HELIX: Design and Synthesis of Hybrid Nanophotonic Application-Specific Network-On-Chip Architectures

Shirish Bahirat and Sudeep Pasricha
ECE Department, Colorado State University, Fort Collins, CO
{shirish.bahirat, sudeep}@colostate.edu

Abstract- Hybrid nanophotonic-electric networks-on-chip (NoC) have been recently proposed to overcome the challenges of high data transfer latencies and significant power dissipation in traditional electrical NoCs. But hybrid NoCs with nanophotonic guided waveguides and silicon microring resonator modulators impose many challenges such as high thermal tune up power and crossing losses. Due to these challenges productization of such architectures has yet to become commercially viable. Unfortunately, increasing embedded application complexity, hardware dependencies, and performance variability makes optimizing hybrid NoCs a daunting task because of the need to explore a massive design space at the system-level. To date, no prior work has addressed the problem of synthesizing application-specific hybrid nanophotonic-electric NoCs with an irregular topology. Considering the above unaddressed major challenges, in this paper we propose the *HELIX* framework for application-specific synthesis of hybrid NoC architectures that combine electrical NoCs with free-space nanophotonic NoCs. Based on our experimental studies, we demonstrate that our *HELIX* framework produces superior NoC architectures that achieve 3.06× power improvements compared to synthesis frameworks proposed in prior work for electrical NoCs.

1. Introduction

To satisfy power density and performance requirements, future *systems-on-chip (SoCs)* will inevitably integrate multiple cores on a single die [1]. *Network-On-Chip (NoC)* architectures will play a crucial role to ensure reliable and scalable interconnects between processing cores, memory modules, cache banks, and I/O devices. Unfortunately, traditional electrical NoCs are already severely constrained due to their long multi-hop latencies and high power consumption, making it very difficult in practice to stay within on-chip power budgets. Not surprisingly, the NoC is expected to be the key limiting factor for achieving operational goals related to power and latency constraints in emerging highly parallel SoCs [2].

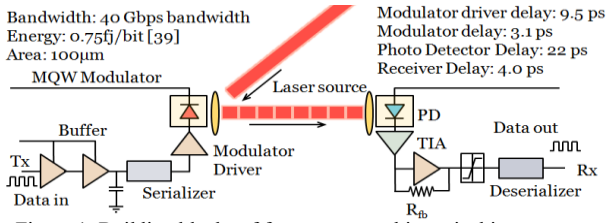

Figure 1: Building blocks of free-space on-chip optical interconnects

CMOS compatible on-chip photonic interconnects with silicon-on-insulator waveguides provide a potential substitute for electrical interconnects, particularly for global on-chip communication, allowing data to be transferred across a chip with much faster light signals. Based on recent progress, the critical length at which photonic links are advantageous over electrical links has fallen well below expected chip die size dimensions [3]. Photonic interconnects represent a promising approach for lowering power and delay, and enabling high bandwidth links. However, waveguide based nanophotonic communication fabrics that use silicon microring resonators are yet to overcome many challenges for practical implementation even with recent promising developments. The key challenges for waveguide photonics include: *(i)* high complexity and overhead of thermally tuning microring resonators to ensure proper coupling of wavelengths, *(ii)* high power footprint due to significant waveguide crossing, propagation, and bending losses, *(iii)* need for complex tapered structures and optimized grating couplers with high coupling efficiency, and *(iv) 0.5-3 μm* inter-waveguide spacing requirements to avoid crosstalk that can lead to lower bandwidth density than in optimized electrical wires [4].

To overcome these challenges with waveguide photonics, free-space nanophotonics based on GaAs/AlAs dense Multiple Quantum Well (MQW) devices [5] have recently been proposed as an alternative. Such free-space configurations can be integrated with standard CMOS fabrication processes and are better suited for high-density optical interconnects due to their small active area and improved misalignment tolerance. MQW devices are projected to consume less than *1* pJ/bit energy and can be configured either as absorption modulators or photo-detectors (PDs). On-chip optical interconnects utilizing MQWs can operate at 40 Gbps bandwidth [6] to instantiate single-hop or multi-hop transfers through free-space optical links. MQW modulators provide significant potential to get around the thermal tuning challenges of silicon microring resonators and can be fabricated in various angles to achieve out-of-plane beam steering directions.

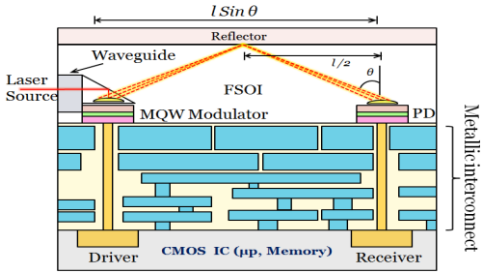

Figure 2: Conceptual view of CMOS integrated free-space on-chip optical link

Figure 1 summarizes the building blocks of free-space on-chip photonic interconnects (FSOI) with MQW modulators and PDs. It is also possible to utilize serializer/deserializer circuits to enable trade-offs between communication power and bandwidth. Figure 2 illustrates a FSOI with logic and photonics planes and vertical through silicon via (TSV) links providing interconnections between the silicon and photonics layers [5]. MQW devices are fabricated on a GaAs substrate and then flip-chip bonded to the logic layer and waveguide coupled with a continuous wave external laser source. The modulated light can be directed through micro-mirrors and micro-lens to transmit data via the free-space medium.

Significant recent research [7]-[15], [41]-[42] has focused on developing hybrid photonic NoC architectures that optimize local and global communication distribution between electrical and photonic links. Emerging SoCs with hundreds of cores in the near future will require customization of such communication fabrics to meet stringent application performance and chip power dissipation goals, with the help of

efficient system-level design tools. However, such system-level design tools for automated application-specific optimization of hybrid photonic NoCs do not exist. Synthesis tools and techniques proposed for application-specific electrical NoC fabrics [16]-[23] are not directly applicable for hybrid nanophotonic-electric NoCs due to significant differences in the underlying network fabric. In this paper, *we propose a novel system-level framework (HELIX) to synthesize application-specific hybrid NoC fabrics*. *HELIX* integrates graph based algorithms, linear programming, and custom heuristics to enable rapid design space exploration and application-specific customization of hybrid electro-photonic NoC fabrics for many-core chip architectures.

2. Related Work

In this section, we summarize prior work in the areas of *(i)* novel hybrid nanophotonic NoC architectures that address challenges of power and performance tradeoffs, and *(ii)* optimization of application-specific electrical NoCs.

A number of recent efforts have presented variants of hybrid nanophotonic NoC architectures. Pan et al. [7] proposed a hierarchical multi-plane photonic crossbar coupled with a concentrated mesh electrical network called Firefly. Firefly implemented R-SWMR (Reservation Assisted Single Write Multiple Read) optical links to reduce power consumption. Shacham et al. [8] proposed a reconfigurable broadband circuit-switched on-chip nanophotonic torus network with a topologically identical torus electrical network. Li et al. [9] presented a planar nanophotonic broadcast bus to transmit latency-critical messages and an electrical packet switched network that handles high bandwidth traffic. Vantrease et al. [10] and Joshi et al. [11] proposed fully photonic crossbars with electrical buffering. Morris et al. [12] developed a combination of an all-photonic crossbar and fat-tree NoC architectures. Bahirat et al. [15], [41]-[42] proposed a hybrid NoC fabric with concentric photonic rings coupled to a reconfigurable electrical mesh.

All of these efforts have utilized nanophotonic waveguides with microring resonators that have drawbacks as discussed earlier. Recently, Xue et al. [13] presented a novel intra-chip single-hop free-space nonophotonic interconnect based on VCSELs (vertical cavity surface emitting lasers) along with an algorithm to address challenges of free-space point-to-point optical link collision. Abousamra el al. [14] extended this work to create a two-hop free-space network that significantly reduced VCSELs required for the on-chip network. *The hybrid nanophotonic NoC we consider in this paper differs compared to previously proposed architectures by: (i)* utilizing CMOS compatible energy-efficient MQW modulators and detectors coupled to an external laser instead of bandwidth-limited and failure-prone VCSEL devices for on-chip FSOI; *(ii)* integrating a novel FSOI hybrid routing and flow control scheme that can be configured either for single or multi-hop communication; and *(iii)* incorporating a synthesizable FSOI collision detection and mitigation mechanism that can be dynamically configured through the serializer/deserializer modules.

Current research on application-specific NoC synthesis [16]-[19] has mainly focused on electrical NoCs. For instance, Murali et al. [16] presented a floorplan aware synthesis technique that considers wiring complexity of the NoC during topology synthesis along with min-cut partitioning to allocate switches to groups of custom cores and minimize NoC power. Chatha et al. [17] presented synthesis techniques for an application-specific NoC that employed integer linear programming (ILP) and min-cut/flow algorithms as well as node-weighted Steiner trees to obtain shortest paths. One limitation of these approaches is that they target single applications, which is increasingly impractical for today's multi-programmed workloads. Pasricha [18] presented a framework to synthesize a low power custom 3D NoC that satisfied thermal and TSV constraints. Other techniques have mainly focused on mapping uniform size cores and their communication flows on regular mesh topologies [20]-[24].

None of these synthesis approaches for regular or irregular NoCs has focused on synthesizing hybrid photonic NoCs. In this paper, for the first time, we address the synthesis of application-specific hybrid nanophotonic-electric NoCs with irregular topologies, using a comprehensive suite of techniques for core-to-tile mapping, floorplanning, and dual level router assignment; and support for architecture synthesis with multiple applications executing simultaneously.

3. Inputs and Problem Formulation

This section summarizes the inputs to our problem and formalizes our problem objective.

Application workload constraints

- Application *communication trace graph G(V,M,L)* for each application in a multi-application workload, where $v_i \in V$ is a set of processing cores, $m_i \in M$ a set of memory blocks, $l_i \in L$ a set of directed communication links;
- Application-specific communication bandwidth constraints $\omega_{i,j}$ in bits/cycle and latency constraints $\lambda_{i,j}$ in cycles between $\{v_i, v_j\}$ or $\{m_i, m_j\}$;

SoC platform constraints

- X_{max} and Y_{max} are the maximum dimensions of the die along the X and Y axes; and the aspect ratio X_{die}/Y_{die} of the synthesized die should be between 0.9 – 1.1 to obtain an approximately square die layout;
- Each network link is constrained by a maximum length γ that represents the maximum distance a signal can travel in a single cycle, based on CMOS process technology;

Problem Objective

- Synthesize a hybrid nanophotonic-electric application-specific NoC architecture $J(R, L_e, L_p, C)$ where R is a set of hybrid routers, L_e and L_p represents the set of electrical and photonic links, and C is a core-to-die mapping function; such that communication power is minimized while meeting bandwidth and latency constraints of the given application(s), and platform constraints of the SoC.

Configuration parameters

- Application task to core mapping;
- Layout of cores and memories on the planar die;
- Number and layout of hybrid electro-photonic and electrical-only routers that utilize a set of photonic $p_i \in P$ or electrical links $e_k \in E$ to support communication for a given multi-application workload;
- Size of photonic concentration region (PCR) that determines the cores/memories allowed to use each hybrid photonic router on the die (see Section 4 for details);
- Serialization degree D_n at electro-photonic interfaces;
- Hop count (1-hop or 2-hop) selection for FSOI links;

4. Hybrid Photonic NoC Architecture: Overview

To maximize performance in SoCs, ideally any two connected cores should communicate with each other using a point-to-point single hop network. For an $(m \times n)$ core SoC architecture, a single hop connectivity NoC fabric requires $O(m \times n)^2$ links. This is prohibitive to implement using a reasonable number of metal and photonic waveguide layers. However a free-space optical interconnect (FSOI) network can

eliminate much of the complexity of laying out multiple waveguides and also reduce global metal interconnect counts, while enabling 1-hop or 2-hop communication paths.

Our chosen FSOI network fabric utilizes micro-mirrors and reflectors, with light traversing through free-space to achieve 1-hop or 2-hop transfers with low overhead. For a 1-hop $(m \times n)$ SoC with flit width of k, each node needs $2\times k[(m \times n) - 1]/(MQW$ Gbps/(GHz CPU clock)) MQW devices, while a 2-hop $(m \times n)$ SoC with the same flit width needs $4\times k[(m + n) - 2]/(MQW$ Gbps/(CPU Clock)) MQW devices [14]. As an example, for a 12×12 core SoC with a 1-hop NoC, with flit width of 256 bits at 40 Gbps/link and a 3.88 GHz CPU clock, *7322* MQW devices are required. The photonic components for a *20mm×20mm* SoC die size will consume < *5 mm*^{2} on-chip area for a 1-hop FSOI-based NoC with *100µm* MQW devices. In contrast, a 2-hop NoC will require only *1128* MQW devices with < 1 mm^2 area and a 5.5× power reduction over a 1-hop NoC, but at the cost of system bandwidth drop from *300 to 45* Tbps. We explore hop-count selection on a per-communication flow basis to enable power-bandwidth trade-offs in our framework.

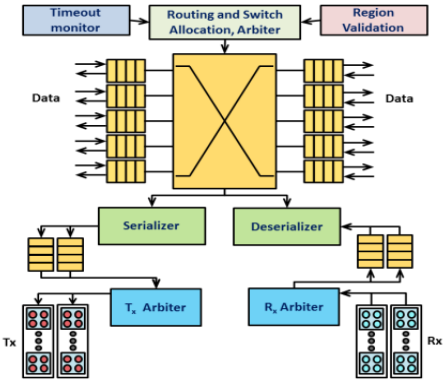

Figure 3. Gateway interface router architecture

We consider a SoC platform with a dedicated photonic layer that supports FSOI links, interfacing with an electrical NoC. Our electrical NoC is composed of two types of routers: *(i)* conventional four stage pipelined electrical routers that have *n* I/O ports and interface with local cores; and *(ii)* gateway interface routers (Figure 3) that are also four-stage pipelined but have additional photonic ports (a total of *n+2* I/O ports). The photonic link interface in gateway routers is responsible for sending/ receiving flits to/from photonic links in the photonic layer. Both types of routers have an input and output queued crossbar with a 4-flit buffer on each input/output port, with the exception of the photonic ports in gateway interface routers that use double buffering to cope more effectively with the higher photonic path throughput. The serializer/deserializer modules support serialization degrees of 2, 4, 8 and 16, enabling trade-offs between performance and freedom from conflicts for FSOI transfers.

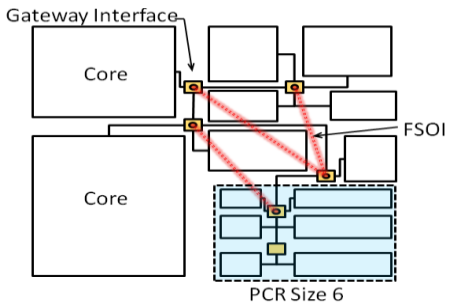

Figure 4. Photonic Concentration Region (PCR)

A unique feature of our hybrid NoC fabric is the reconfigurable traffic partitioning between electrical and photonic links. To minimize implementation cost, our synthesis framework limits the number of gateway interfaces. An adaptive photonic concentration region (PCR) ensures appropriate scaling and utilization with changing communication demands. A PCR is defined as the number of cores around the gateway interface that can utilize the FSOI path for communication (Figure 4). Cores within the same PCR communicate with each other via the electrical NoC (intra-PCR transfers). Cores that need to communicate and reside in different PCRs communicate using photonic paths (inter-PCR transfers). The electrical NoC transfers use XY routing, and a modified PCR-aware routing scheme for selective data transmission through the photonic links, with timeout-based regressive deadlock handling, based on the approach presented in [15].

Our arbitration approach is different from FSOI-based gateway interface routers proposed in [13] and [14] that utilize transfers without any arbitration. These routing schemes directly stream data to destination cores and manage collision of photonic data with a collision handling scheme (e.g., when multiple source nodes send data to the same destination core). But we observed that the performance benefits of eliminating arbitration are overshadowed by high penalties of collision handling and retransmission for high performance communication flows. Therefore in our gateway interface routers we implement support for reservation channels to reserve FSOI data paths. An additional input and output reservation channel port is added to routers for this purpose.

5. HELIX Synthesis Framework: Overview

In this section, we present our novel framework for synthesizing hybrid nanophotonic NoCs, which consists of the following steps as shown in Figure 5 *(i)* task-to-core mapping; *(ii)* floorplanning; *(iii)* Steiner tree based network formation; *(iv)* link clustering and dual level router mapping *(v)* PCR allocation *(vi)* conflict analysis and resolution; and *(vii)* validation with cycle-accurate simulation. Due to lack of space, here we briefly discuss each step. The following subsections provide an overview of these steps.

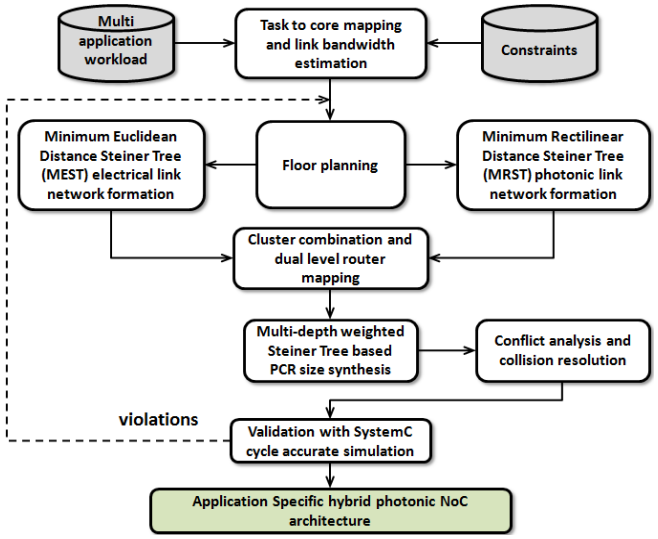

Figure 5. *HELIX* hybrid electro-photonic NoC synthesis flow

5.1 Task-to-core mapping

In this first step, we perform task-to-core mapping and link bandwidth estimation. The step involves mapping of *n* tasks to

m heterogeneous cores for the given application(s) task flow graph. We perform task execution-time estimation as well as estimation of inter-core data transfers using an instruction simulator [31]. We implement a genetic algorithm (GA) [27] to accomplish task to core mapping. The GA *chromosome* consists of possible mappings for the given application tasks to available cores, as well as a *virtual* link (electrical or FSOI) between cores that can satisfy bandwidth and latency constraints at a coarse granularity. The specific parameters used in the GA implementation are described in detail in the experimental setup section (6.1). The GA cost function represents overall communication power and the GA attempts to create a task-to-core mapping to minimize this power.

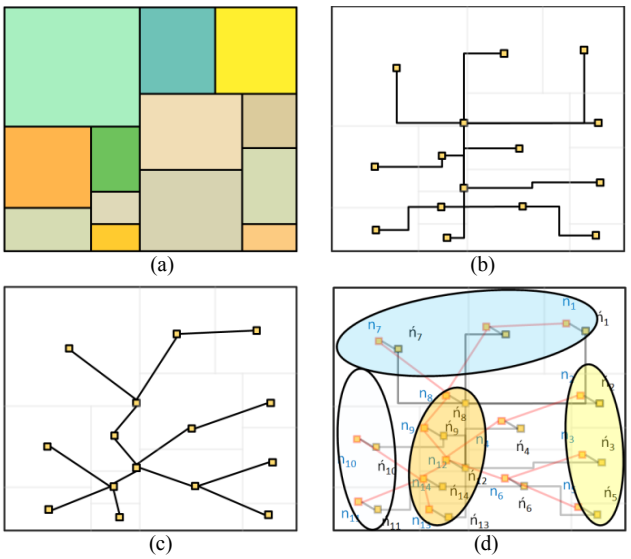

Figure 6. (a) output of floorplanner; (b) Minimum Euclidean Distance Steiner Tree (MEST) for electrical links; (c) Minimum Rectilinear Distance Steiner Tree (MRST) for FSOI links (d) Clustering and dual level router mapping

5.2 Floorplanning

For hybrid nanophotonic-electric NoC architectures, the floorplanning step is significantly more complex than traditional floorplanning, as the power consumption and delay of electrical wire and FSOI links differ significantly. To the best of our knowledge there is no floorplanning tool available that can support such hybrid FSOI and electrical wire based architectures. We therefore designed an enhanced system-level NoC floorplanning tool that uses mixed integer linear programming (MILP) to perform communication-aware and power-aware core placement on the die. Our MILP minimization objective function is a linear combination of the weighed power-latency and the overall chip area, representing the metrics that are optimized in this stage:

$$\left[\sum_{\forall\, c(u,v)\in E}^{i,j} l(u,v)\times Cp_{i,j}\right] \propto +[X_{max}+Y_{max}]\beta \qquad (1)$$

where, $l(u,v)\times Cp_{i,j}$ is weighed communication power and link distance between cores, $\propto$ and β are constants, and X_{max} and Y_{max} are the maximum allowed dimensions of the die along the X and Y axes. The floorplanner also integrates an aspect ratio constraint, to achieve an approximately square shaped floorplan. As FSOI links consume less power than electrical links, the floorplanner allows placing cores communicating via FSOI links farther apart than cores communicating via electrical links. Note also that at this stage, routers have yet to be allocated, and are assigned arbitrary locations with respect to cores (1 virtual router/core). The output of this step is a floorplan as shown in Figure 6(a).

5.3 MEST and MRST based NoC topology formation

In this step, we generate Minimum Rectilinear distance Steiner Trees (MRST) for the electrical network and Minimum Euclidian distance Steiner Trees (MEST) for the free space photonic network. We implemented these separate tree structures because electrical signal transmission occurs through rectilinear wires, and their Manhattan distance is best captured by an MRST; and free space photonic transfers can occur using non-rectilinear links, and their Euclidian distances are best captured by an MEST. Each link $l_{i,j}$ is given a weight that is a function of normalized communication bandwidth, power, and latency:

$$\alpha\times\psi_{i,j}\times\left[\omega_{i,j}/max\_bw\right]+(1-\alpha)\times\left[min\_latency/\lambda_{i,j}\right] \qquad (2)$$

where $\psi_{i,j}$, $\omega_{i,j}$, and $\lambda_{i,j}$ are link power consumption, link bandwidth, and link latency, respectively. We use separate values for the parameter α for FSOI links and electrical links due to their power consumption differences. The MRST structure for electrical links and MEST structure for FSOI links are constructed with the goal of minimizing the aggregate link weights, and an example of these structures is shown in Figures 6(b)-(c). Note again that the routers are still not accurately mapped on the die during this stage, and we approximate virtual router locations at the center of each core. At the end of this step, all cores are connected with FSOI and electrical links.

5.4 Clustering and dual level router mapping

The objective of the subsequent clustering step is to merge the communication links in the MEST and MRST solutions; map hybrid nanophotonic-electric and electrical-only routers such that router counts are minimized and utilization of links and routers is improved; and optimize FSOI links.

We create a *heuristic* that computes connection strength between each node pair based on inter-node link bandwidth and power characteristics. Then starting with no edges between any nodes, we add edges in order of decreasing connection strength to create clusters, as shown in Figure 6(d). The clusters are created utilizing a connection strength threshold such that intra-cluster short distance communication paths can be optimized utilizing electrical links and inter-cluster transmission can be performed using FSOI links. Each cluster represents a router in the final solution. But we still need to determine which communication flows will utilize FSOI links, electrical links, or a combination of both types of links. This problem is solved by using a *push-relabel maximum flow* algorithm. For every core n_i in the system, we create a corresponding pseudo core n_i^i, where inter-core communication for all n cores uses MEST links and all n'cores MRST links. The n and n'cores are linked with weights based on MRST and MEST links. Using the push-relabel maximum flow algorithm we generate a combined Steiner Tree and then merge the n and n'cores.

At the end of this stage, we utilize the *max-flow min cut algorithm* [18] to determine 1-hop or 2-hop routing for FSOI-based communication flows, to maximize bandwidth utilization while minimizing router resources. This process can also add or delete FSOI links as needed to meet any unsatisfied bandwidth or latency constraints; and tradeoff between performance and power requirements as discussed in Section 4.

5.5 PCR size synthesis

In this step, we perform post processing of the combined MEST/MRST and router mapping to develop PCR regions. The root nodes in the MEST that include multiple and multi-depth branches are considered for integration into a PCR region with the nearest gateway interface router. More specifically, PCR regions cover nodes that are directly connected to the root

nodes with *connection strength* lower than the links between the root nodes. For inter-PCR transfers, we set a size threshold M_{th} such that messages with size less than M_{th} transverse electrical links, while messages that exceed the threshold size travel to gateway interface routers and utilize FSOI links. Such a scheme ensures that small message size transfers do not encounter unnecessary E/O and O/E conversion delays, which would make their transfer over FSOI links less advantageous than over electrical links.

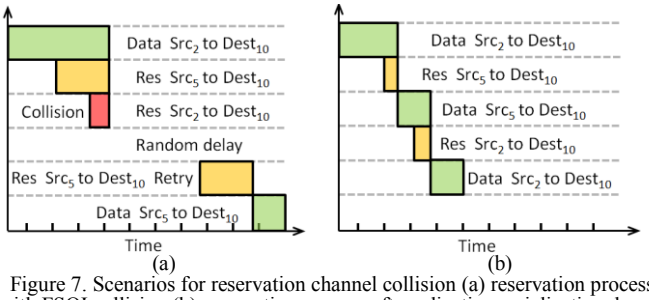

Figure 7. Scenarios for reservation channel collision (a) reservation process with FSOI collision (b) reservation process after adjusting serialization degree

5.6 Conflict analysis and resolution

To reduce collision probability within the FSOI pipelined reservation channel, this final step attempts to minimize interference between various FSOI transactions.

As we implement a pipelined router architecture with separate reservation channels, the reservation process can proceed while data transmission is in progress. Thus, more than one source core can attempt to reserve the same destination core, resulting in reservation collision (i.e., interference in modulated photonic links) at the destination node. This collision can produce erroneous data bits. Such collision can be detected using parity bits. In our architecture, transaction interference is avoided by managing link bandwidth via *modulation of the serialization degree*. Transactions from a source router (connected to the initiating core) to the sink router (connected to the target core) along each FSOI path are evaluated based on detailed communication schedules along a time-axis. In case of any conflicts between two transactions, we serialize these transactions such that both transactions can traverse the same router without interfering with each other.

Figure 7 summarizes this process. The channel reservation time is represented by the yellow colored horizontal bar. In the normal case when there is no collision, the reservation proceeds in parallel to data transmission. Once the reservation phase is complete, the next data transaction can begin. Figure 7(a) depicts a collision scenario with the red colored bar, where two reservation requests arrive in parallel with the first transaction's data transmission. This situation requires a reservation retry for the conflicting nodes after a specified retransmission delay, thus increasing latency. To eliminate this collision latency, our conflict analysis and resolution step utilizes serialization to modulate communication bandwidth such that multiple streams can coexist without collision. Figure 7(b) demonstrates how serialization can eliminate retransmission delays due to collision, thereby achieving overall lower transmission latency (at the cost of a slight increase in area and power due to the need for serialization/deserialization circuitry).

TABLE I MIBENCH APPLICATIONS FOR APPLICATION CATEGORIES

Application category	*Applications*
Industrial	basicmath, bitcount, qsort, susan
Consumer	jpeg, lame, mad, tiff2bw, tiff2rgba
Office	ghostscript, rsynth, stringsearch
Networking	dijkstra, patricia
Security	blowfish, rijndael, sha

6. Experiments

6.1 Experimental Setup

We synthesized application-specific hybrid NoC architectures for multi-application workloads derived from five MiBench [30] benchmark categories: *(i)* Automotive and Industrial Control, *(ii)* Consumer, *(iii)* Office Automation *(iv)* Networking, and *(v)* Security. As the MiBench benchmarks are written for a single processor, we created our own multithreaded implementation using Linux *pthreads*. To create multi-application workloads, we combined multiple MiBench applications executing in parallel, with execution priority assigned to each application in case of any contentions arising during accesses to memories or during task scheduling. We generated instruction and communication traces for the benchmarks via the Shade simulator [31]. Table 1 presents the 17 benchmarks across the five application categories that were considered. We implemented 5 multi-application workloads, corresponding to all applications available in each category, e.g., for the automotive and industrial control multi-application workload, we included parallel implementation of *(i)* basicmath, *(ii)* bitcount, *(iii)* qsort and *(iv)* susan benchmarks. In addition to MiBench benchmarks, we also evaluated our *HELIX* framework with PARSEC [40] application benchmark workloads. The Princeton Application Repository for Shared-Memory Computers (PARSEC) benchmark suite is composed of several multithreaded programs that represent next-generation shared-memory programs for SoCs.

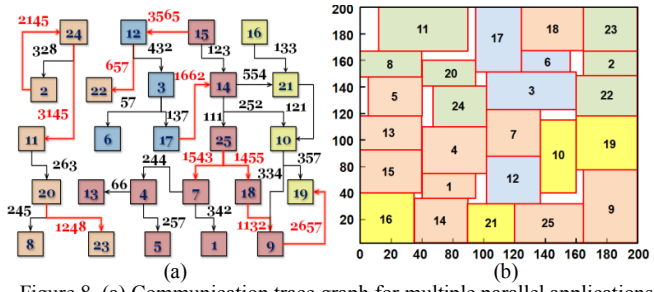

Figure 8. (a) Communication trace graph for multiple parallel applications, nanophotonic links in red color (b) Custom layout with irregular topology.

For our core mapping GA, our initial population size included 2000 randomly generated application mappings. We evaluated a range of crossover mutation and probabilities and ultimately utilized probability values of *0.34* and *0.42* respectively. The best fitness value *chromosome* in each iteration which resulted in minimum power consumption while meeting performance constraints was cached to prevent being overwritten by a non-dominated solution chromosome. As the GA is a stochastic search algorithm, it is not possible to formally specify convergence criteria based on optimality, therefore we terminated our GA when the best solution quality did not change over a predefined number of iterations (2000).

Figure 8(a) shows the enhanced communication trace graph (CTG) of 4 industrial applications running in parallel, which is generated after running the GA algorithm. Vertices in the CTG represent cores on which tasks have been mapped. Note the initial link selection that allocates some flows to electrical links and others to FSOI links. Communication flows with stringent bandwidth and/or latency demands generally get mapped to the more efficient FSOI links in this first step. But note that this initial assignment can be modified as the solution is refined in the later steps of the *HELIX* design flow.

Figure 8(b) shows the floorplanning solution for this CTG graph, where cores of each application are depicted by a separate color. During floorplanning, we set weighed communication power constant $\propto$ and link distance constant β

values at 0.5 each based on experimental analysis. We utilized a public domain GeoSteiner3.1 Steiner Tree solver [33] to generate the MEST and MRST networks and utilized the lp_solve optimizer [34] to solve the Mixed Integer Linear Programming (MILP). We set weight values for $\propto$ to *0.46* and *0.68* respectively, during MEST generation for electrical links and MRST generation for FSOI links. We created clusters utilizing a *0.38* connection strength threshold to optimize electrical intra-cluster short distance communication and inter-cluster transmission using FSOI links. Based on experimental analysis, we set a normalized M_{th} threshold of *0.33* in PCR regions such that communication messages less than the size of M_{th} transverse through electrical links and the messages that exceed the size of the threshold travel through FSOI links. We combined the various components of our *HELIX* framework using a python scripting interface [35].

The static and dynamic power consumption of electrical routers as well as the power consumption for optimally sized repeated Cu wires is obtained from a modified version of the Orion 2.0 simulator [36]. Our synthesis process targeted the *32 nm* node technology and utilized a *400 mm*^{2} SoC die area. The delay of an optimally repeated and sized electrical (Cu) wire at 32 nm was assumed to be *42 ps/mm*. Photonic free space communication delay is *3.3456 ps/mm* (compared to photonic waveguide delay of *15.4 ps/mm* [11]) requiring much less buffering during transfers compared to traditional electrical or waveguide based hybrid photonic NoCs. The intrinsic speed of a MQW is practically limited by the driver electronics and the well-known quantum-confined Stark effect working at sub-picosecond time scales. We modeled a *1*μm thick *5V* modulator with *10*$\times$*10*μm^2 area and capacitance of *11 fF*, calculating the per cycle electrical energy of the device as *140fJ* which is in line with prior estimates [37]. Our implementation assumed modulator driver delay of *9.5 ps*, modulator delay of *3.1 ps*, photo detector delay of 0.*22 ps* and receiver delay of *4.9 ps* [15]. Our hybrid NoC with an irregular topology was modeled at the cycle accurate granularity by extensively modifying our in-house cycle accurate SystemC-based [38] NoC simulator derived from the Noxim [39] simulator.

As no prior published work exists on synthesizing custom application-specific hybrid nanophotonic-electric NoCs for comparison, we compared our synthesis results with respect to application-specific NoC synthesis frameworks in [16] and [18] that synthesize purely electrical NoCs. We implemented the floorplan-aware design process in [16] that accounts for wiring complexity and detects timing violations on the NoC links early in the design cycle. This algorithm was implemented in two phases. Within the first phase, we selected a topology that best optimizes user objectives satisfying all design constraints, and in the second phase we varied a number of design parameters such as NoC clock frequency and link width to find a solution that best optimized all design constraints. Similarly, we implemented the two-stage synthesis methodology as presented in [18] which consists of core to router mapping and custom topology and route generation.

We evaluated various SoC complexities during our experimental analysis to better understand the impact and scalability of our *HELIX* synthesis framework for small (25 cores), medium (64 cores) and large (144 cores) sized SoCs, when compared to the frameworks in [16] and [18].

6.2 Experimental Results

This section analyzes the hybrid nanophotonic-electric NoC designs synthesized by our *HELIX* framework for the various multi-applications workloads. The results of synthesis for the 25, 64, and 144 node complexity SoC platforms are shown in Figure 9, for the five multi-application MiBench workloads. Our *HELIX* synthesis framework provides on average 2.82×, 3.12× and 3.49× reduction in power for the 25, 64, and 144 core SoC platforms respectively compared to application specific electrical NoC utilizing approaches from [16] and [18]. This improvement in power dissipation with *HELIX* relative to [16] and [18] is a result of: *(i)* congestion reduction in the electrical links due to offloading of a large portion of the global communication to FSOI links; *(ii)* reduction in electrical link switching activity; *(iii)* shorter link lengths; and *(vi)* smaller buffer resources compared to electrical-only application-specific NoC architectures synthesized by [16] and [18]. Due to the use of fast and high bandwidth FSOI links as well as reduced congestion in the electrical NoC, the communication latency improved with *HELIX by 1.18×, 1.23× and 1.25×* and throughput by *1.68×, 1.69× and 1.78×* for the 25, 64 and 144 core SoC platforms.

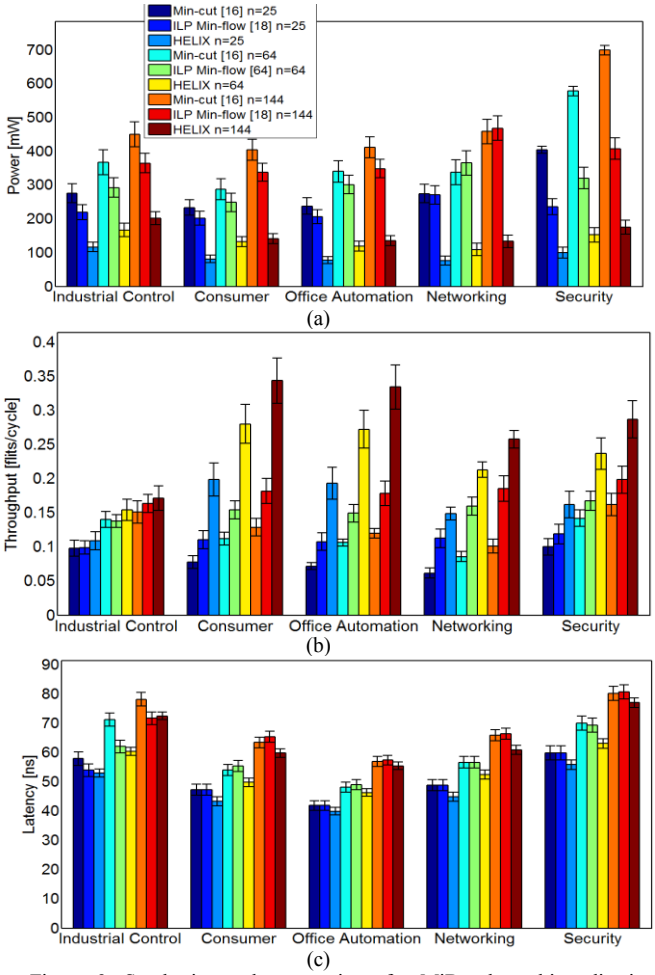

Figure 9. Synthesis result comparison for MiBench multi-application workloads (a) average power (b) throughput (c) average latency

We observed that *HELIX* was able to achieve a significant reduction in the number of gateway interface routers through clustering and dual level router mapping, with as few as 33% gateway interface routers compared to the router count before clustering. The dual level router mapping step also adds paths enabling inter-cluster long distance global communication using 2-hop FSOI links to minimize power. These additions allow the electrical NoC router count and complexity to be reduced compared to results obtained from [16] and [18]. We also observed that the conflict analysis and resolution step in *HELIX* reduced MQW modulator and detector counts by

approximately 50% by intelligent management of serialization degrees at the nanophotonic-electric interfaces.

Our *HELIX* synthesis framework was able to achieve a viable solution for all application workloads and SoC complexities that we evaluated. Each stage in our synthesis framework worked seamlessly, complementing each other to balance conflicting requirements to solve the nontrivial problem of synthesizing application-specific hybrid free-space photonic-electric NoC fabrics. Table 2 summarizes key synthesis parameters for the 25, 64 and 144 SoC sizes, for the MiBench multi-application workloads. It is interesting to observe that the number of photonic free-space gateway interfaces is significantly lower than the number of electrical routers, and accounts for only 20% of all routers. This is in contrast with previously proposed [7]-[12] nanophotonic architectures. The number of clusters and gateway interfaces correlates well with each other and by judiciously selecting FSOI hop counts, the framework minimizes area and the number of modulators and photodetectors required, without violating performance constraints.

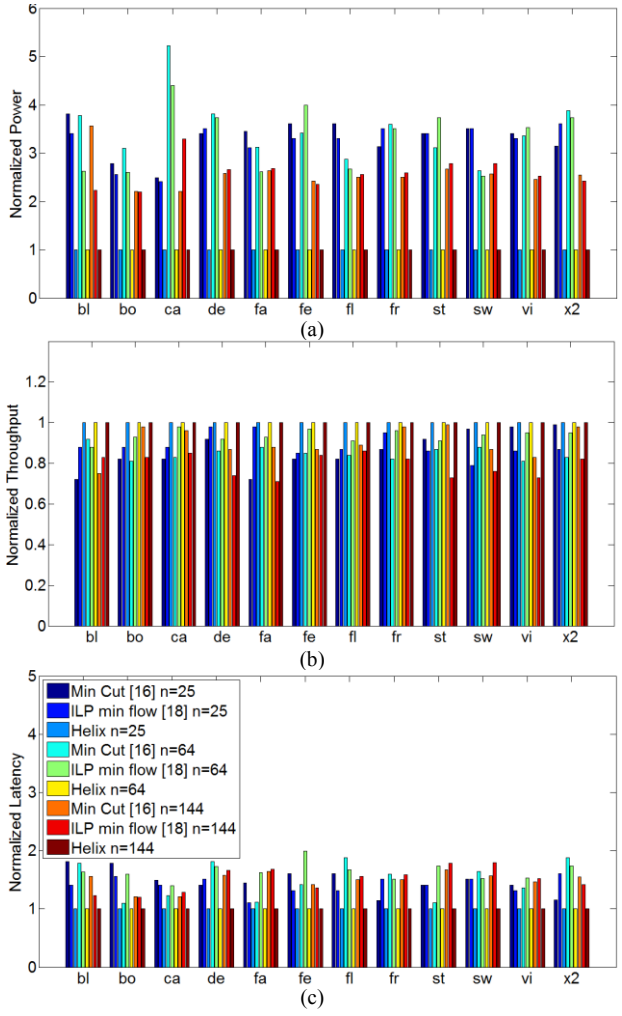

Figure 10. Synthesis result comparison for PARSEC multi-threaded workloads (a) average power (b) throughput (c) average latency

HELIX is also able to reduce the average number of hops in the electrical network by up to 4×, electrical link area by up to 1.24×, and link lengths for the electrical network by up to 2.67× compared to previously published electrical NoC synthesis techniques in [16] and [18]. This improvement is possible due to the *HELIX* floorplanner placing cores communicating via FSOI links farther apart and the cores communicating via electrical links closer, achieving two fold benefits by replacing long distance electrical links with more efficient FSOI links and placing cores closer that communicate with electrical links.

TABLE 2. COMPARISON OF SYNTHESIS PARAMETERS

P =Average Power improvement compared to [16] and [18], PR = Number of photonic routers, C = Number of clusters, PR=Max PCR Size, EH=Max Electrical Hop Count, PH=Max Photonic Hop Count, SD=Serialization Degree

Application	***P[16]***	***P[18]***	***PR***	***C***	***PR***	***EH***	***PH***	***SD***
25 core NoC								
Industrial	2.363	1.883	7	7	4	4	2	2
Consumer	2.880	2.491	6	6	3	3	2	3
Office	3.061	2.663	8	8	3	4	2	3
Networking	3.603	3.551	8	6	4	3	1	4
Security	4.062	2.367	7	6	5	4	1	4
64 core NoC								
Industrial	2.198	1.751	16	15	4	4	2	3
Consumer	2.175	1.879	17	17	5	4	2	3
Office	2.867	2.532	16	16	5	4	2	4
Networking	3.098	3.352	18	16	4	4	2	3
Security	3.796	2.104	18	16	4	4	2	4
144 core NoC								
Industrial	2.226	1.806	28	25	6	4	2	3
Consumer	2.847	2.377	25	23	7	5	2	4
Office	3.044	2.577	26	24	6	5	2	3
Networking	3.437	3.519	25	28	6	4	2	4
Security	3.993	2.327	26	25	7	4	2	4

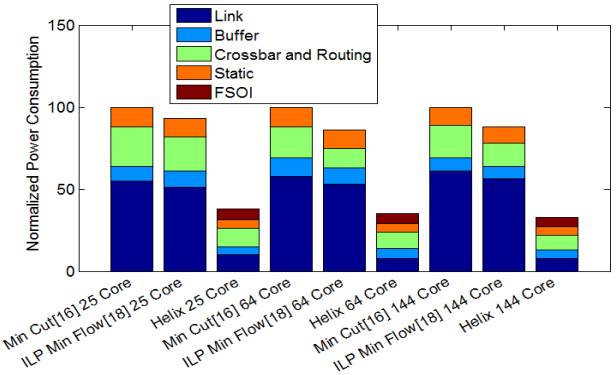

Figure 11. Normalized breakdown of power consumption

The results of synthesis for the 25, 64, and 144 node complexity SoC platforms are shown in Figure 10, for twelve multi-threaded PARSEC benchmarks (blackscholes (bl), bodytrack (bo), canneal (ca), dedup (de), facesim (fa), ferret (fe), fluidanimate (fl), freqmine (fr), streamcluster (st), swaptions (sw), vips (vi), x264 (x2)). Once again, for the *25, 64* and *144* core SoCs, *HELIX* achieves a notable power dissipation improvement of *3.28×, 3.40×, 2.58×* respectively, compared to the results obtained from the synthesis frameworks in [16] and [18], as well as improvements in throughput by *1.11×, 1.14×, and 1.16×* and average transfer latency *by 1.44×, 1.56×, and 1.49×* respectively.

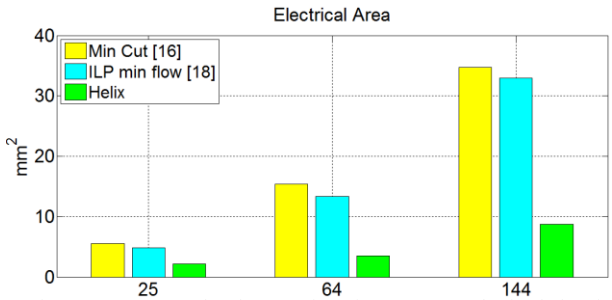

Figure 12. Area overhead comparison for *HELIX* synthesized electrical network for PARSEC application benchmarks

The breakdown of normalized power consumption in Figure 11 for the PARSEC benchmarks demonstrates how *HELIX* can effectively improve power consumption in all categories by managing nontrivial trade-offs during the synthesis process, balancing transfers across electrical and photonic planes to provide superior results. The lower buffer power can be attributed to lower latency in free space paths, allowing for routers with less buffer space within the electrical network. Link power consumption accounts for majority of improvements due to the free space photonic path utilization consuming lower power that also reflects in lower electrical network area overhead as shown in Figure 12.

7. Conclusion

In this paper, we presented the *HELIX* framework to *synthesize application specific hybrid nanophotonic-electric NoCs with irregular topologies*. To the best of our knowledge this problem has not been address before in any prior work. Based on our experimental studies, we demonstrate that the proposed techniques in the *HELIX* framework produce a superior NoC architecture that satisfies all performance requirements for *MiBench* multi-application workloads and *PARSEC* multi-threaded workloads, while achieving an average of *3.06*× reduction in power dissipation across SoC platforms of varying complexity, compared to previously proposed application-specific electrical-only NoC synthesis frameworks. By addressing the many challenges related to the overheads of microring resonators and photonic waveguide based architectures, we also propose a practical framework aimed at bringing hybrid nanophotonic-electric NoCs based on FSOI links and electrical links closer to reality.

Acknowledgements

This research is supported by grants from SRC, NSF (CCF-1252500, CCF-1302693), and AFOSR (FA9550-13-1-0110).

References

[1] ITRS Technology Working Group, http://public.itrs.net, 2012.
[2] S. Pasricha, and N. Dutt. "On-Chip Communication Architectures", Morgan Kauffman, ISBN 978-0-12-373892-9, Apr 2008.
[3] S. Pasricha and N. Dutt, "Trends in Emerging On-Chip Interconnect Technologies", IPSJ Transactions on System LSI Design Methodology, Vol. 1, Sep 2008.
[4] M. Haurylau, G. Chen, J. Zhang, N.A. Nelson, D.H. Albonesi, E.G. Friedman, P.M. Fauchet. "On-chip optical interconnect roadmap: Challenges and critical directions", Proc. IEEE J. Sel. Quantum Electronics, pp (6):1699-1705, 2006.
[5] R. Nair, T. Gu, K. W. Goossen, F. Kiamilev, M. W. Haney, "Demonstration of chip-scale optical interconnects based on the integration of polymer waveguides and multiple quantum well modulators on silicon", Proc. IEEE Photonics 2011 Conference.
[6] L. Chrostowski , B. Faraji , W. Hofmann , R. Shau , M. Ortsiefer and M. C. Amann, "40 GHz bandwidth and 64 GHz resonance frequency in injection-locked 1.55 μm VCSELs", Proc. Int. Semicond. Laser Conf., pp.117 -118, 2006.
[7] Y. Pan, P. Kumar, J. Kim, G. Memik, Y. Zhang, and A.Choudhary, "Firefly: Illuminating future network-on-chip with nanophotonics," Proc. ISCA, pp. 429–440, 2009.
[8] A. Shacham, K. Bergman, and L.P. Carloni, "The Case for Low-Power Photonic Networks on Chip," Proc. DAC, pp. 132-135, 2007.
[9] Z. Li, et al., "Spectrum: a hybrid nanophotonic-electric on-chip network," Proc. DAC 2009: 575-580
[10] D. Vantrease et al., "Corona: System Implications of Emerging Nanophotonic Technology," Proc. ISCA, pp. 153-164, 2008.
[11] A. Joshi et al., "Silicon-Photonic Clos Networks for Global On-Chip Communication", Proc. NOCS 2009.
[12] R.W. Morris, A.K. Kodi, "Power-Efficient and High-Performance Multi-level Hybrid Nanophotonic Interconnect for Multicores," Proc. NOCS, pp.207-214, 2010
[13] J. Xue, A. Garg, B. Ciftcioglu, J. Hu, S. Wang, I. Savidis, M. Jain, R. Berman, P. Liu, M. C. Huang, H. Wu, E. G. Friedman, G. Wicks, and D. Moore, "An intra-chip free-space optical interconnect", Proc. of ISCA, pp 94–105, 2010
[14] A. Abousamra, R.Melhem, R., A. Jones, "Two-hop Free-space based optical interconnects for chip multiprocessors", Proc NoCS, pp 89-96, 2011.
[15] S. Bahirat, S. Pasricha, "Exploring Hybrid Photonic Networks-on-Chip for Emerging Chip Multiprocessors", Proc. CODES+ISSS, 2009.
[16] S. Murali, P. Meloni, F. Angiolini, D. Atienza, S. Carta, L. Benini, G. De Micheli and L. Raffo, "Designing application-specific networks on chips with floorplan information", Proc. ICCAD 2006.
[17] K. Chatha, K. Srinivasan, G. Konjevod, "Automated Techniques for Synthesis of Application-Specific Network-on-Chip Architectures", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 27, Aug. 2008, pp. 1425-1438.
[18] S. Pasricha, "A Framework for TSV Serialization-aware Synthesis of Application Specific 3D Networks-on-Chip", IEEE International Conference on VLSI Design (VLSID 2012), Jan. 2012.
[19] S. Kwon, S. Pasricha and J. Cho, "POSEIDON: A framework for application-specific network-on-chip synthesis for heterogeneous chip multiprocessors", Proc. IEEE ISQED, 2011.
[20] S. Murali, G. D. Micheli, "Bandwidth-Constrained Mapping of Cores onto NoC Architectures," Proc. DATE, 2004.
[21] G. Ascia, V. Catania, and M. Palesi. "Multi-objective Mapping for Mesh-based NoC Architectures", Proc. CODES+ISSS 2004.
[22] J. Hu, R. Marculescu, "Energy-Aware Mapping for Tile-based NoC Architectures under Performance Constraints," Proc. ASPDAC 2003.
[23] N. Kapadia, S. Pasricha, "VISION: a framework for voltage island aware synthesis of interconnection networks-on-chip", Proc. Great Lakes Symposium on VLSI, 2011, pp. 31-36.
[24] S. Bahirat, S. Pasricha, "A Particle Swarm Optimization Approach for Synthesizing Application-specific Hybrid Photonic Networks-on-Chip", Proc. IEEE ISQED 2012.
[25] G. Tian, R. Nair, M. Haney, "Integrated free-space optical interconnects: All optical communications on- and off-chip", Proc. IEEE Optical Interconnects Conference, pp 74-75, 2012
[26] M. Tagel, P. Ellervee, T. Hollstein, G. Jervan, "Communication modelling and synthesis for noc-based systems with real-time constraints", Proc. IEEE Symposium on Design and Diagnostics of Electronic Circuits and Systems, pp. 237-242, Apr 2011.
[27] D. Goldberg, "Genetic algorithms in search, optimization and machine learning", Kluwer Academic Publishers, Boston, MA, 1989
[28] K. Srinivasan, K. S. Chatha and G. Konjevod, "Linear Programming Based Techniques for Synthesis of Network-on-Chip Architectures", Proc. IEEE ICCD, pp. 422-429, October 2004.
[29] S. Yan and B. Lin, "Application-specific network-on-chip architecture synthesis based on set partitions and Steiner trees", Proc. ASPDAC, pp.277 -282, 2008
[30] M. Guthaus, J. Ringenberg, D. Ernst, T. Austin, T. Mudge, and R. Brown, "MiBench: A Free, Commercially Representative Embedded Benchmark Suite", IEEE 4th Annual Workshop on Workload Characterization, Dec. 2001.
[31] B. Cmelik and D. Keppel, "Shade: A fast instruction-set simulator for execution profiling", Proc. ACM SIGMETRICS Conference on Measurement and Modeling of Comp. Systems, pp 128–137, 1994.
[32] S.N. Adya, I.L. Markov, "Fixed-outline floorplanning: Enabling hierarchical design", IEEE TVLSI, 11(6):1120-1135, Dec.2003.
[33] http://www.diku.dk/geosteiner/
[34] http://lpsolve.sourceforge.net/5.5/
[35] http://www.python.org/
[36] A. Kahng, et al., "ORION 2.0: A Fast and Accurate NoC Power and Area Model for Early-Stage Design Space Exploration" Proc. DATE, 2009.
[37] D.A.B. Miller, "Energy consumption in optical modulators for interconnects," Opt. Express 20, pp A293-A308, 2012.
[38] SystemC initiative www.systemc.org.
[39] Noxim simulator http://noxim.sourceforge.net/
[40] C. Bienia, S. Kumar, J. Singh and K. Li. 2008, "The PARSEC benchmark suite: characterization and architectural implications", Proceedings of the 17th International Conference on Parallel Architectures and Compilation Techniques (PACT), Oct 2008.
[41] S. Bahirat, S. Pasricha, "UC-PHOTON: A Novel Hybrid Photonic Network-on-Chip for Multiple Use-Case Applications", PROC. IEEE ISQED, 2010
[42] S. Bahirat, S. Pasricha, "METEOR: Hybrid Photonic Ring-Mesh Network-on-Chip for Multicore Architectures", accepted for publication, IEEE Transactions on Embedded Computing Systems (TECS), 2014.