SIMULAÇÃO DE PROCESSADORES COM MÚLTIPLOS TAMANHOS DE PALAVRA E A ARQUITETURA SPARC16

Rodolfo Azevedo Ivan Sichmann Freitas
rodolfo@ic.unicamp.br ra093887@students.ic.unicamp.br
INSTITUTO DE COMPUTAÇÃO
CNPq
Palavras-Chave: Arquitetura de computador - Simulação - ArchC

Introdução

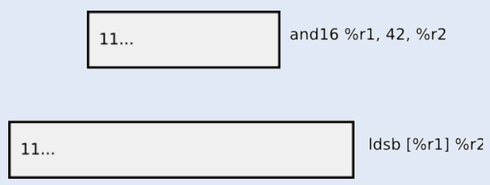
A linguagem ArchC é uma linguagem de descrição de arquiteturas que foi desenvolvida para para simular processadores que possuem tamanho de palavra fixo. No entanto, arquiteturas como SPARC16 podem ser simuladas se houver suporte à múltiplos tamanhos de palavra. Este trabalho implementou o suporte à múltiplos tamanhos de palavra na linguagem ArchC, através da implementação de árvores de decodificação.

Arquitetura SPARC16

SPARC16 é uma arquitetura RISC onde dois conjuntos de instrução coexistem: um conjunto de instrução da arquitetura SPARCv8 de 32 bits, e um conjunto de instrução de 16 bits. Em implementações feitas em *hardware*, o conjunto de 16 bits é implementado como um circuito lógico tradutor que expande cada instrução de 16 bits em sua respectiva instrução instrução de 32 bits de SPARCv8. Como existe conflitos de *opcode* entre instruções de 16 e 32 bits, a implementação em software requer que a decodificação para cada conjunto de instruções seja feita separadamente.

Desafio: decodificação

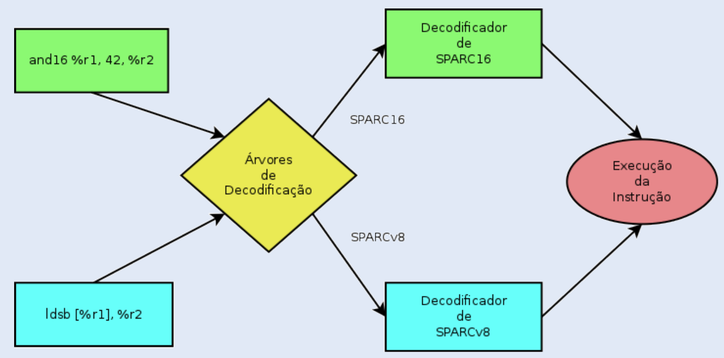
Suponha as seguintes instruções: `and16 %r1, 42, %r2` e `ldsb [%r1], %r2` das arquiteturas, respectivamente, SPARC16 e SPARCv8. Ambas possuem o mesmo valor no campo *opcode*: 0x3. Assim, não é possível decodificar com um único decodificador (ou árvore de decodificação), pois existe uma ambiguidade quanto ao *opcode* da instrução. Ainda, não é possível saber inicialmente se a instrução a ser executada é de 18 ou 32 bits sem um contexto definido no início da simulação.



Metodologia: árvores de decodificação

Os conflitos de *opcode* são resolvidos pela implementação de árvores de decodificação, que são um subconjunto dos tipos de instrução associados à um decodificador distinto. Começamos a execução do sistema em um estado definido, que por definição é SPARCv8, e trocamos de contexto para SPARC16 (ou vice-versa) **explicitamente** através de instruções específicas introduzidas em ambos os conjuntos de instrução. Para cada conjunto de instruções, é criada um árvore de decodificação específica, a qual é relacionada diretamente com o contexto de execução. Dessa forma, conjuntos de instrução conflitantes podem coexistir no mesmo sistema. Abaixo, um exemplo da troca de modo de uma árvore de decodificação:

```
void ac_behavior (callx) {
dbg_printf("callx 0x%x", ac_pc+(simm11«2));
this->set_decoder_tree(SPARCv8)
writeReg(15, ac_pc);
update_pc(1,1,1,0, ac_pc+(simm11«2), ac_pc, npc);
}
```



Declaração de árvores de decodificação

```
ac_format TipoA = "%op:6 %rs:5 %rt:5 %rd:5 %shamt:5 %func:6";
ac_format TipoB = "%op:6 %rs:5 %rt:5 %imm:16:s";
ac_instr<TipoA> add, sub;
ac_instr<TipoB> addi, subi;
ac_decoder_tree<Arvore1> = TipoA;
ac_decoder_tree<Arvore2> = TipoB;
ac_default_decoder_tree Arvore1;
```

Resultados e conclusões

O trabalho realizado neste projeto completou os seguintes objetivos:

- Um simulador do conjunto de instruções do SPARC16 (a menos das instruções estendidas).
- Implementação das árvores de decodificação.
- Implementação de um simulador de SPARC16 e SPARCv8 quase completo.

Referências

[1] Sandro Rigo and Rodolfo J. Azevedo and Guido Araujo The ArchC Architecture Description Language. [2] Ecco, Leonardo Luiz and Lopes, Bruno Cardoso and Xavier, Eduardo Candido and Pannain, Ricardo and Centoducatte, Paulo and de Azevedo, Rodolfo Jardim SPARC16: A New Compression Approach for the SPARC Architecture