

A Cost-Efficient Dependability Management Framework for Self-aware System-on-Chips based on IEEE 1687

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—A cost-efficient framework for executing life-time dependability procedures is presented in this paper. The proposed framework relies on distributed sensors and actuators (embedded instruments) for self-awareness and adaptation, where the IEEE 1687 standard (iJTAG) is utilized for the dependability communications and the on-chip access of the instruments.

Keywords—IEEE 1687, embedded instruments, self-awareness.

I. INTRODUCTION

An adequate-level of self-awareness and self-adaptation in modern System-on-Chips (SoCs) became required in order to mitigate life-time faults and manage variabilities, increased thermal densities and decreased reliabilities of nano-scale technologies. Several procedures that realize self-awareness and adaptation in SoCs have been proposed for enhancing its dependability. For example, Dynamic Reliability Management (DRM) [1], Dynamic Thermal Management (DTM) and Fault Management (FM) [2]. Such procedures depend on in-situ Embedded Instruments (EIs) for measurements and actuation (e.g. temperature sensors, voltage monitors, fault detectors and circuits for dynamic voltage and frequency scaling ‘DVFS’).

It was shown in [3] that a hierarchical management framework is well suited for realizing self-awareness and adaptation in SoCs. The framework consists of hierarchical management units (agents). At the system-level an application agent sends runtime application requirements to a platform agent, which provides system-level services such as components and network reconfiguration. A cluster agent monitors the state of the cluster and performs finer-grained management functions. Finally a cell agent performs the most fine-grained functions such as monitoring and tuning of specific parameters.

The IEEE 1687 standard (iJTAG) enables a hierarchical network infrastructure for connecting the EIs with the Test Access Port (TAP) for off-chip access in case of e.g. testing or debugging. A fundamental component presented by the standard is the Segment Insertion Bit (SIB). A SIB allows to include and exclude a scan segment from the active scan path for optimized instruments access. Consequently, a hierarchical network organization is constructed by nesting multiple SIBs in a tree-like hierarchy with instruments at the leaves.

In this paper the use of iJTAG in realizing a hierarchical design layer for dependability management is presented.

II. A HIERARCHICAL DEPENDABILITY LAYER

A hierarchical design layer based on iJTAG for executing dependability applications is presented in this work. The main components in this layer are: 1) Dependability Managers (DMs), 2) A hierarchical multi-mode iJTAG network and 3) Embedded instruments. The EIs, DMs and the iJTAG network form a parallel processing layer that is decoupled from the functional one; we refer to this layer as the *dependability layer*. This decoupling enables a flexible design of the dependability layer, and dependability procedures could be executed with a minimum impact on the functional processing.

A. The dependability managers

The dependability layer incorporates one or more DMs for executing the dependability procedures and for controlling segments of the iJTAG network. The DMs incorporate a processing unit that is tailored for dependability procedures, an instruction and data cache, a retargeting engine [4] for generating network access vectors and a ROM -referred to as the H-Array [4]- for holding the network organization. Optionally, a DM could incorporate an interrupt management unit for instruments interrupts localization and handling, a set of timers for scheduling periodic procedures and a BIST unit.

Periodic dependability procedures written in a high-level programming language are compiled along with interrupt service routines for handling the instruments interrupts. Dependability procedures include function calls to instruments access procedures that are written in Procedural Description Language (PDL). PDL procedures are compiled as retargeting engine co-processor instructions, and references to instruments are linked to their locations in the H-Array.

B. Hierarchical Multi-mode iJTAG networks

Hierarchical iJTAG networks enable organizing the instruments into nested *hierarchical clusters* which is utilized in the dependability layer. A cluster is accessed by opening its corresponding SIB (updating it with a ‘1’). A DM could be inserted to manage a certain cluster. The DM can access the instruments in the cluster only if the corresponding SIB is closed, such that the accessing priority is always for the higher-level controller. The iJTAG network is a multi-mode network that enables efficient interrupts delivery and localization [5].

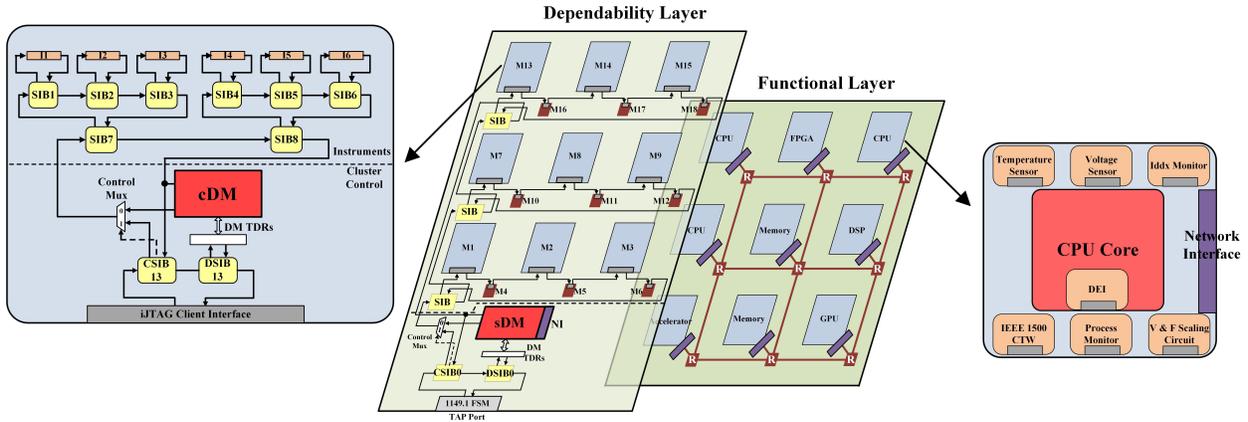


Figure 1: Example of a dependable heterogeneous SoC with a hierarchical dependability management layer.

Figure 1 shows a nine-core SoC with a Network-on-Chip (NoC) of nine routers, and the parallel hierarchical dependability layer. Each core is considered as a cluster controlled by a cluster DM (cDM), routers and cores form the system-level cluster controlled by the system DM (sDM). sDMs correspond to platform agents in [3], cDMs to cluster agents and EIs to cell agents. The sDM communicates with the software-based application agent using a NoC interface.

III. EXAMPLE OF HIERARCHICAL EXECUTION OF DEPENDABILITY PROCEDURES

We discuss the implementation of two dependability procedures and the corresponding hierarchical policies (shown in Figure 2). The procedures are: 1- an application-aware DRM and 2- Fault Management (FM).

The DRM procedure follows the processing flow introduced in [1]. A periodic estimation of each core reliability is performed by the cDMs using the Temperature (T) and Voltage (V) instruments values for each functional unit. The overall reliability at a certain time is calculated using the failure probability models of different failure mechanisms. The cDMs also keep track of the core Utilization (U) using performance monitoring instruments for estimating the remaining workload. The calculated reliabilities along with the estimated remaining workload values are periodically read by the sDM. In addition, the application manager provides the performance requirements for each core in terms of the minimum allowable operation frequency (F_{min}) for the applications running on the corresponding core. In case the required DVFS adaptation for a certain core is below the applications performance requirements, the sDM considers that this core is not fit to perform those applications, and notifies the application agent to start dynamic task remapping to another more reliable core.

Since all instruments are connected to the iJTAG network, additional and/or more accurate failure mechanism modeling could be achieved by using data from other instruments (e.g. NBTI degradation) for a more accurate reliability estimation.

Fault management is performed by both the sDM and the application agent. In case a fault interrupt is received, the sDM immediately initiates a localization procedure using the multi-mode iJTAG network. When the fault is localized and

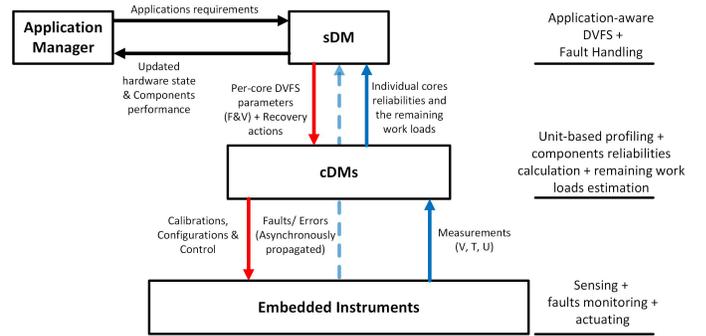


Figure 2: DRM and FM hierarchical policies

classified the sDM notifies the application agent in case a critical uncorrected fault occurred which requires a software-level recovery mechanism. A health map of the system components is maintained and if a component exhibits a number of faults above a certain threshold, the component is isolated and marked as faulty [2]. Since iJTAG enables a scalable integration of fault detectors, enhanced fault coverage could be achieved with minimum design and integration overhead.

IV. CONCLUSIONS

iJTAG networks enable an efficient realization of dependability procedures due to their hierarchical nature and since they connect embedded instruments in a standardized manner. In this work we presented a functionally-decoupled hierarchical dependability layer using iJTAG, then we discussed the hierarchical execution of two dependability procedures and showed how iJTAG allows for their scalable implementation.

REFERENCES

- [1] C. Zhuo, D. Sylvester and D. Blaauw, "Process variation and temperature-aware reliability management," Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 580-585, 2010.
- [2] K. Shibin, S. Devadze and A. Jutman, "On-line fault classification and handling in IEEE1687 based fault management system for complex SoCs," Latin-American Test Symposium (LATS), pp. 69-74, 2016.
- [3] Liang Guang et al., "Hierarchical agent monitoring design approach towards self-aware parallel systems-on-chip," ACM Transactions on Embedded Computing Systems (TECS), vol. 9, no. 3, pp. 124, Feb 2010.
- [4] A. Ibrahim and H. G. Kerkhoff, "Analysis and design of an on-chip retargeting engine for IEEE 1687 networks," European Test Symposium (ETS), pp. 1-6, 2016.
- [5] A. Ibrahim and H. G. Kerkhoff, "Efficient utilization of hierarchical iJTAG networks for interrupts management," Int'l Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), pp. 97-102, 2016.