

# SystemVerilog Vocabulary Extractor

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

October 4, 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 [12]. A linguagem Verilog continuou a evoluir com a tecnologia de design e verificação ate que, em 2002 surge SystemVerilog, sendo esta linguagem uma significativa melhoria de Verilog [12]. A partir disto, Como a complexidade de sistemas digitais modernos continua a aumentar exponencialmente, tem-se que as metodologias de design RTL estão crescendo também [9] e [5].

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 escopo da engenharia de software, poderá con-

tribuir também para o desenvolvimento de sistemas digitais.

Nos últimos anos, SystemVerilog e SystemC estão sendo extensivamente usadas para design e verificação na industria VLSI (Very Large Scale Integration) [8]. Sendo nosso foco SystemVerilog, que é uma unificação entre, design de hardware, e linguagem de verificação [7], SystemVerilog permite o uso de uma linguagem unificada para especificações abstratas e detalhadas do design e verificação, também em [7].

Umas das principais fontes de informações em um código fonte é o vocabulário do mesmo. O vocabulário também denominado de léxico do código em [6] e [2], 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], propõe-se neste trabalho uma ferramenta que possibilita a extração de vocabulário para SystemVerilog, além de fundamentar o termo *Vocabulário de Hardware*.

## 4 Background

Graças aos atuais designs 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 O Que é Uma HDVL?

HDVL (Hardware Description and Verification Language), podemos abstrair como um único ambiente para design e verificação de sistemas digitais, em [4] uma HDVL representa hardware digital em vários níveis de abstração.

#### 4.2 O Hardware Como Um Software

#### 4.3 Vocabulário de Software

Santos, em [11] define 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. No paradigma de programação dominante atualmente OOP (Object-Oriented Programming),

### 5 SystemVerilog Vocabulary Extractor

### 6 Resultados e Discussões

### References

- [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] P. Flake. Why SystemVerilog ?
- [5] 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.
- [6] 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.
- [7] 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.
- [8] P. Kumar. High Level Modeling Of Physical Layer Noise Parameters Using SystemC. pages 344–347, 2014.
- [9] 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.

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