



Evento	Salão UFRGS 2015: SIC - XXVII SALÃO DE INICIAÇÃO CIENTÍFICA DA UFRGS
Ano	2015
Local	Porto Alegre - RS
Título	Análise da influência da reordenação de instruções no desempenho de processadores
Autor	FELIPE SALERNO PRADO
Orientador	ANTONIO CARLOS SCHNEIDER BECK FILHO

Nome: Felipe Salerno Prado
Orientador: Antonio Carlos Schneider Beck Filho
Instituição de origem: UFRGS

Análise da influência da reordenação de instruções no desempenho de processadores

No estado atual em que os processadores superescalares se encontram, utiliza-se uma quantidade desproporcional de área (mensurada em número de transistores) e potência, quando comparada ao retorno em termos de desempenho que este aumento oferece. Desta maneira, a grande motivação atual é conseguir aumentar o desempenho (mesmo que seja em pequenas proporções), desde que o custo para isto seja baixo. Estes processadores superescalares podem executar duas ou mais instruções simultaneamente. Todavia, os recursos que os vários diferentes processadores superescalares têm são diferentes entre si (e.g.: a quantidade disponível de unidades funcionais que cada um tem, quanto tempo cada uma demora para processar sua operação, etc). Em contrapartida, o compilador e o programa gerado são geralmente os mesmos.

Considerando este cenário, este trabalho visa a explorar tais diferenças a fim de tentar aumentar o desempenho, mesmo que em pequena escala, com custo zero em área ou potência. Isto será feito buscando a melhor ordem das instruções de um programa, já compilado, para ser executado em cada processador superescalar. Assim, parte-se da premissa que é muito provável que um mesmo programa executado em vários processadores, com diferentes organizações, apresente uma ordem escolhida pelo compilador que pode ser a melhor para uma organização e não para outra. Para esta exploração, a metodologia utilizada será descrita a seguir.

Primeiro, podemos considerar que um programa pode ser dividido em diversos blocos básicos (i.e.: uma sequência de instruções sem operações de desvio). Dependências entre instruções existem dentro de um bloco básico, e, para a lógica do programa estar correta, é necessário que elas sejam respeitadas. Por isso, a reordenação de instruções proposta tem de ser feita respeitando os limites de cada bloco básico. Através da análise de dependências entre instruções dentro de um bloco básico é possível modelar e linealizar (ordenação topológica) um grafo, representando cada nodo como uma instrução e cada aresta como uma dependência entre duas instruções. Por meio desta avaliação, pode-se obter uma ou mais ordenações topológicas de instruções de um mesmo bloco básico. Portanto, consegue-se gerar programas com lógica idêntica, mas com blocos básicos cuja ordem de suas instruções diferem das geradas pelo compilador.

O parser das instruções e a obtenção de diferentes ordenações topológicas estão sendo desenvolvidas em linguagem Java. Estes programas serão executados no simulador Multi2Sim, que permite um grande espaço de exploração, pois possibilita simular processadores com diversas configurações. Dessa maneira, será possível observar se a ordenação escolhida pela ferramenta proposta é melhor que a do compilador e, caso contrário, detectar possíveis padrões para os quais determinadas ordenações apresentam um melhor desempenho em uma determinada microarquitetura.