Fully Distributed Initialization Procedure for a 2D-Mesh NoC, Including Off Line BIST and Partial Deactivation of Faulty Components

Zhen Zhang, Alain Greiner, Mounir Benabdenbi

Fully Distributed Initialization Procedure for a 2D-Mesh NoC, Including Off-Line BIST and Partial Deactivation of Faulty Components

Zhen Zhang*, Alain Greiner* and Mounir Benabdenbi†
* *University Pierre et Marie Curie, LIP6-SoC laboratory, 4 place Jussieu, 75252 Paris, France*
{zhen.zhang, alain.greiner}@lip6.fr
† *TIMA laboratory (Grenoble INP, CNRS, UJF), 46 avenue Felix Viallet, 38000 Grenoble, France*
mounir.benabdenbi@imag.fr

***Abstract*—In this paper, we present an embedded, at speed, off-line, and fully distributed initialization procedure for 2D-Mesh Network-on-Chip (NoC). This procedure is executed at power boot, and targets the detection and the deactivation of the faulty routers and/or faulty communication channels. The final objective is fault tolerance. The proposed procedure is able to clean the NoC from all destructive malfunctions induced by permanent hardware failures. This initialization procedure has been implemented in a reconfigurable version of the DSPIN micro-network, and evaluated from the point of view of Stuck-at fault coverage, and area overhead.**

I. Introduction

According to the industrial forecast [1], Multi-Processor System-on-Chips (MPSoCs) will integrate 1000s of processors interconnected by a Network-on-Chip (NoC), but with a high failure rate. Thus, the NoC fault tolerance must be taken into account in the early design stage.

In a previous paper [2], we proposed a 2D-Mesh NoC self-reconfiguration strategy and a reconfigurable routing algorithm. This reconfiguration procedure can be used as long as the faulty components (routers or point to point communication channels) have been tested and identified.

In this paper, we propose to address both the off-line test of the NoC, and a fully distributed initialization procedure, including detection and deactivation of the faulty routers and the faulty communication channels. This procedure relies on a distributed, scalable, at-speed, built-in self test (BIST) hardware support, and is systematically executed at power-on. This initialization procedure can be executed off-line when the chip is embedded in its functional environment. The fault coverage of this BIST has been evaluated using the Stuck-at fault model (SAF).

In fact, many papers [3]–[5] have described various techniques to test a NoC and to locate faulty components. However, they are not sufficient for NoC fault tolerance: the whole network must be cleaned, and the faulty components deactivated in such a way that a failing component will not prevent the test/diagnosis/initialization process. To our knowledge, this key issue has not been addressed yet, and a solution is described in this paper.

II. A typical 2D-Mesh NoC architecture: DSPIN

The 2D-Mesh NoC used in our study is the DSPIN [6] (Distributed Scalable Predictable Interconnect Network). It was designed by the LIP6 laboratory and was physically implemented by ST Microelectronics. As shown in Fig.1.{A}, the DSPIN router (inner) is composed of 5 modules (North, East, South, West & Local) interlinked as a full crossbar. In order to support the GALS approach, the adjacent borders of two neighboring routers are connected by two FIFOs: one synchronous, one bi-synchronous [7] for clock boundary traversal. These two FIFOs constitute a point to point communication channel (called a channel), as shown in Fig.1.{B}.

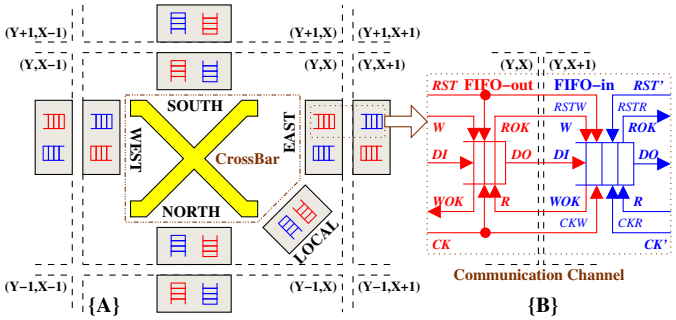

Figure 1. {A} presents a generic DSPIN router architecture; {B} presents a generic communication channel.

Once a router or a channel is corrupted by a stuck-at fault, the malfunctions will prevent the reuse of the NoC itself for test and reconfiguration purpose. For example, in case of SAF1 injected on signal W of FIFO-out, when the channel isn't full, it stores any DI as a packet flit, even if it is not valid. That leads to self-generating fake packets which will quickly fill the channel, eventually block the whole NoC.

In order to avoid such destructive behaviors, faulty routers and faulty channels must be deactivated as soon as they have been detected as faulty. This test and deactivation mechanism must be totally distributed, because it must be done locally, for each router and each channel, when the NoC is not yet operating.

III. The distributed initialization procedure

A. General principle

The proposed initialization procedure is fully decentralized, and is implemented by a set of dedicated FSMs (Finite State Machine) located in each router as shown in Fig.2.

- There is one set of FSMs in each router, and the boot procedure is executed in parallel in all routers.
- A master FSM: ATC (Auto Test Center) tests and boots the router itself.

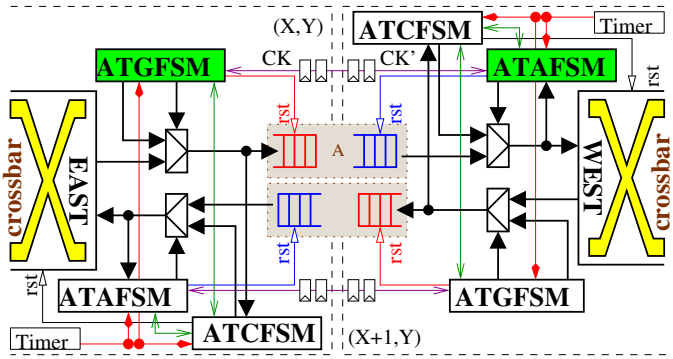

Figure 2. The initialization procedure implementation.

- Several slave FSMs: ATG (Auto Test Generator, one per output channel) and ATA (Auto Test Analyser, one per input channel) test and boot the channels.
- Each ATG cooperates with the corresponding ATA in the neighboring router to test the bi-synchronous channel. As such two FSMs belong to different clock domains, they communicate asynchronously through a limited number of handshaking signals, thanks to re-synchronization flip-flops.
- The multiplexers controlled by ATG and ATA offer 2 functions: to isolate router test and channel test so as to avoid failures propagation; to deactivate a faulty channel.
- A timeout is attached to the initialization procedure, in case one FSM is blocked in an intermediate state.

These FSMs are activated by the global RESET signal, to execute the algorithm described in Fig.3. There is two level of parallelism in this distributed algorithm:

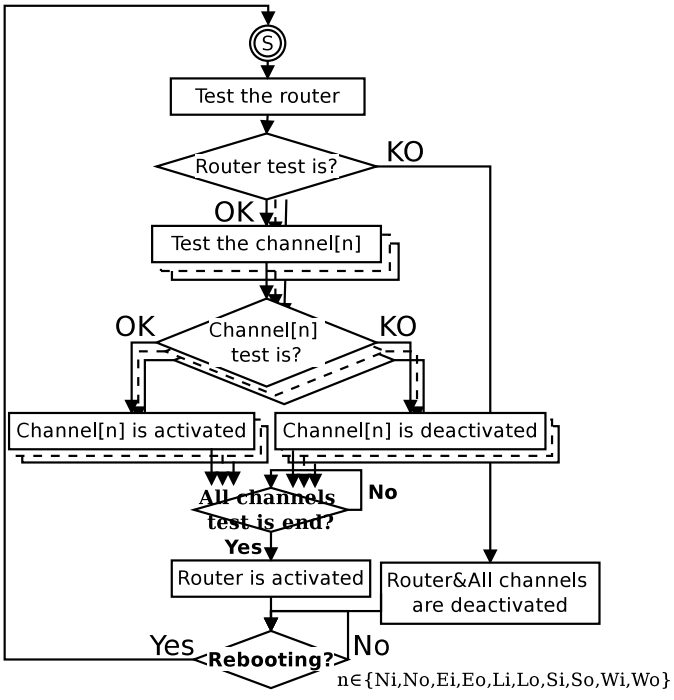

Figure 3. The initialization procedure.

–The router is tested first, without any interaction with the neighbor routers. Then the ten communication channels associated to a router are tested in parallel.

–If the test of a router is KO, the router is considered as faulty, and all input and output channels are deactivated. If the test of a given channel is KO this channel is deactivated.

It should be noted that, a deactivated channel is configured to behave as a "Black Hole". It discards any incoming data, and produces no outgoing data.

B. ATC

As shown in Fig.4. the test is split in two phases: the router test (all inner component, such as crossbar) and the channel test.

(1) ATC uses the CMDRTT and ACKRTT states to check the communications with the local ATGs&ATAs through the cmd/ack signals CMD&END, and to send the proper commands to local ATGs&ATAs, that are in charge of controlling the multiplexers.

(2) ATC uses the RTTEST states to test the router. In each state, ATC generates one pattern to router input and analyzes the router output.

(3) If the router test is ok, ATC uses the CMDCHT state to ask local ATAs&ATGs to start the test of the communication channels, and goes to ACKCHT wait state. The router is deactivated by ATC during this phase.

(4) When this phase is completed, ATC uses the final state FUNC to activate the router.

(5) If ATC enters the final state ERRATC, the router and all channels are deactivated. A time-out mechanism is implemented in each ATC to force the completion of the channel test phase, and to force the ATC FSM in a well defined state.

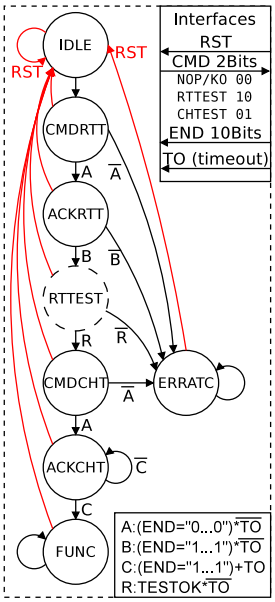

Figure 4. The ATC FSM.

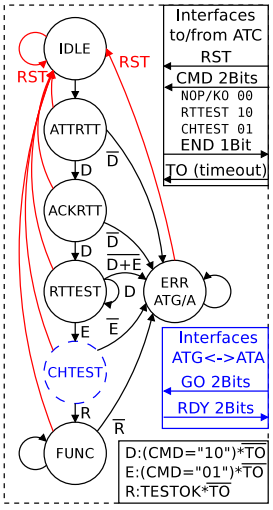

Figure 5. The ATG/A FSM.

C. ATG and ATA

As the ATG and ATA are very similar, we use only one figure (Fig.5) to present ATG/ATA FSMs.

(1) In state ATTRTT, ATG/A checks the CMD signal from ATC. If the command is "RTTEST", ATG/A uses state ACK-RTT to send an acknowledge to ATC through the signal END. It controls the multiplexers to fit the router test.

(2) During the router test, ATG/A uses state RTTEST to check the result. If result is ok, ATG/A receives the "CHTEST" command, and begins the channel test. The details of the channel test will be presented in the next subsection.

(3) Finally, ATG/A enters the FUNC state to activate the channel, or the ERRATG/A state to deactivate the channel. Whatever the end state is, ATG/A sends acknowledge to ATC with the signal END.

Here again we use the timeout approach to reach a well defined state in case the channel test phase does not complete.

D. Channel test

As one ATG/ATA couple works cooperatively in two different clock domains, the CHTEST transitions are driven by the asynchronous handshaking signals "READY" and "GO", like playing pingpong, as shown in Fig.6. For example, the **Round (W0/R0)** : ATG fills the channel with the test patterns and then sends "READY" to ATA; then ATA reads&analyzes the data from the channel and then returns "GO" to ATG. That's it, round-by-round, until the end of test, the channel is considered as "OK". Otherwise, once any one round is broken or timeout, the game is over, the channel is considerd as "KO". It should be noted that, ATG always serves at first in a round.

Round (ACT) ATG activates FIFO-out in state ACT and checks no WOK/SAF0 in state ACTT. If it's OK, ATG sends "READY" in state RDYACT. ATA waits for "READY" in state GOACT. Once "READY" is received, ATA activates FIFO-in in state ACT and checks no ROK/SAF1 in state ACTT. If it's OK, ATA returns "GO" in state GONOP. Once "GO" is received by ATG, the test enters into next round.

Round (NOP) In this round, ATG dose NOP test (do nothing) and ATA verifies again the signal ROK (no self-generating fake packet).

Round (W0/R0) In this round, ATG fills the channel with N test patterns (N=depth of channel) using N states. And then, ATA uses N states to read the data and analyze the results. If it's OK, the ATG/A couple continues the following round W1/R1,...,WM/RM.

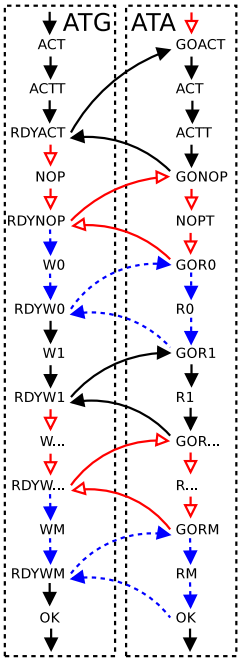

Figure 6. The ATG/ATA CHTEST states

E. Timout

The timeout value is determined by the ratios between the various clock frequencies involved in the NoC. If all clocks have the same frequencies (but different phases), the initialization procedure requires less than 300 clock cycles (in our case study). If we assume that the biggest ratio between two clock frequencies in two neighboring routers is equal to 1000, the upper bound timeout value is equal to $3 \times 10^5 = 300 \times 1000$ local clock cycles.

IV. Experimental Results

The DSPIN router simulated has five input/output ports, the input/output FIFOs have a 4 words depth and 37 bits width.

A VHDL RTL model containing a complete bi-synchronous channel has been simulated with 5 couples of clocks, which prove that the distributed test procedure is robust to large variations of clock frequencies.

CK of ATG	5000ns	7ns	5ns	5ns	5ns
CK of ATA	5ns	5ns	5ns	7ns	5000ns

The fault coverage of a channel, a router (inner), and a complete self testable router (with BIST component) were evaluated separately using the Synopsys Tetramax tool.

Components	Fault Coverage
A Channel	98.35%
A Router (inner)	98.38%
A Self Testable Router	91.07%

The overhead of the silicon area is evaluated in the Synopsys synthesis environment. As DSPIN is a very compact design and represents typically less than 3% [6] of the silicon area in a typical MPSoC, 45% overhead is an affordable cost.

Component	NAND	%
Original DSPIN Router	10153.37	100%
Self Testable Router	14696.65	144.74%
Overhead	4543.28	44.74%

V. Conclusion

In this paper, we presented a fully distributed initialization procedure for a 2D-Mesh NoC in a GALS context. This approach supports automatic detection and deactivation of the faulty routers or communication channels. It relies on a fully parallel and distributed BIST mechanism providing a global fault coverage over 91%, for an acceptable silicon area cost. This distributed test and deactivation mechanism is a key building block to define a truly reconfigurable, fault-tolerant Network on Chip. The presented initialization procedure can be performed at fabrication time or when the chip is embedded in its functional environment. In the first case, this procedure can help improving the yield by avoiding to throw the whole chip when one single component is faulty. In the second case, it can help avoiding a global failure of the overall system through a simple reboot of the chip and thus leverage fault tolerance.

References

[1] International Technology Roadmap for Semiconductors. [Online]. Available: http://www.itrs.net

[2] Z. Zhang, A. Greiner, and S. Taktak, "A reconfigurable routing algorithm for a fault-tolerant 2D-Mesh Network-on-Chip," in *Proc. of DAC'08, Design Automation Conference*, 2008.

[3] A. Amory, E. Briao, E. Cota, M. Lubaszewski, and F. Moraes, "A scalable test strategy for network-on-chip routers," in *Proc. of ITC'05, International Test Conference*, 2005.

[4] K. Petersén and J. Öberg, "Toward a scalable test methodology for 2d-mesh network-on-chips," in *Proc. of DATE'07, Design, Automation and Test in Europe*, 2007.

[5] M. Herve, E. Cota, F. L. Kastensmidt, and M. Lubaszewski, "Diagnosis of interconnect shorts in mesh nocs," in *Proc. of NOCS '09: International Symposium on Networks-on-Chip*, 2009.

[6] I. Miro-Panades, F. Clermidy, P. Vivet, and A. Greiner, "Physical implementation of the dspin network-on-chip in the faust architecture," in *Proc. of NoCS'08, International Symposium on Networks-on-Chip*, 2008.

[7] I. Miro Panades and A. Greiner, "Bi-synchronous fifo for synchronous circuit communication well suited for network-on-chip in gals architectures," in *Proc. of NOCS'07, International Symposium on Networks-on-Chip*, 2007.