

Índice

Índice de Figuras	x
Índice de Tabelas	xi
Lista de Algoritmos	xii
Abreviações	xiii
1 Introdução	1
1.1 Descrição do Problema	2
1.2 Objetivos	2
1.3 Descrição da Solução	2
1.4 Contribuições	2
1.5 Organização da Dissertação	2
Referências Bibliográficas	3
A Publicações	4
A.1 Referente à Pesquisa	4
A.2 Contribuições em outras Pesquisas	4

Índice de Figuras

Índice de Tabelas

Lista de Algoritmos

Capítulo 1

Introdução

Um sistema embarcado é um sistema computacional que tem o objetivo de controlar uma função ou um conjunto de funções em um sistema maior, elétrico ou mecânico [1]. As linguagens C e C++ são duas das principais linguagens de programação para desenvolvimento desses sistemas [2]. Por sua vez, os sistemas embarcados são utilizados em uma gama de aplicações, como controles automotivos e sistemas de entretenimento, de tal forma que os sistemas embarcados vêm se tornando indispensáveis para a vida do homem moderno [2].

Devido a importância dos sistemas embarcados é que se mostra evidente a importância de garantir a robustez na execução dos mesmos. A partir dessa preocupação, nota-se o crescimento da área de testes no processo de desenvolvimento desses sistemas.

Atualmente, Revisão por Pares e Teste de Programas são as duas principais técnicas para teste de programas [3]. Na Revisão por Pares, uma equipe de engenheiros experientes inspeciona o programa e, preferencialmente, não se envolve com o desenvolvimento do programa que está sendo revisado. Estudos empíricos mostram que a técnica é capaz de encontrar entre 31% a 93% dos defeitos, com uma média de 60% [3]. No teste de programas, a busca de defeitos é feita através de extensivas baterias de testes, que forçam o programa a executar nas mais diversas situações, seguida da análise dos resultados desses testes. Nessa abordagem, cerca de 70% do tempo de desenvolvimento é concentrado em encontrar e corrigir falhas no programa [4].

Existe porém, uma técnica que vem crescentemente sendo utilizada na verificação de programas, chamada verificação formal, que é uma técnica baseada em formalismos matemáticos para especificação, projeto e verificação de *software* e *hardware* [5]. Em especial, algoritmos *Bounded Model Checking* (BMC) baseados em Satisfatibilidade Booleana (SAT) ou Teo-

rias do Módulo da Satisfatibilidade (SMT, do inglês *Satisfiability Modulo Theories*) foram aplicados com sucesso para verificação de programas sequenciais e paralelos e na descoberta de defeitos sutis [6, 7, 8, 9].

1.1 Descrição do Problema

Descrição

1.2 Objetivos

Objetivos

1.3 Descrição da Solução

Descrição da Solução

1.4 Contribuições

Contribuições

1.5 Organização da Dissertação

Organização da dissertação

Referências Bibliográficas

- [1] HEATH, Steven. *Embedded Systems Design*. Oxford, Reino Unido: Newnes, 2003. 430 p.
- [2] KOPETZ, Hermann. *Real-Time Systems: Design Principles for Distributed Embedded Applications*. Nova York, Estados Unidos: Springer, 2011. 376 p.
- [3] BAIER, Christel, KATOEN, Joost-Pieter. *Principles of Model Checking*. Cambridge, Reino Unido: MIT Press, 2008. 975 p.
- [4] GOLDSTEIN, Harry. Checking the play in plug-and-play. *Spectrum*, v. 39, n. 4, p. 957–974, 2012.
- [5] BUTLER, Ricky. *What is Formal Methods?* 2001. Disponível em: <http://shemesh.larc.nasa.gov/fm/fm-what.html>. Acesso em: 10 abril 2013.
- [6] CLARKE, Edmund, KROENING, Daniel, LERDA, Flavio. A tool for checking ANSI-C programs. In: *Tools and Algorithms for the Construction and Analysis of Systems, Lecture Notes in Computer Science*. Lancaster, Reino Unido: Springer, 2004. v. 2988, p. 168–176.
- [7] CORDEIRO, Lucas, FISCHER, Bernd, MARQUES-SILVA, João. SMT-based bounded model checking for embedded ANSI-C software. *IEEE Transaction of Software Engineering*, v. 38, n. 6, p. 50–55, 2002.
- [8] CORDEIRO, Lucas. *SMT-Based Bounded Model Checking of Multi-threaded Software in Embedded Systems*. Southampton, Reino Unido: University of Southampton, 2011. 197 p.
- [9] Qadeer, Shaz, REHOF, Jakob. Context-Bounded Model Checking of Concurrent Software. In: *Proceedings of Tools and Algorithms for the Construction and Analysis of Systems*. Redmond, Estados Unidos: Microsoft Research, 2005. p. 93–107.

Apêndice A

Publicações

A.1 Referente à Pesquisa

- **AINDA FALTA MODIFICAR ESSA PARTE**
- **Mikhail Ramalho**, Mauro Freitas, Felipe Sousa, Hendrio Marques, Lucas Cordeiro e Bernd Fischer. *SMT-Based Bounded Model Checking of C++ Programs*. **20th IEEE International Conference and Workshops on the Engineering of Computer Based Systems**, Phoenix, 2013. p. 147-156.
- **Mikhail Ramalho**, Lucas Cordeiro, André Cavalcante e Vicente Lucena. *Verificação Baseada em Indução Matemática para Programas C/C++*. **III Simpósio Brasileiro de Engenharia de Sistemas Computacionais**. Niterói, Rio de Janeiro.

A.2 Contribuições em outras Pesquisas

- **AINDA FALTA MODIFICAR ESSA PARTE**
- Mauro L. de Freitas, **Mikhail Y. R. Gadelha**, Lucas C. Cordeiro, Waldir S. S. Júnior e Eddie B. L. Filho. *Verificação de Propriedades de Filtros Digitais Implementados com Aritmética de Ponto Fixo*. **XXXI Simpósio Brasileiro de Telecomunicações - SBrT**, 2013.