

## Uso de reconfiguração dinâmica em FPGAs para redefinir arquiteturas multicore em tempo de execução

Autor: Téo Sobrino Alves

Orientador: Vanderlei Bonato

ICMC - USP

Email: teo.sobrino.alves@usp.br

### Objetivos

O objetivo deste trabalho é explorar os impactos de aplicar reconfiguração parcial dinâmica (DFX) em processadores *multicore*.

Para isso, será necessário executar a troca de partes dos *cores* em tempo de execução, avaliando os impactos produzidos no tempo de processamento e área ocupada para *testbenches*.

### Métodos e Procedimentos

Os materiais utilizados foram uma FPGA Xilinx, que conta com o recurso de DFX e um processador *soft core* (implementado em HDL) RISC-V de código aberto.

As principais ferramentas usadas neste trabalho foram os softwares Vivado, responsável por implementar um projeto em HDL numa FPGA e PYNQ (1), que consiste numa série de programas que estarão em execução no PS (*hard core* dentro do SoC FPGA) e permitem interagir e controlar o design implementado no PL (o chip FPGA).

A aplicação do DFX consiste em demarcar um componente como reconfigurável, em nível de HDL, gerando *bitstreams* parciais além do *bitstream* principal. O upload de *bitstreams* parciais em tempo de execução garante a

reconfiguração dinâmica do sistema. Tal troca fica como responsabilidade do software PYNQ.

Os *testbenches* utilizado foram uma rede neural do tipo CNN, que identifica números escritos à mão, da base de dados MNIST (3), que possui localidade espacial e temporal de dados forte, uma lista encadeada (LE), que possui localidade espacial e temporal fraca.

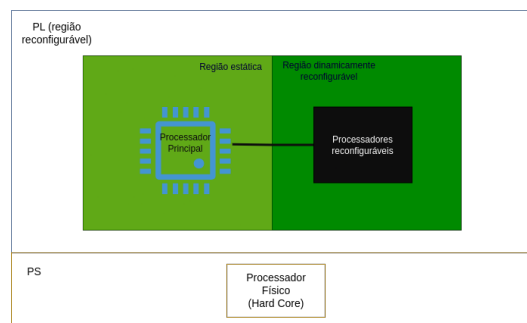


Figura 1: Visão arquitetural do sistema do ponto de vista do SoC FPGA

### Resultados

Os resultados obtidos até o momento refletem as medições de tempo, área e taxa de cache hit, assim como suas mudanças percentuais em relação à arquitetura estática com dois *cores*.

O módulo reconfigurável implementado foi utilizado para reduzir ou aumentar o tamanho da memória cache, gerando duas possíveis configurações de arquiteturas dinamicamente reconfiguráveis.

Para os métodos de obtenção e avaliação dos resultados foram tomados como base (4) e (2).

A utilização dos recursos lógicos estão presentes na tabela 1 e as medidas de interesse na tabela 2, para as quais foram tomadas a média entre 10 execuções, descartando o maior e menor valor.

## Conclusões

É possível ver que houveram ganhos no caso das diferentes arquiteturas dinâmicas para casos específicos, mesmo com o tempo adicional de reconfiguração, foi possível obter ganhos no tempo total de uma execução.

Quando há alta taxa de cache hit, o uso de uma memória cache maior pode trazer benefícios, como é o caso da CNN. Quando há baixa taxa de cache hit, uma cache menor traz menor penalidade no tempo para cada miss, beneficiando a LE.

## Agradecimentos

Essa pesquisa é financiada pelo processo nº 2023/15719-2 da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), na modalidade de Iniciação Científica.

## Referências

- [1] AMD. Pynq project for fpga programming enhancement.
- [2] DE ALENCAR MENDES, D. Enhanced run-time reconfigurable myokinetic interface for prosthetic control of artificial hands, 2024.
- [3] DENG, L. The mnist database of handwritten digit images. *IEEE SPM* 29, 6 (2012), 141–142.
- [4] PAPADIMITRIOU, K., ANYFANTIS, A., AND DOLLAS, A. An effective framework to evaluate dfx in fpga systems. *IEEE TIM* 59, 6 (2010), 1642–1651.

Tabela 1: Utilização de recursos lógicos para as arquiteturas testadas.

Arquitetura	LUTs	BRAM	Var. LUTs (%)	Var. mem.(%)
2 cores Estáticos	31.096	153	–	–
Cache diminuída	30.546	98	-5,65	-35.94
Cache aumentada	31.543	212	+1,43	+40.14
1 core Estático	18.330	76	-41,02	-51.33

Tabela 2: Tempos de execução e taxa de cache hit para os *testbenches*

Arquitetura	CNN: $T_{exec}$ (ms)	CNN: $T_{tot}$ (ms)	CNN: Cache hit (%)	LE: $T_{exec}$ (ms)	LE: $T_{tot}$ (ms)	LE: Cache hit (%)
2 cores Estáticos	173,849	173,849	70	349,586	349,586	33
Cache diminuída	189,113	195,042	44	324,132	330,061	29
Cache aumentada	164,583	170,770	88	389,812	395,999	35
1 core Estático	304,968	304,968	37	411,100	411,100	26