計算機アーキテクチャ　95－8
(1992. 8. 19)

ネットワークを用いた高速スヌープ機構

村田 浩樹　　清水 茂則
日本アイ・ビー・エム株式会社 東京基礎研究所

スヌープ・キャッシュ機構のデータ転送バンド幅を増大させる方式について述べる．データ転送バンド幅を増大するために，キャッシュ・ライン長を長くすると共に，データ転送媒体として従来用いられていたバスの代わりにマルチプル・データ・パスを用いてデータ転送を多重化する．アドレス，コマンド，スヌープ・レスポンスは従来どうりバスを用いて転送し，それらの転送とデータ転送をパイプライン化，もしくはスプリット化することで，キャッシュ・コヒーレンスを維持する．速度の向上について簡単な評価を行なった．

HIGHSPEED SNOOP MECHANISM USING NETWORK

Hiroki Murata　　Shigenori Shimizu
Tokyo Research Laboratory, IBM Japan, Ltd.

5-19, Sanbancho, Chiyoda-ku, Tokyo 102 Japan

The method to increase the data transfer bandwidth of the snoop cache mechanism is discussed. The method consists of lengthening the cache line length and utilizing a multiple data path instead of the data bus. Those multiplex the data transfers and increase the data transfer bandwidth. The cache coherence is kept by pipelining or splitting the data transfer with the transfer of address, command, and snoop response which are transfered with the buses as before. The performance improvement is evaluated simply.

1．はじめに

スヌープ・キャッシュ機構のデータ転送バンド幅を増大させる方式について述べる．

共有バス型のマルチプロセッサ・システムでは，共有メモリへのアクセス競合がシステム性能の向上を妨げる最大のボトル・ネックとなるが，これを緩和する目的で，各プロセッサにプライベート・キャッシュを付加することにより，共有バスに対する要求バンド幅を低減させるという方法がよく用いられる．新たに導入された複数キャッシュ間のコヒーレンス維持の方式としては，スヌープ・キャッシュ方式がよく知られる．この方式では，それぞれのキャッシュが常に共有バス上に発生するメモリ・アクセスを監視し，必要に応じて該当キャッシュ・ブロックに対する適当な操作を実行することによって複数キャッシュ間のコヒーレンス維持がハードウェアによって実現される．この方式は，簡便に，しかも高速にキャッシュ・コヒーレンス制御が実現される点に於て優れた方式であり，現在広く用いられている．しかし，反面，共有バス結合に基礎を置いたスヌープ・キャッシュ方式では，バス・ネックという本質的問題点は解決されず，十数台規模の小規模並列システムまでしか実用に供しないという欠点を有する．

一方，バス・ネックの問題を本質的に解決する方式としてインターコネクション・ネットワークの研究が古くから行われている．インターコネクション・ネットワークで結合されたマルチプロセッサ・システムでは，システムを構成するプロセッサの数の増加にしたがって結合リンクの数も増加するので，プロセッサ台数に比例した転送バンド幅が確保され，数百台規模の大規模並列システムの実現が可能となる．しかし共有メモリ・モデルの実現には困難が伴う．スヌープ・キャッシュ方式のように各プロセッサに付加されたプライベート・キャッシュが，他のプロセッサの行うメモリ・アクセスのすべてを監視するためには，メモリ・アクセスを行なったプロセッサがその旨ブロードキャストする必要がある．これはネットワークの負荷を増大させてシステム性能を低下させるため，より効率のよい実現方法が現在活発に研究されている[1]．

本稿ではデータ転送媒体としてマルチプル・データ・パスを用いたスヌープ・キャッシュ機構について述べるが，これに関連したものとしてスヌープ・バスとインターコネクション・ネットワークを組み合わせる方式[2]がある．この方式では，インターコネクション・ネットワークの他にスヌープ・バスが付け加えられる．コヒーレンシ制御のためにキャッシュ間のコミュニケーションを必要とするメモリ・アクセスはスヌープ・バスを介して処理され，キャッシュ間のコミュニケーションを必要としない通常のメモリ・アクセスはインターコネクション・ネットワークを介して処理される．キャッシュ間のコミュニケーションが必要であるのかどうかを判断するために，各キャッシュにはシステム中のすべての共有コピーの状態を記録したテーブルが付加される．この方式では，転送バンド幅の上限は共有データのアクセスに使用される共有バスあるいは，個有データのアクセスに使用されるインターコネクション・ネットワークのどちらか先に飽和した方で決定される．従って，この方式の転送バンド幅の上限は，実行されるプログラムの性質に大きく依存することになるが，常識的に考えて，キャッシュ・ミス率がかなり低くなるようにうまく設計されたスヌープ・キャッシュ方式のマルチプロセッサ・システムでは，システム・バス上に発生するアクセス要求全体のうちの数分の一がコヒーレンシ制御のためのキャッシュ間コミュニケーションによって発生すると思われるので，この方式では，純粋な共有バス結合方式に比較して高々数倍程度の転送バンド幅が実現されるだけである．また，この方式では，共有バスを用いたアクセスが必要であるのか，インターコネクション・ネットワークを用いたアクセスのみで十分なのかを各キャッシュがローカルに決定することを可能とするために，各キャッシュにシステム全体の状態を記述した管理テーブルが必要となる．さらに，制御機構としては，このテーブルを用いた上

で共有バスおよびインターコネクション・ネットワークを制御する機構が必要であり，制御機構が複雑になる．

本稿では，スヌープ・キャッシュ方式程度の簡素な制御機構でより広いデータ転送バンド幅を実現するために，キャッシュ・ライン長を長くすると共に，データ転送媒体として従来用いられていたバスの代わりにマルチプル・データ・パスを用いてデータ転送を多重化し，データ転送バンド幅を増大した高速スヌープ・キャッシュ機構について述べる．2章では高速スヌープ・キャッシュ機構を構成するために利用するスヌープ・キャッシュ方式の特性について，3章では高速スヌープ・キャッシュ機構の構成と基本的な動作，4章ではキャッシュ・コヒーレンス・プロトコルへの対応，5章ではスプリット・バス方式への対応，6章では簡単な性能評価について述べる．

2．スヌープ・キャッシュ機構の特性

図1はスヌープ・キャッシュ方式のマルチプロセッサを示す．図1において複数のプロセッサはそれぞれのプライベート・キャッシュを介して共有バスおよび共有メモリに接続されている．各プライベート・キャッシュは共有バス上に発生するメモリ・アクセスを監視し，必要に応じて該当キャッシュ・ブロックに対する適当な操作を実行することによって複数キャッシュ間のデータの一致性を実現する．即ち，スヌープ・キャッシュ方式に必要なのは，共有バス上のアドレス／コマンドをすべてのキャッシュが監視することと，そのスヌープとそれに対応するデータ転送がキャッシュ間のデータの一致性を維持するために必要十分なだけ関係付けられること（バスではスヌープとデータ転送が一組で行われるため暗黙のうちに実現されている．）であり，データ・バス自体の監視は必ずしも必要ではない．一方，最近の高速プロセッサでは64バイト以上にも及ぶようなかなり長いキャッシュ・ライン・サイズを用いることも希ではないが，その場合，そのような長いキャッシュ・ラインはビット幅に限度のあるシステム・バス上を複数のバス・サイクルを使用して（例えば，8バイトx8サイクル），ブロック転送される．即ち，バス・スヌープに必要なアドレス／コマンド・サイクルは，1バス・サイクルないし2バス・サイクルという極めて短い期間であるにもかかわらず，長いキャッシュ・ラインを転送するかなり

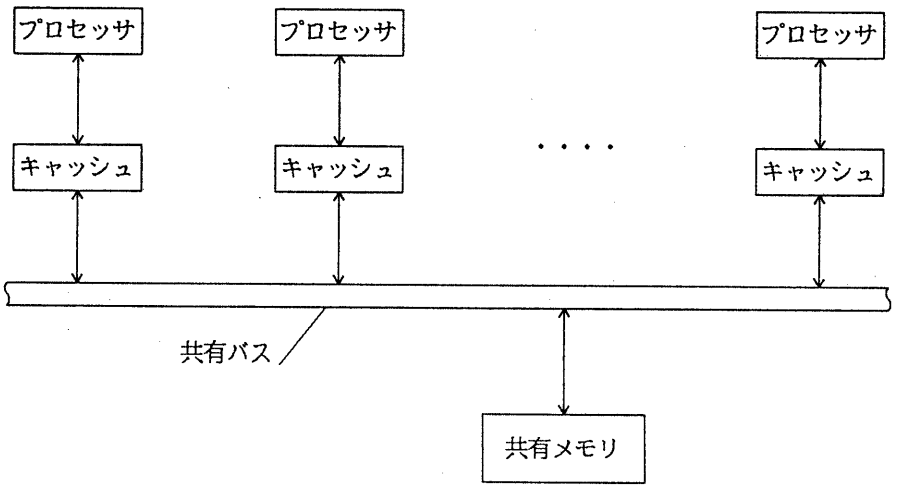

図1：スヌープ・キャッシュ方式のマルチプロセッサ

長い期間，システム・バスが占有される．本稿で述べる高速スヌープ・キャッシュ機構では，上記の２つの事実を積極的に利用して，バス・スヌープが必要なアドレス／コマンド・バスには共有バス結合を用い，バス・スヌープが不要なデータ・バスにはマルチプル・データ・パスを用いる．本方式によって，論理的には，スヌープ・キャッシュ方式をそっくり踏襲したままで，転送バンド幅増大の目的で，インターコネクション・ネットワークを使用することが可能となる．

３．高速スヌープ・キャッシュ機構の構成と基本的な動作

高速スヌープ・キャッシュ機構の構成を図２に示す．

図２において各プロセッサP_1…P_nに付加されているスヌープ・キャッシュSC_1…SC_nは単一のアドレス／コマンド・バスによって，互いに結合される．このアドレス／コマンド・バスによって，キャッシュ・コヒーレンス制御のためのスヌープ動作が維持される．一方，各スヌープ・キャッシュSC_1…SC_nからのデータ・パスDP_{C1}…DP_{Cn}は，データ・パス・スイッチとデータ・パスDP_{M1}…DP_{Mm}を経由して，複数のメモリ・モジュールM_1…M_mにインターリーブされた共有メモリ・システムに結合される．データ・パス・スイッチはアドレス／コマンド・バスを監視して、データ転送に必要なデータ・パスをデータ・パスDP_{C1}…DP_{Cn}とデータ・パスDP_{M1}…DP_{Mm}の間に確立する．キャッシューキャッシュ間のデータ転送が必要な場合，基本的にはデータをキャッシュから対応するメモリ・モジュールに書き込んで，それを新たに読み込むことで行うが，データ・パス・スイッチが、あるキャッシュからメモリ・モジュールへ書き込む

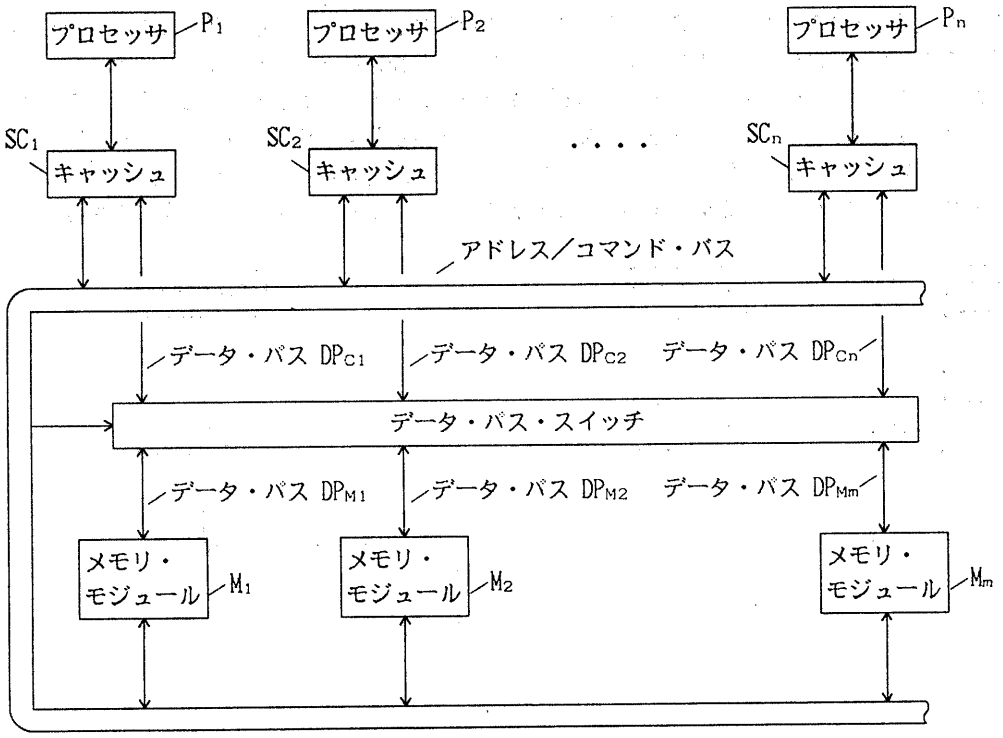

図２：高速スヌープ・キャッシュ機構の構成

データ・パスと，そのメモリ・モジュールから別のキャッシュへ読み込むデータ・パスを同時に確立できるタイプのものであれば，そうすることで，キャッシューキャッシュ間のデータ転送速度を２倍にすることができる．

図３は，このような構成を用いて，メモリ・アクセスおよびバス・スヌープがどのように多重化されるかを示したタイミング・チャートである．プロセッサ，メモリ・モジュールがそれぞれ８台，キャッシュ・ライン長がデータ・パス幅の８倍であると仮定している．またメモリ・モジュールのアクセス時間がすべて一定でパイプライン的にアクセスできるものとしている．横軸方向はバス・サイクルを示しており，１から１０までのバス・サイクルが例として示されている．この例では，バス・サイクル１に於て，あるメモリ・アドレスに対してアクセスが発生し，すべてのキャッシュでのスヌープ動作自体はバス・サイクル１のみで終了したが，バス・サイクル２から９までの８バス・サイクルを使用して長いキャッシュ・ラインがブロック転送されている様子を示している．次のバス・サイクル２で異なるメモリ・モジュールに対してアクセスが発生したならば，このための処理はすぐに開始される．スヌープ動作はバス・サイクル２のみで終了し，バス・サイクル３から１０までの８サイクルを使用してキャッシュ・ラインがメモリ・システムから要求元キャッシュに転送される．以下，そのような状況の繰り返しを示している．実際の動作状況では，アドレス／コマンド・バス上でのコンテンションとメモリ・モジュール上でのコンテンションの両方で実効バンド幅が決定されるが，図３に示されたような理想的な動作状況では，実現可能なスヌープ・サイクルとキャッシュ・ライン・サイズでその理論的最大値が決定される．例として，スヌープ・サイクルを40ns(25MHz)と仮定し，データ・バス幅を8バイト，キャッシュ・ライン・サイズを64バイトとすると，実現可能なバス・バンド幅の上限は25MHz×64バイト＝1.6Gバイト／秒である．ちなみに従来のスヌープ・キャッシュ方式を用いた場合のタイミング・チャートは図４のようになり同じ条件のもとでの実現可能なバス・バンド幅の上限は25MHz×8バイト＝200Mバイト／秒となる．

バス・サイクル	1	2	3	4	5	6	7	8	9	10
バス・スヌープ	Snp 1	Snp 2	Snp 3	Snp 4	Snp 5	Snp 6	Snp 7	Snp 8	Snp 9	Snp10
データ転送1		D1	D2	D3	D4	D5	D6	D7	D8	
データ転送2			D1	D2	D3	D4	D5	D6	D7	D8
データ転送3				D1	D2	D3	D4	D5	D6	D7
データ転送4					D1	D2	D3	D4	D5	D6
データ転送5						D1	D2	D3	D4	D5
データ転送6							D1	D2	D3	D4
データ転送7								D1	D2	D3
データ転送8									D1	D2
データ転送9										D1

図３：高速スヌープ・キャッシュ機構のタイミング・チャート

バス・サイクル	1	2	3	4	5	6	7	8	9	10
バス・スヌープ	Snp 1								Snp 2	
データ転送1		D1	D2	D3	D4	D5	D6	D7	D8	
データ転送2										D1

図4：スヌープ・キャッシュ機構のタイミング・チャート

4．キャッシュ・コヒーレンス・プロトコルへの対応

キャッシュ・コヒーレンス・プロトコルは、データ転送媒体に必要な機能によっておおまかにライト・スルー，（ライト・バック・）インバリデート，（ライト・バック・）アップデートの3方式に分類できる．この中でアップデート方式は複数のキャッシュへのデータ転送を必要とし，図2で示した高速スヌープ・キャッシュ機構では実現が難しいため，ライト・スルー，インバリデート方式の場合について考察する．

4.1. ライト・スルー方式への対応

ライト・スルー方式の場合，共有バス上に現われるアクセスはミス・リードとライトであり，ダーティ・リプライは存在しない．このため図3に示したパイプラインが崩れるのは，あるアクセスのデータ転送が終了しないうちに同じメモリ・モジュールへのアクセスが発生した場合のみであり，その発生を抑えるようにアービトレーションすれば図3のようなタイミングで動作する．またアービトレーションで抑えずにアクセス中のキャッシュもしくはメモリ・モジュールがそのアクセスをアボートさせるという方法もある．

4.2. インバリデート方式への対応

インバリデート方式の場合，共有バス上に現われるアクセスはミス・リード，リプレイス，ダーティ・リプライ，及びライト・ミスの扱いによって異なるがミス・ライトもしくはリード・フォー・オーナーシップである．図3に示したパイプラインが崩れるのはライト・スルー方式同様，あるアクセスのデータ転送が終了しないうちに同じメモリ・モジュールへのアクセスが発生した場合と，ミス・リード，ミス・ライト，リード・フォー・オーナーシップに対してダーティ・リプライが行われた場合である．ダーティ・リプライ時の制御は，簡単には，ダーティをリプライするキャッシュ，メモリ・モジュールとダーティを受け取るキャッシュの間にデータ・パスを確立し，受け取り側を待たせておいて，ダーティを送れる状態になったら待ち状態を解消すればよい．ただしダーティをリプライするキャッシュがダーティを集中して持っていた場合には他のすべてのキャッシュに対してダーティ・リプライする可能性もあり，リプライまでに最悪で，キャッシュ・ライン長／データ・パス幅×（プロセッサ数－1）サイクル，待つ事になる．この場合にはスヌープとダーティ・リプライ及びデータ転送をスプリットすることが，性能の向上に有効と思われる．

5．スプリット・バス方式への対応

スヌープ・キャッシュ機構でスプリット・バス方式を用いる場合の問題は，同じアドレス（ライン）へのアクセスがスプリットされ多重化されるとキャッシュ・コヒーレンスに矛盾が生じることである[3]．キャッシュ・コヒーレンスを維持するためにはスプリットされているアクセスをアービタもしく

は各キャッシュが記憶しておき，同じアドレスへのアクセスを抑えるようにアービトレーションするか，スプリットされたアクセスに関係するメモリ・モジュールもしくはキャッシュが新しいアクセスをアボートさせるという方法がある．

スプリット・バス方式の利点を有効に活かすには，メモリ・モジュールのアクセス時間がばらつくことが望ましい．これにはメモリ・モジュール内でのインターリーブの深化が有効と思われる．また，スプリット・バス方式を用いることでI/Oバスの接続も容易になると思われる．

6．性能についての考察

3章で高速スヌープ・キャッシュ機構のデータ転送バンド幅の上限を求めたので，ここではライト・スルー方式を用いた場合の高速スヌープ・キャッシュ機構，従来のスヌープ・キャッシュ機構，及びキャッシュ自体のデータ転送バンド幅について解析的に求めてみる．

まず高速スヌープ・キャッシュ機構について求める．プロセッサがN台，メモリ・モジュールがM台，キャッシュ・ライン長がLバイト，データ・バス幅がBバイトの場合を考える．プロセッサが毎サイクル命令もしくはデータを要求するとして，キャッシュのミス率をpとすると，あるバス・サイクルでメモリ要求がでる確率は

$$q=Np$$

ここでデータ・パスiが現在のバス・サイクルで使用されている確率をF_{i0}，前のバス・サイクルで使用されている確率をF_{i-1}，さらにその前のバス・サイクルで使用されている確率をF_{i-2}…とすると

$$F_{i0}=(1-F_{i-1})q/M+(1-F_{i-2})q/M+\cdots+(1-F_{i-L/B})q/M$$

データ・パスiが使用されている確率が一定だとすると

$$F_i=F_{i0}=F_{i-1}=\cdots=F_{i-L/B}=Lq/(BM+Lq)$$

データ・パスがj本使用されている確率は

$$G_j=\binom{\min(M,L/B)}{j}(F_i)^j(1-F_i)^{M-j}$$

あるバス・サイクルでのデータ・パス利用期待値は

$$U=\sum_{j=1}^{\min(M,L/B)} jG_j$$

データ転送バンド幅はバス・サイクル（Ｈｚ）をＣとすると

$$BCU$$

従来のスヌープ・バスについて求めると，共有バスが現在のバス・サイクルで使用されている確率をF_0，さらに…と，上と同様に定めると，

$$F_0=(1-F_{-1})q+(1-F_{-2})q+\cdots+(1-F_{-L/B})q$$

共有バスが使用されている確率は

$$F=F_0=F_{-1}=\cdots=F_{-L/B}=Lq/(B+Lq)$$

データ転送バンド幅は

$$BCF$$

キャッシュ自体のデータ転送バンド幅は，１回のデータ転送が幅Bのデータ・パスをL/Bサイクル占有して行われることから，

BL/BCq

3章と同様にプロセッサ，メモリ・モジュールそれぞれ8台，バス・サイクル25MHz，データ・バス幅を8バイト，キャッシュ・ライン・サイズを64バイトとすると，それぞれのデータ転送バンド幅は以下のようになり，高速スヌープ・キャッシュが高いデータ転送バンド幅の提供に成功していることがわかる．

表：ミス率と各キャッシュのデータ転送バンド幅

ミス率	高速スヌープ・キャッシュ	スヌープ・キャッシュ	キャッシュ自体
0.05	457MB/s	152MB/s	640MB/s
0.10	711MB/s	173MB/s	1.28GB/s
0.20	985MB/s	186MB/s	1.86GB/s

7．おわりに

スヌープ・キャッシュ機構のデータ転送媒体として，マルチプル・データ・パスを用いてデータ転送バンド幅を増大する方法についてその基本的な構成と動作について述べた．

キャッシュ・コヒーレンスの維持についてはプロトコルとしてライト・スルー方式とライト・バック・インバリデート方式を用いた場合について考察し，メモリ・モジュールのアクセス時間が一定でパイプライン的にアクセスできる場合について，その実現方法について述べた．従来のスヌープ・キャッシュと比較して，その制御がそれほど複雑でないことを示すことができたと思う．

またスプリット・バス方式を導入する際に必要な機能について述べた．スプリット・バス方式では（実はパイプライン式でも同じなのだが），同じアドレスへの同時アクセスを避けなくてはならない．その方法としてアービトレーションへの参加抑制と，アービトレーション後のアクセスのアボートという2つの方法を述べた．バスの使用効率の点からはアービトレーションへの参加を抑制するものがよいが，制御機構が複雑になる．

性能については理論的な上限とライト・スルー方式を適用した場合について，本方式と通常のスヌープ・キャッシュ方式について解析的に求めた．

今後はシミュレーションや命令レベル・シミュレーションによる評価を行ないたいと考えている．

参考文献

[1]Nitzberg, B. and Lo, V., "Distributed Shared Memory: A Survey of Issues and Algorithms," Computer, Vol.24, No.8, pp.52-60, August 1991.

[2]Bhuyan, L.N., Bao Chyn Liu, and Ahmed, I., "Analysis of MIN based multiprocessors with private cache memories", Proceedings of the 1989 International Conference on Parallel Processing, pp.51-58, August 1989.

[3]福村　好美，平野　正則，塩澤　恒道，"バス結合マルチプロセッサのキャッシュ構成方式，" 電子情報通信学会論文誌，D-I，Vol.J74-D-I，No.10，pp.721-728，1991年10月．