Evaluating Energy Consumption of Homogeneous MPSoCs using Spare Tiles

Alexandre M. Amory, Luciano C. Ost,
César A. M. Marcon, Fernando G. Moraes
FACIN - PUCRS
Porto Alegre, Brazil
{alexandre.amory, luciano.ost,
cesar.marcon, fernando.moraes}@pucrs.br

Marcelo S. Lubaszewski
PPGC - Instituto de Informática - UFRGS
Porto Alegre, Brazil
luba@eletro.ufrgs.br

Abstract— **The yield of homogeneous network-on-chip based multi-processor chips can be improved with the addition of spare tiles. However, the impact of this reliability approach on the chip energy consumption is not documented. For instance, in a homogeneous MPSoC, application tasks can be placed onto any tile of a defect-free chip. On the other hand, a chip with defective tile needs a special task placement, where the faulty tile is avoided. This paper presents a task placement tool and the evaluation of energy consumption of homogeneous NoC-based MPSoCs with spare tiles. Results show NoC energy consumption overhead ranging from 1 to 10% when considering up to three faults randomly distributed over the tiles of a 3x4 mesh network. The results also indicate that faults on the central tiles typically have more impact on energy overhead.** *(Abstract)*

Keywords-component; network-on-chip, homogeneous MPSoCs, reliability estimation.

I. Introduction

NoCs can consume more than one third of the total chip energy consumption of the system [1]. In addition, the required reliability of NoC is becoming harder to achieve due to shrinking feature-sizes and supply voltage scaling [2], which increases the defect rate and reduces the yield.

High reliability and low energy consumption are conflicting design goals, thus both have to be *jointly evaluated* to optimize a NoC design. Bertozzi et al. [3] evaluate the reliability and energy consumption of NoC links, which considers the energy efficiency of two link-level error recovery schemes: correction at the receiver stage versus retransmission of corrupted data. Ejlali et al. [4] estimate performance and reliability of NoCs, evaluating the impact of voltage swing and different error control schemes on the performance/energy trade-off.

While [3] and [4] address the problem of energy/reliability on-chip communication at link level, Manolache et al. [5] address the problem at application level. They assume that 100% reliable communication cannot be achieved in the presence of transient failures, thus they propose a way to combine spatial and temporal redundant message transmission where energy and latency overhead are minimized.

These previous papers address different aspects of energy/reliability co-optimization on NoC-based multiprocessors SoC (MPSoCs) considering *transient faults*. Nevertheless, few papers address the same problem for *manufacturing faults,* where fault tolerance approaches can play an important role to improve economical viability of such complex systems production by increasing the yield.

Redundant hardware is commonly used to tackle the yield problem. It has been successfully applied to all sorts of regular and repetitive hardware, like different types of memories, programmable logic array, field programmable gate array, and recently to MPSoC with homogeneous processor elements.

Shamshiri and Cheng [2] proposed a yield and cost analysis framework used to evaluate the use of spare tiles in an MPSoC. It determines the amount of redundancy required to achieve the minimal cost. For instance, given some input parameters (detailed in [2]), the yield of a block is 94% and the NoC link is 72%, resulting in a system yield of just 21% for a 3x3 2D mesh NoC, i.e. there is a probability of 79% of having at least one faulty block in the system. By including three spare tiles into the system, increasing the number of tiles from 9 to 12, the system yield increases to 99% since only 9 out of 12 tiles are actually required to have a functional system (this redundancy approach based on spare tiles is also called *m*-out-of-*n*). Moreover, the manufacturing cost is 3.2 times less than the original system, demonstrating that the increased yield compensates the additional silicon area of the spare tiles.

Given these promising results, we decided to further investigate the use of spare tiles to improve the cost and yield of NoC-based MPSoCs. Considering manufacturing faults of NoC tiles, the *goal of this paper* is to evaluate the effect of the spare tile approach on the energy consumption efficiency of the system. In this context, this paper extends the CAFES task placement framework [6] in order to create a valid task placement considering the location of faulty tiles and to evaluate the energy overhead compared to a fault-free system. Although Shamshiri and Cheng [2] evaluated the reliability of systems with spare parts, as far as we know, this is the first paper to evaluate the impact of spare tiles on energy consumption of NoC-based MPSoCs.

II. Proposed Manufacturing Test Flow

Fig. 1 illustrates the proposed test approach, which starts as soon as the chip is manufactured. If the tested chip fails, a diagnose step is executed to locate the faulty tiles. Let *n* be the number of system tiles and *m* be the number of necessary tiles to implement the systems functionality, then *n-m* is the number

978-3-9810801-7-9/DATE11/©2011 EDAA

of spare tiles. If the number of faulty tiles is lower or equal than *n-m*, the location of these tiles is sent to CAFES, otherwise the faulty chip is discarded. The task placer, presented in the next Section, loads a NoC model and the application task graph to determine the new task placement avoiding the faulty tiles. Finally, according to the NoC energy model, CAFES is able to estimate the energy consumption of the resulting task placement. Depending on the energy overhead compared to the fault-free chip, the chip can still be sent to the market, perhaps targeting low-end markets.

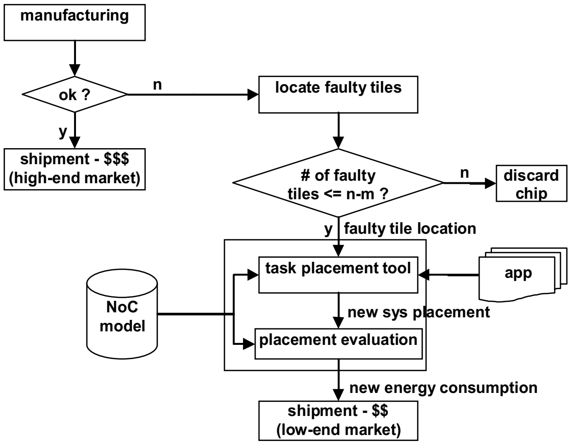

Figure 1. Proposed test flow for homogeneous NoC-based MPSoCs with spare tiles.

This paper assumes that each tile contains three main components: the router, the processor, and a memory block. The present work assumes faults only on processors and memories. Therefore, the NoC is assumed faulty-free.

III. Task Placement Aware of Faulty Tiles

CAFES[1] framework [6] is composed of high-level models, algorithms and tools, whose goal is to aid the designer to get fast prototyping of NoC-based MPSoCs. The mapping of application tasks onto the target architecture tiles aiming to save energy is one of these design tasks. Fig. 2 illustrates a partial mapping flow and the main elements used here.

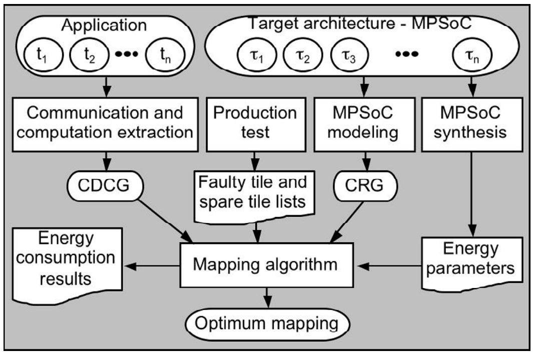
Application
t_1 t_2 ••• t_n
Target architecture - MPSoC
τ_1 τ_2 τ_3 ••• τ_n
Communication and computation extraction
Production test
MPSoC modeling
MPSoC synthesis
CDCG
Faulty tile and spare tile lists
CRG
Energy consumption results
Mapping algorithm
Energy parameters
Optimum mapping

Figure 2. The mapping flow used to obtain application mappings with the proposed framework.

Based on the description of an application already partitioned into tasks t_i, an extraction procedure generates the *Communication Dependence and Computation Graph* (CDCG), which represents the application behavior. Starting from the target architecture, a description composed by tiles τ_i and links allows modeling the *Communication Resource Graph* (CRG), which describes the target architecture topology. The energy parameters are extracted from the target architecture synthesized to a given technology. The MPSoC test enables to achieve the faulty tile list. According to the application description, NoC energy parameters, NoC topology and the faulty tile list, the task placer estimate the NoC energy consumption of different applications mapping, enabling to evaluate the impact of faulty tiles on mappings.

The *mapping problem* stated here consists in finding an association of each task to a given location in the target architecture that minimizes the global energy consumption. Applying simulated annealing techniques, the mapping algorithm tries to associate tasks to tiles avoiding the ones that are marked as spares or faulty. Therefore, when a tile is marked as faulty, the algorithm can replace the faulty tile with a spare tile.

The task placer models the dynamic energy consumption, using the concept of *bit energy* (**EBit**), similarly to the energy model described in [7]. Equation (1) illustrates how **EBit** is composed to model a 2D direct mesh NoC. It computes the dynamic energy consumed by a bit passing in such a NoC from tile **i** (τ_i) to tile **j** (τ_j), where η_{ij} corresponds to the number of routers that the bit traverses.

$$\mathbf{EBit_{ij}} = \eta_{ij} \times (\mathbf{Es}+\mathbf{Eb}) + 2 \times \mathbf{Ec} + (\eta_{ij} - 1)\times\mathbf{El} \qquad (1)$$

- **Es** is the dynamic energy consumption of a single bit on wires and on logic gates of each router;
- **Eb** is the bit dynamic energy consumption on router buffers;
- **Ec** is the bit dynamic energy consumption on links between routers and the local processor;
- **El** is the bit dynamic energy consumption on the links between routers.

IV. Experimental Setup

A *3x4 mesh* with 12 tiles, where three of them are spare tiles (9-out-of-12), is used to evaluate the results. The spare tiles are located at the top line of the NoC, i.e. the routers [0,3], [1,3], and [2,3].

A. Fault Generation Method

Faulty tiles are *exhaustively generated* for all combinations of faulty tile locations, assuming a system with 1 to 3 faulty tiles. Thus, (2) defines the total number of faults injected as the sum of all 1, 2 and 3 faults combination in 9 tiles, which results **129** fault scenarios for a 3x3 mesh NoC (9 single faults, 36 double faults, and 84 triple faults).

$$\binom{9}{1} + \binom{9}{2} + \binom{9}{3} = 9 + 36 + 84 = 129 \qquad (2)$$

[1] Available for download at https://corfu.pucrs.br/redmine/projects/cafes

B. System Application

Results are obtained using an object recognition benchmark named *App1*, which uses a distributed image segmentation algorithm.

App1 has 9 main tasks mapped onto 8 processors and a shared memory (in fact a tile memory). Images placed into the memory are equally size partitioned and distributed to all processors. These ones recognize objects and synchronize the recognition information to avoid false ones. Most of the communication volume is between processors and memory and minor communication volume are inter-processor communication.

The relevant feature of this benchmark is the communication dependence to a given task (the memory task), leading to a natural placement of this task in the middle of the NoC. It is then expected to observe an increase in energy consumption when the central tiles are faulty.

V. Experimental Results

TABLE 1 presents experimental results of the average energy consumption assuming four scenarios: a fault-free chip and one to three faulty tiles in the chip.

TABLE 1. Impact of faulty tiles in the NoC energy consumption for App1 experiment (9-out-of-12).

Fault-free	1-Fault	2-Faults	3-Faults
7534 uJ	7626 uJ	7868 uJ	8341uJ
	(1.22 %)	(4.43 %)	(10.70 %)

The first column of TABLE 1 presents the energy consumption of a fault-free 3x4 NoC, value used as reference. Next columns contains the average energy for 9, 36 and 84 combinations of single, double and triple faulty tiles, respectively. Numbers inside parentheses represent the energy consumption overhead compared to the fault-free system.

A. Detailed Analysis of App1

This sections details the results obtained for *App1* by analyzing the data obtained per fault location. We acquired the results obtained for the population of 84 combinations of triple faults in *App1*, which were divided in 9 groups: the first group contains all combinations with faults in the first tile ([0,0]), the second group contains all combinations with faults in the second tile, and so on. Fig. 3 summarizes the obtained results.

The horizontal line across the groups of Fig. 3 represents the average energy for each group. The vertical line in each group represents the minimal and the maximal energy. The gray box represents the quartiles, i.e. where 50% of the results are located. It confirms that the central tiles ([1,2] and [1,1]) have higher impact on the energy consumption. The tiles next to the central tiles ([0,1] to [2,2]) have the second most relevant impact on energy, and consequently, the most peripheral tiles implies less energy consumption impact.

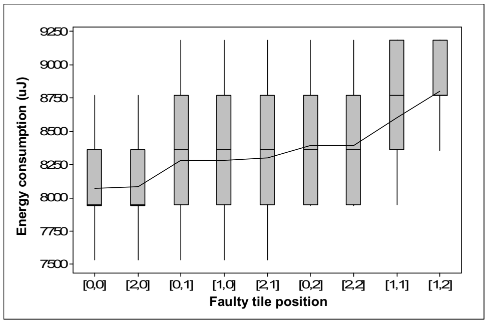

Figure 3. Statistics energy consumption according to faulty tiles position.

Fig. 4 presents a frequency distribution of these results. The data is placed in bins such that the *med bin* is related to data between 8300 and 8400 uJ, the *low bin* represents the results below 8300 uJ, and the *high bin* represents the results above 8400 uJ. As in Fig. 3, there is a clear pattern that increases the energy consumption as a fault is closer to the central tiles.

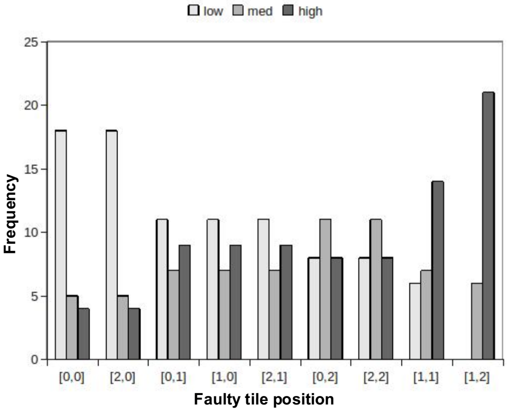

Figure 4. Frequency distribution of energy consumption according to faulty tile position.

Finally, Fig. 5 illustrates the average energy consumption for each tile group. It demonstrates that faults at tile [1,2] induce more energy consumption and the energy decreases gradually as the distance from [1,2] increases.

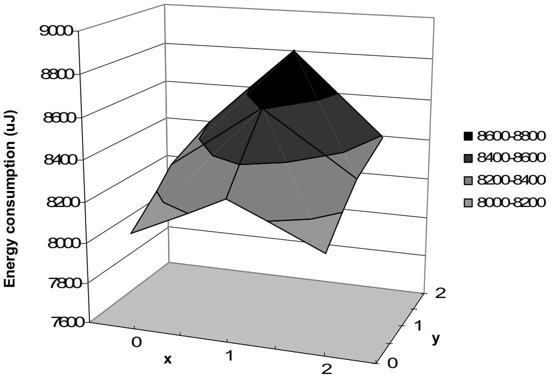

Figure 5. Average energy consumption per tile. The X an Y axes represent the tile position [X,Y].

B. *Estimating the Sample Size*

App1 was executed 129 times, which took about 3 minutes of CPU time for a small network. Even with the economical motivation of spare tiles is appealing, it might be unfeasible to perform an exhaustive fault simulation for multiple faults since the CPU time increases with the NoC size. For instance, while there are 84 combinations of triple faults in a system with 9 routers, there are 560 combinations of triple faults in a system with 16 routers, and 7140 combinations for 36 routers. Moreover, each execution takes longer time as the NoC size increases.

In fact, in practical situations it is not required to perform exhaustive simulations as did in this paper. Statistical tools such as *one-sample t-test* can be used to estimate the sample size. Minitab statistical software has been used to perform the following test.

Initially a pilot simulation is performed with few samples where faults are randomly distributed in the tiles. This pilot gives an *estimated standard deviation* of energy consumed by entire population (i.e. the entire combination of triple faults).

Secondly, the accepted *difference* between the sample and the population must be defined. For instance, a difference of 5% represents that the sample mean and the population mean are equal when the difference between them is below 5%.

Finally, the *α* and *power* (power is the name of the statistical variable -it is not related to power dissipation) are defined. Both of them represent how much we thrust that the sample is able to represent the actual population. Usually we use 5% and 95% for both parameters respectively.

Fig. 6 illustrates the power curves for the sample sizes equal to 21 and 84 combinations of 3 faults in *app1*. The one-sample t-test estimated that, assuming a standard deviation of 6%, difference of 5%, *α* of 5% and *power* of 95%, a sample size of 21 simulations is sufficient (see case1).

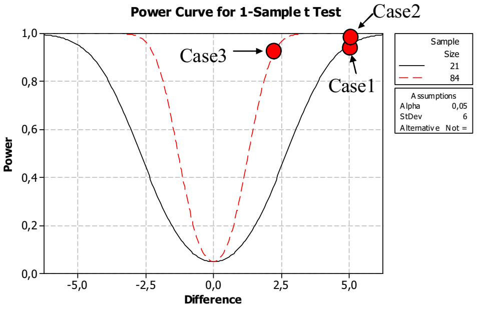

Figure 6. Estimating sample size for app1 with three faults.

Fig. 6 also compares the power curve assuming a sample size of 21 (estimated sample size) and 84 (size of the entire population). It shows that when the difference is 5%, a sample size of 85 increases the power from 95% to 100% (see case2). It also shows that for a power of 95%, a sample size of 84 generates a difference of up to 2.5% instead of 5% when a sample of size 21 is used (see case3).

We randomly select 21 samples out of the population of 84 simulations to test the sample size. In all tested cases the sample mean and standard deviation are according to the estimated.

VI. Conclusions and Future Works

Previous papers demonstrated that the use of spare tiles improve yield and reduce the manufacturing cost of homogeneous NoC-based MPSoCs. The tool presented in this paper determines task placement for chips with faulty tiles and evaluates its energy consumption overhead compared to fault free chips.

The results show that the energy overhead can be about 1%, 5%, and 10% for one, two, and three faults respectively. The results also show that faults in the central tiles typically have a bigger impact on the energy overhead, indicating that these tiles should be carefully designed, and targeting higher reliability.

Future work includes the evaluation of other performance figures such as power dissipation, hot spots, and latency for systems with faulty tiles. Moreover, we also intend to model the effect of faults in NoCs routers.

Acknowledgment

Alexandre is supported by postdoctoral scholarships from Capes-PNPD and FAPERGS-ARD, grants number 02388/09-0 and 10/0701-2, respectively. Cesar Marcon, Marcelo Lubaszewski, and Fernando Moraes are partially supported by CNPq scholarships, grants number 308924/2008-8, 478200/2008-0, and 301599/ 2009-2, respectively. The authors also acknowledge Letícia F. Pettenuzzo for her help with the statistical method for sample size estimation.

References

[1] Kahng, A.; et al. “ORION 2.0: A fast and accurate noc power and area model for early-stage design space exploration”. In: DATE, 2009, pp. 423-428.

[2] Shamshiri, S. and Cheng, K-T; “Yield and cost analysis of a reliable NoC”; VTS, 2009, pp. 173-178.

[3] Bertozzi, D.; Benini, L.; De Micheli, G.; “Error control schemes for on-chip communication links: the energy-reliability tradeoff”. IEEE Transactions on CAD, v.4, n.6, pp. 818-831, 2005.

[4] Ejlali, A. et al.; “Performability/energy tradeoff in error-control schemes for on-chip networks”. IEEE Transactions on VLSI Systems, v.18, n.1, pp. 1-14, 2010.

[5] Manolache, S.; Eles, P.; and Peng, Z.; “Fault and energy-aware communication mapping with guaranteed latency for applications implemented on NoC”, DAC, 2005, pp.266-269.

[6] Marcon, C. A. M. et al. “CAFES: A framework for intrachip application modeling and communication architecture design”. In: JPDC, 2010, in press.

[7] Ghadiry, M.; Nadi, M.; and Rahmati, D. “New approach to calculate energy on NoC”. In: ICCCE, 2008, pp. 1098-1104.