

デジタル交換機用  
マルチプロセッサ  
制御方式に関する研究

昭和63年11月

山田 喬 彦

デジタル交換機用  
マルチプロセッサ  
制御方式に関する研究

昭和63年11月

山田 喬彦

DOC
1988
23
電気系

# 目次

第I章 序論	1
1. 序言	1
2. 研究の背景	2
2.1 電子交換機制御方式の開発と本研究の位置	2
2.2 デジタル交換機制御方式の研究	6
2.2.1 分散制御方式の研究	6
2.2.2 実用デジタル交換機制御方式の動向	7
3. 研究の概要	9
3.1 デジタル交換機におけるマルチプロセッサの位置づけ	9
3.2 マルチプロセッサ構成法	10
3.3 マルチプロセッサの基本設計	11
3.4 マルチプロセッサの高信頼設計	13
3.5 マルチプロセッサの故障回復手法	14
3.6 交換機制御方式の今後の動向	14
第II章 デジタル交換機の特徴と制御方式への課題	15
1. 序言	15
2. デジタルスイッチの特徴とデジタル交換機の方式目標	16
2.1 デジタルスイッチの特徴	16
2.2 スwitchのシステム化とその特徴	19
2.3 デジタル交換機の方式目標	21
2.4 デジタル交換機における制御方式の目標	25
2.5 デジタル交換機制御方式におけるマルチプロセッサの採用	28
2.5.1 マルチプロセッサの有効性	28
2.5.2 マルチプロセッサの実現条件	31
3. 結言	32
第III章 デジタル交換機制御用マルチプロセッサの基本構成	34
1. 序言	34
2. 交換機制御用マルチプロセッサの基本系	35
3. リソース管理方式とマルチプロセッサ構成	35
3.1 リソース管理のモデル化と対案	35
3.2 ダイナミックステップ数によるリソース管理方式の評価	40
3.2.1 リソース管理方式とダイナミックステップ数	40
3.2.2 両方式のダイナミックステップ数の比較	41

3.3	考察	43
4.	機能分散による呼制御処理の最適化	43
4.1	交換処理の機能分割	43
4.2	機能分散方式の比較	44
4.2.1	評価パラメータ	44
4.2.2	分散方式と平均処理時間	46
4.3	考察	48
5.	機能分散による信号処理の最適化	48
5.1	交換処理の処理量の定式化	49
5.1.1	交換処理機能の分類	49
5.1.2	処理量の定式化	50
5.2	信号方式対応の機能分散の効果	52
5.3	考察	55
6.	交換制御用マルチプロセッサ基本構成の確定	55
7.	結言	57
第IV章 デジタル交換機用マルチプロセッサの基本設計		58
1.	序論	58
2.	シングル/マルチプロセッサの ONE-MACHINE 概念による統一	59
2.1	ONE-MACHINE 概念	59
2.1.1	ソフトウェア系列化の要求	59
2.1.2	シングルプロセッサの必要性	59
2.1.3	ONE-MACHINE 概念の構築	60
2.2	ONE-MACHINE 概念の具体化	60
2.2.1	ONE-MACHINE 概念と制御システムの階層構成	60
2.2.2	ONE-MACHINE 概念によるシステム具体化の狙い	62
3.	ONE-MACHINE 概念に基づくマルチプロセッサ構造の階層化	62
3.1	交換制御系の階層化	62
3.2	階層インタフェースの実現法	63
4.	マルチプロセッサの設計事例 I	
	既存交換用プロセッサを用いたマルチプロセッサの設計	71
4.1	通話路駆動階層の設計	71
4.2	呼処理プログラム階層の設計	74
4.3	オペレーティングシステム階層の設計	76
4.4	マルチプロセッサ用実マシン階層の設計	77
4.4.1	マルチプロセッサの命令実行制御系	77
4.4.2	入出力アクセス付加機構とプロセッサ間 通信制御機構の設計	81

4.5	共通メモリの接続法	89
4.6	マルチプロセッサの実現結果	90
4.7	性能評価	92
4.7.1	シングルプロセッサの平均命令実行時間	92
4.7.2	マルチプロセッサのプログラムの結合	94
4.7.3	マルチプロセッサへの対応化	95
4.7.4	マルチプロセッサの平均命令実行時間	96
4.7.5	システムとしての処理能力	100
4.7.6	処理能力の推定	103
5.	マルチプロセッサの設計事例II	
	32ビットカスタムVLSIプロセッサの実現と マルチプロセッサへの適用	105
5.1	交換処理用カスタムVLSIプロセッサの開発	105
5.1.1	汎用VLSIプロセッサと カスタムVLSIプロセッサの比較	105
5.1.2	カスタムVLSIプロセッサの有効性	106
5.1.3	関連分野のプロセッサの応用と カスタムVLSIプロセッサの経済性	110
5.1.4	32ビットカスタムVLSIプロセッサ開発目的	113
5.2	VLSIプロセッサ実現に向けた課題と設計方針	114
5.3	VLSIプロセッサの仕様の概要	119
5.4	VLSIプロセッサの設計法	119
5.4.1	論理設計手法	121
5.4.2	デバッグ手法	121
5.5	VLSIチップの実現結果	128
5.6	マルチプロセッサへの応用と評価	128
6.	結言	141
第V章 デジタル交換機制御用マルチプロセッサの高信頼設計 142		
1.	序言	142
2.	デジタル交換機の規模別信頼度規定	143
2.1	従来の信頼性規定の考え方	143
2.2	規模別信頼度規定	143
3.	交換機制御用マルチプロセッサの高信頼化	145
3.1	フェイルソフトによる規模別信頼度規定への対応	145
3.2	交換処理プログラム配置とフェイルソフト概念	145
3.2.1	プログラム再配置法	145
3.2.2	プログラム固定配置法	147

3.3	フェイルソフト概念と負荷分散概念の対応化	147
4.	予備装置配置	148
4.1	負荷分散概念に基づく予備装置配置	148
4.2	予備の待機状態	149
5.	フェイルソフト設計	150
5.1	フェイルソフトの実現条件	150
5.2	機能充足性の考察	150
5.2.1	縮退系と呼処理	150
5.2.2	不稼働部と呼処理	152
5.3	故障隔離	154
5.3.1	動的影響の排除	154
5.3.2	故障履歴の排除	154
6.	評価	158
6.1	評価の前提	158
6.2	不稼働率の算出	160
6.2.1	システム構成要素の不稼働率	160
6.2.2	収容端子から見た不稼働率	162
7.	結言	165
第VI章 デジタル交換機制御用マルチプロセッサの故障回復法		167
1.	序言	167
2.	マルチプロセッサ高信頼化	168
2.1	高信頼構成の前提	168
2.2	故障処理の遷移	169
3.	階層化思想に基づく故障処理	169
3.1	故障処理の階層化	169
3.1.1	故障処理の局所化	170
3.1.2	系内処理から系外処理へ	170
3.1.3	ソフトウェア処理からハードウェア制御へ	170
3.2	故障処理の階層構成	171
3.2.1	局所故障処理階層	171
3.2.2	マスタープロセッサによる故障処理階層	173
3.2.3	ハードウェア支援による故障処理階層	174
3.2.4	外部監視支援階層	175
4.	再開処理	176
4.1	罹障点復帰再開	176
4.2	個別再開処理	176
4.3	全系初期設定再開	177

5.	評価	177
5.1	評価パラメータ	178
5.2	評価の前提	179
5.2.1	故障波及度	180
5.2.2	中断時間の仮定	180
5.2.3	故障原因と生起頻度	183
5.2.4	故障回復フェーズの遷移	184
5.3	故障処理中断遭遇比率の具体化	184
5.4	シングル/マルチプロセッサ方式の比較	186
5.5	計算結果	186
5.6	考察	186
6.	結言	189
第VII章	分散伝達方式への展開	191
1.	序言	191
2.	高集積化による交換機小型化の可能性	192
2.1	予備構成単純化による小型化	192
2.2	超高集積部品の進展による交換機の小型化	197
3.	超高集積部品の通信ノード構成へのインパクト	200
3.1	小規模交換機の有効化	200
3.2	大規模交換機の高性能化	201
3.3	高信頼効果	202
4.	高速通信サービスの普及に向けたノード構成	204
4.1	小規模加入者ノードによる高速通信 サービスの低コスト化と課題	204
4.2	加入者網の対案	205
4.3	ループ網方式の比較	207
4.4	ノード構成例	208
4.5	呼制御の論理化と階層化	210
5.	システムオンチップ化の課題と実現手法	210
5.1	オンチップ対象とするデジタル伝達ノードアーキテクチャ	210
5.2	デジタル伝達ノードオンチップ化の問題	211
5.3	アーキテクチャシミュレータ "ASCOT"の概要	213
5.4	アーキテクチャシミュレータ "ASCOT"の役割	215
6.	今後の分散システムの動向と研究課題	217
6.1	分散システムの動向	217
6.2	今後の課題	219
7.	結言	221

第VIII章 結論 .....	222
謝辭 .....	225
参考文献 .....	227

# 第I章 序論

## 1. 序言

人間の意志疎通を支援し、最も高級な道具と考えられる電気通信網は多くの改良を経て社会生活における利便性の向上に貢献してきた。中でも蓄積プログラム制御方式電子交換機の導入は通信網の機能高度化の点からは最も大きなインパクトを与えたと考えられる。蓄積プログラム制御方式は通信網の機能仕様を具現する交換機の制御機能をプログラム論理で実現し、プログラムの高論理性により高度な機能の実現を、またプログラムの融通性によって通信網仕様の進化を可能とする。ただ、初期の蓄積プログラム制御方式には単一プロセッサ制御による性能限界というアキレス腱を秘めていたのも事実である。

シングルプロセッサの性能限界を取り除き、通信網の進化を約束する制御の分散化の研究は実用化を指向したデジタル交換機の研究の一環として諸外国を含めて広く行なわれた。高集積部品技術が飛躍的な進展を遂げ、電気通信網においては伝送路のデジタル化だけでなく、交換機のデジタル化を進め、音声や画像やデータを一元的に伝達可能とするデジタル統合網への具体化を開始した。制御の面ではマイクロプロセッサによる処理コストの大幅な低下と処理技術の一大普及を促し、制御の分散化の下地が出来てきた。

分散制御方式の研究対象であるデジタル交換機はアナログ通信からデジタル一元化通信への通信網の世代交代の位置にある。分散制御方式には両者からの幅広い要求に適合化することが求められる。一方、複数プロセッサの組合せからなる分散制御方式は自由度が大きいですが、逆にその分システムの隘路が不明確化し、望ましい方式をいかに確立するかは大きな課題である。さらに交換機制御用には実時間制御や高信頼化などの新たな問題を解決しなければならない。

筆者等は、①将来のサービス拡大への柔軟性、②規模に応じた制御能力の実現、③ハードウェア支援によるマルチプロセッサ複雑性の排除、④規模別信頼度規定に整合する高信頼化等を目標として分散制御方式を検討し、機能・負荷分散併用形のマルチプロセッサを案出し、試作・実験による確認を行なって、実用化へ導いた。本論文は、筆者がハードウェア担当者として進めた交換制御用マルチプロセッサの概念構築、その具体化に関する研究成果をまとめたものである。

## 2. 研究の背景

### 2.1 電子交換機制御方式の開発と本研究の位置

1963年、米国ベル研究所が蓄積プログラム制御方式電子交換機の実用化計画を発表した<sup>(1)</sup>。これは通話路部に電磁・機械スイッチを用いるが、制御部にコンピュータを用いてプログラム論理で交換機を制御する新しい技術であった。この交換機の特徴である蓄積プログラム制御方式は、プログラム制御の融通性を利用して①新電話サービス提供に対する機能の付加を容易化し、②番号計画・課金方式の変更に対する融通性を付与し、③構成装置の単一性（交換機種の違いはプログラムで対処）によってストック生産を可能とするものであった。

この発表は内外に反響を呼び、各国の電子交換機の実用化の促進に多大の影響を与えた。以後、蓄積プログラム制御方式は交換機制御方式として不動の位置を獲得している。

ここで本研究のマルチプロセッサ制御方式に至る電子交換機制御方式の歴史（概略経過を図1.1に示す）と本研究の位置を概述する。筆者は制御方式の研究の歴史を以下の3期に分ける。

#### 第1期 蓄積プログラム制御方式の確立

「蓄積プログラム制御方式はハードウェアをコンピュータで制御するもの」と定義すると、現在ではごく当り前のシステム技術に聞こえる。しかし、蓄積プログラム制御方式が発明された時点では革新的発想であった。コンピュータ技術が未熟であった時期に①刻々変化するハードウェアの状態をメモリ上に写像して処理する概念、②処理対象を代えながら定型処理プログラムを繰り返し実行する概念、③多数の呼を並列に多重処理する概念、④ハードウェアの厳密な時間制約に応答し得る実時間制御の概念が創りだされている。また概念のみでなく、部品技術が未熟な時代に24時間無休止システムを実現し、100kステップを越える大規模プログラムを総合して運用可能とする高精度なプログラム技術を確立している。この技術を盛り込んだNO. 1 ESSを本格的に導入することを米国ベル研究所は1963年に発表している<sup>(1)</sup>。

日本でもそれまで電子部品化による機械的摩耗の無い電子交換機の実現を目指

# 交換機開発の経緯と今後の進展

	1950年代	1960年代	1970年代	1980年代	
交換方式	Step by step ・自動化 ・省力化	クロスバー交換機 ・全国自即化への対応	電子交換機 ・通信網への融通性の付与 (広域時分制、国際自即等) ・新サービスの導入 (キャッチホン、短縮ダイヤル等)	電子交換機 (高水準言語CHILLの導入) 小規模局への展開 (D20, D30...) ・SPC網への展開 (共通線網)	デジタル交換機 ・各種メディアの統一交換 ・電子部品化による経済化 ・加入者線の高機能化、デジタル化 広帯域ISDN交換機 ・広・狭帯域統合化 ・放送・交換機融合化 ・小規模交換機の網的分散化
部品	機構部品	機構部品	機構部品+電子部品IC	機構部品+電子部品(LSI)	電子部品(LSI, VLSI) (VLSI プロセッサ) 超大規模LSI
導入状況	(新規導入) STEP BY STEP	XB交換機	電子交換機 No.1 ESS(米) E10 (仏) No.4 ESS(独)	電子交換機 共通線網(日) No.5 ESS(米) INSモジュール(日) INS 非電話交換機(DDX)	
主たるシステムの形態	XB 電話サービスの全国拡大 ハードウェア 通話路 XBスイッチ ・布線論理 ・共通制御 ・間接制御 (レジスタ) ・トランスレータ	電子交換 既存サービスのSPCでの実現、 新サービスの導入 ハードウェア 通話路 小形XBスイッチ ID10プロセッサ ソフトウェア ・実時間処理 ・超多重処理(数万のI/O) ・運転中フェイル入替、増設 ・多種既存信号方式の制御 ・ソフトウェア構造化 (マザーファイル) SPC化 保守支援 ・故障処理、診断等の プログラム制御化	SPCの普及 通話路 SMMスイッチ プロセッサの局規模別系列化 (D10HCP, D20, D30) ・電話交換、国際接続、 移動体交換 ・D100B(モジュール化) ・きめ細かい再開処理 ・各種ハードウェアに最 適化したソフト制御 ・電子交換 大量導入 への対応 ・システム立ち上げ (IPL)の自動化	多様なサービスへの拡大 通話路 時分割通話路 (メモリ、ゲート) プロセッサの統一 (VLSI シングル、マルチ) ・電話交換、データ交換 ・パケット交換、通信処理 ・各種新サービス毎に最 適化したソフト開発効 率化 ・各種サービス用I/Oの 制御 ・サービスの展開 ・LSI技術の 積極的採用 ・保守インタフェースの 集中化(ダイヤル) ・診断辞書検索の機械化	今後の展開 (一層の高信頼化、経済化 サービス総合化、広帯域 サービスの普及) 多様なメディア 多量の情報 への対応 高集積パワの 活用 分散システム 通話路 ATMスイッチ STM分散スイッチ システムオンチップ化 (制御・伝達系の一体 チップ化、ASICプロセッサ) ・基本のサービス、付加サ ービスのモジュール化 ・電話交換、データ交換、 広帯域交換、多元接 続、パケット交換、衛星 交換通信処理 ・統合保守 機械化による 人的ミス排除 ・保守エキスパート システム
NIHのシステム	XB { C4-5 ..... C400 C80 ..... C82 C60 ..... C63	電子交換機 { DEX1 ..... DEX2 ..... D10 ..... DEX-T1	D20 ..... 改D20 D10-R1 経済化D10 ..... D10HCP	複合CES D30 自動内航船舶・自動車電話 データ交換機 { D50(CS) ..... 改D50 D50(PS) ..... D51 デジタル交換機 (D60 .. D70)	

して通話路と制御部を併せて電子化する全電子式電子交換機の研究を進めていたが、ベル研による革新的発想に刺激を受け、研究の重点を蓄積プログラム制御方式電子交換機に移した。いわば習作としてのDEX-1<sup>(2)</sup>、進展著しい計算機技術を電子交換機向きに最適化し、国産技術として確立したDEX-2<sup>(3)</sup>、これに容量増や各種基準への適合化を図ったDEX-21<sup>(4)</sup>を経て、わが国の商用電子交換機技術が確立した。

しかし蓄積プログラム制御方式電子交換機は革新的な発想を取り込み、実用可能な技術を確立しながら普及という観点からは必ずしも順調ではなかった。経済性については、IC、磁心メモリなどの電子部品がまだまだ高価であり、電磁式クロスバ交換機に比較して、基本電話サービスの範囲では割高という問題が残った。また、新サービスへの期待を担いながらも、最大規模の局の基本サービスを実現するだけで制御装置の能力が限界に達し、その後の新サービスを実現していくには処理能力不足という矛盾もはらんでいた。

## 第2期 電子交換機普及促進に向けた経済化、高性能化

電子交換機の本来の目的を達成するには電子交換機の普及促進が重要な課題であった。電子交換機のみ可能なサービスは、電子交換機が大量に導入され、それらの制御システム間がデータ伝送で結ばれ、網全体があたかも単一計算機のように動作することによりなされるとされた<sup>(5)</sup>。電子交換機制御系には、基本サービスでも既存交換機に経済性で勝って導入が進むこと、しかも将来に渡って電子交換機の融通性を発揮し得る高性能化が必要とされた。

このような電子交換機の普及の促進に向けた研究には二つの大きな流れがあった。一方は既に実用化を果たした大規模局用電子交換機の経済化と高性能化である。制御系に関しては、米国ベル研究所における1Aプロセッサ<sup>(6)</sup>やNTTにおけるD10-高速中央処理系装置<sup>(7)</sup>〔13〕〔14〕がこれに当たる。もう一方の研究は、中小規模局領域へも電子交換機を適用可能とすることであり、米国でのNo. 3ESS<sup>(8)</sup>や日本での中規模局用DEX-A1実験機<sup>(9)</sup>とD20電子交換機<sup>(10)</sup>、これを改良した改良D20交換機<sup>(11)</sup>、小規模局用D30交換機<sup>(12)</sup>がこれにあたる。

まず、大規模局用交換機の改良では新テクノロジー（部品、実装技術）の採用

と新テクノロジーに合わせた回路構成法等の工夫や既存構成方式のスリム化（過度の冗長性の排除）が図られた。コンピュータ産業の刺激を受け、部品や実装技術は急速に進展しつつあり、これらの技術を交換機制御向きに適合化して有効に利用する手法の開発が重要であった。システムハードウェアの経済化によって部品技術の改良が最も大きな要因となるが、システム寿命が短く、開発の早さを競うコンピュータとは同一部品技術といえども適用法が異なる。電子交換機には長期の運用寿命を考慮した信頼性や部品の能力を最大限に活用した経済性が求められる。システムの高性能化を目標にECL高速素子が採用され、高速性が必要な制御中枢部には交換機向きの命令実行の効率化を考慮したカスタムLSIが開発された。ただし、高性能化と同時に、素子速度の余裕を生じる個所には素子速度の余裕を金物量の削減、経済化に活かす理念が生まれた。この研究の中でマイクロプログラム制御による効率の良い多重制御方式が考案され、電子交換機の高性能化と経済化に活かされた<sup>[15]～[23]</sup>。

一方、中小規模局用の電子交換機の開発は<sup>(9)</sup>、大規模局用よりさらに厳しい経済性要求の下で遂行された。共通制御方式による電子交換機にはシステムコストの固定分が存在し、端子当りコストは規模が小さくなるほど割高となる。固定コストの大部分を制御システムが占めることから、制御システムには性能を落しても機能の削減や演算ビット幅の削減など小規模化への対応が必要とされた。

中小規模局用電子交換機D20の改良実用化<sup>[24]～[26]</sup>においては、高集積部品の採用による経済化と同時に、高集積時代のシステム構成すなわち常用と予備の間の交絡を簡略化した中央処理系全体の完全二重化予備構成や磁気バブル制御への多重マイクロプログラム制御手法などが発案され、効率性が追求されている<sup>[27]～[28]</sup>。これらの改良を経て電子交換機が大幅に高性能化・経済化され、大量導入が始まる。これにより、蓄積プログラムの長所を活かした機能の拡大が年々行なわれるようになった。

電子交換機が大量に導入されると、プログラム制御の融通性とは裏腹にソフトウェアへの投資の増加、いわゆるソフトウェアネックの問題がクローズアップされるようになった。電子交換機ではプログラム制御の融通性から機能の追加は当然行なわれるべきものであるが、大中小規模と系列の異なる電子交換機が併存することにより、系列毎のプログラムを独立に維持管理しなければならないこと

が問題とされた。

### 第3期 総合通信網の構築へ向けた蓄積プログラム制御方式の高度化

第2期での残された問題は、次世代交換機としてのデジタル交換機の研究の中で解決されることになった。既に空間分割電子交換機の実用化が一段落し、デジタル部品技術の進展からデジタル交換機の経済性が見通しが立ちつつあった。デジタル交換機の実現に向けた研究が世界的に始まり、同時にデジタル交換機用制御システムの開発が盛んになった。

デジタル交換機は、通信網の情報をデジタル信号に一元化して伝達しうるため、同一端末から各種のメディアを統一して通信可能とし、きたるべき情報社会のインフラストラクチャ、いわゆるデジタル総合通信網の実現を可能とすると期待された。第3期の研究については本論文が主題とするものであり、次節に詳述する。

## 2.2 デジタル交換機制御方式の研究

### 2.2.1 分散制御方式の研究

デジタル交換機の制御システム用に複数プロセッサによる分散制御方式が期待され始めた。交換機的全適用領域を同一プロセッサとソフトウェアで一元化し、ソフトウェアの維持管理工数を削減しようとするもので、前記の残された課題の解となるものである。また、研究の背景として、この他、以下の動向が分析されている<sup>(29)</sup>。

#### (1)集中化への反省

従来の蓄積プログラム制御方式は集中化による効率化、大群化を過度に追求することからプログラムの複雑化を招いている。効率追求のあまり、プログラムの機能実体が密結合となり、機能追加のために修正すべきモジュールの特定が困難となっている。さらに、機能モジュールの組み合わせにより各種交換機の実現を図るモジュール化技術が未成熟であるとされた。

#### (2)マイクロプロセッサ等のLSI電子回路技術の進展の利用

マイクロプロセッサを中心としてLSI電子回路の進展が著しく、マイクロプロセッサの実現領域ではグロシュの法則が崩れていた。制御システムにおいて

は、大規模化追求の精神的支えが無くなりつつあった。このためマイクロプロセッサで交換機の制御を実現する研究が盛んになった。

### (3)通信網の危険分散

通信網の規模が大きくなり、その社会的責任が大きくなったことから信頼性向上が期待された。信頼性向上を図るには、分散を地域的に広げ、危険の分散を図ることが一つの解になりうる との見通しであった。

このような背景のもとに、それぞれ以下のような狙いをもって交換機の分散制御の基礎的研究が数多くなされている。(30)～(36) [37] [38] (39)～(44)。

- (a) L S I 電子回路技術の活用による経済化
- (b) 機能分散化によるソフトウェアのモジュール化とソフトウェアの簡明化
- (c) 交換機の適用規模に合わせた処理能力による総合経済性の達成
- (d) シングルプロセッサの能力限界を越える処理能力の実現
- (e) フェイルソフト技術による交換機の信頼性の向上
- (f) 網的分散による通信網の信頼性向上
- (g) 端末まで機能分散し、端末をインテリジェント化したサービス向上

#### 2.2.2 実用デジタル交換機制御方式の動向

分散制御の理想を追求した基礎的研究と並行して 実用化に向けた着実な研究も進行した。通信網は既存の端末との接続性や互換性が必要であり、分散制御に対する期待も実用化を意識した途端に現実の条件への適合化に絞られていく。実用交換機においては分散制御を指向しても、回線等の資源は交換機全体で共有することになるため、資源はなんらかの方法で共通に管理することが必要となる。実用の観点では集中と分散の調和が必要であり、分散制御の理想追求が強すぎると実用、商用の点では必ずしも成功しない。例えば、商用化されたシステムの中では I T T 社のシステム 1 2 4 0<sup>(45)</sup> が最も分散制御の理想に近いが、商用市場では失敗と見られている。小規模な通話路の制御や信号処理をマイクロプロセッサで実現し、全規模の交換機を小規模モジュールによるビルディングブロックで拡大して、実現しようとするものであった。しかし、結局大規模局の能力の達成が困難であったとされ、現実ではきわめて少ない導入量と予想される。

	Ericsson AXE-10	NTT- D70	ATT- No. 5 ESS	ITT- 1240
信号処理 (中継系)				
信号処理 (加入者系)				
共通リソ ース管理				
ルーティ ング				
運転管理				

注) 各箱がプロセッサを示す。  
CP-A, Bは利用プロセッサが  
異種であることを示す。

: 同一機能の負荷分散

図1.2 世界の代表的デジタル交換機の制御系構成

收容する回線をいかに効率良く運用するかが交換機の1つの使命である。通信網資源を管理する共通制御部をどのように構成して、また通話路等の周辺制御機能をどのように付加するかは現実の解は落ちつく。現在、実用に供されている世界の代表的デジタル交換機の分散制御の概要を図1.2に示す。

スエーデンのエリクソン社 AXE-10<sup>(46)</sup>は複数のスイッチモジュールに対応した周辺制御プロセッサとこれを統合的に制御するシングルプロセッサで制御システムを構成している。米国ATT社 No. 5 ESS<sup>(47)</sup>では、呼処理の機能をさらに周辺プロセッサに移譲しているが、資源の管理は単一プロセッサで管理している。日本のデジタル交換機の代表機種として大量に導入されているD70<sup>(48)</sup>デジタル交換機のマルチプロセッサは、単一プロセッサに相当する機能をVLSIプロセッサによるマルチプロセッサで負荷分散することにより

小規模からの経済的な拡大を狙っている。

設計者の思想や重点の置き方により、必ずしも呼処理の機能分割の位置、分割間のソフトウェア／ハードウェアの結合法は同じでないが、分散と集中の調和の下に各種の交換機用実用マルチプロセッサが実現されていることがわかる。

### 3. 研究の概要

本論文は、1978年（昭和53年）よりN T T電気通信研究所で開始されたデジタル交換機所内試験機D T S - 1からD 7 0デジタル交換機の実用化に至るデジタル交換機用マルチプロセッサ制御方式の研究内容を主にハードウェア面からまとめたものである。

本論文では、まずデジタル交換機におけるマルチプロセッサの位置づけを示す。次に要求条件への適合化へ向けたマルチプロセッサ構成決定への手順を示し、望ましいマルチプロセッサ構成を規定する。さらにこの構成を実際のシステムとして具現する設計思想を構築し、この具体設計を行い、評価を行なう。なお、交換機の具体的設計は、まず、故障を意識せずに効率、性能を最大限に向上させる基本設計を行い、その後、故障時の問題をどのように解決するかの過程を経る。本論文では前者を基本設計と呼び、後者を高信頼設計と呼ぶ。

本論文の内容を以下に要約する。

#### 3.1 デジタル交換機におけるマルチプロセッサの位置づけ

デジタルスイッチの特徴を分析し、その特徴を活かすためのデジタル交換方式への要求条件を求め、制御方式には以下のようにマルチプロセッサ制御方式が必要であることを示す。

##### (i) 大処理能力の実現

デジタル交換機導入の最終の目標は加入者から加入者まであらゆる情報をデジタル一元化して統一的に伝達し得るデジタル統合網の確立である。しかしデジタル交換機といえども当面はアナログ通信を含む既存通信網の中に埋め込みで導入せざるをえない。デジタル交換機の特徴を活かす当面の目標は交換機容量の拡大による既存通信網の経済化にある。この目標に対し、シングル

プロセッサ制御方式では能力不足となり、複数プロセッサによる処理能力拡大が可能なマルチプロセッサ制御方式が求められる。

#### (ii) 制御規模に対応した処理能力の提供

交換機には過疎地用の小規模から大都市用超大規模まで幅広い分布がある。このため、適用規模に応じた制御能力の提供が必要である。また、将来のデジタル総合網構築には様々なサービスの展開がなされると想定される。しかし、サービスの拡大は段階的になされ、高度サービスを提供しない初期に予め処理能力の設備を具備しておくことは不経済になる。規模や時期の状況に応じた処理能力の提供が必要であり、プロセッサ数の調整により処理能力調整が可能なマルチプロセッサ制御方式は今後の交換機制御方式として望ましい。

#### (iii) 交換機系列の統一

デジタル総合網の確立に向け、ソフトウェアの重要性は今後益々増大する。ソフトウェアの維持管理は今後大きな問題である。幅広い適用領域の全領域にわたって一機種化を可能とするマルチプロセッサ制御方式は全適用規模に渡ってソフトウェアの維持管理の統一を可能とすることから望ましい方式である。

#### (iv) 技術動向への対応性

デジタル交換機は高集積部品技術の活用の上に成り立つものである。今後進展する経済性に優れたVLSIプロセッサの利用は必須である。しかし、VLSIプロセッサは比較的low能力であり、複数プロセッサを組み合わせることで処理能力の拡大が可能なマルチプロセッサ制御方式は望ましい。

#### (v) 規模別信頼度への対応

デジタル交換機では社会的影響を考慮して、大規模交換機ほど高信頼であるべきとしている。一方、ハードウェア実現上では大規模ほど信頼性が低下する。このため、故障範囲を限定化し、システム全体では高信頼化を図り得る可能性のあるマルチプロセッサ制御方式は有効な方式である。

### 3.2 マルチプロセッサ構成法

交換処理の定量解析からデジタル交換機に適したマルチプロセッサ構成を求めている。主な内容は以下の通りである。

#### (i) 共通メモリによる資源管理

交換機は生起する呼に通信網の資源を管理分配して通信網の運用を司るものである。有限のサービス資源を如何に効率よく運用するかが交換機の経済性にとって最も重要な要因であり、共通資源の管理運用の効率化を検討している。この中で共通メモリによるサービス資源の管理方式が効率性において優れていることを定量的に求めている。

#### (ii) 信号処理と呼制御処理のモジュール化

交換処理は ①ハードウェアと対応する信号処理、②リソース管理を中心とした論理処理である呼制御処理に大きくは機能分解できる。この機能分解に基づいてプロセッサを付与し、機能分散を行なうと、新信号方式の導入は信号処理を扱うプロセッサ群の階層、また、通信網のリソースを利用するサービス追加は呼制御処理へと機能追加のモジュール化が図れる。この機能分割において信号処理は制御対象とする通話路ハードウェアと一対一に対応して設け、呼制御処理はハードウェアとは独立にすべての信号処理からの処理依頼を均質に負荷分散することが有効であることを定量的に求めている。

#### (iii) 信号方式対応機能分割

(ii)の分割において、呼制御処理は均質な負荷分散をしており、もはや機能分割は不要である。一方信号処理は各種の異なった信号方式を対象としており、またデジタル統合通信網の確立へ向けて新たな信号方式の出現を想定しなければならず、各種信号方式に対応した機能分散の可能性がある。また将来の新信号方式の追加可能性は必須であるが、一方では設備されない初期投資時点に将来の信号方式追加のための設備を具備しておくことは不経済となる。これらを考慮すると信号処理は各信号方式に対応して機能分散を行うことが必要であり、この効果を定量的に求めている。

### 3.3 マルチプロセッサの基本設計

望ましいとしたマルチプロセッサ基本構成を具体化する設計理念の構築と、この理念の下でのシステムの具体的設計と実現結果の評価を示す。

#### (i) 設計理念の構築

マルチプロセッサの具体化に当たっては、ONE-MACHINE 概念と呼ぶ設計理念を

用いる。マルチプロセッサは大規模への拡張性は優れているが、1プロセッサで実現できる程度の小規模の範囲ではシングルプロセッサ制御方式に経済性で劣るため、小規模領域ではシングルプロセッサ制御方式を採用せざるを得ない。

この ONE-MACHINE 概念を制御系を階層構成化して、各階層インタフェースをシングル/マルチプロセッサ両構成間で共通化し、各階層での構成要素を両系間で極力共通化する概念として具体化する。またこの概念は複雑化しやすいマルチプロセッサに対して構成要素のモジュール化を推進し、マルチプロセッサのシングルプロセッサ並みへの単純化を狙うものである。

#### (ii) システムの階層化

システムの構成要素を、通話路駆動階層、呼処理プログラム階層、論理マシン階層と実マシン階層の4層に分け、各階層におけるシングル/マルチプロセッサ両制御系間の共用化の条件を求め、各階層構成の具体化条件を求めている。各階層のシングル/マルチプロセッサ共用化を追求した結果、最終的には実マシン階層に入出力付加機構とプロセッサ間通信機構がマルチプロセッサ用付加機構として必要であることを示している。

#### (iii) 既存交換処理用プロセッサによるマルチプロセッサの実現

ONE-MACHINE 概念に基づき、シングルプロセッサ制御方式の構成要素を極力流用して実現するマルチプロセッサの具体設計を示す。既存の改良 D 2 0 プロセッサを核とし、(ii)で述べたマルチプロセッサ用付加機構を具体化して交換機用マルチプロセッサ技術を確立した。性能面では16台のプロセッサでほぼ12台相当の能力を実現している。特にマルチプロセッサ用付加機構としてハードウェアによる自律的な負荷分散機構を具備したプロセッサ間通信手段を提案し、ONE-MACHINE 概念の有効化を図っている。

#### (V) カスタム V L S I プロセッサの開発とマルチプロセッサへの適用

本マルチプロセッサ構成の本格実用化へ向け、開発した高性能な32ビットカスタム V L S I プロセッサの実現とマルチプロセッサへの適用及びその評価を示している。まず、汎用マイクロプロセッサが進歩した段階でのカスタムプロセッサの有効性を議論し、32ビットカスタム V L S I プロセッサの実現技術を示す。マルチプロセッサへの適用を図り、通話路容量に対して2倍の制御能力の能力余裕を備えた 7 5 0 k BHCA(Busy Hour Calls Attempt)性能の達成や ONE-

MACHINE 概念の達成度を評価している。ソフトウェアの観点では重要な要素を占め、変更の可能性の大きい呼処理プログラム類がシングル/マルチプロセッサの間で80%共用可能とし、ONE-MACHINE 概念が有効であったことを示している。ハードウェアでは小規模ではマルチプロセッサ用の付加機構が大きな比重を占めるが、大規模マルチプロセッサでは相対的に付加部の比率が低下し、マルチプロセッサは大規模向きの方式であることを確認している。

### 3.4 マルチプロセッサの高信頼設計

ここでは基本設計を支える高信頼設計について、主にシステムの構成面から示している。

#### (i) デジタル交換機用マルチプロセッサの高信頼化条件と設計理念

従来、すべての加入者に固定的なサービス性を与える考えから、交換機には規模によらず一定の信頼度を規定している。しかし、社会活動の大規模化等から信頼性への期待は規模により異なるとの定量評価に基づき、デジタル交換機には規模別信頼度を規定した。一方、ハードウェアは本質的に大規模ほど低信頼であり、方式的にこの規定への適合化が必要となる。複数プロセッサで構成されるマルチプロセッサの故障に対しては、一部のプロセッサの故障によってシステム全体が不稼働とならないフェイルソフトの考え方を交換機制御系に適用して高信頼設計することとする。

#### (ii) フェイルソフト手法の対案

フェイルソフト手法は、一部に故障が生じてもサービスを実現できる系構成が存在する限り、この系でサービスを出来る範囲で継続するものである。実現法として、ソフトウェアを正常時の配置と変える再配置法と正常時の配置のままサービス実現を図る固定配置法が考えられることを述べ、交換機には固定配置法により実現すべきことを述べる。

#### (iii) フェイルソフト手法の実現

固定配置法によるフェイルソフトの実現には、負荷分散の考え方が必要であることを示す。ここで実現するマルチプロセッサには、信号処理階層の静的負荷分散と呼制御階層の動的負荷分散があり、それぞれにフェイルソフトの実現条件と実現手法を示している。

#### (iv) 評価

既存のシングルプロセッサ制御方式との対比から、フェイルソフトを実現したマルチプロセッサがシングルプロセッサ制御方式より高信頼となっていること、さらに規模別信頼度規定に適合していることを示している。

### 3.5 マルチプロセッサの故障回復手法

(4)で述べた高信頼構成に故障が生じたとき、ハードウェアとソフトウェアが協調して故障からの回復を行なう動作すなわち故障回復法を示している。

#### (i) 階層構成に基づく故障処理

シングルプロセッサ制御方式の故障回復法は既に空間分割形電子交換機において確立している。しかし複数プロセッサが互いに影響しあう可能性のあるマルチプロセッサではシングルプロセッサ用の単純な故障回復法では解決出来ない問題が存在する。このため、複数プロセッサ制御を考慮して故障回復の階層化思想を構築し、故障からの確実な回復の考え方を示す。

#### (ii) 故障回復処理の手順と中断時間

上記の階層化思想に基づくマルチプロセッサの故障処理からの回復手法を示し、故障の種類と故障の波及により影響範囲が異なることを示し、サービスに与える影響度を故障回復処理によるサービス中断時間により評価する。

#### (iii) 評価

シングルプロセッサとの対比により、本論文で示す故障回復手法が故障回復に係わる中断時間を少なくでき、サービスへの影響を少なく出来ることを示す。

### 3.6 交換機制御方式の今後の動向

集積回路の超大規模化が進行しており、このインパクトが今後の通信網の改革に有効であることを示す。また高集積技術を活かすシステム構成法を示す。また高速通信サービスが今後の通信網への要求条件と考えられ、高集積技術の活用による加入者網の分散化が必要であることを示し、加入者網の例と小規模加入者ノードの構築に向けたシステム確立技術を示している。

## 第II章 デジタル交換機の特徴と制御方式への課題

### 1. 序言

デジタル交換機はデジタル伝送機能と結合して、電話、データ、ファクシミリ画像などのデジタル化された情報を 端末から端末まで そのままの情報形態で伝達出来るデジタル通信網、いわゆるデジタル統合通信網の構築を支える。しかし、このような夢を実現するデジタル交換機も まずは既存の通信網への埋め込みから出発しなければならない。既存のアナログ通信網は世界に広がった通信を可能とするのに、新たに独立したデジタル統合通信網を構築しても加入者の少ない間は通信相手が限定されるためメリットは少ない。一方、既存のアナログ通信網自身、既に伝送路単体では市外回線を主体にデジタル化が進行しており、デジタル交換機の導入によって既存網でも伝送路と交換機の一体化による経済化が図れる。

制御方式については、デジタル交換機といえども空間分割電子交換機と同じく蓄積プログラム制御方式が適用される。デジタル交換機のスイッチは、空間分割電子交換機と同じく他から制御を受ける受動素子にすぎず、制御システムからの援助によって機能が完成するものである。

既存網から将来のデジタル統合通信網への橋渡しをするデジタル交換機の制御方式には新たな展開が求められる。実用化されてから既に10年以上経過した既存の蓄積プログラム制御方式はソフトウェアへの機能追加や維持管理のため、年々多くの工数が費やされており、その削減を図る制御方式が求められている。また将来のデジタル統合通信網では通信網全体がシステム化され、各ノードには従来 of 空間分割電子交換機よりもはるかに高度な機能が要求され、大幅な制御能力の増強が求められる。

本章では次章以降の本題、すなわちデジタル交換機用制御方式の構築に向けた議論の導入部として、デジタル交換機への要求条件を概観する。まず実用化へ向けたデジタル交換機の位置づけを確認し、交換方式への要求条件を求める。次に交換方式への要求条件から制御方式への課題を求め、マルチプロセッサ制御方式がこれに適合し、有効であることを示す。

## 2. デジタルスイッチの特徴とデジタル交換機の方式目標

### 2.1 デジタルスイッチの特徴

デジタルスイッチはデジタル多重化線とデジタル多重化線の間でタイムスロットの情報を入れ替えることで交換を行なうものである。構成スイッチとして時間スイッチ（図2.1）と空間スイッチ（図2.2）がある。

時間スイッチはデジタル時系列情報を一旦メモリに格納し、読出しと書込みで順序を変更することによって入力と出力デジタル多重化線の中のタイムスロットの入れ替えを行なう。図2.1(a)において入力タイムスロット $a_{ij}$ と出力タイムスロット $b_{ik}$ において任意に $j$ と $k$ を置き換え可能とする。空間スイッチは同期した複数のデジタル多重化線の間で情報を入れ換える。図2.2(a)において、入り多重化線 $a_{ij}$ と $b_{kl}$ の間で同一位相 $j=l$ のもとで、任意の $i$ と $k$ を入れ換える。両スイッチを組み合わせることで $a_{ij}$ と $b_{kl}$ の間で任意の $i$ と $k$ 、 $j$ と $l$ を入れ換えることが可能で、極めて多くの選択の自由度がある。

このデジタルスイッチは、既存のクロスバ等の空間分割スイッチに比較して以下の特徴がある。

#### (1) デジタル一元化

従来の空間分割電子交換機では、中継系をデジタル化しても伝送端局でチャンネル毎に音声にデコードして交換し、再度、伝送端局で音声をデジタル情報にコード化することが必要であった。一方、デジタルスイッチはPCM情報を多重化ベースのままスイッチに直接入力し、交換することができる。このため、伝送路を直接交換機に引き込み、伝送／交換を一体としたシステム構成ができ、回路規模の削減や経済化が図れる。

またデジタル交換機は、入力デジタル情報をそのまま出力伝送路に送り出す。デジタル情報の伝達透明性（トランスペアレンシー）を有し、デジタル化した電話や画像通信やデータ通信などの各種メディアを情報の変形を伴わずに端末から端末まで通信可能とする。

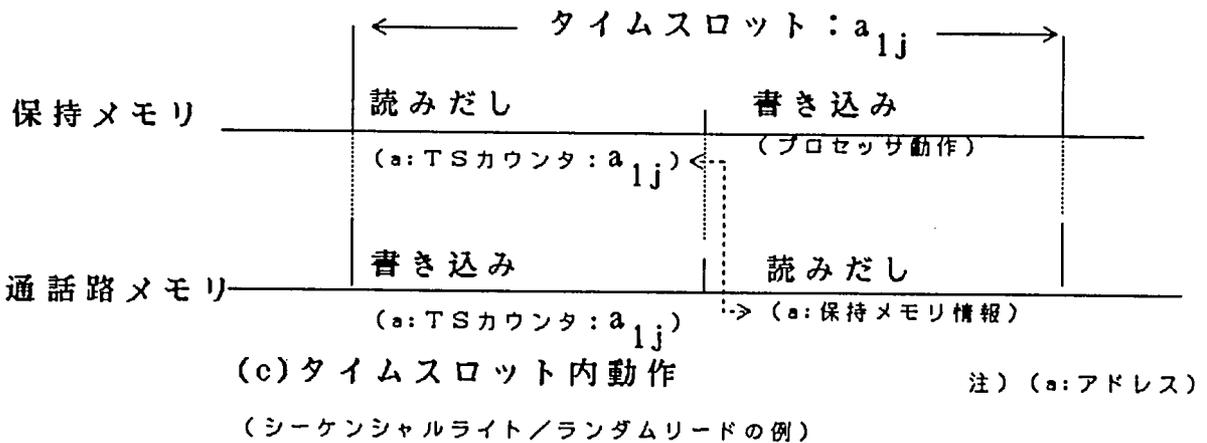
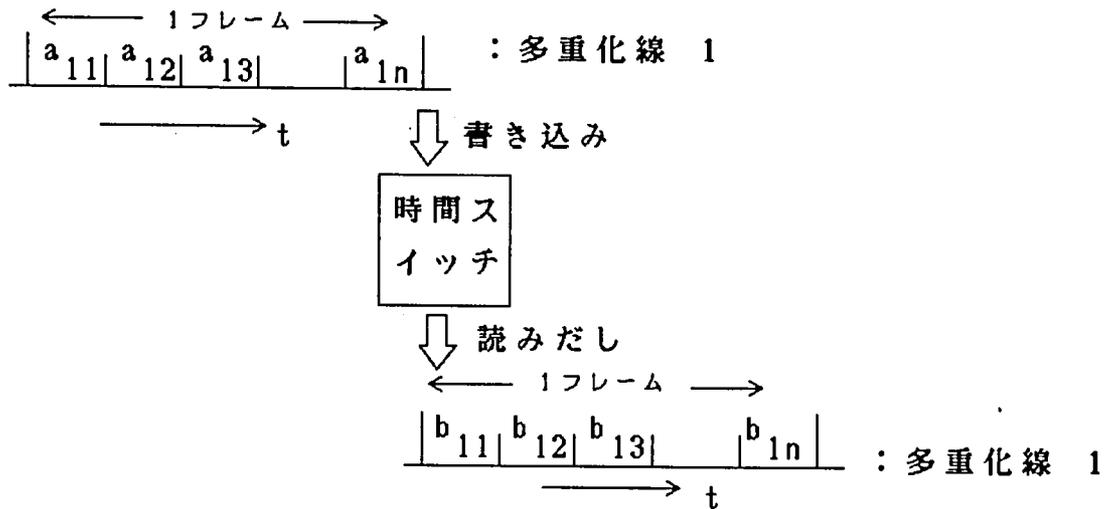
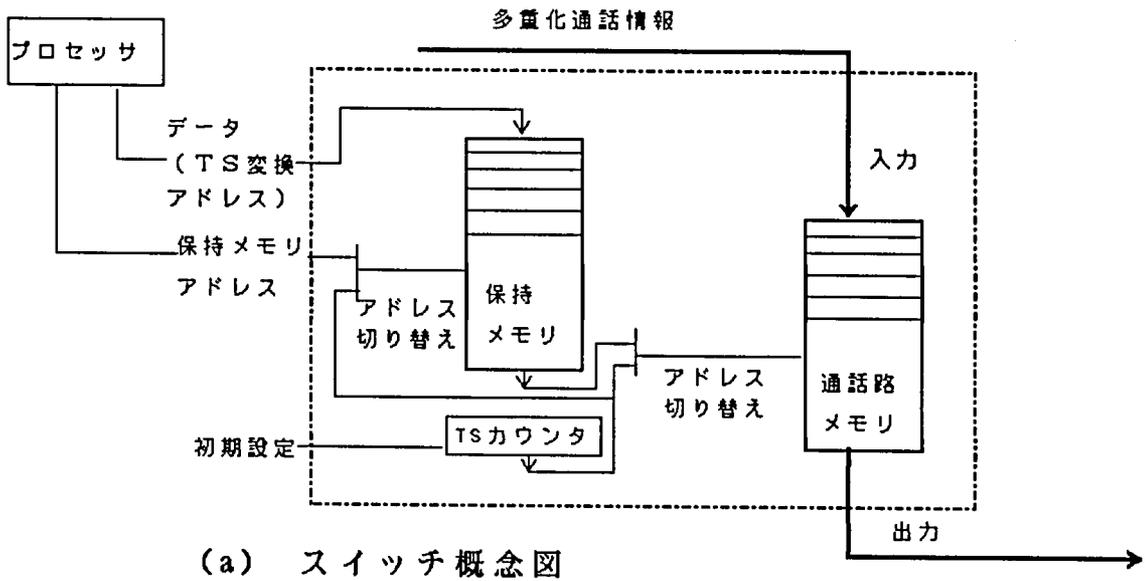
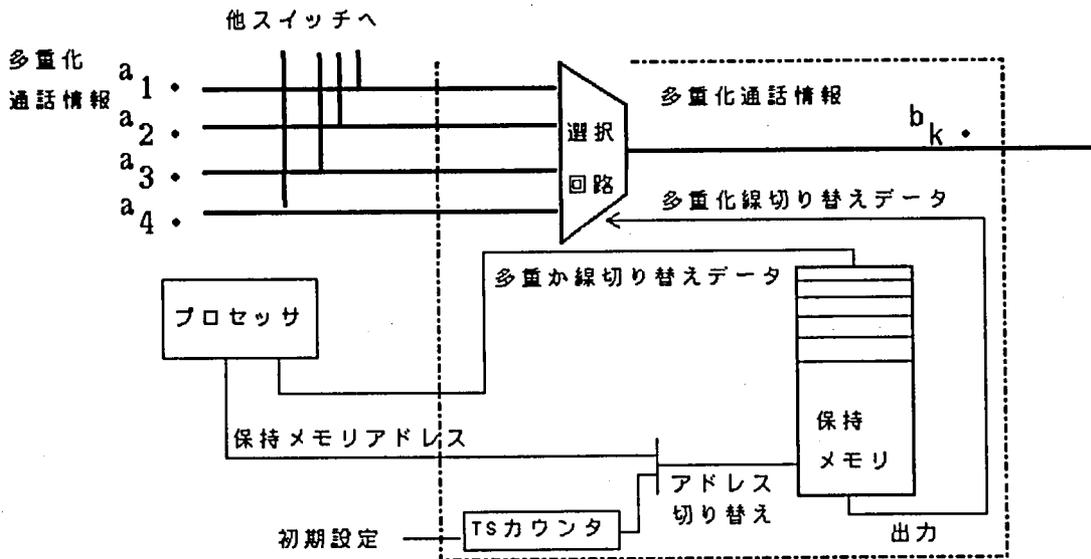
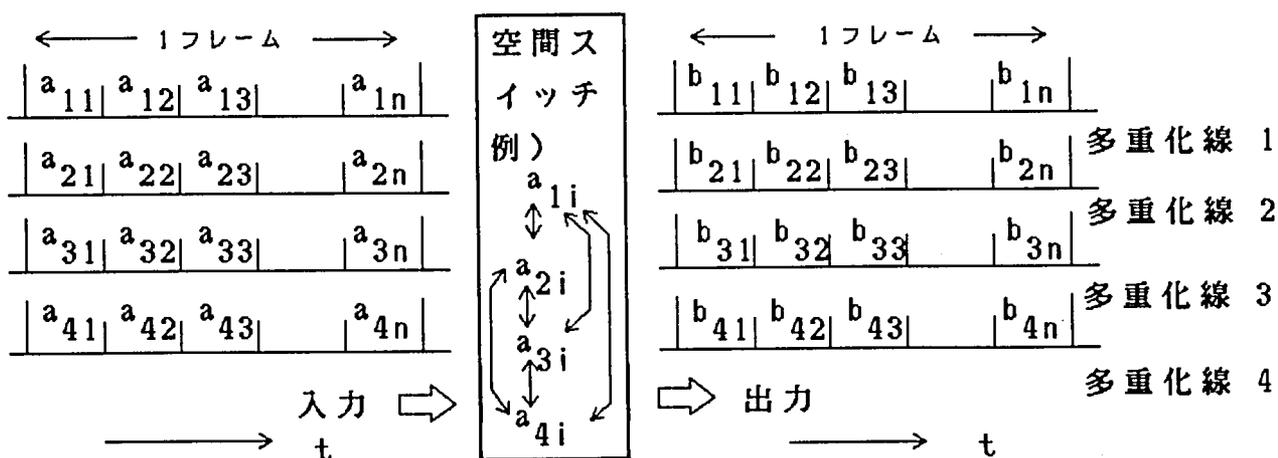


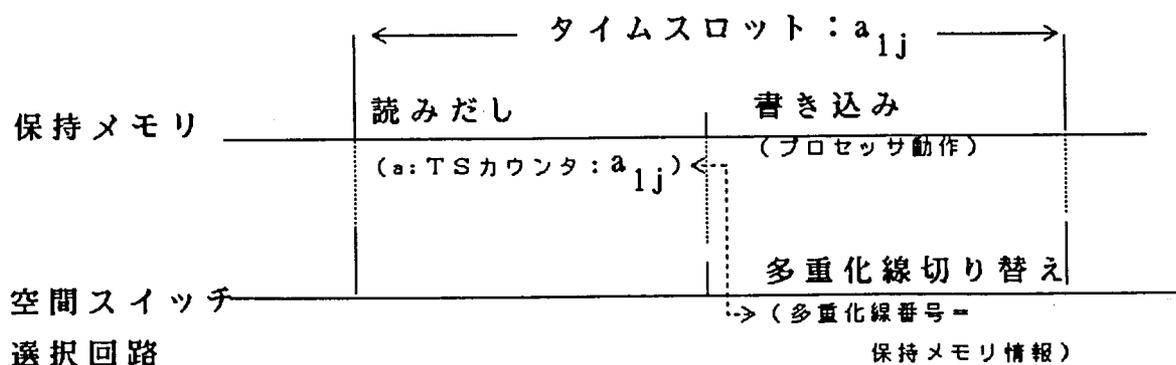
図2.1 デジタル時間スイッチの概念図



(a) スイッチ概念図



(b) タイムスロット動作



(c) タイムスロット内動作

図2.2 デジタル空間スイッチの概念図

## (2)高集積部品親和性

デジタルスイッチはデジタル用高集積部品を用いてコンパクトなスイッチに構成できる。例えば時間スイッチは、図2.1(a)に示すように保持メモリと通話メモリの組合せ、空間スイッチは図2.2(a)に示すように保持メモリと1/n選択回路で構成できる。また保持メモリへの制御はコード化されたアドレス情報を扱い、通話情報もデジタル多重化情報を扱うため入出力用物理線数が少なく、スイッチ全体をを一体IC化した場合にはゲート/ピン比を大きくとれる。またメモリを主体とした回路であるため、回路が整然としており、高集積部品に適している。

## (3)大容量単位スイッチの実現

デジタル多重化により、スイッチ内伝達情報や制御情報はコード化されており、また各回路は時間多重で利用されるため回路の利用率が高く、さらにスイッチは高集積部品で実現されるため、回路規模は空間分割形スイッチと異なり交換容量の増加の影響を受けない。このため比較的規模の大きな単位スイッチを経済的に実現できる。

## 2.2 デジタルスイッチのシステム化とその特徴

デジタル交換機を構成する場合、上記の単位スイッチを組み合わせてシステム化するが、システム化の観点からは更に以下の特徴を生む。

### (1)交換機の大容量化

システムとしてのスイッチの最大容量は、単位スイッチの容量と結合リンクの容量で決まる。ところがデジタル単位スイッチは高集積部品を用いることにより、空間分割スイッチより経済的に大容量化できる。またスイッチ間の結合リンクもデジタル時分割多重化線を利用することにより線数が少なくても大束なリンクが設定できる。従ってデジタルスイッチはシステム化の段階でも経済的に大容量化できる特徴を持つ。

またデジタルスイッチは空間スイッチの組合せに加え、時間スイッチによるタイムスロット位置の変更が可能なことから、入り回線と出回線との間の経路選

択の自由度は大きいという特徴がある。この選択の自由度は空間分割形スイッチよりはるかに大きい。選択の自由度が大きいことによりスイッチの内部呼損率を小さくできる。トラヒック面からも大容量化に適している。

## (2)ビルディングブロック化

デジタル交換機では、単位スイッチを組み合わせて所望の容量を実現するビルディングブロック化が容易である。ビルディングブロック化には、スイッチ間の結合リンクを予め最大規模まで準備しておくことが必要であるが、高多重のリンクを用いることにより、最大規模のリンクを予め準備しておいても小規模時点での負担は大きくはない。

スイッチのビルディングブロック化は交換機の理想であったが、既存空間分割交換機では統一した通話路を構成することは困難であった。空間分割交換機のスイッチは、格子状に叉点を配列したスイッチを単位として、これを多段にリンク結線して構成される。ところが、このコストは叉点数にほぼ比例する。従って小規模局で大規模局へのスイッチの展開能力を予め具備しておくのは不経済であった。小規模局と大規模局では必然的に格子の展開段数が異なってしまった。例えば大規模用D10交換機<sup>(4)</sup>では8×8スイッチを8段接続、中小規模局用D20交換機<sup>(11)</sup>では6段接続としていた。

## (3)高信頼化

デジタル通信は情報伝達に対する信頼性が高い。品質劣化は符号誤りのみであり、アナログ通信網のような距離対応の雑音やレベル変動を生じないという特徴がある。

さらに、デジタル交換機では予備の具備範囲をスイッチにまで拡大可能であり、高信頼化が図れる特徴がある。空間分割スイッチでは、スイッチ本体の回路規模は大きい上、予備を設けた場合には交差回路も音声回線毎に必要なになる。一方デジタル交換機では高集積部品の利用と時分割多重化線の利用により予備用回路も交差回路も小規模ですむ。

## 2.3 デジタル交換機の方式目標<sup>(49)</sup>～<sup>(51)</sup>

以上のデジタルスイッチの特徴からくるシーズ面及び通信網のおかれた状況、将来への展望等のニーズ面を勘案して、デジタル交換機実験機への目標設定がなされた。このうち、実験機を中心としたデジタル交換機の目標設定項目を以下にあげる。なお、デジタル交換機の導入目標はデジタル統合通信網の確立に先立ち、既存通信網の経済化を出来る範囲から進めることにある。これを狙いにデジタル交換機の導入を進め、デジタル統合網の布石を着実に進展させる。

### (1)通信網の経済性の向上

デジタル交換機導入の大きな課題は、まず、既存通信網の経済化と考えられる。デジタル交換機の特性を利用することにより、直接的にも、間接的にも経済化が期待できる。

#### (i)交換機大容量化による通信網の経済化と交換機規模の目標設定

##### 大容量化の効果

デジタルスイッチの特徴を利用して大容量デジタル交換機を実現すると、以下のように通信網の経済化が達成される。

#### (ア)回線利用の効率化

交換機の大容量化は同一対地に向かう回線群、すなわち同一方路へのトラヒックを増大させ、①同一方路への回線の大量化を可能として回線当りの利用効率を向上させる。②方路ごとに6回線単位で運用される回線の量子化損を削減させる。

#### (イ)斜め回線の増加

交換機の規模が小さい場合には6回線単位で運用される斜め回線用の呼量が集まらず、斜め回線を活用できないという問題がある。しかし交換機大容量化は同一方路へのトラヒックの集約効果をもたらし、経済的な斜め回線を増加させる。なお、斜め回線が張れない場合はEO(端局)、TC(集中局)、DC(中心局)RC(総括局)の4階位で構成される基幹回線に迂回するが、基幹回線では多段の交換機を経由せざるを得ないため、少なくとも迂回交換機のための交換機コス

トの増加を招いている。

#### (ウ)トラヒック不平衡の均衡化

近年の社会生活の変化は昼間の大都市間トラヒックや夜間の遠距離トラヒックの増大というトラヒック不均衡を招いている。このため、両者の最大値を考慮した回線設定を行なう必要があり、通信網の効率が低下するという問題を生じている。しかしこのトラヒック不平衡による能率低下は階梯併合を含む交換機の大容量化により解決できる。

#### (I)マルチユニット局の併合

複数交換機を同一局舎に併設したマルチユニット局が大都市を中心に多数存在する。交換機の大容量化が可能となると複数ユニットを単ユニット化することが可能となり、ユニットごとに必要な入出力装置などの同種装置を併合できる。

#### 最大局規模の目標

以上のように、中継系の大容量化はデジタル交換機への最大の要求条件であるが、現在の多ユニット局に適用すると中継線および交換機の大群化効果による全国市外網コスト削減効果は、交換機規模1万～2万アーラン規模で飽和する。従って実現するデジタル中継線交換機では、その方式容量目標値を2万アーラン（商用実用機では9,800アーラン）と設定する。

なお加入者線交換機は既存交換機の置き換えと想定され、既存交換機の規模分布に従った値として良い。従って加入者線交換機の方式容量の目標値は空間分割電子交換機と同程度の4,800アーランに設定する。

#### (ii)デジタル高集積部品の利用によるハードウェアの経済化

##### ハードウェア量の削減

デジタル信号を扱うことによりLSI適合化が増し、進展する高集積部品技術の活用による経済化が可能となる。また、デジタル伝送線を交換機と直結化することにより経済性向上が図れる。デジタル交換機の実用化に向けては高集積部品の利用を推進し、活用を図る。

##### 全容量域での1機種化

デジタル交換機では特にスイッチ面からビルディングブロック化が容易という特徴がある。一方、空間分割電子交換機では大容量、中小容量、小容量の適

用領域毎にD10, D20, D30の3機種 of 交換機を設けていた。交換機を規模毎に複数種設けるとソフトウェアの維持管理が機種ごとに必要となり、維持管理の経費の増大を招く。

デジタル統合網の構築に向けては、空間分割電子交換機よりもはるかに大規模なソフトウェアが必要となることが予想でき、その維持管理が大きな問題となることが予想される。機種統一は必須の条件である。デジタルスイッチの特徴を利用し、交換機の全容量域を1系列で対応する。

## (2)品質, 信頼性の向上

### 高品質化

デジタル化による当然の帰結であるが、端末から端末までデジタル1リンク化が実現すると、伝送品質の劣化は符号誤りのみで、アナログ回線のような距離対応の雑音やレベル変動を生じない。途中の中継回線や交換機にアナログ信号を含まない市外系1リンク網が完成するだけでも端局相互間の損失を現行10dBから6dBに低減できる。既存網への埋め込みにおいてもデジタル1リンク網を部分的であれ早期に実現する。

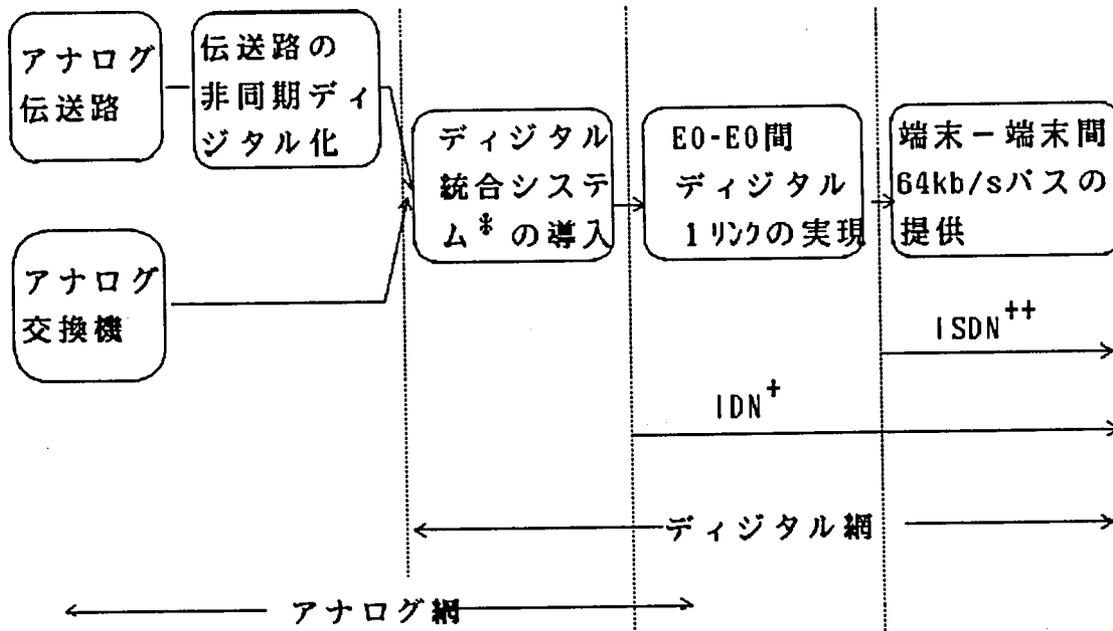
### 高信頼化

従来は同じ通信網に収容されている加入者には、一律に同一信頼度を実現すべきの考え方があり、交換機では規模によらず一律固定的な信頼度規定を行なってきた。社会生活の規模の拡大や社会生活の道具として通信網の重要性が増大しており、システム停止による社会的な迷惑度は、システム停止による影響範囲によって変わるとの考え方は有効である。デジタル交換機においては、交換機の規模ごとに異なった信頼度規定を行なう。

## (3)デジタル交換機の導入目標

### (i)既存通信網とデジタル交換機

通信網のデジタル化は図2.3に示すように、中継線から次第に加入者線へ進み、最終過程としてデジタル統合通信網が実現する。一方、伝送路のデジタル化は大東な市外回線から進行している。またデジタル交換機が大容量域で一層有利となることから、デジタル交換機は大都市相互の市外網から導入される



- \* デジタル統合システム：デジタル多重インタフェースで結合された同期系デジタル伝送端局とデジタル交換機の総称
- + IDN : Integrated Digital Network
- ++ ISDN : Integrated Service Digital Network

図2.3 電話網デジタル化の進展

と想定される。

上位の階梯がすべてデジタル形式で接続可能となると交換機端局（EO）相互間のデジタル1リンクがまず可能となり、その後、加入者線までデジタル化が進行する。端末までデジタル化が進行すると端末相互間が64kb/sパスを提供できるようになる。さらに、これを複数個まとめてユーザに提供する多元サービスが提供できる。伝達すべき情報は音声に限らずデジタルファクシミリやデータなど多様になり、デジタル情報を扱う端末をそのまま収容でき、新しいデジタルサービスを要求するユーザに効率的に対処できる。

#### デジタル交換機導入の位置づけ

デジタル交換機には、将来の機能高度化に備えた機能拡張性が必要である。

しかし、デジタル交換機の当面の目標は既存網の経済化であり、導入初期にはデジタル一元サービスは開始されない。初期段階ではこのサービスを前提としない経済性が必要である。

#### (ii) デジタル交換機の網構成

社会生活における通信網の神経中枢機能的な役割は益々高まってきており、通信網の信頼性向上や障害・災害対策やトラヒック変動ならびに異常トラヒックに対する融通性が重要になってきている。特に通信サービスの多様化や加入者習性の変化や住宅電話の普及など潜在的な異常トラヒック発生源が増大している。トラヒック変動要因が次第に大きくなって来ていることから、通信網の性能や機能の動的適合化が重要である。

より具体的には、デジタル通信網には伝送路の設定を動的に変更する可変トポロジーやトラヒックの変動に応じて動的にルートを変更する可変ルーティングやトラヒックに応じて利用帯域を変更し得る可変容量などの柔軟性のある網への進展を考慮しておくことが必要である。

#### 網構成の柔軟性を意識した制御能力の付与

デジタル網では、幸い、交換点での呼損率を小さくでき、回線の分岐挿入が容易であることから、制御を高度化することにより柔軟な網を構成できる。ただし、この機能高度化は主に制御能力に負うところが大きく、制御システムには能力の余裕が必要である。なお、ここでも不要な段階での初期設備投資は極力抑えなければならない。

### 2.4 デジタル交換機における制御方式の目標<sup>(52)</sup>

以上のデジタル交換機の方式目標は、制御方式に対して以下の目標に具体化できる(図2.4)。

#### (1) 制御能力の拡大

デジタル交換機では最大容量を2万アーランと設定している。これは既存の最大規模の電子交換機の5倍に相当する処理能力拡大が必要である。また、将来のデジタル統合網の構築に向けて大幅な機能追加が予想され、処理量の増大が予想される。この処理能力は導入初期に必要なものでなく、サービスの高

既存通信網の改良

【通信網からの要求条件】

- ・通信網の経済性向上
  - 大容量交換機の実現
  - ハードウェア量の削減
- ・品質や信頼性の向上
  - 規模別信頼度規定への適合
- ・柔軟な網構成
  - 制御能力余裕の具備

デジタル統合網への布石

- ・将来の機能拡大性の具備



【交換方式への要求条件】

交換機基本条件

- ・大容量化
- ・高信頼化（規模別信頼度）
- ・一列機種化
- ・処理能力拡大容易性
- ・初期設備経済性

デジタルの特徴利用

- ・デジタル一元化
- ・ビルディングブロック化
- ・高集積部品利用

【デジタルスイッチの特徴】

- ・デジタル一元化
  - 伝送・交換の統合
  - 高品質
- ・高集積部品親和性
  - 交換機の経済化
  - ビルディングブロック化容易
- ・大容量化容易



【制御方式への要求条件】

- ・大容量制御能力
- ・ビルディングブロック制御系
- ・機能追加容易化
  - 機能分散・モジュール化
- ・規模増大に対応する高信頼化
  - フェイルソフトの確立
- ・VLSIプロセッサの利用



マルチプロセッサ  
制御方式の確立

図2.4 デジタル交換機制御方式への要求条件

度化に併せて追加していけることが必要である。

## (2) 交換機1機種化に合わせたビルディングブロック制御方式の実現

デジタル交換機制御方式には幅広い交換機の適用領域への適合化が必要である。また将来のデジタル総合網へ向けた機能追加に対し、柔軟に処理能力を拡大できる制御方式が求められる。特にデジタル通話路がビルディングブロック化を可能とすることから、通話路系に併せてビルディングブロック化が可能な制御系が求められる。

## (3) サービス総合化へ向けた機能追加の容易化

デジタル統合によるサービス総合化ではソフトウェアの大規模化がますます進展する。ソフトウェアの機能追加や維持管理を容易とするソフトウェアのモジュール化の推進が必要となる。

## (4) 規模別信頼度への対応

新たに規定された交換機の規模別の信頼度の規定を制御方式も充足することが必要である。ただし、ハードウェアは、本来、規模の拡大とともに逆に信頼度が低下する傾向がある。規模別信頼度規定への対応に向け、制御単位を小規模化して故障頻度を下げ、故障発生を分散し、かつ、故障波及範囲を限定化できる方式の開発が必要となる。

## (5) アーキテクチャの統一

空間分割電子交換機以来、ソフトウェアの維持管理工数の削減は大きな課題とされ、複数系列のソフトウェアの維持管理を一元化することが最も維持管理工数の削減に有効である。制御系においても、規模によらずに1種のプロセッサアーキテクチャで統一する事が目標となる。

## (6) VLSIプロセッサの利用

デジタル交換機は高集積部品との親和性が高く、高集積部品によって経済的に構築されるものである。制御系においても高集積部品を活用した経済的なプロセッサを利用することが必要である。

## 2.5 デジタル交換機制御方式におけるマルチプロセッサの採用<sup>[53]～[57]</sup>

### 2.5.1 マルチプロセッサの有効性

制御方式の目標からデジタル交換機制御方式をシングルプロセッサ制御方式のみで実現することは困難であり、マルチプロセッサへの移行は必須と予想される。ここではシングルプロセッサ制御方式に対するマルチプロセッサの有効性を整理する。

#### (1)大容量制御能力の実現

##### シングルプロセッサの問題

プロセッサ単体では部品技術の進展、キャッシュメモリ等のプロセッサの制御構成の改良により着実に処理能力が向上している。しかし、プロセッサ1台で制御するシングルプロセッサ方式単独による処理能力の向上は以下の問題を抱えている。

##### 制御方式における確率利用の問題

単体プロセッサの処理能力向上を極限まで追求するには、キャッシュメモリのようにソフトウェアの確率的特性を利用した高速化が必須となる。キャッシュメモリは命令やデータが利用時に高速なキャッシュメモリに存在するヒット率を向上させて、メモリの等価的速度を向上させ、処理能力の向上を図るものである。通常のプログラムの振舞いでは処理能力を向上させ得ても、異常トラヒックが発生すると常時とは異なったプログラムが走行することを考慮しなければならない。この場合、処理能力低下を考慮しておくことが安全設計である。

一方、制御系の対象とする通話路はトラヒック理論により設計されたものであり、確率を考慮して通話路や回線の経済化を図っている確率システムの典型である。通話路系への確率導入の効果はきわめて大きく、異常が生じた場合には制御で対処されることを前提に設計されている。この観点からは異常時にこそ制御能力の確保が必要であることを考慮しておかなければならない。確率を利用したキャッシュメモリの交換機制御システムへの導入は、通話路の確率要素と併せて通信網を混乱に招く危険がある。異常時のキャッシュメモリの振舞いを明確に出来ない場合、キャッシュメモリの効果を過大評価することは危険であり、キャッシュメモリの導入にはヒット率をかなり安全側に設定し、能力評価を行な

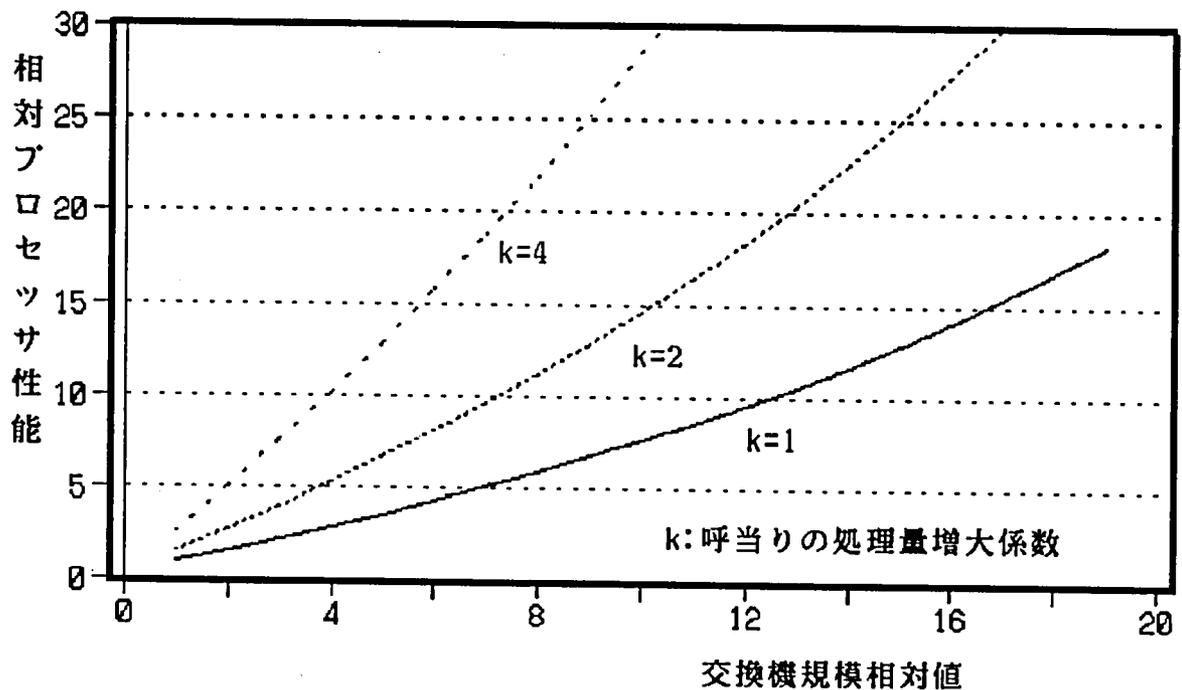


図2.5 交換機規模と所要処理能力

う必要が有ろう。 シングルプロセッサの能力はかなり低下することを覚悟しなければならない。

#### 通話路アクセスによる処理時間無効化

交換処理では通話路の状態をメモリに写像して処理するため、プロセッサは生起呼の有無に関わらず通話路にアクセスする。 この通話路のアクセス時間はプロセッサの命令実行時間に関わらず比較的低速で一定である。 交換機規模と必要処理能力の関係が図2.5の非線形性で示されるように交換機の規模増大により無効処理が増大し、シングルプロセッサの能力が高速になっても有効に活かしきれない問題がある。 制御対象とする通話路を限定し、プロセッサと通話路を対応付けて分散し、通話路アクセスの無効時間を短縮することを考えなければならない。

#### マルチプロセッサの有効性

マルチプロセッサは言うまでもなく、複数プロセッサの協調により処理能力を向上するものであり、大処理能力を実現できる。 ただし、交換処理には汎用プ

ロセッサで処理される数値演算や事務処理となど一般的情報処理とは異なった性質があり、単純なプロセッサの組合せで必要とされる処理能力を実現し得るものではない。交換処理へのマルチプロセッサの適合化が本論文の最大の課題である。

#### (2) プロセッサ実装数の可変性を利用した設備投資の最適化

マルチプロセッサは交換機の規模による処理量の相違や、さらに規模が同一でも、今後生じるサービス追加やソフトウェアの構造改革等による処理量の増大に設備プロセッサ数により調整可能である。しかも、プロセッサを追加可能としておけばシングルプロセッサのように導入初期からその能力を持つのでなく、必要に応じて処理能力の拡大が可能となる。投資繰延べが図れる。

#### (3) 小規模 VLSI プロセッサの利用

マルチプロセッサ制御方式はプロセッサ実装数の可変性を利用した処理能力の拡大が可能であることから、比較的能力の小さいプロセッサを利用することが可能である。今後進展するコストパフォーマンスに優れた高性能マイクロプロセッサの利用が可能となる。

#### (4) 高信頼化

マルチプロセッサは、小規模で高信頼なプロセッサの組合せでシステムを構成できる。したがって構成要素が比較的高信頼で、しかも故障が生じても影響を一部のプロセッサに限定できる可能性がある。デジタル交換機の信頼度規定には大規模交換機ほど高信頼であるべきとする規模別信頼度規定が適用されているが、このマルチプロセッサの特徴を活かし、ソフトウェアで故障影響を限定するフェイルソフト手法を確立すれば規模別信頼度規定に対応できる。

#### (5) ソフトウェアのモジュール化

マルチプロセッサでは複数プロセッサ間の物理境界が存在し、ソフトウェアを分散し、モジュール化を推進する強制力が期待できる。今後、デジタル統合網の確立に向けてソフトウェアの機能追加を極力円滑に進める必要があるが、ハ

ードウェアモジュールとソフトウェアモジュールとの対応化によるソフトウェアのモジュール化が期待出来る。

### 2.5.2 マルチプロセッサの実現条件

前記のマルチプロセッサの有効性に鑑み、デジタル交換機用制御方式として望ましいマルチプロセッサ制御方式を確立する必要がある。この場合、処理能力や信頼性などの基本要件は当然満たされなければならないが、より有効なマルチプロセッサを実現するため、以下の点を目標として、次章以降のマルチプロセッサの検討及び設計条件とする。

#### (1) シングルプロセッサ並みの単純マルチプロセッサ

マルチプロセッサはシングルプロセッサで実現していた機能を複数プロセッサに展開するものである。しかも、複数プロセッサは互いに協調しあいながら、かつ互いに並列に動作することで処理能力を向上する。多数のプロセッサの組合せおよびプロセッサ間の並列動作などからマルチプロセッサはシングルプロセッサに比べ複雑になるが、シングルプロセッサの単純性を参考にしつつ、極力単純化してマルチプロセッサ構成を規定することが重要と考えられる。

またシングルプロセッサは全ての制御構成の基本である。シングルプロセッサは簡明で汎用性に富むが、マルチプロセッサはこれより複雑化するのは必至である。シングルプロセッサ並みの簡明性を具備するマルチプロセッサを追求するためには、ソフトウェアから見て極力単純化を図り得るようマルチプロセッサ構成を隠すハードウェア支援を図ることが必要である。

#### (2) 既存ソフトウェアの構成に適合する機能分散化

デジタル交換機の確立には、膨大なソフトウェアの開発が必要となる。開発の早期化が求められており、しかも交換機用マルチプロセッサという新技術を開発しなければならない。単純なマルチプロセッサは既に常識技術とはいえ、交換処理はソフトウェア全体が一体化した大規模性や数万の端末を処理する多重処理性やハードウェア動作に追随する実時間処理性や24時間サービスの高信頼性など困難で特殊な問題がある。一方、既に確立した電子交換機ソフトウェア

では膨大な資産およびノウハウを蓄積しており、開発の早期達成を図るには既存資産の流用が重要となる。また、マルチプロセッサにはモジュール化や機能分散が必要であるが、新たに強引なモジュール化分解点を求めて混乱を起こすのではなく、もともとシングルプロセッサに存在する機能分解点をそのまま写像しうるマルチプロセッサを追求する。

### (3) 設備投資繰延べ可能な構成

デジタル交換機の当初の導入目的は既存電子交換機の置き換えであり、マルチプロセッサに求められる処理能力もそれほど大きくはない。しかし、将来はデジタル統合通信網へ向けた機能追加から、徐々に要求処理能力は拡大すると考えられる。この観点からは、デジタル統合網としての新しいサービスの開始以前の設備投資は極力少なくすることが求められる。

## 3. 結言

本章はデジタル交換制御方式の確立に向けた導入部であり、マルチプロセッサ制御方式の必要性と条件について述べた。本章で述べた内容をここで要約する。

まず、デジタル交換機制御用マルチプロセッサの開発は以下の点から必要である。

- (1) デジタル交換機の目指す最終目標はデジタル統合通信網の構築であり、これを考えるともはやシングルプロセッサの能力では対応しきれない。
- (2) マルチプロセッサは将来の不確定要因に対応してソフトウェア/ハードウェアの追加可能性を具備するものであり、今後進展するデジタル交換機の機能追加に対応するためにはマルチプロセッサが必須である。
- (3) ソフトウェアの維持管理は今後大きな問題であり、全規模にわたる維持管理の統一が必須であり、一系列で全規模の制御を実現可能とするマルチプロセッサ制御方式の確立が必要である。

(4) デジタル交換機は高集積部品技術の活用の上に成り立つものであり、性能は比較的低いが経済性に優れた V L S I プロセッサの利用を可能とすることからマルチプロセッサは必要である。

(5) デジタル交換機では規模別信頼度規定を要求しており、大規模における高信頼性を確保するには故障の分散と故障範囲の限定化を可能とするマルチプロセッサは必要である。

また、マルチプロセッサの確立にあたっては以下の点に留意する。

(1) シングルプロセッサは制御システムの出発点であり、シングルプロセッサ並みの単純性を実現し得るマルチプロセッサを追求する。また、これにはソフトウェアの単純化を考慮する必要がある、ハードウェアで極力シングルプロセッサ並みの簡明性を支援する。

(2) マルチプロセッサ制御方式の開発に当たっては、極力既存確立技術の流用を図ることが必要である。もともとシングルプロセッサに存在する機能分解点を素直に写像しうるマルチプロセッサハードウェア構成を追求する。

(3) マルチプロセッサの構成の決定に当たっては、将来の機能拡張性を具備しつつ、サービスを提供しない初期にはこの投資を必要としない投資繰延べ効果を考慮する。

# 第III章 デジタル交換機制御用マルチプロセッサの基本構成 <sup>[53][54]</sup>

## 1. 序言

前章ではデジタル交換機の制御方式への要求条件を求めた。この中で既存のシングルプロセッサ制御方式からマルチプロセッサ制御方式への展開が望まれることと、マルチプロセッサ制御方式への課題を求めた。本章ではこの課題に基づき、デジタル交換機用マルチプロセッサの望ましい構成を求める。

メモリ結合の単純なマルチプロセッサは汎用情報処理の分野で既に常識化した技術であるが、交換機制御用マルチプロセッサの確立には実時間、超多重処理、高信頼などの困難で新しい問題を解決しなければならない。この場合もマルチプロセッサには複数のプロセッサの組合せとして複雑なシステムの隘路があり、数多いマルチプロセッサの長所を単純には引き出すことは出来ない。性能に関してもプロセッサ間に負荷の不均衡を生じると一部のプロセッサに集中した負荷によってシステム全体の能力が制限される。またプロセッサ間の通信量が多いと純粋な処理より通信のための処理が増大して結果的に処理能力が向上しないという問題が生じる。さらにプロセッサ間の機能配分を適正化していくと、複数のプロセッサが共通にアクセスする共通リソースのネックによって処理能力が制限を受けるようになる。極力リソースのアクセスネックを少なくし、全体のシステムの能力向上が必要になる。

またマルチプロセッサの確立においては、応用システムの性質を分析し、処理の性質に適合化した機能配備を行うことが必要になる。デジタル交換機には将来のデジタル統合化やサービス総合化に向けて機能高度化への融通性を具備することが求められ、しかもこの機能拡張性はまだ既存サービスのみ段階での初期投資を犠牲にするものであってはならない。

本章ではマルチプロセッサの具体設計に先立ち、デジタル交換機用として望ましい構成を規定する。またこの望ましい構成の追求には、単純で最も基本的と考えられる構成から順次望ましい構成へ改良を加えていく手法をとる。なお構成の検討においては、正常時の処理能力の拡大や効率化に関する基本構成の検討と、故障時にもサービスを継続する高信頼構成の検討があるが、本章では前者の基本構成を検討し、高信頼構成については第V章以降で述べる。

## 2. 交換機制御用マルチプロセッサの基本系

ここではマルチプロセッサ構成の検討の出発点となる交換機制御用マルチプロセッサの基本系を求める。マルチプロセッサ基本系は最も単純なマルチプロセッサ形態であり、構成の変形化に耐える偏りのない形態でなければならない。

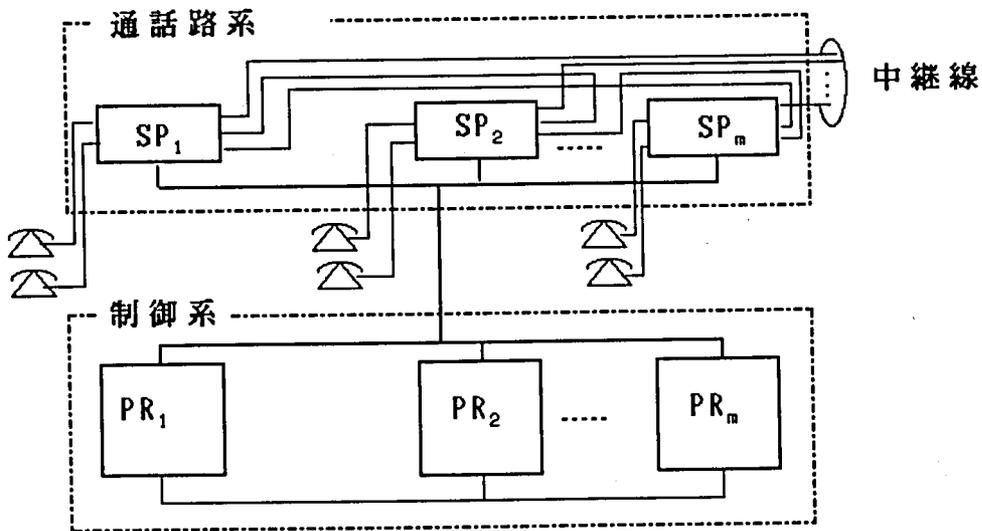
考えられるマルチプロセッサ基本系を図3.1に示す。マルチプロセッサ自身は単純にプロセッサを結合したものであるが、通話路とマルチプロセッサの対応には2種の対案が存在する。一方は従来シングルプロセッサと同様に通話路全体を制御システム全体で制御する通話路と制御の非対応方式(図3.1(a))であり、もう一方は図3.1(b)に示すように通話路のビルディングブロック単位毎にプロセッサを付加した構成である。

両構成においてマルチプロセッサの形態に差はないが、両者の差である通話路とプロセッサ結合方式は蓄積プログラム制御方式の能力に差を与える。蓄積プログラム制御方式は通話路の状態をプロセッサのメモリに写像し、これを加工して、設定すべき次の通話路状態を求め、この状態に至る制御指令を通話路に発する。常に通話路の状態に関する未加工の大量の情報をプロセッサに取り込む必要がある。この未加工情報の転送量は交換機の規模が大きくなるほど増大し、一元的に転送する図3.1(a)の通話路とマルチプロセッサの非対応方式では処理能力制限を与える。これを避けるためには、通話路の一部とプロセッサを対応付け、交換機の規模が大きくなっても未加工情報の転送経路を局所化して転送情報が規模によっても増大しない対策が必須となる。この観点から通話路ビルディングブロック単位とプロセッサを対応付けた単純な交換機のビルディングブロック形式(図3.1(b))をマルチプロセッサの基本系として設定する。

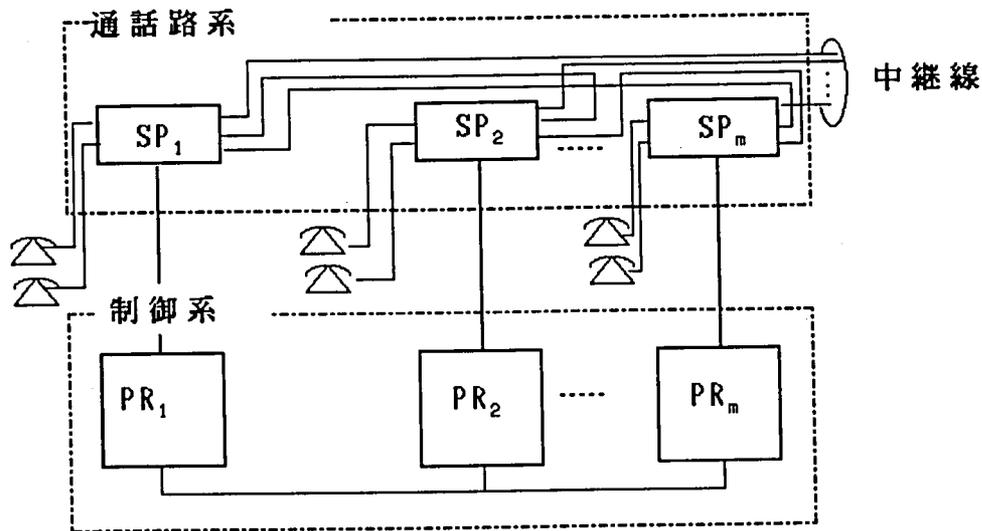
## 3. リソース管理方式とマルチプロセッサ構成

### 3.1 リソース管理のモデル化と対案

ここでは前節で規定したマルチプロセッサ基本系において、交換機のリソース管理をどのように付与するかを検討する。交換機は単に発呼者と着呼者の間のパスを設定するだけでなく、通話成立に必要な交換機および通信網リソースを効率よく管理して極力リソースの節約を図ることが使命である。



(a) 通話路系と制御系の独立構成



(b) 通話路ビルディングブロック単位とプロセッサの対応化

$SP_i$  : 通話路ビルディングブロック単位

$PR_i$  : プロセッサビルディングブロック単位

図3.1 交換機制御用マルチプロセッサの基本形

マルチプロセッサにおいても、通話成立のためのリソースをいかに管理するかが重要な検討課題である。ビルディングブロック単位にプロセッサを付加して並列に処理しても通話の成立に必要な回線等の通話用リソースは交換機全体で共有するものである。互いに並列に動作するプロセッサ間で、この共通リソースを確保するための処理に競合が生じる。従って、交換機の共通リソースをいかに管理するかが、マルチプロセッサ構成決定の第一段階である。

なお、マルチプロセッサ構成において、このリソース管理を集中するか分散するかの対案が考えられるが、ここではマルチプロセッサ基本系で示したようにビルディングブロック性の追求からリソース管理は特定のプロセッサに局在化されるのではなく、全プロセッサに均質に分散する。この前提において交換機制御用マルチプロセッサにおけるリソース管理の対案はリソース管理情報を集中するか、分散するかの選択になる。

通話路リソースの管理にはリソース管理テーブルの配置が必須となる。通話路のリソースは膨大であり、リソースの分配管理には各リソースごとに空塞を表示することが必要である。この管理テーブルの配置と処理法が交換機制御用マルチプロセッサにおけるプロセッサ間結合の基本機構となる。

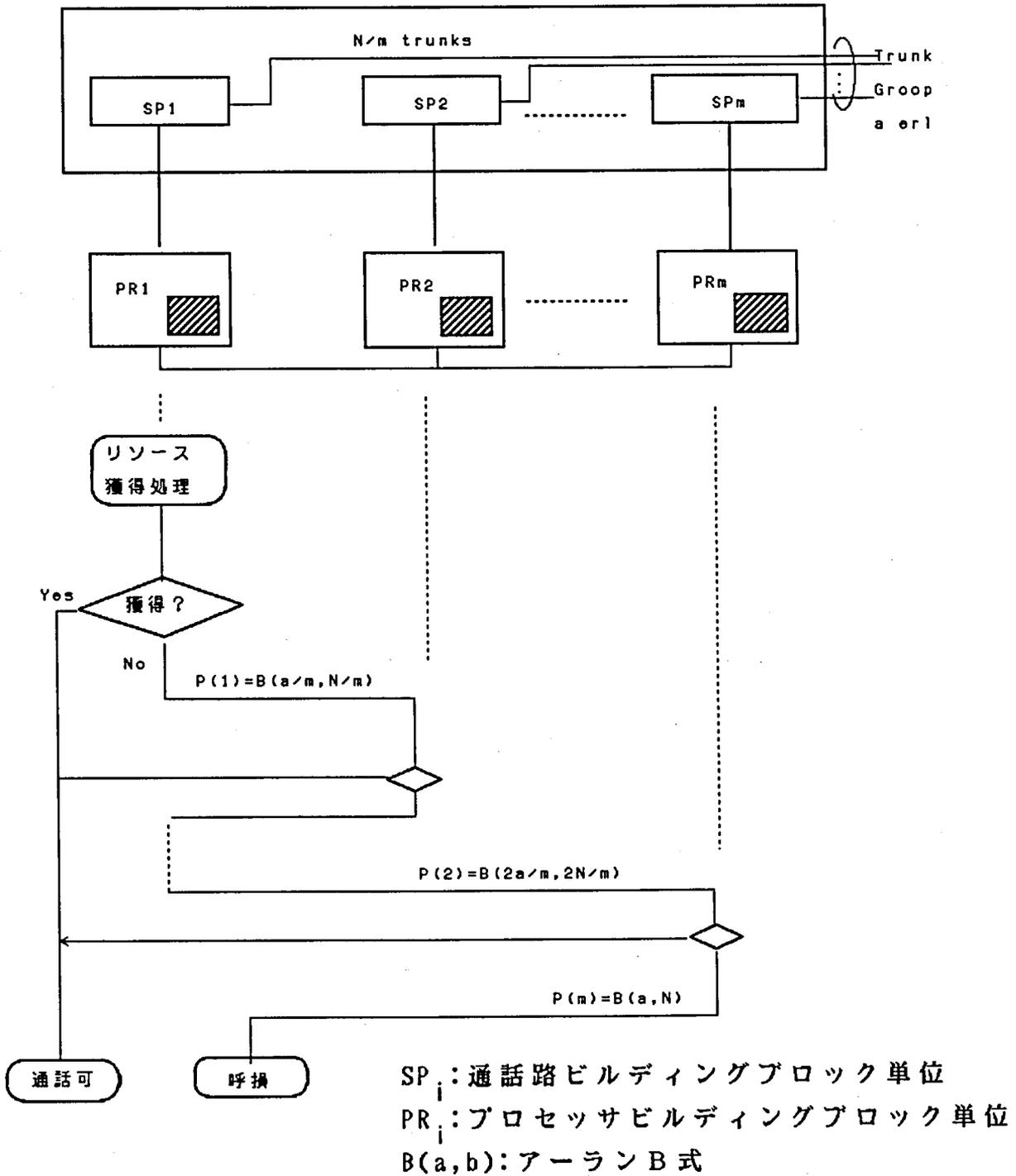
上記のマルチプロセッサ基本構成において、この管理テーブルを集中配置をするか／分散配置をするかの選択により、以下に示す2種の対案が考えられる。

#### (a)個別配置方式

この方式は通話路ビルディングブロック単位と対応付けられたプロセッサにおいて、各通話路ビルディングブロック単位に属するプロセッサにおいて各プロセッサが対応するリソースを個別に管理するものである。ただし、個別に管理してもリソースは交換機全体で共有するものであり、各プロセッサで分配されたリソースが不足した場合、他のプロセッサに取りにいくことが必要である。図3.2(a)に示すように、各プロセッサがリソース管理テーブルを持ち、リソースをそれぞれ管理する。他のモジュールは自プロセッサ内のリソースが不足した場合、必要に応じて他のプロセッサにアクセスし、リソースの分配を受ける。

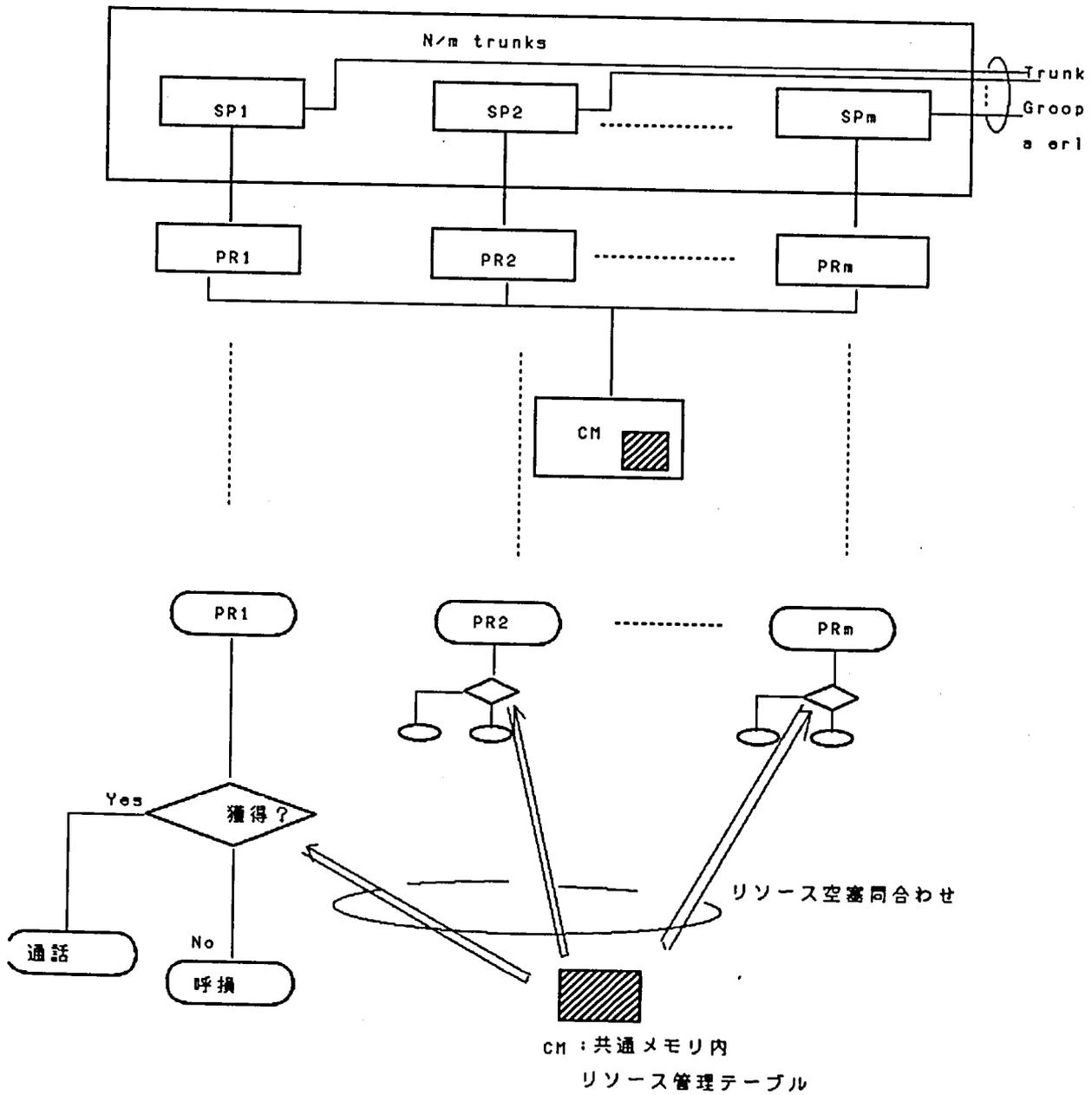
#### (b)集中配置方式

交換機全体のリソース管理情報を1個所に集中し、一元的に管理する考え方である。図3.2(b)に示すように、リソース管理テーブルは全てのプロセッサがア



(a) リソース管理情報の分散配置

図3.2 リソース管理方式の比較(その1)



(b) リソース管理情報の集中配置

図3.2 リソース管理方式の比較 (その2)

アクセス可能なメモリに置く必要があり、このために共通メモリをおく。リソースが必要になると、各プロセッサはこのテーブルを参照して空きのリソースを求め、リソースの分配を受ける。

### 3.2 ダイナミックステップ数によるリソース管理方式の評価

上記の両方式の差は、呼当りのダイナミックステップ数に現れ、これを両方式について比較する。

#### 3.2.1 リソース管理方式とダイナミックステップ数

##### (1) 個別管理方式の呼当りのダイナミックステップ数

個別管理方式において*i*台目のプロセッサに問い合わせても出回線を確保できない確率*P*(*i*)は「*i*台のプロセッサが管理している方路当りの出回線の総量に対し、*i*台のプロセッサが管理している通話路からこの方路に呼量加わった時の呼損率」で近似が出来る。ここで、*B*(*e*,*N*)を入り呼量 *e*アーラン(*erl*)、出回線数 *N*で、ランダム発呼を仮定したときの呼損率を示すアーラン*B*式とすると、*P*(*i*)は以下のように表わされる。

$$P(i) = B(i \cdot a/m, i \cdot N/m) \quad (3.1)$$

なお、*a*は交換機全体から特定の1方路に向かう呼量、*N*は呼量 *a*(*erl*)で呼損率0.01以下を満足する出回線の最小数、*m*はプロセッサ台数である。

各プロセッサでリソースの確保に必要な処理ステップ数(プロセッサ間の通信処理を含む)を*r*とすれば、自プロセッサで管理するリソースでは不足し、他のプロセッサへリソースの分配を受けに行く平均のオーバーヘッドステップ数*O<sub>v</sub>*は次式となる。

$$O_v = r \cdot \sum_{i=1}^{m-1} i \cdot P(i) \quad (\text{ただし, } m \geq 2) \quad (3.2)$$

呼当りのダイナミックステップ数を*U*、新サービスの導入などによるダイナミッ

クステップの増加計数をKとすれば、個別管理方式における呼当りのダイナミックステップ数 $S_a$ は次式となる。

$$S_a = K \cdot U + O_v \quad (3.3)$$

## (2) 共通管理方式の呼当りのダイナミックステップ数

共通管理方式では、各プロセッサがリソースを必要としたとき、共通リソースを管理するデータに直接アクセスできる。ただし、複数プロセッサから共通メモリに同時にアクセスする場合、複数プロセッサが互いに論理矛盾を起こさないように競合アクセスに対する保護を行わなければならない。しかし、この処理は呼当りの総ステップ数に比べ無視できる（経験データによればこの比率は1%以下）。従って、共通管理方式の呼当りのダイナミックステップ数は近似的に次式となる。

$$S_b = K \cdot U \quad (3.4)$$

### 3.2.2 両方式のダイナミックステップ数の比較

共通管理方式と個別管理方式のダイナミックステップ数の比 $R (=S_a/S_b)$ で両方式を比較する。(3.4), (3.5)より、 $R$ は以下となる。

$$R = 1 + r/(K \cdot U) \cdot \sum_{i=1}^{m-1} i \cdot B(i \cdot a/m, i \cdot N/m) \quad (3.5)$$

ただし、 $m \geq 2$ ,

また、個別管理方式のプロセッサ台数 $m$ は交換機全体の総処理ステップ数とプロセッサの平均命令実行時間の関係から以下となる。

$$m = \lceil K \cdot U \cdot R \cdot y \cdot q \rceil = \lceil K \cdot R \cdot q' \rceil \quad (3.6)$$

ただし、 $q' = U \cdot y \cdot q$

ここで、 $y$ は交換機規模（単位時間当りの生起呼数で表わす）、 $q$ をプロセッサの平均命令実行時間、 $\lceil x \rceil$ を $x$ 以上で最小の整数とする。なお、 $q'$ は正規化

したプロセッサの平均命令実行時間を意味し、複数プロセッサで交換機の全処理量（ただし、サービス拡大を考慮していない現在値  $K=1$ ）を実行するに要する処理時間の和である。

前述のように  $B(e,N)=0.01$  であり、 $e$  を与えることにより  $N$  が求まる。さらに、各プロセッサでリソース確保に必要な処理の処理量の相対値、 $r/U$  を与えることにより、 $R$  は (3.5), (3.6) から一意的に求まる。

図3.3に最大局規模を想定して求めた  $R$  と  $q'$  の関係を示す。 $q'$  の増大（プロセッサの能力低下）とともに  $R$ （個別管理方式と共通管理方式のダイナミックステップ数の比）が増大する傾向が見られる。さらに、 $K$ （サービス追加等により生じる呼処理ステップの増大係数）の増大により、 $R$  の増大化傾向が強まる。

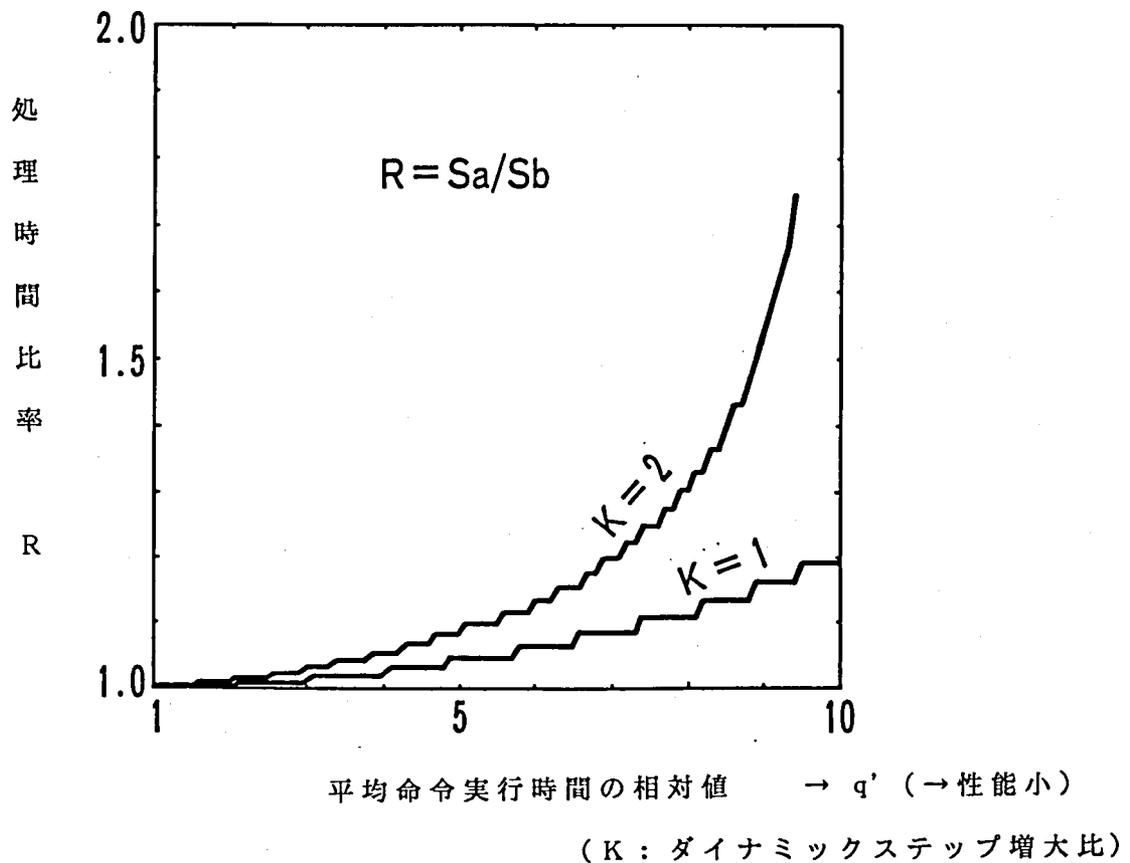


図3.3 リソース管理法と処理時間

注) 交換機に関する経験データより、 $a=50$ ,  $r/U=0.1$ を仮定

### 3.3 考察

プロセッサ能力の低下による個別管理方式の処理量の増大化傾向は以下のように説明できる。プロセッサの能力が低下すると、各プロセッサが管理する目的の対地向けの回線が少なくなると、回線使用中の確率が増大する。このため、他のプロセッサが管理する回線を確保する必要があり、他のプロセッサに迂回することが多くなり、処理量が増大する。一方、共通管理方式ではシステム全体で共通リソースを一元的に管理しており、プロセッサの能力によらず処理量は一定である。

また、 $K$ （呼当り処理量の増大係数）の増大による  $R$  の増大化傾向は以下のように説明できる。 $K$  の増大は、総処理量の増大を意味し、等価的にはプロセッサの能力低下を意味する。従って、上記のプロセッサ能力の低下によるダイナミックステップ数の増大の議論がそのまま当てはまる。

本マルチプロセッサでは小規模における制御系の経済化が一つの狙いであり、比較的能力の小さな小規模プロセッサを利用すること、しかも、今後通信網サービスの拡大から呼当りの処理量が増大することが考えられ、比較的能力の低いプロセッサを利用して交換用マルチプロセッサを実現する必要があり、共通メモリを用いたリソース管理法を採用することが望ましい。

## 4. 機能分散による呼制御処理の最適化

ここでは既に結論付けたリソース共通管理方式を前提にして、呼対応の処理を機能分散する必要性について議論する。

### 4.1 交換処理の機能分割

交換処理の基本機能は加入者線、中継線から信号を受信し、この信号を識別して目的の回線、加入者線を選択し、接続し、通話を成立させることにある。既存の交換処理プログラムの特性は大きくは以下の2種に機能分類できる。

〔信号処理機能〕 通話路ハードウェアの状態を読み取り、回線の監視信号、接続信号の中から有意な情報を抽出し、プログラムが処理可能な論理情報に変化する

る。 信号処理は接続された通話路モジュールと直接つながりを持つ。

[呼制御処理機能] 生起した呼の要求に従い、出中継回線などの共通リソースを選択し、接続処理を行なう。 呼制御処理では、交換機全体で一元管理されたリソースに関する処理が主であり、また信号処理により論理化されたメモリ内の情報を処理するものである。 呼制御処理は通話路とプロセッサの対応は無くてもよい。

#### 4.2 機能分散方式の比較

呼処理プログラムの基本性質に従い、ここでは機能分散の有効性について検討する。 機能分割の対案として、①呼対応の処理を機能分散せずに同一プロセッサで処理するものと、②交換処理機能を信号処理プロセッサ(SGP)と呼制御プロセッサ(CNP)にそれぞれ機能分散したものを比較する。 機能分散する場合、SGPは各通話路モジュール対応に設けるが、CNPは各通話路モジュールと独立に設け、処理量に応じて増設できるものとする。 すなわち、SGPからの処理依頼に対し、任意のCNPが処理可能とする。 なお、本稿では通話路モジュールに対応して単純にプロセッサを付加する前者の方式を単純負荷分散方式、後者を機能分散・負荷分散併用方式と呼ぶ(図3.4)。 この機能・負荷分散併用方式はハードウェア対応の階層を信号処理、論理的なサービス対応の処理を呼制御処理に分離し、ソフトウェアのモジュール化の観点からは、デジタル統合網の構築に向けた機能追加の容易化に有効な機能分割でもある。

##### 4.2.1 評価パラメータ

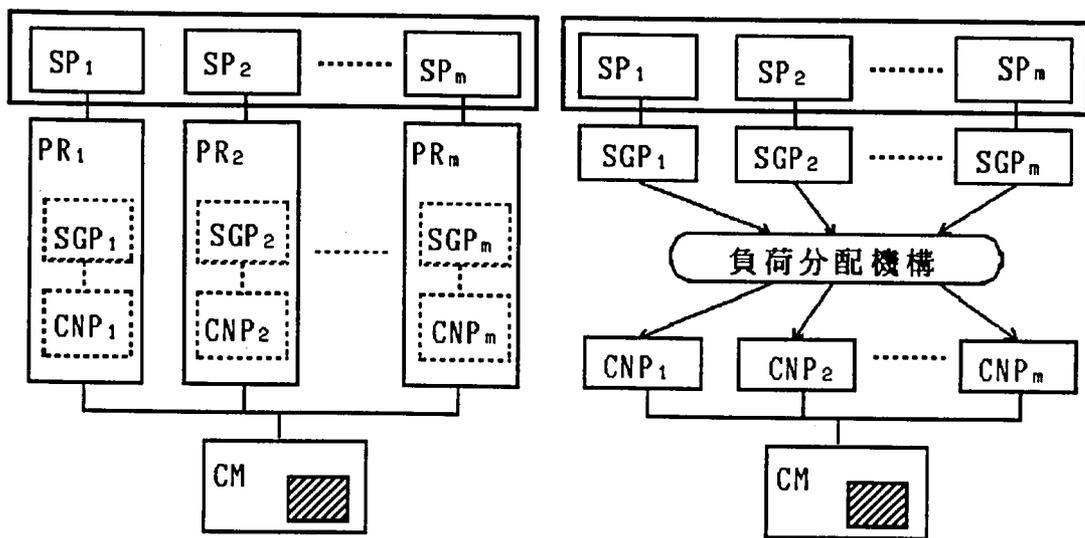
両方式の差を定量比較するパラメータとして、全節で用いたダイナミックステップ数が想定される。 しかし、両対案でのダイナミックステップ数の差は呼処理の機能分割によるものではなく、SGPとCNP間の通信に伴うオーバーヘッドと見なせる。 このプロセッサ間通信のオーバーヘッドはハードウェアとソフトウェアを具体化する段階で極力少なくする工夫をするため、ここでは無視するとダイナミックステップ数は問題とならない。

両方式の差は呼制御処理の処理応答時間の差となって現われる。 信号処理から呼制御処理へ処理依頼する場合、単純負荷分散方式では呼制御処理が単独サー

バ、機能・負荷分散併用方式では呼制御処理が複数サーバの待ち時間モデルとなる。この待ち合わせを含む処理時間は発呼に対する応答時間として現れ、実時間システムでは重要な尺度である。

両方式の差を呼制御処理の処理時間で比較する。なお、呼制御処理は小規模なタスクに分割されており、信号処理側は通話路ハードウェアの状態変化を検出する毎に小規模タスクを呼制御処理担当部へ処理依頼するものである。また、呼制御処理の各小規模タスクの実行時間は2次のアーラン分布に近いことを経験的に得ており、この分布を仮定する。

信号処理と呼制御処理が同一のプロセッサで実行される単純負荷分散方式の処理待ち合わせは出線1の2次アーラン分布、すなわち  $M/E_2/1$ モデル、複数のプロセッサ(CNP:  $s$ 台)が負荷分散を行なう機能・負荷分散併用方式では出線  $s$ の2次アーラン分布、すなわち  $M/E_2/s$ モデルで評価しうる。ここで、機能・負荷分散併用方式では前述の通り、SGP~CNPの通信機構にCNPへの負荷分散を均等化する負荷分配機構が存在することを仮定している。



(a) 単純負荷分散方式

(b) 機能・負荷分散併用方式

図3.4 機能分散の導入

#### 4.2.2 分散方式と平均処理時間

##### (1) 単純負荷分散方式の平均処理時間

前述のパラメータに加えて、タスク当りの平均ダイナミックステップ数を  $e$  (従って、 $U$ は複数タスクの合計ステップ数)、通話路モジュール数は  $m$ 個、また、 $M/E_2/1$ モデルにおけるプロセッサの使用率  $\rho$  の時の平均待ち時間比を  $w(1, \rho)$  とすると、呼処理の各タスクの平均処理時間  $H$  は次式で与えられる。

$$H = K \cdot e \cdot q \cdot (1 + w(1, \rho)) \quad (3.7)$$

ただし、

$$\rho = K \cdot U \cdot y \cdot q / m = K \cdot q' / m \quad (3.8)$$

したがって、

$$H = K \cdot e \cdot q' / (U \cdot y) \cdot (1 + w(1, K \cdot q' / m)) \quad (3.7')$$

##### (2) 機能・負荷分散併用方式の平均処理時間

機能分散・負荷分散併用方式では再繁時の平均使用率が  $\rho_{max}$  を越えるごとにプロセッサを増設する。いま、CNPの台数を  $s$ 、 $w(s, \rho)$  を出線  $s$  の2次のアーラン分布の平均待ち合わせ時間比とし、全てのCNP間で負荷の平均化が図られるとすれば、タスクごとの平均処理時間  $H$  は次式で与えられる。

$$H = K \cdot e \cdot q \cdot (1 + w(s, \rho_{max})) \quad (3.9)$$

$$s = \lceil K \cdot U \cdot y \cdot q / \rho_{max} \rceil = \lceil K \cdot q' / \rho_{max} \rceil \quad (3.10)$$

したがって、

$$H = e / y / U \cdot q' \cdot K \cdot (1 + w(\lceil K \cdot q' / \rho_{max} \rceil, \rho_{max})) \quad (3.9')$$

### (3) 平均処理時間の比較

図3.5に通話路モジュール数  $m$  とCNP台数  $s$  が等しいとし、 $m=s=10$ と大規模局の交換機を想定したプロセッサの性能と平均処理時間の関係を示した。なお、性能として正規化した平均命令実行時間  $q'$  を、平均処理時間も式(3.7)、(3.9)に共通な  $e/y/U$  で正規化した処理時間比を用いている。また、機能分散・負荷分散併用方式では安全側をとってすべてのCNPの使用率  $\rho_{max}$  を0.9に固定している。

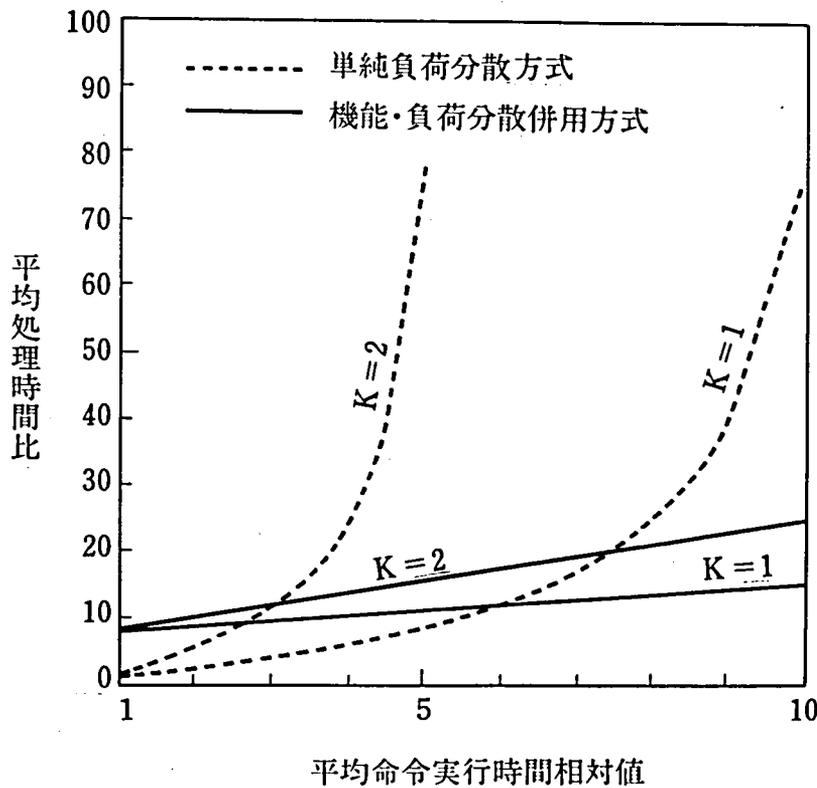


図3.5 機能分散の有無と平均処理時間

注)  $K$  : ダイナミックステップ増加比

負荷分散モジュール  $m=10$ と大規模局を想定。

### 4.3 考察

図3.5に示した処理遅延時間は発呼に対するダイヤルトーン応答遅延時間の増大を招くものである。実時間サービスを行なう交換機ではこの処理遅延時間をより少なくすることが望ましい。

単純負荷分散方式では、処理遅延時間が処理装置の能力低下とともに大きく増大する。将来の処理量増大まで考慮すると、処理遅延時間を少なくするためには高コストで高処理能力のプロセッサを導入初期から必要とする。一方、機能分散・負荷分散併用方式では、この処理遅延の増加は少ないことが示されている。

機能・負荷分散併用方式は比較的低能力のプロセッサを用いることが可能であり、今後のデジタル統合網の進展に対して初期投資の経済化、将来の拡大を持ち、また、前述のソフトウェアのモジュール化を推進する有効な方式である。

## 5. 機能分散による信号処理の最適化

今までの議論から、交換機の共通リソースを共通メモリで管理すべきこと、さらに交換処理を信号処理と呼制御処理に分割し、機能・負荷分散を図る方式が望ましいことを得た。この分割案の中で呼制御処理はリソース管理を行なう均質な処理であるため、これをさらに機能分割することを考える必要はない。一方、信号処理には信号方式ごとに異なった処理があり、信号方式毎の機能分割が本マルチプロセッサの構成を考える上での残された課題である。

本節では、従って、サービス総合化等の機能追加が信号処理方式対応の機能分割に与える影響を考察する。具体的には、マルチプロセッサの単位プロセッサでの処理量を求め、信号処理の機能分割がどのような影響を与えるかを議論する。ただし、信号方式追加、サービス追加によるソフトウェア、ハードウェアの増加の影響も、それらの機能追加による新しいサービスを行なわない初期では不要であり、初期設備を極力少なくすることも大きな課題である。なお、ここで信号方式とは加入者ダイヤルパルス信号、加入者多周波、中継線信号などの各種信号方式、呼種とは自局内接続呼、市外発信呼、市外着信呼などを表わす。将来、高機能な加入者信号が追加されることは必至であり、また、サービス追加として

通信処理サービスやパケット等の呼種の増加も必至である。

## 5.1 交換処理の処理量の定式化

### 5.1.1 交換処理機能の分類

交換用プロセッサの負荷に対する呼のかかわり合いから、交換処理は以下の3種に分類できる。

〔固定処理〕 呼数に関係なくプロセッサに対応して固定的に必要とする処理である。 実時間制御用のプログラムスケジューラ等がこれに当たる。

〔局規模比例処理〕 呼の生起に影響を受けず、局の規模によって処理量が規定できる。 処理の機能面からは信号処理が該当する。 その代表例を図3.6に示す。 負荷の有無によって生じる処理量の変化はタスク登録処理の部分のみの微少な処

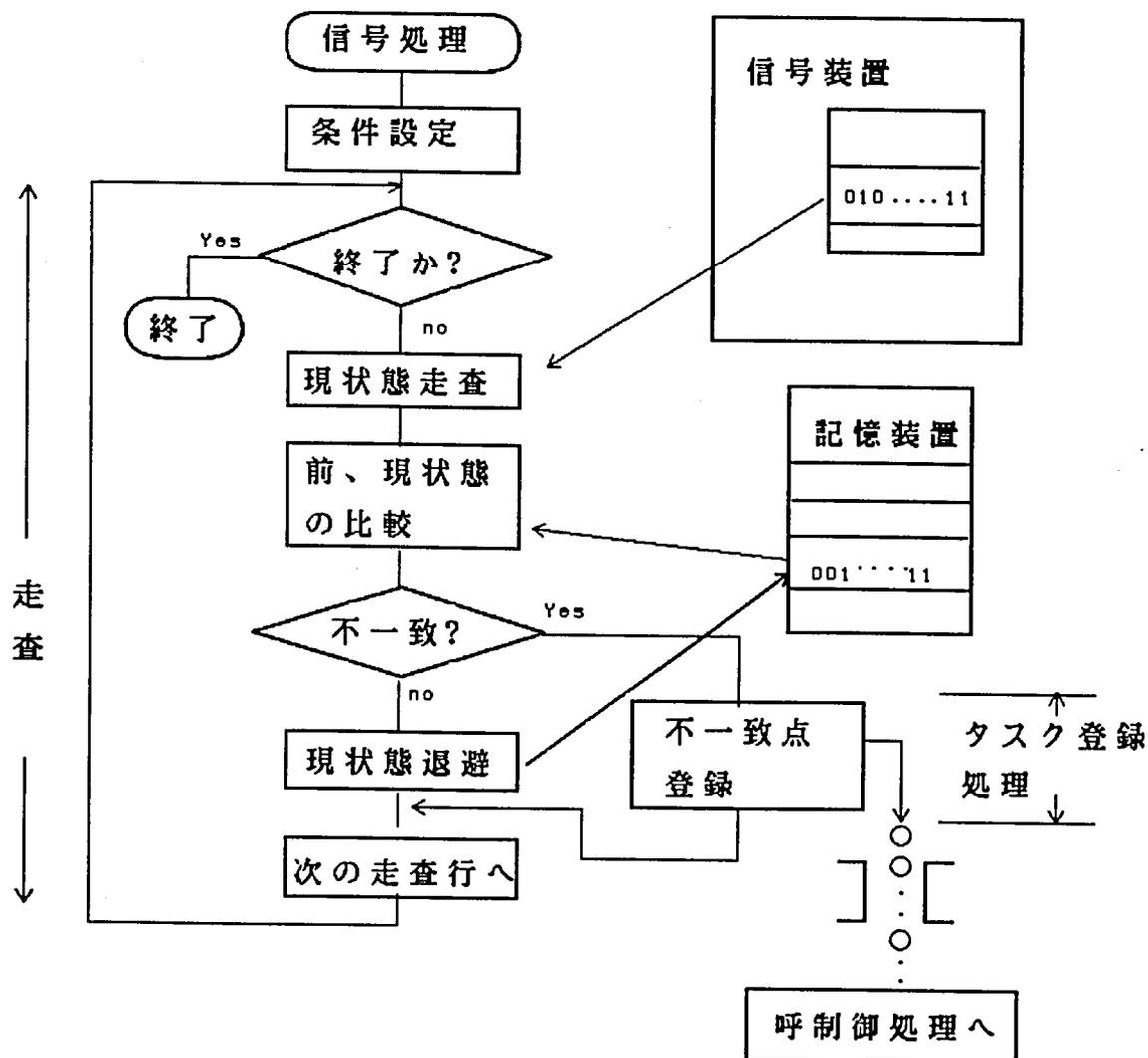


図3.6 信号処理の例

理であり、負荷に関係なく生じる処理であることがわかる。

[呼数比例処理] 呼対応に必要とする処理であり、呼制御処理がこれに該当する。

### 5.1.2 処理量の定式化

まず、1台のプロセッサで処理しうる処理量を以下のようなモデルを用いて算出する。ここで既出のパラメータと同様、 $\rho$ をプロセッサの使用率、 $q$ を平均命令実行時間とし、呼当りの呼数比例処理ステップ数を  $a_0$ 、呼当りに変換した局規模比例処理ステップ数を  $b_0$ 、単位時間当りの固定処理ステップ数を  $c_0$ とし、局規模を  $y$ （単位時間当りの呼数）とする。なお、前述のように局規模比例処理（信号処理）については呼数に直接は比例しない処理であるが、通常は局規模と最繁時の呼数とは比例する関係にあることから、定式化にあたっては局規模比例処理を最繁時の呼数に比例した量として表現している。

これらのパラメータには以下の関係がある。

$$\rho = q \cdot \{ y ( a_0 + b_0 ) + c_0 \} \quad (3.11)$$

呼数に比例する処理量を明確化するため、式(3.11)を以下のように変形する。

$$\{ \rho / (q \cdot c_0) \} - 1 = y \cdot ( a_0 / c_0 + b_0 / c_0 ) \quad (3.12)$$

左辺は実質的にプロセッサの性能を表わす処置をしたものである。すなわち、プロセッサの使用率から呼数比例に無関係なオーバヘッド的な固定処理を除いて、固定処理の処理時間( $q \cdot c_0$ )で正規化している。なお、 $c_0$ は交換機の呼種、信号方式に関わらないオペレーティングシステム等のシステムオーバヘッドであり、ここでは永久固定としている。式(3.12)の右辺は、呼種や信号方式で変化する  $a_0$  や  $b_0$  と局の最大規模  $y$  とを反比例する関係のみで表わせることを示している。したがって、

$$y = \{ \{ \rho / (q \cdot c_0) \} - 1 \} / \{ (a_0 / c_0) + (b_0 / c_0) \} \quad (3.12')$$

ここで、信号方式  $s_i$ 、呼種  $t_i$  に属する呼数比例処理の呼当りのステップ数を

$a_{[s_i, t_i]}$ , 単位時間当りの呼の生起数を  $y_{[s_i, t_i]}$ , 最繁時の呼数比例量に換算した信号方式  $s_i$  に対する局規模比例処理のステップ数を  $b_{[s_i]}$  とする. さらに現在信号方式の集合は  $S_0$ , 呼種の集合は  $T_0$  とする.

式(3.12)の右辺を  $v$  とおき, 呼種, 信号方式に対応した処理に分解すると以下のようになる.

$$v = \sum_{s_i \in S_0} \left[ \sum_{t_i \in T_0} \{a_{[s_i, t_i]} \cdot y_{[s_i, t_i]} / c_0\} + b_{[s_i]} \cdot \sum_{t_i \in T_0} \{y_{[s_i, t_i]} / c_0\} \right] \quad (3.13)$$

ここで本交換機を導入するとした場合の将来への条件を以下のように仮定する.

- (1) 信号方式の集合は現在  $S_0$  であるが, 将来  $S_f$  となる.
- (2) 呼種の集合は現在  $T_0$  であるが, 将来  $T_f$  となる.
- (3) 呼数比例処理ステップ数は将来  $A_{[s_i, t_i]}$  倍に増大する.
- (4) 信号方式  $s_i$ , 呼種  $t_i$  の呼数は, 将来  $N_{[s_i, t_i]}$  倍になる.

このマルチプロセッサを導入する局では, 局の規模は変わらず (規模が変わればハードウェア全ての増設が可能で処理量増大のみへの影響の評価は意味がない) 呼数と処理量が増大したときの対処の可能性を検討する.

将来の処理量  $v_f$  は(3.13)から以下のようになる. なお信号処理量は局規模比例処理であり, 信号装置が扱う回線数で固定され, 呼数増大の影響を受けない.

$$v_f = \sum_{s_i \in S_f} \left[ \sum_{t_i \in T_f} \{A_{[s_i, t_i]} \cdot a_{[s_i, t_i]} \cdot N_{[s_i, t_i]} \cdot y_{[s_i, t_i]} / c_0\} + b_{[s_i]} \cdot \sum_{t_i \in T_f} \{y_{[s_i, t_i]} / c_0\} \right] \quad (3.14)$$

今後出現する信号方式と呼種に対応した呼数比例処理や局規模比例処理は, 現在は存在しないが, 式の上では既存のものと同じく将来値に対し,  $1/A_{[s_i, t_i]}$ ,  $1/N_{[s_i, t_i]}$  で表わしている. また, 信号処理と呼制御処理が機能分割された場合は信号処理の中にプロセッサ間の通信や負荷分配などのオーバヘッドを含むが, この機能を実現する効率の良いハードウェアが具備されているとして省略するのは全節の議論と同様である.

## 5.2 信号方式対応の機能分散の効果

節 4. で結論付けたマルチプロセッサについて信号方式対応の機能分散の有無により分類すると図3.7の2種のマルチプロセッサが考えられる。両者の違いは信号処理にあり、図3.7(a)は信号方式の種類に関係なく、負荷量に応じてSGPを負荷分散している。図3.7(b)は信号方式対応にSGPを設け、負荷に応じて同一信号方式の中で負荷分散している。

ここで将来の処理量の増大に対応できるが、これらのサービス追加や信号方式の追加が生じない段階での初期設備プロセッサ台数  $n$  を算出する。

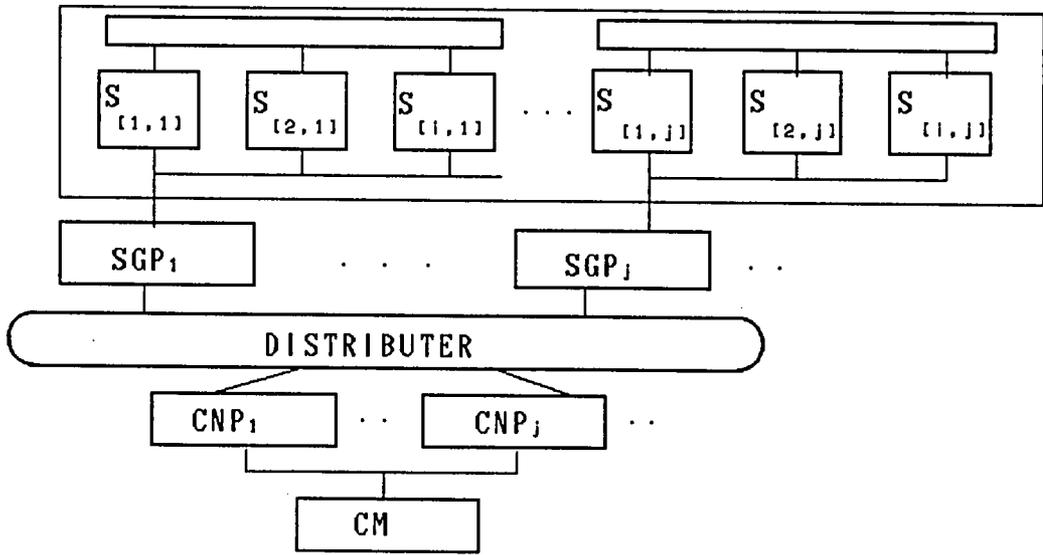
### (1) 単純型機能・負荷分散併用方式

CNPは処理量に応じて追加が可能のため、呼種の増大に伴い処理量が増大した時点でプロセッサを追加すればよいが、SGPの能力は将来の新信号方式の導入による処理量増大を考慮しておく必要がある。したがって、それぞれの交換機規模に必要なプロセッサの台数は交換機全体の呼数比例処理の処理量（ただし、プロセッサの使用比率で換算したもの）と1台のプロセッサが呼数比例処理に使用し得るプロセッサ使用率の比として表わされ、(3.12)、(3.14)から必要なプロセッサの台数  $n$  は以下で表わされる。ただし、 $\rho_{max}$  は固定処理を含んで最大限利用可能なプロセッサの使用率である。

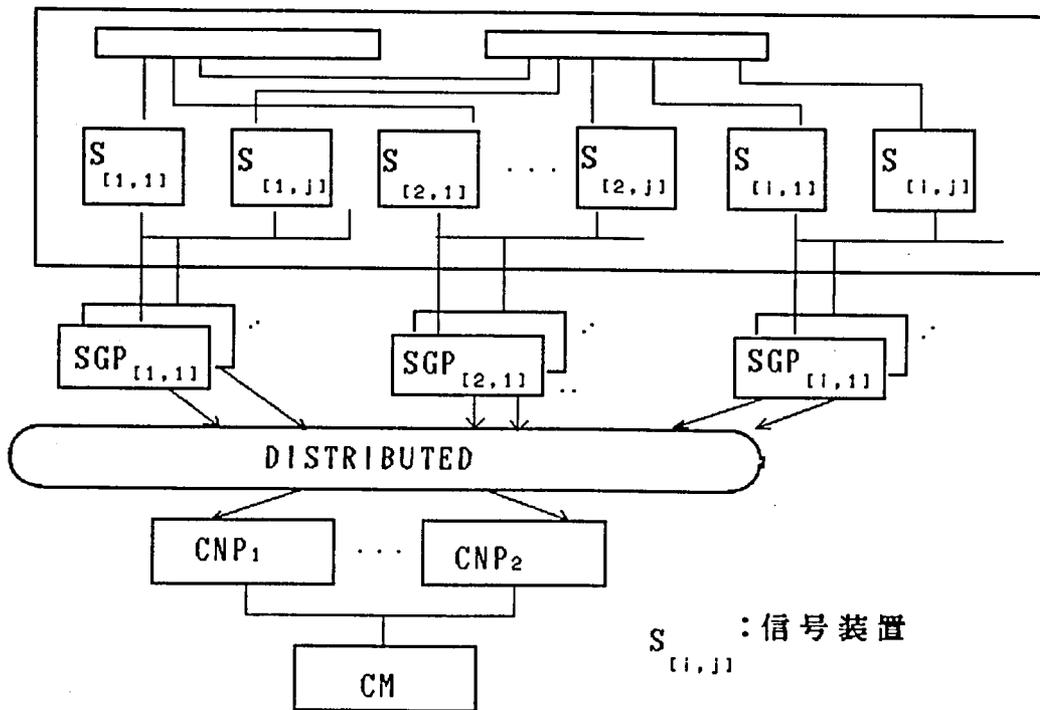
$$n = \left[ \frac{\sum_{s_i \in S_0} \sum_{t_i \in T_0} a_{[s_i, t_i]} \cdot y_{[s_i, t_i]} / c_0}{(\rho_{max} / (c_0 \cdot q) - 1)} \right] + \left[ \frac{\sum_{s_i \in S_f} b_{[s_i]} / c_0 \sum_{t_i \in T_f} y_{[s_i, t_i]}}{(\rho_{max} / (c_0 \cdot q) - 1)} \right] \quad (3.15)$$

### (2) 複合形機能・負荷分散併用方式

本マルチプロセッサ形態では、現在のサービスに使用される信号方式対応にSGPは必要であるが、将来の信号方式の追加、呼数の増加は必要時に追加すればよい。したがって、初期設備数では将来の増加は考慮する必要がない。複合型機能・負荷分散併用方式マルチプロセッサでは初期設備プロセッサの台数  $n$  は(3.15)と同様に求めると以下のようなになる。



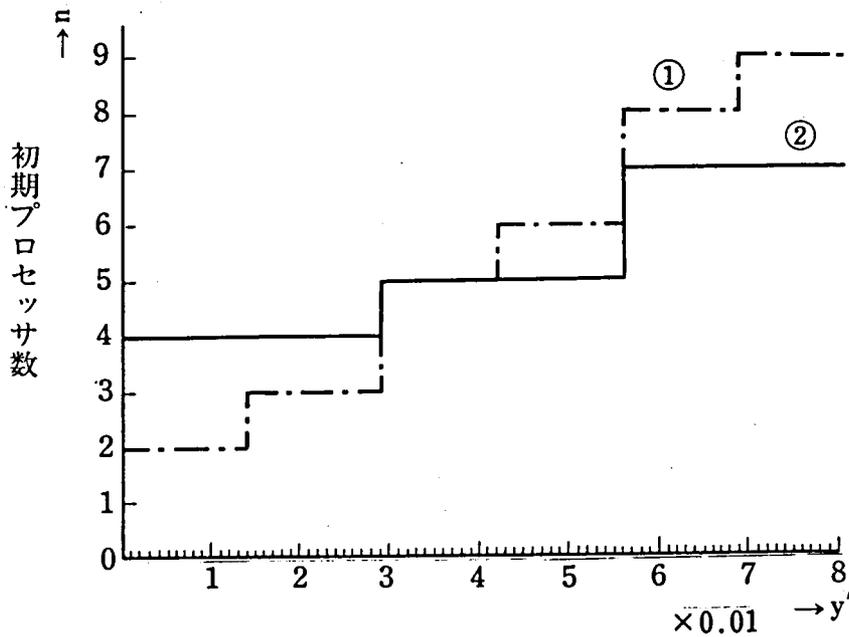
(a) 機能分散・負荷分散併用方式



$S_{[i,j]}$  : 信号装置

(b) 複合形機能分散・負荷分散併用方式

図3.7 信号方式による機能分割



平均命令実行時間相対値 (→性能低下方向)

① 単純機能分散・負荷分散併用方式

② 複合形機能分散・負荷分散併用方式

図3.8 信号処理の機能分割と初期プロセッサ数

注) 経験データ等により以下を仮定している。

$$\text{大規模局を想定し, } \sum_{s_i \in S_0} \sum_{t_i \in T_0} y_{[s_i, t_i]} = 100 \text{ 呼/秒}$$

$S_0 = \{S_1, S_2, S_3\} = \{ \text{加入者線個別信号, MF信号, 共通線信号} \}$

$T_0 = \{T_1, T_2, T_3\} = \{ \text{自局内呼, 発信呼, 着信呼} \}$

$a_{[s_i, t_i]}$  と  $b_{[s_i]}$  は固定処理量  $c_0$  で正規化し, すべて0.3.

$$y_{[s_1, t_1]} = 100/2,$$

$$y_{[s_2, t_2]} = y_{[s_2, t_3]} = y_{[s_3, t_2]} = y_{[s_3, t_3]} = 100/8,$$

$$y_{[s_1, t_2]} = y_{[s_1, t_3]} = y_{[s_2, t_1]} = y_{[s_3, t_1]} = 0.$$

$S_0$  に将来3種の信号方式を導入し, 各呼の生起頻度は2倍になることを想定している。

$$n = \left[ \frac{\sum_{s_i \in S_0} \sum_{t_i \in T_0} a_{(s_i, t_i)} \cdot y_{(s_i, t_i)} / c_0}{(\rho_{max} / (c_0 \cdot q) - 1)} \right] + \sum_{s_i \in S_0} \left[ \frac{b_{(s_i)} / c_0 \sum_{t_i \in T_0} y_{(s_i, t_i)}}{(\rho_{max} / (c_0 \cdot q) - 1)} \right] \quad (3.16)$$

### 5.3 考察

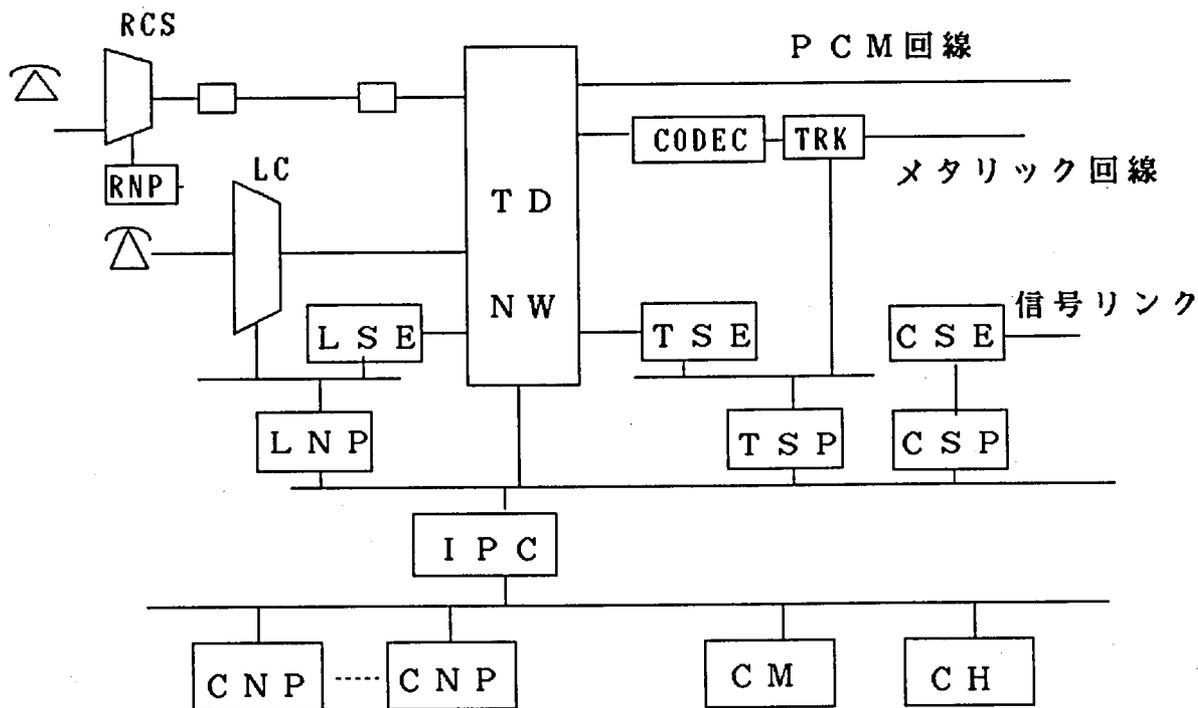
プロセッサの相対平均命令実行時間  $c_0 \cdot q$  をパラメータとし、プロセッサ初期設備数を算出した例を図3.8に示す。プロセッサの性能が小さい領域（右方向）では信号方式対応に信号処理の機能分散を図った複合形機能分散・負荷分散併用方式のプロセッサ初期設備数が少ない。プロセッサの性能が向上すると単純機能分散・負荷分散併用方式の初期設備数が少なくなる。従って、マルチプロセッサの単位プロセッサとして、高処理能力のプロセッサを用いる場合には単純機能分散・負荷分散方式の初期設備数が少ない。比較的能力の低いプロセッサを用いる場合には、信号方式対応に信号処理の機能分散を図った複合形機能分散・負荷分散併用方式がプロセッサの初期設備数が少ない。

ここで用いるマルチプロセッサ用の単位プロセッサの性能はMOS-VLSI技術を想定して、図3.8において相対平均命令実行時間が0.05~0.20程度のプロセッサと推定されるので信号方式対応に信号処理の機能分散を行なった複合形機能分散・負荷分散併用方式が初期設備数が少なく、経済性の点で優れている。

## 6. 交換機制御用マルチプロセッサ基本構成の確定

以上の考察に基づいて結論づけたデジタル交換機のマルチプロセッサ構成は、実験機での確認を経て商用化され、D70デジタル交換機として日本の主要電話局に大量導入されつつある。D70交換機の基本構成を図3.9に示す。

CNP群は共通メモリと結合され、LNP, TSP, CSPからなる信号処理階層とCNP群の呼制御処理階層に分離され、また信号処理階層はLNP, TSP, CSPへ信号種別によって機能分散されている。



- |                 |                      |
|-----------------|----------------------|
| TDNW : 時分割通話路装置 | RNP : 遠隔設置通話路制御プロセッサ |
| LC : 集線装置       | LNP : 加入者線信号処理プロセッサ  |
| LSE : 加入者線信号装置  | TSP : 中継線信号処理プロセッサ   |
| TSE : 中継線信号装置   | CSP : 共通線信号処理プロセッサ   |
| CSE : 共通線信号装置   | CNP : 呼制御プロセッサ       |
| TRK : トランク装置    | IPC : プロセッサ間通信制御装置   |
| RCS : 遠隔制御交換装置  | CM : 共通メモリ           |

図3.9 D70デジタル交換機の基本構成

## 7. 結言

マルチプロセッサの基本構成として以下の結論を得た。

- (1) リソース管理の効率化の考察から、共通メモリで交換機のリソースを集中管理する共通メモリ集中管理が望ましい。
- (2) 呼接続の実時間性の考察から信号処理と呼制御処理を階層的に機能分散し、各階層内では負荷分散する機能分散／負荷分散併用方式が望ましい。
- (3) 将来の信号方式の追加に追従でき、かつ初期設備数を少なくしうるマルチプロセッサ構成を考察し、信号処理階層ではさらに信号方式別に機能分散を図ることが望ましい。

以上により、マルチプロセッサの基本構成を得たので、これを具現する設計について次章以降に議論する。

## 第IV章 デジタル交換機用マルチプロセッサの基本設計

### 1. 序論

前章では 将来のデジタル統合網構築に向けた要求条件に適合し、かつデジタル統合サービスが未実施の段階においても通信網の経済化に貢献する観点およびプロセッサの動向からデジタル交換機制御用に望ましいマルチプロセッサの基本構成を導いた。本章ではこの基本構成を具体化し、評価する。

このマルチプロセッサの実現・評価に向けては以下の段階を踏む。

- ①マルチプロセッサ構成を実システムとして具現する段階での設計理念の構築、
- ②設計理念の下でのマルチプロセッサシステムの構成要素への分解と各構成要素の実現へ向けた設計指針の構築、
- ③設計指針に基づいた具体的システムの設計、
- ④実現結果の動作確認および評価

なお、前章と同様、基本設計は機能と性能に着目した設計に関するものであり、このマルチプロセッサの高信頼化は第V章以降で議論する。

設計理念についてはシングル／マルチプロセッサ両構成の調和を考慮する。マルチプロセッサには多数のプロセッサの結合や処理の並列性による複雑性が本質的に存在するが、これをシングルプロセッサ制御方式並みに簡単化する。また小規模で必要なシングルプロセッサ制御方式との機種統一を図る。

本マルチプロセッサの実現・実証に向けた設計事例として2例を挙げる。その一つはシングルプロセッサ制御方式用の既存プロセッサを用いたマルチプロセッサである。既存部ではマルチプロセッサ化への変更は極力避け、マルチプロセッサ用に必要な付加機構を確立した実験機である。付加機構としてのプロセッサ間通信機構に関する条件を分析し、マルチプロセッサシステム全体をシングルプロセッサに仮想化しうよう負荷分配機構を具備した構成案を示し、その具体化を図っている。もう一方の設計事例はマルチプロセッサへの適用を始めから前提としたカスタムVLSIプロセッサの実現とマルチプロセッサへの適用である。ここでは従来のプロセッサとは世代を画する32ビットカスタムプロセッサを実現し、性能・効率性・経済性・高信頼性で前のマルチプロセッサ実験機の改良を図り、本格商用導入に導いている。

## 2. シングル/マルチプロセッサの ONE-MACHINE 概念による統一<sup>(55)(56)</sup>

### 2.1 ONE-MACHINE 概念

#### 2.1.1 ソフトウェア一系列化の要求

交換機へのサービスの追加とこれを実現するソフトウェアの規模が年々確実に増大し続けている。機能追加、変更に対する融通性は蓄積プログラム制御方式の長所であるが、この長所を活用するに従ってソフトウェアの維持管理工数が増大し始めている。さらに、複数系列の交換機が存在することによりソフトウェア維持管理工数がさらに増大する問題が生じている。機能追加はサービス実現のために本質的に必要であるが、複数系列の交換機の存在は実現する時点での実現技術と経済性との兼ね合いからくるもので本質的なものではない。ソフトウェアの維持管理の工数の削減は大きな課題であり、このため交換機の全規模で一系列化とソフトウェア系列の統一化が求められている。

#### 2.1.2 シングルプロセッサの必要性

マルチプロセッサ制御方式はプロセッサの追加で幅広い容量域に対応し得ることから、この課題を解決するに有力な手段である。しかし、1台のプロセッサでカバーできる容量域ではシングルプロセッサ方式がマルチプロセッサ制御方式よりも経済的であることは自明である。マルチプロセッサの単位プロセッサにはプロセッサ間結合のための付加回路を必要とし、単純なシングルプロセッサより割高となる。また、第II章で定めた機能・負荷分散マルチプロセッサ制御方式は今後のサービス拡大に耐えて、かつ初期投資が少ないという長所を有するが、最小規模でもSGP、CNPの両者を必要とし、シングルプロセッサより多くのプロセッサを必要とすることも事実である。

さらに、蓄積プログラム制御方式は共通制御方式であり、小規模での制御系経済化はデジタル交換機の大きな条件である。経済性追求の原則からは極小規模にシングルプロセッサ制御方式を適用せざるを得ない。通信網全体での総端子数に占める小規模領域の比率は低いものの、局数ではむしろ比率が高いという事情もある。

### 2.1.3 ONE-MACHINE 概念の構築

シングルプロセッサ構成とマルチプロセッサ構成がともに要求される中で、シングルプロセッサとマルチプロセッサの間の系列を統一する必要がある。これはシングルプロセッサとマルチプロセッサが同一通話路に対し、共通のソフトウェアで制御し得ることを意味する（図4-1）。このため、両制御方式の間での系列統一を図る概念を構築する。

## 2.2 ONE-MACHINE 概念の具体化

### 2.2.1 ONE-MACHINE 概念と制御システムの階層構成

シングル／マルチプロセッサ両制御方式の必要性に対し、ONE-MACHINE 概念と呼ぶ統一概念を構築し、この概念の下でシステムを具体的に実現する。ONE-MACHINE 概念はマルチプロセッサのモジュール化推進とシングルプロセッサの簡明性の性質を併せて実現する概念でもある。

ONE-MACHINE 概念は具体的にはシングル／マルチプロセッサ両制御方式の差を隠ぺいするモジュール構成概念である。マルチプロセッサが有効である大規模で複雑なシステムに対し、モジュール化を追求しつつ、モジュール自体は、それぞれシングルプロセッサとしての単純性を追求して設計を進め、またこのモジュールを組合せたものはシングルプロセッサ上でも動作可能とする。

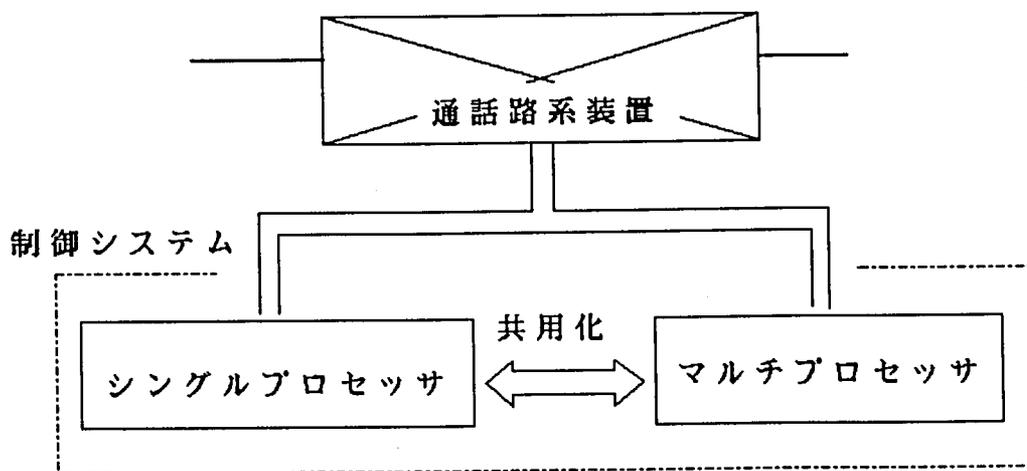


図4-1 シングルプロセッサとマルチプロセッサの共用化

ONE-MACHINE概念は制御系を階層構成化して具体化する（図4.2）。制御系を階層化し、階層インタフェースをシングルプロセッサとマルチプロセッサの間で共通化し、これを参照する上位階層の構成要素が互いに両者で共用化を図る概念である。また、制御系の各構成要素は各階層内でモジュール化して、各階層内のモジュールがシングル／マルチプロセッサ間で共用可能とする。シングルプロセッサは制御系機能の全てを包含した集合であり、各マルチプロセッサ用モジュールは部分集合である。各階層でシングルプロセッサと等価とするためには付加機能が必要となる。各階層内では不足機能をマルチプロセッサでは付加し、上位階層ではシングルプロセッサと等価に仮想化させる。これにより、単純なシングルプロセッサ環境の下で複雑なマルチプロセッサ構成要素が実現できる。

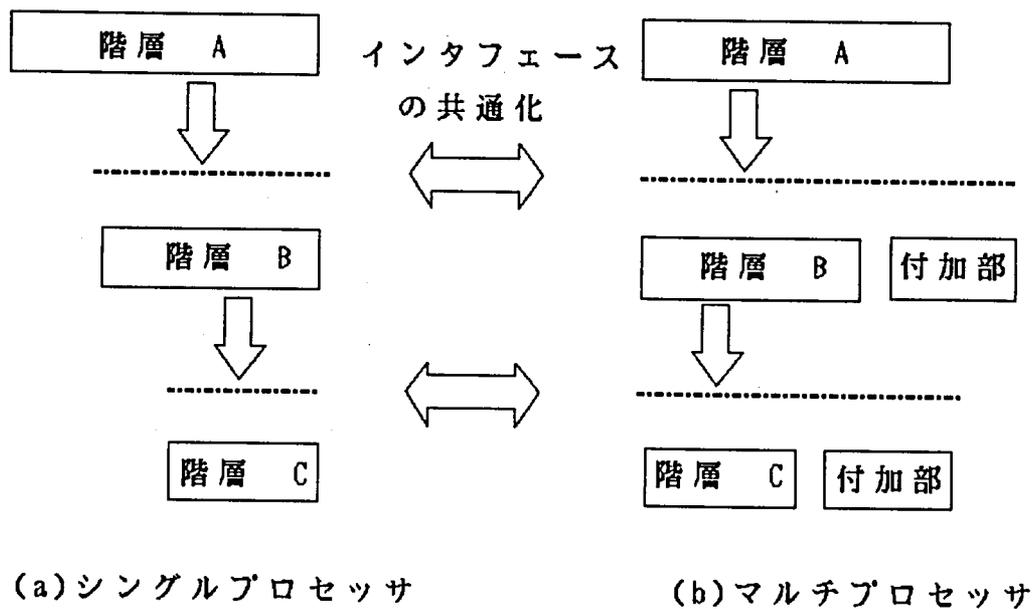


図4-2 ONE-MACHINE 概念に基づく階層化

## 2.2.2 ONE-MACHINE 概念によるシステム具体化の狙い

ONE-MACHINE 概念のねらいを総合して以下に示す。

- ① デジタル交換機的全適用領域に対し、開発品と保守部品の共通化を図る。
- ② 全ソフトウェアがシングル/マルチプロセッサに対して共通化が不可能でも、機能追加変更頻度が高く、規模の大きな呼処理・保守運用関係プログラムはシングル/マルチプロセッサ両制御方式間で共用可能とし、ソフトウェア維持管理の工数削減を図る。
- ③ シングルプロセッサ制御方式が適用された極小規模交換機が、将来 機能・規模において成長し、マルチプロセッサ制御方式に移行せざるを得なかったとき移行容易とする。
- ④ 本来最も単純で簡明、かつ普遍性に富むシングルプロセッサの環境をマルチプロセッサにおいても提供する。

## 3. ONE-MACHINE 概念に基づくマルチプロセッサ構造の階層化

### 3.1 交換機制御系の階層化

シングル/マルチプロセッサ両構成間の構成品の流通化、共用化の観点から設定した交換制御システムの階層構成を図4.3に示す。具体的な階層インタフェースの考え方は以下の通りである。

#### (1) 制御システムインタ

##### フェースの実現法

最上位インタフェースは制御システムインタフェースであり、被制御対象であるシステムである通話路系装置はシングル/マルチ

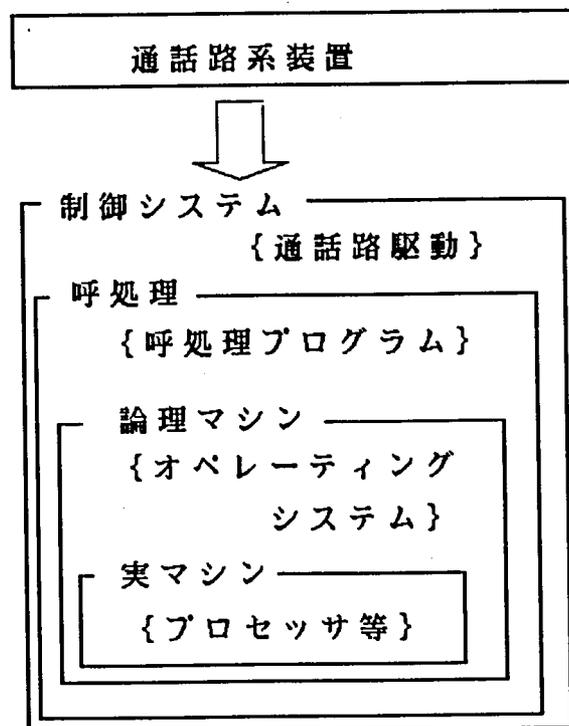


図4.3 制御システムの階層構成

ロセッサ両制御方式に共通化できる。従って、この階層インタフェースは通話路系装置がシングル／マルチプロセッサ両制御方式に接続できるインタフェースを規定する（図4.4）。

#### (2) 呼処理インタフェース

呼処理インタフェースはシングル／マルチプロセッサ両制御システムが通話路系装置の同一の状態変化に対して同一の制御オーダを送出することを規定する（図4.4の論理インタフェース）。

#### (3) 論理マシンインタフェース

論理マシンインタフェースは呼処理プログラムがシングル／マルチプロセッサの両者で走行可能なインタフェースを規定する。これによりシングル／マルチプロセッサ両者の呼処理プログラムが共用できる（図4.5）。

#### (4) 実マシンインタフェース

実マシンインタフェースはオペレーティングシステムがシングル／マルチプロセッサの両者で走行可能なインタフェースを規定する。これにより両制御方式で共用可能なオペレーティングシステムの走行環境を規定する。なお、最下層を構成する実マシン自身もシングル／マルチプロセッサ両制御方式で極力構成品の共通化を図る（図4.6）。

### 3.2 階層インタフェースの実現法

前記の各階層インタフェースの条件を以下の方針で具体化する。

#### (1) 制御システム階層インタフェースの実現法

本インタフェースはプロセッサと通話路装置とを接続する通話路制御信号線および信号線内を伝達する論理信号により具体化される。なおこのインタフェースを構成する信号線の電気、物理条件のシングル／マルチプロセッサ間互換性は容易である。同一のケーブルコネクタや信号駆動素子を用いて、実装条件の統一を図ればよい。

論理的信号については、シングルプロセッサが制御可能な通話路系内の任意の制御点をマルチプロセッサ制御方式も制御可能とし、この逆も保証する。

これに加えて、マルチプロセッサの各プロセッサが利用するプロトコルをシン

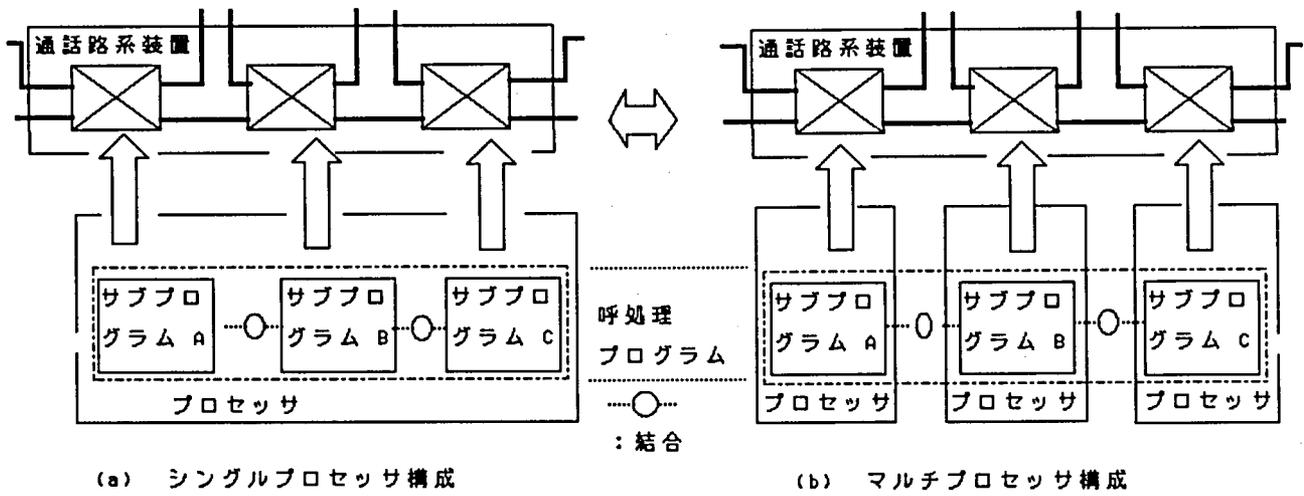


図4.4 通話路系システムの共用化を支援するインターフェース

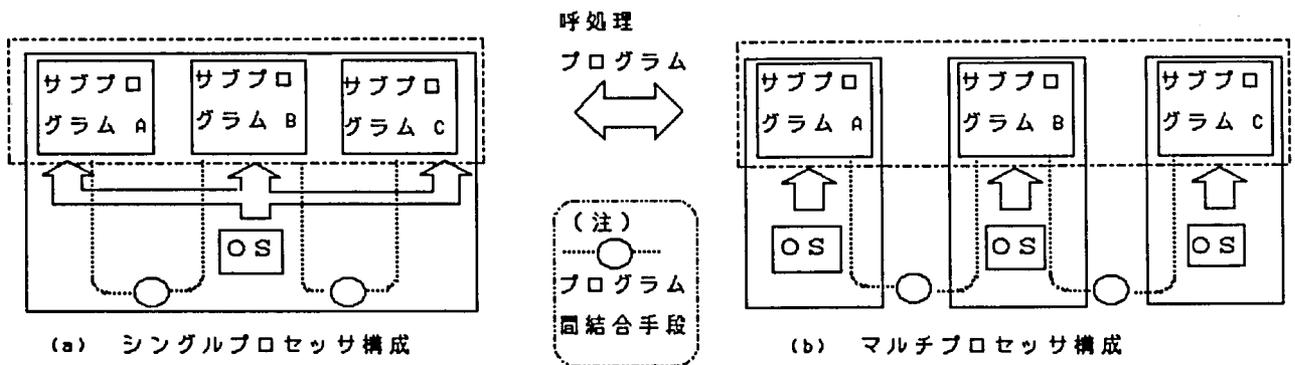


図4.5 呼処理プログラムの共用化を支援するインターフェース

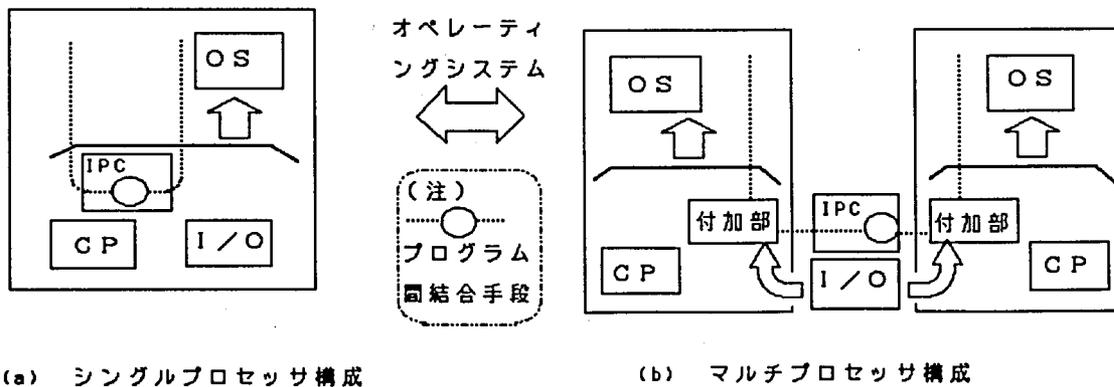
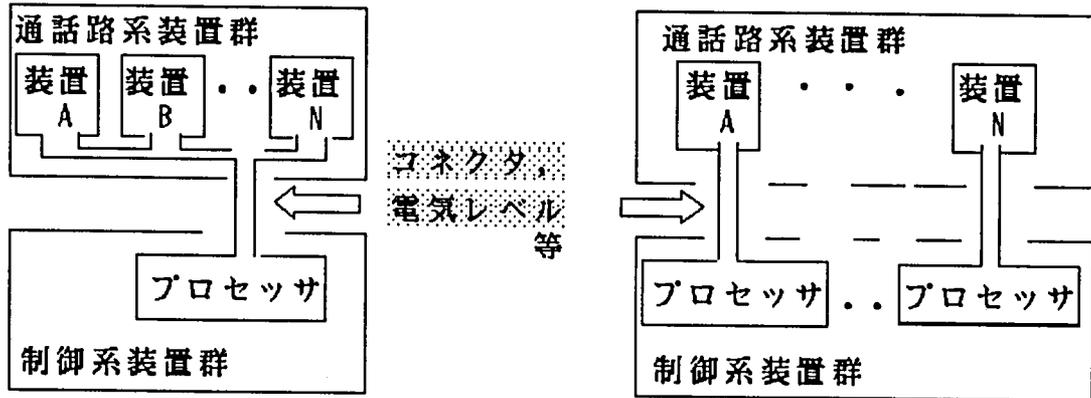


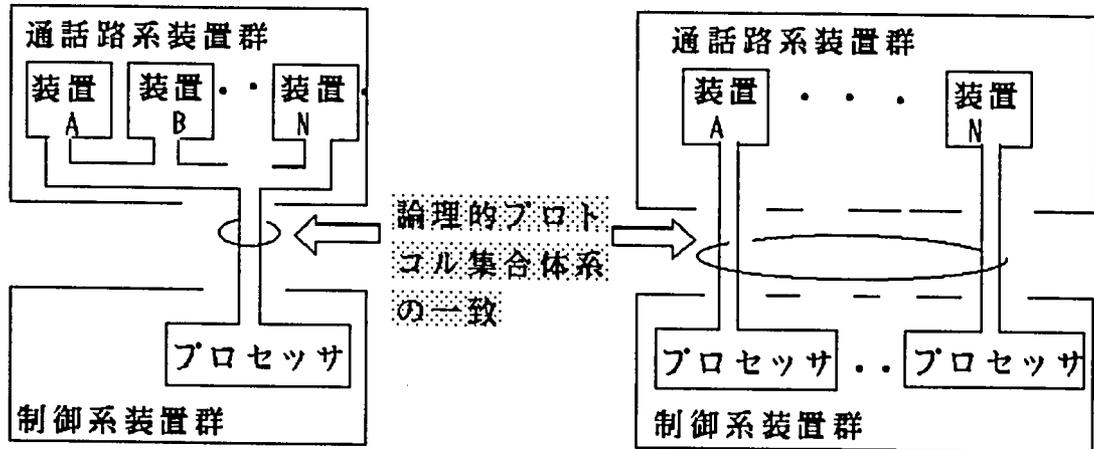
図4.6 実マシンインターフェースの実現

グループプロセッサに縮退させたときシングルプロセッサ内で矛盾を生じないようにする。このため、シングルプロセッサ制御方式のプロトコルとマルチプロセッサの全体のプロトコル集合体系との間に一意的な対応を持たせる(図4.7)。

【物理条件、電気条件の一致】



【論理条件の一致】



(a) シングルプロセッサ

(b) マルチプロセッサ

図4.7 制御システム階層インタフェースの実現条件

## (2) 呼処理プログラム階層インタフェースの実現法

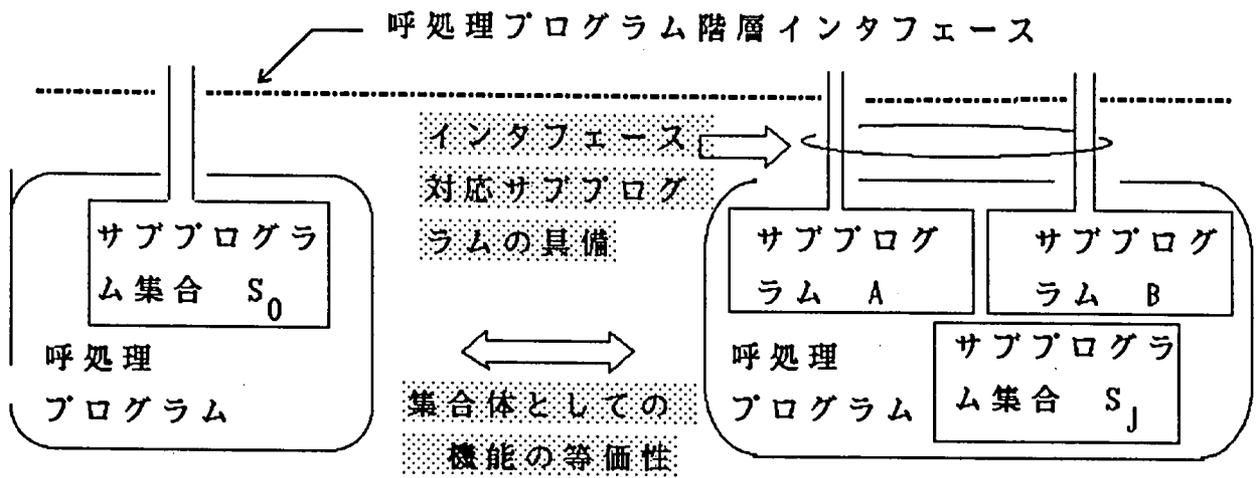
本インタフェースは呼処理プログラム自身で具体化される。呼処理プログラムは、信号の到達などの通話路系装置の状態変化を検出して、処理し、通話路系装置内の通話パスの設定を行なう。この階層では、通話路系装置の同一状態変化に対してシングル/マルチプロセッサ両構成で同一の通話路設定用命令が返送されなければならない。このため呼処理プログラムをサブプログラムに分割して、通話路装置にアクセスするプロセッサには各通話路制御インタフェースに対応したサブプログラムを具備する。さらにマルチプロセッサにおけるサブプログラム全集合とシングルプロセッサの全集合が同じ呼処理プログラム機能を完成させる(図4.8)。具体的には以下の方針で構成する。

- ① シングル/マルチプロセッサ両構成で通話路制御インタフェースに接続された各プロセッサは、各々の信号線に対応するサブプログラムを具備する。
- ② マルチプロセッサでは、呼処理サブプログラムが複数プロセッサに分配されるが、マルチプロセッサの各々のサブプログラムは、シングルプロセッサ内でのサブプログラム間の結合と同様に結合される。
- ③ シングル/マルチプロセッサ両構成でのサブプログラム間結合手段は、下位階層の論理マシンインタフェースを利用する。

## (3) 論理マシンインタフェースの実現法

本インタフェースは、各プロセッサのオペレーティングシステムおよびプロセッサ間通信手段で構成される。論理マシンインタフェースは、呼処理各サブプログラムが複数プロセッサに分散されても、単一プロセッサに縮退されてもそれぞれ走行しえて、かつ互いに連携して呼処理機能を完結させる環境を実現する。この条件を以下の方針で具体化する(図4.9)。

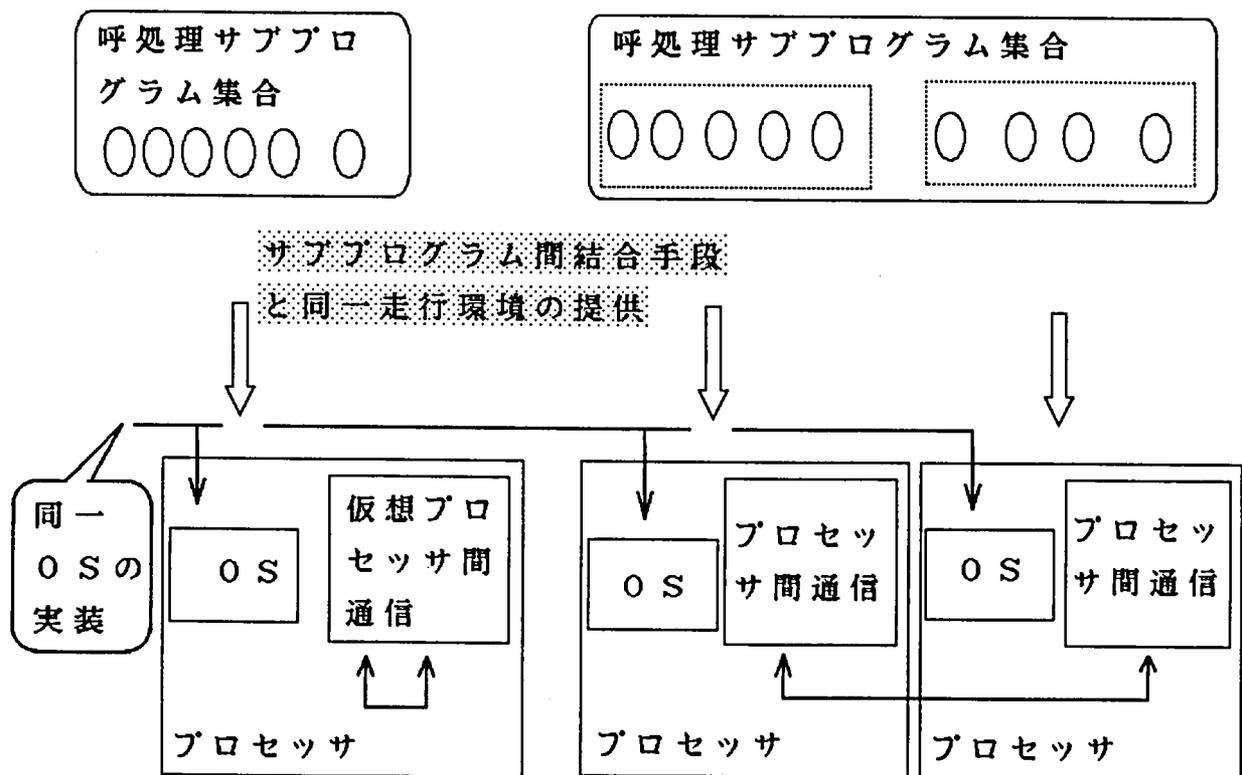
- ① マルチプロセッサの各プロセッサにはシングルプロセッサと同じオペレーティングシステムを搭載する。
- ② サブプログラム間の結合には、シングルプロセッサとマルチプロセッサで同じインタフェースとなるプロセッサ間通信論理手段を具備する。



(a) シングルプロセッサ

(b) マルチプロセッサ

図4.8 呼処理プログラム階層の実現条件



(a) シングルプロセッサ

(b) マルチプロセッサ

図4.9 論理マシンインタフェースの実現条件

#### (4)実マシンインタフェースの実現法

本インタフェースは実マシンそのもので構成され、命令実行制御系、入出力制御系、プロセッサ間通信制御系からなる。シングルプロセッサとマルチプロセッサの各プロセッサで、同一のオペレーティングシステムと同一サブプログラム間結合論理手段が走行しうる環境を実現する。このためには、マルチプロセッサの各単位プロセッサにおいても、シングルプロセッサと同等のハードウェアリソースをアクセス可能とすることが必要である。シングルプロセッサのハードウェアリソースとして以下のものがあり、マルチプロセッサではそれぞれ以下のように実現する。

##### ①命令実行制御部

命令実行制御系ではシングルプロセッサ用のプログラムロードモジュールがマルチプロセッサの各単位プロセッサで走行可能とする(図4.10)。

##### ②入出力系

入出力系はプロセッサと入出力装置(ファイル装置やマンマシンインタフェース等)との間でデータの入出力転送を行なう。シングルプロセッサと等価となるマルチプロセッサ用入出力装置は全てのマルチプロセッサ用プロセッサがシングルプロセッサと同様に入出力系にアクセス可能とすることが必要である。

一方、入出力系は本来システム全体で1式のみであり、プロセッサと1対1で結合するものである。従ってシングルプロセッサ用ハードウェア入出力手段は任意のプロセッサから任意の入出力系に任意のタイミングでアクセスできる機能は具

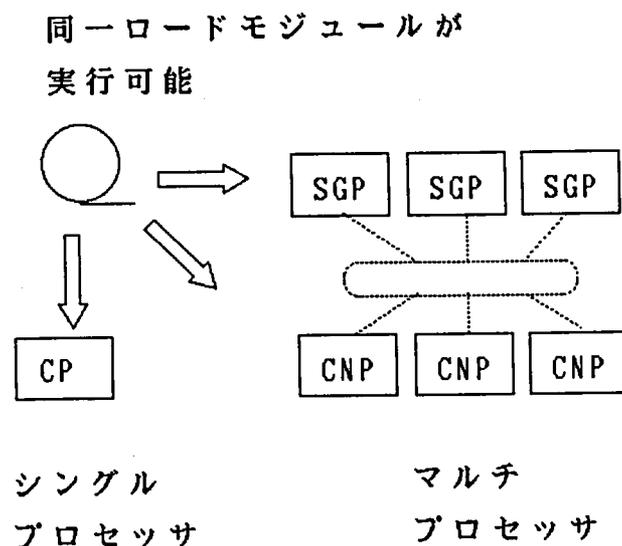


図4.10 命令実行制御部の  
実現条件

備していない。 シングルプロセッサと等価とするためマルチプロセッサ用入出力手段にはプロセッサと入出力系の間付加機構を設ける。 なお、この付加機構の具備すべき基本機能は図4.11に示すようにマルチプロセッサ用プロセッサそれぞれに入出力装置があたかも付加されているように接続待ち合わせを行なうものである。

### ③ プロセッサ間通信制御系

プロセッサ間通信制御系はシングルプロセッサでは単一プロセッサ内に配備される呼処理サブプログラムが、複数プロセッサにまたがって配備されるマルチプロセッサ用呼処理プログラムが同一論理インタフェースとなるサブプログラム間結合手段を規定する。

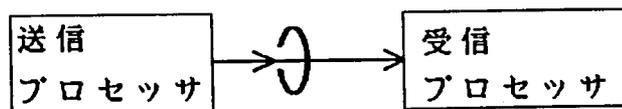
シングル／マルチプロセッサの間で互換となるためには、実質的に異なるハードウェアの構成の違いを吸収するハードウェア手段が必要になる。 複数プロセッサに分かれたマルチプロセッサと同一とするためにはシングルプロセッサにおいても同一インタフェースを持つプロセッサ間通信ハードウェアを具備し、折返し通信を行なうことが構成上で望ましい。 しかし、これはシングル／マルチプロセッサ間の完全な互換性の追求と経済性とのトレードオフとなり、ハードウェアとして具体化する段階での検討課題である。

マルチプロセッサのサブプログラム間通信のシングルプロセッサとの等価化には通信相手の一意的固定化が必要になる。 シングルプロセッサでは各サブプログラムが重複せずに配置されているのに対し、負荷分散されるマルチプロセッサでは同一サブプログラムが複数プロセッサに重複配置される。 この結果、送信先のサブプログラム、すなわちこのサブプログラムが配置されたプロセッサをプロセッサ間通信手段は一意的に決定出来ないという問題がある（図4.12）。

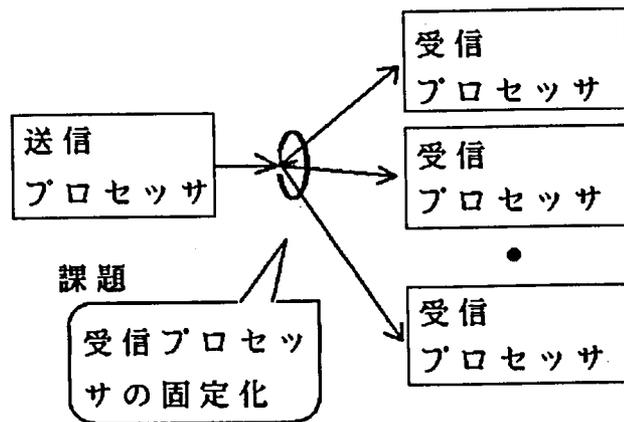
プロセッサ間通信手段は複数プロセッサにまたがった呼処理サブプログラム間を結合する手段としてシングルプロセッサ内のサブプログラム配置と等価な通信手段を提供する。 このため、プロセッサ間通信手段には付加分散された複数のプロセッサの中から単一のプロセッサを選定する負荷分配機構を設ける。 第III章で効率の良いプロセッサ間通信機構を仮定したものの具体化でもある。

	IO接続機構の相違	マルチプロセッサ用IO接続付加機構の機能
シングル		<p>【n-プロセッサと1-IOの接続交換機能】</p> <ul style="list-style-type: none"> <li>・プロセッサからIO起動時，IOへ起動プロセッサ番号を通知</li> <li>・IOからの自律転送要求に転送先プロセッサを指定</li> </ul>
マルチ		<p>【n-プロセッサ接続に伴う入出力の保留時間短縮】</p> <ul style="list-style-type: none"> <li>・プロセッサからIO起動時保留時間を削減</li> </ul>

図4.11 入出力リソースにおける ONE-MACHINE 概念



(a) シングルプロセッサ



(b) マルチプロセッサ

図4.12 プロセッサ間通信における ONE-MACHINE 概念

## 4. マルチプロセッサの設計事例 I

### 既存交換用プロセッサを用いたマルチプロセッサの設計<sup>[57]</sup>

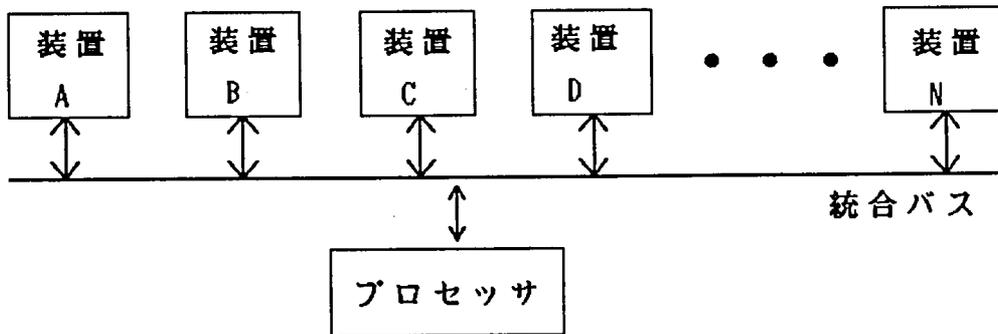
前節まででマルチプロセッサ制御システムの階層の考え方と条件について考察した。本節ではこの階層構成の考え方に従い、実際に試作した例により、各階層構成の実現の要点を述べる。

#### 4.1 通話路駆動階層の設計

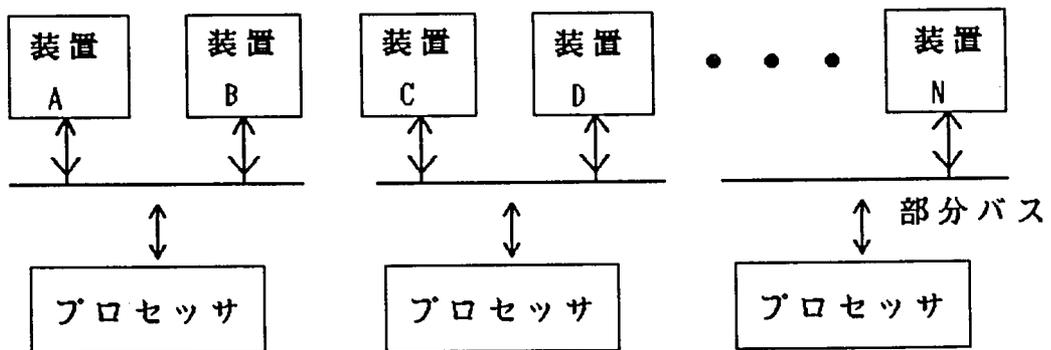
通話路駆動階層は通話路系装置とプロセッサ間の物理的、論理的接続手段を意味する。シングル/マルチプロセッサ構成間での電氣的、物理的條件の互換性にはインタフェース駆動回路を共通として、同一プリント板を利用し、ケーブル、コネクタの統一を図る。また、論理條件の互換性を容易に満たすため、通話路制御インタフェースにはバス（共通母線）形式を用いる。バス形式はケーブルに接続された任意の接続点間の通信が可能であり、またバスを分割することにより分割された部分バスは、その接続された範囲でまたバスを構成する特徴がある。

マルチプロセッサでは各プロセッサごとにそのプロセッサが制御対象とする通話路装置を制御する部分バスを構成する（図4.13）。また、通話路系の全構成装置の制御アドレスを一意的に割当て、一様なアドレス空間を構成して、部分バスでもこれらを集合した統合バスにおいても、一意的に目的の装置にアクセス可能とする。したがってシングルプロセッサ構成では統合バスを構成する。

このプロトコルの例を図4.14に示す。通話路駆動階層の制御オーダは16ビット長の2命令で構成される。第一オーダは制御内容および制御対象の大きな装置の指定、第二オーダはその装置内の内部制御点の指定を行なう。

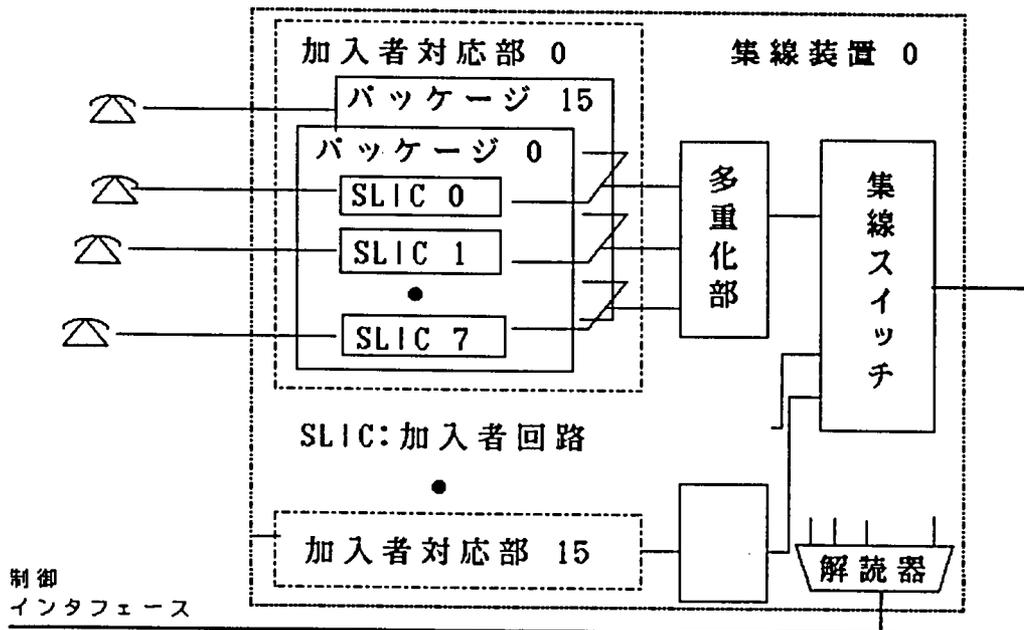


(a) シングルプロセッサ用統合バス



(b) マルチプロセッサ用部分バス

図4.13 共通バスと部分バス



第一語	バス制御情報	* (1)	装置名 (8)	制御オーダ (7)
第二語	バス制御情報	/		指定点 (11)

\*: 予備切替え指定, ( ): ビット数

例)

装置名: 集線装置

制御オーダ: 集線収容指定等

指定点: 加入者対応部番号(4), パッケージ番号(4), SLIC番号(3)

図4.14 通話路インタフェースプロトコル例

## 4.2 呼処理プログラム階層の設計<sup>(58)(59)</sup>

シングル/マルチプロセッサで呼処理プログラムを共用するためには、呼処理プログラム全体をサブプログラムに分解し、分解された結果を各プロセッサに割り付ける方法が有効と考えられる。各サブプログラムへの分割とプロセッサへの割付には、呼処理を機能グループに分割する基準を設け、分割された機能グループをさらにサブプログラムに分けることとする。機能分割に関する大きな基準は前章で述べた信号種別に従う信号処理と呼制御処理との機能分割であり、さらに、図4.15に示す分割要因がある。また、分割された機能グループのプロセッサへの配置を表4.1に示す。

個々の呼の処理を完結するためには、複数プロセッサにまたがった各サブプログラムが同一呼を同一と認識する識別子が必要である。これを呼処理基本番号

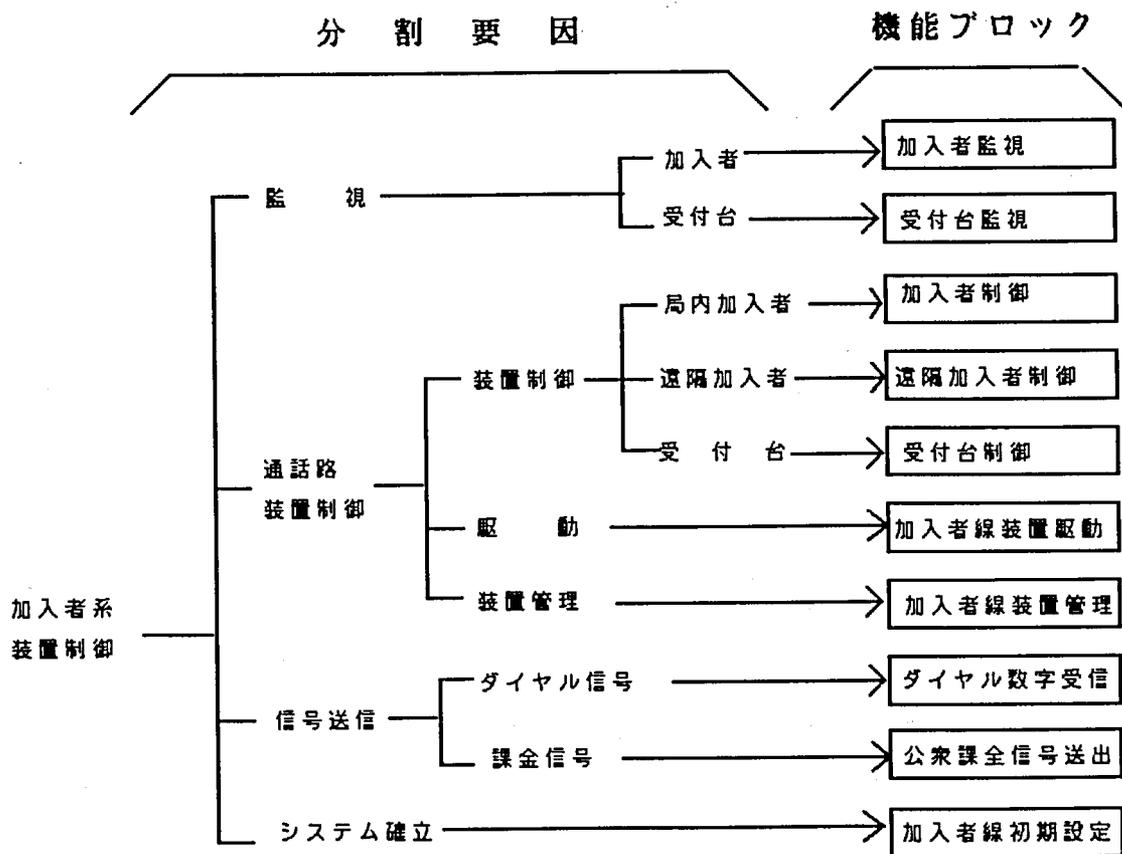


図4.15 呼処理機能分割要因（加入者線制御の例）

表 4.1 サブシステムの機能配置

サブシステム	加入者サブシステム	中継線サブシステム	呼制御サブシステム	通話路サブシステム	共通線信号サブシステム	保守運用サブシステム	主制御(注)サブシステム
機能分担	加入者監視 受付台監視 数字受信 プリトランスレーション 兼線段 接続制御 加入者制御 受付台制御 課金信号送出(対公衆)	トランク監視 選択数字受信 選択数字受信 トランク制御 課金信号送出(対XB)	呼接続状態管理 ルート情報翻訳 加入者情報翻訳 テナント情報翻訳 課金 トランク管理	通話路制御 通話路管理 通話バス管理	共通線信号装置制御	サービス オーダ処理 局データ変更管理 コマンド制御 装置試験 トラヒック観測 トラヒック制御 システム状態表示	実行制御 メモリ管理 再開処理 システム構成管理 制御系装置管理 IOCS
プロセッサ	LNP RNP	TSP	CNP	CNP	CSP	MCP (CNP)	CNP LCP TSP CSP

(注) 各サブシステムに必要なOS機能である

PRID	PRN	IPTN
(2bit)	(3bit)	(16bit)

PRID: プロセッサ種別

(LNP/TSP/CNP/MCP)

PRN: プロセッサ番号

(シングルプロセッサでは一義的に"0")

IPTN: プロセッサ内でのトランク番号

図4.16 トランク番号の階層定義

と呼ぶ。呼処理基本番号は呼の生起した通話路上の加入者線、中継線等の回線端子認識番号として表現できる。基本番号の付与法を図4.16に示す。呼の着信や発信を生じる回線端子は、複数のプロセッサに分割されており、回線端子識別番号（IPTN）の他にプロセッサ識別子も付与して処理の効率化を図る。これにより、CNPが呼制御処理で確保したリソースの所属するプロセッサや処理結果を返送あるいは送信するプロセッサが容易に認識できる。

呼処理機能完結に向けた呼処理シーケンス制御はこの呼処理基本番号を用い、プロセッサ間通信手段を経由してサブプログラムの制御によって行なわれる。呼と1対1に対応するプロセッサへの処理結果の送付は各サブプログラムが処理完了時に呼処理基本番号内のプロセッサ識別情報を用いて、送信先の指定を行なう。負荷分散されたことにより、プロセッサを特定する必要がない送信先には、サブプログラムが処理完了後にプロセッサ任意指定の通信を指示する。下位階層のプロセッサ間通信手段が負荷分配機能を用いて、プロセッサの指定とサブプログラムへの通信を行なう。シングルプロセッサではこのプロセッサ番号を“0”に固定することにより、マルチプロセッサ方式と同じ基本番号体系を利用でき、シングル/マルチプロセッサ互換性が可能となる。

#### 4.3 オペレーティングシステム階層の設計

この階層は呼処理サブプログラムがシングルプロセッサとマルチプロセッサで共通に動作可能な環境を提供する。このため、まずシングルプロセッサの走行環境をマルチプロセッサの各単位プロセッサに実現する。具体的にはシングルプロセッサに適用したオペレーティングシステムをマルチプロセッサの各単位プロセッサに適用する。

また、呼処理サブプログラム間を結合して呼処理プログラム機能を完結しなければならない。これを実現するサブプログラム間通信の論理手段にはインタフェースを統一して各プロセッサ内で閉じたメモリ上での通信手段とプロセッサ間通信制御装置を経由した手段の2種を設ける。全ての構成にプロセッサ間通信ハードウェアを設けることは理念の徹底ではあるが、シングルプロセッサでは小規模領域への適用を考慮して極力ハードウェアを経済化することが必要であり、プロセッサ間通信機構の付与は困難である。また処理量の面からは、プロセッ

サ間通信は呼処理のオーバヘッドであり、シングル/マルチプロセッサの両者で極力単純化することが必要である。

#### 4.4 マルチプロセッサ用実マシン階層の設計<sup>[68]~[65]</sup>

##### 4.4.1 マルチプロセッサの命令実行制御系

###### (1) マルチプロセッサ用単位プロセッサのシングルプロセッサ等価性

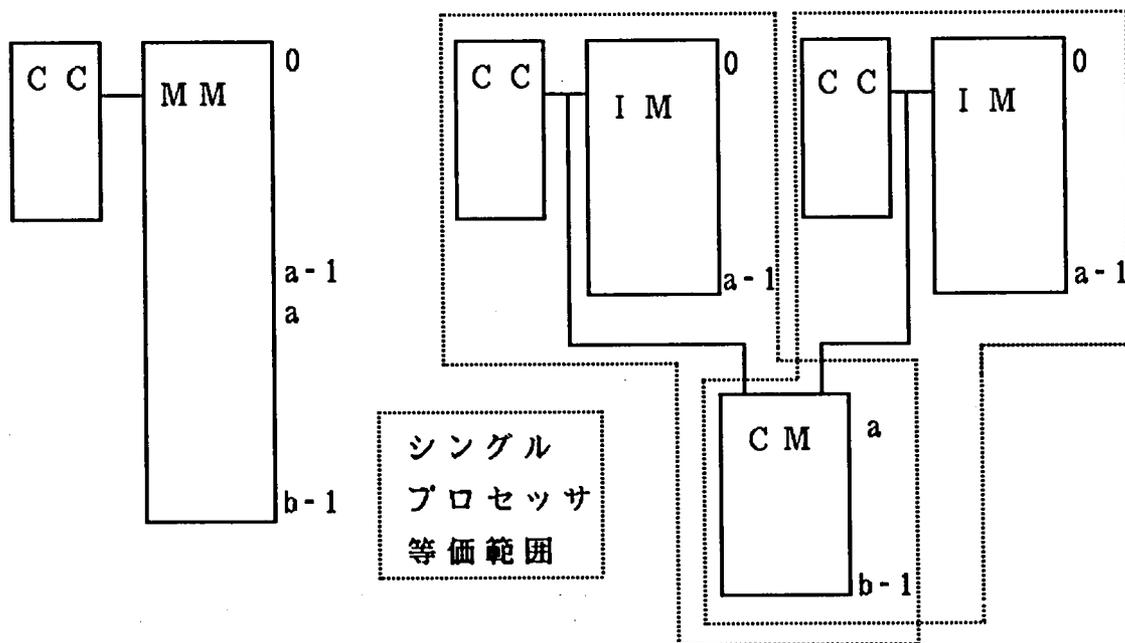
実マシン階層命令実行制御系の ONE-MACHINE 概念の条件はシングルプロセッサ用のロードモジュールがそのままマルチプロセッサの単位プロセッサに適用できることとした。この条件に以下のように適合させる。

信号処理プロセッサ (SGP) : 各信号処理プロセッサにはシングルプロセッサ制御方式と同じプロセッサを用いる。また、メモリ空間はシングルプロセッサ制御方式と同じ0番地から始まる同一アドレス空間を割り当てて、ロードモジュール互換性を保証する。なお、複数プロセッサが同一アドレス空間を共有しても各信号処理プロセッサのメモリアクセスインタフェースを他のプロセッサとは独立にするため矛盾は生じない。

呼制御プロセッサ (CNP) : 呼制御プロセッサもシングルプロセッサとのロードモジュール適用可能性を保証するため、各呼制御プロセッサ間で同一メモリアドレス空間を割り当てる。ただし、呼制御プロセッサ群には全ての呼制御プロセッサからアクセスできる共通メモリの配備が必要であることを考慮しなければならない。

共通メモリの配備法として、呼制御プロセッサ群の全てのメモリを共通メモリ化する方法が考えられるが、メモリアクセスが単一メモリに集中するため、性能向上が図れないという問題がある。このため、呼制御プロセッサのメモリ空間を以下のように構成する(図4.17)。

- ① 呼制御プロセッサ用メモリとして、呼制御プロセッサの各々が個別にアクセス可能な個別メモリと全ての呼制御プロセッサが共通にアクセス可能な共通メモリを設ける。
- ② 全個別メモリには信号処理プロセッサと同様に0番地から始まる同一メモリ空間を割り当てる。
- ③ 共通メモリは個別メモリと重複しないメモリ空間を割り当てる。そのアドレ



(a) シングルプロセッサ用  
命令実行リソース

(b) マルチプロセッサ用  
命令実行リソース

図4.17 命令実行リソースのシングル  
／マルチプロセッサ等価性

CC : 中央制御装置  
MM : 主メモリ  
IM : 個別メモリ  
CM : 共通メモリ

ス空間は呼制御プロセッサ間で同一とする。

なお、ソフトウェアの機能配置は上記のメモリ空間に適合化させて以下の配置とする。他プロセッサとの競合がなく、高速にアクセスできる個別メモリには高頻度で利用されるプログラムと処理中の呼のワークメモリを配置する。複数プロセッサの競合を生じるが、全ての呼制御プロセッサがアクセスできる共通メモリにはリソース管理データ、安定状態にある呼の管理情報の退避領域を配置し、任意のプロセッサが必要に応じてアクセス可能とする。

## (2) 単位プロセッサの構成

マルチプロセッサのねらいはもともとVLSI時代を想定した経済的で効率的な交換機用制御方式の確立である。しかし、ここで述べるマルチプロセッサの実用化実験には以下の理由から中小局向きの改良D20電子交換機用プロセッサを<sup>[24]</sup>用いることにした。

## マイクロプロセッサ技術の未成熟

- ①ここで報告するマルチプロセッサの初期の実験段階ではマイクロプロセッサの技術はまだ未成熟であった。本研究の試作を実際に進めていた段階では、ようやく8ビットマイクロプロセッサが普及し始めた段階であり、実用化を指向した目標には不適切であった。
- ②8ビットのマイクロプロセッサを用いた交換機制御方式は、研究レベルでは試みられていたが、マルチプロセッサ用オーバヘッドの増大からいくら数を増やしても2万erlに及ぶ大局の実現は困難であり、たとえ実現されてもその経済性は実用に耐えるものでないことが机上検討により容易に想定された。

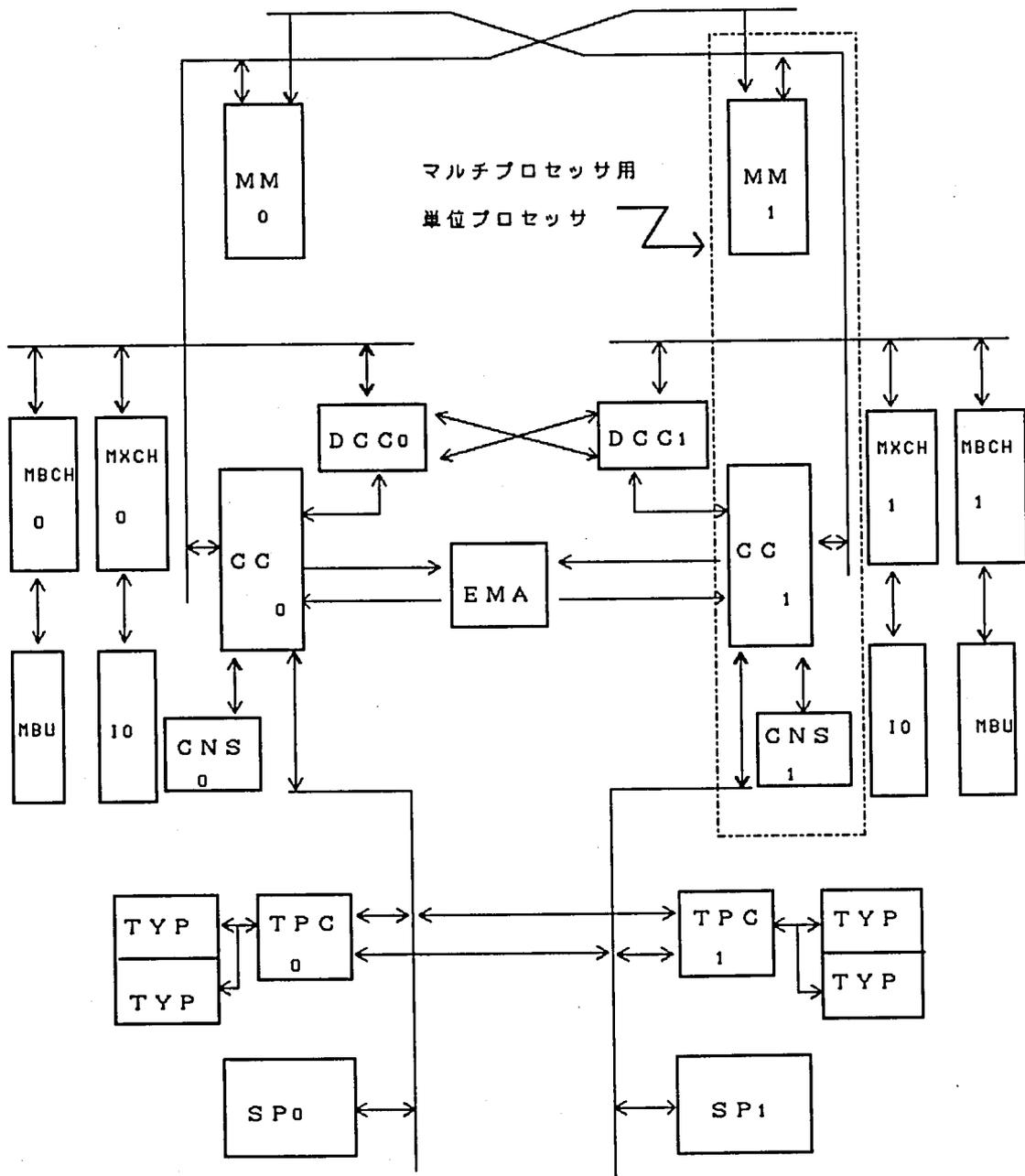
## 交換用プロセッサの効率性

- ①既存交換用プロセッサを利用することにより、空間分割電子交換機において蓄積した膨大なソフトウェア資産、交換用ソフトウェアノウハウが流用できる。
- ②改良D20交換機は空間分割電子交換機であるが、デジタル交換機も処理の面からは空間分割電子交換機と大きく変わるものではなく、交換処理に適した命令を利用でき、効率がよい。
- ③24時間運転マルチプロセッサの実現には、初期設定や起動や停止などの制御を行なう必要があるが、交換用プロセッサはもともと二重化予備制御のために装置間制御のインタフェースを具備している。

## 小規模交換用プロセッサの選定

- ①交換用プロセッサには、D10形電子交換機用高速プロセッサがあるが、高速プロセッサではほとんどの適用領域を1プロセッサでまかなえ、マルチプロセッサの狙いとするビルディングブロック化制御系の実現が困難となる。極力、小規模で廉価なプロセッサの利用が必要であった。
- ②本格実用化には、将来VLSIプロセッサを用いることを前提としたが、VLSIプロセッサの能力は改良D20交換機用プロセッサの性能程度と想定された。

図4.18に改良D20処理装置の構成を示す。従来、電子交換機用プロセッサでは、高信頼化のため中央制御装置と主メモリは互いに独立に予備を取れる構成



MBCH : 磁気バブルチャンネル  
 MXCH : マルチプレクサチャンネル  
 MBU : 磁気バブル装置  
 DCC : 制御交差回路  
 EMA : 緊急制御回路  
 CC : 中央制御装置

MM : 主記憶装置  
 TPC : タイプライタ制御装置  
 TYP : タイプライタ  
 PTR : 紙テープ読取り装置  
 SP : 通路路制御装置  
 CNS : コンソール

図4.18 改良D20処理装置の構成

としていたが、改良 D 2 0 処理装置は中央制御装置と主メモリを一体とした完全二重化予備方式を採用している。このため、中央制御装置と主メモリを合わせてマルチプロセッサ用単位プロセッサがそのまま実現できる。なお、マルチプロセッサの性能に関する仕様は本章「4.7 性能評価」において逐次説明する。

#### 4.4.2 入出力アクセス付加機構とプロセッサ間通信制御機構の設計

マルチプロセッサの単位プロセッサには、既存プロセッサがほぼ流用できるのに対し、入出力アクセス付加機構とプロセッサ間通信制御機構はマルチプロセッサ用の新たな装置として実現しなければならない。なお両者は以下のように類似の装置であり、統一して実現する。実現するハードウェアはプロセッサ間通信制御装置（IPC：Interprocessor-Communication-Controller）と呼ぶ。

- ① 両付加機構はともに任意のプロセッサから起動を受けて、プロセッサのメモリとの間で自律的に情報転送を行なうものである。
- ② 転送に伴う制御の詳細は異なっても、プロセッサからの起動に対する応答、転送処理、転送後の報告処理など両者は同種の処理シーケンスをたどる。
- ③ 制御の詳細はマイクロプログラムで吸収すれば制御機構は同じとなる。
- ④ 入出力系には、将来入出力プロセッサが付加され、高度化されることが予想でき、入出力動作もプロセッサ間通信として統一されると考えられる。

##### (1) 機能の実現

###### 入出力アクセス付加機構

ONE-MACHINE 概念を具現するマルチプロセッサ用入出力アクセス付加機構は、マルチプロセッサの各単位プロセッサがあたかも各入出力装置を自プロセッサに固定接続されているように仮想させる機構である。本来 1 対 1 で固定接続されていたプロセッサと入出力装置の間に本機構が介在し、任意のプロセッサが 1 式の入出力装置に任意にアクセス可能とする。この機能をプロセッサ間通信制御装置のマイクロプログラム実行制御機構が具体化する。主要な機能は以下の通りである。

## ① プロセッサからの入出力系アクセス手段

### a) プロセッサからのアクセス競合防止：

入出力装置へのアクセス機能はIPC内の単一のマイクロプログラムにより実現されている。各マイクロ命令はリエントラント構造となっており、マイクロ命令毎に制御対象とする装置をハードウェア論理で指定する。したがってプロセッサ毎にマイクロ命令実行権を確保する機構を具備すればよい。IPC内マイクロプログラムの実行制御回路内にマイクロプログラム実行要求要求を競合整理する回路を設けて実現する。

### b) プロセッサと指定入出力装置の接続：

入出力アクセス権を獲得したプロセッサ番号を保持する回路と、これを制御するマイクロプログラムが保持情報を参照してプロセッサと指定入出力装置の接続を行なう。

## ② 入出力系からのプロセッサアクセス手段

### a) 転送中断からの回復機能：

機械動作と電子回路動作の速度の違いや通信速度の規定等により入出力装置はプロセッサの速度に追従できず、入出力装置は断続的にデータ転送と中断を行なう。各入出力装置でデータ転送の準備が整うと起動プロセッサへのデータ転送のため、入出力装置側から転送要求が生じる。この場合、シングルプログラムでは入出力装置が通信するプロセッサが固定されていたが、マルチプロセッサでは任意のプロセッサが入出力装置にアクセスするため、入出力装置の転送毎に起動プロセッサを認識してアクセスしうる機能が必要になる。この機能はマイクロプログラムで実現されるものであり、①-a)と同様、対応するマイクロプログラムによって利用中のプロセッサ番号を識別するテーブルを具備し、この番号を用いてプロセッサにアクセスする。

### b) 入出力装置からの自発的な転送要求：

タイプライタからの転送要求のようにプロセッサからの起動がなくても転送要求が生じることがある。この要求は通常システムの中核となるプロセッサを指定するが、均質化を指定した本マルチプロセッサでは任意のプロセッサが対象となり得る。ただし、第V章高信頼化で述べるように全体で1個

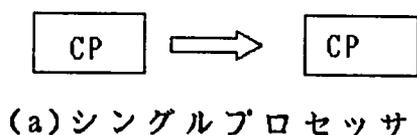
のマスタープロセッサを規定している。 マスタープロセッサ番号を指定できるメモリをIPC内に具備し、システムの初期設定時に本メモリにマスタープロセッサ番号を設定する。 マイクロプログラムはこのような自発通信が生じるとメモリを参照して通信相手のプロセッサを選択する。

### プロセッサ間通信機構

呼処理サブプログラム間の結合を図るプロセッサ通信には図4.19に示す形態がある。 これらの通信はシングルプロセッサと同様に送信先を一意的に固定化できなければならない。 これらの機能の実現には以下の考え方をとる。

#### ① CNP から SGP への通信

CNP から SGP への通信はシングルプロセッサの通信に対応する。 送信すべき SGP の選択は接続された通話路装置の選択に一致することによる。 呼と送信すべき SGP は呼の識別がもともと回線端子を用いて記述されていることによる。 結果として SGP への通信のために IPC は特別の機構を必要としない。 送信先を解釈して情報転送を行なうマイクロプログラムを起動するだけでよい。



#### ② SGP から CNP への送信

CNP への通信には動的に負荷分散した CNP 群の一つを如何に選ぶかが課題である。 プロセッサ間通信が送信先を単一プロセッサと仮想化するのに対し、動的に負荷分散しているプロセッサ群は均質であり、区別がつかない。

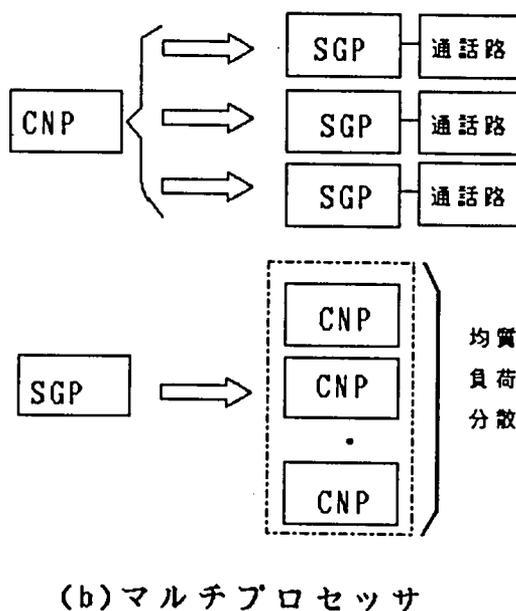


図4.19 プロセッサ間通信の例

なお SGP から CNP への通信には負荷均衡を図ることが前章での課題でもあった。このため、個々のプロセッサ名を指定せず、CNP 群ヘータを送れば負荷分散 CNP のうち処理余裕のある一つを選定して通信を行なう自律的負荷分散機構を設ける。

自律負荷分散機構はマイクロプログラムの制御能力を利用してプログラム制御により実現する。処理余裕のある CNP の認識は、まずハードウェア初期設定時に各 CNP から IPC に、また通常時には CNP からの通信に併せて処理余裕のあることを制御コマンドで登録しておく。マイクロプログラムは登録された CNP の中から 1 つを選択して順次送信する。

## (2) IPC の動作と制御課題

入出力やプロセッサ間の各種の情報転送制御に関する IPC の動作を図 4.20 に示す。多種多様なマイクロプログラム制御があり、これを効率的に実現することが IPC の課題である。これらの動作シーケンスと対応するマイクロプログラム制御への課題を以下に示す。

### ① 起動シーケンス：

プロセッサ間通信や入出力が必要となって IPC を起動するプロセッサは IPC の起動命令を発出する。次に、IPC がこの起動命令に正確に対応していることを確認することにより、命令を発したプロセッサは次命令に進め、処理を進行させる。一方、プロセッサからの起動を受けた IPC は、通信相手と通信種別を識別し、通信相手の状態を認識して通信可否を判断し、この結果を起動プロセッサに返送する。プロセッサは IPC からの返答が来なければ、起動命令実行中のまま待機しており、次命令へ進行することが出来ない。プロセッサの処理効率の向上のためには、IPC は速やかに返答しなければならない。

### ② 転送シーケンス

ファイル装置や入出力装置の転送に係わるマイクロプログラムは、それまで休止していたが、転送要求が発生すると動作状態に入る。このマイクロプログラムの制御の下でデータ転送のためのメモリあるいは入出力装置へのアクセスがなされる。また、データ転送に併せて転送完了語数や転送先アドレスの更新がなされる。

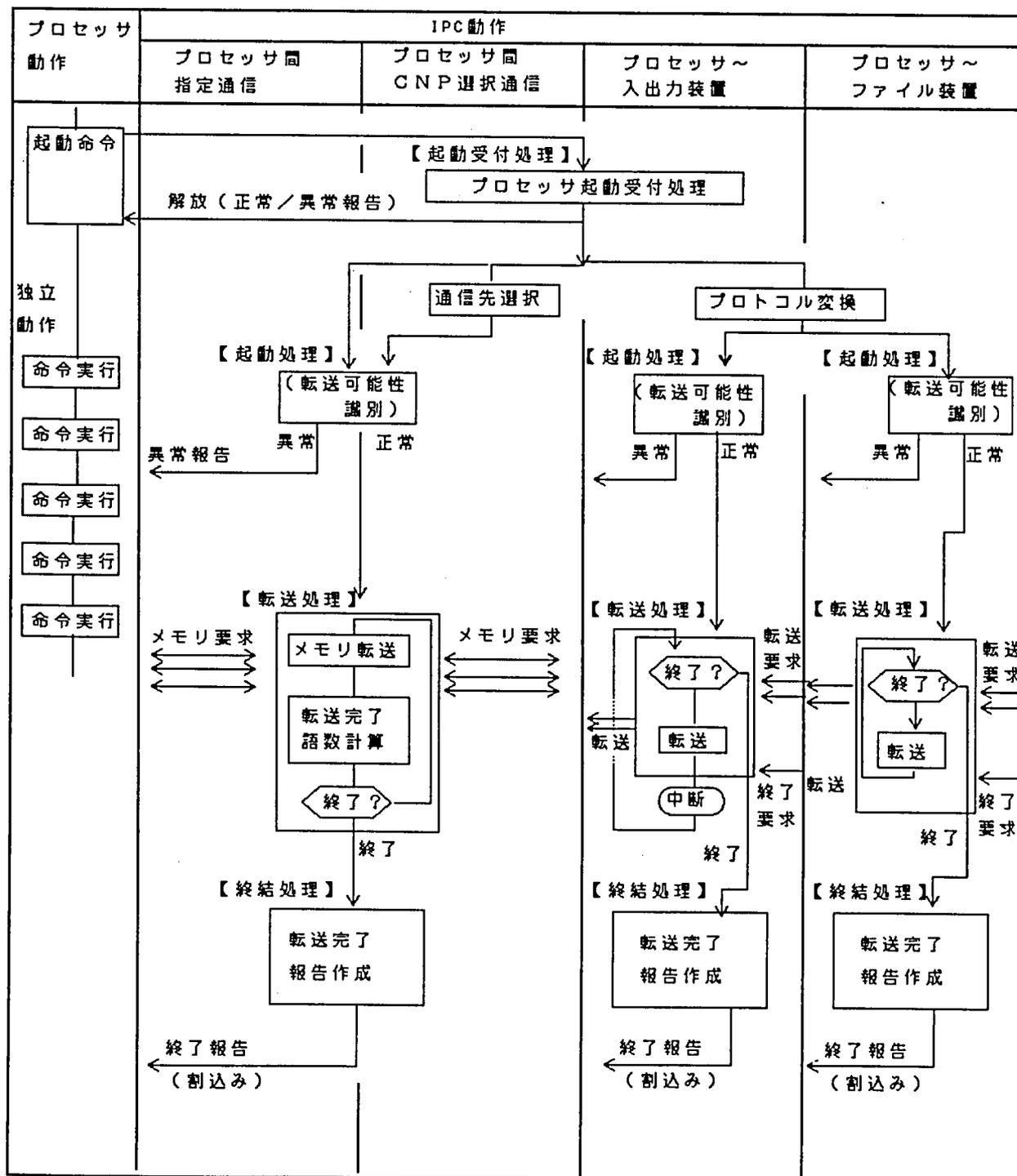


図4.20 IPCの代表的動作

この段階では、各種転送要求元や転送形態に合わせてこれらに対応するマイクロプログラムを効率的に切り替える手法が必要となる。また、入出力系にはファイル装置やデータ通信制御装置のように自律クロックで転送するものがあり、これに合わせた厳しい実時間制御能力が必要である。

### ③ 終了報告シーケンス

プロセッサの指定した語数の転送完了や機器の異常等をマイクロプログラムが確認し、転送の成功または不成功を起動プロセッサに報告する。この制御もマイクロプログラムの休止と動作の状態遷移を伴い、マイクロプログラムの状態の遷移に利用される処理ステップの効率化が必要となる。

### (3) マイクロプログラム制御法

IPCの制御には、多種の入出力動作や情報転送に伴う複雑な手順及び多重処理や処理即応性（実時間保証）の実現が必要である。動作の多様性や手順の複雑さの吸収にはマイクロプログラムによるプログラム論理の利用が有効である。しかし、マイクロプログラムには1命令毎の逐次処理とこれに伴う制御の低速性という欠点がある。

IPCの制御構成にはマイクロプログラムの低速性という欠点を補う改良が必要である。このため、以下の考え方によりマイクロプログラム制御の改良を図る。

#### (i) マイクロプログラム実行制御回路へのハードウェア支援機構の付加

複数装置からの実行要求ごとのマイクロプログラムを次々と切り替えて実行するマイクロプログラム多重制御機構を付加し、マイクロプログラム自身による多重切り替えのオーバーヘッドをなくす。また、外部からのマイクロプログラム起動要求にハードウェアキューで待ち合わせ、マイクロプログラムによる制御要求受付処理をなくして命令実行の効率化を図る。

#### (ii) 実時間性の厳しい制御の優先実行

ファイル装置のデータ転送のようにマイクロプログラム実行待ち合わせによってオーバーランを生じるものがあり、マイクロプログラム実行選択に優先レベルを設けて制御する。

(iii) 正常処理の見込み実行と例外処理の後処理化

データ転送は一連の転送を終えて完了確認を行なう。この制御には最終過程で異常を示せば良く、途中では必ずしも逐一正常/異常を表示する必要はない。このため、実時間性の厳しい処理は正常とみなして先行制御し、異常が生じていた場合は後の過程で修正を行なう。

以上の考え方に基づくIPCの構成概念図を図4.21に示す。主要な動作は以下の通りである。

#### マイクロ命令シーケンス制御

マイクロプログラムのシーケンス制御はマイクロ命令内の実行シーケンス制御指定フィールドにより制御される。マイクロ命令が中断可能を指示するとシーケンス制御部が高優先のマイクロプログラム実行要求の有無を判断し、高優先のものがあれば、実行中のマイクロアドレスを退避し、次マイクロ命令として高優先要求を受け付け実行する。

#### マイクロプログラムリエントラント性の保証

マイクロプログラム実行中には要求源ごとの保持回路により実行中を表示し、マイクロ命令解読器とこの実行表示の論理積により対象回路が選択され、制御される。マイクロ命令の論理シーケンスの保証は優先レベルに対応したマイクロ命令アドレス保持回路を具備することにより行なう。

#### マイクロプログラム実行優先制御

マイクロプログラム実行要求を登録するハードウェアキューを具備し、このキューには第1から第3までの3レベルの優先度を設ける。

#### マイクロプログラム実行開始アドレス

マイクロプログラム起動要求源に対応してマイクロプログラム固定アドレス発生回路を具備し、要求が受け付けられたときこの固定アドレスをマイクロプログラム起動アドレスとする。

#### プロセッサ起動要求への対応

プロセッサからの起動要求は一旦受付キューに登録して、登録後ただちにプロセッサへ受け付け報告を出す。受付キューの登録結果はさらに第1レベルへ登録し、マイクロプログラムの制御を受ける。

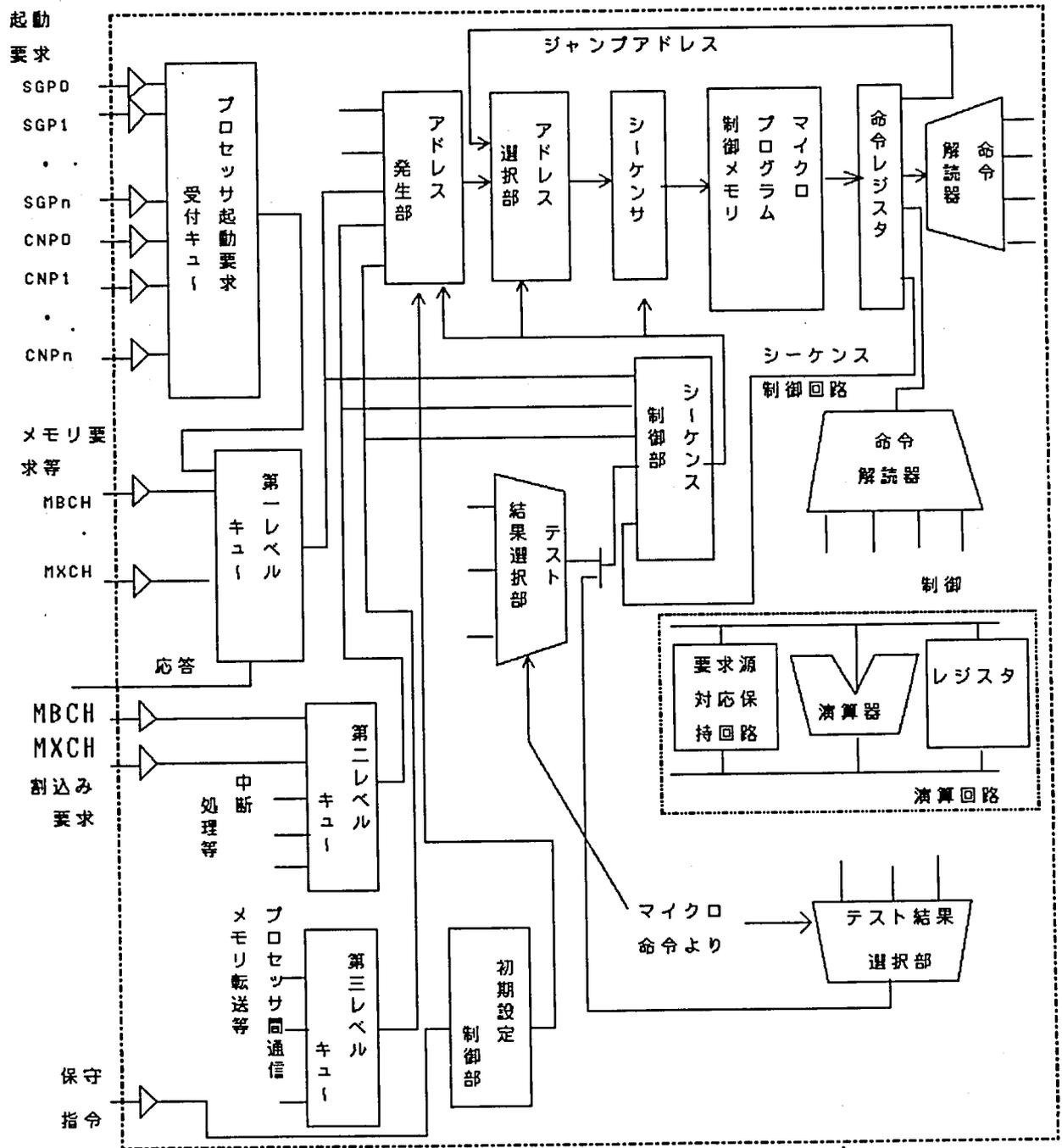
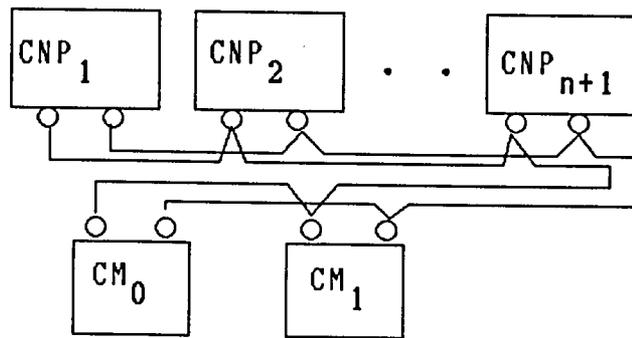


図4.21 IPC回路構成の概念図

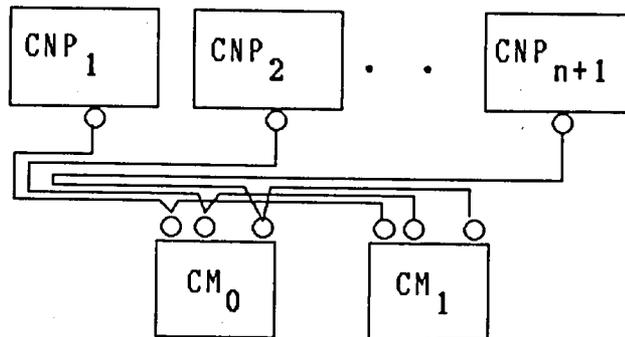
#### 4.5 共通メモリの接続法

共通メモリ装置はプロセッサ間通信制御装置と並びマルチプロセッサ用付加装置として重要な装置である。本装置においては、プロセッサと共通メモリを接続する装置間の接続法が重要な検討課題となる。この接続機構が共通メモリ装置の経済性や性能を、さらにはマルチプロセッサの性能規定要因となる。接続機構の対案には、図4.22に示すように共通バス接続法と直接リード接続法があり、両者を比較検討する。

まず、接続に要する金物量を比較する。これらの接続に要する金物量は各方式の信号数と接続点の積和で表される。両方式の信号数と接続点数の積和および両者の差は図4.22からそれぞれ以下のように示される。



(a) 共通バス接続



(b) 直接リード接続

図4.22 共通メモリ接続法の対案

$$CN_{BUS} = 2 \cdot \{(n+1)+2\} \cdot a = (2n+6) \cdot a \quad (4.1)$$

$$CN_{LEAD} = 3 \cdot (n+1) \cdot a = (3n+3) \cdot a \quad (4.2)$$

$$CN_{BUS} - CN_{LEAD} = (3-n) \cdot a \quad (4.3)$$

なお、 $a$ は各方式の信号数を表す定数である。ただし、共通バス接続法ではバスの使用要求と使用許可を示すための信号が増加するが、全体数からは無視できる範囲であるため、信号数を同一と近似している。従って、 $N_c$ が3以上ではバス接続が有利である。

次に性能面から両方式を考察する。共通バス方式は共通メモリアクセスに伴う保留時間に共通メモリのサイクルタイムとバスの使用時間の両者を含み、保留時間が増大する問題がある。バス保留時間についても高処理能力が必要な大規模ではバスの接続点が増大し、ファンアウトの増大やバス接続線長の増大を招き、バス保留時間を増大させる問題がある。

直接リード結合には競合整理が共通メモリ装置内で行いいう特徴がある。このため、共通メモリにアクセスしたときの保留時間は、共通メモリ内の動作に限定され、しかも線長を短くできることから性能向上には有効である。

共通バス方式は性能面で劣るが、大規模領域における金物量で優れる。一方、交換機の導入数の分布は、小規模ほど大きい特徴があり、分布に従って金物量を荷重平均すると直接リード結合方式の金物量は多いとは言えず、性能面で優れる直接リード結合方式がむしろ有効と考えられる。このため、直接リード結合方式により実現する。

#### 4.6 マルチプロセッサの実現結果

実現したマルチプロセッサハードウェアの全体構成を図4.23に示す。SGPとCNPからプロセッサ間結合用インタフェース付加部を除いたものがシングルプロセッサとの共用化範囲である。また、入出力装置1式もシングルプロセッサと共用可能としている。CMやIPCがマルチプロセッサ用の大きな付加装置であり、金物量の観点からはマルチプロセッサ化に伴う大きなオーバヘッドで

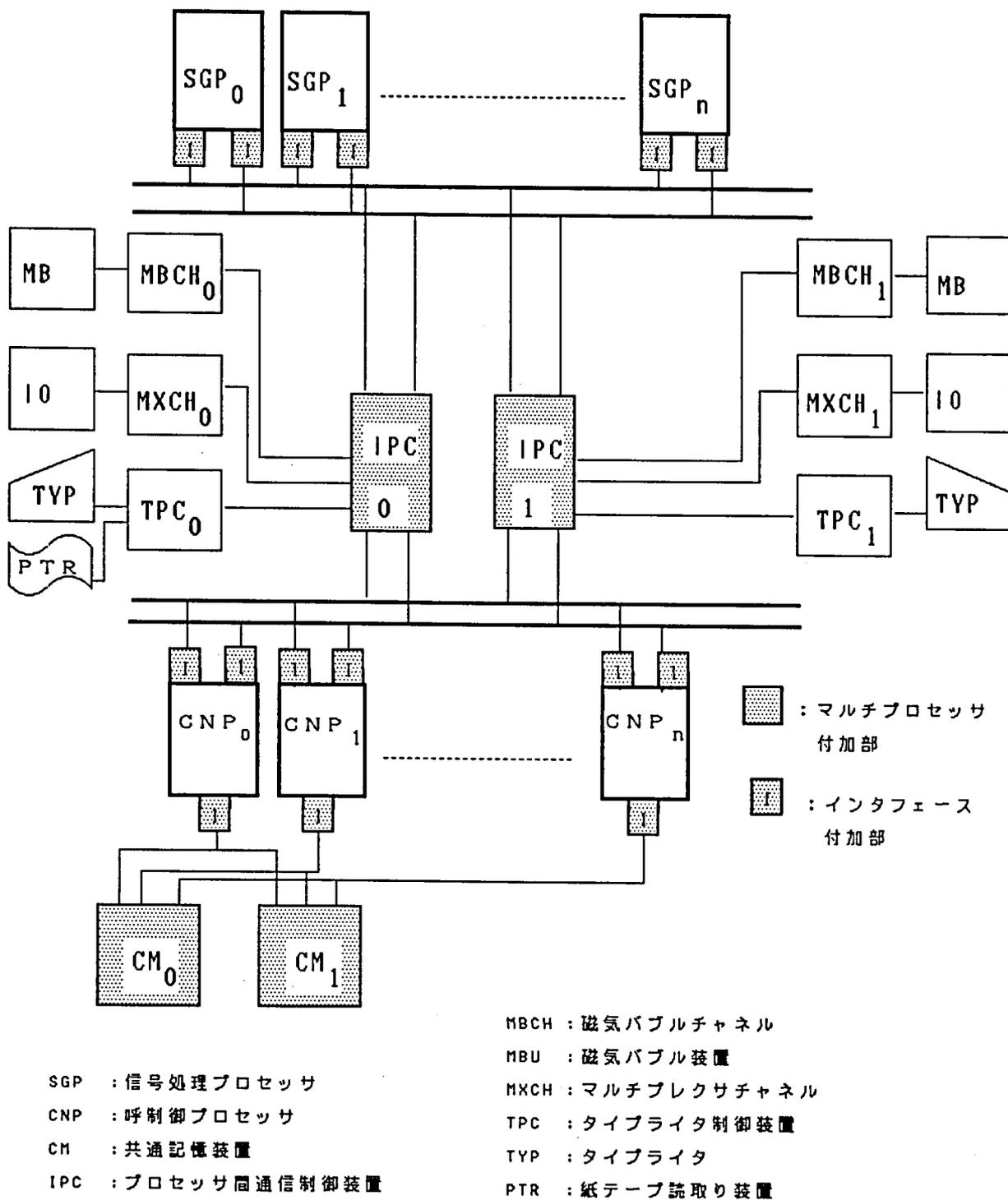


図4.23 DTS-1 マルチプロセッサの全体構成図

ある。ただし、これら付加部の機構により、システム全体に新たな設計を行わなくてもマルチプロセッサの実現を可能とし、ONE-MACHINE 概念を実現している。なお、高信頼化に向けた予備構成の議論は行なっていないが、図4.23は故障に備えた予備を含んでいる。この予備構成については次章で議論する。

#### 4.7 性能評価

ここで述べる実験機は、交換機においてマルチプロセッサ制御が成り立つかどうかを実証する段階のものであり、ここでは特に性能面からの評価を行なう。

##### 4.7.1 シングルプロセッサの平均命令実行時間

###### (1) 平均命令実行時間と交換規模

シングルプロセッサの性能を示す平均命令実行時間 $T_s$ は全命令の集合 $I_s$ の内第 $i$ 番目の命令 $I_i$ の実行時間 $t_i$ およびその出現頻度 $p_i$ を用いて以下のように表される。

$$T_s = \sum_{I_i \in I_s} t_i \cdot p_i \quad (4.4)$$

ただし、

$$\sum_{I_i \in I_s} p_i = 1 \quad (4.5)$$

また、プロセッサの処理能力を通路容量に換算した時、最大容量 $S_s$  (erl) は次のように表される。

$$S_s = h_0 (\rho_0 - c_0 \cdot T_s) / (x_0 \cdot T_s) \quad (4.6)$$

ここで、単位時間において呼数比例処理に与えられる割合は $(\rho_0 - c_0 \cdot T_s)$ であり、単位呼の処理が占める割合は $(x_0 \cdot T_s) / h_0$ である。

なお、各パラメータは次の量を示す。

$h_0$  : 呼の平均保留時間

$\rho_0$  : プロセッサの使用率の上限値

$c_0$  : 単位時間当りの固定処理処理ステップ数 (前章5.1.2における  $c_0$  に同じ)

$x_0$  : 呼当たりの処理ステップ (前章5.1.2における  $a_0$  と  $b_0$  の合計に等しい)

## (2) 改良 D 2 0 処理装置の命令実行モデル

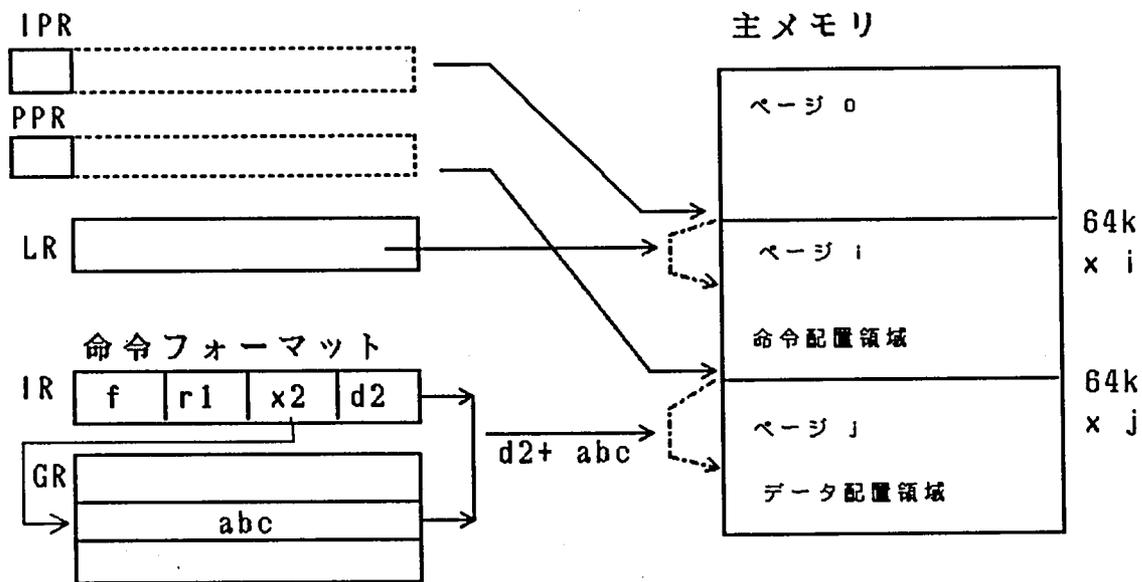
### ソフトウェアのアドレッシング (ページ方式とアドレッシング)

比較的小規模のプロセッサではアドレス空間の16ビット化などでメモリ量を減らすなどの経済化が図られており、アドレス空間を減らす手法としてページ方式が一般的である。一方、ページ方式をとるプロセッサに交換プログラムのような大規模なプログラムを適用すると、プログラムおよびデータが複数ページにまたがるという問題が生じる。この結果、プログラム実行時にジャンプ命令やデータによる異ページ参照が生じ、ページ切替えによって処理能力が低下する。

ここで、シングルプロセッサにおける交換処理サブプログラム間の結合はジャンプ命令を用いた物理アドレスによる直接結合を前提としている。なお、蓄積プログラム制御方式ではプログラムを記憶装置に固定的に格納し、これを繰り返し利用して交換機を制御するものであり、直接結合による効率化は当然である。また、交換処理プログラムでは小規模な変更は原則として行なわず、大きな変更は1年毎程度と頻度が少なく、影響範囲も大きいいため、変更時には全プログラムを入れ換える手法をとることが要因となっている。

### ハードウェア構成

改良 D 2 0 形処理装置のメモリアドレッシングは図4.24のように構成している。プログラムの命令部とデータ部を分離して、両者を別のページに配置する。すなわち、ページの先頭アドレスを示すプロセッサ内のページレジスタも命令アクセス用 (IPR) とオペランドアクセス用 (PPR) を別々に備えている。プログラムからの命令アドレス、オペランドアドレスの指定はページ内の相対アドレスとし、アドレス指定のコンパクト化を図りつつ、ページ切替えなしにアクセス可能とするメモリ空間の拡大を図っている。



IPR: 命令用ページレジスタ	f: 命令コード
PPR: オペランド ページレジスタ	r1: 第一オペランド
LR: ロケーションレジスタ	x2: インデックスレジスタ
IR: 命令格納レジスタ	d2: 第二オペランド
GR: 汎用レジスタ	

図4.24 基本命令のアドレッシング

#### 4.7.2 マルチプロセッサのプログラムの結合

ソフトウェアの基本構成要素として①シングルプロセッサで用いられたスケジューラ等のオペレーティングシステム(OS)と②マルチプロセッサで新たに必要となったプロセッサ間通信用プログラムおよび③これらのOS上で動作する呼処理プログラムがそれぞれのプロセッサの個別メモリに配置される。なお、故障等の異常時に複数プロセッサの系構成を統一的に管理、再構成するプログラムがマルチプロセッサに設けられるが、これは通常処理能力に影響しないため考察外とする。このプログラムを実行する権利はマスタプロセッサ(MCP)と呼ばれるCNPの内の1台だけが持ち、故障等の異常時のみに動作するものである。

OS上で動作する呼処理プログラムはリエントラント構成であり、複数の呼によって共有される。呼処理プログラムは①プログラム命令群(テンプレー

ト), ②交換機の設備等を表示する読みだし専用のデータ, ③リエントラント構造を保証する呼ごとのワークエリア, ④複数の呼の間でアクセスされ, 競合を起こすリソース管理データの構成要素からなる。このうち, シングルプロセッサでは一つのプロセッサに格納されていたテンプレートはSGPとCNPに分割され, 分散配置される。

このうち, 呼ごとのワークエリアは, ①ワークメモリ内でのみ操作されるローカル変数, ②呼管理データ(入力信号に対応した一連の処理完了後, 次の入力信号が到達するまでの呼処理結果の呼状態を保持する)からなる。ローカル変数は同一CNPで継続した処理の中で使用されるものであり, 共通メモリのアクセス頻度を減らすため個別メモリに配属される。呼管理データは, 他のCNPに切替えても矛盾を生じない呼状態を保持しており, 任意に負荷分配が可能なよう共通メモリに配置される。また, 継続した処理内ではSGP内の処理は他のプロセッサと無関係であり, SGPワークエリアはSGPの個別メモリに配置される。

リソース管理データは, 交換機リソースを一元的に管理するため, 共通メモリに配置される。このデータへアクセスするプログラムはCNPに配置される。なおSGPは接続された通話路に対応しており, 直接共通メモリにアクセスする必要はない。

#### 4.7.3 マルチプロセッサへの対応化

マルチプロセッサの性能上の問題点を明確にするため, シングルプロセッサのプログラムがマルチプロセッサではどのように変わるかを図4.25に示した。データへのアクセスはシングルプロセッサでは主メモリのみであったが, マルチプロセッサでは個別メモリ(シングルプロセッサの主メモリと同等)と共通メモリへのアクセスの二つがある。交換プログラムを構成するサブプログラム間の結合は, シングルプロセッサではジャンプ命令を用いて直接リンクしていたが, SGPとCNPに分散配置された場合, 両者はプロセッサ間通信を介した結合となる。また, シングルプロセッサでは主メモリ上のみにあったデータがマルチプロセッサのCNPでは主メモリ同等の個別メモリと共通メモリに分離される。

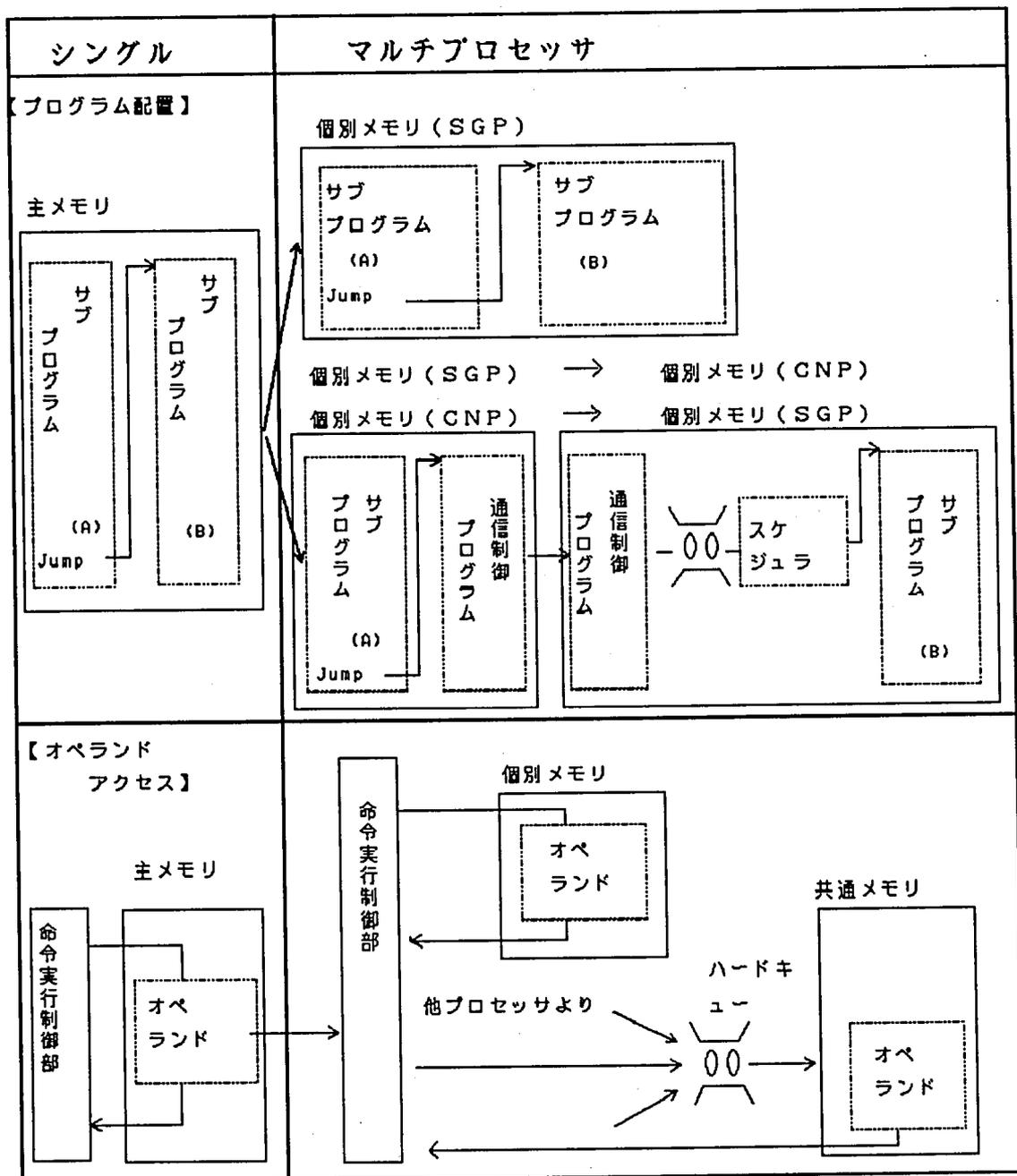


図4.25 シングル/マルチプロセッサのプログラム結合

#### 4.7.4 マルチプロセッサの平均命令実行時間

ここでは上記の改良D20形交換機用のプロセッサを用い、上記のソフトウェア条件に従ったマルチプロセッサの平均命令実行時間を評価する。

## (1) 機能面からの命令頻度分布変化

マルチプロセッサ化により新たに生じた命令頻度に関する条件には以下のように対処する。

プロセッサ間通信：全体を呼処理オーバヘッドとして扱い、通信機能増加分の命令分布の偏差は考慮しない。

Test and Set命令：共通メモリ内の論理競合を防止するため、データの使用中／未使用を矛盾なく識別しうる周知の Test and Set命令が共通メモリ結合のマルチプロセッサには必要である。これにより命令分布が変わる可能性があるが、生起頻度は0.1%のオーダーと微少であること、実行時間も他の既存命令と大きく変わらないことから、この命令による影響は無視する。

保守命令：マルチプロセッサの系構成を変更するための保守命令を新たに設けているが、これらは故障時など異常時に用いるものであり、通常時の呼処理には関係がないことからここでは省略する。

## (2) ページ切替え頻度の変化

命令アクセスとページ切り替え：シングルプロセッサの呼処理プログラムがマルチプロセッサではSGPとCNPに分割收容される。従って、マルチプロセッサの個々のプロセッサのプログラム規模は、シングルプロセッサに比較してむしろ減少している。シングルプロセッサ方式でページ内に入っておれば、マルチプロセッサでもページあふれをおこすことはない。

一方、マルチプロセッサのみに必要なプログラムもあり、一部追加されてプログラム規模が増大したものもある。通常の処理に関するプログラムの増大はTest and Set命令に関するもの及びプロセッサ間通信に関するものおよび後述の共通メモリオペランドアクセスに関するものである。これらの増大分は、上記のSGP、CNPへの分割により、それぞれが減少した分を越えるものではない。

従って、マルチプロセッサの命令アクセスによるページ切替え頻度は、シングルプロセッサに対して増大することはない。

オペランドアクセスとページ切り替え：改良D20処理装置を用いる交換機は中小規模であったが、マルチプロセッサ化した場合、対象とする交換規模は拡大する。この場合、前述の呼管理データやリソース管理データが増大し、シング

ルプロセッサでは1ページに入っていたものが複数ページにまたがる。オペランドアクセスのためのページレジスタの書替えが頻繁に生じる可能性がある。これによりマルチプロセッサでは大幅な処理能力低下の懸念があり、改良を要する。

### (3) 命令実行時間の改善

複数ページに展開した呼処理関連データアクセスのため、処理能力の低下の危険性があり、複数ページにまたがるオペランドアクセスのための効率の良い命令を追加しなければならない。このオペランドアクセスは共通メモリに対してなされるため、2語長で任意のページのデータに直接アクセスできる拡張形式をマルチプロセッサ用に設けた。

処理能力評価のため、この命令の効果を分析する。シングルプロセッサではすべてのデータアクセス命令が基本形式であったが、マルチプロセッサでは共通メモリにアクセスする場合には拡張形式に変わっている。両命令形式の命令実行時間を比較するため、Load命令の基本形式と拡張形式の差を比較した(図4.26)。Load命令以外にも拡張形式の命令を設けているが、保守命令を除けば、命令実行形態はメモリからレジスタへ転送するLoad命令に代表される。保守命令は通常は動作せず、処理能力評価で考慮する必要はない。

両形式の実行時間はそれぞれ各実行段階の時間の和で示される。拡張命令では命令形式が2語命令に拡大され、命令フェッチに要する時間が2倍に拡大される。なお、図4.26において、両命令形式とも命令実行の先行制御を行なうため、最後の実行段階(Arithmetic Operation)は次命令の命令フェッチ段階と重複する。従って、命令実行時間には現われず、命令実行時間の算出から除かれる。

ここで、マルチプロセッサ化により大きく命令実行時間が変化する可能性のあるCNPについて、平均命令実行時間 $T_m$ を算出する。CNPの平均命令実行時間 $T_m$ は、シングルプロセッサの平均命令実行時間 $T_s$ とこれに対する共通メモリアクセスによる変分を用いて表わされる。基本命令と拡張命令の実行時間差分には、図4.26の命令実行段階の所要時間を用い、CNPの平均命令分布のうちで拡張命令を利用する割合は $\sigma$ とすると、 $T_m$ は以下のように表される。

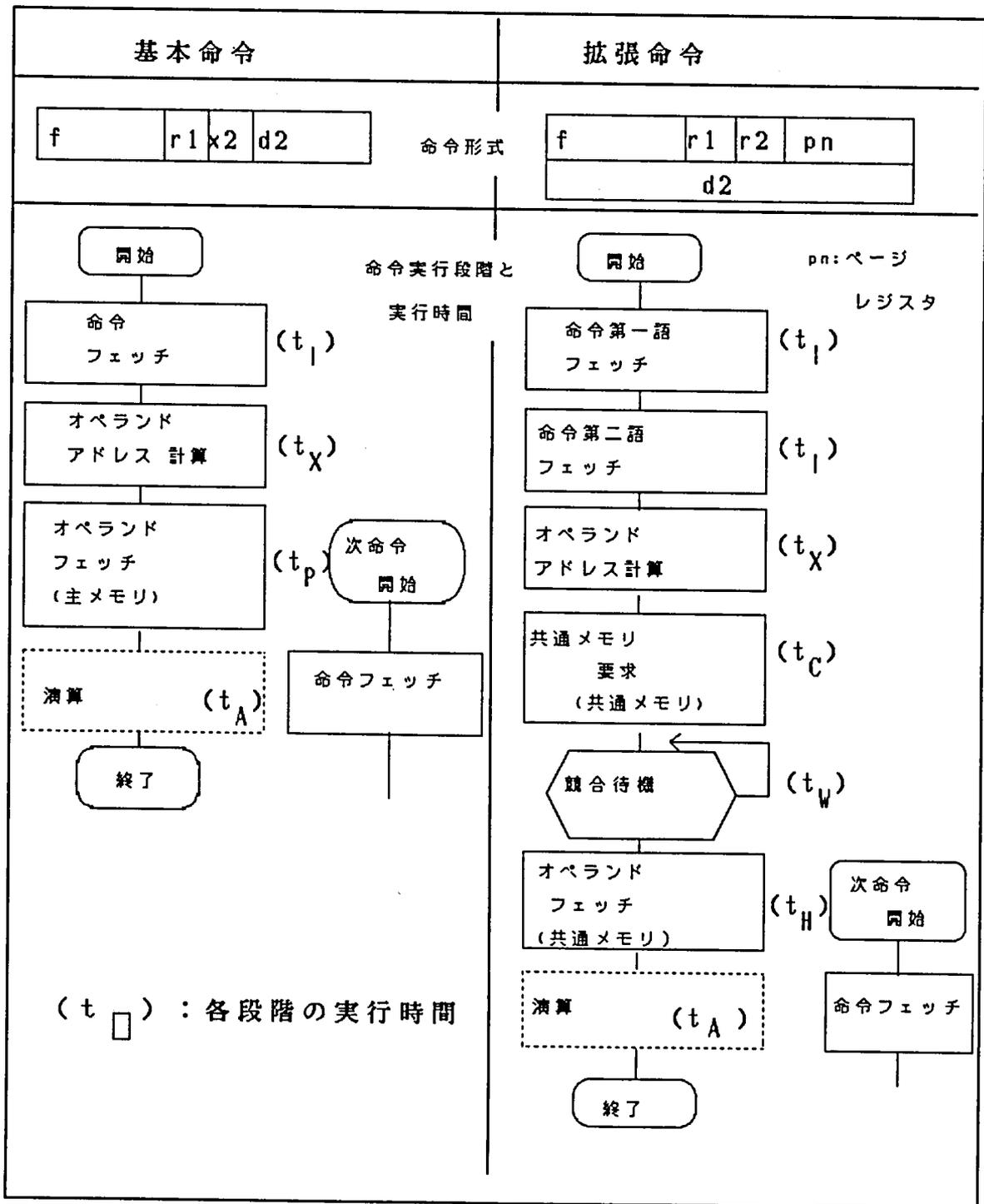


図4.26 Load命令における基本命令と拡張命令の比較

$$T_m = T_s + \sigma (t_1 + t_c - t_p + t_H + t_w) \quad (4.7)$$

ここで  $t_w$  は複数プロセッサからの共通メモリへのアクセス競合による待ち合わせ時間であり、固定的に決まるものではない。4.7.5の数値計算で具体的に示す。

いま、使用率  $\lambda$ ，サーバ1の一定保留時間モデル（いわゆるM/D/1モデル）の正規化待ち合わせ時間関数を  $W(\lambda)$  と表わすと、共通メモリアクセスに伴う平均待ち合わせ時間  $t_w$  は  $W(\lambda) \cdot t_H$  である。ただし、CNP実装数は  $N_c$ ，共通メモリアクセスのうちでプロセッサ間の競合を起こす保留時間は  $t_H$ ，共通メモリの使用率  $\lambda$  は  $N_c \cdot \sigma \cdot t_H / T_m$  となる。

ここで式の簡単化のため、 $t_H = t_1 + t_c - t_p + t_H$  とすれば、(4.7)は以下のようになる。

$$T_m = T_s \cdot \left( 1 + \sigma \left( \frac{t_H + W(N_c \cdot \sigma \cdot t_H / T_m) \cdot t_H}{T_s} \right) \right) \quad (4.8)$$

#### 4.7.5 システムとしての処理能力

ここでマルチプロセッサ制御方式により実現可能な交換機の最大規模を算出する。本マルチプロセッサによる交換機の最大規模は制御系の能力、通話路系の最大設備可能数によって規定される。なお、制御系ではSGPの能力、CNPの能力のいずれかが制限を与える。

まず、式(4.6)をマルチプロセッサ用に変形して各プロセッサが制御可能な通話路容量を求める。シングルプロセッサにおける呼当たりの処理ステップ  $x_0$  に相当するものをマルチプロセッサでは  $x_1$  とする。また、プロセッサ間通信に必要な処理量はIPCを中心に対称形であることからCNP、SGPで両者等量と近似し、送受あわせて  $2y_1$  としする。このとき、マルチプロセッサの呼当たりの総処理ステップは  $x_1 + 2y_1$  となる。また、マルチプロセッサの呼当たりの処理量  $x_1$  がCNP、SGPに  $k, 1-k$  の比で機能分割されたとすれば、CNP、SGPにおける呼当たりの処理ステップはそれぞれ  $\{x_1 \cdot k + y_1\}$ ， $\{x_1(1-k) + y_1\}$  と表される。従って、CNP、SGPの各1台が制御可能な通話路容量を  $S_{CNP}$ ， $S_{SGP}$  とすると、これらは式(4.6)から

$$S_{CNP} = \frac{h_0 (\rho_1 - c_1 \cdot T_n)}{T_n \cdot (x_1 \cdot k + y_1)} \quad (4.6')$$

$$S_{SGP} = \frac{h_0 \cdot (\rho_1 - c_1 \cdot T_n)}{T_n \cdot (x_1 \cdot (1-k) + y_1)} \quad (4.6'')$$

CNP, SGPの接続台数をそれぞれ  $N_o$ ,  $N_s$  とすると, マルチプロセッサ制御による交換機の最大規模  $S_n$  (erl) は以下のようなになる.

$$S_n = [N_o \cdot S_{CNP}, N_s \cdot [S_{SGP}, z]] \\ = \left[ \frac{N_o \cdot h_0 (\rho_1 - c_1 \cdot T_n)}{T_n \cdot (x_1 \cdot k + y_1)}, N_s \left[ \frac{h_0 \cdot (\rho_1 - c_1 \cdot T_n)}{T_n \cdot (x_1 \cdot (1-k) + y_1)}, z \right] \right] \quad (4.9)$$

ここで,  $[x, y]$  は  $x$  と  $y$  のうち小さい値を表し,  $\rho_1$  はマルチプロセッサにおける各プロセッサの使用率,  $z$  はSGPに接続された通話路のモジュール(ビルディングブロック化された通話路装置の分割単位)の容量である.

通話路モジュールの容量  $z$  と比較されるSGPの能力は通話路モジュールの容量  $z$  を処理するに必要な能力より十分大きく設定される. また, SGPの分担機能はハードウェアの制御が中心であるため, 将来も呼当たりの処理量が大きく増加しないとすれば, 式(4.9)の第2〔〕内では  $z$  が小である. 従って, 第1〔〕内ではCNPの能力  $N_o \cdot S_{CNP}$  と通話路容量の全体  $N_s \cdot z$  が比較される.

十分大きな通話路容量  $N_s \cdot z$  に対し, マルチプロセッサの能力を比較するとすれば, 結局, 交換機  $S_n$  はCNPの能力の最大値を求めることになる. 交換機の最大規模を求めるための以上の議論はマルチプロセッサの能力を求めるという当然の結果に帰結する.

従って, (4.9)は以下になる.

$$S_n = \frac{N_o \cdot h_0 (\rho_1 - c_1 \cdot T_n)}{T_n \cdot (x_1 \cdot k + y_1)} \quad (4.10)$$

ここでシングルプロセッサとマルチプロセッサの相対性能を求めるため, パラメータの整合を行なう.

CNPのプロセッサ使用率 マルチプロセッサにおけるCNPのプロセッサ使用率を  $\rho_1 = \rho_0$  とシングルプロセッサと同じに設定する。これは、マルチプロセッサの性能評価上、厳しい設定であり、安全側となっている。本来、同一平均待ち時間となるサーバの使用率は、複数サーバでは単一サーバより大きくとれ、処理を依頼するSGPから見たときCNPは複数サーバとして働く。複数サーバのCNPの使用率をシングルプロセッサと同一にしたことは安全側となっている。

固定処理の比率 オペレーティングシステムなどの固定処理量の比率をマルチプロセッサとシングルプロセッサと同じとして  $c_1 = c_0$  と仮定する。これもマルチプロセッサには厳しい設定、安全側評価となる。本来、固定的な処理はすべてのCNPには必要はないものである。また、固定処理の平均命令実行時間はシングルプロセッサと同じく  $T_s$  とする。固定処理は各プロセッサに閉じた処理であり、共通メモリにアクセスする必要はないことによる。

呼当りの処理量 呼当りの処理量もマルチプロセッサとシングルプロセッサで同じとして  $x_1 = x_0$  とおく。呼当り処理の大きな差分はプロセッサ間通信のサーバヘッドで吸収されたとする考え方である。

以上の前提を設けると(4.10)は以下になる。

$$S_m = N_0 \cdot h_0 (\rho_0 - c_0 \cdot T_s) / (T_m \cdot (x_0 \cdot k + y_1)) \quad (4.10')$$

ここで、マルチプロセッサの性能をシングルプロセッサに対する相対性能値すなわち等価プロセッサ台数で表わす。シングルプロセッサに対する交換機実現規模  $S_s$  は式(4.6)で求められており、等価プロセッサ台数は(4.10')と(4.6)の比となり、以下のようなになる。

$$\begin{aligned} S_m/S_s &= \frac{N_0 \cdot h_0 (\rho_0 - c_0 \cdot T_s) / (T_m \cdot (x_0 \cdot k + y_1))}{h_0 (\rho_0 - c_0 \cdot T_s) / x_0 / T_s} \\ &= N_0 \cdot / ((k + y_1/x_0) \cdot (T_m/T_s)) \\ &= \frac{N_0}{(k + y_1/x_0) \{ (1 + \sigma \cdot (t_H/T_s + W(N_0 \cdot \sigma \cdot t_H/T_m) \cdot t_H/T_s)) \}} \quad (4.11) \end{aligned}$$

#### 4.7.6 処理能力の推定

実験機で得た値からマルチプロセッサの性能すなわち等価プロセッサ台数を求める。概略式(4.11)に実験機で得た値を以下のように適用する。

##### (1) ソフトウェア実現結果

一次評価で得た概略値から以下の値を用いる。

- ・ SGPとCNPへの機能分割の程度： $k = 0.5,$
- ・ 通信のオーバヘッド： $y_1/x_0 = 0.1$
- ・ 拡張命令の発生頻度： $\sigma = 0.1$

ただしソフトウェアに関する条件の規定には困難な問題を含む。これは大きくは以下の理由に基づく。すなわち、局種、局条件などの違いがあり、呼のモデルの確定化が困難である。また、サービスの実現程度や機能追加など可変的要素を含む。さらに、ソフトウェアには膨大なデータによる集計の困難性がある。定量評価では特に危険要素となる共通メモリアクセス率には幅を持たせて評価することが必要である。

##### (2) 改良D20中央処理装置のハードウェア実現値

シングルプロセッサ段階でのメモリアクセス時間は平均命令実行時間に対する相対値として  $t_1/T_s = t_p/T_s = 0.3$  である。

##### (3) 共通メモリアクセスに関する諸数値

共通メモリアクセスに関する諸数値は図4.26の拡張命令実行時間の各段階の実行時間で表わし、共通メモリアクセスの中で待ち合わせを除くアクセス時間は  $(t_c + t_H)/T_s = 0.6$  である。なお、複数プロセッサで競合を生じる共通メモリの単位使用時間(サイクルタイム)は  $t_H/T_s = 0.45$  と高速に実現している。これは、前節に示したように共通メモリの接続法には性能向上に有利な個別リード接続法を採用した結果である。なお、もしバス接続法を採用した場合は  $(t_c + t_H)/T_s = 1.7, t_H/T_s = 0.8$  と推定され、能力低下は避けられない。

なお、 $t_H = t_I + t_c - t_p + t_H$  であり、式(4.11)における  $t_H/T_s = 0.6$  は以上の結果から明かであるが、待ち合わせ時間比率  $W(N_0 \cdot \sigma \cdot t_H/T_s)$  は得られていない。なお  $W()$  は前述のように、()内にマルチプロセッサの平均命令実行時間を含むこと

から単純には計算できない。このため、一次近似として  $T_m = T_s$  から出発し、得られた  $T_m$  を繰り返し(4.11)に代入して収束値  $T_m$  を数値計算する。

以上の計算法により求めたマルチプロセッサの最大性能すなわち等価プロセッサ台数を図4.27に示す。CNPの最大実装台数は8であり、共通メモリアクセスによる処理能力飽和はほとんどない。プロセッサ間通信のオーバーヘッド、共通メモリアクセスによる命令実行時間の増大が相対能力低下を招いている。

ここでは、機能分散比率を 0.5，すなわち，SGPの常用側の台数もCNPと同じとしている。従って、最大16台のプロセッサに対し、12台の等価プロセッサ能力を実現している。

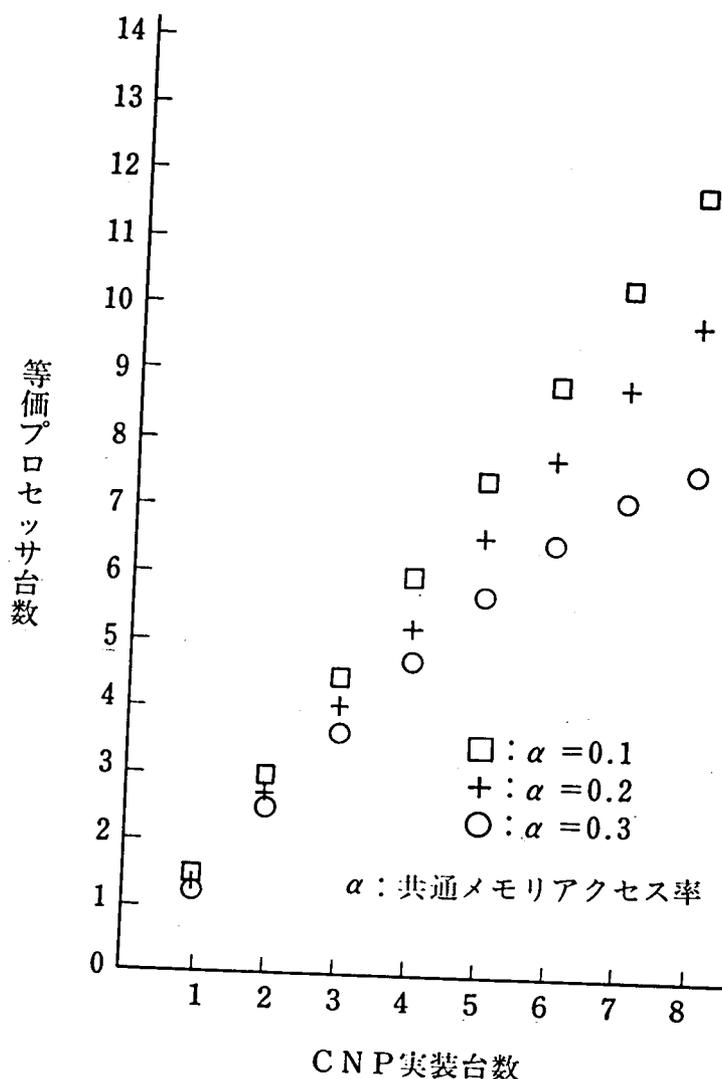


図4.27 マルチプロセッサの処理能力

## 5. マルチプロセッサの設計事例 II

### 32ビットカスタムVLSIプロセッサの実現とマルチプロセッサへの適用

ここまで述べてきたマルチプロセッサは実験で確認され、その後32ビットカスタムVLSIプロセッサの適用による性能や経済性や信頼性等の改良を経て商用実用化された。本節では、このマルチプロセッサの改良において重要な位置を占める32ビットカスタムVLSIプロセッサの実現に向けての設計方針および設計経過とデジタル交換機制御用マルチプロセッサへの適用について述べる。

#### 5.1 交換処理用カスタムVLSIプロセッサの開発<sup>[66]</sup>

##### 5.1.1 汎用マイクロプロセッサとカスタムVLSIプロセッサの比較

VLSI技術の進展に支えられて、汎用マイクロプロセッサは進展を続けている。高性能化や高機能の具備へと技術進展は続き、その普及にはめざましいものがある。従来カスタムプロセッサが用いられてきた分野においても、このような状況から汎用プロセッサを用いるか、カスタムプロセッサを開発するかが大きな選択枝となってきている。汎用プロセッサにもカスタムプロセッサにもそれぞれ長所や短所があるが、ここでは交換機へ適用するプロセッサの考え方を整理するため両者の特徴を分析する。

##### 汎用マイクロプロセッサの特徴

汎用マイクロプロセッサは各種用途に利用されるプロセッサをLSI部品として共通に供給するものである。適用用途が広いため、利用数が多い。また、LSIの量産性からプロセッサのチップのみでは経済性が高いという特徴がある。

しかし、幅広い用途の共通の要求条件を仕様としており、アーキテクチャは必然的に焦点の定まらないものとなる。個々の応用システムに着目したとき、必ずしも効率のよいものではない。

##### カスタムプロセッサの特徴

カスタムプロセッサは応用システムに適合化して実現するものであり、LSIとしての量産性はなく、その開発費も応用システムでまかなうことから、汎用マイクロプロセッサよりコストは高い。しかし、命令仕様等のアーキテクチャやシステム化する場合のハードウェア構成は、応用システムに最適化するものであ

り、効率は良い。

従来、交換機制御システムのような特殊な情報処理システムには、適用可能な汎用プロセッサが無く、専用プロセッサを開発して高性能化や経済化や高信頼化を図ることが常識であった。しかし、近年の汎用マイクロプロセッサの進展からマイクロプロセッサの部品としての利用が一般化し、プロセッサチップの廉価性に注目した適用可能性の議論が行なわれるようになった。また、ソフトウェア産業、システムハウス産業の台頭により、プロセッサを部品として利用する技術も定着を始め、この議論に拍車がかかりつつある。

汎用プロセッサの交換機への適用についてはもはや可能／不可能を議論する時代ではなく、それが交換機としての商品戦略に有効かの議論となる。現実にITT-1240のように、全面的に汎用マイクロプロセッサを商用交換機に適用した例もある。ただし、現在量産機と見なしているものの中には、全面的に汎用マイクロプロセッサを採用した物がないのも事実である。

#### 5.1.2 カスタムVLSIプロセッサの有効性

汎用マイクロプロセッサとの対比の下で交換機へのカスタムプロセッサの適用の有効性を確認する。カスタムプロセッサの有効性は以下の点にあると考えられる。

##### (1) カスタムプロセッサの効率性

###### (a) 効率化要因

命令の効率化 交換機制御用プロセッサは電話機や回線や信号装置など多数の端末を制御する。すでに述べたが、交換機制御用プロセッサには多数の端末の状態を主メモリ上に写像して制御する。通話路装置のハードウェア状態を示す膨大な未加工の情報から有効な情報を抽出して処理することが必要である。このためハードウェアの状態をメモリへ写像するなどの交換処理に適した命令が存在する。例えば、回線の状態を監視して変化を検出する群処理命令である。これはハードウェアの制御対象点のon/offをメモリの1ビットに対応させ、複数の制御点を1語にまとめて処理する命令である。具体的には、発呼や終話などを監視するSUP命令、信号が変化した点を数字化して識別できるFRM命令があ

る(67)(68)。 プロセッサの1命令で1ビットのみを処理するのでは効率が悪いが、複数ビットを1命令でまとめて処理すると効率が向上する。このような命令は汎用処理には不要であるが、カスタムプロセッサでは応用システムに合わせて効率のよい命令が実現出来る。

ハードウェアの効率化 24時間運転を前提とする交換機には、故障に備えて予備を具備し、故障時の予備への自動切り替えや起動と停止のための付加回路を多数必要とする。また交換機制御用のプロセッサ自身にも高信頼化へ向けた予備装置間の交差を制御する回路や実時間制御へ向けた時計管理や通話路を制御する回路が存在する。これらのシステム毎の専用回路を、カスタム回路で実現し、プロセッサVLSIチップに組み込むことにより、ハードウェアの大幅な部品点数削減が図れる。

#### (b)カスタムプロセッサの効率性

上記の(a)の議論を、ここでマクロに定量化して示す。カスタムプロセッサでは命令の効率化とハードウェア構成の効率化が総合的に性能と経済性の向上を果たす。今、x(erl)の局を性能pのプロセッサによるマルチプロセッサで実現するとき、そのコストyは以下のように表わせよう。

$$y = (a+b) * \lceil cx/p \rceil \quad (4.12)$$

ただし、

a: プロセッサの基本回路のコスト

b: 応用別回路のコスト

c: 単位 erlの局の交換に要するプロセッサ性能。

交換機制御用プロセッサにはプロセッサの核となる部分(コスト:a)の他に、実時間制御、高信頼化などのための応用別回路(コスト:b)が存在する。応用別回路を専用のVLSIとして実現する対案には、プロセッサの核部分に汎用マイクロプロセッサチップを用いる案も存在し、これを半カスタムプロセッサと呼ぶ。半カスタムプロセッサのハードウェア規模やコストは、全カスタムプロセッサと大きくは変わらないが、性能は汎用マイクロプロセッサに相当する。想定される概略値によって対案を比較したものを図4.28に示す。

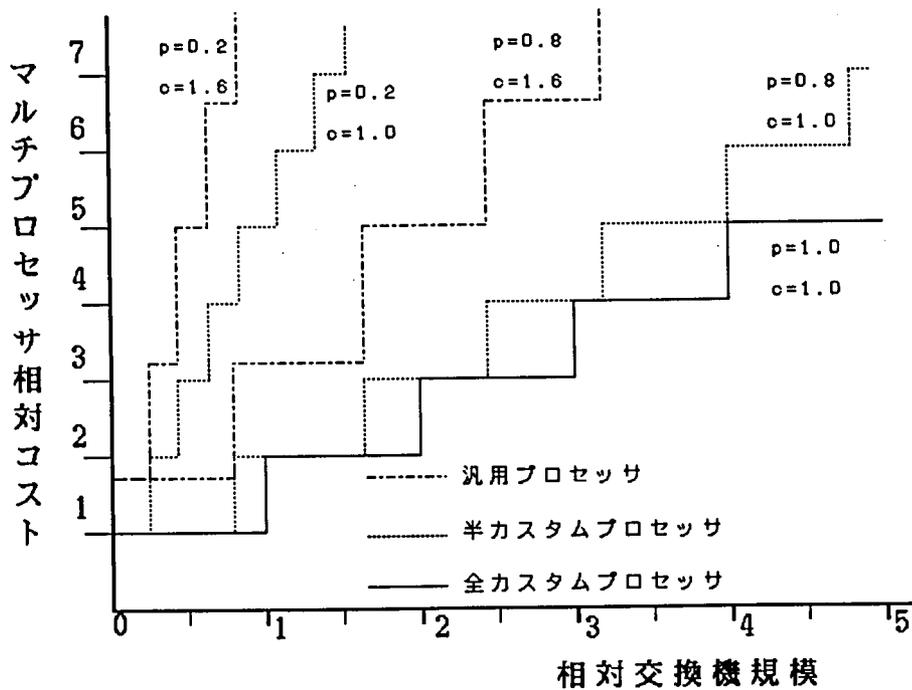


図4.28 プロセッサ種別とコスト傾向

応用システムに対して最適化したカスタムプロセッサでは、局規模の増大と共に経済性が向上する。ここでカスタムプロセッサに対して汎用マイクロプロセッサの性能比は0.2程度と小さい値になっているが、これは後述するように両者の採用部品技術の差による。カスタムプロセッサでは最先端部品技術を活用し、他に先駆けて高性能なプロセッサを実現できることから生じる差である。例えば、ここで述べる32ビットカスタムVLSIプロセッサが既の実現されたとき、汎用マイクロプロセッサの市場ではようやく16ビットプロセッサが普及を始めていた段階であった。汎用マイクロプロセッサの低能力は、これを考慮して設定したものである。

## (2) 既存ソフトウェア資産の継承

電子交換機に関連して莫大なソフトウェア資産が蓄積されており、この資産を活用できることが必要である。既に存在する交換機仕様に適合したカスタムプロセッサのアーキテクチャを継承して新プロセッサを開発することがソフトウェアの継承性を保証するに、最も簡便で、かつ効果のある方法である。

### (3) 部品技術先導性

市販汎用プロセッサの部品技術に対する戦略は、最先端部品の採用ではなく、短周期の成熟部品への乗り換えである。部品技術が成熟するごとに新しい商品を開発し、商品サイクルを短くして部品技術追随性を保証する。

一方、長期方式寿命の通信システムでは、これと異なった部品採用戦略が必要である。システムの性能向上、経済性の達成には最新の部品技術を用いることが最も有効な方法であるが、一度採用を決めた部品は比較的長い間変更せずに使用するのが通信システムの特徴である。このため、出来る限り先を見て先端部品を採用することがシステムの陳腐化を避けるに有効である。大規模システムの長い開発期間を考慮すると、製品化時期に最先端となる部品を採用できるよう計画を組む必要があり、自らの手の内で最新部品技術を選定することが可能なカスタムプロセッサは通信システム向きに有効と言える。

### (4) 管理の自主性

#### 仕様管理の自主性

市販プロセッサにおいては、その仕様記述の中でユーザに無断で仕様変更を行なうと宣言することが常識である。汎用マイクロプロセッサでは、ユーザへの影響を無視して供給側の都合で仕様変更が行なわれる危険がある。

一方、交換機ではプロセッサの仕様変更にきわめて臆病になる必要がある。交換機は24時間無休止システムであり、動作中にプログラム変更やハードウェア置き換えを行なうことは容易ではない。また、仕様変更を行なう場合には導入されたシステム全体にほぼ同時に行なわなければならない。交換機は同一仕様でかつ同一の管理の下で、全国的に大量に運用されており、導入数の量的な問題からも仕様変更を簡単に行なうことは出来ない。

仕様管理の観点からは、仕様の自主管理ができるカスタムプロセッサは交換用プロセッサとして望ましいと考えられる。

#### 部品供給の自主管理

汎用部品には販売量が少なくなると供給側の判断により生産中止になるという問題がある。一般に部品生産中止の場合、小規模な汎用部品では通常それなりに互換性のある代替部品が存在する。しかし、プロセッサはきわめて大きな論

理規模であり、またソフトウェアを含めて影響範囲が大きく、単純に代替部品への移行は出来ないという問題がある。

また、交換機には一般消費物品や汎用商品に比較して方式寿命が長いという特殊性がある。その間、保守部品等を含め、予備部品の確保が必要である。この点を考慮すると、プロセッサには部品供給の自主管理が可能なカスタム化が有効と考えられる。

### 5.1.3 関連分野のプロセッサの応用とカスタムプロセッサの経済性<sup>[69]</sup>

#### (1) 交換機関連分野のプロセッサ応用

カスタムプロセッサは交換機に有効とは言え、カスタムプロセッサ自身も量産性による開発投資分担効果を期待される。開発費を多数の応用分野で分担すればシステムの経済性はさらに向上する。製品サイクルの適合化、関連する応用間での利用可能性を高めて適用範囲を多くし、カスタムプロセッサの利用拡大を図ることが必要である。

最近の交換機では制御装置に多くのプロセッサを利用するようになってきている。プロセッサが高価な時代には布線論理で実現せざるをえなかった小さい論理規模に対しても、近年のマイクロプロセッサ技術の進展から融通性に富むプログラム制御の適用が可能となってきた。しかもプログラム制御の融通性や高機能化により省力化や人間操作の支援、高信頼化を図ることが可能となってきている。

交換機に関連するプロセッサの利用分野を図4.29に示す。ここでは、既述の交換機制御用プロセッサの他に付帯系制御プロセッサの分類を設けている。なお、前者を後者と区別し、主制御プロセッサと呼ぶ。

付帯系制御プロセッサはマンマシンインタフェースの高度化に適用され、交換機開発用、保守運用用に用いられるいわば交換機用ワークステーションである。

交換機本体とは独立に改良され、寿命も短くてよい。付帯系制御プロセッサには市販のマイクロプロセッサでも十分であり、主制御用の高性能なカスタムプロセッサは経済性を除けば、当然適用条件を満足する。

— 交換機ソフトウェア支援系 —

— 交換機本体系 —

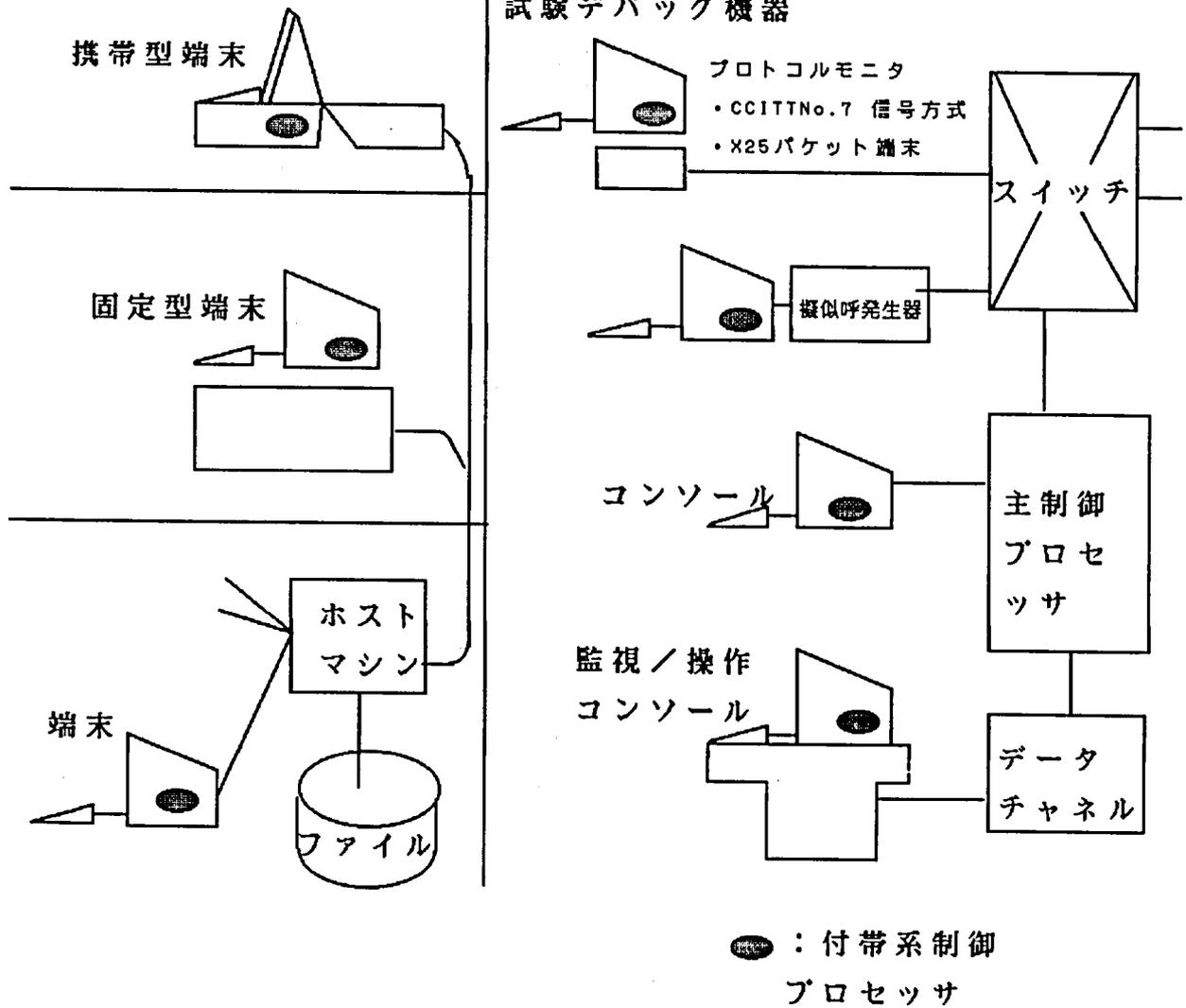


図4.29 交換機周辺のプロセッサ利用

## (2) 付帯系におけるカスタムプロセッサのコスト

プロセッサチップのコスト要因は開発費等の生産性によらない固定コストと各々の生産に要するランニングコスト分があり、一次近似としては以下のように表わせる。

$$\text{カスタムプロセッサコスト} = \text{ランニングコスト} + \text{開発コスト} / \text{導入数}$$

$$\text{汎用プロセッサコスト} = \text{ランニングコスト}$$

汎用プロセッサは生産量が十分に大きいため、各プロセッサで分担する開発等の固定コストは無視でき、製造等に要するランニングコストのみとなる。従って、カスタムプロセッサは導入数が少ないと高コストになる。

ここで、交換機のすべてにカスタムプロセッサを導入する場合のプロセッサ総コストを $y_0$ 、付帯領域に市販汎用プロセッサを導入する場合のプロセッサ総コストを $y_1$ とすると、両者はそれぞれ以下のようなになる。

$$y_0 = n \cdot (k_0 + k_1) \cdot \{h_0/n / (k_0 + k_1) + z_0\} \quad (4.13)$$

$$y_1 = n \cdot k_0 \cdot (h_0/n / k_0 + z_0) + n \cdot k_1 \cdot z_1 \quad (4.14)$$

$$y_0 - y_1 = n \cdot k_1 \cdot z_0 - n \cdot k_1 \cdot z_1 \quad (4.15)$$

$$= n \cdot k_1 \cdot (z_0 - z_1)$$

ただし、

$h_0$  : カスタムプロセッサの固定コスト、

$z_0$  : カスタムプロセッサのランニングコスト

$z_1$  : 汎用マイクロプロセッサのランニングコスト

$k_0$  : 各交換機に利用される主プロセッサ台数

$k_1$  : 各交換機に利用される付帯制御プロセッサ台数

$n$  : 交換機の総導入数

両者のコスト差の要因は(4.14)からランニングコストのみであることがわかる。カスタムプロセッサにおいても、通常商用導入が進んだ段階では部品技術は十分成熟していると考えられ、この差は大きくはない。さらに、同一の交換機に異種プロセッサアーキテクチャ、さらにソフトウェアの系列を持ち込むことは望ましくない。経済性において差が無ければ、主制御用として効率の良いカスタムVLSIプロセッサを開発し、これを付帯制御プロセッサにも適用し、ソフトウ

エア環境を統一することが望まれる。

#### 5.1.4 32ビットカスタムVLSIプロセッサの開発目的<sup>[66][70][71]</sup>

上記の議論からは、依然として交換機関連にはカスタムプロセッサの開発が有効であり、交換機に関連した幅広いプロセッサ適用用途に共通に利用できるカスタムプロセッサを実現することが望まれる。このため、最新部品技術を活用し、新たにデジタル交換機制御用のVLSIプロセッサを開発する。その目的は以下の通りである。

##### (1) 大規模LSI技術の活用

既に述べたように、交換用プロセッサには命令実行機能に加えて、高信頼化や実時間制御のための各種付加回路が存在する。カスタムVLSIプロセッサでは、これらの付加回路を極力VLSIチップ中に取り込み、プロセッサの小型化や経済化やコストパフォーマンスの向上を果たす。

##### (2) CMOS技術の活用

デジタル交換機は電子部品のみで実現可能であり、摩耗のある電磁部品を皆無にできる可能性がある。交換機から摩耗部品を排除することは、保守を簡単化して交換機を経済化でき、有効な考え方である。プロセッサにおける摩耗部品は冷却用のファンであり、ファンを排除して自然空冷を採用することは大きな方針となりうる。プロセッサの発熱量を極力削減することが求められる。CMOS部品技術は低消費電力、高集積の特徴があり、CMOS技術を用いてカスタムVLSIプロセッサを実現することが有効となる。

なお、CMOS技術では既存のECL素子を用いた交換機制御用プロセッサより処理能力は低下する。しかし高集積部品の利用による単体プロセッサの大幅な経済化を達成し、単体プロセッサの処理能力の不足にはマルチプロセッサ化により処理能力拡大を図ることが大きな目的である。

### (3) 32ビットアーキテクチャの実現

交換機には今後のISDNサービスの実現に向けて、ソフトウェアのさらなる大規模化が予想される。ソフトウェアの大規模化に対し、アドレス空間が狭小では大きな足かせとなる。ソフトウェアの簡明化に向けてアドレス空間の拡張や、さらにアドレス空間拡張による処理能力拡大が有効となる。この観点からアドレス空間を拡張した32ビットVLSIプロセッサが有効である。

## 5.2 VLSIプロセッサ実現に向けた課題と設計方針

現在では汎用マイクロプロセッサは商用チップとして多種存在し、標準化論争が必要なほど大衆化した技術である。ただし、ここで述べるカスタムVLSIプロセッサの開発はようやく8ビット汎用マイクロプロセッサが普及を始めた時期に始まったものである。マイクロプロセッサ技術はまだ未成熟で、32ビットマイクロプロセッサは研究として進めなければならなかった。32ビットVLSIプロセッサを実現していくにあたり、想定した大きな問題と解決の方向付けを以下に示す。

### (1) 高精度な大規模回路の実現

カスタムVLSIプロセッサの回路規模は、既にSSIやMSIや、一部にLSIを用いた既存のプロセッサ開発で経験している。しかもこのような大規模回路を実現したとき、ハードウェアとして実現した段階では、論理バグの残存は避けられないのが常識的な認識である。従来のSSIやMSIを主流部品としている装置では、ハードウェア化しても論理バグはプリント配線板の布線の張り替え等で修正可能であったし、このため望ましくはないが、ある程度バグの存在に対する覚悟は出来ていた。

VLSI装置の設計と従来装置の設計の違いは、実現すべき回路の精度にあると考えなければならない。VLSI装置においては論理バグの修正はチップ内回路の修正を必要とし、チップの再製造のため数カ月の修正期間を要するという問題がある。VLSI装置においてはバグの存在を前提に設計を行なうことは出来ない。

VLSIプロセッサでは回路の高精度化が必要であることから、実現すべき回

路をモジュール化するとともに、それぞれの機能の単純化を図ることを方針とする。この観点から必要な機能を以下の3種のチップに分けて実現する。

#### プロセッサ核チップ (CCAチップ)

交換用に適した命令を含む命令実行機能等のプロセッサ基本機能を具備する。このチップは主制御プロセッサと付帯系プロセッサに共用する。

#### 主プロセッサ周辺チップ (CCBチップ)

主制御プロセッサをコンパクトに実現するため、高信頼化、実時間制御等の主制御プロセッサに必要な付加機能群をオンチップ化する。

#### DCHチップ (DCH)

交換機制御用にもファイルやプロセッサ間通信や共通線信号転送制御や入出力装置など周辺装置が必要である。プロセッサとこれらの周辺装置との間の情報転送制御を司るデータチャネルは回路規模が大きく、導入数もプロセッサより多い。このため、データチャネルの機能をVLSIで実現する。

### (2)集積能力の制限

VLSIには、微細加工精度やチップサイズから生じる集積能力の制限がある。チップに何を載せ、チップの搭載能力をどのように有効に利用するかは大きな課題である。ただし、今回のVLSIではモジュール化して上記の3種のチップに分離して実現したため、部品の集積能力には、むしろ余裕があり、特別な対策は不要であった。

### (3)回路バグ修正困難性に対する対策

#### (i)プロセッサの残存バグの特徴

24時間サービスを行い、高信頼性が要求される交換機にとって、残存バグへの対策を考慮しておくことは必要である。設計、デバッグにはバグの除去に十分留意しても大規模回路を実現する人間作業にバグが皆無という保証はない。

ただし、長期の電子交換機の運用経験からは、運用中のシステムでプロセッサのバグが見つかることはない。交換機におけるプロセッサは、システムの運用開始前に呼処理プログラムによるいわば過酷な試験を行っており、運用開始後プロセッサのバグが直接問題を起こすことはない。他のシステム開発に関連し

て、新しいシステムやソフトウェアを実現し、新しい条件を生じたときにバグが顕在化する。

#### (ii) プロセッサのバグ処置に対する考え方

今までの経験では、プロセッサのバグが直接運用システムに影響を与えることはなかったが、バグの残存の可能性がある以上、バグ対策は十分考えておかなければならない。プロセッサのバグ対策は以下の2フェーズに分けられる。その第1フェーズとは、運用中にバグが顕在化した瞬間の対策である。バグが顕在化しても、それが原因となって直接システム停止に至らないようにするものである。第2のフェーズはシステムからバグ自身を除くものである。

第1フェーズについては、初期設定で対処が可能と考えられる。運用開始からバグが顕在化するまで、長期間バグが潜在化しており、バグの顕在化にはなんらかの低頻度の要因が作用しているものと考えられる。システム初期設定を行えば低頻度のバグ顕在化の要因は除かれ、その要因は一時的にしても除かれる。

一方、一時的にバグの影響を除けたとしても、第2フェーズの対策としてのバグの除去は速やかに行なわなければならない。もし、運用中のシステムにバグが影響を与えることがあったとしたら、バグ要因を除かないかぎり間欠的にバグが顕在化し、その度に初期設定が行なわれることになる。このため、たとえ最終的にVLSIを修正するとしても、VLSIの中をさわらず、バグを除きうる必要がある。

#### (iii) VLSIプロセッサのバグ対策

このため、既存プロセッサの開発時のデータからバグの種類を分析し、バグ対策として、VLSIの中を修正しなくても仮処置ができる対処法を検討し、以下の結論を得た。なお、この詳細は、5.4.2 デバッグ手順の項で詳述する。

(a) 従来のプロセッサ開発の手順により回路設計、デバッグを行えばほぼ満足のいく論理回路の精度が得られる。

(b) 実現予定の3種類のVLSIチップでプロセッサを実現すれば、たとえバグが残存していてもかなりのバグがVLSIの外で対処出来る。

(c) 仕様に影響するバグについても、マイクロプログラム制御メモリをVLSIの

外におけばマイクロプログラムの修正で対処ができる。

#### (4)入出力信号数の制限

チップの入出力ピンは駆動電力の大きいインタフェースとなり、チップサイズやチップ消費電力の大きな部分を占める。このため、LSIを実現する上では極力入出力ピン数が少ないことが望まれる。想定するCMOS技術ではVLSIチップの入出力信号数は200以下と制限を受けている。

一方、交換機制御プロセッサには、この要求条件に反して多数の外部制御インタフェースがある。通話路制御用や入出力装置等の外部装置制御用インタフェースがある。また、実時間で制御する上で、極力多量の情報を一度に伝達できるようにするために各インタフェースの語幅は大きい。外部装置インタフェースをそのまま収容するのではピン数制限を超過する。また、1回当りの制御信号に対する保留時間も数 $\mu$ 秒と一般命令の1 $\mu$ 秒程度の命令実行時間に比較して長く、VLSIの観点からは従来の外部インタフェースは効率が悪い。

このため、VLSI向きの新しいインタフェースを規定する。これにはVLSIによる装置の高密度実装と信号伝搬遅延の減少化のインパクトを利用することが必要である。以下のように極力、信号の集約化を図るとともに情報転送の効率化をはかっている。

(i)共通母線（バス）化 まず、主記憶、通話路制御、ファイル等との情報転送を共通母線（バス）構成に統一する。これにより、インタフェース線数は1/3に削減できる。しかし、単純にバス形式にまとめるだけでは、従来並列にアクセスできた外部制御処理が直列アクセスとなり、性能低下を招く。

(ii)制御送信と応答受信の独立化 外部装置制御では制御オーダの送信とそれに対する応答の受信が対となっているが、保留時間が長いと制御オーダ送信と応答受信を分離してバスを利用する。このようにすると、従来インタフェースを保留し続けたものが送受に独立化でき、他の信号がこの間に割り込め、転送効率が向上する。この方式を実現するには、どの装置に応答を返すべきかを識別可能とする必要がある。制御指令と同時に発信元と送信先を相手に知らせなければならない。すなわち、指令送信側は制御オーダとともに送信元識別記号も送らなければならない。



規定したプロセッサバスプロトコルを図4.30に示す。入出力装置アクセスやプロセッサ間通信にはすべての情報転送に対し、送信元と送信先を付与する。また、送信元や受信先がプロセッサであっても同じ手段を利用する。マルチプロセッサでは各プロセッサが送信元、送信先になりうる。これにより各プロセッサはプラグイン化が可能となり、ビルディングブロック形によるマルチプロセッサが容易に実現できる。なお、メモリアクセスは保留時間が短く、送信と受信とを分離する必要はないこと、またアドレス空間を極力拡張する意味から送信元アドレスは省略してアドレスビットに利用する。

#### (5)チップ出入りの信号遅延

VLSI設計ではチップ内で閉じる論理遅延に比較し、チップ内外に出入りする信号は大幅に遅延する。この対策としては、個々の設計でクリティカルバスを含む回路は同一チップへ一括搭載するよう考慮している。

### 5.3 VLSIプロセッサの仕様の概要<sup>(73)</sup>

VLSIプロセッサの仕様には、基本的には既存D10プロセッサとの互換性を維持しつつ、上位拡張を図る。実現する仕様一覧を表4.2に示す。

特徴的な拡張機能は以下の通りである。

- (a)メモリアドレスは今後のソフトウェアの拡大傾向を考慮し、27ビットに拡張する。
- (b)交換処理用国際標準言語CHILLの並列記述機能を効率的にサポートするためスタック機能を具備する。
- (c)既述のプロセッサバス仕様にに基づき、マルチプロセッサがプラグインで実現可能とする。

### 5.4 VLSIプロセッサの設計手法<sup>(74)</sup>

VLSIプロセッサの設計工程は、仕様から論理回路の作成に至る設計工程と作成した回路の正常性を検証するデバッグ工程からなる。本VLSIプロセッサは、カスタムプロセッサとしての開発であり、この特徴を活かすこと、また、VLSI内部の回路修正が困難な点を考慮して進めなければならない。VLS

表4.2 論理仕様

項目		VLSIチップ具備機能	VLSI処理装置	(参考) D10HCP	
中 央 制 御 部	命 令 仕 様	命令語長 (ビット)	32, 64, 96	32	32
		命令数(種)	233	168 <sup>1*</sup>	148
		データ長 (ビット)	32, 16, 8/語 語内の任意のビットフィールド	同左	32ビット/語 ビットフィールド
		汎用レジスタ 数	16	16	16
		専用ベース レジスタ数	16	(汎用レジスタ兼用)	(汎用レジスタ兼用)
		制御レジスタ 数(個)	17	12	12
	スタック構成	システムスタック ユーザスタック } 2面	-	-	
	マルチプロセッサ 制御構成	有	有	-	
	割込仕様	2レベル53要因 固定割込 16レベル256要因 ベクタ割込	2レベル43要因 固定割り込み方式	2レベル41要因 固定割り込み方式	
	主メモリ最大アド レス容量(語)	$2^{19}, 2^{24}, 2^{27}$	$2^{24}$	$2^{19}$	
データチャネル 接続数	8, 32 <sup>2*</sup>	8	8		
(CC)	マ イ ク ロ 制 御 プ ロ ゲ ム ラ リ ム	アドレス容量	64k(32ビット/語)	4K(32ビット/語)	2K(28ビット/語)
		書き込み可否	可	不可	不可
		ワークレジ スタ数	12	12	4
		アドレス スタック数	4	4	1
IO インタフェース	DEX IO インタフェース DIPS IO インタフェース	DEX IOインタフェース	DEX IOインタフェース		
転送モード	IOG・ローカルバースト、マルチ プレクス、ブロックマルチプレクス	IOG・ローカルバースト、マルチ プレクス	同左		
主メモリ最大アド レス容量(語)	$2^{19}, 2^{24}, 2^{27}$	$2^{24}$	$2^{19}$		
チャネル語(CHW) 数(語)	4語×256 IO=1K(主メモリ部に格納) 8語×256 IO=2K(DHC部に格納)	2K	1K		
最終転送語のバイ ト位置指定	可	可	不可		
1語の下位バイト からの転送	可	可	不可		
制 御 メ モ リ	アドレス 容量(語)	16K(32ビット/語)	4K <sup>3*</sup> (32ビット/語)	1K(28ビット/語)	
	書き込み	可	可	不可	

\*1 D10 HCPの命令にマルチプロセッサ制御用の命令用を追加

\*2 ベクタ割り込み方式と併用

\*3 4Kのうち2Kはチャネル語格納エリア

I プロセッサの設計に関する総合的な流れ図を図4.31に示す。

#### 5.4.1 論理設計手法

VLSI プロセッサの設計は方式側と部品側の連携によりなされる。VLSI プロセッサのほとんどの回路がVLSI チップの中で実現され、また部品技術が未確立の段階から設計を始め、プロセッサチップの完成と部品技術の確立がほぼ同じとなる設定で進める。各種論理回路の設計条件はLSIの部分試作で確認した値を用いる。方式側は仕様から論理回路までを実現し、部品側は論理回路をチップ上の実回路として実現してLSI化する。

論理回路の記述は、従来のSSIやMSI相当の論理セルを用いて記述する。また、本VLSIプロセッサはカスタムプロセッサである特徴を活かして設計を進める。すなわち応用システムに適合させたVLSI回路を実現する。VLSIチップ内部回路はVLSIプロセッサ装置の中核であるが、あくまで一部の回路と位置づけて設計している。設計した論理回路はTTL-IC等の汎用部品を用いたハードウェアシミュレータの製造にも利用する。VLSIが完成すればハードウェアシミュレータの中の対応部分がそのままVLSIに置き代わり、ただちにVLSIが装置やシステムレベルで試験可能となる。

以上のように、システムの中の一部としてVLSIチップを扱うこと、VLSI上の回路もSSI、MSI相当の論理回路記法を用いて設計することから論理設計手法は従来の論理装置設計と大きく変わるものではない。部品が予め規定した条件通りに動作する保証があれば、方式側からVLSIを見たとき、VLSIチップは回路を描くキャンバスが異なっただけである。

#### 5.4.2 デバッグ手法<sup>[72]</sup>

論理設計の段階における方式側の作業は、既存方式とは大きくはかわらなかった。しかし極めて高精度の論理回路が要求されることからデバッグ段階では既存システムの条件と大きく異なると考えなければならない。従来の開発手順では回路の修正は、いわば運用開始直前まで可能であった。しかし、VLSIではクリーンファイルを部品側に渡す時点、すなわちまだハードウェアが無い時点で紙の上で回路の完全凍結を行なわなければならない。実システムで回路の

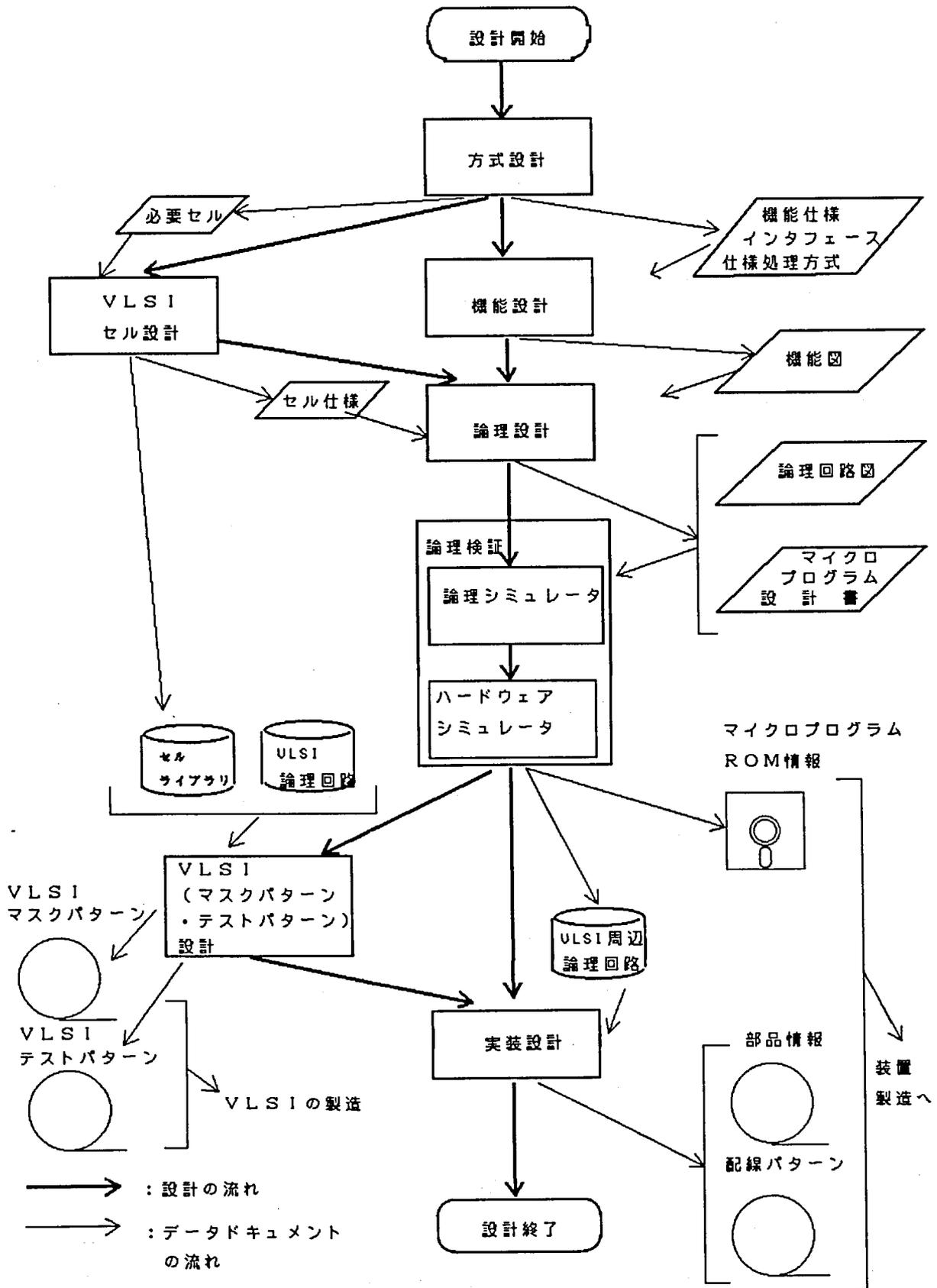


図4.31 交換用VLSI処理装置の設計手順

正常性を確認する以前に、システムにも相当する大規模な回路を凍結しなければならない。

(1) 交換用カスタムVLSIプロセッサ向きデバッグ手法の確立法

高精度の論理回路を実現するため、VLSIプロセッサ向きの新しいデバッグ手法を開発しなければならない。しかし、無から有は生まれず、まず既存のプロセッサ開発の経験を復習して、その後、各種条件をVLSI開発に置き換え、カスタムVLSIプロセッサ向きの新しいデバッグ手法を確立することとする。

(2) 既存装置によるデバッグ状況分析

(i) 試験対象装置の概要

既存プロセッサでのデバッグ経過を元にVLSIプロセッサ向きの有効なデバッグ手法を検討する。既存のプロセッサは16kゲートの論理回路と約30kビットのマイクロプログラム制御メモリからなり、実現しようとするVLSIプロセッサとほぼ同規模のプロセッサである。このプロセッサを交換機制御用に用いる場合には図4.32のような構成をとる。システム高信頼化を図るために交換機制御用には、二重化構成を前提とする。

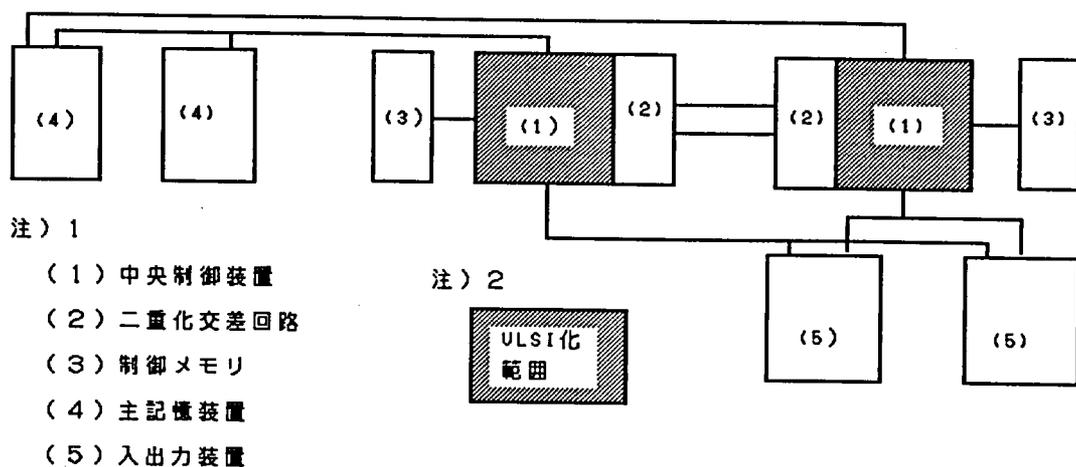


図4.32 バグ検証装置構成

(ii)デバッグ経過

交換機制御系のデバッグは、各装置単体だけでなく複数装置を組合せてデバッグを行なうことが必要である。複雑な装置間の干渉を含むあらゆる事象に対してデバッグを行なわなければならない。高信頼化二重化システムでは、一方に故障が生じた場合、他の系が正常である限り、この故障が他の系に影響を与えないことをあらゆる条件で確認する必要がある。一方を強制的に故障状態にし、他方の系で装置間の複合試験を行なう。

上記の試験対象装置における試験工程とバグ検出状況を図4.33に示す。各工程では、以下の特徴を有するデバッグを行なう。

机上デバッグ

技術者自身が共同で他人の回路をトレースするものである。CAD技術が未熟な時期には、机上デバッグの精度が後の工程に大きく影響するほど効果は大きく、この工程に多くの工数を費やした。現在は図的入力と直結した計算機による論理シミュレーションが可能であり、この工程の意味は初心者の教育効果を期待する程度であろう。

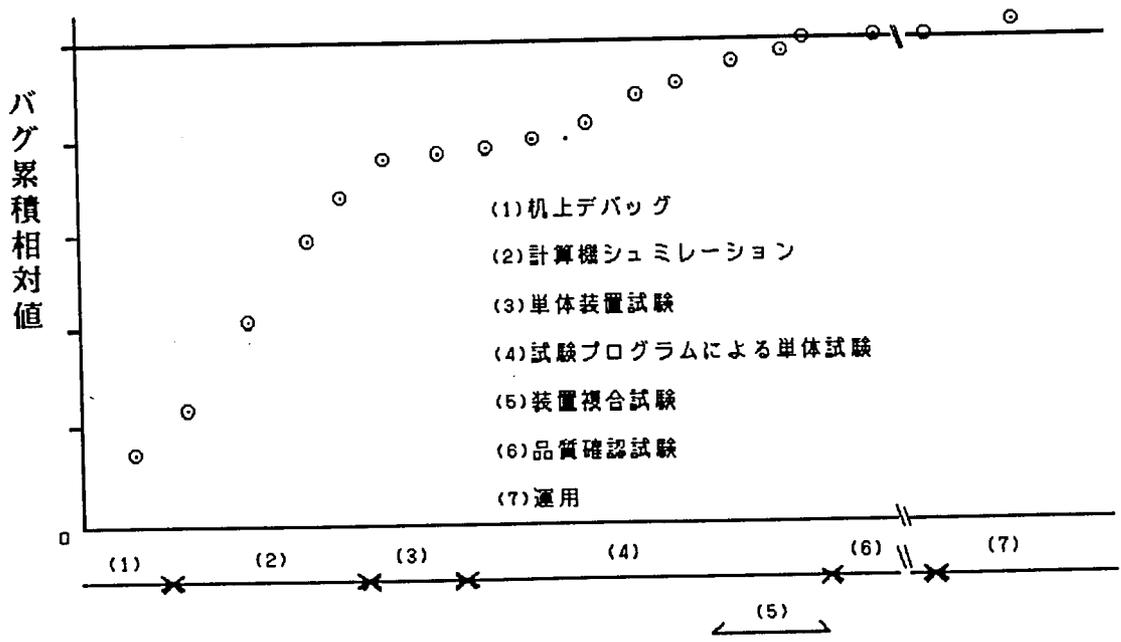


図4.33 デバッグ過程とバグ検出状況

## 計算機シミュレーション

机上デバッグにより おおよその論理検証がすむと、論理ファイルの記述通りに計算機上で回路をトレースさせる。ただし、回路の膨大さから複雑な動作の検証は困難であり、プロセッサの場合は各命令を実行させる程度である。各種の条件の組合せを大規模に行なうことは不可能である。

## 単体装置試験

論理ファイルに従ってハードウェアが実現されると計算機シミュレーションと同程度の項目の試験を行なう。本来、この工程はハードウェアの製造に伴うミスを検証するものであり、論理シミュレーションが完了しておればただちに次の工程に移れる。高速論理シミュレーションが可能なら、この工程での論理バグの検出量は少なく工程も短くなる。

## 試験プログラムによる単体試験

予めプロセッサの仕様に従った条件設定を行なったプログラムを作成しておき、このプログラムを動作させて正常性を確認する。プログラムの走行により幅広い機能の確認が可能であるが、試験項目は人間の考える範囲であり、条件の組合せには限界がある。

## 装置複合試験

試験プログラムによる装置間の結合機構やソフトウェアとハードウェアの組合せにより生じる複雑な条件を試験する。通常、限られた試験項目を消化するハードウェアの試験だけでは、プロセッサのような論理深度のある装置のデバッグは十分には行えない。実際に運用するソフトウェアを用いた複合試験で、設計者が想定し得ない複雑な組合せ現象を生じさせて、デバッグを行なうことが必要である。工程(4)の試験プログラムによる単体試験では一見バグが収束したように見えるが、この工程(5)によるソフトウェアとの複合試験の併用で、新たにバグが生じてくる状況が図4.33から読み取れる。

## 品質確認試験

最終工程(6)では、実際の呼処理プログラムを用い、運用状態と同じ状況でシステム全体にわたるバグが顕在化するかどうかを検証する。バグが顕在化せず、安定状態がどれだけ長期間確保出来るかにより、運用に入れるかどうかを判断する。

### (iii) 残存バグの状況

図4.33の例では、運用開始後にも、バグが顕在化するように読み取れる。しかし、実はこのバグは運用システム自身で検出されたものでなく、同一プロセッサを用いた他のシステム開発中にバグが検出したものである。他のシステムで新しい条件を構成したときバグは顕在化する。

### (iv) デバッグ手法の評価

以上の経過から、適用対象システムと同等のソフトウェアを用いた複合試験で確認してからVLSI回路を凍結することが重要であることが解る。幸い、カスタムVLSIプロセッサは既存のプロセッサと仕様継承の範囲が大きく、ハードウェアシミュレータにおいては、既に開発済みの呼処理プログラムが実時間で動作可能である。従って、既開発の呼処理プログラムを用いた複合試験や品質確認試験を行なうことが重要である。ソフトウェアの絡む大規模なシステムではハードウェアシミュレータの位置づけが大きいことが解る。

### (3) 残存バグの解析

この分析では、運用に入ってから、すなわち図4.33(7)工程でそれまでのバグ累計に対して約3%の残存バグが検出された。このバグは運用システム自身には影響は無かったが、ここでは、もし上記の検証対象装置がVLSIで実現されていたら、このバグをVLSIを作り替えなくても修正できるかどうかを分析する。なお、既存装置はSSI, MSI, 一部LSIで実現しているため、これらのバグに対しては布線変更などの理想的な修正を行なっている。VLSIの条件を以下のように設定した。

- (a) 16kゲートの論理部を1チップのVLSIで実現する。
- (b) マイクロプログラム制御メモリはチップの外にあり、1マイクロ命令実行ごとにマイクロ命令を外から供給する。
- (c) VLSIはCCA, CCB, DCHの3種のチップで実現されているとする。

上記の例ではVLSI化後（すなわち運用開始後）に約3%のバグが顕在化するが、以下の4案の処置法でVLSIの内部を修正しないで対処できることが判

明した。これらの対処法に対応するバグの比率を図4.34に示す。

(a) マイクロプログラム制御メモリをVLSIチップの外に置くことにより、約30%のバグ、しかも論理仕様にかからむ重要なバグがマイクロプログラムの修正で処置可能である。

(b) 装置間インタフェースにかからむようなバグはチップの外の付加回路で処置可能となる。

(c) クロック幅を延ばし、ある程度時間緩和を図ることにより仮処置ができる。

(d) 保守命令に関係したバグはサービスに影響を与えないため、ハードウェアの変更より、むしろ装置の試験診断プログラムの変更で対処する。

筆者等はマイクロプロセッサの開発の経験を持たなかったが、この解析により従来の論理設計の手法を用いれば、たとえ、交換機のような厳しい条件のシステムに適用するカスタムVLSIプロセッサも実現可能の自信を得た。この観点からも実時間で論理検証ができるハードウェアシミュレータは大規模システムの開発には必須と言える。

#### (4) 採用したデバッグ手法

設計した論理回路は2種のシミュレータを用いて回路検証を行なっている。論理回路をファイルのまま計算機で動作させる計算機シミュレータによるデバッグと、論理回路を標準TTL素子を用いて実際にハードウェアとして実現したハードウェアシミュレータによるデバッグである。

両者それぞれに得失があるが、計算機シミュレータの利点は試験条件の設定が容易なこと、データの修正だけでバグ修正ができること、またバグの要因が識別しやすいことにある。さらに、デバッグしている論理ファイル自身で論理回路が一元的に管理ができ、デバッグ後の修正の経過がそのまま実回路にフィードバ

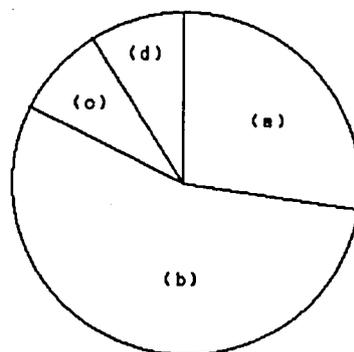


図4.34 バグ処置法とその比率

- (a) 外付マイクロプログラムの修正
- (b) VLSI外付加回路修正
- (c) 時間緩和
- (d) 保守命令関連(仕様変更)

ックされていく点にある。一方、その弱点は実行時間が遅い点にある。試験規模、回路規模の増大に対し、大幅にシミュレーション時間が増大する。

ハードウェアシミュレータの利点はシミュレーションの高速性にある。その弱点はバグがあることが分かってもバグの探索には、素子動作を実際に順に追跡していくことが必要であり、バグの修正もハードウェアを実際に修正するための時間を要する。試験項目ごとのTAT (Turn Around Time) が比較的長いことが問題となる。また正確なタイミングシミュレーションが出来ない問題もある。さらに、VLSI化での大きな問題はハードウェアシミュレータと論理ファイル原本との不一致の可能性にある。ハードウェアを実際に修正して論理確認を行なっても、論理ファイルの原本の修正は別途独立に作業することによる問題である。VLSI化すべき本物の回路にバグが残る危険がある。

計算機シミュレーションはデバッグ初期の比較的バグが多いとき、ハードウェアシミュレータはデバッグ後半でバグが収束し、大量にシミュレーションを行なう必要があるときと使い分けることが必要である。なお、ハードウェアシミュレータの応用例として交換機の実環境での論理デバッグへの利用が重要であるが、これについては既に述べた。

#### 5.5 VLSIチップの実現結果<sup>[66][70](73)</sup>

設計、デバッグを完了した論理ファイル、すなわちクリーンファイルは、部品側でパターン設計等の工程を経て、LSI上の実際の回路として実現される。さらに、LSI製造工程を経てカスタムVLSIチップが完成する。実現したチップの主要諸元を表4.3に各VLSIの機能ブロックを図4.35に示す。

#### 5.6 マルチプロセッサへの応用と評価<sup>(71)</sup>

システムに実際に応用するプロセッサの実装形態を処理装置と呼ぶ。実現したVLSIチップを用いて、各種の処理装置を実現している。交換機制御用主制御処理装置および付帯系制御処理装置におけるVLSIチップの応用の概念図と実現イメージをそれぞれ図4.36と図4.37に示す。また、実現したプロセッサの仕様を表4.4に示す。

実現した性能は、交換処理の命令分布で平均命令実行時間が約1.5 $\mu$ 秒となっ

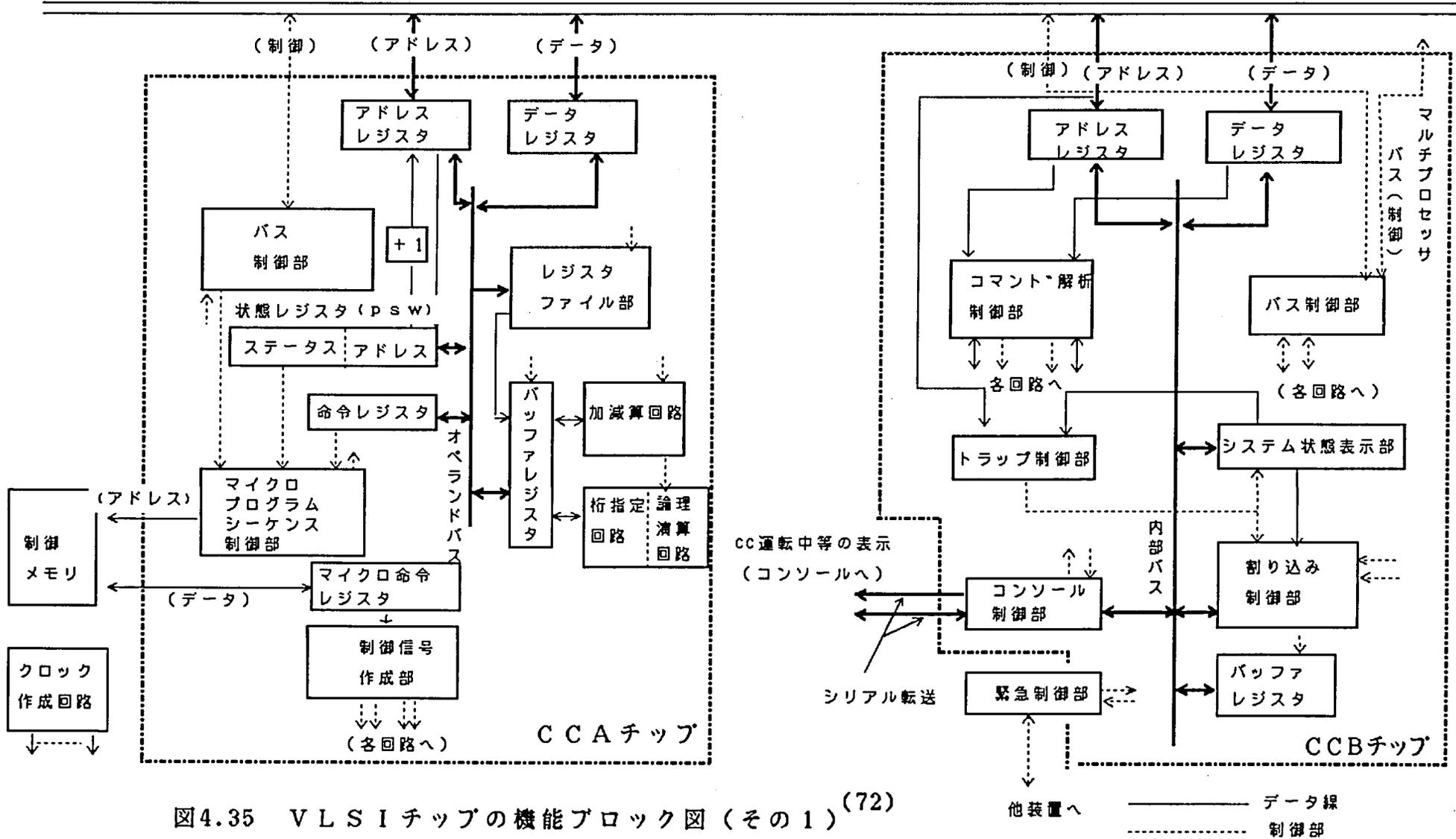


図4.35 VLSIチップの機能ブロック図(その1)<sup>(72)</sup>

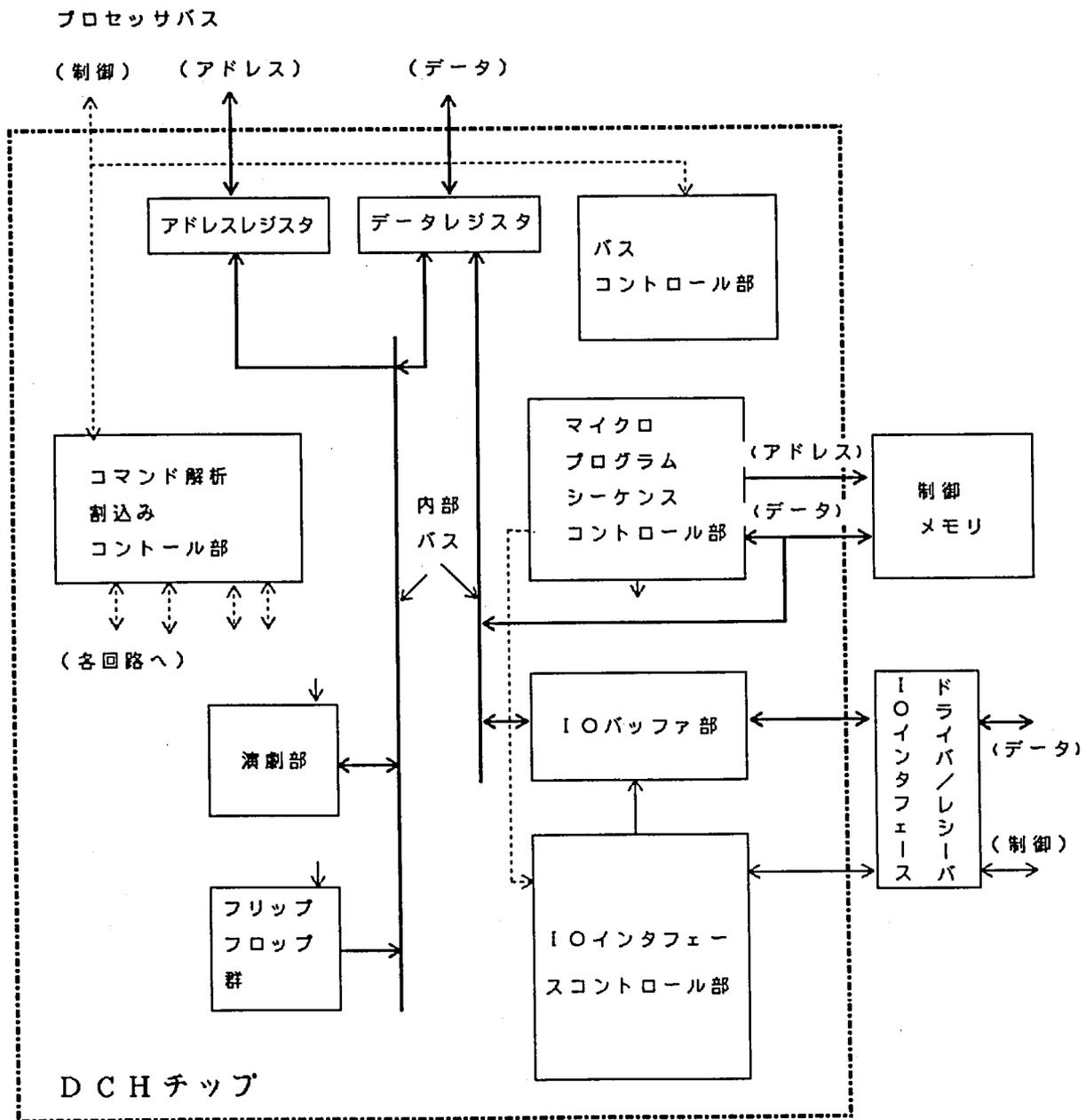
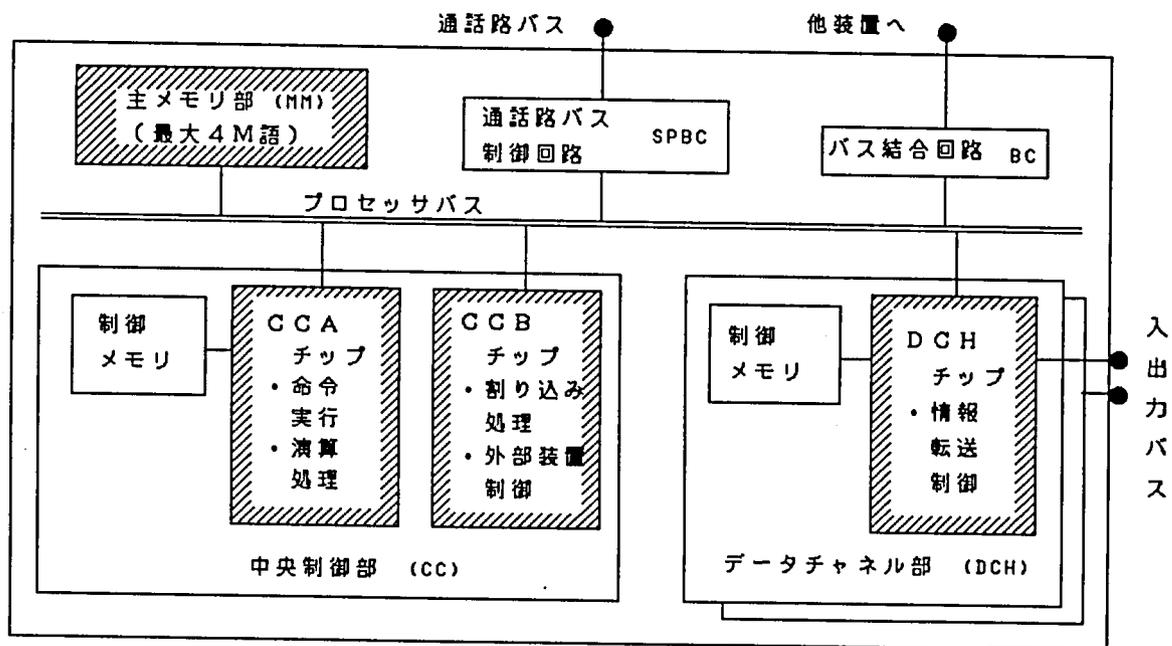
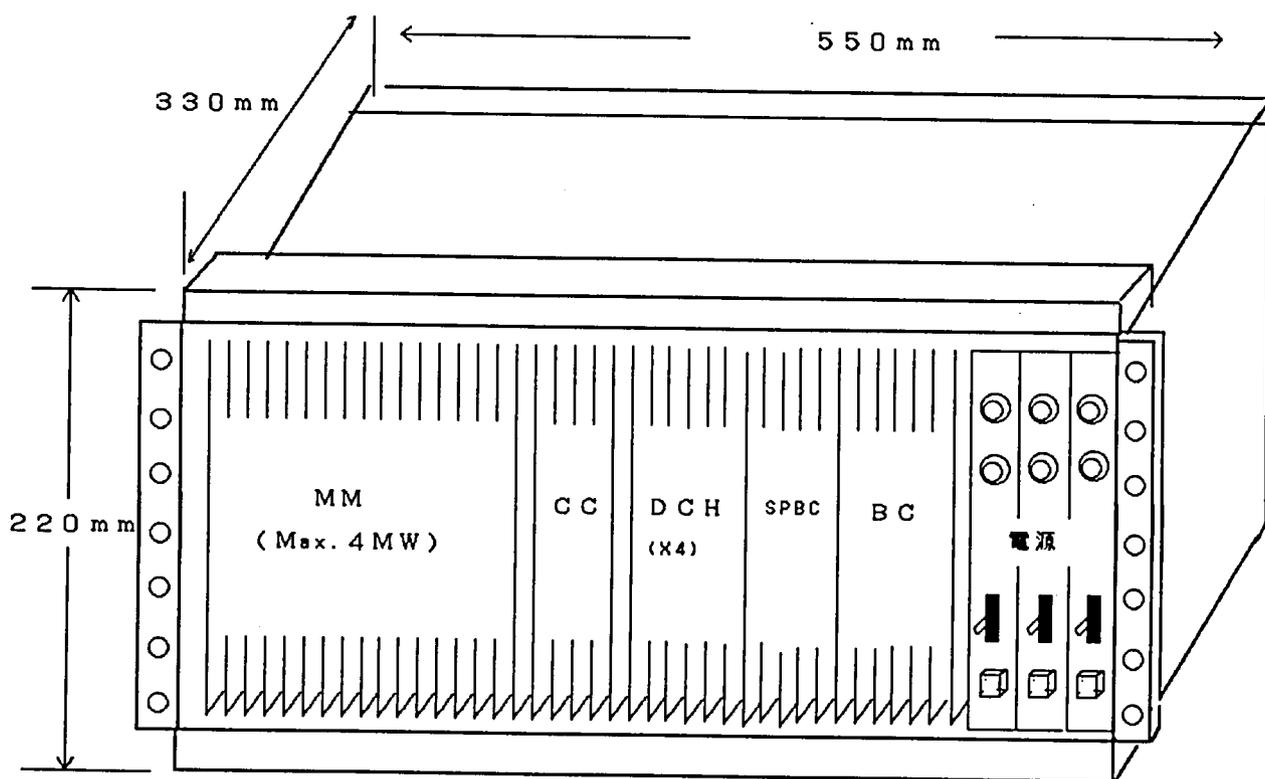


図4.35 VLSIチップの機能ブロック図(その2)<sup>(72)</sup>



(a) 交換用VLSI処理装置のブロック構成



(b) 交換用VLSI処理装置の実装概念図

図4.36 VLSI処理装置の概要 (70), (72)

付帯制御 1 ボードプロセッサ

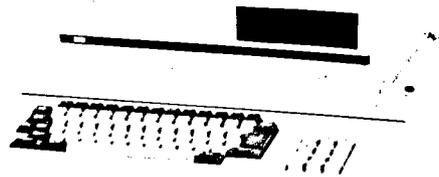
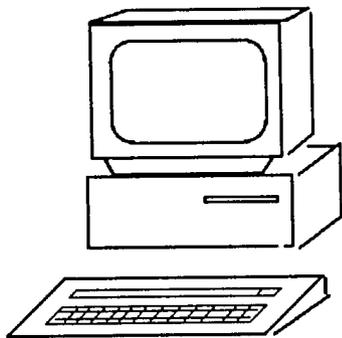
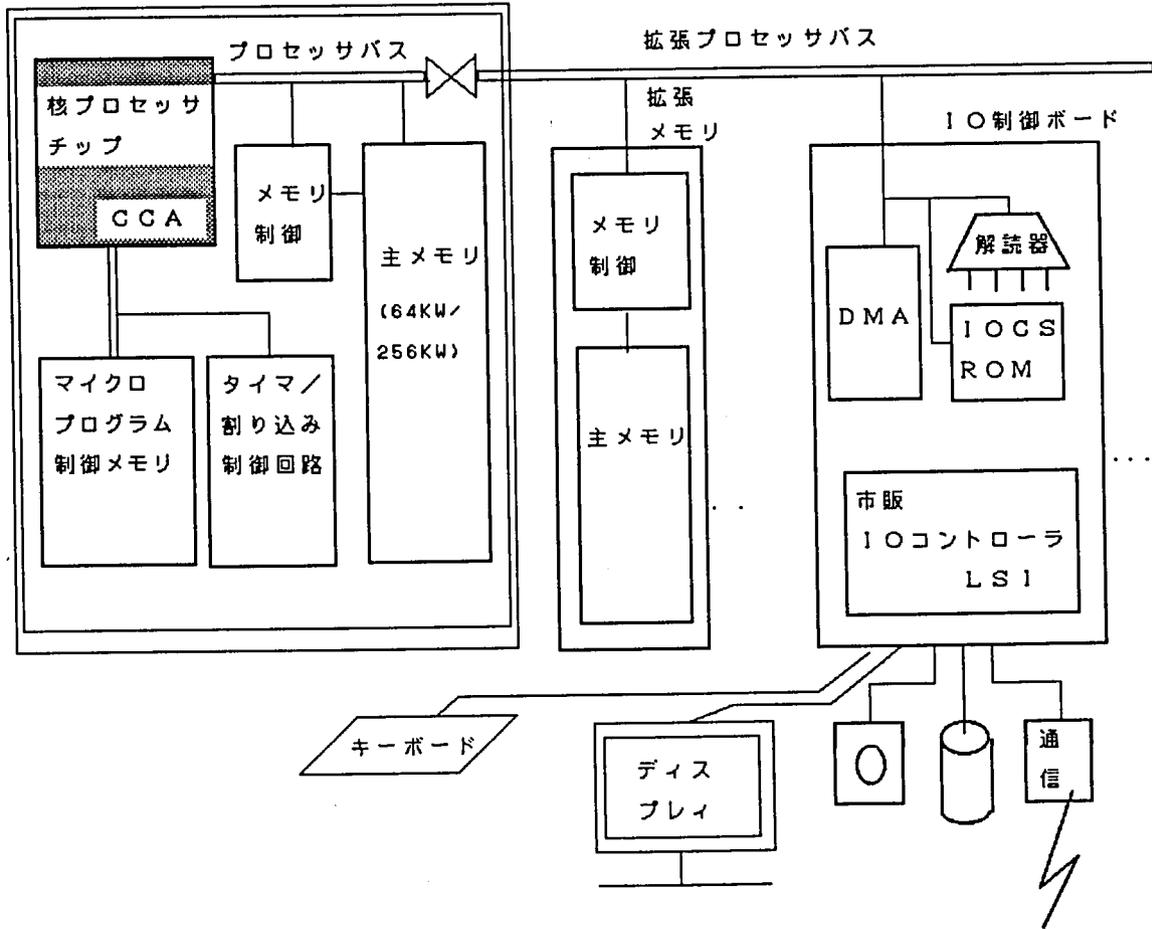


図4.37 支援処理装置の例

表4.4 処理装置の諸元（デュプレックス構成時の片系）<sup>(72)</sup>

諸元		内容
パッケージ枚数		28枚（電源を除く）
大きさ（H）×（W）×（D）		220×550×330mm
消費電力（平均値）		約200W
通話路バス 制御回路 （SPBC）	通話路（SP） インタフェース	D70SP インタフェース
	増設単位	2方路
	最大接続方路数	5方路
データ チャンネル部 （DCH）	10インタフェース 最大転送速度	0.95Mバイト／秒
	10インタフェース	DEX 10インタフェース
	増設単位	1台
	最大実装台数	4台
主メモリ部 （MM）	語構成	32ビット + 8ビット （情報ビット） +（チェックビット）
	増設単位	256k語／1M <sup>*1</sup>
	最大実装台数	1M語／4M語 <sup>*1</sup>
クロックサイクル		220ns
平均命令実行時間		1.5μs

\*1 64Kビット素子/256Kビット素子を使用したとき

表4.3 VLSIチップ諸元

項目 \ チップ	CCA	CCB	DCH
ゲート数	12 k	9 k	14 k
トランジスタ数	58 k	46 k	73 k
端子数	185	165	207
消費電力(MAX)*	600 mW	200 mW	300 mW
チップ面積 (mm×mm)	9.7×9.7	9.4×9.8	12×12

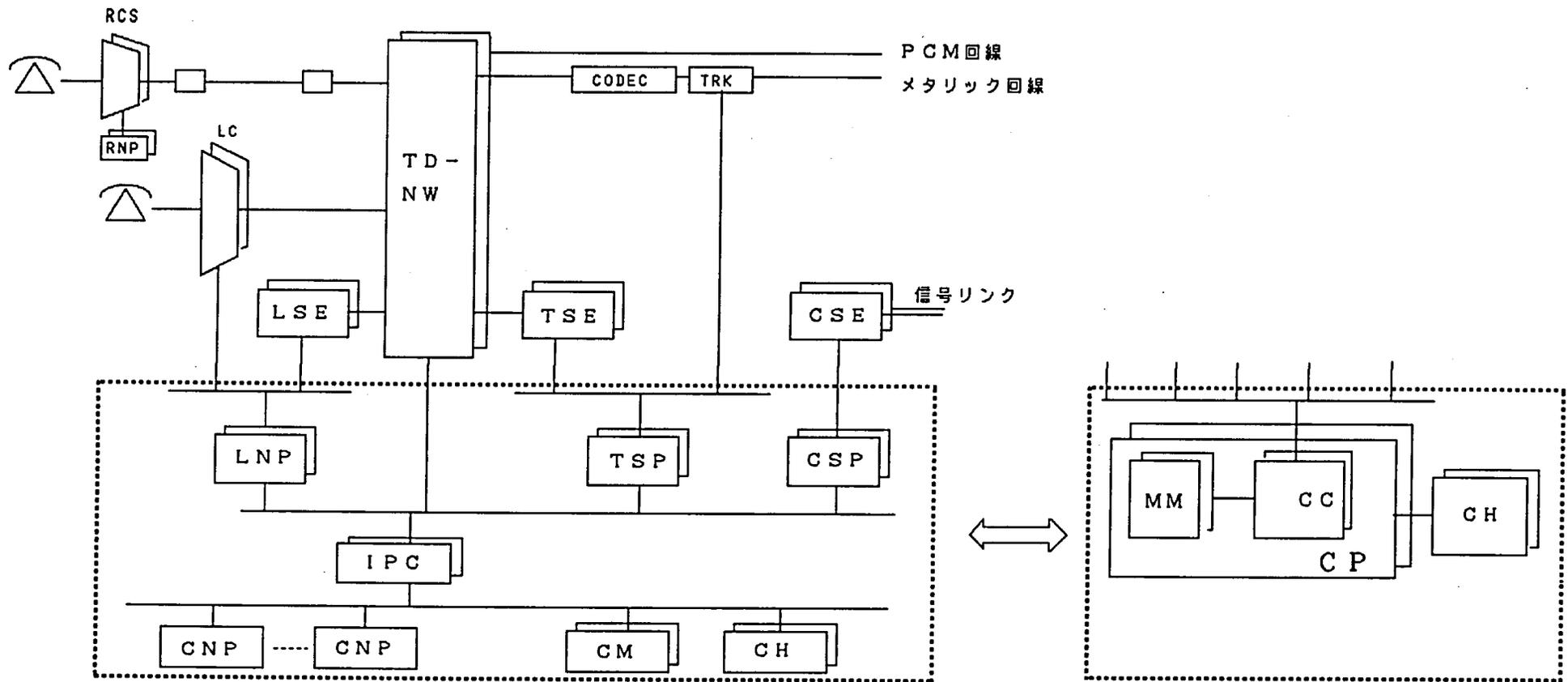
\*クロック5MHz時の最大値を示す。

ている。利用するメモリにより性能は変わるが、既存のECL素子を用いた高速プロセッサは平均命令実行時間が0.6μ秒であり、能力低下は著しい。ただし、VLSIプロセッサの狙いは、もともとコストパフォーマンスの向上にあり、マルチプロセッサにより処理能力向上を図ることから能力低下は問題でない。

このVLSI処理装置を用いて構成したマルチプロセッサの例を図4.38に示す。この接続構成に従い、ケーブルコネクタの裏面への挿入によりプロセッサ間の結合が可能であり、図4.36のVLSI処理装置をプラグイン形式でマルチプロセッサに組立てる。規模や機能拡張に合わせた処理装置の追加も同様にプラグインで実現され、ビルディングブロック構成を実現している。実現したカスタムVLSIプロセッサによるマルチプロセッサの主な評価を以下に示す。

#### (1)性能の評価<sup>[78]</sup>

マルチプロセッサの性能に関するハードウェアの条件の規定は容易であるが、ソフトウェアの条件は局種や局による呼種の分布やサービスの実現レベルによってさまざまに振れる。電話交換のモデル条件による評価の一例を図4.39に示す。単純な電話交換呼では、750kBHCA (Busy Hour Call Attempt) に達する。



(i) マルチプロセッサ制御方式

(ii) シングルプロセッサ制御方式

TDNW : 時分割通話路装置  
 LC : 集線装置  
 LSE : 加入者線信号装置  
 TSE : 中継線信号装置  
 CSE : 共通線信号装置  
 TRK : トランク装置  
 RCS : 遠隔制御交換装置

RNP : 遠隔地通話路制御プロセッサ  
 LNP : 加入者線制御プロセッサ  
 TSP : 中継線信号制御プロセッサ  
 CSP : 共通線信号制御プロセッサ  
 CNP : 呼制御プロセッサ  
 IPC : プロセッサ間通信制御装置  
 CM : 共通メモリ

図4.38 D70デジタル交換機の構成

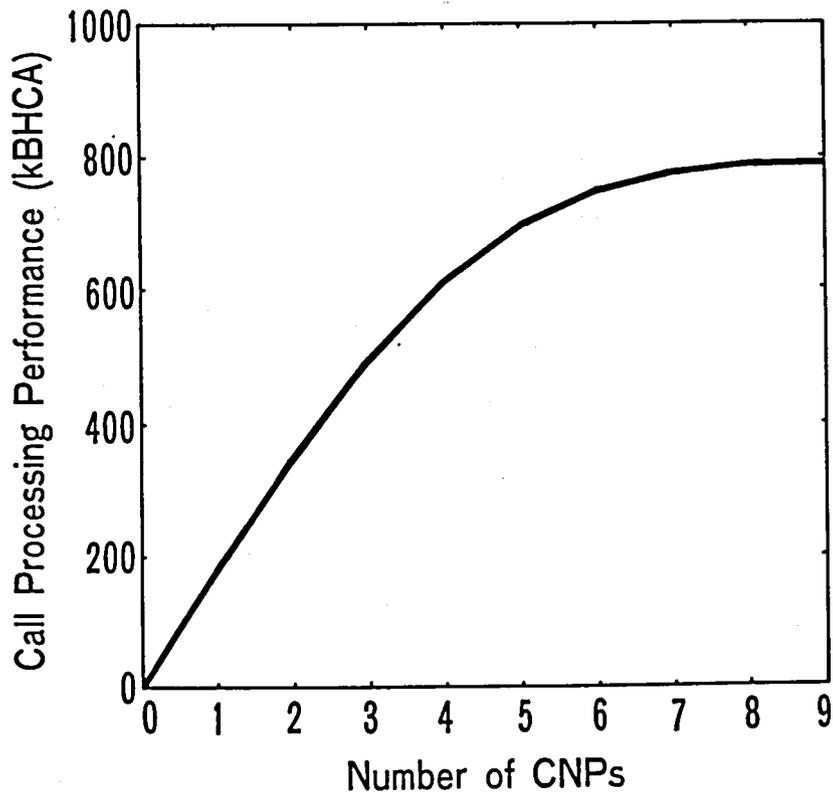


図4.39 VLSI マルチプロセッサの呼処理性能

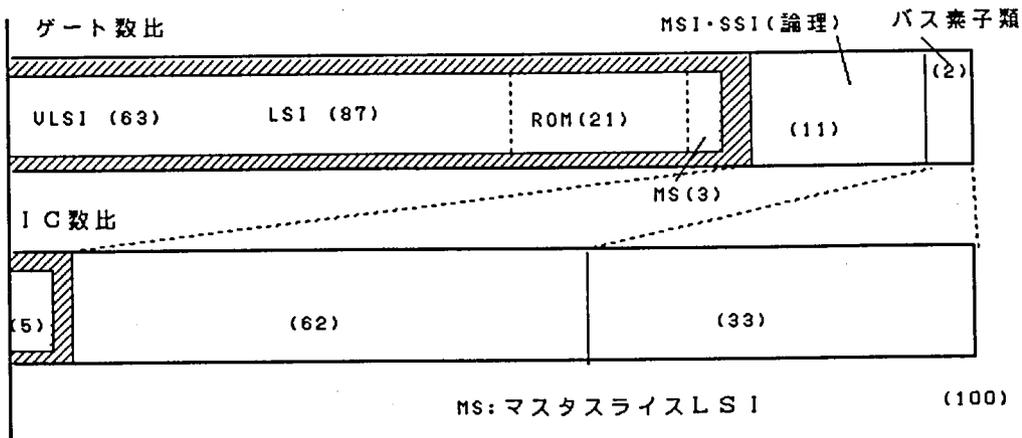


図4.40 VLSI化比率

既存の16ビットプロセッサで実現した実験機に対して、約4倍の性能向上と推定される。

## (2) VLSI利用の考察

VLSIが装置にどの程度貢献しているかを評価する。主制御VLSI処理装置におけるVLSI比率をもとめたものを図4.40に示す。回路規模(ゲート数)では90%近くがVLSI素子に含まれており、その比率はきわめて高い。一方、部品点数では、高集積部品の占有率は5%に過ぎない。交換機では外部装置制御、高信頼化のための交差など対外装置との接続インタフェースが多く、しかもこの部分はケーブル駆動するため比較的大電力を消費する素子の利用が必要であり、高集積化出来ないという問題が残った。

しかし、約90%の回路規模を高集積化できた効果は大きい。VLSIの最も大きな特徴である体積減少効果を中心に、従来の処理装置からの改良効果を総合的に評価したものを図4.41に示す。

### ・体積削減効果

既存の100ゲートクラスのLSIを用いた交換用主制御プロセッサの二重化構成に対し、単位プロセッサでは約1/5へ体積が削減されている。

### ・コストパフォーマンス向上

体積削減は部品点数の削減を反映して、コスト指標としても有効と考えられる。この体積削減効果をそのままコスト削減率と見なすとコストパフォーマンスは約2.5倍向上したことになる。

### ・高信頼化

高集積化による部品点数の削減により、信頼度は100倍のオーダで向上している。

## (3) ONE-MACHINE 概念の達成

システムの具体化に当たっては、ONE-MACHINE 概念を設計概念とした。ここで、ONE-MACHINE 概念の達成度をハードウェアおよびソフトウェアについて評価する。

(名称)	D10 中央処理装置	D10 高速 中央処理装置	VLSI 処理装置	
(開発年次)	(昭45)	(昭52)	(昭58)	
(技術のポイント)	・高信頼、高性能 実用交換用処理 装置技術の確立	・高速化 ・効率化 (高速素子能力を性能 以外にハードウェア規 模の削減に利用) ・マイクロプログラム 多重制御方式の確立	・32ビットプロセッサ 核機能のオンチップ化 ・交換制御用マルチ プロセッサ制御方式の 確立	
(主要部品 技術)	・SSI(CSL) ・磁気コア	・MSI(ECL) ・ICメモリ 4/16kDRAM	・ULSI(CMOS/TTL) ・ICメモリ 64/256kDRAM	
(メモリ容量の 拡大)	0.25M語	0.5M語	16M語	
(小型・経済 化傾向)	12架	4架	シングル 0.3架	マルチ 2-4架
(性能の向上) -相対値-	0.26	1	0.4	1-5
(信頼性の向上) -相対値-	0.8	1	10	

図4.41 交換用処理装置の歴史的進展経過 (70)

#### ・ハードウェアの ONE-MACHINE 概念達成度

ハードウェアの印刷配線板の品種数から、シングルプロセッサとマルチプロセッサの構成品の共用の度合を評価したものを図4.42に示す。シングルプロセッサ用部品はほぼマルチプロセッサと共用可能であり、ONE-MACHINE 概念の達成がうかがえる。ただし、小規模ではマルチプロセッサ用の付加回路が多い点が問題とされるべきであろう。なお、この付加回路の主体はケーブル接続用の単純な機能がほとんどである。マルチプロセッサの大処理能力は複数プロセッサの結合により実現されるが、この大処理能力を実現するための複数プロセッサ間の結合機構が大きい。しかも高信頼を図るために予備間の結合も必要になり、この結合機構がさらに肥大化している。また大規模では相対的にマルチプロセッサ用付加回路の比率が減少し、マルチプロセッサが大規模で有効なことを示している。

#### ・ソフトウェアの ONE-MACHINE 概念の達成度<sup>(58)</sup>

ソフトウェアについて、のシングル/マルチプロセッサ両構成の共用度を図4.43に示す。全体の80%はシングル/マルチプロセッサ両構成で共用可能としている。特に交換機として変更や機能追加の大きい呼処理・保守運用プログラムの大部分を両制御系で共用可能としている。なお、この共用はソフトウェアの管理単位での評価であり、詳細な部分での共用度は、実際にはさらに高い。ソフトウェアの観点では ONE-MACHINE 概念は十分達成できたと考えられる。

ONE-MACHINE 概念は、もともとソフトウェア系列の一元化を図るための考え方であり、ソフトウェアの ONE-MACHINE 概念による効果がハードウェアの ONE-MACHINE 概念による効果を越えているのは当然の結果である。そもそもマルチプロセッサのハードウェア付加回路は、ソフトウェアの ONE-MACHINE 概念を実現するための支援機構である。

とは言え、今後このようなマルチプロセッサのオーバヘッドを削減することは必要である。これらケーブル結合のための低集積部品を用いたハードウェア機構は、現在の段階でのマルチプロセッサではやむ得ない本質的な問題であるが、今後さらに高集積化が進行する。より高集積向きの新しいマルチプロセッサを追求していくことが必要である。

シングルプロセッサ

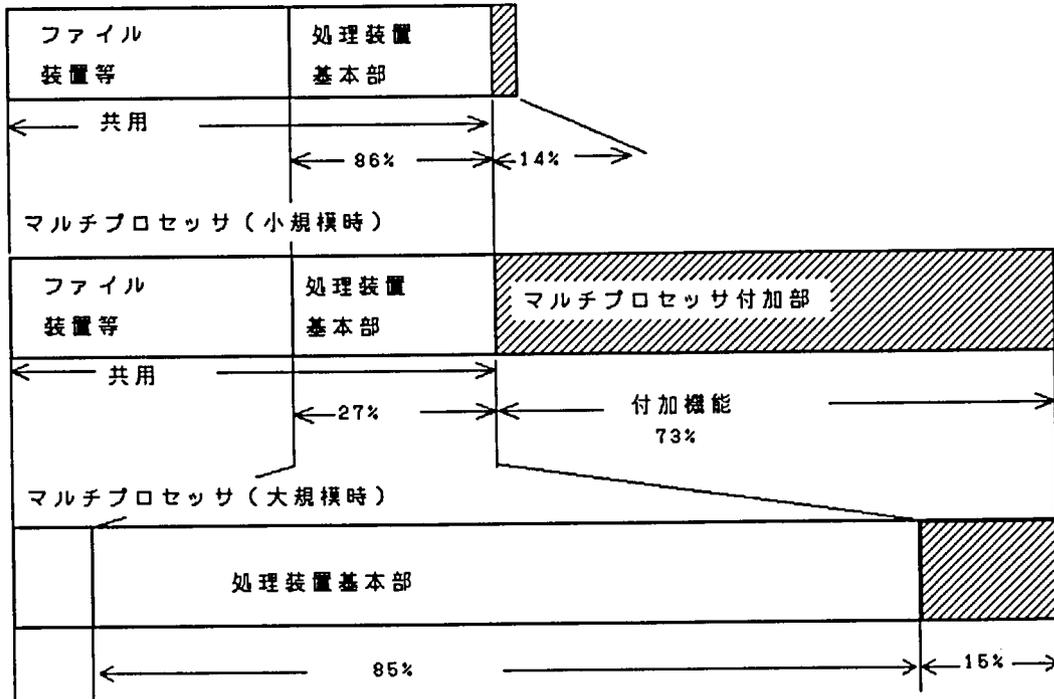
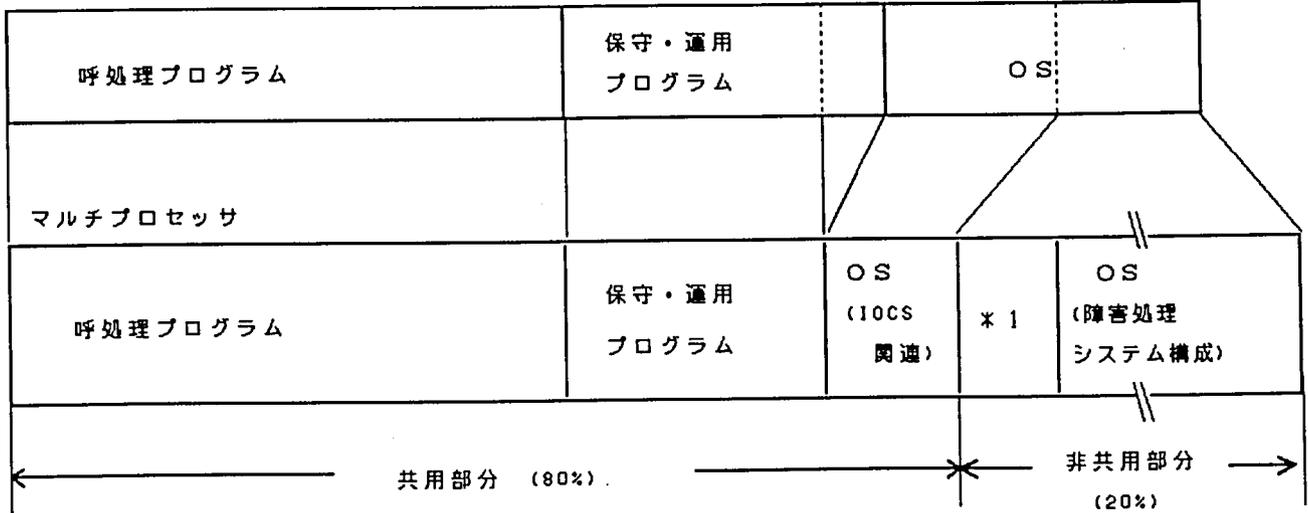


図4.42 制御系ハードウェアにおける  
シングル/マルチプロセッサ間の共通性

シングルプロセッサ



\* 1 保守運用(工事・増設)

図4.43 シングル/マルチプロセッサ間のプログラム共用度 (60)

## 6. 結言

第3章で規定したマルチプロセッサ基本構成を具現化する段階での設計思想の構築と、これを具体化する実現手法について述べた。要点は以下の通りである。

### (1) システム実現に当たっての設計理念の構築

小規模用のシングルプロセッサ制御方式と大規模向きのマルチプロセッサ制御方式による交換機を1系列化して実現する統一概念 ONE-MACHINE 概念を明らかにした。具体的には制御システムを階層的に分解し、各階層におけるインタフェースをシングル/マルチプロセッサ両系間で統一し、各階層内での構成部品をそれぞれ共通化するものである。同時にマルチプロセッサの構成要素をシングルプロセッサと等価にすることにより、複雑化しやすいマルチプロセッサの単純化を図るものである。

### (2) 改良 D 2 0 プロセッサを用いたマルチプロセッサ制御方式実験機

既存のシングルプロセッサ用プロセッサを用いて ONE-MACHINE 概念を具体化する手法について示した。シングル/マルチプロセッサ両方式間でソフトウェアを極力共通化するプロセッサ間通信機構や共通メモリの具体的実現について述べた。プロセッサ間通信機構にはハードウェアで自律的に負荷分散を行なう機構を提案した。また、実現結果について評価し、最大16台のプロセッサを実装して、プロセッサ12台分に相当する処理能力を得ることを示した。

### (3) V L S I プロセッサによる改良

カスタムプロセッサの有効性を示し、32ビットカスタム V L S I プロセッサの実現上での課題、すなわち V L S I の入出力ピンネックや回路の高精度化について解決手法を示した。この V L S I プロセッサにより、実験機の少なくとも4倍の処理能力を実現している。また、上記の ONE-MACHINE 概念の達成度を評価し、ソフトウェアでは十分に目標を達成し、ハードウェアでは小規模でオーバーヘッドが多いものの大規模では共用比率が向上していることを示した。また今後、V L S I 技術を方式に適用していく場合、装置間インタフェースを改良することが重要であることを指摘している。

## 第V章 デジタル交換機制御用マルチプロセッサの高信頼設計<sup>[75]</sup>

### 1. 序言

ハードウェアはいつかは故障し、巨大ソフトウェアには必ずバグが残存する。このため24時間無休止サービスを前提とする交換機には、故障やソフトウェアのバグの顕在化などの擾乱を取り除いてサービスを継続し得る高信頼化機構を具備しなければならない。このような交換機の高信頼化機構は既に空間分割電子交換機で確立している。しかしマルチプロセッサ制御方式においては構成装置数や装置種別の増大の問題がある。さらに単一故障の複数プロセッサへの波及による故障状態の複雑化や複数プロセッサに故障が現われた時の故障処理の競合などのより困難な問題の解決を図らなければならない。

またデジタル交換機では、「交換機の規模が増大するほどより高信頼でなければならない」という新たな規模別信頼度規定を設けている。ただし規模が大きくなってハードウェア量が増大すると通常は信頼度が低下する。デジタル交換機における信頼度規定はこの原則に反するものであり、この観点からは新たな高信頼化手段の考案が必要である。なお高信頼化の具体的実現に当たっても高信頼機構は正常時には必要の無いものであり、厳しい経済化が求められる。最小の機構で効率よく高信頼化を図らなければならない。

本章は、前章までで示したマルチプロセッサ基本設計結果に高信頼化機構を付与する考え方と、その設計について述べる。なお高信頼化には構成面の検討と回復処理に関する検討が必要であるが、本章は高信頼設計の導入部として、前者の構成面からの議論を行なう。後者の動的な故障への対処法、すなわち故障回復法は次章で詳細化する。

本章では、まずフェイルソフト思想に基づくマルチプロセッサ高信頼化の設計思想を述べ、次にこの考えに従ってマルチプロセッサ基本構成に予備装置を付与する。さらに、予備を含む高信頼構成に対して、フェイルソフトの実現条件と実現法を議論する。最後に、結果として実現したマルチプロセッサの信頼性を既存のシングルプロセッサ方式と比較評価する。

## 2. デジタル交換機の規模別信頼度規定

### 2.1 従来の信頼性規定の考え方

シングルプロセッサで制御された従来の電子交換機では、交換機に收容されている各加入者は交換機の規模によらず等しく高信頼性を享受すべきの考え方から各端子が一律に一定の信頼度という不稼働率規定を行なっていた。図5.1において、各端子には交換機の規模に係わらず、一定信頼度を与えている。例えば、初期の空間分割電子交換機では小規模から大規模まで全ての電子交換機に20年間に1時間の不稼働を許容し、この不稼働時間を電子交換機の各構成要素に振り分けることが行なわれてきた。

シングルプロセッサ方式では、構成要素がそのままシステムの共通要素であり、構成要素の不稼働はそのままシステムの不稼働になる。従って、この不稼働率規定はシングルプロセッサ制御方式にふさわしい規定でもあった。ただし、数100加入の極小規模の交換機でも、数万の加入者を收容した大都市の大規模の交換機でも不稼働率は同一ということから感覚的には矛盾をはらんでいた。

### 2.2 規模別信頼度規定<sup>(76)</sup>

デジタル交換機においては新たに規模別信頼度規定が導入された。大きな交換機の不稼働は小さい交換機の不稼働よりも社会に与える影響が大きいという考え方である。大きな交換機は小さい交換機よりも高信頼でなければならないとされる。

規模別信頼度の規定は具体的には以下のように規定される。Xアーランの交換を要する地域に処理量Xアーランの交換機を1台導入するときとX/nアーランの交換機をn台導入するときでは信頼度が同等になるべきとの考え方である。信頼度同等とはXアーラン規模の故障発生時に与える

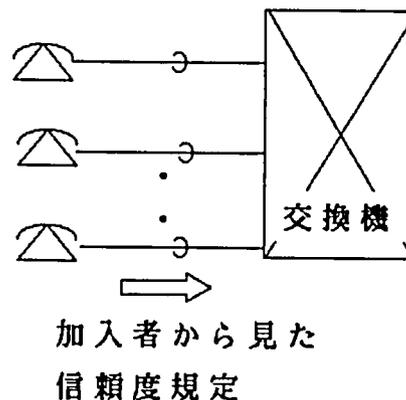


図5.1 交換機の信頼度規定

損失の期待値（損失の大きさ $L(X)$ と不稼働率期待値 $f(X)$ の積）が両構成で等しいとする。これは以下の式を意味する。

$$L(X) \cdot f(X) = n \cdot \{ L(X/n) \cdot f(X/n) \} \quad (5.1)$$

従って

$$\frac{f(X)}{f(X/n)} = \frac{n}{\frac{L(X)}{L(X/n)}} \quad (5.1')$$

この関係を満たす $f(X)$ は

$$f(X) \propto X/L(X) \quad (5.2)$$

と導出される。すなわち、不稼働率が呼量当りの損失に逆比例するように設計を行なえばよい。なお、 $L(X)$ は次の2者の損失の和で求まる。

- (i) 個々の加入者損失：呼量 $X$ に比例するものと考えられる。
- (ii) 電話が同時に不通となることにより与えられる迷惑度（これを社会的損失と呼ばれる）：不通となる加入者数（ $\propto X$ ）と円形を仮定した加入者収容域の半径（ $\propto X^{1/2}$ ）の積（ $\propto X^{2/3}$ ）に比例するものと考えられる。

これにより、 $C_1, C_2$ を常数として $L(X)$ は

$$L(X) = C_1 X + C_2 X^{2/3} \quad (5.3)$$

と求められる。ここで既存交換機の設計条件との整合を考慮して比例常数を定め、加入者線交換機には次式の不稼働率目標値を設定している。

$$f(X) = 3700 / (7 + X^{1/2}) \quad (\text{分}/20\text{年}) \quad (5.4)$$

この不稼働率目標値をハードウェア故障によるシステムダウンおよび保守者の誤操作やソフトウェアバグによるシステムダウンに配分し、交換機の設計指針としている。

### 3. 交換機制御用マルチプロセッサの高信頼化

#### 3.1 フェイルソフトによる規模別信頼度規定への対応

上記の交換機の規模別信頼度規定の考え方を制御系に実際に適応させることが本章での課題である。しかし幸いなことにマルチプロセッサ制御方式は規模別信頼度規定の考え方に整合しやすい性質がある。マルチプロセッサは複数プロセッサから構成されており、一部のプロセッサが故障しても全体に影響を与えないようにすれば、大規模全体ではより高信頼になる。

このような「一部に故障が生じても全体を不稼働にしない」というマルチプロセッサの高信頼化の考え方はフェイルソフト概念として知られている。たとえ、最大限のサービスの実行が出来なくとも段階的にサービスを縮小する考え方である。規模別信頼度規定に従う交換機制御用マルチプロセッサの高信頼化は、一般概念として存在するフェイルソフト概念を交換機制御用マルチプロセッサに具体化することにある。

#### 3.2 交換処理プログラム配置とフェイルソフト概念

交換機制御用マルチプロセッサでは複数のプロセッサに呼処理プログラムを配置し、それら全体が相互に結合して、交換機制御機能が完成する。制御主体はプログラムにあり、このプログラムを故障時にどのように配置するかによって具体的フェイルソフト手法が異なる。

故障時のプログラムの配置には図5.2に示すように (1)故障後のプログラム配置を故障状況に応じて配置替えを行なうプログラム再配置法と (2)故障の前後でプログラムの配置を変更しないプログラム固定配置法が考えられる。両方式の交換機への適応性を以下に評価する。

##### 3.2.1 プログラム再配置法

故障の状況に応じて最適構成、最高の処理能力を発揮し得る可能性があるプログラム再配置法はフェイルソフトの実現形態としてきわめて望ましい。交換機制御用マルチプロセッサにもプログラム再配置法によるフェイルソフトの実現が望まれる。しかし、再配置法の確立には任意のマルチプロセッサ構成における

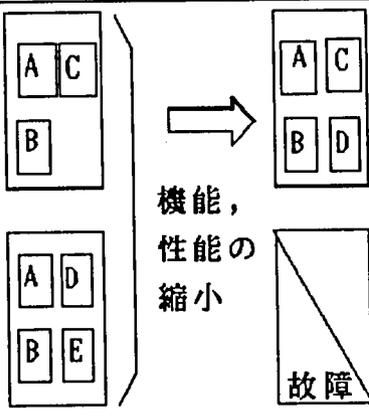
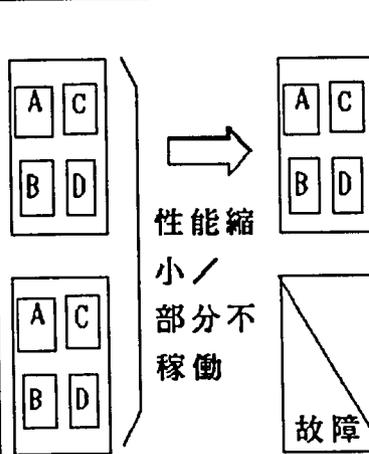
方式	備考
<p style="writing-mode: vertical-rl; position: absolute; left: -40px; top: 50px;">再配置法</p> 	<p>(1)フェイルソフト有効性</p> <ul style="list-style-type: none"> <li>○ 故障時も最大限のサービスを提供できる可能性がある。</li> <li>× ただし動的最適化は極めて困難である</li> </ul> <p>(2)適用対象親和性と実現性</p> <ul style="list-style-type: none"> <li>○ 適用対象は単純なマルチプロセッサ</li> <li>× プログラム構成上、動的配置に耐えるアドレス結合が必要（正常時にオーバーヘッドが大きい）。</li> </ul>
<p style="writing-mode: vertical-rl; position: absolute; left: -40px; top: 50px;">固定配置法</p> 	<p>(1)フェイルソフト有効性</p> <ul style="list-style-type: none"> <li>△ 固定化された範囲内で機能と性能を縮小。</li> <li>△ 一部の部分停止でサービス停止にならないよう負荷分散の範囲内が前提</li> </ul> <p>(2)適用対象親和性と実現性</p> <ul style="list-style-type: none"> <li>○ プログラム間を固定結合した既存の交換機プログラムに適合する。</li> </ul>

図5.2 フェイルソフト方式の対案

最適な系構成決定のアルゴリズムと任意のプログラム配置におけるプログラム間のアドレス結合の無矛盾性を保証しなければならない。

一方交換処理プログラムにはこれに適合しない以下の特徴がある。

(i) 交換機のプログラムはハードウェアの状態をプロセッサのメモリに写像して処理し、全体が結合してシステム全体で機能完結を図る。

(ii) ハードウェアの動作に追随して処理しうる厳しい実時間性を必要とする。

汎用情報処理のように人間に対する応答程度の実時間性や、各種の独立な処理の集合としての情報処理であればメモリ結合の単純なマルチプロセッサを用いてプログラムの再配置法による効果をあげることも可能であろう。しかしメモリ結合程度の単純なマルチプロセッサでは交換機の要求条件に適合しないことは第III章の議論からも明かである。また交換機における再配置法の困難性は第

III章での基本構成決定の議論からも推測がつく。各種条件を考察して初めて最適と考えられる系構成に到達した。

再配置法で、予め最適縮退系を求めておくのが困難であれば学習しながら最適化していく方法も考えられる。しかし、学習程度が低い段階では故障回復に時間を要する点も考えておかなければならない。また原理的に可能としても、実時間システムとしての時間範囲内で、また縮小による処理能力低下の中で最適化処理に処理リソースを費やすのは非現実的である。実時間システムとしてのプログラム再配置法は現状では実用の域に達せず、知識処理の応用分野として、今後の課題となる。

### 3.2.2 プログラム固定配置法

故障の前後でプログラムの配置を変更しないプログラム固定配置法は呼処理正常処理の実行の観点からは比較的簡単である。交換用プログラムには蓄積プログラム制御方式としてのプログラム固定性や、性能の向上を狙いとしたプログラムの固定配置とプログラム間の直接アドレス結合の特徴から交換機用故障処理法として適応性がある。従って交換機制御用マルチプロセッサのフェイルソフトはプログラム固定配置法に基づくことになる。

プログラム固定配置法では初期プログラム配置をそのままにして、ハードウェア/ソフトウェアを含めて不稼働部分を切り捨てることになる。フェイルソフトを実現するためには、システム内部に部分停止するところがあっても、これを除いてサービス機能の充足性を満たすことが必要となる。従ってシステムの他の部分に同一機能を具備している負荷分散の範囲がフェイルソフトの対象である。

### 3.3 フェイルソフト概念と負荷分散概念の対応化

交換機制御用マルチプロセッサのフェイルソフト手法はプログラム固定配置法に基づき、またプログラム固定配置法では負荷分散の範囲がフェイルソフトの対象となることが示された。ここではフェイルソフト実現に向けた負荷分散概念の整理を行なう。なお、議論するマルチプロセッサは、既に機能や性能の実現に向けて議論したように図5.3に示す信号処理と呼制御処理に機能分散したマルチプロセッサである。本マルチプロセッサでは、収容されている端子に固定的に

接続されている部分と、端子収容位置に関係なく処理を動的に分散する部分がある。このマルチプロセッサには、以下の2種の負荷分散の形態がある。

#### 静的負荷分散

交換機に収容されている全端子が個々のSGPに分散収容されている。すなわち、負荷が固定的、静的に分散されていると見なせる。SGPに故障が生じて、他のSGPに収容されている加入者線や中継線にはサービスを提供できる。故障したSGPは図5.2の部分不稼働に相当する。

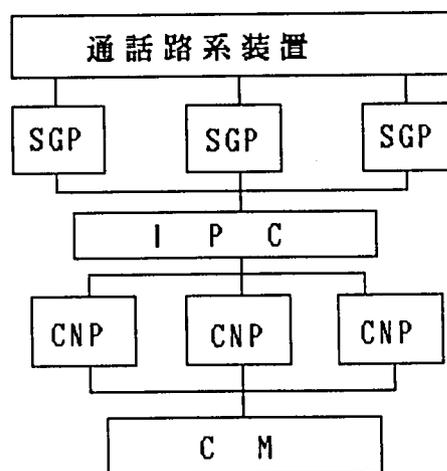


図5.3 対象とする構成

#### 動的負荷分散

いずれのSGPから処理を依頼されても、呼種を問わず、すべてのCNPは処理できる。CNPは一般的な負荷分散形態をとるが、ここでは静的負荷分散に対応させて動的負荷分散と区別する。CNPが故障しても、全端子に対して性能が低下するだけである。ただし、交換機は最繁時を想定した性能設計を行っており、しかも最繁時に相当する時間比率は大きくないことで、動的負荷分散にフェイルソフトを実現すれば顧客から見て、故障の影響が極めて少なくなる。

## 4. 予備装置配置

### 4.1 負荷分散概念に基づく予備装置配置

フェイルソフト思想に基づき、マルチプロセッサの各構成要素には図5.4に示す予備を配置する。

#### (1) 静的負荷分散の予備

負荷分散の単位ごとに、それぞれが故障対策を行なう。既に述べたが、SGPがこれに対応し、各SGPはそれぞれ二重化予備とする。

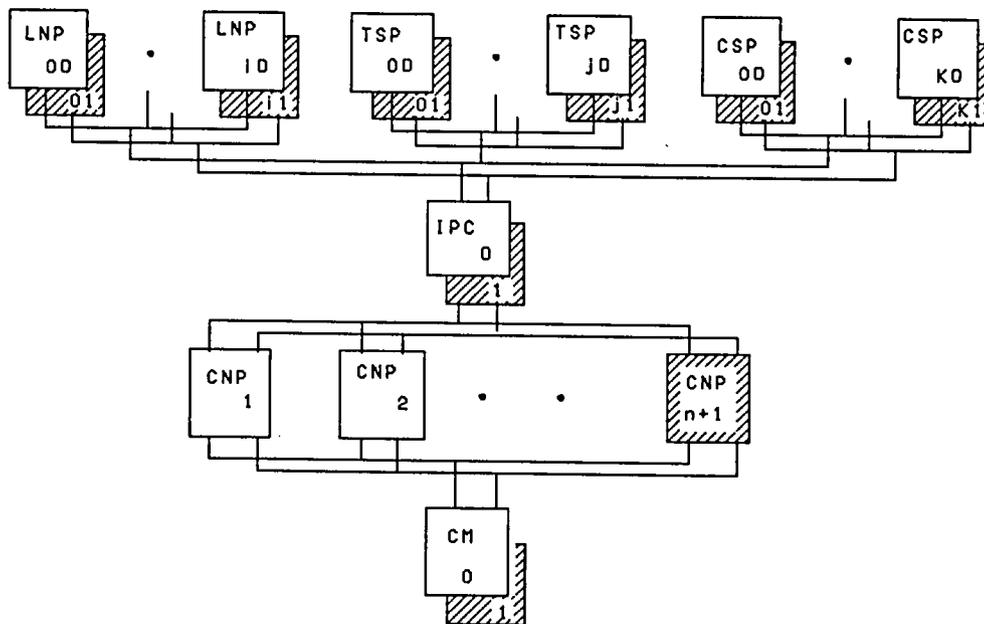


図5.4 マルチプロセッサの冗長構成

(2) 動的負荷分散の予備

CNPがこれに対応する。どのCNPも他のいずれのCNPの代替になりうることから、全体で共通に予備を持たば良い。CNPは経済的なN+1予備構成とする。

(3) システムの共通装置

CMやIPCはシステム全体に一組必要な共通装置であり、フェイルソフトの概念に対応しない。それぞれ最小の予備構成である二重化予備とする。

4.2 予備の待機状態

通常、故障の潜在化防止や故障回復の高速化を狙って、出来る限り熱予備形態をとる。二重化予備のメモリの熱予備はメモリからの読みだしは現用から、書き込みは同一内容を常に現用と予備の両系にして現用と予備の内容の一致を図る。現用系の故障時には単純に予備へ切り替えるだけで、故障前後の処理の連続性を保証できる。SGPの個別メモリや共通メモリがこれに当たる。

N+1予備をとるCNPは、予備を含めて全CNPが均質負荷分散で動作する。

## 5. フェイルソフト設計

### 5.1 フェイルソフトの実現条件

故障が生じたとき、故障装置を除いた縮退系においてもサービスを可能とするためには、以下の条件を満足しなければならない。

- (1)機能充足性：縮退系だけでサービスを実現できる機能を具備していることが条件となる。主にソフトウェアから見た条件を規定する。固定配置法では予めプログラムが配置されているものを用いるため、予め配置されているプログラムによってサービス機能の実現が可能かどうかは課題である。なお、固定配置法は予めシステム全体が結合していることから、縮退した生き残り部で機能を満たすと同時に、正常処理として切り放し部の影響がないことも確認しなければならない。
- (2)故障隔離性：故障の影響を排除して縮退系が動作可能であるかどうかは条件となる。たとえ(1)の機能が具備されていても故障を除去したはずの系で故障が再度検出されるとサービスの実行が不可能となる。なお、交換機は状態を保持するシステムであり、故障の隔離には以下の2種の問題を解決しなければならない。
  - (a)故障を除いて、サービスを再開したのに同一故障が原因となって再度故障現象が生じないようにしなければならない。
  - (b)故障発生時の影響が残存してサービス再開を不能とならないようにしなければならない。

### 5.2 機能充足性の考察

#### 5.2.1 縮退系と呼処理

##### (1)固定配置法における機能充足性

実現を目指すプログラム固定配置法では、故障で縮退系になってもプログラムによる機能補充はしない。正常時に配置された呼処理機能が縮退系でもサービスを実現しうるものでなければならない。

ところが既にのべた ONE-MACHINE 概念に基づく呼処理プログラムは固定配置法フェイルソフト手法との整合性がよい。第IV章で述べた ONE-MACHINE 概念に基

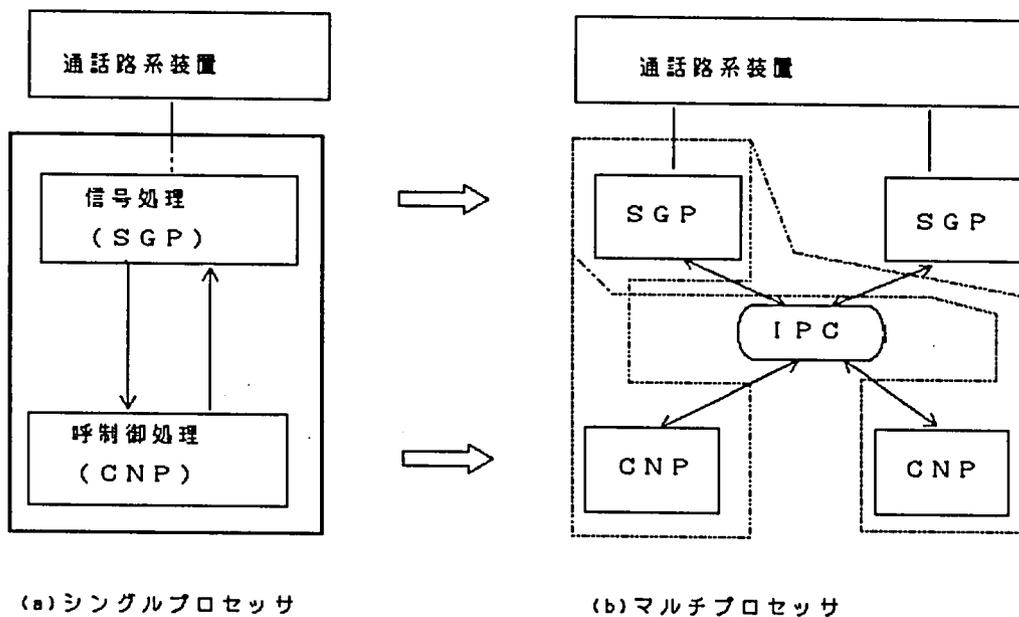


図5.5 呼処理プログラムの系構成等価性

づく呼処理プログラムは、SGPとCNPの任意の組合せがシングルプロセッサ機能を完結させる考え方で構成されている（図5.5）。この考え方によれば、SGPおよびCNPのいずれに故障を生じても残りのSGPとCNPの任意の組合せは、やはりシングルプロセッサとの等価性を有し、組合せ範囲での呼処理機能を完備している。

## (2) フェイルソフトと呼処理動作

呼処理は呼の状態、外部入力、対応する処理（タスクと呼び、この処理結果で呼の状態が遷移する）の3要素により表わされる。この状況を図5.6に示す。信号処理は、外部からの信号IN1が到達したとき信号内容を論理化し、関連する回線情報から呼と対応付けてCNPへ呼制御依頼を行なう。外部入力とは、発呼信号や応答信号や終話信号などの外部からの処理起動を起こす要因である。

CNPは、送られた情報に対応した呼の状態を共通メモリから読みだす。さらに、到達した信号と呼状態を対応させて、なすべきタスクを決定し、これを実行する。このタスクの実行結果として、SGPへの処理結果の返送や通話路の駆動命令や次の呼の状態が決まる。これでタスクの1サイクルが完了する。

ここで呼の状態とは、信号受信中や呼び出し中や通話中などの各呼が次の外部信号を待っている安定状態を言う。その実体は、呼処理のシーケンスの途中の

位置を示す呼ごとの制御データである。呼の状態は任意のCNPからアクセス可能な共通メモリに配置され、CNPが処理を完了する度に状態を更新し、共通メモリに退避される。

SGPからの出力は、直接他のSGPに流れることはなく、一部のSGPの不稼働が他のSGPに影響を与えることはない。すなわち、一部SGPが不稼働となっても残りの部分のSGPに関係した呼処理は完結する。

CNPについては、CNP間で同一処理の負荷分散をしているため処理依頼が不稼働CNPでなく、動作可能なCNPに渡ればよい。呼状態が任意のCNPからアクセス可能な共通メモリに配置されているため、任意のCNPで同一処理を実行できる。なお、1呼づつ呼の状態を共通メモリから読みだして処理するため、CNPが停止したとき罹障する呼は処理中の1呼に過ぎない。

### 5.2.2 不稼働部と呼処理

フェイルソフトを実行可能とするためには、切り放された不稼働部がシステムに影響を与えないことを保証しなければならない。ただし、不稼働に対する考

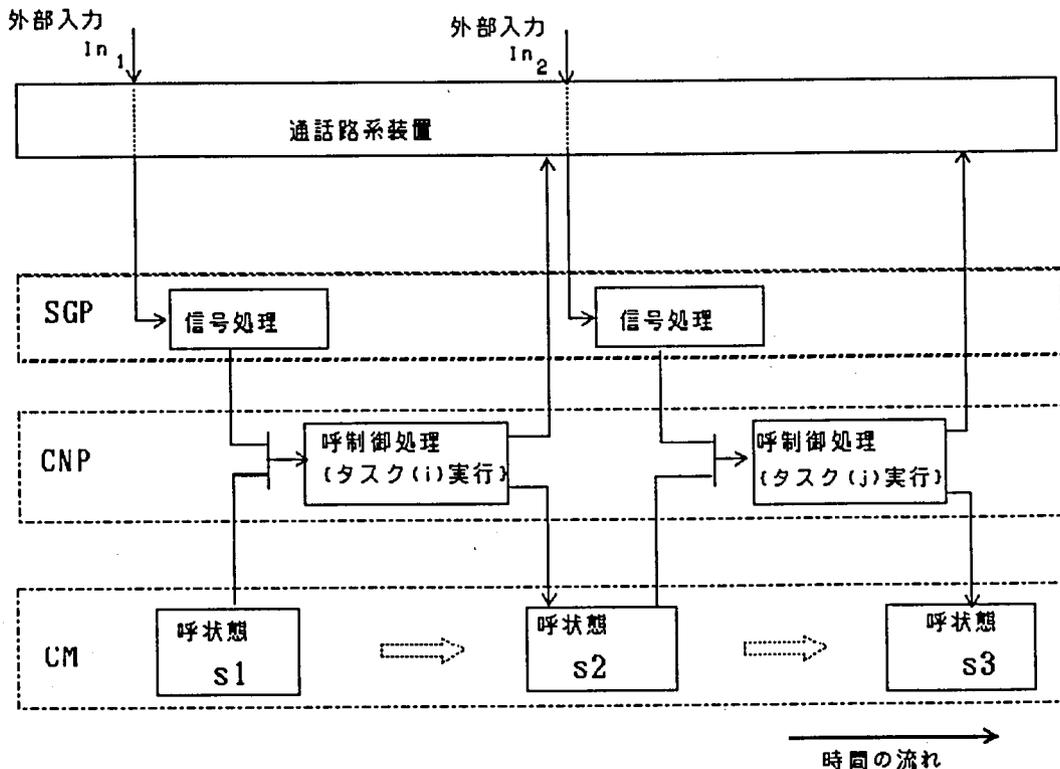


図5.6 フェイルソフトから見た呼処理動作

え方は、回線と固定的に接続されたSGPと、回線とは直接には対応しないCNPでは対処法が異なる。

#### (1)停止SGPへの対処

呼処理の特性から、SGPの停止に対しては、停止したSGPに関する呼のみに着目すればよい。SGPは、それぞれ接続されている回線に対する呼のみを処理し、他のSGPに接続されている回線は扱わない。また、複数SGPにまたがる呼はすべて一度CNPの処理を経由してから接続相手のSGPに制御が引き継がれる。

従って、発信側と着信側に対応して、それぞれ以下のように対処する。

##### 停止SGPからの発信

停止したSGPはもはや呼を生起することはない。停止しているSGPは閑散時と同様に、無発呼時と見なせば他に与える影響はない。

##### 停止SGPへの着信

故障により、停止の状態にあるSGPに收容されている回線は、すべてを使用中状態と見なせる。CNPのリソース管理において、停止SGPに所属する回線に対し、使用状態を表示すれば他のSGPからは回線使用中に見えるにすぎない。

#### (2)停止CNPへの対処

CNPは自ら呼を生起することはなく、SGPから処理依頼されたものに応答するだけである。停止CNPが処理依頼を受け、処理不能のまま処理依頼が蓄積されてシステムのデッドロックとなることを避けなければならない。

IPCの負荷分配機能を用いれば、停止CNPによるデッドロックは以下のように防止できる。CNPが空きであれば受信可能信号をCNPからIPCに発する。IPCは受信可能CNPを保持し、SGPからCNPへ処理依頼があれば、IPCが受信可能CNPのうちの一つを選んで送出する。停止CNPはソフトウェアの処理を伴う受信可能信号をIPCに送れないため、停止CNPに処理依頼が到達することはない。

### 5.3 故障隔離

故障の影響を排除できなければ、フェイルソフトの構成は可能でも縮退系における動作の継続はできない。ここでは故障隔離について述べる。故障の隔離は故障自身と故障の履歴の隔離の問題があり、それぞれ以下のように解決する。

#### 5.3.1 動的影響の排除

故障プロセッサが与える影響は故障プロセッサを停止させることにより防げる。これは既存のシングルプロセッサ制御方式の二重化予備制御で確立された考え方であり、技術である。マルチプロセッサにおいても、故障を検出したときは故障プロセッサを停止させ、他からの起動制御を受けない限り、故障プロセッサは自身で起動出来ないようにする。

故障プロセッサを停止させる技術も、またシングルプロセッサ制御方式で確立している。シングルプロセッサ方式と同様、緊急制御回路を設け、自己停止を実現する。各単位プロセッサはシングルプロセッサとして各種故障検出回路を具備し、暴走検出手段としての監視タイマを持つ。

ただし、マルチプロセッサでは複数プロセッサ制御の特徴を活かし、故障プロセッサ以外は停止や初期設定は行なわない点が異なる。このため故障履歴が縮退した非故障プロセッサ群に影響を与えないようにしなければならない。この問題は故障履歴の排除として扱う。

#### 5.3.2 故障履歴の排除

マルチプロセッサでは、プロセッサに故障が生じてても、極力全体の初期設定を行なわないようにし、故障の影響を極力少なくすることとしている。このため故障の履歴は速やかに消去する必要がある。また、次章で述べる動的な故障処理、すなわち故障回復手順でプログラムの処理を継続するためには、システム全体で複数の故障現象が生じないことを前提にしている。このため単一の故障であれば極力故障の影響を他に波及させないことが必要である。

##### (1) S G P 故障の履歴排除

S G P は多数の回線を収容しており、呼を保持したまま停止に至る危険がある。

履歴排除や通話リソースの解除の観点から、停止し、無効となった呼を解除しなければならない。非故障部の全初期設定を行なえば問題がないが、マルチプロセッサでは故障SGP以外は呼処理を継続するとしている。

停止したSGPの呼の履歴では、CNPや共通メモリに残された対応する呼の履歴が問題となる。この履歴はCNPが故障SGPとの対応から徐々に解除する。第IV章で述べたように、回線の管理情報はプロセッサ番号を対応付けており、呼とSGPの対応がとれる。CNPはCM上の呼状態を識別し、停止したSGPに関する呼を解放する。

## (2) CNP故障の履歴の排除

CNPは処理対象の呼情報のみを保持しており、CNP故障により罹障するのは単一呼のみである。この罹障した単一呼に対しては、処理を依頼したSGP側で応答がない等による判断ができ、再度処理要求の送出などで救済が可能である。

一方、CNPには呼処理の過程で交換機全体のリソースをアクセスする問題がある。リソースアクセス中に故障が生じるとリソースを凍結したままCNPが停止し、システムのデッドロックに至る。CNPにおける故障履歴排除には故障によるリソースの凍結を防止することが必要となる。

故障プロセッサがリソースをアクセスしている状態には2状態ある。共通リソースを確保しようとして ①共通リソース全体にロックをかけている状態（ロック機構の凍結）と ②自分に必要なリソースを確保してしまった状態である。それぞれ以下のように対処する。

### (a) ロック機構凍結防止

負荷分散を行なうシステムでは、複数プロセッサ間の競合を排除する排他制御ロック機構が必要であるが、故障時にロック機構を設定したまま停止した場合の対策が必要である。もし、ロックをかけたままプロセッサが故障になるとロックにより、競合データが保持されたままになり、他のプロセッサがそのデータにアクセスすることが必要になってもアクセスできない。結局システムはデッドロック状態になり、故障からの回復が出来ないことになる。

共通メモリでの排他制御に用いられるロックフラグは共通メモリ上の競合処理を必要とする共用データにアクセスするとき設定され、そのデータへの処理が

完了したとき解除される。故障プロセッサによって設定された競合データロックフラグを解除するためには、競合データにどのプロセッサがアクセスしているかを識別できることが必要である。このため、ロックフラグの設定と同時にプロセッサ番号を保持する機構を設ける(図5.7)。負荷分散を行なっているプロセッサに故障が生じたとき、システム管理を司る特定のプロセッサMCPがこの故障を認識し、ロックフラグを強制的に解除する。

#### (b) リソース保持中の呼の解除

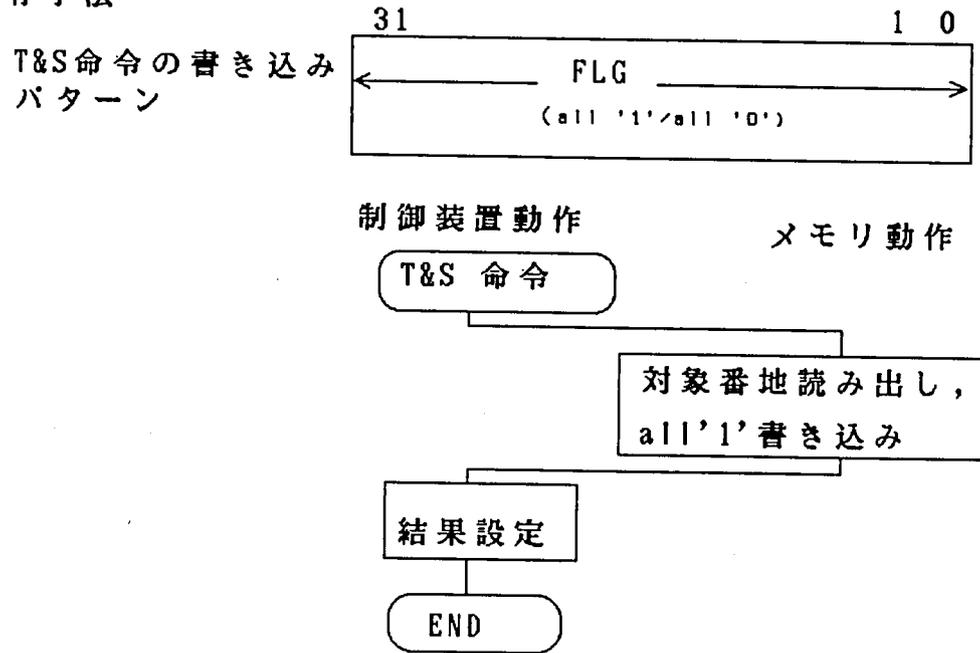
CNPが故障したとき、処理していた呼は自用リソースを確保したままの状態に凍結されてしまう可能性がある。しかし、この場合も処理を依頼していたSGPは正常に動作していると想定される。SGPは時間監視等により、異常を認識し、異常な呼の処理に対しては、CNPに解除信号を送出する。解除信号に対してはCNPは高レベルで必ず返答を送ることとしており、返答の無い場合には再度解除信号を送出することにより、他のCNPが罹障呼の解放、リソースの解放を行なう。

#### (3) 故障波及に対する防御

今まで故障が明らかになったときの議論をおこなった。マルチプロセッサでは、複数プロセッサが連携して動作するため、故障を認識する以前に異常な情報が正常なプロセッサに波及する可能性がある。

並列処理を行なっているマルチプロセッサで、他のプロセッサからの異常信号を防ぐには通信情報の論理的合理性による検証が有効である。論理的合理性検証は通常の呼処理の中で実行する。呼処理に伴うプロセッサ間の通信は呼処理の流れに従った信号を送出してくる。他から到達する入力、その呼の状態に適合する信号かどうかを常に判断することにより信号の妥当性が検証できる。例えば、トランクの状態が空きの時、TSPから発呼以外の入力がCNPに送られてくれば、異常であり、この信号を送ってきたTSPに切断を行なうよう指令する。

(a) 既存手法



(b) CNP番号表示方式

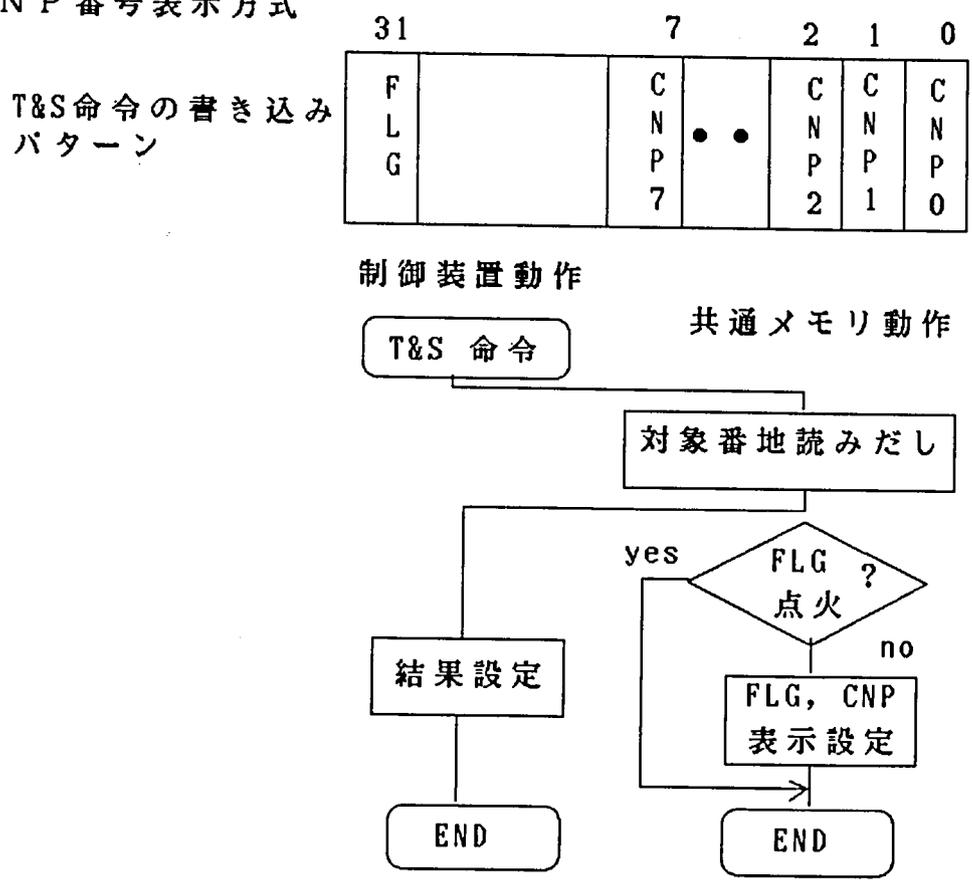


図5.7 実時間・高信頼システム用マルチプロセッサのT&S命令

## 6. 評価

本節ではマルチプロセッサの高信頼化の効果について、構成面から評価する。本マルチプロセッサは既存シングルプロセッサ方式を改良することが基本的考え方であり、評価にあたってはシングルプロセッサ制御方式と対比しつつ議論する。また、デジタル交換機では規模別信頼度を規定しており、マルチプロセッサの規模別信頼度の規定への適合性についても述べる。

### 6.1 評価の前提

シングル/マルチプロセッサ両構成の信頼度を比較評価するため、両構成を図5.8のようにモデル化する。各装置の不稼働率 $U_i$ はMTTR（平均修理時間）とMTBF（平均故障間隔）を用いて、 $U_i = \text{MTTR}_i / (\text{MTTR}_i + \text{MTBF}_i)$ で表わされる。しかし、MTTRはMTBFよりきわめて小さく、 $U_i = \text{MTTR}_i / \text{MTBF}_i$ と近似できる。また、交換機の故障修理時間の支配項は、保守者の駆けつけ時間やハードウェアの取り替え時間である。診断プログラムなどのツールを利用するため、装置や構成による差は小さい。このため、各装置ごとの不稼働率 $U_i$ は、それぞれの故障率 $F_i = 1/\text{MTBF}_i$ に比例すると仮定できる。共通のMTBFを $T_0$ とすれば、 $U_i = T_0 \cdot F_i$ である。

また、本評価は厳密な数値を求めるのではなく方式の違いによる信頼性の傾向を示すものであり、この範囲で以下の単純化を行なう。

#### (1) マルチプロセッサ用単位プロセッサの故障率

マルチプロセッサには、すべて同じ種類のプロセッサを用いる。従って、CNP、SGPのそれぞれ単体の不稼働率 $U_{\text{CNP}}$ 、 $U_{\text{SGP}}$ は同一である（ $U_{\text{CNP}} = U_{\text{SGP}}$ ）。

#### (2) 共通装置の不稼働率

共通装置IPCとCMの単体の不稼働率、 $U_{\text{IPC}}$ 、 $U_{\text{CM}}$ は、両者のハードウェア規模がほぼ同等であり、以下のように表わす。（ $U_{\text{IPC}} = U_{\text{CM}} = r \cdot U_{\text{CNP}}$ ）

#### (3) SGPとCNPの呼処理分担率

SGP、CNPの呼処理機能分割は、呼当りの処理量で換算すると同程度であることから、両者のプロセッサの数は予備を除き、それぞれ $n$ 個とする。

(4) プロセッサの性能と不稼働率

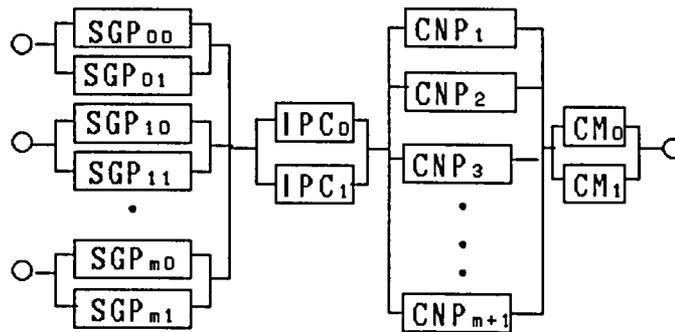
プロセッサの平均故障率には、以下の理由から計算機における性能とコストの関係でよく引用されるグロッシュの法則を引用して議論する。すなわち、プロセッサの平均故障率はその構成部品の故障率に支配され、高速なプロセッサほど発熱量が増大（部品低信頼化）し、低集積である高速部品を必要とすることから、性能と故障率の関係は、性能とコストとの関係と同様の傾向をとる。従って、プロセッサPRCの性能（単位時間あたりの平均命令の実行数） $P_{PRC}$ と、このプロセッサの平均故障率 $F_{PRC}$ は

$$F_{PRC} = a_0 (P_{PRC})^q \quad (5.5)$$

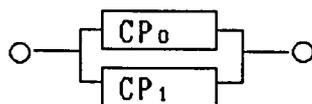
( $a_0, q$ : 定数.  $q=0\sim 1$ . グロッシュの法則では $q=1/2$ )

となる。

マルチプロセッサの最大規模（このときのCNPの台数を予備を除き、 $M$ とす



(a) マルチプロセッサ構成



(b) シングルプロセッサ構成

図5.8 信頼性評価ブロックダイヤグラム

る) 相当のシングルプロセッサでは

$$\begin{aligned}
 F_{CP} &= a_0 (2M \cdot P_{CNP})^q \\
 &= (2M)^q \cdot F_{CNP} = (2M)^q \cdot F_{CNP}
 \end{aligned}
 \tag{5.6}$$

$$U_{CP} = (2M)^q \cdot U_{CNP}
 \tag{5.6-1}$$

## 6.2 不稼働率の算出

### 6.2.1 システム構成要素の不稼働率

マルチプロセッサ、シングルプロセッサ構成のそれぞれの不稼働率UASを算出する。また、システムの不稼働率は故障波及度 S を用いて、サービス停止範囲を明確にして表わす。なお、S とはサービス停止を伴う範囲をプロセッサの等価的能力 (1 台のプロセッサが等価的にカバーする通話路容量) で正規化した値である。すると、構成要素ごとの不稼働率UASは近似的に以下のように表わせる。

#### (1) マルチプロセッサの場合

$$\begin{aligned}
 UAS_{SGP} &= {}_m C_1 (U_{SGP})^2 & (0 < S \leq 1) \\
 &= 0 & (1 < S \leq m) \\
 UAS_{CNP} &= {}_{m+1} C_2 (U_{CNP})^2 & (0 < S \leq 1) \\
 &= 0 & (1 < S \leq m) \\
 UAS_{IPC} &= (U_{IPC})^2 & (0 < S \leq m) \\
 UAS_{CM} &= (U_{CM})^2 & (0 < S \leq m)
 \end{aligned}
 \tag{5.7}$$

#### (2) シングルプロセッサの場合

$$UAS_{duo} = (U_{CP})^2, \quad (0 < S \leq m)
 \tag{5.8}$$

表5.1 罹障規模と相対不稼働率

	罹障規模	0 < S ≤ 1								1 < S ≤ m
	交換規模	m=1	m=2	m=3	m=4	m=5	m=6	m=7	m=8	
マルチ プロセッサ		3.5	8.5	15.5	24.5	35.5	48.5	63.5	80.5	0.5
シングル プロセッサ	q=0	1								
	q=1/2	16								
	q=1	256								

なお、近似的とは、2乗以上の項はSGPでは4台以上のプロセッサの同時故障（4乗）、CNPでは3台のプロセッサの同時故障（3乗）とオーダが大きく異なることから省略している意味である。

ここで、(5.7)、(5.8)を $U_{CNP}$ で統一し、 $(U_{CNP})^2$ で正規化して、マルチプロセッサ、シングルプロセッサ方式の不稼働率を求めると

$$\begin{aligned}
 UAS_{multi} / (U_{CNP})^2 &= (UAS_{SGP} + UAS_{CNP} + UAS_{IPC} + UAS_{CM}) / (U_{CNP})^2 \\
 &= m + m(m+1) + (r)^2 + (r)^2 \quad (0 < S \leq 1) \\
 &= (r)^2 + (r)^2 \quad (1 < S \leq m) \quad (5.9)
 \end{aligned}$$

$$UAS_{duo} / (U_{CNP})^2 = (2M)^{2q}, \quad (0 < S \leq m) \quad (5.10)$$

シングルプロセッサの能力をマルチプロセッサの最大規模（M=8）に設定し、罹障規模と相対的不稼働率の関係を求めたものを表5.1に示す。

この表から以下を結論とする。

- (1) マルチプロセッサでは罹障規模がビルディングブロック単位（Sが1以下）に閉じる範囲では不稼働率は大きい、罹障規模が複数ビルディングブロック単位にまたがる規模ではきわめて小さい。規模別信頼度規定への適合性が示されている。
- (2) マルチプロセッサでは小規模の範囲での不稼働率はむしろシングルプロセッサ

より、増大する傾向にある。機能分割により、CNPとSGPの故障が直列化したもので、小規模では負担となるが、大規模では分割により影響が小さくなる。規模別信頼度規定の方針に基づくオーバーヘッドと見なすべきであろう。

(3)マルチプロセッサでは規模が大きくなると $S \leq 1$ の範囲での不稼働率が極端に増大するように見える。この問題は後に議論するように、安全側の評価に基づくものであり(後注)、また、マルチプロセッサ制御方式交換機に收容された加入者の故障が、シングルプロセッサ制御方式交換機に收容された加入者の故障よりも増大するものでもない(6.2.2で評価)。

(4)シングルプロセッサではプロセッサの性能と故障率の間に大きな関係が無ければ(例えば $q=0$ )、むしろ高信頼といえる。しかし、現実としてはプロセッサの性能と故障率は大きな相関関係を持ち、 $q$ は $1/2$ 以上と考えるべきであろう。

また、 $m=8$ に相当する大処理能力のプロセッサを実現することも不可能である点についても留意しなければならない。

(注)マルチプロセッサの $0 < S \leq 1$ の範囲における部分不稼働の大部分はCNPの不稼働によるものである。この評価では固定的にCNPが処理を受持ち、CNPの停止がそのままサービスの停止を招くとした安全側の評価を行なっている。しかし、 $m$ が増大するほど不稼働率が増えるが、実際はCNP全体で動的負荷分散を行なっているため、CNP1台の停止の影響は $1/m$ と逆に小さくなる。さらにCNPの数 $m$ の増大は、呼制御処理のサーバの増大を意味し、大群化効果で影響は更に少なくなる。

#### 6.2.2 收容端子から見た不稼働率

既に述べたシステムの規模別信頼度への適合性の評価に対し、ここでは個々の加入者から見たマルチプロセッサの信頼性をシングルプロセッサと比較して評価する。システムの不稼働率の評価において、小規模な単位の不稼働率がマルチプロセッサではむしろ増大していることが示されている。小規模な範囲での不稼働率の増大がマルチプロセッサに收容されている加入者から見てサービス低下

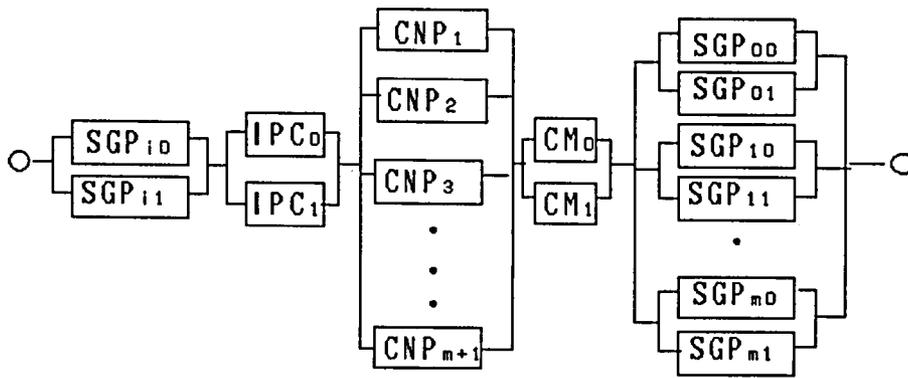
にならないかどうかを吟味する必要がある。

収容された端子から見た不稼働率を求める評価モデルを図5.9に示す。あるSGPから発呼された呼がいずれかのCNPで呼制御処理され、元のSGP、または他のいずれかのSGPに着信するとき、呼が不稼働に遭遇する確率を求める。

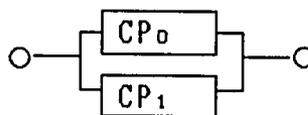
収容された端子から生起する呼が不稼働に遭遇する確率、すなわちマルチプロセッサの端子当りの不稼働率 $U_{Multi}$ は以下のように表わされる。

$$U_{Multi} = U_{SGP,0} + U_{IPC} + U_{CM} + R_{CNP} \cdot U_{CNP} + R_{SGP,1} \cdot U_{SGP,1} \quad (5.10)$$

ここで $U_{SGP,0}$ は対象とする端子が収容されたSGPの不稼働率、 $U_{SGP,1}$ は対象とする端子の収容されたSGPが稼働中で、発呼した呼が着信するSGPが不稼働となる確率である。また、 $R_{CNP}, R_{SGP,1}$ は負荷分散されたCNP, SGPが注目する端子に与える影響係数である。



(a) マルチプロセッサ構成



(b) シングルプロセッサ構成

図5.9 信頼性評価ブロックダイヤグラム

表5.2 端子から見た不稼働率

交換規模		m=1	m=2	m=3	m=4	m=5	m=6	m=7	m=8
マルチ プロセッサ		2.5	3.5	4.2	4.8	5.3	5.8	6.4	6.9
シングル プロセッサ	q=0	1							
	q=1/2	16							
	q=1	256							

上式に対し、厳密には高次の項を含まなければならないが、有効値を2次の項で統一すると

$$\begin{aligned}
 U_{A_{multi}} = & (U_{SGP})^2 + (U_{IPC})^2 + (U_{CM})^2 + {}_{m+1}C_2 (U_{CNP})^2 / m \\
 & + {}_{m-1}C_1 (U_{SGP})^2 / m
 \end{aligned} \tag{5.11}$$

なお、 $U_{xxx}$ は6.2.2で用いた各装置単体の不稼働率である。また、 $CNP$ 、 $SGP$ は $m$ 系統のうち、1系統の故障であり、故障影響度は $R_{CNP} = R_{SGP.1} = 1/m$ である。

シングルプロセッサでは

$$U_{A_{dup}} = (U_{CP})^2 \tag{5.12}$$

これらに6.1の前提条件を適用し、 $U_{CNP}$ で正規化すると

$$\begin{aligned}
 U_{A_{multi}} / (U_{CNP})^2 &= 1 + r^2 + r^2 + {}_{m+1}C_2 / m + {}_{m-1}C_1 / m \\
 &= 2r^2 + (m^2 + 5m - 2) / (2m)
 \end{aligned} \tag{5.13}$$

$$U_{A_{dup}} / (U_{CNP})^2 = (2m)^{2q} \tag{5.14}$$

$IPC$ 、 $CM$ の $CNP$ に対するハードウェア規模から、 $r=0.5$ とし、(5.13)と

(5.14)を数値計算した結果を表5.2に示す。プロセッサの性能が向上しても信頼度が低下しないとすれば( $q=0$ )、シングルプロセッサの方が高信頼である。ただし、6.2.1において既に述べたが、現実にはこの仮定はありえない。また、シングルプロセッサの最大規模をシングルプロセッサで実現できない。 $q=1/2\sim 1$ と考えられこの範囲で表5.2を見ると以下の点が結論つけられる。

- (1)マルチプロセッサでの端子当り不稼働率は小規模から大規模へ行くほど増大する。ただし、その増大化傾向は大きくない。
- (2)シングルプロセッサの不稼働率は交換機の最大規模を実現するプロセッサによって規定され、小規模の交換機でも不稼働率は高い。
- (3)端子当り不稼働率はマルチプロセッサの方が低く、影響範囲を限定するマルチプロセッサ構成は信頼性を向上させている。

マルチプロセッサの信頼度は、現実にはさらに向上する。上の評価は不稼働の状態にあるCNP、SGP(着信)に無条件に遭遇する確率を求めたが、以下の点から、マルチプロセッサには厳しすぎる評価と言える。CNPでは動的に負荷分散処理をしているため、通常、呼は不接とならず、最繁時に接続遅延時間が増大する程度である。また、着信SGPでは中継線は複数のSGPに分散收容され、最繁時に回線話中に遭遇する確率が増大するだけである。1日のうち、最繁時は比較的短く、最繁時をはずせば、負荷分散の範囲では故障の影響はなく、故障に遭遇する確率はさらに低下し、信頼性はさらに向上していると思われる。

## 7. 結言

マルチプロセッサ制御方式は規模比例のコスト、大処理能力、処理能力拡大性など積極的な面では有効と評価されながら、高信頼化ではプロセッサ間の制御競合、構成の複雑性などによる困難性が指摘され、採用が手控ええられる傾向にあった。本章ではフェイルソフト概念を交換機制御に適用する考え方からマルチプロセッサ制御方式の高信頼化の考え方と実現法を示した。その内容は以下のよ

うに要約される。

#### (1)規模別信頼度規定への対応

社会的迷惑度の尺度に基づく規模対応の信頼度規定に対し、マルチプロセッサ制御の特徴を活かした高信頼化の考え方を示した。マルチプロセッサの構成要素を小規模交換機と見なす考え方から、故障影響範囲を局在化し、マルチプロセッサで実現する大規模交換機でのシステム全体の不稼働を避ける方法を示した。

#### (2)交換機制御システムへのフェイルソフト概念の適用

システムの一部の不稼働をシステム全体の不稼働としないフェイルソフトの考え方を実時間システムに適用する考え方を示した。フェイルソフトを実現する故障時のプログラムの配置には動的配置と固定的配置があることを示し、交換機には固定配置がふさわしいことを示した。また、固定配置法においてフェイルソフトを実現するには負荷分散の考え方が必要なこと、また、故障の影響として部分不稼働と処理能力低下の現象があることを示した。

#### (3)フェイルソフト手法の具体化

フェイルソフト実現の条件として故障を除外した系構成での機能充足性と故障隔離性が必要であることを示した。特に、故障隔離性において、システムデッドロックを防ぐ諸対策を示した。

#### (4)評価

フェイルソフトの実現による高信頼効果を既存のシングルプロセッサ制御方式と比較して、評価し、ここで実現したマルチプログラム高信頼化手法が規模別信頼度規定の考え方に合致した方式であることを示した。また、マルチプロセッサ高信頼化の実現により、既存のシングルプロセッサ制御方式に対し、より高信頼化が図れたことを示した。

## 第VI章 デジタル交換機制御用マルチプロセッサの故障回復法<sup>(77)(78)</sup>

### 1. 序言

本章では、前章で示した高信頼構成を有効化する手順すなわち故障処理について述べる。故障処理は多数ある系構成の組合せの中から故障装置を除外した系構成を選定して、動作可能な系構成を確立し、呼処理プログラムを再起動し、サービスを再開させるものである。故障からの回復の確実性ととも、サービス中断時間を極力短くする観点から短時間で故障処理を完了することが求められる。

故障処理は既にシングルプロセッサ制御方式で確立し、実績ある技術である。しかし、基本的考え方は利用できるとしてもマルチプロセッサ制御方式には以下のような新たな要因があり、マルチプロセッサ用の故障回復手法を新たに開発しなければならない。

- (1) マルチプロセッサでは故障による誤った情報が他のプロセッサに波及する可能性があり、故障現象が複雑化する可能性がある。二重化予備のシングルプロセッサ制御方式のように故障が生じたとき単純に予備に切り替えるといった処理ではすまない。
- (2) 複数プロセッサが並行に動作しており、複数のプロセッサが互いに無秩序に故障回復の系構成の指令を出す可能性がある。これにより故障回復を図るプロセッサ間で競合を生じ、結果として誤った系構成に到達する可能性がある。
- (3) 本マルチプロセッサではシングルプロセッサを改良することを狙っている。故障からの回復は両方式で同様に確実としても、故障処理の中断時間をシングルプロセッサ方式より短くすることが必要である。

本章では故障回復を確実化し、かつ故障回復に伴うサービス中断を極力少なくして、加入者に与える影響を少なくするマルチプロセッサ用故障回復手法を明らかにする。まず、故障処理の設計思想を述べ、故障回復動作を具体的に説明して故障回復の確実性を示す。次に故障原因、故障頻度をモデル化して定式化、定量化して故障回復による中断時間の影響を既存のシングルプロセッサ構成と比較評価する。

## 2. マルチプロセッサの高信頼化

### 2.1 高信頼構成の前提

故障回復処理の議論に先立ち、前提とするマルチプロセッサの高信頼構成を本章で確認する。前提とするマルチプロセッサは第IV章までの基本設計、第V章の高信頼構成の検討において予備装置の付与を行なった図6.1のマルチプロセッサである。SGP群は通話路装置に直接結合され、通話路装置群に合わせた二重化予備である。CNP群は特定の通話路装置に従属せず、いずれも他の代替になりうる均質な処理を分担するもので、経済的なN+1予備構成である。CMやIPCはシステムに一組必要な共通装置であり、二重化予備である。これらの二重化予備のメモリへは通常同一内容を同時に書き込んでおり、熱予備である。SGP内の個別メモリは予備へ切り替え後直ちに通話路の状態を回復するため、CMは故障前後の通話中呼の連続性を保つために利用する。

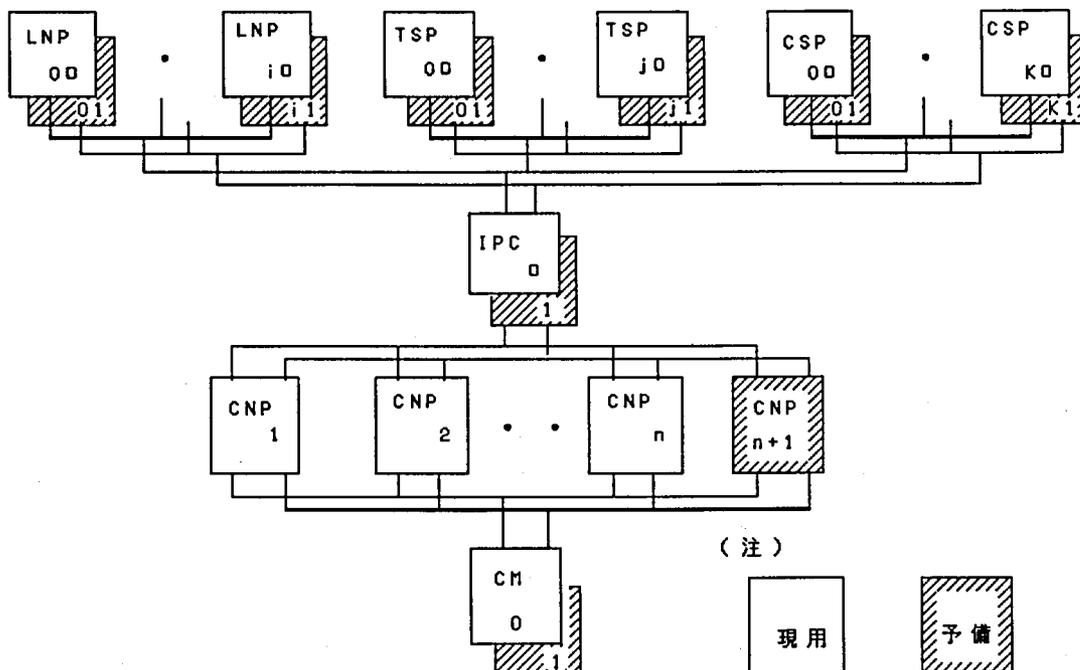


図6.1 マルチプロセッサの冗長構成

## 2.2 故障処理の遷移

故障処理は図6.2に示すように ①故障が生じたとき故障処理の動作を起動するための故障検出, ②故障装置を認識する故障識別, ③故障装置を除いた系を確定するための系再構成, ④ファイル装置から主記憶装置への呼処理プログラムの読みだしやデータ類の初期設定およびハードウェアの初期設定を行なう再開処理からなる。このうち①故障検出, ②故障識別, ③系再構成がサービスを行い得る正常な系を確立するための基本機構である。故障処理の確実化にはまずこの故障処理の基本機構を完成しなければならない。

## 3. 階層化思想に基づく故障処理

本節では複雑な系構成, プロセッサ間の競合など多くの問題を抱えるマルチプロセッサの故障処理について, その設計指針を述べる。

### 3.1 故障処理の階層化

一般に複雑なシステムの設計では実現すべき手順を階層化して整理し, 上位階層は下位の階層の機能の組み合わせで実現し, 極力思考の単純化や設計工数の削減や設計の高信頼化を図ることが行われる。

ここでも故障処理を階層化して構成し, 複雑化し易いマルチプロセッサの故障処理を単純化し, 高信頼化する。

なお階層構成を具体化するための基本方針が必要であるが, ここでは以下の3方針に基づ

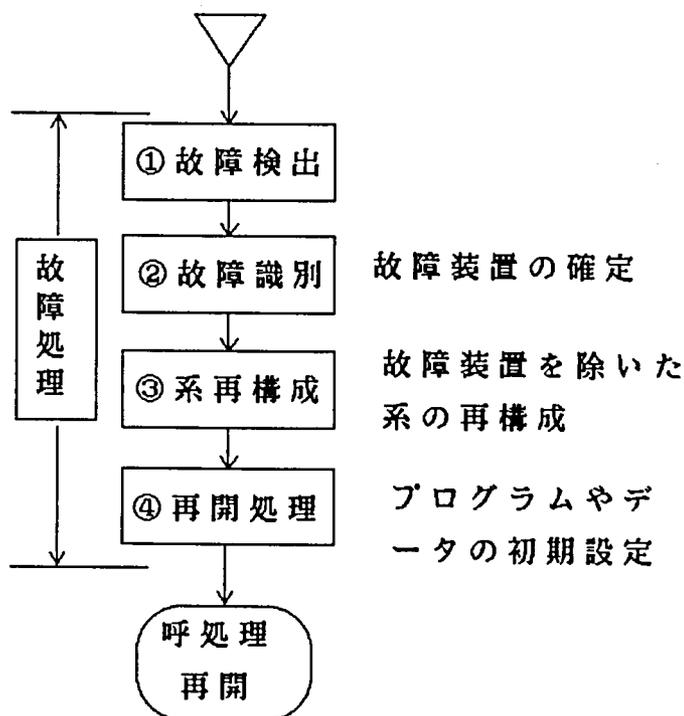


図6.2 故障処理の遷移

き具体化を図っている。

### 3.1.1 故障処理の局所化

故障回復のアルゴリズムの単純化や回復処理の迅速化を図るためには被疑故障範囲を広げず、極力小規模な単位で故障検出を行ない処理を行なうことが有効であり、階層化思想の基本である。このためマルチプロセッサでは故障波及により故障現象が複雑化する可能性があるが、まず故障原因と故障現象は対応付けて現われるとし、故障が生じる度に故障現象に応じた故障処理を局所的に行なう。早期に故障が検出されるものが故障の大部分であり、局所処理により解決が図れば故障の影響範囲が少なくすむ。故障がメモリ汚染などによって論理的に波及してもそれぞれのプロセッサごとに初期設定すれば故障からの回復が可能な場合もある。局所処理によっても解決できない場合は次項の指針により解決する。

### 3.1.2 系内処理から系外処理へ

大規模システムでは、あらゆる場合について故障処理の确实性の検証を行なうことは不可能であり、故障を生じた系内での論理による故障処理には限界がある。このため上記の局所化思想から故障を検出した系内での故障処理を優先しつつ、系外からの監視と援助を行なう。例えば個々の負荷分散モジュール（二重化されたSGPでは二重化の組合せが、N+1予備構成をとるCNPでは個々のCNP）内での処理をまず優先するが、モジュール内に閉じて解決不可能な場合は段階的に故障処理の拡大を図る。最終的には本来遂行すべきサービスすなわち呼処理の确实性を系外で独立に監視し、異常時には系外から故障処理を起動する。

### 3.1.3 ソフトウェア処理からハードウェア制御へ

故障回復の确实化には故障処理用制御機構（故障回復のハードコア）を确实化することが必須である。この故障処理用専用制御機構をハードウェア論理で実現することが一案であるが、マルチプロセッサでは多数の装置で構成され、組合せ数が多いため制御論理の複雑化を招く。またハードウェアを大規模化させる。従ってマルチプロセッサではこのような専用機構を設けることはむしろ信頼性を低下させることになりかねない。

マルチプロセッサの故障処理には極力プログラム論理で解決を図り、プログラム論理で解決できない場合にはハードウェア論理を併用することが有効である。故障処理機構の簡単化のためには、さらに故障処理のプログラムの実行も呼処理と同じプロセッサを用いることが有効である。しかしこの場合、故障からの独立性、すなわち確実性は保証が出来ないため、これを補償する機構が必要である。これには故障処理プログラムは故障でないプロセッサにより実行する考え方をとる。プログラムが正常に走行しているかどうかの検出は、たとえば監視タイマ (Watch dog) 等の単純な独立機構で実現できるため故障が検出されると異なったプロセッサに故障処理プログラムの実行プロセッサを変更して再度故障処理を行なう考え方である。ハードウェア制御機構の補助により故障のないプロセッサが存在する限り故障処理を正しいプロセッサで実行できる。

### 3.2 故障処理の階層構成

前節の指針に従って以下の4階層で故障処理を実現する。また、故障報告および故障回復制御の経路を図6.3に示す。

#### 3.2.1 局所故障処理階層

この階層ではモジュール内に閉じた故障処理を行う。図6.3においてCNPおよびLNP, TSP, CSP中での故障処理である。負荷分散の単位であるSGP (LNP, TSP, CSPの総称) モジュールでは二重化構成、CNPモジュールでは各1台のプロセッサ内の処理であり、モジュール内の閉じた故障処理は故障を検出してもマルチプロセッサを意識することなく、実績を積んだシングルプロセッサの故障回復技術が利用できる。

二重化予備のSGPは熱予備運転形態をとるため単に予備への切替えを行う。すでにメモリが汚染しているなどでプログラムが正常に走行できない場合、予備切り替えだけでは回復が不可能である。このような場合には現用と予備を共に停止して、上位に停止したことを報告する。

N+1予備構成をとるCNPでは、予備を含めて全装置を通常は運用状態にしておき、予備への切り替えはない。プロセッサおよびメモリの故障では自プロセッサが単に停止し、上位に停止を報告する。この階層でのCNPの故障処理

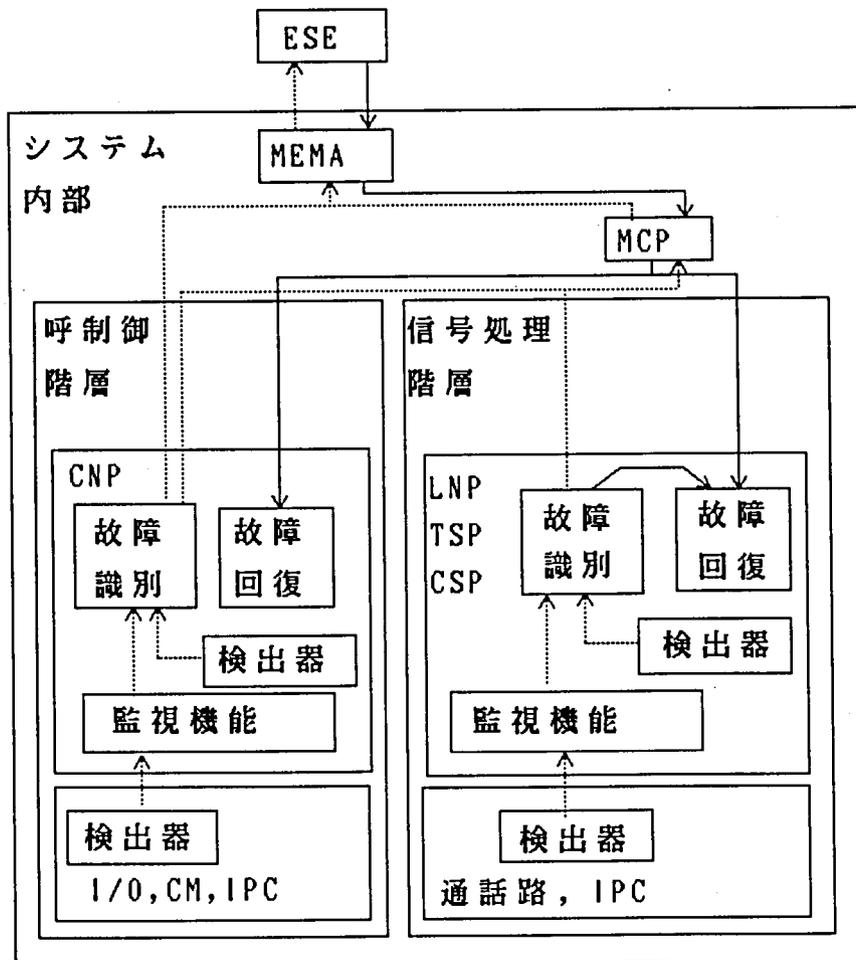


図6.3 故障処理の制御の流れ

はCNP自身の故障処理よりむしろ付属するIPCやCMの故障処理が有効になる。

IPC, CMはすべてのプロセッサの共通装置であるが、プロセッサ故障と同様に故障を検出したモジュール内で故障処理を行う。これらは受動的な装置であり、故障を検出したプロセッサの付属装置として扱える。ただし、個々のプロセッサが独立に共通装置の切り替えを行うと混乱を起こすので実質的な処置は後述するMCP（マスタプロセッサ）とよぶ1台の特権プロセッサを利用する。局所処理階層で他のプロセッサを利用することは概念上は矛盾に見えるが、MCPは各プロセッサから見ればむしろ付属する受動的な故障切替え手段である。

### 3.2.2 マスタプロセッサによる故障処理階層

#### (1) 故障処理機能の集約

個別モジュール内に閉じた故障回復が困難な場合は局所処理を超えて、システムを一元的に扱うMCPにより故障回復を行う。MCPは故障回復における複数プロセッサの競合を防ぐものでシステム内に同時に1個のみ存在する。またMCPは階層化構成により個々のモジュールの内部を意識せず、1個の回路と考えて制御する。MCPは複数装置間の組合わせである系構成を確定し、装置間接続ルートのフリップフロップのオン/オフや他のプロセッサの起動、停止などをプログラムで実施する。

MCP用ハードウェアは単純な機能であり、全ての局の基本構成として存在するCNPがハードウェア機能を具備し、そのうちの1台だけがソフトウェア機能としてMCP権限を持つ。MCPの権限を持つCNPが故障したときは他に故障の影響を与えないよう直ちに停止し、次にMCPになるべき予定のCNPを決めておきプログラム論理に基づいてMCPの権限の移譲を受ける。なお、MCPとなっているCNPも通常は他のCNPと同様、呼処理を行っている。

#### (2) 故障処理機能の内訳

下位階層で自己再開が不可能なモジュールは停止することとした。この階層では停止したプロセッサをMCPが検出して故障処理を行う。

プロセッサが停止した場合には極力システムを稼動状態に維持する観点から、この原因をまずプログラムバグと考えプログラムバグ対策をとる。高集積化によるハードウェアの高信頼化によりプロセッサが停止する場合にはハードウェアの故障よりもプログラムバグの可能性が高い。

プログラムバグに対する故障処理は、具体的には停止したモジュールの初期設定を行う（充分デバッグを行ったシステムにおいては極めて稀に発生する現象の組合わせによりバグが顕在化するものであり、通常は端末を含む初期設定によって通常はこのバグ顕在化の要因を除去できる）ことにより可能である。その後MCPはファイル装置からのプログラム引上げと停止したSGP、CNPの再起動を行う。MCPはこの初期設定の履歴を保存し、再度故障が生じたときハードウェア故障と認定し、疑わしい装置を固定的に停止させる。

### (3) 故障処理アルゴリズム

故障処理アルゴリズムとしてローテーション方式と呼ばれ、既存の交換方式に既に導入された考え方<sup>(80)</sup>を用いる。故障が検出される度にあらかじめ決められた手順で全ての系構成の組合わせを試み、故障装置を除いた系構成が存在するかぎり、いつかはこれに到達することを保証する。故障状態が論理的に波及するなどから故障原因を間違えて特定し、故障からの復帰が不能となる場合に備えたものである。

ローテーション法は故障除去のアルゴリズムとして有効であるが、マルチプロセッサは系構成の組合せが多いため単純にローテーション法を導入すると故障回復までの時間が長くなる。このためローテーション法を以下のように修正する。第一の考え方は最も疑わしい装置を最初に取り替えることである。例えば、通話を運ぶPCMキャリアのクロックがスイッチ系の故障で乱れたことを中継信号装置で検出したときはスイッチ系と中継信号装置間に被疑範囲がまたがるが、まず故障確率の高いスイッチ系の機能ブロックを取り替える。しかし、故障が除かれないときは次に信号装置の機能ブロックも予備に切り替える。

第二の考え方は他に与える影響が少ない装置を優先的に切り替えることである。例えばSGPとIPCの間で被疑箇所指定が困難な故障が生じた場合、故障範囲が限定されるSGPを先行して切り替える。

#### 3.2.3 ハードウェア支援による故障処理階層<sup>[79]</sup>

次の上位階層としてプログラムの正常実行を監視し、ソフトウェア制御を回復するハードウェア支援階層すなわちマルチプロセッサ用緊急制御回路(MEMA)を設ける。MEMAは図6.4のように全CNPの停止を検出して新たなMCPを選定し、起動する機構をである。ソフトウェア制御能力喪失に対して動作する。

下位階層では少なくとも1個のCNPが生き残っておれば、これがMCPとなり、プログラム制御の下で他のプロセッサの再起動等により系構成を確立し、故障が回復がなされるものであった。しかし、共通メモリを介した故障状態の波及等によりMCP、CNPの全てが故障を検出し、停止する可能性がある。MEMAはMCPを核としたプログラム制御による自律的回復を支援するものである。

MEMAはシステム再構成の核であり、高信頼でなければならない。この観点から極力回路規模を小さくし、故障率を下げる工夫が必要である。

これに対し、故障処理を階層化し、下位階層でプログラム論理を利用したことによりMEMAにはプログラムの動作を保証する機構を設け、プログラムを起動するだけで機能実現が図れる。

具体的には全プロセッサの停止を検出するアンド回路、MCPになるべきCNPを選択するカウンタ、MCPを起動する微分回路で構成することができる。

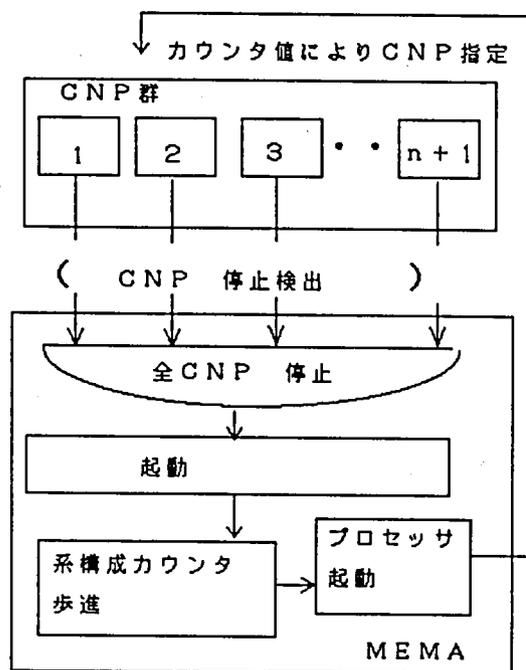


図6.4 MEMA機能の概要

MEMAは故障プロセッサを認識せずカウンタの値によってCNPを順に選定し、MCPとして起動すればよい。選択されたMCPが起動不能な場合は再びMEMAが動作し、歩進されたカウンタで次のCNPがMCPとして再起動される。初期設定により故障の残骸を消し、新たな系構成を選定しつづければ故障装置を除外した系構成に達する。

このようにして故障を除いた系に到達すると、数あるプログラムルートの中でも運用開始前にすでに検証した故障回復ルートをとどれる。従って確実な故障回復の保証ができる。

### 3.2.4 外部監視支援階層

ローテーション法によれば故障が検出される限りMEMA, MCPを利用して故障原因が除去できることを示した。しかし、故障検出機構が十分でなければローテーション機構を起動できないし、故障回復を保証しえない。経験的には設計者の思考による合理的手順を確立すれば十分であるが、完全である証明は出来ないし、バグの可能性もありうる。このため制御システムの外部に故障検出の最終手段を具備する。

故障検出の最終手段として外部監視回路(ESE: Emergency Supervision Circuit)を設ける。これは交換機の端末として動作し、自動的にかつ定期的に発呼し、交換機がこれに回答しなければ故障と認定するものである。交換機全体の正常性を交換機とは独立に総合的に監視するものであり、最上位の階層である。もし発呼に対して回答がない場合は前記のMEMA回路の起動を行う。

## 4. 再開処理

階層化思想に基づき、故障装置が除去できた後は確定された系において呼処理再開のためのメモリ論理的設定や端末のハードウェア初期設定を行なわなければならない。この処理を再開処理と呼ぶ。再開処理は故障処理の段階に応じて以下の遷移があり、これを図示したものを図6.5に示す。

### 4.1 障害点復帰再開

障害点復帰再開は故障を検出した時点に実行中の命令番地に復帰し、処理の連続性を保証するものである。熱予備運転を行っている二重化予備構成の装置を対象とする。熱予備装置への切り替えは極めて短時間であり、再開処理も少量のプログラム実行で完了するため、この再開処理で回復する故障についてはサービス中断時間は無視できる程度に小さい。

### 4.2 個別再開処理

熱予備運転を行っていない場合や単純な切り替えで回復しない場合にはモジュールに閉じて初期設定再開を行う。この再開処理はソフトウェアバグが故障原因である可能性を考慮したものである。

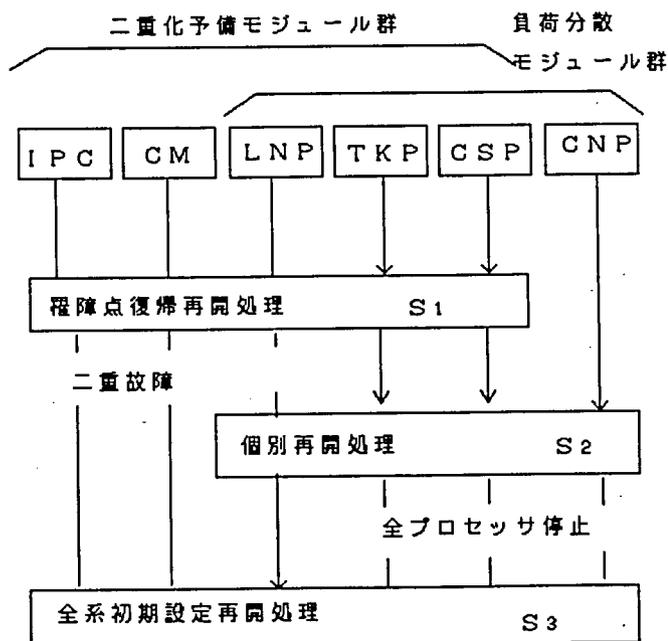


図6.5 再開処理の遷移図

他のモジュールは個別再開の間も無関係に正常な処理を続ける。ただし、初期設定中のモジュールへの他からのアクセスは禁止し、極力影響を小さくする。

複数モジュールにまたがって処理され、矛盾を生じる呼（たとえば、LNPが発呼側で初期設定されればその呼に対応するTSP側も中継線の解放がなされなければならない）には故障回復後、両モジュールが同期をとって呼の状態の合理性を検査し、個別に回復処理を行う。

この再開処理の中の大きな処理はファイル装置からのプログラムの引き上げ、呼情報用データの初期設定、さらにSGPでは通話路系装置初期設定がある。局条件等によりこれらの内、いずれかがサービス中断時間の支配項を占める。

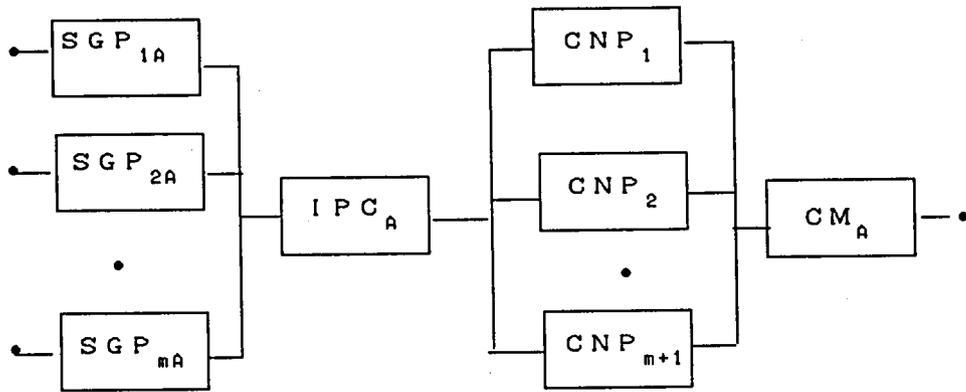
#### 4.3 全系初期設定再開

故障処理が全系にわたる場合のシステム全体に渡る初期設定再開である。この全初期設定の間はシステム全体の呼処理が中断される。

中断時間の支配要因は個別再開処理と同様である。ただし、中断時間の短縮のためMCPにより起動されたプロセッサは自モジュール内のメモリ初期設定、通話路初期設定などを並列に行うため、各要因およびモジュールの最悪のものに支配される。

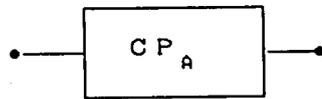
## 5. 評価

故障処理については故障からの回復の確実化と、これに伴う中断時間の最小化が目標である。このうち確実な故障回復の考え方と回復手順は既に示した。次の課題はこの確実な回復手法が中断時間の増大を招かないかにある。従ってここでは故障回復に伴う中断時間の大小を評価する。評価にあたっては個々の回線からみた統計的期待値を用い、また、図6.6のように構成を単純化し、各種変数をパラメータ化して議論する。プログラムの処理法など機能的な面は重視するが、実現世代の特定の部品技術に制約された絶対値を扱うのではなく、マルチプロセッサの性質を議論する。



(a) マルチプロセッサ構成

(注) 添字Aは二重化予備構成の現用側を意味する



(b) シングルプロセッサ構成

図6.6 評価対象のシステム構成

### 5.1 評価パラメータ

評価にあたって以下のパラメータを用いる。

$U_E$ : 構成装置の集合

マルチプロセッサでは

$$U_E = \{(SGP_{10}, SGP_{11}), \dots, (SGP_{m0}, SGP_{m1}), (IPC_0, IPC_1), (CNP_1, CNP_2, \dots, CNP_{m+1}), (CM_0, CM_1)\}$$

シングルプロセッサでは

$$U_E = \{(CP_0, CP_1)\}$$

となる。ここで、( )内は対をなす装置群を示す。

$E_k$ : サービス可能な系構成 ( $k=1, 2, \dots, n$ )

$e_i$ :  $E_k$ に組込まれている装置

$S_i$ : 故障後の再開フェーズ ( $i=1, 2, 3$ )

$T_{i..e_i}$ :  $S_i$ に至る装置  $e_i$ の平均故障生起間隔

$t_{i..e_i}$ : 各フェーズの再開処理時間

$R_{i..e_i}$ :  $e_i$ による $S_i$ での故障波及度(特定故障で故障中断に遭遇する回線

数と全回線数の比)

RE: ある端子から見た中断に遭遇する確率

RE<sub>multi</sub>, RE<sub>duo</sub>: マルチ/シングルプロセッサ各構成に対するRE.

RE<sub>i, multi</sub>, RE<sub>i, duo</sub>: 再開フェーズS<sub>i</sub>に対するRE<sub>multi</sub>, RE<sub>duo</sub>

なお、ここで再開フェーズとは図6.5における再開処理の段階を略称したもので、  
罹障点復帰処理をS<sub>1</sub>、個別再開処理をS<sub>2</sub>、全系初期設定再開処理をS<sub>3</sub>とする。

なおシングルプロセッサ制御方式ではプロセッサの部分構成がないため罹障点復帰再開処理の後は直接全系初期設定再開処理に跳ぶ。

各フェーズの再開処理時間は後述するように装置種別による差分が少なく、一定と近似すると、各構成のREは以下のように表わされる。

$$RE_{multi} = RE_{1, multi} + RE_{2, multi} + RE_{3, multi} \quad (6.1)$$

ただし

$$\begin{aligned} RE_{1, multi} &= \sum_{e_i \in E_k} (R_{1, \cdot i} \cdot t_{1, \cdot i} / T_{1, \cdot i}), \\ RE_{2, multi} &= \sum_{e_i \in E_k} (R_{2, \cdot i} \cdot t_{2, \cdot i} / T_{2, \cdot i}), \\ RE_{3, multi} &= \sum_{e_i \in E_k} (R_{3, \cdot i} \cdot t_{3, \cdot i} / T_{3, \cdot i}), \end{aligned}$$

$$RE_{duo} = RE_{1, duo} + RE_{3, duo} \quad (6.2)$$

ただし

$$RE_{1, duo} = R_{1, CP} \cdot t_{1, CP} / T_{1, CP},$$

$$RE_{3, duo} = R_{3, CP} \cdot t_{3, CP} / T_{3, CP}.$$

## 5.2 評価の前提

以下の前提を設け、更に個別のパラメータに対しては各小節に述べる仮定により議論する。

(1) マルチプロセッサでは装置数が多く、故障と故障回復の組合せ数は極めて多いが、ここでは多重故障などの確率の低い特殊な処理ルートはあらかじめ除いて議論する。

(2) 交換機の規模は収容回線数を単位として線形増加するが、ここではマクロな議

論としてSGP、CNPの台数で量子化して議論する。

(3) 中断時間の支配項は4章で述べた再開処理時間であり、図6.6の再開処理遷移に基づいて評価する。

### 5.2.1 故障波及度

罹障点復帰とMCP支援による個別初期設定の段階ではSGPモジュールの故障に対して1モジュールのみ影響を受ける。従って、 $R_{1.SGP}$ 、 $R_{2.SGP}$ はSGPモジュール数との比で $1/m$ である。なお、各SGPの扱う回線数はそれぞれ異なる可能性があるが、ここでは同量と仮定している。

CNP故障による影響は極めて少なく、 $R_{2.CNP} = 0$ とできる。CNPの故障発生の瞬間には処理中の呼のみが罹障するが、CNPに滞留する呼は一時には1呼のみであり、全体量から無視出来る程度に小さいことによる。また、通常の処理能力を満たす $m$ 台に加え、予備を加えた $m+1$ 予備のモジュールが動作しており、単一故障の仮定の下では影響は生じない。

全初期設定再開はシステム全体に影響を与え、いかなる装置 $e_i$ の故障に対しても $R_{3.e_i} = 1$ となる。また、シングルプロセッサでは常に全システムに影響を与え、 $R_{1.CP} = R_{3.CP} = 1$ である。

### 5.2.2 中断時間の仮定

#### (1) 中断時間の支配要因

前述のように故障に伴う中断時間は再開処理フェーズ $S_1$ 、 $S_2$ 、 $S_3$ が支配要因となる。このうち、 $S_1$ に相当する二重化予備モジュールでの熱予備装置への切替えは短い時間であり、無視できる。また、シングルプロセッサでは個別処理 $S_2$ は存在しない。

ここでマルチ/シングルプロセッサの中断時間を相対比較するため中断時間の比ITRを求める。個別再開に対しては $ITR = t_{2.SGP} / t_{3.CP}$ 、全系再開に対しては $ITR = t_{3.multi} / t_{3.CP}$ である。なお、 $t_{2.SGP}$ はマルチプロセッサの再開フェーズ $S_2$ のうち有意なSGPの故障に対する再開時間、 $t_{3.multi}$ はマルチプロセッサの $S_3$ を起動する構成装置の故障に対する再開時間、 $t_{3.CP}$ はシングルプロセッサの $S_3$ に相当する再開時間である。

S<sub>2</sub>、S<sub>3</sub>の中断時間の主要な要因には下記のものがある。

#### 通話路系初期設定時間

距離が長く電気容量の大きい加入者線にからむハードウェア初期設定は個々の初期設定を少電流で徐々に、しかも直列に行う必要がある。ただし、通話路装置はビルディングブロック化されており、ビルディングブロックモジュール間では並列に処理される。

この要因による初期設定時間は規模によらずモジュールごとの初期設定時間で規定され、 $t_{2.SGP} = t_{3.multi} = t_{3.CP} = k_p$  ( $k_p$ :一定)を最大値とできる。従って、 $ITR = t_{2.SGP} / t_{3.CP} = t_{3.multi} / t_{3.CP} = 1$ である。

#### メモリ初期設定時間

電子交換機はハードウェアの状態をメモリ上に写像して論理処理を行っており、通話路の状態と通話路の状態を示すメモリ内容を一致させる必要がある。この初期設定時間は局規模すなわちSGP、CNP台数に比例する。呼当たりの処理量はSGP、CNPでほぼ2分されていることからSGP、CNPの現用モジュール数をそれぞれ $m$ 台、初期設定対象のメモリ量をそれぞれ $k_m$ (一定)とすれば、初期設定対象の総メモリ量は $2k_m \cdot m$ となる。

個別初期設定は単位モジュールを初期設定すれば良く、 $t_{2.SGP} = k_m \cdot r$  ( $r$ は平均命令実行時間に比例する値)である。全初期設定でも各モジュールが並列に行うと近似すれば、 $t_{3.multi} = k_m \cdot r$ である。

シングルプロセッサ方式ではマルチプロセッサ最大規模の $2M$ 台分の能力のプロセッサを用いる。これにより総規模のメモリを初期設定するため  $t_{3.CP} = k_m \cdot m \cdot r / M$  である。従って、この場合は  $ITR = t_{2.SGP} / t_{3.CP} = t_{3.multi} / t_{3.CP} = M / m$  である。

#### ファイル転送時間

各プロセッサのメモリ上の破壊されたプログラムをファイル装置から再読み込しする必要がある。転送時間は転送パスのスループットとプロセッサ台数によって規定される。ここでプログラム量はSGP、CNPに2分割されるとし、各プロセッサ1台への転送時間を $k_f$ (一定)とする。また、シングルプロセッサでも同一転送手段を用い、SGP、CNPを合わせたファイルが転送されるとする。

それぞれの転送時間は  $t_{2.SGP} = k_f$ ,  $t_{3.multi} = 2m \cdot k_f$ ,  $t_{3.CP} = 2k_f$  であ

る。従って、この場合は個別再開では  $ITR = t_{2.SGP} / t_{3.CP} = 1/2$ 、全系初再開処理では  $ITR = t_{3.multi} / t_{3.CP} = m$  である。

(2) 実現結果による確認

中断時間については実験システムでの実現結果を得ており、これにより上記議論の妥当性を確認する。マルチ/シングルプロセッサの中断時間相対値 ITR の実現結果と実験システムに合わせて  $M=8$  とした中断時間モデル評価（全系初期設定分）を図6.7に示す。

個別初期設定による中断時間は各種のSGPのいずれでもメモリ上の呼設定データの処理が支配要因で、規模  $m$  の増加に対し、1以下で漸減し、マルチプロセッサの方が相対的に有利となる。マルチプロセッサでは各SGPが並列に処理するのにシングルプロセッサでは1台のプロセッサで処理するため1台で処理すべきデータ量が増大することによる。

全初期設定ではSGPへのファイル転送時間が支配項であり、規模によらず2程度でほぼ一定である。前記の仮定でファイル転送時間はプロセッサ数に比例

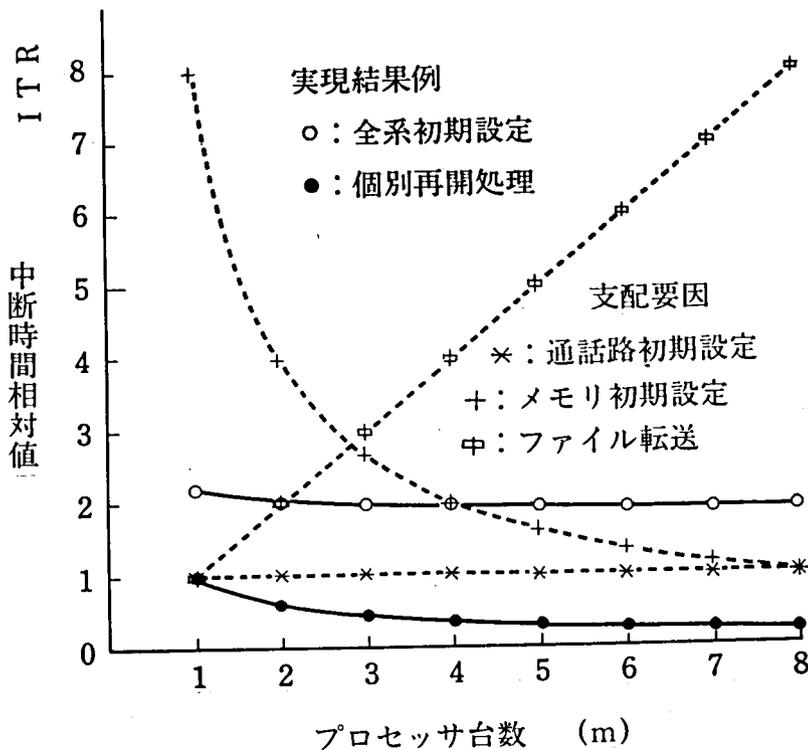


図6.7 マルチ/シングルプロセッサの中断時間比率

するとしたが、実験システムはプロセッサ数には比例せず、機能分担のプロセッサ種別数には比例している。これはSGPとしてLNP、TSPの2種の典型的な交換機で、転送手段として種別ごとに複数SGPへ放送形式を用いることとしたためである。

個別初期設定、全系初期設定ともモデルから大きく外れるものではない。

### 5.2.3 故障原因と生起頻度

実際のシステム設計では対案ごとに実績データを積上げて比較するが、ここではマルチプロセッサの性質、傾向を示すための議論であり、以下の仮定を設ける。

#### (1) ハードウェア故障

シングルプロセッサ制御方式との比較のため、第V章と同様にプロセッサの平均故障率とパフォーマンスの関係が必要であるが、ここでも前章と同一の仮定を行なう。すなわち式(5.6)からマルチプロセッサの最大規模相当の能力のシングルプロセッサの故障頻度は

$$\begin{aligned} 1/\text{MTBF}_{CP} &= a_0 (2M \cdot P_{CNP})^\alpha \\ &= (2M)^\alpha / \text{MTBF}_{CNP} = (2M)^\alpha / T_{f,CNP} \end{aligned} \quad (6.3)$$

なお  $T_{f,e_j}$  は装置  $e_j$  の平均故障間隔である。またSGPとCNPでは同種のプロセッサが用いられるため  $T_{f,CNP} = T_{f,SGP}$ 、共通装置も設計結果の金物規模から第V章と同様、 $T_{f,IPC} = T_{f,CM} = T_{f,CNP} * 2$ とする。

#### (2) ソフトウェアエラー

通常、ソフトウェアエラー（バグの顕在化）の出現頻度は走行する命令頻度に比例すると考えられる。また電子交換機の運用結果ではCNP規模のプロセッサのハードウェア故障発生率とソフトウェアのエラー出現率はほぼ同程度であり、これを前提とする。すなわち、SGP、CNPのソフト/ハードウェアの故障とも故障発生間隔を  $T_{f,CNP}$  とする。なお、シングルプロセッサではマルチプロセッサの最大能力すなわち、CNPの2M倍の最大能力を有するが、局規模に対応した実効的に命令走行数はCNP、SGPの合計2M台相当である。同様にCNP

も通常  $m+1$  が稼動しているが、有効な命令実行数は  $m$  台分である。

#### 5.2.4 故障回復フェーズの遷移

以上の条件および仮定から求めた故障処理の遷移を遷移確率と中断時間とともに図6.8に示す。なお  $S_1$  以降への遷移の原因には故障処理プログラムのプログラムバグとハードウェアの多重故障が考えられるが、多重故障の生起頻度は数オーダー低いため省略した。シングル/マルチプロセッサ両構成のデバッグは両者とも同程度の試験項目によって管理されており、従って同程度のデバッグ精度が保たれるとする。これにより両者の次フェーズへの遷移確率を一律に  $v$  と仮定する。またローテーション法は故障装置を除外しうるまで全系故障処理を繰り返すが、ここでは既述のように故障の可能性の高い装置にあらかじめ重味づけした優先付きローテーション法を用いていることから、単一故障時にはほぼ1度で故障回復するとした。

#### 5.3 故障処理中断遭遇比率の具体化

ここで両構成のRE式(6.1)、(6.2)を図6.8に従って変形し、比較のため、 $R = RE_{muti} / RE_{duo}$  とおくと以下になる。なお、故障波及が少なかったり、滞留時間が極めて短い再開フェーズ(図6.8の点線部)は省略している。

(a) 通話路系初期設定が支配項の場合

$$R = \{3(m+1)v+1\} / \{(2M)^{\circ}v+(2m)\} \quad (6.4)$$

(b) メモリ初期設定時間が支配項の場合

$$R = \{3(m+1)v+1\}(M/m) / \{(2M)^{\circ}v+(2m)\} \quad (6.5)$$

(c) ファイル転送時間が支配項の場合

$$R = \{(6m^2+4m+1)v+1\} / 2 / \{(2M)^{\circ}v+(2m)\} \quad (6.6)$$

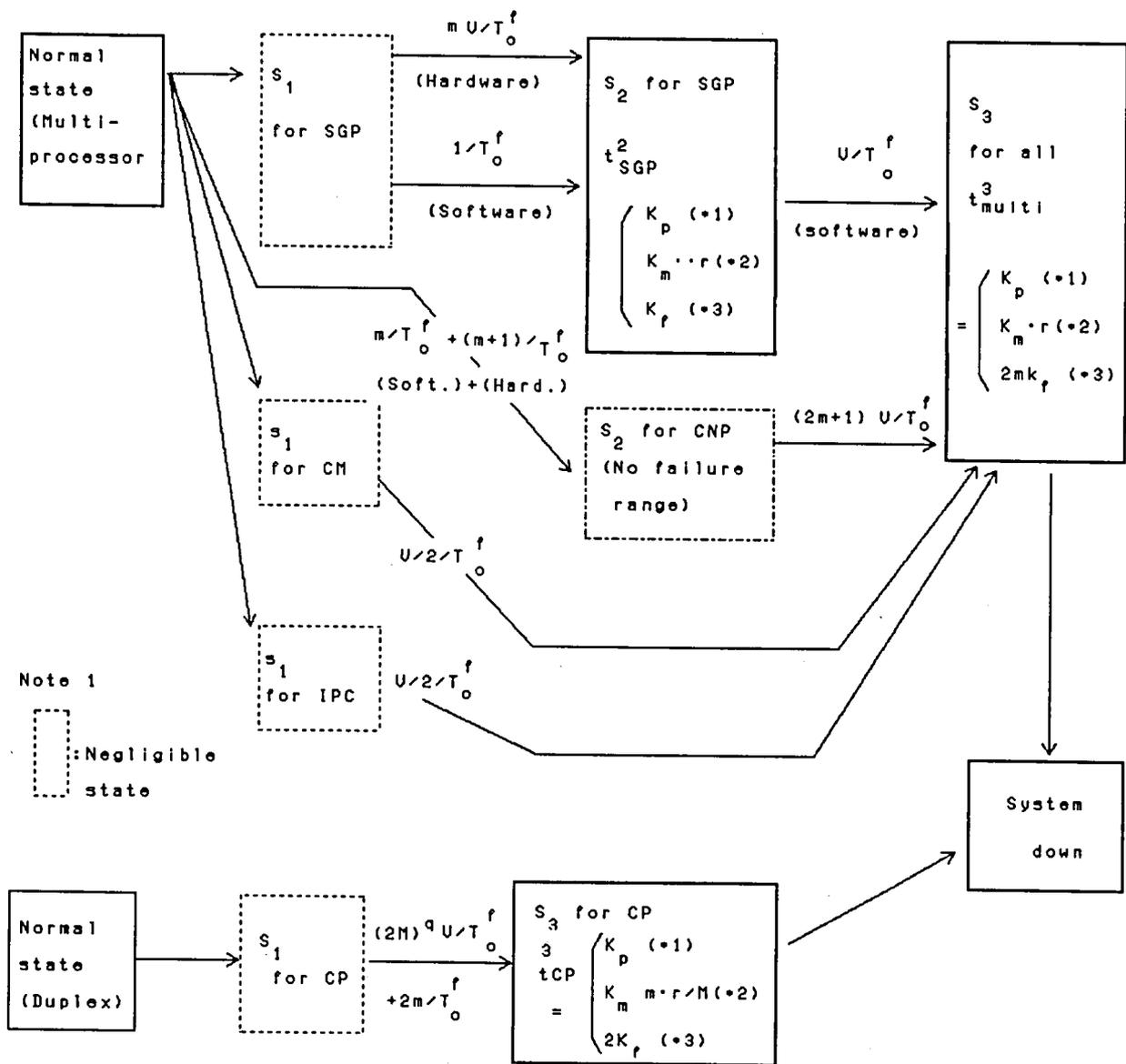


図6.8 確率を考慮した再開処理の遷移

#### 5.4 シングル/マルチプロセッサ方式の比較

次にマルチプロセッサの特徴である局所化の割合 ( $v$  に対応) をどの程度にすればマルチプロセッサが有利になるかを議論する。このため、両者のREが等しい  $R=1$  となる  $m$  と  $v$  の関係をもとめると以下になる。

(a) 通話路初期設定が支配項の場合

$$v = (2m-1) / \{3(m+1) - (2M)^q\} \quad (6.4')$$

(b) メモリ初期設定時間が支配項の場合

$$v = (2m^2 - M) / \{3M(m+1) - m(2M)^q\} \quad (6.5')$$

(c) ファイル転送時間が支配項の場合

$$v = (4m-1) / \{6m^2 + 4m + 1 - 2(2M)^q\} \quad (6.6')$$

#### 5.5 計算結果

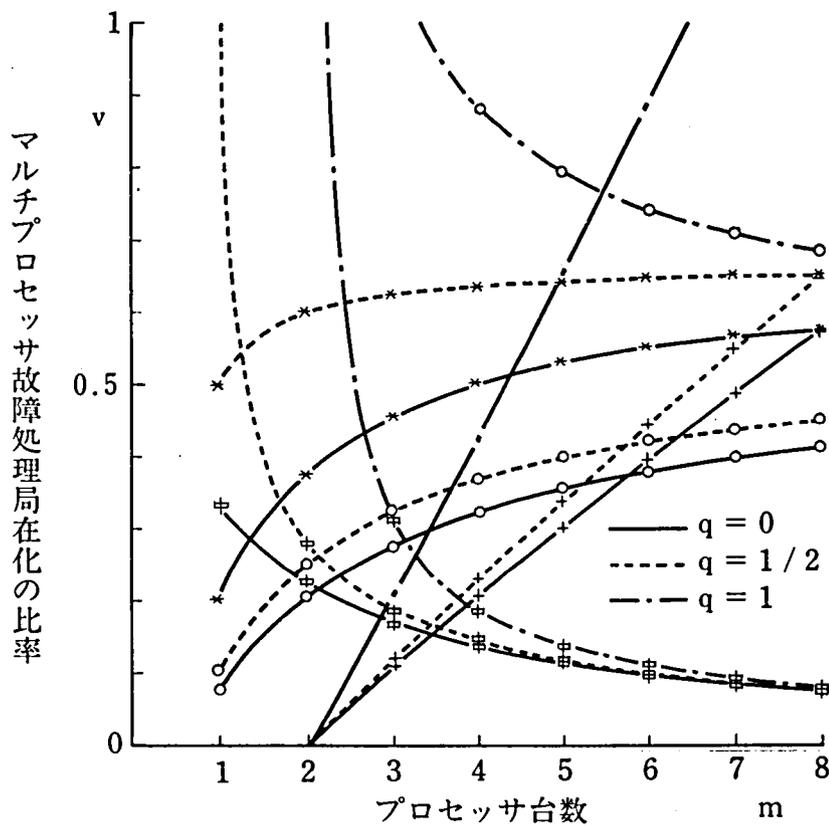
典型的な中断時間 3 支配要因、実験システムの実現結果に基づいて数値計算したものを図6.9に示す。ここでシングルプロセッサの能力を  $CNP$ 、 $SGP$  の最大モジュール数の合計  $2M=16$  とし、故障率に関する  $q$  を  $q=0$  (性能によらずプロセッサの故障率は一定)、 $q=1/2$  (グロシュの法則程度)、 $q=1$  (故障率は性能に比例) の3条件を仮定した。プロットした点の下部がマルチプロセッサ有利領域で、 $v$  がこの値より小さければマルチプロセッサの方が有利となる。

#### 5.6 考察

特徴を明確化するためのモデルを扱っており、絶対値を議論することは意味がないが、図6.9の傾向からシングルプロセッサに対するマルチプロセッサの故障処理中断時間の影響度について以下のことが言える。

##### 中断時間を支配する各要因に関する考察

- (1) メモリ初期設定が支配項の小規模領域を除き、マルチプロセッサが有利となる領域が存在する。
- (2) 通話路初期設定時間が支配項のとき比較的大きな  $v$  に対し、マルチプロセッサが有利となる。  $q=1$  では計算結果が  $v > 1$  となるように、マルチプロセッサが全



(注) ○ : 実システムの評価値  
 × : 通話路初期設定  
 + : メモリ初期設定  
 # : ファイル転送

図6.9 シングル/マルチプロセッサ故障中断時間の等価境界

面的に有利となる。 中断時間が両構成で同一であるのに対し、マルチプロセッサでは故障波及範囲が限定されること、 $q=0$ 以外ではシングルプロセッサの故障率が大きくなることによる。

- (3) メモリ初期設定時間が支配項のとき、小規模では $v$ が極めて小さく、全初期設定への移行率が小さくてもマルチプロセッサの方が不利となる可能性がある。小規模領域では初期設定すべきデータが少ないのに高速のプロセッサで処理するため中断時間が短いことによる。システム設計で留意すべき点である。大規模領域では単位プロセッサの性能は低くても並列に処理する効果が現れて

来てマルチプロセッサ有利領域が増大する。

- (4) ファイル転送時間が支配項のときは規模の増大とともにマルチプロセッサ有利領域が減少する。 ファイル転送をプロセッサごとに直列に行なうと仮定したこと、交換機規模が増大したことによりファイル転送すべきプロセッサ数が増大することによる。

#### 実現例に関する考察

- (1) 実現したシステムの例ではモデル評価の通話路初期設定支配項と同様の傾向となる。 初期設定時間がほぼ一定であることから当然の結果である。
- (2)  $q=0$ 、 $q=1/2$ では規模増大とともにマルチプロセッサ有利領域が増大する。 これは局所故障処理、個別再開により影響範囲を限定しうることによる。 すなわち、規模増大とともにマルチプロセッサの故障波及度 $R$ が減少するものである。
- (3)  $q=1$ では逆に規模増大とともにマルチプロセッサ有利領域が減少する。 シングルプロセッサ方式では小規模でも高性能のプロセッサを用いることにより相対的に故障率が高いが、規模増大とともにマルチプロセッサの故障頻度も高くなっていくことによる。

#### マルチプロセッサの有利性に関する結論

いずれの場合も故障処理の次フェーズへの遷移を数回に1回程度とすればマルチプロセッサが有利になる。 DTSデジタル交換機では、故障波及を防ぎ、故障処理フェーズの進展を抑える狙いから、他プロセッサからの通信に対し呼処理のレイヤで呼の状態の合理性を検査するなど考慮している。 従って、この遷移率は十分実現可能な値と考えらる。 また経験、運用実績を積むことにより更に改良されていくものである。

以上のように故障処理の面でもマルチプロセッサ制御方式は従来から用いられて来たシングルプロセッサ方式よりも有利になりうる事が分かる。

## 6. 結言

マルチプロセッサは必要な処理量に応じてプロセッサを追加出来るため、幅広い規模分布や長期に渡る維持管理が必要な交換機に対し、統一機種化が図れ、望ましい方式である。しかし、複数のプロセッサが並列に走行するため、故障時にはこれらの干渉を排し、確実な故障回復が求められる。この観点から故障処理の階層化を図り、極力故障回復の迅速化をはかるとともにサービスを実現しうる系が存在する限り、その系に確実に到達し、呼処理への復帰を図りうる手順を実現した。また、この故障処理によるサービス中断の影響を評価する定量化の1手法を示し、部分的な故障処理で故障回復を図り、全初期設定を極力少なくすれば故障処理の観点からもシングルプロセッサ方式に比べ、サービス性が向上することを示した。

その内容は以下のように要約される。

### (1)故障処理の階層化

マルチプロセッサの複雑な故障処理を以下の4階層の故障処理で故障回復を確実化する手法を示した。

局所故障処理： 故障を検出したプロセッサが個別に故障回復を行なう。

マスタプロセッサによる故障処理： プロセッサ全体の中で1個の特権プロセッサを設けてソフトウェア処理で故障回復を行なう。

ハードウェア支援故障処理： 全プロセッサが停止したことを検出して新たな特権プロセッサを指定するハードウェア機構を起動する。

外部監視支援故障処理： 定期的に交換機の端末と同様の発呼動作を行い、交換機全体の正常性を監視し、異常時マスタプロセッサの再選定と起動を行なう。

### (2)再開処理の具体的手順

マルチプロセッサの故障からの再開処理として以下の3種類があり、故障からのサービス再開にいたるまでのサービス中断では、再開処理時間が支配項であることを示した。

障害点復帰： 故障を検出した命令に復帰する。

個別再開処理： 故障を検出したプロセッサのみの再開処理を行なう。

全系初期設定再開処理： システム全体の再開処理を行なう。

### (3) 評価

故障要因としてソフトウェア、ハードウェアの故障を規定し、バグの顕在化によるソフトウェアの故障は処理規模に比例し、ハードウェアの故障はプロセッサの性能との相関性を規定したモデル化を行なった。また、上記の故障処理に基づく故障遷移と確率を明かにし、故障遷移モデルを示した。また、故障中断時間として通話路初期設定、メモリ初期設定、ファイル転送時間が支配項になる可能性が有ることを示し、その中断時間を求めた。これらを総合し、シングルプロセッサとマルチプロセッサの故障による中断時間を比較した。

その結果、

- ① マルチプロセッサでは故障波及を防ぐことにより既存のシングルプロセッサよりも中断時間期待値を小さくできる。マルチプロセッサはサービス中断時間を小さくできる。
- ② マルチプロセッサでは単一プロセッサに閉じた故障は多いが、システム全体に影響を与える故障は少なく、規模別信頼度規定の考え方に適合する。

## 第VII章 分散伝達方式の展開<sup>[81]~[86]</sup>

### 1. 序言

前章まででデジタル交換機の実用化に併せて進めた交換機用マルチプロセッサの確立について述べた。本章では実時間処理，超多重処理，24時間無休止高信頼を特徴とするマルチプロセッサ制御方式の確立以後の交換機制御方式の動向について展望する。

デジタル交換機においては，現在CCITT（国際電信電話諮問委員会）のISDN国際標準（Iシリーズ）が導入されており，加入者までのデジタルリンク化が実現された。デジタル交換機の次の課題は高速デジタル通信サービスの確立と考えられる。高速通信サービスは，現在はビジネス利用に特化して，専用線サービスとして提供されている。しかし通信サービスは不特定多数間の任意の通信を可能として社会基盤としての役割を果たすものであり，高速通信サービスも当然大衆普及へ拡大して行かなければならない。

シーズについては近い将来実現可能な数100kゲートの集積能力の利用が最も大きなインパクトとなる。10kゲート台のVLSI素子は装置として構成されたものをオンチップ化可能とし，マイクロプロセッサの一大普及を推進して，情報処理の大衆化を実現した。次に来るULSIは，100kゲート台の集積能力を実現し，デジタル交換機のオンチップ化を可能とする。大衆普及力を有する高集積技術のインパクトを通信網では高速通信サービスの大衆普及化に利用すべきであろう。歴史的に見ても，高集積技術は全く新しいものを生み出す力はなく，実現可能ではあるが高価なものを経済化させ，大衆化させることにインパクトを与えてきている。高速通信サービスはまさしく実現可能であるが，高価なもの代表である。

本章では以上の考え方により，まず高集積技術によって交換機の小型経済化が可能かどうかを検証する。次に高集積技術の最も大きなインパクトである小型経済化をシステムにどのように活かすかを議論する。また，システム構成イメージを示し，高速サービス実現に向けた網へのインパクトを述べる。さらにオンチップシステムの実現に向けたシステム検討の一手法を提案する。最後に今後研究すべき残された課題について述べる。

## 2. 高集積化による交換機の小型・経済化の可能性

VLSIプロセッサでは回路規模の90%がLSI化されながら、低集積のSSIやMSIが多く、IC数比率では5%がLSI化されているに過ぎなかったことを第IV章 5.6節で示した。これは今後高集積技術が更に進展してもシステム全体の小型・経済化が十分には達成できないことを暗示している。交換用プロセッサにとって高集積技術が進展してもSSIやMSIを多用せざるを得ないのは装置間のインターフェースが多いことが原因である。さらに装置間インターフェースの多い原因は24時間実時間サービスを提供するための複雑な予備構成に大きな原因があることが解っており、この観点からまず予備構成の問題を議論し、小型化への道程を示す。

### 2.1 予備構成単純化による小型化

交換機は24時間サービスシステムである。このため、部品の故障に備えて予備を具備する。予備は交換機の構成に必須であるものの、常時のサービス提供には必要がなく、性能や経済性の達成の観点からは足かせとなる。したがって、有効な予備構成方式の確立は交換機設計上、常に大きな課題である。ここでは部品技術の進歩に影響を受けた交換機の予備構成の進化の経過を分析し、今後の予備構成方式の動向を推測する。

#### (1) 予備を具備する範囲

予備切り替えは人間による故障部品の取り替えよりもはるかに高速に故障の影響を除去できるため、適切に予備対象範囲を設定することは交換機の高信頼化に貢献する。ここでは空間分割電子交換機とデジタル交換機との予備対象範囲の差について構成部品高集積化の観点から振り返る。

##### (i) 空間分割電子交換機の予備対象範囲

通話路に電磁部品を用いた空間分割電子交換機は図7.1に示すように構成要素を通話路系と制御系に分解でき、さらに通話路系は通話路部と駆動部に分解できる。スイッチやトランクなどの電磁部品を駆動する通話路系駆動部と制御系は複数の

呼に共通に利用されるが、スイッチやトランクなどの通話路自身は、呼ごとに個別に利用される。前者は電子部品で構成され、回路の論理規模が比較的大きいのに対し、後者は電磁部品や機械部品で構成され、論理規模が小さいという特徴がある。

この構成において、電子部品で構成される共通部、すなわち通話路系駆動部と制御系は予備を具備するが、個別部である通話路は予備を持たない。予備を具備すると予備切り替え回路が本体回路よりも大規模化し、予備を設けることがむしろ信頼性を低下させる危険性があることから当然の帰結である。

#### (ii) デジタル交換機の予備対象範囲の拡大

デジタル交換機では規模別信頼度規定に基づき影響範囲の大きな共通部には極力予備を具備する方向がとられている。デジタルスイッチはLSIで構成され、この故障が与える影響範囲は大きいことによる。しかも予備を具備するに際しても、高集積化の進展は以下のように予備の経済的な実現を支えている。

① 多重化単位に予備切り替え機構を持てばよく、予備切り替え機構が経済的に実現できる。

② 高集積部品を利用することにより予備自身も経済的に実現できる。

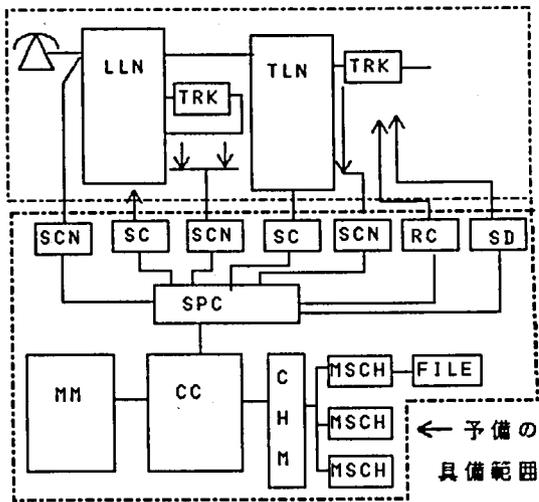
従来は予備を具備しなかった通話路部にも図7.1(b)のように予備を具備する範囲を拡大し、高信頼化を図っている。

#### (iii) 予備具備範囲に関する結論

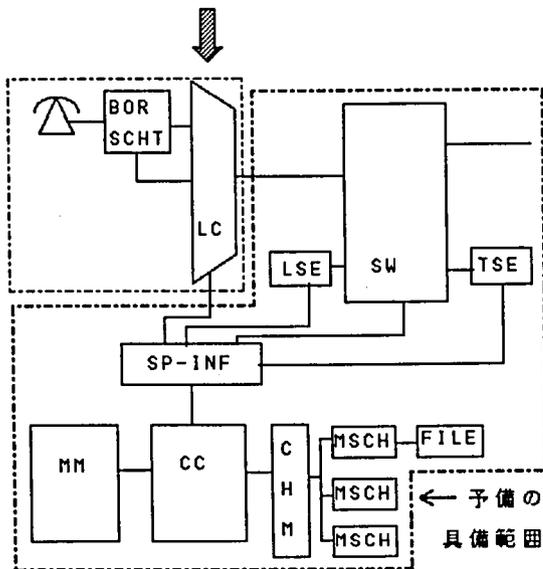
部品高集積化が予備を具備する範囲の拡大を推進していることがうかがえる。デジタル交換機ではBORSCHT等の個別回線対応部を除き、すべての構成要素が予備を具備するに至った。

#### (2) 予備構成方式の動向

ここでは高集積化の観点から予備構成方式の進化を振り返る。



(a) 空間分割電子交換機



(b) デジタル交換機

- LLN: ラインリンクネットワーク
- TLN: トランクリンクネットワーク
- TRK: トランク
- SCN: 走査装置
- SC: 通話路駆動装置
- RC: 継電気駆動装置
- SD: 信号分配装置
- SPC: 通話路制御共通装置
- CC: 中央制御装置
- MM: 主記憶装置
- CHM: チャンネル多重化装置
- MSCH: マルチプレクササブチャンネル装置
- FILE: ファイル装置
- LC: 集線装置
- LSE: 加入者線信号装置
- SW: スイッチ
- TSE: 中継線信号装置
- SP-INF: 通話路インタフェース

図7.1 電子交換機における予備の具備範囲の拡大

## (i)問題提起

### 装置間接続の複雑性

予備構成単位間の接続は通常、柔軟性に富むケーブルで接続される。サービスを実行する常用系に影響を与えないで、予備に回された故障装置の故障部品の取り替えを可能とするためである。装置内の部分的部品の取り替えには電源を切断してから行なうため電源供給の独立性を保証するだけで良い。しかし、装置全体の取り替えを含む大規模な故障修理の可能性もあり、取り外しの出来る予備装置間のケーブル接続は必須となる。

さらに交換機はもともと線を繋ぐノードシステムであり、収容する線数の多いシステムであるのに加えて、交換機内部にもケーブル接続インタフェースが以下のように多量に存在する。

- ①蓄積プログラム制御方式電子交換機では通話路ハードウェアの状態を制御系主メモリに写像してメモリ上で処理するため、絶えず通話路ハードウェアの状態を走査しなければならない。走査や通話路の制御のための情報転送能力の高いインタフェースを必要とし、並列度が高く幅の広いインタフェースを必要とする。
- ②このような実時間制御インタフェースが制御系、通話路系の各予備間の交差にも適用される。
- ③主記憶装置に写像された通話路状態の内容は、制御系の故障に備えて予備の記憶装置にもコピーを持つことを原則とする。中央制御装置から現用と予備の両方の主メモリに二重書き込みを行なうため、各予備間でも高速のメモリアクセスインタフェースを具備する必要がある。

### 高集積効果の鈍化

部品高集積化とともに装置の本体論理部は小型・高信頼化されていくが、ケーブル接続インタフェースが存在するため、高集積部品を適用してもシステム全体の小型、経済化の効果は鈍化することになる。ケーブル接続には信号駆動用に電力を必要とするため低集積部品を必要とすることが原因である。交換用VLSI処理装置では既存の処理装置に対し、十分な小型・経済化を達成しているものの、高集積部品利用率の観点では、回路規模で90%に達し、IC数では5%に過ぎないという例を既に述べた。

予備構成図	近似評価モデル	備考
<p>n+1予備構成</p>		<p>I-回路数： 2n+4</p> <p>実現例： D10-CP D10-HCP D20-CP</p>
<p>二重化予備構成</p>		<p>I-回路数：4</p> <p>実現例： 改D20-CP D60-CP D70-CP</p>

I:装置間接続インタフェース回路

図7.2 交換機制御系の予備構成例

一層の高集積部品技術の進展動向に対し、IC数でも高集積部品の適用比率、適用効果を高める方式の確立が必要である。このためには接続インタフェースに影響の大きい予備構成方式を改革することが課題である。

(ii) 制御系における予備構成の進化の例

ここでは空間分割電子交換機の時代から進化を続けた制御系の予備構成方式の動向を復習する。制御系は電子部品で構成され、空間分割電子交換機の時代から度々の改良が加えられてきた。予備構成の進化してきた典型例といえる。

制御系の予備接続法には多くの変種があるが、基本と考えられる予備構成を図7.2に示した。ここで両構成の信頼性を大まかに比較する。

部品低集積の極限

本体論理部の規模が極めて大きくインタフェース部は無視できるとし、両構成で修理時間を一定とすれば、信頼性は構成要素の故障率によって決まる。ここでn+1予備、二重化予備の不稼働率をそれぞれ  $F_{n+1}$ 、 $F_{2n}$  とし、CC、単位MMの故障率をそれぞれ  $f_{cc}$ 、 $f_m$  とすると

$$\begin{aligned}
F_{n+1} - F_{2n} &\propto (f_{oc}^2 + {}_{n+1}C_2 \cdot (f_m)^2) - (f_{oc} + n \cdot f_m)^2 \\
&= -(n(n-1) \cdot f_m^2 / 2 + 2n \cdot f_{oc} \cdot f_m) < 0 \qquad (7.1)
\end{aligned}$$

n + 1 予備構成が高信頼であり，低集積では構成機能を極力細分割して装置を構成し，小規模予備とすることが望ましい。

### 部品高集積の極限

図7.2の近似評価モデルにおいて，高集積化により本体論理部の回路規模は無視できるほど小さくなる。この結果，二重化構成では故障率が無視できる程度に小さくなるが，n + 1 予備においては低集積部品を必要とするCCのインタフェースの故障率の2乗オーダーが残る。二重化予備構成では高集積部品が支配項であり，故障率は無視できる。二重化予備の信頼性が優れる。

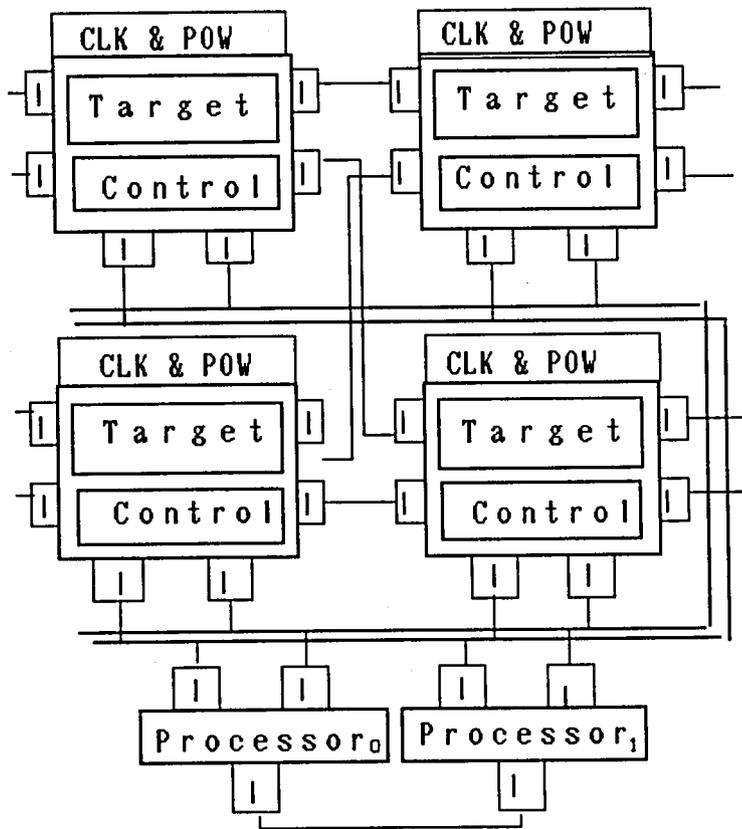
### (iii) 予備構成に関する結論

部品高集積化とともにインタフェースを極力削減する観点から単純な予備構成を指向し，予備構成単位が大規模化する方向へ進んでいる。

## 2.2 超高集積部品の進展による交換機の小型化

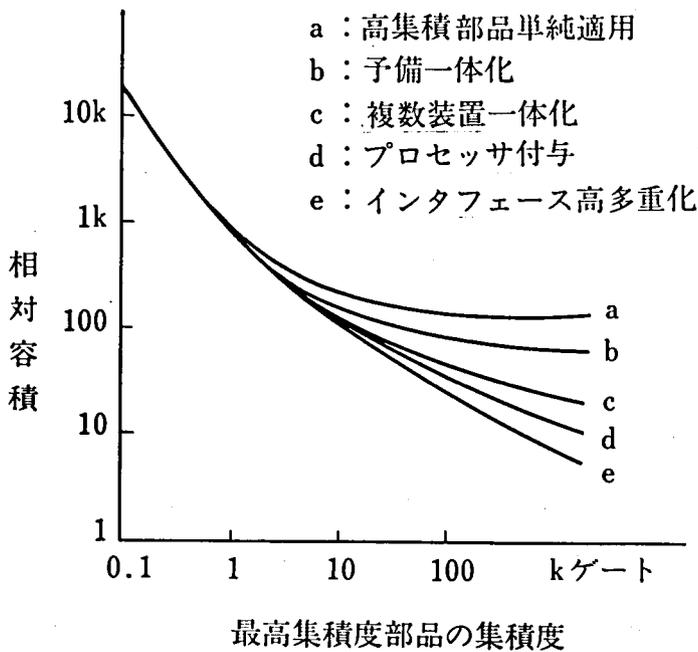
### (1) 小型化を達成する方式対策

今後一層の高集積化が進展するとして，デジタル交換機の小型化の可能性を検討した。一般的な交換機のモデルはプロセッサと複数の被制御回路（通話路等）で構成されている図7.3(i)と考えられる。このモデルに対し，各種方式対策の結果，交換機の小型が図られる様子を図7.3(ii)に示す。ここで横軸はシステムに採用される最大規模の高集積素子の集積度をパラメータとしている。既存のシステム構成をそのままに高集積化してもケーブル接続の各種のインタフェース回路が隘路となって高集積部品の特徴である小型・経済化の達成には限界があることがわかる（図7.3 (ii)の aの曲線）。このため，前節で得た高集積効果の活用には予備の単純化と大規模化が有効であるの結論を考慮して以下のように方式対策を採る。またその効果はそれぞれ図7.3(ii)の対応する曲線で示さ



CLK & POW : クロック分配,  
電源回路  
Target : 通話路等の被  
制御回路  
Control : 制御指令解読  
及び順序化制御信  
号発生回路  
Processor : プロセッサ  
I : インタフェース回路

(i) 交換機ハードウェアのモデル



(ii) 小型化の対策と効果

図7.3 高集積化とシステムの小型化

れている。

(小型化の対策と効果)

- b:各装置の予備単位ごとにあった交差を廃止する。これにより S S I 主体の交差インタフェース回路が削減できる。
- c:複数装置(機能)を一体化する。ただし、このためには①の予備間交差の廃止が必須である。これにより S S I が多い装置間インタフェースの削減が図れる。
- d:被制御装置にプロセッサを配置する。これにより、受動機能のみであった被制御装置(機能)が処理機能を含む機能完結したモジュールとして構成でき、対外部との通信はプロセッサによる処理結果であり、対外部インタフェースは疎な結合となり、インタフェースの削減が可能となる。
- e:交換機に收容するデジタル回線等の多重度を向上する。これにより同一情報を送る物理回線数は減り、回線收容のための低集積の S S I 等が削減できる。

以上の結果、交換機の中枢部では今後まだ小型化できることがわかる。ただし、加入者交換機では必然的に加入者対応部の単独收容部(デジタル交換機の黒電話收容回路 BORSCHT 等)があり、この部分は回線対応部として現状では議論の対象外である。

(2) 小規模交換機のオンチップ化

高集積部品を適用しても、その効果が飽和しつつあった交換機構成において、上記の方式対策により小規模な領域では交換機本体部のオンチップ化が実現される可能性がある。コンピュータにおける 1 チップマイクロコンピュータのように 1 チップ交換機が実現可能と考えられる。

交換機本体をオンチップ化した高集積交換機においては、各予備単位ごとに交換機能を完備することになる。各予備は独立に交換動作を実行できる。このため現用と予備とも並行に交換動作を実行し、交換機全体の熟予備化が可能である。この結果、現用と予備のメモリの間で情報を一致させるための命令毎の逐次的メモリ二重書き込みは不要となり、現用と予備が疎な結合となり、小型・経済

化が可能となる。なお、上記は予備を具備することを前提としたが、規模別信頼度規定の考え方によれば小規模領域では無予備化の可能性も考慮すべきである。

### 3. 超高集積部品の通信ノード構成へのインパクト

前節では方式的工夫をすれば、交換機が今後とも高集積化によって小型・経済化を達成し得ることを示した。ここでは小型・経済化の効果をシステムとしてどのように活かして行くべきかを議論する。

#### 3.1 小規模交換機の有効化

収容端子数  $n$  の蓄積プログラム制御の交換機では端子当りりのコストは一般に BORSCHT等の回線対応部のコスト  $a$  と固定分コスト  $b$  を用いて、 $a+b/n$  ( $a, b$ : 定数) と表わせる。システムの固定コスト  $b$  が大きいとき図7.4に示すように小規模では経済的に不利であった。今後、高集積部品技術の進展によるシステムオンチップ化などからこのシステム固定分のコストが大幅に経済化され、 $b$  が小さくなる。  $b$  が小さくなると端子当りコストの変化は大規模と小規模で大きく変わらな

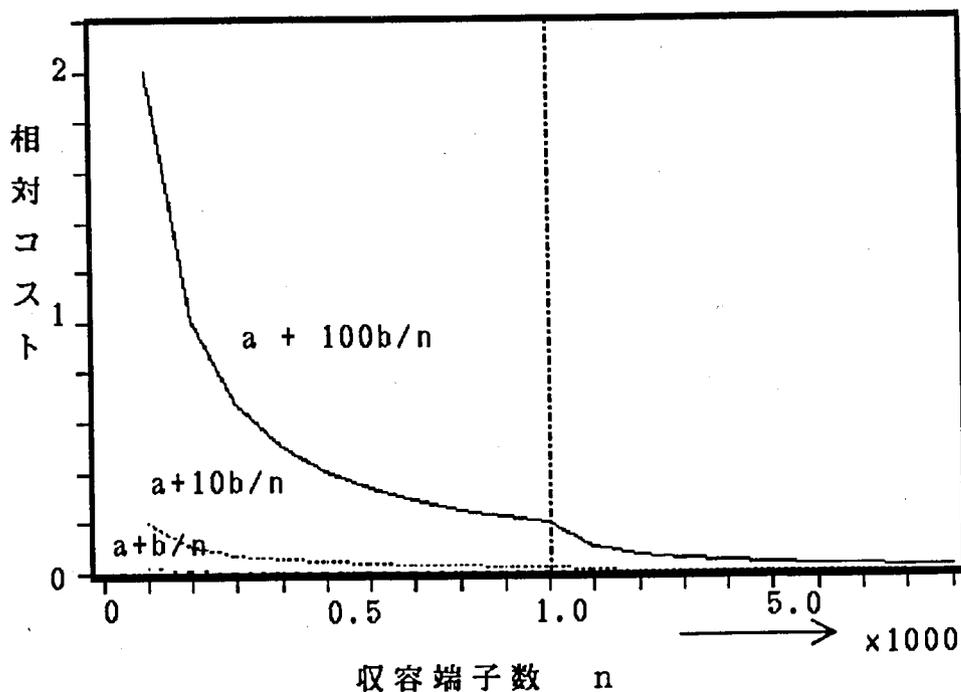


図7.4 交換機の規模と端子当りコスト

くなり、蓄積プログラム制御交換機の適用可能領域が小規模領域に大きく移行することが図7.4から読み取れる。コンピュータの世界ではマイクロプロセッサの出現が既存の経済原則、「大規模ほどコストパフォーマンスに優れる」とするグロシュの法則を無能化した。大規模ほど有利とされた交換機においても、むしろ小規模ほど有利になる可能性がある。

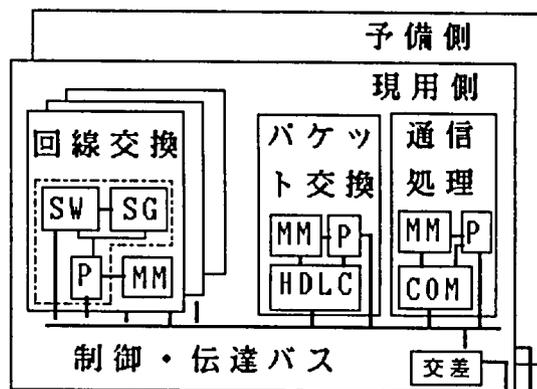


図7.5 高集積交換機の構成例

SW:スイッチ, SG:信号処理, P:プロセッサ  
MM:メモリ, HDLC:データ伝送制御手順,  
COM:伝達情報処理

### 3.2 大規模交換機の高性能化

最大規模の交換機をオンチップ化できれば、集積回路の量産性と廉価性により全ての適用領域を1種の交換機チップでまかなえる。しかし、現実的には交換機には常に機能や規模の需要増大を伴い、また現在予想される高集積能力からは全規模をそのままオンチップ化するのは不可能である。このため、適用規模に合わせたビルディングブロック化が必要である。この場合、オンチップ化した上記の小規模交換機のように、ソフト/ハードを一体として各モジュール内で機能を完結化させることが有効となる。小規模交換機の実装はプリント配線板でモジュール化して実現し、単純なプラグインで機能・規模の拡大が可能なビルディングブロック化の可能性がある。この場合も予備構成方式は図7.5に示すように複数モジュールを一体予備とした面予備とし、予備間の交差を少なくすることが有効である。

各面予備内ではビルディングブロックモジュール間を裏面配線板上で結合する密結合により高速な情報伝達が可能となり、大容量な交換機の実現が可能となる。また、伝達、制御インタフェースを含めてバス形式で結合することにより任意のモジュール間の通信が可能となり、融通性に富む交換機が実現できる。上記バ

スは交換機能だけでなく、回線終端機能を有する通信処理やパケット機能を有するモジュールも付加可能であり、ノード内のISDNバスとも言える。マイクロプロセッサにおけるVMEバスのように多様なノード機能をプラグイン可能とするためのバスとして標準化が重要な課題となろう。

### 3.3 高信頼効果

#### (1) システムの高信頼化

高集積化により部品点数が減少し、故障率が低下してシステムが高信頼化されることは自明である。この他に高集積化には以下のような効果がある。ただし、これらは高信頼化以外の効果を含むが、ここでは高信頼化として整理しておく。

##### ① システムの単純化とその効果：

オンチップ化が達成されれば予備構成がシステム全体の機能を一体とした予備単位となり、構成が単純化する。また少数のプラグインパッケージにより大規模システムが実現でき、操作性が向上する。故障が発生すれば単純に予備に切り替えれば良く、故障時の系構成に絡むソフトウェアが簡単になる。また故障の被疑対象が少なくなることから故障回路の取り替えも簡単になり、短時間で故障回復が図れる。

##### ② オンチップ化による固定化効果：

従来サービスを極力安く提供するためには最小規模のハードウェアでシステムを実現する考えが必須であった。このため規模比例の金物は極力必要時に増設することとしており、交換機においては増設対策がソフトウェアを複雑にしている一つの要因である。また故障を想定して各種の例外処理をソフトウェアは具備している。この例外処理もソフトウェアの複雑性を招いている要因である。

しかし以上のようなソフトウェアの複雑性はシステムオンチップ化が可能となると削除できる可能性がある。システムオンチップが実現されれば従来は増設対象とされた信号処理機能などの交換機の内部リソースは予め最大規模、すなわち最終設備容量を具備しておくことになる。これによりソフトウェア

は最大規模を想定して固定化して設計すればよく、内部リソースの増設に絡む処理を考慮する必要がなくなる。またオンチップシステムでは故障が生じれば、運用している現用系全体の故障であり、自から動作停止するなどきわめて簡単なソフトウェア処理になる。

近年の部品技術の進展からハードウェアが高信頼になり、人為操作やソフトウェアに絡む問題がシステム不稼働の大きな原因になりつつある。このような人為操作やソフトウェアにからむ問題の解決が重要とされる動向から、ソフトウェアの簡明化はシステムの高信頼化に有効である。さらにシステムが廉価になれば従来二重化程度の予備構成を採用していたが、今後三重予備など予備の多重化により高信頼化される可能性も考えられる。

## (2) 加入者線の網構成化による効果

システムオンチップ化によって極小規模交換機の経済性が考えられ、従来の電話局から交換機機能が加入者側により分散していくこととなろう。この結果以下の効果が期待できる。

- ① 加入者線交換機はマンホール、ビルなど加入者近くに地域分散され、加入者線の短縮が可能になる。単独加入者線部分が減少し、分散ノードによりトラヒック集束が図られ、網コストの低下に寄与する。
- ② 地域分散された極小規模交換機群は網を構成し、加入者線の高信頼化に貢献する。現在は加入者線ごとに比較的長距離の単独加入者線を引いているため、加入者線部分が通信網の信頼性の隘路となっている。小規模ノードに収容された後は網を構成するため、分散ノード間は回線多方路化が図られ、信頼性が向上する。

## (3) 交換機の仕様の簡明化

交換機にはシステム更改時にも既存の仕様との継承性が求められ、過去の仕様の上に新仕様を積み上げていくことから仕様が複雑化していく問題がある。例えばその要因の一つがサービス仕様である。不特定多数を対象とする公衆通信網ではサービス仕様の継承性が必須とされ、過去のサービスとして利用率が小さ

くなくても、また一部の地域にわずかに残った場合もサービス提供に向けたソフトウェアの維持管理が必要となる。他の要因としてハードウェアとソフトウェアのは行性の問題がある。交換機は寿命が長く、長い寿命の間にソフトウェアやハードウェアの大小の世代交代がある。このため同一機種ハードウェアが異なった世代のソフトウェアに対応し、ソフトウェアはまた複数世代のハードウェア機種に対応せざるを得ないという宿命により生じるものである。

高集積技術を活用した小規模経済的ノードの考え方は継承性を廃して通信網仕様を簡明化する要因となる。例えば小規模ノードは経済性が高く、初期投資は少なくすむ。この経済性を活かせば交換機の使い捨て、世代交代を容易に行い得る。この結果ソフトウェアと同期したハードウェアの世代交代が可能となり、ソフトウェアとハードウェアのは行性の問題が解決できる。また経済的なノードが実現できれば既存通信網との接続性を維持しつつ既存の通信網とは並列に身軽な通信網の実現でき、徐々に新しい通信網に移行できる可能性がある。

## 4. 高速通信サービスの普及へ向けたノード構成

ここでは小規模交換機を利用して高速・広帯域サービスを廉価に、また大衆サービスとして実現する方式について考察する。高速、広帯域サービスは、現在の64kビット/秒の基本接続サービスよりはるかに多くの網資源を使用する。このため高速・広帯域サービスを現在の音声通信並に大衆普及化させていくためには、現在よりはるかに効率的な網資源の運用を図らなければならない。

### 4.1 小規模加入者ノードによる高速通信サービスの低コスト化と課題

網資源の効率化の観点からはノードと加入者の間を単独回線で結ぶ加入者系の効率化が必要となる。中継階梯は集束されたトラフィックを扱うことから効率はずっと良い。

高速サービスの実現に向けては加入者線収容の加入者線交換機の配置を見直すことが必要である。加入者ノードはノードコストと加入者回線のコストの均衡点に設置されているが、これはメタリック回線と大規模化指向の交換機のコスト均衡点に適合化されて配置されたものである。今後高集積化によるノードコス

トの低下と高速回線の高価格化が進行し、現在の加入者線交換機の配置点の均衡は崩れることが考えられる。特に回線コストが割高になることから非集束の加入者回線を短縮することが必要になり、小規模加入者交換機を現在の加入者収容領域内に分散することが必要になる。

ただし、小規模交換機を地域的に分散すると加入者線の短縮化には大きな効果を生じるが、逆に交換機の小規模化はノード間の中継回線の利用効率の低下を招く。交換機はもともと回線を集約化して回線の設備の経済化を図る役目があるが、交換機の小規模化では回線集約効果を利用できず、回線効率が低下するという問題が生じる。小規模分散化を推進する場合、回線利用効率の低下を補完する手法の開発が必要となる。

#### 4.2 加入者網の対案

ここでは小規模分散化された加入者線交換機の回線利用効率等を向上する方式について検討する。小規模加入者線ノードを配置する代表的な通信網トポロジーはスター網とメッシュである。ここでは回線節約を図る網構成を検討していることから、メッシュ網は検討の対象外とする。

なお小規模加入者ノードによる分散網は現在の市内交換機領域内程度と考えられ、比較的小規模の網を想定する。このような小規模な通信網に近い通信網として構内通信がある。これにはLAN技術が進展している。これはノード間をバス結合するループ網である。このループ網はノード間をバス結合することから各ノードが小規模でも回線の利用効率が向上する。

一般的な通信網構成の代表としてのスター網とバス結合のループ網の概略評価を図7.6に示す。網を構成する上で最も有効なパラメータである等価回線長（物理的な総合線長とタイムスロット数の積）と必要なノードの規模で比較する。ループ網の等価回線長はバス結合により効率化を図っているにしては効果が小さいが、ここではノード間の回線利用を局建設時に固定化して設定することを前提としたためである。呼ごとに動的に回線運用を図る手法を開発すればループ網の回線利用効率はさらに向上する。

一方加入者網構成案の比較上、必要なノードの規模の大小も重要な要素である。現状の音声帯域の交換機では問題ないとしても、高速サービスを実用化していく

網イメージ			
等価* 回線長	100	1	0.9
所要交換 機規模*	10	9	1

図7.6 加入者網の対案 \* 相対値

呼率：基本呼(64k)0.1, 高速呼(1.5M)0.01  
 自ノード内：外呼の比率：0.1：0.9, ループ網は網外と分散交流

		中継でのTS並べ替え	
		並べ換える	スルー
高速 / 基本 呼TS 運用	分離	A1案 	B1案 
	混在	A2案 	B2案 
回線利用効率		A2 > B2 > A1 > B1	
遅延時間少		B1, B2 > A1, A2	
時間順序 制御容易性		B1, B2 > A1, A2	

図7.7 ループ網の対案 \* ———：高速呼TS群  
 ———：基本呼TS群

場合、交換機の規模の大小が通信網の経済性に大きな影響を与える可能性がある。高速通信サービスでは呼当りの交換情報が増大し、等価的に規模の増大を招く。高集積部品を有効に利用しないと高価な高速素子を多用する高価な大規模交換機を必要とする可能性がある。経済的なノードの構成にはMOS高集積技術が必須であろう。従来より加入者に近いノード、すなわち加入者線交換機は大衆的な部品技術を利用することにより経済化を達成し、成功を収めている。電子管で実現しようとした全電子式交換機は当時の最も大衆的な電磁部品による通話路と蓄積プログラム制御方式の電子交換機にとってかわられ、パラメトロンやコンデンサを用いた時分割交換機は高集積技術の進展まで実用化出来なかった。今後とも加入者ノードは最も大衆的な部品技術を利用して発展すると考えられる。

小規模ノードのみで構成しうる点、さらに回線利用効率化の観点からはノード規模を平均化しうるループ網が望ましい。スター網はその中心に大規模なノードを必要とし、このような大規模ノードは性能面からMOS系列の高集積部品の能力を越えるものとなり、経済的でない。

#### 4.3 ループ網方式の比較

回線の効率化を狙うループ網では各ノードで中継呼をどのように運用して効率化を図るかが課題である。中継呼の制御法の対案の分類要因には ①各ノードで中継呼間のタイムスロットの入れ替えを行なうかどうか ②高速呼と低速呼が存在する中で回線の管理を高速呼と低速呼を分離して扱うか／一体として扱うかがある。一方、ループ網における高速呼の制御法に関して考慮すべき点には ①時間順序保存の実現と ②ループの多段交換に伴う伝搬遅延の大小の問題である。

上記の観点からループ網構成方式には図7.7の4対案があり、この対案の中ではB2案が有効と考えられる。B2案はタイムスロットを入れ換えないため、遅延時間を短縮できる。また各ノードで順序を入れ換えないため、時間順序保存の制御が簡単になり、中継制御を簡単化できる。ただし、B2案ではループ網全体を統一に運用管理する管理ノードを必要とする。

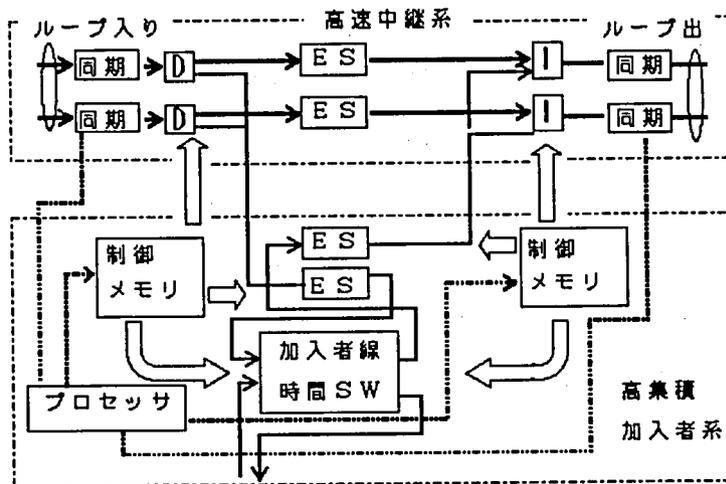


図7.8 加入者ノード構成例

ES: エラスティックメモリ  
 D: TS抜取り、I: TS挿入

#### 4.4 ノード構成例

前節で望ましいとしたB2案のノード構成の具体化を検討する。この構成では中継ループの回線上ののタイムスロットをすべて均質として扱い、低速呼も高速呼もループの回線上の任意のタイムスロット選択する。各ノードの中継点ではタイムスロットの入替えを行わず、中継する呼の到着順に送り出す構成である。B2案のノード構成の概略図を図7.8に示す。

ここではループ網とインタフェースをとる中継部と加入者を収容して加入者間の交換を行なう加入者交換部に大きく機能分割している。それぞれの機能ブロックの概要は以下の通りである。

##### (1) 中継部の構成

- ① 中継対象の呼は網同期に合わせる時間調整のエラスチックメモリを経由するだけで短いノード滞留時間で時間順序を保存して通過する
- ② 中継ループと自ノードの出入り制御のためには中継ループのタイムスロット管理情報に基づいたタイムスロット情報の抜取り機構（ドロップ）／挿入機構（インサータ）を持つ。抜取り／挿入回路には制御メモリ持つ。フレーム同期に合わせたタイムスロットのカウンタの値をアドレスとした該制御メモリを持つ。制御メモリの各語にタイムスロット毎の抜取り／挿入／通過の情報を持ち、この情報に従って自ノード落ち、自ノードからの出呼に対する操作を行

なう。この制御メモリの値は蓄積プログラム制御部プログラム論理で設定する。

③異なった中継ループ間や中継ループと時間スイッチ間の位相差の制御には蓄積プログラム制御機構を利用する。プログラムが網同期を認識して、抜取り／挿入制御や交換スイッチの位相差を算出して制御メモリのアドレス操作を個々に行う。位相差に合わせたアドレス情報の変更を行い、全体の同期をとる。

この中継インタフェース機能ブロックは高速の情報の流れを操作するため、高速素子を必要とするが、制御論理が簡単であるため回路規模は小さくでき、高価な高速素子の利用率を下げられる。

## (2) 加入者線対応部

加入者線対応部は既存の加入者線デジタル交換機の小規模版である。この機能ブロックは回路規模の大きい時間スイッチや信号機能を含み、しかもこれらを回線対応に必要とするため全体としての回路規模は多くなる。しかし、速度については自ノードに関する伝達機構のみであり、制御や情報転送は比較的低速でよい。したがって素子速度は比較的低速でよい。回路規模は大きいですが、素子速度は比較的低速でよく、MOS系素子が利用できる。システムオンチップに適した分野である。

## (3) 中継部と加入者部の整合

回路規模は小さいが、高速の中継部と大規模回路で低速の加入者対応部の速度整合も実現上の課題である。これらの機能ブロック間の伝達情報は高速と低速のエラスティックメモリを経由して速度整合を行い、転送する。中継部と加入者部の制御メモリがエラスティックメモリの入出力制御を行なう。

以上のように伝送線の終端機能と交換機の直結化を図り、伝送機能と交換機機能を統一したプログラム制御で統合し、回路構成の単純化と高速呼の制御の容易化を図っていく必要がある。以上の構成によれば高速サービスに必須となる高価な高速素子の利用は単純な中継部に限定でき、加入者交換機側は廉価な高集積部品を用いて実現できる。小規模ノードの構成要素として今後重要になる可能性がある。

#### 4.5 呼制御の論理化と階層化

ループ網の比較ではループ網全体を統一管理する管理ノードが前提であった。この管理ノードはノード間のルーティング機構や加入者番号の翻訳データベースを持つ。しかもノード間のタイムスロット割当等のルーティングだけでなくループ網全体をあたかも単一ノードのように論理的に制御する。この技術はデータベースを用いた呼の論理制御化であり、移動体交換サービスや着信課金などの高度な通信サービスとして一部の分野では実現されている。この技術の延長として網分散した小規模ノード間で局番を統一し、複数ノード全体を仮想単一ノード化して制御することが必要である。

制御の論理化や階層化が進展して仮想単一ノードが進展すると、分散したノード全体を管理ノードが統一して管理し、この結果加入者番号が回線物理番号から論理移動番号への展開が可能となる。個人番号等のパーソナルIDが公衆網でも実現可能となる。またデータベースを網的に重複配置し、ループへの出入りを複数点設ければノード間を多重化した回線群で接続でき加入者網の多重収容が可能となり、網的信頼度がさらに向上する。

### 5. システムオンチップ化の課題と実現手法

今後の通信網の構成において超大規模のLSIの利用は大きな魅力であるが、その集積規模はシステムレベルに達し、オンチップシステム向きのアーキテクチャを検討して望ましい伝達システムを実現しなければならない。また数100kゲートに及ぶ大規模なオンチップ回路を高精度で実現するにも多くの困難な問題を伴う。ここでは方式設計段階でオンチップシステム設計支援を行なうアーキテクチャシミュレータについて述べる。このシミュレータを便宜上ASCOT (Architecture Simulator for Communication Transfer Node) と名付ける。

#### 5.1 オンチップ対象とするデジタル伝達ノードアーキテクチャ

通信網のノードでは信号の再生中継や方路の群分けを行なう伝送機能および所望の通話要求者間を確実に接続してサービス運用を行なう交換機能からなる。デジタル化されたノード構成では、図7.8に示したように回路機能上はもはや

明確な伝送・交換機能の分離は意味が無く、これらを一体化して実現することにより融通性に富む網が構築できる。ここでは従来の伝送・交換機能を統合して実現した通信ノードを伝達ノードと呼ぶ。

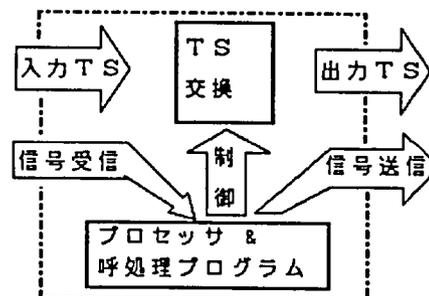


図7.9 デジタル交換機概念図

デジタル伝達ノードでは回線を終端し、網同期との整合を行い、所定のタイムスロットに関する信号を得て、呼処理プログラムが入線タイムスロットから出線タイムスロットへの交換則を定め、時間スイッチや空間スイッチに交換則を設定することにより通話を成立させる。伝達ノードアーキテクチャとは呼処理プログラムが操作しうるタイムスロット伝達構造と考えられる(図7.9参照)。今後、高度な通信サービスの提供に向け、また、高集積能力を利用した廉価なサービスを提供して行くために、新しい伝達ノードアーキテクチャの研究を進めることが必要である。

## 5.2 デジタル伝達ノードのオンチップ化の問題

デジタル伝達ノードは ①メモリ応用システムである ②小規模から超大規模までの適用領域があり、集積度の進展をいかようにも吸収しうる特徴がある。デジタル伝達ノードは超LSI利用の有望な応用分野でもある。これをオンチップシステムとして実現するとき、従来のシステム開発に対する相違点として以下の問題を考えておくことが必要である。

### (1)仕様、回路凍結時点の早期化の問題

試験工程に注目し、従来の手法と想定されるオンチップシステム開発手法を対比したものを図7.10に示す。既存手法では大規模回路を分割し、装置として構成し、複数技術者が独立に責任を持って、並列開発を行なった。装置の組合せとして構成されたシステムでは装置間インタフェースの修正の可能性を残し、システム総合試験段階でソフトウェアとのインタフェースの修正の可能性を残していた。LSI外部の回路はシステム総合試験の最終工程まで修正が不可能では

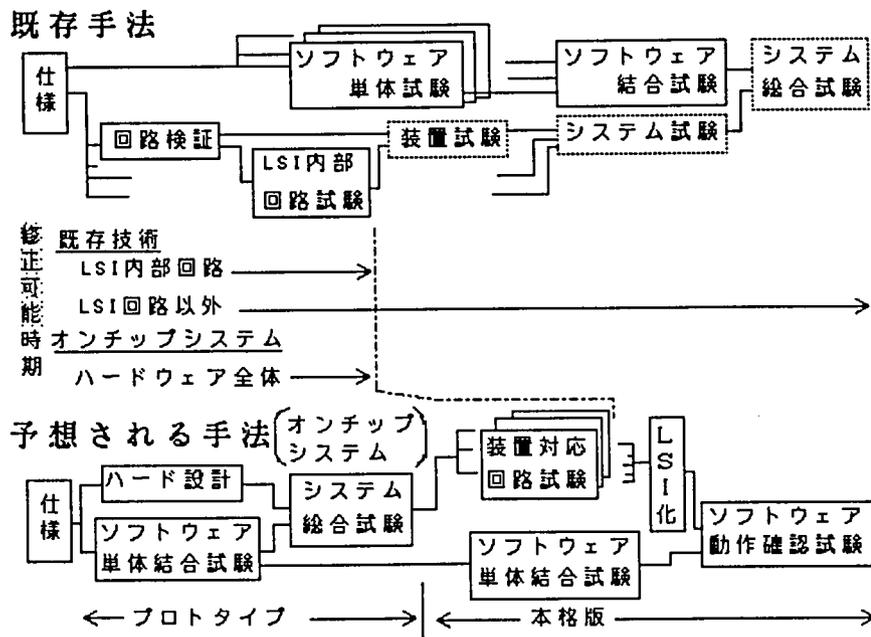


図7.10 開発工程の各種試験と修正時期

なかった。

一方、オンチップシステムではチップ内の回路修正はチップ上で出来ないため、回路凍結時点にソフトウェアとの相互試験を行なわなければ完全にバグが取りきれないという問題を残す。すなわち、ソフトウェアはチップ（システム）が完成しないと試験が出来ないのに、チップ内回路はソフトウェアによる試験を完了しないと回路凍結はできないという相互矛盾となる。

これを解決するには呼処理プログラムを含むソフトウェアのプロトタイプ化が必須である。回路凍結以前に少なくとも呼処理プログラムプロトタイプによるソフトウェア／ハードウェアの相互試験を終え、密度の濃い試験を行なっておく必要がある。

## (2) システムのチップ化の問題

従来のシステムでは装置と呼ぶ実装単位を構成し、これらの間は極力疎な結合インタフェースとし、仕様の明確化を行なってきた。一方、オンチップシステムではすべてがチップ内にある。このためオンチップシステムでは旧装置段階に相当して仕様を明確化するモジュール化の新しい規範を作ることが必要であ

る。このため、チップ内伝達ノードシステムアーキテクチャの構成要素を明確にする手段が必要である。

### (3)大規模性による問題

VLSIプロセッサの開発の経験から装置段階のモジュール内に閉じた試験だけでは不十分であり、装置の組合せに相当するシステム全体の複合事象のデバッグは必要であることが示されている。論理回路には論理シミュレーションという有力な手段があるが、論理回路がシステムレベルまで巨大化したときシステム全体をシミュレートすることは論理規模の大きさや実行時間から不可能と考えられる。また機能の大規模さから全体を試験しうる試験項目の設定さえ困難になると考えられる。

このため、試験の階層化が必須となる。初期段階では(2)で述べたモジュール間の結合仕様を試験し、結合仕様を試験することが必要である。このモジュール間結合仕様を試験した後は、それぞれの回路は既存のVLSI程度に集束することから既存の開発、デバッグ手法の利用が可能となる。

### 5.3 アーキテクチャシミュレータ "ASCOT" の概要

以上のオンチップシステム開発上の問題を解決するには、システム開発の最前段でプロトタイピングを行なう必要がある。これにはソフトウェアによるシミュレータが有効と考えられる。ハードウェアを含むプロトタイピングには多くの人手と投資を必要とし、しかも実現しても1対案しか確認できず、望ましい構成を規定することは困難である。このソフトウェアによるプロトタイピングを支援するのがASCOTである。

アーキテクチャシミュレータ"ASCOT"のシステム擬似概念を図7.11に示す。

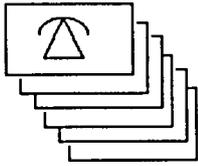
ASCOTはワークステーションとターゲットシステムからなる。擬似する伝達ノードのうち、伝達アーキテクチャの擬似機構と呼処理プログラムはターゲットプロセッサに配置され、ターゲットシステム上で動作する。ワークステーションは呼処理ソフトウェア、各種伝達アーキテクチャ擬似プログラムを作成支援するとともに、擬似端末として動作する。

伝達アーキテクチャは時間スイッチや空間スイッチや信号装置およびハイウエー等の機能要素で構成されており、これらの機能要素は予め部品としてライブラ

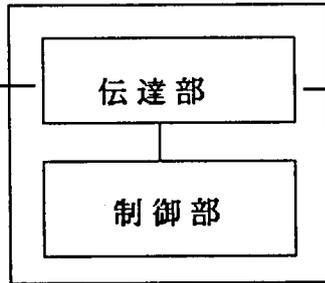
(1) 検討対象

システム

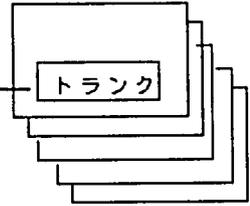
端末



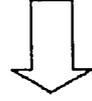
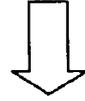
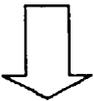
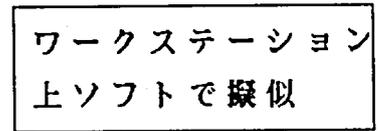
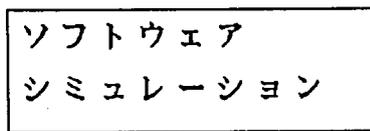
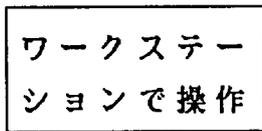
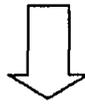
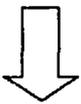
伝達ノードシステム



相手局

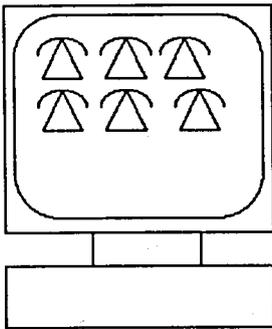


(2) 実験の  
考え方

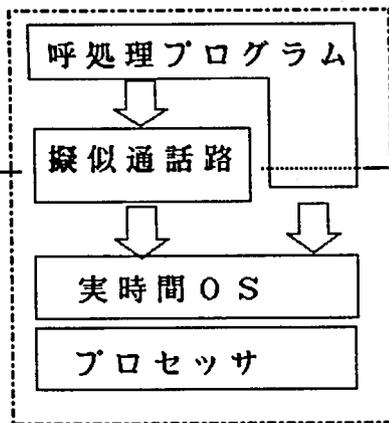


(3) 実験

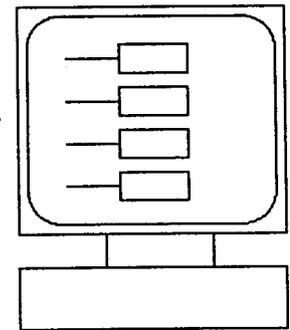
システム



ワークステーション  
(端末擬似手順  
入出力操作環境)



ターゲットプロセッサ



ワークステーション  
(局間手順擬似)

図7.11 アーキテクチャシミュレータ ASCOT の概念図

りに登録しておく。部品はソフトウェア上ではタイムスロットを伝達できる隣接部品を示す部品間接続情報と該部品でタイムスロット交換を行なう変換則、さらに呼処理プログラムからアクセスされた場合に応答すべき機能を記述した関数からなる。新しい伝達構成の案が考えられた場合はこれを新規部品としてライブラリに登録する。

アーキテクチャ擬似のためにはこのライブラリから任意の部品を取り出して集成してシステム機能を構成する(図7.12)。また同時にこの擬似伝達機構を制御する呼処理プログラムもプロトタイピングで実現し、トータルシステムとして動作確認を行なう。この手順に基づいて各種伝達機構を構成し、性能やソフトウェアから見たアーキテクチャの複雑性や回路規模を想定し、望ましいアーキテクチャを決定する。

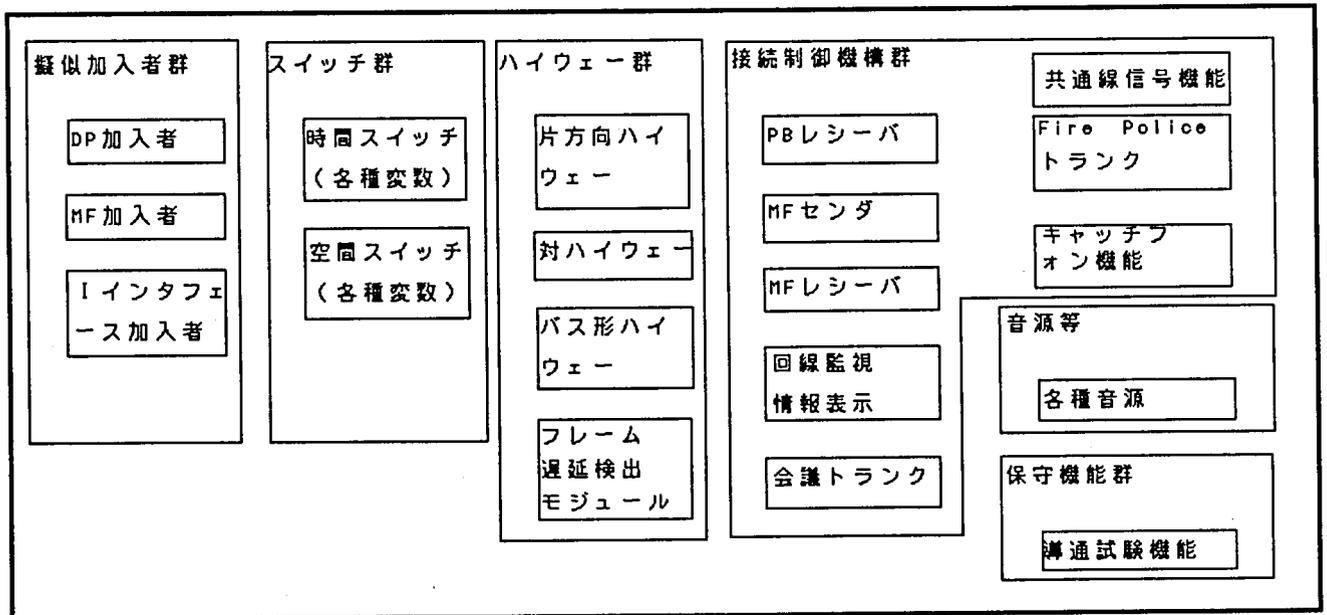
#### 5.4 アーキテクチャシミュレータ "ASCOT" の役割

"ASCOT"を用いたシステム開発手順の概念図を図7.13に示す。アーキテクチャシミュレータ"ASCOT"は伝達ノードの新しいアーキテクチャを追求する手段である。

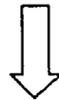
ソフトウェアを含めたアーキテクチャを研究するには伝達ノードハードウェアの実物が必要であり、ハードウェアを実現するには多くの時間と投資を必要とする。しかもハードウェアを実現しても1対案しか試みられないという問題があるが、ソフトウェアでアーキテクチャを擬似するASCOTは複数の対案を容易に実現する手段である。アーキテクチャシミュレータASCOTはソフトウェア/ハードウェアを含めてシステム段階での対案を比較し得るラピッドプロトタイピングを行なうための手段でもある。また、同時に上記のオンチップシステム開発の問題解決に有効な手段となる。プロトタイプ化手段として早期の段階でソフト/ハードを含めた仕様検証を可能とし、"ASCOT"で定めたノードシステム交換機部品が従来の装置に対応するモジュールと考えればマクロな機能ブロックの仕様確定が容易になる。この機能ブロックを段階的に回路に落としければよい。

さらに、"ASCOT"で実現した呼処理ソフトウェアは実時間動作デバッグのために必要なハードシミュレータの試験プログラムとして有効となる。ASC

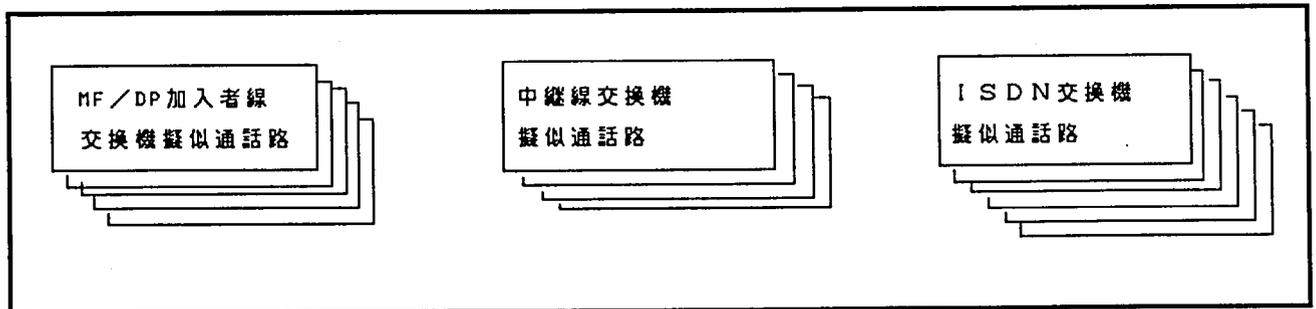
# 擬似通話路部品ライブラリ



分散モジュールへの集成



部品集成と部品間結合



実験システムへの搭載



分散モジュール間結合

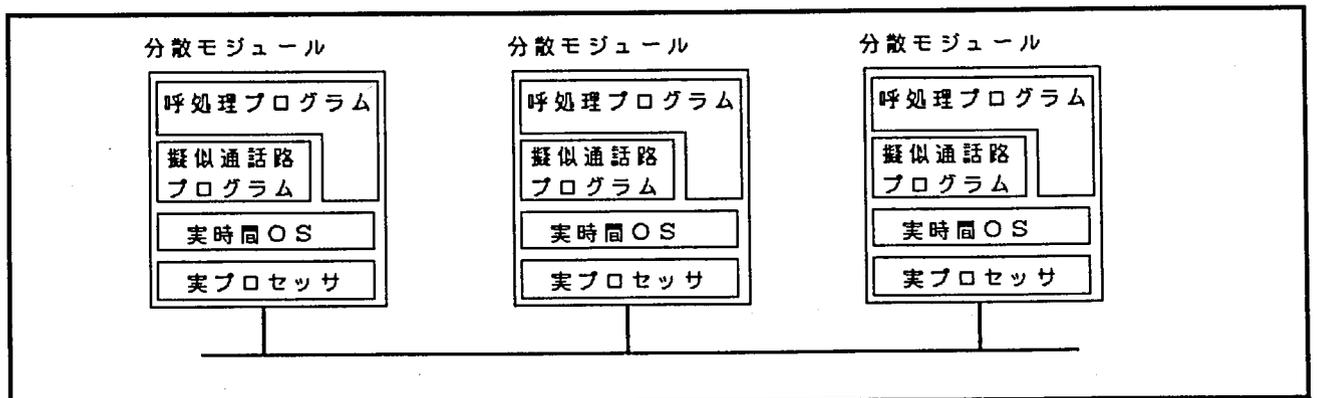


図7.12 擬似通話路構成手順

OT上で実現した呼処理プログラムは伝達アーキテクチャ仕様を守るものであり、伝達アーキテクチャをハードウェア化した場合は呼処理プログラムはハードウェア化した伝達アーキテクチャを制御する。 実時間試験を行なうた

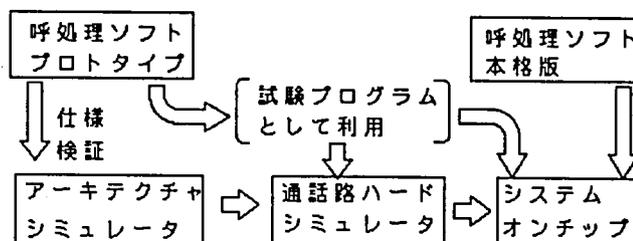


図7.13 オンチップシステム試験手法

めのLSI回路のハードウェアシミュレータやオンチップシステムが完成したときオンチップシステム仕様の検証に利用できる。

## 6. 今後の分散システムの動向と研究課題

### 6.1 分散システムの動向

交換機のデジタル化を契機に交換機の分散化が進展している。 交換機の分散システムの進化を筆者の分類に従い図7.14に示した。 各進化の段階には背景となる条件および背景となる技術が存在し、これに対する狙い（技術動向）が存在する。

第1世代：分散システムの考え方は制御系から始まった。 この段階の研究経過は本論文で既に述べた通りである。

第2世代：加入者線までのデジタル化が進んでISDN統合交換機へ具体化が始まっている。 この段階でのデジタル交換機はデジタル加入者線やパケット交換や高速広帯域などの多様な交換サービスを一列交換機として統合して実現するものとされる。

前段階で制御の分散に留まったものが、この段階では制御と伝達を一体とした分散モジュールを構成し、分散モジュール化がより進展している。 各モジュールはアナログ音声を扱うアナログ交換機能やデジタル1リンクを実現するISDN交換機能や高速広帯域交換機能を構成する。 また各交換機能内では規模に応じたビルディングブロック化を図る。

この分散化の要因として、アナログ交換機やISDN交換機や高速広帯域交換

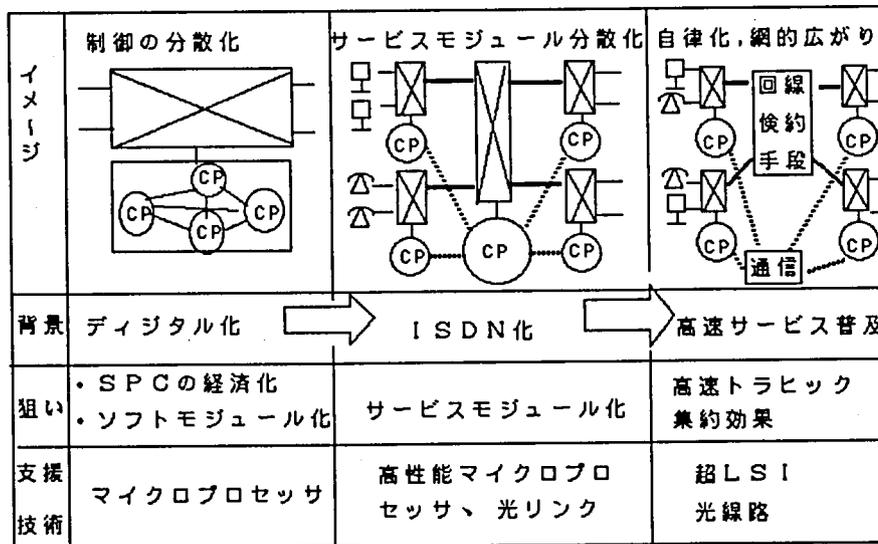


図7.14 交換機における分散化の進展

機の間では、加入者線も中継線も結局それぞれ別の回線を取容するもので、おの必要とするリソースが異なって統一制御を行なう意味が無かったことによる。

また技術的にはソフトウェアの複雑性等からソフトウェアは極力独立モジュールとして構成しようとする狙いがある。これを可能とした技術的背景としてマイクロプロセッサの性能の進展から、制御と伝達を合わせて一体モジュールとして構成してもプロセッサの性能ネックが生じないこと、またプロセッサ間の通信リンクが光ケーブルの導入により高速化され、ビルディングブロック間の呼接続遅延を少なくできるようになったことにもよる。

第3世代：本章で示した加入者ノードの分散化がこれに対応する。これまで局内に閉じていた分散は小規模システムの経済化により、網的分散に進展することが必要と考えられる。既に述べたが高速通信サービスが大衆普及するには加入者線の経済化が必須であり、ノードの分散化による加入者近くでのトラフィックの集束が必須となる。高速通信サービスの大衆化には必須であり、今後必ず進展すると考えられる。

現在大都市の通信サービスの高度化に向けてはMAN (Metropolitan Area Network) として研究されている。しかし、これはビジネス利用に向けて特化し、L

ANから派生した段階である。やがてオンチップシステム技術が確立し、大規模な光ネットワークが手の内の技術となるときには公衆網としての高速通信サービスの一大普及が望まれ、網的分散ノードによる公衆網の高速通信サービスが新たな展開を向かえると考えられる。

## 6.2 今後の課題

ISDN交換機は64 kb/sの基本帯域でデジタル1リンクを達成した。より多くの情報を求めるのは人間の習性であり、デジタル通信サービスは帯域圧縮を行ないつつ、1 Mb/s帯域の会議用の動画、40 Mb/s帯域の商用テレビ、100 M/s帯域の高精細テレビなどの通信の高速化は進展していく。

このような超高速のサービスを全ての人々が享受するのはまだ遠い段階であるが、高速サービスを普及させるには本章で述べた小規模ノードの分散化が必須と考えられる。この観点からは筆者は以下の点が課題と考えている。なお通信サービスには通信網だけでなく、端末があって成り立つものであるが、画像・映像を扱う機器は液晶テレビ、レーザーディスク、ビデオカメラ、携帯コピー機、ビデオコーデックなど進展はめざましく、極力仕様の軽い通信網を実現して行けば当然に普及すると考える。

公衆通信網の発想の転換：公衆通信網は巨大資産であり、巨大な投資を必要とするの発想で各種の規定・設計・導入がなされている。通信網の技術者には既存通信網の概念、仕様にとらわれ、大規模ノード指向、継承性の上での機能の拡大を求める傾向が強い。既存の継承性の上に新しい機能を作り上げて行くといずれはマンモスの滅亡と同様の経過をたどると考えられる。手動電話以来営々として築いてきた公衆網であるが、今や高集積技術を利用した身軽なノードによる通信網への改革の時代である。使い捨て交換機や仕様の捨てられる通信網を実現していくことが課題である。通信網のワイヤ部分はそのまま利用してもノードを取り替え、機能の変更が可能な進化しうる公衆通信網の実現が課題である。

分散ノード技術の確立：機能の面では複数ノードを単一仮想ノードとして実現する網管理の論理化がまず必須である。現状の市内交換機を複数ノードに分割する場合には同一局番を複数ノードで共有する必要がある。将来も個々の小規模ノードを単一論理ノードとして扱うと量的な面での通信網仕様が複雑化する。

複数ノードをまとめて管理する網データベースで番号翻訳，ルーティング等の回線管理を行なうことが必要になろう。これは将来の通信網の個人番号，移動端末サービスへもつながっていく。通信網は伝達機能の分散化と論理的集中化の調和によって高度化することになろう。

高速通信の伝達モードについては同期系 (Synchronous Transfer Mode) と非同期系 (Asynchronous Transfer Mode) の研究が進行しているが，サービス普及の観点からは加入者ノードをいかに経済的に実現するかが課題である。これには高集積技術の適用が課題であり，高集積技術との親和性が伝達モード決定の大きな要素になろう。

さらなる高集積化：膨大な情報を高速で，複数回線間を交換するためにはより

回線間の情報交流を効率化する必要がある。これを実現しうるのは高集積技術である。現状の平面 L S I から三次元 L S I などの一層の高集積化を進め，実装遅延を少なくして高密度性から多くの回線間の交差，交換を可能とする必要がある。

光技術と論理の親和化：大容量の情報伝達のためには今後大規模な光伝送が必

須となる。光のみの伝達と制御論理で通信網を実現するのは理想であろうが，高度な論理を通信ノードで実現するには光部品だけでは困難と考えられる。機能素子は今後とも高集積電子回路に依存するため，論理を実現する高集積回路上が直接光-電気変換機構を具備する可能性も考慮したい。

ソフトウェアの簡明化：通信網の機能仕様は呼処理ソフトウェアが具現する。

今後ますます大規模化するソフトウェアに対し，簡明化は大きな課題である。ただし，実現すべき機能が存在する以上，ソフトウェア技術だけのソフトウェア問題の解決は困難と考えられる。むしろ方式の支援により，ソフトウェアの簡明化を行なっていく必要がある。サービス拡大に向けて通信網仕様を着実に増大し続けている中で，複雑なシステム構成や既存仕様の継承性維持によるソフトウェアの負担は大きなオーバーヘッドである。高集積素子のインパクトを利用してシステム構成の単純化や保守の単純化が必要であろう。

将来の通信網に対して予測困難な問題はあるが，高速通信サービスは近い将来に C C I T T 標準の I インタフェース一次群サービスとしてまず普及する可能性

がある。この段階でも本章で述べた分散システムは特に有効である。オンチップシステムも射程内と考えられる。

また低域帯での高速通信サービスを実現するには、普及の手順も重要である。大規模な投資を必要とする光線路の導入に対して、比較的簡便に導入できる無線系のノードを併用することも重要と考えられる。無線基地の送受アンテナ間に本章で述べた伝達ノードを埋め込めば効率の良い交換機能を含む無線網が実現でき、融通性に富み、効率の良い網が実現できる。現状ISDNが大都市近傍でのみのサービス提供であるが、無線網の併用をすれば早期にISDNサービスの普及を遂げられると考えられる。

## 7. 結言

シーズとニーズ面から交換機制御方式の発展動向について、筆者の見解を述べた。筆者が進めている研究自身まだまだ途上のものであるが、高集積化と分散化は必然の流れと考えられる。本章で述べた範囲での結論を以下に示す。

- (1) 高集積素子を有効に活かすためには交換機全体の予備構成を見直さなければならない。これにより、交換機全体のシステムオンチップ化は可能となる。
- (2) システムオンチップ化が進展すると通信ノードは網的に分散され、加入者線の短縮が可能となり、網が高信頼化し、高速サービスを廉価に実現できる。
- (3) システムオンチップ化が進展するとソフトウェアとハードウェアを一体としたモジュール化が進展し、モジュール間の結合の高速性から高速サービス普及の時代に必要となる超大容量中継交換機の実現の可能性がある。
- (4) 小規模ノードの分散化に当たっては分散ノードをループ結合し、多元タイムスロットを網的に運用する加入者線網が有効と考えられる。
- (5) ノードの分散化は着実に進展しており、分散範囲の拡大と独立モジュール化の方向をたどっている。今後オンチップシステムを想定した網的分散を推進する必要がある。
- (6) システムオンチップには困難な問題を含むが、通話路ハードウェアを含むシステムをソフトウェアでシミュレートするアーキテクチャシミュレータから、ハードウェアシミュレータ、チップの実現へ進むことが有効と考えられる。

## 第VIII章 結論

1977年より日本電信電話公社武蔵野電気通信研究所で進められたデジタル交換機用分散制御方式の研究の中で、筆者が担当したマルチプロセッサ制御方式に関する研究を本論文にとりまとめた。まず第I章では本研究の歴史的背景を、第II章では本研究の前提条件を述べた。筆者の研究は第III章以降である。

各章でも既に結論を述べているが、ここで全体のまとめとして結論を要約する。

### (1) マルチプロセッサの基本構成

交換機実現上の基本条件である①リソース管理を効率化しうるマルチプロセッサ、②実時間制御を維持し得るマルチプロセッサ、③将来の機能拡大性を具備しつつ、機能拡大の以前には予備投資を極力削減し得るマルチプロセッサを定量的に追求した結果、①中継回線や信号装置などの共通リソースを共通メモリ上で管理すべきこと、②ハードウェアに対応する信号処理と論理的な処理である呼制御処理を機能分散すべきこと、③信号処理を信号方式対応に機能分散することが望ましいことを得た。

### (2) マルチプロセッサの基本設計

極小規模で必要なシングルプロセッサと大規模で有効なマルチプロセッサを一機種で実現する ONE-MACHINE 概念の具体化を図り、本概念に基づくシステム設計を行なった。ONE-MACHINE 概念の具体化は制御システムを階層構成に分解し、各階層毎にシングル/マルチプロセッサの構成要素を共通化し、両構成で共通要素の抽出と差分の具体化を図るものである。この概念に基づき、マルチプロセッサを実現し、試験評価を行い、望ましいとしたマルチプロセッサ基本構成と ONE-MACHINE 概念に基づくマルチプロセッサの有効性を確認した。

### (4) マルチプロセッサの高信頼設計

交換機用マルチプロセッサの高信頼化に向け、フェイルソフト概念に基づく耐故障設計の思想を構築し、負荷分散と対応づけた交換機フェイルソフト設計を具体化した。またフェイルソフトに基づく高信頼対策の結果、デジタル交換機

で新たに設定された「大きな交換機ほど高信頼であるべき」とする規模別信頼度規定へ適合していることを示した。

#### (5) マルチプロセッサの故障回復法

既存のシングルプロセッサよりはるかに複雑となるマルチプロセッサ故障処理について、故障回復手順の階層化構成の思想を構築し、故障回復処理の単純化を実現して故障回復を確実化する手法を示した。また故障回復の確実さは既存のシングルプロセッサと同じとしても故障による中断時間はマルチプロセッサではより小さくできることを示した。

以上の結果は現在すでに日本の電気通信網における交換機として商用に大量に供されているD70デジタル交換機の制御系に用いられ、その実用化に寄与した。

現在デジタル交換機の実用化は一段落を遂げたが、急速な技術進展の中で通信網は大きな転換期を向かえている。従来は継承性が大きな要素を占めていたが、情報社会の要求やデジタル化、使い捨てを促す高集積化の技術の中で、今や通信網の革新化が求められていると考えられる。制御方式についても単にプロセッサとソフトウェアにとどまらずデジタルノードシステムとして複合化へ向けた研究を進めて行かなければならない。この観点から今後の研究課題を第VII章に述べた。

すなわち今後の通信サービスは高速通信サービスの普及を重点に考える必要があり、また技術面では超高集積部品の進展が大きなインパクトを与える。ただし単純に高集積部品を適用しても交換機には高集積効果の利用限界があり、システムオンチップ化、通信ノードの分散化を進めること、これによって高速通信サービスを廉価に実現でき、高速通信サービスの普及を促すことを述べた。方式技術としてシステムオンチップ化は未踏の技術であり、手順を踏んで超大規模回路のオンチップ化を実現する必要がある、ソフトウェアシミュレータに始まるシステム構築手法を提案した。

以上、本論文はシステム実用化への問題の設定、概念の構築、方式選定に係わる手順、機能実現手法に関する方式研究について述べた。一般に方式技術は重点をどこに置くかにより、結論は変わる。唯一解が存在するものでなく、一般性を述べるのもなかなか困難である。多くの技術者がこの分野に従事しながら論文としてまとめられた物は少なく、学問体系化は難しい。筆者の本報告も実用システムとしての必要十分条件を議論しているが、唯一解の証明ではなく、一面ではケーススタディでもある。しかし、筆者は本論文の類が積み重ねられて、さらに大きな体系化がなされて行くと考えている。

社会のインフラストラクチャとして電気通信網の重要性はますます高まっており、今後新たに多くの通信ノードシステムを構築して行かなければならない。技術の積み上げが必要であり、方式技術の体系化が求められる。本論文が発表されずに埋もれた技術の論文化、さらに体系化への刺激となり、またシステム構築技術としていささかでも役に立てば幸いである。

## 謝 辞

本研究をまとめるに際し、あたたかい励ましと懇切なる御指導を頂きました京都大学工学部 池上文夫教授に深謝致します。また、在学以来、研究の進め方を含めて御指導頂きました高木俊宜 京都大学名誉教授、京都大学工学部 佐々木昭夫教授に厚く感謝致します。

本研究は著者がNTTにおける業務の一環として研究したものであり、本研究遂行の過程で直接ご指導頂いた当時の研究室長 保坂務博士（現在 NTTエレクトロニクステクノロジー株式会社営業部長）ならびに丹羽昭男氏（現在 NTT LSI研究所主席研究員）に感謝致します。両指導者には筆者の未熟な研究提案を採用し、推進して頂き、また研究の直接の御指導を頂きました。提案技術が実用システムに採用され、実用システムとしてサービスに供されることは技術者の本望であります。本研究がここに至ったのは両指導者のご推進のおかげであると感謝しております。

また、交換機用マルチプロセッサの実現に向けて研究実用化を共に進めた方々、ご協力頂いた方々に感謝します。デジタル交換機プロジェクトは極めて大規模なシステムであり、多数の方々の御貢献の成果であります。ここですべての方々の御名前を挙げる訳にはいきませんが、中でも初期のマルチプロセッサ研究の段階より討論や御協力を頂いたネットワーク開発センタ須山正人主幹技師、交換システム研究所三木修次主任研究員に感謝致します。さらにマルチプロセッサの最初の試作である所内試験機を筆者が担当した後、現場試験機、商用試験機を完成に導かれましたネットワーク開発センタ主幹技師 菱沼千明博士、VLSIプロセッサチップ開発を実質的に推進された伊藤芳孝氏（現在 NTTエレクトロニクステクノロジー株式会社研究部長）、設計支援関連を担当された交換システム研究所 松浦洋征主任研究員、VLSI制御系の開発を実質的に推進されたネットワーク開発センタ 福田晴幸主幹技師に感謝します。また、本マルチプロセッサのソフトウェアの実現を推進された堀好徳氏（現在 NTT技術移転株式会社 技術部長）、ネットワーク開発センタ 小川聡主幹技師、交換システム研究所 渡辺晴光主幹研究員の御討論、ご助言、ご協力に感謝します。

本研究をまとめる機会を与えてくださった上司の方々，国際電気通信基礎技術研究所所長 葉原耕平博士（元基幹交換研究部長），NTT通信網総合研究所所長 五嶋一彦博士（元基幹交換研究部長），交換システム研究所所長 池田博昌博士（元基幹交換研究部長），交換システム研究所交換方式研究部長 青木利晴博士（元基幹交換研究部統括役）に感謝致します。

また，筆者の研究生活においては，NTTにおける筆者の研究の進路を導いて頂いた東京工科大学教授 伊吹公夫博士（入社当時の研究室長）に感謝します。

また，筆者がハードウェア研究者としての最初の仕事となったD10高速中央処理装置データチャネル装置に関する提案を受け入れてくださった当時の研究室長 鶴野忠一氏に感謝します。筆者がこのような研究成果をまとめることができるのも両先輩のお力添え，御導きによるものであります。

## 参考文献

— 自著を〔〕で、その他を（）で示す —

- (1) W. Keister, R. W. Ketchledge and H. E. Vaughan: "NO.1 ESS :System Organization and Objectives" B.S.T.J., 43, pp.1831, Sept., 1964.
- (2) 川田: "電子交換研究の目的と概要", 通研実報 電子交換特集号, 第16巻 第11号, 1967.
- (3) 高山・藤本・城水: "D E X - 2号交換機のシステム設計", 通研実報 D E X - 2号電子交換機特集号, 第18巻 第10号, 1969.
- (4) 高原・岸上・城水・高村・渡部・増村・大和久・豊田: "D E X - 2 1 電子交換方式", 通研実報 D E X - 2 1号電子交換機特集号, 第20巻 第3号, 1971.
- (5) 伊吹: "電子交換機の現状と将来", 情報処理, Vol.11, No.5, pp.285, May, 1970.
- (6) R. E. Staeler: "Organization and Objectives", Special issue on 1A Processor of B.S.T.J., Vol.56 No.2, pp.119, Feb., 1977.
- (7) 神宮司・保坂: "D 1 0用高速中央処理系装置の概要", 通研実報, Vol.26, No.12, pp.3235, Dec., 1977.
- (8) J. E. Yaetes and G. P. Zahorodny: "Powerful features for a small ESS", Bell Record, Vol.58 No.7, pp.219, Julu/Aug., 1980.
- (9) 秋丸・楠・山本: "D E X - A 1 電子交換方式", 通研実報 D E X - A 1 電子交換機特集号, 第22巻 第1号, 1973.
- (10) 矢頭: "中小局および遠隔制御電子交換方式の実用化", 通研実報, 第23巻 第5号, pp.761, 1974.
- (11) 加藤・加藤・保坂・鈴木: "D 2 0自動交換機の改良", 通研実報, 第28巻 第3号, pp.337, 1979.
- (12) 畑・内山・松井・広川・上野: "D 3 0自動交換機", 施設, Vol.32, No.3, 1980.
- (13) T. Hosaka, A. Niwa and T. Yamada: "High Speed Processor for Wide Area Control", International Communication Conference, ICC'77, Ses, 20.3 June, 1977.
- (14) A. Niwa, T. Yamada and N. Ohnishi: "A High Speed Central Processor for D10 Electronic Switching System", Review of ECL, Vol.25, No.5-6

pp. 379-389, 1977.

- [15] 山田・畑・山田：“D10高速中央処理装置データチャネル装置の構成”，  
通研実報， 26巻12号， pp.3261-3280, 1977.
- [16] 山田・山田：“データチャネル高能率制御の一方式”， 昭和51年度  
信学会総合全国大会 1447.
- [17] 山田・山田：“DCHマイクロ命令効率化の一検討”， 昭和51年度電子通信学  
会通信部門全国大会 81.
- [18] 山田・畑・山田・埴淵・若林：“データチャネル制御方式の一評価”，  
昭和51年度信学会通信部門全国大会 82.
- [19] 山田・山田・矢沢・桜井・大川・兎耳山：“高能率データチャネル装置の一  
構成”， 昭和52年度信学会通信部門全国大会 83.
- [20] 山田・畑・若林・大川・古川・梶塚・淡路：“D10用高速中央処理装置データ  
チャネルの構成”， 昭和52年度電子通信学会総合全国大会 講演番号1543.
- [21] 山田・山田・埴淵・岡・庄田・梶塚・淡路：“D10用高速中央処理装置データ  
チャネルのマイクロプログラム制御”， 昭和52年度電子通信学会総合全国  
大会 講演番号1544.
- [22] 山田・若林：“多重制御型マイクロプログラムの融通性利用”， 昭和51年度  
電子通信学会通信部門全国大会 80.
- [23] 山田・畑・若林：“マイクロプログラムの融通性を利用した異種データチャ  
ネル制御方式”， 昭和53年度信学会総合全国大会 80.
- [24] 山田・宇敷・新谷・石川：“D20形自動交換機の改良中央処理系装置”， 通研実  
報， 第28巻 第3号， pp. 369-384, 1979.
- [25] T. Yamada, M. Miyake and Y. Ishikawa: "Improved Central Processor for  
D20 Electronic Switching System", Review of ECL, Vol.27, No.5-6,
- [26] 山田・小林：“中小局用電子交換機中央処理系装置へのビットスライスLSIの  
適用”， 信学会交換研究会， SE78-74, 1978.
- [27] 山田・小林：“中小局用プロセッサの磁気バブルメモリ制御方式”， 昭和53年  
度電子通信学会総合全国大会 講演番号1546.
- [28] 保坂・山田・隅田：“大局用電子交換機への磁気バブル装置の適用について”，  
昭和53年度電子通信学会総合全国大会 講演番号1542, 1979.
- [29] 秋山：“分散制御方式”， 信学会誌， Vol.61, No.4, pp.362-369, (4/'78)
- [30] 秋山・西山・笠原：“分散制御に関する検討”， 信学技報， SE 77-24, (1977)
- [31] 岡田・梶原・葉原：“交換機構成に関する一アプローチ”， 信学技報，  
SE 75-58, (1975-11)

- (32) 岡田・佐藤・三宅：“分散制御交換方式に関する一検討”，信学技報，  
SE 77-91 (1977-03)
- (33) 中島・鈴木：“モジュール形交換制御方式の提案”，信学技報，  
SE 77-76 (1977-10)
- (34) 岩崎・富永：“マルチマイクロプロセッサの電子交換機における一検討”，信  
学技報， SE76-89, (Oct. 1976) .
- (35) 清水・濃沼：“マイクロプロセッサの交換機への適用”，信学技報，  
SE76-10 (1976-05).
- (36) 西沢・柴田・林：“マイクロプロセッサの交換機への適用評価”，信学技報，  
SE76-11, (1976-05).
- (37) 山田・岡田：“マイクロプロセッサの電子交換制御系への適用に関する一検  
討”，昭和54年度信学会総合全国大会 1709.
- (38) 山田・岡田：“マイクロプロセッサ複合体による交換処理”，信学会交換研究  
会SE78-98, 1978.
- (39) K.Okada and M.kajiwara :”Functionary distributed telephone exchange  
control system using microprocessor”,Euromicro Newsl., 3,4,p.79  
(Oct. 1977)
- (40) S.G.Pitroda, W.A.Fechalos and c.J.Stehaman :”The micro controlled  
580 digital switching system”,ISS'76,213-4 (Oct. 1976)
- (41) D.Voegtlen:”CP 44-A modular SPC system for local transit and trunk  
exchanges with decetralized processors”,ISS '76 422-2 (Oct. 1977)
- (42) J.A.Bloc-Daude, J.Blanche and J.Tessier:”J2000 multiprocessor  
system. Software architecture advantages of J2000 distributed  
control system” ISS'76, 422-3 (Oct. 1977)
- (43) F.J.Schramel:”TCP 36-A control complex for telephone switching  
centers”,ISS '76, 422-4 (Oct. 1977)
- (44) Z.Koono, A.shoda and Y.Tokita:”A distributed control system for  
electronic switching systems”, ISS'76, 412-3 (Oct. 1976)
- (45) D.A.Lawson: “A New Software Architecture for Switching System” ,  
IEEE Trans. Commun., 30.6, pp. 1281-1286 (June 1982).
- (46) T.Anderson and O.Ljungfeldt : “Digital Transit Exchange AXE10” ,  
ERICSSON Review, No.2, PP.56-67 (1981).
- (47) T.Duncan and W.H.Huen : “Software Structure of No.5 ESS-  
A Distributed Telephone Switching System” , IEEE Trans. Commun.,

- 30,6, PP. 1379-1385 (1982).
- (48) 五嶋・松尾：“D60/D70デジタル交換方式”，信学誌，Vol.67，No.5，pp.501-523(昭和59-05)。
- (49) 都丸：“デジタル通信網への展望”，通研実報 デジタル電話市外系システムD T S - 1 特集，Vol.28 No.7，pp.1221，1979。
- (50) 葉原・荒谷：“デジタル電話網の意義と研究経緯”，通研実報，Vol.28 No.7，pp.1221，1979。
- (51) 五嶋・岩橋：“デジタル市内系システムの開発”，通研実報，Vol.31，No.11，pp.1937，1982。
- (52) 武田・保坂・吉岡・丹羽：“デジタル中継交換機の制御方式”，通研実報，Vol.28 No.7，PP. 1311-1320 (1979)。
- [53] T.Yamada and S.Miki: "Switching Control Multiprocessor for Information Network System", GLOBECOM'84.
- [54] 山田・三木：“交換処理用マルチプロセッサシステム”，信学論(B)，J68-B No.4，pp.476-483，(昭和60-04)。
- (55) K.Gotoh and M.Ikeda:"SYSTEM ARCHITECTURE OF DIGITAL SWITCHING SYSTEM D70(D) FOR INS: AN OVERVIEW", ISS'84 Proceeding 31A-1, (Florence, 7-11 May 1984).
- [56] 山田・花沢：“ONE-MACHINE 概念に基づいた交換機制御システムの階層構成”，信学技報 SSE-88-109，(昭和63-9)。
- [57] 山田・丹羽：“交換機設計用マルチプロセッサの設計と評価”，信学論(B)，Vol.J69-B No.12 pp.1625-1636，1986。
- (58) 渡辺・河辺・斉藤：“デジタル加入者線交換機のプログラム構成”，通研実報，Vol.31 No.11，pp.2003-2016，(昭和57-05)。
- (59) H.Watanabe, T.Sato and K.Saito:"Operating System for VLSI multiprocessor system", Fifth International Conference on Software Engineering for Telecommunication Switching System pp.167-172,(1983).
- [60] 山田・伊藤・齋藤・北野・須山：“デジタル中継交換用制御系の装置構成”，通研実報，Vol.28 No.7，PP. 1321-1338 (1979)。
- [61] 山田・丹羽：“交換制御用マルチプロセッサのハードウェア構成”，昭和53年度信学会通信部門全国大会 S5-7。
- [62] 山田・須山：“デジタル交換マルチプロセッサ用処理装置の一検討”，昭和54年度信学会全国大会 1712。
- [63] 山田・須山・三木他：“デジタル交換マルチプロセッサ用プロセッサ間

- 通信制御装置について”，昭和54年度信学会総合全国大会 1713.
- [64] 山田・須山・三木：“大局用マルチプロセッサ形交換機中央処理系ハードウェア構成”，信学会交換研究会 SE79-22.
- [65] 山田・須山・三木：“マルチプロセッサ形交換機のプロセッサ間通信手段と入出力手段”，信学会交換研究会 SE79-23.
- [66] A.Niwa and T.Yamada：“A 32-Bit Custom VLSI Processor for Communication Network Nodes”，IEEE Journal on Selected Areas in Communications, Vol.SAC-4 No.1, Jan., 1986.
- [67] 岸上：“DEX-1号機 中央制御装置”，通研実報, Vol.16 No.11, pp.2199-2253, (昭42-11).
- [68] 楠, 飯田, 伊藤：“DEX-2号交換機 中央制御装置の構成”，通研実報, Vol.18 No.10, pp.2513-2528 (昭和44-10)
- [69] T.Yamada, M.Kubota and T.Ohmiya：“CHILL Personal OS for MMI Integration around Switching System”，Sixth International Conference on Software Engineering for Telecommunication Switching Systems, pp.72-77, 1986.
- [70] A.Niwa and T.Yamada：“32bit VLSI Processor for Switching and Communications Processing”，GLOBECOM'82.
- [71] 丹羽・福田・隅田・伊藤：“交換用VLSI処理装置の実用化”，通研実報, Vol.33 No.7, PP. 1685-1693 (1984).
- [72] 山田：“VLSI化を想定した論理装置のバグ救済法に関する一検討”，昭和57年度信学会総合全国大会 講演番号1534.
- [73] 新谷・脇村・岡田・山田：“交換用VLSI処理装置の方式構成”，通研実報, Vol.33 No.7, pp.1695-1710(1984).
- [74] 松浦, 小林：“VLSI化装置用設計自動化システム”，通研実報, 第33巻 第7号, pp.1725-1736(1984).
- [75] 山田・小川：“デジタル交換機制御用マルチプロセッサのフェイルソフト設計”，信学技報（フォールトトレラントシステム研究会），（昭和63-9）.
- [76] 能条：“冗障規模を考慮した信頼度同等性の評価”，信学論(A), Vol.J64-A No.1 pp.9-14, 1981.
- [77] Y. Hori and S. Ogawa：“Fault Detection and Recovery for a Multi-processor Controlled Digital Switching System”，NTC'81 (New Orleans), A6.1.1-A6.1.5, (November 1981).
- [78] 山田・小川・須山・堀：“デジタル交換機用マルチプロセッサの故障処理”，

信学会論文誌(B), Vol.71-B No.3, pp.339-349 (1988).

- [79] 山田・須山：“交換用マルチプロセッサ制御系の障害処理用ハードウェア”，昭和53年度電子通信学会部門別全国大会 講演番号202.
- [80] S.Takamura, H.Kawashima and H.Nakajima：“Software Design for Electronic Switching Systems” IEE, London: Peter Peregrinus Ltd., pp41-45,1979.
- [81] 山田：“実時間処理装置の予備構成に関する考察”，昭和56年度信学会情報・システム部門別全国大会 講演番号382.
- [82] 山田：“部品高集積化による交換機予備構成方式の進化”，信学会情報通信網の安全性・信頼性時限研究専門委員会（山梨県石和），1988年2月5日
- [83] 山田：“デジタル交換機への超高集積部品の適用法と効果に関する考察”，昭和63年信学会春季全国大会，講演番号B-343
- [84] 村上・山田：“制御機能付きモジュラスイッチによる多段交換機”，昭和63年信学会春季全国大会，講演番号B-341
- [85] 山田・村上・栗原：“高速通信サービスの普及に向けた加入者ノードの小規模化と分散化”，昭和63年信学会秋期全国大会シンポジウム SB-4-6.
- [86] 山田：“デジタル伝達ノードアーキテクチャシミュレータ”ASCOT””昭和63年度情報処理学会秋期全国大会 2U-1.

## 筆者の関連発表一覧

【高速高集積部品を適用したマイクロプログラム多重制御方式の研究】

- [1] T.Hosaka, A.Niwa and T.Yamada: "High Speed Processor for Wide Area Control", International Communication Conference, ICC'77, Ses,20.3, June, 1977.
- [2] A.Niwa, T.Yamada and N.Ohnishi: "A High Speed Central Processor for D10 Electronic Switching System", Review of ECL, Vol.25, No.5-6, pp.379-389, 1977.
- [3] 山田・畑・山田: "D10高速中央処理装置データチャネル装置の構成", 通研実報, 26巻12号、pp.3261-3280、1977.
- [4] 山田・山田: "データチャネル高能率制御の一方式", 昭和51年度信学会全国大会 1447.
- [5] 山田・山田: "DCHマイクロ命令効率化の一検討", 昭和51年度電子通信学会通信部門全国大会 講演番号81.
- [6] 山田・畑・山田・埴淵・若林: "データチャネル制御方式の一評価", 昭和51年度信学会通信部門全国大会 82.
- [7] 山田・山田・矢沢・桜井・大川・兎耳山: "高能率データチャネル装置の一構成", 昭和51年度信学会通信部門全国大会 83.
- [8] 山田・畑・若林・大川・古川・梶塚・淡路: "D10用高速中央処理装置データチャネルの構成", 昭和52年度電子通信学会総合全国大会 講演番号1543.
- [9] 山田・山田・埴淵・岡・庄田・梶塚・淡路: "D10高速中央処理装置データチャネルのマイクロプログラム制御", 昭和52年度電子通信学会総合全国大会 講演番号1544.
- [10] 山田・若林: "多重制御型マイクロプログラムの融通性利用", 昭和51年度電子通信学会通信部門全国大会 80.
- [11] 山田・畑・若林: "マイクロプログラムの融通性を利用した異種データチャネル制御方式", 昭和53年度信学会通信部門全国大会 80.
- [12] 山田・小林: "中小局用電子交換機中央処理系装置へのビットスライスLSIの適用", 信学会交換研究会, SE78-74、1978.
- [13] 山田・小林: "中小局用プロセッサの磁気バブルメモリ制御方式", 昭和53年度電子通信学会総合全国大会 講演番号1546.
- [14] 保坂・山田・隅田: "大局用電子交換機への磁気バブル装置の適用について", 昭和53年度電子通信学会総合全国大会 講演番号1542.

[15] T.Yamada, M.Miyake and Y.Ishikawa: "Improved Central Processor for D20 Electronic Switching System", Review of ECL, Vol.27, No.5-6, 1979.

【VLSIプロセッサを活用した交換機制御用マルチプロセッサの研究】

- [16] 鈴木・山田・川崎: "電子交換機のプロセッサ間通信手段について", 昭和49年度信学会全国大会 1944.
- [17] 伊藤・山田・鍋沢: "交換用処理装置LSI化の一検討", 昭和55年度信学会総合全国大会一般講演 1662.
- [18] 山田: "VLSI化を想定した論理装置のバグ救済法に関する一検討", 昭和57年度信学会総合全国大会 講演番号1534.
- [19] 山田・須山: "交換用マルチプロセッサ制御系の障害処理用ハードウェア", 昭和53年度電子通信学会部門別全国大会 講演番号202.
- [20] 山田・伊藤・斉藤・北野・須山: "デジタル中継交換用制御系の装置構成", 通研実報, 第28巻 第7号.
- [21] 山田・岡田: "マイクロプロセッサの電子交換制御系への適用に関する一検討", 昭和54年度信学会総合全国大会 709.
- [22] 山田・岡田: "マイクロプロセッサ複合体による交換処理", 信学会交換研究会SE78-98, 1978.
- [23] 山田・丹羽: "交換制御用マルチプロセッサのハードウェア構成", 昭和53年度信学会通信部門全国大会 シンポジウム S5-7.
- [24] 山田・須山: "デジタル交換マルチプロセッサ用処理装置の一検討", 昭和54年度信学会全国大会 1712.
- [25] 山田・須山・三木: "デジタル交換マルチプロセッサ用入出力制御方式に関する一検討", 昭和54年度信学会総合全国大会 1710.
- [26] 山田・須山・三木: "デジタル交換マルチプロセッサ用プロセッサ間通信制御装置について", 昭和54年度信学会総合全国大会 1713.
- [27] 山田・三木・須山: "大局用マルチプロセッサ形交換機中央処理系ハードウェア構成", 信学会交換研究会 SE79-22.
- [28] 山田・須山・三木: "マルチプロセッサ形交換機のプロセッサ間通信手段と入出力手段", 信学会交換研究会 SE79-23.
- [29] T.Yamada, M.Kubota and T.Ohmiya: "CHILL Personal OS for MMI Integration around Switching System", Sixth International

- Conference on Software Engineering for Telecommunication Switching Systems, pp.72-77, 1986.
- [30] T.Yamada and S.Miki: "Switching Control Multiprocessor for Information Network System", GLOBECOM'84.
- [31] 山田・丹羽: "交換機設計用マルチプロセッサの設計と評価", 信学会論文誌(B), Vol.J69-B No.12 pp.1625-1636, 1986.
- [32] 山田・三木: 交換処理用マルチプロセッサシステム, 信学会論文誌(B), Vol.J68-B,4,1985.
- [33] 山田・小川・須山・堀:"デジタル交換機用マルチプロセッサの故障処理", 信学会論文誌(B), Vol.71-B No.3, pp.339-349 (1988).
- [34] 山田・花沢:"VLSI処理装置を用いたデジタル交換機の制御方式", 通研実報, 第33巻 第7号, pp.1711-1724, 1984.
- [35] A.Niwa and T.Yamada: "32bit VLSI Processor for Switching and Communications Processing ", GLOBECOM'82.
- [36] A.Niwa and T.Yamada: "32bit Custom VLSI Processor for Communications Network Nodes", The IEEE Journal on Selected Areas in Communications, Vol.SAC-4,No.1, Jan., 1986.
- [37] 山田・花沢:" ONE-MACHINE 概念に基づいた交換機制御システムの階層構成", 信学技報 SE-88-109, (昭和63-9) .
- [38] 山田・小川:"デジタル交換機制御用マルチプロセッサのフェイルソフト設計", 信学技報 (フォールトトレラントシステム研究会), FTS88-15 (昭和63-9) .
- 【超高集積部品を活用した制御システム構成の研究】
- [39] 小松原・山田: 遠隔制御方式における子局制御法の検討", 昭和59年度電子通信学会総合全国大会 1917.
- [40] 久保田・山田: " 実時間多重処理の効率化を指向したVLSIプロセッサアーキテクチャ", 信学会交換研究会, 信学技報 SE85-93., 1985.
- [41] 山田・佐藤・増田: "コンカレントCHILLを実行するマルチプロセッサの一検討", 昭和58年度信学会総合全国大会 1813.
- [42] 増田・山田: "蓄積交換用マルチプロセッサ構成の一検討", 昭和58年度信学会部門別全国大会 550.
- [43] 大宮・山田・久保田: "交換・通信処理用OSと処理ハードウェア構成の一検討", 信学会交換研究会, 信学技報 EC-84,1984

- [44] 山田：“デジタル交換機への超高集積部品の適用法と効果に関する考察”，  
昭和63年信学会春季全国大会，講演番号B-343
- [45] 村上・山田：“制御機能付きモジュラスイッチによる多段交換機”，  
昭和63年信学会春季全国大会，講演番号B-341
- [46] 山田・村上・栗原：“高速通信サービスの普及に向けた加入者ノードの小規模化と分散化”，昭和63年信学会秋期全国大会 シンポジウム SB-4-6.
- [47] 山田：“デジタル伝達ノードアーキテクチャシミュレータ”ASCOT”  
昭和63年度情報処理学会秋期全国大会 2U-1.

【部品高集積化を利用した制御システム高信頼化の研究】

- [48] 山田・新谷・宇敷・石川・斉藤：“交換局用プロセッサの冗長構成”，  
昭和53年度全国大会一般講演 1519.
- [49] 山田・宇敷・新谷・石川：“D20形自動交換機の改良中央処理系装置”，通研  
実報，第28巻 第3号，pp.369-384、1979.
- [50] 山田・新谷・小林・宇敷・若林：“中小局用電子交換機中央処理系装置のハ  
ードウェア構成”，信学会交換研究会，SE78-73，1978.
- [51] 山田：“実時間処理装置の予備構成に関する考察”，昭和56年度信学会情報・  
システム部門別全国大会 講演番号382.
- [52] 山田：“部品高集積化による交換機予備構成方式の進化”，信学会情報通信  
網の安全性・信頼性時限研究専門委員会（山梨県石和），1988年2月5日

【その他】

- [53] 山田・浦崎：“オンライントレース法による呼処理特性の調査”，昭和49年度  
信学会総合全国大会 1942.
- [54] 山田：“ハードウェアによる集中トラヒック過負荷制御方式”，昭和55年度  
信学会部門別全国大会 143.
- [55] 山田・脇村他：“高集積化処理装置における周辺接続インタフェースの  
考察”，昭和56年度信学会情報・システム部門別全国大会 講演番号380.
- [56] 山田・増田：“障害時のリンク切り替えを考慮したハイレベル通信制御法の  
構成法”，昭和56年度信学会部門別全国大会 284.
- [57] 鈴木・三木・山田：“ファームウェア化IICSの論理インタフェースに関する  
一検討”，昭和57年度信学会総合全国大会 1962.
- [58] 増田・倉矢・山田：“VLSI化論理装置の制御メモリインタフェースに関

する一検討”，昭和57年度信学会総合全国大会 講演番号1535.

- [59] 三木・山田：ニブルメモリにおけるプロセッサ高速化手法の一考察”，昭和58年度信学会部門別全国大会 543.
- [60] 小松原・山田・三木：“入出力制御用ハードウェアインタフェース仮想化の一手法”，昭和58年度信学会部門別全国大会 349.
- [61] 山田：“交換・通信処理用ソフトウェアデバッグ支援の一手法”，昭和59年度信学会部門別全国大会 349.
- [62] 山田・宮山：“蓄積プログラム制御用処理装置のメモリアクセス高速化手法”，信学会，交換研究会，信学技報EC84-9,1984

以上

## 筆者の関連特許一覧

### 【特許】

- [1] 山田：情報処理装置用電鍵設定装置，公告昭52-40943（昭和47年6月6日出願）
- [2] 桜井・山田：障害検出方式，公告昭54-38024（昭和51年4月20日出願）
- [3] 梶塚・山田・矢沢・庄田・安井：マイクロプログラムアドレス制御方式  
公告昭55-39217（昭和51年5月6日出願）
- [4] 山田・山田・大川・庄田・淡路：データチャネル装置  
公告昭56-9726（昭和51年2月24日出願）
- [5] 細野・遠藤・山田・字敷：割り込み制御方式  
公告昭56-26868（昭和52年8月24日出願）
- [6] 桜井・山田・高瀬・庄田・淡路：データチャネル装置  
公告昭56-40854（昭和51年11月4日出願）
- [7] 山田・山田・桜井・古川・梶塚・淡路：マイクロプログラム制御方式  
公告昭56-46607（昭和51年2月17日出願）
- [8] 畑・山田：優先順位切替方式 公告昭57-5029（昭和53年7月22日出願）
- [9] 岡田・山田：光信号方式 公告昭57-33746（昭和53年10月17日出願）
- [10] 岡田・山田：電子交換機 公告昭57-40713（昭和53年10月17日出願）
- [11] 山田・五嶋：マルチプロセッサシステム  
公告昭58-47746（昭和56年4月30日出願）
- [12] 山田・小林・若林：入出力制御装置 公告昭58-50367（昭和53年8月15日出願）
- [13] 山田・字敷・斉藤・石川・中村：情報処理装置  
公告昭59-258（昭和53年9月2日出願）