

## Arquitetura Flexível de uma Interface de Linha Física para o Computador ATM COMATM\*

Mamadou Touré<sup>(2)</sup>  
mamadou@lasic.ufpb.br

Hamilton Soares da Silva<sup>(1)</sup>  
hamilton@lasic.ufpb.br

William Ferreira Giozza<sup>(1)</sup>  
giozza@lasic.ufpb.br

(1) Departamento de Informática - CCEN/UFPB - Campus-I - João Pessoa - Pb

(2) Bolsista DTI do Projeto COMATM/PROTEM-II/CNPq

### Resumo

O presente trabalho apresenta uma arquitetura flexível para a Interface de Linha Física (ILF) do Computador ATM COMATM. A ILF é uma arquitetura flexível e modular podendo trabalhar com os padrões SONET (155 ou 622 Mbps) e TAXI (100 Mbps). A sua interface com o módulo de comutação ATM (MCOA) do COMATM é feita usando sinais do padrão UTOPIA. Apresentaremos também detalhes da organização do *hardware* e do *software* e da interligação dos blocos constituindo a ILF, concluindo com uma proposta de uma arquitetura de interface linha física genérica para comutadores ATM.

### Abstract

*This document presents the physical interface (ILF) architecture of a LAN ATM switch (COMATM). The ILF is a flexible and modular architecture which can be use with physical layer standards SONET (155 or 622Mbps) or TAXI (100 Mbps). Its interface with the ATM switching module (MCOA), implementing the ATM layer functions, is based on UTOPIA standard signals. We'll point out the details of the hardware/software organization and the junction of the ILF block, and a proposition of a generic physical interface for ATM switches.*

## 1. INTRODUÇÃO

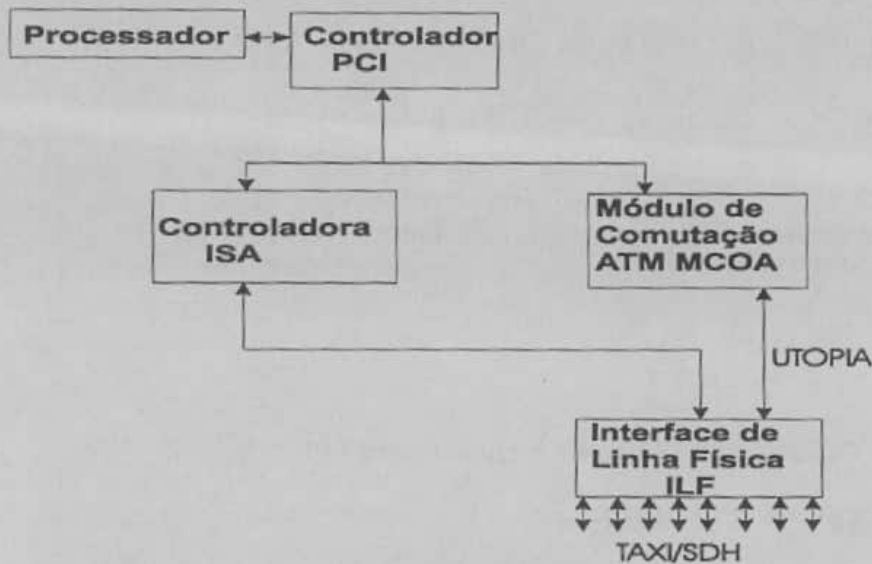
O Modo de Transferência Assíncrono (ATM) permite a concepção de redes locais com banda passante e vazão superiores às das redes locais com meio compartilhado. Além disso o ATM é uma tecnologia de transporte flexível que pode integrar todos os tipos de mídia tais como voz, dados e vídeo [1,2].

Nos últimos anos constatou-se uma crescente demanda da tecnologia ATM nas redes locais e/ou corporativas, isto para prover interconexão de servidores de redes a diversas redes locais ou simplesmente a interconexão das próprias redes de uma forma quase que transparente. Esta demanda vai continuar a crescer pois está tendo ultimamente uma grande procura de serviços multimídia, principalmente os que envolvem vídeo, necessitando portanto de acessos a alta velocidade. A utilização do ATM para redes locais e públicas permitirá compatibilidade total na interconexão das redes, evitando a necessidade de se prover modos de interfuncionamento, como os utilizados para a interconexão de redes de alta velocidade como DQDB (150 Mbps) e FDDI (100 Mbps).

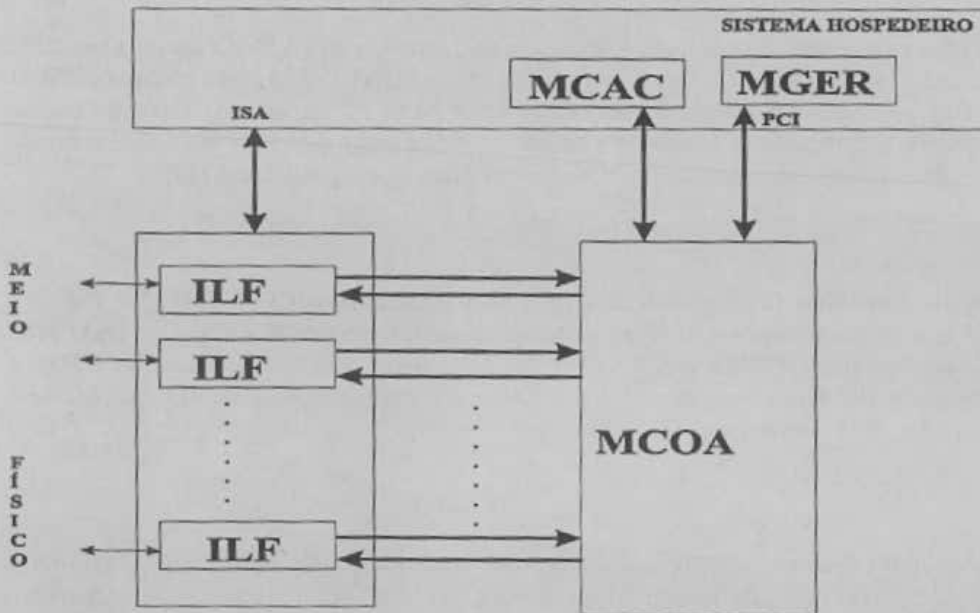
O presente documento apresenta a especificação técnica da Interface de Linha Física (ILF) para o projeto de um comutador ATM (COMATM) concebido por Giozza e Zerrouk [3,4].

A ILF é um componente do COMATM [4], ilustrado na Figura 1, que faz a interface entre o Módulo de Comutação ATM (MCOA), utilizando padrão UTOPIA [5,6], e o meio físico segundo os padrões SDH/SONET (155 OU 622 Mbps) ou TAXI (100 Mbps) [2].

\* Trabalho suportado pelo CNPQ (projeto COMATM/PROTEM-II)



a) Sistema Hospedeiro





ILF: Interface de Linha Física

MGER: Módulo de Gerência

MCOA: Módulo de Comutação ATM

MCAC: Módulo de Controle de Admissão

Obs: a seta  indica linha paralela  
a seta  indica linha serial

## b) Modelo Funcional do Comutador

Figura 1: Arquitetura do Comutador COMATM [4]

A Interface de Linha Física (ILF) do COMATM implementa as seguintes funções:

- \* Interface com o meio físico e sincronização dos bits
- \* Geração da seqüência de HEC (*Header Error Control*) e verificação de erros no cabeçalho da célula.
- \* Embaralhamento e desembaralhamento no caso do sistema de transmissão baseado em SDH/SONET
- \* Mapeamento das células dentro dos quadros SDH.
- \* Delimitação das células
- \* Detecção e geração de células de gerenciamento de nível Físico

A ILF pode ser utilizada para prover mecanismos de transferência de células ATM usando o sistema de transmissão **SDH/SONET** ou **TAXI**. A escolha de um desses dois sistemas de transmissão será feita a partir de um *software* que acompanhará a ILF.

A interface ILF no seu funcionamento típico recebe células das saídas do Módulo de Comutação ATM (MCOA) gera o octeto de controle de erros no cabeçalho HEC (*Header Error Control*) e serializa cada célula completa para transmissão no meio físico, segundo um protocolo específico ( por exemplo TAXI 100 Mbps, SDH/SONET 155 Mbps etc...). Por outro lado, a ILF recebe do meio físico seqüências de bits em série, identifica as células com seus cabeçalhos, verifica erros no cabeçalho e transmite para o MCOA em paralelo as células válidas e descarta as corrompidas. Além disso, a ILF é responsável pela detecção e geração de células de gerenciamento de nível Físico.

A ligação física entre cada ILF e o MCOA é feita através dos circuitos elementares de comutação ATM (ABSE [*ATM Basic Switch Element*]) realizados com ASICs que compõem as portas da matriz de comutação ATM implementada pelo MCOA [4]. Em cada circuito ABSE existe um bloco denominado Interface com a Camada Física (ICF/ABSE) que implementa as funções de interfuncionamento Camada Física/Camada ATM segundo o padrão UTOPIA [5]. A comunicação entre estas unidades (ILF, ABSE) é feita na base de octetos seguindo o padrão de sinais de controle UTOPIA conforme ilustrado na Figura 2. Na implementação baseada no protótipo **COMATM**, a ILF provê também funções de serialização/paralelização em conformidade com o padrão de transmissão **TAXI** (*Transparent Asynchronous Xmitter Interface*) [7], ou **SDH/SONET**. A Figura 1a mostra como a ILF se liga com o sistema hospedeiro e o Módulo de Comutação ATM (MCOA).

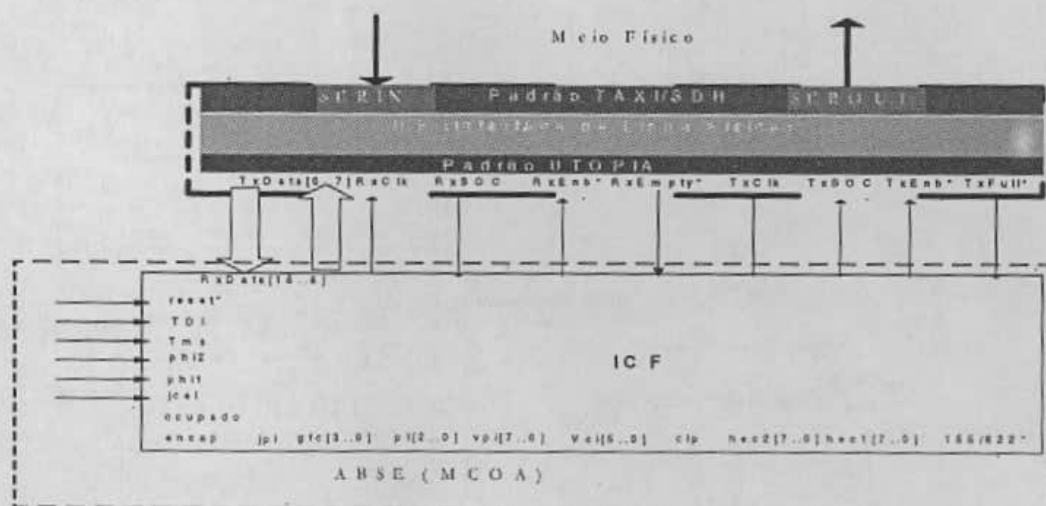


Figura 2: Interface UTOPIA da ILF com o MCOA (ICF/ABSE)

Na Seção 2 apresentaremos a descrição dos módulos da ILF com seus respectivos blocos funcionais. Na Seção 3 iremos apresentar uma proposta de uma arquitetura de ILF genérica para comutadores ATM. Faremos na última seção a conclusão do trabalho como um todo.

## 2. DESCRIÇÃO DOS MÓDULOS DA ILF

Para desenvolver uma arquitetura flexível e modular, partimos do princípio de particionar a interface de linha física (ILF) em vários blocos funcionais independentes mas interligados entre si, cada bloco efetuando tarefas específicas bem determinadas. A Figura 1a mostra como a ILF se liga aos outros módulos e ao sistema hospedeiro.

A ILF recebe as células ATM do meio físico numa taxa de 100 Mbps no caso do padrão **TAXI** e 155 ou 622 Mbps no caso do **SDH/SONET**, envia e recebe informações de gerenciamento de nível físico ao sistema hospedeiro através de um barramento ISA, e se comunica com o ICF/ABSE do Módulo MCOA, através da interface UTOPIA padronizada pelo Fórum ATM. A ILF é composta de dois módulos (Transmissão e Recepção) que são constituídos por dezenove blocos conforme mostrado na Figura 3. Os blocos desses módulos são: **Controle de Recepção**, **Verificar Cabeçalho**, **Diagnóstico de Erros**, **TaxiChip Receiver**, **Detectar Células de Gerenciamento**, **Controlador de Fluxo**, **Buffer de Recepção**, **Desembaralhador**, **SDH-Receiver**, **Mux para Sistema de Recepção**, **Controle de Transmissão**, **Gerador de Cabeçalho**, **Buffer de Transmissão**, **Mux para**

Sistema de Transmissão, Embaralhador, SETI 3005, TaxiChip Transmitter, e ISA Bus. A descrição detalhada destes módulos será feita nas seções a seguir.

U  
M  
C  
O  
P  
I  
A

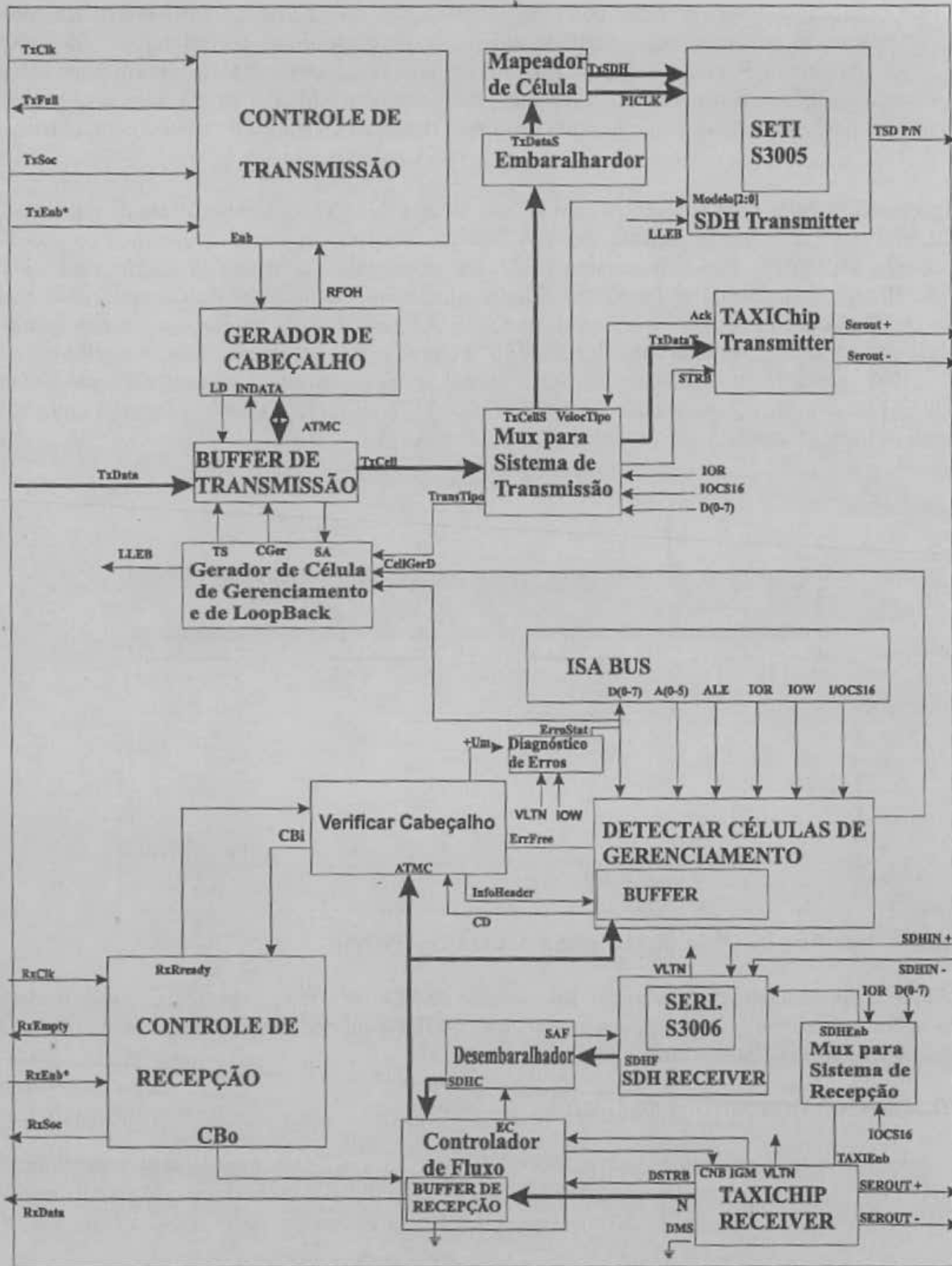


Figura 3: Arquitetura da Interface da Linha Física ILF



## 2.1 Módulo de Transmissão

Este módulo, constituído de um conjunto de blocos funcionais, é responsável pela recepção das células oriundas do MCOA (ABSE /ICF), da geração de cabeçalho no nível físico, do embaralhamento e mapeamento das células (se for empregado o SDH como sistema de transmissão), da serialização das células montadas, da sua codificação (por exemplo 4B/5B para sistema de transmissão baseado no padrão TAXI), e da transmissão destas no meio físico. O Módulo de Transmissão é composto dos blocos: **Controle de Transmissão**, **Gerador de Cabeçalho**, **Buffer de Transmissão**, **TaxiChip Transmitter**, **Mux para Sistema de Transmissão**, **Embaralhador**, **Mapeador de Célula** e do **SDH Transmitter**. A descrição funcional destes blocos será feita a seguir.

### 2.1.1 Controle de Transmissão

O bloco **Controle de Transmissão**, ilustrado na Figura 4 trata da transferência das células vindas do MCOA para a ILF e armazenadas no **Buffer de Transmissão**. O protocolo utilizado para receber e controlar as células vindas da ICF é o padrão UTOPIA. O bloco **Controle de Transmissão** se comunica com o bloco **Gerador de Cabeçalho** recebendo dele o sinal **RFOH** (*Ready For an Other Header*), e enviando o sinal **Enb**. O sinal **RFOH** avisa ao bloco **Controle de Transmissão** que o **Gerador de Cabeçalho** está pronto para iniciar uma nova operação de cálculo do HEC (*Header Error Control*) e assim gerar um novo cabeçalho. Com a ativação do sinal **TxEhb\***, o MCOA fica liberado para enviar uma outra célula e, conseqüentemente, o bloco **Controle de Transmissão** ativa o sinal **Enb** para indicar ao **Gerador de Cabeçalho** a presença de dados válidos.

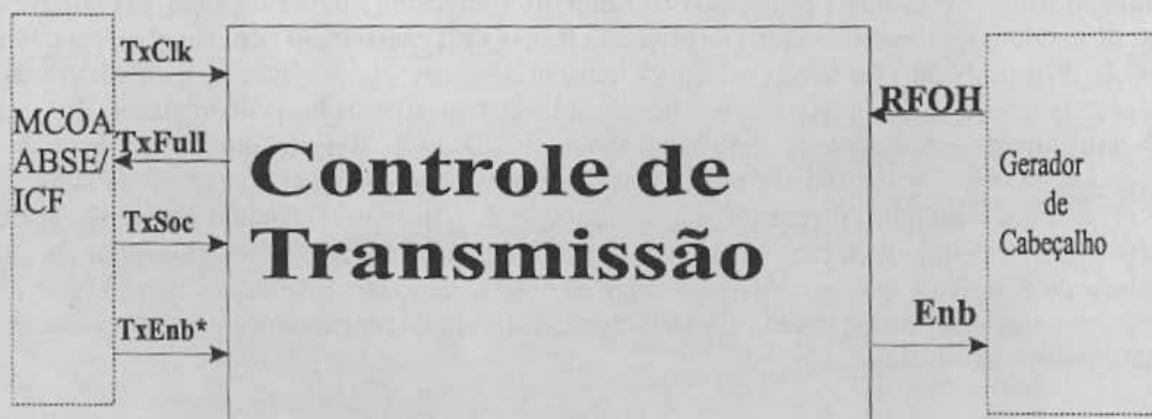


Figura 4: Bloco Controle de Transmissão

### 2.1.2 Gerador de Cabeçalho

Este bloco, ilustrado na Figura 5, calcula o HEC e o insere no **Buffer de Transmissão** para completar o cabeçalho da célula a ser transmitida. A geração do cabeçalho é feita utilizando o polinômio gerador  $X^8 + X^2 + X + 1$  que vai servir de divisor dos 4 primeiros octetos da célula constituída pelos campos **VPI**, **GFC**, **VCI**, **PT**, **CLP** (armazenados no **Buffer de Transmissão**), deslocados de 8 posições a esquerda. Uma operação **XOR** será efetuada entre o resto da divisão e o octeto 10101010, resultando no 5º octeto que será o **HEC**. Uma vez feito isto, o **Gerador de Cabeçalho** insere o **HEC** no **Buffer de Transmissão** e ativa os sinais, **RFOH** para avisar ao bloco **Controle de Transmissão** que está pronto para montar uma nova célula, e o sinal **LD**, utilizado para solicitar a liberação da célula armazenada no **Buffer de Transmissão**. O **Gerador de Cabeçalho** espera a ativação do sinal **Enb** por parte do **Controle de Transmissão** para iniciar uma nova operação de cálculo do HEC.

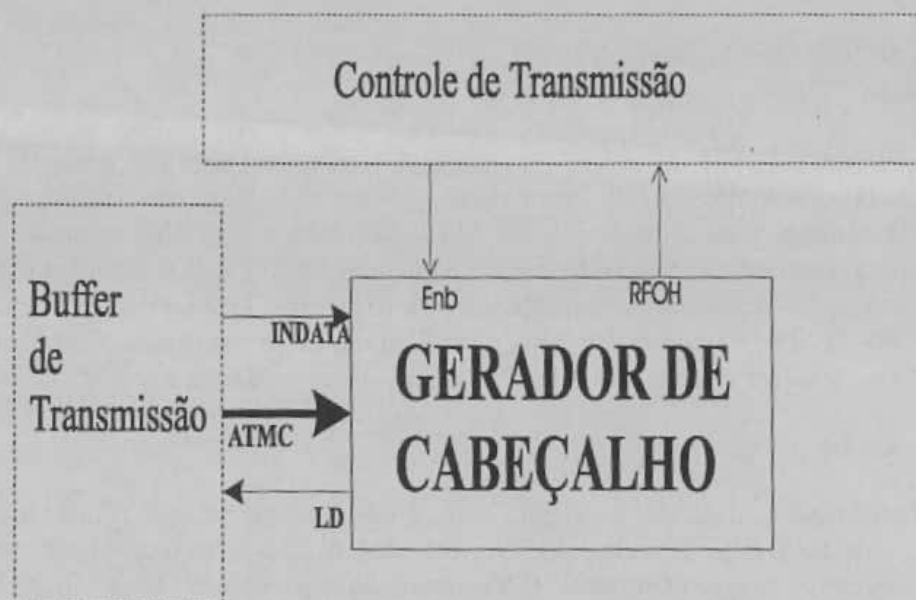


Figura 5: Bloco Gerador de Cabeçalho

### 2.1.3 Buffer de Transmissão

O *Buffer de Transmissão*, ilustrado na Figura 6, recebe as células do **ABSE/MCOA** na base de octetos e as transmite para o **Mux para Sistema de Transmissão**. A transferência das células (octetos) armazenadas no *Buffer de Transmissão* é controlada pelo bloco **Gerador de Cabeçalho** através do sinal **LD** que fica ativado quando o HEC tiver sido inserido no cabeçalho da célula. O *Buffer de Transmissão* se comunica com o bloco **Mux para Sistema de Transmissão** (ver seção a seguir) transmitindo para ele a célula já montada. O *Buffer de Transmissão* pode receber também células (de gerenciamento, etc...) do sistema hospedeiro através do **Gerador de Células de Gerenciamento e de Loopback**, e do bloco **Detectar Células de Gerenciamento**, células estas a serem transmitida para o meio físico. A transmissão das células de gerenciamento de nível Físico, é feita através de troca dos sinais **TS** (Transmissão Solicitada) para solicitar a permissão de transmitir. Quando o *Buffer de Transmissão* estiver pronto para atender esta solicitação, ele envia o sinal **SA** (Solicitação Aceita) para o **Gerador de Células de Gerenciamento e de Loopback** que, por sua vez, envia as células de gerenciamento de nível Físico pelo sinal **CGer**. O mesmo protocolo de sinais é usado pelo **Detectar Células de Gerenciamento** para solicitar a geração de células de gerenciamento de nível Físico.

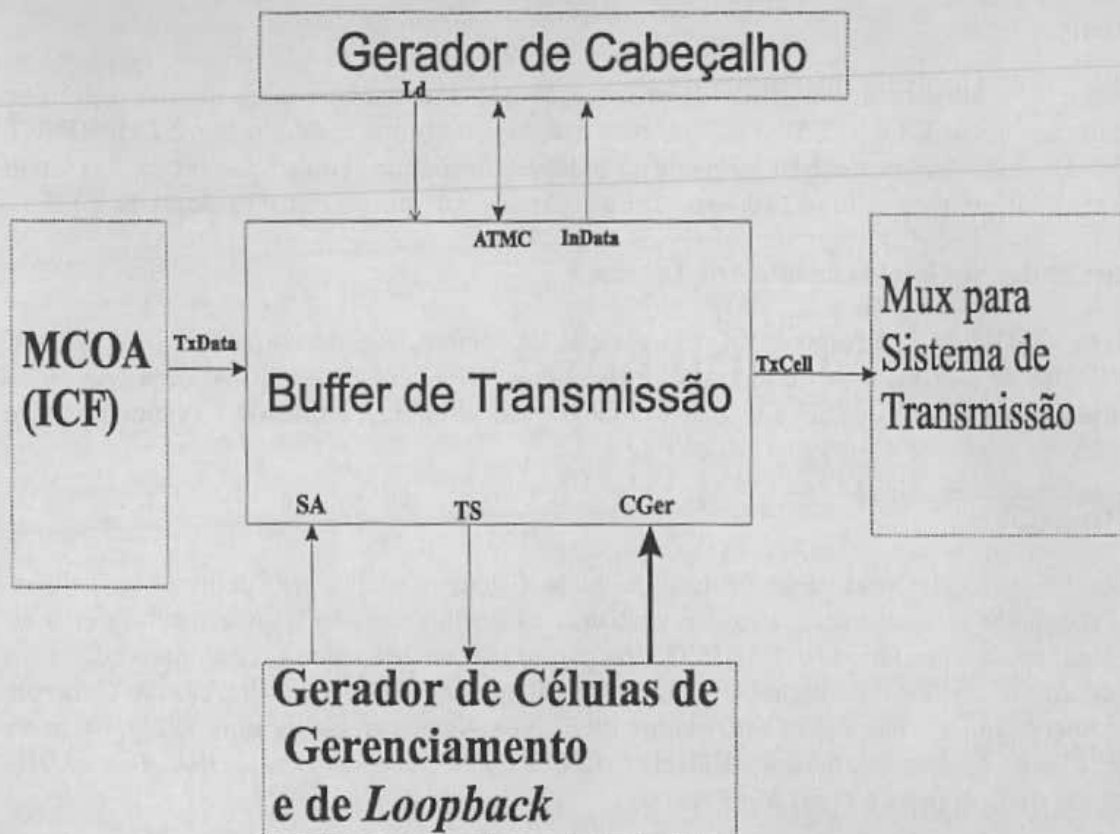


Figura 6: Buffer de Transmissão

#### 2.1.4 Mux para Sistema de Transmissão

Durante a inicialização do sistema, este bloco através dos sinais **IOR**, **IOCS16 D(0-7)** recebidos do barramento ISA fica, fixado para usar um dos sistema de transmissão. Uma vez fixado ele habilita uma das saídas para o *TaxiChip Transmitter* ou o *SDH-Transmitter*, e sua função será de receber células do *Buffer de Transmissão* e transmiti-las a um dos dispositivos inicialmente selecionado.

#### 2.1.5 TaxiChip Transmitter

O *TaxiChip Transmitter* é um bloco consistindo de um *Latch* de entrada, um codificador, um *Shift Register* paralelo-serial, um multiplicador "*Phase Locked Loop*" (PLL), e algumas lógicas de controle [7]. Ele recebe um conjunto de octetos do *Mux para Sistema de Transmissão*, sendo que esses octetos ficam armazenados no *latch* após serem codificados e depois colocados na sua saída serial. O esquema de codificação utilizado é o 4B/5B especificado pelo padrão ANSI X3T9.5. Esta codificação divide 8 bits em dois *nibbles* (i.e. 2 blocos de 4 bits), sendo que cada *nibble* é codificado em um símbolo de 5 bits. Os "10-bit byte" codificados são formatados numa fila de dados NRZI (*Non Return to Zero Invert on One*) para a saída do *TaxiChip Transmitter*. O *TaxiChip Transmitter* usa a padronização TAXI para sua comunicação com o meio físico.

#### 2.1.6 Embaralhador

Este bloco é responsável no caso da interface SDH/SONET, pelo embaralhamento do *payload* da célula. Para fazer o embaralhamento, aplica-se no campo de *payload* da célula o polinômio gerador de sequência pseudo-aleatória  $X^{43} + 1$ . Ele recebe a célula completa (sinal TxCell na Figura 3) do *Mux para Sistema de Transmissão*, efetua o embaralhamento seguindo o procedimento acima citado, e entrega a célula (sinal TxDataS na Figura 3) embaralhada para o bloco *Mapeador de Célula*.

#### 2.1.7 Mapeador de Célula

Este bloco recebe células ATM (sinal TxDataS na Figura 3) já prontas para serem transmitidas e as mapeia dentro dos quadros SDH para serem transmitidos (sinal TxSDH na Figura 3) para o *SDH-Transmitter*.

### 2.1.8 SDH-Transmitter

O *SDH-Transmitter* [8] é um circuito integrado de serialização que trabalha com taxas de transmissão de 139,264 para sistemas de transmissão E4 e 155,52 a 622,08 para sistemas de transmissão do tipo SDH/SONET OC-3 e SDH/SONET OC-12 respectivamente. Esse *chip* realiza todas as funções de serialização das células vindo do Mapeador de Célula em conformidade com os padrões definidos para os sistemas de transmissão SDH e E4.

### 2.1.9 Gerador de Células de Gerenciamento e de Loopback

Este bloco (conferir na Figura 3) é responsável pela geração de células de gerenciamento e de *loopback*. Após o bloco Detectar Células de Gerenciamento detectar uma solicitação de geração de células de *loopback* ou de gerenciamento por parte do meio físico, ele ativa o sinal CellGerD Gerador de Células de Gerenciamento e de *Loopback*

## 2.2 Módulo de Recepção

O Módulo de Recepção recebe uma série de bits do meio físico, os paraleliza, delimita as células, desembaralha a *payload* da célula se for preciso, extrai e analisa o cabeçalho. Se não houver nenhum erro no cabeçalho da célula recebida, esta é transmitida para o MCOA ou para o sistema hospedeiro (caso que a célula for de gerenciamento), caso contrário, a célula é descartada. O Módulo de Recepção é composto dos blocos: Controle de Recepção, Buffer de Recepção (sub-bloco do Controlador de Fluxo), Verificar Cabeçalho, Diagnóstico de Erros, Controlador de Fluxo, Desembaralhador, Detectar Células de Gerenciamento, ISA Bus, SDH-Receiver, Mux para Sistema de Recepção e TaxiChip Receiver.

### 2.2.1 Controle de Recepção

Este bloco da ILF, ilustrado na Figura 7, é responsável pelo controle da transferência das células recebidas do meio físico (sem erro de cabeçalho) para o MCOA, usando sinais padrão UTOPIA. Ele controla o fluxo de células trafegando da ILF para o MCOA, envia um sinal para o Buffer de Recepção através do Controlador de Fluxo, para habilitar a transmissão da célula para o MCOA (CBo=0) ou, para solicitar o seu descarte (CBo=1), desativando ou ativando, respectivamente, o sinal CBo. Quando o bloco Controle de Recepção recebe do MCOA a permissão para transmitir uma nova célula (com RxEnb\* = 0), ele avisa então o bloco Verificar Cabeçalho através do sinal RxReady que está pronto para receber uma nova célula. Se o bloco Controle de Recepção recebe do bloco Verificar Cabeçalho o sinal CBi desativado ou ativado, ele respectivamente, desativa ou ativa também o sinal CBo, para descartar ou transmitir a célula armazenada no Buffer de Recepção do bloco Controlador de Fluxo



Figura 7: Bloco Controle de Recepção



### 2.2.2 Controlador de Fluxo

Este bloco, ilustrado na Figura 8, é responsável pelo controle do fluxo das células vindas de uma das interfaces de serialização (*TaxiChip Receiver* ou *SDH-Receiver*) e destinadas aos blocos *Verificar Cabeçalho*, *Detectar Células de Gerenciamento* e ao MCOA.

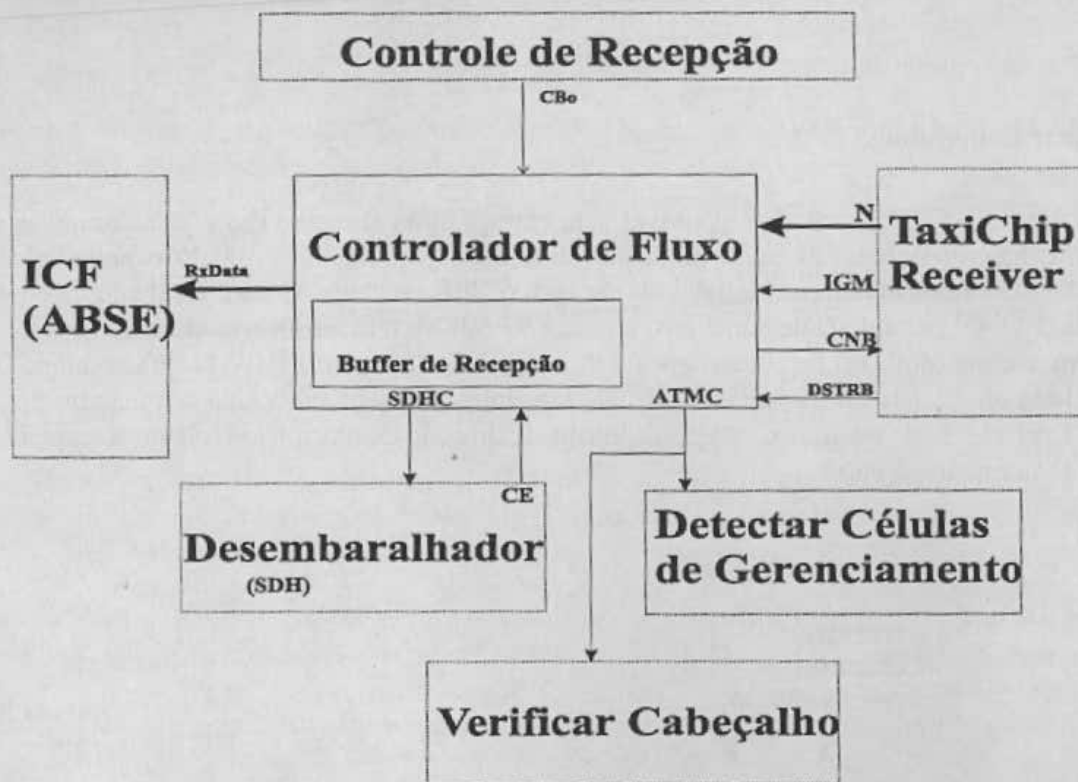


Figura 8: Controlador de Fluxo

Caso o sistema de transmissão empregado seja o SDH/SONET, o **Controlador de Fluxo** envia para o **Desembaralhador** o sinal EC (Envia Célula) para que este último possa liberar a célula desembaralhada SDHC. Uma vez a célula desembaralhada recebida, o **Controlador de Fluxo** a armazena no **Buffer de Recepção**, e a transmite para os blocos **Verificar Cabeçalho** e **Detectar Células de Gerenciamento**. Do bloco **Controle de Recepção**, ele recebe o sinal CBo solicitando-o a transmitir para o MCOA a célula armazenada no **Buffer de Recepção**, caso  $CBo=0$ ; ou esvaziar o **Buffer de Recepção**, caso contrário. No caso de um sistema de transmissão baseado no padrão TAXI, o **Controlador de Fluxo** fica avisado da presença de dados válidos através do sinal DSTRB do *TaxiChip Receiver* e então pode autorizar a liberação das células ativando o sinal CNB.

### 2.2.3 Buffer de Recepção

O **Buffer de Recepção**, um sub-bloco do bloco **Controlador de Fluxo**, é responsável pelo armazenamento das células vindas do *TaxiChip Receiver* e do *SDH Receiver*. Os cabeçalhos das células armazenadas nele, serão tratados pelo bloco **Verificar Cabeçalho** antes das células serem descartadas ou transmitidas para o MCOA. Para transmitir ou descartar as células, o **Buffer de Recepção** espera uma sinalização positiva ou negativa do bloco **Verificar Cabeçalho** através do sinal CBo.

Uma sinalização positiva do bloco **Verificar Cabeçalho** significa para o **Buffer de Recepção** que a célula recebida está sem erro no cabeçalho e pode ser entregue ao MCOA. Neste caso, o sinal CBo (Clear Buffer[limpa Buffer]) é colocado em nível baixo e (i.e.  $CBo=0$ ). Ao avisar o **Buffer de Recepção** que o MCOA está pronto para receber a célula, o bloco **Controle de Recepção** ativa o sinal RxReady para avisar ao bloco **Verificar Cabeçalho** que o **Buffer de Recepção** está pronto para enviar uma outra célula para o MCOA. O bloco **Controle de Recepção** solicita uma liberação das células armazenadas no **Buffer de Recepção** ao MCOA, se e somente se, o sinal RxEnb\* estiver baixo (i.e.,  $RxEnb^*=0$ ).

Uma sinalização negativa do bloco **Verificar Cabeçalho** significa que a célula recebida está com erro no cabeçalho e deve ser descartada; isto é, o **Buffer de Recepção** deve ser esvaziado. Neste caso, o sinal **CBo** é alto (i.é. **CBo** = 1).

A ativação do sinal **DSTRB** pelo **TaxiChip Receiver** é usada para avisar ao **Buffer de Recepção** a presença de dados (octetos) na sua saída.

### 2.2.4 Verificar Cabeçalho

Este bloco, ilustrado na Figura 9, é responsável pela extração do cabeçalho das células oriundas do meio físico. Ele verifica a integridade do cabeçalho efetuando uma divisão polinomial usando o polinômio gerador padrão ( $X^8 + X^2 + X + 1$ ) e comparando o resultado obtido com o HEC recebido. Caso o resultado da comparação indicar erro, a célula deve ser descartada devido a erro no seu cabeçalho e o bloco **Diagnóstico de Erros** é avisado a partir do sinal **+Um**. Caso a célula for de gerenciamento de nível Físico (i.é., sinal **CD** = 1) então o bloco **Detectar Células de Gerenciamento** é alertado da validade ou não da célula detectada através da ativação ou desativação respectiva do sinal **ErrFree**. Caso contrário, este procedimento é aplicado ao bloco **Controle de Recepção** através do sinal **CBi** que vai estar no nível baixo.

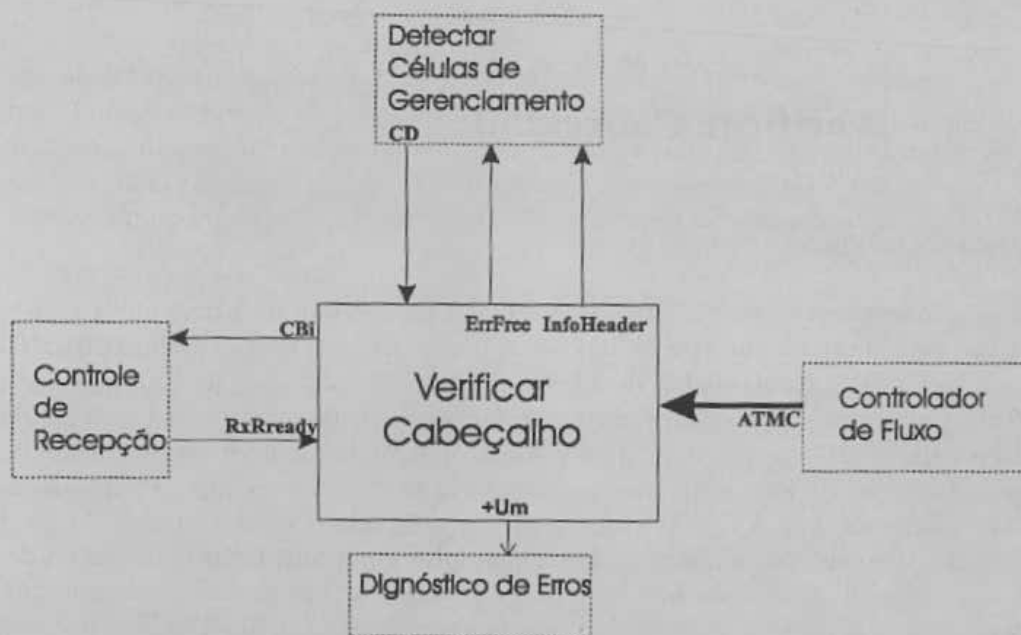


Figura 9 Bloco Verificar Cabeçalho

### 2.2.5 Detectar Células de Gerenciamento

Este bloco, ilustrado na Figura 10, é responsável pela detecção de células de gerenciamento de nível Físico tipo F1, F2 e F3 [2]. Ele recebe o status do cabeçalho da célula de gerenciamento através do sinal **InfoHeader** transmitido pelo bloco **Verificar Cabeçalho**. Uma vez detectadas células de gerenciamento de nível Físico, e que estas foram recebidas sem erro (sinal **ErrFree** = 1), elas são entregues ao sistema hospedeiro via o barramento ISA, usando operações de I/O. Caso a célula de gerenciamento de nível Físico seja de um tipo que necessita a transmissão de uma outra célula de gerenciamento de nível Físico, o bloco através do sinal **CellGerH** ativa o **Gerador de Células de Gerenciamento e Loopback** para gerar e transmitir célula de gerenciamento de nível Físico ao transmissor.

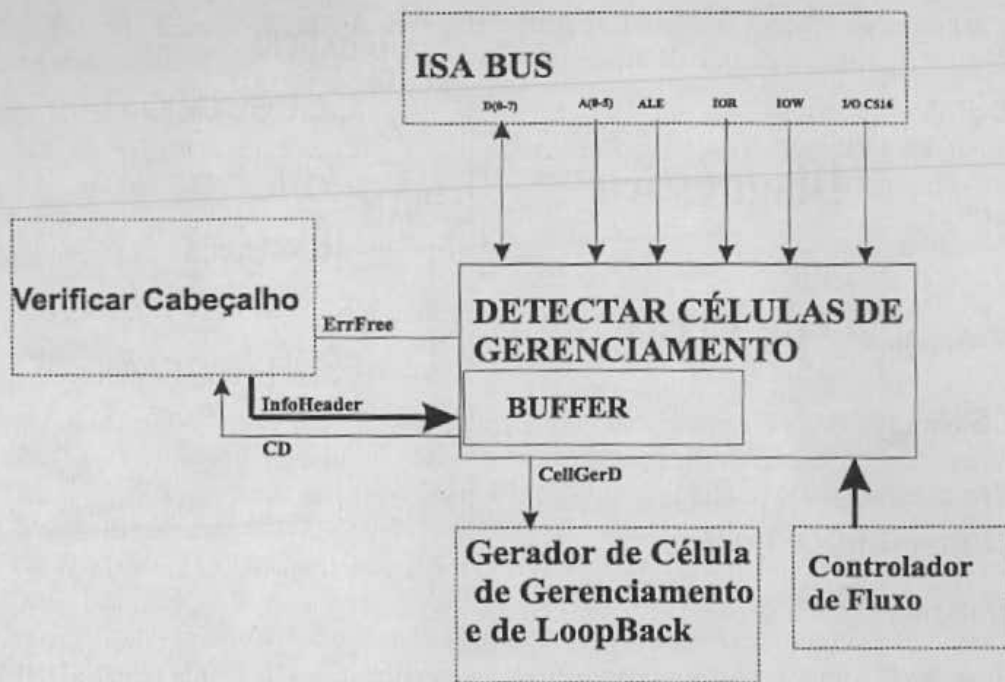


Figura 10. Detectar Células de Gerenciamento

### 2.2.6 TaxiChip Receiver

O *TaxiChip Receiver* [7] recebe as células em um conjunto de bits em série codificados 4B/5B na sua entrada, efetua a decodificação e paraleliza estas para serem entregues ao bloco **Controle de Recepção**. Caso forem detectados erros pelo *TaxiChip Receiver*, ele ativa o sinal **VLTN** para avisar o bloco **Diagnóstico de Erros**. A presença de dados válidos nas saídas do *TaxiChip Receiver* é sinalizada pelo nível alto do sinal **DSTRB**. O sinal **DMS** do *TaxiChip Receiver* (Figura 3) está conectado ao referencial de tensão elétrica **VDD** para que possamos trabalhar em modo 8 bits de dados.

### 2.2.7 ISA Bus

O bloco **ISA Bus** [9] é o barramento por onde trafegam as células da ILF para o sistema hospedeiro e vice versa. Utilizaremos operações de I/O para transmitir e receber células para/do sistema hospedeiro.

### 2.2.8 Diagnóstico de Erro

Este bloco, ilustrado na Figura 11 é utilizado para avisar o sistema hospedeiro de algumas circunstâncias de falhas repetidas no cabeçalho da célula, ou da ocorrência de erro na transmissão. Cada vez que acontecer um erro no cabeçalho da célula recebida, o bloco **Diagnóstico de Erro** fica informado através do sinal **+Um**. Ele por sua vez aciona um contador para registrar o acontecimento, e avisa ao sistema hospedeiro através do sinal **ErroStat** usando o barramento ISA. Quando houver um erro de transmissão, o bloco **Diagnóstico de Erros** fica avisado através do sinal **VLTN**, que lhe notifica de um erro de bit duplo (*Double Bit Error*) no dado, deixando ao sistema hospedeiro efetuar o tratamento adequado relativo a esse tipo de informação.

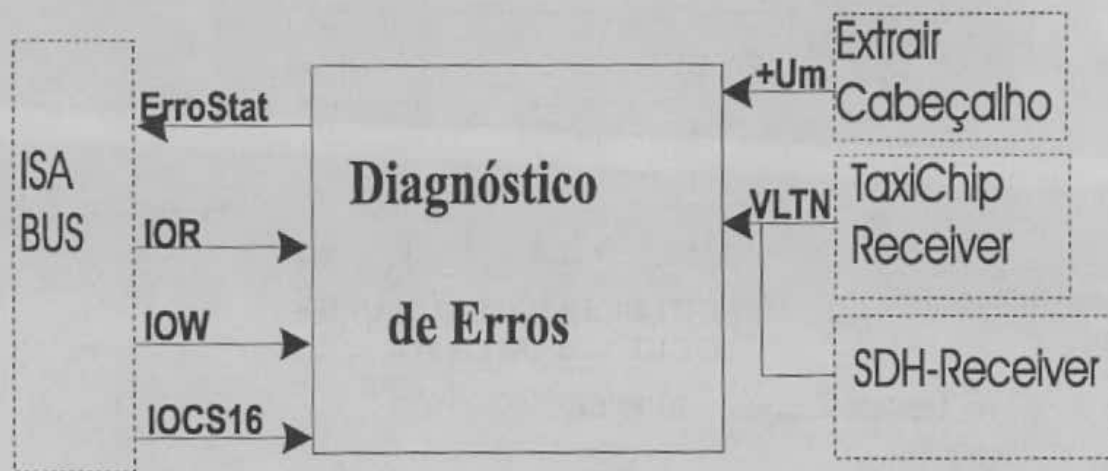


Figura 11: Bloco Diagnóstico de Erros

### 2.2.9 Desembaralhador

Este bloco, ilustrado na Figura 12, é responsável pelo desembaralhamento da célula (sinal SDHF) recebida usando para tal o polinômio  $X^{43} + 1$ . Uma vez a operação efetuada, ele transmite a célula pronta ao **Controlador de Fluxo**. E daí ativa o sinal SAF para solicitar uma outra célula ao **SDH Receiver**.

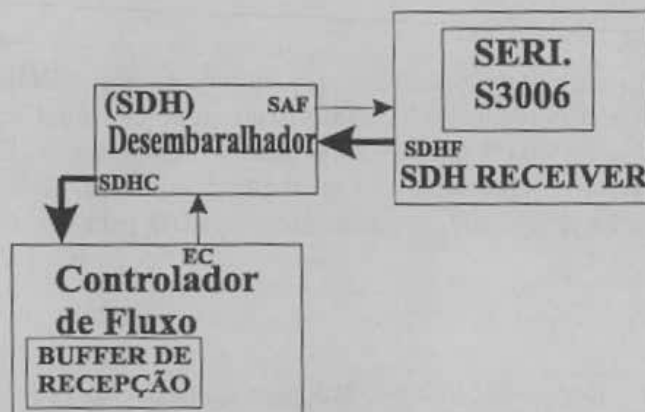


Figura 12. Desembaralhador

### 2.2.10 SDH Receiver

O **SDH Receiver** implementado pelo circuito *Synchronous Electrical Receiver Interface (SERI) S3006* [8], ilustrado na Figura 13, é um dispositivo integrado de paralelização trabalhando com sistemas de transmissão E4 (139,264 Mbps), SONET OC-3 (155,52 Mbps) e SONET OC-12 (622,08 Mbps). Ele efetua todas as funções de serialização e paralelização em conformidade com os padrões de transmissão SDH/SONET e E4. O circuito **S3006** efetua a recuperação do relógio sincronizando-se diretamente ao *stream* de dados. Este circuito realiza também a detecção de quadro SDH/SONET. O *chipset* pode ser utilizado com relógios de 19,44, 38,88, 51,84 e 77,76 Mhz quando opera os modos OC-3, OC-12 do SDH/SONET.



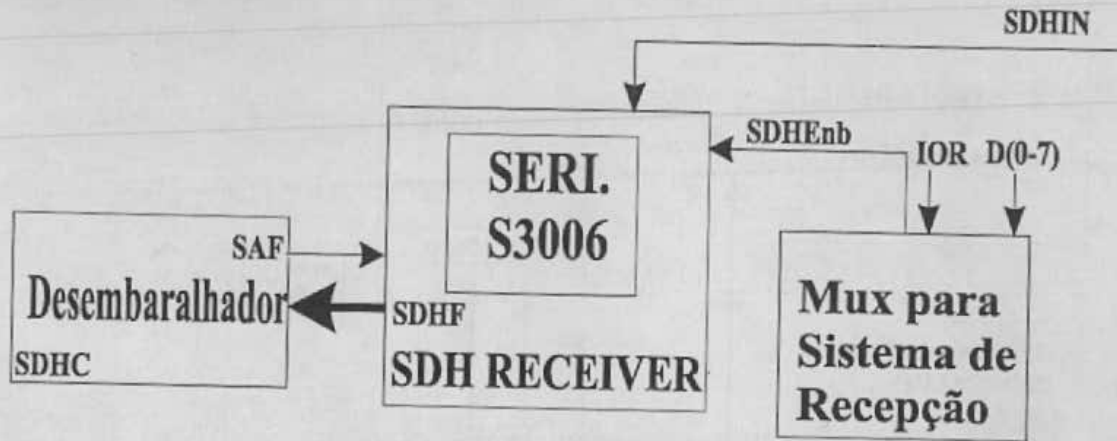


Figura 13. SDH Receiver

### 3 ARQUITETURA DE INTERFACE DE LINHA FÍSICA GENÉRICA PARA COMUTADORES ATM

O objetivo desta proposta é dar uma referência para projetos de arquitetura de Interface de Linha Física para comutadores ATM. A partir desta arquitetura poder-se-ia implementar uma ILF usando SDH/SONET a 155 ou 622 Mbps, par trançado blindado ou não a 100 ou 155 Mbps, DS3 etc...

Para poder elaborar tal arquitetura partimos do princípio em usar o conceito de herança funcional que permite que todos os objetos (i.e., os sistemas de transmissão. E.g. sistemas de transmissão baseadas em: SDH/SONET, TAXI, DS3, etc ) pertencendo a mesma classe (por exemplo sistemas de transmissão baseadas em célula: e.g TAXI, 155Mbps usando par trançado não blindado) possam herdar (i.e., aproveitar) das funcionalidades dos módulos que atendem este tipo de objetos. Por exemplo sabemos que a codificação usada para sistemas de transmissão de 100 Mbps é o 4B/5B; portanto podemos criar um bloco de codificação 4B/5B atendendo todos sistemas de transmissão que necessitam tal operação. Um outro exemplo mais simples é a geração e verificação do HEC das células ATM na transmissão e na recepção respectivamente; operações usadas para todos os tipos de sistemas de transmissão; portanto podemos usar somente um bloco de geração de HEC e um de verificação do mesmo para todos os tipos de sistemas de transmissão. Foi usando este conceito que, criamos blocos funcionais atendendo classes (conjunto de objetos que tem mesmo comportamento em determinadas situações) de sistemas de transmissão. A arquitetura desta proposta é descrita na Figura 14.

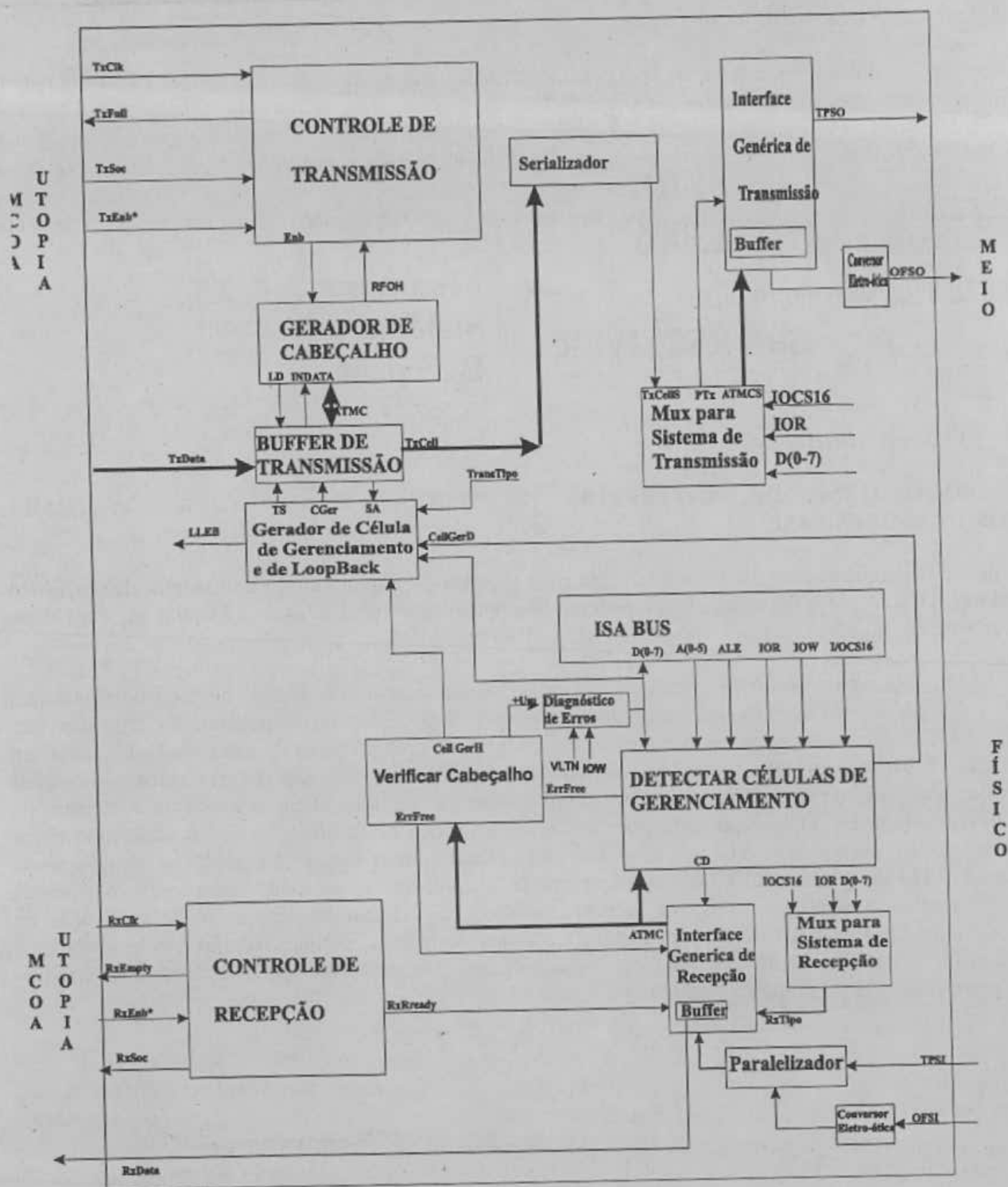


Figura: 14 Arquitetura de uma ILF Genérica

### 3.1 Descrição da Arquitetura

Nessa arquitetura como foi mencionado previamente, criamos blocos funcionais que são usados para todos os tipos de sistemas de transmissão. Blocos esses sendo **Controle de Transmissão**, **Gerador de Cabeçalho**, **Buffer de Transmissão**, **Gerador de Células de Gerenciamento e de Loopback**, **Verificar Cabeçalho**, **Controle de Recepção**, **Detectar Células de Gerenciamento**, **Diagnóstico de Erro**, **Serializador** e **Paralelizador**. Exceto os blocos **Serializador** e **Paralelizador**, todos os outros se comportam do mesmo jeito que os da arquitetura da Figura 3 e descritos nas seções 2.1 e 2.2. Os blocos **Serializador** e **Paralelizador** como os seus nomes sugerem efetuam as operações de serialização e paralelização das células a serem transmitidas ou recebidas respectivamente pela ILF. Temos também os blocos **Mux para Sistema de Transmissão** e **Mux para Sistema de Recepção**, que durante a fase de inicialização do sistema receberão parâmetros (tipo de sistema de transmissão empregada, taxa de transmissão, etc), do sistema hospedeiro, que permitirá ao bloco **Interface Genérica de Transmissão** determinar o sistema de transmissão e suas funcionalidades.

### 3.1.1 Interface Genérica de Transmissão

Este bloco, ilustrado na Figura 15, recebe do **Mux para Sistema de Transmissão** o sinal **PTx** que contém todas informações necessárias para o sistema de transmissão (por exemplo SDH/SONET a 155 Mbps). Uma vez isso feito, a **Interface Genérica de Transmissão** poderá efetuar todas operações necessárias de adaptação das células recebidas (**ATMCS**) conforme sistema empregado. Caso o meio de transmissão for fibra óptica, a célula sai do bloco já dentro dos padrões de transmissão inicialmente escolhidos, para ser entregue ao **Conversor Eletro-óptica** que vai efetuar a conversão dos sinais elétricos em sinais ópticos. Caso contrário, a célula é diretamente transmitida para o meio físico por intermédio do sinal **TPSO**.

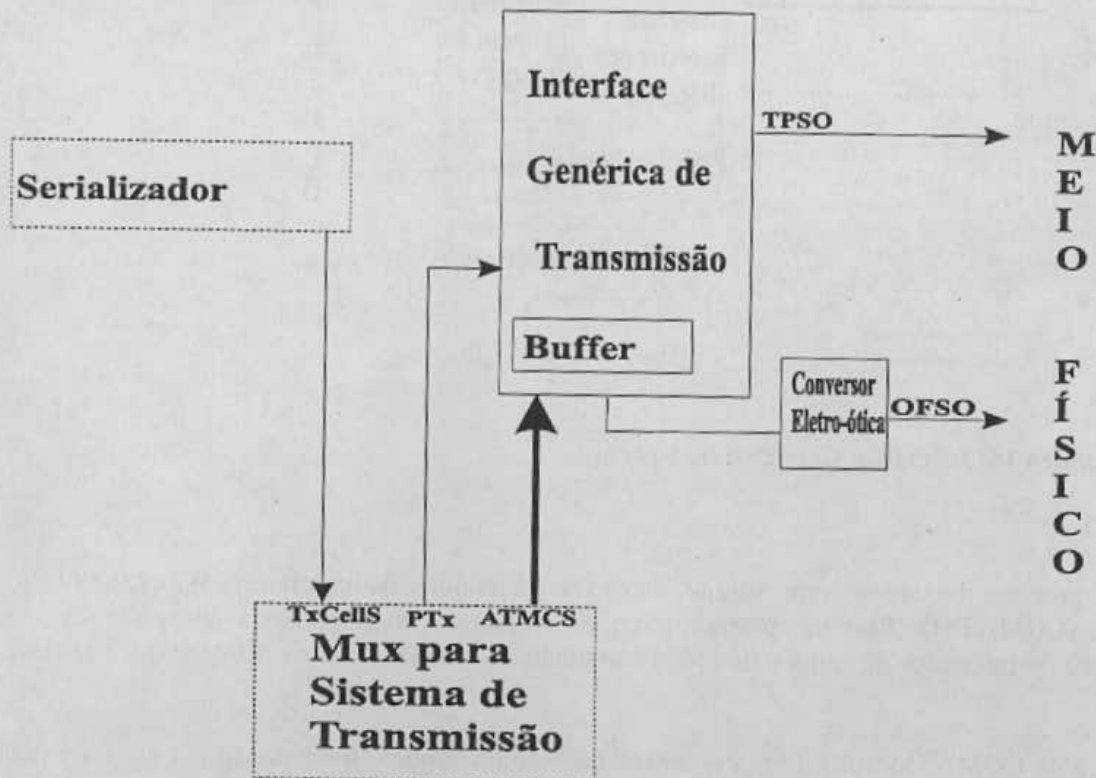


Figura 15: Interface Genérica de Transmissão

### 3.1.2 Interface Genérica de Recepção

Este bloco, ilustrado na Figura 16, será encarregado de receber quadros ou células vindas do meio físico, delimitar as células e entregar estas aos blocos **Detectar Células de Gerenciamento** e **Verificar Cabeçalho**. Caso for detectado que a célula é de gerenciamento de nível Físico ou vem com erro no cabeçalho, o **Buffer** deste bloco é esvaziado; caso contrário (i.e., cabeçalho sem erro **ErrFree=1**, e a célula não for de gerenciamento **CD=0**) então o conteúdo do **Buffer** (a célula) é transmitido para o **MCOA** quando o sinal **RxReady** estiver em nível alto (i.e., **RxReady=1**). A Figura 16 mostra como este bloco está ligado com os outros elementos do sistema.

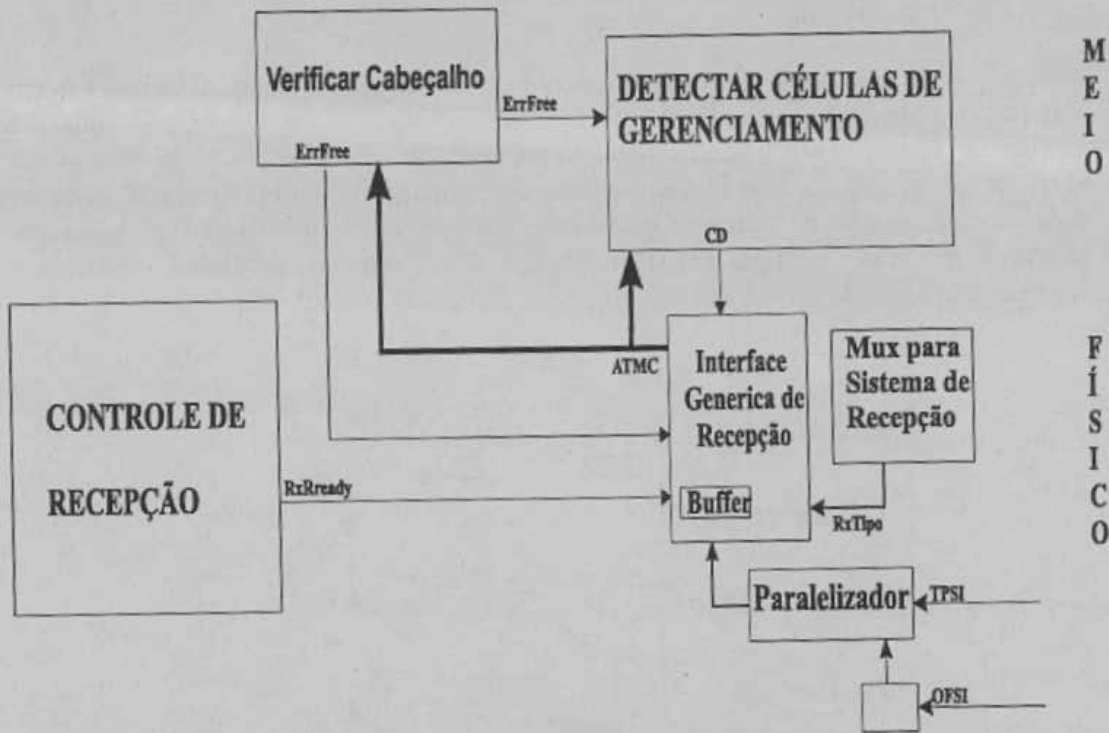


Figura 16: Interface Genérica de Recepção

#### 4. CONCLUSÃO

Apresentamos no presente documento uma arquitetura flexível e modular de uma Interface de Linha Física para o Comutador ATM (COMATM). Também propusemos uma arquitetura mais genérica que pode servi de roteiro para implementação de interfaces de linha física para comutadores ATM usando os mais variados sistemas de transmissão.

No caso específico do COMATM utilizaremos o par trançado como meio de transmissão por ele ser o mais utilizado nas redes locais. Com a proposta feita da arquitetura de uma ILF genérica poderemos futuramente adaptar o COMATM a uma outra tecnologia emergente ou dominante.

O projeto da ILF encontra-se com sua especificação técnica concluída, e já tem alguns blocos em fase avançada de implementação. Está sendo utilizada a ferramenta CADENCE para essa implementação. Mas estudos também estão sendo feito para ver a viabilidade da implementação da ILF em FPGA; por que isto nos permitirá concluir de maneira mais rápida a implementação total da ILF.



## 5. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] \_ Yiwei T. Hou e outros. Overview Of Implementing ATM-Based Enterprise Local Area Network for Desktop Multimedia Computing. IEEE Communication Magazine - April 1996
- [2] \_ José Augusto Suruagy Monteiro. Rede Digital de Serviços Integrados de Faixa Larga (RDSI-FL). IX Escola de Computação - Recife - Agosto 1994.
- [3] \_ Giozza, William F.; Zerrouk, Belkacem. Arquitetura de Comutador ATM Usando O Roteador Rcube. Anais Do 12<sup>o</sup> SBRC, CURITIBA, 18-20 MAIO 1994.
- [4] \_ Silva, Hamilton S.; Lima, José A. G, Monteiro Suruagy, Giozza F.WilliamProjeto COMATM - Arquitetura Básica e Descrição Funcional do Comutador ATM - Versão 3.0 -Relatório Técnico Rt-01.96 - Fevereiro de 1996
- [5] \_ The ATM Forum Technical Committee. UTOPIA, An ATM-PHY Interface Specification Level 1, Version 2.01- March 21,1994
- [6] \_Forum ATM - Utopia ATM Physical Interface Specification Level 2, Version 1.0 USA - Junho 1996
- [7] \_ TAXIChip(TM) Integrated Circuit - Advanced Micro Devices (<http://www.amd.com>)- 12/09/1995
- [8] AMCC ATM Products. SETI-S3005/SERI-S3006 SONET/SDH/ATM OC-3/OC-12 Transmitter and Receiver(<http://www.amcc.com>)
- [9] \_ Sargent Murray III, Richard L Shoemaker. The IBM PC(TM) From The Inside Out. Ed Addison Wesley Publishing