Extraindo Vocabulário de Sistemas Digitais

Katyusco de F. Santos
Federal Institute of Paraba
Campus Campina Grande
Email: katyusco.santos@ifpb.edu.br

Filipe C. Cavalcanti
and Leandro de S. Albuquerque
Federal Institute of Paraba
Campus Campina Grande
Email: {filipe.cazuza, leandro.albuquerue }@academico.ifpb.edu.br

***Resumo*—Desde a criao das primeiras HDLs(Hardware Description Language) at os presentes dias, cada vez mais o desenvolvimento de sistemas digitais se assemelha ao desenvolvimento de sistemas de software. Como a complexidade no desenvolvimento de sistemas digitais aumentou exponencialmente, as metodologias em desenvolvimento de tais sistemas tambm evolui extensivamente.**

O processo de desenvolvimento de sistemas digitais desde o surgimento das HDLs, usa de conceitos e paradigmas da engenharia de software, indo da programao procedural at OOP(Object-Oriented Programming).

O vocabulrio de software consiste no conjunto de termos repetidos ou nicos que compem identificadores e que esto presentes no textos dos comentrios. Sendo o vocabulrio ou lxico, uma das principais fontes de informao no mbito da engenharia de software.

Neste trabalho demonstramos a similaridade entre HDLs e linguagens de programao, definimos o termo *Hardware Vocabulary* baseado na definio de *Vocabulario de Software*, alm de extrair o *Vocabulrio de Hardware* pertencente projetos descritos em SystemVerilog.

***Keywords*—HDL, SystemVerilog,Software Vocabulary, Digital Systems, Hardware Vocabulary.**

I. Introduction

When the Verilog hardware description language was create in the mid-1980s, the typical design size was of the order of five to ten thousand gates. O método de concepço dos circuitos utilizava-se de esquema gráfico, e a simulação comeava a ser ferramenta essencial para verificação[1]. Com a evolução da tecnologia de descrição e verificação de hardware, em 2002 surge significante melhoria de Verilog que é SystemVerilog [1].

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. As metodologias de projetos em sistemas digitais estã evoluindo extensivamente [2] e [3].

Com tal avano, 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, pode ser estendido também para o desenvolvimento de sistemas digitais.

Umas 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:

- Localização de bugs;
- Identificação de uma arquitetura;
- Métricas sobre o código fonte;
- Identificao de especialista [4].

O vocabulário de software também denominado de léxico do código em [5], consiste no conjunto de termos repetidos ou únicos que compõem identificadores e que estão presentes no textos dos comentários[6].

Usando os princípios da engenharia reversa como uma coleção de metodologias e técnicas capazes de realizar a extração e abstração de informações[7], propõe-se neste trabalho apresentar a similaridade entre HDLs e linguagens de programação, usar a definição formal de Santos em [4] sobre vocabulário de software para embasar e fundamentar o termo *Hardware Vocabulary*, e extrair vocabulário pertencentes a projetos de hardware descritos em SystemVerilog.

II. Background

Graas aos atuais projetos eletrnicos baseado em HDL, metodologias e ferramentas para simulao, sntese, verificao, modelagem fsica e teste ps-fabricao agora esto bem inseridos e so essenciais para designers digitais [8].

Nos ltimos anos as linguagens de descrio e verificao de hardware tornaram-se to importantes para a modelagem de sistemas digitais, quanto as linguagens de programao o so para a engenharia de software.

A. Software Vocabulary

Santos em [7], define que vocabulrio de software compreende as cadeias de caracteres que identificam os elementos estruturais e as palavras que compem as sentenas dos comentrios de um cdigo fonte.

Dentro dos campos de estudo da engenharia de software outro termo bastante conhecido e isomorfo a *Software Vocabulary* o *lxico do cdigo* que em [5] so os elementos que nomeiam as entidades estruturais da linguagem alm dos comentrios escritos em linguagem natural.

Software Vocabulary um multiconjunto de *Strings*, i.e., uma aplicao $V : \mathbb{S} \rightarrow \mathbb{N}$, que mapeia *Strings* para nmeros naturais. Elementos de um vocabulrio so chamados *termos*. Para qualquer termo t, $V(t)$ representa o nmero de ocorrncias do termo t no vocabulrio V. Se $V(t) > 0$ dizemos que t um termo do vocabulrio [4].

No paradigma de programao dominante atualmente OOP(*Object-Oriented Programming*), nomear os elementos

estruturais da linguagem de forma concisa e representativa alm de document-las, tem sido mais do que uma boa prtica.

No desenvolvimento de grandes sistemas de software o vocabulrio ou lxico, quando condizente ao problema, reduz o tempo de manuteno, facilita o entendimento do cdigo e encontro de *bugs*.

O lxico de um programa representa um investimento substancial para uma empresa de software, portanto, sua importncia deve preservada e elevada ao longo do tempo, para aproveitar ao mximo seus efeitos e benficos na compreenso do programa [9].

B. Hardware Description Language (HDL)

Uma HDL(*Hardware Description Language*) descreve o funcionamento do circuito, a sua concepo e organizao e ainda o testa para verificar seu funcionamento por meio de simulao[10].

As HDLs modernas so fundamentais para o desenvolvimento de sistemas digitais, possibilitando suas descries de forma estrutural, comportamental e nos ltimos anos, seguindo conceitos bsicos de orientao a objetos [3], fornecendo assim um mecanismo efetivo para o desenvolvimento de projetos medida que evoluem da abstrao para a realidade [8].

Algumas HDls evoluram de tal forma que, alm de descrever hardware, passaram tambm a englobar todo um ambiente de verificao. Tais linguagens foram tipificadas de HDVL (Hardware Description and Verification Language).

O escopo deste trabalho limita-se anlise de informaes somente para projetos descritos em SystemVerilog. A escolha dessa linguagem deve-se ao seu grande uso na industria VLSI (*Very Large Scale Integration* [11] e tambm por SystemVerilog ser uma HDVL [12], ampliando assim nosso horizonte de informaes, uma vez que temos dados tanto sobre descrio quanto verificao de hardware.

C. Entidades Em SystemVerilog

Na engenharia de software, uma entidade é um elemento do código fonte como uma classe ou um método[5], ou seja, são elementos estruturais que compõem a linguagem, sendo que, algumas entidades são formadas por outras subentidades, dando assim, origem a uma hierarquia, e uma maior modularidade no código fonte.

Há em SystemVerilog um grande numero de entidades (elementos estruturais que compõem a linguagem), uma vez que, esta linguagem engloba todos os seis dominions de uma HDVL [13], que são:

- Netlist;
- Register Transfer;
- General Programming;
- Testbench;
- Temporal Properties;
- Functional Coverage.

Tabela I. ENTIDADES DE SYSTEMVERILOG

Structural Element	Entity keyword	Element Composition
module	module, endmodule	function, task, union, struct, enum
package	package, endpackage	function, task, union, struct, enum
function	function, endfunction	union, struct, enum
task	task, endtask	union, struct, enum
class	class, endclass	function, task, union, struct, covergroup, enum
interface	interface, endinterface	function, task, modport, union, enum
modport	modport	
covergroup	covergroup	converpoint
coverpoint	coverpoint	bin
union	union	
enum	enum	
struct	struct	

D. O Hardware Como Um Software

A evoluo no processo de desenvolvimento de software atrela-se ao fato do constante avano dos paradigmas de programao, desde a programao procedural at a orientao a objetos.

De forma anloga, a evoluo do processo de desenvolvimento de hardware, atrela-se tambm ao constante avano dos mtodos em desenvolver sistemas digitais. Indo desde esquemas grficos descrio baseada em orientao a objetos e verificao UVM (*Universal Verification Methodology*) [3].

A aproximao (similaridade no processo de desenvolvimento) do software ao hardware surge da criao das HDLs. Neste ponto da evoluo tecnolgica, tanto hardware quanto software eram desenvolvidos por meio de uma representao textual unidemensional, seguindo uma determinada sintaxe e semntica.

Em [3] classes e mdulos so coisas bastantes semelhantes, trazendo assim a possibilidade de usar classes dentro de um contexto de verificao de hardware. A partir desta similaridade, o inverso tambm possvel, ou seja, utilizar de conceitos e convenes de orientao a objetos na descrio de mdulos.

Uma demonstrao da similaridade entre as linguagens de programao e as HDLs lista-se abaixo, em que, dado as principais caracteristicas de algumas linguagens de alto nivel, verifica se em SystemVerilog tambm h tais propriedades.

Tabela II. COMPARISON OF MODULES AND TRADITIONAL CLASSES

Feature	C++/Java	SystemVerilog
hereanç	Yes	Yes
objetos Dinamicos	Yes	Yes
Encapsulamento	Yes	Yes
polimorfismo	Yes	Yes
Classes Abstratas	Yes	Yes
hierarquia	Yes	yes

E. Hardware Vocabulary

Baseado fortemente pela definio formal de vocabulrio de software em [4], e embasado tambm por todo o levantamento feito sobre as semelhanas entre os processos de desenvolvimento entre hardware e software feitas na sesso anterior. Chegamos a primeira definio sobre vocabulrio de hardware:

(1) Dada uma linguagem de descrio e (ou) verificao de hardware, temos que as sequencias de caracteres que nomeiam

as entidades estruturais, assim como os blocos de comentrios so chamados de *Vocabulrio de Hardware*.

Complementando tal definio, o vocabulrio de hardware refere-se tambm a como estes dados esto organizados, ou seja, a estruturao da informao refletindo as entidades a qual o lxico pertence, de modo que s com o vocabulrio possa ser possvel criar uma entidade complementar a original (entidade que seu vocabulrio foi extrado).

Notando tal semelhana a definio de vocabulrio de software e sabendo de seu isomorfismo com lxico do cdigo(definio dada por Biggers em [5]), chegamos a concluso que (1) tambm pode ser chamado de *lxico do hardware*.

Uma exemplificao dada a partir de um trecho de cdigo mostrado na *Listing 1*, onde seu vocabulrio extrado, est numa representao VXL(Vocabulary XML).

Listing 1: Execerpt of code SystemVerilog

```
/* the module nextAddress returns the next address of the 2-bit offset
program */
module nextAddress(
    input logic[7:0] regCounter,
    input logic[7:0] regAddress,
    output logic[7:0] regPC);
  always_comb begin
    regPC = (regCounter + regAddress) << 2;
  end
endmodule: nextAddress
```

Listing 2: Extracted Vocabulary of Listing 1

```
<mdl name= "nextAddress">
  <prm>
    <fld type="input logic[7:0]" name=
    "regCounter"/>
    <fld type="input logic[7:0]" name=
    "regAddress"/>
    <fld type="output logic[7:0]" name= "regPC"/>
  </prm>
  <cmt cmm="the module nextAddress returns the next address of the 2-bit
offset program"/>
</mdl>
```

Como pode ser notado no resultado da extrao do vocabulrio, a estruturao dos dados segue fielmente a modelagem original do problema, assim sendo a partir do vocabulrio pode-se recriar uma entidade complementar a original.

Notamos que a definio formal de Santos em [4] sobre vocabulrio de software, aplica-se igualmente ao vocabulrio de hardware, assim chegamos a definio formal de vocabulrio de hardware:

(2) *Hardware Vocabulary definition*: Definimos *Hardware Vocabulary* como um multiconjunto de *Strings*, isto , uma aplicao $V : \mathbb{S} \rightarrow \mathbb{N}$, que mapeia *Strings* para nmeros naturais. Elementos de um vocabulrio so chamados *termos*. Para qualquer termo t, $V(t)$ representa o nmero de ocorrncias do termo t no vocabulrio V. Se $V(t) > 0$ dizemos que t um termo do vocabulrio.

Como um exemplo considere o trecho de cdigo em *Listing 1*. De acordo com nossa definio o *Vocabulrio de Hardware* pode ser expresso como o seguinte multconjunto de termos:

$$V = 2'regCounter + 2'regAddress + 2'regPC + 2'nextAddress$$

III. SystemVerilog Vocabulary Extractor

Para extrair o Vocabulario de hardware pertencente a projetos descritos em SystemVerilog desenvolvemos o ferramental *SystemVerilog Vocabulary Extractor tool* . A implementao desta ferramenta, baseia-se em entidades que so capazes de processar o vocabulrio de hardware dos principais elementos estruturais bsicos de SystemVerilog. Usando de delegao montam-se as estruturas mais complexas da linguagem (classes, interfaces, mdulos e pacotes) Ex.: Uma classe SystemVerilog uma entidade que possui um nome e compem-se principalmente de atributos, funes, tasks, comentrios.

Para posterior anlise, o resultado da extrao salvo em arquivo .xml seguindo a estruturao da *Listing 2*, contendo a extrao de todas as entidades estruturais da linguagem, sequindo a hierarquia do projeto.

Afim de validarmos a ferramenta calculamos uma porcentagem de extraco que melhor represente a eficiłncia do software proposto, elaborando um design genrico com todas as estruturas possveis da linguagem SistemVerilog. Os resultados obtidos so apresentados na tabela abaixo:

Tabela III. Porcentagem de extrao das principais entidades de SystemVerilog.

Entidade	Porcentagem de extrao
module	90%
class	98%
interface	90%
function	100%
task	90%
fields	100%

Os resultados apresentados na tabela acima mostram que, calculando uma mdia aritmtica, ou seja, deduzindo que o vocabulrio de hardware pertencente a cada entidade aproximadamente igual, o percentual de extrao total atualmente de 94% dado um cdigo genrico.

Extraindo o vocabulario pertencentes a projetos de hardware *opensource*, descritos em SystemVerilog desenvolvidos por terceiros em repositrios pblicos no site *https://github.com*, afim de aferir o Tempo de Execuo da extrao (TE) em milissegundo (ms), Quantidade de Linhas de código (LoC, *Lines of Code*), o consumo mdio de memória medido em Bytes(MEM) e por fim, o Tamanho do Arquivo .XML gerado a partir da extrao do vocabulário(TA). Os resultados obtidos esto expostos na tabela 2.

IV. Conclusion

Os resultados das extraes do vocabulrio de hardware dos projetos listados na tabela 2 encontram-se disponveis para download no repositrio GitHub `https://github.com/systemExtractor/SystemVerilogVocabularyExtractor`.

Cada Desenvolvedor tem seu vocabulrio, que a acumulao de termos manipulados (adicionados, removidos e substitudos)

Tabela IV. EXTRAO DO *Vocabulrio de Hardware* DE PROJETOS *opensource*.

Nome do Projeto	TP(Bytes)	LoC	TE (ms)
ahb_apb_bridge_uvm_tb https://github.com/mayur13/UVM/tree/master/projects/ahb_apb_bridge_uvm_tb	122.880	997	260
apb2_uvm_tb https://github.com/mayur13/UVM/tree/master/projects/apb2_uvm_tb	147.456	806	278
fpga_fast_serial_sort https://github.com/Poofjunior/fpga_fast_serial_sort	49.152	351	205
System-Verilog https://github.com/zricethezav/System-Verilog	49.221.632	4420	970
sha3_sv_tb https://github.com/mayur13/SystemVerilog/tree/master/projects/sha3_sv_tb	65.536	242	297
sha3_uvm_tb https://github.com/mayur13/UVM/tree/master/projects/sha3_uvm_tb	176.128	925	278
uvm-tutorial-for-candy-lovers https://github.com/cluelogic/uvm-tutorial-for-candy-lovers	700.416	11.559	1.877
spi_uvm_tb https://github.com/mayur13/UVM/tree/master/projects/spi_uvm_tb	139.264	172	177
uvm-utest https://github.com/nosnhojn/uvm-utest	311.296	5281	1.379
zynq-sandbox https://github.com/swetland/zynq-sandbox	450.560	8671	546

durante suas contribuies no projeto [4]. Isso equivale desenvolvimento de hardware, uma vez citada a similaridade entre as HDLs e as linguagens de programao.

Com a definio de *Vocabulrio de Hardware* formalizada, abrimos ento um novo campo de estudos para a engenharia de software. A anlise e extrao de informao pertencentes ao vocabulrio de projetos de hardware descritos por alguma HDL.

Levantada todas as semelhanas do processo de desenvolvimento de hardware ao software, conclumos que toda analise, extrao, mtricas e concluses feitas com base no vocabulrio de software so equivalentes quando aplicadas ao Vocabulrio de Hardware.

Ento com o *Vocabulrio de Hardware* possvel encontrar *bugs*, identificar uma arquitetura, fazer mtricas sobre o cdigo e identificar especialista. Com todo o arcabouo de informaes e teorias que a engenharia de software desenvolveu ao longo dos anos baseando-se em lxico e vocabulrio de software.

REFERÊNCIAS

[1] S. Sutherland, S. Davidmann, and P. Flake, *SystemVerilog for Design Second Edition: A Guide to Using SystemVerilog for Hardware Design and Modeling*, 2006. [Online]. Available: http://www.amazon.com/SystemVerilog-Design-Second-Hardware-Modeling/dp/0387333991

[2] 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," pp. 515–518, 2012.

[3] V. Hahanov, D. Melnik, O. Zaharchenko, and S. Zaychenko, "Overview of Object-Oriented Approach to HDL- Testbench Construction for System-on-Chips," pp. 621–625, 2008.

[4] 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*, pp. 451–456, 2015.

[5] 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*, pp. 492–495, 2011.

[6] 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*, pp. 189–198, 2009.

[7] K. D. F. Santos, "Webservice De Extração De Vocabulário De Código Para Pesquisas Empíricas Em Engenharia De Software," 2009.

[8] Z. Navabi, "HDLs Evolve as they Affect Design Methodology for a Higher Abstraction and a Better Integration," p. 4799, 2015.

[9] G. Antoniol, Y. G. Guéhéneuc, E. Merlo, and P. Tonella, "Mining the lexicon used by programmers during sofware evolution," *IEEE International Conference on Software Maintenance, ICSM*, pp. 14–23, 2007.

[10] D. L. Miller-Karlow and E. J. Golin, "vVHDL: A Visual Hardware Description Language."

[11] P. Kumar, "High Level Modeling Of Physical Layer Noise Parameters Using SystemC," pp. 344–347, 2014.

[12] IEEE Computer Society and IEEE Standards Association Corporate Advisory Group, "IEEE Standard for SystemVerilogUnified Hardware Design, Specification, and Verification Language," vol. 2012, no. February, p. 1315, 2013.

[13] P. Flake, "Why SystemVerilog ?" pp. 1–6, 2013.