PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

Graduação em Ciência da Computação

André Brochado Campolina

**ARQUITETURA DE COMPUTADORES I: ATIVIDADE DO ARTIGO**

Resumo de Artigo

Belo Horizonte

2012

As linguagens de descrição de hardware (HDL) sempre receberam uma atenção especial da indústria, dado que alterações no design de baixo nível dos componentes pode influenciar fortemente a eficácia de circuitos lógicos. A grande variedade de linguagens HDL ofertada em certa época dificultava a integração e o uso de diferentes tecnologias em conjunto, visto que para cada fornecedor havia uma diferente documentação e padrões diferentes a serem seguidos.

A necessidade de se trabalhar em uma mesma linguagem fez com que o governo dos Estados Unidos, junto de grandes companhias do setor, formalizasse uma linguagem HDL que seria adotada pelas demais empresas. Surge daí o VHDL, que desde o início já apresentava falhas nos níveis mais baixos da programação, tornando o trabalho de designers de circuitos mais difícil e extenso. Pensando no lado comercial da descrição por linguagem de circuitos lógicos, criou-se a linguagem Verilog HDL, que tem grande semelhança com as linguagens C-Like e foi amplamente absorvida pelos designers.

Além da facilidade de programação, o Verilog apresenta vantagens operacionais em relação ao VHDL. Do ponto de vista de simulações, a primeira tem comandos nativos que permitem ao programador analisar delays e transições com maior facilidade e sem se valer de ferramentas caras e complexas para VHDL.

Quando colocadas lado a lado, Verilog e VHDL, é possível notar uma diferença clara entre as linguagens. Como citado no artigo, um projeto de milhares de portas escrito em Verilog demandou recursos consideravelmente menores que o mesmo projeto executado em VHDL. No entanto, as preocupações da indústria no que diz respeito às linguagens VHDL não está somente na linguagem atual, e sim na linguagem que substituirá, ou não, o Verilog. Atualmente, as produções de chips de silício estão alcançando precisões micrométricas de construção, o que demanda da linguagem uma eficiência maior no baixo nível, cabendo às futuras linguagens, ou atualizações das já existentes, lidar com os novos padrões de contrução para tirar o melhor proveito do hardware.