高性能スイッチングアーキテクチャの一考察

2H－3　High Performance Switching Architecture

田中 敦
Atsushi Tanaka
NEC エンジニアリング コンピュータ事業部
Computer Division, NEC Engineering,Ltd.

1 はじめに

高性能なスイッチングハブやルータ装置を実現する，パケットスイッチ・ファブリックのアーキテクチャを考案したので紹介する。スイッチングの基本方式として共有バッファ方式を採用した(図 1)。このファブリックは，ネットワーク側のインタフェースに複数の IO ポートバスを持ち，ここから受信したパケットデータをすべて共有バッファに格納する。その後，各送信チャネルの出力キューに従って読み出され，IO ポートバスに送出することでスイッチ動作を実現する。特にマルチキャスト・パケットにおいては，バッファ上でのパケットデータの複写を必要としない点，性能面で有利な方式といえる。キューを管理するメモリは共有バッファと分離させることで，データ転送の処理性能とキュー管理性能は分離して考える。共有バッファメモリおよびキューメモリは，汎用の高速メモリである。半二重のデータ処理レートで 32Gbps，パケットのフォワーディングレートで24Mppsを目標性能とした。これは昨今注目されているギガビットイーサネット[1]のワイヤスピードで 16 回線分に相当する。

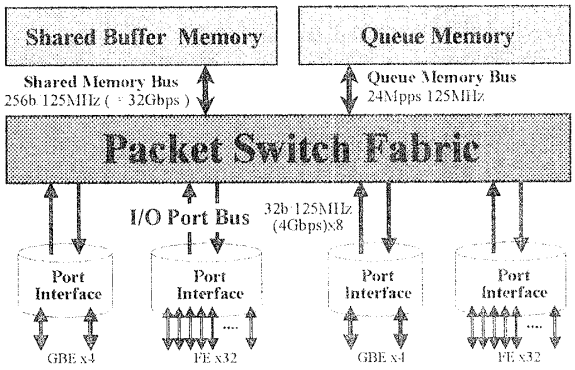

図1　共有バッファ方式

2 課題

高性能なファブリックをハードウエア(LSI)で実現することに対する課題を示す。

2.1 パケットデータをスイッチ処理する帯域の確保

帯域はデータを並列処理するビット幅と処理周波数で決定する。メモリ構成のコストを考慮すれば，ビット幅はできるだけ小さくしたい。表1にビット幅と周波数の関係から計算したバス性能を示す。32Gbps の目標性能には少なくとも 256 ビット幅以上のデータを 125MHz で処理する必要がある。入出力のピン数がネックである。

表1　共有メモリバスの性能

Data-Width(bit)	Half-Duplex-Rate(Gbps)				
	66.67MHz	80MHz	100MHz	125MHz	133MHz
32	2.13	2.56	3.20	4.00	4.26
64	4.27	5.12	6.40	8.00	8.51
128	8.53	10.24	12.80	16.00	17.02
256	17.07	20.48	25.60	32.00	34.05
512	34.14	40.96	51.20	64.00	68.10

2.2 パケットのキュー管理性能とゲート規模

キュー管理の実現手段として，機能実現のリソースに着目して，2つの方式に分けられる。キューの実体はメモリ上で構成し，その制御に必要なヘッドおよびテールを示すポインタは LSI 内部のゲートで実現する分担aの方式と，ヘッドおよびテールの情報までメモリ上に展開し，LSI ではベースのポインタとチャネル番号で示すオフセットの管理のみ行う分担bの方式である。

図2で相対的な比較を示すように，分担 a の方式は，キューのポインタを知るためのメモリアクセスが不要であるため，分担 b と比較して，スイッチ性能を向上できる。24Mpps の性能を達成するためには，分担 a の方式は必須と考える。反面，回路のゲート規模は大きくなる。目標性能をファーストイーサネットで換算すると 128 ポート以上で，QoS 制御のためのレベルキューを 8 レベルとすれば，128×8=1024 チャネル以上の装置となる。分担aの方式による回路規模は約 5M ゲート程度と予測でき，キュー管理機能を専用チップで実現するとしても，ゲート規模がネックとなる。

3 アーキテクチャ

先の課題を解決し目標性能を達成するために，考案した方策について述べる。

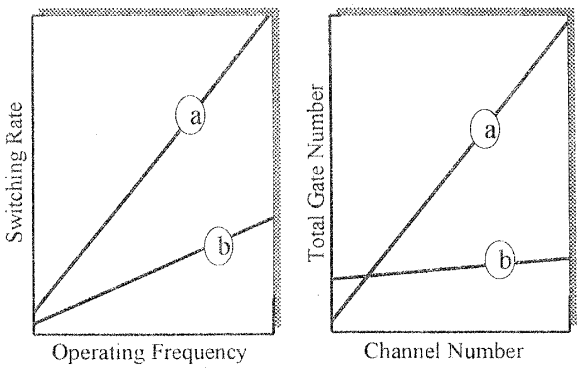

図2　キュー実現方式の比較

3.1 ビットスライスによるピンネックの解消

処理するパケットデータのビット幅をビットスライスし，ファブリックを複数の LSI に分割した。ビットスライスロジック(bit-slice-logic)とは，最も近世代のビルディングブロックと呼ばれ，CPU の ALU やレジスタのスライス技術として有名である[2]。図1の共有メモリバスおよび I/O ポートバスにこのビットスライスの技術を適用して，ファブリックを機能単位の分割ではなく，性能単位の分割にした。今回は，4チップのビルディング構成にし，256 ビットの共有メモリバスは 64 ビット単位に，32 ビットの I/O ポートバスはそれぞれ 8 ビット単位にビットスライス接続にした。ピンネック問題を解消し，32Gbps の装置ソリューションは4チップ構成で，8Gbps の装置ソリューションは1チップ構成で実現できる。

3.2 キューの分割方式

共有バッファ方式におけるキュー管理のモデルを図3に示す。主に，受信パケットに割り当てる共有バッファの空きアドレスを管理する A キューと，パケットの送信を順次制御する出力キューの B キューからなる。先に説明した分担 a のキュー管理方式を想定するが，本モデルでは便宜上キューメモリの存在も含めてチップと表現する。パケットを受信すると A キューからバッファアドレスを1つデキューし，パケットはバッファに格納する。またバッファアドレスは送信先のチャネルに該当する B キューにエンキューされ，送信の順序待ちとなる。次に，パケットの送信が完了すれば，B キューからバッファアドレスをデキューし，A キューへ返送され，バッファは解放される。B キューはサポートするチャネル数だけ存在する。

ゲート規模の分割と並列処理による性能向上を目的に，キューを分割する案に対して解説する。 まず分割案1(図3)は，単純に全キューを均等分割する方式である。この場合，他チップの A キューから自チップの B キュー宛および自チップの A キューから他チップの B キュー宛エンキューパスと，他チップのBキューから自チップのAキュー宛および自チップのBキューから他チップの A キュー宛デキューパスが新たに必要で，分割チップ数が多くなると，対数的に接続が増加する。

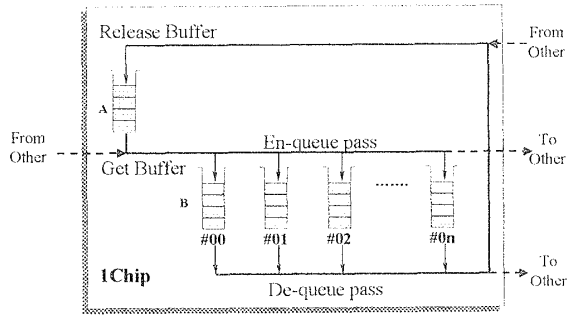

図3 共有バッファ方式のキュー管理モデル(分割案1)

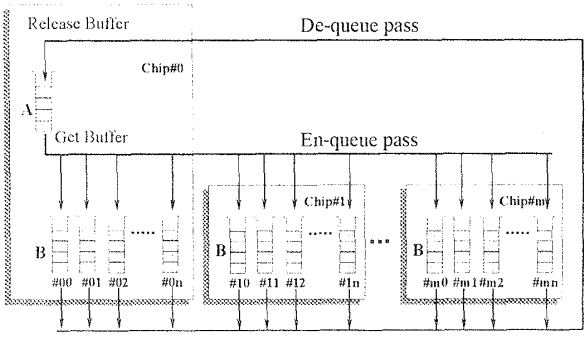

図4 キューの分割案2

また，Aキューを分割したことは，共有メモリの容量をチップ単位に分割したと同じで，全受信ポートで全メモリを共有できなくなり，本来の共有バッファ方式の効果が減少する問題もあり，ベストの解とはいえない

今回提案する分割案2(図4)は，Aキューをシステム内で1つ存在させ，B キューをチップ単位に分割する。A キューから B キュー宛のエンキューパスと，B キューから A キュー宛のデキューパスは，チップ間にまたがるマルチドロップ型のバス構造にする。B キューのエンキュー動作は，共有バッファにパケットを書き込むタイミングに同じで，デキュー動作はパケットを読み出すタイミングに同じである。つまり，このエンキューパスとデキューパスは，完全にタイムスロットが排他であるため，物理的に同一のバスとする事が可能である。また，共有バッファ・メモリのアドレスのラインと論理的に同一でもあり，特別なピンの追加も伴わない。

分割案2では，A キューのみ一元管理することで完全な共有バッファ方式を実現できる。かつ4チップのビルディング構成にも対応した性能単位の分割で，チップあたりに要求される性能は，6Mpps にまで下げられる。また，先に説明した分担 a のキュー管理方式を採用しても，チップあたりのゲート規模は 1.3M ゲート程度と予測でき，現在のゲートアレイ技術で実現可能な範囲となる。

4 まとめ

本稿では，同じ種類の LSI チップを複数使用し，ビルディング構成により高性能な共有バッファ方式のスイッチファブリックを構築できる，データパスの接続方法と，キューの分割方式について述べた。今後もネットワークに対する社会の要求に応えられるよう研究を継続し，ネットワークの発展に貢献する製品の開発に取り組む所存である。本研究にあたりご指導いただいた NEC・LAN 事業部の関係各位に深く感謝の意を表したい。

【参考文献】

[1]IEEE Draft P802.3z/D3.2
[2]ビットスライスLSIとその応用／Glenfordj.Myers 著／奥川峻史 訳補／共立出版