SUGESTÕES PARA O ENSINO DE LINGUAGEM DE DESCRIÇÃO DE HARDWARE

Luis Antonio Prata Barbosa – luisprata@ig.com.br
Ricardo Zelenovsky – zele@unb.br
UnB, Faculdade de Tecnologia , Departamento de Engenharia Elétrica
Campus Universitário Darcy Ribeiro
CEP: 70919-970
Brasília, DF

Resumo: *Linguagens de descrição de hardware podem representar grande potencial de crescimento tecnológico e econômico para o país, principalmente quando aplicadas ao desenvolvimento de circuitos integrados. Neste contexto, a Universidade assume papel fundamental na formação de profissionais capazes de trabalhar com esta tecnologia. Porém, é importante discutir e ponderar a forma como este assunto é muitas vezes abordado. As ferramentas empregadas para o ensino de tais linguagens, geralmente, abstraem os detalhes de criação do projeto e, muitas vezes, os usuários acabam por relegá-los a segundo plano. Este é um equivoco comum em cursos de linguagem de hardware, especialmente quando o curso tem uma duração exígua. A deficiência nestes conhecimentos irá se refletir em dificuldades no aproveitamento dos resultados de outros trabalhos, isolando os esforços e prejudicando o ritmo de crescimento. É importante que a universidade tenha a preocupação de formar profissionais com um entendimento completo, e que não se limite apenas à própria linguagem. Este trabalho apresenta como sugestão a inclusão de alguns tópicos no ensino de linguagens de programação de hardware, com a finalidade de aumentar o domínio do profissional sobre todas as partes envolvidas no processo de síntese de hardware e, por conseqüência, contribuir para melhorar a cooperação entre o meio acadêmico, assim como, o emprego dos resultados no mercado.*

Palavras-chave*: ensino, linguagem de programação de hardware, HDL*

1. INTRODUÇÃO

As linguagens de descrição de hardware, identificadas pela sigla HDL, do inglês "hardware description language", favorecidas pelo seu grande potencial para o crescimento econômico do país, vêm ganhando espaço no cenário acadêmico. São muitas as universidades brasileiras que, preocupadas com a modernidade de seus currículos, passaram a abordar esse tema. Com o surgimento de novos cursos e profissionais recém-formados é natural que ocorram equívocos e impropriedades, além de deficiências em passar aos alunos os conceitos precisos e os aspectos fundamentais dessa área. Ao lado disso tudo, são inevitáveis as discussões sobre a melhor maneira de se transmitir os conhecimentos necessários para o emprego de tais linguagens.

O presente artigo expõe, baseado nas dificuldades pessoais de seus autores, alguns aspectos importantes no ensino desta disciplina. É enfatizado o desequilíbrio entre o ensino da própria linguagem e o ensino sobre todo o contexto que envolve o assunto e que se torna fundamental ou forte catalisador na geração de resultados.

A motivação para criação deste artigo se deve, principalmente, à disparidade de desempenho de alunos com boa formação acadêmica sobre o assunto e que, de forma surpreendente, apresentaram baixa produtividade quando submetidos a desafios reais ou quando empregados para dar continuidade a trabalhos já iniciados, onde precisavam integrar pacotes de código já prontos.

O artigo apresenta, segundo critério de seus autores, alguns aspectos identificados como pontos vulneráveis. Esse critério baseia-se em experiências práticas e dificuldades reais vivenciadas durante a orientação de alunos em final de curso e em pós-graduação.

2. ORIENTAÇÃO AO PROFESSOR

Este trabalho não tem a intenção de apresentar o conteúdo completo da disciplina nem de entrar em detalhes. O objetivo é apenas defender a abordagem de alguns pontos importantes para um bom entendimento do assunto, pois esses pontos, muitas vezes, acabam sendo relegados a segundo plano. É sugerida a abordagem de uma série de tópicos, mas o grau de profundidade e a ordem de apresentação depende da análise do próprio professor, levando em consideração o tempo disponível e o grau de conhecimento prévio dos alunos.

3. ASPECTOS ABORDADOS

Os aspectos discutidos neste artigo foram definidos como proposta para complementar o ensino de HDL e estão baseados nas deficiências identificadas em alunos de final de curso e de pós-graduação. A partir dessas deficiências ou dificuldades, foi possível enumerar alguns itens que certamente minimizariam ou sanariam estes problemas. Os aspectos que serão tratados são:

- Descrição do Problema a ser resolvido com HDL;
- Fluxo de Projeto;
- Particularidades da Linguagem;
- Exercícios para Melhor Entendimento do Sistema e
- Utilização de Sistemas de Teste.

3.1 Descrição do Problema a ser resolvido com HDL

Todo curso de HDL deve iniciar com uma apresentação clara sobre o que é uma linguagem de descrição de hardware e quais problemas se resolvem com sua utilização. Este ponto, apesar de ser o ponto de partida do curso, nem sempre é tratado com a importância devida. Uma boa descrição dos problemas que serão solucionados com essa tecnologia facilitará em muito o progresso do aluno durante o curso. Não raro é o equívoco inicial de alguns alunos pensando que HDL é uma forma de descrição semelhante às usadas por sistemas de auxilio por computador (CAD) para representação de esquemáticos e placas de circuito impresso. A possibilidade de entrada de dados no formato de esquemático pode confundi-los ainda mais. Portanto, é recomendada cautela e uma apresentação precisa do problema, enfatizando principalmente o caráter comportamental da síntese.

Inevitavelmente, com o tempo o aluno passa a entender corretamente os conceitos, entretanto, a ótica correta desde o início do estudo pode representar um ganho adicional, especialmente em cursos com duração exígua.

Num nível mais profundo pode-se também abordar quais são as vantagens e desvantagens de se utilizar esta tecnologia. Analisar alguns casos onde fica evidente o uso destes dispositivos e outros onde não vale a pena usá-los. Neste contexto existem as questões de custo, escala e velocidade de processamento. Por exemplo, podem ser apresentados dispositivos programáveis substituindo lógica de interconexão, ressaltando que tais

dispositivos são simples e baratos. Pode-se, também, citar situações onde usar estes dispositivos fica muito desvantajoso, já que chips prontos teriam um custo muito menor. Seria interessante mostrar aplicações de processamento de imagens que geralmente requerem grande esforço computacional e são favorecidas pelo uso de dispositivos programáveis. Além de apresentar exemplos onde a escala de produção indica o uso de um chip dedicado (ASIC) que utilizou HDL em sua construção.

Para turmas de computação, cujo contato com hardware é menor do que em turmas de engenharia elétrica, esses cuidados devem ser ainda maiores. Por conhecer menos sobre os detalhes eletrônicos, é mais natural para esses alunos aceitarem um primeiro entendimento, que, se não for bem apresentado, pode ser equivocado.

3.2 Fluxo de Projeto

Em muitos cursos, a prática da linguagem acaba por eclipsar as demais etapas do projeto usando HDL. Por isso, antes de se começar o estudo da sintaxe e dos detalhes da linguagem sugere-se que seja abordado o fluxo do projeto. É importante que o aluno saiba quais são as etapas envolvidas e qual a finalidade de cada uma delas. A maioria dos ambientes de desenvolvimento acaba por abstrair esses detalhes e muitas vezes dificultam a idéia global do que é um projeto dessa área. Neste caso é sugerido explicitar quais as entradas e saídas de cada uma das etapas, com atenção especial aos arquivos de restrições de projeto e de informações específicas do dispositivo. O aluno precisa saber que, mesmo quando o código já está pronto, algumas informações são ainda necessárias para completar todo o processo. Mesmo que a disciplina não tenha um laboratório e que o código nunca venha a ser programado em um dispositivo, realizar todos os passos dará ao aluno uma visão mais ampla e condições para lidar com todos os detalhes que um projeto real exige.

Com a finalidade de se favorecer a clareza, pode-se optar em deixar os arquivos de simulação e testes para serem apresentados num segundo momento. Entretanto, deve ser ressaltada a importância das simulações e da construção dos arquivos para essa finalidade. É perfeitamente cabível que neste ponto se faça alusão às diversas ferramentas existentes no mercado para a realização de cada fase, mostrando ao aluno que existem diversas alternativas para se de realizar o processo.

3.3 Particularidades da Linguagem

As linguagens de descrição de hardware, na sua compilação, possuem particularidades que podem gerar muita confusão entre alunos habituados com programação de software. Principalmente, no que diz respeito a aspectos como o paralelismo e a forma não seqüencial de execução dos comandos. Apesar de serem importantes, estas particularidades estão diretamente ligadas à sintaxe e ao ensino da própria linguagem, de forma que vêm sendo bem trabalhadas em sala de aula e os alunos avaliados não apresentaram deficiências neste item. Este aspecto foi inserido apenas pela sua importância e não há maiores sugestões para sua abordagem.

3.4 Exercícios para Melhorar o Entendimento do Sistema

Uma das grandes dificuldades identificadas entre os estudantes desta área é com relação à utilização de sistemas de desenvolvimento ao qual não estão habituados. É bastante razoável que haja dificuldade na utilização de um sistema inédito, entretanto, o problema toma um maior vulto quando a dificuldade está associada ao não entendimento do processo. Desta forma, a inserção de um novo ambiente pode se tornar numa boa ferramenta de avaliação do entendimento do processo pelos alunos e uma boa oportunidade para o próprio aluno fixar solidamente seus conhecimentos.

É sugerido que, além do ambiente inédito, sejam inseridos exercícios que trabalhem o emprego de códigos já prontos e códigos pré-compilados. A falta de exercícios desse tipo no ambiente acadêmico, gera enormes dificuldades nas aplicações comerciais, onde tal tarefa é bastante comum. Além disso, uma deficiência desse tipo acaba induzindo o aluno ou o profissional a tentar desenvolver tudo novamente, ou seja, reescrever todo o código, quando poderia aproveitar o trabalho já disponível.

Alguns aspectos importantes que devem ser trabalhados nos exercícios são: a identificação da hierarquia do projeto; a adaptação de projetos às necessidades específicas, quer seja alterando o dispositivo alvo, quer seja alterando a disposição dos pinos de entrada e saída; e ainda a substituição de bibliotecas específicas de um fabricante por outras equivalentes.

Uma boa proposta de exercício, caso haja tempo hábil para tal, seria alterar um pacote de código já existente na Internet de forma a realizar algumas tarefas adicionais e implementá-lo em um dispositivo real. Boas sugestões de pacotes prontos e com código fonte aberto aparecem em OPENCORES (2004).

Finalmente, outro ponto importante a se trabalhar é a análise de erros e avisos que os sistemas apresentam. A falta de conhecimento sobre esses pontos também prejudica a formação do aluno. Normalmente tais erros levam o programador desavisado ou inexperiente a desperdiçar muito tempo tentando resolvê-los. A proposta, neste caso, é mostrar como tratar os erros mais comuns e ensinar como o aluno deve proceder caso se depare com um erro ou sinal de alerta ainda não conhecido.

3.5 Utilização de Sistemas de Teste

Uma parte importante do processo de emprego de HDL são os testes. Eles são importantes não só para a garantia do funcionamento do projeto, como também para auxiliar na compreensão prática durante o curso. As próprias linguagens especificam uma série de recursos, disponíveis apenas para simulação e testes do sistema.

É fundamental conscientizar o aluno do quanto as simulações e os testes são importantes para seus projetos e, é claro, ensiná-lo a usar estes recursos de maneira apropriada. Para conseguir um entendimento completo e oferecer flexibilidade, deve-se trabalhar com as várias formas disponíveis para testes, tanto as gráficas, quanto as que utilizam a própria linguagem para gerar sinais e elementos específicos de teste.

Deve-se mostrar os níveis de simulação e teste. Num primeiro momento, podem ser gerados arquivos de simulação funcional importantes para validação da funcionalidade do sistema. Em seguida, usando os arquivos de simulação e testes gerados na fase após o roteamento pode-se comparar os resultados, enfatizando que estes últimos levam em consideração os atrasos dos sinais e das portas já mapeados dentro do dispositivo.

Como a linguagem possui atributos específicos de teste e de síntese, é importante trabalhar com os dois cenários, mas mantendo clara a separação entre eles e, assim como acontece na síntese, é importante que o aluno consiga não apenas criar seus próprios ambientes de teste, como também empregar códigos de teste já existentes ou escritos por terceiros, isso pode economizar muito esforço.

4. CONSIDERAÇÕES FINAIS

As linguagens de descrição de hardware podem realmente trazer avanços no desenvolvimento tecnológico do país e as universidades brasileiras estão contribuindo com este cenário. Novos cursos nesta área estão surgindo e, desta forma, quase que inevitavelmente, surgem alguns equívocos no tratamento do assunto. Não raro os cursos de HDL se resumem apenas à programação da linguagem deixando o aluno distante de uma posição onde os resultados possam ser utilizados na prática. O aluno, ao completar sua

formação, ainda enfrenta dificuldades, principalmente, para aproveitar trabalhos já realizados, fazendo com que muito tempo seja desperdiçado e que o compartilhamento das informações, assim como o crescimento tecnológico, sejam muito prejudicados.

Este artigo apresentou alguns tópicos que, de acordo com a avaliação de seus autores, merecem melhor atenção nos cursos de HDL e podem ajudar muito no processo de aprendizagem. Porém, sabe-se que a fluência efetiva nessa área só pode ser alcançada com o tempo e a experiência, e que em boa parte dos cursos, não existe tempo hábil para trabalhar todo o conteúdo necessário.

REFERÊNCIAS BIBLIOGRÁFICAS

Empresa Xilinx, **ISE In-Depth Tutorial**. www.xilinx.com

Organização OpenCores, **www.opencores.org**

DOUGLAS, J. S. **HDL Chip Design**. Estados Unidos: Donne Publications, 2000.

SUGGESTIONS FOR THE EDUCATIONAL SCHEME OF HARDWARE DESCRIPTION LANGUAGES

Abstract: *Hardware Description Languages may represent a great potential of technological and economic growth for the country, mostly when applied to the development of integrated circuits. In this context, the University assumes a basic role in the formation of professionals capable to work with this technology. However, it is important to question how this subject is studied. The tools used to teach such languages, generally, abstract the details of the creation of the project and, many times, the users leave it for the second plan. This is a common mistake in courses of HDL, especially with short-time courses. The deficiency in this knowledge will be reflected in difficulties and in the exploitation of the results of other work, isolating the efforts and harming the growth rhythm. It is important that the university has the concern to form professionals with a complete insight of it, and not so restricted to the proper language only. This work suggests the inclusion of some topics in the education of hardware programming languages, with the purpose to increase the domain of the professional on all the involved parts and in the process of hardware synthesis, for consequence, to contribute to improve the cooperation between the academics, as well as, applying the results within the market.*

Key-words*: education, hardware description language, HDL;*