A Low-Power Network-on-Chip Architecture for Tile-based Chip Multi-Processors

Anastasios Psarras
ECE, Democritus University of Thrace, Xanthi, Greece

Junghee Lee
ECE, University of Texas at San Antonio, USA

Pavlos Mattheakis
Mentor Graphics
Grenoble, France

Chrysostomos Nicopoulos
ECE, University of Cyprus, Nicosia, Cyprus

Giorgos Dimitrakopoulos
ECE, Democritus University of Thrace, Xanthi, Greece

ABSTRACT

Technology scaling of tiled-based CMPs reduces the physical size of each tile and increases the number of tiles per die. This trend directly impacts the on-chip interconnect; even though the tile population increases, the inter-tile link distances scale down proportionally to the tile dimensions. The decreasing inter-tile wire lengths can be exploited to enable swift link traversal between neighboring tiles, after appropriate wire engineering. Building on this premise, we propose a technique to rapidly transfer flits between adjacent routers in *half a clock cycle*, by utilizing both edges of the clock during the sending and receiving operations. Half-cycle link traversal enables, for the first time, substantial reductions in (a) link power, *irrespective of the data switching profile*, and (b) buffer power (through buffer-size reduction), without incurring any latency/throughput loss. In fact, the proposed architecture also yields some latency improvements over a baseline NoC. Detailed hardware analysis using placed-and-routed designs, and cycle-accurate full-system simulations corroborate the significant power and latency improvements.

1. INTRODUCTION

Technology scaling and power constraints have led to a fundamental paradigm shift in digital system design: the transition to the multi-core paradigm. Consequently, the role of the on-chip communication fabric has become pivotal to the system's operation. Networks-on-Chip (NoC) have been established as the dominant communication backbone in multi-core environments, primarily due to their innate scalability attributes.

There are two primary variants of multi-core systems: (1) Multi-Processor Systems-on-Chip (MPSoC), and (2) Chip Multi-Processors (CMP). These two multi-core incarnations have distinct attributes, which directly impact their NoC architecture. This work focuses on CMPs and aims to capitalize on one of their key (and differentiating) characteristic: the regularity in their physical layout. Unlike MP-SoCs, which typically integrate a variety of IP cores of disparate physical sizes, homogeneous CMPs contain a number of identical CPU cores. This homogeneity in physical size has given rise to tile-based CMPs, whereby the system comprises a number of identically-sized logic blocks, called *tiles*. Each tile contains one, or more, CPU cores, private L1 instruction and data caches, a slice of L2 cache (more cache levels are possible), and a gateway to a NoC connecting the tiles [26, 21]. Tiling facilitates easy integration of multiple cores on the same die, and it enhances the design's scalability to increasing numbers of cores. The prevalent NoC topology in CMP designs is the 2D mesh, which is amenable to tile-based layouts [4].

Since the overall size of a CMP die tends to remain constant across different processor generations (due to yield and cost issues), any increase in the number of on-chip tiles is inevitably accompanied by a corresponding decrease in the size of each tile. This attribute has a profound impact on the system's NoC. The effective network throughput per core decreases (due to elevated cross traffic), while the average source-to-destination hop-count increases (due to increasing network diameter). Thus, both latency and throughput suffer as the NoC mesh size increases to accommodate the extra CMP tiles. At the same time, the escalating numbers of on-chip tiles inevitably affect the NoC's *power* budget. The router buffers and the inter-router links constitute the two major NoC power consumers [4, 11]. To sustain scalability into the many-core realm, it is imperative to curtail the power expended in NoC buffering and link traversal, *without* incurring any latency/throughput penalties.

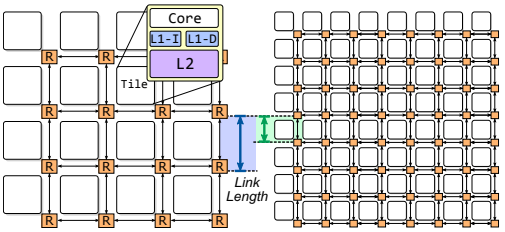


Figure 1: Under equal die size, as the population of tiles increases, the inter-tile link length decreases. This phenomenon is illustrated here by comparing a 16-core CMP (left) to a 64-core CMP (right).

On the other hand, even though the number of tiles increases, the inter-tile link distances scale *down* proportionally to the decreasing tile dimensions. Figure 1 illustrates

GLSVLSI '16, May 18-20, 2016, Boston, MA, USA

 ISBN 978-1-4503-4274-2/16/05…$15.00
DOI: http://dx.doi.org/10.1145/2902961.2903010

this effect by comparing the link lengths of two different processor generations, comprising 16 and 64 cores, respectively. For instance, at the 32 nm technology node, the longest side of a typical CMP tile (i.e., CPU core + 32 kB L1 instruction and data caches + 512 kB L2 cache slice) is 3.27 mm, as demonstrated in [23], while this distance is reduced to only 0.9 mm in 15 nm technology [1]. Thus, the inter-tile wire connections of future CMPs will hardly exceed 1 mm.

In this paper, we aim to harness the swift wire traversal made possible by wire engineering to enable low-power NoCs for future CMPs, which are characterized by: (a) scaled inter-tile wire lengths and increasing network sizes (i.e., the tile size decreases due to technology scaling, while the on-chip tile population increases), and (b) constant, or modestly increasing, clock frequencies. Rather than using the wire speed to cover longer distances in a given cycle [23, 1, 13, 14], we propose exploiting said swiftness to *rapidly transfer flits* between adjacent routers in *half a clock cycle*, by *utilizing both edges of the clock* during the sending and receiving of flits. By leveraging half-cycle link traversal we achieve a double-faceted objective:

- **Link-power reduction**, by substantially lowering the effective wire capacitance. The latter is reduced irrespective of the data profile, and by altering only the physical layout of the link wires. The attained wire-capacitance reduction renders, in turn, the half-cycle constraint easier to satisfy.
- **Buffer reduction**, by significantly reducing the credit Round-Trip Time (RTT), which lowers the full-throughput buffering requirements to a size smaller than the current minimum. The decrease in RTT is facilitated by a novel switch allocator micro-architecture.

Both aforementioned achievements translate into **tangible and extensive NoC power savings**, without adversely impacting the latency/throughput performance. On the contrary, the new architecture also yields some latency improvements over a baseline NoC. The proposed design is investigated in detail and quantitatively explored to highlight its potential. Extensive cycle-accurate simulations using both synthetic traffic patterns, and execution-driven full-system simulations with real application workloads, validate the efficiency of the new low-power NoC architecture. Furthermore, detailed hardware analysis using placed-and-routed designs synthesized in a 45 nm standard-cell library corroborates the achieved significant power improvements.

2. BACKGROUND & RELATED WORK

There is a rich body of work related to low-power NoC design, which can be broadly categorized into two main thrusts: (a) *policy-based*, and (b) *structural* approaches. The former category includes such techniques as Dynamic Voltage and Frequency Scaling (DVFS) and power gating, which reap power benefits based on prevailing conditions. Specifically, DVFS techniques have been used to lower the power consumption of the links [24], and/or other components [21, 20, 8]. Power gating has also been employed – at various implementation granularities – to contain the NoC's power envelope [12, 7]. Being policy-driven, all these approaches are complementary to the design proposed here.

The proposed design falls into the second category of *structural* low-power approaches that involve modifications to the NoC topology and architecture, as in [25].As such, the extracted benefits are not traffic- or scenario-dependent; they are inherent to the design and are always present.

Recently, researchers have leveraged ultra-fast wire traversal techniques, enabled by wire engineering, to expedite the traversal of flits across the NoC of tile-based CMPs. The resulting solutions led to either high-radix networks [23, 1, 13] with long connecting links, or to networks that allow flits to traverse multiple network hops of shorter wires in a single clock cycle (through router bypassing) [14]. Both techniques reduce the average hop count, which results in a decrease in the number of buffer read/write operations, and, hence, a decrease in power consumption. Nevertheless, while some power is saved, there is *no impact on buffer sizing*; the buffer size requirements are unaffected, since the credit-notification loop (aka round-trip time) does not change. On the other hand, buffer-less alternatives [9] effectively reduce the total number of buffers, but with significant throughput losses.

The effectiveness of the aforementioned solutions relies on their fundamental property of transferring flits over longer distances in a single clock cycle. This property becomes increasingly difficult to sustain for constant-length wires, since wire delay increases with technology scaling. However, appropriate semi-custom wire engineering, which involves (a) routing in upper metal layers to reduce resistance, (b) increased wire spacing to reduce the wire coupling capacitance, and (c) appropriate repeater placement, can partially alleviate the problem [17]. Several recent NoC designs [1, 14, 17, 15] have achieved single-cycle traversal of medium-to-long links by assuming repeated wire delays of 70–200 ps/mm. Similar wire speeds have been reported by Intel and IBM [21, 10]. Hence, achieving fast wire traversal speeds in *scaled-down* links (i.e., wire lengths that scale *down* as a result of decreasing tile dimensions) is more easily attainable.

The proposed architecture in this paper harnesses: (a) the swift link traversal speed of 70–200 ps/mm facilitated by repeaters and wire spacing in upper metal layers [1, 14, 17, 15], and (b) a fundamental reduction in link-wire capacitance provided by the new design, to reap **benefits in *both* power consumption and network latency**. By deliberately creating an *artificial asymmetry* between the *intra-* and *inter*-router delays of tile-based CMPs, we achieve half-cycle link traversal. Currently, the fastest state-of-the-art NoC routers for 2D meshes exhibit intra-router delays ranging from 600 ps to 1100 ps [21, 23, 14] (measured at voltages of around 0.8 V), i.e., more than 2× longer than the typical inter-tile link delay for scaled tile dimensions. By leveraging this attribute, the proposed NoC architecture can markedly **reduce both the link power and the buffer size**, in addition to improving performance. The designs of [6, 19] also employ both edges of the clock for link traversal. However, their aim is the simplification of clock distribution, without offering any power benefits.

The asymmetry between intra- and inter-router delays is also evident in asynchronous NoCs [3]. However, asynchronous circuits tend to be very complex to design and validate. Thus, the design proposed here infuses some of the benefits of asynchronous NoCs into a synchronous setup.

3. FACILITATING HALF-CYCLE LINKS

In the proposed architecture, adjacent routers (placed at the corners of neighboring tiles, as shown in Figure 2) operate under opposite clock edges. The 2D mesh topology makes the connectivity look like a checkerboard pattern. Flits have a full cycle to execute all router operations, but only half a cycle to get from one router to the next.

Figure 3 illustrates a single-flit packet traversing three network nodes. In the top timing diagram of Figure 3, the network – which employs the proposed architecture – consists of white and gray nodes, which operate on the positive and negative clock edges, respectively. In cycle 0, the flit is injected and written in the input buffer of router A, on the

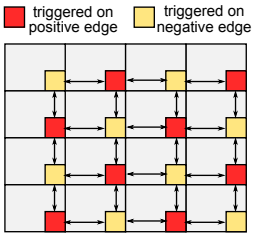


Figure 2: Mesh topology for a tiled CMP employing alternate clock edges for adjacent routers.

positive clock edge. The flit spends a whole cycle executing router operations and is eventually written on the next positive clock edge in the router's output register. Half a cycle later, the flit has crossed the whole link (Link Traversal, LT) and is written in the input buffer of router B, captured on the negative clock edge. A whole cycle is again spent inside the router, until the flit appears at the output link after the negative edge of cycle 2. Router C captures the flit on the positive edge of cycle 3, allowing the flit to eventually appear in the output register, and leave in cycle 4.

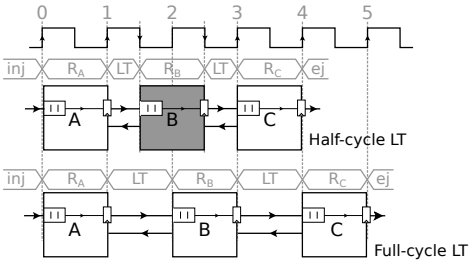


Figure 3: Networks-on-Chip with half-cycle links are built by restricting adjacent routers to operating on opposite clock edges. The small wire delay (for reasonably scaled inter-tile wire lengths) allows wire traversal to be completed in half a cycle.

The same scenario without half-cycle links is depicted in the bottom timing diagram of Figure 3, in which only positive-edge routers are used. The flit spends a full cycle for both link and router traversals, eventually being ejected one cycle later than in the top diagram. Note that, in both cases, the links have the same length; scaling is merely used for illustrative purposes (to depict the timing difference).

Half-cycle links decrease the NoC's zero-load latency, using the same routers, and while working under the same clock frequency, relative to a baseline NoC with full-cycle links. In baseline NoCs with single-cycle routers (i.e., one cycle is spent in the router and one cycle on the link), zero-load latency in absolute time (number of cycles divided by the clock frequency) equals $T_B = (2H + L - 1)/f_{CLK}$, assuming an L-flit packet that has to traverse H hops to its destination. When the same packet traverses a NoC with half-cycle links, it saves half a cycle in each hop, resulting in a latency of $T_{RL} = (1.5H + L - 1)/f_{CLK}$. This translates to 8% to 25% latency savings, considering a packet length range of 1 to 5 flits, which travel in the NoC from 1 to 8 hops away.

4. LINK CAPACITANCE REDUCTION

Half-cycle links allow for significant effective wire capacitance reduction without any performance overhead. Lower wire capacitance leads to lower power dissipation in the links of the NoC, while it enables faster wire traversal.

The total interconnect capacitance is the sum of ground and coupling capacitances, C_{GND} and C_C. The effective C_C depends on the switching behavior of adjacent wires, which is characterized by the Miller Coupling Factor (MCF) in Equation (1):

$$C_{WIRE} = C_{GND} + 2 \times MCF \times C_C \quad (1)$$

The MCF parameter is 0 when all adjacent wires switch in the same direction, and the total wire capacitance is only C_{GND}. On the contrary, MCF is 2 when adjacent wires switch in the opposite direction.

It is possible to reduce coupling capacitance by increasing the wire spacing, or by introducing shielding (interleaving constant VDD/GND wires and data wires), but this comes at the cost of an area penalty. Wire spacing effectively reduces the value of C_C, while shielding decreases MCF. A key challenge in interconnect design is to reduce either C_C, or the worst-case MCF, while maintaining the same physical footprint of the interconnect, thereby reducing the effective wire capacitance, the delay, and energy consumption.

In the new NoC design proposed in this paper, wire capacitance is significantly reduced by exploiting the out-of-phase operation of the two unidirectional links that connect two adjacent routers. Assume, for example, the routers A and B shown in Figure 4. The A→B link is used to transfer data from router A to router B. Data is launched on the positive edge of the clock and captured on the negative edge of the clock by the input buffers of router B. On the contrary, data from B to A is transferred on the other half-period, starting from the negative edge and ending on the next positive edge. In this case, when one direction is activated in one half of the clock cycle, the other one remains idle, waiting for the other half to send its data (the timing diagram of Figure 4 depicts this behavior).

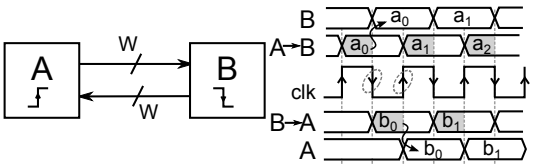


Figure 4: Since flits are launched and captured on alternate clock edges, only one of the two inter-router links is active in each half-cycle.

Based on this property of link activation in different clock phases, we can employ wire interleaving in the physical layout. The wires of the two uni-directional buses connecting adjacent routers are interleaved, as shown in Figure 5. This pattern eliminates the undesired MCF=2 of the coupling capacitances of neighboring wires.

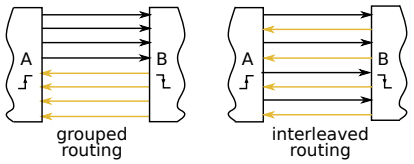


Figure 5: Wire-interleaving of half-cycle links limits the worst-case MCF to 1 (rather than 2), independent of the data-switching activity profile.

Wire-interleaving prohibits adjacent wires from switching in opposite directions, thus **achieving a worst-case MCF of 1 (rather than 2)**, *independent of the switching activity* profile of the transmitted data. Bit i of the link from A to B is routed between bits i and $i-1$ of the link from B to A. Since the two links are **never active in the same half-cycle**, when bit i of the A→B link makes a new transition, it is guaranteed that its adjacent bits – that belong to the B→A link – are "quiet," since they have completed their transitions in the previous half-cycle. The adjacent wires can be considered as active shields for the bit that changes its value. The same situation occurs in the next half-cycle, which drives the wires of the opposite direction. Overall, in any half-cycle, every bit will make a transition with the *guarantee that its neighbors remain constant*. Due to this property, MCF cannot exceed 1, thereby offering a significant reduction in the effective wire capacitance. This guarantee holds for all links of the NoC in either direction, which provides significant overall energy savings. As proven in [22], the energy savings in the links when limiting the largest value of MCF to 1 (as opposed to 2) is equal to

$$E_{saving} = 1 - \frac{C_{WIRE(MCF=1)}V_{DD}^2}{C_{WIRE(MCF=2)}V_{DD}^2} = 1 - \frac{C_{GND} + 2C_C}{C_{GND} + 4C_C}$$

The energy savings range between 25% and 40%, depending on the value of C_C, relative to C_{GND}, which depends on the metal layer and the wire geometry and spacing [22]. Moreover, note that the lower wire capacitance makes link traversal even faster, thus making the half-cycle requirement easier to achieve for the scaled inter-tile links.

5. ROUTER BUFFER REDUCTION

Half-cycle links change the notification loop of credit-based flow control, since data is sent in the forward direction on one edge of the clock, while credit updates are returned in the reverse direction on the opposite edge of the clock. This effectively *reduces the Round-Trip Time (RTT) by one cycle*. The RTT is the number of elapsed cycles between the time the receiver sends back a credit, and the time a flit using that particular credit arrives at the receiver.

The credit notification loop starts from and ends at flow-controlled storage elements, such as the FIFO buffers. Any pipeline register (without embedded flow control), e.g., the ones used at the outputs of the routers, increase the RTT. The minimum buffering required on the receiver side to achieve full throughput operation is determined by the RTT. A smaller RTT value translates into lower input-buffering requirements to achieve full-throughput operation.

To enjoy the decreased RTT, the credit updates that arrive at the middle of the cycle must be reused immediately. In this way, if credit updates are not handled appropriately, combinational paths launching in the middle of the cycle are created. These could potentially degrade the router's speed. To avoid this negative effect and still enjoy the reduced RTT and its associated buffer-saving properties, we redesign the Switch Allocation (SA) stage of the NoC routers.

In fast VC-based router architectures, such as [16], the SA stage receives only *qualified* requests. A request is qualified to participate in SA, as long as it refers to an output VC that has available credits. If credit-checking is not performed before-hand, then a grant may be given to an input VC that cannot actually use it, thus leaving the selected output port idle. In fast implementations [16], VC allocation occurs through SA: once a head flit is assigned to an output port, it also receives an available output VC for that output port.

In the proposed architecture, we employ two SA units, the

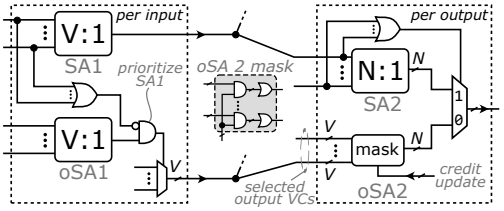


Figure 6: The proposed architecture employs two switch allocation units (one primary and a simplified oblivious one), in order to allow any credit updates arriving at the middle of the cycle to be reused immediately. Parameters N and V represent the number of router input ports and the number of VCs per port, respectively.

Primary SA (*pSA*) and the Oblivious SA (*oSA*) units, placed in parallel, as shown in Figure 6. Being the primary unit, *pSA* performs the normal SA operation, including credit-checking. The *oSA* unit performs, in parallel, the same task (albeit in a simplified manner), but it only accepts the requests of the inputs (or input VCs) that refer to outputs (or output VCs) without any credit available.

When an output VC has available credits and an input VC is requesting it, then this resource will definitely be utilized. In this case, handling the credits returning in the middle of the cycle is not critical. The half-cycle credit updates become critical when the requested resource appears to be unavailable on one edge of the clock, but becomes available on the next opposite edge of the clock (recall that neighboring routers operate on alternate clock edges). For this case, the *oSA* unit allows one request from each input port to qualify for an output request, as long as it refers to an *unavailable* output VC. The request can only eventually be granted if the specific output VC becomes available in the middle of the cycle, through a credit update. In any case, the *pSA* unit's decisions are always prioritized over *oSA*'s.

The operation of *oSA* is split in two stages, as shown in Figure 6. In the first stage (*oSA1*), each input port independently selects one input VC headed to an unavailable output VC. The *oSA1* winners reach their destined output port, where at most one winner is determined in *oSA2*. At this point, half the critical path has been traversed, and, thus, the credit-update signals have arrived. Since only one credit update for a single output VC may arrive for each output port, only the input request that refers to that output VC can be granted. All other requests refer to output VCs that are still unavailable and are discarded through a masking process, without the need for any arbitration. Thus, the *oSA2* stage is a simple masking process, unlike the more complicated second-stage arbitration process typically found in SA units. Note that the only extra logic added by *oSA* can be seen in the lower half of Figure 6; the upper half illustrates the organization of a baseline SA.

In this way, the incoming credit-update signal is used with no delay cost, and this enables immediate credit consumption, which effectively **lowers the credit notification loop by 1 cycle**. One half-cycle is saved in the forward data direction, and one half-cycle is saved in the backward (credit-update) direction. Reducing the RTT by one cycle translates to a saving of one buffer slot per VC. In the case of skewed traffic, each VC (independently of the rest) should have as many buffer slots as the link's RTT, in order to ensure full-throughput of data transfers.

6. EXPERIMENTAL EVALUATION

In this section, we evaluate the performance of the proposed NoC architecture and compare it – in terms of **hardware complexity** and **network performance** – with baseline NoC architectures that assume full-cycle link traversal.

As previously explained, the two fundamental contributions of the new design are: (1) the *link* power reduction, and (2) the *buffer size* reduction (thereby leading to further overall power reduction). We begin our evaluation by assessing the improvements in power consumption obtained when using the proposed architecture.

Figure 7 depicts the normalized power of an 8 × 8 (2D mesh) baseline NoC with full-cycle links ("Base"), as compared to the proposed design with half-cycle links ("Proposed"). Both designs operate at 1 GHz, support 2, 4, and 6 VCs per port, and the NoC routers (with 5 input/output ports, as needed by the 2D mesh topology) are connected with 2 mm 64-bit links (following the floor-planning sizes of [23]). In each configuration, VC buffers include 3 slots/VC for the baseline case and 2 slots/VC for the proposed architecture, which are enough to cover the RTT in each case. All NoC designs under investigation were implemented in VHDL, synthesized, and placed-and-routed using a 45 nm 0.8 V standard-cell library.

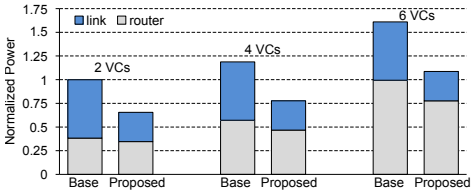


Figure 7: The normalized power consumption of the proposed architecture, as compared to a baseline NoC with full-cycle links. Power savings are due to both link- and buffer-power reductions.

From the obtained results, measured under equal NoC utilization (throughput), we observe that the savings in both the *link* and *buffer* power consumption (due to reduced wire capacitance and smaller buffers, respectively) offered by the new design translate to significant *overall* NoC power savings of 34%, on average. It is evident that the contributions of both the link optimizations and the buffer reduction are significant for all examined VC configurations. Specifically, link power reduction accounts for 75%, on average, of the total power savings, with the remaining 25% being attributed to the buffer power reduction.

Besides power reduction, the proposed router design also achieves small area savings, mostly due to the buffer reduction, which outweighs the small area added by the second SA unit. Recall that the latter is added to handle the credit updates arriving at the middle of the cycle, without any delay penalty.

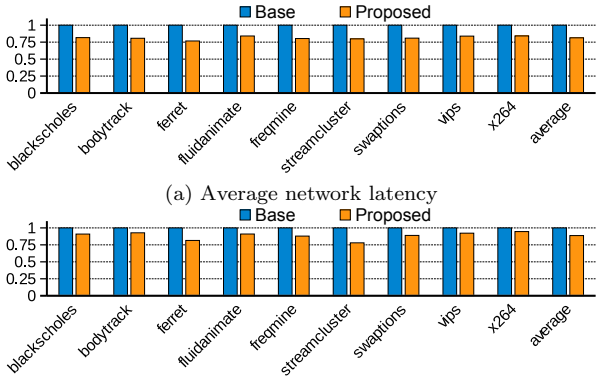
Note that the results obtained here (i.e., at 45 nm and assuming a 2 mm inter-tile distance) can also scale to smaller technologies, assuming (a) a diminishing tile area, and (b) slowly increasing clock frequencies. As previously mentioned, these two assumptions are already part of the prevailing tendency in current CMP design practices.

Having established the substantial power reduction enabled by the proposed design, it is now imperative to assess its impact on overall system *performance*. Toward this end, we simulate a 64-core tiled CMP system running real application workloads on a commodity operating system.

Table 1: Full-system simulation parameters.

Processor	64 in-order x86 cores in a tiled CMP
OS	Linux Fedora
L1 caches	Private, separate 32 KB I & D, 4-way set associative, 2-cycle latency, 64 B cache-line
L2 cache	Shared NUCA LLC, 4-way set associative, 16 MB total (64 cores×256 KB slice/core), 10-cycle latency, 64 B cache-line
Coherence	MOESI directory-based protocol
Main memory	4 GB, 300-cycle latency
Network	8×8 2D Mesh, 1-cycle routers +1 or +0.5 cycle link delay, XY routing, 128-bit inter-router links (flit width)

The simulation framework employs the Simics simulator, extended with GEMS [18] and the GARNET [2] cycle-accurate NoC simulator. Table 1 shows the full-system simulation parameters.



(a) Average network latency

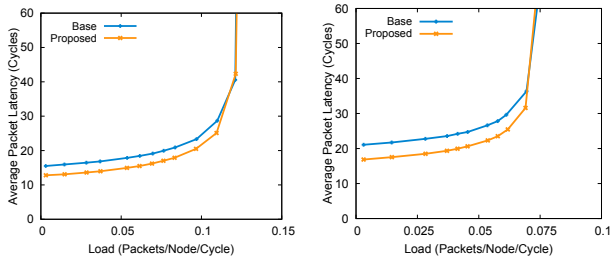
(b) Total execution time

Figure 8: Full-system, execution-driven simulation results: (a) average network latency, and (b) total execution time of various PARSEC applications. The results are normalized to the baseline NoC.

Figure 8 shows the *average network latency* and the *total execution time* of various PARSEC multi-threaded benchmark applications [5], normalized to a baseline NoC with full-cycle links. The proposed architecture offers significant average network latency savings, which range from 16% to 23% (19%, on average). This latency reduction translates to a direct reduction in the total execution time, ranging from 5% to 22% (11%, on average). Most importantly, such savings are achieved while still enjoying significant power reductions.

Despite the authenticity provided by execution-driven, full-system simulations, the flexibility to stress the NoC is somewhat limited, due to the fixed characteristics of the running applications. Hence, we also employ *synthetic* traffic patterns in our evaluation, by operating GARNET in a "network-only" mode, assuming the same NoC configurations. The performance evaluation involves Uniform Random (UR) and Bit-Complement (BC) traffic patterns. Packet lengths follow a bimodal distribution, with half the packets being 1-flit long, and the other half being 5-flit long.

Figure 9 depicts the average network latency versus input load of a baseline NoC with full-cycle links and a NoC using the proposed architecture with half-cycle links. The latter achieves, on average, 18% and 20% lower latency under UR and BC traffic, respectively, as compared to the baseline



(a) Uniform Random (UR) (c) Bit-Complement (BC)

Figure 9: Latency vs. load curves under UR and BC traffic for a baseline NoC (with full-cycle links) and the proposed NoC (with half-cycle links).

NoC. In terms of throughput, both designs achieve equal performance. The positive effect of half-cycle link traversal is more pronounced under traffic patterns with increased average zero-load latency (i.e., larger average hop count).

Hence, the substantial power savings reaped by the proposed new design come with no negative impact on network performance. On the contrary, our evaluation indicates substantial performance *gains*, in terms of network latency and real-world application performance.

7. CONCLUSIONS

Tile-based homogeneous CMPs are characterized by a regularity in their physical layout. As technology scales, the number of on-chip tiles increases, while the size of each tile decreases. Consequently, the inter-tile distances scale *down* proportionally to the decreasing tile dimensions. In this paper, we exploit decreasing inter-tile wire lengths to facilitate ultra-fast NoC link traversal in half a clock cycle. This feat is achieved using appropriate wire engineering techniques, and by utilizing both edges of the clock during the sending/receiving of flits. The proposed NoC design leverages half-cycle link traversal to reap substantial power savings. The savings are derived by commensurate reductions in (a) link power (irrespective of the data-switching patterns), and (b) buffer power (by reducing the buffer size). Most importantly, the power benefits do not incur any performance penalty; on the contrary, the latency is actually improved. Extensive hardware implementation analysis using placed-and-routed designs validates the efficacy of the proposed architecture, which yields significant power savings of 34%, on average. Additionally, cycle-accurate full-system simulations using real benchmark applications also demonstrate significant performance improvements, i.e., average reductions in network latency and total execution time of 19% and 11%, respectively.

8. REFERENCES

[1] N. Abeyratne and et al. Scaling toward kilo-core processors with asymmetric high-radix topologies. In *HPCA*, 2013.

[2] N. Agarwal and et al. GARNET: A detailed on-chip network model inside a full-system simulator. In *ISPASS*, 2009.

[3] A.Ghiribaldi, D.Bertozzi, and S.Nowick. A transition-signaling bundled data NoC switch architecture for cost-effective GALS multicore systems. In *Proc. of DATE*, pages 332–337, 2013.

[4] J. Balfour and W. J. Dally. Design tradeoffs for tiled CMP on-chip networks. In *ICS*, 2006.

[5] C. Bienia and et al. The parsec benchmark suite: Characterization and architectural implications. In *PACT*, 2008.

[6] T. Bjerregaard and et al. A scalable, timing-safe, noc architecture with an integrated clock distribution method. In *DATE*, 2007.

[7] L. Chen and et al. Power punch: Towards non-blocking power-gating of noc routers. In *Proc. of HPCA*, pages 378–389, 2015.

[8] X. Chen and et al. In-network monitoring and control policy for dvfs of cmp networks-on-chip and last level caches. *ACM TODAES*, (4):47, 2013.

[9] B. Daya, L.-S. Peh, and A. Chandrakasan. Towards high-performance bufferless nocs with scepter. *IEEE Computer Architecture Letters*, 2015.

[10] A. Golander and et al. A cost-efficient L1–L2 multicore interconnect: Performance, power, and area considerations. *IEEE Trans. on Circuits and Systems-I*, (3):529–538, 2011.

[11] S. M. Hassan and S. Yalamanchili. Centralized buffer router: A low latency, low power router for high radix nocs. In *Proc. of NoCS*, 2013.

[12] H.Matsutani and et al. Ultra fine-grained run-time power gating of on-chip routers for cmps. In *Proc. of NoCS*, pages 61–68, 2010.

[13] J. Kim and et al. Flattened butterfly topology for on-chip networks. In *MICRO*, 2007.

[14] T. Krishna, C.-H. O. Chen, W. C. Kwon, and L.-S. Peh. Smart: Single-cycle multi-hop traversals over a shared network-on-chip. *IEEE Micro*, May/June 2014.

[15] P. Lotfi-Kamran and et al. NOC-Out: Microarchitecting a scale-out processor. In *MICRO*, 2012.

[16] Y. Lu and et al. Design of interlock-free combined allocators for networks-on-chip. In *IEEE SOC Conf.*, pages 358–363, 2012.

[17] R. Manevich and et al. Designing single-cycle long links in hierarchical nocs. *Microprocessors and Microsystems*, (8):814–825, 2014.

[18] M. M. K. Martin and et al. Multifacet's general execution-driven multiprocessor simulator (gems) toolset. *SIGARCH Comput. Archit. News*, 33(4):92–99, Nov. 2005.

[19] I. Miro-Panades and et al. Physical implementation of the dspin network-on-chip in the faust architecture. In *Proc of. NoCS*, 2008.

[20] A. Mishra and et al. A case for dynamic frequency tuning in on-chip networks. In *Proc. of MICRO*, pages 292–303, 2009.

[21] P. Salihundam et al. A 2Tb/s 6x4 Mesh Network with DVFS and 2.3Tb/s/W router in 45nm CMOS. In *VLSI Circuits*, 2010.

[22] J. Seo and et al. A robust edge encoding technique for energy-efficient multi-cycle interconnect. *IEEE Trans. VLSI Syst.*, pages 264–273, 2011.

[23] K. Sewell and et al. Swizzle-switch networks for many-core systems. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2(2):278–294, 2012 2012.

[24] L. Shang, L.-S. Peh, and N. K. Jha. Dynamic voltage scaling with links for power optimization of interconnection networks. In *Proc. of HPCA*, pages 91–102, 2003.

[25] H. Wang, L.-S. Peh, and S. Malik. Power-driven design of router microarchitectures in on-chip networks. In *Proc of. MICRO*, 2003.

[26] D. Wentzlaff and et al. On-Chip Interconnection Architecture of the Tile Processor. *IEEE Micro*, pages 15–31, Sep./Oct. 2007.