

Introdução

No projeto e desenvolvimento de um sistema embarcado (um sistema computacional aplicado a uma tarefa específica, como uma TV digital, um celular, etc.), é muito valioso um protótipo da arquitetura do sistema para análise das propriedades emergentes. A linguagem de descrição de arquiteturas ArchC propicia um largo espectro de simulações a partir da descrição de um modelo de arquitetura, sendo portanto excelente para as atividades de prototipagem.

Acasm é uma ferramenta cujo objetivo é ampliar o suporte do ArchC no processo produtivo de um sistema embarcado. Acasm irá, a partir de um modelo ArchC, criar ferramentas binárias para esta arquitetura, como montadores e depuradores, essenciais para o suporte ao desenvolvimento dos softwares. Este trabalho visa estender Acasm para gerar montadores mais complexos de arquiteturas como o ARM, importante por ser extensivamente adotado como processador em sistemas embarcados.

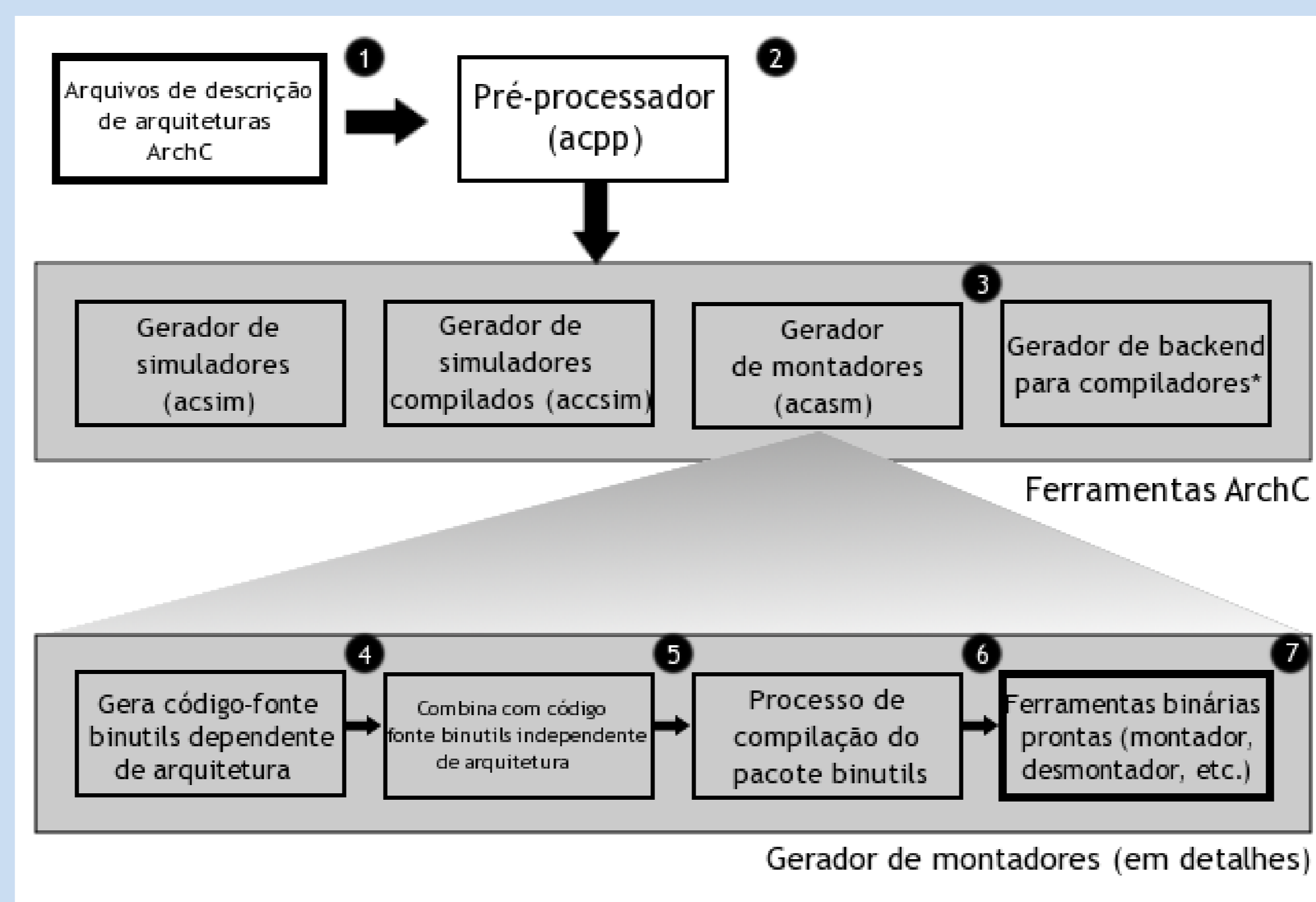


Figura 1: Diagrama de blocos do processo de geração das ferramentas binárias para uma arquitetura descrita em ArchC. Outras ferramentas ArchC são mostradas para ilustrar o conjunto completo. A ferramenta gerador de *backend* para compiladores é um projeto futuro.

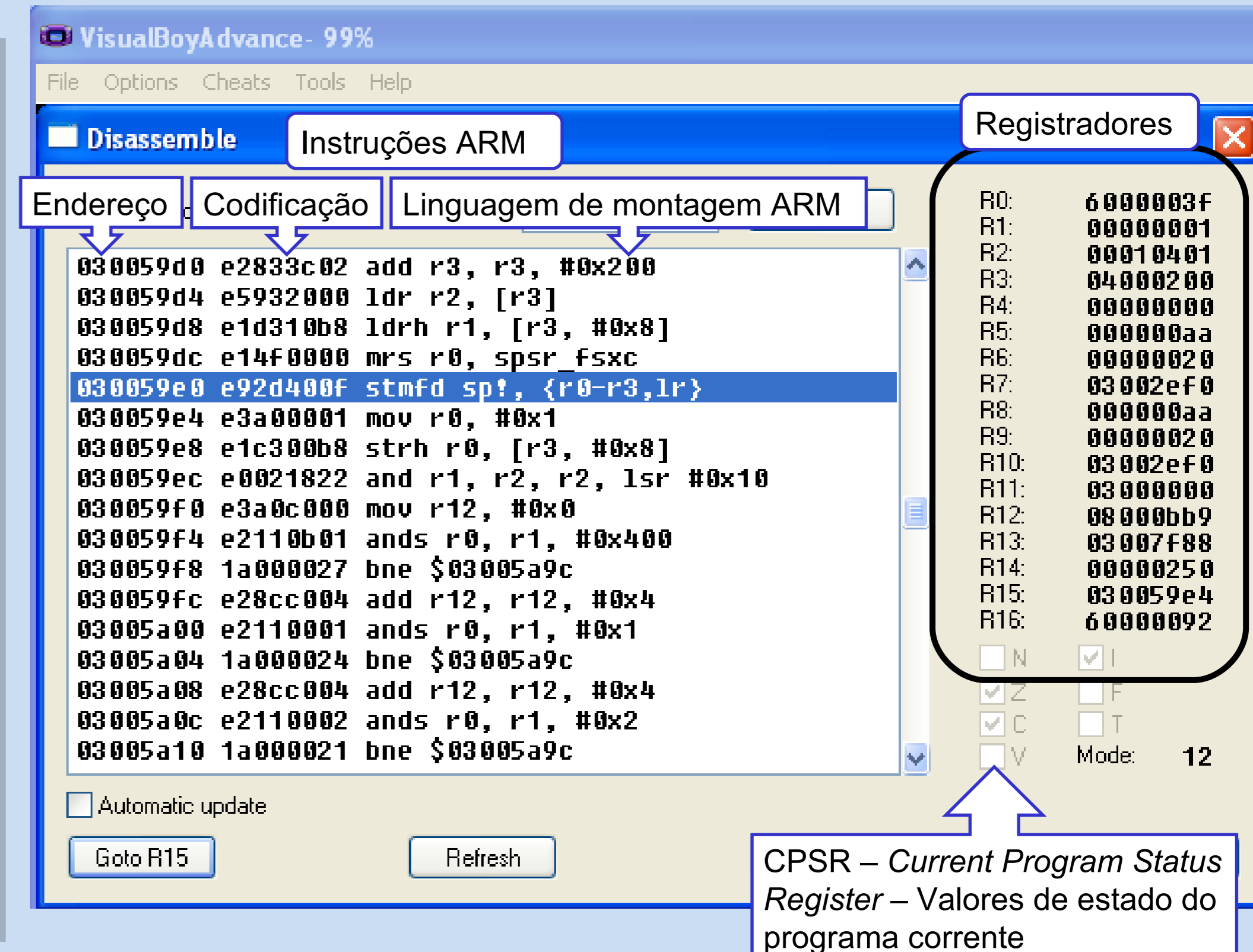


Figura 2: Exemplo de código em linguagem de montagem ARM em um emulador da plataforma Game Boy Advance (um portátil que ilustra o uso do processador ARM em sistemas embarcados). Este exemplo já é suficiente para mostrar diversas particularidades que desafiam o gerador de montadores para este alvo, como a instrução em destaque que possui número arbitrário de operandos, bem como a segregação em duas categorias de código (ARM e THUMB, sua versão compacta de instruções em 16 bits).

Metodologia

A figura 1 ilustra o processo de geração de ferramentas utilizando como entrada um modelo de arquitetura descrito com ArchC. O objeto de estudo é o gerador de montadores (Acasm) da etapa 3. A partir das informações extraídas de um modelo no passo 2 pelo pré-processador, o papel de Acasm é produzir código fonte correspondendo a um *backend* para a suíte de ferramentas binárias de software livre GNU binutils.

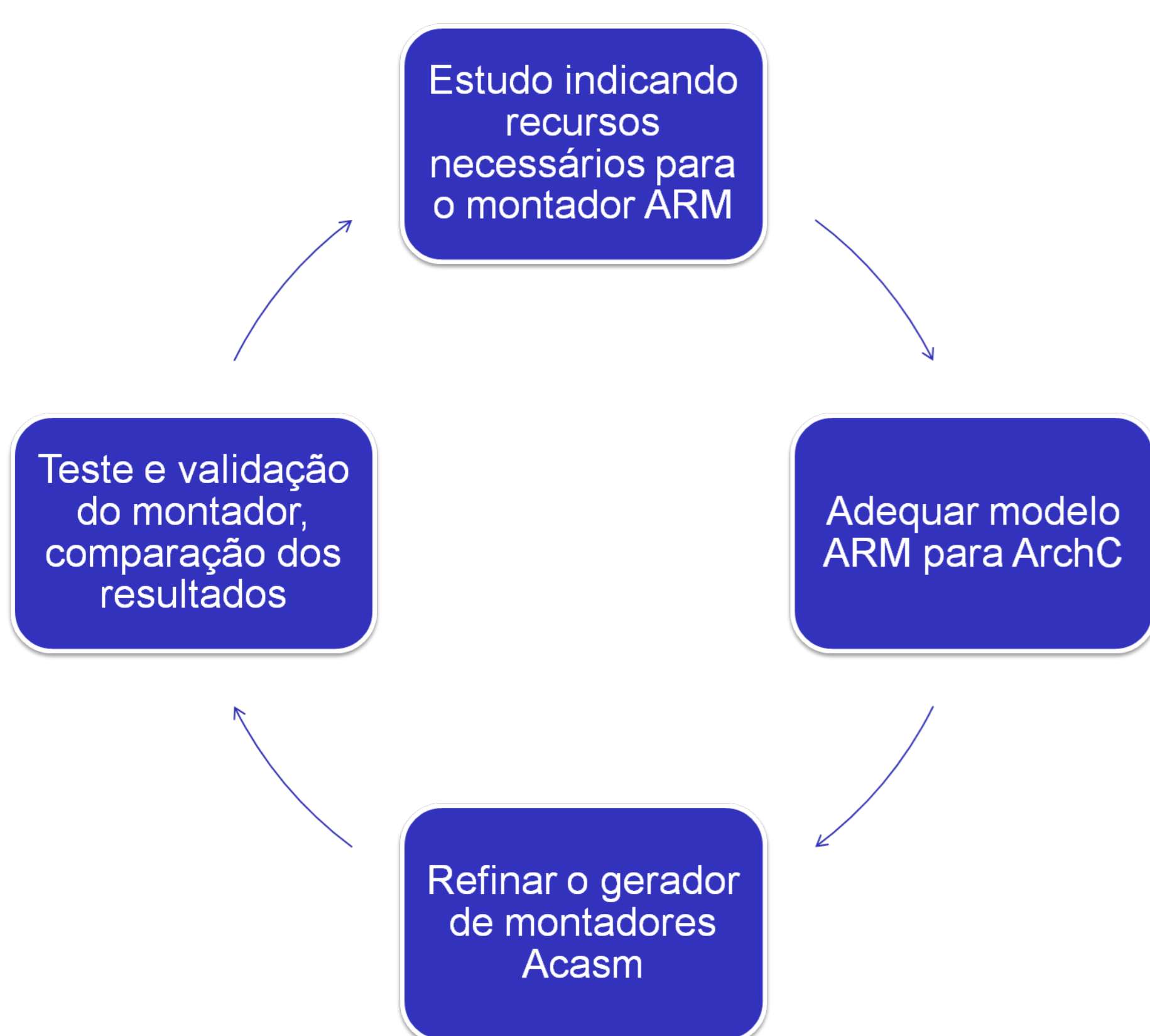


Figura 3: Diagrama cujo objetivo é mostrar o ciclo de desenvolvimento empregado na execução do trabalho

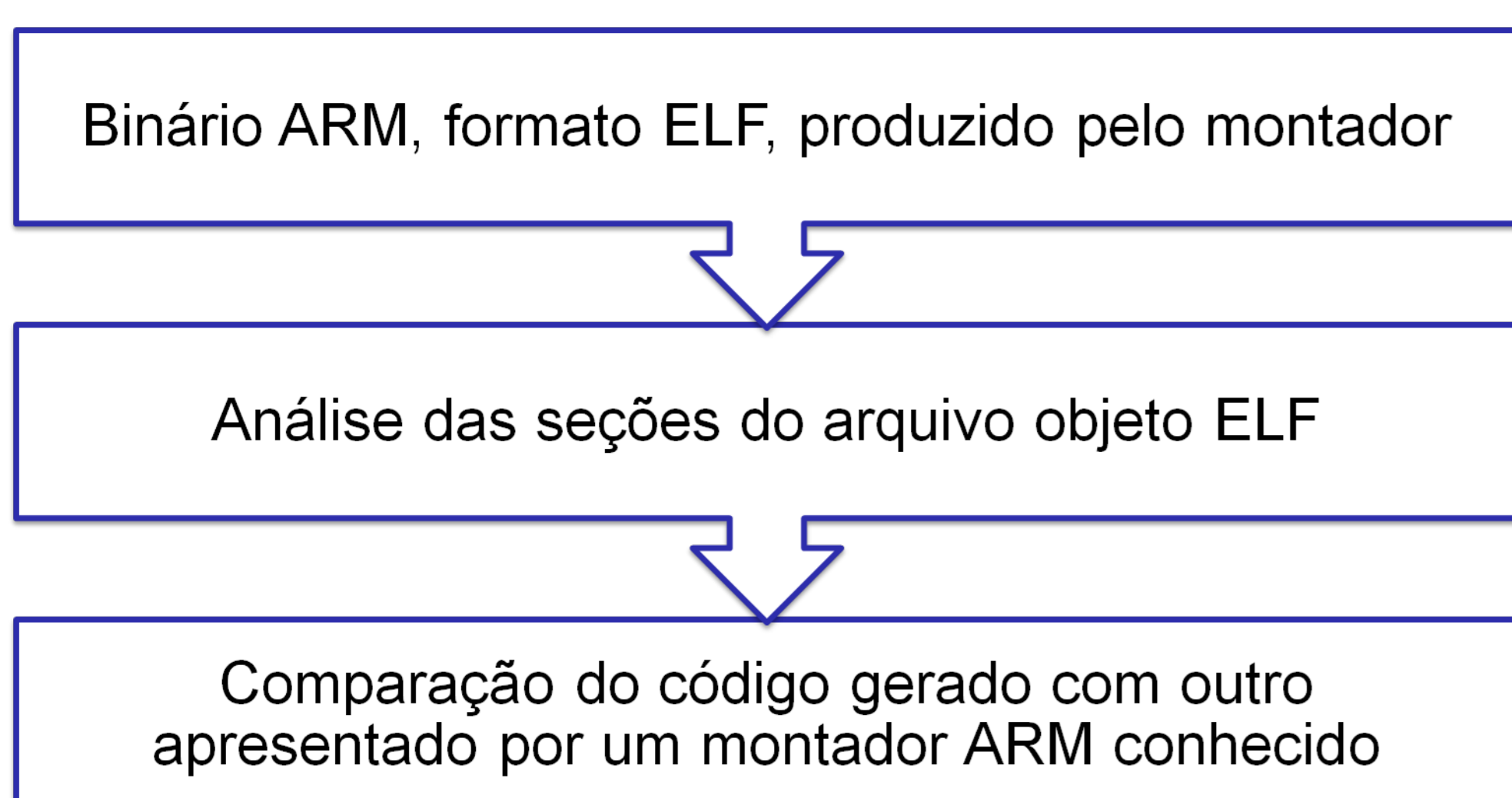


Figura 4: Fluxo das atividades de validação do montador produzido

Resultados e Discussão

No desenvolvimento do projeto foi produzido um modelo do montador da linguagem de montagem da arquitetura ARM, bem como nova semântica de linguagem de descrição de arquitetura foi adicionada ao ArchC para ampliar as informações oferecidas por um modelo de modo a viabilizar a geração de montadores mais complexos que antes não era possível.

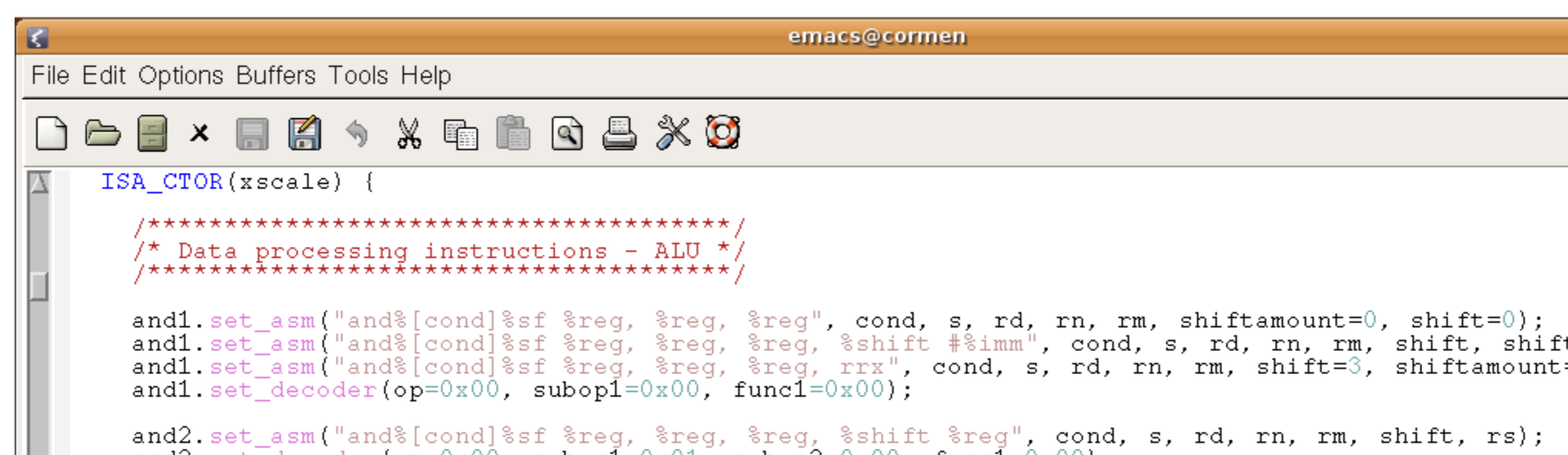


Figura 5: Trecho do modelo ARM descrito em ArchC, exemplificando como a sintaxe das instruções é estabelecida.

Exemplos da fase de modificações implementadas no Acasm para geração de um montador a partir do modelo criado incluem:

- Elaboração do mecanismo de mnemônico sobrecarregado com mais de 1 sufixo, necessário para linguagens que abrangem um segmento inicial comum (um mnemônico) associado a uma rica combinação possível de sufixos (figura 4). No ARM isto é valioso, pois todas suas instruções possuem pelo menos 1 sufixo (o condicional, que atrela uma condição de execução à instrução);
- Elaboração do recurso de listas de símbolos. Até então, no Acasm, a especificação da linguagem de montagem se limitava basicamente a definir onde são esperados determinados símbolos (como um número imediato, um endereço ou um nome). O ARM, entretanto, possui instruções que utiliza um número arbitrário de símbolos, como instruções de carregamento/armazenamento de múltiplos registradores. Tais instruções utilizam como parâmetro uma lista de registradores, o que configura um cenário onde não há como prever exatamente o número de símbolos com antecedência.

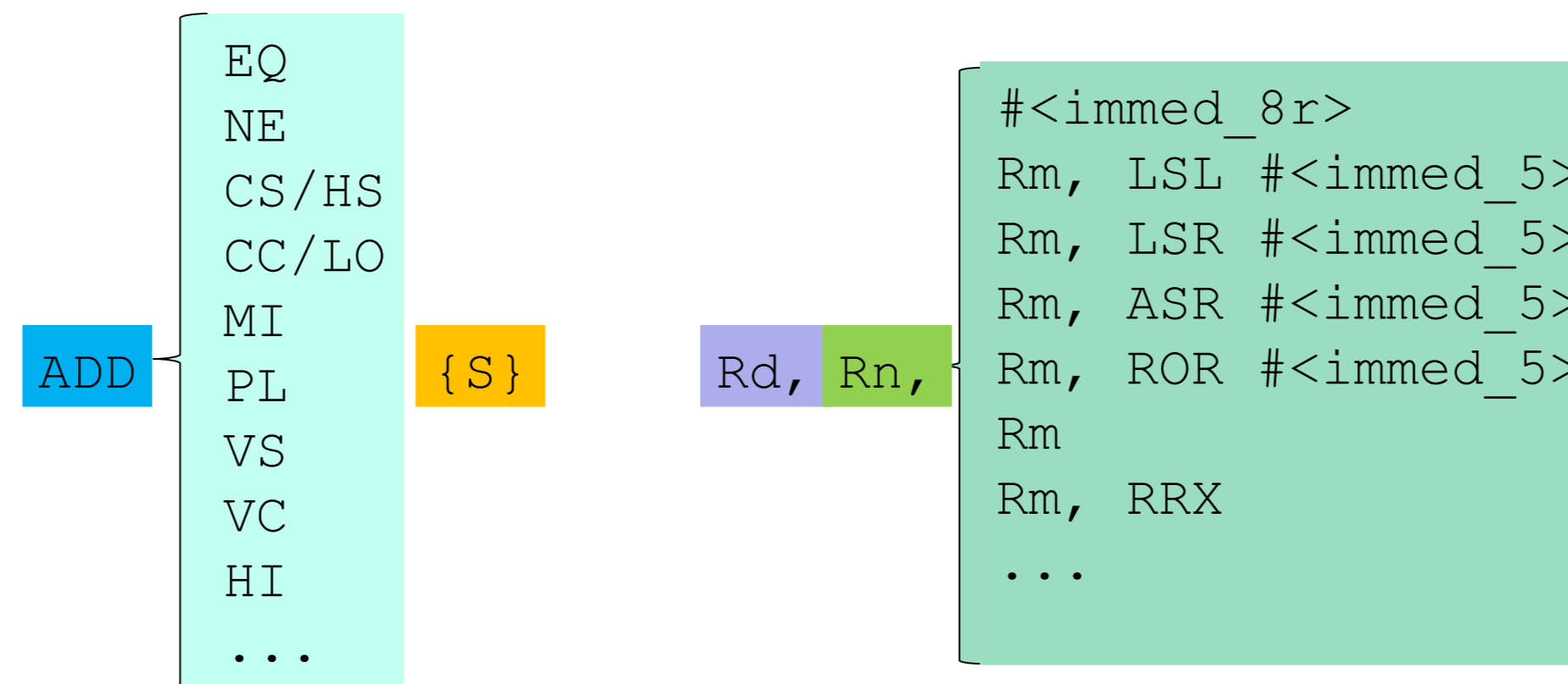


Figura 6: Exemplo de uma instrução ARM de processamento de dados (ADD) e sua sintaxe. Esta é a sintaxe básica da maioria das instruções de processamento de dados, e já apresenta complexidade moderada, como o uso de dois sufixos opcionais após o mnemônico. Um exemplo concreto de uso da sintaxe é "ADDEQ R0,R1,R2".

Conclusão

Neste trabalho apresentou-se o mecanismo de geração automática de montadores para modelos de arquitetura de sistemas embarcados descritos em ArchC, bem como uma modelagem de montador ARM. As características adquiridas para o gerador de montadores aperfeiçoado incluem:

- Maior fatoração e facilidade para descrições de sintaxes para as linguagens de montagem;
- Refinamento do mecanismo de símbolos em listas definidos pelo usuário do ArchC;
- Maior eficiência na manipulação da tabela de espalhamento de instruções, no algoritmo implementado no montador gerado.

Referências Bibliográficas

[1] The ArchC Team. The ArchC Language Reference Manual, 2007.

[2] Prabhat Mishra, Nikil Dutt. Processor Description Languages. Morgan Kaufmann, 2008.

[3] ARM Limited. ARM Architecture Reference Manual, 2000.

[4] Alexandre Baldassin. Geração automática de montadores em ArchC. Dissertação de mestrado, Instituto de Computação, UNICAMP, Campinas, Abril 2005.

[5] Steve Furber. ARM System-on-Chip Architecture. Addison-Wesley Professional, 2000.