

# Uma Arquitetura de Multiprocessadores em Chip Reconfiguráveis, RCMP

*Raphael Fonte Boa Trindade, Carlos Augusto Paiva Silva Martins*  
*Pontifícia Universidade Católica de Minas Gerais*  
*Av. Dom José Gaspar 500 – CEP 30535 610 – MG - Brasil*  
[rfonteboa@gmail.com](mailto:rfonteboa@gmail.com) , [capsm@pucminas.br](mailto:capsm@pucminas.br)

## Resumo

*Como os processos de fabricação de circuitos integrados produzem CIs cada vez mais densos, melhores formas de aproveitar estes recursos devem ser desenvolvidas. Os CMPs se apresentam como uma solução para melhorar a utilização destes recursos. Mas além de boa utilização é preciso prover flexibilidade e escalabilidade e ainda fornecer bom desempenho. Para atender a estes requisitos, este trabalho apresenta uma arquitetura RCMP reconfigurável, para atender a diversas cargas de trabalho e fornecer alto desempenho.*

## 1. Introdução

Arquiteturas paralelas têm sido utilizadas, como solução para executar cargas de trabalho que requerem grande desempenho computacional, e podem ser paralelizadas. Estas arquiteturas podem ser classificadas em dois principais grupos: multicomputadores e multiprocessadores [1]. O primeiro grupo é a classe das máquinas onde a memória de cada elemento de processamento (EP) é privativa, e deve existir algum tipo de mecanismo para a troca de dados entre os EPs (ex: MPI). No caso dos multiprocessadores, a memória dos EPs é compartilhada, e a comunicação se dá principalmente através de operações de leitura e escrita na memória.

Estas arquiteturas apresentam dois problemas. O primeiro problema é a comunicação. Ao utilizar vários EPs, algum modo de realizar a comunicação entre eles deve ser fornecido, e esta comunicação se torna o gargalo do sistema, mais comumente existente nos sistemas multicomputador. Outro problema que as arquiteturas paralelas, especificamente os multiprocessadores, apresentam, é a baixa flexibilidade e escalabilidade. Nas arquiteturas paralelas, uma das características mais importantes é que os EPs se encontram agrupados e trabalham em conjunto. Por causa deste fato é possível utilizar EPs menos complexos. Ao fazer uso de EPs mais simples, recursos disponíveis em silício podem ser utilizados para implementar mais de um EP em um único chip.

Algumas arquiteturas deste tipo são conhecidas como Chip Multiprocessor (CMP) [2].

Mesmo as arquiteturas CMP, que fazem melhor uso dos recursos *on-chip*, ainda apresentam alguns problemas em relação à flexibilidade, escalabilidade e desempenho. Nas arquiteturas CMP os elementos de processamento ficam agrupados em um único chip, podem acessar elementos dentro ou fora deste e, muitas vezes, compartilham recursos internos (ex: cache L2) [2]. Estas arquiteturas fornecem uma interconexão veloz entre os EPs, logo a comunicação entre os EPs consegue ser muito mais veloz do que em sistemas multiprocessadores tradicionais (com comunicação por memória externa ao chip).

Os CMPs podem ser utilizados para executar tanto cargas de trabalho de grão fino (grande comunicação e pouco processamento) quanto para cargas de grão grosso (pouca comunicação e muito processamento). Com cargas de trabalho de grão grosso, o CMP pode se comportar como um multicomputador, apesar da comunicação entre EPs não necessitar de um mecanismo específico. Mesmo com a capacidade de se comportar como um multicomputador, os CMPs não possuem as características de flexibilidade e escalabilidade existentes nesta classe. Um Multiprocessador em chip comum, não pode ter processadores agregados ou removidos para suprir as necessidades de processamento e de desempenho de diversas cargas de trabalho. Uma opção para a solução dos problemas encontrados nos CMPs estáticos, é o uso de conceitos de computação reconfigurável. Isto pode prover um grande ganho de flexibilidade e escalabilidade, e ainda ganhos de desempenho [3].

O objetivo deste artigo é apresentar uma arquitetura RCMP reconfigurável (RCMP), onde as características desejáveis de ambas, arquiteturas multiprocessador e multicomputador, sejam unificadas em um único sistema. Para verificar a arquitetura, três protótipos foram implementados em FPGA, e medições para a avaliação e análise de desempenho foram realizadas.

## 2. Trabalhos Relacionados

Várias arquiteturas CMP foram propostas e implementadas [4, 5], e empresas de influência no

mercado de processadores adotaram este tipo de arquitetura como solução para os limites das arquiteturas sequenciais [6]. Apesar dos benefícios apresentados pelas arquiteturas CMP, ainda sim existe o problema da falta de flexibilidade da arquitetura. Ainda que apresente características desejáveis, um CMP estático não consegue executar uma grande diversidade de cargas de trabalho, e ainda manter bom desempenho na maioria dos casos. Isto ocorre pois os recursos fornecidos pela arquitetura são definidos em tempo de projeto, e não são ótimos para todos os tipos de aplicações. Após sua confecção, o chip não pode ser alterado, e as cargas de trabalho devem utilizar os recursos existentes.

Para trazer flexibilidade, escalabilidade e desempenho, conceitos de reconfiguração foram adotados em algumas arquiteturas. Estas arquiteturas utilizam conceitos de reconfiguração para obter ganhos de desempenho na execução das cargas de trabalho. O problema destas arquiteturas é que o uso de reconfiguração é feito somente em algumas partes da arquitetura, mais especificamente apenas em blocos designados para reconfiguração e implementação de lógica dedicada, como laços aninhados etc. [7] [8].

Para conseguir executar uma maior variedade de aplicações, é desejável que uma arquitetura além de ser CMP, apresente características de reconfiguração arquitetural. A reconfiguração não deve existir somente em blocos reconfiguráveis para implementação de lógica específica, mas sim como um recurso para toda a arquitetura, de modo que esta possa ser configurada para diferentes cargas de trabalho, ou seja, diferentes números de EPs, EAs e EIs. Até o presente levantamento do estado da arte, nenhum trabalho relacionado com o problema e proposta de solução deste foi encontrado.

### 3. Arquitetura RCMP

Neste trabalho uma arquitetura CMP reconfigurável (RCMP) é proposta, para executar uma grande variedade de cargas de trabalho e ainda apresentar grande flexibilidade e escalabilidade. A arquitetura possui uma estrutura flexível, composta por elementos de processamento (EP), elementos de armazenamento (EA), e elementos de intercomunicação (EI). Cada um destes elementos pode ser configurado e arranjado de diferentes modos, com o propósito de fornecer melhor desempenho. Além de possuir uma estrutura flexível, o comportamento da arquitetura também pode ser configurado. Ao alterar a lógica que cada elemento implementa, o comportamento da arquitetura pode variar, e em conjunto com sua estrutura flexível, o

RCMP pode executar diferentes cargas de trabalho com alto desempenho.

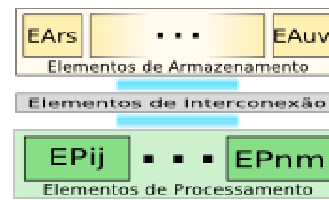


Figura 1 – Arquitetura RCMP

A arquitetura RCMP mostrada na Figura 1 apresenta características de reconfiguração em todos os seus elementos. Os elementos arquiteturais podem variar em número, configuração e tipo. Para atender as necessidades de processamento de diferentes cargas de trabalho, os EPs podem ser configurados de diversas maneiras e suas quantidades podem ser alteradas. Os recursos de memória podem ser aumentados ou diminuídos para se adequarem aos requisitos de memória *on-chip* e, a topologia definida pelos EIs pode mudar, e fornecer diversos modos para que a comunicação entre os elementos ocorra. A arquitetura RCMP pode ter elementos de diferentes tipos e diferentes configurações. Grupos de elementos, ainda que do mesmo tipo (ex: EPs) podem ter configurações diferentes ou implementar lógicas diferentes.

Os EPs podem ser processadores de propósito geral, de propósito específico ou ainda hardware dedicado. Dependendo das características da carga de trabalho, diferentes configurações de elementos de processamento podem existir. Arranjos heterogêneos podem possuir EPs com diferentes arquiteturas, tamanho de palavra, propósitos e frequências de clock.

Os EAs, são também reconfiguráveis, e a configuração destes pode ser alterada a qualquer momento. Os EAs podem ser bancos de registradores, compartilhados ou não, caches ou buffers, memórias privadas ou compartilhadas. O tamanho da palavra das memórias pode ser alterado dependendo da arquitetura dos EPs ou de quais elementos a memória se conecta (EIs etc). Problemas na comunicação com a memória podem ser solucionados ao configurá-la para suportar diferentes números de acessos simultâneos e diferentes velocidades de acesso. Do mesmo modo, as políticas de comunicação com EAs podem ser diferentes.

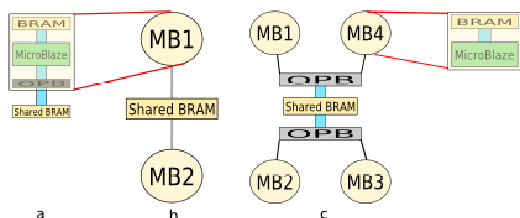
Para realizar a interconexão entre os EAs e os EPs, elementos de interconexão reconfiguráveis são utilizados. Estes elementos podem unir elementos arquiteturais heterogêneos, e podem ser de vários tipos, como crossbars, barramentos ou ainda conexões diretas. Os EIs podem ter largura de dados, políticas de acesso e velocidades diferentes.

### 3.1 Implementação da arquitetura

Para verificar a arquitetura proposta, três protótipos foram implementados. Os protótipos foram implementados com o uso da tecnologia de FPGAs. Esta tecnologia foi escolhida por estar disponível para uso no laboratório onde esta pesquisa é desenvolvida. Apesar da arquitetura ser completamente reconfigurável, e poder ser dinamicamente reconfigurada, os protótipos implementados somente apresentam reconfiguração estática.

Como plataforma de reconfiguração foi utilizado o dispositivo Xilinx XC2V1500, os elementos arquiteturais foram implementados através de softcores. Os *softcores* utilizados são de propriedade da Xilinx e foram todos fornecidos junto com o *Xilinx Embedded Development Kit (EDK)* [9]. Como elemento de processamento, foi utilizado o Xilinx MicroBlaze soft-processor, como elementos de armazenamento on-chip foram utilizados *cores* BRAM. Para conectar o MicroBlaze à memória local foi utilizado o link LMB (Local Memory Bus) e para fornecer um meio de comunicação entre os EAs e EPs foi utilizada a implementação da Xilinx do IBM *CoreConnect OPB (On-chip Peripheral Bus)* [9].

O Xilinx MicroBlaze é um soft-processor que implementa uma máquina RISC de 32-bits com um pipeline de execução em ordem, uma instrução executada por vez (*single-issue*) e três estágios. Este foi configurado a 100 MHz. Nas implementações realizadas nenhum tipo de cache foi utilizada. As BRAMs utilizadas são os *cores* responsáveis pela interface entre as memórias implementadas fisicamente no dispositivo e a memória configurada para implementação do protótipo. O LMB é um link de alta velocidade utilizado para conectar o processador à memória local (BRAM). E o barramento OPB que pode ser utilizado para conectar o processador a memórias, periféricos ou outros.



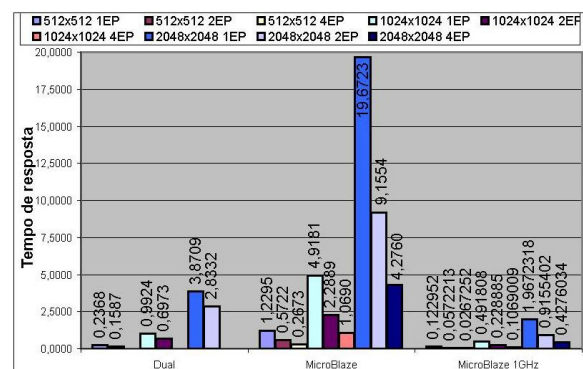
**Figura 2–Configurações CMP com MicroBlaze. (a) um EP. (b) dois EPs. (c) quatro EPs.**

A Figura 2 mostra em detalhes as implementações realizadas. As configurações apresentam um (a), dois (b) e quatro (c) EPs respectivamente. Em todas as implementações, cada processador possui uma memória local de instruções (16KB). Cada processador

foi conectado a um OPB, e este foi conectado a uma memória (compartilhada no caso de vários EPs) utilizada como memória de dados (32KB). Esta memória possui duas portas de acesso e dependendo da implementação, cada porta foi conectada a um OPB para fornecer dados aos processadores (Fig. 2).

### 4. Resultados

Para realizar a verificação da arquitetura e dos ganhos de desempenho obtidos através de seu uso, um estudo de caso com a operação de convolução de imagens foi escolhido. Esta operação foi escolhida, pois é utilizada em diversas áreas. Além disso, esta operação requer alto desempenho devido a sua grande complexidade computacional igual dada por:  $O(n^2 \cdot m^2)$ .



**Figura 3 –Tempos de resposta.**

**Tabela 1 – Speedups para diferentes imagens**

	512x512	1024x1024	2048x2048
Speedup 1EP/2EP Dual	1,4921235	1,4232037	1,3662642
Speedup 1EP/2EP MicroBlaze	2,1487116	2,1487116	2,1487116
Speedup 1EP/4EP	4,6005985	4,6005985	4,6005985
Speedup Dual/MicroBlaze 1GHz 1EP	1,9259550	2,0178608	1,9676887
Speedup Dual/MicroBlaze 1GHz 2EP	2,7734447	3,0465075	3,0945666

Como foram implementados três protótipos da arquitetura (Fig. 2), os testes puderam ser realizados nos mesmos e os valores foram obtidos através de medições. Foram utilizadas simulações somente nas configurações monoprocessadas, devido a limitações do EDK. Nos protótipos, o MicroBlaze foi utilizado como EP, e foi configurado com frequência de 100 MHz. Um programa que realiza operações de convolução foi implementado, para verificação dos protótipos. Para comparar os resultados obtidos, outro programa implementado em C utilizando pthreads, foi executado em uma máquina Pentium-III Dual Processor 1GHz. A implementação dos protótipos não possui nenhum tipo de cache, enquanto o Pentium-III possui uma cache L1 de 32 KB (instruções + dados) e L2 de 256 KB. Esta máquina foi escolhida como base para comparação, pois é um multiprocessador que se encontra disponível para a realização de testes, no laboratório onde este trabalho é desenvolvido (LSDC).

Os protótipos implementados somente apresentam reconfiguração estática, apesar de não ser uma limitação da arquitetura proposta.

Como pode ser observado na Figura 3, os tempos de resposta do MicroBlaze ficaram apenas cinco vezes piores que os do Pentium-III, no pior caso. Apesar de não possuir cache, não contar com uma arquitetura superescalar e ter frequência dez vezes menor do que o Pentium, o MicroBlaze conseguiu um resultado bastante satisfatório, dentro de suas limitações de implementação. A Figura 3 mostra ainda, uma projeção para um MicroBlaze com frequência igual a do Pentium-III. Caso fosse implementado desta forma, os tempos obtidos poderiam ser melhores do que os do Pentium-III em até 3.09 vezes. Esta implementação não foi possível devido a limitações do dispositivo reconfigurável utilizado.

Na Tabela 1 os speedups alcançados com o Pentium-III e MicroBlaze mostram que as implementações dos protótipos conseguiram obter, em todos os casos, speedups maiores do que os do Pentium III. O MicroBlaze conseguiu um speedup maior que dois em todas as configurações. Isto se deve ao fato de que, nas configurações com mais de um EP, o MicroBlaze executa menos ciclos do que nas versões mono-EP. Este comportamento está atualmente em investigação. Além disso, a Tabela 1 mostra que os speedups alcançados com a projeção para 1GHz do Microblaze apresentam ganhos de desempenho em relação ao Pentium III. Isto é observado pois os protótipos não enfrentam nenhuma sobrecarga ao compartilhar memória ou overheads de sistema operacional (SO), visto que na implementação da arquitetura proposta nenhum SO foi utilizado.

Os resultados obtidos mostram que a arquitetura proposta consegue apresentar bom desempenho através de características de reconfiguração. A flexibilidade apresentada pela arquitetura, permite que o número de EPs seja alterado, e com isto fornece escalabilidade para os protótipos. Este fato permite que a arquitetura seja configurada para melhor executar as cargas de trabalho, como visto através das implementações e dos resultados mostrados na Figura 1 e Tabela 1. Este tipo de flexibilidade não existe e nem é possível em sistemas como o Pentium III Dual Processor. Da mesma forma, arquiteturas que apresentam reconfiguração em alguns blocos [7] [8], não conseguem ser flexíveis o bastante para permitir alterações como as que foram realizadas nos protótipos da arquitetura proposta.

## 5. Conclusão

Neste trabalho foi apresentada uma arquitetura CMP reconfigurável. Através dos experimentos realizados foi possível verificar que a arquitetura proposta pode apresentar ganhos de desempenho significativos através do uso de conceitos de reconfiguração. Se a plataforma de implementação não apresentar limitações, os ganhos da arquitetura podem superar arquiteturas como a do Pentium-III.

Como observado nos resultados, os protótipos da arquitetura obtiveram um desempenho satisfatório dentro das limitações de implementação, e a projeção para um sistema próximo ao do Pentium III obteve ótimos resultados quando comparado com este. O objetivo deste trabalho foi alcançado, e duas principais contribuições foram feitas: a arquitetura RCMP foi proposta, e três protótipos da arquitetura foram implementados. Esta arquitetura une características desejáveis de multicomputadores e multiprocessadores em um único sistema e ainda prover alto desempenho.

Para conseguir que arquiteturas como a proposta sejam utilizadas amplamente, ideias como [10] têm sido desenvolvidas. Deste modo é possível utilizar a reconfiguração de propósito geral de modo transparente ao software.

Alguns dos trabalhos futuros são: implementação da arquitetura em FPGAs com maior capacidade e que permitam maiores frequências. Além disso, utilização de algum tipo de sistema operacional e implementar protótipos que apresentem reconfiguração dinâmica.

## Referências

- [1] Hwang, K; Xu, Z. Scalable Parallel Computing: Technology, Architecture, Programming. McGraw Hill, 1998
- [2] K. Olukotun et al., "The Case for a Single Chip Multiprocessor," Proc. 7th Int'l Conf. ASPLOS 1996, pp. 2-11.
- [3] Z. Guo et al. A Quantitative Analysis of the Speedup Factors of FPGAs over Processors, Int. Symp. FPGA. Fev 2004. pp 162-170
- [4] Barroso, L. A. et al. Piranha: A Scalable Architecture Based on Single-Chip Multiprocessing, Proc. of the ISCA 2000 pp. 282-293
- [5] Sang-Won, L. et al. RAPTOR: A Single Chip Multiprocessor. The first IEEE Asia Pacific Conference on ASIC's, 1999 pp. 43-49
- [6] Intel [www.intel.com](http://www.intel.com), AMD [www.amd.com](http://www.amd.com) Acesso: maio/2006.
- [7] D. B. Gottlieb. et al. Clustered Programmable-Reconfigurable Processors. In Proc. of the IEEE ICFPT, Dec. 2002 pp. 134-141
- [8] Z. A. Ye. et al. Chimaera, a high performance architecture with a tightly-coupled reconfigurable functional unit, Proc. of the ISCA 2000, pp. 225-235
- [9] Xilinx Inc. [www.xilinx.com](http://www.xilinx.com) Acesso em: maio/2006
- [10] M. Vuletić. et al. Programming Transparency and Portable Hardware Interfacing: Towards General-Purpose Reconfigurable Computing, Proceedings of the ASAP'04 pp. 339-351