

Design of an Infrastructural IP Dependability Manager for a Dependable Reconfigurable Many-Core Processor

Hans G. Kerkhoff and Xiao Zhang

Testable Design and Test of Integrated Systems (TDT) Group

Centre of Telecommunication and Information Technology (CTIT), Enschede, the Netherlands

h.g.kerkhoff@utwente.nl and x.zhang@utwente.nl

Abstract

Reconfigurable many-core processors have many advantages over conventionally designed devices, such as low power consumption and very high flexibility. For an increasing number of safety-critical applications, these processors must have an ultra high dependability. This paper discusses the design and verification of an infrastructural IP, the Dependability Manager, which takes care of most essential dependability issues. Several additional innovative approaches with regard to dependability have been incorporated, like the NoC, wrapper and Network Interface design. The Dependability Manager design has been verified on an FPGA and is being processed in UMC CMOS technology as part of a many-core processor.

Keywords: dependability, availability, reliability, many-core processors, reconfiguration, DFX, SoC, BIST

1 Introduction

The advances in digital processors are often related to many-core processors, using more (dual, quad etc.) than one processor IP in a processor SoC. In order to cope with the huge data communication requirements between these cores, the cores are often interconnected by a Network-on-Chip (NoC). If the cores are identical, they are often referred to as “tiles”.

On the other hand, these highly complex SoCs are increasingly used in safety-critical applications, like in the automotive, medical and military arena. This demands ultra dependable processor SoCs [1]. Because there are many tasks to be performed to accomplish this goal, the design of a dedicated Dependability Manager (DM) is considered nowadays to be a promising approach. As the DM is not related to a functional task of the SoC, it is referred to as an Infrastructural IP (IIP). This paper deals with the design and verification of a DM for a Reconfigurable Fabric Device (RFD) as being developed within the European CRISP¹ project.

The paper is organised in the following way:

First, the global architecture of the RFD is briefly discussed. It shows a very high regularity in terms of the tiles, interconnected by a NoC. The tile is a reconfigurable pipelined Xentium processor core from Core Systems and associated local memories. This high regularity provides a clue with regard to the

periodic structural testing of these tiles, which is the starting point of our dependability approach. Repair is accomplished via *run-time mapping* of remaining fault-free reconfigurable Xentiums on the application.

Next, the environment of the DM in the SoC is explained in more detail, including Network Interfaces (NI), Xentium tile and DM wrappers and the NoC.

The central part of the paper discusses the functional blocks in the DM and their interaction, being the test-pattern generator (TPG), the test-response evaluator (TRE) and the controller (FSM). Simulation results, as well as FPGA hardware tests, are shown. Finally, some conclusions are provided.

2 The Dependable Reconfigurable Fabric Device

For many applications, like e.g. beam-forming, a flexibility of the functionality of the processing elements in real time in a SoC is an advantage to cope with changing requirements of the application due to actual circumstances. A possible set-up of such a SoC is shown in Figure 1. It consists of many reconfigurable processing tiles, being a Xentium processor core and its local associated memories, interconnected by a high performance (wormhole) NoC. The configuration for the individual tiles is taken care of by a General Purpose Device (GPD), which can be on-chip (e.g. ARM9-based IP) or off-chip. As the RFD is meant to be used for safety-critical applications, the dependability has to be very high. The high degree of regularity, as well as the NoC communication provides new innovative ways to guarantee dependability.

¹ This research is conducted within the FP7 Cutting edge Reconfigurable ICs for Stream Processing (CRISP) project (ICT-215881) supported by the European Commission.

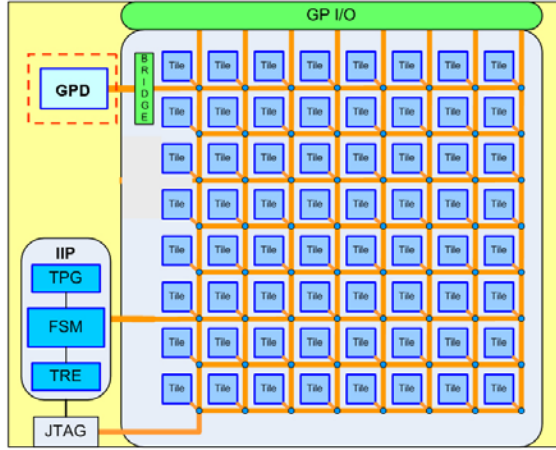


Figure 1: Basic setup of a Reconfigurable Fabric Device (RFD) including 64 Xentium tiles [2].

In our case, two attributes [1] are of key importance:

- on-chip detection of stuck-at faults in the tiles and NoC occurring during its life-time, relating to reliability (0,9783, 15 years), and subsequent repair
- fast recovery time (10ms), being the time from the occurrence of a fault up to repair and re-initialization, resulting in a very high availability

The central point of focus in this paper is the left bottom IIP in Figure 1, the Dependability Manager [1, 2]. It receives its commands from the GPD, which includes a dependability API, over the NoC. As first step, the NoC is functionally tested by the GPD. Basically, the hardware TPG generates test-patterns for the Xentium core which are distributed over three Xentium cores via the NoC, chosen by the GPD. Subsequently, the three test responses are sent to the TRE via the NoC which compares the results, and flags in case of a fault. In the latter case, the GPD starts a run-time remapping operation (software), thereby omitting faulty tiles and/or NoC segments.

3 The Dependability Manager in the RFD SoC

Because the DM communicates via the NoC with the Xentium tiles as well as the GPD, special measures have to be taken care of. An important condition of our approach is the fault-free behaviour of the NoC. This is taken care of via software running on the GPD, which basically verifies the functional behaviour of the NoC; this will not be further discussed in this paper. In the first paragraph, the environment of the DM in relation with the tiles is discussed, while in the second paragraph the NI is dealt with in more detail.

3.1 Environment of the Dependability Manager

Figure 2 shows the most essential parts in the communication between the DM and the Xentium

tiles. The NoC is a dedicated design of the packet-switched wormhole type, capable of multi-casting and running at 200 MHz. The multi-casting is required for providing the test-vectors at multiple Xentium locations. The NoC has routers at each crossing, determining the actual routing of the packet. More detailed information can be found in references [3, 4].

Each scan-based Xentium core has a specially designed wrapper, which is used during normal SoC final testing as well as during its life-time for accomplishing the dependability scenario. The associated Xentium memories are locally BISTed, and finally OR-ed with the final scan-result (OK-NOK). The design of the wrappers will be subject of another paper. The Xentium network interface (NI) has been designed by Recore Systems and will not be treated here either.

The Dependability Manager Network Interface (DM-NI), shown in the left-hand IIP has been specially designed for this purpose, and is discussed in detail in the next paragraph. The TPG can generate 32-bit test vectors on demand, which are subsequently multicast to three chosen Xentium tiles. The control part (FSM) also sets the Xentium wrappers for the dependability scenario. The test responses are routed via two channels to the TRE being a result of bandwidth requirements. The DM can be configured by the GPD via the NoC and a Multi-Channel Port (MCP) in the case the GPD is off-chip.

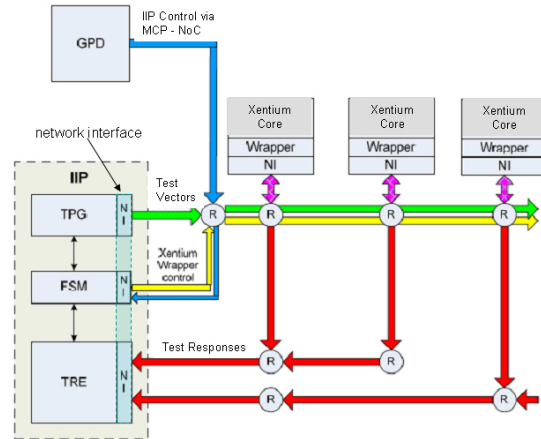


Figure 2: Essential parts of the DM communication in the RFD. The DM wrappers have been omitted.

3.2 Network Interface of the DM

As shown in Figure 2, the network interface (DM-NI) is an essential part of the DM-IIP. It takes care of the bidirectional communication between the TPG, TRE and FSM on one side, and the NoC at the other side.

The basic scheme of the DM-NI, divided in a sending and receiving part, is shown in Figure 3. From the In links, data arrives from the NoC, while from the Out links data departs to the NoC. Because of our

bandwidth requirements, two virtual channel handlers are required of 4 virtual channels each.

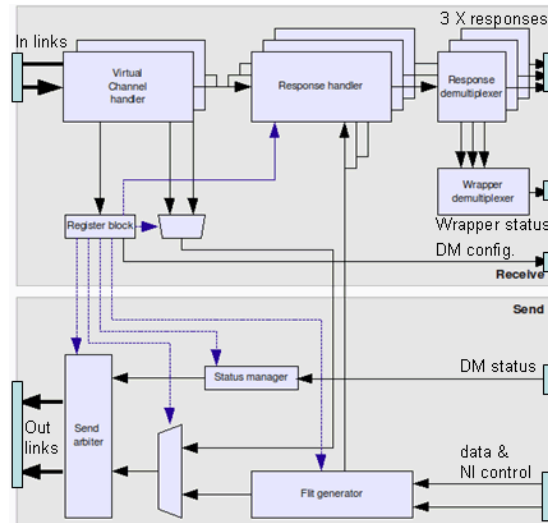


Figure 3: Simplified scheme of the DM network interface.

The data from the three Xentium (X) tile responses, for instance, are buffered in the response handlers, and then separated in Xentium response scan data and memory BIST data via the Xentium wrapper status. This data is subsequently handled by the TRE. In the case the GPD is activating the DM via the NoC, this configuration data is routed towards the DM configuration input. In the lower right part of the NI (Figure 3), the generated test-vectors of the TPG (data) are loaded in the flit generator. In the succeeding multiplexer, the chosen Xentium tiles or their internal addresses are chosen and finally multi-

casted over the NoC via the Send arbiter.

Figure 4 shows a Modelsim simulation to illustrate the communication in the NI. For the sake of simplicity, only a part will be discussed. Box “a” consists of the In links and Out links. Box “b”, includes the DM configuration and status. The three Xentium responses and the white line TPG data is shown in box “c”. The last box “d” shows NI data and control lines. In (1), Figure 4, the GPD addresses the NI via the NoC. As a result, the NoC Out link and connection is being configured (2), as well as the DM (3). In (4) and (5), the Xentium wrappers and Xentiums are configured for testing first and their status read subsequently. Responses are shown in (6). In (7), the commands for test-vector generation (TPG) are given in the DM, which starts in (8). This TPG data is subsequently put on the NoC in (9). In the next paragraph, the DM parts will be discussed in more detail.

4 The Dependability Manager in Detail

This paragraph will provide detailed information on the Dependability Manager. As Figure 5 shows, it consists of three main blocks, the Test Pattern Generator (TPG), the Test-Response Evaluator (TRE), and the local controller based on finite state machines (FSM). For completeness it is noted that the embedded memories, which are part of the Xentium tiles, are locally BISTed. Hence no TPG or TRE is required for this purpose. The parts only concern the Xentium *cores*. The combined network interface (NI) has been previously described. First, the TPG is

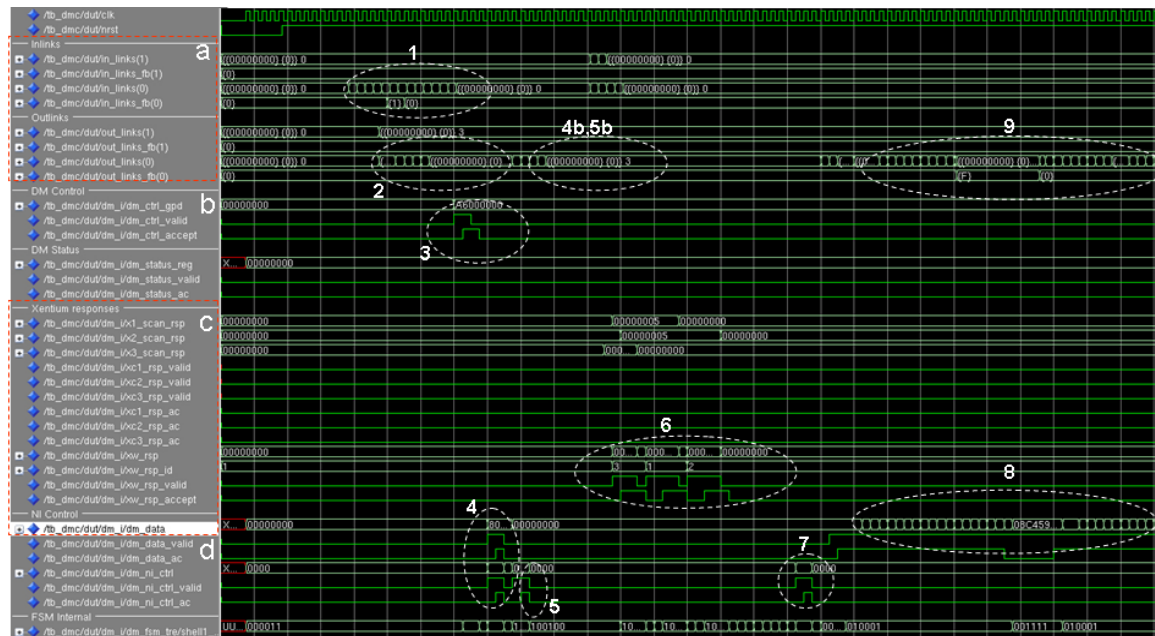


Figure 4: Communication via de NI between NoC and TPG, TRE and FSM.

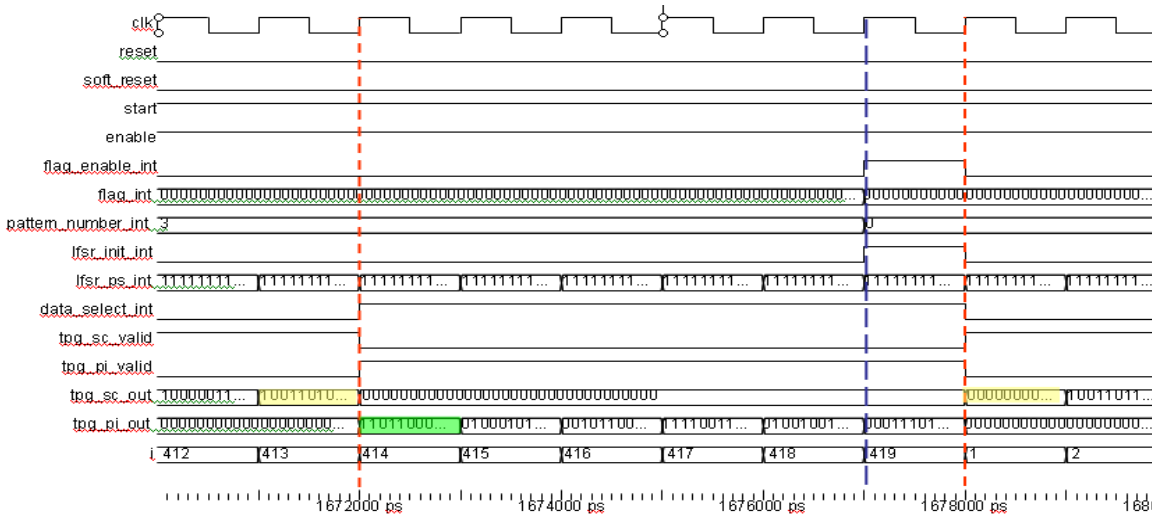


Figure 7: Four test vectors generated from compiler implementation.

within the Dependability Manager IIP. The fact that many Xentium tiles are present, enables the use of comparison between (3) Xentium cores, assuming that identical faults will not occur at 3 locations simultaneously [4, 7]. This greatly reduces the area required otherwise for evaluation. The basic design of the TRE is essentially a 3-input 32-bits comparator, preceded by three buffer FIFOs and a crossbar and a dedicated controller unit; it has already been published in reference [4]. However at that stage, the TRE was still considered a separate IIP, requiring its own network interface (NI). As a consequence, the TRE has been adapted later on, simulated in QuestaSim, implemented on a Xilinx Virtex4 board and subsequently tested. The simulation results of the new TRE are shown and explained in Figure 8.

In the first 3 signals (black boxed at top left), the clock and resets are shown. After that (first arrow top left), the 32-bit results from the Xentium(s) are shown serially. In the middle arrow labelled “a”, a fault has been introduced/injected. This results in the arrow labelled “b” (signal full_pass) indicating that an error

has occurred during comparison, and hence a Xentium core failed. The arrow labelled ‘c’ indicates that the buffers in the TRE are full, and hence no new data can be read in. The bottom signal “full_fail_pointer” using the bidirectional arrow indicates during which test vector the comparison noticed a difference. The simulations showed that the circuit could operate beyond the required 200 MHz.

4.3 The Control Part of the DM (FSM)

The DM accepts commands from the dependability software running on the GPD (Figure 2), via the DM configuration register. The DM can carry out tests as specified, e.g. which tiles are involved, and update the register to report the test results to the GPD. The internal control in the DM is carried out via a finite state machine (FSM), which was designed using the StateCAD software of Xilinx.

The design was extensively verified by simulation for several dependability and debugging scenarios, including emulating faults in the Xentium core.

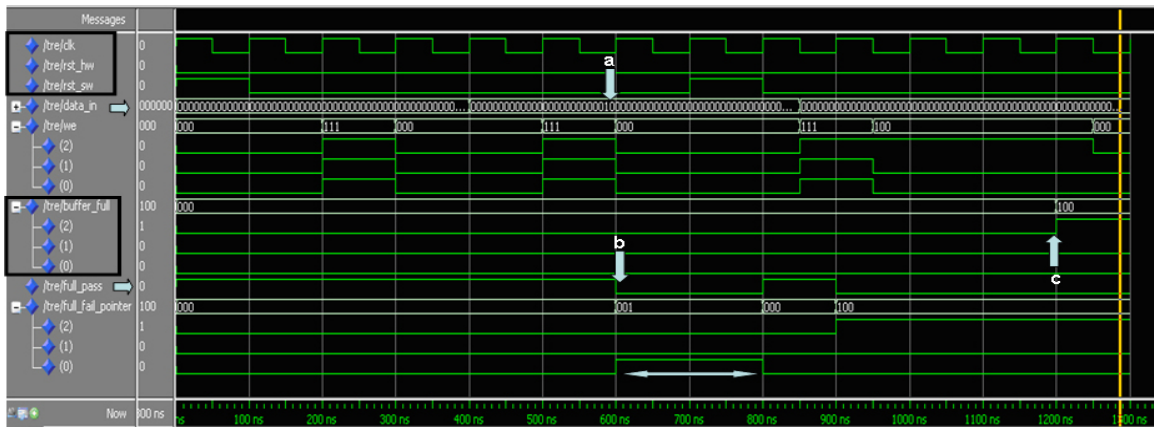


Figure 8: Simulation of the TRE functionality.

4.4 Hardware Test Verification of the DM

The complete DM, being the TPG, TRE, FSM and the NI was synthesised and implemented on a Xilinx Virtex 4 board for carrying out hardware tests. The total space required was 13%. Synopsys synthesis resulted in around 78k equivalent logic gates. For DM hardware test evaluation purposes, a RS232 data communication between FPGA and a PC was used in combination with a developed GUI in Visual Basic. A maximum test frequency of 212 MHz was used. As example, Figure 9 shows the GUI of the set-up to test the TRE part which includes some test results. The status block refers to failures in buffers (full) or the data streams (Xentium test responses). The control block provides TPG options, where data fault responses can be automatically generated. Most interesting is the communication viewer. It shows the three 32-bit data streams, the control commands (yellow/grey boxes) and the measured responses from the TRE (Virtex-4). The first result shows a fault in the first data stream, while the second occurs in the third data stream. It illustrates the correct operation of the TRE [8].

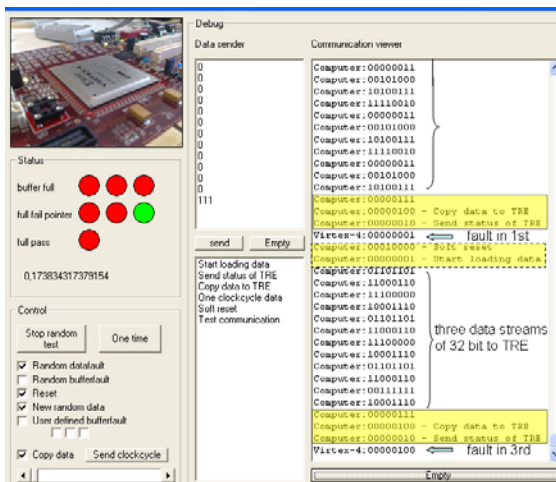


Figure 9: The TRE part hardware verification by means of an FPGA [8].

After completing the full verification on FPGA, the next step will be the processing in an UMC CMOS process. The current chip layout shows a total Silicon area of 0.24mm² for the DM in 90nm technology.

4.5 Debugging and Dependability of the DM

The DM is equipped with scan-cells, wrappers and dedicated pins for test and debug. During production test, the DM is scan tested in a conventional way. For prototype evaluation, 21 pins are available for direct-pin debugging, in which case the DM is considered as stand-alone IIP. In the current version of the DM, no additional means have been incorporated to increase the hardware dependability of the IIP itself. However,

it is possible to include fault-tolerant comparators and TPG hardware. During its lifetime, the current DM can be internally tested periodically; means have been included to take over its function in the case of failure, by software or external hardware. In both cases, however, at the cost of a significantly decreased (especially in the software approach) availability.

5 Conclusions

In this paper we have discussed the design and verification of an infrastructural IP, the Dependability Manager (DM). It is the essential part for enhancing the dependability of our many-cores Reconfigurable Fabric (RFD). It can be controlled with an internal or external General Purpose Device via a Network-on-Chip.

6 Acknowledgements

The authors would like to acknowledge the discussions with the partners of the CRISP consortium, especially Bart Vermeulen, and Mark Westmijze for the network interface development.

7 References

- [1] S. Sakai, M. Goshima and H. Irie, "Ultra Dependable Processor", *IEICE Trans. Electronics*, vol. E91-C no. 9, pp. 1386-1393 (2008).
- [2] X. Zhang and H.G. Kerkhoff, "Design of a Highly Dependable Beam Forming Chip", in *Proc. Euromicro on Digital System Design (DSD09)*, Patras Greece, pp. 729-735 (2009).
- [3] O.J. Kuiken, X. Zhang and H.G. Kerkhoff, "Built-In Self-Diagnostics for a NoC-Based Reconfigurable IC for Dependable Beamforming Applications", in *Proc. IEEE Intern. Symp. on Defect and Fault Tolerance in VLSI Systems (DFT08)*, Cambridge USA, pp. 45-53 (2008).
- [4] H.G. Kerkhoff, O. Kuiken and X. Zhang, "Increasing SoC Dependability via Known Good Tile NoC Testing", *IEEE Intern. Conf. on Dependable Systems and Networks (DSN08)*, Anchorage USA (2008).
- [5] M. Duiven and F. van der Ende, "Design of a Generic 90nm Test-Pattern Generator", *Technical Report University of Twente*, 69200912, July, 78 pages (2009).
- [6] T. Bruintjes and T. Jongsma, "A Xentium TPG at Transistor Level", *Technical Report University of Twente*, 69200914, July, 42 pages (2009).
- [7] H.G. Kerkhoff and J.J.M. Huijts, "Testing of a highly Reconfigurable Processor Core for Dependable Data Streaming Applications", in *Proc. Symposium on Electronic Design Test and Applications (DELTA08)*, Hong Kong China, pp. 38-44 (2008).
- [8] W. van den Beld and J. Huiting, "Simulation and Implementation of a Test Response Evaluator on a FPGA", *Technical Report University of Twente*, 69200918, July, 71 pages (2009).