

# SystemVerilog Vocabulary Extractor

Filipe C. Cavalcanti  
Leandro de S. Albuquerque  
Orientador: Tio Kat

5 de outubro de 2017

## 1 ABSTRACT

## 2 RESUMO

Desde a criação da primeira HDL até os presentes dias, cada vez mais o desenvolvimento de sistemas digitais se assemelha e aproxima-se a codificações de programas descritos em linguagem de programação

## 3 Introdução

Quando Verilog foi criado em meados da década de 80, o tamanho típicos dos projetos era na ordem de 5 a 10 mil portas lógicas, o método de concepção dos circuitos era usando esquema gráfico, e a simulação estava começando a ser uma ferramenta essencial para verificação [13]. A linguagem Verilog continuou a evoluir com a tecnologia de *design* e verificação até que, em 2002 surge SystemVerilog, sendo esta linguagem uma significativa melhoria de Verilog [13].

A partir disto, como a complexidade de sistemas digitais modernos continua a aumentar exponencialmente, tem-se que as metodologias de projetos RTL(Register Transfer Level) estão evoluindo também [8] e [4].

Com tal avanço, elevou-se o nível de abstração no desenvolvimento de hardware por meio de uma linguagem de descrição e verificação de hardware (HDVL), de tal forma que, o uso de ferramentas de análise de informações que antes eram somente do es-

copo da engenharia de software, poderá contribuir também para o desenvolvimento de sistemas digitais.

Nos últimos anos, SystemVerilog e SystemC estão sendo extensivamente usadas para descrição e verificação na indústria VLSI(Very Large Scale Integration) [7]. Sendo que utilizou-se neste trabalho SystemVerilog, que é uma unificação entre, descrição de hardware e linguagem de verificação [6], permitindo o uso de uma linguagem unificada para especificações abstratas e detalhadas para descrição e verificação.

Umas das principais fontes de informações em um código fonte é o vocabulário. Denominado de léxico do código em [5], [2] e [11], consiste no conjunto de termos repetidos ou únicos que compõem identificadores e que estão presentes no textos dos comentários [1].

Usando os princípios da engenharia reversa como uma coleção de metodologias e técnicas capazes de realizar a extração e abstração de informações [3] e [11], propõe-se neste trabalho a extração do vocabulário pertencentes a projetos de hardware descritos em SystemVerilog e usando a definição formal de Santos em [12] sobre vocabulário de software para embasar e fundamentar o termo *Hardware Vocabulary*.

## 4 Background

Graças aos atuais projetos eletrônicos baseado em HDL, metodologias e ferramentas

para simulação, síntese, verificação, modelagem física e teste pós-fabricação agora estão bem inseridos e são essenciais para designers digitais [10]. Nos últimos anos as linguagens de descrição e verificação de hardware tornaram-se tão importantes para a modelagem de sistemas digitais, quanto as linguagens de programação o são para a engenharia de software.

## 4.1 Software Vocabulary

Santos em [11], define informalmente que vocabulário de código fonte compreende as cadeias de caracteres que identificam os elementos estruturais e as palavras que compõem as sentenças dos comentários de um código fonte. Dentro dos campos de estudo da engenharia de software outro termo bastante conhecido e isomorfo a *Software Vocabulary* é o *léxico do código* que em [1] e [2] são os elementos que nomeiam as entidades estruturais da linguagem além dos comentários escritos em linguagem natural.

*Software Vocabulary* é um multiconjunto de *Strings*, i.e. uma aplicação  $V : \mathbb{S} \rightarrow \mathbb{N}$ , que mapeia *Strings* para números naturais. Elementos de um vocabulário são chamados *termos*. Para qualquer termo  $t$ ,  $V(t)$  representa o número de ocorrências do termo  $t$  no vocabulário  $V$ . Se  $V(t) > 0$  dizemos que  $t$  é um termo do vocabulário [12].

No paradigma de programação dominante atualmente OOP (*Object-Oriented Programming*), nomear os elementos estruturais da linguagem de forma concisa e representativa além de documentá-las, tem sido mais do que uma boa prática .

No desenvolvimento de grandes sistemas o léxico ou SV (*Software Vocabulary*), quando condizente ao problema, reduz o tempo de manutenção, entendimento do código e encontro de *bugs*.

O léxico de um programa representa um investimento substancial para uma empresa de software, portanto, sua importância deve preservada e elevada ao longo do tempo, para aproveitar ao máximo seus efeitos e

benéficos na compreensão do programa [2].

## 4.2 O Hardware Como Um Software

## 4.3 O Que é Uma HDL?

Uma descrição HDL (*Hardware Description Language*) é uma representação precisa que pode ser usada para documentar, comunicar e simular o projeto [9].

As HDLs modernas são fundamentais para o desenvolvimento de sistemas digitais, possibilitando suas descrições de forma estrutural, comportamental e nos últimos anos, seguindo conceitos básicos de orientação a objetos, fornecendo assim um mecanismo efetivo para o desenvolvimento de projetos à medida que evoluem da abstração para a realidade.

## 4.4 Hardware Vocabulary

# 5 SystemVerilog Vocabulary Extractor

Afim de calcularmos uma porcentagem de extração que melhor represente a eficiência do software proposto, foi elaborado um design genérico com todas as estruturas possíveis em SystemVerilog. Os resultados obtidos são apresentados na tabela abaixo:

Tabela 1: Hello Word Table

posição	País	IDH
1	Noruega	.955
2	Austrália	.938
3	EUA	.937
4	Holanda	.921
5	Alemanha	.920

Os resultados apresentados na tabela acima mostra que ...

Foram realizados, também, outros testes com hardware *opensource* obtidos em repositórios no Github. Os resultados obtidos estão expostos na tabela abaixo:

## 6 Resultados e Discussões

Tabela 2: Hello Word Table

posição	País	IDH
1	Noruega	.955
2	Austrália	.938
3	EUA	.937
4	Holanda	.921
5	Alemanha	.920

## Referências

- [1] S. L. Abebe, S. Haiduc, A. Marcus, P. Tonella, and G. Antoniol. Analyzing the evolution of the source code vocabulary. *Proceedings of the European Conference on Software Maintenance and Reengineering, CSMR*, pages 189–198, 2009.
- [2] G. Antoniol, Y. G. Guéhéneuc, E. Merlo, and P. Tonella. Mining the lexicon used by programmers during software evolution. *IEEE International Conference on Software Maintenance, ICSM*, pages 14–23, 2007.
- [3] P. Benedusi, A. Cimitile, and U. D. Carlini. Reverse engineering processes, design document production, and structure charts. *Journal of Systems and Software*, 19(3):225 – 245, 1992.
- [4] V. Hahanov, D. Melnik, O. Zaharchenko, and S. Zaychenko. Overview of Object-Oriented Approach to HDL-Testbench Construction for System-on-Chips. pages 621–625, 2008.
- [5] E. W. Høst and B. M. Østvold. The programmer’s lexicon, volume I: The verbs. *SCAM 2007 - Proceedings 7th IEEE International Working Conference on Source Code Analysis and Manipulation*, I:193–202, 2007.
- [6] IEEE Computer Society and IEEE Standards Association Corporate Advisory Group. IEEE Standard for SystemVerilog–Unified Hardware Design, Specification, and Verification Language. 2012(February):1315, 2013.
- [7] P. Kumar. High Level Modeling Of Physical Layer Noise Parameters Using SystemC. pages 344–347, 2014.
- [8] J. D. M. DAIGNEAULT. RAISING THE ABSTRACTION LEVEL OF HDL FOR CONTROL-DOMINANT APPLICATIONS Marc-Andre Daigneault and Jean Pierre David Department of Electrical Engineering , Ecole Polytechnique de Montreal. pages 515–518, 2012.
- [9] D. L. Miller-Karlow and E. J. Golin. vVHDL: A Visual Hardware Description Language.
- [10] Z. Navabi. HDLs Evolve as they Affect Design Methodology for a Higher Abstraction and a Better Integration. page 4799, 2015.
- [11] K. D. F. Santos. Webservice De Extração De Vocabulário De Código Para Pesquisas Empíricas Em Engenharia De Software. 2009.
- [12] K. D. F. Santos, D. D. S. Guerrero, and J. C. A. D. Figueiredo. Using Developers Contributions on Software Vocabularies to Identify Experts. *Proceedings - 12th International Conference on Information Technology: New Generations, ITNG 2015*, pages 451–456, 2015.

- [13] S. Sutherland, S. Davidmann, and P. Flake. *SystemVerilog for Design Second Edition: A Guide to Using SystemVerilog for Hardware Design and Modeling*. 2006.