Malleable NoC: Dark Silicon Inspired Adaptable Network-on-Chip

Haseeb Bokhari† Haris Javaid§ Muhammad Shafique‡ Jörg Henkel‡
Sri Parameswaran†
†School of Computer Science and Engineering, University of New South Wales, Sydney, Australia
§Google Inc., USA
‡Chair for Embedded Systems, Karlsruhe Institute of Technology, Karlsruhe, Germany
{hbokhari, sridevan}@cse.unsw.edu.au,harisjavaid@google.com, {muhammad.shafique, henkel}@kit.edu

ABSTRACT

Network on Chip (NoC) has been envisioned as a scalable fabric for many core chips. However, NoCs can consume a considerable share of chip power. Moreover, diverse applications are executed in these multicore, where each application imposes a unique load on the NoC. To realise a NoC which is Energy and Delay efficient, we propose combining multiple VF optimized routers for each node (in traditional NoCs, we have only a single router per node) for efficient NoC for Dark Silicon chips. We present a generic NoC with routers designed for different VF levels, which are distributed across the chip. At runtime, depending on application profile, we combine these VF optimized routers to form constantly changing energy efficient NoC fabric. We call our architecture *Malleable NoC*. In this paper, we describe the architectural details of the proposed architecture and the runtime algorithms required to dynamically adapt the NoC resources. We show that for a variety of multi program benchmarks executing on *Malleable NoC*, Energy Delay product (EDP) can be reduced by up to 46% for widely differing workloads. We further show the effect on EDP savings for differing amounts of dark silicon area budget.

1. INTRODUCTION

Network-on-Chips(NoCs) are used in multi-core chips as a scalable communication fabric. As technology nodes shrink and number of cores in chip increase, designers continue to face the challenging task of balancing NoC power and performance. NoC can consume considerable share of chip power, e.g., up to 18% and 33% in Intel SCC [1] and RAW [2] architectures, respectively. It is anticipated that the share of NoC energy will increase as the technology nodes shrink [3]. Thus, researchers have proposed using power saving techniques such as Dynamic voltage and frequency (DVFS) [4, 5] for NoC. Three aspects motivate this work:

• The first aspect is that general purpose multi-cores execute a diverse set of applications. Each application interacts with cache architecture uniquely, hence the NoC traffic injection rate can vary across all applications. We observed in our experiments that misses per kilo instruction (MPKI) is very different for the two applications *H264*, and *MPEG2ENC*. On average the MPKI for *H264* is 40× the MPKI of *MPEG2ENC*. Due to higher cache miss rate, *H264* application requires a NoC clocked at higher frequnecy. On the other hand, *MPEG2ENC* has a very low cache miss rate, hence providing an opportunity to save energy by clocking the NoC at a lower frequency. However, simultaneous execution of the two application will require the NoC to be executed at a higher frequency to support the worst case application.

• The second aspect is that in addition to the applicable MPKIs, the application to core mapping in mesh NoC based multi-core also have a a significant impact on performance and energy. An example 4×4 mesh architecture is shown in Figure 1 in which cache load and store misses from the cores get serviced by the memory controller. The cache load miss penalty depends on the hop distance between the core and the memory controller. The two applications discussed earlier can be mapped to either core A or core B. If *H264* is mapped on core B, NoC frequency will not significantly effect the system performance. However, if *H264* is mapped on core A, NoC frequency could play a vital role as *H264* has a higher MPKI. On the other hand, due to lower MPKI of *MPEG2ENC*, mapping *MPEG2ENC* on core A or B doesn't matter, and irrespective of mapping, NoC can be operated at lower frequency. Intuitively, if the two application have to be mapped at the same time, *H264* and *MPEG2ENC* will be mapped on core B and A respectively. Given such mapping, the grey routers can be clocked at lower frequency, while green routers can be clocked at higher frequency. Therefore, per router DVFS (where certain routers work faster, while others run slower at the same time) can be useful for executing heterogeneous applications.

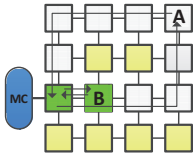


Figure 1: Application Mapping for 4×4 Mesh NoC (MC: Memory Controller)

• Although DVFS is a useful technique, the effectiveness of this techniques has been reduced with shrinking node size/technology. Leakage power is a major source of total NoC power (and is expected to grow as a proportion in the future). This leakage power largely remains unaffected with reduction in operating voltage. To counter this phenomena, darkNoC[6] was recently proposed which uses redundant area in dark silicon to complement system level DVFS by the use of muti-Vt based multiple routers instead of a single router optimized for the highest frequency. However, the proposed architecture in [6] is limited to cases where all routers lie in a single VF island. Thus the third and final aspect is the need for a NoC where the routers can work simultaneously at differing frequencies and utilize the dark silicon area to save energy.

1.1 Key Idea

Based on our observations, we introduce a new and novel low power NoC for dark silicon chips. We call this architecture *Malleable NoC*. At each mesh node (where traditionally only one router lies), we integrate *architecturally homogenous, yet circuit wise heterogeneous routers* designed for different frequencies and voltages that support pre-set VF switching of each router. Given such a NoC, at runtime, depending on dynamic profiling of application, we choose a different router from each node (hence the term *Malleable*), and connect these routers to create a low power NoC fabric. Composition of this NoC fabric adapts to the dynamic changes in the application phase. Routers that are not used in a particular phase remain switched off (*dark*). We further explore the fact that intelligent mapping of applications to cores can be exploited for efficient per node frequency and router selection.

1.2 Contributions

We make following contribution in this paper:

- We introduce dark silicon inspired NoC architecture called *Malleable NoCs*. We discuss the technical challenges that are required to realize such an architecture with per router VF switching control.

978-3-9815370-4-8/DATE15/©2015 EDAA

- We introduce a novel per router VF selection algorithm. This algorithm uses application profiling and mapping information for runtime management of *Malleable NoC* resources.
- We use full system simulations with diverse applications to show the efficacy of the proposed architecture. We show that our architecture is more useful when the applications have diverse MPKIs.

2. RELATED WORK

Recently various heterogenous NoCs have been proposed for general purpose multi-core chips. Mishra et al. [7] proposed integrating two architecturally heterogenous NoCs in a chip. Each NoC is designed using different operating frequency, data width and Virtual Chancel (VC) configuration, optimizing one NoC for low latency while the other NoC for high bandwidth. In contrast, our proposed architecture create only one NoC fabric out of various per node routers. Bakhoda et al. [8] proposed NoC with heterogenous routers using *checkerboard* configuration. In this proposed architecture, some of the regular routers are replaced by router with limited connectivity. Mishra et al. [9] also proposed heterogenous NoC at router granularity by integrating *big* and *small* routers that differ in terms of channel width and number of VCs. In contrast, we introduce heterogeneity by integrating architecturally homogenous but VF optimized routers at each node and switching on one router at each node depending on application profile.

DVFS for on-chip networks has been proposed at various granularity such as link level [10], router level [4, 5] and region level [11, 12, 6]. The DVFS algorithms used in [4, 5, 10] are oblivious to applications mapping and performance, and use NoC queue occupancy and/or network congestions for selecting frequency. Chen et al. [11] presented number of DVFS algorithms based on application profiling. However, these algorithms do not take into account scenarios where potentially heterogenous application are executed. Furthermore, all of these research studies assume fast on-chip voltage regulators which are impractical for per router DVFS due to low voltage conversion efficiency and design complexities [13]. In contrast , we consider both application to core mapping and application profiling to decide the optimum frequency at the granularity of router and explore the use of multiple supply voltage rails. To reduce NoC leakage power, run-time power gating techniques have been proposed [14, 15, 16, 17]. These techniques are orthogonal to our proposed architecture and can be used to further improve the energy efficiency.

3. Malleable NoC ARCHITECTURE

3.1 NoC Architecture

Our target architecture is a general purpose tiled multi-core chip as shown in Figure2, which closely match commercial SoC chips such as Intel Single Chip Cloud Computer(SCC) [1] architecture. We assume a processing core with private last level cache. Cache load and store requests are injected into the NoC by cores. Memory controllers are connected to border routers as shown in the Figure 2 to serve these load and store requests. We target a $N \times N$ Mesh topology based NoC architecture in this paper as it is commonly used by commercial and academic multi-core chips. Each core runs a different program therefore at a given time, up to N^2 applications are executed in parallel.

3.1.1 Per Router VF Selection

Normally high efficiency off-chip voltage regulators are used for scaling high supply voltage to low voltage for the purpose of DVFS (e.g, Intel SCC [1]). However, the voltage transition takes considerable time (e.g, up to 1 millisecond [1]), thus limiting the usefulness of DVFS scheme. Due to limited chip pins, using multiple off-chip voltage regulator is impractical for per router DVFS. There are two different approaches to support fast per-router DVFS: 1) use high switching frequency on-chip voltage regulators[18] for fine grain DVFS, or 2) use multi-voltage rails [13] for coarse-grain VF selection. On-chip voltage regulators enable fast voltage switching within 100-200 ns [18], at a cost of low conversion efficiency of about 75-80%. Multi-voltage rail (MVR) design enables fast coarse grain switching between different voltages without compromising efficiency at the cost of limited VF options. The voltages are generated through high efficiency off-chip voltage regulators. Miller et al. [13] showed that voltages can be switched within 7-10 ns at runtime.

In our architecture, we employ MVR for each router to realize per router voltage scaling. For frequency scaling, a global clock is generated and distributed using normal clock tree. At each node a scaled frequency is generated using local clock dividers. In this architecture the master frequency is scaled down in power of 2 only i.e., f , $f/2$ and $f/4$. This makes it easier to design and verify bi-synchronous FIFOs [19]. Router control is aware of operating frequency of neighbor routers to select correct frequency for input FIFO write logic. We implemented the per router VF switching by combining fast voltage switching using MVR and local frequency division.

3.1.2 Dark Silicon Inspired Multiple Routers

Due to limited power budget, large amount of silicon on deep sub-micron chips is expected to remain unpowered (*Dark Silicon*). However, researchers argue that *Dark Silicon* can be used to improve the energy efficiency of exiting chip components through employing clever circuit design techniques [20]. Multi-vt optimization is one of such techniques.

Fabrication foundries provide cell libraries with various gate threshold voltages (Vt), such as normal Vt (NVt), Low Vt (LVt), and High Vt (HVt). Low Vt cells can switch faster but consume high energy, whereas high Vt cell consume low energy but switch at much lower frequency, Through Multi-Vt Optimization, synthesis tools exploit multi-vt libraries to improve the power efficiency of circuits by inserting low Vt cells on circuit logic paths with negative slack to meet the latency constraint and replacing normal cells with high Vt cells on paths with positive slack to save energy. Bokhari et al. [6] have shown usefulness of Multi-Vt Optimization for NoC routers.

We use the multi-vt optimization technique to design routers for different VFs and integrate them at each mesh node. Our novel *Malleable NoC* is flexible NoC fabric that enables combining one router from each node to create a adaptable NoC at runtime. An example template for *Malleable NoC* architecture is shown in Figure 2. On the right, routers for 3 consecutive nodes are shown. As discussed earlier, we use MVR for per router VF selection. In the architecture shown in Figure 2, we have 2 different voltage rails $V1$ and $V2$, where $V1 > V2$. These voltages are associated with 2 frequencies f and $f/2$. At each node we can have maximum of 2 different routers that are optimized for 2 VF pairs (e.g, $node\ i$ and $node\ i+1$ in Figure 2). Similarly, there can be nodes which only contain router for maximum VF (e.g, $node\ i+2$ in Figure 2).

The single router at $node\ i+2$ in Figure 2 is connected to both of the voltage rails. Whereas, at $node\ i+1$, the router designed for [f,$V1$] is attached to voltage rail $V1$ and router designed for [$f/2$,$V2$] is connected to $V2$. If there is no router available for a target VF, the router designed for closest VF is used instead. Therefore, if a node needs to be operated at [$f/2$,$V2$] and there is no router designed for [$f/2$,$V2$] integrated at that node, the router designed for [f,$V1$] is used by connecting it to $V2$ rail. Multiplexers are used to connect one of the routers at a node to upstream routers. These multiplexers add minimum hardware and does not affect the critical path of the router.

3.1.3 Malleable NoC Management

Routers at each node are managed by a Local Manager (LM). The LM controls the power gating signal and also selects the clock for each node. The LM can be addressed by sending single flit packet over NoC. The LM also monitors if all the buffers in the routers are empty or not. All the LMs are managed by one Global Manager(GM) on the chip. This GM can be a part of existing NoC power controllers. At runtime, the LM and GM coordinate autonomously to implement the VF selection scheme.

GM runs the actual per router DVFS selection algorithm which will be explained in details later in this section. The DVFS function provides the frequency setting for each router in the mesh. First, GM stops all Network Interfaces(NIs) from injecting new traffic to ensure no data is lost when routers are switched on/off. Once the NoC is free of any traffic, GM sends information for the new frequency to each router. Depending on the architectural configuration of routers at each node and the new frequency, LM turns on new router, change the voltage and select the appropriate frequency. Once the new frequency setting for each routers has been set, GM sends control flit to each LM to enable normal operation.

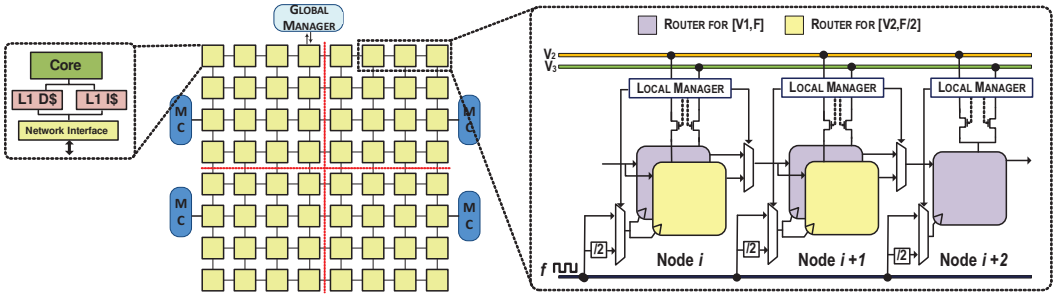


Figure 2: Malleable NoC

Each LM also propagate the current router frequency to neighboring LMs using single bit wire network. This information is used by the local LM to clock the input FIFOs accordingly.

3.2 Application Mapping and Profiling

As discussed earlier in Section 1, we exploit the application mapping to perform efficient per router VF selection. Our mapping is based on policies introduced by Das et al. [21]. We logicaly divide our multi-core chip into $memory\ islands$ based on the number of memory controllers. For example, in the architecture in Figure 2, there are four memory controllers and therefore we have 4 $memory\ islands$. Applications that are required to be mapped are sorted on the basis of L1 MPKI. These applications are then allocated to each memory Island in a round robin order. Within each $memory\ island$, applications with highest MPKI are mapped closest to memory controller.

We assume that an average L1 MPKI for each application is known *a priori*. However for the purpose of runtime per router VF selection, we regularly sample the L1 MPKI (using instruction and cache miss counters) after every control interval and send it through NoC to GM.

3.3 VF Selection Algorithm

Our VF selection algorithm is based on two key observations. The first observation is that applications have different L1 MPKIs and this value can change over time, hence providing an opportunity to switch to low VF for low MPKI program phase and high VF for high MPKI program phase. The second observation is that the execution time of the high MPKI application increases when they are mapped further away from memory controller as it takes longer for cache read misses to be served.

The VF selection algorithms is as follows. Applications are mapped as explained earlier and their MPKIs are monitored at runtime. The MPKI value for each application is multiplied by the distance of the core from the memory controller node. We call the resultant product $Senstivity\ Metric$. Then, the value of $Senstivity\ Metric$ is compared with already set threshold values. At start, for each algorithm run, routers are set to operate at low frequency. For each core, if the $Senstivity\ Metric$ is greater than the threshold, all the routers that lie on the request path from core to memory controller and the reply path from memory controller to core are selected to run at higher frequency. This results in a 2D frequency map which is then enforced by GM as explained earlier. Designer can select a suitable value for threshold for balancing performance and NoC power. The time complexity of the VF selection algorithm is $O(N^2)$, however, the NoC operates normally during the time algorithms in executed. Normal NoC operation is only suspended once the new frequency map is calculated.

4. EXPERIMENT AND RESULTS

4.1 Experiment Setup

NoC Synthesis For this paper, we use a 2 stage pipelined wormhole switched router with 64 bit data channel width. In the first stage flit travel on link and get written in the input buffer. The second stage consist of route computation, switch arbitration and switch transversal. The router architecture RTL is written in Verilog. We synthesized the RTL using Synopsys Design Compiler version *H-2013.03-SP4*. We used commercial TSMC 45nm libraries that are characterized for Low Vt(LVt), Normal Vt(NVt) and High Vt(HVt). We enable leakage and dynamic power optimization which automatically invoke multi-vt optimization. We target two VF level for the router design in this paper, i.e, [1GHz, 0.9V] and [500MHz, 0.81V].

MPSoC Details We target a 64 core general purpose MPSoC organized in 8×8 2D-mesh. The processors are an in-order Tensilica Xtensa core with 16KB private L1 instruction and data cache. Caches are 2-way associative with 16 byte line size, The core stalls when there is a outstanding cache load miss. The frequency for the cores is fixed at 1GHz. For the baseline system, we assume that there are 4 memory controllers placed at the boundaries as shown in Figure 2. DRAM access latency is assumed to be $40\ ns$.

Benchmark Application We use a diverse set of benchmark application from MediaBench and SPEC2006 packages to evaluate our proposed scheme. We created eight multiprogrammed workloads by choosing 4 applications and executing 16 copies of each application. Application are chosen to create workloads with varying MPKIs. For example workloads $W1$, $W2$ and $W4$ have a lower MPKI while rest of the workloads ($W2$, $W5-8$) have moderate to high MPKIs.

Simulator We use an in-house cycle accurate simulator for our study. The simulation is carried out in two step: First, the applications are executed on Xtensa instruction set simulator and information about cache load and store memory traces is collected along with timing information. In full system simulation, these memory traces are replayed in a closed loop simulator which accurately models NoC latency effect on application execution. Our simulator accurately models the low level hardware details for NoC VF switching, operation of LM and GM, and memory controller queuing delays.

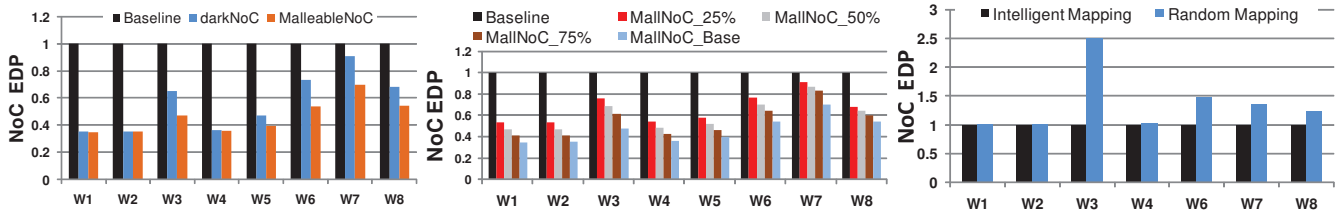
VF Switching Setup We explore designs with dual voltage coarse-grain VF selection scheme. Two possible VF levels for NoC are [1GHz, 0.9V] and [500MHz, 0.81V]. We assume that it takes 10ns to switch between two voltages and switch on a router. We used a control interval of $50K$ clock cycles for the VF selection algorithm based on analysis presented by Chen et al. [11]. Threshold value for VF selection algorithm is selected to be 150 based on on prior evaluation.

For power characterization of routers, we passed the netlist generated by synthesis tool to Synopsys PrimeTime. The netlist generated for [1GHz, 0.9V] router is evaluated at [1GHz, 0.9V] and [500MHz, 0.81V], while the netlist for [500MHz, 0.81V] router is evaluated at the design VF only. The energy values extracted from PrimeTime tool are fed in the simulator for calculating the NoC energy.

Malleable NoC Configurations We explored different configurations for *Malleable NoC*. In all configurations, the NoC contain routers designed for [1GHz, 0.9V]. In the baseline configuration, all the nodes contain router for both [1GHz, 0.9V] and [500MHz, 0.81V]. We then explore designers where the percentage of nodes with [500MHz, 0.81V] routers is 75%, 50% and 25%.

4.2 Results and Discussion

Efficacy of Malleable NoC Architecture Figure 3(a) reports the normalized NoC energy delay product(EDP) for *Malleable NoC*. We also compare our proposed architecture with darkNoC architecture [6]. All results are normalized to EDP of *Baseline* architecture, where no DVFS is used and NoC is run at maximum VF. Overall, *Malleable NoC* resulted in lowest NoC EDP for all the workloads. For workloads $W1$, $W2$ and $W4$, *Malleable NoC* and darkNoC



(a) Efficacy of Malleable NoC

(b) Effect of different Malleable NoC Configurations

(c) Effect of Mapping on Malleable NoCs

Figure 3: Experiment Results

performed equally well. These workloads have very low MPKIs and it is expected that the VF selection algorithm will operate all routers in the mesh at lower frequency. Therefore, per router VF selection provided by *Malleable NoC* performs similarly to single VF domain switching used in darkNoC.

On the other hand, for workloads $W3$, $W5$, $W6$, $W7$ and $W8$, *Malleable NoC* provided considerably lower EDP than darkNoC. For example, for workload $W7$, *Malleable NoC* resulted in relative EDP saving of 30%, while darkNoC only provided EDP savings of 9%. Similarly, for workload $W6$, *Malleable NoC* resulted in relative EDP saving of 46%, whereas darkNoC only provided EDP saving of 26.4%. Workloads $W3$, $W5$, $W6$, $W7$, $W8$ have very higher diversity in terms of MPKI. Therefore, we can conclude that *Malleable NoC* can provide superior NoC EDP savings for chips that are expected to run applications with mix of high and lower MPKIs.

Analysis of Malleable NoC Configurations In Figure 3(b) we report the normalized EDP of different configurations of *Malleable NoC*. The *MallNoC_Base* is the *Malleable NoC* configuration in which every node contain routers for [1GHz, 0.9V] and [500MHz, 0.81V]. As expected, *MallNoC_Base* result in lowest EDP. For $W7$ and $W8$, the use of *MallNoC_25%* result in 30% and 25% increase in EDP over *MallNoC_Base*, respectively. Whereas for both $W1$ and $W2$, the use of *MallNoC_25%* result in 52% increase in EDP over *MallNoC_Base*. From this result, we can deduce that workloads will highly diverse MPKIs (such as $W7$ and $W8$) can potentially use partial *Malleable NoC* configuration, thus saving dark silicon area and using it for other lower power components. The reason is that in workloads with high and low MPKI applications, it is expected that routers on the outer periphery of mesh will be operated at high VF level, and therefore such applications are least affected by the absence of low VF routers on the mesh outer periphery.

Importance of Mapping The graph in Figure 3(c) shows the importance of using intelligent application to core mapping scheme [21]. We randomly mapped the applications to cores and then used the *Malleable NoC* for the evaluation. For workloads $W3$, $W6$, $W7$, $W8$ the EDP with respect to intelligent mapping increased by 150%, 48%, 35% and 25%, respectively. This shows that it is important to use intelligent mapping with *Malleable NoC* to get the best results for workloads containing applications with heterogenous MPKI . On the other hand, for workloads with lower MPKIs, i.e., $W1$ $W2$ $W4$, the EDP is increased negligibly.

5. CONCLUSION

In this paper, we presented a low power NoC fabric called *Malleable NoC* which contain multiple VF optimized routers for each node. At runtime, depending on the workload, a router from each mesh node is combined to form a NoC which can adapt to heterogenous application requirements. We show that for a variety of multi program benchmarks executing on *Malleable NoC*, Energy Delay product (EDP) can be reduced by up to 46% for widely differing workloads.

6. REFERENCES

[1] J. Howard, S. Dighe, S. Vangal, G. Ruhl, N. Borkar, S. Jain, V. Erraguntla, M. Konow, M. Riepen, M. Gries, G. Droege, T. Lund-Larsen, S. Steibl, S. Borkar, V. De, and R. Van Der Wijngaart, "A 48-core ia-32 processor in 45 nm cmos using on-die message-passing and dvfs for performance and power scaling," *Solid-State Circuits, IEEE Journal of*, vol. 46, no. 1, pp. 173–183, 2011.

[2] J. S. Kim, M. B. Taylor, J. Miller, and D. Wentzlaff, "Energy characterization of a tiled architecture processor with on-chip networks," in *Proceedings of the 2003 international symposium on Low power electronics and design*, ISLPED '03, (New York, NY, USA), pp. 424–427, ACM, 2003.

[3] S. Borkar and A. A. Chien, "The future of microprocessors," *Commun. ACM*, vol. 54, pp. 67–77, May 2011.

[4] P. Bogdan, R. Marculescu, S. Jain, and R. Gavila, "An optimal control approach to power management for multi-voltage and frequency islands multiprocessor platforms under highly variable workloads," in *Networks on Chip (NoCS), 2012 Sixth IEEE/ACM International Symposium on*, pp. 35–42, 2012.

[5] A. Mishra, R. Das, S. Eachempati, R. Iyer, N. Vijaykrishnan, and C. Das, "A case for dynamic frequency tuning in on-chip networks," in *Microarchitecture, 2009. MICRO-42. 42nd Annual IEEE/ACM International Symposium on*, pp. 292–303, 2009.

[6] H. Bokhari, H. Javaid, M. Shafique, J. Henkel, and S. Parameswaran, "darknoc: Designing energy-efficient network-on-chip with multi-vt cells for dark silicon," in *Proceedings of the The 51st Annual Design Automation Conference on Design Automation Conference*, DAC '14, (New York, NY, USA), pp. 161:1–161:6, ACM, 2014.

[7] A. K. Mishra, O. Mutlu, and C. R. Das, "A heterogeneous multiple network-on-chip design: an application-aware approach," in *Proceedings of the 50th Annual Design Automation Conference*, DAC '13, (New York, NY, USA), pp. 36:1–36:10, ACM, 2013.

[8] A. Bakhoda, J. Kim, and T. Aamodt, "Throughput-effective on-chip networks for manycore accelerators," in *Microarchitecture (MICRO), 2010 43rd Annual IEEE/ACM International Symposium on*, pp. 421–432, Dec 2010.

[9] A. K. Mishra, N. Vijaykrishnan, and C. R. Das, "A case for heterogeneous on-chip interconnects for cmps," in *Proceedings of the 38th annual international symposium on Computer architecture*, ISCA '11, (New York, NY, USA), pp. 389–400, ACM, 2011.

[10] L. Shang, L.-S. Peh, and N. Jha, "Dynamic voltage scaling with links for power optimization of interconnection networks," in *High-Performance Computer Architecture, 2003. HPCA-9 2003. Proceedings. The Ninth International Symposium on*, pp. 91–102, 2003.

[11] X. Chen, Z. Xu, H. Kim, P. V. Gratz, J. Hu, M. Kishinevsky, U. Ogras, and R. Ayoub, "Dynamic voltage and frequency scaling for shared resources in multicore processor designs," in *Proceedings of the 50th Annual Design Automation Conference*, DAC '13, (New York, NY, USA), pp. 114:1–114:7, ACM, 2013.

[12] G. Liang and A. Jantsch, "Adaptive power management for the on-chip communication network," in *Digital System Design: Architectures, Methods and Tools, 2006. DSD 2006. 9th EUROMICRO Conference on*, pp. 649–656, 2006.

[13] T. Miller, X. Pan, R. Thomas, N. Sedaghati, and R. Teodorescu, "Booster: Reactive core acceleration for mitigating the effects of process variation and application imbalance in low-voltage chips," in *High Performance Computer Architecture (HPCA), 2012 IEEE 18th International Symposium on*, pp. 1–12, Feb 2012.

[14] H. Matsutani, M. Koibuchi, D. Ikebuchi, K. Usami, H. Nakamura, and H. Amano, "Performance, area, and power evaluations of ultrafine-grained run-time power-gating routers for cmps," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 30, no. 4, pp. 520–533, 2011.

[15] G. Kim, J. Kim, and S. Yoo, "Flexibuffer: Reducing leakage power in on-chip network routers," in *Design Automation Conference (DAC), 2011 48th ACM/EDAC/IEEE*, pp. 936–941, 2011.

[16] A. Samih, R. Wang, A. Krishna, C. Maciocco, C. Tai, and Y. Solihin, "Energy-efficient interconnect via router parking," in *High Performance Computer Architecture (HPCA2013), 2013 IEEE 19th International Symposium on*, pp. 508–519, 2013.

[17] R. Das, S. Narayanasamy, S. K. Satpathy, and R. G. Dreslinski, "Catnap: energy proportional multiple network-on-chip," in *Proceedings of the 40th Annual International Symposium on Computer Architecture*, ISCA '13, (New York, NY, USA), pp. 320–331, ACM, 2013.

[18] W. Kim, M. Gupta, G.-Y. Wei, and D. Brooks, "System level analysis of fast, per-core dvfs using on-chip switching regulators," in *High Performance Computer Architecture, 2008. HPCA 2008. IEEE 14th International Symposium on*, pp. 123 –134, 2008.

[19] P. Teehan, M. Greenstreet, and G. Lemieux, "A survey and taxonomy of gals design styles," *IEEE Des. Test*, vol. 24, pp. 418–428, Sept. 2007.

[20] 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 2014 International Conference on Hardware/Software Codesign and System Synthesis*, CODES '14, (New York, NY, USA), pp. 13:1–13:10, ACM, 2014.

[21] R. Das, R. Ausavarungnirun, O. Mutlu, A. Kumar, and M. Azimi, "Application-to-core mapping policies to reduce memory system interference in multi-core systems," in *High Performance Computer Architecture (HPCA2013), 2013 IEEE 19th International Symposium on*, pp. 107–118, Feb 2013.