3D-HELIX: Design and Synthesis of Hybrid Nanophotonic Application-Specific 3D Network-On-Chip Architectures

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

Abstract— *With the advent of 3D chip stacking technology, application-specific and heterogeneous three dimensional chip multi-processors (3D CMPs) are projected to become the building blocks of future parallel processing systems. In such 3D CMPs, network-on-chip (NoC) architectures will enable communication between multiple heterogeneous cores. However, NoCs face several challenges, including limited bandwidth, high latency, and high power dissipation. Hybrid nanophotonic-electric NoCs are being considered as a solution to address the above challenges due to their desirable performance and power characteristics. These emerging communication architectures require substantial optimization to realize their full potential. Optimizing hybrid nanophotonic-electric 3D NoCs requires intelligent traversal through a massive design space, which is non-trivial. No prior work has addressed the problem of synthesizing and optimizing application-specific hybrid nanophotonic-electric 3D NoCs with an irregular topology to connect heterogeneous cores on a 3D CMP. Considering the above unaddressed major challenge, in this paper we propose a synthesis framework called 3D-HELIX that can optimize application-specific hybrid nanophotonic-electric 3D NoCs. Our target communication fabric combines electrical 3D NoCs with an irregular topology and a single layer of free-space nanophotonics. Based on our experimental studies, we demonstrate that 3D-HELIX can produce superior NoC architectures achieving up to 6× power savings while meeting application performance constraints, compared to 3D electrical NoC-based communication fabrics.*

1. Introduction

With the push towards integrating more and more cores on a die to enhance parallel processing capabilities, 3D integration has emerged as an interesting way to achieve high core densities on a chip. Wafer-to-wafer bonded 3D integrated circuits (3D-ICs) place active devices (processors, memories) within multiple active layers and vertical Through Silicon Vias (TSVs) connect cores across the stacked layers. Multiple active layers in 3D-ICs can enable increased integration of cores within the same area footprint as traditional single layer 2D-ICs. In addition, long global interconnects between cores can be replaced by shorter inter-layer TSVs, improving performance and reducing on-chip power dissipation. Recent 3D-IC test chips from IBM [1], Tezzaron [2] and Intel [3] have confirmed the benefits of 3D-IC technology.

Electrical network-on-chip (NoC) communication fabrics are commonly used in 2D processing chip architectures to connect various heterogeneous cores together. These fabrics are however severely constrained due to their long multi-hop latencies, low bandwidth density, and high power dissipation [4]. In 3D-ICs that utilize 3D NoC fabrics, the fundamental power, delay, and noise susceptibility limitations of traditional copper (Cu) interconnects are still severe. To overcome these limitations, alternative interconnect materials are needed. Photonic interconnects [5] represent one promising emerging solution that can replace Cu interconnects on a chip and help overcome their latency, bandwidth, and power bottlenecks. Photonic interconnects can transfer data with much more energy efficiency than Cu interconnects especially over long distances across a chip. Thus on-chip photonic interconnects are being actively explored as a promising alternative to Cu interconnects for global communication, allowing data to be transferred across a chip at a much faster light speed and with power dissipation that is independent of link length [6].

The most commonly proposed implementation of nanophotonic links requires waveguides to guide light and silicon microring resonators to modulate and filter out light from the waveguides. Such waveguide-based implementations face several challenges: *(i)* the coupling efficiency of microring resonators is sensitive to changes in temperature. This requires high complexity and high power overhead to thermally tune the resonator 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) 0.5~3.0 μm* inter-waveguide spacing requirements to avoid crosstalk that can lead to lower bandwidth density than in optimized Cu wires [7]. Because of these challenges, productization of waveguide-based nanophotonic NoCs has yet to become commercially viable.

In this paper, we consider free-space nanophotonics as an alternative to waveguide-based photonic interconnects. Our main contribution is a framework called *3D-HELIX* for synthesizing and optimizing hybrid nanophotonic-electric 3D NoC architectures that combines electrical NoCs with free-space nanophotonic interconnect (FSNPI) NoCs. Based on our experimental studies, we demonstrate that the presented algorithms in this paper produce superior 3D NoC architectures when compared to traditional 3D electrical NoCs.

2. Related Work

Significant recent research has focused on developing hybrid nanophotonic-electric NoC architectures that optimize local and global communication distribution between electrical and photonic links for next generation CMPs [8]. In this section, we discuss prior work in the areas of *(i)* hybrid nanophotonic-electric NoC architectures, and *(ii)* optimization of 2D and 3D heterogeneous NoC architectures.

2.1 Hybrid Nanophotonic-electric NoC architectures

A number of recent efforts have presented variants of hybrid nanophotonic-electric NoC architectures that use waveguide-based photonic links. Pan et al. [9] proposed a hierarchical multi-plane photonic crossbar coupled with a concentrated mesh electrical NoC called Firefly. Firefly implemented R-SWMR (Reservation Assisted Single Write Multiple Read) optical links to reduce power consumption. Shacham et al. [10] proposed a reconfigurable broadband circuit-switched on-chip nanophotonic torus network with a topologically identical torus electrical network. The electrical network worked in conjunction with the photonic network to reserve photonic channels. Li et al. [11] presented a planar nanophotonic broadcast bus to transmit latency-critical messages and an electrical packet switched NoC that handles high bandwidth traffic. Vantrease et al. [12] and Joshi et al. [13] proposed fully photonic crossbars with electrical buffering. Morris et al. [14] developed a combination of an all-photonic crossbar and fat-tree NoC architectures. Bahirat et al. [15] presented a hybrid NoC fabric with concentric photonic rings coupled to a reconfigurable electrical mesh NoC architecture.

All of these efforts have utilized nanophotonic waveguides with microring resonators that have drawbacks as discussed in Section 1. Recently, Xue et al. [16] presented a novel intra-chip single-hop free-space nanophotonic 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. [17] extended the work from [16] to create a two-hop free-space network that significantly reduced VCSELs required for the on-chip network. In this paper, we also consider free-space nanophotonic links, but our architecture differs compared to previously proposed architectures by *(i)* utilizing CMOS compatible energy-efficient MQW modulators [18] and detectors coupled to an external laser instead of bandwidth-limited and failure-prone VCSEL devices for on-chip FSNPI; *(ii)* integrating a novel FSNPI hybrid routing and flow control scheme that can be configured either for single or multi-hop communication; and *(iii)* incorporating a synthesizable FSNPI collision detection and mitigation mechanism that can be dynamically configured through the serializer/ deserializer modules.

2.2 Optimization of 2D and 3D NoC Architectures

Current research on heterogeneous 2D NoC synthesis [19]-[21] has mainly focused on electrical NoCs. For instance, Murali et al. [19]

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. Srinivasan et al. [20] presented a low complexity genetic algorithm based approach to synthesize a low power custom NoC topology. Chatha et al. [21] 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. Other techniques aimed at regular NoC synthesis have mainly focused on mapping uniform size cores and their communication flows on regular mesh topologies [22]-[25]. Our recent work [26] was the first to propose a framework for synthesizing regular hybrid nanophotonic-electric NoC fabrics using particle swarm optimization (PSO) based design space traversal and optimization techniques.

A few efforts have explored techniques for synthesizing 3D NoCs. Huang et al. [27] propose a NoC topology-aware flooplanning methodology that adjusts the positions of cores and routes to optimize the 3D NoC topology during the floorplanning phase using simulated annealing. Kapadia et al. [28] proposed a simulated annealing based framework for the co-synthesis of Voltage Island based 3D electrical mesh NoCs, using linear programming (LP) to evaluate the IR-drop distribution. Sepulveda et al. [29] proposed hybrid-on-chip communication structures (HoCs) for 3D-ICs that combined TSV buses and 2D NoCs. Rahmani et al. [30] proposed a congestion-aware adaptive routing algorithm for this architecture. Pasricha [31] proposed a *MORPHEUS* framework for the automated synthesis of application-specific 3D NoCs that combines techniques for thermal-aware core layout, TSV serialization, allocation of routers, NIs, and TSVs, and deadlock-free path generation with the goal of generating low power solutions that satisfy all application performance constraints.

None of these existing NoC synthesis approaches has focused on synthesizing 3D hybrid nanophotonic-electric NoC architectures. In this paper we address the synthesis of heterogeneous application-specific 3D hybrid nanophotonic-electric NoC architectures, 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. *3D-HELIX* is an extension of our recent work *HELIX [32]*. The HELIX framework enables rapid design space exploration and application-specific customization of 2D hybrid nanophotonic-electric NoCs that utilize free-space photonics, for heterogeneous many-core chip architectures.

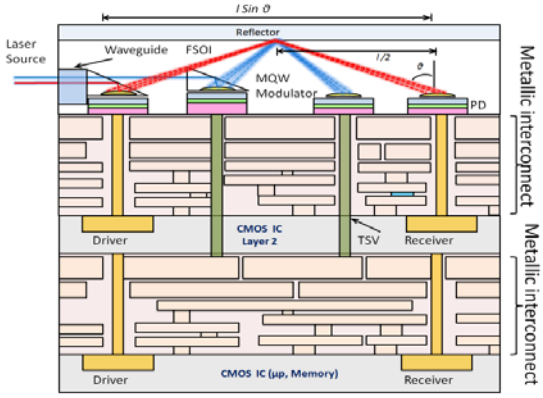


Fig 1: Conceptual view of 3D CMOS IC with logic and FSNPI layers

3. Background: FSNPI Architecture

To overcome challenges with waveguides and silicon microring resonators, on-chip free-space nanophotonic interconnects (FSNPIs) have recently been proposed[16][17]. Dense Multiple Quantum Well (MQW) devices are used for electro-optic modulation, consuming less than *1* pJ/bit energy. These MQW devices can be configured either as absorption modulators or photo-detectors (PDs). On-chip optical interconnects utilizing MQWs can operate at 40 Gbps bandwidth [33] to instantiate single-hop point-to-point or multi-hop transfers through free-space optical links. Most interestingly, MQW modulators do not suffer from thermal tuning challenges of silicon microring resonators and can be fabricated in various angles to achieve out-of-plane beam steering directions. 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. Fig. 1 illustrates a free space nanophotonic interconnect (FSNPI) configuration with two logic and one nanophotonic planes, and vertical through silicon via (TSV) links interconnecting the silicon and photonic layers [18]. 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. Modulated light can be directed through micro-mirrors and micro-lens to transmit data via the free-space medium.

Fig. 2*(a)* summarizes the building blocks of FSNPIs with MQW modulators and PDs, and its system level integration with an electrical NoC fabric. Serializer/deserializer circuits enable trade-offs between communication power, area and bandwidth by reducing photonic components through higher serialization degree. A unique feature of our hybrid nanophotonic-electric NoC fabric shown in Fig. 2*(b)* is the reconfigurable traffic partitioning between electrical and photonic links. To minimize implementation costs, our synthesis framework limits the number of gateway interfaces between the electrical and photonic layers. 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 FSNPI path for communication. 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 XYZ routing, and a modified PCR-aware routing scheme for selective data transmission through the FSNPI links, with timeout-based regressive deadlock handling, based on the approach presented in [13].

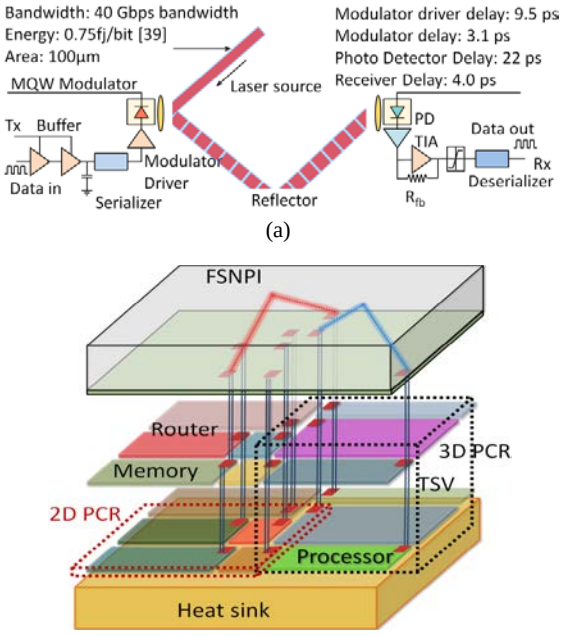


Fig 2: Building blocks of free-space on-chip photonic interconnects: (a) modulator and receiver circuit (b) 3D integration of electrical and FSNPI layer interconnect including Photonic Concentration Region (PCR)

As shown in Fig. 2(b), we consider a heterogeneous CMP platform with a dedicated photonic layer that supports FSNPI 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)* hybrid gateway interface routers (Fig. 3) that are also four-stage pipelined but have additional photonic ports (a total of *n+3* 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 can support serialization degree of 2, 4, 8 and 16, and allow for very high optical I/O pad density. The resulting high-density 2D array of surface-normal optoelectronic MQW devices can provide the necessary intra chip bandwidth density without the complexity of wavelength division multiplexing (WDM) [34].

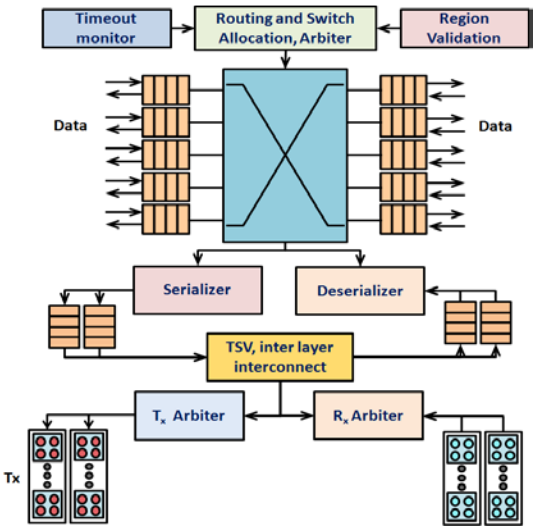


Fig 3. 3D Gateway interface FSNPI router architecture

Our arbitration approach is different from FSNPI-based gateway interface routers proposed in [16] and [17] 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 implemented support for reservation channels to reserve FSNPI data paths. An additional input and output reservation channel port is added to the routers for this purpose.

Finally, our FSNPI architecture supports multiple hops to balance performance and power goals. For a 1-hop $(m \times n \times l)$ path with flit width of k, each node on the 3D CMP needs $2{\times}k[(m \times n \times l) - 1]/(MWQ$ Gbps/(GHz CPU clock)) MQW devices, while a 2-hop $(m \times n \times l)$ path with the same flit width needs $4{\times}k[(m + n + l) - 2]/(MWQ$ Gbps/(CPU Clock)) MQW devices [17]. As an example, for a 3D CMP with 128 cores and 2 active layers, a 1-hop FSNPI based hybrid 3D NoC with flit width of 256 bits at 40 Gbps/link and a 3.88 GHz CPU clock requires *6308* MQW devices. These photonic components for a *20mm×20mm* CMP die size will consume < *4.2* mm^2 on-chip area for a 1-hop FSNPI-based NoC with $100\mu m$ MQW devices. In contrast, a 2-hop FSNPI based hybrid 3D NoC will require only *1590* MQW devices with < 1 mm^2 area and a 5.0× 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 3D-HELIX synthesis framework.

4. Problem Formulation

This section presents an overview of our problem formulation for synthesizing application-specific heterogeneous 3D hybrid nanophotonic-electric NoC architectures with *3D-HELIX*:

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}, Y_{max} and Z_{max} are the maximum dimensions of the die along the X, Y and Z 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;
- Single layer FSNPI network constrained by on-chip area and communication requirements;

Problem Objective

- Synthesize a 3D hybrid nanophotonic-electric application-specific heterogeneous NoC architecture *J(R, L_e, L_p, C)* where *R* is a set of hybrid (photonic-electric gateway interface) 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;
- Mapping and layout of cores and memories for each active layer;
- Number and layout of hybrid electro-photonic and electrical-only routers that utilize a set of photonic $p_i \in P$ and/or electrical links $e_k \in E$ to support communication for a given multi-application workload;
- Sizes of photonic concentration regions (PCRs) that determine the cores/memories allowed to use each hybrid photonic routers on the die (see Section 3 for details);
- Serialization degree D_n at electro-photonic interfaces;
- Hop count (1-hop or 2-hop) selection for FSNPI links;

5. *3D-HELIX* Synthesis Framework Overview

In this section, we present our framework for synthesizing hybrid nanophotonic-electric 3D NoCs for heterogeneous CMPs. The framework consists of the following steps as shown in Fig. 4 *(i)* task-to-core mapping; *(ii)* formulating 3D layers; *(iii)* floorplanning; *(vi)* Steiner tree based network formation; *(v)* link clustering and dual level router mapping; *(vi)* TSV assignment; *(vii)* PCR allocation; *(vii)* conflict analysis and resolution; and *(viii)* 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.

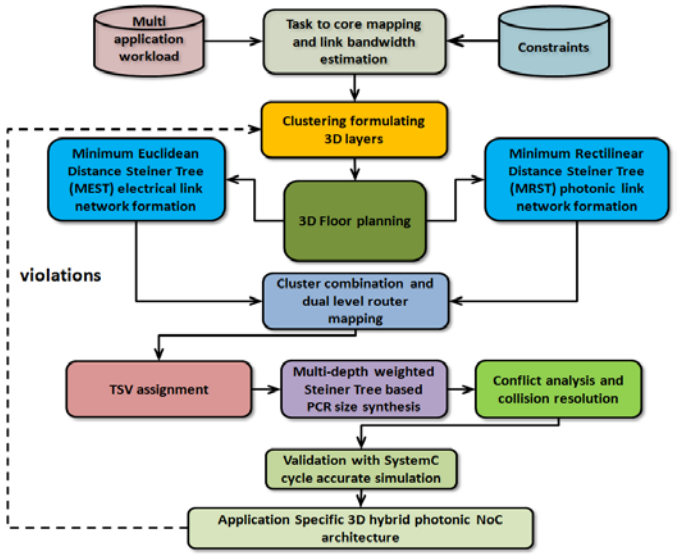


Fig 4. *3D-HELIX* hybrid nanophotonic-electric NoC synthesis flow

5.1 Task to core mapping

In this first step, we perform task to core mapping, scheduling, 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 [35]. We implement a genetic algorithm (GA) [36] to accomplish task to core mapping. The GA *chromosome* consists of possible mappings for the given application tasks to available cores, as well as virtual link type (electrical or photonic) between cores to 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 mapping and task schedules to minimize this power.

5.2 Cluster formulation for 3D layers

As a planar 2D electrical NoC is efficient for transmission of small length messages, whereas interlayer and FSNPI links can provide

significant benefits for large packet length messages, we implement a k-means clustering algorithm to partition cores to dies based on their communication characteristics. Here k-stands for number of clusters that represents number of layers within the 3D CMP. We utilized sum of weighed bandwidth squares as a minimization function. The major constraint was to ensure that the sum of areas of cores for each die was less than the planar die area. This constraint allows a 2D floorplanner to work seamlessly to position cores on each die (Section 5.3). The sum of weighted bandwidth squares within a cluster is minimized so that horizontal communication bandwidth in each layer is minimized while the vertical communication bandwidth is maximized.

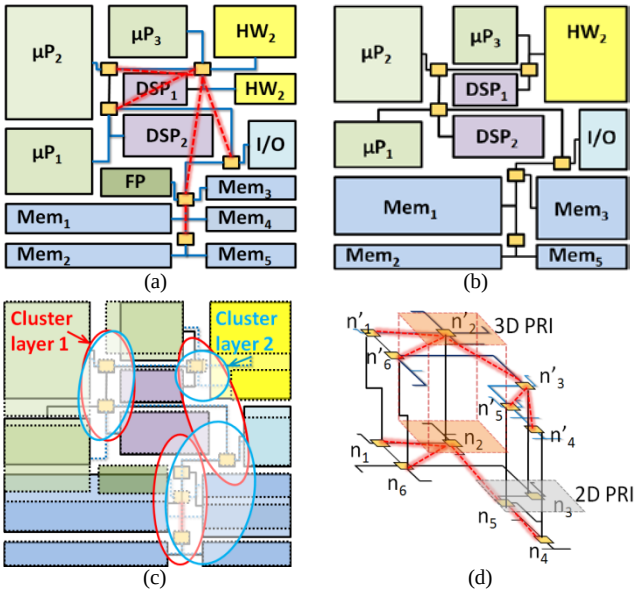


Fig 5. (a) Layer one, Steiner Tree (MEST) for electrical network and Minimum Rectilinear Distance Steiner Tree (MRST) for FSNPI links (b) Layer two, MEST and MRST (c) clustering for dual level router mapping (d) TSV assignment and PCR generation

5.3 Floorplanning

In the next step, floorplanning is performed within each cluster obtained by k-means clustering for each 3D layer, to determine more precise placements for cores in dies. This step also influences the network topology and therefore must be cognizant of the communication between cores during floorplanning. In general, communication delay and power consumption play a significant role in determining the optimal network topology for an application. With hybrid NoC architectures, the floorplanning step becomes more complicated as the power consumption and delay of electrical wires and FSNPI links differ significantly. As per our best knowledge there is no floorplanning tool available that can support such hybrid FSNPI and electrical wire based network architectures during floorplanning. We therefore designed an enhanced system-level NoC floorplanning tool that uses mixed integer linear programming (MILP) to perform core placement on a die. Our MILP minimization objective function is a linear combination of the weighed communication power-latency and overall chip area, representing the metrics optimized in this step:

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

where, $l(u,v) \times Cp_{i,j}$ is the weighed communication power and link distance between cores, $\propto$ and β are constants, and X_{max} and Y_{max} are the dimensions of the die along the X and Y axes. During the floor planning process we also define unity aspect ratio X_{max}/Y_{max} as a constraint to obtain an approximately square shaped floorplan. As FSNPI links consume less power than electrical links, the floorplanner allows placing cores communicating via FSNPI links farther apart than cores communicating via electrical links. The floorplanner also works independently for each layer. 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 Fig. 5(a) and (b).

5.4 MEST and MRST based network formation

In this step, we initiate the network formation by generating Rectilinear and Euclidian Minimum distance Steiner Trees, where the link weight for link $l_{i,j}$ is a function of normalized communication bandwidth, power, and latency:

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

where $\psi_{i,j}$, $bw_{i,j}$, and $latency_{i,j}$ are link power consumption, link bandwidth, and link latency, respectively. We use separate weight values (α) for FSNPI links and electrical links due to their power consumption differences. We ultimately generate a Minimum Euclidean Distance Steiner Tree (MEST) for electrical links and a Minimum Rectilinear Distance Steiner Tree (MRST) for FSNPI links, as shown in Fig. 5(a)-(b). 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. Finally, we connect all cores (virtual routers) utilizing FSNPI and electrical links.

5.5 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, and map conventional and hybrid routers such that router counts are minimized and utilization of links and routers is improved. This step considers tradeoffs between local and global communication assignment to electrical or FSNPI links. We developed a heuristic that computes *connection strength* between each node pair on the same layer based on 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 Fig. 5(c). We repeat this process for each layer. The clusters are created utilizing a connection strength threshold such that intra-cluster short distance communication paths can be performed by utilizing electrical links and inter-cluster transmission can be performed using FSNPI links.

Each cluster represents a router in the final solution. But we still need to determine which communication flows will utilize FSNPI links, electrical links, or a combination of both. This is accomplished 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 uses MRST links. The n and n'cores are linked with weights based on MRST and MEST links as shown in Fig. 5(d). 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 to determine 1-hop or 2-hop routing for FSNPI-based communication flows, to maximize bandwidth utilization while minimizing router resources.

5.6 TSV assignment

In this step, we assign vertical TSV electrical links and FSNPI links for inter-layer communication. This is achieved by again applying the *push-relabel maximum flow* algorithm. For every core n_i in the system, we create a corresponding pseudo core n_i^i, where inter-layer communication for all n cores uses electrical links and all n'cores uses FSNPI links. Using *the push-relabel maximum flow* algorithm we generate a combined interconnect in 3D layers and then merge the n and n'cores. Similarly to the intra-layer mapping discussed in Section 5.5, we utilize *the max-flow min cut* algorithm to determine 1-hop or 2-hop routing for FSNPI-based inter-layer communication flows, to maximize bandwidth utilization while minimizing router resources. This process also can add or delete FSNPI links as needed to meet any unsatisfied bandwidth or latency constraints and balance performance and power requirements based on 1-vs-2 hop routing trade-offs.

5.7 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 as shown in Fig. 5(d). 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 less than the size of M_{th} transverse electrical links, while messages that exceed the threshold size travel to gateway interface routers and utilize FSNPI 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 FSNPI links less advantageous than over electrical links.

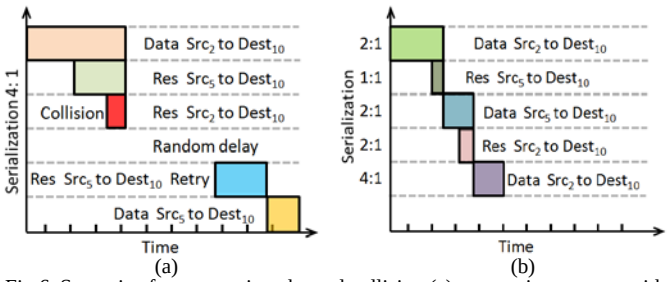


Fig 6. Scenarios for reservation channel collision (a) reservation process with FSNPI collision (b) reservation process after adjusting serialization degree

5.8 Conflict analysis and resolution

To reduce collision probability within FSNPI channels, this final step attempts to minimize interference between the various FSNPI communication transactions. Our implementation uses multiple pipelined FSNPI links with separate reservation and data transfer channels, so that 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 however 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 FSNPI 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. Fig. 6 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. Fig 6(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. Fig 6(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 serialization circuitry).

6. Experiments

6.1 Applications

We synthesized application-specific hybrid 3D-NoC architectures for five MiBench [37] application benchmark categories: *(i)* Automotive and Industrial Control, *(ii)* Consumer, *(iii)* Office Automation *(iv)* Networking, and *(v)* Security. Applications across these categories generally possess different communication characteristics. As the MiBench benchmarks are written for a single processor, we created our own multithreaded implementations of these benchmarks using Linux *pthreads*. We then generated multi-application workloads that 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 [35].

TABLE 1. MIBENCH [37] APPLICATIONS FOR APPLICATION CATEGORIES

Industrial	Consumer	Office	Networking	Security
basicmath[8]	jpeg [9]	ghostscript[12]	dijkstra[12]	blowfish[9]
bitcount[11]	lame [6]	rsynth[11]	patricia[14]	rijndael[8]
qsort [8]	mad [7]	stringsearch[11]		sha [9]
susan [9]	tiff2bw[8]			
	tiff2rgba[9]			

Table 1 presents the various application categories, the 17 applications and their corresponding number of threads that we implemented. We created 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 *3D-HELIX* framework with NAS [38] and PARSEC [39] 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 CMPs. We considered the following benchmarks: *(i)* blackscholes, *(ii)* bodytrack, *(iii)* canneal, *(iv)* dedup, *(v)* facesim, *(vii)* ferret, *(viii)* fluidanimate, *(ix)*freqmine, *(x)* streamcluster, *(xii)* swaptions, *(xiii)* vips, *(xiv)* x264. NAS benchmarks are derived from computational fluid dynamics (CFD) applications. We considered the following benchmarks: *(i)* Embarrassingly Parallel *(ii)* Conjugate Gradient, *(iii)* Multi-Grid, *(iv)* Fourier Transform, *(v)* Integer Sort, *(vi)* Lower-Upper Gauss, *(vii)* Block Tri-diagonal, *(viii)* Scalar Penta.

TABLE 2. COMMUNICATION SYNTHESIS GA PARAMETER RANGES

Synthesis Parameters	*Range low*	*Range high*
Source Processor ID	1	$m \times n \times l$
Destination Processor ID	1	$m \times n \times l$
Generation Index	1	20
Number of Data Packets	1	4096

6.2 Experimental Setup

Our core mapping GA *chromosome* consists of parameters with ranges as defined in Table 2. Based on our empirical analysis, we chose an initial population size that included 3200 randomly generated application mappings. We evaluated a range of crossover mutation and probabilities and ultimately utilized probability values of *0.56* and *0.27* respectively. The best fitness value *chromosome* in each iteration that 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 (set to a value of 7500).

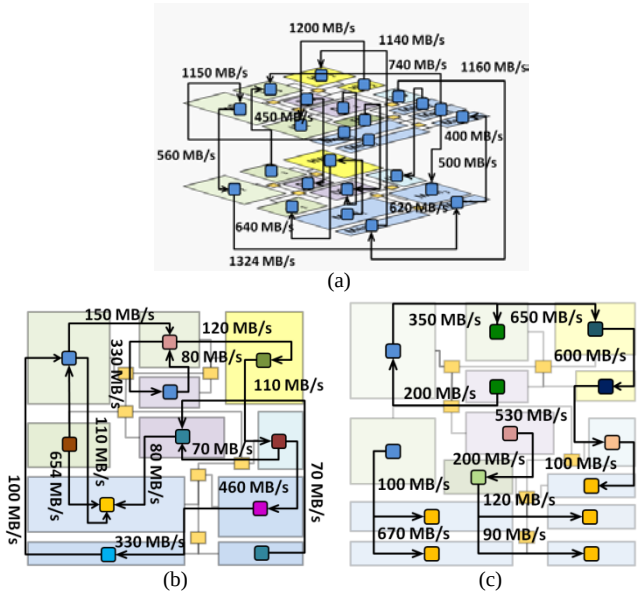


Fig 7. Communication trace graph for multiple parallel applications (a) inter layer communication graph (b) layer 1 communication graph (c) layer 2 communication graph.

Fig. 7(a) shows the enhanced communication trace graph (CTG) of three office applications running in parallel, which is generated after running the GA algorithm and performing core to die mapping. Note the initial link selection that allocates some flows to electrical links and others to FSNPI links. Fig. 7(b) shows the floorplanning solution for this CTG graph for layer 1 and Fig. 7(c) shows the floorplanning solution for layer 2, where each application is 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

empirical analysis. We utilized a public domain GeoSteiner3.1 Steiner Tree solver [40] to generate the MEST and MRST networks and utilized the lp_solve optimizer [41] to solve the Mixed Integer Linear Programming (MILP). We set weight values for $\propto$ to *0.38* and β to *0.62*, during MEST generation for electrical links and MRST generation for FSNPI links. We created clusters utilizing a *0.44* connection strength threshold to optimize electrical intra-cluster short distance communication and inter-cluster transmission using FSNPI links. Based on our analysis, we set a normalized M_{th} threshold of *39* bytes 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 FSNPI links. We combined the various components of our *3D-HELIX* framework using a python scripting interface [42].

The static and dynamic power consumption of electrical routers as well as the power consumption for optimally sized repeated Cu wires was obtained from a modified version of the Orion 2.0 simulator [43]. Our synthesis process targeted the *18 nm* node technology and utilized a *400 mm*^{2} SoC die area. The delay of an optimally repeated and sized electrical (Cu) wire at 18 nm was assumed to be *32 ps/mm*. 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 *1µm* thick *5V* modulator with *10×10µm*^{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 [44]. 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* [13]. Our hybrid NoC was modeled at the cycle accurate granularity by extensively modifying an in-house cycle accurate SystemC-based [45] NoC simulator derived from the Noxim [46] simulator.

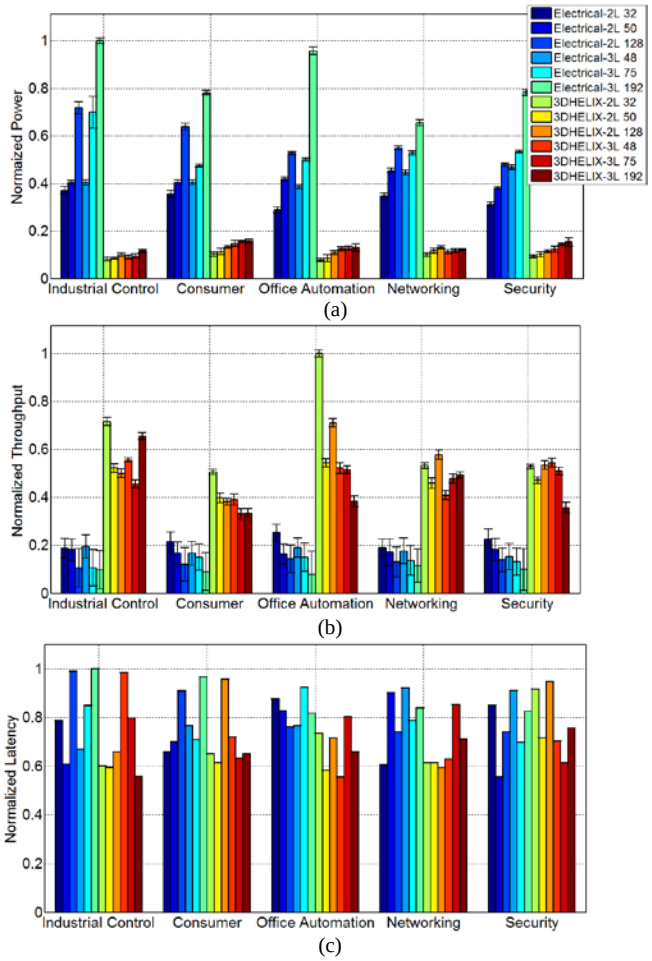


Fig 8. MiBench [37] synthesis results (a) Power (b) Throughput (c) Latency

6.3 Experimental Results

In this section, we present experimental results obtained by our *3D-HELIX* framework for the 5 MiBench multi-application workloads [37] and the NAS and PARSEC benchmarks. To compare the quality of the synthesized solution, we considered synthesized 3D electrical NoCs generated by using a subset of steps from 3D-HELIX relevant to electrical NoC synthesis (i.e., the photonic link design components were not used). We synthesized hybrid nanophotonic-electric 3D NoCs with small, medium and large sizes defined by (number of cores-number of layers): 32-2L, 50-2L, 128-2L, 48-3L, 75-3L and 192-3L; and compared the results against electrical NoCs of the same sizes.

Fig. 8 shows the results for the MiBench multi-application workloads across the various NoC sizes. Our *3D-HELIX* synthesis framework provides on average 3.66×, 4.11×, 4.97×, 3.53×, 4.30×, 6.16× reduction in power for 32-2L, 50-2L, 128-2L, 48-3L, 75-3L and 192-3L NoC platform sizes respectively compared to synthesized application-specific electrical 3D NoCs of the same sizes. *The results indicate the significant potential of including free-space photonic links into 3D-ICs.* The improvement obtained is a result of *(i)* congestion reduction in the electrical links due to offloading of a large portion of the global communication to FSNPI links; *(ii)* reduction in electrical link and buffer switching activity; *(iii)* shorter link lengths and hop counts; and *(vi)* smaller buffer resources, compared to the synthesized 3D electrical application-specific NoC. Our synthesis framework is able to achieve a significant reduction in the number of gateway interface routers through clustering and dual level router mapping, with as few as 37% gateway interface routers compared to the router count before clustering. The dual level router mapping step also added paths enabling inter-cluster long distance global communication using 2-hop FSNPI links to minimize power. We also observed that our conflict analysis and resolution step reduced MQW modulator and detector counts by approximately 50% by intelligent management of serialization degrees.

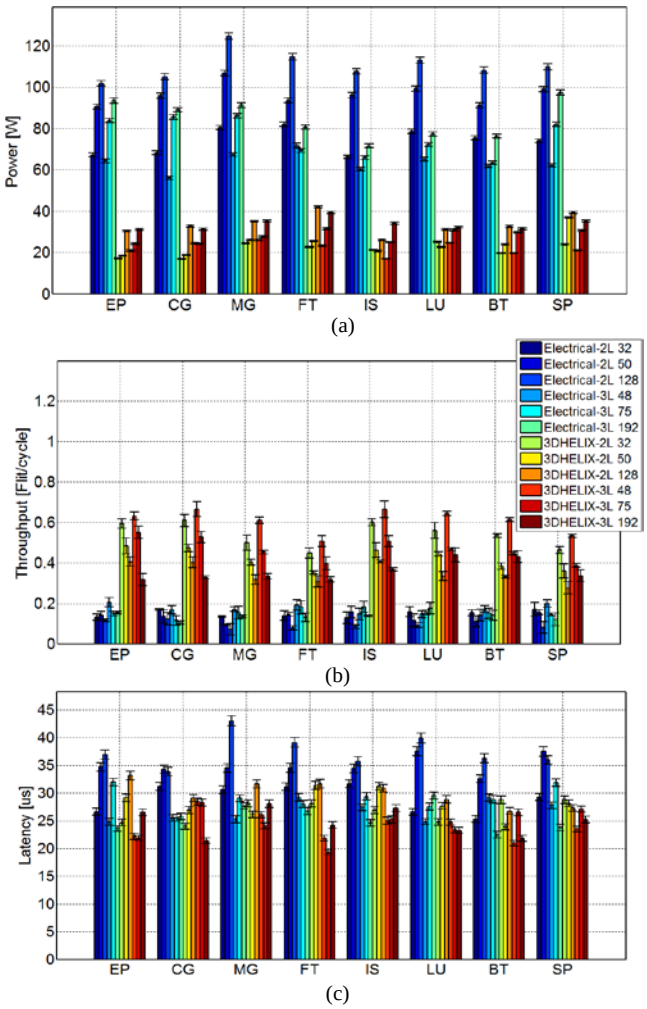
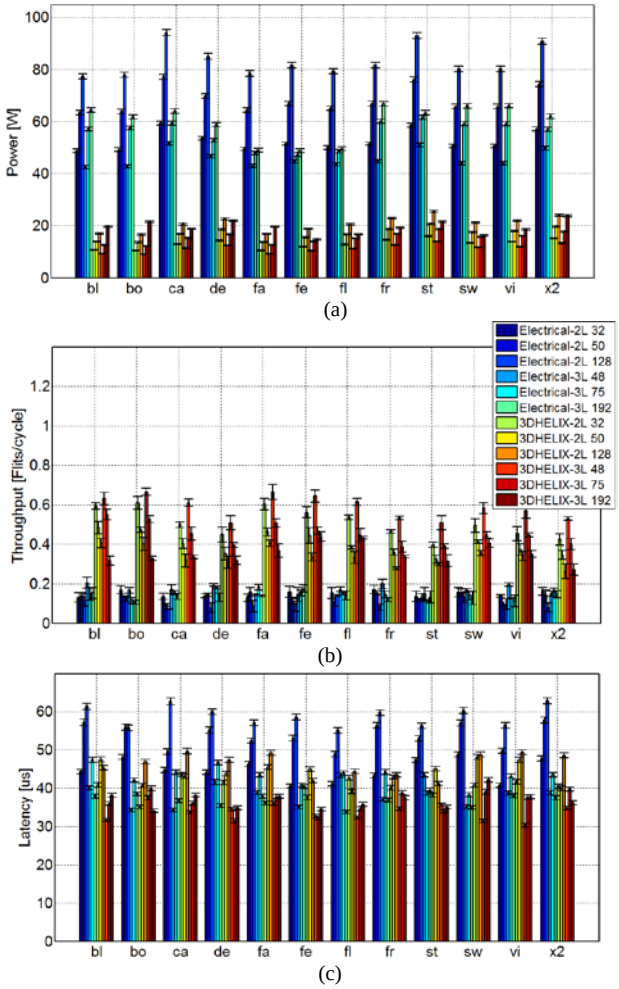


Fig 9. NAS [38] synthesis results (a) Power (b) Throughput (c) Latency

Fig. 9 shows the results for the NAS benchmarks [38] (Embarrassingly Parallel (EP), Conjugate Gradient (CG), Multi-Grid (MG), Fourier Transform (FT), Integer Sort (IS), Lower-Upper Gauss (LU), Block Tri-diagonal (BT), Scalar Penta (SP)). For NAS

workloads, our *3D-HELIX* synthesis framework provides on average 3.46×, 4.01×, 3.28×, 2.86×, 2.71× and 2.50× reduction in power for 32-2L, 50-2L, 128-2L, 48-3L, 75-3L and 192-3L NoC platform sizes respectively, compared to synthesized application-specific electrical 3D NoCs of the same sizes. Fig. 10 shows the results for the 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)). For PARSEC workloads [39], our *3D-HELIX* synthesis framework provides on average 4.02×, 4.02×, 4.02×, 4.02×, 3.64× and 3.09× reduction in power for 32-2L, 50-2L, 128-2L, 48-3L, 75-3L and 192-3L NoC platform sizes respectively, compared to synthesized application-specific electrical 3D NoCs of the same sizes.



(a)

(b)

(c)

Fig 10. PARSEC [39] synthesis results (a) Power (b) Throughput (c) Latency

Our *3D-HELIX* synthesis framework was able to achieve a viable solution for all application workloads and platform 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 nanophotonic-electric 3D NoC fabrics. The *3D-HELIX* framework took around 6.5 to 14 hours synthesis time depending on the platform size and requirements of the problem being solved. We observed that the number of clusters and gateway interface correlates well with each other and by judiciously selecting FSNPI hop counts, our framework minimizes area and the number of modulators and photodetectors required, without violating performance constraints. This improvement is possible due to the *3D-HELIX* floorplanner placing cores communicating via FSNPI links farther apart and the cores communicating via electrical links closer, achieving two fold benefits by replacing long distance electrical links with more efficient FSNPI links and placing cores closer that communicate with electrical links. The breakdown of normalized power consumption in Fig. 11 demonstrates how *3D-HELIX* can effectively improve power consumption in all categories by managing nontrivial trade-offs during the synthesis process, balancing transfers across the electrical and photonic planes to provide superior results than the synthesized application-specific electrical 3D NoCs.

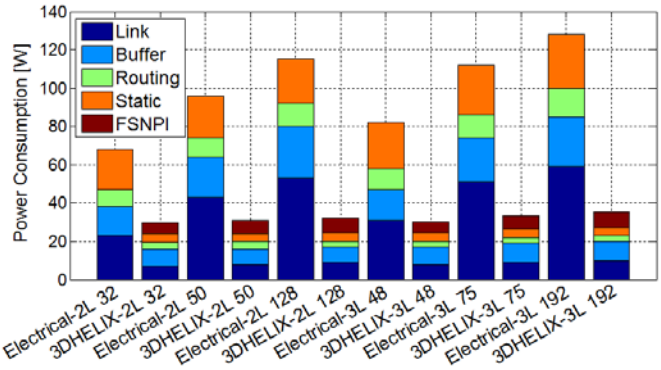


Fig 11. Normalized breakdown of power consumption for 32, 50, 128 core 2-layer and 48, 75, 192 core 3-layer configurations

7. Conclusion

In this paper, we presented the *3D-HELIX* framework to *synthesize heterogeneous application-specific hybrid nanophotonic-electric 3D NoCs for emerging 3D chip multiprocessors*. To the best of our knowledge this problem has not been addressed before in any prior work. Based on our experimental studies, we demonstrate that the proposed techniques in the *3D-HELIX* framework produce a superior hybrid nanophotonic-electric 3D NoC architecture that satisfies all performance requirements for multi-application workloads, while achieving an average from 2.5× to 6× reduction in power for multi-layer small, medium and large sized 3D-NoC based heterogeneous 3D CMP architectures, compared to synthesized application-specific electrical 3D NoCs. We believe that our proposed practical framework will bring hybrid nanophotonic-electric 3D NoCs based on FSNPI and electrical links closer to reality for future heterogeneous 3D CMPs.

Acknowledgements

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

References

[1] K. Bernstein, P. Andry, J. Cann, P. Emma, D. Greenberg, W. Haensch, M. Ignatowski, S. Koester, J. Magerlein, R. Puri and A. Young, "Interconnects in the Third Dimension: Design Challenges for 3D ICs," Proc. DAC 2007, pp.562-567.

[2] R. S. Patti, "Three-Dimensional Integrated Circuits and the Future of System-on-Chip Designs", Proc IEEE, Vol 94, No. 6, Jun 2006.

[3] S. Vangal, J. Howard, G, Ruhl, S. Dighe, H. Wilson, J. Tschanz, D. Finan, P. Iyer, A. Singh, T. Jacob, S. Jain, S. Venkataraman, Y. Hoskote, N. Borkar, "An 80-Tile 1.28 TFLOPS Network-on-Chip in 65 nm CMOS," Proc. IEEE Int'l Solid State Circuits Conf., Feb. 2007.

[4] S. Pasricha, and N. Dutt. "On-Chip Communication Architectures", Morgan Kauffman, ISBN 978-0-12-373892-9, Apr 2008.

[5] M. Haurylau, G. Chen, H. Chen, J. Zhang, N. Nelson, D. Albonesi, E. Friedman and P. Fauchet, "On-chip Optical Interconnect Roadmap: Challenges and Critical Directions," IEEE JQE 12(6), Nov 2006.

[6] H. Chen, M. Haurylau, N. Nelson, D. Albonesi, P. Fauchet, and E. Friedman, "Predictions of CMOS Compatible On-Chip Optical Interconnect," Proc. of SLIP, pp. 13-20, 2005.

[7] M. Haurylau, G. Chen, J. Zhang, N.A. Nelson, D.H. Albonesi, E.G. Friedman, and P.M. Fauchet. "On-chip optical interconnect roadmap: Challenges and critical directions", Proc. IEEE J. Sel. Quantum Electronics, pp (6):1699-1705, 2006.

[8] ITRS Technology Working Groups, http://public.itrs.net. International Technology Roadmap for Semiconductors (ITRS) 2007.

[9] 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.

[10] A. Shacham, K. Bergman, and L.P. Carloni, "The Case for Low- Power Photonic Networks on Chip," Proc. DAC, pp. 132-135, 2007.

[11] Z. Li, D. Fay, A. Mickelson, L. Shang, M. Vachharajani, D. Filipovic, W. Park and Y. Sun, "Spectrum: a hybrid nanophotonic-electric on-chip network," Proc. DAC 2009: 575-580

[12] D. Vantreasey, R. Schreiberz, M. Monchieroz, M. McLarenz, N. Jouppiz, M. Fiorentinoz, A. Davisx, N. Binkertz, R. Beausoleilz and J. Ahnz, "Corona: System Implications of Emerging Nanophotonic Technology," Proc. ISCA, pp. 153-164, 2008.

[13] A. Joshi, C. Batten, Y. Kwon, S. Beamer, I. Shamim, K. Asanovic and V. Stojanovic´, "Silicon-Photonic Clos Networks for Global On-Chip Communication", Proc. NOCS 2009.

[14] R.W. Morris, A.K. Kodi, “Power-Efficient and High-Performance Multi-level Hybrid Nanophotonic Interconnect for Multicores," Proc. NOCS, pp.207-214, 2010
[15] S. Bahirat, S. Pasricha, “Exploring Hybrid Photonic Networks-on-Chip for Emerging Chip Multiprocessors”, Proc. CODES+ISSS, 2009.
[16] J. Xue, A. Garg, B. Ciftcioglu, J. Hu, S. Wang, I. Savidis, M. Jain, R. Berman, P. Liu, M.Huang, H. Wu, E. G. Friedman, G. Wicks, D. Moore, “An intra-chip free-space optical interconnect”, Proc. ISCA, 2010
[17] A. Abousamra, R.Melhem, R., A. Jones, “Two-hop Free-space based optical interconnects for chip multiprocessors”, Proc Fifth IEEE/ACM International Symposium on Networks on Chip (NoCS), pp 89-96, 2011.
[18] R. Nair, T. Gu, K. W. Goossen, F. Kiamilev, and 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, Arlington, VA, USA, pp 9-13 October 2011.
[19] 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, pp. 355-362.
[20] K. Srinivasan and K. Chatha, “A Low Complexity Heuristic for Design of Custom Network-on-Chip Architectures”, Proc. DATE 2006
[21] K. Chatha, K. Srinivasan, and 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.
[22] S. Murali, G. D. Micheli, “Bandwidth-Constrained Mapping of Cores onto NoC Architectures,” Proc. DATE, 2004.
[23] G. Ascia, V. Catania, and M. Palesi. “Multi-objective Mapping for Mesh-based NoC Architectures”, Proc. CODES+ISSS 2004.
[24] J. Hu, R. Marculescu, “Energy-Aware Mapping for Tile-based NoC Architectures under Performance Constraints,” Proc. ASPDAC 2003.
[25] 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
[26] S. Bahirat, S. Pasricha, “A Particle Swarm Optimization Approach for Synthesizing Application-specific Hybrid Photonic Networks-on-Chip”, IEEE International Symposium on Quality Electronic Design (ISQED 2012), Mar. 2012
[27] B, Huang, S. Chen, W. Zhong, T. Yoshimura: Topology-aware floorplanning for 3D application-specific Network-on-Chip synthesis. ISCAS 2013: 1732-1735.
[28] N. Kapadia, S. Pasricha, “A Co-Synthesis Methodology for Power Delivery and Data Interconnection Networks in 3D ICs”, IEEE International Symposium on Quality Electronic Design (ISQED 2013), Mar. 2013.
[29] J. Sepulveda, C. Pedraza, G. Gogniat, R. Pires, C. Wang, M. Strum: QoS 3D-HoC Hybrid-on-Chip Communication Structure for dynamic 3D-MPSoC. LASCAS 2013.
[30] A.-M. Rahmani, K. Rao Vaddina, P. Liljeberg, J. Plosila, H. Tenhunen: Power and Thermal Analysis of Stacked Mesh 3D NoC Using AdaptiveXYZ Routing Algorithm. DSD 2012: 208-215
[31] S. Pasricha, “A Framework for TSV Serialization-aware Synthesis of Application Specific 3D Networks-on-Chip”. In VLSI Design, pp. 268-273, 2012.
[32] S. Bahirat, S. Pasricha, "HELIX: Design and Synthesis of Hybrid Nanophotonic Application-Specific Network-On-Chip Architectures", To appear IEEE ISQED, 2010.
[33] 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.
[34] G. Tian, R. Nair, M. Haney, “Integrated free-space optical interconnects: All optical communications on- and off-chip”, Optical Interconnects Conference, 2012 IEEE, pp 74-75, 2012.
[35] B. Cmelik and D. Keppel, “Shade: A fast instruction-set simulator for execution profiling”, SIGMETRICS ’94: Proceedings of the 1994 ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems, pp 128–137, 1994.
[36] D. Goldberg, “Genetic algorithms in search, optimization and machine learning”, Kluwer Academic Publishers, Boston, MA, 1989.
[37] 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 (WWC-4), December 2001.
[38] H. Jin, M. Frumkin and J. Yan. 1999. The OpenMP Implementation of NAS Parallel Benchmarks and Its Performance, NASA Ames Research Center. Technical Report NAS-99-011, Edition [Online]. Available: citeseer.ist.psu.edu/408248.html (1999).
[39] C. Bienia, S. Kumar, J. Singh and K. Li. 2008. The PARSEC benchmark buite: characterization and architectural implications. Proceedings of the 17th International Conference on Parallel Architectures and Compilation Techniques (PACT), ACM Press New York, NY, USA, 72-81(Oct 2008).
[40] http://www.diku.dk/geosteiner/
[41] http://lpsolve.sourceforge.net/5.5/
[42] http://www.python.org/
[43] A. Kahng, B. Li, L. Peh and K. Samadi, “ORION 2.0: A Fast and Accurate NoC Power and Area Model for Early-Stage Design Space Exploration” Proc. DATE, 2009.
[44] D.A.B. Miller, “Energy consumption in optical modulators for interconnects,” Opt. Express 20, pp A293-A308, 2012
[45] SystemC initiative www.systemc.org.
[46] Noxim simulator http://noxim.sourceforge.net/