リレー解説

海外の並列処理研究動向

スタンフォード大学の並列計算機研究†

天　野　英　晴††

1. はじめに

筆者は 1989 年9月から 1990 年9月まで米国 Stanford 大学 Computer Systems Laboratory に Visiting Assistant Professor として滞在した．滞在の目的は F. A. Tobagi 教授の高速パケットスイッチ用 LSI の実装プロジェクトに参加するためであった．また，滞在中 Cheriton 教授の VMP-MC（現在の Paradigm プロジェクトの前身）マルチプロセッサプロジェクトに協力する機会を得た．本稿は主として滞在中に見学の機会を得た並列計算機関連のプロジェクトに関して紹介する．本稿の内容は筆者自身の仕事の中で発表の許可を受けた範囲以外は全て公開された資料に基づいている．

2. Stanford 大学の組織と各プロジェクトの概要

Stanford 大学の並列計算機関係のプロジェクトは，組織上はほとんど全て Computer Systems Laboratory (CSL) で行われている．しかし，CSL 自体 Electrical Engineering (EE), Computer Science (CS), Applied Electoronics (AE) などいくつかの学科の計算機関連のスタッフが集まって大学院を形成している組織である．このため，各教授のオフィスのある建物や研究が行われている場所も異なり，例外を除いてプロジェクト間の交流も少ない．**表-1** に並列計算機関連のプロジェクトをまとめる．

表中で最近最も活発なプロジェクトはスケーラブルマルチプロセッサ DASH[1] である．DASH はスヌープキャッシュをもつマルチプロセッサクラスタをディレクトリキャッシュを用いて結合した構成（**図-1**）をもつ．この構成で問題になるのは，マルチプロセッサクラスタ間でキャッシュされたデータの無矛盾性を保証する方法である．クラスタ間の交信はコストが大きいため，全てのデータ書き込みに対しキャッシュされたデータの一致をとろうとすると，オーバヘッドが大きい．この問題を解決するため，DASH では，同期の解放時点のみで一致をとる Release Consistency を提案し，プロトタイプにも実装している．ほかにも，ディレクトリキャッシュの制御法，ほかのクラスタからのプリフェッチなど新しい提案が数多く採り入れられている．現在4クラスタ64プロセッサからなるプロトタイプが稼働中であり，現実的なアプリケーションが実装され興味深い結果が得られている．この DASH に関してはプロジェクトに参加している沖電気工業(株)漆原茂氏より詳細な解説論文を本学会誌にいただくことができたため，そちらの記述に譲り，他のプロジェクトの概観を紹介する．

3. 高速パケット交換用チップ

このプロジェクトと並列計算機の関係について明らかにするために，まず筆者がプロジェクトに参加するに至ったいきさつについて解説する．

筆者のプロジェクトとの関連

1988 年当時筆者らは大規模マルチプロセッサのプロセッサ-メモリ間の結合網としての Multi-stage Interconnection Network (MIN)[2),3)] に関して研究しており，従来の MIN と違ったアプローチを検討していた．

従来マルチプロセッサ用に提案されてきた MIN では，メモリに対するアクセスはパケットの形で 8～64 bit 単位でネットワークに入力される．ネットワークは Omega 網を代表とするブロ

† Projects on Parallel Processing in Stanford University by Hideharu AMANO (Department of Electric Engineering, Keio University).

†† 慶應義塾大学電気工学科

表-1 スタンフォード大学の並列計算機関連プロジェクト

プロジェクト名	概　略	プロジェクトリーダ
DASH	スケーラブルマルチプロセッサの開発	J. Hennessy, A. Gupta
TBSF チップ	高速パケット交換チップの開発	F. Tobagi
Paradigm	スケーラブルマルチプロセッサの開発	D. Cheriton
SNAP	超高速数値演算プロセッサの開発	M. Flynn. et al.

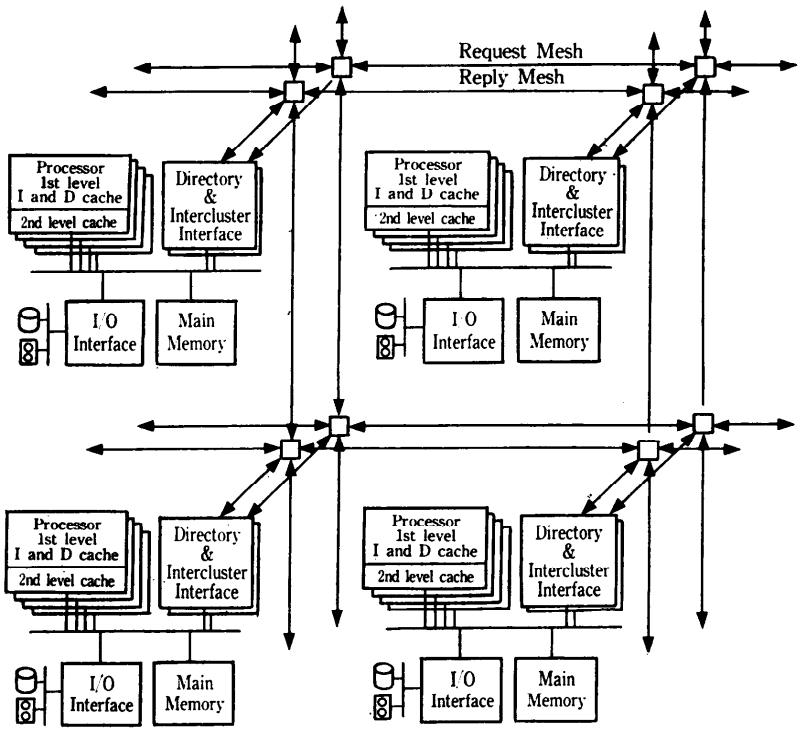


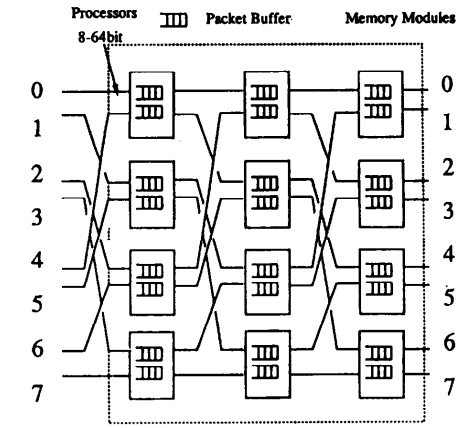
図-1 DASH プロトタイプのアーキテクチャ[1]

ッキング網であり，主に 2×2 のスイッチングエレメントを多段結合する構成である（図-2(a)）．エレメントスイッチ内で衝突が起きた場合片方のパケットはエレメントスイッチ中のバッファに格納される．この形の MIN は負荷が大きい場合の性能低下が大きく，また，プログラム中に同一メモリ番地へのアクセスが集中すると (Hot Spot)，ネットワーク全体が飽和し性能が大幅に低下する現象 (Tree Saturation) が指摘されている．従来の研究ではアクセスを結合する Combine 機能など，エレメントの機能を高めることによる解決を指向しており，その構造はますます複雑になる傾向にある．

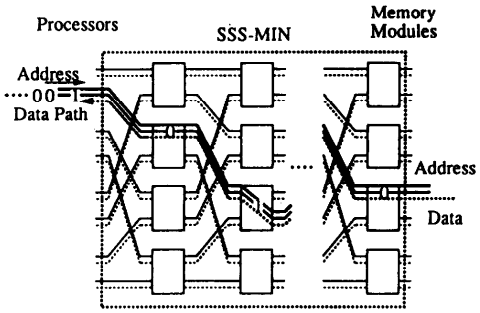
ところがこのような形式の MIN は高密度実装が困難で，特に Combine 機能をもつ MIN はいまだ実現に至っていない．まず，各スイッチングエレメントはプロセッサ当たり 8〜64 bit の入力をもつためピンネックを引き起こす．次に，スイッチングエレメントの構造が複雑である．結局，従来の MIN は 1 チップに数エレメントを実装するのが限度であり，結果として MIN は全体として巨大な基板数枚となり，クロックの調整など，実装，デバッグに大きな困難がある．

そこで筆者らは，単純なエレメントからなるノンブロッキングまたはそれに近い能力をもつネットワークに，逐次的に同期してパケットを入力する新しいクラスの MIN, SSS (Simple, Serial, and Synchronized)-MIN[4] の実現可能性を探っていた．

SSS-MIN において全パケットはプロセッサに接続された入力バッファから，フレーム信号に同



（a） 従来の MIN



（b） SSS-MIN の概念図

図-2

期して 1 bit ずつ交換網に入力される（図-2(b)）．交換網は 2 × 2 のスイッチングエレメントを多数用いたノンブロッキング網，またはそれに近い特性をもつ網が用いられる．各エレメントの構造は非常に単純であり，基本的にはパケットの 1 bit 分の記憶のみを行う．このため，網全体は交換機能をもったシフトレジスタとして働き，パケットは交換網の段数分の遅延の後，出口から 1 bit ずつ出力される．網全体の制御はパイプライン化サーキットスイッチング方式で行う．この方法はアドレス転送時に設定されたパスをエレメントに記憶させておき，そのままのパスを用いてデータを交換する動作をパイプライン的に処理する．Combine 操作は 1 bit 転送とパイプライン化サーキットスイッチング方式の特徴を利用する仮想木 Combine と呼ばれる方法で非常に容易に実装可能である[5]．

SSS-MIN の動作の基本は，従来の並列計算機用の MIN よりも，B-ISDN 用に開発が進んでいる ATM (Asynchronous Transfer Mode) パケット交換機の中の入力バッファ方式に近い．そこで筆者らは沖電気工業(株)と共同で ATM パケット交換機の検討を行いいくつか新しい提案を行ったが，この中で最も SSS-MIN として適していると考えられるネットワークにワープ網[6]があった．ところが，Stanford 大学の F. A. Tobagi 教授が 1988 年に来日の際，TBSF (Tandem Banyan Switching Fabrics)[7] と呼ばれるまったく同様のアイディアを検討中で，チップの実装プロジェクトを開始しようとしていることが明らかになった．このため相談の結果，筆者が一年間同大学を訪問し，プロジェクトに参加することになった．

Tandem Banyan Switching Fabrics

ワープ網または TBSF（以降混乱を避けるため TBSF で統一する）は，**図-3** に示すように Banyan 網 (Omega 網) を次々と接続し，各網の出口にバイパス路を設けた構造をもつ．宛先に到着したパケットはバイパス路によりそのままメモリモジュールに送られ，衝突により到着しなかったパケットのみが次の段の Banyan 網に入力される．

パケットの制御は，ヘッダ中のアドレスによる Destination Routing を基本とし，これにパケットヘッダ中の Damage bit を加えて行う．Damage bit は，あるパケットが他のパケットと衝突して，自分の希望する方向に進めなかった場合にセットされ，各 Banyan 網の入口でクリアされる．Banyan 網の出口でこの bit がチェックされ，セットされていないパケットはメモリモジュールに送られ，セットされていたパケットは次の段の Banyan 網に送られる．エレメント内で宛先の衝突が起きた場合，Damage bit がセットされているパケットはセットされていないパケットに常に道を譲る．このことにより，到着する可能性のあるパケットの妨害を避けることができる．Damage bit がセットされていないパケット同士が衝突した場合，どちらのパケットを優先してもよいが，Pri-

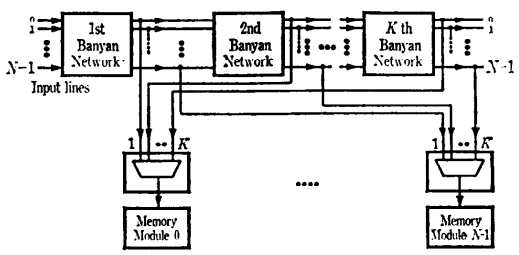


図-3 TBSF (Tandem Banyan Switohing Fabrics) のブロック図

ority bit をパケットヘッダにつけて優先順位を制御することもできる.

TBSF は ATM 方式の交換機としては，各段の Banyan 網から時間がずれて出力されたパケットを揃える部分が問題だが，SSS-MIN として用いた場合，到着順にメモリのアクセスを行えばよいので，逆にメモリ利用率をあげることができ利点に転じる.

以上のように基本的なアイディアは大変単純なので，偶然の一致があってもまったく不思議はない．ただし，Tobagi 教授は完全に ATM 方式の交換機として考えており，筆者のほうは並列計算機に用いようとした点が大きく異なるほか，転送が失敗したパケットの処理の方針など細部についての相違点は存在し，何度か議論したが意見の一致をみなかった.

チップの実装プロジェクト

Tobagi 教授はコンピュータネットワークのプロトコル，解析，B-ISDN, ATM 交換機アーキテクチャの分野で著名であり，その論文の緻密さ，研究に対する厳格さには定評があり，筆者も何度か驚嘆させられた．研究室は筆者の滞在当時，大学院の学生 9 人，訪問研究員 3 人を擁し，上記のテーマを中心に広い分野で研究が行われていた．Stanford 大学は世界各国から学生が集まるが，この研究室はとりわけ国際色豊かで，アメリカの学生のほうが少数で，フランス，イタリア，ブラジル，中国など多くの国の学生がおり，その点で実に気楽で居心地が良かった．ちなみに訪問研究員は 3 人とも日本人であった.

TBSF チップ[8] の設計は ISL (Information System Laboratory) の El Gamal 教授の提案した方式の Sea-of-Gates 用の開発環境を借りて行い，実際の製造は Signetics で行っている．人員は筆者のほかに博士課程の大学院生 F. Chiussi 氏一人だけできわめて小規模だが，設計に当たっては El Gamal 教授と ISL のスタッフ，実装については Signetics のスタッフの協力が得られた．エレメント自体の設計は Chiussi 氏が担当し，筆者はライブラリ作成，全体のフロアプランと組み上げ，テストおよび故障回避回路を担当した.

プロセスは 0.8μ の BiCMOS の Sea-of-Gates を用いている．この Sea-of-Gates はバイポーラトランジスタと巨大な面積の NMOS-FET を組み合わせる BiNMOS と呼ばれる方式[6]で，高負荷時の高速動作 (1 pF 負荷で約 800 psec) を比較的低消費電力で実現できる点に特徴がある．しかし，プロセス，セル方式共に新しいため，ライブラリセルの作成，シミュレーション，フロアプラン，電源のルーティングなどを全て自分で作成する必要があった．設計は SCS (Silicon Compiler System) の CAD, GDT を用い，主として SPARC Station 上で行った (図-4).

32 入力 5 ステージの Banyan 網が 1 チップに実装され，158 MHz のクロックで動作する (図-5)．表-2 にハードウェア仕様を示す．このチップの特徴は目的地の Tag を見てルーティングする Destination Routing を 1 bit 分の遅延 (実際は Damage bit と Priority bit を入れて 3 bit 分必要) で達成している点である．Destination Routing は各ステージごとに，行き先を決定する bit が異な

図-4 レイアウト中の Chiussi 氏

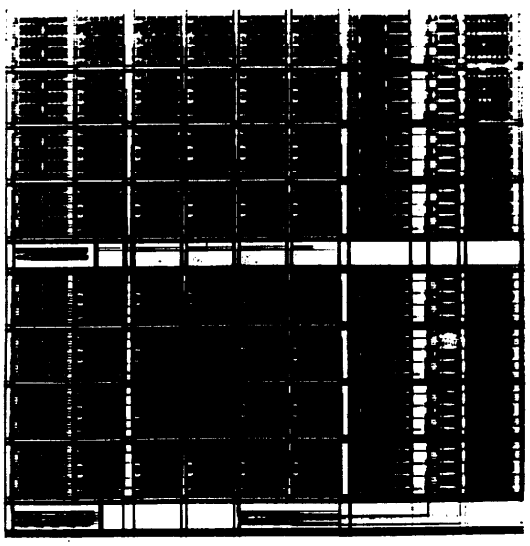


図-5 TBSF チップのレイアウト[9]

表-2 TBSF チップの仕様

テクノロジ	0.8μm BiCMOS Sea-of-Gates
面 積	5.6mm×5.8mm
セル利用率	47%
動作周波数	158.45 MHz
入出力数	32 bit

る．このため，判断を行う bit を先頭にもってくるためにヘッダをサイクリックにシフトする必要がある．この部分の処理の高速化のため，回路構成を工夫するとともに特殊な複合ライブラリセルを多数作成し，利用している．さらに，チップには自己診断機能とオンライン故障検出，回避回路が組み込まれており，パケットのルーティングエラーについては交換中に訂正することが可能である．

プロジェクトの現状

現在4入力のテストチップが稼働し，32入力のチップはレイアウトが完成し，テストチップの結果に基づく細部調整を行っている．筆者は，帰国後慶應大学に GDT システムを入手し最終チップのレイアウトの手直し，シミュレーションなどを継続して行っている．

4. Paradigm プロジェクト

プロジェクトの概観

Paradigm プロジェクト[9]は，仮想アドレスに基づくディレクトリ方式のマルチプロセッサ VMP-MC[10),11)] の継続プロジェクトで，DASH 同様スケーラブルな共有メモリ空間をもつ大規模マルチプロセッサの実現を目標としている．VMP-MC の構想を基本とし，ネットワーク，ファイル管理を含め，スケールの大きなプロジェクトになっている．DASH と比べ，ファイル管理のレベルから共有メモリへのアクセスを統合的に実現することを目標においており，アプローチが分散処理的な印象を受ける．プロジェクトを主導する Cheriton 教授は，分散 OS V System，ネットワークプロトコル，ディレクトリキャッシュなど，並列，分散，ネットワークの分野で著名である．腰に携帯電話，足にはローラスケートで学内を走り回り，精力的で会話は率直，ずばり本質を突く．筆者が滞在中は，プロジェクトは VMP-MC から Paradigm への過渡期であり，スタッフはシステムおよびアプリケーションを担当する大学院生二人，ハードウェア担当の研究員一人であった．筆者は VMP-MC プロトタイプのデバッグに協力し，何回かディスカッションの機会を得た．

Paradigm マルチプロセッサシステムの概観

Paradigm は階層バス結合マルチプロセッサからなる Node とディスクを図-6 に示すように高速パケットスイッチングネットワークを用いて接続した構造をもつ．Paradigm はハードウェア/ソフトウェアの統合システムで，分散 OS がハードウェアのサポートの下で，多数の共有仮想アドレス空間を実現する．ネットワークに関するキャッシュのコンシステンシィはファイルのコンシステンシィ機構に基づき，クライアントファイルキャッシュとファイルサーバ間のオーナシッププロトコルを用いる．

マルチプロセッサモジュール (MPM) グループ

MPM グループは図-7 に示すようにメモリバス，グループバス，ボードバスの3階層バスからなるマルチプロセッサである．キャッシュの制御[12),13)] は最近の多くのマルチプロセッサと異なり，スヌープキャッシュではなく，ディレクトリ方式である．Memory Module (MM) を例に管理の方式を解説する．

Memory Module Directory (MMD) は MM 中の各キャッシュブロック単位に下のような 16 bit のブロック状態エントリを用意する．

$$CCLP_{12}P_{11}\cdots P_2P_1P_0$$

CC は 2 bit のコードで shared, private, request-notification, undefined を表す．L はロックビットで，ロック操作とメッセージ転送で用いる．P_i はプロセッサ番号（ここでは MPM グループの番号で 0 から 12）を示す．複数のプロセッサが Read shared アクセスを出すと CC は shared の状態になり，プロセッサに対応する bit がセットされる．ここでプロセッサが Read private または Write アクセスを発生すると CC は private

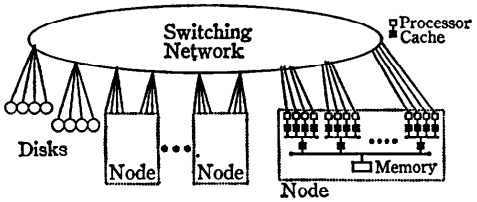


図-6 Paradigm の全体図

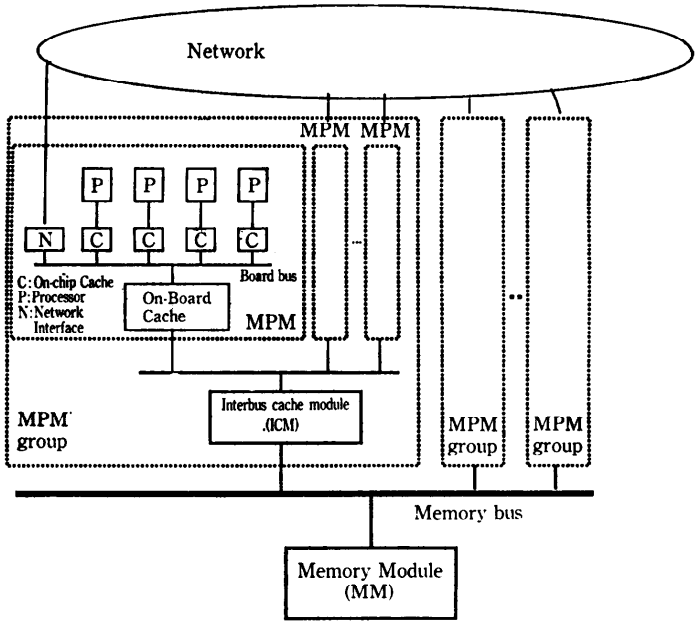


図-7 MPM Group

にセットされ，ビットがセットされているプロセッサのキャッシュが無効化される．ハードウェアは必要なイベントが発生した場合の割り込みと，キャッシュメモリ間の高速ブロック転送をサポートし，プロトコル自体はソフトウェアが制御する．

この機構は共有メモリ上でのメッセージ転送にも用いることができる．CC が request-notification 状態のとき，送信側のプロセッサが書き込みを行うと P_i がセットされているプロセッサに割り込みがかかると同時にロックビットLがセットされる．全ての受信側のプロセッサがデータを読み出すとLはクリアされる．同様の方法でメモリベースのロック操作を実現することも可能である．

Interbus Cache Module (ICM), オンボードキャッシュにもほぼ同様な機構が備えられ，統一したソフトウェアで制御を行うことができる．オンボードキャッシュは仮想アドレス空間でアクセスされ，アドレスの変換は MPM とグループバスとの間で行われる．

プロジェクトの現状

MPM グループの前身ともいえる VMP-MC のプロトタイプは，プロセッサボードとメモリボードを VME バス（2エッジハンドシェークを導入し高速化している）に接続した構成をもつ．筆者の帰国の時点で各プロセッサボードは稼働し，V Kernel が走っており，ディレクトリ構造をもつ共有メモリのデバッグ中である．このプロトタイプでの経験を生かし，Paradigm としての新しいプロトタイプを設計中とのことである．同時に Paradigm を想定したキャッシュを効果的に使うアプリケーションの研究が進められている[15]．

5. SNAP プロジェクト

Stanford Nanosecond Arithmatic Processor (SNAP) プロジェクト[16]は表-3 に示す超高速の算術演算プロセッサを構築する目的をもつ．この目標を達成するためには，アルゴリズム，データ表現，CAD, 回路，デバイス，パッケージングの広い分野での研究が必要である．プロジェクトを主導するのは，アーキテクチャの大家で温厚な性格で知られる M. J. Flynn 教授で，CAD, デバイス，回路，集積実装技術の専門家スタッフがそれぞれの分野でプロジェクトを支えている．筆者の滞在中京都大学の高木直史先生が演算アルゴリズムの分野でこのプロジェクトに参加されていた．

SNAP は図-8 に示すようにアクティブサブストレート上に，浮動小数点加算，乗算，除算ユニ

表-3 SNAP プロセッサの速度目標値

整数加算	0.5-1 ns
浮動小数点加算	1.5-3 ns
浮動小数点乗算	3-5 ns
除算，平方根	10-15 ns

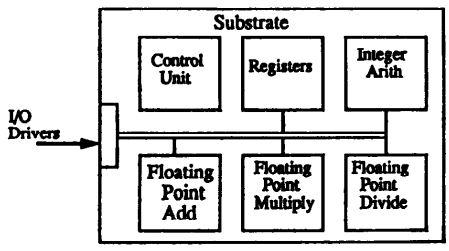


図-8 SNAP プロセッサ

ットと整数演算ユニットを置く．現在放熱法，高速 CMOS 加算器などで興味深い結果が報告されている[17]．また，各段ごとにレジスタを用いず，複数のデータ流が波状にストレージ間を伝搬する Wave Pipelining[18] という手法が検討されている．伝搬遅延時間の調整などが困難であるが，高機能な CAD の利用による実現を目指している[19]．現在それぞれの研究フィールドにおいてプロトタイプチップを作成している．

6. おわりに

以上 Stanford 大学の CSL における並列計算機関連の三つのプロジェクトを紹介した．筆者が関連したプロジェクト中心になった点をご容赦されたい．

筆者にとって，並列計算機のハードウェアを作るという点に関しては，たとえば筆者の所属する慶應大学と比較して，各プロジェクトの研究レベル，実装技術，設備が特に優れているとは感じられなかった．しかし，並列計算機アーキテクチャを研究するうえの底力の違いは感じられた．

まず，ソフトウェアである．並列計算機の研究は，現実的な並列アプリケーションに基づくトレースデータや，シミュレーション環境が不可欠であるが，Stanford 大学ではこれらのソフトウェア資産が豊富である．この環境により開発時に，現実的なデータに基づいて，アーキテクチャ上のトレードオフを検討できる．次に，チップの開発技術である．Stanford 大学では，研究室レベルで使えるチップ開発用の実用的 CAD が多く，設計したチップを実装する機会も多い．全米的な組織である MOSIS を利用する，Silicon Valley の半導体開発能力をもつ企業の協力を得る，学内の研究所で作る，という三つの選択肢がある．SNAP など，チップの実装技術自体を対象とするプロジェクトは学内の研究施設を用いるが，マスクパターンまで大学で作り，製造は企業の協力による場合が多い．学生も授業で設計の経験をもつため技術力も高い．この二点が日本の大学では欠けている場合が多い．最後に，とにかく現実的なアプリケーションに基づき，定量的な評価をとろうと努力する姿勢が印象的であった．

謝辞 スタンフォード大学に滞在中プロジェクトに参加させていただき貴重な経験をさせていただいた F. A. Tobagi 教授，D. A. Cheriton 教授に感謝する．また滞在中大変親切にしてくれた学生の皆さま，訪問研究員の皆さまに心から感謝する．

参考文献

1) Lenoski, D., Laudon, J., Charachorloo, K., Weber, W., Gupta, A. and Hennessy, J.: Overview and Status of the Stanford DASH Multiprocessor, Proc. Inter. Symp. on Shared Memory Multiprocessing (Apr. 1991).
2) 奥川峻史：並列計算機アーキテクチャ，コロナ社 (1991).
3) 富田眞治：並列計算機構成論，昭晃堂 (1986).
4) 天野英晴，Kalidou, G.: SSS (Simple Serial Synchronized) スイッチングアーキテクチャに基づく並列計算機，信学技報 CPY 91, No. 7 (1991).
5) Amano, H. and Kalidou, G.: A Batcher Double Omega Network with Combining, Proc. ICPP 91 (Aug. 1991).
6) 坂元，荒川，正木，井上，天野：自己ルーチングスイッチの構成とその評価，信学技報 SSE 88 No. 31 (1988).
7) Tobagi, F. A. and Kwok, T.: The Tandem Banyan Switching Fabric: A Simple High-Performance Fast Packet Switch, Proc. INFOCOM 91, Miami, Florida (Apr. 1991).
8) Chiussi, F., Amano, H. and Tobagi, F. A.: A 0.8-μm BiCMOS Sea-of-gates Implementation of the Tandem Banyan Fast Packet Switch, Proc. of IEEE Custom Integrated Circuits Conference (1991).
9) Gamal, A. E., Kouloheris, J. L., How, D. and Morf, M.: BiNMOS: A Basic Cell for BiCMOS Sea-of-Gates, Proc. of IEEE Custom Integrated Circuits Conference (1989).
10) Cheriton, D. R., Goosen, J. A. and Boyle, P. D.: Paradigm: A Highly Scalable Shared-Memory Multiprocessor Architecture, IEEE Computer (Feb. 1991).
11) Cheriton, D. R., Slavenburg, G. and Boyle, P. D.: Software-Controlled Caches in the VMP Multiprocessor, Proc. 13th Int'l Symp. Computer Architecture (June 1986).
12) 高橋義造編：並列処理機構，丸善 (1989).
13) 浦城恒雄：キャッシュメモリの一致性について，情報処理，Vol. 32, No. 1 (Jan. 1991).
14) Cheriton, D. R. et al.: The VMP Multiprocessor: Initial Experience, Refinements, and Performance Evaluation, Proc. 15th Int'l Symp. Computer Architecture (June 1988).
15) Cheriton, D. R., Goosen, H. A. and Machanick, P.: Restructuring a Parallel Simulation to Improve Cache Behavior in a Shared-Memory Multiprocessor: A First Experience, Proc. of Inte. Symp. on Shared Memory Multiprocessing (Apr. 1991).

16) Flynn, M., DeMicheli, G., Dutton, R., Wooley, B. and Pease, F.: SUB-NANOSECOND Arithmetic, Stanford University Technical Report CSL-TR-90-428 (May 1990).
17) Quach, N. T. and Flynn, M. J.: High-Speed Addition in CMOS, Stanford University Technical Report CSL-TR-90-415 (Feb. 1990).
18) Wong, D., De Micheli, G. and Flynn, M.: Designing High-Performance Digital Circuits Using Wave Pipelining, VLSI '89 (Aug. 1989).
19) Wong, D., De Micheli, G. and Flynn, M.: Inserting Active Delay Elements to Achieve Wave Pipelining, ICCAD '89 (Nov. 1989).

(平成3年7月2日受付)

天野 英晴（正会員）

1958年生．1986年慶應義塾大学理工学部大学院博士課程修了．工学博士．並列計算機の研究に従事．現在慶應義塾大学理工学部専任講師．著書「誰にもわかるディジタル回路」（オーム社），「並列処理機構第5章」（丸善）．