3D-NoC: Reconfigurable 3D Photonic On-Chip Interconnect for Multicores

Randy Morris †, Avinash Karanth Kodi †, and Ahmed Louri ‡
†Electrical Engineering and Computer Science, Ohio University, Athens, OH 45701
‡Electrical and Computer Engineering, University of Arizona, Tucson, AZ 85721
E-mail: rm700603@ohio.edu, kodi@ohio.edu, louri@ece.arizona.edu

***Abstract*—The power dissipation of metallic interconnects in future multicore architectures is projected to be a major bottleneck as we scale to sub-nanometer regime. This has motivated researchers to develop alternate power-efficient technology solutions to the performance limitations of future multicores. Nanophotonic interconnects (NIs) is a disruptive technology solution that is capable of delivering the communication bandwidth at low power dissipation when the number of cores is scaled to large numbers. Similarly, 3D stacking is another interconnect technology solution that can lead to low energy/bit for communication. In this paper, we propose to combine NIs with with 3D stacking to develop a scalable, reconfigurable, power-efficient and high-performance interconnect for future many-core systems called 3D-NoC. We propose to develop a multi-layer NIs that can dynamically reconfigure without system intervention and allocate channel bandwidth from less utilized links to more utilized communication links. Our simulation results indicate that the performance can be further improved by 10%-25% for Splash-2, PARSEC and SPEC CPU2006 benchmarks.**

I. Introduction

If the performance improvements expected from multicores is to be satisfied in the future, the underlying on-chip communication fabric must be designed carefully to be both energy-efficient and high-throughput networks. Future projections based on ITRS roadmap indicates that complementary metal oxide semiconductor (CMOS) feature sizes will shrink to sub-nanometer within a few years, and we could possibly have as many as 256 cores on-chip by the next decade. While Network-on-Chips (NoCs) design paradigm offers modular and scalable performance, increasing core counts leads to increase in serialization latency and power dissipation as packets are processed at many routers. Metallic interconnects can provide the required bandwidth due to abundance of wires in NoCs, ensuring high-speed inter-core communication within the allocated power budget in the face of technology scaling (and increased leakage currents) will become a major bottleneck for future multicore designs [1], [2].

Emerging technologies such as nanophotonic interconnects (NIs) and 3D stacking are under serious consideration for meeting the communication challenges posed by the multicores. NIs provides several advantages such as: (1) bit rates independent of distance, (2) higher bandwidth due to multiplexing of wavelengths, (3) larger bandwidth density by multiplexing wavelengths on the same waveguide/fiber, (4) lower power by dissipating only at the endpoints of the communication channel and many more [3], [4], [5], [6], [7], [8]. Similarly, 3D stacking of multiple layers have shown to be advantageous due to (1) shorter inter-layer channel, (2) reduced number of hops and (3) increased bandwidth density. A prevalent way to connect 3D interconnects is to use TSVs (through-silicon vias), micro-bump or flip-chip bonding. The pitch of these vertical vias is very small (4μm$\sim10\mu$m), and delays on the order of 20 ps for a 20-layer stack. Jalali's group at UCLA has fabricated a SIMOX (Separation by IMplantation of Oxygen) 3D sculpting to stack optical devices in multiple layers [9]. Lipson group at Cornell has successfully buried active optical ring modulator in polyscrystalline silicon [10]. Moreover, recent work on using silicon nitride have shown the possibility of designing multi-layer 3D integration of photonic layers.

To address the requirements of energy-efficient and high-throughput NoCs, we leverage the advantages of two emerging technologies, NIs and 3D stacking with architectural innovations to design high-bandwidth, low-latency, multi-layer, reconfigurable network, called **3D-NoC**. 3D-NoC consists of 16 decomposed NI based crossbars placed on four optical communication layers, thereby eliminating waveguide crossing and reducing the optical power losses. The proposed architecture divides a single large monolithic crossbar into several smaller and manageable crossbars which reduces the optical hardware complexity and provides additional disconnected waveguides which provide opportunities for reconfiguration. We also propose a reconfiguration algorithm whose purpose is to improve the performance (throughput, latency) by adapting available network bandwidth to application demand by multiplexing signals on crossbar channels that are idle. This is accomplished by monitoring the traffic load and applying a reconfiguration algorithm that works in the background without disrupting the on-going communication. Our simulation results on 64-cores and 256-cores using synthetic traffic, SPEC CPU2006, Splash-2 [11] and PARSEC [12] benchmarks provide an energy savings up to 23% and outperforms other leading NIs by more than 10% - 25% for adversial traffic via reconfiguration.

II. 3D-NoC Architecture

The proposed 3D-NOC architecture consists of 256 cores in 64 tile configuration on a 400 mm^2 3D IC. As shown in Figure 1, 256 cores are mapped on a 8×8 network with a concentration factor of four, called a $tile$ [13]. From Figure 1(a), the bottom layer, called the $electrical$ die, adjacent to the heat sink, contains the cores, caches and memory controllers. To utilize the advantage of a vertical implementation of

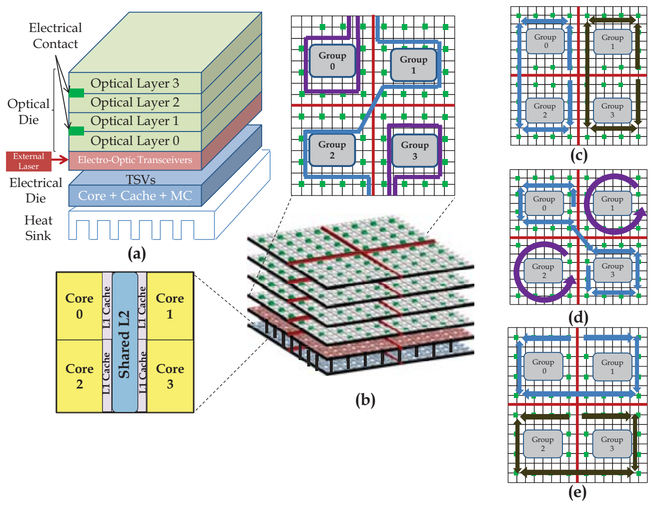


Fig. 1. Proposed 256-core 3D chip layout. (a) Electrical die consists of the core, caches, the memory controllers and TSVs to transmit signals between the two dies. The optical die on the lower most layer contain the electro-optic transceivers and four optical layers. (b) 3D chip with four decomposed nanophotonic crossbars with the top inset showing the communication among one group (layer 0) and the bottom inset showing the tile with a shared cache and 4 cores. The decomposition, slicing and mapping of the three additional optical layers: (c) optical layer 1, (d) optical layer 2 and (e) optical layer 3.

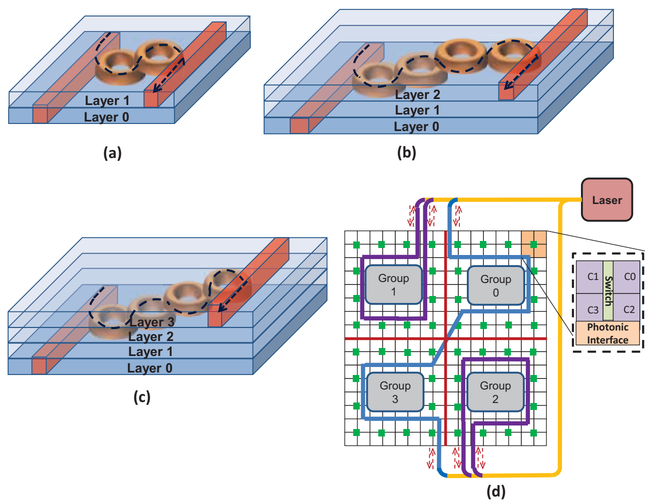


Fig. 2. Layout of micro-ring resonators for coupling light from Layer 0 to: (a) Layer 1, (b) Layer 2 (c) Layer 3 and (d) In layer 0, Group 0 cores will communicate with Group 3 cores and vice versa. The cores within Groups 1 and 2 will communicate within themselves.

signal routing, we propose the use of separate optical and core/cache systems unified by a single set of connector vias. The upper die, called the *optical* die, consists of the electro-optic transceivers layer which is driven by the cores via TSVs and four decomposed nanophotonic crossbar layers. Layers 0-3 only contain optical signal routing elements, composed almost exclusively of MRRs and bus waveguides. The top inset of Figure 1(b) shows the interconnect for layer 0 whereas Figures 1(c-e) show layers 1-3. We also provide electrical contact between layers 0/1 and 2/3 to tune ring resonators required for reconfiguration. The approach taken here is to couple between the optical layers through the use of vertically coupled ring resonators. Figure 2 shows the layout of micro-ring resonators for communication with the top 3 layers on different nanophotonic layers. Figures 2(a), (b) and (c) show the inter-layer coupling between layer 0/1 and layer 2/3 respectively. As shown in the figure, the MRR that is located adjacent to layer 1 waveguide is used to allow light to couple into the selected upper layers.

In the proposed 3D layout, we divide tiles into four groups based on their physical location. Each group contains 16 tiles. Unlike the global 64 $\times$ 64 nanophotonic crossbar design in [3] and the hierarchical architecture in [7], 3D-NOC consists of 16 decomposed individual nanophotonic crossbars mapped on four optical layers. Each nanophotonic crossbar is a 16 $\times$ 16 crossbar connecting all tiles from one group to another (Inter-group). It is composed of Multiple-Write-Single-Read (MWSR) nanophotonic channels. A MWSR nanophotonic channel allows multiple nodes the ability to write on the channel but only one node can read the channel. Figure 2(d) shows the detailed floor plan for the first optical layer. For optical layer 0, a 32 waveguide bundle is used for communication between Groups 0 and 3 and two 16 waveguide bundles are used for communication within Groups 1 and 2. For inter-group communication between 0 and 3, the first 16 waveguide bundle is routed past Group 0 tiles so that any tile within Group 0 can transmit data to any destination tile in Group 3. The remaining two independent waveguide bundles (16 waveguides) are used for intra-group communication for Groups 1 and 2 respectively. Token slot [3] is adopted in this architecture to improve the arbitration efficiency for the channel. Each waveguide used within a nanophotonic crossbar has only one receiver which we define the as receivers *home channel*. During communication, the source tile sends packets to their destination tile by modulating the light on the home channel of the destination tile. An off-chip laser generates the required 64 continuous wavelengths, $\bigwedge = \lambda_0, \lambda_1, \lambda_2 .... \lambda_{63}$. Each wavelength in a waveguide operates at 10 Gbps. In 3D-NOC, we consider a 128 bit per flit size to achieve a 640 Gbps bandwidth per link and one waveguide bundle with 64 wavelengths for each crossbar channel. Considering the total number of optical channels on the chip, 3D-NOC can achieve 20.48 TFLOPS peak performance (20.48 TB/s bandwidth).

III. Reconfiguration

As future multicores will run diverse scientific and commercial applications, networks that can adapt to communication traffic at runtime will maximize the available resources while simultaneously improving the performance. To implement reconfiguration, we propose to include additional MRRs that can switch the wavelengths from different layers to create a reconfigurable network. Further, we also propose a reconfiguration algorithm to monitor traffic load and dynamically adjust the bandwidth by re-allocating excess bandwidth from under-utilized links to over-utilized links.

A. *Implementation*

While reconfiguration can improve performance, it is essential to reduce the redundancy of components that are needed to achieve reconfiguration. Therefore, dynamic reconfiguration in R-3D-NOC (reconfigured-3D-NOC) will be limited to adjacent communication layers where bandwidth from one layer will be routed to another under different traffic and load conditions. Due to hardware and reconfiguration complexity, we restrict the reconfiguration that can take place to layers 0/1 and layers 2/3. MRRs are placed between the two layers (0/1 and 2/3) at locations where the waveguides are routed above each other. These micro-ring resonators, when activated, will switch data from one waveguide to another with similar design as shown in Figure 2.

To illustrate with an example, consider a situation where tiles in Group 0 communicate only with tiles in Group 3. Figure 3 shows the reconfiguration mechanism. The *static* allocation of channel for communication are in layer 2 as shown in Figure 3(a). This is also shown in Figure 3(b) (top layer) where tile 0 (Group 0) communicates with tile 63 (Group 3). Suppose no tile within Group 1 (in layer 1) communicates with Group 3, then we can re-allocate the bandwidth from Group 1 to Group 0 to communicate with Group 3. To implement reconfiguration, however, we need to satisfy two important requirements: (1) There should be a source waveguide which should be freely available to start the communication on a source layer, and (2) there should be a destination waveguide which also should be freely available to receive the extra packets. As mentioned before, as the two Groups 0 and 3 talk only to each other, we have the first set of waveguides on layer 0 (generally used to communicate within the group) available, therefore this satisfies the first condition. Here, we choose the waveguide which is used to communicate to destination tile 12 as the source waveguide to be reconfigured. As Group 1 does not communicate with Group 3, we can utilize the destination waveguide available in layer 1 and satisfy the second condition. Figure 3(b) shows the waveguide selected in layer 1 is the destination waveguide for tile 63 (shown as the dynamic allocation in Figure 3(a)). Therefore, during reconfiguration tile 0 has doubled the bandwidth to communicate with tile 63 by communicating with 2X bandwidth via layers 2 (static) and 1 (dynamic). Two different communication are disrupted when the reconfiguration occurs, namely, Group 0 in layer 0 can no long communicate with itself (to destination tile 12) and Group 1 in layer 1 can no longer communicate with Group 3 (tile 63).

B. *Dynamic Reconfiguration Technique*

In R-3D-NOC, reconfiguration algorithm re-allocates bandwidth based on historical information. Historical statistics such as link utilization (Link$_{util}$) and buffer utilization (Buffer$_{util}$) are collected at the optical receiver of every communication channel by hardware counters [14]. This implies that each tile within a group will have four hardware counters (one for each of the three groups) that will monitor traffic utilization. Both link and buffer utilization are used as link utilization

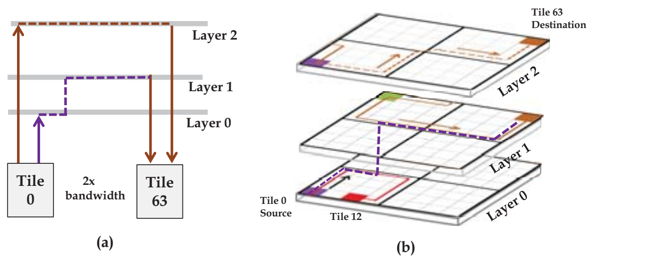


Fig. 3. Illustration of reconfiguration where additional bandwidth is given to Group 0 to communicate with Group 3. (a) Inter-layer static and dynamic bandwidth re-allocation and (b) 3D switching of bandwidth.

provides accurate information at low-medium network loads and buffer utilization provides accurate information regarding high network loads [14]. All these statistics are measured over a sampling time window called *Reconfiguration window* or phase, R$_W{}^t$, where t represents the reconfiguration time number t. This sampling window impacts performance, as reconfiguring finely incurs latency penalty and reconfiguring coarsely may not adapt in time for traffic fluctuations. For calculation of Link$_{util}$ at configuration window t, we use the following equation: Link$^t_{util}$ = $\frac{\sum_{cycle=1}^{R_W} Activity(cycle)}{R_W}$ where $Activity(cycle)$ is 1 if a flit is transmitted on the link or 0 if no flit is transmitted on the link for a given cycle. For calculation of Buffer$_{util}$ at configuration window t, we use the following equation: Buffer$^t_{util}$ = $\frac{\sum_{cycle=1}^{R_W} Occupy(cycle)/Total_{buffers}}{R_W}$ where $Occupy(cycle)$ is the number of buffers occupied at each cycle and Total$_b$$uffers$ is the total number of buffers available for the given link. When traffic fluctuates dynamically due to short term bursty behavior, the buffers could fill up instantly. This can adversely impact the reconfiguration algorithm as it tries to re-allocate the bandwidth faster leading to fluctuating bandwidth allocation. To prevent temporal and spatial traffic fluctuations affecting performance, we take a weighted average of current network statistics (Link$_{util}$ and Buffer$_{util}$). We calculate the Buffer$_{util}$ as follows: Buffer^t_w = $\frac{\sum Buffer^t_{util} \times weight + Buffer^{t-1}_{util}}{weight+1}$ where $weight$ is a weighting factor and we set this to three in our simulations [15].

After each R$_W{}^t$, each tile will gather its link statistics (Link$_{util}$ and Buffer$_{util}$) from the previous window R$_W{}^{t-1}$ and send to its local reconfiguration controller (RC) for analysis. We assume that Tile 0 of every group gathers the statistics from the remaining tiles and this can be few bytes of information that is periodically transmitted. Next, when each RC$_i$, ($\forall$ i = 0, 1, 2, 3), has finished gathering link and buffer statistics from all its hardware controllers, each RC$_i$ will evaluate the available bandwidth for each link depending on the Link$_{util}{}^{t-1}$ and Buffer$_{util}{}^{t-1}$ and will classify its available bandwidth into a select range of thresholds β_{1-4} corresponding to 0%, 25%, 50% and 90%. We never allocate 100% of the bandwidth as the source group may have new packets to transmit when the destination tile before the next R$_W$. RC$_i$ will send link information (availability) to its neighbor RC$_j$ (j $\neq$ i). If RC$_j$ needs the available bandwidth, RC$_j$ will notify

TABLE I
PSEUDO ALGORITHM FOR 3D-NOC.

```
Step 1:   Wait for Reconfiguration window, $R_W{}^t$
Step 2:   $RC_i$ sends a request packet to all local tiles requesting
          $Link_{Util}$ and $Buffer_{Util}$ for previous $R_W{}^{t-1}$
Step 3:   Each hardware counter sends $Link_{Util}$ and $Buffer_{Util}$
          statistics from the pervious $R_W{}^{t-1}$ to $RC_i$
Step 4:   $RC_i$ classifies the link statistic for each hardware counter as:
               If $Link_{util} = 0.0$
                   Not-Utilized: Use $\beta_4$
               If $Link_{util} \leq L_{min}$
                   Under-Utilized: Use $\beta_3$
               If $Link_{util} \geq L_{min}$ and $Buffer_{util} < B_{con}$
                   Normal-Utilized: Use $\beta_2$
               If $Buffer_{util} > B_{con}$
                   Over-Utilized: Use $\beta_1$
Step 5:   Each $RC_i$ sends bandwidth available information to $RC_j$, ($i \neq j$).
Step 6:   If $RC_j$ can use any of the free links then notify $RC_i$ of their
          use, else $RC_j$ will forward to next $RC_j$
Step 7a:  $RC_i$ receives response back from $RC_j$ and activates
          corresponding microrings
Step 7b:  $RC_j$ notifies the tiles of additional bandwidth and $RC_i$
          notifies $RC_j$ that the additional bandwidth
          is now available
Step 8:   Goto Step 1
```

the source and the destination RCs so that they can switch the MRRs and inform the tiles locally of the availability. Once the source/desitnation RCs have switched their reconfiguration MRRs, RC_i will notify RC_j that the bandwidth is available for use. On the other hand, if a node within RC_i that throttled its bandwidth requires it back due to increase in network demand, RC_i will notify that it requires the bandwidth back and afterwards will deactivate the corresponding MRRs. The above reconfiguration completes a three-way handshake where RC_i first notifies RC_j, then RC_j notifies RC_i that RC_j will use the addition bandwidth, and finally RC_i notifies RC_j that the bandwidth can be used. Table 1 shows a psuedo-reconfiguration algorithm implemented in R-3D-NOC. We assume $Link_{util} = 0.0$ to indicate if the link is not being used, $L_{min} = 0.10$ to indicate if the link is under-utilized, $L_{min} = 0.25$ and $B_{con} = 0.25$ to indicate if the link is normal-utilized and $B_{con} = 0.5$ to indicate that the link is over-utilized [14].

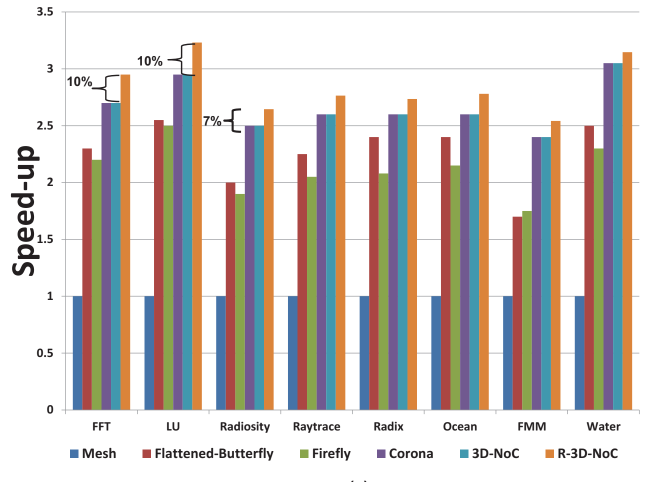
IV. PERFORMANCE EVALUATION

A. Simulation Setup

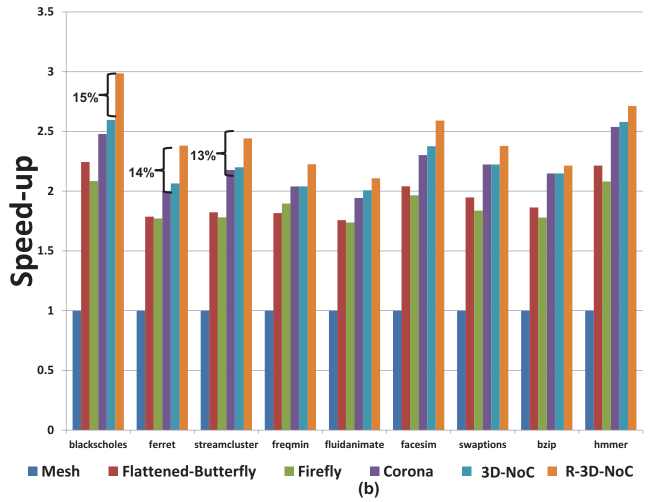
We first describe the simulation setup of the proposed architecture. Our simulator models in detail the router pipeline, arbitration, switching and flow control. An aggressive single cycle electrical router is applied in each tile and the flit transversal time is one cycle from the local core to electrical router [16]. As the delay of Optical/Electrical (O/E) and Electrical/Optical (E/O) conversion can be reduced to less than 100 ps, the total optical transmissions latency is determined by physical location of source/destination pair (1 - 5 cycles) and two additional clock cycles for the conversion delay. In addition, a latency of 1 to 3 cycles was assumed for a tile to capture an optical token. We assume a input buffer of 16 flits with each flit consisting of 128 bits. The packet size is 4 flits which will be sufficient to fit a complete cache line of 64 bytes. We assume a supply voltage V_{dd} of 1.0 V and a router clock frequency of 5 Ghz [3], [7].

We compare 3D-NOC architecture to two other crossbar-like NIs, Corona [3] and Firefly [7] and two electrical interconnects (mesh and Flattened Butterfly) [17]. We implement all architectures such that four cores (one tile) are connected to a single router. We assume token slot for both 3D-NOC and Corona to pipeline the arbitration process to increase the efficiency. Multiple requests can be sent from the four local cores to optical channels to increase the arbitration efficiency. We use Fly_Src routing algorithm [7] for Firefly architectures, where intra-group communication via electrical mesh is implemented first and then inter-group via NIs. For a fair comparison, we ensure that each communication channel in either electrical or optical network is 640 Gbps with 64 wavelengths. For closed-loop measurement, we collect traces from real applications using the full execution-driven simulator SIMICS from WindRiver, with the memory package GEMS enabled [18]. We evaluate the performance of 64- core versions of the networks on Splash-2 [11], PARSEC [12] and SPEC CPU2006 workloads and 256-core version on synthetic and workload completion traffic (a mixture of synthetic traces). We assume a 2 cycle latency to access the L1 cache (64 KB, 4-way), a 4 cycle latency to access the L2 cache (4MB, 16-way), cache line size of 64 bytes and a 160 cycle latency to access the main memory. In addition, there are 16 memory controllers used to access main memory and each processor can issue two threads.

1) Splash-2: 64 Cores: Figure 4(a) shows the speed-up for the Splash-2 applications [11]. 3D-NOC has about an average speed up of about 2.5 for each benchmark over the mesh network. In the water application, 3D-NOC has the highest speed-up with a factor of over 3 relative to mesh. This is a result of 3D-NOC's decomposed crossbars allowing for fast arbitration of network resources (less contention) and the reduced hop count relative to the mesh network. In Raytrace and FMM benchmarks, 3D-NOC has the lowest speed-up factor of 2.2, which is contributed to the higher local (few hops) traffic. Nearest-neighbor traffic creates more contention for optical tokens with locally concentrated destinations in 3D-NOC. When 3D-NOC is compared to flattened-butterfly, 3D-NOC has a 25% - 30% improvement. As flattened-butterfly is a two-hop network and most traffic under Splash-2 suite are two hops, the intermediate router reduces the throughput of the network. When 3D-NOC is compared to Firefly, 3D-NOC outperforms Firefly by about as much as 38% which is a result of Firefly routing its traffic through both an electrical and optical network. R-3D-NOC has about a 5-12% improvement improvement over 3D-NOC for the select range of Splash-2 traffic traces. For FFT and LU applications, R-3D-NOC has the highest performance improvement over 3D-NOC at about 12%. Both FFT and LU have communication patterns that can take advantage of the reconfiguration algorithm as their communication patterns do not quickly change over time forcing the network to keep reconfiguring and improving performance. In the other applications (radiosity, raytrace,



(a)



(b)

Fig. 4. Simulation speed-up for 64-core using (a) SPLASH-2 traffic traces and (b)PARSEC traffic traces.

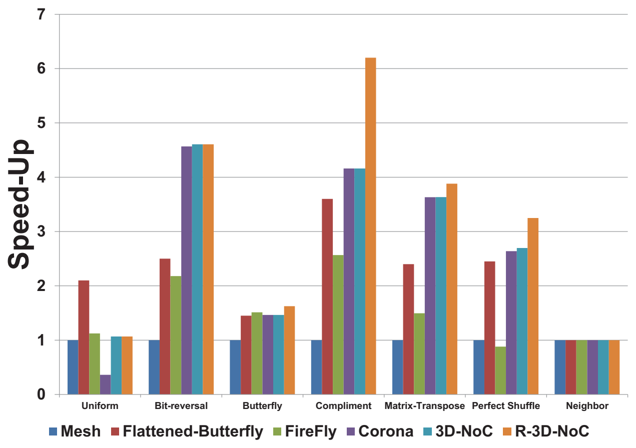


Fig. 5. (Simulation results showing normalized saturation throughput for seven traffic patterns for 256 cores.

radix, ocean, rmm, and water), R-3D-NOC has about a 5% increase in performance over 3D-NOC. This is a direct result of Splash-2 traffic traces resembling uniform traffic, which reduces the bandwidth available for reconfiguration.

2) PARSEC and SPEC CPU2006: 64 Cores: Figure 4(a) shows the speedup for 64 wavelengths. 3D-NOC shows an average of 2X speedup compared to mesh and 10-40% improvement over Flattened-Butterfly and Firefly architectures. When Corona and 3D-NOC are compared to each other, 3D-NOC is able to outperform Corona for most applications except swaptions and bzip. The reason for improved performance over Corona is primarily due to the communication pattern which makes use of all the four decomposed crossbars to be used simultaneously, thereby sending more data on the network when compared to Corona. For swaption and bzip application traffic, their communication patterns do not take advantage of 3D-NOC decomposed crossbars and as such there is no significant improvement. R-3D-NOC shows better improvement in performance for PARSEC/SPEC CPU2006 benchmarks compared to Splash-2 traffic. Blackscholes has the largest jump in performance by almost 20% when compared to Corona and 15% when compared to 3D-NOC. This large increase in performance is contributed to the nature of PARSEC applications which are more communication intensive and therefore, the reconfiguration algorithm maximizes the performance. PARSEC applications emphasize on emerging workloads and future shared-memory applications for the study of CMPs, rather than the network.

3) Synthetic Traffic: 256 Cores: The throughput for all synthetic traffic traces for 256-core implementations are shown in Figure 5 and is normalized to mesh network (for Uniform, the mesh has a throughput of 624 GBytes/sec). 3D-NOC has about a 2.5 $\times$ increase in throughput over Corona for uniform traffic due to the decomposition of the nanophotonic crossbar. The decomposed crossbars allow for a reduction in contention for optical tokens as now a single token is shared between 16 tiles instead of 64 tiles as in Corona. Firefly slightly outperforms 3D-NOC for uniform traffic due to the contention found in the decomposed nanophotonic crossbars. Moveover, Firefly uses a SWMR approach for communication which does not require optical arbitration. From the figure, 3D-NOC slightly outperforms Corona for bit-reversal and complement traffic traces. This is due to lower contention for optical tokens in the decomposed crossbars. 3D-NOC significantly outperforms mesh for the bit-reversal, matrix-transpose and complement traffic patterns. In these traffic patterns, packets need to traversal across multiple mesh routers which in turn increases the packet latency and thereby reduces the throughput. R-3D-NOC is able to out perform 3D-NOC for complement, matrix-transpose and perfect shuffle traffic traces. These permutation traffic traces exhibit adversial patterns which will benefit R-3D-NOC. In complement traffic, R-3D-NOC has about a 55% increase in performance when compared to 3D-NOC. Complement traffic pattern show cases the best performance as a single source tile will communicate with a single destination tile, thereby providing opportunities to improve performance via reconfiguration.

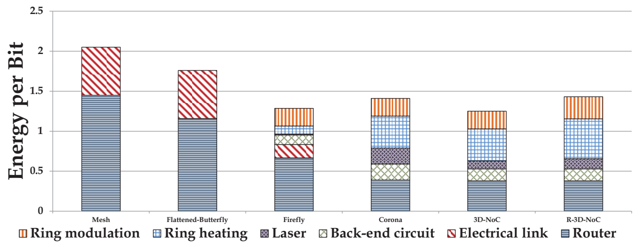


Fig. 6. Average energy per-bit for electrical and NIs.

B. Energy Comparison

The energy consumption of a NI can be divided into two parts, electrical energy and optical energy. Optical energy consists of the off-chip laser energy and on-chip MRRs heating energy. In what follows, we first discuss the electrical energy and then optical energy consumption. We use ORION 2.0 [19] to obtain the energy dissipation values for an electrical link and router and modified their parameters for 22nm technology according to ITRS. We assume all electrical links are optimized for delay and the injection rate to be 0.1. For each optical transmitted bit, we need to provide electrical back end circuit for transmitter end and receiver end. We assume the O/E and E/O converter energy is 100fJ/b, as predicted in [20].

We test uniform traffic with 0.1 injection rate to the four architectures and obtain energy per-bit comparison shown in Figure 6. Although Firefly has $\frac{1}{4}$ as many MRRs as Corona and 3D-NOC, which results in $\frac{1}{4}$ energy consumption per bit on ring heatings, it still consumes more energy per bit than 3D-NOC and CORONA due to the energy consumption overhead of routers and electrical links. In general, 3D-NOC saves 6.5%, 23.1%, 36.1% energy per bit compared to Corona, Firefly, and mesh respectively. It should be noted that when the network injection rate increases, 3D-NOC becomes much more energy efficient than other three architectures. R-3D-NOC has a slight increase in power dissipation over 3D-NOC due to the additional MRRs required for reconfiguration.

V. Conclusions

In this paper, we propose an on-chip multilayer NI called 3D-NOC. 3D-NOC uses emerging NIs and 3D integration to reduce the optical power losses found in 2D planar NoCs by decomposing a large 2D nanophotonic crossbar into multiple smaller nanophotonic crossbar layers. In addition, we proposed R-3D-NOC, a reconfigurable version of 3D-NoC that maximizes the available bandwidth through run-time monitoring of network resources and dynamically re-allocating channel bandwidth. Our results indicate that R-3D-NoC reduces the power dissipation by 23% while improving performance from 10-15% for Splash-2, PARSEC and SPEC CPU2006 benchmarks.

Acknowledgment

This research was partially supported by NSF awards, ECCS-0725765, CCF-0915537, CCF-0915418, CCF-1054339 (CAREER) and ECCS-1129010 and by the IR/D program while Ahmed Louri was serving at the National Science Foundation.

References

[1] J. D. Owens, W. J. Dally, R. Ho, D. N. Jayasimha, S. W. Keckler, and L. S. Peh, "Research challenges for on-chip interconnection networks," *IEEE Micro*, vol. 27, no. 5, pp. 96–108, September-October 2007.

[2] R. G. Beausoleil, P. J. Kuekes, G. S. Snider, S.-Y. Wang, and R. S. Williams, "Nanoelectronic and nanophotonic interconnect," *Proceedings of the IEEE*, vol. 96, no. 2, pp. 230–247, February 2008.

[3] D. Vantrease, R. Schreiber, M. Monchiero, M. McLaren, N. Jouppi, M. Fiorentino, A. Davis, N. Binker, R. Beausoleil, and J. H. Ahn, "Corona: System implications of emerging nanophotonic technology," in *Proceedings of the 35th International Symposium on Computer Architecture*, June 2008, pp. 153–164.

[4] C. Batten, A. Joshi, J. Orcutt, A. Khilo, B. Moss, C. Holzwarth, M. Popovic, H. Li, H. Smith, J. Hoyt, F. Kartner, R. Ram, V. Stojanovi, and K. Asanovic, "Building manycore processor-to-dram networks with monolithic silicon photonics," in *Proceedings of the 16th Annual Symposium on High-Performance Interconnects*, August 27-28 2008.

[5] A. Shacham, K. Bergman, and L. P. Carloni, "Photonic networks-on-chip for future generations of chip multiprocessors," in *IEEE Transactions on Computers*, September 2008, pp. 1246–1260.

[6] R. K. Dokania and A. B. Apsel, "Analysis of challenges for on-chip optical interconnects," in *GLSVLSI '09: Proceedings of the 19th ACM Great Lakes symposium on VLSI*, 2009, pp. 275–280.

[7] Y. Pan, P. Kumar, J. Kim, G. Memik, Y. Zhang, and A. Choudhary, "Firefly: Illuminating future network-on-chip with nanophotonics," in *the Proceedings of the 36th annual International Symposium on Computer Architecture*, 2009.

[8] M. Cianchetti, J. Kerekes, and D. Albonesi, "Phastlane: A rapid transit optical routing network," in *Proceedings of 36th International Symposium on Computer Architecture*, June 2009.

[9] P. Koonath and B. Jalali, "Multilayer 3-d photonics in silicon," *Opt. Express*, vol. 15, pp. 12 686–12 691, 2007.

[10] K. Preston, S. Manipatruni, A. Gondarenko, C. B. Poitras, and M. Lipson, "Deposited silicon high-speed integrated electro-optic modulator," *Opt. Express*, vol. 17, pp. 5118–5124, 2009.

[11] S. Woo, M. Ohara, E. Torrie, J. Singh, and A. Gupta, "The splash-2 program: Characterization and methodological considerations," 1995, pp. 24–36.

[12] C. Bienia, S. Kumar, J. P. Singh, and K. Li, "The parsec benchmark suite: Characterization and architectural implications," in *Proceedings of the 17th International Conference on Parallel Architectures and Compilation Techniques*, October 2008.

[13] W. J. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. San Fransisco, USA: Morgan Kaufmann, 2004.

[14] X. Chen, L.-S. Peh, G.-Y. Wei, Y.-K. Huang, and P. Pruncal, "Exploring the design space of power-aware opto-electronic networked systems," in *11th International Symposium on High-Performance Computer Architecture (HPCA-11)*, February 2005, pp. 120–131.

[15] V. Soteriou, N. Eisley, and L.-S. Peh, "Software-directed power-aware interconnection networks," *ACM Trans. Archit. Code Optim.*, vol. 4, March 2007.

[16] A. Kumar, P. Kundu, A. P. Singh, L.-S. Peh, and N. K. Jha, "A 4.6tbits/s 3.6ghz single-cycle noc router with a novel switch allocator in 65nm cmos," in *ICCD 2007*, October 2007.

[17] J. Kim, W. J. Dally, and D. Abts, "Flattened butterfly: Cost-efficient topology for high-radix networks," in *Proceedings of 34th Annual International Symposium on Computer Architecture(ISCA)*, June 2007, pp. 126 – 137.

[18] M. Martin, D. Sorin, B. Beckmann, M. Marty, M. Xu, A. Alameldeen, K. Moore, M. Hill, and D. Wood, "Multifacet's genreal execution-driven multiprocessor simulator (gems) toolset," *ACM SIGARCH Computer Architecture News*, no. 4, pp. 92–99, November 2005.

[19] 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 *in Proceedings of Design, Automation & Test in Europe Conference & Exhibition*, Nice, France, April 20-24 2009, pp. 423–428.

[20] A. V. Krishnamoorthy, R. Ho, X. Zheng, H. Schwetman, J. Lexau, P. Koka, G. Li, I. Shubin, and J. E. Cunningham, "Computer systems based on silicon photonic interconnects," in *Proceedings of the IEEE*, vol. 97, no. 7, June 2009, pp. 1337–1361.