
Título: Especificação, Optimização e Teste de Algoritmos de Controlo Hierárquicos.

Title: *Specification, Optimization and Testing of Hierarchical Control Algorithms.*

Autor/Author: Andreia Barbosa de Melo

Orientador/Advisor: Valery Sklyarov

Data Apresentação/Acceptance Date: 06/01/00

Palavras Chave: Técnicas de Especificação e Optimização, Algoritmos de Controlo, Implementação e Teste em FPGAs.

Key Words: *Specification and Optimization Techniques, Control Algorithms, Implementation and Testing in FPGAs.*

Mestrado/M.S.

Resumo

Este trabalho incide sobre a especificação de algoritmos de controlo, a sua verificação, optimização, conversão em código VHDL sintetizável (*VHSIC Hardware Description Language*) e depuração.

O comportamento dos algoritmos de controlo é especificado utilizando Esquemas de Grafos Hierárquicos (*HGSs – Hierarchical Graph Schemes*). Esta é uma linguagem gráfica formal de especificação que se baseia na utilização de nodos ligados entre si de forma semelhante a um fluxograma. Este método de especificação suporta hierarquia e permite construir algoritmos utilizando uma notação que é independente do modelo de implementação do circuito.

No âmbito deste trabalho foi construída uma ferramenta de software, chamada *GraphBuilder*, que consiste num ambiente integrado para especificação, optimização e teste de algoritmos de controlo hierárquicos. A construção de hierarquias de esquemas de grafos é suportada pelo *GraphBuilder* cujas principais funções são a construção/edição de forma gráfica e a gestão das hierarquias. Nesta aplicação é possível verificar a especificação do algoritmo, assim como gerar algoritmos aleatoriamente, sob determinadas restrições, para efeitos de teste. O *GraphBuilder* integra a aplicação *Optimaker*, responsável pela optimização e a aplicação *SynGraph* que converte os esquemas de grafos em código VHDL sintetizável. A partir do momento em que possuímos uma descrição do circuito em VHDL delegamos os passos posteriores do projecto às ferramentas de software já existentes. Assim, todos os passos desde a síntese do código VHDL até à implementação do circuito em hardware serão assegurados por ferramentas externas que são normalmente dependentes da tecnologia usada para implementar o circuito.

Após a implementação do circuito em hardware torna-se necessário verificar se o seu funcionamento é o que realmente foi especificado e para tal, a aplicação *GraphBuilder* permite efectuar a depuração do algoritmo. Desta forma fecha-se o ciclo do projecto do circuito pois tornou-se possível efectuar a depuração utilizando o mesmo ambiente que foi usado na sua especificação. No

caso de ser necessário efectuar modificações ao circuito, estas poderão ser feitas e testadas mais rapidamente pois do ponto de vista do utilizador está a utilizar-se apenas uma ferramenta de software no projecto do circuito porque o ambiente *GraphBuilder* encarrega-se de invocar todas as aplicações necessárias de forma transparente.

Finalmente, os algoritmos de controlo são implementados em circuitos reconfiguráveis. Neste trabalho utilizaram-se as FPGAs (*Field Programmable Gate Arrays*) das famílias XC6200 (reconfigurável dinâmica e parcialmente) e XC4000 (reconfigurável estaticamente) da Xilinx.

Abstract

This work is devoted to the specification of control algorithms, their verification, optimisation, conversion to VHDL (*VHSIC Hardware Description Language*) code and real-time debugging.

The behaviour of control algorithms is described by HGSs (*Hierarchical Graph Schemes*), which is a formal graphical specification language similar to flowcharts. This method supports hierarchy and parallelism and allows building control algorithms using a notation that is independent of the circuit implementation model.

Within the scope of this work, a software tool, called *GraphBuilder*, was developed. It consists of an integrated environment for specification, experimentation and test of hierarchical specifications. These are supported by *GraphBuilder*, whose main functions are graphical construction, edition and hierarchy management. It also performs verification and generates hierarchical control algorithms randomly, with some constraints, for testing purposes.

GraphBuilder integrates the application *Optimaker*, responsible for optimisation, and *SynGraph*, which converts Graph Schemes to synthesisable VHDL code. After building the VHDL description, we delegate the following design steps to the commercially available software tools, which are dependent on the technology used to implement the circuit.

After the hardware implementation it is necessary to verify if the circuit behaves as specified. For that purpose *GraphBuilder* provides real-time debugging facilities. Consequently, this tool closes the design cycle since it allows for specifying and debugging the circuit using the same environment. This simplifies the task of modifying, improving and correcting the initial specification.

Finally, the control algorithms are implemented in reconfigurable circuits. In this work FPGAs (*Field Programmable Gate Arrays*) of the Xilinx XC6200 (partially dynamically reconfigurable) and XC4000 (statically reconfigurable) families were used.