COREswitch におけるマルチキャスト方式

3G－8　小倉 毅　高橋 直久　丸山 充　八木 哲　川野 哲生
NTT 光ネットワークシステム研究所

1　はじめに

インターネット上での同報通信を利用したアプリケーションの普及に伴い、スイッチやルータではマルチキャストのハードウェアサポートが必須となっている。本稿ではCOREswitch におけるマルチキャストの概要、および、その実現のための内部資源アービトレーション方式について説明し、特性および今後の検討項目を述べる。

2　内部データフロー

COREswitch の内部データフローを図1に示す。

通信回線対応のプロセッサ (以下 CIF) は、回線速度が622Mbps、および、156Mbps 用の2種類選択でき、回線MTU は 65280byte[1] である。CIF はフレームの受信ごとにアービトレーションモジュールに内部転送許可リクエストを出し、Ack が返ると内部転送を開始する。リクエスト毎に優先／非優先の2段階の優先度を指定できる。内部転送許可待ちのフレームは、優先／非優先、および、ユニキャスト／マルチキャストに関係なく単一の FIFO に格納され、HOL(Head of Line) フレームのみ内部転送を開始することができる。

マルチキャスト時のフレームのコピーはバックプレーン上のクロスバスイッチ内で行なわれる。全宛先へのフレームが同一のタイミングで送出される One-shot 型 [2] マルチキャストである。図中のアービトレーションモジュールはクロスバスイッチの出力ポートの衝突回避を行なう。

3　アービトレーションモジュール

前述の構成要素のうち、マルチキャストを含めた内部データフローの特性に大きく影響するのがアービトレーションモジュールの動作である。以下に、今回試作したアービトレーションモジュールについて述べ、その特性、および、今後の検討項目について述べる。

3.1　概要

今回の試作では、できるだけシンプルなハードウェア構成とするため、アービトレーションモジュールの主要機能を1チップ FPGA 内 (以下アービタチップ) に実装した。アービタチップの内部ブロック図を2に示す。

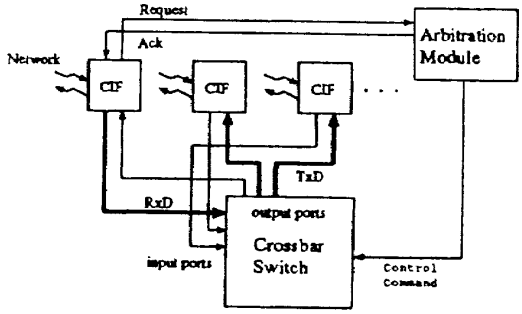

図 1: 内部データフロー

各 CIF に対応する Request Fetch 部は、宛先／優先度などの情報を含む CIF からの転送リクエストをフェッチし、当該 CIF からの要求の発生を示す内部リクエストを発行する。CIF からのリクエストフェッチと内部リクエストの発行はパイプライン的に行なわれる。

Sequencer は、優先度を考慮したプライオリティエンコーダにより上記内部リクエストを一つずつ選択し、出力ポートの衝突チェック、クロスバスイッチの設定、転送許可 Ack のアサートを行なう。出力ポートが使用中の場合、当該 CIF のスロット番号を Reserve FIFO に格納し、リクエストを予約扱いにする (後述)。

Release Request 部は、転送許可 Ack のアサート中に CIF からのリクエストのネゲートを検出すると、データ転送の終了を示すシグナルを発行する。Sequencer 部はこのシグナルを検出すると、当該スイッチ資源を解放し、Ack をネゲートする。このとき、Reserve FIFO 内に予約中のリクエストがあればまとめてリトライする。

Request Fetch 部、および、Release Request 部は全て並列に動作する。Sequencer の動作とも非同期である。

3.2　出力ポートの衝突回避

出力ポートの衝突チェックは、使用中の出力ポートを示す USE TABLE、アービトレーションに失敗した全リクエストの要求ポートを示す RSV TABLE、および、両者のビット毎の OR を保持する OR TABLE を用いておこなう。リクエストの出力先ポートと OR TABLE の内容をビット毎に比較し、重複がなければ転送許可し (図3)、その宛先ポートが USE TABLE に加えられ、OR TABLE の内容が更新される。許可されなかったときはその宛先が RSV TABLE に加えられ、OR TABLE の内容が更新される。RSV TABLE の導入によりポートの獲得に失敗したリクエストの要求が予約され、後続リクエストによってブロックされ続けるのを防ぐことができる。

Frame multicast mechanism for COREswitch.
Tsuyoshi OGURA, Naohisa TAKAHASHI, Mitsuru MARUYAMA, Satoru YAGI, and Tetsuo KAWANO
NTT Optical Network Systems Laboratories

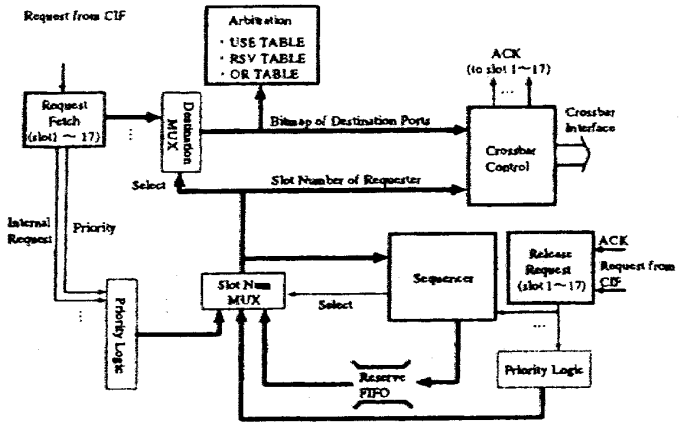

図 2: アービタチップの内部ブロック

3.3 考察

今回試作したアービトレーションモジュールによって得られる内部データフローの特性について、以下の観点から評価／考察する。

フレーム処理能力　アービタチップの動作クロック周波数は40MHz である。クロックサイクル数から算出したアービタチップの各種処理時間を表 1に示す。(1) は Request Fetch 部が CIF から出力先ポート、優先度などの情報を含んだ転送リクエストのフェッチを完全に終了するまでの時間である。(2) は Sequencer 部が 1 つの内部リクエストを選択し、出力ポートの衝突チェック、クロスバスイッチの設定、転送許可 Ack のアサートを行ない次の処理に移るまでの時間である (出力ポート獲得成功時)。(3)、(4) はそれぞれデータ転送終了検出時のスイッチ解放処理、予約扱いとなったリクエストのリトライにかかる時間である。(2) ～ (4) は出力先ポート数によって処理時間が異なる。なお、(1), (2) は一部パイプライン的に処理が行なわれ、CIF からみたときのリクエスト発行から Ack を受けとるまでの最小時間は 525ns となる。

出力ポートの衝突によるスループット低下が少なくアービタの性能が内部転送スループットの支配項となるユニキャスト転送の場合を考える。Request Fetch 部の処理 (1) は、Sequencer 部の処理 (2) と非同期に行なわれるので、出力ポートの衝突がなく各リクエストについて転送許可／スイッチ解放処理だけを行なう場合、フレーム処理能力 (frame per second) は以下のようになる。

$$1sec/(500ns + 375ns) \approx 1.1Mfps$$

表 1: アービタチップの各種処理時間 (40MHz 動作時)

(1) CIF からのリクエストフェッチ	150ns
(2) 通常の転送リクエスト処理	500 ～ 725ns
(3) 解放リクエスト処理	375 ～ 600ns
(4) 予約リクエストのリトライ	500 ～ 725ns

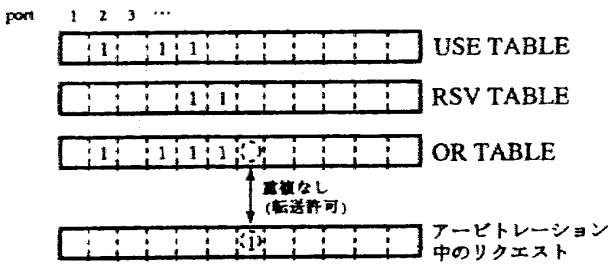

図 3: 出力ポートのアービトレーション

この数字の妥当性は、回線速度とスイッチに到着するフレームサイズの分布に依存する。今後、実環境での測定をもとにこの点を検証する。

内部転送遅延　Sequencer 部が処理する内部リクエストの選択は、実装上の制約から単純なプライオリティエンコーダで実現した。したがって、この部分で低優先度リクエストが高優先度リクエストにブロックされ続ける可能性がある。ラウンドロビン方式などによる解決が考えられるが、実装コストや処理オーバヘッドとのトレードオフであり、今後必要性を検討する。

Sequencer 部による処理が開始されてから、出力ポートの獲得に成功し転送許可を得るまでの時間は、出力ポートの予約機能により有限値に抑えることができる。全ての出力ポートの獲得が必要で他の転送要求にブロックされやすいマルチキャスト転送について特に有効である。

優先度　本方式における優先度とは、複数のリクエストが存在するときの Sequencer 部の処理順序に対するものである。今後、本方式の有効性を検証し、本方式以外の優先度サポートの必要性を検討する。

4 おわりに

本稿では、COREswitch におけるマルチキャスト方式、および、その実現のための内部資源アービトレーション方式について説明し、その特性や今後の検討項目について述べた。今後は ASIC 化などによる高速化を検討しており、本方式の検討結果もこれらに反映させていく。

謝辞

今回の試作にご協力頂いた小林正之氏、ならびに、日頃から有益な助言を頂く吉田敏明氏に感謝いたします。

参考文献

[1] K. Murakami, M. Maruyama, "IPv4 over MAPOS Version1", RFC2176, June. 1997.

[2] Xing Chen, Jeremiah F. Hayes, "Call scheduling in multicasting packet switching", in *Proc. ICC*, 1992, pp.895-899.