A System-Level Solution for Dependable Heterogeneous MPSoCs

Ahmed Ibrahim, Hans G. Kerkhoff
Testable Design and Test of Integrated Systems Group (TDT),
Centre of Telematics and Information Technology (CTIT),
University of Twente,
Enschede, the Netherlands
a.m.y.ibrahim@utwente.nl and h.g.kerkhoff@utwente.nl

Abstract—Mission-critical systems require a dependable operation during their lifetime. However, the current ongoing aggressive scaling of technology has resulted in increasing reliability issues. Dependability of such systems has become a major concern in the design process. In this work we aim to enhance the dependability of heterogeneous Multi-Processor System-on-Chips (MPSoCs) by introducing a system-level solution that utilizes the inherent redundancy in MPSoCs in order to achieve a dependable operation in the case of failing processor cores.

Keywords—Dependability; Heterogeneous MPSoCs; IEEE 1687 Standard; iJTAG Standard

I. INTRODUCTION

Heterogeneous MPSoCs emerged as an efficient processing platform, by utilizing different types of processor cores during the processing of a certain application. This was done by mapping different tasks in the application to task-efficient processor cores. An MPSoC is usually characterized by its inherent redundancy, where multiple cores of the same type are processing parts of a certain task in parallel in order to meet the ever increasing applications timing and power requirements.

In this work such redundancy is utilized in a system-level solution for heterogeneous MPSoCs dependability. The idea is to monitor the different cores for the occurrence of permanent faults that could not be masked via the processor fault-tolerance capabilities, if any, then isolate the faulty core and remap its load to another identical spare or lightly loaded one. In that way, an MPSoC could maintain its dependable operation in the presence of a fault at the cost of a degraded performance.

II. BACKGROUND AND PREVIOUS WORK

A heterogeneous MPSoC can be viewed as a processing platform with multiple homogeneous subsystems. Each of these subsystems consists of a number of identical processor cores, each used for a certain type of task. According to the throughput requirements of the designated application, a subsystem can be considered as correctly functioning if it contains a certain number of correctly functioning processor cores. Such systems are referred to as load-sharing $K$-out-of-$N:G$ systems [1], where $K$ is the minimum number of components out of $N$ existing ones, that are required to be fault-free, in order for the system to be considered as correctly functioning. The heterogeneous MPSoCs fail if only one of their homogeneous $K$-out-of-$N:G$ subsystems fail to have $K$ fault-free cores.

Dependability includes a number of attributes which can be used to evaluate the dependability of a certain system. The ones that are of interest in this work are the reliability, availability and maintainability attributes. The proposed dependability technique aims at enhancing those three attributes and hence the total system dependability.

In [1] a system-level dependability solution was introduced for homogeneous MPSoCs, where it utilizes the inherent redundancy in such systems, by replacing faulty cores with spare, fault-free ones. That was accomplished by periodically running high quality deterministic tests on the cores and evaluating the responses, using a specific IP for dependability operations referred to as the Dependability Manager (DM). In this work, the latter solution will be extended for the more general case of heterogeneous MPSoCs. In the next section, the major research points for enabling such techniques in heterogeneous MPSoCs will be discussed.

III. ENHANCING THE DEPENDABILITY OF HETEROGENEOUS MPSoCs

The aim of this work is to introduce a reusable and scalable dependability solution for heterogeneous MPSoCs. Figure 1 shows an example of a dependable heterogeneous MPSoC following the proposed approach. At the centre of this approach is a programmable DM. The processor cores are supposed to be scan-inserted and wrapped using an IEEE 1500 compliant core wrapper. If the core supports an error-detection mechanism, an Error Flag register is assumed to be present for notifying the operating system of the occurrence of an unexpected error. The following research points are investigated in this work.

A. A Programmable DM

The DM contains two essential components for its operation, a Test Pattern Generator (TPG) and a Test Response
Evaluator (TRE). Both components are required to be able to target different types of processor cores, thus they should be fully programmable.

Different techniques are being evaluated for the use in the TPG, examples of those techniques are the ones proposed in [2]-[5]. An important parameter for evaluating the TPG performance is the realized fault coverage; a high fault coverage is required for all the processor core types to be tested. A generated test set with low fault coverage will degrade the system reliability by not detecting faulty cores. The required configurations size is another important parameter in evaluating the different techniques, where a minimum size is favorable as it affects the storage overhead.

The TRE was chosen to be based on the time compaction of test responses using a reconfigurable Multiple Input Shift Register (MISR). One performance measure of such technique is the aliasing probability of the output signatures which mainly depends on the used MISR length. Both the TPG and the TRE are designed so they could be used as stand-alone instruments of their own. Using programmable TPG and TRE enables not only targeting different processor types in a heterogeneous MPSoC, but also reusing the DM in different MPSoCs.

B. A Standardized Configuration Interface

A standardized interface is required to enable reusing the DM in different MPSoCs as a stand-alone IP. The IEEE 1687 standard also known as iJTAG [6] was chosen as a configuration interface for the DM, as it provides a flexible network architecture for connecting the increasing number of on-chip testing instruments to a single controller. Using iJTAG, the DM can be easily included in an existing instruments network without any design overhead. The DM could be accessed by an iJTAG controller, given both the instrument-level access procedures written in the Procedure Description Language (PDL) and the DM interface organization written in the Instruments Connectivity Language (ICL); both languages are defined by the iJTAG standard.

The iJTAG instruments network was initially intended to be controlled via an external controller through the Test Access Port (TAP), however using an internal iJTAG controller is also possible. An iJTAG controller is added to control the iJTAG network internally when the iJTAG instruction is not loaded to the TAP instruction register. Using such controller enables internal reconfiguration of the DM. Finally the DM is considered as a complex instrument where iJTAG is heavily used for its reconfiguration and control. Such instruments require a careful integration procedure in order to achieve an optimum access time and minimum area overhead. In [7] such integration techniques were discussed in detail, and it was adopted in the integration of the DM in this work.

C. Event-Triggered Scheduling of the DM Tests

In [1] only periodic tests were executed on the cores, thus the time-to-detect of the faults is mainly dependent on the period of the scheduled tests. In this work, the error-detection mechanisms in the cores are utilized to instruct the DM to execute dependability tests to detect any emerging permanent faults with a minimum time to detect. In order to accomplish that, an iJTAG-based error network is introduced for error-flag propagation and error localization, this network is similar to the one introduced in [8] for fault management. However the error network is separated from the instrument network and less registers and logic are used in our case.

As the DM is envisioned as a stand-alone IP, opposite to what was introduced in [8], prior knowledge of the error network is not assumed. So in order to trace down an error network, an algorithm was developed for this purpose, and it was implemented as one of the DM features. Using this algorithm, once the error flag is raised in one of the processor cores, the DM can quickly localize the error source without prior knowledge of the error network, and then a dependability test procedure could be initiated for this core.

IV. CONCLUSIONS AND FUTURE WORK

A dependability approach for heterogeneous MPSoCs is being investigated, and it was shown here in a brief explanation of the work done in three of the necessary research aspects of this investigation. This approach will be implemented in an experimental MPSoC, and different parameters will be evaluated for their impact on the dependability of this MPSoC (ex. Error detection latency). Then methods for reducing such impact will be investigated and evaluated.

REFERENCES