

Projeto de MC723

Relatório Parcial

Nesta primeira parte do trabalho nós estudamos o microprocessador AVR. Apesar de ser um RISC, ele é muito mais complicado do que esperávamos. O fato dele ter memória junto no mesmo chip pode complicar a implementação. A memória flash dele é necessária já que todas instruções de programa devem residir na sua memória flash não volátil. Ele tem 32 registradores de um byte.

A criação do simulador AVR. Após ler o manual de referência sobre ArchC, começamos a desenvolver o simulador **funcional** com o conjunto de instruções do AVR. Este modelo, funcional, simula o comportamento do conjunto de instruções, executando todas as operações de uma dada instrução durante um único ciclo do SystemC. Nesta etapa, criamos o arquivo `avr1.ac`, com as características do processador Atmel AVR como tamanho da palavra de 16 bits, a memória do chip e formato little endian. Também começamos a descrever o conjunto de instruções da arquitetura no arquivo `avr1_isa.ac`. Para tanto, nós nos baseamos no conjunto de instruções fornecido pela Atmel e pode ser encontrado em : http://www.atmel.com/dyn/resources/prod_documents/doc0856.pdf .

No processador AVR existem um total de 13 modos de endereçamento de dados. Colocamos isto no arquivo `avr1_isa.ac`, na parte inicial `ac_format`. Tivemos problemas em entender o formato dos bits na instrução, pois não ficam consecutivos, por exemplo, uma instrução de soma sem carry temos: 0000 11rd dddd rrrr, em que os “r”s se referem aos bits que endereça os registradores Rr fonte (de 0 até 31) e os “d”s o registrador destino Rd (também de 0 até 31).

Para a segunda parte, prosseguiremos com o desenvolvimento do simulador funcional e a descrição do conjunto de instruções. Nos concentraremos em entender mais a arquitetura AVR, pois a implementação em ArchC é bem amigável.