

The RISC penalty - Análise contextual.

André Marcelino de Souza Neves

Resumo - Uma breve análise dos tópicos discutidos no artigo *The RISC penalty*, de Tom Pittman, que mostra porque e em quais hipóteses processadores com arquitetura RISC podem ter desempenho inferior aos CISC's.

I. INTRODUÇÃO

O artigo analisado expõe alguns resultados experimentais que confrontam as arquitetura RISC e CISC. O significado de ambos estão expostos a seguir:

- **RISC:** *Reduced Instruction Set Computer*, ou Computador com Conjunto Reduzido de Instruções. Ao mesmo tempo que os arquitetos simplificam o conjunto de instruções, e consequentemente, a complexidade do circuito controlador, aproveiam para adicionar outros melhoramentos. No geral, as arquiteturas **RISC** compartilham as seguintes características:

- ISA (*Instruction Set Architecture*, ou Conjunto de Instruções da Arquitetura) pequena;
- Poucos formatos de instrução e modos de endereçamento;
- Instruções de mesmo tamanho;
- Instruções de execução em um único ciclo;
- Muitos registradores;
- Dois tipos de instruções: baseadas em registradores e baseadas em memória;
- * Acesso à memória apenas por instruções específicas de *load* e *store*

- * Outras instruções operam diretamente nos registradores, apenas.

- Execução pipeline facilitada.

O conjunto de instruções selecionado para fazer parte da arquitetura é baseado nas instruções mais comuns dos softwares em geral.

Consequência do uso de **RISC**:

- Instruções executam mais facilmente em um único ciclo, pois existe pouco delay de hardware no caminho crítico;
- Instruções simples e pequenas utilizam circuito eletrônico menor, o que abre espaço para empregar outras tecnologias no chip, como cache interna;
- Se a ISA possui instruções de mesmo tamanho, é mais fácil a implementação de arquitetura superescalar ou pipeline, que permite a execução de várias instruções simultaneamente, ainda que não estejam relacionadas.

- **CISC:** *Complex Instruction Set Computer*, ou Computador com conjunto complexo de instruções:

- Poucos Registradores;
- Vários formatos de instruções;
- Várias maneiras para endereçar memória;
- Operações baseadas em memória e registradores, podendo acontecer das formas registrador-memória, memória-registrador, registrador-registrador, e até

mesmo memória-memória, para alguns casos.

- ISA complexa e com várias instruções especializadas em determinadas tarefas.

Como é possível perceber, arquiteturas **RISC** e **CISC** são fundamentadas em filosofias opostas.

O autor afirma que não há fundamentos sólidos de pesquisa que permitem afirmar que o RISC é pior que CISC, e na época seria até uma falácia dizer algo do tipo, já que a arquitetura **RISC** estava tendo grande apoio por empresas que acreditavam ser a melhor opção

II. A PENALIDADE DO RISC

A penalidade do RISC se mostra quando resultados experimentais mostram que um menor número de transistores utilizados não afetam tanto a performance quanto uma menor largura de banda (*bandwidth*) da memória principal. Ao implementar cache (de dados e instruções) internas ao chip, espera-se que sejam necessárias menos buscas na memória principal, o que poupa tempo de execução.

O seguinte teste foi feito: Um programa de teste com um laço de repetição simples, escrito para 68000 (**CISC**), teve código emulado para execução em RISC, e posteriormente recompilado. Nesse caso, a recompilação dinâmica para execução em RISC teve bons resultados. Com isso, o mesmo processo de recompilação foi aplicado a softwares comerciais, mas nesse caso a performance caiu drasticamente. A explicação é que o programa simples, mesmo recompilado para **RISC**, poderia caber na memória cache, enquanto os software comerciais não.

Um resultado contraditório foi obtido com os software Word e Excel, da microsoft. Testes com os dois software mostraram maior eficiência em Pentium (**CISC**) do que no PowerPC (**RISC**), enquanto todos outros testes apontaram o contrário. Houve especulação de possível sabotagem da microsoft no PowerPC, devido a esse fato. Mas a explicação era simples: Word e Excel são programas grandes, que não cabem inteiramente na cache. Pentium's são **CISC**, e instruções **CISC** podem tirar proveito em situações onde há gargalo de acesso à memória, tendo melhor performance do que arquiteturas **RISC**.

O problema foi ter assumido que todos programas passam a maior quantidade de tempo em subprogramas pequenos, como foi o caso do loop, que foi bem executado em uma arquitetura RISC.

III. CONCLUSÃO

A tendência tem sido programadores se afastarem de programação em Assembly e ir em direção à códigos em C e C++, que costumam ser menos otimizados. Com isso, os programas tendem a ficar parecidos com Word e Excel, no contexto de testes citado acima.

Dado que cada tipo de programa tem demandas distintas, torna-se necessário uma análise lógicas das características de comportamento de hardware da execução do programas em cada um dos dois tipos de arquitetura. Só dessa forma é possível definir se naquele contexto específico, a melhor opção é usar arquitetura **RISC** ou **CISC**.

REFERÊNCIAS

- [1] Tom Pittman. *The RISC penalty*. IEEE Micro, Jan 1996.