

A System-Level Cosynthesis Framework for Power Delivery and On-Chip Data Networks in Application-Specific 3-D ICs

Nishit Kapadia, *Student Member, IEEE*, and Sudeep Pasricha, *Senior Member, IEEE*

Abstract—With increasing core counts ushering in power-constrained 3-D multiprocessor system-on-chips (MPSoCs), optimizing communication power dissipated by the 3-D network-on-chip (NoC) fabric is critical. At the same time, with increased power densities in 3-D ICs, problems of IR drops in the power delivery network (PDN) as well as thermal hot spots on the 3-D die are becoming very severe. Even though the PDN and NoC design goals are nonoverlapping, both the optimizations are interdependent. Unfortunately, designers today seldom consider the design of the PDN, while designing NoCs. Moreover, for each new configuration of computation core and communication mapping on an MPSoC, the corresponding intercore communication patterns, 3-D on-chip thermal profile, as well as IR-drop distribution in the PDN can vary significantly. Based on this observation, we propose a novel design-time system-level application-specific cosynthesis framework that intelligently maps computation and communication resources on a die, for a given workload. The goal is to minimize the NoC power as well as chip-cooling power and optimize the 3-D PDN architecture; while meeting performance goals and satisfying thermal constraints, for a microfluidic cooling-based application-specific 3-D MPSoC. Our experimental results indicate that the proposed 3-D NoC-PDN cosynthesis framework is not only able to meet PDN design goals unlike prior 3-D NoC synthesis approaches, but also provides better overall optimality with the solution quality improvement of up to 35.4% over a probabilistic metaheuristic-based cooptimization approach proposed in prior work.

Index Terms—3-D ICs, multicore systems, network-on-chip (NoC) synthesis, power delivery networks (PDNs), system-level CAD.

I. INTRODUCTION

DESIGNING a robust power delivery network (PDN) is critical to the overall performance of today's multiprocessor system-on-chips (MPSoCs). The PDN is required to deliver a stable power supply across the chip that is within a desired voltage range and tolerate large variations in load currents [1]. For the case of multiple voltage islands (VIs) that are used in modern MPSoC designs to minimize power dissipation,

the PDN is required to supply power at different voltage levels corresponding to the VIs, while keeping power loss to a minimum. Unfortunately, with increasing on-chip device density and decreasing voltage levels, the supply currents have risen; however, the scaling of PDN impedance has not kept up with this trend [2]. The resulting worsening of IR drops in the PDN has led to a reduction in the quality of voltage supply and negatively impacted MPSoC performance, because circuit delay in modern technologies has been shown to have a strong nonlinear relationship to the supply voltage drop [3]. This problem is even more severe in 3-D MPSoCs as the current in the PDN can be as many times more as the number of device layers compared with a 2-D MPSoC. Moreover, the number of I/O pins on an n -layered 3-D design is about n times smaller than its 2-D counterpart, thus exacerbating the problem of a degraded voltage supply in 3-D designs [4].

The design of 3-D MPSoCs faces another well-documented challenge: that of achieving an acceptable thermal profile across the 3-D die. Due to increased power densities and thermal resistivity in 3-D chips, conventional air-cooled heat sinks may be insufficient to remove heat dissipated in high-performance 3-D chips [37]. Microfluidic cooling has recently been proposed as an attractive alternative due to the superior heat removal capability of liquids in comparison with air [5], [6]. However, cooptimizing cooling and PDN costs remain a challenging and unaddressed problem for 3-D MPSoCs.

Another critical component at the heart of emerging 3-D MPSoCs is the network-on-chip (NoC) architecture that enables intralayer and interlayer communication between multiple cores. As the power dissipated in the NoC has become a significant portion of the total on-chip power, optimizing communication power has become a critical step in today's chip design methodologies. Prior works on NoC synthesis do not consider the design of the PDN, while mapping cores and designing the NoC fabric, and typically generate a single power and performance optimized NoC configuration that may or may not meet thermal constraints. Performing synthesis of the PDN for the generated NoC configuration in these cases can put stringent demands on the already strained PDN, making it extremely difficult to meet PDN constraints such as max-IR-drop or leading to overmargining for the PDN (e.g., requiring large grid-wire width) that can be wasteful and prohibitively increase overall system cost.

We recognize the key insight that different instances of voltage partitioning and core-to-tile mapping can significantly

Manuscript received March 27, 2014; revised August 14, 2014 and December 16, 2014; accepted January 29, 2015. Date of publication February 26, 2015; date of current version December 24, 2015. This work was supported in part by the National Science Foundation under Grant CCF-1252500 and Grant CCF-1302693 and in part by the Semiconductor Research Corporation.

The authors are with the Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO 80523 USA (e-mail: nkapadia@colostate.edu; sudeep@colostate.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TVLSI.2015.2399279

alter the IR-drop distribution map observed by the PDN as well as the thermal profile of the 3-D chip. Therefore, our framework integrates efficient mechanisms for VI-aware core mapping on the 3-D die and exploits the interdependence between the synthesized 3-D NoC configuration, the resulting thermal profile of the 3-D chip, and its corresponding IR-drop distribution across the 3-D PDN. The novel contributions of our cosynthesis framework are summarized as follows.

- 1) We integrate models of the PDN, microfluidic cooling, VIs, and 3-D NoCs into a system-level optimization framework.
- 2) We propose a design time force-directed algorithm to map cores onto the tiles of a 3-D die, while cooptimizing communication, thermal, and PDN design objectives to generate an optimized application-specific 3-D MPSoC.
- 3) We also propose a simulated-annealing (SA)-based cosynthesis approach to optimize communication, thermal, and PDN goals.
- 4) We design an algorithm for 3-D routing path allocation that considerably minimizes power in the 3-D NoC.
- 5) Our framework ultimately generates a set of design points (Pareto mappings) that allow a designer to tradeoff the quality of the PDN against NoC power costs and cooling power costs, and select a suitable solution that meets power, performance, and PDN cost-based design goals.

II. RELATED WORK

The problem of 2-D NoC synthesis on regular structures with multiple supply VIs has been addressed in [18] and [25]. Jang *et al.* [25] limit power overhead by reducing the total number of voltage level converters (VLCs) and multiple-clock first-in first-out (MCFIFO) frequency level converters needed for interisland transfers. Our prior work [18] improves upon [25] with heuristics that generate a better core-to-tile mapping, and a routing scheme that more aggressively optimizes interisland communication.

Given the promise of 3-D technologies, 3-D NoC synthesis in recent years has also attracted significant research efforts [7], [27], [31]. Seiculescu *et al.* [27] perform min-cut partitioning of cores to establish core-to-router connectivity in 3-D ICs, find paths for the communication flows, and place network components on the 3-D layers; while reporting savings in NoC power consumption and delay. In [31], improvements are shown over [27] by additionally considering the impact of through silicon via (TSV) serialization and network interface placement in irregular 3-D NoCs. Recently, Matsutani *et al.* [7] have proposed a novel 3-D NoC topology employing randomized long links at one or more dies of the 3-D stack, thereby significantly reducing the average hop count, communication latency, and communication energy compared with a regular 3-D mesh topology. However, none of these existing approaches have considered the impact of 3-D NoC synthesis on the efficiency and overheads associated with 3-D PDN design; in other words, these approaches are not PDN aware. These prior efforts also do not consider optimization for microfluidic cooling-based 3-D ICs.

On the other hand, techniques for optimizing PDNs in 3-D ICs have been studied in [1], [2], [4], [11], and [12]. In [4], the impact of TSV size/spacing on IR drops and voltage droops in several 3-D PDN configurations is analyzed, for SPEC benchmarks running on a quad-core SoC. In [11], SA is used to cosynthesize the floorplan and power-ground (P/G) network, optimizing wire length, area, P/G routing area, and IR drops. In [12], an integrated 3-D TSV, thermal, and power distributed network (STDN) is proposed; and an SA floorplanner is used to minimize the voltage drop and temperature in STDN. None of these PDN optimization techniques considers the impact of the 3-D NoC fabric and core mapping across layers, or considers 3-D chips with microfluidic cooling as an optimization platform. Kapadia and Pasricha [32] motivate the need for a PDN-NoC cosynthesis methodology, and proposes an SA-based cosynthesis framework for 3-D MPSoCs, which is shown to generate more efficient overall solutions compared with a PDN-unaware synthesis approach. However, the proposed approach fails to consider the thermal costs and cooling costs associated with the generated solutions.

Unlike any prior work, in this paper, we present a novel thermal-aware cosynthesis framework for core mapping, PDN design, and mesh-based NoC design in 3-D MPSoCs. To the best of our knowledge, this is the first work that proposes a system-level cosynthesis framework to cooptimize a 3-D NoC fabric, 3-D chip thermal profile, and 3-D PDN fabric to produce a more efficient overall MPSoC design. This paper significantly extends our recent conference publication [32] with new heuristics and the integration of thermal and cooling awareness during the optimization flow. Our experimental studies present detailed comparisons between our new framework in this paper and the approach from [32].

III. PDN DESIGN FOR 3-D MPSoC

High circuit density in small footprint 3-D ICs presents a unique challenge for designers of PDNs, as it requires the network to deliver significantly more current than in 2-D ICs with fewer P/G bumps, while overcoming increasingly daunting IR-drop issues. As circuit delay is strongly correlated to the supply voltage drop in modern technologies, the PDN should at least be able to restrict IR drops at each core-input within the set tolerance limit, usually 5%–10% of the rated core voltage [3]. In this paper, we consider an MPSoC platform with a 3-D mesh of tiles, with multiple voltage domains (i.e., VIs), where there is a one-to-one mapping of processing cores onto these tiles. The voltage assignments of cores running a multiprogrammed workload represent the operating frequency requirements of the tasks mapped on them, and the supply-current assignments of cores represent the power requirements for running the mapped tasks. In order for the PDN to handle multiple voltage domains, we assume that the MPSoC design is partitioned into 3-D-VIs such that cores of the same voltage are vertically aligned in the 3-D stack (as shown in Fig. 1), similar to the voltage volume concept introduced in [12] (although, we consider multiple VI configurations using all possible 2-D-shapes of VIs in our

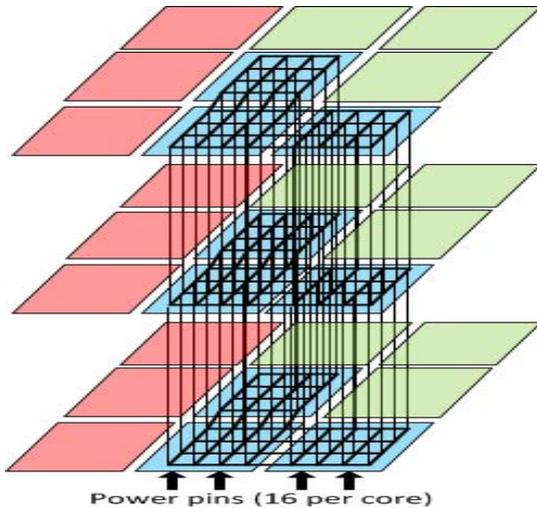


Fig. 1. Example of contiguous 3-D VIs in a 27 core ($3 \times 3 \times 3$) 3-D MPSoC. Cores within a single 3-D-VI are vertically aligned (VIs are color-coded). Only the 3-D power grid for the blue VI is shown for clarity. Each VI has a separate power grid, with 16 grid nodes per core.

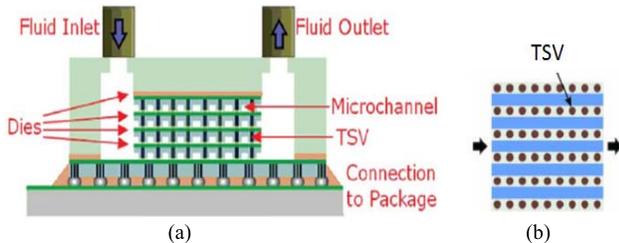


Fig. 2. Interlayer liquid cooling [5]. (a) Cross section of a 3-D chip. (b) Top-view of microchannels (blue area) with fluid flowing from left to right.

synthesis approach). This enables independent and physically disjoint 3-D power supply grids to connect all cores operating at the same voltage. The points of intersection in any 3-D power grid are termed as grid nodes, which supply power to the cores (16 grid nodes are assumed per core, as shown in Fig. 1). The interlayer connections in the grid are made using power TSVs. Even though all VIs are contiguous, we do not restrict the VI shapes to rectangles (as assumed in prior works, such as [33]). It is shown in [34] that sizing of pitches and widths of the power grid does not change the on-chip voltage distribution; therefore, in this paper, we assume fixed uniform power grids, as in [34] and [35], i.e., the grid nodes and the corresponding power-TSVs/power-pins are located uniformly over each core (as shown in Fig. 1). Note, this paper considers the steady-state characterization of the thermal, PDN, communication, and computation profiles of MPSoCs. Therefore, we investigate the steady-state effects of the global power grid in terms of static IR drop, while time-varying network characteristics, such as transient noise, are not considered.

IV. INTERTIER LIQUID COOLING IN 3-D ICs

The structure of the microfluidic layers with microchannels for fluid delivery and with the possible locations of intertier TSVs is shown in Fig. 2(b). Given the effectiveness

of microfluidic cooling, we do not need thermal TSVs in 3-D ICs, as also assumed in the other works on microfluidic cooling [5], [6], [39]. In this paper, we use a microfluidic layer under each device layer as suggested in [5]. Note that as the microfluid within microchannels progresses from the fluid inlet toward the fluid outlet [Fig. 2(a)], its capacity to absorb heat flux from device layers steadily decreases [39] from the inlet to the outlet. By increasing fluid flow rates, it is possible for the fluid to absorb more heat flux from the cores closer to the outlet, helping to reduce their temperatures.

In this paper, we assume that the hot water exiting the 3-D chip is cooled (to a fixed ambient temperature) by a heat sink with a radiator-fan assembly and fed back into the fluid inlets of the chip (as in [28]–[30]). Higher fluid flow rates require the fan-based heat sink to accommodate higher heat transfer rates (as discussed in Section VI-E), with higher number of rotations per minute (r/min). Higher r/min values in turn increase the fan power (P_{fan}), in fact, $P_{\text{fan}} \propto (\text{r/min})^3$ [28]. In addition, the electrical power required to pump the fluid into the 3-D-chip is directly proportional to the square of the flow rate ($P_{\text{pump}} \propto \text{Fl}^2$) [5]. Therefore, even though serious thermal problems can generally be alleviated by increasing fluid flow rates, indiscriminately ramping up flow rates in practice can potentially lead to prohibitively high cooling power ($P_{\text{fan}} + P_{\text{pump}}$), and thus violate chip power budgets. Consequently, the design of a microfluidic cooled 3-D MPSoC requires careful planning during the allocation of computational and network resources on the die.

V. PROBLEM FORMULATION

In this section, we present our problem formulation. We assume the following inputs to our problem.

- 1) A 3-D IC with a regular 3-D mesh NoC, with dimensions (dim_x , dim_y , and dim_z) and number of tiles $T = \text{dim}_x \times \text{dim}_y \times \text{dim}_z$ with each tile containing a core and a NoC router.
- 2) A core graph $G(V, E)$ with a set of T vertices $\{V_1, V_2, \dots, V_T\}$ representing homogenous compute cores on which application tasks have already been mapped, and the set of M edges $\{e_1, e_2, \dots, e_M\}$ that represent communication flow dependence and requirements between cores.
- 3) A minimum communication bandwidth constraint for each communication flow; and latency constraints represented as an upper bound on number of hops $\{h_1, h_2, \dots, h_M\}$ for each communication flow.
- 4) A set of preassigned triplets constituting operating voltages, operating frequencies, and maximum supply currents for the T cores $\{(v_1, f_1, i_1), (v_2, f_2, i_2), (v_3, f_3, i_3), \dots, (v_T, f_T, i_T)\}$ in the core graph, to meet compute performance and power requirements of tasks mapped to the cores.
- 5) A set of Ω supply voltage (v_{dd}) levels, which also represents the total number of VIs.
- 6) Ω separate regular 3-D power grids (each corresponding to a VI), with multiple power input pins for every grid.

7) A maximum temperature constraint Ψ for the entire chip, and a max-IR-drop constraint Γ for the PDN.

Based on the minimum voltage/frequency requirements of the T cores (which in turn depends on the performance demands of tasks assigned on each core), we assume that voltage partitioning (assignment of available {voltage and frequency} pairs to the T cores) has already been performed using techniques from [18] and [25]. Note that as the (v_x, f_x, i_x) triplets for each core are fixed, total computation power in our cosynthesis framework is predetermined. However, the manner in which compute and communication resources are mapped on the die can alter communication power, cooling power, and 3-D PDN design costs.

Objective: Given the above inputs, the goal of our proposed framework is to obtain a core-to-die mapping and synthesize a regular 3-D mesh NoC for a specific application, such that the application performance constraints (bandwidth and latency constraints), 3-D-VI contiguity constraints (all cores are contiguously placed within individual 3-D VIs), and the maximum chip temperature constraint are satisfied, while minimizing total communication power and cooling power, as well as PDN cost (represented by the max-IR-drop). Note that our proposed framework is also applicable to 2-D MPSoCs [with the vertical NoC dimension (dim_z) set to 1].

VI. COSYNTHESIS FRAMEWORK OVERVIEW

We first present a brief overview of the design flow (shown in Fig. 3) of our cosynthesis framework (FDS-CoSyn). Based on the sizes (in terms of a number of cores) of the Ω VIs and the dimensions of the mesh, a 2-D-VI-shape library is initially generated. Given the core assignments (v_x, f_x, i_x) , this library is used to generate N core-to-tile initial mapping solutions (IMs). Given the core graph $G(V, E)$, our force-directed swapping algorithm is applied on each of these IMs, to produce up to N feasible mapping solutions that are cooptimized for communication power, maximum temperature, and max-IR-drop, while satisfying hop constraints for all communication flows. These solutions undergo 3-D routing path allocation [3-D_Routing()], thermal simulation [Thermal_Eval()], and 3-D PDN synthesis [PDN_Solver()] to produce a three-tuple set of costs in terms of communication power, cooling power, and max-IR-drop. For any given core-to-tile mapping solution, our 3-D_Routing() step satisfies communication constraints, while optimizing NoC power, Thermal_Eval() determines the corresponding minimum cooling power required to satisfy the thermal constraint (Ψ), and PDN_Solver() synthesizes a 3-D PDN and evaluates the corresponding max-IR-drop. Finally, up to N candidate cosynthesis solutions are obtained. Out of these candidate solutions, the ones that have both max-IR-drop and (communication + cooling) power greater than some other solution are pruned to produce a set of Pareto optimal cosynthesis solutions, each optimized for the PDN, NoC, and thermal design objectives by varying degrees.

Thus, FDS-CoSyn essentially has two major stages: 1) N IMs are generated (as discussed in Section VI-A) and 2) candidate final solutions (for the synthesized MPSoC) are

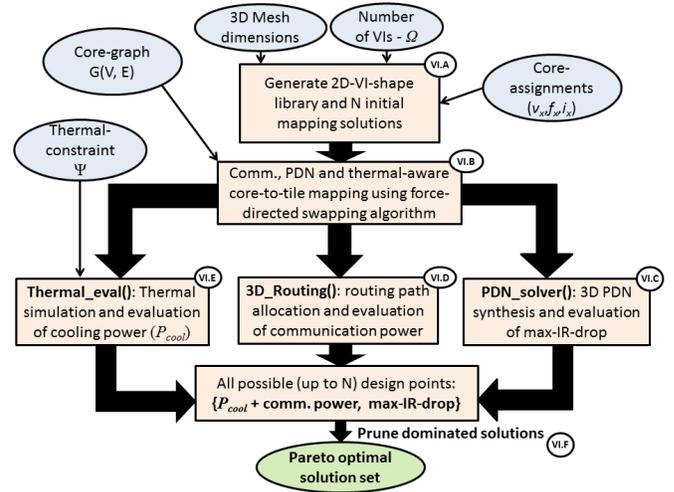


Fig. 3. FDS-CoSyn. Block arrows: transfer of multiple (up to N) design solutions. Small circles: section numbers that corresponding blocks are discussed in.

Algorithm 1 Generation of N Distinct VI Configurations

inputs: 2D-VI-shape library for all Ω VIs and mesh dimensions

```

1:  $i = 0$ 
2: while ( $i < N$ ) {  $j = 0$ 
3:   while ( $j < \Omega$ ) {
4:     Randomly choose a VI
5:     choose 2D-VI-shapes from library for  $j^{\text{th}}$  VI in an arbitrary order until one
       can be fitted onto the unused tiles of the current ( $i^{\text{th}}$ ) VI-configuration
6:   if no VI-shape can be fitted, drop this VI-configuration, go to step2
7:   else place the VI-shape, increment  $j$  }
8:   compare the new ( $i^{\text{th}}$ ) VI-configuration with all previous (0 to  $(i-1)^{\text{th}}$ )
       configurations to check if it is duplicated
9:   if duplicated, drop this VI-configuration else save it and increment  $i$  }

```

output: N distinct 2D-VI-configurations

produced from the N IMs (discussed in Sections VI-B–VI-F). In Sections VI-A–VI-F, we describe the algorithms used in each stage of our cosynthesis framework in detail.

A. Generation of Initial Mapping Solutions

Given that the MPSoC is partitioned into contiguous 3-D-VIs (as shown in Fig. 1), the number of cores within each VI is required to be a multiple of dim_z (number of device layers) in the 3-D IC. As the number of tiles in a tier occupied by each VI is known *a priori*, in addition to mesh dimensions, a set of all possible 2-D shapes of VIs can easily be enumerated. For example, Fig. 4 shows all shapes for a 2-D-VI of size 2, 3, and 4 tiles.

With the knowledge of all possible 2-D shapes for all VIs, we create a 2-D-VI-shape library for all Ω VIs. Using such a library, we invoke arbitrary shapes for the arbitrarily chosen VI and place them on the 2-D mesh to generate a set of valid (nonoverlapping) VI configurations containing all Ω VIs. The pseudocode for generating a set of N distinct VI configurations is given in Algorithm 1.

For each (i^{th}) VI-configuration (Step 2), all possible 2-D-VI-shapes corresponding to the selected (j^{th}) VI are chosen in an arbitrary order until one is found which can be fitted on to the unused tiles of the current VI-configuration (Step 5). Note that the Ω VIs are also selected in a random

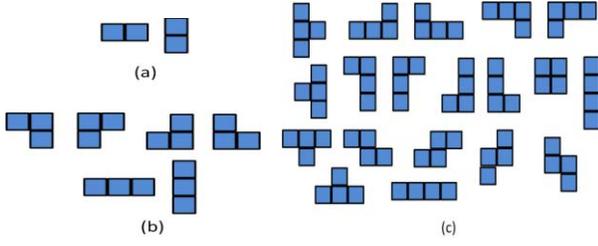


Fig. 4. Library of all possible 2-D-VI-shapes. (a) Two 2-D-VI-shapes of size 2 tiles. (b) Six 2-D-VI-shapes of size 3 tiles. (c) 19 2-D-VI-shapes of size 4 tiles.

order (Step 4). If no 2-D-VI-shape (corresponding to the size of j th VI) can be fitted on to the unused tiles, the current VI configuration is dropped and the (i th) iteration is restarted (Step 6). This procedure continues until a valid VI configuration is obtained. The new found (i th) VI configuration is saved only if it is different from all previously found ($i - 1$) VI-configurations (Steps 8 and 9). Finally, a set of N distinct VI configurations are obtained.

Corresponding to the N 2-D-VI configurations, N 3-D core-to-tile mapping solutions are arbitrarily constructed with all 3-D-VIs contiguously placed (as shown in Fig. 1).

B. Core-to-Tile Mapping With Force-Directed Swapping

Each IMS generated in the previous stage corresponds to a distinct global VI configuration. In this step, for each of the N global VI configurations, we perform swaps between adjacent cores within the same VI (i.e., intra-VI swaps), without altering the VI mapping of each generated configuration from the previous stage (Section VI-A). We associate any given mapping solution (for a VI configuration) with a force-directed map, where every core in the 3-D mesh is acted upon by various pulling forces in one or more directions.

For a given mapping, these forces are dependent on: 1) estimated intercore communication in the NoC; 2) estimated thermal profile; and 3) estimated IR-drop distribution in the corresponding PDN. The main goal of this stage is to minimize the sum total of these forces, which corresponds to reducing costs associated with communication power, cooling power, and the PDN fabric. To capture this multiobjective optimization in the force-directed map, we define the following five force-components (FC_1 – FC_5) for a core C_i (core under consideration), with α , γ , β_1 , and β_2 being weighting factors for four of these optimizations corresponding to FC_1 – FC_4 .

- 1) *Communication Traffic Optimizing Force Component (FC_1)*: The basic premise of using FC_1 is that by mapping cores such that communication flows with higher bandwidth requirements have shorter routing paths (Manhattan distances for minimal routing), the total network traffic (and thus the NoC power) is optimized. FC_1 is derived from the communication dependence (ingress as well as outgress) of core C_i with other cores on the 3-D mesh. FC_1 is subdivided into its x -, y -, and z -directional components FC_{1x} , FC_{1y} ,

and FC_{1z}

$$FC_{1\{C_i\}x} = \alpha * \sum_j MD_{jx} * BW_j \quad (1)$$

$$FC_{1\{C_i\}y} = \alpha * \sum_j MD_{jy} * BW_j \quad (2)$$

$$FC_{1\{C_i\}z} = \alpha * \sum_j MD_{jz} * BW_j \quad (3)$$

where FC_1 is computed over all communication flows associated with the core C_i under consideration. For a given mapping, MD_j is the Manhattan distance (in the respective direction) between the two communicating cores for the j th flow, and BW_j is the communication bandwidth of the j th flow. Note that Manhattan distances in negative x -, y -, and z -directions are treated as negative numbers; thus, the directional subcomponents of FC_1 can take positive or negative values.

- 2) *Max-IR-Drop Optimizing Force Component (FC_2)*: The basic premise of using FC_2 is that by mapping cores with higher supply current requirements to tiles on the 3-D mesh closer to the external input power pins, the max-IR-drop can be reduced. Assuming that the power pins are at the bottom of the 3-D stack [Fig. 2(a)], the bottom tier is indexed \dim_z and top tier has an index of 1. Therefore, FC_2 for each core is always directed in the positive z -direction (downward and toward I/O pins), with a magnitude depending on its tier index (tier#) as well as its maximum current requirement (i_{C_i}) in relation to the lowest current requirement of any core on the 3-D mesh (i_{\min})

$$FC_2\{C_i\} = \gamma * (\dim_z - \text{tier}\#_{C_i}) * (i_{C_i} - i_{\min}). \quad (4)$$

- 3) *Temperature Optimizing Force Components (FC_3 and FC_4)*: The central idea behind incorporating these force components is that by mapping the cores with higher power values ($i_{C_i} \times v_{C_i}$) closer to the fluid inlet of the microfluidic-cooling system, the chip cooling power can be optimized, while meeting the thermal constraints. We assume that the microfluidic channels are laid out along the x -direction, i.e., inlet at $x = 1$ and outlet at $x = \dim_x$. FC_3 represents the horizontal force directed in the negative x -direction (toward the microfluid inlet)

$$FC_3\{C_i\} = \beta_1 * \text{dist}_x * (i_{C_i} - i_{\min}) \quad (5)$$

where dist_x is the distance of C_i (in the negative x -direction) from the leftmost (with least x -coordinate value) core within its own VI. As discussed earlier, no swaps are performed across the 3-D-VI boundaries; therefore, the horizontal force component FC_3 is based on the location of C_i within its own VI. In addition, for a well-distributed temperature profile (which facilitates efficient cooling), the total power dissipation on each tier should be generally balanced. Therefore, FC_4 is a force component in the vertical (up $\{-z\}$ or down $\{+z\}$) direction for relatively higher current cores,

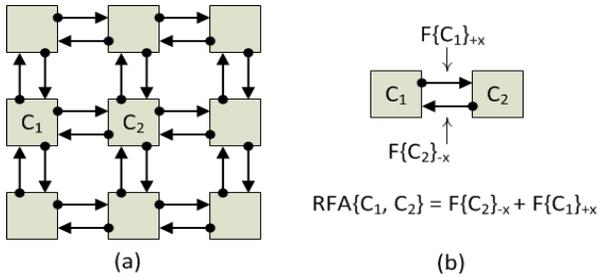


Fig. 5. Forces in the force-directed swapping algorithm. (a) Force-directed map generated for any mapping solution (z -dimension is omitted for clarity). (b) RFA between pair of adjacent cores.

which attempts to balance the power between adjacent tiers

$$\begin{aligned} & \text{if } ((i_{C_i} > i_{\text{avg}}) \text{ AND } (\text{Power}_{\text{tier}\#} > \text{Power}_{\text{tier}\#+1})) \\ & \text{FC}_4\{C_i\}_{+z} = \beta_2 * (\text{Power}_{\text{tier}\#} - \text{Power}_{\text{tier}\#+1}) \\ & \quad * (i_{C_i} - i_{\text{avg}}) \\ & \text{else FC}_4\{C_i\}_{+z} = 0. \end{aligned} \quad (6)$$

Similarly

$$\begin{aligned} & \text{if } ((i_{C_i} > i_{\text{avg}}) \text{ AND } (\text{Power}_{\text{tier}\#} > \text{Power}_{\text{tier}\#-1})) \\ & \text{FC}_4\{C_i\}_{-z} = \beta_2 * (\text{Power}_{\text{tier}\#} - \text{Power}_{\text{tier}\#-1}) \\ & \quad * (i_{C_i} - i_{\text{avg}}) \\ & \text{else FC}_4\{C_i\}_{-z} = 0 \end{aligned} \quad (7)$$

where tier# is the tier that C_i belongs to, (tier#-1) and (tier#+1) are the tiers directly above and directly below tier#, respectively; and i_{avg} is the average value of maximum rated current out of all cores on the 3-D mesh.

Note that our force-directed mapping approach can adapt to the type of cooling mechanism employed. For instance, if a conventional air-cooled heat sink is used, dist_x can be replaced by dist_z in FC_3 , to map potentially hot cores closer to the heat sink, and the power gradient between the tiers can be formulated in FC_4 .

4) *Force Component (FC₅) to Satisfy Hop Constraints:* Force component $\text{FC}_5\{C_i\}$ goes into effect (becomes nonzero) only when at least one of the following conditions hold.

- a) One or more hop-constraint violations exist on ingress or outgress flows of C_i .
- b) One or more hop-constraint(s) associated with C_i are barely met, i.e., one wrong swap can introduce hop-constraint violation(s).

If condition-I holds, the value of FC_5 (for the corresponding direction) becomes infinite, so that C_i will be swapped in this direction for the violation to be corrected. If condition-II holds, FC_5 acts as an infinite inertia force (instead of a pulling force) prohibiting C_i from being swapped in the opposite direction, thereby avoiding the introduction of a hop-constraint violation. Thus, FC_5 will only contribute to the overall force-

Algorithm 2 Core Mapping With Force-Directed Swapping

inputs: core-graph $G(V, E)$ and N initial mapping solutions

```

1: for each mapping-solution do {
2: do {
3: build the force-directed map for the current mapping solution
4: choose the pair of adjacent cores (which are not invalidated) with the
   maximum reciprocal forces of attraction (RFA)
5: if all swaps on 3D mesh are invalidated (equilibrium achieved),
   save the mapping solution and go to (step 1)
6: if the 3 pre-conditions for this swap are met, execute it and re-validate all
   swaps on the 3D mesh; go to next iteration (step 2)
7: else invalidate this swap for the current iteration; go to step 4}

```

output: up to N feasible mapping solutions optimized for all 3objectives

directed mapping if a violation exists; otherwise it will constantly act as a deterrent for new violations.

Having summarized the five main force components, we now present details of our iterative force-directed swapping algorithm that optimizes the given IMS for all three objectives (communication power, cooling power, and PDN cost). For any given mapping, a force-directed map [as shown in Fig. 5(a)] is first created, which represents the net forces in x -, y -, and z -directions. Each directed edge in Fig. 5(a) represents a force between the two cores it connects. These directed forces are computed by summing all individual directed force components (in the same direction), of the corresponding core

$$\begin{aligned} F\{C_i\} = & \text{FC}_1\{C_i\} + \text{FC}_2\{C_i\} + \text{FC}_3\{C_i\} \\ & + \text{FC}_4\{C_i\} + \text{FC}_5\{C_i\}. \end{aligned} \quad (8)$$

Note that a positive net force in $+x$ -direction is equivalent to a negative net force in $-x$ -direction. As the inertia force of FC_5 only precludes any violation causing swaps and does not contribute to the force-directed map, the FC_5 term in (8) represents just the pulling force component. Corresponding to the resulting force-directed map, reciprocal forces of attraction (RFAs) between each pair of adjacent cores on the 3-D mesh are computed, as shown in Fig. 5(b).

The pseudocode for our force-directed swapping algorithm is shown in Algorithm 2. The algorithm is intended to reduce the sum total force in the entire 3-D mesh. It is applied to all N IMSs (Step 1). After every swap on the current mapping solution, the force-directed map is rebuilt (Step 3). The algorithm searches for a pair of adjacent cores with the highest RFA on the current 3-D force-directed map to choose the next swap (Step 4). Any swap under consideration is executed only if the following preconditions are met.

- 1) Both cores belong to the same VI.
- 2) Swap will reduce the sum of all net forces in the 3-D mesh.
- 3) Inertia force (of FC_5) does not prohibit this swap.

If all three preconditions of the swap under consideration are not met (Step 7), this swap is invalidated to restrict it from being considered again until a valid swap is found. On the other hand, if all preconditions are met (Step 6), then the swap is executed and all previously invalidated swaps on the 3-D mesh are revalidated to be considered for swapping in the next iteration. With every successive swap, this iterative swapping process continues to reduce the sum of all net forces

Algorithm 3 Routing Path Allocation*input: core graph $G(V, E)$, candidate-mapping-solution*

- 1: Sort all communication flows in the increasing order of path lengths and in decreasing order of bandwidths for the same path length
- 2: **for** each communication flow in sorted list **do** {
- 3: Out of all the candidate minimal paths, choose the set of paths that would require the least number of inter-island link insertions
- 4: Out of the chosen paths, choose the subset of paths that would result in the minimum number of intra-island link insertions
- 5: Out of the chosen paths, choose the one path that uses the least number of inter-island links
- 6: Insert the necessary links and allocate the current flow bandwidth over the chosen routing path }

output :Synthesized NoC with all communication flows routed

in the 3-D mesh. Finally, when all swaps are invalidated on the 3-D mesh (i.e., no swap meets the three preconditions), a state of equilibrium is reached on the force-directed 3-D map and the swapping algorithm terminates (for the current mapping solution). As our force-directed algorithm is applied to all N IMS, up to N feasible candidate mapping solutions are output at the end of this stage.

C. LP-Formulation for 3-D PDN Synthesis [PDN_Solver()]

As discussed earlier, there are Ω regular 3-D power grids in our PDN, one for each 3-D-VI, and the individual VIs (on the 2-D plane) are not necessarily rectangular in shape. We assume equal number of grid nodes (in an $n \times n$ 2-D grid) supplying to each core on the MPSoC, where $n = 4$ in Fig. 1. Thus, given T cores on the 3-D mesh, the total number of grid nodes in the PDN are $T \times n^2$. We also assume that the power pins are located below the bottom tier, i.e., grid nodes of just the bottom tier are connected to the power pins; therefore, the total number of external power inputs are $(T \times n^2)/\text{dim}_z$ and all vertical PDN branch currents flow in the upward direction. Note that grid nodes of the Ω power grids are not connected to each other. For any given candidate mapping solution, we use a linear programming (LP) formulation to solve for the grid-node voltages and currents flowing in the branch resistances of the PDN with multiple 3-D grids. Our final metric of interest is the percentage max-IR-drop in the entire PDN, which we obtain from all the grid-node voltages. For lack of space, the reader is referred to [32] for details of our LP-formulation. Note that we employ an LP-solver [40] as it seamlessly integrates into our system-level framework; nevertheless, we validated our LP-based PDN synthesis approach using SPICE simulations.

D. Routing Path Allocation [3-D_Routing()]

For interisland communication, VLCs and frequency level converter resources (MCFIFOs) are required in the VI boundary routers. These frequency and voltage conversion components incur power dissipation and delay overheads. Thus, the main objective of our 3-D routing path allocation step is to find a minimal path for each communication flow such that the number of additional interisland link insertions is minimized. Additional interisland links are inserted only when minimal paths cannot be found within the residual bandwidth capacities of the existing interisland links.

Algorithm 3 summarizes our routing path allocation algorithm. The order in which communication flows are routed is determined in the following manner. The flows with longer minimal paths (MDs) have more choices for routing and thus have a larger scope for optimization. In addition, flows with smaller bandwidths, require less residual capacities to be accommodated within existing links. Therefore, flows are sorted in the increasing order of their path lengths, in decreasing order of their bandwidths for the same path length; and considered for routing in that order (Step 1).

For each communication flow (Step 2), we consider all candidate minimal paths. Note that, the number of all possible minimal paths between two cores on a 3-D mesh, which are N hops apart ($N = x + y + z$; where x , y , and z are the number of x -hops, y -hops, and z -hops on the 3-D path) is given by $\{^N C_x\} \times \{^{(N-x)} C_z\}$. Here, $\{^N C_x\}$ represents the number of possible ways the x -path can be constructed; $\{^{(N-x)} C_z\}$ represents the number of possible ways the z -path can be constructed, for a given x -path. Out of these candidate minimal paths, we choose a path based on the following optimization objectives (in that order).

- 1) Minimize the total number of interisland link-insertions needed on the path.
- 2) Minimize the total number of intrainland link-insertions needed on the path.
- 3) Minimize the number of interisland links used by the path.

Note that as power optimization is the principle goal of our routing and link-insertion algorithm, the primary objectives 1) and 2) reduce NoC power dissipation, whereas objective 3) reduces path latencies by incurring minimal delay overheads. To meet these objectives, we first choose paths that need the minimum total number of interisland link insertions (Step 3). Out of the chosen paths (with the same number of interisland link insertions), we choose the ones that need the minimum total number of intrainland link insertions (Step 4). Then, out of the chosen paths (with the same number of interisland and intrainland link insertions), we choose the path which crosses the minimum number of interisland links (Step 5). When a path is chosen, the current communication flow (bandwidth) is allocated to its constituent links (Step 6). Note that, while routing any flow over a given path, links insertions are performed whenever the existing link(s) cannot support the bandwidth of the current flow or if no links are available. We also perform a postprocessing design time cyclic dependence analysis using the algorithm in [38], to ensure freedom from cyclic dependence that can cause deadlock at runtime. After this step, a mapped and routed 3-D NoC-based MPSoC is obtained for the given application.

After routing is completed for all communication flows, the aggregate communication power dissipation and path latencies in the NoC are computed taking into consideration the number of links inserted, link loads, router sizes, number of VLCs, and MCFIFOs used, and corresponding voltage/frequency values.

E. Thermal Evaluation [Thermal_Eval()]

To perform thermal evaluation of any given mapping solution in our framework, we utilize the open-source thermal

emulator 3-D-ICE 2.2.5 [36], which supports steady-state thermal analysis of 3-D ICs with interlayer liquid cooling. As discussed in Section IV, by increasing fluid flow rates, the maximum temperature on the 3-D die can be reduced. Thus, to evaluate the minimum flow rate required to satisfy the maximum temperature design constraint Ψ , we invoke the 3-D-ICE tool multiple times (on each mapping solution), while increasing the flow rate in fixed increments until the thermal constraint is satisfied. Once the required flow rate (FI) is determined, the cooling power (power consumed by the pump and the cooling fan $\{P_{\text{cool}} = P_{\text{fan}} + P_{\text{pump}}\}$) corresponding to this flow rate is calculated as follows.

- 1) *Evaluation of P_{pump}* : The pump power is generally defined as $P_{\text{pump}} \propto \Delta P \times \text{FI}$, where ΔP is the pressure difference between the inlet and the outlet, and FI is the fluid flow rate. Given the number and dimensions of micro-channels, and the fluid flow rate (FI), we evaluate the pump power using the power model in [8].
- 2) *Evaluation of P_{fan}* : The thermal resistance (R_{th}) of the heat sink (in kelvin/watt) is generally expressed as

$$R_{\text{th}} = \Delta T / (dq/dt) \quad (9)$$

where ΔT is the difference between the temperature of hot fluid entering the fan-based heat sink (or exiting the 3-D-chip) and the target ambient temperature that the heat-sink is required to cool the fluid to (to feedback into the chip), and dq/dt is the required heat-transfer rate (or thermal power in watts) for this purpose.

Based on the definition of a calorie and assuming 1 mL = 1 gm for water, the required heat transfer rate to achieve fixed target fluid temperature, dq/dt , for the given FI and ΔT can be expressed as

$$dq/dt = 4.18 \times \Delta T \times \text{FI}. \quad (10)$$

Once R_{th} is calculated from (9) and (10), the corresponding fan speed (r/min) is found using the relationship between R_{th} and r/min from [30] for a 120-mm diameter fan. We assume 6 W of fan-power for a nominal fan speed of 2000 r/min for a typical commercially available 120-mm fan-based heat sink. Thus, P_{fan} is scaled based on different r/min values [$P_{\text{fan}} \propto (\text{r/min})^3$].

Finally, the cooling power ($P_{\text{cool}} = P_{\text{pump}} + P_{\text{fan}}$) is recorded for the mapping solution under consideration.

F. Solution Pruning

The set of up to N solution points produced by our cosynthesis flow will each represent a three-tuple set of costs {communication power, cooling power, and PDN max-IR-drop}. As our goal is to minimize the sum of communication power and cooling power, as well as the max-IR-drop in the PDN, we construct an optimized 2-D Pareto front of solution points, each representing {communication power + cooling power, max-IR-drop}. To this end, solution pruning is performed to remove clearly dominated solutions. The designer is ultimately presented with a set of nondominated Pareto solution points that tradeoff PDN versus (NoC + cooling power) design objectives by varying

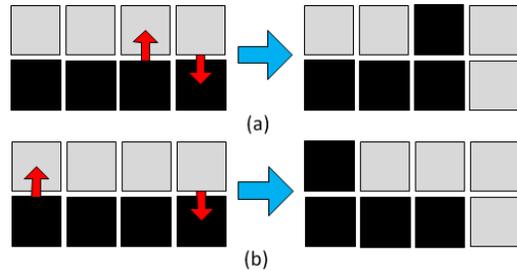


Fig. 6. Example of a tile-exchange perturbation. (a) Invalid perturbation where the contiguity of VIs is not retained. (b) Valid perturbation.

degrees, while satisfying application performance and thermal constraints.

VII. COSYNTHESIS WITH PROBABILISTIC METAHEURISTIC

SA is a probabilistic metaheuristic that has found widespread application in several physical design problems [26] and is a widely accepted technique for multiobjective optimization. For the same problem formulation (as in Section V) as well as for the same set of assumptions pertaining to 3-D-VIs, 3-D-PDN, and microfluid cooling, we also create an SA-based cosynthesis (SA-CoSyn) framework for cooptimization of the same objectives {cooling+communication power, max-IR-drop}.

In this SA-CoSyn framework, we utilize the same functions PDN_solver(), 3-D_Routing(), and Thermal_eval() discussed in Sections VI-C–VI-E, respectively. This framework is loosely based on the SA-based approach for PDN-NoC cosynthesis in [32]; however, that work did not consider thermal awareness and microfluidic cooling, and used an SA-based core-to-tile mapping approach that is different from the force-directed swapping-based core-to-tile mapping proposed in this paper.

We designed a dual-objective SA-based algorithm. Instead of saving just the best solution at each iteration, we maintain a Pareto front of 2-D [PDN and (NoC + cooling)] costs that are not dominated by any solution found so far. The three possible solution perturbations used in the algorithm are as follows.

- 1) *Perturbation 1 (Core-Swap)*: Two cores randomly selected from the same 3-D-VI are swapped. As the swap takes place within the same 3-D-VI, VI-contiguity is not a concern.
- 2) *Perturbation 2 (3-D-VI-Swap)*: Two 3-D-VIs (of equal sizes) are randomly selected to be swapped.
- 3) *Perturbation 3*: Tile exchange between two separate 3-D-VIs—the VI configuration in the 3-D mesh is changed by reciprocal occupation of a tile (tile exchange) between two adjacent VIs (with at least 2 tiles in each VI being adjacent to the other VI) on every tier of the 3-D MPSoC. A tile exchange is valid only if both participating VIs retain their contiguity (Fig. 6).

Note that entire columns of cores (across all tiers) need to be exchanged between the 3-D-VIs, and if this perturbation is valid on one tier it is valid on all tiers because all 3-D-VIs

Algorithm 4 SA-Based Co-Synthesis Framework*inputs: as described in section V*

- 1: Generate a valid initial mapping and evaluate cost (current solution)
- 2: Set T to T_{init} and counter to 0
- 3: **while** (*terminating-condition* not reached) { $K=0$
- 4: **while** ($K < K_{\text{max}}$) {
- 5: Perturb current mapping solution in one of 3 ways to get new solution:
- 6: (i) Run `PDN_solver()` to evaluate new mapping solution for max-IR-drop
- 7: (ii) Perform `3D_Routing()` and evaluate communication power
- 8: (iii) Perform `Thermal_eval()` and evaluate cooling power (P_{cool})
- 9: Compute: $\text{cost}(\text{new solution}) = \omega \times (\text{communication power} + P_{\text{cool}}) + \lambda \times (\text{max-IR-drop}) + \phi \times (\# \text{ of hop-constraint violations}) + \{\text{if } (\text{max-IR-drop} > \Gamma) \text{ then } \phi' \times (\text{max-IR-drop} - \Gamma) \text{ else } 0\}$
- 10: If no hop-constraint violations exist and PDN constraint satisfied, update the Pareto front and the counter
- 11: Set value of ACCEPT according to the SA-acceptance criterion
- 12: If (ACCEPT=1), then $\text{cost}(\text{current solution}) = \text{cost}(\text{new solution})$
- 13: $K = K + 1$ }
- 14: $T = \delta \times T$ }

output: final set of feasible solutions on the 2D Pareto front with the associated (NoC+cooling) costs and PDN costs

are vertically aligned across tiers. The pseudocode for our SA-CoSyn framework is described in Algorithm 4.

An IMS is arbitrarily generated, which satisfies basic 3-D-VI contiguity constraints. The cost of this initial solution is computed by calling the `PDN_solver()`, `3-D_Routing()`, and `Thermal_eval()` functions. The initial solution now becomes the current solution in the SA process (Step 1). To initiate the SA process, the SA-temperature parameter (T) is set to T_{init} (Step 2). At each iteration, one of the three perturbations is randomly chosen to perturb the current solution. To generate enough mapping solutions for every 3-D-VI configuration, we choose perturbations 1, 2, and 3 with probabilities 10/13, 1/13, and 2/13, respectively (Step 5). To evaluate the new (perturbed) mapping for communication power, cooling power (P_{cool}) as well as PDN max-IR-drop, our `PDN_solver()`, `3-D_Routing()`, and `Thermal_eval()` are invoked (Steps 6–8). Then, the cost of this new solution is computed (Step 9). Here, the coefficients of the terms with total number of hop-constraint violations (ϕ) and PDN constraint violation (ϕ') are set to high values, in order to penalize infeasible mapping solutions. If the new solution corresponds to no violations and does not have both power cost and PDN cost greater than any solution on the current Pareto front (nondominated solution), it is inserted into the Pareto front. Any Pareto solutions that get dominated as a result are discarded from the front. Thus, the Pareto front of solutions (with nondominated costs), is checked for an update at every iteration of SA (Step 10). Solutions with better (smaller) cost than the current solution are accepted, and solutions with worse costs are either accepted or rejected according to the following acceptance criteria (Step 11):

$$\text{if } (r < \exp([\text{cost}(\text{current solution}) - \text{cost}(\text{new solution})]/T)) \text{ ACCEPT} = 1$$

where r is a random number between 0 and 1.

(11)

After every K_{max} iterations, the SA temperature is scaled by the parameter δ . Finally, when no new solution is added

to the Pareto front for L number of consecutive SA iterations, the SA process terminates (terminating-condition), and a set of 2-D Pareto design points are produced.

VIII. EXPERIMENTAL STUDIES

A. Experimental Setup

We use the ARM Cortex-A9 multicore processors [24] as the baseline MPSoC compute cores in our experiments. These processors support three operating voltage levels ($\Omega = 3$): 0.9, 1, and 1.1 V and corresponding operating frequencies of 1310, 1550, and 1775 Mz, at the 45-nm process technology node. The maximum current requirements for the processing cores range from 1 to 4 A, based on the level of compute intensity of the tasks assigned to the respective cores. Even though we consider ARM processors in the platform used to evaluate our framework, it should be noted that our framework is applicable to multicore platforms with any processor architecture, given the operating ranges of supply voltages, frequencies, and maximum-powers for the processor. We assume that the rated maximum supply current (and the corresponding maximum power) of each core is sufficient for the associated router as well. In our studies, we use a 60-core 3-D-mesh for the MPSoC with dimensions $5 \times 3 \times 4$ ($\text{dim}_x \times \text{dim}_y \times \text{dim}_z$). In addition, we also use a 100-core 3-D-mesh of similar homogeneous cores, with dimensions $5 \times 4 \times 5$ assuming a 32-nm process technology node. The values of core voltages, frequencies, powers, and area are scaled by the factors of $0.925\times$, $1.1\times$, $0.626\times$, and $0.57\times$, respectively, to capture the effects of technology scaling (from 45 to 32 nm), as suggested in [14].

Our experiments were conducted using six parallel application benchmarks from the SPLASH-2 [21] and PARSEC benchmark suites [19]: *streamcluster*, *ocean*, and *cholesky* for 60-cores (*fluidanimate*, *lu*, and *vips* for 100-cores) of low, medium, and high communication intensities, respectively. Our core graphs are modeled based on intercore communication characterizations given in [9]. Each vertex in a core graph corresponds to a core and the edge weights represent intercore communication latency and bandwidth requirements (based on our observations of traces and communication patterns between cores). Our synthesis framework ensures that the minimum communication bandwidth constraints are satisfied, with hop-constraints representing the actual communication latency constraints.

The power values of routers and links (32-bit wide) for different voltages, frequencies, and router complexities at varying communication loads, for 32- and 45-nm process nodes are obtained from ORION 2.0 [16]. As the VLCs and MCFIFOs required to enable error-free interactions between VIs incur a power overhead that is proportional to their voltage supply, we consider the power overhead of these components, with the actual power values based on reported overheads from [15], [22], and [23]. In `Thermal_eval()`, we conservatively set the maximum thermal constraint (Ψ) to 75 °C and for the given mapping solution FI values are increased in until Ψ is satisfied. The corresponding P_{pump} and P_{fan} values are calculated, as discussed in Section VI-E. Single-phase cooling with deionized water as coolant material is assumed (similar

to [39]). 3-D-ICE assumes identical microchannels, uniformly placed, in all microfluidic layers. We set the height and the cross-sectional width of microchannels to 100 and 200 μm , respectively. Microchannels are assumed to be horizontally separated by silicon walls, each of width 400 μm (to accommodate the intertier signal and power TSVs). The temperature of the coolant fed back into the 3-D-chip is assumed to be fixed at 30 $^{\circ}\text{C}$ ambient, which would satisfy most environments. Our regular 3-D-PDN power grids are modeled based on the guidelines provided in [4]. For the 60-core mesh, with 15 cores on each tier (20 cores for the 100-core mesh), a total of 240 (320 for 100-core mesh) input power pins are used with $n^2 = 16$ grid nodes for each core. Thus, the total number of grid nodes in the entire PDN is equal to 960 (1600 for 100-core mesh). For the PDN corresponding to the 60-core mesh, the values of $R_h = 40 \text{ m}\Omega$ and $R_v = 80 \text{ m}\Omega$ are assumed (based on [4]) for the horizontal and vertical branch resistances. For the 100-core mesh, $R_h = 28 \text{ m}\Omega$ is assumed in accordance to the reduced area, whereas the TSV height is kept unchanged across technologies (as assumed in [13]).

For the implementation of our FDS-CoSyn framework, we generate N IMSs from an equal number of distinct 2-D-VI configurations, where $N = 66$ for the 60-core case ($N = 68$ for the 100-core case), is the number of 2-D-VI configurations that can be readily found using Algorithm 1. Values of 0.3, 0.1, 0.1, and 0.5 are used for α , β_2 , β_1 , and γ , respectively. For the SA-based cosynthesis algorithm, we set $T_{\text{init}} = 100$, $K_{\text{max}} = 100$, the scaling factor $\delta = 0.9$, and $L = 250$.

B. Experimental Results

In this section, we present the experimental results to highlight several interesting insights of our proposed framework. Sections VIII-B1 and VIII-B2 show the importance of employing PDN awareness and thermal awareness during the synthesis of MPSoCs, respectively. Section VIII-B3 shows the efficacy of the proposed FDS core-mapping approach, in comparison with SA approaches. Section VIII-B4 shows the advantages of using our VI-aware routing algorithm in comparison with the traditional dimension-order routing schemes.

1) *Importance of PDN-Awareness:* The SA-based PDN-NoC cosynthesis framework (SA-Basic), introduced in [32] performs PDN-NoC cosynthesis for MPSoCs. To illustrate the importance of PDN-awareness during MPSoC synthesis, we compare the approach from [32] (SA-Basic), which is PDN-aware with an SA-based PDN-unaware framework (SA-PDN unaware), where the optimization objective is primarily to minimize NoC power. The 2-D solution space generated by the two frameworks for six SPLASH-2 and PARSEC benchmarks is shown in Fig. 7, with each candidate solution characterized by its communication power and the percentage max-IR-drop in the PDN. The 2-D Pareto front enables the designer to choose a design point for an appropriate tradeoff between NoC costs and PDN costs. In the SA-PDN-unaware framework, we run

TABLE I
IMPORTANCE OF PDN-AWARENESS: % GROSS IMPROVEMENTS
IN TERMS OF COMMUNICATION POWER AND PDN COST
FOR SA-BASIC VERSUS SA-PDN-UNWARE

	<i>streamcluster-100</i>	<i>cholesky-100</i>	<i>ocean-100</i>	<i>fluidanimate-60</i>	<i>lu-60</i>	<i>vips-60</i>
SA-PDN-unaware	17.9,10.2	36.6,10.2	31.2,10.4	17.3,8.8	20.5,8.5	28.1,8.8
SA-Basic	18.5,9.3	36.6,9.1	32.1,9.5	15.6,8.9	21.3,7.7	28.6,7.4
% Gross Imp.	5.4	10.8	5.7	8.0	5.4	13.9

the SA algorithm exactly $10\times$ (starting with different initial mappings) to generate 10 distinct solution points (shown by black stars in Fig. 7). The PDN unaware approach, which is representative of system-level 3-D NoC synthesis approaches proposed in literature to date, optimizes for NoC power, but the solutions generated generally have much higher values of max-IR-drop. Observe that all of the solution points from SA-PDN-unaware corresponding to the three 100-core benchmarks [shown as black stars in Fig. 9(a)–(c)] violate the PDN constraint Γ (max-IR-drop of 10%), rendering infeasible final solutions.

Each design point on the Pareto front in Fig. 7 represents two different quantities: 1) communication power and 2) percentage max-IR-drop. For a quantitative comparison of the results obtained, we select the best point (knee point) on each Pareto front. The knee point of a 2-D Pareto front is characterized by a small improvement in one objective causing a large deterioration in the other objective, thus making it unattractive to move in either direction on the front [10]. In our analysis, we define the knee point as the point on the Pareto front with the smallest ratio of (improvement in one objective)/(deterioration in other objective) when moving in either directions. The encircled points in Fig. 7 represent the knee points associated with their respective Pareto fronts. As the goal of our cosynthesis framework is overall optimization, evaluating solutions on the basis of only power or IR drop will be inappropriate. Therefore, while comparing knee-points of two different Pareto fronts, we consider the percentage gross-improvement: (% improvement in power) + (% improvement in max-IR-drop).

The solution costs associated with the knee-point of each Pareto front in Fig. 7 for SA-Basic and SA-PDN-unaware are tabulated in Table I. Each cell in the first two rows shows two numbers that represent communication cost (power) and PDN cost (max-IR-drop), respectively. The last row for each benchmark suite shows the percentage gross-improvement per benchmark obtained with SA-Basic over SA-PDN-unaware. This improvement ranges from 5.4% to 13.9%. Thus, it can be observed from Fig. 7 and Table I that considering PDN-awareness not only results in feasible PDN solutions but also solutions with better overall optimality.

2) *Importance of Thermal Awareness:* To show the significance of considering thermal awareness in our cosynthesis framework, we use a thermal-unaware subset of our FDS-CoSyn framework (FDS-Basic). FDS-Basic has no temperature optimizing force components (FC_3 and FC_4), where we use the following coefficient values: 1) $\alpha = 0.5$; 2) $\beta_2 = 0$; 3) $\beta_1 = 0$; and 4) $\gamma = 0.5$. Therefore, with

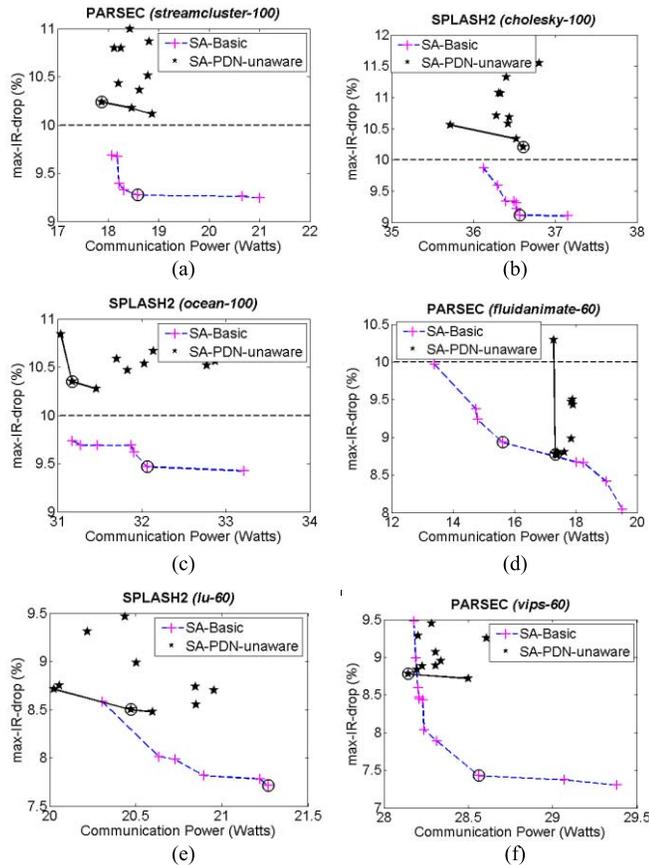


Fig. 7. Results for SA-PDN-unaware and SA-Basic (PDN-aware) synthesis frameworks. (a) streamcluster-100. (b) cholesky-100. (c) ocean-100. (d) fluidanimate-60. (e) lu-60. (f) vips-60.

FDS-Basic, communication power is somewhat better optimized compared with FDS-CoSyn, while on the other hand FDS-CoSyn tradesoff communication power with cooling power for better overall (communication + cooling) power. The 2-D solution space produced with FDS-Basic and FDS-CoSyn for six benchmarks is shown in Fig. 8, with each candidate solution characterized by its (communication + cooling) power and the percentage max-IR-drop in the PDN. By optimizing cooling-power with a slight degradation in communication power, it can be observed from Fig. 8 that FDS-CoSyn produces solutions with better overall optimality compared with the thermal-unaware FDS-Basic framework.

The solution costs associated with the knee-point of each Pareto front in Fig. 8 for FDS-Basic and FDS-CoSyn are tabulated in Table II. The power term in the solution-cost (defined in Section VIII-B1) of each design point now includes the cooling power in addition to communication power (power = communication power + cooling power). Each entry in the second and third columns of the table corresponds to {communication + cooling} power and PDN cost. Compared with FDS-Basic, the thermal-awareness in FDS-CoSyn produces improvements in cooling power, by up to 57.8% (as shown in the fourth column of Table II). As the communication power with FDS-CoSyn is generally slightly worse compared with FDS-Basic, lower improvements of up to 13.9% in (communication + cooling) power are obtained

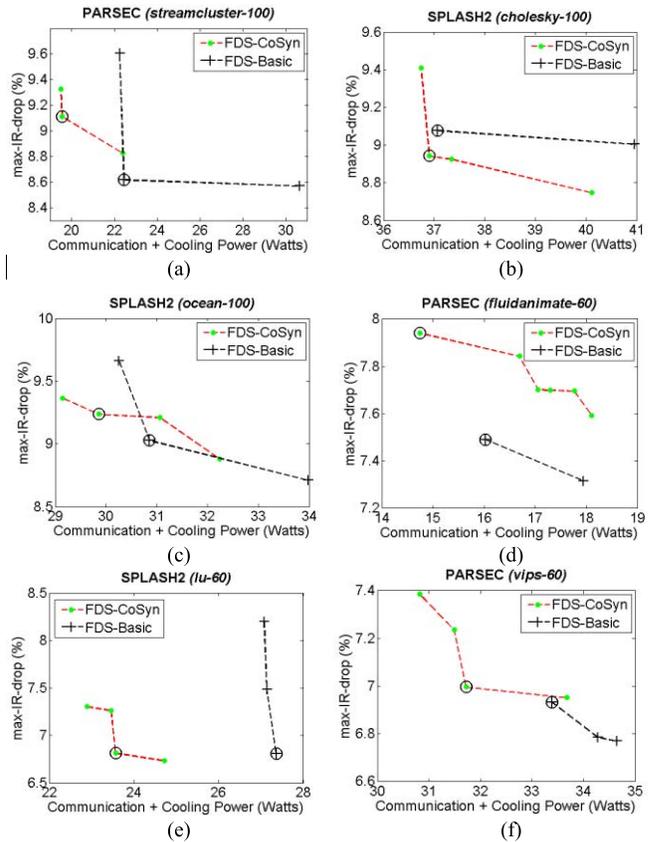


Fig. 8. Results for FDS-Basic (thermal-unaware) and FDS-CoSyn (thermal-aware) cosynthesis frameworks. (a) streamcluster-100. (b) cholesky-100. (c) ocean-100. (d) fluidanimate-60. (e) lu-60. (f) vips-60.

TABLE II
IMPORTANCE OF THERMAL-AWARENESS: PERCENTAGE GROSS-IMPROVEMENTS IN (COMMUNICATION+COOLING) POWER AND PDN COST FOR FDS-BASIC VERSUS FDS-CoSyn

	FDS-Basic	FDS-CoSyn	%Imp.Cooling-power	%Imp. power	%Gross Imp.
<i>streamcluster-100</i>	{14.6+7.8=22.5},8.6	{14.3+5.3=19.6},9.1	32.1	12.9	7.1
<i>cholesky-100</i>	{31.0+6.1=37.1},9.1	{32.3+4.6=36.9},8.9	24.6	0.4	1.9
<i>ocean-100</i>	{24.0+6.9=30.9},9.0	{25.2+4.6=29.8},9.2	33.3	3.2	0.9
<i>fluidanimate-60</i>	{12.6+3.5=16.1},7.5	{12.5+2.2=14.8},7.9	37.1	8.0	2.0
<i>lu-60</i>	{19.1+8.3=27.4},6.8	{20.1+3.5=23.6},6.8	57.8	13.9	13.7
<i>vips-60</i>	{28.0+5.4=33.4},6.9	{27.8+3.9=31.7},7.0	27.7	5.0	4.1

for FDS-CoSyn, as shown in the fifth column. Observe that benchmarks with the lowest communication-intensities (streamcluster-100 and fluidanimate-60) with a much weaker force-component for communication and a relatively strong thermal component in the FDS algorithm produce higher power-improvements with the thermal-aware FDS-CoSyn framework. In addition, due to the stronger thermal-component for these benchmarks, max-IR-drop values obtained using FDS-CoSyn are slightly higher than that obtained using FDS-Basic (even with the same value of $\gamma = 0.5$). Percentage

TABLE III
ALGORITHM COMPARISON: % GROSS-IMPROVEMENTS IN
(COMMUNICATION + COOLING) POWER AND PDN COST
FOR FDS-CoSyn OVER SA-BASIC AND SA-CoSyn

	<i>stream-cluster-100</i>	<i>cholesky-100</i>	<i>ocean-100</i>	<i>fluidanimate-60</i>	<i>lu-60</i>	<i>vips-60</i>
SA-Basic	23.9,9.3	40.9,9.6	37.4,9.5	22.5,8.0	25.2,7.8	33.7,7.7
SA-CoSyn	22.6,9.7	40.0,9.6	37.4,9.2	23.3,7.7	25.2,7.8	33.2,7.7
FDS-CoSyn	19.6,9.1	36.9,8.9	29.9,9.2	14.8,7.9	23.6,6.8	31.7,7.0
%Gross Imp. vs. SA-Basic	19.9	16.5	22.6	35.4	19.1	15.3
%Gross Imp. vs.SA-CoSyn	19.6	14.6	20.3	32.9	19.1	13.7

gross-improvements of up to 13.7% (from the last column of Table II) are obtained for FDS-CoSyn over FDS-Basic. Thus, it can be observed from Fig. 8 and Table II that considering thermal-awareness allows for a reduction in cooling power that translates into solutions with better overall optimality.

3) *Algorithm Comparison:* To the best of our knowledge, this is the only work besides [32] that proposes a cosynthesis framework to cooptimize the 3-D NoC fabric with the PDN fabric to produce a more efficient overall 3-D MPSoC design. Therefore, in order to evaluate the efficacy of our proposed force-directed heuristic-based framework (FDS-CoSyn) in comparison with the SA-based cosynthesis approach in [32] (SA-Basic), we perform a comparison study between these frameworks. We also compare these frameworks against the SA-based thermal-aware framework (SA-CoSyn) that is also proposed in this paper and described in Section VII. The results of this paper are summarized in Fig. 9 and Table III. Here, each candidate solution is characterized by its (communication + cooling) power and the percentage max-IR-drop in the PDN. Note that the solution points of SA-Basic are transferred from Fig. 7 to the plots of the same benchmark in Fig. 9, with the added cooling power corresponding to individual solution points, and the new Pareto fronts thus formed are plotted in Fig. 9. Note that our PDN_solver() and 3-D_Routing() in addition to Thermal_eval() are utilized to evaluate the max-IR-drop, communication power, and cooling power in all three framework implementations.

It can be observed from Fig. 9 and Table III that our force-directed swapping algorithm-based FDS-CoSyn framework finds excellent mapping solutions corresponding to the given IMSs. Therefore, our approach of applying the swapping algorithm to numerous seeds (initial mappings) generates better overall solutions compared with an SA-based approach. In addition, we observed that the SA algorithm spends considerable time searching for feasible mapping solutions meeting all hop constraints and the PDN constraint, while FDS-CoSyn has an inherent mechanism for overcoming hop-constraint violations and the max-IR-drop optimizing force component (FC₂) ensures that the PDN constraint is always satisfied. Thus, the force-directed swapping technique can optimize the design objectives more efficiently.

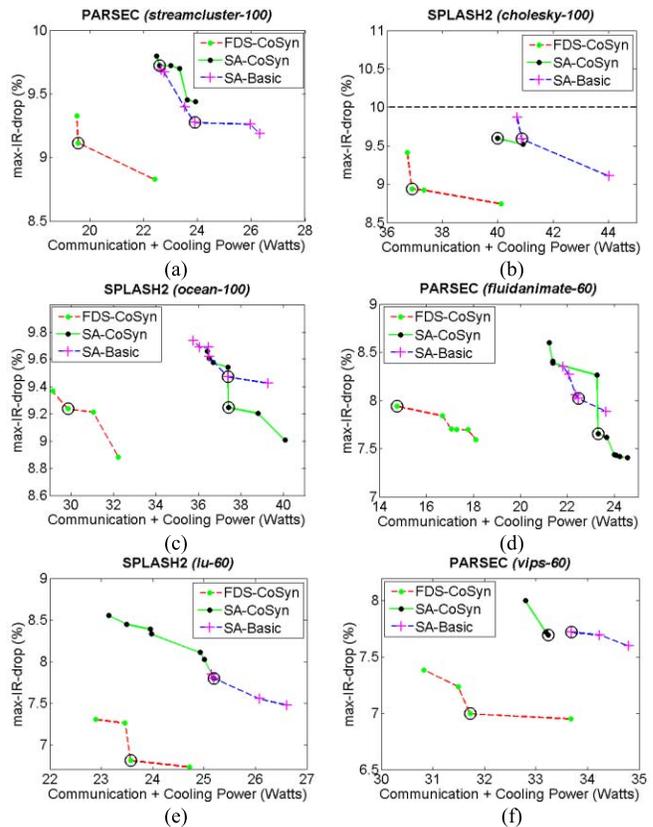


Fig. 9. Results for SA-Basic (thermal-unaware), SA-CoSyn, and FDS-CoSyn cosynthesis frameworks. (a) streamcluster-100. (b) cholesky-100. (c) ocean-100. (d) fluidanimate-60. (e) lu-60. (f) vips-60.

Our FDS-CoSyn framework produces solutions with much better overall optimality in terms of cooling costs, communication costs, as well as PDN costs, as shown in Fig. 9. In Table III, the last two rows show the percentage gross improvements (per benchmark) obtained with FDS-CoSyn over SA-Basic and SA-CoSyn. FDS-CoSyn produces percentage gross-improvements ranging from 13.7% to 32.9% over SA-CoSyn, and 15.3% to 35.4% over the thermal-unaware SA-Basic framework.

As discussed earlier, the SA frameworks (SA-Basic and SA-CoSyn) execute until the solution quality fails to improve for 250 consecutive iterations. On the other hand, we observed that our FDS framework FDS-CoSyn runs for a maximum of 68 iterations. At each of these iterations, the thermal-unaware (Basic) frameworks execute PDN_solver() and 3-D_Routing(), whereas the thermal-aware (CoSyn) framework executes Thermal_eval() as well (although in SA-CoSyn, 3-D_Routing(), and Thermal_eval() are performed only for mapping solutions that satisfy all hop-constraints as well as the PDN-constraint).

All the synthesis frameworks were simulated on a machine with Intel Core2-Duo CPU running Linux OS. Table IV shows a comparison of our FDS-CoSyn and SA-CoSyn frameworks, in terms of execution time in seconds. On average, the FDS-CoSyn, produces 12.9 \times and 4.2 \times reductions in execution time over SA-CoSyn and SA-CoSyn. This can be attributed to the following two factors (in addition to the reduced number

TABLE IV
EXECUTION TIMES (IN SECONDS) OF SA-CoSyn
AND FDS-CoSyn, AND PERCENTAGE AVERAGE
REDUCTION IN EXECUTION TIME WITH
FDS-CoSyn OVER SA-CoSyn

	<i>streamcluster-100</i>	<i>ocean-100</i>	<i>cholesky-100</i>	<i>fluidanima-te-60</i>	<i>lu-60</i>	<i>vips-60</i>
SA-Basic	19989	9392	12310	3195	3890	9100
SA-CoSyn	37210	26055	62835	20064	20834	8156
FDS-CoSyn	2934	2788	2704	1731	1761	1718

of iterations for the FDS frameworks).

- 1) Unlike in SA-CoSyn, every mapping solution in FDS-CoSyn is optimized for the thermal profile; therefore, the intermediate mapping solutions generated in SA-CoSyn correspond to worse thermal profiles compared with the final candidate solutions produced by FDS-CoSyn. Therefore at each execution of `Thermal_eval()`, compared with FDS-CoSyn, SA-CoSyn requires on average more 3-D-ICE runs of incrementing flow rates, to satisfy the thermal constraint.
- 2) The FDS technique generally produces mapping solutions with shorter communication paths, thus requiring `3-D_Routing()` to evaluate lesser number of candidate minimal paths on average.

4) *Efficacy of Routing Path Allocation:* To show the effectiveness of our VI-aware NoC routing path allocation (discussed in Section VI-D) in reducing power and latency overheads, we compare the results of our allocation approach with routing paths obtained by integrating traditional dimension-order routing schemes `XYX` and `YXZ` within the FDS-CoSyn framework. For NoC-latency calculations, we assume a 5 clock cycle latency for router-traversal, 1 cycle for link traversal, 1 cycle for VLC traversal [23], and 2 cycles for MCFIFO traversal [15], [22]. As discussed earlier, we assume minimum BW constraints for sustaining communication flows, and congestion is assumed to be negated by additional link-insertions. The latency of a particular communication flow is defined as the time taken by a flit to traverse the path from source to destination. Due to the presence of VIs, different frequencies would be encountered on a single path. We sum-up the cycle-times along the path for all the flows and then calculate the average latency (across all flows).

Table V shows the NoC-power and average NoC latency results for the knee points from the Pareto front generated for the FDS-CoSyn framework with the corresponding routing scheme. The two numbers in every table cell indicate the NoC power (in watts) and average NoC latency (in nanoseconds). Table V shows that power savings of 6.2% and 6.3%, and latency savings of 2.4% and 2.5% are obtained on average, with our VI-aware routing over `XYZ` and `YXZ` routing schemes, respectively, for the FDS-CoSyn framework. Although the savings in average latency may seem modest, savings of up to 35% (not tabulated) in the number of overhead clock cycles (cycles spent for MCFIFOs/VLCs) are obtained, which highlights the effectiveness of our VI-aware-routing scheme in minimizing the latency overheads of interisland communication. These savings could potentially be much more

TABLE V
RESULTS FOR NoC POWER (WATTS) AND AVERAGE-LATENCY
(NANOSECOND) FOR FDS-CoSyn WITH DIFFERENT
ROUTING SCHEMES: `XYZ`, `YXZ`, AND
OUR-VI-AWARE ROUTING

	<i>Streamcluster-100</i>	<i>ocean-100</i>	<i>cholesky-100</i>	<i>fluidanimamate-60</i>	<i>lu-60</i>	<i>vips-60</i>
<code>XYZ</code>	16.8,20.3	26.3,19.8	33.0,17.8	13.9,20.0	21.8,17.0	29.5,19.8
<code>YXZ</code>	16.9,20.4	26.5,19.9	33.0,17.8	13.9,20.0	21.7,17.0	29.5,19.8
<i>VI-aware</i>	15.2,19.9	24.2,19.4	31.5,17.7	12.6,19.1	20.3,16.8	28.8,19.1

pronounced in the presence of additional VIs (we assume three VIs in our experiments).

IX. CONCLUSION

This paper proposes an automated system-level framework for thermal-aware cosynthesis of PDN design and NoC design in 3-D MPSoCs. Our cosynthesis framework (FDS-CoSyn) uses a novel force-directed swapping algorithm for generating better overall core-to-tile mapping solutions, while efficiently trading off among communication, thermal, and PDN design goals. In addition, we propose an efficient routing algorithm that minimizes total NoC power. Our experimental results show that the proposed cosynthesis framework not only generates solutions with viable PDN designs unlike traditional approaches but also generates better design solutions (by up to 32.9% and 35.4%) much faster (by up to 12.9 \times and 4.2 \times) when compared with SA-based thermal-aware and thermal-unaware cooptimization approaches, respectively. Therefore, our cosynthesis approach can potentially ease design effort and improve time-to-market of emerging 3-D MPSoC designs.

We note that a routing methodology where long-range links are selectively inserted between cores to improve overall network latency (as suggested in [17] and [20]), could also be used within our FDS frameworks. Such an approach would possibly require careful considerations of the increase in complexity of design and physical implementation due to issues, such as repeater overheads, asymmetric interconnect lengths, and more complex wire-routing. As part of our future work, we intend to focus our research efforts on irregular 3-D-NoC topologies as well.

REFERENCES

- [1] B. Amelifard and M. Pedram, "Optimal design of the power-delivery network for multiple voltage-island system-on-chips," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 28, no. 6, pp. 888–900, May 2009.
- [2] P. Jain, T.-H. Kim, J. Keane, and C. H. Kim, "A multi-story power delivery technique for 3D integrated circuits," in *Proc. ACM/IEEE ISLPED*, Aug. 2008, pp. 57–62.
- [3] T. Okumura, F. Minami, K. Shimazaki, K. Kuwada, and M. Hashimoto, "Gate delay estimation in STA under dynamic power supply noise," in *Proc. 15th ASP-DAC*, Jan. 2010, pp. 775–780.
- [4] N. H. Khan, S. M. Alam, and S. Hassoun, "Power delivery design for 3-D ICs using different through-silicon via (TSV) technologies," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 4, pp. 647–658, Apr. 2011.
- [5] F. Zanini, M. M. Sabry, D. Atienza, and G. De Micheli, "Hierarchical thermal management policy for high-performance 3D systems with liquid cooling," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 1, no. 2, pp. 88–101, Jun. 2011.

- [6] A. Sridhar, A. Vincenzi, M. Ruggiero, T. Brunswiler, and D. Atienza, "3D-ICE: Fast compact transient thermal modeling for 3D ICs with inter-tier liquid cooling," in *Proc. IEEE/ACM ICCAD*, Nov. 2010, pp. 463–470.
- [7] H. Matsutani *et al.*, "Low-latency wireless 3D NoCs via randomized shortcut chips," in *Proc. DATE*, Mar. 2014, pp. 1–6.
- [8] B. Shi and A. Srivastava, "Optimized micro-channel design for stacked 3-D-ICs," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 33, no. 1, pp. 90–100, Jan. 2014.
- [9] N. Barrow-Williams, C. Fensch, and S. Moore, "A communication characterisation of Splash-2 and Parsec," in *Proc. IEEE IISWC*, Oct. 2009, pp. 86–97.
- [10] P. M. Chaudhari, R. V. Dharaskar, and V. M. Thakare, "Computing the most significant solution from Pareto front obtained in multi-objective evolutionary," *Int. J. Adv. Comput. Sci. Appl.*, vol. 1, no. 4, Oct. 2010, pp. 63–68.
- [11] P. Falkenstern, Y. Xie, Y.-W. Chang, and Y. Wang, "Three-dimensional integrated circuits (3D IC) floorplan and power/ground network co-synthesis," in *Proc. 15th ASP-DAC*, Jan. 2010, pp. 169–174.
- [12] H.-T. Chen, H.-L. Lin, Z.-C. Wang, and T. Hwang, "A new architecture for power network in 3D IC," in *Proc. DATE*, Mar. 2011, pp. 1–6.
- [13] K. Ghosh *et al.*, "Strategy for TSV scaling with consideration on thermo-mechanical stress and acceptable delay," in *Proc. 7th IMPACT*, Oct. 2012, pp. 49–51.
- [14] W. Huang, K. Rajamani, M. R. Stan, and K. Skadron, "Scaling with design constraints: Predicting the future of big chips," *IEEE Micro*, vol. 31, no. 4, pp. 16–29, Jul./Aug. 2011.
- [15] R. W. Apperson, Z. Yu, M. J. Meeuwse, T. Mohsenin, and B. M. Baas, "A scalable dual-clock FIFO for data transfers between arbitrary and haltible clock domains," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 10, pp. 1125–1134, Oct. 2007.
- [16] A. B. Kahng, B. Li, L.-S. Peh, and K. Samadi, "ORION 2.0: A fast and accurate NoC power and area model for early-stage design space exploration," in *Proc. DATE*, Apr. 2009, pp. 423–428.
- [17] U. Y. Ogras and R. Marculescu, "'It's a small world after all': NoC performance optimization via long-range link insertion," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 7, pp. 693–706, Jul. 2006.
- [18] N. A. Kapadia and S. Pasricha, "VISION: A framework for voltage island aware synthesis of interconnection networks-on-chip," in *Proc. 21st GLSVLSI*, 2011, pp. 31–36.
- [19] C. Bienia, S. Kumar, J. P. Singh, and K. Li, "The PARSEC benchmark suite: Characterization and architectural implications," in *Proc. 17th PACT*, Oct. 2008, pp. 72–81.
- [20] A. Ganguly, K. Chang, S. Deb, P. P. Pande, B. Belzer, and C. Teuscher, "Scalable hybrid wireless network-on-chip architectures for multicore systems," *IEEE Trans. Comput.*, vol. 60, no. 10, pp. 1485–1502, Oct. 2011.
- [21] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh, and A. Gupta, "The SPLASH-2 programs: Characterization and methodological considerations," in *Proc. 22nd Annu. ISCA*, Jun. 1995, pp. 24–36.
- [22] M. E. S. Elrabaa, "A new FIFO for transferring data between two unrelated clock domains," *Int. J. Electron.*, to be published.
- [23] P. Zhao *et al.*, "Low-power clocked-pseudo-NMOS flip-flop for level conversion in dual supply systems," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 17, no. 9, pp. 1196–1202, Sep. 2009.
- [24] *ARM Cortex-A9*. [Online]. Available: <http://www.arm.com/products/processors/selector.php>, accessed Feb. 12, 2015.
- [25] W. Jang, D. Ding, and D. Z. Pan, "A voltage-frequency island aware energy optimization framework for networks-on-chip," in *Proc. IEEE/ACM ICCAD*, Nov. 2008, pp. 264–269.
- [26] C. J. Alpert, D. P. Mehta, and S. S. Sapatnekar, Eds., *Handbook of Algorithms for Physical Design Automation*. Boca Raton, FL, USA: CRC Press, 2009.
- [27] C. Seiculescu, S. Murali, L. Benini, and G. De Micheli, "SunFloor 3D: A tool for networks on chip topology synthesis for 3D systems on chips," in *Proc. DATE*, Apr. 2009, pp. 9–14.
- [28] D. Shin, S. W. Chung, E.-Y. Chung, and N. Chang, "Energy-optimal dynamic thermal management: Computation and cooling power co-optimization," *IEEE Trans. Ind. Informat.*, vol. 6, no. 3, pp. 340–351, Aug. 2010.
- [29] X. Wei, G. Goth, P. Kelly, R. Zoodsma, and A. VanDeventer, "Air-water hybrid cooling for computer servers: A case study for optimum cooling energy allocation," in *Proc. IEEE ITherm*, May 2014, pp. 568–573.
- [30] J.-Y. Chang, H. S. Park, J. I. Jo, and S. Julia, "A system design of liquid cooling computer based on the micro cooling technology," in *Proc. IEEE 10th ITherm*, May/Jun. 2006, pp. 157–160.
- [31] S. Pasricha, "A framework for TSV serialization-aware synthesis of application specific 3D networks-on-chip," in *Proc. 25th Int. Conf. VLSID*, Jan. 2012, pp. 268–273.
- [32] N. Kapadia and S. Pasricha, "A co-synthesis methodology for power delivery and data interconnection networks in 3D ICs," in *Proc. 14th ISQED*, 2013, pp. 73–79.
- [33] W.-P. Lee, D. Marculescu, and Y.-W. Chang, "Post-floorplanning power/ground ring synthesis for multiple-supply-voltage designs," in *Proc. ISPD*, 2009, pp. 5–12.
- [34] Q. Zhou, J. Shi, B. Liu, and Y. Cai, "Floorplanning considering IR drop in multiple supply voltages island designs," *Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 4, pp. 638–646, Apr. 2011.
- [35] S. Kose and E. G. Friedman, "Fast algorithms for power grid analysis based on effective resistance," in *Proc. IEEE ISCAS*, May/Jun. 2010, pp. 3661–3664.
- [36] "LP_solve open-source tool," *3D-ICE Open-Source Tool*. [Online]. Available: <http://esl.epfl.ch/3d-ice.html>, accessed Feb. 12, 2015.
- [37] A. K. Coskun, J. Meng, D. Atienza, and M. M. Sabry, "Attaining single-chip, high-performance computing through 3D systems with active cooling," *IEEE Micro*, vol. 31, no. 4, pp. 63–75, Jul./Aug. 2011.
- [38] D. Starobinski, M. Karpovsky, and L. A. Zakrevski, "Application of network calculus to general topologies using turn-prohibition," *IEEE/ACM Trans. Netw.*, vol. 11, no. 3, pp. 411–421, Jun. 2003.
- [39] M. M. Sabry, A. K. Coskun, D. Atienza, T. S. Rosing, and T. Brunswiler, "Energy-efficient multiobjective thermal control for liquid-cooled 3-D stacked architectures," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 30, no. 12, pp. 1883–1896, Dec. 2011.
- [40] *Introduction to Lp_Solve 5.5.2.0*. [Online]. Available: <http://lpsolve.sourceforge.net/5.5/>, accessed Feb. 12, 2015.



variation-, and reliability-aware design methodologies, and VLSI Automation.



Nishit Kapadia (SM'12) received the Master's Degree in Electrical and Computer Engineering at California State University, Northridge, CA, USA in 2005. From 2006 to 2008, he worked as a VLSI faculty at the Department of Electronics, Sardar Vallabhbhai National Institute of Technology at Surat, India. He is currently pursuing the Ph.D. degree in Electrical and Computer Engineering at Colorado State University, Fort Collins, CO, USA. His research interests are CAD for multi-core 2D and 3D Systems, networks-on-chip, power-, thermal-, and reliability-aware design methodologies, and VLSI Automation.

Sudeep Pasricha (M'02–SM'13) received the B.E. degree in electronics and communication engineering from Delhi Institute of Technology, India, in 2000, and the M.S. and Ph.D. degrees in computer science from the University of California, Irvine, CA, USA, in 2005 and 2008, respectively.

He is currently an Associate Professor of Electrical and Computer Engineering at Colorado State University, Fort Collins, CO, USA. His research interests are in the areas of energy efficiency and fault tolerant design for high performance computing, embedded systems, and mobile computing. Dr. Pasricha is or has been an Organizing Committee Member and/or Technical Program Committee member of various IEEE/ACM conferences such as DAC, DATE, CODES+ISSS, NOCS, ISQED, VLSID, and GLSVLSI. He was the recipient of the George T. Abell Outstanding Mid-Career Faculty Award in 2014, AFOSR Young Investigator Award in 2013, ACM SIGDA Technical Leadership Award in 2012, and Best Paper Awards at the IEEE AICCSA 2011, IEEE ISQED 2010 and ACM/IEEE ASPDAC 2006 conferences. He is a senior member of the IEEE.