

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

CHRIS DENNIS TOMÁS HORNA

**Arquiteturas para um Dispositivo de  
Demarcação Ethernet**

Dissertação apresentada como requisito parcial  
para a obtenção do grau de Mestre em Ciência  
da Computação

Prof. Dr. Ricardo Reis  
Orientador

Porto Alegre, agosto de 2009.

## CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Horna, Chris

Normas para Apresentação de Dissertações do Instituto de Informática e do PPGC / Chris Horna – Porto Alegre: Programa de Pós-Graduação em Computação, 2009.

115 f.:il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação. Porto Alegre, BR – RS, 2009. Orientador: Ricardo Reis.

1.ABNT. 2.Processadores de Texto 3.Formatção eletrônica de documentos. I. Reis, Ricardo. II. Arquiteturas para um Dispositivo de Demarcação Ethernet.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. José Carlos Ferraz Hennemann

Vice-reitor: Prof. Pedro Cezar Dutra Fonseca

Pró-Reitora Adjunta de Pós-Graduação: Profa. Valquiria Linck Bassani

Diretor do Instituto de Informática: Prof. Flávio Rech Wagner

Coordenador do PPGC: Prof<sup>a</sup> Luciana Porcher Nedel

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

*"Cuantos creyeron que podía lograrlo?  
Nunca lo sabré... Lo importante es que YO creí.  
Daniel Iparraguirre-Cardenas*

## AGRADECIMENTOS

Dedico este trabajo y todo el esfuerzo colocado en él a mis padres, Dalila Horna Saravía y Felix Tomás Alcalá, por el precioso apoyo que me dan día a día y por demostrarme que hay personas ejemplares en este mundo. Los amo.

A todos mis familiares, en especial a mi segunda madre Nelly Horna por incentivar me a ser cada día mejor y a mi abuelo Javier Horna Campos por su fortaleza. Es esa mi familia, que me enseñó todo este tiempo el valor de la humildad y de la perseverancia. Les agradezco el haberme formado como persona, y solo me resta decirles que me siento muy orgulloso de tenerlos.

A mis grandes amigos en Perú y en el exterior, les agradezco la preocupación que tuvieron y tienen hasta ahora conmigo. En especial, a Jimmy Tarrillo Olano por sus acertadas contribuciones en este trabajo, así como también a Natália Sorgato Nunes por haberme soportado durante la maestría!. Los quiero mucho.

A mi orientador Ricardo Reis, por haber apoyado las ideas que dieron fruto a este trabajo y por abrirme las puertas en el Brasil. Soy grato por la confianza que me diste Ricardo y por la amistad que formamos en estos años.

A mis colegas de la empresa DATACOM. Principalmente, a Marcelo Barcelos por contribuir con las bases de este trabajo y a Sérgio Spader por sus consejos y buenos ánimos. Ambos tienen a su cargo una elite de ingenieros increíbles, del cual afortunadamente formo parte. También quiero incluir a Rafael Ramos gerente del Convênio TERACOM-UFRGS, colega y referencia profesional que siempre estuvo pendiente de este trabajo. A todos ustedes les agradezco por lo aprendido en estos años en Porto Alegre.

A los colegas Jean Hamerski por sus contribuciones en el desarrollo de OAM Ethernet y a Fábio Ramos por haberme ayudado en los testes físicos y en el desarrollo de la denominada Packet FIFO. Sin ustedes no hubiera sido posible completar este trabajo, muchas gracias!

A los chicos del Grupo de Microelectrónica de la PUCP y a su director Carlos Silva Cárdenas. A ustedes les dedico esta contribución, como prueba de que somos capaces de hacer cosas importantes. Yo me inicié en este grupo, donde aprendí los misterios de la microelectrónica y donde pase muy buenos momentos antes de mi partida del Perú.

Y a los tios de Trujillo, solo decirles que nunca bajé la guardia mientras estuve lejos de mi Pátria y que al final venceremos!

# SUMÁRIO

<b>LISTA DE FIGURAS.....</b>	<b>9</b>
<b>LISTA DE TABELAS.....</b>	<b>11</b>
<b>RESUMO.....</b>	<b>12</b>
<b>ABSTRACT.....</b>	<b>14</b>
<b>1 INTRODUÇÃO.....</b>	<b>16</b>
<b>2 A TECNOLOGIA ETHERNET E SUAS EXTENSÕES NAS REDES METROPOLITANAS .....</b>	<b>18</b>
2.1 HISTÓRICO.....	18
2.2 CARACTERÍSTICAS DA ETHERNET NAS REDES LOCAIS.....	19
2.2.1 <i>O Quadro Ethernet segundo a norma IEEE 802.3.....</i>	<i>20</i>
2.2.2 <i>O Protocolo de Acesso ao Meio: CSMA/CD.....</i>	<i>21</i>
2.2.3 <i>Enlace Duplex e Mecanismo de Controle de Fluxo.....</i>	<i>22</i>
2.2.4 <i>Interface Independente do Meio e o Gerenciamento da Camada Física... </i>	<i>23</i>
2.2.5 <i>A Terceira Geração: Gigabit Ethernet.....</i>	<i>26</i>
2.2.6 <i>Parâmetros de Implementação da Ethernet.....</i>	<i>29</i>
2.3 OAM.....	29
2.3.1 <i>EFM– IEEE 802.3ah.....</i>	<i>30</i>
2.3.2 <i>CFM – IEEE 802.1ag .....</i>	<i>34</i>
<b>3 O DISPOSITIVO DE DEMARCAÇÃO ETHERNET .....</b>	<b>35</b>
3.1 FUNÇÕES ESSENCIAIS DO EDD.....	35
3.2 TIPOS DE DEMARCAÇÃO.....	36
3.2.1 <i>Demarcação UNI.....</i>	<i>36</i>
3.2.2 <i>Demarcação NNI.....</i>	<i>36</i>
3.3 USO DO OAM NO EDD .....	37
3.3.1 <i>Uso da função OAM Discovery.....</i>	<i>37</i>
3.3.2 <i>Uso da função OAM Remote Failure: Link Fault/Critical Event .....</i>	<i>38</i>
3.3.3 <i>Uso da função OAM Remote Failure: Dying Gasp .....</i>	<i>38</i>
3.3.4 <i>Uso da função OAM Port Loopback: Link .....</i>	<i>39</i>
3.3.5 <i>Uso da função OAM Loopback (não-intrusivo): Nível do serviço .....</i>	<i>39</i>
3.4 TWO-PORT MAC RELAY – IEEE P802.1AJ.....	40
<b>4 SISTEMAS EMBARCADOS RECONFIGURÁVEIS.....</b>	<b>42</b>
4.1 ARQUITETURA DE UM FPGA.....	42
4.2 SYSTEM-ON-A-CHIP (SOC) .....	44
4.2.1 <i>Linguagens para Descrição de SoCs.....</i>	<i>44</i>
4.2.2 <i>Arquitetura Genérica de um SoC.....</i>	<i>45</i>
4.3 SOFT PROCESSOR MICROBLAZE .....	46

<b>5</b>	<b>DESCRIÇÃO DO EDD PROPOSTO.....</b>	<b>48</b>
5.1	CENÁRIO DE REFERÊNCIA DO EDD PROPOSTO .....	48
5.2	ESPECIFICAÇÃO TÉCNICA DO EDD PROPOSTO.....	49
5.3	ARQUITETURA DO EDD PROPOSTO VISANDO UMA IMPLEMENTAÇÃO EM FPGAs .....	50
<b>6</b>	<b>DESENVOLVIMENTO DOS NÚCLEOS IP PARA O EDD PROPOSTO.....</b>	<b>54</b>
6.1	PROJETO DOS NÚCLEOS IP DO EDD PROPOSTO .....	54
6.1.1	<i>IP Soft Core FEMAC</i> .....	54
6.1.2	<i>IP Soft Core GEMAC</i> .....	65
6.1.3	<i>IP Soft Core MIIM</i> .....	73
6.1.4	<i>IP Soft Core Packet FIFO</i> .....	75
6.1.5	<i>IP Soft Core OAM Ethernet</i> .....	76
6.1.6	<i>IP Hard/Soft Core PHY1000X</i> .....	78
6.2	IMPLEMENTAÇÃO DOS MÓDULOS IP .....	83
6.2.1	<i>Resultados da implementação dos IP Soft Cores FEMAC e GEMAC</i> .....	83
6.2.2	<i>Resultados da implementação do IP Soft Core MIIM</i> .....	84
6.2.3	<i>Implementação do IP Soft Core Packet FIFO</i> .....	85
6.2.4	<i>Implementação do IP Soft Core OAM Ethernet</i> .....	85
6.2.5	<i>Resultados da implementação do IP Hard/Soft Core PCS/PMA 1000x</i> .....	90
6.3	VALIDAÇÃO DOS MÓDULOS IP .....	91
6.3.1	<i>Cenário de testes do IP Soft Core FEMAC</i> .....	91
6.3.2	<i>Cenário de testes dos IP Soft Cores GEMAC 1000, PCS/PMA 1000X e PacketFIFO</i> .....	92
6.3.3	<i>Cenários de testes do IP Soft Core OAM Ethernet</i> .....	94
<b>7</b>	<b>DESENVOLVIMENTO DOS CONVERSORES DE MEIOS ETHERNET.....</b>	<b>102</b>
7.1	A JIGA BERT PARA VALIDAÇÃO DOS CONVERSORES DE MEIOS ETHERNET .....	102
7.2	A SOLUÇÃO MECALIMBA .....	103
7.3	PROPOSTAS DE VALIDAÇÃO DOS CONVERSORES DE MEIOS ETHERNET .....	105
7.3.1	<i>Validação Tráfego em taxas críticas:</i> .....	105
7.3.2	<i>Validação de Desempenho</i> .....	107
7.3.3	<i>Validação de Conformidade com a norma IEEE 802.3</i> .....	108
7.4	RESULTADOS DE IMPLEMENTAÇÃO DO EDD PROPOSTO.....	109
<b>8</b>	<b>CONCLUSÕES E TRABALHOS FUTUROS.....</b>	<b>110</b>
8.1	PRINCIPAIS REALIZAÇÕES .....	111
8.2	TRABALHOS FUTUROS .....	111
<b>9</b>	<b>REFERÊNCIAS.....</b>	<b>112</b>

## LISTA DE ABREVIATURAS E SIGLA

ASIC	Application-Specific Integrated Circuit
BERT	Bit-Error Rate Test
CFM	Connectivity Fault Management
CLE	Customer Located Equipment
CSMA/CD	Carrier Sense Multiple Access/Collision Detection
EDD	Ethernet Demarcation Device
EFM	Ethernet in the First Mile
FIFO	First in, First output
FPGA	Field Programmable Gate array
GMII	Gigabit Media Independent Interface
IFG	Inter-frame Gap
IP	Intellectual Property
LAN	Local Area Network
LLC	Logical Link Control
MAC	Media Access Control
MAN	Metropolitan Area Network
MDC	Management Data Clock
MDIO	Management Data Input/Output
MEF	Metro Ethernet Forum
MEN	Metropolitan Ethernet Network
MII	Media Independent Interface
MIIM	Media Independent Interface Management
NID	Network Interface Device
NNI	Network-to-Network Interface
NTE	Network Termination Equipment
NTU	Network Termination Unit
OAM	Operation, Administration and Maintenance

PCS	Physical Coding Sublayer
PDU	Protocol Data Unit
PMA	Physical Medium Attachment
PMD	Physical Medium Dependent
PRBS	Pattern Pseudo-Random Bit Sequence
SFP	Small Form-factor Pluggable
SoC	System-on-a-Chip
SoPC	System-on-a-Programmable-Chip
SLA	Service Level Agreement
STA	Station Management Entity
TPMR	Two-Port Media Access Control (MAC) Relay
UNI	User/Network Interface
VLAN	Virtual LAN
WAN	Wide Area Network



## LISTA DE FIGURAS

Figura 2.1: Relação entre a arquitetura IEEE 802.3 e o RM-OSI (IEEE, 2005a) .....	19
Figura 2.2: Estrutura dos quadros 802.3 (básico e <i>tagged</i> ), baseado em (IEEE, 2005a) ..	20
Figura 2.3: Diagrama de Fluxo simplificado do protocolo CSMA/CD.....	22
Figura 2.4: Estrutura do quadro especial de <i>pause</i> .....	23
Figura 2.5: Gerenciamento dos Dispositivos da Camada Física .....	24
Figura 2.6: Operação de leitura típica no gerenciamento do PHY .....	25
Figura 2.7: Operação de escrita típica no gerenciamento do MII .....	25
Figura 2.8: Estrutura do quadro Ethernet com extensão de portadora.....	26
Figura 2.9: Transmissão de rajada de quadros .....	26
Figura 2.10: Máquina de estados da entidade Multiplexador (IEEE, 2005a).....	34
Figura 3.1: Cenário de atuação do EDD, (Bray, 2006).....	35
Figura 3.2: Funções do EDD, (Bray, 2006).....	36
Figura 3.3: Cenário da demarcação UNI, (Bray, 2006).....	36
Figura 3.4: Cenário de demarcação NNI .....	37
Figura 3.5: Função OAM <i>Discovery</i> , (Bray, 2006).....	37
Figura 3.6: Função OAM <i>Remote Failure: Link Fault/Critical Event</i> , (Bray, 2006) .....	38
Figura 3.7: Função OAM <i>Remote Failure (Dyng Gasp)</i> , (Bray, 2006).....	39
Figura 3.8: Função OAM <i>Port Loopback: Link</i> , (Bray, 2006).....	39
Figura 3.9: OAM <i>Loopback</i> (não-intrusivo): Nível do serviço, (Bray, 2006).....	39
Figura 4.1: Arquitetura de um FPGA genérico (ZEIDMAN, 2001).....	42
Figura 4.2: Interconexão programável .....	44
Figura 4.3: Arquitetura genérica de um SoC (OST, 2004).....	45
Figura 4.4: Diagrama em blocos do MicroBlaze (MicroBlaze, 2007).....	47
Figura 5.1: Cenário de Referência do EDD proposto (MARTIN, 2007).....	48
Figura 5.2: Diagrama de Blocos do EDD proposto.....	50
Figura 5.3: Placa de desenvolvimento AVNET Virtex-II Pro .....	51
Figura 5.4: Duas arquiteturas para o EDD proposto .....	52
Figura 5.5: Arquitetura HW/SW do EDD proposto.....	53
Figura 6.1: IP <i>Soft Core</i> FEMAC e o modelo OSI .....	54
Figura 6.2: Módulos funcionais do IP <i>Soft Core</i> FEMAC.....	55
Figura 6.3: Diagrama de blocos do módulo Receptor.....	56
Figura 6.4: Diagrama de blocos simplificado do módulo Transmissor .....	57
Figura 6.5: Diagrama de blocos simplificado do módulo de Controle de fluxo .....	59
Figura 6.6: Diagrama temporal de uma típica recepção no lado do Cliente .....	64
Figura 6.7: Diagrama temporal de uma típica transmissão no lado do Cliente .....	64
Figura 6.8: IP <i>Soft Core</i> GEMAC e o modelo OSI.....	65
Figura 6.9: Módulos funcionais do IP <i>Soft Core</i> GEMAC.....	66
Figura 6.10: Diagrama de blocos simplificado do módulo Receptor .....	67
Figura 6.11: Diagrama de blocos simplificado do módulo transmissor.....	68
Figura 6.12: IP <i>Soft Core</i> MIIM e o modelo OSI .....	73
Figura 6.13: Diagrama de blocos simplificado do MIIM .....	74
Figura 6.14: Módulos do IP <i>Soft Core</i> Packet-FIFO .....	75
Figura 6.15: IP Módulos funcionais do <i>Soft Core</i> OAM Ethernet e o modelo OSI.....	76

Figura 6.16: Módulos funcionais do IP <i>Soft Core</i> OAM Ethernet .....	77
Figura 6.17: Diagrama de Blocos funcional do IP Hard/ <i>Soft Core</i> PHY1000X.....	78
Figura 6.18: Máquina de estados do Receptor PCS, parte a (IEEE, 2005a) .....	79
Figura 6.19: Máquina de estados do Receptor PCS, parte b (IEEE, 2005a) .....	80
Figura 6.20: Máquina de estados de <i>ordered_set</i> do transmissor PCS (IEEE, 2005a) .....	81
Figura 6.21: Máquina de estados de <i>code-group</i> do transmissor PCS (IEEE, 2005a).....	82
Figura 6.22: IP <i>Soft Core</i> OAM Ethernet e sua integração com o processador MicroBlaze .....	86
Figura 6.23: Plataforma para a validação hardware dos Cores.....	91
Figura 6.24: Cenário de testes para a validação física do IP <i>Soft Core</i> FEMAC .....	92
Figura 6.25: Cenário de testes para a validação física dos IP <i>Soft Cores</i> GEMAC e PHY1000X.....	93
Figura 6.26: Quadros replicados visualizados no Ethereal .....	94
Figura 6.27: Cenário OAM1 de teste da Interface OAM_Sublayer/MAC e OAM_Sublayer/OAM_Client .....	95
Figura 6.28: Fluxo de pacotes gerado no Cenário OAM1 .....	96
Figura 6.29: Cenário OAM2 de teste da Interface OAM_Sublayer/MAC e OAM_Sublayer/OAM_Client .....	97
Figura 6.30: Fluxo de pacotes gerado no Cenário OAM2.....	98
Figura 6.31: Cenário OAM3 de validação da Interface OAM_Client e teste do OAM_Client rodando em software .....	99
Figura 6.32: Fluxo de pacotes gerado no Cenário 3 com OAM_Client no modo passivo .....	100
Figura 6.33: Fluxo de pacotes gerado no Cenário 3 com OAM_Client no modo ativ ...	101
Figura 7.1: Jiga BERT.....	103
Figura 7.2: Conversor de Meios FE2GE sem mecanismo MECALIMBA.....	104
Figura 7.3: Conversor de Meios FE2GE com mecanismo MECALIMBA .....	105
Figura 7.4: Teste Full Line Rate do Conversor de Meios GE2GE.....	106
Figura 7.5: Testes de tráfego utilizando ferramenta “Ping Flood” .....	107
Figura 7.6: Cenários de validação utilizando trafego UDP/TCP .....	108
Figura 7.7: Cenário de validação para testes de conformidade IEEE 802.3 .....	109

## LISTA DE TABELAS

Tabela 2.1: Sinais que compõem a Interface Independente do Meio.....	23
Tabela 2.2: Codificação do protocolo de Comunicação MDIO.....	25
Tabela 2.3: Codificação GMII para transmissão de quadros, baseado em (IEEE, 2005a) .....	27
Tabela 2.4: Codificação GMII para recepção de quadros, baseado em (IEEE, 2005a).	28
Tabela 2.5: Valores limites para implementação da tecnologia Ethernet, baseado em (IEEE, 2005a) .....	29
Tabela 3.1: Projetos do grupo de trabalho IEEE802.1 (ROCHOL, 2006) .....	40
Tabela 6.1: Definição dos bits do vetor de estatística da recepção no Core .....	60
Tabela 6.2: Definição dos bits do vetor de estatística da transmissão no Core .....	61
Tabela 6.3: Parâmetros de implementação do Core .....	62
Tabela 6.4: Sinais da Interface Cliente de 8 bits do Core.....	63
Tabela 6.5: Definição dos bits do vetor de estatística da recepção no IP <i>Soft Core</i> GEMAC .....	69
Tabela 6.6: Definição dos bits do vetor de estatística da transmissão no IP <i>Soft Core</i> GEMAC .....	70
Tabela 6.7: Vetores de configuração do IP <i>Soft Core</i> GEMAC .....	71
Tabela 6.8: Sinais da Interface Cliente de 8 bits do Core.....	72
Tabela 6.9: Resultados da síntese dos Cores FEMAC e GEMAC .....	83
Tabela 6.10: Parâmetros de área obtidos após Síntese lógica dos Módulos FEMAC e GEMAC .....	84
Tabela 6.11: Desempenho do <i>Soft Core</i> MIIM .....	84
Tabela 6.12: Resultados da síntese dos sub-módulos que compõem o PacketFIFO .....	85
Tabela 6.13: Sinais do módulo IP <i>Soft Core</i> OAM Ethernet .....	86
Tabela 6.14: Sinais do módulo OAMAC_Subla.....	88
Tabela 6.15: Mapeamento dos 32 bits na interface OPB.....	90
Tabela 7.1: Estatísticas de <i>Ping Flood</i> .....	107
Tabela 7.2: Recursos utilizados pelo Conversor de Meios GE2GE .....	109

## RESUMO

Na atualidade, as redes públicas de comunicação de dados representam uma nova oportunidade para a aplicação das tecnologias IEEE 802 (baseadas na tecnologia Ethernet). Tanto nas redes de acesso, como nas redes metropolitanas e de núcleo, existe uma crescente demanda pela instalação de equipamentos com grande número de interfaces Ethernet. Em alguns casos, opta-se por equipamentos capazes de emular os serviços das tecnologias legadas ATM/SDH sobre Ethernet e viceversa. Nesse sentido, novos protocolos e novas formas de transmissão de dados utilizando a tecnologia Ethernet estão surgindo para consolidar a convergência das redes de comutação de circuitos (infraestrutura legada) e as redes de comutação de pacotes; com a ideia de constituir uma rede mais homogênea, flexível e de baixo custo. Um claro exemplo é a adesão dos protocolos de Operação, Administração e Manutenção (OAM) nas redes Ethernet, os quais permitem um nível de controle semelhante ao de tecnologias como ATM e SDH. OAM possibilita a monitoração de falhas na rede, a configuração e o acompanhamento dos eventos de segurança, assim como também a contabilização de tráfego por assinante; permitindo desta forma o atendimento de diferentes SLAs (*Service-Level Agreements*) de clientes. Para que isto seja uma realidade, é muito importante reforçar o controle da borda que delimita a rede do cliente final da rede pública. Com esse fim, estão surgindo normas como a IEEE P802.1aj, que define um dispositivo de demarcação de rede que serve como entidade controladora de serviços entre o provedor e o cliente final; sendo sua principal característica o suporte OAM no enlace com o provedor de serviços. Este dispositivo - conhecido comercialmente como *Ethernet Demarcation Device* (EDD)- é o foco do presente trabalho. Este trabalho tem como objetivo principal desenvolver arquiteturas *System-on-a-Programmable Chip* (SoPC) para um EDD de duas portas, partindo do desenvolvimento de módulos de propriedade intelectual (IP). Foram projetadas duas arquiteturas de EDD, as quais permitem o encaminhamento de pacotes entre duas portas Ethernet e incorporam um processador MicroBlaze para implementação Software do protocolo OAM, segundo a norma IEEE 802.3ah. Como resultado, foram elaborados 7 módulos IP: Módulo *Fast Ethernet MAC* (FEMAC), Módulo *Gigabit Ethernet MAC* (GEMAC), Módulo *Packet FIFO*, Módulo *OAM Ethernet*, Módulo *MII Management* (MIIM), Módulo *PHY Ethernet PCS/PMA 1000Base-X* (PHY1000X) e Módulo *Bit-Error Rate Tester* (BERT). Todos os módulos foram descritos em VHDL e logo sintetizados para um dispositivo FPGA da família Virtex-II Pro da Xilinx, quanto para *standard-cells* utilizando a tecnologia CMOS AMS 0.35µm unicamente nos Módulos FEMAC e GEMAC. Os resultados de síntese mostram que o Módulo MIIM e o Módulo PHY1000X possuem um melhor aproveitamento de recursos de área que seus equivalentes disponíveis no OpenCores e no CoreGen da Xilinx, respectivamente. As arquiteturas SoPC foram prototipadas sobre a placa de desenvolvimento AVNET Virtex-II Pro, a qual permite comunicação com

dispositivos de rede através de interfaces elétricas e ópticas. Finalmente, é proposta uma metodologia de validação física das arquiteturas alvo para estas atenderem o regime de vazão máxima (1Gbit/s ou 100Mbit/s), assim como também testes de conformidade como os definidos pela norma IEEE 802.3.

**Palavras-Chave:** Ethernet, Redes Metropolitanas, Módulos IP, Dispositivo de Demarcação Ethernet, Lógica Programável, Microeletrônica.

## ABSTRACT

Nowadays, public networks represent a new opportunity for the application of IEEE 802 technologies, which have their basis on Ethernet Technology. In both Access and Metropolitan and Core networks there is a growing demand for the installation of equipments with a large number of Ethernet interfaces. In some cases, equipments capable of emulating the services of the ATM/SDH legacy technologies over Ethernet (and vice versa) are chosen. In this manner, new protocols and data transmission forms using Ethernet technology are emerging in order to consolidate the convergence of circuit switching networks (traditional infrastructure) and packet switching networks; with the common objective of constituting a more uniform, flexible, low-cost network. A good example is the incorporation of Operation, Administration and Maintenance (OAM) protocols in Ethernet networks, which allow a control level similar to that one of technologies such as ATM and SDH. OAM allows the monitoring of network fails, the configuration and tracking the security events, as well as the counting of traffic per client in a way that permits to attend several SLAs (Service-Level Agreements). In order to bring this to reality, it is critical to reinforce the control of the edge which limits the client network from the public networks. With this aim, standards such as IEEE P802.1aj are emerging; this standard defines a network demarcation device, which is used as a service controlling entity between the provider and the end customer, having as main feature the OAM support in the link with the service provider. This work is focused on this device, commercially known as Ethernet Demarcation Device (EDD). The principal objective of this work is to develop SoPC (System-on-a-Programable chip) architectures for an EDD, starting from the development of Intellectual Property Cores (IP). Two EDD architectures were designed, which allow the packet forwarding between two Ethernet interfaces and incorporate a Soft processor Microblaze for the SW implementation of the OAM protocol according to the standard IEEE802.3ah. As a result, eight IP cores were elaborated: Soft IP Core Fast Ethernet MAC (FEMAC), Soft IP Core Gigabit Ethernet MAC (GEMAC), Soft IP Core Packet FIFO, Soft IP Core OAM Ethernet, Soft IP Core MII Management (MIIM), Soft IP Core PHY Ethernet PCS/PMA 1000Base-X (PHY1000X) and the Soft IP Core Bit-Error Rate Tester (BERT). All IP modules were described in VHDL and then synthesized for the FPGA Xilinx Virtex-II Pro device, as well as for standard-cells using the CMOS AMS 0.35um technology for the modules FEMAC and GEMAC. The synthesis results show that the module MIIM and module PHY1000X have a better use of the area resources than the ones available in OpenCores and CoreGen of Xilinx respectively. The SoPC architectures were prototyped on AVNET Virtex-II Pro Development kit Board, which allows communication with network devices through electrical and optical interfaces. Finally, we propose a validation methodology of both architecture so these are able to

attend a maximum throughput regimen (1Gbit/s ou 100Mbit/s), as well as appropriate levels of approval with what standard IEEE 802.3 defines.

**Keywords:** Ethernet, Metropolitan networks, IP Cores, Ethernet Demarcation Device, Programmable Logic, Microelectronics.

# 1 INTRODUÇÃO

Os clientes residenciais e corporativos das redes de comunicação de dados estão demandando por serviços e aplicações cada vez mais sofisticados. Os provedores de serviços respondem a essa tendência de maneira limitada utilizando tecnologias legadas de comutação de circuitos, tais como ATM (*Asynchronous Transfer Mode*), *Frame Relay* ou SONET/SDH (*Synchronous Optical Networking/ Synchronous Digital Hierarchy*). Uma solução alternativa é a inserção da tecnologia Ethernet nestes novos cenários, permitindo desta forma, uma redução dos custos operacionais gastos pelas operadoras de rede e a possibilidade de transportar serviços com uma largura de banda mais flexível que as oferecidas pelas infra-estruturas tradicionais. Para isto, estão sendo desenvolvidas diversas extensões da Ethernet nos grupos de trabalho da IEEE (*Institute of Electrical and Electronics Engineers*), ITU-T (*International Telecommunication Union*) e do *Metro Ethernet Forum* (MEF); os quais definem uma série de novas funções e serviços, com o objetivo comum de utilizar a Ethernet como tecnologia de transporte sobre Redes Metropolitanas (*Metropolitan Area Networks* ou MANs) e Redes de Longo Alcance (*Wide Area Networks* ou WANs).

No âmbito das redes metropolitanas e de acesso, existe uma crescente demanda pela criação de dispositivos de demarcação de rede, capazes de proporcionar um ponto claro de demarcação entre as redes de usuário e do provedor de serviços. Recentemente, estes dispositivos têm adquirido o nome comercial de *Ethernet Demarcation Device* (EDD), e se caracterizam por ter a capacidade de monitorar pontos estratégicos da rede para facilitar o controle ponta-a-ponta dos serviços IEEE 802 ou “IEEE 802 emulados”. Em conclusão, este dispositivo ajuda o cumprimento dos SLAs contratados pelos clientes.

Atualmente, a norma IEEE P802.1aj (ainda em processo de desenvolvimento) é o responsável pela especificação deste dispositivo de demarcação, chamando-o de *Two-Port Media Access Control (MAC) Relay* ou TPMR.

Do exposto, pode-se extrair que a tecnologia relacionada à demarcação LAN/MAN é ainda insipiente. Porém, a principal motivação do presente trabalho é atender aos novos desafios impostos pela tecnologia “Ethernet Grau de Operadora” (*Carrier-Class Ethernet*). Nesse contexto, o presente trabalho propõe duas arquiteturas para um EDD de duas portas, através do desenvolvimento de um conjunto de Módulos de Propriedade Intelectual Flexíveis (do inglês, *Soft Intellectual Property Cores* ou *Soft IP Cores*); visando, inicialmente, uma implementação sobre um dispositivo reconfigurável FPGA e tendo como escopo futuro o projeto ASIC (*Application-Specific Integrated Circuit*) do EDD proposto.



A metodologia utilizada neste trabalho segue a seguinte sequência: Estudo das especificações e os critérios de implementação em hardware digital das principais camadas da Ethernet (definidas na norma IEEE 802.3) e dos mecanismos necessários para tratamento de quadros, assim como também opções de validação e testes de enlaces Ethernet (segundo recomendações seguidas na indústria). Em seguida, aperfeiçoar o uso das ferramentas de CAD (*Computer-Aided Design*) para microeletrônica e a linguagem de descrição de hardware VHDL (*Very High Speed Integrated Circuit Hardware Description Language*) para o projeto de Módulos IP com funções específicas. Finalmente, realizar a integração dos Módulos IP desenvolvidos e outros componentes de hardware dedicados numa plataforma do tipo SoPC (*System-on-a-Programmable Chip*) para a implementação e validação das duas arquiteturas do EDD de duas portas.

Este trabalho integra-se com o setor industrial através do Convênio TERACOM-UFRGS – em parceria com a Empresa Datacom-Telemática (empresa brasileira líder na fabricação de equipamentos de telecomunicações)–, com a finalidade de obter uma especificação técnica do trabalho mais adequada com a realidade tecnológica brasileira.

O trabalho está organizado em sete capítulos: O Capítulo 2, denominado “A Tecnologia Ethernet e suas Extensões nas Redes Metropolitanas”, apresenta uma visão geral da tecnologia Ethernet, descrevendo suas principais características tanto para Redes Locais (*Local Area Networks* ou LANs) como para redes MAN.

O Capítulo 3, denominado “O Dispositivo de Demarcação Ethernet”, contém o estado-da-arte deste novo mecanismo; isto é: suas funções primárias e os seus cenários de atuação.

O Capítulo 4, denominado “Sistemas Embarcados Reconfiguráveis”, apresenta conceitos básicos dos dispositivos FPGAs, SoCs e o *Soft Processor Microblaze*.

O Capítulo 5, denominado “Descrição do EDD Proposto”, expõe as especificações do EDD proposto e a descrição de duas (2) possíveis arquiteturas Hardware/Software.

O Capítulo 6, denominado “Desenvolvimento dos Módulos IP para o EDD proposto”, descreve o projeto, implementação e validação de sete (7) *IP Soft Cores*, os quais serão úteis posteriormente para a validação das duas arquiteturas do EDD proposto.

O Capítulo 7, denominado “Desenvolvimento dos Conversores de Meios Ethernet”, apresenta a integração e validação dos Conversores de Meios que fazem parte do EDD proposto, partindo do desenvolvido no Capítulo 6.

Por fim, o Capítulo 8 apresenta as conclusões deste trabalho, principais contribuições para as áreas acadêmica e industrial, e algumas sugestões para trabalhos futuros.

## 2 A TECNOLOGIA ETHERNET E SUAS EXTENSÕES NAS REDES METROPOLITANAS

Depois da consolidação da tecnologia Ethernet nas redes locais e corporativas, uma série de aperfeiçoamentos vem sendo impulsionados pela IEEE e pelo MEF, com o objetivo de expandir a utilização da Ethernet nos ambientes metropolitanos e nos *backbones* de longa distância. Novos protocolos estão sendo incorporados de maneira a atribuir bons índices de qualidade de serviço, robustez e segurança; de modo a se aproximar das características das redes baseadas na comutação de circuitos, mantendo as vantagens que oferecem as redes de pacotes.

Este capítulo tem o objetivo de descrever uma visão geral da tecnologia Ethernet, no âmbito LANs, e sua introdução nas MANs. No final, é descrita uma nova funcionalidade utilizada nas Redes Metropolitanas Ethernet (*Metropolitan Ethernet Network* ou MEN) conhecida como OAM (*Operation, Administration and Maintenance*). OAM é responsável pelo suporte à detecção automática de falhas e medição de desempenho da rede.

### 2.1 Histórico

Na década de 70, Bob Metcalfe e David Boggs (ambos na então *Xerox Palo Alto Research Center*) desenvolveram um sistema de rede titulado com o nome “Ethernet”. Essa rede permitia a transferência de dados entre estações de trabalho e impressoras a laser fazendo uso de um único canal de comunicação (METCALFE, 1976). Posteriormente, no ano 1980, a idéia de Metcalfe e Boggs foi padronizada pela primeira vez pelo consórcio de empresas DEC-Intel-Xerox com o nome de padrão DIX-Ethernet. Finalmente, no ano 1985, sob a direção da IEEE *Standards Association* definiu-se o padrão Ethernet oficial para redes locais, identificando-o com o código IEEE 802.3.

Este foi o ponto de partida da tecnologia de rede mais difundida das últimas três décadas, desde sua concepção para redes LAN até sua atual extensão sobre redes MAN e WAN. Sua popularidade e crescimento é devido ao acelerado processo de padronização da sua taxa de transmissão: 10 Mbits/s em 1983, 100 Mbits/s em 1995, 1 Gbits/s em 1998, e 10 Gbits/s em 2002 (FRAZIER, 1999). Ao mesmo tempo, a tecnologia também evoluiu para suportar vários meios de transmissão (cabo coaxial, par trançado, fibra óptica) e diversas topologias de rede (físicas e lógicas).

O sucesso da Ethernet deve-se às suas características de baixo custo por porta, escalabilidade, simplicidade de uso, suporte a uma ampla gama de serviços (incluindo dados, voz e vídeo) e seu alto desempenho nas redes locais. Estas características

somadas ao rigor da sua especificação permitiram uma excelente integração com outras infraestruturas de rede legadas, tais como ATM, *Frame Relay* ou SONET/SDH.

Com o advento da Internet, Ethernet teve garantida sua adoção global nas redes locais, sendo que 90% do tráfego baseado no protocolo IP é originado em LANs Ethernet (CHIRUVOLU *et al.*, 2004). Com tamanha infraestrutura implantada nas redes locais do mundo, a Ethernet passou ser vista como uma nova alternativa nas redes MAN e WAN, tendo como objetivo simplificar a ligação cliente-provedor e otimizar o acesso aos *backbones* das redes.

Conseqüentemente foi necessária a adição de uma serie de novas funções e serviços para permitir com que a Ethernet alcance o acesso metropolitano (*Metro Ethernet*) e finalmente o grau de operadora (*Carrier-Class Ethernet*). Uma das principais preocupações na introdução da tecnologia Ethernet das redes de longo alcance é a manutenibilidade das mesmas. Nesse sentido, OAM nas redes Ethernet torna-se essencial (McFARLAND *et al.*, 2005) (OHTA, 2006).

## 2.2 Características da Ethernet nas Redes Locais

Ao longo dos anos, as redes locais foram projetadas a partir das normas definidas pelo grupo de trabalho IEEE 802; quem desde 1983 vem sendo dirigido por um comitê da IEEE *Computer Society*. Estas normas são adotadas nos Estados Unidos pelo *American National Standards Institute* (ANSI), e posteriormente revisados e republicados como padrões internacionais pela *International Standards Organization* (ISO), com a designação ISO8802 (SOARES *et al.*, 1995). Da lista de normas do grupo de trabalho IEEE 802, a norma IEEE 802.3 é uma das mais destacadas e se caracteriza por definir a arquitetura de rede da tecnologia Ethernet. As camadas da norma IEEE 802.3 são compatíveis com as duas primeiras camadas do Modelo de Referência para Interconexão de Sistemas Abertos (*Reference Model for Open Systems Interconnection* ou RM-OSI). A segunda camada<sup>1</sup> do RM-OSI é conhecida como camada de Enlace, e é representada na norma IEEE 802.3 pela sub-camada de Controle de Acesso ao Meio (*Media Access Control* ou MAC) e pela sub-camada de Controle de Enlace Lógico (*Logical Link Control* ou LLC) (IEEE, 2005a), como observado na figura 2.1.

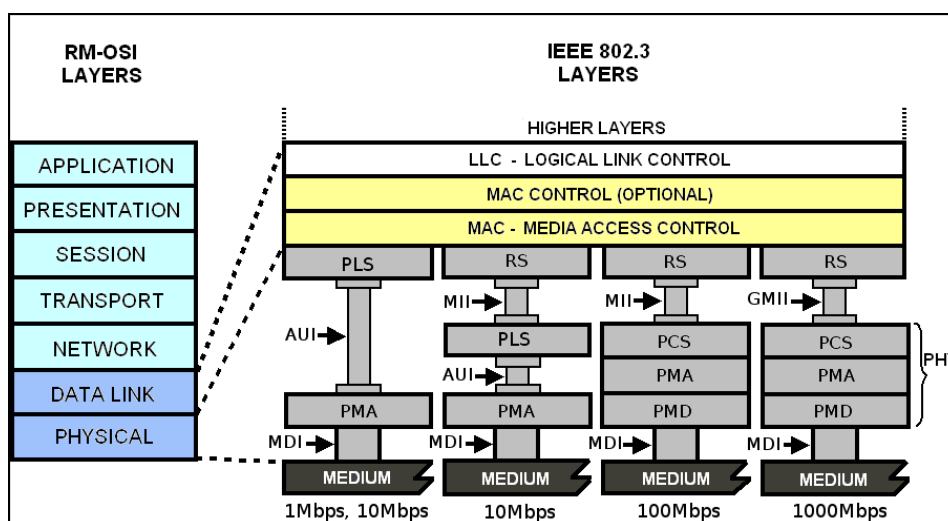


Figura 2.1: Relação entre a arquitetura IEEE 802.3 e o RM-OSI (IEEE, 2005a)

1. No presente trabalho os termos “camada” e “sub-camada” são equivalentes aos termos “nível” e “sub-nível”. Da mesma forma acontece com os termos “padrão” e “norma”

A sub-camada MAC refere-se ao protocolo CSMA/CD (*Carrier Sense Multiple Access / Collision Detection*), encarregado por determinar qual dos dispositivos em um ambiente de meios físicos compartilhados, tem permissão para transmitir dados (transmissão *semi-duplex*). Além disso, a sub-camada MAC permite a transmissão de dados em ligação ponto-a-ponto (*duplex*) e opcionalmente, é possível suportar mecanismos de controle de fluxo (*MAC Control*) (SPURGEON *et al.*, 2002).

A camada física da arquitetura IEEE 802.3 é representada por diferentes sub-camadas e interfaces (segundo a taxa de transmissão e o tipo de meio físico escolhidos), e é responsável pela definição das especificações elétricas, mecânicas e procedimentos para uma correta transmissão e recepção de bits através de um enlace de comunicação.

### 2.2.1 O Quadro Ethernet segundo a norma IEEE 802.3

Os termos *pacote* e *quadro*, assim como suas respectivas siglas, serão aqui definidos procurando manter coerência com a norma IEEE 802.3 (IEEE, 2005a), pois estes termos serão utilizados constantemente no texto. Conforme mostrado na figura 2.2, o termo *pacote* (em inglês, *packet*) refere-se a o fluxo de bits que trafega através do meio físico e o termo *quadro* (em inglês, *frame*), é a porção do pacote que carrega a mensagem (dados) que será transferida de ou para a camada superior à sub-camada MAC.

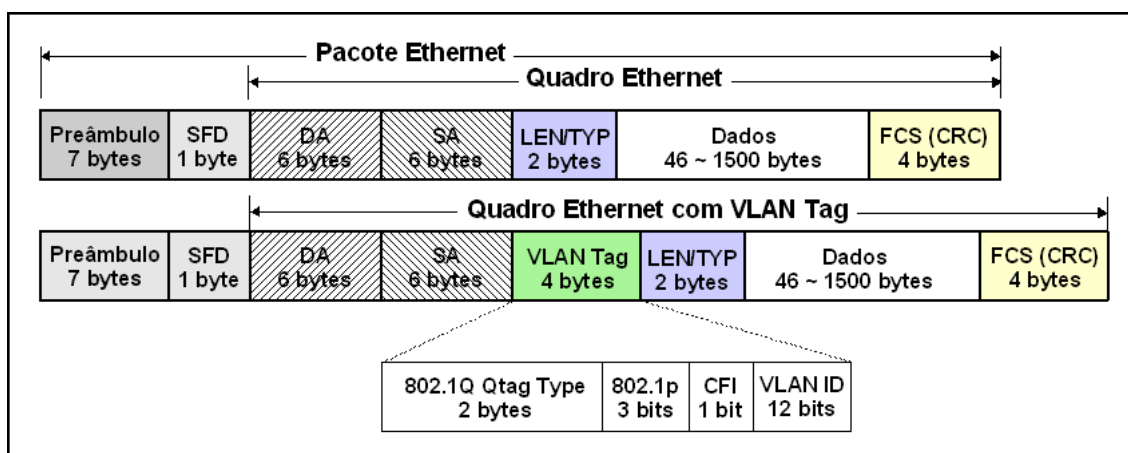


Figura 2.2: Estrutura dos quadros 802.3 (básico e *tagged*), baseado em (IEEE, 2005a)

A seguir é apresentada uma breve descrição dos campos do quadro Ethernet padrão, representados na figura 2.2:

- **Preâmbulo:** É um padrão alternado de 1's e 0's, utilizado para efeitos de sincronização de quadro;
- **Start Frame Delimiter (SFD):** Octeto responsável pela sinalização de início de quadro;
- **VLAN Tag:** Opcional. É especificado na norma IEEE 802.1q. O identificador VLAN (*Virtual LAN*) 12-bits permite a construção de um máximo de 4096 VLANs. Um identificador de protocolo é utilizado para detectar o “tag” no quadro, e é representado pelo valor hexadecimal 0x8100. O campo de *User Priority* é útil para definir um nível de prioridade no

quadro Ethernet. O campo CFI (*Canonical Format Indicator*) permite identificar a presença de um campo de informação de *router*.

- **Endereço Destino (DA):** Especifica o endereço do dispositivo que será o destino do quadro, ou o endereço que representa um grupo de dispositivos de destino (*multicast* ou *broadcast*);
- **Endereço Origem (SA):** Especifica o endereço físico MAC do dispositivo que enviou o quadro;
- **Tamanho do Quadro (LEN/TYP):** Indica o valor do tamanho do quadro Ethernet. Se o valor representado neste campo for maior ou igual que 0x600 (hexadecimal), então o campo indica o tipo de quadro;
- **Dados:** Possui um tamanho mínimo de 46 bytes e máximo de 1500 bytes. Contem os dados a enviar através da rede. Se a quantidade de dados a enviar é menor que 46 bytes, então um conjunto de bits de enchimento (denominados em inglês *padding bits* ou PAD) é acrescentado ao final dos dados para completar o tamanho mínimo do quadro;
- **Sequencia de Verificação de Quadro:** O *Frame Check Sequence* (FCS) é o valor resultante do cálculo CRC-32 (*32bit-Cyclic Redundancy Check*), executado sobre o conteúdo dos campos de endereço destino (DA), endereço origem (SA), tamanho/tipo do quadro (LEN/TYP), e os dados (incluído PAD, se for gerado). O FCS possui 4 bytes e é utilizado para verificação de erros no quadro recebido.

Um quadro Ethernet é inválido se cumpre pelo menos uma das seguintes condições:

- O tamanho do quadro é um número não inteiro de octetos.
- Os bits de um quadro que está sendo recebido (a exceção do seu campo de FCS) não geram um valor de CRC idêntico ao recebido.
- O tamanho do quadro é menor que o tamanho mínimo padrão (64 bytes) ou maior que o tamanho máximo padrão (1522 bytes, no caso de um quadro VLAN).

### 2.2.2 O Protocolo de Acesso ao Meio: CSMA/CD

“Esse protocolo funciona de modo semelhante a um jantar numa sala escura, onde os participantes só conseguem ouvir uns aos outros. Qualquer um à mesa precisa ouvir um período de silêncio antes de falar (*Carrier Sense*). Quando ocorre um espaço, todos têm a mesma chance de dizer algo (*Multiple Access*). Se duas pessoas começam a falar ao mesmo tempo, elas detectam o fato e param de falar (*Collision Detection*)” (SPURGEON, 2002).

Traduzindo isso para os termos da tecnologia Ethernet, o protocolo CSMA/CD arbitra o acesso de várias estações interligadas em um ambiente de meios físicos compartilhados, permitindo assim, a comunicação no modo de operação *semi-duplex*. O protocolo encontra-se especificado na cláusula 4 da norma IEEE 802.3 (IEEE, 2005a) e pode ser dividido em dois processos:

- *Carrier Sense:* Verifica se cada estação precisa aguardar até que não haja sinal no meio físico antes de transmitir. Se outra estação está transmitindo, há um

sinal no canal, o que define a condição conhecida como *Carrier*. Todas as outras estações devem aguardar até que o sinal cesse e o meio volte a ficar ocioso antes de tentar transmitir.

- *Multiple Access*: Todas as estações têm a mesma prioridade ao transmitirem um quadro na rede e todas elas podem tentar acessar o canal ao mesmo tempo. Caso duas ou mais estações estejam transmitindo simultaneamente, elas detectam a colisão dos sinais (*collision detection*). Cada uma delas escolhe um tempo aleatório de retransmissão antes de fazer uma nova tentativa, em um processo denominado *Back-off*. A figura 2.3 mostra o diagrama de fluxo simplificado do protocolo CSMA/CD.

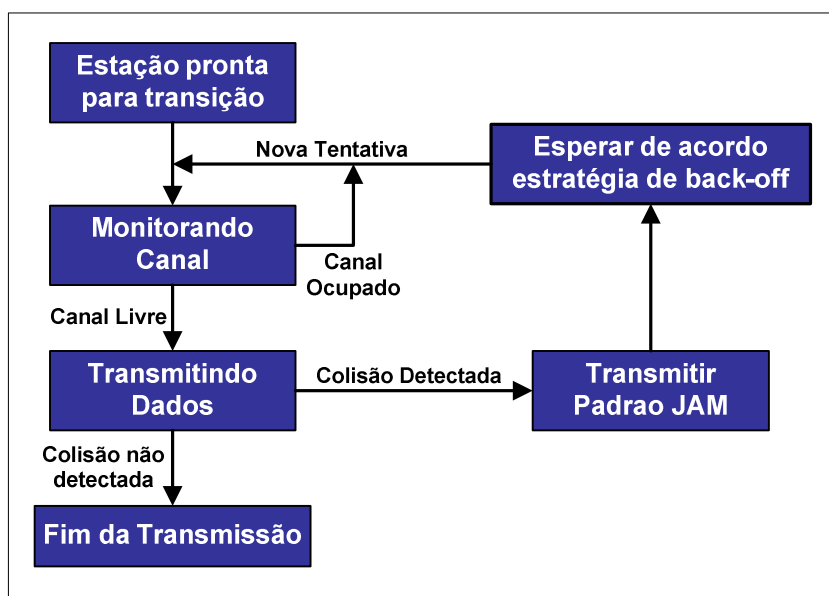


Figura 2.3: Diagrama de Fluxo simplificado do protocolo CSMA/CD

### 2.2.3 Enlace Duplex e Mecanismo de Controle de Fluxo

Ambas características foram introduzidas no suplemento IEEE 802.3x, no ano 1997. O modo de operação *duplex* desativa o protocolo CSMA/CD e permite a comunicação simultânea entre dois dispositivos. Este modo é amplamente utilizado pelos comutadores de rede (*switches*), os quais são os elementos básicos para construção de redes de interconexão cabeadas de alto desempenho, tais como as redes *Fast* e *Gigabit Ethernet* (SPURGEON *et al.*, 2002).

No suplemento é incluído um mecanismo opcional para o controle de fluxo sobre enlaces *duplex* (conhecido como *MAC Control*), sendo especificado na cláusula 31 da norma IEEE 802.3 (IEEE, 2005a). Sua função principal é bloquear temporariamente a transmissão de um dispositivo que está fornecendo quadros em excesso (além da capacidade do *buffer* do dispositivo receptor). Isto é realizado através da geração e detecção do quadro especial de *pause*, representado na figura 2.4.

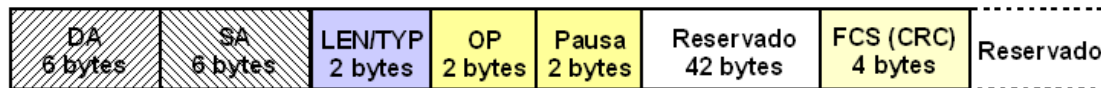


Figura 2.4: Estrutura do quadro especial de *pause*

O processo de detecção de quadros de *pause* pode ser resumido da seguinte maneira:

“Quando um quadro livre de erros é recebido pela sub-camada MAC, este faz uma verificação nos campos de DA e LEN/TYP, assim como também, nos dois primeiros bytes do campo de dados. Um quadro especial de *pause* carrega o endereço *multicast* de controle 0x0108020001h no campo de DA e o identificador especial de controle 0x8808h no campo de LEN/TYP. O *opcode* especial de controle 0x0001 é transportado nos dois primeiros bytes do campo de dados. Se todos os campos são válidos então é extraído o “Tempo de Pausa” que ocupa os dois bytes seguintes ao *opcode* especial de controle. Esse valor indica o tempo de bloqueio do transmissor (em unidades de tempo de *slot*)” (SPURGEON, 2002).

#### 2.2.4 Interface Independente do Meio e o Gerenciamento da Camada Física

A ligação entre a sub-camada MAC e a camada física nos sistemas 10/100 Mbits/s é realizada a través da sub-camada de Reconciliação (em inglês, *Reconciliation Sublayer* ou RS) e da Interface Independente do Meio (em inglês, *Media Independent Interface* ou MII), ambas especificadas na cláusula 22 da norma IEEE 802.3 (IEEE, 2005a). A sub-camada de Reconciliação mapeia os sinais da MII sobre as primitivas PLS (*Physical Layer Signaling*), para fornecer serviços de envio/recepção de bits, serviços de detecção de colisões e presença de portadora, assim como também detecção e propagação de erros no canal.

A Interface Independente do Meio permite que qualquer um dos meios físicos padronizados possa se comunicar facilmente com a sub-camada MAC, com taxas de transferência de dados de 10 Mbits/s ou 100 Mbits/s. Essa interface também possui um mecanismo para o Gerenciamento de dispositivos da camada física (em inglês, *Physical Layer Devices*) ou também conhecidos como PHYs. O dispositivo que realiza a gerência do PHY é conhecido como *Station Management Entity* (STA). A tabela 2.1 mostra a descrição de cada um dos sinais que compõem a MII (IEEE, 2005a)

Tabela 2.1: Sinais que compõem a Interface Independente do Meio

Sinal	Responsável pela geração	Descrição
RX_CLK	PHY	Relógio para recepção dos dados com uma frequência nominal de 2.5 MHz (10 Mbits/s) e 25 MHz (100 Mbits/s)
RX_DV	PHY	Indica que um dado válido foi colocado no canal RXD.
RX_ER	PHY	Indica os erros no canal de recepção.
RXD[3:0]	PHY	Canal que recebe os dados em formato de <i>nibbles</i> .
TX_CLK	PHY	Relógio para transmissão dos dados com uma frequência

Sinal	Responsável pela geração	Descrição
		nominal de 2.5 MHz (10 Mbits/s) e 25 MHz (100 Mbits/s)
TX_EN	RS	Habilita a transmissão dos dados.
TX_ER	RS	Indica a propagação de um erro no canal de transmissão.
TXD[3:0]	RS	Canal que transmite os dados em formato de <i>nibbles</i> .
CRS	PHY	<i>Carrier Sense</i> , indica a ocupação do meio compartilhado.
COL	PHY	Indica um evento de colisão no meio compartilhado.
MDC	STA	Sinal de relógio de 2.5MHz para o gerenciamento do PHY.
MDIO	STA	Transfere as informações de controle e estado do PHY.

#### 2.2.4.1 Gerenciamento dos Dispositivos da Camada Física

É utilizado um mecanismo independente capaz de configurar um ou vários PHY's desde um único STA. Esse mecanismo também é conhecido como *Media Independent Interface Management (MIIM)*. A comunicação com o dispositivo PHY é feita através de dois canais seriais: um canal bidirecional chamado *Management Data Input/Output (MDIO)* e um sinal de relógio de 2.5MHz chamado *Management Data Clock (MDC)*, como é mostrado na figura 2.5. Esta interface serial implementa um protocolo de comunicação capaz de acessar até 32 PHYs, cada um contendo 32 registradores de 16 bits. Desta forma, é possível levar informações de estado, erros e capacidades do PHY para as camadas superiores.

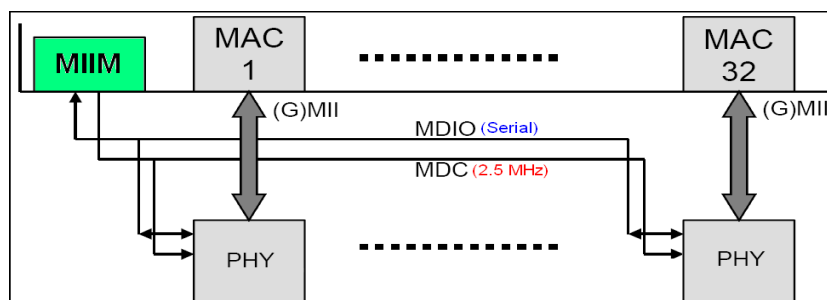


Figura 2.5: Gerenciamento dos Dispositivos da Camada Física

O protocolo de comunicação entre o STA e o PHY, conhecido como “Protocolo MDIO”, é baseado na codificação dos dados transferidos pelo MDIO. Na tabela 2.2 se mostra a codificação do canal bidirecional MDIO.



Tabela 2.2: Codificação do protocolo de Comunicação MDIO

Nome	Valor	Tamanho	Comentários
<i>Start</i>	01b	2 bits	Constante.
<i>Opcode</i> (Op)	01b – escrita 10b – leitura	2 bits	Indica se é leitura ou escrita.
Endereço PHY	XXXXXXb	5 bits	Suporta até 32 PHYs na rede.
Endereço registrador	XXXXXXb	5 bits	Cada PHY pode ter até 32 registradores.
<i>Turn-around</i> (TA)	10b – escrita Z0b – leitura	2 bits	Constante.
Dado	Dado	16 bits	Valores de 16 bits.

A figura 2.6 representa o diagrama temporal de uma operação de leitura típica dos registradores do PHY. O campo “*Start*” permite que o canal MDIO fique fora do estado ocioso (*idle*). O campo de TA é definido como dois *idle* bits de tempo inseridos entre o campo “Endereço registrador” e o campo “Dado”. Numa operação de leitura, durante o primeiro bit do TA, nenhum dispositivo deve ficar ativo (alta impedância), e no segundo bit do TA, o PHY deve colocar um zero no canal MDIO.

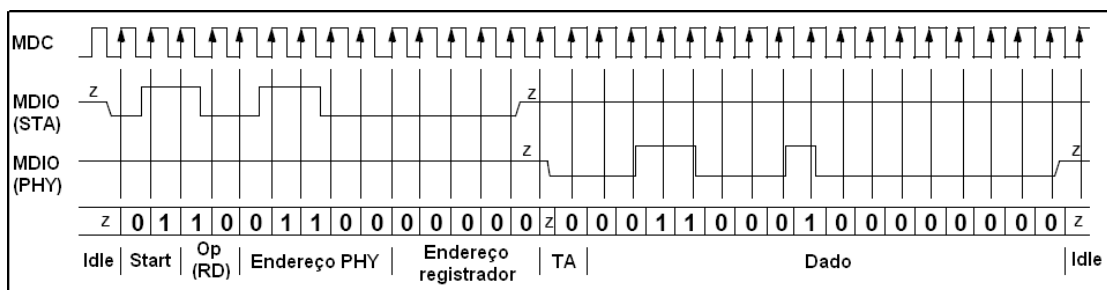


Figura 2.6: Operação de leitura típica no gerenciamento do PHY

Por sua parte, a operação de escrita dos registradores do PHY, é realizada eliminando o efeito do TA (fixado-o em “10”). A figura 2.7 mostra o diagrama temporal de uma típica operação de escrita.

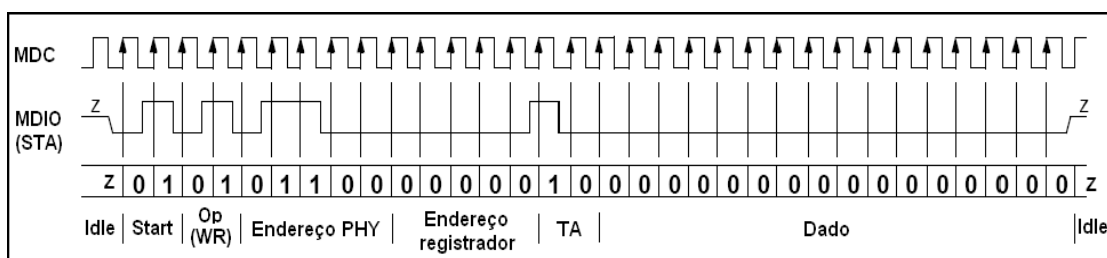


Figura 2.7: Operação de escrita típica no gerenciamento do MII

### 2.2.5 A Terceira Geração: Gigabit Ethernet

A tecnologia Gigabit Ethernet é a terceira geração da Ethernet, capaz de suportar uma taxa de transmissão de 1000 Mbits/s mantendo compatibilidade com o sistema Ethernet original. Essa extensão arquitetural foi definida pelas normas IEEE 802.3z e IEEE 802.3ab, nos anos 1998 e 1999 respectivamente. O modo de operação *semi-duplex*, utiliza o método de acesso CSMA/CD original com duas importantes modificações (FRAZIER, 1999):

- Extensão da portadora (*carrier extension*): Uma das exigências do próprio protocolo CSMA/CD é manter o domínio de colisão em 200 metros, mesmo em taxas de transmissão de *gigabits* por segundo. Em resposta a esse dilema, o padrão implementa um mecanismo que estende o tempo ocupado pela portadora do sinal associada a uma transmissão do quadro mínimo, sem modificar os campos do quadro Ethernet original e mantendo o requisito de tamanho mínimo do quadro (64 bytes). Para isto, o tempo de *slot* teve que ser aumentado de 512 tempos de bit (utilizado nos sistemas 10 e 100 Mbits/s), para 4,096 tempos de bit. Assim, na transmissão *semi-duplex* de quadros Gigabit Ethernet menores que 512 bytes são necessários bits de extensão, o que reduz significativamente a eficiência do canal. A figura 2.8 mostra a estrutura do quadro Ethernet com extensão de portadora.

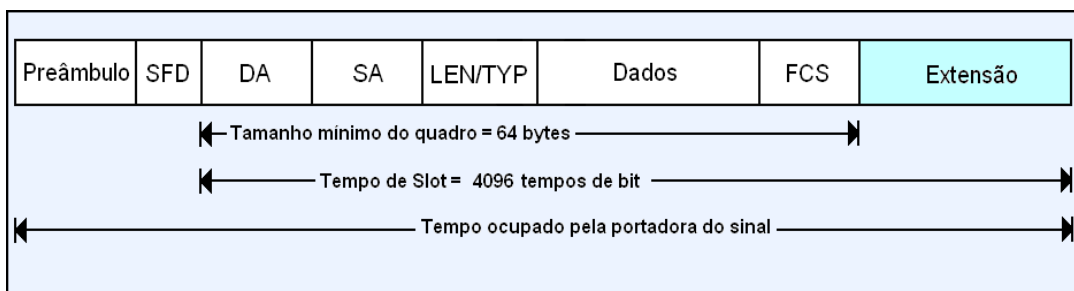


Figura 2.8: Estrutura do quadro Ethernet com extensão de portadora

- Rajada de quadros (*frame bursting*): É um recurso opcional para melhorar o desempenho na transmissão de quadros pequenos sobre canais *semi-duplex*. Isso permite que uma estação envie mais de um quadro durante um determinado evento de transmissão. A rajada de quadros é iniciada depois da transmissão de um quadro normal (incluindo quaisquer bits de extensão) e pode continuar até que o limite de rajada de quadros (em inglês, *Frame Burst Limit* ou FBL) seja alcançado. Para evitar colisões durante a rajada, a estação deve manter o canal ativo inserindo símbolos de extensão durante as lacunas entre quadros (em inglês, *Inter-Frame Gap* ou IFG). A figura 2.9 mostra uma representação da transmissão de rajada de quadros.

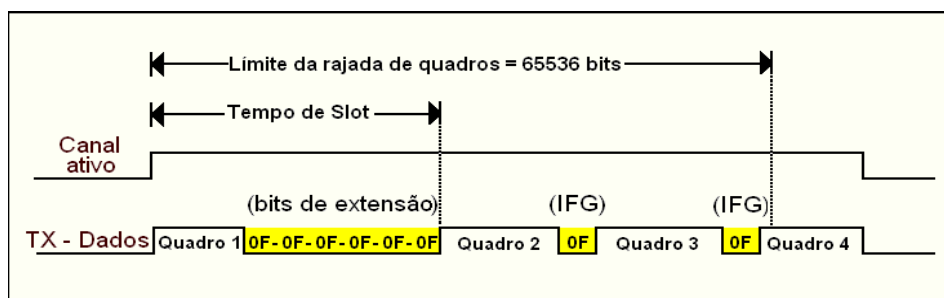


Figura 2.9: Transmissão de rajada de quadros

No modo de operação *duplex*, a extensão de portadora e o recurso de rajada de quadros não são utilizados, já que o protocolo CSMA/CD está desabilitado. Em compensação, este modo de operação oferece uma taxa de transmissão de quadro mínimo dez vezes maior que no sistema Fast Ethernet *duplex*, segundo a cláusula 34 da norma IEEE 802.3 (IEEE, 2005a). Estas e outras vantagens provocaram que a maioria de dispositivos *Gigabit Ethernet* produzidos até o hoje utilizem preferencialmente o modo de operação *duplex* (FRAZIER, 1999).

A figura 2.1 mostra a ligação entre a sub-camada MAC e o nível físico da tecnologia Gigabit Ethernet. Esta ligação é realizada através da sub-camada de Reconciliação (RS) e da Interface Gigabit Independente do Meio (em inglês, *Gigabit Media Independent Interface* ou GMII), ambas especificadas na cláusula 35 da norma IEEE 802.3 (IEEE, 2005a). GMII é uma extensão da MII, com a diferença de que a transferência dos dados (transmissão e recepção) é realizada sobre dois canais síncronos independentes de 8 bits com uma frequência de operação de 125 MHz. A tabela 2.3 mostra a codificação válida dos sinais da GMII na transmissão de quadros (TXD, TX\_EN e TX\_ER). Da mesma forma, a tabela 2.4 mostra a codificação válida dos sinais da GMII na recepção de quadros (RXD, RX\_DV e RX\_ER). A gerência dos dispositivos PHY Gigabit é realizada através do mecanismo STA, como descrito na subseção 2.2.4.1

Tabela 2.3: Codificação GMII para transmissão de quadros, baseado em (IEEE, 2005a)

TX_EN	TX_ER	TXD[7:0]	Descrição
0	0	0x00 até 0xFF	Lacuna entre quadros (IFG)
0	1	0x00 até 0x0E	Reservado.
0	1	0x0F	Extensão de portadora.
0	1	0x10 até 0x1E	Reservado.
0	1	0x1F	Erro durante a extensão de portadora.
0	1	0x20 até 0xFF	Reservado.
1	0	0x00 até 0xFF	Transmissão normal de dados.
1	1	0x00 até 0xFF	Propagação de erro na transmissão.

Tabela 2.4: Codificação GMII para recepção de quadros, baseado em (IEEE, 2005a)

RX_DV	RX_ER	RXD[7:0]	Descrição
0	0	0x00 até 0xFF	Lacuna entre quadro (IFG)
0	1	0x00	Lacuna entre quadros (IFG)
0	1	0x01 até 0x0D	Reservado.
0	1	0x0E	Indicação de <i>False Carrier</i> .
0	1	0x0F	Extensão de portadora.
0	1	0x10 até 0x1E	Reservado.
0	1	0x1F	Erro durante a extensão de portadora.
0	1	0x20 até 0xFF	Reservado.
1	0	0x00 até 0xFF	Recepção normal de dados.
1	1	0x00 até 0xFF	Erro na recepção dos dados.

O nível físico do *Gigabit Ethernet* suporta dois tipos de sistemas de mídia:

- O sistema de mídia par trançado 1000BASE-T: As especificações do sistema 1000BASE-T são descritas na cláusula 40 da norma IEEE 802.3 (IEEE, 2005a). Se caracteriza pelo uso de cabeamento de par trançado não blindado categoria 5 (UTP-CAT 5).
- O sistema de mídia fibra ótica 1000BASE-X: As especificações do sistema 1000BASE-X são descritas nas cláusulas 36-39 da norma IEEE 802.3 (IEEE, 2005a). Este sistema foi adaptado a partir do padrão ANSI X3.230-1994 Fiber Channel (FC-0 e FC-1) e suporta três formas de implementação física (1000BASE-SX, 1000BASE-LX, e 1000BASE-CX).

Geralmente, os sistemas de mídia mencionados acima são encapsulados num único dispositivo PHY, como mostra a figura 2.1. A seguir serão descritas as principais características das três sub-camadas que compõem o PHY no sistema de mídia 1000BASE-X:

a) PCS (*Physical Coding Sublayer*): Esta sub-camada é responsável pela geração dos sinais de controle da GMII (*Carrier Sense* e *Collision Detect*) e pela codificação/decodificação 8B/10B dos octetos transferidos pela GMII em *code-groups* de 10 bits. Também suporta o mecanismo de *Auto-negotiation*, o qual informa as camadas superiores se o enlace está pronto para transferir pacotes.

b) PMA (*Physical Medium Attachment*) Sublayer: É o encarregado da serialização dos códigos vindos do PCS e da deserialização dos bits vindos do meio físico. Também é responsável pela recuperação do relógio e pelo alinhamento dos dados recebidos.

c) PMD (*Physical Medium Dependent*) *Sublayer*: Esta sub-camada define a sinalização utilizada no nível físico e suporta vários tipos de conectores e mídias, tais como: fibra ótica mono-modo ou fibra ótica multi-modo.

### 2.2.6 Parâmetros de Implementação da Ethernet

Na tabela 2.5 são listados os valores limites dos parâmetros definidos para implementação da tecnologia Ethernet, segundo a taxa de transmissão:

Tabela 2.5: Valores limites para implementação da tecnologia Ethernet, baseado em (IEEE, 2005a)

Parâmetros	Valores		
	10/100Mbps/s	1Gbits/s	10Gbits/s
Tempo de <i>Slot</i>	512 (tempos de bit)	4096 (tempos de bit)	Não utilizado
Lacuna entre quadros	96 bits	96 bits	96 bits
Número máximo de Retransmissões	16	16	Não utilizado
Limite do <i>Back-off</i>	10	10	Não utilizado
Tamanho máximo do quadro sem <i>tag</i>	1518 octetos	1518 octetos	1518 octetos
Tamanho mínimo do quadro	64 octetos	64 octetos	64 octetos
Limite de Rajada de quadros	Não utilizado	65536 bits	Não utilizado

## 2.3 OAM

OAM foi desenvolvido originalmente pelas operadoras de rede para incrementar o nível de confiabilidade das redes TDM (*Time Division Multiplexing*). Como resultado, os sistemas TDM alcançaram o famoso “cinco noves” de disponibilidade e confiabilidade (99,99999%) e um tempo de recuperação de até 50 milissegundos na presença de falhas, permitindo assim reduzir satisfatoriamente os custos de manutenção das redes (STEIN, 2006). Posteriormente, os mecanismos de OAM tiveram sua expansão nas redes de comutação de pacotes, sendo considerada recentemente dentro da norma IEEE 802.3 na sua versão do ano 2005.

OAM refere-se ao conjunto de ferramentas necessárias para monitorar, detectar e isolar falhas numa rede. As funcionalidades do OAM permitem com que as operadoras de rede trabalhem eficientemente, sendo possível oferecer SLAs confiáveis e conseqüentemente uma comunicação fim-a-fim segura. As redes sem OAM demandam

um alto custo operacional e maior tempo para resolução de falhas, devido à limitada visibilidade da rede por parte das operadoras; o que obriga a contínua intervenção de técnicos em campo (MEF, 2005a)

Até um tempo atrás Ethernet não suportava funcionalidades de OAM, já que nas redes locais era possível corrigir defeitos utilizando ferramentas de uso simples tais como o *Ping* ou *Traceroute*. Com a introdução das redes MENs e com o atual desenvolvimento das redes *Carrier-Class* Ethernet, esta situação tem mudado radicalmente. Os provedores de serviços baseados na tecnologia Ethernet precisam ter controle absoluto das suas redes, portanto é essencial que as MENs suportem detecção automática de falhas, assim como também condições favoráveis para medição do desempenho da rede. Por tudo isso é que nasceu a necessidade de incluir os mecanismos de OAM na tecnologia Ethernet, mantendo compatibilidade com os dispositivos Ethernet já existentes (MEF, 2005a).

Dois protocolos OAM para Ethernet surgiram recentemente. Um deles, chamado aqui de “OAM da camada de enlace” foi desenvolvido para aplicações na primeira milha (EFM, *Ethernet in the First Mile*), e se caracteriza por monitorar continuamente um enlace único (MEF, 2005b). O responsável pelo seu desenvolvimento foi o grupo de trabalho IEEE 802.3ah, e os resultados do trabalho estão disponíveis na cláusula 57 da norma IEEE 802.3 (IEEE, 2005a). O outro protocolo OAM, aqui chamado de “OAM da camada de serviços”, permite uma comunicação fim-a-fim e possibilidades de medida do desempenho na rede. Na IEEE, este tipo de OAM foi designado com o code-nome de IEEE 802.1ag - CFM (*Connectivity Fault Management*) e na ITU-T é conhecido com o nome de Y.1731 (*Performance Monitoring*). Um terceiro nível do OAM está sendo desenvolvido pelo MEF, e está sendo orientado à gerencia das outras camadas de OAM.

### 2.3.1 EFM– IEEE 802.3ah

É a encarregada de proporcionar o conjunto de funcionalidades “OAM da camada de enlace” para a tecnologia Ethernet. Este tipo de OAM foi desenvolvido pelo grupo de estudo IEEE 802.3ah e os resultados foram publicados na cláusula 57 da norma IEEE 802.3 (IEEE, 2005a). A expressão "primeira milha" refere-se à primeira área de acesso onde é estabelecida a comunicação entre o usuário final e os equipamentos de borda das operadoras de rede. OAM provê mecanismos de monitoração e detecção de falhas na rede, o que permite:

- Garantir que as operadoras de rede cumpram com a qualidade de serviço oferecida aos seus clientes.
- Detectar anomalias antes que elas escalem a outros domínios da rede. A operação das redes sem OAM demanda mais recursos e torna-se mais custosa, devido à contínua intervenção manual para a detecção das falhas na rede.

O EFM suporta três diferentes topologias e camadas físicas. As três topologias são complementares e cada uma apresenta-se útil para atender determinado ambiente de rede (MEF, 2005b):

- *Ethernet in the First Mile Copper* (EFMC): Conexão ponto-a-ponto sobre o par metálico utilizado para telefonia. Prevê uma banda de acesso de até 10Mbits/s para distâncias de até 750 metros ou 2Mbits/s para uma distancia máxima de 2.7Km.

- *Ethernet in the First Mile Fibber* (EFMF): Conexão ponto-a-ponto utilizando fibra ótica mono-modo. Esta topologia prevê uma velocidade de até 100Mbps/s para distâncias de até 10Km.
- *Ethernet in the First Mile Passive Optical Network* (EFMP): Conexão ponto-multiponto utilizando *splitters* para dividir o sinal óptico de entrada em várias saídas. Opera sobre fibras mono-modo com velocidades de 1Gbits/s em distâncias de até 20Km.

### 2.3.1.1 Funções da Sub camada OAM Ethernet

As principais funções do protocolo OAM da camada de enlace são (IEEE, 2005a):

1. Monitoramento do desempenho no enlace.
2. *OAM Discovery*: É a primeira fase do protocolo OAM e permite determinar se os dispositivos interligados suportam OAM. Se o processo de descoberta for satisfatório, então é efetuada a troca de informações locais (configuração e estado) para habilitar a comunicação OAM no enlace.
3. *Link monitoring*: Define os atributos e informações de estado úteis para detecção e indicação de falhas no enlace Ethernet. Estas informações são enviadas ao dispositivo remoto utilizando OAMPDUs (*OAM Protocol Data Units*) (descrito no item 2.3.1.3).
4. *Remote Fault Detection*: Há uma *flag* no OAMPDU que permite enviar condição de falha ao dispositivo remoto:
  - Falha no enlace: A perda do sinal é detectado pelo receptor, e é enviada uma mensagem por segundo através de *Information OAMPDUs*.
  - *Dying Gasp*: Condição sem recuperação, devido a uma falha na alimentação do equipamento. Envia um determinado número de mensagens até o equipamento desligar completamente (normalmente após alguns milisegundos).
5. *Remote Loopback*: Teste de *loopback* no enlace (tipo intrusivo). Esta função obriga ao equipamento ficar fora de serviço.
6. *MIB variable retrieval*: Fornece informações desde uma base de dados para gerenciamento (MIB, *Management Information Base*)
7. *Organization specific enhancements*: Opções para customizar o protocolo OAM, com o objetivo de criar características proprietárias.
8. *Timers*: O protocolo é controlado por dois *timers*, um que controla a frequência com que estão sendo enviados os OAMPDUs (mínimo um pacote por segundo), e outro que controla a frequência com que devem ser recebidos os OAMPDUs (mínimo um pacote cada cinco segundos) para manter ativo o enlace entre os dispositivos.
9. *Modos de operação*: Ativo ou passivo. No modo ativo, o equipamento força a comunicação utilizando o protocolo OAM, enviando requisições e comandos ao dispositivo remoto. No modo passivo, o equipamento espera que o dispositivo remoto inicie o protocolo OAM e somente esta em condições de responder as mensagens recebidas.

### 2.3.1.2 Pacotes OAM Ethernet

As operações do OAM não afetam o tráfego de dados, já que é utilizado um tipo de *slow protocol* que utiliza uma largura de banda mínima. As mensagens utilizadas para implementação do protocolo OAM são chamadas de OAMPDUs, estas são interceptadas pela sub-camada MAC e logo transferidas desde/para um processo “Cliente OAM”, evitando assim serem propagados através de múltiplos *hops* na rede.

Os pacotes OAM Ethernet apresentam restrições no número de pacotes que podem ser enviados por segundo (para OAMPDUs, não mais do que 10), facilitando assim a implementação do cliente OAM em software.

Os pacotes OAMPDUs contêm informações de controle e status necessários para monitorar, testar e detectar defeitos nos enlaces OAM. Essas informações são codificadas usando campos específicos em um formato de pacote padrão, que pode ser visualizado na figura 2.10.

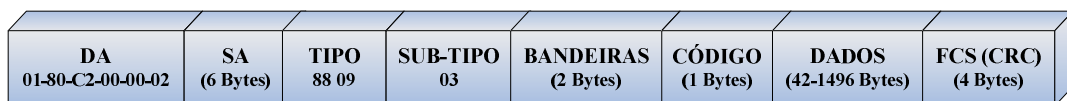


Figura 2.10: O datagrama OAMPDU, baseado em (IEEE, 2005a)

Todos os pacotes do tipo *Slow Protocol* possuem no campo *Destination Address* (DA) o endereço *multicast* “01-80-C2-00-00-02”, assim como 0x8809 (hexadecimal) no campo *Type*. OAMPDUs diferenciam-se pelo campo *Sub-Type* igual a 0x03 (hexadecimal).

### 2.3.1.3 Blocos da Sub camada OAM Ethernet

OAM é apresentado como uma subcamada opcional dentro da camada de enlace de dados da arquitetura de rede IEEE 802.3, entre as sub-camadas MAC e LLC. Os componentes da sub-camada OAM Ethernet são apresentados na figura 2.11, e posteriormente descritos.

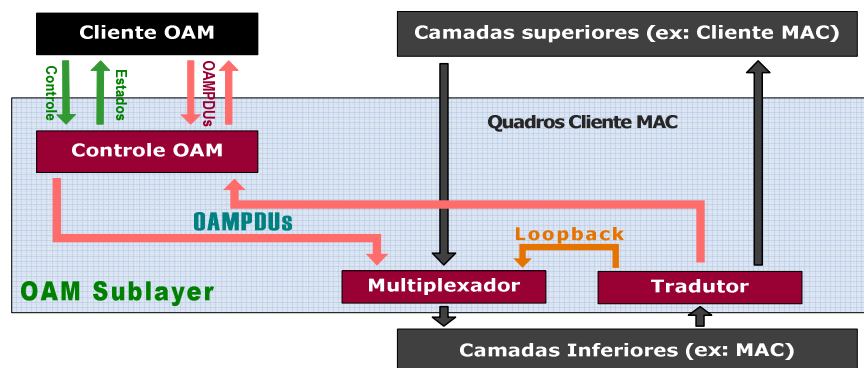


Figura 2.11: Diagrama de blocos da sub-camada OAM Ethernet, baseado em (IEEE, 2005a)



### 2.3.1.3.1 MAC Client

Fornecer dados ao MAC, para serem encapsulados e posteriormente enviado ao módulo multiplexador. O recebimento dos dados é realizado através do módulo Tradutor (*Parser*).

### 2.3.1.3.2 OAM Client.

Configura a sub-camada OAM através do bloco de Controle, processa os OAMPDUs recebidos e transmite OAMPDUs.

### 2.3.1.3.3 Bloco de Controle (Controle OAM)

Provê uma interface com a entidade OAM Client.

### 2.3.1.3.4 Tradutor (*Parser*)

Verifica os quadros recebidos e envia os OAMPDUs ao bloco de controle (e logo ao OAM Client). Em função das configurações, envia quadros do tipo *non-OAMPDUs* às camadas superiores ou ao módulo multiplexador. Esta entidade é definida na norma IEEE 802.3 através de máquina de estados exibidas respectivamente na figura 2.12.

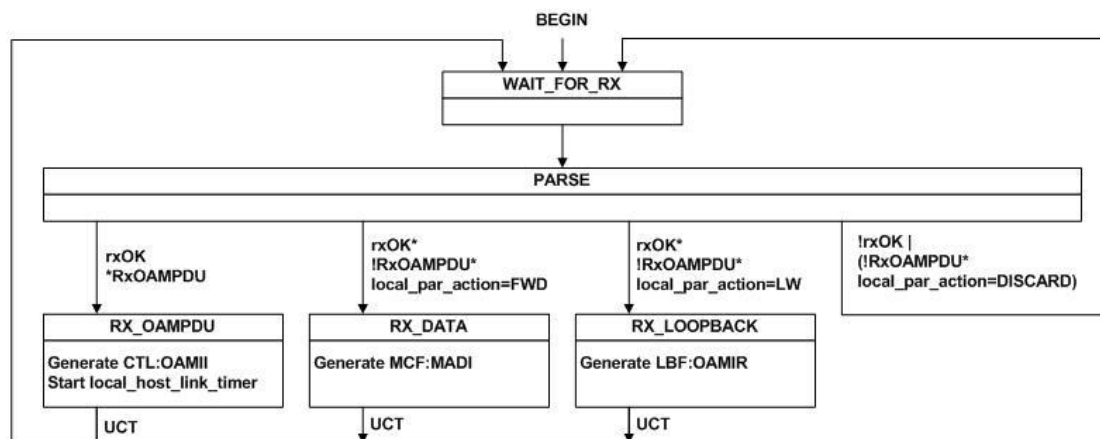


Figura 2.12: Máquina de estados da entidade Tradutor (IEEE, 2005a)

### 2.3.1.3.5 Multiplexador (*Multiplexer*)

Multiplexa OAMPDUs e *non-OAMPDUs*. Esta entidade é definida na norma IEEE 802.3 através da máquina de estados exibidas na figura 2.13.

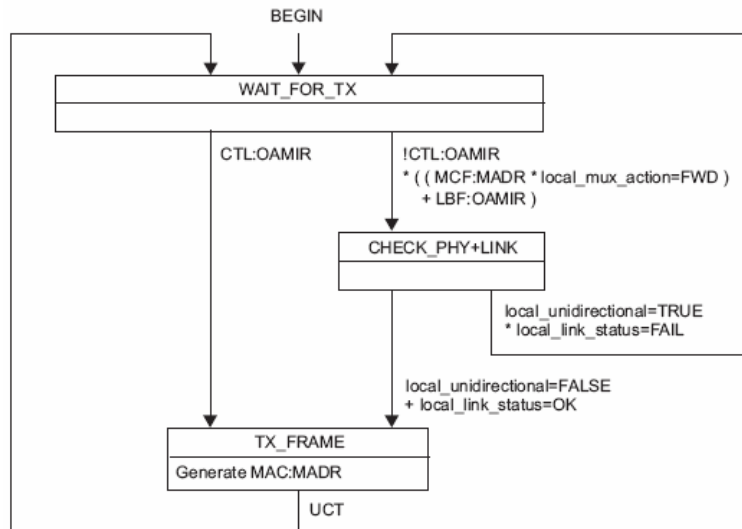


Figura 2.13: Máquina de estados da entidade Multiplexador (IEEE, 2005a)

### 2.3.2 CFM – IEEE 802.1ag

OAM da camada de serviços provê gerenciamento e visibilidade fim-a-fim para as conexões Ethernet sobre qualquer tipo de infraestrutura de transporte (tais como *Multi Protocol Label Switching* [MPLS], *Virtual Private LAN Services* [VPLS], e *provider bridges*). Também provê mecanismos para suportar o monitoramento de desempenho dos serviços. CFM (*Connectivity Fault Management*) está sendo desenvolvido pelo grupo de estudo IEEE 802.1ag (IEEE, 2007).

IEEE 802.1ag precisa que todos os dispositivos na rede suportem o protocolo, já que utiliza um endereço MAC reservado especial que se não é reconhecido pelo equipamento não será possível suportar o OAM da camada de serviços.

Os tipos de mensagens disponíveis no CFM são:

- *Continuity Check Messages*: Mensagem do tipo "heart-beat" transferida periodicamente entre os *Maintenance Endpoints* (MEs). Com este recurso é possível a descoberta de outros MEs dentro de um mesmo domínio de rede, e permite que as MIPs (*Maintenance Intermediate Points*) descubram MEs.
- *Traceroute Messages*: São transmitidas por uma ME, com a finalidade de descobrir o caminho (*hop-by-hop*) até um ME destino. É similar ao conceito de *UDP traceroute*.
- *Loopback Messages*: São transmitidos por MEs para verificar a conectividade de um ME particular. Testes de *loopback* indicam se o equipamento destino está operativo ou não. É similar ao conceito *ICMP Echo* (*Ping*)
- *AIS Messages*: São geradas por um ME ou MIP para descobrir uma falha de conexão, e conseqüentemente comunicar o evento a outros dispositivos.

### 3 O DISPOSITIVO DE DEMARCAÇÃO ETHERNET

O dispositivo de demarcação Ethernet (EDD) serve como entidade controladora dos serviços de borda, entre os clientes e os provedores de serviço. A figura 3.1 mostra a arquitetura de rede necessária para o controle dos serviços fim-a-fim, assim como também o monitoramento e diagnóstico das SLAs. Os EDDs podem fornecer as funções de segurança dos SLAs, *loopback* remoto, limitação da banda, *VLAN stacking*, *priority queuing*, gerenciamento do desempenho, etc (Bray, 2006).

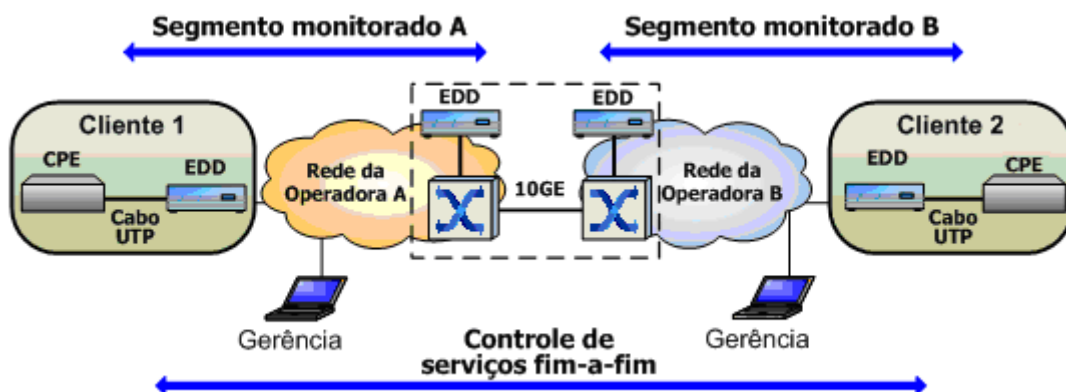


Figura 3.1: Cenário de atuação do EDD, (Bray, 2006)

#### 3.1 Funções essenciais do EDD

Há duas funções essenciais para a demarcação Ethernet (Bray, 2006):

1. **Ethernet Network Interface Device (NID):** Também conhecido como *Network Termination Equipment* (NTE) ou *Network Termination Unit* (NTU), isto envolve as camadas de Enlace/Transporte e conectividade OAM (IEEE 802.3ah e IEEE 802.1ag), e provê monitoramento remoto e testes.
2. **Ethernet Service User/Network Interface (UNI):** Define e administra a camada de serviços (MEF/ITU). A função UNI é necessária no dispositivo de demarcação, especialmente se o Ethernet é transportado sobre outros meios, tais como: SONET, DS1/DS3, e DSL por exemplo.

Os elementos anteriormente definidos são mostrados na figura 3.2, onde é mostrado o ponto de demarcação numa estrutura de rede básica. *Network-to-Network Interface* (NNI) é equivalente ao UNI, quando o EDD é utilizado dentro da rede da operadora.

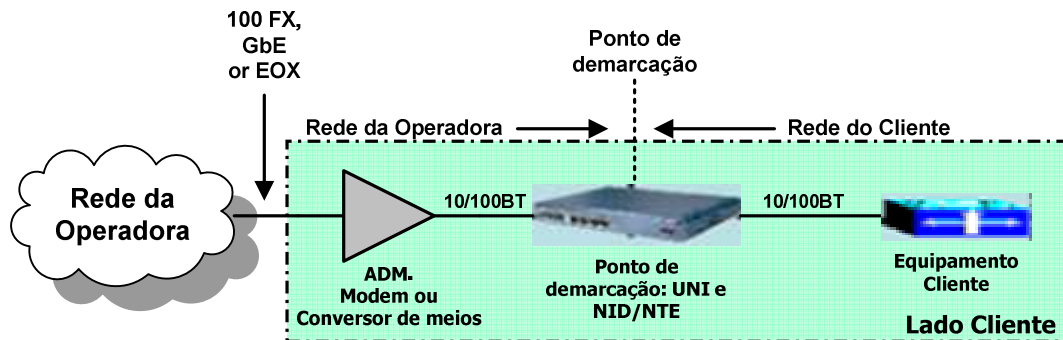


Figura 3.2: Funções do EDD, (Bray, 2006)

## 3.2 Tipos de demarcação

### 3.2.1 Demarcação UNI

O EDD é essencial para estabelecer uma interface entre a operadora e o cliente (geralmente referido como UNI), que tem a função de separar a rede WAN das operadoras e as redes LAN dos clientes. O EDD permite testar e monitorar os dois segmentos (LAN e WAN) para uma correta distribuição dos serviços Ethernet, e também ajuda no suporte das SLAs. Figura 3.3 mostra a posição da UNI e do EDD numa rede típica (Bray, 2006).

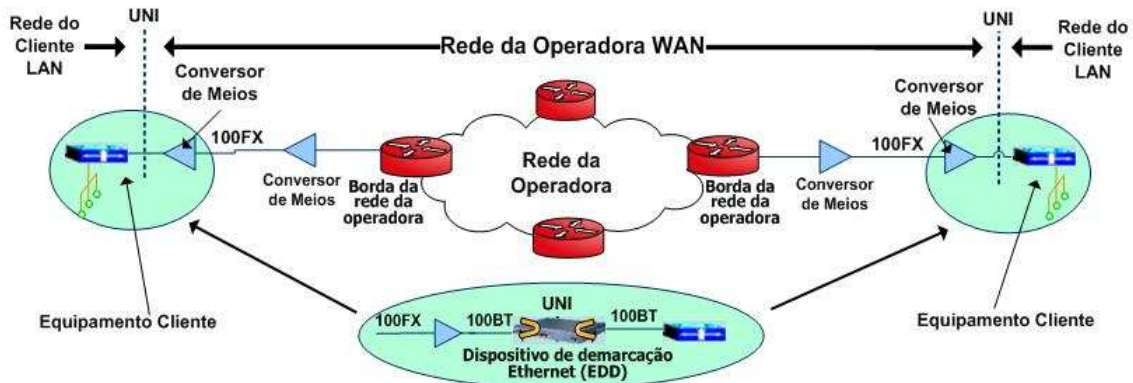


Figura 3.3: Cenário da demarcação UNI, (Bray, 2006)

### 3.2.2 Demarcação NNI

Os EDDs também podem ser utilizados entre diferentes operadoras para manter um nível apropriado de visibilidade fim-a-fim e para controlar os serviços que são fornecidos aos clientes localizados numa área fora do negociado (OOF – *out-of-franchise*) e dependentes de uma segunda operadora. Esse nível de demarcação NNI é crítico para assegurar que duas ou mais operadoras consigam trabalhar conjuntamente

para cumprir o SLA prometida pela primeira operadora ao seu cliente. A figura 3.3 mostra o conceito básico.

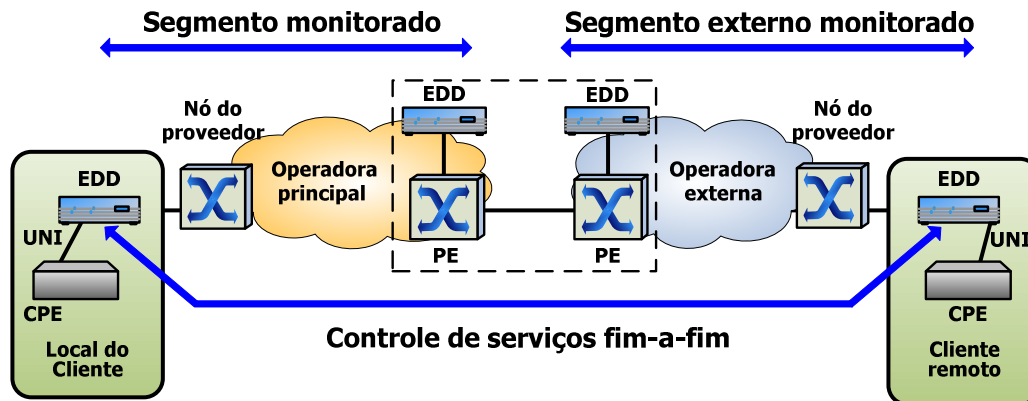


Figura 3.4: Cenário de demarcação NNI

### 3.3 Uso do OAM no EDD

Os seguintes itens descrevem alguns modos de uso das funções do OAM (Bray, 2006).

#### 3.3.1 Uso da função OAM *Discovery*

OAM *Discovery* é a função mais importante do OAM, e permite ao equipamento de borda do provedor de serviços determinar se o dispositivo de demarcação possui funções de OAM.. A figura 3.5 apresenta o modo de uso desta função.

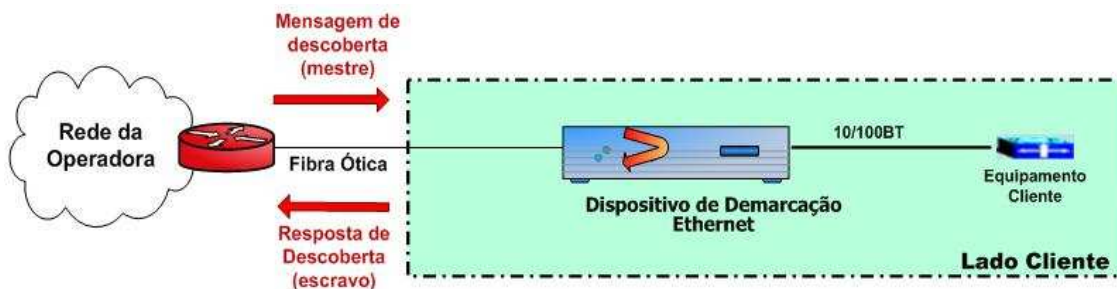


Figura 3.5: Função OAM *Discovery*, (Bray, 2006)

Os padrões OAM são relativamente novos, portanto é muito importante que os equipamentos tenham a capacidade de descobrir de maneira rápida se os dispositivos remotos suportam OAM, já que EFM OAM é opcional e não necessariamente é suportado nos dois lados de um enlace.

O processo de descoberta é, geralmente, ativado do lado da rede da operadora (dispositivo ativo), e pelo lado do EDD (dispositivo passivo) serão respondidas todas as requisições da operadora para estabelecer o estado e configuração do EDD. Outras informações relevantes na camada de transporte/enlace são trocadas, tais como: tamanho do OAMPDU, a identidade do EDD, e se a opção de *loopback* é suportada.

Nas camadas de serviço e conectividade OAM, há um conjunto de funções diferenciadas, que utilizam mensagens diferentes que as definidas no EFM OAM que somente podem ser transferidas num único segmento da rede (*single hop*). Isto é insuficiente para as camadas de serviço e conectividade, já que lidam o tempo todo com provisionamento fim-a-fim.

### 3.3.2 Uso da função OAM *Remote Failure: Link Fault/Critical Event*

A figura 3.6 mostra uma forma de indicação remota de falha por parte do EFM OAM. Isto acontece quando houver uma falha no enlace, tal que uma perda do sinal seja detectada no receptor do EDD. Neste caso, uma mensagem de falha no enlace é enviada uma vez por segundo ao dispositivo que está transmitindo dados ao EDD. Um pouco similar acontece com a mensagem de evento crítico, a qual pode ser enviada quando houver uma falha especificada pelo fabricante, tal como um alarme de HW ou SW, sendo enviada imediatamente e continuamente.

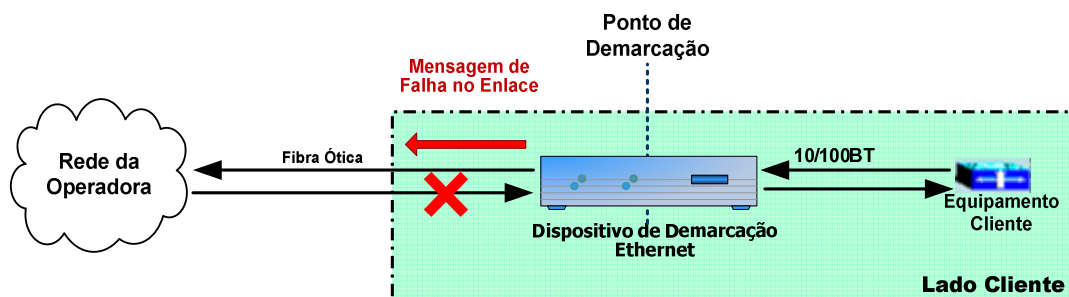


Figura 3.6: Função OAM *Remote Failure: Link Fault/Critical Event*, (Bray, 2006)

### 3.3.3 Uso da função OAM *Remote Failure: Dying Gasp*

A figura 3.7 mostra um cenário onde é potencialmente possível uma falha do tipo *Dying Gasp* e é necessária a indicação remota de falha por parte do EFM OAM. Para que isto seja possível, o EDD precisa incorporar um circuito de armazenamento de energia, para que em casos de perda de alimentação elétrica da unidade, então o EDD seja capaz de manter-se ligado entre 10 e 100ms. Desta forma é possível o envio de uma ou mais mensagens indicando uma falha do tipo *Dying Gasp* à operadora (isto também pode acontecer devido à desconexão do equipamento sem intenções por parte do cliente). Consequentemente, a operadora (através do provedor de serviços) pode entrar em contato com o usuário final para fazer uma avaliação do problema mais exata, evitando assim gastos em deslocamentos até o local da falha.

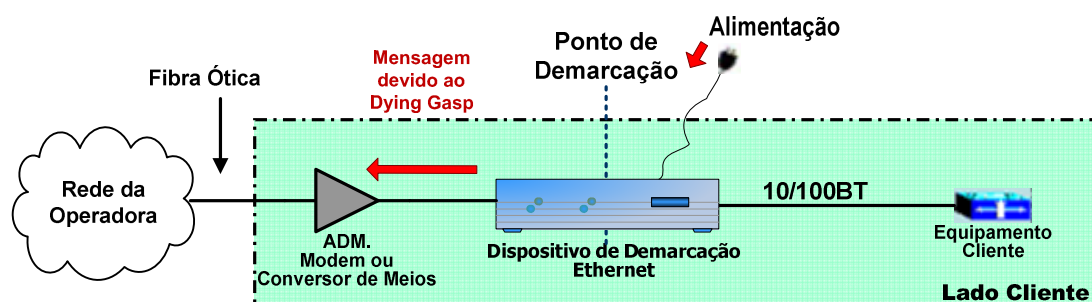


Figura 3.7: Função OAM *Remote Failure* (*Dyng Gasp*), (Bray, 2006)

### 3.3.4 Uso da função OAM *Port Loopback: Link*

OAM *Port Loopback* no nível da porta é definido pelo EFM OAM e pode ser utilizado conjuntamente com um gerador de estímulos para testar um serviço (como mostra a figura 3.8). O teste é intrusivo, portanto deve ser iniciado antes de fornecer o serviço ou quando estiver fora de serviço, sendo muito útil no isolamento e localização das falhas na rede. Alguns EDDs podem suportar funções de teste integrados no próprio equipamento. A RFC2544 - *Benchmarking Methodology for Network Interconnect Devices* é uma referência para avaliação de enlaces Ethernet (RFC, 1999).

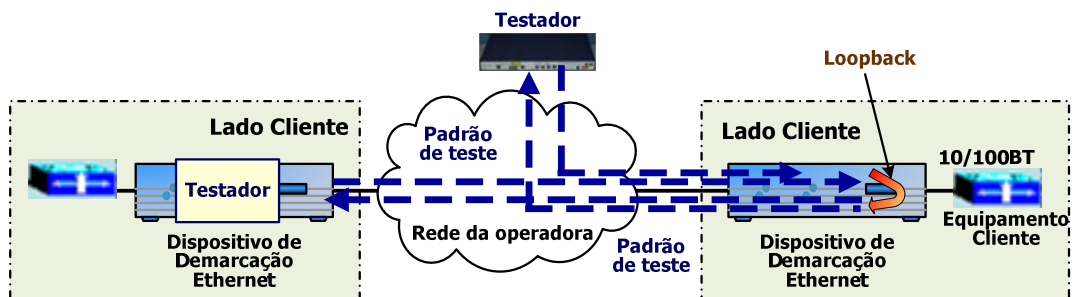


Figura 3.8: Função OAM *Port Loopback: Link*, (Bray, 2006)

### 3.3.5 Uso da função OAM *Loopback (não-intrusivo): Nível do serviço*

Há vários padrões trabalhando no desenvolvimento de OAM *loopback* seletivo os quais, não trabalham no nível da porta e sim no nível de VLANs ou no nível de serviços específicos. Este recurso é poderoso e muito útil em cenários multi-serviço Ethernet, já que é possível testar individualmente cada serviço sem interromper o funcionamento dos outros. Por exemplo, clientes com serviços de *Voice over IP* (VOIP), acesso à internet ou conexão remota, todas sobre o Ethernet, podem ter um serviço adicional de verificação *on-line* sem interromper os serviços oferecidos (Bray, 2006).

A técnica é mostrada na figura 3.9. Esta insere um tráfego de teste com uma taxa reduzida conjuntamente com o tráfego do usuário, para verificar a conectividade fim-a-fim através da rede da operadora (similar ao *ping* das redes IP). Isto permite a extração dos parâmetros úteis para definição de uma SLA.

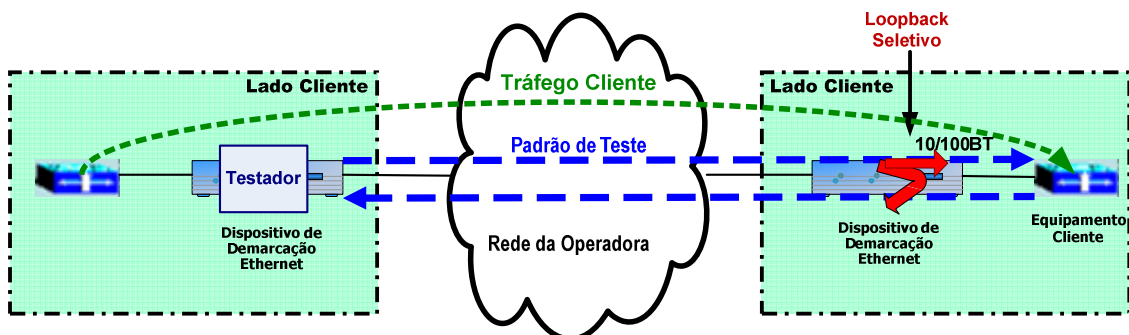


Figura 3.9: OAM *Loopback* (não-intrusivo): Nível do serviço, (Bray, 2006)

### 3.4 Two-Port MAC Relay – IEEE P802.1aj

O grupo de trabalho IEEE 802.1 é o responsável pela aplicação da tecnologia Ethernet nas redes metropolitanas e nas redes de longa distancia. As áreas de atuação do grupo são: Arquiteturas, Inter-operação, Segurança e Gerenciamento global das redes LAN e MAN IEEE 802. A tabela 1 mostra os projetos do grupo de trabalho, sendo o IEEE P802.1aj, uma das referências do presente trabalho (IEEE, 2008).

Tabela 3.1: Projetos do grupo de trabalho IEEE802.1 (ROCHOL, 2006)

Norma	Função	Estado Atual
IEEE 802.1D	Mac Bridges (Spanning Tree, Multi Spanning Tree)	Ultima consolid. 2004
IEEE 802.1Q	Virtual LAN Bridges (Priority, CoS, QoS)	Última revisão 2005
IEEE 802.1G	Remote MAC Bridging	Andamento (2006)
IEEE 802.1p	Traffic Classes and dynamic multi filtering	Em 802.1D (2004)
IEEE 802.1ad	Provider Bridges	Ultima revisão 2005
IEEE 802.1ag	Connectivity fault management	Andamento (2006)
IEEE 802.1ah	Provider Backbone Bridges	Andamento (2006)
IEEE 802.1aj	Two port MAC Relay	Andamento (2006)
IEEE 802.1ak	Multiple Registration Protocol	Andamento (2006)
IEEE 802.1ap	VLAN Bridge MIB	Andamento (2006)
IEEE 802.1aq	Shortest path bridging	Andamento (2006)
IEEE 802.1at	Stream Reservation Protocol	Andamento (2006)
IEEE 802.1au	Congestion Notification	Andamento (2006)
IEEE 802.1AC	Media Access Control Service (Revision)	Andamento (2006)
IEEE 802.1AR	Secure Device Identity (Security)	Andamento (2006)
IEEE 802.1AS	Timing and Sincronization (Áudio, Video bridg.)	Andamento (2006)

O *Two port MAC Relay* se encontra atualmente em desenvolvimento e pode-se encontrar parte da sua especificação no Draft 3.1 da norma P802.1aj (IEEE, 2008). O objetivo deste padrão é especificar as funções, protocolos e procedimentos para a operação de um dispositivo de demarcação do tipo "*MAC Relay*", que faz uso de dois MACs para conectar dois meios da tecnologia IEEE 802 ou meios IEEE 802 emulados (Ex: *Ethernet-over-SONET*). O dispositivo *Two-Port MAC Relay* (TPMR) é transparente a todos os protocolos baseados em pacotes que são independentes do meio, a exceção de aqueles protocolos dirigidos explicitamente a este dispositivo.

Os componentes requeridos do TPMR (*Two-Port MAC Relay*), segundo o IEEE P802.1aj (IEEE, 2008), são:

- a. É compatível com o padrão IEEE 802.1D.
- b. Suporta exatamente duas portas externas (MACs) compatíveis com a norma IEEE 802.3.
- c. Implementa o *Internal Sublayer Service* (ISS) em cada uma das portas. O ISS está definida na interface de serviços do subnível MAC.
- d. Suporta a operação das seguintes funções que definem o processo de encaminhamento de pacotes no TPMR:



1. Filtragem de quadros.
  2. Armazenamento temporal de quadros em cada uma das portas (FIFO – *First-In First Output*).
  3. Gerenciamento das FIFOs.
  4. Seleção da transmissão.
- e. Suporta uma entidade encarregada da propagação do estado entre as duas portas MAC e entre outros TPMR conectados entre si.
- f. Suporta um tipo de Classe de Tráfego em cada uma das portas:
1. Monitoramento do estado da FIFO, e
  2. Se é compatível com a norma IEEE 802.3, então o mecanismo de controle de fluxo também suportado (*MAC Control*).
- g. O TPMR é remotamente gerenciável através de uma de suas portas externas, segundo o protocolo escolhido neste padrão.
- h. Implementa o padrão IEEE 802.3ah OAM na porta do lado do provedor de serviços (opcional)

## 4 SISTEMAS EMBARCADOS RECONFIGURÁVEIS

### 4.1 Arquitetura de um FPGA

A arquitetura genérica de um FPGA (*Field Programmable Gate Array*) é ilustrada na Figura 4.1. Esta consiste em uma matriz de elementos agrupados em blocos lógicos configuráveis, que podem ser interconectados, por barramentos de interconexão configuráveis. Igualmente semelhante a uma PAL (*Programmable Array Logic*), as interconexões entre os elementos são implementados por blocos de chaves configuráveis pelo usuário. Através de blocos de entradas e/ou saídas configuráveis é realizado o interfaceamento com o mundo externo.

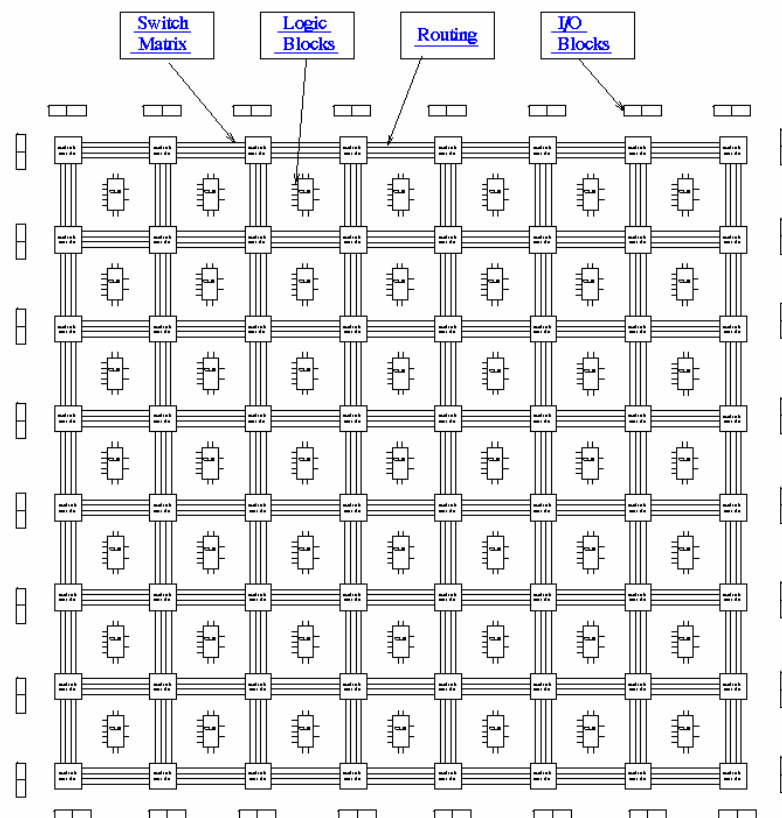


Figura 4.1: Arquitetura de um FPGA genérico (ZEIDMAN, 2001)

Cada fabricante de FPGA tem sua própria arquitetura, mas em termos gerais são todos uma variação da arquitetura mostrada na Figura 4.1. A arquitetura consiste em:

- **Blocos de configuração lógica:** Os blocos lógicos configuráveis contêm a lógica para o FPGA. O CLB (*Configurable Logic Block*) conterà bastante lógica para criar uma máquina de estados, que contem uma memória RAM para criar as funções de lógicas combinatórias. Contem também *flip-flops* para elementos de armazenamento chaveados por *clock*, e multiplexadores a fim de distribuir a lógica dentro do bloco e a dos recursos externos. Os multiplexadores permitem também a seleção da polaridade do reset e a seleção da entrada (TOROK, 2001).
- **Blocos de I/O configurável:** Um bloco de I/O configurável é usado para receber sinais do chip e emiti-los para fora outra vez. Ele consiste em um buffer de entrada e um buffer de saída *tri-state* e controles de saída em coletor aberto. Tipicamente há resistores de *pull-up* nas saídas e às vezes dependendo da arquitetura há resistores *pull-down*. A polaridade da saída pode geralmente ser programada para ser ativo baixo ou ativo alto e frequentemente a taxa de saída pode ser programada para ter tempos de subida e queda altos e baixos. Além disso, há frequentemente um *flip-flop* na saída de modo que os sinais controlados pelo *clock* possam ter saída diretamente aos pinos sem encontrar algum atraso significativo. (TOROK, 2001)
- **Interconexão programável:** A interconexão de um FPGA é muito diferente do que aquela em um CPLD (*Complex Programmable Logic Device*), mas é particularmente similar a um vetor de portas ASIC. Na Figura 4.2, uma hierarquia de recursos de interconexão pode ser vista, onde observa-se linhas longas que podem ser usadas para conectar CLBs críticos onde estão fisicamente distantes dentro do chip sem provocar atrasos maiores que o desejado. Podem também ser usados como barramento dentro do chip. Há também as linhas curtas que são usadas para conectar CLBs individuais fisicamente perto um dos outros. As chaves programáveis dentro do chip, permitem a conexão de CLBs para interconectarem linhas umas as outras e à matriz de chaves (TOROK, 2001).
- **Elementos programáveis:** Em um FPGA, a matriz de blocos lógicos é configurada e interconectada eletricamente através de chaves eletrônicas configuráveis. As propriedades destas chaves, tais como tamanho, resistência (em ohms), e capacitância (em farads), delimitam as características elétricas destes circuitos integrados (ROSE, 1993).

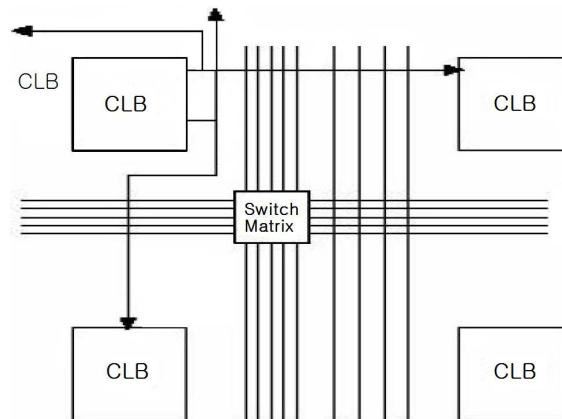


Figura 4.2: Interconexão programável

Os FPGAs foram introduzidos em 1985 pela empresa Xilinx. Desde então, grande variedade de FPGAs foi desenvolvida por várias outras companhias, entre elas: Actel, Altera, Atmel, Plessey, Plus Logic, Advanced Micro Devices (AMD), Quicklogic, Algotronix, Concurrent Logic, e Crosspoint Solutions (ZEIDMAN, 2001). Os FPGA's são chamados assim porque além de terem uma estrutura matricial similar a um PAL ou a outro dispositivo programável, são muito bem estruturados como uma disposição de portas ASIC. Isto faz dos FPGAs muito bons para o uso em prototipação de ASICs, ou em lugares onde ASICs serão usados eventualmente. (ZEIDMAN, 2001).

## 4.2 System-on-a-Chip (SoC)

SoCs podem ser compostos por processadores, memórias, módulos dedicados em hardware para realização de funções específicas, módulos de software e inclusive tecnologias mais recentes como MEMs (do inglês, *Micro-Electro-Mechanical Systems*) integrados no mesmo circuito integrado (CI) (MARTIN, 2001) (JUNEIDI, 2001) (SEMATECH, 2002). A heterogeneidade apresentada por esses sistemas, ou seja, a possibilidade de combinar diferentes tecnologias no mesmo CI tem sido aproveitada em diversos setores da indústria. Tal fato é evidenciado pelo sucesso de produtos eletrônicos como telefones celulares, PDAs, aparelhos de jogos eletrônicos, aparelhos DVDs, máquinas fotográficas digitais (integração de sensores ópticos com lógica digital e analógica), entre outros produtos (BENINI, 2001) (RUNNER, 2000) (SIA, 1999). Porém, combinar diferentes tecnologias no mesmo CI é um processo difícil, que implica o aumento de tempo e custo de projeto de novos produtos.

### 4.2.1 Linguagens para Descrição de SoCs

O procedimento usual para o projeto destes sistemas é utilizar uma dada linguagem para descrever os módulos de hardware e outra linguagem para descrever os procedimentos de software. Linguagens de descrição de hardware, tais como Verilog (THOMAS, 1991) e VHDL (PERRY, 1998), têm como características comuns hierarquia (descrição estrutural com utilização de componentes), paralelismo e temporização. Linguagens para descrição de software, como C ou C++, são baseadas em um modelo de execução seqüencial, adaptadas para a execução em processadores de

propósito geral. Estas linguagens para descrição de software geralmente não têm suporte para modelar paralelismo e temporização.

Exemplos de linguagens utilizadas para modelagem concorrente de hardware e software são SystemC (SYSTEMC, 1993), SystemVerilog (THOMAS, 1991) e SpecC (GAJSKI, 2000). Estas linguagens são derivadas de linguagens de descrição de software, acrescentando-se bibliotecas que simulam as características de paralelismo e temporização. A descrição de sistemas computacionais através destas linguagens permite também aumentar o nível de abstração de projeto. A elevação dos níveis de abstração de projeto para os chamados níveis sistêmicos permite que detalhes de baixo nível sejam abstraídos, tornando mais fácil descrever a funcionalidade de cada núcleo e suas interconexões.

#### 4.2.2 Arquitetura Genérica de um SoC

Como ilustrado na Figura 4.3 (MADISETTI, 1997), um SoC é composto por núcleos de Hardware, processadores e memórias que comunicam-se através de uma estrutura de interconexão e interfaces com o mundo externo.

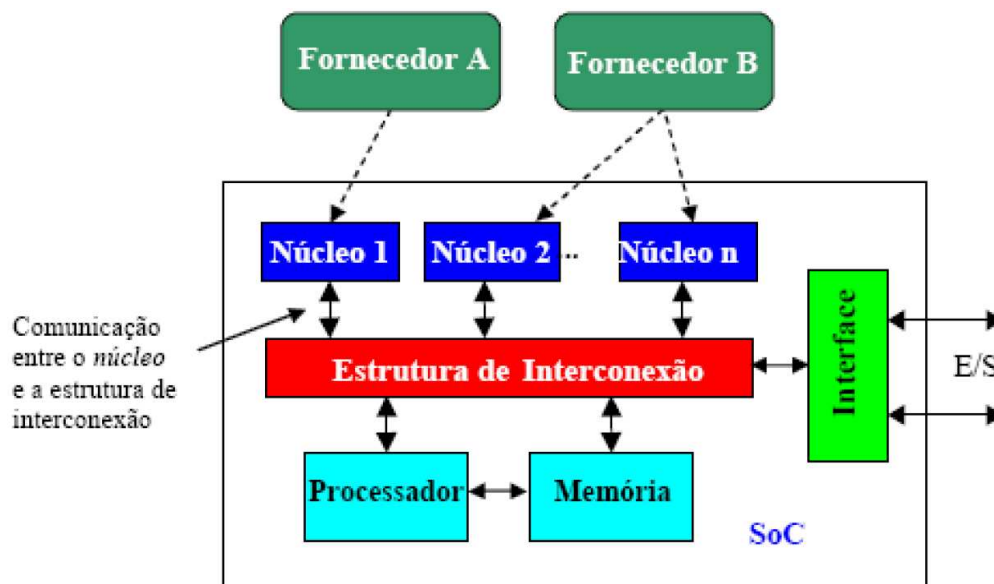


Figura 4.3: Arquitetura genérica de um SoC (OST, 2004).

A arquitetura acima é composta por Módulos IP. Estes módulos ou núcleos podem ter funções voltadas à telecomunicações, DSP (*Digital Signal Process*), decodificadores de MPEG2 (*Motion Picture Experts Group Layer-2*) ou MP3 (*Motion Picture Experts Group Layer-3*), entre outros. Os núcleos provêm de fornecedores distintos (A, B) e estão integrados a uma estrutura de interconexão que pode ser um barramento ou uma rede intra-chip. A(s) interface(s) com o mundo externo (I/O) é (são) utilizada(s) para interconectar periféricos, como: portas USB (*Universal Serial Bus*) ou uma UART (*Universal Assynchronous Receiver Transmitter*).

Gupta *et al.* (GUPTA, 1997) definem um núcleo como um módulo de hardware pré-projetado e pré-verificado, que pode ser usado na construção de uma aplicação maior ou mais complexa em um CI. Estes núcleos podem ser classificados em três categorias: (i) *Soft core*, (ii) *Firm core* e (iii) *Hard core*.

- a. *Soft Core*, é a descrição de um núcleo em uma linguagem de descrição de hardware (e.g. VHDL, Verilog, SystemC). As principais vantagens apresentadas por um *soft core* são: a independência de tecnologia e a flexibilidade. É possível que o projetista/usuário modifique o mesmo, visando adequar a funcionalidade deste em prol do sistema desejado. Em contrapartida a esta flexibilidade, é responsabilidade do projetista atender às restrições temporais. Normalmente, os *soft cores* são acompanhados de *scripts* de síntese, que guiam o projetista na etapa de concepção.
- b. *Firm Core* é um *netlist* que apresenta menos flexibilidade em relação ao *soft core* e na maioria dos casos dependem de uma tecnologia específica (e.g. *netlist* EDIF obtido a partir de síntese lógica). As vantagens em relação aos *soft cores* são: melhor proteção da propriedade intelectual e estimativa de desempenho mais próxima da realidade.
- c. *Hard Core* é geralmente uma descrição de um leiaute posicionado e roteado de um ASIC/FPGA. Para atingir desempenho de processamento, baixo consumo e menor área, estes são otimizados para uma dada tecnologia. Logo, os *hard cores* garantem os tempos de propagação do núcleo (*timing*), além de proverem alta proteção à propriedade intelectual. Como consequência, a flexibilidade é mínima e este é fortemente dependente da tecnologia.

### 4.3 *Soft Processor MicroBlaze*

No segmento de processamento embarcado para aplicações variadas, a Xilinx disponibiliza o *soft processor* MicroBlaze (MicroBlaze, 2007). O MicroBlaze é comercializado pela Xilinx como parte do software EDK (XILINX, 2009). Este software é na realidade um conjunto de vários aplicativos já desenvolvido pela Xilinx e que oferecem suporte ao desenvolvimento de projetos baseados em lógica programável, sejam eles projetos que envolvam o uso de *soft processors* ou não.

A ocupação de lógica do MicroBlaze em um dispositivo é de aproximadamente 800 a 2600 *look-up tables* (LUTs). Alguns dos sistemas operacionais já portados para o MicroBlaze incluem o Nucleus da Mentor Graphics, o ThreadX, o uC/OS-II da Micrium, uClinux da LinuxWorks, dentre outros.

O MicroBlaze é um processador RISC (*Reduced Instruction Set Computing*) de 32 bits, com arquitetura do tipo HARVARD. Este dispositivo é evidentemente otimizado para utilização nas FPGA's da Xilinx. A organização básica deste microprocessador consiste em 32 registros de uso geral, uma ULA (unidade lógica aritmética) e dois níveis de interrupções. A figura 4.6, retirada do guia de referência do MicroBlaze, mostra um diagrama em blocos do núcleo do microprocessador.

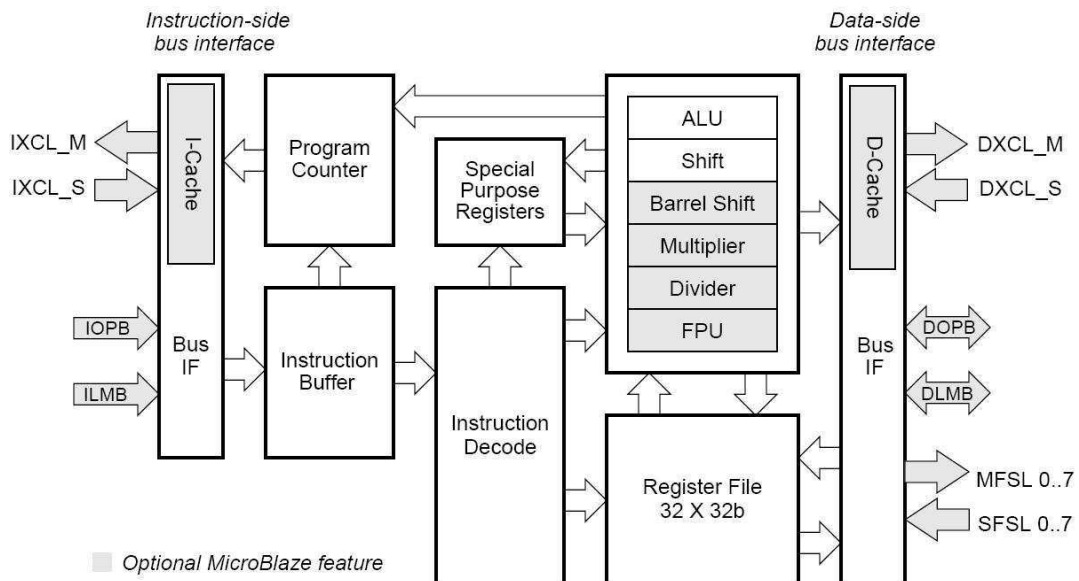


Figura 4.4: Diagrama em blocos do MicroBlaze (MicroBlaze, 2007)

Como abordamos anteriormente, a grande vantagem deste tipo de processador reside no fato de ser possível a inclusão de blocos com características de acordo com a necessidade da aplicação. Pelo diagrama em blocos, podemos observar que algumas partes do núcleo são opcionais. Estes blocos podem ser colocados ou retirados de acordo com a necessidade. Dentre estas características configuráveis estão:

- Profundidade do *pipeline*
- Lógica de *debug* de hardware
- Memória *cache* de instrução
- Memória *cache* de dados
- Suporte à captura de exceções de hardware
- Unidade de ponto flutuante
- Divisor em hardware

## 5 DESCRIÇÃO DO EDD PROPOSTO

### 5.1 Cenário de referência do EDD proposto

O EDD proposto no presente trabalho será especificado para trabalhar num cenário de referência como o definido em (MARTIN, 2007). Ver figura 5.1.

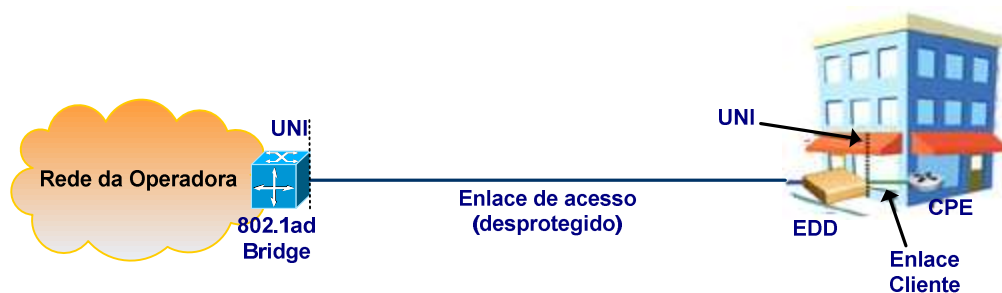


Figura 5.1: Cenário de Referência do EDD proposto (MARTIN, 2007)

Neste cenário de referência é possível identificar três equipamentos:

- CPE (*Customer Premises Equipment*): Termo técnico utilizado pelas operadoras de rede e fornecedores de serviços de comunicação para identificar o equipamento dentro das instalações do cliente.
- 802.1ad *Provider Bridge* (PB): É um tipo de equipamento de borda na operadora de rede. Trata-se de um dispositivo que permite a inserção/detecção de *VLAN stacking* (também conhecido como Q-in-Q) (IEEE, 2005b).
- TPMR: É o EDD de duas portas, baseado na especificação da norma IEEE P802.1aj “*Two-Port MAC relay*” (IEEE, 2008). Neste cenário o TPMR cumpre a função de CLE (*Customer Located Equipment*), e permite o estabelecimento de dois tipos de enlaces: Um enlace ponto-a-ponto entre o EDD-CPE (conhecido como *User-link*) e outro enlace entre o EDD-PB (também conhecido como *Up-link*).

As características que serão consideradas no desenvolvimento do EDD proposto, segundo o cenário de referência mostrado na figura 5.1 são:



- Deve ser transparente a qualquer tipo de serviço Ethernet.
- Meio físico do *User-link*: 10/100BASE-T ou 1000BASE-T (par trançado cobre)
- Meio físico do *Up-link*: 1000BASE-X (fibra óptica mono-modo e multi-modo)
- Não leva em consideração nenhum tipo de serviço, unicamente demarca a rede no nível físico (especificamente nos níveis 1-2 do RM-OSI).
- Proporcionar funcionalidades do OAM Ethernet no Up-link, devido a sensibilidade do enlace de fibra óptica.
- Não suporta funções de uma bridge comum: Filtragem e Aprendizado (*learning*) de endereços MAC, Prevenção de loops e xSTP (todas as opções de *Spanning Tree Protocol*) (MARTIN, 2007)
- Acompanhar as recomendações da norma IEEE P802.1aj, no que refere a função de “*relay*” entre as duas portas do dispositivo.

## 5.2 Especificação técnica do EDD proposto

- Interface Usuário: 10/100BaseTX ou 1000Base-T:
  - Conector: RJ45
  - *Auto-negotiation/Auto MDIX*
- Interface Operadora: 1000BaseX
  - Conector: SFP (multi-modo e mono-modo)
  - Sem *Auto-negotiation* (Gigabit full-duplex forçado)
- Funções NID
  - IEEE 802.3ah Ethernet OAM, na interface Operadora. Sem incluir geração de mensagem devido a condição de *Dying Gasp*.
  - Função *Relay* entre as duas interfaces, baseado na norma IEEE P802.1aj.
  - *Loopback*
    - o *Layer 1*, na interface Usuário;
    - o *Layer 2*, na interface Operadora.
  - Estatísticas do *layer 2* nas duas portas
  - *VLAN Tagging/De-tagging* (IEEE 802.3q)
  - Suporte para *Jumbo frames*
- Desempenho
  - *Throughput*: Encaminhamento de pacotes em *Wire-speed* (taxa máxima)
- Gerenciamento
  - Porta de gerência serial RS-232

### 5.3 Arquitetura do EDD proposto visando uma implementação em FPGAs

O diagrama de blocos apresentado na figura 5.2, mostra o modelo do EDD que será proposto no presente trabalho. O EDD é composto essencialmente por um conversor de meios de duas portas capaz de estabelecer comunicação com componentes de nível PHY Ethernet para a criação das interfaces de usuário e operadora especificadas no modelo apresentado na figura 5.1. O conversor de meios Ethernet é composto por dois controladores MAC Ethernet e buffers para controle do fluxo dos pacotes transferidos pelas interfaces Usuário e Operadora, segundo o recomendado na norma IEEE P802.1aj. Seu comportamento pode ser resumido da seguinte maneira: “Repassar tráfego no modo *duplex* entre suas interfaces de forma transparente, mantendo independência no meio físico e na taxa de transferência escolhida”. A passagem do tráfego através do conversor de meios Ethernet deve respeitar-as restrições das principais camadas e sub-camadas da norma IEEE 802.3.

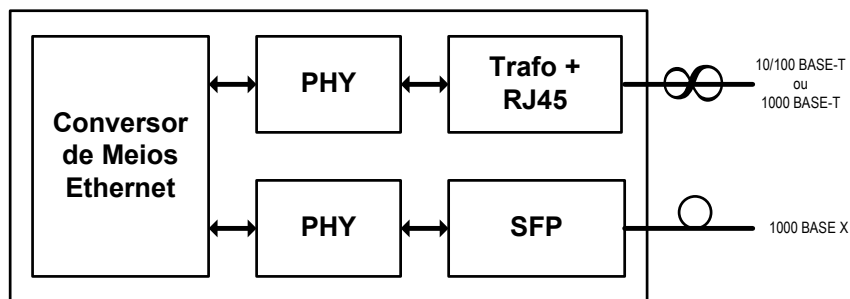


Figura 5.2: Diagrama de Blocos do EDD proposto

A plataforma escolhida no presente trabalho para prototipação do EDD é a placa de desenvolvimento AVNET Virtex-II Pro (AVNET, 2003), baseada no dispositivo FPGA Virtex-II Pro XC2VP30. Esta placa permite o uso do chip PHY *Ethernet Gig PHYTER* ® V 10/100/1000 da *National* que permite a ligação com uma porta RJ-45 para conexão de um cabo de rede par trançado. Além disso, estão disponíveis dois conectores para módulos SFP *transceivers*, sendo possível a conexão de dois pares de fibra ótica mono-modo ou multi-modo (depende do tipo de SFP escolhido). Tanto o chip PHY como o os conectores para módulo SFP estão conectados ao FPGA Virtex-II Pro na (o último conectado no MGT Rocket-IO do Virtex-II Pro). O FPGA Virtex-II Pro recebe um relógio diferencial de 125MHz e o chip PHY possui seu próprio cristal de 25MHz. A figura 5.3 mostra os principais periféricos da placa de desenvolvimento AVNET Virtex-II Pro.



Figura 5.3: Placa de desenvolvimento AVNET Virtex-II Pro

Em (Horna, 2006)(Horna, 2007), são especificados os detalhes do projeto, implementação e validação de três núcleos de propriedade intelectual (também conhecidos como *IP Soft Cores*) destinados a executar funções essenciais da tecnologia Ethernet. Dois dos três *IP Soft Cores*, implementam as funções correspondentes à sub-camada de controle de acesso ao meio (MAC) para taxas de transmissão de 10/100 e 1000 Mbits/s, respectivamente. O terceiro *IP Soft Core* implementa as sub-camadas PCS e PMA do sistema de mídia 1000BASEX, para prover interligação com o meio físico de fibra ótica. Para auxílio na validação dos núcleos de propriedade intelectual foi necessário projetar um quarto módulo IP, chamado de *Packet FIFO*, para a contenção dos quadros Ethernet transferidos pelo sub-camada MAC. A descrição dos circuitos foi totalmente modular e realizada na linguagem VHDL.

Partindo do modelo do EDD proposto e das especificações técnicas definidas em 5.2, é possível re-aproveitar os núcleos IP desenvolvidos pelo autor deste trabalho (Horna, 2006)(Horna, 2007) para posteriormente elaborar uma arquitetura de HW de um EDD visando uma implementação em FPGAs. Nesse sentido e para satisfazer as especificações definidas no presente capítulo, é possível propor duas arquiteturas para o EDD. Cada arquitetura será identificada em função do tipo do Conversor de Meios Ethernet:

1. Arquitetura I: Baseada num Conversor de Meios Ethernet “FE2GE”, 10/100Mbits/s (porta elétrica) para 1000 Mbit/s (porta ótica)
2. Arquitetura II: Baseada num Conversor de Meios Ethernet “GE2FE”, 1000Mbits/s (porta elétrica) para 1000Mbit/s (porta ótica).

A figura 5.4 mostra as duas arquiteturas para o EDD proposto, tendo em consideração o uso de um conjunto de núcleos IPs. Além disso, na figura são mostrados os domínios de relógio.

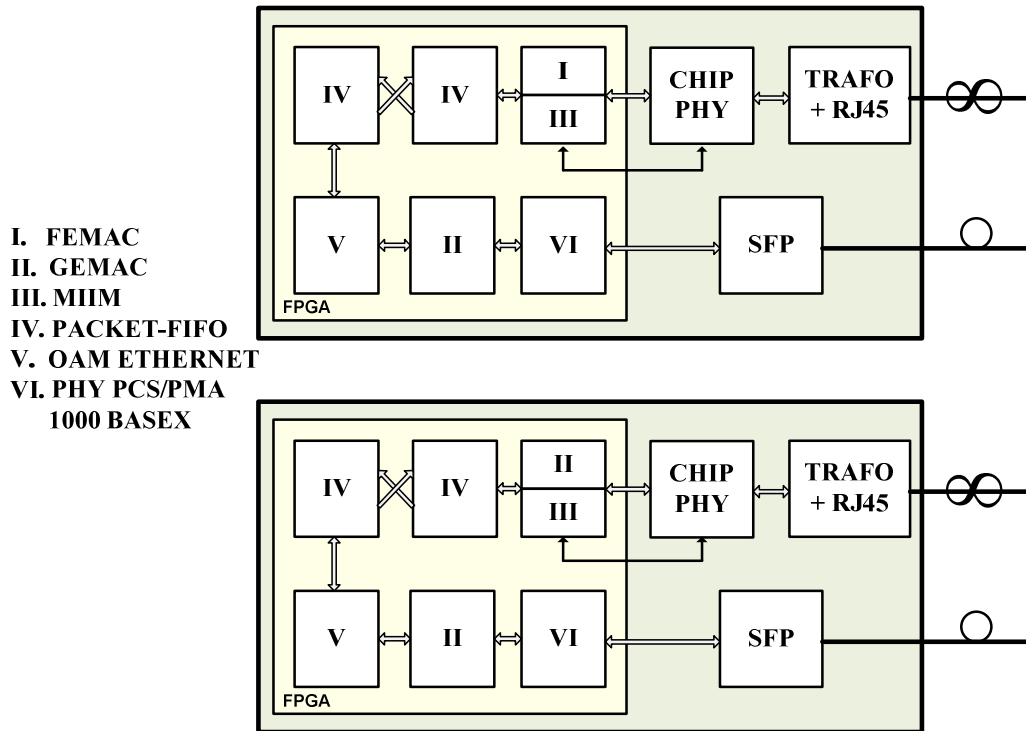


Figura 5.4: Duas arquiteturas para o EDD proposto

A presença do IP *Soft Core* OAM Ethernet na definição das duas arquiteturas apresentadas acima, introduz a idéia do desenvolvimento de arquiteturas HW/SW, já que parte das funções do OAM Ethernet (OAM *client*) serão implementadas em software, que será executado no processador embarcado MicroBlaze. Na figura 5.5 é mostrada a Arquitetura HW/SW do EDD proposto, nas suas duas versões.

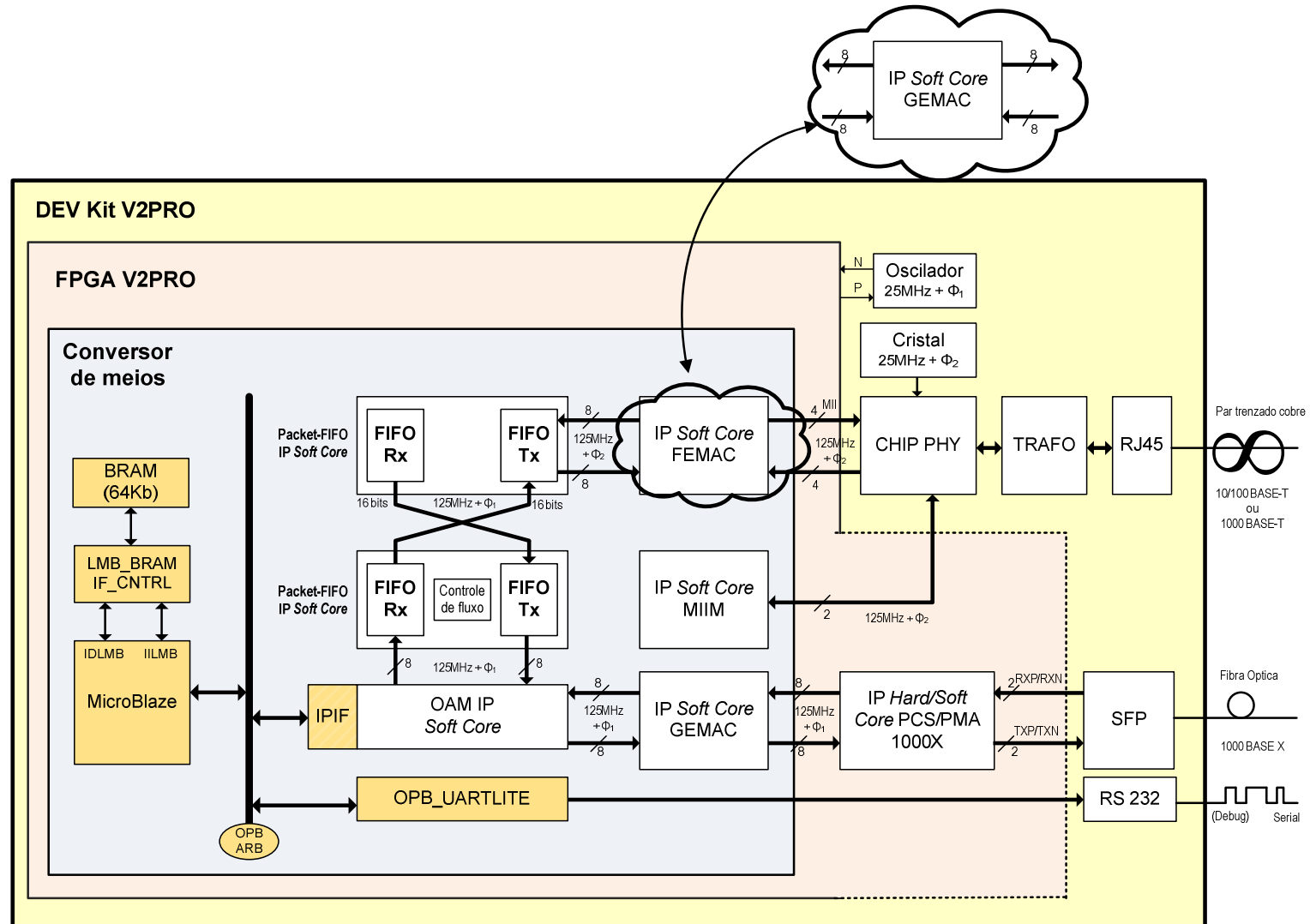


Figura 5.5: Arquitetura HW/SW do EDD proposto

## 6 DESENVOLVIMENTO DOS NÚCLEOS IP PARA O EDD PROPOSTO

### 6.1 Projeto dos núcleos IP do EDD proposto

A seguir a lista de Módulos IP que serão apresentados neste capítulo:

- IP *Soft Core* FEMAC e GEMAC
- IP *Soft Core* MIIM
- IP *Soft Core* Packet FIFO
- IP *Soft Core* OAM Ethernet
- IP *Hard/Soft Core* PHY1000X

#### 6.1.1 IP *Soft Core* FEMAC

O core é responsável pela implementação em hardware digital do sub-nível MAC para taxas de transmissão de 10/100 Mbps (*half-duplex* e *full-duplex*) segundo as cláusulas 2, 3 e 4 da norma IEEE 802.3 (IEEE, 2005a). Na figura 6.1 são apresentadas com linhas sombreadas (RS, MAC e MAC *Control*) os elementos do modelo OSI implementados pelo IP *Soft Core* FEMAC.

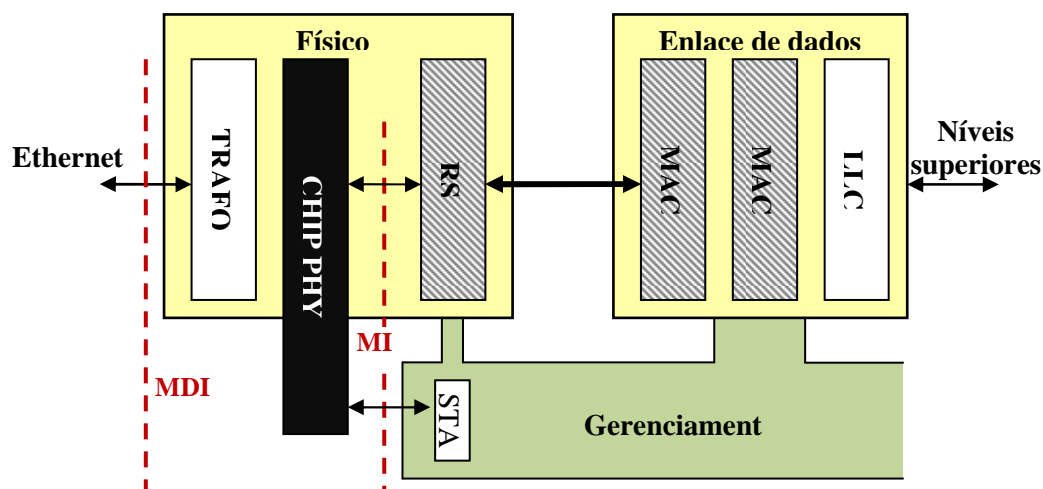


Figura 6.1: IP *Soft Core* FEMAC e o modelo OSI

As funcionalidades do IP *Soft Core* FEMAC estão organizadas em sete módulos como mostra a figura 6.2.

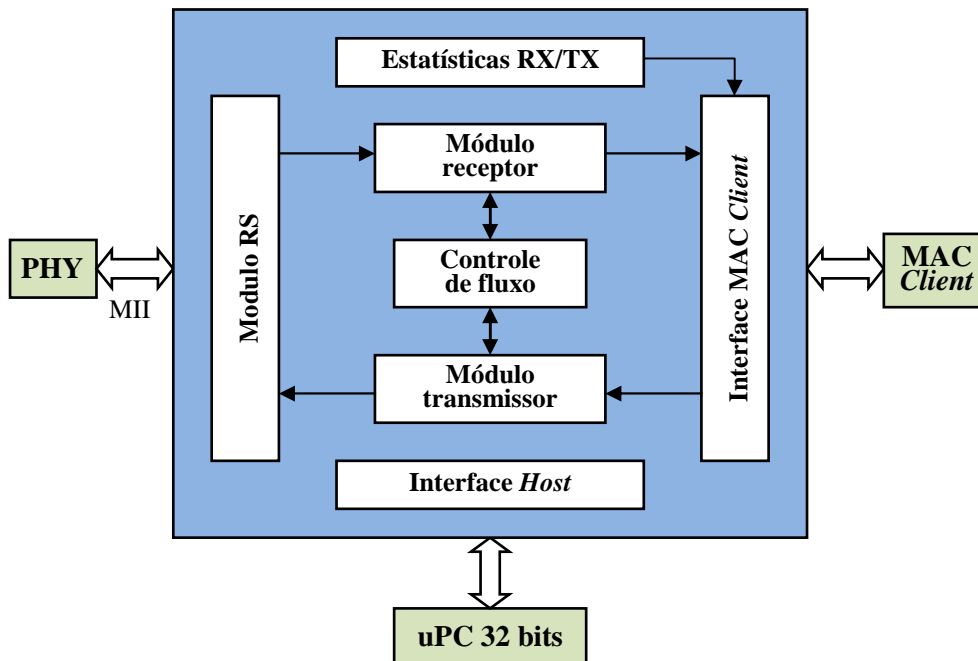


Figura 6.2: Módulos funcionais do IP *Soft Core* FEMAC

A seguir é descrito brevemente cada um dos módulos que compõem o IP *Soft Core* FEMAC.

#### 6.1.1.1 Módulo Receptor

Esse módulo implementa as funções do protocolo CSMA/CD para a recepção de quadros Ethernet e algumas funções extras fora do padrão definidas segundo as exigências do projeto. Na continuação, são descritas as principais características funcionais do módulo:

- Remove os campos de preâmbulo, SFD e FCS.
- Detecta automaticamente quadros com número não inteiro de octetos ou com quantidade de octetos não compatível segundo a norma IEEE 802.3.
- Implementa o mecanismo CRC-32 para detecção de erros no quadro recebido.
- Filtra endereços broadcast, *multicast*, e detecta quadros com *Tag* VLAN.
- Funções adicionais fora do padrão: Filtra cinco endereços MAC diferentes, suporta modo promíscuo e recepção de quadros tipo Jumbo de até 64KBytes.

Na figura 6.3 são mostrados os blocos do módulo Receptor, os quais são descritos brevemente.

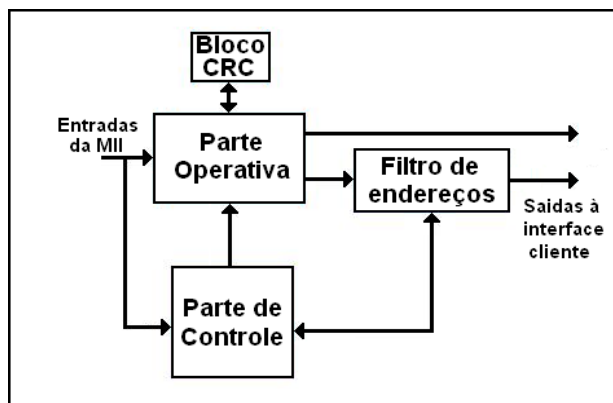


Figura 6.3: Diagrama de blocos do módulo Receptor

#### 6.1.1.1.1 Bloco CRC.

Implementa o algoritmo CRC-32 definido na cláusula 3 da norma IEEE 802.3 (IEEE, 2005a). Possui uma LFSR (*Linear feedback shift register*) com entrada de dados de 4 bits, utilizado para o computo do valor de CRC de 32 bits. Para a detecção de erro no quadro recebido, o bloco computa continuamente o valor de CRC (a partir dos dados recebidos, incluindo o próprio campo de FCS) e no final da recepção, compara o último valor computado com um valor constante de 32 bits. Se o resultado da comparação difere então é detectado um erro de CRC no quadro recebido.

#### 6.1.1.1.2 Parte Operativa.

Esse bloco é agrupado em dois sub-blocos:

- Sub-bloco *RxDatapath*: Realiza a conversão dos *nibbles* vindos da MII para o formato de bytes. Isto é feito utilizando um montador de palavras composto por registradores para concatenação dos dados. Também possui *flip-flops* para geração dos sinais de início de quadro (*Start of Frame* ou SOF) e fim de quadro (*End of frame* ou EOF), os quais serão encaminhados à Interface Cliente de 8 bits.
- Sub-bloco *RxCounters*: Utiliza um contador de 16 bits com saída decodificada para contabilizar os octetos recebidos e controlar a posição dos octetos na filtragem de endereços. Também implementa comparadores para verificar se o comprimento do quadro possui um número inteiro de octetos e se os comprimentos máximo e mínimo do quadro cumprem os requisitos que impõe o padrão. No modo promíscuo são aceitos quadros de tipo *Jumbo*.

#### 6.1.1.1.3 Filtro de Endereços.

Esse módulo realiza comparações de maneira seqüencial (*byte-a-byte*) do campo de “Endereço Destino”, para detecção do endereço *broadcast* e para detecção de cinco endereços *unicast* MAC de 48 bits pré-definidos. Para isto, são utilizados cinco conjuntos de registradores e comparadores para verificação dos endereços em forma paralela. Utilizando o mesmo princípio, o sub-bloco armazena e compara os dados seguintes ao campo de “Endereço Origem”, para detecção de *Tag* VLAN. Também possui uma lógica seqüencial baseada em tabelas de *Hash* para acelerar a detecção de



endereços *multicast*, verificando somente os primeiros 10 bits dos 48 bits que conformam o endereço.

#### 6.1.1.1.4 Módulo de controle de fluxo.

Implementa uma máquina de estados com as seguintes características:

- Número de estados: 5.
- Número de transições: 13.
- Número de entradas: 4.
- Número de saídas: 5.
- Ativada na borda de subida do sinal RX\_CLK da MII.
- *Reset* assíncrono.

#### 6.1.1.2 Módulo Transmissor

Esse módulo implementa as funções do protocolo CSMA/CD para a transmissão de quadros Ethernet e algumas funções extras fora do padrão definidas segundo as exigências do projeto. Na continuação, são descritas as principais características funcionais do módulo:

- Gera os campos de preâmbulo, SFD e FCS.
- Gera automaticamente o Inter-Frame Gap (IFG).
- Complementa automaticamente o tamanho mínimo do quadro transmitido (*padding*).
- Detecta colisões e gera o sinal de Jam.
- Implementa o algoritmo randômico de “*Truncated binary exponential backoff*” e o processo de deferência.
- Funções adicionais fora do padrão: Habilita a transmissão de quadros tipo Jumbo.

Na figura 6.4 é mostrado o diagrama de blocos simplificado do módulo Transmissor.

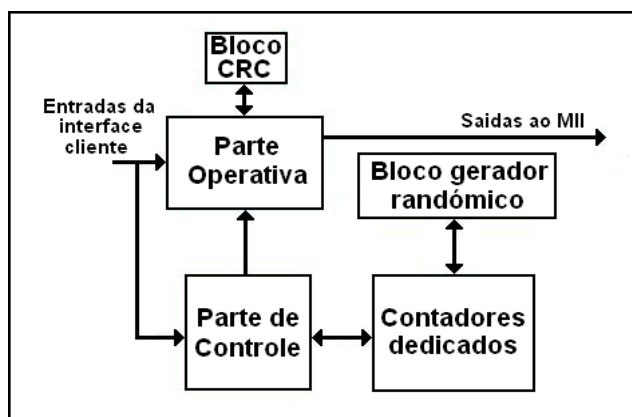


Figura 6.4: Diagrama de blocos simplificado do módulo Transmissor  
A seguir é descrito brevemente cada um dos blocos que compõem o módulo:

#### 6.1.1.2.1 Bloco de CRC.

Implementa o algoritmo CRC-32 definido na cláusula 3 da norma IEEE 802.3 (IEEE, 2005a). Possui uma LFSR com entrada de dados de 4 bits, utilizada para o computo do valor de CRC de 32 bits a partir dos campos de “Endereço Destino”, “Endereço Origem”, “Tamanho do quadro” e o campo de dados (incluindo os zeros introduzidos no processo de *padding*). O valor de 32 bits gerado é encapsulado no campo de FCS (em oito *nibbles*), para posteriormente ser transmitido.

#### 6.1.1.2.2 Parte Operativa.

Realiza a conversão dos bytes vindos da Interface Cliente de 8 bits para o formato de *nibbles* da MII. Isto é feito utilizando um multiplexador e um conjunto de registradores de 4 bits, os quais operando de forma conjunta, permitem o encaminhamento de vários tipos de dados, tais como: o *nibble* de dados mais significante, o *nibble* de dados menos significante, o padrão de *Jam*, o valor de CRC, o valor de SFD, o valor do preâmbulo e os zeros de preenchimento do processo de *padding*. A parte operativa também realiza o ajuste da janela de colisão, isto é, o tempo no qual uma colisão é válida durante a transmissão de um quadro. Finalmente, possui *flip-flops* para geração das sinais de estado da transmissão de quadros.

#### 6.1.1.2.3 Contadores Dedicados.

Implementa 4 tipos de contadores com saídas decodificadas:

- Contador de 16 bits para contabilizar os octetos transmitidos.
- Contador de 16 bits para contabilizar os *nibbles* transmitidos (é utilizado no processo de deferência, assim como também, na parte operativa do módulo e pelo gerador randômico)
- Contador de 4 bits para contabilizar a quantidade de retransmissões.

#### 6.1.1.2.4 Bloco Gerador Randômico.

Implementa o algoritmo de randomização chamado “*truncated binary exponential backoff*” especificado na cláusula 3 da norma IEEE 802.3 (IEEE, 2005a). O bloco gera um tempo de postergação da transmissão de quadros quando acontece uma colisão no canal compartilhado. Para isto, foi necessário escolher uma LFSR para a geração dos números aleatórios.

#### 6.1.1.2.5 Bloco de controle.

Implementa uma máquina de estados com as seguintes características:

- Número de estados: 10.
- Número de transições: 33.
- Número de entradas: 10.
- Número de saídas: 10.
- Ativada na borda de subida do sinal TX\_CLK da MII.
- *Reset* assíncrono.

### 6.1.1.3 Módulo de Controle de Fluxo

Esse módulo trabalha de maneira concorrente com o módulo Receptor e o módulo Transmissor. A comunicação é feita através de uma lógica de *handshaking* devido aos diferentes domínios de relógio dos módulos Receptor e Transmissor. A continuação, são descritas as principais características funcionais do módulo:

- Gera e/ou detecta quadros de *pause*.
- Extração do tempo de pausa e bloqueio do transmissor durante o período de pausa.

Na figura 6.5 é mostrado o diagrama de blocos simplificado do módulo de Controle de fluxo, e logo é descrito brevemente cada um dos blocos que compõem o módulo:

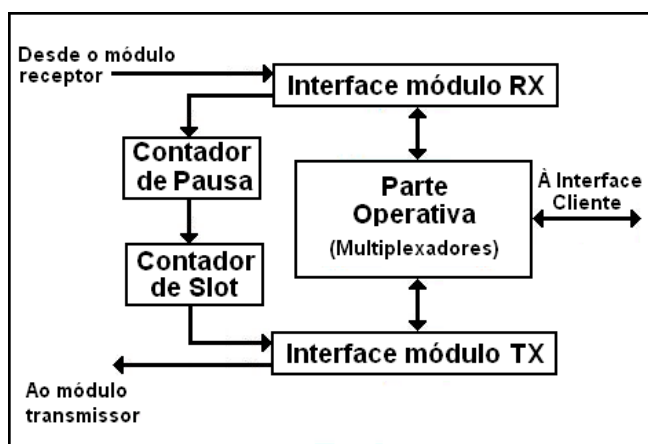


Figura 6.5: Diagrama de blocos simplificado do módulo de Controle de fluxo

#### 6.1.1.3.1 Interface Módulo RX.

Utiliza um contador de 5 bits com saída decodificada para contabilizar os octetos dos quadros recebidos e para detecção de quadros de *pause*. Caso seja detectado um quadro de *pause*, se extrai o campo de “Tempo de pausa”.

#### 6.1.1.3.2 Os contadores de Pausa e de Slot.

O contador de pausa é de 16 bits e conta até o valor indicado no campo de “Tempo de pausa”. O incremento do contador de pausa acontece somente no fim da conta do contador de Slot. O contador de Slot é de módulo 512 devido ao critério de implementação definido no padrão. Os contadores ativam o sinal de bloqueio do módulo Transmissor.

#### 6.1.1.3.3 Interface Módulo TX.

Utiliza um contador de 6 bits para a geração dos octetos especiais do quadro de *pause*, os que serão transmitidos pelo módulo Transmissor no caso exista uma requisição da Interface Cliente de 8 bits. Também possui um conjunto de registradores de 8 bits e um multiplexador de saída para transferir os campos do quadro de *pause*.

#### 6.1.1.3.4 Parte Operativa.

Utiliza multiplexadores para encaminhar os quadros normais ou os quadros especiais de *pause*. Também gera os sinais de SOF e EOF para os quadros de *pause*. Finalmente possui um conjunto de *flip-flops* para habilitar o *padding* e o CRC nos quadros normais ou de *pause*.

##### 6.1.1.4 Módulo de Estatísticas RX/TX

O módulo armazena num banco de registradores a atividade do Core IP *Soft MAC* 10/100. As estatísticas dos eventos de recepção ou transmissão do Core são encapsuladas num vetor que é proporcionado pela Interface Cliente de 8 bits. As tabelas 6.1 e 6.2 mostram os vetores de estatística de recepção e transmissão respectivamente. Na continuação, são descritas as principais características funcionais do módulo:

- Estatísticas RX: Indica se o quadro recebido é do tipo: *pause*, *unicast*, *broadcast*, *multicast*. Também indica o tamanho do quadro recebido e se está livre de erros.
- Estatísticas TX: Indica o tamanho do quadro transmitido e se é do tipo *pause*. Também indica os eventos do protocolo CSMA/CD.

Tabela 6.1: Definição dos bits do vetor de estatística da recepção no Core

Vetor[21:0]	Nome	Descrição
21	FLOW_CONTROL_FRAME	Ativado se o quadro recebido está livre de erros e é um quadro especial de <i>pause</i> .
20	VLAN_FRAME	Ativado se o quadro recebido contém um identificador VLAN ( <i>Tag</i> ) no campo de tamanho do quadro.
19	RECEIVED_LENGTH_OK	Indica se o quadro recebido possui um comprimento correto segundo o padrão.
18	RESERVED	Retorna o valor lógico 0.
[17:7]	FRAME_LENGTH_COUNT	Comprimento do quadro recebido em número de bytes. O contador chega até 2047 para qualquer quadro <i>Jumbo</i> maior que esse valor.
6	RESERVED	Retorna o valor lógico 0.
5	MULTICAST_FRAME	Ativada se o quadro recebido contém um endereço <i>multicast</i> no campo de endereço destino.
4	BROADCAST_FRAME	Ativada se o quadro recebido contém um endereço <i>broadcast</i> no campo de endereço destino.
3	UNICAST_FRAME	Ativada se o quadro recebido contém um dos cinco endereços MAC no campo de endereço destino.

Vetor[21:0]	Nome	Descrição
2	FCS_ERROR	Ativada se o quadro recebido contem erro de CRC.
1	BAD_FRAME	Ativada se o quadro recebido contém erros.
0	GOOD_FRAME	Ativada se o quadro recebido está livre de erros.

Tabela 6.2: Definição dos bits do vetor de estatística da transmissão no Core

Vetor[17:0]	Nome	Descrição
17	PAUSE_FRAME_TX	Ativado se o quadro transmitido é um quadro especial de <i>pause</i> .
16	RESERVED	Retorna o valor lógico 0.
[15:5]	FRAME_LENGTH_COUNT	Comprimento do quadro transmitido em número de bytes. O contador chega até 2047 para qualquer quadro <i>Jumbo</i> maior que esse valor.
4	RESERVED	Retorna o valor lógico 0.
3	EXCESSIVE_COLLISION	Ativada se acontece uma colisão depois do número máximo de colisões permitidas.
2	LATE_COLLISION	Ativada se acontece uma colisão tardia durante a transmissão.
1	EXCESSIVE_DEFERRED	Ativada se a transmissão de um quadro é anulada pelo processo de deferência.
0	TX_DEFERRED	Ativada se o quadro transmitido é deferido.

#### 6.1.1.5 Módulo RS

O módulo adapta os sinais da MII, que posteriormente são utilizadas pelos módulos de recepção e transmissão. Na continuação, são descritas as principais características funcionais do módulo:

- Implementa o sub-nível de reconciliação da MII, para transferências de dados em formato de *nibbles* com uma frequência de operação de 2.5MHz (10 Mbps) ou 25 MHz (100 Mbps).
- Funções adicionais fora do padrão: Opção de *loopback* na MII para validação funcional do Core.

### 6.1.1.6 Interface Host

Essa interface permite a configuração do Core através de um banco de registradores que armazena os parâmetros de implementação 10/100 Mbps, os quais são definidos na cláusula 4 da norma IEEE 802.3 (IEEE, 2005a) e mostrados na tabela 6.3.

Tabela 6.3: Parâmetros de implementação do Core

Vetor	Valor	Descrição
MINFL[15:0]	0x40	Indica o tamanho mínimo do quadro (64 bytes).
MAXFL[15:0]	0x600	Indica o tamanho máximo do quadro (1536 bytes, considerando espaço adicional para os <i>Tags</i> VLAN).
HASH_REG0[31:0]	(*)	Tabela#1 de <i>Hash</i> para detecção do quadro <i>Multicast</i> .
HASH_REG1[31:0]	(*)	Tabela#2 de <i>Hash</i> para detecção do quadro <i>Multicast</i> .
COLLVALID[5:0]	0x3F	Representa a janela de tempo de colisão.
IFG[6:0]	0x17	Indica o valor do <i>Inter-Frame Gap</i> (96 us/10Mbps e 0.96 us/100Mbps).
IFG_FRACTION[6:0]	0x0F	Indica 2/3 do valor do <i>Inter-Frame Gap</i> (utilizado para fazer mais robusta a detecção de colisões).
MAXRET[3:0]	0x0F	Indica o número máximo de retransmissões.
TX_PAUSE[13:0]	(**)	Indica o valor do tempo de pausa utilizado no processo de controle de fluxo.
PHY_ADDRESS[4:0]	(**)	Endereço do PHY utilizado no protocolo MDIO.
REG_ADDRESS[4:0]	(***)	Endereço do registrador do PHY que se quer acessar pelo protocolo MDIO.

(\*) O valor certo na tabela de *Hash* é igual ao valor do endereço *multicast* que se quer verificar.

(\*\*) O valor do tempo de pausa é escolhido pelo Cliente.

(\*\*\*) O valor do registrador que se quer acessar vai depender do STA.

### 6.1.1.7 Interface MAC Client

A função principal da interface é a de comunicar o IP *Soft Core* FEMAC com o seu cliente, localizado num nível superior dentro do RM-OSI. O presente trabalho propõe uma interface de *MAC Client* como é apresentada na tabela 6.4.

A recepção e transmissão de quadros no lado do cliente devem manter a mesma taxa de transmissão (10/100Mbps). A transferência de dados no formato de bytes obriga o uso de uma frequência de relógio menor (dividida por dois). Por esse motivo, a interface utiliza dois divisores de frequência de módulo 2 para criação dos relógios

FEMAC\_CLIENT\_RX\_CLK e FEMAC\_CLIENT\_TX\_CLK, ambos com uma frequência de operação de 1.25MHz(10Mbps) ou 12.5 MHz (100Mbps).

Tabela 6.4: Sinais da Interface Cliente de 8 bits do Core

Sinal	Direção	Descrição
FEMAC_CLIENT_RX_CLK	Saída	Relógio para recepção de quadros no Cliente.
FEMAC_CLIENT_RXD[7:0]	Saída	Dados do quadro recebido pelo Core em formato de bytes.
FEMAC_CLIENT_RX_SOF	Saída	Indica o início da recepção de um quadro no Cliente.
FEMAC_CLIENT_RX_EOF	Saída	Indica o fim da recepção de um quadro no Cliente
FEMAC_CLIENT_RX_ABORT	Saída	Indica que o quadro recebido no Core contém erros.
RX_STATISTICS_VECTOR[21:0]	Saída	É o vetor de estatística da recepção do Core.
RX_STATISTICS_VALID	Saída	É um sinal ativado durante um ciclo do relógio no final da recepção do quadro.
FEMAC_CLIENT_TX_CLK	Saída	Relógio para transmissão de quadros pelo Cliente.
FEMAC_CLIENT_TXD[7:0]	Entrada	Dados a serem transmitidos em formato de bytes pelo Cliente.
FEMAC_CLIENT_TX_SOF	Entrada	Indica o início da transmissão de um quadro pelo Cliente.
FEMAC_CLIENT_TX_EOF	Entrada	Indica o fim da transmissão de um quadro pelo Cliente
FEMAC_CLIENT_TX_ACK	Saída	Indica ao Cliente iniciar a transmissão do segundo byte do quadro (o primeiro byte do quadro é colocado pelo sinal de FEMAC_CLIENT_TX_SOF).
FEMAC_CLIENT_TX_DONE	Saída	Indica que o quadro foi corretamente transmitido pelo Core.
FEMAC_CLIENT_TX_ABORT	Saída	Indica que o quadro foi transmitido com erro pelo Core.

Sinal	Direção	Descrição
FEMAC_CLIENT_TX_RETRY	Saída	Indica ao Cliente que o quadro deve ser retransmitido.
TX_STATISTICS_VECTOR[17:0]	Saída	É o vetor de estatística da transmissão do Core.
TX_STATISTICS_VALID	Saída	É um sinal ativado durante um ciclo do relógio TX_CLK da MII ao final da recepção do quadro.
TX_PAUSE_REQ	Entrada	Requisição do Cliente para a transmissão de um quadro especial de <i>pause</i> .
WILLSEND_CONTROLFRAME	Saída	Indica que o Core está enviando um quadro especial de <i>pause</i> (para diferenciar de uma transmissão normal).

O diagrama temporal mostrado na figura 6.6 representa um processo típico de recepção de quadro Ethernet pelo Cliente. Os parâmetros mostrados na figura são: DA → *Destination Address*, SA → *Source Address*, L/T → *Length or Type*, DATA → *Data*, FCS → *Frame Check Sequence*

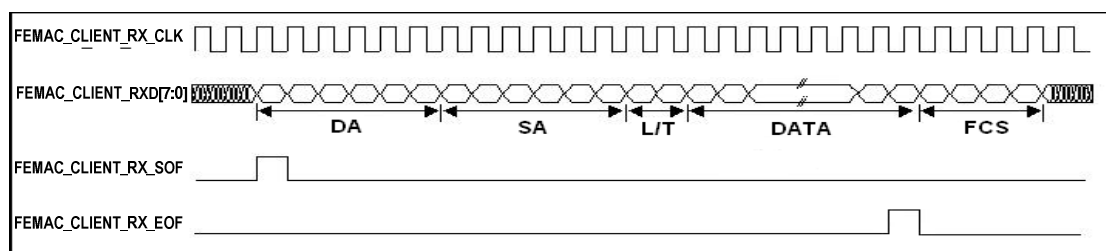


Figura 6.6: Diagrama temporal de uma típica recepção no lado do Cliente

O diagrama temporal mostrado na figura 6.7 representa um processo típico de transmissão de quadro Ethernet pelo Cliente. Os parâmetros mostrados na figura são: DA → *Destination Address*, SA → *Source Address*, L/T → *Length or Type*, DATA → *Data*, FCS → *Frame Check Sequence*.

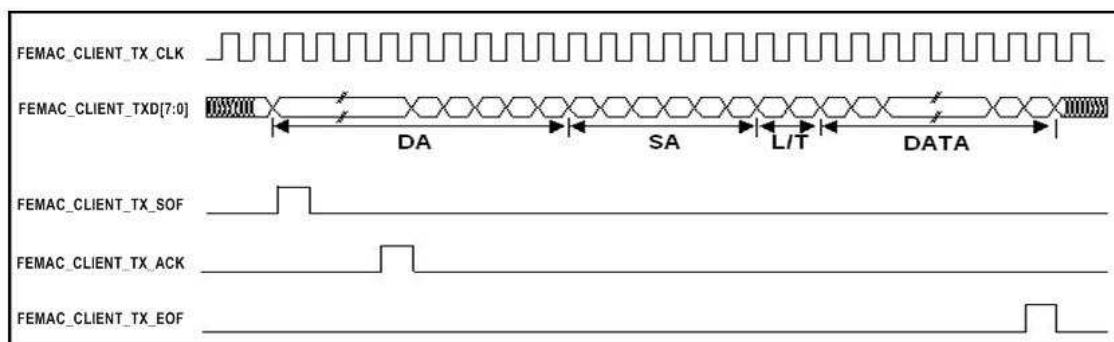


Figura 6.7: Diagrama temporal de uma típica transmissão no lado do Cliente



### 6.1.2 IP Soft Core GEMAC

Na figura 6.8 são apresentadas com linhas sombreadas (RS, MAC e MAC Control) os elementos do modelo OSI implementados pelo IP Soft Core GEMAC. Além disto, saliente-se que a interface entre RS e o CHIP PHY é GMII.

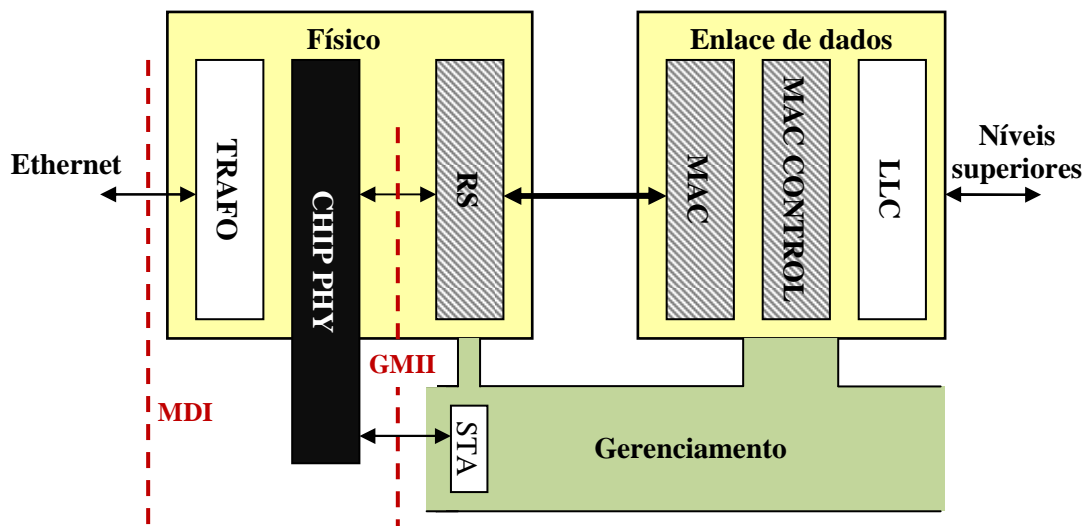


Figura 6.8: IP Soft Core GEMAC e o modelo OSI

O Core é responsável pela implementação em hardware do sub-nível MAC para uma taxa de transmissão de 1000 Mbps (*full-duplex*) segundo as cláusulas 2, 3 e 4 da norma IEEE 802.3 (IEEE, 2005a). As funcionalidades do Core estão organizadas em sete módulos como se mostra na figura 6.9. Esse Core é equivalente ao IP Soft Core FEMAC, com as seguintes diferenças funcionais:

- Suporta unicamente o modo de operação full-duplex.
- Não implementa as funcionalidades próprias do protocolo CSMA/CD.
- Para interação com o nível físico, implementa um módulo baseado na GMII.
- A frequência de operação do Core é 125MHz, a diferença do Core MAC 100 que trabalha com uma frequência de operação de 2.5MHz(10Mbps) - 25MHz(100Mbps).

A figura 6.9 apresenta os módulos funcionais que conformam o IP Soft Core GEMAC, e logo são descritos cada um deles.

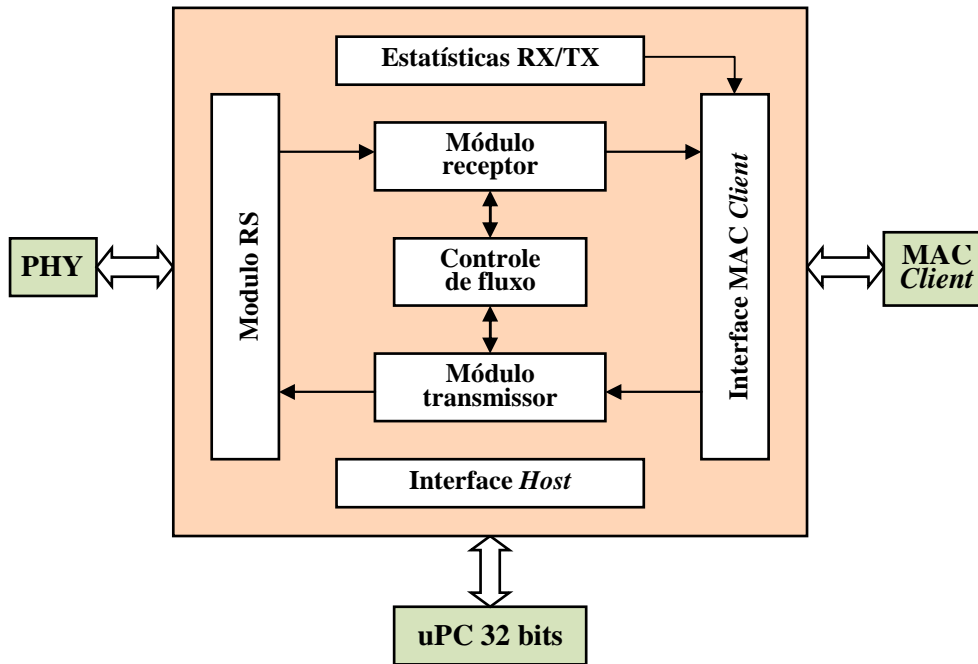


Figura 6.9: Módulos funcionais do IP *Soft Core* GEMAC

#### 6.1.2.1 Módulo Receptor

Esse módulo não implementa o protocolo CSMA/CD para a recepção de quadros Ethernet, mas mantém as funções básicas do modo *Full duplex* e implementa outras funções fora do padrão. As principais características funcionais do módulo são:

- a. Remove os campos de preâmbulo, SFD e FCS.
- b. Detecta automaticamente quadros com número não inteiro de octetos ou com quantidade de octetos não compatível segundo a norma IEEE 802.3.
- c. Implementa o mecanismo CRC-32 para detecção de erros no quadro recebido.
- d. Filtra endereços *broadcast*, *multicast*, e detecta quadros com *Tag* VLAN.
- e. Funções adicionais fora do padrão: Filtra cinco endereços MAC diferentes, suporta modo promíscuo e recepção de quadros tipo *Jumbo* de até 64KBytes.

A figura 6.10 mostra o diagrama de blocos simplificado do módulo Receptor.

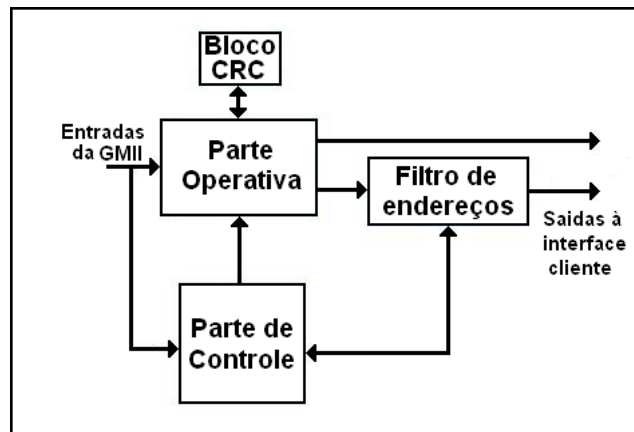


Figura 6.10: Diagrama de blocos simplificado do módulo Receptor

A seguir é descrito brevemente cada um dos blocos que compõem o módulo.

#### 6.1.2.1.1 Bloco de CRC

Implementa o algoritmo CRC-32 definido na cláusula 3 da norma (IEEE, 2005a). Possui uma LFSR com entrada de dados de 8 bits, utilizada para o computo do valor de CRC de 32 bits. Para a detecção de erro no quadro recebido, o bloco computa continuamente o valor de CRC (a partir dos dados recebidos, incluindo o próprio campo de FCS) e no final da recepção, compara o último valor computado com um valor constante de 32 bits. Se o resultado da comparação difere então é detectado um erro de CRC no quadro recebido.

#### 6.1.2.1.2 Parte Operativa

Esse bloco é agrupado em dois sub-blocos:

- Sub-bloco RxDatapath: Registra os dados vindos da GMII e os encaminha aos outros blocos do módulo. Também possui flip-flops para geração dos sinais de início de quadro (*Start of Frame* ou SOF) e fim de quadro (*End of frame* ou EOF), os quais serão encaminhados à Interface Cliente de 8 bits.
- Sub-bloco RxCounters: Utiliza um contador de 16 bits com saída decodificada para contabilizar os octetos recebidos e controlar a posição dos octetos na filtragem de endereços. Também implementa comparadores para verificar se o comprimento do quadro possui um número inteiro de octetos e se os comprimentos máximo e mínimo do quadro cumprem os requisitos que impõe o padrão. No modo promíscuo são aceitos quadros de tipo Jumbo.

#### 6.1.2.1.3 Filtro de Endereços:

Esse bloco realiza comparações de maneira seqüencial (byte-a-byte) do campo de “Endereço Destino”, para detecção do endereço *broadcast* e para detecção de cinco endereços *unicast* MAC de 48 bits predefinidos. Para isto, são utilizados cinco conjuntos de registradores e comparadores para verificação dos endereços em forma paralela. Utilizando o mesmo principio, o sub-bloco armazena e compara os dados seguintes ao campo de “Endereço Origem”, para detecção de *Tag* VLAN. Também possui uma

lógica seqüencial baseada em tabelas de *Hash* para acelerar a detecção de endereços *Multicast*, verificando somente os primeiros 10 bits dos 48 bits que conformam o endereço.

#### 6.1.2.1.4 Bloco de controle:

Implementa uma máquina de estados com as seguintes características:

- Número de estados: 4.
- Número de transições: 11.
- Número de entradas: 4.
- Número de saídas: 2.
- Ativada na borda de subida do sinal RX\_CLK da GMII.
- Reset assíncrono.

#### 6.1.2.2 Módulo Transmissor

Esse módulo não implementa o protocolo CSMA/CD para a transmissão de quadros Ethernet, mas mantém as funções básicas do modo *full-duplex* e implementa outras funções fora do padrão. As principais características funcionais do módulo são:

- Gera os campos de preâmbulo, SFD e FCS.
- Gera automaticamente o IFG.
- Complementa automaticamente o tamanho míni-modo quadro transmitido (*padding*).
- Funções adicionais fora do padrão: Habilita a transmissão de quadros tipo Jumbo.

A figura 6.11 mostra o diagrama de blocos simplificado do módulo Transmissor.

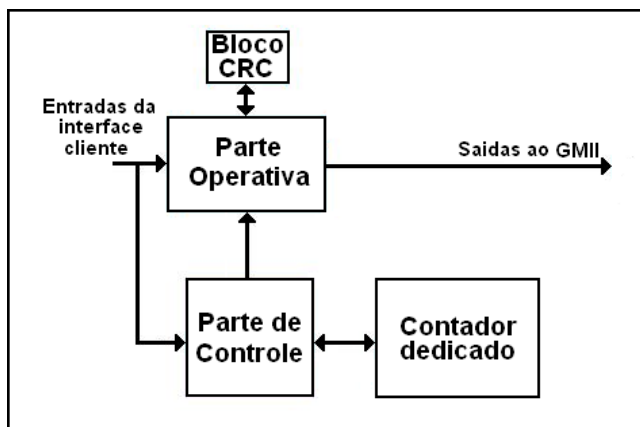


Figura 6.11: Diagrama de blocos simplificado do módulo transmissor

A seguir é descrito brevemente cada um dos blocos que compõem o módulo:

##### 6.1.2.2.1 Bloco de CRC

Implementa o algoritmo CRC-32 definido na cláusula 3 da norma IEEE 802.3 (IEEE, 2005a). Possui uma LFSR com entrada de dados de 8 bits, utilizada para o

computo do valor de CRC de 32 bits a partir dos campos de “Endereço Destino”, “Endereço Origem”, “Tamanho do quadro” e o campo de dados (incluindo os zeros introduzidos no processo de *padding*). O valor de 32 bits gerado é encapsulado no campo de FCS (em 4 bytes), para posteriormente ser transmitido.

#### 6.1.2.2.2 Parte Operativa

Recebe os bytes vindos da Interface Cliente de 8 bits e os encaminha à GMII (conservando o formato). Também possui flip-flops para geração dos sinais de estado da transmissão de quadros.

#### 6.1.2.2.3 Contador Dedicado

Implementa um contador de 16 bits com saídas decodificadas, para contabilizar os octetos transmitidos.

#### 6.1.2.2.4 Bloco de controle

- Implementa uma máquina de estados com as seguintes características:
- Número de estados: 7.
- Número de transições: 17.
- Número de entradas: 7.
- Número de saídas: 7.
- Ativada na borda de subida do sinal TX\_CLK da GMII.
- Reset assíncrono.

#### 6.1.2.3 Módulo de Controle de Fluxo

Esse módulo implementa o mesmo módulo de do Controle de Fluxo descrito no item 6.1.1.3, com a diferença de que o contador de *Slot* é de módulo 4096 (segundo o padrão).

#### 6.1.2.4 Módulo de Estatísticas RX/TX

O módulo armazena num banco de registradores a atividade do Core IP *Soft MAC* 1000. As estatísticas dos eventos de recepção ou transmissão do Core são encapsuladas num vetor que é proporcionado pela interface Cliente de 8 bits. As tabelas 6.5 e 6.6 mostram os vetores de estatística de recepção e transmissão respectivamente.

Tabela 6.5: Definição dos bits do vetor de estatística da recepção no IP *Soft Core* GEMAC

Vetor[21:0]	Nome	Descrição
21	FLOW_CONTROL_FRAME	Ativado se o quadro recebido está livre de erros e é um quadro especial de <i>pause</i> .

Vetor[21:0]	Nome	Descrição
20	VLAN_FRAME	Ativado se o quadro recebido contem um <i>Tag</i> VLAN no campo de tamanho do quadro.
19	RECEIVED_LENGTH_OK	Indica se o quadro recebido possui um comprimento correto segundo o padrão
18	RESERVED	Retorna o valor lógico 0.
[17:7]	FRAME_LENGTH_COUNT	Comprimento do quadro recebido em número de bytes. O contador chega até 2047 para qualquer quadro <i>Jumbo</i> maior que esse valor.
6	RESERVED	Retorna o valor lógico 0.
5	MULTICAST_FRAME	Ativada se o quadro recebido contem um endereço <i>multicast</i> no campo de endereço destino.
4	BROADCAST_FRAME	Ativada se o quadro recebido contem um endereço <i>broadcast</i> no campo de endereço destino.
3	UNICAST_FRAME	Ativada se o quadro recebido contem um dos cinco endereços MAC no campo de endereço destino.
2	FCS_ERROR	Ativada se o quadro recebido contem erro de CRC.
1	BAD_FRAME	Ativada se o quadro recebido contém erros.
0	GOOD_FRAME	Ativada se o quadro recebido está livre de erros.

Tabela 6.6: Definição dos bits do vetor de estatística da transmissão no IP *Soft Core* GEMAC

Vetor[11:0]	Nome	Descrição
11	PAUSE_FRAME_TX	Ativado se o quadro transmitido é um quadro especial de <i>pause</i> .
[10:0]	FRAME_LENGTH_COUNT	Comprimento do quadro transmitido em número de bytes. O contador chega até 2047 para qualquer quadro <i>Jumbo</i> maior que esse valor.

Na continuação, são descritas as principais características funcionais do módulo:

- Estatísticas RX: Indica se o quadro recebido é do tipo: *pause*, *unicast*, *broadcast*, *multicast*. Também indica o tamanho do quadro recebido e se está livre de erros.
- Estatísticas TX: Indica o tamanho do quadro transmitido e se é do tipo *pause*.

### 6.1.2.5 Módulo RS

O módulo adapta os sinais da GMII, que posteriormente são utilizadas pelos módulos de recepção e transmissão. Na continuação, são descritas as principais características funcionais do módulo:

- Implementa o sub-nível de reconciliação da GMII, para transferências de dados em formato de bytes com uma frequência de operação de 125MHz (1000 Mbps).
- Não implementa as sinais utilizadas no modo *half-duplex* (CRS e COL)
- Funções adicionais fora do padrão: Opção de *loopback* na GMII para validação funcional do Core.

### 6.1.2.6 Interface de Host

Essa interface permite a configuração do Core através de um banco de registradores que armazena os parâmetros de implementação 1000 Mbps, os quais são definidos na cláusula 4 da norma IEEE 802.3 (IEEE, 2005a) e mostrados na tabela 6.7.

Tabela 6.7: Vetores de configuração do IP *Soft Core* GEMAC

Vetor	Valor	Descrição
MINFL[15:0]	0x40	Indica o tamanho mínimo do quadro (64 bytes).
MAXFL[15:0]	0x600	Indica o tamanho máximo do quadro (1536 bytes, considerando espaço adicional para os <i>Tags</i> VLAN).
HASH_REG0[31:0]	(*)	Tabela#1 de <i>Hash</i> para detecção do quadro <i>Multicast</i> .
HASH_REG1[31:0]	(*)	Tabela#2 de <i>Hash</i> para detecção do quadro <i>Multicast</i> .
IFG[6:0]	0x0B	Indica o valor do <i>Inter-Frame Gap</i> (0.096 us/1000Mbps).
TX_PAUSE[13:0]	(**)	Indica o valor do tempo de pausa utilizado no processo de controle de fluxo.
PHY_ADDRESS[4:0]	(**)	Endereço do PHY utilizado no protocolo MDIO.
REG_ADDRESS[4:0]	(***)	Endereço do registrador do PHY que se quer acessar pelo protocolo MDIO.

(\*) O valor certo na tabela de *Hash* é igual ao valor do endereço *multicast* que se quer verificar.

(\*\*) O valor do tempo de pausa é escolhido pelo Cliente.

(\*\*\*) O valor do registrador que se quer acessar vai depender do STA

### 6.1.2.7 Interface MAC Client

A função principal desta interface é a comunicação com o Cliente localizado nos níveis superiores do RM-OSI. A tabela 6.8 mostra as sinais da Interface Cliente de 8 bits para recepção e transmissão de quadros Ethernet entre o sub-nível MAC e o Cliente.

Tabela 6.8: Sinais da Interface Cliente de 8 bits do Core

Sinal	Direção	Descrição
GEMAC_CLIENT_RX_CLK	Saída	Relógio para recepção de quadros no Cliente.
GEMAC_CLIENT_RXD[7:0]	Saída	Dados do quadro recebido pelo Core em formato de bytes.
GEMAC_CLIENT_RX_SOF	Saída	Indica o início da recepção de um quadro no Cliente.
GEMAC_CLIENT_RX_EOF	Saída	Indica o fim da recepção de um quadro no Cliente
GEMAC_CLIENT_RX_ABORT	Saída	Indica que o quadro recebido pelo Core contém erros.
RX_STATISTICS_VECTOR[21:0]	Saída	É o vetor de estatística da recepção do Core.
RX_STATISTICS_VALID	Saída	É um sinal ativado durante um ciclo do relógio RX_CLK da GMII ao final da recepção do quadro.
GEMAC_CLIENT_TX_CLK	Saída	Relógio para transmissão de quadros pelo Cliente.
GEMAC_CLIENT_TXD[7:0]	Entrada	Dados a serem transmitidos em formato de bytes pelo Cliente.
GEMAC_CLIENT_TX_SOF	Entrada	Indica o início da transmissão de um quadro pelo Cliente.
GEMAC_CLIENT_TX_EOF	Entrada	Indica o fim da transmissão de um quadro pelo Cliente
GEMAC_CLIENT_TX_ACK	Saída	Indica ao Cliente iniciar a transmissão do segundo byte do quadro (o primeiro byte do quadro é colocado pelo sinal de GEMAC_CLIENT_TX_SOF).
GEMAC_CLIENT_TX_DONE	Saída	Indica que o quadro foi corretamente transmitido pelo Core.
GEMAC_CLIENT_TX_ABORT	Saída	Indica que o quadro foi transmitido com erro pelo Core.
TX_STATISTICS_VECTOR[11:0]	Saída	É o vetor de estatística da transmissão do Core.
TX_STATISTICS_VALID	Saída	É um sinal ativado durante um ciclo do relógio TX_CLK da GMII ao final da recepção do quadro.



Sinal	Direção	Descrição
TX_PAUSE_REQ	Entrada	Requisição do Cliente para a transmissão de um quadro especial de <i>pause</i> .
WILLSEND_CONTROLFRAME	Saída	Indica que o Core está enviando um quadro especial de <i>pause</i> (para diferenciar de uma transmissão normal).

Os relógios de recepção e transmissão de quadros no lado do Cliente utilizam os mesmos relógios da GMII. Por esse motivo, a frequência dos relógios HOST\_RX\_CLK e HOST\_TX\_CLK é 125MHz. Os diagramas temporais para os processos de recepção e transmissão típicos no lado do Cliente são equivalentes aos descritos na subsecção 6.1.1.7 do presente trabalho.

### 6.1.3 IP *Soft Core* MIIM

O IP *Soft Core* MIIM implementa em HW o STA (descrito no ítem 2.2.4), que é encarregada da gerencia do chip PHY. Como é mostrado Na figura 6.12, o MIIM realiza a função do STA do modelo OSI. Na cláusula 22 da norma IEEE 802.3 (IEEE, 2005a), especifica-se que o mecanismo para o gerenciamento de dispositivos de nível físico é o mesmo em sistemas *Fast* e *Gigabit* Ethernet.

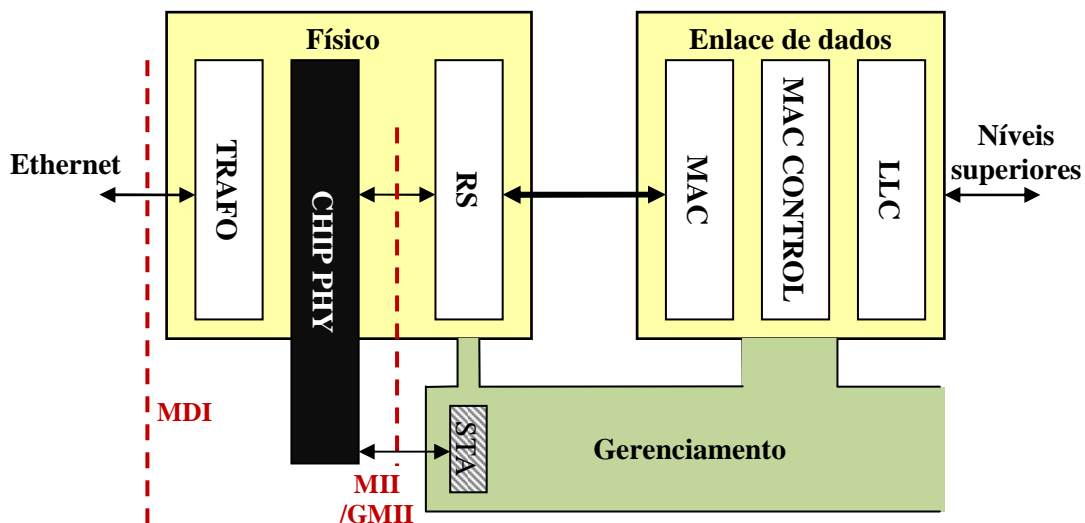


Figura 6.12: IP *Soft Core* MIIM e o modelo OSI

Na figura 6.13 se mostra o diagrama de blocos simplificado do MIIM.

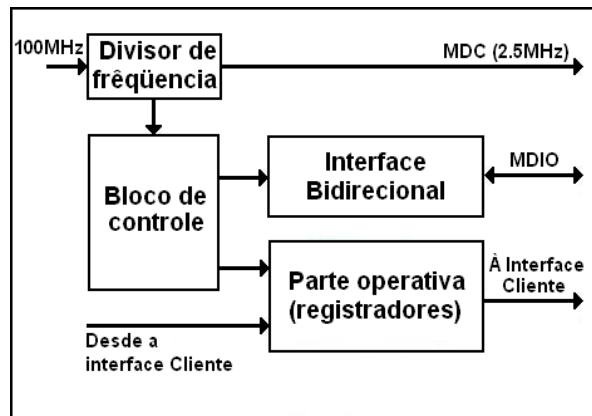


Figura 6.13: Diagrama de blocos simplificado do MIIM

A seguir é efetuada uma breve descrição dos blocos que compõem o módulo:

#### 6.1.3.1 Bloco Divisor de frequência.

É um divisor de frequência de módulo 40. O sinal de relógio fonte possui uma frequência de 100MHz e é gerado por um oscilador externo da placa de desenvolvimento utilizada no presente trabalho (AVNET, 2003). O resultado da divisão de frequência é o sinal MDC com uma frequência estável de 2.5 MHz.

#### 6.1.3.2 Parte Operativa.

Possui um registrador de deslocamento de 8 bits para a serialização e desserialização dos dados. Este registrador opera de maneira conjunta com circuitos combinacionais e um registrador de armazenamento paralelo de 8 bits. Juntos realizam as seguintes operações:

1. Na operação de escrita, transmite byte por byte os 32 bits do protocolo MDIO especificados na tabela 1.2 da subseção 1.3.1 do presente trabalho.
2. Na operação de leitura, transmite byte por byte os primeiros 14 bits (na realidade envia 16 bits mais só são aceitos 14 bits), para em seguida receber os outros 16 bits com informações do PHY.

#### 6.1.3.3 Bloco de Controle.

Implementa um contador de módulo 32 para o controle da comunicação serial no protocolo MDIO. Também é responsável pela habilitação dos processos de serialização de dados (escrita) e desserialização dos dados (leitura).

#### 6.1.3.4 Interface Bidirecional.

Sua função principal é colocar em alta impedância o canal MDIO quando o módulo está no estado de *idle* e no primeiro bit de *Turn-around* do processo de leitura. A interface é implementada com *flip-flops* e um *buffer tri-state* para o controle de acesso ao canal MDIO.

### 6.1.4 IP Soft Core Packet FIFO

Esse módulo cumpre a função de Cliente do sub-nível MAC. Sua função principal é utilizar o princípio de *First-In-First-Out* para o armazenamento temporário dos quadros Ethernet transferidos pela interface MAC. A figura 6.14 apresenta os módulos básicos do IP Soft Core Packet FIFO.

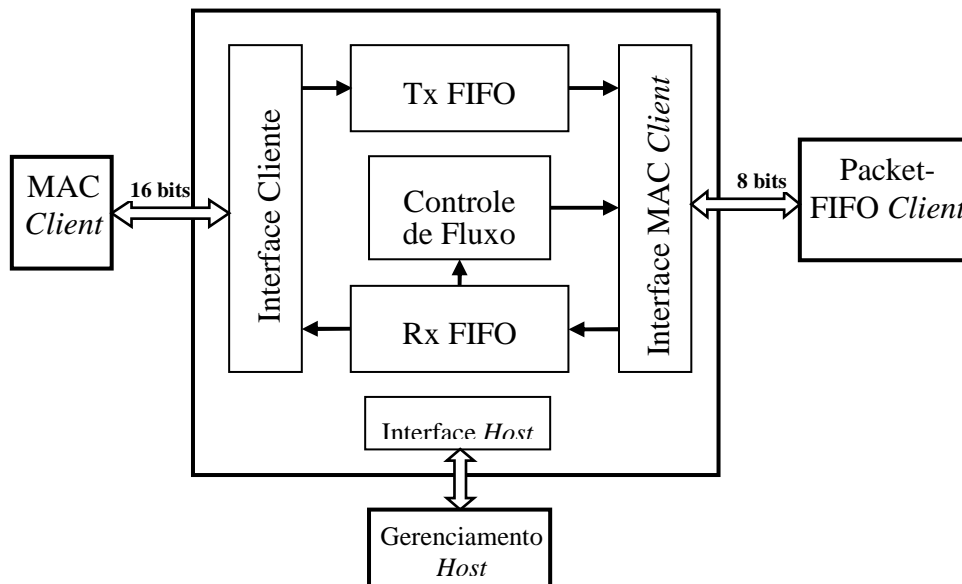


Figura 6.14: Módulos do IP Soft Core Packet-FIFO

As principais características funcionais do módulo são:

- É composto por dois módulos totalmente independentes (Rx FIFO e Tx FIFO), para suportar transferências de dados no modo Full-duplex. Ver figura 6.14.
- Funcionamento assíncrono: Apresenta domínios de relógio diferentes, tanto para escrita como para leitura de dados.
- Possui duas memórias RAM com duas portas de tamanho parametrizável e largura de palavra de 16 bits internas ao FPGA.
- É compatível com a Interface MAC Client dos IP Soft Cores FEMAC e GEMAC (mediante o módulo Interface MAC da figura 6.14).
- Oferece uma interface cliente (módulo Interface Cliente da figura 6.14) de 16 bits para os níveis superiores do RM-OSI.
- Utiliza uma lógica de descritores (implementada tanto no módulo Tx FIFO quanto no módulo Rx FIFO da figura 6.14) para gerenciar adequadamente o armazenamento dos quadros na FIFO. Os descritores guardam informações vitais do quadro, tais como: tamanho do quadro, quadro completo ou incompleto e sinalização de erros.
- Possui um mecanismo de controle de fluxo (complemento IEEE802.3x).
- Possui um mecanismo de controle capaz de realizar auto-descarte de quadros nos casos de *Overrun* de memória e erro de CRC.

- Sinais de controle para gerenciar o preenchimento da FIFO, mediante o módulo da Interface *Host* da figura 6.14.

Salienta-se que funcionalidade assíncrona implementada no IP *Soft Core* Packet-FIFO prevê prováveis problemas de meta-estabilidade na passagem de dados entre domínios de relógios diferentes (frequência e fase). Por isto são seguidas as recomendações de (Cummings, 2001) (Cummings, 2002), os quais citam uma solução específica utilizando códigos Gray nos ponteiros de escrita e leitura das FIFOs.

### 6.1.5 IP *Soft Core* OAM Ethernet

A entidade OAM é definida na cláusula 57 da norma IEEE 802.3 (IEEE, 2005a). A area sombreada da figura 6.15 apresenta o IP *Soft Core* Ethernet no contexto do modelo OSI.

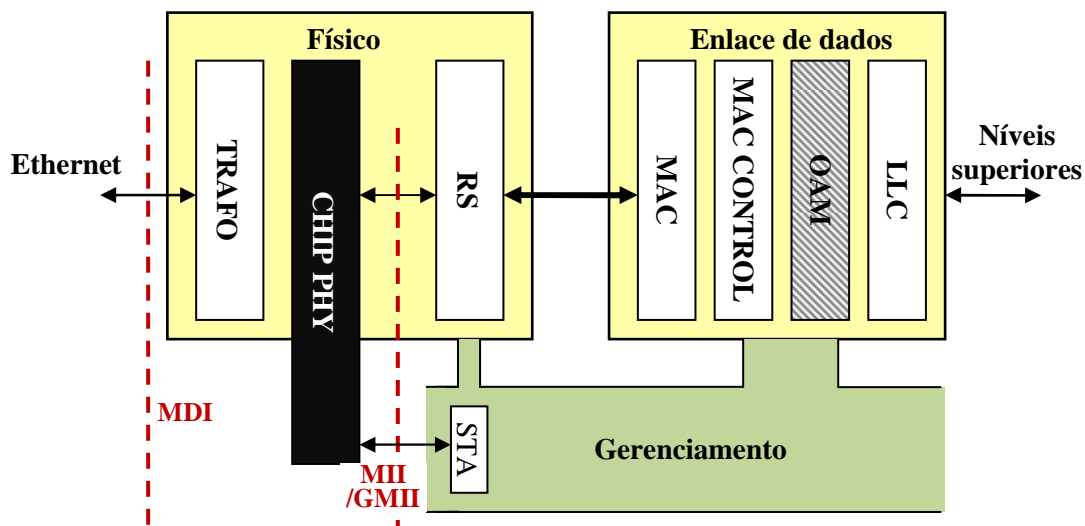


Figura 6.15: IP Módulos funcionais do *Soft Core* OAM Ethernet e o modelo OSI

Visando uma implementação em hardware, verificou-se a existência de um caminho crítico entre a camada de mais baixo nível, as entidades *Multiplexer* e *Parser* do OAM e o *MAC Client*. Ou seja, a implementação dessas entidades em software poderia resultar na inserção de um gargalo de comunicação, já que dificilmente uma implementação em software conseguiria comunicar a um *throughput* que pode chegar a 1Gbits/s. Sendo assim, a implementação em módulos hardware do *Multiplexer* e do *Parser* torna-se essencial.

A figura 6.16 apresenta os módulos projetados no IP *Soft Core* OAM Ethernet. Salienta-se que *OAM Client* realiza funções do *OAM Control* definido na seção 2.3.1.3, e foi implementado numa rotina de Software utilizando o processador embarcado Microblaze. As funções que realiza *OAM Client* são: *OAM Discovery* e *Transmit*, este último controla algumas regras para geração de OAMPDUs.



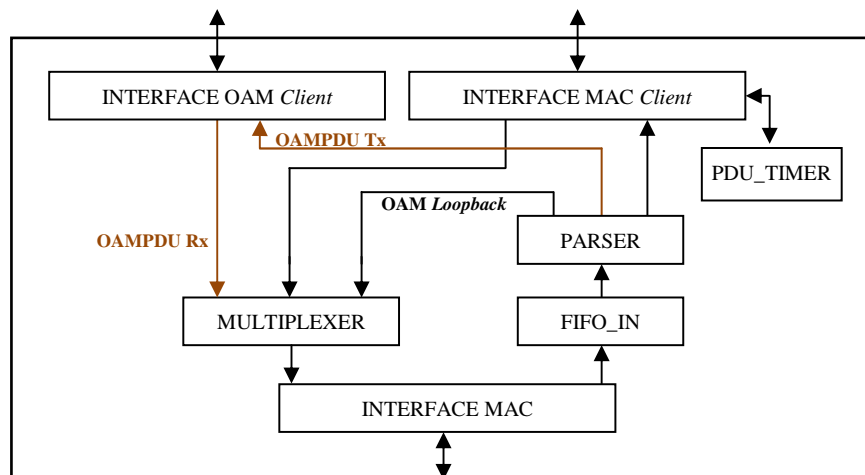


Figura 6.16: Módulos funcionais do IP *Soft Core* OAM Ethernet

#### 6.1.5.1 Módulo Parser

Decodifica pacotes recebidos do MAC e os repassa para o OAM *Client* se o pacote for do tipo OAMPDU, para o MAC *Client* se o pacote não for do tipo OAMPDU, e para o sub-módulo MULTIPLEXER se a entidade estiver em modo *Loopback*. Para realizar a decodificação, o pacote é colocado em uma FIFO (FIFO\_IN) para sincronizar o campo onde a informação de tipo de pacote é exibida

O projeto do módulo *Parser* foi baseado na máquina de estados definida na cláusula 57 da norma IEEE 802.3, e apresentada na figura 2.12.

#### 6.1.5.2 Módulo Multiplexer

Responsável por multiplexar os dados do pacote a ser enviado para um dispositivo remoto. Ele verifica a indicação de envio de OAMPDUs, pacotes do MAC *Client* ou pacotes *loopback*, e procede com o envio do pacote para a rede

O projeto do módulo *Multiplexer* foi baseado na máquina de estados definida na cláusula 57 da norma IEEE 802.3, e apresentada na figura 2.13.

#### 6.1.5.3 PDU Timer

Temporizador utilizado para contabilizar o envio de um número máximo de OAMPDUs por segundo e emitir ao menos um OAMPDU por segundo.

#### 6.1.5.4 Interface OAM Client

Serve para realizar a comunicação entre o IP *Soft Core* OAM e o OAM *Client*. Esta comunicação é realizada através de 5 registradores de 32 bits cada e duas FIFOs (uma de leitura – FIFO\_RD, e uma de escrita – FIFO\_WR). Os registradores são usados para mapear os sinais de comunicação entre o IP *Soft Core* OAM e o OAM *Client*, e as FIFOs são utilizadas para armazenar o fluxo de dados na transmissão e recebimento do pacote para o OAM *Client*.

### 6.1.5.5 Interface MAC Client e Interface MAC

Estas interfaces têm como objetivo fazer com que o IP *Soft Core* OAM Ethernet seja transparente na comunicação entre o IP *Soft Core* GEMAC e seu cliente.

### 6.1.6 IP Hard/Soft Core PHY1000X

Este Core é responsável pela implementação em hardware dos sub-níveis físicos PCS e PMA que compõem o PHY no sistema de mídia 1000BASE-X (modo *duplex* forçado) segundo as cláusulas 36 e 37 da norma IEEE 802.3 (IEEE, 2005a). As funcionalidades do Core estão organizadas em 6 módulos. Ver figura 6.17.

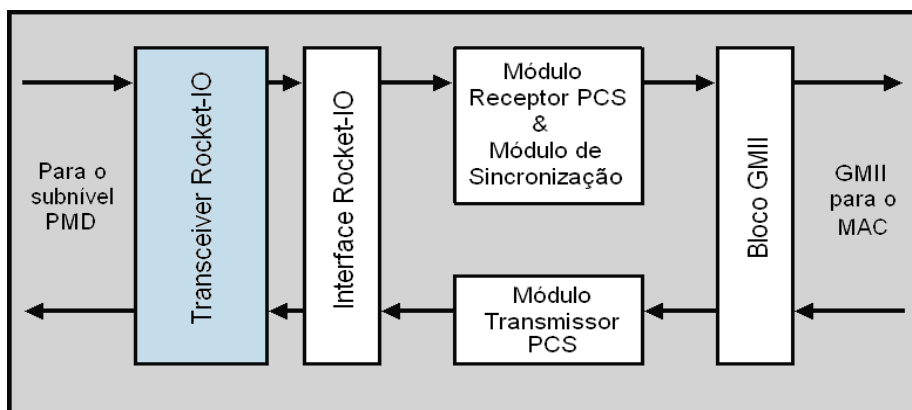


Figura 6.17: Diagrama de Blocos funcional do IP Hard/Soft Core PHY1000X

#### 6.1.6.1 Módulo Transceiver Rocket-IO

Esse módulo está imerso no silício no modo de *Hard IP Core* na família de dispositivos FPGA Virtex-II Pro da Xilinx. Na continuação, são descritas as principais características funcionais do *transceiver*:

- Implementa o codificador/decodificador 8B/10B do sub-nível PCS.
- Implementa as principais funções do sub-nível PMA: serialização/deserialização de dados, buffers elásticos e os circuitos necessários para recuperação do relógio. É compatível com dispositivos *transceivers* ópticos externos tais como: GBIC (*gigabit interface converter*) ou SPF (*Small Form-factor Pluggable*) que suportam o sub-nível PMD.

#### 6.1.6.2 Interface Rocket-IO

Fornece os sinais de relógio que são utilizados no *transceiver* Rocket-IO (125 MHz e 62.5 Mhz), fazendo uso de um DCM (*Digital Clock Manangment*) da Xilinx. O sinal de relógio fonte possui uma frequência de 125MHz e é gerado de maneira diferencial por um oscilador externo da placa de desenvolvimento AVNET (AVNET, 2003).

#### 6.1.6.3 Módulo Receptor PCS

Esse módulo converte a seqüência de códigos recebidos pelo decodificador 8B/10B em octetos da GMII segundo a máquina de estados representada nas figuras 36-7a e 36-7b da norma IEEE 802.3 (IEEE, 2005a) e mostrada nas figuras 6.18 e 6.19

respectivamente. Nestas figuras observe-se que foram marcadas (com circunferências) as funções não implementadas pelo módulo: recepção no modo half-duplex e comunicação com o dispositivo de *Auto-negotiation*. Esta escolha é justificada na especificação da arquitetura do EDD proposto.

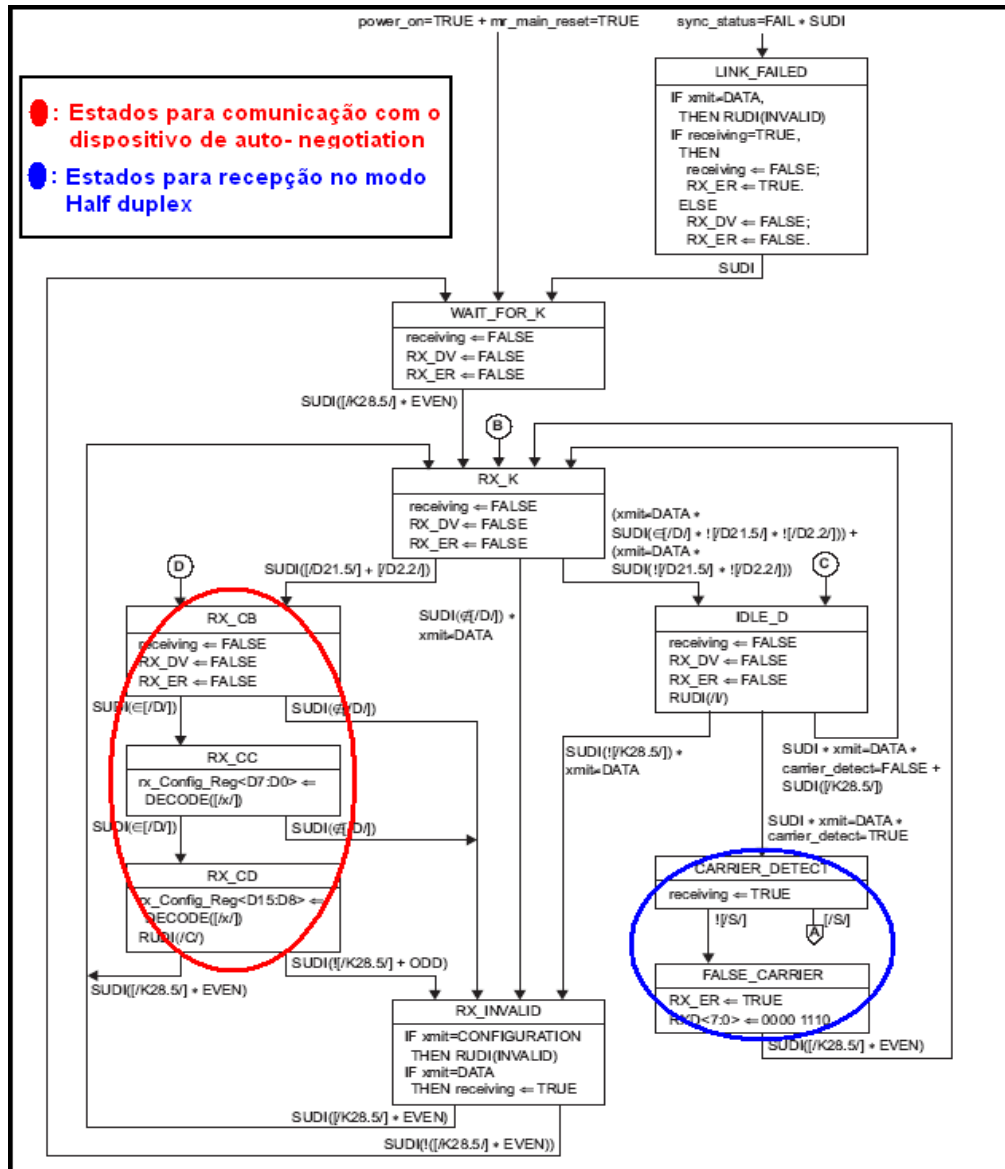


Figura 6.18: Máquina de estados do Receptor PCS, parte a (IEEE, 2005a)

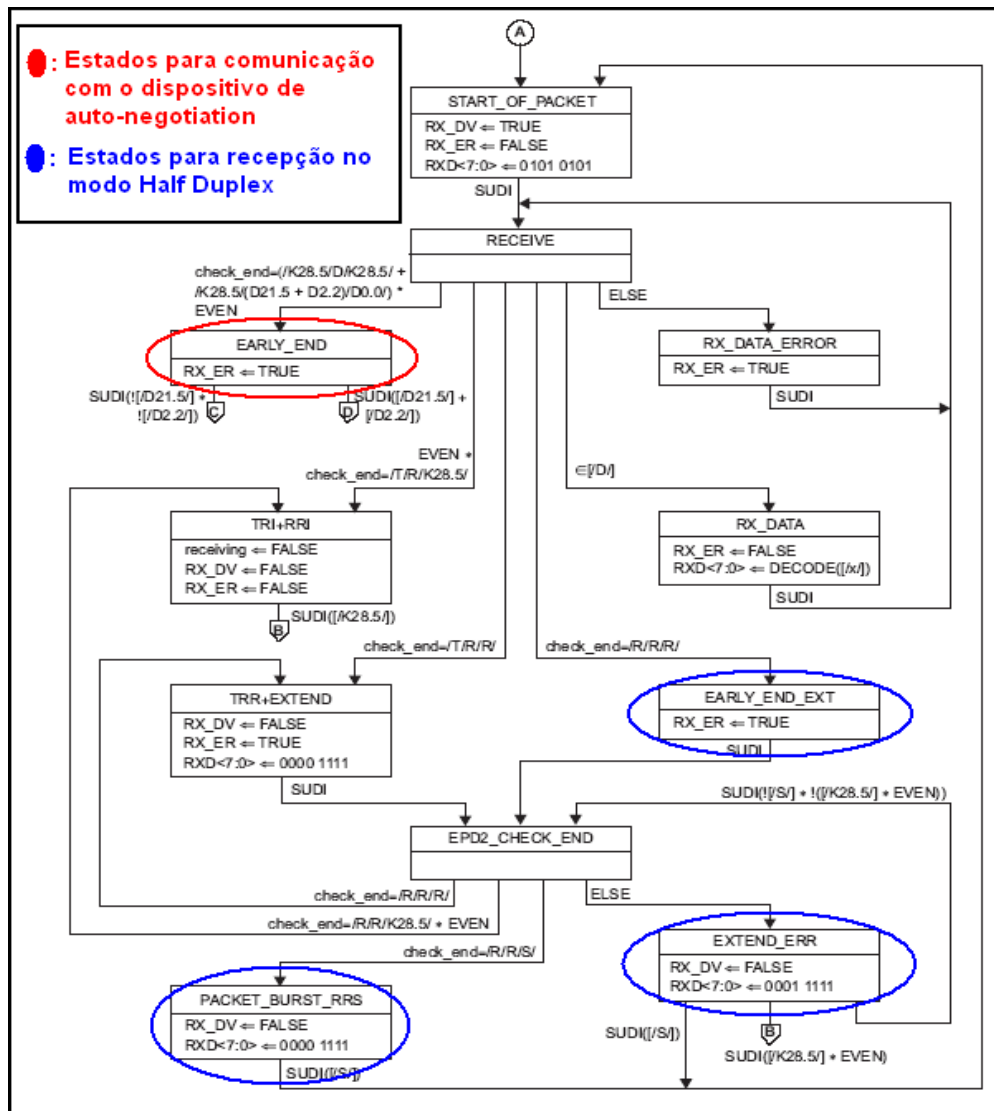


Figura 6.19: Máquina de estados do Receptor PCS, parte b (IEEE, 2005a)

#### 6.1.6.4 Módulo de Sincronização

O módulo de sincronização verifica se o meio físico está pronto para uma adequada recepção dos dados. Implementa de forma completa a máquina de estados representada na norma IEEE 802.3 (IEEE, 2005a)

#### 6.1.6.5 Módulo Transmissor PCS

Esse módulo converte a seqüência de octetos da GMII em símbolos para a codificação 8B/10B segundo as máquinas de estados representadas nas figuras 36-5 e 36-6 da norma IEEE 802.3 (IEEE, 2005a). O módulo não implementa os estados que representam as seguintes funções: transmissão no modo *Half-duplex*, mecanismo de corrupção de dados e comunicação com o dispositivo de *Auto-negotiation*. Esta escolha é justificada na especificação da arquitetura do EDD proposto. Nas figuras 6.20 e 6.21 são representados os estados não implementados pelo módulo (marcadas com circunferências).



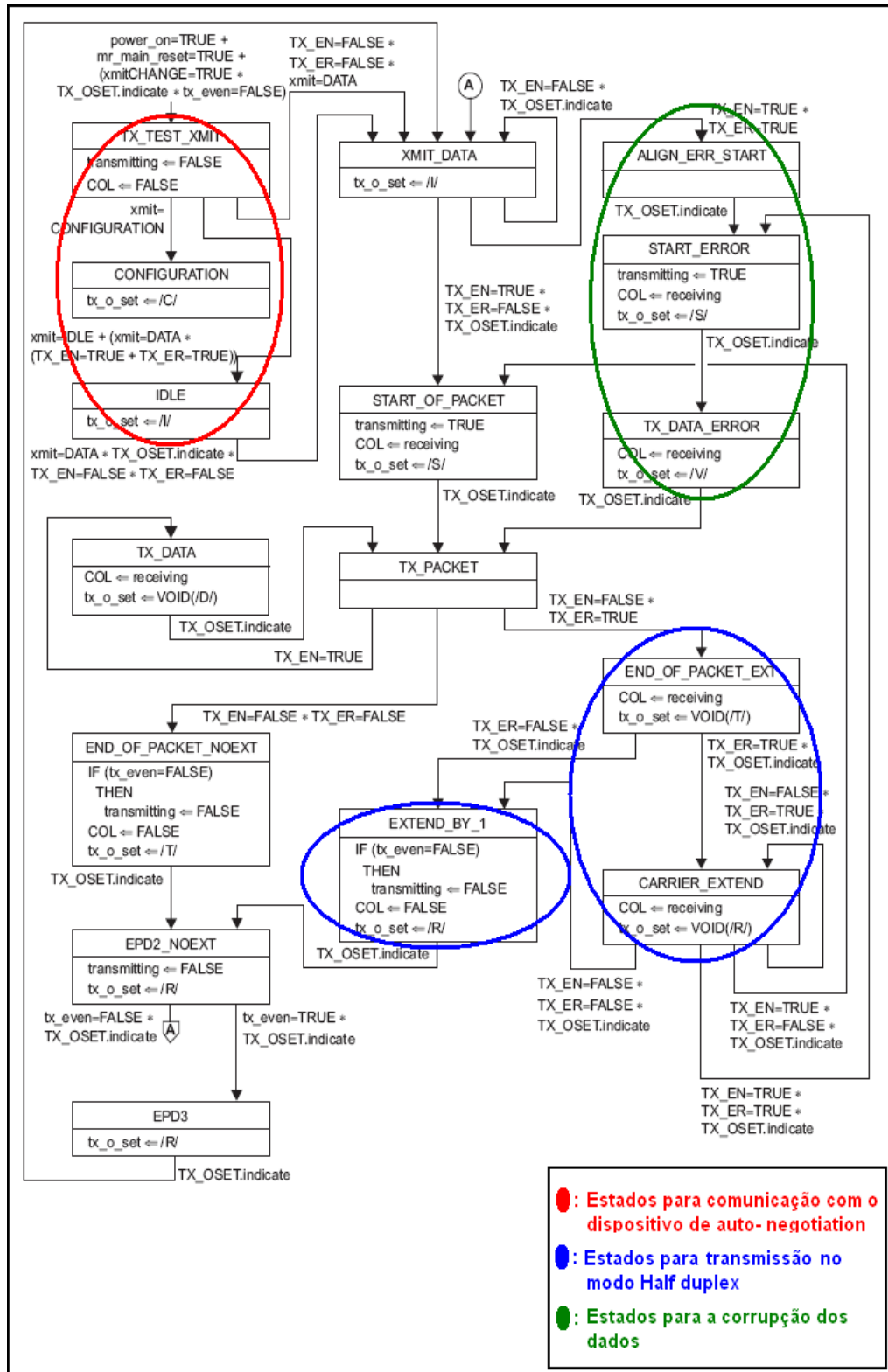


Figura 6.20: Máquina de estados de *ordered\_set* do transmissor PCS (IEEE, 2005a)

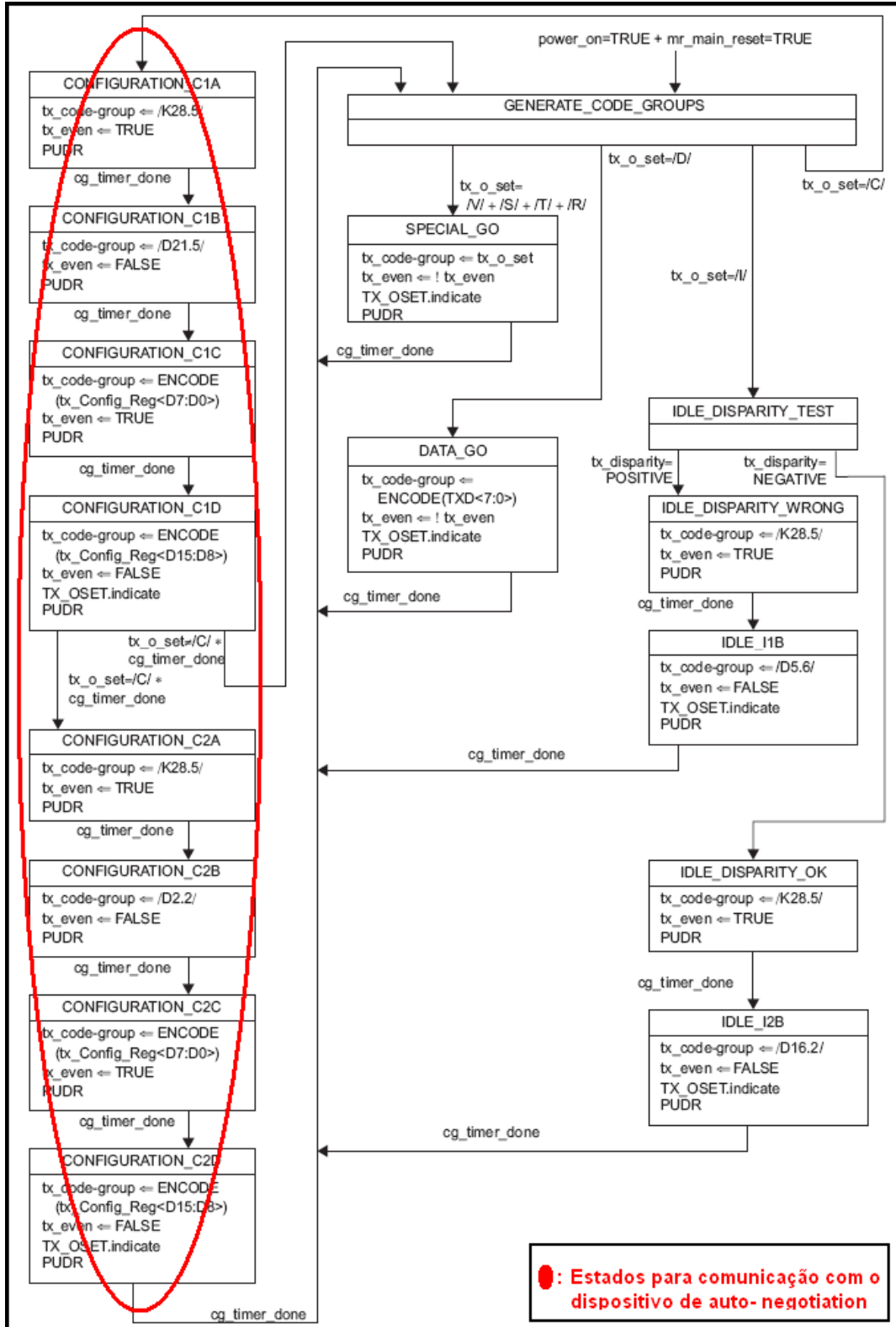


Figura 6.21: Máquina de estados de *code-group* do transmissor PCS (IEEE, 2005a)

### 6.1.6.6 Modulo GMII

Esse módulo fornece o lado cliente da GMII. Isto obriga ao Core trabalhar com uma frequência de operação de 125 MHz.

## 6.2 Implementação dos módulos IP

Foi realizada a síntese dos módulos IP desenvolvidos no presente trabalho sobre um dispositivo reconfigurável FPGA Virtex-II Pro XC2VP30. As ferramentas de síntese utilizadas foram: XST (*Xilinx Synthesis Tool*) do ISE Foundation 7.1i da XILINX e Synplify Pro 8.1. A seguir, são apresentadas em tabelas os resultados das implementações.

### 6.2.1 Resultados da implementação dos IP *Soft Cores* FEMAC e GEMAC

A tabela 6.9 apresenta o consumo de recursos e a frequência máxima obtidas na síntese dos IP *Soft Cores* FEMAC e GEMAC. A frequência de operação do IP *Soft Core* FEMAC é 25MHz (segundo o padrão), por esse motivo, o critério de otimização no XST foi visando “área”, e no *Synplify* foi usado um *constraint* de frequência de 25MHz. Da mesma forma, sabendo que a frequência de operação do IP *Soft Core* GEMAC é 125 MHz (segundo o padrão), o critério de otimização no XST foi visando “velocidade” e no *Synplify* foi usado um *constraint* de frequência de 125MHz.

Tabela 6.9: Resultados da síntese dos Cores FEMAC e GEMAC

		XST	Synplify
IP Soft Core FEMAC	LUTs	918	981
	Flip Flops	507	337
	Fmax (MHz)	88	92.3
IP Soft Core GEMAC	LUTs	741	746
	Flip Flops	399	377
	Fmax (MHz)	147.4	128.3

Também foi executado o fluxo de projeto ASIC para os módulos FEMAC e GEMAC. Para isto utilizou-se o EDA Cadence e o Design Kit XL035 da XFAB. O Design kit estabelece as condições de contorno das células estandares, sendo que elas estão projetadas para trabalhar com uma tensão de 3.3V, baixa potência (L – Low Power) e na tecnologia CMOS 0,35um. A partir da síntese lógica, foi possível construir a tabela 6.10. Existem duas entradas verticais na tabela: Ambas entradas são referentes às condições típicas de operação do design kit, que considera a tensão de 3,3V e a temperatura de 25°C. A diferença está no esforço aplicado para fazer a síntese lógica, que foi alto para a primeira entrada e baixo para a segunda. Utilizando o esforço alto, podemos ver significativas melhoras nos parâmetros de consumo, área e número de células utilizadas.

Tabela 6.10: Parâmetros de área obtidos após Síntese lógica dos Módulos FEMAC e GEMAC

	Esforço para Síntese Lógica	Alto	Baixo
<b>IP Soft Core FEMAC</b>	<b>Área das Células (<math>\mu\text{m}^2</math>)</b>	700876	773091
	<b>Área das Nets (<math>\mu\text{m}^2</math>)</b>	253151	306907
	<b>Número de Células</b>	4805	5940
<b>IP Soft Core GEMAC</b>	<b>Área das Células (<math>\mu\text{m}^2</math>)</b>	665039	714835
	<b>Área das Nets (<math>\mu\text{m}^2</math>)</b>	243392	297012
	<b>Número de Células</b>	5176	6450

Um detalhe relevante obtido após a síntese física, é que a o bloco Interface Host ocupa aproximadamente 60% do Módulo FEMAC ou GEMAC. Isto se deve ao grande número de registradores ali descritos e aos circuitos de passagem de domínio de relógio Host (100MHz) para domínio de *clock* do MAC (125MHz para GEMAC e 2.5/25MHz para FEMAC)

### 6.2.2 Resultados da implementação do IP *Soft Core* MIIM

A tabela 6.11 mostra uma comparação de desempenho entre o IP *Soft Core* MIIM apresentado na subsecção 6.1.3 do presente trabalho e o Core MIIM desenvolvido em Verilog por (MOHOR, 2002). Ambos implementam as mesmas funções segundo a cláusula 22 da norma IEEE 802.3 (IEEE, 2005a). O protocolo MDIO implementado pelo *Soft Core* trabalha em baixas frequências, já que é limitado pelo sinal MDC de 2.5MHz. Por tanto, o critério de otimização no XST foi visando “área” e no *Synplify* foi usado um *constraint* de frequência de 100MHz. A comparação mostra que a síntese efetuada por nós apresenta um menor consumo de área.

Tabela 6.11: Desempenho do Soft Core MIIM

		XST	Synplify
<b>Soft Core MIIM</b>	<b>LUTs</b>	91	66
	<b>Flip Flops</b>	75	42
	<b>Fmax (MHz)</b>	190.80	139.7
<b>OpenCores MIIM (MOHOR, 2002)</b>	<b>LUTs</b>	146	82
	<b>Flip Flops</b>	75	72
	<b>Fmax (MHz)</b>	225.42	151.2

### 6.2.3 Implementação do IP *Soft Core* Packet FIFO

O módulo *Packet FIFO* necessita atingir o requisito de frequência de 125 MHz para manter a compatibilidade com *Soft Cores* FEMAC e GEMAC, portanto, o critério de otimização no XST foi visando “velocidade”. A tabela 6.12 mostra os resultados da síntese dos sub-módulos RxFIFO e TxFIFO modificando o tamanho da FIFO (*FIFO Depth*).

Tabela 6.12: Resultados da síntese dos sub-módulos que compõem o PacketFIFO

	<b>FIFO Depth</b>	<b>LUTs</b>	<b>Flips Flops</b>	<b>Fmax (MHz)</b>	<b>Blocks RAM</b>
<b>Rx FIFO</b>	128	318	167	153.5	1
	256	497	235	146.3	1
	512	707	368	142.9	1
<b>Tx FIFO</b>	128	285	128	156.5	1
	256	386	196	156.4	1
	512	576	322	153.6	1

### 6.2.4 Implementação do IP *Soft Core* OAM Ethernet

Como descrito no item 6.1.5, a implementação do IP *Soft Core* OAM Ethernet foi realizado parte em SW e parte em HW. As funções OAM *Discovery* e *Transmit* foram implementadas em software (no processador Microblaze), sendo que apenas o timer que uma delas utiliza foi implementada em hardware (*PDU Timer*). A implementação em software dessas funcionalidades não reduz sua capacidade, pelo contrário, facilita a comunicação com o OAM *Client*, já que essas duas entidades fazem parte da interface de comunicação entre o IP *Soft Core* OAM Ethernet e o OAM *Client*. Além disso, a implementação da função *Transmit* em hardware iria gerar uma ocupação de área considerável devido a complexidade das regras de geração de OAMPDUs definidos na norma IEEE802.3. A figura 6.22 apresenta a implementação do IP *Soft Core* OAM Ethernet trabalhando conjuntamente com o procesador Microblaze.

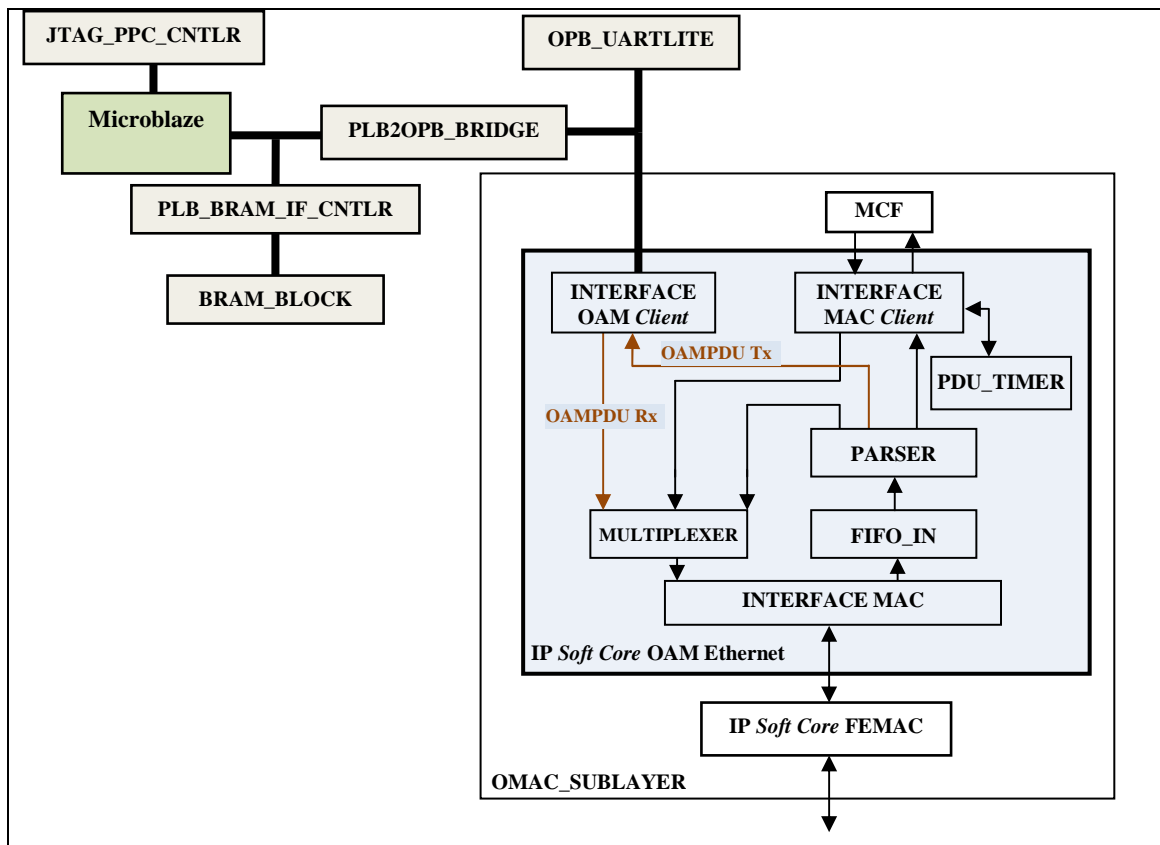


Figura 6.22: IP Soft Core OAM Ethernet e sua integração com o processador MicroBlaze

#### 6.2.4.1 Sinais de interface do IP Soft Core OAM Ethernet

A tabela 6.13 mostra os sinais que formam a interface entre o IP Soft Core OAM Ethernet e as entidades MAC, OAM\_Client e MAC Client.

Tabela 6.13: Sinais do módulo IP Soft Core OAM Ethernet

Sinal	Direção	Responsável pela geração	Descrição
MAC_RX_CLK	Entrada	DCM	Relógio para recepção de quadros vindos do MAC
RST	Entrada	OAM_CLIENT	Reset do OAM_SUBLAYER
MAC_RXD[7:0]	Entrada	MAC	Canal que recebe os dados em formato de bytes vindo do MAC
MAC_RX_SOF	Entrada	MAC	Indica o início do recebimento de um quadro vindo do MAC
MAC_RX_EOF	Entrada	MAC	Indica o fim do recebimento de um quadro vindo do MAC
MAC_RX_ABORT	Entrada	MAC	Indica que o quadro recebido do MAC contém erros

MAC_TXD[7:0]	Saída	OAM_SUBLAYER	Canal de transmissão de dados em formato de bytes para o MAC
MAC_TX_SOF	Saída	OAM_SUBLAYER	Indica o início da transmissão de um quadro para o MAC
MAC_TX_EOF	Saída	OAM_SUBLAYER	Indica o fim da transmissão de um quadro para o MAC
MAC_TX_ACK	Entrada	MAC	Indica ao OAM_SUBLAYER iniciar a transmissão do 1º byte do quadro
OAM_TXD[7:0]	Entrada	OAM_CLIENT	Canal que recebe os dados transmitidos pelo OAM_CLIENT
OAM_TX_SOF	Entrada	OAM_CLIENT	Indica o início da transmissão de um quadro pelo OAM_CLIENT
OAM_TX_EOF	Entrada	OAM_CLIENT	Indica o fim da transmissão de um quadro pelo OAM_CLIENT
OAM_TX_ACK	Saída	OAM_SUBLAYER	Indica ao OAM_CLIENT iniciar a transmissão do 1º byte do quadro
OAM_RXD[7:0]	Saída	OAM_SUBLAYER	Canal de transmissão de dados para o OAM_CLIENT
OAM_RX_SOF	Saída	OAM_SUBLAYER	Indica o início da transmissão de um quadro para o OAM_CLIENT
OAM_RX_EOF	Saída	OAM_SUBLAYER	Indica o fim da transmissão de um quadro para o OAM_CLIENT
OAM_RX_ACK	Entrada	OAM_CLIENT	Indica ao OAM_SUBLAYER iniciar a transmissão do 1º byte do quadro
MCF_TXD[7:0]	Entrada	MAC_CLIENT	Canal de recebimento de dados em formato de bytes vindo do MAC_CLIENT
MCF_TX_SOF	Entrada	MAC_CLIENT	Indica o início do recebimento de um quadro vindo do MAC_CLIENT
MCF_TX_EOF	Entrada	MAC_CLIENT	Indica o fim do recebimento de um quadro vindo do MAC_CLIENT
MCF_RXD[7:0]	Saída	OAM_SUBLAYER	Canal de transmissão de dados em formato de bytes para o MAC_CLIENT
MCF_RX_SOF	Saída	OAM_SUBLAYER	Indica o início da transmissão de um quadro para o MAC_CLIENT
MCF_RX_EOF	Saída	OAM_SUBLAYER	Indica o fim da transmissão de um quadro para o MAC_CLIENT
local_link_status	Entrada	OAM_CLIENT	Indica o status do link estabilizado. Utilizado pelo submódulo MULTIPLEXER para saber o estado atual do link (OK ou FAIL)

local_mux_action	Entrada	OAM_CLIENT	Governa o fluxo de quadros do MAC no submódulo MULTIPLEXER (FORWARD ou DISCARD)
local_par_action(1:0)	Entrada	OAM_CLIENT	Governa o fluxo de non-OAMPDUs no submódulo PARSER (FORWARD, LOOPBACK ou DISCARD)
local_unidirectional	Entrada	OAM_CLIENT	Indica se o DTE é capaz de enviar OAMPDUs quando o link na direção de recebimento não está operacional
Pdu_timer_init	Entrada	OAM_CLIENT	Inicializa o timer PDU_TIMER
Pdu_timer_done	Saída	OAM_SUBLAYER	Indica que o timer PDU_TIMER expirou
local_lost_link_timer_done	Saída	OAM_SUBLAYER	Indica que timer LINK_TIMER expirou

#### 6.2.4.2 Interface com o OAM Client

Para estabelecer a comunicação com o processador MicroBlaze foi necessário encapsular o IP Soft Core OAM Ethernet num bloco maior chamado de OAMAC\_Sublayer que inclui o IP Soft Core FEMAC como se pode apreciar na figura 6.22.

OAMAC\_Sublayer é instanciado dentro da Interface OPB (*On-chip Peripheral Bus*) fornecido pelo processador MicroBlaze como se fosse um controlador dedicado (User\_logic). O módulo OAMAC\_Sublayer é composto pelos sinais descritos na tabela 6.14.

Tabela 6.14: Sinais do módulo OAMAC\_Subla

Sinal	Direção	Responsável pela geração	Descrição
PHY_RESET	Entrada	OAM_CLIENT	Reset do PHY
DCM_RST	Entrada	OAM_CLIENT	Reset do DCM interno (divisor de frequência)
HOST_RST	Entrada	OAM_CLIENT	Reset do MAC
OAM_RST	Entrada	OAM_CLIENT	Reset do OAM_SUBLAYER
MII_RX_CLK	Entrada	PHY	Relógio para a recepção dos dados com uma frequência nominal de 25 MHZ (100 Mbps)
MII_TX_CLK	Entrada	PHY	Relógio para a transmissão dos dados com uma frequência nominal de 25 MHZ (100 Mbps)
MII_RXD[3:0]	Entrada	PHY	Canal que recebe os dados em formato de nibbles recebidos pelo MII



MII_RX_DV	Entrada	PHY	Indica que um dado válido foi colocado no canal RXD
MII_TXD[3:0]	Saída	MAC	Canal de transmissão de dados em formato de nibbles para a rede
MII_TXEN	Saída	MAC	Habilita a transmissão de dados
MII_TXERR	Saída	MAC	Indica a propagação de um erro na transmissão
OAM_TXD[7:0]	Entrada	OAM_CLIENT	Canal que recebe os dados transmitidos pelo OAM_CLIENT
OAM_TX_SOF	Entrada	OAM_CLIENT	Indica o início da transmissão de um quadro pelo OAM_CLIENT
OAM_TX_EOF	Entrada	OAM_CLIENT	Indica o fim da transmissão de um quadro pelo OAM_CLIENT
OAM_TX_ACK	Saída	OAM_SUBLAYER	Indica ao OAM_CLIENT iniciar a transmissão do 1º byte do quadro
OAM_RXD[7:0]	Saída	OAM_SUBLAYER	Canal de transmissão de dados para o OAM_CLIENT
OAM_RX_SOF	Saída	OAM_SUBLAYER	Indica o início da transmissão de um quadro para o OAM_CLIENT
OAM_RX_EOF	Saída	OAM_SUBLAYER	Indica o fim da transmissão de um quadro para o OAM_CLIENT
OAM_RX_ACK	Entrada	OAM_CLIENT	Indica ao OAM_SUBLAYER iniciar a transmissão do 1º byte do quadro
local_link_status	Entrada	OAM_CLIENT	Indica o status do link estabilizado. Utilizado pelo submódulo MULTIPLEXER para saber o estado atual do link (OK ou FAIL)
local_mux_action	Entrada	OAM_CLIENT	Governa o fluxo de quadros do MAC no submódulo MULTIPLEXER (FORWARD ou DISCARD)
local_par_action(1:0)	Entrada	OAM_CLIENT	Governa o fluxo de non-OAMPDUs no submódulo PARSER (FORWARD, LOOPBACK ou DISCARD)
local_unidirectional	Entrada	OAM_CLIENT	Indica se o DTE é capaz de enviar OAMPDUs quando o link na direção de recebimento não está operacional
Pdu_timer_init	Entrada	OAM_CLIENT	Inicializa o timer PDU_TIMER
Pdu_timer_done	Saída	OAM_SUBLAYER	Indica que o timer PDU_TIMER expirou
local_lost_link_timer_done	Saída	OAM_SUBLAYER	Indica que timer LINK_TIMER expirou

Há 5 registradores da interface no OAM *Client* (como descrito na subsecção 6.1.5.4), os quais são visualizados como um sinal de 32 bits dentro da Interface/Barramento OPB e como um inteiro sem sinal de 32 bits na rotina de Software do OAM *Client*. O mapeamento é realizado bit a bit, segundo é mostrado na tabela 6.15

Tabela 6.15: Mapeamento dos 32 bits na interface OPB

Registro	Bit	Sinais de OAMAC_Sublayer
Slv_reg0	31	oam_rst, host_rst, phy_rst
	30..10	Disponível
	9..8	local_par_action
	7	local_mux_action
	6..4	Disponível
	3	local_link_status
	2	local_unidirectional
	1	Disponível
Slv_reg1	0	oam_rst
	31	pdu_timer_done
	30..1	Disponível
Slv_reg3	0	local_lost_link_timer_done
	31	pdu_timer_init
	30..0	Disponível

### 6.2.5 Resultados da implementação do IP *Hard/Soft Core PCS/PMA 1000x*

A tabela 6.16 mostra uma comparação entre IP *Soft Core PCS/PMA 1000X* apresentado na secção 6.1.6 do presente trabalho e um Core da Xilinx setado para implementar as mesmas funcionalidade do *IP Soft Core PCS/PMA 1000X* (XILINX, 2009). A frequência máxima pós-síntese do Core desenvolvido no presente trabalho é 158.2 MHz, alcançando assim o requisito de velocidade imposta pela GMII (125MHz). Um menor consumo de recursos e uma menor latência do caminho de dados no transmissor e no receptor PCS (Rx PCS e Tx PCS) foram obtidos comparado com Core da Xilinx.

Tabela 6.16: Desempenho do IP *Soft Core Ethernet 1000BASE-X PCS/PMA*

Cores	Slices	LUTs	Flip Flops	Latência do caminho de dados (ciclos)	
				Rx PCS	Tx PCS
<b>IP Soft Core Ethernet 1000BASE-X PCS/PMA</b>	75	123	88	4	2
<b>Xilinx Ethernet 1000BASE-X PCS/PMA 2009(XILINX, 2009)</b>	166	204	168	6	4

### 6.3 Validação dos módulos IP

A validação funcional dos módulos que compõem cada um dos Cores desenvolvidos no presente trabalho foi feita através do uso de *testbenchs*. A ferramenta utilizada para verificação funcional foi o ModelSim e a ferramenta utilizada para avaliar a prototipação no FPGA foi o Chipscope Pro 7.1i. No caso da validação em hardware teve que ser montado um cenário de teste composto pela placa de desenvolvimento (AVNET, 2003) baseado no FPGA Virtex-II Pro XC2VP30, que contem os Cores a serem validados, um switch Metro Ethernet comercial (DATACOM, 2006) e um computador com placa de rede *Fast Ethernet*, como se mostra na figura 6.23.

A idéia principal da proposta de validação hardware é interligar os dispositivos anteriormente mencionados para montar uma rede capaz de replicar os quadros Ethernet que são transferidos desde o computador. Por tanto, será necessário criar um *loop* no dispositivo de destino dos quadros, que neste caso será a placa de desenvolvimento. A interconexão dos dispositivos no cenário de teste é realizada em ligação ponto-a-ponto, por tanto, será feita a validação hardware dos IP *Soft Cores* no modo de operação full-duplex. O modo de operação half-duplex do IP *Soft Core* FEMAC, foi somente validado funcionalmente com o simulador ModelSim.

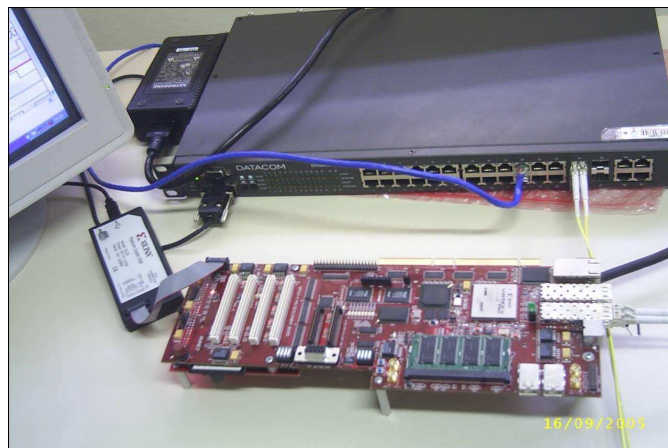


Figura 6.23: Plataforma para a validação hardware dos Cores

#### 6.3.1 Cenário de testes do IP *Soft Core* FEMAC

A figura 6.24 mostra o cenário de testes utilizado para a validação física do IP *Soft Core* FEMAC. O fluxo dos quadros transferidos na rede é o seguinte: Computador → Porta 10/100 #2 do switch → Porta 10/100 #1 do switch → Contector RJ-45 → Chip PHY → Core FEMAC (modo recepção) → Packet FIFO (RxFIFO) → Circuito *Loop-back* → Packet FIFO (TxFIFO) → Core FEMAC (modo transmissão) → Chip PHY → Porta 10/100 #1 do switch → Porta 10/100 #2 do switch → Computador.

No computador foi implementado um software gerador de trafego determinístico, que realiza a função de transmissor *multicast* para a geração de quadros de tamanho fixo do tipo UDP (*User Datagram Protocol*). Esse software utiliza os *sockets* de comunicação da API (*Application Programming Interface*) de Java (ELLIOTTE, 2000). No entanto, existe uma transferência de quadros não determinística gerada pelo próprio computador (ARP, BROWSER, NBNS, IGMP, etc), a qual também será considerada na

validação dos Cores. Os quadros transmitidos (gerados ou não gerados) e os quadros recebidos produto da replica, serão detectados fazendo uso do analisador de protocolos Ethernet (ETHERREAL, 2006). Ao mesmo tempo, é utilizada a ferramenta Chipscope para identificar os possíveis erros e o comportamento das sinais internos no FPGA.

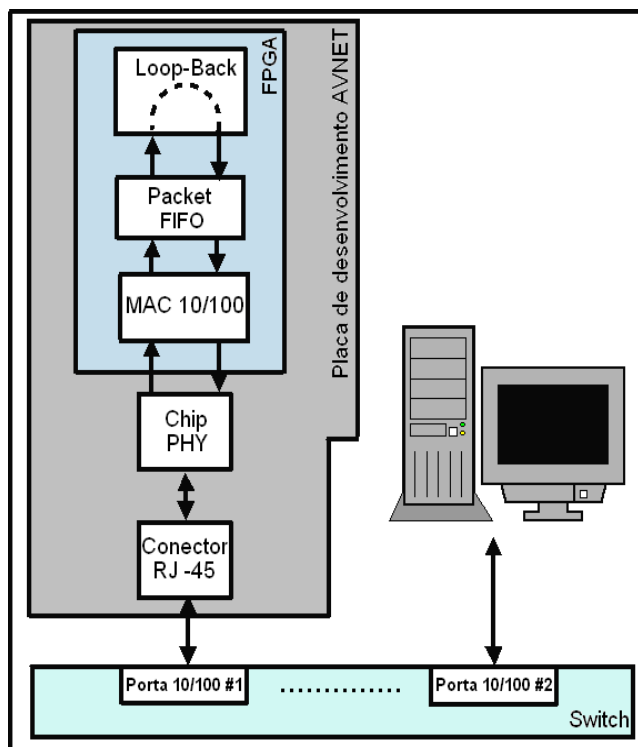


Figura 6.24: Cenário de testes para a validação física do IP *Soft Core* FEMAC

Esses quadros são enviados à porta 10/100 # 2 do switch, para em seguida serem espelhados à porta 10/100 #1. Os quadros ingressam na placa de desenvolvimento através do conector RJ-45, que os encaminha ao chip *transceiver* de nível físico (Chip PHY) Gig PHYTER® (NATIONAL SEMICONDUCTORS, 2004). Já no formato digital, os quadros são recebidos pelo Core FEMAC e o módulo de *Packet FIFO*. Um circuito de *loop-back* (composto por um adaptador de sinais de controle baseado em *flip-flops*) retorna os quadros ao core FEMAC, através do módulo *Packet FIFO*, para serem transmitidos de volta ao computador. No computador o analisador de protocolos Ethernet mostra a replicação dos quadros e a integridade dos dados transferidos na rede.

### 6.3.2 Cenário de testes dos IP *Soft Cores* GEMAC 1000, PCS/PMA 1000X e PacketFIFO

A figura 6.25 mostra o cenário de testes utilizado para a validação física dos IP *Soft Cores* GEMAC e PCS/PMA 1000X. O fluxo dos quadros transferidos na rede é o seguinte: Computador → Porta 10/100 do switch → Porta óptica do switch → Dispositivo SPF → Core PHY1000X (modo recepção) → Core GEMAC (modo recepção) → PacketFIFO (Rx FIFO) → Circuito *Loop-back* → PacketFIFO (Tx FIFO) → Core GEMAC (modo transmissão) → Core PHY1000X (modo transmissão) → Dispositivo SPF → Porta óptica do switch → Porta 10/100 do switch → Computador.

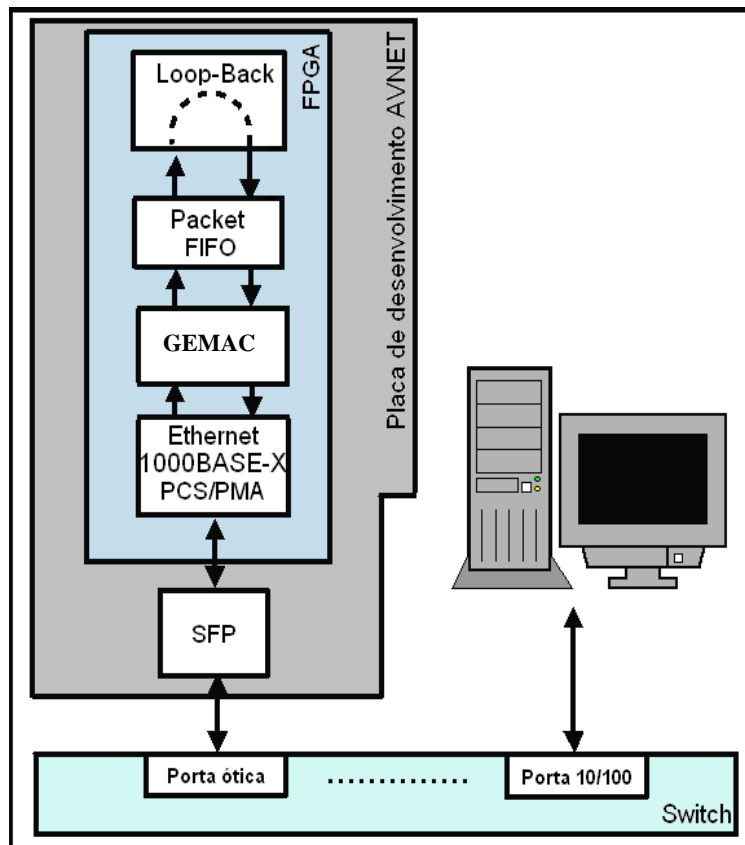


Figura 6.25: Cenário de testes para a validação física dos IP *Soft Cores* GEMAC e PHY1000X

Utilizando o mesmo tipo de tráfego exposto na seção 6.3.1 do presente trabalho, os quadros são transferidos desde o computador ao switch, que é o encarregado do espelhamento dos quadros que ingressam à porta 10/100 para a porta óptica. Nesse ponto, um link óptico transporta os quadros à placa de desenvolvimento. O *transceiver* óptico SFP encaminha os dados ao IP *Soft Cores* PHY1000X, que por sua vez, implementa as funcionalidades do nível físico dentro do FPGA disponibilizando os quadros em formato digital para o IP *Soft Cores* GEMAC. A partir daí, o IP *Soft Cores* GEMAC recebe os quadros e os encaminha ao módulo Packet FIFO. O circuito de *loop-back* realiza a função de realimentação de dados e permite que os IP *Soft Cores* GEMAC e PHY1000X transmitam de volta os mesmos quadros ao computador. No computador o analisador de protocolos Ethernet mostra a replicação dos quadros e a integridade dos dados transferidos na rede.

A replicação de quadros deve ser equivalente nos dois cenários de teste anteriormente descritos, já que a rede de interconexão é a mesma (somente são modificados os tipos de *links*). Na figura 6.26 é mostrado um *screenshot* retirado do Ethernet indicando a replicação de quadros.

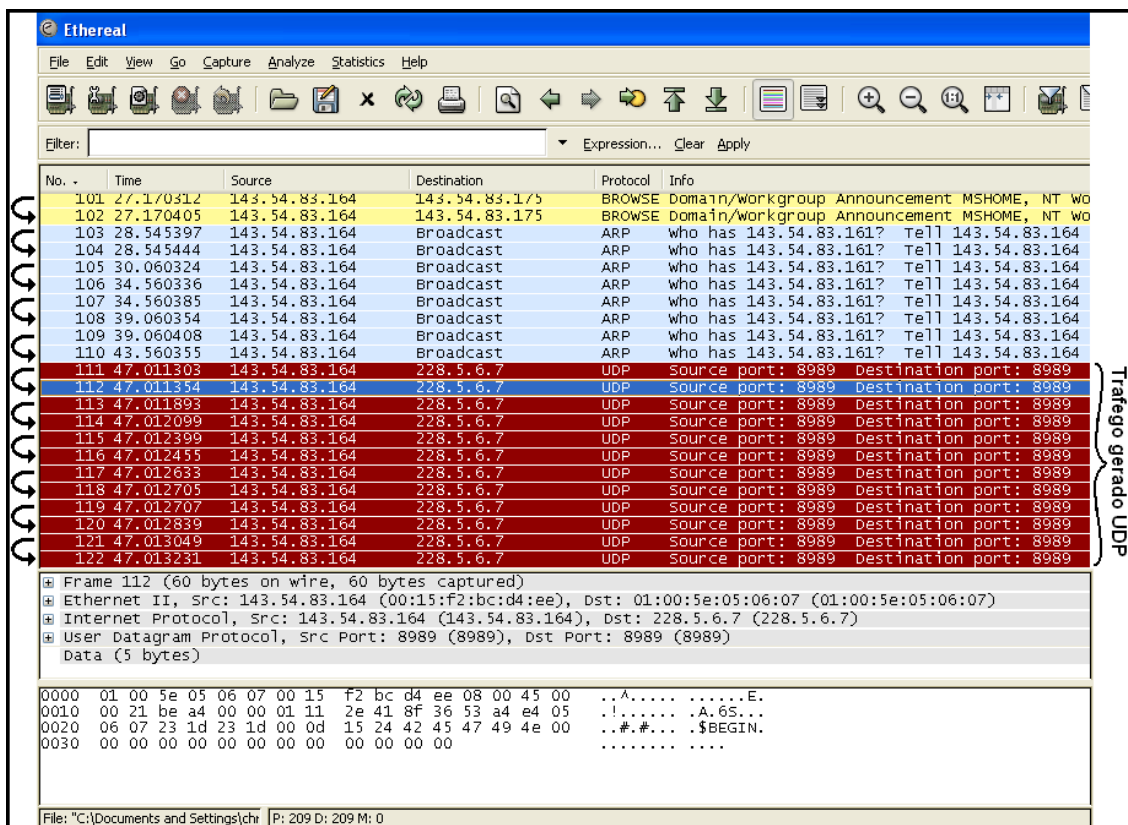


Figura 6.26: Quadros replicados visualizados no Ethereal

O uso do analisador de protocolos Ethereal, do Chipscope e da arquitetura de rede montada para os testes, é uma alternativa eficiente para a validação física dos *IP Soft Cores* desenvolvidos. Essa proposta permite de maneira simples e segura manipular um elevado volume de informação e uma rápida detecção de erros nos quadros transferidos.

### 6.3.3 Cenários de testes do *IP Soft Core OAM Ethernet*

Para validar fisicamente o projeto, foi utilizada a ferramenta ChipScope Pro v8.1i da Xilinx (avaliação dos sinais internos do módulo prototipado em FPGA) e o software Ethereal (avaliação dos pacotes enviados e recebidos tanto pela placa como pelo host remoto).

O *IP Soft Core OAM Ethernet*, como demonstrado na figura 6.22, possui três interfaces: Interface *OAM Client*, Interface *MAC Client*, e Interface *MAC*. Com o objetivo de validar essas três interfaces foram criados três cenários de teste chamados OAM1, OAM2 e OAM3 respectivamente.

Cabe salientar que todos os cenários foram validados na arquitetura hardware/software com uma versão do *OAM Client* rodando sobre o processador Microblaze, sendo que no cenário OAM1 e OAM2, o *OAM Client* estava configurado em modo passivo, e no cenário OAM3, o *OAM Client* estava configurado em modo ativo. No modo ativo, o *OAM Client* gera pacotes *OAMPDU Discovery* e solicita ao *IP Soft Core OAM Ethernet* através da interface *OAM Client* o envio desses pacotes. Esses pacotes são gerados periodicamente em um intervalo de tempo de 1 minuto.

### 6.3.3.1 Cenário OAM1 para validação da Interface MAC

Para validar a interface foi criado um cenário de teste onde os pacotes recebidos pelo sistema prototipado em FPGA eram retornados para o MAC simulando uma função de *loopback* configurado pelo OAM Client. A figura 6.27 ilustra esse cenário.

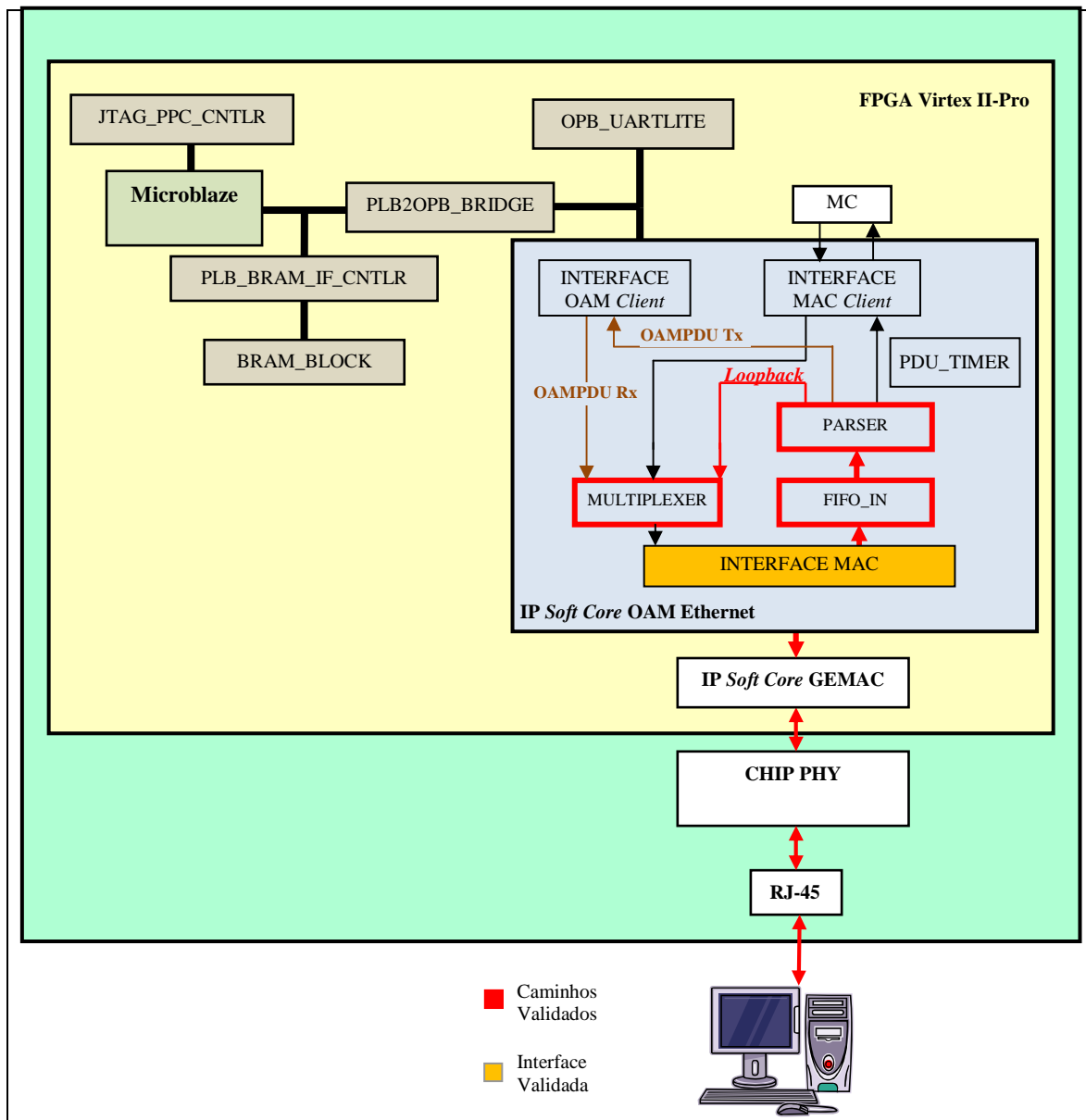


Figura 6.27: Cenário OAM1 de teste da Interface OAM\_Sublayer/MAC e OAM\_Sublayer/OAM\_Client

A arquitetura HW/SW foi sintetizada em FPGA e prototipada na placa de desenvolvimento AVNET. Para geração dos pacotes, foi efetuada uma comunicação com um PC (host remoto) através de um cabo do tipo UTP-5 em um link Ethernet de 100 Mbits. Para visualização do fluxo pacotes enviados e recebidos pelo PC, foi utilizado o software Ethereal. A figura 6.28 mostra o fluxo gerado.

No. -	Time	Source	Destination	Protocol	Info
PC 1	0.000000	143.54.83.169	Broadcast	ARP	who has 143.54.83.161? Tell 143.54.83.169
PD 2	0.000224	143.54.83.169	Broadcast	ARP	who has 143.54.83.161? Tell 143.54.83.169
PC 3	0.018469	143.54.83.169	Broadcast	ARP	who has 143.54.83.161? Tell 143.54.83.169
PD 4	0.018659	143.54.83.169	Broadcast	ARP	who has 143.54.83.161? Tell 143.54.83.169
PC 5	0.260766	143.54.83.169	Broadcast	ARP	who has 143.54.83.161? Tell 143.54.83.169
PD 6	0.260955	143.54.83.169	Broadcast	ARP	who has 143.54.83.161? Tell 143.54.83.169
PC 7	7.414558	143.54.83.169	143.54.83.175	NBNS	Name query NB TALK.GOOGLE.COM<00>
PD 8	7.415285	143.54.83.169	143.54.83.175	NBNS	Name query NB TALK.GOOGLE.COM<00>
PC 9	8.161946	143.54.83.169	143.54.83.175	NBNS	Name query NB TALK.GOOGLE.COM<00>
PD 10	8.162028	143.54.83.169	143.54.83.175	NBNS	Name query NB TALK.GOOGLE.COM<00>
PC 11	8.914199	143.54.83.169	143.54.83.175	NBNS	Name query NB TALK.GOOGLE.COM<00>
PD 12	8.914933	143.54.83.169	143.54.83.175	NBNS	Name query NB TALK.GOOGLE.COM<00>
PC 13	9.516355	143.54.83.169	Broadcast	ARP	who has 143.54.83.161? Tell 143.54.83.169
PD 14	9.516611	143.54.83.169	Broadcast	ARP	who has 143.54.83.161? Tell 143.54.83.169
PC 15	10.561913	143.54.83.169	143.54.83.175	BROWSE	Domain/workgroup Announcement DATACOM, NT workstation,
PD 16	10.562031	143.54.83.169	143.54.83.175	BROWSE	Domain/workgroup Announcement DATACOM, NT workstation,
PC 17	10.663394	143.54.83.169	Broadcast	ARP	who has 143.54.83.161? Tell 143.54.83.169
PD 18	10.663572	143.54.83.169	Broadcast	ARP	who has 143.54.83.161? Tell 143.54.83.169
PC 19	12.014898	143.54.83.169	Broadcast	ARP	who has 143.54.83.161? Tell 143.54.83.169
PD 20	12.014964	143.54.83.169	Broadcast	ARP	who has 143.54.83.161? Tell 143.54.83.169
PC 21	13.664080	143.54.83.169	Broadcast	ARP	who has 143.54.83.161? Tell 143.54.83.169
PD 22	13.664262	143.54.83.169	Broadcast	ARP	who has 143.54.83.161? Tell 143.54.83.169

Frame 1 (42 bytes on wire, 42 bytes captured)  
 Ethernet II, Src: 143.54.83.169 (00:15:58:54:25:2b), Dst: Broadcast (ff:ff:ff:ff:ff:ff)  
 Address Resolution Protocol (request)

Figura 6.28: Fluxo de pacotes gerado no Cenário OAM1

Como pode ser observado no fluxo de pacotes da figura 6.27, todos os pacotes enviados pelo PC (PC) para a placa de desenvolvimento (PD) foram retornados. Isso valida o caminho de loopback interno do módulo OAM\_Sublayer.

Como nesse cenário o cliente OAM está configurado em modo passivo, não são gerados pacotes do tipo OAM discovery. Da mesma forma o PC não possui um cliente OAM e, conseqüentemente, não gera pacotes desse tipo.

### 6.3.3.2 Cenário OAM2 – Validação da Interface OAM\_Sublayer/MAC Client

Um outro cenário foi criado para validar a Interface OAM\_Sublayer/MAC Client. Nesse cenário, todos os pacotes identificados como não sendo OAMPDU devem ser encaminhados para o MAC Client. No lado do MAC Client foi implementado um módulo em hardware que simula um cliente MAC em modo loopback retornando para a fonte todos os pacotes recebidos. A figura 6.29 ilustra esse cenário.



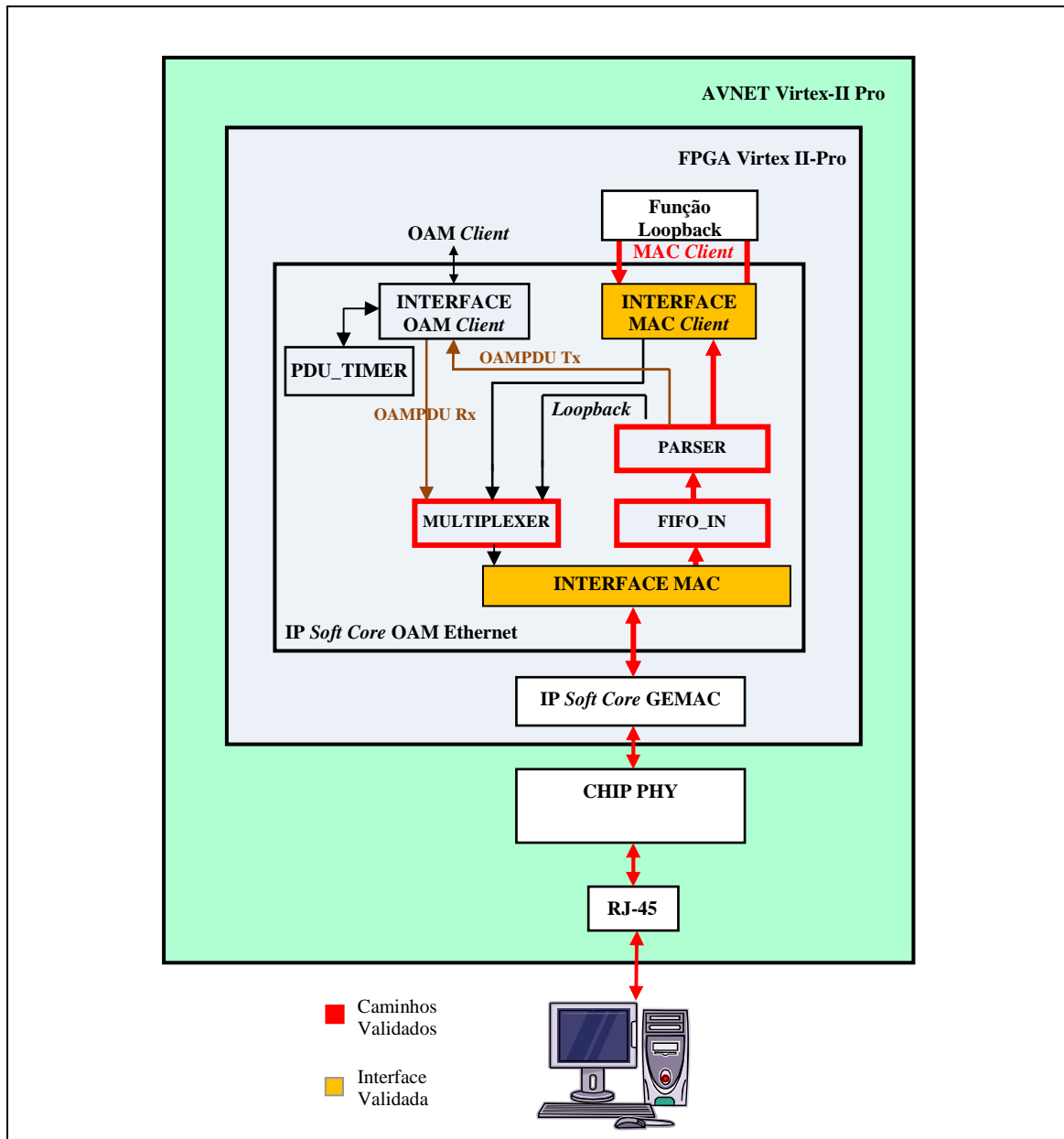
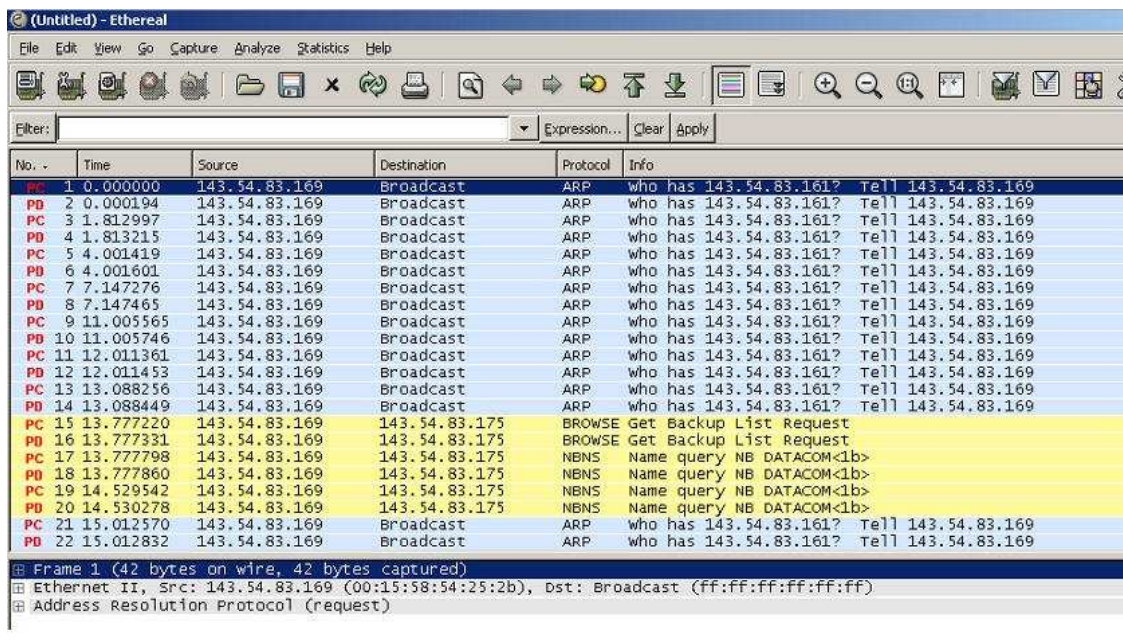


Figura 6.29: Cenário OAM2 de teste da Interface OAM\_Sublayer/MAC e OAM\_Sublayer/OAM\_Client

A arquitetura HW/SW foi sintetizada em FPGA e prototipada na placa de desenvolvimento AVNET. Para geração dos pacotes, foi efetuada uma comunicação com um PC (host remoto) através de um cabo do tipo UTP-5 em um link Ethernet de 100 Mbits. Para visualização do fluxo pacotes enviados e recebidos pelo PC, foi utilizado o software Ethernet. A figura 6.30 mostra o fluxo gerado.



No. -	Time	Source	Destination	Protocol	Info
PC 1	0.000000	143.54.83.169	Broadcast	ARP	who has 143.54.83.161? Tell 143.54.83.169
PB 2	0.000194	143.54.83.169	Broadcast	ARP	who has 143.54.83.161? Tell 143.54.83.169
PC 3	1.812997	143.54.83.169	Broadcast	ARP	who has 143.54.83.161? Tell 143.54.83.169
PB 4	1.813215	143.54.83.169	Broadcast	ARP	who has 143.54.83.161? Tell 143.54.83.169
PC 5	4.001419	143.54.83.169	Broadcast	ARP	who has 143.54.83.161? Tell 143.54.83.169
PB 6	4.001601	143.54.83.169	Broadcast	ARP	who has 143.54.83.161? Tell 143.54.83.169
PC 7	7.147276	143.54.83.169	Broadcast	ARP	who has 143.54.83.161? Tell 143.54.83.169
PB 8	7.147465	143.54.83.169	Broadcast	ARP	who has 143.54.83.161? Tell 143.54.83.169
PC 9	11.005565	143.54.83.169	Broadcast	ARP	who has 143.54.83.161? Tell 143.54.83.169
PB 10	11.005746	143.54.83.169	Broadcast	ARP	who has 143.54.83.161? Tell 143.54.83.169
PC 11	12.011361	143.54.83.169	Broadcast	ARP	who has 143.54.83.161? Tell 143.54.83.169
PB 12	12.011453	143.54.83.169	Broadcast	ARP	who has 143.54.83.161? Tell 143.54.83.169
PC 13	13.088256	143.54.83.169	Broadcast	ARP	who has 143.54.83.161? Tell 143.54.83.169
PB 14	13.088449	143.54.83.169	Broadcast	ARP	who has 143.54.83.161? Tell 143.54.83.169
PC 15	13.777220	143.54.83.169	143.54.83.175	BROWSE	Get Backup List Request
PB 16	13.777331	143.54.83.169	143.54.83.175	BROWSE	Get Backup List Request
PC 17	13.777798	143.54.83.169	143.54.83.175	NBNS	Name query NB DATACOM<lb>
PB 18	13.777860	143.54.83.169	143.54.83.175	NBNS	Name query NB DATACOM<lb>
PC 19	14.529542	143.54.83.169	143.54.83.175	NBNS	Name query NB DATACOM<lb>
PB 20	14.530278	143.54.83.169	143.54.83.175	NBNS	Name query NB DATACOM<lb>
PC 21	15.012570	143.54.83.169	Broadcast	ARP	who has 143.54.83.161? Tell 143.54.83.169
PB 22	15.012832	143.54.83.169	Broadcast	ARP	who has 143.54.83.161? Tell 143.54.83.169

Frame 1 (42 bytes on wire (42 bytes captured))  
 Ethernet II, Src: 143.54.83.169 (00:15:58:54:25:2b), Dst: Broadcast (ff:ff:ff:ff:ff:ff)  
 Address Resolution Protocol (request)

Figura 6.30: Fluxo de pacotes gerado no Cenário OAM2

Como pode ser observado no fluxo de pacotes da figura 6.31, todos os pacotes enviados pelo PC para a placa de desenvolvimento foram retornados. Isso valida o caminho de loopback implementado no lado do *MAC Client*.

### 6.3.3.3 Cenário OAM3 – Validação do OAM Client rodando em software

Neste cenário, todos os pacotes do tipo OAM são identificados e repassados para o *OAM Client* através da Interface *OAM Client*. Os demais pacotes são descartados.

Uma versão do *OAM Client* configurada em modo passivo com funcionalidade de Discovery foi compilada e executada sobre o processador Microblaze da arquitetura hw/sw. Essa versão, em conjunto com os demais componentes da arquitetura hw/sw foi sintetizada em FPGA e prototipada na placa de desenvolvimento AVNET. Para a interface *OAM Client* bem como o software *OAM Client* rodando sobre o Microblaze, foi efetuada uma conexão a um Switch Ethernet em um link de 100 Mbits. Esse Switch possui um cliente OAM habilitado e, portanto, pode realizar a comunicação através do protocolo OAM.

A figura 6.31 ilustra o cenário criado.

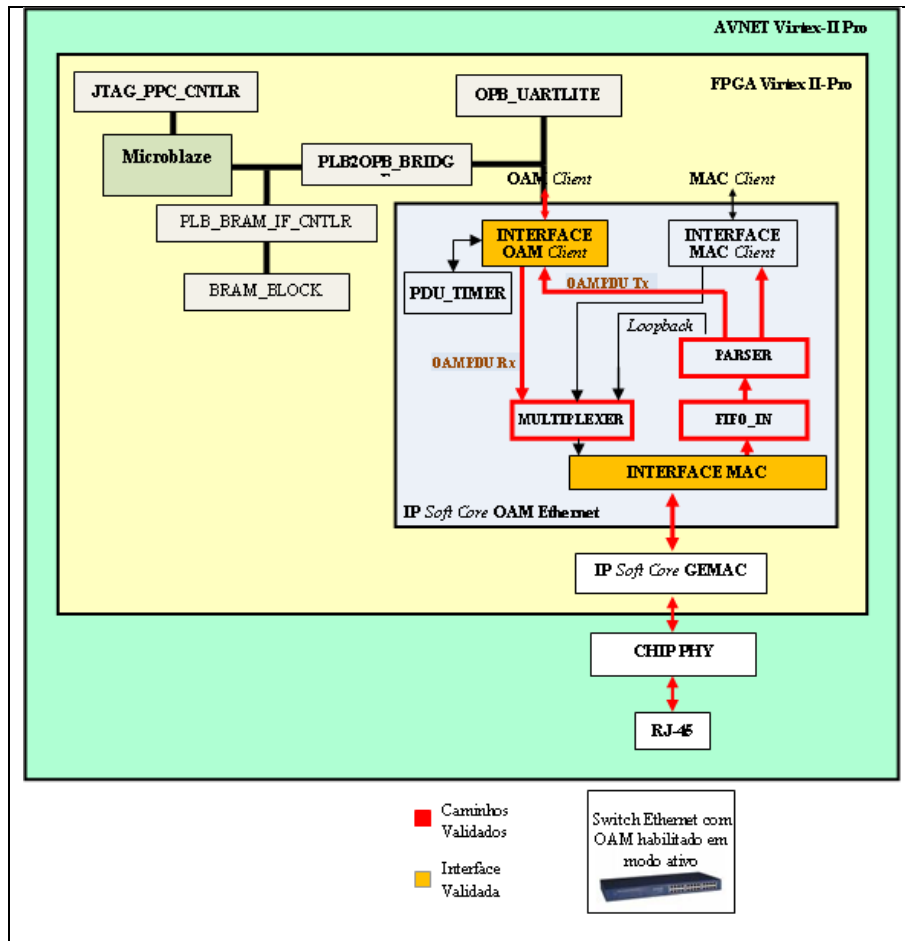


Figura 6.31: Cenário OAM3 de validação da Interface OAM\_Client e teste do OAM\_Client rodando em software

A figura 6.32 exibe o fluxo de pacotes gerado nesse cenário de validação.

```

***** Pacote Recebido do Switch *****
1*80*C2*0*0*2*0*4*0F*0*8*8B*88*9*3*0*30*0*1*10*1*0*0*0*0*37*0*4*0F*0*0*0*0*0*
0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*

***** Enviando pacote *****

***** Pacote enviado para o Switch *****

1*80*C2*0*0*2*0*1*2*3*4*5*88*9*3*0*30*0*1*10*1*0*0*0*1*0*37*0F*4*0*0*0*0*0*2*1*1
0*0*1*0*0*0*0*4*0*37*0*0*0*0F*0*21*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*

***** Enviando pacote *****

***** Pacote enviado para o Switch *****

1*80*C2*0*0*2*0*1*2*3*4*5*88*9*3*0*30*0*1*10*1*0*0*0*1*0*37*0F*4*0*0*0*0*0*2*1*1
0*0*1*0*0*0*0*4*0*37*0*0*0*0F*0*21*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*

***** Recebendo Pacote *****

***** Pacote Recebido do Switch *****
1*80*C2*0*0*2*0*4*0F*0*8*8B*88*9*3*0*50*0*1*10*1*0*0*0*0*37*0*4*0F*0*0*0*0*0*2*1
0*1*0*0*0*0*1*0*37*0F*4*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*

***** Enviando pacote *****

***** Pacote enviado para o Switch *****

1*80*C2*0*0*2*0*1*2*3*4*5*88*9*3*0*50*0*1*10*1*0*0*0*1*0*37*0F*4*0*0*0*0*0*2*1*1
0*0*1*0*0*0*0*4*0*37*0*0*0*0F*0*21*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*

***** Recebendo Pacote *****

***** Pacote Recebido do Switch *****
1*80*C2*0*0*2*0*4*0F*0*8*8B*88*9*3*0*50*0*1*10*1*0*0*0*0*37*0*4*0F*0*0*0*0*0*2*1
0*1*0*0*0*0*1*0*37*0F*4*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*

***** Enviando pacote *****

***** Pacote enviado para o Switch *****

1*80*C2*0*0*2*0*1*2*3*4*5*88*9*3*0*50*0*1*10*1*0*0*0*1*0*37*0F*4*0*0*0*0*0*2*1*1
0*0*1*0*0*0*0*4*0*37*0*0*0*0F*0*21*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*

```

Figura 6.32: Fluxo de pacotes gerado no Cenário 3 com OAM Client no modo passivo

Também foi efetuada a validação com o *OAM Client* no modo ativo. O mesmo comportamento de comunicação entre a placa de desenvolvimento e o switch realizado na experimentação anterior é esperado agora. A diferença é que agora o *OAM Client* que gerará um pacote OAM Discovery com o campo Flag igual a “08”. A partir daí, os dois lados irão efetuar a comunicação até estabilizar no pacote com o campo *Flag* igual a “50”. A figura 6.33 ilustra o fluxo de pacotes gerado nessa validação.

```
***** Enviando pacote *****
***** Pacote enviado para o Switch *****
1+80+C2+0+0+2+0+1+2+3+4+5+88+9+3+0+8+0+1+10+1+0+0+1+0+37+0F+4+0+0+0+0+0+0+
0+0+0+0+0+0+0+0+0+0+21+0+0+55+80+0+0+4F+88+

***** Recebendo Pacote *****
***** Pacote Recebido do Switch *****
1*80*C2*0*0*2*0*4*0F*0*8*0B*88*9*3*0*50*0*1*10*1*0*0*0*0*37*0*4*0F*0*0*0*0*2*1
0*1*0*0*0*1*0*37*0F*4*0*0*0*0*0*0*0*0*0*0*0*0*0*0*

***** Enviando pacote *****
***** Pacote enviado para o Switch *****
1+80+C2+0+0+2+0+1+2+3+4+5+88+9+3+0+10+0+1+10+1+0+0+1+0+37+0F+4+0+0+0+0+2+1+1
0+0+1+0+0+0+4+0+37+0+0+0F+0+21+0+0+0+0+0+0+

***** Recebendo Pacote *****
***** Pacote Recebido do Switch *****
1*80*C2*0*0*2*0*4*0F*0*8*0B*88*9*3*0*50*0*1*10*1*0*0*0*0*37*0*4*0F*0*0*0*0*2*1
0*1*0*0*0*1*0*37*0F*4*0*0*0*0*0*0*0*0*0*0*0*0*0*0*

***** Enviando pacote *****
***** Pacote enviado para o Switch *****
1+80+C2+0+0+2+0+1+2+3+4+5+88+9+3+0+50+0+1+10+1+0+0+1+0+37+0F+4+0+0+0+0+2+1+1
0+0+1+0+0+0+4+0+37+0+0+0F+0+21+0+0+0+0+0+0+

***** Recebendo Pacote *****
***** Pacote Recebido do Switch *****
1*80*C2*0*0*2*0*4*0F*0*8*0B*88*9*3*0*50*0*1*10*1*0*0*0*0*37*0*4*0F*0*0*0*0*2*1
0*1*0*0*0*1*0*37*0F*4*0*0*0*0*0*0*0*0*0*0*0*0*0*0*
```

Figura 6.33: Fluxo de pacotes gerado no Cenário 3 com OAM *Client* no modo ativ

## 7 DESENVOLVIMENTO DOS CONVERSORES DE MEIOS ETHERNET

### 7.1 A Jiga BERT para validação dos Conversores de Meios Ethernet

Esta seção apresenta o projeto da Jiga BERT (*Bit-Error Rate Tester*) para apoio na validação física dos dois Conversores de Meios Ethernet em regime “*Full-Line Rate*” (injeção de tráfego *wire-speed* ou de taxa máxima no meio físico).

A Jiga BERT tem duas funções principais:

- 1) Geração de Tráfego *wire-speed Ethernet*: Deve ter capacidade para geração de um padrão PRBS (*Pattern Pseudo-Random Bit Sequence*) do tipo  $1 + x^5 + x^9$  definido pelo ITU-T para testes sobre enlaces Ethernet, sendo possível gerar tráfego tanto numa porta elétrica *Fast Ethernet* como numa porta óptica *Gigabit Ethernet* (ambas na taxa máxima)
- 2) Detecção e Análise de PRBS: Tem que ter a capacidade de detectar o início do PRBS, que neste caso começa no primeiro byte do *payload* o quadro Ethernet (ou após o recebimento dos cabeçalhos Ethernet). Também deve ser possível a contagem de erros de bit e opções para zerar os contadores e processos do análise de PRBS.

No projeto dos componentes que farão as funções de geração/análise PRBS, é muito importante a escolha correta do recurso computacional que fará ditas ações. Uma solução rápida e comumente utilizada em implementações HW é o uso de LFSR (Chavas, 2007)

A LFSR é um registrador de deslocamento que utiliza lógica combinacional realimentada para gerar randomicamente, em cada ciclo de relógio, uma seqüência de valores binários. Notar que o nome pseudo-randômico é atribuído pelo fato de depender de uma função determinística para geração e análise da seqüência binária. Alguns o chamam de “*n-bit LFSR*”, para o LFSR rodando sobre vetores de *n*-bits, considerando que não qualquer “*n*” garante o bom funcionamento da LFSR (Chavas, 2007). No presente trabalho é utilizado o vetor  $n=9$ , tanto para geração como para análise de seqüências PRBS recebidas.

Na continuação, a figura 7.1 o diagrama de blocos da Jiga BERT, para o teste *Full-Line Rate* das duas arquiteturas de EDD proposto. É possível diferenciar dois módulos, um chamado BERT TX e outro chamado BERT RX, ambos possuem a mesma seqüência PRBS (também conhecido como “semente”). Salienta-se que três *IP Soft Cores* (FEMAC, GEMAC e PHY1000X) são re-aproveitados para construção da Jiga BERT com a finalidade de oferecer os níveis 1 e 2 do RM-OSI.

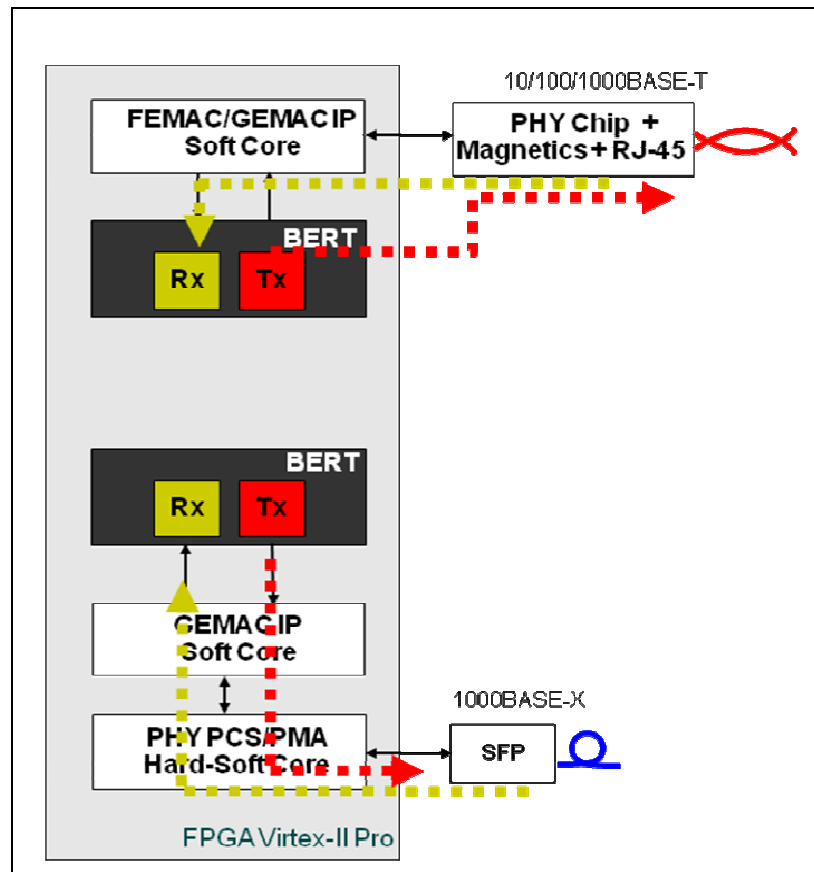


Figura 7.1: Jiga BERT

## 7.2 A solução MECALIMBA

Segundo o definido no Capítulo 5, existem duas arquiteturas para o EDD proposto. Cada uma delas utiliza um Conversor de Meios Ethernet (FE2GE ou GE2GE) para implementar as principais camadas e sub-camadas definidas em (IEEE, 2005a). Um dos Conversores de Meios Ethernet, o Conversor de Meios FE2GE, precisa de um mecanismo específico para adaptar as diferentes taxas de transferência e prover a máxima taxa (neste caso limitada em 10Mbits/s ou 100Mbits/s). Para isto, no presente capítulo será mostrado um mecanismo capaz de realizar esta função, chamado aqui de MECALIMBA – MECanismo Automático para LIMitação de Banda –.

O mecanismo MECALIMBA é baseado no mecanismo de controle de fluxo IEEE802.x (IEEE, 2005a) e é implementado nos *IP Soft Cores* GEMAC e *Packet FIFO* da interface Gigabit Ethernet do Conversor de Meios FE2GE. No caso, o *IP Soft Core*

GEMAC, ativa o bloqueio do transmissor do dispositivo remoto que injeta tráfego na interface ótica, a través do envio de quadros especiais de *pause*. Para que isto seja possível, o *IP Soft Core Packet FIFO* ligado no módulo GEMAC deve lhe informar que seus níveis de preenchimento estão fora dos seus limites e num futuro próximo podem ocorrer perdas de quadros por *Overrun* ou *Underrun*.

A figura 7.2 apresenta uma simulação em Modelsim do Conversor de Meios FE2GE, quando a porta ótica é estressada para trabalhar em 1000Mbps/s e a porta elétrica está estressada para trabalhar em 100Mbps/s. Para isto é utilizada a Jiga BERT para geração e análise do tráfego *wire-speed* PRBS nas duas taxas ao 100%. O grupo de sinais da parte superior da figura mostra como o tráfego *wire-speed* 1000Mbps/s é injetado na porta ótica do Conversor de Meios FE2GE e o mesmo tráfego sendo repassado para porta elétrica do mesmo conversor numa velocidade de 100Mbps/s (grupo de sinais inferior). Notar que existe um sinal chamado *FIFO\_LEVEL* que indica o nível de preenchimento do Packet FIFO ligada à porta *gigabit* ótica e tem um comportamento crescente (linear). Este comportamento se deve ao fato de que os pacotes recebidos na FIFO de recepção são escritos muito mais rápido do que lidos da mesma (10X), até chegar no nível de preenchimento máximo e indicando erro por condição de FIFO cheia, o que traz como consequência a perda de pacotes do lado da interface elétrica 100Mbps/s.

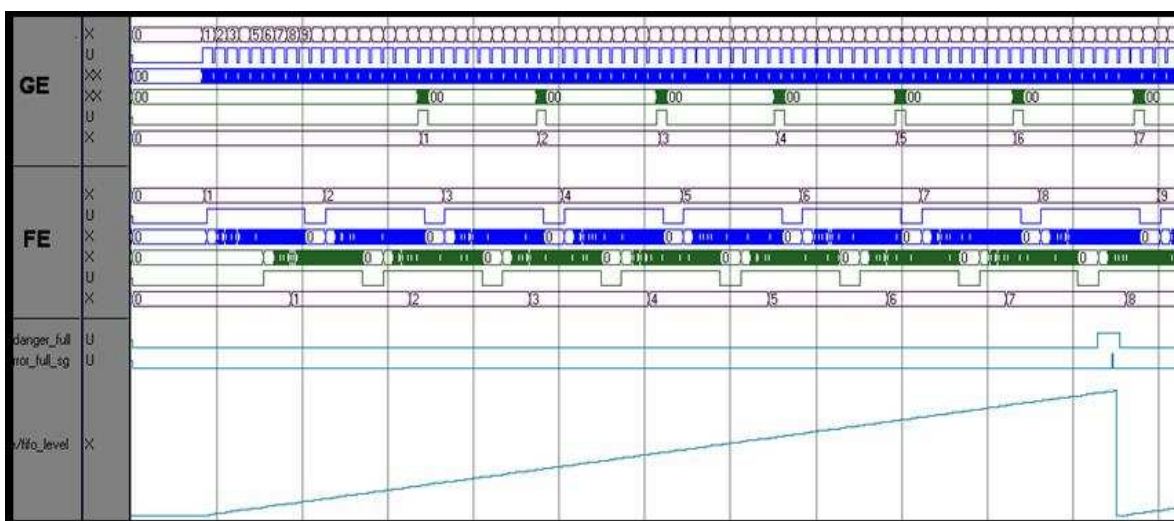


Figura 7.2: Conversor de Meios FE2GE sem mecanismo MECALIMBA

A figura 7.3 apresenta a solução MECALIMBA, nota-se que o nível de preenchimento da FIFO de recepção da porta *gigabit* do Conversor de Meios FE2GE é crescente até certo nível da FIFO e logo cai até regularizar a ocupação da FIFO de recepção. Para que isto aconteça, o *IP Soft Core Packet FIFO* deve disparar o processo de geração de pacote de *pause*, os quais serão recebidos pela Jiga BERT, para que esta “pare” de transmitir durante um tempo que permita liberar todo o espaço ocupado na FIFO de recepção (nesse intervalo o nível de preenchimento começa cair linearmente). Na media, a taxa recebida pela porta *gigabit* passa a ser 100Mbps/s. Desta forma é possível evitar a perda de pacotes na passagem de tráfego a través do Conversor de Meios FE2GE.



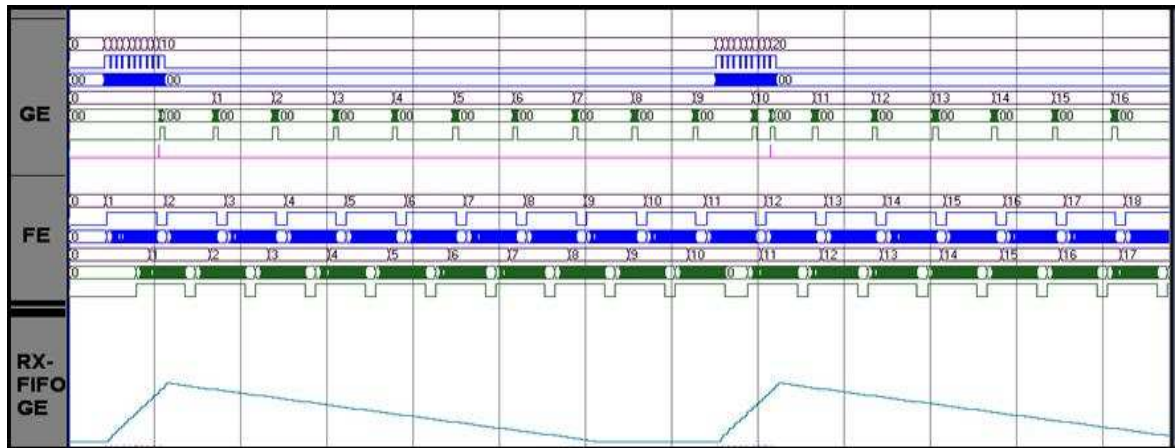


Figura 7.3: Conversor de Meios FE2GE com mecanismo MECALIMBA

O dimensionamento dos níveis de preenchimento do *IP Soft Core Packet FIFO* foram calculados empiricamente a partir de simulações com diferentes tamanhos de quadros e para uma taxa máxima nas duas portas. No caso do Conversor de Meios GE2GE não é aplicada a solução MECALIMBA devido à mesma largura de banda nas duas portas do conversor.

### 7.3 Propostas de Validação dos Conversores de Meios Ethernet

Nesta seção serão apresentadas três propostas de validação física dos Conversores de Meios FE2GE e GE2GE. Cada método proposto no presente trabalho visa atender uma serie de testes, alguns deles quantitativos e outros qualitativos, com o fim de mostrar que a integração dos Módulos IP desenvolvidos no presente trabalho é satisfaz as especificações definidas para o EDD proposto. Na continuação a organização dos métodos e seus respectivos testes relacionados:

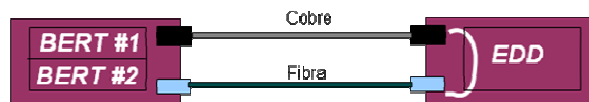
#### 7.3.1 Validação Tráfego em taxas críticas:

##### 7.3.1.1 Teste Full-Line Rate utilizando a Jiga BERT

Os recursos necessários para este tipo de validação são:

- 1 Placa de Desenvolvimento AVNET Virtex-II Pro para prototipagem da Jiga BERT: Deve ser possível implementar duas versões de Jiga BERT; uma versão utiliza a interface elétrica Gigabit Ethernet e a interface óptica Gigabit Ethernet, a outra versão utiliza a interface elétrica Fast Ethernet e a interface ótica Gigabit Ethernet.
- 1 Placa de Desenvolvimento AVNET V2PRO para prototipagem dos dois Conversores de Meios Ethernet FE2GE e GE2GE.
- Software ChipScope para *debug* das estatísticas implementadas nas duas versões da Jiga BERT (como por exemplo: Contadores de pacotes recebidos, transmitidos, pacotes descartados, pacotes com erro de bit, etc)

O tráfego *wire-speed* gerado por cada uma das portas da Jiga BERT deve ser configurado de tal forma que seja possível detectar os cabeçalhos corretamente e conseqüentemente analisar o comportamento dos Conversores de Meios Ethernet em regime *Full-Line Rate*. Na figura 7.4, são mostradas as duas possíveis ligações entre os dois kits de desenvolvimento AVNET disponíveis:



- Conversor de Meios GE2GE => OK! BER < 10<sup>-12</sup>
- Conversor de Meios FE2GE, com MECALIMBA => OK! BER < 10<sup>-12</sup>

Figura 7.4: Teste Full Line Rate do Conversor de Meios GE2GE

Segundo (IEEE, 2005a) há probabilidade de erro para sistemas Gigabit Ethernet deve ser menor que um BER (Bit Error Rate) igual a 10<sup>-12</sup>. Isto quer dizer que a probabilidade de erros é de 1 bit em 10<sup>12</sup> bits transmitidos. Os resultados obtidos encontram-se baixo desta probabilidade, tanto nos testes com o Conversor de Meios FE2GE como no Conversor de Meios GE2GE.

### 7.3.1.2 Injeção e Análise de Tráfego utilizando a ferramenta “Ping Flood”

Os recursos necessários para este tipo de validação são:

- Duas Placas AVNET V2PRO: Para implementação de um par de Conversores de Meios FE2GE ou um par de Conversores de Meios GE2GE interligados entre si, para que seja possível a interconexão ponto-a-ponto com dois PC's
- Dois PC's rodando no sistema operativo Linux: Com isto será possível utilizar o recurso de “*Ping Flood*” entre os dois PCS, passando através de um par de EDDs em serie.

O *Ping flood* não permite uma validação *Full-Line Rate*, já que o PC encontra-se limitado pelos processos executados ao mesmo tempo pelo sistema operativo e pelos *drivers* que controlam a placa de rede do PC. Estas limitações fazem que não seja possível por exemplo, configurar um IFG mínimo definido pela norma IEEE802.3 (96ns para Gigabit Ethernet e 960ns para Fast Ethernet). O *Ping flood* só permite a configuração do IFG na ordem dos milissegundos. Por outro lado, o *Ping Flood* acrescenta um nível de complexidade maior comparado com o teste da seção anterior 7.3.1.1, já que o tráfego gerado utiliza o protocolo ICMP que é baseado numa comunicação do tipo *request/reply* o que permite a validação nos dois sentidos (TX/RX).

A figura 7.5, mostra os dois cenários possíveis para validação dos dois tipos de Conversores de Meios. A estratégia é disparar *Ping flood* entre os dois PCs e verificar a transparência do EDD proposto.



- Ping **Flood** pelas NIC's dos PC's => OK! BER <  $10^{-12}$ , Throughput << 100 Mbits/s



- Ping **Flood** pelas NIC's dos PC's => OK! BER <  $10^{-12}$ , Throughput << 1Gbits/s

Figura 7.5: Testes de tráfego utilizando ferramenta “*Ping Flood*”

A tabela 7.1, mostra os resultados obtidos na validação do Conversor de Meios FE2GE

Tabela 7.1: Estatísticas de *Ping Flood*

	PC#1	PC#2
<b>IP Address</b>	192.168.0.1	192.168.0.2
<b>Pacotes transmitidos</b>	207578888	450164835
<b>Pacotes recebidos</b>	207578888	450164835
<b>Packet loss</b>	0%	0%
<b>Time(ms)</b>	53584643 (aprox. 15hrs)	53601330 (aprox. 15hrs)
<b>RTT(ms) min/avg/max/dev</b>	0,044/0,057/21,696/0,029	0,045/0,057/21,659/0,033
<b>IPG/EWMA (ms)</b>	0,258/0,062	0,119/0,052

### 7.3.2 Validação de Desempenho

O objetivo desta proposta de validação é tratar tráfego UDP e TCP, para verificar se o EDD proposto permanece transparente mesmo na presença de protocolos de alto nível

#### 7.3.2.1 Injeção de Tráfego UDP e TCP utilizando NetPerf

Os recursos necessários para este tipo de validação são:

- Duas Placas AVNET V2PRO: Para implementação de um par de Conversores de Meios FE2GE ou um par de Conversores de Meios GE2GE interligados entre si, para que seja possível a interconexão ponto-a-ponto com dois PCs

- Dois PCs, uma delas rodando como Servidor *NetPerf* e a outra rodando como Cliente *NetPerf*.

Na figura 7.6, pode-se ver os dois cenários possíveis para validação dos dois tipos de Conversores de Meios. A estratégia é rodar tráfego UDP e tráfego TCP e calcular o *throughput* atingido para cada um dos tipos de tráfego.

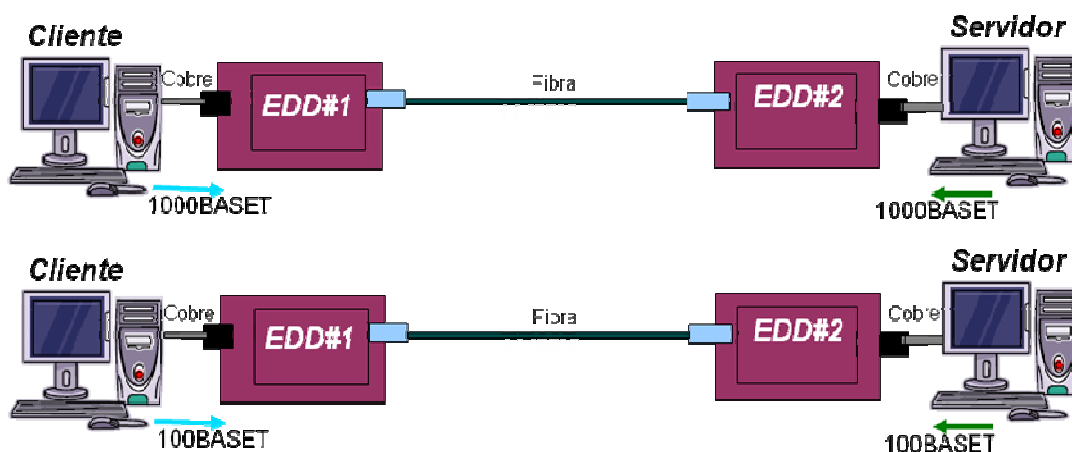


Figura 7.6: Cenários de validação utilizando trafego UDP/TCP

A continuação se mostra os resultados obtidos na validação do Conversor de Meios GE2GE:

- Trafego UDP => *Throughput* ~ 612,99 Mbits/s, sem perda de pacotes.
- Trafego TCP => *Throughput* ~ 490,84 Mbits/s, sem perda de pacotes.

Sendo que os parâmetros configurados no *NetPerf* são: tempo de teste 10 seg, 520544 quadros de 1472-bytes.

### 7.3.3 Validação de Conformidade com a norma IEEE 802.3

O objetivo desta proposta de validação é mais qualitativo do que quantitativo, a idéia é passar tráfego real através do EDD proposto com a finalidade de validar a conformidade no tratamento dos protocolos e aplicações de rede mais populares.

Os recursos necessários para este tipo de validação são:

- Duas Placas AVNET V2PRO: Para implementação de um par de Conversores de Meios FE2GE ou um par de Conversores de Meios GE2GE interligados entre si, para que seja possível a interconexão ponto-a-ponto entre um único PC e uma rede local.
- Um PC com um analisador de protocolos do tipo *Wireshark* ou *Ethereal*, assim como também aplicativos de rede populares.

Na figura 7.7 se mostram os dois cenários possíveis para validação dos dois tipos de Conversores de Meios. A estratégia é rodar no PC as seguintes aplicações:

- Navegador HTML
- Cliente FTP
- Cliente CVS
- ICMP
- MSN (Windows® Messenger)
- etc.



Figura 7.7: Cenário de validação para testes de conformidade IEEE 802.3

Nas aplicações mencionadas acima, o EDD proposto comportou-se corretamente sem dificultar o acesso, seja por alteração do tráfego ou por latência intrínseca. Um dos testes mais críticos efetuados nesta abordagem foi baixar arquivos de servidores FTP de grande tamanho (Gigabytes) e verificar a integridade dos arquivos baixados utilizando a ferramenta de encriptação *md5*.

#### 7.4 Resultados de Implementação do EDD proposto

Na tabela 7.2, se mostra o uso de recursos do EDD proposto na sua versão utilizando Conversor GE2GE num dispositivo FPGA Virtex-II Pro 2VP30, com ou sem processador Microblaze (SoC).

Tabela 7.2: Recursos utilizados pelo Conversor de Meios GE2GE

	<i>Slices (2VP30)</i>	<i>Block-RAMs (2VP30)</i>	<i>Gates</i>
<b>TPMR (Conversor + OAM)</b>	2,485 (9%)	8 (5%)	579,234
<b>TPMR (SoC)</b>	3,836 (14%)	98 (72%)	6,564,183

## 8 CONCLUSÕES E TRABALHOS FUTUROS

- Esta dissertação apresentou duas opções de arquiteturas para um EDD de duas portas tendo como plataforma de implementação um dispositivo FPGA Virtex-II Pro. As arquiteturas propostas estão compostas por uma unidade central chamada Conversor de Meios Ethernet, a qual por sua vez, é composta por um conjunto de módulos IP. Sete módulos IP foram desenvolvidos e integrados com o processador embarcado MicroBlaze para cumprir a especificação técnica proposta no presente trabalho. Um oitavo módulo IP chamado módulo BERT foi desenvolvido para fins de validação física das duas arquiteturas alvo.

- As arquiteturas escolhidas para o EDD, de alguma forma, são dependentes dos recursos de implementação disponíveis para realização deste trabalho. Por exemplo, o IP Soft Core PHY1000X teve que ser desenvolvido com a finalidade de habilitar a interface com um transceiver SFP óptico na placa de desenvolvimento AVNET Virtex-II Pro. Mesmo sendo um componente desenvolvido para fins de prototipação, procurou-se minimizar a latência no caminho de dados TX/RX, o que permitiu obter um melhor desempenho em relação à solução equivalente da Xilinx.

- O módulo MIIM apresentou um ganho de área (LUTs) comparado com uma solução funcionalmente idêntica disponível no OpenCores. Nossa solução consegue este ganho após efetuar uma série de otimizações na lógica (todas elas independentes da tecnologia), e compete com a solução do OpenCores que vem sendo melhorada por um comunidade de pessoas há alguns anos.

- Os Módulos FEMAC e GEMAC foram sintetizados em *standard-cells* utilizando a tecnologia AMS 0,35um. Esse tipo de síntese ajudou descobrir que o bloco “Interface Host” ocupa aproximadamente 60% da área total do módulo. Considerando que este bloco não é descrito na norma IEEE 802.3 e foi desenvolvido unicamente para configuração dinâmica do módulo, é possível dispensar este bloco e definir uma configuração estática (forçada) para aplicações *low-cost*. Se a plataforma de implementação escolhida for um FPGA, os módulos GEMAC e FEMAC podem chegar ocupar aproximadamente 300 LUTs com o bloco “Interface Host” desabilitado.

- Os módulos IP desenvolvidos mostraram-se flexíveis e confiáveis após passar por uma série de testes críticos. Cenários reais e ferramentas de análise de protocolos de rede foram aplicados as versões disponíveis do EDD proposto, tendo como resultado uma elevada transparência em relação ao tratamento do fluxo de dados. Nesse sentido, se cumpre uma das especificações mais exigentes que é a “demarcação no nível físico”.

- A ideia de integrar o maior número de funcionalidades do EDD num único Chip (neste caso num dispositivo FPGA) minimiza o uso de componentes discretos externos

numa placa e reduz o roteamento da placa de circuito impresso (PCI) que for alocar o EDD. Como resultado, se obtém um baixo custo na produção em massa do EDD e também são evitados problemas de integridade de sinal pelo grande número de interconexões na PCI. Da mesma forma, é possível oferecer outras opções arquiteturais unicamente retirando ou mapeando os módulos IP desenvolvidos segundo a aplicação a ser atendida.

## 8.1 Principais realizações

- No âmbito Acadêmico:

a) Os IP Soft Cores FEMAC e GEMAC foram utilizados no Projeto TERACOM-UNISC (dirigido pelo Prof. Rafael Ramos dos Santos), no Projeto TERACOM-UFRGS (dirigido pelo Prof. Ricardo Reis), na dissertação de mestrado do ex-aluno Jean Hamersky, nos trabalhos de conclusão dos ex-alunos Felipe Kuentzer e Cristiano Battisti (da Eng. da computação UNISC) e do aluno Fábio Ramos (da Eng. da computação UFRGS)

b) Este trabalho é fruto de dois anos de contínuo desenvolvimento no marco do Convênio TERACOM-UFRGS, onde conseguiu-se integrar as necessidades da indústria com as pesquisas realizadas dentro do PPGC, para finalmente propor uma arquitetura inédita que conseguiu ser validada em cenários críticos e reais com sucesso.

- No âmbito industrial

a) Alguns IP Soft Cores estão sendo utilizados pelos grupos de P&D da Empresa DATACOM-Telemática para o desenvolvimento de Jigas de Teste Ethernet e novos produtos que envolvem interfaces Ethernet.

## 8.2 Trabalhos Futuros

- No marco do Projeto TERACOM-UFRGS, está planejado utilizar o desenvolvimento do presente trabalho para o projeto de um ASIC com funcionalidades de EDD e *switch layer 2* (financiado por FINEP)

- Implementar a rigor as especificações do TPMR, quando este seja aprovado como padrão oficial da IEEE. Isto seria de interesse para o grupo de trabalho IEEE P802.1aj e para aplicações de demarcação nas principais redes metropolitanas do país.

## 9 REFERÊNCIAS

METCALFE, M.; BOGGS, D. Ethernet: Distributed Packet Switching for Local Computer Networks. Communications of the ACM, v.19, n.5, p.395 – 404, July 1976. Disponível em <http://orkinos.cmpe.boun.edu.tr/~kosar/met/apr96.htm>. Acesso em: Maio 2009.

FRAZIER, H.; JOHNSON, H. Gigabit Ethernet: From 100 to 1,000 Mbps. IEEE Internet Computing. Los Alamitos, v.3, n.1, p.24-31, Jan./Feb. 1999.

CHIRUVOLU, G. et al. Issues and Approaches on Extending Ethernet Beyond LANs. In: IEEE Communications Magazine, Vol. 42, No. 3, pp. 80-86, March 2004.

McFARLAND et al. Ethernet OAM: Key Enabler for Carrier Class Metro Ethernet Services. In: IEEE Communications Magazine, Vol. 43, pp. 152-157, November 2005.

OHTA, H. Standardization Status on Carrier Class Ethernet OAM. In: IEICE Transactions on Communication, Vol. E89-B, No. 3, March 2006.

SOARES et al. Redes de computadores: das LANs, MANs e WANs às Redes ATM Editora Campus, 2a Edição, 1995.

IEEE Std 802.3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) access method and physical layer specifications. New York, 2005. (IEEE, 2005a)

SPURGEON et al. Ethernet: The Definitive Guide. Sebastopol: O'Reilly, 2002.

STEIN, Y. Ethernet OAM. RAD Data Communications, White Paper. 2006



METRO ETHERNET FORUM (MEF). Ethernet in the First Mile: Operations, Administration and Maintenance (OAM) – A Tutorial. November 2005a. Disponível em <<http://www.metroethernetforum.org>>. Acesso em: Maio 2009. (MEF, 2005a)

METRO ETHERNET FORUM (MEF). Ethernet in the First Mile: Making Universal Broadband Access a Reality. November 2005b. Disponível em <<http://www.metroethernetforum.org>>. Acesso em: Maio 2009. (MEF, 2005b)

IEEE P802.1ag. Connectivity Fault Management (CFM). Draft Standard 8.1. New York, 2007

BRAY, A. Ethernet OAM & Demarcation Devices: Boxing Clever. ADVA Optical Networking. Webinar March 2006

REQUEST FOR COMMENTS (RFC). Benchmarking Methodology for Network Interconnect Devices. (RFC2544). Network Working group. March 1999.

IEEE P802.1aj. Virtual Bridged Local Area Networks – Amendment 08: Two-Port Media Access Control (MAC) Relay. Draft Standard 3.1 for Local and Metropolitan Area Networks. New York, 2007

ROCHOL, J. Novas arquiteturas em Redes MAN e WAN baseadas na plataforma Ethernet. In: IV Escola Regional de Redes de Computadores (ERRC2006). Passo Fundo, RS-Brasil, 2006.

ZEIDMAN, B. Introduction to CPLD and FPGA Design. President The Chalkboard Network, 2001.

TOROK, D. Projeto Visando a Prototipação do Protocolo de Acesso ao Meio em Redes Ethernet. 2001. Dissertação de Mestrado – Programa de Pós-Graduação em Ciência da Computação da Universidade Católica do Rio Grande do Sul (PUCRS), Porto Alegre, 2001.

ROSE, J. et al. Architecture of field programmable gate arrays. Proceedings of the IEEE, 81, 1013-1029, 1993.

IEEE802.1ad – Provider Bridges. March 2006. Disponível em: <http://www.ieee802.org/1/pages/802.1ad.html> [Acessado em 25 novembro 2008]. (IEEE, 2005b)

MARTIN, G.; CHANG, H. System on Chip Design. In: International Symposium on Integrated Circuits, Devices & Systems (ISIC'01), Tutorial 2, 2001.

JUNEIDI, Z. et al. Global Modeling and Simulation of System-on-Chip embedding MEMS devices. In: International Conference on ASIC, pp. 666 –669, 2001.

SEMATECH. International Sematech. International Technology Roadmap for Semiconductors Update, Capturado em: <http://public.itrs.net>, dezembro 2006.

BENINI, L.; DE MICHELI, G. Powering Networks on Chip. In: International Symposium on System Synthesis, pp. 33 –38, 2001.

RUNNER, S. et al. Building an Infrastructure for IP Reuse. EE Times, May 15, 2000. Capturado em: [www.eetimes.com](http://www.eetimes.com), dezembro 2006.

SEMICONDUCTOR INDUSTRY ASSOCIATION (SIA), The National Technology Roadmap for Semiconductors 1999. Capturado em: <http://public.itrs.net/>, dezembro 2006.

THOMAS, D.; MOORBY, P. The Verilog Hardware Description Language. Kluwer Academic Publishers, p 376, 1991.

PERRY, D. VHDL 3rd Edition, McGraw-Hill, New York, US, 493 p, 1998.

SYSTEM C v2.0.1 Language Reference Manual, Capturado em:

[http://www.systemc.org/projects/systemc/document/SystemC\\_v201\\_LRM/](http://www.systemc.org/projects/systemc/document/SystemC_v201_LRM/), 2003.

MADISETTI, V.; SHEN L. Interface Design for Core-Based Systems. IEEE Design & Test of Computers, v. 14(4), pp. 45-51, October-December 1997.

OST, L. Redes Intra-Chip Parametrizáveis com Interface Padrão para Síntese em Hardware. 2004. Dissertação de Mestrado – Programa de Pós-Graduação em Ciência da Computação da Universidade Católica do Rio Grande do Sul (PUCRS), Porto Alegre, 2004.

GUPTA, R. Introducing Core-Based System Design. IEEE Design & Test of Computers, v.14(4), pp. 15-25, Oct.-Dec. 1997.

MicroBlaze Soft processor Core. Acessado em [http://www.xilinx.com/ise/embedded/mb\\_ref\\_guide.pdf](http://www.xilinx.com/ise/embedded/mb_ref_guide.pdf), 2007.

Xilinx Platform Studio (XPS) and the EDK. Disponível em [http://www.xilinx.com/ise/embedded\\_design\\_prod/platform\\_studio.htm](http://www.xilinx.com/ise/embedded_design_prod/platform_studio.htm)

MARTIN, D. Ethernet Access OAM & Management (EFM OAM & TPMR). Join ITU-T/IEEE workshop on Carrier –class Ethernet

AVNET Inc. Xilinx Virtex-II Pro development Kit. [S.l.], 2003. ADS-003704 Literature Number

HORNA, C. Projeto, Implementação e Validação de IP Soft Cores para Interfaces Ethernet 10/100 e 1000 Mbps sobre Dispositivos Reconfiguráveis. Trabalho Individual (Programa de Pós-graduação em Ciências da Computação, PPGC) – Instituto de Informática, UFRGS, Porto Alegre.

HORNA, C. Implementação e Validação de IP Soft Cores para Interfaces Ethernet 10/100 e 1000 Mbps sobre Dispositivos Reconfiguráveis. In: XIII Workshop Iberchip, Lima-Perú, March 2007.

CHAVAS. Bit-Error Rate Tester of the gigabit links of Xilinx FPGAs. Dissertação de mestrado (Curso di Master in Progettazione Microelettronica) – Università Degli Studi di Padova, Facoltà di Scienze MM. FF. NN. 2007. Disponível em: [http://www.pd.infn.it/~chavas/papers/these\\_master\\_joel.pdf](http://www.pd.infn.it/~chavas/papers/these_master_joel.pdf) [Acessado em 12 agosto 2009].