



UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
DEPARTAMENTO DE SEMICONDUTORES, INSTRUMENTOS E
FOTÔNICA.

*Desenvolvimento e prototipagem de um nó de acesso para redes
de chaveamento de pacotes ópticos*

Rodrigo Bernardo

Orientador:

Prof. Dr. Furio Damiani (FEEC)

Composição da banca examinadora:

Prof. Dr. Furio Damiani - FEEC/Unicamp - Presidente

Prof. Dr. Peter Jürgen Tatsch - FEEC/Unicamp

Dr. João Batista Rosolem - Fundação CPqD

Dissertação apresentada à Faculdade de Engenharia Elétrica e de Computação da
Universidade Estadual de Campinas, como parte dos requisitos exigidos para
obtenção do grau de MESTRE EM ENGENHARIA ELÉTRICA.

Campinas, 14 de dezembro de 2009.

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

B456d Bernardo, Rodrigo
Desenvolvimento e prototipagem de um nó de acesso
para redes de chaveamento de pacotes ópticos / Rodrigo
Bernardo. --Campinas, SP: [s.n.], 2009.

Orientador: Furio Damiani.
Dissertação de Mestrado - Universidade Estadual de
Campinas, Faculdade de Engenharia Elétrica e de
Computação.

1. Prototipagem. 2. Hardware. 3. Comutação de
pacotes. 4. FPGA. I. Damiani, Furio. II. Universidade
Estadual de Campinas. Faculdade de Engenharia Elétrica
e de Computação. III. Título.

Título em Inglês: Development and prototyping of an access node for optical
packet switching networks

Palavras-chave em Inglês: Prototyping, Hardware, Packet switching, FPGA

Área de concentração: Eletrônica, Microeletrônica e Optoeletrônica

Titulação: Mestre em Engenharia Elétrica

Banca examinadora: Peter Jürgen Tatsch, João Batista Rosolem

Data da defesa: 14/12/2009

Programa de Pós Graduação: Engenharia Elétrica

COMISSÃO JULGADORA - TESE DE MESTRADO

Candidato: Rodrigo Bernardo

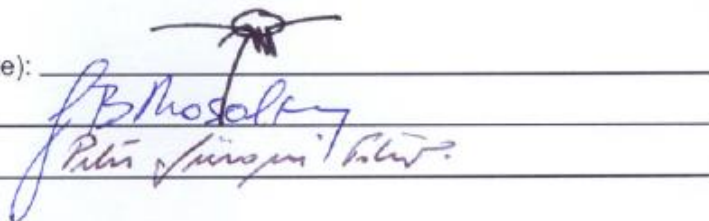
Data da Defesa: 14 de dezembro de 2009

Título da Tese: "Desenvolvimento e Prototipagem de um Nó de Acesso para Redes de Chaveamento de Pacotes Ópticos"

Prof. Dr. Furio Damiani (Presidente): _____

Dr. João Batista Rosolem: _____

Prof. Dr. Peter Jürgen Tatsch: _____



The image shows three handwritten signatures in blue ink over horizontal lines. The first signature is at the top, followed by the second, and the third at the bottom. The signatures are stylized and cursive.

Agradecimentos

Agradeço a Deus pela saúde na execução deste trabalho e a todas as pessoas que me ajudaram direta e indiretamente para a sua conclusão.

Ao CPqD pela estrutura propícia e aos gerentes Alberto Paradisi e Marcos Rogério Salvador por permitir fazer parte deste projeto.

Ao Eduardo Mobilon por acreditar no meu trabalho e me orientar profissionalmente.

Aos profissionais e amigos Luis, Ronaldo, Valentino, Vinícius, Giovanni (Eloim), Cláudio, Fábio, Rosolem, Vitor, Bruno, Júlio, Juliano, Akira e tantos outros que convivo no dia a dia e que me enriquecem de sabedoria e alegria.

A Universidade de Campinas, em especial aos professores Furio Damiani e Peter Jürgen Tatsch pela oportunidade, orientação e ensinamentos.

Ao FUNTTEL pelo financiamento da pesquisa aplicada.

Ao Henrique (CETUC) e Wlamir (Unisal) pelos esclarecimentos sobre FPGA e VHDL.

Aos professores da graduação Maurício José Bordon e Eli Carneiro Paiva pela recomendação.

Aos meus pais Servilio e Inês e a minha irmã Priscila que sempre estiveram ao meu lado.

A minha noiva Janaína pelo incentivo em todos os momentos.

Resumo

Este trabalho apresenta o desenvolvimento e a prototipagem de um nó de acesso utilizado como prova de conceito de redes de chaveamento de pacotes ópticos. Ele descreve as arquiteturas propostas para a rede e o nó de acesso, juntamente com o desenvolvimento detalhado do *hardware*, desde a concepção até os testes finais dos módulos (placas), e do núcleo de processamento implementado em dispositivo de lógica programável, que constitui a inteligência da rede. O nó de acesso foi concebido de forma modular, com quatro módulos desenvolvidos para compor o elemento principal da rede, cada um com tecnologia e função específica. O trabalho também descreve os testes realizados com os protótipos, demonstrando que os requisitos inicialmente propostos foram alcançados.

Palavras-chave: prototipagem, *hardware*, comutação de pacotes, FPGA.

Abstract

This work presents the development and prototyping of an access node for an optical packet switching network. The network's architecture and the access node proposals are described, as well as the detailed hardware development, from the conception to modules' (boards) final tests and core processing implemented on PLDs, which constitutes the intelligence of the network. The access node was conceived in a modular way, with four modules developed to compose the main element of the network, each with its proper technology and function. The work also describes the tests performed on the prototypes, showing that the proposed requirements were met.

Keywords: *prototyping, hardware, packet switching, FPGA.*

Índice

Agradecimentos	v
Resumo	vii
Abstract	vii
Índice	ix
Capítulo 1 Introdução	1
1.1 Técnicas de chaveamento no domínio óptico	1
1.2 Divisão do trabalho	3
1.3 Trabalhos publicados	4
Capítulo 2 A rede de chaveamento de pacotes ópticos (RCPO)	5
2.1 Motivação do projeto RCPO	5
2.2 Arquitetura da rede	7
2.2.1 Serviço de transporte	9
2.2.2 Controle de acesso ao meio	11
2.3 Arquitetura do nó de acesso da rede	15
2.3.1 Unidade de controle	16
2.3.2 Unidade de transporte de dados	18
Capítulo 3 Concepção do protótipo	23
3.1 Hardware	23
3.1.1 Domínio elétrico	23
3.1.2 Domínio óptico	27
3.1.3 Tecnologias adotadas	28
3.2 Modularidade	32
3.3 Lógica de processamento	33
3.3.1 Unidade de controle e barramento de comunicação	34
3.3.2 Unidade de transporte de dados	35
Capítulo 4 Desenvolvimento do protótipo	37
4.1 Placa Processadora	37
4.1.1 Estudo estrutural do protótipo do nó da rede	41

4.2	Placa de Acesso aos Clientes	43
4.2.1	Projeto elétrico da Placa de Acesso aos Clientes.....	43
4.2.2	Layout da Placa de Acesso aos Clientes	53
4.2.3	Montagem e testes da placa	61
4.3	Placa de Acesso ao Anel Óptico	65
4.3.1	Projeto elétrico da Placa de Acesso ao Anel Óptico.....	65
4.3.2	Layout da Placa de Acesso ao Anel Óptico.....	70
4.3.3	Montagem e testes da placa	72
4.4	Placa de Chaveamento Óptico	74
4.4.1	Projeto elétrico da Placa de Chaveamento Óptico.....	75
4.4.2	Layout da Placa de Chaveamento Óptico	78
4.4.3	Montagem e testes da placa	79
Capítulo 5	Núcleo de processamento, integração e resultados	81
5.1	Descrição dos blocos do núcleo de processamento	81
5.1.1	Bloco Adaptador Ethernet	82
5.1.2	Bloco Classificador de entrada	84
5.1.3	Bloco Montador de rajada de entrada e de saída	84
5.1.4	Bloco Transmissor de rajada.....	86
5.1.5	Bloco Receptor de rajada.....	87
5.1.6	Bloco Classificador de saída.....	87
5.2	Integração	88
5.2.1	União dos blocos.....	88
5.2.2	Teste do nó da rede	90
5.3	Resultados	97
Capítulo 6	Considerações finais	101
6.1	Trabalhos futuros.....	103
Referências bibliográficas	105	
Índice de figuras	109	
Índice de tabelas	113	
Lista de acrônimos	115	
Apêndice I - Código em C desenvolvido para a Placa de Acesso aos Clientes ..	119	

Capítulo 1

INTRODUÇÃO

A convergência das aplicações, dos serviços, das mídias e das redes especializadas para a Internet está mudando o mundo e o jeito como as pessoas fazem suas coisas. Embora com limitações, computação distribuída, teleconferência, tele-educação, rádio, televisão e jogos já são possíveis na Internet. Num futuro próximo, essas limitações desaparecerão e será possível suportar aplicações bem mais sofisticadas e complexas na Internet, tais como realidade virtual e tele-presença.

Para que este futuro se torne realidade, é necessário que as arquiteturas das redes de telecomunicações evoluam para um novo paradigma que abranja todas as camadas de transporte, de serviço e de gerência.

Esta evolução deverá começar pelas redes metropolitanas e de acesso, pela proximidade física e lógica dessas com os usuários finais. Nessas condições, as necessidades e a dinâmica das aplicações se fazem mais presentes do que nas redes de longa distância, nacionais e internacionais, onde a agregação de tráfego ocorre e suaviza os impactos das aplicações que requerem transmissão em altas taxas.

Podemos citar o gargalo eletrônico como uma grande barreira para fornecimento de altas taxas de dados aos usuários finais. Este se dá pela impossibilidade tecnológica atual de efetuar o processamento dos dados no domínio elétrico na mesma velocidade com que eles são transportados no domínio óptico.

1.1 Técnicas de chaveamento no domínio óptico

Para evitar ao máximo o processamento eletrônico, algumas técnicas de chaveamento no domínio óptico foram definidas com o conceito de processar a informação transmitida apenas no nó destino, evitando-o nos nós intermediários. São elas: chaveamento de rajadas ópticas (*Optical Burst Switching* - OBS) [1] [2] [3], chaveamento de pacotes

ópticos (*Optical Packet Switching* - OPS) [4] [5] e chaveamento de circuito (*Optical Circuit Switching* - OCS) [6].

As redes que utilizam OCS são caracterizadas por efetuar a transmissão dos dados somente após o estabelecimento de um caminho óptico. O estabelecimento deste ocorre mediante a configuração das matrizes de comutação ópticas nos nós onde passará o tráfego. O envio de uma mensagem de solicitação e confirmação é feita e requisitada pelo transmissor. As mensagens de sinalização trafegam em um meio exclusivo, e são processadas eletronicamente em cada nó ao longo do caminho. Uma vez transmitida, o tráfego segue o caminho óptico pré-estabelecido sem processamento ou conversão nos nós de rede intermediários. Ao término da transmissão, o caminho deve ser desfeito.

Já as redes que utilizam OBS são similares às que usam OCS, entretanto, esta técnica se diferencia por não precisar de mensagem de confirmação de estabelecimento de caminho óptico. Assim, o nó transmissor periodicamente agrega um número de pacotes cujo tamanho total seja inferior ao tamanho da rajada acordada na rede e envia uma mensagem de solicitação de caminho óptico em direção ao nó receptor. Ciente do tempo de processamento da mensagem em cada nó, o transmissor envia a mensagem de reserva de caminho e inicia a transmissão. Após o término do tempo da rajada o caminho óptico é automaticamente destruído. A principal diferença entre as redes OCS e OBS está na eficiência da utilização do meio. Na primeira, o caminho é criado porém não é certo se o tráfego gerado será constante por todo o tempo que o caminho existir. Na segunda, o caminho é criado para dar vazão a um tráfego que já está aguardando pela transmissão, introduzindo atraso por esta espera.

A rede que utiliza OPS pode ser vista como uma evolução das que usam OBS. Nesta a sinalização de reserva de caminho trafega em conjunto com os dados. Ao chegar em um nó, a sinalização é removida e convertida para o domínio elétrico, sendo analisada por um processador que, baseado nas informações recebidas, atuará em uma matriz de chaveamento óptico de forma a encaminhar os dados corretamente. Na saída do nó, uma nova sinalização contendo a informação necessária para o chaveamento no próximo nó será adicionada à frente do pacote de dados. Durante o tempo de processamento da sinalização no nó, os dados circulam por um atraso de fibra até a matriz de chaveamento

ser propriamente configurada. A rede que utiliza OPS difere das redes com OCS e OBS por não existir a alocação de um caminho fim a fim. Em cada nó os dados serão chaveados individualmente até atingir o seu destino final.

A Rede de Chaveamento de Pacotes Ópticos (RCPO) [7] proposta neste trabalho é um passo nesta direção. Com foco bem definido nas camadas de transporte, o seu objetivo é conceber uma tecnologia de rede óptica transparente, viável tecnicamente e com potencial para assumir posição de destaque na próxima transição tecnológica das arquiteturas de redes de telecomunicações de acesso e metropolitana. O projeto especifica de forma preliminar a arquitetura e as funcionalidades da rede e do nó de acesso. Esta arquitetura combina os benefícios das técnicas OCS e OBS com OPS para oferecer capacidade, desempenho e flexibilidade na adaptação a mudanças dinâmicas.

1.2 Divisão do trabalho

A contribuição deste trabalho para o tema está no desenvolvimento de uma plataforma para prova de conceito, ou seja, a construção de um protótipo constituído por módulos com funções específicas, capaz de suportar as técnicas inerentes de chaveamento de pacotes ópticos. O trabalho apresenta, em 6 capítulos, a concepção arquitetônica da rede, a construção e testes das placas e a montagem de um protótipo de um nó de acesso para operar na configuração de anel e prover as características definidas da rede.

No Capítulo 2 a proposta arquitetônica da rede e conseqüentemente do nó de acesso são apresentadas. A descrição da arquitetura da rede se refere ao tipo de transporte, como classe e qualidade de serviço, e quais são os protocolos utilizados para o controle de acesso ao meio. Com relação ao nó de acesso da rede, são descritos como os blocos que fazem o controle, classificação e transporte dos pacotes de dados na lógica de processamento devem operar em conjunto com um plano de controle.

O Capítulo 3 descreve como o protótipo foi concebido seguindo os requisitos definidos no capítulo anterior. São mostradas as tecnologias adotadas para o desenvolvimento dos módulos (placas) e como eles foram denominados para cada função. Também é apresentada a arquitetura da lógica de processamento a ser desenvolvida num

componente de lógica programável.

O Capítulo 4 apresenta detalhadamente o desenvolvimento de todas as placas que compõem o nó da rede. São descritas as técnicas utilizadas para o desenvolvimento do projeto elétrico, as funcionalidades dos principais componentes e circuitos, técnicas de *layout* como posicionamento e roteamento, fabricação e montagem das placas e também o resultado dos testes realizados.

O Capítulo 5 traz a descrição dos blocos do núcleo de processamento, da integração das placas e da inteligência do nó de acesso e também os resultados alcançados com os testes realizados. Também são apresentados os problemas encontrados e as soluções adotadas com a integração do *hardware* e da lógica de processamento.

O Capítulo 6 apresenta as considerações finais do projeto, contemplando o custo do protótipo e os trabalhos futuros sugeridos.

No Apêndice I foi incluído o código desenvolvido em C para o microcontrolador da Placa de Acesso aos Clientes.

1.3 Trabalhos publicados

A partir das propostas deste trabalho foram publicados vários artigos, entre os quais “*Hardware Design and Prototype of an Optical Packet-Switched Ring Network Node*”, apresentado no 49th IEEE International Midwest Symposium on Circuits and Systems [8]. “*Hardware Architecture for Optical Packet and Burst Switching Applications*”, apresentado no 6th International Telecommunications Symposium [9]. “Arquitetura e Protótipo de Rede de Pacotes em Anel com Transmissão e Comutação no Domínio Óptico”, apresentado no 24º Simpósio Brasileiro de Redes de Computadores [7]. Também foram publicados dois artigos no caderno de tecnologia do CPqD. São eles: “*Protocolo de controle de acesso do anel de rede de comutação óptica de pacotes*” [10] e “*Rede Óptica de Pacotes de Serviços Avançados*” [11].

Capítulo 2

A REDE DE CHAVEAMENTO DE PACOTES ÓPTICOS (RCPO)

2.1 Motivação do projeto RCPO

O objetivo principal do projeto da RCPO foi pesquisar, desenvolver e demonstrar uma solução para rede óptica metropolitana [12] e inovadora, fundamentada no chaveamento óptico e na reconfiguração dinâmica em alta velocidade. Esta solução deve ser capaz de satisfazer as necessidades das aplicações multimídia, transportar grandes volumes de tráfego com baixa latência e adaptar-se às mudanças abruptas e dinâmicas de perfil de tráfego eficientemente.

Dado que o projeto tem como objetivos estratégicos a geração de patentes e a transferência de tecnologia de ponta, foram adotadas as soluções técnicas que possibilitem a evolução do protótipo do nó de acesso (através de técnicas de engenharia de produto) para um equipamento comercial que possa ser fabricado e comercializado pela indústria nacional.

O desenvolvimento das atividades de projeto do nó de acesso da RCPO foi iniciado a partir da análise de todos os requisitos previstos no projeto e de quais poderiam ser implementados num protótipo de laboratório.

Como qualquer arquitetura de rede, a da RCPO deve seguir as diretrizes genéricas da Tabela 2.1.

Realizamos reuniões técnicas com os membros da equipe no Centro de Pesquisa e Desenvolvimento em Telecomunicações (CPqD) para avaliação da tecnologia envolvida no projeto para adotar soluções que caracterizam o estado da arte em eletrônica e optoeletrônica e que viabilizam a evolução do protótipo para um equipamento comercial,

atendendo assim às restrições técnico-econômicas inerentes.

Requisitos	Objetivos
Desempenho	Atender a demanda de tráfego atual com baixa latência independentemente da característica e padrão de distribuição do tráfego.
Multi-serviços	Suportar múltiplos serviços para satisfazer os requisitos das várias aplicações.
Flexibilidade	Suportar novos serviços sem mudanças na arquitetura.
Expansibilidade	Crescer em cobertura, em número de nós e em taxa de transmissão sem mudanças na arquitetura.
Simplicidade	Facilitar e baratear o projeto, a manutenção, a operação e a evolução da rede.
Robustez	Recuperar-se de falhas variadas rapidamente.
Eficiência	Usar os recursos da rede eficientemente, independentemente da característica e padrão de distribuição do tráfego.
Segurança	Dificultar o acesso às informações em trânsito e facilitar a inclusão de mecanismos de criptografia.
Custo	Ter baixo custo para facilitar sua aceitação.

Tabela 2.1 - Diretrizes genéricas para arquiteturas de rede

Embora de grande importância, baixo custo não está entre as prioridades da arquitetura da RCPO por esta utilizar tecnologia recente e ser baseada em componentes com custo consideravelmente alto.

Para satisfazer aos requisitos da tabela acima, a arquitetura da rede deve:

- ser aberta e modular, visando flexibilidade, expansão e evolução;
- se basear na técnica da multiplexação por divisão em comprimentos de onda (*Wavelength Division Multiplexing* - WDM) [13] visando capacidade e expansão;
- usar comunicação do tipo *single-hop* (sem qualquer conversão entre os domínios ópticos e elétricos durante o trânsito) visando desempenho, segurança,

transparência de protocolo de transmissão, de taxa de transmissão e de formato de sinal e conseqüente suporte a multi-serviços;

- suportar comunicação do tipo um-para-muitos no domínio óptico visando eficiência e desempenho tanto quantitativo quanto qualitativo;
- ter controle distribuído visando robustez.

2.2 Arquitetura da rede

A RCPO, denominada BlasteRing (*Burst and lambda services for a transparent and Ethernet services Ring*) [7], é uma arquitetura de rede em anel, como ilustrado na Figura 2.1, que combina as técnicas OCS e OBS com OPS apoiando-se em tecnologias ópticas de redes disponíveis comercialmente.

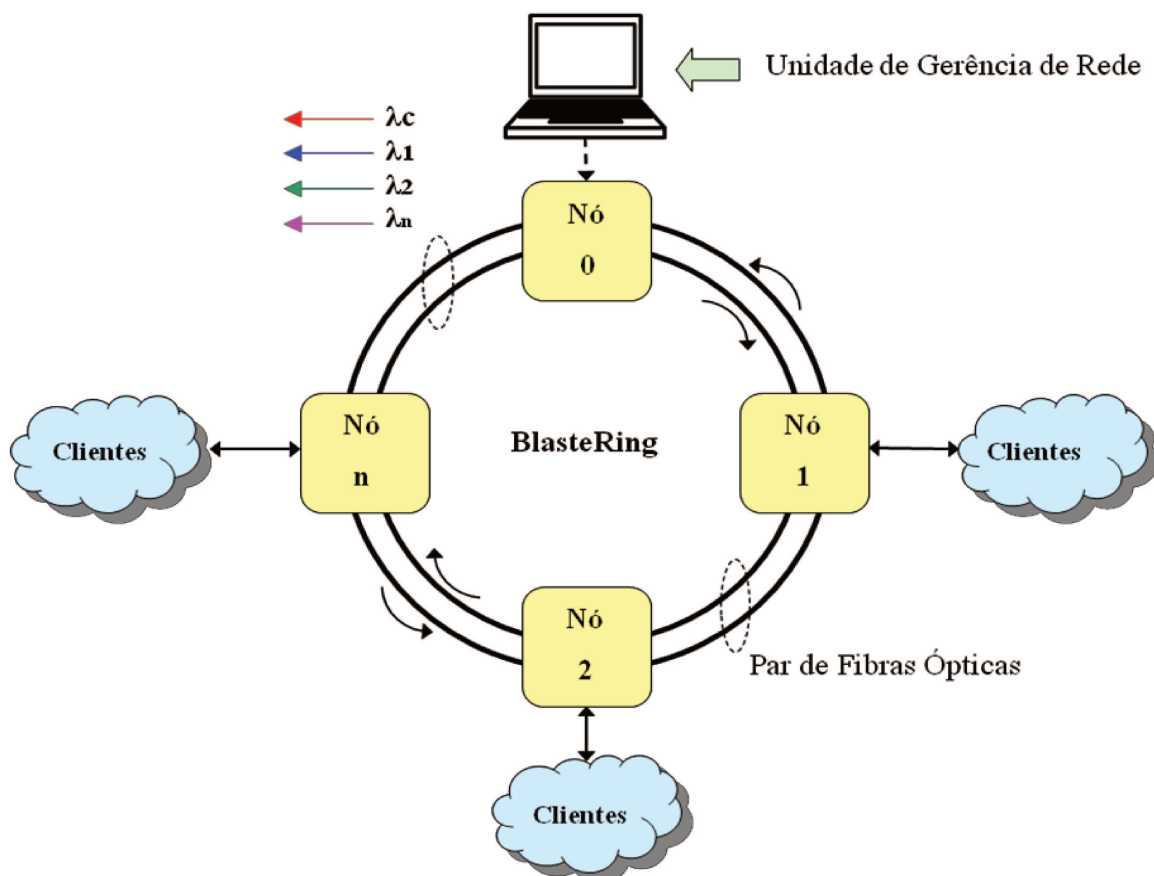


Figura 2.1- Macro arquitetura da RCPO

A RCPO consiste de um anel de chaveamento de pacotes ópticos de múltiplos comprimentos de onda (*Multiple-wavelength Optical Packet Switching* - MOPS) [14] com acesso por *slots* de tamanho fixo, onde um *slot* pode levar apenas um pacote óptico por vez. O pacote óptico é de tamanho variável, até o máximo dado pelo tamanho do *slot*. Uma vez transmitido, um pacote óptico é convertido para o domínio elétrico apenas no seu nó de destino, caracterizando a RCPO como uma rede óptica *single-hop*. A rede também suporta *multicast* (pacotes de dados com mais de um destino) realizado no domínio óptico. São utilizados um ou mais pares de fibras ópticas, com comprimento não superior a 100 km, dada a sua aplicação em rede metropolitana. Cada fibra óptica usa dois ou mais comprimentos de onda (λ) unidirecionais, com mesmo espaçamento utilizado por equipamentos WDM [13], utilizando um ou mais λ s na faixa de 1550 nm (especificado pelo comutador óptico), para o transporte de dados (λ_n) e outro λ na faixa de 1310 nm ou na de 1550 nm, para o transporte das informações de controle (λ_c).

Para todo par de fibras ópticas, o sentido de transmissão numa fibra deve, necessariamente, ser contrário ao sentido de transmissão da outra fibra que compõem o par. Um nó de rede não necessariamente faz acesso a todos os comprimentos de onda de dados de um anel. Por motivos de simplicidade na operação e custo no desenvolvimento, são utilizadas apenas unidades de transmissão e recepção fixas.

A rede suporta $N \leq 16$ nós, sendo um dos nós responsável pela inserção dos cabeçalhos na fase de inicialização. Os cabeçalhos contêm as informações de disponibilidade dos *slots*, e são projetados de forma a buscar otimização no acesso dos recursos da rede pelos nós. A subseção 2.2.2 descreve em mais detalhes a questão do uso dos cabeçalhos e o controle de acesso ao meio.

A RCPO opera com três planos funcionais bem definidos: gerência, controle e transporte de dados. O plano de gerência é centralizado e responsável pela coordenação da rede de uma forma geral, atuando diretamente no plano de controle utilizando para isto o plano de transporte de dados. O plano de controle é distribuído e atua diretamente nos elementos do plano de transporte de dados a fim de realizar a sua função principal, o controle de acesso ao meio. O plano de transporte de dados também é distribuído, sendo responsável pelo serviço de transporte, tanto no domínio elétrico quanto óptico. Através

dele os clientes enviam os seus dados, que serão classificados na entrada do nó e enviados para filas de transmissão, de acordo com os seguintes critérios: o nó de acesso que se pretende alcançar, a classe de serviço e a qualidade de serviço do tráfego. Após a classificação, os dados são agregados em rajadas que serão traduzidas em pacotes ópticos para a transmissão. O objetivo da agregação é otimizar a utilização do *slot* através de um pacote óptico que se aproxime ao máximo do tamanho do *slot*. Em contrapartida, a agregação aumenta o tempo médio de espera para transmissão dos pacotes, o que não é suportado por muitos tipos de demanda de tráfego. Assim, o tempo de montagem de uma rajada deve ser parametrizado de acordo com a necessidade. Cada nó de acesso da rede é constituído por uma unidade de transporte de dados (UTD) e uma unidade de controle (UC). A Figura 2.2 ilustra a relação entre esses planos.

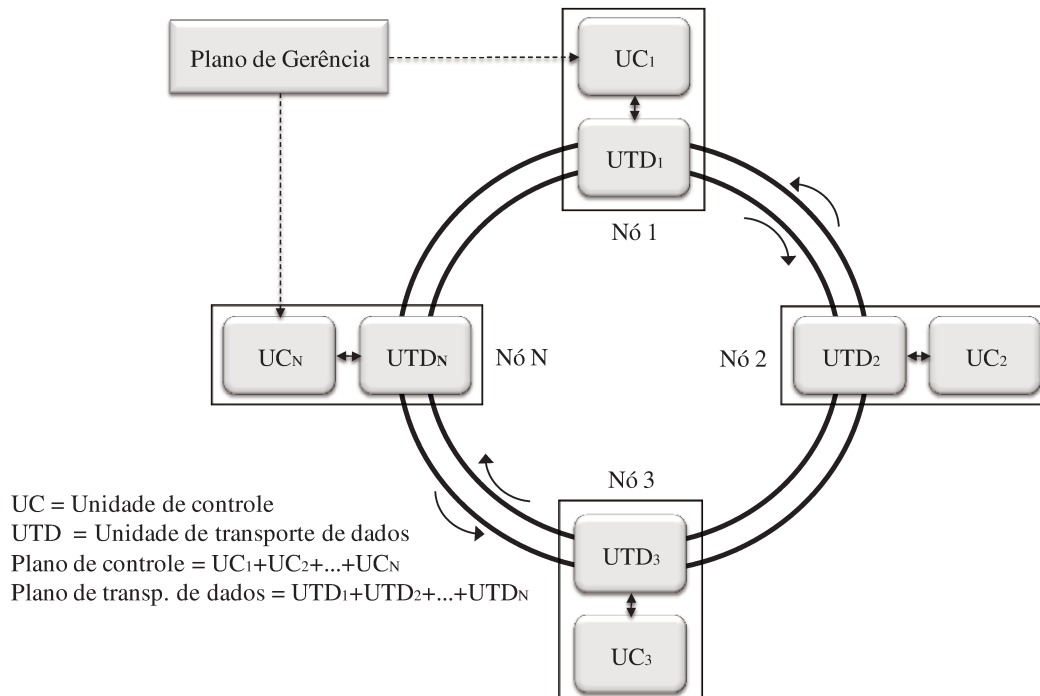


Figura 2.2 - Relacionamento entre os planos funcionais

2.2.1 Serviço de transporte

Uma característica da arquitetura da RCPO é a grande flexibilidade e facilidade que ela apresenta com relação à diferenciação e o tratamento do tráfego, permitindo a implementação de três diferentes tipos de classe de serviço (*Class of Service - CoS*), sendo elas: melhor esforço, reservada e transparente.

O serviço de transporte de melhor esforço consiste no compartilhamento de um dado número de *slots* entre os nós da rede, tendo todos os nós o mesmo direito de acesso a esses *slots*, sendo assim considerado “justo” para todo o tipo de demanda. Essa classe de serviço não garante banda pré-definida para nenhum fluxo de dados, e nem tempo de latência máxima. Entretanto, é possível realizar a priorização do tráfego através de diferentes tipos de qualidade de serviço (*Quality of Service* - QoS). Para tal, as filas de transmissão são divididas em sub-filas, tendo cada uma delas um diferente valor de QoS. A diferenciação no tratamento das sub-filas ocorre através do ajuste de tempo de montagem da rajada e na prioridade de transmissão, controladas por parâmetros definidos pelo plano de controle da rede. A Figura 2.3 ilustra o conceito. Neste exemplo existem quatro sub-filas para cada fila de transmissão, sendo associada uma fila de transmissão para cada nó destino. A sub-fila com valor de QoS mais prioritário possui tempo de montagem de rajada menor que as demais, e, desta forma, mesmo que os dados recebidos não sejam suficientes para preencher 100% da capacidade do *slot*, é solicitada a transmissão assim que o tempo de montagem for vencido. Isso obviamente gera uma ineficiência na utilização do canal, porém diminui o tempo de latência, de forma a tentar atingir os requisitos que um dado serviço demanda.

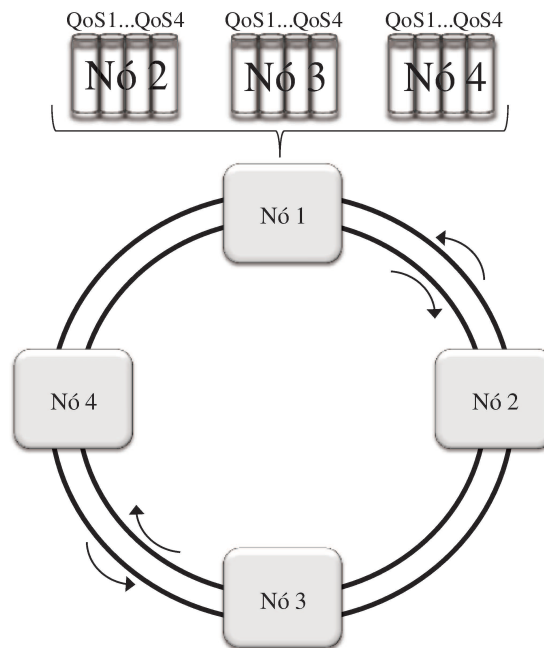


Figura 2.3 - Classe de serviço de melhor esforço

Além do tempo de montagem mais baixa, uma rajada mais prioritária deve ter preferência na transmissão quando outras rajadas concorrem por um mesmo *slot*.

Essa classe de serviço é definida como de melhor esforço pois não há como garantir o tempo máximo de espera pela transmissão, nem banda mínima, sendo esses valores diretamente relacionados com o tráfego total carregado pela rede versus sua capacidade.

A segunda classe de serviço suportada pela RCPO é a reservada. Com essa classe é possível garantir o tempo de acesso e a banda de forma pré-determinada. A implementação ocorre, como o próprio nome indica, com a reserva de certo número de *slots* para um fluxo. Essa reserva ocorre com a utilização de tabelas na unidade de plano de controle. De forma similar à CoS de melhor esforço, esta também permite a separação do seu tráfego em diferentes tipos de QoS, de forma a priorizar os dados dos serviços que demandam uma menor latência.

A última classe de serviço suportada pela RCPO é a transparente. Sua realização é possível dada a uma peculiaridade da rede: toda a sinalização do plano controle ocorre através de um único canal, que é dedicado para esse fim. Desta forma, quando não ocorre transmissão em nenhum *slot* de um canal num segmento da rede, não existe nenhum sinal neste comprimento de onda, permitindo o transporte de qualquer tipo de informação e em qualquer taxa, alcançando a mesma transparência dos sistemas WDM. Essa característica, típica das redes de chaveamento de comprimento de onda, também é utilizada na RCPO, realizando neste caso o chaveamento do comprimento de onda baseado na solicitação do cliente, sem qualquer necessidade de analisar os campos de endereçamento dos quadros transmitidos.

2.2.2 Controle de acesso ao meio

O protocolo adotado para a realização do controle de acesso ao meio (*Medium Access Control* - MAC) na RCPO é de nível médio tanto em desempenho e implementação, e baseia-se na combinação de versões modificadas dos protocolos de transporte agregado de pacotes (*Packet Aggregated Transport* - PAT) [15] e *Satisfied* (SAT) [16].

O protocolo PAT responde pelo controle de acesso ao meio propriamente dito. Este protocolo utiliza a mesma abordagem da RCPO, onde a capacidade total do anel é dividida em *slots* de tempo com duração fixa, podendo cada *slot* transportar um único

pacote óptico que se traduz num ou mais pacotes elétricos. É importante notar que uma arquitetura de rede OPS [4] [5] original define um pacote óptico como um meio capaz de transportar um único pacote elétrico. Entretanto, para atender esse requisito seriam necessários pacotes ópticos com duração da ordem de algumas dezenas de ns, inviabilizando qualquer tipo de implementação, dada à necessidade de se processar eletricamente o cabeçalho do pacote óptico. O protocolo PAT propõe essa abordagem de agregação de pacotes elétricos; entretanto, com PAT original a solicitação da transmissão do pacote eletrônico ocorre assim que este chega a uma fila vazia, agregando mais pacotes à fila enquanto a transmissão não for liberada. Porém, mesmo que não cheguem outros pacotes elétricos, ou que alguns outros poucos cheguem, mas não atinjam a capacidade de transmissão do pacote óptico, este é alocado da mesma forma, gerando assim uma subutilização do recurso.

Por esse motivo, a modificação do protocolo PAT proposto para a RCPO, chamado de PAT+, utiliza o conceito de montagem de rajada das redes OBS, visando uma melhor utilização dos *slots*. Nesta abordagem, uma solicitação de transmissão ocorre dada a necessidade de se transmitir uma rajada já formada. Isso resulta num melhor rendimento de transmissão, e na possibilidade de se utilizar pacotes ópticos de maior duração sem impactar demasiadamente a eficiência de utilização dos canais.

Na RCPO os *slots* de cada canal são sincronizados de forma a chegarem juntos num nó, possuindo um cabeçalho compartilhado que é convertido para o domínio elétrico e processado no nó da rede. Através do processamento do cabeçalho são determinadas as ações sobre os *slots*, sendo divididas em quatro casos: o pacote pode estar livre, podendo ser utilizado para a transmissão; o pacote pode ter como destino o próprio nó, sendo realizada assim uma operação de *drop* (remoção do pacote do anel óptico); o pacote pode ser endereçado a um nó de acesso à frente, sendo realizada uma operação de *forward* (encaminhamento do pacote para o anel óptico); o pacote pode conter um endereço de *multicast* (o pacote é endereçado para mais de um nó da rede), neste caso são realizadas as duas operações anteriores de forma simultânea. Para evitar que alguns nós da rede tenham mais oportunidades de acesso que outros, é utilizada uma versão modificada do protocolo SAT. No protocolo original, um sinal de controle denominado

satisfied (satisfeito), implementado por um bit no cabeçalho dos *slots*, circula pelos nós do anel atribuindo quotas de transmissão (em número de *slots*) pré-determinadas. No caso do nó receber o campo SAT com valor 1 e possuir algum dado para transmitir e ainda não tiver utilizado o seu número predeterminado de *slots*, este irá reter o SAT (atualizando em ‘0’ o campo no cabeçalho que o trouxe) até que não haja mais dados para transmitir ou que tenha sido utilizada a sua cota de *slots*.

A não liberação do SAT impede que os outros nós tenham suas quotas renovadas, fazendo com que em algum momento parem de transmitir e desta forma permitindo que o nó que possui o SAT tenha oportunidade para utilizar sua quota. Uma vez satisfeito, o nó libera o SAT (atualizando em ‘1’ o bit apropriado do primeiro cabeçalho que aparecer).

O protocolo SAT alterado, denominado de SAT+, segue o mesmo princípio do PAT+, substituindo a abordagem de pacotes por rajadas.

A Figura 2.4 apresenta a estrutura básica do cabeçalho utilizada na RCPO.

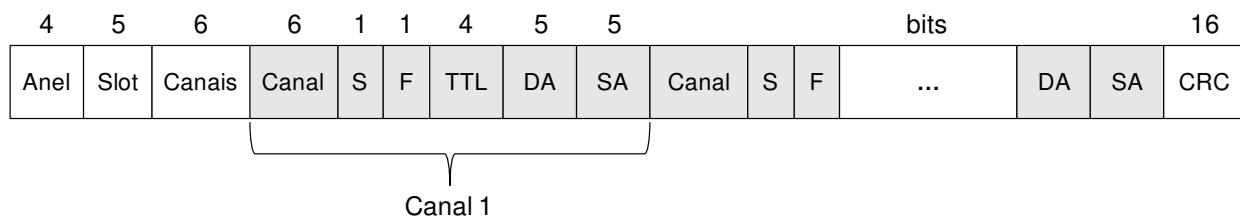


Figura 2.4 - Estrutura do cabeçalho

Anel: indica por qual anel o cabeçalho está circulando;

Slot: indica qual é o *slot* de tempo deste canal;

Canais: indica o número de canais descritos no cabeçalho;

Canal: indica a qual canal que os campos S, F, TTL, DA e SA se referem;

S (Status): indica se o slot está vazio ‘0’ ou ocupado ‘1’. Um canal vazio contém os campos S, DA e SA preenchidos com zeros;

F (Fairness): indica se o sinal SAT+ está presente ‘1’ ou ausente ‘0’;

TTL (Time-to-Live): indica o número máximo de nós (*hops*) que a rajada contida no *slot* do canal pode atravessar no anel. Cada nó da rede decrementa este campo toda vez que encaminha o *slot* para o próximo nó. Se o valor for zero, então o nó descarta a rajada contida no *slot* e marca o *slot* do canal como vazio;

DA (Destination Address): indica o destinatário daquele *slot*. Se *DA* for igual ao endereço do nó que está processando o *slot*, então a rajada é recebida e o *slot* é marcado como vazio, podendo ser reusado pelo próprio nó para a transmissão de uma nova rajada ou por nós subseqüentes. Também podem ser usados endereços de *multicast* e *broadcast*, uma vez que a rede possui esse suporte, pois é realizado no domínio óptico;

SA (Source Address): indica o remetente daquele *slot*. Se *SA* for igual ao endereço do nó que está processando o *slot*, então a rajada é descartada e o *slot* é marcado como vazio, podendo ser reusado pelo próprio nó para a transmissão de uma nova rajada ou por nós subseqüentes;

CRC (Cyclic Redundancy Check): calculado sobre o conteúdo de todos os outros campos, permite a cada nó determinar se o cabeçalho está íntegro. Ao receber o cabeçalho, todo nó calcula um CRC sobre o conteúdo do cabeçalho (excluindo o campo CRC) e o compara com o CRC contido no cabeçalho. Se forem diferentes, então a rajada é descartada (se houver uma) e o *slot* é marcado como vazio.

A Figura 2.5 apresenta a organização dos pacotes numa rajada elétrica, que pode ser denominada de pacote óptico. O campo Preâmbulo é necessário para permitir a sincronização do receptor, uma vez que a comunicação no canal óptico é comutada e não permanente. As informações são transmitidas em segmentos, cada um contendo um ou mais quadros Ethernet [17]. Todo segmento contém um campo com Bytes de preenchimento (*Bytes of Padding* - BP), um campo definindo o tamanho do campo BP (*Padding Size* - PS) e um último campo de verificação de integridade do quadro, o CRC. Com isso, mesmo que uma rajada sofra interferências que causem erro em alguns bits, nem todas as informações serão perdidas, apenas um ou alguns segmentos serão descartados.

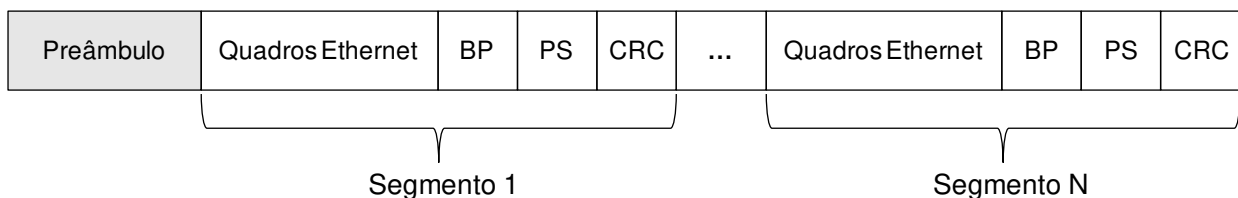


Figura 2.5 - Formato do pacote óptico

2.3 Arquitetura do nó de acesso da rede

Esta seção especifica a arquitetura do nó de acesso da RCPO que é dividido em dois blocos funcionais: Unidade de Controle (UC) e Unidade de Transporte de Dados (UTD), como mostrado na Figura 2.6. O conjunto das unidades de controle e transporte de dados, implementadas em todos os nós da rede, resultam, respectivamente, nos planos de controle e transporte de dados da RCPO.

Ao contrário desses planos, a gerência da rede é centralizada, sendo ligada diretamente a uma única unidade do plano de transporte de dados, podendo assim atingir através desse plano todas as demais unidades do plano de controle. Nesta arquitetura do nó de acesso, o comprimento de onda reservado para o fluxo dos cabeçalhos em cada um dos anéis é removido na entrada do nó através de um filtro, entrando diretamente na unidade de controle. Os demais comprimentos de onda seguem para a unidade de transporte de dados, passando antes por um *delay* (atraso) de fibra. Assim, os comprimentos de onda de dados serão atrasados em relação ao comprimento de onda de controle, tendo a unidade de plano de controle um maior tempo para processar o cabeçalho. Em paralelo ao processamento do cabeçalho, a unidade de transporte de dados recebe e classifica os quadros Ethernet enviados pelos clientes, montando a rajada eletrônica e solicitando a unidade de plano de controle a transmissão dos pacotes ópticos já montados.

Por sua vez, a unidade de controle analisa o cabeçalho, verificando se existem dados destinados a este nó e se há disponibilidade para a transmissão de seus próprios dados.

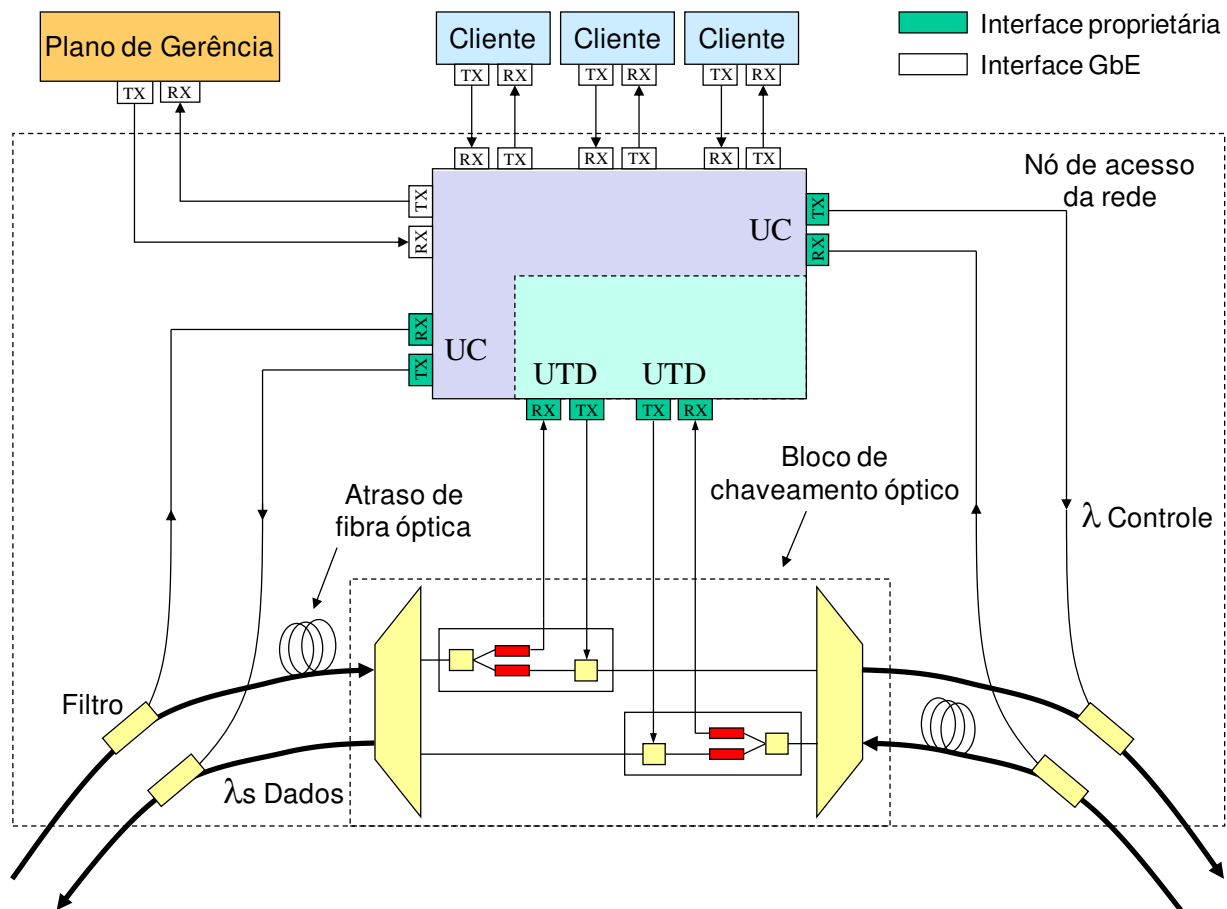


Figura 2.6 - Arquitetura do nó de acesso da rede

2.3.1 Unidade de controle

Como descrito na subseção 2.2.2, esta unidade de controle realiza fundamentalmente a função de controle de acesso ao meio através do processamento do cabeçalho dos *slots*. Para tal, esta é dividida em blocos com funções específicas, como mostrado na Figura 2.7.

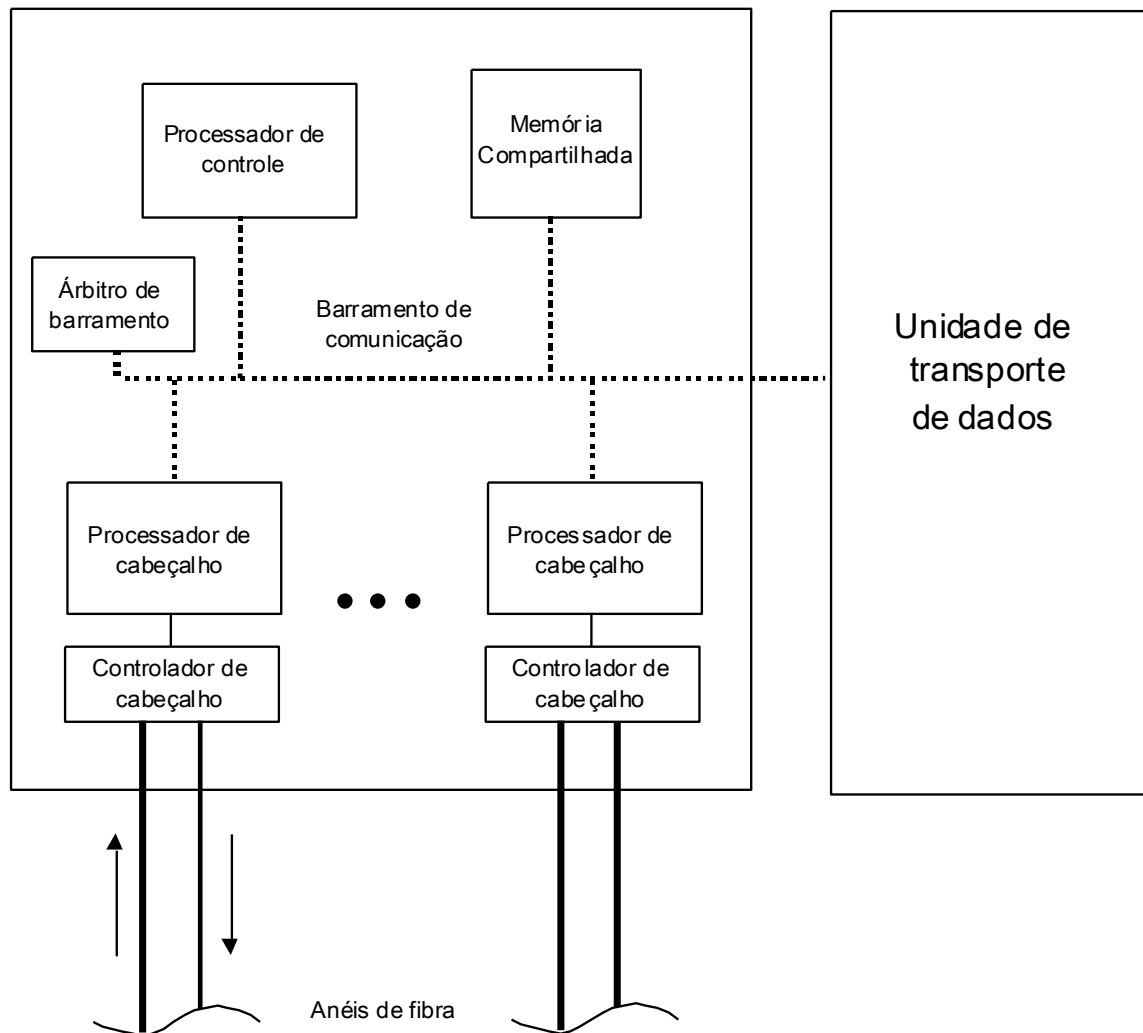


Figura 2.7 - Unidade de controle do nó de acesso da rede

O Processador de controle recebe comandos do plano de gerência via o barramento de dados através do plano de transporte de dados; esta comunicação será mais bem detalhada na próxima subseção. Através dos comandos da gerência, o Processador de controle organiza a tabela de alocação dos *slots* e a tabela de demandas, como explicado na subseção 2.2.1. Estas tabelas são armazenadas na memória compartilhada e serão utilizadas pelos processadores de cabeçalhos para as tomadas de decisão na alocação do *slot*.

O Processador de controle deve ainda realizar o mapeamento das filas ou grupos de filas de dados (controladas pela unidade de transporte de dados) com os nós da rede, mantendo uma tabela que traduz o destino que cada fila alcança.

O Processador de cabeçalho atua em conjunto com um Controlador de cabeçalho

dedicado ao seu uso, sempre existindo um par de processador e controlador por anel da rede. O controlador é um bloco necessário para garantir os requisitos de sincronização dos cabeçalhos. Nos processadores de cabeçalho são executados os protocolos PAT+ e SAT+, descritos na subseção 2.2.2. As questões de reinserção de cabeçalhos, modo de operação e recuperação de falha também são executados nestes processadores de forma individual para cada anel de fibra na rede.

Como o barramento é utilizado para toda a comunicação do nó, tanto entre a unidade de gerência e controle, quanto entre a unidade de controle e transporte de dados, e internamente na própria unidade de controle, torna-se necessária a utilização de um bloco chamado árbitro de barramento, que controlará todo o acesso entre os dispositivos pertencentes a esse barramento, evitando assim colisões resultantes das tentativas de dois ou mais periféricos tentarem acessar um mesmo recurso.

2.3.2 Unidade de transporte de dados

A unidade de transporte de dados se divide em dois blocos macros: o bloco eletrônico de classificação de quadros e montagem de rajadas, e o bloco de chaveamento óptico, já ilustrado na arquitetura do nó de acesso. Esses dois blocos em conjunto realizam todo o transporte de dados carregados pelo nó de acesso da RCPO.

O bloco eletrônico de classificação de quadros e montagem de rajadas, ou simplesmente bloco eletrônico, tem a função de classificar o tráfego dos clientes ligados a RCPO separando-o em filas e sub-filas de transmissão de acordo com o nó destino, a classe e a qualidade de serviço. Além desta, também são funções deste bloco: a montagem de rajadas de acordo com os critérios de tamanho e tempo definidos pelo plano de controle, a solicitação para o plano de controle da transmissão das rajadas prontas e a transmissão do pacote óptico. A Figura 2.8 apresenta os sub-blocos funcionais que compõem o bloco eletrônico.

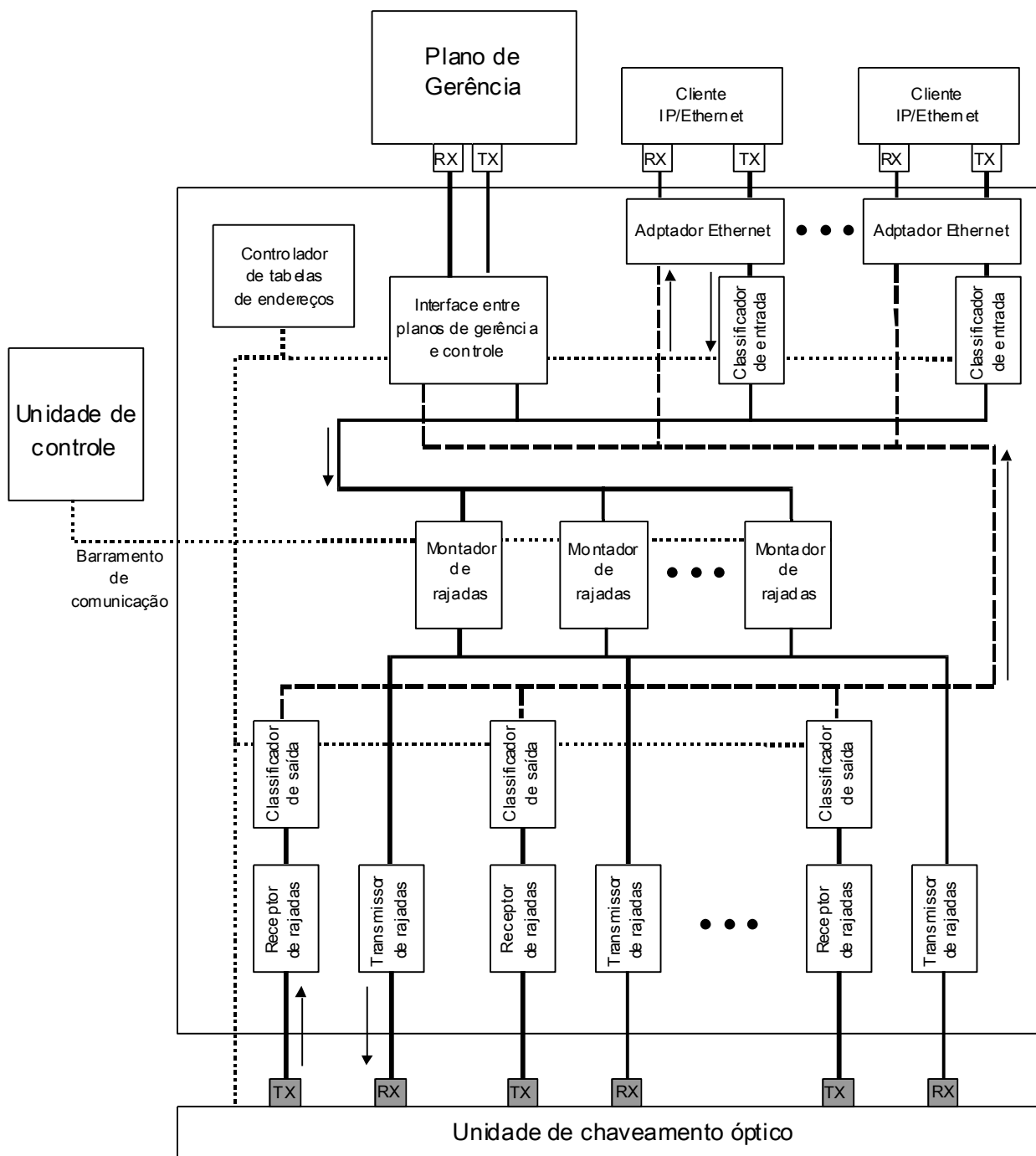


Figura 2.8 - Bloco eletrônico de classificação de quadros e montagem de rajadas do nó de acesso da rede

Os quadros dos clientes que entram no nó passam inicialmente pelo sub-bloco Adaptador Ethernet. Como o atual padrão Ethernet não prevê um campo apropriado que contenha o tamanho do quadro, que é variável, é necessário acrescentar esse campo ao quadro para facilitar o seu encaminhamento pelos próximos sub-blocos. O Adaptador Ethernet deve

então realizar toda a comunicação inerente ao protocolo Ethernet e acrescentar o campo extra contendo a informação do tamanho do quadro.

Após a adaptação, o quadro segue para o Classificador de entrada, que consulta o Controlador de tabelas de endereço a fim de decidir para qual fila de transmissão o quadro deve ser comutado. Para essa tomada de decisão diversos campos do quadro podem ser utilizados, como o endereço MAC de destino, os parâmetros de QoS e outros. Para realizar este encaminhamento de quadros, diversas abordagens podem ser seguidas, como a criação de uma tabela estática de encaminhamento. A implementação da política de encaminhamento de quadros é realizada pelo sub-bloco Controlador de tabelas de endereços, e a sua abordagem fica a critério do projeto de implementação.

Já propriamente classificado, o quadro segue para o Montador de rajada. Este sub-bloco controla a montagem da rajada eletrônica baseado nos critérios de tempo e tamanho de cada fila. Quando uma fila atinge um desses dois critérios de montagem, o sub-bloco solicita à unidade de plano de controle sua transmissão. Assim, quando houver algum *slot* disponível, a unidade de controle enviará a liberação da transmissão ao montador de rajada que passará todos os quadros que compõem essa rajada ao sub-bloco Transmissor de rajadas apropriado.

Deve existir um sub-bloco de Transmissão de rajadas para cada canal de dados suportado pelo nó. Este sub-bloco deve preparar a rajada eletrônica de forma a constituir o pacote óptico visto na subseção 2.2.1.

Para a recepção de pacotes ópticos, o processo inicia-se na chegada de um cabeçalho que contenha a informação de que um *slot* carrega dados destinados a este nó, assim, após a atuação da unidade de controle no bloco de chaveamento óptico (detalhado mais a frente), os dados chegarão ao Receptor de rajadas que removerá a formatação realizada pelo Transmissor de rajadas, passando os quadros Ethernet ao Classificador de saída, que consultará o Controlador de tabelas de endereços para decidir a qual cliente o quadro será encaminhado.

Antes de chegar ao cliente, o quadro passa novamente por um Adaptador Ethernet, que removerá o campo de tamanho, tornando-o um quadro Ethernet padrão.

Além de classificar, comutar e transmitir o tráfego dos clientes, o plano de transporte de

dados é utilizado ainda para o fluxo de informações entre a unidade de gerência e as unidades de controle. Para tal, a unidade de gerência se conecta a uma unidade de transporte de dados através do sub-bloco Interface entre planos de gerência e controle. Este sub-bloco se interliga com a unidade de controle do nó em que reside através do barramento de comunicação, e com as demais unidades de controle da rede através do plano de transporte de dados, sendo estes dados encaminhados a uma fila de transmissão que atinja o nó desejado da mesma forma como ocorre com os dados dos clientes.

Outro elemento que compõem a unidade de transporte de dados é o bloco de chaveamento óptico. É formado por elementos ópticos que dão uma grande flexibilidade nas operações de chaveamento de comprimentos de onda. O principal elemento é o comutador óptico, apresentado na Figura 2.9. O comutador é composto por três chaves e dois acopladores também ópticos. Por se tratar de uma rede de topologia em anel este comutador deve permitir a execução de três operações: adição, remoção e passagem.

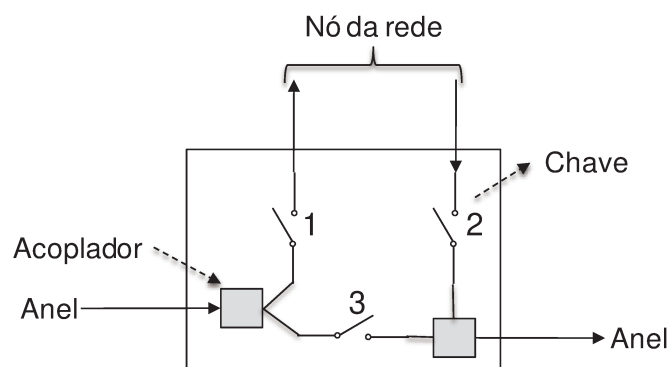


Figura 2.9 - Comutador óptico do nó de acesso da rede

Desta forma, quando um *slot* contém um pacote destinado ao nó, a unidade de controle deve preparar o comutador óptico do canal deste *slot* para a operação de remoção. Nesta operação fecha-se a chave 1, permitindo que os dados fluam para dentro do nó, e abre-se a chave 3, impedindo que os dados continuem a circular pelo anel. No caso contrário, quando o nó transmite um pacote óptico, fecha-se a chave 2 e abre-se a chave 3. É importante notar que ambas as operações podem ser realizadas simultaneamente, assim, quando um nó remove um pacote óptico do anel, o *slot* já estará livre para a transmissão, podendo ser reusado imediatamente pelo nó.

No caso do *slot* não ser destinado ao nó, ele deverá passar por este sem que ocorra

remoção ou adição, desta forma, o comutador deve realizar a operação de passagem, abrindo as chaves 1 e 2 e fechando a chave 3. A capacidade de realizar as operações de remoção e passagem simultaneamente dá à rede a funcionalidade de *multicast* no domínio óptico.

O bloco de chaveamento óptico deve conter um comutador para cada canal de dados suportado no nó. Também deve existir um par de demultiplexador e multiplexador óptico para cada anel de fibra da rede caso existam mais de dois comprimentos de onda em uso.

Capítulo 3

CONCEPÇÃO DO PROTÓTIPO

Neste capítulo será descrito como o protótipo foi estruturado, ou seja, com a arquitetura da rede RCPO e do nó de acesso previamente definidas, o segundo passo foi idealizar a estrutura do protótipo analisando onde e como cada bloco funcionará e será interligado. Este capítulo está dividido em dois temas: *hardware* e lógica de processamento. A seção *hardware* mostra a divisão modular que foi adotada para contemplar todos os blocos e unidades definidas no capítulo anterior, e na lógica de processamento teremos uma visão de como a inteligência do nó de acesso da RCPO foi estruturada. Também será descrita a tecnologia que foi adotada para efetuar todo o processamento do nó da rede e o chaveamento óptico.

3.1 Hardware

Podemos dividir este tema em dois mundos, o elétrico e o óptico, pois o nó da rede é responsável por receber/enviar os dados dos clientes no domínio óptico e processá-los no domínio elétrico e ambas as frentes apresentam desafios comuns.

3.1.1 Domínio elétrico

Do ponto de vista de processamento de dados, o nó da RCPO apresenta dois principais desafios: o processamento de cabeçalho e o transporte de dados dos clientes.

Como visto anteriormente, processar o cabeçalho significa analisar os seus campos a fim de verificar se existem *slots* carregando informações para o nó, ou se existem *slots* vazios para a transmissão de seus próprios dados. Feita essa verificação, diversas consultas a tabelas e a atuação em outros blocos devem ocorrer de acordo com os procedimentos descritos no capítulo anterior.

Assim, para o processamento do cabeçalho, deve-se buscar uma tecnologia que permita a

implementação dos protocolos PAT+ e SAT+, além de ser capaz de executar as tarefas acima descritas com os tempos que precisamos.

A escolha natural para essa função, e amplamente utilizada em comutadores comerciais como *switches* e roteadores, são os microprocessadores. Estes permitem a fácil implementação de protocolos com linguagens de programação de nível médio como o ANSI-C. Dado nosso requisito de tempo, da ordem de μs , devemos escolher uma arquitetura de processador de 32 bits ou superior, com frequência de relógio da ordem de centenas de MHz.

Para o aumento da flexibilidade na programação, os microprocessadores podem ainda operar munidos de um sistema operacional. O uso de sistemas operacionais facilita a programação devido aos serviços que este provê, como: o interfaceamento entre os *softwares* de aplicação e o *hardware*, o escalonamento de processos, o controle de acesso à memória, entre outros. Em contrapartida, um sistema operacional introduz características não desejáveis, como o aumento da latência na execução de tarefas e a perda da determinação do tempo de tratamento de eventos. Um sistema operacional em tempo real (*Real Time Operating System* - RTOS) [18], visa a diminuição desses efeitos, sendo comumente utilizado em aplicações similares.

Para a execução da sincronização dos cabeçalhos, uma diferente abordagem deve ser realizada. O desafio neste caso está em garantir o mesmo tempo de retenção do cabeçalho no nó de acesso, para que assim a rede opere corretamente. Entretanto, o tempo de processamento do cabeçalho pode sofrer grandes variações, dependendo de variáveis como o número de *slots* destinados ao nó e o número de transmissões solicitadas pela unidade de transporte dos dados. Em ambos os casos, quando o cabeçalho chega ao nó, o Controlador de cabeçalho deve iniciar um temporizador antes de encaminhá-lo ao Processador, que, independentemente da variação, deverá devolver o cabeçalho já processado num tempo menor que o especificado. Ao recebê-lo de volta, o Controlador manterá o cabeçalho no nó até que o tempo especificado se esgote, reinserindo-o então na rede.

Para garantir que esta contagem seja fixa, é necessário à utilização de um temporizador em conjunto com uma lógica que execute a comunicação com o Processador de

cabeçalho, a recepção, e a re-inserção do cabeçalho na rede. Dado o requisito de tempo determinístico, o Controlador de cabeçalho deve ser implementado em *hardware*, evitando assim o uso de *software* para essa tarefa.

Para tal, podem ser utilizados componentes de lógica programável ou circuitos integrados para aplicação específica (*Application Specific Integrated Circuits* - ASIC).

Como ainda não existem ASICs para este fim, esta tecnologia não foi usada. O emprego de componentes discretos, como contadores e portas lógicas, é de baixo custo e de fácil utilização. Porém uma vez utilizado impossibilita a alteração do projeto de forma simples. A prototipagem com componentes de lógica programável (*Field Programmable Gate Array* - FPGA) [19] apresenta-se como solução intermediária. Com boa flexibilidade de reprogramação, capacidade em termos de quantidade de elementos lógicos e velocidade de operação, o desenvolvimento com tais componentes torna-se ainda mais atrativo devido à possibilidade do uso de linguagens de descrição de hardware (*VHSIC Hardware Description Language* - VHDL) e o Verilog [20], que facilitam o desenvolvimento de elementos mais complexos e diminuem o tempo de projeto.

Para o transporte dos dados dos clientes, diversos pontos devem ser analisados para a escolha dos componentes capazes de atender os requisitos. Um dos primeiros aspectos é o interfaceamento entre os clientes e o nó. Como visto, ele ocorre através de interfaces Ethernet, que é uma tecnologia muito difundida e de fácil acesso. Assim, para a realização deste interfaceamento diversos componentes integrados estão disponíveis no mercado fornecendo conectividade nas taxas de 10/100 Mbit/s, e 1/10 Gbit/s. Em geral são utilizados dois circuitos integrados que realizam a adaptação elétrica (PHY, da palavra *Physical*), e o controle de acesso ao meio.

Uma vez recebidos os quadros dos clientes, é necessário realizar a classificação, a montagem da fila e a transmissão dos pacotes ópticos. Estas operações apresentam o maior desafio do ponto de vista de implementação eletrônica no nó da RCPO, pois, além de serem tarefas complexas, necessitam de performance. Para tal, algumas tecnologias se destacam como possíveis soluções.

Os processadores de rede (*Network Processors* - NP) [21], por exemplo, são

componentes integrados desenvolvidos especialmente para o processamento de pacotes. Em sua maioria são compostos por um microprocessador que opera em conjunto com uma série de processadores específicos para o tratamento de pacotes. Coordenados por *softwares* que interagem no microprocessador, os processadores de pacotes executam os procedimentos desejados.

Porém, apesar de serem flexíveis, os NPs apresentam limitações e podem não ser apropriados caso seja necessária a realização de operações muito específicas, como por exemplo, a adição de campos especiais no quadro Ethernet.

Outro componente é a matriz de comutação (*Switch On a Chip* - SOC) [22], muito usada em equipamentos comutadores padrão. Os SOC's possuem altíssima performance e capacidade de comutação da ordem de dezenas de Gbit/s, porém apresentam baixa flexibilidade, não permitindo, por exemplo, a alteração do processo de encaminhamento de pacotes. Para que a utilização do SOC seja possível na RCPO, é necessária a adição de componentes que implementem funcionalidades específicas de uma rede de pacotes ópticos, como a agregação de pacotes numa fila de transmissão.

Uma terceira possibilidade é a utilização de componentes de lógica programável também para essa função. Tais componentes apresentam performance e capacidade compatíveis com as necessidades e dão total flexibilidade para a implementação de qualquer funcionalidade desejada.

A escolha entre NP, SOC, ou lógica programável, deve levar em conta principalmente as questões de simplicidade de implementação, funcionalidades desejadas, escalabilidade e performance, sendo necessário priorizar um ou dois desses pontos em relação aos demais, visto que nenhuma tecnologia possui todas essas características.

Por fim, o último desafio no que se diz respeito ao transporte de dados é a transmissão do pacote óptico no anel. Dada a característica de rede de chaveamento óptica, um transmissor apenas se conecta a um receptor quando efetivamente existirem dados para serem transmitidos. Isso impede que a sincronização entre transmissor e receptor seja permanente, assim como ocorre com a interface Ethernet. Nesta ocorre constantemente a troca de quadros especiais, chamados de quadros de *idle* (ociosos), que mantém a sincronização entre as interfaces, mesmo quando não existem dados para serem

transmitidos.

Em redes de chaveamento de pacotes ópticos, a sincronização ocorre por demanda, imediatamente antes da recepção dos dados dos usuários. Como o *slot* possui duração de dezenas de μs , a sincronização deve ocorrer em bem menos tempo, da ordem de poucos μs , caso contrário o *overhead* (cabeçalho) do pacote óptico será grande.

Com isto, a interface de transmissão no anel é implementada de forma proprietária, pois não há interface conhecida com tais características.

Esta interface proprietária deve ser composta por um componente serializador e deserializador (*Serializer/Deserializer* - SerDes) com rápida recuperação de relógio. Existem componentes integrados SerDes disponíveis que atendem os requisitos acima mencionados, com taxa de transmissão de até 10Gbit/s, assim a implementação da interface de transmissão torna-se viável.

3.1.2 Domínio óptico

Ao contrário do processamento eletrônico, que abrange diversas atividades, apenas uma tarefa deve ser realizada no domínio óptico, o chaveamento dos pacotes ópticos. Existem vários tipos de chaves ópticas para este fim e a escolha entre elas está relacionada ao seu tempo de chaveamento. A Tabela 3.1 [23] mostra os tempos de chaveamento requeridos para as principais necessidades de um sistema óptico.

Aplicação	Tempo de chaveamento requerido
Aprovisionamento	1 – 10 ms
Proteção	1 – 10 μs
Chaveamento de pacotes ópticos	1 ns
Modulação externa	10 ps

Tabela 3.1 - Tempos de chaveamento

Além do tempo de chaveamento, outras características importantes de uma chave óptica são:

- Razão de extinção: é a razão entre a potência de saída com a chave ligada e a potência de saída quando desligada. Essa razão deve ser a maior possível.
- Perda de potência: é a fração de potência de um sinal perdida pela utilização da chave, e deve ser a menor possível.

- *Crosstalk*: chaves, como qualquer outro componente, não são ideais. O *crosstalk* é a razão entre a potência de um sinal de entrada, que a priori não deveria aparecer numa saída, com a potência do sinal de entrada que de fato está associado àquela saída.

Tipo	Perda (dB)	<i>Crosstalk</i> (dB)	Tempo de Chaveamento
Mecânica/Termo-óptica	3	55	10 ms
Sílica	10	15	2 ms
Eletro-óptica	10	30	2 ms
SOA	0	40	1 ns
LiNbO₃	8	35	10 ps

Tabela 3.2 - Características de chaves ópticas

Na Tabela 3.2 [23], nota-se que dois tipos de chaves ópticas atendem o requisito de tempo da RCPO: as de Niobato de Lítio (LiNbO₃) e as baseadas em amplificador óptico a semicondutor (*Semiconductor Optical Amplifier* - SOA) [25].

As chaves de Niobato de Lítio em geral são usadas como moduladores externos. Apesar de serem as chaves ópticas mais rápidas, com atuação na ordem de ps, não possuem amplificação do sinal óptico como as chaves SOAs e possuem perda de potência.

Assim, o SOA, operando em modo de chaveamento, é a opção mais razoável para o desenvolvimento do comutador óptico descrito na seção 2.3.2.

3.1.3 Tecnologias adotadas

Uma vez analisadas as principais tecnologias candidatas, são apresentados agora os componentes escolhidos para a composição do nó da rede.

Dada a grande flexibilidade dos componentes de lógica programável, a escolha para a implementação do bloco eletrônico da unidade de transporte de dados foi um componente FPGA de alta densidade, modelo Stratix, do fabricante Altera. Esse componente possui internamente, além dos elementos de lógica programável, blocos de memória de acesso aleatório (*Random-Access Memory* - RAM), multiplicadores e recuperadores de relógio (*Phase Locked Loop* - PLL) que são necessários nos sub-blocos descritos no capítulo anterior. Esse FPGA também foi utilizado para a implementação da unidade de controle. Para tal foi adotado um processador de 32 bits do tipo *softcore*. Um

softcore é um componente eletrônico digital, geralmente desenvolvido em linguagem de descrição de hardware, que pode ser utilizado em diversas aplicações, sem que a pessoa que o está utilizando necessite conhecer as complexidades internas do componente. Entretanto, um *softcore* possui diversas flexibilidades em comparação aos componentes integrados, como a possibilidade de ser removido ou alterado mesmo depois da finalização do projeto de *hardware*, dada às próprias características de reconfiguração dos componentes de lógica programável.

Existem inúmeros *softcores* disponíveis hoje no mercado, realizando as mais diversificadas funções. Os *softcores* em geral são fornecidos gratuitamente pelos fabricantes de componentes de lógica programável, ou comercializados por empresas especializadas neste tipo de desenvolvimento. O *softcore* de processador utilizado NIOS-II [24] é fornecido pelo fabricante Altera. Possui uma arquitetura de 32 bits e pode atingir performance superior a 100 milhões de instruções por segundo. Para o desenvolvimento do *software* executado neste processador é utilizada a linguagem C, usando a ferramenta de desenvolvimento e compilador fornecidos pelo fabricante.

Para a interconexão dos clientes Ethernet foi escolhido o componente integrado IXF1104 do fabricante Cortina Systems [26]. Este componente possui 4 MACs, cada um com capacidade de 1 Gbit/s. A Figura 3.1 ilustra a interconexão do componente, ligado ao FPGA pelo barramento SPI-3, há um microcontrolador de 8 bits para a configuração dos registradores no processo de inicialização e monitoramento, e os componentes PHY. Os PHYs são necessários para prover conectividade Ethernet nas taxas 10/100/1000 bit/s nos modos *half* e *full duplex* em interfaces elétricas. No caso dos clientes se conectarem por interfaces ópticas, os componentes PHY não são necessários, sendo o IXF1104 ligado diretamente a *transceivers* do padrão *Small Form Pluggable* (SFP) [27], que converterão o sinal elétrico em óptico e vice-versa.

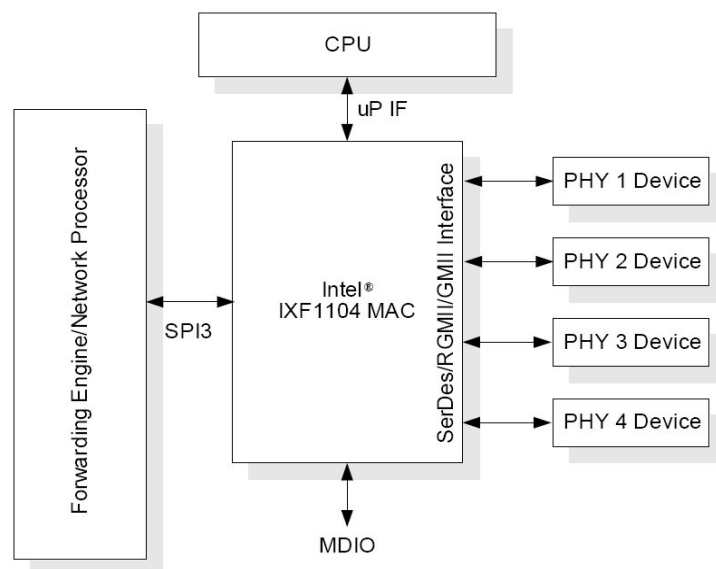


Figura 3.1 - IXP1104 e interconexões [26]

Para a transmissão e recepção dos pacotes ópticos e dos cabeçalhos adotamos o componente integrado TLK2201A da Texas Instruments [28]. Este componente SerDes provê uma rápida sincronização, necessária para esse tipo de comunicação, realizando a recuperação de relógio com no máximo 1024 bits, levando aproximadamente 1 μ s para sincronizar quando operando com taxa de 1 Gbit/s. Além dessa característica, a taxa de transmissão é de Gbit/s, de forma a manter a compatibilidade com o tráfego gerado pelos clientes. Esse componente também é conectado a um módulo do tipo SFP, a fim de realizar a conversão do sinal óptico em elétrico e vice-versa.

A Figura 3.2 mostra uma aplicação típica do componente escolhido. Note que o componente possui uma interface serial conectada aos módulos ópticos e uma paralela de 10 bits ligada ao componente que irá processar os dados recebidos e, por outro lado, enviá-los para a interface serial. Há também uma interface JTAG utilizada para testes. A alimentação deste componente é bastante simples, ao contrário do controlador MAC como veremos no próximo capítulo.

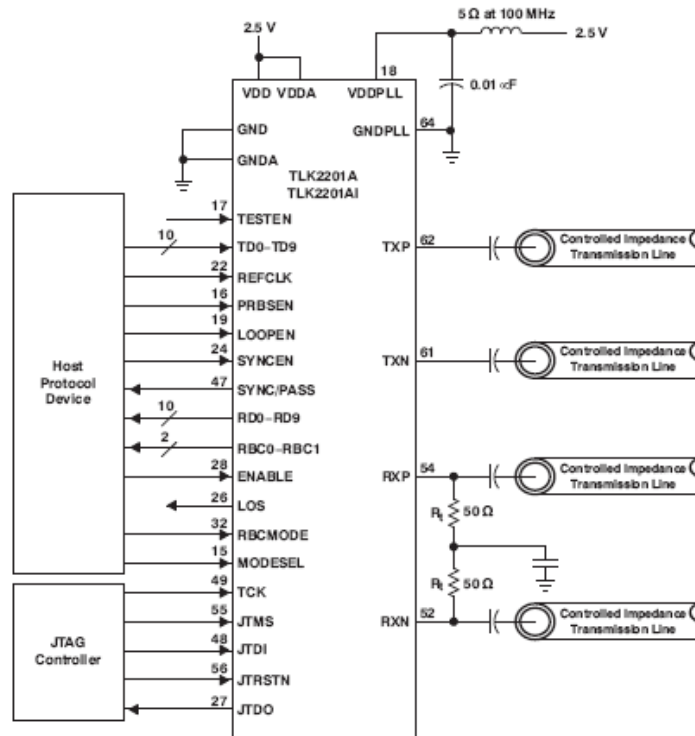


Figura 3.2 - SerDes da Texas Instruments [28]

O bloco de chaveamento óptico, descrito na seção 2.3.2, é implementado utilizando-se SOAs como chaves ópticas. O SOA possui chaveamento na ordem de ns, é um amplificador de sinal óptico que pode compensar as perdas causadas pelos demais componentes do nó de acesso como: *delays* de fibra óptica, *splitters* (divisores), *muxs* (multiplexadores), etc. A Figura 3.3 mostra o encapsulamento de um SOA. Apesar de existirem SOAs encapsulados disponíveis no mercado, os utilizados na RCPO foram encapsulados no próprio CPqD [29].

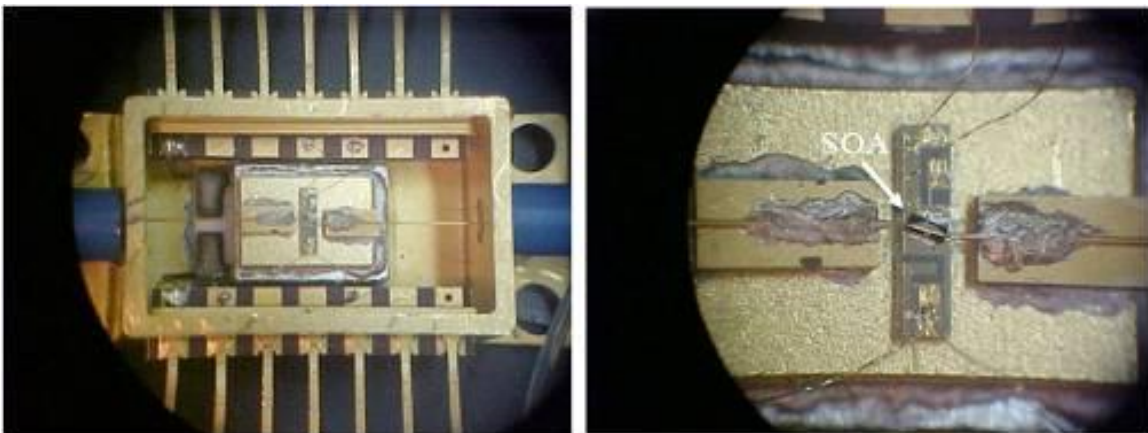


Figura 3.3 - SOA encapsulado no CPqD

3.2 Modularidade

Para facilitar o processo de depuração do protótipo, decidimos fazê-lo de forma modular, assim cada módulo poderia ser desenvolvido separadamente e também com a vantagem de ser testado individualmente, tornando a integração mais simples e rápida.

A divisão foi motivada pela tecnologia utilizada em cada módulo. Deste modo, todo o processamento dos dados que foi desenvolvido num componente de lógica programável ficou no Módulo de Processamento de Dados, que deveria conter um FPGA, memória para a criação e armazenamento das filas de pacotes e interfaces para interconexão com os demais módulos.

Seguindo este caminho, o próximo módulo foi chamado de Módulo de Acesso aos Clientes, onde estaria contido o IXF1104, responsável por fazer a interconexão com os clientes através de 04 interfaces ópticas Gigabit Ethernet (GbE) e com o Módulo de Processamento de Dados, através de uma interface SPI-3. Além do IXF e dos módulos ópticos, este contém um microcontrolador para configuração e gerenciamento do controlador MAC.

O terceiro módulo é o de Acesso ao Anel Óptico, responsável por transmitir e receber os pacotes no anel óptico através de 4 interfaces ópticas operando em Gbit/s com protocolo proprietário e com o Módulo de Processamento de Dados através de uma interface paralela. Nele está contido um SerDes por interface. Este circuito integrado requer simples configuração, não sendo necessário o uso de microcontrolador.

O quarto e último módulo é o de Chaveamento Óptico, que contém um SOA cada. Cada nó utiliza três módulos iguais a este com a função de abrir ou não o caminho para os pacotes de dados e amplificar o sinal se for o caso.

A Figura 3.4 mostra a interligação de cada módulo e seus principais componentes.

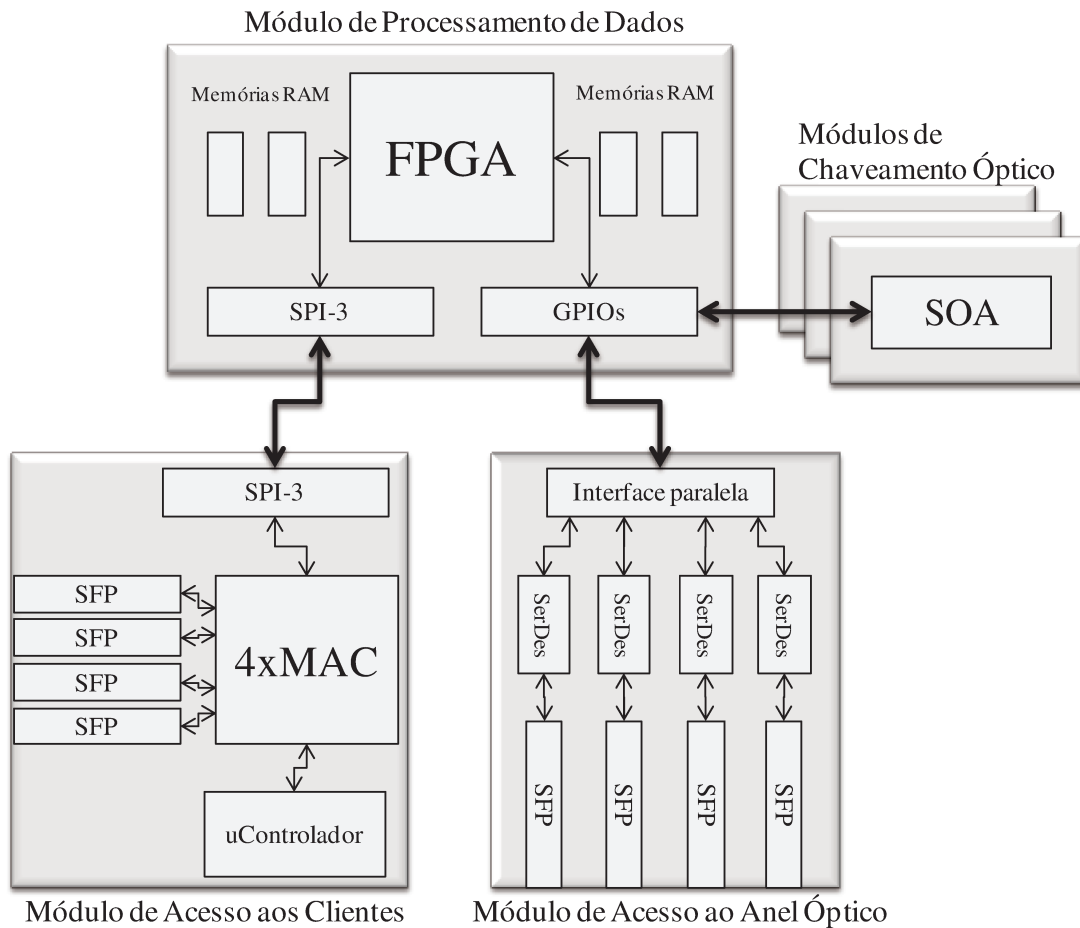


Figura 3.4 - Interligação dos módulos do nó de acesso da rede

Com a definição dos módulos e seus principais componentes mostrado na figura acima, o protótipo foi desenvolvido.

3.3 Lógica de processamento

Com os módulos que constituem o nó da rede e seus respectivos componentes definidos, o próximo passo foi elaborar os blocos de processamento que compõem a unidade de controle e transporte de dados. Estes blocos foram implementados no FPGA do Módulo de Processamento de Dados em linguagem VHDL. A Figura 3.5 ilustra os blocos e também suas conexões interna e externamente do FPGA.

Todas as funcionalidades especificadas em relação à inteligência do nó da RCPO no capítulo anterior são executadas, e foram divididas em vários blocos para facilitar o desenvolvimento, os testes e a integração.

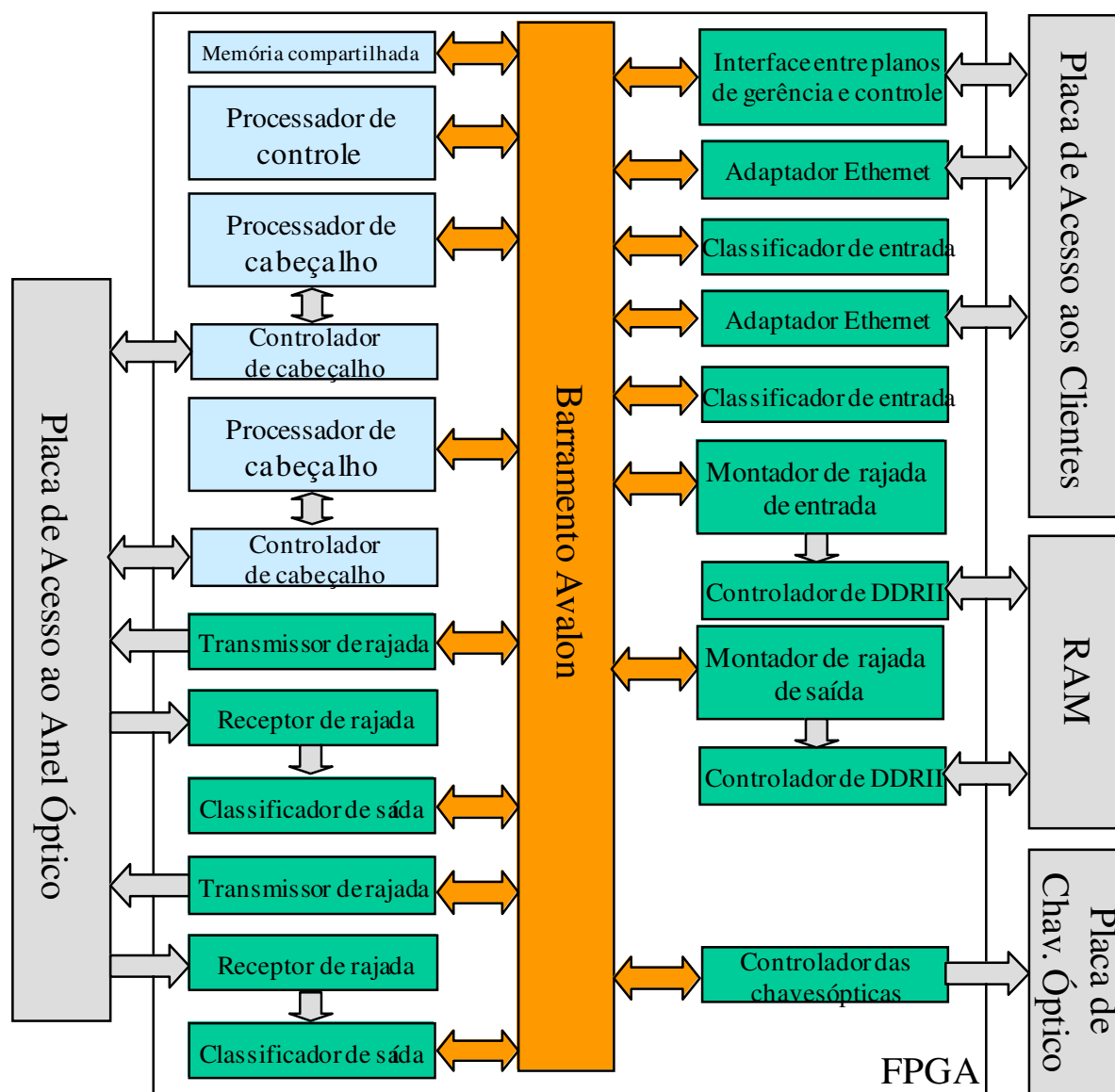


Figura 3.5 - Diagrama dos blocos de processamento

As seções seguintes descrevem sucintamente a função de cada bloco ilustrado na Figura 3.5.

3.3.1 Unidade de controle e barramento de comunicação

Os blocos Memória Compartilhada, Processadores e Controladores de Cabeçalho na Figura 3.5 compõem a unidade de controle do nó da rede, e seguem o comportamento descrito na seção 2.3.1.

A unidade é composta por três processadores. O primeiro processador, chamado de processador de controle, realiza as funções de configuração inicial, preenchimento de

tabelas de endereços, e comunicação com o plano de gerência.

As configurações iniciais dizem respeito aos parâmetros da rede, como o tempo de rotação do anel, tamanho do *slot*, tempo de processamento de cabeçalho. Todos esses parâmetros são configurados de forma estática na rede, e devem ser fornecidos pelo usuário diretamente na programação desse processador, ou via plano de gerência. O mesmo acontece com as tabelas de endereçamento.

Os Processadores de cabeçalhos são dedicados para cada anel da rede. Assim, existem dois Processadores de cabeçalho no protótipo que em conjunto com os Controladores de cabeçalho, processam os cabeçalhos dos dois anéis de acordo com os critérios definidos na seção 2.2.

O barramento de comunicação apresentado, chamado de barramento *Avalon*, é uma solução de barramento de dados fornecida pelo fabricante Altera e igualmente ao processador NIOS-II, é uma solução do tipo *softcore*, que permite a criação de um barramento de multi-conexões, onde diversos blocos podem trocar dados simultaneamente sem condições de concorrência.

Quando dois ou mais blocos tentam acessar o mesmo recurso, por exemplo, dois classificadores de entrada transmitindo dados para o montador de rajada, a condição de concorrência é inevitável. Para tal, o barramento *Avalon* possui árbitros que impedem que os acessos colidam.

Desta forma, o *Avalon* é uma solução eficaz, que simplifica a implementação da comunicação das unidades de controle e transporte de dados.

3.3.2 Unidade de transporte de dados

Os demais blocos menos o barramento *Avalon* dentro do FPGA da Figura 3.5 compõem a unidade de transporte de dados do nó e foram elaborados para executarem as funcionalidades citadas na seção 2.3.2. Por motivos de simplificação da integração do protótipo, algumas das funcionalidades descritas, como, por exemplo, o Controlador de tabelas de endereços, podem não ser implementadas na primeira versão.

Como visto no item anterior, essa função é exercida pelos próprios classificadores, sendo suas tabelas de endereçamento preenchidas estaticamente pelo Processador de controle.

Ainda com relação aos classificadores e simplificação, a única forma de classificação

necessária para uma primeira versão do protótipo é a por endereços MAC. Outras formas de classificação como vlans (caminhos virtuais) e bits de prioridade podem ser dispensadas. Como consequência, o montador de rajada não suportará a separação do tráfego em sub-filas, não implementando a diferenciação de qualidade de serviço, tanto para a classe de serviço de melhor esforço, quanto para a classe de serviço reservada.

Para a agregação do tráfego em rajadas, as filas controladas pelo montador são armazenadas no banco de memória RAM DDR-II disponível no módulo. Para a realização do acesso a este banco de memória, um controlador de RAM DDR-II é utilizado.

Os blocos: adaptador Ethernet, transmissor e receptor de rajadas e comunicação entre planos de controle e gerência realizam as funcionalidades descritas na seção 2.3.2.

Por fim, o controlador das chaves ópticas executa os comandos enviados pela unidade de controle nas chaves ópticas de acordo com a operação desejada.

Capítulo 4

DESENVOLVIMENTO DO PROTÓTIPO

O desenvolvimento do protótipo inclui desde a escolha dos componentes de cada módulo descrito no capítulo anterior, aqui também chamados de placas, até sua fabricação e montagem.

Com exceção da Placa Processadora, as demais foram totalmente projetadas, desenvolvidas e montadas nos laboratórios do CPqD, com exceção da fabricação propriamente dita da placa de circuito impresso (PCI) e montagem de alguns componentes com encapsulamentos que exigiram equipamentos específicos e não eram disponíveis.

Neste capítulo serão descritas as técnicas utilizadas para o desenvolvimento das placas de circuito impresso, os pontos críticos de cada projeto e o estudo estrutural realizado.

O desenvolvimento das atividades de projeto do nó de acesso da RCPO foi iniciado a partir da análise dos requisitos de funcionalidades que foram previstos no projeto e quais deles poderiam, efetivamente, ser implementados num protótipo de laboratório.

Diversas reuniões técnicas foram realizadas no CPqD para avaliação de toda a tecnologia envolvida no projeto até que, finalmente, foram adotadas soluções técnicas que caracterizam o estado da arte em eletrônica e optoeletrônica, mas que viabilize assim a evolução do protótipo para um equipamento comercial, atendendo assim as restrições técnico-econômicas inerentes.

4.1 Placa Processadora

No capítulo anterior descrevemos esta placa como um Módulo de Processamento de Dados que deveria prover um componente de lógica programável (FPGA), memória suficiente para a função de armazenamento de pacotes e uma gama de interfaces para comunicação com as placas filhas. Assim sendo, antes de desenvolver e produzi-la, foi

feita uma pesquisa no mercado, como ocorreu com as demais placas, para verificar se já existia algum produto comercial que atendesse os requisitos necessários. Como resultado nós encontramos uma placa que possuía os requisitos mínimos, já que o desenvolvimento deste módulo iria consumir mais recursos humanos e somar variáveis que poderiam prejudicar o andamento do projeto caso algum erro fosse cometido no seu desenvolvimento, além de que a grande capacitação deste módulo está na lógica de processamento desenvolvida e não no desenvolvimento da placa propriamente dita.

A placa disponível comercialmente dispunha dos seguintes itens:

- 01 FPGA Altera Stratix com alta densidade de elementos lógicos (30.000), quantidade suficiente para suprir a necessidade estimada [19];
- 128 MB de memória RAM para armazenar os pacotes de dados;
- 140 Pinos de propósito geral (*General Purpose I/Os* - GPIOs) para serem utilizados na comunicação com as demais placas;
- 01 interface RS232C para acesso ao processador NIOS-II.

Como os itens acima atendiam os requisitos necessários, ela se tornou uma forte candidata a ser adquirida.

A Figura 4.1 mostra a Placa Processadora e seus principais componentes.

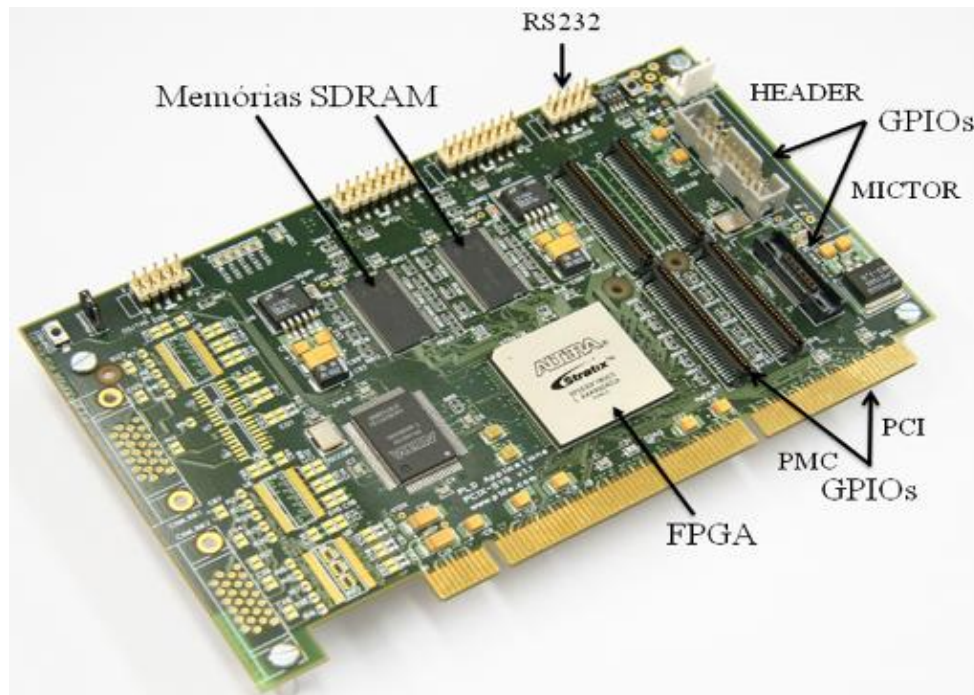


Figura 4.1 - Placa Processadora adquirida da PLD Applications [30]

Podemos observar na figura que a maioria dos 140 pinos de I/Os estão disponíveis em duas diferentes interfaces, a *Peripheral Component Interconnect* (PCI) e a *PCI Mezzanine Card* (PMC). Esta última combina o padrão elétrico da interface PCI com a mecânica de outro padrão chamado *Common Mezzanine Card*.

Antes da aquisição deste módulo, foi realizada uma consulta com o fornecedor para verificar se era possível utilizar os pinos das interfaces PCI e PMC genericamente, ou seja, se não eram dedicados. O fornecedor confirmou que não eram dedicados e poderiam ser utilizados como GPIOs.

Também foi realizado um estudo dimensional da placa para saber se seria possível a adaptação mecânica das outras placas que seriam desenvolvidas, dando o início a uma idéia de como seria o aspecto físico do protótipo quando pronto.

Com todas as questões respondidas e aprovadas, foi confirmado o uso desta placa para o projeto e, como resultado, o desenvolvimento deste módulo se ateve no entendimento de seu funcionamento e no desenvolvimento da lógica programável que foi especificada no capítulo anterior. Com isso, as outras placas desenvolvidas seguiram os padrões elétricos e mecânicos desta como referência.

A Figura 4.2 mostra um diagrama de blocos da Placa Processadora com seus dispositivos e interfaces.

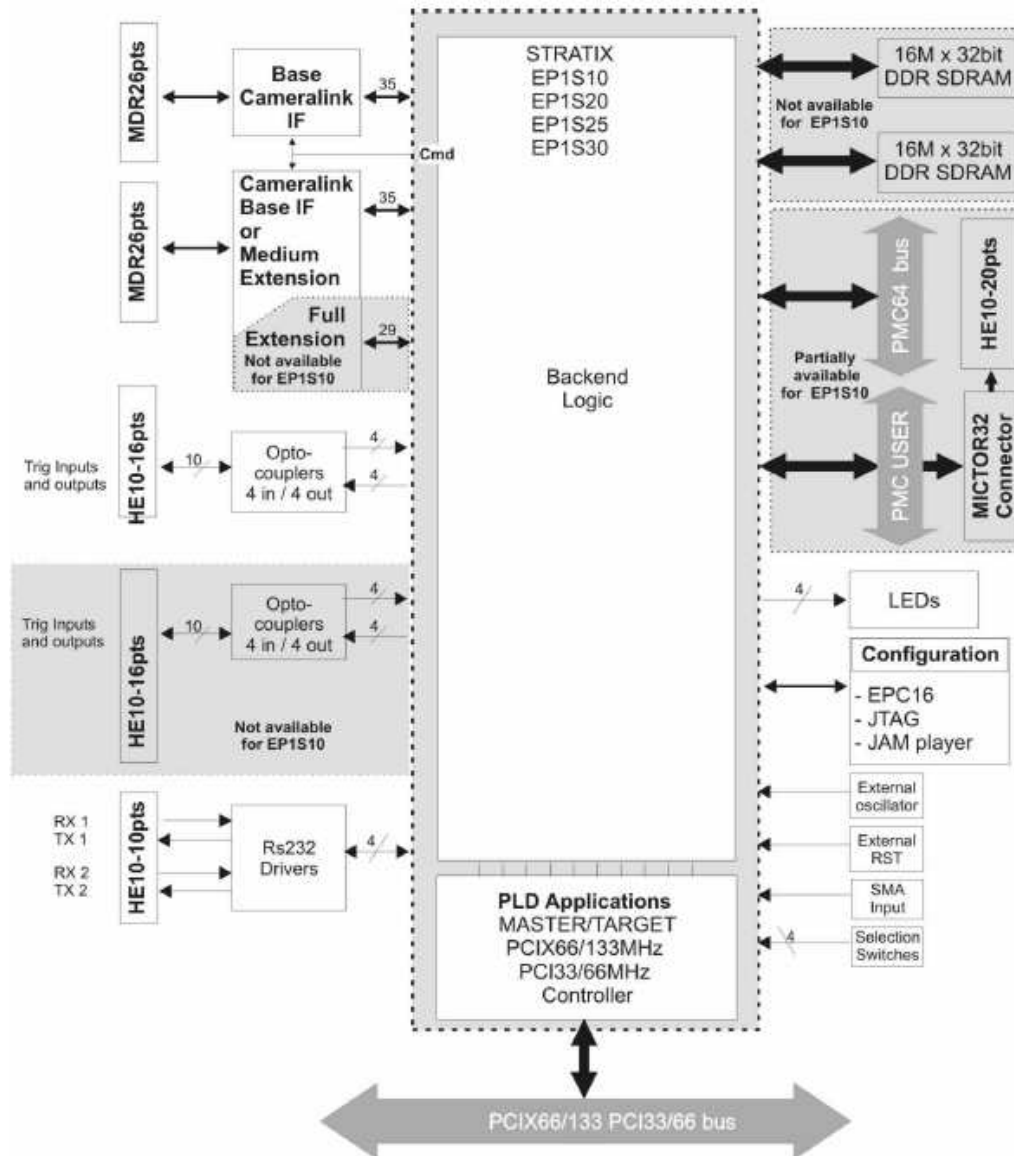


Figura 4.2 - Diagrama de blocos da Placa Processadora [30]

Como a placa adquirida possui o FPGA modelo EP1S30 (30.000 elementos), todos os dispositivos ilustrados estão disponíveis, menos o circuito da interface Cameralink, que não foi requisitado afim de diminuição de custos, pois não seria utilizado no projeto.

No diagrama de blocos, podemos observar que existe uma minoria de GPIOs interligados em paralelo com o conector PMC. Dois tipos de conectores são utilizados, um de padrão MICTOR (utilizado para prover acesso a analisadores lógicos e permitir medição de sinais de até 250 MHz) e o outro modelo utilizado é o HEADER (para uso geral). Os quais foram bem vindos, principalmente o HEADER para interconexão do

Módulo de Chaveamento Óptico.

Os restantes dos dispositivos mostrados no diagrama de blocos da Placa Processadora são para auxílio e controle de componentes, como diodos emissores de luz (*Light Emission Diode* - LED), botões de *reset*, osciladores, etc.

Uma vantagem desta placa é que apenas um valor de tensão de entrada (5 Vdc) é suficiente para o seu funcionamento, já que ela possui reguladores lineares e conversores DC-DC para alimentar todos os circuitos. A placa também possui um oscilador na frequência de 125 MHz compatível com os requisitos das demais placas e é disponibilizado nos conectores que iremos utilizar.

4.1.1 Estudo estrutural do protótipo do nó da rede

Uma vez definida, foi possível estudar como seria a conexão mecânica da placa com os outros módulos. Como ponto de partida, tinha-se como base as interfaces que seriam utilizadas nas outras placas, pois neste momento já estavam definidas as arquiteturas que seriam utilizadas nos outros módulos.

A interface de comunicação de dados com a Placa Processadora da Placa de Acesso aos Clientes é que mais necessita de pinos para conexão, seguida pela Placa de Acesso ao Anel Óptico e depois a Placa de Chaveamento Óptico.

A interface PMC da Placa Processadora, que pode ser utilizada como GPIO, é a que possui maior quantidade de pinos e então foi escolhida para fazer a interconexão com a Placa de Acesso aos Clientes, já que a interface PCI não tinha pinos suficientes para isto. Já o número de pinos da interface PCI foi suficiente para a interconexão da Placa de Acesso ao Anel Óptico. Por último foi escolhido o local para a conexão da Placa de Chaveamento Óptico, que por necessitar de apenas um pino, com a função de liga-desliga, foi escolhido um pino do conector HEADER.

Após esta definição, o protótipo do nó da rede já possuía uma característica mecânica, que pode ser observada na Figura 4.3.

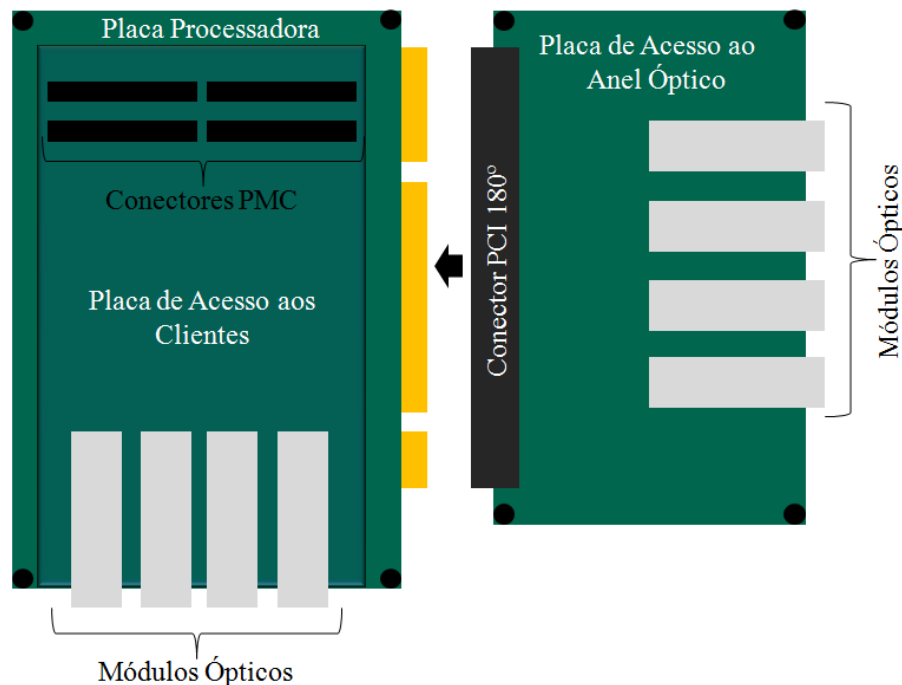


Figura 4.3 - Concepção modular do protótipo do nó da rede

A figura acima não mostra a Placa de Chaveamento Óptico, pois a conexão com a Placa Processadora é feita através de um cabo e não como as demais, que tem o papel de placas filhas (*daughter boards*) e são conectadas diretamente sem a necessidade de cabos. Essa característica da Placa de Chaveamento Óptico possibilita que ela seja instalada distante das demais.

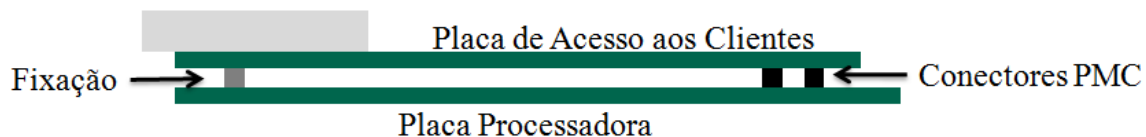


Figura 4.4 - Desenho da vista lateral das placas Processadora e de Acesso aos Clientes

Na Figura 4.4 pode-se observar o método de fixação das placas Processadora e de Acesso aos Clientes. A parte frontal da placa é fixada pelos conectores PMC e a traseira fixada por espaçadores e parafusos. O primeiro desafio do projeto da Placa de Acesso aos Clientes já foi visto nesta etapa, que é de garantir o encaixe mecânico entre os conectores e os parafusos na parte traseira com a Placa Processadora.

4.2 Placa de Acesso aos Clientes

A Placa de Acesso aos Clientes, chamada anteriormente de Módulo de Acesso aos Clientes, foi a primeira placa filha desenvolvida. Tratou-se de um projeto de alta complexidade e desafios concentrados no desenho do layout da PCI, que possui várias linhas de transmissão de até 1,25 Gbit/s que exigem impedância controlada e comprimento equalizado e um grande esforço no estudo de componentes com diversos tipos de interfaces e registros para configuração.

Como já descrito no capítulo anterior, essa placa disponibiliza quatro interfaces ópticas Gigabit Ethernet do padrão SFP para conexão de três clientes e mais um computador para o plano de controle e é conectada através de uma interface SPI-3 com a Placa de Processamento para envio e recebimento dos dados. O elemento principal dessa placa é um controlador de acesso ao meio (*Medium Access Controller* - MAC) que provê todo o controle de tráfego até a camada 2 do modelo OSI [17].

4.2.1 Projeto elétrico da Placa de Acesso aos Clientes

A Figura 4.5 mostra o diagrama de blocos da Placa de Acesso aos Clientes. Nela podemos ver as interfaces de dados (RS232, SPI-3, CPU e SFPs), de alimentação e de controle (LEDs e botões). Os principais componentes do sistema são o controlador MAC (IXF1104), e a CPU (microcontrolador de 8 bits da Atmel).

4.2.1.1 Descrição do funcionamento do controlador MAC

O MAC IXF1104 é o principal componente deste projeto. Ele é o responsável por fazer o tratamento dos quadros GbE recebidos e enviados através das interfaces com os clientes e com a Placa de Processamento. Esta placa atende os requisitos das camadas 1 (Física) e 2 (Enlace) do modelo OSI. As funções da CPU que controla o MAC são realizadas por um microcontrolador de 8 bits e o código para definir as rotinas de controle também foi desenvolvido pela equipe.

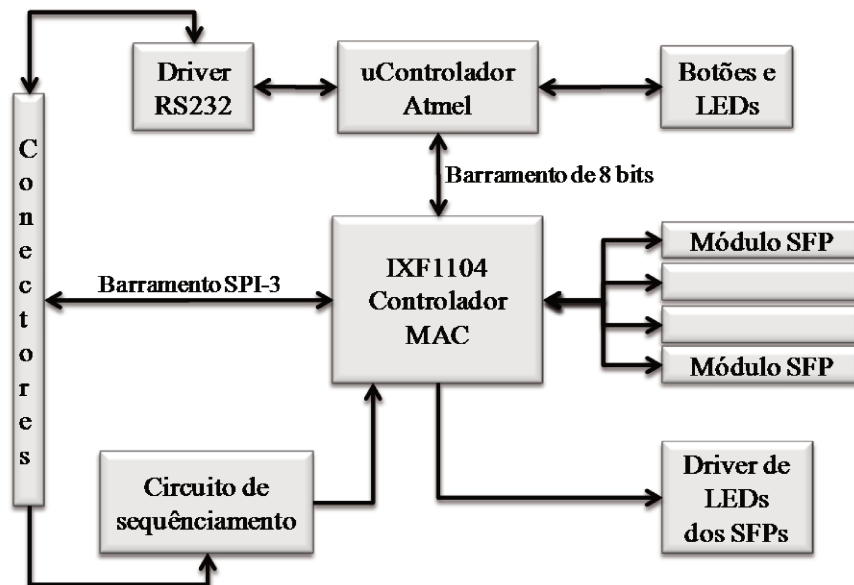
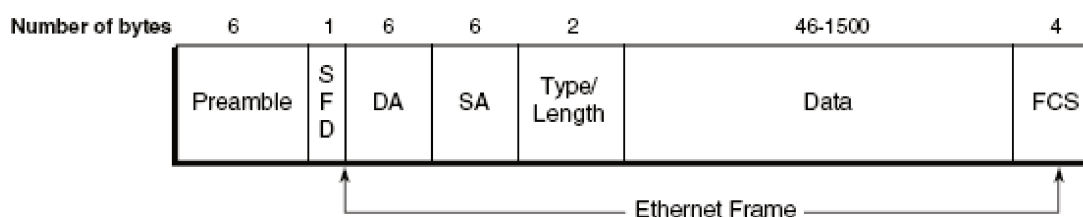


Figura 4.5 - Diagrama de blocos da Placa de Acesso aos Clientes

A Figura 4.6 e a Figura 4.7 irão ajudar na compreensão do papel do IXF1104.

Podemos observar na Figura 4.6 todos os campos do quadro do padrão Ethernet. O primeiro Byte que carrega informação está no campo DA (*Destination Address*) que contém 6 Bytes. É nele que está indicado o endereço de destino do quadro Ethernet, seguido pelo SA (*Source Address*), *Type/Length*, *Data* e por último o FCS (*Quadro Check Sequence*). Este último com 4 Bytes contém informação para que o dispositivo MAC possa verificar a integridade do quadro e tomar providências pré determinadas.



Note: 64 Byte Minimum / 1518 bytes Maximum

Figura 4.6 - Formato do quadro de dados do padrão Ethernet [26]

Os dois primeiros campos *Preamble* e SFD (*Start of Quadro Delimiter*), também chamados de “limpa-trilhos”, são utilizados para que o dispositivo MAC consiga detectar que está recebendo um quadro GbE. O primeiro campo possui uma seqüência com 6 Bytes que é a identificação de um quadro Ethernet e o segundo possui 1 Byte que

indica o início da informação.

Quando um quadro GbE é recebido pela interface do cliente, o IXF1104 verifica se não existem erros no quadro através da informação contida no campo FCS e retira-o juntamente com os dois primeiros campos (*Preamble* e SFD) e encaminha o restante do quadro para a Placa Processadora de acordo com as configurações pré-programadas. Estas configurações podem ser:

- enviar todos os quadros recebidos;
- sinalizar ou descartar quadros contendo erro;
- descartar quadros com um determinado endereço de destino;
- descartar *broadcast*;
- etc.

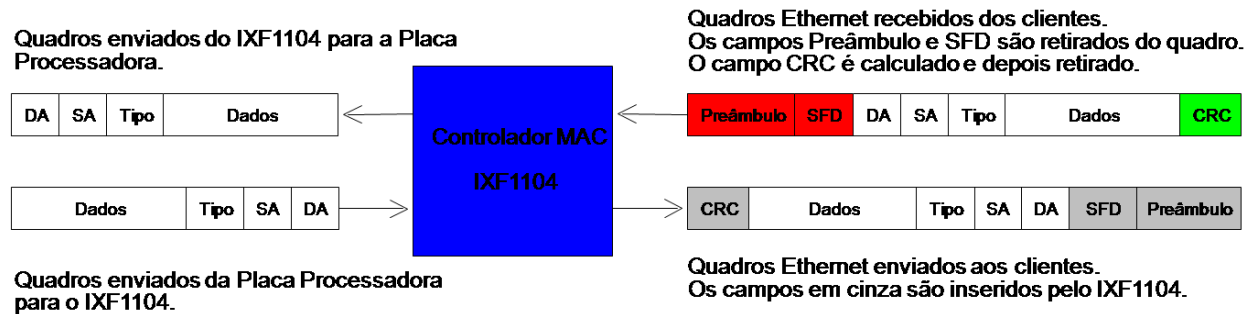


Figura 4.7 - Formato dos quadros enviados e recebidos pelo IXF1104

Quando a Placa Processadora necessita enviar dados para os clientes, ela não envia os quadros GbE contendo os campos *Preamble*, SFD e FCS. O controlador IXF1104 se encarrega da tarefa de inserir estes campos e encaminhar o quadro GbE completo para as interfaces clientes.

O controlador possui, para auxiliar nos testes da placa, duas configurações de *loopback*. São chamados de *local* e *external*. O Local Loopback transfere os dados que chegam à interface SPI-3 de volta, sem fazer qualquer tipo de alteração, permitindo o projetista verificar a integridade dos dados enviados para esta interface pelo processador de rede. O External Loopback transfere os dados que chegam nas quatro interfaces GMII de volta, também sem fazer alterações, permitindo avaliar se o meio físico entre os módulos ópticos da placa e o dispositivo do cliente está com algum problema. Estes *loopbacks* foram muito úteis para o teste da placa, pois encontramos problemas de mau-contato,

devido à oxidação, no conector utilizado para a interface SPI-3.

4.2.1.2 Interfaces

A Figura 4.8 ilustra a arquitetura interna do IXF1104 e as interfaces pertinentes.

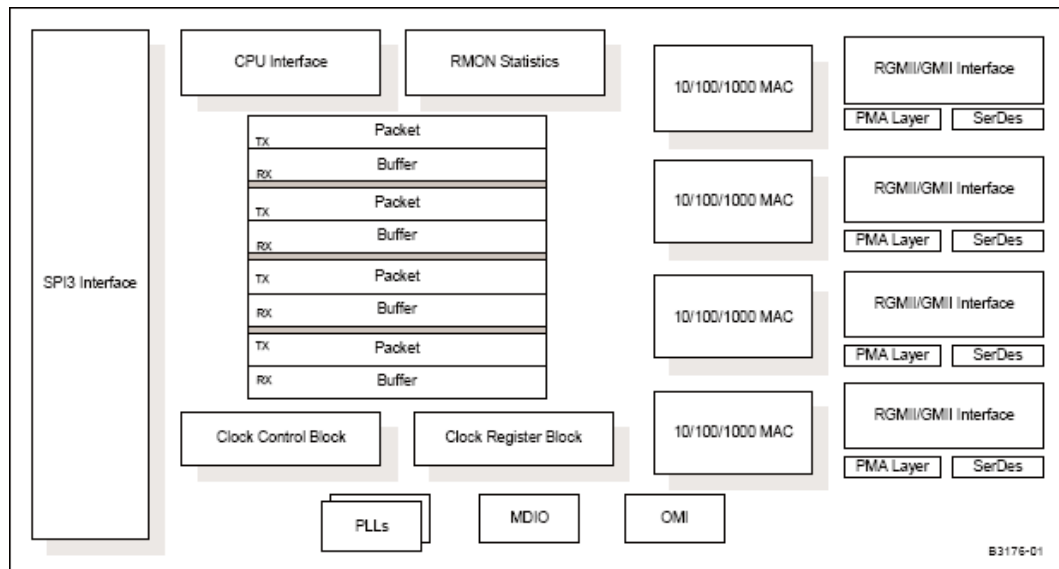


Figura 4.8 - Interfaces do controlador IXF1104 [26]

- **GMII**

Podemos observar que ele possui 04 MACs independentes de 10/100/1000 Mbit/s. O acesso a estes MACs, que são responsáveis por prover o controle da camada 2, pode ser feito através de uma interface de acesso ao meio (*Gigabit Media Independent Interface* - GMII) ou de um SerDes. A interface GMII é utilizada quando os dados trafegam em cabos de cobre e o SerDes quando são utilizados cabos de fibra óptica, que é o utilizado no protótipo. Para a interface GMII é necessário o uso de PHYs (*Physical Layer*) externos, que provêm acesso a camada 1 do modelo OSI. O SerDes possui internamente uma interface com suporte ao módulo óptico, e portanto, nenhum componente externo é necessário.

Com isso, pelo lado do cliente foram utilizados 04 módulos ópticos padrão SFP ligados diretamente aos SerDes internos do IXF1104, como ilustrado na Figura 4.9.

A figura extraída do esquema elétrico da placa indica apenas a conexão de um módulo SFP, pois os demais são idênticos.

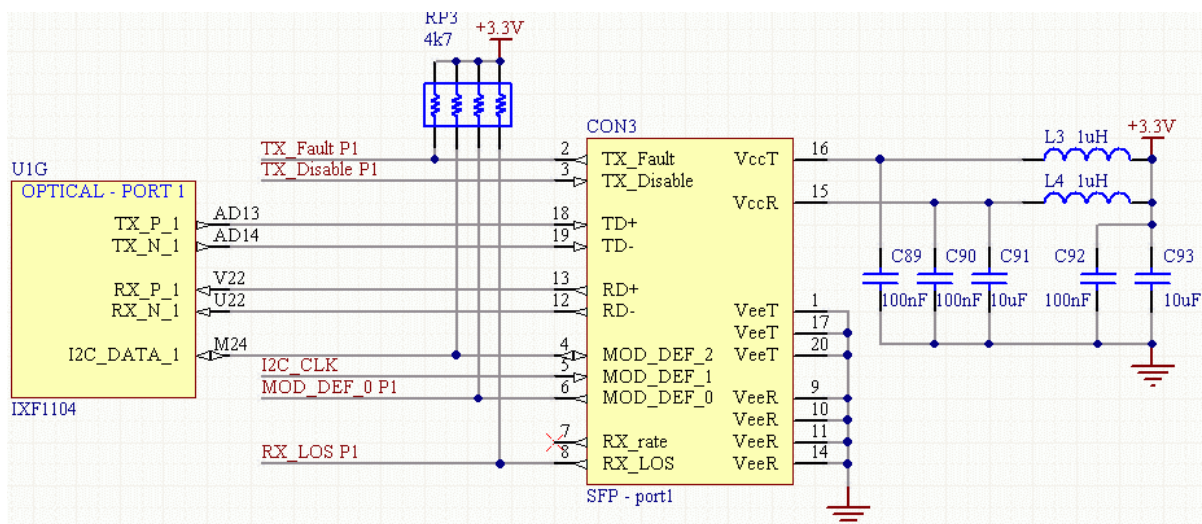


Figura 4.9 - Conexão entre o SerDes do IXF1104 e o módulo óptico

• SPI-3

Há também uma interface paralela específica para transporte de pacotes (*System Packet Interface level 3* - SPI-3) que permite a transferência dos dados entre o IXF1104 e a Placa Processadora. Esta interface é otimizada para aplicações onde existe transferência de dados entre MACs e processadores de rede, que no nosso caso é o FPGA. A SPI-3 permite transferência de dados de até 4,256 Gbit/s quando operando na frequência máxima de relógio que é de 133 MHz. Seguem os sinais utilizados pela interface:

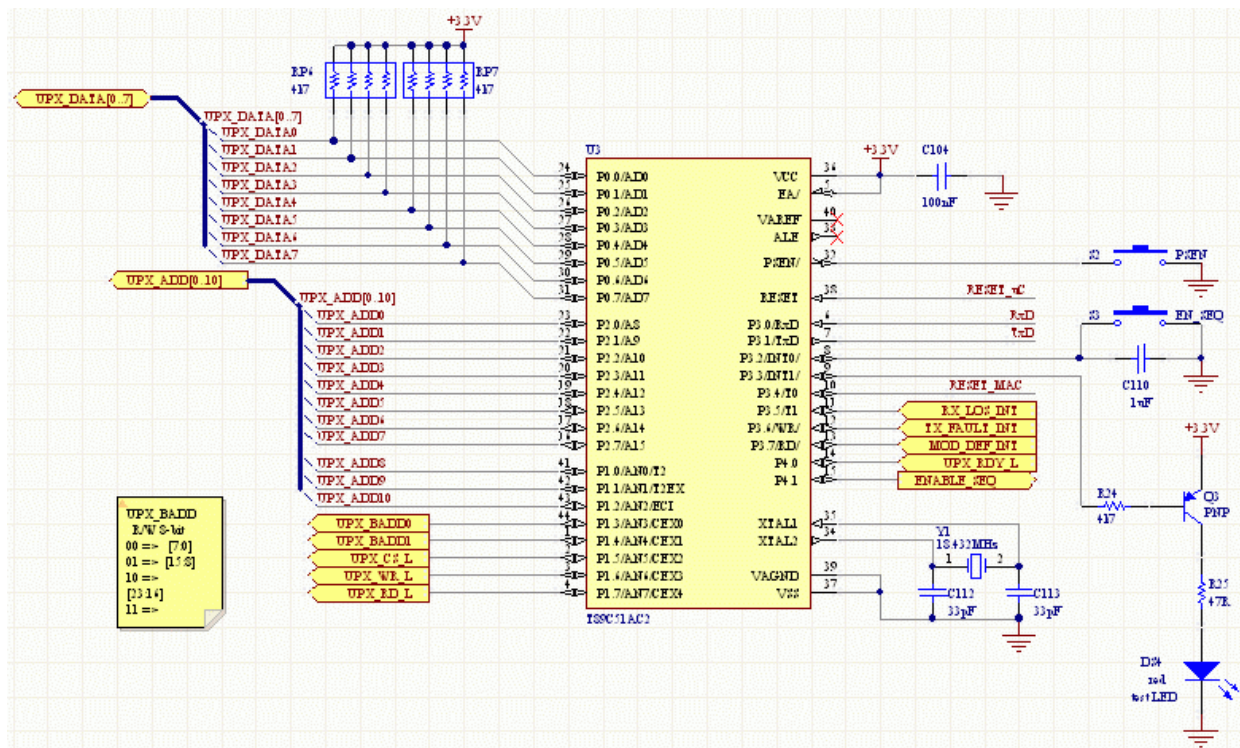
- 32 sinais de dados (LVTTTL);
- 08 sinais de controle (LVTTTL);
- 01 sinal de relógio (LVTTTL).

Como a Placa Processadora disponibiliza um relógio de 125 MHz no conector que escolhemos para utilizar com a SPI-3, a transferência máxima de dados se limitou a 4 Gbit/s, sendo esta taxa suficiente para transferir os dados das quatro interfaces GbE, mesmo a taxa de dados da interface GbE sendo de 1,25 Gbit/s. Isso é possível pois só a codificação 8b/10b dos pacotes já aumenta em 20% o valor e, ainda descontando os campos do quadro Ethernet que o IXF1104 retira para enviar à Placa Processadora, a taxa necessária para transferir somente a informação é menor que 1 Gbit/s.

• CPU

Para controle e gerenciamento é utilizada uma interface de central de processamento (*Central Processing Unit* - CPU), que através de escrita e leitura de registradores,

Esta interface opera com 32 bits de dados e 11 bits de endereço, mas para facilitar o desenvolvimento é possível utilizar microcontroladores de 8 bits para configuração do IXF. Para operações com 8 bits, 3 ciclos adicionais para escrita ou leitura são necessários. O IXF disponibiliza dois pinos (UPX_BADD0 e UPX_BADD1) para seleccionar qual Byte, dos 4 necessários para completar um endereço, se refere o ciclo atual. A Figura 4.10 mostra as ligações da interface CPU do IXF referente ao microcontrolador com arquitetura 8051. O microcontrolador utilizado foi um Atmel da série T59C51 [33].



Devido ao fato de utilizarmos um microcontrolador de 8 bits e a interface nativa do IXF ser de 32 bits, não foi possível utilizar o recurso de mapeamento de memória que facilitaria a comunicação entre os dispositivos. O método citado permite ao programador não se preocupar em como a interface funciona, e programar os ciclos de escrita e leitura manualmente. Portanto, cada ciclo foi estudado e programado manualmente para o microcontrolador atender as especificações de tempo do IXF no momento do acesso à

interface. Na Figura 4.11 observa-se o diagrama de tempo do ciclo de escrita da interface CPU do IXF1104. Com essa informação foi possível programar a rotina de escrita nos registradores. Note que cada pino/função da interface possui um determinado tempo a ser obedecido, tanto para iniciar quanto para terminar. A rotina de leitura também possui um diagrama parecido.

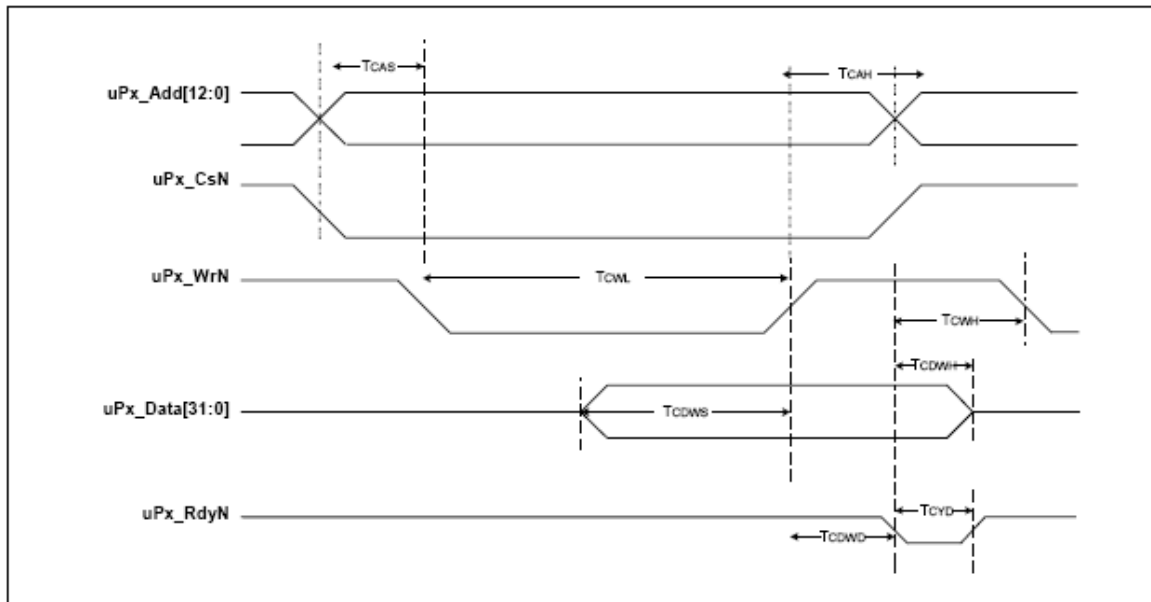


Figura 4.11 - Diagrama de tempo para um ciclo de escrita do IXF1104 [26]

Além das três principais interfaces citadas acima, o controlador MAC possui as seguintes interfaces: relógio, JTAG (*Joint Test Action Group*), LEDs e serial.

- **Relógio**

O controlador MAC possui uma única interface de relógio que é utilizada para todos os circuitos através dos PLLs internos. Como a Placa Processadora disponibiliza um sinal de relógio de 125 MHz que atendem os requisitos do IXF, ele também foi utilizado para prover o relógio geral da placa, além de já ser utilizado para a interface SPI-3, não necessitando o uso de osciladores locais.

- **JTAG e LEDs**

A interface JTAG é utilizada para *boundary-scan*, sendo este um recurso comum entre os componentes com encapsulamento BGA (*Ball Grid Array*) utilizado para verificar se não há curto-circuito ou pinos que ficaram sem conexão no processo de montagem. Como os componentes ligados ao IXF não tem o recurso de *boundary-scan*, o único teste

possível é verificar se há curto-circuito entre os pinos, pois é necessária um outro dispositivo que provê este recurso para verificar se a ligação no processo de soldagem foi executada com sucesso. Existem várias ferramentas no mercado que executam este teste e a que nós utilizamos é a *OnTAP* da Flynn Systems [31].

A interface de LEDs indica o status de cada uma das portas GbE. Como o IXF1104 disponibiliza os dados através de uma interface serial, poderíamos utilizar deslocadores (*shift registers*) ou um controlador específico para esta função. Decidimos usar o controlador por facilitar o roteamento da placa. Os LEDs indicam se há conexão (*link*) ativa e se as portas estão recebendo e/ou transmitindo dados.

- **CLI e Serial**

Foi desenvolvida uma interface de linha de comandos (*Command Line Interface* - CLI) para interagir com o microcontrolador e através dela configurar e monitorar as funções da placa. A interface utilizada entre a placa e o microcomputador utilizado para visualizar a CLI foi a RS232C, pois o IXF também possui uma porta de transmissão e recepção assíncrona universal (*Universal Asynchronous Receiver-Transmitter* - UART). A Figura 4.12 mostra a tela inicial da CLI. As funções desenvolvidas para o controle da placa são:

- *Init*: executa as configurações iniciais do IXF. Devido ao grande número de registradores para ser configurado, esta rotina deixa o MAC apto a operar assim que o número 1 é selecionado;

- *Read*: faz a leitura do valor num registrador a partir do endereço desejado. Ao entrar com o endereço em hexadecimal do registrador, o valor correspondente é mostrado na tela;

- *Write*: escreve um valor determinado num registrador. Ao entrar com o endereço do registrador e o valor correspondente em hexadecimal, o registro é realizado imediatamente no IXF;

- *Local Loopback*: executa *loopback* na interface SPI-3;

- *External Loopback*: executa *loopback* nas interfaces GbE;

- *Reset IXF1104*: executa um *reset* físico no controlador MAC.



Figura 4.12 - Tela principal da CLI

O código desenvolvido para o microcontrolador encontra-se no Apêndice I.

4.2.1.3 Circuito de alimentação da placa

O IXF1104 requer três diferentes valores de tensão para funcionamento: 1,8 Vdc, 2,5 Vdc e 3,3 Vdc. Os demais componentes da placa, com exceção do controlador de LEDs que utiliza 5,0 Vdc, requerem apenas 3,3 Vdc para funcionamento.

O controlador MAC ainda requer uma sequência de *power-up* (alimentação dos circuitos) e *power-down* (retirar alimentação dos circuitos) que foi obedecida utilizando um circuito para seqüenciamento. Este seqüenciamento é exigido para não danificar a estrutura do circuito de proteção contra descargas eletrostáticas que o IXF1104 possui.

A Figura 4.13 mostra a sequência com que as tensões de alimentação do controlador MAC devem obedecer. Note que não há exigência para a tensão de 3,3 Vdc. Há também um tempo de estabilização de 10 μ s (mín.) que deve ser obedecido.

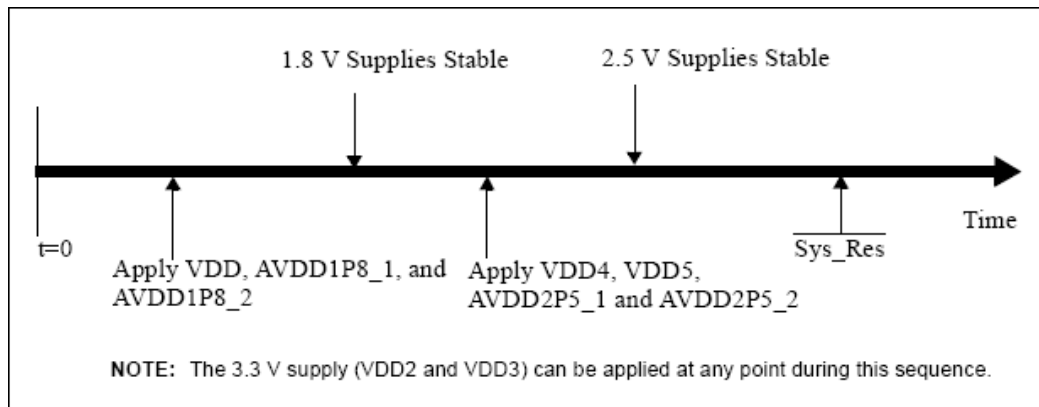


Figura 4.13 - Seqüência de alimentação do IXF1104 [26]

A seqüência de *power-down* é especificada pelo fabricante do IXF para ser inversa a de *power-up*, ou seja, a tensão de 2,5 Vdc deve ser retirada primeiro que a de 1,8 Vdc. Como informado, utilizamos um circuito específico para esta função. A Figura 4.14 ilustra o diagrama elétrico do seqüenciador de alimentação.

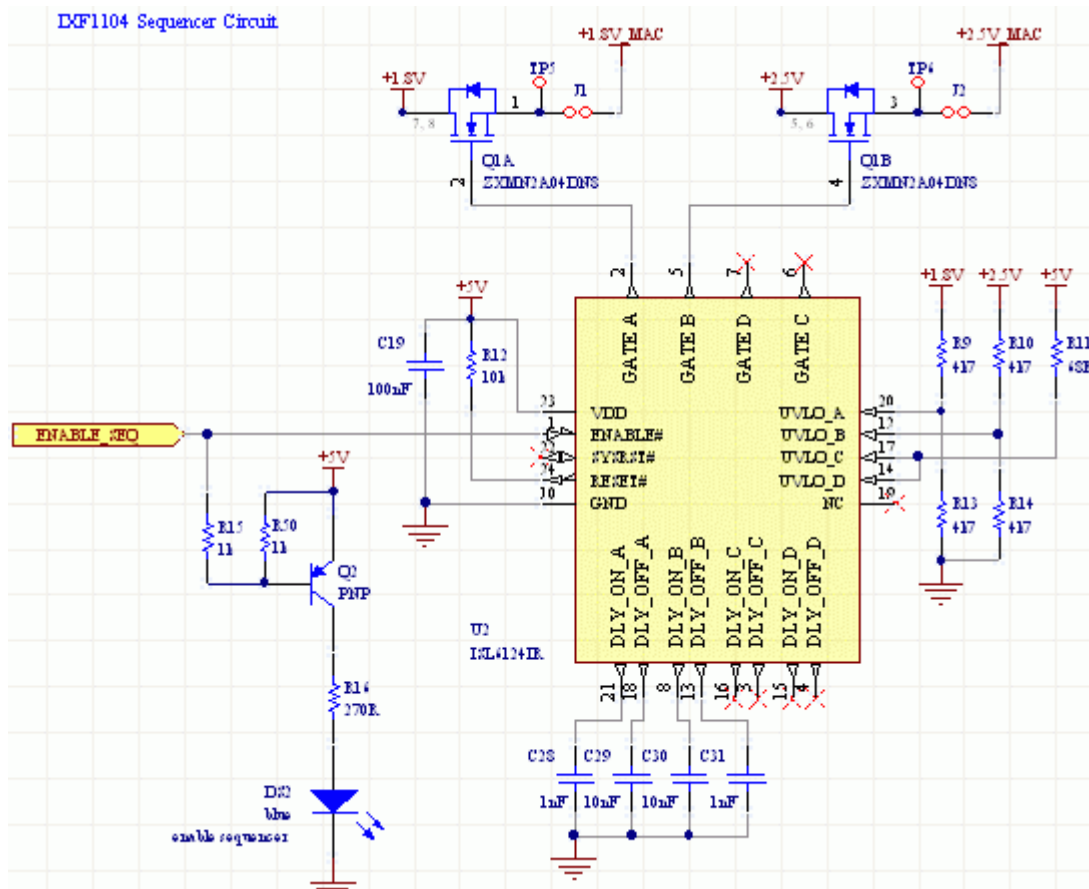


Figura 4.14 - Circuito de seqüenciamento da Placa de Acesso aos Clientes

No circuito de seqüenciamento temos os capacitores C28, C29, C30 e C31 que controlam o atraso entre uma alimentação e outra nas seqüências de *power-up* e *power-down* referente ao valor da capacitância e o sinal ENABLE-SEQ que quando em nível lógico ‘0’ habilita a seqüência para ligar o IXF e ‘1’ desliga o controlador MAC.

4.2.2 Layout da Placa de Acesso aos Clientes

Com o projeto elétrico finalizado, o próximo passo foi fazer o *layout* (roteamento) da placa. A ferramenta utilizada para o projeto elétrico e o *layout* foi o *Altium Designer*, da Altium [32]. Esta ferramenta permite uma interação total entre o projeto elétrico e o *layout*, possibilitando o uso de vários instrumentos de verificação, como a checagem de erros de conexões (*Electrical Rule Check* - ERC) que é utilizado para verificar, por exemplo, se há terminais de componentes sem conexão, a checagem de erros do layout (*Design Rule Check* - DRC) que verifica, por exemplo, se há curto-circuito em alguma trilha, integridade de sinais e outros, para evitar que erros comuns ocorram.

O primeiro passo foi identificar os itens críticos do projeto, e a integridade dos sinais de dados é um deles. Cuidados devem ser tomados quando se trabalha com sinais de alta velocidade, ou seja, for considerado um sinal de alta frequência. É importante avaliar tomar cuidado com a linha de transmissão onde o sinal irá trafegar, quando ela é maior que $\frac{1}{4}$ do comprimento de onda (λ) do sinal.

Nesta placa, como já descrito, temos sinais com taxa de até 1,25 Gbit/s e para detectar se o sinal é de alta frequência e facilitar o cálculo, aproximamos este sinal a uma senóide de 1,25 GHz, não levando em conta às harmônicas e os requisitos para os tempos de subida e descida.

Para encontrar o comprimento de onda consideramos a constante dielétrica (ϵ_r) da placa igual a 4, e o resultado pode ser observado a seguir:

$$\lambda = \frac{cm}{F}, \text{ onde}$$

$$cm \text{ (velocidade da luz no material)} = \frac{3 \times 10^8 \text{ m/s}}{\sqrt{\epsilon_r}} = 1,5 \times 10^8 \text{ m/s};$$

$$F \text{ (frequência)} = 1,25 \times 10^9 \text{ Hz};$$

$$\lambda = \frac{1,5 \times 10^8}{1,25 \times 10^9} = 120 \text{ mm};$$

$$\text{então, } \frac{\lambda}{4} = 30 \text{ mm}$$

Seguindo a regra citada, linhas maiores que 30 mm são consideradas de alta frequência. Como não é possível conectar todos os componentes críticos num espaço reduzido mantendo as linhas menores que 30 mm de comprimento, algumas técnicas de roteamento foram adotadas para manter a integridade dos sinais de dados, como impedância característica e espaçamento entre um par diferencial e outro para evitar *crosstalk*.

4.2.2.1 Roteamento da PCI

Com todos os símbolos mecânicos (*footprints*) criados e importados do projeto elétrico, há ainda uma etapa anterior ao roteamento, que é o posicionamento deles e a delimitação mecânica da placa.

A delimitação mecânica pode ser estabelecida por várias razões; neste caso, foi determinada pela dimensão da Placa Processadora. O posicionamento dos componentes é diretamente ligado ao número de camadas (*layers*) utilizadas para o roteamento da placa, ao comportamento do projeto em ensaios de compatibilidade eletromagnética e também na integridade de sinais, pois dependendo da posição onde os componentes estão, é preciso ou não cruzar as linhas de dados e quanto mais existir sinais cruzados, mais camadas e interferências podem surgir.

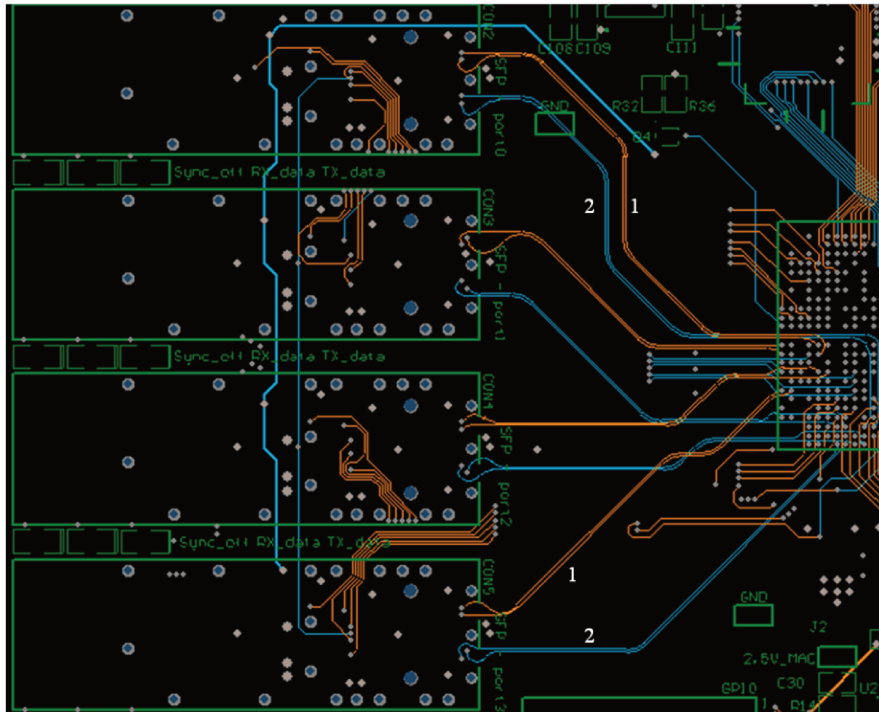


Figura 4.15 - Posicionamento dos componentes para facilitar o roteamento

A Figura 4.15 mostra um exemplo de como o posicionamento adequado dos componentes pode facilitar o roteamento da PCI. Note pelas linhas (1) e (2) que foram necessárias o uso de 2 camadas apenas para ligar o IXF1104 aos conectores dos 4 módulos ópticos. Se o módulo CON2 (acima) estivesse no lugar do CON5 (embaixo), muito provavelmente mais camadas seriam necessárias para fazer o roteamento.

Após o posicionamento dos *footprints*, foi dado início ao roteamento propriamente dito. É importante priorizar o roteamento dos sinais mais críticos (de alta frequência) em primeiro lugar, pois os demais não precisam ter comprimentos baseados em sinais de relógio e também não há problema com a inserção de furos de passagem para troca de camadas.

Para os sinais de alta frequência, as linhas dos sinais não diferenciais, como do barramento SPI-3, foram construídas de tal forma a manter uma impedância característica de $50\ \Omega$, e para os sinais diferenciais, transmissão e recepção dos módulos ópticos, esse valor passa para $100\ \Omega$.

É muito importante manter esses valores ao longo da linha, pois o não casamento de impedância pode causar reflexões no sinal. Por exemplo, se o valor do descasamento da

impedância da linha for de 10%, isto resultará em 5% do sinal refletido [34]. Um fator que propicia a não continuidade deste valor na linha são os furos de passagem, ou vias, que devem ser evitadas ao máximo por inserirem capacitâncias e indutâncias parasitas que afetam a impedância em função da frequência.

Para identificar a geometria apropriada para construir as linhas com impedância controlada, foi utilizado o aplicativo *Zcalc*, visualizado na Figura 4.16, que calcula as espessuras, o atraso de propagação e a capacitância das trilhas a serem roteadas de acordo com a estrutura da PCI.

Nesta placa utilizamos sinais não diferenciais nas camadas internas (*MidLayers*) e externas (*Top e Bottom Layers*) e sinais diferenciais apenas nos *MidLayers*.

Utilizando o *Zcalc*, chegamos nas dimensões informadas na Tabela 4.1:

Camadas	Largura p/ linha não diferencial	Geometria p/ linha diferencial
Externas	7 mils	—
Internas	6 mils	5 mils - 6 mils ¹ - 5 mils

Tabela 4.1 - Geometria das trilhas para sinais de alta frequência da Placa de Acesso ao Anel Óptico

¹O valor de 6 mils informado na tabela acima, sendo 1 mils igual a 1 milésimo de polegada, é referente a separação entre as duas trilhas que formam o sinal diferencial.

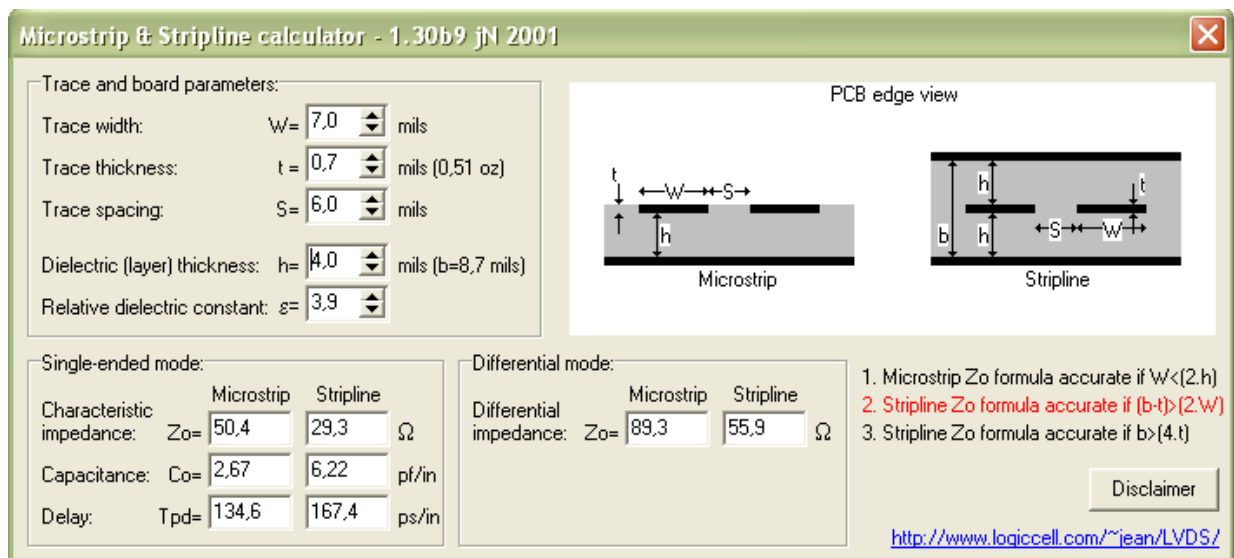


Figura 4.16 - Aplicativo para cálculo de geometria de trilhas

A Figura 4.17 mostra a construção da placa de circuito impresso. Pode-se observar a espessura dos dielétricos (*Prepeg* e *Core*) e o número de camadas para o roteamento dos sinais (*Top*, *Bottom* e *Mid Layers*), para as diferentes tensões de alimentação (*InternalPlane*). Os parâmetros, constante dielétrica e espessura do cobre, não são visíveis na figura abaixo, mas também são necessários para o cálculo da geometria das trilhas. Outro ponto de observação é que a construção da PCI é simétrica, caso não fosse, provavelmente iríamos notar um empenamento da placa após o processo de fabricação.

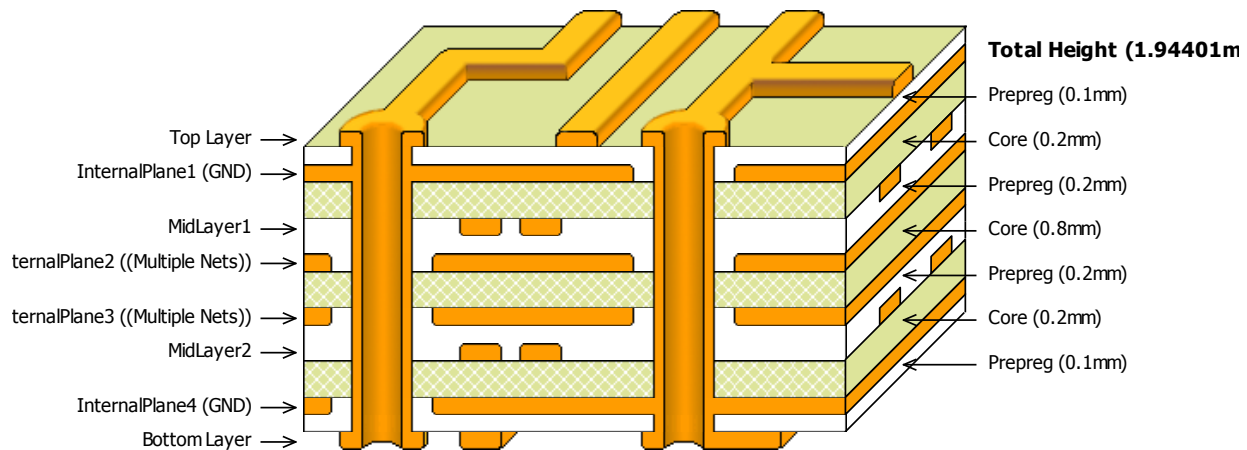


Figura 4.17 - Estrutura da PCI da Placa de Acesso aos Clientes

Pode-se observar na Figura 4.17 as estruturas *microstrip* (formada com as camadas externas e o plano subsequente) e *stripline* (formada por dois planos e uma camada interna). Estes planos além de conduzir as tensões e terra de alimentação dos componentes, servem para isolar os sinais entre as camadas e formar as estruturas citadas.

O segundo ponto a ser considerado é quando se trabalha com barramento de dados. Em nosso caso, a interface SPI-3 do controlador MAC trabalha com um relógio de 155 MHz e por isso o fabricante recomenda que a diferença máxima do comprimento das trilhas dos sinais de dados e de relógio seja de 500 mils (aproximadamente 12 mm). Mas como esse barramento é conectado à Placa de Processamento, tivemos que nos preocupar com as duas placas e quando fomos informados que nas linhas que iríamos utilizar para essa interface não estavam equalizadas, por ser de uso genérico, não foi possível utilizar a equalização automática da nossa ferramenta para roteamento.

Então, foi necessário levantar as diferenças das trilhas da outra placa e compensar nas nossas linhas manualmente, linha por linha, para quando as duas placas fossem conectadas, a diferença máxima permitida do barramento fosse respeitada.

Essa equalização de comprimento das trilhas entre os sinais de dados e o relógio é sempre exigida nestes casos para que não corram o risco deles serem amostrados fora do tempo. Na Figura 4.18 podem ser visualizadas as “serpentinhas” criadas para equalização nas linhas do barramento SPI-3.

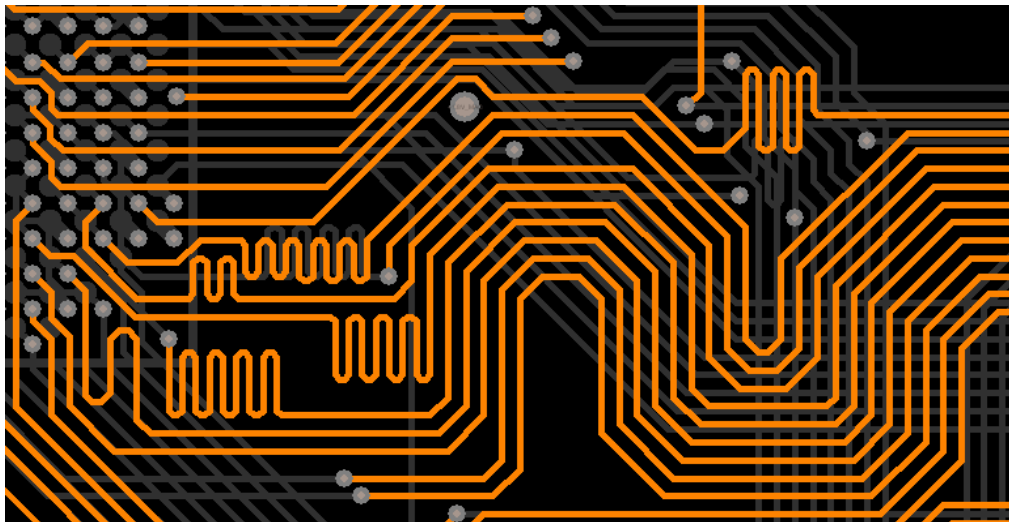


Figura 4.18 - Equalização das linhas da interface SPI-3

Pode ser observado na figura acima outro ponto importante quando se trabalha com barramento de dados que é evitar o *crosstalk*, que significa quando um sinal de uma trilha interfere na linha adjacente. Isto é evitado mantendo um espaço mínimo de 3 vezes a espessura do dielétrico referente entre os sinais do barramento.

Não é apenas a equalização de comprimento entre sinais e relógio que devem ser levados em consideração quando se trata de linhas de alta frequência, no caso de pares diferenciais o cuidado tem que ser o mesmo, mas com relação ao comprimento das trilhas que formam o par propriamente dito. A Figura 4.19 mostra o trabalho realizado para equalizar o comprimento de dois pares diferenciais. Note que a trilha debaixo de cada par possui um alongamento provocado pela equalização.

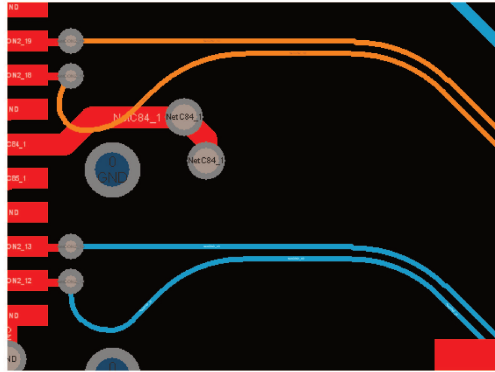


Figura 4.19 - Equalização de comprimento dos pares diferenciais

É bom relatar que as dimensões das trilhas sugeridas pelo aplicativo *Zcalc* são revisados pelo fabricante da PCI sendo capaz de fazer pequenas correções na geometria das linhas ou estrutura da placa se necessário. Os diversos projetos já feitos utilizando este aplicativo mostraram que seu resultado é bem próximo da análise dos fabricantes.

Com os sinais críticos roteados, os restantes foram realizados seguindo as práticas de roteamento, mas sem a preocupação com equalização ou casamento de impedância.

O último passo do roteamento é criar os planos internos para levar alimentação até os componentes que possuem diversos pinos dedicados para este fim, como é o caso de componentes com encapsulamento BGA. Ao invés de conectar uma trilha em cada via, ligamos estas vias num único plano, economizando camadas desta forma.

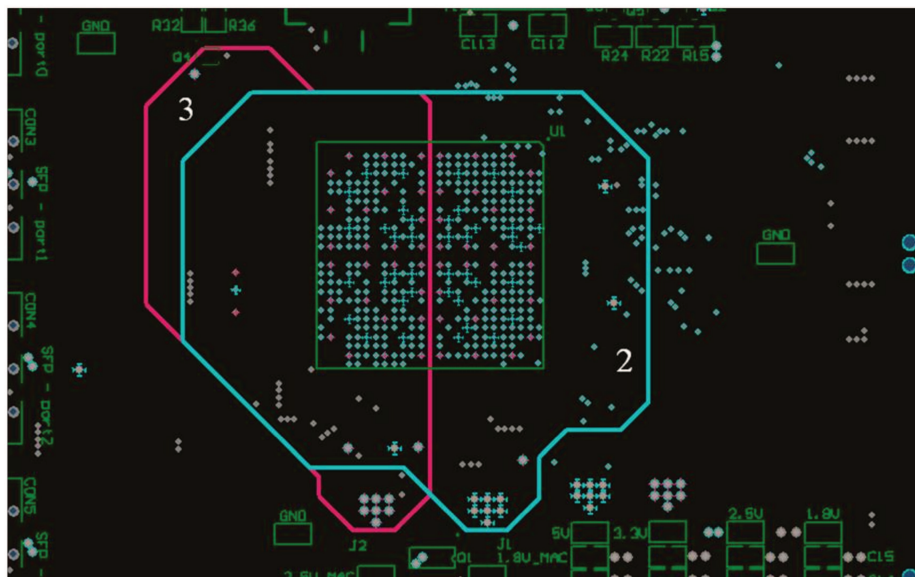


Figura 4.20 - Planos de alimentação

A Figura 4.20 ilustra o desenho de uma divisão no plano de alimentação (*split planes*) que é utilizada para alimentar o IXF1104. Como exemplo, podemos observar um recorte no plano 2 (*InternalPlane2*) e outro no plano 3 (*InternalPlane3*). O plano 2 é conectado em 5,0 Vdc e alimenta com esta tensão todas as vias conectadas somente fora da área recortada, pois dentro desta área ele fornece 3,3 Vdc nas conexões. O mesmo acontece com o plano 3, mas com os valores de tensão de 2,5 Vdc e 1,8 Vdc. Com uma análise rápida, é possível notar que o componente BGA (IXF1104), no centro da figura, não recebe 5,0 Vdc pelo plano de alimentação.

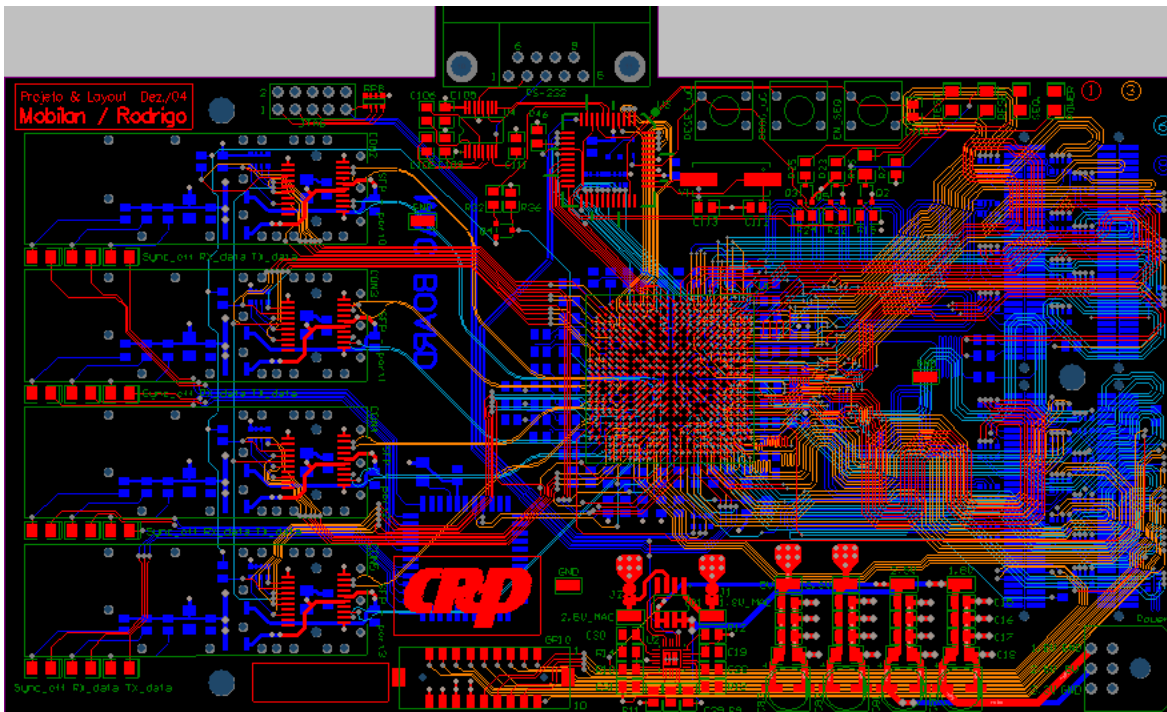


Figura 4.21 - Placa de Acesso aos Clientes roteada

A Figura 4.21 mostra a placa totalmente roteada. Para validar o roteamento, é utilizado o recurso DRC, que verifica se houve violação de alguma regra pré configurada, e também pode ser feita uma análise da integridade de sinais, que irá analisar se o roteamento das linhas de alta frequência possui alguma inconsistência que irá prejudicar os sinais a serem trafegados.

Feitas as análises e verificações, são gerados os arquivos finais no formato Gerber, padrão aceito nas ferramentas CAM (*Computer-Aided Manufacturing*) utilizadas pelos fabricantes de PCI, e enviados para fabricação da placa.

4.2.3 Montagem e testes da placa

Hoje é possível enviar os arquivos de fabricação e a lista de componentes para um fabricante e receber a placa já montada sem se preocupar com a compra dos componentes. Esse procedimento chamado de *turn-key* é muito eficiente para montagem de protótipos, mas possui um custo elevado, pois por enquanto só é possível executá-lo fora do país.

Em nosso caso, como queríamos fazer todo o processo aqui para adquirir conhecimento também na montagem de componentes BGA, resolvemos acompanhar toda a montagem numa empresa aqui no Brasil. Outro requisito do projeto foi desenvolver as placas seguindo requisitos dos fabricantes disponíveis no país para uma possível diminuição de custos. Com a placa fabricada pela Multek do Brasil em mãos, uma das únicas empresas que atendiam os requisitos do projeto, acompanhamos a montagem do componente BGA numa empresa de Campinas-SP, e terminamos de montar os componentes no Laboratório de Hardware Eletrônico (LDHE) do CPqD. A Tabela 4.2 informa os requisitos mais relevantes do fabricante que foram seguidos no roteamento da placa de circuito impresso.

Parâmetros	Mín.	Máx.
Número de camadas	2	16
Diâmetro final para furos metalizados	0,15 mm	-
Espaço entre trilhas p/ camadas internas/externas	0,10 mm	-
Largura da trilha p/ camadas internas/externas	0,10 mm	-
Acabamento ENIG	sim	-

Tabela 4.2 - Requisitos para fabricação de PCI

O acabamento ENIG (*Electroless Nickel/Immersion Gold*) [35], processo este que deposita ouro (Au) nas “ilhas” ao invés de solda quando utilizado o método HASL (*Hot Air Solder Leveling*), é exigido quando se trabalha com componentes com encapsulamento BGA, pois permite um maior controle nas imperfeições na superfície onde cada *ball* “pino” será soldado, evitando possíveis “soldas frias”.

A etapa posterior ao término da montagem foi o teste da placa. Foram montadas 4

unidades e todas obtiveram sucesso nos testes sem a necessidade de ajustes ou reprojeção.

Os testes começaram com a medição das tensões de alimentação dos circuitos da placa 1,8 Vdc, 2,5 Vdc, 3,3 Vdc e 5,0 Vdc, para verificar quedas e ruído, teste do circuito de seqüenciamento de alimentação e osciladores. Após isso, foi gravado o código no microcontrolador e verificado o funcionamento das funções desenvolvidas. Foram encontrados problemas na rotina de escrita dos registradores, mas foi corrigida rapidamente.

Com controle do IXF1104, foi possível iniciar os testes de tráfego pelas interfaces ópticas e elétricas. Iniciamos o teste de tráfego de dados pelas interfaces ópticas fazendo um *external loopback* nas interfaces do IXF1104. Com esse *loopback* foi possível visualizar, através do analisador de protocolo *Ethereal*, os dados enviados sendo retornados em cada porta óptica utilizando os módulos SFPs de 1 GbE. O aplicativo utilizado para gerar os pacotes de dados foi o *Packetgen*, este que é incorporado ao Linux, sistema operacional utilizado no computador do teste. Este teste comprovou que o IXF1104 estava operacional e que não havia problemas nas linhas de alta frequência, as mais críticas. Neste momento já tínhamos comprovado o funcionamento de todos os circuitos da placa, só faltando a conexão da interface SPI-3. A Figura 4.22 mostra o esquema e o caminho utilizado pelos pacotes de dados entre as placas. Neste teste os pacotes eram encaminhados até o IXF1104 e retornavam graças ao *loopback* configurado através da CPU.

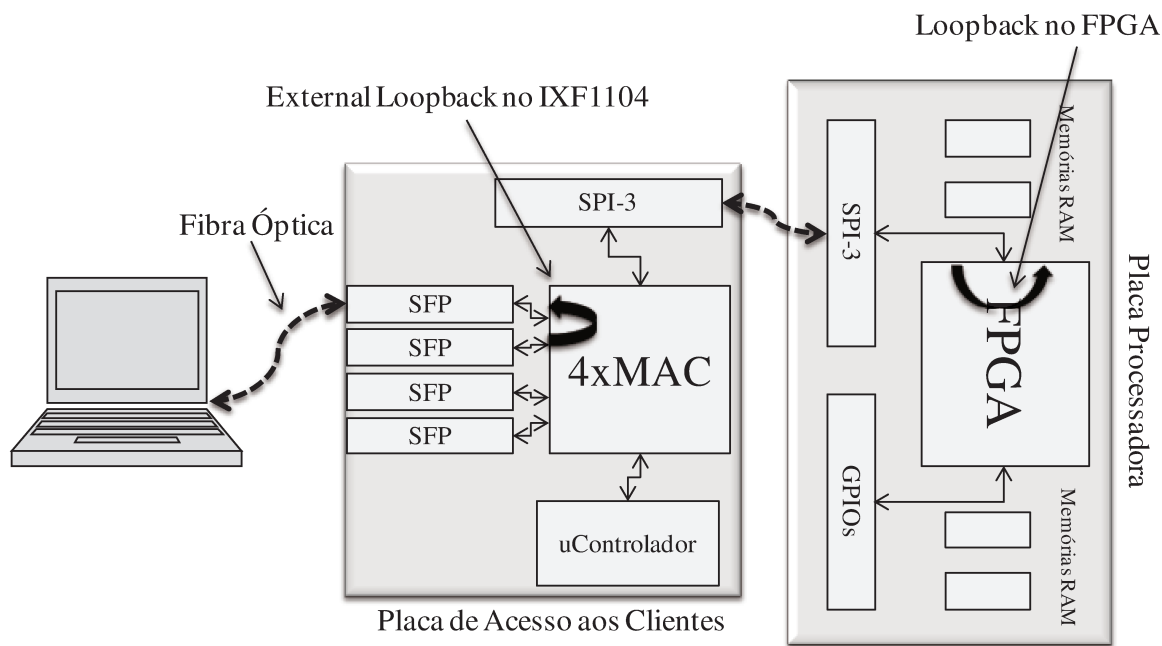


Figura 4.22 - Esquema do teste realizado na placa de Acesso aos Clientes

Para testar a interface paralela SPI-3, desenvolvemos um *loopback* no FPGA da Placa Processadora, apenas ligando os pinos de entrada na saída. Com isso não foi necessário utilizar o *local loopback* do IXF. Neste teste foi utilizando o mesmo método anterior, e foi possível comprovar o funcionamento total do projeto visualizando os dados no analisador de protocolo. No teste, os pacotes de dados entram pela interface óptica da Placa de Acesso aos Clientes, são tratados pelo controlador MAC e enviados para a Placa Processadora e são devolvidos para o IXF, que por sua vez os transmite através dos SFPs até o computador.

Este teste foi feito nas quatro interfaces sem perda de pacotes ou erros, que comprovou também que não há problema no barramento SPI-3 e no posicionamento dos conectores que interligam as duas placas.

A Figura 4.23 mostra a Placa de Acesso aos Clientes conectada na Placa Processadora no momento dos testes.



Figura 4.23 - Placa de Acesso aos Clientes sendo testada

O único problema de *hardware* encontrado no teste da placa foi com o conector do barramento SPI-3. Por motivos desconhecidos, pois eles se encontravam em embalagem apropriada, todos estavam com oxidação nos terminais de contato, gerando mau-contato na conexão. Após uma limpeza dos conectores não observamos mais problemas. Também vale relatar que não foi necessário realizar o teste de *boundary scan*. A Figura 4.24 mostra um conector que cortamos a capa protetora para verificar a oxidação.



Figura 4.24 - Conector de dados com os contatos oxidados

Na Figura 4.25 é possível observar o posicionamento dos conectores de dados da interface SPI-3. Eles são responsáveis por conduzir os sinais de dados de uma placa a outra.

Finalizados os testes, o passo seguinte foi ativar a segunda placa, a Placa de Acesso ao

Anel Óptico, cujo desenvolvimento será descrito na próxima seção.



Figura 4.25 - Vista inferior da Placa de Acesso aos Clientes

4.3 Placa de Acesso ao Anel Óptico

A segunda placa desenvolvida foi a Placa de Acesso ao Anel Óptico. Esta placa leva este nome pois é ela que faz a interface entre o nó de acesso e a rede física. A placa possui quatro interfaces ópticas, desta vez proprietárias mas utilizando os mesmos módulos SFPs da Placa de Acesso aos Clientes, utilizadas para transmissão dos canais de dados e de controle, através do anel óptico da rede.

4.3.1 Projeto elétrico da Placa de Acesso ao Anel Óptico

Na Figura 4.26 podemos ver o diagrama de blocos do módulo. O principal componente do módulo é o SerDes que tem a função de sincronizar num curto tempo, aproximadamente 1 μ s, os pacotes de dados recebidos de forma serial e paralelizar em 10 bits e vice-versa.

Os dados recebidos do anel óptico são convertidos para o domínio elétrico nos módulos SFPs e são enviados para a Placa Processadora através de um conector padrão PCI. Somente o padrão mecânico é PCI, devido a Placa Processadora possuir este formato, e não os níveis lógicos dos sinais.

Os componentes utilizam 2,5 Vdc e 3,3 Vdc como alimentação e ainda existem os LEDs que indicam ausência de sinal (*Loss of Signal* - LOS) para cada módulo óptico.

Figura 4.26 - Diagrama de blocos da Placa de Acesso ao Anel Óptico

4.3.1.1 Descrição do funcionamento do circuito integrado SerDes

Como já dito, este é o componente responsável por sincronizar os pacotes de dados provenientes das interfaces óptico-elétrica e formatá-los para a interface paralela da Placa Processadora.

de alinhamento é '0011111010'.

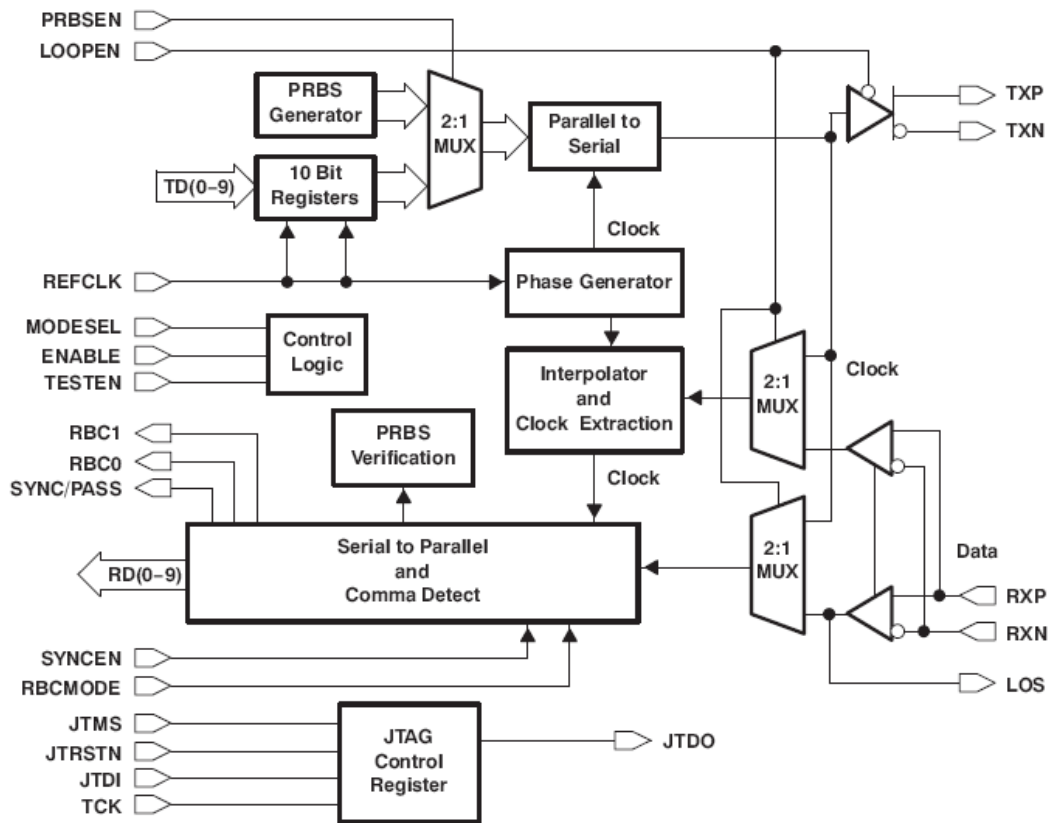


Figura 4.27 - Diagrama de blocos do TLK2201A [28]

Os dados, já no formato paralelo de 10 bits, são enviados para a Placa Processadora em fase com o sinal de relógio RBC (*Receive Byte Clock*) que são amostrados em cada borda de subida. Na Figura 4.28 é possível visualizar as formas de onda dos sinais RBC, SYNC e dos dados nos pinos de saída RD(0-9), sendo o bit 0 o mais significativo. Note o sinal no pino SYNC a cada caractere de alinhamento recebido.

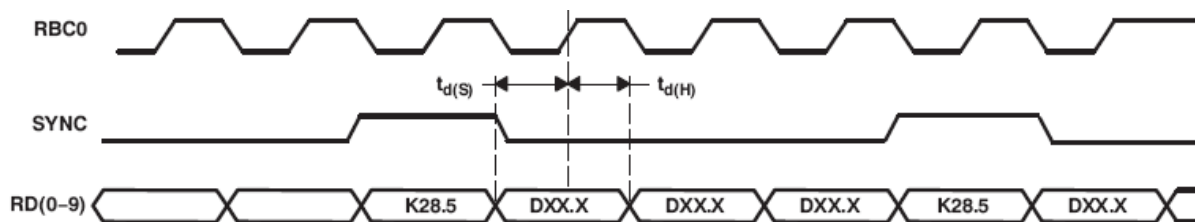


Figura 4.28 - Formas de onda da recepção dos dados do SerDes [28]

Na transmissão, os dados são enviados pela Placa Processadora no formato paralelo de 10 bits, em fase com o relógio de referência de 125 MHz, nos pinos TD(0-9), sendo o bit

O mais significativo, e são transmitidos em forma serial na taxa de 1,25 Gbit/s para os módulos ópticos através das saídas TXP/TXN.

Estamos trabalhando sempre com 10 bits no formato paralelo devido a utilização da codificação de linha 8b/10b [36]. Este método transforma as palavras de 8 bits em 10 bits, seguindo um padrão estabelecido de codificação. Isto explica, por exemplo, a taxa dos pacotes de dados para Gigabit Ethernet no meio físico ser de 1,25 Gbit/s ao invés de 1 Gbit/s.

Este padrão é utilizado para evitar seqüências longas de ‘0s’ ou ‘1s’, pois estas podem causar problemas de sincronização, coeficiente DC e aumento na demanda por banda em baixas freqüências no canal de transmissão.

Outras funcionalidades do componente são LOS, pseudo seqüência de bits aleatória (*Pseudo Random Bit Sequence - PRBS*), *loopback* e JTAG teste.

O SerDes também possui a funcionalidade de *boundary scan*, igualmente utilizada no IXF1104, através da interface JTAG. Como este componente não utiliza encapsulamento BGA, esta funcionalidade não é essencial para verificar curto-circuitos entre os pinos, mas muito útil para ajudar na verificação dos componentes que usam tal encapsulamento quando conectados ao SerDes. No nosso caso, poderíamos verificar a integridade da conexão dos pinos do FPGA da Placa Processadora que são ligados ao TLK2201A utilizando uma ferramenta de *boundary scan*.

A função *loopback* permite ativar o modo de funcionamento, através do pino LOOPEN, no qual os dados recebidos nos pinos TD(0-9) sejam transferidos para os pinos RD(0-9) para fins de checar a integridade da conexão até o SerDes. Pois ao verificar que os dados recebidos são os mesmos enviados, podemos dizer que não há problema nesta etapa da comunicação.

A função PRBS testa a interface serial. Ao ser ativada pelo pino PRBSEN, o componente gera uma seqüência de bits não realmente aleatória, de acordo com o padrão 27-1 [28], que é enviada através dos pinos TXP/TXN. Como esta seqüência é conhecida, é possível utilizar um testador de taxa de erro (*Bit Error Rate Tester - BERT*) externo para checar a integridade dos sinais ou utilizar o BERT interno ao componente. Utilizando o BERT interno, basta conectar a saída na entrada do componente e monitorar

o sinal do pino PASS para analisar o resultado, pois se apenas um único bit falhar, o SerDes identifica o erro alertando através deste. O BERT interno também é ativo pelo pino PRBSEN. Com este recurso é possível verificar a integridade de todo o meio físico. Por último temos a função LOS que indica se há nível de sinal suficiente na entrada serial para conseguir locar o extrator de relógio. No nosso caso ligamos esse sinal num LED, mas esse pino pode também ser conectado à uma lógica qualquer.

4.3.1.2 Interfaces e alimentação da placa

Para relembrar, a placa possui 4 interfaces seriais ópticas de 1,25 Gbit/s que trabalham com qualquer protocolo, contando que ele possua os requisitos de sincronização, preâmbulo com no mínimo de 1024 bits, e palavra de alinhamento de 10 bits, k28.5. No outro lado, a placa possui 4 interfaces elétricas de 10 bits para recepção e transmissão com os devidos sinais de relógio operando em 125 MHz. Podemos dizer que esta placa opera com qualquer protocolo na taxa citada e que possua codificação de linha 8b/10b.

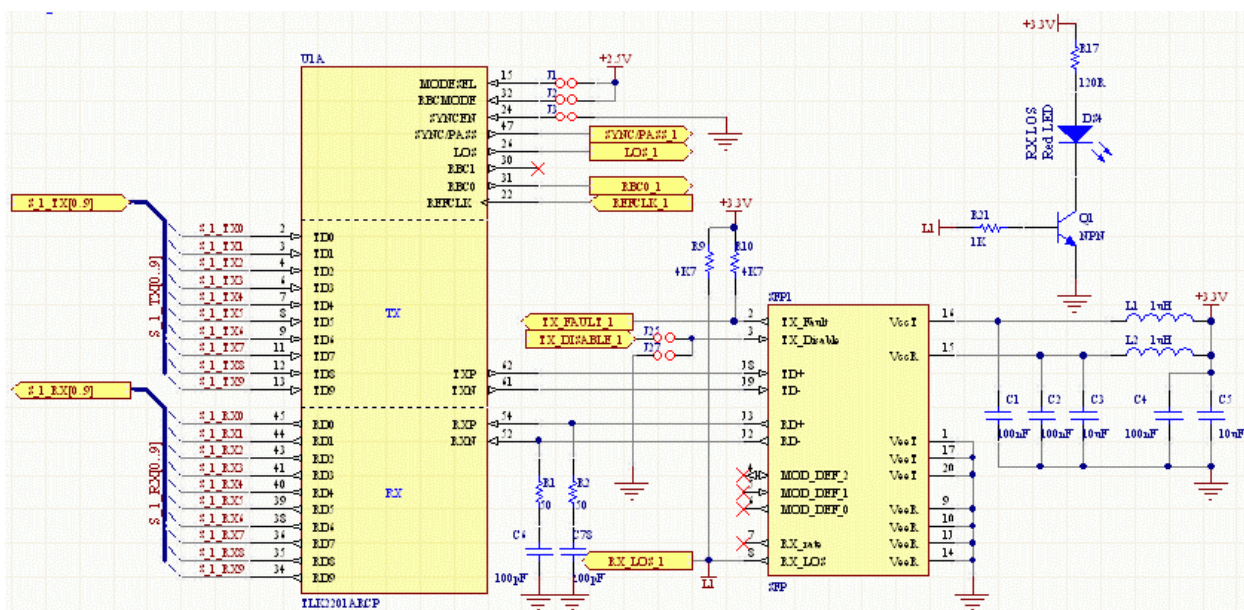


Figura 4.29 - Esquema das conexões das interfaces do SerDes

Podemos verificar na Figura 4.29 as conexões da interface paralela, a esquerda, que são conectadas diretamente ao conector PCI e as ligações do SerDes com o módulo óptico SFP a direita. Este diagrama se repete 4 vezes, que é o número de interfaces da placa. O esquema de filtragem do módulo óptico é o mesmo utilizado pela Placa de Acesso aos

Clientes. Todas as conexões que saem ou entram neste diagrama são direcionadas ao conector PCI para que a lógica de processamento desenvolvida no FPGA tenha acesso.

Como já mencionado, as tensões necessárias para o funcionamento desta placa são 2,5 Vdc e 3,3 Vdc e não requer seqüenciamento. Para facilidade da integração das placas, esta possui entradas para todas as tensões necessárias nas demais placas, assim por meio do conector PCI e outro do mesmo modelo do principal pode-se alimentar as demais placas.

4.3.2 Layout da Placa de Acesso ao Anel Óptico

Para esta placa foram utilizadas todas as técnicas utilizadas na anterior já citadas, como construção de linhas com impedância controlada e equalização do comprimento das trilhas. Diferente da Placa de Acesso aos Clientes, nesta não foi necessário se preocupar com a equalização do comprimento das trilhas no lado da Placa Processadora, pois como a interface que foi utilizada é compatível com o padrão PCI o fabricante já tomou esse cuidado. Portanto, só foi preciso se preocupar em equalizar o comprimento das trilhas desta placa sem compensação.

4.3.2.1 Roteamento da PCI

A única preocupação com o posicionamento dos componentes desta placa foi com a disposição deles em relação às conexões e não com o formato da PCI. Pois diferente da placa anterior, esta não precisou acompanhar o formato da Placa Processadora, tendo apenas o requisito de largura mínima devido ao conector PCI.

Devido à quantidade pequena de componentes, a ausência de encapsulamento BGA e pelo fato da interface paralela ser conectada a um FPGA, que é totalmente flexível com relação ao posicionamento dos pinos, foi possível rotar a placa utilizando apenas 4 camadas. Na Figura 4.30 visualizamos a estrutura da placa e a altura final.

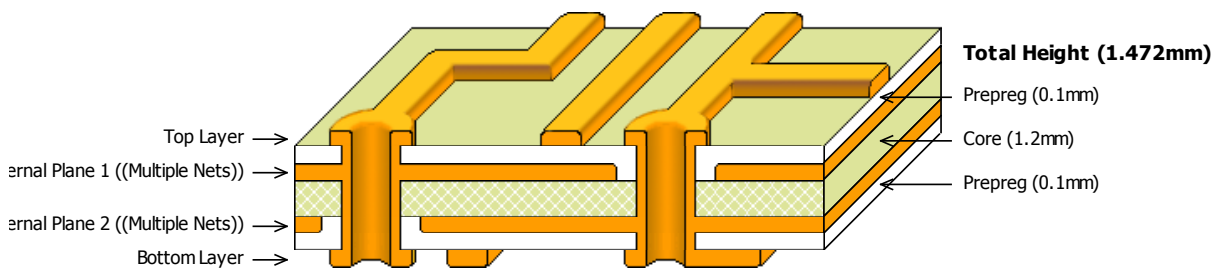


Figura 4.30 - Estrutura da PCI da Placa de Acesso ao Anel Óptico

Como se pode observar, os sinais de dados foram traçados apenas nas camadas externas (*Top* e *Bottom Layers*), e as internas foram utilizadas para formar os planos de alimentação.

A Tabela 4.3 mostra a geometria das trilhas que devem ser roteadas para atingir o valor da impedância de $100\ \Omega$ para as linhas diferenciais e de $50\ \Omega$ para não diferenciais. Estas linhas são referentes às interfaces seriais (diferenciais) e para o barramento paralelo de 10 bits.

Camada	Largura p/ linha não diferencial	Geometria p/ linha diferencial
Externa	6 mils	6 mils - 7 mils - 6mils

Tabela 4.3 - Geometria das trilhas para sinais de alta frequência da Placa de Acesso ao Anel Óptico

A Figura 4.31 mostra a placa totalmente roteada. Note as 4 interfaces ópticas abaixo e o conector PCI acima na figura. É possível também observar as linhas diferenciais entre os SerDes e os módulos ópticos no *Top Layer* e que apenas algumas trilhas do barramento paralelo foram traçadas no *Bottom Layer*. Nas linhas do barramento paralelo também foram criadas as “serpentinhas” para equalizar o comprimento das trilhas.

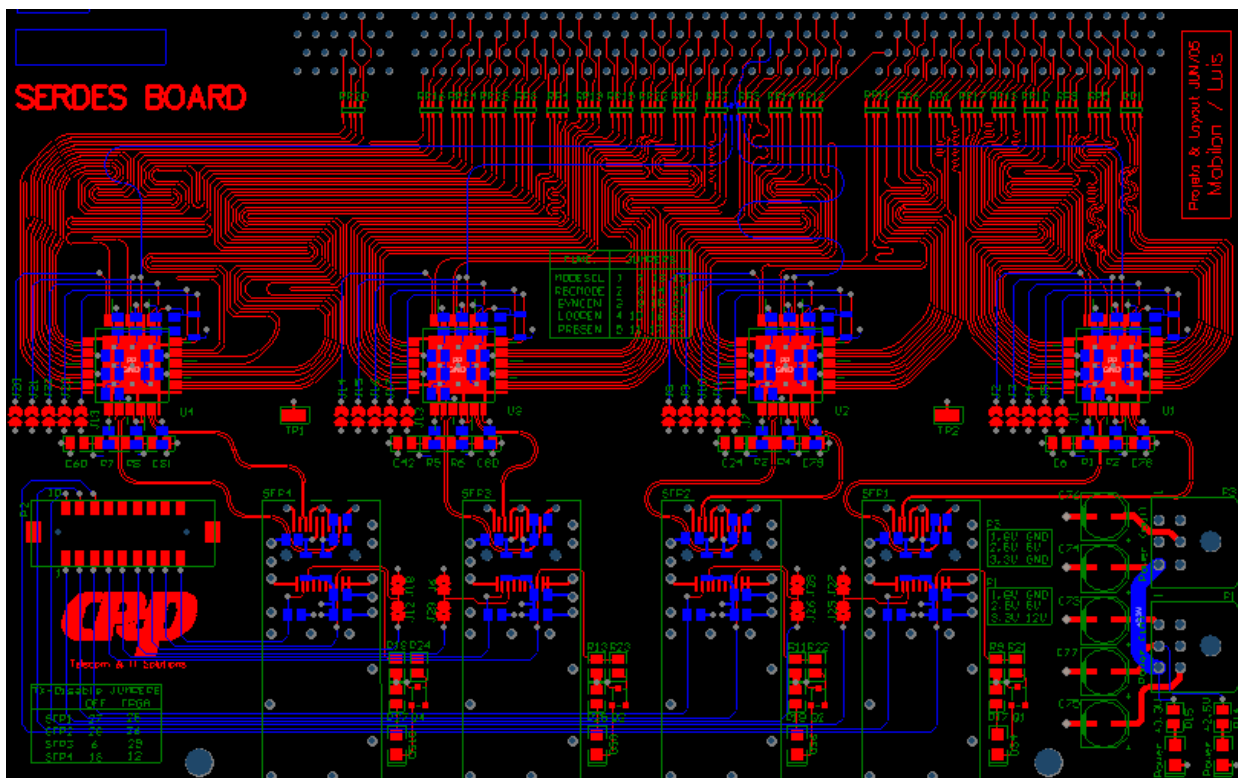


Figura 4.31 - Placa de Acesso ao Anel Óptico roteada

Nesta placa também foi necessário criar *split planes*, pois um plano foi utilizado totalmente para terra e o outro ficou responsável por conduzir as tensões já citadas. Os planos de alimentação também são utilizados para isolar magneticamente os sinais de dados, como já descrito no roteamento da placa anterior.

4.3.3 Montagem e testes da placa

Foram montadas 4 unidades desta placa e todas no CPqD. Esta também foi fabricada na Multek do Brasil e utilizamos o acabamento HASL, pois ela não possui componentes com encapsulamento BGA. Por ter 4 camadas a menos e não utilizar acabamento do tipo ENIG, o custo desta foi menor que a Placa de Acesso aos Clientes. O único problema que tivemos com a montagem da placa foi em relação ao conector PCI, que por possuir muitos pinos (184) e o diâmetro dos furos terem ficado muito pequeno, foi difícil de encaixar todos ao mesmo tempo. Para corrigir esse problema basta aumentar o diâmetro dos furos. Na Figura 4.32 podemos ver a placa montada.

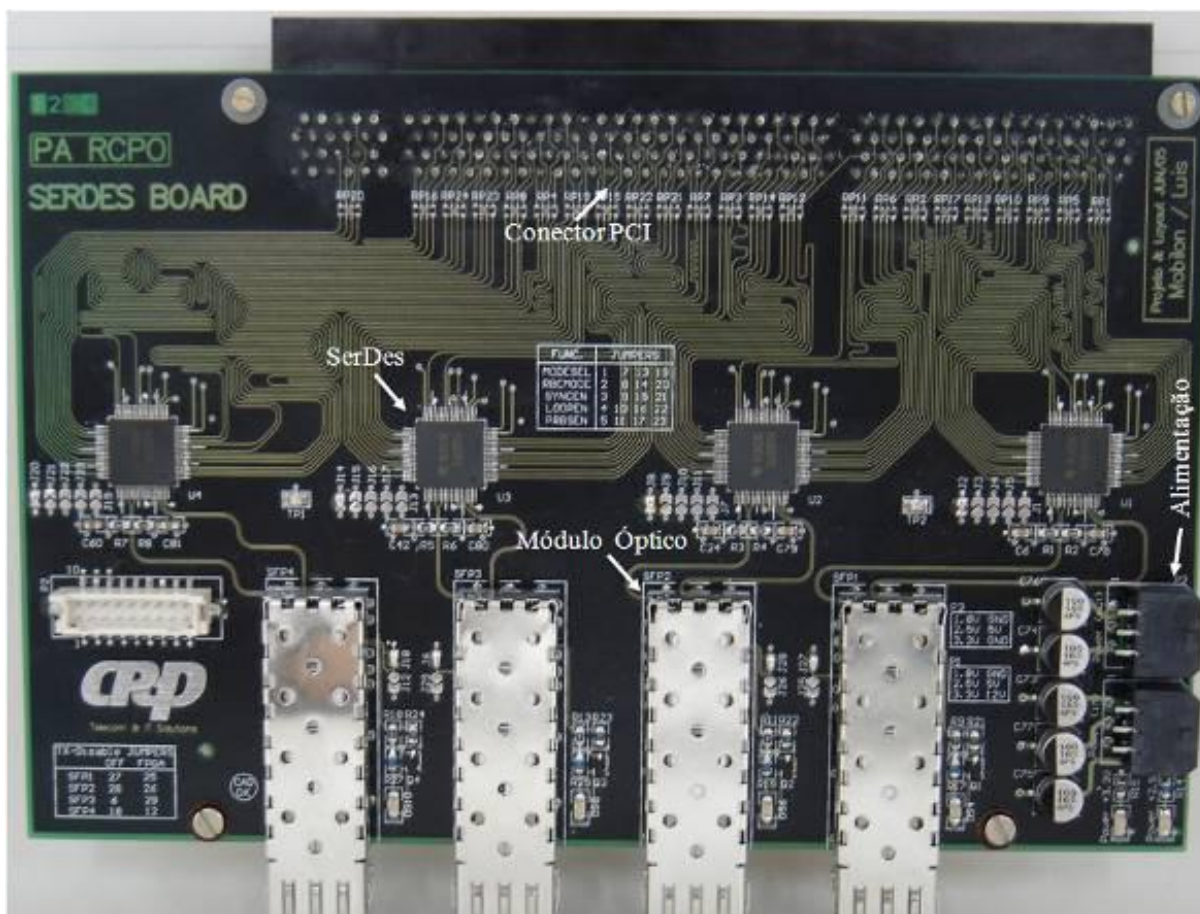


Figura 4.32 - Placa de Acesso ao Anel Óptico montada

Nos testes não foram encontrados problemas, as 4 placas funcionaram sem indicar erro. Primeiro utilizamos o PRBS e o BERT interno para testar a interface serial. Este teste foi realizado nas 4 interfaces das placas e em nenhum momento foi indicado erro. Para testar a interface paralela foi desenvolvido um código no FPGA da Placa Processadora, como no teste da Placa de Acesso aos Clientes, para fazer um *loopback* nos sinais enviados pela placa testada. Como essa placa suporta o padrão Gigabit Ethernet, também foi utilizado o mesmo gerador de pacotes de dados e o analisador de protocolo usado no teste da placa anterior para visualizar os pacotes enviados e recebidos, e checar o funcionamento das interfaces paralelas das 4 unidades montadas. A Figura 4.33 mostra o esquema e o caminho dos pacotes de dados utilizado para testar a placa.

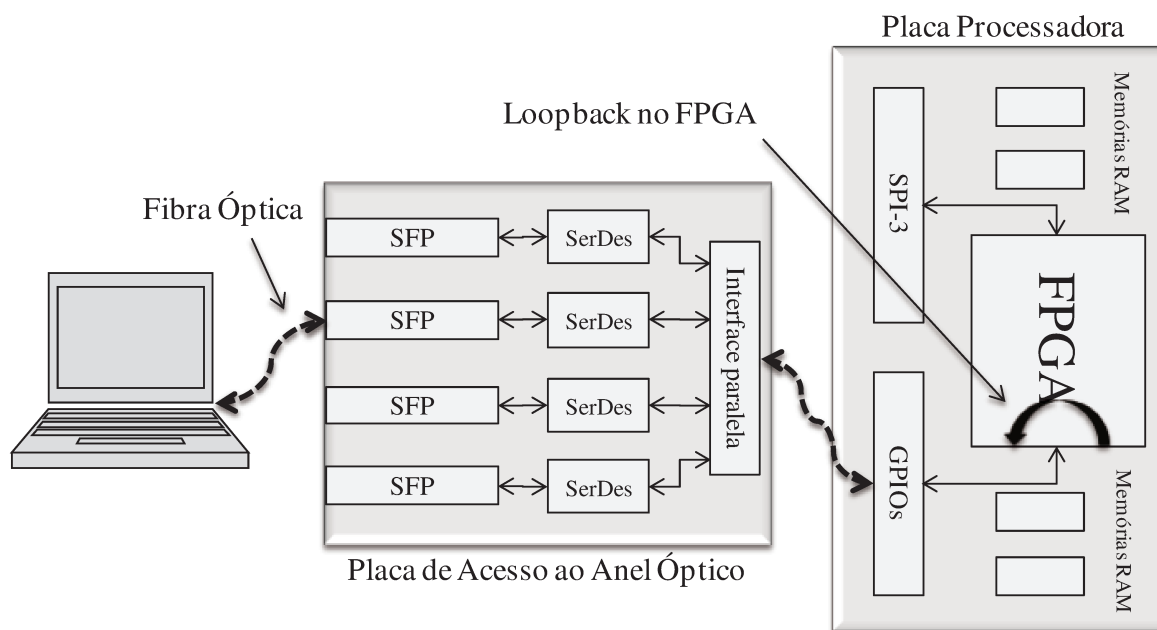


Figura 4.33 - Esquema do teste realizado na placa de Acesso ao Anel Óptico

Os pacotes de dados foram enviados de um computador através da interface óptica da Placa de Acesso ao Anel Óptico. Os pacotes retornaram pela mesma interface até o computador com o analisador de protocolo instalado. Na placa testada esses pacotes passaram pelo SerDes, foram encaminhados até o FPGA da Placa Processadora e retornaram pelo caminho inverso. Como utilizamos na Placa Processadora, não foi necessário utilizar o *loopback* dos SerDes. Com esses dois testes, comprovamos o funcionamento do projeto e da montagem das placas.

4.4 Placa de Chaveamento Óptico

Por último iremos descrever o desenvolvimento da placa responsável pelo chaveamento óptico. Essa placa funciona como uma chave óptica que dependendo da configuração, ao utilizar mais de uma placa, direciona ou não o sinal óptico para o nó de acesso da rede. Essa função é realizada pelo componente principal da placa, que é o amplificador óptico a semiconductor.

Esta é uma placa simples em comparação com as demais descritas, pois a maioria dos componentes utilizados são passivos, como resistores e capacitores.

4.4.1 Projeto elétrico da Placa de Chaveamento Óptico

A Figura 4.34 mostra todos os componentes necessários para o funcionamento correto de um SOA. Podemos observar que estão encapsulados junto ao SOA um termistor com coeficiente de temperatura negativo (*Negative Temperature Coefficient* - NTC) e um refrigerador termoelétrico (*Thermoelectric Coolers* - TEC).

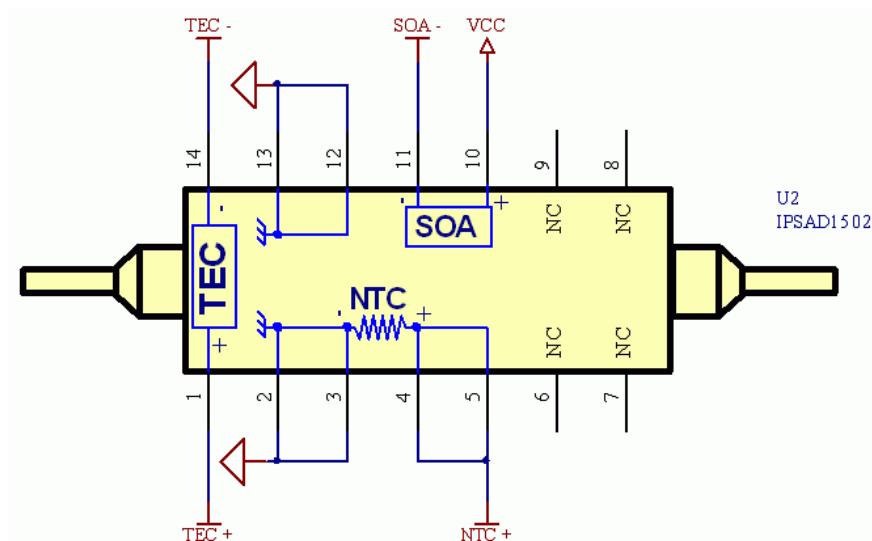


Figura 4.34 - Diagrama esquemático do SOA encapsulado

O NTC é utilizado para monitorar a temperatura através de um circuito de controle. O termistor utilizado possui resistência de 10 k Ω para uma temperatura ambiente de 25°C. Conforme variação da temperatura ambiente, o valor de resistência do NTC é alterado. Por ter um coeficiente negativo, quando o valor da temperatura aumenta, o da resistência diminui. O circuito de controle pode ser observado na Figura 4.35. A alimentação do circuito é de 5 Vdc.

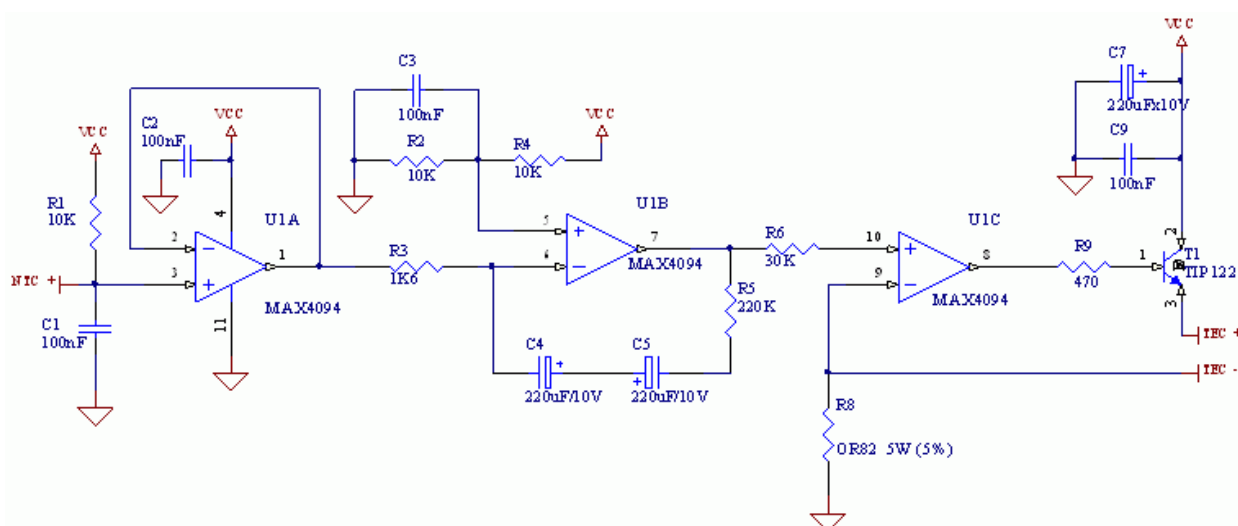


Figura 4.35 - Circuito para controle de temperatura do SOA

O termistor do SOA é conectado na entrada NTC+ do circuito apresentado na Figura 4.35. O circuito integrado (CI) U1A é usado apenas como um *buffer* (amplificador de corrente) para a tensão resultante do divisor entre o R1 e o NTC. Este sinal é utilizado no circuito comparador (U1B). Este circuito então compara o valor de tensão vindo do *buffer* com um valor fixo, resultado da divisão entre R2 e R4. O valor resultante da comparação serve para alimentar o circuito de fonte de corrente controlada por tensão (U1C). Por sua vez esse circuito controla a corrente de alimentação do TEC interno ao encapsulamento, que tem a função de manter o SOA na temperatura especificada. Em nosso caso, esse valor é de 25°C, pois quando a temperatura do NTC estiver com 25°C, sua resistência estará em 10 k Ω e então nas duas entradas do circuito comparador existirá a mesma tensão. Ao variar esta temperatura para mais ou para menos, a resistência do NTC se altera e com isso alterando a saída do circuito comparador. Por sua vez o comparador irá atuar na fonte de corrente para ela fornecer mais ou menos corrente para o TEC manter a temperatura especificada.

O TEC é um dispositivo baseado no efeito *Peltier*, capaz de resfriar ou aquecer o SOA de acordo com o valor e sentido da sua corrente de polarização.

A Figura 4.36 mostra o circuito de polarização do SOA. O ganho do SOA é diretamente proporcional a sua corrente de polarização. Para os primeiros testes do nó da rede, não nos preocupamos em caracterizar seu ganho como amplificador, mas sim pelo seu uso

como uma chave rápida. Por isso, especificamos a corrente de polarização no valor típico indicado pela folha de dados do componente.

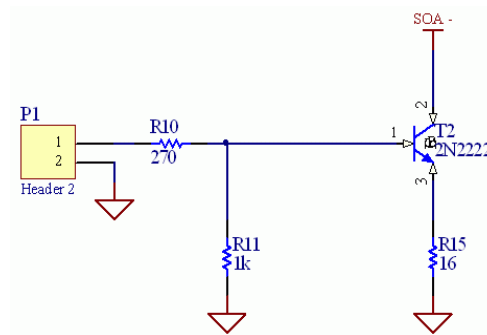


Figura 4.36 - Circuito de polarização do SOA

O sinal de chaveamento, com amplitude de 3,3 Vdc, oriundo da Placa Processadora, é ligado no conector P1, que por sua vez faz o transistor T2, polarizado como fonte de corrente, polarizar o SOA com a corrente especificada pelo resistor R15.

O SOA utilizado foi o IPSAD1502 da INPHENIX [37]. A Tabela 4.4 mostra as especificações do amplificador óptico a semiconductor encapsulado no CPqD para utilização na Placa de Chaveamento Óptico.

Parameter		Symbol	Specifications			Unit
			Min.	Typ.	Max.	
Drive Current		I_F		120	180	mA
Operating Peak Wavelength		λ_p	1510		1570	nm
3 dB Optical Bandwidth		$\Delta\lambda_{3dB}$	50			nm
Small Signal Gain at λ_p @ -25 dBm Signal		G_{max}		10		dB
Gain Ripple with Respect to λ		ΔG		0.2	0.5	dB
Saturation Output Power		P_{sat}	3			dBm
Noise Figure		NF		10		dB
Polarization Dependent Gain		PDG		0.2	0.5	dB
Extinction Ratio		ER	40			dB
Switching Properties	Rise Time	τ_r		500		ps
	Fall Time	τ_f		500		ps

Tabela 4.4 - Especificações elétrico-ópticas do SOA [37]

Maiores informações deste SOA podem ser obtidas no trabalho de caracterização [38] que foi realizado.

4.4.2 Layout da Placa de Chaveamento Óptico

Por ser uma placa que não possui sinais de dados de altas taxas e um número pequeno de componentes, o seu desenho foi fácil de executar e não necessitou de mais de duas camadas para o roteamento.

4.4.2.1 Roteamento da PCI

A Figura 4.37 mostra a estrutura e a altura final da placa de duas camadas que foi desenhada. Ao contrário das demais e não considerando o componente óptico, esta é uma placa de baixíssimo custo e que pode ser fabricada em qualquer fornecedor.

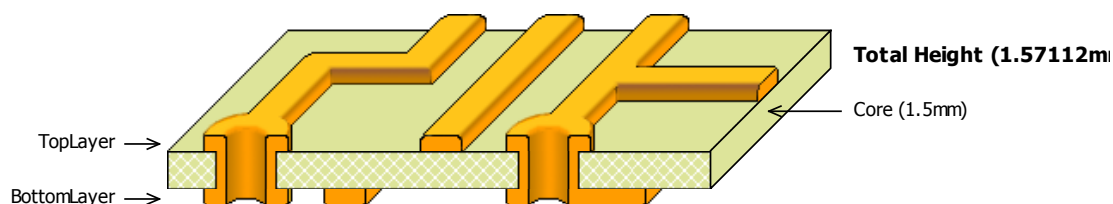


Figura 4.37 - Estrutura da PCI da Placa de Chaveamento Óptico

Na Figura 4.38 observamos a placa roteada e pronta para fabricação. É possível notar algumas trilhas mais espessas. Estas trilhas foram roteadas para suportar a corrente de alimentação do TEC, pois seu consumo pode chegar próximo de 1,5 A.

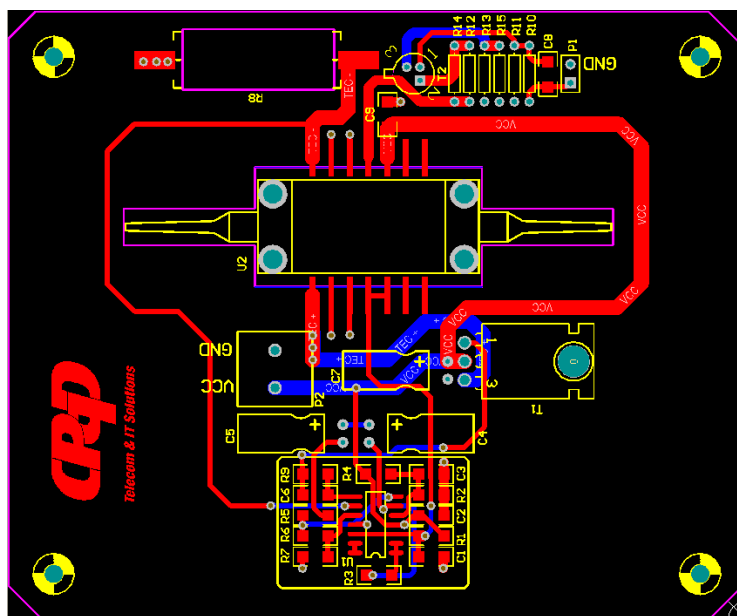


Figura 4.38 - Placa de Chaveamento Óptico roteada

4.4.3 Montagem e testes da placa

Foram fabricadas e montadas 10 placas deste módulo. Todas as placas funcionaram sem necessidade de retrabalhos e seus respectivos SOAs também não apresentaram problema no teste realizado. Como os SOAs foram caracterizados antes da montagem, o único teste que fizemos foi de utilizar o sinal de chaveamento para verificar a atuação do SOA como chave, fazendo ele cortar ou não o sinal óptico inserido. A Figura 4.39 mostra duas placas montadas e prontas para serem testadas.

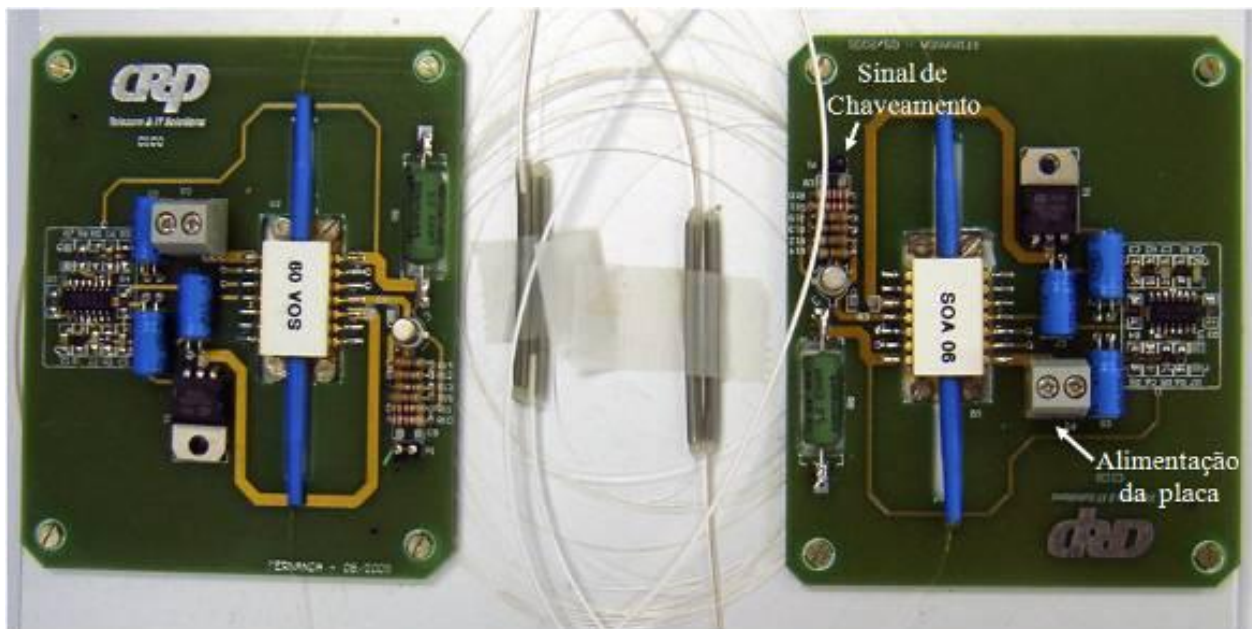


Figura 4.39 - Placa de Chaveamento Óptico montada

Note que como descrito na seção 2.3.2, para formar um bloco de chaveamento completo, são necessárias três placas iguais a esta, por isso foi necessário a montagem de mais de nove placas caso quiséssemos montar três nós de rede completos para testes.

Capítulo 5

NÚCLEO DE PROCESSAMENTO, INTEGRAÇÃO E RESULTADOS

Neste capítulo teremos uma descrição sucinta do funcionamento dos blocos que fazem parte do núcleo de processamento proposto para contemplar as funcionalidades descritas no Capítulo 2, os problemas encontrados na integração das placas e com a lógica desenvolvida no FPGA, e também os resultados alcançados da prototipagem do nó da RCPO.

5.1 Descrição dos blocos do núcleo de processamento

Como já mencionado, a Placa Processadora possui um componente FPGA de alta densidade que permite a implementação de todo um circuito digital composto por máquinas de estado e lógica combinacional. Esse circuito digital é extremamente complexo para ser desenvolvido de uma única vez e, portanto, foi dividido em diversos blocos funcionais para serem desenvolvidos de forma paralela e depois integrados e programados no dispositivo. O projeto desse circuito foi realizado através da linguagem de descrição de hardware denominada VHDL. A Figura 5.1 apresenta um diagrama de blocos que indica o fluxo de dados dentro do núcleo de processamento da Placa Processadora do nó da rede.

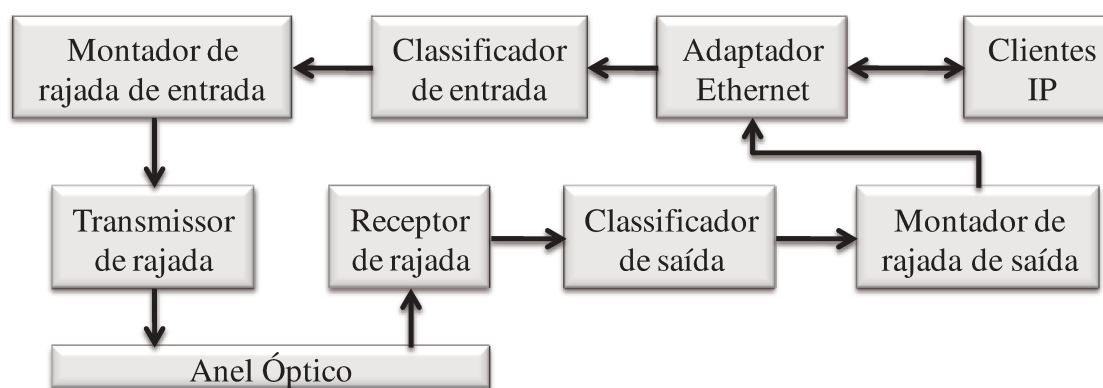


Figura 5.1 - Representação do fluxo de dados no nó de acesso da rede

Seguindo o fluxo de transmissão, do cliente para o anel óptico, temos a inserção dos pacotes Ethernet dos clientes que estão conectados ao bloco Adaptador Ethernet. Neste momento os pacotes de dados são acrescidos de dois Bytes contendo o tamanho do quadro Ethernet. Em seguida, os dados são classificados no Classificador de entrada e armazenados em filas pelo Montador de rajada de entrada, que por sua vez envia os pacotes ao Transmissor de rajada quando um dos critérios de decisão de envio, tamanho de fila ou tempo de retenção do pacote, for atingido. Neste momento ocorre uma troca de sinalização com o plano de controle que informa a necessidade de se iniciar um processo de montagem de rajadas. O Transmissor de rajada executa esta operação e faz a inserção da rajada no anel óptico. Do outro lado, na recepção, do anel óptico para cliente, os dados chegam a Placa Processadora pelo Receptor de rajada que é responsável por desmontar a rajada e encaminhar os pacotes de dados para o Classificador de saída. Após a classificação, os pacotes são armazenados em filas pelo Montador de rajada de saída e então enviados às interfaces de recepção dos clientes novamente através do Adaptador Ethernet.

Nas seções seguintes é feita uma descrição do papel de cada bloco por onde os pacotes de dados circulam.

5.1.1 Bloco Adaptador Ethernet

A principal funcionalidade deste bloco é adicionar e retirar do pacote de dados um campo contendo o tamanho total do quadro, sendo que esta informação não é prevista em algumas versões do padrão Ethernet.

A inclusão deste campo no início de um quadro Ethernet possibilita o desenvolvimento de códigos VHDL mais eficientes, uma vez que este campo permanece ao longo de todo processo de transmissão dos dados de um nó a outro, retirado apenas no momento de envio dos dados ao cliente final.

Um contador de 11 bits é necessário para a inclusão do tamanho do quadro Ethernet que possui no máximo 1518 Bytes. Para facilitar o processo de codificação, dois Bytes são usados deixando os cinco bits mais significativos reservados para uso futuro. O valor armazenado no campo é o número de Bytes total do quadro modificado, ou seja, o tamanho do quadro Ethernet acrescido de dois Bytes.

Por exemplo, um quadro Ethernet com 1000 Bytes irá gerar um quadro modificado com 1002 Bytes. O primeiro Byte será “xxxxx011” e o segundo Byte “11101010”, sendo “x” qualquer valor. Na Figura 5.2 pode-se ver a estrutura do quadro Ethernet modificado pela atuação do Adaptador Ethernet.

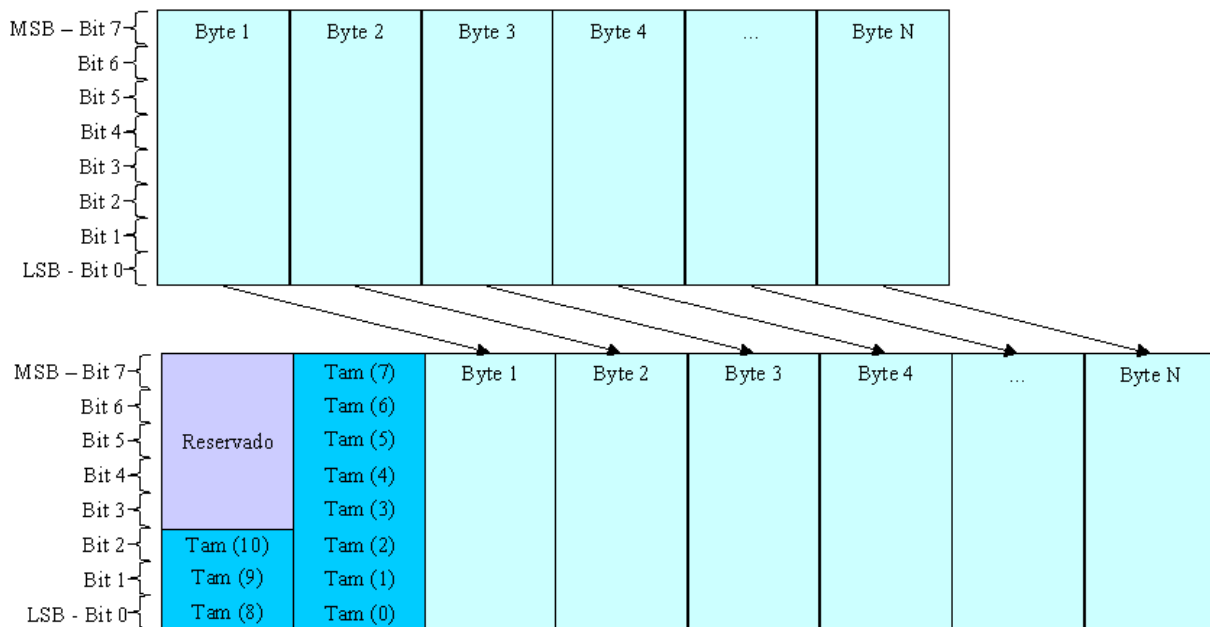


Figura 5.2 - Quadro Ethernet alterado pelo Adaptador Ethernet

O Adaptador Ethernet é responsável pelo recebimento dos quadros Ethernet provenientes da Placa de Acesso aos Clientes e entregá-los, já modificados, ao Classificador de entrada.

Para executar tal funcionalidade, diversos componentes foram desenvolvidos no bloco, como uma interface SPI-3 para se adequar ao controlador MAC da Placa de Acesso aos Clientes. Esta interface tem seus sinais relacionados diretamente aos pinos do FPGA para uma ligação direta com o circuito integrado IXF1104. Esta interface é responsável por todo o controle de transmissão de dados entre as placas.

Em seguida os Bytes do quadro Ethernet são contados e armazenados numa memória do tipo primeiro que entra/primeiro que sai (*First-In-First-Out* - FIFO), sendo este bloco fornecido pela Altera, restando apenas o trabalho de configuração do tamanho da FIFO. Uma segunda FIFO é usada para armazenar o valor desta contagem. Encerrado este processo, o quadro Ethernet é modificado da maneira como foi visto na figura acima com os dois Bytes mais significativos contendo o tamanho total do quadro modificado.

O caminho é feito inversamente para retirar os dois Bytes que contém a informação do tamanho do quadro Ethernet. Os quadros Ethernet são enviados para a Placa de Acesso aos Clientes com o mesmo formato que foram recebidos.

5.1.2 Bloco Classificador de entrada

Neste bloco, os quadros oriundos do Adaptador Ethernet são armazenados também numa memória do tipo FIFO para que os cabeçalho dos quadros Ethernet, que contém o endereço de destino, sejam comparados com os dados contidos na tabela de roteamento fornecida pelo plano de controle (criado no NIOS-II). Depois de realizada a classificação, é iniciada a transmissão para um Montador de rajada de entrada ou para um Montador de rajada de saída, caso este seja um roteamento local. Outra função deste bloco é identificar o endereço do cliente de origem (seu próprio cliente) e armazená-lo num registrador para que o plano de controle possa obter o valor. Cada vez que ocorrer uma troca no endereço do cliente, será gerada uma interrupção para que o plano de controle possa atualizar a sua tabela de endereços dos clientes locais.

5.1.3 Bloco Montador de rajada de entrada e de saída

O Montador de rajada de entrada deve receber os quadros Ethernet devidamente classificados, armazená-los na memória RAM externa (gravando-os no endereço de fila correspondente) e gerenciá-los de acordo com os critérios de montagem de filas (melhor esforço ou circuito virtual).

A Figura 5.3 apresenta um diagrama de blocos que indica o mecanismo de atuação do plano de controle no processo de montagem e gerenciamento das filas.

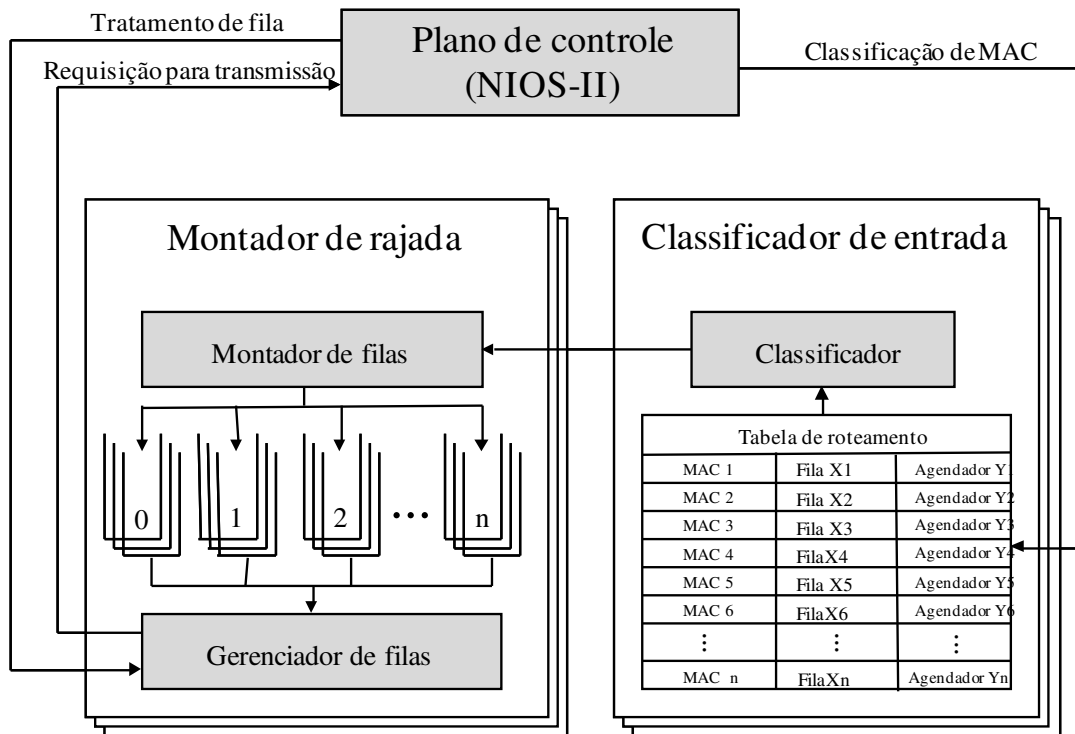


Figura 5.3 - Sistema de gerenciamento de filas

No critério de melhor esforço, a solicitação de transmissão de fila ocorre devido a dois eventos: vencimento do quadro mais antigo ou alcance da capacidade da fila, o que ocorrer primeiro.

Existem quatro operações neste bloco: transferência de dados para a RAM externa, comunicação com o plano de controle (onde são realizadas as solicitações de transmissão e o ajuste do modo de tratamento das filas), entrada dos quadros classificados e encaminhamento das filas para o Transmissor de rajada.

Um conceito importante é que cada Montador de rajada de entrada possui um número determinado de filas, e uma fila não é exclusivamente associada a nenhum nó ou cliente. Cabe ao plano de controle fazer essa associação, montando a tabela de roteamento de cada Montador de rajada de entrada da melhor forma possível. Isso resulta em maior flexibilidade, permitindo ao plano de controle realizar uma melhor partição de cargas. É importante notar que cada fila no Montador de rajada de entrada possui replicações, pois após uma fila ser fechada existe um determinado tempo entre a solicitação de transmissão e o seu efetivo atendimento, ao passo que os quadros continuam chegando durante esse período. Assim, as diversas réplicas de cada fila são ciclicamente utilizadas

pelos Montadores de filas para evitar perda de quadros.

O Montador de rajada de saída funciona de forma semelhante ao de entrada, mas tendo como “cliente” uma rede IP e não um ambiente totalmente proprietário. Sendo assim, este bloco irá também armazenar os quadros Ethernet em filas, cada qual com um endereço de destino, mas sem obedecer aos requisitos de temporização para envio dos pacotes ao Adaptador Ethernet.

5.1.4 Bloco Transmissor de rajada

O Transmissor de rajada é o responsável por montar efetivamente a rajada e transmiti-la no anel óptico.

É importante que, assim como no Montador de rajada de entrada, este bloco não trate nenhuma característica do protocolo Ethernet. Desse modo, o Montador de rajada de entrada envia um mapeamento da rajada que tornam genéricas as informações a serem transmitidas. Através deste mapeamento, o Transmissor de rajada pode segmentar a rajada corretamente e realizar a transmissão na medida em que os dados chegam à memória.

Deve-se observar, ainda, que não é necessária a transmissão total da fila para a montagem da rajada, pois isso demanda uma quantidade muito grande de memória RAM. A transmissão da rajada pode-se iniciar antes de sua completa montagem. Dado também que a comunicação ocorre apenas durante a transmissão dos dados, o Transmissor de rajada acrescenta o preâmbulo para garantir o sincronismo da interface de recepção. Da mesma forma, é necessário garantir que não ocorram seqüências muito longas de bits ‘1s’ ou ‘0s’, o que é conseguido através do uso da codificação de linha 8b/10b, também realizado pelo bloco.

Como já mencionado, para o correto funcionamento, os segmentos não devem quebrar os quadros Ethernet, ou seja, cada segmento deve conter apenas quadros completos. O segmento também deve ser de tamanho fixo e, para isso, é necessária a utilização de Bytes de preenchimento, outra função deste bloco.

Por último, o Transmissor de rajada insere o campo CRC para que o segmento possa ter sua integridade verificada no Receptor de rajada.

5.1.5 Bloco Receptor de rajada

Este bloco tem a função de receber as rajadas provenientes do anel óptico e encaminhá-las ao Classificador de saída.

A primeira tarefa deste bloco é decodificar a rajada que foi previamente codificada pelo bloco Transmissor de rajada utilizando a técnica de codificação e decodificação 8b/10b. Depois disso, de forma contínua os segmentos da rajada são analisados utilizando o verificador de erro CRC. Por isso que a rajada é segmentada e cada segmento contém o seu próprio CRC, no caso de erro de transmissão, apenas os segmentos que sofreram erro serão descartados, ao contrário do tratamento tradicional onde toda a rajada seria desprezada. Essa técnica foi desenvolvida para aumentar a eficiência do sistema. Os campos de preenchimento e CRC também são removidos nesta etapa.

5.1.6 Bloco Classificador de saída

O bloco Classificador de saída tem a função de classificar os quadros Ethernet enviados pelo Receptor de rajada e encaminhá-los para o Montador de rajada de saída. Os quadros recebidos são armazenados numa memória FIFO e em seguida os campos de endereçamento de destino dos quadros Ethernet são analisados e comparados com os endereços de uma tabela de roteamento também preenchida pelo plano de controle. Então estes são mapeados para um endereço físico para onde o quadro deverá ser enviado no bloco Montador de rajada de saída.

Além dos blocos citados, foi observada a existência de outros blocos funcionais, como o Controlador de DDRII, que é responsável pelo controle dos sinais elétricos da interface com o banco de memória RAM da Placa Processadora, o Controlador das chaves ópticas que envia os sinais para a Placa de Chaveamento Óptico reter ou não a rajada, a Interface entre planos de gerência e controle que deve prover a comunicação entre as unidades de plano de controle (dentro do nó e entre nós) e finalmente os processadores NIOS-II do tipo *softcore*, desenvolvidos pelo fabricante do dispositivo lógico programável. Estes processadores são dedicados ao processamento de *slots*, execução do controle das filas e gerenciamento da transmissão das rajadas.

5.2 Integração

Com o teste individual das placas concluído e os blocos principais também, o próximo passo foi integrar todo o trabalho para verificar o funcionamento do conjunto. Digo blocos principais aqueles necessários para validar o *hardware* do nó da rede e que estavam totalmente testados, que são: Adaptador Ethernet, Classificador de entrada e de saída, transmissor e receptor de rajada.

Vale esclarecer que cada bloco desenvolvido também foi testado individualmente utilizando as ferramentas de verificação do *Quartus II* da Altera. As ferramentas utilizadas são o *Vector Waveform* e o *SignalTap*. A primeira, como o próprio nome indica, é um gerador e visualizador de forma de onda digital que é utilizado para simular o comportamento do circuito desenvolvido sem o porte para o FPGA. Já o *SignalTap* é um analisador lógico que é implementado internamente ao FPGA pela ferramenta *Quartus II*. Com o circuito desenvolvido portado no FPGA, basta especificar os sinais desejados para visualização que a ferramenta implementa automaticamente o analisador para verificação dos dados em tempo real. A vantagem deste último é que são levados em consideração os fatores que podem prejudicar o desempenho do circuito, como por exemplo a temporização dos sinais e o roteamento interno ao FPGA. Com a ajuda das ferramentas, por exemplo, foi possível verificar se o resultado analisado na saída de um codificador 8b/10b estava obedecendo o padrão proposto tanto quando ainda em desenvolvimento quanto já no FPGA.

5.2.1 União dos blocos

Como já mencionado, os blocos são interligados através do barramento *Avalon* dentro do FPGA, controlados e monitorados pelo *softcore* NIOS II. Para isto, os blocos desenvolvidos se tornaram componentes de uma biblioteca, tendo os mesmos recursos dos módulos que já incorporam a ferramenta, como PLLs e memórias do tipo FIFO. Assim, o barramento *Avalon* é capaz de detectar os blocos e interligá-los de acordo com as regras estabelecidas de prioridade e acesso. Vale esclarecer que ao desenvolver um bloco, é necessário prever pinos de endereçamento para acesso ao barramento. Para este teste, o processador NIOS-II utilizado não executa as funções estabelecidas para os

blocos Processador de cabeçalho e Processador de controle, pois não serão utilizados todos os blocos previstos na arquitetura completa do nó. A função do processador é apenas para uso do barramento *Avalon* que é atrelado ao NIOS-II. A Figura 5.4 mostra a arquitetura criada para os testes.

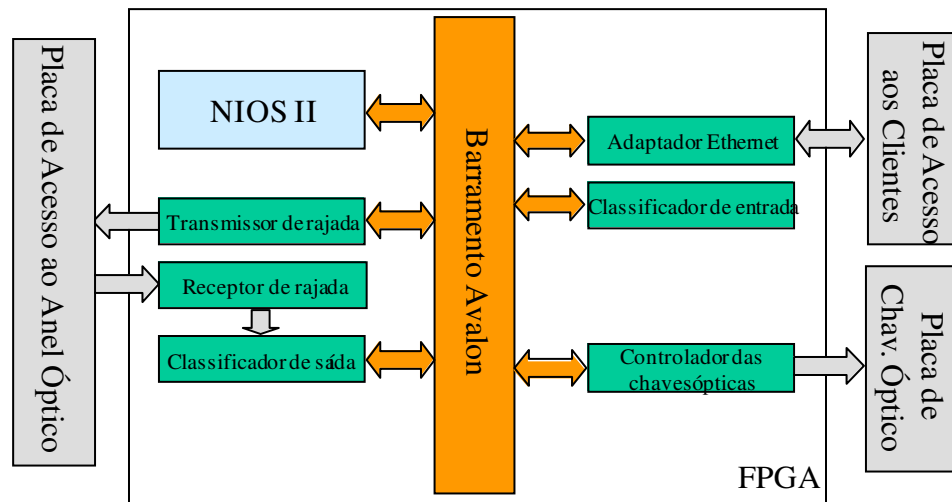


Figura 5.4 - Arquitetura da lógica de processamento criada para o teste do nó de acesso

Note que a figura mostra apenas uma interface para o acesso ao anel óptico e clientes, mas até 4 interfaces podem ser criadas, o que foi feito para testar o *hardware* por completo. Por não possuir os blocos, este teste não contempla montagem da rajada de pacotes, mas para teste do protótipo não houve necessidade de esperar o desenvolvimento destes, portanto, os pacotes que entram no Adaptador Ethernet são enviados diretamente pelo Transmissor de rajada e vice-versa.

Uma vez criado o projeto para unir os blocos num único circuito, as etapas seguintes são: compilar o projeto para verificação de erros, assinalar os pinos desejados para uso e sintetizar o projeto, que é criar um arquivo para configurar os elementos lógicos e outros módulos do FPGA de acordo com o código desenvolvido em VHDL.

Como o FPGA é um componente volátil, ele requer uma memória externa onde o arquivo de configuração fica armazenado e toda vez que o dispositivo é ligado, é iniciado o processo de carregamento automaticamente através de uma máquina de estado interna.

Componentes internos de dispositivos FPGA como PLLs, blocos de memória e conexões customizadas para sinais de alta frequência são fixos. Com isso, se o circuito

desenvolvido utilizar alguns destes componentes, como foi o nosso caso, estes necessitam ser modelados para que fiquem próximos aos módulos fixos para atingir a performance desejada. Não é incomum um circuito não atingir a performance esperada devido a um mau arranjo da lógica interna. Por exemplo, se dois circuitos que compartilham os mesmos sinais ficarem afastados um do outro, os sinais, principalmente de alta frequência, terão que percorrer certa distância que poderá afetar no alcance da frequência exigida.

Ao efetuar a síntese de um determinado projeto, a ferramenta faz uma análise de acordo com o arranjo do circuito e informa se, por exemplo, um sinal que previamente identificado como de relógio irá operar na frequência desejada. Caso o resultado da síntese automática não for adequado, o desenvolvedor pode informar a ferramenta para posicionar os circuitos (blocos) mais críticos de forma que fiquem mais bem arranjados. Este problema se agrava de acordo com o número de blocos existentes no FPGA. Quanto mais blocos, menos espaço existirá para realizar o melhor arranjo e com isso o desempenho do projeto como um todo diminui proporcionalmente ao número de lógica utilizado.

Mesmo com um número de blocos reduzido, encontramos dificuldade para atingir a performance requerida no projeto utilizado para o teste. Várias tentativas foram necessárias para atingir os requisitos mínimos do projeto por ter de trabalhar com barramento de 1 Gbit/s e sinais de relógio de 125 MHz. Como resultado prévio, concluímos que seria muito difícil atingir os requisitos do projeto com todos os blocos previstos no FPGA utilizado.

5.2.2 Teste do nó da rede

A Placa de Chaveamento Óptico não foi incluída nesta fase, pois foi possível aprovar seu funcionamento já no teste individual e na caracterização. Embora as demais placas tenham tido testes individuais, este teste foi mais abrangente e inclui também o comportamento da lógica de processamento contendo os blocos citados.

O primeiro passo foi montar um nó e testá-lo individualmente para validar o conjunto e em seguida com duas ou mais unidades para avaliar o comportamento deles na situação de prova de conceito. A Figura 5.5 mostra o diagrama do teste individual do nó. Com um

computador gerando e recebendo pacotes Ethernet nas interfaces (uma por vez) da Placa de Acesso aos Clientes, foram feitos *loops* de fibra óptica nos módulos ópticos da Placa de Acesso ao Anel para que os dados pudessem ser retornados.

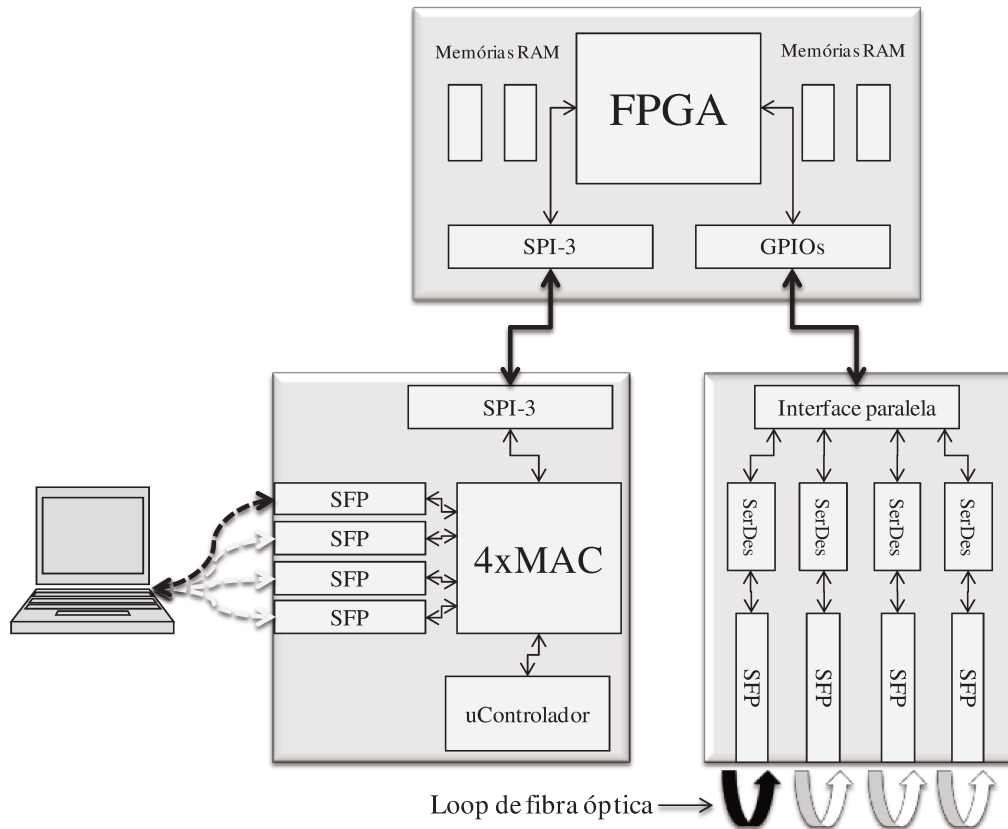


Figura 5.5 - Diagrama do teste do nó da rede

Foram utilizadas as mesmas ferramentas para geração e análise dos pacotes dos testes anteriores e também o *SignalTAP* para visualizar os dados dentro do FPGA. O *SignalTAP* foi muito útil para depurar a lógica desenvolvida no momento do teste. Como exemplo, as figuras seguintes mostram a leitura feita pelo analisador lógico nos sinais de entrada e saída do bloco Adaptador Ethernet.

Na Figura 5.6 pode-se observar os pinos que indicam início e fim do pacote (RSOP e REOP respectivamente) da interface SPI-3 sinalizando para o bloco Adaptador Ethernet que aquela sequência de dados é um pacote válido.

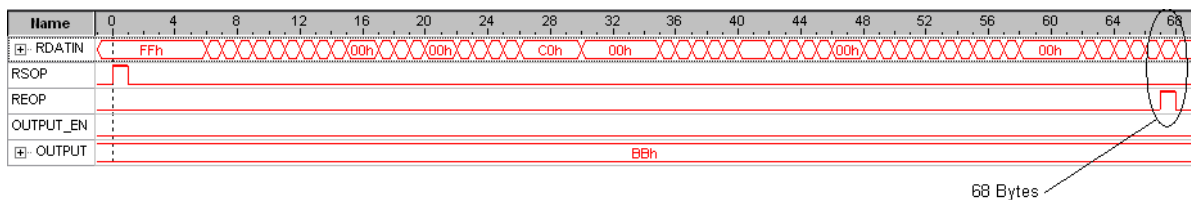


Figura 5.6 - Pacote Ethernet com tamanho de 68 Bytes na entrada do bloco

Ao receber a indicação de início de pacote, o bloco começa a armazenar os dados numa memória para contagem do número de Bytes. O resultado pode ser observado na Figura 5.7. O sinal OUTPUT_EN do módulo indica que os dados saindo do bloco Adaptador Ethernet pelo barramento OUTPUT é um pacote válido. Note que o pacote saindo contém 70 Bytes devido aos 2 Bytes indicando o tamanho.

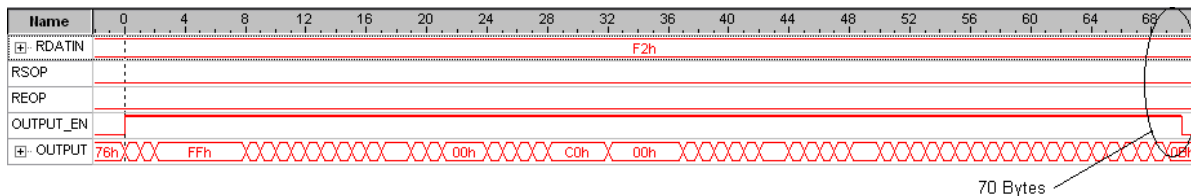


Figura 5.7 - Quadro saindo do bloco com 70 Bytes

A Figura 5.8 mostra em detalhe os Bytes indicando o tamanho total do quadro. Note que como foram acrescentados dois Bytes, o valor final do pacote ficou com 70 Bytes.

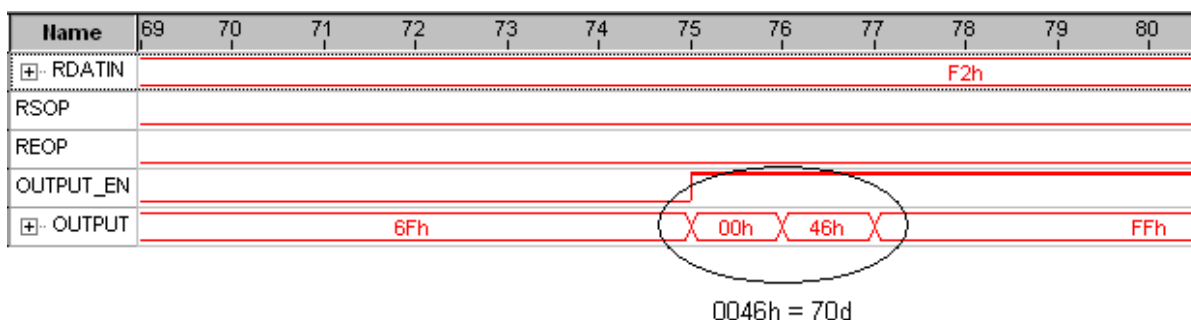


Figura 5.8 - Detalhe dos 2 Bytes no início que indicam o tamanho do pacote

Por fim, é possível verificar o quadro na entrada (barramento RDATIN) e saída do bloco Adaptador Ethernet na Figura 5.9.

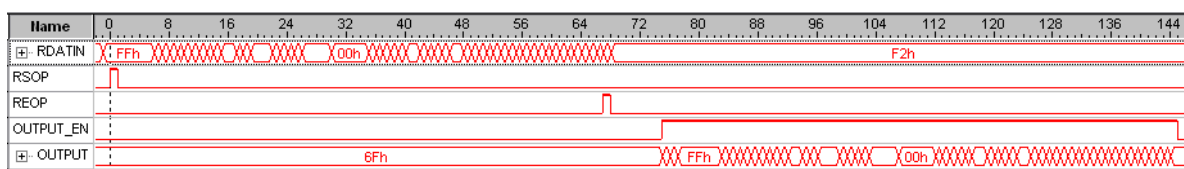


Figura 5.9 - Pacote de dados na entrada e saída do bloco Adaptador Ethernet

Com essa ferramenta foi possível seguir os sinais para depurar e fazer os ajustes finos de cada bloco da lógica de processamento implementada para os testes.

Mesmo simulando os pacotes anteriormente, algumas correções foram necessárias para que a lógica do nó pudesse funcionar adequadamente. Uma vez o caminho de ida e volta funcionando, foram feitos testes com vários tamanhos de pacotes, várias taxas e também nas quatro interfaces com todos os resultados sendo positivos, sem erro e perda de dados. A Figura 5.10 mostra as placas conectadas formando o núcleo do nó de acesso da RCPO.



Figura 5.10 - Placas conectadas para o teste do nó de acesso da rede

O passo seguinte foi montar outro nó e observar o comportamento do conjunto. Imaginávamos que não teríamos surpresas com este teste, pois se utilizando um *loop* de

fibra simulando outros nós havia funcionado, adicionando outro nó o resultado devia ser o mesmo, mas não foi logo na primeira tentativa.

A Figura 5.11 mostra o diagrama do teste com mais de um nó. Agora, não é mais o mesmo computador que transmite e recebe os pacotes de dados como no teste anterior, mas sim um computador em cada “extremidade”.

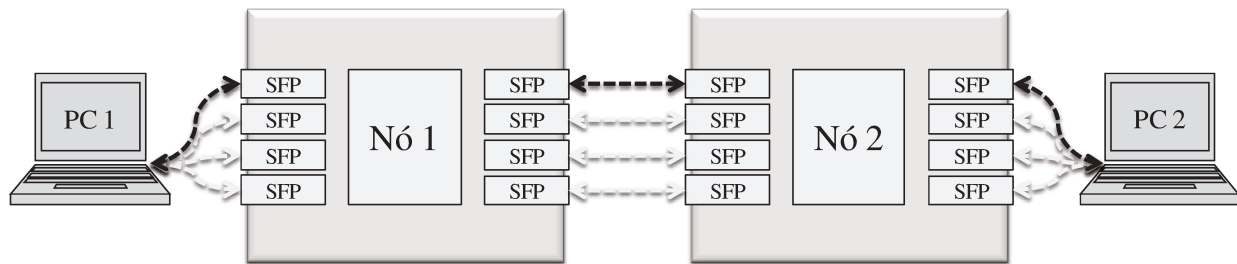


Figura 5.11 - Diagrama do teste inicial com dois nós de acesso

Como já dito, o resultado esperado neste teste era receber os pacotes enviados pelo PC 1 no PC 2 e vice-versa. Mas por um motivo até então desconhecido, os pacotes enviados pelo Nó 1 não eram reconhecidos pelo outro nó, quanto menos pelo cliente no outro lado (PC 2), e quando recebidos eram observados muitos erros (perda de dados).

Vários testes foram realizados na tentativa de descobrir onde estava o problema, pois quando se fazia um *loop*, como no teste anterior, ambos os nós funcionavam perfeitamente. Para descobrir se o problema era da lógica desenvolvida ou das placas, desenvolvemos um projeto que substituiu a lógica por um simples curto circuito, e mesmo assim não conseguíamos ver os pacotes pelo *SignalTAP* no nó destino. O problema também foi observado em ambos os sentidos de transmissão e para excluir definitivamente o meio físico (fibras e módulos ópticos), usamos o recurso de teste dos SerDes da Placa de Acesso ao Anel, que gera uma sequência de dados conhecida. Com isso foi possível observar que a sequência gerada pelos SerDes do Nó 1 era reconhecida sem erros pelo outro nó e vice-versa, o que comprovou que o problema não estava no meio físico.

O problema começou a ser resolvido quando o teste foi realizado sem os ventiladores que usávamos para resfriar as placas. Percebemos que com as placas mais quentes os pacotes eram recebidos pelo outro nó com taxa de erro nula, fenômeno que não ocorria no teste individual das placas e do nó. Nos testes individuais sempre utilizávamos os

ventiladores e nenhum erro ocorreu.

Com a causa detectada, começamos a procurar qual era o circuito com o problema e em que placa estava. Descobrimos que o problema era o circuito de alimentação da Placa Processadora, que quando resfriado injetava mais ruído que o aceitável para os demais circuitos da placa e como consequência dificultava o reconhecimento dos pacotes de dados no outro nó. Era visível a diferença com e sem ventilador.

O circuito de alimentação com mau comportamento da Placa Processadora é composto de um regulador linear de 3,3 Vdc para 1,5 Vdc e de dois capacitores na entrada e saída do componente para ajudar na regulação da tensão de saída. Chegamos a trocar o regulador por outro fabricante e o problema persistia, então trocamos os capacitores e a sensibilidade com relação à variação de temperatura melhorou muito, diminuindo o ruído na saída do regulador mesmo quando frio. A Figura 5.12 mostra o circuito com problema de temperatura da placa.

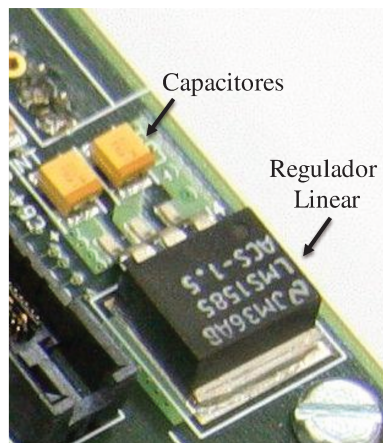


Figura 5.12 - Circuito de alimentação com problema da Placa Processadora

Como não tínhamos o código do capacitor utilizado, não tínhamos como saber a especificação dele, como baixa resistência em série equivalente, parâmetro requisitado quando utilizado com reguladores e conversores chaveados por afetar a estabilidade ou pequena variação da capacitância em relação à temperatura ambiente. O capacitor que utilizamos possuía as características acima, mas não é possível dizer que era este o problema dos componentes originais, pois poderia ser um problema do lote fabricado pelo fato de todas as placas acusarem o mesmo problema.

Mesmo com este novo capacitor, ainda se observa erros na transferência de pacotes

quando o circuito de alimentação é muito resfriado. Mas como este problema não iria atrapalhar o desenvolvimento do restante do projeto, decidimos não despendar mais esforços para o ocorrido por se tratar de uma primeira versão de um protótipo.

Com o problema descoberto, pudemos testar o fluxo de pacotes entre os nós em todas as interfaces da mesma maneira do teste individual, com variação da taxa e tamanho dos pacotes. O resultado dos testes foi positivo, sem erro ou perda de informação, comprovando o funcionamento do *hardware* com dois ou mais nós interligados, como ilustrado na Figura 5.13.



Figura 5.13 - Dois nós de acesso da rede sendo testados com o uso de ventiladores

Para explicar o funcionamento do nó operando individualmente e não com outro quando resfriado, podemos dizer que o circuito de recepção do FPGA está operando no limite de ruído suportável e qualquer injeção de ruído, pelo mínimo que seja, se agrava quando os dados trafegam até o outro nó, alterando seu funcionamento e fazendo o receptor não mais reconhecer os níveis dos sinais corretamente, acarretando num número elevado de erros.

5.3 Resultados

Com os nós da rede operando e estáveis, tanto o *hardware* quanto a lógica de processamento, o próximo passo foi fazer um teste mais elaborado, incluindo a Placa de Chaveamento Óptico, carretéis de fibra óptica para simular um *link* e com tráfego real simultâneo.

Para o teste utilizamos dois carretéis de fibra *single mode* e módulos ópticos operando na janela de 1550 nm com boa sensibilidade, possibilitando um alcance de 20 km. Para este teste, como todos os pacotes são destinados ao outro anel, a Placa de Chaveamento Óptico não está sendo chaveada, ou seja, está sempre habilitada a transferir os dados.

Para teste de tráfego simultâneo, cada computador transmitia um vídeo através das interfaces ópticas de 1 Gbit/s para o outro, possibilitando deste modo testar as interfaces de transmissão e recepção de cada nó ao mesmo tempo. A Figura 5.14 mostra a configuração do teste.

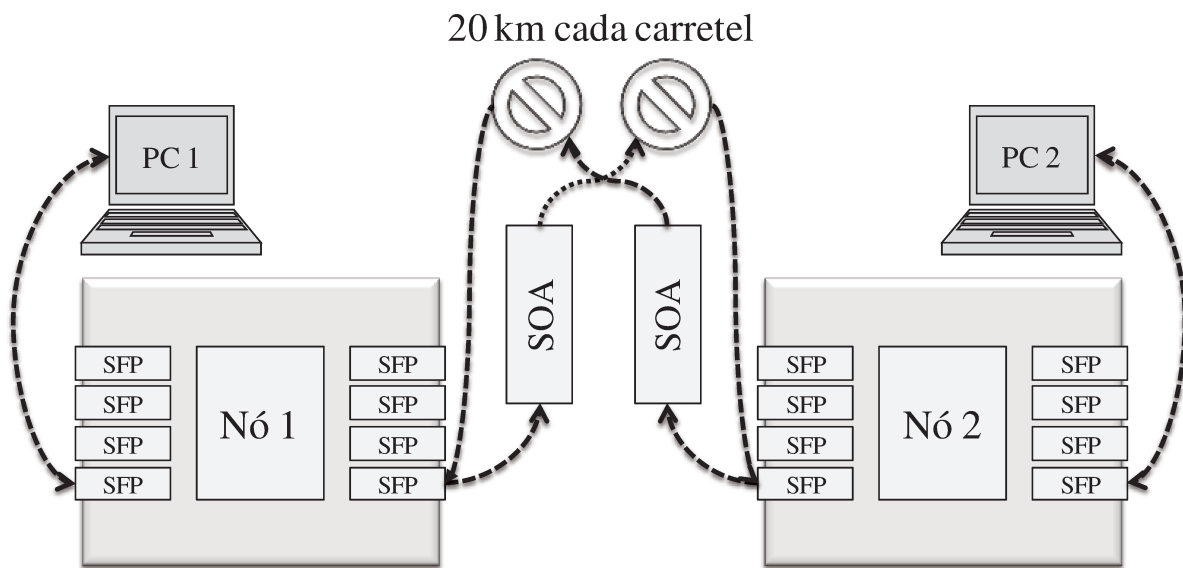


Figura 5.14 - Diagrama do teste com a Placa de Chaveamento Óptico, link de 20 km e transmissão de vídeo

Com esse teste foi possível comprovar o funcionamento simultâneo das interfaces de transmissão e recepção tanto do hardware quanto dos blocos desenvolvidos no FPGA. Vale lembrar que mesmo sem os blocos montadores de rajada, podemos dizer que uma rajada era montada pelos demais blocos a cada pacote Ethernet transferido.

Apenas para ilustração, neste teste foi possível notar a degradação da imagem do vídeo transmitido em relação à perda de dados quando o circuito de alimentação era resfriado. Mas após a substituição dos capacitores, como já mencionado, não houve mais problemas com perda de pacotes quando operando na temperatura ambiente. Os equipamentos utilizados, além dos computadores, foram fontes de alimentação *Suplitech PSL 2002* e monitores de potência óptica com atenuador variável modelo 420 da *EigenLight*. Na Figura 5.15 e Figura 5.16 é possível observar detalhes dos testes realizados no laboratório do CPqD.



Figura 5.15 - Teste com dois nós de acesso no teste de transferência de vídeo simultânea



Figura 5.16 - Detalhe da montagem de dois nós de acesso mais os carretéis de fibra óptica utilizado para o teste de tráfego simultâneo

Com estes testes podemos dizer que os resultados esperados no início do projeto, com relação à construção de um protótipo de *hardware* para prova de conceito da rede, foram alcançados respeitando os requisitos propostos. Em relação à lógica de processamento, pode-se concluir que os blocos desenvolvidos até o momento foram suficientes para validar as placas e as funcionalidades dos que foram criados.

Capítulo 6

CONSIDERAÇÕES FINAIS

Através da montagem do protótipo, foi possível por em prática, além da realização de todo o *hardware* previsto, uma parte da inteligência do nó desenvolvida até o momento dos testes. Com o protótipo operacional será possível testar os demais blocos lógicos que compõem a lógica de processamento, assim que desenvolvidos, e validar a arquitetura proposta da rede.

Neste projeto várias ferramentas, tecnologias e práticas ainda não exploradas foram utilizadas, fazendo com que a equipe desenvolvesse a capacitação técnica. Podemos dizer que este projeto fez a equipe incorporar conhecimento técnico na área de desenvolvimento de placas de circuito impresso com elementos de alta frequência, técnicas para utilização de lógica programável (FPGA), trabalhos com SOAs, etc. Esta capacitação técnica foi adquirida com o desenvolvimento de quatro placas, as quais compõem um nó de acesso da rede, testadas e funcionando como esperado. Também foram desenvolvidos vários blocos em linguagem VHDL para fazer o processamento dos pacotes de dados, operando na taxa de Gbit/s, além de toda arquitetura que foi várias vezes modificadas para alcançar o estágio final que foi apresentado no Capítulo 2.

Também podemos citar alguns pontos que podem ser modificados e melhorados no projeto. Um deles é inserir conversores lineares ou chaveados, dependendo da potência necessária, em todas as placas para serem alimentadas apenas com um único valor de tensão, como 12 Vdc por exemplo, para se evitar o uso de várias fontes de alimentação externas. Outro fator relevante é a capacidade do FPGA, pois observando o uso dos blocos desenvolvidos, pode-se notar que teremos menos problemas para sintetizar todo o projeto proposto da lógica de processamento num componente com maior número de

elementos do qual está sendo utilizado. Ao invés de utilizar um dispositivo maior, também se pode fazer o uso de dois FPGAs, dividindo as funcionalidades entre os dispositivos. Para se ter uma idéia, os blocos utilizados nos testes consumiram 56% do recurso do FPGA de 30.000 elementos lógicos utilizados. Claro que com uma maior experiência no desenvolvimento com este tipo de tecnologia podem-se alcançar números menores com a otimização dos blocos, pois o consumo de elementos lógicos está condicionado a maneira com que o circuito é desenvolvido. Por este motivo é interessante sempre utilizar, quando possível, os circuitos disponibilizados pelo fabricante, pois já estão otimizados. Estes dois são os pontos que se pode melhorar de imediato, pois no restante não observamos problemas sérios. Outro que poderíamos citar é o do regulador linear da Placa Processadora, mas com o uso de outra placa desenvolvida pelo CPqD este problema será sanado.

Com relação ao gasto com material utilizado neste protótipo, cada placa teve um custo diferente, tanto em relação à lista de materiais quanto ao valor da placa de circuito impresso, pois quanto mais camadas a placa possuir, mais caro é para produzi-la. Falando em números, o custo da Placa Processadora foi de 4.500 dólares cada, seguido pela Placa de Acesso aos Clientes com 2.000 dólares, depois a Placa de Acesso ao Anel Óptico com 700 dólares e por fim a Placa de Chaveamento Óptico com 2200 dólares, já incluído o custo médio de um SOA. Estes valores são aproximados e incluem a PCI e os componentes para cada placa, menos a Placa Processadora que consta o preço de mercado, pois ela foi adquirida da empresa *PLD Applications*.

6.1 Trabalhos futuros

Sugere-se como trabalho futuro o desenvolvimento dos demais blocos que compõem a lógica de processamento para que seja possível a realização da montagem dos pacotes ópticos com todos os critérios propostos.

Também é sugerido o desenvolvimento de uma nova Placa Processadora sanando os problemas encontrados na atual e contemplando um FPGA com maior capacidade de elementos lógicos.

Por fim, como as arquiteturas de redes de chaveamento de pacotes ópticos são temas sempre abordados para o futuro das redes ópticas, pela alta capacidade de transmissão de dados e flexibilidade, sugere-se explorar o assunto na próxima fase do projeto GIGA, que terá continuação no CPqD.

Referências bibliográficas

- [1] Y. Chen, C. Qiao, X. Yu, “*Optical Burst Switching: A New Area in Optical Networking Research*”, IEEE Network, vol. 18, nº. 3, pp. 16-23, May/June 2004.
- [2] C. Qiao, M. Yoo, “*Choices, features, and issues in Optical Burst Switching (OBS)*”, Optical Network Magazine, pp. 36-44, April 2000.
- [3] Jason P. Jue, Vinod M. Vokkarane “*Optical Burst Switched Networks*”, Optical Networks Series, 2005.
- [4] OPTICAL NETWORKS MAGAZINE, USA: SPIE, vol. 3, nº. 6, pp. 96, Nov./Dec. 2002.
- [5] F. R. Barbosa, D. Maia, L. Pezzolo, A. C. Sachs, M. R. Salvador, “*Optical Packet Switching Node for Metro-Access Networks*”, European Conference on Optical Communications, ECOC’2003, Rimini, Italy, Sep. 2003.
- [6] OPTICAL NETWORKS MAGAZINE, USA: SPIE, vol. 3, nº. 1, pp. 92, Jan./Feb. 2002.
- [7] M. R. Salvador, E. Mobilon, V. G. de Oliveira, R. Bernardo, et al., “*Arquitetura e Protótipo de Rede de Pacotes em Anel com Transmissão e Comutação no Domínio Óptico*”, 24º Simpósio Brasileiro de Redes de Computadores, Curitiba, Brasil, Maio/Junho 2006.
- [8] E. Mobilon, M. R. Salvador, R. Bernardo, F. Damiani, et al., “*Hardware Design and Prototype of an Optical Packet-Switched Ring Network Node*”, 49th IEEE International Midwest Symposium on Circuits and Systems, San Juan, Puerto Rico, Aug. 2006.

- [9] E. Mobilon, M. R. Salvador, R. Bernardo, F. Damiani, et al., “*Hardware Architecture for Optical Packet and Burst Switching Applications*”, 6th International Telecommunications Symposium, Fortaleza, Brazil, Sep. 2006.
- [10] M. R. Salvador, E. Mobilon, R. Bernardo, et al., “*Protocolo de controle de acesso do anel de rede de comutação óptica de pacotes*”. Cadernos CPQD tecnologia, Brasil, v. 2, p. 10-18, 2006.
- [11] M. R. Salvador, E. Mobilon, F. R. Barbosa, V. G. de Oliveira, R. Bernardo, et al., “*Rede Óptica de Pacotes de Serviços Avançados*”. Cadernos CPQD tecnologia, Brasil, v. 1, n. 1, p. 95-108, 2005.
- [12] J. F. Mollenauer, “*Standards for metropolitan area networks*”, IEEE Communications Magazine, vol. 26, nº. 4, pp. 15 - 19, Apr. 1988.
- [13] R. Roka, “*The utilization of the DWDM/CWDM combination in the metro/access networks*”, Mobile Future and Symposium on Trends in Communications, SympoTIC '03, pp. 160-162, Oct. 2003.
- [14] J. Cai, A. Fumagalli, I. Chlamtac, “*The Multitoken Interarrival Time (MTIT) Access Protocol for Supporting Variable Size Packets Over WDM Ring Network*”, IEEE Journal on Selected Areas of Communications, vol. 18, nº. 10, pp. 2094-2103, Oct. 2000.
- [15] M. R. Salvador, “*MAC Protocols for Optical Packet-Switched WDM Rings*”, Twenty University Press, 2003.
- [16] I. Cidon, Y. Ofek, “*MetaRing - A Full-Duplex Ring with Fairness and Spatial Reuse*”, IEEE Transactions on Communications, vol. 41, nº. 1, pp. 110-120, 1993.
- [17] IEEE 802.3, “*IEEE standards for local area networks: supplements to carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications*”.

- [18] A. S. Tanenbaum, “*Sistemas Operacionais Modernos*”, 2^a ed., São Paulo, Pearson Prentice Hall, 2003.
- [19] Altera, “*Stratix FPGA Family*”, datasheet, disponível em http://www.altera.com/literature/ds/ds_sgx.pdf.
- [20] Ben Cohen, “*Real Chip Design and Verification: Using Verilog and VHDL*”, VhdlCohen Publishing, 2002.
- [21] Intel, “*IXP Network Processor*”, product brief, disponível em <http://www.intel.com/design/network/prodbrf/304022.htm>.
- [22] Broadcom, “*BCM56504*” product brief, disponível em <http://www.broadcom.com/collateral/pb/56504-PB04-R.pdf>.
- [23] R. Ramaswami, K. Sivarajan, “*Optical Networks: A Practical Perspective*”, 2nd ed., MK Publishers, 2001.
- [24] Altera, “*NIOS II Processor Reference*”, handbook, disponível em http://www.altera.com/literature/hb/nios2/n2cpu_nii5v1.pdf.
- [25] M. Guimarães, “*Circuito Equivalente e Extração de Parâmetros num Amplificador Óptico Semicondutor*”, Dissertação de Mestrado, FEEC/UNICAMP, Junho 2007.
- [26] Cortina Systems, “*IXF1104 4-Port Gigabit Ethernet Media Access Controller*”, datasheet, disponível em <http://www.cortina-systems.com/products/view/18>.
- [27] Opnext, “*SFP Transceiver Series*” datasheet, disponível em http://www.opnext.com/products/pdf/SFP_Overview-61704.pdf.
- [28] Texas Instrument, “*TLK2201A Ethernet Transceivers*”, datasheet, disponível em <http://focus.ti.com/lit/ds/symlink/tlk2201a.pdf>.

- [29] F. R. Barbosa, C. Coral, J. R. Caumo, A. Flacker, “*Hermetically packaged semiconductor optical amplifier for application in singlemode fiber systems*”, Proc. SBMO 1993, pp.129, 1993.
- [30] PLD Applications, “*PCI-X SYS User’s Guide*”, disponível em <http://www.plda.com/index.php>.
- [31] OnTAP, “*Series 4000 with PROSCAN*”, product brief, disponível em <http://www.flynn.com/products/ontap-development/>.
- [32] Altium, “*Altium Designer 6.9*”, release notes, disponível em <https://wiki.altium.com/display/ADOH/Whats+New+in+Altium+Designer+6.9>.
- [33] Atmel, “*8-bit MCU with 32K byte Flash and 10 bit A/D*”, datasheet, disponível em www.atmel.com/dyn/resources/prod_documents/doc3b01584676cd4.pdf.
- [34] H. Johnson, M. Graham, “*High Speed Signal Propagation: Advanced Black Magic*”, Prentice Hall PTR, 2006.
- [35] IPC-4552, “*Specification for Electroless Nickel/Immersion Gold (ENIG) Plating for Printed Circuit Boards*”.
- [36] Lattice, “*8b/10b Encoder/Decoder*”, reference design RD1012, 2002.
- [37] INPHENIX, “*Semiconductor Optical Amplifier Device (Switch-type)*”, datasheet, disponível em http://www.inphenix.com/pdfdoc/SOA_Device_Switch-type.pdf.
- [38] F. Delporte, A. D. Coral, F. R. Barbosa, “*Amplification and Noise in Semiconductor Optical Amplifiers*”, MOMAG, São Paulo, Agosto 2004.

Índice de figuras

<i>Figura 2.1- Macro arquitetura da RCPO.....</i>	<i>7</i>
<i>Figura 2.2 - Relacionamento entre os planos funcionais</i>	<i>9</i>
<i>Figura 2.3 - Classe de serviço de melhor esforço</i>	<i>10</i>
<i>Figura 2.4 - Estrutura do cabeçalho</i>	<i>13</i>
<i>Figura 2.5 - Formato do pacote óptico</i>	<i>14</i>
<i>Figura 2.6 - Arquitetura do nó de acesso da rede</i>	<i>16</i>
<i>Figura 2.7 - Unidade de controle do nó de acesso da rede.....</i>	<i>17</i>
<i>Figura 2.8 - Bloco eletrônico de classificação de quadros e montagem de rajadas do nó de acesso da rede</i>	<i>19</i>
<i>Figura 2.9 - Comutador óptico do nó de acesso da rede</i>	<i>21</i>
<i>Figura 3.1 - IXF1104 e interconexões [26].....</i>	<i>30</i>
<i>Figura 3.2 - SerDes da Texas Instruments [28]</i>	<i>31</i>
<i>Figura 3.3 - SOA encapsulado no CPqD.....</i>	<i>31</i>
<i>Figura 3.4 - Interligação dos módulos do nó de acesso da rede.....</i>	<i>33</i>
<i>Figura 3.5 - Diagrama dos blocos de processamento</i>	<i>34</i>
<i>Figura 4.1 - Placa Processadora adquirida da PLD Applications [30].....</i>	<i>38</i>
<i>Figura 4.2 - Diagrama de blocos da Placa Processadora [30].....</i>	<i>40</i>
<i>Figura 4.3 - Concepção modular do protótipo do nó da rede</i>	<i>42</i>
<i>Figura 4.4 - Desenho da vista lateral das placas Processadora e de Acesso aos Clientes</i>	<i>42</i>
<i>Figura 4.5 - Diagrama de blocos da Placa de Acesso aos Clientes</i>	<i>44</i>
<i>Figura 4.6 - Formato do quadro de dados do padrão Ethernet [26].....</i>	<i>44</i>
<i>Figura 4.7 - Formato dos quadros enviados e recebidos pelo IXF1104.....</i>	<i>45</i>
<i>Figura 4.8 - Interfaces do controlador IXF1104 [26].....</i>	<i>46</i>
<i>Figura 4.9 - Conexão entre o SerDes do IXF1104 e o módulo óptico</i>	<i>47</i>
<i>Figura 4.10 - Microcontrolador de 8 bits utilizado para o controle e gerenciamento dos componentes da placa</i>	<i>48</i>
<i>Figura 4.11 - Diagrama de tempo para um ciclo de escrita do IXF1104 [26].....</i>	<i>49</i>
<i>Figura 4.12 - Tela principal da CLI</i>	<i>51</i>

<i>Figura 4.13 - Seqüência de alimentação do IXF1104 [26].....</i>	<i>52</i>
<i>Figura 4.14 - Circuito de seqüenciamento da Placa de Acesso aos Clientes</i>	<i>52</i>
<i>Figura 4.15 - Posicionamento dos componentes para facilitar o roteamento</i>	<i>55</i>
<i>Figura 4.16 - Aplicativo para cálculo de geometria de trilhas</i>	<i>56</i>
<i>Figura 4.17 - Estrutura da PCI da Placa de Acesso aos Clientes</i>	<i>57</i>
<i>Figura 4.18 - Equalização das linhas da interface SPI-3</i>	<i>58</i>
<i>Figura 4.19 - Equalização de comprimento dos pares diferenciais.....</i>	<i>59</i>
<i>Figura 4.20 - Planos de alimentação</i>	<i>59</i>
<i>Figura 4.21 - Placa de Acesso aos Clientes roteada.....</i>	<i>60</i>
<i>Figura 4.22 - Esquema do teste realizado na placa de Acesso aos Clientes</i>	<i>63</i>
<i>Figura 4.23 - Placa de Acesso aos Clientes sendo testada</i>	<i>64</i>
<i>Figura 4.24 - Conector de dados com os contatos oxidados.....</i>	<i>64</i>
<i>Figura 4.25 - Vista inferior da Placa de Acesso aos Clientes.....</i>	<i>65</i>
<i>Figura 4.26 - Diagrama de blocos da Placa de Acesso ao Anel Óptico</i>	<i>66</i>
<i>Figura 4.27 - Diagrama de blocos do TLK2201A [28].....</i>	<i>67</i>
<i>Figura 4.28 - Formas de onda da recepção dos dados do SerDes [28]</i>	<i>67</i>
<i>Figura 4.29 - Esquema das conexões das interfaces do SerDes</i>	<i>69</i>
<i>Figura 4.30 - Estrutura da PCI da Placa de Acesso ao Anel Óptico.....</i>	<i>71</i>
<i>Figura 4.31 - Placa de Acesso ao Anel Óptico roteada</i>	<i>72</i>
<i>Figura 4.32 - Placa de Acesso ao Anel Óptico montada.....</i>	<i>73</i>
<i>Figura 4.33 - Esquema do teste realizado na placa de Acesso ao Anel Óptico.....</i>	<i>74</i>
<i>Figura 4.34 - Diagrama esquemático do SOA encapsulado</i>	<i>75</i>
<i>Figura 4.35 - Circuito para controle de temperatura do SOA.....</i>	<i>76</i>
<i>Figura 4.36 - Circuito de polarização do SOA.....</i>	<i>77</i>
<i>Figura 4.37 - Estrutura da PCI da Placa de Chaveamento Óptico</i>	<i>78</i>
<i>Figura 4.38 - Placa de Chaveamento Óptico roteada.....</i>	<i>78</i>
<i>Figura 4.39 - Placa de Chaveamento Óptico montada</i>	<i>79</i>
<i>Figura 5.1 - Representação do fluxo de dados no nó de acesso da rede</i>	<i>81</i>
<i>Figura 5.2 - Quadro Ethernet alterado pelo Adaptador Ethernet</i>	<i>83</i>
<i>Figura 5.3 - Sistema de gerenciamento de filas</i>	<i>85</i>

<i>Figura 5.4 - Arquitetura da lógica de processamento criada para o teste do nó de acesso</i>	89
<i>Figura 5.5 - Diagrama do teste do nó da rede</i>	91
<i>Figura 5.6 - Pacote Ethernet com tamanho de 68 Bytes na entrada do bloco</i>	92
<i>Figura 5.7 - Quadro saindo do bloco com 70 Bytes</i>	92
<i>Figura 5.8 - Detalhe dos 2 Bytes no início que indicam o tamanho do pacote</i>	92
<i>Figura 5.9 - Pacote de dados na entrada e saída do bloco Adaptador Ethernet</i>	93
<i>Figura 5.10 - Placas conectadas para o teste do nó de acesso da rede</i>	93
<i>Figura 5.11 - Diagrama do teste inicial com dois nós de acesso</i>	94
<i>Figura 5.12 - Circuito de alimentação com problema da Placa Processadora</i>	95
<i>Figura 5.13 - Dois nós de acesso da rede sendo testados com o uso de ventiladores</i>	96
<i>Figura 5.14 - Diagrama do teste com a Placa de Chaveamento Óptico, link de 20 km e transmissão de vídeo</i>	97
<i>Figura 5.15 - Teste com dois nós de acesso no teste de transferência de vídeo simultânea</i>	98
<i>Figura 5.16 - Detalhe da montagem de dois nós de acesso mais os carretéis de fibra óptica utilizado para o teste de tráfego simultâneo</i>	99

Índice de tabelas

<i>Tabela 2.1 - Diretrizes genéricas para arquiteturas de rede</i>	<i>6</i>
<i>Tabela 3.1 - Tempos de chaveamento.....</i>	<i>27</i>
<i>Tabela 3.2 - Características de chaves ópticas</i>	<i>28</i>
<i>Tabela 4.1 - Geometria das trilhas para sinais de alta frequência da Placa de Acesso ao Anel Óptico.....</i>	<i>56</i>
<i>Tabela 4.2 - Requisitos para fabricação de PCI</i>	<i>61</i>
<i>Tabela 4.3 - Geometria das trilhas para sinais de alta frequência da Placa de Acesso ao Anel Óptico.....</i>	<i>71</i>
<i>Tabela 4.4 - Especificações elétrico-ópticas do SOA [37].....</i>	<i>77</i>

Lista de acrônimos

CPqD	Centro de Pesquisa e Desenvolvimento em Telecomunicações
RCPO	Rede de Chaveamento de Pacotes Ópticos
WDM	multiplexação por divisão em comprimento de onda (<i>Wavelength Division Multiplexing</i>)
OBS	chaveamento de rajadas ópticas (<i>Optical Burst Switching</i>)
OPS	chaveamento de pacotes ópticos (<i>Optical Packet Switching</i>)
OCS	chaveamento de circuitos ópticos (<i>Optical Circuit Switching</i>)
MOPS	chaveamento de pacotes ópticos de múltiplos comprimentos de onda (<i>Multiple-wavelength Optical Packet Switching</i>)
CoS	classe de serviço (<i>Class of Service</i>)
UTD	Unidade de Transporte de Dados
UC	Unidade de Controle
QoS	qualidade de serviço (<i>Quality of Service</i>)
MAC	controle de acesso ao meio (<i>Medium Access Control</i>)
PAT	protocolos de transporte agregado de pacotes (<i>Packet Aggregated Transport</i>)
SAT	satisfeito (<i>Satisfied</i>)
TTL	tempo de vida (<i>Time-to-Live</i>)
DA	endereço de destino (<i>Destination Address</i>)
SA	endereço de origem (<i>Source Address</i>)
CRC	<i>Cyclic Redundancy Check</i>
BP	Bytes de preenchimento (<i>Bytes of Padding</i>)
PS	tamanho do campo BP (<i>Padding Size</i>)
RTOS	sistema operacional de tempo real (<i>Real Time Operating System</i>)
ASIC	circuitos integrados para aplicação específica (<i>Application Specific Integrated Circuits</i>)
FPGA	componente de lógica programável (<i>Field Programmable Gate Array</i>)
VHDL	linguagem de descrição de hardware (<i>VHSIC Hardware Description Language</i>)

NP	processadores de rede (<i>Network Processors</i>)
SOC	matriz de comutação integrada (<i>Switch On a Chip</i>)
SerDes	serializador/deserializador (<i>Serializer/Deserializer</i>)
SOA	amplificador óptico a semicondutor (<i>Semiconductor Optical Amplifier</i>)
RAM	memória de acesso aleatório (<i>Random-Access Memory</i>)
PLL	<i>Phase Locked Loop</i>
SFP	<i>Small Form Pluggable</i>
GbE	<i>Gigabit Ethernet</i>
PCI	Placa de Circuito Impresso
GPIO	<i>General Purpose I/O</i>
PCI	<i>Peripheral Component Interconnect</i>
PMC	<i>PCI Mezzanine Card</i>
LED	diodo emissor de luz (<i>Light Emission Diode</i>)
SFD	<i>Start of Quadro Delimiter</i>
FCS	<i>Quadro Check Sequence</i>
GMII	interface de acesso ao meio (<i>Gigabit Media Independent Interface</i>)
SPI-3	<i>System Packet Interface level 3</i>
CPU	unidade central de processamento (<i>Central Processing Unit</i>)
JTAG	<i>Joint Test Action Group</i>
BGA	<i>Ball Grid Array</i>
CLI	interface de linha de comandos (<i>Command Line Interface</i>)
UART	transmissão e recepção assíncrona universal (<i>Universal Asynchronous Receiver-Transmitter</i>)
ERC	checagem de erros de conexões (<i>Electrical Rule Check</i>)
DRC	checagem de erros do layout (<i>Design Rule Check</i>)
CAM	<i>Computer-Aided Manufacturing</i>
LDHE	Laboratório de Hardware Eletrônico
ENIG	<i>Electroless Nickel/Immersion Gold</i>
HASL	<i>Hot Air Solder Leveling</i>
LOS	ausência de sinal (<i>Loss of Signal</i>)

PRBS	pseudo seqüência de bit aleatória (<i>Pseudo Random Bit Sequence</i>)
NTC	coeficiente de temperatura negativo (<i>Negative Temperature Coefficient</i>)
TEC	refrigerador termoelétrico (<i>Thermoelectric Coolers</i>)
FIFO	primeiro que entra/primeiro que sai (<i>First-In-First-Out</i>)

Apêndice I - Código em C desenvolvido para a Placa de Acesso aos Clientes

Segue abaixo o código desenvolvido em ANSI-C para o microcontrolador de 8 bits para controle do MAC IXF1104 e demais componentes:

```
#include <89c51ac2.h>
#include <stdio.h>
#include "definicao.h"
void main(void)
{
    TMOD = 0x20;    // Timer 1 no modo 2 (auto-reload) para gerar o clock de
    comunicação serial
    IEN0 = 0x9D;    // Interrupção serial, INT0/, habilitado timer's 0 e 1
    SCON = 0x50;    // Porta serial modo 1 - 8 bits
    PCON = 0x80;    // serial double rate clock: SMOD = 1
    TCON = 0x01;    // Transição 1 -> 0 em INT0/
    TH1 = 251;    // Valor a ser recarregado no contador = 251 = 9200 bps x 2 = 19200 bps
    TR1 = 1;    // Ativa o Timer 1 (gerador de clock para comunicação serial)
    TI = 1;    // Flag de transmissão - deve estar em '1' para funcionamento correto
    IPL0 = 1;    // Prioridade INT0/
    reset_mac = 0;    // reset_mac = 0 -> reset = 1, ixf1104 em operação
    CONFIGURACAO();    // MODO DE CONFIGURAÇÃO IXF1104
}
void sequencia(void) interrupt 0 // habilita e desabilita a seq. power up/down
{
    if (seq==1)
    {
        seq = 0;
        DELAY1();
        printf("\n\nPower Up Sequence OK!");
        RESET();
        INIT();
    }
    else
    {
        seq = 1;
        printf("\n\nPower Down Sequence OK!");
        printf("\n\n**Pressione Espaco p/ Continuar**");
    }
}
//-----//
```

```

#include <89c51ac2.h>
#include <stdio.h>
#include "definicao.h"
void CONFIGURACAO(void)
{
    unsigned char opcao;    //variável utilizada para saber qual opcao foi utilizada.
    do
    {
        printf("%c[2J",27);    //função para limpar a tela
        printf("\n\n\n");
        printf("\nCompilacao: %s / %s\n",__DATE__, __TIME__);
        printf("-----\n");
        printf("CONFIGURACAO DE REGISTRADORES MAC IXF1104 V1.0\n\n");
        printf("Desenvolvido no LDHE - Laboratorio de Desenvolvimento de Hardware
        Eletronico\n");
        printf("Gerencia de Sistemas de Comunicacoes Opticas - DRT\n");
        printf("CPqD Telecom & IT Solutions\n");
        printf("-----\n");
        printf("Eduardo Mobilon / Rodrigo Bernardo - 12/2004\n");
        printf("-----\n\n");
        printf("1 - Init.\n");
        printf("2 - Read\n");
        printf("3 - Write\n");
        printf("4 - Local Loopback\n");
        printf("5 - External Loopback\n");
        printf("9 - Reset IXF1104\n");
        printf("\nEscolha a Opcao: ");
        opcao = getchar();
        switch(opcao)
        {
            case '1':
                INIT(); // função de inicializacao
                break;
            case '2':
                READ_REG();    // função de leitura nos registradores
                break;
            case '3':
                WRITE_REG();    // função de escrita nos registradores
                break;
            case '4':
                LOCAL();    // função local loopback
                break;
            case '5':
                EXTERNAL();    // função external loopback
                break;
        }
    }
}

```

```

        case '9':
            RESET();    // função para reset do IXF1104
            break;
        }
    while(opcao != ' ')
    {
        opcao = getchar();
    }
    printf("%c[2J",27);
} while(1);
}
//-----//

```



```

#include <89c51ac2.h>
#include <stdio.h>
#include "definicao.h"
unsigned int add;
unsigned char alto, x;
unsigned long dat;
union
{
    unsigned int word;
    unsigned char Byte[2];
}total;
union
{
    unsigned long word32;
    unsigned char read[4];
}saida;
void READ_REG(void)
{
    printf("\n\nDigite o Endereco do Registrador: ");
    scanf("%x",&add);
    total.word = add;
    add0_7 = total.Byte[1];
    alto = total.Byte[0];
    add8_10 = (add8_10 & 0xF8) | alto;    // mascaramento p/ escrever nos 3 LSB de
add8_10
    for (x=0; x < 4; x++)
    {
        if (x==0) {badd0=0; badd1=0;}    // [7:0]
        else if (x==1) {badd0=1; badd1=0;}    // [15:8]
        else if (x==2) {badd0=0; badd1=1;}    // [23:16]
        else {badd0=1; badd1=1;}    // [31:24]
        cs = 0;
        rd_strobe = 0;
        while (cycle_ind);    // aguarda indicaç o que completou o ciclo R/W
        saida.read[3-x] = dados;
        rd_strobe = 1;
        cs = 1;
    }
    printf("\n\nEndereco = 0x%X",total.word);
    printf("\nValor = 0x%LX",saida.word32);
    printf("\n\n**Pressione Espaco p/ Continuar**");
}
void WRITE_REG(void)
{
    printf("\n\nDigite o Endereco do Registrador: ");

```

```

scanf("%x",&add);
total.word = add;
printf("\nDigite o Valor do Registrador: ");
scanf("%lx",&dat);
saida.word32 = dat;
add0_7 = total.Byte[1];
alto = total.Byte[0];
add8_10 = (add8_10 & 0xF8) | alto;
for (x=0; x < 4; x++)
{
    if (x==0) {badd0=0; badd1=0;}
    else if (x==1) {badd0=1; badd1=0;}
    else if (x==2) {badd0=0; badd1=1;}
    else {badd0=1; badd1=1;}
    cs = 0;
    wr_strobe = 0;
    dados = saida.read[3-x];
    wr_strobe = 1;
    cs = 1;
    //while (cycle_ind);
}
printf("\n\nEndereco = 0x%X",total.word);
printf("\nValor = 0x%LX",saida.word32);
printf("\n\n**Pressione Espaco p/ Continuar**");
}
void DELAY(void)          // aprox. 200us
{
    TMOD = 0x21;
    TH0 = 0xFE;
    TL0 = 0xCC;
    TR0 = 1;
    while (!TF0);
    TR0 = 0;
    TF0 = 0;
}
void DELAY1(void)         // aprox. 15ms
{
    TMOD = 0x21;
    TH0 = 0x86;
    TL0 = 0xE7;
    TR0 = 1;
    while (!TF0);
    TR0 = 0;
    TF0 = 0;
}

```

```

void DELAY2(void)          // aprox. 1us
{
    TMOD = 0x22;
    TH0 = 252;
    TL0 = 252;
    TR0 = 1;
    while (!TF0);
    TR0 = 0;
    TF0 = 0;
}
void RESET(void)
{
    reset_mac = 1;
    DELAY2();
    reset_mac = 0;
    DELAY();                      // delay p/ acesso depois do reset
    printf("\n\nReset OK!");
    printf("\n\n**Pressione Espaco p/ Continuar**");
}
void READ_REG_INIT(void)
{
    total.word = add;
    add0_7 = total.Byte[1];
    alto = total.Byte[0];
    add8_10 = (add8_10 & 0xF8) | alto;
    for (x=0; x < 4; x++)
    {
        if (x==0) {badd0=0; badd1=0;}
        else if (x==1) {badd0=1; badd1=0;}
        else if (x==2) {badd0=0; badd1=1;}
        else          {badd0=1; badd1=1;}
        cs = 0;
        rd_strobe = 0;
        while (cycle_ind);
        saida.read[3-x] = dados;
        rd_strobe = 1;
        cs = 1;
    }
}
void WRITE_REG_INIT(void)
{
    total.word = add;
    saida.word32 = dat;
    add0_7 = total.Byte[1];
    alto = total.Byte[0];
}

```

```

add8_10 = (add8_10 & 0xF8) | alto;
for (x=0; x < 4; x++)
{
    if (x==0) {badd0=0; badd1=0;}
    else if (x==1) {badd0=1; badd1=0;}
    else if (x==2) {badd0=0; badd1=1;}
    else {badd0=1; badd1=1;}
    cs = 0;
    wr_strobe = 0;
    dados = saida.read[3-x];
    wr_strobe = 1;
    cs = 1;
    //while (cycle_ind);}
}
void INIT(void)
{
    add = 0x50c;
    READ_REG_INIT();
    if (saida.word32 == 0x10450013) // testa se MAC está ok
    {
        add = 0x505; // mac soft reset
        dat = 0xf;
        WRITE_REG_INIT();
        add = 0x700; // SPI-3 TX reset
        dat = 0xe0000f;
        WRITE_REG_INIT();
        add = 0x59e; // RX FIFO reset
        dat = 0xf;
        WRITE_REG_INIT();
        add = 0x620; // TX FIFO reset
        dat = 0xf;
        WRITE_REG_INIT();
        DELAY2();
        add = 0x505; // remove mac soft reset
        dat = 0x0;
        WRITE_REG_INIT();
        add = 0x700; // SPI-3 4 port enable @ 8-bit
        dat = 0x20000f;
        WRITE_REG_INIT();
        add = 0x59e; // remove reset
        dat = 0x0;
        WRITE_REG_INIT();
        add = 0x620; // remove reset
        dat = 0x0;
        WRITE_REG_INIT();
        DELAY2();
    }
}

```

```

        add = 0x18;          // CRC enable
        dat = 0x114d;
WRITE_REG_INIT();
add = 0x98;
        dat = 0x114d;
WRITE_REG_INIT();
add = 0x118;
        dat = 0x114d;
WRITE_REG_INIT();
add = 0x198;
        dat = 0x114d;
WRITE_REG_INIT();
add = 0x794;
        dat = 0xf;
WRITE_REG_INIT();
add = 0x500;          // enable channel's
        dat = 0xf;
WRITE_REG_INIT();
        DELAY2();
        add = 0x502;          // link LED
        dat = 0xf;
WRITE_REG_INIT();
add = 0x509;          // LED enable
        dat = 0x2;
WRITE_REG_INIT();
        printf("\n\nInicializacao OK!");
        printf("\n\n**Pressione Espaco p/ Continuar**");
    }
else
    {
        printf("\n\nIXF1104 esta ausente");
        printf("\n\n**Pressione Espaco p/ Continuar**");
    }
}
void LOCAL(void)
{
    add = 0x50c;
    READ_REG_INIT();
    if (saida.word32 == 0x10450013)          // testa se MAC está ok
    {
        add = 0x505;          // mac soft reset
        dat = 0xf;
        WRITE_REG_INIT();
        add = 0x700;          // SPI-3 TX reset
        dat = 0xe0000f;
    }
}

```

```

WRITE_REG_INIT();
add = 0x59e;    // RX FIFO reset
dat = 0xf;
WRITE_REG_INIT();
add = 0x620;    // TX FIFO reset
dat = 0xf;
WRITE_REG_INIT();
DELAY2();
add = 0x505;    // remove mac soft reset
dat = 0x0;
WRITE_REG_INIT();
add = 0x700;    // SPI-3 4 port enable @ 8-bit
dat = 0x20000f;
WRITE_REG_INIT();
add = 0x59e;    // remove reset
dat = 0x0;
WRITE_REG_INIT();
add = 0x620;    // remove reset
dat = 0x0;
WRITE_REG_INIT();
DELAY2();
add = 0x5b2;    // local loopback enable
dat = 0xf00;
WRITE_REG_INIT();
add = 0x794;
dat = 0xf;
WRITE_REG_INIT();
add = 0x500;    // enable channel's
dat = 0xf;
WRITE_REG_INIT();
printf("\n\nLocal Loopback Enable!");
printf("\n\n**Pressione Espaco p/ Continuar**");
}
else
{
    printf("\n\nIXF1104 esta ausente");
    printf("\n\n**Pressione Espaco p/ Continuar**");
}
}
void EXTERNAL(void)
{
    add = 0x50c;
    READ_REG_INIT();
    if (saida.word32 == 0x10450013)    // testa se MAC está ok
    {

```

```

        add = 0x505;          // mac soft reset
        dat = 0xf;
        WRITE_REG_INIT();
        add = 0x700;          // SPI-3 TX reset
        dat = 0xe0000f;
        WRITE_REG_INIT();
        add = 0x59e;          // RX FIFO reset
        dat = 0xf;
        WRITE_REG_INIT();
        add = 0x620;          // TX FIFO reset
        dat = 0xf;
        WRITE_REG_INIT();
        DELAY2();
        add = 0x505;          // remove mac soft reset
        dat = 0x0;
        WRITE_REG_INIT();
        add = 0x700;          // SPI-3 4 port enable @ 8-bit
        dat = 0x20000f;
        WRITE_REG_INIT();
        add = 0x59e;          // remove reset
        dat = 0x0;
        WRITE_REG_INIT();
        add = 0x620;          // remove reset
        dat = 0x0;
        WRITE_REG_INIT();
        DELAY2();
        add = 0x61f;          // external loopback enable
        dat = 0xf;
        WRITE_REG_INIT();
        add = 0x794;
        dat = 0xf;
        WRITE_REG_INIT();
        add = 0x500;          // enable channel's
        dat = 0xf;
        WRITE_REG_INIT();
        printf("\n\nExternal Loopback Enable!");
        printf("\n\n**Pressione Espaco p/ Continuar**");
    }
else
{
    printf("\n\nIXF1104 esta ausente");
    printf("\n\n**Pressione Espaco p/ Continuar**");
}
}
//-----//

```

```
#define dados          P0
#define add0_7         P2
#define add8_10       P1
#define badd0          P1_3
#define badd1          P1_4
#define cs             P1_5
#define wr_strobe      P1_6
#define rd_strobe      P1_7
#define cycle_ind      P4_0
#define seq            P4_1
#define reset_mac      P3_4
#define led            P3_3
```