Projeto Mercúrio - Interface de comunicação de Alta Velocidade

Alexandre Ignacio Barboza[1] , Sérgio Takeo Kofuji[2]

Laboratório de Sistemas Integráveis, Escola Politécnica da Universidade de São Paulo
Av. Prof. Luciano Gualberto, 158 – Trav. 03
Cidade Universitária – São Paulo – SP
CEP 05508-900
(1 barboza@lsi.usp.br)
(2 kofuji@lsi.usp.br)

Resumo

Este trabalho apresenta o projeto e implementação de uma interface de comunicação de alta velocidade, destinadas para o uso em redes de alta velocidade do tipo SAN (System Area Network). Discutimos os aspectos de implementação como o processamento de pacotes e o controle de fluxo executados pela interface.

Keywords— **Redes de Alta Velocidade, Lógica programável, VHDL.**

1 Introdução

O projeto SPADE[7] vem sendo desenvolvido no LSI-EPUSP[3], há vários anos, pesquisando o processamento de alto desempenho usando o conceito de cluster. Um cluster pode ser entendido com um aglomerado de computadores que operam de uma forma integrada de modo a dar a impressão de que se trata de apenas um computador. O SPADE vai além ao pesquisar meios de tornar o projeto escalável.

O projeto Mercúrio faz parte de um sistema de interconexão de alta velocidade a ser usado no Projeto SPADE para prover serviços de interconexão entre nós, através de controle de fluxo e controle de erros.

Escalabilidade é a chave do problema, quando temos um sistema escalável queremos dizer que a capacidade do sistema pode ser aumentada adicionando nós de processamento ao sistema a medida em que isso for necessário.

O sistema de interconexão está no centro do problema da escalabilidade, pois, à medida que se acrescenta nós ao sistema aumenta-se a quantidade de informação que deve ser compartilhada e consequentemente o trafego. Assim se faz necessário um sistema que com o aumento na quantidade de nós seja proporcional ao aumento na capacidade transmissão de dados.

Tendo essas premissas em mente, analisamos diversas redes de interconexão comerciais existentes, estudamos topologias de redes, métodos de roteamento, protocolos de comunicação e roteamento, e interfaces de redes das mais diversas existentes. Em particular, estudamos com mais profundidade redes ATM[9] e Myrinet[1]. Depois deste estudo preliminar, vimos a possibilidade de adotarmos um modelo ou até mesmo um padrão, a partir do qual nos baseamos para chegar à concepção do projeto Mercúrio. O modelo[10] adotado foi a rede Myrinet, da Myricom através da especificação da própria Myricom. A Myrinet é derivada do projeto Atomic[2] que foi desenvolvido na USC/Information Sciences Institute[6]. Apesar do modelo escolhido ter sido o da rede Myrinet, decidimos usar a interface numa outra tecnologia de rede (ATM), em desenvolvimento dentro do projeto RECATS[8] (REConfigurable ATm Switch): Comutador ATM com hardware automaticamente reconfigurável. A utilização do padrão UTOPIA[9], permite que a interface projetada possa ser utilizada em diversos projetos.

O projeto RECATS está sendo desenvolvido pelo pesquisador Edson Lemos Horta, dentro da sua Tese de Doutorado, e trata da construção de um comutador ATM de alta velocidade reconfigurável usando o padrão UTOPIA para a conexão com o módulo de interface, o meio físico de transmissão.

O projeto pode ser dividido em duas partes : o comutador de rede e a placa de rede. Dentro do comutador de rede temos os módulos de interface física de rede e o circuito comutador propriamente dito (switching fabric), que nesta fase será o projeto RECATS, e dentro da placa de rede temos um módulo de interface física de rede e um módulo de controle que implementa alguns níveis da pilha de protocolos e realiza a interface entre o enlace de rede e o barramento de entrada-saída do computador. Atualmente, estamos em fase de validação da interface de rede para podermos começar a especificação e posterior implementação do módulo de controle.

Assim a interface de rede tem como atribuições:

- Envio e recepção dos dados através dos cabos da rede ;

•O controle de fluxo;

• Controle e verificação de erros.

No item 2 procuramos mostrar a metodologia usada para alcançar os resultados desejados. O item 2.1 descreve como foi o processo de escolha da plataforma que seria mais adequada para as finalidades do projeto. O item 2.2 discute como foi o processo de escolha e definição da arquitetura usada no projeto. No item 2.3 apresentamos brevemente o padrão UTOPIA. O item 2.4 mostra a tecnologia usada e os motivos para escolha. No item 3 damos um breve histórico de como especificamos a interface para em seguida no item 3.1 descrevermos como ela funciona. A descrição funcional da interface mostra como ela funciona em grande detalhe. No item 4 temos a conclusão. O item 5 possui as referências do trabalho.

2 Metodologia

2.1 Escolha da plataforma

Quando iniciamos o projeto tínhamos em mente que usaríamos como plataforma de hardware o padrão PC. Essa escolha foi calcada em diversas premissas:

♦ Disponibilidade: atualmente há inúmeros fornecedores de hardware para a plataforma PC no mercado de forma que este se tornou uma commodity.

♦ Preço: o preço das máquinas compatíveis com PC tem diminuído continuamente tornando viáveis projetos de computação paralela antes caros demais para este tipo de aplicação.

♦ Desempenho: a evolução dos processadores utilizados nos PCs tem sido continua a vários anos o que permite supor que esse movimento irá persistir por mais algum tempo.

Dentro da arquitetura do PC temos como interface de I/O o barramento PCI, que tem as seguintes possibilidades de throughput :

Largura da via	Clock	Taxa máxima de transmissão
32	33 MHz	1,1 Gbit/s
32	66 MHz	2,1 Gbit/s
64	33 MHz	2,1 Gbit/s
64	66 MHz	4,2 Gbit/s

Assim podemos ver que a taxa máxima possível é bem alta. A Myricom já tem disponível versões de seus dispositivos funcionando com 64 bits e 33MHz operando numa taxa de 1.12 Gbit/s

Entretanto nosso objetivo é conseguir taxas na ordem de 1 GByte/s, que demanda aproximadamente 10 Gbit/s .

Inicialmente usaremos como interface o barramento PCI pela possibilidade de se obter bons resultados num tempo relativamente curto.

Para atingir nossos objetivos deveremos estudar outras alternativas e tecnologias que ainda estão por vir.

2.2 Escolha da arquitetura

Ao estudar diversas possibilidades de arquitetura uma que chamou muito a atenção foi a da rede myrinet, por ser topo de linha entre as redes LAN e SAN comerciais existentes. Estudamos a estrutura de suas placas de rede e comutadores e decidimos fazer algo o mais semelhante possível. Iniciamos o projeto pela interface de rede.

Durante o desenvolvimento da interface de rede nos deparamos com a possibilidade de usarmos o que já tínhamos aprendido nas etapas anteriores para prover um meio físico para o projeto RECATS. Isso envolveu uma quase total reestruturação dos circuitos internos, mas a arquitetura se manteve quase intacta. A fig. 2.1 exemplifica a arquitetura básica de uma placa de rede conectada a um comutador, não está mostrado na figura mas o comutador está ligado a outras interfaces de rede que por sua vez podem estar ligados a outros comutadores ou placas de rede.

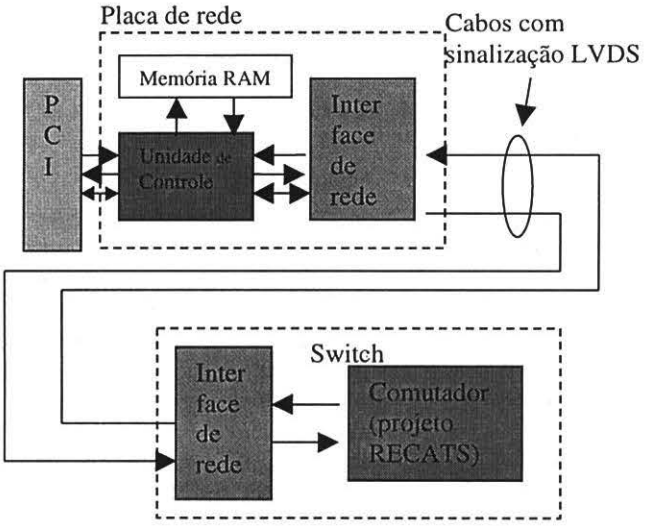

Fig. 2.1 Arquitetura típica

2.3 O padrão UTOPIA

O padrão UTOPIA nível 3 foi definido pelo ATM Forum Technical Committee, e foi focalizado para atender às necessidades de uma rede ATM e tem larguras de bandas nominais definidas abaixo:

Largura do barramento	Taxa nominal de transferencia
8	800 Mbit/s
16	1,6 Gbit/s
32	3,2 Gbit/s

Vê-se que temos taxas altas mas que ainda estão abaixo dos nossos objetivos.

Atualmente estamos buscando compatibilidade com o UTOPIA nível 3 com largura do barramento de 32 linhas, mas passada essa fase estaremos estudando formas de aumentar a largura de banda nominal. Uma alteração que desejamos implementar é o suporte para pacotes grandes serem transmitidos sem serem quebrados, para que tenhamos nos pacotes grandes a eficiência do padrão Myrinet e nos pacotes pequenos a eficiência de uma rede ATM.

2.4 Tecnologia

A interface de rede é um projeto de hardware relativamente complexo, sendo assim a sua construção foi feita com o auxilio da linguagem VHDL, que foi desenvolvida nos Estados Unidos da América como uma forma de tornar padronizada a descrição de circuitos digitais entre as diversas companhias produtoras de chips e fornecedoras do governo Norte Americano, e que acabou sendo adotado como padrão da indústria. Assim, se uma empresa que não fabrica chips mas os utiliza em seus produtos e precisa de um chip específico, pode projetá-lo dentro da própria empresa para depois mandar confeccioná-lo numa empresa especializada na produção de chips.

Dado o número de versões na fase de projeto optamos por utilizar a tecnologia das FPGAs, que são chips sem função lógica pré definida, mas uma quantidade muito grande de portas lógicas que podem ser interligadas para se obter o efeito desejado. Utilizamos durante muito tempo o software da ALTERA[5] para modelar e simular a arquitetura do projeto, antes de iniciar a adaptação ao padrão UTOPIA. No software da ALTERA conseguimos em simulação freqüências de 80 MHz com largura de 32 linhas de dados usando como modelo os chips da família FLEX 10KE da ALTERA. Isso nos fornece uma taxa bruta de 2,56 Gbit/s.

Atualmente estamos utilizando a tecnologia VIRTEX da XILINX[11] devido principalmente ao padrão UTOPIA nível 3 que impõe freqüência de 104 MHz.

Deste modo utilizamos a linguagem VHDL para gerar uma descrição da interface de rede com técnicas de projeto digital, usando na compilação e simulação o software MAX+PLUS II da ALTERA e FOUNDATION da XILINX.

Porém depois que o projeto for testado e comprovado o seu funcionamento, será necessário partir para chips de aplicação específica como um ASIC para obter as taxas necessárias aos nossos objetivos, usando cobre como meio físico.

3 Implementação

A primeira etapa do projeto, que é anterior à edição de qualquer linha em VHDL, consiste em fazer a especificação da interface que compreendia a definição dos pinos de entrada e saída dos blocos funcionais e modo de operação de cada módulo e seu relacionamento com os outros blocos. Para isso tomamos como modelo a interface de rede Myrinet da Myricom, com modificações para atender as nossas necessidades no projeto.

Entretanto o projeto original foi modificado para ser compatível com o padrão UTOPIA. Essa modificação nos fez rever quase todo o projeto de forma que na essência ele mantivesse sua forma de funcionamento mas que conseguisse conversar com um dispositivo UTOPIA.

3.1 Descrição Funcional

3.1.1. Interface Mercúrio x Mercúrio

Pacote:

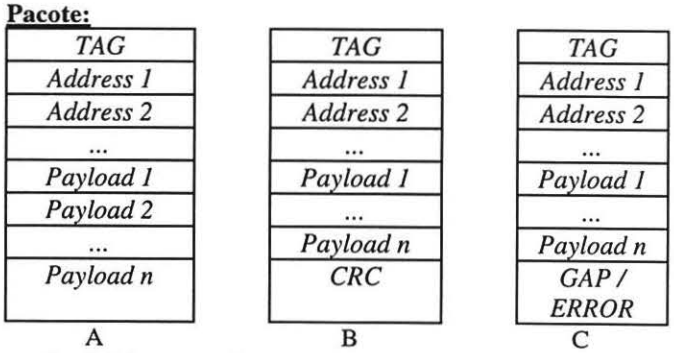

fig. 3.1 formatos de pacotes usados na transmissão

Onde os índices A, B, C correspondem à:

A: Pacote recebido pela interface do Host;
B: Pacote transmitido pela interface para outra interface;
C: Pacote recebido pelo Host.

Caracteres

Há dois tipos de caracter definidos:

Dado;

Endereçamento/Controle.

Controle de Fluxo

Há um bit (RxFLOW_CONTROL) utilizado para controlar a transmissão e recepção de dados, utilizando a idéia da "caixa d'água" exemplificado na fig. 3.2. A idéia é parar a transmissão num nível seguro para evitar perda de dados, e só iniciar depois de haver folga suficiente para a transmissão seguir o mais continua possível.

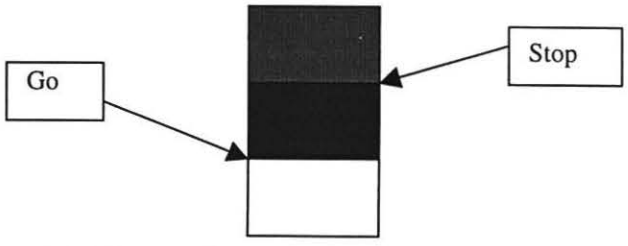

fig. 3.2 gerenciamento dos níveis de memória

Descrição

O projeto pode ser dividido em três blocos descritos abaixo : OUTPUT CONTROL, INPUT CONTROL e BUFFER CONTROL.

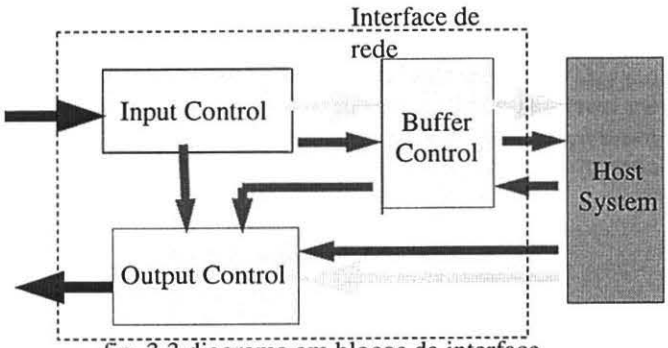

fig. 3.3 diagrama em blocos da interface

OUTPUT CONTROL

Montagem do Pacote

O HOST SYSTEM fica encarregado do roteamento dos pacotes, assim quando um pacote chega à interface o seu destino é a próxima interface e já está quase pronto. Ao OUTPUT CONTROL cabe a tarefa de calcular e inserir o CRC ao final do pacote. Além disso o OUTPUT CONTROL identifica todos os tipos de frames possíveis e que foram listados acima. Dessa forma, haverá dois tipos de frames: aqueles formados pelos dados gerados pelo processador (tais como endereços e dados propriamente ditos) e aqueles contendo o CRC do pacote (gerado pelo OUTPUT CONTROL).

Transmissão

Os dados são disponibilizados pelo processador seguindo o protocolo UTOPIA. A cada caracter, é efetuado também o cálculo do CRC. O último frame do pacote é indicado pelo número de caracteres já transmitidos, em seguida é inserido o caracter com o CRC calculado. O TxClav que indicará se há espaço para que o dispositivo UTOPIA possa escrever na interface. Na verdade como o OUTPUT CONTROL não possui memória, esse controle de fluxo refletirá o estado da memória no BUFFER CONTROL na outra interface a menos de uma margem de segurança.

INPUT CONTROL

Fluxo de Dados

Internamente, há três registradores pelos quais a informação passa. Inicialmente, os dados são armazenados no primeiro registrador, com o clock sensível a nível baixo, além de ser efetuado o cálculo do CRC na borda de subida. Na borda de descida, esses dados, armazenados no primeiro registrador, são repassados ao segundo registrador. Na borda de subida seguinte, são repassados ao terceiro. É neste momento, ou seja, um ciclo após a chegada do último frame de dados, que é realizada a montagem final do pacote a ser transmitido ao BUFFER, através da escrita do byte de Gap ou de erro. Na borda de descida seguinte os dados são colocados na saída.

Verificação de Erros

Na borda de subida em que os dados são capturados pela primeira vez, é iniciado também o cálculo do CRC, cujo resultado é posteriormente comparado àquele transmitido. Além disso, é feita a verificação do tipo de caracter, pois no caso de pacotes sem tamanho definido os Switches podem receber somente pacotes iniciados por endereços, enquanto que hosts devem receber somente os dados, sem endereço.

BUFFER CONTROL

O bloco BUFFER CONTROL é responsável pela armazenagem temporária dos dados, pelo controle de fluxo e pela sincronização com o clock do processador Host.

Internamente ele possui dois contadores de módulo igual ao do número de caracters possiveis de serem armazenados. O contador que posiciona os dados na entrada é acionado após cada escrita para prepará-lo para a próxima escrita. Na leitura da memória, o modo de transferência de dados obedece ao padrão UTOPIA. Assim, uma vez iniciada a transmissão, ela somente pode ser encerrada ao termino de uma célula mínima. Como temos vários tamanhos de pacote, convencionamos que eles devem ser múltiplos entre si.

Isso se explica pelo funcionamento da transmissão: a transmissão só se inicia depois que a diferença entre o contador de entrada no buffer e o contador de saida atingir um nivel mínimo, correspondente ao tamanho do menor pacote definido, e só vai parar ao final deste número mínimo se : a) não tiver permissão para escrever outra célula, b) se a diferença entre os contadores ficar abaixo do tamanho do menor pacote definido.

Toda a interface entre o Mercúrio e o elemento UTOPIA será síncrono com o relógio (clock) do barramento UTOPIA. Isso é possível graças ao artifício da memória DUAL PORT, que permite a escrita e leitura assíncronas entre si e ao mesmo tempo desde que em endereços diferentes.

Atualmente as memórias usadas são modeladas por funções proprietárias da XILINX e ALTERA com o objetivo de usar a memória já construida na FPGA e que não seria usada utilizando outro método para modelá-la.

3.1.2 Interface Mercurio x Utopia

Controle de Fluxo

O controle de Fluxo é baseado nos sinais RxClav e TxClav. Como os pacotes tem tamanhos diferentes estes sinais não significam mais que há espaço para uma célula e sim que há espaço para uma célula mínima, ou seja, para um pedaço de pacote com o tamanho correspondente ao do menor pacote definido. Assim, se a interface tem espaço para uma célula mínima, ela ativa o sinal TxClav e o deixa ativo até que não possa mais receber células, mas garante que a célula que está sendo transmitida pode ser terminada. Se acabada a transmissão da célula, ainda não houver espaço, ela mantém o sinal desativado até que possa voltar a transmitir. No caso da recepção, ela indica que há uma célula inteira a ser transmitida enquanto o RxClav estiver ativo. Caso ele fique desativado, a célula que estiver sendo transmitida terminará. O RxClav só voltará a ficar ativo quando uma nova célula inteira chegar.

Fim do pacote

O final de pacote será sinalizado segundo o protocolo UTOPIA, onde a indicação é o número de caracteres.

4 Conclusão

A interface esta sendo finalizada e seus últimos detalhes sendo acertados tais como uma versão com múltiplas interfaces no mesmo chip para tornar a construção de um comutador de até quatro portas mais simples e confiável.

Até agora as simulações teêm mostrado que a freqüência de 104 MHz está no estremo de funcionamento de um circuito com esta complexidade dentro de uma FPGA. Talvez tenhamos que diminuir a freqüência para podermos comprovar a eficiência do circuito para uma posterior implementação num ASIC (Aplication Specific Integrated Circuit, Circuito Integrado de Aplicação Específica), que possibilitaria atingirmos nossos objetivos.

A construção desta interface é o primeiro passo na construção de uma rede de alta velocidade. Somando este projeto a outros esforços teremos em breve um sistema de interconexão de alta velocidade que será usado no projeto SPADE.

5 Referências

[1]Myrinet e Myricom - www.myri.com;

[2]Robert Felderman, Annette DeSchon, Danny Cohen, Gregory Finn - Journal of High Speed Networks, Vol. 3, No. 1, 1994, pp. 1-30 Atomic: A High Speed Local Communication Architecture - www.myri.com/research/publications/index.html;

[3]Laboratório de Sistemas Integráveis – www.lsi.usp.br;

[4]J. BHASKER - A VHDL PRIMER - Prentice Hall PTR, Englewood Cliffs, New Jersey;

[5]Altera corporation home page - www.altera.com;

[6]USC/Information Sciences Institute - www.isi.edu;

[7]Projeto SPADE II - www.lsi.usp.br/hpcac/spade/spade.htm;

[8] Página do projeto RECATS: www.lsi.usp.br/~recats/

[9] The ATM Forum Technical Committee - UTOPIA 3 Physical Layer Interface - af-phy-0136.000 / November, 1999

[10]Myrinet-on-VME Protocol Specification Draft Standart - VITA 26-199X Draft 1.1 - 31 de agosto de 1998;

[11] XILINX corporation home page - www.xilinx.com