

082

IMPLEMENTAÇÃO EFICIENTE DE UMA ARQUITETURA LOAD/STORE EM VHDL. *Everton Höfler Ferreira, Fernando Gehm Moraes* (Faculdade de Informática, Departamento de Fundamentos da Computação, PUCRS).

Este trabalho compreende o desenvolvimento completo de uma arquitetura Load/Store utilizando unicamente a linguagem de descrição de hardware VHDL com os passos de: especificação da arquitetura; uso de simulador e tradutor de linguagem de montagem; simulação funcional em VHDL; implementação VHDL sintetizável; síntese lógica e física; implementação física em plataforma de prototipação (FPGA); e validação do hardware com analisador lógico. As motivações para o desenvolvimento deste trabalho são: (i) dispor de um núcleo processador reconfigurável que possa ser facilmente modificado de acordo com a aplicação do usuário (ou seja, um ASIP – application specific instruction-set processor); (ii) utilizar este processador como o módulo executor de software em projetos integrados de sistemas computacionais compostos por hardware-software para aplicações embarcadas; (iii) capacitar o grupo de pesquisa à implementação de sistemas digitais completos em um único FPGA (systems-on-a-chip). A relação entre diversas estruturas de uso geral na linguagem VHDL e seu efeito quantitativo sobre as dimensões da implementação foram avaliados e aproveitados durante as fases de projeto e implementação da arquitetura. Por exemplo, para o banco de registradores implementou-se duas versões: array de 16 palavras de 16 bits e estrutura de seleção por multiplexação. A versão com array tem código VHDL muito mais compacto, porém no momento da síntese obteve-se resultados com ocupação de 700 CLBs (blocos lógicos), já a versão utilizando multiplexadores apresentou uma taxa de ocupação de 400 CLBs. Este exemplo mostra como uma implementação cuidadosa no nível estrutural permite economia de área (número de CLBs), e eventualmente viabiliza a implementação da arquitetura com limitados recursos de hardware. Este processador, assim como o controlador, encontra-se operacional na placa de prototipação, e seu código aberto para a comunidade acadêmica (CNPq/PUCRS).