Automatic Insertion of Fault Tolerance Techniques into VHDL Descriptions

Ana Carla dos Oliveira Santos, Sérgio Vanderlei Cavalcante
{acos,svc}@cin.ufpe.br

Universidade Federal de Pernambuco - UFPE
Centro de Informática
Cx. Postal 7851 CEP 50732-970
Recife – PE – Brazil

Abstract

Computer systems have become more and more popular and their impact on people's lives have become stronger. As users depend more on the performance of machines, a failure in such systems cause a more visible and larger damage. The use of fault tolerance techniques can face this problem by increasing system's reliability. This paper presents a proposal of a tool to insert fault tolerance techniques into hardware system designs described in VHDL.

1 Introduction

Fault tolerance is a systems' characteristic that allows them to continue working even in the occurrence of faults. It is always achieved by using some sort of redundancy, such as hardware, software, information or time redundancy. Hence, applying fault tolerance to a system always increases its development cost and time, being usually done only to critical systems, in which a fault occurrence can cause a huge financial damage, or even threat human lives.

Embedded systems are often likely to be very critical, as they are used into airplanes mechanisms, ABS brakes and other control systems. In embedded systems design, fault tolerance has always been applied in a manual way, usually based on designer's experience.

The natural trend of computer systems design methodologies is to use automatic mechanisms to assist the system designer in some parts of the product development. Automation implies more reliability in the design, since automatic processes can assure the correctness of the final product, avoiding the insertion of human errors during the system development. By making the development faster and easier, automatic processes also make it possible the design of more complex and innovative systems.

Embedded systems design has been following this trend. Methodologies of building embedded systems have emerged in the last few years, improving the quality of the designs. Some examples of these methodologies are the hardware/software co-design ones[Vahid99].

Designing fault tolerant embedded systems, however, is not covered by these methodologies, because the reliability requirements are usually put aside during the development, being considered only at system‘s validation, in which the design can be considered insufficiently reliable, and the system must be redesigned.

As a solution to this problem, one could propose a tool that should be able to analyze embedded system's specifications, decide which fault tolerance technique should be applied in each case, implement this technique automatically in the system's design and finally validate the reliability requirements. This tool's general architecture is shown in Fig.1.

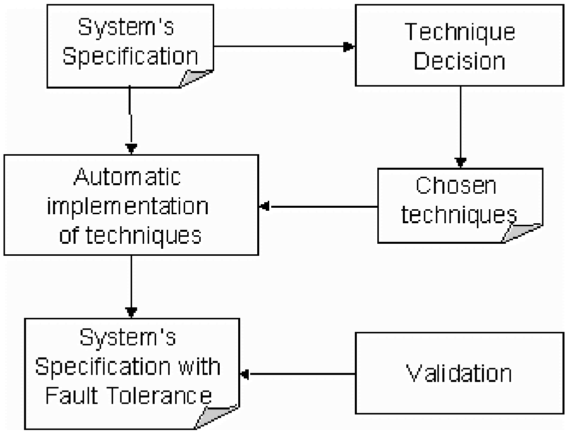
System's Specification
Technique Decision
Automatic implementation of techniques
Chosen techniques
System's Specification with Fault Tolerance
Validation

Figure 1 - The tool architecture.

The development of such a tool presents some challenges due to some research areas still under study. Non-functional requirements, such as reliability ones, are not well formalized yet in order to be treated by automatic processes; decision among several fault tolerance techniques, choosing the one that better meets the system's requirements is not trivial and it is still done intuitively by the designer; lastly, it is necessary to generalize fault tolerance techniques in order to be applied automatically on account of the fact that most of them are strongly dependent on the system in development.

2 Tool proposal

The goal of this paper is the proposal of a tool that implements part of the ideal tool presented in the previous section.

The proposed tool is responsible for the automatic insertion of some fault tolerant techniques into embedded system design, described in VHDL.

In order to insert automatically the techniques into a VHDL description, the tool receives as inputs the original VHDL description and the specifications of which techniques should be applied to each module (VHDL entity or vector) of the system. As a result, the tool generates a new VHDL description based in the original one, but including the fault tolerance features determined by the designer. The tool architecture is illustrated in Fig.2.

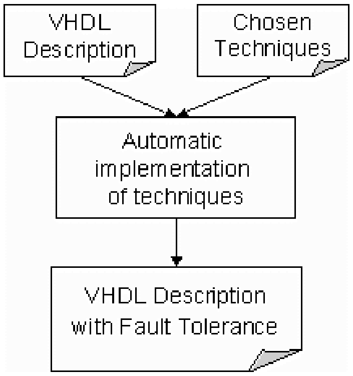
VHDL Description
Chosen Techniques
Automatic implementation of techniques
VHDL Description with Fault Tolerance

Figure 2 - The proposed tool architecture.

The techniques to be offered by the tool are: NMR, mid-value select, flux summing and error detection and correction codes. The following subsections present a summary of each one of them.

2.1 NMR

N-modular redundancy is the most used fault tolerance technique. It consists in replicating a module *n* times and using a component to choose the fault-free output, based in a majority vote among all the *n* outputs provided by the replicated modules. Fig.3 illustrates this technique.

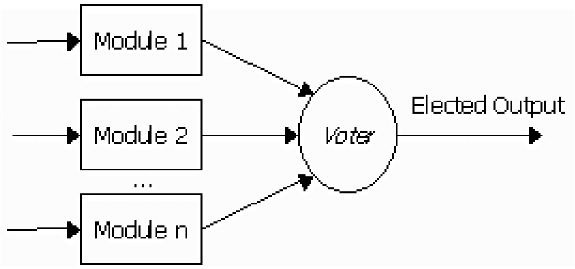

Figure 3 - NMR technique.

2.2 Flux Summing

Feedback control systems would be more benefited by using a NMR variation named flux-summing. In this technique, the voter is substituted by a component that receives the outputs of all *n* modules and generates as output a value that is proportional to the sum of the *n* values. This new value is the one that is used as feedback to the modules. Working this way, all modules can detect a failure and compensate the variation of the system's output[Pradhan96]. See Fig.4.

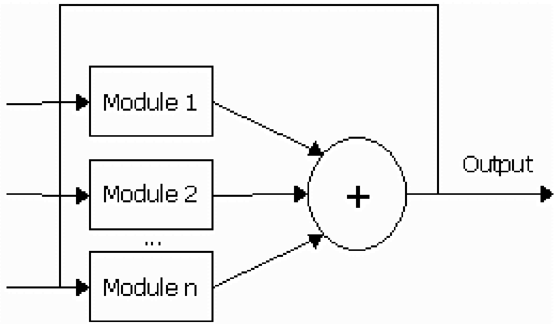

Figure 4 - Flux-summing technique.

2.3 Mid-Value Select

When small variation of outputs is not considered an error of the system, as in analog-digital conversions for instance, the usual NMR technique cannot be applied. In this case the mid-value select technique is more suitable.

The technique consists in choosing among all the *n* outputs of the replicated modules, the one that would lie in the middle if the outputs were sorted[Pradhan96]. The following Fig5. shows how this technique works.

2.4 Error detection/correction codes

More applied to data transferring operations, error detection and correction codes use information redundancy in order to verify the occurrence of data corruption. Some codes are very simple, such as parity bit or duplication code, and others are more elaborated, like Hamming code for instance, offering the possibility of recovering some lost information in the corrupted data.

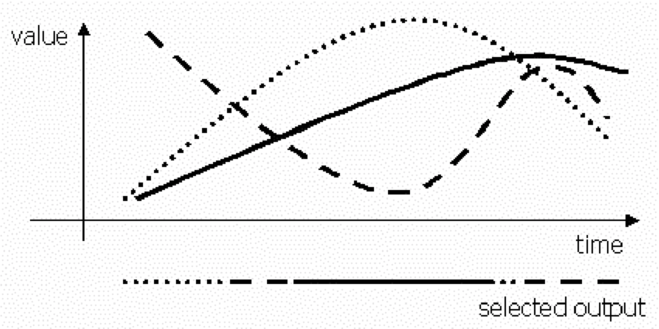

Figure 5 - Mid-value select technique.

3 Current State

The tool is currently under development. A library of VHDL component templates, related to each technique presented in this paper, has been built. A program, developed in Java, is responsible for customizing the templates, in order to generate the VHDL components required to apply the chosen technique to the design. Some parameters requested by the templates are: number of bits of I/O vectors, number of used replicas, and number of clocks before timeout – to be used on implementing synchronism protocols between modules.

4 Conclusions and Future Work

The presented tool would help the designer to determine, a more suitable fault tolerance technique to the system, by offering the possibility of exploring a variety of solutions in an efficient way. It will also increase the quality of fault tolerant embedded systems through the automatic implementation of the techniques by helping the designer to avoid the insertion of errors into the final product.

In the future, it is desired to incorporate to this tool a mechanism to decide which technique should be applied in the design, taking into consideration system characteristics, such as architecture and required level of reliability.

5 References

[Pradhan96] PRADHAN, D. K.; ***Fault-Tolerant Computer System Design.*** Prentice Hall 1996

[Vahid99] VAHID, F.; GIVARGIS, T. ***Embedded System Design: A Unified Hardware/Software Approach.*** Department of Computer Science and Engineering – University of California, 1999.