

A Lifetime-aware Mapping Algorithm to Extend MTTF of Networks-on-Chip

Letian Huang*, Shuyu Chen*, Qiong Wu*, Masoumeh Ebrahimi†, Junshi Wang*, Shuyan Jiang*, Qiang Li*

*University of Electronic Science and Technology of China, Chengdu, China, 611731

†Royal Institute of Technology (KTH), Stockholm, Sweden

huanglt@uestc.edu.cn

Abstract—Fast aging of components has become one of the major concerns in Systems-on-Chip with further scaling of the submicron technology. This problem accelerates when combined with improper working conditions such as unbalanced components’ utilization. Considering the mapping algorithms in the Networks-on-Chip domain, some routers/links might be frequently selected for mapping while others are underutilized. Consequently, the highly utilized components may age faster than others which results in disconnecting the related cores from the network. To address this issue, we propose a mapping algorithm, called lifetime-aware neighborhood allocation (LaNA), that takes the aging of components into account when mapping applications. The proposed method is able to balance the wear-out of NoC components, and thus extending the service time of NoC. We model the lifetime as a resource consumed over time and accordingly define the lifetime budget metric. LaNA selects a suitable node for mapping which has the maximum lifetime budget. Experimental results show that the lifetime-aware mapping algorithm could improve the minimal MTTF of NoC around 72.2%, 58.3%, 46.6% and 48.2% as compared to NN, CoNA, WeNA and CASqA, respectively.

Keywords—many-core system; Network-on-Chip; mapping algorithm; lifetime reliability

I. INTRODUCTION

Nowadays, Network-on-Chip (NoC) has been widely used in Multi-Processor System-on-Chip (MPSoC) but the aging issue of NoC, similar to other platforms, is emerging as a major research concern. Escalating device defects, shrinking feature-size and growing transistor density have negatively impacted the reliability which can be seen in the increase of the failure rate (both permanent and transient) [1]. Permanent faults reduce the system lifetime [2] and therefore, techniques are proposed to improve the systems lifetime in terms of mean time to failure (MTTF). Currently, there are mature methods and techniques to tolerate failures in cores due to aging. For instance, researchers have already proposed architectures that can gracefully tolerate up to a few hundred (>500) processor-logic permanent faults, in a 64-node CMP [3]. Even if a certain core is broken, it can be isolated because the core is an independent part, then the rest of MPSoC will continue to work. However, such well-protected components, cores, are connected to a less-protected infrastructure, NoC. Faults in a router can cause disabling a well-functioning core along with the router [4]. As another consequence, the connectivity of NoC will be devastated and the performance of MPSoC will be severely reduced. So, enhancing the lifetime of NoC has

the same level of importance as cores in MPSoC. LaNA tries to manage NoC reliability at run-time through task mapping as a low overhead approach.

Mapping algorithms try to allocate applications to the cores in an optimal way and aim to minimize the overall data latency and/or the power dissipation of the network. Mapping algorithms are mostly evaluated based on their defined cost functions. An application is composed of a set of communication tasks. To map an application onto the NoC-based multi-core system could be defined as a one-to-one mapping function from a set of application tasks to a set of cores. Mapping algorithms can decide the allocation of an application to the cores based on the usage of cores and routers in the NoC-based multi-core system.

The activity rate of one circuit is directly related to the aging of this circuit. Since the traffic in NoC is unbalanced, the activity of each part of a router is quite different. So, the activity of the routers should be represented by the activity of different paths. As shown Fig. 1, a path between the crossbars of two routers is composed of a mux connected with output register, an output register, a switch allocator for controlling this mux and this register, the wires between two routers, an input buffer, a routing computation unit and a virtual channel allocator. For example, Fig. 1 shows the activated circuit as a result of delivering a packet from Router A to Router B. So, the wear condition of one path is positively correlated with the wear condition of the wires subordinate to this path. In this paper, we model and refer to the aging of wires which represents the aging of paths as well.

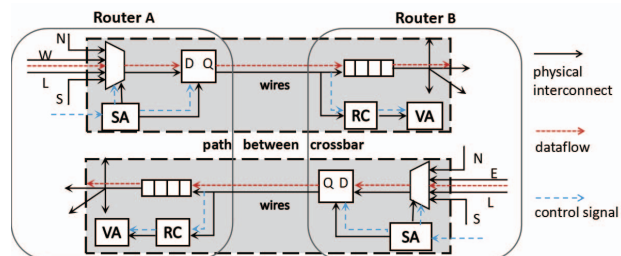


Fig. 1. The activity of one router's different parts are represented by the activity of links

Most of the dynamic mapping algorithms do not consider NoC aging, so that some wires could be aged much faster than the others. Fig. 2(a) and (b) show a case study that

presents the distribution of the NoCs' MTTF under two mapping algorithms as CoNA [5] and LaNA. The case study is evaluated in 8×8 2D mesh NoC. The traffic pattern is random and the injection rate is 0.05 flits/cycle. For example in CoNA, the minimum MTTF is 0.2 while the maximum is 1 which means that the links (a group of wires) with minimum MTTF are aging 5 times faster than the links with maximum MTTF. The unbalanced MTTF distribution would become a bottleneck for system reliability. In our proposed LaNA mapping, the minimal MTTF and the average MTTF are enhanced, which indicates the positive impact of aging consideration in mapping algorithms.

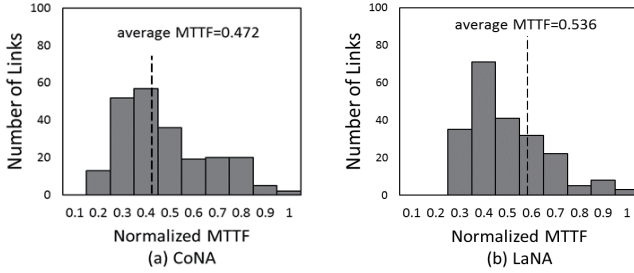


Fig. 2. In 8×8 NoC, the normalized MTTF of links is evaluated under different mapping algorithms

This paper presents a method for wear-leveling in NoCs to balance the usage and extend the overall lifetime. The main contributions of this paper are as follows: (1) Studying the effect of different mapping algorithms on unbalanced wires utilization and analyze the factors that cause changes on the MTTF values. (2) Modeling the aging of NoC's wires using the analyzed factors and then present a lifetime budget for the wires. (3) A lifetime-aware mapping algorithm based on the lifetime budget metric to enhance the NoC's lifetime.

II. RELATED WORK

Due to the adverse impact of the deep submicron technology, the system reliability have received a lot of research attention over the past few decades. Task mapping and scheduling-based system-level design techniques can provide a low overhead approach for enhancing the reliability.

The existing aging-aware task mapping techniques suffer from the following limitations: First, as the reliability of system highly depends on temperature, most prior works solely consider mapping tasks on an MPSoC platform with the objective of balancing the temperature of the cores [6] [7]. However, these methods neglect other factors of reliability such as switching activity and operating frequency. Second, previous works have completely ignored the role of routers in their reliability analysis, focusing only on the cores.

A lifetime-aware task mapping technique based on ant colony optimization (ACO) is proposed in [8]. A wear-based heuristic method is proposed in [9] that is combined with runtime task mapping. These techniques improve the cores' lifetime but they ignore about the aging of the NoC. When applications are mapped to the cores, data is communicated on

the NoC platform. If the aging of NoC is not considered in the application mapping, the NoC components might be soon aged which gradually results in isolating the cores from the system. If NoC is not equipped with proper fault-tolerant techniques, then the whole system may crash. Even by assuming such techniques, disabling NoC components (and consequently the cores) not only can impact the system performance, but also the lifetime of other cores due to the increased load.

A dynamic lifetime aware adaptive routing is proposed in [10]. This work introduced a metric for each router called lifetime budget. The adaptive routing aims to balance the lifetime distribution based on the lifetime budget metric. However, the capability of routing algorithms in enhancing NoC lifetime is small due to the limited path diversity and lack of global knowledge.

III. AGING EVALUATION

A. The Simulation Framework

To measure aging, we need to extract the temperature values under which the NoC is working. We set up a simulation framework for observing the temperature values on each component of a tile (i.e. core, router, cache, tags, and MC) to see the relations between the temperature values in different components. The simulation system used in this part is mainly composed of two mainstream open-source tools: McPAT [11] and Hotspot [12].

To understand the temperature distribution of NoC during mapping, we quantified the impact of the mapping process on the core temperature. We consider two different NoC floorplans similar to the ones in [13], illustrated in Fig. 3(a) and (d). we examine two workloads and depict the temperature of the tile components in a 3×3 mesh NoC. In this figure, the horizontal axis shows nine PEs with each PE including L2 cache data, core, router, L2 cache tags, and MC. The vertical axis represents temperature of each component and the ambient temperature is 45°C . Fig. 3(b) and (e) show the temperature of different tile components when all PEs are active. As expected, in both scenarios, the temperature of the cores is around 67°C which is the highest among all the other components. The router temperature is however around 47°C on average. In Fig. 3(c) and (f), we turn off four PEs (i.e. the PE number 3, 5, 6, and 9) to observe the effect of PE's activity on the routers' temperature. As can be seen from these figures, turning off the PEs has a small effect on the the routers' temperature. In other words, the router temperature mostly depends on its own activity. For this reason, in modeling the router aging, we only consider the router temperature which is nearly a constant value close to 47°C .

In the prior work [10], the entire tile temperature is assumed as the router's temperature. However, our observation shows that this approach is not accurate because the router's temperature is much lower than that of the core. This is because the distance of the core and router is relatively far in the NoC floorplan. Therefore in this paper, we use the ambient temperature as the router's temperature when modeling the

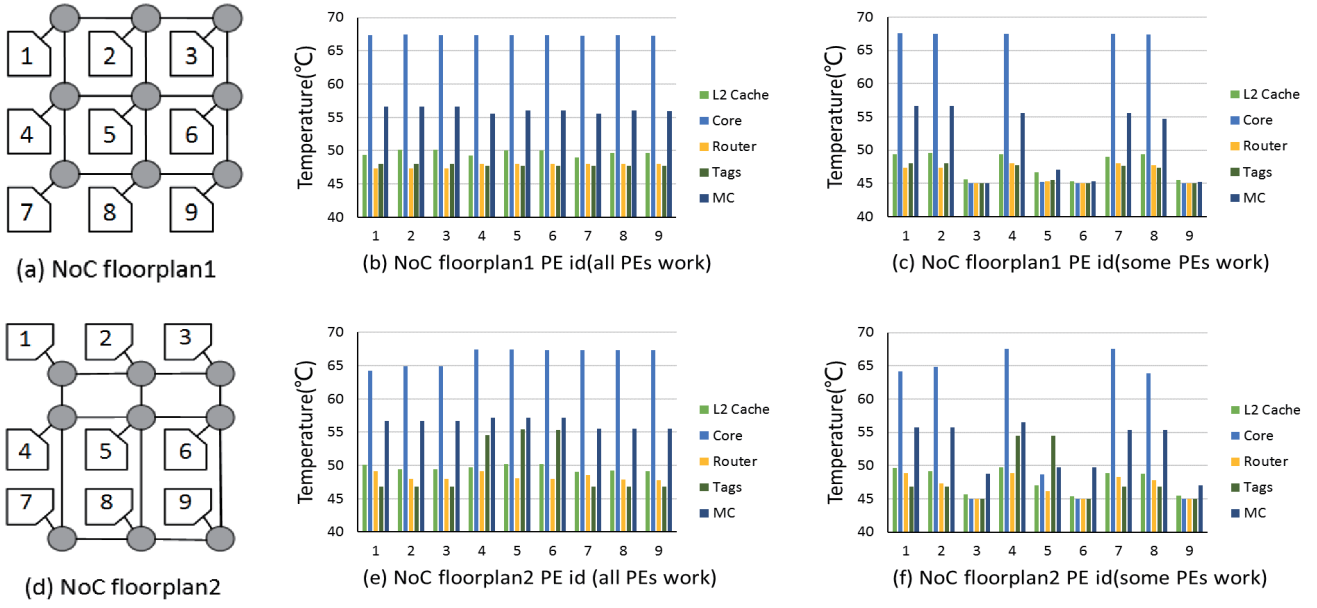


Fig. 3. The temperature of different tile components in a 3×3 mesh network with two different floorplans.

aging of routers. We assume the temperature as a constant value of 47°C .

B. Aging Formulation

Devices are aged for different reasons such as electromigration (EM), Negative Bias Temperature Instability (NBTI), and Time Dependent Dielectric Breakdown (TDDB) [14]. In this paper, EM is used as the wear-out related permanent faults when modeling the system lifetime. The aging rate is derived from [15]:

$$r(t) = j(t) \left(\frac{\exp(\frac{-Q}{kT_t})}{kT_t} \right) \quad (1)$$

where Q is the activation energy (e.g. 1.5eV for copper), $j(t)$ is current, and kT_t is the temperature. As was already discussed, the temperature is assumed to be a constant value, (T_a : ambient temperature), and thus T_t can be replaced by T_a . In this equation, $r(t)$ is a continuous value while under the discrete condition, the aging rate can be expressed by $r(n)$ where n refers to the n^{th} time interval. The current $j(t)$ is proportional to links' activity rate and can be measured by [2]:

$$j(t) = \frac{CV_{dd}}{WH} \times f \times p \quad (2)$$

where C , W and H are the capacitance, width, and thickness of the wire, respectively. f is the clock frequency and V_{dd} is the working voltage and both are constant values. p is the switching activity that is proportional to the incoming flit rate at the n^{th} time interval which is the only stimuli to the wires. Using Equation 1 and Equation 2, MTTF under time-varying current density and temperature stresses can be calculated by [15]:

$$T^f = \frac{A}{E[r(n)]} \quad (3)$$

where A is a constant related to the system structure and $E[\cdot]$ is expectation. Based on these equations, the wires lifetime budget under the discrete monitored condition can be expressed by [10]:

$$LB(n) = \begin{cases} 0 & \text{if } n \text{ is } 0 \\ LB(n-1) + r_n - r(n) & \text{otherwise} \end{cases} \quad (4)$$

where $LB(n)$, r_n , and $r(n)$ are the lifetime budget, the normal failure rate and the actual failure rate of wires, respectively at the n^{th} time interval. We use r_n to denote the lifetime consumption rate under the nominal condition and r_n is the inverse of the expected MTTF, i.e. $r_n \cdot T^f = A$. At runtime, we monitor the actual operating condition regularly, calculate the consumption rate $r(n)$, and compare it with the normal failure rate r_n . When $r(n) < r_n$, the wires are consuming their lifetime budget slower than the nominal rate, and vice versa.

Lifetime is modeled as a wire resource consumed over time and a lifetime budget of the wires indicates the maximum allowed workload in a given time. The wires with a small lifetime budget means faster aging.

IV. LANA: LIFETIME-AWARE NEIGHBORHOOD ALLOCATION

The proposed mapping algorithm, named Lifetime-aware Neighborhood Allocation (LaNA), is composed of two novel contributions. The first contribution is to exploit an efficient mechanism for selecting the first node to map. The second contribution is to select some available nodes around the first node for mapping with the consideration of the lifetime metric. The details of these contributions are described in following subsections.

A. First-Node Selection Strategy

For minimizing the probability of congestion, applications should be preferably mapped to the square-shaped regions. The selection of an optimal first node has to be spatially available and contiguous. However, since applications enter and exit the platform at different times, finding a free square area is not always possible. Hence, our approach is to search the nodes with the largest number of available neighbors. Several nodes might be available and these nodes located in the center of a square-based region which fit the application will be the options. Then, in this case the region with the maximum lifetime budget will be selected. The maximum lifetime budget is calculated by summing up the lifetime budgets of all the nodes within the region.

As was mentioned, several available squares could accommodate the incoming application. To select a proper region, we assign a cost, called *PLB*, to each square based on the Equation 4 and choose a region with the maximum lifetime budget. The cost is presented by the following equation:

$$PLB = \sum_{i \in \text{Square}} LB_i \quad (5)$$

where LB_i is the lifetime budget of each node within the region. A higher *PLB* indicates that wires in the chosen square area have experienced a relatively lower workloads and thus they have a longer service lifetime. The first node to map is the node with the maximum *PLB* in the square area. To balance

Algorithm 1 First Node Selection

Input: appSize: Size of the entering application; A_p : Task graph of the application;

Output: C_{fn} : The selected first node for the mapping;

- 1: $ns \leftarrow$ the **node set** with the largest number of free neighbors;
 - 2: $PLB_{max} = 0$;
 - 3: **for** each node i in ns **do**
 - 4: Calculate PLB ;
 - 5: **if** $PLB > PLB_{max}$ **then**
 - 6: $PLB_{max} \leftarrow PLB$;
 - 7: $C_{fn} \leftarrow i$;
 - 8: **end if**
 - 9: **end for**
-

aging over the platform, we propose a first-node selection method for the lifetime-aware mapping algorithm, as described in Algorithm 1. First, all the nodes with the maximum number of free neighbors are chosen as the candidates. Second, depending on the size of the entering application, the radius factor is determined with the candidate nodes in the center. Third, *PLB* for each square area is calculated, i.e. the sum of the wire lifetime budget $LB(n)$ for all the nodes in the square area. Finally, the node with the maximum *PLB* is selected as the first node among all the available nodes.

B. LaNA Mapping Algorithm

To improve the system's minimal MTTF and balance the lifetime distribution, the lifetime-aware neighborhood alloca-

tion mapping algorithm is proposed which aims at finding an optimal placement of nodes for task mapping. The algorithm based on communication flow and the lifetime budget of wires (*PLB*). The formulated problem is as follows. Let us first assume a directed graph $G = (V, A)$ where V represents the set of nodes and A refers to the set of wires. By assuming two nodes $s, d \in V$, then $F_{s,d}$ denotes the communication flow from the source node s to the destination node d according to XY routing. Between the node s and d , there might be several hops. The minimum value of these lifetime budgets will be used to present the lifetime budget of the flow $F_{s,d}$. The lifetime budget of the flow $F_{s,d}$ is:

$$FLB_{s,d} = \min_{i=\{s,\dots,d\}} \{LB_i\} \quad (6)$$

where LB_i is the lifetime budget of the wire and $\{s, \dots, d\}$ are the paths between the routers from node s to node d . With lifetime budget as the cost, the $FLB_{s,d}$ is to select the fastest aging wire as the aging cost of the flow. There is usually a situation where a node has communication with several other nodes, thus the method evaluates the lifetime budget of a node by selecting the related minimal $FLB_{s,d}$, that is the fastest aging flow lifetime metric. $FLB_{s,m}$ and $FLB_{m,d}$ are the node m as a starting node and end node of flow lifetime budget. The worst cost of nodes' lifetime budget is represented as the following equation:

$$PELB_m = \min_{s,d} \{FLB_{s,m}, FLB_{m,d}\} \quad (7)$$

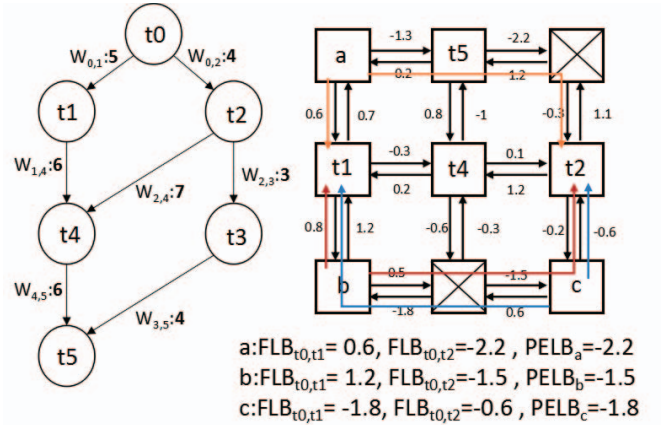


Fig. 4. An example of the LaNA mapping algorithm

The pseudo-code of the proposed mapping algorithm with lifetime improvement is given in Algorithm 2. The mapping is started by initialization the first node, first task and task mapping order (line 1-3). Then, the available nodes in the closest distance from the first node are explored. If there is only one available node, the corresponding task is mapped to it (line 5-7). If not, according to the position of the local and destination node, the cost of $FLB_{s,d}$ and $PELB_i$ are calculated (line 8-9). Finally, the optimal choice of node is the one that has the maximum $PELB$ (line 10-12). As an example, let us follow the mapping process of Fig. 4, where an

application with 6 tasks is going to be mapped to the system. According to Algorithm 2, the first task (t_4) with maximum degree is mapped onto the first node and its connected tasks $\{t_2, t_1, t_5\}$ are mapped to the free neighbors of the first node. The sequential choose order of neighborhood node is based on the value of $PELB$. Then, three nodes (a,b,c) can be selected for t_0 . The value of FLB_{t_0, t_1} and FLB_{t_0, t_2} should be calculated because t_0 has communication with t_1 and t_2 . Then the $PELB$ of these three nodes (a,b,c) are calculated. The value of FLB and $PELB$ of three nodes are shown in Fig. 4. According to LaNA, task t_0 is mapped onto the node b which has the maximum $PELB$.

Algorithm 2 Lifetime-aware Mapping

Input: A_p : Task graph of the application; t_p : the predecessor task of current task
Output: map_result[]: mapping result, map_result[i]=j means task i is mapped to PE_j ;

- 1: Initialize: first_node \leftarrow by Algorithm1;
- 2: Initialize: first_task \leftarrow the task with maximum degree;
- 3: Initialize: $M \leftarrow$ task order by WeNA;
- 4: map_result[first_task] = first_node;
- 5: **for** each task t_c in M **do**
- 6: $PE_p \leftarrow$ map_result[t_p]
- 7: $N \leftarrow$ the free PEs with the minimum distance from PE_p ;
- 8: **if** more than one PE is selected **then**
- 9: Calculate $PELB$ for each PE_i in N
- 10: **for** each PE PE_i in N **do**
- 11: select the PE_i that has the maximum $PELB$
- 12: map_result[t_c] = PE_i ;
- 13: **end for**
- 14: **end if**
- 15: **end for**

V. RESULTS AND DISCUSSION

A. Experiment Setup

Experiments are performed on the many-core simulator, which is an open source NoC simulator [16]. Table I shows the configured parameters in the simulator.

TABLE I
SIMULATION SETUP

Parameters	Values
NoC size	8×8
NoC frequency	1GHz
Packet size	5 flits
Buffer size	12 flits
Routing algorithm	XY
Total simulation time	10 million cycles

Several sets of applications with 4 to 20 tasks are generated using TGFF tool [17], where the communication volume is randomly selected between 6 and 14 packets of data. We employ Electromigration as the failure model. In our experiments, the mapping tables and lifetime budgets are updated after each

1000 cycles. The following parameters are used for computing the MTTF and LB metrics: $A = 1$, $C = 268fF/mm$, $V_{dd} = 1.5V$, $W = 0.6\mu m$, $H = 0.6\mu m$ and the ambient temperature of $45^\circ C$. In the experiments, we compare the lifetime-aware mapping algorithm with NN [18], CoNA [5], WeNA [19], and CASqA [20].

B. Minimum MTTF Evaluation

The wires with smaller MTTF wear sooner than those with higher values. In this section, the minimum MTTF for different mapping methods observe how the proposed method can improve the minimum MTTF. Three configurations are as: the system utilization of 60%, 80%, and 100%. It means that cores are dynamically enabled and disabled but ensuring that for example 60% of all cores are active at a time. Fig.5 shows the minimum MTTF values for different mapping algorithms. As can be seen from this figure, LaNA improves the minimal MTTF for 72.2%, 58.3%, 46.6%, and 48.2% than NN, CoNA, WeNA, and CASqA, respectively.

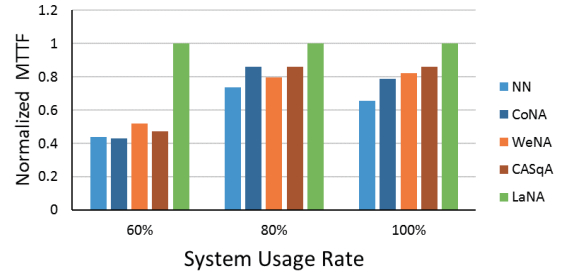


Fig. 5. Minimal MTTF evaluation

C. Average MTTF Evaluation

Fig. 6 (a) shows that the average MTTF of NoC based on LaNA has improved by 12.3%, 13.5%, 12.8% and 11.7% over NN, CoNA, WeNA and CASqA.

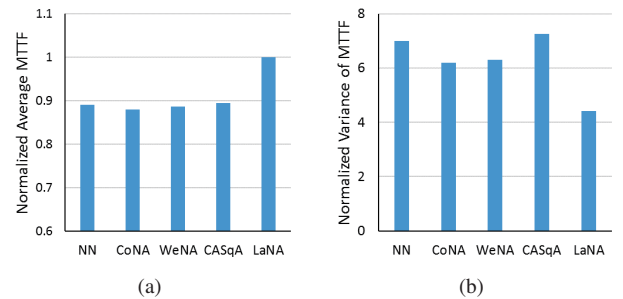


Fig. 6. MTTF evaluation: (a) average MTTF (b) variance of MTTF

Obviously, the more balanced lifetime distribution could improve the reliability of the whole NoC. The normalized variance of MTTF is shown in Fig. 6 (b). The proposed mapping algorithm decreases the variance of MTTF for 36.8%, 28.8%, 29.9% and 39.1% than NN, CoNA, WeNA and CASqA, respectively. In all situations, the MTTF first-node selection mapping algorithm improves the variance, showing its efficiency in achieving a more balanced lifetime distribution.

D. Average Latency and AWMD

The average latency of different mapping algorithms are shown in Fig. 7 (a). Compared with NN, CoNA, and CASqA, the average latency of NoC is decreased by 8.5, 0.6 and 3.1 cycle(s) when the lifetime-aware mapping algorithm is adapted. The average latency of lifetime-aware mapping algorithm is only 1.5 cycle more than WeNA. The result shows that the proposed method leads to a longer lifetime with negligible effect on the average latency.

Fig. 7 (b) shows the Average Weighted Manhattan Distance (AWMD) metrics experiments over different mapping algorithms by regulating the injection rate. As can be seen, the AWMD of LaNA is smaller than NN, CoNA, CASqA and is close to that of WeNA. The smaller AWMD value, the lower power consumption is expected, since the communicating nodes are placed closed to each other.

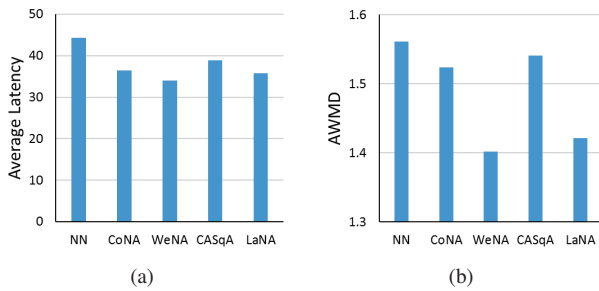


Fig. 7. NoC evaluation (a) average latency (b) AWMD

VI. CONCLUSION

In this paper, we proposed a mapping algorithm to improve and balance MTTF over the NoC platform. First, we investigated MTTF for different mapping techniques which highlighted the problem of unbalanced MTTF on NoC. Second, we analyzed the temperature values on different components of a tile. This analysis confirmed that the router temperature could be considered as a constant. Third, we model the aging of NoC wires, representative of a router activity, based on a router temperature. Finally, we proposed a first-node selection strategy and a lifetime-aware mapping algorithm to improve MTTF by considering the lifetime budget metric. Experimental results showed that our mapping algorithm leads to improvements on minimum, average, and variance of MTTF.

ACKNOWLEDGMENT

This paper is supported by the National Natural Science Foundation of China under grant (NSFC) No.61534002, No.61471407, Central Universities under Grant ZYGX2016J042. It is also supported by VINNOVA-MarieCurie.

REFERENCES

[1] C. Constantinescu, "Trends and challenges in vlsi circuit reliability," *IEEE micro*, vol. 23, no. 4, pp. 14–19, 2003.
 [2] J. Srinivasan, S. V. Adve, and etc., "The case for lifetime reliability-aware microprocessors," *Acm Sigarch Computer Architecture News*, vol. 32, no. 2, pp. 276–287, 2004.

[3] A. Pellegrini, J. L. Greathouse, and V. Bertacco, "Viper: virtual pipelines for enhanced reliability," *Acm Sigarch Computer Architecture News*, vol. 40, no. 3, pp. 344–355, 2012.
 [4] M. Ebrahimi, M. Daneshalab, J. Plosila, and H. Tenhunen, "Minimal-path fault-tolerant approach using connection-retaining structure in networks-on-chip," in *2013 Seventh IEEE/ACM International Symposium on Networks-on-Chip (NoCS)*, April 2013, pp. 1–8.
 [5] M. Fattah, M. Ramirez, M. Daneshalab, P. Liljeberg, and J. Plosila, "Cona: Dynamic application mapping for congestion reduction in many-core systems," in *Computer Design (ICCD), 2012 IEEE 30th International Conference on*. IEEE, 2012, pp. 364–370.
 [6] H. Sarhan, O. Eddash, M. Raymond, A. Wassal, and Y. Ismail, "Temperature-aware adaptive task-mapping targeting uniform thermal distribution in mpso platforms," in *Energy Aware Computing (ICEAC), 2010 International Conference on*. IEEE, 2010, pp. 1–3.
 [7] M. Bao, A. Andrei, P. Eles, and Z. Peng, "Temperature-aware task mapping for energy optimization with dynamic voltage scaling," in *Design and Diagnostics of Electronic Circuits and Systems, 2008. DDECS 2008. 11th IEEE Workshop on*. IEEE, 2008, pp. 1–6.
 [8] A. S. Hartman, D. E. Thomas, and B. H. Meyer, "A case for lifetime-aware task mapping in embedded chip multiprocessors," in *Hardware/Software Codesign and System Synthesis (CODES+ ISSS), 2010 IEEE/ACM/IFIP International Conference on*, 2010, pp. 145–154.
 [9] A. S. Hartman and D. E. Thomas, "Lifetime improvement through runtime wear-based task mapping," in *Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software code design and system synthesis*. ACM, 2012, pp. 13–22.
 [10] L. Wang, X. Wang, and T. Mak, "Dynamic programming-based lifetime aware adaptive routing algorithm for network-on-chip," in *International Conference on Very Large Scale Integration*, 2014, pp. 1–6.
 [11] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi, "Mcpat: an integrated power, area, and timing modeling framework for multicore and manycore architectures," in *Microarchitecture, 2009. MICRO-42. 42nd Annual IEEE/ACM International Symposium on*. IEEE, 2009, pp. 469–480.
 [12] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. R. Stan, "Hotspot: A compact thermal modeling methodology for early-stage vlsi design," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, no. 5, pp. 501–513, 2006.
 [13] É. Cota, F. L. Kastensmidt, and M. e. Cassel, "A high-fault-coverage approach for the test of data, control and handshake interconnects in mesh networks-on-chip," *IEEE Transactions on Computers*, vol. 57, no. 9, pp. 1202–1215, 2008.
 [14] J. Shin, V. Zyuban, Z. Hu, J. A. Rivers, and P. Bose, "A framework for architecture-level lifetime reliability modeling," in *Dependable Systems and Networks, 2007. DSN'07. 37th Annual IEEE/IFIP International Conference on*. IEEE, 2007, pp. 534–543.
 [15] Z. Lu, W. Huang, M. R. Stan, and K. Skadron, "Interconnect lifetime prediction for reliability-aware systems," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 15, no. 2, pp. 159–172, 2007.
 [16] J. Wang, Y. Huang, M. Ebrahimi, L. Huang, Q. Li, A. Jantsch, and G. Li, "Visualnoc: A visualization and evaluation environment for simulation and mapping," in *Proceedings of the Third ACM International Workshop on Many-core Embedded Systems*. ACM, 2016, pp. 18–25.
 [17] R. P. Dick, D. L. Rhodes, and W. Wolf, "Tgff: task graphs for free," in *Hardware/Software Codesign, 1998. (CODES/CASHE '98) Proceedings of the Sixth International Workshop on*, 1998, pp. 97–101.
 [18] E. Carvalho, N. Calazans, and F. Moraes, "Heuristics for dynamic task mapping in noc-based heterogeneous mpso," in *Rapid System Prototyping, 2007. RSP 2007. 18th IEEE/IFIP International Workshop on*. IEEE, 2007, pp. 34–40.
 [19] L. T. Huang, H. Dong, J. S. Wang, and M. Daneshalab, "Wena: Deterministic run-time task mapping for performance improvement in many-core embedded systems," *IEEE Embedded Systems Letters*, vol. 7, no. 4, pp. 93–96, 2015.
 [20] M. Fattah, P. Liljeberg, J. Plosila, and H. Tenhunen, "Adjustable contiguity of run-time task allocation in networked many-core systems," in *Design Automation Conference (ASP-DAC), 2014 19th Asia and South Pacific*. IEEE, 2014, pp. 349–354.