

SystemVerilog Vocabulary Extractor

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

6 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

Verilog foi uma das primeiras linguagens para descrição de hardware (HDL) a ser inventada, em meados da década de 80. O tamanho típico dos projetos era entre de 5 a 10 mil portas lógicas. O método de concepção dos circuitos utilizava-se de esquema gráfico, e a simulação começava a ser uma ferramenta essencial para verificação [10]. Com a evolução da tecnologia de descrição e verificação de hardware, em 2002 surge SystemVerilog.

A partir disto, como a complexidade de sistemas digitais modernos aumentou exponencialmente, tanto que, o tamanho dos atuais projetos chega a ordem dos bilhões de portas lógicas. Assim as metodologias de projetos em sistemas digitais estão evoluindo extensivamente [5] 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, pode ser estendido também para o desenvolvimento de sistemas digitais.

Um das principais fontes de informações em um código fonte no âmbito da engenharia de software, é o vocabulário. Dentre suas principais utilidades listamos o seguinte:

1. Localização de bugs;
2. Identificação de uma arquitetura;
3. Métricas sobre o código fonte;
4. Identificação de especialista [9].

O vocabulário de software também denominado de léxico do código em [3], 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 [8], propõe-se neste trabalho a extração do vocabulário pertencentes a projetos de hardware descritos em SystemVerilog, e usando a definição formal de Santos em [9] 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 [7]. 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 [8], 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. 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 [3] 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 [9].

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, facilita o 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 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 [6].

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.

4.3 O Hardware Como Um Software

A evolução no processo de desenvolvimento de software atrela-se ao fato do constante avanço dos paradigmas de programação, desde a programação procedural até a orientação a objetos. De forma análoga, a evolução do processo de desenvolvimento de hardware, atrela-se também ao constante avanço dos métodos em desenvolver sistemas digitais. Indo desde esquemas gráficos à descrição baseada em orientação a objetos e verificação UVM (*Universal Verification Methodology*).

A aproximação (similaridade no processo de desenvolvimento) do software ao Hardware surge da criação das HDLs. Neste ponto da evolução tecnológica, tanto hardware quanto software eram desenvolvidos por meio de uma representação textual unidimensional, seguindo uma determinada sintaxe e semântica. Em [4] classes e módulos são coisas bastante semelhantes, trazendo assim a possibilidade de usar classes dentro de um contexto de verificação de hardware.

4.4 Hardware Vocabulary

Dada uma linguagem de descrição e (ou) verificação de hardware temos que;
As sequências de caracteres que nomeiam as entidades estruturais assim como os blocos de comentários são chamados de *Vocabulário de Hardware*, de forma que .

Sendo S o conjunto de todo o código fonte, e que cada elemento deste conjunto seja α ,

temos que o vocabulário de hardware é definido como a imagem da função $F(\alpha)$, onde $im(F) = \mathbb{H}\mathbb{V}$ é o agrupamento dos dados na forma real da modelagem destes dados. Assim $F : \mathbb{S} \rightarrow \mathbb{H}\mathbb{V}$

5 SystemVerilog Vocabulary Extractor

A fim de analisar e extrair o vocabulário de hardware pertencentes a projetos descritos em SystemVerilog desenvolvemos o ferramenta *Hardware Vocabulary Tool*, e [?]

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] L. R. Biggers, B. P. Eddy, N. A. Kraft, and L. H. Etzkorn. Toward a metrics suite for source code lexicons. *IEEE International Conference on Software Maintenance, ICSM*, pages 492–495, 2011.
- [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] 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.
- [6] D. L. Miller-Karlow and E. J. Golin. vVHDL: A Visual Hardware Description Language.
- [7] Z. Navabi. HDLs Evolve as they Affect Design Methodology for a Higher Abstraction and a Better Integration. page 4799, 2015.
- [8] K. D. F. Santos. Webservice De Extração De Vocabulário De Código Para Pesquisas Empíricas Em Engenharia De Software. 2009.

- [9] 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.
- [10] S. Sutherland, S. Davidmann, and P. Flake. *SystemVerilog for Design Second Edition: A Guide to Using SystemVerilog for Hardware Design and Modeling*. 2006.