Increasing SoC Dependability via Known Good Tile NoC Testing

Hans G. Kerkhoff, Oscar J. Kuiken and Xiao Zhang[1]
Testable Design and Testing of Integrated Systems Group, CTIT,
University of Twente, Enschede, the Netherlands, h.g.kerkhoff@utwente.nl

1. Introduction

Advanced CMOS technology possibilities, power, communication and flexibility issues as well as the design gap are directing System-on-Chip (SoC) platforms towards Network-on-Chip (NoC) interconnected identical processing tiles (PT) such as the Montium processor [1]. It is broadly acknowledged that advanced technologies below 45nm come with significant yield and reliability problems, necessitating dependable designs [2]. Our approach for a dependable SoC heavily depends on the regularity within our streaming-data applications SoC. The chip consists of many identical NoC segments and identical PT's. Boundary condition is that target applications do not require all available fault-free resources, such as routing segments and PTs.

2. Yield and dependability improvement via (self) test and resources remapping

Figure 1 shows part of the architecture of our SoC. It consists of a large number of (reconfigurable) PTs including a network interface, interconnected by a NoC [1]. Bridges take care for communication with a general purpose processor (GPP) and other intellectual properties (IPs) not shown here for clarity. The 200MHz packet-switched NoC is tested first. Scan-based ATPG is carried out resulting in 98+% fault coverage. Next, the search for a fault-free PT is started (Know Good Tile, KGT, e.g. CUT2), where CUT denotes "circuit under test". For yield purposes, production test generation from automatic test equipment (ATE) can be applied [1] via the GPIO (General Purpose Input Output) and NoC to a PT (fault-coverage 99.8 %); output comparison can be carried out either via ATE or an internal test-pattern evaluation (TPE) infrastructural IP (Figure 1). This should result in a proven correctness of a PT and the compare unit. In the case of PT and/or NoC segment fault detection, diagnostic result data is stored in an on-chip non-volatile memory (NVM). Defect resources (NoC segments and/or PT's) will therefore not be available for the application resource mapping tool (SW) running on the GPP. For improving the in-field dependability, an internal fully reconfigurable test-pattern generator (TPG) infrastructural IP is used. This emulates test-generation close to the previous deterministic testing (Figure 1).

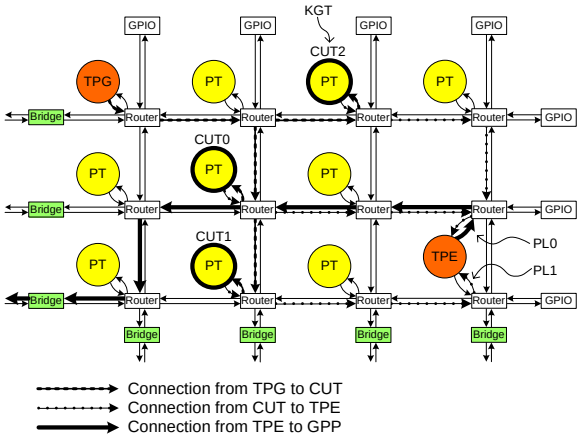

Figure 1. Simplified platform of dependable SoC

In-field dependability actions involve broadcasting test vectors (Figure 1, arrows) from TPG to KGT and other selected PTs (bold circles, CUT0 and CUT1), and subsequent comparison of results in the TPE (arrows). Finally, diagnostic result data is routed to the GPP (thick solid arrow) for on-board storage.

3. Transport of test generation and evaluation data in the NoC

The NoC is being (re-)used as test-access mechanism (TAM) for all PTs. The set-up of the test data packets of the packet-switched NoC is shown in Figure 2. In Figure 2A, the packet is shown of all test

[1] Research is partially supported under the European FP7 ICT-2007.3.4. CRISP project

patterns (TP0-TPm) from TPG to the three CUTs (Figure 1). Each block represents a flit.

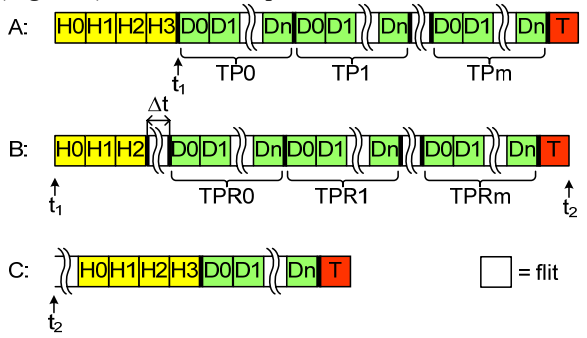

Figure 2. Structure of the NoC test-data packets

Header flits (single Hx means single router hop) are used to set up a connection, while T denotes the tail flit terminating a connection. Figure 2B shows the packet of the responses (TPR0-TPRm) of one PT to the TPE. In Figure 2C, the diagnostic result data packet is shown which is send from the TPE to the GPP.

4. The Tile Test-Pattern Evaluator

The simplified architecture of the TPE infrastructural IP is shown in Figure 3.

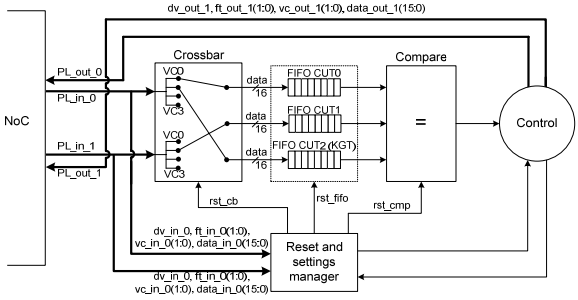

Figure 3. Tile Test-Pattern Evaluator (TPE)

The output results of the three CUTs are transported via the two physical links (PL0, PL1) from the NoC (Figures 1, 3) to the TPE. A physical link is time-multiplexed into four virtual channels (VC), each with their own priority. The crossbar connects a virtual channel to a FIFO, and the output of the KGT is subsequently compared with response data of the other chosen PTs. The comparator can be made self-checking e.g. via [2].

5. Simulation results

Figure 4 shows the VHDL simulation results of the inputs and outputs of the TPE with the KGT (CUT2) and two PTs (see also Figures 1, 3). Below the clock signal, three control signals are shown, as well as the data input coming from NoC (PL_in_0, Figure 3).

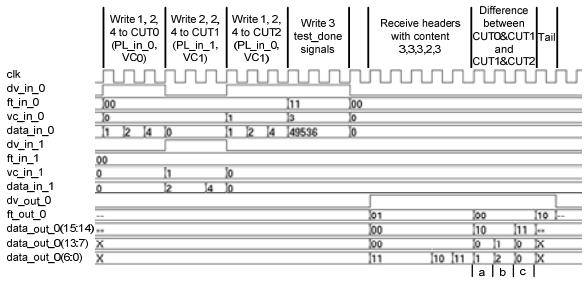

Figure 4. Simulation result of TPE for 3 PTs

The next four signals are the NoC signals of PL_in_1. Below this, only the PL_out_0 is shown for simplicity. The last three signals represent the different parts (2 bits, 7 bits, 7 bits) of this 16-bits data line. Seen from the NoC first the response (decimal equivalent 1, 2, 4) from CUT0 is written to the TPE. Next, the faulty response (2, 2, 4) from CUT1 is written and finally the correct response (1, 2, 4) from the KGT (CUT2). From the NoC, the test done signal is provided. Now, the result of the comparison has to be routed to the GPP. Hence, a connection from the TPE to the GPP has to be set up (Figure 1), using header flits (3=left, 2=down). The difference data can now be described as:

Data_d = “difference code” & CUT# & CUT#

In Figure 4, ‘a’ denotes that CUT0 and CUT1 differ, ‘b’ that CUT1 and CUT2 (KGT) differ, and ‘c’ indicates the ending of the difference description. Hence, the correct comparison has taken place.

6. Conclusions

We have shown an effective approach towards increasing the yield/dependability of a highly regular SoC with many identical PT’s, interconnected by a NoC. The implementation incorporates internal test-pattern generation as well as test-pattern evaluation infrastructural IP’s. The diagnostic results are used to remap routing paths as well as tile resources by software for a particular application in the case of faulty paths and/or tiles. The principle has been validated by simulation of the actual processors and NoC and is being implemented in a high-end FPGA.

7. References

[1] H.G. Kerkhoff and J.J.M Huijts, “Testing of a Highly Reconfigurable Processor Core for Dependable Data Streaming Applications”, *IEEE DELTA08*, Hong Kong, January 2008, pp. 38-44.

[2] G. De Micheli, “On-Chip Networks and Reliable SoC Design”, Date 07, Friday Workshop Diagnostic Services in NoC: test, debug and on-line monitoring, (web), 2007.