

## Arquitetura de Comutador ATM usando o Roteador *RCube*

William F. Giozza  
Grupo de Redes de Computadores  
DSC/CCT/UFPb Campus II  
Campina Grande Pb  
e-mail: giozza@dsc.ufpb.br

Belkacem Zerrouk  
Laboratoire MASI/CAO&VLSI  
Université de Paris 6  
Paris França  
e-mail: zerrouk@masi.ibp.fr

### Resumo

Neste trabalho é proposta uma arquitetura de comutador para redes ATM usando como matriz elementar de comutação um *chipset* composto fundamentalmente pelo roteador reconfigurável de uso geral *RCube* e por um circuito ASIC *companion* implementando as funções específicas de comutação ATM. Em particular, o trabalho apresenta a especificação funcional do elemento básico de comutação ATM e discute os compromissos para sua implementação.

### Abstract

*In this work is proposed an ATM switch architecture based on a chipset mainly composed by the general-purpose reconfigurable router RCube and a companion ASIC implementing specific functions of ATM switching. In particular, it presents the functional specification and the implementation tradeoffs of the ATM basic switching element proposed.*

### 1. Introdução

O crescente desenvolvimento de redes ATM (*Asynchronous Transfer Mode*) tem-se baseado na existência de um *framework* tecnológico favorável formado principalmente pela potencialidade de transmissão das fibras ópticas e pelas facilidades de integração VLSI proporcionada pelos avanços da microeletrônica [1,2]. Maior confiabilidade, melhor vazão e menor atraso, são requisitos básicos para as redes com integração de serviços de transmissão de dados, voz, imagem e vídeo. Portanto, protocolos e mecanismos de controle mais rápidos e eficientes constituem um imperativo na implementação dessas redes. A técnica ATM, concebida inicialmente para atender adequadamente às necessidades de integração de serviços nas redes públicas, tem hoje em dia uma vasta potencialidade de aplicações em redes locais [3]. Se no caso das redes locais, o uso de fibras ópticas não é obrigatório dada a boa confiabilidade dos suportes físicos tradicionais em pequenas distâncias, continua ainda imperativa a integração VLSI dos protocolos e mecanismos de controle de tráfego para atender às exigências de velocidade e atraso das aplicações de tráfego integrado.

A literatura técnica existente relata vários desenvolvimentos experimentais de *chip* ou *chipset* implementando comutadores ATM para RDSI-FL (essencialmente nos laboratórios ligados às grandes empresas de telecomunicações), alguns deles inclusive usados como componentes de equipamentos já comerciais [4,5,6]. No caso de comutadores para redes locais as informações são bastante escassas, em parte por ser um mercado desenvolvido mais recentemente, mas certamente também por refletir o valor estratégico desse *knowhow* face ao acesso de empresas de menor porte às facilidades de projeto de circuitos integrados ASIC. Ao contrário do caso de *Ethernet* (protocolo proprietário), é pouco provável que tenhamos *chips* ATM de prateleira. A padronização básica de ATM não é suficiente para garantir uniformidade mínima de implementação, deixando ao implementador uma boa margem de criatividade para atender funcionalidades internas de seu comutador (gerência de recursos, diagnósticos, etc).

Embora tenha uma arquitetura caracterizada pela escalabilidade dos seus componentes básicos, o comutador ATM proposto neste trabalho é orientado principalmente para um mercado de redes locais ATM de pequeno porte (número de portas reduzido). Foi concebido fundamentalmente em torno de um *chipset* composto pelo *RCube*, um circuito integrado com complexidade da ordem de um milhão de transistores desenvolvido pelo Laboratório MASI em cooperação com a empresa Bull, na França, dentro de um projeto europeu ESPRIT [7], e por um circuito ASIC (*Application Specific Integrated Circuit*) de complexidade intermediária. O circuito *RCube* implementa uma rede de interconexão de uso geral e o circuito ASIC *companion* implementa as funções específicas de comutação ATM.

Na próxima seção serão apresentados os princípios da comutação ATM de modo a situar a terminologia e as escolhas tecnológicas adotadas pelo sistema proposto. A seção 3 apresentará as principais características do *RCube*, componente fundamental para o sistema comutador proposto. Na seção 4 será apresentada a arquitetura geral do comutador ATM e descrita em detalhes a concepção funcional do circuito ASIC que juntamente com o *RCube* implementa a matriz elementar de comutação ATM. Por último, a seção 5 discutirá compromissos de implementação e apresentará algumas conclusões.

## 2. Tecnologia de Comutação ATM

A tecnologia de comutação ATM está associada a um modelo de arquitetura de rede desenvolvido pelo ITU-T (ex-CCITT) para a futura Rede Digital de Serviços Integrados de Faixa Larga (RDSI-FL) [2]. Esse modelo hierárquico, ilustrado na Figura 2.1, diz respeito principalmente ao plano Usuário (suporte à comunicação usuário-usuário) e ao plano Controle (suporte à sinalização), e é composto por:

- camada ATM, responsável basicamente pela comutação de células com conexões em dois níveis: VCs (*Virtual Channel*) e VPs (*Virtual Path*) que são agregados de VCs;
- camada Física, responsável pelo condicionamento e transporte das células através dos meios físicos de transmissão;
- camada de Adaptação a ATM (AAL), para adaptar equipamentos terminais não-ATM para o uso da rede de transporte de células ATM.

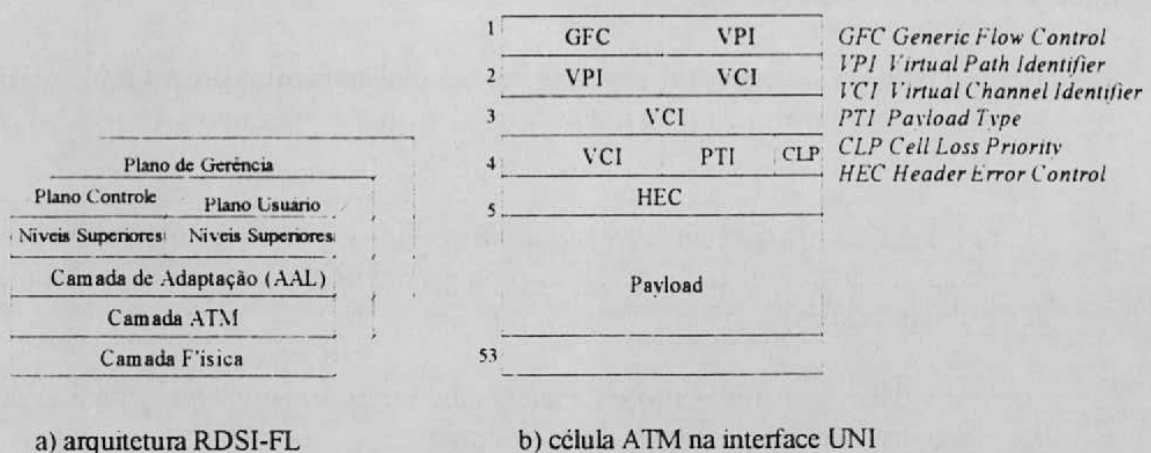


Figura 2.1: Arquitetura e célula ATM

A célula ATM tem 53 octetos (Fig. 2.1b), incluindo um cabeçalho de 5 octetos contendo as seguintes funcionalidades: identificadores de VPs e VCs, tipo de célula, prioridade da célula, controle de erros do cabeçalho e, no caso da interface UNI (*User-Network Interface*), 4 bits para controle de fluxo restrito. ATM é orientado a conexão; os valores de VPI/VCI, qualidade de serviço e outras informações pertinentes são estabelecidos quando do estabelecimento da conexão (fase de *call setup*).

Um comutador ATM tem por função básica a transferência de células das suas entradas para as suas saídas. Uma arquitetura geral típica de comutador pode ser como mostrado na Figura 2.2, onde destacam-se três componentes principais: a matriz de

comutação, os controladores associados às portas de entrada/saída e um processador de controle [8].

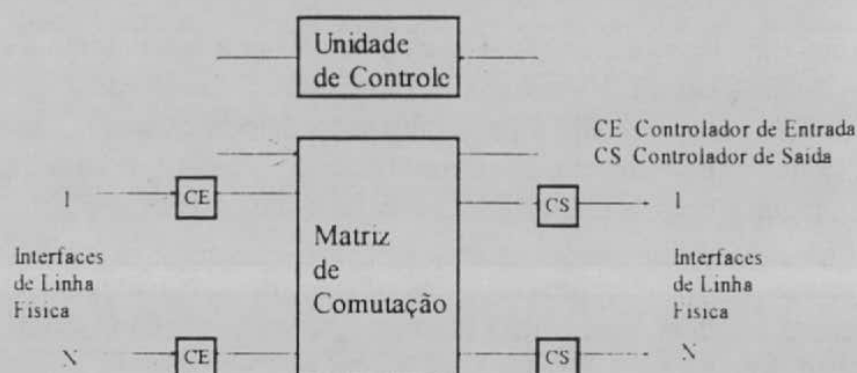


Figura 2.2: Arquitetura geral de comutador ATM.

No caso de ATM, a heterogeneidade de serviços e a incerteza quanto à carga do tráfego suportado (banda passante, duração de chamada, modo de conexão, qualidade de serviço, etc) implicam nos seguintes requisitos básicos na concepção de sistemas de comutação:

- capacidade de receber, processar e reencaminhar **células de tamanho fixo** e cabeçalho com funcionalidade reduzida, a velocidades da ordem de 150 Mbps ou superiores;

- capacidade de armazenamento (*queueing*) temporário de células para tratar a possibilidade de contenção nas saídas (compromisso entre atraso e probabilidade de perda de células);

- conectividade ponto-a-ponto e capacidade de difusão (*multicast*) para suporte, por exemplo, de serviços de vídeo distributivo.

- desempenho caracterizado por uma **alta vazão e disponibilidade**, o que implica por exemplo, em baixa probabilidade de bloqueio das conexões, pequeno atraso de comutação, variação do atraso limitada e baixa probabilidade de perda de células

Num comutador ATM, o roteamento de células é realizado tipicamente por uma matriz de comutação (*switching fabric*) geralmente construída a partir de uma matriz elementar de comutação. Existem vários tipos de matrizes com diferentes topologias e características de desempenho [8]. Por ser o componente hardware mais importante (desempenho, capacidade, custos, etc) a matriz de comutação é uma das principais escolhas tecnológicas na concepção do comutador. Vários tamanhos de matriz elementar de comutação são encontrados na literatura, por exemplo, 2 x 2, 8 x 8 e 16 x 16.

A tradução (comutação) da informação de cabeçalho é implementada por uma unidade de controle responsável pela leitura dos cabeçalhos das células entrantes no comutador (identificação de conexão VPI/VCI, bit de prioridade CLP, tipo de célula PTI, etc) e pelo seu encaminhamento interno a partir das características de cada conexão previamente armazenadas numa tabela de rotas (porta de saída, VPI/VCI na saída, qualidade de serviço, etc).

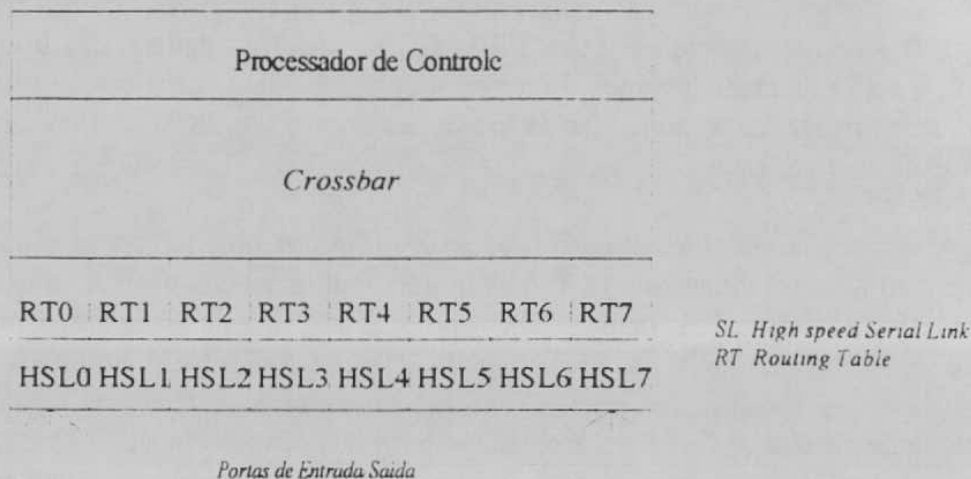
A estratégia de enfileiramento (*queueing*) tem por objetivo tratar a possibilidade de contenção (congestionamento) provocada pela multiplexação estatística característica do ATM [8,9,10]. Duas ou mais células endereçadas simultaneamente para a mesma porta de saída causam uma situação de contenção mesmo sem bloqueio interno por parte do comutador. A localização (interna, externa na entrada, externa na saída, etc) e o dimensionamento dos buffers para armazenamento das células em situação de contenção (tamanho de buffer, velocidade de acesso, esquema de prioridades, etc) têm implicações diretas na qualidade de serviço oferecida, sendo portanto uma outra escolha tecnológica importante na concepção de comutadores ATM.

### 3. *RCube*

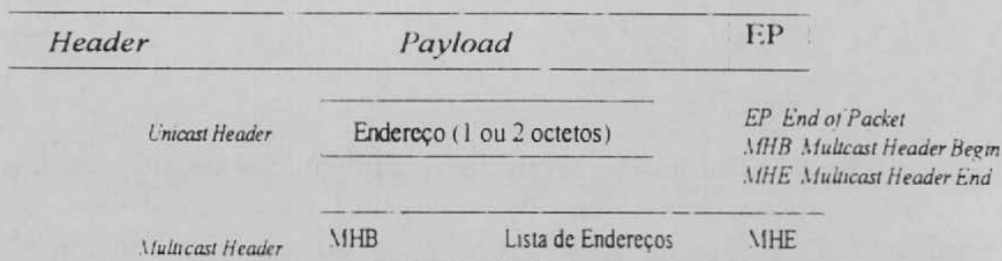
O *RCube* é um dispositivo roteador reconfigurável de alta velocidade de uso geral em redes de interconexão, desenvolvido pelo Laboratório MASI/CAO&VLSI da Universidade de Paris 6 na França [7]. Tem complexidade equivalente a um milhão de transistores e foi inteiramente concebido usando tecnologia CMOS (relógio interno a 100 MHz) e uma tecnologia de ligações seriais a Gbps desenvolvida pela Bull, Les Clayes-sous-Bois, França [11,12]. Baseia-se na técnica de roteamento *wormhole* que permite um roteamento mais rápido bem como mensagens de comprimento qualquer e variável. Usa o esquema de roteamento de etiquetagem por intervalos (*interval labelling*) que permite suportar qualquer topologia de rede com tabelas de roteamento relativamente compactas [13,14,15]. Suporta comunicação *multicast* e é reconfigurável dinamicamente, provendo, por exemplo, facilidades de particionamento, roteamento adaptativo e endereçamento hierárquico [7].

A arquitetura de *RCube*, esquematicamente mostrada na Figura 3.1, inclui 8 portas seriais bidirecionais a alta velocidade (1,2 Gbps cada), uma matriz de interconexão *crossbar*, um processador de controle integrado e 8 tabelas de roteamento compactas e independentes (uma RT por porta)[7].

## a) Arquitetura



## b) Formatos de mensagem

Figura 3.1: Arquitetura do *RCube*.

Uma mensagem constitui uma unidade de comunicação de alto nível entre nodos interconectados pelo *RCube*. De acordo com o roteamento *wormhole*, uma mensagem pode ser subdividida numa sequência de pacotes de diferentes comprimentos que são roteados em modo *pipeline* como um conjunto sucessivo de unidades de transmissão (*flit* = 32 octetos). O mecanismo de controle de fluxo adotado é compatível com o mecanismo usado nas ligações seriais a Gbps [11]. Dois formatos de pacotes são previstos (Fig. 3.1b): pacotes *unicast*, de comprimento qualquer com destinatário único, e pacotes *multicast* (tamanho limitado a 96 octetos) endereçados a vários destinatários (máximo 8) [7]. O processador de controle integrado possui um jogo de instruções específico para inicialização/configuração e diagnóstico (erros e status) do roteador. As facilidades de particionamento da rede de interconexão permitem, por exemplo, agrupar ligações para aumentar a banda passante (vazão) ou ainda separar tráfegos em redes de interconexão com múltiplos serviços. O modo de endereçamento hierárquico implementado pelo *RCube* permite superar as limitações de espaço de endereçamento inerentes ao esquema de etiquetagem por intervalos e prover a necessária escalabilidade [7]. Além disso, o *RCube* suporta roteamento adaptativo, útil por exemplo, para aumento de banda passante, balanceamento de carga ou tolerância a falhas.

## 4. Elemento Básico de Comutação ATM

### 4.1 Arquitetura Geral do Comutador ATM

A arquitetura geral proposta para o comutador ATM usando o roteador de pacotes reconfigurável *RCube* é ilustrada na Figura 4.1, considerando-se como exemplo um comutador com 4 portas. A matriz de comutação ATM é composta por circuito ASIC implementando os elementos básicos de comutação ATM (ABSE) e pela rede de interconexão *RCube*. As interfaces de linha física (ILF) são elementos externos à matriz, responsáveis pelas funções de adaptação com o meio físico de transmissão (protocolos de nível físico, tais como por exemplo, a codificação 8B/10B, 4B/5B, SONET, etc). O roteamento dinâmico de células é implementado através de um sistema multibanco de memória comandado pelo processador controlador central do comutador ATM. A comunicação do controlador com os circuitos ABSE pode ser realizada através do barramento do sistema ou através da própria rede de interconexão *RCube* de modo a usufruir da sua alta vazão (Gbps).

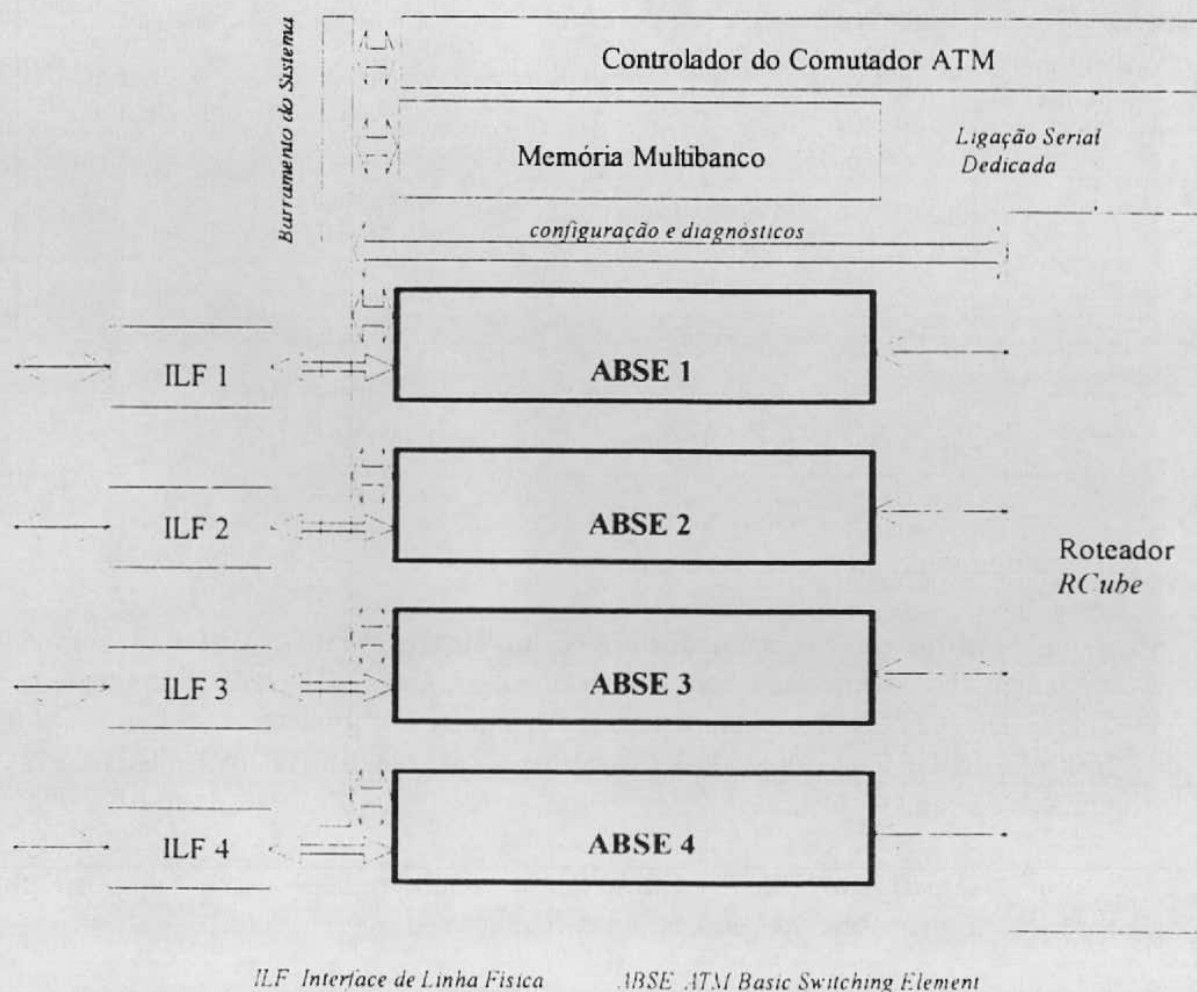


Figura 4.1: Arquitetura de Comutador ATM usando o *RCube*

#### 4.2 Elemento Básico de Comutação ATM (ABSE)

O Elemento Básico de Comutação ATM (ABSE) cujo diagrama em blocos é mostrado na Figura 4.2, é o circuito ASIC *companion* do *RCube* na composição da matriz elementar de comutação ATM. O circuito ABSE é responsável pela implementação das funções de comutação ATM, complementares às funções de roteamento implementadas pelo *RCube*, tais como a multiplexação/demultiplexação, endereçamento e roteamento de células. O circuito ABSE pode também incluir funções adicionais de nível físico já normalizadas tais como, a delimitação de células e o controle de erros no cabeçalho das células (HEC).

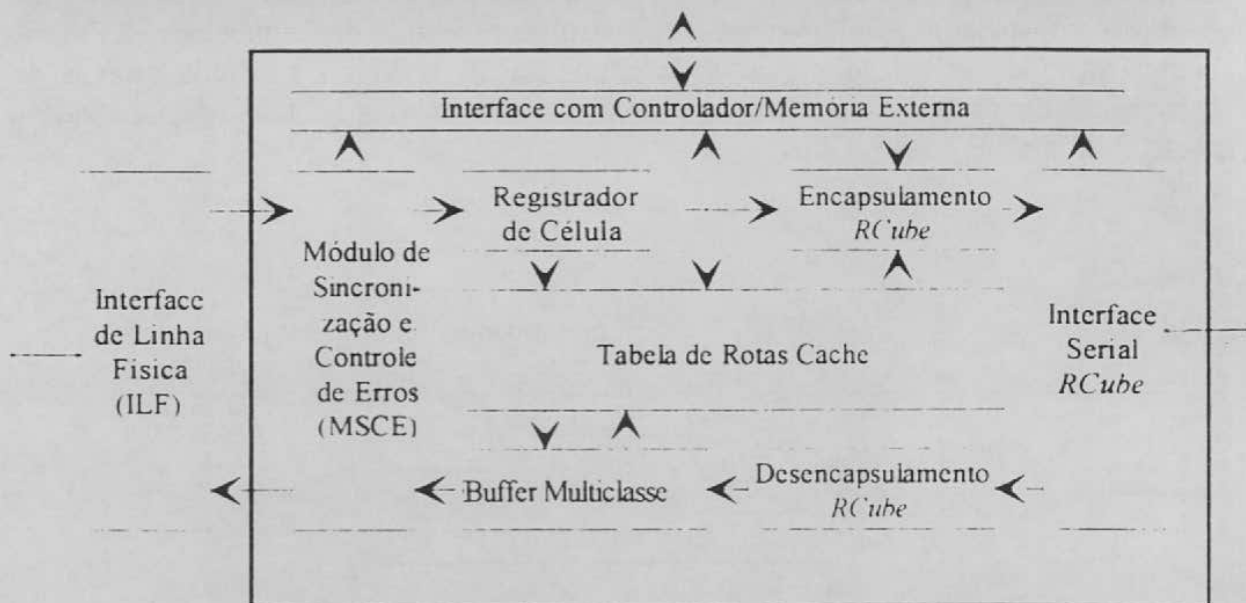


Figura 4.2: Elemento Básico de Comutação ATM (ABSE)

A arquitetura do circuito ABSE inclui:

- um **Módulo de Sincronização e Controle de Erros (MSCE)** opcional, implementando a interface com a Interface de Linha Física (ILF) e algumas das funções da camada Física tais como a delimitação/formatação de células e o controle do HEC; módulo desnecessário no caso de ILF já implementar essas funções da camada Física;

- um **Registrador de Célula** para sincronização dos relógios de recepção/transmissão e de processamento interno;

- um bloco chamado **Encapsulamento *RCube*** servindo para adaptar as células para o roteamento através da rede de interconexão *RCube*;



- . uma **Interface Serial RCube** implementando o protocolo serial do *RCube* a Gbps;
- . um bloco chamado **Desencapsulamento RCube** servindo a recuperar as células transportadas pela rede de interconexão;
- . um **Buffer Multiclasse** implementando uma estratégia de buferização multiclasse de serviços na saída da rede de interconexão;
- . uma **Interface com o Controlador/Memória Externa** disciplinando a comunicação entre o ABSE e o conjunto controlador/memória externa (tradução de VPI/VCI, diagnósticos, bit CLP, etc);
- . enfim, uma **Tabela de Rotas Cache** para aceleração do processo de comutação e roteamento.

A matriz de comutação formada pelo circuito ABSE e pelo *RCube* caracteriza-se pelas seguintes funcionalidades:

- . autoroteamento (*self routing*);
- . buferização na saída;
- . buffers multiprioridades segmentados;
- . *multicast* até 256;
- . vazão adaptativa (p. ex., ligações seriais a 1,2 Gbps + 1,2 Gbps);
- . adaptatividade física (155 Mbps/620 Mbps SONET, 155 Mbps 8B/10B, etc);
- . facilidades de diagnóstico;
- . escalabilidade; etc.

A seguir serão detalhados cada um dos blocos componentes do circuito ABSE.

#### 4.2.1 Módulo de Sincronização e Controle de Erros (MSCE)

O Módulo de Sincronização e Controle de Erros (MSCE), mostrado na Figura 4.3, foi concebido inicialmente para prover uma interface bidirecional ao nível da camada Física entre o circuito ABSE e o circuito ILF externo bem como implementar algumas funções da camada Física (Fig. 2.1), tais como formatação/delimitação de células, controle de erros no cabeçalho (HEC), etc. Os bytes (8 ou 16 bits por exemplo) recebidos da ILF são serializados e decodificados sequencialmente a cada grupo de 53 octetos. A delimitação das células é feita a partir de dois mecanismos básicos: a correlação do HEC com o cabeçalho da célula e o embaralhamento (*scrambling*) do campo de informação da célula segundo especificação padronizada pelo ITU-T (ex-CCITT) [16]. As células válidas assim recuperadas são entregues em paralelo para o Registrador de Célula para o processamento dos cabeçalhos. Do outro lado, as células provenientes do Buffer Multiclasse são primeiramente serializadas para cálculo do HEC

e embaralhamento, e posteriormente entregues ao circuito ILF na forma de bytes sequenciais.

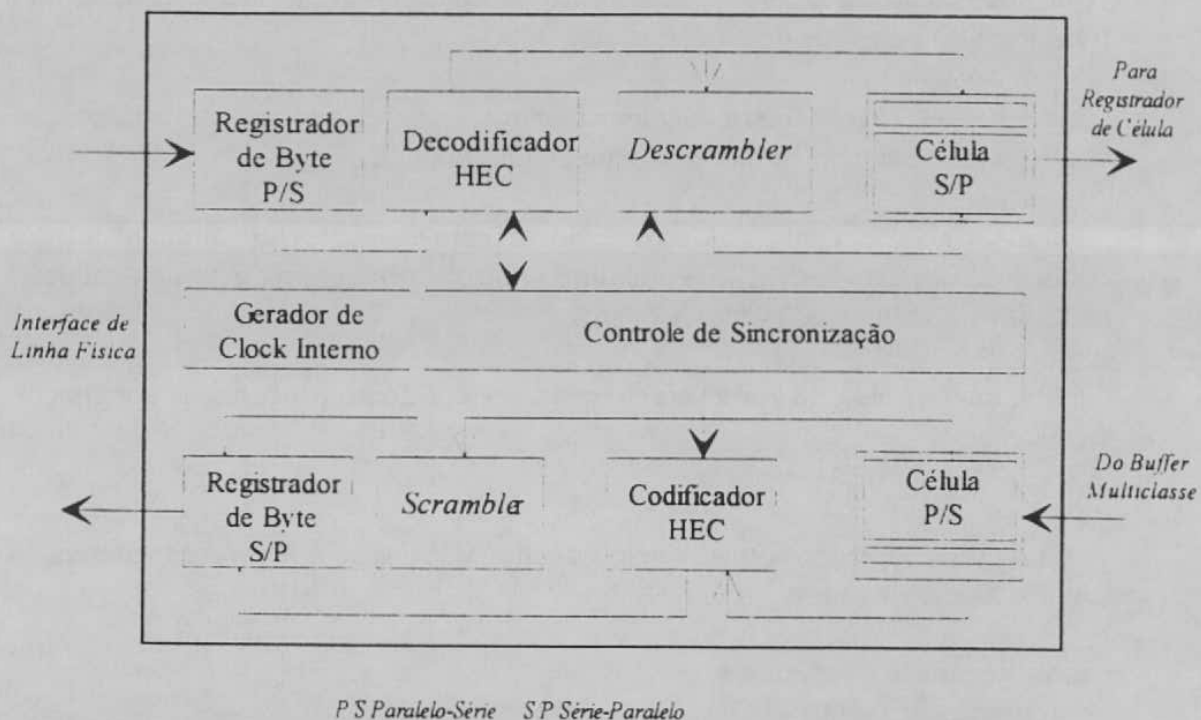


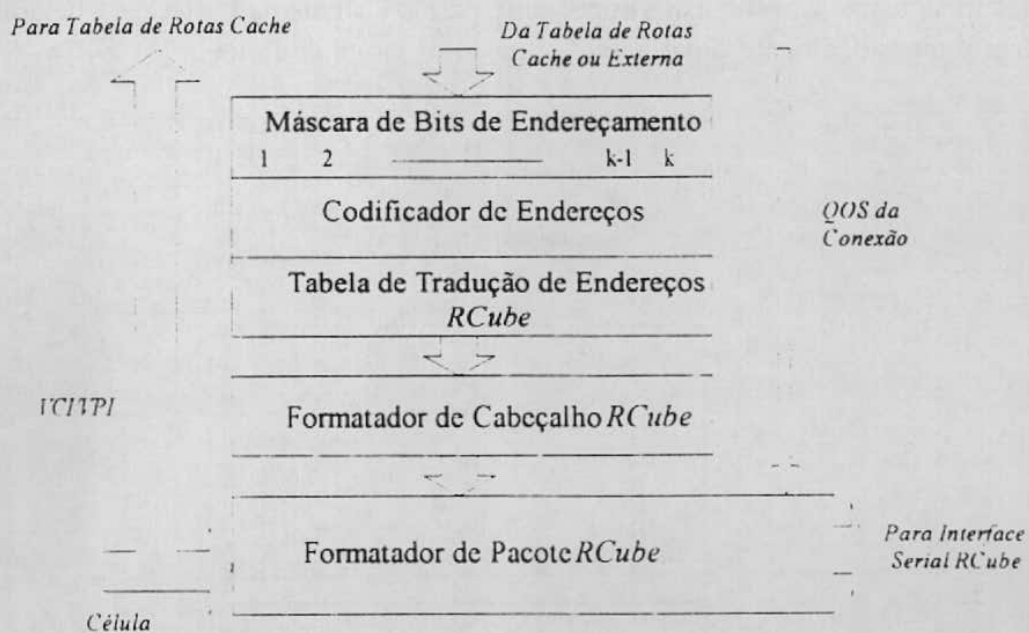
Figura 4.3: Módulo de Sincronização e Controle de Erros (MSCE)

A implementação do MSCE como bloco integrante do ABSE só faz sentido se as funções realizadas por ele não estiverem sendo implementadas pelo ILF. Seria o caso por exemplo de um circuito de interface física tipo SONET [1] de uso geral onde o *payload* de um quadro SONET pode transportar outros formatos de mensagens diferentes das células ATM. O processamento serial de células integrado no ABSE pode se constituir numa importante limitação em termos de velocidade das portas do comutador. Por exemplo, a integração do MSCE para portas com a velocidade padrão de 620 Mbps fica comprometida pelas limitações em frequência da tecnologia CMOS. Com a consolidação dos padrões de interface física para redes locais ATM espera-se o surgimento de circuitos ILF comerciais dedicados, tornando desnecessária a integração do MSCE.

4.2.2 Encapsulamento/Desencapsulamento *RCube*

As células ATM são roteadas internamente pelo *RCube* na forma de pacotes, segundo um dos seus dois formatos básicos: *unicast* ou *multicast*. O esquema de encapsulamento de células é mostrado na Figura 4.4. A informação de roteamento de cada conexão (VPI/VCI, porta de saída, endereços *multicast*, etc) proveniente da Tabela de Rotas (cache ou externa) é codificada em endereços que compõem o cabeçalho do pacote *RCube*. A célula ATM é encapsulada e transportada no *payload* do pacote *RCube* juntamente com um octeto indicando a qualidade de serviço (*QOS*) associada a cada conexão.

a) diagrama esquemático



b) encapsulamento de células *multicast*

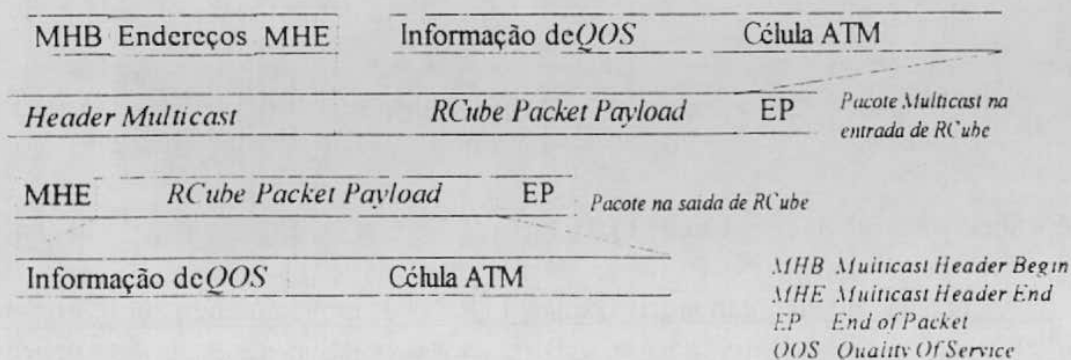


Figura 4.4: Encapsulamento *RCube*

#### 4.2.3 Buffer Multiclasse (BM)

O bloco Buffer Multiclasse (BM), detalhado na Figura 4.5, implementa uma estratégia de buferização nas saídas da matriz de comutação. Esta estratégia, apesar de ser de implementação relativamente mais complexa (tamanho maior), foi escolhida em função de suas qualidades de desempenho e facilidades para suporte de conexões *multicast* quando comparada à estratégia de buferização na entrada. Uma outra característica vantajosa desta estratégia de enfileiramento são as facilidades de suporte a conexões com vários níveis de prioridade o que não ocorre com a estratégia de enfileiramento internamente à rede de interconexão. Um sistema de buffers com múltiplas filas (conforme exemplificado na Figura 4.5) permite atender adequadamente conexões com diferentes prioridades de serviço [17]. Cada porta pode, por exemplo, monitorar sua própria fila e prover informações de carga para suporte ao controle de congestionamento. Um requisito importante para o sistema de buffers é a flexibilidade de seu (re)-dimensionamento dinâmico em função da carga de tráfego.

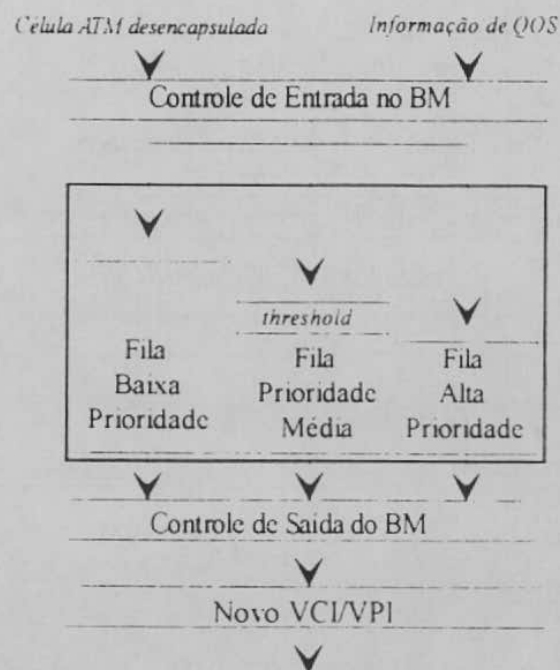


Figura 4.5: Buffer Multiclasse

#### 4.2.4 Tabela de Roteamento Cache (TRC)

A Tabela de Roteamento Cache (TRC) é implementada por um sistema multibanco de memórias integrado no ABSE com o objetivo de acelerar o processo de tradução dos cabeçalhos das células e de controle da qualidade de serviço. De maneira similar à tabela de rotas externa é responsável pelo armazenamento de todas informações úteis para o controle e roteamento de células nas várias conexões ATM estabelecidas. O principal fator limitante do tamanho da tabela cache é a disponibilidade de área

integrável. Memórias com tempo de acesso da ordem de 10 ns permitem atender a vazão necessária.

#### 4.2.5 Interface Serial *RCube*

A Interface Serial *RCube* é uma macrocélula em tecnologia CMOS (0,5 micra, 3,3 volts, 1 mm<sup>2</sup>) [11,12], implementando o protocolo serial ponto-a-ponto bidirecional e os buffers necessários para a comunicação com *RCube*. Os pacotes *RCube* encapsulando células são serializados, decompostos em *flits* para efeito de controle de fluxo e em octetos para efeito de transmissão (código 8B/12B balanceado) a Gbps.

### 5. Conclusões

O comutador ATM proposto caracteriza-se pela simplicidade de implementação em razão da utilização do roteador *RCube*. Beneficia-se de vários homens-ano envolvidos no projeto e desenvolvimento desse circuito que tem sérias chances de se transformar em circuito comercial de catálogo da InMos (parceira do grupo Bull). Outros projetos de centrais de comutação para serviços integrados também tem-se baseado na utilização de um circuito roteador de uso geral, em particular o circuito de catálogo C104 anunciado pela InMos mas ainda não disponível [18]. Os estudos preliminares sobre a realizabilidade do ASIC *companion* a *RCube* indicam uma complexidade bem inferior a *RCube* (da ordem de 300.000 transistores) perfeitamente realizável em tecnologia CMOS com as ferramentas de concepção e testes disponíveis em nossos laboratórios e a disponibilidade de área para integração VLSI experimental em programas institucionais tais como, por exemplo, o PMU (Projeto Multi-Usuário) brasileiro e o CMP francês. O dimensionamento do circuito ASIC depende ainda de estudos relativos à modelagem e avaliação de desempenho envolvendo hipóteses sobre os vários tipos de tráfego e carga a serem suportados pelo comutador ATM. Também está sendo estudado em paralelo a implementação de mecanismos e estratégias para controle de tráfego e congestionamento. Um ambiente de simulação e testes para o sistema comutador deverá ser desenvolvido a partir do ambiente já desenvolvido pelo *RCube*.

### Referências

- [01] DePrycker, M. et al. - *B-ISDN and the OSI Protocol Reference Model*. IEEE Network, mar. 1993, p. 10-18.
- [02] Kawarasaki, M. e Jabbari, B. - *B-ISDN Architecture and Protocol*. IEEE J.S.A.C., v.9, n.9, dez 1991, p. 1405-1415.
- [03] Lyles, J. B. e Swinehart, D. C. - *The Emerging Gigabit Environment and the Role of Local ATM*. IEEE Comm. Mag., abr 1992, p. 53-58.

- [04] Shotobatake, Y. et al. - *A One-Chip Scalable 8\*8 ATM Switch LSI Employing Shared Buffer Architecture*. IEEE JSAC, v. 9, n. 8, out. 1991, p.1248-54.
- [05] Kondoh, H. et al. - *A 622 Mb/s 8 x 8 ATM Switch Chip Set with Shared Multibuffer Architecture*. IEEE J. Solid State Circuits, v.28, n.7, jul 1993, p. 808-815.
- [06] Kosaki, T. et al. - *32 x 32 Shared Buffer Type ATM Switch VLSIs for B-ISDNs*. IEEE J.S.A.C., v.9, n.8, out 1991, p. 1239-47.
- [07] MASI - *RCube: A High Performance Reconfigurable Packet Switching Router*. Internal Technical Report. Laboratório MASI/CAO&VLSI, UPMC, Paris, França, 1993.
- [08] Newman, P. - *ATM Technology for Corporate Networks*. IEEE Comm. Mag., abril 1992, p. 90-101.
- [09] Rathgeb, E. P. et al. - *Buffering Concepts for ATM Switching Networks*. Proc. IEEE GLOBECOM'88, dez 1988, p. 1277-81.
- [10] Karol, M. et al. - *Input versus output queueing on a space division packet switching*. IEEE Trans. on Comm., v.35, n.12, dez 1987.
- [11] Marbot, R. et al. - *Integration of Multiple Bidirectional Point-to-Point Serial Links in the Gigabits per Second Range*. Proc. Hot Interconnects Symposium, Paris, 1993.
- [12] Bull - *BULLIT: The Bull Serial Link Technology*. Technical Report, Laboratório Bull, Les Clayes-sous-Bois, França, 1993
- [13] Zerrouk, B. - *Du Routage par Intervalles*. Colóquio "De la specification à la validation d'architectures de systèmes informatiques". Paris, 1993.
- [14] Santoro, N. e Khatib, R. - *Labelling and Implicit Routing in Networks*. The Computer Journal, v.28, n.1, Fev. 1985, p. 5-8.
- [15] van Leeuwen, J. e Tan, R. B. - *Interval Routing*. The Computer Journal, vol.30, n.1, 1987, p.298-307.
- [16] CCITT Recomendação I.361 - *ATM Layer Specification*, Melbourne 1991.
- [17] Trajkovic, L. e Golestani, S. J. - *Congestion Control for Multimedia Services*. IEEE Network, v.6, n.5, set 1992, p. 20-26.
- [18] Gavras, A. e Xu, X. - *BERGATE: Enhanced Network System - System Concept*. Proc. HPCS'92, fev. 1992.