

## Diminuindo *Overhead* de Software na Comunicação em redes TCP/IP

Vitor Righi<sup>1</sup>, Robert Torrel<sup>1</sup>, Lúcio Renê Prade<sup>1</sup>, Rafael Ramos dos Santos<sup>1</sup>

<sup>1</sup> Universidade de Santa Cruz do Sul

Grupo de Projeto de Sistemas Embarcados e Microeletrônica (GPSEM) – Sala 5341  
Caixa Postal 188 – 96815-900 – Santa Cruz do Sul – RS – Brazil

{vitorrighi, roberttorrel, lucio}@mx2.unisc.br, rsantos@unisc.br

**Resumo.** *O presente trabalho mostra a implementação em hardware de um core Ethernet, com intuito de melhorar o desempenho de comunicação em redes TCP/IP, e tem como base o IP Core disponível na internet denominado Open-Cores MAC 10/100. O MAC está sendo validado através da implementação em hardware reconfigurável e os módulos validados estão no processo de confecção do layout. Após ter-se o MAC validado espera-se melhorar o desempenho da comunicação, pois o encapsulamento e desencapsulamento dos dados para as camadas superiores não será mais feito por software. Para que isto seja possível e para que consiga-se ganhar desempenho, será preciso adaptar o software da pilha TCP/IP para que o mesmo comunique corretamente com o MAC.*

### 1. Introdução

Protocolos de comunicação são geralmente implementados em software, e organizados por camadas. A proposta deste trabalho é primeiramente implementar em hardware as camadas 1 e 2 do modelo OSI (*Open Systems Interconnection*), que é o MAC Ethernet, tanto em FPGA (*Field Programmable Gate Array*) como em ASIC (*Application Specific Integrated Circuit*). Na segunda fase, após o MAC estar completamente testado e validado, será dado início à implementação do protocolo TCP/IP (*Transmission Control Protocol/Internet Protocol*) em hardware. Com estes dois módulos implementados acredita-se que haverá uma diminuição de *overhead* de software, aumentando o desempenho de comunicação em redes TCP/IP. A subcamada MAC, pertence as camadas 1 (Física) e 2 (Link de Dados) do modelo OSI, como mostra a Figura 1.

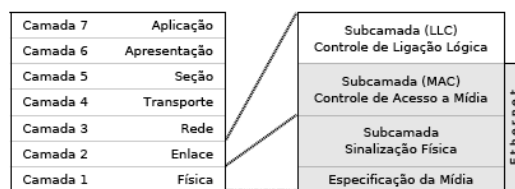


Figura 1. Relação do MAC com o modelo referência OSI.

O MAC controla a transmissão, a recepção e atua diretamente com o meio físico. Consequentemente cada tipo de meio físico requer características diferentes da camada MAC. Sua função é permitir que dispositivos compartilhem a capacidade de transmissão de uma rede, controlando o acesso ao meio de transmissão de modo a se ter um uso ordenado e eficiente deste meio [Tanenbaum 2003].

## 2. A Arquitetura MAC

O MAC estudado utiliza o padrão 802.3, conhecido como Ethernet, que usa o conceito de colisão chamado CSMA/CD (*Carrier-sense Multiple Access with Collision Detection*) para comunicação. O modelo de quadro utilizado no padrão 802.3 é mostrado na Figura 2.

64 bits	48 bits	48 bits	16 bits	46 a 1500 bytes	32 bits
Preâmbulo	Endereço Destino	Endereço Origem	Tipo/Tamanho	Dados	Sequência de Checagem

**Figura 2. Composição do Quadro Ethernet.**

No CSMA/CD a estação primeiro verifica se existe alguém transmitindo no meio e só transmite se este está livre. No caso de detectar uma colisão a estação para de transmitir. O modo de transmissão requer que apenas uma estação transmita enquanto que todas as outras aguardam em silêncio. Esta é uma característica básica de um meio físico compartilhado.

O controle deste processo fica a cargo do método de acesso CSMA/CD onde qualquer estação pode transmitir quando perceber o meio livre. Pode ocorrer que duas ou mais estações tentem transmitir simultaneamente; neste caso, ocorre uma colisão e os pacotes são corrompidos. Quando a colisão é detectada, a estação tenta retransmitir o pacote após um intervalo de tempo aleatório. Isto implica que o CSMA/CD pode estar em três estados: transmitindo, disputando ou inativo. O mecanismo CD obriga que os nós escutem a rede enquanto transmitem dados, razão pela qual o CSMA/CD é também conhecido por *Listen While Talk* (LWT). Como o nó transmissor também escuta a rede, pode detectar a colisão. Neste caso cessa imediatamente a transmissão do pacote e transmite um sinal avisando que ocorreu colisão, que notifica todas as estações. Depois da colisão, o nó espera um período de tempo aleatório e volta a retransmitir. Para evitar colisões sucessivas utiliza-se uma técnica conhecida por "*binary exponential backoff*" [IEEE 2005].

## 3. O IP Core MAC

Inicialmente fez-se uma pesquisa dos *cores* de domínio público disponíveis na internet e optou-se por utilizar o *IP Core* do repositório OpenCores por estar melhor documentado e o código estar descrito na linguagem Verilog (linguagem de descrição de hardware, usada para facilitar o design de circuitos digitais em FPGAs e ASICs). O *core* proposto possui sete submódulos principais, que são, MII, FIFO, Controle, Recepção, Transmissão, Monitoramento, Wishbone [OpenCores.Org 2007].

- MII: responsável por fazer a comunicação entre o MAC e o PHY (Physical Layer).
- FIFO: armazena os dados recebidos a serem enviados as camadas superiores.
- Controle: responsável por controlar o fluxo de dados quando o MAC está operando em 100 Mbps no modo *Full-Duplex*.
- Recepção de quadros: possui uma máquina de estados com seis estados que controla a recepção dos quadros.
- Transmissão de quadros: responsável por enviar os dados em forma de bytes para o módulo de interface Wishbone, possui uma máquina de estados com onze estados.

- Monitoramento: monitora o tráfego de dados do *IP Core* MAC.
- Wishbone: responsável por fazer a comunicação do *IP Core* MAC e dispositivos externos.

Este *IP Core* não poderia ser utilizado da maneira que se encontrava, assim fez-se o estudo do padrão IEEE 802.3 para que se tivesse uma correta verificação e validação. O mesmo possuía algumas inconsistências com o padrão da norma, que foram corrigidas. Pequenas modificações no módulo MII garantiram que este fosse implementado em FPGA e todas suas funcionalidades testadas, assim constatou-se o seu correto funcionamento.

#### 4. Verificação

Os custos envolvidos na etapa de fabricação de um *ASIC* são muito altos, o que requer a validação do circuito antes que os mesmos passem para os demais processos de fabricação, como a confecção do *layout* e do *wafer*.

O processo de verificação de um *IP Core* é a parte mais demorada do projeto e uma das etapas mais importantes. É nesta etapa que verifica-se a funcionalidade do hardware descrito, através da geração de sinais de entrada no módulo que contém o código HDL (*Hardware Description Language*), e da análise dos sinais de saída deste módulo.

Para automatizar este processo, o projetista deve criar um *testbench*, que é o gerador dos sinais de entrada para o código. O *testbench* pode ser escrito na mesma linguagem do *IP Core* e deve seguir um plano de testes onde serão definidas todas as etapas para a verificação funcional. Com isto é possível fazer uma simulação comportamental do hardware, e de acordo com os resultados, verificar se a implementação está funcionando como o especificado.

Para o MAC, os testes devem cobrir todas as funcionalidades de acordo com o padrão IEEE 802.3. Além de garantir que o código está funcional, deverão ser implementado testes em hardware através do uso de FPGA.

Foram feitas simulações comportamentais com *testcases* definidos pelo projetista deste *core* e através destes, uma análise de cobertura de verificação pôde ser feita. Nesta análise, verificou-se que, com os estímulos gerados pelo *testbench* uma grande parte do código HDL (quase 100%) foi atingida, o que garante uma cobertura quase completa do *core*.

#### 5. Projeto Físico

Ao término da verificação de todo o modelo comportamental e garantido que esteja atendendo as normas especificadas, o código é sintetizado e passado para primitivas lógicas.

O módulo MII foi validado, implementado em FPGA, testado em hardware e está sendo feito a confecção do *layout*. Foi escolhido o módulo MII para ser feito a confecção do *layout* por ser um módulo que interage diretamente com o hardware (PHY), tornando os testes mais fáceis de serem implementados. Este passo só será possível se todo o MAC estiver validado, sendo assim, o MAC sem o MII irá ser implementado em FPGA, utilizando a placa Virtex II Pro da Xilinx. O *ASIC* do MII ficará conectado entre FPGA e o PHY, como mostra a Figura 3.

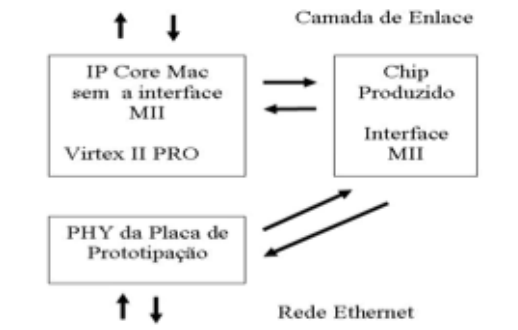


Figura 3. Esquema de ligação de testes do ASIC MII.

## 6. Conclusões

Este trabalho tem como objetivo principal estudar e implementar em hardware um *core* para a interface Ethernet 10/100 Mbps. Além de obter um melhor desempenho com a implementação do MAC e do TCP/IP em hardware, poder-se-á utilizar estes módulos em outros projetos em que o hardware necessite se comunicar com o meio externo, podendo-se fazer otimizações ou customização de cada módulo conforme necessidades de cada projeto. Observou-se também que a parte de validação em um projeto de hardware é extremamente importante, pois a cada etapa que o projeto avança o nível de abstração fica mais complexo, tornando assim mais difícil detectar e corrigir possíveis erros. Finalmente observou-se a importância deste trabalho para ter-se o domínio de tecnologias que estão em grande ascendência como a área de comunicação de dados e microeletrônica.

## Referências

- IEEE (2005). Institute of electrical and electronics engineers, inc., iee 802.3.
- OpenCores.Org (2007). Repositório de ip cores. <http://www.opencores.org>.
- Tanenbaum, A. S. (2003). *Redes de computadores*. Elsevier Rio de Janeiro.