Testing of a Highly Reconfigurable Processor Core for Dependable Data Streaming Applications

Hans G. Kerkhoff, Jarkko J.M. Huijts
Testable Design and Test of Integrated Systems (TDT) Group
Centre of Telecommunication and Information Technology (CTIT), Enschede, the Netherlands
h.g.kerkhoff@utwente.nl

Abstract

The advances of CMOS technology towards 45 nm, the high costs of ASIC design, power limitations and fast changing application requirements have stimulated the usage of highly reconfigurable multi-processor-cores SoCs. These processing cores within the SoC can be subsequently connected with each other by a communication-centric NoC, thereby reducing data-traffic problems. The (repetitive) multi-processor-cores feature inside these SoCs, the programmable routing via NoC, as well as the repetitive hardware in the cores themselves provides new opportunities for efficient testing at different hierarchical levels. These opportunities, and the inserted DfT, test vectors and coverage can be subsequently applied for enhancing the dependability of SoCs as well as these cores via self-repair. As examples of new opportunities we introduce the feedback loop and KGC concept for enhancing diagnosis and reducing external communication respectively. The self-repair can be done either by rerouting of unused resources or software remapping of correct resources to an application.

1. Introduction

The advances of CMOS technology, currently at 45 nm, enable the implementation of 800-million-transistor quad-core SoCs [1]. However, the design gap and power-delay constraints stimulate the reuse of processor cores even within a single design. Multiple-processor-core SoCs are currently common and have already reached more than 64 cores [2]. If this involves the same processor core, it is usually referred to as processor *tile*. As the data communication between cores becomes a speed bottleneck in conventional bus architectures, Networks-on-Chip (NoC) are introduced where the routing path is reconfigurable [3]. Furthermore, with the short life cycles of microelectronic components and rapidly expanding standards, reconfigurability of most hardware is becoming essential in many consumer products [4]. The efficient testing of these complex SoCs is a tremendous task. This is especially disturbing in the case of dependable systems, where (self-) testing/diagnostics and subsequent repair by software-controlled hardware is a recurring event.

This paper describes the testing of a highly reconfigurable data processor for data streaming operations, from the point of view of creating a dependable system. Application is a wireless multi-sensor network for homeland security purposes in a hazardous environment. Three levels of resolution for test (and repair) will be treated.

The repeatability of processor cores and their reconfigurable interconnections form the basis of our work.

1.1. A highly reconfigurable multi-processor-cores SoC

Figure 1 shows the setup of a conceptual System-on-Chip (SoC), in which a cluster of Reconfigurable tile Processors (RP) is included, which are connected by a Network-on-Chip.

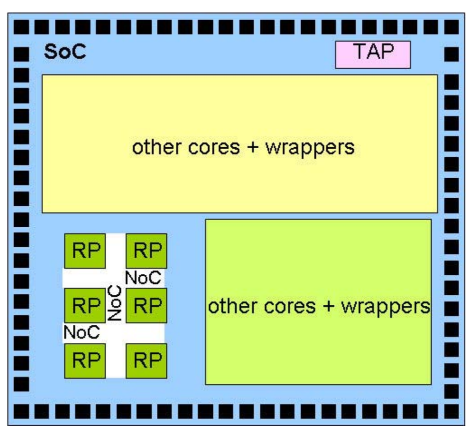

Figure 1. Highly Reconfigurable Processors (RP) interconnected by a Network-on-Chip (NoC) as part of a complex dependable SoC

0-7695-3110-5/08 $25.00 © 2008 IEEE
DOI 10.1109/DELTA.2008.34

Due to the combination of complexity and non-matured processing, the yield and reliability are expected to be low. Hence it is of importance to know which RP(s) is (are) not correctly operating, so that the (software) mapping of streaming data algorithms (e.g. FFT) can be mapped to fault-free RPs only.

With regard to testing, the question arises whether to treat the RPs and NoC as a group or individually. Our approach is to choose one RP randomly and carry out full scan-based automatic test-pattern generation (ATPG), which will be treated later. However, the first step is to verify the NoC [3]. If the chosen RP passes the tests correctly, it is classified as Known Good Core (KGC).

If not, the RP is classified faulty, and the next chosen RP follows the same procedure, until a KGC has been found. Assuming some remaining RPs to be tested, then this KGC will be the reference to determine the correct operation of the others via comparison and utilizing the NoC reconfiguration capabilities. Advantages of this approach are that the comparisons are an internal matter (suitable for BIST), and a high diagnostic content can be preserved (for self-repair). In the last section we will show that this concept can also be applied to other diagnostic resolution levels.

1.2. Example of a highly reconfigurable data processor (RP)

The top level in this paper is the RP tile. As an example we use the Montium tile processor [4, 5]. We define this as the *tile diagnostic level*. We will now go down one level in hierarchy, by looking *inside* the highly reconfigurable Montium processor.

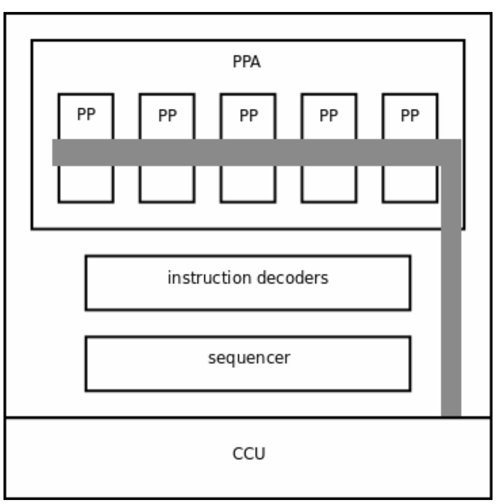

Figure 2. Architecture of the highly reconfigurable Montium tile processor [4, 5]

In this case, we can also distinguish a repetition of identical hardware components (PP), communicating via a reconfigurable interconnect (thick grey bar) as figure 2 shows.

The Processing Part Array (PPA) contains the data path that performs the actual processing. It has five identical Processing Parts (PPs). The remaining components are the sequencer containing the program for actual operation and a series of instruction decoders. A Configuration and Communication Unit (CCU) implements the interface to the NoC (see figure 1) and provides configuration services. The non-PPA blocks are basically sequential circuits, also incorporating a large amount of reconfiguration registers.

The generic VHDL code of the Montium tile for low-power data stream applications was partly available to us [4, 5]. In this design, the DfT infrastructure was still missing. We have used the UMC CMOS library kit and Synopsys Design Compiler to obtain the logic-gate level implementation and scan insertion, while for verification, the logic simulator Modelsim was employed. For scan-based ATPG, the Synopsis TetraMAX tool has been used.

As the CCU VHDL code was still in development and the PPA is the only part containing repetition, ATPG has only been carried out at PPA level and below (see figure 2).

The total area of the PPA takes up 2.9 mm^2 in our technology. This data will be used later on to determine the DfT silicon overhead percentage.

2. The PP and its basic sub-cores in the Montium tile processor

2.1. The Processing Part (PP)

The basic setup of the processing part, of which there are 5 in a PPA (figure 2), is shown in figure 3. It consists of 2 memory units (M1 & M2), the reconfigurable interconnection, and register files (not shown) connected to the reconfigurable ALU. It also includes all relevant reconfiguration registers. As this PP structure is repeated several times, it is a good candidate for self-diagnostics and self-repair purposes.

We define this as the *PP diagnostic level*. The total area of the PP, without DfT, is 0.57 mm^2.

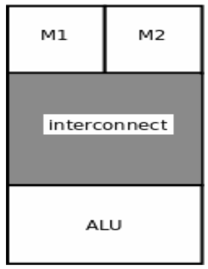

Figure 3. Basic setup of a highly reconfigurable processing part (PP)

2.2. The reconfigurable ALU

The ALU is highly reconfigurable in order to optimize the used hardware for efficient power usage in relation to the required DSP operations as shown in figure 4. Originally, also latches were introduced for the result and status output bits for energy efficiency reasons; however, as it drastically reduces the test coverage we have removed them from the design. The ALU is preceded by register files which are not part of the ALU itself. A single ALU, excluding DfT, requires 0.19 mm^2 area.

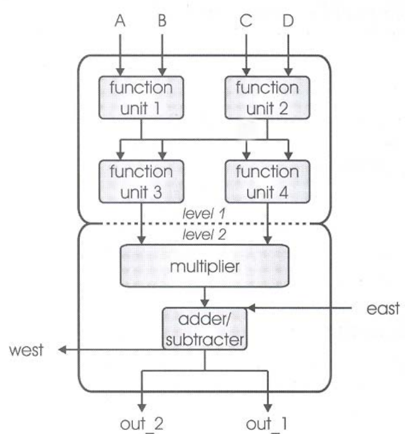

Figure 4. Setup of the reconfigurable ALU

2.3. The memory units (M1 & M2)

The memory within a PP is divided in a left (M1) and right (M2) memory unit. The storage capacity of each is 16 Kbit, and consists of a flexible Address Generation Unit (AGU) and associated SRAM array. The SRAM cells are fully synchronous and single-ported, and available as six-transistor cells. Classical BIST design options are possible for these memories, but the size of the memories (10 times 16 Kbit) does hardly justify the usage of this feature. However, as we will see later on, the number of required test vectors does. The total area of the memory array part is 49k μm^2, while the AGU measures 36k μm^2.

The (16-bit) data input and output of both memories is accessible via the interconnection part (figures 3 and 5). The control of the address generator requires additional control lines.

2.4. The Interconnections

The reconfigurable interconnections which enable the communication between the processor parts (PP) can be considered as the concatenation of identical slices as illustrated in figure 2. As a whole it has been defined as PPA interconnect. The slice *within* a PP is shown in figure 5, defined as PP interconnect. It has a global part in the middle, which links all PPs, and two local parts that remain within the PP. At the left is the local ALU bus, and at the right the local memory bus. The on-board reconfiguration registers determine the routing of the global data to and from the different parts. The total area of a PP interconnect, excluding DfT, is 47k μm^2.

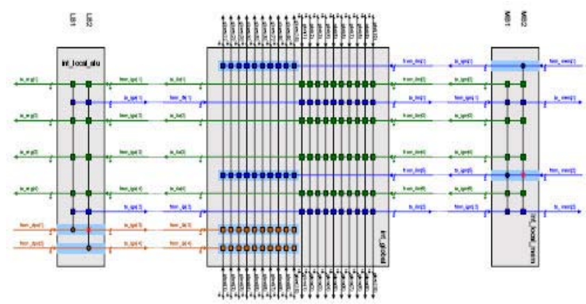

Figure 5. Structure of the reconfigurable interconnection within a single PP

3. Testable design & test of PP and its basic sub-cores

The testable design and test generation of reconfigurable multi-processor cores poses several difficulties [6-8]. As in our case the non-PPA parts are basically state machines (figure 2), a standard scan-based test approach can be used. The PPs and their sub-cores (ALU / interconnect / memory units) turned out to be more challenging. This is because the test generation should also be the basis for realizing a dependable design using internal hardware test generation and subsequent repair via software-reconfigurable hardware in a later stage. Hence, the highest resolution of diagnostics had to be defined. The sub-cores within a PP are the first logical choice because of their repetitiveness within the PPA. This means that one should be able to e.g. determine whether an ALU within a PP is faulty. If in another PP both SRAMs are detected to be faulty, one can *construct a new PP* using the resources of both via the reconfigurable interconnect. We define this as the *sub-core diagnostic level*. It is obvious that fault-free interconnections are a crucial first concern.

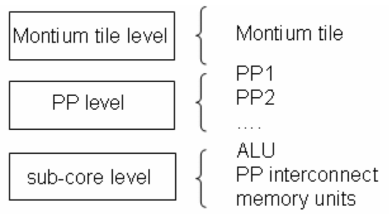

Figure 6. Our three levels of diagnostic resolution

Three levels of diagnostic resolution have been introduced. They are summarized in figure 6. The tile level includes the testing of an entire tile, which enables us to determine which tiles in an SoC can be used (figure 1). The PP level includes the testing of a PP, which enables us to determine which of the five PPs can be used for computations. The sub-core level includes testing the PP's sub-cores, which enables us to determine exactly which ALUs and memory units can be used. The PP interconnect must be fault-free for any of the sub-cores to be usable. This sub-core level is likely to greatly improve the yield by not writing off partly faulty PPs and tiles which can nevertheless still be used with a lower Quality of Service (QoS).

With regard to ATPG, the following strategy was used. Depending on the diagnostic resolution, a test block was chosen.

Table 1. Results of ATPG (TetraMAX) for interconnect within the PPA

Fault class	code	#faults
Detected	DT	86569
Undetectable	UD	18
Not detected	ND	7
Total faults		86594
Test coverage		99.99%
#internal patterns		89

Major difficulty was the test partitioning. Currently, the reconfiguration registers are not taken into account in the netlists for ATPG. However, they were used in order to satisfy the controllability requirements for ATPG. The reconfiguration registers are tested separately via a scan-based approach. After partitioning, all inputs and outputs of the test block were evaluated in terms of full observability and controllability required for ATPG. In the case of testability problems, a capture DfT cell or a set/capture cell has been inserted. The first is a multiplexed D flip-flop (area: 98 μm^2), while the second is a wrapper cell based on IEEE 1500 (area: 122 μm^2).

Finally, it should be mentioned that the *test coverage* in the TetraMAX reports is defined as the ratio of detected faults plus half the possibly detected faults, and all faults minus the undetected faults.

3.1. Testable design & test of the interconnect

In this part, first the PPA interconnect will be treated, then the PP interconnect, and finally the concept of loopback will be introduced. This distinction is required with regard to the different diagnostic levels.

3.1.1. PPA interconnect. This partitioning is required for the PP diagnostic level; the latter also requires the loopback concept (3.1.3). The total area of the PPA interconnect, excluding DfT, is 235k µm2. It does not employ a scan chain, as it consists of pure combinational logic only. Table 1 shows the high test coverage of the ATPG results from TetraMAX for this part.

3.1.2. PP interconnect. This partitioning is required for the sub-core diagnostic level. With regard to in- and outputs full observability and controllability is definitely not a fact, as most are data lines between sub-cores. One requires DfT for all vertical connections (to/from the ALU and memory units): 128 set/capture cells and 96 capture cells. Table 2 shows the results from ATPG providing 100% test coverage.

Table 2. Results of ATPG for the PP interconnect

Fault class	code	#faults
Detected	DT	18994
Undetectable	UD	6
Total faults		19000
Test coverage		100.00%
#internal patterns		44

The silicon area without DfT is 47k μm^2. The DfT cells give an overhead of 53%. Again this does not contain any scan overhead. It is obvious that this DfT for obtaining high diagnostic resolution comes at a (relatively) high price.

3.1.3. The Loopback Concept. As mentioned earlier, the interconnect plays a pivotal role. Its correct functioning has to be established before that of any PP or its sub-cores. As a minimum requirement, each PP global interconnection lines must be able to correctly pass data to the next PP. Otherwise this PP will not be useful anyway.

To enable the testing of these global lines, we have used loopbacks after each PP interconnect, as illustrated in figure 7. Any of the loopbacks can be selected, so that the global lines up to a specific PP can be tested. We start by selecting the fifth loopback to see whether all global lines within the PPA interconnect are intact. If this is not the case, we can subsequently select previous loopbacks to determine up to which PP the global lines are fault-free.

Any PP not reachable by the global lines does not need any further testing, since it cannot be used for repair operations.

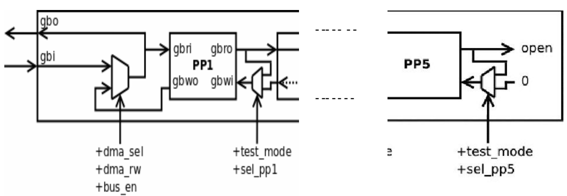

Figure 7. The DfT loopback approach for enabling improved resolution for diagnostics and self-repair

The loopbacks also take care of the controllability and observability of these lines when testing each PP interconnect as part of the PP or sub-core tests. As all global data lines are 160 bits wide, the loopbacks amount to 800 multiplexers.

3.2. Testable design & test of a PP

The PP diagnostic level has been introduced as there are five PPs in the PPA. Hence, one test partitioning involves the complete PP (figure 3). A study on the testability of the in- and outputs with respect to ATPG constraints resulted in the requirement to insert 96 capture cells, and 157 set/capture cells to enable sub-core diagnosis. It results in a DfT overhead of 19.4%. This is not strictly necessary for PP diagnosis only.

It should be noted that *at PPA level*, which is required for PP diagnosis, five times 160 (800) multiplexers are required for loopback purposes for the interconnect at PPA level, as well as five 18-bits set/capture and 18 capture cells for the East-West connections (figure 4). Finally, five 1-bit capture cells and one set/capture cell are required at *Montium tile level*. The results of scan-based ATPG of the PP are shown in table 3.

Table 3. ATPG results for the PP excluding the embedded SRAM array part

Fault class	code	#faults
Detected	DT	104809
Possibly detected	PT	1
Undetectable	UD	5656
ATPG untestable	AU	448
Not detected	ND	144
Total faults		111058
Test coverage		99.44%
#internal patterns		266

These obviously exclude tests for the SRAM cells of both memories; however, it does include the address generation unit tests. The scan chain has a total length of 2092 elements. The ATPG untestable faults (448) are all caused by the data output port of the SRAM, which is actually connected to set/capture cells and should hence cause no testing problems. This seems to be a TetraMAX flaw.

3.3. Testable design & test of the ALU

First, it should be noted that the interconnection tests have already been performed previously. The highly reconfigurable ALU has several inputs and outputs which should be made testable at Montium tile and PPA level. At the tile level one requires a single capture cell per ALU and one set/capture cell for all ALUs (and memory units). At the PPA level one requires 90 set/capture cells and 18 capture cells for all ALUs. Only a single set/capture cell at the ALU level is required. This results in a DfT overhead of 18.1%. Internally, a single 1061-stage scan chain was constructed during scan insertion.

Table 4. Results of ATPG for the sub-core ALU

Fault class	code	#faults
Detected	DT	47198
Undetectable	UD	224
Not detected	ND	118
Total faults		47540
Test coverage		99.75%
#internal patterns		266

Table 4 shows the ATPG results for the ALU. It is stressed that several enhancements with respect to the original design like latch removal were carried out to obtain this result.

3.4. Testable design & test of the memory units

The normal procedure for testing embedded memories of sufficient size in SoCs is the usage of built-in self-testing hardware, usually based on March test algorithms and signature analyzers; these require nowadays between 1% and 2% silicon overhead. They are standard available in many CAD systems. If an on-chip processor is available, like e.g. a PP in our case, this could potentially also be accomplished by soft BIST. If the size of the memory is very small, there are two alternative options:

1) testing via external ATE suitable for algorithmic memory testing (e.g. March testing)
2) testing via a scan-based approach

As initially a memory unit was very small (M1=M2=8Kb), we have focussed on the latter approach. Our bus allows data access, while the DfT isolates the memory array from its logic (AGU) by 13 set/capture cells. Another set/capture cell is needed for one of the memory unit inputs. At the tile level we require a single set/capture cell for all memory units (and all ALUs). The resulting DfT overhead of the total memory unit is around 9%. The *logic* can be tested by a 257-stage scan chain, and the results are shown in table 5. The array as such can be tested using an α.N March test ('α' is an integer, e.g. 12, and N.N the array size), resulting in several thousand test vectors. This strongly suggests the use of internal PPs for soft BIST avoiding external communication.

Table 5. Results of ATPG for the AGU

Fault class	code	#faults
Detected	DT	7777
Undetectable	UD	69
Not detected	ND	2
Total faults		7848
Test coverage		99.97%
#internal patterns		138

From the resulting ATPG, it becomes clear that the small amount of patterns (138) for the AGU is negligible in comparison with the March tests for the SRAM cell arrays.

4. Overall DfT overhead

The DfT overheads for the sub-cores, required for this diagnostic level, have been shown. In addition an overview of what is needed at higher hierarchical levels has been given. For the entire Montium tile the DfT overhead for sub-core diagnosis amounts to 20.6%. This overhead is relatively high because of two reasons: the high diagnostic resolution and the large amount of registers in the highly reconfigurable design. Scan replacement alone poses a DfT overhead of 15.7% for the reconfigurable Montium tile.

5. Known Good Core (KGC) concept

The availability of a number of identical PP blocks in the PPA has stimulated the idea of Known Good Core (KGC). Applying input patterns to all PPs simultaneously is no issue using our bus architecture (figure 2). The idea is to first determine the correctness of a PP block, by means of e.g. conventional scan-based ATPG, using ATE or internal hardware. This is of course not always the first one to test, and in a worst-case situation this process has to be repeated five times.

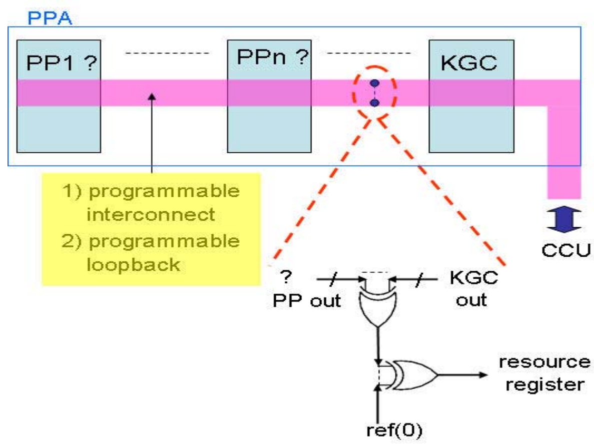

Figure 8. The DfT structure used for the KGC approach. Compare operations use programmable interconnections and loopbacks

If a correct PP has been located, its outputs can subsequently be used as correct reference for all the other blocks. However, all outputs cannot be compared simultaneously, and hence it requires the reconfigurability of the interconnections. The outputs of the first KGC are connected with XOR gates (previously tested), and subsequently the remaining cores (figure 8). The previously discussed loopback option provides the required resolution per PP. Note that the same procedure can also be applied at the sub-core diagnostic level.

Hence, the transfer of data from the chip to ATE and subsequent evaluation is not required anymore. Furthermore, it opens the road to internal self-test and self-repair.

6. Conclusions & recommendations

In this paper we have discussed the testable design and testing of a highly reconfigurable processor core for dependable data-stream applications, employing their use of repeated hardware at SoC system level and two internal levels. This simplifies the testing of PPs and sub-cores and provides interesting dependability features in combination with diagnostic BIST and self-repair via e.g. software-reconfigurable hardware. The interconnections at sub-core diagnostic level require the most DfT.

In section 3 it was mentioned that the reconfiguration registers were not taken into account for the sub-core tests, but were tested separately in advance. Each sub-core has associated reconfiguration registers, between which there is some selection logic. As a recommendation, a test partitioning can be used in which these reconfiguration registers are incorporated in the sub-cores. This will result in less DfT overhead to test all logic.

7. Acknowledgements

The authors would like to acknowledge the discussions with G. Smit of the University of Twente on reconfigurable processors, and the VHDL information provided by G. Rauwerda and P. Heysters of the company Recore Systems with regard to the Montium tile processor.

8. References

[1] "45nm Hi-k next generation Intel® Core™ microarchitecture",http://www.intel.com/technology/architecture-silicon/core/, visited on 16/8/2007.

[2] C. Taylor, "64-core processor tip of the iceberg for start-up Tilera", *Electronic News*, (2007).

[3] "Networks on Chips", *IEEE Design & Test of Computers*, 22(5), pp. 399-460 (2005).

[4] P.M. Heysters, *Coarse-grained Reconfigurable Processors*, ISBN 90-365-2076-2, Enschede (2004).

[5] P. Heysters, G. Smit and E. Molenkamp, "Montium – balancing between energy-efficiency, flexibility and performance", *Proceedings of the International Conference on Engineering of Reconfigurable Systems and Algorithms*, Las Vegas, pp. 235-241 (2003).

[6] T. Inoue, T. Fujii and H. Ichihara, "Optimal contexts for the self-test of coarse grain dynamically reconfigurable processors", *Proceedings European Test Symposium*, Freiburg, pp. 117-122 (2007).

[7] K. Katoh and H. Ito, "Built-in self-test for PEs of coarse grained dynamically reconfigurable devices", *Proceedings of European Test Symposium,* Southampton, pp. 69-74 (2006).

[8] R. Raina, "Method of testing multi-core processors and multi-core processor testing device", *US Patent 6,134,675,* (2000).