システムＬＳＩ設計技術　101－14
（2001. 5. 18）

スケジューリングを考慮した多段結合網スイッチチップの実装

森村 知弘† 田中 健介† 岩井 啓輔‡ 天野 英晴†

†慶應義塾大学

‡当時、慶應義塾大学、現在、防衛大学校

〒223-8522 横浜市港北区日吉3-14-1

{morimura,kensuke,iwai,hunga}@am.ics.keio.ac.jp

あらまし 我々は、コンパイラによる静的スケジューリングを考慮したマルチプロセッサASCA用相互結合網として多段結合網R-Clos IIを提案している。R-Clos IIは、Clos網を複数個、階層的にクラスタリングして相互接続した不等距離間接網である。本研究では、コンパイラがデータ転送をスケジューリングするのを想定したR-Clos IIの要素スイッチとしてMGF switch architecture を提案し、これをゲートアレイに実装した。MGFは、内部に2つのクロスバスイッチに接続した独立した通信路備えており、スケジューリングパケットの転送が乱すことなく、スケジューリングされていないパケット転送も高スループットで行える構成をとっている。

実装したチップのVerilog-HDLにおけるシミュレーションによる転送性能の評価によると、一般的な最もシンプルな構成のスイッチと比較して、40%のハードウェアの増加で平均転送遅延が最大で約60%に抑えられることが確認できた。

キーワード　多段結合網, Clos網, マルチプロセッサ, 静的スケジューリング, スイッチアーキテクチャ

An Implementation of a switch chip for compiler's static data transfer scheduling

Morimura T.† Tanaka K.† Iwai K.‡ H.Amano†

†Keio University

‡National Defence Academy

3-14-1, Hiyoshi, Kohoku-ku, Yokohama 223, JAPAN

{morimura,kensuke,iwai,hunga}@am.ics.keio.ac.jp

Abstract We proposed a multistage interconnection network R-Clos II for the multiprocessor system ASCA to schedule the data transfer statically. R-Clos II is a multistage network which consists of hierarchically clustering multiple Clos networks with extra intermediate stages. To schedule communications between processors easily, it is also important that a switch architecture supports the compiler's data transfer scheduling. Therefore we propose a new switch architecture, called MGF switch architecture, which has two sets of the transfer channel with a crossbar for scheduled packet and non-scheduled packet respectively.

In this paper, the chip feature which we implemented are described and we evaluate it's transfer ability by Verilog-HDL simulation. Additionally, the average latency of MGF switch with 40% larger size hardware are about 40% better than that of single channel normal switch at the most.

key words MIN, Clos Network, multiprocessor, static scheduling, switch architecture

1 はじめに

マルチプロセッサシステムにおいて、プロセッサ間の通信オーバヘッドがしばしばシステム全体の並列処理の性能を大きく低下させる要因となっている。プロセッサ間通信を含めた並列処理の効率の良い実行を達成するために、我々は、マルチプロセッサシステムASCA[1](Advanced Scheduling oriented Computer Architecture)を提案している。ASCAでは、効率の良い並列処理手法の一つであるマルチグレイン並列処理手法[6]にターゲットを置き、これを行う自動並列化コンパイラの静的解析に基づく高速化手法をサポートすることを目標としている。このため、ASCAのプロセッサ・メモリ・ネットワークの各要素は、コンパイラによってその動作が完全に把握可能な単純なもので構成される。これにより、動的な決定性をもたないブロック内においては、完全にコンパイラによってスケジューリングすることが可能である。我々は、データ転送のオーバヘッドを隠蔽するために可能な限りコンパイラによってこれを管理することを目標とし、このためのプロセッサ間相互結合網として階層構造の多段結合網R-Clos II[2]を提案した。

本報告では、R-Clos IIの要素スイッチのLSIチップ実装について述べる。次の章では、マルチプロセッサASCAとその相互結合網R-Clos IIについて述べ、3章では既存のスイッチアーキテクチャについて考察する。4章では、ASCAシステムに対する既存のスイッチアーキテクチャの問題点を解決した新しいMGF switch architectureを提案し、この構成と、LSIチップへの実装について述べ、5章ではVerilog-HDLによるシミュレーションの転送評価を示し、通常の構成のスイッチと転送性能とハードウェア量について比較し、6章ではまとめと今後の課題について述べる。

2 マルチプロセッサシステムASCAと多段結合網R-Clos II

2.1 マルチプロセッサASCA

ASCAは、スケーラブルな共有メモリ型マルチプロセッサシステムで256から4096プロセッサを階層構造の大規模相互結合網で接続して構成される(図 1)。ASCAは様々なレベルでの並列処理(粗粒度並列処理(マクロデータフロー)、中粒度並列処理(ループレベル並列性)、近細粒度並列処理)を行うマルチグレイン並列処理に対応するために、階層化されたクラスタ構造を内包している。ASCA上では、粗粒度並列処理におけるMacro Task(他のステートメントへの飛び込みのないBasic Blockの集まり: 以下 MTと示す)が、これらのクラスタに動的に割り当てられ、それぞれのMT内では、ループレベルの中粒度並列処理、またはステートメントレベルの近細粒度並列処理が行われる。

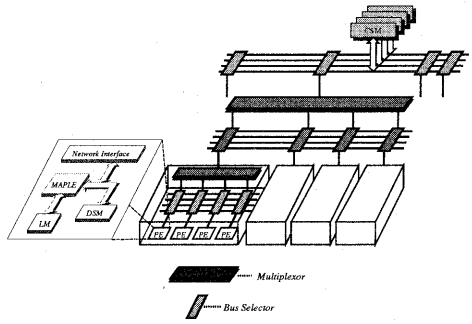


図 1: Overview of ASCA

マルチプロセッサASCAでは、マルチグレイン並列処理を考慮した構成をとるのに加え、並列処理に伴う通信遅延などの分割オーバヘッドを抑えるために、コンパイラによる静的スケジューリングをサポートする。ASCAのProcessing Element、MAPLEは、プロセッサ部MAPLE coreとデータ転送を司るDTC(Data Transfer Controller)から構成される。カスタムプロセッサMAPLE coreは、DLX[3]命令セットを拡張させた命令セットを持ち、単純な5段のパイプライン構造で、コンパイラがパイプラインレベルでの動作を予測することが可能である。

メモリとプロセッサ(キャッシュ)間およびPE間のデータ転送は、コンパイラによって指示され、データ転送専用のData Transfer Controller(DTC)と呼ばれるプロセッサによって行われる。DTCは、コンパイラによって吐き出されたDTC用の命令コードを実行し、キャッシュに予めデータを転送して用意し、極力キャッシュミスが起らないようにする。DTCによって予めデータを用意するためには、データ転送遅延が予測できなければならない。

2.2 多段結合網R-Clos II

R-Clos IIは、3段の多段結合網であるClos網[4]を、複数個階層構造に拡張した多段結合網であるR-Clos[5]に改良を加えた階層構造多段結合網で、ASCA用の相互結合網として提案した。R-Clos IIは、以下の特徴を持つ。

- 集中共有メモリ(CSM:centralized shared memory)とPEを広帯域データ転送
- クラスタ内の近接PEとの高速データ転送
- 容易にスケールアップ可能な再帰的階層構造
- 静的スケジューリングされたパケットのデータ転送遅延を保証

これらの特性を満たすために、R-Clos IIは、クラスタ内高速データ転送用ネットワークとしてClos網に着目し、これを再帰的かつ階層的に複数個接続することによって、CSMとの広帯域データ転送を実現するとともに、規模拡張性を持っている。

階層構造の基本ネットワーク(以降、level-1 network)となるClos網は、図 2に示すようにクラスタを構成する基数をkとした場合(図中では4)、k個の$k \times k$のクロスバスイッチを並べ、これをshuffle exchangeによって3段接続

した多段結合網である。level-1 network内の転送は、可能な限りコンパイラによって静的にスケジューリングが施され、Clos網の転送経路の冗長性を活かして高速に行われる。

CSMは、R-Clos IIの最上位階層に接続され、どのPEからも等しく参照することが可能である。CSMへのアクセスは、主にMTの開始、終了時に発生(Pre-Load, Post-Store)するため、データ転送遅延に関しては近細粒度並列処理のように高速性を要求されることがないが、大量のデータ移動が起るため、バンド幅を確保する必要がある。このため、ASCA用の多段結合網R-Clos IIは、コンパイラのスケジューリングを考慮した高速クラスタ内転送と、動的に行われる大量のデータ転送をサポートするスイッチアーキテクチャが必要となる。

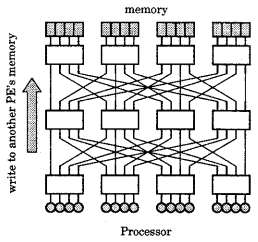


図 2: Clos Network (16 PEs)

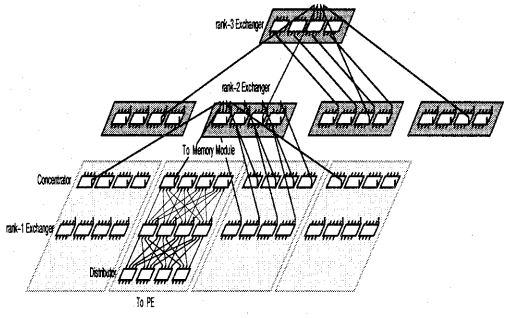


図 3: Structure of R-Clos II (256 PEs)

3 switch architectureの考察

3.1 従来の一般的な多段結合網のスイッチアーキテクチャ

一般的に多段結合網に用いられるスイッチの構成としては、図4のように、クロスバスイッチ、出線競合調停用のアービタ、そしてパケットバッファを持つものが多い。パケットバッファは、出線競合などでパケットが衝突した場合に、パケットを格納するために用いられ、FIFO(First In First Out)の形をとる。パケットバッファの位置は、入力側につける input queueing方式と出力側につける output queueing方式があり、output Queueing方式が優れている[7]ことが知られている。しかしながら最近になって効率良く処理をパイプライン化することにより input queueingでも変わらない性能を示すことが可能[8]である。図4は、input queueing方式のものを示している。まずはじめに、最も一般的な構成である input queueing方式について述べ、次に output queue方式を述べる。

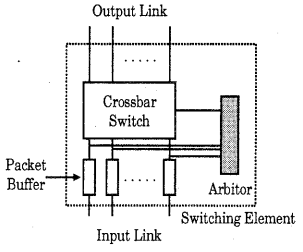


図 4: 一般的なスイッチの構成 - input queueing -

Input Queueing方式

input queueing方式は、queueの構成の仕方で、次の2つに分けることができる。

- 各入力ポートに対してsingle bufferで構成するもの
- 各入力ポートに対してmultiple bufferで構成するもの

single queueによる方式では、衝突によってブロックされたパケットは、行き先に関わらず、このqueueに格納される。そしてqueueの先頭にあるパケットのみが、転送される。仮に先頭パケットが要求した出力ポートが、busyである場合、これに続くパケットもブロックされてしまう(これをHOL(Head Of Line blocking delay)と呼ぶ)という欠点がある。このため、この方法によるスループットはおよそ60%当りで飽和してしまう[7].

1つのinput bufferにmultiple queueを実現する方式は、図5に示したように、1つの入力ポートに各出力ポートに対応したqueueを用意する。このためsingle queueのようなHOLのオーバヘッドはない。これらのmultiple queueは、普通は静的に配置されるが、不均一のトラフィックに対しては、bufferの浪費につながり効率が悪い。

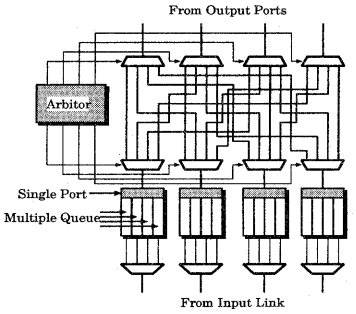


図 5: multiple queueを持つinput queueing

そこで、トラフィックに応じてqueueのスペースを動的に割り当てる、DAMQ[9](Dynamically Allocated Multi-Queue)が提案された。DAMQでは、各queueの先頭のパケットのみが送信される権利を有するが、同じポートのqueueのうちの一つしか同時に送信することができない。それゆえ一つのinput bufferにつき一つのデータ出力となる。

この他に、各入力ポートに各出力に対応した複数のinput bufferを持つ方式(crosspoint queueing[10]という)及び、これの問題点を改善させた方法として出力ポートとは関連づけられてないk個(スイッチの入出力数と同じ数)のinput bufferを各入力ポートに持たせるなどの方法がある。

Output Queueing方式

output queueing は、各出力ポートにそれぞれoutput bufferを持たせる方式と、全出力ポートで一つのoutput bufferを共有する方式に分類できる。後者の方式は、前者よりも良いパフォーマンスを示し、ハードウェアも簡単に実装可能で最も一般的である[11]。以降は、後者の方式について述べる。この方式は、単一のバッファ内に複数のqueueを持つ方式であるため、一時にreadもしくは、writeによるアクセスが1回に制限されることによるオーバヘッドが大きい。バッファが単一のモジュールで実装されている場合、このオーバヘッドをなくし、毎サイクルごとに各ポートがアクセスできるようにする方法は、次の2つがある。

まず1つは、メモリをk(スイッチの入力(出力)数)flitの幅のブロックとして扱い、全てのread、writeはk flit幅のchunkと呼ばれる単位で行う方法である。各ポートは、少なくともk cycleに1回、chunk単位にread、writeすることができる。ただし、bufferに書く前に各入力ポートからフリットをk flitにまとめる作業と、読み出し前にk-flitのchunkをシリアル化する作業が必要となる。この結果、毎サイクルごとに1 flitずつ処理するのと同じスループットを得ることが可能となる。このような手法は、IBMのSP2[12]で用いられている。

もう1つの手法は、メモリモジュールを複数のバンクに分け、一つのバンクをk flit幅で構成する代わりに、モジュールをk-flit幅のバンクとして実装する[10]。そして、1度にk flitのデータを読み書きするのではなく、bank 0からbank $k-1$まで1 cycleずつ順番に、パイプライン化して読み書きを行なう。これにより、ある入力ポートがbank 1に書き込んでいる間に、他の入力ポートがbank 0に書き始めることが可能である。この手法は、先のk flit幅の大きいメモリを使用する方法とほとんど同じパフォーマンスを示すが、より簡単に実装することが可能である。

3.2 従来のスイッチアーキテクチャのまとめと問題点

これまでに提案されている主なスイッチアーキテクチャについて以下のようにまとめることができる。

- いずれの手法もHOLを防ぎ、高スループットを実現することを目標としている
- input queueing , output queueing ともに、HOLを防ぐためにqueue を複数用意している(virtual channelの実装)
- queue のallocation には、動的にする方式と、静的にする方式があり、
- 動的なallocation の方が、高スループットを実現できるが、
- その分、クロスバのハードウェアが複雑なものになる.

これまでに述べられた従来のスイッチアーキテクチャでは、ASCAシステムのマルチグレイン並列処理の処理形態において以下のような問題が存在する。

- データ転送遅延がクリティカルであるスケジューリングされたパケットと、動的に転送されるパケットの混在

この問題は、input queueing かoutput queueing の手法に関わらず存在する。これは、静的にスケジューリングされたパケットが動的に転送されるパケットにブロックされる場合が発生するためで、HOLの一種であると捉えることができる。HOLを克服する手段として提案された各種の手法は、出力ポートとのpacket queue との関係を取り扱っているため、パケットの種別(スケジューリングされているか/いないか)についてはまるで考慮されていない。そこで、次章では、従来のこれらのHOL を克服する手法を拡張した、パケットの種別に応じて仮想チャネルを形成するMGF switch architecture を提案する。

4 MGF switch architecture

4.1 MGF switch architecture 概要

MGF switch architectureでは、前章で述べた、スケジューリングパケットとスケジューリングされていないパケットの同一ネットワーク上の混在によるHOLを防ぐことを目的としている。しかしながら、単にスケジューリングパケットを優先するのみであるとスケジューリングされていない通信のバンド幅が極端に落ちてしまう。このため、MGFスイッチアーキテクチャでは、以下のような構成をとることによって、スケジューリングされていないパケットの転送スループットも可能な限り向上させることを念頭に置いて設計されている。

- スケジューリングパケット専用の通信チャネル(これをs-channelと定義する)の付加し、スケジューリングされていないパケットの通信チャネルと(c-channelと定義)合わせて、それぞれ別の独立した通信路を用意
- 独立した通信路のためのクロスバ(計、2つ)を用意
- パケットの種別(スケジューリングされている/されていない)をスイッチの入力側のデコーダで判別し、それぞれの通信チャネルに振り分けて転送
- 出力部では、2つの通信チャネルを調停してどちらかを出力させるが、s-channelにパケットが存在する場合は、そちらが必ず優先される

次に、MGF switch architecture におけるパケットの流れを示す。

1. 入力ポートに届いたパケットは、その種類(schedule/common)によってデマルチプレクサによって、適切な入力チャネルバッファに送信される
2. 入力チャネルに入ったパケットはqueueの先頭にくると、ルーティングタグに示された出力に送信される
3. 出力側で、そのポートを要求しているパケットの調停が行なわれ、勝ったパケットがマルチプレクサから出力バッファに送出される
4. もし、s-channelの出力バッファにパケットが存在すれば、それが出力ポートに送出される

5. c-channelの出力バッファにパケットが存在し、かつs-channelの出力バッファが空であれば、それが出力ポートに送出される。

このようにMGFでは、schedule packetの転送を優先させつつも、出力ポートが空いていれば、積極的にcommon packetも転送させていく効率の良い、高スループットな転送を実現する。

この構成を示したのが、図6である。

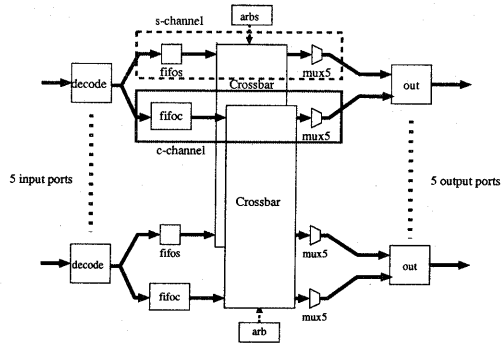


図 6: MGF switch の構成図

今回は、このような内部構成を持つ5入力5出力のスイッチを実装した。

4.2 packet format について

マルチプロセッサASCAシステムでは、以下の6種類の通信を行う。

- CSM(Centralized Shared Memory)への読み込み
  - PE → CSM (読み込み要求)
  - CSM → PE (読み込み)
- CSMへの書き込み
- 他PEの交信用メモリ(CM)への書き込み
  - 近細粒度 並列処理におけるスケジューリングされたデータ転送
  - 粗粒度並列処理におけるデータ転送
- 他PEへのブロードキャスト

このため、それぞれのパケットには、

- K ... パケットの種別を示す1 bit(scheduled/non-scheduled)
- F ... フリットの種別を示す1 bit(header/body)
- T ... 通信の種別を示す 2 bit
- RT ... ルーティングタグ ... 経路情報のタグ
- RP ... ルーティングタグポインタ ... 今どのタグが有効であるかを示す
- ID ... パケットのID
- FC ... フリット数

のような情報を32 bit で構成されるヘッダに持つ。通信の種別は、前述した6つ通信の種類のもので、これによってパケットフォーマットは異なる。この中で、最も重要なものは、パケットの種別を示すKで、スイッチはこのKによって、そのパケットがコンパイラによってスケジューリングされたものであるかどうかをチェックする。本来ならば、32bit を1 flitとして、一度に転送したいところであるが、チップのピン制約によって本実装では、16 bit が1 flitとなっており、2 flitでヘッダが構成されている。ただし、スケジューリングされたパケットのヘッダは、情報を可能な限り省いて、軽量高速転送を実現するために、ヘッダは16 bit 1 flitで実現されている。

4.3 スイッチの種類(モード)について

今回実装したチップは、R-Clos IIのスイッチを一つのチップで実現するため、以下のスイッチをサポートしなければならない。

- Distributor ... 入力側の一段目のスイッチ
- Exchanger ... 中間段のスイッチで、上位の階層、もしくは下位の階層にパケットを配送する
- Concentrator ... 出力側の最終段のスイッチ

これらのスイッチは、マルチキャストパケットの処理、パケットのルーティングタグの処理およびスイッチの入出力数が異なる。異なる動作をするスイッチを一つのLSIで実現するために、外部からチップにそのスイッチのモード(3 bit)を入力し、この入力にしたがって、チップはそれぞれのスイッチの動作を適切に行う。

4.4 スイッチの構成

MGF switch は、入力側からDECODE, FIFO, ARB, MUX5, OUTの5つのモジュールから構成されている。図7に、MGF switchのトップモジュールのブロック図を示した。

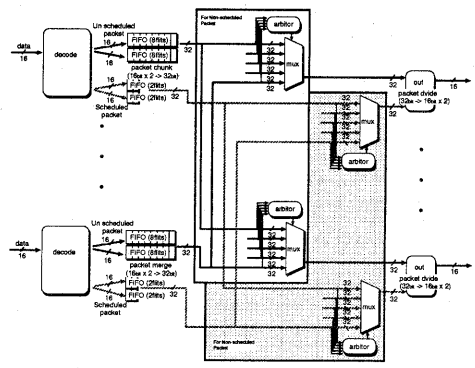


図 7: MGF switch トップモジュール ブロック図

以下、DECODEから順に説明する。

DECODE

スイッチの最も入力側にある、パケットのヘッダを解析するデコーダ。パケットの種別、フリットの種別、通信の種別、フリット数を解析し、スイッチの各モジュールに知らせる。パケットが入力にくると、K,FTによって、そのパケットがヘッダかどうか、そしてスケジューリングが施さ

れているかどうかをチェックし、そうであればs-channel にパケットを転送し、そうでなければc-channel に転送する。MGF switchでは、スケジューリングされていないパケットのフリットの間にスケジューリングされたパケットが割り込む場合があるため、これに的確に対応するために、パケットのフリットのMSBのKの値によって、スケジューリングパケットとそうでないパケットのフリット数をそれぞれカウントして制御している。

FIFO

調停時にパケットが格納されるパケットバッファモジュールである。s-channelのこのモジュールをfifosとし、c-channelのモジュールをfifoとして実装している。fifosでは、コンパイラによって衝突が起らないように制御されているためパケットバッファの深さは要求されない。このため、このモジュールは、36 bit × 2 word (計 4 flit) の循環バッファをレジスタによって実現されている。

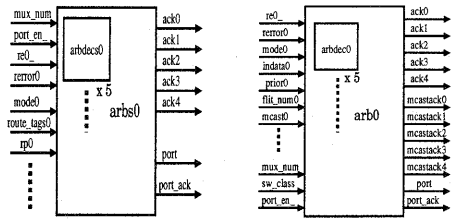
一方で、fifoのモジュールは、バッファの深さを要求されるためにチップの実装制限内で搭載可能な36 bit × 8 word (計 16 flit) 分の循環バッファをメモリセルを用いて実現している。

fifoもfifosも、36bit 中32 bitは、パケットの格納に用いられ、残り4bitで格納したパケットのモード (通信の種別) を保存している。本実装では、チップの制約上、32 bit で1 flitとすることができず、16 bit で1 flitとしたため、このモジュールで16 bitのflitを内部処理用の32 bitで1 flitとしてFIFO queueに保存している。FIFOは、ARBからre(読み込みOK信号) を受け取ると、次のデータ (32 bit) を出力に出す。

fifoがfifosと大きく異なるのは、マルチキャストパケットのための機構を実装していることで、mcastflag,mcastreqという信号線によって、出力に接続されたARB(arbitor)に、パケットがマルチキャスト要求を出しているかどうかを知らせる。mcastflag は、入力パケットがMulticast/Broadcastである場合、スイッチの階層 (rank) が、パケットのヘッダに記述された階層と同じであれば、activate される。mcastflagが立つと、mcastreq0-3までの信号がenableになり、各ARB に、multicast 要求ビットがたてられる。arb$n$($0 \leq n \leq 3$)で、Multicast/Broadcastパケットがポートを獲得すると、arbからmcastack$n$がかえり、mcastreqnはdisable される。全てのmcastreqnがdisable されると、mcastreqはdisableされ、次のclkでFIFOのraddrがインクリメントされ、次のパケットを出力する。

ARB

入力ポートの出力要求を調停するアービターモジュールである。s-channelのモジュール名はarbsで、c-channelではarbとして実装されている。arbs,arb共通に言えることであるが、アービトレション用モジュールは、各出力チャネルごとの出力ポート毎にある。よってこのR-Clos switch chipでは、5つのarbsモジュールがある。各モジュールは、出力ポート0に対応して順にarbs0,···,arbs4 (arb0,···,arb4)となっている。



(a) arbs モジュール　　(b) arb モジュール

図 8: ARB モジュール

パケットのヘッダ部をsub moduleのarbdecs(またはarbdec)で解析し、routing_tagとarbsが繋がっている出力ポート番号が一致した場合、そのパケットはこのポートを要求していると見なし、対応する要求フラグであるreq0,···, req4を立てる。

ここで、arbでは通信の種別に応じて*rank*という値が以下のように設定する。

- CSM ... rank = 2
- DSM ... rank = 4
- Multicast/Broadcast ... rank = 1

一方arbsでは、通信の種別は全て同じであるため、rankという値は設定されない。次にパケットの到着順にranksubという値が設定される。到着したときに値は5に設定されるが、調停に負けるごとに値はデクリメントされていく。

調停は、

1. rankを比較
2. ranksub を比較
3. 入力ポートを比較

という順で行われる。それぞれ値が小さいものが優先される。本来ならば、ポートごとの公平性を期すためにRound Robinによる調停を行うべきであるが、今回はチップの制限によってこの機能を取り入れることができなかった。

割り当て結果は、port と呼ばれる出力信号に、獲得入力ポート番号を載せることによって、multiplexor (MUX) に知らされる。パケットが到着し、調停に勝つとranksubは6に設定される。ranksubが6 に設定されると、パケット転送中の状態に遷移し、trans_stat と呼ばれる信号が1周期刻むごとに、対応する ack0,···, ack4をFIFOに返し、次のflitをFIFOより読み込む。ただし、flit_cntが max_cnt となる時のみ、ackを早めに返して、1clk だけ早く次のパケットのヘッダを読み込み調停の準備に入る。portは、ranksubが6である間、値が保持される。

MUX5

5入力1出力のmultiplexor(mux)。各入力ポートのFIFOからの入力データ (data0,···,data4) を、arbからの制御信

号 arb に従って、out モジュールへ出力する。これと同時に、そのデータ (フリット) のモード (通信の種別) も modeout に出力する。

OUT

出力モジュールで、s-channel と c-channel からのパケットを調停してスイッチ外に出力する。s-channel のパケットと c-channel のパケットのための出力バッファをそれぞれ 2 flit(32 bit x 2) ずつ持つ。s-channel, c-channel それぞれのパケットを出力バッファ s_inbuf, c_inbuf で独立して受信し、s_inbuf にパケットが入っている場合は、出力制御信号である output_sel を、1 にして s_inbuf のパケットを外に出力する。逆に、s_inbuf が空で、c_inbuf にパケットが入っている場合は、output_sel を 0 に設定し、c_inbuf のパケットを外に出力する。s_inbuf にパケットが 2 flit(16 bit x 2) 入った時に、c_inbuf が出力中でも、出力が中断されて代わりに s_inbuf の schedule packet が優先して外に出力される。32 bit で転送されたパケットを 16bit の細かいフリットに分割するために、word_sel という信号を用い、それぞれ upper word と bottom word に分割して、clk ごとに交互に指定して出力している。

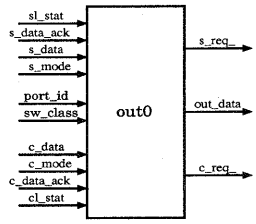


図 9: out モジュール

4.5 チップの実装

上位設計は、Verilog-HDL で各モジュール (DECODE, FIFO, ARB, MUX5, OUT) を記述し、top モジュールでこれらを接続した。

CAD は、VDEC で標準的に使用されている表 1 のものを用いた。

表 1: 設計用 CAD のバージョン

	ツール名	ベンダー名
Simulation	Verilog-XL	Cadence Design Systems, Inc. version: 3.0.p001
論理合成	Design Compiler	Synopsys, Inc. version: 2000.05
配置配線	Milkyway、ApolloGA	Avant! version: 1998.4.3.2.0.25
検証	Dracula DRC	Cadence Design Systems, Inc. version: REV. 4.8

実装にあたって、日立製作所のゲートアレイ (VDEC のサポートによる) を用いた。詳細は、次の表 2 のとおりである。

表 2: 利用したチップのテクノロジ

プロセス	0.35μm, PolySi : 1 層 メタル配線 : 5 層
実効ゲート長	0.25μm
電源電圧	3.3 V
チップサイズ	5.9 mm 角
信号ピン数	190
下地ゲート数	143 kG(2 NAND 換算)
パッケージ	BGA256

4.6 実装結果

Design Compiler で論理合成を行った後のゲート数を表 3 に示した。表中の BC は、Design Compiler における単位で、セル数を表す。日立のゲートアレイの場合、1 Cell = 2 NAND と換算できるので、この値を倍にした数が、ゲート数と見なせる。約 10 万ゲートに収まる回路規模である。

表 3: 論理合成後の回路規模

RAM モジュールなし	43487 BC
RAM モジュール	4680 BC
合計	48,167 BC

配置配線後のクリティカルパスの最大遅延を表 4 に示した。これよりこの回路の最大動作周波数を求めると、約 50Mhz 程になる。

表 4: 配置配線後のクリティカルパスの最大遅延

	立ち上がり時	立ち下がり時
最大遅延時間	9.7148 ns	9.6693 ns

配置配線後のチップのレイアウトを図 10 に示した。

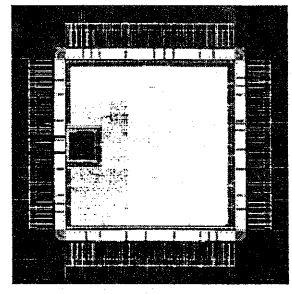


図 10: R-Clos switch chip レイアウト図

5 MGF switch architecture の評価

今回実装した MGF switch chip for R-Clos II の性能評価を Verilog-HDL に基づくシミュレーションによって行った。比較のために、通常のリンクあたり単一の packet queue をもつ single channel のスイッチ (以下、SINGLE と表記) についても同様の評価条件で評価した。

5.1 評価条件

シミュレーションの条件は、以下のとおりである。

- 対象は、スイッチ 1 個 (つまりクロスバ)
- 生成したパケットの行き先のポートはランダムに決定
- パケットフォーマットもランダムに決定
- パケット長は 4flit で固定
- 設定したアクセス発行率にしたがってパケットを発行

5.2 結果: 転送遅延

図 11に、アクセス発行率と全パケットのの平均転送遅延時間をグラフ化したものを示す。比較のために、単一queueのsingle channelのスイッチの評価も載せている。MGFスイッチアーキテクチャを使用した場合には、SINGLEのスイッチを使用した場合と比較して、遅延時間は最大で60%平均約70%となっている。

また、スケジューリングパケットの遅延時間は、アクセス発行率が変化し、全体の遅延時間が増加しても、ほぼ横ばいに推移していることがわかる。スケジューリングパケットの微少な増加は、出線競合のために生じているものである。実際にASCAシステムで使用される場合には、このパケットはスケジューリングが施されるために、衝突することはなく遅延時間は一定で、最大でSINGLEの遅延の約30%程度に抑えられる。他方で、スケジューリングされていないパケットの遅延時間も、MGFでは転送経路が2つ独立して存在しているためにSINGLE よりも最大で60%程の遅延に抑えられている。

よって、このスイッチアーキテクチャは、優先する必要のあるスケジューリングがなされている近細粒度のパケットは遅延時間がほぼ一定であり、優先されないパケットも効率よく転送されることが示された。

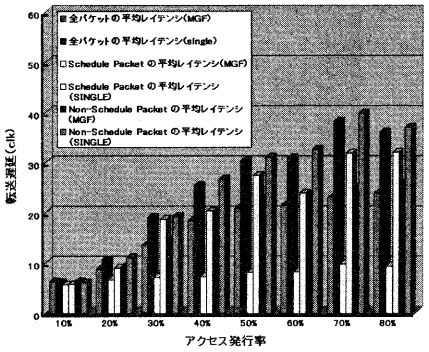


図 11: switch chipの遅延時間

次に、SINGLEのスイッチとMGF switchのハードウェア量の比較を合成後のゲート数によって評価した。この結果を表5 に示した。MGF switchは、一般的な構成の最も単純なスイッチであるSINGLE (single channel single queueing)の約1.4倍程度のハードウェア量で実装することが可能であることがわかった。

表 5: ハードウェア量の比較(合成後ゲート数)

SINGLE	MGF	MGF/SINGLE
34594	48167	1.39

6 結論

本報告では、マルチグレイン並列処理をターゲットとするマルチプロセッサシステムASCAの相互結合網R-Clos IIの要素スイッチをLSIチップに実装した。効率の良い並列処理のために、コンパイラの施すスケジューリングをサポートするMGF switch architectureを提案し、ゲートアレイに実装した。Verilog-HDL によるシミュレーションで転送性能を評価したところ、転送遅延が平均で6割程度に削減されることが示せた。また、スケジューリングを施したパケットがネットワークの負荷によらずほぼ同じ転送遅延で転送できることを示せた。最もプリミティブな単一input queueing方式のスイッチとハードウェア量を比較し、MGF switch architecture の有効性を示した。今後の課題としては、MGF switchを用いてR-Clos IIを構成し、転送性能を評価する予定である。

謝辞

本チップ試作は東京大学大規模集積システム設計教育研究センターを通し、株式会社日立製作所および大日本印刷株式会社の協力のもとに行われたものである。

参考文献

[1] Iwai K., Morimura T., Fujiwara T., Sakamoto K., Kawaguti T., Kimura K., Amano H., Kasahara H.: "AN INTERCONNECTION NETWORK OF ASCA:A MULTIPROCESSOR FOR MULTI-GRAIN PARALLEL PROCESSING", In*Proc. of IASTED International Conference APPLIED INFORMATICS AI'98*, Feb 1998.

[2] Morimura T., Tanaka K., Iwai K., Amano H.: "Multi-stage Interconnection network Recursive-Clos(R-Clos) II: a scalable hierarchical network for a compiler directed multiprocessor ASCA", the 2001 Internatinal Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'2001) in appear, Jun 2001.

[3] John L. Hennessy and David A. Patterson: "COMPUTER ARCHITECTURE A QUANTITATIVE APPROACH SECOND EDITION", Morgan Kaufmann Publishers, 1996.

[4] Clos C.: "A study of non-blocking switching networks", Bell system Tech.J. 32, pp.406-424, Mar 1953.

[5] T. Morimura, K. Iwai,H. Amano: "Multistage Interconnection Network Recursive-Clos(R-Clos): Emulating the hierarchical multibus", In *Proc. of the ISCA 11th International Conference of PARALLEL AND DISTRIBUTED COMPUTING SYSTEMS (PDCS-98)*, pp.99-104, Sept. 1998

[6] Okamoto M., Yamasita K., Kasahara H. and Narita S.: "Hierarchical Macro-Dataflow Computation Scheme on a Multiprocessor System OSCAR", In *Proc. IEEE Pacific Rim Conference on Communications, Computers, and Signal Processing*, pp.44-49, May. 1995.

[7] M.Karol, M.Hluchkj, S.Morgan: "Input versus Output Queueing on a Space-Division Packet Switch", IEEE Trans. on Comm. Vol.35, No.12, pp.1347-1356, Dec. 1987

[8] R.Sivaram, D.K.Panda, C.B.Stunkel: *"HIPIQS:A High Performance Switch Architecture using Input Queueing"*. In *Proc of the 12th International Parallel Processing Symposium*, pp. 134-143, April. 1998

[9] Y.Tamir, G.L.Frazier: *"Dynamically-Allocated Multi-Queue Buffers for VLSI Communicatin Switches"*. IEEE Trans. on Comput. Vol.41, No.6, pp.725-737, June. 1996

[10] M.Karol, M. Hluchyj, S.Morgan: *"Pipilined memory shared buffer for VLSI switches"*. In *Proc of ACM SIGCOMM Conference*, pp.39-48, Aug. 1995

[11] C.Partridge: *"Gigabit Networking"*, Addison-Wesley, 1994

[12] C.B.Stunkel,R.Sivaram,L.Frazier: *"The SP2 High-Performance Switch"*. In *Proc of th 24th IEEE/ACM Annual International Symposium on Computer Architecture(ISCA-24)*,pp.50-61,June. 1997