

Multi-Objective Task Mapping Approach for Wireless NoC in Dark Silicon Age*

Amin Rezaei¹, Dan Zhao², Masoud Daneshtalab³, Hai Zhou¹¹ Northwestern University (NU), Evanston, USA (me@aminrezaei.com, haizhou@northwestern.edu)² Old Dominion University (ODU), Norfolk, USA (zhao@cs.odu.edu)³ Mälardalen University (MDH) and Royal Institute of Technology (KTH), Sweden (masdan@kth.se)

Abstract— Hybrid Wireless Network-on-Chip (HWNOC) provides high bandwidth, low latency and flexible topology configurations, making this emerging technology a scalable communication fabric for future Many-Core System-on-Chips (MCSOCs). On the other hand, dark silicon is dominating the chip footage of upcoming MCSOCs since Dennard scaling fails due to the voltage scaling problem that results in higher power densities. Moreover, congestion avoidance and hot-spot prevention are two important challenges of HWNoC-based MCSOCs in dark silicon age; Therefore, in this paper, a novel task mapping approach for HWNoC is introduced in order to first balance the usage of wireless links by avoiding congestion over wireless routers and second spread temperature across the whole chip by utilizing dark silicon. Simulation results show significant improvement in both congestion and temperature control of the system, compared to state-of-the-art works.

Keywords—Wireless NoC; Dark Silicon; Mapping; Temperature

I. INTRODUCTION

Based on the ITRS report [1], the integration of different technologies in a limited space has revolutionized the semiconductor industry by shifting the main purpose of chip design from a performance-driven approach to a multi-objective one. Moreover, dark silicon is endorsing the above report by dominating the chip footage of future Many-Core System-on-Chips (MCSOCs) since Dennard scaling [2] fails due to the voltage scaling problem that results in higher power densities [3]. Nowadays, commercial MCSOCs are available based on Network-on-Chip (NoC) [4] communication infrastructure [5-7]. Despite the fact that a NoC-based architecture has many advantages, its multi-hop nature has negative impact on both latency and power consumption parameters especially when the network size increases; Therefore, alternative technologies such as Hybrid Wireless NoC (HWNOC) have been introduced [8]. HWNoC provides high bandwidth, low latency, and flexible topology configurations, making this emerging technology a suitable candidate for communication fabric of future MCSOCs. However high energy costs of the Wireless Routers (i.e. the routers equipped with wireless transceivers, WRs) in comparison with the Conventional Routers (CRs) not only limits the integration of WRs on a single chip, but also introduces a direct confrontation with the dark silicon utilization wall [9] in which sustained chip performance is limited primarily by power rather than area.

On the other hand, by employing limited number of WRs, they are more vulnerable to congestion since far apart traffics intend to utilize wireless express links which result in high wireless channel competitions. In fact, the network congestion not only increases the network latency severely [10] but rather raises the network power consumption significantly [11]. Moreover, among all the challenges the semiconductor industry faces, keeping future MCSOCs cool has a high priority, since overheating causes significant reductions in the operating life of a device leading to device failure. Along the same lines, in this paper, we propose a novel temperature and congestion aware task mapping algorithm named Round Rotary Mapping (RRM), targeted at tackling two critical concerns in HWNoC in dark silicon age: Alleviation of the severe congestion on WRs and prevention of persistent hot-spots in the network. The rest of the paper is organized as follows. Section II reviews backgrounds and related works. The preliminaries and motivations of the proposed mapping approach are presented in Section III. The baseline architecture along with the RRM algorithm is proposed in Section IV and Section V. The experimental results are shown in Section VI. Finally, the conclusion is given in Section VII.

II. BACKGROUNDS AND RELATED WORKS

One of the shortcomings of wireless NoC is extensive power overhead that wireless transceivers impose to the system. Thus, instead

of a single NoC spanning the entire system, HWNoC has been proposed using both wired and wireless links [8]. Furthermore, a hierarchical wireless-based NoC Architecture along with performance evaluation parameters has been introduced where the network is divided into subnets [12]. Intra-subnet nodes communicate through wire links while inter-subnet communications are handled by wireless express links. Also, a WR placement has been proposed for HWNoC to allocate optimal number of WRs across the network [13]. Even though by applying an optimized WR placement, a trade-off between performance and power consumption parameters can be achieved, still two more problems remain unsolved. First, by placing limited number of WRs, they are highly vulnerable to congestion since each WR is shared by many traffics within the network. Second, the trade-off is based on an offline profiling and in the case of changing applications at run-time it may not be an accurate placement.

On top of that, HWNoC-based MCSOCs face extremely dynamic workloads where applications, as sets of communicating tasks, enter and leave the system at run-time. Hence, an efficient task mapping technique is required to balance the utilization of available WRs. In [14] a dynamic application mapping algorithm (DMA) has been presented and evaluated for HWNoC. However, since it tries to map the applications as close as possible to the Central Manager (CM) core, a hot-spot area near the CM is created. Persistent hot-spots in the system, increases the permanent failure probability of those highly active cores. On the other hand, recent studies on future trends in dark silicon [15] have predicted that, on average, 52% of a chip's area will stay dark for the 8nm technology node. (i.e. on average only 48% of all the cores in a single chip can be powered on simultaneously.) Moreover, the dark silicon ratio is increasing by technology scaling.

III. PRELIMINARIES AND MOTIVATIONS

A 4×4 HWNoC with two WRs (Fig. 1) is simulated using random task mapping and congestion-aware dynamic task mapping presented in [14] to show the necessity of both congestion avoidance and hot-spot prevention. Several applications, each with 2 to 5 tasks are randomly generated. Each application is considered to have 75% intra-communication among its tasks and 25% inter-communication with the tasks of other applications. Applications are scheduled based on the First-Come-First-Serve (FCFS) policy and the maximum possible scheduling rate is λ_{full} . An allocation request for the scheduled application is sent to the Central Manager (CM) of the system. CM keeps track of the free cores to map the new tasks.

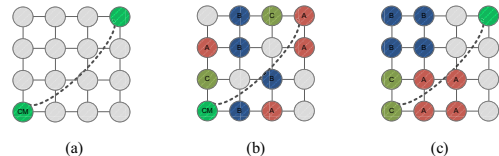


Fig. 1. (a) 4×4 HWNoC with two WRs using (b) Random and (c) Dynamic mappings

A. Congestion

Fig. 2 shows the average network latency comparison between random and dynamic task mapping schemes in 4×4 HWNoC. In random task mapping, the traffic is not evenly distributed in the network, resulting in high congestion surrounding WRs which degrades the network performance significantly. On the other hand, in dynamic task mapping, the traffic is more balanced globally, resulting in great average latency reduction.

B. Hot-Spot

The thermal analysis of 4×4 HWNoC with random and dynamic task mapping in $0.5 \lambda_{full}$ is depicted in Fig. 3a and Fig. 3b respectively. In

* This work is partially supported by NSF under 1441695 and 1533656.

random task mapping, the hot-spot regions are observed around the WRs since most of the traffics are moving toward them. With dynamic task mapping, although the congestion around WRs is decreased, the hot-spot problem is getting worse around CM because the applications are contiguously mapped as close as possible to CM.

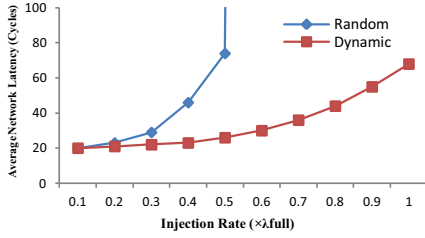


Fig. 2. Average network latency in 4×4 HWNoC with random and dynamic mappings

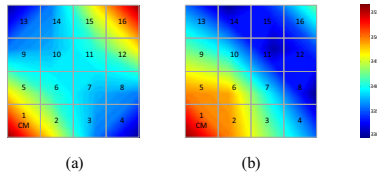


Fig. 3. Thermal distribution for 4×4 HWNoC (a) Random and (b) Dynamic mappings

Based on the above discussions, a reliable mapping is essential for HWNoC that not only improves network performance by reducing severe congestion around WRs, but rather achieves energy efficiency by preventing hot-spots in the network. In other words, we may combine the performance gain obtained by dynamic task mapping shown in Fig. 2 with a temperature-aware method to avoid hot-spots depicted in Fig. 3.

IV. SYSTEM ARCHITECTURE

In this section, we overview the system configuration under study along with the application representation.

A. System Configuration

Without loss of generality, a 2D mesh HWNoC is virtually divided into several regions where the number of regions equals to the number of available WRs. The WRs are interconnected to form a wireless highway if they fall within each other transmission range. Thus, in each dedicated region only one WR exists and the WR is associated to serve as the access point to the highway. For network efficiency, HWNoC is partitioned in a way that any core within a region has the minimum hop-count towards the WR of that region than the WRs of the other regions. For borderline cases that a core may have the same hop-count from two or more WRs, the core will be randomly assigned to one of the candidate regions. Fig. 4 shows two 64-core HWNoC-based MCSoc architectures.

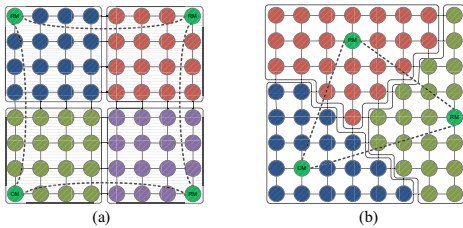


Fig. 4. 8×8 HWNoC (a) 4 and (b) 3 regions

Moreover, regardless of the number of regions, four Cartesian coordinate systems are defined as down-left (DL), top-left (TL), top-right (TR), and down-right (DR) shown in Fig. 5. Origin of each coordinate system is one of the four corners of the network. At each moment, there is one active region (*active_R*) along with one active coordinate system (*active_C*). Furthermore, the WRs are equipped with the control logic to manage the application mapping within their regions. One of the WRs is assigned as CM and the other WRs are named Regional Managers (RMs). Since each manager is responsible to assign the tasks on its own

region, the hierarchical managing scheme helps balance the workload distribution between different managers.

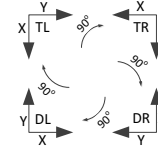


Fig. 5. Cartesian coordinate systems

B. Application Representation

In many embedded system applications like robotics, biomedical systems, and multimedia control systems, not only the tasks of each application are able to communicate with each other (i.e. intra-communication), but also multiple applications may also communicate with each other (i.e. inter-communication).

Intra-communication: An undirected graph, naming Task Graph (TG), represents each application and intra-communication between its tasks. Each vertex denotes one task of the application, while each edge stands for communication between each two tasks as given in Equation 1. The TGs of four applications are shown in Fig. 6. The amount of data transferred between any two tasks is indicated on the edge.

$$\forall t_i \in T, \forall e_{i,j} \in E, app = TG(T, E) \quad (1)$$

Inter-communication: Since the applications are considered to enter the system at run-time, no static graph can be defined for inter-communication between different applications. An incoming application may request to communicate with an already mapped application. Moreover, an existing application in the system may ask to communicate with an application which is not yet mapped onto the system. Thus, the inter-communication graph is highly dynamic.

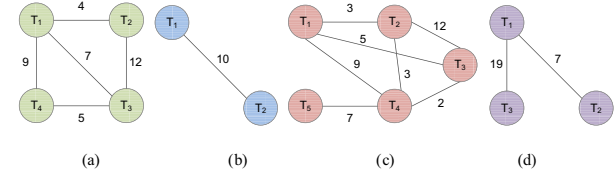


Fig. 6. Task graphs of different applications

V. ROUND ROTARY MAPPING

In this section we propose a novel mapping algorithm which aims at evenly distributing the heat across the whole chip while reducing congestion in around of WRs. The proposed algorithm named *Round Rotary Mapping (RRM)* tries to map incoming applications region by region in a round-robin manner to balance the thermal distribution globally while periodically rotate the Cartesian coordinate system to balance the thermal distribution within each region locally. On the other hand, the tasks of each application are mapped with regard to the minimum Hop-Count Contiguity (HCC) in order to reduce congestion caused by long distance communications of the same tasks of each application. Algorithm 1 represents the RRM algorithm.

In *Initialization()* function, the number of regions (i.e. r), set of regions (i.e. $R = \{R_1, R_2, \dots, R_r\}$), and set of coordinate systems (i.e. $C = \{DL, TL, TR, DR\}$) are initialized. Moreover, the active region and active coordinate system are also initialized to the first element of each set (i.e. $active_R = R_1$ and $active_C = DL$). Then, applications are chosen based on the FCFS policy since no background information is considered about incoming applications. In case of having background information about the incoming applications, an appropriate application selection policy can be applied which is left as a future work.

In each region, RRM first tries to find the set of free cores with the smallest 'Y's. Then among them, the core with the smallest 'X' is chosen in order to map the first task of the application. The first task of each application is returned from the *FirstTask(app, free_XY)* function. This function returns the task of the selected application (i.e. app) with

the equal or smaller number of edges than the available free cores around the chosen core (i.e. $free_XY$). If there is more than one task with the aforementioned criteria, then the first task would be the one with the most intensive communication among the candidates. If there is no task with the equal or smaller edges as the available free cores around $free_XY$, the task with the least intensive communication among all the tasks is chosen. In the case of two or more candidates with the same characteristics, one of them is randomly chosen.

ALGORITHM 1. RRM ALGORITHM

```

r: number of regions (i.e. number of WRs)
C: set of coordinate systems = {DL, TL, TR, DR}
R: set of regions = {R1, R2, ..., Rr}
A: set of applications
active_C: active coordinate system
active_R: active region

Initialization();
while true
  if A is empty then
    Sleep();
  end
  app = choose an unmapped application from A;
  free_Y = choose the set of free cores with the smallest Y;
  free_XY = choose the free core with the smallest X from free_Y;
  ft = FirstTask(app, free_XY);
  Map(ft, free_XY);
  MinHCC(app, free_XY);
  if active_R == Rr, then
    active_C = choose the next coordinate system from C;
  end
  active_R = choose the next region from R;
end

```

Table I shows the first task selection for different values of available free cores around $free_XY$ for the four applications of Fig. 6. For example, in application A (i.e. Fig. 6a) if the number of available free cores around $free_XY$ is '1', there is no task with the equal or smaller number of edges as '1'; Thus the task with the least intensive communication among all the tasks of the application (i.e. T_4) is selected. However, if the available free cores are '2', two candidates (i.e. T_2 and T_4) have equal or smaller number of edges than '2'; Among them, T_2 is selected because it has more intensive communications than T_4 (i.e. 16 vs. 14). If the number of available cores is equal or greater than '3', the task with the most intensive communications (i.e. T_3) will be chosen. Note that in a mesh-based NoC the maximum available cores around each core are eight. Also in each region, only the free cores within that region are considered.

TABLE I. FIRST TASK SELECTION EXAMPLE

# of Free Cores	0	1	2	3	4	5	6	7	8
App. A: Fig. 6a	T_4	T_4	T_2	T_3	T_3	T_3	T_3	T_3	T_3
App. B: Fig. 6b	T_1/T_2	T_1/T_2	T_1/T_2	T_1/T_2	T_1/T_2	T_1/T_2	T_1/T_2	T_1/T_2	T_1/T_2
App. C: Fig. 6c	T_5	T_5	T_5	T_3	T_4	T_4	T_4	T_4	T_4
App. D: Fig. 6d	T_2	T_3	T_1	T_1	T_1	T_1	T_1	T_1	T_1

After mapping the first task to $free_XY$, RRM tries to map the other tasks of the application based on minimum HCC around the first task within that region. Minimum HCC is defined as minimum overall hop-count between all the cores in which the application is mapped into. In the case that the application does not fit into the current region, the current region will be merged with the next region temporarily.

After mapping of each application, the active region is shifted to the next region to balance the thermal distribution globally. Moreover, after a complete round (i.e. all the regions are became active once in one coordinate system) the origin of the coordinate system is rotated to the next origin to balance the thermal distribution within each region as well. Note that when the RRM algorithm reaches the last element of R (or C), it starts from the beginning again, i.e. $active_R$ is set to R_1 (or $active_C$ is set to DL). In the case that there is no available application to be mapped into the system, RRM goes to the $Sleep()$ mode until a new application arrives and signals to wake up. Overall, RRM tries to map

the task of each application as contiguous as possible based on minimum HCC to avoid long distance communications that mostly influence the WRs. Also it tries to spread the temperature across the chip by periodically changing the regions (i.e. global heat distribution) and coordinate systems (i.e. local heat distribution).

VI. EXPERIMENTAL RESULTS

Experiments are performed on a many-core platform implemented in SystemC. A pruned version of an open source simulator for mesh-based NoCs called Noxim [16] is utilized as its communication architecture. Thermal model taken from [17] is integrated as a library into the simulator. Several sets of applications each with 2 to 5 tasks are generated using TGG [18] where the amount of data transferred from the source task to the destination task are randomly distributed between 2 to 36 flits of data. Each application is considered 75% intra-communication and 25% inter-communication between the other applications. The inter-communication between different applications is conducted by the first task of each application mapped to the system. Applications are scheduled based on the FCFS policy and the maximum possible scheduling rate is λ_{full} . An allocation request for the scheduled application is sent to CM of the system. CM then based on the active region sends the information to the responsible RM through the hierarchical managing network. The hierarchical XY routing algorithm taken from [13] is implemented. Two 64-core HWNoC (52% dark silicon) with three and four WRs (Fig. 4) are considered in the simulations. Comparisons are also made between RRM and random task mapping as baseline in addition to the congestion-aware dynamic task mapping algorithm (DMA) presented in [14].

A. Hop-Counts and Energy Saving

As shown in [19], decreasing Manhattan Distance (MD) between tasks of application edges is an effective way to minimize the communication energy consumption of the applications. The percentage of packets that are delivered over different path lengths (i.e. MD) is illustrated in Fig. 7. The experiments have been run for different algorithms in the injection rate of $0.5 \lambda_{full}$. As can be seen, more than 60% (and more than 50%) of the packets are delivered by one-hop distance using RRM algorithm in 64-core HWNoC with four regions (and three regions). Accordingly, Table II represents the average MD for different algorithms in the injection rate of $0.5 \lambda_{full}$ based on different percentages of the intra-communication and inter-communication. By decreasing the percentage of inter-communication between applications, more energy can be saved by RRM since it maps all the tasks of each application based on minimum HCC. RRM outperforms DMA in less than 5% inter-communication between different applications.

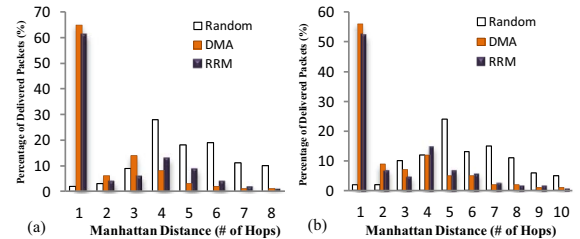


Fig. 7. Percentage of delivered packets in 64-core HWNoC (a) 4 and (b) 3 regions

TABLE II. AVERAGE MANHATTAN DISTANCE COMPARISON

Intra-com (%) / Inter-com (%)		70/30	75/25	80/20	85/15	90/10	95/5	100/0
64-core HWNoC with four regions	Random	5.01	5.08	5.21	5.36	5.14	5.2	5.07
	DMA	2.34	1.93	1.61	1.31	1.13	0.93	0.78
	RRM	2.78	2.3	1.84	1.55	1.27	0.92	0.66
64-core HWNoC with three regions	Random	5.61	5.79	5.6	5.87	6.03	5.92	5.72
	DMA	2.98	2.47	2.07	1.68	1.4	1.16	0.96
	RRM	3.21	2.77	2.2	1.82	1.49	1.09	0.78

B. Network Latency and Congestion Avoidance

Fig. 8 shows the average network latency for different algorithms. It is supposed that there is no gap between application arrivals. As can be seen, RRM has a reasonable average network latency next to DMA. Contiguous mapping for the tasks of each application in both DMA and RRM results in lower network latency than random mapping. Also, in DMA and RRM, by increasing the injection rate, the network becomes uniformly congested because the usage of WRs is more balanced.

C. System Utilization

System utilization is another parameter has been analyzed among the different algorithms. As shown in Fig. 9, RRM has lower average system utilization than DMA mapping but have better maximum system utilization that is defined as the highest percentage of the utilization during the simulation time. Note that the system utilization is based on the number of tasks that can be mapped on non-dark cores which communicate with each other without dropping due to the high congestion. This happens because DMA tries to map not only all the tasks of each application but also all the applications contiguous (i.e. close to CM) that results in better average system utilization. On the other hand, RRM unlike DMA does not suffer from area fragmentation and can reach higher maximum utilization (i.e. almost 98%).

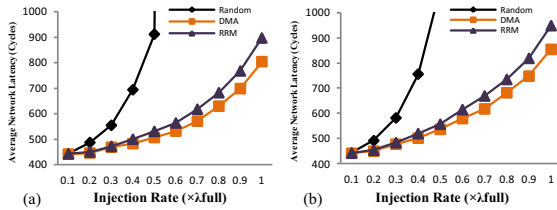


Fig. 8. Average network latency in 64-core HWNoC (a) 4 and (b) 3 regions

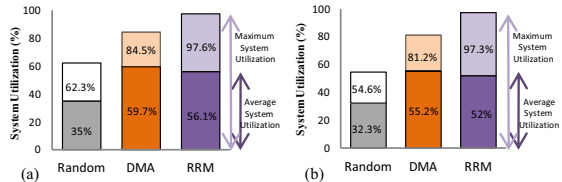


Fig. 9. System utilization in 64-core HWNoC (a) 4 and (b) 3 regions

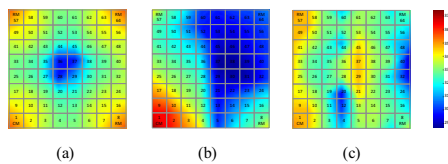


Fig. 10. Thermal analysis in 64-core HWNoC with 4 regions (a) Random (b) DMA (c) RRM

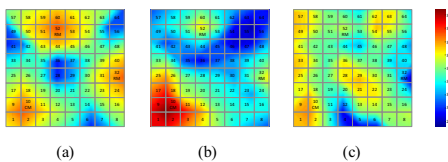


Fig. 11. Thermal analysis in 64-core HWNoC with 3 regions (a) Random (b) DMA (c) RRM

D. Thermal Analysis

Fig. 10 and Fig. 11 demonstrate the thermal distribution of different algorithms in the maximum system utilization based on Fig. 9. Furthermore, Table III shows the average and the peak temperature comparisons. Since in dark silicon age more than half of the chip is dark, RRM utilizes the dark cores in order to efficiently avoid hot-spots in the system. On the other hand, DMA suffers from severe hot-spot around CM and random mapping has multiple hot-spots around WRs. Moreover, unlike RRM, the peak temperature is gotten worse in DMA by decreasing the number of WRs.

TABLE III. AVERAGE AND PEAK TEMPERATURE COMPARISON

		Random	DMA	RRM
64-core HWNoC with four regions	Average temperature (K)	333.6	318.1	329.6
	Peak temperature (K)	356.4	372.8	351.7
64-core HWNoC with three regions	Average temperature (K)	330.1	323.5	329.9
	Peak temperature (K)	357.1	375.2	351.4

VII. CONCLUSION

In this paper, a novel temperature and congestion aware task mapping algorithm named RRM was introduced in order to solve some of the key concerns in future HWNoC-based MCSoCs. Simulation results showed significant improvement in both congestion and temperature control of the system. Contiguous mapping for the tasks of each application results in lower network latency and finally total execution time gain. Moreover, the heat is distributed evenly across the whole chip using the proposed algorithm.

REFERENCES

- [1] ITRS. International Technology Roadmap for Semiconductors, 2013 edition.
- [2] R. H. Dennard, F. H. Gaensslen, H. N. Yu, V. L. Rideout, E. Bassous, and A. R. Leblanc, "Design of ion-implanted MOSFET's with very small physical dimensions," in *IEEE Journal of Solid-State Circuits*, Vol. 9, pp. 256-268, 1974.
- [3] A. Rezaei, D. Zhao, M. Daneshalab, and H. Wu, "Shift sprinting: fine-grained temperature-aware NoC-based MCSoC architecture in dark silicon age," in *ACM/EDAC/IEEE Design Automation Conference (DAC)*, Article 155, 2016.
- [4] L. Benini and G. De Micheli, "Networks on chips: a new SoC paradigm," in *IEEE Computer*, Vol. 35, Issue 1, pp. 70-78, 2002.
- [5] "Adapteva, Inc." [Online]. Available: <http://www.adapteva.com/>.
- [6] "Arteris, Inc." [Online]. Available: <http://www.arteris.com/>.
- [7] "Sonics, Inc." [Online]. Available: <http://sonicsinc.com/>.
- [8] A. Ganguly, K. Chang, S. Deb, P. P. Pande, B. Belzer, and C. Teuscher, "Scalable hybrid wireless network-on-chip architectures for multicore systems," in *IEEE Transactions on Computers*, Vol. 60, Issue 10, pp. 1485-1502, 2011.
- [9] N. Goulding-Hotta, J. Sampson, Q. Zheng, V. Bhatt, J. Auricchio, S. Swanson, and M. B. Taylor, "GreenDroid: an architecture for the dark silicon age," in *Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 100-105, 2012.
- [10] J. W. Brand, C. Ciordas, K. Goossens, and T. Basten, "Congestion-controlled best-effort communication for networks-on-chip," in *Proceedings of Design, Automation and Test in Europe (DATE)*, pp. 1-6, 2007.
- [11] S. Ma, N. E. Jerger, and Z. Wang, "DBAR: an efficient routing algorithm to support multiple concurrent applications in networks-on-chip," in *Proceedings of International Symposium on Computer Architecture (ISCA)*, pp. 413-424, 2011.
- [12] A. Rezaei, F. Safaei, M. Daneshalab, and H. Tenhunen, "HiWA: a hierarchical wireless network-on-chip architecture," in *IEEE International Conference on High Performance Computing & Simulation (HPCS)*, pp. 499-505, 2014.
- [13] A. Rezaei, M. Daneshalab, F. Safaei, and D. Zhao, "Hierarchical approach for hybrid wireless network-on-chip in many-core era," in *Elsevier International Journal of Computers and Electrical Engineering (COMPELEC-Elsevier)*, Vol. 51, Issue C, pp. 225-234, 2016.
- [14] A. Rezaei, M. Daneshalab, D. Zhao, F. Safaei, X. Wang, and M. Ebrahimi, "Dynamic application mapping algorithm for wireless network-on-chip," in *Euromicro International Conference on Parallel, Distributed and Network-Based Computing (PDP)*, pp. 421-424, 2015.
- [15] J. Henkel, H. Khdr, S. Pagani, and M. Shafique, "New trends in dark silicon," in *ACM/EDAC/IEEE Design Automation Conference (DAC)*, Article 119, 2015.
- [16] V. Catania, A. Mineo, S. Monteleone, M. Palesi, D. Patti, "Noxim: an open, extensible and cycle-accurate network on chip simulator," in *IEEE International Conference on Application-specific Systems, Architectures and Processors (ASAP)*, pp. 162-163, 2015.
- [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," in *IEEE Transaction on Very Large Scale Integration (VLSI) Systems*, Vol.14, Issue 5, pp. 501-513, 2006.
- [18] "Task graph generator (TGG)." [Online]. Available: <http://sourceforge.net/projects/taskgraphgen/>.
- [19] C. L. Chou, U. Y. Ogras, and R. Marculescu, "Energy- and performance-aware incremental mapping for networks on chip with multiple voltage levels," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 27, Issue 10, pp. 1866-1879, 2008.