



|                   |  |
|-------------------|--|
| <b>Evento</b>     | Salão UFRGS 2014: SIC - XXVI SALÃO DE INICIAÇÃO CIENTÍFICA DA UFRGS  |
| <b>Ano</b>        | 2014   |
| <b>Local</b>      | Porto Alegre   |
| <b>Título</b>     | CReAMS Heterogêneo: Um estudo do ganho de performance de um sistema multi núcleo heterogêneo sobre um sistema homogêneo. |
| <b>Autor</b>      | JECKSON DELLAGOSTIN SOUZA  |
| <b>Orientador</b> | ANTONIO CARLOS SCHNEIDER BECK FILHO  |

Por muitos anos, a maioria dos sistemas embarcados eram projetados para executar tarefas específicas – especializadas. Contudo, com o avanço na tecnologia de produção de transistores e no projeto de circuitos integrados, os novos processadores embarcados agora são capazes de executar diversos tipos de operações. Assim, é possível satisfazer a demanda dos usuários de hoje, que exigem poder executar suas tarefas diárias usando o menor número de dispositivos possíveis. É esperado que cada nova geração de hardware embarcado acople mais recursos, tenha maior performance e consuma menos energia.

Para acelerar a execução de programas e usar de forma mais eficiente os recursos de um processador, é comum usar estratégias para execução paralela das tarefas. Quando uma sequência de instruções independentes entre si (que não operam sobre os mesmos dados) é despachada, um processador pode alocá-las para diferentes unidades funcionais e processá-las concorrentemente. Dessa forma, há um ganho de desempenho devido a exploração do paralelismo ao nível de instruções (*instruction level parallelism* – ILP). Por outro lado, os sistemas operacionais modernos são capazes de distribuir as *threads* de aplicações entre os diversos processadores de um sistema, de forma que elas possam ser executadas simultaneamente. Essa é a estratégia de explorar o paralelismo ao nível de *threads* (*thread level parallelism* – TLP) e é utilizada pelas organizações *multicore*.

A superescalaridade é a forma mais comum de explorar o ILP e ela é usada tanto pelos processadores de propósito geral – nas arquiteturas Intel x86, por exemplo – como nos embarcados – a exemplo dos ARM. Contudo, ela é cara, pois a cada palavra de instruções lida, o processador precisa avaliar as dependências dos dados para poder alocá-las as unidades corretas. Além disso, a organização superescalar clássica possui limites para a exploração do paralelismo que vão desde o grau de dependência entre as instruções até a quantidade de instruções que o alocador consegue despachar.

Não obstante, outra estratégia possível para aproveitar o ILP são as arquiteturas reconfiguráveis. Estas organizações são projetadas de forma a se adaptar a aplicação sendo executada e, por consequência, a suas dependências. Estas arquiteturas reconfiguram seu *datapath* de forma que o fluxo entre as unidades funcionais reflita as características da aplicação. Além disso, elas também já se mostraram mais eficientes do que os processadores superescalares.

Este trabalho sugere o uso de *Custom Reconfigurable Arrays for Multiprocessors Systems* (CReAMS), que é uma organização já proposta capaz de explorar ambos o ILP – usando arrays dinâmicos e adaptáveis de unidades funcionais no seu *datapath* – e o TLP – através do uso de vários núcleos de processamento. Esta organização já demonstrou, através de simulações, obter melhor performance se comparada com um processador base quando todos os núcleos são homogêneos, especialmente em aplicações que exploram amplamente o ILP.

Uma configuração homogênea de CReAMS é aquela em que todos os núcleos de processamento são iguais, ou seja, o array reconfigurável é igual em todas as réplicas dos núcleos. Por outro lado, também é possível construir CReAMS cujos núcleos sejam diferentes, heterogêneos. A vantagem desta configuração é que *threads* que exploram mais ILP podem ser alocadas para núcleos maiores (com mais unidades funcionais), enquanto que *threads* de menor ILP sejam despachadas para núcleos menores.

O objetivo deste trabalho será encontrar um conjunto de configurações heterogêneas de CReAMS que tenham melhor performance do que versões homogêneas de mesma área. Queremos mostrar que a exploração dos recursos do CReAMS é mais eficiente na versão heterogênea, pois além do ganho em desempenho, há um melhor aproveitamento espacial do chip e um menor consumo de energia. Para tal, usaremos quatro *benchmarks* diferentes – escolhidos para cobrir um amplo escopo de aplicações – a fim de medir a performance do CReAMS nas configurações escolhidas.