

Avaliação de um Chip Multiprocessado com Redes em Chip Implementado em FPGA

Marco A. Z. Alves, Philippe O. A. Navaux , Fernanda L. Kastensmidt
Grupo de Processamento Paralelo e Distribuído
Microeletronics Group
Instituto de Informática
Universidade Federal do Rio Grande do Sul
{marco.zanata, navaux, fglima}@inf.ufrgs.br

Resumo

Multiprocessors systems are been consolidated as future for systems-on-chip technologies. However, the interconnection among these cores must have some desirable good characteristics as speed, scalability and reusability. For these interconnections intentions, the network-on-chip represent nowadays an attractive alternative with great potential. With the increase on the number of projects using networks on chip and multiprocessors, the importance of evaluations on alternative architectures has a high importance, being an important way to supply informations to help in projects strategies, on systems-on-chip technologies. In this work, it was evaluated some aspects of an architecture implemented on FPGA, based on 4 mMips processors and a SoCIN network-on-chip, used to interconnect all the cores. The results showed informations about area and comparisons with others projects with same similarity.

1. Introdução

Nas ultimas décadas [10] o elevado grau de integração proporcionado por novas tecnologias tem dado força para os modelos de aumento de desempenho de sistemas computacionais baseados principalmente na extração de paralelismo no nível de instruções, como por exemplo, a utilização de agressivas predições de desvio, execução fora de ordem, *pipelines* cada vez mais profundos e também a elevação das frequências de trabalho. Porém, pode-se notar uma mudança dessa filosofia, fazendo os projetistas se preocuparem cada vez mais com o ganho de desempenho aumentando o paralelismo no nível de *threads*.

Nos dias atuais, essa mudança em buscar maior nível de paralelismo em nível de *thread* tem sido notada principalmente com o aumento da concepção de processadores mul-

tiprocessados [1] [6] [13], sendo que mesmo tendo mais que um núcleo de processamento, estes tendem a ocupar a mesma área necessária para um *chip* monoprocessado [10]. Isto se deve, a utilização de núcleos de processamentos mais simples, que visam mais do que o aumento do desempenho computacional. A utilização de núcleos de processamento mais simples apresentam outros atrativos, como por exemplo, diminuir a complexidade dos blocos de controle, a fim de reduzir a potência total do sistema e esconder as altas latências ocasionadas principalmente pelo tempo de acesso à memória, além de ser uma abordagem apoiada pela evolução da programação paralela.

Para integrar vários núcleos de processamento dentro de um único *chip*, faz-se necessário à utilização de alguma técnica de comunicação entre núcleos. Tal comunicação pode ser através de uma memória compartilhada, um barramento, uma chave *crossbar* ou uma rede em *chip*. As redes em *chip* [2], tem sido a grande aposta do mercado para o futuro de processadores multiprocessados, entre os motivos que levam a crer nessa tendência podemos apontar o alto desempenho que pode-se obter por essa ser uma abordagem que contém um alto grau de paralelismo intrínseco, podendo assim garantir uma alta disponibilidade e largura de banda nesse tipo de interconexão, além do que para processadores com um elevado número de núcleos, a área ocupada por uma chave *crossbar* tornaria o projeto inviável.

Tendo claro estas importantes questões sobre o projeto de um *chip* multiprocessado podemos notar a importância em fazer uma avaliação sobre a utilização de um *chip* multiprocessado com redes em *chip* para ser implementado em FPGA (*Field Programmable Gate Array*). Ressaltando que ao tratar de projetos em FPGA os recursos totais utilizados pelo projeto e o consumo total de potência juntamente como o desempenho do sistema são peças chaves nesse tipo de implementação.

Assim, o objetivo desse trabalho é avaliar uma arquitetura de fácil manipulação sendo, portanto de grande poten-

cial para uso acadêmico. Onde temos um sistema composto por um *chip* multiprocessado implementado com quatro processadores mMips(mini-mips) [5] interligado por uma rede interna SoCIN (*SoC Interconnection Network*) [15].

Na Seção 2 serão apresentados alguns trabalhos correlatos, a Seção 3 explica a arquitetura utilizada, os resultados e análise dos resultados estão dispostos na Seção 4 e o encerramento do artigo com algumas conclusões será feito na Seção 5.

2. Trabalhos correlatos

Em [4], é apresentada uma avaliação de um sistema multiprocessado homogêneo com comunicação baseada em NoC (*Network-on-Chip*) implementado em FPGA. Neste artigo, o autor compara duas propostas de design para arquiteturas multiprocessadas, a primeira utilizando uma rede em *chip* e a segunda proposta utilizando barramentos e memória compartilhada. No caso foram utilizados quatro *soft-cores* MicroBlazes e na proposta com rede em *chip* foi utilizada a SoCIN 2x2. Com relação aos resultados obtidos, mostra que uma eficiente implementação de NoC em FPGA pode aumentar a velocidade da comunicação em até sete vezes com pouco *overhead* de área, dependendo do número de núcleos conectados.

A proposta de interconectar quatro mMips com uma NoC é apresentada no projeto MiniNoC [7], onde são utilizados quatro processadores mMips interconectados com a mNoC, a NoC qual contém os roteadores. Neste projeto utiliza-se uma interface com a rede, chamada NI (*Network Interface*) para interligar cada mMips com a mNoC. Este projeto implementado em linguagem de descrição de sistemas é utilizado como estudo de caso para validação de ferramentas como ambientes de simulação e modelagem em nível de sistema. Um ponto interessante desse projeto em no uso da NI em conjunto com o bloco chamado MEMDEV, sendo que o bloco MEMDEV fica entre o processador e a memória, assim, ao acessar o endereço específico da NoC o bloco MEMDEV manda as informações para a interface de rede.

No trabalho [14], é apresentado um estudo sobre aproveitamento de espaço de projeto, utilizando um SoC (*System-on-Chip*) e a NoC Hermes com topologia Mesh-2D interligando os núcleos. Os resultados mostrados neste artigo apontam ótimos desempenhos utilizando NoC comparando com a utilização de barramentos, além de concluir que utilizando NoC a complexidade do projeto tende a ser menor que quando se utiliza barramentos de interconexão para interligar um grande número de núcleos.

3. Arquitetura utilizada

No ponto de vista de implementação de sistemas digitais [3], pode-se encontrar uma vasta gama de alternativas para serem utilizadas, porém, para sistemas dedicados e de rápida prototipação, os *Gate Array* e os *Field Programmable Gate Arrays* (FPGAs) são as mais atraentes opções. Portanto a escolha de FPGAs para a implementação realizada neste projeto.

A arquitetura multiprocessada implementada é formada basicamente por quatro processadores mMips cada um com sua memória privada, utilizando uma NoC para intercomunicação entre os processadores. Para integrar os processadores com a rede de interconexão, foi utilizado um dispositivo para automatizar a transferência de pacotes. A arquitetura utilizada pode ser vista na Figura 1.

A implementação foi feita utilizando as ferramentas de síntese da Xilinx ISE 8.1i, sintetizando para o FPGA xc2vp30-ff1152 da família Virtex2-Pro.

O processador mMips (mini-MIPS) tem o código fonte de sua implementação sob licença LGPL. Este é baseado na arquitetura do processador MIPS, que é um processador RISC (*Reduced Instruction Set Computer*). O mMips é um processador de 32 bits, com pipeline de cinco estágios (*Instruction extraction, Instruction decoding, Execution, Memory access e Update registers*). Este processador possui apenas duas instruções de acesso à memória, e contém 32 registradores de 32 bits cada.

O código fonte da implementação do processador, embora completa, não implementa nenhuma memória interna ao *chip*, deixando o processador preparado para a utilização de memória *off-chip*. Assim fez-se a escolha de utilizar um bloco de memória IP (*intellectual property*) de duas portas (*dual port*) disponível pelas ferramentas da Xilinx. Com essa escolha, o barramento de acesso à memória do processador também foi alterado, utilizando um canal para entrada de dados e um canal de saída de dados, sendo os dois canais independentes, ao invés de usar o barramento *tri-state*, que já vem com a descrição deste processador.

A quantidade de memória sintetizada para cada processador foi igual a 32 KB, utilizando assim 60 blocos de memória (BlockRam) do FPGA.

A NoC utilizada chamada de SoCIN [15] é uma rede escalável baseada em um arquitetura de roteadores paramétricos RASoC (*Router Architecture for SoC*). Essa NoC utiliza roteamento baseado em um algoritmo de roteamento XY, o que assegura que a rede seja livre de *deadlocks* por um baixo custo.

Como a SoCIN possui opção de suporte para canais de largura até 32 bits, poucas modificações tiveram que ser feitas por conta da utilização de palavras com 32 bits.

O último elemento da nossa arquitetura utilizada foi o bloco de HandShake entre os processadores e a rede de in-

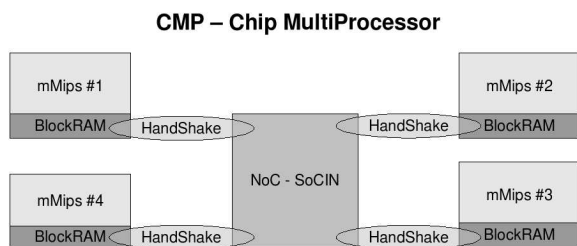


Figura 1. Arquitetura utilizada.

terconexão. Este módulo foi pensado para que em um trabalho futuro os módulos de processamento ou de NoC pudessem ser trocados por outros sem que para isso fosse preciso haver grandes mudanças no projeto.

Porém, para este estudo, o módulo de HandShake foi implementado de maneira simples, para que nesse primeiro momento fosse possível a validação e o estudo sobre todas questões ligadas à utilização de um mecanismo entre o núcleo de processamento e a NoC.

Assim, o Hand-Shake apenas acessa a memória pela segunda porta e na primeira etapa verifica se tem dados a serem enviados para NoC, na segunda etapa envia os dados, na terceira etapa verifica se tem dados a serem recebidos e na quarta e última etapa recebe os dados vindos da NoC. Para o funcionamento completo, uma área da memória foi reservada justamente para a ação entre o processador e a NoC.

4. Resultados

Após a implementação em linguagem de descrição de hardware (VHDL), é possível obter alguns valores expressivos em relação à área ocupada pelo projeto. Uma vez que a maioria dos FPGAs adotam uma arquitetura interna formada basicamente de Elementos Lógicos (*Logic Elements - LE*), e estes são compostos de memórias de programação (*Look Up Tables - LUTs*) e *Flip Flops*. A quantidade de LUTs e *Flip Flops*, serve em muitos casos para uma comparação justa entre diferentes propostas implementadas em FPGAs.

Podemos ver na figura 2, um gráfico mostrando a quantidade de *Flip Flops* e LUTs que os diversos módulos ocupam, sendo que para efeito de comparação foram inseridos dados de outra proposta de NoC e também de um processadores IP cores.

Assim, podemos ver claramente, que o processador mMips, embora seja um processador com número de instruções reduzido, segundo os dados, quatro processadores mMips ocupam uma área bem maior que quatro processadores MicroBlazes, chegando ocupar 35% mais *Flip Flops* e 23% mais LUTs.

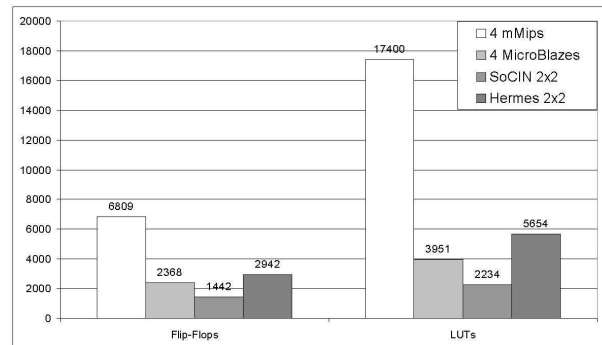


Figura 2. Resultados comparativos de área de processadores [4] e redes em chip [9].

Com relação a NoC, também para efeito de comparação, foram adicionados dados sobre a NoC Hermes 2x2. Nesta comparação podemos ver que existe diferença favorável para a NoC adotada. Sendo que a implementação da SoCIN consumiu 49% menos *Flip Flops* e 40% menos LUTs que a Hermes.

Na figura 3, está presente o gráfico comparativo, entre a implementação total realizada, e uma estimativa de área ocupada se o processadores adotados fossem MicroBlazes. Nesta estimativa pode-se notar uma redução de até 49% no número de *Flip Flops* e 31% menos LUTs.

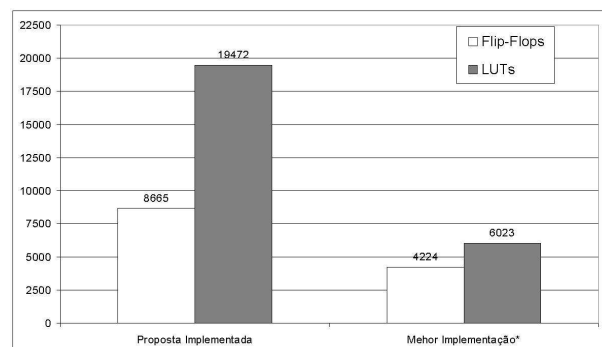


Figura 3. Áreas totais. (* Estimativa caso utilizasse processadores MicroBlazes)

Através dos comparativos apresentados, podemos notar um grande ganho em adotar o IP core MicroBlaze, isso reforça uma tendência em cada vez mais a utilização de módulos IP em projetos. Mas, alguns pontos devem ser levados em consideração, embora o IP MicroBlaze ocupe uma área muito menor de projeto o que reforça a ideia que este processador seja muito otimizado, para a implementação deste IP core é necessário o uso de ferramentas comerciais mais

específicas, podendo gerar um tempo maior de projeto por conta da adoção dessas ferramentas. Além disso, para fins acadêmicos, a necessidade em utilizar processadores com código aberto é um ponto chave para a adoção do mMips para estudos aprofundados de arquitetura e organização de sistemas.

A avaliação que podemos fazer com relação a SoCIN é positiva, uma vez que apresentou bons resultados de área. Porém, mesmo sendo comparada com outra NoC bastante encontrada na literatura [12] [8], é necessário em trabalhos futuros, uma ampliação desses resultados.

5. Conclusões

Com o aparecimento de novas tecnologias de integração, faz-se necessário trabalhos que avaliem plataformas e alternativas para futuras gerações de SoC. E em vista do mercado de sistemas embarcados, a utilização sistemas em FPGAs que ocupem pouco espaço e consumem pouca potência com um desempenho regular é de total importância.

Neste trabalho foi feito um apanhado sobre o futuro de processadores multiprocessados, além da avaliação de uma arquitetura que possui grande potencial para uso acadêmico devido a fácil manipulação de seus componentes.

Como trabalhos futuros podemos citar o melhoramento do dispositivo HandShake levando em conta outros trabalhos e alternativas [11], além da realização de testes com uma aplicação real como um decodificador Jpeg que muito utilizado em testes desse tipo de arquitetura, a fim de poder gerar mais dados a respeito do desempenho e potência da arquitetura proposta.

Referências

- [1] L. A. Barroso et al. Piranha: a scalable architecture based on single-chip multiprocessing. *International Symposium on Computer Architecture (ISCA)*, pages 282–293, 2000.
- [2] L. Benini et al. Network-on-chip architectures and design methods. *Comput. Digit. Tech.*, Março 2005.
- [3] L. Carro. *Projeto e Prototipação de Sistemas Digitais*. Editora da Universidade Federal do Rio Grande do Sul, 2001.
- [4] H. C. Freitas et al. Evaluating network-on-chip for homogeneous embedded multiprocessors in fpgas. *International Symposium on Circuits and Systems (ISCAS)*, 2007.
- [5] J. L. Hennessy and D. A. Patterson. *Arquitetura de Computadores*. Editora Campus, 3 edition, 2003.
- [6] P. Kongetira et al. Niagara: a 32-way multithreaded sparc processor. *IEEE MICRO*, pages 21–29, Março-Abril 2005.
- [7] M. T. M. León. *Fast modelling and analysis of NoC-based MPSoCs*. Eindhoven University of Technology, Holanda, 2006.
- [8] A. Mello et al. Virtual channels in networks on chip: implementation and evaluation on hermes noc. *Symposium on Integrated Circuits and Systems Design (SBCCI)*, Setembro 2005.
- [9] F. Moraes et al. Hermes: an infrastructure for low area overhead packet-switching networks on chip. *The VLSI Journal Integration*, pages 69–93, 2004.
- [10] K. Olukotun et al. The case for a single-chip multiprocessor. *Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2006.
- [11] J. C. P. Ortiz. *DESIGN OF COMPONENTS FOR A NoC-BASED MPSoC PLATFORM*. Eindhoven University of Technology, Holanda, 2005.
- [12] E. Salminen et al. Hibi-based multiprocessor soc on fpga. *IEEE Symposium*, 2005.
- [13] B. Sinharoy et al. Power5 system microarchitecture. *IBM J. RES. & DEV.*, Julho-Setembro 2005.
- [14] M. Vestias et al. Co-synthesis of a configurable soc platform based on a network on chip architecture. *Asia and South Pacific Conference on Design Automation*, pages 24–27, 2006.
- [15] C. A. Zeferino et al. Socin: A parametric and scalable network-on-chip. *Symposium on Integrated Circuits and Systems Design (SBCCI)*, 2003.