ABENA: An Ageing Before Temperature Eletromigration-Aware Neighbour Allocation for Many-Core Architectures

Emmanuel Ofori-Attah[a] and Michael Opoku Agyeman[b]

[a,b]Faculty Of Science and Technology Technology, University of Northampton

ABSTRACT

The percentage of inactive nodes or dark nodes (we refer to dark nodes as dumso) in many-core systems increases because of power dissipation caused by continuous scaling in technology. To address this challenge, existing work employ several techniques. Some techniques place the dumso nodes strategically between active nodes to alleviate the temperature by choosing the nodes which are far away from each other. However, this increases the latency between nodes which require inter communication leading to performance degradation. Others employ Dynamic Thermal Management (DTM) to vary the Voltage Frequency (V/F) Scaling of the nodes whilst Task migration is used to migrate tasks. In this paper, an **A**geing **B**efore Temperature **E**letromigration-Aware **N**eighbour **A**llocation (ABENA) is proposed to alleviate the temperature of the nodes by using the Lifetime of nodes as the main parameter. Experiments show that our approach improves the lifetime of nodes.

Keywords: Many-Core, Dark-Silicon, Embedded Systems, Network-on-Chip, NoC

1. INTRODUCTION

Thermal problems caused by uncontrolled leakage consumption threatens to limit CPU utilization to as low as 50%, particularly in Server Systems where many-cores chips are employed. Therefore, to prevent performance degradation, optimisation techniques are in demand. Fortunately, several efforts have been proposed to increase the 50% by utilizing the dumso nodes.[1–4] Example of these are Architectural Heterogeneity, DTM and Run-Time Management (RTM).[5–14]

Consequently, application mapping is one approach existing work employ to distribute applications across the chip. Application mapping techniques can be categorised into two groups: contiguous and non-contiguous. Contiguous mappings distribute applications to nodes in the same area whilst non-contiguous assign them to nodes across the chip. Whilst contiguous mapping offers better performance, heat generated by active nodes functioning in the same region causes hotspot and over time spreads among neighbouring nodes. In contrast, non-contiguous approaches minimise hotspot and heat dissipation. However, there is an increase in the latency among communication intensive applications. Consequently, this can cause performance degradation.[15] Therefore, a balance between these two approaches is required for an optimised distribution. Additionally, exiting work do not consider the lifetime of nodes as one of the main factors; Hence, our proposed method. The proposed approach combines the advantages of these techniques by employing a cluster-based architecture. The cluster ensures that, a contiguous region is formed and application tasks are distributed across that region by choosing an active node from two neighbouring nodes. The main contributions of this paper are:

- Propose a Dark-Silicon Cluster Based Architecture which incorporates the advantages of contiguous and non-contiguous mapping
- Propose a Neighbouring allocation mapping that selects the highest lifetime node from two neighbouring nodes based on their ageing.

Further author information: (Send correspondence to A.A.A.)
A.A.A.: E-mail: oforiattahe@hotmail.co.uk
B.B.A.: Email Michael.OpokuAgyeman@northampton.ac.uk

The paper is organized as follows: Section II briefly discusses related work about heterogeneous nodes, dynamic application mapping which considers computation and communication intensive application. Section III presents an observation into the round-robin mapping algorithm and in Section IV, both proposed approaches are presented. Section V presents the experimental results. Finally, Section VI concludes the paper and discusses future work.

2. RELATED WORK

The allocation of resources in many-core systems have been the focus of technology since the emergence of the Dark-Silicon phenomenon. However,[13] states that, previous task-resource allocation only considers mapping applications contiguously without regard for generated heat amongst resources. Active components generate heat and over time spread among neighbouring resources causing thermal hotspot. Consequently, over time, this affects the lifetime reliability of the resources. This causes accelerated mechanisms such as electromigation (EM), negative bias temperature instability (NBTI) & time-dependent dielectric breakdown (TDDB) and leads to function slower than the other.[16]

Existing work that considers heat dissipation does not emphasise the lifetime reliability of systems and node utilization are a crucial metric when optimising performance. Xiaohang Wang et al.[17] proposed a virtual mapping algorithm to estimate the number of dark cores required for an application to efficiently execute. The aim of the virtual mapping algorithm is to prevent inefficient use of dark cores to ensure there are enough free cores for incoming applications. This algorithm considers communication and computation applications. However, node utilization is not considered as a metric when choosing the first node.

Kanduri et al.[18] presented adBoost, a thermal aware performance boosting system which boost the performance of active cores by the efficient use of dark cores. The algorithm employed in this system maps applications spatially to avoid hot spot. First Node selection is selected based on a finding a node that is far away using MapPro from an active node with sufficient nodes around it for mapping of application task. Unfortunately, node utilization is not used as a property when selecting the first node.

The following work, however, addresses this challenge by considering computation and communication demands when activating resources.

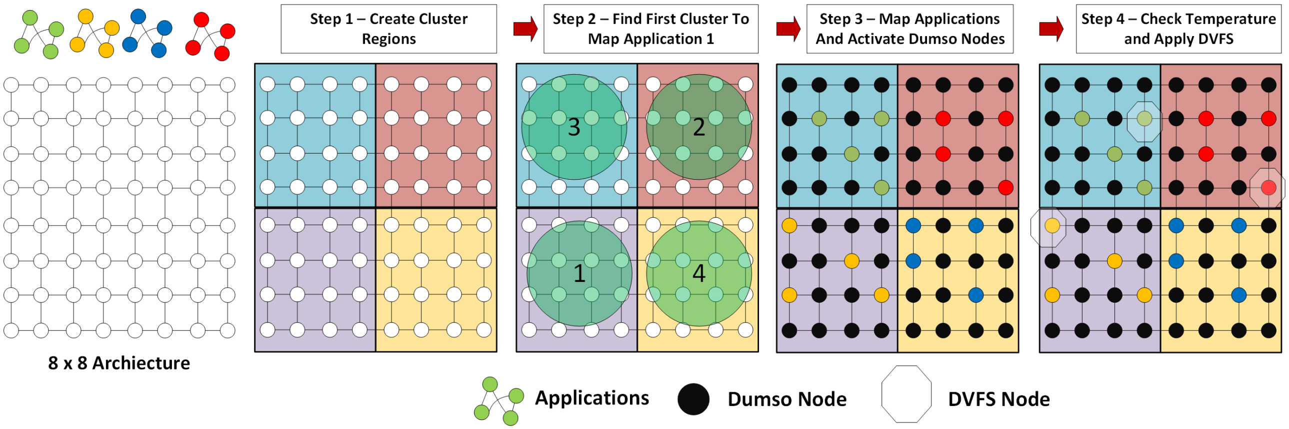

Step 1 – Create Cluster Regions
Step 2 – Find First Cluster To Map Application 1
Step 3 – Map Applications And Activate Dumso Nodes
Step 4 – Check Temperature and Apply DVFS
3
2
1
4
8 x 8 Archiecture
Applications
Dumso Node
DVFS Node

Figure 1: Proposed Architecture

Reza et al.[13] proposed a resource management system for heterogeneous NoCs which examines the performance requirement of an application (Communication or Computation Intensive) before distributing its tasks amongst the appropriate resources. In this management system, the chip is partitioned into clusters with a set of CPUs and GPUs. The architecture present employs a CPU from each cluster and assigns them as Cluster managers to monitor and configure the resources within that cluster. This information is then feed-backed to a global manager which uses the BalancedMap mapping application algorithm to activate the feasible resources based on the application. The BalancedMap Algorithm works as follow: Communication intensive applications are mapped to CPUs whilst Computation intensive application are assigned to GPUs.

The global manager is assigned by calculating the CPU with the shortest distance from all the assigned Cluster Managers. The BalancedMap Mapping Algorithm works by classifying the application into two groups (Communication and Communication Intensive). During CPU and GPU node selection, the node with the lowest peak energy is selected. If more than one node appears to meet the demand, the node with the lowest utilization is selected to improve the thermal hotspot. Power consumption is minimized by configuring the link, node voltage and the shutting down of idle routers when links are not carrying any communication traffic. Task migration is employed when a node being utilized exceeds it warning threshold.

Rahmani et al.[19] considers reliability in Dark-Silicon systems by employing reliability-aware power allocator to monitor the ageing of nodes and to reduce the amount of workload on stressed out regions that are experiencing fast ageing. Consequently, the voltage/Frequency of the nodes are scaled down to prevent the nodes from functioning at full throttle.

Rathore et al.[20] on the other hand, proposed a HiMap, a hierarchical mapping approach which reduces the lifetime of nodes by mapping application to healthier nodes whilst also placing dark nodes amongst the selected region. This mapping approach considers Process Variation when assessing the ageing and reliability of the node. The assessment is done by checking nodes for process variation, temperature and ageing. Weaker nodes are then used as dark nodes.

Mohammed et al.[21] proposed a technique which uses both Task Migration and DVFS for an optimise system performance. The Task migration technique is used to move cores from active to dark cores. In this architecture, the dark and active cores are all running concurrently and therefore makes it easier for task to be swap amongst them. Unfortunately, this increases the temperature and thus, DVFS is applied.

In our previous work,[22] we presented architectural saving techniques to improve the power efficiency for many-core systems. Based on the study conducted, it can be deduced the fraction of powered-off nodes in dark-silicon chips can be improved by power budgeting, architectural heterogeneity, NoC interconnect, Cache Memory, and run-time management. Particularly, resource allocation using application mapping through run-time management methods.

Similar to the above-mentioned techniques, we approach the Dark-Silicon challenge by permitting only 50% of the nodes available to function at full through per time. application spawns a number of T threads. $\{A_0, A_1, A_2, A_3, ...A_{K-1}\}$. The objective is to allocate each application to several N nodes $\{N_0, N_1, N_2, N_3, ...N_{N-1}\}$ to meet the application deadline without aggravating the temperature of the chip. Consequently, the number of applications currently being executed does not dictate the overall temperature. The overall temperature can fluctuate depending on several factors: the number of available dumso nodes, the position of dumso nodes, the temperature of active resources an application mapping algorithm. Consequently, the voltage frequency of nodes also has an impact. Therefore, the heat generated can be model as:

$$H = \sum \mathcal{P}^i \times \tau^{amb} \times \tau^i$$

Where $\mathcal{P}^i$ is the power dissipated by the node, τ^{amb} is the temperature of neighbouring nodes and τ^i is the ambient temperature.

Application mapping can be categorised into two categories: Contiguous mapping and con-contiguous mapping. Contiguous mapping techniques spawn application threads next to each other without considering the temperature and heat generated by neighbouring nodes. The heat dissipated by active nodes haphazardly affects neighbouring nodes and pushes nodes to reach their critical temperatures even when operating under low V/F. Pattern 1 in Fig 3 is an example of such an approach. Non-contiguous mapping avoids this challenge by dispersing application and threads across the chip. However, also causes thermal issues. This is because, communication latency between threads which require inter communication increases the power consumption and temperature.

Consequently, a naive dark and active node selection approach can lead to the mapping of two applications to result in the temperature of the chip being aggravated. Fig 3 depicts three different patterns. Pattern 1 is likely to result in a high temperature chip because all the nodes that have been selected for mapping are in the same region. Additionally, incoming applications will be continuously mapped to the same active nodes. Pattern 2 and 3 on the other hand have dispersed applications across the chip. This offers low temperature;

however, consideration of node performance has to be taken into account when taking such a decision. Without a consideration parameter, applications can be assigned to any node. This can either lead to bottlenecks or better performance. Consideration parameters can be referred to as a condition that allows applications to mapped to a node. This can either be, the temperature of the node, the lifetime of the node, the location of the node or the effect that mapping an application to a node can have on the incoming applications.

Therefore, different mapping approaches results in various thermal profile of the chip. Consequently, an optimal approach can lead up to 10 applications being mapped without the temperature of the chip being aggravated hence, our proposed approaches. However, unlike previous proposed approaches that statically and stochastically generate dark and active nodes, our proposed algorithm gracefully selects dark nodes and active nodes based on their current temperature.

3. MOTIVATION AND PROBLEM STATEMENT

At any given time, A number of applications dynamically arrive in the system. $\{A_0, A_1, A_2, A_3, ...A_{A-1}\}$. The objective is to allocate each application to a number of N nodes $\{N_0, N_1, N_2, N_3, ...N_{N-1}\}$ to meet the application deadline without aggravating the temperature of the chip. Consequently, the number of applications currently being executed does not dictate the overall temperature. The overall temperature can fluctuate depending on several factors: the number of available dumso nodes, the position of dumso nodes, the temperature of active resources an application mapping algorithm. Consequently, the voltage frequency of nodes also has an impact. Therefore, the heat generated can be model as:

$$H = \sum \mathcal{P}^i \times \tau^{amb} \times \tau^i$$

Where $\mathcal{P}^i$ is the power dissipated by the node, τ^{amb} is the temperature of neighbouring nodes and τ^i is the ambient temperature.

Application mapping can be categorised into two categories: Contiguous mapping and non-contiguous mapping. Contiguous mapping techniques spawn application threads next to each other without considering the temperature and heat generated by neighbouring nodes. The heat dissipated by active nodes haphazardly affects neighbouring nodes and pushes nodes to reach their critical temperatures even when operating under low V/F. Pattern 1 in Fig 3 is an example of such an approach. non-contiguous mapping avoids this challenge by dispersing application and threads across the chip. However, also causes thermal issues. This is because, communication latency between threads which require inter communication increases the power consumption and temperature.

4. PROPOSED MANY-CORE PLATFORM

4.1 System Architecture

The proposed many-core platform is partitioned into several clusters. As previously stated, contiguous and non-contiguous mapping both can have a negative impact on the performance of a many-core system. Therefore, to harness both approaches, we employ a Dark-Silicon cluster mapping. This approach allows a flexibility between restricting threads to a specific area whilst enabling applications to be dispersed in that region.

Each cluster has been allocated a module to monitor the temperature by applying DTM techniques to the nodes accordingly. Each module then returns the average temperature of its cluster to a centralised resource manager. Based on the information collected, applications are assigned to the right cluster. For each cluster, we employ a 50% dumso rule. The 50% dumso rule ensures that, half of the nodes in the architecture are powered-off.

In our proposed methods, applications are allocated to a specific and region restricted minimising the extent application threads can be dispersed. Dumso nodes are placed in between application mapping threads and dispersed strategically between a region.

4.2 The Proposed Dynamic Ageing-Aware Algorithm

Majority of dynamic mapping applications do not consider ageing even though some nodes age faster than others. To improve performance, temperature and power budget are usually used as measurement for enabling and disabling nodes. Over time, permanent faults caused by electromigration (EM), Negative Dependent Temperature Instability (NBTI), and Time Dependent Dielectirc Breakdown (TDDB) reduces the lifetime and increases the systems Mean Time To Failure (MTTF). Therefore, efficient techniques are required to improve the lifetime as well the performance. Hence we proposed a new method which uses the MTTF as main option for choosing dumso and active nodes. Algorithm **??** depicts the proposed method.

4.2.1 ABENA: Ageing Before Temperature Eletromigration-Aware Neighbour Allocation

The proposed mapping algorithm named Ageing-Aware Before Eletromigration Temperature Allocation (ABENA) consists of two novel approaches. The First novel contribution is the Choose-Me-Or-My-Neighbour Cluster Allocation. The second is the flow node allocation. The details for these contributions are explained in the following sub-sections.

4.2.2 First Node: Choose-Me-Or-My-Neightbour Cluster

In literature, existing technique selects the first node based on the number of available nodes in that region. In our proposed method, the nodes are divided into clusters and in each cluster, sub-clusters are formed. We assume that, in a periodic workload where the same set of applications are executed, the central manager is aware of the demands for every application and the execution time. Therefore, the appropriate cluster for each application is selected based on the average MTTF from that cluster. This ensures that, clusters with aging nodes are assigned to the right clusters. ABENA chooses first nodes by implementing the Choose-Me-Or-My-Neighbour algorithm which forms clusters between two nodes. This ensures that, there is always an active node next to a dumso node. The node with the highest MTTF is selected as the active node. This effectively reduces the temperature and heat generated by active nodes functioning at full throttle next to each other. Fig 2 depicts a 4x4 architecture implemented with our algorithm. Additionally, ABENA migrates tasks to other nodes after some cycles. DFVS is also employed to keep the V/F down.

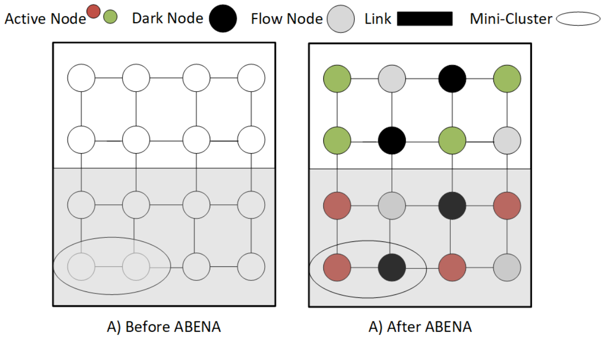


Figure 2: 4x4 Architecture With the Proposed Aging-Aware Algorithm

4.2.3 Flow Node: Inter-Clustering

To accommodate more applications, ABENA utilises the Flow Node Algorithm. This enables an application which is not heavily dependent on inter-communication to be mapped to nodes in different clusters. ABENA ensures that, at a particular time, all nodes from a cluster are not all active at a time or this will exacerbate the performance and increase the system temperature reducing the lifetime of nodes. Unlike existing literature which utilises dark nodes between applications, ABENA utilises dark nodes between tasks. Similar to,[20] at the end of every each epoch, applications are mapped to healthy nodes.

5. SIMULATION SET UP AND RESULTS

Cycle-accurate experiments are conducted using an extended version of Sniper called LifeSim,[23] a simulation tool which integrates Sniper,[24] McPat[25] and HotSpot[26] and the proposed approach. LifeSim simulator consists of a set of applications which spawns a number of threads which run simultaneously and can be picked from any of the available benchmarks. When the simulation starts, application threads are spawned on nodes specified by a mapping file supplied by the algorithm invoked by the simulator. We compare the proposed method with the default round-robin mapping algorithm.

The round-robin algorithm map threads spontaneously and in a serial manner. Pattern 1 and 2 in Fig 3 depicts a round-robin mapping. In order to test the effectiveness and performance of the new method, we tested our approach in several ways. Firstly, we test it on a 16 node 46 nm processor system to compare and analyse the effect that it has on a large technology. Details of the architecture is given in Table 1.

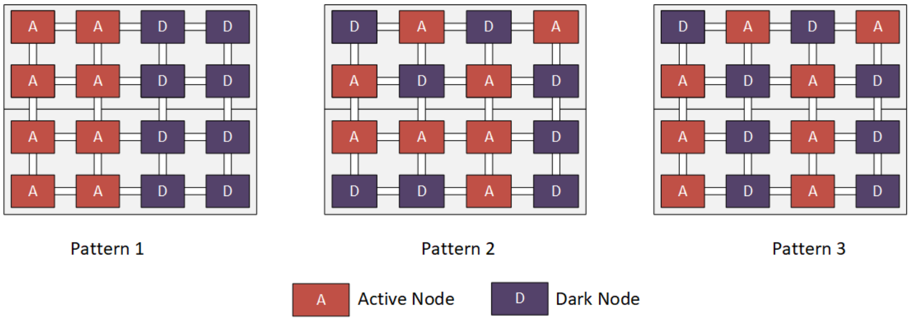


Figure 3: Dark-Silicon Patterns

5.0.1 Experimental Setup

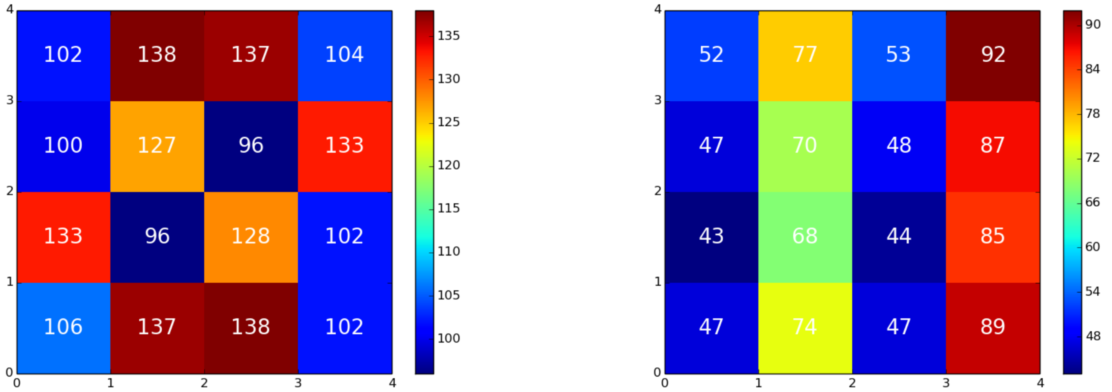
Experiments are performed using LifeSim simulator as previously mentioned. Table 1 depicts the parameters for the simulated configuration. The application used in this simulation is fft. We employ Electromigration as the failure model.

Table 1: Summary of Simulation Parameters

Parameter	**Value**
Nodes	16 (Gainestown)
Core Frequency	1.0 GHZ
Core Size	5.71 x 5.71
L1 Data Cache	32KB, 4-way set-associative, 64B Block size
L1 Instruction Cache	32KB, 4-way set-associative, 64B Block size
L2 Cache	256KB, 4-way set-associative, 64 B block size
L3 Cache	8129
Application	fft
Technology	45 Nm

5.0.2 MTTF Evaluation

Fig 4a and Fig 4b shows the MTTF of ABENA and the round-robin algorithm. It can be observed that the MTTF frequency is higher than the round-robin algorithm. By ignoring the lifetime of nodes before mapping, more pressure is added to already ageing nodes. ABENA ensures, that the youngest nodes amongst the neighbouring nodes are selected. This can significantly improve the lifetime of nodes. There is a significant lifetime increase between the lowest ageing node in the ABENA architecture compared to the round-robin algorithm as depicted in Fig 5. The lowest lifetime in ABENA is 100 compared to 43 with the round-robin algorithm.



(a) MTTF of 4x4 mapping with ABENA

(b) MTTF of 4x4 mapping with Round-Robin

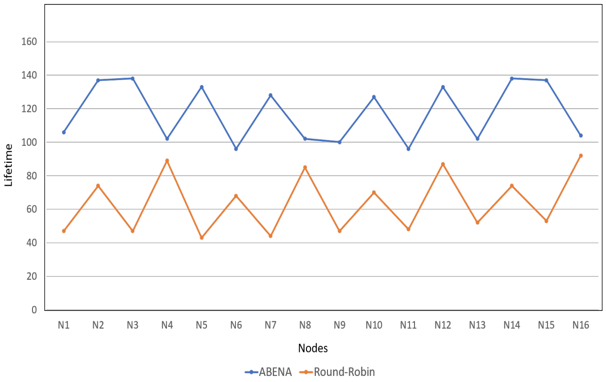
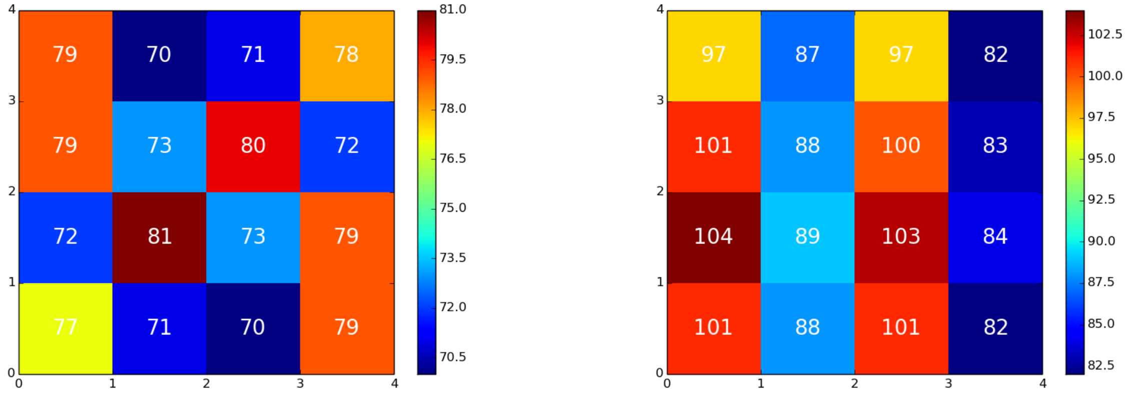


Figure 5: Lifetime of Nodes: ABENA and Round-Robin Algorithm

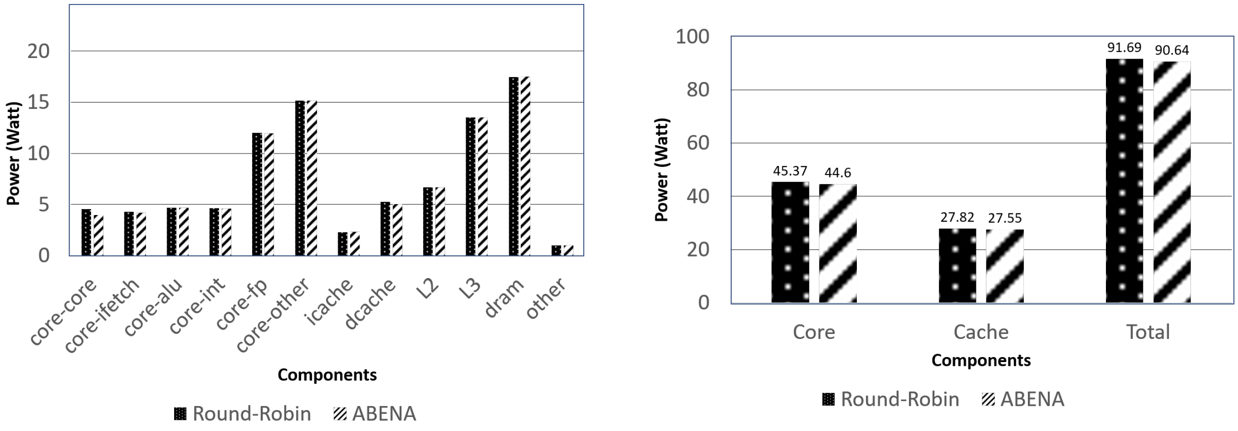
5.0.3 Temperature Evaluation

Fig 6a shows that ABENA has improved the temperature on the many-core architecture compared to the round-robin algorithm in Fig 6b. The highest temperature with ABENA is 80° compared to 104° with the round-robin algorithm. This decrease in temperature will help extend the lifetime of nodes and prevent hotspot.



(a) Temperature of 4x4 Mapping with ABENA

(b) Temperature of 4x4 Mapping with Round-Robin



(a) Power Consumption (W) of Round-Robin & Proposed Architecture Components

(b) Total Power Consumption (W) of Round-Robin & Proposed Architecture

5.0.4 Power Evaluation

Fig 7a shows that both algorithms have similar power consumption. Fig 7b shows the overall total power consumption of both algorithms. Apart from the core-core consumption, all other component has very similar power consumption patterns. However, there is a 2% power reduction in the amount of power consumed by ABENA. Although, this amount appears insignificant, in many core systems consisting of 1000 cores, it can be predicted that, this consumption will prove vital.

5.0.5 Summary of Contribution

Table 2 shows the comparison between both algorithms. It can be deduced that, ABENA improves the temperature and Hotspot whilst also consuming the similar power patterns compared to the round-robin algorithms.

Table 2: Summary of Contributions

Parameter	**Round-Robin**	**ABENA**
Power Consumption	Medium	Medium
Hotspot	Medium	Low
Ageing	High	Low
Temperature	Medium	Low

6. CONCLUSION

In this paper, we presented an ageing-aware algorithm which uses the lifetime of nodes to distribute application threads across the chip. Results show that, the proposed algorithm effectively prevent hostpot and high temperature which significantly improve the lifetime of the nodes. Compared to the conventional round-robin algorithm, our proposed architecture extends the lifetime of the nodes. The method of only allowing one node active out of two adjacent nodes automatically reduces the temperature of the node when the non-active node is used as a dumso node. This helps keep the average chip of the chip low.

REFERENCES

[1] M. Shafique, S. Garg, T. Mitra, S. Parameswaran, and J. Henkel, "Dark silicon as a challenge for hardware/software co-design: Invited special session paper," in *Proceedings of the International Conference on Hardware/Software Codesign and System Synthesis, CODES '14*, 2014.

[2] D. Juan, S. Garg, J. Park, and D. Marculescu, "Learning the optimal operating point for many-core systems with extended range voltage/frequency scaling," in *International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, 2013.

[3] A. Kanduri, M. Haghbayan, A. Rahmani, P. Liljeberg, A. Jantsch, and H. Tenhunen, "Dark silicon aware runtime mapping for many-core systems: A patterning approach," in *33rd IEEE International Conference on Computer Design (ICCD)*, 2015.

[4] T. S. Muthukaruppan, M. Pricopi, V. Venkataramani, T. Mitra, and S. Vishin, "Hierarchical power management for asymmetric multi-core in dark silicon era," in *50th ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2013.

[5] T. S. Muthukaruppan, M. Pricopi, V. Venkataramani, T. Mitra, and S. Vishin, "Hierarchical power management for asymmetric multi-core in dark silicon era," in *50th ACM/EDAC/IEEE Design Automation Conference (DAC)*, pp. 1–9, 2013.

[6] M. H. Haghbayan, A. Miele, A. M. Rahmani, P. Liljeberg, and H. Tenhunen, "A lifetime-aware runtime mapping approach for many-core systems in the dark silicon era," in *Design, Automation Test in Europe Conference Exhibition (DATE)*, pp. 854–857, 2016.

[7] C. Feng, Z. Liao, Z. Lu, A. Jantsch, and Z. Zhao, "Performance analysis of on-chip bufferless router with multi-ejection ports," in *In IEEE 11th International Conference on ASIC (ASICON)*, pp. 1–4, 2015.

[8] C. Fallin, C. Craik, and O. Mutlu, "Chipper: A low-complexity bufferless deflection router," in *In IEEE 17th International Symposium on High Performance Computer Architecture*, pp. 144–155, 2011.

[9] B. K. Daya, L. S. Peh, and A. P. Chandrakasan, "Towards high-performance bufferless nocs with scepter," *In IEEE Computer Architecture Letters* **15**(1), pp. 62–65, 2016.

[10] C. Li and P. Ampadu, "A compact low-power edram-based noc buffer," in *In Low Power Electronics and Design (ISLPED), IEEE/ACM International Symposium on*, pp. 116–121, 2015.

[11] N. Nasirian and M. Bayoumi, "Low-latency power-efficient adaptive router design for network-on-chip," in *In 28th IEEE International System-on-Chip Conference (SOCC)*, pp. 287–291, 2015.

[12] L. Yang, W. Liu, N. Guan, M. Li, P. Chen, and E. H. M. Sha, "Dark silicon-aware hardware-software collaborated design for heterogeneous many-core systems," in *22nd Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2017.

[13] M. F. Reza, D. Zhao, and M. Bayoumi, "Power- thermal aware balanced task-resource co-allocation in heterogeneous many cpu-gpu cores noc in dark silicon era," in *31st IEEE International System-on-Chip Conference (SOCC)*, 2018.

[14] Y. Zhang, L. Peng, X. Fu, and Y. Hu, "Lighting the dark silicon by exploiting heterogeneity on future processors," in *ACM/EDAC/IEEE Design Automation Conference (DAC)*, pp. 1–7, 2013.

[15] W. Liu, L. Yang, W. Jiang, L. Feng, N. Guan, W. Zhang, and N. Dutt, "Thermal-aware task mapping on dynamically reconfigurable network-on-chip based multiprocessor system-on-chip," *IEEE Transactions on Computers* , 2018.

[16] K. A. Bowman, S. G. Duvall, and J. D. Meindl, "Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration," *IEEE Journal of Solid-State Circuits* , 2002.

[17] X. Wang, A. K. Singh, B. Li, Y. Yang, H. Li, and T. Mak, "Bubble budgeting: Throughput optimization for dynamic workloads by exploiting dark cores in many core systems," *IEEE Transactions on Computers* , 2018.

[18] A. kanduri, M. haghbayan, A. M. Rahmani, M. Shafique, A. Jantsch, and P. Liljeberg, "adboost: Thermal aware performance boosting through dark silicon patterning," *IEEE Transactions on Computers* , 2018.

[19] A. M. Rahmani, M. Haghbayan, A. Miele, P. Liljeberg, A. Jantsch, and H. Tenhunen, "Reliability-aware runtime power management for many-core systems in the dark silicon era," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* , 2017.

[20] V. Rathore, V. Chaturvedi, A. K. Singh, T. Srikanthan, R. Rohith, S. Lam, and M. Shaflque, "Himap: A hierarchical mapping approach for enhancing lifetime reliability of dark silicon manycore systems," in *Design, Automation Test in Europe Conference Exhibition (DATE)*, 2018.

[21] M. S. Mohammed, A. K. Al-Dhamari, A. A. ab Rahman, N. Paraman, A. A. M. Al-Kubati, and M. N. Marsono, "Temperature-aware task scheduling for dark silicon many-core system-on-chip," in *8th International Conference on Modeling Simulation and Applied Optimization (ICMSAO)*, 2019.

[22] E. Ofori-Attah, X. Wang, and M. O. Agyeman, “A survey of system level power management schemes in the dark-silicon era for many-core architectures,” *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems* **5**, 9 2018.

[23] R. Rohith, V. Rathore, V. Chaturvedi, A. K. Singh, S. Thambipillai, and S. Lam, “Lifesim: A lifetime reliability simulator for manycore systems,” in *IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, 2018.

[24] T. E. Carlson, W. Heirman, S. Eyerman, I. Hur, and L. Eeckhout, “An evaluation of high-level mechanistic core models,” *ACM Transactions on Architecture and Code Optimization (TACO)* , 2014.

[25] 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 *42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2009.

[26] M. R. S. R. Zhang and K. Skadron, “Hotspot 6.0: Validation, acceleration and extension,” in *University of Virginia, Tech. Rep*, 2015.