

資料 4

# 第5世代のコンピュータ

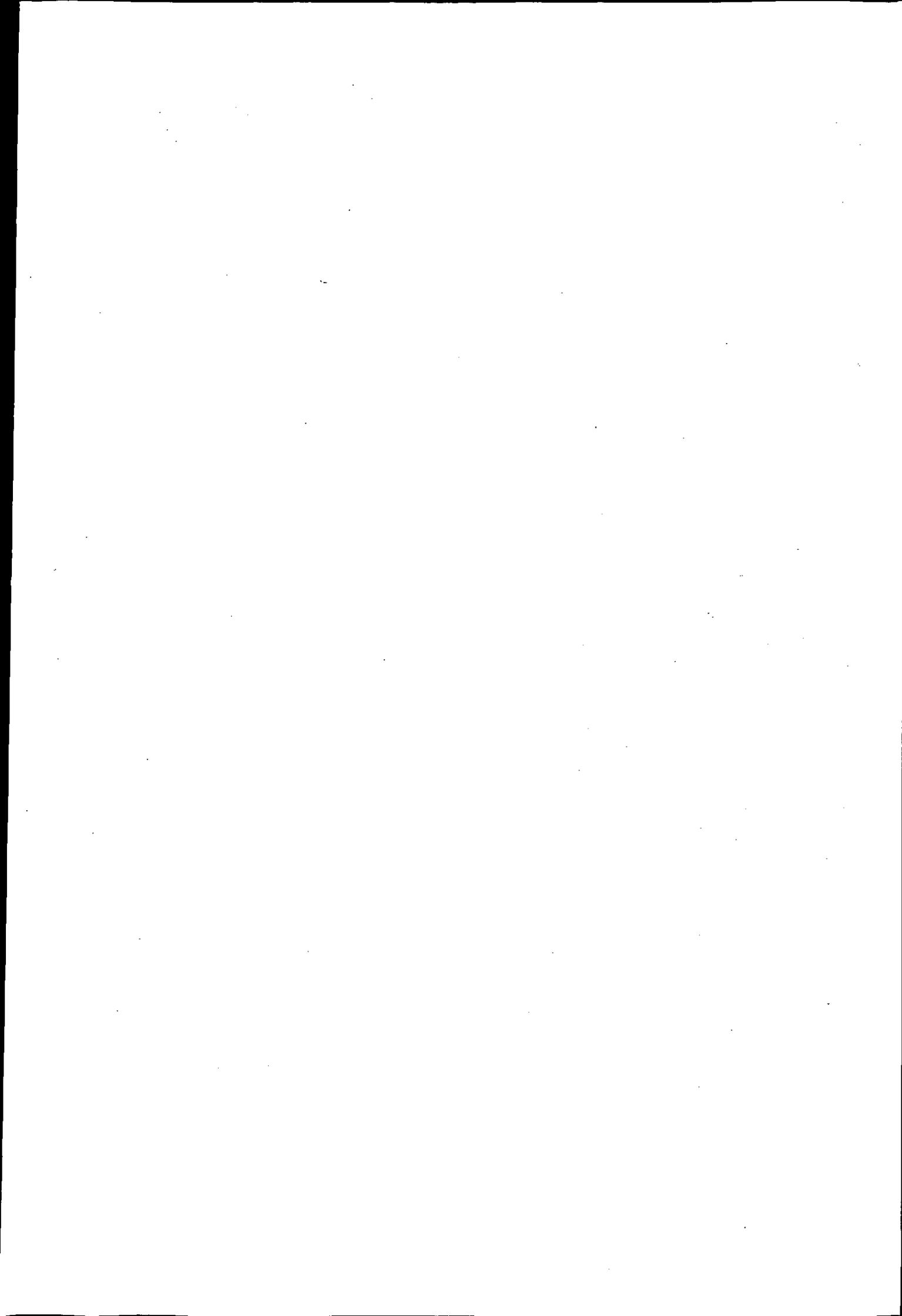
— コンピュータ関連技術動向調査 —

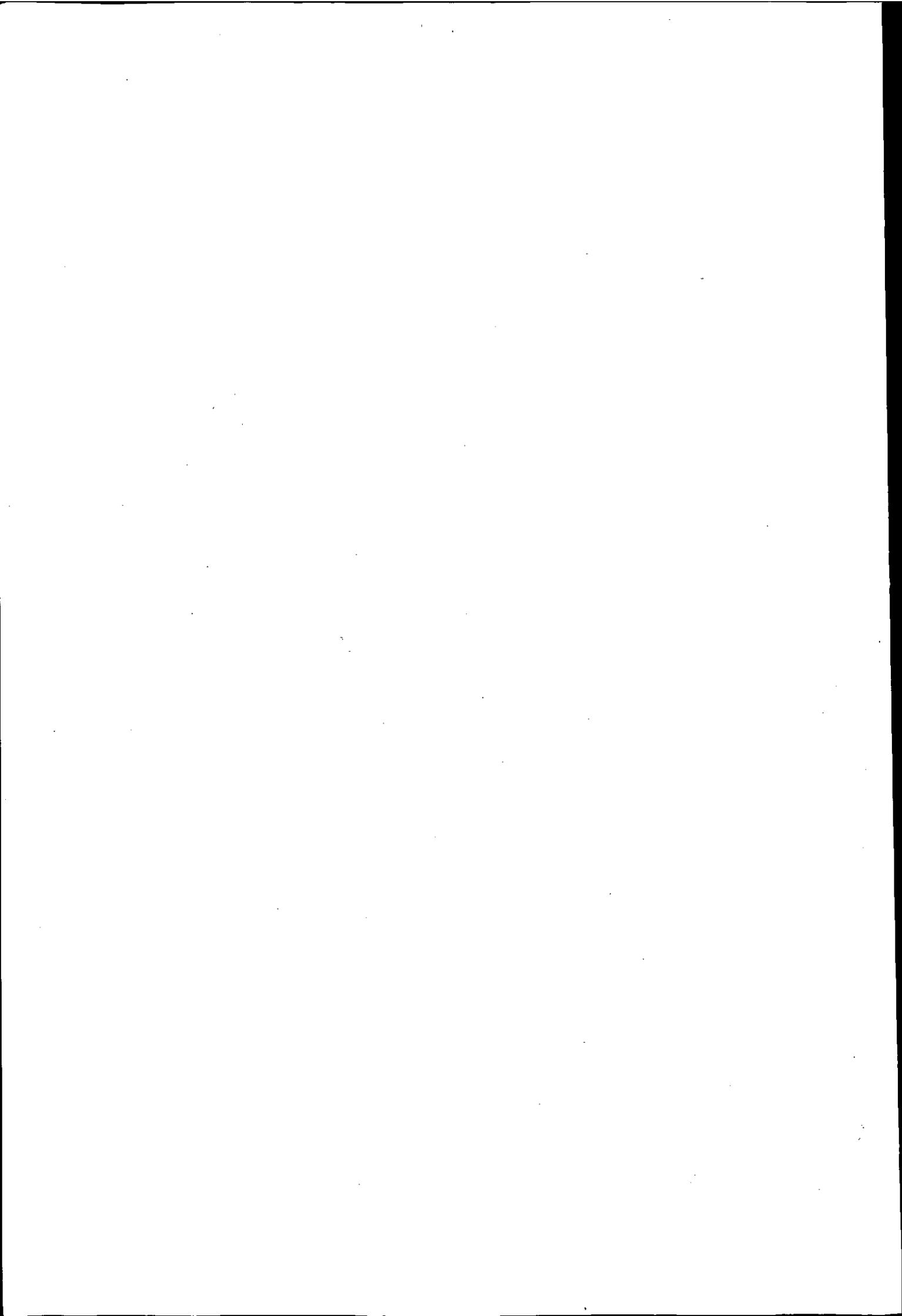
昭和 57 年 2 月

**JIPDEC**

財団法人 日本情報処理開発協会

この報告書は、日本自転車振興会から競輪収益の一部である機械工業振興資金の補助を受けて昭和56年度に実施した「第5世代電子計算機に関する内外技術動向調査」の成果をとりまとめたものであります。





## は　じ　め　に

わが国における社会経済は、資源、エネルギー問題を始めとして国際的な変動と、不確実性の流れのなかにある。同時に、的確な情報の加工利用が重要視される情報化社会の形成が指向されている。

コンピュータは、われわれの情報活用においてすでに不可欠なツールとなっているが、今後10年間には多くの諸問題を解決するため、更に高度な技術が要求され、新たな理論・技術にもとづくコンピュータ・システムの実現が望まれている。

このため、当協会では、1990年代に実用化されるべき新しいコンピュータ（第5世代コンピュータ）について、過去2ケ年間にわたり、総合的な立場から、調査研究を進めてきたが、本年度は、第5世代コンピュータ研究開発プロジェクトを推進するに当り、研究開発内容の具体化、本プロジェクトがひき起こす波及効果、プロジェクト管理のあり方、各種新技術の動向等について調査を行った。

更に、これまでの第5世代コンピュータに関する調査研究成果を内外に周知し、その評価及びテクノロジー・トランスファーの促進を図るとともに開発にあたっての国際協力体制作りの布石とするため、国際会議を開催した。

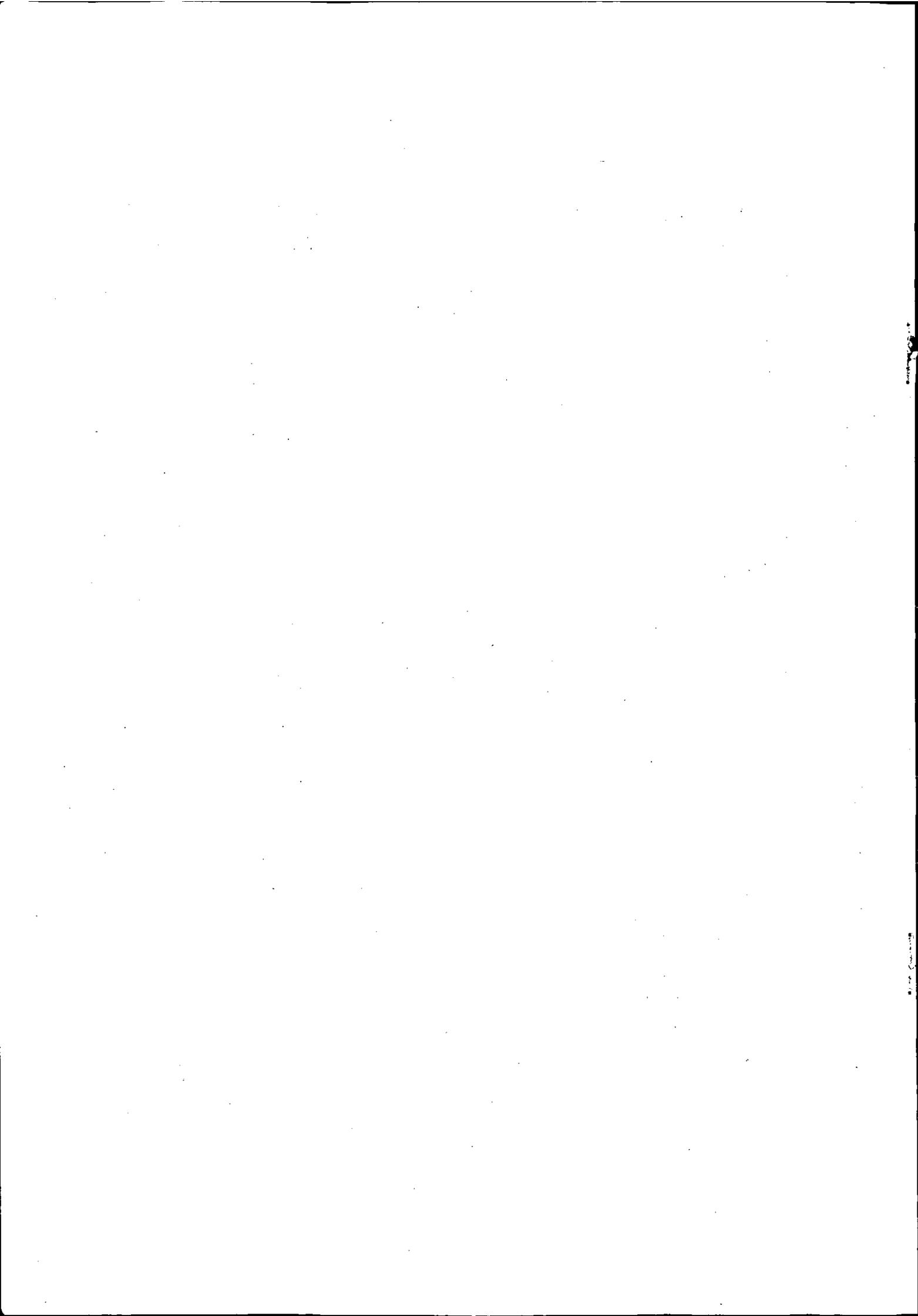
本報告書は、これらの調査研究成果のうち、第5世代コンピュータの関連技術動向調査について日本電子計算機株式会社に調査委託をした結果をとりまとめたものである。

最後に、本調査研究にあたって、ご指導ご協力いただいた委員会委員を始め、関係各位に対し、感謝の意を表します。

昭和57年2月

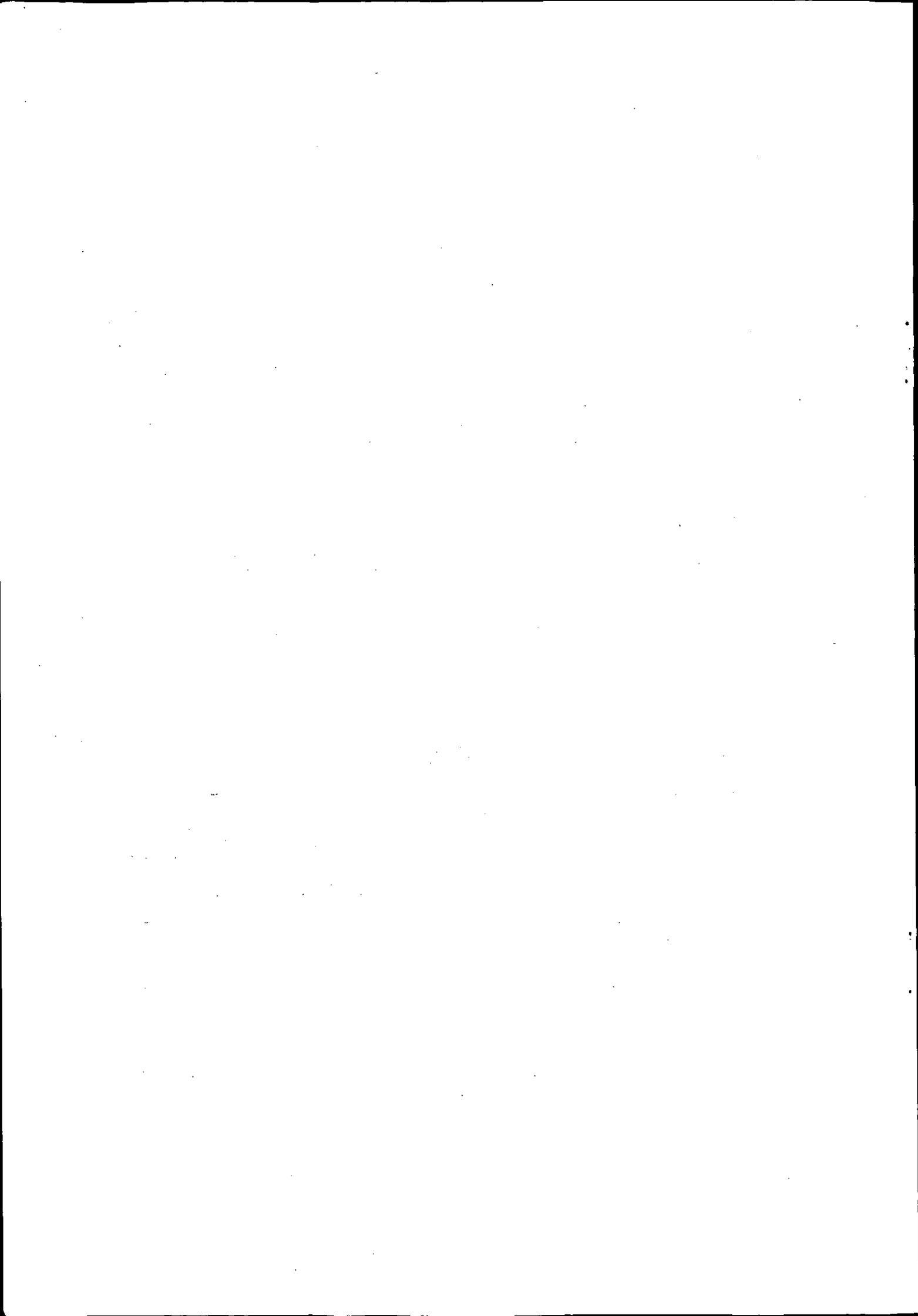
### 本調査で作成した報告書・資料

- ・第5世代のコンピュータ・研究開発計画
- ・第5世代のコンピュータ・データフローマシン/データベースマシン
- ・第5世代のコンピュータ・ロジックプログラミング
- ・第5世代のコンピュータ・波及効果
- ・第5世代のコンピュータ・関連技術動向調査
- ・第5世代のコンピュータ・研究開発計画・付属資料



第5世代コンピュータ関連技術動向調査ワーキング・グループ

委員名	所 属
(主査) 栗田昭平	㈱コンピュータワールド・ジャパン社長 (前・日本電子計算機㈱ 主幹)
植村 學	日電東芝情報システム㈱ コンピュータ統括本部 次長
大上 貴英	三菱電機㈱ 情報電子研究所情報処理開発部 計算機方式グループ
武井 欣二	東京芝浦電気㈱ 総合研究所情報システム研究所 ソフトウェアグループ 主任研究員
福西 俊策	富士通㈱ 電算機事業本部開発部開発技術部第一開発課
堀越 幸雄	日本電子計算機㈱ 技術部事務管理課 副主任
水野 昌美	㈱日立製作所 コンピュータ事業本部
安原 宏	沖電気工業㈱ 総合システム研究所知能システム研究部 情報理解研究室 研究主任
山崎 寛二	日本電気㈱ 情報処理企画室調査部
渡辺 淳三	日本電子計算機㈱ 第5世代、海外調査担当 副主任

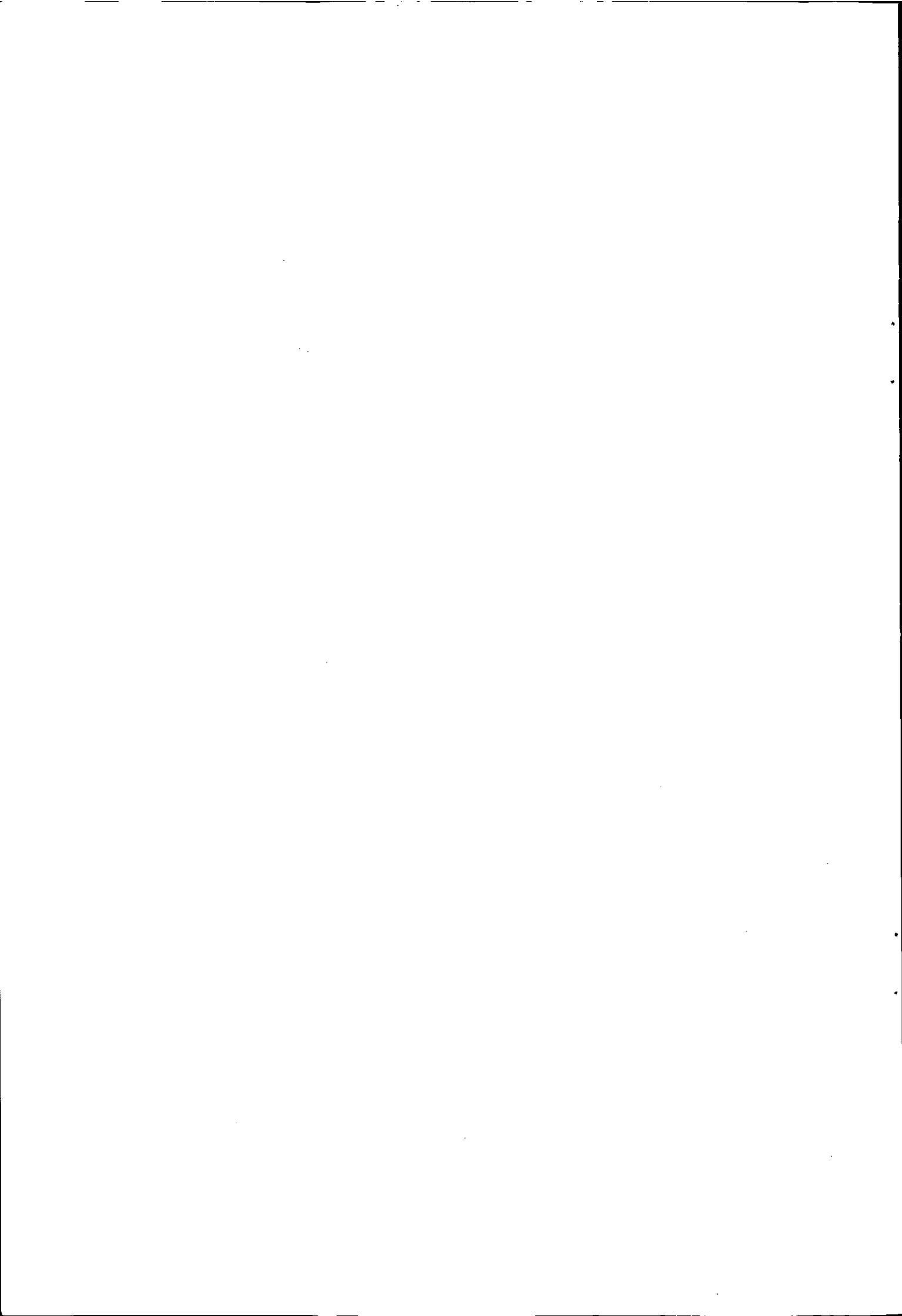


# 目 次

調査の方針	1
コンピュータ関連技術動向調査結果	3
1. 素子技術	5
2. ストージ部品	17
3. 電子光学部品	23
4. 大容量記憶装置	31
5. アーキテクチャ	35
6. 外資系汎用コンピュータの動向	64
6.1 外資系汎用コンピュータの動き	65
6.2 IBMの動向	67
6.2.1 発表機種	67
6.2.2 アーキテクチャの動き	73
6.2.3 将来予測	77
6.3 IBM 3081プロセッサ	80
7. パーソナル・マシン(高機能パーソナル・コンピュータ)	81
8. データベース技術および知識ベース技術	87
8.1 データベースシステム(DBS)	87
8.1.1 データベース管理システム(DBMS)	87
8.1.2 データベースマシン(DBM)	91
8.2 分散データベース	93
8.3 知識ベース	97
8.4 知識ベースマシン	103
9. ソフトウェア(プログラミング言語,特に非手続き型言語など)	104
9.1 プログラミング言語の主要動向	104
9.1.1 プログラミングの動向	104
9.1.2 プログラミング言語の動向	105
9.1.3 人工知能研究とプログラミング言語	106
9.2 非手続き型言語	107
9.2.1 “非手続き型(非手順的)”の概念	107
9.2.2 関連する概念	111

9.2.3	非手続き型言語の分類	113
9.2.4	非手続き性の尺度	114
9.2.5	非手続き型言語の利点と欠点	115
9.3	プログラミング言語の一覧	117
10.	ソフトウェア生産性向上の方策	123
10.1	ソフトウェアの生産性について	123
10.1.1	ソフトウェア生産性問題の背景(ソフトウェア危機)	123
10.1.2	ソフトウェア生産性の定義, 評価	123
10.1.3	ソフトウェアのライフサイクル	124
10.1.4	ソフトウェア生産性向上について	124
10.2	第5世代のプログラミング環境と生産性	125
10.2.1	知的プログラミング・システム	125
10.2.2	プログラム自動合成システム	125
10.2.3	知的プログラミング・システムと生産性	126
10.3	ソフトウェア生産技術年表	127
10.4	代表的な要求/設計仕様化技術	128
11.	コンピュータ・セキュリティ	129
11.1	基礎概念	129
11.2	セキュリティ手段の現状	130
11.3	知識情報処理におけるセキュリティ	132
11.4	技術相関図	133
11.4.1	第5世代コンピュータ・システムの枠組とセキュリティ機構	134

# 調査の方針



## 調 査 の 方 針

第5世代コンピュータ調査研究プロジェクトの56年度作業の一環として、第5世代コンピュータ関連技術動向調査ワーキング・グループは、主として海外動向を中心にコンピュータ関連技術の現状と新しい流れを体系的に整理し、関係者の参考に資することとした。

この資料は、ワーキング・グループが1982年2月現在入手できた最新データにもとづき、次の11の主要分野について技術の現状と現時点における予測を整理したものである。

調査にあたっては、まず次ページの情報関連技術体系図をもとに全体のとらえ方を把握した。内容的には、昨年度の調査結果を見直すとともに、ソフトウェア生産性向上の方策、パーソナル・マシンの動向、コンピュータ・セキュリティなどの新項目を追加してある。

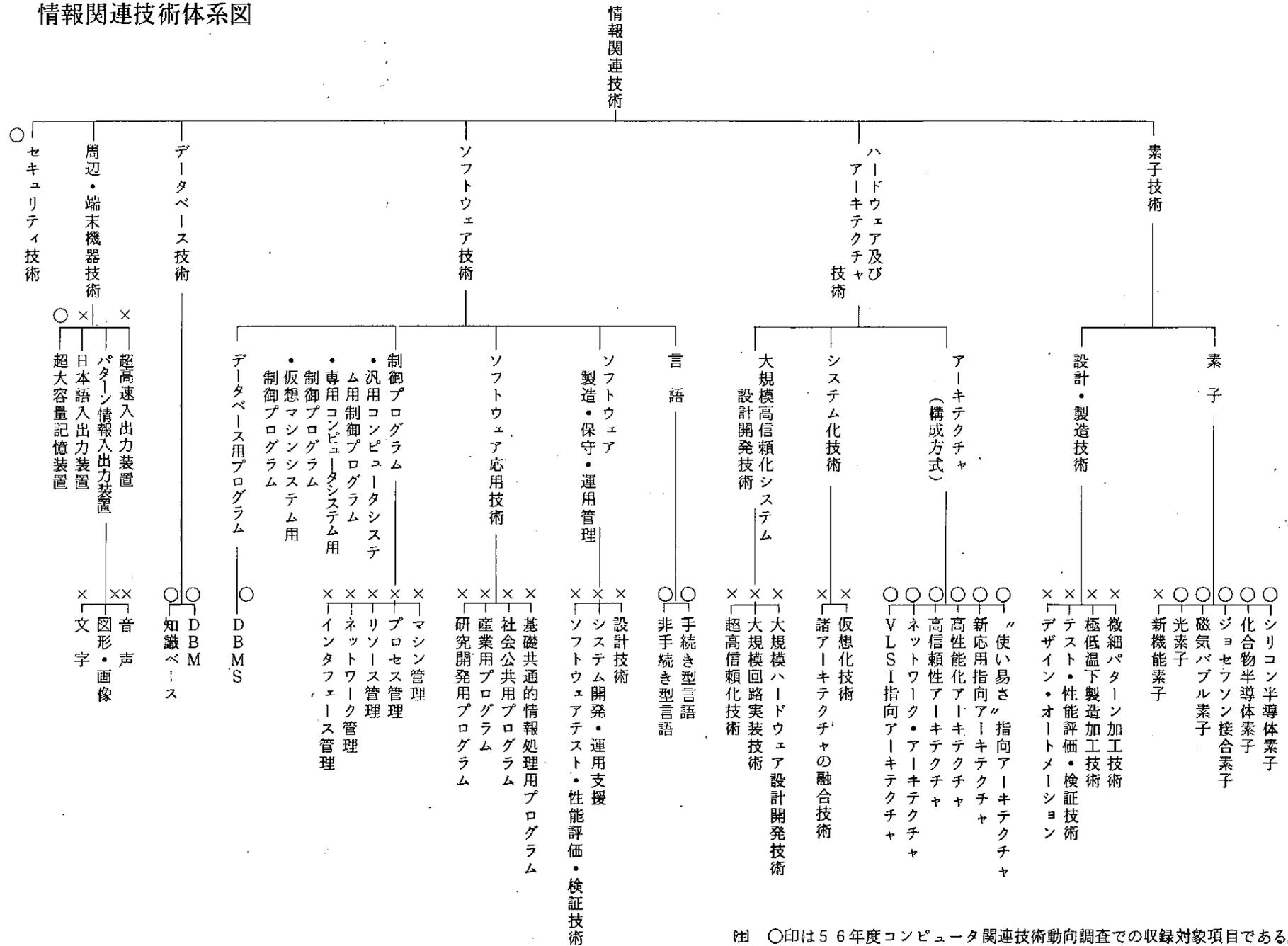
11の主要分野とは次のとおりである。

1. 素子技術
2. ストージ部品
3. 電子光学部品
4. 大容量記憶装置
5. アーキテクチャ
6. 外資系汎用コンピュータの動向
7. パーソナル・マシン（高機能パーソナル・コンピュータ）
8. データベース技術および知識ベース技術
9. ソフトウェア（プログラミング言語、特に非手続き型言語など）
10. ソフトウェア生産性向上の方策
11. コンピュータ・セキュリティ

今後この資料がひきつづき修正、更新、補足されていき、関係者のお役に立てば幸いである。

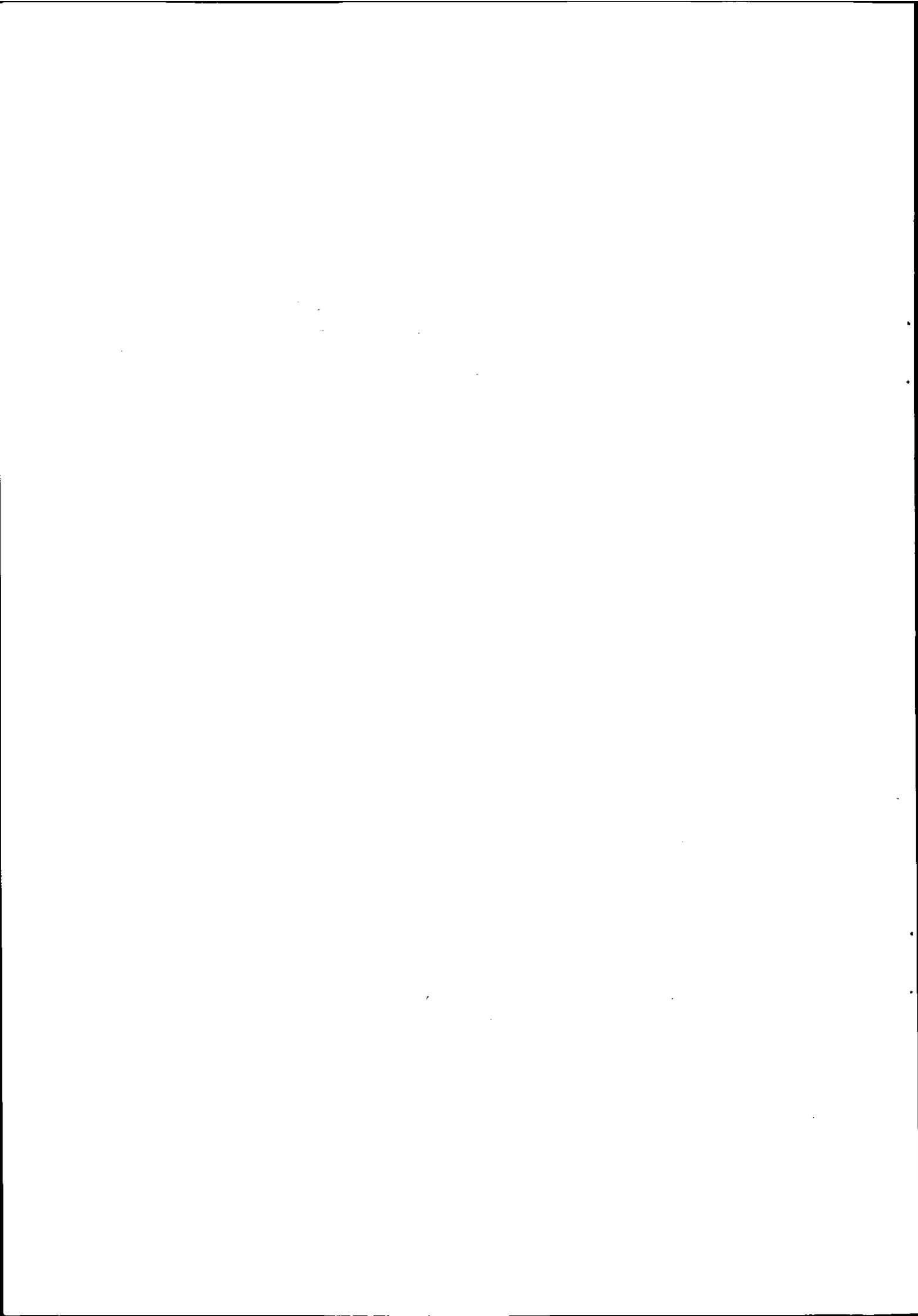
(註) 図表の出典に「(修正)」とあるものは、出典データに一部手を加えたことを示す。また、図表に出典のないものは当ワーキング・グループ調べによるものである。

# 情報関連技術体系図



○印は56年度コンピュータ関連技術動向調査での収録対象項目である。

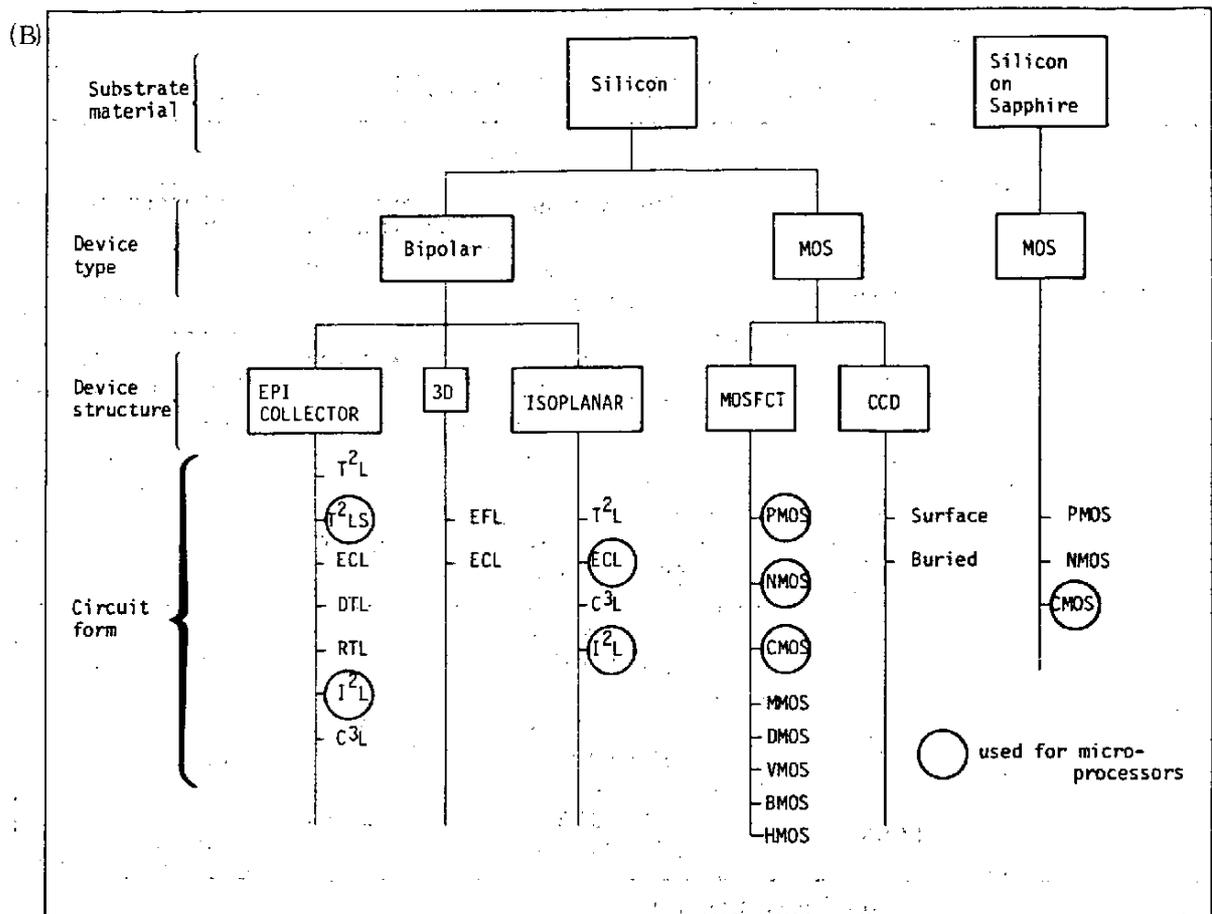
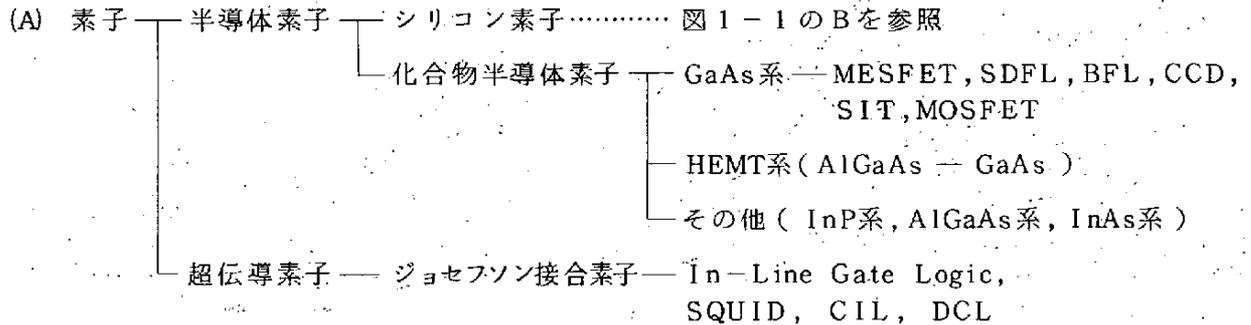
コンピュータ関連技術動向  
調査結果



# 1. 素子技術

図1-1 素子技術

1)~5)



出典: 1) Infotech: The Fifth Generation, State of the Art Report, Series 9, No.1, 1981.  
 2) 岡部力也: 一気に集積度が上がる論理LSI, 日経エレクトロニクス, 1981年4月13日号, pp.148-158.  
 3) 津田建二: 高速, 低消費電力の魅力で開発に拍車がかかり始めたGaAs論理IC, 日経エレクトロニクス, 1979年4月2日号, pp.64-77.  
 4) 西村吉雄他: 新しい高速素子の仲間入りを狙う高移動度デバイス, 日経エレクトロニクス, 1981年3月16日号, pp.108-128.  
 5) S.M.Faris: Josephson LSI Circuits, IEEE Circuits and Systems Magazine, Vol.3, No.3, Sept. 1981, pp.2-8.

図1-2. 素子の性能(A)

図1-3. 素子の性能(B)

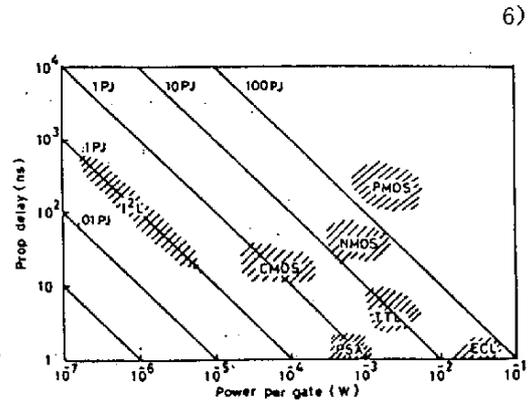
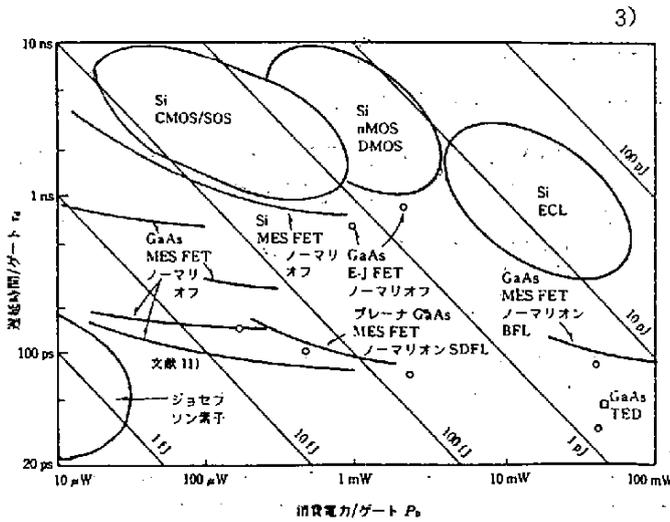


表1-1. TTLとECLの性能比較

(7)~(8)

Device		Gate		ALU
		Propagation Delay	Power Dissipation	Propagation Delay
TTL	Normal	10 ns (SN7402)	14 mW	35 ns (SN74181)
	LS *1	10 ns (SN74LS02)	275 mW	27 ns (SN74LS181)
	S *2	35 ns (SN74S02)	29 mW	155 ns (SN74S181)
	ALS *3	55 ns (SN74ALS02)	19 mW	-
	AS *4	30 ns (SN74AS805)	221 mW	90 ns (SN74AS881)
ECL	10K	20 ns (F10102)	25 mW	80 ns (F10181)
	100K	0.75 ns (F100102)	40 mW	4.5 ns (F100181)

- \*1 LS : Low-Power Schottky
- \*2 S : Schottky
- \*3 ALS : Advanced Low-Power Schottky
- \*4 AS : Advanced Schottky

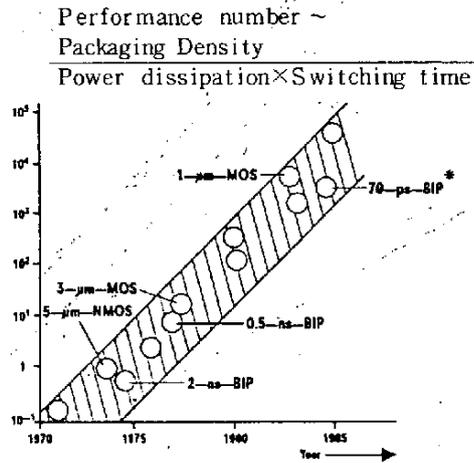
出典: 6) K.Okada et al.: A New Polysilicon Process for a Bipolar Device-PSA Technology, IEEE Trans. Electron Devices, Vol. ED-26, No.4, April 1979, pp.385-389.

7) Texas Instruments: The Bipolar Digital Integrated Circuits Data Book for Design Engineers, Parts 1 and 2, 1981.

8) Fairchild: ECL Data Book, 1977.

図 1-4. 集積回路の性能動向

出典: K.F.Goser: The Challenge of the VLSI  
Technique to Telecommunications Systems,  
IEEE Trans. Electron Devices, Vol.  
ED-27, No.8, Aug. 1980, pp.1341-1345.



\* ) BIP = Bipolar

図 1-5. バイポーラ RAM の性能動向

出典: J.Stinehelfer et al.: Large ECL  
Bipolar RAMs, Digest of Papers,  
COMPCON '81, Spring, 1981, pp.120-124.

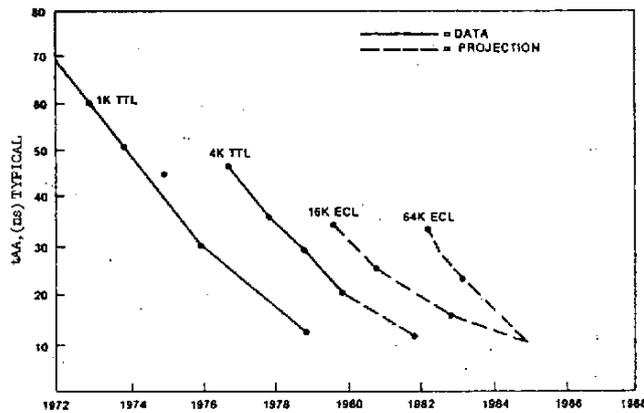


図1-6. ゲート・アレイにおけるゲート・ディレイの動向

出典: Gnostic : 2. Gate Arrays, VLSI Information Service, 1980/1981.

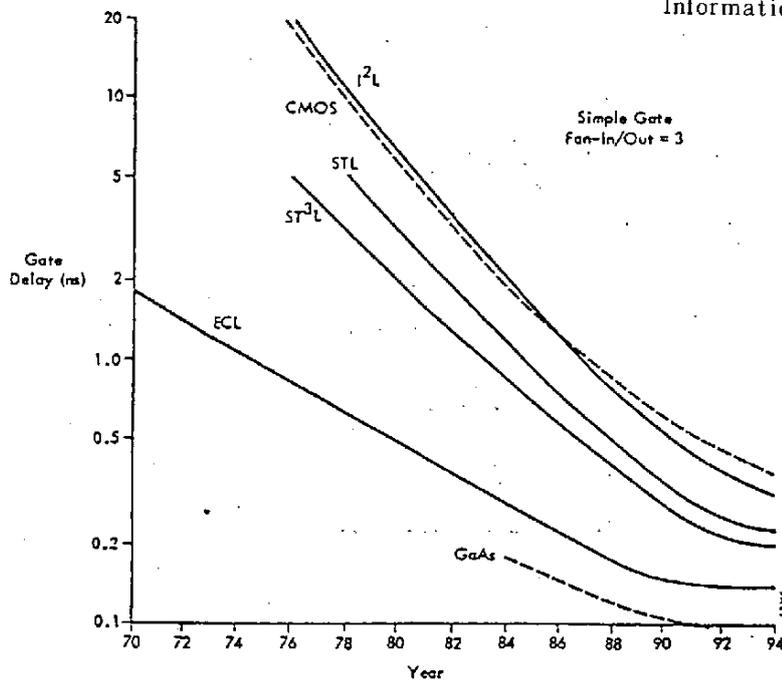


図1-7. 集積回路の集積度の動向

出典: Infotech : The Fifth Generation, State of the Art Report, Series 9, No.1, 1981.

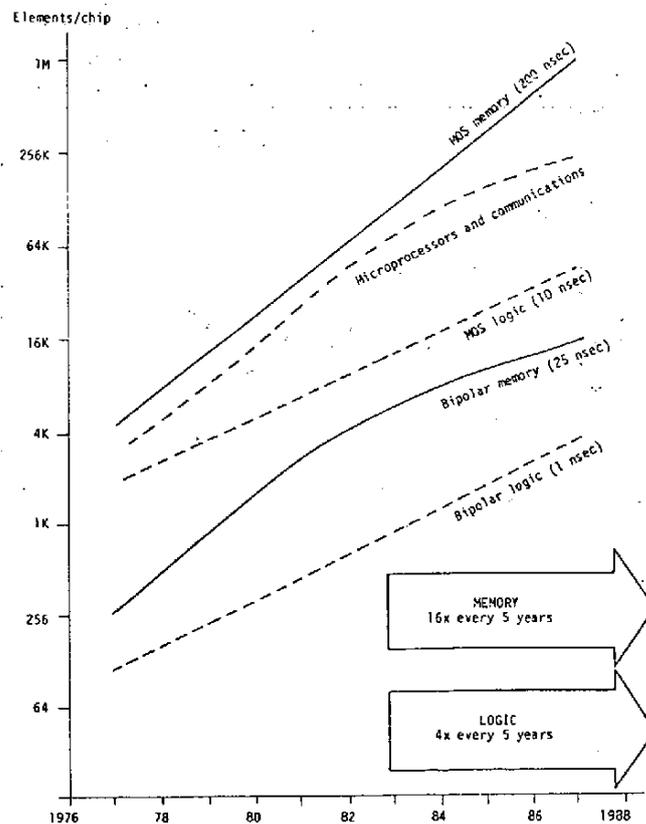


図 1-8 IBM社の集積回路技術の動向

出典：E.J.Rymaszevski：Semiconductor Logic Technology in IBM, IBM Journal of Research and Development, Vol.25, No.5, Sept.1981, pp.603-616.

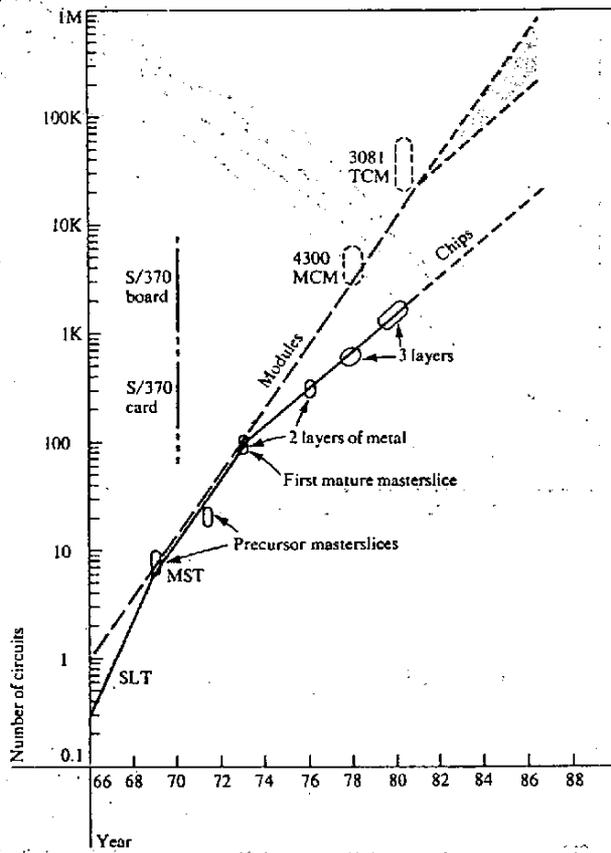


図 1-9 GaAs ICの集積度の動向

出典：片岡照栄：超高速機能デバイスの動向，データショウ'81 国際シンポジウム「スーパーコンピュータ」, pp.1-12.

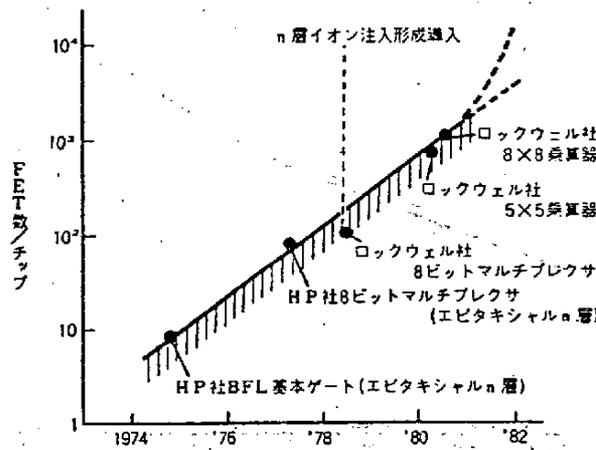


図 1 - 10. ゲート・アレーにおける集積度の動向

出典：Gnostic：2. Gate Arrays, VLSI Information Service, 1980/1981.

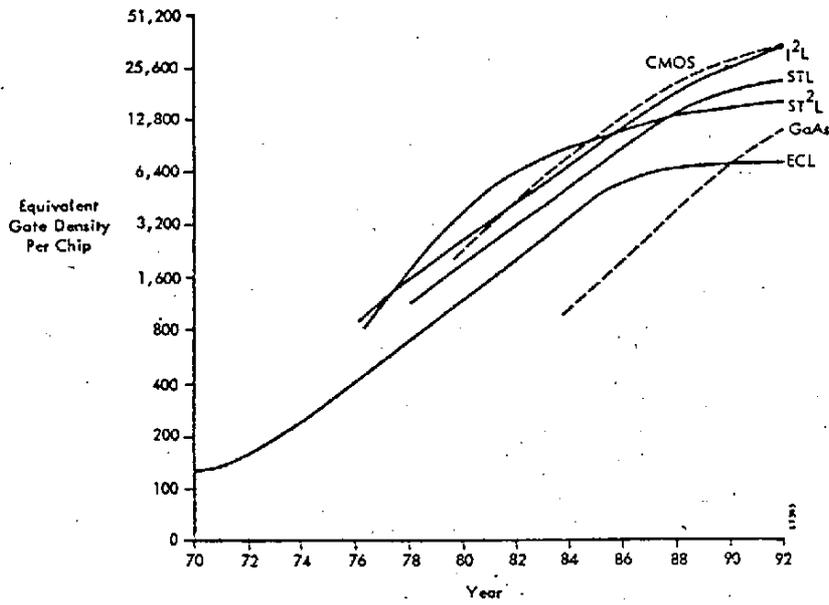


図 1 - 11. 最大チップ面積の動向

出典：Gnostic：7. Memory Logic Completion, VLSI Information Service, 1981.

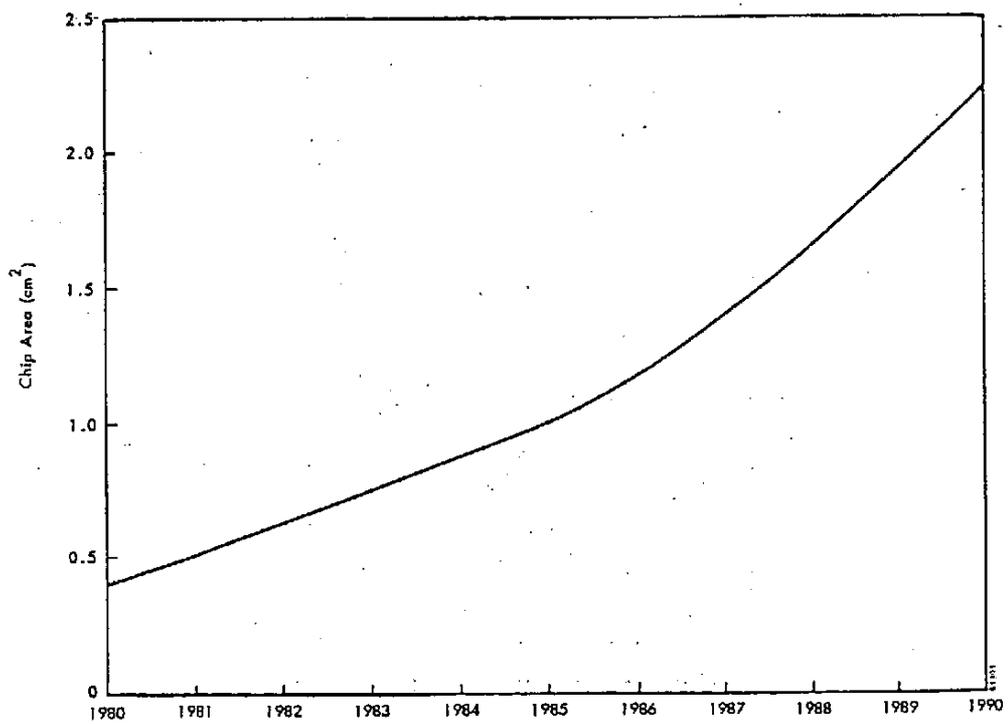


図 1-12. セル面積の動向

出典: K.Ohta et al.: Quadruply Self-Aligned MOS (QSA MOS)—A New Short-Channel High-Speed High-Density MOSFET for VLSI, IEEE Journal of Solid-State Circuits, Vol. SC-15, No.4, Aug. 1980, pp.417-423.

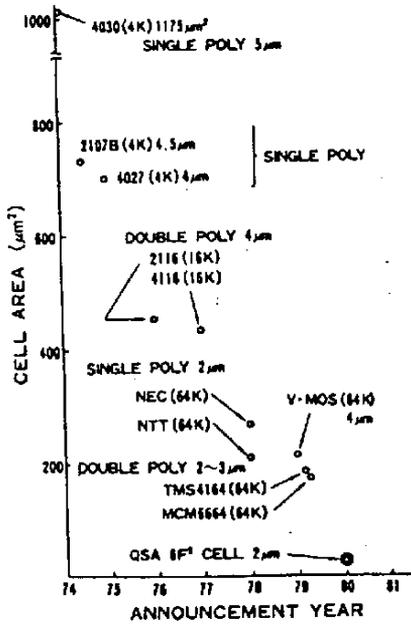


図 1-13. バイポーラ RAM のチップ面積の動向

出典: J. Stinehelfer et al.: Large ECL Bipolar RAMs, Digest of Papers, COMPCON '81, Spring, 1981, pp.120-124.

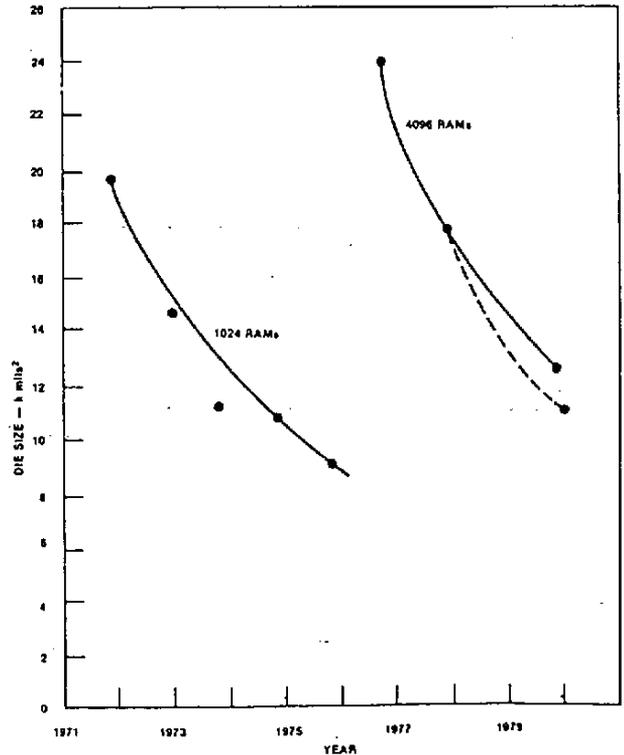


図 1-14. MOS ゲートと金属配線のサイズ動向

出典: Gnostic: 4. VLSI Gates/Interconnections, VLSI Information Service, 1981.

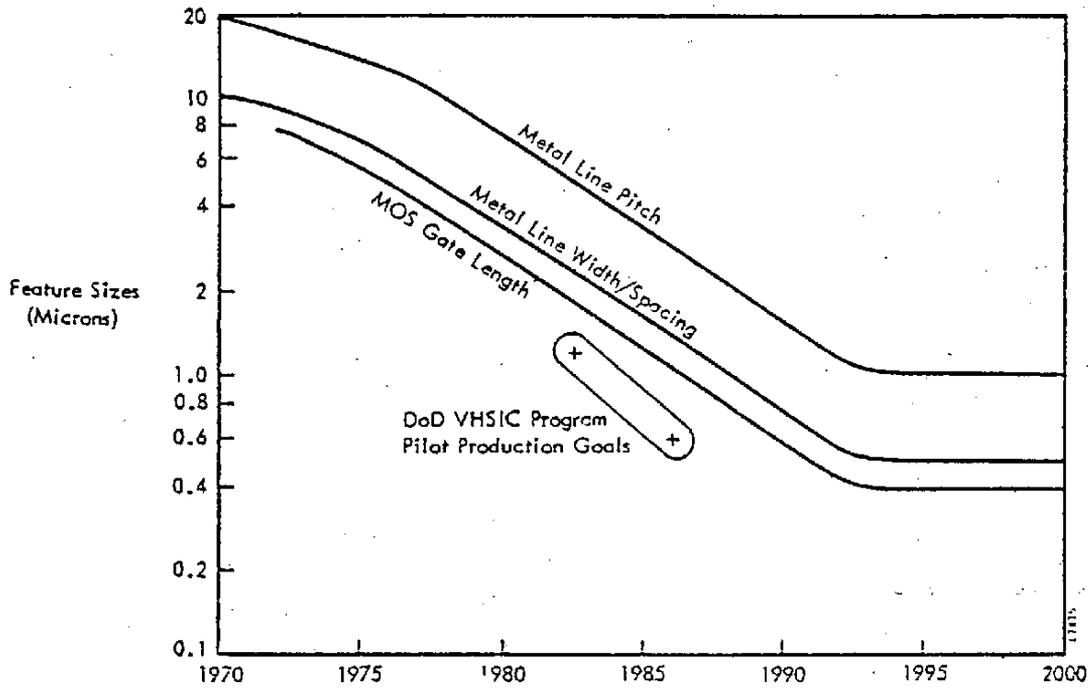


図 1-15. MOS ゲート・テクノロジーの動向

出典: Gnostic: 4. VLSI Gates/Interconnections, VLSI Information Service, 1981.

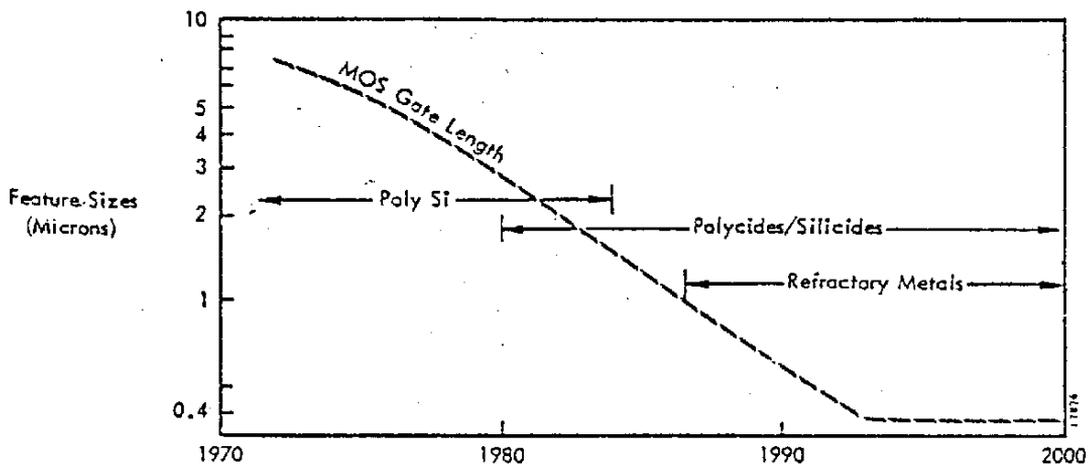


図1-16. Josephsonデバイスの性能

出典: W. Anacker: Computing at 4 Degrees Kelvin, IEEE Spectrum, Vol.16, No.5, May 1979, pp.26-37.

(A) ジョセフソン・テクノロジーによる論理チップとメモリー・チップ

	Logic	Fast RAM	Dense RAM
Capacity, circuits on bit per chip	400	2 k	16 k
Logic or delay access time, ns	.06	1	15
Power, mW/chip	3	5	.04

(B) ジョセフソン・デバイスを用いた仮想コンピュータの諸元

Performance	70 million instructions per second
CPU cycle time	4 s
Cache capacity	32k
Main RAM capacity	16 Mbyte
Cache access time	4 ns
Main RAM access time	20 ns
I/O data rate (max)	360 Mbit/sec
Power at 4°K	7W
Power at 300°K	15 kW
Volume of Mainframe	4 liter
Volume of Cryostat	460 liter

図1-17. GaAs ICの需要予測

出典: 片岡照栄: 超高速機能デバイスの動向, データショー'81国際シンポジウム「スーパーコンピュータ」, pp.1-12.

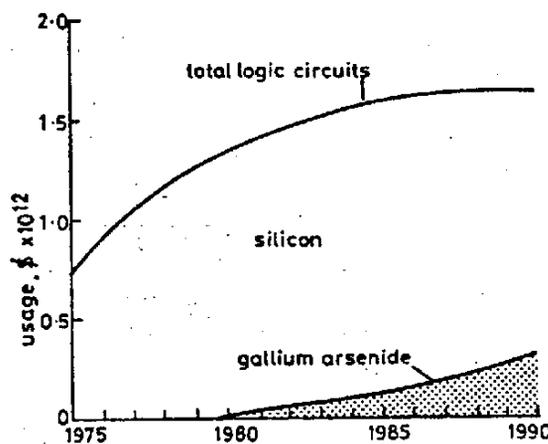
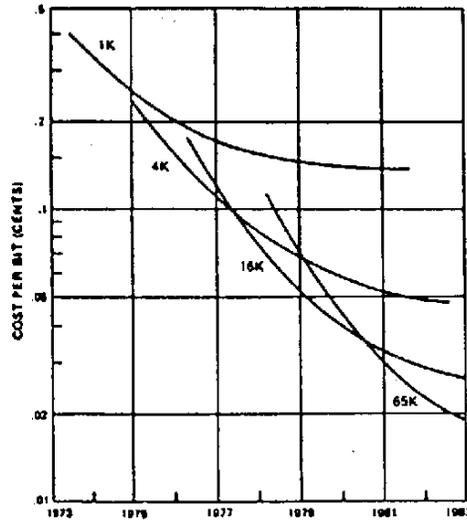


図 1-18. メモリー素子の価格動向

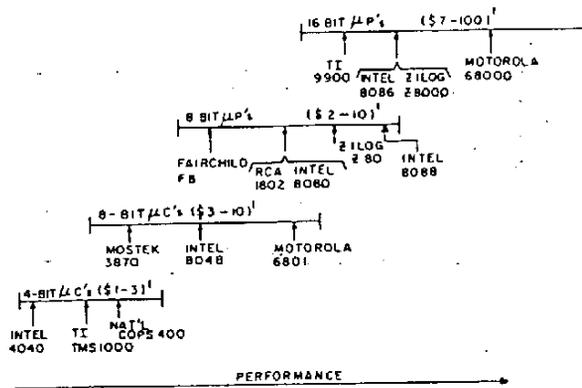
出典: R.F.Spencer, Jr.: Interaction of VLSI Technology Progress with Minicomputer Product Development, IEEE Trans. Electron Devices, Vol. ED-26, No.4, April 1979, pp.284-291.



Note: Cost per bit of computer memory has declined and should continue to decline, as is shown here for successive generations of random-access memory circuits capable of handling from 1024 (1K) to 65536 (65K) bits of memory. Increasing complexity of successive circuits is primarily responsible for cost reduction, but less complex circuits also continue to decline in cost.

図 1-19. マイクロプロセッサの分類

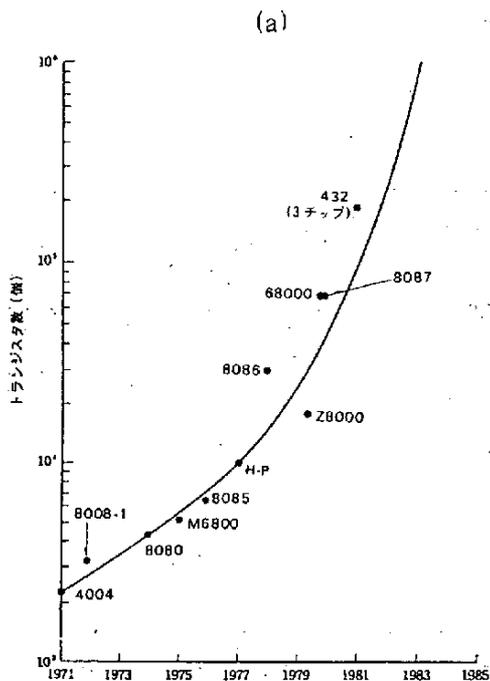
出典: P.M.Russo: VLSI Impact on Microprocessor Evolution, Usage, and System Design, IEEE Trans. Electron Devices, Vol. ED-27, No.8, Aug. 1980, pp.1332-1341.



<sup>1</sup> ALL COSTS ASSUME MATURE PRODUCT AND 1980 DOLLARS

図1-20. マイクロプロセッサの集積度動向

出典：岡部力也：一気に集積度が上がる論理LSI，  
日経エレクトロニクス，1981年4月13日号，  
pp.148-158.



(b)

出典：P.M.Russo：VLSI Impact on Microprocessor Evolution, Usage, and System Design, IEEE Trans. Electron Devices, Vol. ED-27, No.8, Aug.1980, pp.1332-1341.

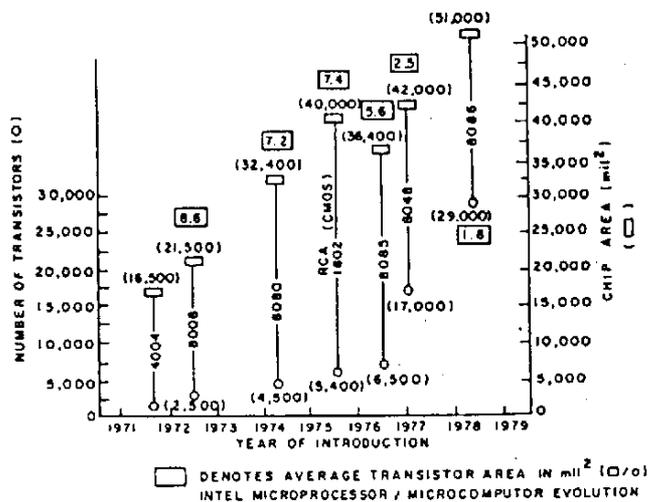


図1-21. MOSマイクロプロセッサのデザイン・ルール動向

出典: Gnostic: Microprocessor/Microcomputer,  
VLSI Information Service, 1981.

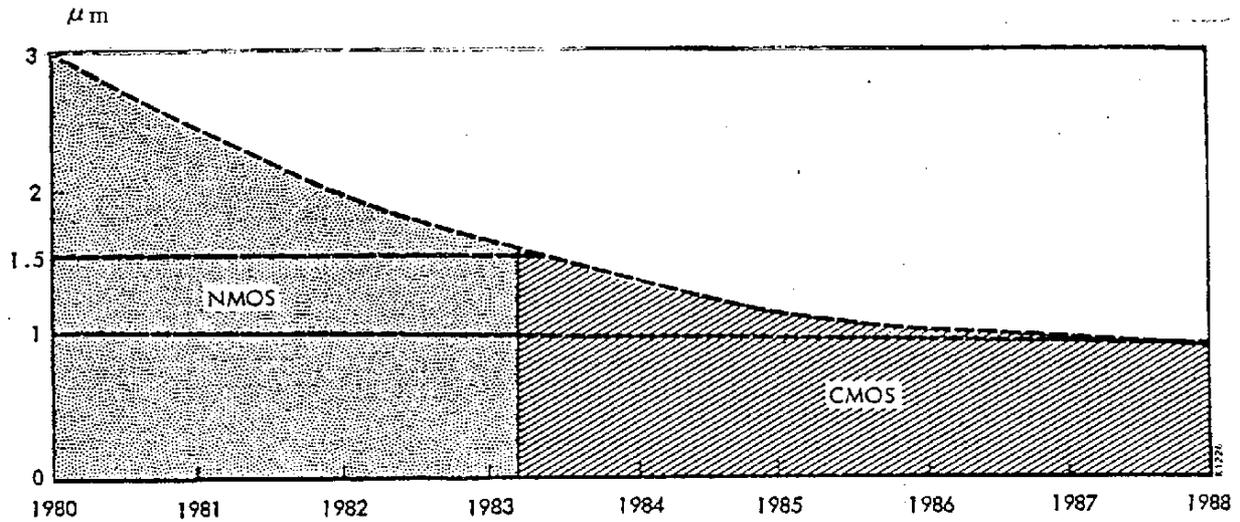
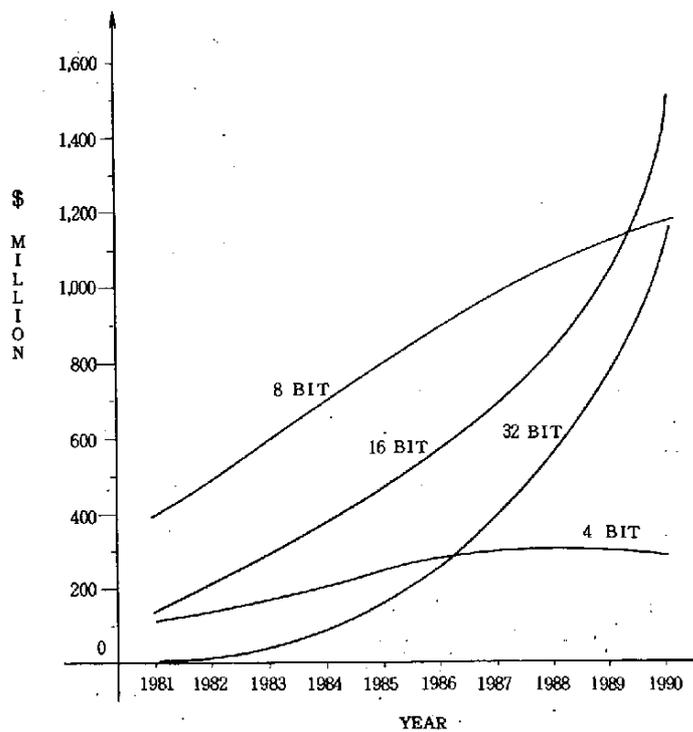


図1-22. 米国におけるマイクロプロセッサ需要予測

出典: Gnostic: Microprocessor/Microcomputer,  
VLSI Information Service, 1981.



## 2. ストージ部品

メモリーは、大別すると①半導体メモリー、②磁気記録メモリー、③光メモリーの3つに分けられる。

第1の半導体メモリーの多くは、アクセス・タイムが高速であるという利点を生かし、コンピュータの内部メモリーとして用いられており、今後もアクセス・タイムは表2-1のとおりアップすると予想される。性能向上については、近年、コンピュータ・メーカー等が莫大なる研究費を投じて研究を行っている超LSI等の開発に負うところが大きく、チップ当りの容量も表2-2のとおり各々5年間で4倍はアップするであろう。またコストについても上記および量産技術の発展等の理由から表2-1, 2-2, 2-3が予想される。

なお、バブル・メモリーについては、表2-1のとおりアクセス・タイムは比較的遅いが、不揮発性であるという性格およびサイズ、コスト等による利点、その他から、半導体メモリーと磁気記録メモリーの中間に位置し、より小規模な機器(ターミナル、ポータブル・システム、デスク・トップ・システム)に使用されるのではないが。またジョセフソンについては、動作原理等は解明されており、非常に速い半導体製造技術の発展ペースを考えると、控え目にみてもコスト/性能は表2-1, 2-2, 2-3のことが達成されるであろう。

第2の磁気記録メモリーの代表である磁気ディスク装置は、この予測時期を通して、性能およびコストとも向上のテンポが鈍化するかも知れないが、将来、アクセス機構の画期的改善、および超高密度記録方式の開発により、大幅に性能、コストを改善する機会があるかも知れない。また、改善されなくとも、前で述べた半導体メモリーよりも、ビット当りのコストは表2-3のとおりはるかに低くおさえられ続けるであろう。

第3のホログラフィック・メモリーについては、最近市場へ出されつつある書き替え不可能な光ディスク・ファイル装置について述べる。

光ディスク開発のねらいを考えると大容量でかつ低価格が要求されるアプリケーション市場用(画像記録システム等)につくられてきたが、今後はこの市場で滲透しつづけるであろう。

そのため、今後も表2-1, 2-2の様な性能および表2-3の様に磁気記録メモリーより少なくとも1桁低い価格範囲を維持しなければならないし、維持するであろう。なお、書き替え可能光ディスクについても近い将来、出現することは間違いないと考えられる。

図 2-1 半導体ストレージ部品

出典: INFOTECH  
 STATE OF THE ART REPORT  
 SERIES 9 NUMBER 1  
 "The Fifth Generation"

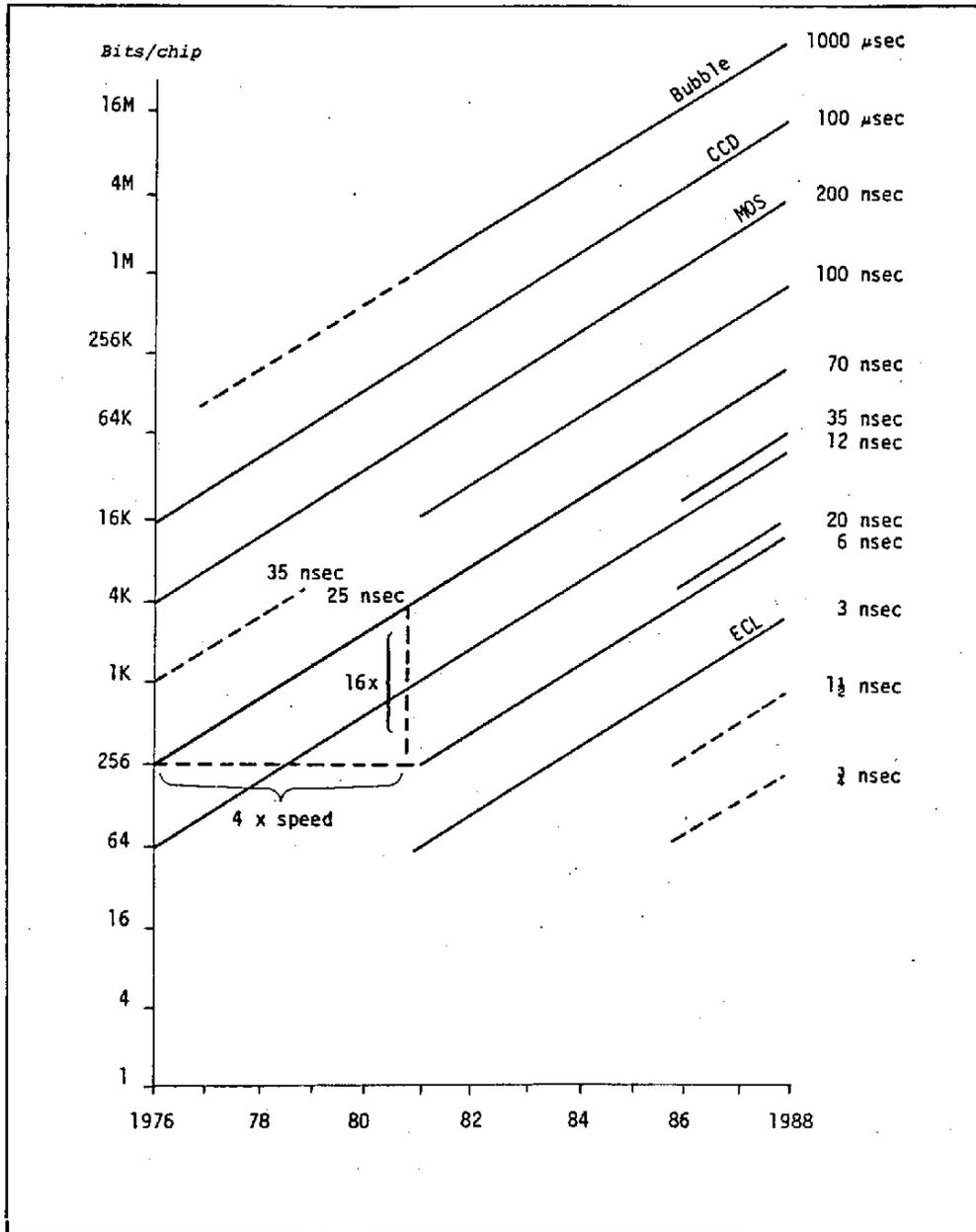
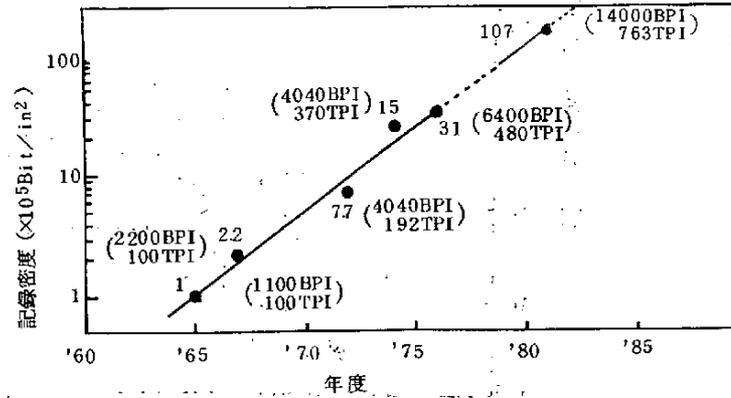


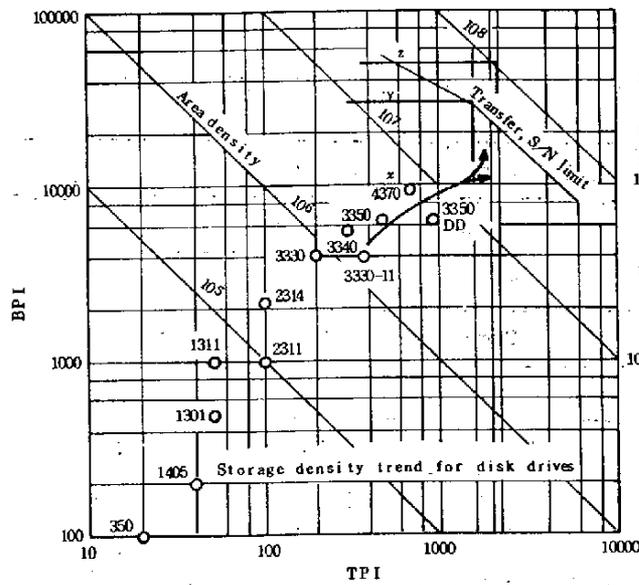
図 2 - 2 磁気記録メモリーの動向

(A)~(C)出典：第5世代の電子計算機に関する調査研究  
報告書—アーキテクチャ研究分科会—  
(修正)

(A) 塗付型ディスクにおける記録密度の進歩



(B) トラック密度 (TPI) と線密度 (BPI) の進化過程と実用限界



(C) 単位体積あたりの記憶容量、面記録密度と1ドライブあたりの記憶容量の関係

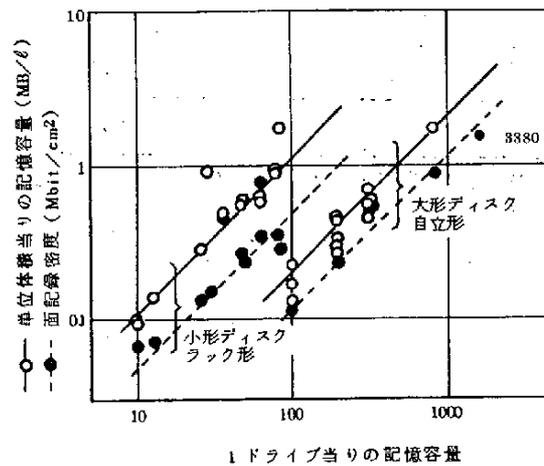


図 2-3 光メモリーの動向

出典: PHILIPS LABS.

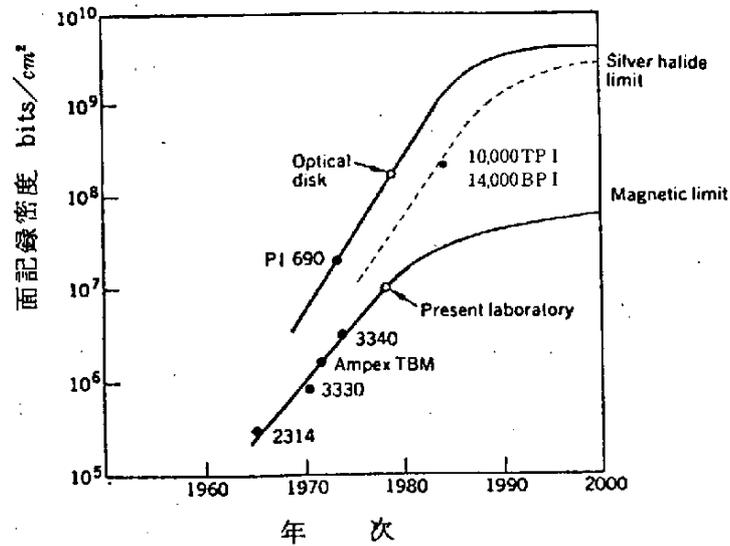


表 2-1 各種メモリー・デバイスの平均的なアクセス・タイム比較

タイプ	1980年	1985年	1990年	1995年
NMOSダイナミックRAM	0.2 $\mu$ s	0.1 $\mu$ s	0.05 $\mu$ s	0.03 $\mu$ s
C C D	8 $\mu$ s	3 $\mu$ s	1 $\mu$ s	1 $\mu$ s
バブル	2 ms	1 ms	0.5 ms	0.5 ms
ジョセフソン			2.5 ns	2 ns
磁気ディスク	25 ms	20 ms	15 ms	10 ms
光ディスク		100 ms	25 ms	20 ms

} 半導体メモリー  
} 磁気記録メモリー  
} 光メモリー

表 2-2 各種メモリー・デバイスの容量比較

NMOSダイナミックRAM	64KB/チップ	256KB/チップ	1MB/チップ	4MB/チップ
C C D	同上	同上	同上	同上
バブル	256KB/チップ 1MB/チップ	1MB/チップ	4MB/チップ	16MB/チップ
ジョセフソン			1MB/チップ	4MB/チップ
磁気ディスク	2500 MB/装置	1000 MB/装置	10 GB/装置	20 GB/装置
光ディスク		1 GB/面	1.5 GB/面	3 GB/面

表2-3 各種メモリー・デバイスのコスト比較

(ビット当りユーザ・コストをミリセントの単位で示している)

タイプ	1980年	1985年	1990年	1995年
NMOSダイナミックRAM	180	40	10-20	5-10
CCD	100	20	5-10	2-5
バブル	70	30	5-15	1-8
ジョセフソン			10-20	5-10
磁気ディスク	0.8	0.4	0.2-0.3	0.2
光ディスク		0.04	0.01-0.02	0.01

表2-1~表2-3 出典：  
 ○ ADL社「将来のコンピューター技術、市場、および環境に関する調査報告書」  
 ○ 電子通信学会誌「高密度光ディスクの現状と将来」  
 ○ 「メモリー特集 Vol.60 No.11、1977」  
 ○ 「ジョセフソン素子」 「情報処理に関する調査研究報告書Ⅱ」  
 ○ The Bell System Technical Journal Vol.58, No.6, pp.1453-1540  
 ○ CCDについては、データがなく推定したものをを用いてある。

図2-4 ストージ部品のコストの動向

(前表より作成)

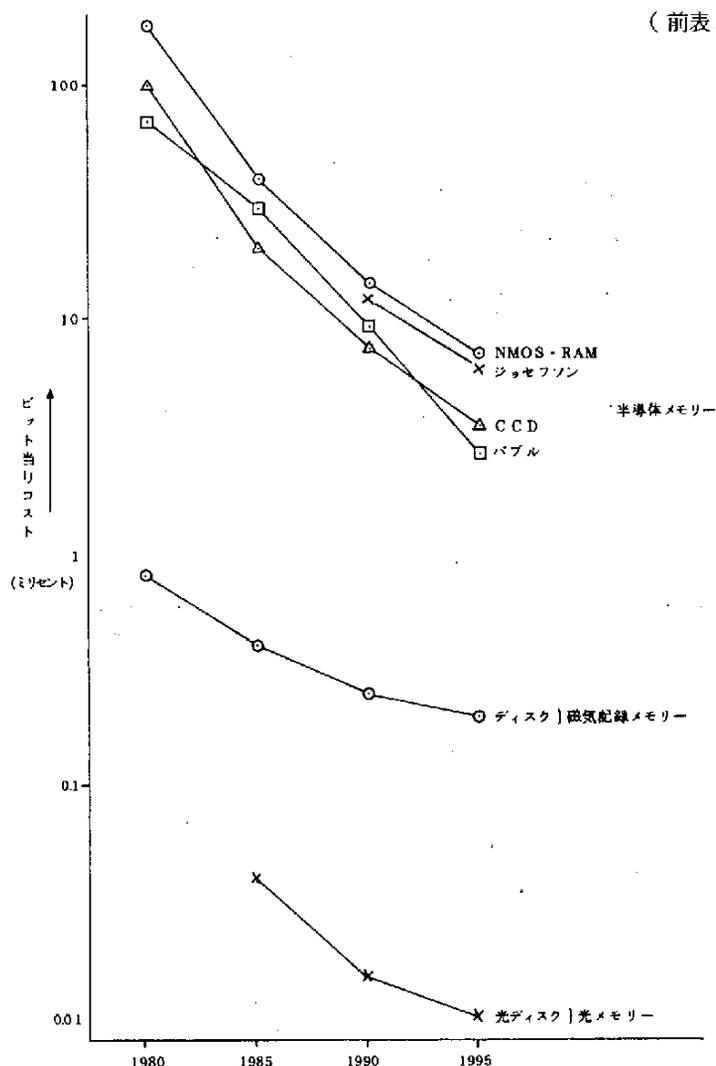


表 2 - 4 ストージ部品のコストの現状

出典：RCA ADVANCED TECHNOLOGY LABORATORIES

	MAGNETIC DISC	COMPUTER COMPATIBLE TAPE	HIGH DENSITY MAGNETIC TAPE	SILVER HALIDE	OPTICAL DISC
10 <sup>11</sup> BITS EQUIVALENT STORAGE	80 DISC-PACKS (200 MEGABYTES PER PACK)	90 TAPES (EACH 2400 FEET, 6250 BITS PER INCH, 8 TRACKS)	ONE 2400-FOOT ROLL OF 2-INCH TAPE	200 FISCHE 4 X 6 INCH	ONE 12-INCH DISC
MEDIA COST FOR 10 <sup>11</sup> BITS	\$40,000 ~ \$ 80,000	\$1350 ~ \$ 1,800	\$100	\$60	\$10

### 3. 電子光学部品

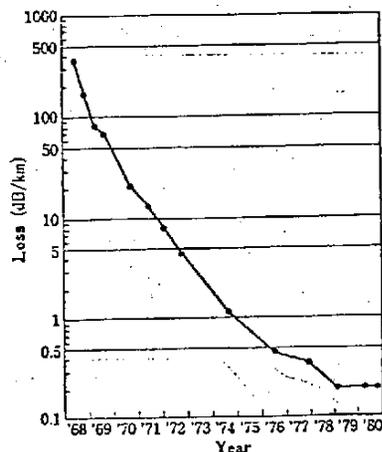
レーザ技術、光ファイバ技術、薄膜導波路技術などの急速な発展にともないオプトエレクトロニクス技術は、従来のエレクトロニクス技術だけではなしえなかった分野にも進出している。電子光学部品は、①レーザ素子（気体レーザ素子、固体レーザ素子、半導体レーザ素子など）、②レーザ素子を除く発光素子（LED発光素子など）、③受光素子、光検出器（PD）、④光ファイバ、光ファイバケーブル、⑤光IC・部品（光スイッチ・変調器、光コネクタなど）、⑥画像関連デバイス（表示装置、撮像装置など）等に分類することができる。

電子光学部品は、光通信をはじめ、光ディスクメモリー、光ビデオディスク、POS、レーザプリンタ、ファクシミリ、レーザ応用計測（レーザレーダ、医療用など）など広範な分野で利用されている。低損失ファイバの開発、それに伴う半導体レーザの開発などにより電子光学部品、特に長波長帯の素子は急速な発展を行なっている。その意味では、現状の値は日々更新を必要とするものである。以下においてはコンピュータ関連の観点から光通信の基本部品に限り、光ファイバ、半導体レーザ（LD）、発光ダイオード（LED）、PINホトダイオード（PIN）、アバランシェホトダイオード（APD）を中心にその動向を掲げる。

現在では、長波長帯（ $1.0 \sim 1.6 \mu\text{m}$ ）で高性能、高信頼性をもつ素子の開発・改良、実用性を高めコストダウンをはかるためのIC化、アレイ化素子の開発、新機能光素子の開発などを中心に研究・開発が行なわれている。

#### (1) 光ファイバ

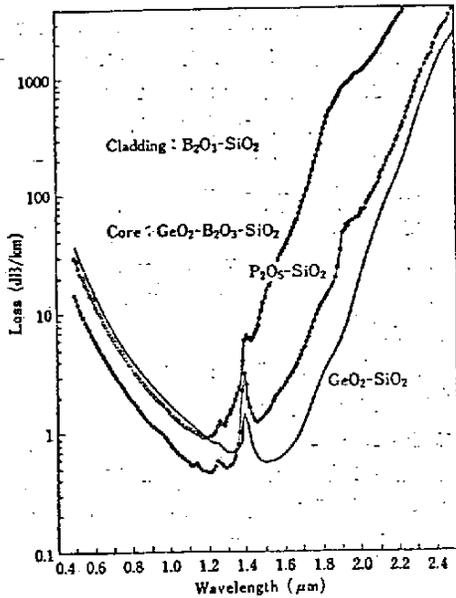
図3-1 光ファイバの低損失化



Loss reduction tendency of optical fibers,  
by the world-wide reported best datum in  
each year.

出典：中原基博、技広隆夫、稲垣伸夫：光ファイバの低損失化、応用物理、Vol. 50、No.10、1981.

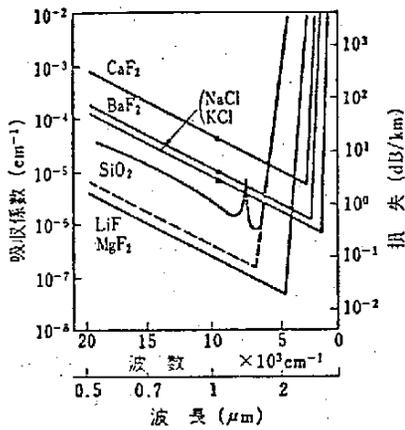
図3-2 光ファイバの損失スペクトル



Transmission loss spectra of three kinds of optical fibers.

出典：中原基博、技広隆夫、稲垣伸夫：光ファイバの低損失化、応用物理、Vol. 50、No.10、1981.

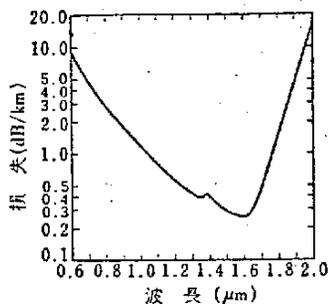
図3-3 低損失材料の損失波長特性



出典：末松安晴：光伝送技術特集—将来展望、電子通信学会誌、Vol. 63、No.11、1980. 10.

伊沢、森、木村；長波長帯超低損失光ファイバ用材料、昭54信学総全大、S3-15.

図3-4 光ファイバの損失波長特性



超低損失シリカ光ファイバの損失波長特性

出典：末松安晴：光伝送技術特集—将来展望、電子通信学会誌、Vol. 63、No.11、1980. 10.

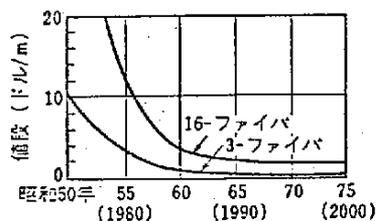
Moriyama, T., Fukuda, O., Sanada, K., Inada, K., Tanaka, S., Chida, F. and Edahiro, T. : "Fabrication of ultra-low content optical fibers with VAD method", to be presented 6th ECOC, York, 1980

表 3 - 1 代表的な光ファイバの仕様

	単一モード ファイバ	マルチモードファイバ				
		集束形ファイバ	石英ステップ形 ファイバ	ポリマクラッド ファイバ	全ポリマファイバ	
コア径	10 $\mu$ m	50~100 $\mu$ m	50~150 $\mu$ m	100~150 $\mu$ m	200~1,000 $\mu$ m	
クラッド径	125 $\mu$ m	125 $\mu$ m	125~200 $\mu$ m	500 $\mu$ m	1mm	
伝送損失	0.8 $\mu$ m	3dB/km	3dB/km	3dB/km	10dB/km	~1,000dB/km
	1.3 $\mu$ m	1dB/km以下	1dB/km以下	1dB/km以下	10dB/km	~1,000dB/km以上
伝送帯域幅	~40GHz $\cdot$ km	~2GHz $\cdot$ km (LD, 1.3 $\mu$ mLED) ~100MHz $\cdot$ km (0.8 $\mu$ mLED)	10~30MHz $\cdot$ km	~10MHz $\cdot$ km	~5MHz $\cdot$ km	

「出典：桑原洋、田中満雄、谷中雅雄、樫尾次郎：産業用光伝送システムの現状と動向、日立評論、Vol. 63、No.3、1981. 3.」

図 3 - 5 価格動向



光ファイバケーブルの値段予測

出典：・末松安晴：光伝送技術特集—将来展望、電子通信学会誌、

Vol. 63、No.11、1980. 10.

・Montgomery, J. D. and Dixon, F. W.: "Fiber optics to the year 2000", Opt. Spectra, PP. 53-56, May 1980.

(2) 発光・受光素子

表 3 - 2 発光・受光素子の現状

光半導体素子の種類と現状

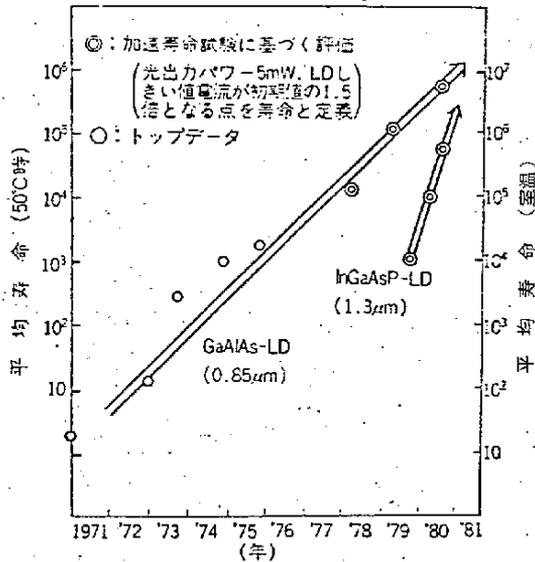
光半導体素子	技術の現状	
発光素子	レーザダイオード (LD)	・ GaAlAs (0.8 $\mu$ m帯) …実用 ・ InGaAsp (1.0~1.6 $\mu$ m) …一部実用途上
	発光ダイオード (LED)	・ GaAlAs (0.8 $\mu$ m帯) …実用 ・ InGaAsp (1.0~1.6 $\mu$ m) …一部実用
受光素子	PIN・ホトダイオード (PIN・PD)	・ Si (0.8 $\mu$ m帯) …実用 ・ InGaAs (1.0~1.6 $\mu$ m) …一部実用途上
	アバランシェ・ホトダイオード (APD)	・ Si (0.8 $\mu$ m帯) …実用 ・ Ge (1.0~1.6 $\mu$ m) …実用 ・ InGaAs (P) (1.0~1.6 $\mu$ m) …開発中

出典：工藤達夫：光半導体素子、電子技術、

Vol. 23、No.1.

① 半導体レーザー

図 3-6 半導体レーザーの長寿命化の歴史



出典：堀口孝雄、岩崎匡利：光ファイバケーブル伝送方式の展望、施設、Vol. 33, No.4, 1981. 4.

表 3-3 代表的な半導体レーザーの仕様

GaAlAs系と InGaAsP系レーザーダイオードの特性

特性項目	GaAlAs系	InGaAsP系
発振波長 $\lambda_p$ ( $\mu\text{m}$ )	0.85	1.3
閾値電流 $I_{th}$ (mA)	~60	~70
光出力 $P_o$ (mW)	~7	~5
横モード	基本	基本
発光スペクトル	単一波長	単一波長
ビーム放射角 (度)	10×40	10×40
特性温度 $T_0$ (°C) 注1)	130	70
立ち上がり・立ち下がり時間 (ns)	0.5	0.5

出典：工藤達夫：光半導体素子、電子技術、Vol. 23, No.1.

注1)  $I_{th} \propto \exp(T/T_0)$

注：光ファイバの低損失領域が  $0.8\mu\text{m}$  帯から  $1.3\mu\text{m}$  帯、更に  $1.5\mu\text{m}$  帯へと変わっていくにつれて、レーザーの波長もこれに対応した展開が必要となっている。

(  $0.8\mu\text{m}$  は GaAlAs )

(  $1.3 \sim 1.5\mu\text{m}$  は InGaAsP や GaAlSbAs )

更に  $3 \sim 5\mu\text{m}$  帯で極低損失ファイバが出現すると、新材料の研究が必要となる。

大容量、長距離通信用としては地上、海底幹線とも当面  $1.3\mu\text{m} \sim 1.5\mu\text{m}$  帯、しかもシングルモード伝送が用いられることになる。一方、中容量、小容量伝送、サブスクライバ用とし

では、0.8 $\mu$ m帯が用いられ、大幅なコストダウンが図られるであろう。

レーザの特性としては出力、スペクトル、温度特性のほかに、モード、ノイズ、直線性、寿命など益々高度な特性が要求されてくる。

レーザのコストダウンを図るためには、需要を増やすことが大切であり、通信用だけでなく民生用、情報処理用などの広範囲な用途を開拓することが必要である。現在すでにVideo disc, Printer, POSなどへの応用が開けつつある。また光ICの一部としての発展がある。

## ② 発光ダイオード (LED)

表 3 - 4 代表的なLEDの仕様

0.8 $\mu$ m帯LEDの特性例			長波長帯 InGaAsP LEDの代表的特性			
特 性 項 目	透明層形LED	透明層レンズ形LED	特 性 項 目	LED (A)	LED (B)	LED (C)
発振波長 ( $\mu$ m)	0.85	0.85	発光ピーク波長 $\lambda_0$ ( $\mu$ m)	1.15	1.27	1.50
スペクトル半値幅 (nm) <sup>注1)</sup>	40	40	スペクトル半値幅 $\Delta\lambda$ ( $\mu$ m)	0.09	0.10	0.13
ファイバ端光出力 ( $\mu$ W) <sup>注1)</sup>	250	700	ファイバ端光出力 $P_f$ <sup>注1)</sup> ( $\mu$ W)	110	200	110
遮断周波数 (MHz)	30	30	遮断周波数 $f_0$ <sup>注2)</sup> (MHz)	30	50	50
順電圧 (V)	1.7	1.7	順電圧 $V_f$ (V)	1.39	1.17	0.95

注1) 動作電流: 100mA

ファイバ: SI形, コア径85 $\mu$ m, NA0.18

注1)  $P_f$ : step index, NA = 0.22, 85 $\mu$ m core

注2)  $f_0$ : -1.5 dB point

出典: 工藤達夫: 光半導体素子, 電子技術, Vol. 23, No.1.

注: LEDもレーザと同様、ファイバの領域に応じた発展がある。長寿命、低価格、簡易のメリットを活かして中、小容量通信に広く使用されると予想されている。

最大のネックはファイバとの結合効率が低いことでこれを解決することが肝要である。アナログ伝送にも使い易く、コンピュータ間伝送を含め到る処に使用されるであろう。

## ③ PINホトダイオード (PIN)

表 3 - 5 代表的なPINの仕様

波 長 帯	0.8 $\mu$ m帯	1.3 $\mu$ m帯
基 板 材 料	Si	Ge InGaAsなど
量 子 効 率	$\geq 60\%$	20~60%
暗 電 流	$\leq 1$ nA	$\sim 1$ $\mu$ A

出典: 桑原洋、田中満雄、谷中雅雄、樫尾次郎: 産業用光伝送システムの現状と動向、日立評論、Vol. 63, No.3, 1981. 3.

注: 1.6 $\mu$ mまではGeや InGaAsP, InGaAs, GaAlSbなどの材料が用いられるが、これを越えると新材料の研究が必要となる。

PIN単体でなく増幅器と集積化して用いられることになろう。

民生用としてもLEDやLDのdetectorとして多数用いられるであろう。

④ アバランシェホトダイオード (APD)

表 3 - 6 代表的な APD の仕様

項目	記号	Si-APD の例			Ge-APD の例			単位		
		条件	最小値	標準値	最大値	条件	最小値		標準値	最大値
量子効率	$\eta$	$\lambda = 0.83 \mu\text{m}$ , 直流	60	68	—	$\lambda = 1.3 \mu\text{m}$ , 直流	40	60	—	%
遮断周波数	$f_c$	$M = 10, R_L = 50 \Omega$ 出力 100KHz 値より -3dB	100	—	—	$M = 10, R_L = 50 \Omega$ 出力 100KHz 値より -3dB	400	600	—	MHz
		$M = 50, R_L = 50 \Omega$ 出力 100KHz 値より -1.5 dB	300	—	—					
降伏電圧	$V_B$	$I_D = 100 \mu\text{A}$	130	—	180	$I_D = 100 \mu\text{A}$	25	30	—	V
降伏電圧の温度係数	$r$ (注1)		—	0.5	0.6		—	0.1	—	%/°C
暗電流	$I_D$	$V_R = 0.9 V_B$	—	0.3	3	$V_R = 0.9 V_B$	—	0.5	5	$\mu\text{A}$
過剰雑音係数	F	$\lambda = 0.83 \mu\text{m}$	—	4.5	6.0	$\lambda = 1.3 \mu\text{m}$	—	10	—	—
		$M = 100, I_{F0} = 2.0 \mu\text{A}$				$M = 10; I_{F0} = 1.0 \mu\text{A}$				
	x	$f = 30 \text{MHz}$	—	0.33	0.39	$f = 30 \text{MHz}$	—	1	—	
端子間容量	$C_t$	$V_R = 90 \text{V}, f = 1 \text{MHz}$	—	1.5	2.0	$V_R = 20 \text{V}, f = 1 \text{MHz}$	—	2	3	pF

注1)  $r = \frac{V_B(25^\circ\text{C} + \Delta T^\circ\text{C}) - V_B(25^\circ\text{C})}{V_B(25^\circ\text{C}) \cdot \Delta T^\circ\text{C}} \times 100 (\%/^\circ\text{C})$

注2) ショット雑音電流  $i_N$  は、増倍過程のゆらぎにより、 $i_N^2 = 2qI_{F0}M^2 + xB$  にしたがって増加し、過剰雑音係数 F は  $F = M^2$  で定義される。

ここに、 $q$  : 電荷素量;  $I_{F0} = M$  が 1 のときの光電流;  $M$  : 増倍率;  $B$  : 信号のバンド幅

出典: 工藤達: 光半導体素子, 電子技術, Vol. 23, No. 1.

註:  $1.6 \mu\text{m}$  を越えると、新材料の研究が必要となる。低雑音の APD にはイオン化率比という物質定数が関係するので、適切なイオン化率比を持つ材料を見つけることも大きい課題である。そのための結晶成長技術も重要である。

逆方向で均質なブレイクダウンを起こさせるためには高度に均質な結晶、界面特性のよい多層成長、イオン注入法、パッシベーション法などの開発が必要である。

(3) 光通信システムの動向

表 3-7 現在の光伝送システム及び研究・開発中のシステム

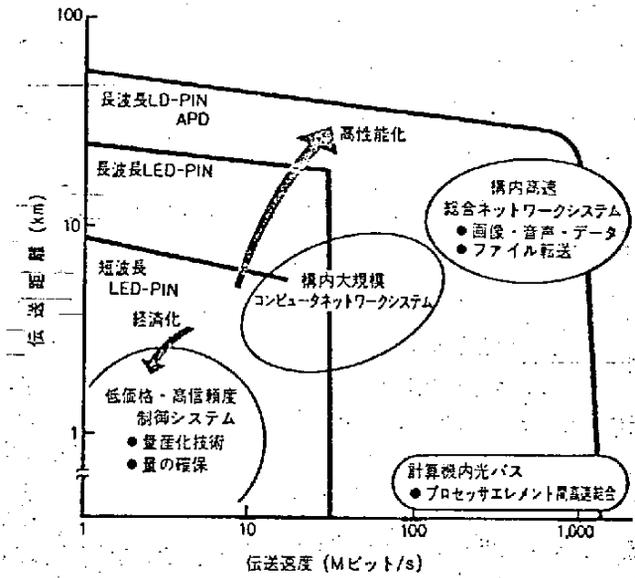
光通信のシステム	特 徴
[第1世代システム] (実用中) <u>短波長光伝送</u> 波 長 : 0.8~0.9 $\mu$ m 光 源 : LD, LED 検 波 器 : LED, pin PD	多モード光伝送技術 (LED-pin PD) (LD-APD) 光強度変復調 波長多重, 双方向通信技術 光スイッチ デバイス材料: GaAlAs/GaAs, Si
[第2世代システム] (実用化中) <u>長波長多モード光伝送</u> 波 長 : 1.3 $\mu$ m 光 源 : LED 検 波 器 : pin PD	低損失多モード光伝送技術 (零屈折率分散波長を使用) 光デバイスのアレー集積 デバイス材料: GaInAsP/InP, Ge
[第3世代システム] (研究・開発・実用化中) <u>長波長単一モード光伝送</u> 波 長 : 1.5~1.65 $\mu$ m 光 源 : LD 検 波 器 : pin PD	極低損失単一モード光伝送技術 光強度変復調 時間・波長多重, 双方向通信技術 コヒーレント光通信 (ヘテロダイン) 集積レーザ, 光集積回路 デバイス材料: GaInAsP/InP, Ge
[第4世代システム] (基礎研究中) <u>全光通信</u> 波 長 : 1.5~1.65 $\mu$ m	光交換 (光スイッチ, 光演算) 高機能光集積回路 (単一モード光伝送技術)
[第5世代システム] (基礎研究中) <u>超長波長光通信</u> 波 長 : >2 $\mu$ m	超長距離光通信の可能性の追求中

出典: 末松安晴: 光伝送技術特集-将来展望、電子通信学会誌、Vol. 63, No 11, 1980, 10.

LD: レーザダイオード, LED: 発光ダイオード,  
 pin PD: 受光ダイオード, APD: アバランシェ光ダイオード

図 3 - 7 今後の動向

出典：桑原洋、田中満雄、谷中雅雄、樫尾次郎  
 ：産業用光伝送システムの現状と動向、  
 日立評論、Vol. 63, No 3, 1981. 3.



光適用システムの今後の動向

光適用システムの今後の方向としては、高性能化と経済化  
 を目指す二つの大きな流れがあることが分かる。

#### 4. 大容量記憶装置

表 4 - 1 記憶装置の現状

	開発者製品名	容 量	アクセスタイム	データ転送速度	そ の 他
磁気バブルメモリー装置	富士通 FBM54DA	1Mb/chip	11.2ms	100Kb/s	バブル径 1.9 $\mu$
	Intel iSBC254-4	0.5MB~1MB	48ms	50KB/s	
	日 立	64Mb/UNIT	2ms	80Mb/s	
磁気ディスク装置	I B M 3380	2.52GB/UNIT	*1.6ms **8.3ms	3MB/s	
	N T T	3.2GB/UNIT	*1.8ms **1.0ms	1.34MB/s	8 inch
磁気テープ装置	I B M 3420-8	6250BPI		1.25MB/s	
光ディスク装置	東 芝 TOSFILE	10 <sup>10</sup> b	0.5 s	1Mb/s	
M S S	I B M 3850	236GB/UNIT	16 s	7Mb/s	

注 \*平均シークタイム \*\*平均回転まち時間 (研究発表段階のものを含む)

B: Byte b: bit MSS: Mass Storage System

図 4 - 1 1981~1985年における記憶装置の推移

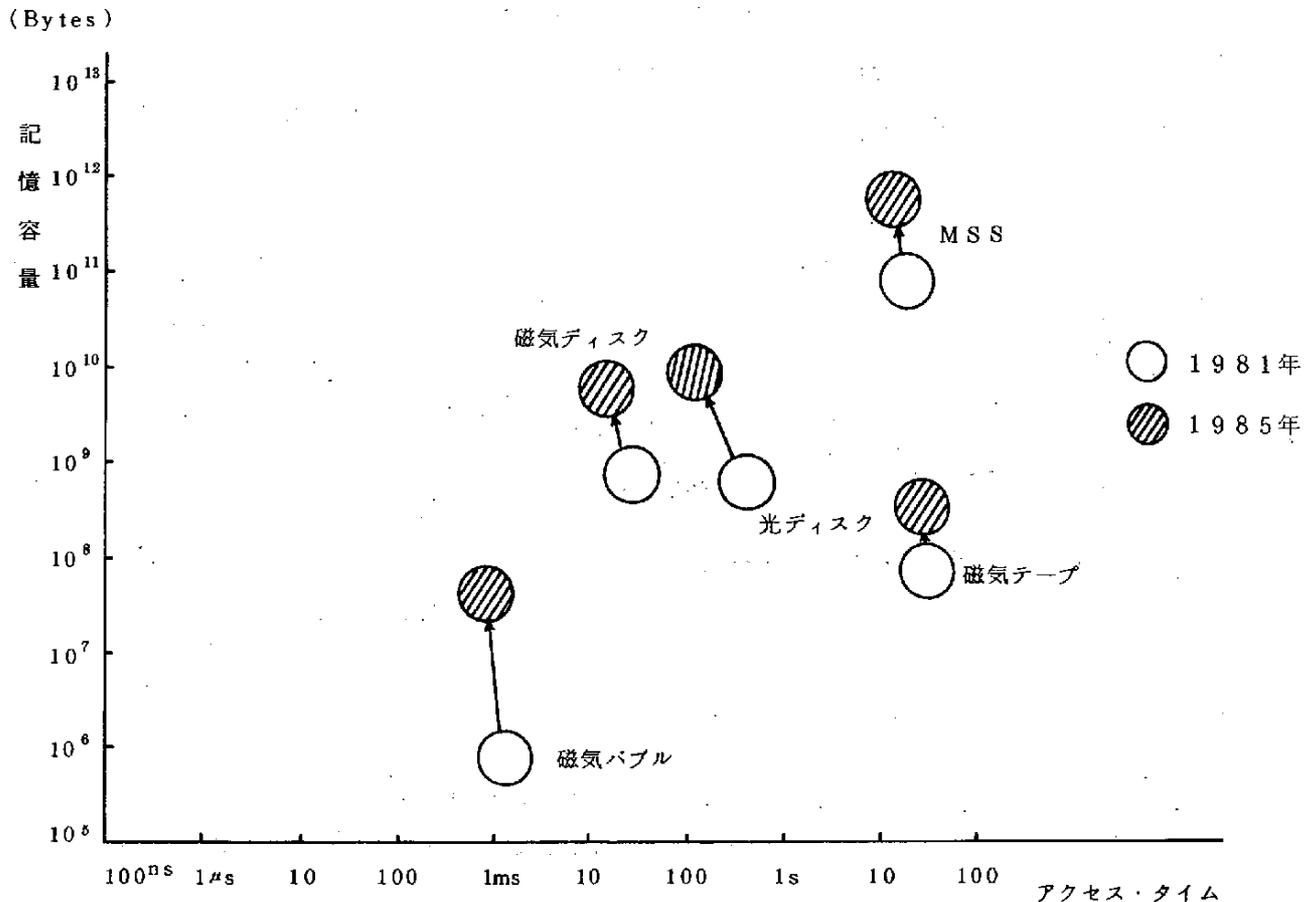


表4-2 記憶装置の性能推移

1981年

	容量	アクセス タイム	データ 転送速度	コスト
磁気バブル メモリー装置	1 MB	2 ms	$5 \times 10^{-2}$ MB/s	100 m¢/b
磁気 ディスク装置	2500	25	3	1
磁気テープ装置	160	70000	1.25	$1 \sim 10^{-2}$
MSS	236000	16000	0.9	$10^{-1}$
光ディスク装置	$10^3$	500	$1 \sim 50$ Mb/s	$10^{-4}$

1985年

容量	アクセス タイム	データ 転送速度	コスト
50~100 MB	1 ms	$5 \times 10^{-2} \sim 10^{-1}$ MB/s	m¢/b
2500~5,000	20	6	$5 \times 10^{-1}$
160~500	70000	1.25~4	$1 \sim 10^{-3}$
$10^6$	10000 12000	1~5	$10^{-1} \sim 10^{-2}$
$10^3 \sim 10^4$	50~100	10~200	$10^{-5}$

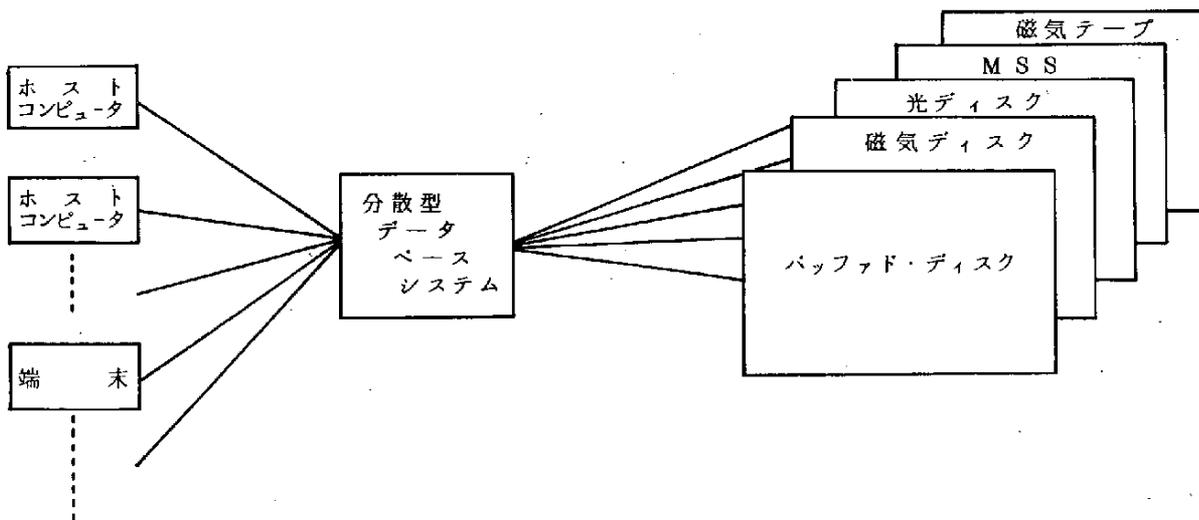
○磁気ディスク装置は、薄膜あるいは垂直磁化技術による記録密度の向上が期待され、アクセスタイムやコスト面から1985年頃においても記憶装置の中心的地位を占めている。

○光ディスクは、1985年までには書き換え可能なものの実用化は難しいと考えられるため、データバンク的な利用に留まるだろう。  
また、現在のところビットエラー率が $10^{-3} \sim 10^{-5}$ と高いため、文書ファイル、画像ファイルなどが主たる用途となる。

大容量記憶システムについて

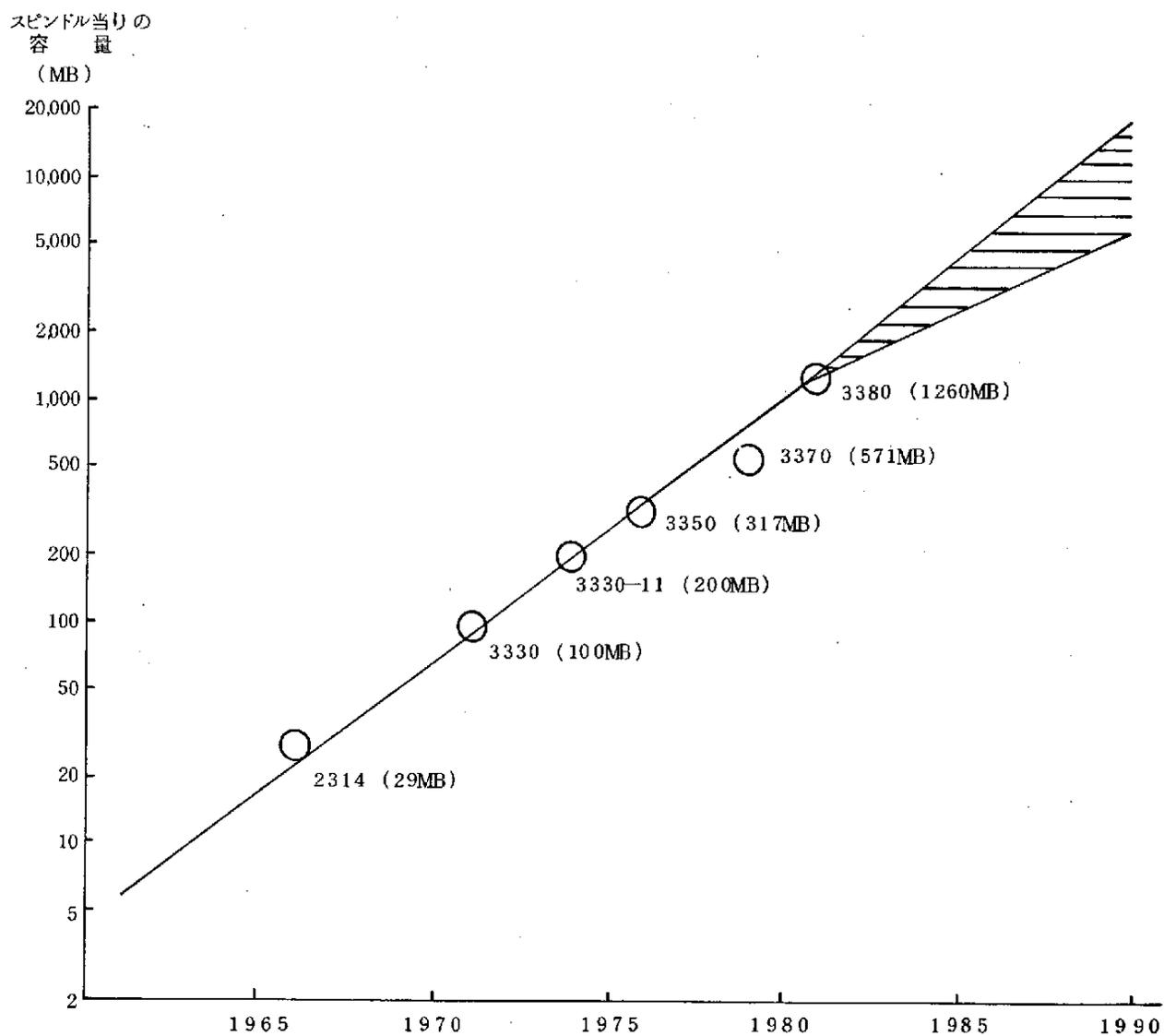
大容量記憶装置の今後の動向は、磁気記憶技術や光ディスクなど、個別のデバイス・レベルの技術の進展に依存するところが大きく、現在のところ画期的な技術は予測できない。

図4-2 大容量記憶システムの例



注) 予測される大容量記憶システムは、ワンレベルストアの概念に基づき、一方でデータベースプロセッサなどの出現によって、データが一元的に管理され、かつ複数のホストマシンあるいは端末から自由にアクセスできるものとなる。

図 4 - 3 磁気ディスクにおける記憶容量の推移



- 記憶密度向上のための技術として、・薄膜技術、・垂直磁気記録技術、が期待される。
- 過去の製品の傾向を単純に延長して考えると1スピンドル当り、1985年には5GB、1990年には20GB近くなると想定される。
- この図はこれまでのIBMの製品の傾向をもとに作成したものの。

表 4 - 3 光ディスク装置の開発状況

	容 量 (bit)	アクセスタイム (set)	転送速度 (b/sec)	備 考
東 芝 (TOSFILE)	$10^{10}$	0.5	1M	モノクロ
日 立	$10^{10}$	0.3		モノクロ
三 菱 TEAC	静止画 24,000枚 動 画 20分	1.2		カ ラ ー
Harris	$10^9$	<15	50M	
R C A	$10^{11}$	0.5	200M (マルチチャンネル)	
Philips	$10^9$		<10M	直径 20 cm
	$10^{11}$		50M	直径 30 cm

\*RCA, Philips 社のものは今後の開発目標

## 5. アーキテクチャ

### 5.1 アーキテクチャの分類と開発状況

#### 1. まえがき

ソフトウェア生産性の重視、VLSI技術への適応、計算機利用層の飛躍的な拡大、通信技術を核とした新アプリケーションの展開など、つぎつぎに折り重なって打寄せる'新しい波'によって、計算機の歴史の中で、これまで不動の規範であり続けてきたノイマン・アーキテクチャの成立基盤が大きく浸蝕を受け、計算機アーキテクチャにおける発想の転換が迫られている。

ノイマン・モデルに代わる新しいアーキテクチャ — 総括的に非ノイマン・アーキテクチャと呼ばれている — の提案は、アイディアのレベルではノイマン・モデルの成立からそれ程遅くない時期にまで遡りうるものもあり、実験的に実現された例も少なくない。しかし、それがアーキテクチャの必然的なあり方の追求として認識されるようになったのは、それ程旧いことではない。この認識が形成されるためには、まず、新しい時代からの計算機に対する要請が何であるかが明らかにされる必要があり、つぎに、この要請とノイマン・アーキテクチャの本質的な矛盾の存在が立証されなければならないのであるが、こうしたことが仮説的であるにせよ、具体的に論議しうるようになったのは、ここ数年来のことである。したがって、それ以前の非ノイマン・アーキテクチャ研究が、今日の問題意識に対して十分に応えるものではないのは仕方のないところである。

これからの計算機アーキテクチャの進むべき方向を支配する追求目標、いいかえれば、うえにいった問題意識の内容のうち、主要なものを挙げるとつぎのようになる。

- 高性能化
- 使い易さ
- 高信頼性
- 新しい応用
- VLSI技術の活用

高性能化には2つの意味がある。すなわち、高速化という意味と、いわゆるコスト/パフォーマンスの改善という意味である。これらはどの時代にも共通の課題である。しかし、問題の具体的な内容と解き方は時代によって大きく変わっている。使い易さが特に強調されるようになったのは、比較的最近のことである。これにも2つの側面があり、専門的なプログラムのソフトウェア生産性向上の面と、マイコン利用などでの非専門家による計算機使用の局面でのものがある。高信頼性もどの時代にも共通する基本的課題であるが、計算機の果たす社会的な役割がすでに致命的な水準に達していることから、これの重要性はいまさらいうまでもないとこ

るである。最近ではハードウェアの信頼性よりプログラムの信頼性に関心に移りつつあるのが特徴である。新しい応用として、とりわけ革新的な意味の強いものとしては、人工知能研究成果の実用化として展開されている分野がある。自然言語処理、画像処理、音声処理、知識ベースなどがその例である。VLSI技術の進歩は極めて目覚ましいものであるが、VLSIの特性が指示する方向は必ずしも従来の計算機技術が追求してきた方向とは一致していない。VLSIの特質を活用するためには、計算機技術をVLSIに都合のよい方向に適合させるしかないというのが、現在の共通認識である。

以下では、いま述べたような追求目標に沿って、新しいアーキテクチャの方向を分類し、技術的な可能性の見取図としてまとめることにしたい。

#### アーキテクチャの分類

##### (1) 高性能化アーキテクチャ

###### ○ 並列処理アーキテクチャ — 高速性の追求<sup>2),3),4)</sup>

並列処理とは、1つのJOBを、複数個の独立実行可能な処理単位(手続のレベルからオペレーション(命令)のレベルまでさまざま)に分割し、これらを複数個の装置を使用して、同時に並行して処理することをいう。

並列処理計算機の分類には、つぎのような概念が用いられるのが普通であるが、(Flynn, 1966)

① SIMD: 単一命令ストリーム・複数データストリーム

② MISD: 複数命令ストリーム・単一データストリーム

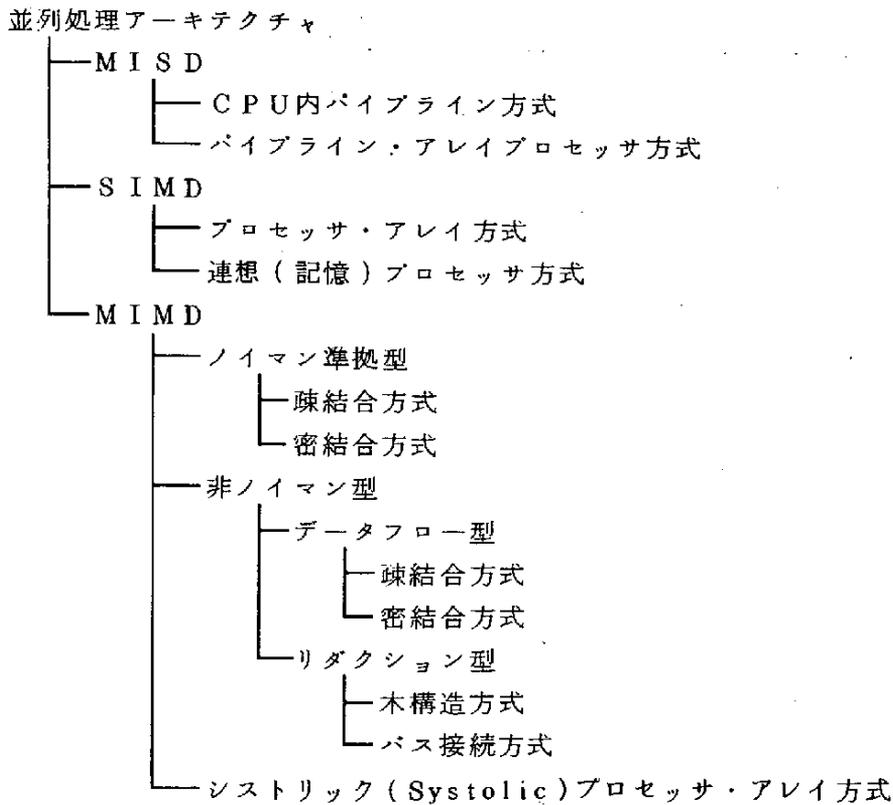
③ MIMD: 複数命令ストリーム・複数データストリーム

この命令ストリームという概念の背景には、命令カウンタの存在が仮定されているので、この分類の中に非ノイマン・アーキテクチャを取込むことができなくなる。また、データストリームという概念もややあいまいなところがある。したがって、ここではSIMDなどの概念から、ストリームということばをはずし、例えば、MIMDについていえば、それは同時に複数の命令と複数のデータが処理されるという意味であるものとする。したがって、SIMDでは一時に処理される命令は1つだけであるが、それで操作されるオペランド群は複数個ということになる。(さらに、高級言語計算機に典型的にみられるような、アーキテクチャの高水準化という事実を考えると、命令とかデータの概念にも再検討を加えるべきかも知れないが、ここでは一応現在の常識の線にしたがっておくことにする。)

表5-1 並列処理アーキテクチャの実例と応用分野

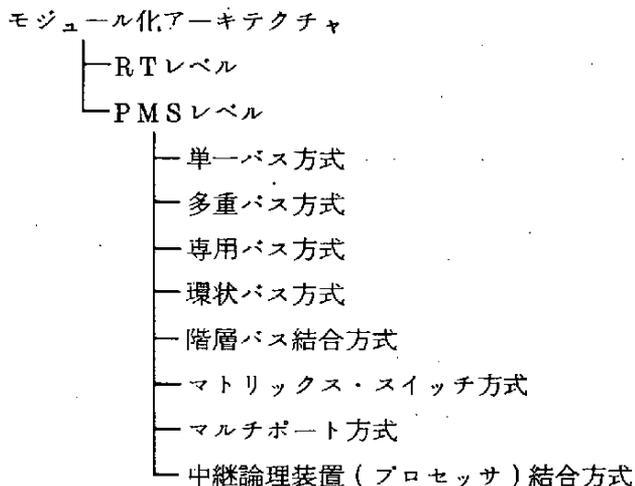
分類	説明		並列処理レベル	代表的なシステム	適用分野			
MISD	CPU内部の並列処理 (パイプライン)		命令またはステートメント・レベル	(大部分のシステムに採用されている)	(ソフトウェアからは、アーキテクチャの並列構造は見えない)			
	パイプライン方式を用いたアレイプロセッサ			CRAY-1 STAR-100 ASC など	• 図形/画像/信号処理			
SIMD	プロセッサ・アレイ型の並列プロセッサ (アレイプロセッサ)			ILLIAC-IV BSP DAP など	• 大規模科学技術計算処理			
	連想プロセッサ			STARAN PEPE ECAM ALAP など	• データベース処理			
MIMD	非ノイマン型	データフロー計算機		密結合	MIT (Dennis) Irvine (Arvind)	• 信号処理 • 大規模科学計算 • 汎用		
				疎結合				
		リダクション計算機		木構造			Keller Mago	• 記号処理
				バス接続			Berkling	• 記号処理
	シストリックアレイ・プロセッサ	2次元		1次元	Kung			
				<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> その他				
	複数台の計算機を結合したマルチプロセッサやポリプロセッサ、マルチマイクロプロセッサノイマン型	密結合	共通バス	タスクまたはサブシステム・レベル	BTI8000 SMS201 など	• 汎用マルチプロセッサによる汎用処理		
			マトリックススイッチ		C. mmp B7800 など	• 多数プロセッサによるシミュレーション、天気予報などの特殊分野の処理		
			マルチポート		UNIVAC1100/80 IMB370/168 など			
			階層構成等		Cm* など			
疎結合			HXDP など		(並列処理の色彩はうすくなる)			

図 5-1 並列処理アーキテクチャの分類



○モジュール化アーキテクチャ — 高速性、コスト/パフォーマンス追求<sup>2),3)</sup>

同一の機能回路、CPU、メモリ、チャネル、あるいは入出力機器等を、モジュールとして用意し、必要に応じて組合せてシステムを構成するものである。これによって、同じハードウェア設計によって広い範囲のパフォーマンスをカバーでき、したがって、設計コストを下げる事ができる。その他の利点は、つぎの通り。



○ 適応型アーキテクチャ — 高速性、コスト/パフォーマンス追求<sup>5),6)</sup>

従来の、いわゆる汎用計算機においては、その命令体系を中心とするアーキテクチャの設定に当って、できるだけ広範な問題に適応することが追求された。そのため、平均的な処理効率がえられた反面において、ある特定の応用や予め想定されていなかった新しい問題への対応において、著しい不利を招く結果になっていた。

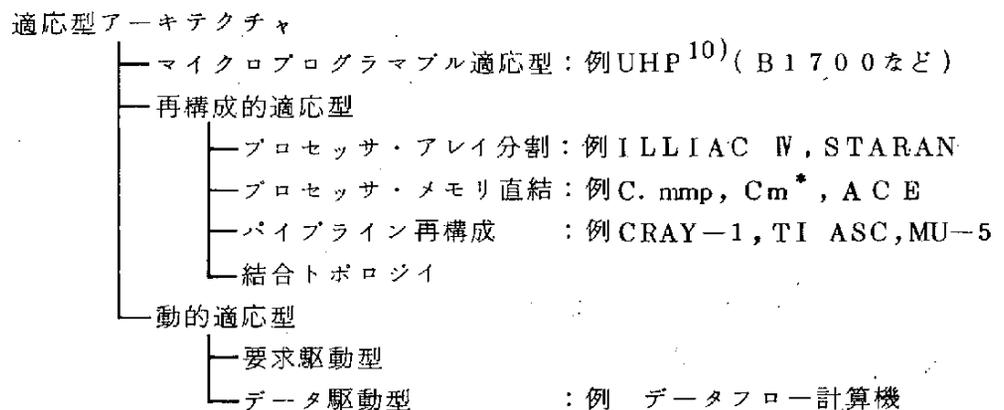
そこで考えられるのは、応用目的に応じて随時命令体系などの設定を変えうるような、柔構造ないし可変構造的なアーキテクチャの可能性である。適応型アーキテクチャはこの可能性を実現しようとするものである。適応型アーキテクチャをやや厳密に定義すると、'ソフトウェア的手段によって、いま計算しようとしているプログラムに適応化することができるような構造と機能を実現するためのアーキテクチャ'ということができる。<sup>6)</sup>

適応型アーキテクチャの分類には、まだ十分に定説化したものはないようであるが、ここでは一応文献<sup>6)</sup>にしたがっておくことにする。

- ① マイクロプログラマブル適応型
- ② 再構成的適応型
- ③ 動的適応型

この分類は、適応化がハードウェア構造のどの水準に対して作用するかに基づくものである。すなわち、マイクロプログラマブル適応型の場合には、レジスタ、加算器、計数器といった機能要素間の接続関係が再構成され、それによって、実行すべきアルゴリズムに合わせてマイクロプログラムの適応化が行われるのである。つぎの再構成的適応型は、プロセッサ、メモリ、そしてI/O装置といったレベルの機能要素間に可変的な接続関係を用意しておくことによって適応化を計るものである。動的適応型は、LSIないしVLSIによる計算機モジュールの相互接続関係を可変にするものである。すなわち、多数の計算機モジュールによって構成されるシステムにおいて、並行処理されるプログラムの間

図 5-2 適応型アーキテクチャの分類



で配分されるべきモジュールの数や組合せを動的に調節することによって、同量のハードウェアによって処理しうるプログラムの個数を最大にしようとするものである。

○特殊化/専用化アーキテクチャ — 高速性、コスト/パフォーマンス追求<sup>7),8)</sup>

汎用計算機という考え方が生まれた背景には、多くの事情が関係している。その第一は、何といてもハードウェアが極めて高価であったため、1台で何にでも使えるようにする必要があったことである。半導体技術の進歩は、この事情を完全にくつがえしつつある。その第二は開発コストの問題である。明らかに機種が少ないほど有利である。この事情は一見現在も同じであるように見えるが、問題の中味はかなり異なる。VLSI設計ならびにソフトウェア開発の困難性からくる影響は、むしろ、新しい事情とみるべきかも知れない。第三の事情は、いわゆる「グロッシュの法則」である。個別に小さな計算機をもつよりも、集中的にできるだけ大きな計算機を入れて多目的に使った方が有利というのが、かつての状況であった。しかし、これも一方で半導体技術の進歩があり、他方でオンライン処理という新しい応用形態が発展したことにより、分散処理の考え方にとって代わられることになった。

この分散処理の考え方は、企業内におけるシステムのあり方を示すものであるが、この考え方は計算機システム内部におけるプロセッサのあり方にもそのまま通ずるものである。すなわち、すべてを1台のCPUプロセッサでまかなう方式から、機能別に複数台のプロセッサをおいて専用化する、いわゆる機能分散方式への推移である。この傾向は、入出力機能の高度化、通信処理機能の普及、画像処理や音声処理などの新しい機能の一般化に伴って、ますます、その勢力を強めており、適用範囲は大型機はもちろん、超小型機(例えばIBMシステム/38)にまで広がり、機能内容も細分化・専門化が深化する一方である。

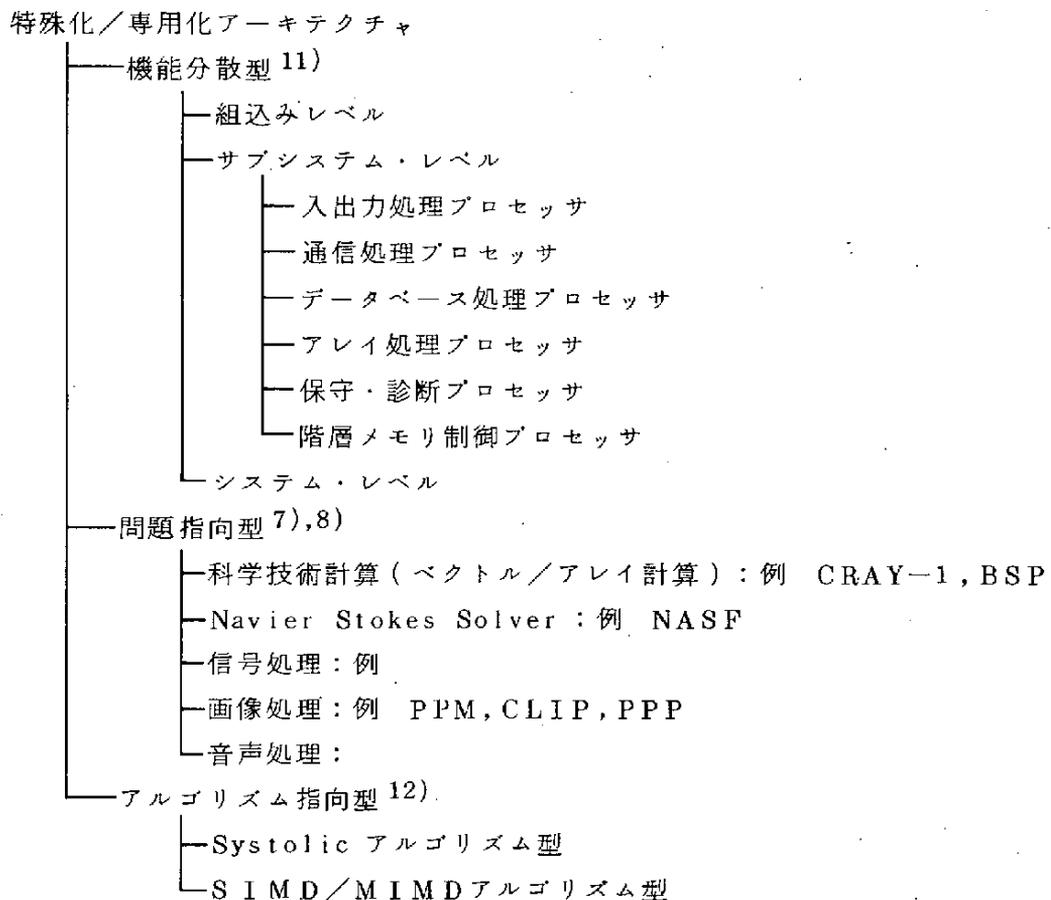
一方、こうした一般商用機での動向とは別に、科学技術計算の分野を中心に、大規模な計算を高速で処理するためのアーキテクチャの追求が、以前から営々と続けられてきている。すなわち、気象予報、地震波の解析(資源探査)、原子核物理、航空工学、画像処理などでは、大量のデータを対象にして、連立偏微分方程式の数値計算など複雑な演算処理を、一定時間内に(天気予報の場合を考えてみればよい)消化する必要がある。こうした要求を満たすためには、計算機を最初から問題向きに設計せざるを得ない。そして、資金的な裏付けを得た問題に対してアーキテクチャの開発が進められることになる。その結果出現してきたのは、偏微分方程式処理を目的にしたベクトルないしアレイ・プロセッサ、波形解析を目的にした信号プロセッサ、航空写真やX線写真の解析のための画像プロセッサ、さらに信号プロセッサを一段と特殊化したものとして音声プロセッサなどである。

これらの問題向き専用計算機では、高速化の手段として高度なパイプライン処理方式または並列処理方式を採用している。前者の方式によるものをパイプライン型、後者の方式によるものを並列処理型と呼んでいる。なかには、両方式を組合せたものもある。最も高度な並

列処理型では、多数のプロセッサ・エレメントをアレイ状（他にツリー状、リング状などもある）に配列したものがあり、特にプロセッサ・アレイ型と呼ばれることが多い。

このプロセッサ・アレイ型のように、多数のプロセッサが一定の形状構造の下に相互結合された計算機を使いこなすことは、かなり面倒なことである。同じ問題に対するプログラムであっても、アルゴリズムの選び方によって、えられる処理性能に雲泥の差が出るからである。すなわち、プロセッサ相互結合の構造に対するアルゴリズムの適合性の幅が極めて狭いのである。しかも、こうした計算機が利用されるのは大規模な問題であり、それに対するアルゴリズムの工夫はそれ自身大きな問題なのである。このために、実際にハードウェアが作られながら、ほとんど活用される機会に恵まれないという高度並列処理計算機の例も実見される。現在予見されるVLSI技術の方向は、高度並列処理アーキテクチャを有利にするものであることを考えると、このプロセッサ相互結合の構造形式とアルゴリズムの強い結びつきという事実をベースにして、アーキテクチャのあり方を再検討する必要があることに気付く。そこで考えられるのは、アルゴリズム開発をアーキテクチャ設計に先行させ、アルゴリズムに合わせてアーキテクチャを決めるというアプローチである。しかも、さらに一歩進

図5-3 特殊化/専用化アーキテクチャの分類



めて、アルゴリズムを直接VLSIチップとして実現するところに行きたくて行くことを想定して  
 である。ここまで徹底するまえに、もう一つのアプローチとして、プロセッサ相互結合に適  
 応型アーキテクチャの考え方を適用することが提案されている。すなわち、再構成可能な相  
 互結合方式を追求しようというものである。<sup>9)</sup>

表5-2 科学技術計算専用コンピュータ

名 称	メーカ	処理方式	稼働または 発表時期
STAR-100	Control Data Corp.	パイプライン	1973
ASC	Texas Instruments	パイプライン	1972
ILLIAC IV	Burroughs	並列処理 (プロセッサ・アレイ)	1973
CRAY-1	Cray Research	パイプライン	1976
BSP	Burroughs	並列処理	1977発表
DAP	International Computer Ltd.	並列処理 (プロセッサ・アレイ)	1979
CYBER-203	Control Data Corp.	パイプライン	1979発表
FACOM 230/75APU	富士通	パイプライン	1976
HITAC M-200H IAP	日立	パイプライン	1979

(2) '使い易さ' 指向アーキテクチャ

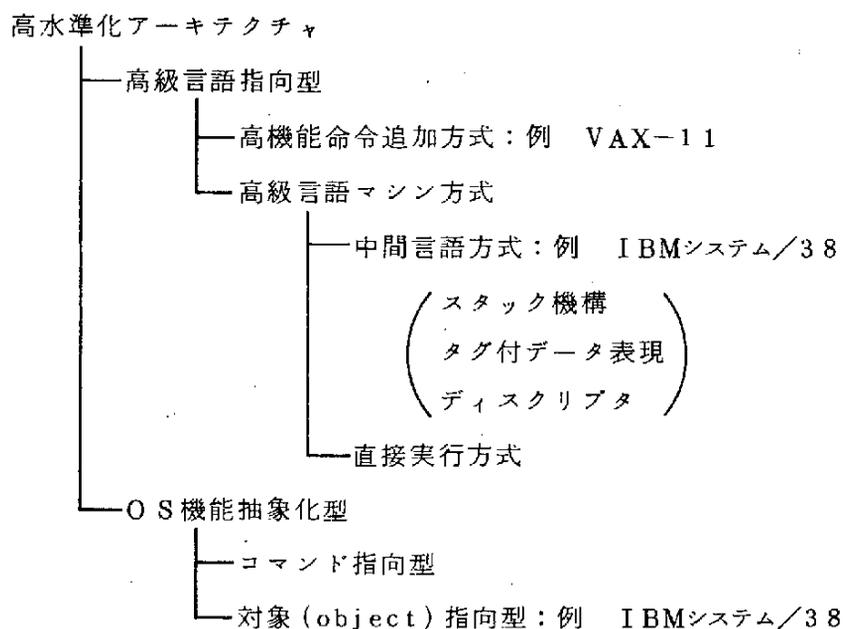
○高水準化アーキテクチャ<sup>13)</sup>

計算機において、特に'使い易さ'ということが問題にされるようになった原因の1つは、  
 計算機システムと、それで処理すべき問題が大規模化し、その複雑さのためにプログラミング  
 の生産性に低下の傾向が心配されるようになったことにある。一般に、複雑さを克服するた  
 めの最も強力な手段は、抽象化を行うことである。高水準化アーキテクチャの目的は、人間と  
 計算機の接触面(マシン・インタフェースと呼ぶ)に現われる諸概念をより抽象度の高いもの  
 にし、しかも、それによって計算機のパフォーマンスが損われることのないように合理化する  
 ことにある。その意味では、つぎの項で扱う新言語指向アーキテクチャと共通の狙いをもつも  
 のであるが、この場合には従来からの高級言語に対して肯定的な立場をとり、専らその枠内で

問題を考えるところに特徴がある。したがって、基本的にはノイマン・アーキテクチャの枠組を脱け出ることがない。

そのため、ここでの関心は専ら、いわゆる機械語レベルでのインタフェースの高水準化に集中することになる。最近では、一般ユーザが直接機械語によってプログラミングすることはなくなっているが、実際に実行されるのはあくまでも機械語プログラムである。この間のギャップはコンパイラによる変換操作で埋められている。この変換操作の介在によって、ユーザ・プログラムとその実行過程の間に成立するはずの直接的な対応関係が崩れるため、例えば、プログラムにエラーが内在したとしても、それがそのものズバリの現象として出現するとは限らず、正体不明の異常現象となって出沒することになる可能性もある。こうした問題を解消するためには、高級言語と直接的な対応づけが可能な水準に機械語を持ち上げることが必要になる。極端な考え方をすれば、高級言語そのものを機械語とする計算機ができないかということになる。しかし、デバッグの完了したプログラムを繰返し実行することの多い、通常の計算機利用の局面では、プログラム実行でのパフォーマンスが高い従来のコンパイル方式を抜くことは困難である。したがって、妥当な線としてはいわゆる中間言語のレベルに落ち着くことが多い。

図5-4 高水準化アーキテクチャの分類





人間 — 計算機インタフェースに機械語が用いられることの多い局面に、システム・プログラムのプログラミングがある。この分野では、標準的な高級言語には用意されていない特殊な機能（OS機能と呼ばれる）が利用される。こうした機能を具現化するのにはもちろんのこと、それらを利用するのにも機械語プログラミングが必要である。そこで、こうした機能を集約化して抽象化し、標準的な言語機能として体系化することができれば、システム・プログラムの作成や保守が大いに効率化されることになる。これを徹底すれば、従来の機械語はほとんど使う必要がなくなり、したがって、機械語を無視して直接そうした機能のハードウェア化を実現する方式が有利になってくる。

#### ○新言語（新計算モデル）指向アーキテクチャ<sup>14)</sup>

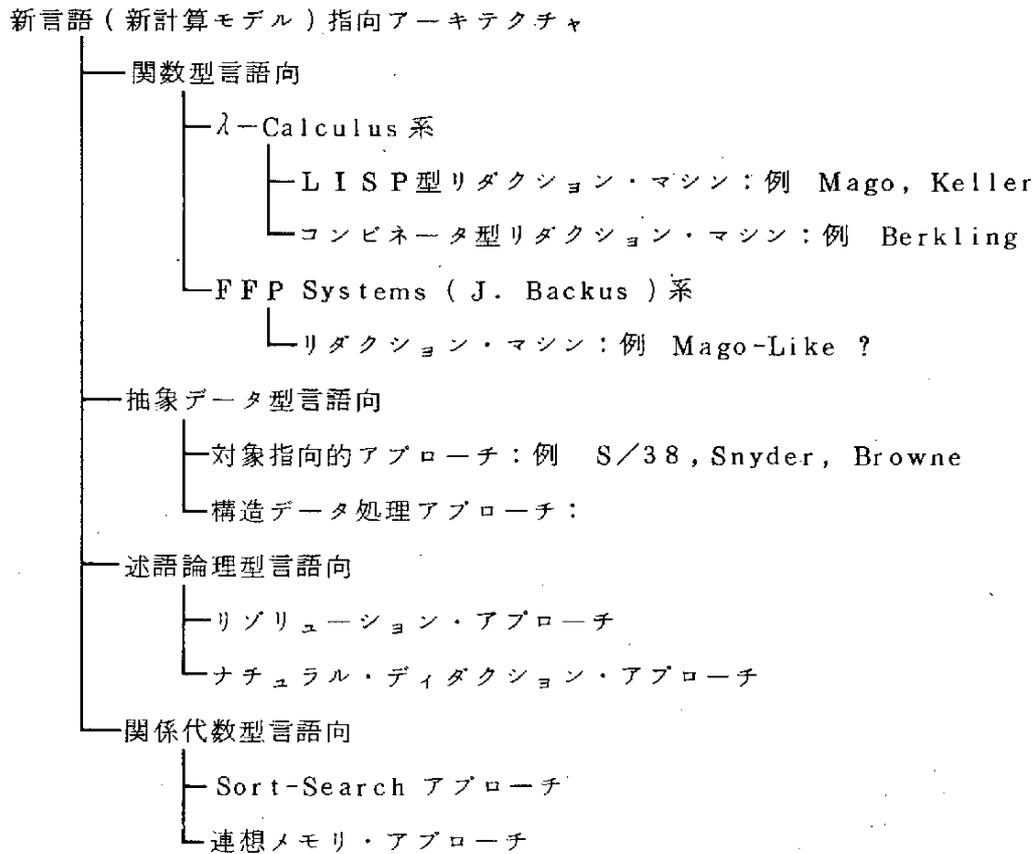
使い易さの追求は、自然な流れとしてよりよいプログラミング言語を求める動きとなる。このために、すでに多数の言語が開発されているが、いくら数が殖えても強力さという点では大同小異の状況が続いている。この頭打ちの原因は何か、それはこれまでの言語設計の最深部にある基礎に本質的な欠陥があるからではないのか、従来はあるマシンのイメージを出発点にとって、その上で使うものとしてプログラミング言語を設計したが、これは本来のプログラミング言語の発達を拘束することになっているのではないか。マシンがあって言語がくるのではなく、言語ができて次にそれにふさわしいマシンが作られるべきではないのか。このような発想の逆転が必要であるというのが、新言語指向アーキテクチャの背景にある考え方である。この発想の逆転は、丁度、計算機システムのコストに占めるハードウェアとソフトウェアの比重の逆転に対応している点が注目される。

こうした考え方の萌芽は、すでにパロースのB1700の開発に認められるが、本当の意味でのマシンの呪縛からの解放を行ったのは、J. バッカスの有名な論文、'Can Programming Be Liberated from the von Neumann Style? A Functional Style and Its Algebra of Programs'である。J. バッカスのノイマン・モデル批判に賛成するかどうかは別にしても、問題の深さがどの程度のものであるかを示ただけでもこの論文の意義は十分に大きいと認められる。

既成のマシン概念から自由になったとして、新しいプログラミング言語のあるべき姿は何か、それをサポートすべきアーキテクチャのあり方は何か。

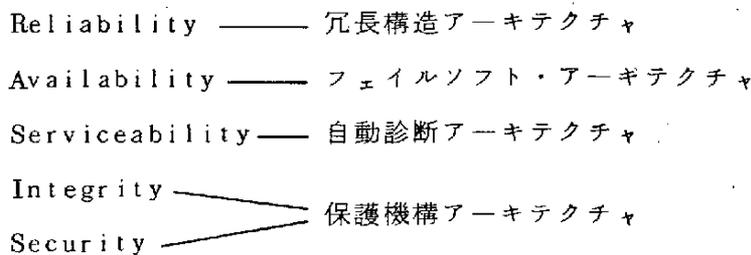
以下に列挙するものは、それへの解答の試みであり、まだ十分な答えになっていないものも多い。

図5-5 新言語（新計算モデル）指向アーキテクチャの分類



(3) 高信頼性アーキテクチャ

ここでは信頼性ということばを最も広い意味に解決して、いわゆる R A S I S (Reliability, Availability, Serviceability, Integrity, Security) の全体を含んでいるものとする。そうすると、これらの各項目ごとに、それぞれ特徴的なアーキテクチャ上の工夫を対応させることができる。

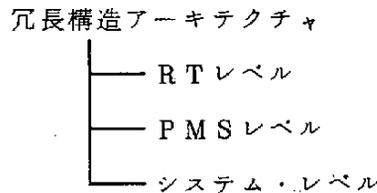


RASあるいはRASISの概念が意味するところは、もちろんこれらを個別に考えるのではなく、統合化して一体のものとも見たときに、それらの最大の効果が発揮されるということであるが、以下では分類の便宜上、それぞれを独立にとりあげることにする。

○冗長構造アーキテクチャ

同じ機能目的をもつハードウェア機構を複数個用意し、相互に代替できるようにして、障害の発生に備えるものであるが、多重化構成の仕組みや運用の仕方には、いろいろなバリエーションが工夫されている。各レベルとも重複方式とスタンド・バイ方式が区別できる。

図5-6 冗長構造アーキテクチャの分類

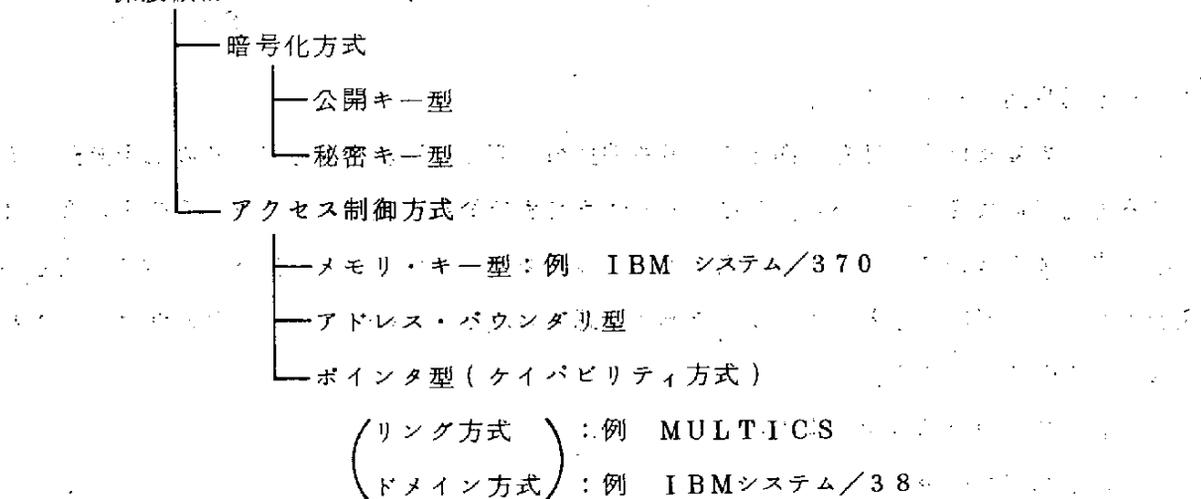


○保護機構アーキテクチャ

情報保護機構の必要性は繰返し強調されながら、一般的な実用はあまり進んでいないのが実情のようである。しかし、その重要性は次第に高まっている。保護機構には物理的な手段によるものと、論理的な手段によるものがあるが、ここで扱うのは後者のみである。

図5-7 保護機構アーキテクチャ

保護機構アーキテクチャ

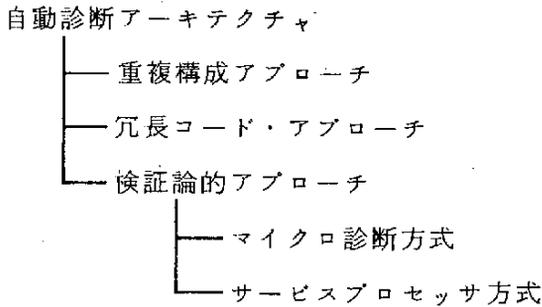


・自動診断アーキテクチャ

計算機システムが複雑化すれば、その内部に発生する障害に対する診断も複雑化し、表面的な現象をみて人間が判断するというような方法では、十分な対処ができないことは明らかである。計算機自身による組織的な診断体系が用意されていなければならない。診断のポイントは、障害の有無を判定することと、あると分かった場合にその所在を知ることの2点である。前者

については、何らかの冗長構造が利用され、後者については検証論的なアプローチが重視されるようになってきている。

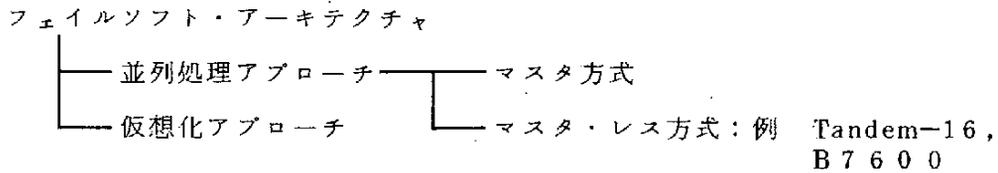
図 5-8 自動診断アーキテクチャの分類



・フェイルソフト・アーキテクチャ

基本的には、並列処理的手法と仮想化手法が有力である。

図 5-9 フェイルソフト・アーキテクチャの分類



(4) 新応用指向アーキテクチャ

コストの大幅な低下、性能の向上など計算機技術の著しい進歩によって、その応用分野はあらゆる方向に拡大されつつある。このようなさまざまな発展の様子をみると、その中の多くは一応従来技術の延長によって十分対応しうるものであるが、いくつかの方向については新しい発想にもとづく対応を必要とするような動きが認められる。そのうちでも、特に注目すべき動きは、つぎの2つである。

- 計算機利用の拡散化・零細化
- 人工知能的応用

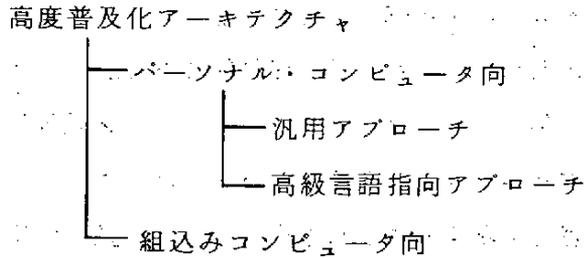
これらに対しては、アーキテクチャの立場からも突込んだ検討が必要である。そこから、アーキテクチャの新しいあり方が生まれてくる可能性が期待される。

・高度普及化アーキテクチャ

すでにマイクロコンピュータの浸透ぶりは目覚ましいものであるが、それを利用するためのプログラミングという意味でのマン・マシン・インタフェースを考えてみると、そこに本当の意味での素人が入ってきているという状況には至っていない。現在の素人プログラミングは、

かなりの学習時間を費やした上で、定型的なプログラムを書いているというのが実態ではないだろうか。プログラミングが自家用車の運転と同じような水準にまで滲透しうるためには、自動車がかようなように、計算機もその内部構造を新しい状況に適応させていく必要がある。

図 5-10 高度普及化アーキテクチャの分類

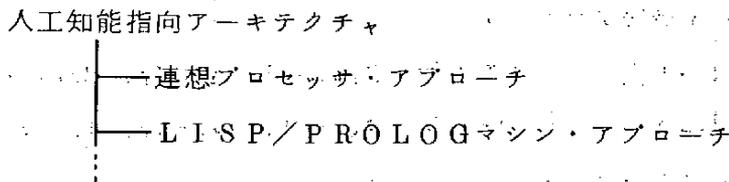


○人工知能指向アーキテクチャ

自然言語処理、画像処理、音声処理など人工知能研究の成果は、すでに着々と実用化されつつあるが、新応用の本質的な部分がここから開拓されていることは疑いのないところである。しかし、よく指摘されるとおり、<sup>14)</sup> 図形や文章の処理など非数値データの処理に計算機を使おうとすると、その性能の低さに驚かされることが多い。人工知能に関連した研究が基礎研究の域を脱することの出来ぬ理由<sup>14)</sup>がここにある。

ではどうすればよいのか、どんな可能性が考えられるのが、それは今後の研究課題である。

図 5-11 人工知能指向アーキテクチャ



(5) VLSI 指向アーキテクチャ

VLSI 技術は計算機にとって多くの利点をもつものである。その利点の主なものを挙げると、

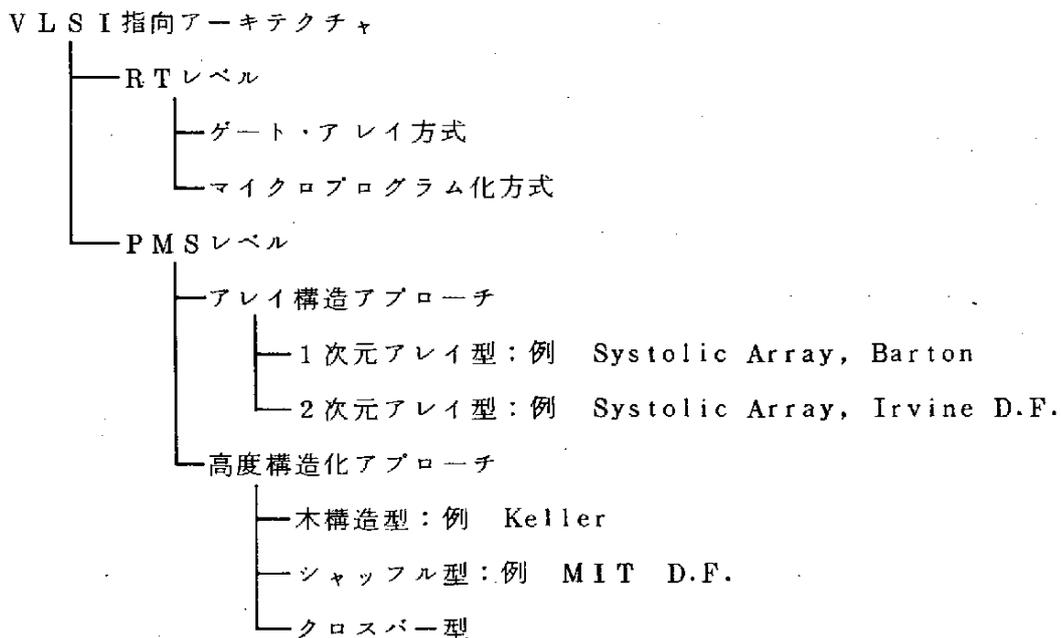
- 要素あたりのコストの低下
- 電力消費量と物理サイズの減少
- 組立作業量の減少
- 回路レベルでの信頼性の向上

などがある。しかし、これらの利点は計算機の設計において、無条件で利用できるものではなく、活用するためにはアーキテクチャ上でもいくつかの難しい問題を解決しなければならない。

このような問題のうちで最も重要なものは、ソフトウェア設計の場合と同じもので、複雑さが増すと設計の生産性が急速に低下するという事実である。これによる設計コストの上昇は、上述の要素あたりコストの低下を打消すほどの大きさに達する可能性がある。これは従来からのCADを利用して克服不可能である。考えられる対応策は、やはりソフトウェアの時と同様で、構造化設計、モジュール化設計ということになる。もし、チップ全体の設計が、少数の簡単な部分設計に分解でき、それらの部分設計を単純なインタフェースを介して繰返しの組合わせれば全体が完成するという形式に還元できれば、最も理想的である。もう少し具体的にいえば、簡単な構造をもった機能要素（例えばプロセッサ・エレメント）を規則的に（例えば行列状に）配列するという形で計算機が実現できるようにしたい、ということである。

これは明らかに極めて高度な並列処理を実現することに相当している。並列処理アーキテクチャの項で説明したように、プロセッサ間の接続関係の構造は、アルゴリズムのあり方を厳しく拘束するものである。プロセッサ配列の規則性と接続関係の自由度をいかにバランスさせるかに、アーキテクチャの課題の1つがある。

図5-12 VLSI指向アーキテクチャの分類



参考文献

- 1) Organick, E.I. : 『コンピュータシステム アーキテクチャの技術動向』, (昭和53年度特別セミナー講演録), 電子協(1979.3)
- 2) JIPDEC : 『並列処理』, 第5世代の電子計算機に関する調査研究報告書 — アーキテクチャ研究分科会 —, PP. 124~135, JIPDEC(1980)

- 3) JIPDEC: "マルチプロセッサ", 第5世代の電子計算機に関する調査研究報告書 — アーキテクチャ研究分科会 —, PP. 136~150, JIPDEC (1980)
- 4) JECC: "将来のコンピュータ技術, 市場, および環境に関する調査報告書", JECC (1979.2)
- 5) JIPDEC: "第5世代の電子計算機に関する調査研究報告書 — 計算機の適応・学習機能", JIPDEC (1980)
- 6) Vick, C. R. et al.: "Adaptable Architectures for Supersystems", Computer, Vol. 13, No. 11, PP. 17~35 (1980)
- 7) 小高, 河辺: "超高速演算の動向", 情報処理, PP. 927~937, (1980.9)
- 8) JIPDEC: "科学計算", 第5世代の電子計算機に関する調査研究報告書 — アーキテクチャ研究分科会 —, PP. 165~181, JIPDEC (1980)
- 9) シーゲル, H. J ほか: "再構成可能な並列処理システムにおける相互結合方式の動向", 日経エレクトロニクス, 1979.12.24号 PP. 49~62
- 10) JIPDEC: "ユニバーサル・ホスト・プロセッサ", 第5世代の電子計算機に関する調査研究報告書 — アーキテクチャ研究分科会 —, PP. 194~203, JIPDEC (1980).
- 11) " : "機能分散型計算機", " PP. 182~193 "
- 12) Kung, H. T.: "The Structure of Parallel Algorithms", Advances in Computers, Vol. 19, PP. 65~112. Academic (1980).
- 13) JIPDEC: "高級言語マシン", 第5世代の電子計算機に関する調査研究報告書 — アーキテクチャ研究分科会 —, PP. 204~221, JIPDEC (1980).
- 14) 元岡: "コンピュータアーキテクチャの動向と展望", 情報処理, Vol. 21, No. 5, PP. 558~565 (1980.5)
- 15) Kung, H. T., : "Notes on VLSI Computation", Lecture Note in CREST Parallel Processing Systems Course, (1980).

### 3. 代表例によるアーキテクチャ分類表

前節に述べた分類概念を、実際の機種に適用した結果の一覧表を以下に示す。ただし、各機種についての情報が不完全な場合が多く、また、誤認も少なくないと思われるので、試行的なものとしての取扱いが必要である。

	開発者	開発水準	並列処理アーキテクチャ										モジュール化アーキテクチャ		
			MISD		SIMD		MIMD						RT レベル	PMS レベル	
			CPU内 パイプ ライン 方式	パイプ ライン アレイ プロセッサ 方式	プロセッサ アレイ 方式	連想 プロセッサ 方式	ノイマン準拠型		非ノイマン型		シストリック プロセッサ・アレイ 方式				
							疎結合 方式	密結合 方式	データフロー型			リダクシオン型			
						疎結合	密結合	木構造	バス接続						
(商用大型機)	(計算機メーカー)	商用	○						○						
スーパーコンピュータ															
ASC	Texas Inst.	実験(1972)		○											
ILLIAC IV	イリノイ大	実験(1973)			○										
STAR-100	CDC	商用(1973)		○											
CRAY-1	Cray Research	商用(1976)		○											
FACOM 230/75 APU	富士通	商用(1976)		○											
BSP	パロース	商用(1977)			○										
DAP	ICL	商用(1979)													
HITAC M-200H IAP	日立	商用(1979)		○											
CYBER-203	CDC	商用(1979)		○											
CYBER-205	"	商用(1980)		○											
S-1	スタンフォード大	実験													
データフロー計算機															
MIT DFM	MIT (Dennis)	研究(1974) 実験(1980)									○				○
RAUシステム/1	ツールーズ大	実験(1976)									○				○
ID DFM	UCI (Arvind)	研究(1978)									○				○
GE DFM	Rumbaugh	研究(1977)									○				○
DDM1	ユタ大 (Davis)	実験(1978)									○				○
TOPSTAR	東大	実験(1978)													
リダクシオンマシン															
Berkling マシン	GMD (Berkling)	研究(1976) 実験(1978)										○	○		
Keller マシン	ユタ大 (Keller)	研究(1978)										○			○
Magó マシン	ノースカロライナ大	研究(1979)										○			
Cambridge マシン	Cambridge 大	実験(1980)													

	開発者	開発水準	並列処理アーキテクチャ										モジュール化アーキテクチャ				
			MISD		SIMD		MIMD						R T レベル	PMS レベル			
			CPU内 パイプ ライン 方式	パイプ ライン アレイ プロセッサ 方式	プロセッサ アレイ 方式	連 想 プロセッサ 方式	ノイマン準拠型		非ノイマン型						シストリック プロセッサ・アレイ 方式		
							疎結合 方式	密結合 方式	データフロー型		リダクション型						
				疎結合	密結合	木構造			バス接続								
データベース・マシン RAP 1/2 DBC SDD-1 (STARAN型) EDC ADABASマシン TRAC	トロント大 オハイオ州立大 CC of America シラキュース大 電研総 Software AG テキサス大(Dale)	実験(1976/7) 研究(1976) 実験(1978) 実験(1979) 実験(1979) 商用(1980) 研究(1979)			○												○ ○ ○ ○ ○
UHP(ユニバーサル・ホスト・プロセッサ) QM-1 B1700/1800 FCPU Cons CDC-5600 ACE IBM5100	Nanodata パロース Data Saab MIT(Greenblatt) CDC 電研研 IBM	商用(1972) 商用(1972) 商用(1972) 研究(1974) 実験(1977) 実験 商用(1977)															
高級言語マシン SYMBOL APL Assist COMBAT FLATS LISPマイクロプロセッサ	フェアチャイルド IBM 日電 理研 MIT(Sussman)	実験(1971) 商用(1976) 実験(1976) 研究(1978) 実験(1980)							○								
その他 C.mmp Cm*	CMU CMU	実験(1972) 実験(1976)								○ ○							○ ○

	開発者	開発水準	適応型アーキテクチャ						特殊化/専用化アーキテクチャ					
			マイクロ プロ ラムブル 適応型	再構成的適応型			動的適応型		機能分散型			問題指向型 科学技術計算/ Navier Stokes/ 信号処理/ 画像処理/ 音声処理	アルゴリズム指向型	
				プロセッサ アレイ 分割	プロセッサ メモリ 直結	パイプ ライン 再構成	結 合 トポロ ジイ	要 求 駆動型	デ ータ 駆動型	組 込 み レ ベル	サ ブ シ ス テ ム レ ベル		シ ス テ ム レ ベル	Systolic Algorithm 型
(商用大型機)	(計算機メーカー)	商 用									○			
スーパーコンピュータ														
ASC	Texas Inst.	実験(1972)			○							○		
ILLIAC IV	イリノイ大	実験(1973)	○									○	科学計算・画像処理	
STAR-100	CDC	商用(1973)										○	科学技術計算	
CRAY-1	Cray Research	商用(1976)			○							○	"	
FACOM 230/75 APU	富士通	商用(1976)										○	"	
BSP	パロース	商用(1977)								○		○	"	
DAP	ICL	商用(1979)										○	"	
HITAC M-200H IAP	日立	商用(1979)										○	"	
CYBER-203	CDC	商用(1979)										○	"	
CYBER-205	"	商用(1980)										○	"	
S-1	スタンフォード大	実験											"	
データフロー計算機														
MIT DFM	MIT (Dennis)	研究(1974) 実験(1980)							○					
RAUシステム/1	シールーズ大	実験(1976)												
ID DFM	UCI (Arvind)	研究(1978)											(科学技術計算)	○
GE DFM	Rumbaugh	研究(1977)												
DDM1	コタ大 (Davis)	実験(1978)						○						
TOPSTAR	東 大	実験(1978)												
リダクションマシン														
Berkling マシン	GMD (Berkling)	研究(1976) 実験(1978)							○					
Keller マシン	コタ大 (Keller)	研究(1978)												
Mago マシン	ノースカロライナ大	研究(1979)												
Cambridge マシン	Cambridge 大	実験(1980)												

	開発者	開発水準	適応型アーキテクチャ						特殊化/専用化アーキテクチャ						
			マイクロ プロ ラム 適 応 型	再構成的適応型			動的適応型		機能分散型			問題指向型 科学技術計算/ Navier Stokes 信号処理/ 画像処理/ 音声処理	アルゴリズム指向型		
				プロセッサ アレイ 分割	プロセッサ メモリ 直結	パイプ ライン 再構成	結 合 トポロ ジイ	要 求 駆動型	デ ー タ 駆動型	組 込 み レ ベ ル	サ ブ シ ス テ ム レ ベ ル		シ ス テ ム レ ベ ル	Systolic Algorithm 型	SIMD/ MIMD Algorithm 型
データベース・マシン															
RAP. 1/2	トロント大	実験(1976/7)										○			
DBC	オハイオ州立大	研究(1976)										○			
SDD-1	OC of America	実験(1978)										○			
(STARAN型)	シラキューズ大	実験(1979)	○									○			
EDC	電総研	実験(1979)										○			
ADABASマシン	Software AG	商用(1980)										○			
TRAC	テキサス大(Dale)	研究(1979)										○			
UHP (ユニバーサル・ホスト・プロセッサ)															
QM-1	Nanodata	商用(1972)	○												
B1700/1800	パロース	商用(1972)	○												
F CPU	Data Saab	商用(1972)	○												
Cons	MIT (Greenblatt)	研究(1974) 実験(1977)	○												
CDC-5600	CDC		○												
ACE	電総研	実験	○		○										
IBM5100	IBM	商用(1977)	○												
高級言語マシン															
SYMBOL	フェアチャイルド	実験(1971)										○			
APL Assist	IBM	商用(1976)								○					
COMBAT	日電	実験(1976)										○			
FLATS	理研	研究(1978)										○			
LISPマイクロプロセッサ	MIT (Sussman)	実験(1980)										○			
その他															
C.mmp	CMU	実験(1972)			○										
Cm*	CMU	実験(1976)			○										

数式処理

	開発者	開発水準	高水準化アーキテクチャ					新言語(新計算モデル)指向アーキテクチャ								
			高級言語指向型			OS機能抽象化型		関数型言語向			抽象データ型言語向		述語論理型言語向		関係代数型言語向	
			高機能 命令 追加方式	高級言語マシン方式		コマンド 指向型	対象 指向型	FPFSy- stem系 リダク ション	λ-Calculus系		対象 指向的 アプローチ	構造デー タ処理 アプローチ	リゾリュ ーション アプローチ	ナチュラル ディダクシ ョン アプローチ	Sort- Search アプローチ	連想 メモリ アプローチ
				中間 言語方式	直接 実行方式				LISP型 リダク ション	コンピネ ータ型リ ダクシ ョン						
(商用大型機)	(計算機メーカー)	商用	○			○										
スーパーコンピュータ																
ASC	Texas Inst.	実験(1972)														
ILLIAC IV	イリノイ大	実験(1973)														
STAR-100	CDC	商用(1973)														
CRAY-1	Cray Research	商用(1976)														
FACOM 230/75 APU	富士通	商用(1976)														
BSP	パロース	商用(1977)														
DAP	ICL	商用(1979)														
HITAC M-200H IAP	日立	商用(1979)														
CYBER-203	CDC	商用(1979)														
CYBER-205	"	商用(1980)														
S-1	スタンフォード大	実験														
データフロー計算機																
MIT DFM	MIT(Dennis)	研究(1974) 実験(1980)									○					
RAUシステム/1	ツールーズ大	実験(1976)									○					
ID DFM	UCI(Arvind)	研究(1978)									○					
GE DFM	Rumbaugh	研究(1977)														
DDM1	ユタ大(Davis)	実験(1978)														
TOPSTAR	東大	実験(1978)														
リダクションマシン																
Berkling マシン	GMD(Berkling)	研究(1976) 実験(1978)			○											
Keller マシン	ユタ大(Keller)	研究(1978)			○			○								
Mago マシン	ノースカロライナ大	研究(1979)			○			○								
Cambridge マシン	Cambridge 大	実験(1980)			○											





	開発者	開発水準	冗長構造アーキテクチャ			保護機構アーキテクチャ					ファイルソフト・アーキテクチャ		
			RT レベル	PMS レベル	システム レベル	暗号化方式		アクセス制御方式			並列処理アプローチ		仮想化 アプローチ
						公開キー型	秘密キー型	メモリ・キー型	アドレス バウンダリ型	ポインティング型	マスタ方式	マスタ・レス 方式	
データベース・マシン													
RAP. 1/2	トロント大	実用(1976/7)											
DBC	オハイオ州立大	研究(1976)											
SDD-1	CC of America	実験(1978)											
(STARAN型)	シラキュース大	実験(1979)											
EDC	電総研	実験(1979)											
ADABASマシン	Software AG	商用(1980)											
TRAC	テキサス大(Dale)	研究(1979)											
UHP(ユニバーサル・ホスト・マシン)													
QM-1	Nanodata	商用(1972)											
B1700/1800	パロース	商用(1972)											
FCPU	Data Saab	商用(1972)											
Cons	MIT(Greenblatt)	研究(1974) 実験(1977)											
CDC-5600	CDC												
ACE	電総研	実験											
IBM5100	IBM	商用(1977)											
高級言語・マシン													
SYMBOL	フェアチャイルド	実験(1971)											
APL Assist	IBM	商用(1975)											
COMBAT	日電	実験(1976)											
FLATS	理研	研究(1978)											
LISPマシン	MIT(Sussman)	実験(1980)											
その他													
C. mmp	CMU	実験(1972)											
Cm*	CMU	実験(1976)											

	開発者	開発水準	自動診断アーキテクチャ			
			重複構成 アプローチ	冗長ロード アプローチ	検証論的アプローチ	
					マイクロ診断 方式	サービス・ プロセッサ方式
(商用大型機)	(計算機メーカー)	商用	○	○		
スーパーコンピュータ						
ASC	Texas Inst.	実験(1972)				
ILLIAC IV	イリノイ大	実験(1973)				
STAR-100	CDC	商用(1973)				
CRAY-1	Cray Research	商用(1976)				○
FACOM 230/75 APU	富士通	商用(1976)				
BSP	パロース	商用(1977)				
DAP	ICL	商用(1979)				
HITAC M-200H IAP	日立	商用(1979)				
CYBER-203	CDC	商用(1979)				○
CYBER-205	"	商用(1980)				○
S-1	スタンフォード大					
データフロー計算機						
MIT DFM	MIT(Dennis)	研究(1974) 実験(1980)				
RAUシステム/1	ツールーズ大	実験(1976)				
ID DFM	UCI(Arvind)	研究(1978)				
GE DFM	Rumbaugh	研究(1977)				
DDM1	ユタ大(Davis)	実験(1978)				
TOPSTAR	東大	実験(1978)				
リダクションマシン						
Berkling マシン	GMD(Berkling)	研究(1976) 実験(1978)				
Keller マシン	ユタ大(Keller)	研究(1978)				
Mago マシン	ノースカロライナ大	研究(1979)				
Cambridge マシン	Cambridge 大	実験(1980)				

	開発者	開発水準	自動診断アーキテクチャ			
			重複構成 アプローチ	冗長コード アプローチ	検証論的アプローチ	
					マイクロ診断 方式	サービス・ プロセッサ方式
データベースマシン						
RAP. 1/2	トロント大	実験(1976/7)				
DBC	オハイオ州立大	商用(1976)				
SDD-1	CC of America	実験(1978)				
(STARAN型)	シラキュース大	実験(1979)				
EDC	電総研	実験(1979)				
ADABASマシン	Software AG	商用(1980)				
TRAC	テキサス大(Dale)	研究(1979)				
UHP(ユニバーサル;ホスト・プロセッサ)						
QM-1	Nanodata	商用(1972)				
BI700/1800	パロース	商用(1972)				
PCPU	Data Saab	商用(1972)				
Cons	MIT (Greenblatt)	研究(1974) 実験(1977)				
CDC-5600	CDC					
ACE	電総研	実験				
IBM5100	IBM	商用(1977)				
高級言語マシン						
SYMBOL	フェアチャイルド	実験(1971)				
APL Assist	IBM	商用(1976)				
COMBAT	日電	実験(1976)				
FLATS	理研	研究(1978)				
LISPマイクロプロセッサ	MIT (Sussman)	実験(1980)				
その他						
C. mmp	CMU	実験(1972)				
Cm*	CMU	実験(1976)				

	開発者	開発水準	高度普及化アーキテクチャ		人工知能指向アーキテクチャ				
			パーソナル・コンピュータ向		組込み コンピュータ向	連想プロセッサ アプローチ	LISP アプローチ	PROLOG アプローチ	Combinatory Logic アプローチ
			汎用 アプローチ	高級言語 指向 アプローチ					
(商用大型機)	(計算機メーカー)	商用							
スーパーコンピュータ									
ASC	Texas Inst.	実験(1972)							
ILLIAC IV	イリノイ大	実験(1973)							
STAR-100	CDC	商用(1973)							
CRAY-1	Cray Research	商用(1976)							
FACOM 230/75 APU	富士通	商用(1976)							
BSP	パロース	商用(1977)							
DAP	ICL	商用(1979)							
HITAC M-200H IAP	日立	商用(1979)							
CYBER-203	CDC	商用(1979)							
CYBER-205	"	商用(1980)							
S-1	スタンフォード大								
データフロー計算機									
MIT DFM	MIT (Dennis)	研究(1974) 実験(1980)							
RAUシステム/1	ツールーズ大	実験(1976)							
ID DFM	UCI (Arvind)	研究(1978)							
GE DFM	Rumbaugh	研究(1977)							
DDM1	コタ大 (Davis)	実験(1978)							
TOPSTAR	東大	実験(1978)					○	○	
リダクションマシン									
Berkling マシン	GMD (Berkling)	研究(1976) 実験(1978)							○
Keller マシン	コタ大 (Keller)	研究(1978)						○	
Mago マシン	ノースカロライナ大	研究(1979)						○	
Cambridge マシン	Cambridge	実験(1980)		○					○

	開発者	開発水準	高度普及化アーキテクチャ		人工知能指向アーキテクチャ				
			パーソナル・コンピュータ向		組込み コンピュータ向	連想プロセッサ アプローチ	LISP アプローチ	PROLOG アプローチ	Combinatory Logic アプローチ
			汎用 アプローチ	高級言語 指向 アプローチ					
データベース・マシン RAP, 1/2 DBC SDD-1 (STARAN型) EDC ADABASマシン TRAC	トロント大 オハイオ州立大 CC of America シラキューズ大 電総研 Software AG テキサス大(Dale)	実験(1976/7) 研究(1976) 実験(1978) 実験(1979) 実験(1979) 商用(1980) 研究(1979)							
UHP (ユニバーサル・ホスト・マシン) QM-1 B1700/1800 FCPU Cons CDC-5600 ACE IBMS100	Nanodata パロース Data Saab MIT(Greenblatt) CDC 電総研 IBM	商用(1972) 商用(1972) 商用(1972) 研究(1974) 実験(1977) 実験 商用(1977)							
高級言語マシン SYMBOL APL Assist COMBAT FLATS LISPマイクロプロセッサ	フェアチャイルド IBM 日電 理研 MIT(Sussman)	実験(1971) 商用(1976) 実験(1976) 研究(1978) 実験(1980)							
その他 C. mmp Cm*	CMU CMU	実験(1972) 実験(1976)							

## 6. 外資系汎用コンピュータの動向

最近のIBMに代表される汎用コンピュータの動向として、アーキテクチャ的にみて全く新しいものは少ない。IBMにおいては、システム/38においては新しい動きがみられた。Eシリーズ(4300シリーズ)、Hシリーズ(3081プロセッサ)においては、革新的な動きはしていないものの漸進的な拡張がみられる。これらのシリーズにおいては既存370マシンとの互換をとることを第一義とし、370マシンの不備な点、ネックとなっていた点を着実に機能拡張、高性能化を行なっている。

### ○使い易さの向上

- ・大型機におけるアドレッシング能力の拡張
- ・エントリレベルシステムに代表されるパッケージ化プログラム

### ○機能分散による高性能化

- ・ダイアディック構成
- ・DCSによるI/O処理の独立化
- ・3880ディスクキャッシュによるスループットの増加

### ○高信頼性化

- ・ダイアディック構成
- ・書き戻しアレイによるチェックポイント再試行
- ・DASによるプロテクション
- ・セグメントプロテクション
- ・チャンネルセットスイッチング

### ○拡張性

- ・フィールドアップグレイダブル

### ○省エネ/省スペース化

- ・テクノロジーの進歩
- ・実装技術の向上

今後の汎用コンピュータの動向としては、より使い易いシステム、より広汎な利用形態を目指して着実に歩むものと思われる。以下に、外資系汎用コンピュータの出荷時期・性能比較、IBMの機種・アーキテクチャの動向・将来予測等を中心に述べる。

# 6.1 外資系汎用コンピュータの動き

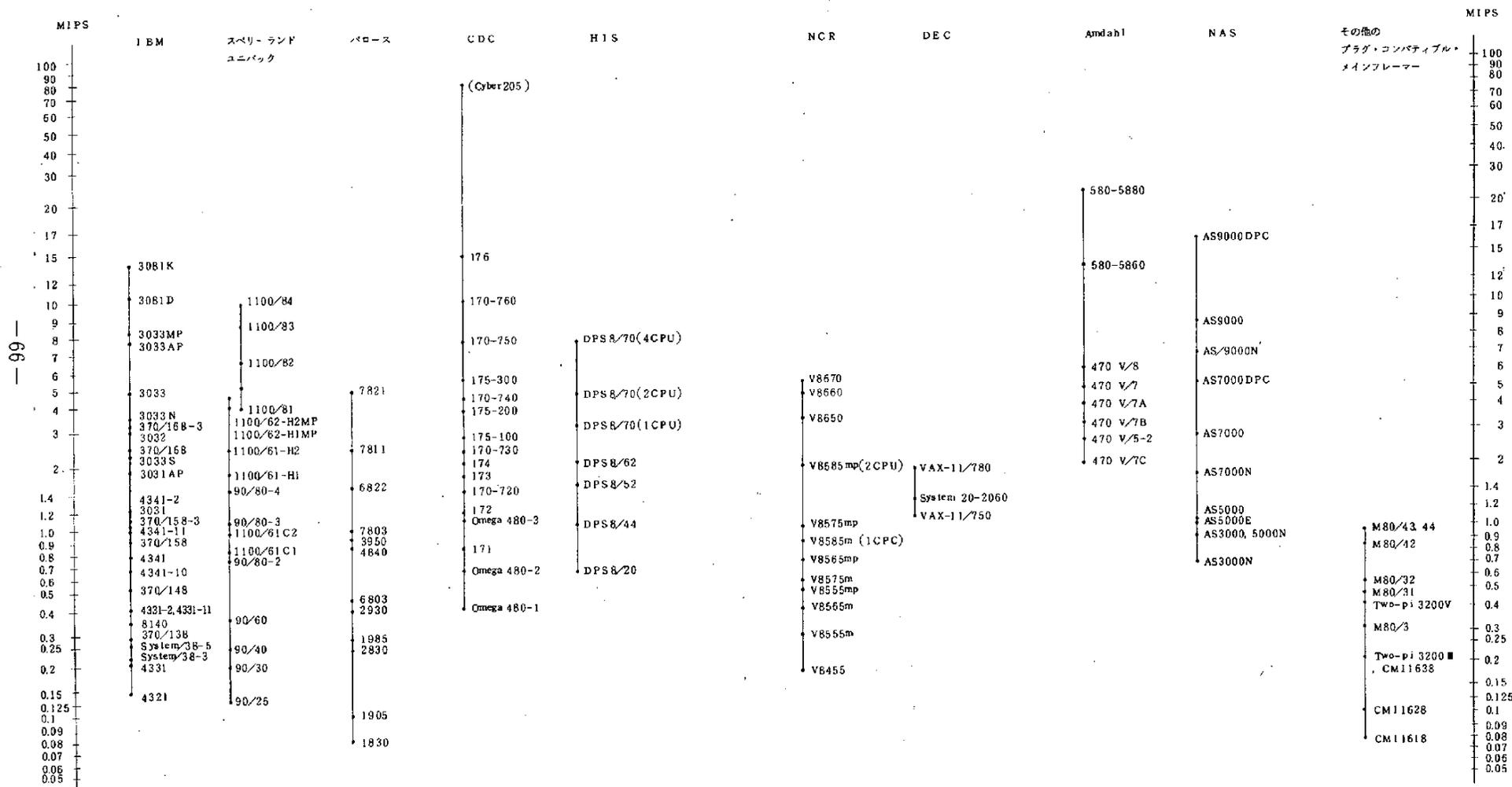
## 表6-1 外資系汎用コンピュータの出荷時期と発展経緯

出典：「ソフトウェア開発の効率化技術を軸に新たな飛躍を迎えるコンピュータ」日経エレクトロニクス1980.1.7 p.130-184  
 ・「機能強化が著しい仮想計算機システム」日経エレクトロニクス1980.5.2 p.76-99.  
 ・J ECCコンピュータ・ノート1982年版

	1967	1968	1969	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	
I B M	IBM S/360 95 IBM 1401H	IBM S/360 25	IBM S/360 85	IBM S/3-6 S/3-10	IBM S/370 145, 155 185 S/360 22, 195	IBM S/370 135	IBM S/370 125, 158 168, 195	IBM S/3-15 S/370-115	IBM S/32 S/3-8	IBM S/3-12 S/3-4 S/370-138 S/370-168APS Series 1	IBM S/370-148 S/370-158APS S/34	IBM 3033, 3032, 3031 3033 MF 3031 AP 813Q, 814Q S/38	IBM4341 IBM4331 IBM4033AP IBM5033J発表	IBM 3081 3035S 4341-2 4341-2 4341-2 5/38 5120発表	3081 K 4321, 4331-11 4341-10, 4341-11 発表	
スペリールランド ユニパック	U9200 U9300		U9400 U414 ■ U1106			U1100 U9700	U90/70	U90/60	U90/30 U1100/20 U1100/40	U1100/10	U1100/81.82 U90/80 BC/7 U90/25	U90/80-4発表 U90/40 U1100/80B 83, 84, 86	V77D1P発表 1100, 50B 61 62発表	1100/62H1MF, H2MP発表 システム80発表 1100/61E1, 61E2発表 1100/61AVP発表 VS/9発表	U77-500, 700 1100/63, 64 発表	
バロース	B2500 B3500	B500	B6500 B8500	B5700	B4700 B4500 B6700 Lシャーズ	B3700 B2700 B1700 B7700	B705, 711 B1728 B1726 B7700 B1776	B7750 B4790 B1710	B80/40, 30 B6807 B6811 B6821 B4800	B2800, 3800 B600 B1830 1850 1870 B80/50		B6617 B6606, 6812, 6818, 6822 B2805, 2835, 3605, 3645, 4800 B1825, 1835, 1865 B80-20, 60 B7800	B2830, 2650 発表 B1815, 1855, 1885 発表 B1906 1955 発表 B91.92	B6830 発表 B5930 発表 B920 発表 CP9452, 9472, 9552 発表 CP9558, 9572 発表	B7830, 7850 発表 B3955	
C D C	CDC6500 6800		CDC3500 7600	CDC6700 3170 6200	CDC Cyber 72, 73, 74, 76		Star100	Cyber 172, 173 174, 175	Cyber71	Cyber171 Omega480-1	Cyber170-100, 200, 300 発表 Omega480-2	Omega480-3 Cyber170-700 発表 730 750, 760 Cyber1203	Cyber205 発表 Cyber170/740 発表			
H I S	GE225 420- H 125	GE405 H4200 3200 110	GE105 130 245, 275 410, 450 440, 615	GE120 H1S 発表 H3200 115	H105 1015 2015 115-2 6030 6050 6070	H6040 6060 6080 2040 2050 2060 2070	H2020 H2030 G6025	H61/58, 60 H62/40, 60 H64/20, 40 H66/20, 40 60, 80 H68/80	H66/70	H64/30 H66/05	H66/07, 17, 27 H64/50, 60 H63/DPS 発表	H66/DPS 発表 H66DPS- 440, 520 発表 H64DPS- 320, 350 発表 H62/15, 25, 35 発表 DP58/20, 8/44, 8/52 8/70 発表	H66DPS- 700 発表 DP58/700 440, 520 発表 DP58/440 62 発表 DP56/92, 96 発表 H64DPS-330 発表 DP58/20 DP54 発表	DPS-6/48, 54, 74, 76, 30, 3L 発表 32, 34, 38 発表 DPS-8/20C, 44C 52C, 62C		
N C R		Century 100	Century 200		Century 50	Century 101 300	Century 251	Century 201	Century 151 Century 8200	NCR499 Criterion 8500 Criterion 8570 Century 75	18250 N8350, 18230 18150, N8450 N8560 V8590 V8590 発表 V8590, 8670 発表	18130 N8370 発表 18430, V8580 V8590 V8590, 8670 発表	B585M, V8575M, V8565M, V8555M, V8550, V8670 発表 1-8250, 8270 発表	1-8140 発表 NCR7550 発表	1-9010, 9020, 9040, 9050 発表	
D E C	DEC1040 1050 1055				DEC System 10 発表	DEC 1060 1070 1077			DEC 1080 1088 1090	DEC2040 (DEC system 20)	DEC2050 PDP-11/60 DEC2030	DEC2060 VAX-11/780 DEC2030	DEC2060 PDP11/23	VAX-11/750 発表 Datasystem 315, 750, 780 発表	PDP11/24 発表	
ブラダ・コンパティブル・メインフレーム									Amdahl 470V/6 (発表)		Amdahl 470V/5 470V/6-II 470V/7 (発表)	Amdahl 470V/5-II 470V/8 (発表) NCSS 3206 3216 3232 Magnason M80/3 Multiprocessors 30/3, 4 Nanodata VAX-200, 400 Cambex 1618, 1628 1638, 1640	Amdahl 470V/7A Magnason M80/4 M80/32 M80/42 M80/43	Amdahl 580-5860 580-5890 470V/7C Caribex 1636 1641 1651 NAS AS/3000N AS/3000 AS/5000N AS/5000E AS/5000 AS/7000N AS/7000 AS/7000DPC AS/9000 Magnason MR0/31 Nanodata QMX 6333 6336 6343	Amdahl 5870 NAS AS/9000 DPC Magnason M80/30 M80/30E M80/41	
主要な技術上の発展経緯	実験的仮想計算機 ベーシング方式 (C/P 67)				仮想計算機の理論的体系化、実用化 (VM/370)				基中処理から分散処理へ							
	実験的仮想記憶 (B5000, MULTICS) (TSS/360)				ダイナミック・マイクロプログラミング方式 (B1700のSマシン)				(NCR 8500の仮想マシン)							
	実験的マルチ・プロセッサシステム (LARC, B5000)				高用システムへの仮想記憶適用、一般化 (VS1, VS2)				(MVS)				(S/38)			
	パイプライン方式 (S/360-91)				パイプライン+キャッシュ (S/360-195)				高圧なパイプライン (Star 100, CRAY-1)				機能拡張、高速化 (3033 MP)			
	キャッシュの採用 (S/360-85)												(3081 (DF)) (Cyber 205)			

図 6 - 1 外資系汎用コンピュータの性能比較

出典：・Europa Report Vol.11, No.7, April 18, 1981.  
 IDC Europa, Ltd.  
 ・DATAMATION, Nov.1980  
 ・COMPUTERWORLD, July 13, 1981



(注) 各社とも最近の機種を中心に収録した。



表 6 - 3 中大型機の動向

出典：A.PADEGS : System/360 and Beyond, IBM J. RES. DEVELOP., Vol.25, No.5, Sept.1981, P. 387-389

Announcement and shipment dates.

Model characteristics.

Model	Announcement and shipment dates		Model characteristics														
	Announced	First shipped (bytes)	CPU		Control storage				Number of TLB entries	Processor storage			Cache				
			Data-flow width	Cycle time (ns)	Size (K words)	Word size (bits)	Type	Cycle time (ns)		Size (K bytes)	Bus width (bytes)	Cycle time (ns)	Size (K bytes)	Line width (bytes)	Cycle time (ns)	Type	Associativity
<i>System/360</i>																	
22	71-4-7	71-6	1	750	4	50+5	RO	750	none	24-32	1	1500 <sup>1</sup>	none				
25	68-1-3	68-10	1	900	8	16+2	RW	900	none	16-48	2	900 <sup>1</sup>	none				
30	64-4-7	65-6	1	750	4	50+5	RO	750	none	16-64	1	1500 <sup>1</sup>	none				
40	64-4-7	65-4	2 <sup>4</sup>	625	4	52+2	RO	625	none	32-256	2	2500 <sup>1</sup>	none				
44	65-8-16	66-9	4	250	none				none	32-256	4	1000 <sup>1</sup>	none				
50	64-4-7	65-8	4	500	2.75	85+3 <sup>2</sup>	RO	500	none	128-512	4	2000 <sup>1</sup>	none				
										1024-8192	4×(1-2)	8000 <sup>1</sup>					
60	64-4-7	not shipped <sup>1</sup>															
62	64-4-7	not shipped <sup>1</sup>															
65	65-4-22	65-11	8	200	2.75	87+4 <sup>4</sup>	RO	200	none	256-1024	8×2	750 <sup>1</sup>	none				
										1024-8192	8×(1-2)	8000 <sup>1</sup>					
67	65-8-16	66-5	8	200	2.75	87+4 <sup>4</sup>	RO	200	8	256-1024	8×2	750 <sup>1</sup>	none				
70	64-4-7	not shipped <sup>1</sup>															
75	65-4-22	66-1	8	195	none				none	256-1024	8×(2-4)	750 <sup>1</sup>	none				
										1024-8192	8×(1-2)	8000 <sup>1</sup>					
85	68-1-30	69-12	8	80	2	105+3 <sup>3</sup>	RO	80	none	512-4096	16×(2-4)	960 <sup>1</sup>	16-32	16×4 <sup>4</sup>	80-160	T	16
					0.5	105+3 <sup>1</sup>	RW	80									
91	64-11-17	67-10	8	60	none				none	2048-6144	8×16	780 <sup>1</sup>	none				
92	64-8-17	not shipped <sup>2</sup>															
95	"	68-2	8	60	none				none	1024	8×16	180	none				
										1024-6144	8×16	280 <sup>1</sup>					
										1024-4096	8×(8-16)	756 <sup>1</sup>	32	8×8	54-162	T	4
<i>System/370</i>																	
115	73-3-13	74-3	1	480	20-28	20+3	RW	480	8	64-192	2	480	none				
115-2	75-11-10	76-4	2	480	12-20 <sup>7</sup>	19+2	RW	480	16	64-384	2	480	none				
125	72-10-4	73-4	2	480	12-20	19+2	RW	480	16	96-256	2	480	none				
125-2	75-11-10	76-2	2	320-480 <sup>8</sup>	16-24	19+2	RW	320	16	96-512	2	480	none				
135	71-3-8	72-4	2	275-1485 <sup>9</sup>	12-24	16+2	RW	275	8	96-512	2	990 <sup>1</sup> R	none				
												935 <sup>8</sup> W					
135-3	76-6-30	77-2	2	275-1485 <sup>9</sup>	64	16+2	RW	275	8	256-512	2	990 <sup>8</sup> R	none				
												935 <sup>8</sup> W					
138	76-6-30	76-11	2	275-1430 <sup>8</sup>	64	16+2	RW	275	8	512-1024	2	935 <sup>8</sup>	none				
145	70-9-23	71-6	4 <sup>10</sup>	203-315 <sup>8</sup>	8-16 <sup>11</sup>	32+4	RW	203	8	160-2048	8	540 R	none				
												608 W					
145-3	76-6-30	77-5	4 <sup>10</sup>	180-270 <sup>8</sup>	32	32+4	RW	180	8	192-1984	8	405 R	none				
												540 W					
148	76-6-30	77-1	4 <sup>10</sup>	180-270 <sup>8</sup>	32	32+4	RW	180	8	1024-2048	8	405 R	none				
												540 W					

155	70-6-30	71-1	4	115	6	69+3	RO	115	none	256-2048	8	2070 <sup>1</sup>	8	16	115-230	T	2
155-II			4	115	8	69+3	RO	115	128	256-2048	8	2070 <sup>1</sup>	8	16	115-230	T	2
158	72-8-2	73-4	4	115	8	69+3	RW	115	128	512-6144	8	1035 R 920 W	8	16	115-230	T	2
158-3	75-3-25	76-9	4	115	8	69+3	RW	115	128	512-6144	8	920	16	16x2	115-230	T	4
165	70-6-30	71-4	8	80	2	105+3	RO	80	none	512-3072	8x4	2000 <sup>1</sup>	8-16	8x4	80-160	T	4
					2	105+3 <sup>1</sup>	RW	80									
165-II			8	80	4	105+3 <sup>1</sup>	RO	80	128	512-3072	8x4	2000 <sup>1</sup>	8-16	8x4	80-160	T	4
					1	105+3 <sup>1</sup>	RW	80									
168	72-8-2	73-5	8	80	2-3.5	105+3 <sup>1</sup>	RO	80	128	1024-8192	8x4	320	8-16	8x4	80-160	T	4-8 <sup>11</sup>
					0.5-1	105+3 <sup>1</sup>	RW	80									
168-3	75-3-25	76-6	8	80	2-3.5	105+3 <sup>1</sup>	RO	80	128	1024-8192	8x4	320	32	8x4	80-160	T	8
					1-2	105+3 <sup>1</sup>	RW	80									
195 <sup>12</sup>	71-6-24	73-8	8	54	none				none	1024-4096	8x16	756	32	8x8	54-162	T	4
<i>System 1370-compatible</i>																	
3031	77-10-6	78-3	4	115	8	69+3	RW	115	128	2048-8192	8x4	345 <sup>14</sup>	32	8x4	115-230	T	8
3032	77-10-6	78-3	8	80	4	105+3	RW	80	128	2048-8192	8x4	320	32	8x4	80-160	T	8
3033	77-3-25	78-3	8	57	3-7	105+3	RW	57	128	4096-25576	8x8	285 <sup>14</sup>	64	8x8	57-114	T	16
					1	122+4	RW	57									
3033-N	79-11-1	80-1	8	57	3-7	105+3	RW	57	128	4096-16384	8x4	285	16	8x8	57-114	T	8
					1	122+4	RW	57									
3033-S	80-11-12	81-L	8	57	3-7	105+3	RW	57	128	4096-16384	8x4	285	0.5	8x4	57-114	T	8
					1	122+4	RW	57									
1081 <sup>15</sup> D	80-11-12		8	26	2 <sup>16</sup>	104+4	RW	52 <sup>17</sup>	128	16384-32768	8x2 <sup>18</sup>	312 <sup>18</sup>	32	8x16	26-52	C	4
3081-K	81-10-21		8	26	2 <sup>16</sup>	104+4	RW			16384-32768		312 <sup>18</sup>	64		26	C	
4321	81-11-18									1024			none				
4331-1	79-1-30	79-3	4	300-1600 <sup>8</sup>	16-32 <sup>11</sup>	32+4	RW	500 <sup>19</sup>	64	512-1024	4	900 R 1300 W	none				
4331-11	81-11-18									1024-2048			none				
4331-2	80-5-7	80-8	4	200-1600 <sup>8</sup>	32 <sup>11</sup>	32+4	RW	500 <sup>19</sup>	64	1024-4096	4	2600 <sup>21</sup> R 3100 <sup>21</sup> W	8	4x16	200	C	4
					3	32+4	RO	100									
4341-10	81-11-18									2048-4096			4				
4341-1	79-1-30	79-11	8	150-300 <sup>8</sup>	14-16	32+4	RW	150	64	2048-4096	8	2400 <sup>21</sup>	8	8x8	225	C	4
4341-11	81-11-18									2048-8192			8				
4341-2	80-9-15		8	120-240 <sup>8</sup>	16-20	32+4	RW	120	64	2048-16384	16	1440 <sup>21</sup>	16	16x4	120 R 180 W	C	8

<sup>1</sup>Replaced by Model 65

<sup>2</sup>Replaced by Model 75

<sup>3</sup>Redesignated as Model 91

<sup>4</sup>Offered on special government contract

*Explanation*

- C Store-in-cache: On storing, the value is placed in the cache; the new value is placed in main storage at the time the cache line is reassigned or the data is requested by a channel or another processor.
- K The number  $2^{10} = 1024$ .
- ns Nanoseconds
- R Access for reading
- RO Read-only
- RW Read-write (writable)
- T Store-through: On storing, the value is placed in main storage; the value is not placed in the cache unless a line has been assigned to the main-storage location.
- TLB Translation-lookaside buffer, which is a part of the dynamic-address-translation mechanism
- W Access for writing

*Footnotes*

<sup>1</sup>The model uses magnetic-core technology.

<sup>2</sup>Certain registers and paths are 17 or 18 bits wide where a main-storage address is processed in one cycle.

<sup>3</sup>Extended to  $90 + 3$  for the 1410 emulator, or  $92 + 3$  for the 7070 emulator.

<sup>4</sup>Extended to  $94 + 4$  when any emulator is installed.

<sup>5</sup>Extended to  $122 + 4$  for part of control storage when any emulator is installed.

<sup>6</sup>Although the 64-byte lines are loaded into the cache only when referred to, an entire cache sector of 1K bytes (16 lines) is assigned as a unit to a 1K-byte storage sector.

<sup>7</sup>The 113-2 contains a separate I/O processing unit for some functions that were executed on the CPU in a 113; hence the smaller CPU control-storage capacity.

<sup>8</sup>Variable, depending on the type of operation performed.

<sup>9</sup>Four bytes can be accessed and transferred in this time.

<sup>10</sup>An 8-byte-wide path is used for instruction fetch.

<sup>11</sup>Part of this capacity is physically in processor storage and thus has to be subtracted from the available processor-storage capacity.

<sup>12</sup>Depends on cache size used.

<sup>13</sup>The System/370 Model 195 has certain facilities (e.g., time-of-day clock, control registers, MOVE LONG) not available on the System/360 Model 195.

<sup>14</sup>The effective transfer rate to the CPU is limited to eight bytes per CPU cycle.

<sup>15</sup>Each of the two CPUs has the indicated control-storage, cache, and TLB capacity.

<sup>16</sup>1K of the control storage is pageable, using an area in processor storage assigned for this purpose.

<sup>17</sup>26 ns when the word is available in the microinstruction buffer (i.e., is within the current set of 16 words).

<sup>18</sup>Interleaving is on the basis of 2K bytes; no interleaving takes place within the access for a cache line of 128 bytes.

<sup>19</sup>An amount equal to a cache line is read or written in one storage cycle. The effective transfer rate to the CPU is limited to eight bytes per CPU cycle.

<sup>20</sup>100 ns when the word is available in the microinstruction buffer.

<sup>21</sup>An entire cache line can be accessed and transferred between the cache and the storage unit in this time.

図6-2 IBM機種のパフォーマンス、最大主記憶容量、価格性能比の推移

出典：  
 ・「ソフトウェア開発の効率化技術を軸に新たな飛躍を迎えるコンピュータ」日経エレクトロニクス1980.1.7 P.130~184  
 ・「大型コンピュータIBM 3081の内部構造を探る」日経エレクトロニクス、1980.6.21 P.172~173  
 ・「IBMが最上位機種3081Kと370拡張アーキテクチャを発表」日経エレクトロニクス、1981.11.23 P.84~86  
 ・コンピュータピア 1980年10月号

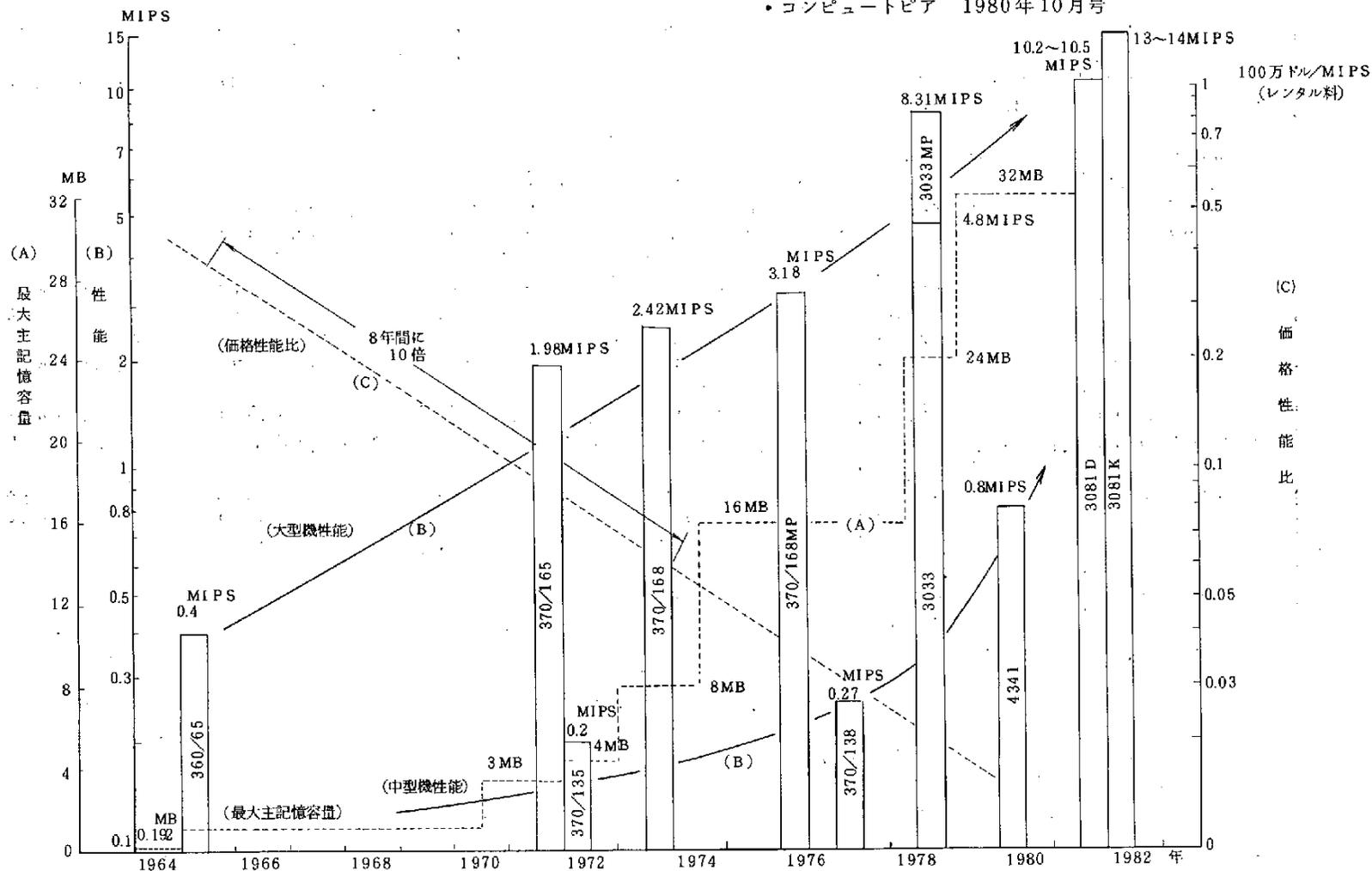


表 6 - 4 IBM 最上位機種の性能・機能の変化

出典：「大型コンピュータ IBM3081 の内部構造を探る」  
日経エレクトロニクス，1980.6.21 P.172~173

機種名	s/370 モデル165	s/370 モデル168	s/370 モデル168-3	3033 プロセサ	3081D プロセサ	3081K プロセサ
発表時期 (カッコ内は米国)	70年7月 (70年6月)	72年8月 (72年8月)	75年4月 (75年3月)	77年3月 (77年3月)	80年11月 (80年11月)	81年12月 (81年10月)
出荷時期 (カッコ内は米国)	71年7月 (71年4月)	73年8月 (73年8月)	75年6月 (75年後半)	78年3月 (78年4月)	81年第4四半期	82年第3四半期 (同第4四半期)
相対処理性能 *1	2.0	2.3	2.6	5.0	10.2~10.5 *2	13.0~14.0 *2
買い取りシステム価格 (円)	20億 (1MBシステム)	14.5億 (1MBシステム)	10億 (4MBのCPU)	9.8億 (4MBのCPU)	約9億 (16MBのCPU)	約11.8億 (32MBのCPU)
● CPU						
演算処理機構成	1	1	1	1	2	2
プロセサ・サイクル	80 ns	80 ns	80 ns	57 ns	26 ns	26 ns
使用テクノロジー	MST *3	MST *3	MST *3	MST *3	TCM *4	TCM *4
マイクロプログラム制御						
▷制御記憶タイプ	静電型ROS *5	半導体ROS *5, RCS *6	半導体ROS *5, RCS *6	半導体RCS *6	半導体ROS *5, RCS *7	半導体ROS *5, RCS *7
▷マイクロ命令長	108ビット	108ビット	108ビット	108/126ビット	108ビット	108ビット
高速バッファ・メモリー						
▷バッファ容量	8KB/16KB	8KB/16KB	32KB	64KB	32KB×2	64KB
▷構成 (行×列)	4×64/4×128	4×64/8×64	8×128	16×64	4×64 (×2)	
▷ブロック・サイズ	32バイト	32バイト	32バイト	64バイト	128バイト	
▷ストア方式	ストア・スルー	ストア・スルー	ストア・スルー	ストア・スルー	ストア・イン	
仮想記憶制御						
▷ページ・サイズ	—	2KB/4KB	2KB/4KB	2KB/4KB	2KB/4KB	2KB/4KB
▷セグメント・サイズ	—	64KB/1MB *8	64KB/1MB *8	64KB/1MB *8	64KB/1MB *8	64KB/1MB *8
▷TLBエントリ数	—	128 (2Kページ対象)	128 (2Kページ対象)	128 (4Kページ対象)	128 (4Kページ対象) × 2	128 (4Kページ対象) × 2
▷STOスタック	—	6	6	29	TLB中に含まれる	TLB中に含まれる
● 主記憶装置						
使用素子	フュライト・コア	1K, 2KスタチックMOS	2KスタチックMOS	4KダイナミックMOS	16KダイナミックMOS	16KダイナミックMOS
サイクル時間	2 μs/32バイト	320ns/8バイト	320ns/8バイト	285ns/8バイト	312ns/8バイト *9	312ns/8バイト *9
最小容量	512KB	1MB	1MB	4MB	16MB	16MB
最大容量	3MB	8MB	8MB	16MB *10	32MB	32MB
インタリーブ	4ウェイ (8バイト単位)	4ウェイ (8バイト単位)	4ウェイ (8バイト単位)	8ウェイ (8バイト単位)	2ウェイ (2KB単位)	2ウェイ (2KB単位)
● チャネル機能						
総チャネル数	最大7, オプションで12	最大7, オプションで12	最大7, オプションで12	標準12, オプションで16	標準16, オプションで24	標準16, オプションで24
マルチプレクサ	1台/2台 (~110KB/秒)	1台/2台 (~110KB/秒)	1台/2台 (~110KB/秒)	内蔵 (2台/3台) *11	内蔵 (最大4台) *12	内蔵 (最大4台) *12
セレクタ	最大6台	最大6台	最大6台	—	—	—
ブロック・マルチプレクサ	最大6, オプションで11	最大6, オプションで11	最大6, オプションで11	内蔵 (10台/14台) *13	内蔵 (最大16/24台) *14	内蔵 (最大16/24台) *14

\*1 IBM3031 プロセサの性能を1とした場合の相対性能比。\*2 推定。\*3 モノリシック・システム・テクノロジー。\*4 熱伝導モジュール。\*5 リード・オンリー・ストレージ。\*6 リローダブル・コントロール・ストレージ。  
\*7 マイクロコードのページング機能を備えた2K語のリローダブル・コントロール・ストレージ。\*8 現行のOSでは1MBセグメントは使用していない。\*9 最初の2バイト・アクセス時の時間。\*10 アドレス拡張機構による  
24MBモデルも発表されている (1980年11月)。\*11 40K~75KB/秒。\*12 45K~500KB/秒。\*13 ブロック・マルチプレクサの転送能力は1.5MB/秒。\*14 データ・ストリーム機構によりブロック・マルチプレ  
クサ・チャネルの転送能力は3MB/秒。

6.2.2 アーキテクチャの動き

70年代の初めにシステム370が発表されて以来、システム370アーキテクチャは受け継がれてきた。しかし、70年代の終わり近くに発表されたシステム/38、Eシリーズ(4300シリーズ)、80年に発表されたHシリーズ(3081)により徐々にではあるがその姿を変えつつある。図6-3にIBMアーキテクチャの変遷、図6-4にIBMアーキテクチャの関連図を示す。

図6-3 IBMアーキテクチャの変遷

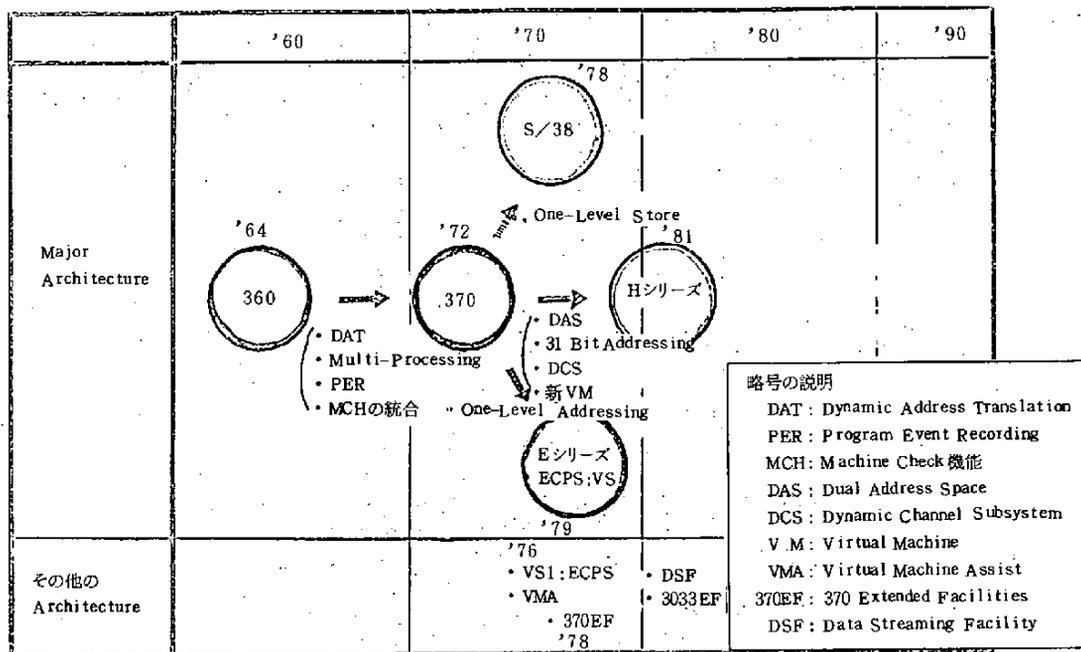
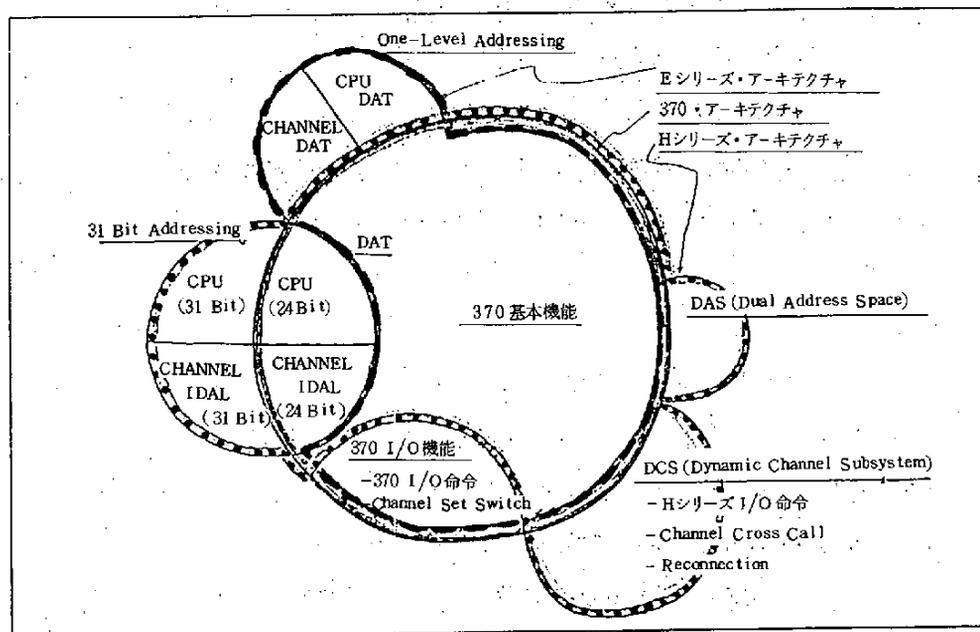


図6-4 IBMアーキテクチャの関連図



① DAT (Dynamic Address Translation)

370アーキテクチャの目的の第一は360アーキテクチャとの互換性をとることにあつたが、さらに使い易さの向上を目指した。その一つがDAT (CPUにおける仮想アドレスから実アドレスへのアドレス変換機構)を用いた仮想記憶の導入である。これによりVM (Virtual Machine)のサポートも行なわれた(360時のCP/67の経験)。

② Multi-Processing

処理性能の向上をはかるためマルチCPUによる処理がとり入れられた(360/65の経験)。

③ PER (Program Event Recording)

デバッグを容易にするため、指定された主記憶領域からの命令フェッチ事象、分岐成功事象、指定された主記憶域への書き込み事象、指定された汎用レジスタへの書き込み事象を割込みを利用して検出することができる。また、モニタリング機能 (Monitor Call)も追加された。

④ マシン・チェックの統合

モデルに依存しないマシン・チェック割込みとして統一された。

⑤ 単一レベル記憶

システム/38は、IBMのいわゆるFSシステムの成果であり、370アーキテクチャとは異なっている。Multicsの影響を強く受けており単一レベル記憶、リレーショナル型データベースなどの画期的なアーキテクチャを持つシステムである。

単一レベル記憶は、プログラムもデータもすべて一次元のアドレス域で管理するものである。システム/38では $2^{48}$ バイトまでアドレス可であり、ディスク装置は、現在最大約2600メガバイトまで拡張できるので、現在のシステム/38の最大アドレスは、2600メガバイトのアドレス空間にプログラムやデータを記録することができる。

システム/370の仮想記憶域管理と大きく異なる点は、データをも含めOSの取り扱うもの(これをシステム/38ではオブジェクトと呼ぶ)すべてがページングの対象となる。ページング作業はVAT (Virtual Address Translator: 仮想アドレス変換機構)というハードウェアにより行なわれる。(図6-5参照)

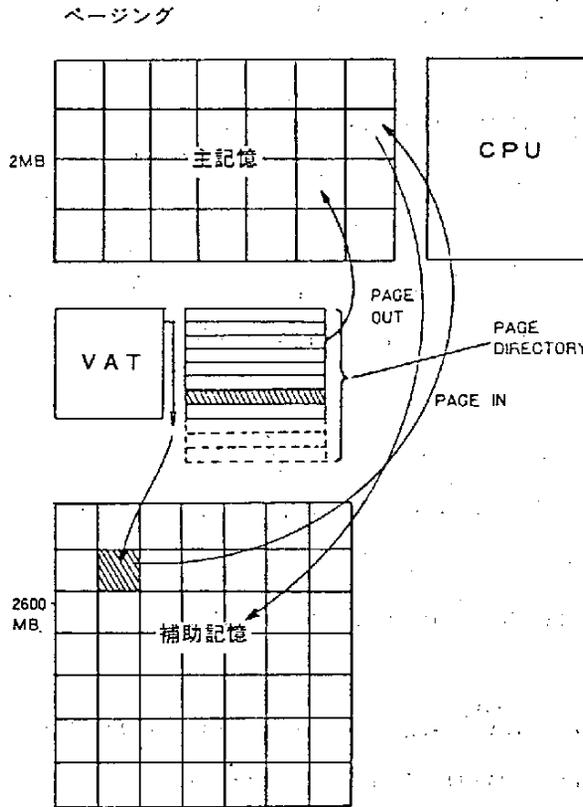
⑥ One-Level Addressing

Eシリーズ(4300シリーズ)では、モード(370モードとECPS:VSEモード)をIML (Initial micro-program loading)時に切り替えることにより370アーキテクチャとの互換性をとっている。Eシリーズの主モードのECPS:VSEモードの第1の特徴はone-level addressingの導入であり、これによりページングスーパーバイザのファームウェア化、チャンネルDATの実現によりOSのオーバヘッドの削減をはかっている。

4300アーキテクチャのone-level addressingでは、従来のTLBに相当する高速

図 6-5 システム/38 のページング

出典：川島徹：システム/38単一レベル記憶と  
リレーショナル型データベース，IBM  
REVIEW 82 1981, P. 101～110



変換バッファ（4300ではDLATと呼ばれる）と論理空間全体にわたるページ・テーブル（8Kエントリ。1ページは2Kバイト）が、CPU内部にもたれ、1レベル・ページング方式が実現されている。

4300シリーズは現在エントリレベルから中型機まで8機種発表されており、ソフトウェアのprepackage化をはじめ使い易さの向上をはかっている。

⑦ DAS (Dual Address Space)

370アーキテクチャの拡張として、2つの論理空間を使用しデータの転送やプログラム制御をオーバーヘッドを少なく行なう機能である。これによりKey Maskによる Authorization, Authority Table による Authorization 機能を用いて、サブシステム間のプロテクションが確実になる。

⑧ 31 bit addressing

3081Kの発表時システム/370拡張アーキテクチャ(370-XA)として、31ビットアドレッシング (Real/Virtual), Bimodal operation (24/32ビットアドレス)、DCSなどの機能強化が行なわれた。3081では、IMLで370モードか370-XAモー

ドを選択することによりシステム/370との互換性を保証している。370-XAモードにおいても、Bimodal operationにより24ビットで書かれたプログラムの実行が可能である。

⑨ DCS (Dynamic Channel Subsystem)

3081外部データ制御機構(EXDC)によりI/Oキューイング、I/O負荷バランシング、チャンネルパス決定のハード化、ディスコネクトコマンドチェーンの再結合処理など従来のスーパーバイザ機能の一部がCPUから独立した。これによりCPUのI/O処理オーバーヘッドの削減、総合I/Oスループットの向上がはかられている。

⑩ 新VM

3081ではPreferred Machine Assist for VM/SPとして、MVS/SPのV=Rに対してnativeに近いスピードをサポートする機構を発表している。

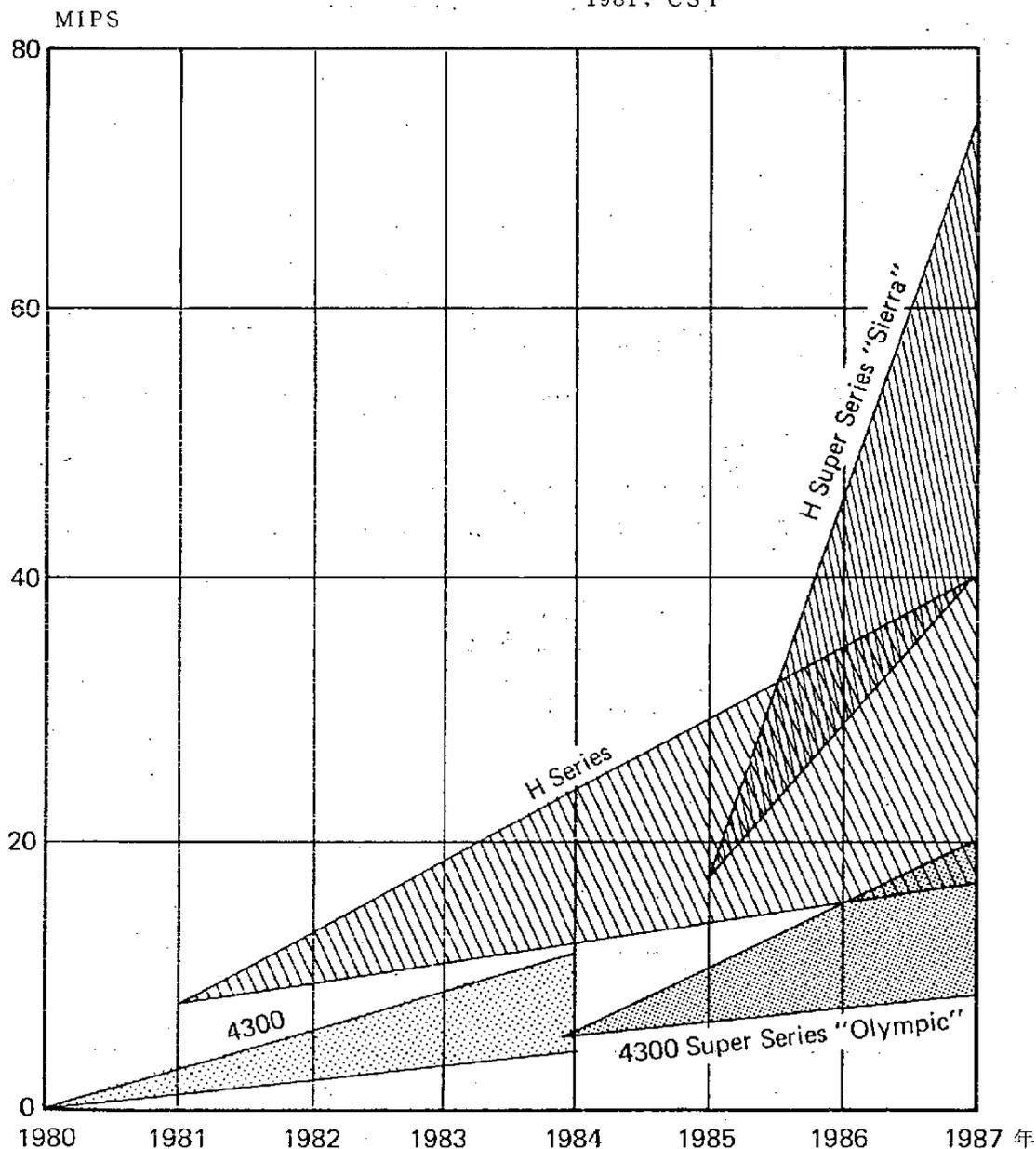
参考文献

- IBM: IBM 4300 Processors Principles of Operation for ECPS: VSE Mode, GA22-7070-C
- IBM: IBM System/370 Principles of Operation, 9th Ed., GA22-7000-8, Oct.1981
- IBM: IBM 3081, New Large Systems Model Group and Extended Architecture, Product Announcement, DPD, Oct. 21, 1981

6.2.3 将来予測

図6-6 IBM 4300シリーズとHシリーズの性能予測

出典: Plug-Compatible Mainframes, Feb. 1981, CSI



註 それぞれのメインフレーム・ファミリーの下線はスタンド・アロン・プロセッサのMIPS、また、上線はメインフレーム・プロセッシング・コンプレックスのMIPS累積を表示している。

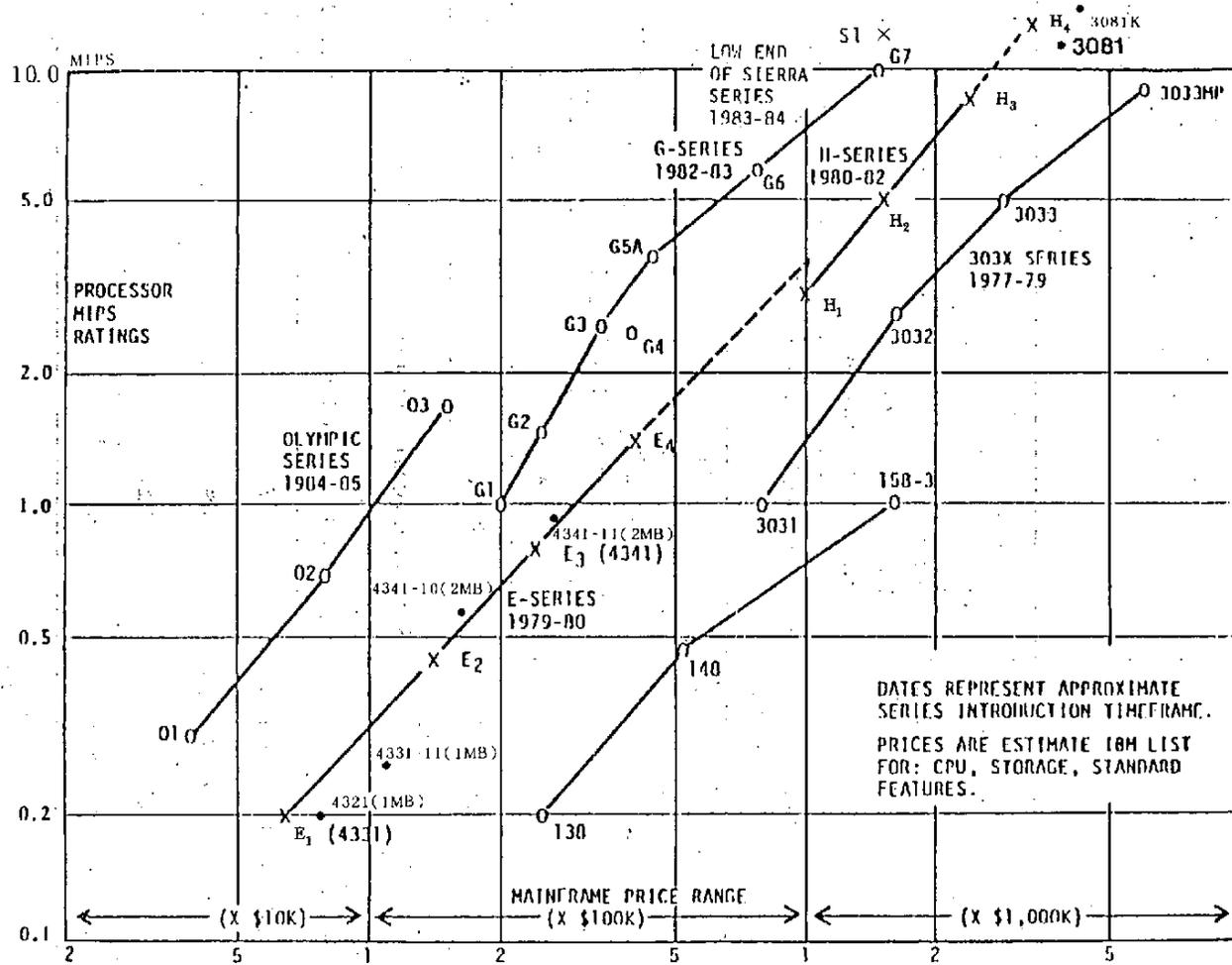
表 6 - 5 I B M の新製品発表予測

出典：・IBM's Implementation Strategies : Conflict and Perspectives, April 1980, Strategic Inc.  
 ・IBM 3081 Report, Dec.1980, ACT

Product	Description	Performance ( )内MIPSはACT社による	Expected Introduction Date [ -はStrategic Inc. ....はACTによる予測 ] *はSI社で1981年以前と予想されたもの			
			1982	1983	1984	1985
H Series (Strategic Inc., ACT予測)	Seven Mainframes Designed to Replace the 30XX Series	MIPS (H1) 3( 3.5) (H2) 5( 5.6) (H3) 9( 7.5) (H4)Dyadic 12(10.5) (H5)Dyadic 15(14.0) (H6)Quad 20(18.0) (H7)Quad 30(25.0)	*			
G Series (Strategic Inc.予測)	Five Mainframes Bridging E and H Series	1.5 3.0 6.0 9.0 Dyadic 12.0				
Sierra (Strategic Inc.予測)	Follow-on to H Series	10 20 30 40 50				

図 6 - 7 I B M 製品ラインの予測

- 出典：・「IBMが最上位機種 3081Kと 370 拡張アーキテクチャを発表」  
 日経エレクトロニクス，1981.11.23. P.84~86  
 ・IBM 3081 Report, Dec.1980, ACT  
 ・Wall Street Journal, Oct.22, 1981  
 ・「新しいプロセッサとプログラミングおよび新しいアーキテクチャを発表」日本アイ・ビー・エム記事資料，1981.12.3  
 ・日経エレクトロニクス 1981.6.22 P.172



### 6.3 IBM 3081 プロセッサ

#### システム / 370 拡張アーキテクチャ

表 6-6 拡張アーキテクチャと従来アーキテクチャの比較

出典：・IBM: IBM 3081, New Large Systems Model Group and Extended Architecture, Product Announcement, Oct. 21, 1981.  
 ・IBM: IBM 3081 Processor Storage Protect Key, Product Announcement, Oct. 21, 1981.  
 ・IBM: IBM System/370 Principles of Operation, 9th Ed., GA 22-7000-8, Oct. 1981.

項 目	拡張アーキテクチャ	従来アーキテクチャ
アドレッシング	<ul style="list-style-type: none"> <li>・ 31ビット(実アドレス/仮想アドレス)</li> <li>・ 2GBまでアクセス可能</li> <li>・ 24ビット・アドレッシングも可能 (Bimodal Operation)</li> </ul>	<ul style="list-style-type: none"> <li>・ 24ビット(実アドレス/仮想アドレス)</li> <li>・ 16MBまでアクセス可能</li> </ul>
仮想空間連絡機能 (クロス・メモリー・サービス)	DAS (Dual Address Space) Facility	無し*1)
セグメント・プロテクション	有り*2)	無し
記憶保護キー	4KB単位 (従来との互換のため、2KB単位も可能)	2KB単位又は4KB単位
チャンネル・セット スイッチング	標準	マルチプロセッサ構成のとき可能
ダイレクト制御	廃止	標準
サービス・プロセッサとの連絡機能	Service-Signal Facility	無し
再 試 行	チェック・ポイント再試行	命令再試行
I/Oリクエストのキューイング	可能*3)	不可能*1), *4)
H/Wによるチャンネルパス制御	可能	不可能
I/Oパスの短縮化	可能*5)	-
新しいI/O割り込み方式	有り*6)	-

\*1) 3033 拡張機構では有り。

\*2) セグメント・テーブル・エントリ中にセグメント・プロテクション・ビットを設けた。

\*3) H/WによるUCWキューイング

\*4) S/Wによる論理チャンネルキューイング

\*5) I/O命令に対する応答を待たずに命令を完了する。

\*6) 詳細不明

## 7. パーソナルマシン(高機能パーソナルコンピュータ)

### 1. パーソナルマシンとは、

ここでとりあげるパーソナルマシンは高機能単一ユーザー向けのコンピュータで、A D A、L I S P、F G K L等、高レベルのプログラム言語を取り扱うマシンをいい、次のような特徴をもっている。

その特徴とは、

- 1) 16ビット/32ビットの(マイクロ)プロセッサを使用
- 2) ドット・ベースで文字数の多いラスタグラフィック・ディスプレイを使用
- 3) 高速インタフェース機能
- 4) 大容量の主記憶装置と仮想メモリーの採用
- 5) 大容量補助メモリーの採用

を持ち、ホビースト・教育・一般事務に使用されるパーソナルコンピュータや科学計算、制御等に使われるミニコンと区別する。

### 2. 変遷

#### 1) 先駆者はゼロックス社(Xerox)

スタンフォード大学の影響を受けて、1972年ゼロックス社は社内にパーソナルマシンの開発プロジェクトをスタートさせた。

1973年には「ALTO」を発表するに到り、この分野の新しい動きが生まれた。「ALTO」は特定の言語向きというのではない汎用のマシンである。OA機器向けの用途を主にして「ALTO」を商品版に発展させたのが「STAR」である。

#### 2) MITの動き

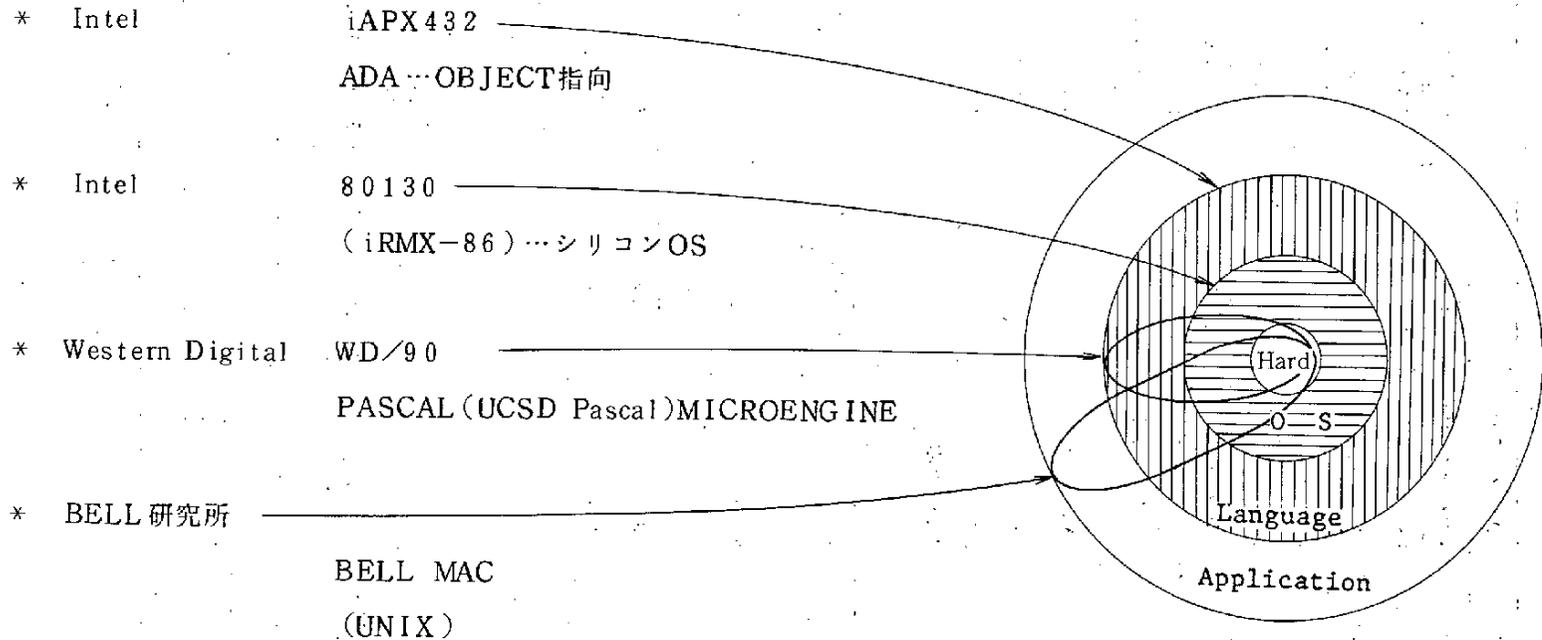
MITでは、L I S P言語向きパーソナルマシン「CADR」を開発した。このマシンは、シンボリック社(Symbolics社)やL I S Pマシン社に受け継がれてより高度なL I S Pベースのパーソナルマシン分野を開いた。

### 3. パーソナルマシンの現状と将来

ここでは高級言語を使用する高機能単一ユーザー向けのコンピュータをパーソナルマシンとしてとらえ、言語系、半導体系の両方にわたって現状の調査と将来の見通しを整理した。



図 7-2 ワン・チップ指向プロセッサ

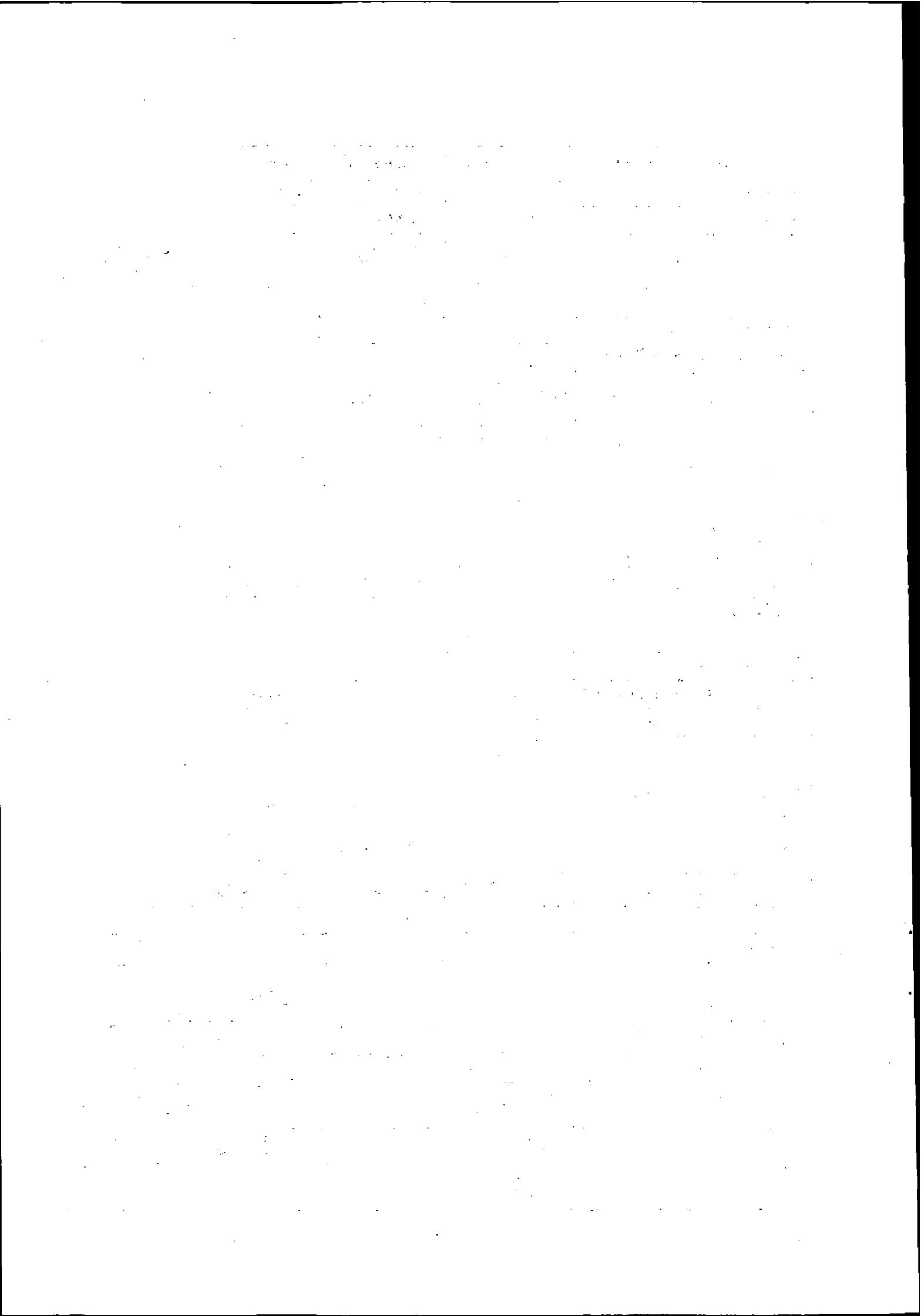


システムの階層構造

表 7-1 パーソナルマシンの性能

マシン名	A L T O	D O R A D O	S T A R	P E R Q	APOLLO DOMAIN	
メーカー	Xerox Palo Alto	Xerox Palo Alto	Xerox	Three Rivers Computer	Apollo Computer	
発表時期	1973	1978	1981	1980	1980	
C P U	プロセッサ	NOVA 1220類似	16ビットカスタム	ビットスライス	ビットスライス(16b)	MC 68000
	ビット数	16	16	16	16	16
	マシンサイクル	170 ns	50 ns		170 ns	8 MHz
	制御記憶	32b×4KW	34b×4KW	4KW	48b×4K~16KW	—
	その他	TTL 0.4MIPS	ECL パイプライン制御		・Pコード: 100万コード/s ・ユーザプログラム	AMD9512(演算チップ) ディスプレイ制御は別の MC 68K
メモ リ	主記憶	512KB	1MB~32MB	384KB	256K~1MB	256K~1MB
	サイクルタイム		14μs		680 ns	
	仮想空間		2 <sup>24</sup>		2 <sup>32</sup>	2 <sup>24</sup>
	キャッシュ		8K~32KB (100ns)			
	その他					
ディスプレイ	性能	808×606 (8'×10')	808×606	809×1024(6画面W) (105'×136')	768×1024 (11'×8.5')	1024×800
	ポインティングデバイス	マウス	マウス	マウス	タブレット (ビットパッド)	
ファイル	D I S K	3MB	80MB	8~23MB	12~24MB	Min.33MB
	F D D			1MB		
ネット ワーク	種類	Ethernet	Ethernet	Ethernet	Ethernet	リング
	転送レート	3Mb/s	3Mb/s	10Mb/s	10Mb/s	10Mb/s
I/Oインタフェース					RS232, IEEE-488	
その他のI/O					音声出力	
ソフ トウ ェア	O S	マルチ・タスク、 シングルユーザ	同 左	同 左	シングルプロセス、 '82 UNIX可能	
	言語	BCPL, Mesa, LISP, Small talk, etc.	同 左	Mesa, CLISP	PASCAL, etc.	FORTRAN, PASCAL, LISP, etc.
	エディタ					
	D B M S					
	応用					
その他						
価 格		—	—	\$ 16,595	\$ 19,950	\$ 28,000 1,300万~1,500万
特 徴						
そ の 他		・CMU、スタンフォード、MITに寄付 ・実験機 ・1200台以上設置	・実験機 (ALTO拡張版)	・ALTOの商品版	・100台以上使用中 (うちCMU 35台) ・ALTOの姉妹版	

マシン名	JERICHO	Symbolics 3600	LAMBDA	S P I C E	iAPX432	
メーカー	B.B.N.	Symbolics	Lisp Machine	CMU	Intel	
発表時期	1981	1981	1982春(発売予定)		1980	
C P U	プロセッサ	Am 2903	マイクロプログラム	LISP(ラムダマシン) +Nuマシン(68000)	マイクロプログラム	GDP(1D, MS(43201) EU(43202) IP(43203)
	ビット数	32	36	32		可変長
	マシンサイクル	230 ns				8MHz、5MHz
	制御記憶	64b×4KW(RAM) 64b×2KW(EPROM)	48b×12K (max.16Kまで可能)*		16KW	16b×4K
	その他	STTL、ビデオはECL	・パリティチェック ・1Mマクロ(LISP、 For. Pas.)命令/s ・FEPとしてMC68K	Nuバス(32b)		マルチプロセッサ指向
メモ リ	主記憶	512K~16MB	1MB(max.32MB)	1MB/ボード	1MB~	
	サイクルタイム	720ns(ページング含)	600ns			
	仮想空間	2 <sup>24</sup>	2 <sup>28</sup> (36bW)	2 <sup>32</sup>	2 <sup>30</sup> ~2 <sup>32</sup>	2 <sup>40</sup>
	キャッシュ					
	その他		ECC			
デブ イル スイ	性能	1024×760(BW) 1024×1024×9(カラー)	1000×1000(カラー)	1024×1024(BW)	600×800	
	ポインティング デバイス	マウス	マウス	マウス		
フ ァ ィ ル	D I S K		67MB		100MB	
	F D D					
ネッ ト ク	種類	Fibernet(リング)	Ethernet	Ethernet	Ethernet	
	転送レート	2Mb/s	10Mb/s	10Mb/s	10Mb/s	
I/Oインターフェース	RS 232	IEEE-796 (マルチバス)				IP経由
その他のI/O	音声出力	音声出力		音声入出力、 イメージ入出力、 A/D、D/A		
ソ フ ト ウ ェ ア	O S		シングルユーザ	68K上ではUNIX	マルチプロセス、 マルチランゲージ UNIXサポート	iMAX (一部シリコンOS)
	言語	InterLisp、PASCAL	ZetaLisp、(FORTRAN 77、PASCAL)	MACLISP	PASCAL、ADA、 LISP、C、FORTRAN	ADA (VAXクロスコンパイラ)
	エディタ		ZMACS	EMACS(68K上)		
	DBMS					
	応用		ZMail			
	その他					CANVAS (グラフィックパッケージ)
価 格	\$60,000 ~80,000	\$75,000	Nuマシン\$15,000 \$50,000 (1MB+300MBDK)			GDP:20~30万円、 評価ボード:110万円
特 徴		・MIT CADR マシン 系の最新版 ・"FLAVOR" (オブジェクト指向)		マルチランゲージ (ファームウェア化)		オブジェクト指向 32bVLSI *μメインフレーム*
そ の 他		・LISPマシン ・1982年第2四半期 に出荷の予定 *Lisp Machine社版の データ		・現在開発中 ・ソフト開発はPERQ を使用		適用分野(インテル) ・組込み型 ・トランザクション処理 システム



## 8. データベース技術および知識ベース技術

### 8.1 データベースシステム(DBS)

#### 8.1.1 データベース管理システム(DBMS)

データベース管理システムは、基本ソフトウェアの1つとして、コンピュータ・メーカーやソフトウェア会社により、パッケージの形で提供されている。その分類は、データモデルによって次の3種類に大別することができる。

- (1) 階層型(トリー型)
- (2) ネットワーク型
- (3) リレーショナル型

(1)はIBMのIMSに代表されるものでデータを親子関係で結んでいる。(2)は米国においてデータベースのタスクグループ(CODASYLの下部機構)が標準化したもので、COBOL言語を親言語としている。データモデルは、スキーマと呼ばれるもので相互の関係を記述する。データ構造は階層型も含んでいるが、ネットワーク型が可能であり、親(オーナー)と子(メンバー)の関係が自由にとれる利点がある。(3)はIBMのCODDが1970年に提唱したもので、データ構造が表形式になっており自然な表現形態であることやデータセットに対する演算が整備されているために注目を集めている。

図8-1 データモデル

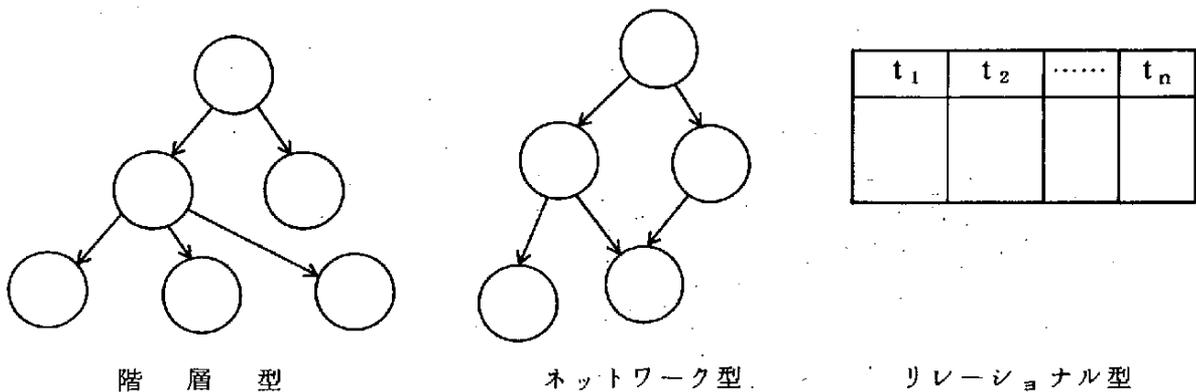


表 8-1 リレーショナルDBMSの比較 (COMPUTERWORLD '82.1.11)

Feature	SQL/DS	Oracle	Ingres	IDM500
<u>Data Definition Support</u>				
Variable Length Rows	Yes	Yes	Yes	Yes
Null Values Supported	Yes	Yes	No	No
Dynamically Add New Columns	Yes	Yes	No	No
Support Stored Commands	Yes	No	No	Yes
Dynamic View Definition	Yes	Yes	Yes	Yes
Automatic Elimination Of Duplicate Rows	Optional	Optional	Yes	Optional
Dynamic Data Base Expansion	Yes-Utility	Yes-Utility	No-O/S Function	Yes-Command
<u>Data Manipulation Support</u>				
Storage of View Text	Yes	Yes	Yes	No
Append Table to Itself	No	No	Yes	Yes
Support for User Files	No	No	No	Yes
Specify Disk the Table Is To Be Placed On	Yes	No	No	Yes
Maximum Size Char. Field	32767	255	255	255
Define Synonyms	Yes	No	No	No
Sort Query Results	Yes	Yes	No	Yes
Data Conversion Functions	No	No	Yes	Yes
Help Facility	Yes	No	Yes	N/A
Integrity Constraints	No	No	Yes-Single Table	No
Three-Walking Syntax	No	Yes	No	No
Outer-Join Syntax	No	Yes	No	No
<u>Data Control Facilities</u>				
Access Permission for Tables and Views	Yes	Yes	Yes	Yes
Permission Structure	Decentralized	Decentralized	Centralized	Centralized
Resource Authorization	Yes	No	No	Yes
Password for Data Base Access	Yes	Yes	No	No

(i) データ定義、データ操作、データ制御

Feature	SQL/DS	Oracle	Ingres	IDM500
<u>Index Support</u>				
Primary Index Required	No	Yes	No	No
Support Dynamic Index Def.	Yes	No	Yes	Yes
Support Multicolumn Indexes	Yes	No	Yes	Yes
Support Variable Length Keys	Yes	Yes	Yes	Yes
Physical Ordering of Data (Clustering)	Yes	Only on Initial Load	Only on Initial Load	Yes
Change Clustering Index	No	No	Yes	Yes
<u>Locking</u>				
Levels	Row, Page, Table, Data Space	Row, Table	Table	Page, Table
Automatic Lock Escalation	Yes	No	No	Yes
Storage Statistics Available	Yes	No	Yes	No
<u>Optimization</u>				
Access Method	Compiled	Interpretive	Interpretive	Interpretive
Path Selection Methods	Evaluate Indexes, Scans, and Two Join Methods	Evaluate Indexes	Evaluate Indexes, Scans, and One Join Method	Evaluate Indexes, Scans, and One Join Method
<u>Transaction Management</u>				
Multi-Statement Transactions	Yes	Yes	No	Yes
User-Controlled Backout	Yes	No	Yes	Yes
Support Nested Transactions	No	Yes	N/A	No
Disk Recovery Techniques	Roll Forward & Backout	Roll Forward	None	Roll Forward & Backout
<u>Host Language Interface</u>				
Type	Precompiler	Call	Call	N/A
Languages Supported	PL/1, Cobol and Assembler	Fortran, Cobol, PL/1, "C" and Assembler	Pascal and "C"	N/A
Support for Multiple Cursors	Yes	Yes	No	N/A
<u>Computer Environment</u>				
CPU's	43XX, 370, 303X	PDP 11/XX, VAX	VAX	N/A
Operating Systems	VSE	RSTS, RSX-11M, IAS, Unix, VMS	VMS	N/A

表 8-2 主要な DBMS

データ モデル	名称	開発企業	通用機種	販売 開始年	言語	販売数		販売価格		備 考
						国内	海外	買取	レンタル	
階 層 型	IMS	IBM	S/360/370/ 303X	1970	ホスト言語 DL/1	70~ 100	約1,300	—	23~ 52万円/月	
	ADM	日立	HITAC 8000シリーズ HITAC Mシリーズ	1976	ホスト言語		—	—		
ネ ッ ト ワ ー ク 型	EDMS	三菱	COSMO 900/700 MELCOM 7700/7500	1970	ホスト言語	約50	—	—	17万円/月	
	DMS 1100	ユニパック	UNIVAC 1100シリーズ	1970	ホスト言語	120	700	—	16~ 22.5万円/月	
	DMSII	パロース	B1800~ B7800	1973	ホスト言語	230	2000~ 2500	300~ 800万円	10~ 23万円/月	受注高 250(国内) B1900は70%DB
	IDSII	ハネウェル	Series 6000 (GCOS)		ホスト言語		400			
	TOTAL	Cincom Systems	IBMS/370、 Cyber、 PDP-11、 Interdata	1974	ホスト言語	53	3,500	500~ 1,500万円	14~ 32.5万円/月	
	SYSTEM 2000	インテル	IBMS/370、 CDC 6000、 UNIVAC	1976	ホスト言語 ユーザ言語	1	約600	2,000~ 4,000万円	50~ 100万円	
	IDMS	Cullinane Corpora- tion	S/360/370/ 303X/4300	1973	ホスト言語	4	1,000	1,400~ 1,800万円	可	国内販売はCRC リレーショナルへの拡張 (1982第1Q)
	PDM	日立	HITAC 8000シリーズ HITAC Mシリーズ	1976	ホスト言語		—	—		
	ADBS	日電	ACOS 77 NEAC 600~900	1976	ホスト言語	16	—	—	無料	
	AIM	富士通	FACOM Mシリーズ	1977	ホスト言語 1部 ユーザ言語	約100	—	—	22~ 50万円/月	
リ レ ー シ ョ ナ ル 型	QBE	IBM		1978 (備考)	QBE言語				17万 900円/月	1975(ZLOOF) Proc. AFIPS NCC 参照
	ADABAS	Software AG	S/360/370/ 303X/4300、 PDP-11、 VAX	1976	ホスト言語	77	850	3,600万円	可	厳密にはCoddのモデル と異なる。
	INQ	日電	ACOS 77 NEAC 600~900	1976	ホスト言語	28	—	—	17~ 24万円/月	同上
	SQL/DS	IBM	S/370/4300	1982	ホスト言語 SQL	—	—	—	7.5~ 23.4万円/月	System Rの商用化
	ORACLE	Relational Software	IBMS/370 DEC PDP11	1979	ホスト言語 C言語	3	80	1,920万円		
	INGRES	Relational Technology	VAX	1981	ホスト言語 C言語		15			
	Model 204	Computer Corpora- tion of America	S360/370/ 303X Mシリーズ	1979	ホスト言語 ユーザ言語	7	70	4,520万円 2,700~ 4,520万円	80~ 133万円/月	国内販売は三井情報開 発
HAIM	富士通	—	—	—	—	—	—	—		

(コンピュータピア1979, 11, 日経コンピュータ1981, 11.2, JECC)

### 8.1.2 データベースマシン (DBM)

データベースマシンは、データベースの大規模化に伴って、ホストマシンでの負荷を軽減し、性能向上を図るため、データベース機能をホストから分散させ、専用マシン化する必要から最近多くの実験システムが試作されている。本格的な商用システムは、80年代の中頃と予想されている。データベースマシンの形態は、ホストのバックエンドが中心でそれをハードウェア化したハードウェア後置型と汎用コンピュータのソフトウェアで実現するソフトウェア後置型に分類される。更にハードウェア後置型は以下のように分類される。

- (1) 統 合 型
- (2) S I M D 型 (セル型)
- (3) M I M D 型
- (4) 連想メモリ型
- (5) データフロー型
- (6) 機能分散型
- (7) そ の 他

(1)は(2)~(6)のいずれかの機能を併せ持つシステムで本格的なデータベースコンピュータを目標とするもの。(2)はデータの読み書きをブロードキャスト的に複数セルで同時に行うもの。(3)は各処理装置が別々にデータベース処理を行う。(4)はデータベースの検索を高速化するために連想メモリ装置を使ったもの。(5)はデータフローの概念で並列に行うもの。(6)はデータベース処理機能をマルチプロセッサで処理するものである。これ以外にS/38のようにDB機能をファームウェア化したものやファイル検索を高速化するインテリジェントファイル等がある。

データベースマシンのデータモデルは、ソフトウェア上、オーバーヘッドの大きいリレーショナル型の高速化をめざすものが多い。

表 8-3 データベースマシン

分類	名称	開発企業	ホスト機種	開発年	データモデル	規模	性能と価格	備考
統合型	DBC (Database Computer)	オハイオ州立大 (米)		1976	独自リレーショナル可	10GB以上		・現状テクノロジーで開発可能 ・セキュリティ保護用プロセッサがある
	CADAM (Content Addressable Data base Access Machine)	沖 (日)		1979	リレーショナル			・マルチマイクロプロセッサでレコード検索 ・提案
SIMD	CASSM (Content Addressed Segment Sequential Memory)	フロリダ大 (米)		1972	リレーショナル階層型ネットワーク型	10 <sup>8</sup> ~10 <sup>10</sup> W (1W=4B)		
	RARES (Rotating Associative Relational Store)	ユタ大 (米)		1975	リレーショナル			
	RAP.1 (Relational Associative Processor)	トロント大 (カナダ)		1976	リレーショナル			・回転体メモリをプロセッサでR/W(セル)
	RAP.2	トロント大 (カナダ)		1977	リレーショナル			
	EDC (ETL Database Computer)	電総研 (日)		1979	リレーショナル			・データモジュール: 1MBバルブ×8台
	RAP.3	アンカラ大 (トルコ)		1980	GDBMS (応用)階層ネットワークリレーショナル			
		INFOPLEX (Information Management System using 3 Micro-Processor Complex)	MIT (米)		1975	リレーショナル階層型ネットワーク	100GB	10 <sup>8</sup> ~10 <sup>10</sup> b/sec
MIMD	DIRECT	ウィスコンシン大 (米)		1979	リレーショナル		1ページ(16KB)のサーチ 12m sec	・1ページ=1CCDモジュール ・マトリックスSW ・試作中
	AIS (Associative Information Structure)	横浜国大		1979	独自(リレーショナルモデルの改良)			・HITAC M-160IIとミニコンピュータによるパイロット・システムを開発
	DIALOG (Distributed Associative Logic DB Machine)	パーデュー大 (米)	PDP11/70	1980	分散データベースリレーショナル			・シミュレーション1983 ・プロトタイプ 1985
		可変構造多量処理データベースマシン	東大 (日)		1980	リレーショナル		
連想メモリ	STARAN型	シラキューズ大 (米)		1979	リレーショナル	10 <sup>8</sup> b以上	全データの検索(1GB) ≤ 2 sec 256×256bのサーチ: 53μs アレイモジュールへのローディング: 500μs	・ソートエンジン、サーチエンジン ・プロトタイプ試作中 ・4ソートエンジン ・4サーチエンジン ・動的マークビット ・データ駆動先回り制御 ・提案
データフロー	DSDBC (Data System Data Base Computer)	北大 (日)		1980	リレーショナル			・事務用計算機 ・DB機能のファームウェア化
機能分散	SPIRIT	慶大 (日)		1980	リレーショナル			・インテリジェントディスク ・FAST3805(アクセス1ns) ・出荷'81第1四半期
	S/38	IBM (米)		1980	リレーショナルCoddモデルとは異なる	2GB~3GB	モデル3(1.2MBメモリ、387、1MBDK)137810ドル モデル5(1.7MBメモリ、387、1MBDK)194245ドル FAST3805 (3250万円) DBMS(1500万~2500万円) DBAP(約625万円)	・インテリジェントディスク ・FAST3805(アクセス1ns) ・出荷'81第1四半期
その他	DBAP (Database Assist Processor)	インテル (米)		1979	ネットワークSystem 2000		4MB/secで処理 ホストのみに比較して10倍	・インテリジェントディスク
	CAPS	ICL (英)		1979			レスポンス時間 1/45~1/13 CPU時間 1/11~1/45	・インテリジェントディスク
	IFC (Intelligent File Controller)	日立		1980				
ソフトウェア後置型	XDMS (Experimental Data Management System)	ベル研 (米)	UNIVAC 1108	1974	ネットワーク(DMS-1100)	小規模	ロック、リカバリ機構	・META-4使用 ・実験用
	GDS (Generalized Database Subsystem)	日電 (日)	ACOS 400	1977	汎用ネットワークリレーショナル		性能2~10倍向上 コスト/性能 2.7倍	・実験システム
	EDDS	東芝 (日)	パターン大型プロジェクト	1978	リレーショナル	可変		・機構 200MBディスク×4台 プロセッサ2台
	IDM500 (Intelligent Database Machine)	Britton-Lee Inc. (米)	GP1B RS 232Cインタフェース	1980	リレーショナルINGRES	8MB~32GB	MAXリレーション50 最小構成約3万ドル データベースアクセスレータ100ns、10MIPS	・SQLを介してSeries Iとインタフェースあり
	ADABAS データベースマシン	ソフトウェアAG (西独)		1980	リレーショナルADABAS		IBM370/148の1.5倍 ADABASユーザ6300万円 それ以外9100万円 (レンタルなし)	・納入1981
	MIX	Copernique		1981	ネットワークリレーショナル		1~16台のDK (各80~600MB) DKキャッシュ (MAX、16MB)	・1982第4Q (ネットワーク型は完了)

## 8.2 分散データベース

分散処理、ローカル／グローバルネットワークの普及により、データベースも分散処理の必要性が高まりつつある。分散データベース管理システムには既に開発されたものがある。分散データベース用の専用マシンは今の所計画、検討中のものしかない。

### (1) 分散DBMS

表 8-4 代表的な分散DBMS

SDD-1 (A System for Distributed Database)
開発企業：CCA (Computer Corporation of America)
開発年：1976～1979
モデル：リレーショナル
マシン：DEC-10 (TENEX)、DEC-20 (TOPS-20)
ネットワーク：ARPA
特徴：トランザクションモジュール(TM)、データモジュール(DM)の2ソフトウェアからなる。遠隔地への問合せは近接に比べ1000倍程度遅れる。 CCAではこれをもとに商用の分散システムMODEL 204 (IBM版)を計画。

分散INGRES
開発企業：カリフォルニア大学(バークレイ分校)
開発年：1976～1980
モデル：リレーショナル
マシン：PDP-11/45 (UNIX)
ネットワーク：校内の専用ネットワーク
特徴：均質型。マスタサイト(トランザクション)・スレーブサイト(オリジナルデータ)・コピーサイト(コピーデータ)からなる。ローカル処理の重視。コンカレンシコントロールとして性能アルゴリズムと信頼アルゴリズムを使い分ける。

<b>POLYPHEME</b>
開発企業：CII-Honeywell-Bull、グルノーブル大学
開発年：1976～1979
モデル：混在（リレーショナル、ネットワーク、階層）
マシン：IRIS 80
ネットワーク：CYCLADES、TRANSPAC
特徴：異種DBMS（URANUS（リレーショナル）、SOCRETES（ネットワーク）、IMS（階層）、IDS II（ネットワーク）等）。1つのグローバルマシンとn個のローカルマシン。 問合せ→GM→LM→GM→回答

<b>VDN</b>
開発企業：Nixdorf、ベルリン工科大学
開発年：1977～1981
モデル：リレーショナル
マシン：Nixdorfのミニコンピュータ
ネットワーク：X. 25
特徴：バンキングや企業内ネットワーク等の decentralize 型の商用目的

(2) 分散データベースマシン

現在計画中のものとして次の2つがある。

① MUFFIN

カリフォルニア大学バークレイ分校で検討されているもので、分散DBMSとしては上記分散INGRESを使用し、処理モジュールとして、トランザクションを処理するAセルとデータベースを保有するDセルをバス上に多数配置したもので、これらはゲートウェイを介して他バス上のシステムへもリンクできる。性能予想としては252のAセル、74のDセル、1MB/sのバスを想定すると1255トランザクション/secといわれている。

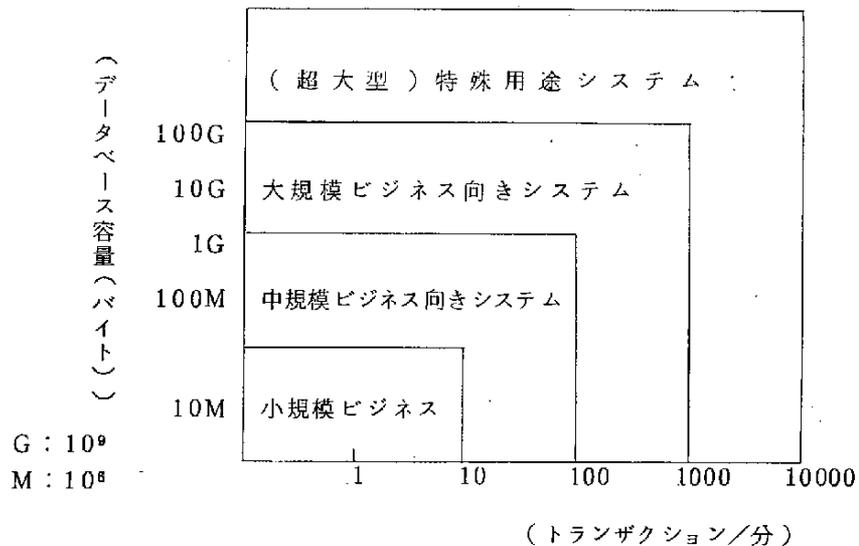
② SABRE

仏の国家規模プロジェクトとしてSIRIUSがあり、その一環として上記POLYPHEMEも開発された。1980年1月以来SABREと呼ばれるプロジェクトが開始されている。その目標は、SIRIUSの応用レベルプロトコルの標準化と、バブル、光ディスク、VLSI技術を使ったデータベースマシンの開発である。

付1. データベースの規模

データベースの規模を正確に分類することは困難であるが、図のような基準がある。この分類にはもちろん例外もある。つまり小容量で非常に高速な応答や大容量でもトランザクション数は少ないものもあるが一般的傾向を表わしている。また特殊用途システムは、このグラフでは大規模にしか現われていないが、小、中規模もある。

図8-2 データベースの規模

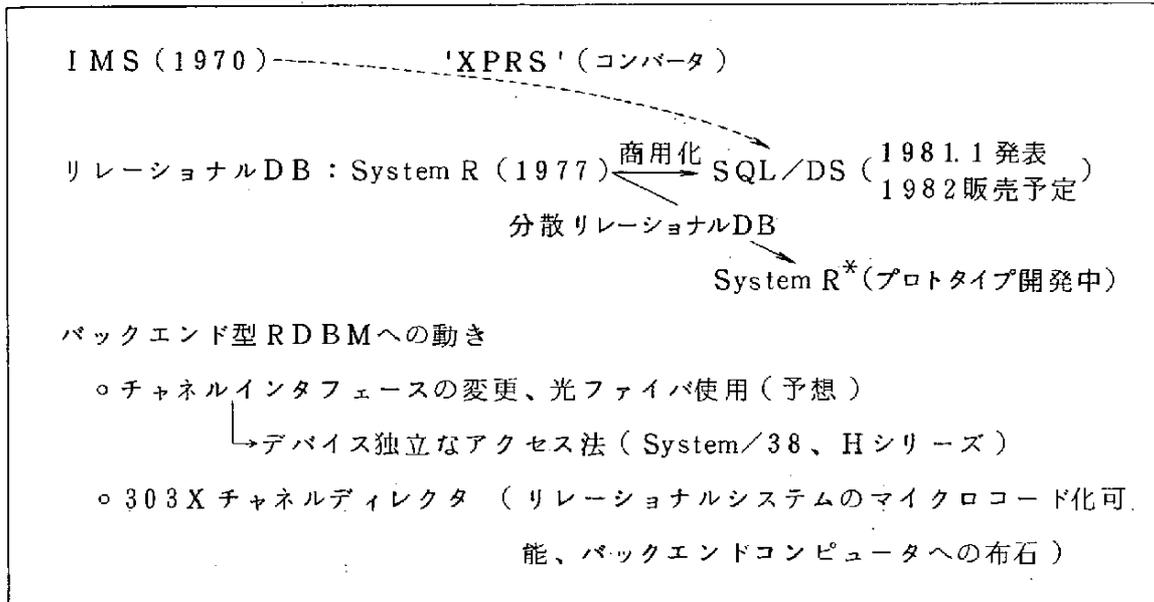


(出所: Proc. AFIPS 1980 NCC, P.237)

## 付2. IBMの動向

IBMが、IMSやCICSのユーザを保持しながらリレーショナルなDBMSへ静かに移行して行く兆候は多くの人々が指摘している所であり、最近の情報からその動きを追うと下のようになる。

図8-3 IBMのDBMSの動向



## 付3. データベース問合せ言語

問合せ言語としては、ホスト言語 (PL/1、COBOL、FORTRAN 等) 内にアクセス機能を有するものや、専用の言語 (DL/1、QBE、SQL 等) を設定するものが多いが、より非専門家向として自然言語による問合せ (いわゆるQAシステムにつながる) も実験的に試みられている。

米国SRIでは1976年以来海軍向のシステムとして、LADDER (Language Access to Distributed Data with Error Recovery) が開発されており、これは自然言語を使用する計算機のアクセス言語に翻訳する方法をとっている。英国エディンバラ大学のWarrenは、1981年度のVLDB学会で自然言語をPROLOG言語に翻訳してリレーショナルDBにアクセスするシステムCHATを開発したと報告している。PROLOG言語を使ってリレーショナルDBをアクセスするのは、別の所でも開発されている。

### 8.3 知識ベース

データベース技術の応用分野は広範囲であるが、知識ベースは、データベースの形式的構造だけでなく、意味構造を持っており、従来のデータベース技術だけでは実現が困難な分野である。今後のデータベースは、シンタクスばかりではなくセマンティクスの記述能力をもつことが予想され、新しいデータモデルの概念が生まれる可能性がある。現在考えられている知識表現形式は多数あるが以下のモデルを基準にして分類してある。

#### 1. プロダクションルール ( Production Rule ; PR )

人間のメモリ機能と認識プロセスを同時に記述するモデルとして、A. Newell, H. A. Simon により提唱されたもので、心理学における刺激→反応機構のひとつの拡張と考えられるプロダクション : Premise (前提) → Action (行動) の形式で記述されるルール集合と、その上に定義された制御構造から成る宣言的知識表現。

#### 2. フレーム ( Frame )

認識理解しようとする対象だけでなく、それに関連した構造化された情報を表現する枠組みとして、フレームと呼ばれる構造化情報形式を用いる。M. Minsky が提唱した。

#### 3. セマンティックネットワーク ( Semantic Network ; SN )

対象の持つ意味、役割の相互関係を取扱うためのラベル付有向グラフ表現形式。ある場合にはフレーム表現と非常に類似した表現形式となる。

#### 4. 述語論理 ( Logic )

知識を述語論理式で記述するもので、そのサブセットである Horn 節に限定すれば PROLOG が代表的である。

#### 5. 手続き型 ( Procedural )

知識を宣言的に記述する上記 1～4 と異なり、How to 形式で表現するものであり、手続き型言語によるプログラムにより記述される。

#### 6. 意味素 ( Semantic Primitives )

知識表現をするための語彙を規定するもので、基本要素として、Y. Wilks は 100 個、Schank は 14 個を設定している。

図 8-4 知識ベースの系譜

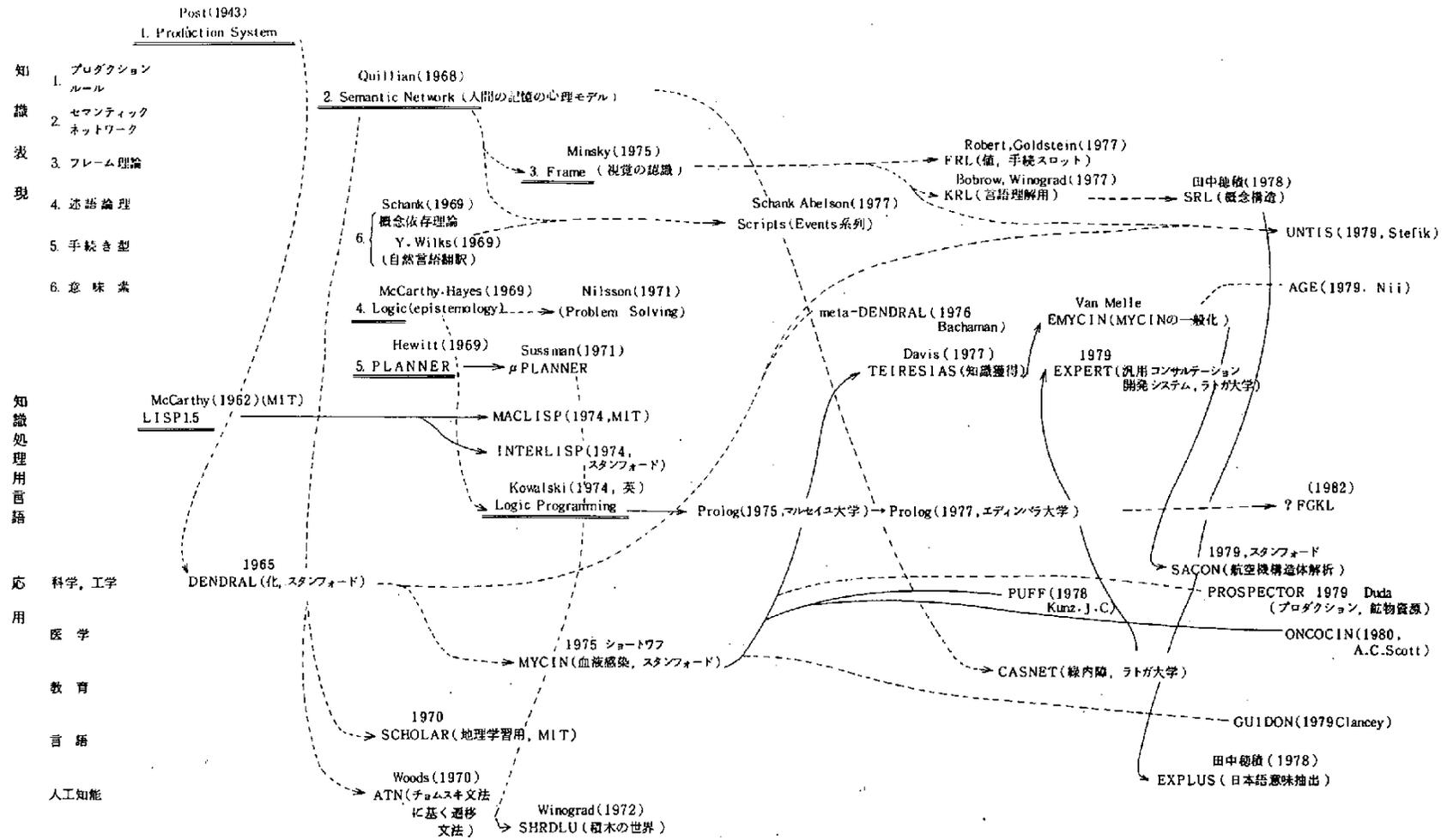


表 8 - 5 知 識 ベ - ス

知識表現 モデル	システム名	開 発 者	発表年	記 述 言 語	概 要、特 徴 等	備 考
プ ロ ダ ク シ ョ ン ル ー ル	ELIZA	J. Weizenbaum (MIT)	65~		精神分析用自然言語質問応答システム。キーワードによる変換。SLIP関数	
	DENDRAL	E. A. Feigenbaum et al. (Stanford Univ.)	71~	INTERLISP FORTRAN SAIL	有機化合物の構造式の決定を行なう。heuristic search 知名度高し。Ph. Dクラスの能力を持つ。	稼 動
	PUFF	E. A. Feigenbaum et al. (Stanford Univ.)	78~		肺機能障害診断 MYCINの推論機構を用いたもの。診断結果は専門家のその90%と言われる。	研究中
	MYCIN	E. Shortliffe B. G. Buchanan (Stanford Univ.)	75~	INTERLISP	血液伝染病と脳膜炎の診断及び投薬。Production Rule 知名度高し。	稼 動
	EMYCIN	W. VanMelle (Stanford Univ.)	79	LISP	MYCINの一般的な部分だけを取り出したシステム。Essential-MYCINこれは、実は、MYCINに対するEXPERTの汎化になっている。メタシステム。	稼 動
	Meta-DENDRAL	E. A. Feigenbaum et al. (Stanford Univ.)	76		DENDRALに推論規則の自動作成を付加。メタシステム。	稼 動
	REQUEST	W. Plath (IBM)	76~	LISP	実験システム、変形文法	
	AM	D. Lert (Stanford Univ.)	76~		Mathematical Discovery、heuristic search	研究中
	TEIRESIAS	P. Davis (Stanford Univ.)	77~	LISP	MYCINのデータを探索する。メタ知識を扱う知識獲得システム、tree search、inference	稼 動
	SU/X	H. P. Nii et al. (Stanford Univ.)	77~		Signal Understanding、blackboard、規則に基づく推論	
	MOLGEN	N. P. Martin et al. (Stanford Univ.)	77~	INTERLISP	分子遺伝学(DNA)実験計画立案	稼 動
	CRYSALIS (SU/P)	R. S. Engelmores H. P. Nii (Stanford Univ.)	77~		蛋白質の3次元構造の指定、blackboard	稼 動

知識表現モデル	システム名	開発者	発表年	記述言語	概要、特徴など	備考
プロダクションルール	EXPERT	S. M. Weiss (Rutger Univ.)	79	FORTRAN	推論機構と知識ベースが完全分離した汎用コンサルテーション開発システム。CASNETの一般化、PS、知名度高し、メタシステム	稼動
	SACON	J. S. Bennett R. S. Engelmores (Stanford Univ.)	79~		構造解析用プログラム使用法コンサルティング EMYCINを使用	
	MECS-AI	開原(東大)他	79~	EPICS LISP	心不全治療コンサルテーション Production Rule.	研究中
	AGE	H. P. Nii et al. (Stanford Univ.)	79~	INTERLISP	知識ベースプログラム作成システム blackboard 階層構造記述メタシステム	研究中
	GUIDON	W. J. Clancey (Stanford Univ.)	79~		CAI(医療、科学分野)	研究中
	ONCOCIN	A. C. Scott (Stanford Univ.)	80~		リンパ系癌治療法の助言	研究中
フレームム	SPEECHLIS	B. Nash-Webber (BBN)	73		音声理解システム。SN, frame( case frame)	
	INTERNIST	Popple, Myer (Pittsberg)	75~		内科診断。階層 tree structure (15000ルール) 内科疾患の80%を含むKDB(Knowledge DB) を持ち、既存のKDBで最大と言われる。	稼動
	PLANES	D. Walty (Illinois Univ.)	76~		US Navy 3-M(Maintenance, Material & Management) D. B. ATN, Case Frame	
	PIP	Tuft Medical Center (MIT)	78~	MACLISP	腎疾患診断、frame structure	研究中
セネマントデウィック	SCHOLAR	A. Collins(MIT) M. L. Miller (BBM)	70~	LISP	地理情報等の開いた世界(不完全な知識)を対象とするシステム。 SN(Semantic Network.), inference(関数型)	
	SOPHIE	J. S. Brawn et al. (BBN)	75~	LISP	CAIシステム(電子工学)、SN、inference	

知識表現 モデル	システム名	開 発 者	発表年	記 述 言 語	概 要、特 徴 等	備 考
セネ マッ ント テ ィ ワ ィ ク	SHRDLU	T. Winograd (MIT)	72~	Micro-PLANNER	積木の世界、SN(対象指向)、Specialist Programの集合体。対話型シミュレーションプログラム	
	TORUS	J. Mylopoulos (Tronto Univ.)	76~		ZETA R.D.B.の自然言語フロントエンドとして設計されたシステム	
	RADDER	E. D. Sacerdoti et al. (SRI)	77~		US Navy, SN DBMS、英語を用いて分散DB(R.DB)のアクセス可	
	CASNET	Kulikowski, Weiss (Rutger Univ.)	78~	Interactive FORTRAN	緑内障の診断システム。Network 表現(モデル指向)	稼 動
拡大 遷移 ネッ トワ ィク	PLATON	長 尾 (京大)	74~		文法解析システム SN PS ATN	研究中
	ELINGOL	電総研	75~		同 上	研究中
	LUNAR	W. Wood (BBN)	76~		月面情報システム ATN (Augmented Transition Network) 拡張推論ネットワーク NLを用いたDBMS	
	ROBOT	L. R. Harris (Dartmouth Univ.)	77~		汎用ADABAS, ATN, DB情報に依るあいまい性除去 基本的機能はPLANESとほぼ同様	
そ(階 層型) 他)	PROSPECTOR	R. O. Duda (SRI)	79~		鉱物資源に関する地質学者アシスタンスシステム 階層構造	開発中
	ADVISOR	M. R. Genesereth	79~		MACSYMA の使用法コンサルティングシステム parameterized procedural net (PPN)	稼 動
そ の 他	MACSYMA	J. Moses (MIT)	71~		不定積分(SlagleのSAINT等と同類)	稼 動
	CHEMICS	佐々木 他 (豊橋技大)	74~		有機化合物の構造決定	
	ヤチマタ	渋谷 他 (日本IBM)	77~		国土統計、名詞句データモデル	

知識表現 モデル	システム名	開 発 者	発 表 年	記 述 言 語	概 要 、 特 徴 な ど	備 考
そ の 他	RENDEZVOUS	E. F. Codd et al. (IBM)	78~		Walter Project. R. D. B. 会話型自然言語質問応答システム 形式的問合せ言語	稼 動
	KAUS	大須賀 (東大)	78~	FORTRAN	合金設計等 一階述語の変形(多層論理)で知識を表現	
	PLIDIS	ドイツ語研究所			シュツットガルト環境汚染対策システム、問題別志向の演繹	
	PHILQAI	R. Scha (フィリップス)			計算機導入に関するD. B., NLからDBLへの段階的変換	
	ESCAシステム	電総研	79~		X線光電子分光(ESCA)に依る組成分析システム。 知識のテーブル表現。	研究中

#### 8.4 知識ベースマシン

本格的な大規模知識ベースを実現するには、高速連想アクセス機構や並列処理方式の研究開発が必要となる。現在の知識ベースの規模は実験レベルのため、それを専用マシンで実行する段階には至っていないが、1980年頃から知識ベースマシンのアイデアが学会で発表されだした。これらはマイクロプロセッサ、VLSI技術によるところが大きく、今後活発に研究されるものと思われる。

表 8-6 代表的な知識ベースマシン

システム名	開発元	開発年	モデル	概要
Apiary Network	C. Hewitt (MIT)	1980	アクタ	3次元格子状にプロセッサを結合(Hyper-torus)複数アクタ/プロセッサ、CADRマシンでシミュレート中。
ZMOB	C. Rieger (メリーランド大学)	1981	分散知識ベース	高速バス上に256のプロセッサを配置。Z80で開発中。
NETL	S. E. Fahlman (MIT)	1979	セマンティックネットワーク	リンクを動的にワイヤリング 10 <sup>8</sup> 個のノードを試作中。
NON-VON	D. E. Shaw (コロンビア大学)	1981	リレーショナル	リレーショナルデータベースマシン上に22の知識ベースの公理系を設定する。MACLISPでシミュレート中。VLSI検討。
Hearsay-II	R. D. Ferrel (CMU)	1977	プロダクション	共有メモリ型マルチプロセッサ、音声理解システム。

## 9. ソフトウェア（プログラミング言語，特に非手続き型言語など）

### 9.1 プログラミング言語の主要動向

#### 9.1.1 プログラミングの動向 1)

##### (a) 記述形式にみられる傾向

非手順的：non-procedural programming LUCID

関数型：functional programming LISP、FFP (Backus)

再帰的：recursive programming LISP、SCHEME

記号化：combinatory programming APL、FFP

非決定的：non-deterministic programming PLANNER、PROLOG

探索指向：search directed programming Micro-PLANNER、CONNIVER

関係型：relational programming VAL、SNOBOL、LINGOL

並行型：concurrent programming Concurrent PASCAL、Solo

並列処理：parallel programming ALGOL 68

駆動型：event driven programming VAL、PLASMA

対象指向：object oriented programming CLU、SMALL TALK

##### (b) 構成方式にみられる傾向

構造化（系統的）：structured (systematic) programming

会話型（対話型）：conversational (interactive) programming

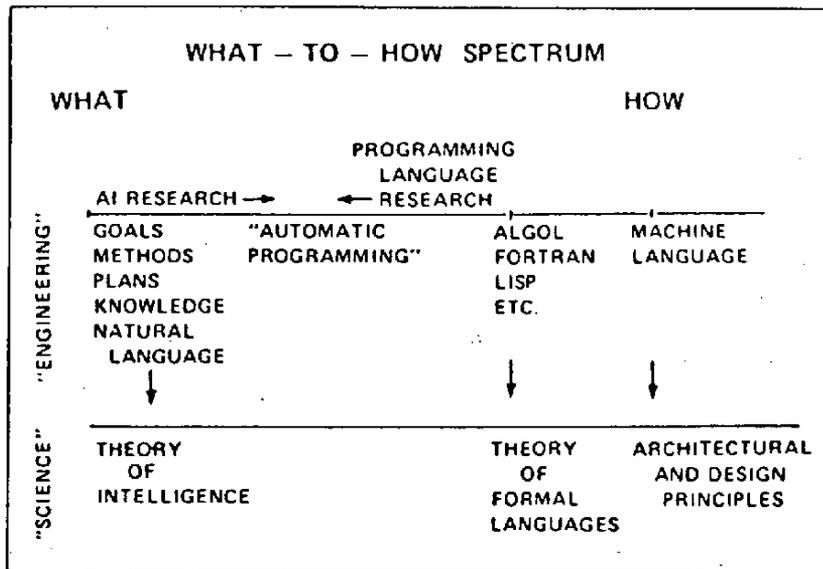
自動：automatic (automated) programming

知識利用：knowledge based programming

9.1.2 プログラミング言語の動向 2)

動 向	要 望 度	実 現 の 見 通 し	
		あ り	な し
自 然 言 語 化	○		○
専用アプリケーション言語 (定義と実装は自動化)	○	?	?
プログラム検証のための実際的手法	○		○
非手順化 (more nonprocedural)	○	○	
理論的なアプローチ	○	○	
DBMSのためのよりよい問合せ言語	○	○	
システムプログラミングにおける リアルタイム 高級言語の利用	○	○	
現行主要言語の継続的な重用		○	
無統制な新言語の氾濫の継続		○	
拡 張 型 言 語	(○)		○
ネットワーク用専用言語			○
ジョブ制御言語の改良	(○)		○
重要な新概念の具現化	(○)		○
ADAによる非DOD系言語の置換			○

9. 1. 3 人工知能研究とプログラミング言語 3)



— Artificial Intelligence research is that part of computer science that is concerned with the symbol-manipulation processes that produce intelligent action.

— The potential uses of computers by people to accomplish tasks can be "one-dimensionalized" into a spectrum representing the nature of instruction that must be given the computer to do its job.

— At one extreme of the spectrum, the user supplies the intelligence to instruct the machine with precision exactly how to do his job step.

— At the other extreme of the spectrum is the user with his real problem (what he wishes the computer, as his instrument, to do for him).

— The user aspires to communicate what he wants done in a language that is comfortable to him (perhaps English); via communication nodes that are convenient for him (including, perhaps, speech or pictures); with some generality, some abstractness, perhaps some vagueness, imprecision, even error; without having to lay out in detail all necessary subgoals for adequate performance with reasonable assurance that he is addressing an intelligent agent that is using knowledge of his world to understand his intent, to fill in his vagueness, to make specific his abstractions, to correct his errors, to discover appropriate subgoals, and ultimately to

translate what he really wants done into processing steps that define how it shall be done by a real computer.

— Creating computer programs that act as "intelligent agents" near the what end of the what-to-how spectrum can be viewed as the long-range goal of AI research.

## 9.2 非手続き型言語

### 9.2.1 「非手続き型（非手順的）」の概念

#### A. 常識的立場での説明

##### ① 共立総合コンピュータ辞典での説明：1)

ALGOL、FORTRANなどの普通のプログラム言語では、アルゴリズムを表現するために言語の基本単位である文の実行順序をプログラマが陽に指示している。こうした言語を「手順的」(Procedural)とよぶのに対して、評価や実行の順序をプログラマが陽に指示しなくともアルゴリズムを表現できるように考えられたのが、「非手順的」(Non-procedural)という概念である。

その考え方が生まれたのは決して新しいことではない。1950年代のプログラム言語の揺籃期から現在にいたるまで、ある時は脚光を浴び、ある時は死滅しかかりながら、生き続けて来たことばである。非手順性の精神に基づいて情報代数 (Information algebra) が創案され、またある程度の普及を見た形態としてレポート・プログラム・ジェネレータ (RPG) や決定表言語を生み出したものの、現在にいたるまで真に汎用性をもった非手順的というにふさわしいプログラム言語が民主化されるにはいたっていない。しかし一方では、新しい言語の開発に対して一つの目標と刺激を与えて来たのである。

「非手順的」という概念がこのような不安定な位置しか獲得できない第一の原因は概念自体の弱さにある。すなわち、絶対的あるいは厳密な意味でこの概念を定式化することが不可能であり、高々、時代の技術に相対的にしか意味をもち得ないということである。たとえば、最初の高級プログラム言語が出現した時代には、機械語レベルのコーディングの代わりに  $(A+B)+C$  と書いてこの算術式の値を計算できることは画期的であり、非手順的であるといえた。

こうした弱さにもかかわらず、特定の機械系、システムに従属する演算方式、データ表現、コントロールの細部を出来るだけプログラマから隠し、問題の記述とその解法に本質的に関連する概念によりプログラムができるようにしたいという願望を支える仮説的概念としてなお意味をもつであろう。

非手順性を言語設計の統一的基盤としている一群のプログラム言語が存在する、というよりも、各種の新プログラム言語に種々の非手順の様相が取り入れられつつある、というのが現状である。これらの様相は、新たな抽象度の高い表現手段を導入しデータやコントロール表現の細部の記述からプログラマを解放するという点で共通している。

② J. E. SAMMET による説明：2)

( " More Nonprocedural ( Equals Less Procedural ) Languages " という )  
テーマに対する説明としてのもの

これに含まれている基本的な考え方は、利用者は " 何が ( what ) " したいかを書き、それが " いか ( how ) " になされるべきかまでを書かなくても済むようにしたいということである。理想的には、利用者は単に問題の要求仕様を示すだけで、あとは適当なシステムによって、( まず ) プログラムが、( ついで ) 与えられた問題に対する解答が生成されるというようになって欲しい。

こうした種類のものに関係する技術は、自動プログラミング、仕様記述言語、問題定義言語、超高水準言語、それに非手続き型言語といった見出しの下に現われている。

非手続き型言語は、技術の推移に従って変わる相対的な意味内容のことばであるので、時がたっても目覚ましい進歩がみられるということのない分野である。いい方をかえれば、これの実現は漸進的に動いていくもので、はっきりした YES / NO で答えられるようなものではない。必要なことは、非手続き性の水準をどれだけ高めるかということである。

筆者の見方では、今後非手続き性の向上がみられることは確かである。したがって、利用者があまり細部にかかずらう必要の少なくなる言語が開発されてくると思われる。そのもっともよい実例は、データベースの分野に出そうである。しかし、これは広範な利用のある分野であるが、これだけが応用分野のすべてというわけではない。一般論としていえば、何らかの大きなブレイクスルーがなければ、今後の進展は小幅にとどまりそうである。そのようなブレイクスルーは筆者の視野には見当たらない。

B. 先進的立場での説明

① P. J. LANDIN による説明：3)

( " 手続き的 " 対 " 非手続き的 " についての ) 重要な区別のひとつは、つぎのような区別にある：

- ・ 計算機にどのような動作をしてもらいたいかをいちいち指示するもの。

・ どのような結果が欲しいかということだけを指示するもの。

これだけのいい方では、この区別を精しく追究するには耐えられない。そこで、「どのような結果が欲しいかということだけを指示する」ということに対する要件として、つぎの a~c の条件を提案したい。

(a) 各表現 (expression) は、入れ子型の部分表現 (subexpression) 構造をとる。

(b) 各部分表現は何かあるもの (通常は数、真理値、あるいは数値関数) を表意 (denote) する。

(c) 表現によって表意されるもの、すなわちその「値」は、その部分表現の値だけに依存し、部分表現のもつその他の性質には依存しない。

## ② ロジック・プログラミングに即した説明：4)

プログラムはアルゴリズムを記述したものである。R. A. KOWALSKI の有名なテーゼによれば、

ALGORITHM = LOGIC + CONTROL

となる。すなわち、アルゴリズムは論理要素と制御要素に分離できる。ここで、論理要素は、解くべき問題の定義とそれを解くために用いるべき知識とから成っている。また、制御要素は、それらの知識をどのように用いて、どのような手順で操作を進めれば求める解に到達できるか、さらには、それらを効率的にやるにはどうすればよいかの戦略などを定めるものである。

以上から明らかなように、アルゴリズムの意味内容 (what to do) は論理要素によって決められるのに対して、その操作的ならびに効率的な内容 (how to do) は制御要素によって支配されることになる。

非手続き型言語 (厳密にはロジック・プログラミング・システム) においては、制御要素の指定は論理要素の指定に従属させられ、システムによって決定されることになる。制御要素を特に指定する必要がある場合には、別の制御指定用言語を用いることになる。

一般に、プログラミング言語の水準が高くなればなるほど、また、プログラムするひとが素人になればなるほど、効率についての責任範囲、与えられた情報の利用管理などに対するシステムの役割は重くなっていく。

## C. 解 説

— 以上では、非手続き型言語の特徴づけとして4つの代表例をとりあげた。これらで現在までに行なわれている定義づけはほぼカバーしうるはずである。

— それぞれの定義の要点を整理すると、「非手続き的」とは、

(1) HOWではなくWHATを記述するもの …… A②

(2) 文の評価や実行の順序を無視してよいもの …… A①

(3) つぎの条件を満足するもの …… B①

(a) 各表現は、入れ子型の部分表現構造をとる。

(b) 各部分表現は何かあるものを表意する。

(c) 表現によって表意されるもの、すなわち、その「値」は、その部分表現の値だけに依存し、部分表現のもつその他の性質には依存しない。

(4) ALGORITHM=LOGIC+CONTROL が成立するとしたとき、

このLOGICの部分だけを記述するもの …… B②

これらの4つの定義のうち、最も厳密で最も基本的なものは(3)の定義である。後にみるように、(2)と(4)は(3)の特殊な場合に相当しており、(1)を厳密に表現すると(3)のようになってしまう。

— 上記(3)の定義に述べられた条件は、「referential transparency」とよばれる言語的性質を特徴づけるものである。

— この referential transparency は、プログラミング言語に即していえば、「副作用がない」ということに相当している。

— referential transparency の性質をもった言語ないし言語のサブセットのことを「applicative」（関数型あるいは関数的）という。これは関数型言語の基本となる概念である。

— 参考までに上記4つの概念が書かれた年代をあげると、つぎの通りである：

A①-1976、 A②-1981、 B①-1966、 B②-1979

B①が最も古いのが意外にうつるかも知れないが、一種のリバイバルとみることができるかも知れない。

9.2.2 関連する概念 3)、5)

前項では、「非手続き型」という概念に対してどのような定義が存在しているかをみたが、それだけではその本当の意味が把みがたいところがある。ここでは、「非手続き型」とほとんど等価的に用いられている他の概念や、それと極めて関係の深い概念について紹介し、前項の補足説明としたい。

	procedural の系列	nonprocedural の系列
等価概念	imperative	denotative declarative descriptive functional applicative
関連する概念	algorithmic	heuristic
	iteration	recursion
	Turing model	Gödel—Church model
	von Neumann machine	Non—von Neumann machine
	⋮	⋮

(1) 「imperative」対「denotative / declarative / descriptive」

— 従来からのプログラミング言語では命令文の形式が基本になっている。日常のことばの場合からも明らかなように、命令文は「HOW(いかに)」を記述するのに適した形式である。それに対して、「WHAT(なに)」を記述するのに適した表現形式が、表意的(denotative)、平叙的(declarative)、表示的(descriptive)とよばれるものである。

— denotative、declarative、descriptive という3つのことばは、プログラミング言語に関して用いられる限り、ほとんど区別して考える必要はないよう

ある。

— imperative な文と descriptive な文との違いは、前者では文の順序が変わると実行結果が異なってくるのに対して、後者では文の順序は実行結果には影響しないところにある。また、descriptive な文に対しては真／偽が定まるのに対して、imperative な文にはそういう性質はない。

(2) "imperative" 対 "functional / applicative"

— functional も applicative も、ともに '関数型' あるいは '関数的' と訳されている。

— applicative ということばの厳密な意味は、1.C項に説明したとおりである。

— applicative に対比させる意味で imperative ということばが用いられるときは、imperative  $\equiv$  non-applicative と解釈すべきである。

— applicative な考え方のひとつの純粋な形が、"関数" という考え方である。その意味で functional と applicative とはほとんど区別なしに用いられることが多い。しかし、Backusなどは区別して考えているように見受けられる。

— functional については、つぎのような注意がある：

"It follows that functional programming has little to do with functional notation. It is a trivial and pointless task to rearrange some piece of symbolism into prefixed operators and heavy bracketing. It is an intellectually demanding activity to characterize some physical or logical system as a set of entities and functional relations among them."

( P. J. Landin )

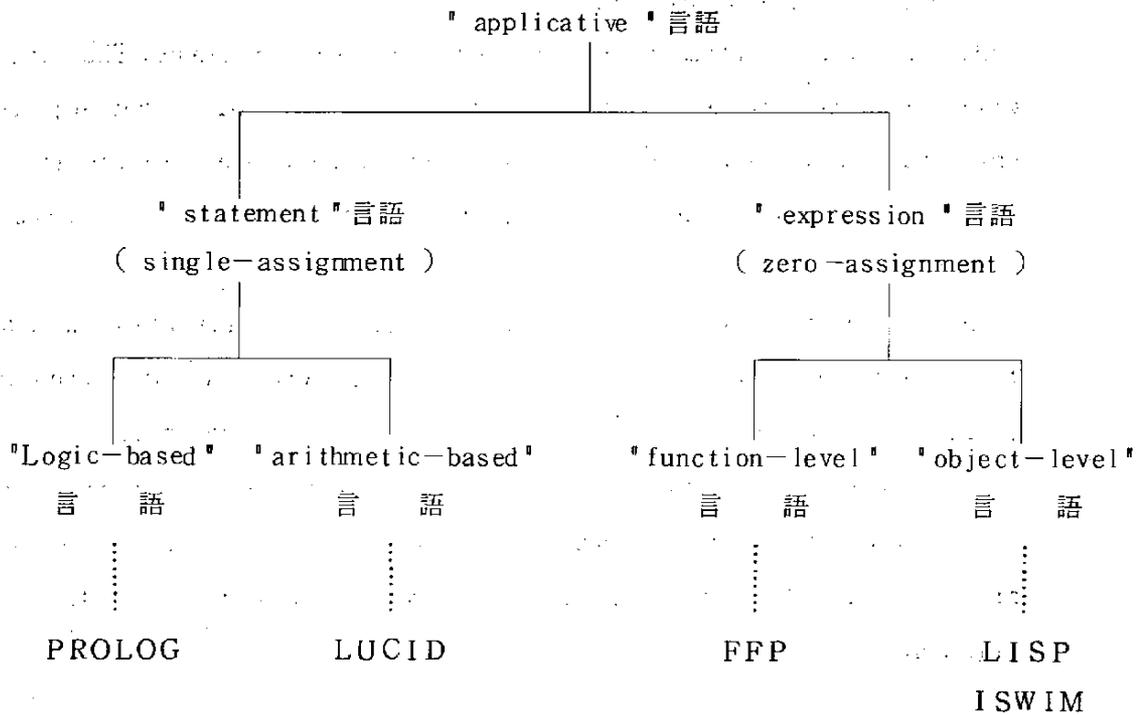
(3) "algorithmic" 対 "heuristic"

— 手続き型言語で書かれたプログラムを実行する場合には、システムのなすべき仕事は、ただプログラムをステップ・バイ・ステップに追っていくだけである。すなわち、プログラムそのものが algorithmic になっている。

— 非手続き型言語で書かれたプログラムでは、問題を解くのに必要な情報が記述されているだけで、問題の解き方は示されていない。問題の解法を発見するのはシステムの責任である。その意味で非手続き型のプログラムは heuristic であるといえる。

— 1.B②項に述べた ALGORITHM = LOGIC + CONTROL という定式化に従って説明すれば、CONTROL 部分をシステムの発見的能力に委ねるものが非手続き型言語である。

9.2.3 非手続き型言語の分類



— 非手続き型言語の総合的な分類が試みられた例があるかどうかは不明である。

— ここでは非手続き型としての純度の高い言語のみを対象とする分類を試みる。すなわち、' applicative ' であることが条件となる。

— 分類の狙いは、非手続き型言語の主なバリエーションを明らかにすることである。

— 第一のバリエーションは、表現スタイルにもとづくものである。statement の右辺には expression が許されるので、この区別は statement 対 expression という対立に本質があるのではなくて、statement を許すものと、許さないものとの対立というところにポイントがあることになる。

— statement を許さないことのひとつの狙いは、副作用の原因となる assignment statement や goto statement を追放することにある。' zero-assignment ' というのはそのことを意味するものである。

— assignment statement を許す場合にも、副作用を避けるために ' single-assignment ' の条件は不可欠である。

— ' logic-based '、' arithmetic-based ' の意味は明らかである。しかし、' arithmetic-based ' といういい方は、ここでの発明であり、これでよいかどうかは保証の限りではない。

— "function-level" と "object-level" の区別は、John Backus によるものである。6) その説明はつぎの通り：

Most programs written today are "object-level" programs. That is, programs describe how to combine various "objects" to form other objects until the final "result objects" have been formed. New objects are constructed from existing ones by the application of various object-to-object functions.

In the function level style a program is built directly from the programs that are given at the outset, by combining them with program-forming operations (PFOs). Thus instead of applying the given programs to objects to form a succession of objects culminating in the result object, the function level approach applies program-forming operations to the given programs to form a succession of programs culminating in the desired program.

#### 9.2.4 非手続き性の尺度 7)

常識的な立場での「非手続き型」の概念では、ある言語が非手続き型といえるかどうかは比較の問題であり、明確な判定基準は期待しえないことになる。このために、同じ言語に対して、それを非手続き型とみるひととそうでないひとが出ることになる。例えば、データベース問合せ言語 SQL の場合には、設計者はそれを非手続き型 (descriptive) としているのに、別のひとは手続き型としてみている。

このことは常識的な立場での考え方をつきつめると、「非手続き性」というのは、ある物質の純度のように、各言語に連続量として分布するものとみる見方が背景にあることを示している。

この見方を徹底すれば、「非手続き性の尺度」といったものを定義して、各言語の非手続き性が客観的に比較することができないかということになる。

このような試みとして、つぎのようなものがある。これは非手続き性をみる代わりに、逆に「手続き性」を評価するようになっている。すなわち、「手続き性の尺度」PMはつぎのように定義される。

$$PM = \frac{\text{変数結合の数}}{\text{変数結合の実行の仕方 (実行順序) が何通り可能かの数}} + \frac{\text{オペレーションの数}}{\text{オペレーションの実行の仕方 (実行順序) が何通り可能かの数}}$$

この尺度は、実行順序が自由なほど非手続き性が高いとみる考え方を数値化したものである。いくつかの典型的な問題について、プログラムを書き、それに上式を適用してPMを出し、それらの平均値をその言語のPMとする。

スケジューリング問合せの場合のPM値の例を下記に示す：

言 語	PM 値
COBOL DBTG	6 6.0
Date の代数	2 0.0
TABLET	1 4.0
DSL ALPHA	5.0
SQL	3.5

#### 9.2.5 非手続き型言語の利点と欠点

##### A. 'declarative' な表現と 'procedural' な表現の利害得失<sup>8)</sup>

評 価 項 目	'declarative' な表現	'procedural' な表現
表現の融通性 (したがって経済性)	○：①ひとつの事実はひとつの表現ですむ。 ②使い方が分からなくても困らない。	×：①知識の使い方がことなるとして別々の表現が必要。 ②使い途が分からないと表現のしかたが決まらない。
表現の分かり易さ (理解性) (学習性)	○：①単純な表現ですむ。 ②各表現の独立性が高い。	×：①まわりくどい表現になる。 ②互に影響し合うことが多い。 ③分割的に表現することが難しいことが多い。
変更の容易さ	○：(同上)	×：(同上)
扱い易さ (accessibility) (communicability)	○：①知っていることを表現するのに最も楽なやり方である。 ②自然言語は主に declarative である。	×：
記述力 (modeling 2nd order knowledge) (heuristic " )	×：①操作的な表現に不向き。 ②知識の使い方の知識(第2階の知識)の表現に不向き。 ③ 'heuristic' な知識の表現に不向き。	○：①手続的にしか表現しにくいことがある。 ②人工知能分野での十分な実績により記述力が保証されている。

B. 非手続き型言語の利害得失

評 価 項 目	非手続き的言語	手続き的言語
モジュラー・プログラミング	○	×
部分間のインタラクション	×	○
プログラミングの容易性	○	×
プログラムの実行効率	×	○
プログラムの変更容易性	○	×
高度なプログラミング	△	○
プログラムの検証性	○	×
プログラミング言語の高水準化	○	△
プログラム独立性	○	×
非決定性プログラミング	○	×
並行処理プログラミング	○	×
エンドユーザ・プログラミング	○	×

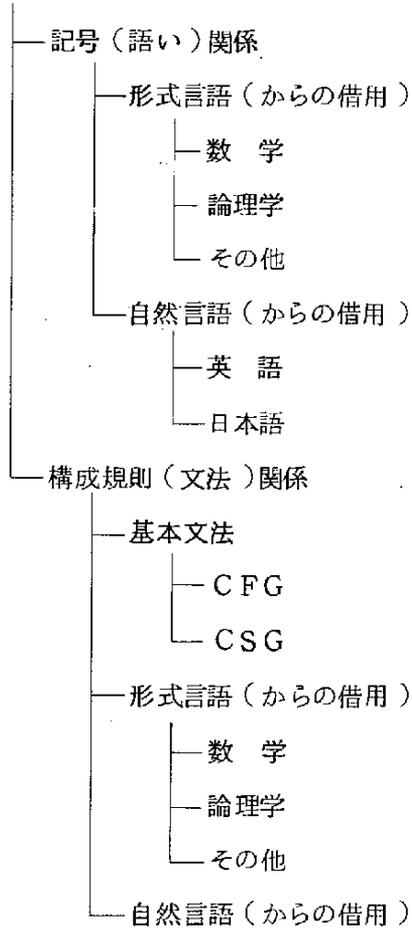
### 9.3 プログラミング言語の一覧

プログラミング言語は多種多様な意図にもとづいて設計されているので、それらを一元的な指標の下に分類することは不可能である。ここでは、機能的な観点を中心にして、最近のプログラミング言語の動向が浮彫りにできそうな特徴項目を選んで、個々の言語の弁別を行うことにした。

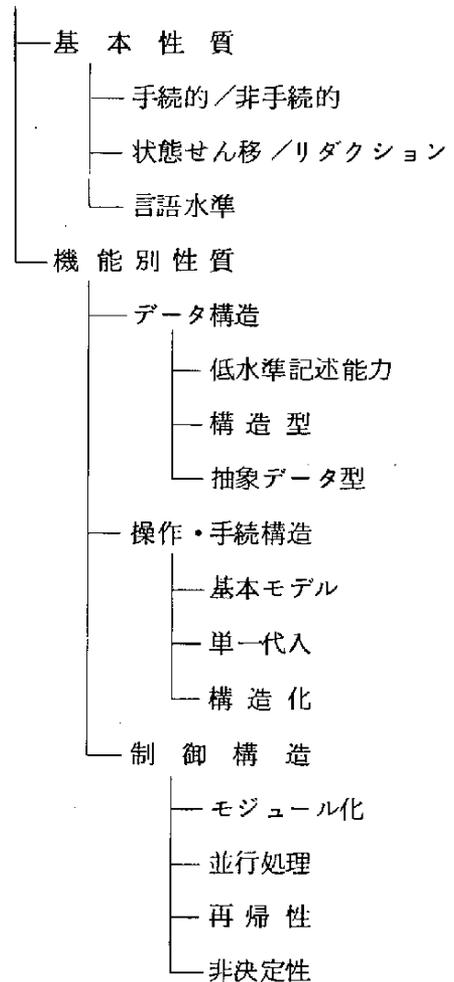
まず、通例にしたがって、シンタックスとセマンティクスに分け、シンタックスについては、自然言語、理論言語指向の側面に照明を当てることを意図した。また、セマンティクスについては最近の議論での焦点を拾い集めてある。

(分類作業は未完であり、表は不完全である。)

#### シンタックス面での分類項目



#### セマンティクス面での分類項目



プログラミング言語の分類

言語名	開発者	開発年	主要用途	記号(語い)関係		
				形式言語		
				数学	論理学	その他
FORTRAN	IBM (Backus)	<input type="checkbox"/> 1954 <input type="checkbox"/> 68	科学技術	演算記号		
ALGOL	Naur ほか	<input type="checkbox"/> 1958 <input type="checkbox"/> 65	"	"		専用記号
COBOL	CODASYL	<input type="checkbox"/> 1959 <input type="checkbox"/> 72	事務計算			
LISP	McCarthy, J.	<input type="checkbox"/> 1960	記号処理	—	—	専用記号
RPG	IBM	<input type="checkbox"/> 1960	事務計算			
APL	Iverson (IBM)	<input type="checkbox"/> 1962	汎用	演算記号		専用記号
SNOBOL	Farber ほか	<input type="checkbox"/> 1964	記号処理			
PL/1	IBM	<input type="checkbox"/> 1964 <input type="checkbox"/> 78	汎用	演算記号		
BASIC	ダートマス大	<input type="checkbox"/> 1964	科学技術	"		
SIMULA67	Dahl ほか	<input type="checkbox"/> 1967	シミュレーション			
PASCAL	Wirth	<input type="checkbox"/> 1968 <input type="checkbox"/> 80	汎用	演算記号		専用記号
ALGOL68	IFIP (Wijngaarden)	<input type="checkbox"/> 1968	科学技術	"	—	"
XPL	スタンフォード大	<input type="checkbox"/> 1970	システム記述			
BCPL	Richards	<input type="checkbox"/> 1969	"	演算記号	演算記号	専用記号
smalltalk	Xerox	<input type="checkbox"/> 1972	教育・学習	"	"	"
DFPL	Kosinski (IBM)	<input type="checkbox"/> 1973	科学技術	演算記号		図形
PL/S	IBM	<input type="checkbox"/> 1974	システム記述	"		
C	ベル研	<input type="checkbox"/> 1974	"	"	演算記号	専用記号
PROLOG	マルセーユ大 (Battani)	<input type="checkbox"/> 1973	記号処理	—	—	—
Clu	MIT (Liskov)	<input type="checkbox"/> 1974	汎用			
concurrent PASCAL	Brinch Hansen	<input type="checkbox"/> 1975	"	演算記号		
Mesa	Xerox	<input type="checkbox"/> 1975	システム記述			
Alphard	CMU	<input type="checkbox"/> 1976	汎用			
LUCID	Ashcroft ほか	<input type="checkbox"/> 1976	"	演算記号		
Euclid	Lampson ほか	<input type="checkbox"/> 1977	"			
CAJOLE	ロンドン大	<input type="checkbox"/> 1978	"	演算記号		
VAL	MIT (Dennis)	<input type="checkbox"/> 1978	"			
"Reduction 言語"	Berkling	<input type="checkbox"/> 1975	"	—	—	専用記号
ID	UCI (Arvind)	<input type="checkbox"/> 1979	"	演算記号	—	"
FFP	Backus (IBM)	<input type="checkbox"/> 1978	"	"	演算・値記号	専用記号
Ada	DoD	<input type="checkbox"/> 1979	"	演算記号	—	"

: 提案レベル  
 : 商用化レベル

( シンタックスによる )

構 成 規 則 ( 文 法 ) 関 係

自 然 言 語	基 本文 法	形 式 言 語			自 然 言 語	構 造 性
	CFG/CSG	数 学	論 理 学	そ の 他		
単語, 文字, 記号		代数式形式	論理式形式		命令文形式	弱
"		"	"		"	強
単語					"	
文字, 特殊記号		関数形式	$\lambda$ 形式	条件式	"	
単語				表形式		
		代数式形式	Combinator			
		代数式形式			命令文形式	強
単語, 文字, 記号		"			"	弱
		"				
単語, 文字, 記号		"	論理式形式		命令文形式	
"		"	"		"	
		"			"	
単語, 文字, 記号		"	論理式形式		"	
単語, 記号		"	"		"	
単語, 文字, 記号				フローチャート	"	
"					"	
"		代数式形式	論理式形式		"	
文字, 特殊記号		関数形式	関係形式		—	弱
					命令文形式	
単語, 文字, 記号		代数式形式	論理式形式		"	
					"	
					"	
単語, 文字, 記号		代数式形式			—	
					命令文形式	
単語, 文字, 記号		代数式形式	(条件式)	ガード付コマンド	"	
					"	
文字			Combinator		—	
単語, 文字, 記号		代数式形式			命令文形式	
文字, 特殊記号		"	論理式形式	条件式	—	
単語, 文字, 記号		"	"		命令文形式	

プログラミング言語の分類

言語名	開発者	基本性質			データ構造		
		手続き的	状態せん移	言語水準	データ構造		
		非手続き的	リダクション		低水準記述	構造型	抽象データ型
FORTRAN	IBM ( Backus )	手続き的	状態せん移	高水準	×	×	×
ALGOL	Naur ほか	"	"	"	×	×	×
COBOL	CODASYL	"	"	"	×	○	×
LISP	McCarthy, J.	非手続き的	リダクション	"	×	○	×
RPG	IBM	手続き的	状態せん移	"	×	—	×
APL	Iverson ( IBM )	"	"	超高水準	×	○	○
SNOBOL	Farber ほか	"	"	高水準	×	○	×
PL/1	IBM	"	"	"	○	○	×
BASIC	ダートマス大	"	"	"	×	×	×
SIMULA 67	Dahl ほか	"	"	"	×	○	○
PASCAL	Wirth, Jensen	"	"	"	○	○	×
ALGOL 68	IFIP (Wijngaarden)	"	"	"	×	○	×
XPL	スタンフォード大	"	"	"	○		×
BCPL	Richards	"	"	"	○		×
smalltalk	Xerox (Alan Key)	"	"	超高水準	○	○	○
DFPL	Kosinski ( IBM )	"	"	高水準			
PL/S	IBM	"	"	"	○		×
C	ベル研	"	"	"	○		×
PROLOG	マルセーユ大 ( Battani )	非手続き的	リダクション	超高水準	×	○	○
Clu	MIT ( Liskov )	手続き的	"	高水準	×		○
concurrent PASCAL	Brinch Hansen	"	"	"	○	○	○
Mesa	Xerox	"	"	"	○	○	○
Alphard	CMU	"	"	"			○
LUCID	Ashcroft ほか	非手続き的	"	"			×
Euclid	Lampson ほか	手続き的	"	"	○		○
CAJOL	ロンドン大	非手続き的	"	"			○
VAL	MIT ( Dennis )		"	"		○	×
'Reduction 言語'	Berkling	非手続き的	リダクション	超高水準	×		○
ID	UCI ( Arvind )	手続き的	状態せん移	高水準		○	○
FFP	Backus ( IBM )	非手続き的	リダクション	超高水準	×	○	○
Ada	DoD	手続き的	状態せん移	高水準	○	○	○

(セマンティクスによる)

機 能 別 性 質							
操 作 ・ 手 続 構 造				制 御 構 造			
基本モデル	単一代入	構 造 化	モジュール化	並 行 処 理	再 帰 性	非 決 定 性	
代 数 式	×	フラット	×	不 可	×	×	
代 数 式	×	ブロック構造	×	"	○	×	
帳簿処理	×	フラット	×	"	×	×	
$\lambda$ -Calculus	×	"	×	可	○	×	
帳簿処理	×	—		不 可	×	×	
Combinator	×	フラット	×	"	×	×	
構文解析	×	"	×	"		×	
代数式+ $\alpha$	×	ブロック構造	×	"	○	×	
代 数 式	×	フラット	×	"	×	×	
代数式+ $\alpha$	×	ブロック構造	×	"		×	
代数式+ $\alpha$	×	"	×	"	○	×	
代 数 式	×	"	×	可	○	×	
代数式+ $\alpha$	×		×	"		×	
代数式+ $\alpha$	×		×	"		×	
Actor理論	—	ブロック構造	○	○			
ペトリネット	×		×	(データフロー)	○	○	
代数式+ $\alpha$	×		×	"		×	
代数式+ $\alpha$	×	ブロック構造	×	"	○	×	
述 語 論 理	—	フラット	○	可		○	
代数式+ $\alpha$	×		×			×	
代数式+ $\alpha$	×		×	monitor	○	×	
代数式+ $\alpha$	×		○	可		×	
代数式+ $\alpha$	×		×			×	
代数式+ $\alpha$	○	ブロック構造	×		○	×	
関 数	×		×			×	
関 数	○		×	(データフロー)	○	×	
代数式+ $\alpha$	○		×	( " )	○	○	
Combinator	—	フラット	○			×	
関 数	○	ブロック構造	×	(データフロー)		○	
Combinator	—	フラット	○	可		×	
代数式+ $\alpha$	×	ブロック構造	○	rendezvous		×	

参 考 文 献

- 1) 横井俊夫：プログラミング言語の新たな潮流，日経エレクトロニクス別冊「コンピュータ」1979-80，1980.
- 2) SAMMET, J. E. : Overview of High-Level Languages, Advances in Computers 81, 1981.
- 3) Heuristic Programming Project 1980.
- 4) 日本ユニパック総合研究所編：共立総合コンピュータ辞典，1976.
- 5) SAMMET, J. E. : Overview of High-Level Languages, Advances in Computers 81, 1981.
- 6) LANDIN, P. J. : The Next 700 Programming Languages, CACM, Vol. 9, No. 3, 1966.
- 7) KOWALSKI, R. A. : Algorithm = Logic + Control, CACM, Vol. 22, No. 7, 1979.
- 8) FRIEDMAN, D. P. & WISE, D. S. : Aspects of Applicative Programming for Parallel Processing, IEEE trans. on Computers Vol. C/27, No. 4, 1978.
- 9) BACKUS, J. : Functional Level Program as Mathematical Objects, Proceedings of the 1981 Conference on Functional Programming Languages and Computer Architecture, 1981.
- 10) WELTY, C. & STEMPLE, D. W. : Human Factors Comparison of a Procedural and a Nonprocedural Query Language, ACM-TODS, Vol. 6, No. 4, 1981.
- 11) WINOGRAD, T. : Frame Representations and the Declarative / Procedural Controversy, Representation and Understanding (Bobrow, D. G., and Collins, A. (Eds.)), 1975.

## 10. ソフトウェア生産性向上の方策

### 10.1 ソフトウェアの生産性について

#### 10.1.1 ソフトウェア生産性問題の背景（ソフトウェア危機）

- (1) システム開発総コスト中に占めるハードウェア・コストとソフトウェア・コストの構成比の推移から、情報システムのコスト・パフォーマンスの効率化には、ソフトウェア面での改善が決定的な要因となりつつある。（図10-1）
- (2) 今後予測されるソフトウェアの需要の伸びに対して、現在のソフトウェアの生産性の伸び率のままでは十分な対応ができない。（図10-2）
- (3) システムの大型化は、「スクラップ・アンド・ビルド」といったような、いままでのソフトウェア開発のあり方ではやっていけない規模になってきている。
- (4) ソフトウェアの生産が、手工業的・労働集約的であることに対して、新しい生産手段の開発が必要である。

図10-1 ハードウェア・コスト対ソフトウェア・コストの変化

出典：B.W.Boehm "Software Engineering"

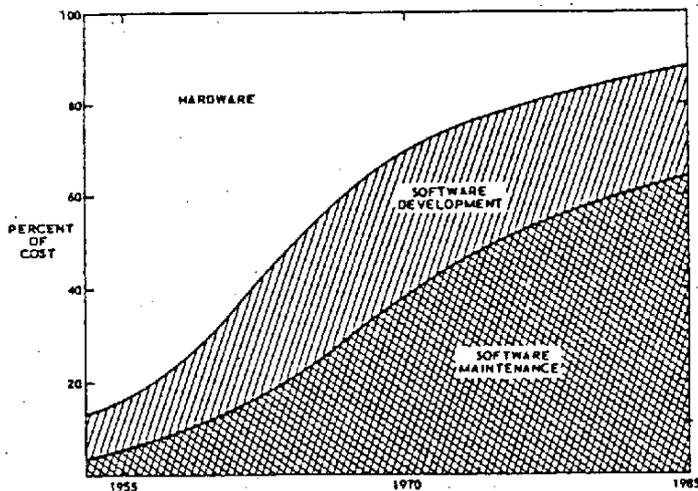
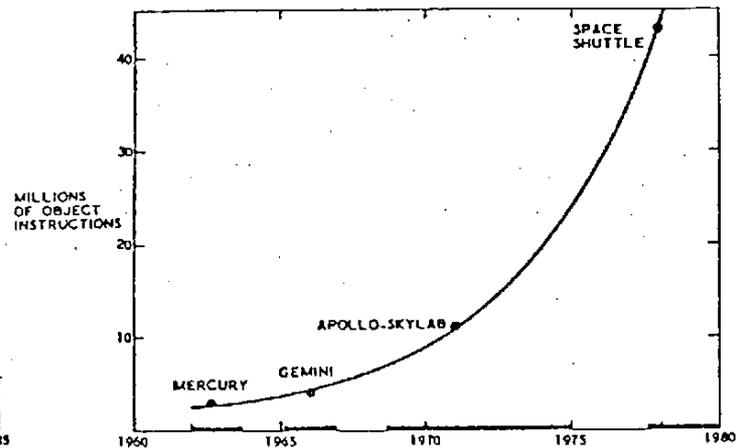


図10-2 ソフトウェア需要の伸び（宇宙計画用ソフトウェアの場合）

出典：COMPUTERWORLD, January 18, 1982



#### 10.1.2 ソフトウェア生産性の定義、評価

##### (1) ソフトウェアの生産性

$$\text{生産に費やされるものの生産上の効果} = \frac{\text{ソフトウェア生産量}}{\text{ソフトウェア生産への投入}}$$

ソフトウェア生産物：プログラム、文書、テストケース、修正バグ件数など

投入：工数、期間、費用、計算機時間など

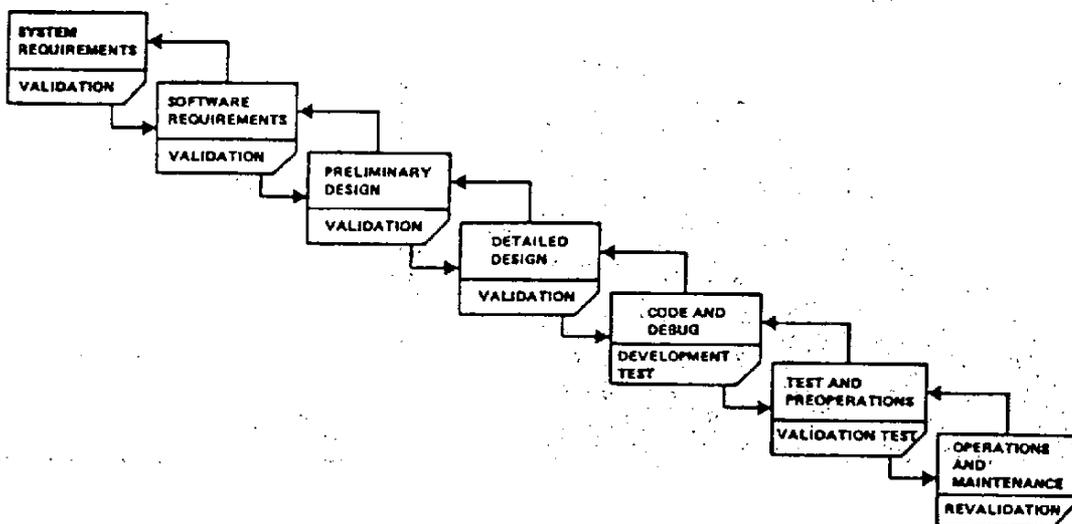
(2) ソフトウェア生産性評価の難しさ

- ・ ソフトウェア生産量の定量的表現
- ・ ソフトウェア生産性と品質の関係
- ・ ソフトウェア生産性と性能の関係
- ・ 影響要因の複雑さ：①プログラムの規模、種類、複雑さ、②記述言語、③開発環境、④生産技術と開発システム、⑤ドキュメンテーション、⑥人間要素などの複雑な組合わせである。

10.1.3 ソフトウェアのライフサイクル

- (1) ソフトウェアのライフサイクルとは、ソフトウェアの開発の要求が発生してから、そのソフトウェアが開発、運用、保守され、最終的に廃却されるまでの期間をさしている。
- (2) ソフトウェアの生産性を、ライフサイクルの観点から評価する。(ソフトウェア・ライフサイクル・コストを最小にするという考え方)

図10-3 Software Life Cycle



10.1.4 ソフトウェア生産性向上について

- (1) ソフトウェア生産への影響要因の多さからも分るように、いろいろなアプローチの仕方があるが、決定的な方策はない。
- (2) エンドユーザ機能、生産性向上ツール、対話処理機能などの導入により、設計工数の削減、プログラミングの容易性、開発期間の短縮、機能拡張の容易性、保守の容易性などの生産性上の効果が得られる。

- (3) 生産性向上ツールには、ソフトウェア・ツールのほかにも、各種技法、各種記述言語、方法論など数多い。さらに計算機アーキテクチャの高度化によるセマンティック・ギャップの縮小、オンライン検査機能も大層有用である。
- (4) 10.4 に、最近徐々に発展しつつある要求/設計仕様技術のうちから、代表的なものの一覧表を付した。

## 10.2 第5世代のプログラミング環境と生産性

### 10.2.1 知的プログラミング・システム

- (1) 第5世代プロジェクトでは、その主要なテーマの一つとして知的プログラミング・システムが実現される。これは、知識情報処理システム開発のための高度な支援システムでもあり、またそれ自身が知識情報処理システム（ソフトウェア開発コンサルティングシステム）として実現されるものである。
- (2) 知的プログラミング・システムによって、非専門家でも容易にコンピュータが利用できる（例えば自然言語での対話によるプログラミング）ようになり、専門家や研究者にとっても、より複雑なシステムの開発が容易になる。
- (3) 知的プログラミングシステムは、その技術の延長上に自動プログラミング・システムの実現を見込んでいる。
- (4) 知的プログラミング・システムの構成

#### プログラム自動合成システム

仕様記述言語

検証システム

モジュール組立による合成システム

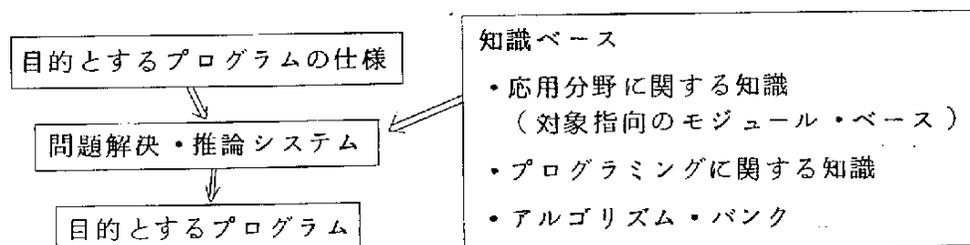
#### 支援ソフトウェア・システム

管理ソフトウェア・システム（ネットワーク管理を含む）

支援ユーティリティ・システム（エディタ、デバッガ、言語変換ソフト、システム解説用ソフト、保守用ソフト等）

### 10.2.2 プログラム自動合成システム

#### (1) 合成の仕組



(2) プログラム自動合成のための基礎技術

機能モジュールと、それに基づく組合せ技術（水平型組合せ技術）

データ抽象化によるモジュール化と、それに基づく組合せ技術

（垂直型組合せ技術）

10.2.3 知的プログラミング・システムと生産性

(1) 知的プログラミング・システムは従来のソフトウェア技術を根本から作り直すツールとなる。

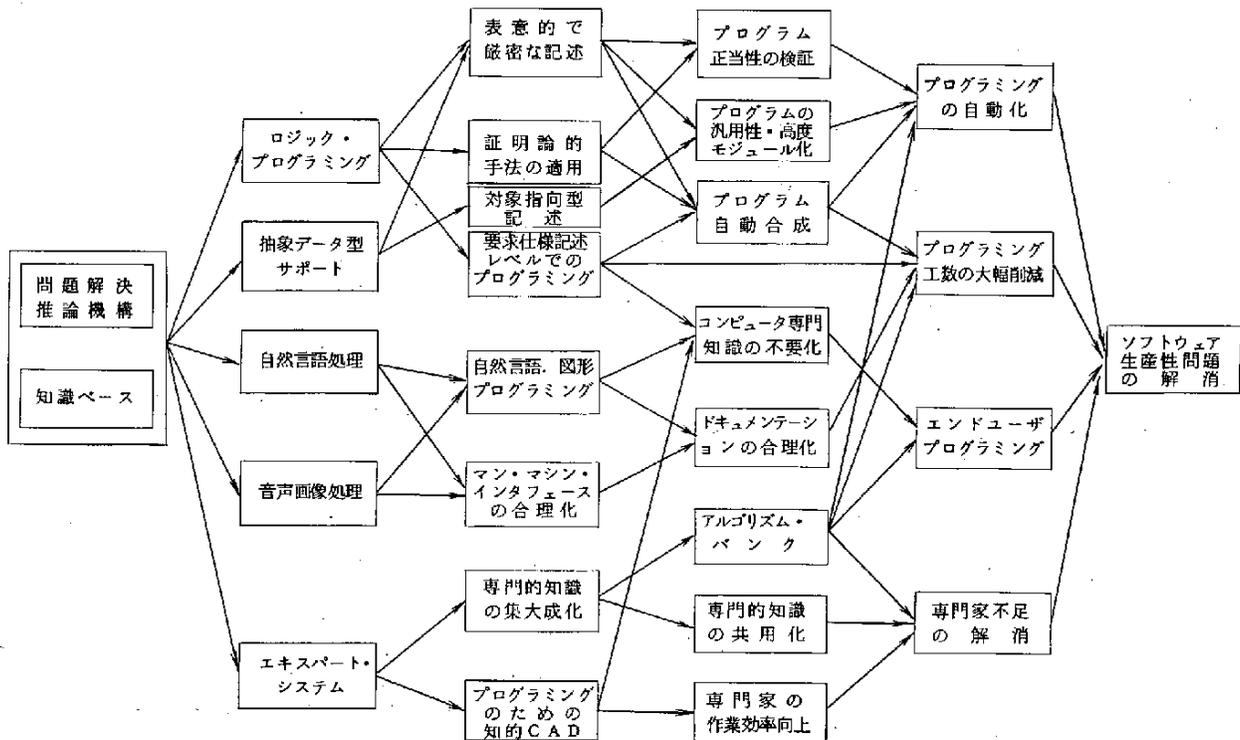
(2) (1)によって、現在かかえているソフトウェアの問題の多く、たとえば、

- ・ プログラミングの困難さ
- ・ プログラミングの低品質
- ・ ソフトウェアの非柔軟性
- ・ 大型ソフトウェアの保守の困難さ

など、いわゆるセマンティック・ギャップに起因する難点の解消が期待される。

(3) 第5世代コンピュータ・システムのもつ機能を、ソフトウェア生産性の観点から展開すると下図のようになる。

図10-4 第5世代コンピュータ・システムとソフトウェア生産性



	~ 1960	1970	1980	1990
時代 (半導体)  生産性  IBM	第2世代 (トランジスタ)  高級言語  *360	第3世代 (IC)  システム記述言語  *370	第3.5世代 (LSI)  要求記述言語 アンバンドリング  *3033	第4世代 (超LSI)  超高級言語 ソフトウェア自動作成  <div style="border: 1px solid black; padding: 2px; display: inline-block;">第5世代プロジェクト</div>
言語 プログラム言語  (言語理論) (エンドユーザ言語)	機械語 アセンブラ  *APL *FORTRAN *ALGOL *LISP *COBOL *BASIC  *PL/I	*PASCAL  *PROLOG *PLANNER *CLU *CONIVER *C  *Mesa *VAL  *LUCID *QLISP *QBE  *Ada	*Iota  (論理型、関数型)  (自然言語化、図形等の利用)	(知的プログラミングシステム)
仕様記述技術	フローチャート デシジョンテーブル		*HIPO  *SADT  *ISDOS *SREM	
設計技術	機能モジュール マクロによる特殊化	構造化プログラム	*ワーニエ法 *モジュール仕様化技術  *複合化設計法  *構造化設計法	アルゴリズム・バンク
ソフト生産環境 (TSS)		*MULTICS	*UNIX	*パーソナルマシン(MIT Lispマシン)  *ALTO *スーパー・パーソナルマシン
その他 検証、テスト データ管理 マンマシン 保守、ソフト管理 プロジェクト管理		*チーププログラマチーム  *構造化ワークスルー		検証理論 プログラムの自動合成 プログラムの保守、改良、管理 の体系化

10.4 代表的な要求/設計仕様化技術

方法論	中心開発者	特性	現状	主要概念	参照文献
セマンティックネット	Quillian	C	研究中	ネットワークによる意味論的表現	Sim 73
	Zadeh	C		ファジー概念や複合システムを扱うより詳細な概念	Zad 71
	Fitzwater	C	研究中	形式的分解 相互作用の基本要素	Fit 76
	Langefors	C	不明		Lan 63
RGF		C			
IA	Willson	M-コンピュータ化	研究中	設計内容の意味論的表現	Wil 75
	Becker	M-手作業 コンピュータによる支援	実験中	設計内容の意味論的表現 ファジー概念	Bec 73
	Wymore	M-手作業	実験中	入出力の軌道 システム仕様機能	Wym 76
HOS	Hamilton, Zeldin	MC-コンピュータ化	研究中	システム公理 分解 仕様記述言語 AXES	Ham 76 a, b, c
SADT	Ross	MC-手作業	運用中	構造化設計 グラフ表記法	Ros 77
F <sup>2</sup> D <sup>2</sup>	RCA	M-手作業	運用中		RCA 72
デシジョンテーブル		M	運用中		Lon 72
	Young, Kent	M	運用中		You 58
システムティクス	Grindley	M	不明		Gri 66
ADS	NCR	M	運用中	定形の用紙使用	Lyn 69
TAG	IBM	M-コンピュータ支援	運用中		IBM 71
HIPO	IBM	M-手作業	運用中		IBM 73 Sta 76
ISDOS	Teichroew	M-コンピュータ支援	運用中	PSL/PSA システム・データベース	Tei 74 b, 77
SODA	CWU	M	不明	設計の択一的な選択の評価法、設計の最適化	Nm 71
BDL	IBM	M-コンピュータ支援	開発中	利用者の仕様からシステムを直接にインプリメント	Gol 75
構造化設計	Constantine Myers Stevens	M-手作業	運用中	設計の指針、構造図、バブル図	Hug 75
LOGOS	Rose	M	不明	形式的グラフ・モデルと解析	Ros 72
CSC-Threads	CSC	M-手作業	運用中		CSC 73
SREM	TRW	M-コンピュータ支援	運用中	拡張RSL、R-ネット REVS-解析システム	Alf 76 a, b Bel 76 a, b
FSM	AFC	M-コンピュータ支援	研究中	システムの有限状態機械モデル	Sal 76
VG	CSC	M-コンピュータ支援	研究中	グラフ・モデル	Bel 76
GRC-ベトリ・ネット	GRC	M-コンピュータ支援	研究中	ベトリ・ネット・モデルと解析	Bel 76
	Booth他	M	研究中	計算構造、性能モデル	Sho 76
SARA	UCLA	M, L-コンピュータ支援			
STEPS	日本電気	M-コンピュータ支援			
UDDT	日立	M-コンピュータ支援			
SSD	富士通研究所	M-コンピュータ支援			
HDM	SRI	M, L-コンピュータ支援			
情報代数	CODASYL委員会	L	休止	ファイルを含んだ複雑なデータの操作形式言語基本概念	COD 62
	Ho, Nunamaker	L	設計中	複雑なデータ関係の形式的仕様	Ho 74
	Bridge Thompson	L-拡張	設計中	データ使用法の分類と宣言	Bri 74
	McGowan	L-手作業	運用中		McG 75
	Parnas	L-手作業	運用中	機能の影響仕様-実現との独立性	Par 72
	Good他	L-コンピュータによる 検証	研究中	形式的入出力仕様	Geo 75
	Hewitt Smith	L-コンピュータによる 検証	研究中	契約(仕様) メタ評価	Hew 75
V-Graph	Earley	L	研究中	データ型のグラフィック仕様	Ear 71
	Guttag他	L	研究中	データ特性の公理 形式的な検証	Gut 76

註 参照文献はBit 1978年8月臨時増刊(ソフトウェア工学)による。  
C:概念的方法 M:方法論 L:言語

# 11. コンピュータ・セキュリティ

## 11.1 基礎概念

### (1) コンピュータ・セキュリティの定義

コンピュータ・セキュリティとはコンピュータ・システムあるいはデータを不意の事故や悪意にもとづく変更、破壊、漏洩などから保護することである。

(COMPUTER SECURITY Hsiao)

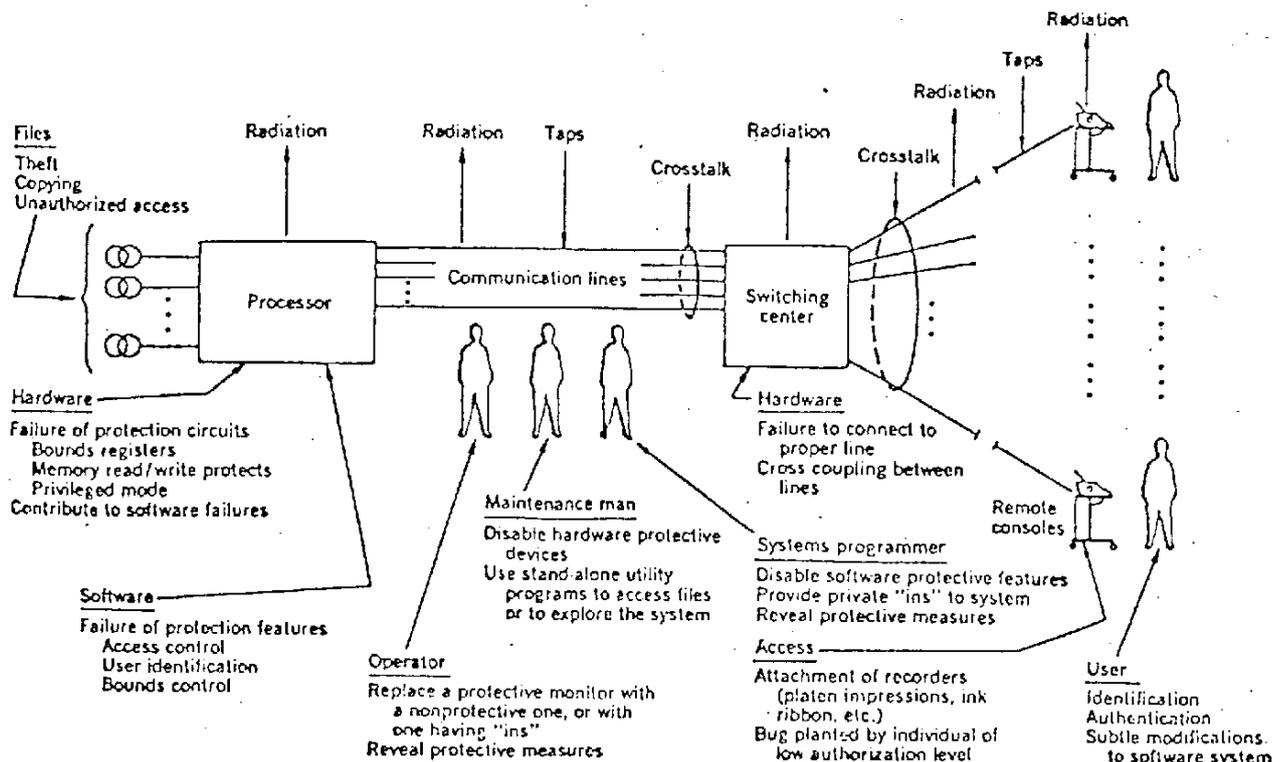
### (2) プライバシーの定義

プライバシーとは個人、団体または協会が彼らに関する情報を、いつ、いかなる方法で、そしてどの程度まで他人に伝達するかを自ら決定する権利である。

(連邦保健教育福祉省諮問委員会 1973年)

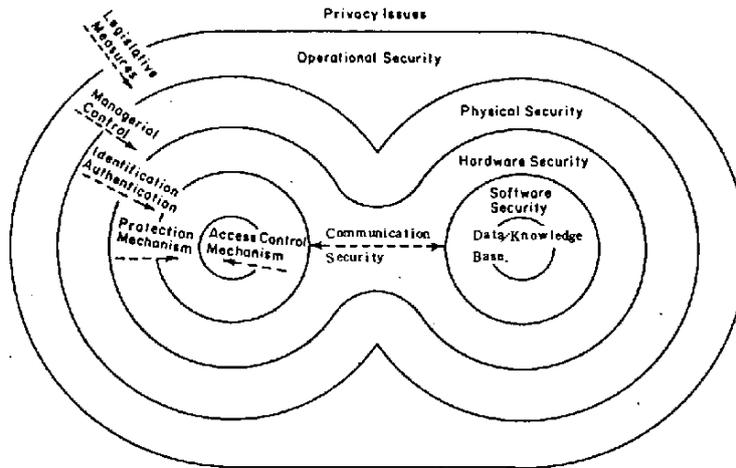
### (3) セキュリティの侵害要因

出典: Modeling Data Secure Systems, Edwin J. McCauley, ILL



## 11.2 セキュリティ手段の現状

### (1) プライバシー問題とセキュリティ手段の関係

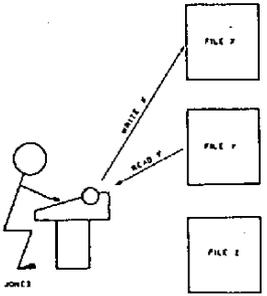
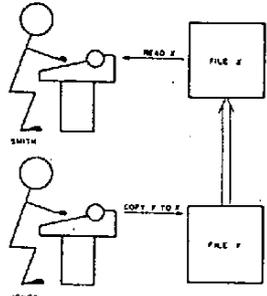
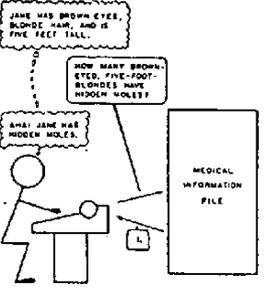
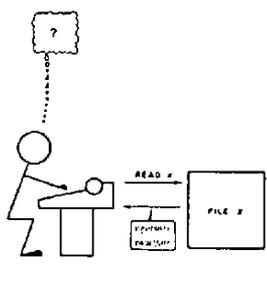


### (2) セキュリティ侵害要因別の対応策

セキュリティ侵害要因	対応策	セキュリティ侵害要因	対応策
Physical security Natural disaster	Site selection Detection Extinguishing Recovery	Software Security User access	Identification and authentication Logging Threat monitoring Access control matrix Capability list Access list Authority-item
Intruder	Boundary protection Perimeter protection Entrance protection Critical-area protection	Access control	Isolation Multiple space method Virtual machine monitor Verification Inductive assertion technique Kernel concept Penetration tests
Authentication and identification	Password "Key"	Propagation of security breach	
Electronic and electromagnetic tampering	Physical characteristics Encryption Intruder protection	Operating system flaws	
Hardware security Memory protection	Bounds registers Locks and keys Access control bits Virtual memory	Data/Knowledge Security Data errors Value-sensitive data	Access restrictions View mechanism Query modification Integrity Controls Prevention of inference Keeping track of the number and type of queries asked Cycle detection Directed graphs Partitioning Compartmentalization Security atom
Execution protection	Binary states Multiple states	History-sensitive data	Improve access precision Compartmentalization Security molecule Authorization hierarchies Access control procedures Resolution capability Secure database design methodology Predicates for control Database machines
I/O protection	Microprocessors Specialized processors	Context-dependent data Restrict actual data accessed	
Sequence protection	Periods processing Monitoring	Multilevel security Access authorization and resolution	
Communications security Short messages Long messages Distributed terminals Network communications Code breaking	Classical cryptography Modern cryptography Terminal interface cryptographic transformation boxes Network cryptographic controller  Based on key Standard transformation	Integrated security	

出典：Computer Security, David K. Hsiao, Computing Surveys, Vol. 11 No3 Sep. 1979

(3) コンピュータシステム内部における4種類のセキュリティ・コントロール

	問 題	解 説	関 係 概 念 等
ACCESS CONTROLS	 <p>FIGURE 1a. ACCESS. Jones can be permitted to read file Y and write in file X, he has no access to file Z.</p>	<p>コンピュータシステムへの無許可のアクセスを検知あるいは防御する一方、正当なアクセスを完全に確保するための方法。</p>	<p>Principle of Least Privilege (最小特権の原則)                  Object-dependent Controls [JONE76]                  Capability addressing [DENN76b, FABR74, LIND76, ORGA72, ORGA73, SALT75]                  Capability list, SWARD [MYER78]                  PSOS [NEUM77]                  Typed memory [GEHR79]                  Ring protection MULTICS [SCHR72]                  Domain protection [SCHR72, NEED72, NEED74]                  Trojan horses [LIND76]</p>
FLOW CONTROLS	 <p>FIGURE 1b. FLOW. Denied access to file Y, Smith gets confederate Jones to make a copy; flow controls could prevent this.</p>	<p>機密情報の横流し的な漏洩を防止する方法。                  (Yについてアクセス権のない者が、Yについてアクセス権のある者と共謀してYについての情報を移し取る様なものに対する方法。)</p>	<ul style="list-style-type: none"> <li>・Dearance (security class) SDC [WEIS69]                      MITRE [MILL76]                      Case Western Reserve Univ. [WALT75]                      SRI INT. [NEUM77]</li> <li>・プログラム・モジュールにおける入出力依存関係                      MITRE [MILL76]                      UCLA [POPE78c]                      SRI [NEUM77]</li> <li>・Data mark machine [FENT74]</li> </ul>
INFERENCE CONTROLS	 <p>FIGURE 1c. INFERENCE. A questioner used prior knowledge to deduce confidential information from a statistical summary; inference controls could prevent this.</p>	<p>間接的な機密情報に対し、その取得者が推論しても直接的機密情報が演繹あるいは帰納できないようにする方法。</p>	<ul style="list-style-type: none"> <li>・Tracker [SCHL75, SCHL79]</li> <li>・Threat monitoring. [HOFF70]</li> <li>・Partitioned database [YU77]</li> <li>・Rounding controls [SCHL77] [KARP70] [ACHU78]</li> <li>・Random sample [HANS71]</li> </ul>
CRYPTOGRAPHIC CONTROLS	 <p>FIGURE 1d. ENCRYPTION. Jones illicitly obtains a copy of file X; but his encrypted contents are meaningless to him.</p>	<p>暗号化による機密確保の方法。</p>	<ul style="list-style-type: none"> <li>・Data encryption standard (DES) [NBS77]</li> <li>・Public-key encryption [DIFF76]</li> </ul>

出典：Data Security, DOROTHY E. DENNING AND PETER J. DENNING

### 11.3 知識情報処理におけるセキュリティ

#### (1) データベースと知識ベースの違い

	データベース	知識ベース
内容	extensional data	extensional data intensional data
アクセス方法	reference by name	reference by name reference by description
表現形式	word(byte) oriented object oriented	logic procedural representation semantic networks production system direct representation semantic primitive frame and script

#### (2) 知識ベース・セキュリティの現状

		Reliability	Availability	Serviceability	Integrity	Security
研究例	DB	○	○	○	○	○
	KB	○	×	×	integrity constraints consistency checks	×
					truth maintenance	

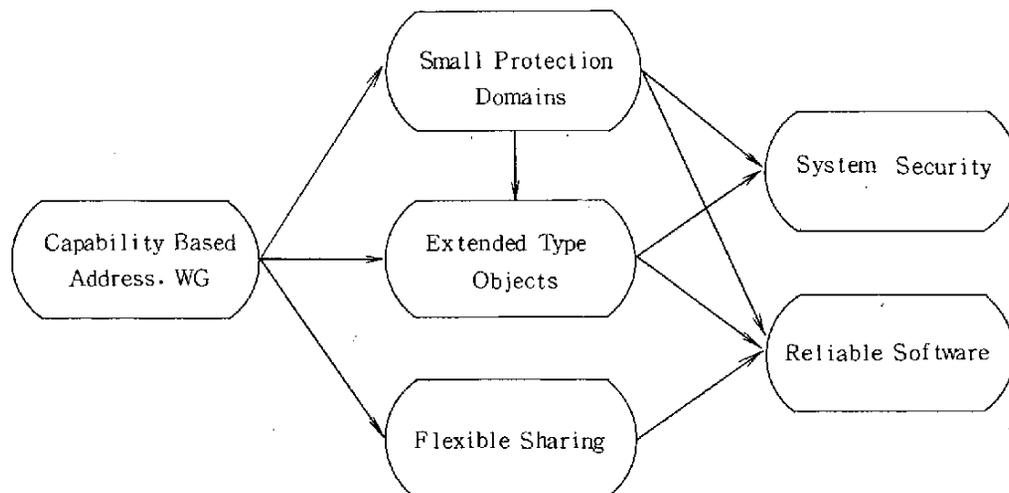
### (3) ACTOR 理論での例

PRIVACY and PROTECTION: Actors enable us to define effective and efficient protection schemes. Ordinary protection falls out as an efficient intrinsic property of actors. The protection is based on the concept of "use". Actors can be freely passed out since they will work only for actors which have the authority to use them. Mutually suspicious "memoryless" subsystems are easily and efficiently implemented. ACTORS are at least as powerful a protection mechanism as domains [Schroeder, Needham, etc.], access control lists [MULTICS], objects [Wulf], and capabilities [Dennis, Plummer, Lampson]. Because actors are locally computationally universal and cannot be coerced there is reason to believe that they are a universal protection mechanism in the sense that all other protection mechanisms can be efficiently defined using actors. The most important issues in privacy and protection that remain unsolved are those involving intent and trust. We are currently considering ways in which our model can be further developed to address these problems.

#### 11.4 技術相関図

細分化プロテクション・ドメインの概念と拡張データ型概念が特権行使権付のアドレス方式の概念下において構築されることはシステム・セキュリティと同時に高信頼性ソフトウェアに寄与する。

[ LINDEN, T.A. "Operating system structures to support security and reliable software" Comput. Surv. 8,4 (DEC, 1976), P.409-445 ]



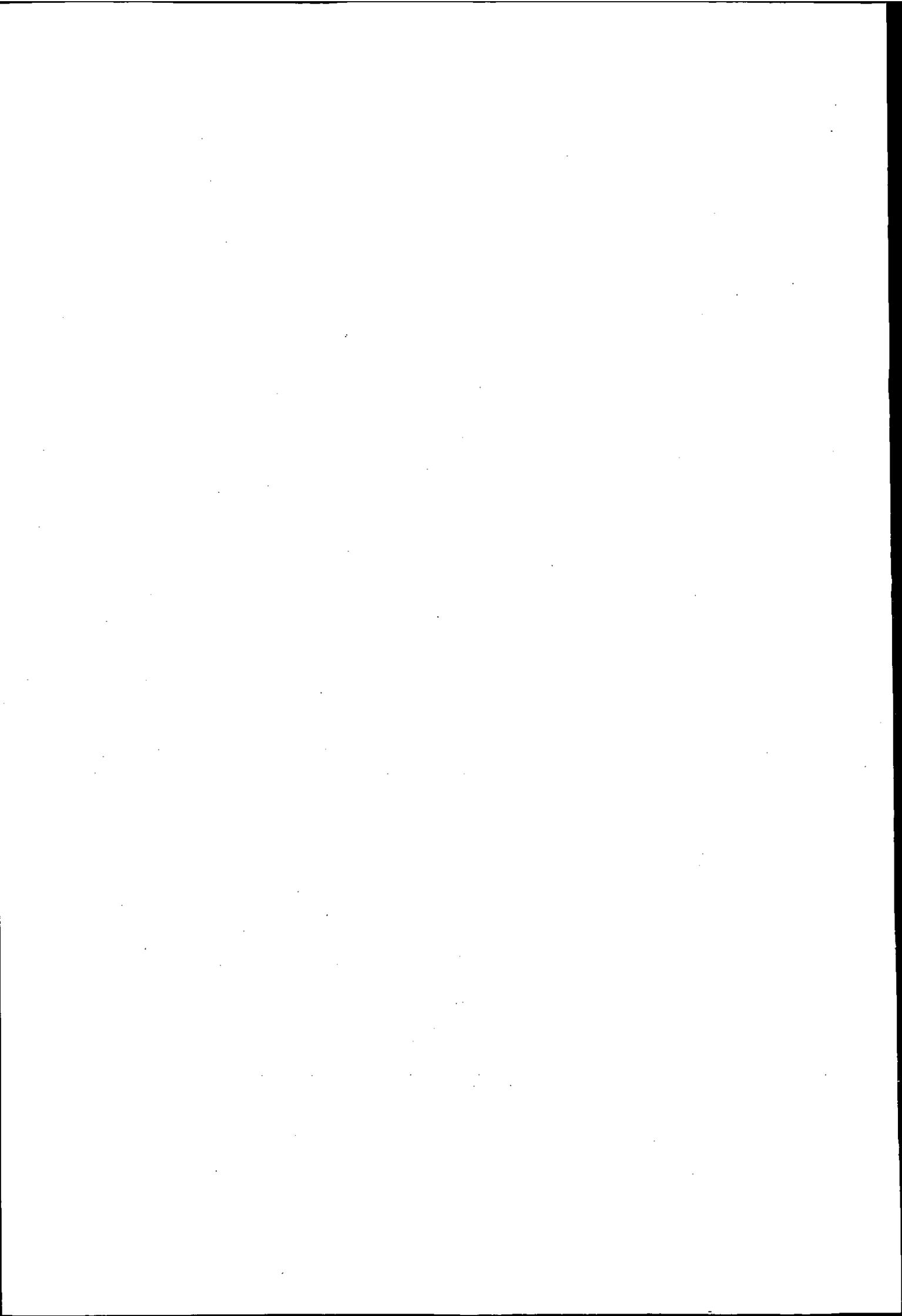


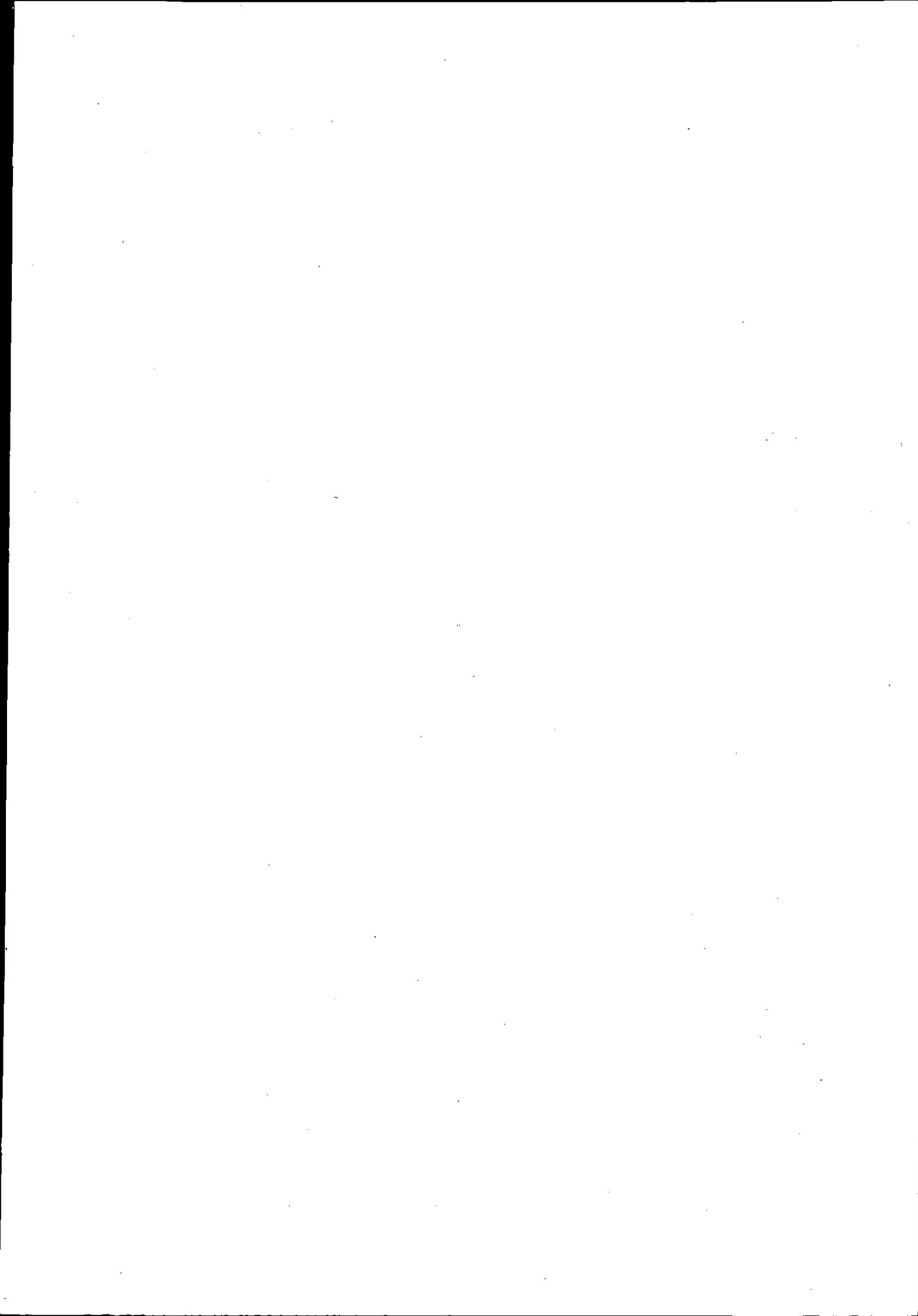
— 禁 無 断 転 載 —

昭和 57 年 2 月 発行

発行所 財団法人 日本情報処理開発協会  
東京都港区芝公園 3-5-8  
機械振興会館内  
TEL (434)8211 (大代表)

印刷所 日英舎印刷株式会社  
東京都中央区京橋 3-3-4  
TEL (281)3800~2





原本 (持出嚴禁)

受 付 No.	E-21
受付年月日	57. 4. 2
作 成 課	