Análise Comparativa Entre Linguagens de Descrição de Hardware: Verilog e VHDL

Janailson Maciel de Queiroz e Ernano Arrais Junior

[1]Universidade Federal Rural do Semi-Árido – UFERSA – Pau dos Ferros – RN - Brasil
Grupo de Desenvolvimento e Simulação - GDeS

Janailsom.jmq@hotmail.com, ernano.arrais@ufersa.edu.br

Abstract. *With the complexity of modern integrated circuits, Hardware Description Languages (HDL) have become extremely important as tools in the development and prototyping of electronic designs. They are primarily useful for describing the behavior and structure of a hardware. Among them the most known and used are Verilog and VHDL. And there is no consensus on which of these two HDLs is the best, so this work, based on various criteria, and basic circuits with the help of the HDL synthesis and simulation tool, Quartus Prime, sought to compare them. Then, criteria for comparative analysis were established, and several logical components were implemented in Field Programmable Gate Array (FPGA) it was concluded that very low complexity projects are equally qualified.*

Resumo. *Com a complexidade dos circuitos integrados modernos, as linguagens de descrição de hardware (Hardware Description Language - HDL) se tornaram extremamente importantes como ferramentas no desenvolvimento e prototipagem de projetos eletrônicos. Elas têm como principal utilidade descrever o comportamento e a estrutura de um hardware. Dentre elas as mais conhecidas e utilizadas são Verilog e VHDL. E não há um consenso sobre qual dessas duas HDLs é a melhor, logo esse trabalho, a partir de vários critérios, e criação de circuitos básicos com o auxílio da ferramenta de síntese e simulação de HDLs, Quartus Prime, buscou compara-las. Então, foram estabelecidos critérios para a análise comparativa, sendo implementados diversos componentes lógicos em FPGA (Field Programmable Gate Array) e concluiu-se que projetos de complexidade muito baixa ambas se mostram igualmente qualificadas.*

Palavras chave: Verilog, VHDL, FPGA, Sistemas Digitais.

1.Introdução

Os circuitos eletrônicos se dividem em dois tipos: digitais e analógicos. Os circuitos digitais possuem diversas vantagens para um grande número de aplicações e por isso seus projetos são de grande importância [VAHID, 2008][TOCCI, 2011][PEDRONI, 2010]. Entre as ferramentas para os projetos de circuitos integrados (CIs), as linguagens de descrição de hardware são as que dão maior suporte para o projetista e permitem facilitar tanto as etapas de projeto como de manutenção e melhoramento dos CIs

[VAHID, 2008][RIESGO; TORROJA; LA TORRE, 1999]. Logo a escolha de uma HDL adequada é de grande importância no desenvolvimento de projetos.

Este artigo está organizado nessa ordem: introdução, logo após, na seção 2, são apresentadas as linguagens de descrição de Hardware enfatizadas neste trabalho. Na seção 3, é apresentada a metodologia utilizada para a análise. Na seção 4 são apresentados os resultados e discussões das comparações efetuadas. Finalmente, na seção 5 são apresentadas as conclusões do trabalho.

2. Linguagens de descrição de Hardware

Linguagem de Descrição de Hardware (Hardware Description Language - HDL) serve para modelar a estrutura e o comportamento de um hardware. As HDLs facilitam na descrição do hardware em dois importantes aspectos: o verdadeiro comportamento abstrato, podendo descrever o seu comportamento em vários níveis de abstração; o outro aspecto é a sua estrutura, pois é possível modelar uma estrutura de um hardware independente do comportamento do circuito [ORDONEZ et al, 2003]. Como Verilog e VHDL são as mais difundidas, esse trabalho foca nelas. Verilog, foi desenvolvido por Gateway Design Automation em 1983 como uma linguagem de modelagem de hardware para seus produtos simuladores. Foi desenvolvida a partir da linguagem C. A Cadence comprou o Verilog da Gateway Design Automation em 1989, então a linguagem ganhou muita popularidade. Em 1990 se tornou domínio público, desde então sofreu várias melhorias e em 1995 foi padronizada por IEEE como padrão IEEE 1364-1995. Depois disso, já houve diversos outros padrões e modificações do Verilog [R.UMA E R.SHARMILA, 2011][SMITH, 1996]. A linguagem VHDL (VHSIC Hardware Description Language, onde VHSIC quer dizer Very High Speed Itegrated Circuits) teve sua criação financiada pelo departamento de defesa dos Estados Unidos em 1980 [PEDRONI 2010]. A linguagem inicialmente era destinada a documentar o comportamento do hardware digital. Ela foi desenvolvida a partir das linguagens de programação ADA e Pascal [PEDRONI, 2010]. Passou por vários processos de incrementos ao longo dos anos até se tornar padrão 1076 do IEEE em 1987[PEDRONI, 2010]. Novas atualizações e melhorias desse padrão de 1076 IEEE foram criadas e novas continuam a serem desenvolvidas [R.UMA E R.SHARMILA, 2011][PEDRONI, 2010][PEDRONI, 2004].

3. Estado da Arte

Nos últimos anos, diversos autores vêm comparando linguagens de descrição de hardware, especialmente VHDL e Verilog. Maginot (1992) comparou VHDL com Verilog, UDL/I (novo padrão japonês), e M (da Mentor Graphics), visando avaliar o desempenho do VHDL frente à outras linguagens de hardware. Ele apresentou um estudo mostrando a potencialidade das linguagens de descrição de hardware na implementação de sistemas digitais, expondo as peculiaridades de algumas linguagens, vantagens e desvantagens, enfatizando o VHDL, onde concluiu que esta apresenta uma maior potencialidade para modelagens comportamentais de sistemas. Coumeri e Thomas (1994) desenvolveram um conjunto de padrões, ou melhor, variáveis de referência para

analisar a velocidade de simuladores de Verilog e VHDL. Smith (1996) fez um trabalho semelhante, contudo enfatizou as características de sintaxe das linguagens VHDL e Verilog, expondo as semelhanças e diferenças entre elas. Bailey (2005) compara Verilog, SystemVerilog e VHDL, enfatizando a utilização do System Verilog em substituição ao Verilog e VHDL. R.Sharmila (2011) apresentou um estudo comparativo entre a implementação de sistemas digitais com Verilog e VHDL analisando diversos aspectos de projeto como, conteúdo, estrutura, reutilização, portabilidade, custo, etc, com o intuito de verificar a existência de superioridade por parte de uma delas. Assim, pode-se afirmar que a comparação entre as linguagens Verilog e VHDL vem gerando diversas discussões no meio acadêmico e científico. E que não existem resultados suficientes que justifiquem a escolha de Verilog ou VHDL para descrição de hardware de uma forma geral, pois nesses trabalhos citados os autores só concluíram a superioridade de uma linguagem sobre a outra em algum aspecto especifico, ou aplicação, e os critérios utilizados na compactação não são os mesmos utilizados nesse trabalho. Para um maior aprofundamento sugere-se a verificação das referências citadas nesse tópico.

4. Metodologia

No ambiente de descrição Quartus Prime foram implementados, em ambas as HDLs, Verilog e VHDL, os códigos das portas lógicas AND, OR, NAND, NOT, XOR e XNOR. Assim como os códigos de memórias RAM (Random Access *Memory*) 16x4 (16 palavras com 4 bits), ROM (*Read-Only Memory*) 8x8 (8 palavras com 8 bits), registador de 4 bits. Um somador completo e uma máquina de estados do temporizador de um laser que cujo circuito descreve um *laser* que após ser iniciado por um sinal de entrada fica ligado durante 3 ciclos de *clock* voltando então para seu estado desligado, também possui uma entrada cujo sinal pode fazer a máquina do *laser* voltar ao estado de desligado [VAHID, 2008].

Os parâmetros utilizados para comparação entre as duas linguagens analisadas são: tempo de compilação, lógica utilizada, número de registadores utilizados, total de pinos utilizados, total de bits de bloco de memórias, total de PLL (*Phase Locked Look*) e analise de tempo (esse parâmetro não foi utilizado na análise das implementações dos códigos das portas lógicas simples). O dispositivo lógico programável escolhido para a análise de implementação, no Quartus Prime, foi um 5CGXFC7C7F23C8 da família Ciclone V.

4. Resultados e Discussões

Dada a simplicidade da porta AND não surpreende que no resultado não haja grandes diferenças entre as HDLs para a implementação de seu código. Como pode ser visto na Tabela 1, o único parâmetro em que os resultados apresentam diferenças entre VHDL e Verilog é no tempo de compilação, porém não foi uma diferença significativa, e pode ser atribuída a outros fatores como a ocupação do processador do computador durante as compilações.

Tabela 1 - Resultado do Código da Porta AND.

Porta AND	Verilog	VHDL
Logica Utilizada	1/56480 (<1%)	1/56480 (<1%)
Registradores	0	0
Total de pinos	3/268 (1%)	3/268 (1%)
Total de bits de blocos de memoria	0/7024640 (0%)	0/7024640 (0%)
Total PLLs	0/13 (0%)	0/13 (0%)
Tempo de Compilação (h:min:seg)	00:03:28	00:02:34

A porta OR é uma das portas lógicas básicas de circuitos digitais assim como a porta AND e possui também a mesma quantidade de entradas e saídas. Os resultados foram bem similares aos da porta AND, dado que não possuem grande complexidade. Segue abaixo seus resultados na Tabela 2.

Tabela 2 - Resultado do Código da Porta OR.

Porta OR	Verilog	VHDL
Logica Utilizada	1/56480 (<1%)	1/56480 (<1%)
Registradores	0	0
Total de pinos	3/268 (1%)	3/268 (1%)
Total de bits de blocos de memoria	0/7024640 (0%)	0/7024640 (0%)
Total PLLs	0/13 (0%)	0/13 (0%)
Tempo de Compilação (h:min:seg)	00:02:50	00:03:30

A porta NOT também é bastante simples, porém só possui uma entrada. Consequentemente a porta NOT utilizou menos pinos que a porta AND e OR na descrição em ambas as HDLs, quanto aos outros resultados foram similares. Seus resultados se encontram abaixo na tabela 3.

Tabela 3 – Resultado do Código da Porta NOT.

Porta NOT	Verilog	VHDL
Logica Utilizada	1/56480 (<1%)	1/56480 (<1%)
Registradores	0	0
Total de pinos	2/268 (1%)	2/268 (1%)
Total de bits de blocos de memoria	0/7024640 (0%)	0/7024640 (0%)
Total PLLs	0/13 (0%)	0/13 (0%)
Tempo de Compilação (h:min:seg)	00:02:54	00:02:44

Uma porta XOR é um pouco mais complexa que as partes que as outras portas já citadas. Em sua implementação a porta XOR não apresentou resultados muito distintos das outras portas lógicas, ambas as HDLs se mostraram com a mesma capacidade. O

parâmetro de tempo de compilação, assim como nos outros códigos, demostrou resultados diferentes para cada HDL, as possíveis causas disso já foram mencionadas anteriormente. Na Tabela 6 abaixo se encontra os resultados para porta XOR.

Tabela 4 - Resultado do Código da Porta XOR.

Porta XOR	Verilog	VHDL
Logica Utilizada	1/56480 (<1%)	1/56480 (<1%)
Registradores	0	0
Total de pinos	3/268 (1%)	3/268 (1%)
Total de bits de blocos de memoria	0/7024640 (0%)	0/7024640 (0%)
Total PLLs	0/13 (0%)	0/13 (0%)
Tempo de Compilação (h:min:seg)	00:02:55	00:02:19

Como pode ser notado vendo a Tabela 5 na próxima página (com os resultados da porta XNOR) e os resultados das outras portas lógicas, não houve grandes diferenças em seus resultados nos critérios de comparação. O único critério que ainda apresentou alguma diferença de uma HDL para outra foi o tempo de compilação, logo, a diferença de complexidade entre as portas lógicas não produziu resultados diferentes quanto à comparação das HDLs.

Tabela 5 - Resultado do Código da Porta XNOR.

Porta XNOR	Verilog	VHDL
Logica Utilizada	1/56480 (<1%)	1/56480 (<1%)
Registradores	0	0
Total de pinos	3/268 (1%)	3/268 (1%)
Total de bits de blocos de memoria	0/7024640 (0%)	0/7024640 (0%)
Total PLLs	0/13 (0%)	0/13 (0%)
Tempo de Compilação (h:min:seg)	00:03:22	00:02:17

De todos os códigos cujos resultados foram apresentados e discutidos aqui, o do somador completo é o primeiro em que é levado em conta o parâmetro de comparação das HDLs quanto a análise de tempo. Nesse critério se analisa quanto tempo de atraso tem as saídas do circuito em relação ao momento que sinais de entrada são inseridos. Nos códigos discutidos antes não se levava em conta esse parâmetro por serem mais simples e consequentemente não se esperava atrasos significativos ou notáveis. Em todos os próximos códigos, o parâmetro da análise de tempo será levado em consideração.

Na tabela 6 é apresentado, que para o somador completo não houve diferença significativa nos resultados entre uma HDL e outra, e nem mesmo atraso no circuito descrito em ambas as HDLs.

Tabela 6 - Resultado do Código de um Somador Completo.

Somador Completo	**Verilog**	**VHDL**
Logica Utilizada	2/56480 (<1%)	2/56480 (<1%)
Registradores	0	0
Total de pinos	5/268 (2%)	5/268 (2%)
Total de bits de blocos de memoria	0/7024640 (0%)	0/7024640 (0%)
Total PLLs	0/13 (0%)	0/13 (0%)
Tempo de Compilação (h:min:seg)	00:03:02	00:02:45
Atraso nas saídas (ns)	0	0

Registradores armazenam valores durante 1 ciclo de *clock*. O que foi implementado armazena 4 bits. Em relação aos outros códigos implementados (como pode ser visto nos resultados da Tabela 7), a sua complexidade é maior, necessita de registradores, utiliza mais pinos, porém também não se verificou diferença significativa entre Verilog e VHDL.

Tabela 7 - Resultado do Código de um Registrador.

Registrador 4 bits	Verilog	VHDL
Logica Utilizada	2/56480 (<1%)	2/56480 (<1%)
Registradores	4	4
Total de pinos	9/268 (3%)	9/268 (3%)
Total de bits de blocos de memoria	0/7024640 (0%)	0/7024640 (0%)
Total PLLs	0/13 (0%)	0/13 (0%)
Tempo de Compilação (h:min:seg)	00:03:45	00:04:05
Atraso na saída (ns)	0	0

Os resultados da memória ROM, que se encontram na Tabela 8 abaixo, até então são os que mostram maior uso de registradores. Na comparação entre Verilog e VHDL ela só fortalece a hipótese de que as duas linguagens possuem a mesma eficácia na descrição de circuitos.

Tabela 8 - Resultado do Código de uma memória ROM 8x8.

Memória ROM 8x8	**Verilog**	**VHDL**
Logica Utilizada	4/56480 (<1%)	4/56480 (<1%)
Registradores	7	7
Total de pinos	12/268 (3%)	12/268 (4%)
Total de bits de blocos de memoria	0/7024640 (0%)	0/7024640 (0%)
Total PLLs	0/13 (0%)	0/13 (0%)
Tempo de Compilação (h:min:seg)	00:02:19	00:02:32
Atraso na saída (ns)	0	0

Como mostra a Tabela 9, o código da memória RAM, apesar de apresentar resultados distintos dos outros códigos (pois é o único que usa blocos de memória e também o que usa mais pinos), de um modo geral na comparação das HDLs não apresenta nada novo e só reafirma o que a implementação dos outros códigos mostra.

Tabela 9 - Resultado do Código de uma memória RAM 16x8.

Memoria RAM 16x8	Verilog	VHDL
Logica Utilizada	1/56480 (<1%)	1/56480 (<1%)
Registradores	0	0
Total de pinos	22/268 (8%)	22/268 (8%)
Total de bits de blocos de memoria	128/7024640 (<1%)	128/7024640 (0%)
Total PLLs	0/13 (0%)	0/13 (0%)
Tempo de Compilação (h:min:seg)	00:04:47	00:03:08
Atraso na saída (ns)	0	0

A máquina de estados finitos do temporizador de um laser basicamente descreve o funcionamento de um laser que, como já foi mencionado, depois de ligado fica ativo durante 3 ciclos de *clock* e depois retorna ao seu estado inicial [VAHID, 2008]. Tal máquina de estados finitos pode ser vista na Figura 1 abaixo [VAHID 2008]. Cada seta ligando os estados representa a transição do *clock.* O b é o sinal de entrada que indica quando laser deve ser iniciado, x é o sinal de saída que aciona diretamente o laser.

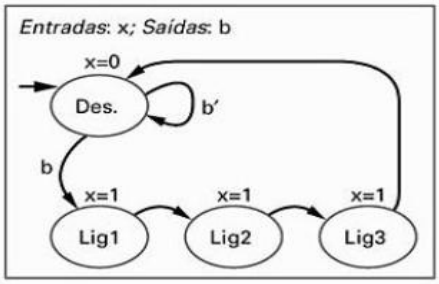

Figura 1: Máquina de Estados Finitos.

Os códigos do temporizador do laser foram os maiores descritos e os que apresentaram maior complexidade para o projetista, porém como é possível ver na Tabela 10 abaixo não houve diferenças significativas entre o código em Verilog em relação ao mesmo em VHDL, a única diferença é no tempo de compilação, porém tal diferença pode ser dada por outros fatores indiferentes as HDLs como já foi citado anteriormente.

Tabela 10 - Resultado do Código de um temporizador de laser.

Temporizador de laser	Verilog	VHDL
Logica Utilizada	2/56480 (<1%)	2/56480 (<1%)
Registradores	4	4
Total de pinos	4/268 (2%)	4/268 (2%)
Total de bits de blocos de memoria	0/7024640 (0%)	0/7024640 (0%)
Total PLLs	0/13 (0%)	0/13 (0%)
Tempo de Compilação (h:min:seg)	00:06:37	00:04:06
Atraso nas saídas (ns)	0	0

4. Conclusão

Analisando os códigos que foram implementados tanto em Verilog quanto em VHDL, ambas as HDLs se mostraram igualmente eficientes, surgindo poucos resultados diferentes entre elas. Consequentemente para projetos de nível de complexidade semelhante aos implementados aqui as duas HDLs se mostraram com o mesmo potencial baseado nos critérios utilizados. E a partir desse estudo ainda há a possibilidade de avaliar outros projetos e outros parâmetros de comparação de Verilog e VHDL para somar novos resultados.

5. AGRADECIMENTOS

Os autores gostariam de agradecer à Universidade Federal Rural do Semi-Árido (UFERSA) e ao Grupo de Desenvolvimento e Simulação (GDeS) da UFERSA pelo suporte ao longo do trabalho.

6. Referências

VAHID, Frank. **Sistemas Digitais:** Projeto, Otimização e HDLs. Porto Alegre: Artemed, 2008. 560 p.

TOCCI, R. J.; WIDMER, N. S.; MOSS, G. L. **Sistemas Digitais: princípios e aplicações.** 11. ed. São Paulo: Prentice Hall, 2011. 840p.

PEDRONI, Volei A.. **Eletrônica Digital Moderna e VHDL.** 5. ed. Rio de Janeiro: Elsevier, 2010. 621 p.

RIESGO, Teresa; TORROJA, Yago; LATORRE, Eduardo de. Design Methodologies Based on Hardware Description Languages. **Ieee Transactions On Industrial Electronics.** [s.l.], p. 3-12. fev. 1999.

ORDONEZ, Edward David Moreno et al. **Projeto, Desempenho e Aplicações de Sistemas Digitais em Circuitos Programáveis (FPGAs).** Pompéia, S.p: Bless Gráfica e Editora Ltda, 2003. 300 p.

R.UMA; R.SHARMILA. Qualitative Analysis of Hardware Description Languages: VHDL and Verilog. **International Journal Of Computer Science And Information Security.** [s.i], p. 127-135. abr. 2011. Disponível em: <https://sites.google.com/site/ijcsis/>. Acesso em: 28 jul. 2016.

SMITH, Douglas J.. VHDL and Verilog compared and contrasted-plus modeled example written in VHDL, Verilog and C. In: DESIGN AUTOMATION CONFERENCE PROCEEDINGS 1996, 33RD, 1., 1996, Las Vegas, Nv. **Design Automation Conference Proceedings 1996, 33rd.** Las Vegas, Nv: Ieee, 1996. p. 771 – 776.

PEDRONI, Volei A.. **Circuit Design with VHDL.** London: Mit Press, 2004.

MAGINOT, Serge. Evaluation criteria of HDLs: VHDL compared to Verilog, UDL/I and M. In: DESIGN AUTOMATION CONFERENCE, 1992., EURO-VHDL '92, EURO-DAC '92, 1992, Hamburg. **Design Automation Conference, 1992., EURO-VHDL '92, EURO-DAC '92.** Hamburg: Ieee, 1992. p. 746 - 751.

COUMERI, Sari L.; THOMAS, Donald E.. Benchmark descriptions for comparing the performance of Verilog and VHDL simulators. In: VERILOG HDL CONFERENCE, 1994., INTERNATIONAL, 1994, Santa Clara, Ca. **Verilog HDL Conference, 1994., International.** Santa Clara, Ca: Ieee, 1994. p. 37 - 42.

BAILEY, Stephen. **Comparison of VHDL, Verilog and SystemVerilog.** 2005. Disponível em: <https://www.mentor.com/products/fv/resources/overview/comparison-of-vhdl-verilog-and-systemverilog-6884d3c1-e7a3-4425-b9ee-08ad281f867d>. Acesso em: 28 jul. 2016.