Automatic Generation of Peak-Power Traffic for Networks-on-Chip

Ioannis Seitanidis, Chrysostomos Nicopoulos, and Giorgos Dimitrakopoulos

Abstract—Early estimation of the peak power consumption of a system under development is crucial in assessing the design's thermal profile and reliability, and in benchmarking the chip-level power management features. In this paper, we present a high-level systematic methodology for generating the appropriate traffic patterns that trigger the peak power consumption in a network-on-chip (NoC), irrespective of the latter's structural and functional properties. The generation of peak-power traffic is performed by solving a novel optimization problem based on integer linear programming, which models the traffic that can realistically flow in the network, thus avoiding any fake and pessimistic scenarios. This formulation can handle arbitrary network configurations and routing algorithms, including heterogeneous network topologies with multiple link widths and voltage/clock domains. The proposed technique maximizes both the network utilization and the data switching activity, thereby causing, on average, 4× higher power consumption than synthetic traffic patterns with random behavior. Most importantly, the proposed method reveals the realistic ceiling of the NoC's peak power consumption, by reporting significantly lower peak power (3× less), as compared to fake worst-case scenarios that can never, in fact, occur during the NoC's normal operation.

Index Terms—Network-on-chip (NoC), optimization, peak power consumption.

I. INTRODUCTION

TECHNOLOGY scaling has enabled digital system designs with billions of transistors integrated on a single chip. Besides the abundance of resources, which has been the driving force behind the multicore archetype, key (micro-)architectural decisions are dictated by power constraints. Excessive power dissipation increases packaging/cooling costs, reduces battery life in mobile devices, and adversely affects hardware reliability, primarily due to elevated temperatures [1]. The increasingly stringent requirement to adhere to a given power budget has rendered power consumption a first-class design constraint [2]. Hence, it is imperative for system architects to understand and accurately quantify their design's power usage from the early stages of the design process [3].

One particular salient attribute is of paramount importance: the peak (i.e., worst-case) power consumption [4], [5]. Both the system's maximum performance and implementation costs (power delivery, packaging, and cooling) are directly impacted by this worst-case power consumption. A pessimistic peak power estimate will unnecessarily curtail performance, while an optimistic estimate could potentially lead to reliability problems.

The accurate identification of worst-case power consumption is extremely challenging, especially as chips become more complex. To further compound the problem, the worst-case power usage of a system is not simply the sum of the maximum power of each component, due to under-utilization and contention for shared resources. Hence, the peak power consumption must be estimated using a stimulus that is *realistic* and resides within the functionally feasible workload space of the system under evaluation. Typically, designers rely on hand-crafted, custom-made so called *power viruses* (also referred to as "stressmarks," or "powermarks") to estimate a system's peak power consumption [6]–[8]. However, the task of manually constructing a program for a specific architecture is very cumbersome and error-prone. Most importantly, the generated virus is not guaranteed to yield the maximum possible power usage. Thus, automatic approaches to the generation of effective power viruses are highly desirable.

Power viruses have been explored within the realm of CPUs, main memory, and off-chip I/O. One notable absentee is the on-chip network; there is currently no methodology to identify the peak power consumption in the system's network-on-chip (NoC) backbone. The NoC has become the de facto communication medium in multicore setups, due to its modularity and scalability traits. Being an integral part of the system, the NoC provides connectivity across the chip, and it synergistically contributes to overall system performance [9]. Given the NoC's functional and performance criticality, it is obvious that peak-power analysis cannot ignore such an elemental actor.

Merely summing the maximum possible link power consumption and the maximum possible router consumption would result in a *fake* worst-case estimate that would be excessively pessimistic. Salient functional characteristics of

I. Seitanidis and G. Dimitrakopoulos are with the Electrical and Computer Engineering Department, Democritus University of Thrace, 67100 Xanthi, Greece (e-mail: iseitani@ee.duth.gr; dimitrak@ee.duth.gr).

C. Nicopoulos is with the Electrical and Computer Engineering Department, University of Cyprus, Nicosia 1678, Cyprus (e-mail: nicopoulos@ucy.ac.cy).

the NoC—such as the employed routing algorithm and the network topology—prohibit the simultaneous full utilization of every NoC component at the gate level.

This paper—an extension and generalization of the work in [10]—presents a high-level systematic methodology to generate realistic traffic patterns that cause peak power consumption within the NoC, by leveraging the intertwined and complementary roles of high network utilization and data switching activity. High network utilization alone is not enough to generate peak power, because, if the data payloads happen to be "favorable" (i.e., yielding low switching activity), the NoC power consumption may be quite low. Moreover, high network utilization is often confused with highly congested networks. Congestion may severely affect certain NoC regions, but may leave other regions under-utilized. Hence, formulating the generic problem of peak power consumption within the NoC involves the intricate interaction of all aforementioned nuances, which is one of the key contributions of this paper.

The proposed framework can generate a peak-power "traffic virus" for any NoC configuration. State-of-the-art NoC architectures follow regular, or irregular, topologies with homogeneous, or heterogeneous, link widths, while their constituent components (routers, links, and network interfaces) may belong to arbitrary clock and voltage domains. Consequently, the various components contribute differently to the overall power consumption of the NoC. The proposed approach handles all the constraints that arise from the heterogeneous nature of modern NoCs and generates the peak-power traffic patterns using a novel formulation based on integer linear programming (ILP), which executes in a reasonable time, even for very large networks.

Even if the proposed methodology constitutes a high-level approach, and it does not formally guarantee peak power consumption at the *gate level*, it still tackles efficiently the problem of verifying—at design time—the NoC's peak power consumption. Extensive and detailed experimental evaluations using various NoC configurations validate the effectiveness of the proposed methodology. The automatically generated traffic patterns are demonstrated to cause an average of $4\times$ higher power consumption than randomly selected traffic patterns, or patterns that result in high saturation throughput. Conversely, the proposed methodology reveals that the realistic peak power of the NoC is $3\times$ lower than the peak power reported by pessimistic scenarios that assume additional switching activity outside the functional workload space of the NoC. In this way, the true bounds of the NoC's peak power consumption are highlighted, thus preventing unnecessary over-design steps that tackle un-realistic worst-case scenarios.

The rest of this paper is organized as follows. Section II describes the intuition we developed for selecting the format of the traffic patterns that are needed to trigger the peak power consumption of the NoC in a controllable manner. Section III introduces the ILP formulation that generates the appropriate traffic pattern for each NoC configuration. Section IV presents the peak-power experimental results. Finally, conclusions are drawn in Section V.

II. PEAK-POWER TRAFFIC: KEY CHARACTERISTICS AND THE PROPOSED PROBLEM FORMULATION

The power consumption of digital systems comprises both dynamic and static components. Although static power consumption remains a serious concern [11], [12], there has been a renewed focus on dynamic power minimization within the digital circuit design industry, due to the of adoption of modern transistor technologies, such as FinFETs [13]. Consequently, the focus of this paper is on dynamic power consumption. Note that static power consumption is an issue whenever system components are idle. However, *peak* total power consumption is achieved when the system is operating at its maximum potential. Thus, the optimization goal when developing power viruses is to maximize system activity, not idleness. Nevertheless, static power consumption is accounted for and included in all subsequent experimental results.

The dynamic power consumption of an NoC jointly depends on: 1) the clock frequency and the supply voltage of its components; 2) the network component utilization that governs also the activation of any clock-gating logic; and 3) the data switching activity caused by the traffic flowing inside the NoC every cycle. A network component that carries data—e.g., the links, the buffers, the crossbar, etc.—is considered utilized, as long as it performs a useful operation in each cycle, while the amount of power consumed is directly proportional to the switching activity caused by the bits of the traversing flits. On the contrary, the dynamic power of the control portion of the NoC does not depend on the bit-level profile of the data, but only on the per-cycle activity of the arbitration and flow control logic. In state-of-the-art NoCs employing wide links (128 bits, or more), the dynamic power of the control logic is minimal compared to the dynamic power of the datapath components [14], [15]. Therefore, in this paper, we aim to trigger the maximum possible power consumption in the *datapath* components of the NoC, which are responsible for the majority of the total power consumption. This goal is achieved by maximizing both the datapath components' utilization, i.e., being active in every clock cycle, and the experienced data switching activity.

A. Interplay of Contention and Data Switching Activity

The power consumption of every NoC component and, especially, the datapath components—considering both utilization and data switching activity—is directly related to the effect of contention and multiplexing. Whenever at least two flows compete for the same resource, e.g., an NoC link through a router's output port, they will possibly gain access to the shared resource in a time-multiplexed manner, depending on the employed arbitration policy. In this case, there is no way to predict the data switching profile seen at the output of the shared resource, since the output data stream is the result of multiplexing-in-time of two (or more) flows that are unrelated in terms of their data properties.

An example of the unpredictability of the output data stream is shown in Fig. 1, assuming two 4-bit-wide data flows. Multiplexing the two flows "corrupts" the cycle-by-cycle bit-level profile of the data traveling through the multiplexer.
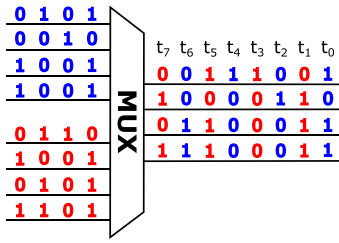
Fig. 1. Process of multiplexing different data flows "corrupts" the data switching profile of each individual incoming data stream, and can possibly lead to very low dynamic power consumption.

Although the two incoming streams exhibit considerable switching activity when viewed independently of each other, the arbitrated traffic that passes to the output of the multiplexer exhibits very few bits switching in every clock cycle.

Our goal is to control the data switching activity triggered inside the network through the data injected at the sources of the network, without any other intervention to the NoC's operation. To achieve this goal, we need to remove the unpredictability caused by intermediate contention/multiplexing points. This is enabled by injecting into the network totally *contention-free* traffic patterns. A traffic pattern is totally contention-free when the flows injected by the sources of the network never contend for the same resource in any part of the NoC. In this way, the data switching activity experienced by all the intermediate NoC components on the path from source to destination would exactly match the data switching activity of the injected traffic. This is facilitated by the fact that the unpredictability caused by multiplexing of unrelated data streams is completely avoided.

At first glance, it may seem contradictory that we employ contention-free traffic patterns, since contention-free flows are not often observed in real-world environments, and our goal from the outset was to generate a realistic stimulus. However, what we mean by a "realistic stimulus" is one that is *functionally feasible* (i.e., permitted to occur), not one that is necessarily frequently encountered. In fact, the goal of any power virus is to generate the *worst possible* load, not a typically encountered load.

It should be noted that, in the presence of traffic contention, the data switching activity may still be (possibly) predicted, and controlled directly from the sources of the NoC, but only if the details of the arbitration policy and the flow control rules [including any virtual channel (VC) allocation policy and the operational details of possibly pipelined routers] are considered. However, since we target a *high-level approach* that is *agnostic* to the NoC's micro-architectural details—in order to be easily applicable to any NoC configuration—we rely on conflict-free operation that allows us to fully control the data switching activity and, thus, maximize the observed power consumption. The efficacy of this approach is demonstrated by the experimental results presented in Section IV.

B. Permutation Traffic and Network Utilization

To identify contention-free traffic patterns, we start by removing contention at the endpoints of the network, i.e., the traffic injected by each source is directed to a different destination. Therefore, out of all possible traffic patterns, we need to identify those permutation traffic patterns where: 1) traffic is exchanged between unique source-destination pairs; 2) no flow contention is caused inside the network; and 3) network utilization is maximized.

An example of such a permutation traffic pattern is shown in Fig. 2(a) for a 3×3 2-D homogeneous mesh NoC, where all links have the same width and operate under the same clock frequency, while the paths of the injected flows are determined by the XY routing algorithm. The terminal nodes are depicted as circles in the figure. The numbers beneath each terminal node indicate the source/destination pair of the flow originating from that terminal node, e.g., the "$1 \rightarrow 8$" designation below node 1 indicates that the traffic generated at this node goes to node 8. Each source/destination pair is unique as needed by permutation traffic. Also, the selected traffic pattern completely avoids contention and allows: 1) for full NoC component utilization (injection and ejection throughput can be 100%), while, 2) at the same time, the sources can control, from outside the network, the data switching activity inside the NoC.

Not all permutation traffic patterns are appropriate for avoiding contention inside the network while still triggering the maximum network utilization. For example, the permutation traffic shown in Fig. 2(b) leaves four links idle (links $6 \rightarrow 3$, $3 \rightarrow 6$, $5 \rightarrow 8$, and $8 \rightarrow 5$), even if it completely avoids contention in the network and enables full injection/ejection throughput transmissions.

Even worse, the permutation traffic shown in Fig. 2(c) not only leaves some links idle, e.g., $1 \rightarrow 0$ and $4 \rightarrow 5$, it also preserves in-network contention. Besides causing unpredictability in the data switching activity, in-network contention also limits the utilization of certain links, i.e., they do not receive a flit every cycle. For instance, the $0 \rightarrow 1$ link is only used with 50% throughput, because the red flow going from nodes 0 to 8 encounters contention in the downstream node 1, and is forced to share the $1 \rightarrow 2$ link with the dark green flow going from nodes 1 to 5. Since the red flow only uses 50% of the bandwidth of the $1 \rightarrow 2$ link, the throughput of the $0 \rightarrow 1$ link also falls to 50%, i.e., it remains idle for 50% of the time. Once a link is un(under)-utilized, the router components connected to the link's endpoints (multiplexers, pipeline registers, flow control logic, and input buffers) also remain idle, or under-utilized.

The permutation traffic shown in Fig. 2(c) is also a very good example of why peak-power traffic selection should not be confused with traffic patterns that cause network congestion and worst-case throughput. This permutation traffic is produced using the method in [16] with the goal being to maximize the load on certain channels, in order to stress the network and highlight its worst-case throughput. However, even though the NoC is congested, many portions of it remain under-/un-utilized, and exhibit lower power consumption.

In regular and homogeneous NoC topologies, peak power is triggered by appropriately selected permutation traffic patterns, similar to the one shown in Fig. 2(a), which yield full utilization of all network links at the maximum throughput. Their
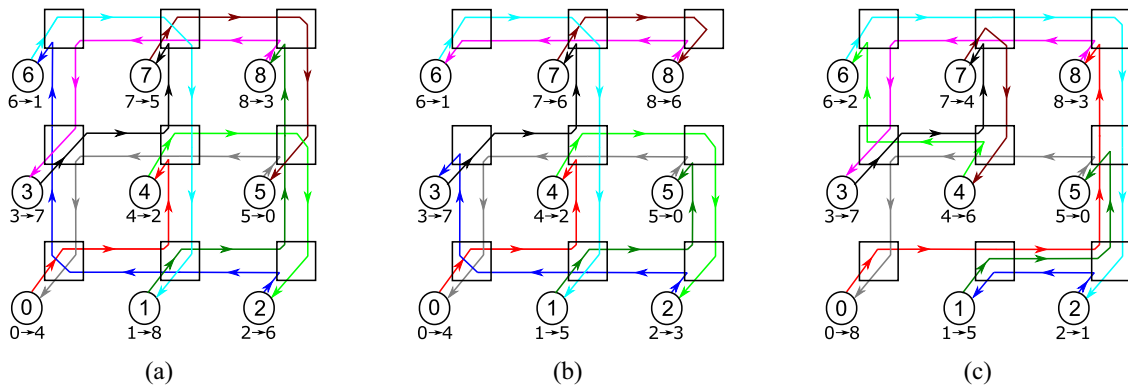
Fig. 2. Three different permutation traffic patterns in a 3×3 2-D mesh. (a) Pattern that causes maximum NoC component utilization, which yields peak power consumption (i.e., the desired pattern). (b) Pattern that leaves certain links idle, even though it achieves full injection/ejection throughput and avoids contention. (c) Pattern that merely congests certain network channels, while leaving parts of the NoC unutilized.
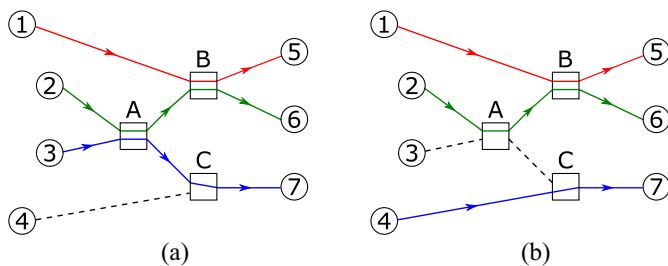

Fig. 3. Example of a heterogeneous NoC with an irregular topology and one asymmetric router. Two traffic patterns are shown (a) contention-free traffic pattern that achieves the maximum possible utilization and (b) traffic pattern that actually achieves peak power consumption, despite leaving two short links idle.
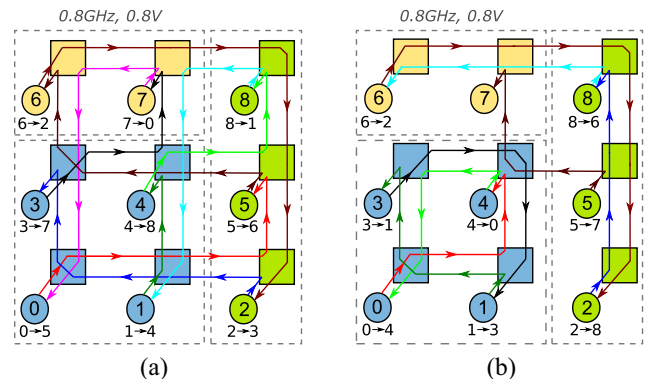

Fig. 4. Example of an NoC that is split across different voltage/frequency domains. Two traffic patterns are shown (a) contention-free permutation traffic pattern that utilizes all the NoC links and (b) traffic pattern that achieves higher peak power consumption by allowing the high-frequency links to operate at their maximum throughput (even if some other links remain idle).

contention-free operation allows the switching activity of each path to be controlled directly from the sources of the network in an end-to-end manner. Even if some NoC components are not identical—in terms of their micro-architecture—they are still fully utilized, irrespective of their differentiated design parameters. For example, all links receive full (100%) utilization, irrespective of their length. Both short and long wires are equally utilized; this attribute translates into longer wires consuming more power than shorter wires. However, identifying the appropriate permutation traffic by selecting randomly one from the set of all possible permutation traffic patters in not feasible. As shown in [10], only a few patterns (0.5% for a 3×3 mesh) fulfill the peak-power traffic requirements of allowing contention free operation and utilize all the links of the NoC. In larger networks the percentage of the favorable traffic patterns drops drastically, as well as the opportunity to find one randomly, e.g., on a 8×8 mesh $1.98 \cdot 10^{87}$ different permutation traffic patterns can occur.

C. Case of Heterogeneous NoCs

In heterogeneous NoCs, many architectural and physical parameters can vary depending on the application scenario. For instance, the NoC may consist of links with different bandwidths (i.e., the combined effect of link width and clock frequency), or parts of the NoC may belong to different voltage and frequency domains. In such cases, uniform full utilization

of all NoC components may either be infeasible, or it may even lead to lower peak-power consumption. In such cases, leaving some links idle may increase the peak power consumption observed, by increasing the utilization of the parts of the NoC with the maximum contribution to the overall power consumption.

In the example shown in Fig. 3, the NoC has an irregular topology that employs uniform link widths, clock frequency, and voltage, but exhibits several other asymmetries in both the routers and the lengths of the links. The depicted contention-free traffic pattern in Fig. 3(a) achieves the maximum utilization and, inevitably, leaves one link idle (i.e., the one that connects source 4 with router C). Note that it is impossible to concurrently utilize all input links of router C, since the number of output ports (one) is smaller than the number of input ports (two), which means that the router's single output port cannot serve both of its inputs in the same cycle. However, in terms of peak power consumption, the most favorable traffic pattern for this topology is the one used in Fig. 3(b). In this case, the links are prioritized according to their power contribution; the ones with the larger power dissipation (i.e., the longer ones) are utilized, while other links with smaller power dissipation are left idle.

Leaving some links idle may also be required when trying to increase the power consumption of NoCs that are split across different voltage/frequency domains, as the one shown in Fig. 4.[1] In Fig. 4(a), the NoC is flooded by a contention-free permutation traffic pattern that utilizes all the NoC links. If the NoC operated under a single voltage/frequency domain, this traffic pattern would have been the most appropriate for triggering the peak power consumption. However, in this case, there are paths, such as $0 \rightarrow 5$, which involve links of different bandwidths (the links have the same bit-width, but they operate at different clock frequencies). This bandwidth asymmetry throttles the fast links (operating at 1.5 GHz), and their effective throughput is determined by the slowest link of the path (i.e., the ones that belong to the 500 MHz domain). The injection throughput of source 0 that is clocked at 1.5 GHz drops inevitably to 1/3 flits/cycle_at_1.5_GHz, due to the backpressure generated by the links that operate at 500 MHz. Therefore, the links of the path $0 \rightarrow 5$ that belong to the fast clock domain are under-utilized and experience only one third of their peak power.

On the contrary, if the same NoC is driven by the permutation traffic shown in Fig. 4(b), the peak power observed would be $1.3\times$ higher than the peak power triggered by the pattern shown in Fig. 4(a). In this case, even if some links remain idle, the overall power consumption increases, since the most power-hungry links (the ones operating at 1.5 GHz) are allowed to operate at their maximum throughput of 1 flit/cycle_at_1.5_GHz, by avoiding mixing links with different bandwidths on the same path.

Overall, triggering the peak power consumption of a homogeneous, or a heterogeneous, NoC with a parameterized configuration *by only injecting traffic from the sources of the network* is a multiparameter problem with a huge design space. The approach presented in the next section can identify legal—with respect to the routing algorithm—and contention-free permutation traffic that tries to maximize—as much as possible—the power consumption of the NoC, by increasing the NoC components' utilization and their data switching activity in a controllable manner, after taking into account the NoC's structural and physical-implementation properties. By eliminating contention, the employed high-level approach allows us to control the utilization and the data switching activity experienced by the datapath components of the NoC. In this way, even though we cannot formally guarantee the generation of the maximum possible power consumption within the NoC at the gate level, we are able to significantly increase the peak power consumption of the NoC, as will be shown in the experimental results.

Other, nonpermutation traffic patterns, whereby each source can send traffic to multiple destinations can be alternatively used for triggering high power consumption within the NoC. However, such patterns either lead to contention-free traffic that under-utilizes some links of the network (e.g., nearest-neighbor traffic in a 2-D mesh, where each node sends all of

its traffic to its immediate neighbors), or they achieve full utilization of all the NoC components while allowing contention during network traversal. As previously mentioned, contention makes the triggered data switching activity unpredictable.

III. AUTOMATIC GENERATION OF PEAK-POWER TRAFFIC

The generation of contention-free traffic patterns that maximize the NoC's power consumption is performed via a novel ILP-based formulation. The goal of the power maximization problem is to find a permutation traffic pattern—i.e., each source node sends all of its traffic to a unique destination different from the other sources—that does not cause any contention inside the network, and prioritizes the usage of the most power-consuming paths.

In its most generic form, the network connects N source nodes and M sink nodes. In reality, the number of source and sink nodes is equal, $N = M$, since each network terminal can both inject and receive data to/from the network.

The source and sink nodes that are allowed to communicate are declared via binary variables $c_{ij} \in \{0, 1\}$. If $c_{ij} = 1$, then source i is eligible to send data to sink j.

The communication between a pair of source and sink nodes is performed using the links and the routers of the network. The path (i.e., the set of links and router input-output connections) that will be used for the pair's communication is solely determined by the routing algorithm. In this paper, we target only deterministic routing algorithms, as employed in the majority of industrial NoC implementations [17], [18]. With deterministic routing, each source i can communicate with each sink node j via a single path, declared as P_{ij}. Therefore, for each one of the N sources, there are M candidate paths to be selected. For each path P_{ij}, we define a binary variable $x_{ij} \in \{0, 1\}$ that declares if the path will be selected in the final permutation traffic pattern, or not. If $x_{ij} = 1$, then source i sends all of its traffic to sink j.

Permutation traffic imposes that each source i can send its traffic to at most one sink j, and each sink j can accept traffic from at most one source i. To satisfy these constraints, we need $\sum_i^N c_{ij}x_{ij} = 1$ and $\sum_j^M c_{ij}x_{ij} = 1$, where c_{ij} declares if endpoints i and j are allowed to communicate. As shown in Section II, in order to maximize the power consumption of the NoC, it may be advantageous (in some cases) if the permutation traffic is not complete, i.e., some sources do not send any traffic, or some sinks do not receive any traffic. Therefore, to allow for this behavior, the constraints should be relaxed as follows: $\sum_i^N c_{ij}x_{ij} \leq 1$ and $\sum_j^M c_{ij}x_{ij} \leq 1$.

The selected permutation traffic should not cause any contention inside the network. This is guaranteed, if each link in the network is utilized for servicing the traffic of only one path P_{ij}. If more than one paths are serviced by the same link, it means that at least two sources send traffic through the same link, thus possibly causing contention and removing the required predictability of the data switching activity (see Section II). If contention is avoided, then the switching activity of all the links and all the router input/output ports along a path P_{ij} can be directly controlled by the data injected from source i.

---

[1]In this example, we assume that voltage and clock-domain interfacing occur on the receiver side of each link. Thus, each link belongs to the voltage/frequency domain of its driver.
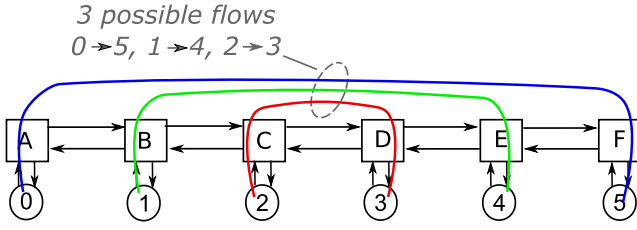
Fig. 5. Choosing traffic-flow paths within the NoC. Link $C \to D$ may be used by three different flows, but, eventually, only one should be assigned to use it, or none of them (i.e., it may remain idle).

For each link k, we enumerate the set of paths that use the corresponding link, following the rules of the selected routing algorithm. In the example shown in Fig. 5, the link that connects routers C and D is used in the set of paths $\{P_{05}, P_{14}, P_{23}\}$. This link should be assigned to only one of the three paths that can possibly use it, or to none of them. This constraint is satisfied when $x_{05} + x_{14} + x_{23} \leq 1$. A link is allowed to stay idle, if this gives more freedom to the overall power maximization problem. Transforming this inequality to an equality would constrain the ILP to use every link for serving some communication path, which is only desirable in the case of *fully homogeneous* NoC designs. The relationship between the links of the network and the paths that can use them is summarized in a 3-D binary matrix with elements $t_{ijk} \in \{0, 1\}$. Element $t_{ijk} = 1$ when link k is used by path P_{ij}. This assignment holds for both the in-network links and the incoming/outgoing links that connect the source and sink terminals to the routers of the network.

Overall, the ILP used for identifying the appropriate traffic pattern that maximizes the power consumption of the network is formulated as follows:

$$\textbf{maximize } \sum_{i=1}^{N} \sum_{j=1}^{M} w_{ij} c_{ij} x_{ij}$$

$$\textbf{subject to } \forall \text{ link } k: \quad \sum_{i=1}^{N} \sum_{j=1}^{M} t_{ijk} c_{ij} x_{ij} \leq 1$$

$$x_{ij} \in \{0, 1\}, \quad i = 1, \ldots, N \quad j = 1, \ldots, M.$$

For the solution of the ILP, i.e., the identification of the optimal values of x_{ij}, the binary variables c_{ij}, t_{ijk} are constants declaring whether source i can send traffic to sink j, and whether link k is used by path P_{ij}, respectively. The power cost of selecting the path P_{ij} is equal to w_{ij}.

For fully *homogeneous* NoCs, as the ones shown in Fig. 2, which merely need a permutation traffic pattern that is conflict-free and utilizes all the links of the NoC, we just need to set $w_{ij} = c_{ij} = 1$. The variables t_{ijk} are not simplified, and are set according to the properties of the routing algorithm.

In *heterogeneous* networks, the power cost w_{ij} of each path P_{ij} is affected by several parameters, since the routers and the links along the path between source i and sink j may belong to different voltage domains, may operate at different clock frequencies, and may have different bit-widths.

The approach in [10] identified the traffic pattern that utilized all the links of the NoC by finding a Hamiltonian path

on the enhanced channel dependency graph. Although this approach is effective in homogeneous NoCs, it cannot be applied to heterogeneous topologies, since full link utilization may either be infeasible, or it may even lead to lower peak-power consumption. On the contrary, the ILP-based optimization covers both cases effectively.

A. Power Cost of Each Path

The dynamic power consumption of a path P_{ij} is the sum of the dynamic power experienced in each segment of the path, which includes moving flits from the input buffer of one router to the input buffer of the next router. This data movement across each path segment consumes power inside the router, on the link that connects the two neighboring routers, and within the input buffer of the next (downstream) router.

Router traversal accounts for the power consumed for a buffer read and for traversing the crossbar, while also including the power expended in the control logic (input-request generation logic, arbitration, and flow control logic). The actual power cost can vary significantly based on the router's configuration, i.e., the number of input/output ports, the flit width, the number of VCs, the buffer depth of each VC, and the number of pipeline stages.

In every configuration, the power spent inside the router is heavily data-dependent. Even if a new flit is sent in every clock cycle, if these flits happen to have almost the same bit-level profile, the actual power consumption remains very low, due to minimal switching activity. Similarly, the dynamic power of the link depends on the data switching activity of the transferred bits.

In either case, we cannot have a clear estimate of the power cost of each path segment, unless we have determined the data switching activity seen by the corresponding NoC components. The data switching activity is determined by two factors. The first one is the throughput of data transfers λ_{ij}, i.e., how often a new data word (flit) enters and leaves path P_{ij} of the NoC, and the second one is the bit-level profile of each transmitted word. Once both factors are known, the assumed power cost would be accurate enough, since it would reflect both the power of the exact NoC configuration (link widths, number of VCs, etc.), and the power consumed due to the selected data profile.

In general, the power cost w_{ij} of each path P_{ij} is equal to

$$w_{ij} = \lambda_{ij} \times \sum_{s}^{\#\text{segments}} \text{Power\_of\_segment}(s).$$

The calculation of the effective injection throughput for each path λ_{ij} is described in Section III-B. The methodology of generating the bit-level data patterns that guarantee high switching activity (and are used for the determination of the Power_of_segment) is presented in Section III-C.

B. Effective Throughput of Each Path

Even if the paths derived from the ILP will be conflict-free, we may not be able to inject new flits at the maximum throughput of 1 flit/cycle in certain paths. This phenomenon appears only in *heterogeneous* NoCs that consist of paths with links
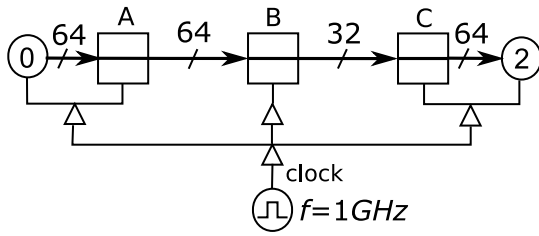
Fig. 6. In this example, all links operate at the same clock frequency (1 GHz), but one link is half as wide as the other links (32 versus 64 bits). Consequently, the narrow link will throttle the flow of data due to serialization, thereby limiting the maximum utilization and switching activity observed on the wide links.

of different bandwidth. In homogeneous NoCs, full-throughput data transfers on all links is *always* achieved.

In the example shown in Fig. 6, all the links operate at the same clock frequency (1 GHz), but one link is narrower. Thus, all links can transfer 64-bit flits, except one link that can transfer 32-bit flits. In this case, the narrow link will throttle the flow of data due to (de)serialization under the same clock frequency. The effective throughput seen at the injection source, which consists of wide links, would be half of the maximum possible. Hence, the effective switching activity experienced on the wide links would be half of the maximum that could be experienced if all the links of the $0 \rightarrow 2$ path had equal bandwidth.

If we denote the bandwidth of each link k as $\mathrm{BW}_k = f_k \times W_k$, where f_k is the clock frequency of the driver of the link and W_k is the link's bit-width, the effective injection throughput of a path P_{ij} that consists of multiple links in series (separated by routers) is equal to

$$\lambda_{ij} = \frac{\min \mathrm{BW}_k}{\mathrm{BW}_i}, \quad \forall \, \mathrm{link}\, k \in P_{ij}.$$

C. Maximizing the Data Switching Activity

The final step for the determination of the power cost w_{ij} of each path P_{ij} is to estimate the power that is dissipated in each router and link of the NoC (i.e., Power_of_segment). To do so, we rely on real NoC implementations using state-of-the-art EDA tools driven by data patterns that maximize switching activity.

Identifying accurately the appropriate (worst-case) data patterns can be done only using specific gate-level techniques [19], [20]. However, such techniques can be applied only in certain sub-modules of the NoC and cannot be extended to the network-path level. Therefore, to tackle the problem of identifying the worst-case data patterns, we need to consider only the microarchitecture-level aspects of the NoC and focus on the microarchitectural features that are found in the majority of NoC designs.

For the links, a repetitive data pattern that switches between $0101 \cdots 01 \rightarrow 1010 \cdots 10$ is enough to trigger worst-case power consumption. Each bit experiences a change in every cycle, either $0 \rightarrow 1$ or $1 \rightarrow 0$, which switches the corresponding capacitance of the wire to ground. Further, this data pattern ensures that neighboring wires always switch in the opposite

direction, thereby causing the worst-case power consumption, due to the link's coupling capacitance [12].

However, for the VC buffers and the internal logic of the router, we cannot be sure of the exact switching activity caused by this 2-data vector pattern. Assume, for example, the case of VC buffers that are built using register-based (i.e., flip-flop-based) FIFO queues, or using SRAM blocks. In either case, power is consumed every time a new flit is written to, or read from, the VC buffers. On each write, a new flit is written to only one VC. Inside the queue of each VC, the flit is written into the register that corresponds to the address pointed to by the tail pointer of the enabled VC queue. Therefore, on each write, only the bits of one register can change value. The rest are not enabled, or remain clock-gated, as normally done in industrial NoC implementations. Therefore, to maximize power, we need to guarantee that: 1) the new value written to the register is different from the one already stored and 2) the two values (the old and the new one) differ by as many bits as possible. This can only occur if we know beforehand the specific slot of the VC queue into which the incoming flit will be stored.

When a repetitive data pattern of D words (flits) is placed—one word after the other—in a buffer with B slots, then we can guarantee that any incoming word will be written (stored) into a register that already stores a different value, as long as the greatest common divisor of D and B is equal to one. When B is odd, the 2-vector data pattern that also maximizes the power on the links is the proper choice. When B is even, we can select a repetitive data pattern of $B + 1$ words. The $B + 1$ words can safely include $B/2$ repetitions of the 2-vector data pattern $0101 \cdots 01 \rightarrow 1010 \cdots 10 \rightarrow 0101 \cdots 01 \rightarrow 1010 \cdots 10$, plus an all-zero vector. Depending on the NoC configuration, the repetitive set of data patterns can also extend across different packets, as long as the flits of the packets flow consecutively in the NoC.

In terms of power, the traffic injected can stay within the same VC from source to destination, as long as one flit is written and read per cycle, and the data values written and read have the maximum bit-wise difference. Distributing traffic across VCs for each nonconflicting flow produced by the proposed method is possible, but it needlessly complicates the derivation of the appropriate data switching patterns that cause the maximum switching activity, without any true impact on the triggered power consumption.

Even though the nonconflicting nature of the traffic patterns can maximize the switching activity in the datapath of the NoC (links, buffers, and crossbar), the arbitration part is kept operating on the same requests and grants in each cycle. This causes minimum switching activity in this portion of the NoC. However, this is not a problem in modern NoCs with wide datapaths of 64 bits or more, since the power of the arbitration logic is low relative to the datapath portion [15]. This argument is also verified by the experimental results in Section IV, when using random traffic. The latter maximizes the switching activity in the arbitration modules, due to the random nature of the input requests. Even when compared with this traffic scenario, the proposed technique achieves significantly higher power consumption by only appropriately
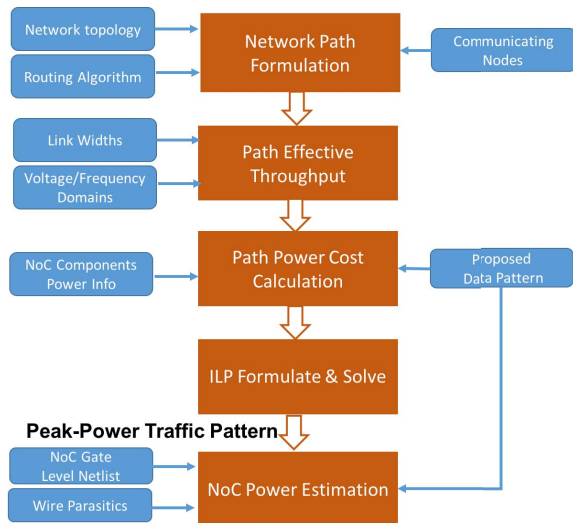
Fig. 7. Overall flow of the proposed methodology, which can be used to derive peak-power traffic patterns.
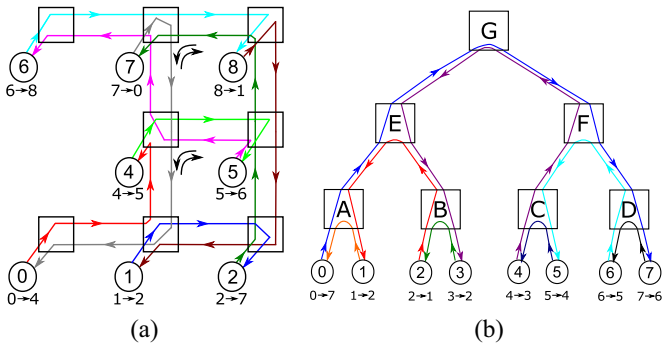

Fig. 8. Peak-power traffic patterns generated by the proposed methodology for (a) irregular network, such as an asymmetric 2-D mesh, and for (b) tree topology.

targeting the switching activity in the part of the NoC that carries data.

D. Overall Methodology Flow and Examples

The steps involved in the application of the proposed methodology is summarized in the flow depicted in Fig. 7. Driven by: 1) the NoC topology that describes the NoC's structure, the voltage/frequency domains, and the link widths; 2) the routing algorithm; and 3) the set of nodes that are allowed to communicate, we form the network paths across all pairs of source and sink nodes, and then calculate the power cost of each path. For the calculation of the power cost of each path, we use the power precomputed for each NoC component. Once the weight for every pair of communicating nodes is known, the ILP is formulated and solved using the Gurobi solver [21]. The derived traffic patterns and the proposed data patterns then drive timing-accurate gate-level logic simulations. These simulations provide the actual switching activity that is subsequently used to calculate the power consumption of the NoC.

Fig. 8(a) depicts the ILP-derived nonconflicting permutation traffic that causes 100% link utilization in an asymmetric
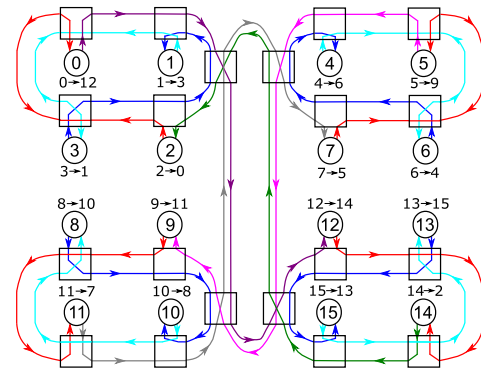

Fig. 9. Peak-power traffic patterns derived by the proposed methodology for a hierarchical ring, which employs VCs for deadlock freedom.
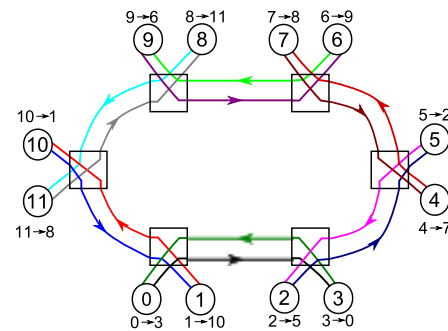

Fig. 10. Peak-power traffic patterns derived by the proposed methodology for a ring that employs concentration, whereby two terminal nodes are connected per router.

2-D mesh network (the asymmetry is the result of a faulty router which is decommissioned and not shown in the figure). In this case, the turning restrictions of the routing algorithm [22] (depicted as small arrows at certain turn-points within the network) guarantee connectivity and deadlock freedom. Equivalently, the peak power traffic for a tree that applies the up/down routing algorithm is highlighted in Fig. 8(b).

Fig. 9 illustrates a peak-power traffic scenario for a hierarchical ring. Ring and tori topologies employ VCs to ensure freedom from possible routing deadlocks. To model the specific use of VCs for deadlock-free routing, we include each link multiple times in the ILP; as many times as the number of supported VCs per network link. Eventually, after the solution of the ILP, only one VC will be used per physical channel of the network to carry the generated peak-power traffic. The derived traffic flows are allowed to change VC in-flight, as long as this is dictated by the routing algorithm; in any other case, each traffic flow remains within the same VC. In this way, the ILP exercises all possible turns at the VC level, but, ultimately, it selects only one VC-to-VC connection. The proposed optimization does not impose any specific rule for acquiring a VC, other than the rules imposed by the routing algorithm.

The proposed methodology can also be applied—without any changes—to the case of NoC topologies that employ *concentration*, i.e., where multiple terminal nodes are connected per router. An example peak-power traffic scenario derived by the proposed approach on a concentrated ring is shown

TABLE I
RUN-TIMES OF THE PROPOSED TECHNIQUE. TIME NEEDED TO DERIVE
PEAK-POWER PERMUTATION TRAFFIC PATTERNS

#Nodes	Ring	Hierarchical Ring	2D Mesh	3D mesh	2D Torus	Hetero 2D Mesh
16	0m 1s	0m 9s	0m 2s	0m 1s	0m 1s	0m 4s
64	0m 5s	0m 6s	0m 4s	0m 3s	0m 2s	0m 8s
256	0m 17s	0m 12s	0m 11s	0m 10s	0m 8s	0m 21s
1024	10m 32s	8m 27s	21m 06s	21m 31s	21m 47s	24m 43s

in Fig. 10. Depending on the NoC configuration, the derived peak-power traffic patterns may involve both local and global traffic. Local traffic involves the exchange of traffic across terminal nodes connected to the same router, while global traffic involves traffic exchanged across terminal nodes that belong to different routers. Both traffic cases enable the maximization of the power consumption across the NoC's datapath, since they utilize—as much as possible—both the routers' internal datapath and the NoC links.

E. Complexity of the ILP

The complexity of the ILP is determined by the number of variables x_{ij}, i.e., $N \times M$, and the number of constraints, which is equal to the number of links in the NoC. For practical NoC cases of up to 1024 nodes, the ILP can easily be solved within a reasonable execution time. The peak-power traffic identification problem, including both path enumeration and weight calculation, as well as the solution of the ILP (assuming that the power of routers and links is computed beforehand), can complete its execution in just a *few minutes* for well-known NoC topologies of *hundreds of nodes*. The run-times required to derive peak-power permutation traffic patterns for various topologies and various network sizes are shown in Table I. The run-times correspond to an implementation that runs on a Linux computer with a 2.3 GHz Intel Core i7-4712HQ Processor and 16 GB of RAM.

The investigated topologies correspond to well-known homogeneous networks and a heterogeneous 2-D mesh that is split into 3 voltage/frequency domains, similar to Fig. 4(a). The size of each domain grows proportionally to the overall network size. From the runtimes reported in Table I, deriving the peak-power traffic patterns for the heterogeneous 2-D mesh (right-most column) requires more time than the homogeneous cases. The extra time is spent for weight calculations and for solving the ILP. Nevertheless, the reported runtimes are manageable even for very large NoCs.

IV. EXPERIMENTAL EVALUATION

The goal of the proposed method is to trigger the peak-power consumption of an NoC by injecting appropriately selected traffic patterns that maximize network component utilization and data switching activity. The proposed traffic patterns do not formally guarantee the maximization of the power consumption of the NoC at the *gate level*. However, they cause significantly higher peak power than random traffic, and, since they do not rely on the gate-level details of the NoC components, they can be successfully applied to multiple NoC configurations. In any case, it should be stressed that the

proposed traffic patterns aim at the on-purpose maximization of the power consumption. Therefore, they represent—by construction—a *corner case* of the NoC's operation, which is expected to occur rarely under normal system operation.

The experiments evaluate 64-node NoCs following 2-D mesh and hierarchical ring topologies. Other tested topologies show similar trends. Both homogeneous and heterogeneous variants of the aforementioned topologies are evaluated. In order to contain the number of possible configurations, we assume a tile-based chip floor-plan similar to the Scorpio chip [23]. Scorpio was built at 45 nm technology (which matches the technology library we used in our implementations), using a tile size of approximately 2×2 mm. Based on the chosen NoC topology, the NoC routers can have a variable number of input and output ports. For every configuration, we assume that the NoC supports four VCs per input port, with five buffer-slots/VC, and the NoC routers employ the 3-stage pipelined organization of Scorpio routers [23].

All NoC components used in the evaluation were implemented in SystemVerilog, mapped to a commercial low-power 45 nm 0.8 V standard-cell library, and placed-and-routed using the Cadence digital implementation flow. Depending on the NoC topology, a different placement-and-routing round was conducted.

Power was measured after performing timing-accurate simulations, using the proposed data patterns and including all back-annotated layout parasitics. Power measurements were performed twice: once for characterizing the power cost of each NoC component, as needed for the computation of the weights of the ILP, and, second, for deriving the final power of the NoC when it operates on the selected traffic pattern produced by the ILP.

A. Homogeneous NoCs

In the first set of experiments we evaluated the proposed methodology on homogeneous NoCs operating at 1 GHz. In this case, the inter-router NoC links carry 64 bits of data, plus some extra flow control information. The header flit, which also includes network-addressing information, carries fewer actual data bits.

In the first set of experiments, we compare the proposed method against random synthetic traffic patterns, under various data switching and network-injection scenarios. The instantaneous power consumed by an NoC when the incoming traffic causes contention across flows with unrelated data (this occurs in almost all cases under normal operation) can vary significantly over time, depending on the switching activity in various parts of the network in each cycle. This behavior is highlighted in Fig. 11, for an 8×8 2-D mesh and a 64-node two-level hierarchical ring that consists of 8-node local rings connected via an 8-node global ring.

Both networks receive uniform-random traffic at a different rate (close to their saturation throughput), as reported in Fig. 11. The two NoCs have equal link width, i.e., 64 bits plus flow-control bits, and both operate at 1 GHz. Therefore, the bisection bandwidth of the 2-D mesh is larger than the bisection bandwidth of the hierarchical ring. The injected packets
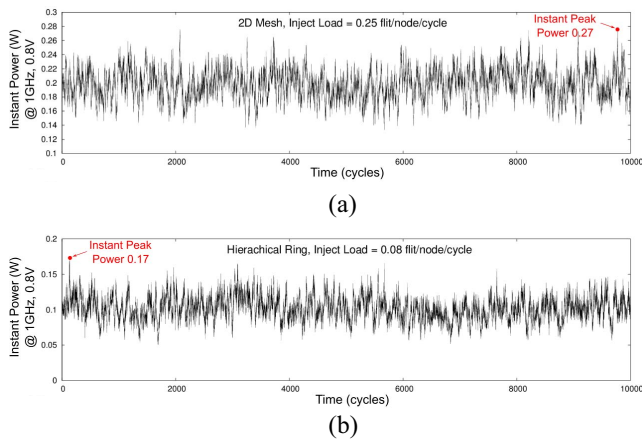
Fig. 11.   10K-cycle snapshot of the instantaneous power consumption of a homogeneous 64-node 2-D mesh (top) and a hierarchical ring (bottom), after the network reaches steady-state operation, using uniform-random traffic and data.
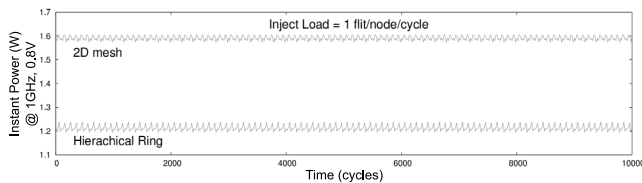

Fig. 12.   10K-cycle snapshot of the instantaneous power consumption of a 64-node 2-D mesh and a hierarchical ring (under steady-state network operation), using traffic/data derived by the proposed methodology, with full injection throughput.
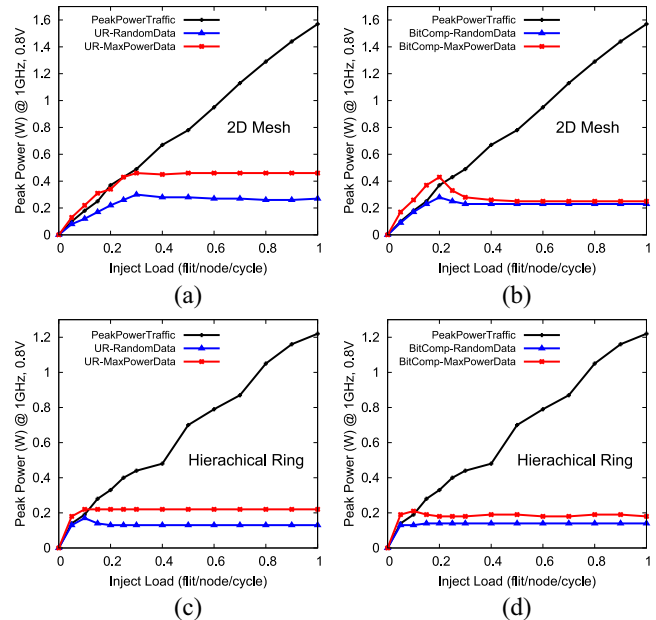

Fig. 13.   Peak power consumption versus injection load in *homogeneous* NoC topologies. The power consumption triggered by the proposed methodology, as compared to the power consumed when using uniform-random (UR) and bit-complement (BitComp) traffic patterns.

are 5-flit long and carry random data in their payload portion. In this experiment, the bit of each flit when entering the network has equal probability of being 0 or 1, independent of the rest of the bits of the same flit, or the previous flits.

The peak power consumption achieved by random traffic is merely the peak instantaneous power observed during the simulation's time frame. There is no guarantee that a large power value can be triggered during simulation, due to the unpredictability in switching activity, and the lower NoC utilization caused by contention among different flows. Additionally, the observed peak power consumption simply represents an *instantaneous* peak. This cannot be sustained over a longer period of time, which would be required to observe possible temperature increases and identify thermal hot-spots in the system.

On the contrary, the proposed method does not have such limitations. In Fig. 12, we report the instantaneous power consumed by the proposed approach under 100% injection load for each case (2-D mesh and hierarchical ring). The results are measured by injecting 5-flit packets in the NoC, following the ILP-derived permutation traffic pattern, and carrying data payloads with the 2-vector data patterns described in Section III-C. Evidently, the proposed methodology keeps power consumption constantly and consistently very high. The minimal variance in the power consumption is due to the switching profile of the header and flow-control bits, which are not controlled by our ILP-based approach.

Next, we compare the peak power consumption of the proposed method and random traffic scenarios (*uniform-random* and *bit-complement* traffic), under the same injection load. For each injection load, the maximum instantaneous power consumption value observed (over 500 000 cycles of simulation) was recorded. The results are depicted in Fig. 13, for the same 2-D mesh and hierarchical ring topologies. The peak power consumption of random traffic (blue curves) follows the throughput behavior of the network itself, and, after saturation (when the utilization of NoC components reaches its limit), the peak power consumption observed is rather constant. On the contrary, the proposed approach can increase the power consumption to its true maximum value, due to its nonconflicting traffic. The data switching activity is directly controllable by the input sources, and it covers all the intermediate router ports and network links that are utilized by the injected flow. When compared against uniform-random traffic [Fig. 13(a) and (c)], the proposed technique triggers maximum power consumption, which can be more than 6× higher than the one achieved under uniform-random traffic with random data (i.e., blue curves).

This is also true when the NoC is driven by uniform-random traffic that allows contention in the network, but the injected *data patterns* are the same as the ones used in the proposed case (by following the guidelines described in Section III-C). This scenario is also depicted in Fig. 13(a) and (c) with the red curves. The ILP-driven approach still consumes significantly more power (4× higher), since it simultaneously takes into account both the network utilization and the data switching activity.

Similar conclusions are derived when the power consumption of the NoC is triggered using other permutation traffic

TABLE II
PEAK POWER OF THE PROPOSED METHOD VERSUS THE POWER OF A
FAKE SCENARIO, WHICH ASSUMES THAT EVERY CIRCUIT NODE
SWITCHES IN EVERY CYCLE

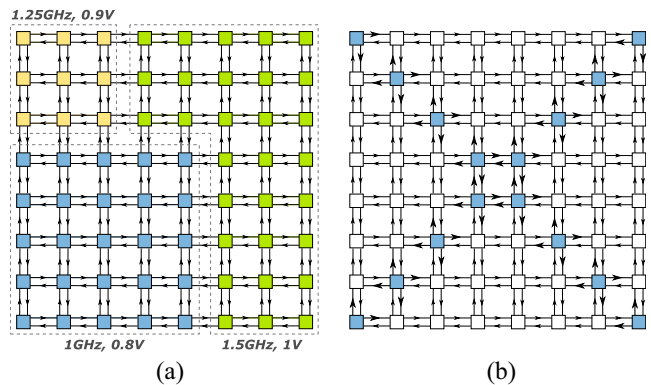| @1 GHz, 0.8 V	Fake (pessimistic)	Proposed
2D Mesh	4.87 W	1.6 W
Hier. Ring	3.76 W	1.23 W

Fig. 14.   Two *heterogeneous* NoC topologies are evaluated: (a) one that includes multiple voltage/frequency domains and (b) one that employs heterogeneous link widths and routers.
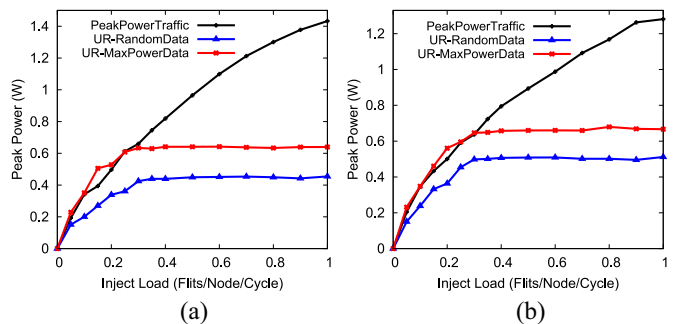

Fig. 15.   Peak power consumption versus injection load in *heterogeneous* NoC topologies. The power consumption triggered by the proposed method, as compared to the power consumed when using uniform-random traffic patterns in the two examined heterogeneous topologies. (a) Multiple voltage domains. (b) Heterogeneous link widths.

patterns, such as bit-complement traffic. In this case [Fig. 13(b) and (d)], the peak power consumption of the proposed method is 4× larger than the *largest* power observed under the bit-complement traffic patterns (red curves).

It should be noted that, at low injection rates, the red curves in Fig. 13 (i.e., random traffic scenarios using the data patterns proposed in this paper) are—in some cases—slightly higher than the black curves. However, this is an artifact of the unpredictability in the data switching activity caused by traffic contention. This unpredictability causes large variance in the recorded peak power consumption, which makes the maximum value reported in Fig. 13 very hard to repeat in a systematic manner.

Finally, we compare the power triggered by the proposed peak-power traffic, versus the peak-power consumption that corresponds to the fake scenario of every circuit node switching in every cycle. In both cases depicted in Table II, the peak power triggered by the proposed approach (measured at 100% injection rate) is lower than the one derived using the fake (un-realistic) approach. The difference between these two maximum power values depends on topology characteristics, and the power expended on the links versus the power expended within the routers. In any case, the significant conclusion out of this comparison is that fake peak-power scenarios overestimate the true maximum power profile of the NoC and unnecessarily increase the overall system power budget. With the proposed optimization, worst-case power analysis is brought closer to what is attainable by a corner-case but realistic traffic pattern.

B. Heterogeneous Topologies

In addition to homogeneous topologies, the proposed methodology can also handle *heterogeneous* topologies. To test the effectiveness of the ILP-driven peak-power generation methodology when dealing with heterogeneous NoCs, we experimented with the two heterogeneous topologies shown in Fig. 14. The first one involves three voltage/frequency domains assuming homogeneous links of 64 bits, while the second one assumes the same voltage and clock frequency of 1 GHz@0.8 V throughout the NoC, but includes two different link widths, 64 and 128 bits, following a topology similar to the one presented in [24].

Similar to the homogeneous case, we initially compare the peak-power consumption of the proposed method and a uniform-random traffic scenario, under the same injection load. The results are depicted in Fig. 15. For each injection load, the maximum instantaneous power consumption value observed (over 500 000 cycles of simulation) was recorded. Note that the injection load reported in the *x*-axis of the diagrams of

Fig. 15 refers to the injection load of the sources with the lower bandwidth. In Fig. 15(a), the injection load refers to the sources with the lower clock frequency, while, in Fig. 15(b), it refers to the sources with the narrower link width.

The ILP-based optimization guarantees the generation of nonconflicting permutation traffic and increases—at the same time—the effective injection bandwidth between each source and destination pair. This enables the application of the worst-case traffic pattern, driven by appropriately selected data, at the maximum possible rate. This property increases the maximum power observed, as compared to the power observed when driving the NoC with a uniform-random traffic pattern. Even if the random traffic uses the proposed worst-case data patterns, it still produces 2× lower power consumption than the proposed traffic. The conflicting nature of uniform-random traffic inevitably saturates the NoC, thus limiting the maximum utilization and, effectively, the power consumption that can be observed within the NoC.

In the last set of experiments, we further highlight the need for nonconflicting traffic that would guarantee the control of the data switching activity in all NoC components, while still offering maximum NoC component utilization. We relax some of the constraints of the ILP to produce one additional traffic pattern that fully utilizes all NoC components, but it *allows contention during packet network traversal*. In this scenario,
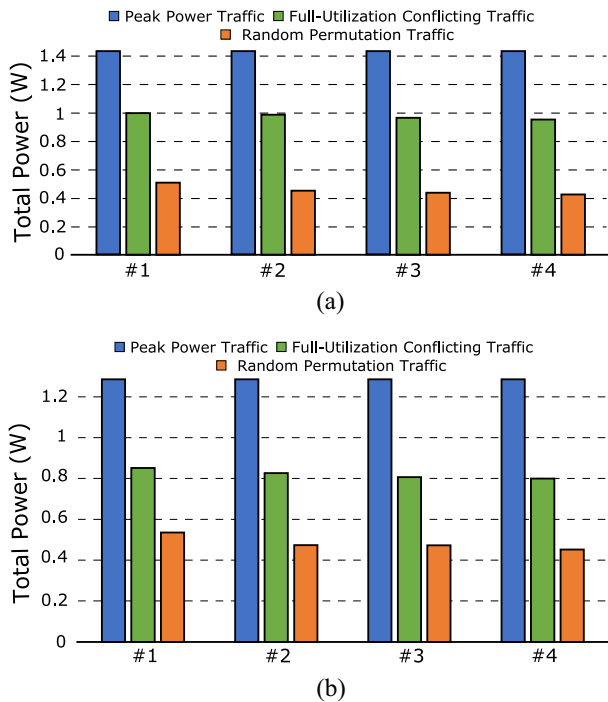
Fig. 16. Peak-power consumption observed using (1) proposed methodology, (2) four best (in terms of peak-power consumption) traffic patterns that allow for *contention* within the NoC, and (3) four best (in terms of peak-power consumption) permutation traffic patterns among 100K randomly generated patterns. (a) Multiple voltage domains. (b) Heterogeneous link widths.

each NoC source is allowed to send traffic to multiple destinations, and each destination can receive traffic from multiple sources. The ILP selects the appropriate percentage of traffic injected for each source-destination pair that guarantees maximum link utilization, but without guaranteeing contention-free network traversal. The derived traffic resembles the traffic that is produced by a maximum-flow-like algorithm [25], with the restriction that the traffic that floods the network links should use paths that are allowed by the routing algorithm of the NoC.

Traffic is injected at the maximum possible rate for 500 000 cycles, using the data patterns proposed in Section III-C. The four highest recorded power values when using these traffic patterns—that allow for network *contention*—are included in the diagrams shown in Fig. 16, next to the peak-power consumption derived by the proposed method, assuming full injection throughput. The largest power measurements recorded for the generated traffic were always lower than the power triggered by the proposed methodology: 32% lower in the case of NoC topologies with multiple voltage domains [Fig. 16(a)], and 36% lower in NoCs with heterogeneous link widths [Fig. 16(b)]. This result is a direct consequence of the contention that appears in the NoC, which, inevitably: 1) leaves some links unutilized for some cycles during the NoC's operation and 2) destroys any predictability in the data switching activity inside the NoC.

The appropriate permutation traffic patterns that yield extremely high link utilization constitute an extremely small subset of the entire set of possible permutation traffic patterns. Therefore, randomly deriving an effective permutation traffic

pattern, without relying on the proposed ILP formulation, should not be considered a viable/safe option.

To test this argument, we randomly generated 100 000 permutation traffic patterns (self-traffic was not allowed), and, for each one, we measured the peak-power consumption after injecting traffic at the maximum possible rate for 500 000 cycles. The four highest peak-power measurements recorded were also included in the diagrams shown in Fig. 16. The highest power measurements we got for the randomly generated traffic were always significantly lower than the power triggered by the proposed method.

V. CONCLUSION

As chips become increasingly more dense and complex, power consumption becomes a primary design constraint. It is imperative for designers to realistically estimate a design's peak power consumption, which directly impacts other salient system attributes, such as performance, implementation costs, battery life, and reliability. This paper introduces a fully automated high-level methodology to generate appropriate traffic and data patterns that cause peak power consumption within the NoC. The peak power consumption triggered by the proposed method is, on average, $4\times$ higher (up to $8\times$ higher) than what is observed after simulating random traffic and data patterns.

The introduced ILP-based optimization enables the power maximization of the majority of NoC components, irrespective of their differentiated design parameters. Heterogeneous and homogeneous NoCs are handled in a unified manner, allowing for the generation of appropriate traffic patterns—even for large topologies—within reasonable execution time.

This paper focuses on maximizing the power consumption of the core of the NoC, i.e., the NoC routers and links. The derived traffic patterns can be applied as a corner-case scenario during hardware simulation/verification. Our future work will focus on extending this methodology to the generation of equivalent read/write transaction-level traffic, which can be reproduced at the software level.

REFERENCES

[1] T. S. Rosing, K. Mihic, and G. De Micheli, "Power and reliability management of SoCs," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 4, pp. 391–403, Apr. 2007.

[2] S. Borkar and A. A. Chien, "The future of microprocessors," *Commun. ACM*, vol. 54, no. 5, pp. 67–77, May 2011.

[3] N. S. Kim, T. Austin, T. Mudge, and D. Grunwald, *Challenges for Architectural Level Power Modeling*. Boston, MA, USA: Springer, 2002.

[4] Y. Jin, E. J. Kim, and K. H. Yum, "Peak power control for a QoS capable on-chip network," in *Proc. Int. Conf. Parallel Process. (ICPP)*, Oslo, Norway, 2005, pp. 585–592.

[5] V. Kontorinis, A. Shayan, D. M. Tullsen, and R. Kumar, "Reducing peak power with a table-driven adaptive processor core," in *Proc. Int. Symp. Microarchit.*, New York, NY, USA, 2009, pp. 189–200.

[6] K. Ganesan *et al.*, "System-level max power (SYMPO): A systematic approach for escalating system-level power consumption using synthetic benchmarks," in *Proc. Int. Conf. Parallel Archit. Compilation Tech. (PACT)*, Vienna, Austria, 2010, pp. 19–28.

[7] S. Polfliet, F. Ryckbosch, and L. Eeckhout, "Automated full-system power characterization," *IEEE Micro*, vol. 31, no. 3, pp. 46–59, May/Jun. 2011.

[8] K. Ganesan and L. K. John, "Maximum multicore power (MAMPO): An automatic multithreaded synthetic power virus generation framework for multicore systems," in *Proc. ACM Intern. Conf. High Perform. Comput. Netw. Stor. Anal. (SC)*, Seattle, WA, USA, 2011, pp. 1–12.

[9] B. Mathewson, "The evolution of SOC interconnect and how NOC fits within it," in *Proc. Design Autom. Conf.*, Anaheim, CA, USA, 2010, pp. 312–313.

[10] I. Seitanidis, C. Nicopoulos, and G. Dimitrakopoulos, "Powermax: An automated methodology for generating peak-power traffic in networks-on-chip," in *Proc. IEEE/ACM Int. Symp. Netw. On-Chip (NOCS)*, Nara, Japan, 2016, pp. 1–8.

[11] R. Chadha and J. Bhasker, *An ASIC Low Power Primer*. New York, NY, USA: Springer, 2013.

[12] N. Weste and D. Harris, *CMOS VLSI Design a Circuits and Systems Perspective*, 3rd ed. Boston, MA, USA: Addison-Wesley, 2010.

[13] A. Narayanan, S. Jilla, and D. Chinnery, "Low-power physical design the with mentor place and route system," Wilsonville, OR, USA, Mentor Graphics, White Paper, 2016.

[14] J. Balfour and W. J. Dally, "Design tradeoffs for tiled CMP on-chip networks," in *Proc. 20th ACM Int. Conf. Supercomput. (ICS)*, Cairns, QLD, Australia, Jun. 2006, pp. 187–198.

[15] A. B. Kahng, B. Li, L.-S. Peh, and K. Samadi, "ORION 2.0: A fast and accurate NoC power and area model for early-stage design space exploration," in *Proc. Conf. Design Autom. Test Europe*, Nice, France, 2009, pp. 423–428.

[16] B. Towles and W. J. Dally, "Worst-case traffic for oblivious routing functions," in *Proc. ACM Symp. Parallel Algorithms Archit. (SPAA)*, Winnipeg, MB, Canada, 2002, pp. 1–8.

[17] P. Boucard and L. Montperrus, "Message switching system," U.S. Patent 7 639 704, 2009. [Online]. Available: https://www.google.com/patents/US7639704

[18] J. Philip, S. Kumar, E. Norige, M. Hassan, and S. Mitra, "Automatic construction of deadlock free interconnects," U.S. Patent 9 244 880, 2016. [Online]. Available: https://www.google.com/patents/US9244880

[19] Q. Wu, Q. Qiu, and M. Pedram, "Estimation of peak power dissipation in vlsi circuits using the limiting distributions of extreme order statistics," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 20, no. 8, pp. 942–956, Aug. 2001.

[20] K. Najeeb *et al.*, "Power virus generation using behavioral models of circuits," in *Proc. IEEE VLSI Test Symmp.*, Berkeley, CA, USA, 2007, pp. 35–42.

[21] *Gurobi Optimizer Reference Manual*. Accessed: Dec. 15, 2017. [Online]. Available: http://www.gurobi.com/documentation/7.0/refman/index.html

[22] P. Ren, X. Ren, S. Sane, M. A. Kinsy, and N. Zheng, "A deadlock-free and connectivity-guaranteed methodology for achieving fault-tolerance in on-chip networks," *IEEE Trans. Comput.*, vol. 65, no. 2, pp. 353–366, Feb. 2016.

[23] B. K. Daya *et al.*, "SCORPIO: A 36-core research chip demonstrating snoopy coherence on a scalable mesh NoC with in-network ordering," in *Proc. Int. Symp. Comput. Archit.*, Minneapolis, MN, USA, Jun. 2014, pp. 25–36.

[24] A. K. Mishra, N. Vijaykrishnan, and C. R. Das, "A case for heterogeneous on-chip interconnects for CMPs," in *Proc. Int. Symp. Comput. Archit. (ISCA)*, San Jose, CA, USA, 2011, pp. 389–400.

[25] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. Cambridge, MA, USA: MIT Press, 2009.

Ioannis Seitanidis received the Diploma degree in electrical and computer engineering from the Democritus University of Thrace, Xanthi, Greece, in 2013, where he is currently pursuing the Ph.D. degree in computer engineering.

His current research interests include electronic design automation algorithms, on-chip interconnection networks, and computer architecture.

Chrysostomos Nicopoulos received the B.S. and Ph.D. degrees in electrical engineering with a specialization in computer engineering from Pennsylvania State University, State College, PA, USA, in 2003 and 2007, respectively.

He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, University of Cyprus, Nicosia, Cyprus. His current research interests include networks-on-chip, computer architecture, multi/many-core microprocessor, and computer system design.

Giorgos Dimitrakopoulos received the B.S., M.Sc., and Ph.D. degrees in computer engineering from the University of Patras, Patras, Greece, in 2001, 2003, and 2007, respectively.

He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Democritus University of Thrace, Xanthi, Greece. His current research interests include design of digital integrated circuits, electronic design automation, and computer architecture, with emphasis in low-power systems design.