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 foi necessário realizar trocas de partes de *cores* em tempo de execução e avaliar os impactos no tempo de processamento e área ocupada para *testbenches* selecionados.

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 com o design implementado no PL – o processador *multicore* implementado.

A aplicação do *DFX* consiste em demarcar um componente como reconfigurável, em nível de HDL, gerando um projeto que pode ser modificado parcialmente em tempo de execução. Tal reconfiguração é feita pelo *software* PYNQ.

A arquitetura básica do sistema consiste em dois *cores*, um estático e outro dinâmico, sofre mudanças de forma automática, usando métricas obtidas em tempo de execução e pode ser vsita na Figura (1).

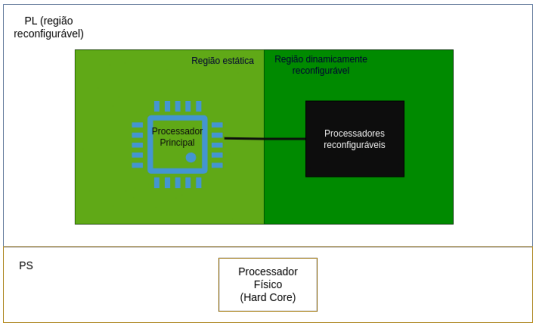

Figura 1: Visão arquitetural do sistema do ponto de vista do SoC FPGA.

Os *testbenches* consistem em executar programas em diferentes arquiteturas de CPU. Para arquiteturas dinâmicas foi garantido que houve uma reconfiguração durante o tempo de execução. Foram utilizados dois *testbenches*. O primeiro é uma rede neural do tipo *CNN*, que identifica números escritos à mão, da base de dados MNIST (3). O segundo, uma lista encadeada (LE).

O módulo reconfigurável implementado foi utilizado para reduzir ou aumentar o tamanho da memória cache, gerando duas possíveis

configurações de arquiteturas dinamicamente reconfiguráveis, uma com cache aumentada e outra com cache diminuída.

Resultados

Os resultados obtidos até o momento refletem as medições de tempo, área e taxa de cache *hit*, dados dispostos na Tabela 2 e o uso de recursos lógicos, dispostos na Tabela 1.

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

Conclusões

Houveram ganhos no tempo total dos *testbenches*, mesmo com o tempo adicional de reconfiguração.

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 python productivity.

[2] DE ALENCAR MENDES, D. Enhanced run-time reconfigurable myokinetic interface, 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