An Effective Solution to Thermal-Aware Test Scheduling on Network-on-Chip Using Multiple Clock Rates

Hassan Salamy
Ingram School of Engineering
Texas State University
San Marcos, Texas, USA

Haidar Harmanani
Department of Computer Science
Lebanese American University
Byblos 1401 2010, Lebanon

***Abstract*—As more cores are being packed on a single chip, bus-based communication is suffering from bandwidth and scalability issues. As a result, the new approach is to use a network-on-chip (NoC) as the main communication platform on a SoC. NoC provides the flexibility and scalability much needed in the era of multi-cores. NoC-based systems also provide the capability of multiple clocking that is widely used in many SoC nowadays. In this paper, a simulated annealing algorithm for thermal and power-aware test scheduling of cores in a NoC-based SoC using multiple clock rates is presented. Results on different benchmarks show the effectiveness of our technique.**

I. Introduction

Bus-based communication is unable to keep up with the increasing complexity of system-on-chip in the multi-core era; it has been argued that there is a need to meet the high communication requirements of large systems-on-chip (SoCs) using a scalable communication methodology. NoC provides several advantages including *modularity*, *higher performance*, *better structure*, and *compatibility with core designs and reuse* [1]. Testing embedded cores is a major bottleneck that involves the reuse of the existing on-chip communication as a test access mechanism where test stimuli and responses are transmitted through the network. Such systems usually use *packet-based routing* algorithms where the test vectors are embedded inside packets. A major challenge in testing embedded cores is test scheduling which determines the order in which various cores are tested. Test scheduling for SoC, even for a simple SoC, is equivalent to the NP-complete *m*-processor open shop scheduling problem [2]. Obviously, minimal test time would be achieved by maximizing the simultaneous test of cores; however, design constraints, such as power consumption, may prevent this full parallelism. However, power constraints do not necessarily achieve thermal safety of each core in the system. Thermal-constrained test scheduling is therefore essential in order to ensure that the maximum thermal safety temperature is not exceeded. Many NoC have the flexibly of multiple clock rates. Not all the cores have to be tested using the same clock rate. Increasing clock rates increases the power consumption of the system cores.

This paper presents a method for test scheduling subject to power and thermal constraints using multiple clock rates. The objective is to minimize the overall test time by properly allocating input and output ports in the NoC to cores in addition to assigning different clock rates to the system's cores. We present a thermal-aware simulated annealing algorithm to solve this problem. The remainder of the paper is organized as follows. Section II presents related work in this area while section III provides a brief overview of NoC. Section IV deals with test scheduling using multiple clock rates, and presents our thermal-aware simulated annealing solution to the test scheduling problem in a NoC using multiple clock rates. Section V summarizes the results. Finally, Section VI presents our conclusions.

II. Related Work

The general concept for NoC testing was first introduced by Vermeulen *et al.* [3] where the communication NoC infrastructure is typically tested before the built-in core test. System's resources are tested next using the NoC as a TAM. Nahvi *et al.* [4] proposed network-oriented test architecture to utilize NoC for testability. Test scheduling was tackled based on two main approaches, namely *core-based* and *packet-based* test scheduling. The basic idea in core-based scheduling is to determine the test order of each cores so that the overall test time is minimized under different constraints [5]. On the other hand, packet-based scheduling cores are tested using test packets. The basic idea of such scheduling algorithms is to determine the order of generation and transmission of test packets for cores. The use of packet switching architecture to test cores is proposed in [6] and [7]. Cota et al. [8] proposed test scheduling based on a packet-switching protocol. Embedded processors have been used for test sources and sinks to increase test parallelism [9]. In [10], multiple test clocks is also used to reduce overall schedule time of NoC-based SoCs.

Power-aware and thermal-aware test scheduling for NoC-based systems have been studied by many researchers [11], [5], [12], [13], [14]. However, it has been shown that power aware constraints do not guarantee thermal safety [15], [16]. In [16], a fast heuristic is used that guarantees thermal safety. In [17], variable test clock frequencies are used to control the power consumption by each core so that the thermal safety

978-1-4673-2527-1/12/$31.00 ©2012 IEEE

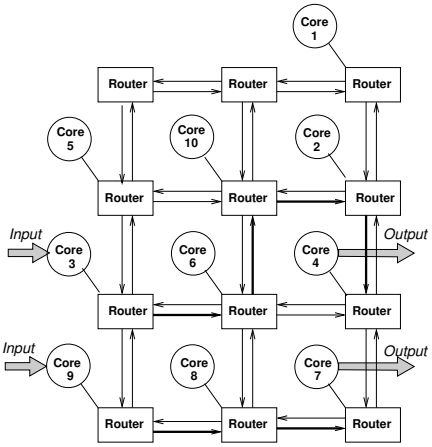

Fig. 1. NoC-based d695 with two routing paths scheduled.

is guaranteed. In [15], layout information and progressive weighting are used to reduce hot spot temperatures. Liu et al. [18] presented two heuristics to achieve thermal optimization based on multiple clock frequencies.

III. NoC Test Scheduling

NoC usually uses a 2-D mesh topology where each router is connected to the four neighboring routers. The communication channels are assumed to be 32-bit width and bidirectional. Switching is usually based on wormhole technique. Each packet is divided into flow control units (flits) where a flit is the smallest unit over which flow control is performed. The flit size is usually equal the channel width. NoCs usually use a message passing communication model where information is sent via packets. A packet is composed of a header, a payload, and a trailer. Test vectors are delivered through packets. Packets are routed from an input port to the core under test and then the results are routed to an output port.

There are two different approaches to go for the test core scheduling namely, *preemptive* and *non-preemptive*. Our approach is based on a non-preemptive core-based approach where a path is dedicated to a core from the beginning of the test till the end. In such a case, the test pipeline of the core under test is not interrupted as this path is always available for test vectors and test responses. The dedicated routing path usually consists of an input port, an output port, and corresponding channels that transport test vectors to the core and transport the test responses from the core. Figure 1 shows two routing path schedules for cores 8 and 10 for the *d695* example benchmark using two input and two outports. Each path will be dedicated to the corresponding core throughout the testing process. The input/output ports in our example are implemented using the functional ports on cores $3, 4, 7$, and 9.

IV. Test Scheduling Using Multiple Clock Rates

A. Problem Formulation

This paper propose the use of multiple clock rates for thermal management and test time minimization. One way to reduce the hot spots is to use a slower clock rate. Using a single slow clock rate is usually inefficient as it may result in a higher testing time for the cores. Hence, a single slower clock adversely affects test time and increases cost. Although slower clock rates can bring down the hot spots, it is not efficient in achieving thermal balance across the chip. As a result, we propose to use multiple clock rates instead of one slow clock. Generally speaking, a slow clock is assigned to hot spots to cool them down. However, a slow clock rate usually means a higher test time. The problem is then to find the best clock rates for different spots and a test schedule to minimize the test time of all the cores such that the core temperatures are below a certain threshold for thermal safety.

In NoC, some cores cannot utilize the entire width of the network channels. Those idle channels can be used to transport more test data and increase the clock rate for that specific core. The general idea is to use slower clock for high spot cores and faster clocks to cool cores so that the thermal safety constraints are met and the overall test time is minimized. To simplify the problem, we assume that there are $2n+1$ different on-chip clock frequencies. For example, assume an on-chip clock rate of f and n equals *three* there are a total of 7 clock frequencies, namely, $f/4$, $f/3$, $f/2$, f, $2f$, $3f$, $4f$. The problem is now to assign the on-chip clocking rates to different cores under test in order to minimize the overall test time while not exceeding the maximum core temperature allowed for safety issues.

B. Simulated Annealing Algorithm

The key elements in implementing the annealing algorithm are: 1) the configuration representation, 2) the definition of a neighborhood on the configuration space, 3) a cost function, and 4) a cooling schedule. In what follows, we describe our thermal and power aware simulated annealing (SA) solution to NOC test scheduling with multiple clock rates.

1) Configuration Representation: We represent our solution using the configuration shown in Figure 2 (a). The configuration shows four cores, two I/O ports, two possible clock rates, f_1 and f_2, and the corresponding schedule.

2) Initial Solution: To ensure feasibility, the initial solution is chosen such that all cores are assigned to the same I/O.

3) Neighborhood Transformations: We explore the solution space based on three neighborhood transformations that we present next in reference to Figure 2. At each annealing iteration, the algorithm randomly selects one operation.

(a) *Change the clock of one core:* Change the clock of a randomly selected core to one of the clock rates available in the system. A different clock rate may imply a different test time and power consumption.
(b) *Change the I/O of one core:* Change the input/output ports of a randomly selected core to one of the input/output ports in the NoC.
(c) *Change the Position of One Core:* Change the starting test time of a randomly selected core while keeping its I/O assignments.
(d) *Swap the I/O of two cores:* Exchange the input/output ports of two randomly selected cores.

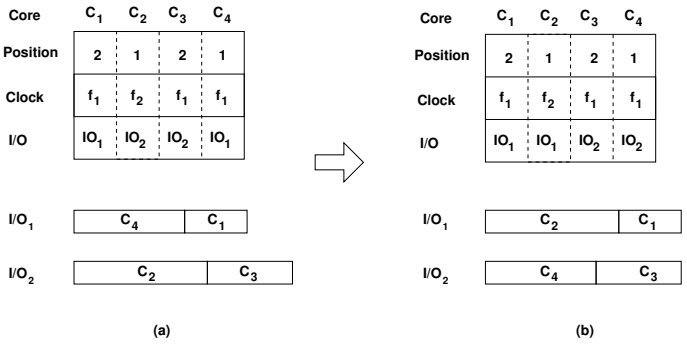

Fig. 2. (a) An example SA solution and its test schedule. (b) The new solution and its test schedule.

(e) *Swap the Positions of two Cores:* Exchanges the position of two randomly selected cores that are assigned to the same input/output ports.

4) Cost Function and Cooling Schedule: Given a test schedule, the cost is the maximum end time for the tests of all cores. The cooling schedule is empirically determined as follows: the initial temperature, T_{init}, is set to 4000; the temperature reduction ration, K, is set to 0.99, and the number of iterations, M, is set to 5. The algorithm stops when the temperature, T, is below 0.001.

V. Results

We have implemented the proposed NoC test scheduling algorithm and attempted the following ITC'02 benchmarks: $d695$, $g1023$, $p22810$, and $p34392$ citeMIC02. We created a hypothetical layout for each NoC system. As the power information of the core in these benchmarks are not available, approximation values based on the number of scan flip-flops of each core is used based on:

$$Power = C_L \cdot V_{dd}^2 \cdot \frac{1}{T} \cdot [(\sigma_{ff} + 1) \cdot nb_{ff} + \sigma_{gt} \cdot nb_{gt}] \qquad (1)$$

Each core defined in Equation 1 is adjusted by replacing n_{ff} by the number of scan flip-flops and n_{gt} by an estimated number of gates for the module. In Equation 1, C_L is the load capacitance, T is the clock period, and σ is the switching activity factor. Since it is hard to approximate the power consumption by the routers we only considered power consumption by the cores. We also compute the cores temperature using thermal simulation, based on a computationally efficient thermal modeling tool, *HotSpot* [19]. HotSpot accounts for heat dissipation within each functional block as well as the heat flow among different blocks.

The simulated annealing was first tested without any power or temperature constraints under the assumption of a single clock rate. Results are shown in Table I. The channel width is assumed to be 32. We compared our results to those reported in [5]. Then, we repeated the experiments using multiple clock rates. We assume that there are three clock rates $f/2$, f, $2f$. The test times for channel width of 32 are shown in Table II. The results clearly show the importance of using multiple clock rates especially to decrease the testing time of cores with long test time. On the other hand, faster clocks imply higher power consumption and consequently higher temperature. We also compared our results to those in [5]. Our method improved over the results in [5] in all cases.

We next tested our method based on stringent power con-

Thermal-Aware SA_Test_Scheduling()
- N_{iter} = The number of Iteration.
- $Time_{stop}$ = Stopping temperature .
- K = Temperature reduction ratio.
- S_{init} = Initial Solution.
- $T_{thermal\_safety}$ = Maximum allowable temperature.
- P_{max} = Maximum allowable power consumption at the same time.
- Boolean $Power$, $Temperature$.
- Build S_{init}.
- Invoke *HOTSPOT(Power File, Floorplan File).*
- $Cost(S_{init})$ = Test time for S_{init}.
- Current solution $S_{curr} = S_{init}$.
- **While** ($Time_{curr} > Time_{stop}$)
  - **For** (i = 1 to N_{iter}) **do**
    - Generate a neighboring solution $S_{new} = Neighbour\ (S_{curr})$
    - Evaluate S_{new}.
    - **If** (S_{new} is feasible) **then**
      - **If** ($Power$)
        - **For** each core i **do**
          - Find all cores j such that.
          - $t_j \leq t_i$ AND $t_j + l_j > t_j$ /*Overlap .
          - Calculate the sum of power consumption of such cores.
          - Find maximum power consumption,P, of all overlapping cores.
          - **If** $P > P_{max}$ **then** Violation.
      - **If** ($Temperature$)
        - Update temperature file based on current configuration.
        - Find core i with maximum temperature T.
        - **If** $T > T_{thermal\_safety}$ **then** Violation.
      - **If** (No Violation) **then**
        - Find test time $Cost(S_{New})$
        - Compute Δ_{Cost} = $CostS_{new}$ - $CostS_{curr}$
        - **If** (Δ_{Cost}) < 0) **then**
          - $S_{curr} = S_{new}$.
        - **Else**
          - Generate random number R, $0 < R < 1$.
          - **If** $R < exp(-\Delta_{Cost}/T_{curr})$ **then**
            - $S_{curr} = S_{new}$.
    - Update *SA parameters* .

Fig. 3. Thermal-Aware Simulated Annealing for Test Scheduling.

TABLE I
Test times using single clock.

Benchmark	I/O	Single Clock ([5])
d695	2/2	13227 (15510)
g1023	3/3	14794 (17925)
p22810	3/3	112793 (141594)
p34392	3/3	544579 (563795)

TABLE II
TEST TIMES USING MULTIPLE CLOCK RATES.

Benchmark	I/O	Multiple Clocks ([5])
d695	2/2	10905 (13336)
g1023	3/3	7397 (10061)
p22810	3/3	92554 (102595)
p34392	3/3	275409 (303364)

TABLE III
TEST TIMES WITH POWER CONSTRAINTS AND MULTIPLE CLOCKS, W = 16

Benchmark	I/O	Power-aware ([5])
d695	2/2	26234 (26683)
g1023	3/3	25682 (29076)
p22810	3/3	227945 (248287)
p34392	3/3	1422927 (1422927)

straint under the scenario of multiple clock rates where we set the power limit at a certain point not to exceed 25% of the total power consumption of all the cores. Table III shows the testing time for different benchmarks using 25% stringent power constraint for W=16. Results from our power-aware techniques assuming a channel width of 32 are presented in Table IV. Our techniques improved over the results in [5] in most cases. Our power-aware single clock approach improved up to 11.67% whereas our power-aware multiple clock technique reduced test time over [5] by 0.6% to 23.4%. We also calculated the temperature of hot spots using 25% stringent power constraint. Although the power constraint is very stringent, the hot spot temperature was still way over the acceptable thermal safe temperature of 127 °C (Column 3 of Table IV). Decreasing the maximum allowable power further will cause a significant increase in test time and will render some benchmarks *unfeasible*.

It is clear from Tables IV that power consumption constraints are not sufficient to keep the chip safe as the temperature of some cores exceeds the highest allowable temperature, even though the overall power is still less than the maximum allowable power. In this part, we test our SA with temperature constraint where we set T_{max} to 127°C. We first set up the parameters for thermal simulation for the *Hotspot* tool such as layers, thermal resistance, thickness of layers and materials, and the chip dimensions. The testing time of our thermal safe schedule is presented in Table V for channel width size of 32. Our thermal-aware SA solution was able not only to ensure thermal safety but also to decrease testing time compared to power constraints solution. On average, our temperature-aware technique improved over those reported in [18] by 6%.

TABLE IV
TEST TIMES WITH POWER CONSTRAINTS AND MULTIPLE CLOCKS, W = 32

Benchmark	I/O	Power-aware ([5])	Max Temp.
d695	2/2	18786 (20040)	136
g1023	3/3	16749 (21873)	203
p22810	3/3	142213 (160689)	227
p34392	3/3	781834 (786274)	214

TABLE V
TEST TIMES USING THERMAL CONSTRAINTS.

Benchmark	I/O	Temperature-aware ([18])
d695	2/2	13227 (13547)
g1023	3/3	16473 (18368)
p22810	3/3	112550 (150714)
p34392	3/3	769284 (785488)

VI. CONCLUSION

We presented a simulated annealing solution to thermal safety test scheduling using multiple clock rates. We showed that stringent power constraints are not sufficient for thermal safety. Results on different benchmarks show the importance of our technique especially as the number of cores built on a single chip is continuously increasing.

REFERENCES

[1] S. Murali and G. D. Micheli, "Bandwidth-constrained mapping of cores Onto NoC architectures," in *Proc. DATE*, 2004, pp. 896–901.

[2] K. Chakrabarty, "Test Scheduling for Core-Based Systems Using Mixed-Integer Linear Programming," *IEEE Trans. on CAD*, vol. 19, no. 10, pp. 1163–1174, 2000.

[3] B. Vermeulen, J. Dielissen, K. Goossens, and C. Ciordas, "Bringing Communication Networks on a Chip: Test and Verification Implications," *IEEE Communication Magazine*, vol. 41, no. 10, pp. 74–81, 2003.

[4] M. Nahvi and A. Ivanov, "Indirect Test Architecture for SoC Testing," *IEEE Trans. on CAD*, vol. 23, no. 7, pp. 1128–1142, 2004.

[5] C. Liu, V. Iyengar, J. Shi, and E. Cota, "Power-Aware Test Scheduling in Network-on-Chip Using Variable-Rate On-Chip Clocking," in *Proc. VTS*, 2005, pp. 349–354.

[6] M. Nahvi and A. Ivanov, "A Packet Switching Communication-Based Test Access Mechanism for System Chips," in *Proc. European Test Workshop*, May 2001, pp. 81 – 86.

[7] C. Aktouf, "A Complete Strategy for Testing an On-Chip Multiprocessor Architecture," *IEEE Design & Test of Computers*, vol. 19, no. 1, pp. 18–28, 2002.

[8] E. Cota, L. Carro, F. Wagner, and M. Lubaszewski, "Power-Aware NoC Reuse on the Testing of Core-Based Systems," in *Proc. ITC*, 2003, pp. 612–621.

[9] A. M. Amory, E. Cota, F. Wagner, L. Carro, M. Lubaszewski, and F. G. Moraes, "Reducing Test Time with Processor Reuse in Network-on-Chip Based System," in *Proc. Symp. on ICSD*, 2004, pp. 111–116.

[10] J. Ahn and S. Kang, "Test Scheduling of NoC-Based SoCs Using Multiple Test Clocks," *ETRI Journal*, vol. 28, pp. 475–485, 2006.

[11] E. Cota, L. Carro, and M. Lubaszewski, "Reusing an On-Chip Network for the Test of Core-Based Systems," *ACM Trans. on Design Automation of Electronic Systems*, vol. 18, no. 4, pp. 471–499, October 2004.

[12] Z. He, Z. Peng, and P. Eles, "Multi-Temperature Testing for Core-Based System-on-Chip," in *Proc. DATE*, 2010, pp. 208–213.

[13] C. Yao, K. K. Saluja, and P. Ramanathan, "TThermal-Aware Test Scheduling Using on-Chip Temperature Sensors," in *Proc. VLSI Design*, 2011, pp. 376–381.

[14] N. Aghaee, Z. He, Z. Peng, and P. Eles, "Temperature-Aware SoC Test Scheduling Considering Inter-Chip Process Variation," in *Proc. ATS*, 2010, pp. 395 – 398.

[15] C. Liu, K. Veeraraghavan, and V. Iyengar, "Thermal-Aware Test Scheduling and Hot Spot Temperature Minimization for Core-Based Systems," in *Proc. ISDFT in VLSI*, 2005, pp. 552 – 560.

[16] P. Rosinger, B. Al-Hashimi, and K. Chakrabarty, "Rapid Generation of Thermal-Safe Test Schedules," in *Proc. DATE*, 2005, pp. 840–845.

[17] E. Tafaj, P. Rosinger, B. Al-Hashimi, and K. Chakrabarty, "Improving Thermal-Safe Test Scheduling for Core Based System-on-Chip Using Shift Frequency Scaling," in *Proc. ISDFT in VLSI*, 2005, pp. 544–551.

[18] C. Liu and V. Iyengar, "Test Scheduling with Thermal Optimization for Network-on-Chip Systems Using Variabe-Rate On-Chip Clocking," in *Proc. DATE*, 2006.

[19] W. Huang, S. Ghosh, K. Sankaranarayanan, K. Skadron, and M. R. Stan, "HotSpot: a Compact Thermal Modeling Methodology for Early-State VLSI Design," *IEEE Trans. on VLSI*, vol. 14, no. 5, pp. 501–513, 2006.