

一時的にディレクトリを保持する MINDIC スイッチの設計と評価

住吉 正人[†] 緑川 隆[†] 茂野 真義[†]
田辺 靖貴[†] 薬袋 俊也[†] 天野 英晴[†]

MIN (Multistage Interconnection Network) を用いたマルチプロセッサにおける効率の良いキャッシュ制御手法 MINDIC を提案している。MINDIC は MIN を構成する各スイッチ内部に小容量のディレクトリを保持することにより低レイテンシなキャッシュ制御が可能であり、ハードウェアコスト面でも有利である。MINDIC の複数のキャッシュ制御プロトコルを検討し性能評価を行なったところ、Eviction プロトコルが無効化パケットの発生数を抑制するのに有効であるとわかった。また、MINDIC のスイッチを設計、論理合成し、54000 ゲート程度の回路規模で実現可能であることを示した。

Design and Evaluation of MINDIC: Multistage Interconnection Network which holds temporary directory in the switching elements

MASATO SUMIYOSHI,[†] TAKASHI MIDORIKAWA,[†] MASAYOSHI SHIGENO,[†]
YASUKI TANABE,[†] TOSHIYA MINAI[†] and HIDEHARU AMANO[†]

Researches on Multistage Interconnection Network (MIN) have been exerted for building a cache coherent multiprocessor, providing cache or cache directory in their switching elements. We proposed MINDIC (MIN with Directory Cache switch), a novel MIN structure that consists of switches with small temporary directory. Building temporary directory dynamically in each switching element, we can maintain cache consistency with low latency and low memory cost. Three variations of MINDIC protocol are evaluated by trace level simulation and we found that Eviction protocol can reduce redundant invalidation packets effectively. Synthesis report using 0.18 μ m CMOS process shows that the hardware cost is small enough for implementation

1. はじめに

中規模並列計算機向けの接続網として、転送能力の高いクロスバススイッチを多段に結合した MIN (Multistage Interconnection Network) が検討されている。MIN では各プロセッシングエレメント (PE) がお互いの共有メモリアクセスを監視することができないために、バス結合型の接続網で用いられているスヌープ方式によるキャッシュ制御が行なえない。このため、MIN におけるキャッシュ制御はメモリモジュール (MM) 側にディレクトリを設ける方法が一般的である。しかし、アドレス空間のサイズに応じてディレクトリが増えてしまうため、ハードウェアコストの点で問題があり、また、ネットワーク越しにディレクトリ情報が参照されるために、キャッシュ制御の際のレイテンシが増加する。このため、キャッシュの一部を MIN のスイッチングエレメント中に設ける方法¹⁾²⁾、ディレクトリを設ける方法³⁾⁴⁾ など様々な方法が提案されている。しかし、これらの方法はいずれもメモリ利用量やアクセス時間の増大、スイッチの複雑化を招き、実機で利用されるに至ってはいない。

我々は、MM に縮約したディレクトリを持たせる MINC⁵⁾ (MIN with Cache coherent mechanism) を提案、実装し、プ

ロタイプマルチプロセッサ SNAIL-2⁶⁾ による性能評価を行なった。その結果、キャッシュを利用したことによる性能向上は確認できたが、ディレクトリを縮約したことにより無駄な無効化パケットが発生し、ネットワークの混雑を招き、アプリケーションによっては十分な性能向上が得られないことが明らかになった。

これは、MINC が用いた縮約階層ディレクトリ方式⁷⁾ が、本来ページ単位でのキャッシュ管理を想定しており、多くのプロセッサがキャッシュを共有し、かつ長時間保持される場合に有効な方式であることに起因している。中規模並列計算機のキャッシュ制御は、共有する PE 数がさほど多くなく、かつ無効化までの時間が短い場合が多いため、より短期的な共有関係に適したキャッシュ制御方式が望ましい。

そこで本研究室では、MIN を構成する各スイッチングエレメント内に共有情報を一時的に保持する小容量のディレクトリキャッシュ (DC) を設け、キャッシュ一貫性を保持する MINDIC⁸⁾ (MIN with Directory Cache switch) を提案した。MINDIC は MM にディレクトリを持たず、DC は短期的な記憶であるため、DC の容量不足などによりコンシステンシの維持ができなくなる可能性がある。これを防ぐためのプロトコルを検討し、それらをシミュレーションにより性能評価する。また、MINDIC のハードウェアコストの見積もりを行なうため、MINDIC を構成するス

[†] 慶應義塾大学 理工学部
Faculty of Science and Technology, Keio University

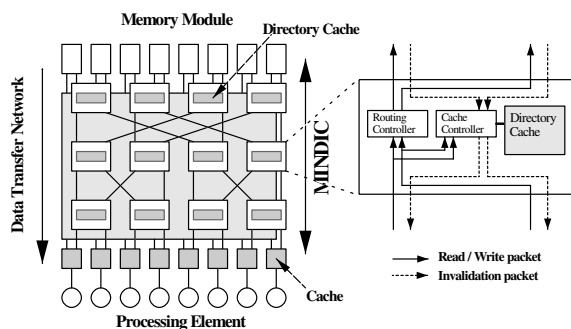


図 1 MINDIC の構造

スイッチングエレメントを設計及び実装し、動作周波数及びハードウェア量の見積もりを行なう。

2. MINDIC

2.1 MINDIC の概観

MINDIC の問題点は、縮約したとはいえ共有メモリにディレクトリを持たせた方法自体に起因している。しかし、各スイッチにキャッシュやディレクトリをそのまま載せることは、既存の方式同様、スイッチ内のメモリ容量と転送時間の遅延を増大させてしまう。MINDIC の評価により、中規模並列計算機における共有データに対するキャッシュは比較的短時間で更新されることがわかった。この結果を踏まえ、本稿では、スイッチ上に短期間のみディレクトリを動的に設ける方式を提案する。

第 1 図に本稿で提案するテンポラリディレクトリを保持するディレクトリキャッシュ(DC)を持つスイッチで構成される多段結合網 MINDIC の構造を示す。MINDIC 同様、キャッシュは PE に直結しており、高速にアクセスすることができる。また、キャッシュと共有メモリは、PE から MM のアクセスリクエストパケット及びキャッシュ制御用パケットの転送を行う MINDIC と、キャッシュラインのブロック転送を行なうデータ転送用の結合網の分離された 2 つの結合網を介して接続される。データ転送用の結合網として単純な構成で高速なクロスバなど様々なネットワークを利用することができる。書き込みは Write Through 方式の Direct Write を用い、MINDIC 内では読み出し要求、書き込み要求、無効化要求の短いパケットのみを転送する。MINDIC を用いたキャッシュコンシステンシの管理の対象とするのは共有データ領域に限定し、複数 PE で共有することのない命令データおよびローカルデータはキャッシュコンシステンシをとらずに管理する。

MINDIC の最大の特徴は、共有メモリを構成する各 MM にはディレクトリを設けずに、MINDIC のスイッチ内に数 K エントリのディレクトリキャッシュ(DC)を設け、キャッシュラインの共有情報を短期的に管理する点である。DC は小容量のメモリを用いるため、共有情報の登録ができるキャッシュラインの数が制限される。このため、DC はキャッシュラインの共有情報を一時的に保持するものとなっている。

2.2 構成要素の基本動作

MINDIC において、各スイッチ内の DC は、図 2(a) に示すように、入力リンクに対応する PE のキャッシュラインの共有情報を示すビットマップを保持する。一般のキャッシュ同様 DC は転送アドレスの一部をインデックスとして参照される。PE による MM の読み出し要求が発生した際、PE に接続されたキャッシュにデータが存在しない場合読み出し要求パケットが MINDIC を介して MM へ転送される。この際、要求パケットが通過する各スイッチ内の DC に読み出しデータの共有情報が登録される。読み出し要求により MM から読み出されたデータは、データ転送用ネットワークを介して PE へ転送される。

PE で書き込み要求が発生すると、書き込み要求パケットが MINDIC を介して共有メモリへ転送される。この際、パケットが通過した各スイッチで DC が参照され、DC のエントリにヒットすれば、図 2(b) に示すように登録されている共有情報に従って無効化パケットが生成される。生成された無効化パケットはキャッシュラインを保持する全ての PE へマルチキャストされ、ラインを無効化する。このようにしてキャッシュの一致制御を行う。

スイッチの基本動作は以下の通りである。MINDIC のスイッチが取り扱うのは読み出し要求、書き込み要求、無効化要求の三種類である。

- 読み出し要求パケットが下位ステージから入力されたら、読み出しアドレスのインデックスを用いてスイッチ上の DC を参照する。DC でミスして登録可能であれば、入力リンクに対応するビットを 1、他を全部 0 として共有情報を登録する。DC にヒットした場合、パケットの入力リンクに対応するビットを 1 として OR-write する。
- 書き込み要求パケットが下位ステージから入力されたら、書き込みアドレスのインデックスを用いてスイッチ上の DC を参照する。DC にヒットした場合、対応するビットが 1 の入力リンク全てに無効化パケットを逆送し、エントリを無効化する。なお、書き込みデータも MINDIC 上で転送される。
- 上位ステージから無効化パケットが入力された場合、無効化パケットアドレスのインデックスを用いて DC を引く。DC にヒットした場合、得られた共有情報に従って無効化パケットを下位ステージにマルチキャストし、DC のエントリを無効化する。

PE のキャッシュは、無効化パケットを受け取った際、該当するキャッシュラインを保持していれば無効化を行う以外、特殊な操作の必要はない。同様に、MM はディレクトリを持たず、キャッシュ制御に関する特別な操作は何もしない。

2.3 転送プロトコル

MINDIC は前記した基本動作を行うが、DC には容量不足やコンフリクトにより共有情報が登録できない場合があり、そのような場合キャッシュの一致制御が正しく行な

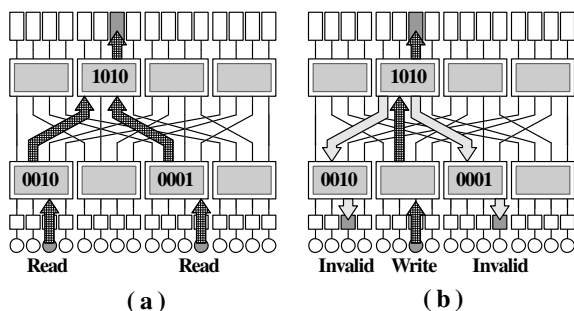


図 2 MINDIC の基本動作

えない．この問題を解決するために，以下の 3 つの転送プロトコルを提案する．

2.3.1 Dangerous bit プロトコル (Dan-P)

DC のインデックス毎に dangerous bit を設ける．同じインデックス内のエンTRIESに空きがない状態で新たな読み出し要求が入力された場合，そのインデックスに対応する dangerous bit をセットする．書き込み要求が入力された場合，DC にヒットせず，且つ書き込みアドレスから指定するインデックスの dangerous bit がセットされていたら，書き込み要求の入力リンク以外の下位ステージへの全リンクに無効化パケットをブロードキャストする．無効化パケットが上位ステージから入力された場合，DC にヒットせず，且つ，無効化パケットのアドレスで指定するインデックスに対応する dangerous bit がセットされているときは下位ステージへの全リンクに無効化パケットをブロードキャストする．dangerous bit がセットされたインデックスでは，先に登録されたENTRIES以外は利用不能になってしまう．このようなインデックスの共有情報と dangerous bit は，バリア同期が発生した時に一斉にクリアする．

2.3.2 Eviction プロトコル (Evi-P)

読み出し要求が入力された際，DC において読み出し要求のアドレスから指定するインデックス内のENTRIESに空きがない場合，先に登録されていたENTRIESを消滅させる．そして消滅させるENTRIESの共有情報を示すビットマップに従って，該当するキャッシュラインの無効化パケットをマルチキャストしてラインを無効化する．こうして空いたENTRIESに新たな読み出し要求に対する共有叙法登録を行う．こうすると読み出し要求は必ず DC へ登録されるので，キャッシュの一致制御の問題が解消される．

2.3.3 Invalidation Broadcast プロトコル (Bro-P)

MM にディレクトリを設けないという基本構造に反し，MM にキャッシュライン毎に 1 ビットの無効化ブロードキャストビットを設ける．DC においてアドレスで指定するインデックスのENTRIESに空きがない状態で新たな読み出し要求が入力された場合，MM の対応するビットをセットする．書き込み要求が MM に到達した際にこのビットがセットしてあった場合，MM は無効化パケットを下位ステージに発行する．このパケットを受け取った

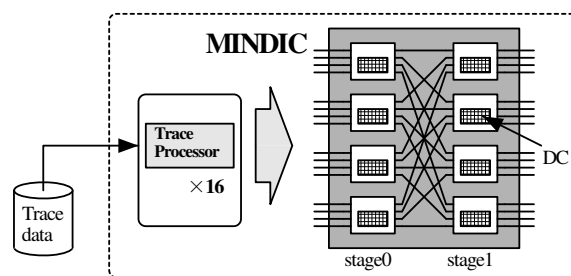


図 3 トレースドリブンシミュレータの構成

スイッチは，下位ステージへの全リンクに無効化パケットをブロードキャストする．こうすることにより，DC に共有情報が登録できなかったラインに対する書き込み要求時も確実にそれらのラインを無効化することができる．

3. 転送プロトコルの評価

3.1 トレースドリブンシミュレータ

トレースドリブンシミュレーションにより MINDIC の各転送プロトコルについて評価を行った．トレースデータには，MINDIC をキャッシュ制御機構として用いたスイッチ結合型並列計算機 SNAIL-2⁶⁾ の，クロックレベルシミュレーションによるメモリアクセス命令のトレースを用いた．トレースデータは，MINDIC におけるデータ転送用ネットワークである PBSF ネットワークへ PE が入力するメモリアクセス要求を抽出し作成した．トレースデータの抽出の際には，PE 数は 16，各 PE のキャッシュサイズは 256KB，連想度は 2，キャッシュラインサイズは 32byte とし，並列ベンチマークプログラム集 SPLASH-2 (Stanford Parallel Applications for SHard memory-2)⁹⁾ の中から 4 つのアプリケーションをそれぞれ実行した．

今回 MINDIC の評価に用いるシミュレータは，本研究室で開発された C++ 言語用のクロックレベル汎用並列計算機シミュレータライブラリ ISIS¹⁰⁾ を用いて実装されており，クロックレベルシミュレータとして動作可能である．

実装したシミュレータの構成を図 3 に示す．トレースプロセッサはトレースデータから入力パケットを生成し，スイッチに投入可能であれば投入する．MINDIC は 2 段の stage で構成され，合計 8 つのスイッチを有し，16PE での評価を行うことができる．各スイッチは，下位ステージ (PE 側) からの書き込み/読み出し要求パケットを上位ステージ (MM 側) に転送する順方向ポートが 4(入力)×4(出力)，上位ステージからの無効化パケットを下位ステージに転送する逆方向ポートが 4(入力)×4(出力) のポートを持ち，DC は連想度 1~4，256~65536 エンTRIESで構成される．

3.2 性能評価

シミュレーションにより，MINDIC のスイッチに 3 つのプロトコルそれぞれを用いた場合の性能を評価した．図 4 に，PE 側よりスイッチに入力された読み出し，書き込み要求に対する DC のキャッシュヒット率を，図 5 に，

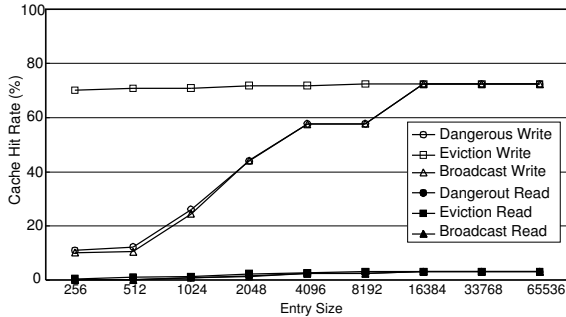


図4 プロトコル毎のキャッシュヒット率 (Radix, 1-way)

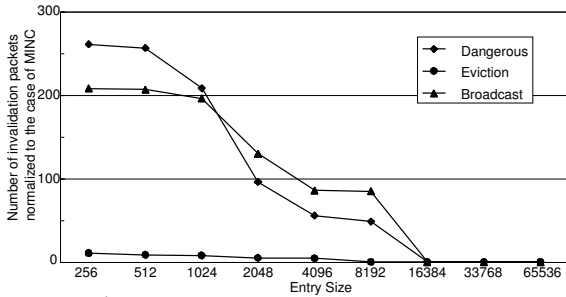


図5 プロトコル毎の無効化パケットの発生数 (Radix, 1-way)

MINDICによりPE側へ送出された無効化パケットの総数(MINCでのPE側へ転送された無効化パケットの総数を1として正規化した値)を示す。アプリケーションはRadixを使用し、DCの連想度は1で評価を行った。

図4の書き込み要求に対するDCヒット率を見ると、16384エントリ以上では全プロトコルにおいて70%程度と高いDCヒット率であることが確認できる。また、Dan-P, Bro-Pにおいて、8192エントリ以下ではエントリが小さくなるにしたがってDCヒット率が低下してしまいましたが、Evi-Pではエントリ数による影響が少なく、少ないエントリ数においても常に高いDCヒット率を維持することができた。

図5は、この際の無効化パケットの発生数を示した図であるが、DCのエントリ数が大きくなると全てのプロトコルにおいて無効化パケットの発生数が少なくなっていることがわかる。16384エントリ以上にすれば無効化パケットの発生数はMINCにおいて発生する場合の0.5倍程度となっており、十分なエントリがあればDCを用いることで無効化パケットの発生数を抑えられる。また、Dan-P, Bro-Pでは、8192エントリ以下で無効化パケットが大幅に増加してしまうが、Evi-Pでは4096エントリ以下でやや増加する程度に抑えられることがわかった。

これらの結果から、常に高いDCヒット率を示し無効化パケットの発生数を低く抑えることのできるEvi-Pが、3種類のプロトコルにおいて最もキャッシュ制御用ネットワークの混雑を抑えるのに有効な転送プロトコルであるといえる。

次に、図6, 7, 8に、無効化パケットの発生数の変化が大きい4096~32768エントリにおける、無効化パケッ

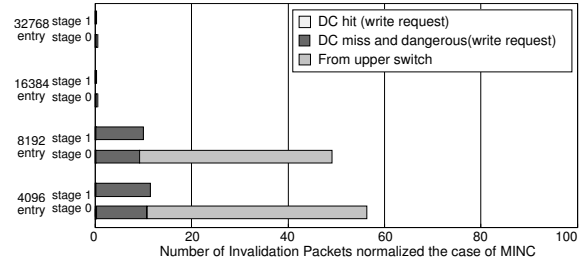


図6 無効化パケットの発生数 (Radix, 1-way, Dan-P)

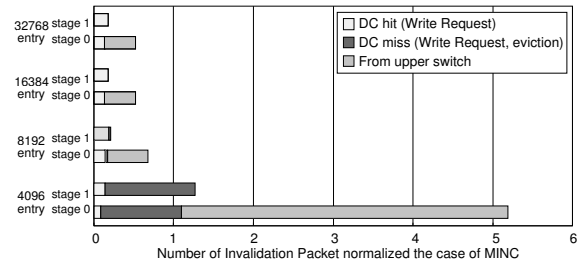


図7 無効化パケットの発生数 (Radix, 1-way, Evi-P)

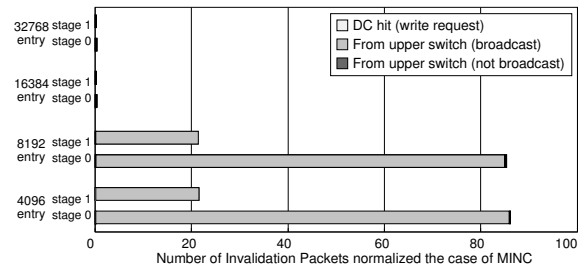


図8 無効化パケットの発生数 (Radix, 1-way, Bro-P)

トの発生数の詳細を示す。各グラフは、各ステージにおいて下位ステージへ出力される無効化パケットの総数を表しており、無効化パケットの発生原因別に色分けされている。Upper switchとは上位ステージから受け取った無効化パケットにより発生した無効化パケットのことである。

図6で、Dan-PはDCが8192エントリ以下で無効化パケットが大幅に増加してしまっている。stage1の内訳を参照すると、上位ステージであるstage1から下位ステージであるstage0へ転送された無効化パケットの99%程度が、書き込み要求時のDC miss時に dangerous bit がセットされていたことにより発生した無効化パケットのブロードキャストが原因となっている。stage0のFrom upper switchが示すように、この無効化パケットはstage0に転送されて更に増幅されるため、無効化パケットの大幅な増加につながってしまう。また、図8のBro-Pでも、DCが8192エントリ以下になると無効化パケットが多くなっている。書き込み要求時にMMのブロードキャストビットがセットされていたことによりMMから発生する無効化ブロードキャストパケットはstage1に入力されstage0へブロードキャストされる。この無効化ブロードキャストパケットを受け取ったstage0では更にPE側へ無効化パケットをブロードキャストするので、無効化パケットの数が大幅

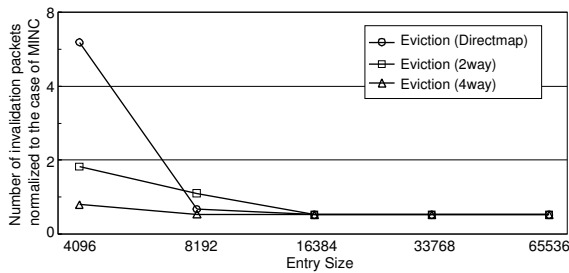


図 9 連想度毎の無効化パケットの発生数 (Radix, Evi-P)

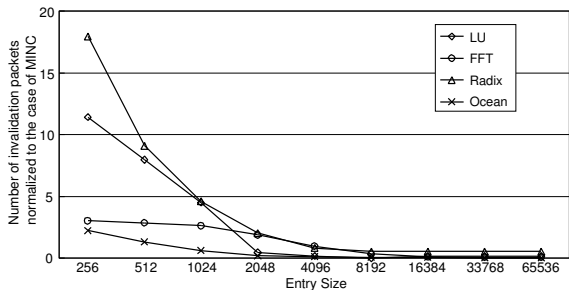


図 10 Application 毎の無効化パケットの発生数 (Evi-P, 4-way)

に増幅してしまう。

Evi-P では、読み出し要求時に DC にミスした場合、すでに DC に登録されていたエントリを追い出すことにより発生する無効化パケットの影響が大きい。しかし、図 7 を見ると、無効化パケットの発生数自体が少なく抑えられており、無駄な無効化パケットの発生を十分に抑制できており、効率的なキャッシュ制御が可能であることがわかる。

次に、無効化パケットの抑制に最も有効な Evi-P における、DC の連想度と無効化パケットの発生数の関係を図 9 に示す。DC のエントリ数が小さいときは連想度が高いほど無効化パケットの発生を抑制することができるが、DC が 16384 エントリ以上あれば連想度 1,2,4 とした場合の無効化パケット発生数は大差なくいずれも低く抑えることができる。全てのエントリにおいて連想度を 4 とすると最も無効化パケットの発生数を少なくできることがわかった。

最後に、Evi-P で連想度 4 としたときの無効化パケット発生数を図 10 に示す。4 つのアプリケーション全てにおいて、Evi-P を用いた連想度 4 の DC が 4096 エントリ以上あれば無効化パケットの発生数を MINC より少なくすることができることが確認できる。

3.3 MINDIC スイッチの実装

16PE、16MM の MINDIC を構成するスイッチを設計し、Verilog-HDL を使用して RTL 記述した。

MINDIC の各スイッチは下位ステージ (PE 側) から上位ステージ (MM 側) へのメモリアクセス要求転送用に 4 入力 4 出力、上位ステージから入力された無効化要求パケットを下位ステージにマルチキャストするために 4 入力 4 出力、合計 8 入力 8 出力のポートをもつ。図 11 に、設計したスイッチの内部構成を示す。

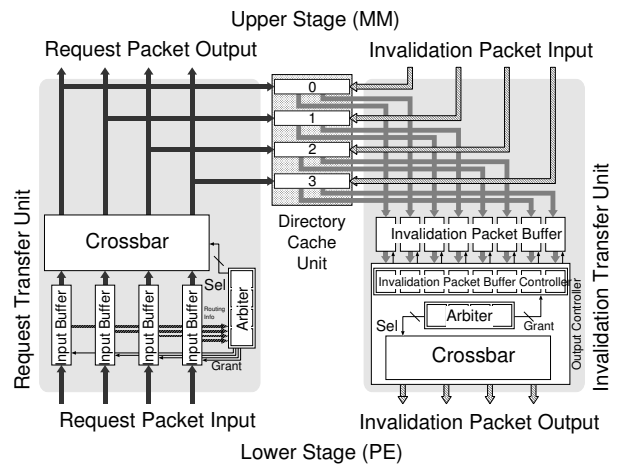


図 11 MINDIC スイッチの構成

下位ステージからのメモリ読み出し/書き込み要求は、スイッチのリクエスト転送部 (図 11 の左側ユニット) にある Input Buffer へ入力される。これらの要求は Crossbar を介して上位ステージへのリンクへ出力されるが、この際に出カリンク毎に設けられた DC が参照され、エントリの登録や更新、無効化パケットの生成が行なわれる。読み出し要求の場合、読み出すキャッシュラインの共有情報が DC へ登録される。書き込み要求の場合、DC からの共有情報をもとに書き込むラインを共有している PE キャッシュを無効化するための無効化要求が生成される。生成された無効化パケットには、下位ステージへマルチキャストするためのビットマップが付加され、無効化パケット転送部にある Invalidation Packet Buffer に蓄えられる。パッファに蓄えられた無効化パケットは、Output Controller により下位ステージへマルチキャストされる。

上位ステージからの無効化パケットは、スイッチ内の無効化パケット転送部 (図 11 の右側のユニット) に入力され、DC を参照する。この DC は上位ステージとのリンク毎に設けられており、リクエスト転送部の対応する上位ステージへの出力リンクと共有している。DC を参照することにより無効化パケットにはキャッシュラインの共有情報を表すビットマップが付加され、無効化パケットパッファに蓄えられる。パッファに蓄えられた無効化パケットは、前述と同様にアウトプットコントローラにより下位ステージへとマルチキャストされる。

DC は、下位ステージからのメモリアクセス要求と、上位ステージからの無効化要求の両方に同時に対応するために、Dual port RAM で実装されている。同アドレスに対する参照が DC メモリの 2 つのポートで同時に発生する場合はアービタが働くことにより、排他的に DC へのアクセスが行なわれる。

3.4 ハードウェア評価

SYNOPTSYS 社の論理合成ツール Design compiler を用いて CMOS エンベデッドアレイ ASIC 0.18 μ m ライブラリを利用した論理合成を行い、動作速度とハードウェア量の評価を行った。

共有メモリサイズを 256 MByte, キャッシュラインサイズを 32 Byte とし, 各スイッチ内の DC のエントリ数は 4096, 8192, 16384, 連想度は 2 または 4 のそれぞれの組み合わせを実装した。

DC に使用するメモリ及び無効化パケットバッファを除いた, スwitch の論理合成結果を表 1 に示す。全てのプロトコルをタイミング制約 4nsec のもとで論理合成しており, 比較的高速な動作が見込まれる。DC のエントリ数が 8192 の場合, どのプロトコルも連想度 2 で 43000 ゲート, 連想度 4 で 54000 ゲート程度と, 実装可能なハードウェア量であることが確認された。無効化パケットの抑制に最も有効な Evi-P は, DC の追い出しに使用する LRU ユニットを備えるためハードウェア量が若干大きくなっているものの, 他のプロトコルと比較しても数百ゲートの差であり, ほぼ同程度の回路規模で実装可能である。

連想度を 4 にすると, 連想度を 2 にした場合と比較してゲート数が約 26.1% 増加してしまう (Evi-P)。十分なエントリ数が確保できる場合は連想度を 2 としてハードウェア量を節約することが可能である。

表 1 論理合成結果 (タイミング制約 4nsec)

連想度	エントリ	Dan-P	Evi-P	Bro-P
		ゲート数		
2	8192	43461	43232	42016
4	8192	54592	54504	53690

また, ディレクトリ管理に使用するメモリ容量は計算により検討が可能である。共有メモリサイズ 256MByte で, 16PU, 16MM に対応した, 8 つのスイッチから構成される MINDIC のディレクトリ管理に必要な全メモリ容量を表 2 に示す。各スイッチ内にある DC のエントリ数は 8192 とした。MINDIC ではディレクトリ管理に用いられるメモリ容量を大幅に削減可能であることを確認するため, MM ヘディレクトリを持たせるフルマップ方式と, MINC で用いた RHBD 方式によりディレクトリを縮約した場合に必要なディレクトリ容量を比較対象として表示する。

Dan-P, Evi-P で DC に必要なメモリ容量は 8 つのスイッチ合計で 160KByte 程度であり, スwitch 内に実装することが十分に可能である。また, Bro-P の場合は MM に合計 1Mbyte の無効化ブロードキャストビットが必要となるが, それでも RHBD 方式の 1/10 程度のメモリ容量で実現可能である。

表 2 ディレクトリ管理に必要な全メモリ容量 (byte)

Dan-P	Evi-P	Bro-P	RHBD	Full Map
154K	160K	152K + 1M	10M	25M

4. 結 論

一時的な共有情報であるテンポラリディレクトリを内部に持つスイッチで構成される多段結合網 MINDIC を提案し, キャッシュ一致制御のための 3 つのプロトコルを示した。シミュレーションによりキャッシュヒット率及び無

効化パケットの発生状況の比較検討を行った結果, 十分な DC のエントリがあれば全てのプロトコルにおいて無効化パケットが抑制できることが確認できた。また, Eviction プロトコルでは DC のエントリ数が少なくとも無効化パケットが大幅に増加することではなく, 3 種類プロトコルのうちで最も有効に無効化パケットによるネットワークの負荷を少なくできることがわかった。更に, Verilog-HDL を用いて実装した結果, MINDIC のスイッチは高速動作可能であり, 約 54000 ゲート及び 160KByte 程度の小容量メモリという現実的な回路規模で実現可能であることが示された。

参 考 文 献

- 1) H.E. Mizrahi, J.L. Baer, E.D. Lazowska, and J. Zahorjan. Introducing memory into the switch elements of multiprocessor interconnection networks. In *Proc. of 16th ISCA*, pp. 158–166, 1989.
- 2) R.Iyer and L.Bhuyan. Design and evaluation of a switch cache architecture for cc-numa multiprocessors. *IEEE Trans. on Comput.*, Vol. 49, No. 8, pp. 779–797, 2000.
- 3) A.K. Nanda and L.N. Bhuyan. Design and analysis of cache coherent multistage interconnection networks. *IEEE Trans. on Computers*, Vol. 42, No. 4, pp. 458–470, 1993.
- 4) R. Iyer, L. N. Bhuyan, and A. Nanda. Using switch directories to speed up cache-to-cache transfers in cc-numa multiprocessors. *Proc. of the 14th Int'l Parallel and Distributed Processing Symposium (IPDPS'00)*, pp. 721–728, 2000.
- 5) T.Midorikawa, T.Kamei, T.Hanawa, and H.Amano. The minc chip: Multistage interconnection network with cache control mechanism chip. In *Proc. on ASICON*, pp. 249–252, 1998.
- 6) T.Midorikawa, D.Shiraishi, M.Shigeno, Y.Tanabe, T.Hanawa, and H.Amano. Snail-2: a sss-min connected multiprocessor with cache coherent mechanism. In *Proc. of Parallel and Distributed Computing, Applications and Technologies*, pp. 17–24, 2002.
- 7) 西村克信, 工藤知宏, 天野英晴. Pruning cache を用いた分散共有メモリのディレクトリ構成法. *情報処理学会論文誌*, Vol. 39, No. 6, pp. 1644–1654, 1998.
- 8) 緑川隆, 田辺靖貴, 天野英晴. ディレクトリキャッシュスイッチを持つキャッシュ制御用多段結合網の検討. *電子情報通信学会コンピュータシステム研究会, CPSY2003-14*, pp. 49–54, 2003.
- 9) S.C.Woo, M.Ohara, E.Torrie, J.P.Singh, and A.Gupta. The splash-2 programs: Characterization and methodological considerations. In *Proceedings of the 22nd International Symposium on Computer Architecture*, pp. 24–36, 1995.
- 10) 若林正樹, 天野英晴. 並列計算機シミュレータの構築支援環境. *電子情報通信学会論文誌*, Vol. J84D-I, No. 3, pp. 1–10, 2001.