

Pontifícia Universidade Católica de Minas Gerais

Curso: Ciência da Computação

Disciplina: Arquitetura de Computadores I

Nome: Roger Rubens Machado

Matrícula: 430533

Verilog HDL VS. VHDL

Para o usuário de primeira vez

As duas línguas têm diferentes forças técnicas que significativamente diferencia o seu foco de mercado. Força técnica Verilog HDL é na sua capacidade para representar hardware digital (e hardware analógico em Verilog-A) a forma como um projetista de hardware seria imaginar e implementar o projeto. Verilog HDL tem built-in tipos de hardware predefinidos líquidos (fio, wor, varinha, tri, etc), eliminando os problemas inerentes à interpretação VHDL. Verilog HDL também tem porta, e modelagem mudar de nível, permitindo fundições ASIC para representar fielmente sua cela bibliotecas. Modelagem mais complexa como pullups / pulldowns, carga dinâmica força de partilha e de sinal pode ser precisamente modelada com relativa facilidade. Verilog HDL proporciona ao desenhista uma sintaxe da linguagem simples e estrutura. Esta capacidade, ao contrário de VHDL, permite ao projetista de aprender a língua rapidamente e desenvolver modelos mais conciso e eficaz. A facilidade de utilização de HDL Verilog traduz bem nos ambientes de simulação apoiados e seu desempenho características. Modelos de VHDL, por outro lado, são inerentemente indecisos, devido a programação e, portanto, não fornecem um quadro para desempenho adequado de suas respectivas ferramentas de simulação. Verilog HDL foi projetado com características que são necessários para modelar o ambiente do sistema. Esta capacidade é uma deficiência significativa em VHDL. Verilog HDL também permite o uso de código de controlo dentro de um modelo para segurar que os erros são capturados no início do processo de design. O código de modelação não tem nenhuma analogia física e simplesmente não pode ser tratada facilmente em VHDL. Há outras áreas de importância técnica que proporciona ao Verilog HDL alguns benefícios significativos quando comparados com VHDL. A capacidade de fazer referência a um sinal além das fronteiras do módulo é necessário porque fornece um acesso arbitrário para as variáveis. Esta ligação pode ser fornecida sem nenhum requisito estrutural, tais como declarar portas. Além disso, a HDL Verilog suporta "eventos chamados" que

fornece uma abstração muito necessária no alto nível em que "os eventos ocorrem" e "eventos se consumido". Este uso da linguagem é extremamente benéfico quando a modelagem do "ambiente de banco de ensaio" de um design. VHDL se baseia exclusivamente em comunicação através de valores de sinal e da inconsistência inerentes associados com este tipo de processo. Linguagem Verilog HDL apóia tarefas de sistema e funções como parte do língua. Um designer pode incorporar comandos de controle para a língua, assim, tornando modelo de desenvolvimento e depuração do design muito mais eficaz e significativamente mais produtivo. VHDL simplesmente não suporta esta vital depuração princípio em tudo.

O impacto da síntese no mercado HDL é uma das mais significativas razões para o crescimento HDLs ao longo dos últimos anos. Verilog HDL permitiu este impacto porque ferramentas de síntese pode mapear diretamente de e para o HDL Verilog sem a necessidade de um "pacote" especial. A maioria das declarações de linguagem pode ser sintetizada, eliminando a necessidade de grandes quantidades de graus de parametrização. VHDL, no entanto, devido à sua capacidade de programação deve limitar drasticamente a utilização da sua construções de linguagem. VHDL deve ser altamente parametrizado (limitado em conteúdo) no desenvolvimento de modelos. Na realidade, isso de forma eficaz limita os "benefícios" da programação únicos VHDL. Como a maioria das pessoas modelo com a finalidade de implementação (estruturas físicas) e o câmbio atual tem sido para o uso de ferramentas de síntese, o usuário vai perceber a síntese benefício maior produtividade da utilização de HDL Verilog. Apesar de síntese benefícios VHDL também, os ganhos de produtividade são mínimos e mais importante sem síntese, utilidade VHDL como uma verdadeira linguagem de design diminui significativamente. Uma das áreas mais importantes que qualquer HDL deve abordar é a área de necessidades futuras (usabilidade futuro). Hoje em dia, a maior parte do semicondutor grande empresas estão implementando ou planejando fazer design submicron profunda (menos de 0,35 microns) nos próximos 12 meses. Para uma HDL para atender às necessidades de projeto submicron profunda, deve ser absolutamente capaz de representar baixo nível construções (Gate e potencialmente alterar modelos de nível) com precisão. Isto inclui a capacidade de lidar com os requisitos de tempo muito específicos para submicron profunda. Como um novo usuário, embora você pode não ver a necessidade imediata de profunda submicron em seus projetos, os fornecedores de ASIC e FPGA estão ocupados trabalhando em fazer esta parte de sua próxima geração de silicone, portanto é importante manter ritmo com esta tendência futura.

Para resumir, VHDL foi desenvolvido (para o Departamento de Defesa dos EUA) para proporcionar uma consistente linguagem de modelagem para a documentação de projetos de hardware digital. A linguagem nunca foi destinada a ser usada para fazer o projeto real. No entanto, a manter uma suposta vantagem competitiva, as empresas individuais EDA exerciam considerável influência, recursos e dólares para forçar a

língua para se tornar uma linguagem de design. Estas mesmas empresas EDA implementavam suas versões semiúnicas próprios da linguagem em fases diferentes durante o seu desenvolvimento. Estes modelos VHDL meios que foram desenvolvidos em um sistema, não pode ser executado em um sistema diferente. O idioma é difícil de aprender e ainda mais difícil de utilizar. É extremamente detalhado, especialmente ao nível da porta, quando a informação de tempo é específica e considerável. Verbosidade VHDL provoca graves problemas de memória ao tentar simular médio a grandes projetos. Fornecedores de ASIC têm sido muito relutantes em fornecer bibliotecas VHDL nível do portão que incluem tempo total por causa do tamanho dos modelos e os tempos de simulação anormalmente longos associados validação de um projeto relativamente simples. Os autores da VHDL foram impulsionados pelo EUA DOD, que não tem interesse material em produtividade de projeto. Complexo de VHDL sintaxe interfere com a produtividade de projeto e não oferece qualquer estratégico vantagem que iria melhorar a qualidade do design. Isto essencialmente enfraquece a força básica do VHDL, a produtividade conseguida através de uma metodologia com base em projeto top-down. Verilog HDL tem sido desenvolvido e continuará a evoluir para abordar o necessidades e aplicações comerciais da comunidade de design que fez dela a língua mais bem sucedido em uso hoje. A comunidade de design tem investido quase 20 bilhões de dólares em Verilog HDL e ferramentas relacionadas ao longo dos últimos 8 anos. A capacidade para tratar maiores construções de linguagem de nível são bem suportados no linguagem, juntamente com seus rocha sólida estruturais (nível portão e interruptor) pontos fortes. Enquanto designers e suas empresas têm de obter alta qualidade inovadora produtos para o mercado do mundo sensível ao tempo em que todos competem, Verilog HDL continua a ser a solução dominante. Quase todos os principais computador fabricante, o sistema de desenvolvedor, ASIC e semicondutores fabricante usa Verilog HDL como sua linguagem de modelagem. Para o usuário primeiro tempo a seleção de HDL Verilog HDL como sua modelagem linguagem será uma decisão muito sábia. Isso vai significar, há uma série de ferramentas disponível a partir de inserção de esquemas de síntese para simulação em várias escalas de preço e em inúmeras plataformas de PCs a mainframes. Há também inúmeras bibliotecas disponíveis a partir de uma variedade de fontes que suportam modelos de tempo completos baseados com todas as funcionalidades necessárias de atraso para atender seu projeto crítico necessidades. Existe também um grande recurso de Verilog HDL talento de engenharia que possui tinham experiência com a linguagem para projeto comercial prático para fornecer assistência crítica, se for necessário. Há muitos que você pode HDLs escolher, mas apenas um que tem provado uma e outra vez que ela é a única escolha para projetos reais.

Observação sobre outros artigos pesquisados

Ambos concordam que Verilog e VHDL são linguagens de descrição de hardware que são usados para escrever programas para chips eletrônicos. Essas línguas são usados em dispositivos eletrônicos que não compartilham a arquitetura básica de um computador. VHDL é o mais velho dos dois, e é baseado em Ada e Pascal, herdando assim as características de ambos os idiomas. Verilog é relativamente recente, e segue os métodos de codificação da linguagem de programação C.

VHDL é uma linguagem fortemente tipada, e scripts que não são fortemente digitados, são incapazes de compilar. A linguagem fortemente tipada como VHDL não permite a miscigenação, ou operação de variáveis, com diferentes classes. Verilog usa tipagem fraca, que é o oposto de uma linguagem fortemente tipada. Outra diferença é a sensibilidade do caso. Verilog é sensível a maiúsculas, e não reconhecer uma variável, se o caso usado não é consistente com o que era antes. Por outro lado, VHDL não é case sensitive, e os usuários podem alterar livremente o caso, desde que os caracteres no nome, ea ordem, permanecer o mesmo.

Em geral, Verilog é mais fácil de aprender do que VHDL. Isto é devido, em parte, à popularidade da linguagem de programação C, fazendo com que a maioria dos programadores familiarizados com as convenções que são usados em Verilog. VHDL é um pouco mais difícil de aprender e programa.

VHDL tem a vantagem de ter muito mais constructos que ajuda na modelagem de alto nível, e reflecte a operação do dispositivo a ser programado. Complexas de dados e pacotes de tipos são muito desejável quando programação de sistemas grandes e complexos, que podem ter um monte de peças funcionais. Verilog tem nenhum conceito de pacotes, e toda a programação deve ser feita com os tipos de dados simples que são fornecidos pelo programador.

Referências:

- <http://www.freewebs.com/gurprasad/downloads/A%20Comparative%20analysis%20of%20Verilog%20HDL%20over%20VHDL.pdf>
- <http://www.differencebetween.net/technology/difference-between-verilog-and-vhdl/>