

Ligação de Alto Desempenho entre FPGAs para Switch Ethernet FTT*

Luís Silva[†], Arnaldo S. R. Oliveira[‡], Paulo Pedreiras[‡], Rui Santos[‡]

[†]DETI, [‡]DETI/IT, [‡]DETI/EETA

Universidade de Aveiro, Portugal

{a35296, arnaldo.oliveira, pbrp, rsantos}@ua.pt

Resumo

A Ethernet como definida na norma IEEE 802.3 não é adequada a aplicações tempo-real. Apesar de terem surgido vários protocolos que tornam possível a sua aplicação em sistemas de tempo-real, as garantias oferecidas são em geral estáticas e pouco flexíveis. O projecto HaRTES tem como objectivo o desenvolvimento de um switch Fast Ethernet com recurso a tecnologia FPGA, capaz de fornecer serviços de comunicação tempo-real com garantias dinâmicas de qualidade de serviço. No entanto, os recursos oferecidos pelas FPGAs e placas de desenvolvimento actuais são limitados para este tipo de projecto, dificultando a escalabilidade do switch em diversos aspectos (e.g. número de portos). O presente artigo discute o desenvolvimento de uma ligação série multi-gigabit entre FPGAs que constituem o switch, com vista a minorar as limitações de escalabilidade no projecto do switch HaRTES.

1. Introdução

O uso de redes de campo especializadas (comumente designadas por *fieldbuses*), capazes de suportar tráfego com requisitos de tempo-real foi durante muito tempo a norma em comunicações entre processos industriais e sistemas embutidos distribuídos. Contudo assistiu-se nos últimos anos a uma enorme evolução da tecnologia Ethernet e à sua massificação nas redes de dados de uso geral, pelo que este tipo de redes apresenta menores custos de equipamento e instalação, maior desempenho médio e melhor suporte quando comparado com os *fieldbuses* convencionais. No entanto, a Ethernet possui ainda factores limitativos para um desempenho temporal determinístico, o qual é um aspecto fundamental em sistemas de tempo-real. Apesar da introdução de redes microsegmentadas com recursos a *switches* ter reduzido o indeterminismo causado pelo protocolo CSMA/CD, os *switches* tradicionais possuem alguns aspectos funcionais e de implementação que afectam negativamente as suas capacidades de comunicação em aplicações de tempo-real. A inexistência de mecanismos de controlo de admissão e de alteração *online* dos requisitos de comunicação, a possibilidade de *overflow* nas filas de pacotes nos portos de saída e o reduzido número de níveis

de prioridade das mensagens são alguns dos factores que dificultam a obtenção de um bom desempenho temporal nos *switches* tradicionais. Para tentar solucionar estes problemas surgiram vários protocolos (e.g. EtherCAT [1], ETHERNET PowerLink[2]) que tentam aproveitar algumas propriedades da tecnologia Ethernet enquanto a tornam útil para aplicações de tempo-real. Contudo, as garantias obtidas por estes protocolos são maioritariamente estáticas e obtidas da especificação funcional e temporal dos sistemas na fase de projecto, e não fornecem mecanismos de gestão de qualidade de serviço que possibilitem alterar dinamicamente o funcionamento do sistema.

Em 2009 surgiu na Universidade de Aveiro o projecto HaRTES (*Hard Real-Time Ethernet Switching*) [3] com o objectivo de desenvolver uma infra-estrutura de comutação para *Fast Ethernet* (*switch*) baseado na tecnologia FPGA e no paradigma FTT-SE [4], capaz de oferecer flexibilidade operacional e gestão de recursos eficiente, cumprindo ao mesmo tempo as garantias exigidas pelas aplicações de tempo-real. O recurso a FPGAs resulta da existência de várias funcionalidades do *switch* que requerem uma elevada previsibilidade, determinismo e velocidade na sua execução, sendo portanto preferível a sua execução em *hardware*. Apesar da sua evolução, a quantidade de recursos disponíveis nas FPGAs e placas de desenvolvimento actuais para este tipo de projectos é ainda limitada ou dispendiosa, dificultando a sua escalabilidade. Devido às suas características e custo a plataforma de *hardware* usada no projeto HaRTES para desenvolvimento do *switch* é a NetFPGA [5], a qual possui uma FPGA *Virtex-II* e um número reduzido de portos Ethernet (apenas 4). Uma possível solução para esta limitação consiste na utilização de várias FPGAs interligadas por canais série de alta velocidade.

Neste artigo será discutido o desenvolvimento de uma ligação ponto-a-ponto entre FPGAs, que permita a integração de duas plataformas NetFPGA como um único componente lógico de forma a escalar o número de portos do *switch*. Esta ligação deverá possuir elevado determinismo e valores reduzidos de latência e *jitter* de modo a não degradar a *performance* temporal do *switch*.

2. A Plataforma NetFPGA

A NetFPGA [5] é uma plataforma de baixo custo desenvolvida pela Universidade de Stanford e disponibilizada em *open-source* para a comunidade. Foi desenvolvida

*Com apoio da Fundação para a Ciência e Tecnologia através do projecto HaRTES - Hard Real-Time Ethernet Switching - PTDC/EEA-ACR/73307/2006

como uma ferramenta para o ensino de *hardware* de redes e desenvolvimento de *routers*. No entanto, ao longo do tempo também se mostrou útil para uso na investigação e desenvolvimento de tecnologias de redes. A NetFPGA inclui todos os recursos lógicos, memória e interfaces Gigabit Ethernet necessários para a construção de um *switch* ou *router*. Todo o seu caminho de dados é inteiramente desenvolvido em *hardware*, tornando o sistema capaz de suportar o processamento de pacotes das ligações Gigabit com baixa latência. A Figura 1 ilustra os seus principais blocos.

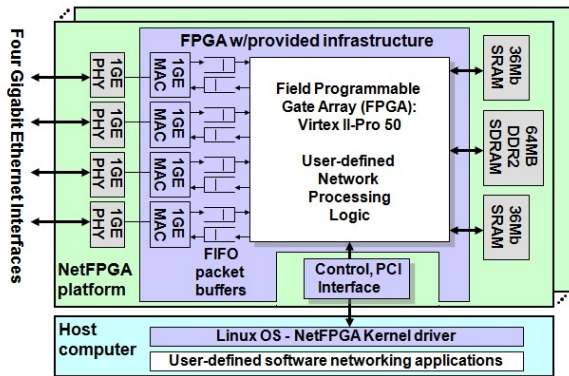


Figura 1. Diagrama de blocos NetFPGA

3. Ligação Série Multi-Gigabit

No âmbito deste projecto pretende-se desenvolver uma ligação ponto-a-ponto de alto desempenho entre *FPGAs* de duas placas NetFPGA de forma a se escalar o *switch* HaRTES e obter-se um único *switch* com um número de portos Fast Ethernet superior, como ilustrado na Figura 2. Futuramente poderão ser implementadas topologias com recurso a ligações deste tipo de modo a agregar um número superior de placas NetFPGA. Devido às características das *FPGAs* e às interfaces das placas de desenvolvimento usadas a conectividade entre elas será baseada em ligações série *multi-gigabit*.

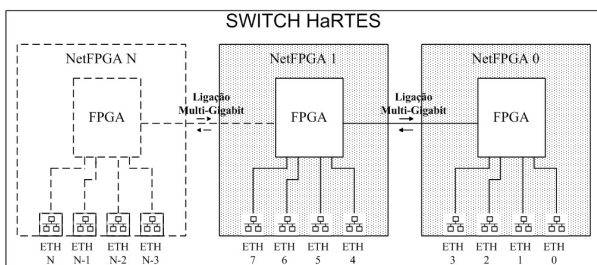


Figura 2. Ligação *multi-gigabit* para o *switch* HaRTES

Na arquitectura actual, as tramas Ethernet são recebidas em octetos de bits nos portos Ethernet do *switch*, aos quais é acrescentado um bit de controlo formando células de 9 bits que são depois armazenadas em memória. Após

a recepção do último octeto da trama Ethernet, é gerada uma palavra de controlo de 13 bits. Depois da análise e determinação do destino da trama Ethernet, são retirados os bits de controlo de cada célula e enviados os octetos originais pelo porto Ethernet destino. Quando os portos de origem e de destino de uma trama Ethernet pertencem à mesma *FPGA*, o encaminhamento é directo não utilizando a ligação *multi-gigabit* entre *FPGAs*. Por outro lado, uma *FPGA* que receba uma trama Ethernet num dos seus portos, a qual têm como destino pelo menos um porto pertencente a outra *FPGA*, deverá ser capaz de utilizar a ligação *multi-gigabit* para transmitir as células/palavra de controlo para essa *FPGA*. Estas células e palavras de controlo serão encapsuladas em pacotes, os quais serão enviados pela ligação *multi-gigabit*. Para ser possível transmitir de forma determinística informação de vários portos utilizando a mesma ligação, o envio dos pacotes será multiplexado no tempo através de um mecanismo básico de *TDMA*¹, como ilustrado na Figura 3.

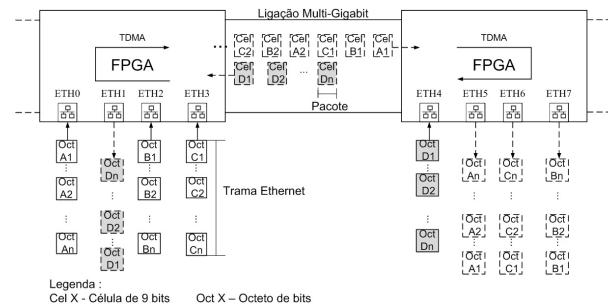


Figura 3. Multiplexagem na transferência dos pacotes

A ligação deverá ter em consideração as características protocolares e estruturais já desenvolvidas do *switch*, devendo cumprir os seguintes requisitos:

- Velocidade de transmissão de 400 Mbps (de forma a servir 4 portos);
- Comunicação *Full-Duplex* para permitir o envio/recepção de pacotes em simultâneo;
- Tamanho dos pacotes enviados 13 bits.

4. Implementação

As *FPGAs* possuem em geral estruturas de I/O que permitem facilmente criar ligações com velocidades elevadas utilizando um número reduzido de recursos. Estes dispositivos denominados na *Virtex-II Rocket I/O Multi-Gigabit Transceivers (MGT)* [6] permitem a transferência de bits em série a alta velocidade e possuem recursos de *hardware* dedicados para a serialização/deserialização dos dados. Para facilitar o seu uso em ligações entre dispositivos, a Xilinx disponibiliza nas ferramentas de desenvolvimento um protocolo de camada de ligação, denominado Aurora [7], que utiliza os *MGTs* para a camada física. Recorrendo apenas a um *MGT* e ao protocolo Aurora, foi possível a implementação da ligação *multi-gigabit*, com as propriedades descritas na Tabela 1.

¹Time Division Multiple Access

Parâmetro	Valor Configurado
Tamanho dos Pacotes	2 bytes
Comunicação	Full-Duplex
Velocidade de Transmissão	1.25 Gbps
Formato de Envio	Streaming

Tabela 1. Propriedades da ligação

Para interface entre o switch e a ligação multi-gigabit foi implementada a arquitectura ilustrada na Figura 4. Para não sobrecarregar a figura é apenas mostrada a arquitectura associada a um porto de uma FPGA, no entanto o sistema é perfeitamente simétrico, i.e. uma estrutura idêntica foi também implementada para os restantes portos bem como na segunda FPGA.

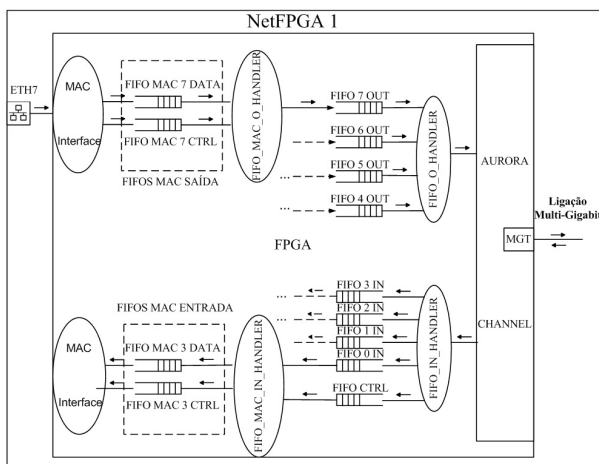


Figura 4. Arquitectura implementada

O switch armazena nos FIFOs MAC de saída as células e palavras de controlo das tramas para posterior reencaminhamento. Estas são depois encapsuladas em pacotes pelo módulo FIFO_MAC_O_HANDLER, e estes armazenados nos FIFOs OUT. O FIFO_O_HANDLER lê os pacotes armazenados, executa a sua multiplexagem e disponibiliza-os ao controlador do protocolo Aurora para serem transmitidos através da ligação multi-gigabit. Um pacote de controlo é enviado após o envio de 4 pacotes, e identifica os pacotes com informação válida (Figura 5). Na recepção de pacotes através da ligação, o módulo FIFO_IN_HANDLER separa e armazena os pacotes recebidos nos FIFOs IN, com base no porto de origem. Estes são depois processados pelo módulo FIFO_MAC_IN_HANDLER, que restaura as células e palavras originais, armazenando-as de novo nos FIFOs MAC de entrada. O switch irá depois retirar as células destes e encaminhá-las para o porto Ethernet de destino.

Devido ao tamanho dos pacotes, foi tomada a decisão de encapsular 2 células ou 1 palavra de controlo num pacote, em que o bit de controlo das células é retirado (Figura 6). No entanto, é enviada uma codificação na palavra de controlo da ligação que permite identificar o conteúdo de cada pacote enviado bem como os bits de controlo retirados. Assim o módulo FIFO_MAC_IN_HANDLER com base nesta codificação é capaz de restaurar correctamente

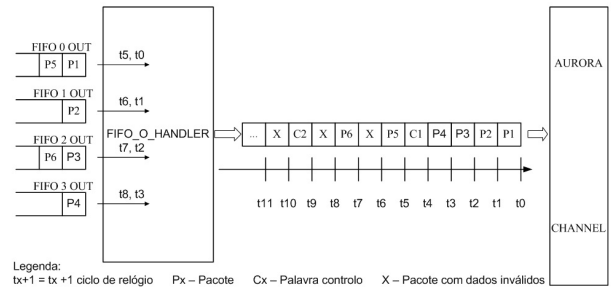


Figura 5. Diagrama de blocos FIFO_O_HANDLER

as células e palavras de controlo.

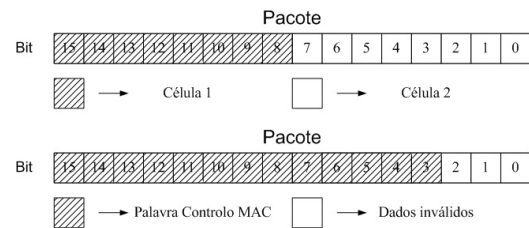


Figura 6. Pacotes enviados através da ligação

5. Recursos Utilizados

O valor dos recursos necessários para implementação da ligação multi-gigabit e da arquitectura apresentada na secção anterior em cada FPGA encontram-se quantificados na Tabela 2. Apesar da utilização de um número razoável de blocos RAM proveniente do uso intensivo de FIFOs, os recursos lógicos necessários são reduzidos permitindo a implementação desta arquitectura em projectos sem comprometer o número de recursos disponíveis.

Recursos	Utilização (%)
Flip Flops	5
Look-Up Tables	3
Occupied Slices	8
Input/Output Blocks	4
RAM Blocks	38
BUFGMUXs	18
MGTs	6

Tabela 2. Recursos totais utilizados em cada FPGA

6. Resultados

Para avaliar o desempenho da ligação, foram medidos os valores de latência e jitter entre o envio e a recepção das células. A Tabela 3 apresenta os valores obtidos da amostragem de 500 000 células por cada porto.

Após a implementação da ligação multi-gigabit no switch, foram medidos os valores da latência e jitter na transmissão de tramas entre portos ethernet do switch pertencentes à mesma FPGA (NetFPGA1/NetFPGA1) e a FPGAs distintas (NetFPGA1/NetFPGA0), com o objectivo

Porto	Latência (ns)			Jitter (ns)	
	Máxima	Mínima	Média	Máximo	Médio
1	600.00	544.00	570.59	56.00	17.87
2	600.00	544.00	570.51	56.00	17.91
3	600.00	544.00	570.58	56.00	17.90
4	600.00	544.00	569.57	56.00	17.87

Tabela 3. Latência e jitter da ligação *multi-gigabit*

de avaliar o impacto da ligação no desempenho do *switch*. Estes valores foram obtidos para tramas de 1512 bytes e estão apresentados na Tabela 4.

Origem/ Destino	Switching Time (μ s)			Jitter (μ s)	
	Máximo	Mínimo	Médio	Máximo	Médio
NetFPGA1/ NetFPGA1	121.96	121.64	121.89	0.32	0.03
NetFPGA1/ NetFPGA0	123.94	122.88	122.98	1.06	0.04

Tabela 4. Valores do tempo de *switching* e *jitter* do *switch*

Tendo em consideração que o *switch* opera em modo *store and forward*, o limite teórico inferior da latência introduzida é igual ao tempo de transmissão de uma trama. No caso vertente foram utilizadas tramas com um tamanho total de 1512 bytes (*overheads* incluídos) e uma velocidade de transmissão de 100Mbps, correspondendo a uma latência mínima de 120.96 μ s. Assim, os valores obtidos para ambas as situações (intra e inter-NetFPGA) representam um agravamento de aproximadamente 0.7% e 1.7% relativamente ao valor mínimo teórico. Trata-se de valores bastante baixos e da ordem de grandeza dos exibidos pelos *switches* comerciais. O aumento no tempo de *switching* e *jitter* na transmissão inter-NetFPGA, na qual a ligação *multi-gigabit* é utilizada, era esperado e deve-se à quantidade superior de lógica pela qual as tramas são processadas. Este aumento é, em termos relativos, reduzido graças à baixa latência e grande determinismo (570ns e 17 ns respectivamente) exibidos pela ligação *multi-gigabit*.

No que concerne especificamente ao protocolo FTT-SE, o aumento dos valores de latência e *jitter* acima referidos repercutem-se directamente na transmissão das tramas de dados entre os diversos nós, bem como nos *overheads* do protocolo, nomeadamente no tempo de resposta à mensagem de *trigger* do *master* (tempo de *turn-around*). Tendo em consideração que o protocolo FTT-SE é tipicamente configurado com tempos de ciclo de 1ms ou superiores e que o tempo de *turn-around* é da ordem de 200 μ s, conclui-se que a penalização máxima de 2 μ s é, para todos os efeitos práticos, desprezável.

7. Conclusões e Trabalho Futuro

O desenvolvimento e implementação da ligação entre *FPGAs* possibilitou o aumento do número de portos *Fast*

Ethernet do *switch* de 4 para 8. Os resultados obtidos mostram uma gestão homogénea da informação de todos os portos e níveis reduzidos de latência e *jitter*.

O uso da ligação *multi-gigabit* na transmissão de tramas entre as *FPGA* causa um aumento do valor do tempo médio de *switching* devido à maior quantidade de lógica pela qual as tramas têm de ser processadas. Contudo, devido ao valor médio de latência e *jitter* da ligação *multi-gigabit* ser reduzido (aproximadamente 570 e 17 ns respectivamente), o efeito no desempenho global é mínimo.

Os valores obtidos são satisfatórios, contudo acreditamos que recorrendo a tecnologias mais recentes (e.g *MGTs* mais rápidos e lógica capaz de operar a velocidades superiores disponível em *FPGAs* mais recentes) seria possível reduzir estes valores. Este aspecto juntamente com a implementação e avaliação desta abordagem para um número de portos ainda maior, constituem duas importantes linhas de trabalho futuro. Dependendo da topologia seguida para interligar um número superior de placas NetFPGA, os requisitos da ligação aqui desenvolvida poderão ser ou não alterados.

Referências

- [1] HMS Industrial Networks. Ethercat industrial ethernet protocol. <http://www.anybus.com/technologies/ethercat.shtml>, Aug 2010.
- [2] Ethernet POWERLINK. Why industrial ethernet. <http://www.ethernet-powerlink.org/index.php?id=15>, Aug 2009.
- [3] HaRTES. Hard real-time ethernet switching. <http://www.ieeta.pt/lse/hartes/project.html>, Apr 2009.
- [4] Almeida L. Pedreiras P. Marau, R. Enhancing real-time communication over COTS ethernet switches. *6th IEEE International Workshop on Factory Communication Systems*, 2006.
- [5] NetFPGA.ORG. Netfpga specifications. <http://www.netfpga.org/php/specs.php>, Sep 2009.
- [6] XILINX. Rocket I/O transceiver user guide. pages 22–143, Feb 2007.
- [7] XILINX. LogiCORE IP aurora 8b/10b user guide. pages 22–93, Apr 2009.