■6 群（コンピューター基礎理論とハードウェア）－ 1 編（コンピュータの歴史）

2 章 技術的発展

（執筆者：児玉祐悦）［2009 年 5 月 受領］

■概要■

コンピュータはそれを構成する素子（デバイス）で分類すると，歯車で構成される機械式，継電器（リレー：電磁石に電流を流すと，その上に置かれたスイッチが機械的にオン／オフする）を使った電気機械式，真空管，トランジスタ，集積回路などを使った電子式に分類される．また，動作方式で分類するとディジタル方式とアナログ方式に分類される．アナログコンピュータは計算をある物理的な量に置き換えて計算する方式であり，機械式アナログコンピュータの代表例は計算尺である．電子式アナログコンピュータでは演算増幅器（オペアンプ．理想的なオペアンプでは，増幅率は無限大，入力抵抗は無限大，出力抵抗はゼロである．加算器や積分器を簡単に構成できる）を結線して回路を作り，微分方程式などを解くことができる．ディジタルコンピュータでは問題を解く手順（アルゴリズム）が与えられる．機械的ディジタルコンピュータの代表例は算盤（そろばん）であり，加算の手順が与えられ，算盤の上位の桁から桁上げを含めた操作がなされ，それが最下位桁までなされて，加算が終了する．本編 1 章では，機械式及び電気機械式のディジタルコンピュータについて述べた．以下，本章で述べるのは電子式でディジタル方式のコンピュータの歴史である．以後これを単にコンピュータと呼ぶことにする．本章では，これまで開発・利用されてきたコンピュータの技術について整理し，時代的な順序で記述する．

【本章の構成】

2-1 節で 1980 年以前のコンピュータについて，2-2 節で 1980 年以降のコンピュータについて，10 年ごとにその年代で開発された技術などを整理して記述している．本編 3 章で取り上げた歴史的意義が大きいコンピュータについては，そちらを参照することとし，概要についてのみ示した．

■6 群 － 1 編 － 2 章

2-1 1940 年代から 1970 年代まで

（執筆者：富田眞治）［2009 年 5 月 受領］

2-1-1 1940 年代

(1) コンピュータの第 1 世代

コンピュータをその使用されている電子デバイスに応じて，真空管を利用した第 1 世代，トランジスタを利用した第 2 世代，集積回路を利用した第 3 世代に分類することがある [1), 2), 3), 7), 8)]．1940 年代は第 1 世代の幕開けであった．本編 1 章で述べたように，コンピュータは1939年～1943年頃にアイオワ州立大学で開発されたアタナソフ（John V. Atanasoff）とベリー（Clifford E. Berry）による ABC（Atanasoff and Berry Computer）が最初といわれている．1946 年にはエッカート（J. Presper Eckert）とモークリ（John W. Mauchly）がペンシルバニア大学で ENIAC（Electronic Numerical Integrator and Computer）を完成させ，弾道計算の応用に利用されている．ENIAC は真空管 18000 本を使った巨大な（30 トン）システムであった（動作周波数は 100 kHz）．プログラムは演算器などを人手で結線する方式であり，後述するプログラム内蔵方式ではなかった．10 桁の 10 進数を記憶する 20 個のアキュムレータがあり，マスタプログラマユニットの制御の下で繰り返し演算を実行できた．演算時間は 10 桁の 10 進数の場合で 3 ms であった．最初の電子式コンピュータが ABC なのか ENIAC なのか論争（裁判）があった [6)] が，現在では，ABC であると結論づけられている．しかし，いずれにせよ，実用に供されたシステムとして ENIAC は歴史にその名を留めている．

(2) プログラム内蔵型コンピュータ

現在我々が使用しているコンピュータの基本原理が 1945 年ノイマン（John von Neumann）によって EDVAC（Electronic Discrete Variable Automatic Computer）プロジェクトの中で発表された．EDVAC はペンシルバニア大学のムーアスクール（Moore School）のエッカートとモークリを中心にノイマンも加わって推進されていたが，論文 11) がノイマンの単著となっており，現在のコンピュータの構成方式をノイマン型コンピュータと呼ぶようになった．ノイマン型コンピュータの基本方式はプログラム内蔵方式（stored program）と呼ばれている（誰が最初にプログラム内蔵方式を提案したのか議論のあるところであるが，ムーアスクールのエッカートとモークリを中心に提案されたのは事実であろう）．プログラム内蔵方式とは記憶装置の中に，問題の解決を図る機械命令列とデータからなるプログラムが置かれ，機械命令列が電子的な速度で一つずつ読み出され実行されていく方式であり，また，プログラムを電子的な速度で入れ替えることによって多様な応用に適用できる方式であった．プログラム内蔵方式の原型を 1941 年のツーゼ（Konrad Zuse）の Z3 や 1944 年のエイケン（Howard Aiken）の Harvard Mark I（いずれもリレーを用いた電気機械式）に見ることができる．これらのコンピュータでは映画フィルムや紙テープにプログラムが記憶され，命令が機械的な速度で読み出され実行されていた．また，条件分岐命令はなく，最後の命令が実行されると最初の命令に帰るようになっていた．

EDVAC は，1 K 語の水銀遅延線メモリ（水銀遅延線内で音波を繰り返し伝搬させることで記憶する方式．後にトランジスタの開発で著名なショックレー（William Shockley）によって

開発された）によるメインメモリ，20 K 語の磁気ワイヤメモリによる 2 次メモリなど容量の大きなメモリを装備していた．また，2 進数を用いていた．最初に提案されたノイマン型コンピュータであったが，ムーアスクールでの様々な内紛があり，開発が遅れ，1951 年に稼動した．

1946 年のムーアスクールでのコンピュータの講義などに触発されて，イギリスのマンチェスタ大学，ケンブリッジ大学，NPL（National Physical Laboratory）などで研究開発が推進され，最初のプログラム内蔵方式のコンピュータはイギリスで産声を上げた [10]．最初の実用的なノイマン型コンピュータは 1949 年にケンブリッジ大学のウィルクス（Maurice V. Wilkes）によって完成された．このコンピュータは EDSAC（Electronic Delay Storage Automatic Computer）と呼ばれ，極めて単純な機械命令形式（後述のアキュムレータ方式）を採用していた．マンチェスタ大学ではキルバーン（Tom Kilburn）らが 1947 年から Manchester Mark I の開発をウィリアム（Freddie C. Williams）とともに開始し，1949 年に稼動している．ウィリアムは同年静電型記憶装置を開発している．この記憶装置は陰極線管（ブラウン管）を用いたもので，電子ビームが蛍光板に照射されると，電荷が蓄積され，これが 200 ms 程度の間保持されることを利用した記憶装置であった．開発当時 1000 ビットの容量が実現されていた．1949 年には Mark I は稼働し，128 語（1 語 : 40 ビット）の静電記憶装置と 1024 語のドラムからなる 2 レベルの主記憶装置を有しており，またインデックスレジスタ（B-lines と呼ばれていた）が装備されていた．また，マンチェスタ大学の MARK I のプロトタイプ Baby がプログラム内蔵方式の点では最初ともいわれている．Baby は 1948 年 6 月 21 日稼動し，32 語（1 語 : 32 ビット）の静電型記憶装置と単純な機械命令（後述のアキュムレータ方式）を装備していた．

ノイマンらは 1946 年からプリンストン大学で IAS（Institute for Advanced Study）コンピュータの開発を進め，1952 年に稼動した．1024 語の静電型記憶装置，20 ビットの機械命令（アキュムレータ型）を装備し，加算命令の実行に 60 μs 要した．ハーバード大学では 1949 年にエイケンが Harvard Mark III を完成させている（電気機械式の Mark I は 1944 年，それを改良した Mark II は 1947 年）．5000 個の真空管（一部電気機械式）を用いたシステムであり，9 つの磁気ドラムを記憶装置として用いていた．命令列とデータは異なった磁気ドラムに格納されており，今日，キャッシュメモリを命令列とデータに分離して格納する分離型キャッシュをハーバードアーキテクチャと呼ぶことがあるが，その語源となっている．

(3) 機械命令形式

機械命令形式は，機械命令内のオペランドとして主記憶アドレスを何個指定できるかによって，3 アドレス方式，2 アドレス方式，1 アドレス方式，0 アドレス方式と分類される．データを供給する（ソース）メモリアドレスをMs，データを格納する（デスティネーション）アドレスをMd，データを供給するレジスタをRs，データを格納するレジスタをRdとし，OPをオペレーションコード（通称オペコードと呼ばれ，四則演算などを表す）とすると，3 アドレス方式ではMd←Ms1 op Ms2，2 アドレス方式ではMd←Md op Ms，1 アドレス形式では，Rd←Rd op Msとなっている．0 アドレス形式はスタック方式（詳しくは本章 2-1-3(2)を参照）のことである．1 アドレス形式の場合で，レジスタが 1 個に限定されている場合をアキュムレータ（Accumulator : ACC）方式と呼び，機械命令としては，ACC←ACC op Ms（演算命令），

ACC←Ms（ロード命令），Md←ACC（ストア命令），PC←Ms if ACC＝0（ACCの内容が 0 なら分岐などの，条件分岐命令．PCは実行すべき機械命令のアドレスを指すプログラムカウンタ）などがある．

先に述べた EDSAC はこのアキュムレータ方式を採用しており，機械命令数 18，水銀遅延線を利用した 1024 語（1 語：18 ビット）のメモリ，3000 本の真空管から構成されていた．EDSAC の機械命令には，加算，減算，乗算，ACC と特殊レジスタとの転送命令，二つの ACC 内容に基づく条件分岐，文字の入出力，ACC の左右シフト，ACC の丸め，停止などの機械命令があったが，無条件分岐命令はなかった．また，インデックスレジスタも装備されていなかった．ウィラー（David J. Wheeler）はサブルーチン呼び出しの方式を考案し（Wheeler Jump と呼ばれた），ブートストラップローダなどにより，リロケータブル（プログラムを主記憶装置の任意の領域で実行できること）なプログラミングが可能となった．EDSAC は 1956 年まで利用されていた．EDSAC も EDVAC も記憶装置としては水銀遅延線を使用しており，アクセス時間は 300 μs から 1 ms であった．IAS コンピュータもアキュムレータ方式であった．

4 アドレス方式がEDVACで採用されていた. EDVACの命令はA1 A2 A3 A4 OPの形式をしており，A1，A2 で指定されるメモリの内容を呼び出し，演算（OP）し，結果をA3 で指定されるメモリに格納し，A4 で指定されるメモリアドレスにある命令に分岐する．この分岐先命令を指定する 4 アドレス命令形式は，後のデータフローコンピュータを彷彿させるものがある（本章 2-1-4(9)参照）．

2-1-2 1950 年代

(1) 商用コンピュータの発展

1950 年代に入って商用コンピュータが続々と開発されてきた [1), 3), 4)]．1951 年にはレミントンランド社が水銀遅延線を主記憶装置に利用した UNIVAC を最初の商用コンピュータとして人口統計局に納入した．同じく 1951 年には ERA 社（Engineering Research Associates）が磁気ドラムを主記憶に使用した ERA-1101 を発表した．1953 年にはマサチューセッツ工科大学（MIT）でフォレスター（J. W. Forrester）が磁心マトリックス（コアメモリ）を開発し，Whirlwind と呼ぶコンピュータを開発した．1954 年には IBM 社が真空管，コアメモリ（12 μs で最小構成で 4 K 語）を採用した IBM 704（3 個のインデックスレジスタや浮動小数点演算器を内蔵）を発表し（納入は 1956 年），科学技術分野で優位に立った．IBM 704 には原始的な OS（control program）が装備されていた．1958 年にノースアメリカン航空会社に IBM 704 向けの FORTRAN II が納入され，それを対象に FORTRAN モニタが開発され，OS のはしりとなった．また，IBM 704 の後継の IBM 709（真空管，1958 年納入）は入出力プロセッサ（メモリとの間で DMA が可能）を装備していた．

(2) 第 2 世代のコンピュータ

1958 年，Philco 社は最初のトランジスタコンピュータ TransacS-2000 を，また，IBM は事務用コンピュータ IBM 7070 を開発し，トランジスタを用いた商用コンピュータが一般化した（トランジスタを使用した実験機では，1953 年の MIT リンカーン研究所の TX-0 がある）．これがコンピュータの第 2 世代の始まりである．IBM 709 のトランジスタ版である科学技術計算用コンピュータ IBM 7090（後に 7094）は 1959 年に納入され，商業的に成功した．IBM 7090

には IBSYS（システムモニタ），IBJOB（ジョブモニタ）という本格的な磁気テープベースの OS が装備されていた．IBM 7094 では，6 つのインデックスレジスタによるインデックスアドレッシング，間接アドレッシング，サブルーチン呼出し命令，入出力プロセッサ，割込みを有していた．

トランジスタを用いた大型汎用（事務用，科学技術計算用を併せもった）コンピュータとして IBM Stretch と UNIVAC の LARC がある．IBM 社はロスアラモス原子力研究所から Stretch の発注を受け，1961 年 IBM 7030 として完成させた．IBM Stretch では現在でいうパイプライン制御方式が採用されており，先行制御（advanced control）と呼ばれていた．パイプライン制御方式は IBM System/360 モデル 91（本章 2-1-3(8)参照）で本格化する．UNIVAC も同時期にリバモア原子力研究所から LARC（Livermore Atomic Research Computer）の発注を受け，1960 年に納入している．Stretch や LARC はパイプライン制御，マルチプログラミング，マルチプロセッサを意識した野心的なコンピュータであったが，商業的には失敗した．

ハネウェル（Honeywell）社は 1958 年 H-800 で多重プログラミング（8 多重）を実用化した（多重プログラミングの元祖は EDSAC（1956 年）にあるといわれている）．多重プログラミングとは，複数のプログラム（プロセス，タスク）を見掛け上，同時に実行する方式である．各プログラムは一定の時間経過したり，時間のかかる入出力操作が必要になったりした際には，ほかのプログラムに実行が切り替えられる．

1959 年，Electrologica 社が X-1 システムで，ハードウェア割込みの方式を開発した（UNIVAC 1103 で商用化）．割込みは第 3 世代コンピュータで本格利用され，多重プログラミング，オンラインリアルタイム処理，対話処理などで不可欠な方式となった．これらのコンピュータのほか，GE（General Electric）社，NCR（National Cash Register）社，RCA（Radio Corporation of America）社などから多様なコンピュータが開発されている．

メモリアクセスために種々のアドレッシングモードが用意されている．よく使用されるものにインデックスモードがある．これは，インデックスレジスタの内容と機械命令内の変位部を加算して実効アドレスを生成する方式である．この方式は 1953 年の ElectroData 社の Datatron で最初に実装された（1949 年のマンチェスタ大の MARK I でも B lines（B register）の名称で用いられていた）．また，間接アドレッシングモードは 1958 年の IBM 709 で用いられている．これらのアドレッシングモードは機械命令のアドレス部の書換えを不要とした点で大きな意義をもっている．

汎用レジスタは 1956 年の Ferranti 社の Pegasus で最初に採用されている．また，浮動小数点数は 1958 年の IBM NORC と 704 で使用されている．

1950 年代後半から，FORTRAN，COBOL など高級言語が使用され始めた．FORTRAN（Formula Translation）は科学技術用の高級言語であり，1954 年から 1957 年に IBM のバッカス（John Backus）らによって開発された．COBOL（Common Business Oriented Language）はビシネス応用向きの高級言語であり，1959 年に CODASYL 委員会（Conference on Data Systems Languages）で仕様策定された（COBOL 以前には 1955 年から 1958 年に開発されたホッパー（Grace Hopper）らによる FLOW-MATIC がある）．また，人工知能用の高級言語 Lisp（List processer）が 1958 年 MIT のマッカーシ（John. McCarthy）によって開発されている．

(3) マイクロプログラミング方式の提案

ウィルクスがプログラム内蔵型のEDSACを 1949 年に開発したことを先に述べたが，ウィルクスはその開発の経験からマイクロプログラム制御方式を 1951 年に発表した. コンピュータの機械命令の実行は機械命令の取り出し（フェッチ），解読（デコード），オペランドデータの読み出し(オペランドフェッチ), 演算実行, 結果の格納などのステージを経てなされる．演算器，レジスタやメモリはバスなどを通して相互に結合されており，これをデータパスと呼ぶ．一方，このデータパスを制御するのが制御装置であり，制御装置は各ステージを一つずつ状態遷移させながら機械命令を実行する．この状態機械に対する従来の設計手法は布線論理 (wired logic) と呼ばれ, 論理ゲートとフリップフロップによる順序回路設計手法であった．高速動作は可能であるが，状態数が多くなり，状態遷移そのものが複雑化すると，順序回路の設計と修正が非常に困難になる．そこでウィルクスは論文 12）を発表し，マイクロプログラムによる制御装置の系統的な設計法を示した．マイクロプログラム制御方式の基本は，制御装置の状態遷移をプログラム化したことであり，これをストアドロジック（stored logic）という [13]. このプログラムは機械命令のプログラムと異なってマイクロプログラムと呼ばれ, 制御装置内に置かれた高速の制御記憶装置（control store）に格納されている．機械命令の実行では, まず命令フェッチのためのマイクロ命令系列が呼び出され実行され, 次にデコード, オペランドフェッチなどのためのマイクロ命令系列が順々に実行されていく．このマイクロ命令を実行する順序回路は非常に簡単な順序回路で実現できる．ウィルクスが提案した当時には高速な制御記憶装置を実現できるメモリがなかったので，マイクロプログラムの実用化は 1964 年のIBM System/360 の出現まで待たなくてはならなかった（本章 2-1-3(3)参照）．マイクロ命令の形式には語長の長い（64 ビット以上）水平型 (horizontal type) と語長の短い（32 ビット程度）垂直型（vertical type）がある．前者は多数のフィールドで多数の演算器などを並列操作でき高速であるが, 並列操作ができないときは, そのフィールドはNOP（No Operation）となり，ビット使用効率はよくない．後者はビット使用効率はよいが，通常の機械命令のように並列操作が少なく，水平型に較べて同一処理に必要なマイクロ命令数が多く，水平型に較べて性能は低い. マイクロプログラム制御方式は 1964 年のIBM System/360 で実用化され, 1970 年代に高級言語コンピュータの実現方式として注目された．また，水平型のマイクロ命令形式を汎用化して，機械命令レベルに格上げしたVLIW（Very Long Instruction Word）方式が 1970 年代半ばに提案された（本章 2-1-4(7)参照）.

2-1-3 1960 年代

(1) 仮想記憶方式の提案

仮想記憶の実現にはページング方式とセグメンテーション方式がある．1961 年，仮想記憶のページング方式がマンチェスタ大学で開発されたAtlas で初めて採用された（論文発表 1962 年）[4), 5)]．Atlas では，磁気ドラムとコアメモリという二つのレベルの記憶装置を一つのレベルの記憶装置に見せるため, One-Level Storage System と呼ばれていた. Atlas の主記憶 (central store）は 96 K 語（1 語：48 ビット）の磁気ドラムと 16 K 語のコアメモリから構成されている．磁気ドラムとコアメモリは共に 512 語のページに分割されている（コアメモリのページをページ枠と呼ぶ）．コアメモリの各ページ枠にはページアドレスレジスタ（P.A.R）が置かれており，各々にはそこに格納されている仮想ページ番号が格納されている（P.A.R は全部

で 32 個）．アクセスすべき仮想ページ番号が与えられると，P.A.R の内容と一斉比較がなされ，一致した P.A.R に対応するページ枠にアクセスされる．現在のコンピュータでの仮想-実ページの変換には，仮想ページ番号を用いてページテーブルを検索する直接写像方式とページ枠テーブルを連想アクセスする連想写像方式があるが，Atlas は後者の方式である．これは現在ほとんどのコンピュータで使用されている TLB（Translation Lookaside Buffer）の原型ともいえる．TLB の数は，GE 645 で 16 個，IBM 360 モデル 67 で 8 個，System 370 では 128 個となっている．また，Atlas には原始的な OS が装備されている．

一方，セグメンテーション方式は 1961 年 Burroughs 社の B5000（本章 2-1-3(2)参照）で開発された．プログラムはあるまとまったサブプログラムやデータ（可変長）に分割され，それぞれに対してセグメント番号（Burroughs ではディスクリプタアドレス）が与えられる．セグメントテーブルがあり，セグメント番号でこれを検索し，メモリに当該セグメントがあれば，そのセグメントの先頭アドレスが得られるので，セグメント内アドレスと加算すると，実メモリのアドレスが得られる．この方式では，各セグメントは実メモリの連続領域に取られるので，使用されない小さな領域（フラグメント）が多数生成されることがある．小さなフラグメントの総計はサイズが大きいのに連続領域ではないので，大きなセグメントに利用できないことがある．そこで，GE 645 では，各セグメントに対応したページテーブルを装備して，最終的にはページング方式で管理している．セグメント数だけページテーブルが存在する．

(2)　高級言語指向のコンピュータ B5000

1961 年，Burroughs 社は B5000 を発表した（1963 年に稼動）[14]．このコンピュータは 1960 年発表の高級言語 Algol 60 の構造をハードウェアレベルで色濃く反映しており，高級言語コンピュータの源流となっている．本章 2-1-4(4)に後述するように高級言語コンピュータには直接実行型，間接実行型，構文指向型があるが，B5000 は構文指向型の代表例である．機械命令形式は 0 アドレス方式となっており，スタック指向のコンピュータとなっている（English Electric 社の KDF-9 もスタックコンピュータとなっている）．B5000 は歴史的意義が大きなコンピュータとして本編 3 章 3-1 節で取り上げてあり，詳細はそちらを参照．

(3)　第 3 世代のコンピュータ IBM System/360

1964 年，IBM は IBM System/360 を発表した．ハイブリッド集積回路を用いて構成されており，第 3 世代コンピュータの幕開けとなった．System/360 ではファミリーシリーズの考えが採用され，発表当時，モデル 30，40，50，60，62，70 などがあり，例えばモデル 30 ではメモリ容量 64 K バイト，固定小数点演算時間 103 μs，モデル 70 ではメモリ容量 512 K バイト，固定小数点加算時間 3.4 μs など，性能，価格が異なる様々なシステムを供給し，ユーザのニーズに応えた．System/360 は歴史的意義が大きなコンピュータとして本編 3 章 3-3 節で取り上げてあり，詳細はそちらを参照．

System/360 で最も重要な点はデータ，プログラム，入出力装置，入出力インタフェースに関して互換性の実現を徹底させた点である．とりわけ，制御装置にはマイクロプログラム制御方式を採用した点が重要である．データパスやマイクロ命令形式は当該コンピュータの機械命令（この場合，IBM System/360）を最も効率よく実行できるように設計されるが，マイ

クロプログラムを書き換えることにより，異なったコンピュータの機械命令を，機械命令プログラムでシミュレーションするよりはるかに高速にシミュレーションできる．これをエミュレーション（emulation）と呼ぶ．IBM System/360 ではそのマイクロプログラムを変更して IBM 704 などの古いコンピュータの機械語プログラムを実行することができた．当時 IBM 704 など 6 機種も存在した IBM コンピュータのエミュレーションを実行でき，これによって過去のコンピュータとの互換性を得ることができた．過去との連続性をマイクロプログラム制御方式によって実現し，将来に向けては上位互換性を保ち，1964 年以降の機械命令レベルのプログラムを後継機で継続利用することが可能となった．また，マイクロプログラム方式のもう一つの特徴として，マイクロ診断がある．マイクロ命令では通常の機械命令ではアクセスできない特殊なレジスタなどにもアクセスでき，ハードウェアの診断に利用できる．

IBM System/360 は OS/360 のオペレーティングシステムを装備しており，汎用大型コンピュータが定着した．また，事務用，科学技術計算用を合わせもった高級言語 PL/I を開発した（1967 年）．

IBM System/360 のアドレス空間は 24 ビットであった．1972 年の IBM System/370 での仮想記憶（アドレス空間は 24 ビット）の導入，1984 年の S370-XA で 31 ビットアドレスへの変更，2000 年の z/Architecture で一挙に 64 ビットアドレスへの変更がなされたが，機械命令セットの大幅な変更はなされていない．

System/360 に関する 1964 年の論文 15) でアムダール（Gene M. Amdahl）は「アーキテクチャ」を次のように定義している．The attributes of a system as seen by the programmer, i.e., the conceptual structure and functional behavior, as distinct from the organization of the data flow and controls, the logical design, and the physical implementation.（アーキテクチャとは，プログラマから見えるシステムの属性，すなわちデータの流れ，制御法，論理設計，物理的な実現法などの機構とは異なった概念的な構造，機能動作である）．

IBM System/360 のほかにも，RCA Spectra70（モノリシック集積回路を最初に使用），Burroughs B-5500/6500（スタックコンピュータ），CDC 6600（本章 2-1-3(8)参照，RISC 方式の源流），GE 645（Multics で使用），Honeywell 2200（2 アドレス命令形式）など多数の汎用大型コンピュータが開発された．

(4) タイムシェアリングシステム（TSS）

MIT では 1961 年 MAC（Multiple Access Computer）プロジェクトを開始し，最初の TSS（Time Sharing System）を開発し，1963 年には電話回線を 30 回線結合した CTSS（Compatible Time Sharing System）を開発している（IBM 7094（コアメモリ 32 K 語）を使用）．また，同時期にダートマス大学などでも研究開発がなされている．1964 年 MIT は大型の TSS システムである MULTICS（Multiplexed Information and Computing Service）の開発を開始した．MULTICS は歴史的意義が大きなコンピュータとして本編 3 章 3-4 節で取り上げてあり，詳細はそちらを参照．

TSS は中央に大型汎用コンピュータを設置し，電話回線によって多数の端末を結合したシステムである．多数のユーザが同時にシステムを利用でき，しかもシステムを占有しているかのように見せかけるシステムであった．多重プログラミングの方式によってユーザプロセスが一定の時間（タイムクアンタムという．例えば 50 ms）ごとに切り替わるようになって

いる．日本では，1967 年ころから電子技術総合研究所で ETSS などの実験システムが開発された後，電電公社が 1971 年に DEMOS のサービスを開始した．TSS 自体は 1970 年代に実用化したが，以後ミニコンピュータやワークステーション，更にはインターネットなどによる分散処理の普及によって衰退した [18].

(5) 並列コンピュータの開発開始

多数の演算器やプロセッサを用いた並列処理の着想は 1920 年のリチャードソン（Lewis F.Richardson）に遡ることができる．数値積分による気象予測をするのに 64000 人を円形劇場に集め，中央に指揮者が立ち，隣同士とのみデータを交換して処理を進める方式であった．スタンフォード大学のフリン（Michael. Flynn）は 1966 年に並列コンピュータを下記の四つに分類している [16), 17)].

①SISD：Single Instruction Stream Single Data Stream

②SIMD：Single Instruction Stream Multiple Data Stream

③MISD：Multiple Instruction Stream Single Data Stream

④MIMD：Multiple Instruction Stream Multiple Data Stream

①SISD は通常の逐次実行のコンピュータである．

②SIMD は一つの命令を多数のデータに作用させる方式で三つの方式がある．一つは演算装置を多数（空間並列）用意し，制御装置から放送される命令を全演算器が一斉に実行するアレイプロセッサ（array processor），もう一つの方式は，演算器を流れ作業のように多数のステージに分割し，データを次々に流していく時間並列のパイプライン方式（pipelined processor），残りひとつは連想プロセッサ（associative processor）である．

③MISD はストリーミング処理に対応するもので，前段プロセッサの処理結果を使用して次段のプロセッサで処理する．あまり注目されていないためか，論文での記述がほとんどない．

④MIMD は自律した動作ができる装置が互いに協調して動作するシステムであり，マルチプロセッサが代表例である，

フリンの定義では Stream の見方によってパイプライン方式が SISD，SIMD，MISD となるので，今日では，制御装置から放送される命令を空間的に多数配置された演算器が一斉に実行する方式を SIMD 方式，演算装置を多数のステージに分割し，多数のデータに対して流れ作業的に各ステージでオーバラップして実行する方式をパイプライン方式，マルチプロセッサ，クラスタコンピュータ，ネットワークコンピュータなど自立した装置が協調して処理を進める方式を MIMD 方式と呼んでいる．SIMD 型並列コンピュータでは，1962 年にウェスティングハウス社のスロットニック（D.L. Slotnick）が SOLOMON（Simultaneous Operation Linked Ordinal Modular Network）コンピュータを開発した．32×32 の格子状に配置した 1 ビット演算器（PE : Processing Element）からなるシステムであった．SIMD 型並列コンピュータは，ベル電話研究所の Unger のコンピュータなどを経て，1965 年 ILLIAC IV がイリノイ大学で設計され Burroughs 社で開発が開始された（完成は 1973 年）. ILLIAC IV では，1 台の制御装置（CU: Control Unit），疑似トーラス結合された 64 台の演算器（PE），各演算器に附属した記憶装置（PEM : Processing Element Memory，2048 語）からなり，64/32 ビット浮動小数点演算，48/24/8 ビットの固定小数点演算が実行される．機械命令は CU でデコードされ，水平型マイクロ命

令（語長 260 ビット）に変換され，各 PE に放送され実行される．150 MFLOPS 程度の性能を有している [19), 20)].

一方，パイプライン型並列コンピュータは，1965 年の IBM のセンジック（D.N. Senzig）の VAMP（vector arithmetic multi-processor），1969 年の IBM 2938 などから始まり，1972 年のテキサスインスツルメント（TI）社の ASC（Advanced Scientific Computer），1973 年の CDC STAR-100（String Array）などを経て，1976 年の CRAY-1，CYBER 205 で方式が確定した [21)]．パイプライン方式は流れ作業と同様に浮動小数点演算器を多数のステージに分割し，ベクトルデータ（1 次元の配列データ）を演算器に投入する方式であり，投入後しばらく演算結果はでないが，出始めると毎サイクル演算結果が得られる（本章 2-1-4(10)参照）．

MIMD 型並列コンピュータについては，1960 年代には 1960 年の Holland のコンピュータ（John.Holland）の提案のほかは，本格的なマルチプロセッサは提案・開発されていない．

(6) ミニコンピュータの普及

1965 年，DEC（Digital Equipment Corporation）社は最初の商用ミニコンピュータ PDP-8 を出荷した．語長は 12 ビットであった．実験的なミニコンピュータとして 1963 年の MIT の LINC（Laboratory Instrument Computer）があった．その他，Varian，HP，Interdata などから多数のミニコンが製品化された．日本でも 1969 年以降 HITAC 10，FACOM R，NEAC M4，OKITAC-4300，MACC7，TOSBAC-40 などが製品化している．1970 年には DEC 社は 16 ビットミニコン PDP-11 を製品化し，1978 年の VAX-11/780（Virtual Address Extension）で 32 ビットスーパーミニコンとして確立した [22)]．また，マイクロプロセッサ版の LSI-11 を 1975 年に製品化している．PDP-8，PDP-11，VAX-11/780 は歴史的意義が大きなコンピュータとして本編 3 章 3-5 節で取り上げてあり，詳細はそちらを参照．

(7) ARPA ネットワークによる分散処理の幕開けと UNIX の普及

1968 年には米国防総省の ARPA（Advanced Research Projects Agency of the Department of Defense）で ARPA ネットワークのプロジェクトが立ち上がり，1969 年には四つのサイトを IMP（Interface Message Processor）を介して 50 kbits/s の回線で結んだネットワークが稼動した [23)]．IMP には DDP-516（マシンサイクル 0.96 μs，12 K 語のコアメモリ）が使用された．ARPA ネットワークは 1973 年に Internet に発展していく．また 1969 年，ベル研究所ではリッチー（Dennis M. Ritchie）とトンプソン（Ken Thompson）が PDP-7，PDP-9 上に UNIX を実装し，1971 年には PDP-11 上で稼動させている [24)]．当時の PDP-11/45 は 144 K バイトのコアメモリの容量があり，その内，UNIX は 42 K バイトを占めていた．ハードディスクは 1 M バイトの容量があった．また，1972 年にはリッチーは高級言語 C を開発し，C で UNIX を記述し，そのソースコードを公開した．

(8) パイプラインの発展形 -アウトオブオーダ（Out-of-Order）実行方式

今日の多くのマイクロプロセッサではスーパースカラ方式によるアウトオブオーダ実行によって高速化がなされている．1964 年開発の CDC 6600，1968 年開発の IBM System/360 モデル 91 にその源流を見ることができる．アウトオブオーダ実行方式では，先行命令と後続命令を可能なら並列実行し，更には後続命令でも実行時間が短ければ先行命令より早く終了させ，

高速化を達成する方式である．しかし，先行命令と後続命令には，フロー依存（先行命令の結果を後続命令が利用する），逆依存（先行命令が読み出してからでないと後続命令は同一記憶箇所に結果を書き込めない），出力依存（先行命令と後続命令が同一記憶箇所に書込みを行う際には，後続命令の結果を残して置く必要がある）があり，並列処理が妨げられる．

1964 年に CDC 6600 がクレイ（Seymour R. Cray）らによって開発された [25]．先に述べた 3 つの依存関係のうち，フロー依存と逆依存を実行時に保証できる．CDC 6600 は後述の RISC（Reduced Instruction Set Computer）の原型にもなっている．機械命令の実行制御はスコアボードでなされ，後続命令で先行命令と依存がなければ，アウトオブオーダ実行がなされるが，出力依存が生じた際には実行が停止する．ソーントン（James E. Thornton）がこの方式を考案した．CDC 6600 は歴史的意義が大きなコンピュータとして本編 3 章 3-2 節で取り上げてあり，詳細はそちらを参照．

1968 年トマスロ（Robert M. Tomasulo）が出力依存も実行時に解消する斬新な方式を発表した [26]．例えば，**図 2･1** の(a)の例では，①と②にフロー依存，①と③に出力依存，②と③に逆依存，③と④にフロー依存，②③と⑤に逆依存がある．ソーントンの方式では，①と③の出力依存のために，この時点で後続の命令の実行はすべて待たされる．トマスロの方式では(b)のように①と②の R0 を tmp0 に実行時に変更する．このようにすると，①と③の出力依存が解消され並列に実行できる．同時に②と③の逆依存も解消される．また，⑤の R4 を tmp1 に実行時に変更することにより逆依存も解消される．この方式はレジスタリネーミングと呼ばれる（レジスタリネーミングの用語はその後 1975 年にケラー（Robert M. Keller）によって初めて使われている [28]）．

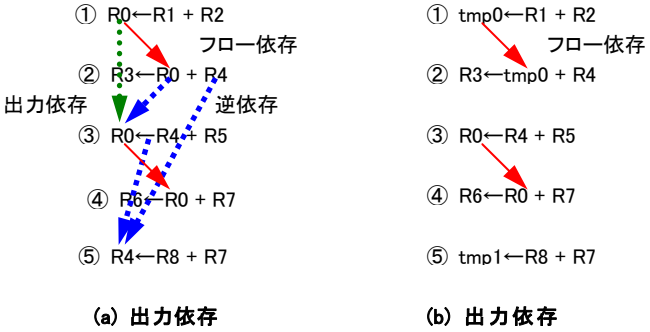

図 2･1　出力依存の解消

トマスロの方式で問題となったのが不正確な割込み問題（Imprecise Interrupt）である．後続命令が先行命令より早く終了してしまうことが可能となったので，プロセッサの中には既に終了した命令，実行途中の命令，実行待ちの命令が組み合わさって存在する．このような場合に先行命令で例外事象（演算エラーなど）が生ずると，例外を起こした命令まで状態を戻し，プログラムの再実行をする必要がある．しかし，このためには例外を起こした命令以後の命令実行状態をすべて保持する必要があり，当時の技術としては困難であった．このため，トマスロの方式は IBM/360 モデル 91 以後，1970 年代，1980 年代には実用には至らな

かった．1985 年スミス（James E. Smith）は正確な割込みを保障する方式として，リオーダバッファ，ヒストリバッファ，フューチャファイルなどの方式について性能評価 [27] を行って以降，実装技術の発展とともに広く利用されるようになった．

(9) キャッシュメモリ

キャッシュメモリは IBM System/360 モデル 85 に最初に採用された（1968 年論文，原始的な形態は IBM 7094 に存在する）．主記憶装置のサイクルタイムは 1.04 μs，容量は 512~4096 K バイトであり，キャッシュメモリの容量は 16 K バイト，アクセスタイムは 80 ns であった．主記憶装置とキャッシュメモリは 1 K バイトのセクタに分割され，この単位で写像されている．しかし，一度に 1 K バイト転送するのではなく，セクタの中で必要となった 64 バイト単位のブロックごとに転送される．アクティビティリストのトップから順に最近アクセスされたセクタ番号が記憶され，最も底のセクタが置き換えの対象となった．今日でいう LRU（Least Recently Used）置き換えアルゴリズムが採用されている．書き込みの際には，主記憶装置とキャッシュに書き込みがなされる（これをライトスルーという．主記憶装置には書込みを行わない方式をライトバックという）．また，キャッシュアクセスには 2 サイクル必要で，第 1 サイクルでセクタのディレクトリとブロックの有効ビット検索，第 2 サイクルでキャッシュデータにアクセスする．これらは実際にはオーバラップ（パイプライン）して実行されるので，基本的には 1 サイクルごとに処理される．19 本のテストプログラムで性能評価した結果，ヒット率は 0.968，全キャッシュを想定した理想的な場合と比較して 81%の性能を得ている [30]．

キャッシュメモリのライトポリシ（ライトスルーかライトバックか）についての議論は 1974 年頃から始まっている．また，命令キャッシュとデータキャッシュを分離する分離型（スプリット）キャッシュは実験機ではローレンスリバモア研究所の S-1（論文 1977 年），IBM ワトソン研究所 801（1976 年論文）などがあり，商用機では NEC の ACOS 1000（1980 年発表．命令キャッシュ，データキャッシュ各々64 K バイト）などがある．今日では，この分離型キャッシュが第 1 階層のキャッシュ（一次キャッシュ）では一般的である．また，キャッシュメモリのディレクトリを検索するのに，仮想アドレスを用いる仮想アドレスキャッシュが MU-5（1977 年論文）や IBM 801 で採用されている．実記憶では同一アドレスで共有されるデータがキャッシュメモリ上では共有されないという別名（シノニム，synonym, alias）問題があり，解決方式が種々提案された [29]．

2-1-4 1970 年代

(1) DRAM とマイクロプロセッサの実用化

1970 年 IBM は System/370 を発表し，モデル 145 でバイポーラ型 IC メモリを，モデル 158 で NMOS の IC メモリを主記憶に採用している．また，モデル 158 には仮想記憶方式を採用している．しかし，System/370 は System/360 と較べてアーキテクチャ的には新規性には乏しいシステムであった．

1970 年に 1 K ビットの DRAM がインテル社で開発された．チップ上に集積できるトランジスタ数は 1.5 年で 2 倍（10 年で約 100 倍）の割合で増大するといわれており，後に Moore の法則と呼ばれることになった．実際に DRAM はそのペースで集積化が進んでいる（しかし近年は 3 年で 2 倍に鈍化している）．1970 年代初頭で 1 K ビット，アクセス時間 300 ns，2000

年頃で 1 G ビット，30 ns 程度となっている．DRAM のアクセス時間は 30 年近く経過しても 10 倍程度しか高速化していない．このため，先に述べたように，キャッシュメモリの階層化が必要となっている．なお，インテル社のムーア（Gordon Moore）は，1965 年に 1 年で集積度は 2 倍となると予測し，1975 年には 2 年で 2 倍となると予測の修正をしているが，1.5 年で 2 倍とはいっていないとしている．

インテル社は 1971 年に 4 ビットマイクロプロセッサ 4004 を製品化した [31]．2300 個のトランジスタで構成され，動作速度は 750 kHz であった．4004 は歴史的意義が大きなコンピュータとして本編 3 章 3-6 節で取り上げてあり，詳細はそちらを参照．1973 年には 8 ビットプロセッサ 8080A，1978 年に 16 ビットプロセッサ 8086 を発表し，機械命令セット IA-32（Intel Architecture）を確立した（その後，マルチメディア向きの命令セット MMX と SSE を加え，拡充を図っている）．一方，モトローラ社では 1974 年 MC 6800（8 ビット），1979 年 MC 68000（16 ビット），ザイログ社は 1976 年に Z80（8 ビット），1979 年に Z8000（16 ビット）を開発してきたが，汎用マイクロプロセッサの分野では衰退している．

(2)　マルチプロセッサ型並列コンピュータ

マルチプロセッサ型の並列コンピュータには，各プロセッサがメモリを共有する共有メモリ型マルチプロセッサとメモリを共有しないメッセージ交換（パッシング）型マルチプロセッサに分類できる [40), 41)]．

(a)　共有メモリ型マルチプロセッサ

C.mmp（Multi-minicomputer）はカーネギーメロン大学（CMU）で 1971 年から開発が進められた共有メモリ型のマルチプロセッサである [32)]．それ以前には，Burroughs D825，IBM 360/67，Honeywell 645 などがあったが，いずれも 4 プロセッサ程度のシステムであった．C.mmp は 16 台の PDP-11 がクロスバスイッチを介して 16 台の主記憶装置と結合されている．主記憶装置は 8 K バイトのページに分割されており，全体で 4096 ページからなっている．PDP-11 のアドレスは 16 ビットであるので，下位の 13 ビットでページ内アドレスを表し，上位の 3 ビットに PSW（Program Status Word）内の 2 ビットを付けた 5 ビットで 32 個のリロケーションレジスタを検索する．そこにページフレーム番号（12 ビット．4096 ページのどれか一つ）が格納されている．共有メモリは 25 ビットアドレスでアクセスされる．リロケーションレジスタには，Non existent，Read-only，Dirty（Write あり），Cachable などの情報も含まれている．また，1 K バイトのキャッシュメモリが各プロセッサに付随していている．キャッシュコヒーレンスはリロケーションレジスタの Cachable 情報でソフト的に行うようになっている．今日のメモリ共有型マルチプロセッサのパイオニア的なシステムであるといえる．また，OS は Hydra である．

Cm*（modular multi-microprocessor）は CMU で 1972 年から検討が開始されたが，マイクロプロセッサ LSI-11 が出荷された 1975 年から本格的な開発が始まった [33)]．100 台規模のシステムを目指している．複数の LSI-11（computer module）をクラスタとしてバス結合し，クラスタでのアドレス変換を K.map で行い，これらをクラスタ間バス（Inter-Cluster Bus）で結合する方式となっている．メモリ共有方式となっており，共有空間は 256 M バイト（2^{28}）となっている．拡張性や信頼性に力点が置かれている．CMU ではフォールトトレラントを指向した C.vmp（voted multiprocessor）も 1975 年から開発している．3 重系による投票システ

ムで信頼性の向上を図っている．

1972 年にカルフォルニア州立大学バークレイ校（UCB）で高信頼性を目指した PRIME が開発されている．ミニコンピュータ Meta-4 を 5 台用いている．4 K 語のメモリバンクが 26 個装備されており，各プロセッサからはこれらの内の 16 台にアクセスされる．各メモリバンクはマルチポートで構成されている．

共有メモリ型マルチプロセッサにはメモリコンシステンシ問題やキャッシュコヒーレンス問題があり，1980 年代以降，様々な方式が提案されている [34), 35), 36)]（本章 2-2-1(11)参照）．

(b)　メッセージパッシング型マルチプロセッサ

当時としては大規模であったシステムとして，1977 年（論文発表），シーメンス社で SMS-201 が開発された．128 個のマイクロプロセッサ 8080 が単一バスに結合されている．通信フェーズでは，制御プロセッサは個々のプロセッサの通信用メモリ（CM）の内容を読み出し，これをすべてのプロセッサに一斉に放送（ブロードキャスト）する．したがって，すべてのプロセッサの CM の内容は同一となる．次に実行フェーズでは，各プロセッサが CM とローカルメモリを使用して独立に動作し，実行終了を制御プロセッサに通知し，全プロセッサが処理終了するまで待機する．この方式を SMIMD（Switched MIMD）と呼ぶ．

1975 年以降 1980 年にかけて，日本でも，NEC の MICS，徳島大学の CORAL，九州大学の HYPHEN，電総研の ACE，東芝の EPOS など数十台規模の実験用マルチプロセッサが開発されている．1980 年代には，カルフォルニア工科大学の Cosmic Cube や筑波大学の PAX など数百台規模のマルチプロセッサの研究開発がなされ，一部実用化された．

メッセージ交換型マルチプロセッサでは，各プロセッサの通信が OS を介してなされるので，このオーバヘッドの減少，通信処理中での無駄なパケットのコピーの削減が大きな課題であった．1980 年代以降，様々な方式が提案されている [37)]（本章 2-2-1(12)参照）．

(c)　並列コンピュータ向けの相互結合網

並列コンピュータの演算器，プロセッサやメモリ（ノードと呼ぶ）を結合（以下リンクと呼ぶ）する相互結合網も重要な要素である．相互結合網は大別して，静的網（直接網）と動的網（間接網）がある．静的網のスーパーセットは完全結合網ですべてのノードがほかのすべてのノードと結合している．リンク数は ${}_nC_2$ で $N(N\text{-}1)/2$ である（N はノード数）．これを縮退したものに，リング網，ツリー網，トーラス（格子）網，ハイパーキューブ網などがある．一方，動的網のスーパーセットはクロスバ網であり，スイッチ数は N^2 である．これを縮退したのが，多段結合網であり，Benes 網，3 ステージ Clos 網，オメガ網，間接 2 進 n-キューブ，ベースライン網などがある．今日使用されている相互結合の多くは 1980 年頃に出揃っている [38), 39)]（本章 2-2-1(14)参照）．

(3)　ユニバーサルホストコンピュータ

1972 年に発表された Burroughs B-1700 はカメレオンマシンと呼ばれたマイクロプログラム制御コンピュータである．通常のマイクロプログラム制御コンピュータでは，その固有の機械命令の実行に最適化したマイクロ命令形式が決定され，マイクロプログラムを格納する制御記憶装置は ROM（Read Only Memory）で構成される．これに対し，特定の機械命令を意識しないマイクロ命令形式を採用し，様々な機械命令セットの高速エミュレーションや，様々な高級言語や応用ごとに最適な機械命令を設定できる方式が B-1700 などのユニバーサルコ

ンピュータである．機械命令セットが自由に変更できるように，マイクロプログラムを格納する制御記憶装置は書き換え可能（Writable Control Store）となっている．書き換え可能なマイクロプログラム方式をダイナミックマイクロプログラミング方式と呼ぶ．B-1700 では，FORTRAN，COBOL，RPG，SDL（システム記述言語）などの高級言語をサポートしているが，機械命令は高級言語ごとに異なっている（FORTRAN S 言語など）．1971 年の SCC（Standard Computer Company）社の MLP-900（Multi Lingual Processor），1974 年の Nanodata の QM-1 なども同様な考え方で作られている．QM-1 では 2 レベルマイクロプログラム方式が採用されていた．水平型のマイクロ命令は語長が長く，また有効な操作が少なければ命令のビット使用率が悪い．したがって，よく使用される水平型マイクロ命令をナノ記憶装置に格納しておき，これを語長の短い垂直型マイクロ命令内のポインタで指定したのが 2 レベルマイクロプログラム方式である．QM-1 の場合にはナノ命令は 360 ビット，マイクロ命令は 18 ビットであった．ユニバーサルホストコンピュータをはじめとするダイナミックマイクロプログラム方式のコンピュータは，次に述べるように，高級言語マシンの構築に広く利用された [42]．

(4)　高級言語コンピュータ

1970 年前後からソフトウェア危機が叫ばれ，ソフトウェアの生産性向上や信頼性向上を図る動きが顕著になった．1968 年のダイクストラ（Edsger W. Dijkstra）の論文 Goto Statement considered harmful が構造化プログラミングの提唱のはしりとなった．1970 年にはヴィルト（NiklausWirth）により高級言語 Pascal が開発され，またプログラミング技法だけでなく，ソフトウェアのライフサイクルを視野に入れたソフトウェア工学の新しい研究領域が形成されてきた（1968 年の Nato の Software Engineering Conference でソフトウェア工学の名前が使用されている）．高級言語コンピュータはソフトウェア危機に対するハードウェア側からのアプローチである．

高級言語と機械語との間には機能上大きなギャップが存在する．マイヤーズ（Glenford J.Myers）はこれをセマンティックギャップと呼んでいる [43]．このギャップを埋めるコンピュータが高級言語コンピュータであり，性能低下，プログラムの信頼性低下，プログラムサイズの増大，コンパイラの複雑化を回避することが目的である．高級言語コンピュータは直接実行型，間接実行型，構文指向型に分類できる．直接実行型では，高級言語プログラムをそのままハードウェアで解釈実行（インタプリット）する方式で，変数名なども連想メモリで検索される．間接実行型は高級言語文と 1 対 1 対応する中間言語（可変長で一般に長い）に変換し，それをハードウェアやマイクロプログラムで解釈実行する方式である．変数などはメモリへのアドレスに変換されている．高級言語と 1 対 1 対応にあるので，高級言語レベルでのデバッグが容易であり，また，配列などに記述子（デスクリプタ）などが付加され，領域外参照などのエラーチェックなどが容易である．1967 年の EULER マシン，アイオワ州立大学で設計されフェアチャイルド社で開発した（1964~1970 年）SYMBOL や 1980 年の NEC の COBOL マシン COMBAT などがある [44]．EULER マシンは，マイクロプログラム制御コンピュータで開発された最初の高級言語コンピュータであり，IBM System360 モデル 30 上に開発された．EULER 言語は動的データタイプ，再帰呼び出し，リスト処理が可能な言語であり，ソースプログラムはバイト列からなるストリング言語にトランスレートされ，これをマイクロプログラムで実行する．SYMBOL では算術式は逆ポーランド記法に変換され，実行される．

また，トランスレータはハードウェアで作成されている．COMBAT では COBOL 文に 1 対 1 対応した可変長の中間言語となっている．

構文指向型は高級言語の制御，演算，データ構造などの構文規則を反映した，上記以外のコンピュータであり，一般的に語長の短い機械命令形式を採用している．代表例は先に紹介した B5000 である [14]．先に述べた B-1700 上の高級言語マシンも構文指向型に属する．また，DEC 社は 1978 年に PDP-11（16 ビットマシン）の後継機 VAX-11 780（32 ビットマシン）を発表した．VAX では直交型の命令形式を有している．すなわち，機械命令の命令操作部とオペランド部が独立しており，各オペランドでは用意されているアドレッシングモードのいずれでも指定できる．アドレッシングモードは豊富に揃えられている [22]．

1973 年，フューステル（Edward A. Feustel）はタグアーキテクチャを提唱した．各データ自体にその属性をもたせる方式である．多くのコンピュータでは浮動小数点加算は FADD，整数加算は IADD などと機械命令の命令操作部で演算すべきデータタイプを指定するが，タグアーキテクチャでは，命令操作部は ADD のみであり，データ自体に浮動小数点数，整数などの属性が付いている．このため，異なるデータタイプの演算があれば，エラーとなる．

多くの研究がなされたが，高級言語コンピュータは衰退した．その理由は本章 2-1-4(8)の RISCの項で述べる [45), 46)]．

(5) 分岐予測方式

1972 年，マンチェスタ大学の MU5 に関する論文に分岐予測方式が最初に述べられている．条件分岐命令の実行でジャンプが起こった場合を Taken，起こらなかった場合を Not Taken と呼ぶ．分岐予測には局所分岐予測と広域分岐予測がある．前者は当該条件分岐命令の過去の分岐情報（Taken か Not Taken）をもとに今回の分岐方向を予測するものである．後者は当該条件分岐命令に至るほかの条件分岐命令の分岐情報から当該条件分岐命令の方向を予測する方式である．また，この二つの方式を混合した方式をハイブリッド方式と呼ぶ．MU5 には Jump Trace と呼ばれる連想記憶があり，過去の 8 つの条件分岐命令とその分岐先が記憶されている．Jump Trace にヒットすれば，前回と同一の分岐先が予測される．分岐予測方式は現在様々な方式が提案されているが，最初に各種の方式の比較検討を行ったのはスミス（James E. Smith）である [47)]．また，1973 年には IBM System 370/168 では複数命令流方式が提案され，Taken 側，Not Taken 側を同時に並行してフェッチして実行するようになっている．より高度な分岐予測方式が 1980 年代以降提案されている（本章 2-2-1(5)参照）．

(6) ワークステーション Alto と Ethernet

1973 年 Xerox パロアルト研究所でワークステーション Alto（演算幅は 16 ビット）が発表され [50)]，Ethernet による LAN に接続された．Alto はビットマップディスプレイをもち，現在のウィンドウシステムの原型を作ったものとしても評価される．Alto は歴史的意義が大きなコンピュータとして本編 3 章 3-7 節で取り上げてあり，詳細はそちらを参照．

(7) VLIW 方式

1976 年，京都大学の QA-1，Floating Point Systems 社の AP-120B が発表された．これらのコンピュータは後にエール大学のフィッシャー（JosephFisher）により VLIW（Very Long

Instruction Word）方式[51]と呼ばれ，非常に語長の長い機械命令形式を採用していた．この方式の源流は水平型マイクロ命令方式であり，それを汎用化したものである．QA-1 は 160 ビット長の長命令で四つの固定小数点演算，四つのメモリアクセス，一つの順序制御を指定することができた．また，AP-120B は 64 ビット長の長命令で浮動小数点加算，同乗算，固定小数点演算の計三つの操作を指定できる科学技術計算用コンピュータであった．長命令内の操作にはコンパイラによって確実に並列実行できる操作が埋め込まれる．したがって，応用自体に並列性がない場合やコンパイラの能力が劣る場合には，長命令内には NOP 操作が多くなり，超命令のビット使用効率は減少する．1980 年代に Trace（1987 年），Cydra5（1987 年）など，汎用目的の商用システムも開発されたが，メディアプロセッサやスーパーコンピュータのスカラ演算器など，十分な並列性が期待できる応用のほかには，成功しなかった．これはこの期の VLIW 方式は単純な構成方式をとっており，高速化機構や長命令のビット使用率の向上手法が十分に揃っていなかったこと，拡張性（例えば長命令長を 2 倍にする場合にはソースプログラムの再コンパイルをする必要があること）に問題があること，などに原因があると考えられる．1980 年以降実装技術やコンパイラ技術の発展により VLIW 方式が実用化されている（本章 2-2-1(4)参照）．

(8) RISC の台頭

1975 年 IBM ワトソン研究所の 801 棟でコック（J. Cocke）は RISC 型の IBM 801 の開発を開始した[52]．同じ頃，カリフォルニア大学バークレイ校（UC Berkeley）のパターソン（David A. Patterson）やスタンフォード大学のヘネシー（John L. Hennessy）が研究を開始し，1980 年代にパターソンは RISC-I を SUN Micro Systems 社で，ヘネシーは MIPS のマイクロプロセッサを MIPS 社で商用化している[53]．IBM 801 は歴史的意義が大きなコンピュータとして本編 3 章 3-9 節で RISC-1 や MIPS とともに取り上げてあり，詳細はそちらを参照．

RISC の設計思想は，

①ある一定の機能を有するプロセッサを限られたトランジスタ数を有するチップに集積しなくては，プロセッサの速度はマルチチップ化により大幅に低下する．したがって，チップに搭載する機能を精選する必要がある．

②以前にはプログラムはアセンブリ言語により，熟練者が非常に複雑な機械命令をも使って効率のよいプログラムを作成していた．しかし，プログラムの作成は高級言語でなされるようになり，コンパイラによって自動的に機械命令に変換される．ところがコンパイラの生成する機械命令は極めて単純なものが多い．

③単純な機械命令を有するコンピュータは開発期間が短く，結果として最新の集積回路技術を早期に利用できる．

以上の理由から RISC はまず非常に単純な機能をもった機械命令のみを選択し，これをマイクロプログラム制御ではなく布線論理で実現した．また，1 機械命令で 1 操作を実行するようにするため，ロード・ストア命令形式（演算はレジスタ間で実行，メモリとの通信はロード命令，ストア命令で対処）とした．機械命令のデコードを単純化させるため，命令長を固定長（32 ビット）とした．アドレッシングモードも極力少なくした．マイクロプログラムのための制御記憶などの領域を大容量レジスタファイルなどにあてた．

CISC 命令形式では，高機能命令が用意されており，全体のプログラムサイズを小さくでき

る，高機能に対応したハードウェアによる高速化や並列処理の利用ができる，などの特徴がある．しかし，欠点としては，

①コンパイラが CISC 命令を使い切れない．

②CISC であるので，マイクロプログラム制御方式が必要となり，制御記憶のチップ内に占める割合が大きくなり，これがほかの機能の実装を阻害する．

③CISC 命令は多くの場合，VAX の場合のように可変長となっており，デコードに時間がかかる．

1980 年から 1990 年代始めまで CISC と RISC の論争 [54] は続いた（本章 2-2-2(2)参照）．

(9) データフロー方式

1974 年，MIT のデニス（Jack Dennis）はデータフロー（data flow）コンピュータ（あるいはデータ駆動（data driven）とも呼ばれる）を発表した．従来のコンピュータはコントロールフローコンピュータと呼ばれ，プログラムカウンタで指された機械命令のみが実行された．データフローコンピュータでは，機械命令がどこに存在しても，そのオペランドデータが揃えば実行開始（これを発火（fire）という）できる．したがって，プログラムカウンタは存在せず，プログラムに並列性があれば，原理的にそれらを並列処理できる．デニスの初期の方式では，機械命令（セル）は操作部（OP），オペランド 1，オペランド 2，行き先 1，行き先 2 のフィールドから構成されていた．オペランド 1，オペランド 2 にデータが揃うと機械命令は発火し，演算終了すると，その結果をトークンとして行き先 1 と行き先 2 で指される機械命令のオペランドに転送する．行き先の機械命令でオペランドがすべて揃えば，その機械命令が発火する．その機械命令の他方のオペランドが未到着であれば，到着するまで発火は待たされる．デニスのこの方式は静的データフロー方式と呼ばれ，同一のプログラムを異なる地点で同時に呼び出す（リエントラントな呼出し．例えば再帰的呼出し）ことは不可能であった．これを可能にしたのが，1977 年の MIT のアービンド（Arvind）や 1978 年のマンチェスタ大学のガード（John Gurd）らによる動的データフロー方式である．タグ付きトークン法に基づいている．タグ付きトークン法では，データフロープログラムが呼び出されるごとに，そのプログラムに異なるタグが与えられる．機械命令の操作部，行き先は命令記憶装置に，オペランドはマッチング記憶装置に格納される．演算結果のトークンには，演算結果を渡す行き先の命令アドレス，演算結果のほかにこのタグが付随している．タグの示すビット列と行き先を示すビット列を単純に連結し，それをアドレスとしてマッチング記憶にアクセスすることで，タグごとに異なったオペランドに結果が渡される．そのタグでのオペランドが揃えば，行き先の示す命令記憶内の命令が発火される．このような方式を循環パイプライン方式と呼び，ガードらの方式に基づいている．しかし，この単純な方式では，マッチング記憶装置のアドレス長が極めて長くなるので，何らかの連想記憶が必要となる．

1980 年代には動的データフロー方式の研究が行われ，プロトタイプ機も開発された [55]（本章 2-2-1(13)参照）．

(10) ベクトルプロセッサ

1976 年，ベクトルプロセッサ CRAY-1 が発表された [56]．1972 年の TI 社の ASC（Advanced Scientific Computer，1 台の演算装置当たり 16.7MFLOPS．4 台まで増設可能）や CDC の

STAR-100（STring ARray，40 MFLOPS）が第 1 世代のベクトルプロセッサと呼ばれるのに対して，CRAY-1 は第 2 世代のベクトルプロセッサと呼ばれる．CRAY-1 は歴史的意義が大きなコンピュータとして本編 3 章 3-8 節で取り上げてあり，詳細はそちらを参照．また，1980 年代以降は第 3 世代のベクトルプロセッサが開発されている（本章 2-2-1(9)参照）．

日本のメーカでは 1977 年の富士通 FACOM/230-75AP，1978 年の日立 IAP（汎用大型コンピュータ M-180 の内蔵型アレイプロセッサ）が開発された．

CRAY-1 と同時期に開発されたスーパーコンピュータに CDC の CYBER205（その後継機は ETA10）がある．このスーパーコンピュータには，STAR-100 と同様にベクトルレジスタがなく，データはすべて主記憶装置から供給され，演算結果も主記憶装置に格納される．ベクトルアクセスの際にレイテンシが長くなるので，ベクトル長の長い演算に有利であった．また，CRAY-1 では中間結果がベクトルレジスタに存在するので，メモリバンド幅を小さくできるが，CYBER 205 では二つのソース，一つのデスティネーションがすべてメモリアクセスとなり，平均的に CRAY-1 と比べて 3 倍のメモリバンド幅が必要になる．CRAY-1 と CYBER 205 の競争はベクトル長の短いベクトル演算に有利で，メモリバンド幅が小さくて済む CRAY-1 に軍配が上がり，今日まで，CRAY-1 タイプのシステムが採用され続けている．

いずれの方式でもプログラム内でベクトル化できる割合（スカラプロセッサでの処理時間の内，ベクトル化できる時間の割合のことで，これをベクトル化率 α という）が高くないとベクトルプロセッサの効果がでない．通常のスカラプロセッサと比較して，ベクトルプロセッサによる性能向上率は非ベクトル化率の逆数（1/（1-α））以上にはならない（これをアムダールの法則（この法則はベクトル化ばかりでなく，マルチプロセッサでの並列化など，様々な場面で利用される）という）．

(11) SIMD 型並列コンピュータ

1977 年に Burroughs 社から BSP（Burroughs Scientific Computer）が発表された．これは SIMD 方式のスーパーコンピュータであり，16 台の演算器と 17 台のメモリバンクがクロスバスイッチを介して結合されていた．性能は 50 MFLOPS であった．2 次元配列のアクセスでは任意の行，任意の列，対角要素（1 次元ベクトル）に同時アクセスすることが多い．BSP ではメモリ台数を演算装置より大きな最小の素数とすることにより，データのアクセス競合を抑えることができた [57), 58)]．

BSP の後継機として 512 台の演算器を有するシステムが計画された．512 台の演算器を実装するシステムはそれを超える最小の素数 521 台のメモリバンクが必要であった．BSP は斬新なアイデアに満ちていたが，複雑であり，CRAY-1 との競争に敗れた．

また，1970 年代のほかの SIMD 型並列コンピュータとしては，1974 年（論文発表）の Goodyear Aerospace 社の STARAN がある [59)]．STARAN では，1 語 256 ビットのデータを 256 語格納できる MDA（Multi Dimensional Access）メモリがあり，任意の行，任意の列にアクセスできる構造となっていた．これを直交メモリという．連想メモリとして使用できる．MDA は 256 ビットの通常メモリを 256 個集めて構成され，ここのメモリに対するアドレスの与え方は非常に単純な排他的論理和を多用した方式となっている．各メモリから読み出された 256 ビットのデータはオメガネットワークに似た Flip Network で並び替えられ，256 個の 1 ビット演算器で処理される．また，Goodyear 社は 1980 年に MPP を開発した．MPP は 128×128

の格子点上に 1 ビット演算器が置かれ，SIMD 制御の下で動作する．主として画像処理用に使用された．

SIMD 方式はその後 1985 年の GF-11，Connection Machine と発展した（本章 2-2-1(10)参照）．しかし，大規模な SIMD 型並列コンピュータは 1990 年代より衰退した．一方，マルチメディア処理の高速化を図る MMX や SSE 命令セット，GPU などで，小規模な SIMD 方式は活用されている（本章 2-2-3(5)参照）．

(12) ハードウェアアルゴリズム

1978 年に IBM のトッド（S. Todd）がパイプラインマージソータを発表した．パイプラインを使用したソータであり，1982 年のクーン（Hsiang-Tsung Kung）によるシストリックアレイとともに，ハードウェアアルゴリズムの研究が盛んになった．ハードウェアアルゴリズムは VLSI 上での容易な実装を考慮に入れた高機能演算器の構成法である [60), 61)]．例えば，

①実装が容易なように，規則的な 1 次元，2 次元配列の演算器と隣接接続を使用する．

②入力データは VLSI の周辺の演算器に与えられ，出力データも同様に VLSI の周辺の演算器から取り出される．

行列計算，画像・信号処理，ソータなど多様なハードウェアアルゴリズムが考案されている．また，近年のリコンフィギュラブルコンピュータでもハードウェアアルゴリズムの考えで構成されるシステムも多い．

(13) マルチスレッド型プロセッサ

1978 年に Denelcor 社のスミス（Burton J. Smith）がマルチプロセッサ HEP を発表した．プロセッサはマルチスレッド方式で動作しており，多数のプロセスが 1 命令ごとに切り替えられて実行される．今日 SMT（Simultaneous Multithreading）と呼ばれる方式に対して，TMT（Temporal Multithreading）と呼ばれている方式の原型である．TMT の基本的な発想は多重プログラミングに依っている．多重プログラミングは，OS の基本機能のひとつであり，多数のプロセスが存在する際に実行中のプロセスが入出力操作を行う場合やある一定時間（タイムクアンタムという）実行された場合に他のプロセスに切り替える（プロセススイッチ）機能のことである．TMT はこの考えを極端に進めたものであり，プロセスの切り替えを 1 命令ごとに行うものである．各プロセスに依存関係がなければ，各プロセスから次々に取り出される命令にも依存関係がないので，命令パイプラインは淀みなく流れることになる．TMT の性能が発揮されるためには，命令パイプライン段数以上のプロセスの実行が必要である．Itanium2-Montecito や SPARC Niagara など最新のマイクロプロセッサでも TMT 方式を採っているものもある [62)]（本章 2-2-3(3)参照）．

■参考文献

1) Michael R. Williams, “History of Computing Technology, 2nd Ed.,” IEEE Computer Society Press, 1997.
2) 情報処理学会歴史特別委員会(編), “日本のコンピュータ発達史,” オーム社, 1998.
3) 高橋茂, “電子計算機 I,” 共立出版, 1975.
4) 石井善昭, “計算機の基本方式,” 共立出版, 1973.
5) 元岡達, “計算機システム技術,” オーム社, 1973.
6) 星野力, “誰がどうやってコンピュータを作ったのか？,” 共立出版, 1995.

7) J. P. Hays, “Computer Architecture and Organization,” McGraw-Hill, 1978.
8) S. Rosen, “Electric Computers: A Historical Survey,” Computing Survey, Vol.1, No1, pp.7-36, 1969.
9) D. P. Siewiorek, C. G.Bell, A. Newell, “Computer Structures: Principles and Examples,” McGraw-Hill, 1982.
10) サイモン・ラヴィングトン, 末包良太訳, “コンピュータの誕生―イギリスを中心に―,” 蒼樹書房, 1981.
11) John von Neumann, “First Draft Report on the EDVAC,” 1945.
12) M. V. Wilkes, “The Best Way to Design an Automatic Computing Machine,” Report of Manchester University Computer, 1951.
13) 萩原宏, “マイクロプログラミング,” 産業図書, 1977.
14) E. A. Hauk, B. A. Dent, “BurroughsB6500/B7500 Stack Mechanism,” Proceedings of the AFIPS Spring Joint Computer Conference (SJCC), pp.245-251, 1968.
15) G.M.Amdahl, G. A. Blaauw, F.P.Brooks, “Architecture of the IBM System/360,” IBM Journal of R&D, Vol.8, No.2, April, pp.87-101, 1964.
16) M. J. Flynn, “Some Computer Organizations and Their Effectiveness,” IEEE Trans. Computers, Vol.C-21, No.9, pp.948-960, 1972.
17) M. J. Flynn, “Very High-Speed Computing Systems,” Proceedings of the IEEE, Vol..54, No.12, pp.1901-1909, 1966.
18) 美間敬之編, “タイムシェアリングシステム,” オーム社, 1972.
19) 加藤満左夫, 苗村憲司, “並列処理計算機,” オーム社, 1976.
20) R. W. Hockney, C. R. Jesshope（奥川峻史, 黒住祥祐訳, “並列計算機,” 共立出版, 1984, 原著 Parallel computers, “architecture, programming and algorithms,” Adam Hilger Ltd, 1981.）
21) 長島重夫, 田中義一, “スーパコンピュータ,” オーム社, 1992.
22) C. G. Bell, J. C. Mudge, J. E. McNamara, “Computer Engineering-A DEC View of Hardware Systems Design,” Digital Press, 1978.
23) F. E. Heart, R. E. Kahn et.al., “The interface message processor for the ARPA computer network,” Proc. of SJCC, pp.551-567, 1970.
24) D. M. Ritchie, K. Thompson, “The UNIX Time-Sharing System,” CACM, Vol.17, No.7, pp365-375, 1974.
25) J. E. Thornton, “Parallel Operation in the Control Data 6600,” Proc. of AFIPS Fall Joint Computer Conference (FJCC), pp.33-40, 1964.
26) R.M.Tomasulo, “An Efficient Algorithm for Exploiting Multiple Arithmetic Units,” IBM Journal of R&D, Vol.11, No.1, pp.25-33, 1967.
27) J. E. Smith, “Implementation of Precise Interruption in Pipelined Processors,” Proc. of Int. Conf. on Computer Architecture (ISCA), pp.36-44, 1985.
28) D. Sima, “The Design Space of Register Renaming Technique,” Proc. of IEEE Micro, pp.70-83, 2000.
29) M. Cekleov, M. Dubois, “Virtual-Address Caches,” Proc. of IEEE Micro, pp.64-71, 1997.
30) A. J. Smith, “Cache Memories,” Computing Surveys, Vol.14, No.3, pp.473-530, 1982.
31) 嶋正利, “マイクロプロセッサの 25 年,” 電子情報通信学会誌, 82 巻, 10 号, pp.997-1017, 1999.
32) W. A. Wulf, C. G. Bell, “C.mmp-A multi-mini-processor,” Proc of Fall Joint Computer Conference, pp.765 -777, 1972.
33) R. J. Swan, S. H. Fuller, D. P. Siewiorek, “Cm*-A Modular, multi-miniprocessor,” National Computer Conference, pp.637-644, 1977.
34) J. Archibald, J. Baer, “Cache Coherence Protocols,” ACM Trans. Computer System, Vol.4, No.4, pp.273-298, 1986.
35) P. Stenstrom, “A Survey of Cache Coherence Schemes for Multiprocessors,” IEEE Computer, Vol.23, No.6, pp.14-24, 1990.
36) K. Gharachorloo et. al., “Memory Consistency and Event Ordering in Scalable Shared Memory Multiprocessors,” Proc. of Int. Conf. on Computer Architecture (ISCA), pp.15-26, 1990.
37) 大森健児, “マイクロプロセッサによるマルチマイクロプロセッサシステムの技術動向,” 情報処理, 23 巻, 7 号, pp.651-658, 1982.
38) T. Feng, “A Survey of Interconnection Network,” IEEE Computer, Vol.14, No.12, pp.12-27, 1981.

39) 富田眞治, “並列コンピュータ工学,” 昭晃堂, 1996.
40) T. Feng (ed), “An Overview of Parallel Processors and Processing,” ACM Computing Surveys, Vol.9, No.1, pp.1-129, 1977.
41) L. S. Hays (ed), “Highly Parallel Computing,”, IEEE Computer, Vol.15, No.1, pp.7-96, 1982.
42) 相磯秀夫, 飯塚肇, 坂村健, “ダイナミックアーキテクチャ,” bit 増刊, 共立出版, 1980.
43) G. Myers, “Advances in Computer Architectures,” John Wiley&Sons, 1978.
44) 箱崎勝也, 山本昌弘, “高級言語マシンの実際,” 産報出版, 1981.
45) 島田俊夫, 坂村健, 山口喜教, “高級言語マシン”, 情報処理, 18 巻, 4 号, pp.386-394, 1977.
46) 日比野靖, “高級言語計算機,” 電子情報通信学会誌, 78 巻, 11 号, pp.1164-1170, 1995.
47) J. E. Smith, “A Study of Branch Prediction Strategies,” Proc. of Int. Conf. on Computer Architecture (ISCA), pp.135-148, 1981.
48) J. K. F. Lee, A. J. Smith, “Branch Prediction Strategies and Branch Target Buffer Design,” IEEE Computer, Vol.17, No.1, pp.6-22, 1984.
49) N. Gloy et.al., “An Analysis of Dynamic Branch Prediction Schemes on System Workloads,” Proc. of Int. Conf. on Computer Architecture (ISCA), pp.12-21, 1996.
50) C. P. Thacker et.al., “Alto:A Personal computer”, CSL-79-11, Xerox PaloAlto Research Center, 1979.
51) J. A. Fisher, “Very Long Instruction Word Architecture and the ELI-512,” Proc. ISCA, pp.140-150, 1983.
52) G. Radin, “The 801 Minicomputer,” Proc. of Int. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS), pp.39-47, 1982.
53) D. A. Patterson, C. H. Sequin, “A VLSI RISC,” IEEE Computer, Vol. 15, No.9, pp.8-21, 1982.
54) D. Bhandarkar, D. W. Clark, “Performance from Architecture, Comparing a RISC and a CISC with Similar Hardware Organization,” Proc. of Int. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS), pp.310-319, 1991.
55) 弓場敏嗣, 山口喜教, “データ駆動型並列計算機,” オーム社, 1993.
56) R. M. Russell, “The CRAY-1 Computer System,” CACM, Vol.21, No.1, pp.63-72, 1978.
57) C. Jensen, “Taking Another Approach to Supercomputing,” Datamation, pp.159-172, February, 1978.
58) D. Lawrie, C. Vora, “The Prime Memory System for Array Access,” Proc. of Int. Conf. Parallel Processing, pp.81-87, 1980.
59) K. E. Batcher, “STARAN parallel processor system hardware,” Proc. of National Computer Conference (NCC), pp.405-410, 1974.
60) 矢島脩三, 富田悦次, 都倉信樹(編), “特集 VLSI 向きハードウェア,” 情報処理, 26 巻, 6 号, 1985.
61) H. T. Kung, “Why Systolic Architecture,” IEEE Computer, Vol.15, No.1, pp.37-46, 1982.
62) 中條拓伯, 河原章二, 上原哲太郎, 並木美太郎, “Simultaneous Multithread (SMT) アーキテクチャの現状と今後,” 情報処理, 43 巻, 3 号, pp.281-287, 2002.

■6 群 － 1 編 － 2 章

2-2 1980 年代から 2000 年代まで

（執筆者：馬場敬信）［2010 年 1 月 受領］

2-2-1 1980 年代

1980 年代のコンピュータ技術の発展は，ダイナミックマイクロプログラミングからスタートして，RISC プロセッサが登場し，命令パイプラインを効率よく動作させるためのアーキテクチャ及びコンパイラ技術が発展した．また，命令パイプラインを基礎として，更に命令レベル並列性を引き出すために，スーパースカラ技術，分岐予測方式と新たな技術開発が行われている．同時にチップ上のトランジスタ数の増大に伴って，キャッシュメモリのオンチップ化などメモリシステムの高性能化も進んだ．一方，並列コンピュータに関しては，1970 年代に引き続いてベクトルスーパーコンピュータの性能向上が図られるとともに，SIMD，MIMD 方式の並列コンピュータ，データフロー方式と種々の方式の並列コンピュータが研究開発されている．

(1) ダイナミックマイクロプログラミングから RISC，VLIW への流れ

書き換え可能な制御記憶を活用したダイナミックマイクロプログラミングは，1970 年代から 1980 年代にかけてのコンピュータシステム技術の重要なトピックであった [1]．ダイナミックマイクロプログラミングは，それまで読出し専用であった制御記憶を，高速な書き換え可能メモリに置き換えた技術である．ソフトウェアで実現していた処理をマイクロプログラム化することにより，応用指向のアーキテクチャを実現することが容易になり，高水準言語処理，オペレーティングシステム，数値計算など，各種の応用に向けてダイナミックマイクロプログラミング技術が活用された [2]．マイクロプログラムは，マイクロ命令によってハードウェアをレジスタ転送レベルで制御するものであり，特定の応用に向けてレジスタレベルの並列性を抽出することを可能としている．マイクロ命令形式は，大きく垂直型と水平型に分類され，小規模のコンピュータでは，制御記憶の総容量を抑えるため符号化を進めた垂直型が基本となり，中・大規模コンピュータでは，レジスタレベルの並列性を活用して命令サイクルを短縮することを重視して，符号化の程度を抑えた水平型が基本となった．

この流れの中から，RISC と VLIW という二つのアーキテクチャ技術が生れた．

RISC は，垂直型マイクロ命令の流れを引いている．命令形式を簡単にして，一命令一サイクル実行を基本とするところも同じである．いわば，垂直型マイクロ命令を，機械命令レベルに引き上げたものと見ることができる．

一方の VLIW は，水平型マイクロ命令の流れを引いている．レジスタレベルでの並列動作の可能性を重視して符号化を抑え，命令長を長く取って，同時に複数のハードウェアユニットを制御できるようにしている．いわば，水平型マイクロ命令を，機械命令レベルに引き上げたものと見ることができる．

(2) RISC の登場

最初に RISC コンピュータを実現したのは，1975 年に IBM のクック（John Cock）らによって開発が開始された IBM 801 CPU であり [3]，1981 年に ROMP（Research OPD (Office Products

Division) Mini Processor）と呼ぶ単一チッププロセッサとして実現された．IBM 801 は歴史的意義が大きなコンピュータとして本編 3 章 3-9 節で取り上げてあり，詳細はそちらを参照．

RISC の基本的な設計指針は，本章 2-1-4(8) に述べたとおりであるが，その効果は，(i) ソフトウェア／ファームウェア／ハードウェアの階層からファームウェア（マイクロプログラム）の層をなくすことにより，チップ上で制御記憶とその周辺回路に使用していた領域を不要としたこと，(ii) 一命令一サイクル実行の簡単な命令形式とすることにより，コンパイラから見て分かりやすく，最適化しやすいアーキテクチャとしたこと，などに要約される．

(ii) の特徴は，コンパイラによる静的な最適化の重要性を大きくした．例えば，メモリへのアクセスはロード・ストア命令に限定することにより，算術論理演算命令はすべて陽にレジスタを使用する形になって，静的に扱いやすい形式である．RISC は，また，CISC における，頻繁に使用される機能のファームウェア化あるいはハードウェア化によって高速化を図る路線が，命令のもつ意味と命令形式の複雑化につながり，コンパイラにとっては使いにくい命令セットとなっていることへの反省でもあった．

ほぼ同時期に米国の大学において，相次いで RISC コンピュータが単一チップとして実現された．

1980 年には，カリフォルニア大学バークレイ校における RISC プロジェクトが，パターソン（David Patterson）らのもとで開始された．1982 年に 44420 個のトランジスタからなる RISC-I プロセッサが試作された [4]．RISC-I の命令数は 32 である．一命令一サイクル実行とし結線論理制御を用いるという一般的な RISC プロセッサの特徴に加えて，RISC-I の特徴は，手続き呼び出しの高速化のために，レジスタウィンドウを設けたことである．これはリング状に構成したレジスタを使用し，手続き間のレジスタによる引数渡し，局所変数用のレジスタ切り替えを，使用するレジスタウィンドウの変更によって高速に実現する．

1981 年には，スタンフォード大学において MIPS プロジェクトがヘネシー（John L. Hennessy）によって開始された [5]．MIPS プロジェクトは，RISC の基本である一命令一サイクル実行を前提に，命令パイプラインのインターロックを避けることを目標としている．このために，遅延分岐（delayed branch）を取り入れるとともに，命令間に空きスロットを生ずる場合には，ほかから空きスロットへのコード移動を行ってこれを埋めることにより，命令パイプライン中に何も実行しないステージ（パイプラインバブル）を生ずることを避けて，高性能化を図る点に特徴がある．

1980 年代初頭に研究開発されたこれらのRISCプロセッサは，その後の商用プロセッサに影響を与えている．IBM 801 をもとにIBM RT-PCが設計され [6]，1986 年にリリースされているが，残念ながら性能は出なかった．しかし，RISCの設計思想は，その後のPOWERアーキテクチャ，PowerPCアーキテクチャへと受け継がれている．RISC-Iに続いて開発されたRISC-IIの設計思想は，サンマイクロシステムズ社（Sun Microsystems）のSPARCに引き継がれ，1985 年にSPARCがリリースされている．スタンフォード大のMIPSは，1984 年にヘネシーらによって立ち上げられたMIPS社（MIPS Computer Systems）に引き継がれ，1985 年にR2000 チップがリリースされた．そのほか 1985 年には，Acom Computer Ltd.によるARM1 がリリースされ，1989 年には，Hewlett-Packard社からPA-RISCがリリースされている．

(3) 32 ビットプロセッサの普及

4 ビットからスタートしたマイクロプロセッサの処理単位が, 8 ビット, 16 ビットと進み, 1980 年代には 32 ビットに到達した．したがって，(2)で述べた RISC の基本語長も 32 ビットであった．

RISC に限らず, CISC の世界においても，1985 年に発表されたインテル社の i80386 に代表される 32 ビットプロセッサが現れた．このアーキテクチャは，その後 IA-32 と呼ばれる．i80386 は，32 ビットをフルに使用した 4 G バイトの物理アドレス空間と，64 T バイトの仮想アドレス空間とをもつ．

同じ 1985 年に発表されたモトローラ 68020 も, CISC の 32 ビットプロセッサである. 68020 は，マイクロプログラムを格納する制御記憶を 2 階層に分け，語長の短い垂直型マイクロ命令から語長が長い水平型のマイクロ命令を指定して実行する 2 レベルマイクロプログラム制御方式を用いている．これによって，制御記憶全体の容量を削減しつつ，水平型マイクロ命令によるレジスタレベルの並列処理を可能としている．

(4) 命令レベル並列性による性能向上

1980 年代初頭に相次いで開発された RISC プロセッサの基本は，命令パイプラインの効率的な実行を基本としている．命令パイプラインを滞りなく動作させるには，続いて実行される命令間の依存関係を実行前に見極める必要がある．

例えば，5 ステージからなる典型的な命令パイプラインを，第 1 ステージ：命令キャッシュからの命令読出し，第 2 ステージ：命令デコード，レジスタ読出し，第 3 ステージ：演算実行，アドレス計算，第 4 ステージ：データキャッシュアクセス，第 5 ステージ：演算結果の書き戻し，と定義する．

このような定義のもとでは，例えば，ロード命令によってアクセスされるデータキャッシュの出力は 4 番目のステージの最後で利用可能になるため，これをバイパス回路で直接 ALU 入力に戻しても，次の命令の 3 番目の実行ステージで行われる演算には間に合わない．このため，1 ステージ分の遅れを生ずることになる．このため，両命令間に 1 命令分の空きスロットを設け，ここに両命令の実行と独立で，逐次実行の意味を損なわない命令を移動することにより，パイプラインは滞ることなく実行を継続することが可能になる．

ここで重要なことは，これらのコード最適化がコンパイラによって静的に行われることである．一命令一サイクル実行を前提にした RISC の命令形式は，命令パイプラインの振る舞いを静的に読みきるのに都合のよい形式であった．また同時に，性能向上のためには，コンパイラとアーキテクチャの共同作業が必要なことを認識させた．ダイナミックマイクロプログラミングで培われたコード最適化技術も活用されている．

命令パイプラインを基礎として，命令レベル並列性の更なる向上技術として，この年代に発展した技術は，スーパースカラ方式と VLIW アーキテクチャである．

スーパースカラ方式の源流は，本章 2-1-3(8) 述べた 1964 年の CDC 6600，あるいは 1968 年のトマスロアルゴリズムにある．アウトオブオーダに実行する際の問題点として指摘された不正確な割込みに対する対策として，1985 年にスミス（J. E. Smith）によりリオーダバッファが提案された [7]．リオーダバッファは，アウトオブオーダでの実行結果と実行中に起った例外を記録する．リオーダバッファに記録された情報は，プログラム順にレジスタファイ

ルに書き込まれるので，例外が起きていれば，その時点までの書き込み結果が正確な割込みを保証する．この機構は更に 1990 年にソヒ（G. S. Sohi）によって，複数の機能ユニットをもち正確な割り込みを保証するパイプラインコンピュータに対する命令発行機構としてまとめられている [8]．

リオーダバッファは，実行のたびに実際に書き込まれるレジスタ（物理レジスタ）と，命令セットアーキテクチャから見えるレジスタ（論理レジスタ）とを分離している．このような方式を，レジスタリネーミングと呼ぶ．レジスタリネーミングは，レジスタの再利用によって生ずる逆依存，出力依存の解消に使用される．レジスタリネーミングの実現法には，リオーダバッファによる方法以外にも，論理レジスタから物理レジスタへのマップ表を用いる方法があり，この方法は 1985 年パット（Y. N. Patt）により提案されている [9]．

単一チップ上にスーパースカラ機構を実現した最初の商用 RISC マイクロプロセッサは，1989 年にアナウンスされたインテル i960CA である．

アウトオブオーダのスーパースカラ方式がハードウェアによる動的なコードの並べ替えを行うのに対し，VLIW アーキテクチャはソフトウェア（コンパイラ）による静的なコードの並べ替えを行う．本章 2-1-4(7) にその起源について述べ，QA-1, AP-120B について述べたが，本章 2-2-1(1) にも述べたようにこれらはレジスタレベルの並列動作の可能性を重視してビット長を長く取った水平型マイクロ命令を機械命令レベルにしたものと見ることができる．1983 年にフィッシャー（J. A. Fisher）によって VLIW と命名された後，1987 年には Trace, 1989 年には Cydra5 といった商用システムも開発された．VLIW アーキテクチャを前提にしたコードスケジューリング法として，1981 年に論文発表されたトレーススケジューリングがある [10), 11)]．トレースとは，基本ブロックを超えて頻繁に実行される命令の系列を指す言葉である．基本ブロック内に限定した命令系列だけをスケジューリングの対象としたのでは，VLIW アーキテクチャのもつ並列性を十分に活用できないために考えられた方法である．トレースを見つけるためには，いったんプログラムを実行して，そのプロファイル情報を取得する必要がある．

VLIW アーキテクチャを活用するための，もう一つの著名なコードスケジューリング法に 1988 年にラム（M. Lam）によって論文発表されたソフトウェアパイプライニングがある [12]．これはループのイタレーション間の依存関係を見極めたうえで，イタレーションごとの命令を少しずつずらしながら，VLIW の命令スロットに割り付けることにより，パイプライン式の実行を可能とする方式である．

(5) 分岐予測方式の発展

逐次コードを命令レベルにおいて並列実行する際の大きな課題の一つが，分岐命令の処理である．分岐命令において実行が分岐条件成立側にいくのか，不成立側にいくのかは実行してみないと分からないので，命令パイプラインあるいは広く命令レベル並列処理を阻害する要因になる．これは制御ハザードとして分類される．

この課題に対して，分岐予測方式が考案されたことと，その初期の成果については，本章 2-1-4(5) に述べた．1981 年には，スミスが各種の方式について比較検討を行っている．分岐予測は大きく実行前に行われる静的な方式と，実行時の情報を使用して行われる動的な方式とがある [13]．

静的な予測では，常に成立側あるいは不成立側と予測する．例えば後方分岐をループの繰返しを制御するものとして成立側に予測するなどが行われる．また，1986 年にマクファーリング（McFarling）はプロファイル情報を取得してこれに基づいて静的に分岐命令ごとに予測する方法を提案し，80～90%の予測精度が達成できることを示している．

静的な予測では，個々の分岐命令の分岐方向予測は，実行中には固定され，処理するデータなどによる分岐方向の変化には追随できず，予測精度を更に高めることが難しい．一方，命令レベル並列性を追求しようとすると 95%を超える予測精度の向上が望まれ，またチップ上に搭載可能なハードウェア資源も増大したことから，動的な分岐予測機構の研究が加速した．動的な分岐予測機構は，*n* ビットカウンタ予測方式から出発する．これは，分岐命令アドレスをインデックスとして，カウンタテーブルを引き，分岐の成立・不成立に応じて該当する飽和型カウンタを増減させる（最大値からの増加，最小値からの減少は前の値のままとする）ものである．カウンタ長としては，2 ビットのものがよく使用されている．

1990 年代初頭には，カウンタ列をアクセスするのに，過去の分岐履歴を使用する 2 レベルの適応型分岐予測方式が相次いで発表されている．1991～1993 年にかけて，イェー（T-Y. Yeh）とパットは，ある分岐命令自身のローカルな過去の分岐履歴をもとにカウンタテーブルを検索する PA（per-address adaptive）分岐予測方式を基本とした方式を論文発表している．これに対して，ほかの分岐命令の分岐とのグローバルな相関を用いた分岐予測方式も考案されている．1992 年には，グローバルな分岐履歴と分岐アドレスの組からカウンタテーブルをアクセスする GAs 予測機構が，更にグローバル履歴のみからアクセスする GAg 予測機構がそれぞれ論文発表された．1993 年には，マクファーリングが gshare 分岐予測方式を提案した．これは，分岐アドレスとグローバルな分岐履歴との排他的論理和をとってカウンタテーブルをアクセスすることにより，テーブルサイズを抑えつつ長い分岐履歴を活用できるようにしたものである [14]．

(6) データ投機方式の発展

本章 2-2-1(5) の制御ハザードと並んで，データハザードが，逐次コードを並列実行する際の大きな課題となる．これは，逐次的に並んだ命令間で，データの定義（Write）と使用（Read）の間の依存関係が存在する場合に発生する．その関係を分類すると次のようになる．

(i) Read-After-Write（RAW）：書き込まれたデータを読み出す

(ii) Write-After-Read（WAR）：読み出したデータに書き込む

(iii) Write-After-Write（WAW）：書き込まれたデータにデータを書き込む

(ii)と(iii)の依存関係は，プロセッサのレジスタ数が有限であるために，コンパイラによって同じレジスタが再利用されることによって生ずるものであり，本章 2-2-1(4) で述べたレジスタリネーミングによって解消することができる．

一方，(i)は，逐次コードの意味上存在する依存関係であって真の依存関係と呼ばれ，この依存関係を破って，定義される前にその値を読み出して使うことはもとのプログラムの意味を損なうことになる．これに対して考えられた方式が，実際に書き込まれる前に，書き込まれるであろうデータの値を予測して使用する，いわゆるデータ投機方式である．

データ投機方式の起源となったのは，1982 年にハービソン（S. P. Harbison）によって発表された値キャッシュ（value cache）である．これは一度計算した結果を値キャッシュに保存

しておき，同じ計算を検出することにより，前の計算結果を再利用（reuse）するものである．この提案の骨子は，データの再利用であり，投機の概念はないが，実行の高速化のために，過去のデータを利用することに着目し，コンパイラとアーキテクチャの協調作業によって実現する方法を提案した点は，後のデータ投機に通じるものがある．

データ投機の考え方が大きく進展したのは，1990 年代になってからである．これは，命令レベル並列性を追及する過程で，データハザードがパイプラインをストールさせることによる性能低下の影響が大きいことによる．1996 年の論文でリパスティ（M. H. Lipasti）らは，記憶装置のあるアドレスにおいて以前の値が繰返し現れる可能性が高いことを示し値の局所性（value locality）と名付けた [15]．データ投機方式はまだ実用化段階にはないが，歴史的経緯を把握しておく価値のある技術である．

(7) キャッシュメモリのオンチップ化

キャッシュメモリの起源については，本章 2-1-3(9) に述べた．1980 年代には，マイクロプロセッサからメインフレームコンピュータまで，幅広くキャッシュが使用されるようになった [16]．これはプロセッサの動作速度と主記憶として使用される DRAM の動作速度のギャップが大きくなったことによる．例えば，1985 年に発表されたインテル 80386 マイクロプロセッサは，クロックスピードが 16～33 MHz であった．これに対して当時の DRAM のアクセスレイテンシは 120 ns 程度と低速であったため，SRAM を用いた少容量のキャッシュが外付けされた．したがって，1980 年代半ばのマイクロプロセッサは，整数演算ユニット，浮動小数点演算ユニット，キャッシュコントローラ，及びキャッシュメモリに使用する SRAM をそれぞれ別々のチップとし，これらの集合体によって形成されていた．

その後，チップ上のトランジスタ数の増大に伴い，1989 年に発表された 80486 は，8 K バイトのレベル 1（L1）キャッシュをチップ上に内蔵した．1995 年に発表された Pentium Pro は，CPU を 0.35 μm プロセスで，L2 キャッシュ（容量 256 K または 512 K バイト）を 0.6 μm プロセスでそれぞれ製作して 1 チップ上に統合している．

以降，キャッシュメモリの多階層化とオンチップ化の流れが続いている．

(8) コプロセッサ及び付加プロセッサよる性能向上

コプロセッサとは，メインの CPU の処理能力が不足するときに，これを強化するために，メイン CPU と共に動作することを前提として付加されるプロセッサである．1980 年代において，特に普遍性が高いコプロセッサとして，浮動小数点演算を目的としたコプロセッサがある．

1980 年に発表されたインテル 8087 プロセッサは，16 ビットプロセッサ用の浮動小数点計算高速化を目的としたコプロセッサで，約 50 KFLOPS の演算性能をもっていた．この後継機が 80287，80387，80487 コプロセッサである．80486 DX 以降のプロセッサは，チップ上に CPU コアとともにコプロセッサを内蔵している．同様にモトローラ 68000 ファミリに対しては，68881/68882 コプロセッサが浮動小数点演算のアクセラレータとして使用されている．1984 年には，WEITEK 社がチップ当たり 1000 ゲート以上を実装することにより，9 つのチップセットで，64 ビットの浮動小数点乗算（7 ステージ）と加算（6 ステージ）の演算パイプラインを 1 クロック 125 ns で実現した．

その後，チップ上のトランジスタ数の増大に伴い，浮動小数点演算ユニットとしてのコプロセッサはメイン CPU とともに単一チップに内蔵されるようになった．

コプロセッサがチップ単位の付加プロセッサであるのに対して，1980 年代には，更に独立性の高い配列処理用の付加プロセッサ（attached processor）が開発され使用されている．本章 2-1-4(7) でも述べた AP-120B は，ミニコンピュータあるいはメインフレームコンピュータをホストマシンとする付加プロセッサである．1980 年に Floating Point Systems 社は，FPS-164 を発表した．これは，主記憶が 15 K 語（1 語 64 ビット）に拡張可能であり，これによってホストコンピュータが必要なくなった．1984 年には，FPS-164 に MAX（Matrix Algebra Accelerator）ボード（各ボードは二つの FPS-164 に相当）が 15 枚まで追加可能となり，全体でピーク性能 341 MFLOPS となった．これは 31 個の FPS-164 に相当する．

(9) ベクトルスーパーコンピュータの性能向上

本章 2-1-4(10) に述べたように，1975 年に発表された CRAY-1 が 1976 年に初めて導入された．CRAY-1 は，ベクトルを処理する命令を備えていたことからベクトルコンピュータと呼ばれ，160 MFLOPS の処理性能をもっていた．CRAY-1 は歴史的意義が大きなコンピュータとして本編 3 章 3-8 節で取り上げてあり，詳細はそちらを参照．1982 年に発表された CRAY X-MP は，2 台の CRAY-1 が 2 M ないし 4 M 語の共有メモリを使って動作した．各 CPU がメモリに 4 ポートをもち，内一つを入出力に，三つをベクトルレジスタで使用するため，メモリバンクで衝突がなければ，二つのベクトル入力と一つのベクトル出力が同時にメモリをアクセスできる．1984 年には，4 CPU モデルが発表された．最上位機種 CRAY X-MP/48 は，ピーク性能 840 MFLOPS をもち，64 バンクにインターリーブされた 8 M 語（64 ビット／語）の ECL バイポーラメモリをもっていた．1981 年には，不活性の液化フッ化炭素（inert liquid fluorocarbon）による液体冷却を用いた CRAY-2 が発表された．CRAY-2 は，冷却能力の向上によって実装密度が向上し，CPU を小さくすることが可能になり，最大四つの CPU をもつとともに，クロック速度 4 ns で動作し，4CPU での最大演算性能 2 GFLOPS を達成している．

CRAY-1 以外の流れとして，CDC STAR 100 の流れをくむ CYBER 203E（後に CYBER 205 と改名）がある．CRAY-1 との違いは，ベクトルレジスタをもたず，すべてのベクトル命令の処理を，主記憶によって行った点にある．CDC は更に ETA システムズ社を立ち上げて CYBER 205 の後継機である ETA マシンを開発した．

日本におけるベクトルプロセッサとしては，1982 年に発表された富士通の FACOM VP-100 と VP-200 がある．これらは，上記 CRAY-1 と CYBER 205 の特徴を踏まえて設計されている．CYBER 205 と同様にスカラ用とベクトル用の別々の演算ユニットをもつが，CRAY-1 と同様にベクトルレジスタをもつ．ベクトルユニットのクロック速度は 7.5 ns で，VP-200 のピーク性能は 533 MFLOPS である．同じ 1982 年には，HITACHI S-810 model 10/model 20（最大 630 MFLOPS）が発表され，1983 年には NEC SX-1/SX-2（最大 1300 MFLOPS）が相次いで発表された．

IBM は，1985 年に System/370 シリーズにおいて IBM 3090 スカラプロセッサに付加可能なベクトル処理ユニット（IBM 3090-VF）を発表した．

(10) SIMD データ並列コンピュータ

ベクトルスーパーコンピュータが SIMD パイプライン方式の並列コンピュータであるのに対して，同じ SIMD 並列を空間的なデータ並列処理に適用したコンピュータもこの時期活発に研究開発されている．

1983 年に Goodyear Aerospace 社が SIMD 型並列計算機 Massively Parallel Processor（MPP）を NASA Goddard に納入した．MPP は，本章 2-1-4(11) に述べた STARAN の設計思想を引き継いでおり，132×128 に構成された 1 ビットの要素プロセッサを 2 次元に結合している．アレイ制御ユニット（Array Control Unit）が，全要素プロセッサと要素プロセッサ間の転送制御を行う．マシン設計の目的は，画像処理であり，毎秒 10^6 画素を処理することを目標としていた．要素プロセッサはそれぞれ算術論理演算装置をもっており，8 要素プロセッサが一つのカスタム LSI に実装された．MPP の動作速度は，10 MHz である．初期のテストにおいて，ランドサット衛星画像を処理するのに DEC VAX 11/780 が 7 時間掛かった処理を 18 秒で行っている．MPP が一般の使用に供されたのは，1985～1991 年の間である．

1981 年に，ヒルズ（W. Daniel Hillis）が，人工知能の課題に対応するには高度の並列性が必要であることを主張し，この主張に沿った Connection Machine の構想を MIT 人工知能研究所のメモとして発表した [17]．最初の Connection Machine プロトタイプは CM-1 である．CM-1 では，ホストプロセッサがマクロ命令を使って指令し，これをマイクロコントローラがマイクロ命令によって解釈して，ナノ命令を生成し，これを全要素プロセッサに放送して要素プロセッサの実行を制御しており，全体としては SIMD アーキテクチャとなっている．プロセッサセルと呼ばれる要素プロセッサは，ビットシリアルな演算を行う．CM-1 専用に設計された VLSI チップは，CMOS テクノロジを使い，約 1 cm^2 のチップ上に作られており，16 個のプロセッサセル，通信ネットワーク用ルータ，及びナノ命令をデコードしてプロセッサセルとルータの動作を制御するコントロールユニットからなる．動作速度は 4 MHz，消費電力約 1 W である．全体として，64 K プロセッサを，*n* 次元ハイパーキューブネットワークで結合している．

CM-1 の経験を踏まえて，更に CM-2 が作られている．CM-2 が CM-1 と際立って違っているのは，ビットシリアル処理の弱点を補強するために，浮動小数点演算アクセラレータを付加している点である．また，データボールト（Data Valt）と呼ぶ大容量 2 次記憶システムが用意されている．また，CM-2 を使いやすくするために，PARIS と呼ぶ各種のプリミティブをフロントエンドマシンから使えるようにして，C*などの高水準の並列プログラミング環境を支援している．

(11) MIMD 共有メモリ型並列コンピュータの普及と実験機の試作

MIMD 型の並列コンピュータがメモリを共有したアーキテクチャは，プロセッサ数のスケーラビリティに制約があるものの，共有メモリを対象としているため従来のプログラミングの延長上にあり，プログラミングが容易であるという特徴から，一般への普及が進むとともに，種々の並列コンピュータが研究開発されている．

本章 2-1-4(2) に述べたように，1970 年代には，C. mmp，Cm*に代表される共有メモリ型マルチマイクロプロセッサの研究が行われた．これらを起源として，1985 年中頃には，本章 2-1-1(3) に述べたように 32 ビットマイクロプロセッサが相次いで発表され，マイクロプロ

セッサの性能が急速に増大したことを踏まえて，10～20 台程度のマイクロプロセッサをバス結合し，共有メモリを配置した共有メモリ型のマルチマイクロプロセッサが試作された．

1982 年にニューヨーク大学（NYU）のゴットリーブ（Allan Gottlieb）らによる Ultracomputer が論文発表された．1990 年代の技術を想定して数千の要素プロセッサからなるコンピュータが設計されたが，Ultracomputer の特徴は，要素プロセッサ間を結合する Omega ネットワークにおいて，同期のためのプリミティブ fetch-and-add 操作を効率よく実現する機構にある．すなわち，同一の共有メモリ上の変数に対する複数の fetch-and-add 操作がネットワーク上で競合した場合，これらをまとめて一つの fetch-and-add 操作に置き換える．

本章 2-1-4(13) に述べたように，HEP は，1974 年にスミス（Burton Smith）によって設計が開始されたが，1978 年に論文発表されていた．1982 年に Denelcor 社によって最初の HEP が製作され，Ballistics Research Laboratories に納入された．Denelcor HEP では，四つのプロセス実行モジュール（PEM）がそれぞれローカルなプログラムメモリをもち，データメモリを共有して実行を進める．相互の通信は，スイッチ結合網によって行われる．除算以外の命令は，8 段の命令パイプラインによって実行される．HEP においては，最大 50 までの命令ストリームが CREATE 機械命令（あるいは FORTRAN コマンド）によって作り出せ，各命令ストリームからの命令が，8 段の命令パイプラインに投入される．クロック速度は 100 ns なので，パイプラインを有効な処理で埋められれば，PEM 当たり 10 MIPS の性能を出すことができる．HEP の実装は，ECL 10K 技術を用いて行われた．

前節で SIMD マシン，本節で MIMD マシンの例をあげたが，SIMD/MIMD だけでは分類しきれないマシンもある．1980 年に宇都宮大学の馬場らにより発表された MUNAP[18] は，4 台の要素プロセッサ中の四つのナノプログラムの開始番地を同時に指定することによって，並列処理を柔軟に制御することを特徴とする．マイクロプログラム記憶，ナノプログラム記憶は，共に書換え可能であるから，マイクロ命令の指すナノ命令あるいはナノプログラムの内容が同一であれば SIMD 処理を行え，異なる場合には MIMD 処理が行える．この指定は，マシンサイクルごとに行える．また，ナノプログラムの終了は，ナノ命令に付けた終了ビットにより任意のナノ命令で行うことができる．マイクロプログラムレベルで見ると単一命令流であるが，ナノプログラムレベルで見ると要素プロセッサごとに独立な命令流によって自律的に動作可能であり，複数命令流となる．

(12) MIMD 分散メモリ型並列コンピュータ

分散メモリ型並列コンピュータは，スケーラビリティに優れるため，高並列コンピュータを目指した研究開発が活発に行われている．

1985 年にカリフォルニア工科大学（Caltech）のザイツ（Seitz）が論文発表した Cosmic Cube は，ハイパーキューブ結合された 64 個のマイクロプロセッサからなる．各ノードは，インテル社の 8086 と先のコプロセッサの項(8)で述べた 8087 とからなり，128KB RAM と 8KB ROM からなるローカルメモリをもつ．各ノードは，ハイパーキューブトポロジに従って六つの隣接ノードと接続され，ノード間の通信は，2 Mbit/s のメッセージ通信によって行われる．8087 は最大性能 50 KFLOPS なので，64 ノード全体では 3 MFLOPS となる．Cosmic Cube は，1983 年から稼動している．

このほか，インテル社の iPSC（1985 年発表）や nCUBE 社の nCUBE が，ハイパーキュー

ブ結合を使用した MIMD 分散メモリ型並列コンピュータとして相次いで発表された．iPSC の各ノードは，インテル社の 80286 マイクロプロセッサと 80287 コプロセッサから構成されている．一方，nCUBE は，専用の 32 ビット VLSI 化プロセッサを用いており，各ノードのピーク性能は 0.3 MFLOPS であった．

1984 年には，Inmos 社（1979 年設立）が Transputer を発表している．Transputer は，相互に接続して並列処理を行うことを前提に設計されており，OCCAM と名付けた並列言語が同時に用意されている．例えば，T800 Transputer は，32 ビットマイクロプロセッサ，浮動小数点コプロセッサ，4 K ビットオンチップメモリ，他ノードとの通信のための四つのシリアルリンクをもっている．

分散メモリ型並列コンピュータの中で，特異な存在が，1978 年にクーン（H. T. Kung）とライザーソン（Charles E. Leiserson）によって論文発表されたシストリックアレイ（systolic array）である．シストリックアレイにおいては，ネットワーク接続されたデータ処理ユニット間を，データがパイプライン式に流れ，データの到着がデータ処理ユニットの処理を駆動する．データ流は一つとは限らず，一般的には，複数のデータ流が，ネットワーク中を独立な方向に流れる．データ処理ユニット間のデータ授受が非同期的なハンドシェイクによって行われるものをウェーブフロントアレイ（wavefront array）と呼ぶ．シストリックアレイは，インテル社によって iWarp プロセッサとして実現された．これは双方向のバスによって 1 次元アレイ状に接続されている．処理の仕方からも明らかなように，シストリックアレイは，規則的なデータ依存関係をもった問題に対して空間的，時間的な並列性を活用できて，性能を発揮することができる．シストリックアレイによる処理は，その後，再構成可能プロセッサに適合した処理の方式として取り入れられている（6 群 5 編 6 章参照）．

(13) データフローマシンプロトタイプの試作と実用化の検証

本章 2-1-4(9) に述べたように，データフローマシンアーキテクチャの提案は，1974 年にデニス（J. Dennis）によって行われた．これは静的データ駆動方式に分類される．1983 年には，静的データ駆動方式に基づく実験システムが試作された．一方，動的データ駆動方式に基づくデータフローマシンについては，1980 年に，アービンド（Arvind）による tagged token を用いたデータフローアーキテクチャの提案があった．また，1983 年に，ガード（J. Gurd）による動的データ駆動方式に基づくデータフローマシンプロトタイプが試作されている [19), 20)]．

1982 年には，元岡らにより手続きレベルでデータ駆動方式を取り入れたマルチマイクロプロセッサが試作されている．

これら 1980 年代初頭にかけてのデータ駆動方式並列コンピュータのプロトタイプ試作の時期を経て，1980 年代後半には，データ駆動型コンピュータの実用性の検証が行われた．この時期に，試作されたデータ駆動型コンピュータには，1987 年の電子技術総合研究所における SIGMA-1 などがある．

データ駆動型コンピュータは，その後，命令レベルでデータ駆動方式を追求することのオーバヘッドを克服するために，従来のプログラム内蔵方式コンピュータとのハイブリッド型，マルチスレッド実行モデルの導入など，プログラム内蔵方式コンピュータとの融合が図られている．この中には，1989 年に発表された電子技術総合研究所の EM-4 や，1990 年に発表された MIT の Monsoon がある．

データ駆動方式コンピュータは，いわば，細粒度の命令レベルではスーパースカラ方式に，中粒度はマルチスレッドコンピュータに，粗粒度はメッセージ駆動コンピュータに，その実行方式の精神が受け継がれたと見ることができる．

(14) ネットワークトポロジ及びメッセージルーティングの新方式提案

並列コンピュータを構築する上で重要な構成要素がネットワーク（相互結合網）である．また，ネットワーク上の通信方式も性能に大きな影響を与える．このため，1980 年代には，並列コンピュータのノード間を結合するネットワークトポロジの研究，及びネットワーク上の通信方式の研究が活発に行われるようになった．例えば，本章 2-2-1(12) で述べたハイパーキューブは，代表的なトポロジの一つである．一方，通信方式に関しては，1985 年ダリ（Dally）とサイツ（Seitz）がワームホールルーティング方式を提案した．ワームホールルーティングは，メッセージパケットを固定長のフリットに分割し，送り元から送り先までを，フリット単位で同一の経路でパイプライン式に転送する．この方式によって，メッセージの総通信時間に及ぼす転送距離の影響を抑えることができる．このことは，ネットワークトポロジの工夫による転送距離の削減の効果を小さくする方向に働くこととなり，実装の容易性と相まって，2 次元メッシュなどの簡明で実装しやすいトポロジが使用される流れを生んだ．更にダリは，仮想チャネル（virtual channel）を考案し，仮想チャネルによってデッドロックしないルーティングが可能であることを示した．

(15) フォールトトレラントコンピュータシステム

1970 年代末から 1980 年代に掛けて，高信頼化を目指すフォールトトレラントコンピュータシステムが開発され，また商用化が進んだ．フォールトトレラントコンピュータの基本は多重系であり，2 重系，3 重系といったフォールトトレラントシステムの考え方が，コンピュータにも取り入れられている [21), 22), 23), 24)].

1977 年に，タンデムコンピュータズ社が発売したタンデムノンストップシステムは，高い可用性を目的とした初めての商用フォールトトレラントコンピュータシステムである．これは，単一要素のハードウェア故障によってシステム全体は停止しない，システムを稼動させたままハードウェアの保守・修理ができる，などを特徴とする．これを実現するため，2～16 台のプロセッサモジュールを結合してシステムを構成し，プロセッサ間結合バス，入出力制御装置，電源装置などを 2 重化して，故障の影響が単一要素外に及ばないようにしている．

1977 年に NASA により打ち上げられた Voyager に搭載された JPL STAR（Self-Testing-And-Repairing computer）は，バックアップ用のメモリアレイをもち，メモリエラーを検出して回復することができる．このコンピュータは，現在でも動作中である．

このほか，1982 年にストラタスコンピュータ社から発売された STRATUS S/32 も初期のフォールトトレラントコンピュータシステムの代表例である．

2-2-2 1990 年代

1990 年代においてもマイクロプロセッサは，命令レベル並列性と動作周波数の向上により，引き続き性能向上が図られた．また，RISC の普及が進むと同時に RISC，CISC いずれが優れているのかの議論も行われた．命令レベル並列性が追及される流れの中で，その限界がどこ

にあるのかについて実験的な検証が行われ，仮に無限のハードウェアを投入したとしても命令レベル並列性の抽出には限界があることが明らかになり，複数プログラムあるいは複数スレッドを対象とした高性能化へと進むことになった．並列コンピュータの世界では，カスタムプロセッサの優位性がなくなり，SIMD 並列コンピュータが次第に姿を消し，一般に普及したマイクロプロセッサを使用した並列コンピュータが大勢となる．クラスタコンピュータもこのような流れの上にある．並列コンピュータのノード数に関するスケーラビリティと，共有メモリモデルのもつプログラミングの容易さを両立させるために，キャッシュコヒーレントな非均質メモリアーキテクチャ（NUMA：Non Uniform Memory Architecture）が提案され，以降の高度に並列なコンピュータのメモリアーキテクチャに採用されるようになっている．

(1)　命令レベル並列性と動作周波数の向上による高性能化

1990 年代のプロセッサ設計の目標は，同時に発行する命令数を多くして命令レベル並列性を最大限に活用しようとすることであった [25]．スーパースカラ実行機構により，同時に発行された複数の命令は，割込みが発生した際には本章 2-2-1(4) に述べたリオーダバッファにより逐次実行と同じ結果を残す．また，動作周波数向上のためには，各ステージでの動作を細分化して，ステージ当たりの仕事を少なくすることが有効であり，これがパイプラインの段数を増加させる流れとなった．これを，スーパーパイプラインアーキテクチャと呼ぶ．

このようなアーキテクチャが性能を発揮するためには，本章 2-2-1(4) に述べたように命令パイプラインをストールさせることなく，スムーズに実行が進む必要がある．このためには，まず制御フローを正しく予測するために，本章 2-2-1(5) に述べた分岐予測器による予測精度を上げる必要がある．初期のスーパースカラプロセッサである DEC Alpha 21064（1992 年），Pentium（1993 年），MIPS R8000（1994 年），IBM POWER（1990 年）は，1 ビットあるいは 2 ビットのカウンタによる予測を行っている．Pentium Processor with MMX Technology（1996 年）以降のマイクロプロセッサでは，更に精度を上げるために 2 レベルの分岐予測器が使われている．

キャッシュミスもパイプラインをストールさせる原因となるので，キャッシュメモリのアーキテクチャも進化している．その進化の過程は，アクセス時間の短縮とともに，容量と階層数の増加に表れている．

表 2・1　インテル社の 1990 年代のプロセステクノロジ

名　称	発表年	プロセス テクノロジ	キャッシュ容量	トランジスタ数	動作周波数
Pentium（P5）	1993	0.8 μm	8KB - IL1, 8KB - DL1	3.1 M	60～66 MHz
Pentium Pro	1995	0.6 μm	8KB - IL1, 8KB - DL1	5.5 M※1	150 MHz
Pentium II （Klamath）	1997	0.35 μm	16KB - IL1, 16KB - DL1 512KB – L2	7.5 M	233～300 MHz
Pentium III （Katmai）	1999	0.25 μm	16KB - IL1, 16KB - DL1, 512KB - L2	9.5 M	450～600 MHz

※1　256 KB L2 キャッシュを別のダイにもつ．

例えば，同じインテル社についてこの年代のプロセステクノロジ，キャッシュ容量，総トランジスタ数，動作周波数の変遷を辿ると**表 2･1** のようになる．同じシリーズでもプロセステクノロジを進化させて性能向上を行っており，表 2･1 は最初に発表された製品について示した．

これらのデータは，CMOS テクノロジの微細化技術の発展と共に，プロセッサの動作周波数が高くなり，総トランジスタ数が増加していること，大容量のキャッシュメモリがチップ上に実装されていることを示している．これらの傾向は，2000 年代に入り，命令レベル並列性の抽出と動作周波数の向上が限界に近づくまで続くことになる．

(2) CISC 対 RISC

1980 年代初頭に提案された RISC は，命令形式を簡単にして，1 命令 1 サイクル実行を基本とし，また，メモリアクセスはロード･ストア命令に限定している．1990 年代においては，RISC の世界でも，MIPS R4000（1991 年），DEC Alpha（1992 年），UltraSPARC（1995 年）といった 64 ビットプロセッサが一般的になっている [26]．

このような背景のもとに，1990 年前後には，CISC と RISC の優位性についての議論が行われている [27]．CISC の流れのもとになったのは，上位互換性であり，一つのファミリープロセッサにおける機能の拡張が新たな命令の追加につながって命令形式を複雑にしている．追加される命令の多くは，コンパイラあるいはオペレーティングシステムの要求に基づくものであるが，結果として，複雑な命令は制御回路の複雑化を招き，またコンパイラにとっても使いにくい命令を増やす結果となった．この反省に立って，RISC は，少数の簡単な命令形式を採用し，制御部も結線論理制御で実現しやすくして，命令パイプラインによる高速化を図った．結果として，コンパイラの役割が大きくなり，また命令パイプラインをストールさせないようなコード最適化技術が重要になった．

要求される機能を直接ハードウェアに反映することを避け，コンパイラの役割を重視して，プリミティブな命令を組み合わせて要求される機能を実現しようとする RISC の精神はスマートに見える．これはまた，一面，先に述べたようにダイナミックマイクロプログラミングの精神にも通じるものがある．

ただし，従来の命令セットとの互換性を重視するCISCにおいても本章 2-2-2(1) のPentiumに見られるようにその動作周波数やメモリアーキテクチャは発展しており，マイクロアーキテクチャ的には，次第にRISCと共通性が高いものになっている．すなわち，スーパースカラ技術による実行を支援するハードウェア機構，制御フローによる乱れを防ぐための分岐予測機構，大容量のキャッシュメモリなどが両者に共通のハードウェアユニットとして備えられるようになり，機械命令を実現するマイクロアーキテクチャには大きな違いがなくなっている．ただし，CISCプロセッサについては，機械命令からマイクロ操作にマッピングするためのユニットが設けられているのが特徴的である．

(3) 命令レベル並列性の限界論の台頭

本章 2-2-2(1) に述べたように命令レベル並列性を向上させるには，同時に発行する命令数を多くすることが必要である．しかし，多くの命令を同時に発行しても，分岐予測ミス，命令間のデータ依存関係，キャッシュミスなどにより，並列実行がスムーズに進まなけれ

ば，性能向上には結びつかない．このため，より精密な分岐予測機構による予測ミス率の減少，レジスタリネーミングによる偽の依存関係の解消，キャッシュメモリ容量の増大によるキャッシュミス率減少などを更に進める必要がある．では，チップ上のトランジスタ数の増大を背景として，これらの阻害要因の解消を極限まで目指したとして，一体，抽出可能な命令レベル並列性の限界はどの辺にあるのだろうかという疑問が湧いてくる．もし，阻害要因を極限まで解消しても，単一スレッドから抽出できる命令レベル並列性がそれほど向上しないのであれば，阻害要因の解消に多大なトランジスタ資源を投入することは割に合わないこととなる．

この課題に対して，1989~1992 年の間に，命令レベル並列性の限界について，実験結果を踏まえての議論が活発に行われている（参考 : 1989 年のジュピ（Jouppi）とワォール（Wall）[28]，1991 年のジョンソン（M. Johnson）[29]，1992 年のラム（M. Lam）[30]などである）．

これらの実験においては，分岐予測などが理想的に行われたとした場合の性能向上を調べることにより，命令の発行数がいくつなら性能向上が達成できるかを検証している．例えば，1991 年にジョンソンは著書の中で，ハードウェアが無制約で理想的なスーパースカラ実行が行われ，分岐予測も完全に成功したという前提において，実験結果を示している．これによって，サイクル当たりの命令発行数が 4 ウェイのスーパースカラであれば性能向上が期待できるが，4 ウェイを超えたところで性能向上が飽和してしまうことを示している．すなわち，単一の逐次コードの高性能化を考えたとき，仮にハードウェア資源が無尽蔵にあったとしても，4 命令を越える発行は性能向上に結びつかないことを明らかにした．

(4) 単一プログラム，単一スレッドから複数プログラム，複数スレッドの高速化へ

単一の逐次コード中の単一スレッドから命令レベル並列性を抽出することの限界が示されたことにより，複数スレッドを対象として並列実行の可能性が検討され，具体的なアーキテクチャの提案がなされた．この中で，大きな影響を与えたのが，1995 年に論文発表された Multiscalar プロセッサ [31] と同じ 1995 年に論文発表された Simultaneous Multithread（SMT）プロセッサ [32] である．両者とも複数スレッドを同時に処理する点は共通であるが，以下に述べるとおり，Multiscalar が投機的なマルチスレッド処理を行うのに対し，SMT は非投機的なマルチスレッド処理を行う．

Multiscalar プロセッサは，単一プログラムをタスクと呼ぶサブプログラムに分割し，これらを要素プロセッサに割り付けて実行することにより，スレッドレベルの並列性を抽出する．分割したタスク間には逐次的な順序関係があることを考慮して，要素プロセッサは単方向のリング状ネットワークで接続する．連続するタスクは，隣り合う要素プロセッサに割り付けて，タスク間の単方向通信によって先行タスクの制御情報やレジスタ値を転送する．メモリへの読み書きも，逐次的な実行の順序関係を保てるようにいったんバッファに書き込み，実行順序に合わせて無効化やコミットなどの操作を行う．全体の親スレッドの実行以外のスレッドの実行は投機的であり，メモリにおける依存違反によって投機は失敗となる．

Multiscalar プロセッサが，一つの要素プロセッサに割り当てるタスク（すなわちスレッド）を一つに限定しているのに対し，SMT プロセッサは，一つのプロセッサに複数のスレッドを割り当てて，スーパースカラ方式により実行する．サイクルごとに，複数のスレッドから取り出した命令を同時に発行することを可能とすることにより，ハードウェア資源の有効利用

を図るとともに，1 スレッドから抽出可能な命令レベル並列性の限界を打破することが目標である．SMT プロセッサの設計思想は，複数スレッドの同時実行により，例えば一つのスレッドがキャッシュミスでストールしたときにはほかのスレッドが空いたハードウェア資源を使用して，仕事を進めることができるという点で，スループット指向コンピューティングである．

(5) SIMD データ並列コンピュータの退潮

1990 年代初頭に相次いで商用の SIMD データ並列コンピュータが発表された．

1990 年に発表された MasPar 社の MP-1 は，SIMD 方式により 4 ビットプロセッサを制御する．Goodyear 社の MPP アーキテクチャを継承し，各プロセッサは 8 つの近傍プロセッサと結合されている．1992 年に発表された MP-2 は，1.0 μm 2 層メタル CMOS チップで作られ，消費電力 0.8 W，208 pin PQFP，12.5 MHz で動作した．

1991 年には，CM-1，CM-2 の流れをくむ CM200 が発表されている．

1990 年代初頭以降，SIMD データ並列方式をアーキテクチャの基本として採用した並列コンピュータは現れてない．その大きな原因は，汎用マイクロプロセッサの発展である．SIMD データ並列コンピュータは，並列コンピュータ全体の制御部の制御のもとに要素プロセッサが動作することにより，個々のプロセッサのハードウェアを簡単化できるのが利点であるが，一方，そのためにオリジナルプロセッサのハードウェアとともに，専用の言語処理系などのソフトウェアシステムを新規に開発することが必要になる．一方，汎用マイクロプロセッサは，ハードウェア的には台数効果があってコストパフォーマンスに優れた要素プロセッサが活用でき，かつ個々のプロセッサについてはソフトウェアシステムも完備している．これらのことから，オリジナルな SIMD プロセッサを新たにチップとして開発しても，汎用プロセッサに勝てなくなったというのが実状である．

ただし，このことは，SIMD データ並列演算が不要になったということではない．MPP の動機となった SIMD データ並列演算は，Intel 社の汎用マイクロプロセッサにおける MMX，SSE といった SIMD 命令に，あるいは GPU などのグラフィック処理専用プロセッサにおいて活用され，その有効性を実証している．

(6) 共有メモリバスマルチプロセッサの普及

1990 年代前半には，共有メモリバスによって 10～20 台のマイクロプロセッサを結合した共有メモリ型マルチマイクロプロセッサがビジネスサーバとして活用された．この背景には，RISC プロセッサを中心とするマイクロプロセッサの発展と，キャッシュコヒーレンスプロトコルの発展がある．キャッシュコヒーレンスプロトコルについては，1980 年代に研究開発されたイリノイプロトコルをはじめとするプロトコルが実用化され，高バンド幅で，スヌープキャッシュのコンシステンシをサポートするものが実現されている [33]．また，マイクロプロセッサの発展に伴って，プロセッサにもスヌープキャッシュコントローラが内蔵されるようになって（例えば，1990 年 MC 68040，1989 年 i860，1991 年 Super SPARC，1991 年 MIPS R4000 など），キャッシュコヒーレントな共有メモリ型マルチプロセッサが実現しやすくなっている．

(7) CC-NUMA 型メモリアーキテクチャによる高並列マシン上への共有メモリモデルの実現

共有メモリバスアーキテクチャは，共有メモリモデルを提供するため並列プログラムの作成が容易であるという特長がある．しかし，欠点はすべてのプロセッサが共通のバスにアクセスするため，バスネックになり，接続する要素プロセッサ数が 10～20 台程度に限定されることである．

1991 年に論文発表されたスタンフォード大学の DASH マルチプロセッサは，共有メモリモデルを，64 台の RISC プロセッサを用いたマルチプロセッサアーキテクチャ上に実現して，共有メモリモデルとスケーラビリティとの両立を図った [34]．これを実現したのは，ディレクトリベースのキャッシュコヒーレンスプロトコルである．DASH プロトタイプでは，まず，4 台の 33 MHz MIPS R3000/R3010 プロセッサをイリノイプロトコルによるスヌープバス方式により接続してクラスタを構成する．クラスタ同士は，リクエストメッセージ用とリプライメッセージ用の二つの 2D メッシュで結合され，ワームホールルーティングによるメッセージ転送を行う．クラスタごとに一つのディレクトリメモリが設けられ，主記憶のメモリブロックごとにディレクトリエントリが割り当てられて，shared/dirty を示すフラグと全クラスタに対応したビットベクトルをもつ．DASH では，リリースコンシステンシモデルが使用され，同期の解放時点においてのみそれ以前のメモリ要求がほかのプロセッサに対し完了することを保障する．これによって，同期による制約を緩め，並列処理のオーバヘッドを軽減する．33 MHz で動作する要素プロセッサに対して，L1 キャッシュアクセスに 1 クロック，L2 キャッシュアクセスに 15 クロック，クラスタ内のローカルメモリアクセスに 30 クロック，ほかのクラスタアクセスに 100-135 クロックと順次アクセスコストが増大する．

このようなメモリアーキテクチャを，キャッシュコヒーレント NUMA（Non-Uniform Memory Access）と呼び，CC-NUMA と略称する．CC-NUMA は，以降の高並列・超並列コンピュータのメモリアーキテクチャに大きな影響を与えた．

(8) 高性能マイクロプロセッサを用いた種々の高並列・超並列コンピュータの開発

1990 年代には，数百～千台規模のプロセッサからなる高並列・超並列コンピュータが活発に研究開発されている．この中で，明らかになった流れは，マイクロプロセッサの高性能化に伴い，最先端のマイクロプロセッサから 1，2 年遅れで，そのマイクロプロセッサを使用した並列コンピュータが現れていることである．また，この期間において，高性能コンピュータのトップの座が，並列ベクトルコンピュータから高性能マイクロプロセッサを多数並べた超並列コンピュータ（MPP : Massively Parallel Processing）へと入れ替わった．おおよそ，1991 年には，LINPACK（GFLOPS）性能において MPP が並列ベクトルコンピュータを追い越し，1994 年には，世界のトップ 500 コンピュータにおいて，MPP の数が，並列ベクトルコンピュータの数を追い越している．

以下，主要な MPP を年代順に概観する．

1991 年の Thinking Machines 社の CM-5 は，CM-2 のようにカスタムプロセッサではなく，40 MHz の SPARC を使用し，これをファットトリーと呼ぶネットワークで結合している．要素プロセッサ（PE : Processing Element）当たり 160 MFLOPS で 1024 PE でのピーク性能 160 GFLOPS である．

1991 年の Kendall Square Research 社 KSR-1 は，1 T バイトの仮想アドレス空間をもつ仮想共有メモリ方式並列コンピュータである．カスタムスーパースカラプロセッサを PE として，PE 当たり 80 MFLOPS，80 PE でピーク性能 6.4 GFLOPS である．DASH のようなホームメモリの概念をもたず，要求に応じてデータが移動する COMA（Cache Only Memory Architecture）と呼ばれるメモリアーキテクチャが特徴である．

1992 年の富士通 AP 1000 は，50MHz の SPARC を 2 次元トーラスで結合しており，1024 PE 結合した場合のピーク性能は，8.5 GFLOPS である．後継機として，AP 1000+，AP 3000（1996 年）がある．

1992 年には，ダリ（W. J. Dally）によりオリジナルプロセッサを用いた J-Machine が開発されてその成果が報告されている．1988 年に開始された J-Machine プロジェクトの中心となるアイデアは，メッセージ駆動計算モデルに基づいたアーキテクチャの設計であり，粗粒度なデータフローアーキテクチャと位置付けることもできる．1990 年に宇都宮大より発表された A-NET は，並列オブジェクト指向を核概念として，言語とマルチコンピュータアーキテクチャを統合的に設計・開発している点で，J-Machine と通じるものがある．

1993 年の NEC Cenju-3 は，75 MHz の VR 4400SC をベースライン網で結合しており，PE 当たりの性能は 50 MFLOPS，256 PE でピーク性能 12.8 GFLOPS である．後継機である 1997 年の Cenju-4 は，200 MHz の VR 10000 を多段網で結合しており，PE 当たりの性能は 400 MFLOPS，1024 PE でピーク性能 400 GFLOPS である．

1993 年の Intel 社 Paragon XP/S は，i860 XP を 2 次元メッシュ結合したもので，PE 当たりの性能は 75 MFLOPS，1840 PE でピーク性能 138 GFLOPS である．

1995 年 MIT で試作された Alewife は，試作段階では 32～128 PE を 2 次元メッシュで結合している．各 PE は，33 MHz で動作する Sparcle と呼ぶオリジナルなノードを中心に，FPU，64 KB ダイレクトマップキャッシュ，4 MB のグローバル共有メモリから構成されている．Alewife の特徴は，キャッシュコヒーレント分散共有メモリとユーザレベルメッセージ通信を同一ハードウェア上に実現していることである．

1996 年の Silicon Graphics 社の Origin 2000 は，195 MHz の MIPS R10000 x 2 を 1 ノードとし，ハイパーキューブ網で結合しており，ノード当たりの性能は 780 MFLOPS，64 ノードでのピーク性能が 50 GFLOPS である．CC-NUMA メモリアーキテクチャを採用している．

1996 年の Intel 社 ASCI Red は，9216 個の 200 MHz Pentium Pro と 590 GB のメモリにより構成されたマシンで，最初に 1 TFLOPS の壁を越えた．

1996 年の CRAY Research 社の T3E-900 は，DEC 21164A を 1 ノードとし，3 次元トーラス網で結合しており，ノード当たりの性能は 900 MFLOPS，2048 ノードでのピーク性能が 1843.2 GFLOPS である．

(9)　コモディティ技術を組み合わせたクラスタコンピュータの発展

クラスタコンピュータは，1990 年代に現れた高性能コンピュータの構成方式の一つであり，市販のコモディティ（commodity）ハードウェアとソフトウェアを組み合わせることにより，高性能コンピュータを低価格で実現することを目指したものである [35), 36)]．

1993 年に構想され，1994 年から開始された Beowulf プロジェクトが起源とされ，1994 年には，16 台の 80486 搭載 PC を用いて，1 GFLOPS のクラスタコンピュータを試作している．

ネットワークには，10 Mbps Ethernet を使用し，メッセージ通信は MPI を用いている．

1992 年から開始された RWC プロジェクトにおいては，PC クラスタ構築のために，SCore と呼ぶグローバルオペレーティングシステムが研究開発された．これは，高度な通信機能とともにクラスタ全体を単一システムとして扱えるように工夫されている．ネットワーク接続は Myrinet を使用し，MPICH-SCore と呼ぶ SCore 版 MPI を提供する．

UC Berkeley で試作された NoW（Network of Workstations）は，1996 年に最速のソーティングを達成し，1997 年に LINPACK ベンチマークで 10 GFLOPS を達成している [37].

2-2-3 2000 年代

2000 年代のコンピュータの特徴は，性能とともに，消費電力，信頼性・安全性といった要素が多面的に要求されるようになったことである．単一プロセッサの性能を上げようとすると，動作周波数を高めることが必要となり，命令レベル並列性を抽出するためにハードウェア機構が複雑になる．これらは，いずれも消費電力を上げることにつながる．低消費電力化のための技術はデバイス技術からアーキテクチャ技術まで多岐にわたるが，マイクロプロセッサの世界で見ると，オンチップのマルチコアプロセッサ構成が一つの解となっている．また，信頼性・安全性が強く求められるようになっている．これは，コンピュータが社会の隅々まで浸透したこと，またコンピュータ同士がネットワークを介して結合されていることなど，コンピュータの及ぼす影響が非常に広範になっていることと深く関わっている．

(1) 命令レベル並列性と動作周波数の向上による高性能化の継続

2000 年代前半においては，単一プロセッサの性能を上げるために，1990 年代に引続き，命令レベル並列性と動作周波数の向上が継続的に進められた．この結果，2000 年には，最先端のマイクロプロセッサの動作周波数が 1 GHz に到達した．このような最先端マイクロプロセッサに共通する特徴として，次のようなことがあげられる．

(i) スーパースカラ技術による命令レベル空間並列性の抽出
(ii) スーパーパイプライン技術による命令レベル時間並列性の抽出と動作周波数向上
(iii) 実装技術の向上による動作周波数の向上
(iv) 制御フローを正確に予測するための分岐予測機構
(v) キャッシュ容量の増大と多階層化

このような流れに沿ったプロセッサの代表例として，2000 年に発表されたインテル社の Pentium 4（Willamette）を見てみると，0.18 μm プロセステクノロジで作られ，動作周波数 1.40～1.50 GHz，8 KB L1 データ及び 16 KB トレースキャッシュと，256 KB L2 キャッシュをもち，トランジスタ数 42 M，パッケージは 423 ピン PGA である．

その後，動作周波数は，2001 年に 1.7～2.0 GHz，2002 年に 2.2～2.4 GHz と増加し続けている．2005 年に発表された 64 ビットプロセッサ Pentium 4（Prescott）では，0.09 μm（90 nm）プロセステクノロジで作られ，動作周波数 3.8 GHz，2MB L2 キャッシュをもち，トランジスタ数 169 M である [38]. 命令パイプラインのステージ数はそれまでの 21 から 31 に増えている．

Pentium 4 では，CISC の複雑な形式をもった命令を，多数段の命令パイプラインによる処理にスムーズにのせることが課題となる．この対策として，命令をいったんμOP と呼ぶ内部命令に変換し，これをトレースキャッシュに置いて，デコード・実行へと進める．これは，

外見は CISC としておいて，内部では RISC に変換して，実質は RISC 命令を実行していることにほかならない．これによって，RISC の整った命令形式によるパイプライン実行のしやすさを実現している．

同様の流れにあるマイクロプロセッサとしては，2003 年にリリースされた AMD Athlon 64（AMD K8）シリーズなどがある．Athlon 64 シリーズの特徴としては，DDR2 SDRAM をサポートするメモリコントローラのプロセッサチップへの実装，TLB の大容量化，分岐予測機構の高性能化，バッファオーバフローなどに対する安全性向上のための No Execute bit の採用などがある．Athlon 64 プロセッサは，130 nm と 90 nm SOI 技術によって作られている．

(2) VLIW プロセッサと命令セット変換技術

(1)がスーパースカラキテクチャ技術をベースとしたマイクロプロセッサであるのに対して，VLIW アーキテクチャをベースとしたマイクロプロセッサの流れがある．

組込み型のプロセッサにおいて VLIW アーキテクチャが使用される場合には，既存の命令セットからの変換を前提とせずに，VLIW 命令をそのままソフトウェアレイヤに提示することが多い．この例としては，NXP1 社の TriMedia，Analog Devices 社の SHARC DSP，Texas Instruments 社の C6000 DSP ファミリ，STMicroelectronics 社の ST200 ファミリなどがある．

一方，既存のソフトウェア資産との連続性を重視する場合には，何らかの方法でこれを VLIW 命令に変換する必要がある．変換の方法としては，(i)高水準言語プログラムから再コンパイルする，(ii)命令レベルにおいて静的に変換する，(iii)命令レベルにおいて動的に変換する，などがある．(i)と(ii)は，いずれも静的に十分な処理時間をとってやることであり，従来の処理の延長で行える．これに対して，(iii)は，変換自体に要する時間も実行時間の中に組み込まれることが要点であり，特に 1990 年代後半から 2000 年代に実用化が進んでいる．

1997 年に発表された IBM の DAISY は，PowerPC の命令セットから VLIW 命令への変換を動的に行っており，CISC 命令セットから VLIW 命令への動的変換技術の先駆となっている．

2000 年に発表された Transmeta 社の Crusoe は，VLIW プロセッサをコアアーキテクチャとしてもつ．VLIW 命令長は 128 ビットで，Intel x86 命令セットを入力として，これを Code Morphing Software（CMS）と呼ぶランタイムのトランスレータによってネイティブの VLIW 命令コードに変換する．変換した VLIW コードは，主記憶上のトランスレーションキャッシュと呼ぶ領域に格納して，実行ごとの変換のコストを節約する．Crusoe は，ソフトウェアレイヤには x86 機械命令を提供しながら，これを CMS によって内部的に VLIW 命令コードに変換することにより，命令レベル並列性を抽出しており，これによってハードウェア機構を軽くして消費電力を削減するとともに，VLIW 命令のもつ並列性によって性能を確保している．

2001 年に発表されたインテル社の最初の IA-64 プロセッサ Itanium では，命令を三つ並べ，更にこれらの命令の組合せを表す 5 ビットのテンプレートを付加する．コードの生成時には，データの依存関係を調べ，依存関係のない命令をグループ化し，更に同時実行可能なことを確かめた上で，3 命令ずつバンドルに並べる．このような機構によって，128 ビットの VLIW 命令による制御を行う．

(3) 投機的・非投機的マルチスレッド実行による高性能化

命令レベル並列処理の限界が見えてくるにつれて，単一スレッドから複数スレッドへと処理の対象を拡張して，抽出する並列性を拡大する努力がなされている．マルチスレッド実行の方式によって，まず大きく非投機的なマルチスレッド処理と投機的なマルチスレッド処理とに分けることができる．

非投機的なマルチスレッド処理に分類される方式として，2004 年にランガン（R. Rangan）らによって提案された Decoupled Software Pipelining 法がある [39]．これは，ループの一イタレーションを複数スレッドに分割して，パイプライン実行することで時間並列性による高速化を目指す方法である．

先に述べた Simultaneous Multithreading（SMT）も非投機的なマルチスレッド実行に分類される（インテル社の Hyperthreading は，SMT タイプのマルチスレッド化である）．

投機的なマルチスレッド処理に分類される方式として，ヘルパースレッドを活用する方法と，逐次コードを投機的なマルチスレッドに分割し並列実行する方法とがある．

ヘルパースレッドは，ロードミスなどによって実行時間が掛かる主スレッドに対して，そのコードの一部を使ったコードをヘルパースレッドとして先行実行することにより，キャッシュへのプリフェッチなどを行って，主スレッドの高速化を図る方式である．2001 年に提案されたバラスラモニアン（R. Balasubramonian）らによる未来スレッド（future thread）[40] やコリンズ（J. Collins）らによる投機的先行実行（speculative precomputation）[41] が先駆的な研究である．

逐次コードを投機的なマルチスレッドに分割し，これを並列実行する方式は，並列化の制約を緩め，静的に実行結果の正しさが保証できない場合でも，これを投機的なマルチスレッドとして並列実行するものである．先にも述べた 1995 年に発表された Multiscalar や，1998 年のクラウサ（A. Klauser）らによる Selective Eager Execution，1997 年にローテンバーグ（E. Rotenberg）らにより発表された Trace Processor などがある [42]．

以上のように，マルチスレッド化のアイデアの多くは，1990 年代半ば以降に初めて提案されたものが多いが，ここでは実用化の流れも踏まえ，2000 年代の技術に位置付けて述べた．

(4) マルチコアプロセッサの発展

2000 年代前半まで，単一プロセッサにおいて単一スレッドからの命令レベル並列性を抽出し，動作周波数を上げて高速実行する方向で進んできたマイクロプロセッサは，命令レベル並列性及び消費電力の限界とともに，2000 年代半ば以降，単一チップ上にマルチコアを搭載するチップマルチプロセッサへと発展した．マルチコアプロセッサにおいては，各コアはもともとの単一コアの性能を維持しながら，更にコア間の並列性を活用することによって，高性能化を図る．

マルチコアプロセッサは，同じコアを複数並べたユニフォームなマルチコアと，異種のコアを並べたヘテロジニアスなマルチコアとがある．

2005 年にリリースされた UltraSPARC T1（コード名 Niagara）は，32 スレッドを同時に実行可能なチップマルチプロセッサである [43]．32 スレッドは，4 スレッドを一つのスレッドグループとして，計 8 グループに分けられる．各スレッドグループは，SPARC pipe と呼ぶ 6 段の命令パイプラインを共用する．スレッド選択ロジックが，命令のタイプ，キャッシュミ

ス，割込み，リソース競合をチェックし，ストールしたスレッドを実行列から外し，同一グループ内のほかのスレッドにオーバヘッドなしに切り替える．このようなアーキテクチャにより，スループットの向上を図っている．

2005 年に発表されたインテル社 Pentium D は，デュアルコアで，L1，L2 キャッシュ共に非共有型であり，動作速度は，2.66～3.6 GHz である．2006 年に，インテル社はマルチコアプロセッサ向けのマイクロアーキテクチャとして，Intel Core microarchitecture を発表した．技術的な特徴は，命令パイプラインを 14 段と少なくする，命令発行数を 4 とする，二つの連続する x86 命令を一つのマイクロ操作に合成する，128 ビット SSE 命令をスループット 1 サイクルで実行する，ランタイムに必要に応じて実行速度を調節して省電力を図る，などである．

2005 年には，AMD 社が，最初のデュアルコアプロセッサとして，デュアルコア Opteron をリリースした. 2006 年に AMD 社は, マルチコアを前提としたアーキテクチャとして, AMD K10 を発表している．K10 の特徴は，新しいビット操作，SSE 命令の追加，512 エントリ間接分岐予測器，不要部分をシャットダウンする省電力機構，メモリサブシステムの改良など多岐にわたる．また，同じ 2006 年に，4 コアのチップ Barcelona をライブデモしている．

ヘテロジニアスなプロセッサコアをもつマルチコアプロセッサも開発されている．

Cell プロセッサは，2001 年にソニー，東芝，IBM の 3 社により設計が開始されたマイクロプロセッサで, Power Processor Element（PPE）と呼ぶ全体を制御する一つの主プロセッサと，Synergistic Processor Element（SPE）と呼ぶデータ並列演算を行う 8 つのプロセッサからなる．2005 年に 65 nm プロセスによるチップの量産が開始された．2008 年には，45 nm プロセスによるチップがアナウンスされると共に，高速な倍精度浮動小数点演算が可能な Cell が PowerXCell 8i として IBM よりアナウンスされた．

(5) グラフィック処理ユニット（GPU）の発展

グラフィック処理は，画面により現実感の高い表示をするために欠かせない処理である．具体的には，図形の回転，拡大，縮小などの幾何学処理，陰面消去，輝度計算，付影などのレンダリング処理があり，実際の処理では，これらの処理間を多数の画素データが流れて処理されることになり，これをグラフィックスパイプラインと呼ぶ．また，このようにパイプライン式に多量のデータを処理する処理形態をストリーム処理と呼ぶ．

2000 年代に入って，前節にも述べたようにマルチコアが一般化するとともに，グラフィック処理を主ターゲットとしてマルチコアの GPU（Graphics Processing Unit）が発売されるようになった．GPU は，SIMD 命令によりデータ並列演算を高速化することで，汎用のプロセッサと比べて，高い演算性能を達成している．

2006 年に発表された GeForce 8800 は, このようなマルチコアによる GPU の代表例である．GeForce 8800 は，パイプライン式にデータを流しながら並列処理を行うストリーミングプロセッサと呼ぶ演算ユニットを 128 個搭載している．ストリーミングプロセッサの動作周波数は 1.5 GHz であり，8 個のストリーミングプロセッサがストリーミングマルチプロセッサを構成する．更に，一つのストリーミングマルチプロセッサに対して，16 KB のシェアードメモリと 2 個のベクトル/超越関数演算器が割り当てられている．ストリーミングプロセッサは，スレッド実行マネジャーの制御のもとにマルチスレッド実行を行うが，メモリのインター

リーブを行うことでアクセスレイテンシを隠蔽している．

GPU はこのようにグラフィック処理を主目的に開発されたものであるが，その高度な計算能力は，一般の応用においても活用され，特に数値計算の高速化に威力を発揮している．これを GPGPU（General-purpose computing on graphics processing unit）と呼ぶ．

(6) 低消費電力化

低消費電力化は，組込みプロセッサにおいては早くから重要な課題であった．汎用マイクロプロセッサにおいては，消費電力よりは性能が重視されて来たが，2000 年代に入って，チップ上のトランジスタ数の増大と動作周波数の向上が共に消費電力を増加する方向に働き，消費電力の問題がクローズアップされるようになり．2005 年頃を境に，消費電力当たりの性能が問われるようになった [44]．並列コンピュータも高性能なマイクロプロセッサを多数集積する方式が一般的となっていることから，この課題は，並列コンピュータについても同様に重要な課題となった．

省電力化のための具体的な技術と，それが適用されるコンピュータを分類すると次のようになる．

(i) 不要部分の電源シャットダウン，低速でも性能に影響のない部分を低電圧化するなどの実装技術を駆使したもの･･･モバイル系をはじめとする全体の系に適用

(ii) ILP に関して，VLIW アーキテクチャを採用し，ソフトウェアによる並列性の抽出を行うことにより，アウトオブオーダ実行のためのハードウェア機構を簡単化したもの･･･主として組み込み系

(iii) 複数コアによる並列処理を採用し，代わりに個々のコアのアーキテクチャを簡単化したもので，複数コア上の複数スレッド実行による高性能化を目指すもの･･･主としてデスクトップ系あるいはサーバ系

モバイル系，あるいは組込み系においては，低消費電力が重要な課題であるため，(i) あるいは (i) と (ii) を組み合わせて，プロセッサ単体の消費電力の低減を目指すものが多い．

これに対して，デスクトップ系・サーバ系では，性能を保ちつつ（あるいは向上しつつ），消費電力の低減が求められるため，(iii) による低消費電力化がより重要となる．

モバイル系の例をあげると，2002 年に発表された Mobile Pentium 4-M は，1.4～2.6 GHz の動作周波数をもち，スリープ電力 5 W（1.2 V），ディープスリープ電力 2.9 W（1.0V）である．

(7) 信頼性・安全性の向上

2000 年代においてコンピュータには，より高い信頼性と安全性が求められるようになった．これは，通常の個人使用環境を考えたとき，性能面ではほぼ満足できる状態にある一方，コンピュータが社会の様々な場面で活用されるようになり，またネットワークに接続して使用することが一般的になった結果，故障しない信頼性の高いコンピュータ，あるいは悪意をもったコンピュータシステムへの侵入の試みに対して安全性が保てるようなコンピュータがより強く求められるようになったことによる．高信頼化と安全性を強化したコンピュータは，ディペンダブルコンピュータと呼ばれる．

信頼性の向上については，本章 2-2-1(15) に述べたフォールトトレラントコンピュータシステム技術が基盤となる．

信頼性が主としてハードウェアの多重化などによって実現されているのに対して，安全性の実現は，ソフトウェアによる部分が大きい．例えば，スタックオーバフローに対するアタックに対する対策などがその一例である．したがって，特定のセキュリティ対応コンピュータが存在するわけではない．その意味で，ディペンダブルコンピュータも信頼性，安全性を達成するための技術の総称と考えるのが妥当である．

(8) リコンフィギャラブルコンピュータ

リコンフィギャラブルコンピュータとは，アプリケーションの処理内容に応じて，その処理に適した構成に再構成可能なコンピュータを意味する．このような方式は，専用 LSI と比べると，再構成可能性をもたせるために柔軟性は勝るが，性能的には劣る．一方，汎用の CPU でソフトウェアを使ってアプリケーションの処理を実現することから比べれば，性能的には勝るが，柔軟性には欠ける．すなわち，リコンフィギャラブルコンピュータは，性能と柔軟性の両面において，専用 LSI と汎用 CPU の間に位置づけられる．

1990 年代はじめに登場した FPGA（Field Programmable Gate Array）は，このような再構成可能なハードウェアを実現するためのデバイスである．FPGA においては LUT（Look Up Table）とフリップフロップから構成される基本論理素子をアレイ状に配置し，これらをプログラマブルスイッチで接続した構成を基本とする．LUT の内容とスイッチの接続情報を SRAM 上に格納し，これを入れ替えることによって，様々な論理回路を実現できるようになっている．

FPGA における再構成は，デバイス内のすべての SRAM の書き換えによるためミリ秒単位の時間を要する．このため，動的な再構成には不向きである．切り替え時間を短縮するには，構成要素を演算器，レジスタ，マルチプレクサといった通常の CPU の機能ユニットにする方法がとられている．このような構成にしたうえで，構成データを各要素プロセッサに転送して再構成を行う方式と，構成データは予め各要素プロセッサに分散配置しておいて，どの構成を使用するかだけを実行時に指定する方法とがある．前者の例として，PACT 社の XPP，Elixent 社の DFA などがある．後者の例として，NEC エレクトロニクス社の DRP-1，アイピーフレックス社の DAPDNA 2 などがある．

この延長上にあって，更に大きな単位であるプロセッサを多数集積し，プロセッサ間のデータの流れを動的に制御することにより，問題にあったプロセッサ接続を実現するのがタイルプロセッサと呼ばれるものである．タイルプロセッサの例としては，MIT の Raw プロセッサ [45]，テキサス大の TRIPS プロセッサ [46]などがある．

(9) ペタフロップスを越える並列コンピュータ・クラスタコンピュータの高性能化

2000 年代において，並列コンピュータは更に高並列・高性能となった．そのアーキテクチャは基本的に 1990 年代の延長上にある．すなわち，要素プロセッサとしては，各年代における最先端マイクロプロセッサあるいはベクトルプロセッサを使用し，これを高性能な通信ネットワークにより結合した形が一般的である．したがって，クラスタコンピュータとの違いは，アーキテクチャレベルよりは，むしろ実装レベルでどこまで性能にこだわった実装をするかにある．

2002 年に発表された地球シミュレータ（Earth Simulator）は，要素プロセッサとしてベク

トルプロセッサを使用し，これを単一段のクロスバスイッチで結合している．地球シミュレータは歴史的意義が大きなコンピュータとして本編 3 章 3-11 節で取り上げてあり，詳細はそちらを参照．要素プロセッサ当たり 8 GFLOPS，ノード当たり 8 要素プロセッサのものを 640 ノード搭載することにより，LINPACK で 35.86 TFLOPS を達成し，2002 年 6 月に Top 500 コンピュータの頂点に立った．

2004 年に発表された IBM の Blue Gene/L は，要素プロセッサとして，二つの 700 MHz PowerPC 440 を一つのチップに実装し，これを 3 種類のネットワークで結合している．それぞれの PowerPC は，倍精度浮動小数点演算パイプラインを 2 重にもち，ノード当たり 5.6 GFLOPS の演算性能をもつ. ネットワークは, 計算ノード間通信用の 3D トロイダル（toroidal）ネットワーク，集合通信のためのネットワーク，高速バリア同期のためのグローバル割込みネットワークの 3 種類である．2004 年 9 月に，BlueGene/L は，LINPACK ベンチマークにおいて 36.01 TFLOPS を達成し，地球シミュレータの 35.86 TFLOPS を上回ったことが IBM により発表された．

2008 年 6 月に発表された Top 500 コンピュータにおいては, IBM の Roadrunner が 1 位となった．Roadrunner は，1.0260 PFLOPS を達成して，初めて 1 PFLOPS を越えた並列コンピュータとなった．

■参考文献

1) 情報処理学会プログラミングシンポジウム委員会(編), “ダイナミック・マイクロプログラミング シンポジウム報告集,” 情報処理学会, p.208, 1973.
2) 相磯秀夫, 飯塚肇, 坂村健, “ダイナミック・アーキテクチャ,” 共立出版, p.1549. 1980.
3) J. Cocke , “The 801 Minicomputerm,” IBM J. Research and Development, vol.27, pp.237-246, 1983.
4) D. A. Patterson, C. H. Sequin, “A VLSI RISC,” Computer, Vol.15, No.9, pp.8-21, 1982.
5) J. Hennessy, N. Jouppi, F. Baskette, J. Gill, “MIPS:A VLSI Processor Architecture,” Tech. Rep. No.223, Computer Systems Lab, 1981.
6) R. O. Simpson, “IBM RT PC に見る RISC アーキテクチャ,” 日経エレクトロニクス, No415, pp.185-204, 1987.
7) J. E. Smith and A. R. Pleszkun, “Implementation of Precise Interrupts in Pipelines Processors,” Proc. 12th Int. Symp. On Computer Architecture, pp.36-44, 1985.
8) G. S. Sohi, “Instruction Issue Logic for High-Performance, Interruptibel, Multiple Functional Units, Pipelined Computers,” IEEE Trans. on Computers, Vol.39, No.3, pp.349-359, 1990.
9) Y. N. Patt, W. W. Hwu, and M. Shebanow, “HPS, A New Microarchitecture : Rational and Introduction,” In Proc. 18th Annual Workshop on Microprogramming, pp.103-108, 1985.
10) J. A. Fisher, “Trace Scheduling : A Technique for Global Microcode Compaction,” IEEE Trans. on Computers, C-30(7) : 478-490, 1981.
11) R. P. Colwell, R.P. Nix, J. J. O'Donnell, D. B. Papworth, P. K. Rodman, “A VLIW architecture for a trace scheduling compiler,” Proc. Second Conf. on Architectural Support for Programming Languages and Operating Systems, IEEE/ACM, pp.180-192, 1987.
12) M. Lam, “Software Pipelining : An Effective Scheduling Technique for VLIW Machines,” In Proc. SIGPLAN 1988 Conference on Programming Language Design and Implementation, pp.318-328, 1988.
13) 安藤秀樹, “命令レベル並列処理－プロセッサアーキテクチャとコンパイラー,” 並列処理シリーズ 3, コロナ社, pp.1-227, 2005.
14) S. McFarling, “Combining Branch Predictors,” WRL Technical Note, TN-36, Digital Equipment Corporation, 1993.
15) M. H. Lipasti, C. B. Wilkerson and J. P. Shen, “Value Locality and Load Value Prediction,” Asplos-VII, 1996.

16) A. J. Smith, “Cache Memories,” Computing Survey, Vol.14, No.3, pp.473-530, 1982.
17) ダニエル・ヒルス(著), 喜連川優(著訳), “コネクションマシン,” パーソナルメディア, p.271, 1985.
18) T. Baba, “Microprogrammable Parallel Computer -MUNAP and Its Applications-,” The MIT Press, p.290, 1987.
19) 弓場敏嗣, 山口喜教, “データ駆動型並列計算機,” コンピュータアーキテクチャシリーズ・電子情報通信学会編, オーム社, pp.1-172, 1993.
20) P. C. Treleaven, D. R. Brownbridge, R. P. Hopkins, “Data-Driven and Demand-Driven Computer Architectures,” ACM Computing Surveys, Vol.14, No.1, pp93-143, 1982.
21) D. P. Siewiorek, C. G. Bell, A. Newell, “Computer Structures: Principles and Examples,” McGraw-Hill Book Company, pp.1-926, 1982.
22) A. D. Singh, ed., “Fault-Tolerant Systems,” IEEE Computer, Vol.23, No.7, 1990.
23) J. Bartlett, “Fault Tolerance in Tandem Computer Systems,” Technical Report 90.5, 1990.
24) 藤原英二, “フォールトトレラントシステム特集,” 電子情報通信学会誌, Vol.73, No.11, 1990.
25) S. Melvin. and Y. Patt, “Exploiting Fine-Grained Parallelism through a Combination of Hardware and Software Techniques,” Proc. Annual Int'l Symposium on Computer Architecture (ISCA), pp.287-296, 1991.
26) J. E. Smith, S. Weiss, “PowerPC601 and Alpha21064:A Tale of Two RISCs,” IEEE Computer, Vol.27, No.6, pp.46-58, 1994.
27) 馬場敬信, “ALL ABOUT RISC,” Computer Design, 電波新聞社, pp19-50, 1990.
28) N. P. Jouppi and D. Wall, “Available Instruction-Level Parallelism for Superscalar and Superpipelined Machines,” ASPLOSⅢ, pp.272-282, 1989.
29) M. Johnson, “Superscalar Microprocessor Design,” Prentice Hall, p.288, 1991.
30) M. S. Lam and R. P. Wilson, “Limits on Control Flow on Parallelism,” Proc.19th Annual Int'l Symposium on Computer Architecture (May), pp.46-57, 1992.
31) G. S. Sohi, S. E. Breach, T. N. Vijaykumar, “Multiscalar Processors,” Proc. 22nd International Symposium on Computer Architecture, pp.414-425, 1995.
32) D. M. Tullsen, S. J. Eggers, H. M. Levy, “Simultaneous multithreading:Maximizing on-chip parallelism,” Proc. 22nd International Symposium on Computer Architecter, pp392-403, 1995.
33) 天野英晴, “並列コンピュータ,” 情報系教科書シリーズ第 18 巻, 昭晃堂, pp.1 -219, 1996.
34) D. Lenoski, J. Laudon, K. Gharachoroloo, W-D Weber, A. Gupta, J. Hennessy, M. Horowitz and M. S. Lam, “The Stanford Dash Multiprocessor,” IEEE COMPUTER, pp.63 -79, 1992.
35) 緑川博子, “特集「計算機クラスタ」,” 情報処理, Vol.39, No.11, pp.1071-1100, 1998.
36) 石川裕, “コモディティハードウェアを用いた並列処理技術,” 情報処理, Vol.39, No.8, pp.784-791, 1998.
37) T. E. Anderson, D.E. Culler, D. Patterson, “A case for NOW,” networks of workstations, IEEE Micro, Vol.15, No.1, pp.54-64, 1995.
38) D. Boggs, A. Baktha, J. Hawkins, J. Miller, P. Roussel, R. Singhal, B.S. Venkatraman, “The microarchitecture of the Intel Pentium 4 processor on 90nm technology,” Intel Technology Journal, vol.8, Issue1, 2004.
39) R. Rangan, N. Vachharajani, M. Vachharajani, D. I. August, “Decoupled software pipelining with the synchronization array,” Proc.13th International Conf. on Parallel Architecture and Complication Techniques, pp.177-188, 2004.
40) R. Balasubramonian, S. Dwarkadas, D. Albonesi, “Dynamically Allocating Processor Resources Between Nearby and Distant ILP,” 28th International Symposium on Computer Architecture, ISCA'01, 2001.
41) J. Collins, H. Wang, D. Tullsen, C. Hughes, Y. Lee, D. Lavery, J. Shen, “Speculative Procomputation: Long-range Prefetching of Delinquent Loads,” Proceedings of the 28h Annual International Symposium on Computer Architecture (ISCA'01), 2001.
42) E. Rotenberg, Q. Jacobson, Y. Sazeides and J. Smith, “Trace Processors,” IEEE Micro-30, 1997.
43) P. Kongetira, K. Aingaran and K. Olukotun, “Niagara: A32-Way Multithreaded Sparc Processor,” IEEE computer Society, pp.21-29, 2005.
44) S. Kaxiras, M. Martonosi, “Architectural Techniques for Low Power,” Morgan & Claypool Publishers, p.100, 2007.

45) M. B. Taylor, J. Kim, J. Miller, D. Wentzlaff, F. Ghodrat, B. Greenwald, H. Hoffman, P. Johnson, J-W. Lee, W. Lee, A. Ma, A. Saraf, M. Seneski, N. Shnidman, V. Strumpen, M. Frank, S. Amarasinghe, A. Agarwal, "The Raw Microprocessor: A Computational Fabric For Software Circuits And General-Purpose Programs," IEEE MICRO March-April 2002, 2002.

46) K. Sankaralingam, R. Nagarajan, R. McDonald, R. Desikan, S. Drolia, M.S. Govindan, P. Gratz, D. Gulati, H. Hanson, C. Kin, H. Liu, N. Ranganathan, S. Sethumadhavan, S. Sharif, P. Shivakumar, S.W. Keckler and D. Burger, "Distributed Microarchitectural Protocols in the TRIPS Prototype Processor," 39th Annual International Symposium on Microarchitecture, 2006.