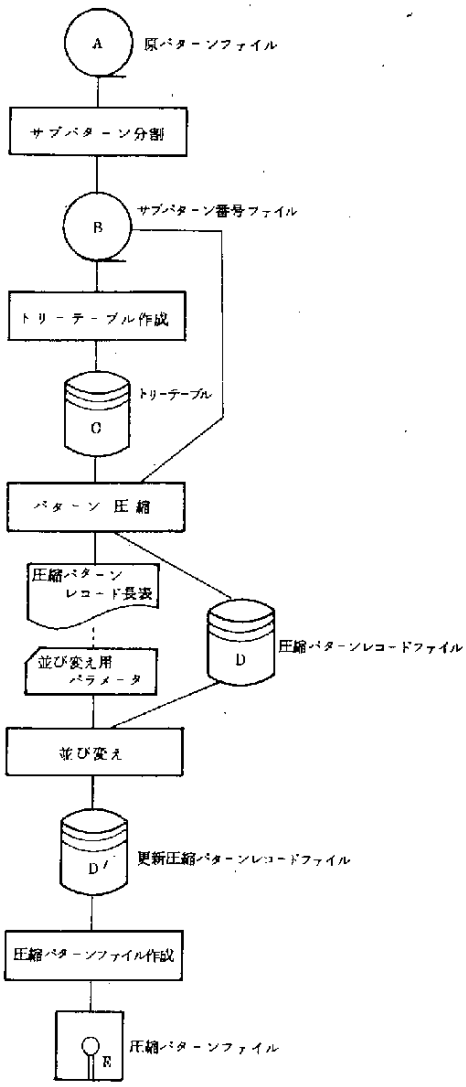


図５－２４　処理フロー

(1)　処理概説

①　サブパターンの分割

原パターン（２４×２４ドット・マトリックス）をサブパターン（２×２ドット）に分割し、この分割したものにサブパターンの種類によって０～１５の番号をふる。このサブパターン番号によりパターンを表現してファイルを作成する。

②　トリー・テーブルの作成

１６種類のサブパターンの出現率と隣接するサブパターン同士の結びつきを調べ、この結果からハフマンの符号化法を使ってコード化する。そして、パターン番号、コード長、コード（ビット列）を表にしてトリー・テーブルを作成する。

③　パターンの圧縮

サブパターン番号ファイルのパターンからトリー・テーブルを使って圧縮レコードファイルを作成する。この際に圧縮文字パターンのレングス長と漢字コードの表をプリントする。

④　並び変え

③で出力した漢字コードと圧縮文字パターンの表と漢字の出現率や関連性の情報により圧縮パターンレコードを並び変える。

⑤　圧縮パターン・ファイルの作成

補助記憶装置（フロッピーディスク）に圧縮パターン・ファイルを作成する。

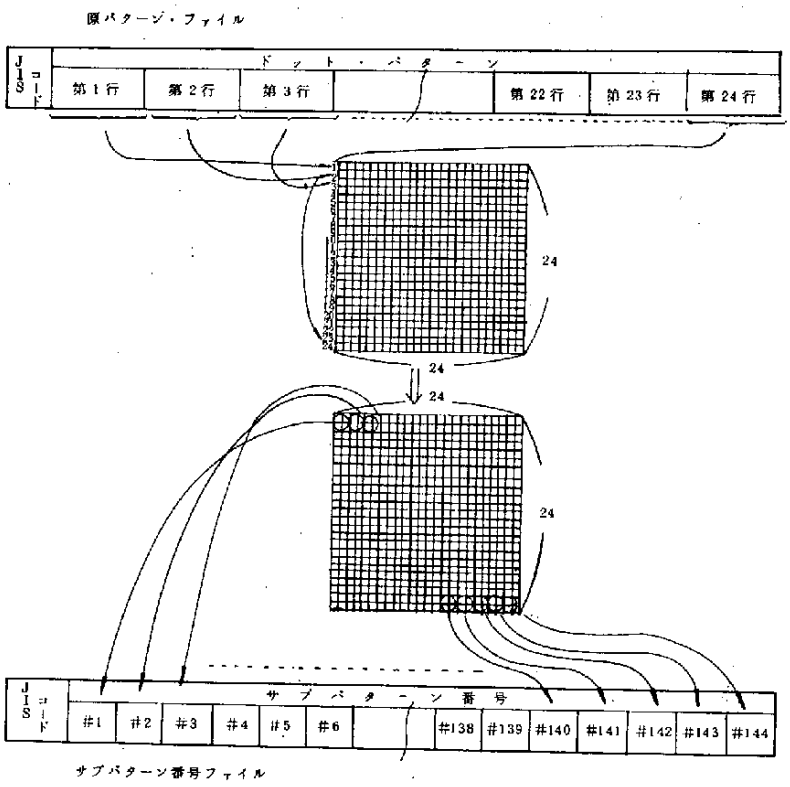


図5－25　サブパターンの分割処理

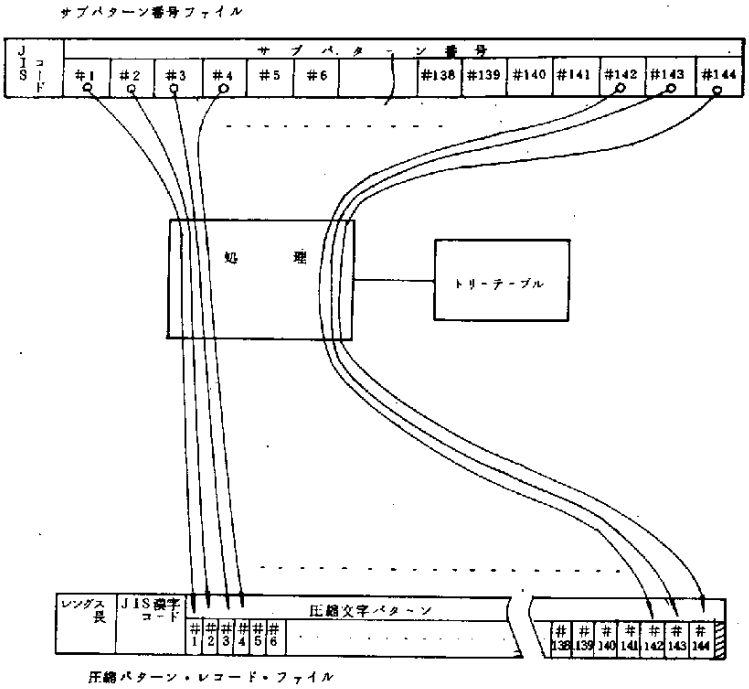


図５－２６　パターンの圧縮処理

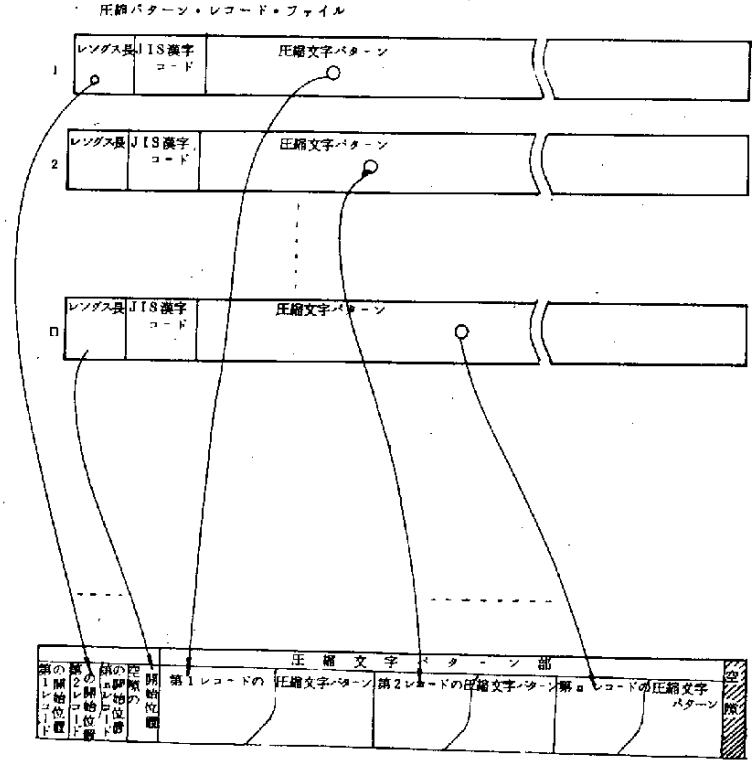
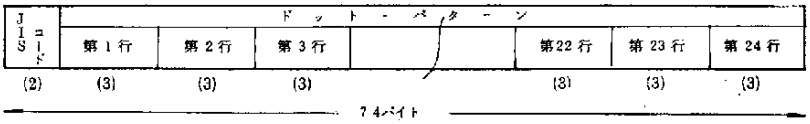


図５－２７　圧縮パターンファイルの作成処理

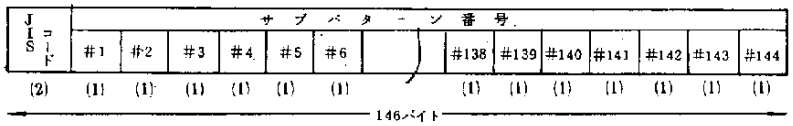
(2) ファイル形式

① 原パターン・ファイル



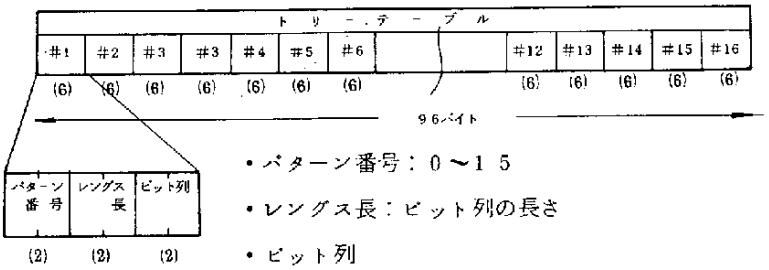
- ＪＩＳコード：ドット・パターンに対応したＪＩＳ漢字コード（２バイト）
- ドット・パターン：２４×２４のドット・マトリックス・パターン

② サブパターン番号ファイル



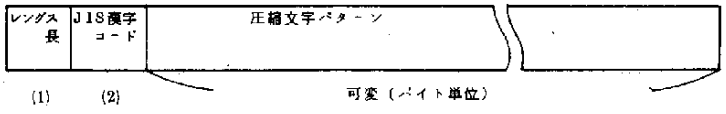
- ＪＩＳコード：ドット・パターンに対応したＪＩＳ漢字コード（２バイト）
- サブパターン番号：０～１５

③ トリーテーブル



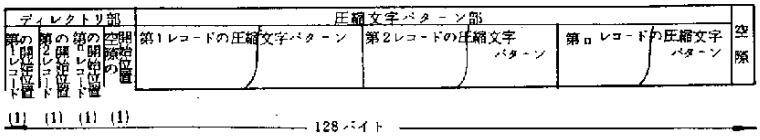
- パターン番号：０～１５
- レングス長：ビット列の長さ
- ビット列

④ 圧縮パターン・レコード・ファイル（更新圧縮パターン・レコード・ファイル）



- レングス長：圧縮文字パターンのビット列の長さ（バイト単位）
- ＪＩＳ漢字コード：圧縮パターンに対応したＪＩＳ漢字コード（２バイト）
- 圧縮文字パターン：最後のバイトに空隙が生じたら０をつめてバイト単位とする。

⑤ 圧縮パターン・ファイル



- ディレクトリ部：１２８バイト内に入る圧縮文字パターン数＋１の部分から成る
    - 第ｉレコードの開始位置：そのレコード開始位置の先頭からのバイト数
- 圧縮文字パターン部：１２８バイトにつめることのできるだけレコードを含んでいる
    - 第ｉレコード：バイト単位
- 空隙：レコードをつめた残り、０がつめられる。

5.7　入力変換辞書の構成

日本語端末の補助記憶装置に入力変換辞書を創成しておくことによって、入力制御機能ではキーボードから入力される項目コードや文字コードを漢字コードに変換することができる。主な変換辞書は、項目変換辞書とカナ漢字変換辞書であり、これらを補助記憶にアプリケーション別にファイル化しておけば、交換も容易で端末の隔通性が発揮できる。

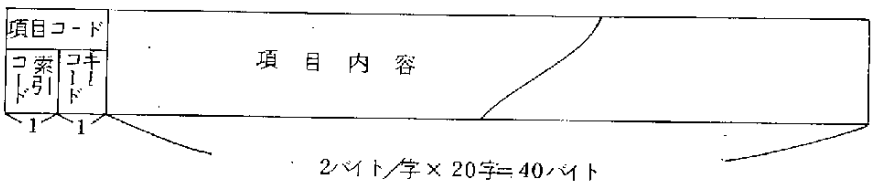
なお補助記憶として安価なフロッピーディスクを想定しているため、アクセス速度が遅いという問題は残るが、機能面の評価を優先して付加構成に組込んでいる。

以下に変換辞書の構成を概説する。

(1)　項目変換辞書

項目変換辞書は、項目キーボードから発信する項目コードから、日本語情報の漢字コード列に変換するための辞書ファイルである。辞書のレコード構成は、項目コードと項目内容（日本語情報）が対応しており、レコード数は項目数分の1000レコードからなっている。ファイルは補助記憶装置（フロッピーディスク）に順編成ファイルとして作成し、これを項目コードからダイレクト・アクセスするために簡単な演算ルーチンを有している。

- レコード構成



(レコード・サイズ：28バイト)

- 項目コード：項目キーボードから発信するコード
    - 索引コード：索引キーの識別コード（１～２０）
    - キーコード：項目キーの識別コード（１～５０）
- 項目内容：漢字コード列。最大２０字。残りゼロ。

項目変換辞書の作成は、センタ・コンピュータを利用してバッチ処理で辞書ファイルを作成する方法が一般的であり、効率もよい。しかし端末にプロセッサ機能を持たせて、スタンドアロン型で作成する方法も可能である。とくに項目入力キーボードの割当は、端末利用者がアプリケーションごとに設定できるようにしておくと便利である。ここでは後者の方法を採用して概述する。

使用する入出力機器は、項目入力キーボードと文字キーボード、モニタ表示用にディスプレイそして辞書を作成する補助記憶（フロッピーディスク）があればよい。

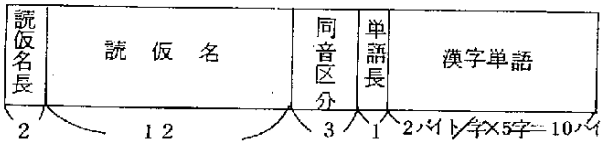
作成手順は、項目入力キーボードの索引キーと項目キーを指示して項目コードを発生し、これに割当てる項目を文字キーボードから入力する。文字キーボードには漢字がないから、漢字部分は外字機能キーとＪＩＳ漢字コード表の区点コード４桁で１字分を入力する方法をとる。

この手順で１０００項目分作成しようとすると長時間かかるため、この方法は項目変換辞書の更新用に効果的であることがわかる。この機能を利用するためには、辞書更新プログラムを端末に作成しておかなければならない。実行プログラムをフロッピーディスクに登録しておいて呼出して利用する。

(2) カナ漢字変換辞書

カナ漢字変換辞書は、漢字・漢語の読仮名から漢字コードの日本語情報に変換するための辞書ファイルである。辞書のレコード構成は、読仮名と漢字単語が対応しているほか、同音異義語の識別をするための同音区分からなっている。レコード数は最大５０００語が登録できるような記憶容量を補助記憶装置（フロッピーディスク）に確保してある。ファイルは4レコードをブロッキングした順編成ファイルとして作成し、これを格納しながらアドレス索引を作成して、全体的に索引編成ファイルとして構成する。

・ レコード構成



（レコード・サイズ：２８バイト）

- 読仮名長：読仮名の文字長
- 読仮名：漢字単語の読仮名
- 同音区分：同音異義語識別区分
- 単語長：漢字単語の文字長
- 漢字単語：漢字コードの単語

３０００語についての分析結果から、読仮名の索引をとってアクセスしようとすると、1字の索引では対応する語数が多すぎてシーケンシャルなアクセスで探索するのと同等になってしまう。2字の索引でさえも、一意に決定するのは約６％にすぎず、他の９４％には平均１０語、最大１６０語の対応がある。3字の索引にすると、一意性は約３０％になり、他の７０％に平均4語、最大５０語の対応になる。このように索引文字長を多くとれば一意性が高まるが、索引のための記憶容量の増加や処理手順のめんどうさにつながって効率はよくない。とくに低速・少容量のフロッピーディスクでは効果は望めないから、索引部分をできるだけ主記憶にもって対処しなければならない。

カナ漢字変換辞書の作成は、センタ・コンピュータを利用したバッチ処理で行なうものとし、端末側での更新は行なわない。ただし、アプリケーションごとにフロッピーにファイル化しておけば、フロッピーを交換するだけで対処できる。

付記1　漢字出力システムの概要

1. システム構成

(1) 漢字パターン記憶サブシステム

２８×２４ドット構成で生成されている文字フォント・パターンを当システム用の２４×２４ドット構成に変換するとともに、２５８２字種のパターンを１枚のフロッピーディスク（２４２.９ＫＢ）に格納するためのレコード編集およびダイレクト・アクセス・ファイルの創成を行なう。

(2) 漢字出力サブシステム

（漢字仮名入力システムで作成された）漢字データ・ファイルから漢字コード（ＪＩＳ漢字コード）データを入力して、各漢字コードに対応した漢字パターンをアクセスし、ドット・プリンタに編集・出力する。また漢字データ中に含まれてくる制御コードによりプリンタの制御を行なう。

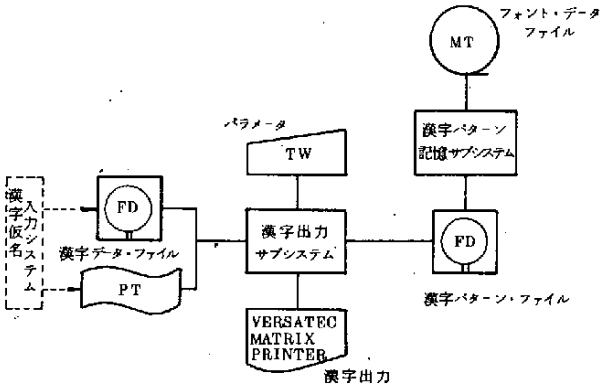


図１　漢字出力システム構成図

2. 処理内容

(1) 漢字パターン記憶サブシステム

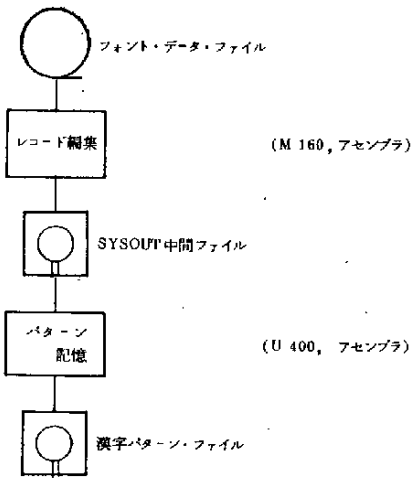


図２　漢字パターン記憶処理

① レコード編集プログラム

ＪＩＳ漢字コードに対応したドット・パターン（２８×２４ドット構成）がフォント・データ・ファイルに生成されている。

当システムでは処理のしやすさとファイル効率の良さから２４×２４ドット構成にしてパターンを取扱う。

そのため１行２８ドット構成のうち、左右２ドットずつ計４ドットを切捨るドット・パターンの変換処理を行なう。

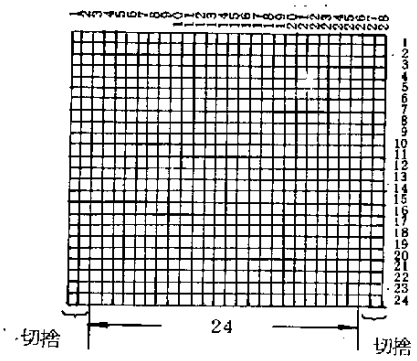
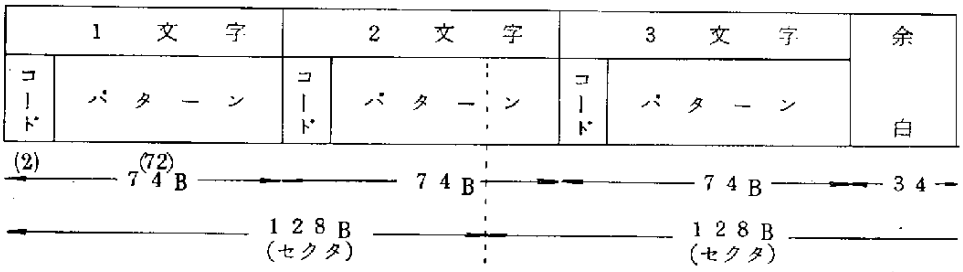


図３　ドット・マトリックス編集

ドット・パターンの変換後、フロッピー・ディスクにＳＹＳＯＵＴ中間ファイルを作成する。

文字数2,５８２字のドット・パターンは、１枚のフロッピー・ディスク1898セクタ(１２８バイト／セクタ）に１字／セクタでは容量不足になることから、２セクタに３字分のドット・パターンを記憶するようにレコード編集を行なったうえでファイル作成する。



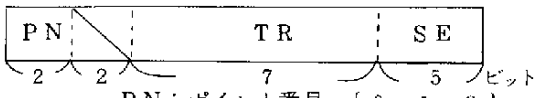
② パターン記憶プログラム

シーケンシャル・アクセス・モードのＳＹＳＯＵＴ中間ファイルから、ダイレクト・アクセス・モードの漢字パターン・ファイルを作成する。

漢字パターン・ファイルのフロッピー・ディスクの割当は、先頭３トラックにアドレス索引を、他の７０トラックにパターン・データを記憶する。

アドレス索引は、ドット・パターンの記憶アドレスとして、トラック番号、セクタ番号およびポイント番号（３レコード／２セクタの開始位置情報）を確保して、これをＪＩＳ漢字コードで対応づけされた索引テーブルに書込んで、全データ記憶終了後にアドレス索引トラックとして格納する。

アドレス索引　　2バイト＝16ビット



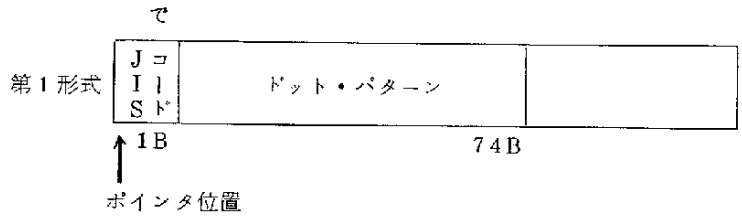
PN：ポイント番号　{0、1、2}

TR：トラック番号　{04～73}

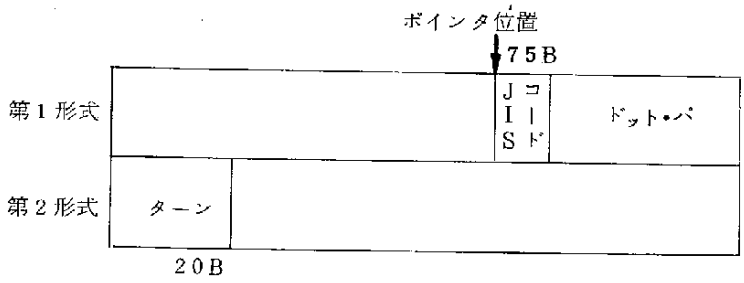
SE：セクタ番号　{01～26}

(注) ポイント番号とレコード開始位置の関係

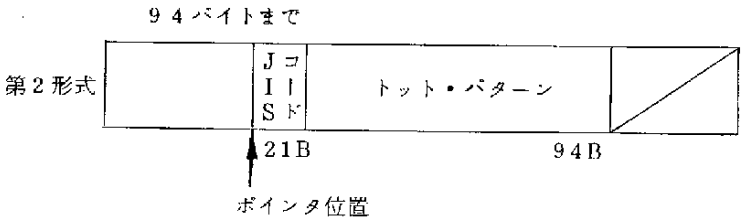
1) PN＝0；第1形式レコード(セクタ)の先頭位置から74バイトまで



2) PN＝1；第1形式レコードの75バイト目をポインタ位置として、第2形式レコードの20バイトまで



3) PN＝2；第2形式レコードの21バイト目をポインタ位置として、94バイトまで



JIS漢字コードの2バイト／字に対し、索引テーブルの第1索引として第1バイトを、第2索引として第2バイトをそれぞれ対応づける。

JIS漢字コードの範囲は、第1水準漢字集合の漢字に限られることから、第1バイトは33～79（16進22～4F）、第2バイト33～126（同22～7E）であり、このうち第1バイトの40～47（同28～2F）の8ブロックが空領域であるため、索引テーブルのマトリックスは39×94になっている。

第1バイト	33	34 …… 39	48 …… 79
第1索引番号	1	2 …… 7	8 …… 39

第2バイト	33	34 …… 125	126
第2索引番号	1	2 …… 93	94

・アドレス索引テーブル

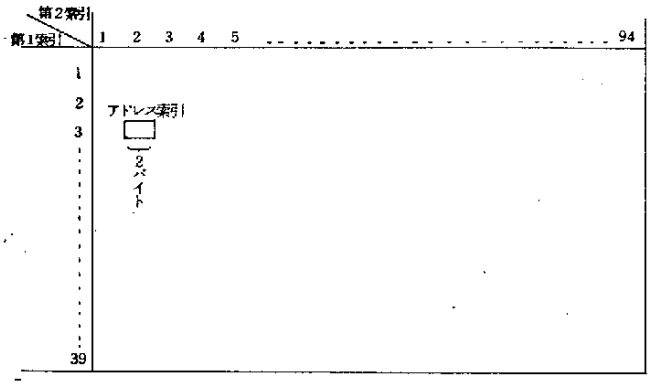


図4　アドレス索引テーブル

・アドレス索引テーブルのトラック割当はファイル形式を参照。

・39×94＝3666字種分の領域のうち、当システムでは当初2582字種を取扱う。そのため1084字種分の領域は空（NUL）である。（領域使用率約70.4％）

(2) 漢字出力サブシステム

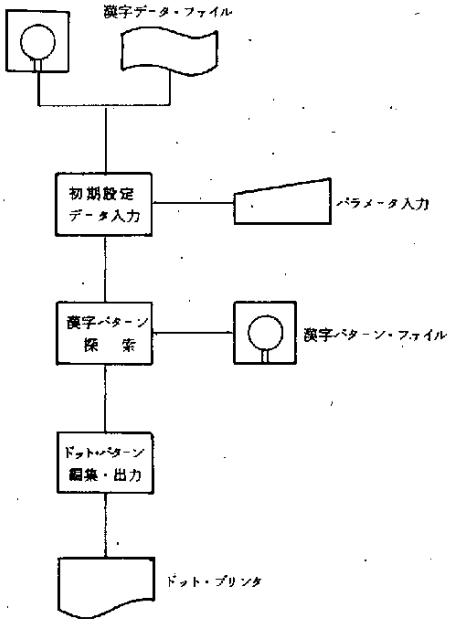


図5　漢字出力処理

① 初期設定・データ入力ルーチン

漢字データを入力する前に次の事項について初期設定を行なう。

- ○ 漢字データ・ファイルのメディア指定
- ○ 出力用紙の指定
- ○ 出力制御情報の初期設定
- ○ アドレス索引テーブルのローディング
- ○ 高頻度出現文字パターンのローディング
- ○ 各種入出力バッファのクリア

A 初期設定

a）漢字データ・ファイルのメディア指定

フロッピーディスク・ファイル、紙テープ・ファイルのメディア指定をタイプライタからパラメータとして受け、指定されたファイルのオープン処理を行なう。

〔パラメータ入力形式〕

＊＊＊INPUT DEVICE=｛FD / PT｝(復改)

(省略((復改)のみ)はFD)

FD：フロッピーディスク

PT：紙テープ

b）出力用紙の指定

ドット・プリンタ（VERSATEC）の出力用紙は、現在（53. 10）折りたたみ紙（ミシン目入り用紙）を標準としているが、ロール紙の使用も可能で、かつ漢字プリンタの機能としても可変サイズで書式仕様が可能であることが望まれる。そのため出力用紙の選定を、タイプライタからパラメータとして入力する。

〔パラメータ入力形式〕

＊＊＊PRINT PAPER=｛Z / R｝(復改)

(省略はZ)

Z：折りたたみ紙

R：ロール紙

c）出力制御情報の初期設定

ドット・プリンタ（VERSATEC）の出力制御情報をタイプライタからパラメータとして受け、初期状態にセットする。

漢字データ・ファイルに（SOH）ヘッディング情報が存在する場合は、その情報をタイプアウトして確認・修正の指示を受ける。

〔処理手順〕

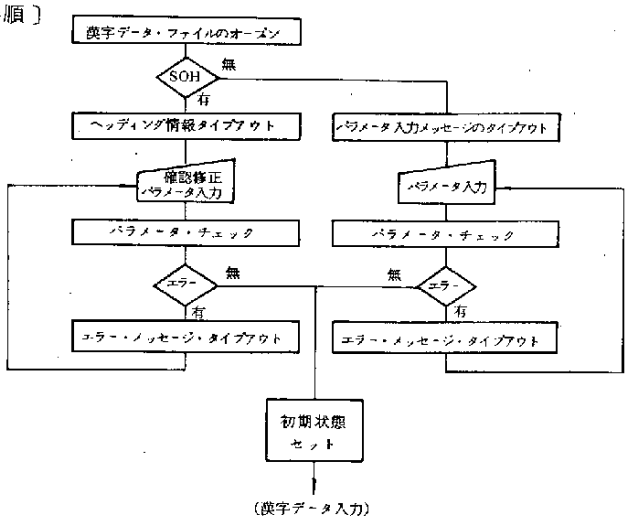


図6　出力制御情報の初期設定

出力制御情報は、ＳＯＨレコード（ＳＯＨコード以下）のヘッディング情報にある制御項目６種と同じであり（以下の(i)〜(vi)）、それぞれについてパラメータの入力あるいは確認・修正を行なう。

なおタイプアウトでは、２バイト・コードの制御文字と１バイト・コードのパラメータをそれぞれタイプライタの印字コードに変換して出力する。

またパラメータの入力は、タイプライタ・キーボードから制御文字とともに行なわれ、タイプインが省略されたときは、確認済またはパラメータ省略の標準値がセットされるものとみなす。

(i)　組版単位モード選択（ＳＣＵ）

〔タイプアウト／パラメータ入力形式〕

ＳＣＵ＝S_1、S_2

（注）　S_1、S_2 ）選択パラメータ（０〜８）

当システムでは０（文字）、４（ミル）、５（メートル法）のみ有効。

０（文字）は、２４×２４ドット文字単位による。

４（ミル）は、１ドット＝１０ミルに相当する。

５（メートル法）は、１ドット＝0.２５㎜に相当する。

省略の場合は、４（ミル）とする。

(ii)　頁サイズ指定（ＰＧＳ）

〔タイプアウト／パラメータ入力形式〕

ＰＧＳ＝N_1

(注)　N_1……数値パラメータ

頁サイズの単位はＳＣＵのS_2 パラメータによる。

省略の場合、８２０ドット（８２００ミル）とする。

(iii) 動作位置指定（ＳＡＰ）

〔タイプアウト／パラメータ入力形式〕

ＳＡＰ＝N_1 、N_2

(注)　N_1, N_2 } 数値パラメータ

字方向の動作位置を指定するパラメータN_1 は、ドット・プリンタの機能上１０２４ドット／行の範囲内で始端を指定する。

N_1 の省略の場合は、第１ドットを始端とする。

行方向の動作位置を指定するパラメータN_2 は、頁サイズの範囲内（折りタタミ紙のときは１枚分、ロール紙のときはＰＧＳ指定による）で始端行を指定する。

N_2 の省略の場合は第１ドット行を始端とする。

いずれもＳＣＵによる単位で指定する。

(iv) 文字ピッチ（ＣＰ）

〔タイプアウト／パラメータ入力形式〕

ＣＰ＝N_1

(注)　N_1 ……数値パラメータ

文字サイズ（２４×２４ドット）を固定にして、文字と文字の間隔を指示する。

単位はＳＣＵの第１パラメータS_1 による。

省略のときは２ドット（２０ミル）とみなす。

(v) 行ピッチ（ＬＰ）

〔タイプアウト／パラメータ入力形式〕

ＬＰ＝N_1

(注)　N_1 ……数値パラメータ

文字サイズ（２４×２４ドット）を固定にして、行と行の間隔を指示する。

単位はＳＣＵの第２パラメータS_2 による。

省略のときは４ドット（４０ミル）とみなす。

(vi) ノンブル（ＰＧ）

〔タイプアウト／パラメータ入力形式〕

ＰＧ＝S_1

（注）　S_1 ……選択パラメータ（0～6）

ページングの書込み位置を指示する。

ＳＡＰにより書込み場所が無い／不足する場合は無効となる。

また、ノンブル値が４～６の場合（下側に印字）には、１行分の領域に出力する。

印字行とノンブル行の間隔は、ＬＰ指示の２倍ピッチの位置とする（不足／下側の場合はＬＰピッチと同様）省略のときはノンブル処理は行なわない。

d）アドレス索引テーブルのローディング

漢字パターン・ファイル（フロッピー・ディスク・ファイル）から、ＪＩＳ漢字コードに対応した漢字パターンをアクセスするために、アドレス索引トラック（トラック＃１～３）を読込んで、アドレス索引テーブルをコアに常駐させる。

・テーブルの記憶容量

第１索引３９×第２索引９４×ＦＤアドレス２バイト＝７３３２バイト

・テーブル・マトリックス

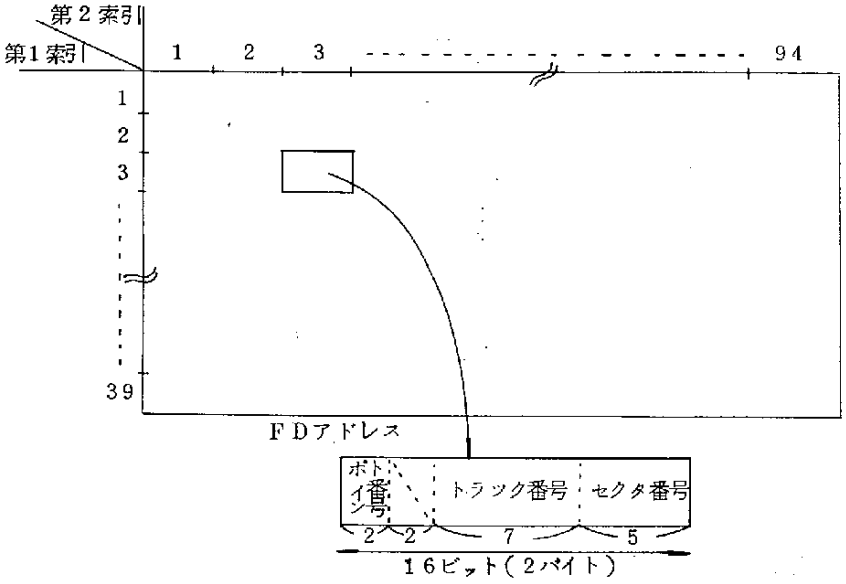


図７　テーブル・マトリックス（詳細はファイル形式参照）

e）高頻度出現文字パターンのローディング

出現頻度の高い文字記号について、文字発生速度を高めるためにフロッピーディスクからコアにパターンをローディングする。

同時に該当文字記号をアクセスするために、アドレス索引テーブルのＦＤアドレスを、コア・アドレスに変更する。

文字記号の指定は、タイプライタからパラメータ入力して行なう。最大２５０字種まで

可能とする。

・パターン・テーブルの記憶容量

７２バイト／字×２５０字種＝１８０００バイト

〔パラメータ入力形式〕

＊＊＊LOAD　PATTERN　CODE＊＊＊

＝ N_1 (復改)

＝ N_2 (復改)

⋮

＝ N_i (復改)

＝ (復改)

（注）　N_1 ～N_i：数値（列）パラメータ

・N_1 ～N_iの形式

（形１）　ＪＩＳ漢字コード　区点４桁

（例）　０１２３　（〃）

区　点

（形２）　区点範囲

（例）　０１０１－０３９０

（ＳＰ）～　（z）（記号＋数字＋英字）

＊－０３９０

（同上）

４６０１－４７９４（←JIS第１水準漢字集合の最大値）

（痢）　～　（腕）

4751（4752～4794は空領域）

４６０１－＊

（同上）

（形３）　形１．区点４桁、形２．区点範囲の繰返し

（例）　０３１６－０３５８、０４０１－０５８６

（０）　～　（Ｚ）　　（あ）　～　（ケ）

１６７６、３８８３、２７１６

（一）　　（二）　　（三）

・いずれの形式でも、区点コードに対応したアドレス索引テーブルの領域が空の場合は文字数に入れない（パターンをローディングすることはできない）。

・記憶領域を越えて（２５０字種以上）指示された場合は上位250

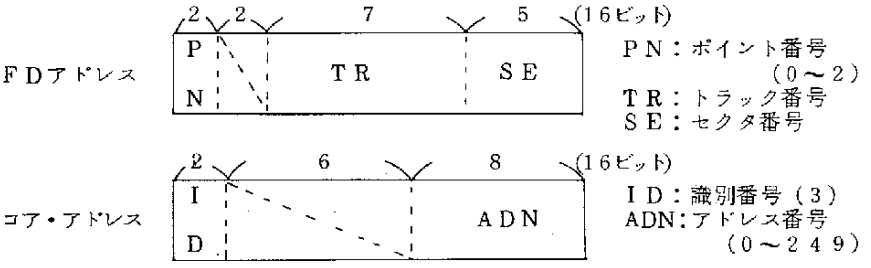
字種をローディングし、満たない場合は、標準高頻度文字セットの中から優先順に重複のない文字を自動的にセットする。

・パラメータ入力の終了は＝の後に (復改) のみがキーインされる。

・パラメータが省略された場合は、標準高頻度文字セットをローディングする。

〔標準高頻度文字セット〕		(優先順)
数　字	10	3
ひらがな	83	1
カタカナ	86	2
英　字	52	5
記　号	12	4

・FDアドレスからコア・アドレスへの変更



先頭2ビットでFDアドレスとコア・アドレスの区別をする。

(00、01、10 …… FDアドレス
11 …… コア・アドレス)

f）各種入出力バッファのクリア

漢字出力プログラムで使用する入出力バッファをクリアする（必要のないエリアは除く）。

B　データ入力

漢字データ・ファイルからデータの入力を開始する。漢字データには、漢字コード・データと制御文字（制御コード）データがあり、前者は入力データをパターン探索ルーチンに渡し、後者は編集・出力ルーチンに渡す。

前者の漢字コード・データの場合は、入力データの行調整を書式に合せて指示する。

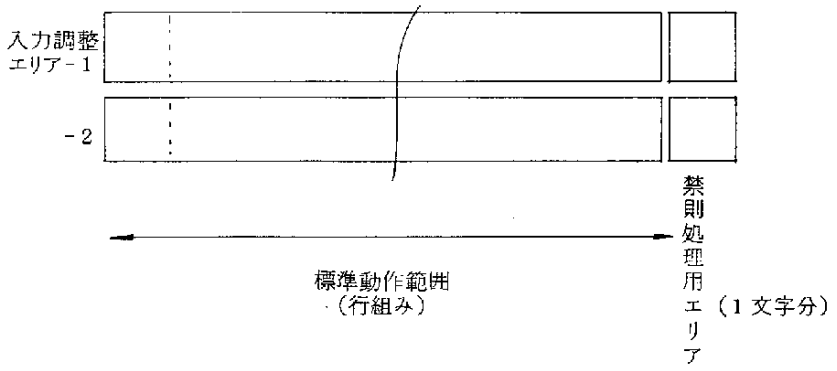
出力用紙の指定と出力制御情報の初期設定によって定められた書式に対し、編集機能の1つとして行頭・行末の禁則処理を行調整で行なう。

（行頭・行末禁則文字については5.で述べる）

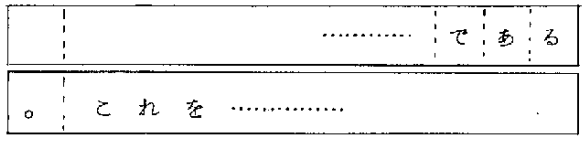
○行頭禁則処理

行頭禁則文字が行頭（行の始端）にきた場合に前行の禁則処理用エリアに移項調整する。

（繰返し記号を除く）

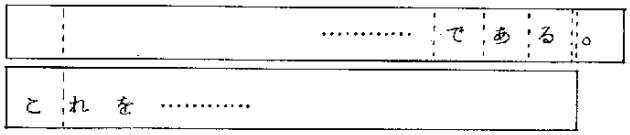


調整前



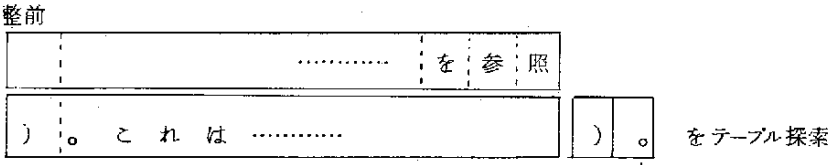
行頭禁則文字

調整後

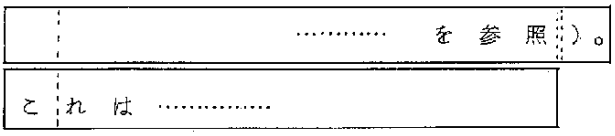


なお行頭禁則文字が2字連続する場合には禁則合成テーブルを探索して禁則合成コードに変換して禁則処理用エリアにセットする。

調整前



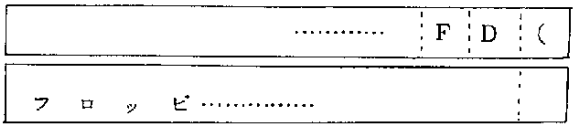
調整後



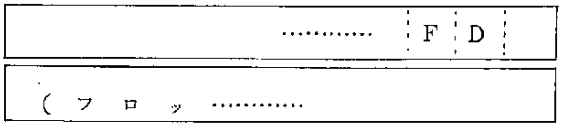
○行末禁則処理

行末禁則文字が行末（行の終端）にきた場合に次行の始端に移項調整する。

調整前　　行末禁則文字



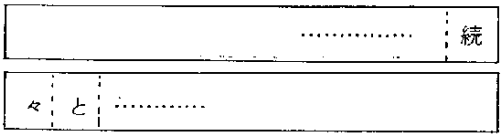
調整後　　SP（間隔）



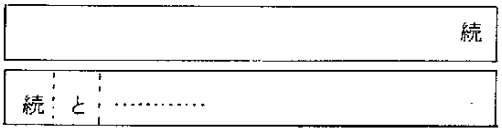
○繰返し記号処理

行頭禁則文字にある5種の繰返し記号については、前行末の文字に変換する。仮名文字のダク点付繰返し記号の場合は、前行末文字のダク点仮名文字に変換する。

調整前



調整後



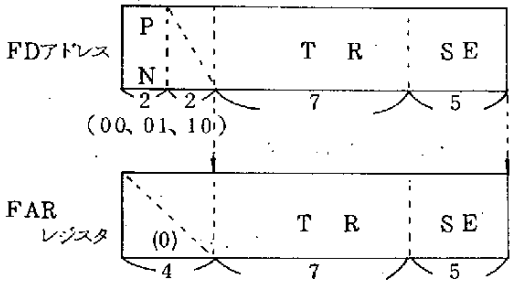
② 漢字パターン探索ルーチン

データ入力ルーチンから受けた漢字コード・データを展開し、アドレス索引テーブルを探索して、漢字パターンをアクセスする。

漢字コードは、ＪＩＳ漢字コードの第１水準漢字集合の範囲に限られることから、第１バイトは３３～７９（１６進２２～４Ｆ）、第２バイトは３３～１２６（同２２～７Ｅ）である。これを展開して、第１バイトを第１索引番号に、第２バイトを第２索引番号にしてアドレス索引テーブルを探索し、パターン・アドレスを求める。

パターン・アドレスは、ＦＤアドレスとコア・アドレスの２種があるから、これらを識別して、フロッピーディスクあるいはコア内のパターン・テーブルをアクセスして、７２バイト／字からなるドット・パターンを発生する。

○ＦＤアドレスによるアクセス法



1）ＴＲ、ＳＥをＦＡＲレジスタに移項セット

2）ＭＡＲレジスタにアクセス・バッファ領域の開始アドレスをセット

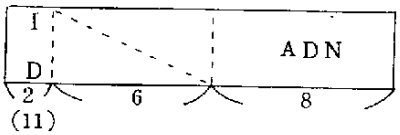
3）ＣＭＲレジスタにリード・コマンドをセット――スタート・ビットをセットして起動

・ＰＮ＝１のときはパターン・データが２セクタに分離しているため１）～３）の手順を２度連続して行なう。

そのタイミングはセクタ間の移動時間内に行なえるように（アクセス・スピードを上げるため）直後に連続してかけるようにする。

・ＰＮによってドット・パターン７２バイト（＋コード２バイト）の有効領域が異なることにに注意。

○コア・アドレスによるアクセス法



ＡＤＮ：アドレス番号（０～２４９）

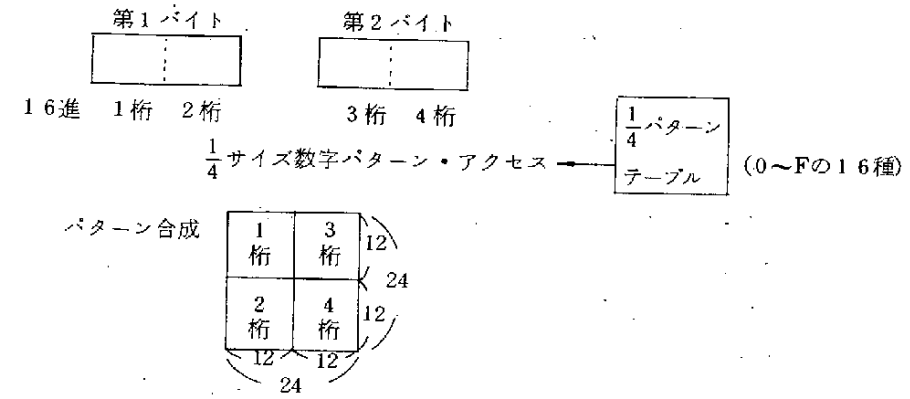
（パターン・テーブルの最大記憶文字数２５０に対応）

１）（ＡＤＮ）×７２（バイト／字）によってパターン・データの開始アドレスを求める。

２）開始アドレスをレジスタにセットしてドット・パターン７２バイトを アクセスする。

ドット・パターン７２バイトのアクセスが完了したら、ドット・パターン編集・出力ルーチンへその情報を渡す。

なお、漢字パターンの無い（登録してない）漢字コードが出現した場合には、２バイト／字からなる漢字コードを展開して１６進４桁の数値を求め、$\frac{1}{4}$サイズの数字パターンを合成してドット・パターンを作成する。



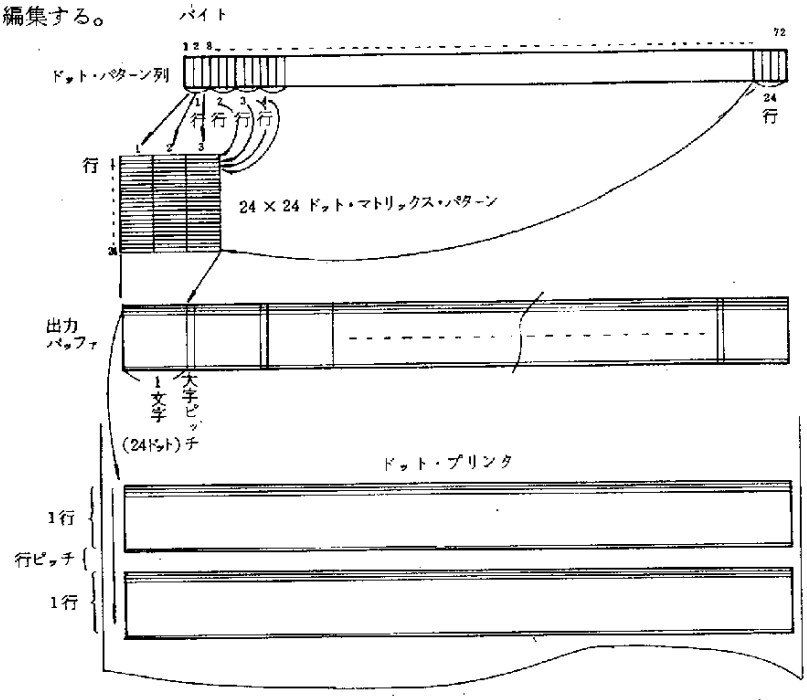
７２バイト・ドット・パターン作成

③　ドット・パターン編集・出力ルーチン

漢字パターン・データと制御文字（制御コード）によってドット・プリンタへの編集と出力制御を行なう。

パターン探索ルーチンでアクセスした７２バイト／字からなるドット・パターンを出力バッ

ファに編集する。



出力バッファに1行分のドット・パターンと文字ピッチが挿入されてから、1ライン・ドット単位にプロット・モードで出力していく。24ライン・ドットで1行分が出力されると、行ピッチ分送られて次の1行分の処理へと進む。

○出力制御機能

制御文字（制御コード）によってそれぞれの機能を処理する。

（制御文字のコードと定義は4.を参照のこと。）

伝送制御文字

SOH、STX、ETX

書式制御文字

CP、CR、FF、LF、LP、NEL、SAP、PGS、PG、SP、KSP

特殊制御文字

NUL、RBS、SCU

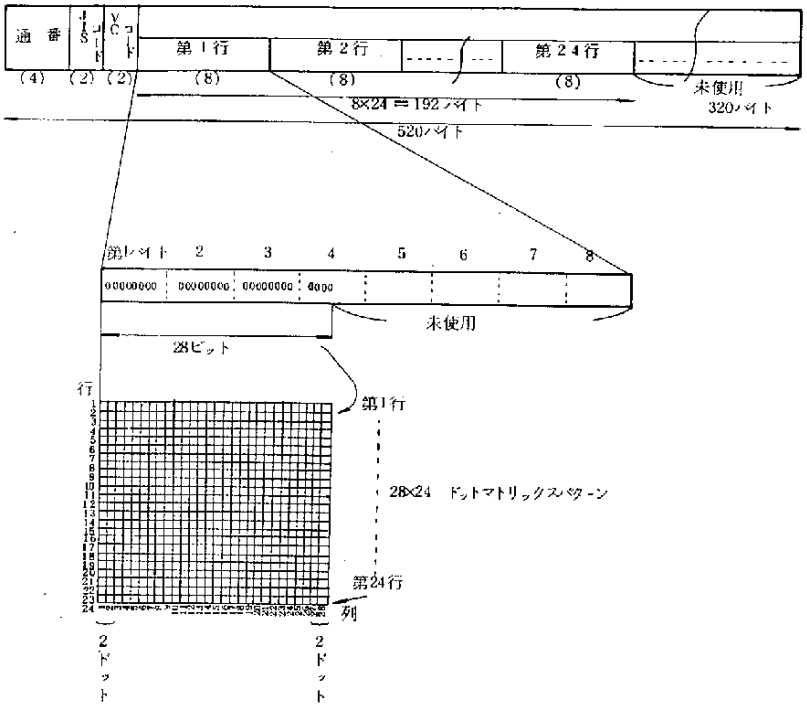
その他の制御文字

CEX（拡張符号）、C1、C2、C3

以上の他、書式範囲を越える文字列（漢字データ）が続く場合には、自動復改、自動改頁の処理を行なう。

3. ファイル形式

(a) フォント・データ・ファイル〔磁気テープ　No Label（TMから始まる）
Rec＝520B、Blk＝1040B〕



（削除するのは1、2列、27、28列の両端2ドット　計4ドット）

① 通番；4桁の通し番号

② ＪＩＳコード；ドット・パターンに対応したＪＩＳ漢字コード（2バイト）

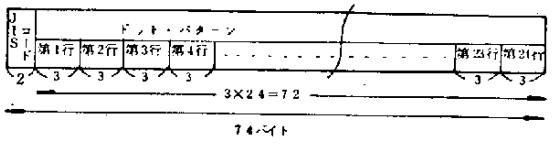
③ ＶＣコード；ＶＣ漢字コード

④ ドット・パターン；28×24のドット・マトリックス・パターン

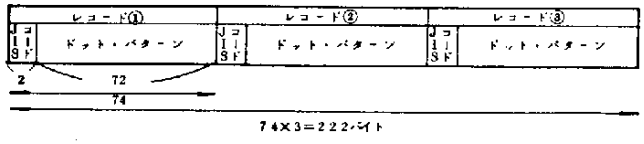
（注） ①通番、③ＶＣコードはこのシステムでは不用

(b) ＳＹＳＯＵＴ中間ファイル

論理レコード形式　（1レコード＝74バイト）

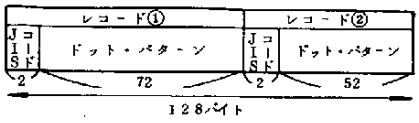


論理ブロック形式　（1ブロック＝222バイト）

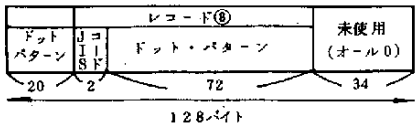


物理レコード形式　（1レコード＝1セクタ＝128バイト）

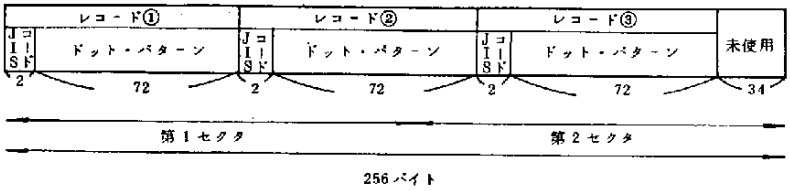
第1形式



第2形式



物理ブロック形式　（1ブロック＝2セクタ＝256バイト）



(c)　漢字パターン・ファイル

フロッピーディスク・トラック割当

フロッピーディスク仕様

装置：　PF7034A－2（2ドライブ・ユニット）

トラック数：　77トラック

データ・トラック	73
インデックス・トラック	1
交替トラック	2
CE用トラック	1

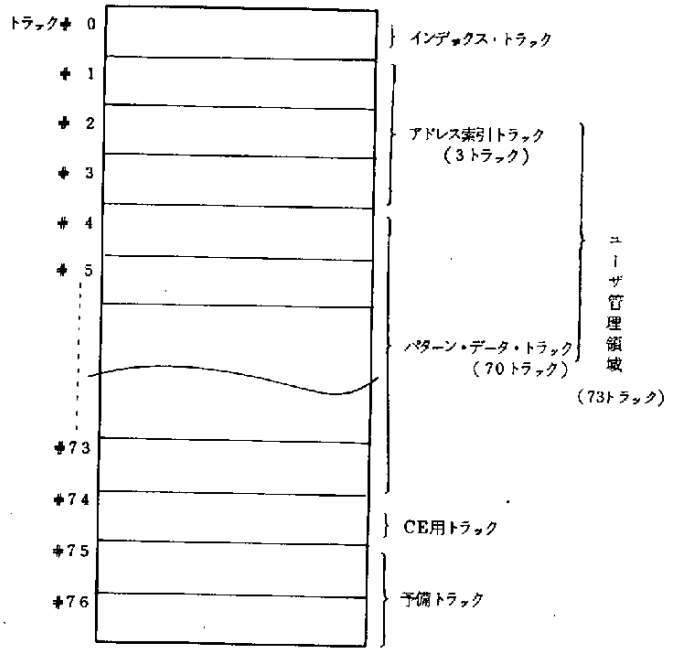
レコード形式：　1レコード＝1セクタ

128バイト／セクタ

26セクタ／トラック

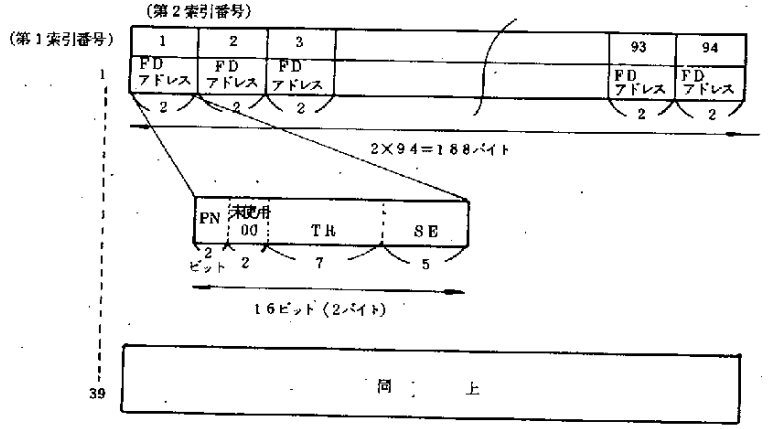
ユーザ記憶容量：　128バイト／セクタ×26セクタ／トラック×73トラック／1台

＝2422,944バイト（242.9KB）



① アドレス索引トラック（トラック#1～#3）

論理レコード形式



第1索引番号；ＪＩＳ漢字コードの第1バイトのシーケンス番号

1～39

第2索引番号；ＪＩＳ漢字コードの第2バイトのシーケンス番号

1～94

ＪＩＳ漢字コードと索引番号との対応

第1バイト	33	34	……	39	48	……	78	79						
第1索引番号	1	2	……	7	8	……	38	39						
第2バイト	33	34	35	……									125	126
第2索引番号	1	2	3	……									93	94

ＦＤアドレス；フロッピーディスク上の記憶アドレス

ＰＮ；ポイント番号（0～2）

TR；トラック番号（04～73）

SE；セクタ番号（01～26）

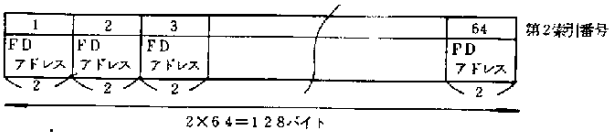
（注）PNについては、2.処理内容②を参照

レコード数；第1索引番号ごとに各レコードを構成

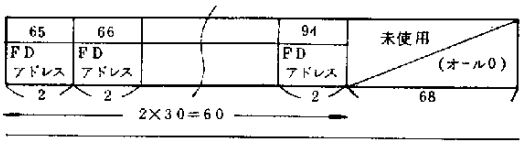
39レコード

物理レコード形式

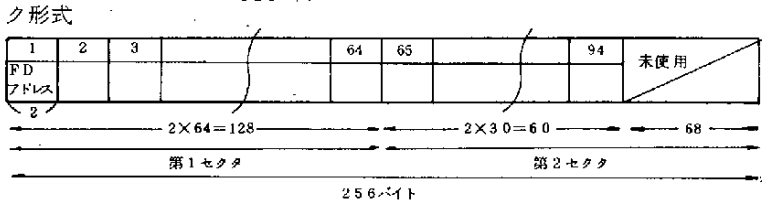
第1形式



第2形式



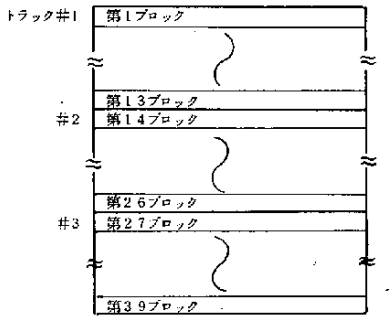
物理ブロック形式



物理ブロック数＝論理レコード数；39ブロック

39ブロック×2セクタ／ブロック＝78セクタ＝3トラック

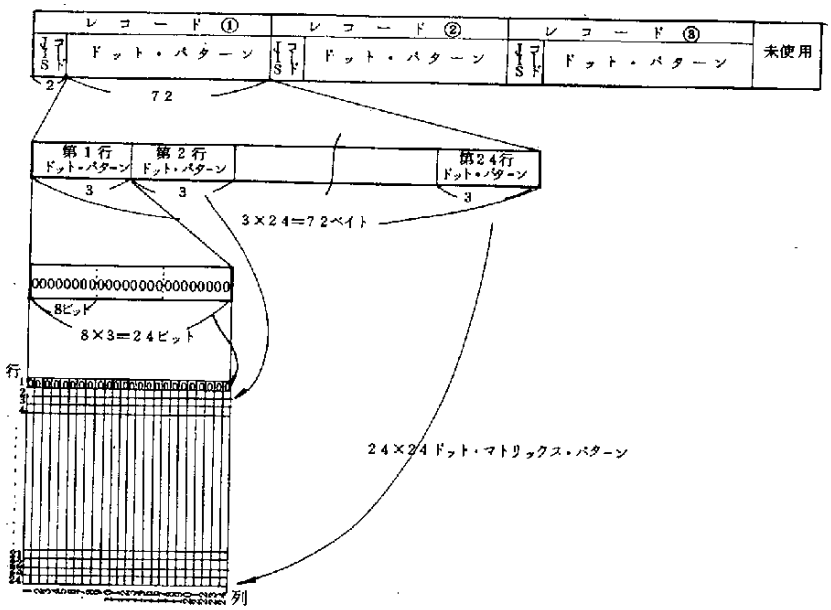
ブロック割当



② パターン・データ・トラック（トラック#4～#73）

論理レコード形式
論理ブロック形式
物理レコード形式
物理ブロック形式
} いずれも(b)SYSOUT中間ファイルと同形式

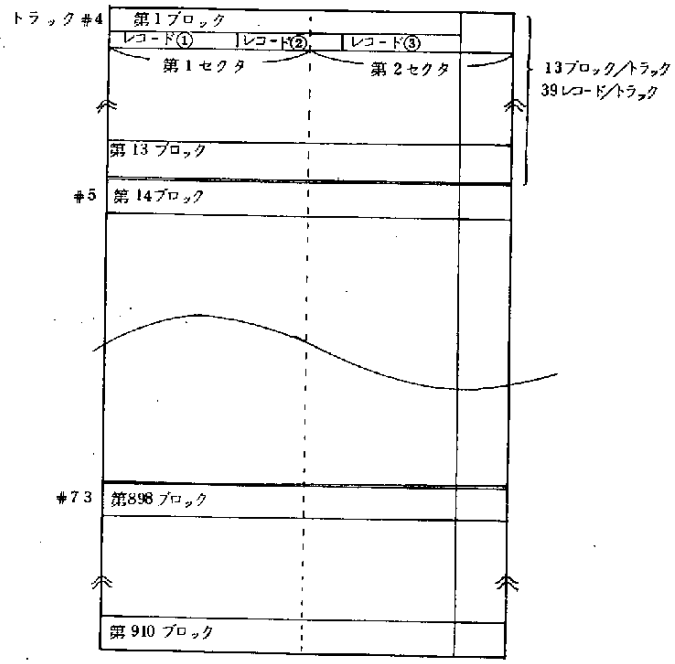
物理ブロック形式（1ブロック＝2セクタ＝256バイト）



最大レコード数；２６セクタ／トラック÷２セクタ／ブロック×３レコード＝３９レコード／トラック

３９レコード／トラック×７０トラック＝２７３０レコード

ブロック割当



(d) 漢字データ・ファイル

① フロッピー・ディスク・ファイル

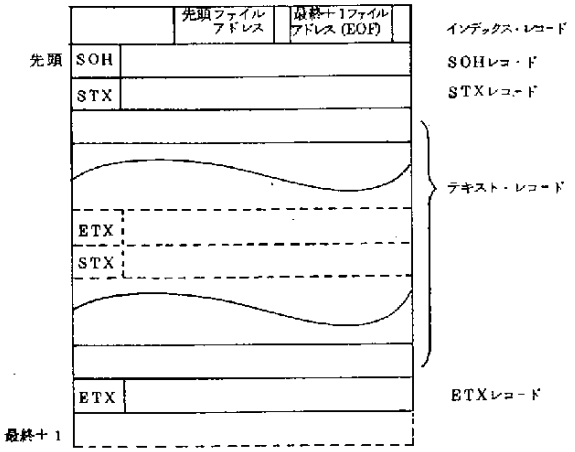
インデックス・トラック領域；トラック＃０（第８セクタ）

データ・トラック領域 ；トラック＃１～＃７３

レコード形式 ； 固定長 １２８バイト／レコード（セクタ）

アクセス・モード；シーケンシャル・アクセス

ファイル構成　；



レコードの種類

インデックス・レコード　；先頭ファイル・アドレスと最終＋１ファイルアドレス（EOF）

ＳＯＨレコード　　；ヘッディング情報レコード

ＳＴＸレコード　　；テキスト開始レコード

ＥＴＸレコード　　；　〃　　終了レコード

テキスト・レコード；漢字コード・データ・レコード

ＮＵＬレコード　　；空レコード

〔備考〕

標準ファイル形式では、ＳＯＨレコードを先頭レコードに、ＳＴＸレコードを第２レコード、以下テキスト・レコードが続いて、最終レコードとしてＥＴＸレコードが存在する。他の非標準ファイル形式では、ＳＯＨレコード、ＳＴＸレコード、ＥＴＸレコードは無くてもよい。その場合は、標準初期状態にセットされて、テキスト・データが処理される。

（注）　データの最終にＦＦ制御文字をセットする。

a）ＳＯＨレコード

先頭がＳＯＨコードで始まるヘッディング情報レコード。

SOH	ヘッディング情報

ＳＯＨコード；Ｘ'01'

ヘッディング情報；　　（ｓは選択パラメータ、ｎは数値パラメータ）

組版単位モード選択：ＳＣＵs_1 s_2

頁サイズ指定：ＰＧＳn_1

動作位置指定：ＳＡＰn_1 n_2

文字ピッチ：ＣＰn_1

行ピッチ：LPn₁

ノンブル：PGs₁

b）STXレコード

先頭がSTXコードで始まるテキスト開始レコード

STX	漢字コード・データ

STXコード；X′02′

漢字コード・データ；JIS漢字コード／制御コード・データ

c）ETXレコード

先頭がETXコードで始まるテキスト終了レコード。

ETX	カウント情報

ETXコード；X′03′

カウント情報；　　（nは数値パラメータ）

レコード数　：　C1n₁

漢字コード延べ数：C2n₁

制御コード延べ数：C3n₁

d）テキスト・レコード

先頭が、SOH、STX、ETXコード以外で始まる漢字コード・データ・レコード

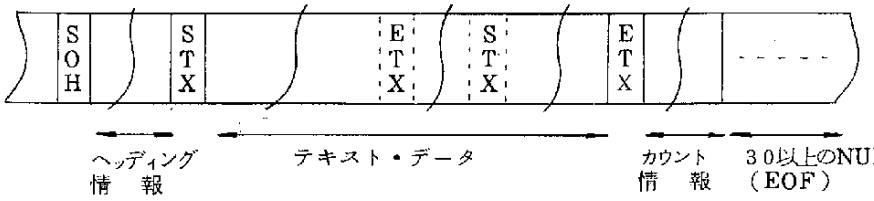
漢字コード・データ

漢字コード・データ；JIS漢字コード／制御コード・データ

② 紙テープ・ファイル

レコード形式；不定長

ファイル構成；



SOHコード；　X′01′　――――　以下にヘッディング情報が続く

STXコード；　X′02′　――――　以下にテキスト・データが続く

ETXコード；　X′03′　――――　以下にカウント情報が続く

ＥＯＦコード；Ｘ′００′が３０個以上連続の場合

ヘッディング情報；フロッピーディスク・ファイルのＳＯＨレコードのヘッディング情報と同様

テキスト・データ；フロッピーディスク・ファイルのテキスト・レコードの漢字コード・データと同様

カウント情報；フロッピーディスク・ファイルのＥＴＸレコードのカウント情報と同様

〔備　考〕

標準ファイル形式では、ＳＯＨコード、ヘッディング情報、ＳＴＸコード、テキスト・データ、ＥＴＸコード、カウント情報、の順で存在する。

他の非標準ファイル形式では、ＳＯＨコード、ヘッディング情報、ＥＴＸコード、カウント情報は無くてもよい。その場合は標準初期状態にセットされてＳＴＸコード以下のテキスト・データが処理される。

（注）　データの最後にＦＦ制御文字をセットする。

基本制御文字集合
(1バイト系)

コード(16進)	制御文字	コード(16進)	制御文字
00	NUL	10	
01	SOH	11	
02	STX	12	
03	ETX	13	
04		14	
05		15	
06		16	
07		17	
08		18	
09		19	
0A	LF	1A	
0B		1B	
0C	FF	1C	CEX
0D	CR	1D	
0E		1E	
0F		1F	

機能識別用制御文字集合
(2バイト系)

コード(16進)	制御文字	コード(16進)	制御文字	コード(16進)	制御文字	コード(16進)	制御文字
1C20	SP	1C30	SAP	1C40	NEL	1C60	
21		31		41		61	
22	CP	32		42		62	
23	LP	33		43		63	
24		34		44		64	
25		35		45		65	
26	SCU	36		46		66	
27		37		47		67	
28		38		48		68	
29		39		49		69	RBS
2A		3A		4A		6A	
2B		3B		4B		6B	
2C		3C		4C		6C	
2D		3D	C1*	4D		6D	
2E	PGS*	3E	C2*	1E		6E	
2F	PG*	3F	C3*	4F		6F	

(注) • *はJIS原案にない制御文字。
• 第1バイトはすべてCEX(1C)。

表 1　制御コード表

(2) 制御文字の定義

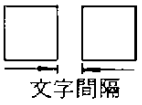
ＣＰ　：文字ピッチ（Character Pitch）文字間隔を指定する書式制御文字。

備　考　1.文字間隔は、数値パラメタで指定する。

2.パラメタの値の単位は、組版単位モードによる。

3.この指定は、次のＣＰが現われるまで有効である。

4.指定のない場合の文字間隔は、標準文字間隔（２ドット）の大きさとする。



組版単位モードで4.ミル指定では１ドット＝１０ミル

ＣＲ　：復　帰（Carriage Return）　動作位置を同一行の初めの文字位置にもどす書式制御文字（ＪＩＳ　Ｃ　６２２０）。

ＥＴＸ：テキスト終結（End of Text）　一つのテキストの終結を示す伝送制御文字（ＪＩＳ Ｃ 6220）。

ＦＦ　：書式送り（Form Feed）　動作位置を同一文字位置のまま、次の書式のあらかじめ定められた行に進める書式制御文字（ＪＩＳ Ｃ 6220）。

ＳＡＰ、ＰＧＳによって頁組みされた書式範囲で改頁する。

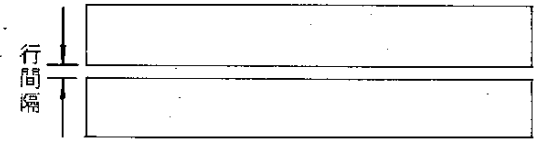
ＬＦ　：改　行（Line Feed）　動作位置を同一文字位置のまま、次の行に進める書式制御文字（ＪＩＳ Ｃ 6220）。

ＬＰ　：行ピッチ（Line Pitch）　行間隔を指定する書式制御文字。

備　考　1.行間隔は、数値パラメタで指定する。

2.パラメタの値の単位は、組版単位モードによる。

3.この指定は、次のＬＰが現わるまで有効である。



4.指定のない場合は、標準行間隔（４ドット）の大きさとする。

組版単位モードで4.ミル指定では、１ドット＝１０ミル

ＮＥＬ：復　改（Next Line）　動作位置を次の行の行始端に移動する書式制御文字。

ＮＵＬ：空　白（Null）　媒体の空き又は時間関係の空きを埋める特殊制御文字。

備　考　情報内容に影響を与えることなく、加えたり除いたりできるが、情報配置又は装置制御に影響を与えることがある（ＪＩＳ Ｃ 6220）。

ＲＢＳ：初期状態復帰（Reset to Basic State）　あらかじめ定められた初期状態に復帰することを指定する特殊制御文字。

ＳＣＵ：組版単位モード選択（Select Composition Unit Mode）組版単位モードを設定する特殊制御文字。

備　考　1.組版単位モードの内容は、字方向（第1パラメタ）及び行方向（第2パラメタ）の二つの選択パラメタで次のように指定する。

0：文　字

1：タイポグラヒック

2：ポイント法

3：ＤＩＤＯＴ

4：ミ　ル

5：メートル法

6：Ｕ

7：Ｑ

8：$\frac{1}{10}$ 文字

2.第2パラメタを省略した場合は、行方向の単位は字方向の単位と同じとみなす。

3.この指定は、次のＳＣＵが現れるまで有効である。

ＳＯＨ：ヘッディング開始（Start of Heading）　ヘッディングの最初の文字として用いる伝送制御文字（ＪＩＳ Ｃ 6220）

以下にヘッディング情報が続く。

ＳＴＸ：テキスト開始（Start of Text）　テキストの最初の文字として用い、かつ、テキストに先行するヘッディングがある場合には、その終結にも用いる伝送制御文字（ＪＩＳ Ｃ 6220）。

ＳＡＰ：動作位置指定（Set Active Position）　動作位置を指定する書式制御文字。

備　考　1.動作位置は、字方向（第1パラメタ）、行方向（第2パラメタ）の二つの数値パラメタによって第1行の始端からの絶対位置で指定する。

2.パラメタの値の単位は、組版単位モードによる。

3.第2パラメタを省略した場合は、行位置は変わらない。

Ｃ1　：カウント情報1（Count Information-1）　レコードの総数。

Ｃ2　：カウント情報2（Count Information-2）　文字（漢字、英数字、記号）の総数。

Ｃ3　：カウント情報3（Count Information-3）　制御文字の総数。

備　考　Ｃ1は、フロッピーディスク・ファイルのレコード（セクタ）数に有効で、紙テープ・ファイルには無効。

Ｃ1、Ｃ2、Ｃ3はＥＴＸレコードに存在し、Ｃ2、Ｃ3は紙テープファイルの場合、ＥＴＸコードの後に存在する。

ＰＧＳ：頁サイズ指定（Page Size）出力用紙がロール紙のときの頁サイズを指定する書式制御文字。

備　考　1.頁サイズは数値パラメータで指定する。

2.パラメータの値の単位は、組版単位モードによる。

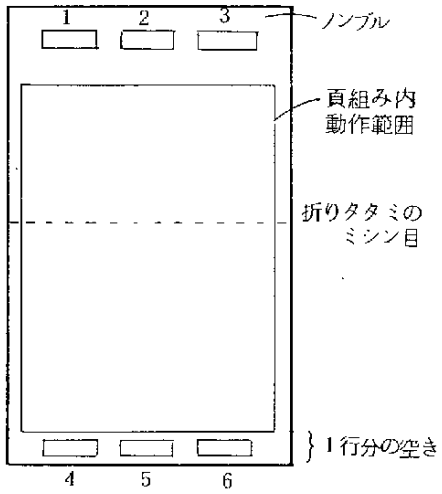
3.この指定は、次のＰＧＳが現われるまで有効である。

ＰＧ　：ノンブル（Paging）　頁組みに対し、指定位置に頁番号をふる制御文字。

備　考　1.ＳＡＰによって頁組みされた動作位置範囲外に頁番号が、ふれる領域があった場合に有効である。

1行分の空きの場合はその行に、2行分以上の空きの場合は、ＬＰ指示の2倍をとった領域にふる。（ノンブル値が1～3の場合）

またノンブルの値が上記以外の4～6の場合は、無条件に1行分の空きにふる。



2.ノンブルの指定場所は頁組みの上下3箇所づつ（左・中央・右）の6箇所である。

パラメータ1：頁の上方左側

2：　〃　中央

3：　〃　右側

4：頁の下方左側

5：　〃　中央

6：　〃　右側

0：ノンブル無し

3.頁番号は両端をハイフンでくくり、ゼロ・サプレスを行なう。

－×××××－

（例）　－1－

－10－

ＳＰ(1)：間　隔（Space）　動作位置を同一行で1文字分前進させる書式制御文字（ＪＩＳ　Ｃ　6220）。

注　(1)　1.ＳＰは、制御文字とも印字しない図形文字ともみなされる。

2.分類は、ＪＩＳ　Ｃ　6220 では特殊制御文字であるが、ここでは、書式制御文字とする。

使用法　ＣＥＸ　ＳＰ

1Ｃ　20（16進）

KSP(2)：漢字間隔（Kanji Space）　動作位置を同一行で1文字分前進させる2バイト符号の書式制御文字。

注　(2)　1.この制御文字は、ＪＩＳ　Ｃ　6220 における間隔（ＳＰ）と同じ働きをするが、シフトイン（ＳＩ）の状態においてだけ有効である。

2.ＫＳＰは、制御文字とも印字しない図形文字ともみなされる。

漢字符号系図形文字（JIS C 6226）の位置1－1と同様

2121（16進）

CEX：制御文字拡張

機能識別用制御文字を表現する拡張符号。

拡張表現の一般形　漢字符号系制御文字の拡張は、次の文字列で表す。

CEX　X　P_1……P_n

CEX：制御文字拡張（1/12）

X　：機能識別用制御文字（2/1 ～ 7/14）

P_1…P_n：パラメタ列

（X……CP、LP、SCU、SAP、NEL）

〔パラメタの表現〕

(1) パラメタの種類　数値パラメタは、パラメタが数値（量）を指定するもので、パラメタの個数と順番は制御文字によって一義的に決まっている。各パラメタの長さは、必要な数値に応じて可変長である。

選択パラメタは、制御文字の機能識別用制御文字によって決まる機能を更に詳細に指定するもので、機能の定義における番号を指定する。相反しない内容のものを二つ以上並記して同時に指定することができる。例えば、領域指定（AS）において、マニュアル入力領域を英数字入力、終端そろえとしたいときは、パラメタ0（非保護領域）、3（数字入力領域）、4（英字入力領域）、5（終端そろえ）を同時に並記すればよい。

なお、相反する内容のものを並記した場合、例えば、上記ASにおけるパラメタ1〔保護領域（送出不可）〕と8〔保護領域（送出可）〕を指定した場合、誤りとするか、後から来たパラメタを有効とするかは、ここでは規定しない。

(2) パラメタの基数　パラメタの基数については、2進・10進の併用案も検討したが、2進表現によるビット使用効率の向上効果が現れる大きい数値の使用頻度は比較的少ないと考えられたので、表現及び入力との対応が簡明な10進法によることとした。

(3) パラメタバイト　パラメタバイトの各ビットは、次の意味を持つ。

(a_8)	a_7	a_6	a_5	a_4 a_3 a_2 a_1
‖	‖	‖	‖	
常に0	0：中間 1：区切り又は終端	常に1	0：中間又は区切 1：終端	10進数（0～9）を表現

すなわち、中間パラメタバイト　：　2/0 ～ 2/9

区切りパラメタバイト　：　6/0 ～ 6/9

終端パラメタバイト　：　7/0 ～ 7/9

を使用する。パラメタバイトとして上記以外の文字を使用した場合は誤りとするが、その処

置についてはここでは規定しない。

なお、パラメタの区切り及び終端については、特定の文字を設ける案も検討したが、表現の効率化を図るため上記方法とした。

（例）　文字ピッチ（ＣＰ、数値パラメータ＝４０の場合）

CEX	CP	4	0
１Ｃ	２２	２４	７０（１６進）

5. 行頭・行末禁則文字

表３　行頭禁則文字表

JIS漢字コード(16進)	記号	意　味
２１２２	、	読点
２１２３	。	区点
２１２４	，	コンマ
２１２５	．	ピリオド
２１２９	？	疑問符
２１２Ａ	！	感嘆符
２１２Ｂ	゛	濁点
２１２Ｃ	゜	半濁点
２１３３	ヽ	片仮名繰返し記号
２１３４	ヾ	〃　（濁点）
２１３５	ゝ	平仮名　〃
２１３６	ゞ	〃　（濁点）
２１３９	々	繰返し記号
２１４７	’	右シングル引用符
２１４９	”	右ダブル引用符
２１４Ｂ	）	終り小括弧
２１４Ｄ	〕	終りきっこう括弧
２１４Ｆ	］	終り大括弧
２１５１	｝	終り中括弧
２１５３	＞	終り山括弧
２１５５	≫	終り二重山括弧
２１５７	」	終りかぎ括弧
２１５９	』	終り二重かぎ括弧
２１５Ｂ	】	終りすみ付き括弧
２１６Ｂ	°	度
２１６Ｃ	′	分
２１６Ｄ	″	秒

表３　行末禁則文字表

JIS漢字コード(16進)	記号	意　味
２１４６	‘	左シングル引用符
２１４８	“	右ダブル引用符
２１４Ａ	（	始め小括弧
２１４Ｃ	〔	始めきっこう括弧
２１４Ｅ	［	始め大括弧
２１５０	｛	始め中括弧
２１５２	＜	始め山括弧
２１５４	≪	始め二重括弧
２１５６	「	始めかぎ括弧
２１５８	『	始め二重かぎ括弧
２１５Ａ	【	始めすみ付き括弧

6. 禁則合成テーブル

表 4　　禁 則 合 成 表

禁則連続文字	ＪＩＳ漢字コード（１６進）	合成コード（１６進）
）、	２１４Ｂ２１２２	２２３０
）。	２１２３	２２３１
），	２１２４	２２３２
）．	２１２５	２２３３
）」	２１５７	２２３４
）』	２１５９	２２３５
」、	２１５７２１２２	２２３８
」。	２１２３	２２３９
」，	２１２４	２２３Ａ
」）	２１４Ｂ	２２３Ｂ
』、	２１５９２１２２	２２３Ｃ
』。	２１２３	２２３Ｄ
』，	２１２４	２２３Ｅ
』）	２１４Ｂ	２２３Ｆ
。）	２１２３２１４Ｂ	２２４０
。）	２１５７	２２４１
。』	２１５９	２２４２
＞、	２１５３２１２２	２２４３
＞。	２１２３	２２４４
＞，	２１２４	２２４５
≫、	２１５５２１２２	２２４６
≫。	２１２３	２２４７
≫，	２１２４	２２４８
’、	２１４７２１２２	２２４９
’。	２１２３	２２４Ａ
’）	２１４Ｂ	２２４Ｂ
’」	２１５７	２２４Ｃ
’』	２１５９	２２４Ｄ
”、	２１４９２１２２	２２４Ｅ
”。	２１２３	２２４Ｆ
”）	２１４Ｂ	２２５０
”」	２１５７	２２５１
”』	２１４９２１５９	２２５２
），	２１４Ｂ２１４７	２２３６
）”	２１４９	２２３７

7. 標準高頻度文字セット

- ひらがな …… ８３種
  (区点コード) ：０４０１～０４８１
- カタカナ …… ８６種
  (区点コード) ：０５０１～０５８６
- 数　字 …… １０種
  (区点コード) ：０３１６～０３２５
- 英　字 …… ５２種
  (区点コード) ：０３３３～０３５８
  ０３６５～０３９０
- 記　号 …… １９種
  (区点コード……記号)

０１０２	、
０１０３	。
０１０４	,
０１０５	.
０１０６	・
０１０７	:
０１０８	;
０１２８	―
０１３１	／
０１３３	～
０１３６	…
０１４２	(
０１４３	)
０１５４	「
０１５５	」
０１６０	＋
０１６１	－
０１６３	×
０１６５	＝

付記２　日本語端末用文字種・コード表

(1)　文字種

漢字	
当用漢字	１８５０字
昭和５２年１月改正案・追加分	８３
人名補正漢字	９２
改正追加分	２８
	２０５３字
重複	－　７
	２０４６字
ＪＩＰＤＥＣ追加分	１９６
	２２４２字
仮名	
ひらがな	８３
カタカナ	８６
英字	５２
数字	１０
記号	
ＪＩＳ漢字記号	１０８
追加分	１
総　計	２５８２字種

(2)　コード表

ＪＩＳ漢字コード表（ＪＩＳ　Ｃ　6226 情報交換用漢字符号系）に準処したコードを用いる。前記文字種は、すべてＪＩＳ漢字コードの第１水準漢字集合２９６５字に含まれている漢字と、記号１種を除けば４５３種の仮名英数字記号中に含まれている。

以下のコード表は、ＪＩＳ漢字コード表の区点コード索引によるものであり、表の見方は、区２桁と点左１桁・上１桁の順に参照する。なお、下線の引いてある文字記号は、前記２５８２字種に含まれていないものである。

区２桁、点２桁はともに１０進数表記であるが、これを２バイト／字のコードにするには、次の１６進数対応表を参照すれば容易に変換できる。

ex）「亜」の区点コード　……１６０１

〃　２バイト／字コード……１００１

２進表記……０００１０００００００００１

１０進数	１６進数	１０進数	１６進数	１０進数	１６進数
０１	０１	３３	２１	６５	４１
０２	０２	３４	２２	６６	４２
０３	０３	３５	２３	６７	４３
０４	０４	３６	２４	６８	４４
０５	０５	３７	２５	６９	４５
０６	０６	３８	２６	７０	４６
０７	０７	３９	２７	７１	４７
０８	０８	４０	２８	７２	４８
０９	０９	４１	２９	７３	４９
１０	０Ａ	４２	２Ａ	７４	４Ａ
１１	０Ｂ	４３	２Ｂ	７５	４Ｂ
１２	０Ｃ	４４	２Ｃ	７６	４Ｃ
１３	０Ｄ	４５	２Ｄ	７７	４Ｄ
１４	０Ｅ	４６	２Ｅ	７８	４Ｅ
１５	０Ｆ	４７	２Ｆ	７９	４Ｆ
１６	１０	４８	３０	８０	５０
１７	１１	４９	３１	８１	５１
１８	１２	５０	３２	８２	５２
１９	１３	５１	３３	８３	５３
２０	１４	５２	３４	８４	５４
２１	１５	５３	３５	８５	５５
２２	１６	５４	３６	８６	５６
２３	１７	５５	３７	８７	５７
２４	１８	５６	３８	８８	５８
２５	１９	５７	３９	８９	５９
２６	１Ａ	５８	３Ａ	９０	５Ａ
２７	１Ｂ	５９	３Ｂ	９１	５Ｂ
２８	１Ｃ	６０	３Ｃ	９２	５Ｃ
２９	１Ｄ	６１	３Ｄ	９３	５Ｄ
３０	１Ｅ	６２	３Ｅ	９４	５Ｅ
３１	１Ｆ	６３	３Ｆ		
３２	２０	６４	４０		

01区

点	0	1	2	3	4	5	6	7	8	9
0		(SP)	、	。	,	.	・	:	;	?
1	!	゛	゜	´	`	¨	^	￣	_	ヽ
2	ヾ	ゝ	ゞ	〃	仝	々	〆	〇	ー	―
3	‐	／	＼	～	∥	｜	…	‥	‘	’
4	“	”	（	）	〔	〕	［	］	｛	｝
5	〈	〉	《	》	「	」	『	』	【	】
6	＋	－	±	×	÷	＝	≠	＜	＞	≦
7	≧	∞	∴	♂	♀	°	′	″	℃	￥
8	$	¢	£	%	#	&	*	@	§	☆
9	★	○	●	◎	◇					

02区

点	0	1	2	3	4	5	6	7	8	9
0		◆	□	■	△	▲	▽	▼	※	〒
1	→	←	↑	↓	＝	㊍				
2										
3										
4										
5										
6										
7										
8										
9										

０３区

点	0	1	2	3	4	5	6	7	8	9
0										
1							0	1	2	3
2	4	5	6	7	8	9				
3				A	B	C	D	E	F	G
4	H	I	J	K	L	M	N	O	P	Q
5	R	S	T	U	V	W	X	Y	Z	
6						a	b	c	d	e
7	f	g	h	i	j	k	l	m	n	o
8	p	q	r	s	t	u	v	w	x	y
9	z									

０４区

点	0	1	2	3	4	5	6	7	8	9
0		ぁ	あ	ぃ	い	ぅ	う	ぇ	え	ぉ
1	お	か	が	き	ぎ	く	ぐ	け	げ	こ
2	ご	さ	ざ	し	じ	す	ず	せ	ぜ	そ
3	ぞ	た	だ	ぢ	ぢ	っ	つ	づ	て	で
4	と	ど	な	に	ぬ	ね	の	は	ば	ば
5	ひ	び	ぴ	ふ	ぶ	ぷ	へ	べ	ぺ	ほ
6	ぼ	ぼ	ま	み	む	め	も	ゃ	や	ゅ
7	ゆ	ょ	よ	ら	り	る	れ	ろ	ゎ	わ
8	ゐ	ゑ	を	ん						
9										

05区

点	0	1	2	3	4	5	6	7	8	9
0		ァ	ア	ィ	イ	ゥ	ウ	ェ	エ	ォ
1	オ	カ	ガ	キ	ギ	ク	グ	ケ	ゲ	コ
2	ゴ	サ	ザ	シ	ジ	ス	ズ	セ	ゼ	ソ
3	ゾ	タ	ダ	チ	ヂ	ッ	ツ	ヅ	テ	デ
4	ト	ド	ナ	ニ	ヌ	ネ	ノ	ハ	バ	パ
5	ヒ	ビ	ピ	フ	ブ	プ	ヘ	ベ	ペ	ホ
6	ボ	ポ	マ	ミ	ム	メ	モ	ャ	ヤ	ュ
7	ユ	ョ	ヨ	ラ	リ	ル	レ	ロ	ヮ	ワ
8	ヰ	ヱ	ヲ	ン	ヴ	ヵ	ヶ			
9										

06区

点	0	1	2	3	4	5	6	7	8	9
0		Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι
1	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ
2	Υ	Φ	Χ	Ψ	Ω					
3				α	β	γ	δ	ε	ζ	η
4	θ	ι	κ	λ	μ	ν	ξ	ο	π	ρ
5	σ	τ	υ	φ	χ	ψ	ω			
6										
7										
8										
9										

07区

点	0	1	2	3	4	5	6	7	8	9
0		А	Б	В	Г	Д	Е	Ё	Ж	З
1	И	Й	К	Л	М	Н	О	П	Р	С
2	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы
3	Ь	Э	Ю	Я						
4										а
5	б	в	г	д	е	ё	ж	з	и	й
6	к	х	м	н	о	п	р	с	т	у
7	ф	я	ц	ч	ш	щ	ъ	ы	ь	э
8	ю									
9										

16区

点	0	1	2	3	4	5	6	7	8	9
0		亜(ア)	唖	娃	阿	哀(アイ)	愛	挨	姶(あい)	逢 あ
1	葵(あおい)	茜(あかね)	穐(あき)	悪(アク)	握	渥	旭(あさひ)	葦(あし)	芦(あし)	鯵(あじ)
2	梓(あずさ)	圧(アツ)	斡	扱(あつかい)	宛(あて)	姐(あね)	虻(あぶ)	飴(あめ)	絢(あや)	綾
3	鮎(あゆ)	或(ある)	粟(あわ)	袷(あわせ)	安(アン)	庵	按	暗	案	闇
4	鞍(あんぎ)	杏	以(イ)	伊	位	依	偉	囲	夷	委 い
5	威	尉	惟	意	慰	易	椅	為	畏	異
6	移	維	緯	胃	萎	衣	謂	違	遺	医
7	井(い)	亥(い)	域(いき)	育(いく)	郁	磯(いそ)	一(イチ)	壱	溢(いつ)	逸
8	稲(いね)	茨(いばら)	芋(いも)	鰯(いわし)	允(イン)	印	咽	員	因	姻
9	引	飲	淫	胤	蔭					

17区

点	0	1	2	3	4	5	6	7	8	9
0		院	陰	隠	韻	吋(いんち)	右(ウ)	宇	烏	羽 う
1	迂	雨	卯(う)	鵜	窺(うかがい)	丑(うし)	碓(うす)	臼	渦	嘘
2	唄(うた)	鬱(ウツ)	蔚(うなぎ)	鰻(うなぎ)	姥(うば)	厩(うまや)	浦(うら)	瓜(うり)	閏(うるう)	噂(うわさ)
3	云(ウン)	運	雲	荏(え)	餌(エイ)	叡	営	嬰	影	映 え
4	曳	栄	永	泳	洩	瑛	盈	穎	頴	英
5	衛	詠	鋭	液(エキ)	疫	益	駅	悦(エツ)	謁	越
6	閲	榎(えのき)	厭(エン)	円	園	堰	奄	宴	延	怨
7	掩	援	沿	演	炎	焔	煙	燕	猿	縁
8	艶	苑	薗	遠	鉛	鴛	塩	於(お)	汚	甥(おい) お
9	凹(オウ)	央	奥	往	応					

18区

点	0	1	2	3	4	5	6	7	8	9
0		押	旺	横	欧	殴	王	翁	襖	鴬
1	鷗	黄	岡	沖	荻	億	屋	憶	臆	桶
2	牡	乙	俺	卸	恩	温	穏	音	下	化
3	仮	何	伽	価	佳	加	可	嘉	夏	嫁
4	家	寡	科	暇	果	架	歌	河	火	珂
5	禍	禾	稼	箇	花	苛	茄	荷	華	菓
6	蝦	課	嘩	貨	迦	過	霞	蚊	俄	峨
7	我	牙	画	臥	芽	蛾	賀	雅	餓	駕
8	介	会	解	回	塊	壊	廻	快	怪	悔
9	恢	懐	戒	拐	改					

19区

点	0	1	2	3	4	5	6	7	8	9
0		魁	晦	械	海	灰	界	皆	絵	芥
1	蟹	開	階	貝	凱	劾	外	咳	害	崖
2	慨	概	涯	碍	蓋	街	該	鎧	骸	浬
3	馨	蛙	垣	柿	蠣	鈎	劃	嚇	各	廓
4	拡	攪	格	核	殻	獲	確	穫	覚	角
5	赫	較	郭	閣	隔	革	学	岳	楽	額
6	顎	掛	笠	樫	橿	梶	鰍	潟	割	喝
7	恰	括	活	渇	滑	葛	褐	轄	且	鰹
8	叶	椛	樺	鞄	株	兜	竈	蒲	釜	鎌
9	噛	鴨	栢	茅	萱					

20区

点	0	1	2	3	4	5	6	7	8	9
0		粥(かゆ)	刈(かり)	苅←	瓦(かわら)	乾(カン)	侃	冠	寒	刊
1	勘	勧	巻	喚	堪	姦	完	官	寛	干
2	幹	患	感	慣	憾	換	敢	柑	桓	棺
3	款	歓	汗	漢	澗	灌	環	甘	監	看
4	竿	管	簡	緩	缶	翰	肝	艦	莞	観
5	諫	貫	還	鑑	間	閑	関	陥	韓	館
6	舘	丸(ガン)	含	岸	巌	玩	癌	眼	岩	翫
7	贋	雁	頑	顔	願	企(キ)	伎	危	喜	器 き
8	基	奇	嬉	寄	岐	希	幾	忌	揮	机
9	旗	既	期	棋	棄					

21区

点	0	1	2	3	4	5	6	7	8	9
0	、	機	帰	毅	気	汽	畿	祈	季	稀
1	紀	徽	規	記	貴	起	軌	輝	飢	騎
2	鬼	亀	偽(ギ)	儀	妓	宜	戯	技	擬	欺
3	犠	疑	祇	義	蟻	誼	議	掬(キク)	菊	鞠
4	吉(キチ)	吃(キツ)	喫	桔	橘	詰	砧(きぬた)	杵(きね)	黍(きび)	却(キャク)
5	客	脚	虐(ギャク)	逆	丘(キュウ)	久	仇	休	及	吸
6	宮	弓	急	救	朽	求	汲	泣	灸	球
7	究	窮	笈	級	糾	給	旧	牛(ギュウ)	去(キョ)	居
8	巨	拒	拠	挙	渠	虚	許	距	鋸	漁(ギョ)
9	禦	魚	亨(キョウ)	享	京					

22区

点	0	1	2	3	4	5	6	7	8	9
0		供	俠	僑	兇	競	共	凶	協	匡
1	卿	叫	喬	境	峡	強	彊	怯	恐	恭
2	挟	教	橋	況	狂	狭	矯	胸	脅	興
3	蕎	郷	鏡	響	饗	驚	仰(ギョウ)	凝	尭	暁
4	業(キョク)	局	曲	極	玉(ギョク)	桐	粁(きりめ・とる)	僅(キン)	勤	均
5	巾	錦	斤	欣	欽	琴	禁	禽	筋	緊
6	芹	菌	衿	襟	謹	近	金	吟(ギン)	銀	九(ク)く
7	倶	句	区	狗	玖	矩	苦	躯	駆	駈
8	駒	具(グ)	愚	虞	喰(くい)	空(クウ)	偶(グウ)	寓	遇	隅
9	串(くし)	櫛	釧(くしろ)	屑(くず)	屈(クツ)					

23区

点	0	1	2	3	4	5	6	7	8	9
0		掘	窟	沓(くつ)	靴	轡(くつわ)	窪(くぼ)	熊(くま)	隈	粂(くめ)
1	栗(くり)	繰	桑(くわ)	鍬	勲(クン)	君	薫	訓	群(グン)	軍
2	郡	卦(ケ)	袈	祁	係(ケイ)	傾	刑	兄	啓	圭け
3	珪	型	契	形	径	恵	慶	慧	憩	掲
4	携	敬	景	桂	渓	畦	稽	系	経	継
5	繋	罫	茎	荊	蛍	計	詣	警	軽	頚
6	鶏	芸(ゲイ)	迎	鯨	劇(ゲキ)	戟	撃	激	隙(けた)	桁
7	傑(ケツ)	欠	決	潔	穴	結	血	訣	月(ゲツ)	件(ケン)
8	倹	倦	健	兼	券	剣	喧	圏	堅	嫌
9	建	憲	懸	拳	捲					

24区

点	0	1	2	3	4	5	6	7	8	9
0		検	権	牽	犬	献	研	硯	絹	県
1	肩	見	謙	賢	軒	遣	鍵	険	顕	験
2	鹸	元(ゲン)	原	厳	幻	弦	減	源	玄	現
3	絃	舷	言	諺	限	乎(コ)	個	古	呼	固
4	姑	孤	己	庫	弧	戸	故	枯	湖	狐
5	糊	袴	股	胡	菰	虎	誇	跨	鈷	雇
6	顧	鼓	五(ゴ)	互	伍	午	呉	吾	娯	後
7	御	悟	梧	檎	瑚	碁	語	誤	護	醐
8	乞(こい)	鯉	交(コウ)	佼	侯	候	倖	光	公	功
9	効	勾	厚	口	向					

25区

点	0	1	2	3	4	5	6	7	8	9
0		后	喉	坑	垢	好	孔	孝	宏	工
1	巧	巷	幸	広	庚	康	弘	恒	慌	抗
2	拘	控	攻	昂	晃	更	杭	校	梗	構
3	江	洪	浩	港	溝	甲	皇	硬	稿	糠
4	紅	紘	絞	綱	耕	考	肯	肱	腔	膏
5	航	荒	行	衡	講	貢	購	郊	酵	鉱
6	砿	鋼	閤	降	項	香	高	鴻	剛(ゴウ)	劫
7	号	合	壕	拷	濠	豪	轟	麹(こうじ)	克(コク)	刻
8	告	国	穀	酷	鵠	黒	獄(ゴク)	漉(こし)	腰	餓(こしき)
9	忽(コツ)	惚	骨	狛(こま)	込(こみ)					

26区

点	0	1	2	3	4	5	6	7	8	9
0		此	頃	今	困	坤	墾	婚	恨	懇
1	昏	昆	根	梱	混	痕	紺	艮	魂	些 さ
2	佐	叉	唆	嵯	左	差	査	沙	瑳	砂
3	詐	鎖	裟	坐	座	挫	債	催	再	最
4	哉	塞	妻	宰	彩	才	採	栽	歳	済
5	災	采	犀	砕	砦	祭	斎	細	菜	裁
6	載	際	剤	在	材	罪	財	冴	坂	阪
7	堺	榊	肴	咲	崎	埼	碕	鷺	作	削
8	咋	搾	昨	朔	柵	窄	策	索	錯	桜
9	鮭	笹	匙	冊	刷					

27区

点	0	1	2	3	4	5	6	7	8	9
0		察	拶	撮	擦	札	殺	薩	雑	皐
1	鯖	捌	錆	鮫	皿	晒	三	傘	参	山
2	惨	撒	散	桟	燦	珊	産	算	纂	蚕
3	讃	賛	酸	餐	斬	暫	残	仕	仔	伺 し
4	使	刺	司	史	嗣	四	士	始	姉	姿
5	子	屍	市	師	志	思	指	支	孜	斯
6	施	旨	枝	止	死	氏	獅	祉	私	糸
7	紙	紫	肢	脂	至	視	詞	詩	試	誌
8	諮	資	賜	雌	飼	歯	事	似	侍	児
9	字	寺	慈	持	時					

28区

点	0	1	2	3	4	5	6	7	8	9
0		次	滋	治	爾	璽	痔	磁	示	而
1	耳	自	蒔	辞	汐	鹿	式	識	鳴	竺
2	軸	宍	雫	七	叱	執	失	嫉	室	悉
3	湿	漆	疾	質	実	蔀	篠	偲	柴	芝
4	屢	蘂	縞	舎	写	射	捨	赦	斜	煮
5	社	紗	者	謝	車	遮	蛇	邪	借	勺
6	尺	杓	灼	爵	酌	釈	錫	若	寂	弱
7	惹	主	取	守	手	朱	殊	狩	珠	種
8	腫	趣	酒	首	儒	受	呪	寿	授	樹
9	綬	需	囚	収	周					

29区

点	0	1	2	3	4	5	6	7	8	9
0		宗	就	州	修	愁	拾	洲	秀	秋
1	終	繍	習	臭	舟	蒐	衆	襲	讐	蹴
2	輯	週	酋	酬	集	醜	什	住	充	十
3	従	戎	柔	汁	渋	獣	縦	重	銃	叔
4	夙	宿	淑	祝	縮	粛	塾	熟	出	術
5	述	俊	峻	春	瞬	竣	舜	駿	准	循
6	旬	楯	殉	淳	準	潤	盾	純	巡	遵
7	醇	順	処	初	所	暑	曙	渚	庶	緒
8	署	書	薯	藷	諸	助	叙	女	序	徐
9	恕	鋤	除	傷	償					

30区

点	0	1	2	3	4	5	6	7	8	9
0		勝	匠	升	召	哨	商	唱	嘗	奨
1	妾	娼	宵	将	小	少	尚	庄	床	廠
2	彰	承	抄	招	掌	捷	昇	昌	昭	晶
3	松	梢	樟	樵	沼	消	渉	湘	焼	焦
4	照	症	省	硝	礁	祥	称	章	笑	粧
5	紹	肖	菖	蒋	蕉	衝	裳	訟	証	詔
6	詳	象	賞	醤	鉦	鍾	鐘	障	鞘	上(ジョウ)
7	丈	丞	乗	冗	剰	城	場	壌	嬢	常
8	情	擾	条	杖	浄	状	畳	穣	蒸	譲
9	醸	錠	嘱(ショク)	埴	飾					

31区

点	0	1	2	3	4	5	6	7	8	9
0		拭	植	殖	燭	織	職	色	触	食
1	蝕	辱(ジョク)	尻(しり)	伸(シン)	信	侵	唇	娠	寝	審
2	心	慎	振	新	晋	森	榛	浸	深	申
3	疹	真	神	秦	紳	臣	芯	薪	親	診
4	身	辛	進	針	震	人(ジン)	仁	刃	塵	壬
5	尋	甚	尽	腎	訊	迅	陣	靱	笥(ス)	諏 す
6	須	酢(す)	図(ズ)	厨	逗	吹(スイ)	垂	帥	推	水
7	炊	睡	粋	翠	衰	遂	酔	錐	錘	随(ズイ)
8	瑞	髄	崇(スウ)	嵩	数	枢	趨	雛	据(すえ)	杉(すぎ)
9	椙(すげ)	菅(すこぶる)	頗(すすめ)	雀(すず)	裾(すそ)					

32区

点	0	1	2	3	4	5	6	7	8	9
0		澄	摺	寸	世	瀬	畝	是	凄	制
1	勢	姓	征	性	成	政	整	星	晴	棲
2	栖	正	清	牲	生	盛	精	聖	声	製
3	西	誠	誓	請	逝	醒	青	静	斉	税
4	脆	隻	席	惜	戚	斥	昔	析	石	積
5	籍	績	脊	責	赤	跡	蹟	碩	切	拙
6	接	摂	折	設	窃	節	説	雪	絶	舌
7	蝉	仙	先	千	占	宣	専	尖	川	戦
8	扇	撰	栓	栴	泉	浅	洗	染	潜	煎
9	煽	旋	穿	箭	線					

33区

点	0	1	2	3	4	5	6	7	8	9
0		繊	羨	腺	舛	船	薦	詮	賤	践
1	選	遷	銭	銑	閃	鮮	前	善	漸	然
2	全	禅	繕	膳	糎	噌	塑	岨	措	曾	そ
3	曽	楚	狙	疏	疎	礎	祖	租	粗	素	
4	組	蘇	訴	阻	遡	鼠	僧	創	双	叢	
5	倉	喪	壮	奏	爽	宋	層	匝	惣	想	
6	捜	掃	挿	掻	操	早	曹	巣	槍	槽	
7	漕	燥	争	痩	相	窓	糟	総	綜	聡	
8	草	荘	葬	蒼	藻	装	走	送	遭	鎗	
9	霜	騒	像	増	憎					

34区

点	0	1	2	3	4	5	6	7	8	9
0		臓	蔵	贈	造	促(ソク)	側	則	即	息
1	捉	束	測	足	速	俗(ゾク)	属	賊	族	続
2	卒(ソツ)	袖(そで)	其(それ)	揃(そろい)	存(ゾン)	孫	尊	損	村	遜
3	他(タ)	多	太	汰	詑	唾(タ)	堕	妥	惰	打(だ)
4	柁	舵	楕	陀	駄	騨	体(タイ)	堆	対	耐
5	岱	帯	待	怠	態	戴	替	泰	滞	胎
6	腿	苔	袋	貸	退	逮	隊	黛	鯛(たい)	代(ダイ)
7	台	大	第	醍	題	鷹(たか)	滝(たき)	瀧	卓(タク)	啄
8	宅	托	択	拓	沢	濯	琢	託	鐸	濁(だく)
9	諾	茸(たけ)	凧(たこ)	蛸	只(ただ)					

35区

点	0	1	2	3	4	5	6	7	8	9
0		叩(たたき)	但(ただし)	達(タツ)	辰(たつ)	奪(ダツ)	脱	巽(たつみ)	竪(たて)	辿(たどり)
1	棚(たな)	谷(たに)	狸(たぬき)	鱈(たら)	樽(たる)	誰(だれ)	丹(タン)	単	嘆	坦
2	担	探	旦	歎	淡	湛	炭	短	端	箪
3	綻	耽	胆	蛋	誕	鍛	団(ダン)	壇	弾	断
4	暖	檀	段	男	談	値(チ)	知	地	弛	恥(ち)
5	智	池	痴	稚	置	致	蜘	遅	馳	築(チク)
6	畜	竹	筑	蓄	逐	秩(チツ)	窒	茶(チャ)	嫡(チャク)	着
7	中(チュウ)	仲	宙	忠	抽	昼	柱	注	虫	衷
8	註	酎	鋳	駐	樗(チョ)	瀦	猪	苧	著	貯
9	丁(チョウ)	兆	凋	喋	寵					

36区

点	0	1	2	3	4	5	6	7	8	9
0		帖	帳	庁	弔	張	彫	徴	懲	挑
1	暢	朝	潮	牒	町	眺	聴	脹	腸	蝶
2	調	諜	超	跳	銚	長	頂	鳥	勅(チョク)	捗
3	直	朕(チン)	沈	珍	賃	鎮	陳	津(つ)	墜(ツイ)	椎 つ
4	槌	追	鎚	痛(ツウ)	通	塚(つか)	栂(つが)	摑(つかみ)	槻(つき)	佃(つくだ)
5	漬(つけ)	柘(つげ)	辻(つじ)	蔦(つた)	綴(つづり)	鍔(つば)	椿(つばき)	潰(つぶれ)	坪(つぼ)	壺
6	嬬(つま)	紬(つむぎ)	爪(つめ)	吊(つり)	釣	鶴(つる)	亭(テイ)	低	停	偵 て
7	剃	貞	呈	堤	定	帝	底	庭	廷	弟
8	悌	抵	挺	提	梯	汀	碇	禎	程	締
9	艇	訂	諦	蹄	逓					

37区

点	0	1	2	3	4	5	6	7	8	9
0		邸	鄭	釘	鼎	泥(デイ)	摘(テキ)	擢	敵	滴
1	的	笛	適	鏑	溺(デキ)	哲(テツ)	徹	撤	轍	迭
2	鉄	典(テン)	塡	天	展	店	添	纏	甜	貼
3	転	顚	点	伝(デン)	殿	澱	田	電	兎(ト)	吐 と
4	堵	塗	妬	屠	徒	斗	杜	渡	登	菟
5	賭	途	都	鍍(ど)	砥	礪(ド)	努	度	土	奴
6	怒	倒(トウ)	党	冬	凍	刀	唐	塔	塘	套
7	宕	島	嶋	悼	投	搭	東	桃	樋	棟
8	盗	淘	湯	涛	灯	燈	当	痘	禱	等
9	答	筒	糖	統	到					

38区

点	0	1	2	3	4	5	6	7	8	9
0		董	蕩	藤	討	謄	豆	踏	逃	透
1	鐙	陶	頭	騰	闘	働	動	同	堂	導
2	憧	撞	洞	瞳	童	胴	萄	道	銅	峠
3	鴇	匿	得	徳	瀆	特	督	禿	篤	毒
4	独	読	栃	橡	凸	突	椴	届	鳶	苫
5	寅	酉	瀞	噸	屯	惇	敦	沌	豚	遁
6	頓	呑	曇	鈍	奈	那	内	乍	凪	薙
7	謎	灘	捺	鍋	楢	馴	縄	畷	南	楠
8	軟	難	汝	二	尼	弐	邇	匂	賑	肉
9	虹	廿	日	乳	入					

な に

39区

点	0	1	2	3	4	5	6	7	8	9
0		如	尿	韮	任	妊	忍	認	濡	禰
1	祢	寧	葱	猫	熱	年	念	捻	撚	燃
2	粘	乃	廼	之	埜	嚢	悩	濃	納	能
3	脳	膿	農	覗	蚤	巴	把	播	覇	杷
4	波	派	琶	破	婆	罵	芭	馬	俳	廃
5	拝	排	敗	杯	盃	牌	背	肺	輩	配
6	倍	培	媒	梅	楳	煤	狽	買	売	賠
7	陪	這	蝿	秤	矧	萩	伯	剝	博	拍
8	柏	泊	白	箔	粕	舶	薄	迫	曝	漠
9	爆	縛	莫	駁	麦					

ぬ・ね

の は

40区

点	0	1	2	3	4	5	6	7	8	9
0		函	箱	硲	箸	肇	筈	櫨	幡	肌
1	畑	畠	八	鉢	溌	発	醗	髪	伐	罰
2	抜	筏	閥	鳩	噺	塙	蛤	隼	伴	判
3	半	反	叛	帆	搬	斑	板	氾	汎	版
4	犯	班	畔	繁	般	藩	販	範	釆	煩
5	頒	飯	挽	晩	番	盤	磐	蕃	蛮	匪 ひ
6	卑	否	妃	庇	彼	悲	扉	批	披	斐
7	比	泌	疲	皮	碑	秘	緋	罷	肥	被
8	誹	費	避	非	飛	樋	簸	備	尾	微
9	枇	毘	琵	眉	美					

41区

点	0	1	2	3	4	5	6	7	8	9
0		鼻	柊	稗	匹	疋	髭	彦	膝	菱
1	肘	弼	必	畢	筆	逼	檜	姫	媛	紐
2	百	謬	俵	彪	標	氷	漂	瓢	票	表
3	評	豹	廟	描	病	秒	苗	錨	鋲	蒜
4	蛭	鰭	品	彬	斌	浜	瀕	貧	賓	頻
5	敏	瓶	不	付	埠	夫	婦	富	冨	布 ふ
6	府	怖	扶	敷	斧	普	浮	父	符	腐
7	膚	芙	譜	負	賦	赴	阜	附	侮	撫
8	武	舞	葡	蕪	部	封	楓	風	葺	蕗
9	伏	副	復	幅	服					

42区

点	0	1	2	3	4	5	6	7	8	9
0		福	腹	複	覆	淵	弗	払	沸	仏
1	物	鮒	分	吻	噴	墳	憤	扮	焚	奮
2	粉	糞	紛	雰	文	聞	丙	併	兵	塀
3	幣	平	弊	柄	並	蔽	閉	陛	米	頁
4	僻	壁	癖	碧	別	瞥	蔑	箆	偏	変
5	片	篇	編	辺	返	遍	便	勉	娩	弁
6	鞭	保	舗	鋪	圃	捕	歩	甫	補	輔
7	穂	募	墓	慕	戊	暮	母	簿	菩	倣
8	俸	包	呆	報	奉	宝	峰	峯	崩	庖
9	抱	捧	放	方	朋					

43区

点	0	1	2	3	4	5	6	7	8	9
0		法	泡	烹	砲	縫	胞	芳	萌	蓬
1	蜂	褒	訪	豊	邦	鋒	飽	鳳	鵬	乏
2	亡	傍	剖	坊	妨	帽	忘	忙	房	暴
3	望	某	棒	冒	紡	肪	膨	謀	貌	貿
4	鉾	防	吠	頬	北	僕	卜	墨	撲	朴
5	牧	睦	穆	釦	勃	没	殆	堀	幌	奔
6	本	翻	凡	盆	摩	磨	魔	麻	埋	妹
7	昧	枚	毎	哩	槙	幕	膜	枕	鮪	柾
8	鱒	桝	亦	俣	又	抹	末	沫	迄	儘
9	繭	麿	万	慢	満					

44区

点	0	1	2	3	4	5	6	7	8	9
0		漫	蔓	味	未	魅	巳	箕	岬	密
1	蜜	湊	蓑	稔	脈	妙	粍	民	眠	務
2	夢	無	牟	矛	霧	鵡	椋	婿	娘	冥
3	名	命	明	盟	迷	銘	鳴	姪	牝	滅
4	免	棉	綿	緬	面	麺	摸	模	茂	妄
5	孟	毛	猛	盲	網	耗	蒙	儲	木	黙
6	目	杢	勿	餅	尤	戻	籾	貰	問	悶
7	紋	門	匁	也	冶	夜	爺	耶	野	弥
8	矢	厄	役	約	薬	訳	躍	靖	柳	藪
9	鑓	愉	愈	油	癒					

45区

点	0	1	2	3	4	5	6	7	8	9
0		諭	輸	唯	佑	優	勇	友	宥	幽
1	悠	憂	揖	有	柚	湧	涌	猶	猷	由
2	祐	裕	誘	遊	邑	郵	雄	融	夕	予
3	余	与	誉	輿	預	傭	幼	妖	容	庸
4	揚	揺	擁	曜	楊	様	洋	溶	熔	用
5	窯	羊	耀	葉	蓉	要	謡	踊	遙	陽
6	養	慾	抑	欲	沃	浴	翌	翼	淀	羅
7	螺	裸	来	萊	頼	雷	洛	絡	落	酪
8	乱	卵	嵐	欄	濫	藍	蘭	覧	利	吏
9	履	李	梨	理	璃					

46区

点	0	1	2	3	4	5	6	7	8	9
0		痢	裏	裡	里	離	陸(リク)	律(リツ)	率	立
1	葎	掠(リャク)	略	劉(リュウ)	流	溜	琉	留	硫	粒
2	隆	竜	龍	侶(リョ)	慮	旅	虜	了(リョウ)	亮	僚
3	両	凌	寮	料	梁	涼	猟	療	瞭	稜
4	糧	良	諒	遼	量	陵	領	力(リョク)	緑	倫(リン)
5	厘	林	淋	燐	琳	臨	輪	隣	鱗	麟
6	瑠(ル)	塁(ルイ)	涙	累	類	令(レイ)	伶	例	冷	励る・れ
7	嶺	怜	玲	礼	苓	鈴	隷	零	霊	麗
8	齢	暦(レキ)	歴	列(レツ)	劣	烈	裂	廉(レン)	恋	憐
9	漣	煉	簾	練	聯					

47区

点	0	1	2	3	4	5	6	7	8	9
0		蓮	連	錬	呂(ロ)	魯	櫓	炉	賂	路ろ
1	露	労(ロウ)	婁	廊	弄	朗	楼	榔	浪	漏
2	牢	狼	籠	老	聾	蠟	郎	六(ロク)	麓	禄
3	肋	録	論(ロン)	倭(ワ)	和	話	歪	賄(わい)	脇	惑(ワク)わ
4	枠(わく)	鷲(わし)	亙(わたり)	亘	鰐(わに)	詫(わび)	藁(わら)	蕨(わらび)	椀(ワン)	湾
5	碗	腕								
6										
7										
8										
9										

(3) 漢 字 区 分 表

区　　分　　①	区分コード	字　　種
ＪＩＳ第１水準漢字集合	１０	２９６５
ＪＩＳ第２水準漢字集合	２０	３３８４
ＪＩＳ英数・仮名・記号	３０	４５３
その他文字記号	９０	

区　　分　　②	区分コード	字　　種
当用漢字　教育漢字	１０	８８１
教育外漢字	１１	９３６（９６９－３３）
人名用漢字	１２	８６（９２－６）
改正追加漢字（51.7）	１３	２７（２８－１）
当用漢字改正案（52.1）追加漢字	１４	７６（８３－７）
削除漢字	１５	３３
〔12〕人名と〔14〕追加の重複漢字	１６	６
〔13〕人名・追加と〔14〕追加の重複漢字	１７	１
ＪＩＰＤＥＣ用追加漢字	１８	１９６
その他漢字	１９	４１０７（７２３＋３３８４）
仮名文字　ひらがな	２０	
（異字体）	２１	
カタカナ	２５	
（異字体）	２６	
欧米文字　英　　字	３０	
小文字	３１	
（異字体）	３２	
ギリシャ文字	４０	
小文字	４１	
（異字体）	４２	
ロシア文字	５０	
小文字	５１	
（異字体）	５２	
その他文字	６０	
数　　字　アラビア数字（算用数字）	７０	
その他数字	７１	
記　　号	８８	

(4) 読 区 分 表

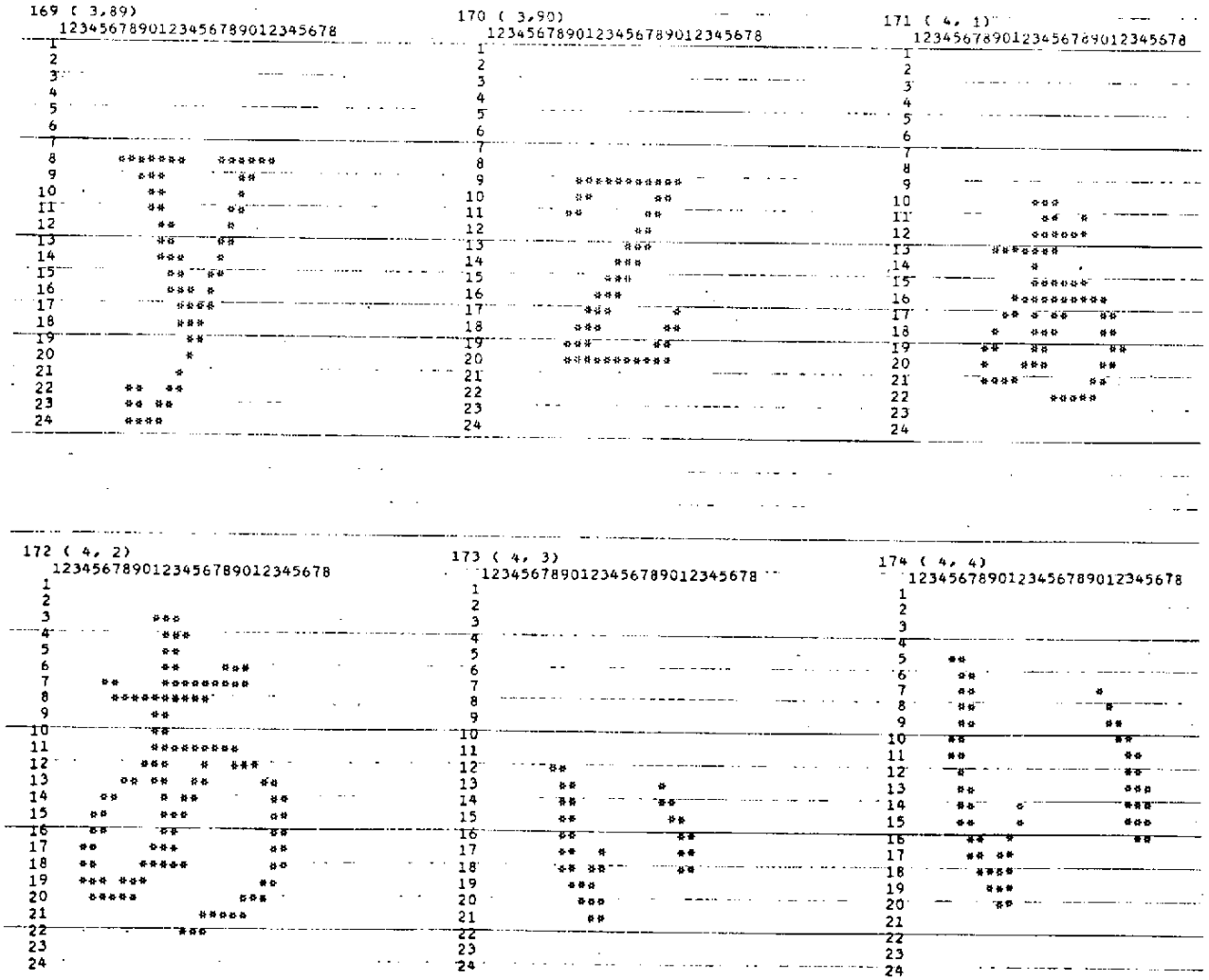
区　分	対　象　音　訓　読
００	当用漢字音訓表にある音読
０１	当用漢字音訓表にある訓読
１０	当用漢字音訓表にない音読 表外漢字の音読
１１	当用漢字音訓表にない訓読 表外漢字の訓読

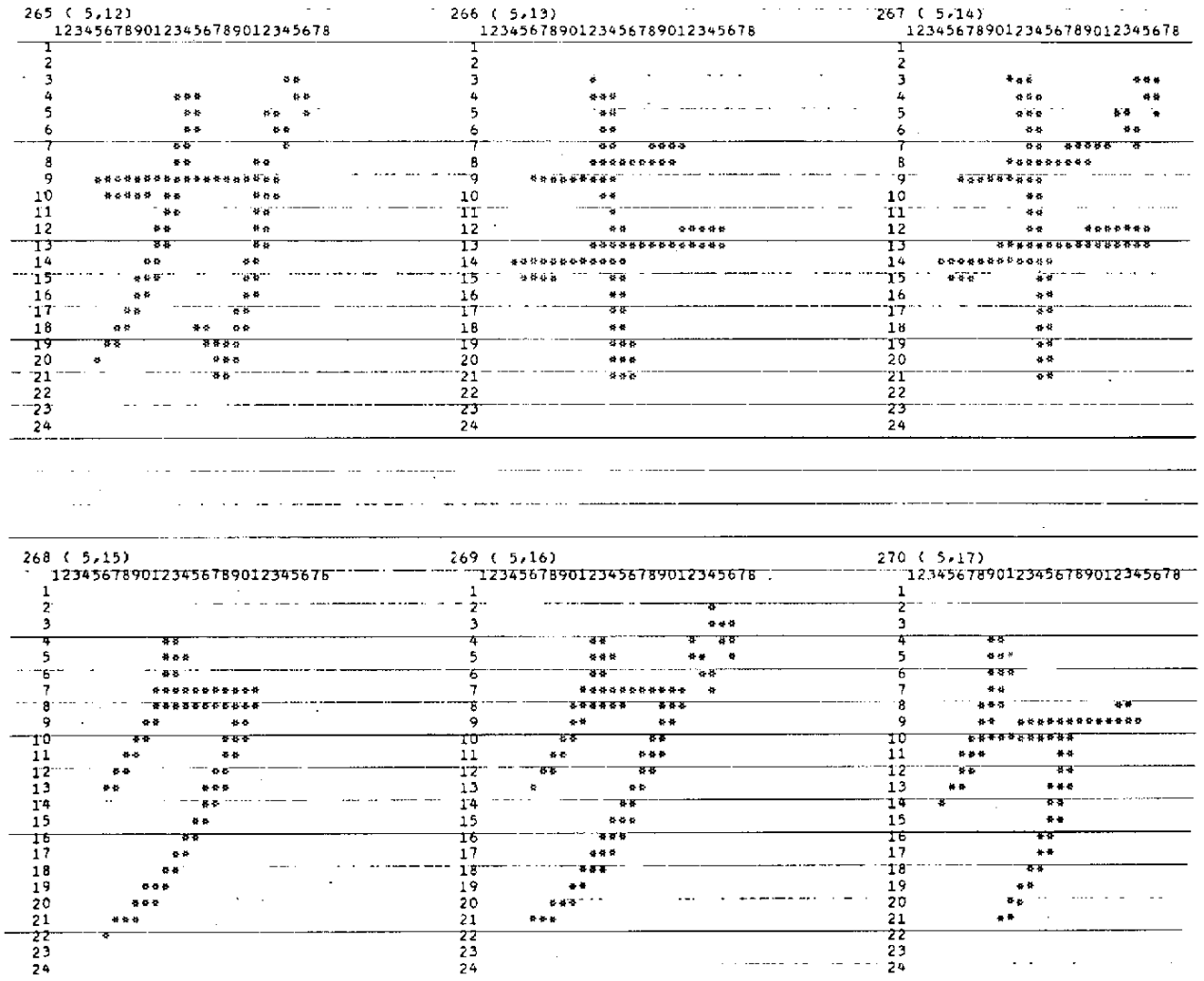
付記３　文字パターン出力例

(1) ２４×２４ドット・マトリックス・パターンのラインプリンタ出力リスト例(Ａ).

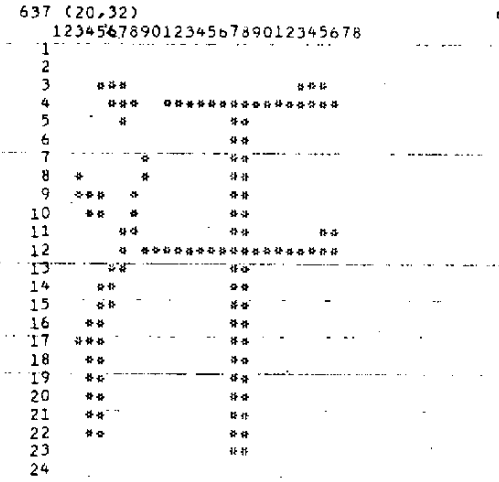
(2) 漢字出力システムにおけるドット・マトリックス・プリンタ出力リスト例(Ｂ)

なお、いずれのドット・マトリックス・パターンも、原フォントは㈱学習研究社の提供によるものである。

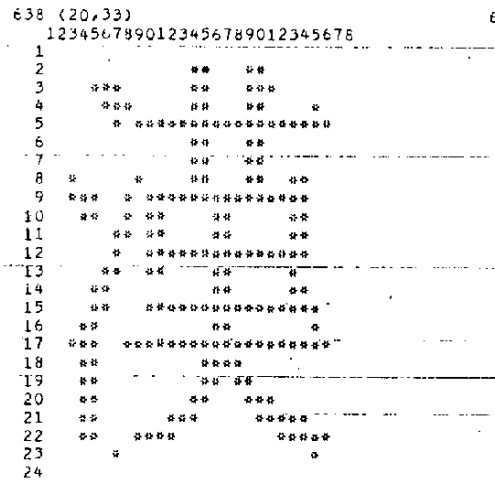

169 (3,89)
170 (3,90)
171 (4, 1)
172 (4, 2)
173 (4, 3)
174 (4, 4)
12345678901234567890123456789012345678


265 (5,12)
266 (5,13)
267 (5,14)
268 (5,15)
269 (5,16)
270 (5,17)

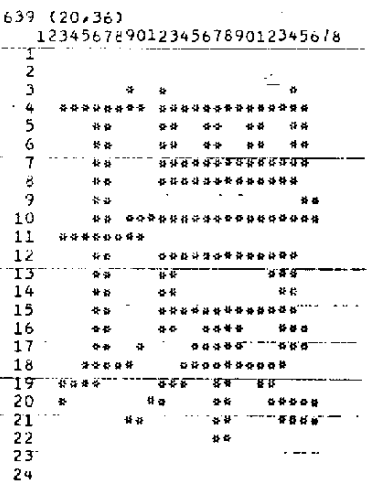
637 (20,32)



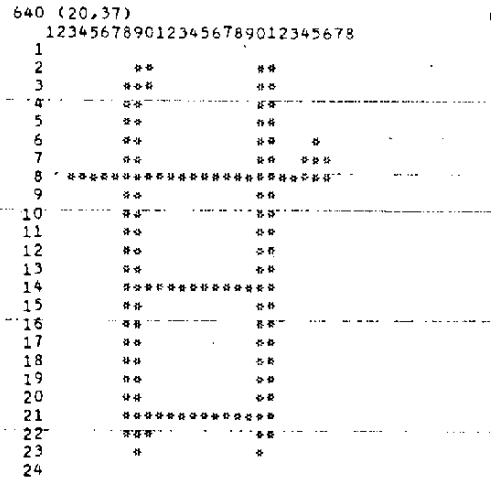
638 (20,33)



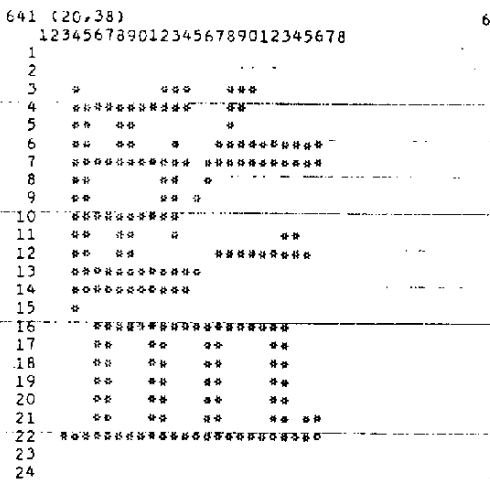
639 (20,36)



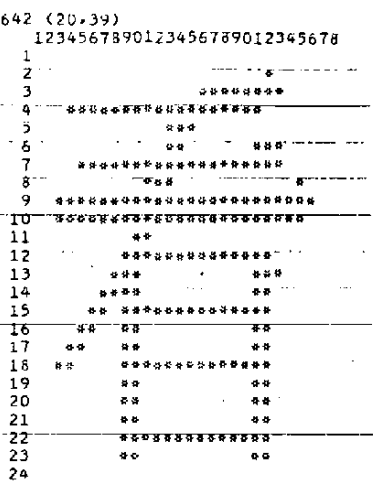
640 (20,37)



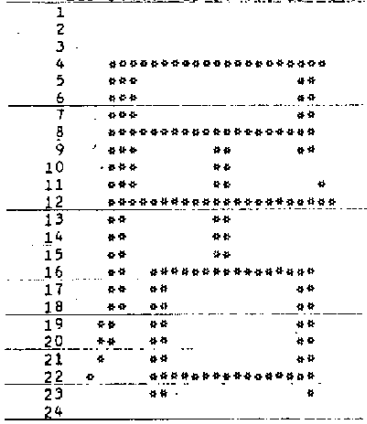
641 (20,38)



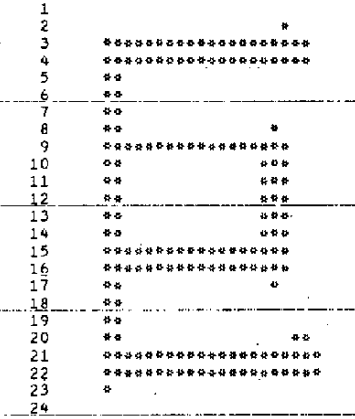
642 (20,39)



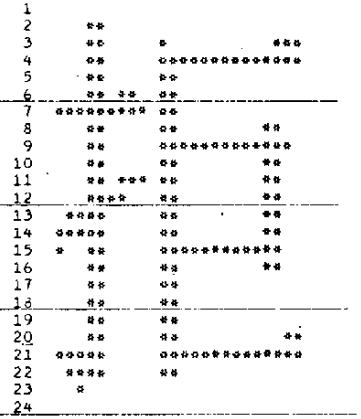
751 (21,79)



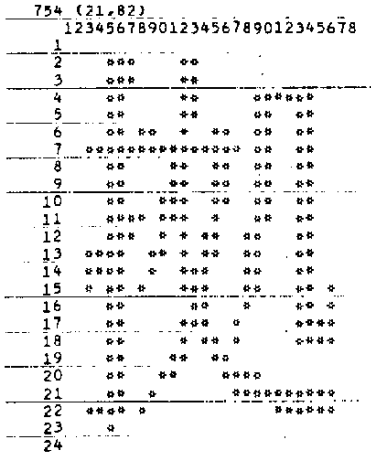
752 (21,80)



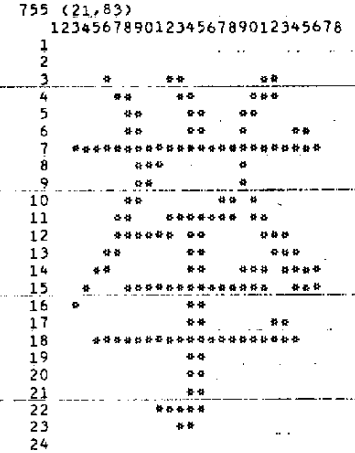
753 (21,81)



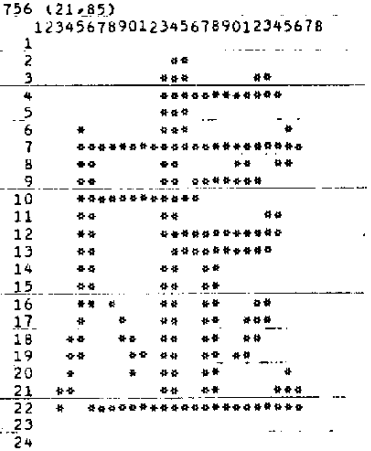
754 (21,82)

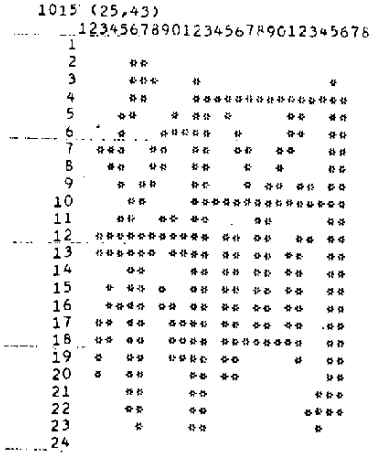


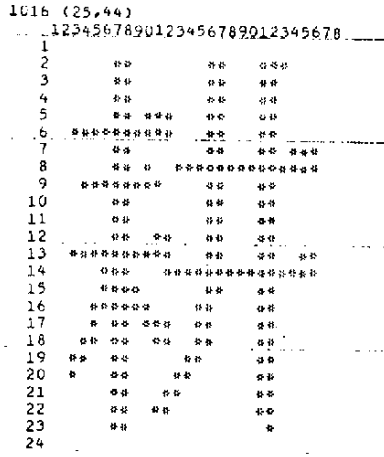
755 (21,83)

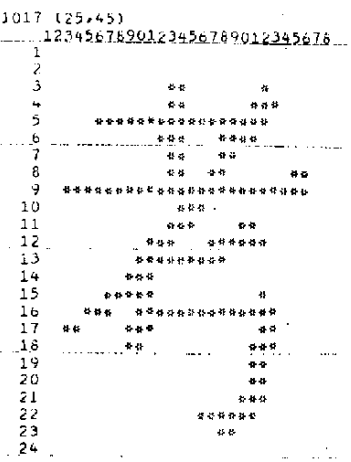


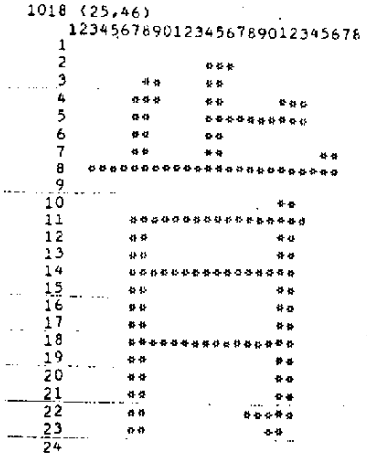
756 (21,85)

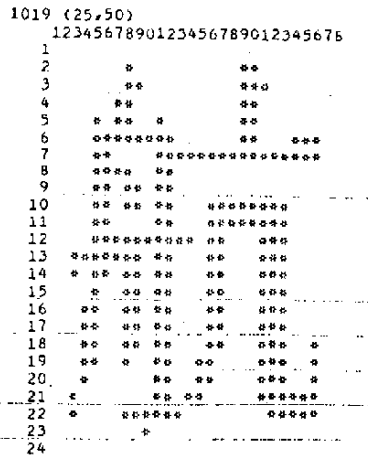


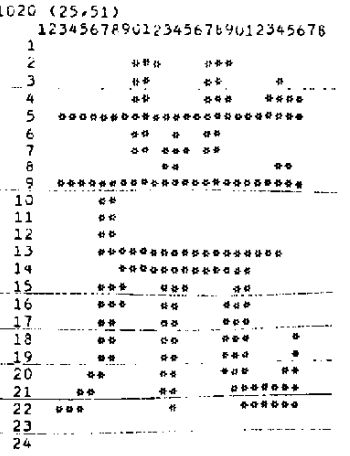

1015 (25,43)
1234567890123456789012345678


1016 (25,44)
1234567890123456789012345678


1017 (25,45)
1234567890123456789012345678


1018 (25,46)
1234567890123456789012345678


1019 (25,50)
1234567890123456789012345678


1020 (25,51)
1234567890123456789012345678

リスト(B)

漢字出力システム

(1) 出力装置
・VERSATEC MATRIX PRINTER 1100A
印字幅：260mm
針電極数：1024本
ドット密度：3.9ドット/mm
印刷速度：30.5mm/秒
印字用紙：静電記録紙

(2) 文字発生装置
・フロッピーディスク PF7034A-2
トラック数：77トラック
データ・トラック：73
インデックス・トラック：1
交換トラック：2
CE用リザーブ・トラック：1
記憶容量：242.944KB
128バイト/セクタ
26セクタ/トラック
128×26×73＝242944バイト
データ転送速度：31.2KB/秒
回転速度：360RPM
ヘッド移動時間：10mS/トラック
ヘッドセットリング時間：50mS

アクセスセットリング時間：10mS
平均アクセス時間：83mS
文字発生方式
ドットマトリックス方式
ドット構成：縦24×横24
576ビット（72バイト）/字
文字種：2582字
漢字：2242字
仮名文字：169字
(平仮名：83字)
(片仮名：86字)
英字：52字
数字：10字
記号：109種

(3) 制御装置
・PANA-FACOM U-400
制御方式：マイクロプログラム
語長：16ビット+2パリティビット
[illegible]：0.8マイクロセカンド (R-R加算)
メモリ容量：128KB

付記4　マイクロコンピュータ機器構成

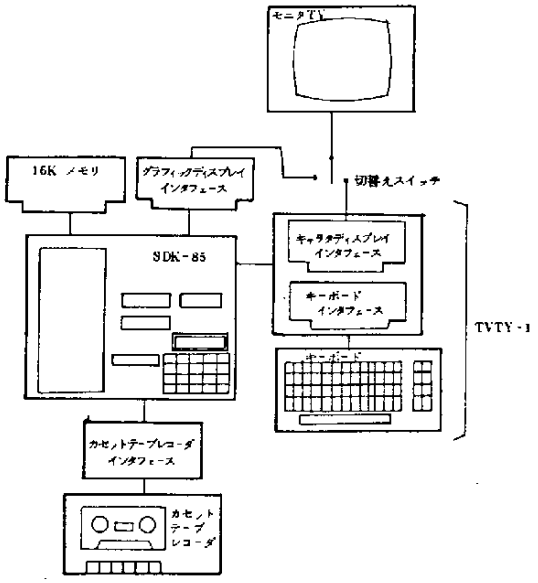


図1　実験システムのシステムダイアグラム

今年度、購入したマイクロ・コンピュータ・システムのシステム・ダイアグラムは図1に示すとおりである。このシステムは、SDK-85を中心として、これに16Kバイトのメモリ、カセットテープレコーダ・インタフェース、グラフィックディスプレイ・インタフェース、キャラクタディスプレイ・インタフェースを拡張して、モニタTV、カセットテープレコーダを接続したものである。またこのシステムはプログラムの開発と開発したプログラムの実行両方ができるようになっている。

まずプログラム開発では、切替スイッチでモニタテレビをキャラクタディスプレイ・インタフェースに接続する。モニタとしてはSDK－85の2Kバイト・モニタを使用する。このモニタに対して、キーボードよりコマンド等を入力し、その結果をモニタテレビに表示しながらプログラムを作成する。プログラム開発中は適当な時点でカセットテープにプログラムを保存しておくことにより毎回プログラムを入力する手間をはぶくことができる。プログラムが完成した後では、切替スィッチをグラフィック・ディスプレイ・インタフェースに接続することにより、モニタテレビをグラフィック・ディスプレイとして扱うことが可能となる。このグラフィック機能を使用して、モニタTVに漢字等を表示することができる。

以下に、このシステムの各コンポーネントについて述べる。

(1)　SDK－85

SDK－85はCPU8085とそのファミリ素子でシステムを構成したマイクロコンピュータキットである。8085は標準的なCPU8080の経験と実績を基に機能強化と使いやすさ、システムのコンパクト化をはかった8ビット・マイクロプロセッサである。8085の主な特徴は以下のようである。

1）単一5V電源で動作する。

2）より高集積化されクロック発振回路（８２２４）、システム・コントローラ（８２２８）の機能を内蔵している。

3）処理スピードが高速化し、パフォーマンスが向上した（８０８０に比べ約５０％向上）

4）割込み機能が強化されマルチレベルの割込みが直接処理できる。

5）シリアルデータの入出力ラインが追加された。

6）８０８０とソフトウェア完全コンパチブル

このキットのブロック図と仕様、メモリマップは図２、図３、表１のとおりである。

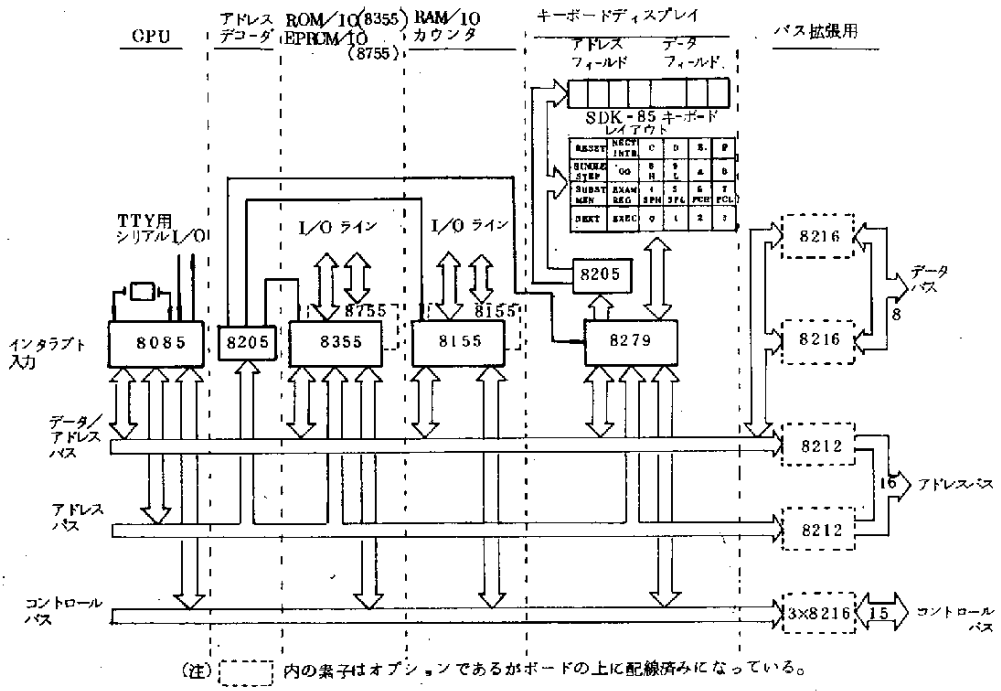


図５　ＳＤＫ－８５のブロック図

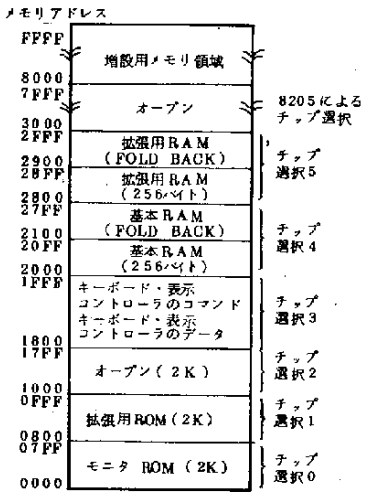


図３　ＳＤＫ－８５のメモリマップ

表1　SDK-85の仕様

C　P　U	8085(A) 命令サイクル時間……1.3μs Tcy …… 330ns
メ　モ　リ	ROM…2Kバイト(4Kバイトまで配線ずみ) 8355(モニタ 内蔵)×1 RAM…256バイト(512バイトまで配線ずみ) 8155×1
I/O	パラレルI/O…38本(76本まで配線などで拡張可能) シリアルI/O…8085のシリアル入出力端子 ボーレートはソフトウェアで生成 ボーレート;110ボー
インタフェース	バ　ス…すべてのTTLコンパチブル パラレルI/O…すべてのTTLコンパチブル シリアルI/O…20mA電流ループ
割り込み	RST 5.5 …キーボード割り込み RST 6.5 …TTL入力割り込み INTR ……TTL入力割り込み
D　M　A	Hold要求はジャンパ選択
ソフトウェア	システムモニタ・8355に書込みずみ ドライブされるI/Oはキーボード,表示およびTTY.　アドレス…0000～07FF
電　源	Vcc+5V±5%　1.3A (TTYを接続する場合はV_{TTY}= -10V.0.3Aを追加)
動作温度 寸　法 重　さ	0～55℃ 30.5cm×25.4cm×1.3cm (12インチ)×(10インチ)×(0.5インチ) 約340g

なお、このシステムでは増設メモリ領域を図４のように使っている。

また、このシステムにさらにメモリを増設するとした場合、増設用メモリ領域内にはアドレスこC０００～Ｆ７ＦＦの１４Ｋバイトとアドレス３０００～７ＦＦＦの２０Ｋバイトを使用することができる。

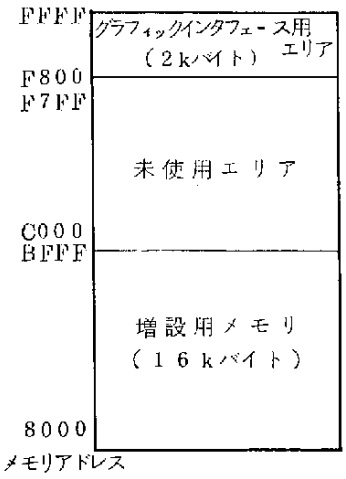


図４　増設メモリ領域

(2)　ＴＶＴＹ－１（キーボードとキャラクタ・ディスプレイ・インタフェース）

・　キーボード

キーボードの配列は図５のようであり、そのコード体系はＪＩＳコードに準拠している。このキーボードではパワーオン時には英数モードになり、カナモードにする場合はＣＴＲＬ－Ｎを入力し、英字モードにする場合はＣＴＲＬ－Ｏを入力する必要がある。このキーボードにはキャラクタ・ディスプレイ・インタフェース用の制御キーがあり、たとえば、ＣＬＥＲキーを押すことで、キャラクタ・ディスプレイの全面クリアができる。

													HOME	CLER
カナ									TAB			LF	CR	
CTRL			EOT		BELL	BS								
SHFT												SHFT		
スペース														

図５　キーボード配列

・　キャラクタ・ディスプレイ・インタフェース

このインタフェースでは画面上に表示する文字のために５１２バイトのＲＡＭを持っていて、コードリフレッシュ方式で文字を表示している。しかし、キーボード等との接続により、このメモリはＣＰＵからは見えなくなっている。このインタフェースにより発生される文字パターン（７×９ドット）は図６に示したとおりである。このインタフェースの

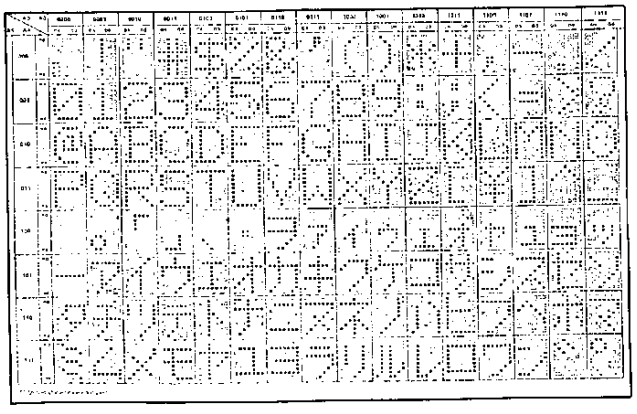


図６　７×９ドット・パターン

表示文字数は３２桁×１６行の５１２文字である。また、このインタフェースはＶＨＦモジュレータを持っており家庭用テレビのＴＶアンテナに接続することも可能である。ただし、このシステムではグラフィック・ディスプレイ・インタフェースとの関係でモニタＴＶと接続するため多少改造してある。

(3) グラフィック・ディスプレイ・インタフェース

このインタフェースはテレビ画面に、１２８（横）×１０８（縦）ドット構成の任意の図形を表示することができる。ＣＰＵ側からみると、このインタフェースは２Ｋバイトの ＲＡＭにみえる。また、このインタフェースはモニタＴＶが前提となっている。ボードアドレスは２Ｋバイト単位に決めることができ、ＤＩＰ－ＳＷにより行なう。このインタフェースの持っているメモリとディスプレイ画面上のドットとの関係は、ボードアドレスをＣ０００番地（メモリはＣ０００～Ｃ７ＦＦ番地）を指定した場合、図７のようになる。ただしメモリは２Ｋバイトあるが、表示画面には１０８行しか表示できないため残ったメモリ３２バイトは表示画面のドットと関係のないメモリになる。なお、メモリ内容の各ビットは"１"ならば白、"０"ならば黒と画面上ではなる。

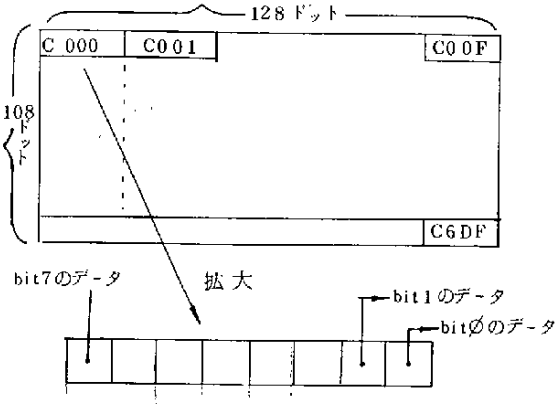


図７　アドレス対応表

(4) カセット・テープレコーダ・インタフェース

現在一般化しているカセットのインタフェース方式には、カンサスシティ・スタンダードとトーンバースト方式がある。この両者の違いは図８に示すとおりである。このシステムではトーン　ースト方式のインタフェースを用いている。ＳＤＫ－８５のＣＰＵ８０８５には内蔵したプログラマブルなシリアル入出力ポートがあるので、このシステムではこれを利用している。

	トーンバースト方式	カンサスシティースタンダード
転送形式	直列非同期	
転送速度	１１０ビット/秒	３００ビット/秒
論理"１"	約2000Hz～4000Hz	2400Hz
論理"０"	無　信　号	1200Hz

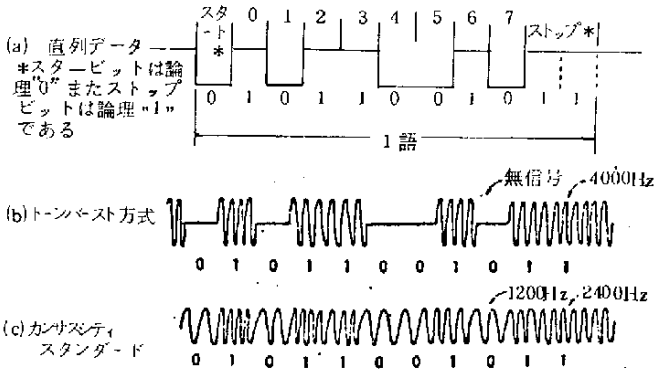


図８　カセットインタフェース方式

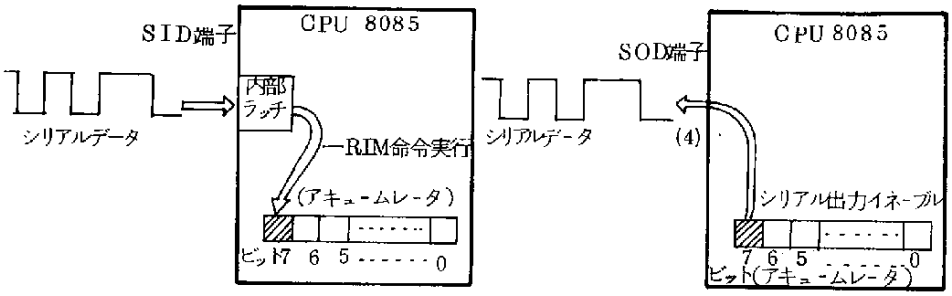


図9　8085のシリアル入出力機能

(5)　メモリボード（１６Kバイト）

このメモリボードは２１１４（１K×４ビットスタティックRAM）３２個より成っていて、ボードアドレスは１６Kバイト単位に指定できる。

参 考 文 献

(1) 日本語情報処理技術の研究開発

JIPDEC、52−S−001分散型リソース処理技術の研究開発、'78.3、PP. 471−577

(2) 日本語情報処理の技術動向調査報告書

JIPDEC、47−R−001.'73.3、P. 282

(3) 文書情報処理に関する調査研究

JIPDEC、48−S−004.'74.3、P. 206

(4) 文書情報処理に関する研究開発

JIPDEC、49−S−006.'75.3、P. 151

(5) 日本語情報処理システムの研究開発

JIPDEC、50−S−001.'76.3、P. 300

(6) 日本語情報処理システムの研究開発

JIPDEC、51−S−001.'77.3、P. 321

(7) 日本語情報処理シンポジウム報告集

情報処理学会プログラミング・シンポジウム委員会、'78.7、P. 370

(8) 漢字入力法の人間工学的検討

情報処理学会、昭和53年度第19回全国大会パネル討論会資料、'78.8、P. 12

(9) 情報交換用漢字符号系 JIS C 6226−1978

日本工業標準調査会、日本規格協会発行

(10) 情報交換用符号 JIS C 6220

日本工業標準調査会、日本規格協会発行

(11) 日本語ワードプロセッサ JW−10

JICST、情報管理、'79.3、Vol. 21 No. 12、PP. 972−974

(12) 専門分野別による漢字および漢字単語の調査分析

JIPDEC、'74.12、P. 554 (2分冊)

(13) 日本語端末の構成と機能について

野村、小池（JIPDEC）、情報処理学会第19回全国大会、'78.8、2G−3、PP. 661 − 662

(14) PROMによる漢字発生器

内藤祥雄、南谷崇（日本電気）、電子科学、1976、5月号、PP. 37−43

(15) 画素間の相互情報を利用した文字パターンの符号化法

称津孔二（日本電気）、電子通信学会論文誌 '72/4 Vol. 55−D、No. 4、PP. 277−284

(16) 計算機内における画像のデータ圧縮

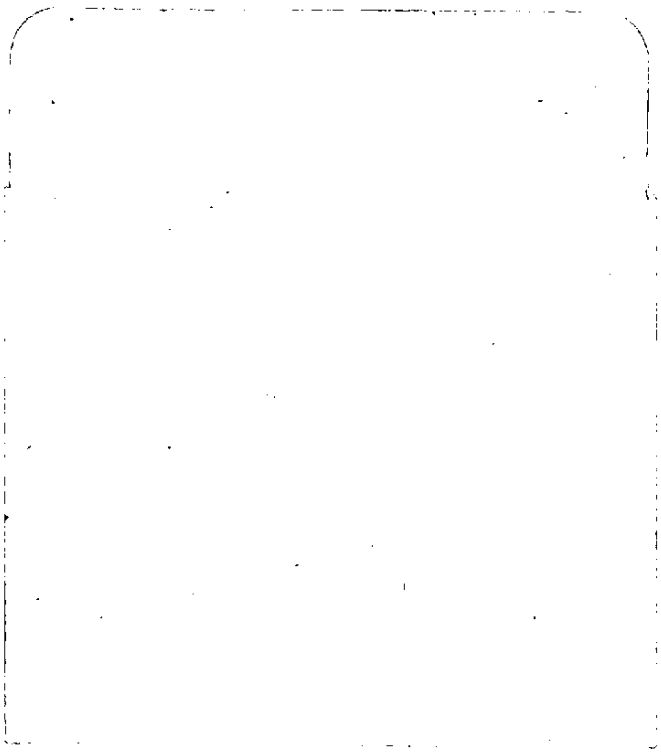
尾上守夫、岩下正雄（東京大学）、情報処理、Aug. 1977、PP. 776−780

(17) Intel SDK-85 USER'S MANUAL
Intel Corporation 1977, Manual Order Number 9800451A
(18) MCS-85™ USER'S MANUAL
Intel Corporation 1978, 9800366D
(19) 8080/8085 ASSEMBLY LANGUAGE PROGRAMING MANUAL
Intel Corporation 1978, 9800301C

—— 禁 無 断 転 載 ——

昭和５４年３月発行

発行所　財団法人　日本情報処理開発協会
東京都港区芝公園３－５－８
機械振興会館内
ＴＥＬ　(434)8211(代表)

印刷所　株式会社　タケミ印刷
東京都千代田区神田司町２－16
ＴＥＬ　(254)5840(代表)

53－S001