Online Monitoring and Adaptive Routing for Aging Mitigation in NoCs

Zana Ghaderi[1], Ayed Alqahtani[2], Nader Bagherzadeh[1,2]
[1]Computer Science Department, University of California, Irvine, CA, USA
[2]Department of Electrical Engineering and Computer Science, University of California, Irvine, CA, USA
{zghaderi, ayeda, nader}@uci.edu

***Abstract*—Scalability of *Network-on-Chip* (NoC) as a promising solution for many-core systems can be jeopardized due to reliability challenges such as aging in advanced silicon technology. Previous mitigation techniques to protect NoC are either offline, while aging is strictly influenced by runtime operating conditions, or impose significant overheads to the system. This paper presents an online monitoring method through a *Centralized Aging Table* (CAT) for routers in NoCs. Router's capacity in flits, which are the main stimuli in routers, is predictable and limited for a given period of time. Consequently, stress rate and temperature, which are the major sources of aging mechanisms such as *Bias Temperature Instability* (BTI) and *Hot Carrier Injection* (HCI), will be in the predictable ranges, as well. Hence, our methodology uses CAT which is populated by values that represent aging degradation for each different pairs of stress and temperature ranges during a given period of time. Furthermore, utilizing CAT, we propose an online adaptive aging-aware routing algorithm in order to avoid highly aged routers which eventually leads to age balancing between routers. Additionally, our proposed routing algorithm reduces maximum age of routers by changing the shortest paths between source-destination pairs adaptively, considering routers' ages across them in each given period of time. Extensive experimental analysis using gem5 simulator demonstrates that our online routing algorithm and monitoring methodology, CAT, improves delay degradation of maximum aged router and aging imbalance on average by 39% and 52% compared to XY routing, respectively. The impact of our proposed methodology on network latency, *Energy-Delay-Product* (EDP) and link utilization is negligible.**

Keywords—Aging; NoCs; Monitoring; Delay Degradation; Adaptive Routing.

I. Introduction

Delay degradation due to aging mechanisms becomes a reliability challenge in advanced semiconductor technology [1]. It imposes a large design margin to the critical paths which results in design complexity and overhead [2, 3]. BTI and HCI are two dominant aging mechanisms causing accelerated transistor aging, which increase the transistor threshold voltage (V_{th}) over time [1-3]. This impacts the lifetime of the chip in long term and its performance (or critical path) in short term. Consequently, threatening the performance and scalability for many-core designs. Therefore, *Networks-on-Chip* (NoC), that consist of packet switched routers for providing high bandwidth, parallelism, and scalability for many-core systems requires careful aging investigation.

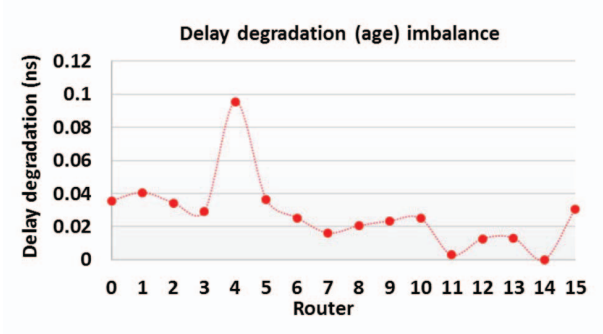

Fig. 1) Age imbalance of different routers in FFT

Critical path's age is affected by operating conditions such as stress (i.e. usage of transistors along it) and temperature, which change during the time because of variation in the running application on the system. In other words, higher temperature and stress leads to higher aging rate. Since, flits are the only router's stimuli, change in *number of flits* (fl) inside the router and their *residence time* (rs) affect the temperature and stress of the router. Therefore, by monitoring "*fl*" and "*rs*" we can predict temperature and stress; thus aging rate. Additionally, "*fl*" and "*rs*" are stimulated by routing algorithm. Hence, router's age and reliability have direct relations with routing algorithm. Therefore, by changing the routing algorithm and controlling source-destination shortest paths selection, the aging impact on NoC can be mitigated.

Since router as a component in NoCs, has a predictable and limited capacity of flits for a given period of time (P), then stress (S) and temperature (T) as two main sources of BTI and HCI are in limited ranges. Considering this observation, we propose a methodology based on a *Centralized Aging Table* (CAT). CAT is populated by the amount of aging degradation for different ranges of "*fl*" and "*rs*" in a router from zero up to the router capacity. This makes CAT independent from the running application. CAT, which is stored in one of the cores, can be accessed by all routers through the NoC in order to accumulate their current age to the pre-evaluated respective aging degradation. To compute "*fl*" and "*rs*", a counter and a timer is embedded into each router (elaborated in Section III.C).

As shown in Fig. 1, routers due to different usage (i.e. different "*fl*" and "*rs*") experience different temperatures and stresses, thus are impacted by aging differently. This leads to imbalance in age of routers, which may lead to reliability and

978-3-9815370-8-6/17/$31.00 ©2017 IEEE

scalability challenges in NoC. In this paper, we proposed an aging-aware routing algorithm which selects shortest paths between destination-source pairs adaptively based on router's age using CAT. CAT helps our proposed algorithm to adapt shortest paths online *periodically* as opposed to state-of-the-art works, which adapt routing based on offline aging information through profiling [4,5]. Therefore, our routing algorithm finds k-best shortest paths and selects between them periodically based on impact of aging on routers (using CAT) to reduce maximum aged router and balance the age between routers (elaborated in Section IV). Since aging mechanisms impacts critical path's delay gradually, we update the routing tables in periodic time (P) (e.g. each week).

Our extensive experimental analysis using gem5 simulator reveals that the proposed aging-aware routing algorithm reduces maximum routers' age and imbalanced routers' age by 39% and 52% on average in different benchmarks, respectively. Since, we select the best shortest path between k-best shortest paths (i.e. with same latency cost but different aging costs) the network latency overhead is negligible.

The rest of this paper is organized as follows. In section II, we elaborate and discuss related works. After that, an overview of the impact of aging mechanisms in NoC is demonstrated in section III. Our proposed aging monitoring technique using CAT is presented in Section IV. Section V proposes our aging-aware routing algorithm. Then, experimental setup and results are discussed in Section VI. Finally, the paper is concluded in section VII.

II. Related Work

A multi-objective *Integer linear programming* (ILP) based routing algorithm is proposed in [4]. This technique assigns lifetime budgets to each router offline and using ILP finds the best route considering aging, power estimation, and performance. Since the lifetime budget assignment is offline, any online variation due to change in workload cannot be captured. Authors in [5] proposed an adaptive aging aware algorithm considering the assigned lifetime budget to each router. Although, their proposed routing algorithm adapts online but the lifetime budget that is assigned to each router is offline, which may overestimate or underestimate actual workload. The main shortcoming of these two solutions is that the lifetime budgets are assigned offline by profiling. Therefore, they are application dependent.

Authors in [6] propose *Wearout Monitoring System* (WMS) for different components of a router to monitor aging online. After that, based on the packet's criticality, their algorithm chooses between buffered or bufferless routers to mitigate aging by deflecting non-critical packets to bufferless routers. This technique not only induces hardware overheads to routers due to complex WMSs, but also is only applicable in certain type of networks with different router architectures (i.e. heterogeneous NoC). Determining which packet is critical is also a crucial decision which may induce overhead to the system. A *dynamic programming* (DP) based routing algorithm is proposed in [7], which requires a parallel DP network as overhead to the system to propagate the lifetime budget of each router inside the network. Additionally, a complicated circuitry added to routers to find the lifetime budget of them which induce hardware overhead to the system, as well. The main shortcoming of these two solutions is large overheads. Authors in [8, 9] proposed aging aware task mapping for many-core heterogeneous architectures and a scalable sensor design that can be utilized in many core systems, respectively.

III. Aging in NoC

Transistors' delay degradation (Δd) (i.e. increment in V_{th}) manifests itself as delay degradation of critical paths along them. Thus, it results in timing failure or performance degradation of the system. In this section, we describe how induced Δd due to BTI and HCI is computed for routers in NoC.

A. BTI Aging Effect

The available models for BTI [10,11] describe it in two phases: *stress phase* and *recovery phase*. During the stress phase (i.e. transistor is ON and in high temperature), BTI occurs due to generation of the interface traps at Si-SiO_2, which gradually increase V_{th}. During the recovery phase (i.e. transistor is OFF) some of these traps are eliminated and partially recover the shift on V_{th}. Based on [10-12], the delay degradation due to BTI can be simplified as:

$$\Delta d_{BTI} = C_{BTI} \times Y^n \times t^n \times e^{-(\frac{E_a}{kT})} \times d_0 \quad (1)$$

Where, d_0 is pre-aged delay of the transistor, t is the transistor age, Y is the duty cycle of the transistor (how long the transistor is ON), T is temperature, n is constant depending on fabrication process, E_a is activation energy, k is Boltzmann's constant and C_{BTI} is BTI fitting parameter which depends on fabrication process.

B. HCI Aging Effect

By changing the current-voltage characteristic of transistor due to accelerated carrier within electric field inside transistor channel, HCI increases the V_{th}. Based on [13,14], the delay degradation due to HCI can be simplified as:

$$\Delta d_{HCI} = C_{HCI} \times \alpha \times f \times t^{0.5} \times e^{-(\frac{E_a}{kT})} \times d_0 \quad (2)$$

Where, α is the switching activity of the transistor, f is clock frequency, C_{HCI} is HCI fitting parameter which depends on fabrication process and the remaining symbols are as represented in (1). Duty cycle (Y) and activity factor ($\alpha \times f$) are both considered as *stress* (S) for BTI and HCI aging mechanisms, respectively.

IV. Online Aging Monitoring in NoC

Based on Eq. 1 and Eq. 2, the delay degradation (Δd) of transistors due to BTI and HCI are exponential function of temperature (T) and non-linear function of transistor usage, the so called *Stress* (S) [9-13]. Since, the only stimuli in routers are flits, we leveraged the flits *residence time* (rs) inside the router and *number of flits* (fl) to predict and monitor router's age online. *Stress* (S) and average temperature (T) of a router are function of number of flits (fl) inside routers and their resident

time (rs) (i.e. how long a router is busy) during a given period of time, epsilon (ε). Furthermore, considering the NoC characteristics (e.g. flit injection rate, topology, etc.) the maximum number of flits as well as their maximum residence time in ε is bounded by FL_{max} and RS_{max}, respectively. Therefore, we monitor *fl* and *rs*, which can be utilized to map their corresponding temperatures and stresses of a specific router.

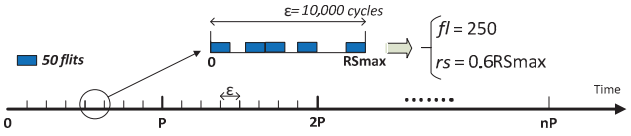

Fig. 2) Aging monitoring for each period of P

For instance, as shown in Fig. 2, each period P is divided to smaller period of ε. Therefore, each P is equal to (n×ε). For a specific ε (e.g. 10,000 cycles), *fl* is equal to 250 and *rs* is equal to 6,000 cycles (RS_{max} = ε = 10,000 cycles). This pair of *fl* and *rs* corresponds to a specific temperature, stress, and consequently aging rate. Fig. 4, illustrates our proposed architecture to monitor *fl* and *rs* which is embedded into router architecture [15]. Each core "*i*" is connected to a router "r_i". A parallel 12-bit counter [16] counts *fl* for each ε (upper counter in Fig. 4). It monitors valid incoming flits to the router from different ports to the router using *valid (V)* and *ready (R)* signals. More will be elaborated in Section VI.

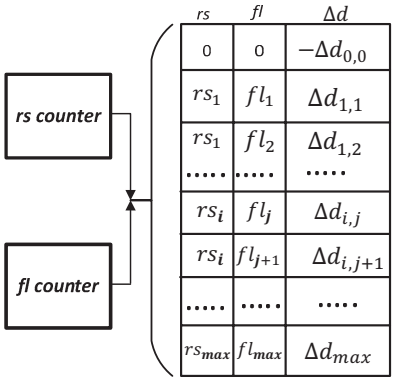

Fig. 3) CAT and required counters

The second parallel counter (lower counter in Fig. 4) is a 14-bit parallel counter that counts the resident time (*rs*) of the flits. Since RS_{max} (i.e. ε) can be presented by 14 bits, the counter is preceded by a 14-bit subtractor to subtract the exit time of an outgoing flit, which is the current cycle when the flit is exiting the router, from the en-queue time, which is saved inside the flit when it is en-queued inside input buffer. we assume the maximum *rs* of a flit inside a 5-stage router is 15 cycles. Therefore, following the subtractor, a 4-bit MUX is connected to drop any possible negative subtractions in the boundaries of each 10,000 cycles. Then it is fed to the parallel counter to keep accumulating resident time (*rs*) of all flits exiting the router through all possible five output ports. Moreover, these two counters reset after ε cycles. We use a timer to count ε and whenever its reaches to ε a reset signal is sent to the two parallel counters inside each router to be ready for next ε.

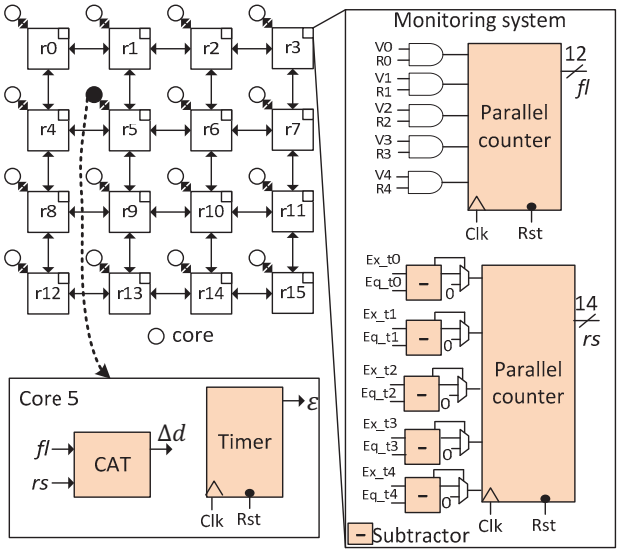

Fig. 4) Proposed online monitoring

To minimize the distance between CAT and all routers, CAT must be located in one of the middle routers. Fig. 4 illustrates that CAT resides in core 5. CAT will be accessed using (*fl, rs*) pair from all routers to read back their age degradation in each ε. Age degradation of a router can be computed for each temperature and stress (Eq. 1 and Eq. 2). To this end, we determine conditions that may happen to a router. Each condition, $C_{i,j}$, is represented by its respective rs_i and fl_j. Each pair of (rs_i, fl_j) corresponds to temperature $T_{i,j}$ and stress $S_{i,j}$ (i.e. ($T_{i,j}$, $S_{i,j}$)). Hence, each condition is a function of rs_i and fl_j and each condition corresponds to a specific aging rate. For example, in Fig. 3, when number of flits is fl_2 and they reside inside the router for queueing, processing and traversal through the router for rs_1 cycles out of ε cycles, the delay degradation is $\Delta d_{1,2}$. If the router is not busy (*fl* and *rs* are equal to zero) BTI recovery phase happens and CAT is filled by a negative corresponding amount of recovery.

Algorithm 1 shows how CAT is constructed. The inputs to this algorithm are maximum residence time RS_{max} (or the updating time period (ε)), the steps for each residence time rs_{steps}, the number of steps for counting flits inside the router $fl_{steps,}$ and the injection rate to the system, *Ijrate.* The algorithm's output is CAT which can be accessed from each router to read back its own age based on *fl* and *rs* during each ε. At the beginning, the maximum number of flits (FL_{max}) that can occupy a router during RS_{max} (or ε) considering maximum *Ijrate* is extracted (line 1). After that, the list of residence time (*rs*) and number of flits (*fl*) will be created based on their number of steps (line 2, 3). Using these two lists we calculate power consumption that can be used in HotSpot [17] for temperature extraction map (line 6). Each different residence time rs_i and number of flits fl_j have different power and temperature maps ($T_{i,j}$). Similarly, the stress will be extract as $S_{i,j}$ based on HCI and BTI aging mechanism (line 7). As shown in Eq. 1 and Eq. 2, *Stress* (S) is a function of duty cycle (Y) in BTI and switching activity (α) multiplied by

clock frequency (f) in HCI. In this work, Eq. 3 is utilized to calculate S as follows:

$$S = m_1 \times Y + m_2 \times \alpha \times f \quad (3)$$

Since impact of BTI is higher than HCI, m_1 is three times greater than m_2. For BTI, Y is equal to *"rs"* and for HCI α is equal to the ratio between *fl* and FL_{max}. After that the delay degradation for each pair of temperature and stress will be extracted as $\Delta d_{i,j}$ using Eq. 1 and Eq. 2 (line 8). Finally, the *CAT* will be filled of flits residence time (rs_i), number of flits (fl_j), and their corresponding delay degradation ($\Delta d_{i,j}$).

ALGORITHM 1. Generating CAT

Input: Maximum resident time RS_{max}, number of resident time steps rs_{steps}, number of flits steps fl_{steps}, injection rate *Ijrate*
Output: *CAT*

1. $FLmax \leftarrow FindMaxFlit(Ijrate,\ RSmax);$
2. $\{rs\} \leftarrow CreateRsList\ (RS_{max}, rs_{steps});$
3. $\{fl\} \leftarrow CreateFlList\ (FL_{max}, fl_{steps});$
4. **for** each $rs_i \in \{rs\}$ **do**
5. **for** each $fl_j \in \{fl\}$ **do**
6. $T_{i,j} \leftarrow CalTempreture\ (rs_i, fl_j);$ *// Call HotSpot*
7. $S_{i,j} \leftarrow CalStress(rs_i, fl_j);$ *// Eq. 3*
8. $\Delta d_{i,j} \leftarrow CalDelayDeg(T_{i,j}, S_{i,j});$ *//Eq. 1&2*
9. $FillCAT(rs_i, fl_j, \Delta d_{i,j});$
10. **End for**
11. **End for**
12. **Return** *CAT;*

V. Aging-aware routing algorithm

As mentioned earlier, due to operating conditions and different *fl* and *rs* in routers during the time, aging rate in routers is different. This imbalanced aging may result in timing failure in the highly aged (i.e. used) router and impact the scalability and reliability of the system. However, there are different shortest paths between source-destination pairs or paths that are very close to the shortest paths in term of cost (delay). These paths use different routers to transfer flits inside network. Algorithm 2 proposes an aging-aware routing algorithm using an added tag to the routers as their age. The age tag in each router will be updated online using CAT, periodically (P = n × ε). This tag is leveraged for choosing best aging aware shortest path between all available shortest paths from each source-destination pairs. Therefore, routing table in each router will be updated adaptively at each period of time P.

The inputs to Algorithm 2 are list of source-destination pairs, $\{(Src, Dest)\}$ and list of routers' age, $\{RAg\}$. The algorithm's output is the list of shortest paths for each source-destination pairs, $\{ShortPathPair\}$. Using *CalShortestPath* (), we find k-best shortest paths list for each pair of source-destination. *Dijkstra's* shortest path algorithm is leveraged to find this list. There are different algorithms that can be utilized for this purpose [18, 19]. After that, for each pair we check which paths do not include the maximum aged router by calling *MaxAgeR*() and then find the best paths based on minimum summation of ages on their routers using the list of ages by calling *MinAge*() (line 7). The new shortest paths for each source-destination pairs are found for the next ε and are added to the list of shortest paths (line 8).

ALGORITHM 2. Aging aware routing algorithm

Input: Src-Dest pair list $\{(Src, Dest)\}$, Routers' age list $\{RAg\}$
Output: List of shortest paths $\{ShortPathPair\}$

1. ShortPathPair= {}
2. **for** each $Pair_i \in \{(Src, Dest)\}$ **do**
3. $K\_ShortPath\{\} \leftarrow CalShortestPath(Pair_i);$ *//Dijkstra's*
4. **End for**
5. **for** each $Pair_i \in \{(Src, Dest)\}$ **do**
6. **for** each $Path_j \in \{K\_ShortPath_i\}$ **do**
7. **if !**MaxAgeR($Path_j, \{RAg\}$) and MinAge ($Path_j, \{RAg\}$) **then**
8. ShortPathPair. Add ($Path_j$)
9. **End for**
10. **End for**
11. **Return** ShortPathPair

VI. Experimental Setup and Results

A. Setup

Our modeling and experiments are conducted using gem5 [20] which is an event driven simulator that can simulate the behavior of a full system. In addition, we adopt a ruby memory model with mesh interconnect network. Furthermore, Garnet [21] network model is used with 5-stage routers that is embedded inside gem5. In order to extract power estimation results for these stages, we used Mcpat [22] for different ranges of *"fl"* and *"rs"*. HotSpot [17] is used to extract temperature maps of a router for different extracted powers. To get the router's floorplan for temperature analysis, the architecture in [15] is used. The floorplan is extracted for 45nm technology using Cadence® tool chain. In aging model (Eq. 1 and Eq. 2), the values for C_{BTI} and C_{HCI} are chosen such that the maximum delay degradation in 3 years is 20% in worst case (transistors always ON, the maximum frequency ($\alpha \times f$ = 0.5GHz) at temperature 380 Kelvin).

In modeling stage, RS_{max} is assumed to be 10,000 cycles which can be counted using a 14-bit counter. In order to get the maximum number of flits, FL_{max}, we use a representative synthetic traffic patterns with flit injection rate = 0.05 for ε (or RS_{max}). As an observation, *fl* cannot exceed 2,300 flits. To confirm that 2×2 and 4×4 mesh NoC experiment are performed assuming a full system mode with SPLASH -2 [23] and similar observations are detected. As a result, we fixed our FL_{max} at 2,300 which can be counted by 12-bit counter.

SPLASH-2 benchmarks are adopted for our experiments. Each run is a full-system architectural simulation of 16 cores interconnected via 4×4 mesh topology. All routers can accept 16-byte flits and assume a virtual channel group that has 4 virtual channels which holds four flits. Each router in the system has 5 input ports and 5 output ports including the ones for the local processor caches. Each router is connected locally to one core with one L1 instruction cache, one L1 data cache, and one private L2 cache with sizes of 32kB, 32kB, and 16M respectively. The simulation setup is listed in Table I.

To evaluate the impact of our proposed technique, 7 different benchmarks are selected from different applications. For each benchmark we extracted their aging impact on different router in the above mentioned NoC. The results are

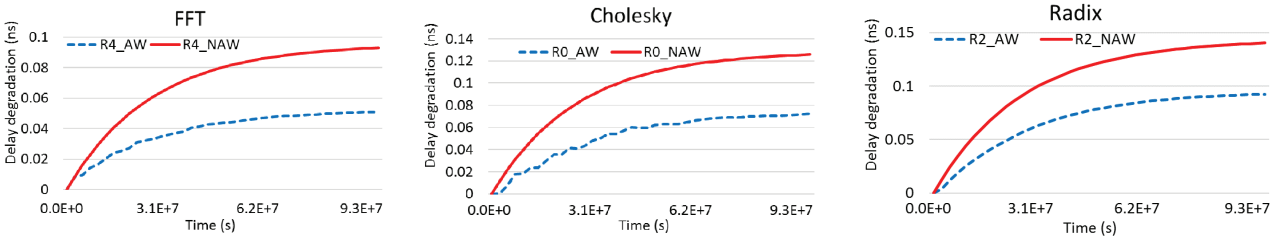

Fig. 5. Maximum aged router comparision in 3 years (9.3E+7 seconds)

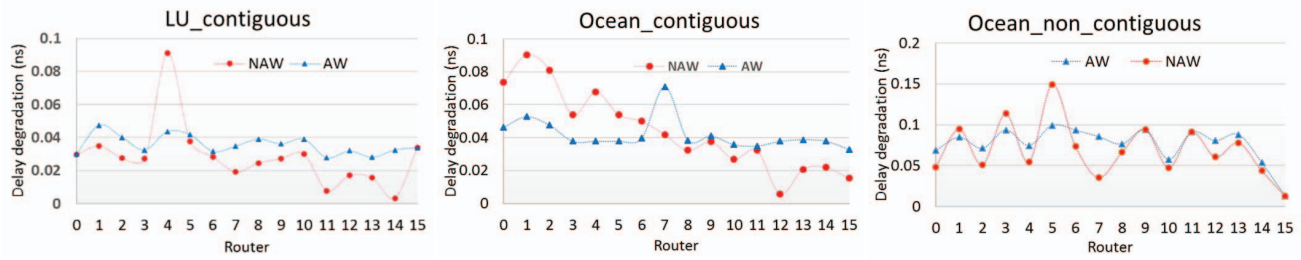

Fig. 6. Delay degradation (age) imbalance (Δ) between different routers in NoC

TABLE 1
SIMULATION SETUP

Name	Value
Frequency	1GHz
Number of cores	16 cores (X86 ISA)
Main memory	512 MB
L1 icahce	32KB, 2way, 64B blocks, 4cycles, pseudo LRU
L1 dcahce	32KB, 2way, 64B blocks, 4cycles, pseudo LRU
L2 size	16MB, 64B blocks, 12 cycles
Mesh	4x4
NoC routers' flit size	16B
VC number	4
VC buffer size	4* 16B

extracted for our proposed adaptive and online aging-aware routing algorithm (AW) and non-aging aware XY routing algorithm (NAW).

B. Results

Fig. 5 demonstrates the aging rate of highly used router in different benchmarks and effectiveness of our proposed method. The fluctuations in AW curves (dotted-blue) demonstrate that highly aged router goes to recovery phase (in BTI) some times. As a result, its aging rate is diminished. For example, the age of maximum aged router is improved by 45%, 42%, and 38% in FFT (router 4), Cholesky (router 0), and Radix (router 2), respectively. Second, fifth and eighth columns in Table 2 represents maximum age for routers in NAW, maximum age for routers in AW, and the percentage of improvement on maximum age router for each benchmark, respectively. The maximum aged router age is improved by 39% on average. It needs to be noted that our proposed aging aware algorithm monitors aging online and reacts accordingly as opposed to previous works that determine budgets offline and through profiling for each router [3-6]. While changing benchmarks and operating conditions, they have direct impact on changing the aging rate. This may lead to timing failure due to underestimation of the overhead or due to overestimation of the workload's attitude.

Furthermore, in Fig. 6 the delay degradation imbalance for different routers in the NoC is presented. As it is illustrated, in AW the routers' ages are more balanced in comparison to NAW algorithm. Hence, the load on highly aged routers are moved to lower aged routers. The age imbalance (Δ) is defined as the difference between highly aged router (maximum) and least aged router (minimum) in NoC. For LU_contiguous, Ocean_contiguous, and Ocean_non_contiguous benchmarks imbalance is improved by 78%, 55%, and 37% respectively. In Table 2 columns fourth, seventh, and tenth shows imbalance in NAW, imbalance in AW, and percentage of improvement for different benchmarks, respectively. Our technique improves imbalance in routers' ages by 52%.

Third, sixth and ninth columns in Table 2 are dedicated to average aging of routers in NoC. As can be seen, our technique increases the average age of routers by 24%. The reason is that our technique balances age on all routers equally and avoid highly aged routers in the network. Therefore, the average aging of routers will increase. Our adaptive routing algorithm chooses between different shortest paths of source-destination pairs considering the age of routers across them to avoid increase in network latency and the negative impact of performance on the system.

C. Overhead analysis

To calculate the impact of sending aging information (i.e. *fl* and *rs*) in a 4×4 mesh, 12 bits for *fl* and 14 bits for *rs* are required. Since they make a total of 26 bits, they can be encapsulated in one flit given that the number of bits per flit is 128 bits. The impact of sending that flit in traffic is estimated by

Table 2. Aging degradation for maximum aged router, average, and its imbalance (Δ) in NoC									
	NAW			Proposed method (AW)			Improvement		
	MAX (ns)	AVG (ns)	Δ (ns)	MAX (ns)	AVG (ns)	Δ (ns)	MAX (%)	AVG (%)	Δ (%)
FFT	0.095391	0.027563	0.095391	0.052433	0.034459	0.028642	45.03	-25.02	69.97
Cholskey	0.124119	0.037332	0.12418	0.072308	0.065592	0.072308	41.74	-75.69	41.77
LU_Con	0.091217	0.028434	0.087929	0.047564	0.035712	0.019708	47.85	-25.59	77.58
LU_Ncon	0.093700	0.028465	0.090576	0.048088	0.035989	0.021183	48.67	-26.43	76.61
Ocean_Con	0.09031	0.044061	0.0845528	0.071026	0.052713	0.038201	21.35	-19.63	54.81
Ocean_Ncon	0.149229	0.069564	0.136566	0.099229	0.076434	0.086566	33.50	-9.87	36.61
Radix	0.13362	0.066006	0.115422	0.082685	0.072103	0.082675	38.11	-9.23	28.37
Amean	0.111084	0.043068	0.0873868	0.067619	0.053286	0.0498976	**39.12**	**-23.74**	**52.45**
Gmean	0.108951	0.04016	0.074362	0.0652827	0.0505541	0.0421504	**40.08**	**-25.88**	**59.20**

counting the total number of cycles that are needed to reach router 5 from all other routers. We found out that at most 320 cycles are needed to send all required information in each ε or 10,000 cycles. This account for 3.2% overhead of total traffic. We can calculate the overhead impact also by counting the number of overhead flits per router in each ε. The percentage of overhead flits is less than 0.2%. In addition, our method has minimal impact on EDP, network latency and link utilization, as it is demonstrated in Table 3. On average, EDP in AW is only 0.31% higher than NAW. Similarly, network latency is higher by 0.15 and link utilization is lower by only 0.25%.

Table 3. Overhead		
EDP	Network Latency	Link Utilization
0.31%	0.15%	-0.25%

VII. Conclusion

In this paper we proposed online monitoring technique for aging in NoC routers, which is utilized in our aging-aware routing algorithm. Since routers' capacity of flits is predictable and limited in a given period of time we can predict aging rate as well. The router is analyzed for different number of flits for temperature and stress to extract a *Centralized Aging Table* (CAT). CAT is placed in one of the middle cores that has minimal distance to the other cores inside the network, which can be accessed by each NoC router based on their number of flits and resident time during a given period of time. Our experimental analysis shows 39% and 52% improvement on critical path degradation of maximum aged router and aging imbalance, respectively, with negligible overheads.

References

[1] The International Technology Roadmap for Semiconductors (ITRS), 2015.

[2] Ebrahimi, M., Ghaderi, Z., Bozorgzadeh, E. and Navabi, Z., *"Path selection and sensor insertion flow for age monitoring in FPGAs."* In Design, Automation & Test in Europe Conference & Exhibition (DATE'16), pp. 792-797, 2016.

[3] Ghaderi, Z. and Bozorgzadeh, E., *"Aging-aware high-level physical planning for reconfigurable systems."* In Asia and South Pacific Design Automation Conference (ASP-DAC'16), pp. 631-636, 2016.

[4] Bhardwaj, K., K. Chakraborty, and S. Roy, *"An MILP-based aging-aware routing algorithm for NoCs.*", In Proceedings of the Conference on Design, Automation and Test in Europe, (DATE'12), pp. 326-331, 2012.

[5] Bhardwaj, K., K. Chakraborty, and S. Roy, "*Towards graceful aging degradation in NoCs through an adaptive routing algorithm.*" In Design Automation Conference (DAC'12), pp. 382-391, 2012.

[6] Ancajas, D. M., Chakraborty, K., and Roy, S, *"Proactive aging management in heterogeneous NoCs through a criticality-driven routing approach.",* In Proceedings of the Conference on Design, Automation and Test in Europe (DATE'13), pp. 1032-1037, 2013.

[7] Wang, L., Wang, X., & Mak, T, *"Dynamic programming-based lifetime aware adaptive routing algorithm for network-on-chip.",* In International Conference on Very Large Scale Integration (VLSI-SoC'14), pp. 1-6, 2014

[8] Muck, T.R., Ghaderi, Z., Dutt, N.D. and Bozorgzadeh, E., *"Exploiting Heterogeneity for Aging-aware Load Balancing in Mobile Platforms."* IEEE Transactions on Multi-Scale Computing Systems (TSMCS). 2016.

[9] Ghaderi, Z., Ebrahimi, M., Navabi, Z., Bozorgzadeh, E., and Bagherzadeh, N., *"SENSIBLE: A highly scalable sensor design for path-based age monitoring in FPGAs,"* IEEE Transactions Computers (TC), p. 1, 2016.

[10] Ogawa, S., and N. Shiono, *"Generalized diffusion-reaction model for the low-field charge-buildup instability at the Si-SiO 2 interface."* Physical Review B 51, no. 7 (1995): 4218.

[11] Wang, W., S. Yang, S. Bhardwaj, S. Vrudhula, F. Liu, and Y. Cao, *"The impact of NBTI effect on combinational circuit: modeling, simulation, and analysis."* IEEE Transactions on Very Large Scale Integration (VLSI) Systems 18, no. 2 (2010): 173-183.

[12] Huard, V., M. Denais, and C. Parthasarathy, *"NBTI degradation: From physical mechanisms to modelling."* Microelectronics Reliability 46, no. 1 (2006): 1-23.

[13] Tiwari, A., and J. Torrellas, *"Facelift: Hiding and slowing down aging in multicores."* In International Symposium on Microarchitecture, pp. 129-140. IEEE, 2008.

[14] Takeda, E., and N. Suzuki, *"An empirical model for device degradation due to hot-carrier injection."* IEEE electron device letters 4, no. 4 (1983).

[15] Open Source NoC Router RTL. https://nocs.stanford.edu/cgi-bin/trac.cgi/wiki/Resources/Router.

[16] Swartzlander, E. E, *"Parallel counters."* IEEE Transactions on computers 100, no. 11 (1973): 1021-1024.

[17] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron,and M. 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.

[18] Eppstein, D., *"Finding the k shortest paths."* SIAM Journal on computing 28, no. 2 (1998): 652-673.

[19] Aljazzar, H., and S. Leue, *"K*: A heuristic search algorithm for finding the k shortest paths."* Artificial Intelligence 175, no. 18 (2011): 2129-2154.

[20] Binkert, N. and others, *"The gem5 simulator."* SIGARCH Computer Architecture. News 39, 2 (Aug. 2011), 1–7.

[21] N. Agarwal, T . Krishna, L.-S. Peh and N. K . Jha, *"GARNET: A Detailed On-Chip Network Model inside a Full-System Simulator"*, ISPASS, 2009

[22] 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 International Symposium on Microarchitecture, ser. MICRO 42. 2009, pp. 469–480

[23] SPLASH-2. http://www-flash.standford.edu/apps/SPLASH.