5
Realização do equipamento

Neste capítulo vamos detalhar todo o processo de desenvolvimento e realização do equipamento de teste. São descritos os equipamentos usados e a arquitetura adotada para a solução, e mencionados problemas que foram contornados ao longo do trabalho, além das limitações práticas encontradas.

5.1.
Detalhes da tecnologia adotada

A primeira grande decisão de um projeto envolvendo tecnologia FPGA está exatamente na escolha dos componentes que serão usados. Além dos *chips*, que existem em inúmeras versões, outras peças fundamentais à solução final devem ser definidas quanto a sua existência e quantidade. Memórias, portas de comunicação de diferentes padrões, freqüência de operação, conversores A/D são elementos fundamentais no desenvolvimento de um projeto. O principal entrave está no fato de que não é fácil estimar no início de um projeto os requisitos exatos que serão necessários. Mesmo desenvolvedores com grande experiência podem preferir não precisar todos os requisitos. Exatamente para diminuir os impactos dessa realidade, inúmeros fabricantes vendem placas chamadas de "desenvolvimento". São placas, às vezes de alto custo, que possuem diversos recursos que possam vir a ser necessários no desenvolvimento do projeto. Ainda assim, existem variações de placas de desenvolvimento com diferentes características e recursos, o que obriga os desenvolvedores a ter uma mínima idéia da dimensão do seu projeto. Geralmente, opta-se por iniciar os trabalhos em placas com recursos superdimensionados aos quais se acredita precisar. O aparente desperdício será convertido em maior maleabilidade ao longo do trabalho e permitirá a otimização do produto final para implantação em *hardware* adequado.

Neste trabalho foi usada uma placa de desenvolvimento da empresa Xilinx, modelo ML405. Dentre as principais características dessa placa, podemos citar:

- *chip* FPGA fixo, modelo XC4VFX20-FF668, família Virtex4[11]. Este *chip* possui um processador Power-PC embarcado, permitindo execução simultânea de programas. Possui ainda 4 blocos DCM (*Digital Clock Manager*) e 8 MGT (*Multi-Gigabit Transceiver*) que serão importantes neste trabalho e posteriormente melhor explicados;
- entrada de cartão de memória CompactFlash, para transporte de programação;
- porta serial UART, RS-232[12];
- 256 MB de memória tipo DDR2 DIMM;
- 64 MB de memória DDR SDRAM;
- saída Ethernet em cobre, padrão 1000Base-T;
- conector SFP para instalação de transceptores de cobre ou fibra-óptica;
- conectores comuns em padrões de computadores pessoais, ideal para este desenvolvimento, como USB, SATA, PS/2 e VGA.

As placas de desenvolvimento permitem aos desenvolvedores trabalhar com maior flexibilidade e menos restrições. Em trabalhos que envolvem a prova de conceitos, são muito importantes: podem ser as principais responsáveis pela viabilidade do desenvolvimento. A criação de uma placa, ou seja, projeto e montagem do protótipo, é um processo bastante caro e não pode ser repetido inúmeras vezes ao longo do desenvolvimento de um projeto, sob pena de exigir um orçamento elevadíssimo. Idealmente, o produto final é montado apenas quando praticamente todos os trabalhos e estudos envolvidos já estão concluídos. Ainda assim, é pratica comum dos fabricantes produzir placas prevendo alterações futuras ou que possam servir a várias versões e modelos futuros daquele produto.

5.1.1. O Processador embarcado Power-PC

O *chip* FPGA disponível possui embutido um microprocessador PowerPC, fisicamente construído, modelo 405 de 32 bits, desenvolvido pela IBM e voltado

para dispositivos embarcados SoC (*System-on-a-chip*)[13], sendo capaz de operar até 450 MHz.

A principal vantagem do processador está na possibilidade de incorporar lógicas em programas (*software*) à solução. Já foi mencionado que a lógica implantada no *hardware* FPGA é sempre limitada pela capacidade do *chip,* e que lógicas mais complexas ocupam mais blocos do FPGA, mas deve-se ter ciência de que as lógicas executadas em programa levam mais tempo para fornecer os resultados. Uma dúvida comum que pode surgir é de que, se optamos por um FPGA porque precisamos de alto desempenho, qual seria a vantagem de transferirmos parte do trabalho para um mecanismo que oferecerá baixo desempenho? A resposta está no fato de que, na maioria das aplicações, nem todas as lógicas que a compõem, precisam de alto desempenho. O processador nos oferece a oportunidade de transferir lógicas complexas e com menor exigência de desempenho, para programas que serão executados pelo processador. Cabe ressaltar que aplicações integralmente baseadas em *software* costumam ter custos muito inferiores.

Por exemplo, imagine que a um determinado equipamento esteja conectada uma tela de cristal líquido, responsável por exibir ao operador informações básicas sobre o andamento do trabalho. Telas, em geral, possuem tempo de atualização na ordem de milissegundos e, se atualizássemos as informações exibidas a esta velocidade, seria impossível a um ser humano acompanhá-las. A tarefa de montar informações para exibição em telas pode ser complexa, por envolver a montagem e alinhamento de textos ou geração de imagens, exigindo grande esforço para ser realizada e desenvolvida em lógica FPGA. No entanto, por possuir baixa exigência de velocidade, em comparação ao desempenho de lógica FPGA, pode ser realizada no processador.

Neste trabalho, as tarefas básicas de controle e comunicação com um microcomputador externo, serão realizadas por programas sendo executados no processador. As ferramentas de desenvolvimento da Xilinx que serão usadas possuem compilador cruzado para o processador PowerPC, na linguagem C/C++.

5.1.2.
DCM - *Digital Clock Manager*

O DCM[14] é um gerenciador digital de *clock*, capaz de operar como sintetizador flexível de freqüência, por multiplicação e/ou divisão, e produzir saídas com defasagem configurável. Todo circuito digital, baseado em FPGA, deve possuir um *clock* próprio, normalmente através de cristal, em freqüência fixa. É possível, no entanto, que diferentes componentes do circuito trabalhem em freqüências distintas e, no caso de um *chip* FPGA, isso pode ocorrer entre blocos implantados dentro do mesmo *chip*. Como a introdução de outros cristais para a criação de outros *clocks* aumenta o custo e a complexidade do projeto, o DCM se mostra como uma excelente saída para a obtenção destes sinais, sem a necessidade de alterações no projeto de construção do circuito. O DCM também pode ser usado para permitir a comunicação com interfaces externas como memórias, portas de comunicação ou unidades de disco, em suas freqüências específicas de trabalho.

O funcionamento básico de um bloco DCM se dá pela injeção de um sinal de referência, conhecido, e uma de suas saídas em uma porta específica de referência. É essa realimentação que permitirá ao bloco funcionar com estabilidade.

A figura abaixo representa um módulo DCM típico:

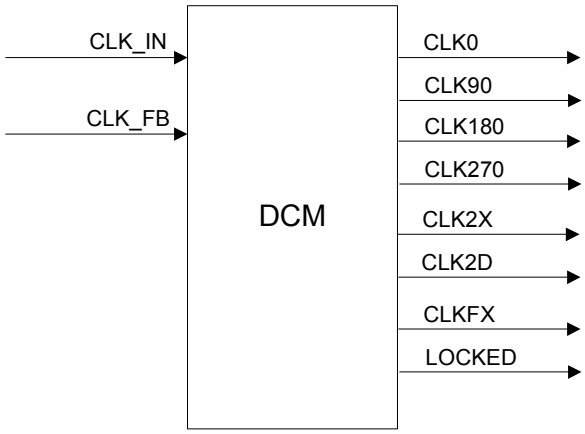

Figura 10 – Bloco DCM

- CLK_IN: Entrada de sinal de *clock*;
- CLK_FB: Entrada de referência. Pode ser ligado às portas CLK0 ou CLK2X dependendo do objetivo desejado;
- CLK0: Saída com a mesma freqüência da entrada;
- CLK90: Saída com desvio de 90° em relação a CLK0;
- CLK180: Saída com desvio de 180° em relação a CLK0;
- CLK270: Saída com desvio de 270° em relação a CLK0;
- CLK2X: Saída com a mesma fase e dobro da freqüência de CLK0;
- CLK2D: Saída com a mesma fase e a metade da freqüência de CLK0;
- CLKFX: Saída com freqüência sintetizada, conforme multiplicador e divisor programados. Mesma fase de CLK0;
- LOCKED: Indica que os sinais de saída estão estáveis.

5.1.3. MGT - Multi-Gigabit Transceiver

O bloco MGT[15] é um componente capaz de transmitir e receber dados em altas velocidades visando comunicação através de diversos padrões de redes e dispositivos de armazenamento. O MGT disponível na família Virtex-4, chamado de MGT Rocket-IO Transceiver, pode operar entre 622 Mb/s e 6,5 Gb/s, faixa na qual trabalham vários padrões, entre eles:

Padrão	Taxa (Gb/s)
Gigabit Ethernet	1,25
SONET OC-12	0,622
Fibre-Channel	1,06 / 2,12 / 4,25
PCI Express	2,5
Serial ATA	1,5 / 3,0

Tabela 2 – Velocidades de operação de alguns padrões de comunicação, que podem ser obtidas com o uso do MGT

O MGT ainda é capaz de codificar os dados conforme padrões comuns e pré-definidos, ou programados em lógica FPGA. Será de especial interesse para esta dissertação o padrão 8B/10B[2] usado nas aplicações Gigabit-Ethernet.

5.1.4. PLB – Processor Local Bus

No capítulo 2 foi apresentado o barramento OPB, ferramenta responsável por permitir a comunicação entre diversos dispositivos ou módulos implantados no *chip* FPGA. Conforme foi dito, o OPB não possui alta capacidade de transferência de dados, sendo, portanto, inadequado para dispositivos que requeiram trabalhar em alta velocidade. O exemplo mais comum é o processador que precisa de respostas o mais curtas possível ao acessar ou gravar dados em memória RAM. Para atender especialmente esta necessidade, foi desenvolvido um barramento de dados veloz, chamado de PLB[16].

O barramento PLB será usado por um grupo seleto de dispositivos e é completamente independente de barramentos OPB. Para permitir a comunicação entre dispositivos conectados ao OPB e ao PLB, deverá ser usada uma ponte, que fará a tradução de endereçamento entre os dois barramentos. Um exemplo de uso da ponte é a possibilidade de permitir a um programa rodando no processador, buscar ou enviar informações para módulos conectados no OPB. A figura abaixo demonstra o uso da ponte PLB-OPB[17].

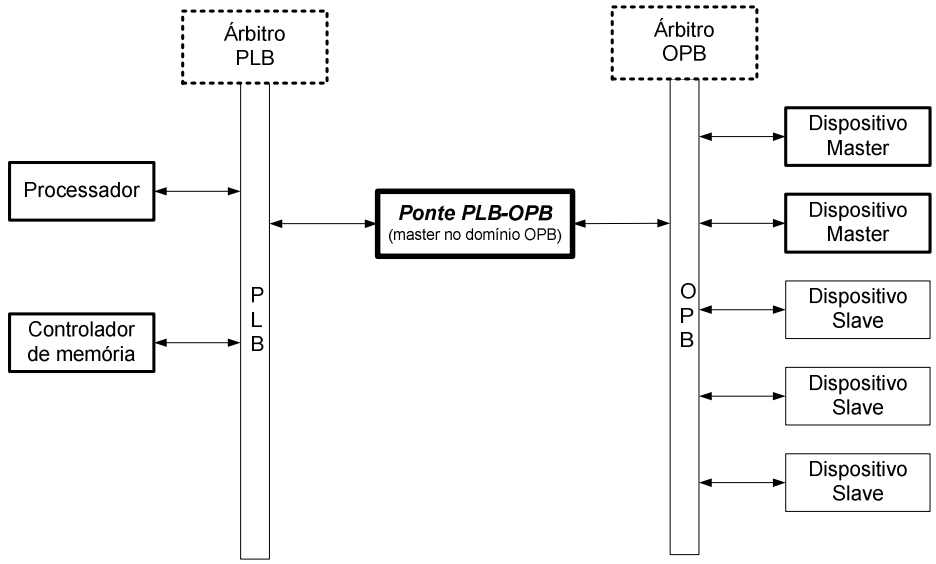

Figura 11 – Diagrama dos barramentos PLB e OPB, e a ponte que permite a comunicação entre eles

5.2. Módulos

A partir deste ponto, vamos começar a detalhar tecnicamente todo o desenvolvimento da solução, desde a lógica inicial até o primeiro protótipo. No capitulo 4 foi apresentada uma curta idéia da organização que seria adotada neste projeto, como base para a compreensão da lógica de trabalho descrita na seqüência.

Conforme foi mencionado, o equipamento de testes será composto de, basicamente, 3 módulos: transmissão, recepção e controle. O funcionamento básico de cada um deles será descrito em detalhes, assim como suas conexões.

5.2.1. Módulo de transmissão

É responsável por gerar um número pré-determinado de quadros, todos com um mesmo tamanho fixo, também pré-determinado. O controle do módulo de transmissão é realizado pelo módulo controlador, através de uma série de sinais

entre ambos. Através destes sinais, que serão a seguir relacionados, o processo de transmissão é configurado e ativado. O diagrama abaixo representa o módulo transmissor relativo à primeira versão apresentada, capaz de avaliar apenas a perda de quadros.

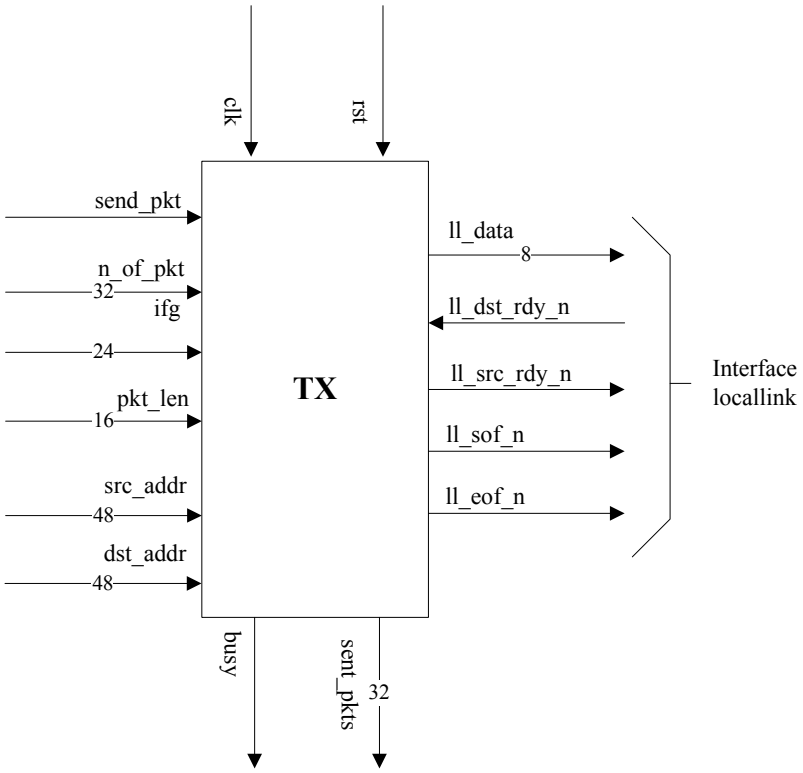

Figura 12 – Diagrama do módulo transmissor – primeira versão proposta

Sinais do módulo controlador para o módulo transmissor:

- dst_addr – 48 bits – Endereço MAC de destino que será inserido nos quadros;

- src_addr – 48 bits – Endereço MAC de origem que será inserido nos quadros. Na recepção, apenas os quadros com determinados endereços de origem e destino (src_addr e dst_addr) serão analisados;

- pkt_len – 16 bits – Tamanho fixo, em octetos, dos quadros transmitidos;

- IFG – 24 bits – Comprimento, em octetos, do intervalo entre o fim da transmissão de um quadro e o início do seguinte. É importante para permitir controle da taxa de transmissão desejada;

- n_of_pkt – 32 bits – Define o número de quadros que serão transmitidos no teste, ininterruptamente;

- send_pkt – 1 bit – Deve ser ligado pelo controlador durante 1 ciclo de CPU, o suficiente para o transmissor perceber sua existência e iniciar o processo de transmissão;

Sinais do módulo transmissor para o módulo controlador

- busy – 1 bit – Indica que há um teste em andamento;

- sent_pkts – 32 bits – Informação do número de quadros já transmitidos. Esse dado permite o acompanhamento do teste e é usado pelo módulo de controle no cálculo de estatísticas em tempo-real.

Sinais de uso comum: sinais que são compartilhados por todos, ou parte dos módulos da solução.

- Clk – 1 bit – Sinal de *clock*. É comum aos módulos de transmissão, recepção e controle de maneira que todos trabalhem sincronamente. Este sinal também alimenta as lógicas de rede em camadas inferiores, sendo a referência de tempo para a comunicação pela interface *local-link*;

- Rst – 1 bit – Sinal de ‘Reset’, compartilhado por todos os módulos da solução que, ao perceberem sua presença, retornam ao estado inicial, interrompendo qualquer operação em curso.

Conforme foi descrito no capítulo anterior, para possibilitar a avaliação do retardo dos quadros, será necessário incluir em cada quadro a informação do instante em que foi transmitido. Abaixo, o diagrama do bloco de

transmissão com a inclusão de uma entrada do relógio de onde será retirada essa nova informação, representada pelo sinal ‘delay_clk’, de 48 bits, em destaque:

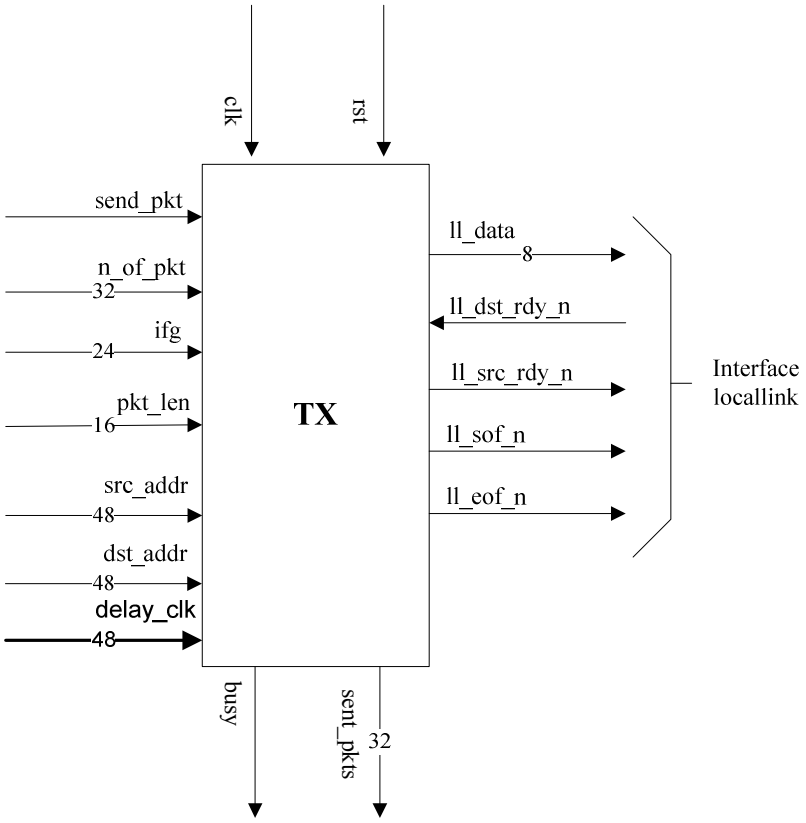

Figura 13 - Diagrama do módulo transmissor – segunda versão proposta, incluindo informações necessárias para o cálculo do retardo.

5.2.1.1. Cálculo do intervalo entre quadros (IFG)

O sinal IFG define o intervalo de tempo que o transmissor deve aguardar entre o término de um quadro e o início de outro. O IFG é o mecanismo que permite controlar a taxa de transmissão durante o teste. Uma vez que a velocidade de transmissão do GbE, ou velocidade de linha, é constante e igual a 1,25 Gbps, é necessário reduzir a taxa de transmissão de quadros para conseguirmos diminuir a taxa média de dados transmitidos.

A unidade que foi definida neste trabalho para o IFG equivale ao tempo de transmissão de um octeto no padrão Gigabit Ethernet. O motivo para esta escolha é devido ao fato de que, segundo o padrão, o intervalo entre dois quadros deve ser múltiplo de um octeto, ou byte.

Seja R a taxa de transmissão do Gigabit Ethernet

$$R = 1{,}25Gbps = 1{,}25 \times 10^9 bits/s$$

Como o Gigabit Ethernet utiliza, em quase[1] todas as configurações, a codificação 8B/10[2] que transforma cada bloco de 8 bits em 10 bits, a taxa de dados transmitidos, excluindo-se o adicional devido à codificação, será:

$$R = \frac{1{,}25 \times 10^9}{10} = 1{,}25 \times 10^8 bytes/s$$

Portanto, o tempo de transmissão de um byte de dados é:

$$T = \frac{1}{1{,}25 \times 10^8} = 8ns$$

A taxa de quadros transmitidos é:

$$FR = \frac{R}{preâmbulo + L + ifg} \text{ (quadros/s)}$$

Sendo:

- preâmbulo: limitador dos quadros Ethernet – 64 bits;
- L: tamanho dos quadros (supondo todos os quadros com o mesmo tamanho);
- IFG: 12 bytes (mínimo) * 8

  O tamanho mínimo do IFG, definido pelo padrão, é de 12 bytes. A taxa de transmissão de quadros e dados será controlada pela variação do tamanho deste intervalo.

A taxa de transmissão de dados (Rd) pode então ser calculada como:

[1] Exceto em cabo de par trançado, padrão 1000Base-T

$$Rd = FR \times L$$

$$Rd = \frac{R}{preâmbulo + L + IFG} \times L$$

Dada a taxa de transmissão de dados desejada para o teste, calculamos o IFG, em octetos, a ser usado:

$$IFG = (\frac{R \times L}{Rd}) - preâmbulo - L$$

5.2.1.2. Capacidade máxima de transmissão

O item anterior dá uma informação sobre a capacidade de transmissão no padrão Ethernet: devido a existência do preâmbulo e do IFG mínimo, a máxima taxa de transmissão de dados, incluindo os cabeçalhos, poderá ser próxima, mas nunca igual à taxa líquida de bits transmitidos.

Quanto maior o tamanho dos quadros transmitidos, proporcionalmente menor será o efeito desses intervalos obrigatórios sobre a capacidade máxima de transmissão.

Como o IFG deve ter um valor mínimo de 12 octetos e o preâmbulo de 8 octetos, podemos calcular a máxima taxa de quadros e bits transmitidos para determinados tamanhos de quadro:

L (bytes)	Máx FR	Máx Mbps	Rel Tx Max (%)
64	1.488.095	761,90	76,2
128	844.595	864,86	86,5
256	452.899	927,54	92,8
512	234.962	962,41	96,2
768	158.629	974,62	97,5
1024	119.732	980,84	98,1
1280	96.154	984,62	98,5
1518	81.274	987,00	98,7

Tabela 3 – Máxima capacidade de transmissão[18], no padrão Gigabit-Ethernet, para quadros de diferentes tamanhos

Onde:

- L: tamanho (fixo) dos quadros transmitidos;
- FR: taxa de quadros transmitidos por segundo;
- Rel Tx: relação porcentual da máxima taxa de transmissão em relação à taxa de linha de 1 Gbps.

A relação entre a máxima taxa de transmissão, o IFG utilizado e o tamanho do quadro é expressa no gráfico a seguir, onde cada curva está associada à dimensão do quadro utilizado:

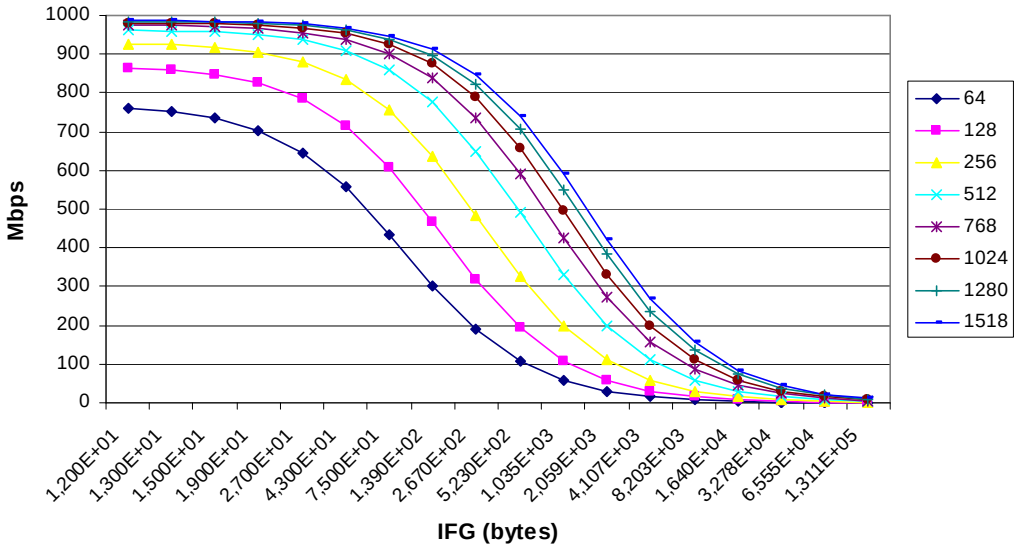

Figura 14 – Relação entre a taxa de transmissão e o IFG, para diferentes tamanhos de quadro

5.2.1.3. Fluxograma do módulo de transmissão

O fluxograma a seguir ilustra a lógica elaborada para o módulo transmissor e que foi utilizada na descrição da máquina de estados que coordena sua operação.

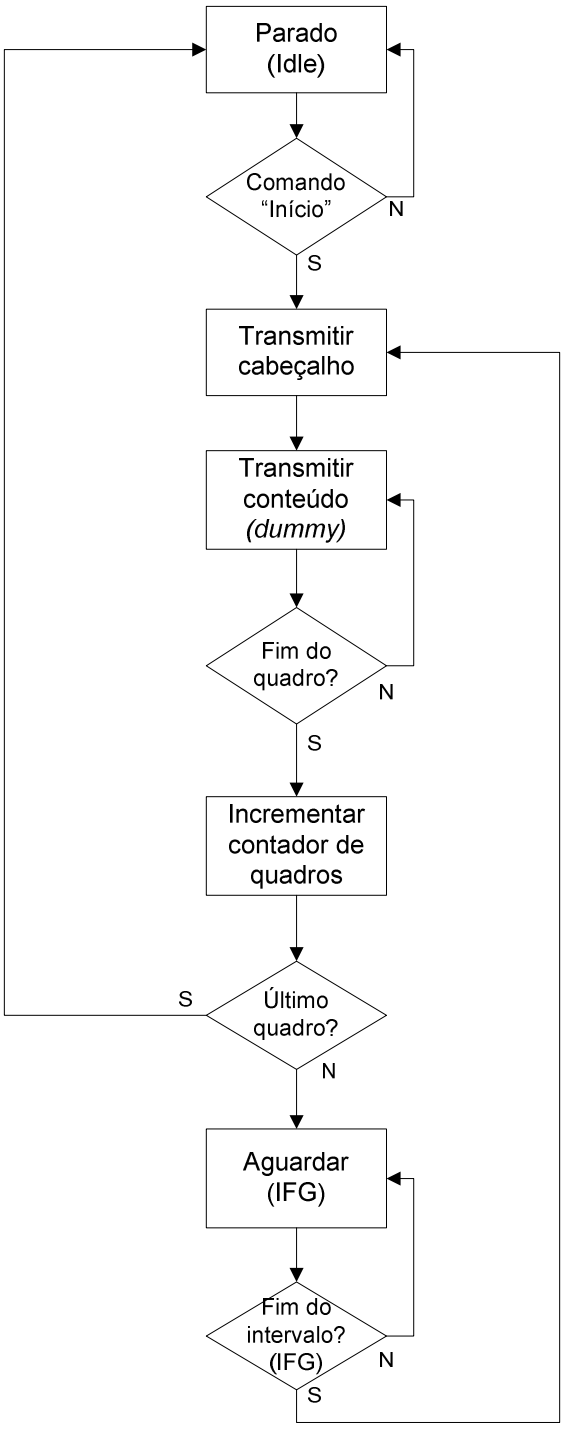

Figura 15 – Fluxograma da lógica do módulo transmissor, que descreve a sua máquina de estados

O conteúdo do quadro é composto, além dos cabeçalhos e identificadores relativos ao teste, de octetos para complementação do tamanho desejado. Esse preenchimento pode ser feito com bytes aleatórios ou seqüenciais. É de grande importância ressaltar que o módulo transmissor não é responsável pela montagem final dos quadros. Essa incumbência é designada às camadas inferiores do padrão

Ethernet, para quem o módulo transmissor entrega apenas uma parte do quadro final. Dentre as funções das camadas inferiores está o delineamento dos quadros, por meio do preâmbulo, o cálculo do código verificador de erro, CRC, e a codificação 8B/10B.

5.3. Módulo de recepção

O módulo de recepção é responsável por receber os quadros transmitidos e realizar as avaliações estatísticas iniciais. As duas versões mencionadas no capítulo 4 serão apresentadas. A primeira, apenas com a avaliação de quadros perdidos, e a segunda, capaz de também avaliar os retardos envolvidos.

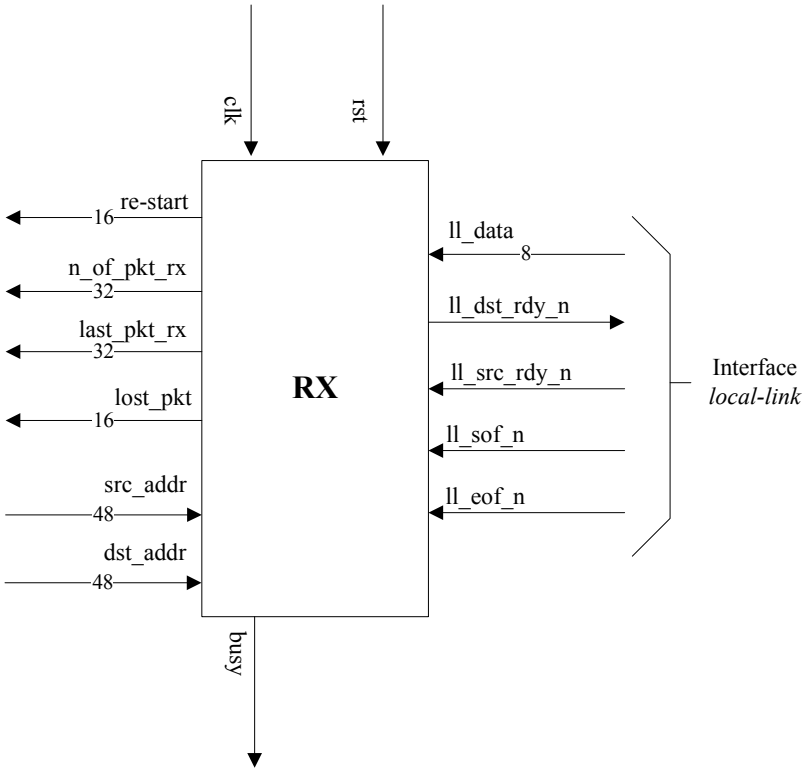

Figura 16 – Diagrama do módulo receptor – primeira versão proposta

Sinais do módulo controlador para o módulo receptor:

- dst_addr: 48 bits – endereço MAC de destino inserido nos quadros;

- src_addr: 48 bits – endereço MAC de origem inserido nos quadros. Apenas os quadros com os respectivos endereços de origem e destino (src_addr e dst_addr) são analisados;

- Re-start – 16 bits – Indica o número de testes que foram iniciados desde o último *reset*. Vinculado a um contador, é útil para permitir a identificação de eventuais problemas. Esse valor só pode sofrer incremento no início de um teste. Caso exista a ocorrência deste evento em outros momentos, pode indicar recebimento duplicado de quadros ou a presença de elementos interferentes na rede. Os procedimentos a adotar diante desses cenários variam conforme a necessidade e objetivos das pessoas responsáveis, sendo, o mais simples deles, a emissão de um alerta visual ou sonoro pelo software no computador;

- n_of_pkt_rx: 32 bits – indica o número de quadros recebidos desde o início do teste;

- last_pkt_rx: 32 bits – indica o número identificador do último quadro recebido até aquele instante;

- lost_pkt: 32 bits – indica o número de quadros perdidos, armazenado no acumulador correspondente;

- busy: 1 bit – indica que o módulo está processando um teste no presente momento. O sinal ‘busy’ é ativado quando o primeiro quadro, com a identificação adequada, é recebido; é desligado após o recebimento do último quadro ou um determinado tempo após o módulo transmissor tê-lo enviado. Esse intervalo de tempo é chamado de *timeout*.

Sinais de uso comum: sinais que são compartilhados por todos, ou parte dos módulos da solução.

- Clk: 1 bit – sinal de c*lock*. É comum aos módulos de transmissão, recepção, controle e blocos de rede (MAC), de maneira que todos trabalhem sincronamente;

- Rst: 1 bit – sinal de *reset*, compartilhado por todos os módulos da solução que, ao perceberem sua presença, retornam ao estado inicial, interrompendo qualquer operação em curso.

A segunda versão do módulo de recepção foi desenvolvida conforme indicado pelo segundo modelo apresentado, e é capaz de aferir o tempo médio de retardo sofrido pelos quadros.

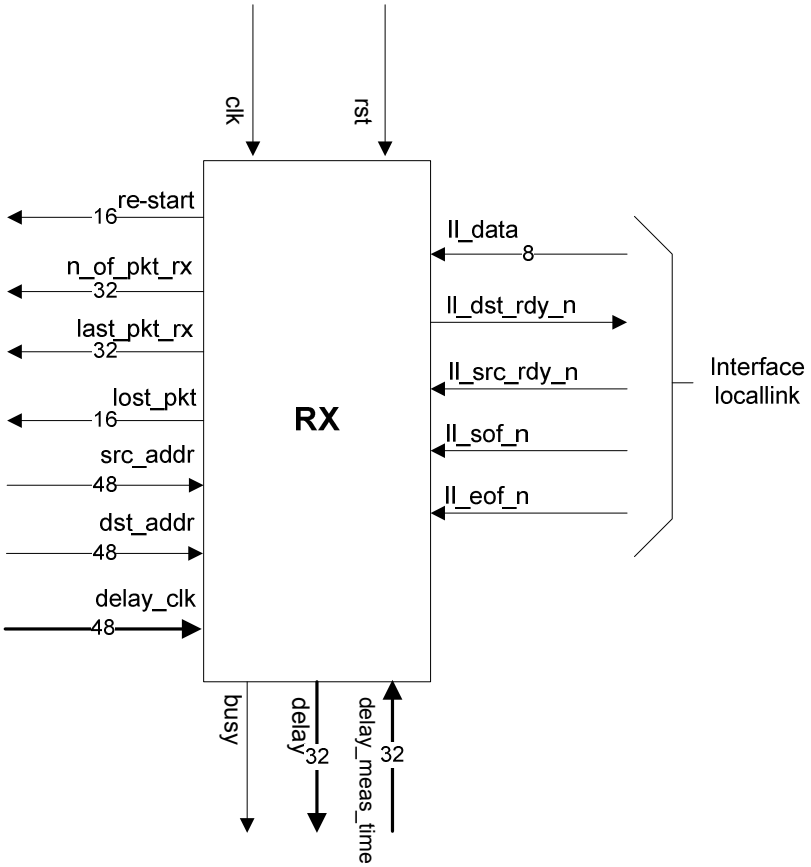

Figura 17 - Diagrama do módulo receptor – segunda versão proposta, capaz de calcular o retardo dos quadros

O diagrama do bloco é similar ao anterior, porém com a adição de novos sinais ligados ao módulo controlador. São eles:

- delay_clk: 48 bits – recebe o valor corrente do contador de tempo do teste, usado como base para a avaliação do retardo;

- delay: 32 bits – indica o retardo médio sofrido pelos quadros no último intervalo de tempo avaliado. A unidade é a mesma base de tempo usada pelo contador de tempo;

- delay_meas_time: 32 bits – define o intervalo de tempo que será usado pelo módulo, com base no relógio mestre, para a avaliação do retardo médio. Esse intervalo deve ser definido considerando-se o período com que as estatísticas do teste serão repassadas pelo módulo de controle ao computador.

5.3.1. Fluxograma do módulo de recepção

* Um ciclo de testes chega ao fim em duas situações: quando o último quadro da seqüência é recebido, ou após um tempo limite sem o recebimento de novos quadros *timeout*. Em ambos os casos, será necessário o cálculo do retardo médio em relação ao tempo decorrido desde a última avaliação. Na hipótese de término por *timeout*, o número de quadros restantes deverá ser somado a contagem de quadros perdidos. Lembrando que este valor é conhecido pelo módulo receptor devido à transmissão da numeração dos quadros em ordem decrescente. O último quadro de um teste é identificado quando a sua numeração for igual a zero.

** O trecho de lógica no interior do retângulo tracejado (figura 18) é específica da segunda versão do módulo receptor, capaz de avaliar o retardo médio dos quadros. A primeira versão é igual à segunda, excluindo-se apenas o conteúdo deste retângulo.

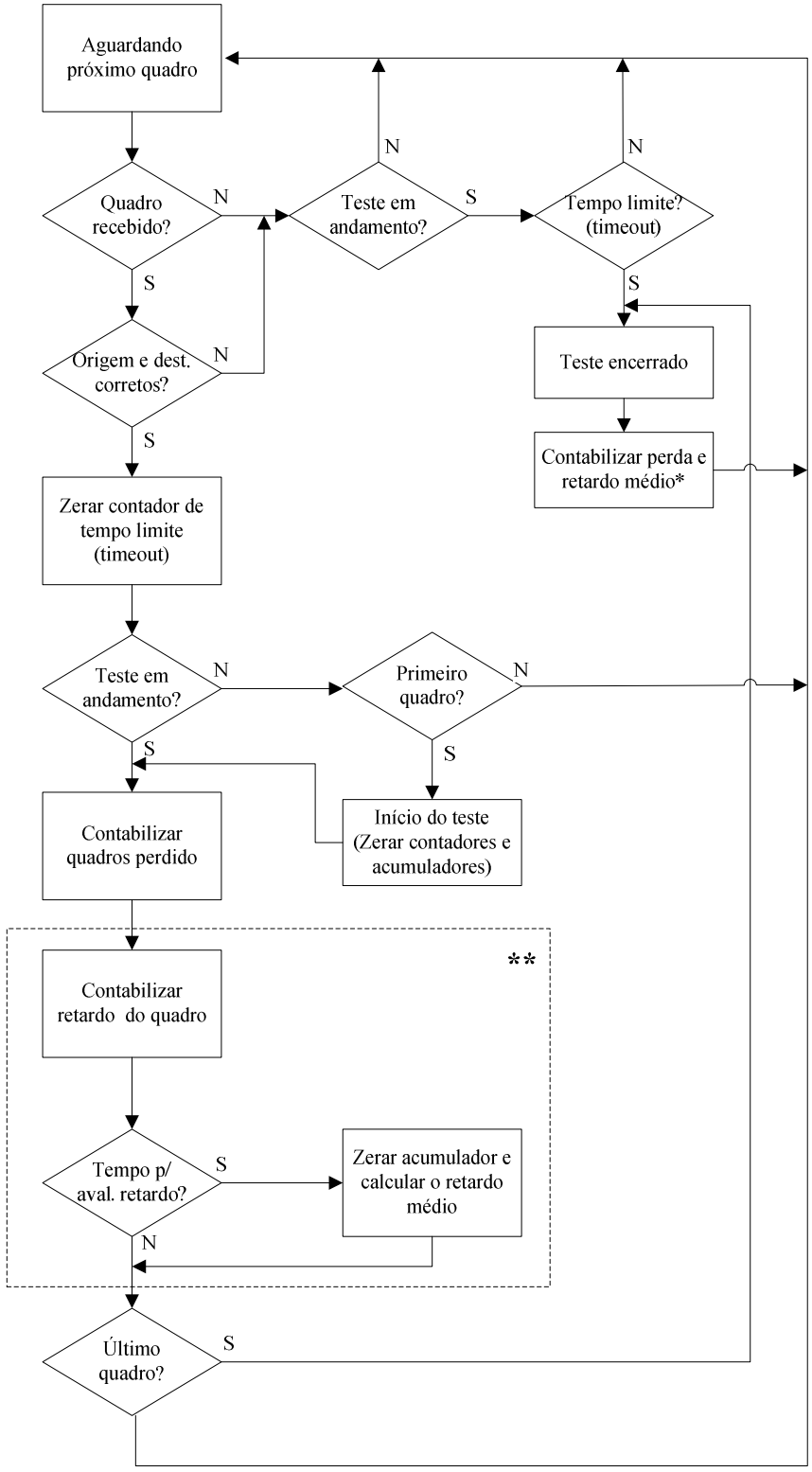

Figura 18 – Diagrama detalhado da lógica do módulo receptor, que descreve a sua máquina de estados

5.3.2.
Envio de quadros para aprendizado

Antes de iniciar uma seqüência de testes, é importante ter certeza de que o dispositivo, ou dispositivos, ao longo do teste saberão encaminhar os quadros corretamente, até serem entregues ao módulo receptor. Para garantir este conhecimento, o módulo receptor deve transmitir, antes do início do teste, um quadro de aprendizado. O recebimento desse quadro pelos dispositivos resultará na atualização das suas tabelas de encaminhamento, de forma que estes saberão encaminhar corretamente os quadros destinados ao módulo receptor.

O quadro de aprendizado varia conforme o ambiente em teste. Em redes com o uso de IP, ele pode ser um pacote do tipo ARP[19], que informará a associação do endereço IP ao endereço MAC do receptor, além de ensinar aos elementos comutadores, como *switches*, a porta onde o receptor está conectado. No caso deste trabalho, em que focamos apenas o padrão Ethernet, sem nos preocupar com demais protocolos de camada superior, o quadro de aprendizado é mais simples, sendo apenas necessário transmitir um quadro tipo *broadcast*[1], sem necessidade de preenchimento com conteúdo específico. Certamente, a evolução deste equipamento exigirá mecanismos mais sofisticados e capazes de atender aos demais casos de aprendizado que possam ser necessários.

5.4.
Controle e comunicação externa

Completando a solução, vamos descrever os mecanismos responsáveis pelo controle das funções básicas dos módulos de transmissão e recepção, e como a comunicação externa, com o microcomputador, é realizada. Essas tarefas estão distribuídas em dois módulos distintos que, comunicando-se entre si, permitem a transferência de comandos de operação e a apresentação dos resultados obtidos ao longo do teste. A primeira parte da lógica está implantada em um bloco de *hardware*, no FPGA, conectado diretamente aos módulos de transmissão e

[1] O quadro Ethernet, onde todos os bits do campo endereço de destino são ‘1’, é chamado de *broadcast* e é encaminhado a todas as estações conectadas àquela rede.

recepção e operando sincronamente com eles. A segunda é composta por um programa sendo executado no microprocessador PowerPC. Como já foi mencionado anteriormente, a comunicação entre esses dois módulos é realizada através dos barramentos PLB e OPB, lembrando que a troca de informações entre os dois se dá através de uma ponte PLB-OBP.

A figura abaixo ilustra o diagrama do bloco de controle no circuito FPGA.

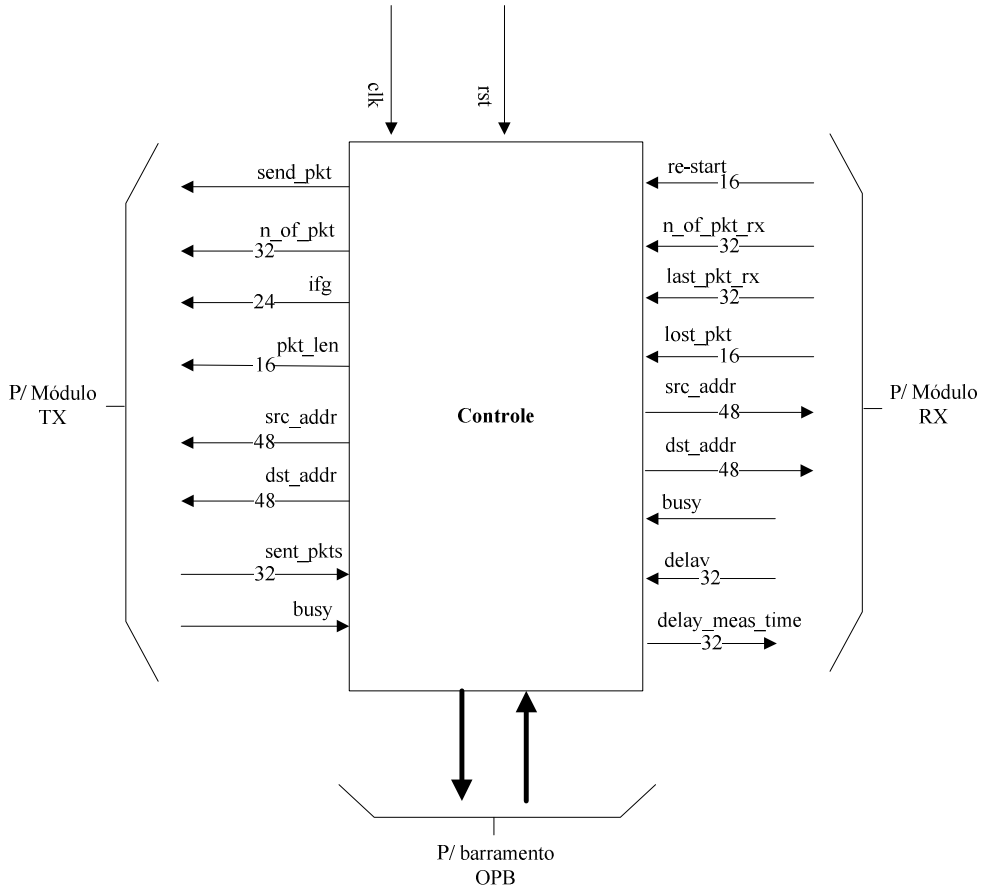

Figura 19 - Diagrama do módulo de controle

Um detalhe de extrema importância está no fato de que o sinal de *clock* é o mesmo para os três módulos: de controle, transmissor e receptor. O sincronismo entre os três é necessário para garantir a integridade das informações propagadas nos sinais entre eles. Se as freqüências fossem distintas, ou se os sinais estivessem defasados, uma informação poderia ser lida durante o período em que estava sendo atualizada, ou seja, seria obtido um valor incorreto.

O barramento OPB, por definição, utiliza um sinal próprio de *clock*. Apesar de ser possível injetar o mesmo sinal já compartilhado pelos outros módulos, devemos lembrar que a estrutura criada para a implantação do barramento possui uma considerável complexidade. O sinal de *clock* usado no processo de transmissão e recepção deste trabalho visa a operação em uma taxa elevada, e utilizá-lo para alimentar o barramento OPB criará grandes restrições ao tempo máximo de propagação do sinal através das linhas de roteamento do FPGA. Por isso optamos, neste trabalho, por alimentar o barramento OPB com a metade da freqüência do *clock* original.

O sinal de *clock* básico de 125 MHz, gerado por um cristal, foi dividido por dois, com o uso de um bloco DCM, e só então entregue ao barramento OPB. Este sinal é obrigatoriamente usado em todas as comunicações com o barramento, que trabalha de maneira síncrona. Pelo barramento OPB chegam requisições, que podem ser comandos ou pedidos de informação, conforme protocolo definido na implementação.

5.4.1. Isolamento do sinal de *clock* do barramento OPB

Os sinais que ativam o OPB e o *local-link* possuem freqüências distintas ou defasadas. No presente trabalho, a freqüência do sinal OPB resultará da divisão por dois da freqüência do *local-link*, estando estes sinais alinhados em fase. Desta forma, será garantido que uma determinada transição positiva de referência sempre ocorrerá no sinal de menor freqüência junto com o sinal de freqüência superior. Como conseqüência, existe a garantia de integridade dos dados disponíveis nos sinais neste instante. Uma solução mais segura, capaz de trabalhar independentemente dos sinais de *clock*, envolve o uso de registradores intermediários e um semáforo responsável por sinalizar a disponibilidade das informações.

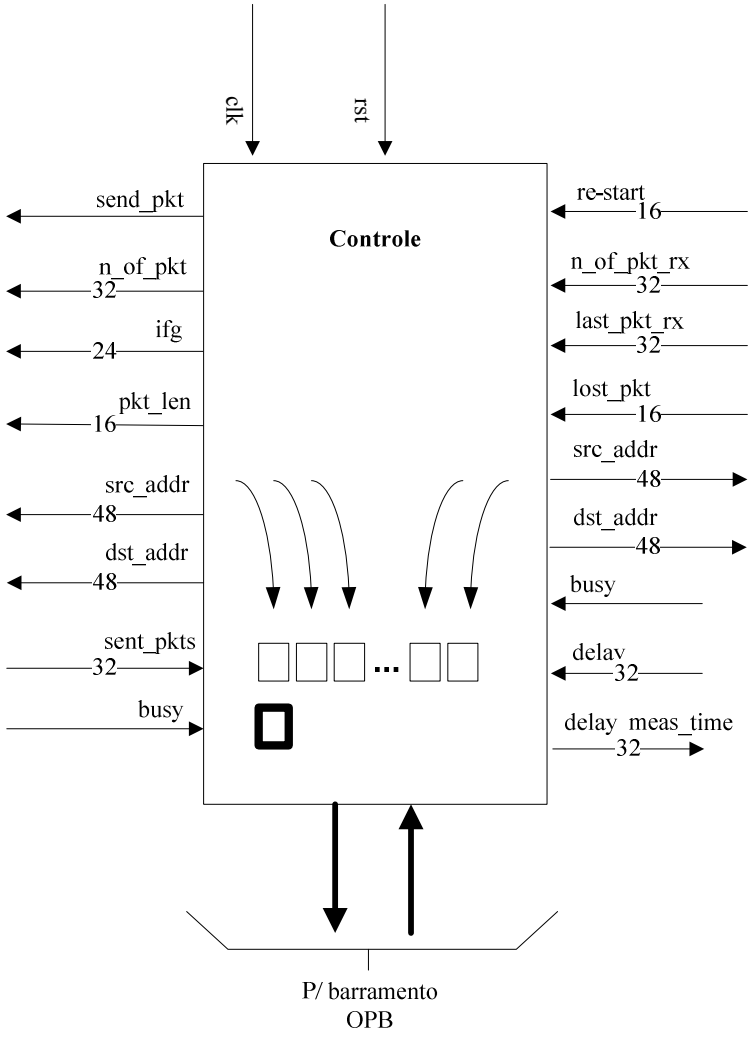

Figura 20 – Diagrama do mecanismo de troca de informações entre pontos alimentados por sinais de *clock* distintos

No diagrama acima, uma evolução do anteriormente apresentado, os blocos em cinza no interior do módulo representam esses registradores intermediários, enquanto o bloco em destaque representa o semáforo de disponibilidade dos dados. O funcionamento é bastante simples: os dados nesses blocos são atualizados em períodos referentes a ‘n’ ciclos do sinal de *clock*, ‘clk’ do módulo. Durante esse período o semáforo é mantido ligado, de forma a indicar que os dados nesses registradores estão disponíveis. Após ‘n’ ciclos, o estado do semáforo é alterado, de forma a sinalizar que os dados não estão disponíveis, e então as informações nos registradores são atualizadas. Durante este período, qualquer solicitação de leitura recebida pelo OPB será aceita, mas a resposta só poderá ser enviada quando o semáforo indicar a disponibilidade da informação. O

término da atualização dos dados nesses registradores é indicado pela mudança no semáforo.

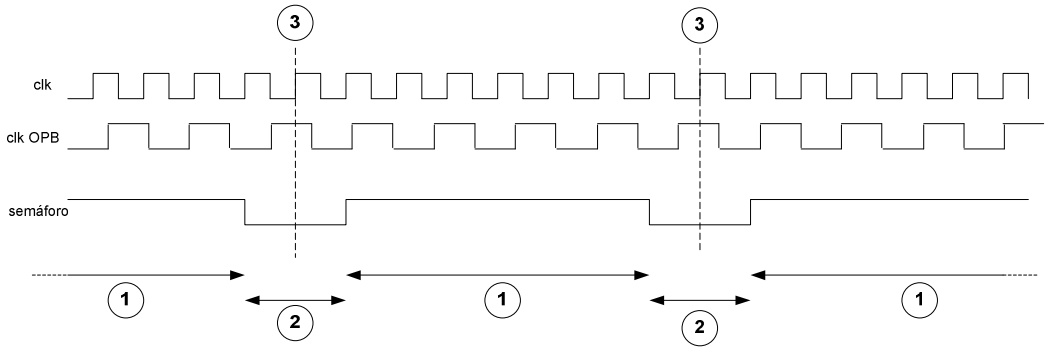

Figura 21 – Detalhe do funcionamento do semáforo proposto

Onde o intervalo:

1: representa o período em que a informação está disponível para leitura pelo dispositivo OPB;

2: representa o período em que a informação não está disponível. Se alguma requisição de leitura for recebida nesse intervalo, será necessário aguardar que o semáforo volte a indicar a disponibilidade dos dados;

3: representa o instante em que os registradores são atualizados com as informações enviadas pelos módulos de transmissão e recepção.

Uma conseqüência imediata deste mecanismo é a existência de um período mínimo necessário entre duas operações de leitura consecutivas. No entanto, devemos ressaltar que este tempo ainda é muito inferior ao requerido pelos demais processos envolvidos em uma operação de leitura ou escrita pelo barramento OPB e pela interface serial.

5.4.2. A gerência através de *software*

A segunda parte das operações de controle e gerência é realizada através de um programa que é constantemente executado pelo processador PowerPC. A este programa é designada a função de estabelecer a interface entre o módulo de

controle, em *hardware*, e o programa de administração, em execução em um microcomputador externo, conectado através de uma porta serial. O programa tem como funções principais receber comandos e solicitações de informações, além de traduzir dados brutos, obtidos do módulo de controle, ou alternar representações numéricas.

A principal vantagem no uso do programa é a simplicidade técnica que ele proporciona, se comparado ao trabalho necessário para a mesma realização em blocos de *hardware*. A comunicação serial, através de uma porta UART já existente, a leitura de comandos e a montagem de seqüências de caracteres como resposta, são exemplos dessas operações. Uma exploração mais aprofundada deste recurso possibilita ainda uma infinidade de tarefas de processamento de resultados de grande complexidade.

5.5. Programa de gerenciamento e interface com o usuário

No extremo oposto da solução, um programa sendo executado em um microcomputador externo é responsável por duas importantes funções:

1: permitir a interação do usuário com todo o equipamento, repassando comandos;
2: exibir os resultados obtidos ao longo do teste.

O programa que será mostrado a seguir consiste em uma interface básica de testes, onde o usuário deve definir os parâmetros desejados e pode visualizar os resultados obtidos. Foi desenvolvido na linguagem C#[20] com o uso do ambiente "Microsoft Visual Studio 2008 Express", na plataforma "Microsoft .NET"® versão 3.5, em modo gráfico ('*winform*') para ambientes "Microsoft Windows"®. A escolha desta plataforma se deu pela sua simplicidade ao longo do desenvolvimento deste projeto. Versões em outras plataformas, inclusive independentes do sistema operacional, podem ser desenvolvidas com grande facilidade.

A imagem abaixo mostra a tela principal do programa:

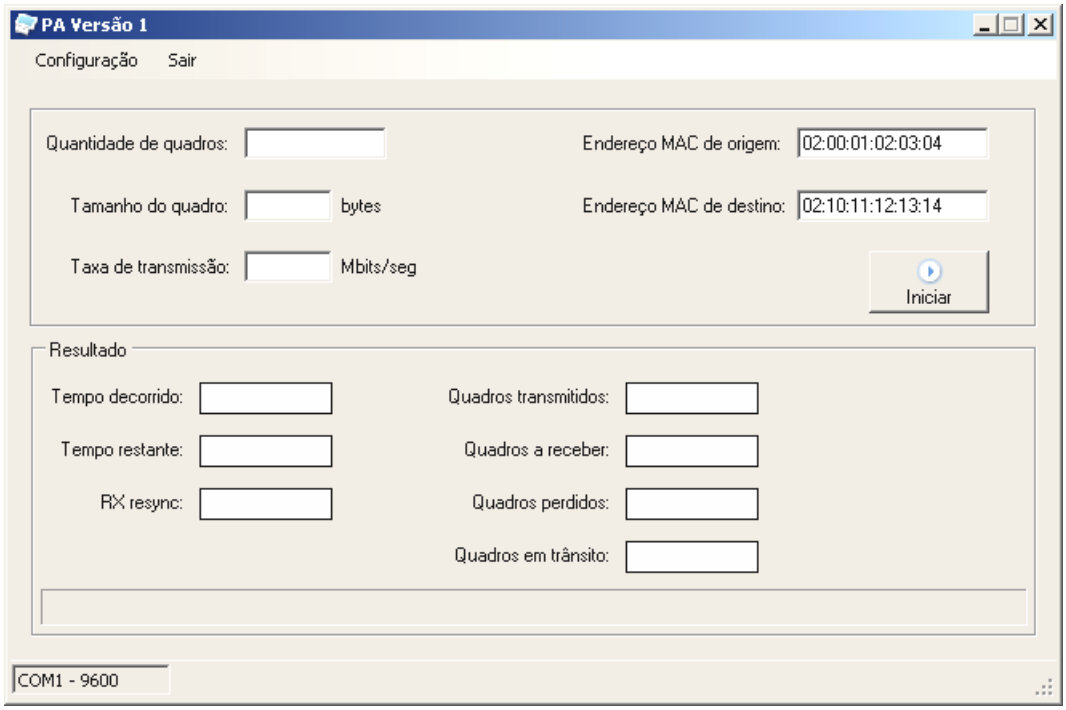

Figura 22 – Tela principal do programa de administração

Na parte superior da janela, o usuário deve preencher os parâmetros desejados para o teste:

- a quantidade de quadros que será transmitida ininterruptamente ao longo do teste;
  - mínimo: 1
  - máximo: 2^{32}
- o tamanho fixo dos quadros, em bytes;
  - mínimo: 64
  - máximo: 1518
- taxa de transmissão, em Mbps;
  - mínimo: 0,1
  - máximo: 1000

  Obs: São admitidas até 2 casas decimais
- endereços MAC de origem e destino.
  - Formatados em 6 octetos com representação hexadecimal e separados por ‘dois-pontos’.

Para iniciar o teste o usuário precisa apenas acionar o botão ‘Iniciar’.

Ao longo do teste, os resultados parciais são exibidos, como na imagem abaixo.

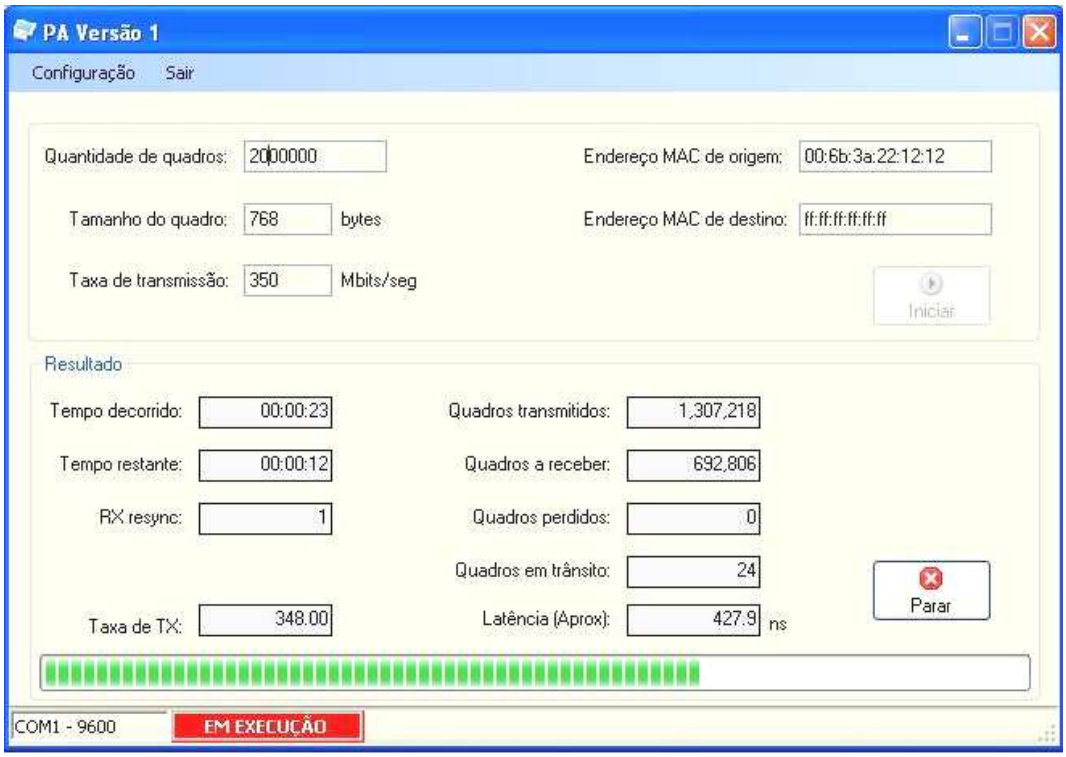

Figura 23 - Tela principal do programa de administração, exibindo resultados parciais, durante um teste

Duas telas adicionais de configuração fazem parte do programa. A primeira delas permite a configuração da comunicação pela porta serial do microcomputador:

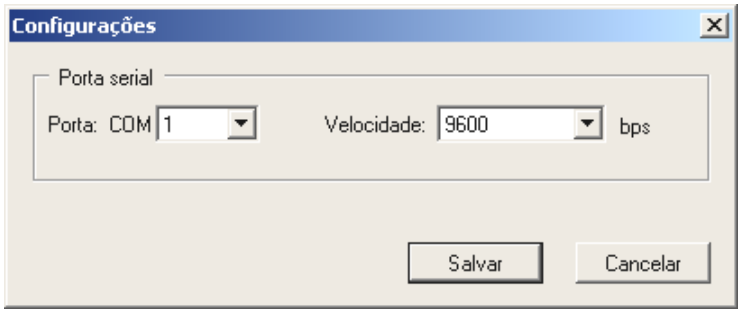

Figura 24 – Tela de configuração do acesso pela porta serial

A segunda oferece a opção de ajuste do período entre atualizações de resultados ao longo dos testes:

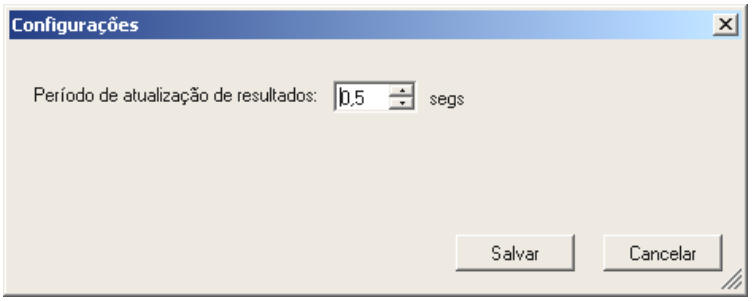

Figura 25 – Tela de configuração do período entre atualizações durante a realização de um teste

A configuração acima instrui o programa a emitir um comando para obtenção de resultados parciais do teste a cada meio segundo. Ao obter as respostas, as informações na tela são atualizadas.

5.6. Comentários e conclusões

Ao longo do presente capítulo foram apresentadas as principais etapas da realização do equipamento proposto. Inicialmente foram introduzidas as características da placa de desenvolvimento utilizada, destacando detalhes do processador Power-PC embarcado, do gerenciador de *clock* (DCM), do transceptor *Multi-Gigabit* (MGT) e do barramento PLB. Em seguida, os detalhes técnicos da solução adotada, incluindo os módulos de transmissão, recepção e controle, foram apresentados. Também foram detalhadas questões relativas ao cálculo do intervalo entre quadros e à avaliação da capacidade máxima de transmissão. Finalmente, foram apresentados os fluxogramas dos módulos desenvolvidos.